DRA75x, DRA74x SoC for Automotive Infotainment Silicon Revision 2.0, 1.1 Texas Instruments Jacinto6 Ex, Jacinto6 EP, and Jacinto6 Infotainment Families of Products

Technical Reference Manual

TEXAS INSTRUMENTS

Literature Number: SPRUI30H
NOVEMBER 2015 – REVISED MAY 2024

Table of Contents

Read This First 249
Support Resources 249
Glossary 249
About This Manual 250
Information About Cautions and Warnings 250
Register, Field, and Bit Calls 251
Coding Rules 252
Flow Chart Rules 253
Export Control Notice 253
DRA75x, DRA74x MIPI® Disclaimer 254
Trademarks 255
1 Introduction 256
1.1 DRA75x, DRA74x Overview 257
1.2 DRA75x, DRA74x Environment 259
1.3 DRA75x, DRA74x Description 260
1.3.1 MPU Subsystem 261
1.3.2 DSP Subsystems 261
1.3.3 EVE Subsystems 261
1.3.4 IPU Subsystems 261
1.3.5 IVA-HD Subsystem 262
1.3.6 Display Subsystem 262
1.3.7 Video Processing Subsystem 262
1.3.8 Video Capture 263
1.3.9 3D GPU Subsystem 263
1.3.10 BB2D Subsystem 263
1.3.11 On-Chip Debug Support 264
1.3.12 Power, Reset, and Clock Management 264
1.3.13 On-Chip Memory 265
1.3.14 Memory Management 265
1.3.15 External Memory Interfaces 265
1.3.16 System and Connectivity Peripherals 265
1.3.16.1 System Peripherals 265
1.3.16.2 Media Connectivity Peripherals 266
1.3.16.3 Car Connectivity Peripherals 266
1.3.16.4 Audio Connectivity Peripherals 267
1.3.16.5 Serial Control Peripherals 267
1.3.16.6 Radio Accelerators 267
1.4 DRA75x, DRA74x Family 268
1.5 DRA75x, DRA74x Device Identification 268
1.6 DRA75x, DRA74x Package Characteristics Overview 270
2 Memory Mapping 271
2.1 Introduction 272
2.2 L3_MAIN Memory Map 273
2.2.1 L3_INSTR Memory Map 276
2.3 L4 Memory Map 278
2.3.1 L4_CFG Memory Map 278
2.3.2 L4_WKUP Memory Map 282
2.4 L4_PER Memory Map 283
2.4.1 L4_PER1 Memory Space Mapping 283

2.4.2 L4_PER2 Memory Map....285
2.4.3 L4_PER3 Memory Map....287
2.5 MPU Memory Map....290
2.6 IPU Memory Map....292
2.7 DSP Memory Map....294
2.8 EVE Memory Map....295
2.9 TILER View Memory Map....296
3 Power, Reset, and Clock Management....297
3.1 Device Power Management Introduction....299
3.1.1 Device Power-Management Architecture Building Blocks....299
3.1.1.1 Clock Management....299
3.1.1.1.1 Module Interface and Functional Clocks....299
3.1.1.1.3 Module-Level Clock Management....300
3.1.1.1.4 Clock Domain....305
3.1.1.1.5 Clock Domain-Level Clock Management....306
3.1.1.1.6 Clock Domain HW_AUTO Mode Sequences....307
3.1.1.1.7 Clock Domain Sleep/Wake-up....310
3.1.1.1.8 Clock Domain Dependency....311
3.1.1.2 Power Management....316
3.1.1.2.1 Power Domain....317
3.1.1.2.2 Module Logic and Memory Context....318
3.1.1.2.3 Power Domain Management....318
3.1.1.3 Voltage Management....320
3.1.1.3.1 Voltage Domain....320
3.1.1.3.2 Voltage Domain Management....321
3.1.1.3.3 AVS Overview....321
3.1.2 Power-Management Techniques....323
3.1.2.1 Standby Leakage Management....323
3.1.2.2 Dynamic Voltage and Frequency Scaling....324
3.1.2.3 Dynamic Power Switching....325
3.1.2.4 Adaptive Voltage Scaling....326
3.1.2.5 Adaptive Body Bias....326
3.1.2.6 SR3-APG (Automatic Power Gating)....326
3.1.2.7 Combining Power-Management Techniques....326
3.1.2.7.1 DPS Versus SLM....327
3.2 PRCM Subsystem Overview....328
3.2.1 Introduction....328
3.2.2 Power-Management Framework Features....329
3.3 PRCM Subsystem Environment....330
3.3.1 External Clock Signals....330
3.3.2 External Boot Signals....330
3.3.3 External Reset Signals....331
3.3.4 External Voltage Inputs....332
3.4 PRCM Subsystem Integration....334
3.4.1 Device Power-Management Layout....334
3.4.2 Power-Management Scheme, Reset, and Interrupt Requests....338
3.4.2.1 Power Domain....338
3.4.2.2 Resets....338
3.4.2.3 PRCM Interrupt Requests....338
3.5 Reset Management Functional Description....340
3.5.1 Overview....340
3.5.1.1 PRCM Reset Management Functional Description....340
3.5.1.1.1 Power-On Reset....340
3.5.1.1.2 Warm Reset....340
3.5.1.2 PRM Reset Management Functional Description....340
3.5.2 General Characteristics of Reset Signals....340
3.5.2.1 Scope....340
3.5.2.2 Occurrence....341
3.5.2.3 Source Type....341
3.5.2.4 Retention Type....341
3.5.3 Reset Sources....342

3.5.3.1 Global Reset Sources ... 342
3.5.3.2 Local Reset Sources ... 342
3.5.4 Reset Logging ... 343
3.5.5 Reset Domains ... 343
3.5.6 Reset Sequences ... 367
3.5.6.1 MPU Subsystem Power-On Reset Sequence ... 367
3.5.6.2 MPU Subsystem Warm Reset Sequence ... 368
3.5.6.3 MPU Subsystem Reset Sequence on Sleep and Wake-Up Transitions From RETENTION State ... 369
3.5.6.4 IVA Subsystem Power-On Reset Sequence ... 370
3.5.6.5 IVA Subsystem Software Warm Reset Sequence ... 371
3.5.6.6 DSP1 Subsystem Power-On Reset Sequence ... 372
3.5.6.7 DSP1 Subsystem Software Warm Reset Sequence ... 372
3.5.6.8 DSP2 Subsystem Power-On Reset Sequence ... 373
3.5.6.9 DSP2 Subsystem Software Warm Reset Sequence ... 374
3.5.6.10 IPU1 Subsystem Power-On Reset Sequence ... 374
3.5.6.11 IPU1 Subsystem Software Warm Reset Sequence ... 375
3.5.6.12 IPU2 Subsystem Power-On Reset Sequence ... 376
3.5.6.13 IPU2 Subsystem Software Warm Reset Sequence ... 377
3.5.6.14 EVE1 Subsystem Power-On Reset Sequence ... 378
3.5.6.15 EVE1 Subsystem Software Warm Reset Sequence ... 379
3.5.6.16 EVE2 Subsystem Power-On Reset Sequence ... 379
3.5.6.17 EVE2 Subsystem Software Warm Reset Sequence ... 380
3.5.6.18 Global Warm Reset Sequence ... 381
3.6 Clock Management Functional Description ... 383
3.6.1 Overview ... 383
3.6.2 External Clock Inputs ... 383
3.6.2.1 FUNC_32K_CLK Clock ... 383
3.6.2.2 High-Frequency System Clock Input ... 384
3.6.2.3 External Reference Clock Input ... 384
3.6.3 Internal Clock Sources and Generators ... 384
3.6.3.1 PRM Clock Source ... 384
3.6.3.2 CM Clock Source ... 386
3.6.3.2.1 CM_CORE_AON Clock Generator ... 386
3.6.3.2.2 CM_CORE_AON_CLKOUTMUX Overview ... 392
3.6.3.2.3 CM_CORE_AON_TIMER Overview ... 396
3.6.3.2.4 CM_CORE_AON_MCASP Overview ... 401
3.6.3.3 Generic DPLL Overview ... 404
3.6.3.3.1 Generic APLL Overview ... 404
3.6.3.3.2 DPLLs Output Clocks Parameters ... 405
3.6.3.3.3 Enable Control, Status, and Low-Power Operation Mode ... 407
3.6.3.3.4 DPLL Power Modes ... 407
3.6.3.3.5 DPLL Recalibration ... 409
3.6.3.3.6 DPLL Output Power Down ... 410
3.6.3.4 DPLL_PER Description ... 410
3.6.3.4.1 DPLL_PER Overview ... 410
3.6.3.4.2 DPLL_PER Synthesized Clock Parameters ... 410
3.6.3.4.3 DPLL_PER Power Modes ... 411
3.6.3.4.4 DPLL_PER Recalibration ... 411
3.6.3.5 DPLL_CORE Description ... 412
3.6.3.5.1 DPLL_CORE Overview ... 412
3.6.3.5.2 DPLL_CORE Synthesized Clock Parameters ... 412
3.6.3.5.3 DPLL_CORE Power Modes ... 413
3.6.3.5.4 DPLL_CORE Recalibration ... 413
3.6.3.6 DPLL_ABE Description ... 414
3.6.3.6.1 DPLL_ABE Overview ... 414
3.6.3.6.2 DPLL_ABE Synthesized Clock Parameters ... 414
3.6.3.6.3 DPLL_ABE Power Modes ... 415
3.6.3.6.4 DPLL_ABE Recalibration ... 415
3.6.3.7 DPLL_MPU Description ... 416
3.6.3.7.1 DPLL_MPU Overview ... 416
3.6.3.7.2 DPLL_MPU Tactical Clocking Adjustment ... 416

3.6.3.7.3 DPLL_MPU Synthesized Clock Parameters 416
3.6.3.7.4 DPLL_MPU Power Modes 416
3.6.3.7.5 DPLL_MPU Recalibration 417
3.6.3.8 DPLL_IVA Description 417
3.6.3.8.1 DPLL_IVA Overview 417
3.6.3.8.2 DPLL_IVA Synthesized Clock Parameters 418
3.6.3.8.3 DPLL_IVA Power Modes 418
3.6.3.8.4 DPLL_IVA Recalibration 419
3.6.3.9 DPLL_USB Description 419
3.6.3.9.1 DPLL_USB Overview 419
3.6.3.9.2 DPLL_USB Synthesized Clock Parameters 419
3.6.3.9.3 DPLL_USB Power Modes 420
3.6.3.9.4 DPLL_USB Recalibration 420
3.6.3.10 DPLL_EVE Description 420
3.6.3.10.1 DPLL_EVE Overview 420
3.6.3.10.2 DPLL_EVE Synthesized Clock Parameters 421
3.6.3.10.3 DPLL_EVE Power Modes 421
3.6.3.10.4 DPLL_EVE Recalibration 421
3.6.3.11 DPLL_DSP Description 422
3.6.3.11.1 DPLL_DSP Overview 422
3.6.3.11.2 DPLL_DSP Synthesized Clock Parameters 422
3.6.3.11.3 DPLL_DSP Power Modes 422
3.6.3.11.4 DPLL_DSP Recalibration 423
3.6.3.12 DPLL_GMAC Description 423
3.6.3.12.1 DPLL_GMAC Overview 423
3.6.3.12.2 DPLL_GMAC Synthesized Clock Parameters 424
3.6.3.12.3 DPLL_GMAC Power Modes 425
3.6.3.12.4 DPLL_GMAC Recalibration 425
3.6.3.13 DPLL_GPU Description 425
3.6.3.13.1 DPLL_GPU Overview 425
3.6.3.13.2 DPLL_GPU Synthesized Clock Parameters 426
3.6.3.13.3 DPLL_GPU Power Modes 426
3.6.3.13.4 DPLL_GPU Recalibration 427
3.6.3.14 DPLL_DDR Description 427
3.6.3.14.1 DPLL_DDR Overview 427
3.6.3.14.2 DPLL_DDR Synthesized Clock Parameters 427
3.6.3.14.3 DPLL_DDR Power Modes 428
3.6.3.14.4 DPLL_DDR Recalibration 428
3.6.3.15 DPLL_PCIE_REF Description 429
3.6.3.15.1 DPLL_PCIE_REF Overview 429
3.6.3.15.2 DPLL_PCIE_REF Synthesized Clock Parameters 429
3.6.3.15.3 DPLL_PCIE_REF Power Modes 429
3.6.3.16 APLL_PCIE Description 430
3.6.3.16.1 APLL_PCIE Overview 430
3.6.3.16.2 APLL_PCIE Synthesized Clock Parameters 430
3.6.3.16.3 APLL_PCIE Power Modes 430
3.6.4 Clock Domains 431
3.6.4.1 CD_WKUPAON Clock Domain 431
3.6.4.1.1 Overview 431
3.6.4.1.2 Clock Domain Modes 431
3.6.4.1.3 Clock Domain Dependency 432
3.6.4.1.4 Clock Domain Module Attributes 435
3.6.4.2 CD_DSP1 Clock Domain 437
3.6.4.2.1 Overview 437
3.6.4.2.2 Clock Domain Modes 437
3.6.4.2.3 Clock Domain Dependency 438
3.6.4.2.4 Clock Domain Module Attributes 439
3.6.4.3 CD_DSP2 Clock Domain 440
3.6.4.3.1 Overview 440
3.6.4.3.2 Clock Domain Modes 440
3.6.4.3.3 Clock Domain Dependency 440

3.6.4.3.4 Clock Domain Module Attributes....442
3.6.4.4 CD_CUSTEFUSE Clock Domain....442
3.6.4.4.1 Overview....442
3.6.4.4.2 Clock Domain Modes....443
3.6.4.4.3 Clock Domain Dependency....443
3.6.4.4.4 Clock Domain Module Attributes....443
3.6.4.5 CD_MPU Clock Domain....444
3.6.4.5.1 Overview....444
3.6.4.5.2 Clock Domain Modes....444
3.6.4.5.3 Clock Domain Dependency....445
3.6.4.5.4 Clock Domain Module Attributes....446
3.6.4.6 CD_L4PER1 Clock Domain....447
3.6.4.6.1 Overview....447
3.6.4.6.2 Clock Domain Modes....447
3.6.4.6.3 Clock Domain Dependency....448
3.6.4.6.4 Clock Domain Module Attributes....472
3.6.4.7 CD_L4PER2 Clock Domain....478
3.6.4.7.1 Overview....478
3.6.4.7.2 Clock Domain Modes....479
3.6.4.7.3 Clock Domain Dependency....480
3.6.4.7.4 Clock Domain Module Attributes....489
3.6.4.8 CD_L4PER3 Clock Domain....492
3.6.4.8.1 Overview....492
3.6.4.8.2 Clock Domain Modes....493
3.6.4.8.3 Clock Domain Dependency....493
3.6.4.8.4 Clock Domain Module Attributes....494
3.6.4.9 CD_L4SEC Clock Domain....495
3.6.4.9.1 Overview....495
3.6.4.9.2 Clock Domain Modes....496
3.6.4.9.3 Clock Domain Dependency....496
3.6.4.9.4 Clock Domain Module Attributes....497
3.6.4.10 CD_L3INIT Clock Domain....498
3.6.4.10.1 Overview....498
3.6.4.10.2 Clock Domain Modes....498
3.6.4.10.3 Clock Domain Dependency....499
3.6.4.10.4 Clock Domain Module Attributes....505
3.6.4.11 CD_IVA Clock Domain....508
3.6.4.11.1 Overview....508
3.6.4.11.2 Clock Domain Modes....508
3.6.4.11.3 Clock Domain Dependency....509
3.6.4.11.4 Clock Domain Module Attributes....509
3.6.4.12 CD_GPU Description....510
3.6.4.12.1 Overview....510
3.6.4.12.2 Clock Domain Modes....510
3.6.4.12.3 Clock Domain Dependency....511
3.6.4.12.4 Clock Domain Module Attributes....511
3.6.4.13 CD_EMU Clock Domain....512
3.6.4.13.1 Overview....512
3.6.4.13.2 Clock Domain Modes....512
3.6.4.13.3 Clock Domain Dependency....513
3.6.4.13.4 Clock Domain Module Attributes....513
3.6.4.14 CD_DSS Clock Domain....513
3.6.4.14.1 Overview....513
3.6.4.14.2 Clock Domain Modes....514
3.6.4.14.3 Clock Domain Dependency....515
3.6.4.14.4 Clock Domain Module Attributes....519
3.6.4.15 CD_L4_CFG Clock Domain....520
3.6.4.15.1 Overview....520
3.6.4.15.2 Clock Domain Modes....520
3.6.4.15.3 Clock Domain Dependency....521
3.6.4.15.4 Clock Domain Module Attributes....521

3.6.4.16 CD_L3_INSTR Clock Domain....523
3.6.4.16.1 Overview....523
3.6.4.16.2 Clock Domain Modes....524
3.6.4.16.3 Clock Domain Dependency....524
3.6.4.16.4 Clock Domain Module Attributes....524
3.6.4.17 CD_L3_MAIN1 Clock Domain....526
3.6.4.17.1 Overview....526
3.6.4.17.2 Clock Domain Modes....526
3.6.4.17.3 Clock Domain Dependency....527
3.6.4.17.4 Clock Domain Module Attributes....527
3.6.4.18 CD_EMIF Clock Domain....530
3.6.4.18.1 Overview....530
3.6.4.18.2 Clock Domain Modes....530
3.6.4.18.3 Clock Domain Dependency....531
3.6.4.18.4 Clock Domain Module Attributes....531
3.6.4.19 CD_IPU Clock Domain....532
3.6.4.19.1 Overview....532
3.6.4.19.2 Clock Domain Modes....532
3.6.4.19.3 Clock Domain Dependency....533
3.6.4.19.4 Clock Domain Module Attributes....534
3.6.4.20 CD_IPU1 Clock Domain....536
3.6.4.20.1 Overview....536
3.6.4.20.2 Clock Domain Modes....536
3.6.4.20.3 Clock Domain Dependency....537
3.6.4.20.4 Clock Domain Module Attributes....538
3.6.4.21 CD_IPU2 Clock Domain....539
3.6.4.21.1 Overview....539
3.6.4.21.2 Clock Domain Modes....539
3.6.4.21.3 Clock Domain Dependency....539
3.6.4.21.4 Clock Domain Module Attributes....541
3.6.4.22 CD_DMA Clock Domain....541
3.6.4.22.1 Overview....541
3.6.4.22.2 Clock Domain Modes....542
3.6.4.22.3 Clock Domain Dependency....542
3.6.4.22.4 Clock Domain Module Attributes....543
3.6.4.23 CD_ATL Clock Domain....544
3.6.4.23.1 Overview....544
3.6.4.23.2 Clock Domain Modes....544
3.6.4.23.3 Clock Domain Module Attributes....544
3.6.4.24 CD_CAM Clock Domain....545
3.6.4.24.1 Overview....545
3.6.4.24.2 Clock Domain Modes....546
3.6.4.24.3 Clock Domain Dependency....546
3.6.4.24.4 Clock Domain Module Attributes....547
3.6.4.25 CD_GMAC Clock Domain....548
3.6.4.25.1 Overview....548
3.6.4.25.2 Clock Domain Modes....549
3.6.4.25.3 Clock Domain Dependency....549
3.6.4.25.4 Clock Domain Module Attributes....549
3.6.4.26 CD_VPE Clock Domain....550
3.6.4.26.1 CD_VPE Overview....550
3.6.4.26.2 Clock Domain Modes....551
3.6.4.26.3 Clock Domain Dependency....551
3.6.4.26.4 Clock Domain Module Attributes....552
3.6.4.27 CD_EVE1 Clock Domain....553
3.6.4.27.1 CD_EVE1 Overview....553
3.6.4.27.2 Clock Domain Modes....553
3.6.4.27.3 Clock Domain Dependency....553
3.6.4.27.4 Clock Domain Module Attributes....554
3.6.4.28 CD_EVE2 Clock Domain....555
3.6.4.28.1 CD_EVE2 Overview....555

3.6.4.28.2 Clock Domain Modes....555
3.6.4.28.3 Clock Domain Dependency....556
3.6.4.28.4 Clock Domain Module Attributes....556
3.6.4.29 CD_RTC Clock Domain....557
3.6.4.29.1 CD_RTC Overview....557
3.6.4.29.2 Clock Domain Modes....558
3.6.4.29.3 Clock Domain Dependency....558
3.6.4.29.4 Clock Domain Module Attributes....559
3.6.4.30 CD_PCIE Clock Domain....560
3.6.4.30.1 CD_PCIE Overview....560
3.6.4.30.2 Clock Domain Modes....560
3.6.4.30.3 Clock Domain Dependency....561
3.6.4.30.4 Clock Domain Module Attributes....563
3.7 Power Management Functional Description....565
3.7.1 PD_WKUPAON Description....565
3.7.1.1 Power Domain Modes....566
3.7.1.1.1 Logic and Memory Area Power Modes....566
3.7.2 PD_DSP1 Description....566
3.7.2.1 Power Domain Modes....566
3.7.2.1.1 Logic and Memory Area Power Modes....566
3.7.2.1.2 Logic and Memory Area Power Modes Control and Status....567
3.7.3 PD_DSP2 Description....568
3.7.3.1 Power Domain Modes....568
3.7.3.1.1 Logic and Memory Area Power Modes....568
3.7.3.1.2 Logic and Memory Area Power Modes Control and Status....568
3.7.4 PD_CUSTEFUSE Description....569
3.7.4.1 Power Domain Modes....569
3.7.4.1.1 Logic and Memory Area Power Modes....569
3.7.4.1.2 Logic and Memory Area Power Modes Control and Status....570
3.7.5 PD_MPU Description....570
3.7.5.1 Power Domain Modes....570
3.7.5.1.1 Logic and Memory Area Power Modes....570
3.7.5.1.2 Logic and Memory Area Power Modes Control and Status....571
3.7.5.1.3 Power State Override....571
3.7.6 PD_IPU Description....572
3.7.6.1 Power Domain Modes....572
3.7.6.1.1 Logic and Memory Area Power Modes....572
3.7.6.1.2 Logic and Memory Area Power Modes Control and Status....573
3.7.7 PD_L3INIT Description....574
3.7.7.1 Power Domain Modes....574
3.7.7.1.1 Logic and Memory Area Power Modes....574
3.7.7.1.2 Logic and Memory Area Power Modes Control and Status....575
3.7.8 PD_L4PER Description....576
3.7.8.1 Power Domain Modes....578
3.7.8.1.1 Logic and Memory Area Power Modes....578
3.7.8.1.2 Logic and Memory Area Power Modes Control and Status....579
3.7.9 PD_IVA Description....580
3.7.9.1 Power Domain Modes....580
3.7.9.1.1 Logic and Memory Area Power Modes....580
3.7.9.1.2 Logic and Memory Area Power Modes Control and Status....581
3.7.10 PD_GPU Description....582
3.7.10.1 Power Domain Modes....582
3.7.10.1.1 Logic and Memory Area Power Modes....582
3.7.10.1.2 Logic and Memory Area Power Modes Control and Status....582
3.7.11 PD_EMU Description....583
3.7.11.1 Power Domain Modes....583
3.7.11.1.1 Logic and Memory Area Power Modes....583
3.7.11.1.2 Logic and Memory Area Power Modes Control and Status....583
3.7.12 PD_DSS Description....584
3.7.12.1 Power Domain Modes....584
3.7.12.1.1 Logic and Memory Area Power Modes....584

3.7.12.1.2 Logic and Memory Area Power Mode Control and Status....... 585
3.7.13 PD_CORE Description....... 585
3.7.13.1 Power Domain Modes....... 587
3.7.13.1.1 Logic and Memory Area Power Modes....... 587
3.7.13.1.2 Logic and Memory Area Power Mode Control and Status....... 588
3.7.14 PD_CAM Description....... 589
3.7.14.1 Power Domain Modes....... 590
3.7.14.1.1 Logic and Memory Area Power Modes....... 590
3.7.14.1.2 Logic and Memory Area Power Mode Control and Status....... 590
3.7.15 PD_MPUAON Description....... 590
3.7.15.1 Power Domain Modes....... 591
3.7.16 PD_MMAON Description....... 591
3.7.16.1 Power Domain Modes....... 591
3.7.17 PD_COREAON Description....... 591
3.7.17.1 Power Domain Modes....... 592
3.7.18 PD_VPE Description....... 592
3.7.18.1 Power Domain Modes....... 592
3.7.18.1.1 Logic and Memory Area Power Modes....... 592
3.7.18.1.2 Logic and Memory Area Power Modes Control and Status....... 593
3.7.19 PD_EVE1 Description....... 593
3.7.19.1 Power Domain Modes....... 593
3.7.19.1.1 Logic and Memory Area Power Modes....... 593
3.7.19.1.2 Logic and Memory Area Power Modes Control and Status....... 594
3.7.20 PD_EVE2 Description....... 594
3.7.20.1 Power Domain Modes....... 595
3.7.20.1.1 Logic and Memory Area Power Modes....... 595
3.7.20.1.2 Logic and Memory Area Power Modes Control and Status....... 595
3.7.21 PD_RTC Description....... 595
3.7.21.1 Power Domain Modes....... 596
3.7.21.1.1 Logic and Memory Area Power Modes....... 596
3.8 Voltage-Management Functional Description....... 597
3.8.1 Overview....... 597
3.8.2 Voltage-Control Architecture....... 597
3.8.3 Internal LDOs Control....... 598
3.8.3.1 VDD_MPU_L, VDD_CORE_L, and VDD_IVAHD_L, VDD_GPU_L, VDD_DSPEVE_L Control....... 598
3.8.3.1.1 Adaptive Voltage Scaling....... 598
3.8.3.2 Memory LDOs....... 599
3.8.3.3 ABB LDOs Control....... 600
3.8.3.4 ABB LDO Programming Sequence....... 600
3.8.3.4.1 ABB LDO Enable Sequence....... 600
3.8.3.4.2 ABB LDO Disable Sequence (Entering in Bypass Mode)....... 601
3.8.3.5 BANDGAPs Control....... 601
3.8.4 DVFS....... 601
3.9 Device Low-Power States....... 602
3.9.1 Device Wake-Up Source Summary....... 602
3.9.2 Wakeup Upon Global Warm Reset....... 604
3.9.3 Global Warm Reset During a Device Wake-Up Sequence....... 604
3.9.4 I/O Management....... 604
3.9.4.1 Isolation / Wakeup Sequence....... 605
3.9.4.1.1 Software-Controlled I/O Isolation....... 606
3.10 PRCM Module Programming Guide....... 607
3.10.1 DPLLs Low-Level Programming Models....... 607
3.10.1.1 Global Initialization....... 607
3.10.1.1.1 Surrounding Module Global Initialization....... 607
3.10.1.1.2 DPLL Global Initialization....... 607
3.10.1.2 DPLL Output Frequency Change....... 608
3.10.2 Clock Management Low-Level Programming Models....... 610
3.10.2.1 Global Initialization....... 610
3.10.2.1.1 Surrounding Module Global Initialization....... 610
3.10.2.1.2 Clock Management Global Initialization....... 610
3.10.2.2 Clock Domain Sleep Transition and Troubleshooting....... 611

3.10.2.3 Enable/Disable Software-Programmable Static Dependency....611
3.10.3 Power Management Low-Level Programming Models....611
3.10.3.1 Global Initialization....612
3.10.3.1.1 Surrounding Module Global Initialization....612
3.10.3.1.2 Power Management Global Initialization....612
3.10.3.2 Forced Memory Area State Change With Power Domain ON....612
3.10.3.3 Forced Power Domain Low-Power State Transition....612
3.12 PRCM Software Configuration for OPP_PLUS....613
3.13 PRCM Register Manual....613
3.13.1 PRCM Instance Summary....613
3.13.2 CM_CORE_AON__CKGEN Registers....614
3.13.2.1 CM_CORE_AON__CKGEN Register Summary....614
3.13.2.2 CM_CORE_AON__CKGEN Register Description....616
3.13.3 CM_CORE_AON__DSP1 Registers....658
3.13.3.1 CM_CORE_AON__DSP1 Register Summary....658
3.13.3.2 CM_CORE_AON__DSP1 Register Description....658
3.13.4 CM_CORE_AON__DSP2 Registers....663
3.13.4.1 CM_CORE_AON__DSP2 Register Summary....663
3.13.4.2 CM_CORE_AON__DSP2 Register Description....663
3.13.5 CM_CORE_AON__EVE1 Registers....668
3.13.5.1 CM_CORE_AON__EVE1 Register Summary....668
3.13.5.2 CM_CORE_AON__EVE1 Register Description....668
3.13.6 CM_CORE_AON__EVE2 Registers....671
3.13.6.1 CM_CORE_AON__EVE2 Register Summary....671
3.13.6.2 CM_CORE_AON__EVE2 Register Description....671
3.13.7 CM_CORE_AON__INSTR Registers....673
3.13.7.1 CM_CORE_AON__INSTR Register Summary....673
3.13.7.2 CM_CORE_AON__INSTR Register Description....674
3.13.8 CM_CORE_AON__IPU Registers....677
3.13.8.1 CM_CORE_AON__IPU Register Summary....677
3.13.8.2 CM_CORE_AON__IPU Register Description....678
3.13.9 CM_CORE_AON__MPU Registers....692
3.13.9.1 CM_CORE_AON__MPU Register Summary....692
3.13.9.2 CM_CORE_AON__MPU Register Description....692
3.13.10 CM_CORE_AON__OCP_SOCKET Registers....697
3.13.10.1 CM_CORE_AON__OCP_SOCKET Register Summary....697
3.13.10.2 CM_CORE_AON__OCP_SOCKET Register Description....698
3.13.11 CM_CORE_AON__RESTORE Registers....700
3.13.11.1 CM_CORE_AON__RESTORE Register Summary....700
3.13.11.2 CM_CORE_AON__RESTORE Register Description....701
3.13.12 CM_CORE_AON__RTC Registers....705
3.13.12.1 CM_CORE_AON__RTC Register Summary....705
3.13.12.2 CM_CORE_AON__RTC Register Description....705
3.13.13 CM_CORE_AON__VPE Registers....707
3.13.13.1 CM_CORE_AON__VPE Register Summary....707
3.13.13.2 CM_CORE_AON__VPE Register Description....707
3.13.14 CM_CORE__CAM Registers....710
3.13.14.1 CM_CORE__CAM Register Summary....710
3.13.14.2 CM_CORE__CAM Register Description....710
3.13.15 CM_CORE__CKGEN Registers....716
3.13.15.1 CM_CORE__CKGEN Register Summary....716
3.13.15.2 CM_CORE__CKGEN Register Description....717
3.13.16 CM_CORE__COREAON Registers....735
3.13.16.1 CM_CORE__COREAON Register Summary....735
3.13.16.2 CM_CORE__COREAON Register Description....735
3.13.17 CM_CORE__CORE Registers....741
3.13.17.1 CM_CORE__CORE Register Summary....741
3.13.17.2 CM_CORE__CORE Register Description....742
3.13.18 CM_CORE__CUSTEFUSE Registers....788
3.13.18.1 CM_CORE__CUSTEFUSE Register Summary....788
3.13.18.2 CM_CORE__CUSTEFUSE Register Description....788

3.13.19 CM_CORE__DSS Registers....790
3.13.19.1 CM_CORE__DSS Register Summary....790
3.13.19.2 CM_CORE__DSS Register Description....790
3.13.20 CM_CORE__GPU Registers....795
3.13.20.1 CM_CORE__GPU Register Summary....795
3.13.20.2 CM_CORE__GPU Register Description....796
3.13.21 CM_CORE__IVA Registers....799
3.13.21.1 CM_CORE__IVA Register Summary....799
3.13.21.2 CM_CORE__IVA Register Description....799
3.13.22 CM_CORE__L3INIT Registers....803
3.13.22.1 CM_CORE__L3INIT Register Summary....803
3.13.22.2 CM_CORE__L3INIT Register Description....804
3.13.23 CM_CORE__L4PER Registers....829
3.13.23.1 CM_CORE__L4PER Register Summary....829
3.13.23.2 CM_CORE__L4PER Register Description....830
3.13.24 CM_CORE__OCP_SOCKET Registers....901
3.13.24.1 CM_CORE__OCP_SOCKET Register Summary....901
3.13.24.2 CM_CORE__OCP_SOCKET Register Description....901
3.13.25 CM_CORE__RESTORE Registers....903
3.13.25.1 CM_CORE__RESTORE Register Summary....903
3.13.25.2 CM_CORE__RESTORE Register Description....903
3.13.26 CAM_PRM Registers....907
3.13.26.1 CAM_PRM Register Summary....907
3.13.26.2 CAM_PRM Register Description....908
3.13.27 CKGEN_PRM Registers....914
3.13.27.1 CKGEN_PRM Register Summary....914
3.13.27.2 CKGEN_PRM Register Description....915
3.13.28 CORE_PRM Registers....942
3.13.28.1 CORE_PRM Register Summary....942
3.13.28.2 CORE_PRM Register Description....943
3.13.29 CUSTEFUSE_PRM Registers....976
3.13.29.1 CUSTEFUSE_PRM Register Summary....976
3.13.29.2 CUSTEFUSE_PRM Register Description....977
3.13.30 DEVICE_PRM Registers....979
3.13.30.1 DEVICE_PRM Register Summary....979
3.13.30.2 DEVICE_PRM Register Description....980
3.13.31 DSP1_PRM Registers....1002
3.13.31.1 DSP1_PRM Register Summary....1002
3.13.31.2 DSP1_PRM Register Description....1002
3.13.32 DSP2_PRM Registers....1006
3.13.32.1 DSP2_PRM Register Summary....1006
3.13.32.2 DSP2_PRM Register Description....1006
3.13.33 DSS_PRM Registers....1010
3.13.33.1 DSS_PRM Register Summary....1010
3.13.33.2 DSS_PRM Register Description....1010
3.13.34 EMU_CM Registers....1019
3.13.34.1 EMU_CM Register Summary....1019
3.13.34.2 EMU_CM Register Description....1020
3.13.35 EMU_PRM Registers....1022
3.13.35.1 EMU_PRM Register Summary....1022
3.13.35.2 EMU_PRM Register Description....1023
3.13.36 EVE1_PRM Registers....1025
3.13.36.1 EVE1_PRM Register Summary....1025
3.13.36.2 EVE1_PRM Register Description....1025
3.13.37 EVE2_PRM Registers....1030
3.13.37.1 EVE2_PRM Register Summary....1030
3.13.37.2 EVE2_PRM Register Description....1030
3.13.38 GPU_PRM Registers....1034
3.13.38.1 GPU_PRM Register Summary....1034
3.13.38.2 GPU_PRM Register Description....1035
3.13.39 INSTR_PRM Registers....1037

3.13.39.1 INSTR_PRM Register Summary............ 1037
3.13.39.2 INSTR_PRM Register Description............ 1038
3.13.40 IPU_PRM Registers............ 1041
3.13.40.1 IPU_PRM Register Summary............ 1041
3.13.40.2 IPU_PRM Register Description............ 1041
3.13.41 IVA_PRM Registers............ 1058
3.13.41.1 IVA_PRM Register Summary............ 1058
3.13.41.2 IVA_PRM Register Description............ 1058
3.13.42 L3INIT_PRM Registers............ 1064
3.13.42.1 L3INIT_PRM Register Summary............ 1064
3.13.42.2 L3INIT_PRM Register Description............ 1064
3.13.43 L4PER_PRM Registers............ 1089
3.13.43.1 L4PER_PRM Register Summary............ 1089
3.13.43.2 L4PER_PRM Register Description............ 1092
3.13.44 MPU_PRM Registers............ 1200
3.13.44.1 MPU_PRM Register Summary............ 1200
3.13.44.2 MPU_PRM Register Description............ 1200
3.13.45 OCP_SOCKET_PRM Registers............ 1203
3.13.45.1 OCP_SOCKET_PRM Register Summary............ 1203
3.13.45.2 OCP_SOCKET_PRM Register Description............ 1204
3.13.46 RTC_PRM Registers............ 1235
3.13.46.1 RTC_PRM Register Summary............ 1235
3.13.46.2 RTC_PRM Register Description............ 1235
3.13.47 VPE_PRM Registers............ 1238
3.13.47.1 VPE_PRM Register Summary............ 1238
3.13.47.2 VPE_PRM Register Description............ 1238
3.13.48 WKUPAON_CM Registers............ 1242
3.13.48.1 WKUPAON_CM Register Summary............ 1242
3.13.48.2 WKUPAON_CM Register Description............ 1243
3.13.49 WKUPAON_PRM Registers............ 1252
3.13.49.1 WKUPAON_PRM Register Summary............ 1252
3.13.49.2 WKUPAON_PRM Register Description............ 1253
4 Dual Cortex-A15 MPU Subsystem............ 1269
4.1 Dual Cortex-A15 MPU Subsystem Overview............ 1270
4.1.1 Introduction............ 1270
4.1.2 Features............ 1272
4.2 Dual Cortex-A15 MPU Subsystem Integration............ 1274
4.2.1 Clock Distribution............ 1274
4.2.2 Reset Distribution............ 1276
4.3 Dual Cortex-A15 MPU Subsystem Functional Description............ 1278
4.3.1 MPU Subsystem Block Diagram............ 1278
4.3.2 Cortex-A15 MPCore (MPU_CLUSTER)............ 1278
4.3.2.1 MPU L2 Cache Memory System............ 1279
4.3.2.1.1 MPU L2 Cache Architecture............ 1280
4.3.2.1.2 MPU L2 Cache Controller............ 1280
4.3.3 MPU_AXI2OCP............ 1281
4.3.4 Memory Adapter............ 1281
4.3.4.1 MPU_MA Overview............ 1281
4.3.4.2 AXI Input Interface............ 1283
4.3.4.3 Interleaving............ 1284
4.3.4.3.1 High-Order Fixed Interleaving Model............ 1284
4.3.4.3.2 Lower 2-GiB Programmable Interleaving Model............ 1286
4.3.4.3.3 Local Interconnect and Synchronization Agent (LISA) Section Manager............ 1286
4.3.4.3.4 MA_LSM Registers............ 1286
4.3.4.3.5 Posted and Nonposted Writes............ 1287
4.3.4.3.6 Errors............ 1287
4.3.4.4 Statistics Collector Probe Ports............ 1288
4.3.4.5 MPU_MA Firewall............ 1288
4.3.4.6 MPU_MA Power and Reset Management............ 1288
4.3.4.7 MPU_MA Watchpoint............ 1288
4.3.4.7.1 Watchpoint Types............ 1288

4.3.4.7.2 Transaction Filtering Options....1288
4.3.4.7.3 Transaction Match Effects....1289
4.3.4.7.4 Trigger Generation....1289
4.3.4.7.5 Programming Options Summary....1290
4.3.5 Realtime Counter (Master Counter)....1290
4.3.5.1 Counter Operation....1290
4.3.5.2 Frequency Change Procedure....1291
4.3.6 MPU Watchdog Timer....1292
4.3.7 MPU Subsystem Power Management....1293
4.3.7.1 Power Domains....1293
4.3.7.2 Power States of MPU_Cx....1294
4.3.7.3 Power States of MPU Subsystem....1298
4.3.7.4 MPU_WUGEN....1299
4.3.7.5 Power Transition Sequence....1299
4.3.7.6 SR3-APG Technology Fail-Safe Mode....1300
4.3.8 MPU Subsystem AMBA Interface Configuration....1300
4.4 Dual Cortex-A15 MPU Subsystem Register Manual....1301
4.4.1 Dual Cortex-A15 MPU Subsystem Instance Summary....1301
4.4.2 MPU_CS_STM Registers....1301
4.4.3 MPU_INTC Registers....1301
4.4.4 MPU_PRCM_OCP_SOCKET Registers....1301
4.4.4.1 MPU_PRCM_OCP_SOCKET Register Summary....1301
4.4.4.2 MPU_PRCM_OCP_SOCKET Register Description....1301
4.4.5 MPU_PRCM_DEVICE Registers....1302
4.4.5.1 MPU_PRCM_DEVICE Register Summary....1302
4.4.5.2 MPU_PRCM_DEVICE Register Description....1302
4.4.6 MPU_PRCM_PRM_C0 Registers....1304
4.4.6.1 MPU_PRCM_PRM_C0 Register Summary....1304
4.4.6.2 MPU_PRCM_PRM_C0 Register Description....1304
4.4.7 MPU_PRCM_CM_C0 Registers....1308
4.4.7.1 MPU_PRCM_CM_C0 Register Summary....1308
4.4.7.2 MPU_PRCM_CM_C0 Register Description....1308
4.4.8 MPU_PRCM_PRM_C1 Registers....1309
4.4.8.1 MPU_PRCM_PRM_C1 Register Summary....1309
4.4.8.2 MPU_PRCM_PRM_C1 Register Description....1309
4.4.9 MPU_PRCM_CM_C1 Registers....1312
4.4.9.1 MPU_PRCM_CM_C1 Register Summary....1312
4.4.9.2 MPU_PRCM_CM_C1 Register Description....1312
4.4.10 MPU_WUGEN Registers....1313
4.4.10.1 MPU_WUGEN Register Summary....1313
4.4.10.2 MPU_WUGEN Register Description....1314
4.4.11 MPU_WD_TIMER Registers....1331
4.4.11.1 MPU_WD_TIMER Register Summary....1331
4.4.11.2 MPU_WD_TIMER Register Description....1331
4.4.12 MPU_AXI2OCP_MISC Registers....1335
4.4.12.1 MPU_AXI2OCP_MISC Register Summary....1335
4.4.12.2 MPU_AXI2OCP_MISC Register Description....1335
4.4.13 MPU_MA_LSM Registers....1335
4.4.13.1 MPU_MA_LSM Register Summary....1335
4.4.13.2 MPU_MA_LSM Register Description....1335
4.4.14 MPU_MA_WP Registers....1335
4.4.14.1 MPU_MA_WP Register Summary....1336
4.4.14.2 MPU_MA_WP Register Description....1336
5 DSP Subsystems....1348
5.1 DSP Subsystems Overview....1349
5.1.1 DSP Subsystems Key Features....1350
5.2 DSP Subsystem Integration....1354
5.3 DSP Subsystems Functional Description....1360
5.3.1 DSP Subsystems Block Diagram....1360
5.3.2 DSP Subsystem Components....1361
5.3.2.1 C66x DSP Subsystem Introduction....1361

5.3.2.2 DSP TMS320C66x CorePac....1362
5.3.2.2.1 DSP TMS320C66x CorePac CPU....1362
5.3.2.2.2 DSP TMS320C66x CorePac Internal Memory Controllers and Memories....1362
5.3.2.2.3 DSP C66x CorePac Internal Peripherals....1363
5.3.2.3 DSP Debug and Trace Support....1369
5.3.2.3.1 DSP Advanced Event Triggering (AET)....1369
5.3.2.3.2 DSP Trace Support....1369
5.3.3 DSP System Control Logic....1369
5.3.3.1 DSP System Clocks....1370
5.3.3.2 DSP Hardware Resets....1371
5.3.3.3 DSP Software Resets....1372
5.3.3.4 DSP Power Management....1372
5.3.3.4.1 DSP System Powerdown Protocols....1372
5.3.3.4.2 DSP Software and Hardware Power Down Sequence Overview....1372
5.3.3.4.3 DSP IDLE Wakeup....1375
5.3.3.4.4 DSP SYSTEM IRQWAKEEN registers....1375
5.3.3.4.5 DSP Automatic Power Transition....1375
5.3.4 DSP Interrupt Requests....1376
5.3.4.1 DSP Input Interrupts....1377
5.3.4.1.1 DSP Non-maskable Interrupt Input....1377
5.3.4.2 DSP Event and Interrupt Generation Outputs....1378
5.3.4.2.1 DSP MDMA and DSP EDMA Mflag Event Outputs....1378
5.3.4.2.2 DSP Aggregated Error Interrupt Output....1378
5.3.4.2.3 Non-DSP C66x CorePac Generated Peripheral Interrupt Outputs....1380
5.3.5 DSP DMA Requests....1380
5.3.5.1 DSP EDMA Wakeup Interrupt....1385
5.3.6 DSP Intergated Memory Management Units....1385
5.3.6.1 DSP MMUs Overview....1385
5.3.6.2 Routing MDMA Traffic through DSP MMU0....1386
5.3.6.3 Routing EDMA Traffic thorugh DSP MMU1....1386
5.3.7 DSP Integrated EDMA Subsystem....1387
5.3.7.1 DSP EDMA Overview....1387
5.3.7.2 DSP System and Device Level Settings of DSP EDMA....1388
5.3.8 DSP L2 interconnect Network....1390
5.3.8.1 DSP Public Firewall Settings....1391
5.3.8.2 DSP NoC Flag Mux and Error Log Registers....1391
5.3.8.3 DSP NoC Arbitration....1391
5.3.9 DSP Boot Configuration....1392
5.3.10 DSP Internal and External Memory Views....1392
5.3.10.1 C66x CPU View of the Address Space....1392
5.3.10.2 DSP_EDMA View of the Address Space....1394
5.3.10.3 L3_MAIN View of the DSP Address Space....1395
5.4 DSP Subsystem Register Manual....1396
5.4.1 DSP Subsystem Instance Summary....1396
5.4.2 DSP_ICFG Registers....1397
5.4.2.1 DSP_ICFG Register Summary....1397
5.4.2.2 DSP_ICFG Register Description....1401
5.4.3 DSP_SYSTEM Registers....1401
5.4.3.1 DSP_SYSTEM Register Summary....1401
5.4.3.2 DSP_SYSTEM Register Description....1402
5.4.4 DSP_FW_L2_NOC_CFG Registers....1415
5.4.4.1 DSP_FW_L2_NOC_CFG Register Summary....1415
5.4.4.2 DSP_FW_L2_NOC_CFG Register Description....1417
6 IVA Subsystem....1431
7 Dual Cortex-M4 IPU Subsystem....1432
7.1 Dual Cortex-M4 IPU Subsystem Overview....1433
7.1.1 Introduction....1433
7.1.2 Features....1435
7.2 Dual Cortex-M4 IPU Subsystem Integration....1436
7.2.1 Dual Cortex-M4 IPU Subsystem Clock and Reset Distribution....1438
7.2.1.1 Clock Distribution....1438

7.2.1.2 Reset Distribution....1439
7.3 Dual Cortex-M4 IPU Subsystem Functional Description....1441
7.3.1 IPUx Subsystem Block Diagram....1441
7.3.2 Power Management....1442
7.3.2.1 Local Power Management....1442
7.3.2.2 Power Domains....1443
7.3.2.4 Voltage Domain....1443
7.3.2.5 Power States and Modes....1444
7.3.2.6 Wake-Up Generator (IPUx_WUGEN)....1445
7.3.2.6.1 IPUx_WUGEN Main Features....1445
7.3.3 IPUx_UNICACHE....1446
7.3.4 IPUx_UNICACHE_MMU....1447
7.3.5 IPUx_UNICACHE_SCTM....1448
7.3.5.1 Counter Functions....1448
7.3.5.1.1 Input Events....1448
7.3.5.1.2 Counters....1449
7.3.5.2 Timer Functions....1450
7.3.5.2.1 Periodic Intervals....1450
7.3.5.2.2 Event Generation....1450
7.3.6 IPUx_MMU....1450
7.3.6.1 IPUx_MMU Behavior on Page-Fault in IPUx Subsystem....1451
7.3.7 Interprocessor Communication (IPC)....1451
7.3.7.1 Use of WFE and SEV....1451
7.3.7.2 Use of Interrupt for IPC....1452
7.3.7.3 Use of the Bit-Band Feature for Semaphore Operations....1452
7.3.7.4 Private Memory Space....1453
7.3.8 IPU Boot Options....1453
7.4 Dual Cortex-M4 IPU Subsystem Register Manual....1454
7.4.1 IPUx Subsystem Instance Summary....1454
7.4.2 IPUx_UNICACHE_CFG Registers....1454
7.4.2.1 IPUx_UNICACHE_CFG Register Summary....1454
7.4.2.2 IPUx_UNICACHE_CFG Register Description....1455
7.4.3 IPUx_UNICACHE_SCTM Registers....1460
7.4.3.1 IPUx_UNICACHE_SCTM Register Summary....1460
7.4.3.2 IPUx_UNICACHE_SCTM Register Description....1461
7.4.4 IPUx_UNICACHE_MMU (AMMU) Registers....1466
7.4.4.1 IPUx_UNICACHE_MMU (AMMU) Register Summary....1466
7.4.4.2 IPUx_UNICACHE_MMU (AMMU) Register Description....1467
7.4.5 IPUx_MMU Registers....1473
7.4.6 IPUx_Cx_INTC Registers....1473
7.4.7 IPUx_WUGEN Registers....1474
7.4.7.1 IPUx_WUGEN Register Summary....1474
7.4.7.2 IPUx_WUGEN Register Description....1474
7.4.8 IPUx_Cx_RW_TABLE Registers....1477
7.4.8.1 IPUx_Cx_RW_TABLE Register Summary....1477
7.4.8.2 IPUx_Cx_RW_TABLE Register Description....1478
8 Embedded Vision Engine....1480
8.1 Embedded Vision Engine (EVE) Subsystem....1481
8.1.1 EVE Overview....1481
8.1.1.1 EVE Memories....1483
8.1.2 EVE Integration....1483
8.1.2.1 Multi-EVE Recommended Connections....1487
8.1.3 EVE Functional Description....1488
8.1.3.1 EVE Connection ID (ConnID) Mapping....1488
8.1.3.2 EVE Processors Overview....1489
8.1.3.2.1 Scalar Core (ARP32)....1489
8.1.3.2.2 VCOP....1489
8.1.3.2.3 Scalar-Vector Interaction....1489
8.1.3.3 Internal Memory Overview....1489
8.1.3.3.1 Program Cache/Memory....1490
8.1.3.3.2 ARP32 Data Memory (DMEM)....1490

8.1.3.3.3 WBUF....1490
8.1.3.3.4 Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB....1491
8.1.3.3.5 Memory Switch Error Registers....1492
8.1.3.3.6 Memory Error Detection....1492
8.1.3.3.7 VCOP System Error Halt Conditions....1495
8.1.3.4 Program Cache Architecture....1495
8.1.3.4.1 Basic Operation....1496
8.1.3.4.2 Line Buffer....1497
8.1.3.4.3 Software Direct Preload....1497
8.1.3.4.4 User Coherence Operation....1498
8.1.3.4.5 Demand-Based Prefetch....1499
8.1.3.4.6 Debug Support....1499
8.1.3.4.7 Error Detection....1501
8.1.3.5 EDMA....1501
8.1.3.5.1 DMA Channel Events....1502
8.1.3.5.2 DMA Parameter Set....1502
8.1.3.5.3 Channel Controller....1503
8.1.3.5.4 EVE-Level Bus Width and Throughput....1505
8.1.3.6 General-Purpose Inputs/Outputs....1506
8.1.3.7 CME Signaling....1506
8.1.3.8 Multi-EVE and VIP Usage Models....1508
8.1.3.8.1 Data Partitioning....1508
8.1.3.8.2 Task Partitioning....1508
8.1.3.9 Memory Management Unit....1509
8.1.3.10 Interrupt Control....1509
8.1.3.10.1 EVE Interrupt Sources – Memory Switch and Parity Error Interrupts....1510
8.1.3.10.2 ARP32 INTC....1512
8.1.3.10.3 Output Interrupt Reduction....1515
8.1.3.10.4 End of Interrupt Mapping....1515
8.1.3.11 Interprocessor Communication....1516
8.1.3.11.1 Mailbox Configuration....1516
8.1.3.12 Powerdown....1517
8.1.3.12.1 Extended Duration Sleep....1517
8.1.3.13 Hardware-Assisted Software Self-Test – MISRs....1519
8.1.3.13.1 Mapping of MISRs to Different Width Buses....1520
8.1.3.13.2 Detection of Valid Address and Data Cycles....1520
8.1.3.13.3 Creating a Unique Signature – Software Self-Test Implications....1520
8.1.3.13.4 Multipass Tests Using WBUF MISR....1520
8.1.3.14 Error Recovery – ARP32 and OCP Disconnect....1520
8.1.3.14.1 ARP32 Disconnect....1521
8.1.3.14.2 OCP Initiator Disconnect....1521
8.1.3.15 Lock and Unlock Feature....1521
8.1.3.16 EVE Memory Map....1522
8.1.3.16.1 VCOP and Local EDMA: IBUF Memory Map Aliasing....1523
8.1.3.16.2 ARP32 Write Model – Avoiding Race Conditions....1524
8.1.3.17 Debug Support....1526
8.1.3.17.1 ARP32 Debug Support....1526
8.1.3.17.2 SCTM....1526
8.1.3.17.3 SMSET....1528
8.1.3.18 EVE L2_FNOC Interconnect....1530
8.1.3.18.1 EVE L2_FNOC Flag Mux and Error Log Registers....1530
8.1.4 EVE Programming Model....1530
8.1.4.1 Boot....1530
8.1.4.2 Task Change and Program Cache Prefetch....1531
8.1.4.2.1 Simple or Unoptimized Branch to New Task....1531
8.1.4.2.2 Prefetch, Wait, then Branch to New Task....1532
8.1.4.2.3 Hidden Prefetch....1532
8.1.4.3 Interrupts....1532
8.1.4.4 Safety Considerations....1533
8.1.4.4.1 Memory Error Detection....1533
8.1.4.4.2 MMU....1533

8.1.4.4.3 Firewall....1533
8.1.4.4.4 Interconnect....1533
8.1.4.4.5 Application Stability/Sequencing....1533
8.1.4.4.6 Interrupt Servicing....1533
8.1.5 EVE Subsystem Register Manual....1533
8.1.5.1 EVE Instance Summary....1533
8.1.5.2 EVE Register Summary and Description....1534
8.1.5.2.1 EVE Register Summary....1534
8.1.5.2.2 EVE Register Description....1537
8.1.5.3 EVE L2_FNOC Register Summary and Description....1590
8.1.5.3.1 EVE L2_FNOC Register Summary....1590
8.1.5.3.2 EVE L2_FNOC Register Description....1590
8.1.6 Subsystem Counter Timer Module....1597
8.1.6.1 Introduction....1597
8.1.6.1.1 Overview....1597
8.1.6.1.2 Top-Level Requirements....1597
8.1.6.1.3 Configuration....1598
8.1.6.1.4 Block Diagram....1598
8.1.6.2 Functional Description....1599
8.1.6.2.1 Configuration Interface....1599
8.1.6.2.2 Counter Function....1599
8.1.6.2.3 Timer Function....1601
8.1.6.2.4 System Trace Integration....1604
8.1.6.3 Use Case Examples....1606
8.1.6.3.1 Counter Enable....1606
8.1.6.3.2 Timer Enable....1607
8.1.6.3.3 Periodic STM Export Enable....1607
8.1.6.3.4 Disabling the SCTM....1608
8.1.6.4 SCTM Register Manual....1609
8.1.6.4.1 SCTM Instance Summary....1609
8.1.6.4.2 SCTM Registers....1609
8.1.7 Software Message and System Event Trace....1616
8.1.7.1 Introduction....1616
8.1.7.1.1 Overview....1616
8.1.7.1.2 Configuration....1616
8.1.7.1.3 Block Diagram....1616
8.1.7.2 Functional Description....1617
8.1.7.2.1 Connectivity....1617
8.1.7.2.2 SMSET Event Mapping....1617
8.1.7.2.3 Software Messages....1618
8.1.7.2.4 SMSET Master Port....1618
8.1.7.2.5 SMSET Debug Features....1619
8.1.7.2.6 Component Ownership....1619
8.1.7.3 Use Case Examples....1620
8.1.7.3.1 Procedure to Enable System Event Capture....1620
8.1.7.3.2 Procedure to Start and Stop System Event Capture from External Trigger Detection....1620
8.1.7.3.3 Procedure to Disable System Event Capture....1621
8.1.7.4 SMSET Register Manual....1621
8.1.7.4.1 SMSET Instance Summary....1621
8.1.7.4.2 SMSET Register Summary....1621
8.1.7.4.3 SMSET Register Description....1621
8.2 ARP32 CPU and Instruction Set....1625
8.2.1 Overview....1625
8.2.2 Features....1627
8.2.3 Block Diagram....1627
8.2.4 Architecture....1627
8.2.4.1 Interface Description....1627
8.2.4.1.1 Data Memory Interface....1629
8.2.4.1.2 Instruction Memory Interface....1629
8.2.4.2 Pipeline....1630
8.2.4.2.1 Overview....1630

8.2.4.2.2 Pipeline Operation....1630
8.2.4.2.3 Pipeline Interlocks....1631
8.2.4.3 Data Format....1632
8.2.4.4 Endian Support....1632
8.2.4.5 Architectural Register File....1632
8.2.4.6 CPU Control Registers....1632
8.2.4.6.1 Control Status Register (CSR)....1635
8.2.4.6.2 Interrupt Enable Register (IER)....1636
8.2.4.6.3 Interrupt Flag Register (IFR)....1637
8.2.4.6.4 Interrupt Set Register (ISR)....1638
8.2.4.6.5 Interrupt Clear Register (ICR)....1639
8.2.4.6.6 Nonmaskable Interrupt (NMI) Return Pointer Register (NRP)....1640
8.2.4.6.7 Interrupt Return Pointer Register (IRP)....1640
8.2.4.6.8 Stack Pointer Register (SP)....1641
8.2.4.6.9 Global Data Pointer Register (GDP)....1641
8.2.4.6.10 Link Register (LR)....1641
8.2.4.6.11 Loop 0 Start Address Register (LSA0)....1642
8.2.4.6.12 Loop 0 End Address Register (LEA0)....1642
8.2.4.6.13 Loop 0 Iteration Count Register (LCNT0)....1643
8.2.4.6.14 Loop 1 Start Address Register (LSA1)....1644
8.2.4.6.15 Loop 1 End Address Register (LEA1)....1644
8.2.4.6.16 Loop 1 Iteration Count Register (LCNT1)....1645
8.2.4.6.17 Loop 0 Iteration Count Reload Value Register (LCNT0RLD)....1645
8.2.4.6.18 Shadow Control Status Register (SCSR)....1646
8.2.4.6.19 NMI Shadow Control Status Register (NMISCSR)....1647
8.2.4.6.20 CPU Identification Register (CPUID)....1648
8.2.4.6.21 Decode Program Counter Register (DPC)....1649
8.2.4.6.22 Time Stamp Counter Registers (TSCL and TSCH)....1649
8.2.4.7 CPU Shadow Registers....1651
8.2.4.8 Functional Units....1652
8.2.4.9 Instruction Fetch....1652
8.2.4.10 Alignment of 32-bit Instructions....1653
8.2.4.11 Instruction Execution in Branch Delay Slot....1653
8.2.4.12 Address Space....1653
8.2.4.13 Program Counter Convention....1654
8.2.4.14 Stack Pointer Convention....1654
8.2.4.15 Global Data Pointer Convention....1655
8.2.4.16 Conditional Execution....1655
8.2.4.17 Hardware Loop Acceleration....1655
8.2.4.17.1 Overview....1655
8.2.4.17.2 Loop Registers....1656
8.2.4.17.3 Loop Setup Instructions....1656
8.2.4.17.4 Loop Operation....1656
8.2.4.17.5 Call and Branch within Loop Context....1658
8.2.4.17.6 Dynamic Changes to Loop Iteration Count....1658
8.2.4.17.7 Interrupt Processing During HLA....1658
8.2.4.17.8 HLA Usage in Interrupt Context....1658
8.2.4.17.9 HLA Usage Restrictions....1658
8.2.4.17.10 HLA Mapping Examples....1659
8.2.4.18 Interrupts....1663
8.2.4.18.1 Overview....1663
8.2.4.18.2 Interrupt Processing....1664
8.2.4.18.3 Interrupt Acknowledgment....1665
8.2.4.18.4 Interrupt Priorities....1666
8.2.4.18.5 Interrupt Service Table (IST)....1667
8.2.4.18.6 Interrupt Flags....1667
8.2.4.18.7 Interrupt Behavior....1668
8.2.4.18.8 Interrupt Context Save and Restore....1671
8.2.4.18.9 Nested Interrupts....1673
8.2.4.18.10 Non-nested Interrupt Latency....1675
8.2.A Instruction Set....1675

8.2.A.1 Instruction Operation and Execution Notations....1675
8.2.A.2 Instruction Syntax and Opcode Notations....1677
8.2.A.3 Instruction Scheduling Restrictions....1677
8.2.A.3.1 Restrictions Applicable to a Branch Delay Slot....1677
8.2.A.3.2 Restrictions on Loops Using Hardware Loop Assist (HLA)....1677
8.2.A.3.3 Restrictions on Other Types of Control Flow Instructions....1678
8.2.A.3.4 Restrictions for Write Data Bypass to Control Register Reads....1678
8.2.A.3.5 Restrictions for Write Data Bypass to Shadow Register Reads....1678
8.2.A.3.6 Restrictions for Link Register Update....1678
8.2.A.4 Instruction Set Encoding....1678
8.2.A.5 Instruction Descriptions....1678
8.2.B Clock, Reset, and Dynamic Power Management....1815
8.2.B.1 Introduction....1815
8.2.B.2 CPU Reset Modes....1815
8.2.B.3 Dynamic Power Management....1817
8.2.C Notes on Programming Model....1818
8.2.C.1 Booting....1818
8.2.C.2 Enabling and Disabling Interrupts....1818
8.2.C.2.1 Globally Enabling or Disabling Maskable Interrupts....1818
8.2.C.2.2 Enabling or Disabling Individual Interrupts....1819
8.2.C.3 Stack Usage in Interrupt Service Routine....1819
8.2.C.4 General Restrictions....1820
8.3 VCOP CPU and Instruction Set....1820
8.3.1 Module Overview....1820
8.3.2 Features....1820
8.3.3 Block Diagram....1821
8.3.4 System Interfaces....1822
8.3.4.1 Interrupts....1822
8.3.4.2 Configuration Bus Slave Port....1822
8.3.4.3 Performance Counter Interface....1822
8.3.4.4 Data Memory Map....1823
8.3.5 Functional Description....1825
8.3.5.1 Scalar-Vector Architecture....1825
8.3.5.1.1 Scalar Core....1825
8.3.5.1.2 Scalar-Vector Interaction....1825
8.3.5.2 Vector Core Overview....1826
8.3.5.2.1 Nested for Loop Model....1826
8.3.5.2.2 Instruction Organization....1828
8.3.5.3 Vector Control....1828
8.3.5.3.1 Repeat End Count....1828
8.3.5.3.2 Parameter Pointer....1829
8.3.5.3.3 Switch Buffers....1829
8.3.5.4 Vector-Scalar Synchronization....1830
8.3.5.4.1 Wait for Vector Core Done....1830
8.3.5.4.2 Wait for Vector Core Ready....1830
8.3.5.5 Vector Computation....1830
8.3.5.5.1 Vector Loop....1830
8.3.5.5.2 Vector Register Initialization....1831
8.3.5.5.3 Address Generator (agen)....1831
8.3.5.5.4 Vector Load....1833
8.3.5.5.5 Vector Arithmetic/Logic Operations....1838
8.3.5.5.6 Vector Store....1841
8.3.5.5.7 Table Lookup Operation....1843
8.3.5.5.8 Histogram Operation....1848
8.3.5.5.9 Circular Buffer Addressing Support....1852
8.3.5.5.10 Load/Store Address Alignment Constraints....1853
8.3.5.6 Load/Store Buffer and Scheduling....1854
Example 8-9. 3-Tap Horizontal Filtering, Byte Type....1854
Example 8-10. Horizontal Filtering, Short Type....1855
8.3.5.7 VCOP Per-Loop Overhead....1856
8.3.5.8 VCOP Error Handling....1857

8.3.5.9 Vector Operation Details....... 1858
8.3.5.9.1 VABS....... 1858
8.3.5.9.2 VABSDIF....... 1858
8.3.5.9.3 VADD....... 1858
8.3.5.9.4 VADDH....... 1858
8.3.5.9.5 VADDSUB....... 1859
8.3.5.9.6 VADD3....... 1859
8.3.5.9.7 VADIF3....... 1859
8.3.5.9.8 VAND....... 1859
8.3.5.9.9 VANDN....... 1859
8.3.5.9.10 VAND3....... 1860
8.3.5.9.11 VBINLOG....... 1860
8.3.5.9.12 VBITC....... 1861
8.3.5.9.13 VBITDI....... 1861
8.3.5.9.14 VBITI....... 1861
8.3.5.9.15 VBITPK....... 1862
8.3.5.9.16 VBITR....... 1862
8.3.5.9.17 VBITTR....... 1862
8.3.5.9.18 VBITUNPK....... 1863
8.3.5.9.19 VCMOV....... 1863
8.3.5.9.20 VCMPEQ....... 1863
8.3.5.9.21 VCMPGE....... 1863
8.3.5.9.22 VCMPGT....... 1864
8.3.5.9.23 VDINTRLV....... 1864
8.3.5.9.24 VDINTRLV2....... 1865
8.3.5.9.25 VEXITNZ....... 1865
8.3.5.9.26 VINTRLV....... 1866
8.3.5.9.27 VINTRLV2....... 1866
8.3.5.9.28 VINTRLV4....... 1866
8.3.5.9.29 VLMBD....... 1867
8.3.5.9.30 VMADD....... 1868
8.3.5.9.31 VMAX....... 1868
8.3.5.9.32 VMAXSETF....... 1869
8.3.5.9.33 VMIN....... 1869
8.3.5.9.34 VMINSETF....... 1869
8.3.5.9.35 VMPY....... 1869
8.3.5.9.36 VMSUB....... 1870
8.3.5.9.37 VNOP....... 1870
8.3.5.9.38 VNOT....... 1870
8.3.5.9.39 VOR....... 1870
8.3.5.9.40 VOR3....... 1871
8.3.5.9.41 VRND....... 1871
8.3.5.9.42 VSAD....... 1871
8.3.5.9.43 VSEL....... 1871
8.3.5.9.44 VSHF....... 1872
8.3.5.9.45 VSHFOR....... 1872
8.3.5.9.46 VSHF16....... 1872
8.3.5.9.47 VSIGN....... 1872
8.3.5.9.48 VSORT2....... 1873
8.3.5.9.49 VSUB....... 1873
8.3.5.9.50 VSWAP....... 1873
8.3.5.9.51 VXOR....... 1873
8.3.6 Debug Support....... 1873
8.3.7 VCOP Register Manual....... 1874
8.3.7.1 VCOP Instance Summary....... 1874
8.3.7.2 VCOP Registers....... 1874
8.3.7.2.1 VCOP Registers Mapping Summary....... 1874
8.3.7.2.2 VCOP Register Description....... 1875
9 Video Input Port....... 1882
9.1 VIP Overview....... 1883
9.2 VIP Environment....... 1885

9.3 VIP Integration.... 1891
9.4 VIP Functional Description.... 1894
9.4.1 VIP Block Diagram.... 1894
9.4.2 VIP Software Reset.... 1894
9.4.3 VIP Power and Clocks Management.... 1895
9.4.3.1 VIP Clocks.... 1895
9.4.3.2 VIP Idle Mode.... 1895
9.4.3.3 VIP StandBy Mode.... 1895
9.4.4 VIP Slice.... 1895
9.4.4.1 VIP Slice Processing Path Overview.... 1895
9.4.4.2 VIP Slice Processing Path Multiplexers.... 1897
9.4.4.2.1 VIP_CSC Multiplexers.... 1897
9.4.4.2.2 VIP_SC Multiplexer.... 1897
9.4.4.2.3 Output to VPDMA Multiplexers.... 1897
9.4.4.3 VIP Slice Processing Path Examples.... 1897
9.4.4.3.1 Input: A=RGB, B=YUV422; Output: A=RGB, B=RGB.... 1898
9.4.4.3.2 Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=RGB.... 1899
9.4.4.3.3 Input: A=RGB, B=YUV422; Output: A=RGB, B=Scaled YUV420.... 1900
9.4.4.3.4 Input: A=YUV444, B=YUV422; Output: A=YUV422, A=Scaled YUV422, B=YUV422.... 1901
9.4.4.3.5 Input: A=YUV444; Output: A=Scaled YUV420, A=YUV420.... 1902
9.4.4.3.6 Input: A=YUV444; Output: A=Scaled YUV420, A=YUV444.... 1903
9.4.4.3.7 Input: A=YUV422 8/16; Output: A=Scaled YUV420, A=YUV444.... 1904
9.4.4.3.8 Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=YUV420.... 1905
9.4.4.3.9 Input: A=YUV422 8/16, B=YUV422; Output: A=YUV420, B=YUV420.... 1906
9.4.5 VIP Parser.... 1906
9.4.5.1 Features.... 1907
9.4.5.2 Repacker.... 1907
9.4.5.3 Analog Video.... 1909
9.4.5.4 Digitized Video.... 1909
9.4.5.5 Frame Buffers.... 1911
9.4.5.6 Input Data Interface.... 1912
9.4.5.6.1 8b Interface Mode.... 1912
9.4.5.6.2 16b Interface Mode.... 1913
9.4.5.6.3 24b Interface Mode.... 1913
9.4.5.6.4 Signal Relationships.... 1914
9.4.5.6.5 General 5 Pin Interfaces.... 1914
9.4.5.6.6 Signal Subsets—4 Pin VSYNC, ACTVID, and FID.... 1916
9.4.5.6.7 Signal Subsets—4 Pin VSYNC, HSYNC, and FID.... 1916
9.4.5.6.8 Vertical Sync.... 1917
9.4.5.6.9 Field ID Determination Using Dedicated Signal.... 1918
9.4.5.6.10 Field ID Determination Using VSYNC Skew.... 1919
9.4.5.6.11 Rationale for FID Determination By VSYNC Skew.... 1920
9.4.5.6.12 ACTVID Framing.... 1921
9.4.5.6.13 Ancillary Data Storage in Descrete Sync Mode.... 1921
9.4.5.7 BT.656 Style Embedded Sync.... 1923
9.4.5.7.1 Data Input.... 1923
9.4.5.7.2 Sync Words.... 1923
9.4.5.7.3 Error Correction.... 1925
9.4.5.7.4 Embedded Sync Ancillary Data.... 1925
9.4.5.7.5 Embedded Sync RGB 24-bit Data.... 1926
9.4.5.8 Source Multiplexing.... 1927
9.4.5.8.1 Multiplexing Scenarios.... 1927
9.4.5.8.2 2-Way Multiplexing.... 1928
9.4.5.8.3 4-Way Multiplexing.... 1928
9.4.5.8.4 Line Multiplexing.... 1928
9.4.5.8.5 Super Frame Concept in Line Multiplexing.... 1929
9.4.5.8.6 8-bit Data Interface in Line Multiplexing.... 1929
9.4.5.8.7 16-bit Data Interface in Line Multiplexing.... 1929
9.4.5.8.8 Split Lines in Line Multiplex Mode.... 1930
9.4.5.8.9 Meta Data.... 1930
9.4.5.8.10 TI Line Mux Mode, Split Lines, and Channel ID Remapping.... 1931

9.4.5.9 Channel ID Extraction for 2x/4x Multiplexed Source....1932
9.4.5.9.1 Channel ID Extraction Overview....1932
9.4.5.9.2 Channel ID Embedded in Protection Bits for 2- and 4-Way Multiplexing....1932
9.4.5.9.3 Channel ID Embedded in Horizontal Blanking Pixel Data for 2- and 4-Way Multiplexing....1932
9.4.5.10 Embedded Sync Mux Modes and Data Bus Widths....1933
9.4.5.11 Ancillary and Active Video Cropping....1933
9.4.5.12 Interrupts....1936
9.4.5.13 VDET Interrupt....1937
9.4.5.14 Source Video Size....1937
9.4.5.15 Clipping....1938
9.4.5.16 Current and Last FID Value....1938
9.4.5.17 Disable Handling....1938
9.4.5.18 Picture Size Interrupt....1939
9.4.5.19 Discrete Sync Signals....1939
9.4.5.19.1 VBLNK and HBLNK....1940
9.4.5.19.2 BLNK and ACTVID (1)....1940
9.4.5.19.3 VBLNK and ACTVID(2)....1941
9.4.5.19.4 VBLNK and HSYNC....1941
9.4.5.19.5 VSYNC and HBLNK....1941
9.4.5.19.6 VSYNC and ACTIVID(1)....1942
9.4.5.19.7 VSYNC and ACTIVID(2)....1942
9.4.5.19.8 VSYNC and HSYNC....1942
9.4.5.19.9 Line and Pixel Capture Examples....1943
9.4.5.20 VIP Overflow Detection and Recovery....1944
9.4.6 VIP Color Space Converter (CSC)....1945
9.4.6.1 CSC Features....1945
9.4.6.2 CSC Functional Description....1945
9.4.6.2.1 HDTV Application....1945
9.4.6.2.2 SDTV Application....1948
9.4.6.3 CSC Bypass Mode....1950
9.4.7 VIP Scaler (SC)....1950
9.4.7.1 SC Features....1950
9.4.7.2 SC Functional Description....1951
9.4.7.2.1 Trimmer....1951
9.4.7.2.3 Peaking....1952
9.4.7.2.4 Vertical Scaler....1954
9.4.7.2.5 Horizontal Scaler....1955
9.4.7.2.6 Basic Configurations....1960
9.4.7.2.7 Coefficient Memory....1960
9.4.7.3 SC Code....1963
9.4.7.3.1 Generate Coefficient Memory Image....1963
9.4.7.3.2 Scaler Configuration Calculation....1965
9.4.7.3.3 Typical Configuration Values....1968
9.4.7.4 SC Coefficient Data Files....1969
9.4.7.4.1 HS Polyphase Filter Coefficients....1969
9.4.7.4.2 VS Polyphase Filter Coefficients....1973
9.4.7.4.3 VS (Bilinear Filter Coefficients)....1980
9.4.8 VIP Video Port Direct Memory Access (VPDMA)....1981
9.4.8.1 VPDMA Introduction....1981
9.4.8.2 VPDMA Basic Definitions....1981
9.4.8.2.1 Client....1981
9.4.8.2.2 Channel....1981
9.4.8.2.3 List....1982
9.4.8.2.4 Data Formats Supported....1982
9.4.8.4 VPDMA Client Buffering and Functionality....1983
9.4.8.5 VPDMA Channels Assignment....1985
9.4.8.6 VPDMA MFLAG Mechanism....1989
9.4.8.7 VPDMA Interrupts....1989
9.4.8.8 VPDMA Descriptors....2005
9.4.8.8.1 Data Transfer Descriptors....2005
9.4.8.8.2 Configuration Descriptor....2014

9.4.8.8.3 Control Descriptor........ 2016
9.4.8.9 VPDMA Configuration........ 2020
9.4.8.9.1 Regular List........ 2020
9.4.8.9.2 Video Input Ports........ 2021
9.4.8.10 VPDMA Data Formats........ 2021
9.4.8.10.1 YUV Data Formats........ 2021
9.4.8.10.2 RGB Data Formats........ 2026
9.4.8.10.3 Miscellaneous Data Type........ 2031
9.5 VIP Register Manual........ 2031
9.5.1 VIP Instance Summary........ 2031
9.5.2 VIP Top Level Registers........ 2032
9.5.2.1 VIP Top Level Register Summary........ 2032
9.5.2.2 VIP Top Level Register Description........ 2032
9.5.3 VIP Parser Registers........ 2066
9.5.3.1 VIP Parser Register Summary........ 2066
9.5.3.2 VIP Parser Register Description........ 2068
9.5.4 VIP CSC Registers........ 2104
9.5.4.1 VIP CSC Register Summary........ 2104
9.5.4.2 VIP CSC Register Description........ 2104
9.5.5 VIP SC registers........ 2108
9.5.5.1 VIP SC Register Summary........ 2108
9.5.5.2 VIP SC Register Description........ 2110
9.5.6 VIP VPDMA Registers........ 2124
9.5.6.1 VIP VPDMA Register Summary........ 2124
9.5.6.2 VIP VPDMA Register Description........ 2128
10 Video Processing Engine........ 2304
10.1 VPE Overview........ 2305
10.2 VPE Integration........ 2306
10.3 VPE Functional Description........ 2308
10.3.1 VPE Block Diagram........ 2308
10.3.2 VPE VC1 Range Mapping/Range Reduction........ 2308
10.3.3 VPE Deinterlacer (DEI)........ 2309
10.3.3.1 Functional Description........ 2309
10.3.3.2 Bypass Mode........ 2310
10.3.4 VPE Scaler (SC)........ 2318
10.3.4.1 SC Features........ 2318
10.3.4.2 SC Functional Description........ 2318
10.3.4.2.1 Trimmer........ 2319
10.3.4.2.3 Peaking........ 2319
10.3.4.2.4 Vertical Scaler........ 2321
10.3.4.2.5 Horizontal Scaler........ 2322
10.3.4.2.6 Basic Configurations........ 2327
10.3.4.2.7 Coefficient Memory........ 2327
10.3.4.3 SC Code........ 2330
10.3.4.3.1 Generate Coefficient Memory Image........ 2330
10.3.4.3.2 Scaler Configuration Calculation........ 2332
10.3.4.3.3 Typical Configuration Values........ 2335
10.3.4.4 SC Coefficient Data Files........ 2336
10.3.4.4.1 HS Polyphase Filter Coefficients........ 2336
10.3.4.4.2 VS Polyphase Filter Coefficients........ 2340
10.3.4.4.3 VS (Bilinear Filter Coefficients)........ 2348
10.3.5 VPE Color Space Converter (CSC)........ 2348
10.3.5.1 CSC Features........ 2348
10.3.5.2 CSC Functional Description........ 2349
10.3.5.4 CSC Bypass Mode........ 2354
10.3.6 VPE Chroma Up-Sampler (CHR_US)........ 2354
10.3.6.1 Features........ 2354
10.3.6.2 Functional Description........ 2354
10.3.6.3 For Interlaced YUV420 Input Data........ 2356
10.3.6.4 Edge Effects........ 2357
10.3.6.5 Modes of Operation (VPDMA)........ 2357

10.3.6.6 Coefficient Configuration....2358
10.3.7 VPE Chroma Down-Sampler (CHR_DS)....2359
10.3.8 VPE YUV422 to YUV444 Conversion....2359
10.3.9 VPE Video Port Direct Memory Access (VPDMA)....2359
10.3.9.1 VPDMA Introduction....2359
10.3.9.2 VPDMA Basic Definitions....2360
10.3.9.2.1 Client....2360
10.3.9.2.2 Channel....2360
10.3.9.2.3 List....2360
10.3.9.2.4 Data Formats Supported....2360
10.3.9.3 VPDMA Client Buffering and Functionality....2361
10.3.9.4 VPDMA Channels Assignment....2362
10.3.9.5 VPDMA Interrupts....2362
10.3.9.6 VPDMA Descriptors....2367
10.3.9.6.1 Data Transfer Descriptors....2367
10.3.9.6.2 Configuration Descriptor....2377
10.3.9.6.3 Control Descriptor....2379
10.3.9.7 VPDMA Configuration....2383
10.3.9.7.1 Regular List....2383
10.3.9.7.2 Video Input Ports....2383
10.3.9.8 VPDMA Data Formats....2383
10.3.9.8.1 YUV Data Formats....2384
10.3.9.8.2 RGB Data Formats....2388
10.3.9.8.3 Miscellaneous Data Type....2399
10.3.10 VPE Software Reset....2399
10.3.11 VPE Power and Clocks Management....2399
10.3.11.1 VPE Clocks....2399
10.3.11.2 VPE Idle Mode....2400
10.3.11.3 VPE StandBy Mode....2400
10.4 VPE Register Manual....2400
10.4.1 VPE Instance Summary....2400
10.4.2 VPE_CSC Registers....2400
10.4.2.1 VPE_CSC Register Summary....2400
10.4.2.2 VPE_CSC Register Description....2400
10.4.3 VPE_SC Registers....2404
10.4.3.1 VPE_SC Register Summary....2404
10.4.3.2 VPE_SC Register Description....2405
10.4.4 VPE_CHR_US Registers....2417
10.4.4.1 VPE_CHR_US Register Summary....2417
10.4.4.2 VPE_CHR_US Register Description....2417
10.4.5 VPE_DEI Registers....2421
10.4.5.1 VPE_DEI Register Summary....2421
10.4.5.2 VPE_DEI Register Description....2422
10.4.6 VPE_VPDMA Registers....2431
10.4.6.1 VPE_VPDMA Register Summary....2432
10.4.6.2 VPE_VPDMA Register Description....2435
10.4.7 VPE_TOP_LEVEL Registers....2580
10.4.7.1 VPE_TOP_LEVEL Register Summary....2580
10.4.7.2 VPE_TOP_LEVEL Register Description....2580
11 Display Subsystem....2598
11.1 Display Subsystem Overview....2599
11.1.1 Display Subsystem Environment....2600
11.1.1.1 Display Subsystem LCD Support....2601
11.1.1.1.1 Display Subsystem LCD with Parallel Interfaces....2601
11.1.1.2 Display Subsystem TV Display Support....2603
11.1.1.2.1 Display Subsystem TV With Parallel Interfaces....2603
11.1.1.2.2 Display Subsystem TV With Serial Interfaces....2603
11.1.2 Display Subsystem Integration....2603
11.1.2.1 Display Subsystem Clocks....2605
11.1.2.2 Display Subsystem Resets....2607
11.1.2.3 Display Subsystem Power Management....2607

11.1.2.3.1 Display Subsystem Standby Mode....2608
11.1.2.3.3 Display Subsystem Wake-Up Mode.... 2609
11.1.3 Display Subsystem DPLL Controllers Functional Description.... 2610
11.1.3.1 DPLL Controllers Overview....2610
11.1.3.2 OCP2SCP2 Functional Description....2611
11.1.3.2.1 OCP2SCP2 Reset.... 2612
11.1.3.2.2 OCP2SCP2 Power Management.... 2612
11.1.3.2.3 OCP2SCP2 Timing Registers....2612
11.1.3.3 DPLL_VIDEO Functional Description....2612
11.1.3.3.1 DPLL_VIDEO Controller Architecture....2612
11.1.3.3.2 DPLL_VIDEO Operations....2613
11.1.3.3.3 DPLL_VIDEO Error Handling.... 2614
11.1.3.3.4 DPLL_VIDEO Software Reset....2614
11.1.3.3.5 DPLL_VIDEO Power Management.... 2614
11.1.3.3.6 DPLL_VIDEO HSDIVIDER Loading Operation.... 2615
11.1.3.3.7 DPLL_VIDEO Clock Sequence.... 2615
11.1.3.3.8 DPLL_VIDEO Go Sequence....2617
11.1.3.3.9 DPLL_VIDEO Recommended Values.... 2618
11.1.3.4 DPLL_HDMI Functional Description....2619
11.1.3.4.1 DPLL_HDMI and PLLCTRL_HDMI Overview....2619
11.1.3.4.2 DPLL_HDMI and PLLCTRL_HDMI Architecture.... 2619
11.1.3.4.3 DPLL_HDMI Operations....2621
11.1.3.4.4 DPLL_HDMI Register Access....2621
11.1.3.4.5 DPLL_HDMI Error Handling.... 2622
11.1.3.4.6 DPLL_HDMI Software Reset....2622
11.1.3.4.7 DPLL_HDMI Power Management.... 2622
11.1.3.4.8 DPLL_HDMI Lock Sequence....2623
11.1.3.4.9 DPLL_HDMI Go Sequence....2625
11.1.3.4.10 DPLL_HDMI Recommended Values.... 2625
11.1.4 Display Subsystem Programming Guide....2627
11.1.5 Display Subsystem Register Manual.... 2628
11.1.5.1 Display Subsystem Instance Summary....2628
11.1.5.2 Display Subsystem Registers....2628
11.1.5.2.1 Display Subsystem Registers Mapping Summary....2628
11.1.5.2.2 Display Subsystem Register Description....2628
11.1.5.3 OCP2SCP2 registers.... 2631
11.1.5.3.1 OCP2SCP2 Register Summary....2631
11.1.5.3.2 OCP2SCP Register Description....2632
11.1.5.4 DPLL_VIDEO Registers....2633
11.1.5.4.1 DPLL_VIDEO Register Summary....2633
11.1.5.4.2 DPLL_VIDEO Register Description....2634
11.1.5.5 DPLL_HDMI Registers....2641
11.1.5.5.1 DPLL_HDMI Registers Mapping Summary....2641
11.1.5.5.2 DPLL_HDMI Register Description.... 2641
11.1.5.6 HDMI_WP Registers.... 2647
11.1.5.6.1 HDMI_WP Registers Mapping Summary.... 2647
11.1.5.6.2 HDMI_WP Register Description.... 2648
11.1.5.7 DSI Registers.... 2649
11.1.5.7.1 DSI Register Summary....2649
11.1.5.7.2 DSI Register Description.... 2649
11.2 Display Controller....2650
11.2.1 DISPC Overview....2650
11.2.2 DISPC Environment....2655
11.2.2.1 DISPC LCD Output and Data Format for the Parallel Interface....2656
11.2.2.2 DISPC Transaction Timing Diagrams....2660
11.2.2.3 DISPC TV Output and Data Format for the Parallel Interface....2664
11.2.3 DISPC Integration....2665
11.2.4 DISPC Functional Description.... 2667
11.2.4.1 DISPC Clock Configuration....2668
11.2.4.2 DISPC Software Reset....2669
11.2.4.3 DISPC Power Management....2669

11.2.4.3.1 DISPC Idle Mode....2669
11.2.4.3.2 DISPC StandBy Mode....2670
11.2.4.3.3 DISPC Wakeup....2671
11.2.4.4 DISPC Interrupt Requests....2671
11.2.4.5 DISPC DMA Requests....2673
11.2.4.6 DISPC DMA Engine....2673
11.2.4.6.1 DISPC Addressing and Bursts....2674
11.2.4.6.2 DISPC Immediate Base Address Flip Mechanism....2675
11.2.4.6.3 DISPC DMA Buffers....2676
11.2.4.6.4 DISPC MFLAG Mechanism and Arbitration....2678
11.2.4.6.5 DISPC Predecimation....2679
11.2.4.6.6 DISPC Progressive-to-Interlaced Format Conversion....2680
11.2.4.6.7 DISPC Arbitration....2681
11.2.4.6.8 DISPC DMA Power Modes....2681
11.2.4.7 DISPC Rotation and Mirroring....2682
11.2.4.8 DISPC Memory Format....2684
11.2.4.9 DISPC Graphics Pipeline....2686
11.2.4.9.1 DISPC Replication Logic....2687
11.2.4.9.2 DISPC Antiflicker Filter....2688
11.2.4.10 DISPC Video Pipelines....2689
11.2.4.10.1 DISPC Replication Logic....2690
11.2.4.10.2 DISPC VC-1 Range Mapping Unit....2691
11.2.4.10.3 DISPC CSC Unit YUV to RGB....2692
11.2.4.10.4 DISPC Scaler Unit....2694
11.2.4.11 DISPC Write-Back Pipeline....2703
11.2.4.11.1 DISPC Write-Back CSC Unit RGB to YUV....2703
11.2.4.11.2 DISPC Write-Back Scaler Unit....2704
11.2.4.11.3 DISPC Write-Back RGB Truncation Logic....2709
11.2.4.12 DISPC Hardware Cursor....2709
11.2.4.13 DISPC LCD Outputs....2710
11.2.4.13.1 DISPC Overlay Manager....2710
11.2.4.13.2 DISPC Gamma Correction Unit....2718
11.2.4.13.3 DISPC Color Phase Rotation Unit....2720
11.2.4.13.4 DISPC Color Space Conversion....2721
11.2.4.13.5 DISPC BT.656 and BT.1120 Modes....2722
11.2.4.13.6 DISPC Active Matrix....2724
11.2.4.13.7 DISPC Synchronized Buffer Update....2729
11.2.4.13.8 DISPC Timing Generator and Panel Settings....2730
11.2.4.14 DISPC TV Output....2731
11.2.4.14.1 DISPC Overlay Manager....2732
11.2.4.14.2 DISPC Gamma Correction Unit....2732
11.2.4.14.3 DISPC Synchronized Buffer Update....2732
11.2.4.14.4 DISPC Timing and TV Format Settings....2732
11.2.4.15 DISPC Frame Width Considerations....2734
11.2.4.16 DISPC Extended 3D Support....2735
11.2.4.16.1 DISPC Extended 3D Support - Line Alternative Format....2735
11.2.4.16.3 DISPC Extended 3D Support - Frame Packing Format Format....2737
11.2.4.16.4 DISPC Extended 3D Support - DLP 3D Format....2738
11.2.4.17 DISPC Shadow Registers....2738
11.2.5 DISPC Programming Guide....2745
11.2.5.1 DISPC Low-Level Programming Models....2745
11.2.5.1.1 DISPC Global Initialization....2745
11.2.5.1.2 DISPC Operational Modes Configuration....2745
11.2.6 DISPC Register Manual....2760
11.2.6.1 DISPC Instance Summary....2760
11.2.6.2 DISPC Logical Register Mapping....2760
11.2.6.3 DISPC Registers....2765
11.2.6.3.1 DISPC Register Summary....2765
11.2.6.3.2 DISPC Register Description....2771
11.3 High-Definition Multimedia Interface....2897
11.3.1 HDMI Overview....2897

11.3.1.1 HDMI Main Features....2897
11.3.1.2 HDMI Video Formats and Timings....2898
11.3.1.2.1 HDMI CEA-861-D Video Formats and Timings....2898
11.3.1.2.2 VESA DMT Video Formats and Timings....2899
12 3D Graphics Accelerator....2901
12.1 GPU Overview....2902
12.1.1 GPU Features Overview....2903
12.1.2 Graphics Feature Overview....2903
12.2 GPU Integration....2905
12.3 GPU Functional Description....2907
12.3.1 GPU Block Diagram....2907
12.3.2 GPU Clock Configuration....2908
12.3.3 GPU Software Reset....2908
12.3.4 GPU Power Management....2908
12.3.5 GPU Thermal Management....2908
12.3.6 GPU Interrupt Requests....2909
12.4 GPU Register Manual....2910
12.4.1 GPU Instance Summary....2910
12.4.2 GPU Registers....2910
12.4.2.1 GPU_WRAPPER Register Summary....2910
12.4.2.2 GPU_WRAPPER Register Description....2910
13 2D Graphics Accelerator....2926
13.1 BB2D Overview....2927
13.1.1 BB2D Key Features Overview....2927
13.2 BB2D Integration....2929
13.3 BB2D Functional Description....2931
13.3.1 BB2D Block Diagram....2931
13.3.2 BB2D Clock Configuration....2931
13.3.3 BB2D Software Reset....2932
13.3.4 BB2D Power Management....2932
13.4 BB2D Register Manual....2933
13.4.1 BB2D Instance Summary....2933
13.4.2 BB2D Registers....2933
13.4.2.1 BB2D Register Summary....2933
13.4.2.2 BB2D Register Description....2935
14 Interconnect....2966
14.1 Interconnect Overview....2967
14.1.1 Terminology....2967
14.1.2 Architecture Overview....2968
14.2 L3_MAIN Interconnect....2970
14.2.1 L3_MAIN Interconnect Overview....2970
14.2.2 L3_MAIN Interconnect Integration....2970
14.2.3 L3_MAIN Interconnect Functional Description....2972
14.2.3.1 Module Use in L3_MAIN Interconnect....2972
14.2.3.2 Module Distribution....2972
14.2.3.2.1 L3_MAIN Interconnect Agents....2972
14.2.3.2.2 L3_MAIN Connectivity Matrix....2974
14.2.3.2.3 Master NIU Identification....2977
14.2.3.3 Bandwidth Regulators....2979
14.2.3.4 Bandwidth Limiters....2980
14.2.3.5 Flag Muxing....2981
14.2.3.5.1 Flag Mux Time-out....2982
14.2.3.6 Statistic Collectors Group....2984
14.2.3.7 L3_MAIN Protection and Firewalls....2984
14.2.3.7.1 L3_MAIN Firewall Reset....2985
14.2.3.7.2 Power Management....2986
14.2.3.7.3 L3_MAIN Firewall Functionality....2987
14.2.3.8 L3_MAIN Interconnect Error Handling....2998
14.2.3.8.1 Global Error-Routing Scheme....2998
14.2.3.8.2 Slave NIU Error Logging....2999
14.2.3.8.3 Flag Mux Error Logging....3000

14.2.3.8.4 Severity Level of Standard and Custom Errors....... 3002
14.2.3.8.5 Example for Decoding Standard/Custom Errors Logged in L3_MAIN....... 3002
14.2.4 L3_MAIN Interconnect Programming Guide....... 3003
14.2.4.1 L3 _MAIN Interconnect Low-Level Programming Models....... 3003
14.2.4.1.1 Global Initialization....... 3003
14.2.4.2 Operational Modes Configuration....... 3003
14.2.4.2.1 L3_MAIN Interconnect Error Analysis Mode....... 3003
14.2.5 L3_MAIN Interconnect Register Manual....... 3006
14.2.5.1 L3_MAIN Register Group Summary....... 3006
14.2.5.1.1 L3_MAIN Firewall Registers Summary and Description....... 3006
14.2.5.1.2 L3_MAIN Host Register Summary and Description....... 3018
14.2.5.1.3 L3_MAIN TARG Register Summary and Description....... 3029
14.2.5.1.4 L3_MAIN FLAGMUX Registers Summary and Description....... 3049
14.2.5.1.5 L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description....... 3053
14.2.5.1.6 L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description....... 3056
14.2.5.1.7 L3_MAIN BW Regulator Register Summary and Description....... 3062
14.2.5.1.8 L3_MAIN Bandwidth Limiter Register Summary and Description....... 3071
14.2.5.1.9 L3_MAIN STATCOLL Register Summary and Description....... 3077
14.3 L4 Interconnects....... 3126
14.3.1 L4 Interconnect Overview....... 3126
14.3.2 L4 Interconnect Integration....... 3127
14.3.3 L4 Interconnect Functional Description....... 3128
14.3.3.1 Module Distribution....... 3128
14.3.3.1.1 L4_PER1 Interconnect Agents....... 3128
14.3.3.1.2 L4_PER2 Interconnect Agents....... 3129
14.3.3.1.3 L4_PER3 Interconnect Agents....... 3130
14.3.3.1.4 L4_CFG Interconnect Agents....... 3131
14.3.3.1.5 L4_WKUP Interconnect Agents....... 3132
14.3.3.2 Power Management....... 3133
14.3.3.3 L4 Firewalls....... 3133
14.3.3.3.1 Protection Group....... 3134
14.3.3.3.2 Segments and Regions....... 3139
14.3.3.3.3 L4 Firewall Address and Protection Register Settings....... 3147
14.3.3.4 L4 Error Detection and Reporting....... 3148
14.3.3.4.1 IA and TA Error Detection and Logging....... 3148
14.3.3.4.2 Time-Out....... 3149
14.3.3.4.3 Error Reporting....... 3150
14.3.3.4.4 Error Recovery....... 3151
14.3.3.4.5 Firewall Error Logging in the Control Module....... 3152
14.3.4 L4 Interconnect Programming Guide....... 3153
14.3.4.1 L4 Interconnect Low-level Programming Models....... 3153
14.3.4.1.1 Global Initialization....... 3153
14.3.4.1.2 Operational Modes Configuration....... 3153
14.3.5 L4 Interconnects Register Manual....... 3157
14.3.5.1 L4 Interconnects Instance Summary....... 3157
14.3.5.2 L4 Initiator Agent (L4 IA)....... 3160
14.3.5.2.1 L4 Initiator Agent (L4 IA) Register Summary....... 3160
14.3.5.2.2 L4 Initiator Agent (L4 IA) Register Description....... 3162
14.3.5.3 L4 Target Agent (L4 TA)....... 3170
14.3.5.3.1 L4 Target Agent (L4 TA) Register Summary....... 3170
14.3.5.3.2 L4 Target Agent (L4 TA) Register Description....... 3183
14.3.5.4 L4 Link Agent (L4 LA)....... 3187
14.3.5.4.1 L4 Link Agent (L4 LA) Register Summary....... 3187
14.3.5.4.2 L4 Link Agent (L4 LA) Register Description....... 3187
14.3.5.5 L4 Address Protection (L4 AP)....... 3194
14.3.5.5.1 L4 Address Protection (L4 AP) Register Summary....... 3194
14.3.5.5.2 L4 Address Protection (L4 AP) Register Description....... 3195
15 Memory Subsystem....... 3211
15.1 Memory Subsystem Overview....... 3212
15.1.1 DMM Overview....... 3212
15.1.2 TILER Overview....... 3213

15.1.3 EMIF Overview....3213
15.1.4 GPMC Overview....3214
15.1.5 ELM Overview....3215
15.1.6 OCM Overview....3215
15.2 Dynamic Memory Manager....3217
15.2.1 DMM Overview....3217
15.2.2 DMM Integration....3217
15.2.2.1 DMM Configuration....3219
15.2.3 DMM Functional Description....3220
15.2.3.1 DMM Block Diagram....3220
15.2.3.2 DMM Clock Configuration....3221
15.2.3.3 DMM Power Management....3221
15.2.3.4 DMM Interrupt Requests....3221
15.2.3.5 DMM....3223
15.2.3.5.1 DMM Concepts....3223
15.2.3.5.2 DMM Transaction Flows....3228
15.2.3.5.3 DMM Internal Macro-Architecture....3229
15.2.3.6 TILER....3232
15.2.3.6.1 TILER Concepts....3232
15.2.3.6.2 TILER Macro-Architecture....3250
15.2.3.6.3 TILER Guidelines for Initiators....3253
15.2.4 DMM Use Cases and Tips....3255
15.2.4.1 PAT Use Cases....3255
15.2.4.1.1 Simple Manual Area Refill....3255
15.2.4.1.2 Single Auto-Configured Area Refill....3255
15.2.4.1.3 Chained Auto-Configured Area Refill....3256
15.2.4.1.4 Synchronized Auto-Configured Area Refill....3257
15.2.4.1.5 Cyclic Synchronized Auto-Configured Area Refill....3258
15.2.4.2 Addressing Management with LISA....3259
15.2.4.2.1 Case 1: Use of One Memory Controller....3259
15.2.4.2.2 Case 2: Use of Two Memory Controllers....3260
15.2.5 DMM Basic Programming Model....3263
15.2.5.1 Global Initialization....3263
15.2.5.2 DMM Module Global Initialization....3263
15.2.5.3 DMM Operational Modes Configuration....3263
15.2.5.3.1 Different Operational Modes....3263
15.2.5.3.2 Configuration Settings and LUT Refill....3263
15.2.5.3.3 Interleaving Settings....3264
15.2.5.3.4 Aliased Tiled View Orientation Settings and LUT Refill....3264
15.2.5.3.5 Priority Settings....3264
15.2.5.3.6 Error Handling....3265
15.2.5.3.7 PAT Programming Model....3265
15.2.5.4 Addressing an Object in Tiled Mode....3265
15.2.5.4.1 Frame-Buffer Addressing....3265
15.2.5.4.2 TILER Page Mapping....3267
15.2.5.5 Addressing an Object in Page Mode....3267
15.2.5.6 Sharing Containers Between Different Modes....3268
15.2.6 DMM Register Manual....3269
15.2.6.1 DMM Instance Summary....3269
15.2.6.2 DMM Registers....3269
15.2.6.2.1 DMM Register Summary....3269
15.2.6.2.2 DMM Register Description....3270
15.3 EMIF Controller....3303
15.3.1 EMIF Controller Overview....3303
15.3.2 EMIF Module Environment....3303
15.3.3 EMIF Module Integration....3308
15.3.4 EMIF Functional Description....3311
15.3.4.1 Block Diagram....3311
15.3.4.1.1 Local Interface....3311
15.3.4.1.2 FIFO Description....3312
15.3.4.1.3 MPU Port Restrictions....3313

15.3.4.1.4 Arbitration of Commands in the Command FIFO....3313
15.3.4.2 Clock Management....3314
15.3.4.2.1 EMIF_FICLK Overview....3314
15.3.4.2.2 EMIF Dependency on MPU Clock Rate....3315
15.3.4.3 Reset....3315
15.3.4.4 System Power Management....3315
15.3.4.4.1 Power-Down Mode....3315
15.3.4.4.2 Self-Refresh Mode....3316
15.3.4.5 Interrupt Requests....3317
15.3.4.6 SDRAM Refresh Scheduling....3318
15.3.4.7 SDRAM Initialization....3318
15.3.4.7.1 DDR2 SDRAM Initialization....3318
15.3.4.7.2 DDR3 SDRAM Initialization....3320
15.3.4.8 DDR3 Read-Write Leveling....3322
15.3.4.8.1 Full Leveling....3322
15.3.4.8.2 Software Leveling....3323
15.3.4.9 EMIF Access Cycles....3324
15.3.4.10 Turnaround Time....3324
15.3.4.11 PHY DLL Calibration....3324
15.3.4.12 SDRAM Address Mapping....3325
15.3.4.12.1 Address Mapping for IBANK_POS = 0 and EBANK_POS = 0....3325
15.3.4.12.2 Address Mapping for IBANK_POS = 1 and EBANK_POS = 0....3326
15.3.4.12.3 Address Mapping for IBANK_POS = 2 and EBANK_POS = 0....3326
15.3.4.12.4 Address Mapping for IBANK_POS = 3 and EBANK_POS = 0....3327
15.3.4.12.5 Address Mapping for IBANK_POS = 0 and EBANK_POS = 1....3327
15.3.4.12.6 Address Mapping for IBANK_POS = 1 and EBANK_POS = 1....3328
15.3.4.12.7 Address Mapping for IBANK_POS = 2 and EBANK_POS = 1....3328
15.3.4.12.9 Address Mapping for IBANK_POS = 3 and EBANK_POS = 1....3329
15.3.4.13 DDR3 Output Impedance Calibration....3329
15.3.4.14 Error Correction And Detection Feature....3329
15.3.4.15 Class of Service....3330
15.3.4.16 Performance Counters....3331
15.3.4.16.1 Performance Counters General Examples....3332
15.3.4.17 Forcing CKE to tri-state....3333
15.3.5 EMIF Programming Guide....3335
15.3.5.1 EMIF Low-Level Programming Models....3335
15.3.5.1.1 Global Initialization....3335
15.3.5.1.2 Operational Modes Configuration....3342
15.3.6 EMIF Register Manual....3344
15.3.6.1 EMIF Instance Summary....3344
15.3.6.2 EMIF Registers....3344
15.3.6.2.1 EMIF Register Summary....3344
15.3.6.2.2 EMIF Register Description....3348
15.4 General-Purpose Memory Controller....3446
15.4.1 GPMC Overview....3446
15.4.2 GPMC Environment....3446
15.4.2.1 GPMC Modes....3446
15.4.2.2 GPMC Signals....3449
15.4.3 GPMC Integration....3451
15.4.4 GPMC Functional Description....3454
15.4.4.1 GPMC Block Diagram....3454
15.4.4.2 GPMC Clock Configuration....3455
15.4.4.3 GPMC Software Reset....3456
15.4.4.4 GPMC Power Management....3456
15.4.4.5 GPMC Interrupt Requests....3456
15.4.4.6 L3 Interconnect Interface....3457
15.4.4.7 GPMC Address and Data Bus....3457
15.4.4.7.1 GPMC I/O Configuration Setting....3458
15.4.4.7.2 GPMC CS0 Default Configuration at Device Reset....3458
15.4.4.8 Address Decoder and Chip-Select Configuration....3460
15.4.4.8.1 Chip-Select Base Address and Region Size....3460

15.4.4.8.2 Access Protocol....3461
15.4.4.8.3 External Signals....3462
15.4.4.8.4 Error Handling....3471
15.4.4.9 Timing Setting....3471
15.4.4.9.1 Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME)....3472
15.4.4.9.2 nCS: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY)....3472
15.4.4.9.3 nADV/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME)....3473
15.4.4.9.4 nOE/nRE: Output Enable/Read Enable Signal Control Assertion/Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME)....3473
15.4.4.9.5 nWE: Write Enable Signal Control Assertion/Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY)....3474
15.4.4.9.6 GPMC_CLK....3474
15.4.4.9.7 GPMC_CLK and Control Signals Setup and Hold....3475
15.4.4.9.8 Access Time (RDACCESSTIME / WRACCESSTIME)....3475
15.4.4.9.9 Page Burst Access Time (PAGEBURSTACCESSTIME)....3476
15.4.4.9.10 Bus Keeping Support....3476
15.4.4.10 NOR Access Description....3476
15.4.4.10.1 Asynchronous Access Description....3477
15.4.4.10.2 Synchronous Access Description....3484
15.4.4.10.3 Asynchronous and Synchronous Accesses in Nonmultiplexed Mode....3493
15.4.4.10.4 Page and Burst Support....3497
15.4.4.10.5 System Burst vs External Device Burst Support....3497
15.4.4.11 pSRAM Access Specificities....3498
15.4.4.12 NAND Access Description....3498
15.4.4.12.1 NAND Memory Device in Byte or 16-bit Word Stream Mode....3499
15.4.4.12.2 NAND Device-Ready Pin....3505
15.4.4.12.3 ECC Calculator....3506
15.4.4.12.4 Prefetch and Write-Posting Engine....3522
15.4.5 GPMC Basic Programming Model....3530
15.4.5.1 GPMC High-Level Programming Model Overview....3530
15.4.5.2 GPMC Initialization....3532
15.4.5.3 GPMC Configuration in NOR Mode....3532
15.4.5.4 GPMC Configuration in NAND Mode....3533
15.4.5.5 Set Memory Access....3535
15.4.5.6 GPMC Timing Parameters....3535
15.4.5.6.1 GPMC Timing Parameters Formulas....3538
15.4.6 GPMC Use Cases and Tips....3548
15.4.6.1 How to Set GPMC Timing Parameters for Typical Accesses....3548
15.4.6.1.1 External Memory Attached to the GPMC Module....3548
15.4.6.1.2 Typical GPMC Setup....3548
15.4.6.2 How to Choose a Suitable Memory to Use With the GPMC....3554
15.4.6.2.1 Supported Memories or Devices....3554
15.4.6.2.2 GPMC Features and Settings....3556
15.4.7 GPMC Register Manual....3558
15.4.7.1 GPMC Register Summary....3558
15.4.7.2 GPMC Register Descriptions....3559
15.5 Error Location Module....3582
15.5.1 Error Location Module Overview....3582
15.5.2 ELM Integration....3583
15.5.3 ELM Functional Description....3584
15.5.3.1 ELM Software Reset....3584
15.5.3.2 ELM Power Management....3584
15.5.3.3 ELM Interrupt Requests....3585
15.5.3.4 Processing Initialization....3585
15.5.3.5 Processing Sequence....3586
15.5.3.6 Processing Completion....3586
15.5.4 ELM Basic Programming Model....3587
15.5.4.1 ELM Low-Level Programming Model....3587

15.5.4.1.1 Processing Initialization....3587
15.5.4.1.2 Read Results....3588
15.5.4.2 Use Case: ELM Used in Continuous Mode....3589
15.5.4.3 Use Case: ELM Used in Page Mode....3590
15.5.5 ELM Register Manual....3593
15.5.5.1 ELM Instance Summary....3593
15.5.5.2 ELM Registers....3593
15.5.5.2.1 ELM Register Summary....3593
15.5.5.2.2 ELM Register Description....3594
15.6 On-Chip Memory (OCM) Subsystem....3605
15.6.1 OCM Subsystem Overview....3605
15.6.2 OCM Subsystem Integration....3605
15.6.3 OCM Subsystem Functional Desctiption....3608
15.6.3.1 Block Diagram....3608
15.6.3.2 Resets....3609
15.6.3.3 Clock Management....3609
15.6.3.4 Interrupt Requests....3609
15.6.3.5 OCM Subsystem Memory Regions....3613
15.6.3.6 OCM Controller Modes Of Operation....3613
15.6.3.7 ECC Associated FIFOs....3614
15.6.3.8 ECC Counters And Corrected Bit Distribution Register....3614
15.6.3.9 ECC Support....3615
15.6.3.10 Circular Buffer (CBUF) Support....3617
15.6.3.11 CBUF Mode Error Handling....3618
15.6.3.11.1 VBUF Address Not Mapped to a CBUF Memory Space....3618
15.6.3.11.2 VBUF Access Not Starting At The Base Address....3619
15.6.3.11.3 Illegal Address Change Between Two Same Type Accesses....3619
15.6.3.11.4 Illegal Frame SIze (Short Frame Detection)....3620
15.6.3.11.5 CBUF Overflow....3620
15.6.3.11.6 CBUF Underflow....3621
15.6.3.12 Status Reporting....3622
15.6.4 OCM Subsystem Register Manual....3622
15.6.4.1 OCM Subsystem Instance Summary....3622
15.6.4.2 OCM Subsystem Registers....3622
15.6.4.2.1 OCM Subsystem Register Summary....3622
15.6.4.2.2 OCM Subsystem Register Description....3623
16 DMA Controllers....3658
16.1 System DMA....3659
16.1.1 DMA_SYSTEM Module Overview....3659
16.1.2 DMA_SYSTEM Controller Environment....3661
16.1.3 DMA_SYSTEM Module Integration....3662
16.1.3.1 DMA Requests to the DMA_SYSTEM Controller....3667
16.1.3.2 Mapping of DMA Requests to DMA_CROSSBAR Inputs....3673
16.1.4 DMA_SYSTEM Functional Description....3677
16.1.4.1 DMA_SYSTEM Controller Power Management....3677
16.1.4.2 DMA_SYSTEM Controller Interrupt Requests....3678
16.1.4.2.1 Interrupt Generation....3679
16.1.4.3 Logical Channel Transfer Overview....3679
16.1.4.4 FIFO Queue Memory Pool....3681
16.1.4.5 Addressing Modes....3681
16.1.4.6 Packed Accesses....3686
16.1.4.7 Burst Transactions....3687
16.1.4.8 Endianism Conversion....3687
16.1.4.9 Transfer Synchronization....3687
16.1.4.9.1 Software Synchronization....3687
16.1.4.9.2 Hardware Synchronization....3688
16.1.4.10 Thread Budget Allocation....3690
16.1.4.11 FIFO Budget Allocation....3691
16.1.4.12 Chained Logical Channel Transfers....3691
16.1.4.13 Reprogramming an Active Channel....3692
16.1.4.14 Packet Synchronization....3692

16.1.4.15 Graphics Acceleration Support.......... 3693
16.1.4.16 Supervisor Modes.......... 3693
16.1.4.17 Posted and Nonposted Writes.......... 3694
16.1.4.18 Disabling a Channel During Transfer.......... 3694
16.1.4.19 FIFO Draining Mechanism.......... 3694
16.1.4.20 Linked List.......... 3694
16.1.4.20.1 Overview.......... 3694
16.1.4.20.2 Link-List Transfer Profile.......... 3695
16.1.4.20.3 Descriptors.......... 3695
16.1.4.20.4 Linked-List Control and Monitoring.......... 3698
16.1.5 DMA_SYSTEM Basic Programming Model.......... 3701
16.1.5.1 Setup Configuration.......... 3701
16.1.5.2 Software-Triggered (Nonsynchronized) Transfer.......... 3701
16.1.5.3 Hardware-Synchronized Transfer.......... 3703
16.1.5.4 Synchronized Transfer Monitoring Using CDAC.......... 3705
16.1.5.5 Concurrent Software and Hardware Synchronization.......... 3705
16.1.5.6 Chained Transfer.......... 3706
16.1.5.7 90-Degree Clockwise Image Rotation.......... 3706
16.1.5.8 Graphic Operations.......... 3707
16.1.5.9 Linked-List Programming Guidelines.......... 3708
16.1.6 DMA_SYSTEM Register Manual.......... 3709
16.1.6.1 DMA_SYSTEM Instance Summary.......... 3709
16.1.6.2 DMA_SYSTEM Registers.......... 3709
16.1.6.2.1 DMA_SYSTEM Register Summary.......... 3709
16.1.6.2.2 DMA_SYSTEM Register Description.......... 3710
16.2 Enhanced DMA.......... 3732
16.2.1 EDMA Module Overview.......... 3732
16.2.1.1 EDMA Features.......... 3733
16.2.1.3 EDMA Controllers Configuration.......... 3734
16.2.2 EDMA Controller Environment.......... 3736
16.2.3 EDMA Controller Integration.......... 3737
16.2.3.1 EDMA Requests to the EDMA Controller.......... 3741
16.2.4 EDMA Controller Functional Description.......... 3744
16.2.4.1 Block Diagram.......... 3744
16.2.4.1.1 Third-Party Channel Controller.......... 3744
16.2.4.1.2 Third-Party Transfer Controller.......... 3746
16.2.4.2 Types of EDMA controller Transfers.......... 3747
16.2.4.2.1 A-Synchronized Transfers.......... 3748
16.2.4.2.2 AB-Synchronized Transfers.......... 3750
16.2.4.3 Parameter RAM (PaRAM).......... 3750
16.2.4.3.1 PaRAM.......... 3751
16.2.4.3.2 EDMA Channel PaRAM Set Entry Fields.......... 3754
16.2.4.3.3 Null PaRAM Set.......... 3756
16.2.4.3.4 Dummy PaRAM Set.......... 3756
16.2.4.3.5 Dummy Versus Null Transfer Comparison.......... 3756
16.2.4.3.6 Parameter Set Updates.......... 3757
16.2.4.3.7 Linking Transfers.......... 3759
16.2.4.3.8 Constant Addressing Mode Transfers/Alignment Issues.......... 3762
16.2.4.3.9 Element Size.......... 3762
16.2.4.4 Initiating a DMA Transfer.......... 3762
16.2.4.4.1 DMA Channel.......... 3762
16.2.4.4.2 QDMA Channels.......... 3764
16.2.4.4.3 Comparison Between DMA and QDMA Channels.......... 3764
16.2.4.5 Completion of a DMA Transfer.......... 3765
16.2.4.5.1 Normal Completion.......... 3766
16.2.4.5.2 Early Completion.......... 3766
16.2.4.5.3 Dummy or Null Completion.......... 3766
16.2.4.6 Event, Channel, and PaRAM Mapping.......... 3766
16.2.4.6.1 DMA Channel to PaRAM Mapping.......... 3767
16.2.4.6.2 QDMA Channel to PaRAM Mapping.......... 3767
16.2.4.7 EDMA Channel Controller Regions.......... 3768

16.2.4.7.1 Region Overview....3768
16.2.4.7.2 Channel Controller Regions....3770
16.2.4.7.3 Region Interrupts....3771
16.2.4.8 Chaining EDMA Channels....3771
16.2.4.9 EDMA Interrupts....3772
16.2.4.9.1 Transfer Completion Interrupts....3772
16.2.4.9.2 EDMA Interrupt Servicing....3776
Example 16-2. Interrupt Servicing....3776
Example 16-3. Interrupt Servicing....3777
16.2.4.9.4 Interrupt Evaluation Operations....3777
16.2.4.9.5 Error Interrupts....3779
16.2.4.10 Memory Protection....3781
16.2.4.10.1 Active Memory Protection....3781
16.2.4.10.2 Proxy Memory Protection....3784
16.2.4.11 Event Queue(s)....3786
16.2.4.11.1 DMA/QDMA Channel to Event Queue Mapping....3786
16.2.4.11.2 Queue RAM Debug Visibility....3786
16.2.4.11.3 Queue Resource Tracking....3787
16.2.4.11.4 Performance Considerations....3787
16.2.4.12 EDMA Transfer Controller (EDMA_TPTC)....3787
16.2.4.12.1 Architecture Details....3787
16.2.4.12.2 Memory Protection....3789
16.2.4.12.3 Error Generation....3790
16.2.4.12.4 Debug Features....3790
16.2.4.12.5 EDMA_TPTC Configuration....3790
16.2.4.13 Event Dataflow....3791
16.2.4.14 EDMA controller Prioritization....3791
16.2.4.14.1 Channel Priority....3792
16.2.4.14.2 Trigger Source Priority....3793
16.2.4.14.3 Dequeue Priority....3793
16.2.4.15 EDMA Power, Reset and Clock Management....3793
16.2.4.15.1 Clock and Power Management....3793
16.2.4.15.2 Reset Considerations....3794
16.2.4.16 Emulation Considerations....3794
16.2.5 EDMA Transfer Examples....3795
16.2.5.1 Block Move Example....3795
16.2.5.2 Subframe Extraction Example....3797
16.2.5.3 Data Sorting Example....3798
16.2.5.4 Peripheral Servicing Example....3800
16.2.5.4.1 Non-bursting Peripherals....3800
16.2.5.4.2 Bursting Peripherals....3802
16.2.5.4.3 Continuous Operation....3804
16.2.5.4.4 Ping-Pong Buffering....3807
16.2.5.4.5 Transfer Chaining Examples....3810
16.2.5.5 Setting Up an EDMA Transfer....3814
16.2.6 EDMA Debug Checklist and Programming Tips....3816
16.2.6.1 EDMA Debug Checklist....3816
16.2.6.2 EDMA Programming Tips....3818
16.2.7 EDMA Register Manual....3819
16.2.7.1 EDMA Instance Summary....3819
16.2.7.2 EDMA Registers....3819
16.2.7.2.1 EDMA Register Summary....3819
16.2.7.2.2 EDMA Register Description....3839
17 Interrupt Controllers....3967
17.1 Interrupt Controllers Overview....3968
17.2 Interrupt Controllers Environment....3970
17.3 Interrupt Controllers Integration....3972
17.3.1 Interrupt Requests to MPU_INTC....3972
17.3.2 Interrupt Requests to DSP1_INTC....3981
17.3.3 Interrupt Requests to DSP2_INTC....3987
17.3.4 Interrupt Requests to IPU1_Cx_INTC....3993

17.3.5 Interrupt Requests to IPU2_Cx_INTC.......... 3997
17.3.6 Interrupt Requests to EVE1_INTC1.......... 4001
17.3.7 Interrupt Requests to EVE2_INTC1.......... 4003
17.3.8 Mapping of Device Interrupts to IRQ_CROSSBAR Inputs.......... 4005
17.4 Interrupt Controllers Functional Description.......... 4014
18 Control Module.......... 4015
18.1 Control Module Overview.......... 4016
18.2 Control Module Environment.......... 4018
18.3 Control Module Integration.......... 4019
18.4 Control Module Functional Description.......... 4021
18.4.1 Control Module Clock Configuration.......... 4021
18.4.2 Control Module Resets.......... 4021
18.4.3 Control Module Power Management.......... 4021
18.4.3.1 Power Management Protocols.......... 4021
18.4.4 Hardware Requests.......... 4021
18.4.5 Control Module Initialization.......... 4021
18.4.6 Functional Description Of The Various Register Types In CTRL_MODULE_CORE Submodule.......... 4021
18.4.6.1 Pad Configuration.......... 4022
18.4.6.1.1 Pad Configuration Registers.......... 4022
18.4.6.1.2 Pull Selection.......... 4023
18.4.6.1.3 Pad multiplexing.......... 4024
18.4.6.1.4 IOSETs.......... 4024
18.4.6.1.5 Virtual IO Timing Modes.......... 4024
18.4.6.1.6 Manual IO Timing Modes.......... 4025
18.4.6.1.7 Isolation Requirements.......... 4026
18.4.6.1.8 IO Delay Recalibration.......... 4026
18.4.6.2 Thermal Management Related Registers.......... 4027
18.4.6.2.1 Temperature Sensors Control Registers.......... 4029
18.4.6.2.2 Registers For The Thermal Alert Comparators.......... 4029
18.4.6.2.3 Thermal Shutdown Comparators.......... 4031
18.4.6.2.4 Temperature Timestamp Registers.......... 4031
18.4.6.2.5 Other Thermal Management Related Registers.......... 4032
18.4.6.2.6 Summary of the Thermal Management Related Registers.......... 4033
18.4.6.2.7 ADC Values Versus Temperature.......... 4034
18.4.6.3 PBIAS Cell And MMC1 I/O Cells Control Registers.......... 4037
18.4.6.4 IRQ_CROSSBAR Module Functional Description.......... 4040
18.4.6.5 DMA_CROSSBAR Module Functional Description.......... 4043
18.4.6.6 SDRAM Initiator Priority Registers.......... 4046
18.4.6.7 L3_MAIN Initiator Priority Registers.......... 4046
18.4.6.8 Memory Region Lock Registers.......... 4047
18.4.6.9 NMI Mapping To Respective Cores.......... 4047
18.4.6.10 Software Controls for the DDR2/DDR3 I/O Cells.......... 4047
18.4.6.11 Reference Voltage for the Device DDR2/DDR3 Receivers.......... 4052
18.4.6.12 AVS Class 0 Associated Registers.......... 4054
18.4.6.13 ABB Associated Registers.......... 4056
18.4.6.14 Registers For Other Miscellaneous Functions.......... 4057
18.4.6.14.1 System Boot Status Settings.......... 4057
18.4.6.14.2 Force MPU Write Nonposted Transactions.......... 4057
18.4.6.14.3 Firewall Error Status Registers.......... 4057
18.4.6.14.4 Settings Related To Different Peripheral Modules.......... 4058
18.4.7 Functional Description Of The Various Register Types In CTRL_MODULE_WKUP Submodule.......... 4058
18.4.7.1 Registers For Basic EMIF Configuration.......... 4058
18.5 Control Module Register Manual.......... 4059
18.6 IODELAYCONFIG Module Integration.......... 4671
18.7 IODELAYCONFIG Module Register Manual.......... 4672
19 Mailbox.......... 5106
19.1 Mailbox Overview.......... 5107
19.2 Mailbox Integration.......... 5107
19.2.1 System MAILBOX Integration.......... 5108
19.2.2 IVA Mailbox Integration.......... 5110
19.2.3 EVE Mailbox Integration.......... 5112

19.3 Mailbox Functional Description....5115
19.3.1 Mailbox Block Diagram....5116
19.3.2 Mailbox Software Reset....5116
19.3.3 Mailbox Power Management....5116
19.3.4 Mailbox Interrupt Requests....5117
19.3.5 Mailbox Assignment....5117
19.3.5.1 Description....5117
19.3.6 Sending and Receiving Messages....5118
19.3.6.1 Description....5118
19.3.7 16-Bit Register Access....5118
19.3.7.1 Description....5118
19.3.8 Example of Communication....5120
19.4 Mailbox Programming Guide....5121
19.4.1 Mailbox Low-level Programming Models....5121
19.4.1.1 Global Initialization....5121
19.4.1.1.1 Surrounding Modules Global Initialization....5121
19.4.1.1.2 Mailbox Global Initialization....5121
19.4.1.2 Mailbox Operational Modes Configuration....5122
19.4.1.2.1 Mailbox Processing modes....5122
19.4.1.3 Mailbox Events Servicing....5122
19.4.1.3.1 Events Servicing in Sending Mode....5122
19.4.1.3.2 Events Servicing in Receiving Mode....5123
19.5 Mailbox Register Manual....5124
19.5.1 Mailbox Instance Summary....5124
19.5.2 Mailbox Registers....5124
19.5.2.1 Mailbox Register Summary....5124
19.5.2.2 Mailbox Register Description....5128
20 Memory Management Units....5147
20.1 MMU Overview....5148
20.2 MMU Integration....5151
20.3 MMU Functional Description....5153
20.3.1 MMU Block Diagram....5153
20.3.1.1 MMU Address Translation Process....5153
20.3.1.2 Translation Tables....5154
20.3.1.2.1 Translation Table Hierarchy....5154
20.3.1.2.2 First-Level Translation Table....5155
20.3.1.2.3 Two-Level Translation....5158
20.3.1.3 Translation Lookaside Buffer....5161
20.3.1.3.1 TLB Entry Format....5162
20.3.1.4 No Translation (Bypass) Regions....5163
20.3.2 MMU Software Reset....5163
20.3.3 MMU Power Management....5163
20.3.4 MMU Interrupt Requests....5163
20.3.5 MMU Error Handling....5164
20.4 MMU Low-level Programming Models....5165
20.4.1 Global Initialization....5165
20.4.1.1 Surrounding Modules Global Initialization....5165
20.4.1.2 MMU Global Initialization....5165
20.4.1.2.1 Main Sequence - MMU Global Initialization....5165
20.4.1.2.2 Subsequence - Configure a TLB entry....5166
20.4.1.3 Operational Modes Configuration....5167
20.4.1.3.1 Main Sequence - Writing TLB Entries Statically....5167
20.4.1.3.2 Main Sequence - Protecting TLB Entries....5167
20.4.1.3.3 Main Sequence - Deleting TLB Entries....5167
20.4.1.3.4 Main Sequence - Read TLB Entries....5167
20.5 MMU Register Manual....5168
20.5.1 MMU Instance Summary....5168
20.5.2 MMU Registers....5168
20.5.2.1 MMU Register Summary....5168
20.5.2.2 MMU Register Description....5173
21 Spinlock....5187

21.1 Spinlock Overview....5188
21.2 Spinlock Integration....5189
21.3 Spinlock Functional Description....5190
21.3.1 Spinlock Software Reset....5190
21.3.2 Spinlock Power Management....5190
21.3.3 About Spinlocks....5190
21.3.4 Spinlock Functional Operation....5191
21.4 Spinlock Programming Guide....5192
21.4.1 Spinlock Low-level Programming Models....5192
21.4.1.1 Surrounding Modules Global Initialization....5192
21.4.1.2 Basic Spinlock Operations....5192
21.4.1.2.1 Spinlocks Clearing After a System Bug Recovery....5192
21.4.1.2.2 Take and Release Spinlock....5192
21.5 Spinlock Register Manual....5194
21.5.1 Spinlock Instance Summary....5194
21.5.2 Spinlock Registers....5194
21.5.2.1 Spinlock Register Summary....5194
21.5.2.2 Spinlock Register Description....5194
22 Timers....5198
22.1 Timers Overview....5199
22.2 General-Purpose Timers....5200
22.2.1 General-Purpose Timers Overview....5200
22.2.1.1 GP Timer Features....5201
22.2.2 GP Timer Environment....5202
22.2.2.1 GP Timer External System Interface....5202
22.2.3 GP Timer Integration....5204
22.2.4 GP Timer Functional Description....5208
22.2.4.1 GP Timer Block Diagram....5208
22.2.4.2 TIMER1, TIMER2 and TIMER10 Power Management....5211
22.2.4.2.1 Wake-Up Capability....5211
22.2.4.3 Power Management of Other GP Timers....5212
22.2.4.3.1 Wake-Up Capability....5212
22.2.4.4 Software Reset....5213
22.2.4.5 GP Timer Interrupts....5213
22.2.4.6 Timer Mode Functionality....5213
22.2.4.6.1 1-ms Tick Generation (Only TIMER1, TIMER2 and TIMER10)....5214
22.2.4.7 Capture Mode Functionality....5216
22.2.4.8 Compare Mode Functionality....5218
22.2.4.9 Prescaler Functionality....5218
22.2.4.10 Pulse-Width Modulation....5218
22.2.4.11 Timer Counting Rate....5220
22.2.4.12 Timer Under Emulation....5221
22.2.4.13 Accessing GP Timer Registers....5221
22.2.4.13.1 Writing to Timer Registers....5222
22.2.4.13.2 Reading From Timer Counter Registers....5223
22.2.4.14 Posted Mode Selection....5224
22.2.5 GP Timer Low-Level Programming Models....5226
22.2.5.1 Global Initialization....5226
22.2.5.1.1 Global Initialization of Surrounding Modules....5226
22.2.5.1.2 GP Timer Module Global Initialization....5226
22.2.5.2 Operational Mode Configuration....5226
22.2.5.2.1 GP Timer Mode....5226
22.2.5.2.2 GP Timer Compare Mode....5227
22.2.5.2.3 GP Timer Capture Mode....5227
22.2.5.2.4 GP Timer PWM Mode....5228
22.2.6 GP Timer Register Manual....5229
22.2.6.1 GP Timer Instance Summary....5229
22.2.6.2 GP Timer Registers....5229
22.2.6.2.1 GP Timer Register Summary....5229
22.2.6.2.2 GP Timer Register Description....5232
22.2.6.2.3 TIMER1, TIMER2, and TIMER10 Register Description....5245

22.3 32-kHz Synchronized Timer (COUNTER_32K)........ 5249
22.3.1 32-kHz Synchronized Timer Overview........ 5249
22.3.1.1 32-kHz Synchronized Timer Features........ 5249
22.3.2 32-kHz Synchronized Timer Integration........ 5250
22.3.3 32-kHz Synchronized Timer Functional Description........ 5251
22.3.3.1 Reading the 32-kHz Synchronized Timer........ 5251
22.3.4 COUNTER_32K Timer Register Manual........ 5253
22.3.4.1 COUNTER_32K Timer Register Mapping Summary........ 5253
22.3.4.2 COUNTER_32K Timer Register Description........ 5254
22.4 Watchdog Timer........ 5256
22.4.1 Watchdog Timer Overview........ 5256
22.4.1.1 Watchdog Timer Features........ 5256
22.4.2 Watchdog Timer Integration........ 5258
22.4.3 Watchdog Timer Functional Description........ 5260
22.4.3.1 Power Management........ 5260
22.4.3.1.1 Wake-Up Capability........ 5260
22.4.3.2 Interrupts........ 5260
22.4.3.3 General Watchdog Timer Operation........ 5261
22.4.3.4 Reset Context........ 5261
22.4.3.5 Overflow/Reset Generation........ 5261
22.4.3.6 Prescaler Value/Timer Reset Frequency........ 5262
22.4.3.7 Triggering a Timer Reload........ 5263
22.4.3.8 Start/Stop Sequence for Watchdog Timer (Using the WSPR Register)........ 5263
22.4.3.9 Modifying Timer Count/Load Values and Prescaler Setting........ 5263
22.4.3.10 Watchdog Counter Register Access Restriction (WCRR)........ 5264
22.4.3.11 Watchdog Timer Interrupt Generation........ 5264
22.4.3.12 Watchdog Timer Under Emulation........ 5265
22.4.3.13 Accessing Watchdog Timer Registers........ 5265
22.4.4 Watchdog Timer Low-Level Programming Model........ 5266
22.4.4.1 Global Initialization........ 5266
22.4.4.1.1 Surrounding Modules Global Initialization........ 5266
22.4.4.1.2 Watchdog Timer Module Global Initialization........ 5267
22.4.4.2 Operational Mode Configuration........ 5267
22.4.4.2.1 Watchdog Timer Basic Configuration........ 5267
22.4.5 Watchdog Timer Register Manual........ 5267
22.4.5.1 Watchdog Timer Instance Summary........ 5267
22.4.5.2 Watchdog Timer Registers........ 5268
22.4.5.2.1 Watchdog Timer Register Summary........ 5268
22.4.5.2.3 Watchdog Timer Register Description........ 5270
23 Real-Time Clock (RTC)........ 5279
23.1 RTC Overview........ 5280
23.1.1 RTC Features........ 5280
23.2 RTC Environment........ 5282
23.2.1 RTC External Interface........ 5282
23.3 RTC Integration........ 5283
23.4 RTC Functional Description........ 5285
23.4.1 Clock Source........ 5285
23.4.2 Interrupt Support........ 5285
23.4.2.1 CPU Interrupts........ 5285
23.4.2.2 Interrupt Description........ 5286
23.4.2.2.1 Timer Interrupt (timer_intr)........ 5286
23.4.2.2.2 Alarm Interrupt (alarm_intr)........ 5286
23.4.3 RTC Programming/Usage Guide........ 5286
23.4.3.1 Time/Calendar Data Format........ 5286
23.4.3.2 Register Access........ 5287
23.4.3.3 Register Spurious Write Protection........ 5287
23.4.3.4 Reading the Timer/Calendar (TC) Registers........ 5288
23.4.3.4.1 Rounding Seconds........ 5289
23.4.3.5 Modifying the TC Registers........ 5289
23.4.3.5.1 General Registers........ 5290
23.4.3.6 Crystal Compensation........ 5290

23.4.4 Scratch Registers....5291
23.4.5 Debouncing....5291
23.4.6 Power Management....5292
23.4.6.1 Device-Level Power Management....5292
23.4.6.2 Subsystem-Level Power Management — PMIC Mode....5292
23.5 RTC Low-Level Programming Guide....5293
23.5.1 Global Initialization....5293
23.5.1.1 Surrounding Modules Global Initialization....5293
23.5.1.2 RTC Module Global Initialization....5293
23.5.1.2.1 Main Sequence – RTC Module Global Initialization....5293
23.6 RTC Register Manual....5294
23.6.1 RTC Instance Summary....5294
23.6.2 RTC_SS Registers....5294
23.6.2.1 RTC_SS Register Summary....5294
23.6.2.2 RTC_SS Register Description....5294
24 Serial Communication Interfaces....5315
24.1 Multimaster High-Speed I^{2}C Controller....5316
24.1.1 HS I^{2}C Overview....5316
24.1.2 HS I^{2}C Environment....5319
24.1.2.1 HS I^{2}C Typical Application....5319
24.1.2.1.1 HS I^{2}C Pins for Typical Connections in I^{2}C Mode....5319
24.1.2.1.2 HS I^{2}C Interface Typical Connections....5319
24.1.2.2 HS I^{2}C Typical Connection Protocol and Data Format....5320
24.1.2.2.1 HS I^{2}C Serial Data Format....5320
24.1.2.2.2 HS I^{2}C Data Validity....5320
24.1.2.2.3 HS I^{2}C Start and Stop Conditions....5320
24.1.2.2.4 HS I^{2}C Addressing....5321
24.1.2.2.5 HS I^{2}C Master Transmitter....5322
24.1.2.2.6 HS I^{2}C Master Receiver....5322
24.1.2.2.7 HS I^{2}C Slave Transmitter....5322
24.1.2.2.8 HS I^{2}C Slave Receiver....5322
24.1.2.2.9 HS I^{2}C Bus Arbitration....5323
24.1.2.2.10 HS I^{2}C Clock Generation and Synchronization....5323
24.1.3 HS I^{2}C Integration....5325
24.1.4 HS I^{2}C Functional Description....5329
24.1.4.1 HS I^{2}C Block Diagram....5329
24.1.4.2 HS I^{2}C Clocks....5329
24.1.4.2.1 HS I^{2}C Clocking....5329
24.1.4.2.2 HS I^{2}C Automatic Blocking of the I^{2}C Clock Feature....5331
24.1.4.3 HS I^{2}C Software Reset....5331
24.1.4.4 HS I^{2}C Power Management....5332
24.1.4.5 HS I^{2}C Interrupt Requests....5333
24.1.4.6 HS I^{2}C DMA Requests....5334
24.1.4.7 HS I^{2}C Programmable Multislave Channel Feature....5334
24.1.4.8 HS I^{2}C FIFO Management....5334
24.1.4.8.1 HS I^{2}C FIFO Interrupt Mode....5334
24.1.4.8.2 HS I^{2}C FIFO Polling Mode....5336
24.1.4.8.3 HS I^{2}C FIFO DMA Mode....5336
24.1.4.8.4 HS I^{2}C Draining Feature....5338
24.1.4.9 HS I^{2}C Noise Filter....5339
24.1.4.10 HS I^{2}C System Test Mode....5339
24.1.5 HS I^{2}C Programming Guide....5341
24.1.5.1 HS I^{2}C Low-Level Programming Models....5341
24.1.5.1.1 HS I^{2}C Programming Model....5341
24.1.6 HS I^{2}C Register Manual....5355
24.1.6.1 HS I^{2}C Instance Summary....5355
24.1.6.2 HS I^{2}C Registers....5355
24.1.6.2.1 HS I^{2}C Register Summary....5355
24.1.6.2.2 HS I^{2}C Register Description....5357
24.2 HDQ/1-Wire....5382
24.2.1 HDQ1W Overview....5382

24.2.2 HDQ1W Environment....5382
24.2.2.1 HDQ1W Functional Modes....5382
24.2.2.2 HDQ and 1-Wire (SDQ) Protocols....5383
24.2.2.2.1 HDQ Protocol Initialization (Default)....5383
24.2.2.2.2 1-Wire (SDQ) Protocol Initialization....5384
24.2.2.2.3 Communication Sequence (HDQ and 1-Wire Protocols)....5384
24.2.3 HDQ1W Integration....5385
24.2.4 HDQ1W Functional Description....5388
24.2.4.1 HDQ1W Block Diagram....5388
24.2.4.2 HDQ1W Clocking Configuration....5389
24.2.4.2.1 HDQ1W Clocks....5389
24.2.4.3 HDQ1W Hardware and Software Reset....5390
24.2.4.4 HDQ1W Power Management....5390
24.2.4.4.1 Auto-Idle Mode....5390
24.2.4.4.2 Power-Down Mode....5390
24.2.4.5 HDQ Interrupt Requests....5390
24.2.4.6 HDQ Mode (Default)....5391
24.2.4.6.1 HDQ Mode Features....5391
24.2.4.6.2 Description....5391
24.2.4.6.3 Single-Bit Mode....5391
24.2.4.6.4 Interrupt Conditions....5391
24.2.4.7 1-Wire Mode....5392
24.2.4.7.1 1-Wire Mode Features....5392
24.2.4.7.2 Description....5392
24.2.4.7.3 1-Wire Single-Bit Mode Operation....5393
24.2.4.7.4 Interrupt Conditions....5393
24.2.4.7.5 Status Flags....5393
24.2.4.8 BITFSM Delay....5393
24.2.5 HDQ1W Low-Level Programming Model....5393
24.2.5.1 Global Initialization....5394
24.2.5.1.1 Surrounding Modules Global Initialization....5394
24.2.5.1.2 HDQ1W Module Global Initialization....5394
24.2.5.2 HDQ Operational Modes Configuration....5394
24.2.5.2.1 Main Sequence - HDQ Write Operation Mode....5394
24.2.5.2.2 Main Sequence - HDQ Read Operation Mode....5395
24.2.5.3 1-Wire Operational Modes Configuration....5395
24.2.5.3.1 Main Sequence - 1-Wire Write Operation Mode....5395
24.2.5.3.2 Main Sequence - 1-Wire Read Operation Mode....5395
24.2.5.3.3 Sub-sequence - Initialize 1-Wire Slave....5396
24.2.6 HDQ1W Register Manual....5396
24.2.6.1 HDQ1W Instance Summary....5396
24.2.6.2 HDQ1W Registers....5396
24.2.6.2.1 HDQ1W Register Summary....5396
24.2.6.2.2 HDQ1W Register Description....5397
24.3 UART/IrDA/CIR....5400
24.3.1 UART/IrDA/CIR Overview....5400
24.3.1.1 UART Features....5400
24.3.1.2 IrDA Features....5401
24.3.1.3 CIR Features....5401
24.3.2 UART/IrDA/CIR Environment....5402
24.3.2.1 UART Interface....5402
24.3.2.1.1 System Using UART Communication With Hardware Handshake....5402
24.3.2.1.2 UART Interface Description....5402
24.3.2.1.3 UART Protocol and Data Format....5403
24.3.2.2 IrDA Functional Interfaces....5404
24.3.2.2.1 System Using IrDA Communication Protocol....5404
24.3.2.2.2 IrDA Interface Description....5404
24.3.2.2.3 IrDA Protocol and Data Format....5405
24.3.2.3 CIR Functional Interfaces....5411
24.3.2.3.1 System Using CIR Communication Protocol With Remote Control....5411
24.3.2.3.2 CIR Interface Description....5411

24.3.2.3.3 CIR Protocol and Data Format.......... 5411
24.3.3 UART/IrDA/CIR Integration.......... 5416
24.3.4 UART/IrDA/CIR Functional Description.......... 5420
24.3.4.1 Block Diagram.......... 5420
24.3.4.2 Clock Configuration.......... 5421
24.3.4.3 Software Reset.......... 5421
24.3.4.4 Power Management.......... 5421
24.3.4.4.1 UART Mode Power Management.......... 5421
24.3.4.4.2 IrDA Mode Power Management (UART3 Only).......... 5422
24.3.4.4.3 CIR Mode Power Management (UART3 Only).......... 5422
24.3.4.4.4 Local Power Management.......... 5422
24.3.4.5 Interrupt Requests.......... 5423
24.3.4.5.1 UART Mode Interrupt Management.......... 5423
24.3.4.5.2 IrDA Mode Interrupt Management.......... 5424
24.3.4.5.3 CIR Mode Interrupt Management.......... 5425
24.3.4.6 FIFO Management.......... 5425
24.3.4.6.1 FIFO Trigger.......... 5426
24.3.4.6.2 FIFO Interrupt Mode.......... 5427
24.3.4.6.3 FIFO Polled Mode Operation.......... 5428
24.3.4.6.4 FIFO DMA Mode Operation.......... 5429
24.3.4.7 Mode Selection.......... 5435
24.3.4.7.1 Register Access Modes.......... 5435
24.3.4.7.2 UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection.......... 5437
24.3.4.8 Protocol Formatting.......... 5440
24.3.4.8.1 UART Mode.......... 5440
24.3.4.8.2 IrDA Mode (UART3 Only).......... 5447
24.3.4.8.3 CIR Mode (UART3 Only).......... 5450
24.3.5 UART/IrDA/CIR Basic Programming Model.......... 5453
24.3.5.1 Global Initialization.......... 5453
24.3.5.1.1 Surrounding Modules Global Initialization.......... 5453
24.3.5.1.2 UART/IrDA/CIR Module Global Initialization.......... 5453
24.3.5.2 Mode selection.......... 5453
24.3.5.3 Submode selection.......... 5454
24.3.5.4 Load FIFO trigger and DMA mode settings.......... 5454
24.3.5.4.1 DMA mode Settings.......... 5454
24.3.5.4.2 FIFO Trigger Settings.......... 5455
24.3.5.5 Protocol, Baud rate and interrupt settings.......... 5455
24.3.5.5.1 Baud rate settings.......... 5455
24.3.5.5.2 Interrupt settings.......... 5455
24.3.5.5.3 Protocol settings.......... 5456
24.3.5.5.4 UART/IrDA(SIR/MIR/FIR)/CIR.......... 5456
24.3.5.6 Hardware and Software Flow Control Configuration.......... 5456
24.3.5.6.1 Hardware Flow Control Configuration.......... 5456
24.3.5.6.2 Software Flow Control Configuration.......... 5456
24.3.5.7 IrDA Programming Model (UART3 Only).......... 5457
24.3.5.7.1 SIR mode.......... 5457
24.3.5.7.2 MIR mode.......... 5458
24.3.5.7.3 FIR mode.......... 5458
24.3.6 UART/IrDA/CIR Register Manual.......... 5460
24.3.6.1 UART/IrDA/CIR Instance Summary.......... 5460
24.3.6.2 UART/IrDA/CIR Registers.......... 5460
24.3.6.2.1 UART/IrDA/CIR Register Summary.......... 5460
24.3.6.2.2 UART/IrDA/CIR Register Description.......... 5465
24.4 Multichannel Serial Peripheral Interface.......... 5504
24.4.1 McSPI Overview.......... 5504
24.4.2 McSPI Environment.......... 5505
24.4.2.1 Basic McSPI Pins for Master Mode.......... 5505
24.4.2.2 Basic McSPI Pins for Slave Mode.......... 5505
24.4.2.3 Multichannel SPI Protocol and Data Format.......... 5506
24.4.2.3.1 Transfer Format.......... 5508
24.4.2.4 SPI in Master Mode.......... 5510

24.4.2.5 SPI in Slave Mode....5511
24.4.3 McSPI Integration....5513
24.4.4 McSPI Functional Description....5517
24.4.4.1 McSPI Block Diagram....5517
24.4.4.2 Reset....5517
24.4.4.3 Master Mode....5518
24.4.4.3.1 Master Mode Features....5518
24.4.4.3.2 Master Transmit-and-Receive Mode (Full Duplex)....5518
24.4.4.3.3 Master Transmit-Only Mode (Half Duplex)....5519
24.4.4.3.4 Master Receive-Only Mode (Half Duplex)....5519
24.4.4.3.5 Single-Channel Master Mode....5520
24.4.4.3.6 Start-Bit Mode....5521
24.4.4.3.7 Chip-Select Timing Control....5522
24.4.4.3.8 Programmable SPI Clock....5522
24.4.4.4 Slave Mode....5524
24.4.4.4.1 Dedicated Resources....5524
24.4.4.4.2 Slave Transmit-and-Receive Mode....5526
24.4.4.4.3 Slave Transmit-Only Mode....5526
24.4.4.4.4 Slave Receive-Only Mode....5527
24.4.4.5 3-Pin or 4-Pin Mode....5528
24.4.4.6 FIFO Buffer Management....5528
24.4.4.6.1 Buffer Almost Full....5530
24.4.4.6.2 Buffer Almost Empty....5530
24.4.4.6.3 End of Transfer Management....5531
24.4.4.7 Interrupts....5531
24.4.4.7.1 Interrupt Events in Master Mode....5531
24.4.4.7.2 Interrupt Events in Slave Mode....5532
24.4.4.7.3 Interrupt-Driven Operation....5534
24.4.4.7.4 Polling....5534
24.4.4.8 DMA Requests....5534
24.4.4.9 Power Saving Management....5535
24.4.4.9.1 Normal Mode....5535
24.4.4.9.2 Idle Mode....5535
24.4.5 McSPI Programming Guide....5538
24.4.5.1 Global Initialization....5538
24.4.5.1.1 Surrounding Modules Global Initialization....5538
24.4.5.1.2 McSPI Global Initialization....5538
24.4.5.2 Operational Mode Configuration....5538
24.4.5.2.1 McSPI Operational Modes....5538
24.4.5.3 Common Transfer Procedures Without FIFO – Polling Method....5550
24.4.5.3.1 Receive-Only Procedure – Polling Method....5550
24.4.5.3.2 Receive-Only Procedure – Interrupt Method....5551
24.4.5.3.3 Transmit-Only Procedure – Polling Method....5551
24.4.5.3.4 Transmit-and-Receive Procedure – Polling Method....5551
24.4.6 McSPI Register Manual....5552
24.4.6.1 McSPI Instance Summary....5552
24.4.6.2 McSPI Registers....5552
24.4.6.2.1 McSPI Register Summary....5552
24.4.6.2.2 McSPI Register Description....5553
24.5 Quad Serial Peripheral Interface....5577
24.5.1 Quad Serial Peripheral Interface Overview....5577
24.5.2 QSPI Environment....5578
24.5.3 QSPI Integration....5579
24.5.4 QSPI Functional Description....5580
24.5.4.1 QSPI Block Diagram....5580
24.5.4.1.1 SFI Register Control....5582
24.5.4.1.2 SFI Translator....5582
24.5.4.1.3 SPI Control Interface....5583
24.5.4.1.4 SPI Clock Generator....5584
24.5.4.1.5 SPI Control State-Machine....5584
24.5.4.1.6 SPI Data Shifter....5585

24.5.4.2 QSPI Clock Configuration.. 5586
24.5.4.3 QSPI Interrupt Requests.. 5586
24.5.4.4 QSPI Memory Regions.. 5588
24.5.5 QSPI Register Manual.. 5590
24.5.5.1 QSPI Instance Summary.. 5590
24.5.5.2 QSPI registers.. 5590
24.5.5.2.1 QSPI Register Summary.. 5590
24.5.5.2.2 QSPI Register Description.. 5590
24.6 Multichannel Audio Serial Port.. 5606
24.6.1 McASP Overview.. 5606
24.6.2 McASP Environment.. 5609
24.6.2.1 McASP Signals.. 5609
24.6.2.2 Protocols and Data Formats.. 5612
24.6.2.2.1 Protocols Supported.. 5612
24.6.2.2.2 Definition of Terms.. 5613
24.6.2.2.3 TDM Format.. 5616
24.6.2.2.4 I2S Format.. 5617
24.6.2.2.5 S/PDIF Coding Format.. 5618
24.6.3 McASP Integration.. 5621
24.6.4 McASP Functional Description.. 5627
24.6.4.1 McASP Block Diagram.. 5627
24.6.4.2 McASP Clock and Frame-Sync Configurations.. 5629
24.6.4.2.1 McASP Transmit Clock.. 5629
24.6.4.2.2 McASP Receive Clock.. 5631
24.6.4.2.3 Frame-Sync Generator.. 5632
24.6.4.2.4 Synchronous and Asynchronous Transmit and Receive Operations.. 5634
24.6.4.3 Serializers.. 5634
24.6.4.4 Format Units.. 5636
24.6.4.4.1 Transmit Format Unit.. 5636
24.6.4.4.2 Receive Format Unit.. 5639
24.6.4.5 State-Machines.. 5641
24.6.4.6 TDM Sequencers.. 5641
24.6.4.7 McASP Software Reset.. 5642
24.6.4.8 McASP Power Management.. 5642
24.6.4.9 Transfer Modes.. 5642
24.6.4.9.1 Burst Transfer Mode.. 5642
24.6.4.9.2 Time-Division Multiplexed (TDM) Transfer Mode.. 5643
24.6.4.9.3 DIT Transfer Mode.. 5646
24.6.4.10 Data Transmission and Reception.. 5648
24.6.4.10.1 Data Ready Status and Event/Interrupt Generation.. 5649
24.6.4.11 McASP Audio FIFO (AFIFO).. 5654
24.6.4.11.1 AFIFO Data Transmission.. 5655
24.6.4.11.2 AFIFO Data Reception.. 5656
24.6.4.11.3 Arbitration Between Transmit and Receive DMA Requests.. 5656
24.6.4.12 McASP Events and Interrupt Requests.. 5656
24.6.4.12.1 Transmit Data Ready Event and Interrupt.. 5657
24.6.4.12.2 Receive Data Ready Event and Interrupt.. 5657
24.6.4.12.3 Error Interrupt.. 5658
24.6.4.12.4 Multiple Interrupts.. 5658
24.6.4.13 DMA Requests.. 5658
24.6.4.14 Loopback Modes.. 5659
24.6.4.14.1 Loopback Mode Configurations.. 5661
24.6.4.15 Error Reporting.. 5661
24.6.4.15.1 Buffer Underrun Error -Transmitter.. 5661
24.6.4.15.2 Buffer Overrun Error-Receiver.. 5662
24.6.4.15.3 DATA Port Error - Transmitter.. 5662
24.6.4.15.4 DATA Port Error - Receiver.. 5662
24.6.4.15.5 Unexpected Frame Sync Error.. 5662
24.6.4.15.6 Clock Failure Detection.. 5663
24.6.5 McASP Low-Level Programming Model.. 5666
24.6.5.1 Global Initialization.. 5666

24.6.5.1.1 Surrounding Modules Global Initialization.... 5666
24.6.5.1.2 McASP Global Initialization.... 5666
24.6.5.2 Operational Modes Configuration.... 5676
24.6.5.2.1 McASP Transmission Modes.... 5676
24.6.5.2.2 McASP Reception Modes.... 5680
24.6.5.2.3 McASP Event Servicing.... 5685
24.6.6 McASP Register Manual.... 5691
24.6.6.1 McASP Instance Summary.... 5691
24.6.6.2 McASP Registers.... 5691
24.6.6.2.1 MCASP_CFG Register Summary.... 5691
24.6.6.2.2 MCASP_CFG Register Description.... 5696
24.6.6.2.3 MCASP_AFIFO Register Summary.... 5740
24.6.6.2.4 MCASP_AFIFO Register Description.... 5740
24.6.6.2.5 MCASP_DAT Register Summary.... 5743
24.6.6.2.6 MCASP_DAT Register Description.... 5744
24.7 SuperSpeed USB DRD.... 5746
24.7.1 SuperSpeed USB DRD Subsystem Overview.... 5746
24.7.1.1 Main Features.... 5749
24.7.2 SuperSpeed USB DRD Subsystem Environment.... 5751
24.7.2.1 SuperSpeed USB DRD Subsystem I/O Interfaces.... 5751
24.7.2.2 SuperSpeed USB Subsystem Application.... 5753
24.7.2.2.1 USB3.0 DRD Application.... 5753
24.7.2.2.2 USB2.0 DRD Internal PHY.... 5755
24.7.2.2.3 USB2.0 DRD External PHY.... 5756
24.7.2.2.5 Host Mode.... 5756
24.7.2.2.6 Device Mode.... 5756
24.7.3 SuperSpeed USB Subsystem Integration.... 5756
24.8 SATA Controller.... 5761
24.8.1 SATA Controller Overview.... 5761
24.8.1.1 SATA Controller.... 5762
24.8.1.1.1 AHCI Mode Overview.... 5762
24.8.1.1.2 Native Command Queuing.... 5763
24.8.1.1.3 SATA Transport Layer Functionalities.... 5763
24.8.1.1.4 SATA Link Layer Functionalities.... 5763
24.8.1.2 SATA Controller Features.... 5763
24.8.2 SATA Controller Environment.... 5764
24.8.3 SATA Controller Integration.... 5767
24.8.4 SATA Controller Functional Description.... 5769
24.8.4.1 SATA Controller Block Diagram.... 5769
24.8.4.2 SATA Controller Link Layer Protocol and Data Format.... 5769
24.8.4.2.1 SATA 8b/10b Parallel Encoding/Decoding.... 5770
24.8.4.2.2 SATA Stream Dword Components.... 5771
24.8.4.2.3 Scrambling/Descrambling Processing.... 5773
24.8.4.3 Resets.... 5773
24.8.4.3.1 Hardware Reset.... 5773
24.8.4.3.2 Software Initiated Resets.... 5773
24.8.4.4 Power Management.... 5774
24.8.4.4.1 SATA Specific Power Management.... 5774
24.8.4.4.2 Master Standby and Slave Idle Management Protocols.... 5775
24.8.4.4.3 Clock Gating Synchronization.... 5776
24.8.4.5 Interrupt Requests.... 5777
24.8.4.5.1 Interrupt Generation.... 5777
24.8.4.5.2 Levels of Interrupt Control.... 5777
24.8.4.5.3 Interrupt Events Description.... 5778
24.8.4.5.4 Interrupt Condition Control.... 5780
24.8.4.5.5 Command Completion Coalescing Interrupts.... 5781
24.8.4.6 System Memory FIS Descriptors.... 5782
24.8.4.6.1 Command List Structure Basics.... 5782
24.8.4.6.2 Supported Types of Commands.... 5784
24.8.4.6.3 Received FIS Structures.... 5784
24.8.4.6.4 FIS Descriptors Summary.... 5784

24.8.4.7 Transport Layer FIS-Based Interactions....5785
24.8.4.7.1 Software Processing of the Port Command List....5785
24.8.4.7.2 Handling the Received FIS Descriptors....5785
24.8.4.8 DMA Port Configuration....5786
24.8.4.9 Port Multiplier Operation....5786
24.8.4.9.1 Command-Based Switching Mode....5786
24.8.4.9.2 Port Multiplier Enumeration....5787
24.8.4.10 Activity LED Generation Functionality....5787
24.8.4.11 Supported Types of SATA Transfers....5787
24.8.4.11.1 Supported Higher Level Protocols....5787
24.8.4.12 SATA Controller AHCI Hardware Register Interface....5787
24.8.5 SATA Controller Low Level Programming Model....5789
24.8.5.1 Global Initialization....5789
24.8.5.1.1 Surrounding Modules Global Initialization....5789
24.8.5.1.2 SATA Controller Global Initialization....5789
24.8.5.1.3 Issue Command - Main Sequence....5791
24.8.5.1.4 Receive FIS—Main Sequence....5792
24.8.6 SATA Controller Register Manual....5792
24.8.6.1 SATA Controller Instance Summary....5792
24.8.6.2 DWC_ahsata Registers....5792
24.8.6.2.1 DWC_ahsata Register Summary....5792
24.8.6.2.2 DWC_ahsata Register Description....5793
24.8.6.3 SATAMAC_wrapper Registers....5824
24.8.6.3.1 SATAMAC_wrapper Register Summary....5824
24.8.6.3.2 SATAMAC_wrapper Register Description....5824
24.9 PCIe Controller....5827
24.9.1 PCIe Controller Subsystem Overview....5827
24.9.1.1 PCIe Controllers Key Features....5828
24.9.2 PCIe Controller Environment....5830
24.9.3 PCIe Controllers Integration....5831
24.9.4 PCIe SS Controller Functional Description....5834
24.9.4.1 PCIe Controller Functional Block Diagram....5834
24.9.4.2 PCIe Traffics....5836
24.9.4.3 PCIe Controller Ports on L3_MAIN Interconnect....5836
24.9.4.3.1 PCIe Controller Master Port....5837
24.9.4.3.2 PCIe Controller Slave Port....5838
24.9.4.4 PCIe Controller Reset Management....5839
24.9.4.4.1 PCIe Reset Types and Stickiness....5839
24.9.4.4.2 PCIe Reset Conditions....5839
24.9.4.5 PCIe Controller Power Management....5842
24.9.4.5.1 PCIe Protocol Power Management....5842
24.9.4.5.2 PCIE Controller Clocks Management....5845
24.9.4.6 PCIe Controller Interrupt Requests....5847
24.9.4.6.1 PCIe Controller Main Hardware Management....5848
24.9.4.6.2 PCIe Controller Legacy and MSI Virtual Interrupts Management....5850
24.9.4.6.3 PCIe Controller MSI Hardware Interrupt Events....5853
24.9.4.7 PCIe Controller Address Spaces and Address Translation....5854
24.9.4.8 PCIe Traffic Requesting and Responding....5857
24.9.4.8.1 PCIe Memory-type (Mem) Traffic Management....5857
24.9.4.8.2 PCIe Configuration Type (Cfg) Traffic Management....5858
24.9.4.8.3 PCIe I/O-type (IO) traffic management....5859
24.9.4.8.4 PCIe Message-type (Msg) traffic management....5860
24.9.4.9 PCIe Programming Register Interface....5861
24.9.4.9.1 PCIe Register Access....5861
24.9.4.9.2 Double Mapping of the PCIe Local Control Registers....5861
24.9.4.9.3 Base Address Registers (BAR) Initialization....5862
24.9.5 PCIe Controller Low Level Programming Model....5864
24.9.5.1 Surrounding Modules Global Initialization....5864
24.9.5.2 Main Sequence of PCIe Controllers Initalization....5864
24.9.6 PCIe Standard Registers vs PCIe Subsystem Hardware Registers Mapping....5868
24.9.7 PCIe Controller Register Manual....5870

24.9.7.1 PCIe Controller Instance Summary.......... 5870
24.9.7.2 PCIe_SS_EP_CFG_PCIe Registers.......... 5870
24.9.7.2.1 PCIe_SS_EP_CFG_PCIe Register Summary.......... 5870
24.9.7.2.2 PCIe_SS_EP_CFG_PCIe Register Description.......... 5873
24.9.7.3 PCIe_SS_EP_CFG_DBICS Registers.......... 5889
24.9.7.3.1 PCIe_SS_EP_CFG_DBICS Register Summary.......... 5889
24.9.7.3.2 PCIe_SS_EP_CFG_DBICS Register Description.......... 5891
24.9.7.4 PCIe_SS_RC_CFG_DBICS Registers.......... 5908
24.9.7.4.1 PCIe_SS_RC_CFG_DBICS Register Summary.......... 5908
24.9.7.4.2 PCIe_SS_RC_CFG_DBICS Register Description.......... 5909
24.9.7.5 PCIe_SS_PL_CONF Registers.......... 5926
24.9.7.5.1 PCIe_SS_PL_CONF Register Summary.......... 5926
24.9.7.5.2 PCIe_SS_PL_CONF Register Description.......... 5928
24.9.7.6 PCIe_SS_EP_CFG_DBICS2 Registers.......... 5946
24.9.7.6.1 PCIe_SS_EP_CFG_DBICS2 Register Summary.......... 5946
24.9.7.6.2 PCIe_SS_EP_CFG_DBICS2 Register Description.......... 5948
24.9.7.7 PCIe_SS_RC_CFG_DBICS2 Registers.......... 5962
24.9.7.7.1 PCIe_SS_RC_CFG_DBICS2 Register Summary.......... 5962
24.9.7.7.2 PCIe_SS_RC_CFG_DBICS2 Register Description.......... 5963
24.9.7.8 PCIe_SS_TI_CONF Registers.......... 5979
24.9.7.8.1 PCIe_SS_TI_CONF Register Summary.......... 5979
24.9.7.8.2 PCIe_SS_TI_CONF Register Description.......... 5980
24.10 DCAN.......... 5998
24.10.1 DCAN Overview.......... 5998
24.10.1.1 Features.......... 5999
24.10.2 DCAN Environment.......... 6000
24.10.2.1 CAN Network Basics.......... 6000
24.10.3 DCAN Integration.......... 6002
24.10.4 DCAN Functional Description.......... 6006
24.10.4.1 Module Clocking Requirements.......... 6007
24.10.4.2 Interrupt Functionality.......... 6007
24.10.4.2.1 Message Object Interrupts.......... 6007
24.10.4.2.2 Status Change Interrupts.......... 6008
24.10.4.2.3 Error Interrupts.......... 6008
24.10.4.3 DMA Functionality.......... 6009
24.10.4.4 Local Power-Down Mode.......... 6009
24.10.4.4.1 Entering Local Power-Down Mode.......... 6009
24.10.4.4.2 Wakeup From Local Power Down.......... 6010
24.10.4.5 Parity Check Mechanism.......... 6011
24.10.4.5.1 Behavior on Parity Error.......... 6012
24.10.4.5.2 Parity Testing.......... 6012
24.10.4.6 Debug/Suspend Mode.......... 6012
24.10.4.7 Configuration of Message Objects Description.......... 6013
24.10.4.7.1 Configuration of a Transmit Object for Data Frames.......... 6013
24.10.4.7.2 Configuration of a Transmit Object for Remote Frames.......... 6014
24.10.4.7.3 Configuration of a Single Receive Object for Data Frames.......... 6014
24.10.4.7.4 Configuration of a Single Receive Object for Remote Frames.......... 6014
24.10.4.7.5 Configuration of a FIFO Buffer.......... 6015
24.10.4.8 Message Handling.......... 6015
24.10.4.8.1 Message Handler Overview.......... 6015
24.10.4.8.2 Receive/Transmit Priority.......... 6015
24.10.4.8.3 Transmission of Messages in Event Driven CAN Communication.......... 6016
24.10.4.8.4 Updating a Transmit Object.......... 6016
24.10.4.8.5 Changing a Transmit Object.......... 6017
24.10.4.8.6 Acceptance Filtering of Received Messages.......... 6017
24.10.4.8.7 Reception of Data Frames.......... 6017
24.10.4.8.8 Reception of Remote Frames.......... 6017
24.10.4.8.9 Reading Received Messages.......... 6018
24.10.4.8.10 Requesting New Data for a Receive Object.......... 6018
24.10.4.8.11 Storing Received Messages in FIFO Buffers.......... 6018
24.10.4.8.12 Reading From a FIFO Buffer.......... 6018

24.10.4.9 CAN Bit Timing....6020
24.10.4.9.1 Bit Time and Bit Rate....6021
24.10.4.9.2 DCAN Bit Timing Registers....6026
24.10.4.10 Message Interface Register Sets....6028
24.10.4.10.1 Message Interface Register Sets 1 and 2....6028
24.10.4.10.2 IF3 Register Set....6029
24.10.4.11 Message RAM....6030
24.10.4.11.1 Structure of Message Objects....6030
24.10.4.11.2 Addressing Message Objects in RAM....6032
24.10.4.11.3 Message RAM Representation in Debug/Suspend Mode....6033
24.10.4.11.4 Message RAM Representation in Direct Access Mode....6034
24.10.4.12 CAN Operation....6035
24.10.4.12.1 CAN Module Initialization....6035
24.10.4.12.2 CAN Message Transfer (Normal Operation)....6037
24.10.4.12.3 Test Modes....6038
24.10.4.13 GPIO Support....6041
24.10.5 DCAN Register Manual....6042
24.10.5.1 DCAN Instance Summary....6042
24.10.5.2 DCAN Registers....6042
24.10.5.2.1 DCAN Register Summary....6042
24.10.5.2.2 DCAN Register Description....6043
24.11 Gigabit Ethernet Switch (GMAC_SW)....6093
24.11.1 GMAC_SW Overview....6093
24.11.1.1 Features....6093
24.11.2 GMAC_SW Environment....6096
24.11.2.1 G/MII Interface....6096
24.11.2.2 RMII Interface....6097
24.11.2.3 RGMII Interface....6098
24.11.3 GMAC_SW Integration....6100
24.11.4 GMAC_SW Functional Description....6103
24.11.4.1 Functional Block Diagram....6103
24.11.4.2 GMAC_SW Ports....6103
24.11.4.2.1 Interface Mode Selection....6104
24.11.4.3 Clocking....6104
24.11.4.3.1 Subsystem Clocking....6104
24.11.4.3.2 Interface Clocking....6104
24.11.4.4 Software IDLE....6104
24.11.4.5 Interrupt Functionality....6105
24.11.4.5.1 Receive Packet Completion Pulse Interrupt (RX_PULSE)....6105
24.11.4.5.2 Transmit Packet Completion Pulse Interrupt (TX_PULSE)....6105
24.11.4.5.3 Receive Threshold Pulse Interrupt (RX_THRESH_PULSE)....6106
24.11.4.5.4 Miscellaneous Pulse Interrupt (MISC_PULSE)....6107
24.11.4.5.5 Interrupt Pacing....6108
24.11.4.6 Reset Isolation....6109
24.11.4.6.1 Reset Isolation Functional Description....6109
24.11.4.7 Software Reset....6109
24.11.4.8 CPSW_3G....6110
24.11.4.8.1 CPDMA RX and TX Interfaces....6110
24.11.4.8.2 Address Lookup Engine (ALE)....6112
24.11.4.8.3 Packet Priority Handling....6119
24.11.4.8.4 FIFO Memory Control....6119
24.11.4.8.5 FIFO Transmit Queue Control....6119
24.11.4.8.6 Audio Video Bridging....6121
24.11.4.8.7 Ethernet MAC Sliver (CPGMAC_SL)....6127
24.11.4.8.8 Embedded Memories....6131
24.11.4.8.9 Flow Control....6131
24.11.4.8.10 Short Gap....6133
24.11.4.8.11 Switch Latency....6133
24.11.4.8.12 Emulation Control....6134
24.11.4.8.13 FIFO Loopback....6134
24.11.4.8.14 Device Level Ring (DLR) Support....6134

24.11.4.8.15 Energy Efficient Ethernet Support (802.3az)........ 6135
24.11.4.8.16 CPSW_3G Network Statistics........ 6136
24.11.4.9 Static Packet Filter (SPF)........ 6139
24.11.4.9.1 SPF Overview........ 6139
24.11.4.9.2 SPF Functional Description........ 6139
24.11.4.9.3 Programming Guide........ 6154
24.11.4.10 Common Platform Time Sync (CPTS)........ 6155
24.11.4.10.1 CPTS Architecture........ 6155
24.11.4.10.2 CPTS Initialization........ 6156
24.11.4.10.3 Time Stamp Value........ 6156
24.11.4.10.4 Event FIFO........ 6156
24.11.4.10.5 Time Sync Events........ 6156
24.11.4.10.6 CPTS Interrupt Handling........ 6160
24.11.4.11 CPPI Buffer Descriptors........ 6160
24.11.4.11.1 TX Buffer Descriptors........ 6161
24.11.4.11.2 RX Buffer Descriptors........ 6163
24.11.4.12 MDIO........ 6166
24.11.4.12.1 MDIO Frame Formats........ 6166
24.11.4.12.2 MDIO Functional Description........ 6167
24.11.5 GMAC_SW Programming Guide........ 6167
24.11.5.1 Transmit Operation........ 6167
24.11.5.2 Receive Operation........ 6169
24.11.5.3 MDIO Software Interface........ 6171
24.11.5.3.1 Initializing the MDIO Module........ 6171
24.11.5.3.2 Writing Data To a PHY Register........ 6171
24.11.5.3.3 Reading Data From a PHY Register........ 6172
24.11.5.4 Initialization and Configuration of CPSW........ 6172
24.11.6 GMAC_SW Register Manual........ 6173
24.11.6.1 GMAC_SW Instance Summary........ 6173
24.11.6.2 SS Registers........ 6173
24.11.6.2.1 SS Register Summary........ 6173
24.11.6.2.2 SS Register Description........ 6173
24.11.6.3 PORT Registers........ 6179
24.11.6.3.1 PORT Register Summary........ 6179
24.11.6.3.2 PORT Register Description........ 6181
24.11.6.4 CPDMA registers........ 6222
24.11.6.4.1 CPDMA Register Summary........ 6222
24.11.6.4.2 CPDMA Register Description........ 6223
24.11.6.5 STATS Registers........ 6249
24.11.6.5.1 STATS Register Summary........ 6249
24.11.6.5.2 STATS Register Description........ 6250
24.11.6.6 STATERAM Registers........ 6264
24.11.6.6.1 STATERAM Register Summary........ 6264
24.11.6.6.2 STATERAM Register Description........ 6265
24.11.6.7 CPTS registers........ 6278
24.11.6.7.1 CPTS Register Summary........ 6278
24.11.6.7.2 CPTS Register Description........ 6279
24.11.6.8 ALE registers........ 6284
24.11.6.8.1 ALE Register Summary........ 6284
24.11.6.8.2 ALE Register Description........ 6285
24.11.6.9 SL registers........ 6294
24.11.6.9.1 SL Register Summary........ 6294
24.11.6.9.2 SL Register Description........ 6295
24.11.6.10 MDIO registers........ 6303
24.11.6.10.1 MDIO Register Summary........ 6303
24.11.6.10.2 MDIO Register Description........ 6303
24.11.6.11 WR registers........ 6312
24.11.6.11.1 WR Register Summary........ 6312
24.11.6.11.2 WR Register Description........ 6312
24.11.6.12 SPF Registers........ 6320
24.11.6.12.1 SPF Register Summary........ 6320

24.11.6.12.2 SPF Register Description....6321
24.12 Media Local Bus (MLB)....6333
24.12.1 MLB Overview....6333
24.12.2 MLB Environment....6333
24.12.2.1 MLB IO Cell Controls....6334
24.12.3 MLB Integration....6337
24.12.4 MLB Functional Description....6339
24.12.4.1 Block Diagram....6339
24.12.4.1.1 MediaLB Core Block....6340
24.12.4.1.2 Routing Fabric Block....6341
24.12.4.1.3 Data Buffer RAM....6341
24.12.4.1.4 Channel Table RAM....6341
24.12.4.1.5 DMA Block....6348
24.12.4.2 Software and Data Flow for MLBSS....6353
24.12.4.2.1 Data Flow For Receive Channels....6354
24.12.4.2.2 Data Flow for Transmit Channels....6355
24.12.4.3 MLB Priority On The L3_MAIN Interconnect....6355
24.12.5 MLB Programming Guide....6355
24.12.5.1 Global Initialization....6356
24.12.5.1.1 Surrounding Modules Global Initialization....6356
24.12.5.1.2 MLBSS Global Initialization....6356
24.12.5.2 MLBSS Operational Modes Configuration....6361
24.12.5.2.1 Channel Servicing....6361
24.12.5.2.2 Channel Table RAM Access....6363
24.12.6 MLB Register Manual....6363
24.12.6.1 MLB Instance Summary....6363
24.12.6.2 MLB registers....6364
24.12.6.2.1 MLB Register Summary....6364
24.12.6.2.2 MLB Register Description....6364
25 eMMC/SD/SDIO....6380
25.1 eMMC/SD/SDIO Overview....6381
25.1.1 eMMC/SD/SDIO Features....6382
25.2 eMMC/SD/SDIO Environment....6385
25.2.1 eMMC/SD/SDIO Functional Modes....6385
25.2.1.1 eMMC/SD/SDIO Connected to an eMMC, SD, or SDIO Card....6385
25.2.2 Protocol and Data Format....6386
25.2.2.1 Protocol....6387
25.2.2.2 Data Format....6389
25.3 eMMC/SD/SDIO Integration....6392
25.4 eMMC/SD/SDIO Functional Description....6398
25.4.1 Block Diagram....6398
25.4.2 Resets....6398
25.4.2.1 Hardware Reset....6398
25.4.2.2 Software Reset....6398
25.4.3 Power Management....6399
25.4.4 Interrupt Requests....6403
25.4.4.1 Interrupt-Driven Operation....6406
25.4.4.2 Polling....6406
25.4.4.3 Asynchronous Interrupt....6407
25.4.5 DMA Modes....6407
25.4.5.1 Master DMA Operations....6407
25.4.5.1.1 Descriptor Table Description....6408
25.4.5.1.2 Requirements for Descriptors....6409
25.4.5.1.3 Advanced DMA Description....6410
25.4.5.2 Slave DMA Operations....6412
25.4.5.2.1 DMA Receive Mode....6412
25.4.5.2.2 DMA Transmit Mode....6413
25.4.6 Mode Selection....6414
25.4.7 Buffer Management....6414
25.4.7.1 Data Buffer....6414
25.4.7.1.1 Memory Size, Block Length, and Buffer-Management Relationship....6417

25.4.7.1.2 Data Buffer Status....6418
25.4.8 Transfer Process....6418
25.4.8.1 Different Types of Commands....6418
25.4.8.2 Different Types of Responses....6418
25.4.9 Transfer or Command Status and Errors Reporting....6419
25.4.9.1 Busy Time-Out for R1b, R5b Response Type....6419
25.4.9.2 Busy Time-Out After Write CRC Status....6420
25.4.9.3 Write CRC Status Time-Out....6420
25.4.9.4 Read Data Time-Out....6421
25.4.9.5 Boot Acknowledge Time-Out....6422
25.4.10 Auto Command 12 Timings....6423
25.4.10.1 Auto CMD12 Timings During Write Transfer....6423
25.4.10.2 Auto CMD12 Timings During Read Transfer....6423
25.4.11 Transfer Stop....6423
25.4.12 Output Signals Generation....6425
25.4.12.1 Generation on Falling Edge of MMC Clock....6425
25.4.12.2 Generation on Rising Edge of MMC Clock....6425
25.4.13 Sampling Clock Tuning....6426
25.4.14 Card Boot Mode Management....6426
25.4.14.1 Boot Mode Using CMD0....6427
25.4.14.2 Boot Mode With CMD Line Tied to 0....6427
25.4.15 MMC CE-ATA Command Completion Disable Management....6427
25.4.16 Test Registers....6428
25.4.17 eMMC/SD/SDIO Hardware Status Features....6428
25.5 eMMC/SD/SDIO Programming Guide....6430
25.5.1 Low-Level Programming Models....6430
25.5.1.1 Global Initialization....6430
25.5.1.1.1 Surrounding Modules Global Initialization....6430
25.5.1.1.2 eMMC/SD/SDIO Host Controller Initialization Flow....6430
25.5.1.2 Operational Modes Configuration....6433
25.5.1.2.1 Basic Operations for eMMC/SD/SDIO Host Controller....6433
25.5.1.2.2 Bus Voltage Selection....6449
25.5.1.2.3 Boot Mode Configuration....6450
25.5.1.2.4 SDR104/HS200 DLL Tuning Procedure....6453
25.6 eMMC/SD/SDIO Register Manual....6455
25.6.1 eMMC/SD/SDIO Instance Summary....6455
25.6.2 eMMC/SD/SDIO Registers....6455
25.6.2.1 eMMC/SD/SDIO Register Summary....6455
25.6.2.2 eMMC/SD/SDIO Register Description....6456
26 Shared PHY Component Subsystem....6519
26.1 SATA PHY Subsystem....6520
26.1.1 SATA PHY Subsystem Overview....6520
26.1.2 SATA PHY Subsystem Environment....6522
26.1.2.1 SATA PHY I/O Signals....6522
26.1.3 SATA PHY Subsystem Integration....6524
26.1.4 SATA PHY Subsystem Functional Description....6527
26.1.4.1 SATA PLL Controller L4 Interface Adapter Functional Description....6527
26.1.4.2 SATA PHY Serializer and Deserializer Functional Descriptions....6527
26.1.4.2.1 SATA PHY Reset....6527
26.1.4.2.2 SATA_PHY Clocking....6527
26.1.4.2.3 SATA_PHY Power Management....6528
26.1.4.2.4 SATA_PHY Hardware Requests....6528
26.1.4.3 SATA Clock Generator Subsystem Functional Description....6528
26.1.4.3.1 SATA DPLL Clock Generator Overview....6528
26.1.4.3.2 SATA DPLL Clock Generator Reset....6530
26.1.4.3.3 SATA DPLL Low-Power Modes....6530
26.1.4.3.4 SATA DPLL Clocks Configuration....6530
26.1.4.3.5 SATA DPLL Subsystem Architecture....6531
26.1.4.3.6 SATA DPLL Clock Generator Modes and State Transitions....6532
26.1.4.3.7 SATA PLL Controller Functions....6534
26.1.5 SATA PHY Subsystem Low-Level Programming Model....6537

26.2 USB3_PHY Subsystem....6539
26.2.1 USB3_PHY Subsystem Overview....6539
26.2.2 USB3_PHY Subsystem Environment....6541
26.2.2.1 USB3_PHY I/O Signals....6541
26.2.3 USB3_PHY Subsystem Integration....6542
26.2.4 USB3_PHY Subsystem Functional Description....6545
26.2.4.1 Super-Speed USB PLL Controller L4 Interface Adapter Functional Description....6545
26.2.4.2 USB3_PHY Serializer and Deserializer Functional Descriptions....6545
26.2.4.2.1 USB3_PHY Module Resets....6545
26.2.4.2.2 USB3_PHY Subsystem Clocking....6545
26.2.4.2.3 USB3_PHY Power Management....6546
26.2.4.2.4 USB3_PHY Hardware Requests....6547
26.2.4.3 USB3_PHY Clock Generator Subsystem Functional Description....6547
26.2.4.3.1 USB3_PHY DPLL Clock Generator Overview....6547
26.2.4.3.2 USB3_PHY DPLL Clock Generator Reset....6548
26.2.4.3.3 USB3_PHY DPLL Low-Power Modes....6548
26.2.4.3.4 USB3_PHY DPLL Clocks Configuration....6549
26.2.4.3.5 USB3_PHY DPLL Subsystem Architecture....6549
26.2.4.3.6 USB3_PHY DPLL Clock Generator Modes and State Transitions....6551
26.2.4.3.7 USB3_PHY PLL Controller Functions....6554
26.2.5 USB3_PHY Subsystem Low-Level Programming Model....6556
26.3 USB3 PHY and SATA PHY Register Manual....6558
26.3.1 USB3 PHY and SATA PHY Instance Summary....6558
26.3.2 USB3_PHY_RX Registers....6558
26.3.2.1 USB3_PHY_RX Register Summary....6558
26.3.2.2 USB3_PHY_RX Register Description....6559
26.3.3 USB3_PHY_TX Registers....6563
26.3.3.1 USB3_PHY_TX Register Summary....6563
26.3.3.2 USB3_PHY_TX Register Description....6563
26.3.4 SATA_PHY_RX Registers....6565
26.3.4.1 SATA_PHY_RX Register Summary....6565
26.3.4.2 SATA_PHY_RX Register Description....6565
26.3.5 SATA_PHY_TX Registers....6570
26.3.5.1 SATA_PHY_TX Register Summary....6570
26.3.5.2 SATA_PHY_TX Register Description....6570
26.3.6 DPLLCTRL Registers....6572
26.3.6.1 DPLLCTRL Register Summary....6572
26.3.6.2 DPLLCTRL Register Description....6572
26.4 PCIe PHY Subsystem....6577
26.4.1 PCIe PHY Subsystem Overview....6577
26.4.1.1 PCIe PHY Subsystem Key Features....6578
26.4.2 PCIe PHY Subsystem Environment....6580
26.4.2.1 PCIe PHY I/O Signals....6580
26.4.3 PCIe Shared PHY Subsystem Integration....6582
26.4.4 PCIe PHY Subsystem Functional Description....6585
26.4.4.1 PCIe PHY Subsystem Block Diagram....6585
26.4.4.2 OCP2SCP Functional Description....6586
26.4.4.2.1 OCP2SCP Reset....6586
26.4.4.2.2 OCP2SCP Power Management....6586
26.4.4.2.3 OCP2SCP Timing Registers....6586
26.4.4.3 PCIe PHY Serializer and Deserializer Functional Descriptions....6587
26.4.4.3.1 PCIe PHY Module Resets....6587
26.4.4.3.2 PCIe PHY Subsystem Clocking....6587
26.4.4.3.3 PCIe PHY Power Management....6589
26.4.4.3.4 PCIe PHY Hardware Requests....6589
26.4.4.4 PCIe PHY Clock Generator Subsystem Functional Description....6590
26.4.4.4.1 PCIe PHY DPLL Clock Generator....6591
26.4.4.4.2 PCIe PHY APLL Clock Generator....6597
26.4.4.4.3 ACSPCIE reference clock buffer....6600
26.4.5 PCIePHY Subsystem Low-Level Programming Model....6600
26.4.6 PCIe PHY Subsystem Register Manual....6603

26.4.6.1 PCIe PHY Instance Summary....6603
26.4.6.1.1 PCIe_PHY_RX Registers....6603
26.4.6.1.2 PCIe_PHY_TX Registers....6609
26.4.6.1.3 OCP2SCP Registers....6612
27 General-Purpose Interface....6614
27.1 General-Purpose Interface Overview....6615
27.2 General-Purpose Interface Environment....6618
27.2.1 General-Purpose Interface as a Keyboard Interface....6618
27.2.2 General-Purpose Interface Signals....6619
27.3 General-Purpose Interface Integration....6623
27.4 General-Purpose Interface Functional Description....6628
27.4.1 General-Purpose Interface Block Diagram....6628
27.4.2 General-Purpose Interface Interrupt and Wake-Up Features....6629
27.4.2.1 Synchronous Path: Interrupt Request Generation....6629
27.4.2.2 Asynchronous Path: Wake-Up Request Generation....6630
27.4.2.3 Wake-Up Event Conditions During Transition To/From IDLE State....6631
27.4.2.4 Interrupt (or Wake-Up) Line Release....6632
27.4.3 General-Purpose Interface Clock Configuration....6633
27.4.3.1 Clocking....6633
27.4.4 General-Purpose Interface Hardware and Software Reset....6633
27.4.5 General-Purpose Interface Power Management....6634
27.4.5.1 Power Domain....6634
27.4.5.2 Power Management....6634
27.4.5.2.1 Idle Scheme....6634
27.4.5.2.2 Operating Modes....6634
27.4.5.2.3 System Power Management and Wakeup....6635
27.4.5.2.4 Module Power Saving....6635
27.4.6 General-Purpose Interface Interrupt and Wake-Up Requests....6637
27.4.6.1 Interrupt Requests Generation....6637
27.4.6.2 Wake-Up Requests Generation....6638
27.4.7 General-Purpose Interface Channels Description....6639
27.4.8 General-Purpose Interface Data Input/Output Capabilities....6640
27.4.9 General-Purpose Interface Set-and-Clear Protocol....6640
27.4.9.1 Description....6640
27.4.9.2 Clear Instruction....6641
27.4.9.2.1 Clear Register Addresses....6641
27.4.9.2.2 Clear Instruction Example....6641
27.4.9.3 Set Instruction....6641
27.4.9.3.1 Set Register Addresses....6641
27.4.9.3.2 Set Instruction Example....6642
27.5 General-Purpose Interface Programming Guide....6643
27.5.1 General-Purpose Interface Low-Level Programming Models....6643
27.5.1.1 Global Initialization....6643
27.5.1.1.1 Surrounding Modules Global Initialization....6643
27.5.1.1.2 General-Purpose Interface Module Global Initialization....6643
27.5.1.2 General-Purpose Interface Operational Modes Configuration....6644
27.5.1.2.1 General-Purpose Interface Read Input Register....6644
27.5.1.2.2 General-Purpose Interface Set Bit Function....6644
27.5.1.2.3 General-Purpose Interface Clear Bit Function....6644
27.6 General-Purpose Interface Register Manual....6645
27.6.1 General-Purpose Interface Instance Summary....6645
27.6.2 General-Purpose Interface Registers....6645
27.6.2.1 General-Purpose Interface Register Summary....6645
27.6.2.2 General-Purpose Interface Register Description....6648
28 Keyboard Controller....6660
28.1 Keyboard Controller Overview....6661
28.2 Keyboard Controller Environment....6663
28.2.1 Keyboard Controller Functions/Modes....6664
28.2.2 Keyboard Controller Signals....6664
28.2.3 Protocols and Data Formats....6664
28.3 Keyboard Controller Integration....6666

28.4 Keyboard Controller Functional Description....6668
28.4.1 Keyboard Controller Block Diagram....6668
28.4.2 Keyboard Controller Software Reset....6669
28.4.3 Keyboard Controller Power Management....6669
28.4.4 Keyboard Controller Interrupt Requests....6669
28.4.5 Keyboard Controller Software Mode....6669
28.4.6 Keyboard Controller Hardware Decoding Modes....6669
28.4.6.1 Functional Modes....6669
28.4.6.2 Keyboard Controller Timer....6670
28.4.6.3 State-Machine Status....6672
28.4.6.4 Keyboard Controller Interrupt Generation....6672
28.4.6.4.1 Interrupt-Generation Scheme....6672
28.4.6.4.2 Keyboard Buffer and Missed Events (Overrun Feature)....6674
28.4.7 Keyboard Controller Key Coding Registers....6674
28.4.8 Keyboard Controller Register Access....6675
28.4.8.1 Write Registers Access....6675
28.4.8.2 Read Registers Access....6676
28.5 Keyboard Controller Programming Guide....6677
28.5.1 Keyboard Controller Low-Level Programming Models....6677
28.5.1.1 Global Initialization....6677
28.5.1.1.1 Surrounding Modules Global Initialization....6677
28.5.1.1.2 Keyboard Controller Global Initialization....6677
28.5.1.2 Operational Modes Configuration....6678
28.5.1.2.1 Keyboard Controller in Hardware Decoding Mode (Default Mode)....6678
28.5.1.2.2 Keyboard Controller Software Scanning Mode....6679
28.5.1.2.3 Using the Timer....6679
28.5.1.2.4 State-Machine Status Register....6679
28.5.1.3 Keyboard Controller Events Servicing....6679
28.6 Keyboard Controller Register Manual....6681
28.6.1 Keyboard Controller Instance Summary....6681
28.6.2 Keyboard Controller Registers....6681
28.6.2.1 Keyboard Controller Register Summary....6681
28.6.2.2 Keyboard Controller Register Description....6681
29 Pulse-Width Modulation Subsystem....6692
29.1 PWM Subsystem Resources....6693
29.1.1 PWMSS Overview....6693
29.1.1.1 PWMSS Key Features....6693
29.1.1.2 PWMSS Unsupported Fetaures....6694
29.1.2 PWMSS Environment....6694
29.1.2.1 PWMSS I/O Interface....6694
29.1.3 PWMSS Integration....6697
29.1.3.1 PWMSS Module Interfaces Implementation....6700
29.1.3.1.1 Device Specific PWMSS Features....6700
29.1.3.1.2 Daisy-Chain Connectivity between PWMSS Modules....6701
29.1.3.1.3 eHRPWM Modules Time Base Clock Gating....6702
29.1.4 PWMSS Subsystem Power, Reset and Clock Configuration....6703
29.1.4.1 PWMSS Local Clock Management....6703
29.1.4.2 PWMSS Modules Local Clock Gating....6703
29.1.4.3 PWMSS Software Reset....6704
29.1.5 PWMSS_CFG Register Manual....6704
29.1.5.1 PWMSS_CFG Instance Summary....6704
29.1.5.2 PWMSS_CFG Registers....6704
29.1.5.2.1 PWMSS_CFG Register Summary....6704
29.1.5.2.2 PWMSS_CFG Register Description....6704
29.2 Enhanced PWM (ePWM) Module....6708
29.3 Enhanced Capture (eCAP) Module....6802
29.4 Enhanced Quadrature Encoder Pulse (eQEP) Module....6824
30 Viterbi-Decoder Coprocessor....6866
30.1 VCP Overview....6867
30.1.1 VCP Features....6867
30.2 VCP Integration....6869

30.3 VCP Functional Description....6871
30.3.1 VCP Block Diagram....6871
30.3.2 VCP Internal Interfaces....6872
30.3.2.1 VCP Power Management....6872
30.3.2.1.1 Idle Mode....6872
30.3.2.2 VCP Clocks....6873
30.3.2.3 VCP Resets....6873
30.3.2.4 Interrupt Requests....6873
30.3.2.5 EDMA Requests....6873
30.3.3 Functional Overview....6874
30.3.3.1 Theoretical Basics of the Convolutional Code....6874
30.3.4 VCP Architecture....6875
30.3.4.1 Sliding Windows Processing....6876
30.3.4.1.1 Tailed Traceback Mode....6876
30.3.4.1.2 Mixed Traceback Mode....6876
30.3.4.1.3 Convergent Traceback Mode....6877
30.3.4.1.4 F, R, and C Limitations....6877
30.3.4.1.5 Yamamoto Parameters....6878
30.3.4.1.6 Input FIFO (Branch Metrics)....6879
30.3.4.1.7 Output FIFO (Decisions)....6880
30.3.5 VCP Input Data....6880
30.3.5.1 Branch Metrics Calculations....6880
30.3.6 Soft Input Dynamic Ranges....6882
30.3.7 VCP Memory Sleep Mode....6882
30.3.8 Decision Data....6882
30.3.9 Endianness....6883
30.3.9.1 Branch Metrics....6883
30.3.9.1.1 Hard Decisions....6886
30.3.9.1.2 Soft Decisions....6886
30.3.10 VCP Output Parameters....6886
30.3.11 Event Generation....6886
30.3.11.1 VCPnXEVT Generation....6886
30.3.11.2 VCPnREVT Generation....6887
30.3.12 Operational Modes....6887
30.3.12.1 Debugging Features....6887
30.3.13 Errors and Status....6888
30.4 VCP Modules Programming Guide....6890
30.4.1 EDMA Resources....6890
30.4.1.1 VCP1 and VCP2 Dedicated EDMA Resources....6890
30.4.1.2 Special VCP EDMA Programming Considerations....6890
30.4.1.2.1 Input Configuration Parameters Transfer....6891
30.4.1.2.2 Branch Metrics Transfer....6892
30.4.1.2.3 Decisions Transfer....6893
30.4.1.2.4 Hard-Decisions Mode....6893
30.4.1.2.5 Soft-Decisions Mode....6894
30.4.1.2.6 Output Parameters Transfer....6895
30.4.2 Input Configuration Words....6896
30.5 VCP Register Manual....6897
30.5.1 VCP1 and VCP2 Instance Summary....6897
30.5.2 VCP Registers....6897
30.5.2.1 VCP Register Summary....6897
30.5.2.2 VCP1 and VCP2 Data Registers Description....6898
30.5.2.3 VCP1 and VCP2 Configuration Registers Description....6904
31 Audio Tracking Logic....6913
31.1 ATL Overview....6914
31.2 ATL Environment....6915
31.2.1 ATL Functions....6915
31.2.2 ATL Signals Descriptions....6915
31.3 ATL Integration....6916
31.3.1 ATL Distribution on Interconnects....6917
31.3.2 ATL Regions Allocations....6917

31.4 ATL Functional Description....6918
31.4.1 Block Diagram....6918
31.4.2 Source Signal Control....6919
31.4.3 ATL Clock and Reset Configuration....6919
31.5 ATL Register Manual....6921
31.5.1 ATL Instance Summary....6921
31.5.2 ATL Register Summary....6921
31.5.3 ATL Register Description....6922
32 Initialization....6936
32.1 Initialization Overview....6937
32.1.1 Terminology....6937
32.1.2 Initialization Process....6937
32.2 Preinitialization....6939
32.2.1 Power Requirements....6939
32.2.2 Interaction With the PMIC Companion....6942
32.2.3 Clock, Reset, and Control....6943
32.2.3.1 Overview....6943
32.2.3.2 Clocking Scheme....6944
32.2.3.3 Reset Configuration....6945
32.2.3.3.1 ON/OFF Interconnect and Power-On-Reset....6945
32.2.3.3.2 Warm Reset....6945
32.2.3.3.3 Peripheral Reset by GPIO....6945
32.2.3.3.4 Warm Reset Impact on GPIOs....6945
32.2.3.4 PMIC Control....6946
32.2.3.5 PMIC Request Signals....6946
32.2.4 Sysboot Configuration....6947
32.2.4.1 GPMC Configuration for XIP/NAND....6947
32.2.4.2 System Clock Speed Selection....6948
32.2.4.3 QSPI Redundant SBL Images Offset....6948
32.2.4.4 Booting Device Order Selection....6948
32.2.4.6 Boot Peripheral Pin Multiplexing....6950
32.3 Device Initialization by ROM Code....6951
32.3.1 Booting Overview....6951
32.3.1.1 Booting Types....6951
32.3.1.2 ROM Code Architecture....6951
32.3.2 Memory Maps....6952
32.3.2.1 ROM Memory Map....6952
32.3.2.2 RAM Memory Map....6954
32.3.3 Overall Booting Sequence....6957
32.3.4 Startup and Configuration....6959
32.3.4.1 Startup....6959
32.3.4.2 Control Module Configuration....6960
32.3.4.3 PRCM Module Mode Configuration....6960
32.3.4.4 Clocking Configuration....6961
32.3.4.5 Booting Device List Setup....6962
32.3.5 Peripheral Booting....6963
32.3.5.1 Description....6963
32.3.5.2 Initialization Phase for UART Boot....6968
32.3.5.3 Initialization Phase for USB Boot....6968
32.3.5.3.1 Initialization Procedure....6968
32.3.5.3.2 SATA Peripheral Device Flashing over USB Interface....6969
32.3.5.3.3 USB Driver Descriptors....6970
32.3.5.3.5 USB Customized Vendor and Product IDs....6973
32.3.5.3.6 USB Driver Functionality....6974
32.3.6 Fast External Booting....6976
32.3.6.1 Overview....6976
32.3.6.2 Fast External Booting Procedure....6976
32.3.7 Memory Booting....6978
32.3.7.1 Overview....6978
32.3.7.2 Non-XIP Memory....6979
32.3.7.3 XIP Memory....6980

32.3.7.3.1 GPMC Initialization....6981
32.3.7.4 NAND....6981
32.3.7.4.1 Initialization and NAND Detection....6982
32.3.7.4.2 NAND Read Sector Procedure....6986
32.3.7.5 SPI/QSPI Flash Devices....6988
32.3.7.6 eMMC Memories and SD Cards....6989
32.3.7.6.1 eMMC Memories....6989
32.3.7.6.2 SD Cards....6990
32.3.7.6.3 Initialization and Detection....6993
32.3.7.6.4 Read Sector Procedure....6996
32.3.7.6.5 File System Handling....6997
32.3.7.7 SATA Device Boot Operation....7003
32.3.7.7.1 SATA Booting Overview....7004
32.3.7.7.2 SATA Power-Up Initialization Sequence....7005
32.3.7.7.3 System Conditions and Limitations for SATA Boot....7007
32.3.7.7.4 SATA Read Sector Procedure in FAT Mode....7007
32.3.8 Image Format....7009
32.3.8.1 Overview....7009
32.3.8.2 Configuration Header....7009
32.3.8.2.1 CHSETTINGS Item....7011
32.3.8.2.2 CHFLASH Item....7012
32.3.8.2.3 CHMMCSD Item....7013
32.3.8.2.4 CHQSPI Item....7014
32.3.8.3 GP Header....7014
32.3.8.4 Image Execution....7015
32.3.9 Tracing....7017
32.4 Services for HLOS Support....7020
32.4.1 Hypervisor....7020
32.4.2 Caches Maintenance....7020
32.4.3 CP15 Registers....7020
32.4.4 Wakeup Generator....7021
32.4.5 Arm Timer....7022
33 On-Chip Debug Support....7023
33.1 Introduction....7024
33.1.1 Key Features....7025
33.2 Debug Interfaces....7028
33.2.1 IEEE1149.1....7028
33.2.2 Debug (Trace) Port....7028
33.2.3 Trace Connector and Board Layout Considerations....7029
33.3 Debugger Connection....7031
33.3.1 ICEPick Module....7031
33.3.2 ICEPick Boot Modes....7031
33.3.2.1 Default Boot Mode....7032
33.3.2.2 Wait-In-Reset....7032
33.3.3 Dynamic TAP Insertion....7032
33.3.3.1 ICEPick Secondary TAPs....7032
33.4 Primary Debug Support....7035
33.4.1 Processor Native Debug Support....7035
33.4.1.1 Cortex-A15 Processor....7035
33.4.1.2 Cortex-M4 Processor....7035
33.4.1.3 DSP C66x....7035
33.4.1.4 IVA Arm968....7035
33.4.1.5 ARP32....7036
33.4.2 Cross-Triggering....7036
33.4.2.1 SoC-Level Cross-Triggering....7037
33.4.2.2 Cross-Triggering With External Device....7038
33.4.3 Suspend....7038
33.4.3.1 Debug Aware Peripherals and Host Processors....7038
33.5 Real-Time Debug....7040
33.5.1 Real-Time Debug Events....7040
33.5.1.1 Emulation Interrupts....7040

33.6 Power, Reset, and Clock Management Debug Support....7041
33.6.1 Power and Clock Management....7041
33.6.1.1 Power and Clock Control Override From Debugger....7041
33.6.1.1.1 Debugger Directives....7041
33.6.1.1.2 Intrusive Debug Model....7041
33.6.1.2 Debug Across Power Transition....7041
33.6.1.2.1 Nonintrusive Debug Model....7041
33.6.1.2.2 Debug Context Save and Restore....7042
33.6.2 Reset Management....7042
33.6.2.1 Debugger Directives....7042
33.6.2.1.1 Assert Reset....7042
33.6.2.1.2 Block Reset....7042
33.6.2.1.3 Wait-In-Reset....7042
33.7 Performance Monitoring....7044
33.7.1 MPU Subsystem Performance Monitoring....7044
33.7.1.1 Performance Monitoring Unit....7044
33.7.1.2 L2 Cache Controller....7044
33.7.2 IPU Subsystem Performance Monitoring....7044
33.7.2.1 Subsystem Counter Timer Module....7044
33.7.2.2 Cache Events....7045
33.7.3 DSP Subsystem Performance Monitoring....7046
33.7.3.1 Advanced Event Triggering....7046
33.7.4 EVE Subsystem Performance Monitoring....7046
33.7.4.1 EVE Subsystem Counter Timer Module....7046
33.7.4.2 EVE Subsystem SCTM Events....7047
33.8 MPU Memory Adaptor (MPU_MA) Watchpoint....7048
33.9 Processor Trace....7049
33.9.1 Cortex-A15 Processor Trace....7049
33.9.2 DSP Processor Trace....7050
33.9.3 Trace Export....7050
33.9.3.1 Trace Exported to External Trace Receiver....7051
33.9.3.2 Trace Captured Into On-Chip Trace Buffer....7051
33.9.3.3 Trace Exported Through USB....7051
33.10 System Instrumentation....7053
33.10.1 MIPI STM (CT_STM)....7053
33.10.2 System Trace Export....7053
33.10.2.1 CT_STM ATB Export....7053
33.10.2.2 Trace Streams Interleaving....7054
33.10.3 Software Instrumentation....7054
33.10.3.1 MPU Software Instrumentation....7054
33.10.3.2 SoC Software Instrumentation....7055
33.10.4 OCP Watchpoint....7055
33.10.4.1 OCP Target Traffic Monitoring....7055
33.10.4.2 Messages Triggered from System Events....7058
33.10.4.3 DMA Transfer Profiling....7058
33.10.5 IVA Pipeline....7058
33.10.6 EVE SMSET....7058
33.10.7 L3 NOC Statistics Collector....7058
33.10.7.1 L3 Target Load Monitoring....7063
33.10.7.2 L3 Master Latency Monitoring....7064
33.10.7.2.1 SC_LAT0 Configuration....7064
33.10.7.2.2 SC_LAT1 Configuration....7065
33.10.7.2.3 SC_LAT2 Configuration....7066
33.10.7.2.4 SC_LAT3 Configuration....7067
33.10.7.2.5 SC_LAT4 Configuration....7068
33.10.7.2.6 SC_LAT5 Configuration....7068
33.10.7.2.7 SC_LAT6 Configuration....7069
33.10.7.2.8 SC_LAT7 Configuration....7070
33.10.7.2.9 SC_LAT8 Configuration....7071
33.10.7.2.10 Statistics Collector Alarm Mode....7072
33.10.7.2.11 Statistics Collector Suspend Mode....7072

33.10.8 PM Instrumentation....7072
33.10.9 CM Instrumentation....7073
33.10.10 Master-ID Encoding....7074
33.10.10.1 Software Masters....7074
33.10.10.2 Hardware Masters....7074
33.11 Concurrent Debug Modes....7076
33.12 DRM Register Manual....7077
33.12.1 DRM Instance Summary....7077
33.12.2 DRM Registers....7077
33.12.2.1 DRM Register Summary....7077
33.12.2.2 DRM Register Description....7078
34 Glossary....7109
35 Revision History....7110

List of Figures

Figure 1-1. DRA75x, DRA74x Sample Environment Diagram....259
Figure 1-2. DRA75x, DRA74x Block Diagram....260
Figure 2-1. Interconnect Overview....272
Figure 3-1. Clock Tree Tool (CTT)....297
Figure 3-2. Functional and Interface Clocks....299
Figure 3-3. Generic Clock Domain....305
Figure 3-4. Clock Domain State Transitions....306
Figure 3-5. Clock Domain/Slave Module Clock-Management Interaction Sequence 1....308
Figure 3-6. Clock Domain/Slave Module Clock-Management Interaction Sequence 2....309
Figure 3-7. Clock Domain/Slave Module Clock-Management Interaction Sequence 3....310
Figure 3-8. Sliding Window for Dynamic Dependency....315
Figure 3-9. Generic Power Domain....317
Figure 3-10. Power Domain Transitions....319
Figure 3-11. Generic Voltage Domain....320
Figure 3-12. Generic Logic Voltage Management....321
Figure 3-13. Generic Memory Voltage Management....321
Figure 3-14. SmartReflex Static Voltage Adjustment....322
Figure 3-15. SmartReflex Voltage Control Functional Overview....323
Figure 3-16. Comparison of Energy Consumed With/Without DVFS....324
Figure 3-17. Comparison of Energy Consumed With/Without DPS....325
Figure 3-18. Performance Level and Applied Power-Management Techniques....327
Figure 3-19. PMFW Overview....329
Figure 3-20. MPU Power-On Reset Sequence....368
Figure 3-21. MPU Warm Reset Sequence....369
Figure 3-22. MPU Reset Sequence on Sleep and Wake-Up Transition....370
Figure 3-23. IVA Power-On Reset Sequence....371
Figure 3-24. IVA Software Warm Reset Sequence....371
Figure 3-25. DSP1 Subsystem Power-On Reset Sequence....372
Figure 3-26. DSP1 Subsystem Software Warm Reset Sequence....373
Figure 3-27. DSP2 Subsystem Power-On Reset Sequence....373
Figure 3-28. DSP2 Subsystem Software Warm Reset Sequence....374
Figure 3-29. IPU1 Power-On Reset Sequence....375
Figure 3-30. IPU1 Subsystem Software Warm Reset Sequence....376
Figure 3-31. IPU2 Power-On Reset Sequence....377
Figure 3-32. IPU2 Subsystem Software Warm Reset Sequence....378
Figure 3-33. EVE1 Subsystem Power-On Reset Sequence....378
Figure 3-34. EVE1 Subsystem Software Warm Reset Sequence....379
Figure 3-35. EVE2 Subsystem Power-On Reset Sequence....380
Figure 3-36. EVE2 Subsystem Software Warm Reset Sequence....380
Figure 3-37. Global Warm Reset Sequence....381
Figure 3-38. PRCM Module Clock Manager Overview....383
Figure 3-39. PRM Clock Manager Overview....385
Figure 3-40. CM_CORE_AON Overview (a)....388
Figure 3-41. CM_CORE_AON Overview (b)....390
Figure 3-42. CM_CORE_AON_CLKOUTMUX Clock Manager Overview (CLKOUTMUX0)....393

Figure 3-43. CM_CORE_AON_CLKOUTMUX Clock Manager Overview (CLKOUTMUX1 and CLKOUTMUX2)................. 395
Figure 3-44. CM_CORE_AON_TIMER1 Clock Manager Overview.. 397
Figure 3-45. CM_CORE_AON_TIMER2 Clock Manager Overview.. 398
Figure 3-46. CM_CORE_AON_TIMER3 Clock Manager Overview.. 399
Figure 3-47. CM_CORE_AON_TIMER4 Clock Manager Overview.. 400
Figure 3-48. CM_CORE_AON_MCASP1 Clock Manager Overview...401
Figure 3-49. CM_CORE_AON_MCASP2 Clock Manager Overview...402
Figure 3-50. CM_CORE_AON_MCASP3 Clock Manager Overview...403
Figure 3-51. Generic DPLL Functional Diagram...405
Figure 3-52. DPLL_PER Overview... 410
Figure 3-53. DPLL_CORE Overview.. 412
Figure 3-54. DPLL_ABE Overview... 414
Figure 3-55. DPLL_MPU Overview... 416
Figure 3-56. DPLL_IVA Overview...418
Figure 3-57. DPLL_USB Overview..419
Figure 3-58. DPLL_EVE Overview.. 420
Figure 3-59. DPLL_DSP Overview..422
Figure 3-60. DPLL_GMAC Overview..424
Figure 3-61. DPLL_GPU Overview...426
Figure 3-62. DPLL_DDR Overview...427
Figure 3-63. DPLL_PCIE_REF Overview..429
Figure 3-64. APLL_PCIE Overview... 430
Figure 3-65. CD_WKUPAON Overview...431
Figure 3-66. CD_DSP1 Overview..437
Figure 3-67. CD_DSP2 Overview..440
Figure 3-68. CD_CUSTEFUSE Overview.. 443
Figure 3-69. CD_MPU Overview... 444
Figure 3-70. CD_L4PER1 Overview..447
Figure 3-71. CD_L4PER2 Overview..479
Figure 3-72. CD_L4PER3 Overview..493
Figure 3-73. CD_L4SEC Overview..496
Figure 3-74. CD_L3INIT Overview.. 498
Figure 3-75. CD_IVA Overview..508
Figure 3-76. CD_GPU Overview... 510
Figure 3-77. CD_EMU Overview... 512
Figure 3-78. CD_DSS Overview..514
Figure 3-79. CD_L4_CFG Overview..520
Figure 3-80. CD_L3_INSTR Overview... 524
Figure 3-81. CD_L3_MAIN1 Overview... 526
Figure 3-82. CD_EMIF Overview...530
Figure 3-83. CD_IPU Overview... 532
Figure 3-84. CD_IPU1 Overview... 536
Figure 3-85. CD_IPU2 Overview... 539
Figure 3-86. CD_DMA Overview... 542
Figure 3-87. CD_ATL Overview...544
Figure 3-88. CD_CAM Overview... 546
Figure 3-89. CD_GMAC Overview.. 548
Figure 3-90. CD_VPE Overview.. 551
Figure 3-91. CD_EVE1 Overview...553
Figure 3-92. CD_EVE2 Overview... 555
Figure 3-93. CD_RTC Overview...557
Figure 3-94. CD_PCIE Overview..560
Figure 3-95. PRM Voltage Control Architecture..598
Figure 3-96. SmartReflex Integration..599
Figure 3-97. I/O Pads Daisy-Chain Configuration... 605
Figure 3-98. DPLL Output-Frequency Change..609
Figure 4-1. MPU Subsystem Overview...1271
Figure 4-2. MPU Subsystem Integration...1274
Figure 4-3. MPU Subsystem Clocking Scheme..1275
Figure 4-4. MPU Subsystem Reset Scheme.. 1276
Figure 4-5. MPU Subsystem Block Diagram.. 1278

Figure 4-6. MPU_MA Overview....... 1283
Figure 4-7. Fixed MPU-to-EMIF Address Mapping Without High-Order Interleaving....... 1285
Figure 4-8. Fixed MPU-to-EMIF Address Mapping With High-Order Interleaving....... 1286
Figure 4-9. MPU Subsystem Power Domains Overview....... 1294
Figure 4-10. MPU_WUGEN Overview....... 1299
Figure 5-1. DSP Subsystem Highlight....... 1350
Figure 5-2. DSP Subsystem Integration....... 1354
Figure 5-3. DSP_SYSTEM Block Diagram....... 1370
Figure 5-4. Extended Duration Sleep Software and Hardware Sequence....... 1374
Figure 5-5. DSP Subsystem Interrupt Management....... 1376
Figure 5-6. ERRINT Diagram....... 1379
Figure 5-7. DSP DMA Requests....... 1382
Figure 7-1. IPUx Subsystem Overview....... 1434
Figure 7-2. IPU1 Subsystem Integration....... 1436
Figure 7-3. IPU2 Subsystem Integration....... 1437
Figure 7-4. IPUx Subsystem Clocking Scheme....... 1439
Figure 7-5. IPUx Subsystem Reset Scheme....... 1440
Figure 7-6. IPUx Subsystem Block Diagram....... 1442
Figure 7-7. IPUx_WUGEN Overview....... 1446
Figure 7-8. SCTM Block Diagram....... 1448
Figure 7-9. Event Communication Connection in IPUx Subsystem....... 1452
Figure 8-1. EVE Overview....... 1481
Figure 8-2. EVE Integration....... 1484
Figure 8-3. EVE1 Signals....... 1487
Figure 8-4. EVE2 Signals....... 1488
Figure 8-5. WBUF Bank Organization....... 1490
Figure 8-6. IBUF Bank Organization....... 1491
Figure 8-7. VCOP VersusSystem Memory Ownership Examples....... 1492
Figure 8-8. Parity Error ARP32/OCP Disconnect....... 1495
Figure 8-9. EVE Program Cache Architecture....... 1497
Figure 8-10. EDMA Block Diagram....... 1502
Figure 8-11. Structure of Parameter RAM Sets and Contents....... 1503
Figure 8-12. Channel Controller Block Diagram....... 1505
Figure 8-13. CME Done Logic....... 1507
Figure 8-14. Data Partitioning....... 1508
Figure 8-15. Task Partitioning....... 1508
Figure 8-16. EVE Interrupt Block Diagram....... 1510
Figure 8-17. EVE Memory Switch Error Interrupt....... 1511
Figure 8-18. EVE Parity/Error Detect Interrupt....... 1512
Figure 8-19. EVE INTC for ARP32....... 1513
Figure 8-20. Extended Duration Sleep Software/Hardware Sequence....... 1518
Figure 8-21. SCTM Block Diagram....... 1599
Figure 8-22. Watchdog Operation....... 1604
Figure 8-23. Typical STM System Integration....... 1605
Figure 8-24. SMSET Block Diagram....... 1617
Figure 8-25. SMSET Interfaces....... 1617
Figure 8-26. ARP32 Versions and ISA/Feature Space....... 1626
Figure 8-27. ARP32 CPU Block Diagram....... 1627
Figure 8-28. ARP32 CPU Pipeline....... 1630
Figure 8-29. ARP32 CPU Pipeline Operation....... 1631
Figure 8-30. ARP32 CPU Little Endianness....... 1632
Figure 8-31. Control Status Register (CSR)....... 1635
Figure 8-32. Interrupt Enable Register (IER)....... 1636
Figure 8-33. Interrupt Flag Register (IFR)....... 1637
Figure 8-34. Interrupt Set Register (ISR)....... 1638
Figure 8-35. Interrupt Clear Register (ICR)....... 1639
Figure 8-36. NMI Return Pointer Register (NRP)....... 1640
Figure 8-37. Interrupt Return Pointer Register (IRP)....... 1640
Figure 8-38. Stack Pointer Register (SP)....... 1641
Figure 8-39. Global Data Pointer Register (GDP)....... 1641
Figure 8-40. Link Register (LR)....... 1641

Figure 8-41. Loop 0 Start Address Register (LSA0)....1642
Figure 8-42. Loop 0 End Address Register (LEA0)....1642
Figure 8-43. Loop 0 Iteration Count Register (LCNT0)....1643
Figure 8-44. Loop 1 Start Address Register (LSA1)....1644
Figure 8-45. Loop 1 End Address Register (LEA1)....1644
Figure 8-46. Loop 1 Iteration Count Register (LCNT1)....1645
Figure 8-47. Loop 0 Iteration Count Reload Value Register (LCNT0RLD)....1645
Figure 8-48. Shadow Control Status Register (SCSR)....1646
Figure 8-49. NMI Shadow Control Status Register (NMISCSR)....1647
Figure 8-50. CPU Identification Register (CPUID)....1648
Figure 8-51. Decode Program Counter Register (DPC)....1649
Figure 8-52. Time Stamp Counter Register - Low Half (TSCL)....1649
Figure 8-53. Time Stamp Counter Register - High Half (TSCH)....1649
Figure 8-54. Loop Operation....1657
Figure 8-55. Interrupt Processing....1665
Figure A-56. Power-On-Reset....1816
Figure A-57. CPU Standby and Wakeup Procedure....1817
Figure 8-58. EVE Block Diagram....1821
Figure 8-59. VCOP Block Diagram....1822
Figure 8-60. EVE Memory Map....1824
Figure 8-61. VCOP Instruction Buffering....1825
Figure 8-62. Addressing a Four-Dimensional Data Object....1832
Figure 8-63. Load Word Distribution Options....1835
Figure 8-64. Load halfword Distribution Options....1836
Figure 8-65. Load Byte Distribution Options....1837
Figure 8-66. VST Rounding and Saturation Parameters....1842
Figure 8-67. Lookup Table Organization for Various Entry Size and Parallel Tables (NWAY = 8)....1847
Figure 8-68. Example of Operation Delay Slots....1857
Figure 8-69. Binlog Function....1860
Figure 8-70. VMPY, VMADD and VMSUB Rounding Parameters....1868
Figure 9-1. VIP Overview....1883
Figure 9-2. VIP Environment....1885
Figure 9-3. VIP Integration....1892
Figure 9-4. VIP Block Diagram....1894
Figure 9-5. VIP Slice Processing Path Block Diagram....1896
Figure 9-6. Input: A=RGB, B=YUV422; Output: A=RGB, B=RGB....1898
Figure 9-7. Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=RGB....1899
Figure 9-8. Input: A=RGB, B=YUV422; Output: A=RGB, B=Scaled YUV420....1900
Figure 9-9. Input: A=YUV444, B=YUV422; Output: A=YUV422, A=Scaled YUV422, B=YUV422....1901
Figure 9-10. Input: A=YUV444; Output: A=Scaled YUV420, A=YUV420....1902
Figure 9-11. Input: A=YUV444; Output: A=Scaled YUV420, A=YUV444....1903
Figure 9-12. Input: A=YUV422 8/16; Output: A=Scaled YUV420, A=YUV444....1904
Figure 9-13. Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=YUV420....1905
Figure 9-14. Input: A=YUV422 8/16, B=YUV422; Output: A=YUV420, B=YUV420....1906
Figure 9-15. Bytelane Swapping Modes....1908
Figure 9-16. RAW16 to RGB565 Mapping....1908
Figure 9-17. RAW12 Swap....1909
Figure 9-18. NTSC Analog Video Waveform for One Horizontal Line....1909
Figure 9-19. Digitized Video....1910
Figure 9-20. Code Word Embedded Video Format....1910
Figure 9-21. Digitized Video with F, V, and H Flags in EAV/SAV....1911
Figure 9-22. Planar Buffer Storage Description....1912
Figure 9-23. 8-bit Interface Discrete Sync Pixel Multiplexing....1913
Figure 9-24. 16-bit Interface Discrete Sync Pixel Multiplexing....1913
Figure 9-25. 24b Interface RGB Discrete Sync....1913
Figure 9-26. Discrete Sync Signals....1914
Figure 9-27. Type 1, First Horizontal Blanking Pixel....1915
Figure 9-28. Type 1, First Vertical Ancillary Data Pixel....1915
Figure 9-29. Type 1, Horizontal Blanking in Video Region....1915
Figure 9-30. Type 1, First Video Pixel....1916
Figure 9-31. 4-Pin Reduced ACTVID Signaling with Vertical Ancillary Data....1916

Figure 9-32. 4-Pin Reduced ACTVID Signaling with No Vertical Ancillary Data....1916
Figure 9-33. 4-Pin Reduced HSYNC Signaling with Vertical Ancillary Data....1917
Figure 9-34. VSYNC Pre and Post Window....1917
Figure 9-35. VSYNC Equivalence When Using Transition Window....1918
Figure 9-36. FID Registering When Using HSYNC....1919
Figure 9-37. FID Registering When Using ACTVID....1919
Figure 9-38. Field ID Determination By VSYNC Skew....1920
Figure 9-39. Example of 525-line FID Determination By VSYNC Skew....1921
Figure 9-40. Horizontal Ancillary Data Packing When HSYNC Used as Sync Signal....1922
Figure 9-41. Interlaced Field Vertical Blanking Ancillary Data Storage....1922
Figure 9-42. Progressive Frame Vertical Blanking Ancillary Data Storage....1923
Figure 9-43. Embedded Sync Data Entry....1923
Figure 9-44. Code Word Format Example Followed by Video Data....1924
Figure 9-45. Embedded Sync Packing....1926
Figure 9-46. RGB Frame Storage....1927
Figure 9-47. 2-Way Multiplexing....1928
Figure 9-48. Example of 4-Way Multiplexing....1928
Figure 9-49. Example of Line Multiplexing....1929
Figure 9-50. 8-bit Line Mux Interface....1929
Figure 9-51. 16-bit Line Mux Interface....1930
Figure 9-52. BOP/EOP Definition of a Period....1931
Figure 9-53. Channel ID Inserted Into Horizontal Blanking....1933
Figure 9-54. Vertical Ancillary Data Cropping....1934
Figure 9-55. Active Video Cropping....1934
Figure 9-56. Problematic Error Cropping Case....1935
Figure 9-57. Endline/Endframe Behavior for Error Cropping Case....1935
Figure 9-58. Generic External Sync Signals....1940
Figure 9-59. vblnk and hblnk....1940
Figure 9-60. VBLNK and ACTID (1)....1940
Figure 9-61. VBLNK and ACTVID(2)....1941
Figure 9-62. VBLNK and HSYNC....1941
Figure 9-63. VSYNC and HBLNK....1941
Figure 9-64. VSYNC and ACTIVID(1)....1942
Figure 9-65. VSYNC and ACTIVID(2)....1942
Figure 9-66. VSYNC and HSYNC....1942
Figure 9-67. Ancillary and Active Video Line Determination....1943
Figure 9-68. HSYNC Pixel Capture....1943
Figure 9-69. ACTVID Pixel Capture....1943
Figure 9-70. Matrix Format....1945
Figure 9-71. Conversion from RGB to YCbCr....1946
Figure 9-72. Conversion from YCbCr to RGB....1946
Figure 9-73. Conversion from RGB to YCbCr....1946
Figure 9-74. Conversion from YCbCr to RGB....1947
Figure 9-75. Conversion from RGB to YCbCr....1948
Figure 9-76. Conversion from YCbCr to RGB....1948
Figure 9-77. Conversion from RGB to YCbCr....1949
Figure 9-78. Conversion from YCbCr to RGB....1949
Figure 9-79. High Level Block Diagram....1951
Figure 9-80. SC Block Diagram....1951
Figure 9-81. Input Image Trimming....1952
Figure 9-82. Filter Implementation and Parameter Description....1953
Figure 9-83. Peaking Filter at fs/4....1954
Figure 9-84. Vertical Scaler Block Diagram....1954
Figure 9-85. Horizontal Scaler Block Diagram....1956
Figure 9-86. Polyphase Filtering Example....1957
Figure 9-87. Non-linear Scaling Example....1958
Figure 9-88. SRAM Layout for 7tap Coefficient....1961
Figure 9-89. SRAM Layout for 5tap Coefficient....1961
Figure 9-90. VPI Control I/F Coef Data Format (7tap)....1961
Figure 9-91. VPI Control I/F Coef Data Format (5tap)....1961
Figure 9-92. VPI Control I/F Coef Data Format (3tap)....1961

Figure 9-93. VPI Control I/F Memory Map (Write)........ 1962
Figure 9-94. VPI Control I/F Memory Map (Read)........ 1962
Figure 9-95. Inbound Data Transfer Descriptor Format........ 2006
Figure 9-96. Outbound Data Transfer Descriptor Format........ 2006
Figure 9-97. Y 4:4:4 (Data Type 0)........ 2022
Figure 9-98. Y 4:2:2 (Data Type 1)........ 2022
Figure 9-99. Y 4:2:0 (Data Type 2)........ 2023
Figure 9-100. C 4:4:4 (Data Type 4)........ 2023
Figure 9-101. C 4:2:2 (Data Type 5)........ 2024
Figure 9-102. C 4:2:0 (Data Type 6)........ 2024
Figure 9-103. YC 4:2:2 (Data Type 7)........ 2025
Figure 9-104. YC 4:4:4 (Data Type 8)........ 2025
Figure 9-105. CY 4:2:2 (Data Type 23h)........ 2026
Figure 9-106. RGB16-565 (Data Type 0)........ 2026
Figure 9-107. ARGB-1555 (Data Type 1)........ 2027
Figure 9-108. ARGB-4444 (Data Type 2)........ 2027
Figure 9-109. RGBA-5551 (Data Type 3)........ 2028
Figure 9-110. RGBA-4444 (Data Type 4)........ 2028
Figure 9-111. ARGB24-6666 (Data Type 5)........ 2029
Figure 9-112. RGB24-888 (Data Type 6)........ 2029
Figure 9-113. ARGB32-8888 (Data Type 7)........ 2030
Figure 9-114. RGBA24-6666 (Data Type 8)........ 2030
Figure 9-115. RGBA32-8888 (Data Type 9)........ 2031
Figure 10-1. VPE Integration........ 2306
Figure 10-2. VPE Block Diagram........ 2308
Figure 10-3. Block Diagram of Motion-Adaptive Deinterlacer........ 2309
Figure 10-4. Motion Detection (MDT) Block Diagram........ 2311
Figure 10-5. Motion Detection (MDT) MV Calc Data Path........ 2311
Figure 10-6. Motion Detection (MDT) Max Filter Data Path........ 2312
Figure 10-7. Motion Detection (MDT) Uniform Area Data Path........ 2312
Figure 10-8. Motion Detection (MDT) Detect Motion Data Path........ 2313
Figure 10-9. Edge Directed Interpolation (EDI) Block Diagram........ 2313
Figure 10-10. Edge Directed Interpolation Edge Vector Calculation........ 2314
Figure 10-11. Edge Directed Interpolation Luma Interpolation Calculationn........ 2314
Figure 10-12. Edge Directed Interpolation Chroma Interpolation Calculation........ 2314
Figure 10-13. Film Mode Detection (FMD) Block Diagram........ 2315
Figure 10-14. Film Mode Detection (FMD) Frame/Field Difference Calculation........ 2316
Figure 10-15. Film Mode Detection (FMD) Combing Artifacts Calculation........ 2316
Figure 10-16. Film Mode Detection (FMD) Combing Artifacts Data Path........ 2317
Figure 10-17. Output Data Path........ 2317
Figure 10-18. High Level Block Diagram........ 2318
Figure 10-19. SC Block Diagram........ 2319
Figure 10-20. Input Image Trimming........ 2319
Figure 10-21. Filter Implementation and Parameter Description........ 2320
Figure 10-22. Peaking Filter at fs/4........ 2321
Figure 10-23. Vertical Scaler Block Diagram........ 2321
Figure 10-24. Horizontal Scaler Block Diagram........ 2323
Figure 10-25. Polyphase Filtering Example........ 2324
Figure 10-26. Non-linear Scaling Example........ 2325
Figure 10-27. SRAM Layout for 7tap Coefficient........ 2328
Figure 10-28. SRAM Layout for 5tap Coefficient........ 2328
Figure 10-29. VPI Control I/F Coef Data Format (7tap)........ 2328
Figure 10-30. VPI Control I/F Coef Data Format (5tap)........ 2328
Figure 10-31. VPI Control I/F Coef Data Format (3tap)........ 2328
Figure 10-32. VPI Control I/F Memory Map (Write)........ 2329
Figure 10-33. VPI Control I/F Memory Map (Read)........ 2329
Figure 10-34. Matrix Format........ 2349
Figure 10-35. Conversion from RGB to YCbCr........ 2350
Figure 10-36. Conversion from YCbCr to RGB........ 2350
Figure 10-37. Conversion from RGB to YCbCr........ 2350
Figure 10-38. Conversion from YCbCr to RGB........ 2351

Figure 10-39. Conversion from RGB to YCbCr.......... 2352
Figure 10-40. Conversion from YCbCr to RGB.......... 2352
Figure 10-41. Conversion from RGB to YCbCr.......... 2353
Figure 10-42. Conversion from YCbCr to RGB.......... 2353
Figure 10-43. 4:2:0 YCrCb Color Space with Chroma Left-aligned.......... 2355
Figure 10-44. 4:2:0 YCrCb Color Space with Chroma Left-aligned.......... 2355
Figure 10-45. 4:2:0 YCrCb Color Space with Chroma Left-aligned.......... 2355
Figure 10-46. Anchor Pixels.......... 2356
Figure 10-47. 4:2:0 Interlaced Scan.......... 2356
Figure 10-48. Ideal 4:2:2 Chroma Upsampling for Interlaced Scan.......... 2357
Figure 10-49. Inbound Data Transfer Descriptor Format.......... 2367
Figure 10-50. Outbound Data Transfer Descriptor Format.......... 2368
Figure 10-51. Y 4:4:4 (Data Type 0).......... 2384
Figure 10-52. Y 4:2:2 (Data Type 1).......... 2385
Figure 10-53. Y 4:2:0 (Data Type 2).......... 2385
Figure 10-54. C 4:4:4 (Data Type 4).......... 2386
Figure 10-55. C 4:2:2 (Data Type 5).......... 2386
Figure 10-56. C 4:2:0 (Data Type 6).......... 2387
Figure 10-57. YC 4:2:2 (Data Type 7).......... 2387
Figure 10-58. YC 4:4:4 (Data Type 8).......... 2388
Figure 10-59. CY 4:2:2 (Data Type 23h).......... 2388
Figure 10-60. RGB16-565 (Data Type 0).......... 2389
Figure 10-61. ARGB-1555 (Data Type 1).......... 2390
Figure 10-62. ARGB-4444 (Data Type 2).......... 2390
Figure 10-63. RGBA-5551 (Data Type 3).......... 2391
Figure 10-64. RGBA-4444 (Data Type 4).......... 2391
Figure 10-65. ARGB24-6666 (Data Type 5).......... 2392
Figure 10-66. RGB24-888 (Data Type 6).......... 2392
Figure 10-67. ARGB32-8888 (Data Type 7).......... 2393
Figure 10-68. RGBA24-6666 (Data Type 8).......... 2393
Figure 10-69. RGBA32-8888 (Data Type 9).......... 2394
Figure 10-70. RGB16-565 (Data Type 0).......... 2394
Figure 10-71. ARGB-1555 (Data Type 1).......... 2395
Figure 10-72. ARGB-4444 (Data Type 2).......... 2395
Figure 10-73. RGBA-5551 (Data Type 3).......... 2396
Figure 10-74. RGBA-4444 (Data Type 4).......... 2396
Figure 10-75. ARGB24-6666 (Data Type 5).......... 2397
Figure 10-76. RGB24-888 (Data Type 6).......... 2397
Figure 10-77. ARGB32-8888 (Data Type 7).......... 2398
Figure 10-78. RGBA24-6666 (Data Type 8).......... 2398
Figure 10-79. RGBA32-8888 (Data Type 9).......... 2399
Figure 11-1. Display Subsystem Overview.......... 2600
Figure 11-2. Display Subsystem Environment.......... 2601
Figure 11-3. Display Subsystem Integration.......... 2604
Figure 11-4. Display Subsystem Clock Tree.......... 2605
Figure 11-5. Display Subsystem Reset Scheme.......... 2607
Figure 11-6. Display Subsystem SIdleAck/MStandby Generation.......... 2609
Figure 11-7. Display Subsystem Wake-Up Generation.......... 2610
Figure 11-8. PLL Controller Overview.......... 2611
Figure 11-9. OCP2SCP2 Overview.......... 2611
Figure 11-10. VIDEO PLL Reference Diagram.......... 2613
Figure 11-11. DPLL_VIDEO Functional Block Diagram.......... 2613
Figure 11-12. VIDEO PLL Power State Diagram.......... 2615
Figure 11-13. VIDEO PLL Programming Sequence.......... 2616
Figure 11-14. VIDEO PLL Go Sequence (Manual Mode).......... 2617
Figure 11-15. VIDEO PLL Go Sequence (Automatic Mode).......... 2618
Figure 11-16. DPLL_HDMI and PLLCTRL_HDMI Overview.......... 2619
Figure 11-17. DPLL_HDMI and PLLCTRL_HDMI Reference Diagram.......... 2620
Figure 11-18. DPLL_HDMI Functional Block Diagram.......... 2620
Figure 11-19. PLLCTRL_HDMI Power State Diagram.......... 2623
Figure 11-20. DPLL_HDMI Programming Sequence.......... 2624

Figure 11-21. DPLL_HDMI GO Sequence (Manual Mode)....2625
Figure 11-22. DISPC Overview....2651
Figure 11-23. DISPC LCD Support Parallel Interface....2655
Figure 11-24. DISPC LCD Pixel Data Color12 Active Matrix....2657
Figure 11-25. DISPC LCD Pixel Data Color16 Active Matrix....2658
Figure 11-26. DISPC LCD Pixel Data Color18 Active Matrix....2658
Figure 11-27. DISPC LCD Pixel Data Color24 Active Matrix....2659
Figure 11-28. DISPC Active Matrix Timing Diagram of Configuration 1 (Start of Frame)....2661
Figure 11-29. DISPC Active Matrix Timing Diagram of Configuration 1 (Between Lines)....2661
Figure 11-30. DISPC Active Matrix Timing Diagram of Configuration 1 (Between Frames)....2661
Figure 11-31. DISPC Active Matrix Timing Diagram of Configuration 1 (End of Frame)....2661
Figure 11-32. DISPC Active Matrix Timing Diagram of Configuration 2 (Start of Frame)....2662
Figure 11-33. DISPC Active Matrix Timing Diagram of Configuration 2 (Between Lines)....2662
Figure 11-34. DISPC Active Matrix Timing Diagram of Configuration 2 (Between Frames)....2662
Figure 11-35. DISPC Active Matrix Timing Diagram of Configuration 2 (End of Frame)....2662
Figure 11-36. DISPC Active Matrix Timing Diagram of Configuration 3 (Start of Frame)....2663
Figure 11-37. DISPC Active Matrix Timing Diagram of Configuration 3 (Between Lines)....2663
Figure 11-38. DISPC Active Matrix Timing Diagram of Configuration 3 (Between Frames)....2663
Figure 11-39. DISPC Active Matrix Timing Diagram of Configuration 3 (End of Frame)....2663
Figure 11-40. DISPC TV Output Pixel Data....2664
Figure 11-41. DISPC Integration....2665
Figure 11-42. DISPC Architecture Overview....2668
Figure 11-43. DISPC Clock Tree Overview....2669
Figure 11-44. DISPC DMA Engine Overview....2674
Figure 11-45. YUV4:2:2 Predecimation....2680
Figure 11-46. DISPC 90-Degree Rotation With Mirroring....2682
Figure 11-47. DISPC Graphics Pipeline....2687
Figure 11-48. DISPC Configuration 1: Video Pipeline....2690
Figure 11-49. DISPC Configuration 2: Video Pipeline....2690
Figure 11-50. DISPC YCbCr to RGB Registers (FULLRANGE = 0), 8-Bit Outputs....2692
Figure 11-51. DISPC YCbCr to RGB Registers (FULLRANGE = 1), 8-Bit Outputs....2693
Figure 11-52. DISPC YCbCr to RGB Registers (FULLRANGE = 0), 10-Bit Outputs....2693
Figure 11-53. DISPC YCbCr to RGB Registers (FULLRANGE = 1), 10-Bit Outputs....2693
Figure 11-54. DISPC Averaging of the Chrominance Formula....2693
Figure 11-55. DISPC Averaging of the Chrominance Representation....2694
Figure 11-56. DISPC YUV4:2:2 to RGB30 Using Averaging of the Chrominance....2694
Figure 11-57. DISPC YUV4:2:0 to RGB30 Using Scaler Unit for Resampling Chrominance....2694
Figure 11-58. DISPC YUV4:2:2 to RGB30 Using Scaler Unit for Resampling Chrominance....2694
Figure 11-59. DISPC Video Upsampling....2695
Figure 11-60. DISPC Macro-Architecture of the Horizontal Scaling for A, R, G, B, and Y Components (5-tap Restriction)..2697
Figure 11-61. DISPC Macro-Architecture of the Vertical Scaling for A, R, G, B, and Y Components (5 and 3 taps)....2697
Figure 11-62. DISPC Macro-Architecture of the Horizontal Scaling for Cr and Cb Components (5-tap Restriction)....2698
Figure 11-63. DISPC Macro-Architecture of the Vertical Scaling for Cr and Cb Components (5 and 3 taps)....2698
Figure 11-64. DISPC Vertical Upsampling and Downsampling Algorithm....2701
Figure 11-65. DISPC Horizontal Up/Downsampling Algorithm....2702
Figure 11-66. DISPC Write-Back Pipeline....2703
Figure 11-67. DISPC RGB to YCbCr (FULLRANGE = 0)....2704
Figure 11-68. DISPC RGB to YCbCr (FULLRANGE = 1)....2704
Figure 11-69. DISPC Macro-Architecture of the Vertical Scaling for A, R, G, B, and Y Components....2705
Figure 11-70. DISPC Macro-Architecture of the Horizontal Scaling for A, R, G, B, and Y Components....2705
Figure 11-71. DISPC Macro-Architecture of the Vertical Scaling for Cr and Cb Components....2705
Figure 11-72. DISPC Macro-Architecture of the Horizontal Scaling for Cr and Cb Components....2706
Figure 11-73. DISPC LCD Output Architecture....2710
Figure 11-74. DISPC Priority Rule Architecture....2712
Figure 11-75. DISPC Example of Priority Rule: From Lower to Higher VID1, VID2, VID3, GFX....2713
Figure 11-76. DISPC Alpha Blending Architecture With Premultiplied Alpha Support....2714
Figure 11-77. DISPC Source Transparency Color Key Example....2717
Figure 11-78. DISPC Destination Transparency Color Key Example....2718
Figure 11-79. DISPC LCD1 Gamma Correction Architecture....2719
Figure 11-80. DISPC Data Memory Organization for Gamma Mode in LCD Output....2719
Figure 11-81. DISPC CPR Matrix....2720

Figure 11-82. DISPC CPR Macro-Architecture....2721
Figure 11-83. DISPC RGB to YUV Registers (FullRange=0)....2721
Figure 11-84. DISPC RGB to YUV Registers (FullRange=1)....2722
Figure 11-85. DISPC Signal Mapping in BT.656 Mode....2722
Figure 11-86. DISPC Signal Mapping in BT.1120 Mode....2722
Figure 11-87. Bit-assignment for the fourth byte of EAV/SAV codes....2723
Figure 11-88. DISPC 8-Bit Interface Settings....2726
Figure 11-89. DISPC 9-Bit Interface Settings....2727
Figure 11-90. DISPC 12-Bit Interface Settings....2728
Figure 11-91. DISPC 16-Bit Interface Settings....2729
Figure 11-92. DISPC Timing Values (Active Matrix Display)....2731
Figure 11-93. DISPC TV Output Architecture....2732
Figure 11-94. DISPC Data Memory Organization for Gamma Mode in TV Output....2732
Figure 11-95. DISPC Example Timing TV Formats....2733
Figure 11-96. DISPC Frame Width Control....2735
Figure 11-97. DISPC Illustration of 3D Interleaving....2736
Figure 11-98. DISPC Illustration of a Non-zero Position of 3D Window....2737
Figure 11-99. DISPC Illustration of DLP 3D Format....2738
Figure 11-100. DISPC GFX Pipeline Processing Configuration....2749
Figure 11-101. DISPC Video Pipeline Processing Configuration....2751
Figure 11-102. DISPC Scaler Unit Programming Flow....2753
Figure 11-103. HDMI Overview....2897
Figure 12-1. GPU Overview....2902
Figure 12-2. GPU Integration....2905
Figure 12-3. GPU Block Diagram....2907
Figure 13-1. BB2D Overview....2927
Figure 13-2. BB2D Integration....2929
Figure 13-3. BB2D Block Diagram....2931
Figure 14-1. Interconnect Overview....2969
Figure 14-2. L3_MAIN Interconnect Overview....2970
Figure 14-3. Connectivity Matrix....2977
Figure 14-4. Bandwidth Regulator Pressure Settings....2979
Figure 14-5. Flag Mux Structure....2982
Figure 14-6. L3 Interconnect Region Overlay and Priority Level Overview....2991
Figure 14-7. L3_MAIN Global Error-Routing Scheme....2999
Figure 14-8. Typical Error Analysis Sequence....3003
Figure 14-9. L4 Interconnect Overview....3126
Figure 14-10. L4 Initiator-Target Connectivity....3133
Figure 14-11. Example of CONNID_BIT_VECTOR L4_AP_PROT_GROUP_MEMBERS_k....3135
Figure 14-12. L4 Segmentation....3140
Figure 14-13. L4 Error Reporting....3151
Figure 14-14. Protection Violation Out-of-Band Error Reporting....3152
Figure 14-15. Typical Error Analysis Sequence....3154
Figure 15-1. Memory Subsystem Functional Diagram....3212
Figure 15-2. DMM and Tiler Overview....3217
Figure 15-3. DMM Integration....3218
Figure 15-4. DMM Block Diagram....3220
Figure 15-5. DMM Sections and Memory Mapping....3224
Figure 15-6. PAT Direct Access Translation....3226
Figure 15-7. PAT Indirect Access Translation....3227
Figure 15-8. Physical Address Translation Table....3228
Figure 15-9. PAT Descriptors....3230
Figure 15-10. TILER Address Space Structure for Tiled Modes....3233
Figure 15-11. TILER Object Containers and Views....3236
Figure 15-12. TILER Memory Footprint With PAT and Shared Physical Address Translation LUT....3236
Figure 15-13. Object Container Geometry With 4-kiB Pages....3237
Figure 15-14. TILER Page Mapping When Using 4-kiB Pages....3237
Figure 15-15. Isometric Transforms in the TILER Container....3238
Figure 15-16. Tile Geometry....3239
Figure 15-17. Subtile Mapping....3240
Figure 15-18. Page Mode Virtual Addressing....3241

Figure 15-19. Tiled Mode Addressing in 0- or 180-Degree Orientation (S = 0)....................3241
Figure 15-20. Tiled Mode Addressing in 90- or 270-Degree Orientation (S = 1)....................3242
Figure 15-21. Tiled Mode Ordering of Elements in Natural View....................3243
Figure 15-22. Page Mode Ordering of Elements in Natural View....................3243
Figure 15-23. Tiled Mode Ordering of Elements in 0-Degree View With Vertical Mirror....................3244
Figure 15-24. Page Mode Ordering of Elements in 0-Degree View With Vertical Mirror....................3244
Figure 15-25. Tiled Mode Ordering of Elements in 0-Degree View With Horizontal Mirror....................3245
Figure 15-26. Page Mode Ordering of Elements in 0-Degree View With Horizontal Mirror....................3245
Figure 15-27. Tiled Mode Ordering of Elements in 180-Degree View....................3246
Figure 15-28. Page Mode Ordering of Elements in 180-Degree View....................3246
Figure 15-29. Tiled Mode Ordering of Elements in 90-Degree View With Vertical Mirror....................3247
Figure 15-30. Page Mode Ordering of Elements in 90-Degree View With Vertical Mirror....................3247
Figure 15-31. Tiled Mode Ordering of Elements in 270-Degree View....................3248
Figure 15-32. Page Mode Ordering of Elements in 270-Degree View....................3248
Figure 15-33. Tiled Mode Ordering of Elements in 90-Degree View....................3249
Figure 15-34. Page Mode Ordering of Elements in 90-Degree View....................3249
Figure 15-35. Tiled Mode Ordering of Elements in 90-Degree View With Horizontal Mirror....................3250
Figure 15-36. Page Mode Ordering of Elements in 90-Degree View With Horizontal Mirror....................3250
Figure 15-37. TILER Port Address Map....................3251
Figure 15-38. TILER Aliased View Orientation....................3252
Figure 15-39. Simple Manual Area Refill Scheme....................3255
Figure 15-40. Single Auto-Configured Area Refill Scheme....................3256
Figure 15-41. Chained Auto-Configured Area Refill Scheme....................3257
Figure 15-42. Synchronized Auto-Configured Area Refill Scheme....................3258
Figure 15-43. Cyclic Synchronized Auto-Configured Area Refill Scheme....................3259
Figure 15-44. Example of 8-Bit Frame-Buffer Addressing in any Orientation....................3266
Figure 15-45. EMIF Controller Overview....................3303
Figure 15-46. EMIF DDR2/DDR3 Configuration Without ECC....................3304
Figure 15-47. EMIF DDR2/DDR3 Configuration With ECC....................3305
Figure 15-48. EMIF Modules Integration....................3308
Figure 15-49. EMIF Block Diagram....................3311
Figure 15-50. FIFO Block Diagram....................3312
Figure 15-51. Example for Using the CKE Tri-state Functionality....................3334
Figure 15-52. GPMC Overview....................3446
Figure 15-53. GPMC to 16-Bit Address/Data-Multiplexed Memory....................3447
Figure 15-54. GPMC to 16-Bit Nonmultiplexed Memory....................3447
Figure 15-55. GPMC to 8-Bit Nonmultiplexed Memory....................3448
Figure 15-56. GPMC to 8-Bit NAND Device....................3448
Figure 15-57. GPMC Integration....................3451
Figure 15-58. GPMC Block Diagram....................3455
Figure 15-59. Chip-Select Address Mapping and Decoding Mask....................3461
Figure 15-60. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1)....................3464
Figure 15-61. Wait Behavior During a Synchronous Read Burst Access....................3466
Figure 15-62. Read-to-Read for an Address-Data Multiplexed Device, on Different Chip-Select, Without Bus Turnaround (nCS Attached to a Fast Device)....................3468
Figure 15-63. Read- to-Read/Write for an Address-Data Multiplexed Device, on Different Chip-Select, With Bus Turnaround....................3468
Figure 15-64. Read-to-Read/Write for a Address-Data or AAD-Multiplexed Device, on Same Chip-Select, With Bus Turnaround....................3469
Figure 15-65. Asynchronous Single Read on an Address/Data-Multiplexed Device....................3478
Figure 15-66. Two Asynchronous Single-Read Accesses on an Address/Data-Multiplexed Device (32-Bit Read Split Into 2 x 16-Bit Read)....................3479
Figure 15-67. Asynchronous Single-Write on an Address/Data-Multiplexed Device....................3480
Figure 15-68. Asynchronous Single Read on an AAD-Multiplexed Device....................3482
Figure 15-69. Asynchronous Single Write on an AAD-Multiplexed Device....................3483
Figure 15-70. Synchronous Single Read (GPMCFCLKDIVIDER = 0)....................3485
Figure 15-71. Synchronous Single Read (GPMCFCLKDIVIDER = 1)....................3486
Figure 15-72. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0)....................3488
Figure 15-73. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1)....................3489
Figure 15-74. Synchronous Single Write on an Address/Data-Multiplexed Device....................3490
Figure 15-75. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode....................3491

Figure 15-76. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode................................ 3492
Figure 15-77. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device.. 3494
Figure 15-78. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device...3495
Figure 15-79. Asynchronous Multiple (Page Mode) Read...3496
Figure 15-80. NAND Command Latch Cycle..3501
Figure 15-81. NAND Address Latch Cycle.. 3502
Figure 15-82. NAND Data Read Cycle.. 3503
Figure 15-83. NAND Data Write Cycle.. 3503
Figure 15-84. Hamming Code Accumulation Algorithm (1/2).. 3508
Figure 15-85. Hamming Code Accumulation Algorithm (2/2).. 3509
Figure 15-86. ECC Computation for a 256-Byte Data Stream (Read or Write)... 3509
Figure 15-87. ECC Computation for a 512-Byte Data Stream (Read or Write)... 3510
Figure 15-88. 128 Word16 ECC Computation.. 3511
Figure 15-89. 256 Word16 ECC Computation.. 3511
Figure 15-90. Manual Mode Sequence and Mapping...3516
Figure 15-91. NAND Page Mapping and ECC: Per-Sector Schemes... 3520
Figure 15-92. NAND Page Mapping and ECC: Pooled Spare Schemes...3521
Figure 15-93. NAND Page Mapping and ECC: Per-Sector Schemes, With Separate ECC...3522
Figure 15-94. NAND Read Cycle Optimization Timing Description..3529
Figure 15-95. Programming Model Top-Level Diagram..3531
Figure 15-96. NOR Interfacing Timing Parameters Diagram..3536
Figure 15-97. NAND Command Latch Cycle Timing Simplified Example...3539
Figure 15-98. Synchronous NOR Single Read Simplified Example... 3545
Figure 15-99. Asynchronous NOR Single Write Simplified Example..3547
Figure 15-100. GPMC Connection to an External NOR Flash Memory.. 3549
Figure 15-101. Synchronous Burst Read Access (Timing Parameters in Clock Cycles)...3551
Figure 15-102. Asynchronous Single Read Access (Timing Parameters in Clock Cycles).. 3552
Figure 15-103. Asynchronous Single Write Access (Timing Parameters in Clock Cycles).. 3553
Figure 15-104. ELM Overview...3582
Figure 15-105. ELM Integration...3583
Figure 15-106. OCMC_RAMi (i = 1 to 3) Overview.. 3605
Figure 15-107. OCM Subsystem Integration... 3606
Figure 15-108. OCMC Block Diagram...3608
Figure 15-109. VBUF to CBUF Address Mapping..3618
Figure 16-1. DMA_SYSTEM Overview...3660
Figure 16-2. Example of External DMA Requests Use..3661
Figure 16-3. Transition-Sensitive DMA Request Scheme.. 3662
Figure 16-4. DMA_SYSTEM Controller Integration..3663
Figure 16-5. DMA_SYSTEM Controller Top-Level Block Diagram..3677
Figure 16-6. Example Showing Double-Index Addressing, Elements, Frames, and Strides...3683
Figure 16-7. Addressing Mode Example (a)..3683
Figure 16-8. Addressing Mode Example (b)..3683
Figure 16-9. Addressing Mode Example (c)... 3684
Figure 16-10. Example of a 90-Degree Clockwise Image Rotation...3686
Figure 16-11. 2-D Graphic Transparent Color Block Diagram...3693
Figure 16-12. EDMA module Overview.. 3733
Figure 16-13. Example of External DMA Requests Use...3736
Figure 16-14. EDMA Controller Integration.. 3738
Figure 16-15. EDMA Controller Block Diagram..3744
Figure 16-16. EDMA Channel Controller Block Diagram..3745
Figure 16-17. TPTC Block Diagram...3746
Figure 16-18. Definition of ACNT, BCNT, and CCNT...3748
Figure 16-19. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)...3749
Figure 16-20. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)... 3750
Figure 16-21. PaRAM Set..3752
Figure 16-22. Linked Transfer..3760
Figure 16-23. Link-to-Self Transfer..3761
Figure 16-24. DMA Channel and QDMA Channel to PaRAM Mapping..3767
Figure 16-25. QDMA Channel to PaRAM Mapping.. 3768
Figure 16-26. Shadow Region Registers...3770
Figure 16-27. Interrupt Diagram.. 3775

Figure 16-28. Error Interrupt Operation....3780
Figure 16-29. PaRAM Set Content for Proxy Memory Protection Example....3784
Figure 16-30. Channel Options Parameter (OPT) Example....3785
Figure 16-31. Proxy Memory Protection Example....3785
Figure 16-32. EDMA Prioritization....3792
Figure 16-33. Block Move Example....3795
Figure 16-34. Block Move Example PaRAM Configuration....3796
Figure 16-35. Subframe Extraction Transfer....3797
Figure 16-36. Subframe Extraction Example PaRAM Configuration....3797
Figure 16-37. Data Sorting Example....3798
Figure 16-38. Data Sorting Example PaRAM Configuration....3799
Figure 16-39. Servicing Incoming McASP Data Example....3800
Figure 16-40. Servicing Incoming McASP Data Example PaRAM Configuration....3801
Figure 16-41. Servicing Peripheral Burst Example....3802
Figure 16-42. Servicing Peripheral Burst Example PaRAM Configuration....3803
Figure 16-43. Servicing Continuous McASP Data Example....3804
Figure 16-44. Servicing Continuous McASP Data Example PaRAM Configuration....3805
Figure 16-45. Servicing Continuous McASP Data Example Reload PaRAM Configuration....3806
Figure 16-46. Ping-Pong Buffering for McASP Data Example....3807
Figure 16-47. Ping-Pong Buffering for McASP Example PaRAM Configuration....3809
Figure 16-48. Ping-Pong Buffering for McASP Example Pong PaRAM Configuration....3810
Figure 16-49. Ping-Pong Buffering for McASP Example Ping PaRAM Configuration....3810
Figure 16-50. Intermediate Transfer Completion Chaining Example....3812
Figure 16-51. Single Large Block Transfer Example....3813
Figure 16-52. Smaller Packet Data Transfers Example....3813
Figure 17-1. Interrupt Controllers in the Device....3969
Figure 17-2. Interrupts From External Devices....3970
Figure 18-1. Control Module Overview Block Diagram....4017
Figure 18-2. Control Module Environment....4018
Figure 18-3. Control Module Integration....4019
Figure 18-4. Pad Configuration Register Bits....4022
Figure 18-5. Thermal Management Functional Block Diagram....4028
Figure 18-6. Behavior Of The Thermal Alert Logic....4030
Figure 18-7. Behavior Of The Thermal Shutdown Logic....4031
Figure 18-8. PBIAS Cell And Its Connections....4038
Figure 18-9. IRQ_CROSSBAR Module Functional Diagram....4042
Figure 18-10. DMA_CROSSBAR Module Functional Diagram....4045
Figure 18-11. Vref-Generation Cells and Their Controls....4052
Figure 18-12. AVS Class 0 Procedure....4056
Figure 18-13. Combined Firewall Error Interrupt....4058
Figure 18-14. IODELAYCONFIG Integration....4671
Figure 19-1. MAILBOX1 Integration....5108
Figure 19-2. MAILBOX2..13 Integration....5108
Figure 19-3. IVA Mailbox Integration....5111
Figure 19-4. EVE1_MBOX Integration....5112
Figure 19-5. EVE2_MBOX Integration....5112
Figure 19-6. Mailbox Block Diagram....5116
Figure 19-7. Example of Communication....5120
Figure 20-1. System MMU1 Overview....5149
Figure 20-2. System MMU2 Overview....5150
Figure 20-3. System MMU1 Integration....5151
Figure 20-4. System MMU2 Integration....5151
Figure 20-5. MMU Block Diagram....5153
Figure 20-6. Translation Process....5154
Figure 20-7. Translation Hierarchy....5155
Figure 20-8. First-level Descriptor Address Calculation....5155
Figure 20-9. Detailed First-Level Descriptor Address Calculation....5156
Figure 20-10. Section Translation Summary....5157
Figure 20-11. Supersection Translation Summary....5158
Figure 20-12. Two-Level Translation....5159
Figure 20-13. Small Page Translation Summary....5160

Figure 20-14. Large Page Translation Summary....5161
Figure 20-15. TLB Entry Lock Mechanism....5162
Figure 20-16. TLB Entry Structure....5163
Figure 20-17. MMU Global Initialization....5165
Figure 21-1. Spinlock Overview....5188
Figure 21-2. Spinlock Integration....5189
Figure 21-3. Lock Register State Diagram....5191
Figure 21-4. Take and Release Spinlock....5193
Figure 22-1. Timers Overview....5199
Figure 22-2. GP Timers Overview....5200
Figure 22-3. GP Timers External System Interface....5202
Figure 22-4. GP Timer Integration....5204
Figure 22-5. Block Diagram of TIMER3 Through TIMER9 and TIMER11 Through TIMER16....5209
Figure 22-6. Block Diagram of TIMER1, TIMER2 and TIMER10....5210
Figure 22-7. Wake-Up Request Generation....5211
Figure 22-8. Wake-Up Request Generation....5213
Figure 22-9. TCRR Timing Value....5214
Figure 22-10. Block Diagram of the 1-ms Tick Module....5215
Figure 22-11. Capture Wave Example for TCLR[13] CAPT_MODE = 0....5217
Figure 22-12. Capture Wave Example for TCLR[13] CAPT_MODE = 1....5218
Figure 22-13. Timing Diagram of PWM With TCLR[7] SCPWM Bit = 0....5219
Figure 22-14. Timing Diagram of PWM With TCLR[7] SCPWM Bit = 1....5220
Figure 22-15. 32-kHz Synchronized Timer Block Diagram....5249
Figure 22-16. Reset Resynchronization Timing Diagram....5251
Figure 22-17. CONTER_32K Block Diagram....5251
Figure 22-18. Watchdog Timer Block Diagram....5256
Figure 22-19. Watchdog Timer Integration....5258
Figure 22-20. 32-Bit Watchdog Timer Functional Block Diagram....5261
Figure 22-21. Watchdog Timers General Functional View....5262
Figure 23-1. RTC Block Diagram....5280
Figure 23-2. RTC External Signals....5282
Figure 23-3. RTC Module Integration....5283
Figure 23-4. RTC Functional Block Diagram....5285
Figure 23-5. Kick Register State Machine Diagram....5288
Figure 23-6. Flow Control for Updating RTC Registers....5290
Figure 23-7. Compensation Illustration....5291
Figure 24-1. HS I2C Controllers....5317
Figure 24-2. HS I2C and Typical Connections to I2C Devices....5319
Figure 24-3. HS I2C Interface Signals....5319
Figure 24-4. HS I2C Data Transfer....5320
Figure 24-5. HS I2C Bit Transfer on the I2C Bus....5320
Figure 24-6. HS I2C S and P Condition Events....5321
Figure 24-7. HS I2C Data Transfer Formats in F/S Mode....5321
Figure 24-8. HS I2C Data Transfer in HS Mode....5322
Figure 24-9. HS I2C Arbitration Between Master Transmitters....5323
Figure 24-10. HS I2C Clock Generators Synchronization....5324
Figure 24-11. HS I2C Integration....5325
Figure 24-12. HS I2C Block Diagram....5329
Figure 24-13. HS I2C Clock Generation....5329
Figure 24-14. HS I2C Receive FIFO Interrupt Request Generation....5335
Figure 24-15. HS I2C Transmit FIFO Interrupt Request Generation....5335
Figure 24-16. HS I2C Receive FIFO DMA Request Generation....5337
Figure 24-17. HS I2C Transmit FIFO Request Generation (High Threshold)....5337
Figure 24-18. HS I2C Transmit FIFO Request Generation (Low Threshold)....5338
Figure 24-19. HS I2C Setup Procedure....5343
Figure 24-20. HS I2C Master Transmitter Mode, Polling Method, in F/S and HS Modes....5344
Figure 24-21. HS I2C Master Receiver Mode, Polling Method, in F/S and HS Modes....5346
Figure 24-22. HS I2C Master Transmitter Mode, Interrupt Method, in F/S and HS Modes....5347
Figure 24-23. HS I2C Master Receiver Mode, Interrupt Method, in F/S and HS Modes....5349
Figure 24-24. HS I2C Master Transmitter Mode, DMA Method in F/S and HS Modes....5350
Figure 24-25. HS I2C Master Receiver Mode, DMA Method in F/S and HS Modes....5352

Figure 24-26. HS I^{2}C Slave Transmitter/Receiver Mode, Polling........ 5353
Figure 24-27. HS I^{2}C Slave Transmitter/Receiver Mode, Interrupt........ 5354
Figure 24-28. HDQ1W Overview........ 5382
Figure 24-29. HDQ1W Typical Application System Overview........ 5383
Figure 24-30. HDQ Break-Pulse Timing Diagram........ 5384
Figure 24-31. 1-Wire (SDQ) Reset Timing Diagram........ 5384
Figure 24-32. HDQ1W Transmitted Bit Timing........ 5385
Figure 24-33. HDQ/1-Wire Communication Sequence........ 5385
Figure 24-34. HDQ1W Integration........ 5386
Figure 24-35. HDQ1W Block Diagram........ 5388
Figure 24-36. Protocol Registers Description........ 5389
Figure 24-37. UART Overview........ 5400
Figure 24-38. UART Mode Bus System Overview........ 5402
Figure 24-39. UART Frame Data Format........ 5404
Figure 24-40. IrDA System Overview........ 5404
Figure 24-41. IrDA SIR Frame Format........ 5406
Figure 24-42. IrDA SIR Encoding Mechanism........ 5407
Figure 24-43. IrDA SIR Decoding Mechanism........ 5407
Figure 24-44. SIR FF Mode........ 5408
Figure 24-45. MIR Transmit Frame Format........ 5408
Figure 24-46. MIR Baud Rate Adjustment Mechanism........ 5409
Figure 24-47. SIP........ 5410
Figure 24-48. FIR Transmit Frame Format........ 5410
Figure 24-49. CIR System Overview........ 5411
Figure 24-50. CIR Pulse Modulation........ 5412
Figure 24-51. CIR Modulation Duty Cycle........ 5412
Figure 24-52. RC-5 Bit Encoding........ 5413
Figure 24-53. SIRC Bit Encoding........ 5414
Figure 24-54. RC-5 Standard Packet Format........ 5414
Figure 24-55. SIRC Packet Format........ 5414
Figure 24-56. SIRC Bit Transmission Example........ 5415
Figure 24-57. UART/IrDA/CIR Integration........ 5416
Figure 24-58. UART/IrDA/CIR Functional Block Diagram........ 5421
Figure 24-59. FIFO Management Registers........ 5426
Figure 24-60. RX FIFO Interrupt Request Generation........ 5428
Figure 24-61. TX FIFO Interrupt Request Generation........ 5428
Figure 24-62. Receive FIFO DMA Request Generation (32 Characters)........ 5431
Figure 24-63. Transmit FIFO DMA Request Generation (56 Spaces)........ 5431
Figure 24-64. Transmit FIFO DMA Request Generation (8 Spaces)........ 5432
Figure 24-65. Transmit FIFO DMA Request Generation (1 Space)........ 5433
Figure 24-66. Transmit FIFO DMA Request Generation Using Direct TX DMA Threshold Programming. (Threshold = 3; Spaces = 8)........ 5434
Figure 24-67. DMA Transmission........ 5434
Figure 24-68. DMA Reception........ 5435
Figure 24-69. Baud Rate Generation........ 5441
Figure 24-70. Baud Rate Generator........ 5447
Figure 24-71. CIR Mode Block Components........ 5451
Figure 24-72. Multichannel SPI Modules........ 5504
Figure 24-73. McSPI Interface Signals in Master Mode........ 5505
Figure 24-74. McSPI Interface Signals in Slave Mode........ 5506
Figure 24-75. Phase and Polarity Combinations........ 5508
Figure 24-76. Full-Duplex Transfer Format With PHA = 0........ 5509
Figure 24-77. Extended SPI Transfer With a Start-Bit (SBE = 1)........ 5510
Figure 24-78. McSPI Master Mode (Full Duplex)........ 5510
Figure 24-79. McSPI Master Single Mode (Receive Only)........ 5511
Figure 24-80. McSPI Slave Mode (Full Duplex)........ 5511
Figure 24-81. McSPI Slave Single Mode (Transmit Only)........ 5512
Figure 24-82. McSPI Integration........ 5513
Figure 24-83. McSPI Block Diagram........ 5517
Figure 24-84. SPI Full-Duplex Transmission (Example)........ 5519
Figure 24-85. Continuous Transfers With SPIEN[x] Maintained Active (Single-Data-Pin Interface Mode)........ 5521

Figure 24-86. Continuous Transfers With SPIEN[x] Maintained Active (Dual-Data-Pin Interface Mode)............................. 5521
Figure 24-87. CS (SPIEN) Timing Controls.......... 5522
Figure 24-88. Example of McSPI Slave With One Master and Multiple Slave Devices on Channel 0.......... 5525
Figure 24-89. SPI Half-Duplex Transmission (Transmit-Only Slave).......... 5527
Figure 24-90. SPI Half-Duplex Transmission (Receive-Only Slave).......... 5528
Figure 24-91. Buffer Used in Transmit Direction Only.......... 5529
Figure 24-92. Buffer Used in Receive Direction Only.......... 5529
Figure 24-93. Buffer Used for Transmit and Receive Directions.......... 5529
Figure 24-94. Buffer Almost Full Level (AFL).......... 5530
Figure 24-95. Buffer Almost Empty Level (AEL).......... 5531
Figure 24-96. FIFO Mode Transmit-and-Receive With Word Count (Master).......... 5546
Figure 24-97. FIFO Mode Transmit-and-Receive Without Word Count (Master).......... 5547
Figure 24-98. FIFO Mode Transmit-Only (Master).......... 5548
Figure 24-99. FIFO Mode Receive-Only With Word Count (Master).......... 5549
Figure 24-100. FIFO Mode Receive-Only Without Word Count (Master).......... 5550
Figure 24-101. QSPI Overview.......... 5577
Figure 24-102. QSPI Connected to an External Quad-SPI Flash Memory.......... 5578
Figure 24-103. QSPI Integration.......... 5579
Figure 24-104. QSPI Block Diagram.......... 5581
Figure 24-105. SPI_CLKGEN Block.......... 5584
Figure 24-106. SPI Clock Modes.......... 5586
Figure 24-107. Logical Representation of the QSPI Interrupt Generation Scheme.......... 5587
Figure 24-108. McASP Modules Overview.......... 5607
Figure 24-109. McASP Environment.......... 5609
Figure 24-110. Definition of Bit, Word, and Slot.......... 5614
Figure 24-111. Bit Order and Word Alignment Within a Slot Examples.......... 5615
Figure 24-112. Definition of Frame and Frame-Sync Width.......... 5616
Figure 24-113. TDM Format - 6 channel example.......... 5617
Figure 24-114. I2S Format Overview.......... 5618
Figure 24-115. Biphase-Mark Code.......... 5619
Figure 24-116. S/PDIF Subframe Format.......... 5620
Figure 24-117. S/PDIF Frame Format.......... 5620
Figure 24-118. McASP Integration.......... 5622
Figure 24-119. McASP Module Block Diagram.......... 5628
Figure 24-120. Transmit Clock Generator Block Diagram.......... 5630
Figure 24-121. Receive Clock Generator Block Diagram.......... 5632
Figure 24-122. Frame Sync Generator Block Diagram.......... 5633
Figure 24-123. Individual Serializer and Connections Within McASP.......... 5635
Figure 24-124. Transmit Format Unit.......... 5637
Figure 24-125. Receive Format Unit.......... 5640
Figure 24-126. Burst Frame Sync Mode.......... 5643
Figure 24-127. Transmit DMA Event (AXEVT) Generation in TDM Time Slots.......... 5645
Figure 24-128. MPU Service Time Upon Transmit DMA Event (AXEVT).......... 5650
Figure 24-129. CPU Service Time Upon Receive Event (AREVT).......... 5651
Figure 24-130. DMA Transmit and Receive Event in an Audio Example – One Event.......... 5654
Figure 24-131. McASP Audio FIFO (AFIFO) Block Diagram.......... 5655
Figure 24-132. McASP Serializers Operation in Loopback Mode.......... 5660
Figure 24-133. Transmit Clock Failure Detection Circuit Block Diagram.......... 5664
Figure 24-134. Receive Clock Failure Detection Circuit Block Diagram.......... 5665
Figure 24-135. McASP DIT- /TDM- Transmission Polling Method.......... 5676
Figure 24-136. Subsequence – DIT-/TDM- Transmission Startup Procedure.......... 5678
Figure 24-137. McASP Polling Reception Method.......... 5681
Figure 24-138. Subsequence – TDM - Reception Startup Procedure.......... 5683
Figure 24-139. McASP Transmit Interrupt Events Servicing.......... 5686
Figure 24-140. McASP Receive Interrupt Events Servicing.......... 5687
Figure 24-141. McASP Transmit Error Handling.......... 5688
Figure 24-142. McASP Receive Error Handling.......... 5689
Figure 24-143. USB1 Highlight.......... 5747
Figure 24-144. USB2 Highlight.......... 5748
Figure 24-145. USB3 and USB4 Highlight.......... 5749
Figure 24-146. SuperSpeed USB Subsystem Environment.......... 5751

Figure 24-147. SuperSpeed USB Controller Application: USB3.0 DRD....5754
Figure 24-148. SuperSpeed USB Controller Application: USB2.0 DRD (Internal PHY)....5755
Figure 24-149. SuperSpeed USB Controller Application: USB2.0 DRD (ULPI PHY)....5756
Figure 24-150. SuperSpeed USB Subsystem Integration....5757
Figure 24-151. SATA Host Controller Subsystem Overview....5762
Figure 24-152. SATA Subsystem Environment....5766
Figure 24-153. SATA Controller Integration....5767
Figure 24-154. SATA Controller Functional Block Diagram....5769
Figure 24-155. Simplified Schema of Link Dword Processing....5770
Figure 24-156. SATA Data Stream Components....5773
Figure 24-157. SATA Controller Interrupt Propagation Schema....5778
Figure 24-158. Command List Descriptor Structures....5783
Figure 24-159. PCIe Controller Subsystem Overview....5828
Figure 24-160. PCIe Controllers Integration....5831
Figure 24-161. PCIe Controller Functional Block Diagram....5835
Figure 24-162. PCIe D-state (function power state) FSM diagram....5844
Figure 24-163. PCIe Core to PCIe Core Address Mapping and Translation....5856
Figure 24-164. DCAN1 Overview....5998
Figure 24-165. DCAN2 Overview....5999
Figure 24-166. DCAN Typical Application....6000
Figure 24-167. DCAN1 Integration....6002
Figure 24-168. DCAN2 Integration....6003
Figure 24-169. DCAN Block Diagram....6006
Figure 24-170. Error and Status Change Interrupts....6008
Figure 24-171. Message Objects Interrupts....6009
Figure 24-172. Local Power-Down Mode Flow Diagram....6011
Figure 24-173. Software Handling of a FIFO Buffer (Interrupt Driven)....6020
Figure 24-174. Bit Timing....6021
Figure 24-175. The Propagation Time Segment....6022
Figure 24-176. Synchronization on Late and Early Edges....6024
Figure 24-177. Filtering of Short Dominant Spikes....6025
Figure 24-178. Structure of the CAN Core's CAN Protocol Controller....6026
Figure 24-179. Data Transfer Between IF1/IF2 Registers and Message RAM....6029
Figure 24-180. CAN Module General Initialization Flow....6035
Figure 24-181. CAN Bit-Timing Configuration....6036
Figure 24-182. CAN Core in Silent Mode....6039
Figure 24-183. CAN Core in Loopback Mode....6039
Figure 24-184. CAN Core in External Loopback Mode....6040
Figure 24-185. CAN Core in Loop Back Combined With Silent Mode....6041
Figure 24-186. GMAC_SW Overview....6093
Figure 24-187. MII Interface Typical Application....6096
Figure 24-188. RMII Interface Typical Application....6097
Figure 24-189. RGMII Interface Typical Application....6098
Figure 24-190. GMAC_SW Integration....6100
Figure 24-191. GMAC_SW Top Level Block Diagram....6103
Figure 24-192. CPSW_3G Block Diagram....6110
Figure 24-193. The Network Static with AVB....6121
Figure 24-194. AVB Network & PTP Clock Entities....6123
Figure 24-195. IEEE 1722 Packets....6123
Figure 24-196. Cross Time Stamping and Presentation Timestamps....6124
Figure 24-197. AV Stream Queuing/Policing....6126
Figure 24-198. SPF Block Diagram....6140
Figure 24-199. Packet Octets as Stored in the Packet Buffer....6144
Figure 24-200. CPTS Block Diagram....6156
Figure 24-201. Event FIFO Misalignment Condition....6157
Figure 24-202. HW1/4_TSP_PUSH Connection....6158
Figure 24-203. Partial Ethernet-II Frames Showing Register Mapping of EtherTypes for a Simple Frame (1), a Single 1Q Tag Added (2), and Two 1Q Tags Added (3)....6159
Figure 24-204. TX Queue Head Descriptor....6169
Figure 24-205. RX Queue Head Descriptor....6171
Figure 24-206. MLB Overview....6333

Figure 24-207. MLB Sub System Environment....................6334
Figure 24-208. MLB I/O Cells And Their Controls....................6336
Figure 24-209. MLB Sub System Integration....................6338
Figure 24-210. MLBSS Structural Overview....................6340
Figure 24-211. DMA Descriptor Table Endian Options....................6350
Figure 24-212. Ping-Pong System Memory Structure....................6350
Figure 24-213. Single-Packet Mode Memory Space....................6352
Figure 24-214. Multi-Packet Mode System Memory....................6353
Figure 24-215. MLBSS Software and Data Flow Overview....................6354
Figure 25-1. eMMC/SD/SDIOi Overview (i = 1 to 4)....................6382
Figure 25-2. eMMC/SD/SDIOi Controller Connected to an eMMC, SD, or SDIO Card (where i = 1 to 4)....................6385
Figure 25-3. Sequential Read Operation (MMC Cards Only)....................6387
Figure 25-4. Sequential Write Operation (MMC Cards Only)....................6387
Figure 25-5. Multiple Block Read Operation....................6388
Figure 25-6. Multiple Block Write Operation With Card Busy Signal....................6388
Figure 25-7. Command Token Format....................6389
Figure 25-8. Response Token Format (R1, R3, R4, R5, R6, R7)....................6389
Figure 25-9. Response Token Format (R2)....................6389
Figure 25-10. Data Token Format for 1-Bit Transfers....................6390
Figure 25-11. Data Token Format for 4-Bit Transfers....................6390
Figure 25-12. Data Token Format for 8-Bit Transfers....................6391
Figure 25-13. Integration of MMC1 and MMC2 Controllers – Master and Slave Capable....................6393
Figure 25-14. Integration of MMC3 and MMC4 Controllers – Slave Capable Only....................6394
Figure 25-15. eMMC/SD/SDIO Diagram....................6398
Figure 25-16. ADMA Block Diagram Overview....................6408
Figure 25-17. ADMA Finite State-Machine....................6411
Figure 25-18. DMA Receive Mode....................6413
Figure 25-19. DMA Transmit Mode....................6414
Figure 25-20. Buffer Management for a Write....................6416
Figure 25-21. Buffer Management for a Read....................6417
Figure 25-22. Busy Time-Out for R1b, R5b Response Type....................6420
Figure 25-23. Busy Time-Out After Write CRC Status....................6420
Figure 25-24. Write CRC Status Time-Out....................6421
Figure 25-25. Read Data Time-Out....................6421
Figure 25-26. Boot Acknowledge Time-Out When Using CMD0....................6422
Figure 25-27. Boot Acknowledge Time-Out When CMD Line Tied to 0....................6422
Figure 25-28. Auto CMD12 Timings During Write Transfer....................6423
Figure 25-29. Auto CMD12 Timings During Read Transfer....................6423
Figure 25-30. Output Driven on Falling Edge....................6425
Figure 25-31. Output Driven on Rising Edge....................6426
Figure 25-32. Boot Mode Using the CMD0 Timing Diagram....................6427
Figure 25-33. Boot Mode With CMD Line Tied to 0 Timing Diagram....................6427
Figure 25-34. eMMC/SD/SDIO Controller Software Reset Flow....................6431
Figure 25-35. eMMC/SD/SDIO Controller Bus Configuration....................6432
Figure 25-36. eMMC/SD/SDIO Controller Card Identification and Selection – Part 1....................6433
Figure 25-37. eMMC/SD/SDIO Controller Card Identification and Selection – Part 2....................6434
Figure 25-38. eMMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Slave Mode With interrupt....................6436
Figure 25-39. eMMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Mode With Polling....................6437
Figure 25-40. eMMC/SD/SDIO Controller Read/Write Transfer Flow Without DMA and With Polling....................6439
Figure 25-41. eMMC/SD/SDIO Controller Read/Write in CE-ATA Mode....................6440
Figure 25-42. eMMC/SD/SDIO Controller Suspend Flow....................6442
Figure 25-43. eMMC/SD/SDIO Controller Resume Flow....................6443
Figure 25-44. eMMC/SD/SDIO Controller Command Transfer Flow With Polling....................6444
Figure 25-45. eMMC/SD/SDIO Controller Command Transfer Flow With interrupts....................6446
Figure 25-46. eMMC/SD/SDIO Controller Clock Frequency Change Flow....................6447
Figure 25-47. eMMC/SD/SDIO Controller Bus Width Configuration Flow....................6448
Figure 25-48. eMMC/SD/SDIO Power Switching Procedure....................6449
Figure 25-49. eMMC/SD/SDIO Controller Boot Using CMD0....................6451
Figure 25-50. eMMC/SD/SDIO Controller Boot With CMD Line Tied to 0....................6452
Figure 25-51. SDR104/HS200 DLL Tuning Procedure....................6454
Figure 26-1. SATA PHY Subsystem Overview....................6521

Figure 26-2. SATA PHY I/O Signals....6522
Figure 26-3. SATA PHY Subsystem Integration....6525
Figure 26-4. SATA DPLL Clock Generator Overview....6529
Figure 26-5. DPLL_SATA Functional Block Diagram....6531
Figure 26-6. SATA PLL GO Sequence....6533
Figure 26-7. SATA PLL Programming Sequence....6535
Figure 26-8. USB3_PHY Subsystem Overview....6540
Figure 26-9. USB3_PHY I/O Signals....6541
Figure 26-10. USB3_PHY Subsystem Integration....6543
Figure 26-11. USB3_PHY DPLL Clock Generator Overview....6547
Figure 26-12. DPLL_USB_OTG_SS Functional Block Diagram....6550
Figure 26-13. USB3_PHY PLL GO Sequence....6552
Figure 26-14. USB3_PHY PLL Programming Sequence....6555
Figure 26-15. PCIe Controller Subsystem Overview....6578
Figure 26-16. PCIe PHY I/O Signals....6581
Figure 26-17. PCIe PHY Subsystem Integration....6583
Figure 26-18. PCIe PHY Subsystem Block Diagram....6585
Figure 26-19. PCIe PHY Clock Generator Overview....6590
Figure 26-20. DPLL_PCIE_REF Functional Block Diagram....6593
Figure 26-21. DPLL_PCIE_REF Programming Sequence....6596
Figure 26-22. APLL_PCIE Functional Block Diagram....6599
Figure 27-1. General-Purpose Interface Overview....6616
Figure 27-2. General-Purpose Interface Typical Application....6618
Figure 27-3. General-Purpose Interface Used as a Keyboard Interface....6619
Figure 27-4. GPIO1 Signal Connections....6620
Figure 27-5. GPIO2 Through GPIO8 Signal Connections....6621
Figure 27-6. GPIO Integration....6623
Figure 27-7. General-Purpose Interface Block Diagram....6628
Figure 27-8. Synchronous Path....6628
Figure 27-9. Asynchronous Path....6629
Figure 27-10. Interrupt Request Generation....6630
Figure 27-11. Wake-Up Request Generation....6631
Figure 27-12. Wake-Up Event Conditions....6632
Figure 27-13. GPIO_CLEARDATAOUT Register Example....6641
Figure 27-14. Write in GPIO_IRQSTATUS_SET_0 Register Example....6642
Figure 28-1. Keyboard Controller Overview....6661
Figure 28-2. Typical Keyboard Environment....6663
Figure 28-3. Multikey Limitation Example....6665
Figure 28-4. Keyboard Controller Integration....6666
Figure 28-5. Keyboard Controller Block Diagram....6668
Figure 28-6. Functional Modes and Related Interrupt Events....6673
Figure 28-7. Key Coding Registers....6674
Figure 29-1. PWMSS Block Diagram....6693
Figure 29-2. PWMSS External Interface I/Os....6696
Figure 29-3. PWMSS Integration....6698
Figure 29-4. Synchronization between PWMSS1, PWMSS2 and PWMSS3....6702
Figure 29-5. Multiple ePWM Modules....6709
Figure 29-6. Submodules and Signal Connections for an ePWM Module....6710
Figure 29-7. ePWM Submodules and Critical Internal Signal Interconnects....6711
Figure 29-8. ePWM Time-Base Submodule Block Diagram....6715
Figure 29-9. ePWM Time-Base Submodule Signals and Registers....6717
Figure 29-10. ePWM Time-Base Frequency and Period....6719
Figure 29-11. ePWM Time-Base Counter Synchronization Scheme 1....6720
Figure 29-12. ePWM Time-Base Up-Count Mode Waveforms....6722
Figure 29-13. ePWM Time-Base Down-Count Mode Waveforms....6723
Figure 29-14. ePWM Time-Base Up-Down-Count Waveforms, EPWM_TBCTL[13] PHSDIR = 0 Count Down on Synchronization Event....6723
Figure 29-15. ePWM Time-Base Up-Down Count Waveforms, EPWM_TBCTL[13] PHSDIR = 1 Count Up on Synchronization Event....6725
Figure 29-16. ePWM Counter-Compare Submodule....6726
Figure 29-17. ePWM Counter-Compare Submodule Signals and Registers....6726

Figure 29-18. ePWM Counter-Compare Event Waveforms in Up-Count Mode....... 6729
Figure 29-19. ePWM Counter-Compare Events in Down-Count Mode....... 6729
Figure 29-20. ePWM Counter-Compare Events in Up-Down-Count Mode, EPWM_TBCTL[13] PHSDIR = 0 Count Down on Synchronization Event....... 6730
Figure 29-21. ePWM Counter-Compare Events in Up-Down-Count Mode, EPWM_TBCTL[13] PHSDIR = 1 Count Up on Synchronization Event....... 6730
Figure 29-22. ePWM Action-Qualifier Submodule....... 6731
Figure 29-23. ePWM Action-Qualifier Submodule Inputs and Outputs....... 6732
Figure 29-24. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs....... 6733
Figure 29-25. ePWM Up-Down-Count Mode Symmetrical Waveform....... 6736
Figure 29-26. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High....... 6737
Figure 29-27. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low....... 6739
Figure 29-28. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA....... 6741
Figure 29-29. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low....... 6743
Figure 29-30. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary....... 6745
Figure 29-31. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low....... 6747
Figure 29-32. Dead-Band Generator Submodule....... 6749
Figure 29-33. Configuration Options for the ePWM Dead-Band Generator Submodule....... 6750
Figure 29-34. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)....... 6752
Figure 29-35. PWM-Chopper Submodule....... 6753
Figure 29-36. PWM-Chopper Submodule Signals and Registers....... 6754
Figure 29-37. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only....... 6755
Figure 29-38. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses....... 6755
Figure 29-39. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses 6756
Figure 29-40. ePWM Trip-Zone Submodule....... 6757
Figure 29-41. ePWM Trip-Zone Submodule Mode Control Logic....... 6760
Figure 29-42. ePWM Trip-Zone Submodule Interrupt Logic....... 6761
Figure 29-43. ePWM Event-Trigger Submodule....... 6761
Figure 29-44. ePWM Event-Trigger Submodule Inter-Connectivity to Interrupt Controller....... 6762
Figure 29-45. ePWM Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs....... 6763
Figure 29-46. ePWM Event-Trigger Interrupt Generator....... 6765
Figure 29-47. HRPWM System Interface....... 6766
Figure 29-48. Resolution Calculations for Conventionally Generated PWM....... 6767
Figure 29-49. Operating Logic Using MEP....... 6768
Figure 29-50. Required PWM Waveform for a Requested Duty = 40.5%....... 6770
Figure 29-51. Low % Duty Cycle Range Limitation Example When PWM Frequency = 1 MHz....... 6772
Figure 29-52. High % Duty Cycle Range Limitation Example when PWM Frequency = 1 MHz....... 6772
Figure 29-53. Multiple eCAP Modules....... 6803
Figure 29-54. Capture and APWM Modes of Operation....... 6804
Figure 29-55. Capture Function Diagram....... 6805
Figure 29-56. Event Prescale Control....... 6806
Figure 29-57. Prescale Function Waveforms....... 6806
Figure 29-58. eCAP Continuous/One-shot Block Diagram....... 6807
Figure 29-59. eCAP Counter and Synchronization Block Diagram....... 6808
Figure 29-60. Interrupts in eCAP Module....... 6810
Figure 29-61. PWM Waveform Details Of eCAP APWM Mode Operation....... 6811
Figure 29-62. Optical Encoder Disk....... 6824
Figure 29-63. QEP Encoder Output Signal for Forward/Reverse Movement....... 6825
Figure 29-64. Index Pulse Example....... 6825
Figure 29-65. Functional Block Diagram of the eQEP Peripheral....... 6828
Figure 29-66. Functional Block Diagram of Decoder Unit....... 6830
Figure 29-67. Quadrature Decoder State Machine....... 6832
Figure 29-68. Quadrature-clock and Direction Decoding....... 6832
Figure 29-69. Position Counter Reset by Index Pulse for 1000 Line Encoder (QPOSMAX = 3999 or F9Fh)....... 6834
Figure 29-70. Position Counter Underflow/Overflow (QPOSMAX = 4)....... 6835

Figure 29-71. Software Index Marker for 1000-line Encoder (EQEP_QEPCTL[5:4] IEL = 0b01)..6837
Figure 29-72. eQEP Strobe Event Latch (EQEP_QEPCTL[6] SEL = 0b1)..6838
Figure 29-73. eQEP Position-compare Unit.. 6839
Figure 29-74. eQEP Position-compare Event Generation Points...6840
Figure 29-75. eQEP Position-compare Sync Output Pulse Stretcher... 6841
Figure 29-76. eQEP Edge Capture Unit.. 6842
Figure 29-77. Unit Position Event for Low Speed Measurement (EQEP_QCAPCTL[UPPS] = 0010)..................................6843
Figure 29-78. eQEP Edge Capture Unit - Timing Details.. 6843
Figure 29-79. eQEP Watchdog Timer..6844
Figure 29-80. eQEP Unit Time Base... 6845
Figure 29-81. EQEP Interrupt Generation... 6845
Figure 30-1. VCP Highlight..6867
Figure 30-2. VCP1 and VCP2 Integration..6869
Figure 30-3. VCP Block Diagram...6871
Figure 30-4. Convolutional Encoder Example Block Diagram..6874
Figure 30-5. Trellis Diagram for Convolutional Encoder Example Block Diagram..6875
Figure 30-6. Viterbi Processing Unit.. 6876
Figure 30-7. Tailed Traceback Mode... 6876
Figure 30-8. Mixed Traceback Mode - Example with Five Sliding Windows.. 6877
Figure 30-9. Convergent Traceback Mode - Example With Five Sliding Windows...6877
Figure 30-10. Input FIFO (Branch Metrics)..6879
Figure 30-11. Output FIFO (Decisions Data)... 6880
Figure 30-12. Data Source - EDMA (BM = 1).. 6883
Figure 30-13. Data Destination - Kernel for Processing Unit (BM = 1)... 6884
Figure 30-14. Data Source - EDMA (BM = 0).. 6885
Figure 30-15. Data Destination - Kernel for Processing Unit (BM = 0)... 6885
Figure 30-16. Trellis Stage Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 0)... 6886
Figure 30-17. Trellis Stage Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 1)... 6886
Figure 30-18. EDMA Parameters Structure...6891
Figure 31-1. ATL Module Overview... 6914
Figure 31-2. ATL Integration.. 6916
Figure 31-3. ATL Block Diagram..6919
Figure 32-1. Initialization Process..6938
Figure 32-2. Power Supply Connections Example.. 6940
Figure 32-3. Clock, Reset, and Control Environment Overview.. 6943
Figure 32-4. ROM Code Architecture.. 6952
Figure 32-5. ROM Memory Map..6952
Figure 32-6. RAM Memory Map.. 6954
Figure 32-7. Overall Booting Sequence..6957
Figure 32-8. ROM Code Multiprocessor Start-Up Sequence.. 6959
Figure 32-9. Synchronization Phase for UART..6963
Figure 32-10. Synchronization Phase for USB..6964
Figure 32-11. Peripheral Booting Procedure..6967
Figure 32-12. USB Initialization Procedure..6968
Figure 32-13. SATA Flashing Over USB..6969
Figure 32-14. Fast External Boot Procedure... 6977
Figure 32-15. Memory Booting Procedure...6979
Figure 32-16. Image Shadowing 6980
Figure 32-17. NAND Device Detection.. 6985
Figure 32-18. Bad NAND – Invalid Block Detection.. 6986
Figure 32-19. ECC Data Mapping for 2-KiB Page and 8b BCH Encoding.. 6987
Figure 32-20. ECC Data Mapping for 4-KiB Page and 16b BCH Encoding.. 6988
Figure 32-21. eMMC Connection...6990
Figure 32-22. MMC/SD Card Connection..6991
Figure 32-23. eMMC and SD Booting..6992
Figure 32-24. SD/eMMC Detection Procedure (part 1)... 6994
Figure 32-25. SD/eMMC Detection Procedure (part 2)... 6995
Figure 32-26. SD/eMMC Detection Procedure (part 3)... 6996
Figure 32-27. SD/MMC Get Booting File...6998
Figure 32-28. MBR Detection Procedure...7000
Figure 32-29. MBR, Get Partition.. 7001

Figure 32-30. Booting from a Permanently-Attached SSD Device....... 7005
Figure 32-31. SATA Power-on Initialization Sequencing....... 7006
Figure 32-32. Image Formats....... 7009
Figure 32-33. CH Format....... 7010
Figure 33-1. Register Descriptor Tool (RDT)....... 7025
Figure 33-2. MPU and DSP Processor Traces Flow....... 7049

List of Tables

Table 1-1. Device Identification Register Fields....... 268
Table 1-2. DIE_ID....... 268
Table 1-3. DRA75x, DRA74x Part Number Identifier....... 269
Table 1-4. ID_CODE....... 269
Table 1-5. DRA75x, DRA74x ID_CODE Values....... 269
Table 1-6. PROD_ID....... 269
Table 1-7. DEVICE_TYPE....... 269
Table 2-1. L3_MAIN Memory Map....... 274
Table 2-2. L3_INSTR Memory Map....... 277
Table 2-3. L4_CFG Memory Map....... 278
Table 2-4. L4_WKUP Memory Map....... 282
Table 2-5. L4_PER1 Memory Map....... 283
Table 2-6. L4_PER2 Memory Map....... 286
Table 2-7. L4_PER3 Memory Map....... 287
Table 2-8. MPU Memory Map....... 290
Table 2-9. IPU Memory Map....... 292
Table 2-10. DSP Memory Map....... 294
Table 2-11. EVE Memory Map....... 295
Table 2-12. TILER View Memory Map....... 296
Table 3-1. Master Module Standby Mode Settings....... 301
Table 3-2. Master Module Standby Status....... 301
Table 3-3. Master Module Clock Enabling Conditions....... 301
Table 3-4. Module Idle Mode Settings....... 302
Table 3-5. Slave Module Idle Status....... 302
Table 3-6. Slave Module Clock Activity Settings....... 303
Table 3-7. Slave Module Mode Settings in PRCM....... 303
Table 3-8. Slave Module Interface Clock Enabling Conditions....... 304
Table 3-9. Slave Module Functional Clock Enabling Conditions....... 304
Table 3-10. Clock Domain Functional Clock States....... 306
Table 3-11. Clock Domain Interface Clock States....... 306
Table 3-12. Clock Domain Clock States....... 307
Table 3-13. Clock Domain Clock Transition Mode Settings....... 307
Table 3-14. Clock Domain Wake-Up Conditions....... 310
Table 3-15. Clock Domain Sleep Conditions....... 311
Table 3-16. Device Domain Dependencies (Table 1)....... 312
Table 3-17. Device Domain Dependencies (Table 2)....... 312
Table 3-18. States of a Logic Area in a Power Domain....... 317
Table 3-19. States of a Memory Area in a Power Domain....... 318
Table 3-20. Power Domain Wake-Up Conditions....... 319
Table 3-21. Power Domain Sleep Conditions....... 319
Table 3-22. Power Domain Control and Status Registers....... 320
Table 3-23. External Clock Signals....... 330
Table 3-24. External Boot Signals....... 330
Table 3-25. External Reset Signals....... 331
Table 3-26. Voltage Sources....... 332
Table 3-27. PMFW Device-Level Layout....... 334
Table 3-28. PMFW Module Power Domains....... 338
Table 3-29. PMFW Module Reset Signals....... 338
Table 3-30. PMFW Hardware Requests....... 338
Table 3-31. Global Reset Sources....... 342
Table 3-32. Local Reset Sources....... 342
Table 3-33. Modules, Power Domains, and Reset Domains Association....... 343

Table 3-34. Reset Sources for the Reset Domains....347
Table 3-35. Reset Domains Attributes....364
Table 3-36. Internal Clock Sources....384
Table 3-37. PRM Clock Division and Muxing Control....385
Table 3-38. CM_CORE_AON (a) Clock Division and Muxing Control....389
Table 3-39. CM_CORE_AON (b) Clock Division and Muxing Control....391
Table 3-40. CM_CORE_AON_CLKOUTMUX Clock Division and Muxing Control....396
Table 3-41. CM_CORE_AON_TIMER Clock Division and Muxing Control....400
Table 3-42. CM_CORE_AON_MCASP Clock Division and Muxing Control....403
Table 3-43. CLKOUT_M2, CLKOUTX2_M2, and CLKOUTX2_M3 Frequencies With DPLL State....406
Table 3-44. CLKOUTX2_Hmn Frequencies With DPLL State....406
Table 3-45. DPLL Power Modes....408
Table 3-46. DPLL Recalibration Control Parameters....409
Table 3-47. DPLL Power-Down Control Parameters....410
Table 3-48. DPLL_PER Clock Synthesis Parameters....410
Table 3-49. DPLL_PER Clock Output Parameters....411
Table 3-50. DPLL_PER Modes....411
Table 3-51. DPLL_PER Mode Control Parameters....411
Table 3-52. DPLL_PER Recalibration Feature Parameters....411
Table 3-53. DPLL_CORE Clock Synthesis Parameters....412
Table 3-54. DPLL_CORE Clock Output Parameters....413
Table 3-55. DPLL_CORE Modes....413
Table 3-56. DPLL_CORE Mode Control Parameters....413
Table 3-57. DPLL_CORE Recalibration Feature Parameters....413
Table 3-58. DPLL_ABE Clock Synthesis Parameters....414
Table 3-59. DPLL_ABE Clock Output Parameters....414
Table 3-60. DPLL_ABE Modes....415
Table 3-61. DPLL_ABE Mode Control Parameters....415
Table 3-62. DPLL_ABE Recalibration Feature Parameters....415
Table 3-63. DPLL_MPU Clock Synthesis Parameters....416
Table 3-64. DPLL_MPU Clock Output Parameters....416
Table 3-65. DPLL_MPU Modes....417
Table 3-66. DPLL_MPU Mode Control Parameters....417
Table 3-67. DPLL_MPU Recalibration Feature Parameters....417
Table 3-68. DPLL_IVA Clock Synthesis Parameters....418
Table 3-69. DPLL_IVA Clock Output Parameters....418
Table 3-70. DPLL_IVA Modes....418
Table 3-71. DPLL_IVA Mode Control Parameters....418
Table 3-72. DPLL_IVA Recalibration Feature Parameters....419
Table 3-73. DPLL_USB Clock Synthesis Parameters....419
Table 3-74. DPLL_USB Clock Output Parameters....420
Table 3-75. DPLL_USB Modes....420
Table 3-76. DPLL_USB Mode Control Parameters....420
Table 3-77. DPLL_USB Recalibration Feature Parameters....420
Table 3-78. DPLL_EVE Clock Synthesis Parameters....421
Table 3-79. DPLL_EVE Clock Output Parameters....421
Table 3-80. DPLL_EVE Modes....421
Table 3-81. DPLL_EVE Mode Control Parameters....421
Table 3-82. DPLL_EVE Recalibration Feature Parameters....421
Table 3-83. DPLL_DSP Clock Synthesis Parameters....422
Table 3-84. DPLL_DSP Clock Output Parameters....422
Table 3-85. DPLL_DSP Modes....423
Table 3-86. DPLL_DSP Mode Control Parameters....423
Table 3-87. DPLL_DSP Recalibration Feature Parameters....423
Table 3-88. DPLL_GMAC Clock Synthesis Parameters....424
Table 3-89. DPLL_GMAC Clock Output Parameters....424
Table 3-90. DPLL_GMAC Modes....425
Table 3-91. DPLL_GMAC Mode Control Parameters....425
Table 3-92. DPLL_GMAC Recalibration Feature Parameters....425
Table 3-93. DPLL_GPU Clock Synthesis Parameters....426
Table 3-94. DPLL_GPU Clock Output Parameters....426

Table 3-95. DPLL_GPU Modes... 426
Table 3-96. DPLL_GPU Mode Control Parameters... 426
Table 3-97. DPLL_GPU Recalibration Feature Parameters... 427
Table 3-98. DPLL_DDR Clock Synthesis Parameters... 427
Table 3-99. DPLL_DDR Clock Output Parameters... 428
Table 3-100. DPLL_DDR Modes... 428
Table 3-101. DPLL_DDR Mode Control Parameters... 428
Table 3-102. DPLL_DDR Recalibration Feature Parameters... 428
Table 3-103. DPLL_PCIE_REF Clock Synthesis Parameters... 429
Table 3-104. DPLL_PCIE_REF Clock Output Parameters... 429
Table 3-105. DPLL_PCIE_REF Modes... 429
Table 3-106. DPLL_PCIE_REF Mode Control Parameters... 430
Table 3-107. APLL_PCIE Clock Output Parameters... 430
Table 3-108. APLL_PCIE Modes... 430
Table 3-109. APLL_PCIE Mode Control Parameters... 431
Table 3-110. CD_WKUPAON Clock Domain Modes... 431
Table 3-111. CD_WKUPAON Control and Status Parameters... 431
Table 3-112. CD_WKUPAON Wake-Up Dependency Association Parameters... 432
Table 3-113. CD_WKUPAON Modules Clocks Association... 435
Table 3-114. CD_WKUPAON Modules Wake-Up Request... 436
Table 3-115. CD_WKUPAON Modules Clock-Management Modes and Control... 436
Table 3-116. CD_WKUPAON Modules Slave Clock-Management Modes and Control... 436
Table 3-117. CD_DSP1 Clock Domain Modes... 438
Table 3-118. CD_DSP1 Control and Status Parameters... 438
Table 3-119. CD_DSP1 Static Dependency Association Parameters... 438
Table 3-120. CD_DSP1 Dynamic Dependency Association Parameters... 439
Table 3-121. CD_DSP1 Modules Clocks Association... 439
Table 3-122. CD_DSP1 Modules Wake-Up Request... 439
Table 3-123. CD_DSP1 Modules Clock-Management Modes and Control... 439
Table 3-124. CD_DSP1 Modules Slave Clock-Management Modes and Control... 440
Table 3-125. CD_DSP2 Clock Domain Modes... 440
Table 3-126. CD_DSP2 Control and Status Parameters... 440
Table 3-127. CD_DSP2 Static Dependency Association Parameters... 441
Table 3-128. CD_DSP2 Dynamic Dependency Association Parameters... 442
Table 3-129. CD_DSP2 Modules Clocks Association... 442
Table 3-130. CD_DSP2 Modules Wake-Up Request... 442
Table 3-131. CD_DSP2 Modules Clock-Management Modes and Control... 442
Table 3-132. CD_DSP2 Modules Slave Clock-Management Modes and Control... 442
Table 3-133. CD_CUSTEFUSE Clock Domain Modes... 443
Table 3-134. CD_CUSTEFUSE Control and Status Parameters... 443
Table 3-135. CD_CUSTEFUSE Modules Clocks Association... 443
Table 3-136. CD_CUSTEFUSE Modules Wake-Up Request... 444
Table 3-137. CD_CUSTEFUSE Modules Clock-Management Modes and Control... 444
Table 3-138. CD_CUSTEFUSE Modules Slave Clock-Management Modes and Control... 444
Table 3-139. CD_MPU Clock Domain Modes... 444
Table 3-140. CD_MPU Control and Status Parameters... 445
Table 3-141. CD_MPU Static Dependency Association Parameters... 445
Table 3-142. CD_MPU Dynamic Dependency Association Parameters... 446
Table 3-143. CD_MPU Modules Clocks Association... 446
Table 3-144. CD_MPU Modules Wake-Up Request... 446
Table 3-145. CD_MPU Modules Clock-Management Modes and Control... 446
Table 3-146. CD_MPU Modules Slave Clock-Management Modes and Control... 447
Table 3-147. CD_L4PER1 Clock Domain Modes... 447
Table 3-148. CD_L4PER1 Control and Status Parameters... 447
Table 3-149. CD_L4PER1 Dynamic Dependency Association Parameters... 448
Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters... 448
Table 3-151. CD_L4PER1 Modules Clocks Association... 472
Table 3-152. CD_L4PER1 Modules Wake-Up Request... 474
Table 3-153. CD_L4PER1 Modules Clock-Management Modes and Control... 475
Table 3-154. CD_L4PER1 Modules Slave Clock-Management Modes and Control... 477
Table 3-155. CD_L4PER2 Clock Domain Modes... 479

Table 3-156. CD_L4PER2 Control and Status Parameters....479
Table 3-157. CD_L4PER2 Dynamic Dependency Association Parameters....480
Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters....480
Table 3-159. CD_L4PER2 Modules Clocks Association....489
Table 3-160. CD_L4PER2 Modules Wake-Up Request....490
Table 3-161. CD_L4PER2 Modules Clock-Management Modes and Control....491
Table 3-162. CD_L4PER2 Modules Slave Clock-Management Modes and Control....491
Table 3-163. CD_L4PER3 Clock Domain Modes....493
Table 3-164. CD_L4PER3 Control and Status Parameters....493
Table 3-165. CD_L4PER3 Dynamic Dependency Association Parameters....493
Table 3-166. CD_L4PER3 Modules Clocks Association....494
Table 3-167. CD_L4PER3 Modules Wake-Up Request....494
Table 3-168. CD_L4PER3 Modules Clock-Management Modes and Control....494
Table 3-169. CD_L4PER3 Modules Slave Clock-Management Modes and Control....495
Table 3-170. CD_L4SEC Clock Domain Modes....496
Table 3-171. CD_L4SEC Control and Status Parameters....496
Table 3-172. CD_L4SEC Static Dependency Association Parameters....496
Table 3-173. CD_L4SEC Dynamic Dependency Association Parameters....497
Table 3-174. CD_L4SEC Modules Clocks Association....497
Table 3-175. CD_L4SEC Modules Clock-Management Modes and Control....497
Table 3-176. CD_L4SEC Modules Slave Clock-Management Modes and Control....498
Table 3-177. CD_L3INIT Clock Domain Modes....499
Table 3-178. CD_L3INIT Control and Status Parameters....499
Table 3-179. CD_L3INIT Static Dependency Association Parameters....499
Table 3-180. CD_L3INIT Dynamic Dependency Association Parameters....500
Table 3-181. CD_L3INIT Wake-Up Dependency Association Parameters....500
Table 3-182. CD_L3INIT Modules Clocks Association....505
Table 3-183. CD_L3INIT Modules Wake-Up Request....506
Table 3-184. CD_L3INIT Modules Clock-Management Modes and Control....506
Table 3-185. CD_L3INIT Modules Slave Clock-Management Modes and Control....507
Table 3-186. CD_IVA Clock Domain Modes....508
Table 3-187. CD_IVA Control and Status Parameters....509
Table 3-188. CD_IVA Static Dependency Association Parameters....509
Table 3-189. CD_IVA Dynamic Dependency Association Parameters....509
Table 3-190. CD_IVA Modules Clocks Association....509
Table 3-191. CD_IVA Modules Wake-Up Request....509
Table 3-192. CD_IVA Modules Clock-Management Modes and Control....509
Table 3-193. CD_IVA Modules Slave Clock-Management Modes and Control....510
Table 3-194. CD_GPU Clock Domain Modes....510
Table 3-195. CD_GPU Control and Status Parameters....510
Table 3-196. CD_GPU Static Dependency Association Parameters....511
Table 3-197. CD_GPU Dynamic Dependency Association Parameters....511
Table 3-198. CD_GPU Modules Clocks Association....511
Table 3-199. CD_GPU Modules Wake-Up Request....511
Table 3-200. CD_GPU Modules Clock-Management Modes and Control....511
Table 3-201. CD_GPU Modules Slave Clock-Management Modes and Control....512
Table 3-202. CD_EMU Clock Domain Modes....512
Table 3-203. CD_EMU Control and Status Parameters....513
Table 3-204. CD_EMU Dynamic Dependency Association Parameters....513
Table 3-205. CD_EMU Modules Clocks Association....513
Table 3-206. CD_EMU Modules Wake-Up Request....513
Table 3-207. CD_DSS Clock Domain Modes....514
Table 3-208. CD_DSS Control and Status Parameters....514
Table 3-209. CD_DSS Static Dependency Association Parameters....515
Table 3-210. CD_DSS Dynamic Dependency Association Parameters....515
Table 3-211. CD_DSS Wake-Up Dependency Association Parameters....515
Table 3-212. CD_DSS Modules Clocks Association....519
Table 3-213. CD_DSS Modules Wake-Up Request....519
Table 3-214. CD_DSS Modules Clock-Management Modes and Control....519
Table 3-215. CD_DSS Modules Slave Clock-Management Modes and Control....520
Table 3-216. CD_L4_CFG Clock Domain Modes....520

Table 3-217. CD_L4_CFG Control and Status Parameters....521
Table 3-218. CD_L4_CFG Dynamic Dependency Association Parameters....521
Table 3-219. CD_L4_CFG Modules Clocks Association....521
Table 3-220. CD_L4_CFG Modules Wake-Up Request....521
Table 3-221. CD_L4_CFG Modules Clock-Management Modes and Control....522
Table 3-222. CD_L4_CFG Modules Slave Clock-Management Modes and Control....523
Table 3-223. CD_L3_INSTR Clock Domain Modes....524
Table 3-224. CD_L3_INSTR Control and Status Parameters....524
Table 3-225. CD_L3_INSTR Modules Clocks Association....524
Table 3-226. CD_L3_INSTR Modules Wake-Up Request....525
Table 3-227. CD_L3_INSTR Modules Clock-Management Modes and Control....525
Table 3-228. CD_L3_INSTR Modules Slave Clock-Management Modes and Control....525
Table 3-229. CD_L3_MAIN1 Clock Domain Modes....526
Table 3-230. CD_L3_MAIN1 Control and Status Parameters....527
Table 3-231. CD_L3_MAIN1 Dynamic Dependency Association Parameters....527
Table 3-232. CD_L3_MAIN1 Modules Clocks Association....528
Table 3-233. CD_L3_MAIN1 Modules Wake-Up Request....528
Table 3-234. CD_L3_MAIN1 Modules Clock-Management Modes and Control....528
Table 3-235. CD_L3_MAIN1 Modules Slave Clock-Management Modes and Control....529
Table 3-236. CD_EMIF Clock Domain Modes....530
Table 3-237. CD_EMIF Control and Status Parameters....530
Table 3-238. CD_EMIF Modules Clocks Association....531
Table 3-239. CD_EMIF Modules Wake-Up Request....531
Table 3-240. CD_EMIF Modules Clock-Management Modes and Control....531
Table 3-241. CD_EMIF Modules Slave Clock-Management Modes and Control....532
Table 3-242. CD_IPU Clock Domain Modes....532
Table 3-243. CD_IPU Control and Status Parameters....532
Table 3-244. CD_IPU Static Dependency Association Parameters....533
Table 3-245. CD_IPU Dynamic Dependency Association Parameters....534
Table 3-246. CD_IPU Modules Clocks Association....534
Table 3-247. CD_IPU Modules Wake-Up Request....535
Table 3-248. CD_IPU Modules Clock-Management Modes and Control....535
Table 3-249. CD_IPU Modules Slave Clock-Management Modes and Control....535
Table 3-250. CD_IPU1 Clock Domain Modes....536
Table 3-251. CD_IPU1 Control and Status Parameters....536
Table 3-252. CD_IPU1 Static Dependency Association Parameters....537
Table 3-253. CD_IPU1 Dynamic Dependency Association Parameters....538
Table 3-254. CD_IPU1 Modules Clocks Association....538
Table 3-255. CD_IPU1 Modules Wake-Up Request....538
Table 3-256. CD_IPU1 Modules Clock-Management Modes and Control....538
Table 3-257. CD_IPU1 Modules Slave Clock-Management Modes and Control....539
Table 3-258. CD_IPU2 Clock Domain Modes....539
Table 3-259. CD_IPU2 Control and Status Parameters....539
Table 3-260. CD_IPU2 Static Dependency Association Parameters....539
Table 3-261. CD_IPU2 Dynamic Dependency Association Parameters....541
Table 3-262. CD_IPU2 Modules Clocks Association....541
Table 3-263. CD_IPU2 Modules Wake-Up Request....541
Table 3-264. CD_IPU2 Modules Clock-Management Modes and Control....541
Table 3-265. CD_IPU2 Modules Slave Clock-Management Modes and Control....541
Table 3-266. CD_DMA Clock Domain Modes....542
Table 3-267. CD_DMA Control and Status Parameters....542
Table 3-268. CD_DMA Static Dependency Association Parameters....542
Table 3-269. CD_DMA Dynamic Dependency Association Parameters....543
Table 3-270. CD_DMA Modules Clocks Association....543
Table 3-271. CD_DMA Modules Wake-Up Request....543
Table 3-272. CD_DMA Modules Clock-Management Modes and Control....543
Table 3-273. CD_DMA Modules Slave Clock-Management Modes and Control....544
Table 3-274. CD_ATL Clock Domain Modes....544
Table 3-275. CD_ATL Control and Status Parameters....544
Table 3-276. CD_ATL Modules Clocks Association....545
Table 3-277. CD_ATL Modules Wake-Up Request....545

Table 3-278. CD_ATL Modules Clock-Management Modes and Control....... 545
Table 3-279. CD_ATL Modules Slave Clock-Management Modes and Control....... 545
Table 3-280. CD_CAM Clock Domain Modes....... 546
Table 3-281. CD_CAM Control and Status Parameters....... 546
Table 3-282. CD_CAM Static Dependency Association Parameters....... 546
Table 3-283. CD_CAM Modules Clocks Association....... 547
Table 3-284. CD_CAM Modules Wake-Up Request....... 547
Table 3-285. CD_CAM Modules Clock-Management Modes and Control....... 547
Table 3-286. CD_CAM Modules Slave Clock-Management Modes and Control....... 548
Table 3-287. CD_GMAC Clock Domain Modes....... 549
Table 3-288. CD_CAM Control and Status Parameters....... 549
Table 3-289. CD_GMAC Static Dependency Association Parameters....... 549
Table 3-290. CD_GMAC Dynamic Dependency Association Parameters....... 549
Table 3-291. CD_GMAC Modules Clocks Association“....... 549
Table 3-292. CD_GMAC Modules Wake-Up Request....... 550
Table 3-293. CD_GMAC Modules Clock-Management Modes and Control....... 550
Table 3-294. CD_GMAC Modules Slave Clock-Management Modes and Control....... 550
Table 3-295. CD_VPE Clock Domain Modes....... 551
Table 3-296. CD_VPE Control and Status Parameters....... 551
Table 3-297. CD_VPE Static Dependency Association Parameters....... 551
Table 3-298. CD_VPE Wake-Up Dependency Association Parameters....... 552
Table 3-299. CD_VPE Modules Clocks Association....... 552
Table 3-300. CD_VPE Modules Wake-Up Request....... 552
Table 3-301. CD_VPE Modules Clock-Management Modes and Control....... 552
Table 3-302. CD_VPE Modules Slave Clock-Management Modes and Control....... 553
Table 3-303. CD_EVE1 Clock Domain Modes....... 553
Table 3-304. CD_EVE1 Control and Status Parameters....... 553
Table 3-305. CD_EVE1 Static Dependency Association Parameters....... 554
Table 3-306. CD_EVE1 Wake-Up Dependency Association Parameters....... 554
Table 3-307. CD_EVE1 Modules Clocks Association....... 554
Table 3-308. CD_EVE1 Modules Wake-Up Request....... 554
Table 3-309. CD_EVE1 Modules Clock-Management Modes and Control....... 555
Table 3-310. CD_EVE1 Modules Slave Clock-Management Modes and Control....... 555
Table 3-311. CD_EVE2 Clock Domain Modes....... 555
Table 3-312. CD_EVE2 Control and Status Parameters....... 555
Table 3-313. CD_EVE2 Static Dependency Association Parameters....... 556
Table 3-314. CD_EVE2 Wake-Up Dependency Association Parameters....... 556
Table 3-315. CD_EVE2 Modules Clocks Association....... 557
Table 3-316. CD_EVE2 Modules Wake-Up Request....... 557
Table 3-317. CD_EVE2 Modules Clock-Management Modes and Control....... 557
Table 3-318. CD_EVE2 Modules Slave Clock-Management Modes and Control....... 557
Table 3-319. CD_RTC Clock Domain Modes....... 558
Table 3-320. CD_RTC Control and Status Parameters....... 558
Table 3-321. CD_RTC Wake-Up Dependency Association Parameters....... 558
Table 3-322. CD_RTC Modules Clocks Association....... 559
Table 3-323. CD_RTC Modules Wake-Up Request....... 559
Table 3-324. CD_RTC Modules Clock-Management Modes and Control....... 559
Table 3-325. CD_RTC Modules Slave Clock-Management Modes and Control....... 560
Table 3-326. CD_PCIE Clock Domain Modes....... 560
Table 3-327. CD_PCIE Control and Status Parameters....... 560
Table 3-328. CD_PCIE Static Dependency Association Parameters....... 561
Table 3-329. CD_PCIE Wake-Up Dependency Association Parameters....... 562
Table 3-330. CD_PCIE Modules Clocks Association....... 563
Table 3-331. CD_PCIE Modules Wake-Up Request....... 563
Table 3-332. CD_PCIE Modules Clock-Management Modes and Control....... 563
Table 3-333. CD_PCIE Modules Slave Clock-Management Modes and Control....... 564
Table 3-334. PD_WKUPAON Modules Power Attributes....... 565
Table 3-335. PD_WKUPAON Memory Area Power Modes....... 566
Table 3-336. PD_DSP1 Modules Power Attributes....... 566
Table 3-337. PD_DSP1 Logic Area Power Modes....... 567
Table 3-338. PD_DSP1 Memory Area Power Modes....... 567

Table 3-339. PD_DSP1 Power Modes Control Parameters.......567
Table 3-340. PD_DSP1 Power Modes Status Parameters.......567
Table 3-341. PD_DSP2 Modules Power Attributes.......568
Table 3-342. PD_DSP2 Logic Area Power Modes.......568
Table 3-343. PD_DSP2 Memory Area Power Modes.......568
Table 3-344. PD_DSP2 Power Modes Control Parameters.......568
Table 3-345. PD_DSP2 Power Modes Status Parameters.......569
Table 3-346. PD_CUSTEFUSE Modules Power Attributes.......569
Table 3-347. PD_CUSTEFUSE Logic Area Power Modes.......569
Table 3-348. PD_CUSTEFUSE Power Modes Control Parameters.......570
Table 3-349. PD_CUSTEFUSE Power Modes Status Parameters.......570
Table 3-350. PD_MPU Module Power Attributes.......570
Table 3-351. PD_MPU Logic Area Power Modes.......570
Table 3-352. PD_MPU Memory Area Power Modes.......571
Table 3-353. PD_MPU Power Modes Control Parameters.......571
Table 3-354. PD_MPU Power Mode Status Parameters.......571
Table 3-355. MPU Allowed Low-Power Mode.......572
Table 3-356. PD_IPU Module Power Attributes.......572
Table 3-357. PD_IPU Logic Area Power Modes.......573
Table 3-358. PD_IPU Memory Area Power Modes.......573
Table 3-359. PD_IPU Power Modes Control Parameters.......573
Table 3-360. PD_IPU Power Mode Status Parameters.......573
Table 3-361. PD_L3INIT Modules Power Attributes.......574
Table 3-362. PD_L3INIT Logic Area Power Modes.......574
Table 3-363. PD_L3INIT Memory Area Power Modes.......575
Table 3-364. PD_L3INIT Power Modes Control Parameters.......575
Table 3-365. PD_L3INIT Power Modes Status Parameters.......576
Table 3-366. PD_L4PER Modules Power Attributes.......576
Table 3-367. PD_L4PER Logic Area Power Modes.......578
Table 3-368. PD_L4PER Memory Area Power Modes.......579
Table 3-369. PD_L4PER Power Modes Control Parameters.......579
Table 3-370. PD_L4PER Power Modes Status Parameters.......579
Table 3-371. PD_IVA Modules Power Attributes.......580
Table 3-372. PD_IVA Logic Area Power Modes.......580
Table 3-373. PD_IVA Memory Area Power Modes.......580
Table 3-374. PD_IVA Power Modes Control Parameters.......581
Table 3-375. PD_IVA Power Modes Status Parameters.......581
Table 3-376. PD_GPU Modules Power Attributes.......582
Table 3-377. PD_GPU Logic Area Power Modes.......582
Table 3-378. PD_GPU Memory Area Power Modes.......582
Table 3-379. PD_GPU Power Modes Control Parameters.......582
Table 3-380. PD_GPU Power Mode Status Parameters.......582
Table 3-381. PD_EMU Modules Power Attributes.......583
Table 3-382. PD_EMU Logic Area Power Modes.......583
Table 3-383. PD_EMU Memory Area Power Modes.......583
Table 3-384. PD_EMU Power Modes Control Parameters.......583
Table 3-385. PD_EMU Power Modes Status Parameters.......584
Table 3-386. PD_DSS Modules Power Attributes.......584
Table 3-387. PD_DSS Logic Area Power Modes.......584
Table 3-388. PD_DSS Memory Area Power Modes.......584
Table 3-389. PD_DSS Power Modes Control Parameters.......585
Table 3-390. PD_DSS Power Modes Status Parameters.......585
Table 3-391. PD_CORE Modules Power Attributes.......586
Table 3-392. PD_CORE Logic Area Power Modes.......587
Table 3-393. PD_CORE Memory Area Power Modes.......588
Table 3-394. PD_CORE Power Modes Control Parameters.......588
Table 3-395. PD_CORE Power Mode Status Parameters.......589
Table 3-396. PD_CAM Modules Power Attributes.......589
Table 3-397. PD_CAM Logic Area Power Modes.......590
Table 3-398. PD_CAM Memory Area Power Modes.......590
Table 3-399. PD_CAM Power Mode Control Parameters.......590

Table 3-400. PD_CAM Power Modes Status Parameters....................590
Table 3-401. PD_MPUAON Modules Power Attributes....................591
Table 3-402. PD_MMAON Module Power Attributes....................591
Table 3-403. PD_COREAON Module Power Attributes....................591
Table 3-404. PD_VPE Modules Power Attributes....................592
Table 3-405. PD_VPE Logic Area Power Modes....................592
Table 3-406. PD_VPE Memory Area Power Modes....................592
Table 3-407. PD_VPE Power Modes Control Parameters....................593
Table 3-408. PD_VPE Power Modes Status Parameters....................593
Table 3-409. PD_EVE1 Modules Power Attributes....................593
Table 3-410. PD_EVE1 Logic Area Power Modes....................593
Table 3-411. PD_EVE1 Memory Area Power Modes....................594
Table 3-412. PD_EVE1 Power Modes Control Parameters....................594
Table 3-413. PD_EVE1 Power Modes Status Parameters....................594
Table 3-414. PD_EVE2 Modules Power Attributes....................594
Table 3-415. PD_EVE2 Logic Area Power Modes....................595
Table 3-416. PD_EVE2 Memory Area Power Modes....................595
Table 3-417. PD_EVE2 Power Modes Control Parameters....................595
Table 3-418. PD_EVE2 Power Modes Status Parameters....................595
Table 3-419. PD_RTC Modules Power Attributes....................596
Table 3-420. PD_RTC Logic Area Power Modes....................596
Table 3-421. Wake-Up Sources During Device Low Power Mode....................603
Table 3-422. Global Initialization of Surrounding Modules....................607
Table 3-423. DPLL Global Initialization....................607
Table 3-424. DPLL Recalibration Parameter Configuration....................607
Table 3-425. DPLL Synthesized Clock Parameter Configuration....................607
Table 3-426. DPLL Output Clock Parameter Configuration....................608
Table 3-427. Register Call Summary for Sequence – DPLL Output Frequency Change....................610
Table 3-428. Subprocess Call Summary for Sequence – DPLL Output Frequency Change....................610
Table 3-429. Global Initialization of Surrounding Modules....................610
Table 3-430. Clock Domain Global Initialization....................610
Table 3-431. Slave Module Clock-Management Parameter Configuration....................611
Table 3-432. Clock Domain Sleep Transition and Troubleshooting....................611
Table 3-433. Enable/Disable Software-Programmable Static Dependency....................611
Table 3-434. Power Domain Global Initialization....................612
Table 3-435. Forced Memory Area State Change With Power Domain ON....................612
Table 3-436. Forced Power Domain Low-Power State Transition....................612
Table 3-437. PRCM L4_CFG Instance Summary....................613
Table 3-438. PRCM L4_WKUP Instance Summary....................614
Table 3-439. CM_CORE_AON__CKGEN Registers Mapping Summary....................614
Table 3-440. CM_CLKSEL_CORE....................616
Table 3-441. CM_CLKSEL_ABE....................617
Table 3-442. CM_DLL_CTRL....................617
Table 3-443. CM_CLKMODE_DPLL_CORE....................617
Table 3-444. CM_IDLEST_DPLL_CORE....................618
Table 3-445. CM_AUTOIDLE_DPLL_CORE....................619
Table 3-446. CM_CLKSEL_DPLL_CORE....................619
Table 3-447. CM_DIV_M2_DPLL_CORE....................620
Table 3-448. CM_DIV_H12_DPLL_CORE....................621
Table 3-449. CM_DIV_H13_DPLL_CORE....................621
Table 3-450. CM_DIV_H14_DPLL_CORE....................622
Table 3-451. CM_DIV_H22_DPLL_CORE....................622
Table 3-452. CM_DIV_H23_DPLL_CORE....................623
Table 3-453. CM_DIV_H24_DPLL_CORE....................623
Table 3-454. CM_CLKMODE_DPLL_MPU....................624
Table 3-455. CM_IDLEST_DPLL_MPU....................625
Table 3-456. CM_AUTOIDLE_DPLL_MPU....................625
Table 3-457. CM_CLKSEL_DPLL_MPU....................626
Table 3-458. CM_DIV_M2_DPLL_MPU....................627
Table 3-459. CM_BYPCLK_DPLL_MPU....................627
Table 3-460. CM_CLKMODE_DPLL_IVA....................627

Table 3-461. CM_IDLEST_DPLL_IVA.. 628
Table 3-462. CM_AUTOIDLE_DPLL_IVA...629
Table 3-463. CM_CLKSEL_DPLL_IVA...630
Table 3-464. CM_DIV_M2_DPLL_IVA..630
Table 3-465. CM_BYPCLK_DPLL_IVA...631
Table 3-466. CM_CLKMODE_DPLL_ABE.. 631
Table 3-467. CM_IDLEST_DPLL_ABE..632
Table 3-468. CM_AUTOIDLE_DPLL_ABE.. 633
Table 3-469. CM_CLKSEL_DPLL_ABE.. 633
Table 3-470. CM_DIV_M2_DPLL_ABE... 634
Table 3-471. CM_DIV_M3_DPLL_ABE... 634
Table 3-472. CM_CLKMODE_DPLL_DDR..635
Table 3-473. CM_IDLEST_DPLL_DDR...636
Table 3-474. CM_AUTOIDLE_DPLL_DDR..636
Table 3-475. CM_CLKSEL_DPLL_DDR..637
Table 3-476. CM_DIV_M2_DPLL_DDR...638
Table 3-477. CM_DIV_H11_DPLL_DDR... 638
Table 3-478. CM_CLKMODE_DPLL_DSP.. 639
Table 3-479. CM_IDLEST_DPLL_DSP... 640
Table 3-480. CM_AUTOIDLE_DPLL_DSP.. 640
Table 3-481. CM_CLKSEL_DPLL_DSP.. 641
Table 3-482. CM_DIV_M2_DPLL_DSP... 642
Table 3-483. CM_DIV_M3_DPLL_DSP... 642
Table 3-484. CM_BYPCLK_DPLL_DSP..643
Table 3-485. CM_SHADOW_FREQ_CONFIG1... 643
Table 3-486. CM_SHADOW_FREQ_CONFIG2... 644
Table 3-487. CM_DYN_DEP_PRESCAL..645
Table 3-488. CM_CLKMODE_DPLL_EVE.. 645
Table 3-489. CM_IDLEST_DPLL_EVE..646
Table 3-490. CM_AUTOIDLE_DPLL_EVE.. 647
Table 3-491. CM_CLKSEL_DPLL_EVE.. 648
Table 3-492. CM_DIV_M2_DPLL_EVE... 648
Table 3-493. CM_BYPCLK_DPLL_EVE.. 649
Table 3-494. CM_CLKMODE_DPLL_GMAC...649
Table 3-495. CM_IDLEST_DPLL_GMAC.. 650
Table 3-496. CM_AUTOIDLE_DPLL_GMAC...651
Table 3-497. CM_CLKSEL_DPLL_GMAC...651
Table 3-498. CM_DIV_M2_DPLL_GMAC..652
Table 3-499. CM_DIV_M3_DPLL_GMAC..653
Table 3-500. CM_DIV_H11_DPLL_GMAC.. 653
Table 3-501. CM_DIV_H12_DPLL_GMAC.. 653
Table 3-502. CM_DIV_H13_DPLL_GMAC.. 654
Table 3-503. CM_CLKMODE_DPLL_GPU..654
Table 3-504. CM_IDLEST_DPLL_GPU... 655
Table 3-505. CM_AUTOIDLE_DPLL_GPU..656
Table 3-506. CM_CLKSEL_DPLL_GPU..656
Table 3-507. CM_DIV_M2_DPLL_GPU...657
Table 3-508. CM_CORE_AON__DSP1 Registers Mapping Summary... 658
Table 3-509. CM_DSP1_CLKSTCTRL..658
Table 3-510. Register Call Summary for Register CM_DSP1_CLKSTCTRL.. 659
Table 3-511. CM_DSP1_STATICDEP...659
Table 3-512. Register Call Summary for Register CM_DSP1_STATICDEP..661
Table 3-513. CM_DSP1_DYNAMICDEP... 661
Table 3-514. Register Call Summary for Register CM_DSP1_DYNAMICDEP... 662
Table 3-515. CM_DSP1_DSP1_CLKCTRL... 662
Table 3-516. Register Call Summary for Register CM_DSP1_DSP1_CLKCTRL..663
Table 3-517. CM_CORE_AON__DSP2 Registers Mapping Summary... 663
Table 3-518. CM_DSP2_CLKSTCTRL..663
Table 3-519. Register Call Summary for Register CM_DSP2_CLKSTCTRL.. 664
Table 3-520. CM_DSP2_STATICDEP.. 664
Table 3-521. Register Call Summary for Register CM_DSP2_STATICDEP..666

Table 3-522. CM_DSP2_DYNAMICDEP....666
Table 3-523. Register Call Summary for Register CM_DSP2_DYNAMICDEP....667
Table 3-524. CM_DSP2_DSP2_CLKCTRL....667
Table 3-525. Register Call Summary for Register CM_DSP2_DSP2_CLKCTRL....667
Table 3-526. CM_CORE_AON__EVE1 Registers Mapping Summary....668
Table 3-527. CM_EVE1_CLKSTCTRL....668
Table 3-528. Register Call Summary for Register CM_EVE1_CLKSTCTRL....669
Table 3-529. CM_EVE1_STATICDEP....669
Table 3-530. Register Call Summary for Register CM_EVE1_STATICDEP....670
Table 3-531. CM_EVE1_EVE1_CLKCTRL....670
Table 3-532. Register Call Summary for Register CM_EVE1_EVE1_CLKCTRL....670
Table 3-533. CM_CORE_AON__EVE2 Registers Mapping Summary....671
Table 3-534. CM_EVE2_CLKSTCTRL....671
Table 3-535. Register Call Summary for Register CM_EVE2_CLKSTCTRL....671
Table 3-536. CM_EVE2_STATICDEP....672
Table 3-537. Register Call Summary for Register CM_EVE2_STATICDEP....672
Table 3-538. CM_EVE2_EVE2_CLKCTRL....672
Table 3-539. Register Call Summary for Register CM_EVE2_EVE2_CLKCTRL....673
Table 3-540. CM_CORE_AON__INSTR Registers Mapping Summary....673
Table 3-541. CMI_IDENTICATION....674
Table 3-542. Register Call Summary for Register CMI_IDENTICATION....674
Table 3-543. CMI_SYS_CONFIG....674
Table 3-544. Register Call Summary for Register CMI_SYS_CONFIG....674
Table 3-545. CMI_STATUS....674
Table 3-546. Register Call Summary for Register CMI_STATUS....675
Table 3-547. CMI_CONFIGURATION....675
Table 3-548. Register Call Summary for Register CMI_CONFIGURATION....675
Table 3-549. CMI_CLASS_FILTERING....675
Table 3-550. Register Call Summary for Register CMI_CLASS_FILTERING....676
Table 3-551. CMI_TRIGGERING....676
Table 3-552. Register Call Summary for Register CMI_TRIGGERING....677
Table 3-553. CMI_SAMPLING....677
Table 3-554. Register Call Summary for Register CMI_SAMPLING....677
Table 3-555. CM_CORE_AON__IPU Registers Mapping Summary....677
Table 3-556. CM_IPU1_CLKSTCTRL....678
Table 3-557. Register Call Summary for Register CM_IPU1_CLKSTCTRL....678
Table 3-558. CM_IPU1_STATICDEP....679
Table 3-559. Register Call Summary for Register CM_IPU1_STATICDEP....681
Table 3-560. CM_IPU1_DYNAMICDEP....681
Table 3-561. Register Call Summary for Register CM_IPU1_DYNAMICDEP....681
Table 3-562. CM_IPU1_IPU1_CLKCTRL....681
Table 3-563. Register Call Summary for Register CM_IPU1_IPU1_CLKCTRL....682
Table 3-564. CM_IPU_CLKSTCTRL....682
Table 3-565. Register Call Summary for Register CM_IPU_CLKSTCTRL....684
Table 3-566. CM_IPU_MCASP1_CLKCTRL....684
Table 3-567. Register Call Summary for Register CM_IPU_MCASP1_CLKCTRL....686
Table 3-568. CM_IPU_TIMER5_CLKCTRL....686
Table 3-569. Register Call Summary for Register CM_IPU_TIMER5_CLKCTRL....687
Table 3-570. CM_IPU_TIMER6_CLKCTRL....687
Table 3-571. Register Call Summary for Register CM_IPU_TIMER6_CLKCTRL....688
Table 3-572. CM_IPU_TIMER7_CLKCTRL....688
Table 3-573. Register Call Summary for Register CM_IPU_TIMER7_CLKCTRL....689
Table 3-574. CM_IPU_TIMER8_CLKCTRL....689
Table 3-575. Register Call Summary for Register CM_IPU_TIMER8_CLKCTRL....690
Table 3-576. CM_IPU_I2C5_CLKCTRL....691
Table 3-577. Register Call Summary for Register CM_IPU_I2C5_CLKCTRL....691
Table 3-578. CM_IPU_UART6_CLKCTRL....691
Table 3-579. Register Call Summary for Register CM_IPU_UART6_CLKCTRL....692
Table 3-580. CM_CORE_AON__MPU Registers Mapping Summary....692
Table 3-581. CM_MPU_CLKSTCTRL....692
Table 3-582. CM_MPU_STATICDEP....693

Table 3-583. CM_MPU_DYNAMICDEP.......... 695
Table 3-584. CM_MPU_MPU_CLKCTRL.......... 696
Table 3-585. CM_MPU_MPU_MPU_DBG_CLKCTRL.......... 697
Table 3-586. CM_CORE_AON__OCP_SOCKET Registers Mapping Summary.......... 697
Table 3-587. REVISION_CM_CORE_AON.......... 698
Table 3-588. CM_CM_CORE_AON_PROFILING_CLKCTRL.......... 698
Table 3-589. CM_CORE_AON_DEBUG_OUT.......... 699
Table 3-590. CM_CORE_AON_DEBUG_CFG0.......... 699
Table 3-591. CM_CORE_AON_DEBUG_CFG1.......... 699
Table 3-592. CM_CORE_AON_DEBUG_CFG2.......... 700
Table 3-593. CM_CORE_AON_DEBUG_CFG3.......... 700
Table 3-594. CM_CORE_AON__RESTORE Registers Mapping Summary.......... 700
Table 3-595. CM_CLKSEL_CORE_RESTORE.......... 701
Table 3-596. CM_DIV_M2_DPLL_CORE_RESTORE.......... 701
Table 3-597. CM_DIV_H12_DPLL_CORE_RESTORE.......... 701
Table 3-598. CM_DIV_H13_DPLL_CORE_RESTORE.......... 702
Table 3-599. CM_DIV_H14_DPLL_CORE_RESTORE.......... 702
Table 3-600. CM_DIV_H22_DPLL_CORE_RESTORE.......... 702
Table 3-601. CM_DIV_H23_DPLL_CORE_RESTORE.......... 702
Table 3-602. CM_DIV_H24_DPLL_CORE_RESTORE.......... 703
Table 3-603. CM_CLKSEL_DPLL_CORE_RESTORE.......... 703
Table 3-604. CM_CLKMODE_DPLL_CORE_RESTORE.......... 703
Table 3-605. CM_SHADOW_FREQ_CONFIG2_RESTORE.......... 703
Table 3-606. CM_SHADOW_FREQ_CONFIG1_RESTORE.......... 704
Table 3-607. CM_AUTOIDLE_DPLL_CORE_RESTORE.......... 704
Table 3-608. CM_MPU_CLKSTCTRL_RESTORE.......... 704
Table 3-609. CM_CM_CORE_AON_PROFILING_CLKCTRL_RESTORE.......... 704
Table 3-610. CM_DYN_DEP_PRESCAL_RESTORE.......... 705
Table 3-611. CM_CORE_AON__RTC Registers Mapping Summary.......... 705
Table 3-612. CM_RTC_CLKSTCTRL.......... 705
Table 3-613. Register Call Summary for Register CM_RTC_CLKSTCTRL.......... 706
Table 3-614. CM_RTC_RTCSS_CLKCTRL.......... 706
Table 3-615. Register Call Summary for Register CM_RTC_RTCSS_CLKCTRL.......... 707
Table 3-616. CM_CORE_AON__VPE Registers Mapping Summary.......... 707
Table 3-617. CM_VPE_CLKSTCTRL.......... 707
Table 3-618. Register Call Summary for Register CM_VPE_CLKSTCTRL.......... 708
Table 3-619. CM_VPE_VPE_CLKCTRL.......... 708
Table 3-620. Register Call Summary for Register CM_VPE_VPE_CLKCTRL.......... 709
Table 3-621. CM_VPE_STATICDEP.......... 709
Table 3-622. Register Call Summary for Register CM_VPE_STATICDEP.......... 709
Table 3-623. CM_CORE__CAM Registers Mapping Summary.......... 710
Table 3-624. CM_CAM_CLKSTCTRL.......... 710
Table 3-625. CM_CAM_STATICDEP.......... 711
Table 3-626. CM_CAM_VIP1_CLKCTRL.......... 712
Table 3-627. CM_CAM_VIP2_CLKCTRL.......... 713
Table 3-628. CM_CAM_VIP3_CLKCTRL.......... 714
Table 3-629. CM_CAM_CSI1_CLKCTRL.......... 715
Table 3-630. CM_CAM_CSI2_CLKCTRL.......... 715
Table 3-631. CM_CORE__CKGEN Registers Mapping Summary.......... 716
Table 3-632. CM_CLKSEL_USB_60MHZ.......... 717
Table 3-633. Register Call Summary for Register CM_CLKSEL_USB_60MHZ.......... 717
Table 3-634. CM_CLKMODE_DPLL_PER.......... 717
Table 3-635. Register Call Summary for Register CM_CLKMODE_DPLL_PER.......... 718
Table 3-636. CM_IDLEST_DPLL_PER.......... 718
Table 3-637. Register Call Summary for Register CM_IDLEST_DPLL_PER.......... 719
Table 3-638. CM_AUTOIDLE_DPLL_PER.......... 719
Table 3-639. Register Call Summary for Register CM_AUTOIDLE_DPLL_PER.......... 720
Table 3-640. CM_CLKSEL_DPLL_PER.......... 720
Table 3-641. Register Call Summary for Register CM_CLKSEL_DPLL_PER.......... 721
Table 3-642. CM_DIV_M2_DPLL_PER.......... 721
Table 3-643. Register Call Summary for Register CM_DIV_M2_DPLL_PER.......... 721

Table 3-644. CM_DIV_H11_DPLL_PER....722
Table 3-645. Register Call Summary for Register CM_DIV_H11_DPLL_PER....722
Table 3-646. CM_DIV_H12_DPLL_PER....722
Table 3-647. Register Call Summary for Register CM_DIV_H12_DPLL_PER....723
Table 3-648. CM_DIV_H13_DPLL_PER....723
Table 3-649. Register Call Summary for Register CM_DIV_H13_DPLL_PER....723
Table 3-650. CM_DIV_H14_DPLL_PER....723
Table 3-651. Register Call Summary for Register CM_DIV_H14_DPLL_PER....724
Table 3-652. CM_CLKMODE_DPLL_USB....724
Table 3-653. Register Call Summary for Register CM_CLKMODE_DPLL_USB....725
Table 3-654. CM_IDLEST_DPLL_USB....725
Table 3-655. Register Call Summary for Register CM_IDLEST_DPLL_USB....725
Table 3-656. CM_AUTOIDLE_DPLL_USB....725
Table 3-657. Register Call Summary for Register CM_AUTOIDLE_DPLL_USB....726
Table 3-658. CM_CLKSEL_DPLL_USB....726
Table 3-659. Register Call Summary for Register CM_CLKSEL_DPLL_USB....727
Table 3-660. CM_DIV_M2_DPLL_USB....727
Table 3-661. Register Call Summary for Register CM_DIV_M2_DPLL_USB....728
Table 3-662. CM_CLKDCOLDO_DPLL_USB....728
Table 3-663. Register Call Summary for Register CM_CLKDCOLDO_DPLL_USB....728
Table 3-664. CM_CLKMODE_DPLL_PCIE_REF....728
Table 3-665. Register Call Summary for Register CM_CLKMODE_DPLL_PCIE_REF....729
Table 3-666. CM_IDLEST_DPLL_PCIE_REF....729
Table 3-667. Register Call Summary for Register CM_IDLEST_DPLL_PCIE_REF....730
Table 3-668. CM_AUTOIDLE_DPLL_PCIE_REF....730
Table 3-669. Register Call Summary for Register CM_AUTOIDLE_DPLL_PCIE_REF....730
Table 3-670. CM_CLKSEL_DPLL_PCIE_REF....731
Table 3-671. Register Call Summary for Register CM_CLKSEL_DPLL_PCIE_REF....732
Table 3-672. CM_DIV_M2_DPLL_PCIE_REF....732
Table 3-673. Register Call Summary for Register CM_DIV_M2_DPLL_PCIE_REF....732
Table 3-674. CM_CLKMODE_APLL_PCIE....732
Table 3-675. Register Call Summary for Register CM_CLKMODE_APLL_PCIE....733
Table 3-676. CM_IDLEST_APLL_PCIE....733
Table 3-677. Register Call Summary for Register CM_IDLEST_APLL_PCIE....734
Table 3-678. CM_DIV_M2_APLL_PCIE....734
Table 3-679. Register Call Summary for Register CM_DIV_M2_APLL_PCIE....734
Table 3-680. CM_CLKVCOLDO_APLL_PCIE....734
Table 3-681. Register Call Summary for Register CM_CLKVCOLDO_APLL_PCIE....735
Table 3-682. CM_CORE__COREAON Registers Mapping Summary....735
Table 3-683. CM_COREAON_CLKSTCTRL....735
Table 3-684. Register Call Summary for Register CM_COREAON_CLKSTCTRL....736
Table 3-685. CM_COREAON_USB_PHY1_CORE_CLKCTRL....737
Table 3-686. Register Call Summary for Register CM_COREAON_USB_PHY1_CORE_CLKCTRL....737
Table 3-687. CM_COREAON_USB_PHY2_CORE_CLKCTRL....737
Table 3-688. Register Call Summary for Register CM_COREAON_USB_PHY2_CORE_CLKCTRL....738
Table 3-689. CM_COREAON_USB_PHY3_CORE_CLKCTRL....738
Table 3-690. Register Call Summary for Register CM_COREAON_USB_PHY3_CORE_CLKCTRL....738
Table 3-691. CM_COREAON_CLKOUTMUX1_CLKCTRL....738
Table 3-692. Register Call Summary for Register CM_COREAON_CLKOUTMUX1_CLKCTRL....739
Table 3-693. CM_COREAON_CLKOUTMUX2_CLKCTRL....739
Table 3-694. Register Call Summary for Register CM_COREAON_CLKOUTMUX2_CLKCTRL....740
Table 3-695. CM_COREAON_L3INIT_60M_GFCLK_CLKCTRL....740
Table 3-696. Register Call Summary for Register CM_COREAON_L3INIT_60M_GFCLK_CLKCTRL....740
Table 3-697. CM_COREAON_ABE_GICLK_CLKCTRL....740
Table 3-698. Register Call Summary for Register CM_COREAON_ABE_GICLK_CLKCTRL....741
Table 3-699. CM_CORE__CORE Registers Mapping Summary....741
Table 3-700. CM_L3MAIN1_CLKSTCTRL....742
Table 3-701. Register Call Summary for Register CM_L3MAIN1_CLKSTCTRL....743
Table 3-702. CM_L3MAIN1_DYNAMICDEP....743
Table 3-703. Register Call Summary for Register CM_L3MAIN1_DYNAMICDEP....745
Table 3-704. CM_L3MAIN1_L3_MAIN_1_CLKCTRL....745

Table 3-705. Register Call Summary for Register CM_L3MAIN1_L3_MAIN_1_CLKCTRL....... 746
Table 3-706. CM_L3MAIN1_GPMC_CLKCTRL....... 746
Table 3-707. Register Call Summary for Register CM_L3MAIN1_GPMC_CLKCTRL....... 746
Table 3-708. CM_L3MAIN1_MMU_EDMA_CLKCTRL....... 746
Table 3-709. Register Call Summary for Register CM_L3MAIN1_MMU_EDMA_CLKCTRL....... 747
Table 3-710. CM_L3MAIN1_MMU_PCIESS_CLKCTRL....... 747
Table 3-711. Register Call Summary for Register CM_L3MAIN1_MMU_PCIESS_CLKCTRL....... 748
Table 3-712. CM_L3MAIN1_OCMC_RAM1_CLKCTRL....... 748
Table 3-713. Register Call Summary for Register CM_L3MAIN1_OCMC_RAM1_CLKCTRL....... 748
Table 3-714. CM_L3MAIN1_OCMC_RAM2_CLKCTRL....... 748
Table 3-715. Register Call Summary for Register CM_L3MAIN1_OCMC_RAM2_CLKCTRL....... 749
Table 3-716. CM_L3MAIN1_OCMC_RAM3_CLKCTRL....... 749
Table 3-717. Register Call Summary for Register CM_L3MAIN1_OCMC_RAM3_CLKCTRL....... 750
Table 3-718. CM_L3MAIN1_TPCC_CLKCTRL....... 750
Table 3-719. Register Call Summary for Register CM_L3MAIN1_TPCC_CLKCTRL....... 750
Table 3-720. CM_L3MAIN1_TPTC1_CLKCTRL....... 750
Table 3-721. Register Call Summary for Register CM_L3MAIN1_TPTC1_CLKCTRL....... 751
Table 3-722. CM_L3MAIN1_TPTC2_CLKCTRL....... 751
Table 3-723. Register Call Summary for Register CM_L3MAIN1_TPTC2_CLKCTRL....... 752
Table 3-724. CM_L3MAIN1_VCP1_CLKCTRL....... 752
Table 3-725. Register Call Summary for Register CM_L3MAIN1_VCP1_CLKCTRL....... 753
Table 3-726. CM_L3MAIN1_VCP2_CLKCTRL....... 753
Table 3-727. Register Call Summary for Register CM_L3MAIN1_VCP2_CLKCTRL....... 753
Table 3-728. CM_IPU2_CLKSTCTRL....... 753
Table 3-729. Register Call Summary for Register CM_IPU2_CLKSTCTRL....... 754
Table 3-730. CM_IPU2_STATICDEP....... 754
Table 3-731. Register Call Summary for Register CM_IPU2_STATICDEP....... 756
Table 3-732. CM_IPU2_DYNAMICDEP....... 757
Table 3-733. Register Call Summary for Register CM_IPU2_DYNAMICDEP....... 757
Table 3-734. CM_IPU2_IPU2_CLKCTRL....... 757
Table 3-735. Register Call Summary for Register CM_IPU2_IPU2_CLKCTRL....... 758
Table 3-736. CM_DMA_CLKSTCTRL....... 758
Table 3-737. Register Call Summary for Register CM_DMA_CLKSTCTRL....... 759
Table 3-738. CM_DMA_STATICDEP....... 759
Table 3-739. Register Call Summary for Register CM_DMA_STATICDEP....... 761
Table 3-740. CM_DMA_DYNAMICDEP....... 761
Table 3-741. Register Call Summary for Register CM_DMA_DYNAMICDEP....... 761
Table 3-742. CM_DMA_DMA_SYSTEM_CLKCTRL....... 761
Table 3-743. Register Call Summary for Register CM_DMA_DMA_SYSTEM_CLKCTRL....... 762
Table 3-744. CM_EMIF_CLKSTCTRL....... 762
Table 3-745. Register Call Summary for Register CM_EMIF_CLKSTCTRL....... 763
Table 3-746. CM_EMIF_DMM_CLKCTRL....... 763
Table 3-747. Register Call Summary for Register CM_EMIF_DMM_CLKCTRL....... 764
Table 3-748. CM_EMIF_EMIF_OCP_FW_CLKCTRL....... 764
Table 3-749. Register Call Summary for Register CM_EMIF_EMIF_OCP_FW_CLKCTRL....... 764
Table 3-750. CM_EMIF_EMIF1_CLKCTRL....... 764
Table 3-751. Register Call Summary for Register CM_EMIF_EMIF1_CLKCTRL....... 765
Table 3-752. CM_EMIF_EMIF2_CLKCTRL....... 765
Table 3-753. Register Call Summary for Register CM_EMIF_EMIF2_CLKCTRL....... 766
Table 3-754. CM_EMIF_EMIF_DLL_CLKCTRL....... 766
Table 3-755. Register Call Summary for Register CM_EMIF_EMIF_DLL_CLKCTRL....... 767
Table 3-756. CM_ATL_ATL_CLKCTRL....... 767
Table 3-757. Register Call Summary for Register CM_ATL_ATL_CLKCTRL....... 768
Table 3-758. CM_ATL_CLKSTCTRL....... 768
Table 3-759. Register Call Summary for Register CM_ATL_CLKSTCTRL....... 769
Table 3-760. CM_L4CFG_CLKSTCTRL....... 769
Table 3-761. Register Call Summary for Register CM_L4CFG_CLKSTCTRL....... 769
Table 3-762. CM_L4CFG_DYNAMICDEP....... 770
Table 3-763. Register Call Summary for Register CM_L4CFG_DYNAMICDEP....... 771
Table 3-764. CM_L4CFG_L4_CFG_CLKCTRL....... 771
Table 3-765. Register Call Summary for Register CM_L4CFG_L4_CFG_CLKCTRL....... 771

Table 3-766. CM_L4CFG_SPINLOCK_CLKCTRL....................771
Table 3-767. Register Call Summary for Register CM_L4CFG_SPINLOCK_CLKCTRL....................772
Table 3-768. CM_L4CFG_MAILBOX1_CLKCTRL....................772
Table 3-769. Register Call Summary for Register CM_L4CFG_MAILBOX1_CLKCTRL....................773
Table 3-770. CM_L4CFG_SAR_ROM_CLKCTRL....................773
Table 3-771. Register Call Summary for Register CM_L4CFG_SAR_ROM_CLKCTRL....................773
Table 3-772. CM_L4CFG_OCP2SCP2_CLKCTRL....................773
Table 3-773. Register Call Summary for Register CM_L4CFG_OCP2SCP2_CLKCTRL....................774
Table 3-774. CM_L4CFG_MAILBOX2_CLKCTRL....................774
Table 3-775. Register Call Summary for Register CM_L4CFG_MAILBOX2_CLKCTRL....................775
Table 3-776. CM_L4CFG_MAILBOX3_CLKCTRL....................775
Table 3-777. Register Call Summary for Register CM_L4CFG_MAILBOX3_CLKCTRL....................775
Table 3-778. CM_L4CFG_MAILBOX4_CLKCTRL....................775
Table 3-779. Register Call Summary for Register CM_L4CFG_MAILBOX4_CLKCTRL....................776
Table 3-780. CM_L4CFG_MAILBOX5_CLKCTRL....................776
Table 3-781. Register Call Summary for Register CM_L4CFG_MAILBOX5_CLKCTRL....................777
Table 3-782. CM_L4CFG_MAILBOX6_CLKCTRL....................777
Table 3-783. Register Call Summary for Register CM_L4CFG_MAILBOX6_CLKCTRL....................777
Table 3-784. CM_L4CFG_MAILBOX7_CLKCTRL....................777
Table 3-785. Register Call Summary for Register CM_L4CFG_MAILBOX7_CLKCTRL....................778
Table 3-786. CM_L4CFG_MAILBOX8_CLKCTRL....................778
Table 3-787. Register Call Summary for Register CM_L4CFG_MAILBOX8_CLKCTRL....................779
Table 3-788. CM_L4CFG_MAILBOX9_CLKCTRL....................779
Table 3-789. Register Call Summary for Register CM_L4CFG_MAILBOX9_CLKCTRL....................779
Table 3-790. CM_L4CFG_MAILBOX10_CLKCTRL....................779
Table 3-791. Register Call Summary for Register CM_L4CFG_MAILBOX10_CLKCTRL....................780
Table 3-792. CM_L4CFG_MAILBOX11_CLKCTRL....................780
Table 3-793. Register Call Summary for Register CM_L4CFG_MAILBOX11_CLKCTRL....................781
Table 3-794. CM_L4CFG_MAILBOX12_CLKCTRL....................781
Table 3-795. Register Call Summary for Register CM_L4CFG_MAILBOX12_CLKCTRL....................781
Table 3-796. CM_L4CFG_MAILBOX13_CLKCTRL....................781
Table 3-797. Register Call Summary for Register CM_L4CFG_MAILBOX13_CLKCTRL....................782
Table 3-798. CM_L3INSTR_CLKSTCTRL....................782
Table 3-799. Register Call Summary for Register CM_L3INSTR_CLKSTCTRL....................783
Table 3-800. CM_L3INSTR_L3_MAIN_2_CLKCTRL....................783
Table 3-801. Register Call Summary for Register CM_L3INSTR_L3_MAIN_2_CLKCTRL....................784
Table 3-802. CM_L3INSTR_L3_INSTR_CLKCTRL....................784
Table 3-803. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL....................785
Table 3-804. CM_L3INSTR_OCP_WP_NOC_CLKCTRL....................785
Table 3-805. Register Call Summary for Register CM_L3INSTR_OCP_WP_NOC_CLKCTRL....................786
Table 3-806. CM_L3INSTR_DLL_AGING_CLKCTRL....................786
Table 3-807. Register Call Summary for Register CM_L3INSTR_DLL_AGING_CLKCTRL....................786
Table 3-808. CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL....................787
Table 3-809. Register Call Summary for Register CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL....................787
Table 3-810. CM_CORE__CUSTEFUSE Registers Mapping Summary....................788
Table 3-811. CM_CUSTEFUSE_CLKSTCTRL....................788
Table 3-812. Register Call Summary for Register CM_CUSTEFUSE_CLKSTCTRL....................789
Table 3-813. CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL....................789
Table 3-814. Register Call Summary for Register CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL....................790
Table 3-815. CM_CORE__DSS Registers Mapping Summary....................790
Table 3-816. CM_DSS_CLKSTCTRL....................790
Table 3-817. Register Call Summary for Register CM_DSS_CLKSTCTRL....................792
Table 3-818. CM_DSS_STATICDEP....................792
Table 3-819. Register Call Summary for Register CM_DSS_STATICDEP....................792
Table 3-820. CM_DSS_DYNAMICDEP....................793
Table 3-821. Register Call Summary for Register CM_DSS_DYNAMICDEP....................793
Table 3-822. CM_DSS_DSS_CLKCTRL....................793
Table 3-823. Register Call Summary for Register CM_DSS_DSS_CLKCTRL....................794
Table 3-824. CM_DSS_BB2D_CLKCTRL....................795
Table 3-825. Register Call Summary for Register CM_DSS_BB2D_CLKCTRL....................795
Table 3-826. CM_CORE__GPU Registers Mapping Summary....................795

Table 3-827. CM_GPU_CLKSTCTRL....796
Table 3-828. Register Call Summary for Register CM_GPU_CLKSTCTRL....797
Table 3-829. CM_GPU_STATICDEP....797
Table 3-830. Register Call Summary for Register CM_GPU_STATICDEP....797
Table 3-831. CM_GPU_DYNAMICDEP....797
Table 3-832. Register Call Summary for Register CM_GPU_DYNAMICDEP....798
Table 3-833. CM_GPU_GPU_CLKCTRL....798
Table 3-834. Register Call Summary for Register CM_GPU_GPU_CLKCTRL....799
Table 3-835. CM_CORE__IVA Registers Mapping Summary....799
Table 3-836. CM_IVA_CLKSTCTRL....799
Table 3-837. Register Call Summary for Register CM_IVA_CLKSTCTRL....800
Table 3-838. CM_IVA_STATICDEP....800
Table 3-839. Register Call Summary for Register CM_IVA_STATICDEP....801
Table 3-840. CM_IVA_DYNAMICDEP....801
Table 3-841. Register Call Summary for Register CM_IVA_DYNAMICDEP....801
Table 3-842. CM_IVA_IVA_CLKCTRL....801
Table 3-843. Register Call Summary for Register CM_IVA_IVA_CLKCTRL....802
Table 3-844. CM_IVA_SL2_CLKCTRL....802
Table 3-845. Register Call Summary for Register CM_IVA_SL2_CLKCTRL....803
Table 3-846. CM_CORE__L3INIT Registers Mapping Summary....803
Table 3-847. CM_L3INIT_CLKSTCTRL....804
Table 3-848. Register Call Summary for Register CM_L3INIT_CLKSTCTRL....806
Table 3-849. CM_L3INIT_STATICDEP....806
Table 3-850. Register Call Summary for Register CM_L3INIT_STATICDEP....807
Table 3-851. CM_L3INIT_DYNAMICDEP....807
Table 3-852. Register Call Summary for Register CM_L3INIT_DYNAMICDEP....808
Table 3-853. CM_L3INIT_MMC1_CLKCTRL....808
Table 3-854. Register Call Summary for Register CM_L3INIT_MMC1_CLKCTRL....809
Table 3-855. CM_L3INIT_MMC2_CLKCTRL....809
Table 3-856. Register Call Summary for Register CM_L3INIT_MMC2_CLKCTRL....810
Table 3-857. CM_L3INIT_USB_OTG_SS2_CLKCTRL....810
Table 3-858. Register Call Summary for Register CM_L3INIT_USB_OTG_SS2_CLKCTRL....811
Table 3-859. CM_L3INIT_USB_OTG_SS3_CLKCTRL....811
Table 3-860. Register Call Summary for Register CM_L3INIT_USB_OTG_SS3_CLKCTRL....812
Table 3-861. CM_L3INIT_USB_OTG_SS4_CLKCTRL....812
Table 3-862. Register Call Summary for Register CM_L3INIT_USB_OTG_SS4_CLKCTRL....813
Table 3-863. CM_L3INIT_MLB_SS_CLKCTRL....813
Table 3-864. Register Call Summary for Register CM_L3INIT_MLB_SS_CLKCTRL....814
Table 3-865. CM_L3INIT_IEEE1500_2_OCP_CLKCTRL....814
Table 3-866. Register Call Summary for Register CM_L3INIT_IEEE1500_2_OCP_CLKCTRL....815
Table 3-867. CM_L3INIT_SATA_CLKCTRL....815
Table 3-868. Register Call Summary for Register CM_L3INIT_SATA_CLKCTRL....816
Table 3-869. CM_PCIE_CLKSTCTRL....816
Table 3-870. Register Call Summary for Register CM_PCIE_CLKSTCTRL....817
Table 3-871. CM_PCIE_STATICDEP....817
Table 3-872. Register Call Summary for Register CM_PCIE_STATICDEP....819
Table 3-873. CM_PCIE_PCIESS1_CLKCTRL....819
Table 3-874. Register Call Summary for Register CM_PCIE_PCIESS1_CLKCTRL....821
Table 3-875. CM_PCIE_PCIESS2_CLKCTRL....821
Table 3-876. Register Call Summary for Register CM_PCIE_PCIESS2_CLKCTRL....822
Table 3-877. CM_GMAC_CLKSTCTRL....822
Table 3-878. Register Call Summary for Register CM_GMAC_CLKSTCTRL....824
Table 3-879. CM_GMAC_STATICDEP....824
Table 3-880. Register Call Summary for Register CM_GMAC_STATICDEP....824
Table 3-881. CM_GMAC_DYNAMICDEP....824
Table 3-882. Register Call Summary for Register CM_GMAC_DYNAMICDEP....825
Table 3-883. CM_GMAC_GMAC_CLKCTRL....825
Table 3-884. Register Call Summary for Register CM_GMAC_GMAC_CLKCTRL....826
Table 3-885. CM_L3INIT_OCP2SCP1_CLKCTRL....826
Table 3-886. Register Call Summary for Register CM_L3INIT_OCP2SCP1_CLKCTRL....827
Table 3-887. CM_L3INIT_OCP2SCP3_CLKCTRL....827

Table 3-888. Register Call Summary for Register CM_L3INIT_OCP2SCP3_CLKCTRL....... 828
Table 3-889. CM_L3INIT_USB_OTG_SS1_CLKCTRL....... 828
Table 3-890. Register Call Summary for Register CM_L3INIT_USB_OTG_SS1_CLKCTRL....... 829
Table 3-891. CM_CORE__L4PER Registers Mapping Summary....... 829
Table 3-892. CM_L4PER_CLKSTCTRL....... 830
Table 3-893. Register Call Summary for Register CM_L4PER_CLKSTCTRL....... 833
Table 3-894. CM_L4PER_DYNAMICDEP....... 833
Table 3-895. Register Call Summary for Register CM_L4PER_DYNAMICDEP....... 834
Table 3-896. CM_L4PER2_L4_PER2_CLKCTRL....... 834
Table 3-897. Register Call Summary for Register CM_L4PER2_L4_PER2_CLKCTRL....... 835
Table 3-898. CM_L4PER3_L4_PER3_CLKCTRL....... 835
Table 3-899. Register Call Summary for Register CM_L4PER3_L4_PER3_CLKCTRL....... 835
Table 3-900. CM_L4PER_TIMER10_CLKCTRL....... 835
Table 3-901. Register Call Summary for Register CM_L4PER_TIMER10_CLKCTRL....... 836
Table 3-902. CM_L4PER_TIMER11_CLKCTRL....... 837
Table 3-903. Register Call Summary for Register CM_L4PER_TIMER11_CLKCTRL....... 837
Table 3-904. CM_L4PER_TIMER2_CLKCTRL....... 838
Table 3-905. Register Call Summary for Register CM_L4PER_TIMER2_CLKCTRL....... 839
Table 3-906. CM_L4PER_TIMER3_CLKCTRL....... 839
Table 3-907. Register Call Summary for Register CM_L4PER_TIMER3_CLKCTRL....... 840
Table 3-908. CM_L4PER_TIMER4_CLKCTRL....... 840
Table 3-909. Register Call Summary for Register CM_L4PER_TIMER4_CLKCTRL....... 841
Table 3-910. CM_L4PER_TIMER9_CLKCTRL....... 841
Table 3-911. Register Call Summary for Register CM_L4PER_TIMER9_CLKCTRL....... 842
Table 3-912. CM_L4PER_ELM_CLKCTRL....... 842
Table 3-913. Register Call Summary for Register CM_L4PER_ELM_CLKCTRL....... 843
Table 3-914. CM_L4PER_GPIO2_CLKCTRL....... 843
Table 3-915. Register Call Summary for Register CM_L4PER_GPIO2_CLKCTRL....... 844
Table 3-916. CM_L4PER_GPIO3_CLKCTRL....... 844
Table 3-917. Register Call Summary for Register CM_L4PER_GPIO3_CLKCTRL....... 845
Table 3-918. CM_L4PER_GPIO4_CLKCTRL....... 845
Table 3-919. Register Call Summary for Register CM_L4PER_GPIO4_CLKCTRL....... 846
Table 3-920. CM_L4PER_GPIO5_CLKCTRL....... 846
Table 3-921. Register Call Summary for Register CM_L4PER_GPIO5_CLKCTRL....... 847
Table 3-922. CM_L4PER_GPIO6_CLKCTRL....... 847
Table 3-923. Register Call Summary for Register CM_L4PER_GPIO6_CLKCTRL....... 847
Table 3-924. CM_L4PER_HDQ1W_CLKCTRL....... 848
Table 3-925. Register Call Summary for Register CM_L4PER_HDQ1W_CLKCTRL....... 848
Table 3-926. CM_L4PER2_PWMSS2_CLKCTRL....... 848
Table 3-927. Register Call Summary for Register CM_L4PER2_PWMSS2_CLKCTRL....... 849
Table 3-928. CM_L4PER2_PWMSS3_CLKCTRL....... 849
Table 3-929. Register Call Summary for Register CM_L4PER2_PWMSS3_CLKCTRL....... 850
Table 3-930. CM_L4PER_I2C1_CLKCTRL....... 850
Table 3-931. Register Call Summary for Register CM_L4PER_I2C1_CLKCTRL....... 851
Table 3-932. CM_L4PER_I2C2_CLKCTRL....... 851
Table 3-933. Register Call Summary for Register CM_L4PER_I2C2_CLKCTRL....... 851
Table 3-934. CM_L4PER_I2C3_CLKCTRL....... 851
Table 3-935. Register Call Summary for Register CM_L4PER_I2C3_CLKCTRL....... 852
Table 3-936. CM_L4PER_I2C4_CLKCTRL....... 852
Table 3-937. Register Call Summary for Register CM_L4PER_I2C4_CLKCTRL....... 853
Table 3-938. CM_L4PER_L4_PER1_CLKCTRL....... 853
Table 3-939. Register Call Summary for Register CM_L4PER_L4_PER1_CLKCTRL....... 853
Table 3-940. CM_L4PER2_PWMSS1_CLKCTRL....... 854
Table 3-941. Register Call Summary for Register CM_L4PER2_PWMSS1_CLKCTRL....... 854
Table 3-942. CM_L4PER3_TIMER13_CLKCTRL....... 854
Table 3-943. Register Call Summary for Register CM_L4PER3_TIMER13_CLKCTRL....... 855
Table 3-944. CM_L4PER3_TIMER14_CLKCTRL....... 855
Table 3-945. Register Call Summary for Register CM_L4PER3_TIMER14_CLKCTRL....... 856
Table 3-946. CM_L4PER3_TIMER15_CLKCTRL....... 857
Table 3-947. Register Call Summary for Register CM_L4PER3_TIMER15_CLKCTRL....... 857
Table 3-948. CM_L4PER_MCSPI1_CLKCTRL....... 858

Table 3-949. Register Call Summary for Register CM_L4PER_MCSPI1_CLKCTRL....858
Table 3-950. CM_L4PER_MCSPI2_CLKCTRL....858
Table 3-951. Register Call Summary for Register CM_L4PER_MCSPI2_CLKCTRL....859
Table 3-952. CM_L4PER_MCSPI3_CLKCTRL....859
Table 3-953. Register Call Summary for Register CM_L4PER_MCSPI3_CLKCTRL....860
Table 3-954. CM_L4PER_MCSPI4_CLKCTRL....860
Table 3-955. Register Call Summary for Register CM_L4PER_MCSPI4_CLKCTRL....861
Table 3-956. CM_L4PER_GPIO7_CLKCTRL....861
Table 3-957. Register Call Summary for Register CM_L4PER_GPIO7_CLKCTRL....861
Table 3-958. CM_L4PER_GPIO8_CLKCTRL....862
Table 3-959. Register Call Summary for Register CM_L4PER_GPIO8_CLKCTRL....862
Table 3-960. CM_L4PER_MMC3_CLKCTRL....863
Table 3-961. Register Call Summary for Register CM_L4PER_MMC3_CLKCTRL....864
Table 3-962. CM_L4PER_MMC4_CLKCTRL....864
Table 3-963. Register Call Summary for Register CM_L4PER_MMC4_CLKCTRL....865
Table 3-964. CM_L4PER3_TIMER16_CLKCTRL....865
Table 3-965. Register Call Summary for Register CM_L4PER3_TIMER16_CLKCTRL....866
Table 3-966. CM_L4PER2_QSPI_CLKCTRL....866
Table 3-967. Register Call Summary for Register CM_L4PER2_QSPI_CLKCTRL....867
Table 3-968. CM_L4PER_UART1_CLKCTRL....867
Table 3-969. Register Call Summary for Register CM_L4PER_UART1_CLKCTRL....868
Table 3-970. CM_L4PER_UART2_CLKCTRL....868
Table 3-971. Register Call Summary for Register CM_L4PER_UART2_CLKCTRL....869
Table 3-972. CM_L4PER_UART3_CLKCTRL....869
Table 3-973. Register Call Summary for Register CM_L4PER_UART3_CLKCTRL....870
Table 3-974. CM_L4PER_UART4_CLKCTRL....870
Table 3-975. Register Call Summary for Register CM_L4PER_UART4_CLKCTRL....871
Table 3-976. CM_L4PER2_MCASP2_CLKCTRL....871
Table 3-977. Register Call Summary for Register CM_L4PER2_MCASP2_CLKCTRL....872
Table 3-978. CM_L4PER2_MCASP3_CLKCTRL....873
Table 3-979. Register Call Summary for Register CM_L4PER2_MCASP3_CLKCTRL....874
Table 3-980. CM_L4PER_UART5_CLKCTRL....874
Table 3-981. Register Call Summary for Register CM_L4PER_UART5_CLKCTRL....875
Table 3-982. CM_L4PER2_MCASP5_CLKCTRL....875
Table 3-983. Register Call Summary for Register CM_L4PER2_MCASP5_CLKCTRL....876
Table 3-984. CM_L4SEC_CLKSTCTRL....876
Table 3-985. Register Call Summary for Register CM_L4SEC_CLKSTCTRL....877
Table 3-986. CM_L4SEC_STATICDEP....877
Table 3-987. Register Call Summary for Register CM_L4SEC_STATICDEP....878
Table 3-988. CM_L4SEC_DYNAMICDEP....878
Table 3-989. Register Call Summary for Register CM_L4SEC_DYNAMICDEP....878
Table 3-990. CM_L4PER2_MCASP8_CLKCTRL....878
Table 3-991. Register Call Summary for Register CM_L4PER2_MCASP8_CLKCTRL....880
Table 3-992. CM_L4PER2_MCASP4_CLKCTRL....880
Table 3-993. Register Call Summary for Register CM_L4PER2_MCASP4_CLKCTRL....881
Table 3-994. CM_L4SEC_AES1_CLKCTRL....881
Table 3-995. Register Call Summary for Register CM_L4SEC_AES1_CLKCTRL....882
Table 3-996. CM_L4SEC_AES2_CLKCTRL....882
Table 3-997. Register Call Summary for Register CM_L4SEC_AES2_CLKCTRL....883
Table 3-998. CM_L4SEC_DES3DES_CLKCTRL....883
Table 3-999. Register Call Summary for Register CM_L4SEC_DES3DES_CLKCTRL....884
Table 3-1000. CM_L4SEC_FPKA_CLKCTRL....884
Table 3-1001. Register Call Summary for Register CM_L4SEC_FPKA_CLKCTRL....884
Table 3-1002. CM_L4SEC_RNG_CLKCTRL....884
Table 3-1003. Register Call Summary for Register CM_L4SEC_RNG_CLKCTRL....885
Table 3-1004. CM_L4SEC_SHA2MD51_CLKCTRL....885
Table 3-1005. Register Call Summary for Register CM_L4SEC_SHA2MD51_CLKCTRL....886
Table 3-1006. CM_L4PER2_UART7_CLKCTRL....886
Table 3-1007. Register Call Summary for Register CM_L4PER2_UART7_CLKCTRL....887
Table 3-1008. CM_L4SEC_DMA_CRYPTO_CLKCTRL....887
Table 3-1009. Register Call Summary for Register CM_L4SEC_DMA_CRYPTO_CLKCTRL....888

Table 3-1010. CM_L4PER2_UART8_CLKCTRL....888
Table 3-1011. Register Call Summary for Register CM_L4PER2_UART8_CLKCTRL....888
Table 3-1012. CM_L4PER2_UART9_CLKCTRL....889
Table 3-1013. Register Call Summary for Register CM_L4PER2_UART9_CLKCTRL....889
Table 3-1014. CM_L4PER2_DCAN2_CLKCTRL....889
Table 3-1015. Register Call Summary for Register CM_L4PER2_DCAN2_CLKCTRL....890
Table 3-1016. CM_L4SEC_SHA2MD52_CLKCTRL....890
Table 3-1017. Register Call Summary for Register CM_L4SEC_SHA2MD52_CLKCTRL....891
Table 3-1018. CM_L4PER2_CLKSTCTRL....891
Table 3-1019. Register Call Summary for Register CM_L4PER2_CLKSTCTRL....894
Table 3-1020. CM_L4PER2_DYNAMICDEP....894
Table 3-1021. Register Call Summary for Register CM_L4PER2_DYNAMICDEP....895
Table 3-1022. CM_L4PER2_MCASP6_CLKCTRL....895
Table 3-1023. Register Call Summary for Register CM_L4PER2_MCASP6_CLKCTRL....896
Table 3-1024. CM_L4PER2_MCASP7_CLKCTRL....897
Table 3-1025. Register Call Summary for Register CM_L4PER2_MCASP7_CLKCTRL....898
Table 3-1026. CM_L4PER2_STATICDEP....898
Table 3-1027. Register Call Summary for Register CM_L4PER2_STATICDEP....899
Table 3-1028. CM_L4PER3_CLKSTCTRL....899
Table 3-1029. Register Call Summary for Register CM_L4PER3_CLKSTCTRL....900
Table 3-1030. CM_L4PER3_DYNAMICDEP....900
Table 3-1031. Register Call Summary for Register CM_L4PER3_DYNAMICDEP....901
Table 3-1032. CM_CORE__OCP_SOCKET Registers Mapping Summary....901
Table 3-1033. REVISION_CM_CORE....901
Table 3-1034. Register Call Summary for Register REVISION_CM_CORE....902
Table 3-1035. CM_CM_CORE_PROFILING_CLKCTRL....902
Table 3-1036. Register Call Summary for Register CM_CM_CORE_PROFILING_CLKCTRL....902
Table 3-1037. CM_CORE_DEBUG_CFG....902
Table 3-1038. Register Call Summary for Register CM_CORE_DEBUG_CFG....903
Table 3-1039. CM_CORE__RESTORE Registers Mapping Summary....903
Table 3-1040. CM_L3MAIN1_CLKSTCTRL_RESTORE....903
Table 3-1041. Register Call Summary for Register CM_L3MAIN1_CLKSTCTRL_RESTORE....903
Table 3-1042. CM_L4CFG_CLKSTCTRL_RESTORE....903
Table 3-1043. Register Call Summary for Register CM_L4CFG_CLKSTCTRL_RESTORE....904
Table 3-1044. CM_L4PER_CLKSTCTRL_RESTORE....904
Table 3-1045. Register Call Summary for Register CM_L4PER_CLKSTCTRL_RESTORE....904
Table 3-1046. CM_L3INIT_CLKSTCTRL_RESTORE....904
Table 3-1047. Register Call Summary for Register CM_L3INIT_CLKSTCTRL_RESTORE....904
Table 3-1048. CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE....904
Table 3-1049. Register Call Summary for Register CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE....905
Table 3-1050. CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE....905
Table 3-1051. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE....905
Table 3-1052. CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE....905
Table 3-1053. Register Call Summary for Register CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE....905
Table 3-1054. CM_CM_CORE_PROFILING_CLKCTRL_RESTORE....905
Table 3-1055. Register Call Summary for Register CM_CM_CORE_PROFILING_CLKCTRL_RESTORE....906
Table 3-1056. CM_L3MAIN1_DYNAMICDEP_RESTORE....906
Table 3-1057. Register Call Summary for Register CM_L3MAIN1_DYNAMICDEP_RESTORE....906
Table 3-1058. CM_L4CFG_DYNAMICDEP_RESTORE....906
Table 3-1059. Register Call Summary for Register CM_L4CFG_DYNAMICDEP_RESTORE....906
Table 3-1060. CM_L4PER_DYNAMICDEP_RESTORE....906
Table 3-1061. Register Call Summary for Register CM_L4PER_DYNAMICDEP_RESTORE....907
Table 3-1062. CM_DMA_STATICDEP_RESTORE....907
Table 3-1063. Register Call Summary for Register CM_DMA_STATICDEP_RESTORE....907
Table 3-1064. CAM_PRM Registers Mapping Summary....907
Table 3-1065. PM_CAM_PWRSTCTRL....908
Table 3-1066. PM_CAM_PWRSTST....908
Table 3-1067. PM_CAM_VIP1_WKDEP....909
Table 3-1068. RM_CAM_VIP1_CONTEXT....910
Table 3-1069. PM_CAM_VIP2_WKDEP....911
Table 3-1070. RM_CAM_VIP2_CONTEXT....912

Table 3-1071. PM_CAM_VIP3_WKDEP....912
Table 3-1072. RM_CAM_VIP3_CONTEXT....913
Table 3-1073. CKGEN_PRM Registers Mapping Summary....914
Table 3-1074. CM_CLKSEL_SYSCLK1....915
Table 3-1075. CM_CLKSEL_WKUPAON....916
Table 3-1076. CM_CLKSEL_ABE_PLL_REF....916
Table 3-1077. CM_CLKSEL_SYS....916
Table 3-1078. CM_CLKSEL_ABE_PLL_BYPAS....917
Table 3-1079. CM_CLKSEL_ABE_PLL_SYS....917
Table 3-1080. CM_CLKSEL_ABE_24M....917
Table 3-1081. CM_CLKSEL_ABE_SYS....918
Table 3-1082. CM_CLKSEL_HDMI_MCASP_AUX....918
Table 3-1083. CM_CLKSEL_HDMI_TIMER....919
Table 3-1084. CM_CLKSEL_MCASP_SYS....919
Table 3-1085. CM_CLKSEL_MLBP_MCASP....919
Table 3-1086. CM_CLKSEL_MLB_MCASP....920
Table 3-1087. CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX....920
Table 3-1088. CM_CLKSEL_TIMER_SYS....921
Table 3-1089. CM_CLKSEL_VIDEO1_MCASP_AUX....921
Table 3-1090. CM_CLKSEL_VIDEO1_TIMER....922
Table 3-1091. CM_CLKSEL_VIDEO2_MCASP_AUX....922
Table 3-1092. CM_CLKSEL_VIDEO2_TIMER....922
Table 3-1093. CM_CLKSEL_CLKOUTMUX0....923
Table 3-1094. CM_CLKSEL_CLKOUTMUX1....924
Table 3-1095. CM_CLKSEL_CLKOUTMUX2....926
Table 3-1096. CM_CLKSEL_HDMI_PLL_SYS....928
Table 3-1097. CM_CLKSEL_VIDEO1_PLL_SYS....929
Table 3-1098. CM_CLKSEL_VIDEO2_PLL_SYS....929
Table 3-1099. CM_CLKSEL_ABE_CLK_DIV....929
Table 3-1100. CM_CLKSEL_ABE_GICLK_DIV....930
Table 3-1101. CM_CLKSEL_AESS_FCLK_DIV....930
Table 3-1102. CM_CLKSEL_EVE_CLK....931
Table 3-1103. CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX....931
Table 3-1104. CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX....931
Table 3-1105. CM_CLKSEL_DSP_GFCLK_CLKOUTMUX....932
Table 3-1106. CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX....932
Table 3-1107. CM_CLKSEL_EMU_CLK_CLKOUTMUX....933
Table 3-1108. CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX....933
Table 3-1109. CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX....934
Table 3-1110. CM_CLKSEL_GPU_GCLK_CLKOUTMUX....934
Table 3-1111. CM_CLKSEL_HDMI_CLK_CLKOUTMUX....935
Table 3-1112. CM_CLKSEL_IVA_GCLK_CLKOUTMUX....935
Table 3-1113. CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX....936
Table 3-1114. CM_CLKSEL_MPU_GCLK_CLKOUTMUX....936
Table 3-1115. CM_CLKSEL_PCIE1_CLK_CLKOUTMUX....937
Table 3-1116. CM_CLKSEL_PCIE2_CLK_CLKOUTMUX....937
Table 3-1117. CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX....938
Table 3-1118. CM_CLKSEL_SATA_CLK_CLKOUTMUX....938
Table 3-1119. CM_CLKSEL_OSC_32K_CLK_CLKOUTMUX....939
Table 3-1120. CM_CLKSEL_SYS_CLK1_CLKOUTMUX....939
Table 3-1121. CM_CLKSEL_SYS_CLK2_CLKOUTMUX....940
Table 3-1122. CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX....940
Table 3-1123. CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX....941
Table 3-1124. CM_CLKSEL_ABE_LP_CLK....941
Table 3-1125. CM_CLKSEL_EVE_GFCLK_CLKOUTMUX....941
Table 3-1126. CORE_PRM Registers Mapping Summary....942
Table 3-1127. PM_CORE_PWRSTCTRL....943
Table 3-1128. PM_CORE_PWRSTST....944
Table 3-1129. RM_L3MAIN1_L3_MAIN_1_CONTEXT....946
Table 3-1130. RM_L3MAIN1_GPMC_CONTEXT....946
Table 3-1131. RM_L3MAIN1_MMU_EDMA_CONTEXT....947

Table 3-1132. RM_L3MAIN1_MMU_PCIESS_CONTEXT....947
Table 3-1133. PM_L3MAIN1_OCMC_RAM1_WKDEP....948
Table 3-1134. RM_L3MAIN1_OCMC_RAM1_CONTEXT....949
Table 3-1135. PM_L3MAIN1_OCMC_RAM2_WKDEP....950
Table 3-1136. RM_L3MAIN1_OCMC_RAM2_CONTEXT....951
Table 3-1137. PM_L3MAIN1_OCMC_RAM3_WKDEP....952
Table 3-1138. RM_L3MAIN1_OCMC_RAM3_CONTEXT....953
Table 3-1139. PM_L3MAIN1_TPCC_WKDEP....953
Table 3-1140. RM_L3MAIN1_TPCC_CONTEXT....955
Table 3-1141. PM_L3MAIN1_TPTC1_WKDEP....955
Table 3-1142. RM_L3MAIN1_TPTC1_CONTEXT....956
Table 3-1143. PM_L3MAIN1_TPTC2_WKDEP....957
Table 3-1144. RM_L3MAIN1_TPTC2_CONTEXT....958
Table 3-1145. RM_L3MAIN1_VCP1_CONTEXT....959
Table 3-1146. RM_L3MAIN1_VCP2_CONTEXT....959
Table 3-1147. RM_IPU2_RSTCTRL....960
Table 3-1148. RM_IPU2_RSTST....960
Table 3-1149. RM_IPU2_IPU2_CONTEXT....961
Table 3-1150. RM_DMA_DMA_SYSTEM_CONTEXT....962
Table 3-1151. RM_EMIF_DMM_CONTEXT....963
Table 3-1152. RM_EMIF_EMIF_OCP_FW_CONTEXT....963
Table 3-1153. RM_EMIF_EMIF1_CONTEXT....964
Table 3-1154. RM_EMIF_EMIF2_CONTEXT....964
Table 3-1155. RM_EMIF_EMIF_DLL_CONTEXT....965
Table 3-1156. RM_ATL_ATL_CONTEXT....965
Table 3-1157. RM_L4CFG_L4_CFG_CONTEXT....966
Table 3-1158. RM_L4CFG_SPINLOCK_CONTEXT....967
Table 3-1159. RM_L4CFG_MAILBOX1_CONTEXT....967
Table 3-1160. RM_L4CFG_SAR_ROM_CONTEXT....968
Table 3-1161. RM_L4CFG_OCP2SCP2_CONTEXT....968
Table 3-1162. RM_L4CFG_MAILBOX2_CONTEXT....969
Table 3-1163. RM_L4CFG_MAILBOX3_CONTEXT....969
Table 3-1164. RM_L4CFG_MAILBOX4_CONTEXT....970
Table 3-1165. RM_L4CFG_MAILBOX5_CONTEXT....970
Table 3-1166. RM_L4CFG_MAILBOX6_CONTEXT....971
Table 3-1167. RM_L4CFG_MAILBOX7_CONTEXT....971
Table 3-1168. RM_L4CFG_MAILBOX8_CONTEXT....972
Table 3-1169. RM_L4CFG_MAILBOX9_CONTEXT....972
Table 3-1170. RM_L4CFG_MAILBOX10_CONTEXT....973
Table 3-1171. RM_L4CFG_MAILBOX11_CONTEXT....973
Table 3-1172. RM_L4CFG_MAILBOX12_CONTEXT....974
Table 3-1173. RM_L4CFG_MAILBOX13_CONTEXT....974
Table 3-1174. RM_L3INSTR_L3_MAIN_2_CONTEXT....975
Table 3-1175. RM_L3INSTR_L3_INSTR_CONTEXT....975
Table 3-1176. RM_L3INSTR_OCP_WP_NOC_CONTEXT....976
Table 3-1177. CUSTEFUSE_PRM Registers Mapping Summary....976
Table 3-1178. PM_CUSTEFUSE_PWRSTCTRL....977
Table 3-1179. Register Call Summary for Register PM_CUSTEFUSE_PWRSTCTRL....977
Table 3-1180. PM_CUSTEFUSE_PWRSTST....977
Table 3-1181. Register Call Summary for Register PM_CUSTEFUSE_PWRSTST....978
Table 3-1182. RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT....978
Table 3-1183. Register Call Summary for Register RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT....979
Table 3-1184. DEVICE_PRM Registers Mapping Summary....979
Table 3-1185. PRM_RSTCTRL....980
Table 3-1186. PRM_RSTST....981
Table 3-1187. PRM_RSTTIME....982
Table 3-1188. PRM_PSCON_COUNT....982
Table 3-1189. PRM_IO_COUNT....983
Table 3-1190. PRM_IO_PMCTRL....983
Table 3-1191. PRM_SRAM_COUNT....984
Table 3-1192. PRM_SLDO_CORE_SETUP....985

Table 3-1193. PRM_SLDO_CORE_CTRL....986
Table 3-1194. PRM_SLDO_MPU_SETUP....986
Table 3-1195. PRM_SLDO_MPU_CTRL....988
Table 3-1196. PRM_SLDO_GPU_SETUP....988
Table 3-1197. PRM_SLDO_GPU_CTRL....989
Table 3-1198. PRM_ABBLDO_MPU_SETUP....990
Table 3-1199. PRM_ABBLDO_MPU_CTRL....991
Table 3-1200. PRM_ABBLDO_GPU_SETUP....992
Table 3-1201. PRM_ABBLDO_GPU_CTRL....992
Table 3-1202. PRM_BANDGAP_SETUP....993
Table 3-1203. PRM_DEVICE_OFF_CTRL....993
Table 3-1204. PRM_PHASE1_CNDP....994
Table 3-1205. PRM_PHASE2A_CNDP....994
Table 3-1206. PRM_PHASE2B_CNDP....995
Table 3-1207. PRM_SLDO_DSPEVE_SETUP....995
Table 3-1208. PRM_SLDO_IVA_SETUP....996
Table 3-1209. PRM_ABBLDO_DSPEVE_CTRL....997
Table 3-1210. PRM_ABBLDO_IVA_CTRL....998
Table 3-1211. PRM_SLDO_DSPEVE_CTRL....999
Table 3-1212. PRM_SLDO_IVA_CTRL....1000
Table 3-1213. PRM_ABBLDO_DSPEVE_SETUP....1000
Table 3-1214. PRM_ABBLDO_IVA_SETUP....1001
Table 3-1215. DSP1_PRM Registers Mapping Summary....1002
Table 3-1216. PM_DSP1_PWRSTCTRL....1002
Table 3-1217. PM_DSP1_PWRSTST....1003
Table 3-1218. RM_DSP1_RSTCTRL....1004
Table 3-1219. RM_DSP1_RSTST....1004
Table 3-1220. RM_DSP1_DSP1_CONTEXT....1005
Table 3-1221. DSP2_PRM Registers Mapping Summary....1006
Table 3-1222. PM_DSP2_PWRSTCTRL....1006
Table 3-1223. PM_DSP2_PWRSTST....1007
Table 3-1224. RM_DSP2_RSTCTRL....1008
Table 3-1225. RM_DSP2_RSTST....1008
Table 3-1226. RM_DSP2_DSP2_CONTEXT....1009
Table 3-1227. DSS_PRM Registers Mapping Summary....1010
Table 3-1228. PM_DSS_PWRSTCTRL....1010
Table 3-1229. Register Call Summary for Register PM_DSS_PWRSTCTRL....1011
Table 3-1230. PM_DSS_PWRSTST....1011
Table 3-1231. Register Call Summary for Register PM_DSS_PWRSTST....1012
Table 3-1232. PM_DSS_DSS_WKDEP....1012
Table 3-1233. Register Call Summary for Register PM_DSS_DSS_WKDEP....1015
Table 3-1234. RM_DSS_DSS_CONTEXT....1015
Table 3-1235. Register Call Summary for Register RM_DSS_DSS_CONTEXT....1016
Table 3-1236. PM_DSS_DSS2_WKDEP....1016
Table 3-1237. Register Call Summary for Register PM_DSS_DSS2_WKDEP....1018
Table 3-1238. RM_DSS_BB2D_CONTEXT....1019
Table 3-1239. Register Call Summary for Register RM_DSS_BB2D_CONTEXT....1019
Table 3-1240. EMU_CM Registers Mapping Summary....1019
Table 3-1241. CM_EMU_CLKSTCTRL....1020
Table 3-1242. Register Call Summary for Register CM_EMU_CLKSTCTRL....1020
Table 3-1243. CM_EMU_DEBUGSS_CLKCTRL....1020
Table 3-1244. Register Call Summary for Register CM_EMU_DEBUGSS_CLKCTRL....1021
Table 3-1245. CM_EMU_DYNAMICDEP....1021
Table 3-1246. Register Call Summary for Register CM_EMU_DYNAMICDEP....1022
Table 3-1247. CM_EMU_MPU_EMU_DBG_CLKCTRL....1022
Table 3-1248. Register Call Summary for Register CM_EMU_MPU_EMU_DBG_CLKCTRL....1022
Table 3-1249. EMU_PRM Registers Mapping Summary....1022
Table 3-1250. PM_EMU_PWRSTCTRL....1023
Table 3-1251. Register Call Summary for Register PM_EMU_PWRSTCTRL....1023
Table 3-1252. PM_EMU_PWRSTST....1023
Table 3-1253. Register Call Summary for Register PM_EMU_PWRSTST....1024

Table 3-1254. RM_EMU_DEBUGSS_CONTEXT....1024
Table 3-1255. Register Call Summary for Register RM_EMU_DEBUGSS_CONTEXT....1025
Table 3-1256. EVE1_PRM Registers Mapping Summary....1025
Table 3-1257. PM_EVE1_PWRSTCTRL....1025
Table 3-1258. PM_EVE1_PWRSTST....1026
Table 3-1259. RM_EVE1_RSTCTRL....1027
Table 3-1260. RM_EVE1_RSTST....1027
Table 3-1261. PM_EVE1_EVE1_WKDEP....1028
Table 3-1262. RM_EVE1_EVE1_CONTEXT....1029
Table 3-1263. EVE2_PRM Registers Mapping Summary....1030
Table 3-1264. PM_EVE2_PWRSTCTRL....1030
Table 3-1265. PM_EVE2_PWRSTST....1031
Table 3-1266. RM_EVE2_RSTCTRL....1032
Table 3-1267. RM_EVE2_RSTST....1032
Table 3-1268. PM_EVE2_EVE2_WKDEP....1033
Table 3-1269. RM_EVE2_EVE2_CONTEXT....1034
Table 3-1270. GPU_PRM Registers Mapping Summary....1034
Table 3-1271. PM_GPU_PWRSTCTRL....1035
Table 3-1272. Register Call Summary for Register PM_GPU_PWRSTCTRL....1035
Table 3-1273. PM_GPU_PWRSTST....1036
Table 3-1274. Register Call Summary for Register PM_GPU_PWRSTST....1036
Table 3-1275. RM_GPU_GPU_CONTEXT....1037
Table 3-1276. Register Call Summary for Register RM_GPU_GPU_CONTEXT....1037
Table 3-1277. INSTR_PRM Registers Mapping Summary....1037
Table 3-1278. PMI_IDENTICATION....1038
Table 3-1279. Register Call Summary for Register PMI_IDENTICATION....1038
Table 3-1280. PMI_SYS_CONFIG....1038
Table 3-1281. Register Call Summary for Register PMI_SYS_CONFIG....1038
Table 3-1282. PMI_STATUS....1038
Table 3-1283. Register Call Summary for Register PMI_STATUS....1039
Table 3-1284. PMI_CONFIGURATION....1039
Table 3-1285. Register Call Summary for Register PMI_CONFIGURATION....1039
Table 3-1286. PMI_CLASS_FILTERING....1039
Table 3-1287. Register Call Summary for Register PMI_CLASS_FILTERING....1040
Table 3-1288. PMI_TRIGGERING....1040
Table 3-1289. Register Call Summary for Register PMI_TRIGGERING....1040
Table 3-1290. PMI_SAMPLING....1040
Table 3-1291. Register Call Summary for Register PMI_SAMPLING....1041
Table 3-1292. IPU_PRM Registers Mapping Summary....1041
Table 3-1293. PM_IPU_PWRSTCTRL....1041
Table 3-1294. PM_IPU_PWRSTST....1042
Table 3-1295. RM_IPU1_RSTCTRL....1043
Table 3-1296. RM_IPU1_RSTST....1044
Table 3-1297. RM_IPU1_IPU1_CONTEXT....1045
Table 3-1298. PM_IPU_MCASP1_WKDEP....1046
Table 3-1299. RM_IPU_MCASP1_CONTEXT....1047
Table 3-1300. PM_IPU_TIMER5_WKDEP....1048
Table 3-1301. RM_IPU_TIMER5_CONTEXT....1049
Table 3-1302. PM_IPU_TIMER6_WKDEP....1049
Table 3-1303. RM_IPU_TIMER6_CONTEXT....1050
Table 3-1304. PM_IPU_TIMER7_WKDEP....1051
Table 3-1305. RM_IPU_TIMER7_CONTEXT....1052
Table 3-1306. PM_IPU_TIMER8_WKDEP....1052
Table 3-1307. RM_IPU_TIMER8_CONTEXT....1054
Table 3-1308. PM_IPU_I2C5_WKDEP....1054
Table 3-1309. RM_IPU_I2C5_CONTEXT....1056
Table 3-1310. PM_IPU_UART6_WKDEP....1056
Table 3-1311. RM_IPU_UART6_CONTEXT....1057
Table 3-1312. IVA_PRM Registers Mapping Summary....1058
Table 3-1313. PM_IVA_PWRSTCTRL....1058
Table 3-1314. PM_IVA_PWRSTST....1059

Table 3-1315. RM_IVA_RSTCTRL....... 1061
Table 3-1316. RM_IVA_RSTST....... 1061
Table 3-1317. RM_IVA_IVA_CONTEXT....... 1062
Table 3-1318. RM_IVA_SL2_CONTEXT....... 1063
Table 3-1319. L3INIT_PRM Registers Mapping Summary....... 1064
Table 3-1320. PM_L3INIT_PWRSTCTRL....... 1064
Table 3-1321. Register Call Summary for Register PM_L3INIT_PWRSTCTRL....... 1065
Table 3-1322. PM_L3INIT_PWRSTST....... 1065
Table 3-1323. Register Call Summary for Register PM_L3INIT_PWRSTST....... 1067
Table 3-1324. RM_PCIESS_RSTCTRL....... 1067
Table 3-1325. Register Call Summary for Register RM_PCIESS_RSTCTRL....... 1067
Table 3-1326. RM_PCIESS_RSTST....... 1067
Table 3-1327. Register Call Summary for Register RM_PCIESS_RSTST....... 1068
Table 3-1328. PM_L3INIT_MMC1_WKDEP....... 1068
Table 3-1329. Register Call Summary for Register PM_L3INIT_MMC1_WKDEP....... 1069
Table 3-1330. RM_L3INIT_MMC1_CONTEXT....... 1069
Table 3-1331. Register Call Summary for Register RM_L3INIT_MMC1_CONTEXT....... 1070
Table 3-1332. PM_L3INIT_MMC2_WKDEP....... 1070
Table 3-1333. Register Call Summary for Register PM_L3INIT_MMC2_WKDEP....... 1071
Table 3-1334. RM_L3INIT_MMC2_CONTEXT....... 1071
Table 3-1335. Register Call Summary for Register RM_L3INIT_MMC2_CONTEXT....... 1072
Table 3-1336. PM_L3INIT_USB_OTG_SS2_WKDEP....... 1072
Table 3-1337. Register Call Summary for Register PM_L3INIT_USB_OTG_SS2_WKDEP....... 1073
Table 3-1338. RM_L3INIT_USB_OTG_SS2_CONTEXT....... 1073
Table 3-1339. Register Call Summary for Register RM_L3INIT_USB_OTG_SS2_CONTEXT....... 1074
Table 3-1340. PM_L3INIT_USB_OTG_SS3_WKDEP....... 1074
Table 3-1341. Register Call Summary for Register PM_L3INIT_USB_OTG_SS3_WKDEP....... 1075
Table 3-1342. RM_L3INIT_USB_OTG_SS3_CONTEXT....... 1075
Table 3-1343. Register Call Summary for Register RM_L3INIT_USB_OTG_SS3_CONTEXT....... 1076
Table 3-1344. PM_L3INIT_USB_OTG_SS4_WKDEP....... 1076
Table 3-1345. Register Call Summary for Register PM_L3INIT_USB_OTG_SS4_WKDEP....... 1077
Table 3-1346. RM_L3INIT_USB_OTG_SS4_CONTEXT....... 1077
Table 3-1347. Register Call Summary for Register RM_L3INIT_USB_OTG_SS4_CONTEXT....... 1078
Table 3-1348. RM_L3INIT_MLB_SS_CONTEXT....... 1078
Table 3-1349. Register Call Summary for Register RM_L3INIT_MLB_SS_CONTEXT....... 1079
Table 3-1350. RM_L3INIT_IEEE1500_2_OCP_CONTEXT....... 1079
Table 3-1351. Register Call Summary for Register RM_L3INIT_IEEE1500_2_OCP_CONTEXT....... 1079
Table 3-1352. PM_L3INIT_SATA_WKDEP....... 1079
Table 3-1353. Register Call Summary for Register PM_L3INIT_SATA_WKDEP....... 1081
Table 3-1354. RM_L3INIT_SATA_CONTEXT....... 1081
Table 3-1355. Register Call Summary for Register RM_L3INIT_SATA_CONTEXT....... 1081
Table 3-1356. PM_PCIE_PCIESS1_WKDEP....... 1081
Table 3-1357. Register Call Summary for Register PM_PCIE_PCIESS1_WKDEP....... 1083
Table 3-1358. RM_PCIE_PCIESS1_CONTEXT....... 1083
Table 3-1359. Register Call Summary for Register RM_PCIE_PCIESS1_CONTEXT....... 1083
Table 3-1360. PM_PCIE_PCIESS2_WKDEP....... 1083
Table 3-1361. Register Call Summary for Register PM_PCIE_PCIESS2_WKDEP....... 1085
Table 3-1362. RM_PCIE_PCIESS2_CONTEXT....... 1085
Table 3-1363. Register Call Summary for Register RM_PCIE_PCIESS2_CONTEXT....... 1085
Table 3-1364. RM_GMAC_GMAC_CONTEXT....... 1085
Table 3-1365. Register Call Summary for Register RM_GMAC_GMAC_CONTEXT....... 1086
Table 3-1366. RM_L3INIT_OCP2SCP1_CONTEXT....... 1086
Table 3-1367. Register Call Summary for Register RM_L3INIT_OCP2SCP1_CONTEXT....... 1087
Table 3-1368. RM_L3INIT_OCP2SCP3_CONTEXT....... 1087
Table 3-1369. Register Call Summary for Register RM_L3INIT_OCP2SCP3_CONTEXT....... 1087
Table 3-1370. PM_L3INIT_USB_OTG_SS1_WKDEP....... 1087
Table 3-1371. Register Call Summary for Register PM_L3INIT_USB_OTG_SS1_WKDEP....... 1089
Table 3-1372. RM_L3INIT_USB_OTG_SS1_CONTEXT....... 1089
Table 3-1373. Register Call Summary for Register RM_L3INIT_USB_OTG_SS1_CONTEXT....... 1089
Table 3-1374. L4PER_PRM Registers Mapping Summary....... 1089
Table 3-1375. PM_L4PER_PWRSTCTRL....... 1092

Table 3-1376. Register Call Summary for Register PM_L4PER_PWRSTCTRL........ 1093
Table 3-1377. PM_L4PER_PWRSTST........ 1093
Table 3-1378. Register Call Summary for Register PM_L4PER_PWRSTST........ 1094
Table 3-1379. RM_L4PER2_L4PER2_CONTEXT........ 1094
Table 3-1380. Register Call Summary for Register RM_L4PER2_L4PER2_CONTEXT........ 1095
Table 3-1381. RM_L4PER3_L4PER3_CONTEXT........ 1095
Table 3-1382. Register Call Summary for Register RM_L4PER3_L4PER3_CONTEXT........ 1095
Table 3-1383. PM_L4PER_TIMER10_WKDEP........ 1095
Table 3-1384. Register Call Summary for Register PM_L4PER_TIMER10_WKDEP........ 1097
Table 3-1385. RM_L4PER_TIMER10_CONTEXT........ 1097
Table 3-1386. Register Call Summary for Register RM_L4PER_TIMER10_CONTEXT........ 1097
Table 3-1387. PM_L4PER_TIMER11_WKDEP........ 1097
Table 3-1388. Register Call Summary for Register PM_L4PER_TIMER11_WKDEP........ 1099
Table 3-1389. RM_L4PER_TIMER11_CONTEXT........ 1099
Table 3-1390. Register Call Summary for Register RM_L4PER_TIMER11_CONTEXT........ 1099
Table 3-1391. PM_L4PER_TIMER2_WKDEP........ 1099
Table 3-1392. Register Call Summary for Register PM_L4PER_TIMER2_WKDEP........ 1100
Table 3-1393. RM_L4PER_TIMER2_CONTEXT........ 1100
Table 3-1394. Register Call Summary for Register RM_L4PER_TIMER2_CONTEXT........ 1101
Table 3-1395. PM_L4PER_TIMER3_WKDEP........ 1101
Table 3-1396. Register Call Summary for Register PM_L4PER_TIMER3_WKDEP........ 1102
Table 3-1397. RM_L4PER_TIMER3_CONTEXT........ 1102
Table 3-1398. Register Call Summary for Register RM_L4PER_TIMER3_CONTEXT........ 1103
Table 3-1399. PM_L4PER_TIMER4_WKDEP........ 1103
Table 3-1400. Register Call Summary for Register PM_L4PER_TIMER4_WKDEP........ 1104
Table 3-1401. RM_L4PER_TIMER4_CONTEXT........ 1104
Table 3-1402. Register Call Summary for Register RM_L4PER_TIMER4_CONTEXT........ 1105
Table 3-1403. PM_L4PER_TIMER9_WKDEP........ 1105
Table 3-1404. Register Call Summary for Register PM_L4PER_TIMER9_WKDEP........ 1106
Table 3-1405. RM_L4PER_TIMER9_CONTEXT........ 1106
Table 3-1406. Register Call Summary for Register RM_L4PER_TIMER9_CONTEXT........ 1107
Table 3-1407. RM_L4PER_ELM_CONTEXT........ 1107
Table 3-1408. Register Call Summary for Register RM_L4PER_ELM_CONTEXT........ 1107
Table 3-1409. PM_L4PER_GPIO2_WKDEP........ 1107
Table 3-1410. Register Call Summary for Register PM_L4PER_GPIO2_WKDEP........ 1109
Table 3-1411. RM_L4PER_GPIO2_CONTEXT........ 1109
Table 3-1412. Register Call Summary for Register RM_L4PER_GPIO2_CONTEXT........ 1110
Table 3-1413. PM_L4PER_GPIO3_WKDEP........ 1110
Table 3-1414. Register Call Summary for Register PM_L4PER_GPIO3_WKDEP........ 1112
Table 3-1415. RM_L4PER_GPIO3_CONTEXT........ 1112
Table 3-1416. Register Call Summary for Register RM_L4PER_GPIO3_CONTEXT........ 1112
Table 3-1417. PM_L4PER_GPIO4_WKDEP........ 1113
Table 3-1418. Register Call Summary for Register PM_L4PER_GPIO4_WKDEP........ 1114
Table 3-1419. RM_L4PER_GPIO4_CONTEXT........ 1115
Table 3-1420. Register Call Summary for Register RM_L4PER_GPIO4_CONTEXT........ 1115
Table 3-1421. PM_L4PER_GPIO5_WKDEP........ 1115
Table 3-1422. Register Call Summary for Register PM_L4PER_GPIO5_WKDEP........ 1117
Table 3-1423. RM_L4PER_GPIO5_CONTEXT........ 1117
Table 3-1424. Register Call Summary for Register RM_L4PER_GPIO5_CONTEXT........ 1118
Table 3-1425. PM_L4PER_GPIO6_WKDEP........ 1118
Table 3-1426. Register Call Summary for Register PM_L4PER_GPIO6_WKDEP........ 1120
Table 3-1427. RM_L4PER_GPIO6_CONTEXT........ 1120
Table 3-1428. Register Call Summary for Register RM_L4PER_GPIO6_CONTEXT........ 1120
Table 3-1429. RM_L4PER_HDQ1W_CONTEXT........ 1121
Table 3-1430. Register Call Summary for Register RM_L4PER_HDQ1W_CONTEXT........ 1121
Table 3-1431. RM_L4PER2_PWMSS2_CONTEXT........ 1121
Table 3-1432. Register Call Summary for Register RM_L4PER2_PWMSS2_CONTEXT........ 1122
Table 3-1433. RM_L4PER2_PWMSS3_CONTEXT........ 1122
Table 3-1434. Register Call Summary for Register RM_L4PER2_PWMSS3_CONTEXT........ 1122
Table 3-1435. PM_L4PER_I2C1_WKDEP........ 1122
Table 3-1436. Register Call Summary for Register PM_L4PER_I2C1_WKDEP........ 1124

Table 3-1437. RM_L4PER_I2C1_CONTEXT....1124
Table 3-1438. Register Call Summary for Register RM_L4PER_I2C1_CONTEXT....1124
Table 3-1439. PM_L4PER_I2C2_WKDEP....1125
Table 3-1440. Register Call Summary for Register PM_L4PER_I2C2_WKDEP....1126
Table 3-1441. RM_L4PER_I2C2_CONTEXT....1126
Table 3-1442. Register Call Summary for Register RM_L4PER_I2C2_CONTEXT....1127
Table 3-1443. PM_L4PER_I2C3_WKDEP....1127
Table 3-1444. Register Call Summary for Register PM_L4PER_I2C3_WKDEP....1128
Table 3-1445. RM_L4PER_I2C3_CONTEXT....1128
Table 3-1446. Register Call Summary for Register RM_L4PER_I2C3_CONTEXT....1129
Table 3-1447. PM_L4PER_I2C4_WKDEP....1129
Table 3-1448. Register Call Summary for Register PM_L4PER_I2C4_WKDEP....1131
Table 3-1449. RM_L4PER_I2C4_CONTEXT....1131
Table 3-1450. Register Call Summary for Register RM_L4PER_I2C4_CONTEXT....1131
Table 3-1451. RM_L4PER_L4PER1_CONTEXT....1131
Table 3-1452. Register Call Summary for Register RM_L4PER_L4PER1_CONTEXT....1132
Table 3-1453. RM_L4PER2_PWMSS1_CONTEXT....1132
Table 3-1454. Register Call Summary for Register RM_L4PER2_PWMSS1_CONTEXT....1132
Table 3-1455. PM_L4PER_TIMER13_WKDEP....1132
Table 3-1456. Register Call Summary for Register PM_L4PER_TIMER13_WKDEP....1134
Table 3-1457. RM_L4PER3_TIMER13_CONTEXT....1134
Table 3-1458. Register Call Summary for Register RM_L4PER3_TIMER13_CONTEXT....1134
Table 3-1459. PM_L4PER_TIMER14_WKDEP....1134
Table 3-1460. Register Call Summary for Register PM_L4PER_TIMER14_WKDEP....1136
Table 3-1461. RM_L4PER3_TIMER14_CONTEXT....1136
Table 3-1462. Register Call Summary for Register RM_L4PER3_TIMER14_CONTEXT....1136
Table 3-1463. PM_L4PER_TIMER15_WKDEP....1136
Table 3-1464. Register Call Summary for Register PM_L4PER_TIMER15_WKDEP....1137
Table 3-1465. RM_L4PER3_TIMER15_CONTEXT....1137
Table 3-1466. Register Call Summary for Register RM_L4PER3_TIMER15_CONTEXT....1138
Table 3-1467. PM_L4PER_MCSPI1_WKDEP....1138
Table 3-1468. Register Call Summary for Register PM_L4PER_MCSPI1_WKDEP....1139
Table 3-1469. RM_L4PER_MCSPI1_CONTEXT....1139
Table 3-1470. Register Call Summary for Register RM_L4PER_MCSPI1_CONTEXT....1140
Table 3-1471. PM_L4PER_MCSPI2_WKDEP....1140
Table 3-1472. Register Call Summary for Register PM_L4PER_MCSPI2_WKDEP....1141
Table 3-1473. RM_L4PER_MCSPI2_CONTEXT....1141
Table 3-1474. Register Call Summary for Register RM_L4PER_MCSPI2_CONTEXT....1142
Table 3-1475. PM_L4PER_MCSPI3_WKDEP....1142
Table 3-1476. Register Call Summary for Register PM_L4PER_MCSPI3_WKDEP....1143
Table 3-1477. RM_L4PER_MCSPI3_CONTEXT....1143
Table 3-1478. Register Call Summary for Register RM_L4PER_MCSPI3_CONTEXT....1144
Table 3-1479. PM_L4PER_MCSPI4_WKDEP....1144
Table 3-1480. Register Call Summary for Register PM_L4PER_MCSPI4_WKDEP....1145
Table 3-1481. RM_L4PER_MCSPI4_CONTEXT....1145
Table 3-1482. Register Call Summary for Register RM_L4PER_MCSPI4_CONTEXT....1146
Table 3-1483. PM_L4PER_GPIO7_WKDEP....1146
Table 3-1484. Register Call Summary for Register PM_L4PER_GPIO7_WKDEP....1148
Table 3-1485. RM_L4PER_GPIO7_CONTEXT....1148
Table 3-1486. Register Call Summary for Register RM_L4PER_GPIO7_CONTEXT....1148
Table 3-1487. PM_L4PER_GPIO8_WKDEP....1149
Table 3-1488. Register Call Summary for Register PM_L4PER_GPIO8_WKDEP....1150
Table 3-1489. RM_L4PER_GPIO8_CONTEXT....1151
Table 3-1490. Register Call Summary for Register RM_L4PER_GPIO8_CONTEXT....1151
Table 3-1491. PM_L4PER_MMC3_WKDEP....1151
Table 3-1492. Register Call Summary for Register PM_L4PER_MMC3_WKDEP....1152
Table 3-1493. RM_L4PER_MMC3_CONTEXT....1152
Table 3-1494. Register Call Summary for Register RM_L4PER_MMC3_CONTEXT....1153
Table 3-1495. PM_L4PER_MMC4_WKDEP....1153
Table 3-1496. Register Call Summary for Register PM_L4PER_MMC4_WKDEP....1155
Table 3-1497. RM_L4PER_MMC4_CONTEXT....1155

Table 3-1498. Register Call Summary for Register RM_L4PER_MMC4_CONTEXT....1155
Table 3-1499. PM_L4PER_TIMER16_WKDEP....1156
Table 3-1500. Register Call Summary for Register PM_L4PER_TIMER16_WKDEP....1157
Table 3-1501. RM_L4PER3_TIMER16_CONTEXT....1157
Table 3-1502. Register Call Summary for Register RM_L4PER3_TIMER16_CONTEXT....1157
Table 3-1503. PM_L4PER2_QSPI_WKDEP....1157
Table 3-1504. Register Call Summary for Register PM_L4PER2_QSPI_WKDEP....1158
Table 3-1505. RM_L4PER2_QSPI_CONTEXT....1159
Table 3-1506. Register Call Summary for Register RM_L4PER2_QSPI_CONTEXT....1159
Table 3-1507. PM_L4PER_UART1_WKDEP....1159
Table 3-1508. Register Call Summary for Register PM_L4PER_UART1_WKDEP....1160
Table 3-1509. RM_L4PER_UART1_CONTEXT....1160
Table 3-1510. Register Call Summary for Register RM_L4PER_UART1_CONTEXT....1161
Table 3-1511. PM_L4PER_UART2_WKDEP....1161
Table 3-1512. Register Call Summary for Register PM_L4PER_UART2_WKDEP....1163
Table 3-1513. RM_L4PER_UART2_CONTEXT....1163
Table 3-1514. Register Call Summary for Register RM_L4PER_UART2_CONTEXT....1163
Table 3-1515. PM_L4PER_UART3_WKDEP....1164
Table 3-1516. Register Call Summary for Register PM_L4PER_UART3_WKDEP....1165
Table 3-1517. RM_L4PER_UART3_CONTEXT....1165
Table 3-1518. Register Call Summary for Register RM_L4PER_UART3_CONTEXT....1166
Table 3-1519. PM_L4PER_UART4_WKDEP....1166
Table 3-1520. Register Call Summary for Register PM_L4PER_UART4_WKDEP....1167
Table 3-1521. RM_L4PER_UART4_CONTEXT....1167
Table 3-1522. Register Call Summary for Register RM_L4PER_UART4_CONTEXT....1168
Table 3-1523. PM_L4PER2_MCASP2_WKDEP....1168
Table 3-1524. Register Call Summary for Register PM_L4PER2_MCASP2_WKDEP....1169
Table 3-1525. RM_L4PER2_MCASP2_CONTEXT....1169
Table 3-1526. Register Call Summary for Register RM_L4PER2_MCASP2_CONTEXT....1170
Table 3-1527. PM_L4PER2_MCASP3_WKDEP....1170
Table 3-1528. Register Call Summary for Register PM_L4PER2_MCASP3_WKDEP....1172
Table 3-1529. RM_L4PER2_MCASP3_CONTEXT....1172
Table 3-1530. Register Call Summary for Register RM_L4PER2_MCASP3_CONTEXT....1172
Table 3-1531. PM_L4PER_UART5_WKDEP....1172
Table 3-1532. Register Call Summary for Register PM_L4PER_UART5_WKDEP....1174
Table 3-1533. RM_L4PER_UART5_CONTEXT....1174
Table 3-1534. Register Call Summary for Register RM_L4PER_UART5_CONTEXT....1174
Table 3-1535. PM_L4PER2_MCASP5_WKDEP....1175
Table 3-1536. Register Call Summary for Register PM_L4PER2_MCASP5_WKDEP....1176
Table 3-1537. RM_L4PER2_MCASP5_CONTEXT....1176
Table 3-1538. Register Call Summary for Register RM_L4PER2_MCASP5_CONTEXT....1177
Table 3-1539. PM_L4PER2_MCASP6_WKDEP....1177
Table 3-1540. Register Call Summary for Register PM_L4PER2_MCASP6_WKDEP....1178
Table 3-1541. RM_L4PER2_MCASP6_CONTEXT....1178
Table 3-1542. Register Call Summary for Register RM_L4PER2_MCASP6_CONTEXT....1179
Table 3-1543. PM_L4PER2_MCASP7_WKDEP....1179
Table 3-1544. Register Call Summary for Register PM_L4PER2_MCASP7_WKDEP....1181
Table 3-1545. RM_L4PER2_MCASP7_CONTEXT....1181
Table 3-1546. Register Call Summary for Register RM_L4PER2_MCASP7_CONTEXT....1181
Table 3-1547. PM_L4PER2_MCASP8_WKDEP....1181
Table 3-1548. Register Call Summary for Register PM_L4PER2_MCASP8_WKDEP....1183
Table 3-1549. RM_L4PER2_MCASP8_CONTEXT....1183
Table 3-1550. Register Call Summary for Register RM_L4PER2_MCASP8_CONTEXT....1183
Table 3-1551. PM_L4PER2_MCASP4_WKDEP....1184
Table 3-1552. Register Call Summary for Register PM_L4PER2_MCASP4_WKDEP....1185
Table 3-1553. RM_L4PER2_MCASP4_CONTEXT....1185
Table 3-1554. Register Call Summary for Register RM_L4PER2_MCASP4_CONTEXT....1186
Table 3-1555. RM_L4SEC_AES1_CONTEXT....1186
Table 3-1556. Register Call Summary for Register RM_L4SEC_AES1_CONTEXT....1186
Table 3-1557. RM_L4SEC_AES2_CONTEXT....1186
Table 3-1558. Register Call Summary for Register RM_L4SEC_AES2_CONTEXT....1187

Table 3-1559. RM_L4SEC_DES3DES_CONTEXT.......... 1187
Table 3-1560. Register Call Summary for Register RM_L4SEC_DES3DES_CONTEXT.......... 1188
Table 3-1561. RM_L4SEC_FPKA_CONTEXT.......... 1188
Table 3-1562. Register Call Summary for Register RM_L4SEC_FPKA_CONTEXT.......... 1188
Table 3-1563. RM_L4SEC_RNG_CONTEXT.......... 1188
Table 3-1564. Register Call Summary for Register RM_L4SEC_RNG_CONTEXT.......... 1189
Table 3-1565. RM_L4SEC_SHA2MD51_CONTEXT.......... 1189
Table 3-1566. Register Call Summary for Register RM_L4SEC_SHA2MD51_CONTEXT.......... 1190
Table 3-1567. PM_L4PER2_UART7_WKDEP.......... 1190
Table 3-1568. Register Call Summary for Register PM_L4PER2_UART7_WKDEP.......... 1191
Table 3-1569. RM_L4PER2_UART7_CONTEXT.......... 1191
Table 3-1570. Register Call Summary for Register RM_L4PER2_UART7_CONTEXT.......... 1192
Table 3-1571. RM_L4SEC_DMA_CRYPTO_CONTEXT.......... 1192
Table 3-1572. Register Call Summary for Register RM_L4SEC_DMA_CRYPTO_CONTEXT.......... 1192
Table 3-1573. PM_L4PER2_UART8_WKDEP.......... 1193
Table 3-1574. Register Call Summary for Register PM_L4PER2_UART8_WKDEP.......... 1194
Table 3-1575. RM_L4PER2_UART8_CONTEXT.......... 1194
Table 3-1576. Register Call Summary for Register RM_L4PER2_UART8_CONTEXT.......... 1195
Table 3-1577. PM_L4PER2_UART9_WKDEP.......... 1195
Table 3-1578. Register Call Summary for Register PM_L4PER2_UART9_WKDEP.......... 1196
Table 3-1579. RM_L4PER2_UART9_CONTEXT.......... 1196
Table 3-1580. Register Call Summary for Register RM_L4PER2_UART9_CONTEXT.......... 1197
Table 3-1581. PM_L4PER2_DCAN2_WKDEP.......... 1197
Table 3-1582. Register Call Summary for Register PM_L4PER2_DCAN2_WKDEP.......... 1198
Table 3-1583. RM_L4PER2_DCAN2_CONTEXT.......... 1198
Table 3-1584. Register Call Summary for Register RM_L4PER2_DCAN2_CONTEXT.......... 1199
Table 3-1585. RM_L4SEC_SHA2MD52_CONTEXT.......... 1199
Table 3-1586. Register Call Summary for Register RM_L4SEC_SHA2MD52_CONTEXT.......... 1199
Table 3-1587. MPU_PRM Registers Mapping Summary.......... 1200
Table 3-1588. PM_MPU_PWRSTCTRL.......... 1200
Table 3-1589. Register Call Summary for Register PM_MPU_PWRSTCTRL.......... 1201
Table 3-1590. PM_MPU_PWRSTST.......... 1201
Table 3-1591. Register Call Summary for Register PM_MPU_PWRSTST.......... 1202
Table 3-1592. RM_MPU_MPU_CONTEXT.......... 1202
Table 3-1593. Register Call Summary for Register RM_MPU_MPU_CONTEXT.......... 1203
Table 3-1594. OCP_SOCKET_PRM Registers Mapping Summary.......... 1203
Table 3-1595. REVISION_PRM.......... 1204
Table 3-1596. Register Call Summary for Register REVISION_PRM.......... 1204
Table 3-1597. PRM_IRQSTATUS_MPU.......... 1204
Table 3-1598. Register Call Summary for Register PRM_IRQSTATUS_MPU.......... 1206
Table 3-1599. PRM_IRQSTATUS_MPU_2.......... 1206
Table 3-1600. Register Call Summary for Register PRM_IRQSTATUS_MPU_2.......... 1207
Table 3-1601. PRM_IRQENABLE_MPU.......... 1207
Table 3-1602. Register Call Summary for Register PRM_IRQENABLE_MPU.......... 1209
Table 3-1603. PRM_IRQENABLE_MPU_2.......... 1209
Table 3-1604. Register Call Summary for Register PRM_IRQENABLE_MPU_2.......... 1209
Table 3-1605. PRM_IRQSTATUS_IPU2.......... 1209
Table 3-1606. Register Call Summary for Register PRM_IRQSTATUS_IPU2.......... 1211
Table 3-1607. PRM_IRQENABLE_IPU2.......... 1212
Table 3-1608. Register Call Summary for Register PRM_IRQENABLE_IPU2.......... 1213
Table 3-1609. PRM_IRQSTATUS_DSP1.......... 1213
Table 3-1610. Register Call Summary for Register PRM_IRQSTATUS_DSP1.......... 1215
Table 3-1611. PRM_IRQENABLE_DSP1.......... 1216
Table 3-1612. Register Call Summary for Register PRM_IRQENABLE_DSP1.......... 1217
Table 3-1613. CM_PRM_PROFILING_CLKCTRL.......... 1217
Table 3-1614. Register Call Summary for Register CM_PRM_PROFILING_CLKCTRL.......... 1218
Table 3-1615. PRM_IRQENABLE_DSP2.......... 1218
Table 3-1616. Register Call Summary for Register PRM_IRQENABLE_DSP2.......... 1220
Table 3-1617. PRM_IRQENABLE_EVE1.......... 1220
Table 3-1618. Register Call Summary for Register PRM_IRQENABLE_EVE1.......... 1221
Table 3-1619. PRM_IRQENABLE_EVE2.......... 1222

Table 3-1620. Register Call Summary for Register PRM_IRQENABLE_EVE2....1223
Table 3-1621. PRM_IRQENABLE_IPU1....1223
Table 3-1622. Register Call Summary for Register PRM_IRQENABLE_IPU1....1225
Table 3-1623. PRM_IRQSTATUS_DSP2....1225
Table 3-1624. Register Call Summary for Register PRM_IRQSTATUS_DSP2....1227
Table 3-1625. PRM_IRQSTATUS_EVE1....1227
Table 3-1626. Register Call Summary for Register PRM_IRQSTATUS_EVE1....1229
Table 3-1627. PRM_IRQSTATUS_EVE2....1229
Table 3-1628. Register Call Summary for Register PRM_IRQSTATUS_EVE2....1231
Table 3-1629. PRM_IRQSTATUS_IPU1....1231
Table 3-1630. Register Call Summary for Register PRM_IRQSTATUS_IPU1....1233
Table 3-1631. PRM_DEBUG_CFG1....1233
Table 3-1632. Register Call Summary for Register PRM_DEBUG_CFG1....1234
Table 3-1633. PRM_DEBUG_CFG2....1234
Table 3-1634. Register Call Summary for Register PRM_DEBUG_CFG2....1234
Table 3-1635. PRM_DEBUG_CFG3....1234
Table 3-1636. Register Call Summary for Register PRM_DEBUG_CFG3....1234
Table 3-1637. PRM_DEBUG_CFG....1234
Table 3-1638. Register Call Summary for Register PRM_DEBUG_CFG....1235
Table 3-1639. PRM_DEBUG_OUT....1235
Table 3-1640. Register Call Summary for Register PRM_DEBUG_OUT....1235
Table 3-1641. RTC_PRM Registers Mapping Summary....1235
Table 3-1642. PM_RTC_RTCSS_WKDEP....1235
Table 3-1643. Register Call Summary for Register PM_RTC_RTCSS_WKDEP....1237
Table 3-1644. RM_RTC_RTCSS_CONTEXT....1237
Table 3-1645. Register Call Summary for Register RM_RTC_RTCSS_CONTEXT....1238
Table 3-1646. VPE_PRM Registers Mapping Summary....1238
Table 3-1647. PM_VPE_PWRSTCTRL....1238
Table 3-1648. Register Call Summary for Register PM_VPE_PWRSTCTRL....1239
Table 3-1649. PM_VPE_PWRSTST....1239
Table 3-1650. Register Call Summary for Register PM_VPE_PWRSTST....1240
Table 3-1651. PM_VPE_VPE_WKDEP....1240
Table 3-1652. Register Call Summary for Register PM_VPE_VPE_WKDEP....1241
Table 3-1653. RM_VPE_VPE_CONTEXT....1242
Table 3-1654. Register Call Summary for Register RM_VPE_VPE_CONTEXT....1242
Table 3-1655. WKUPAON_CM Registers Mapping Summary....1242
Table 3-1656. CM_WKUPAON_CLKSTCTRL....1243
Table 3-1657. Register Call Summary for Register CM_WKUPAON_CLKSTCTRL....1245
Table 3-1658. CM_WKUPAON_L4_WKUP_CLKCTRL....1245
Table 3-1659. Register Call Summary for Register CM_WKUPAON_L4_WKUP_CLKCTRL....1245
Table 3-1660. CM_WKUPAON_WD_TIMER2_CLKCTRL....1246
Table 3-1661. Register Call Summary for Register CM_WKUPAON_WD_TIMER2_CLKCTRL....1246
Table 3-1662. CM_WKUPAON_GPIO1_CLKCTRL....1246
Table 3-1663. Register Call Summary for Register CM_WKUPAON_GPIO1_CLKCTRL....1247
Table 3-1664. CM_WKUPAON_TIMER1_CLKCTRL....1247
Table 3-1665. Register Call Summary for Register CM_WKUPAON_TIMER1_CLKCTRL....1248
Table 3-1666. CM_WKUPAON_TIMER12_CLKCTRL....1248
Table 3-1667. Register Call Summary for Register CM_WKUPAON_TIMER12_CLKCTRL....1249
Table 3-1668. CM_WKUPAON_COUNTER_32K_CLKCTRL....1249
Table 3-1669. Register Call Summary for Register CM_WKUPAON_COUNTER_32K_CLKCTRL....1250
Table 3-1670. CM_WKUPAON_KBD_CLKCTRL....1250
Table 3-1671. Register Call Summary for Register CM_WKUPAON_KBD_CLKCTRL....1250
Table 3-1672. CM_WKUPAON_UART10_CLKCTRL....1251
Table 3-1673. Register Call Summary for Register CM_WKUPAON_UART10_CLKCTRL....1251
Table 3-1674. CM_WKUPAON_DCAN1_CLKCTRL....1251
Table 3-1675. Register Call Summary for Register CM_WKUPAON_DCAN1_CLKCTRL....1252
Table 3-1676. WKUPAON_PRM Registers Mapping Summary....1252
Table 3-1677. RM_WKUPAON_L4_WKUP_CONTEXT....1253
Table 3-1678. Register Call Summary for Register RM_WKUPAON_L4_WKUP_CONTEXT....1254
Table 3-1679. PM_WKUPAON_WD_TIMER2_WKDEP....1254
Table 3-1680. Register Call Summary for Register PM_WKUPAON_WD_TIMER2_WKDEP....1255

Table 3-1681. RM_WKUPAON_WD_TIMER2_CONTEXT....1255
Table 3-1682. Register Call Summary for Register RM_WKUPAON_WD_TIMER2_CONTEXT....1255
Table 3-1683. PM_WKUPAON_GPIO1_WKDEP....1256
Table 3-1684. Register Call Summary for Register PM_WKUPAON_GPIO1_WKDEP....1257
Table 3-1685. RM_WKUPAON_GPIO1_CONTEXT....1257
Table 3-1686. Register Call Summary for Register RM_WKUPAON_GPIO1_CONTEXT....1258
Table 3-1687. PM_WKUPAON_TIMER1_WKDEP....1258
Table 3-1688. Register Call Summary for Register PM_WKUPAON_TIMER1_WKDEP....1259
Table 3-1689. RM_WKUPAON_TIMER1_CONTEXT....1259
Table 3-1690. Register Call Summary for Register RM_WKUPAON_TIMER1_CONTEXT....1260
Table 3-1691. PM_WKUPAON_TIMER12_WKDEP....1260
Table 3-1692. Register Call Summary for Register PM_WKUPAON_TIMER12_WKDEP....1261
Table 3-1693. RM_WKUPAON_TIMER12_CONTEXT....1261
Table 3-1694. Register Call Summary for Register RM_WKUPAON_TIMER12_CONTEXT....1262
Table 3-1695. RM_WKUPAON_COUNTER_32K_CONTEXT....1262
Table 3-1696. Register Call Summary for Register RM_WKUPAON_COUNTER_32K_CONTEXT....1262
Table 3-1697. PM_WKUPAON_KBD_WKDEP....1263
Table 3-1698. Register Call Summary for Register PM_WKUPAON_KBD_WKDEP....1264
Table 3-1699. RM_WKUPAON_KBD_CONTEXT....1264
Table 3-1700. Register Call Summary for Register RM_WKUPAON_KBD_CONTEXT....1264
Table 3-1701. PM_WKUPAON_UART10_WKDEP....1264
Table 3-1702. Register Call Summary for Register PM_WKUPAON_UART10_WKDEP....1266
Table 3-1703. RM_WKUPAON_UART10_CONTEXT....1266
Table 3-1704. Register Call Summary for Register RM_WKUPAON_UART10_CONTEXT....1266
Table 3-1705. PM_WKUPAON_DCAN1_WKDEP....1266
Table 3-1706. Register Call Summary for Register PM_WKUPAON_DCAN1_WKDEP....1268
Table 3-1707. RM_WKUPAON_DCAN1_CONTEXT....1268
Table 3-1708. Register Call Summary for Register RM_WKUPAON_DCAN1_CONTEXT....1268
Table 4-1. MPU Subsystem Clocks Frequency Value Versus OPP....1275
Table 4-2. AXI Access Memory Mapping....1284
Table 4-3. MPU_MA Registers Duplicated From the DMM Register Map....1287
Table 4-4. COUNTER_REALTIME Increment Values....1291
Table 4-5. MPU_Cx State Transitions....1295
Table 4-6. MPU_Cx Supported Power States....1297
Table 4-7. Available MPU_Cx Power States in Single and Coherency Mode....1298
Table 4-8. MPU Subsystem Legal Power States....1298
Table 4-9. Dual Cortex-A15 MPU Subsystem Instance Summary....1301
Table 4-10. Local PRCM Revision Register Mapping Summary....1301
Table 4-11. REVISION_PRCM_MPU....1301
Table 4-12. Register Call Summary for Register REVISION_PRCM_MPU....1302
Table 4-13. MPU_PRCM_DEVICE Registers Mapping Summary....1302
Table 4-14. PRM_RSTST....1302
Table 4-15. PRM_PSCON_COUNT....1302
Table 4-16. PRM_FRAC_INCREMENTER_NUMERATOR....1303
Table 4-17. PRM_FRAC_INCREMENTER_DENUMERATOR_RELOAD....1303
Table 4-18. MPU_PRCM_PRM_C0 Registers Mapping Summary....1304
Table 4-19. PM_CPU0_PWRSTCTRL....1304
Table 4-20. Register Call Summary for Register PM_CPU0_PWRSTCTRL....1305
Table 4-21. PM_CPU0_PWRSTST....1305
Table 4-22. Register Call Summary for Register PM_CPU0_PWRSTST....1306
Table 4-23. RM_CPU0_CPU0_RSTCTRL....1306
Table 4-24. Register Call Summary for Register RM_CPU0_CPU0_RSTCTRL....1306
Table 4-25. RM_CPU0_CPU0_RSTST....1306
Table 4-26. Register Call Summary for Register RM_CPU0_CPU0_RSTST....1307
Table 4-27. RM_CPU0_CPU0_CONTEXT....1307
Table 4-28. Register Call Summary for Register RM_CPU0_CPU0_CONTEXT....1308
Table 4-29. MPU_PRCM_CM_C0 Registers Mapping Summary....1308
Table 4-30. CM_CPU0_CLKSTCTRL....1308
Table 4-31. Register Call Summary for Register CM_CPU0_CLKSTCTRL....1308
Table 4-32. CM_CPU0_CPU0_CLKCTRL....1308
Table 4-33. Register Call Summary for Register CM_CPU0_CPU0_CLKCTRL....1309

Table 4-34. MPU_PRCM_PRM_C1 Registers Mapping Summary 1309
Table 4-35. PM_CPU1_PWRSTCTRL 1309
Table 4-36. PM_CPU1_PWRSTST 1310
Table 4-37. RM_CPU1_CPU1_RSTCTRL 1311
Table 4-38. RM_CPU1_CPU1_RSTST 1311
Table 4-39. RM_CPU1_CPU1_CONTEXT 1312
Table 4-40. MPU_PRCM_CM_C1 Registers Mapping Summary 1312
Table 4-41. CM_CPU1_CLKSTCTRL 1312
Table 4-42. Register Call Summary for Register CM_CPU1_CLKSTCTRL 1313
Table 4-43. CM_CPU1_CPU1_CLKCTRL 1313
Table 4-44. Register Call Summary for Register CM_CPU1_CPU1_CLKCTRL 1313
Table 4-45. MPU_WUGEN Registers Mapping Summary 1313
Table 4-46. WKG_CONTROL_0 1314
Table 4-47. WKG_ENB_A_0 1315
Table 4-48. WKG_ENB_B_0 1316
Table 4-49. WKG_ENB_C_0 1317
Table 4-50. WKG_ENB_D_0 1318
Table 4-51. WKG_ENB_E_0 1319
Table 4-52. WKG_CONTROL_1 1320
Table 4-53. WKG_ENB_A_1 1321
Table 4-54. WKG_ENB_B_1 1322
Table 4-55. WKG_ENB_C_1 1323
Table 4-56. WKG_ENB_D_1 1325
Table 4-57. WKG_ENB_E_1 1326
Table 4-58. AUX_CORE_BOOT_0 1327
Table 4-59. AUX_CORE_BOOT_1 1327
Table 4-60. STM_HWEVENTS_INV 1327
Table 4-61. AMBA_IF_MODE 1330
Table 4-62. TIMESTAMPCYCLELO 1330
Table 4-63. TIMESTAMPCYCLEHI 1331
Table 4-64. MPU_WD_TIMER Registers Mapping Summary 1331
Table 4-65. WDT_LOAD_REGISTER_i 1331
Table 4-66. Register Call Summary for Register WDT_LOAD_REGISTER_i 1331
Table 4-67. WDT_COUNT_REGISTER_i 1332
Table 4-68. Register Call Summary for Register WDT_COUNT_REGISTER_i 1332
Table 4-69. WDT_WARNING_REGISTER_i 1332
Table 4-70. Register Call Summary for Register WDT_WARNING_REGISTER_i 1332
Table 4-71. WDT_PRESCALER_REGISTER_i 1333
Table 4-72. Register Call Summary for Register WDT_PRESCALER_REGISTER_i 1333
Table 4-73. WDT_CONTROL_REGISTER_i 1333
Table 4-74. Register Call Summary for Register WDT_CONTROL_REGISTER_i 1334
Table 4-75. WDT_RESET_STATUS_REGISTER_i 1334
Table 4-76. Register Call Summary for Register WDT_RESET_STATUS_REGISTER_i 1334
Table 4-77. MPU_AXI2OCP_MISC Register Mapping Summary 1335
Table 4-78. MA_PRIORITY 1335
Table 4-79. MPU_MA_LSM Register Mapping Summary 1335
Table 4-80. MPU_MA_WP Registers Mapping Summary 1336
Table 4-81. DBG_HWWP_CAP 1336
Table 4-82. Register Call Summary for Register DBG_HWWP_CAP 1337
Table 4-83. TRIG_CTRL 1337
Table 4-84. Register Call Summary for Register TRIG_CTRL 1338
Table 4-85. DBG_HWWP0_LW_ADDR0 1338
Table 4-86. Register Call Summary for Register DBG_HWWP0_LW_ADDR0 1338
Table 4-87. DBG_HWWP0_HG_ADDR0 1338
Table 4-88. Register Call Summary for Register DBG_HWWP0_HG_ADDR0 1339
Table 4-89. DBG_HWWP0_MAIN_CNTL 1339
Table 4-90. Register Call Summary for Register DBG_HWWP0_MAIN_CNTL 1340
Table 4-91. DBG_HWWP0_AUX_CNTL 1340
Table 4-92. Register Call Summary for Register DBG_HWWP0_AUX_CNTL 1341
Table 4-93. DBG_HWWP0_MEM_CNTL 1341
Table 4-94. Register Call Summary for Register DBG_HWWP0_MEM_CNTL 1341

Table 4-95. DBG_HWWP0_CHAIN_CNTL....1342
Table 4-96. Register Call Summary for Register DBG_HWWP0_CHAIN_CNTL....1342
Table 4-97. DBG_HWWP0_LW_ADDR0_LOG....1342
Table 4-98. Register Call Summary for Register DBG_HWWP0_LW_ADDR0_LOG....1343
Table 4-99. DBG_HWWP0_HG_ADDR0_LOG....1343
Table 4-100. Register Call Summary for Register DBG_HWWP0_HG_ADDR0_LOG....1343
Table 4-101. DBG_HWWP0_DATA0_LOG....1343
Table 4-102. Register Call Summary for Register DBG_HWWP0_DATA0_LOG....1344
Table 4-103. DBG_HWWP0_DATA1_LOG....1344
Table 4-104. Register Call Summary for Register DBG_HWWP0_DATA1_LOG....1344
Table 4-105. DBG_HWWP0_DATA2_LOG....1344
Table 4-106. Register Call Summary for Register DBG_HWWP0_DATA2_LOG....1344
Table 4-107. DBG_HWWP0_DATA3_LOG....1344
Table 4-108. Register Call Summary for Register DBG_HWWP0_DATA3_LOG....1345
Table 4-109. DBG_HWWP0_TRANS_ATTR0_LOG....1345
Table 4-110. Register Call Summary for Register DBG_HWWP0_TRANS_ATTR0_LOG....1346
Table 4-111. DBG_HWWP0_TRANS_ATTR1_LOG....1346
Table 4-112. Register Call Summary for Register DBG_HWWP0_TRANS_ATTR1_LOG....1347
Table 4-113. DBG_HWWP0_DATA_TRANS_ATTR0_LOG....1347
Table 4-114. Register Call Summary for Register DBG_HWWP0_DATA_TRANS_ATTR0_LOG....1347
Table 5-1. DSP Integration Attributes....1354
Table 5-2. DSP Clocks and Resets....1355
Table 5-3. DSP Hardware Requests....1356
Table 5-4. Summary of the DSP1 and DSP2 Hardware Resets....1371
Table 5-5. DSP ERRINT Interrupt Mapping....1379
Table 5-6. DSP1_EDMA Default Request Mapping....1383
Table 5-7. DSP2_EDMA Default Request Mapping....1384
Table 5-8. DSP_NoC Defined Connectivities....1390
Table 5-9. C66x CPU View Map....1393
Table 5-10. DSP EDMA Controller View Map....1394
Table 5-11. SDMA Target Port Memory Map....1395
Table 5-12. DSP Subsystem Instance Summary....1396
Table 5-13. DSP_ICFG Registers Mapping Summary....1397
Table 5-14. DSP_SYSTEM and DSP1_SYSTEM Registers Mapping Summary....1401
Table 5-15. DSP2_SYSTEM Registers Mapping Summary....1402
Table 5-16. DSP_SYS_REVISION....1402
Table 5-17. DSP_SYS_HWINFO....1403
Table 5-18. DSP_SYS_SYSCONFIG....1403
Table 5-19. DSP_SYS_STAT....1404
Table 5-20. DSP_SYS_DISC_CONFIG....1405
Table 5-21. DSP_SYS_BUS_CONFIG....1405
Table 5-22. DSP_SYS_MMU_CONFIG....1407
Table 5-23. DSP_SYS_IRQWAKEEN0....1408
Table 5-24. DSP_SYS_IRQWAKEEN1....1408
Table 5-25. DSP_SYS_DMAWAKEEN0....1409
Table 5-26. DSP_SYS_DMAWAKEEN1....1409
Table 5-27. DSP_SYS_EVTOUT_SET....1409
Table 5-28. DSP_SYS_EVTOUT_CLR....1410
Table 5-29. DSP_SYS_ERRINT_IRQSTATUS_RAW....1410
Table 5-30. DSP_SYS_ERRINT_IRQSTATUS....1410
Table 5-31. DSP_SYS_ERRINT_IRQENABLE_SET....1411
Table 5-32. DSP_SYS_ERRINT_IRQENABLE_CLR....1411
Table 5-33. DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW....1412
Table 5-34. DSP_SYS_EDMAWAKE0_IRQSTATUS....1412
Table 5-35. DSP_SYS_EDMAWAKE0_IRQENABLE_SET....1412
Table 5-36. DSP_SYS_EDMAWAKE0_IRQENABLE_CLR....1413
Table 5-37. DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW....1413
Table 5-38. DSP_SYS_EDMAWAKE1_IRQSTATUS....1413
Table 5-39. DSP_SYS_EDMAWAKE1_IRQENABLE_SET....1414
Table 5-40. DSP_SYS_EDMAWAKE1_IRQENABLE_CLR....1414
Table 5-41. DSP_SYS_HW_DBGOUT_SEL....1414

Table 5-42. DSP_SYS_HW_DBGOUT_VAL....1415
Table 5-43. DSP_FW_L2_NOC_CFG and DSP1_FW_L2_NOC_CFG Registers Mapping Summary....1415
Table 5-44. DSP2_FW_L2_NOC_CFG Registers Mapping Summary....1416
Table 5-45. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0....1417
Table 5-46. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0....1418
Table 5-47. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL....1418
Table 5-48. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0...1418
Table 5-49. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0..1419
Table 5-50. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1....1420
Table 5-51. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1....1420
Table 5-52. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1...1421
Table 5-53. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1..1421
Table 5-54. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0....1422
Table 5-55. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0....1423
Table 5-56. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL....1423
Table 5-57. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0...1423
Table 5-58. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0..1424
Table 5-59. DSPNOC_FLAGMUX_ID_COREID....1425
Table 5-60. DSPNOC_FLAGMUX_ID_REVISIONID....1425
Table 5-61. DSPNOC_FLAGMUX_FAULTEN....1425
Table 5-62. DSPNOC_FLAGMUX_FAULTSTATUS....1426
Table 5-63. DSPNOC_FLAGMUX_FLAGINEN0....1426
Table 5-64. DSPNOC_FLAGMUX_FLAGINSTATUS0....1426
Table 5-65. DSPNOC_ERRORLOG_ID_COREID....1427
Table 5-66. DSPNOC_ERRORLOG_ID_REVISIONID....1427
Table 5-67. DSPNOC_ERRORLOG_FAULTEN....1427
Table 5-68. DSPNOC_ERRORLOG_ERRVLD....1428
Table 5-69. DSPNOC_ERRORLOG_ERRCLR....1428
Table 5-70. DSPNOC_ERRORLOG_ERRLOG0....1428
Table 5-71. DSPNOC_ERRORLOG_ERRLOG1....1429
Table 5-72. DSPNOC_ERRORLOG_ERRLOG3....1429
Table 5-73. DSPNOC_ERRORLOG_ERRLOG5....1429
Table 7-1. IPUx Integration Attributes....1437
Table 7-2. IPUx Hardware Requests....1437
Table 7-3. IPUx Clocks and Resets....1437
Table 7-4. Local Clock Gating....1443
Table 7-5. IPUx Subsystem Power Modes....1444
Table 7-6. Power Mode Transitions....1444
Table 7-7. IPUx_UNICACHE Configuration....1446
Table 7-8. IPUx_UNICACHE_MMU Configuration....1447
Table 7-9. IPUx_MMU Behavior on Page-Fault....1451
Table 7-10. IPU1 Subsystem Instance Summary....1454
Table 7-11. IPU2 Subsystem Instance Summary....1454
Table 7-12. IPU1_UNICACHE_CFG Registers Mapping Summary....1454
Table 7-13. IPU2_UNICACHE_CFG Registers Mapping Summary....1455
Table 7-14. CACHE_CONFIG....1455
Table 7-15. Register Call Summary for Register CACHE_CONFIG....1456
Table 7-16. CACHE_INT....1456
Table 7-17. Register Call Summary for Register CACHE_INT....1456
Table 7-18. CACHE_OCP....1456
Table 7-19. Register Call Summary for Register CACHE_OCP....1457
Table 7-20. CACHE_MAINT....1457
Table 7-21. Register Call Summary for Register CACHE_MAINT....1458
Table 7-22. CACHE_MTSTART....1458
Table 7-23. Register Call Summary for Register CACHE_MTSTART....1459
Table 7-24. CACHE_MTEND....1459
Table 7-25. Register Call Summary for Register CACHE_MTEND....1459
Table 7-26. CACHE_CTADDR....1459
Table 7-27. Register Call Summary for Register CACHE_CTADDR....1460
Table 7-28. CACHE_CTDATA....1460
Table 7-29. Register Call Summary for Register CACHE_CTDATA....1460

Table 7-30. IPU1_UNICACHE_SCTM Registers Mapping Summary....1460
Table 7-31. IPU2_UNICACHE_SCTM Registers Mapping Summary....1461
Table 7-32. CACHE_SCTM_CTCNTL....1461
Table 7-33. CACHE_SCTM_TINTVLR_i....1462
Table 7-34. CACHE_SCTM_CTDBGNUM....1462
Table 7-35. CACHE_SCTM_CTGNBL....1462
Table 7-36. CACHE_SCTM_CTGRST....1463
Table 7-37. CACHE_SCTM_CTCR_WT_i....1463
Table 7-38. CACHE_SCTM_CTCR_WOT_j....1464
Table 7-39. CACHE_SCTM_CTCNTR_k....1466
Table 7-40. IPU1_UNICACHE_MMU (AMMU) Registers Mapping Summary....1466
Table 7-41. IPU2_UNICACHE_MMU (AMMU) Registers Mapping Summary....1467
Table 7-42. CACHE_MMU_LARGE_ADDR_i....1467
Table 7-43. CACHE_MMU_LARGE_XLTE_i....1467
Table 7-44. CACHE_MMU_LARGE_POLICY_i....1468
Table 7-45. CACHE_MMU_MED_ADDR_j....1469
Table 7-46. CACHE_MMU_MED_XLTE_j....1469
Table 7-47. CACHE_MMU_MED_POLICY_j....1469
Table 7-48. CACHE_MMU_SMALL_ADDR_k....1470
Table 7-49. Reset Value for CACHE_MMU_SMALL_ADDR_k[31:12] ADDRESS....1471
Table 7-50. CACHE_MMU_SMALL_XLTE_k....1471
Table 7-51. Reset Value for CACHE_MMU_SMALL_XLTE_k[31:12] ADDRESS....1471
Table 7-52. CACHE_MMU_SMALL_POLICY_k....1471
Table 7-53. CACHE_MMU_SMALL_MAINT_k....1472
Table 7-54. CACHE_MMU_MMUCONFIG....1473
Table 7-55. IPU1_WUGEN Registers Mapping Summary....1474
Table 7-56. IPU2_WUGEN Registers Mapping Summary....1474
Table 7-57. CORTEXM4_CTRL_REG....1474
Table 7-58. STANDBY_CORE_SYSCONFIG....1474
Table 7-59. IDLE_CORE_SYSCONFIG....1475
Table 7-60. WUGEN_MEVT0....1475
Table 7-61. WUGEN_MEVT1....1476
Table 7-62. IPU1_Cx_RW_TABLE Register Summary....1477
Table 7-63. IPU2_Cx_RW_TABLE Register Summary....1478
Table 7-64. CORTEXM4_RW_PID1....1478
Table 7-65. Register Call Summary for Register CORTEXM4_RW_PID1....1478
Table 7-66. CORTEXM4_RW_PID2....1478
Table 7-67. Register Call Summary for Register CORTEXM4_RW_PID2....1478
Table 8-1. EVE Subsystem Memory Blocks....1483
Table 8-2. EVE Integration Attributes....1485
Table 8-3. EVE Clocks and Resets....1485
Table 8-4. EVE Hardware Requests....1485
Table 8-5. EVE Internal ConnID Mapping....1488
Table 8-6. Internal Memory Blocks....1489
Table 8-7. Error Detection Modes....1494
Table 8-8. EVE Tag Visibility....1500
Table 8-9. Cache Profiling Signal List....1500
Table 8-10. EDMA Configuration....1502
Table 8-11. Fields in Parameter RAM....1503
Table 8-12. EVE-Level Effective Bus Width....1505
Table 8-13. MMU Configuration....1509
Table 8-14. EVE_MSW_ERR Register Interrupt Mapping....1511
Table 8-15. EVE Local and Output Error Detect Error Interrupt Mapping....1512
Table 8-16. EVE ARP32 Interrupt Event Mapping Group0/INTC0....1513
Table 8-17. EVE ARP32 Interrupt Event Mapping Group1/INTC1....1514
Table 8-18. ARP32 Interrupt Mapping for Group2/INTC2....1515
Table 8-19. ARP32 Interrupt Mapping for Group3/INTC3....1515
Table 8-20. EVE EOI Mapping....1516
Table 8-21. EVE to DSP1, DSP2 and MPU Mapping....1516
Table 8-22. EVE to Other Hosts Mapping....1517
Table 8-23. EVE to EVE Mapping....1517

Table 8-24. MISR Mapping.......... 1519
Table 8-25. Lock Register Mapping.......... 1521
Table 8-26. EVE Subsystem Memory Map.......... 1523
Table 8-27. VCOP IBUF Aliasing Truth Table.......... 1524
Table 8-28. Local EDMA IBUF Aliasing Truth Table.......... 1525
Table 8-29. SCTM Configuration in EVE.......... 1526
Table 8-30. SCTM Events.......... 1527
Table 8-31. SMSET Configuration in EVE.......... 1529
Table 8-32. List of SMSET Events.......... 1529
Table 8-33. EVE Instance Summary.......... 1533
Table 8-34. EVE Register Mapping Summary.......... 1534
Table 8-35. EVE_REVISION.......... 1537
Table 8-36. Register Call Summary for Register EVE_REVISION.......... 1538
Table 8-37. EVE_HWINFO.......... 1538
Table 8-38. Register Call Summary for Register EVE_HWINFO.......... 1538
Table 8-39. EVE_SYSCONFIG.......... 1538
Table 8-40. Register Call Summary for Register EVE_SYSCONFIG.......... 1539
Table 8-41. EVE_STAT.......... 1539
Table 8-42. Register Call Summary for Register EVE_STAT.......... 1541
Table 8-43. EVE_DISC_CONFIG.......... 1541
Table 8-44. Register Call Summary for Register EVE_DISC_CONFIG.......... 1542
Table 8-45. EVE_BUS_CONFIG.......... 1542
Table 8-46. Register Call Summary for Register EVE_BUS_CONFIG.......... 1542
Table 8-47. EVE_VCOP_HALT_CONFIG.......... 1542
Table 8-48. Register Call Summary for Register EVE_VCOP_HALT_CONFIG.......... 1543
Table 8-49. EVE_MMU_CONFIG.......... 1543
Table 8-50. Register Call Summary for Register EVE_MMU_CONFIG.......... 1544
Table 8-51. EVE_MEMMAP.......... 1544
Table 8-52. Register Call Summary for Register EVE_MEMMAP.......... 1545
Table 8-53. EVE_MSW_CTL.......... 1545
Table 8-54. Register Call Summary for Register EVE_MSW_CTL.......... 1546
Table 8-55. EVE_MSW_ERR.......... 1546
Table 8-56. Register Call Summary for Register EVE_MSW_ERR.......... 1547
Table 8-57. EVE_MSW_ERRADDR.......... 1547
Table 8-58. Register Call Summary for Register EVE_MSW_ERRADDR.......... 1548
Table 8-59. EVE_PC_INV.......... 1548
Table 8-60. Register Call Summary for Register EVE_PC_INV.......... 1548
Table 8-61. EVE_PC_IBAR.......... 1548
Table 8-62. Register Call Summary for Register EVE_PC_IBAR.......... 1549
Table 8-63. EVE_PC_IBC.......... 1549
Table 8-64. Register Call Summary for Register EVE_PC_IBC.......... 1549
Table 8-65. EVE_PC_ISAR.......... 1549
Table 8-66. Register Call Summary for Register EVE_PC_ISAR.......... 1550
Table 8-67. EVE_PC_ISAR_DONE.......... 1550
Table 8-68. Register Call Summary for Register EVE_PC_ISAR_DONE.......... 1550
Table 8-69. EVE_PC_PBAR.......... 1550
Table 8-70. Register Call Summary for Register EVE_PC_PBAR.......... 1551
Table 8-71. EVE_PC_PBC.......... 1551
Table 8-72. Register Call Summary for Register EVE_PC_PBC.......... 1551
Table 8-73. EVE_PMEM_ED_CTL.......... 1551
Table 8-74. Register Call Summary for Register EVE_PMEM_ED_CTL.......... 1552
Table 8-75. EVE_PMEM_ED_STAT.......... 1552
Table 8-76. Register Call Summary for Register EVE_PMEM_ED_STAT.......... 1552
Table 8-77. EVE_PMEM_EDADDR.......... 1553
Table 8-78. Register Call Summary for Register EVE_PMEM_EDADDR.......... 1553
Table 8-79. EVE_DMEM_ED_CTL.......... 1553
Table 8-80. Register Call Summary for Register EVE_DMEM_ED_CTL.......... 1553
Table 8-81. EVE_DMEM_ED_STAT.......... 1554
Table 8-82. Register Call Summary for Register EVE_DMEM_ED_STAT.......... 1554
Table 8-83. EVE_DMEM_EDADDR.......... 1554
Table 8-84. Register Call Summary for Register EVE_DMEM_EDADDR.......... 1555

Table 8-85. EVE_DMEM_EDADDR_BO... 1555
Table 8-86. Register Call Summary for Register EVE_DMEM_EDADDR_BO...1555
Table 8-87. EVE_WBUF_ED_CTL.. 1555
Table 8-88. Register Call Summary for Register EVE_WBUF_ED_CTL...1556
Table 8-89. EVE_WBUF_ED_STAT...1556
Table 8-90. Register Call Summary for Register EVE_WBUF_ED_STAT.. 1557
Table 8-91. EVE_WBUF_EDADDR... 1557
Table 8-92. Register Call Summary for Register EVE_WBUF_EDADDR... 1557
Table 8-93. EVE_WBUF_EDADDR_BO...1557
Table 8-94. Register Call Summary for Register EVE_WBUF_EDADDR_BO... 1558
Table 8-95. EVE_IBUF_ED_CTL...1558
Table 8-96. Register Call Summary for Register EVE_IBUF_ED_CTL... 1558
Table 8-97. EVE_IBUF_ED_STAT...1558
Table 8-98. Register Call Summary for Register EVE_IBUF_ED_STAT... 1559
Table 8-99. EVE_IBUF_EDADDR... 1559
Table 8-100. Register Call Summary for Register EVE_IBUF_EDADDR..1559
Table 8-101. EVE_IBUF_EDADDR_BO.. 1560
Table 8-102. Register Call Summary for Register EVE_IBUF_EDADDR_BO..1560
Table 8-103. EVE_ED_ARP32_DISC_EN...1560
Table 8-104. Register Call Summary for Register EVE_ED_ARP32_DISC_EN.. 1560
Table 8-105. EVE_ED_OCPI_DISC_EN.. 1560
Table 8-106. Register Call Summary for Register EVE_ED_OCPI_DISC_EN...1561
Table 8-107. EVE_MSW_ERR_IRQSTATUS_RAW...1561
Table 8-108. Register Call Summary for Register EVE_MSW_ERR_IRQSTATUS_RAW..1561
Table 8-109. EVE_MSW_ERR_IRQSTATUS.. 1561
Table 8-110. Register Call Summary for Register EVE_MSW_ERR_IRQSTATUS... 1562
Table 8-111. EVE_MSW_ERR_IRQENABLE_SET... 1562
Table 8-112. Register Call Summary for Register EVE_MSW_ERR_IRQENABLE_SET..1562
Table 8-113. EVE_MSW_ERR_IRQENABLE_CLR..1562
Table 8-114. Register Call Summary for Register EVE_MSW_ERR_IRQENABLE_CLR... 1563
Table 8-115. EVE_ED_LCL_IRQSTATUS_RAW.. 1563
Table 8-116. Register Call Summary for Register EVE_ED_LCL_IRQSTATUS_RAW...1563
Table 8-117. EVE_ED_LCL_IRQSTATUS... 1563
Table 8-118. Register Call Summary for Register EVE_ED_LCL_IRQSTATUS...1564
Table 8-119. EVE_ED_LCL_IRQENABLE_SET...1564
Table 8-120. Register Call Summary for Register EVE_ED_LCL_IRQENABLE_SET.. 1564
Table 8-121. EVE_ED_LCL_IRQENABLE_CLR.. 1564
Table 8-122. Register Call Summary for Register EVE_ED_LCL_IRQENABLE_CLR..1564
Table 8-123. ARP32_NMI_IRQSTATUS_RAW...1564
Table 8-124. Register Call Summary for Register ARP32_NMI_IRQSTATUS_RAW... 1565
Table 8-125. ARP32_NMI_IRQSTATUS...1565
Table 8-126. Register Call Summary for Register ARP32_NMI_IRQSTATUS... 1565
Table 8-127. ARP32_NMI_IRQENABLE_SET... 1565
Table 8-128. Register Call Summary for Register ARP32_NMI_IRQENABLE_SET...1566
Table 8-129. ARP32_NMI_IRQENABLE_CLR... 1566
Table 8-130. Register Call Summary for Register ARP32_NMI_IRQENABLE_CLR...1566
Table 8-131. ARP32_INTn_IRQSTATUS_RAW..1566
Table 8-132. Register Call Summary for Register ARP32_INTn_IRQSTATUS_RAW..1566
Table 8-133. ARP32_INTn_IRQSTATUS..1566
Table 8-134. Register Call Summary for Register ARP32_INTn_IRQSTATUS.. 1567
Table 8-135. ARP32_INTn_IRQENABLE_SET.. 1567
Table 8-136. Register Call Summary for Register ARP32_INTn_IRQENABLE_SET..1567
Table 8-137. ARP32_INTn_IRQENABLE_CLR.. 1567
Table 8-138. Register Call Summary for Register ARP32_INTn_IRQENABLE_CLR... 1568
Table 8-139. ARP32_IRQWAKEEN..1568
Table 8-140. Register Call Summary for Register ARP32_IRQWAKEEN..1568
Table 8-141. MMR_LOCKi...1568
Table 8-142. Register Call Summary for Register MMR_LOCKi.. 1569
Table 8-143. MISR_CTL.. 1569
Table 8-144. Register Call Summary for Register MISR_CTL...1569
Table 8-145. MISR_CLEAR..1569

Table 8-146. Register Call Summary for Register MISR_CLEAR........ 1569
Table 8-147. MISR0_A........ 1569
Table 8-148. Register Call Summary for Register MISR0_A........ 1570
Table 8-149. MISR0_D........ 1570
Table 8-150. Register Call Summary for Register MISR0_D........ 1570
Table 8-151. MISR1_A........ 1570
Table 8-152. Register Call Summary for Register MISR1_A........ 1571
Table 8-153. MISR1_D........ 1571
Table 8-154. Register Call Summary for Register MISR1_D........ 1571
Table 8-155. MISR2_Dk........ 1571
Table 8-156. Register Call Summary for Register MISR2_Dk........ 1572
Table 8-157. EVE_IRQ_EOI........ 1572
Table 8-158. Register Call Summary for Register EVE_IRQ_EOI........ 1572
Table 8-159. EVE_ED_OUT_IRQSTATUS_RAW........ 1572
Table 8-160. Register Call Summary for Register EVE_ED_OUT_IRQSTATUS_RAW........ 1573
Table 8-161. EVE_ED_OUT_IRQSTATUS........ 1573
Table 8-162. Register Call Summary for Register EVE_ED_OUT_IRQSTATUS........ 1573
Table 8-163. EVE_ED_OUT_IRQENABLE_SET........ 1573
Table 8-164. Register Call Summary for Register EVE_ED_OUT_IRQENABLE_SET........ 1574
Table 8-165. EVE_ED_OUT_IRQENABLE_CLR........ 1574
Table 8-166. Register Call Summary for Register EVE_ED_OUT_IRQENABLE_CLR........ 1574
Table 8-167. EVE_INTk_OUT_IRQSTATUS_RAW........ 1574
Table 8-168. Register Call Summary for Register EVE_INTk_OUT_IRQSTATUS_RAW........ 1574
Table 8-169. EVE_INTk_OUT_IRQSTATUS........ 1574
Table 8-170. Register Call Summary for Register EVE_INTk_OUT_IRQSTATUS........ 1575
Table 8-171. EVE_INTk_OUT_IRQENABLE_SET........ 1575
Table 8-172. Register Call Summary for Register EVE_INTk_OUT_IRQENABLE_SET........ 1575
Table 8-173. EVE_INTk_OUT_IRQENABLE_CLR........ 1575
Table 8-174. Register Call Summary for Register EVE_INTk_OUT_IRQENABLE_CLR........ 1576
Table 8-175. ARP32_INTj_IRQSTATUS_RAW........ 1576
Table 8-176. Register Call Summary for Register ARP32_INTj_IRQSTATUS_RAW........ 1576
Table 8-177. ARP32_INTj_IRQSTATUS........ 1576
Table 8-178. Register Call Summary for Register ARP32_INTj_IRQSTATUS........ 1576
Table 8-179. ARP32_INTj_IRQENABLE_SET........ 1577
Table 8-180. Register Call Summary for Register ARP32_INTj_IRQENABLE_SET........ 1577
Table 8-181. ARP32_INTj_IRQENABLE_CLR........ 1577
Table 8-182. Register Call Summary for Register ARP32_INTj_IRQENABLE_CLR........ 1577
Table 8-183. ARP32_INT14_IRQSTATUS_RAW........ 1577
Table 8-184. Register Call Summary for Register ARP32_INT14_IRQSTATUS_RAW........ 1578
Table 8-185. ARP32_INT14_IRQSTATUS........ 1578
Table 8-186. Register Call Summary for Register ARP32_INT14_IRQSTATUS........ 1578
Table 8-187. ARP32_INT14_IRQENABLE_SET........ 1578
Table 8-188. Register Call Summary for Register ARP32_INT14_IRQENABLE_SET........ 1578
Table 8-189. ARP32_INT14_IRQENABLE_CLR........ 1578
Table 8-190. Register Call Summary for Register ARP32_INT14_IRQENABLE_CLR........ 1579
Table 8-191. ARP32_INT15_IRQSTATUS_RAW........ 1579
Table 8-192. Register Call Summary for Register ARP32_INT15_IRQSTATUS_RAW........ 1579
Table 8-193. ARP32_INT15_IRQSTATUS........ 1579
Table 8-194. Register Call Summary for Register ARP32_INT15_IRQSTATUS........ 1580
Table 8-195. ARP32_INT15_IRQENABLE_SET........ 1580
Table 8-196. Register Call Summary for Register ARP32_INT15_IRQENABLE_SET........ 1580
Table 8-197. ARP32_INT15_IRQENABLE_CLR........ 1580
Table 8-198. Register Call Summary for Register ARP32_INT15_IRQENABLE_CLR........ 1580
Table 8-199. EVE_GPOUTm........ 1580
Table 8-200. Register Call Summary for Register EVE_GPOUTm........ 1581
Table 8-201. EVE_GPOUTm_SET........ 1581
Table 8-202. Register Call Summary for Register EVE_GPOUTm_SET........ 1581
Table 8-203. EVE_GPOUTm_CLR........ 1581
Table 8-204. Register Call Summary for Register EVE_GPOUTm_CLR........ 1582
Table 8-205. EVE_GPOUTm_PULSE........ 1582
Table 8-206. Register Call Summary for Register EVE_GPOUTm_PULSE........ 1582

Table 8-207. EVE_GPIN0 1582
Table 8-208. Register Call Summary for Register EVE_GPIN0 1583
Table 8-209. EVE_GPIN1 1583
Table 8-210. Register Call Summary for Register EVE_GPIN1 1583
Table 8-211. EVE_CME_DONE_GPOUT 1583
Table 8-212. Register Call Summary for Register EVE_CME_DONE_GPOUT 1583
Table 8-213. EVE_CME_DONE_GPOUT_SET 1583
Table 8-214. Register Call Summary for Register EVE_CME_DONE_GPOUT_SET 1584
Table 8-215. EVE_CME_DONE_GPOUT_CLR 1584
Table 8-216. Register Call Summary for Register EVE_CME_DONE_GPOUT_CLR 1584
Table 8-217. EVE_CME_DONE_GPOUT_PULSE 1584
Table 8-218. Register Call Summary for Register EVE_CME_DONE_GPOUT_PULSE 1585
Table 8-219. EVE_CME_DONE_SEL 1585
Table 8-220. Register Call Summary for Register EVE_CME_DONE_SEL 1585
Table 8-221. EVE_CME_DONE_EN 1586
Table 8-222. Register Call Summary for Register EVE_CME_DONE_EN 1586
Table 8-223. EVE_PM_STAT0 1586
Table 8-224. Register Call Summary for Register EVE_PM_STAT0 1587
Table 8-225. EVE_PM_STAT1 1587
Table 8-226. Register Call Summary for Register EVE_PM_STAT1 1588
Table 8-227. EVE_DBGOUT 1588
Table 8-228. Register Call Summary for Register EVE_DBGOUT 1588
Table 8-229. EVE_RSVD0 1589
Table 8-230. Register Call Summary for Register EVE_RSVD0 1589
Table 8-231. EVE_RSVD1 1589
Table 8-232. Register Call Summary for Register EVE_RSVD1 1589
Table 8-233. EVE_TEST 1589
Table 8-234. Register Call Summary for Register EVE_TEST 1590
Table 8-235. EVE L2_FNOC Register Summary 1590
Table 8-236. ERRLOGGER_i_ID_COREID 1590
Table 8-237. Register Call Summary for Register ERRLOGGER_i_ID_COREID 1591
Table 8-238. ERRLOGGER_i_ID_REVISIONID 1591
Table 8-239. Register Call Summary for Register ERRLOGGER_i_ID_REVISIONID 1591
Table 8-240. ERRLOGGER_i_FAULTEN 1591
Table 8-241. Register Call Summary for Register ERRLOGGER_i_FAULTEN 1592
Table 8-242. ERRLOGGER_i_ERRVLD 1592
Table 8-243. Register Call Summary for Register ERRLOGGER_i_ERRVLD 1592
Table 8-244. ERRLOGGER_i_ERRCLR 1592
Table 8-245. Register Call Summary for Register ERRLOGGER_i_ERRCLR 1592
Table 8-246. ERRLOGGER_i_ERRLOG0 1592
Table 8-247. Register Call Summary for Register ERRLOGGER_i_ERRLOG0 1593
Table 8-248. ERRLOGGER_i_ERRLOG1 1593
Table 8-249. Register Call Summary for Register ERRLOGGER_i_ERRLOG1 1593
Table 8-250. ERRLOGGER_i_ERRLOG3 1593
Table 8-251. Register Call Summary for Register ERRLOGGER_i_ERRLOG3 1593
Table 8-252. ERRLOGGER_i_ERRLOG5 1594
Table 8-253. Register Call Summary for Register ERRLOGGER_i_ERRLOG5 1594
Table 8-254. FLAGMUX_i_ID_COREID 1594
Table 8-255. Register Call Summary for Register FLAGMUX_i_ID_COREID 1594
Table 8-256. FLAGMUX_i_ID_REVISIONID 1594
Table 8-257. Register Call Summary for Register FLAGMUX_i_ID_REVISIONID 1595
Table 8-258. FLAGMUX_i_FAULTEN 1595
Table 8-259. Register Call Summary for Register FLAGMUX_i_FAULTEN 1595
Table 8-260. FLAGMUX_i_FAULTSTATUS 1595
Table 8-261. Register Call Summary for Register FLAGMUX_i_FAULTSTATUS 1595
Table 8-262. FLAGMUX_i_FLAGINEN0 1595
Table 8-263. Register Call Summary for Register FLAGMUX_i_FLAGINEN0 1596
Table 8-264. FLAGMUX_i_FLAGINSTATUS0 1596
Table 8-265. Register Call Summary for Register FLAGMUX_i_FLAGINSTATUS0 1596
Table 8-266. SCTM Configuration 1598
Table 8-267. List of SCTM Events 1602

Table 8-268. EVE_SCTM Instance Summary....1609
Table 8-269. EVE_SCTM Registers Mapping Summary 1....1609
Table 8-270. SCTM_CTCNTL....1609
Table 8-271. Register Call Summary for Register SCTM_CTCNTL....1610
Table 8-272. SCTM_CTSTMCNTL....1610
Table 8-273. Register Call Summary for Register SCTM_CTSTMCNTL....1611
Table 8-274. SCTM_CTSTMMSTID....1611
Table 8-275. Register Call Summary for Register SCTM_CTSTMMSTID....1611
Table 8-276. SCTM_CTSTMINTVL....1611
Table 8-277. Register Call Summary for Register SCTM_CTSTMINTVL....1611
Table 8-278. SCTM_CTSTMSEL....1612
Table 8-279. Register Call Summary for Register SCTM_CTSTMSEL....1612
Table 8-280. SCTM_TINTVLR_i....1612
Table 8-281. Register Call Summary for Register SCTM_TINTVLR_i....1612
Table 8-282. SCTM_CTDBGNUM....1612
Table 8-283. Register Call Summary for Register SCTM_CTDBGNUM....1613
Table 8-284. SCTM_CTDBGEVT....1613
Table 8-285. Register Call Summary for Register SCTM_CTDBGEVT....1613
Table 8-286. SCTM_CTGNBL....1613
Table 8-287. Register Call Summary for Register SCTM_CTGNBL....1613
Table 8-288. SCTM_CTGRST....1613
Table 8-289. Register Call Summary for Register SCTM_CTGRST....1614
Table 8-290. SCTM_CTCR_WT_m....1614
Table 8-291. Register Call Summary for Register SCTM_CTCR_WT_m....1614
Table 8-292. SCTM_CTCR_WOT_n....1614
Table 8-293. Register Call Summary for Register SCTM_CTCR_WOT_n....1615
Table 8-294. SCTM_CTCNTR_k....1615
Table 8-295. Register Call Summary for Register SCTM_CTCNTR_k....1615
Table 8-296. SMSET Configuration....1616
Table 8-297. List of SMSET Events....1618
Table 8-298. Ownership Commands....1620
Table 8-299. SMSET Instance Summary....1621
Table 8-300. EVE_SMSET Registers Mapping Summary 1....1621
Table 8-301. SMSET_ID....1621
Table 8-302. Register Call Summary for Register SMSET_ID....1622
Table 8-303. SMSET_SCFG....1622
Table 8-304. Register Call Summary for Register SMSET_SCFG....1622
Table 8-305. SMSET_SR....1622
Table 8-306. Register Call Summary for Register SMSET_SR....1623
Table 8-307. SMSET_CFG....1623
Table 8-308. Register Call Summary for Register SMSET_CFG....1624
Table 8-309. SMSET_SESW....1624
Table 8-310. Register Call Summary for Register SMSET_SESW....1624
Table 8-311. SMSET_SEDEN_i....1624
Table 8-312. Register Call Summary for Register SMSET_SEDEN_i....1625
Table 8-313. Interface Signals....1627
Table 8-314. Control Registers....1634
Table 8-315. Control Status Register (CSR) Field Descriptions....1635
Table 8-316. Interrupt Enable Register (IER) Field Descriptions....1636
Table 8-317. Interrupt Flag Register (IFR) Field Descriptions....1637
Table 8-318. Interrupt Set Register (ISR) Field Descriptions....1638
Table 8-319. Interrupt Clear Register (ICR) Field Descriptions....1639
Table 8-320. NMI Return Pointer Register (NRP) Field Descriptions....1640
Table 8-321. Interrupt Return Pointer Register (IRP) Field Descriptions....1640
Table 8-322. Link Register (LR) Field Descriptions....1641
Table 8-323. Loop 0 Start Address Register (LSA0) Field Descriptions....1642
Table 8-324. Loop 0 End Address Register (LEA0) Field Descriptions....1642
Table 8-325. Loop 0 Iteration Count Register (LCNT0) Field Descriptions....1643
Table 8-326. Loop 1 Start Address Register (LSA1) Field Descriptions....1644
Table 8-327. Loop 1 End Address Register (LEA1) Field Descriptions....1644
Table 8-328. Loop 1 Iteration Count Register (LCNT1) Field Descriptions....1645

Table 8-329. Loop 0 Iteration Count Reload Value Register (LCNT0RLD) Field Descriptions... 1645
Table 8-330. Shadow Control Status Register (SCSR) Field Descriptions..1646
Table 8-331. NMI Shadow Control Status Register (NMISCSR) Field Descriptions..1647
Table 8-332. CPU Identification Register (CPUID) Field Descriptions...1648
Table 8-333. Decode Program Counter Register (DPC) Field Descriptions.. 1649
Table 8-334. CPU Shadow Registers.. 1651
Table 8-335. Hardware Loop Control Registers...1656
Table 8-336. Example 1 of Generated Assembly Code (relevant instructions only)..1659
Table 8-337. Example 2 of Generated Assembly Code (relevant instructions only)..1659
Table 8-338. Example 1 of Generated Assembly Code (relevant instructions only)..1660
Table 8-339. Example 2 of Generated Assembly Code (relevant instructions only)..1660
Table 8-340. Example of Generated Assembly Code (relevant instructions only)..1661
Table 8-341. Example of Generated Assembly Code (relevant instructions only)..1662
Table 8-342. Interrupt Summary.. 1663
Table 8-343. cpu_inum_o Values...1666
Table 8-344. Interrupt Priorities..1666
Table 8-345. Interrupt Service Table (IST)... 1667
Table A-346. Instruction Pseudo Code Notations.. 1676
Table A-347. Instruction Syntax and Opcode Notations.. 1677
Table A-348. Instruction Summary...1678
Table A-349. CPU Reset Types..1815
Table A-350. CPU Reset Modes...1815
Table 8-351. Performance Counter Signal List.. 1822
Table 8-352. EVE Vector Data Memory Map... 1823
Table 8-353. VLD Data Distribution Options.. 1834
Table 8-354. VCOP Arithmetic/Logic Operations...1838
Table 8-355. Example of Operation Delay Slots.. 1840
Table 8-356. EVE ST Data Distribution Options for NWAY = 8...1842
Table 8-357. Lookup Constraints for 8-Way SIMD...1844
Table 8-358. Table Lookup and Histogram Hardware Resources... 1851
Table 8-359. Load and Store Address Alignment Constraints... 1853
Table 8-360. Load and Store Buffering Example, Byte-Type Horizontal Filter...1854
Table 8-361. Load and Store Buffering Example, Short-Type Horizontal Filter... 1855
Table 8-362. Parameter Indexed by rnd_param Field Descriptions...1868
Table 8-363. VCOP Instance Summary...1874
Table 8-364. VCOP Registers Mapping Summary 1... 1874
Table 8-365. VCOP_PID..1875
Table 8-366. Register Call Summary for Register VCOP_PID.. 1875
Table 8-367. VCOP_CTRL.. 1875
Table 8-368. Register Call Summary for Register VCOP_CTRL...1875
Table 8-369. VCOP_STATUS.. 1875
Table 8-370. Register Call Summary for Register VCOP_STATUS...1876
Table 8-371. VCOP_MAX_ITERS... 1876
Table 8-372. Register Call Summary for Register VCOP_MAX_ITERS...1876
Table 8-373. VCOP_ERROR..1876
Table 8-374. Register Call Summary for Register VCOP_ERROR.. 1878
Table 8-375. VCOP_VLOOP_PTR.. 1878
Table 8-376. Register Call Summary for Register VCOP_VLOOP_PTR..1878
Table 8-377. VCOP_PARAM_PTR.. 1878
Table 8-378. Register Call Summary for Register VCOP_PARAM_PTR..1879
Table 8-379. VCOP_I0_I1..1879
Table 8-380. Register Call Summary for Register VCOP_I0_I1... 1879
Table 8-381. VCOP_I2_I3..1879
Table 8-382. Register Call Summary for Register VCOP_I2_I3... 1879
Table 8-383. VCOP_I4...1879
Table 8-384. Register Call Summary for Register VCOP_I4.. 1880
Table 8-385. VCOP_LD_PTR_i... 1880
Table 8-386. Register Call Summary for Register VCOP_LD_PTR_i...1880
Table 8-387. VCOP_ST_PTR_j.. 1880
Table 8-388. Register Call Summary for Register VCOP_ST_PTR_j...1881
Table 9-1. VIP1 Interface Signals.. 1886

Table 9-2. VIP2 Interface Signals....1887
Table 9-3. VIP3 Interface Signals....1888
Table 9-4. VIP Port A Input Data Signals to RGB and YUV Color Components Mapping....1890
Table 9-5. VIP Port B Input Data Signals to YUV Color Components Mapping....1890
Table 9-6. VIP Integration Attributes....1892
Table 9-7. VIP Clocks and Resets....1893
Table 9-8. VIP Hardware Requests....1893
Table 9-9. VIP Slice Processing Path Control....1896
Table 9-10. Polarity Table for FID Determination By VSYNC Skew....1920
Table 9-11. Fourth Byte of EAV/SAV Code Word....1924
Table 9-12. Error Correction Matrix....1925
Table 9-13. Multiplexing Configurations and Pixel Clock Rates....1927
Table 9-14. Split Line Table....1930
Table 9-15. Meta Data Layout....1930
Table 9-16. TI Line Mux Mode Channel ID Remapping....1931
Table 9-17. Channel ID Embedded in EAV/SAV....1932
Table 9-18. Valid Embedded Sync Mux Mode and Data Bus Width Combinations....1933
Table 9-19. VIP_PARSER Interrupt Events....1936
Table 9-20. Quantized Coefficients of HDTV Application with Video Data Range....1947
Table 9-21. Quantized Coefficients of HDTV Application with Graphics Data Range....1947
Table 9-22. Quantized Coefficients of SDTV Application with Video Data Range....1949
Table 9-23. Quantized Coefficients of SDTV Application with Graphics Data Range....1950
Table 9-24. Parameter Description....1953
Table 9-25. Vertical Scaler Configuration Parameters....1955
Table 9-26. Register Group 1....1959
Table 9-27. Register Group 2....1959
Table 9-28. Register Group 3....1959
Table 9-29. Scaler Configuration....1960
Table 9-30. Vertical Scaler Configuration....1960
Table 9-31. Coefficient Data Files....1963
Table 9-32. VPDMA Client Buffering and Functionality....1983
Table 9-33. VPDMA Channels Assignment....1985
Table 9-34. VPDMA Interrupt Events....1990
Table 9-35. VIP Interrupt Sources....1991
Table 9-36. Data Packet Descriptor Word 0 Field Descriptions....2007
Table 9-37. Common ARGB in Memory (Byte Order)....2008
Table 9-38. Common ARGB in 32-bit Memory/CPU Register....2008
Table 9-39. VPDMA ARGB in Memory (Byte Order)....2009
Table 9-40. VPDMA ARGB in 32-bit Memory/CPU Register....2009
Table 9-41. VPDMA Descriptor RGB Data Type Mapping....2009
Table 9-42. VPDMA Descriptor YUV Data Type Mapping....2010
Table 9-43. Data Packet Descriptor Word 1 Field Description....2011
Table 9-44. Data Packet Descriptor Word 2 Field Descriptions....2011
Table 9-45. Data Packet Descriptor Word 3 Field Descriptions....2011
Table 9-46. Data Packet Descriptor Word 4 Inbound Data Field Descriptions....2012
Table 9-47. Data Packet Descriptor Word 4 Outbound Data Field Descriptions....2013
Table 9-48. Data Packet Descriptor Word 5 Outbound Data Field Descriptions....2014
Table 9-49. Configuration Descriptor Header Word0 Field Descriptions....2015
Table 9-50. Configuration Descriptor Header Word1 Field Descriptions....2015
Table 9-51. Configuration Descriptor Header Word2 Field Descriptions....2015
Table 9-52. Configuration Descriptor Header Word3 Field Descriptions....2015
Table 9-53. Address Data Block Format Field Descriptions....2016
Table 9-54. Destination Field Description....2016
Table 9-55. Control Descriptor Header Description....2017
Table 9-56. Control Descriptor Types Summary....2017
Table 9-57. Sync on Client Field Descriptions (Word - 1)....2017
Table 9-58. Sync on Client Field Descriptions (Word - 3)....2018
Table 9-59. Sync on List Field Descriptions (Word - 3)....2018
Table 9-60. Sync on External Event Field Descriptions (Word - 3)....2018
Table 9-61. Sync on Channel Field Descriptions (Word - 3)....2019
Table 9-62. Change Client Interrupt Field Descriptions (Word - 1)....2019

Table 9-63. Change Client Interrupt Field Descriptions (Word - 2)........ 2019
Table 9-64. Change Client Interrupt Field Descriptions (Word - 3)........ 2019
Table 9-65. Send Interrupt Field Descriptions (Word - 3)........ 2019
Table 9-66. Reload List Field Descriptions (Word - 0)........ 2020
Table 9-67. Reload List Field Descriptions (Word - 1)........ 2020
Table 9-68. Reload List Field Descriptions (Word - 3)........ 2020
Table 9-69. Abort Channel Field Descriptions (Word - 3)........ 2020
Table 9-70. VIP Instance Summary........ 2031
Table 9-71. VIP Top Level Registers Mapping Summary........ 2032
Table 9-72. VIP_CLKC_PID........ 2032
Table 9-73. Register Call Summary for Register VIP_CLKC_PID........ 2033
Table 9-74. VIP_SYSCONFIG........ 2033
Table 9-75. Register Call Summary for Register VIP_SYSCONFIG........ 2034
Table 9-76. VIP_INTC_INTR0_STATUS_RAW0........ 2034
Table 9-77. Register Call Summary for Register VIP_INTC_INTR0_STATUS_RAW0........ 2036
Table 9-78. VIP_INTC_INTR0_STATUS_RAW1........ 2036
Table 9-79. Register Call Summary for Register VIP_INTC_INTR0_STATUS_RAW1........ 2037
Table 9-80. VIP_INTC_INTR0_STATUS_ENA0........ 2037
Table 9-81. Register Call Summary for Register VIP_INTC_INTR0_STATUS_ENA0........ 2039
Table 9-82. VIP_INTC_INTR0_STATUS_ENA1........ 2039
Table 9-83. Register Call Summary for Register VIP_INTC_INTR0_STATUS_ENA1........ 2040
Table 9-84. VIP_INTC_INTR0_ENA_SET0........ 2040
Table 9-85. Register Call Summary for Register VIP_INTC_INTR0_ENA_SET0........ 2042
Table 9-86. VIP_INTC_INTR0_ENA_SET1........ 2042
Table 9-87. Register Call Summary for Register VIP_INTC_INTR0_ENA_SET1........ 2043
Table 9-88. VIP_INTC_INTR0_ENA_CLR0........ 2043
Table 9-89. Register Call Summary for Register VIP_INTC_INTR0_ENA_CLR0........ 2045
Table 9-90. VIP_INTC_INTR0_ENA_CLR1........ 2045
Table 9-91. Register Call Summary for Register VIP_INTC_INTR0_ENA_CLR1........ 2047
Table 9-92. VIP_INTC_INTR1_STATUS_RAW0........ 2047
Table 9-93. Register Call Summary for Register VIP_INTC_INTR1_STATUS_RAW0........ 2048
Table 9-94. VIP_INTC_INTR1_STATUS_RAW1........ 2048
Table 9-95. Register Call Summary for Register VIP_INTC_INTR1_STATUS_RAW1........ 2050
Table 9-96. VIP_INTC_INTR1_STATUS_ENA0........ 2050
Table 9-97. Register Call Summary for Register VIP_INTC_INTR1_STATUS_ENA0........ 2051
Table 9-98. VIP_INTC_INTR1_STATUS_ENA1........ 2051
Table 9-99. Register Call Summary for Register VIP_INTC_INTR1_STATUS_ENA1........ 2053
Table 9-100. VIP_INTC_INTR1_ENA_SET0........ 2053
Table 9-101. Register Call Summary for Register VIP_INTC_INTR1_ENA_SET0........ 2054
Table 9-102. VIP_INTC_INTR1_ENA_SET1........ 2054
Table 9-103. Register Call Summary for Register VIP_INTC_INTR1_ENA_SET1........ 2056
Table 9-104. VIP_INTC_INTR1_ENA_CLR0........ 2056
Table 9-105. Register Call Summary for Register VIP_INTC_INTR1_ENA_CLR0........ 2058
Table 9-106. VIP_INTC_INTR1_ENA_CLR1........ 2058
Table 9-107. Register Call Summary for Register VIP_INTC_INTR1_ENA_CLR1........ 2059
Table 9-108. VIP_INTC_EOI........ 2059
Table 9-109. Register Call Summary for Register VIP_INTC_EOI........ 2060
Table 9-110. VIP_CLKC_CLKEN........ 2060
Table 9-111. Register Call Summary for Register VIP_CLKC_CLKEN........ 2060
Table 9-112. VIP_CLKC_RST........ 2060
Table 9-113. Register Call Summary for Register VIP_CLKC_RST........ 2061
Table 9-114. VIP_CLKC_DPS........ 2061
Table 9-115. Register Call Summary for Register VIP_CLKC_DPS........ 2062
Table 9-116. VIP_CLKC_VIP0DPS........ 2062
Table 9-117. Register Call Summary for Register VIP_CLKC_VIP0DPS........ 2064
Table 9-118. VIP_CLKC_VIP1DPS........ 2064
Table 9-119. Register Call Summary for Register VIP_CLKC_VIP1DPS........ 2065
Table 9-120. VIP Parser Registers Mapping Summary 1........ 2066
Table 9-121. VIP Parser Registers Mapping Summary 2........ 2067
Table 9-122. VIP_MAIN........ 2068
Table 9-123. VIP_PORT_A........ 2069

Table 9-124. VIP_XTRA_PORT_A........ 2071
Table 9-125. VIP_PORT_B........ 2071
Table 9-126. VIP_XTRA_PORT_B........ 2073
Table 9-127. VIP_FIQ_MASK........ 2074
Table 9-128. VIP_FIQ_CLEAR........ 2075
Table 9-129. VIP_FIQ_STATUS........ 2077
Table 9-130. VIP_OUTPUT_PORT_A_SRC_FID........ 2078
Table 9-131. VIP_OUTPUT_PORT_A_ENC_FID........ 2080
Table 9-132. VIP_OUTPUT_PORT_B_SRC_FID........ 2081
Table 9-133. VIP_OUTPUT_PORT_B_ENC_FID........ 2083
Table 9-134. VIP_OUTPUT_PORT_A_SRC0_SIZE........ 2085
Table 9-135. VIP_OUTPUT_PORT_A_SRC1_SIZE........ 2085
Table 9-136. VIP_OUTPUT_PORT_A_SRC2_SIZE........ 2085
Table 9-137. VIP_OUTPUT_PORT_A_SRC3_SIZE........ 2086
Table 9-138. VIP_OUTPUT_PORT_A_SRC4_SIZE........ 2086
Table 9-139. VIP_OUTPUT_PORT_A_SRC5_SIZE........ 2087
Table 9-140. VIP_OUTPUT_PORT_A_SRC6_SIZE........ 2087
Table 9-141. VIP_OUTPUT_PORT_A_SRC7_SIZE........ 2087
Table 9-142. VIP_OUTPUT_PORT_A_SRC8_SIZE........ 2088
Table 9-143. VIP_OUTPUT_PORT_A_SRC9_SIZE........ 2088
Table 9-144. VIP_OUTPUT_PORT_A_SRC10_SIZE........ 2088
Table 9-145. VIP_OUTPUT_PORT_A_SRC11_SIZE........ 2089
Table 9-146. VIP_OUTPUT_PORT_A_SRC12_SIZE........ 2089
Table 9-147. VIP_OUTPUT_PORT_A_SRC13_SIZE........ 2090
Table 9-148. VIP_OUTPUT_PORT_A_SRC14_SIZE........ 2090
Table 9-149. VIP_OUTPUT_PORT_A_SRC15_SIZE........ 2090
Table 9-150. VIP_OUTPUT_PORT_B_SRC0_SIZE........ 2091
Table 9-151. VIP_OUTPUT_PORT_B_SRC1_SIZE........ 2091
Table 9-152. VIP_OUTPUT_PORT_B_SRC2_SIZE........ 2091
Table 9-153. VIP_OUTPUT_PORT_B_SRC3_SIZE........ 2092
Table 9-154. VIP_OUTPUT_PORT_B_SRC4_SIZE........ 2092
Table 9-155. VIP_OUTPUT_PORT_B_SRC5_SIZE........ 2093
Table 9-156. VIP_OUTPUT_PORT_B_SRC6_SIZE........ 2093
Table 9-157. VIP_OUTPUT_PORT_B_SRC7_SIZE........ 2093
Table 9-158. VIP_OUTPUT_PORT_B_SRC8_SIZE........ 2094
Table 9-159. VIP_OUTPUT_PORT_B_SRC9_SIZE........ 2094
Table 9-160. VIP_OUTPUT_PORT_B_SRC10_SIZE........ 2094
Table 9-161. VIP_OUTPUT_PORT_B_SRC11_SIZE........ 2095
Table 9-162. VIP_OUTPUT_PORT_B_SRC12_SIZE........ 2095
Table 9-163. VIP_OUTPUT_PORT_B_SRC13_SIZE........ 2096
Table 9-164. VIP_OUTPUT_PORT_B_SRC14_SIZE........ 2096
Table 9-165. VIP_OUTPUT_PORT_B_SRC15_SIZE........ 2096
Table 9-166. VIP_PORT_A_VDET_VEC........ 2097
Table 9-167. VIP_PORT_B_VDET_VEC........ 2097
Table 9-168. VIP_ANC_CROP_HORZ_PORT_A........ 2097
Table 9-169. VIP_ANC_CROP_VERT_PORT_A........ 2098
Table 9-170. VIP_CROP_HORZ_PORT_A........ 2098
Table 9-171. VIP_CROP_VERT_PORT_A........ 2099
Table 9-172. VIP_ANC_VIP_CROP_HORZ_PORT_B........ 2099
Table 9-173. VIP_ANC_VIP_CROP_VERT_PORT_B........ 2100
Table 9-174. VIP_CROP_HORZ_PORT_B........ 2100
Table 9-175. VIP_CROP_VERT_PORT_B........ 2101
Table 9-176. VIP_XTRA6_PORT_A........ 2101
Table 9-177. VIP_XTRA7_PORT_B........ 2102
Table 9-178. VIP_XTRA8_PORT_A........ 2103
Table 9-179. VIP_XTRA9_PORT_B........ 2103
Table 9-180. VIP CSC Registers Mapping Summary 1........ 2104
Table 9-181. VIP CSC Registers Mapping Summary 2........ 2104
Table 9-182. VIP CSC Registers Mapping Summary 3........ 2104
Table 9-183. VIP_CSC00........ 2104
Table 9-184. Register Call Summary for Register VIP_CSC00........ 2105

Table 9-185. VIP_CSC01....2105
Table 9-186. Register Call Summary for Register VIP_CSC01....2105
Table 9-187. VIP_CSC02....2106
Table 9-188. Register Call Summary for Register VIP_CSC02....2106
Table 9-189. VIP_CSC03....2106
Table 9-190. Register Call Summary for Register VIP_CSC03....2107
Table 9-191. VIP_CSC04....2107
Table 9-192. Register Call Summary for Register VIP_CSC04....2107
Table 9-193. VIP_CSC05....2107
Table 9-194. Register Call Summary for Register VIP_CSC05....2108
Table 9-195. VIP SC Registers Mapping Summary 1....2108
Table 9-196. VIP SC Registers Mapping Summary 2....2109
Table 9-197. VIP SC Registers Mapping Summary 3....2110
Table 9-198. VIP_CFG_SC0....2110
Table 9-199. Register Call Summary for Register VIP_CFG_SC0....2112
Table 9-200. VIP_CFG_SC1....2112
Table 9-201. Register Call Summary for Register VIP_CFG_SC1....2113
Table 9-202. VIP_CFG_SC2....2113
Table 9-203. Register Call Summary for Register VIP_CFG_SC2....2113
Table 9-204. VIP_CFG_SC3....2113
Table 9-205. Register Call Summary for Register VIP_CFG_SC3....2114
Table 9-206. VIP_CFG_SC4....2114
Table 9-207. Register Call Summary for Register VIP_CFG_SC4....2114
Table 9-208. VIP_CFG_SC5....2115
Table 9-209. Register Call Summary for Register VIP_CFG_SC5....2115
Table 9-210. VIP_CFG_SC6....2115
Table 9-211. Register Call Summary for Register VIP_CFG_SC6....2116
Table 9-212. VIP_CFG_SC8....2116
Table 9-213. Register Call Summary for Register VIP_CFG_SC8....2116
Table 9-214. VIP_CFG_SC9....2117
Table 9-215. Register Call Summary for Register VIP_CFG_SC9....2117
Table 9-216. VIP_CFG_SC10....2117
Table 9-217. Register Call Summary for Register VIP_CFG_SC10....2118
Table 9-218. VIP_CFG_SC11....2118
Table 9-219. Register Call Summary for Register VIP_CFG_SC11....2118
Table 9-220. VIP_CFG_SC12....2118
Table 9-221. Register Call Summary for Register VIP_CFG_SC12....2119
Table 9-222. VIP_CFG_SC13....2119
Table 9-223. Register Call Summary for Register VIP_CFG_SC13....2119
Table 9-224. VIP_CFG_SC18....2119
Table 9-225. Register Call Summary for Register VIP_CFG_SC18....2120
Table 9-226. VIP_CFG_SC19....2120
Table 9-227. Register Call Summary for Register VIP_CFG_SC19....2120
Table 9-228. VIP_CFG_SC20....2120
Table 9-229. Register Call Summary for Register VIP_CFG_SC20....2121
Table 9-230. VIP_CFG_SC21....2121
Table 9-231. Register Call Summary for Register VIP_CFG_SC21....2121
Table 9-232. VIP_CFG_SC22....2122
Table 9-233. Register Call Summary for Register VIP_CFG_SC22....2122
Table 9-234. VIP_CFG_SC24....2122
Table 9-235. Register Call Summary for Register VIP_CFG_SC24....2123
Table 9-236. VIP_CFG_SC25....2123
Table 9-237. Register Call Summary for Register VIP_CFG_SC25....2123
Table 9-238. VIP VPDMA Registers Mapping Summary....2124
Table 9-239. VIP_PID....2128
Table 9-240. Register Call Summary for Register VIP_PID....2128
Table 9-241. VIP_LIST_ADDR....2128
Table 9-242. Register Call Summary for Register VIP_LIST_ADDR....2128
Table 9-243. VIP_LIST_ATTR....2128
Table 9-244. Register Call Summary for Register VIP_LIST_ATTR....2129
Table 9-245. VIP_LIST_STAT_SYNC....2129

Table 9-246. Register Call Summary for Register VIP_LIST_STAT_SYNC........ 2131
Table 9-247. VIP_BG_RGB........ 2131
Table 9-248. Register Call Summary for Register VIP_BG_RGB........ 2131
Table 9-249. VIP_BG_YUV........ 2131
Table 9-250. Register Call Summary for Register VIP_BG_YUV........ 2131
Table 9-251. VIP_VPDMA_SETUP........ 2132
Table 9-252. Register Call Summary for Register VIP_VPDMA_SETUP........ 2132
Table 9-253. VIP_MAX_SIZE1........ 2132
Table 9-254. Register Call Summary for Register VIP_MAX_SIZE1........ 2132
Table 9-255. VIP_MAX_SIZE2........ 2132
Table 9-256. Register Call Summary for Register VIP_MAX_SIZE2........ 2133
Table 9-257. VIP_MAX_SIZE3........ 2133
Table 9-258. Register Call Summary for Register VIP_MAX_SIZE3........ 2133
Table 9-259. VIP_INT0_CHANNEL0_INT_STAT........ 2133
Table 9-260. Register Call Summary for Register VIP_INT0_CHANNEL0_INT_STAT........ 2136
Table 9-261. VIP_INT0_CHANNEL0_INT_MASK........ 2136
Table 9-262. Register Call Summary for Register VIP_INT0_CHANNEL0_INT_MASK........ 2137
Table 9-263. VIP_INT0_CHANNEL1_INT_STAT........ 2137
Table 9-264. Register Call Summary for Register VIP_INT0_CHANNEL1_INT_STAT........ 2141
Table 9-265. VIP_INT0_CHANNEL1_INT_MASK........ 2141
Table 9-266. Register Call Summary for Register VIP_INT0_CHANNEL1_INT_MASK........ 2143
Table 9-267. VIP_INT0_CHANNEL2_INT_STAT........ 2143
Table 9-268. Register Call Summary for Register VIP_INT0_CHANNEL2_INT_STAT........ 2148
Table 9-269. VIP_INT0_CHANNEL2_INT_MASK........ 2148
Table 9-270. Register Call Summary for Register VIP_INT0_CHANNEL2_INT_MASK........ 2151
Table 9-271. VIP_INT0_CHANNEL3_INT_STAT........ 2151
Table 9-272. Register Call Summary for Register VIP_INT0_CHANNEL3_INT_STAT........ 2156
Table 9-273. VIP_INT0_CHANNEL3_INT_MASK........ 2156
Table 9-274. Register Call Summary for Register VIP_INT0_CHANNEL3_INT_MASK........ 2158
Table 9-275. VIP_INT0_CHANNEL4_INT_STAT........ 2158
Table 9-276. Register Call Summary for Register VIP_INT0_CHANNEL4_INT_STAT........ 2163
Table 9-277. VIP_INT0_CHANNEL4_INT_MASK........ 2163
Table 9-278. Register Call Summary for Register VIP_INT0_CHANNEL4_INT_MASK........ 2166
Table 9-279. VIP_INT0_CHANNEL5_INT_STAT........ 2166
Table 9-280. Register Call Summary for Register VIP_INT0_CHANNEL5_INT_STAT........ 2170
Table 9-281. VIP_INT0_CHANNEL5_INT_MASK........ 2170
Table 9-282. Register Call Summary for Register VIP_INT0_CHANNEL5_INT_MASK........ 2173
Table 9-283. VIP_INT0_CLIENT0_INT_STAT........ 2173
Table 9-284. Register Call Summary for Register VIP_INT0_CLIENT0_INT_STAT........ 2175
Table 9-285. VIP_INT0_CLIENT0_INT_MASK........ 2176
Table 9-286. Register Call Summary for Register VIP_INT0_CLIENT0_INT_MASK........ 2177
Table 9-287. VIP_INT0_CLIENT1_INT_STAT........ 2177
Table 9-288. Register Call Summary for Register VIP_INT0_CLIENT1_INT_STAT........ 2182
Table 9-289. VIP_INT0_CLIENT1_INT_MASK........ 2182
Table 9-290. Register Call Summary for Register VIP_INT0_CLIENT1_INT_MASK........ 2184
Table 9-291. VIP_INT0_LIST0_INT_STAT........ 2184
Table 9-292. Register Call Summary for Register VIP_INT0_LIST0_INT_STAT........ 2187
Table 9-293. VIP_INT0_LIST0_INT_MASK........ 2187
Table 9-294. Register Call Summary for Register VIP_INT0_LIST0_INT_MASK........ 2190
Table 9-295. VIP_INT1_CHANNEL0_INT_STAT........ 2190
Table 9-296. Register Call Summary for Register VIP_INT1_CHANNEL0_INT_STAT........ 2192
Table 9-297. VIP_INT1_CHANNEL0_INT_MASK........ 2192
Table 9-298. Register Call Summary for Register VIP_INT1_CHANNEL0_INT_MASK........ 2193
Table 9-299. VIP_INT1_CHANNEL1_INT_STAT........ 2194
Table 9-300. Register Call Summary for Register VIP_INT1_CHANNEL1_INT_STAT........ 2197
Table 9-301. VIP_INT1_CHANNEL1_INT_MASK........ 2197
Table 9-302. Register Call Summary for Register VIP_INT1_CHANNEL1_INT_MASK........ 2199
Table 9-303. VIP_INT1_CHANNEL2_INT_STAT........ 2199
Table 9-304. Register Call Summary for Register VIP_INT1_CHANNEL2_INT_STAT........ 2204
Table 9-305. VIP_INT1_CHANNEL2_INT_MASK........ 2204
Table 9-306. Register Call Summary for Register VIP_INT1_CHANNEL2_INT_MASK........ 2207

Table 9-307. VIP_INT1_CHANNEL3_INT_STAT....2207
Table 9-308. Register Call Summary for Register VIP_INT1_CHANNEL3_INT_STAT....2212
Table 9-309. VIP_INT1_CHANNEL3_INT_MASK....2212
Table 9-310. Register Call Summary for Register VIP_INT1_CHANNEL3_INT_MASK....2214
Table 9-311. VIP_INT1_CHANNEL4_INT_STAT....2214
Table 9-312. Register Call Summary for Register VIP_INT1_CHANNEL4_INT_STAT....2219
Table 9-313. VIP_INT1_CHANNEL4_INT_MASK....2219
Table 9-314. Register Call Summary for Register VIP_INT1_CHANNEL4_INT_MASK....2222
Table 9-315. VIP_INT1_CHANNEL5_INT_STAT....2222
Table 9-316. Register Call Summary for Register VIP_INT1_CHANNEL5_INT_STAT....2226
Table 9-317. VIP_INT1_CHANNEL5_INT_MASK....2226
Table 9-318. Register Call Summary for Register VIP_INT1_CHANNEL5_INT_MASK....2229
Table 9-319. VIP_INT1_CLIENT0_INT_STAT....2229
Table 9-320. Register Call Summary for Register VIP_INT1_CLIENT0_INT_STAT....2231
Table 9-321. VIP_INT1_CLIENT0_INT_MASK....2232
Table 9-322. Register Call Summary for Register VIP_INT1_CLIENT0_INT_MASK....2233
Table 9-323. VIP_INT1_CLIENT1_INT_STAT....2233
Table 9-324. Register Call Summary for Register VIP_INT1_CLIENT1_INT_STAT....2238
Table 9-325. VIP_INT1_CLIENT1_INT_MASK....2238
Table 9-326. Register Call Summary for Register VIP_INT1_CLIENT1_INT_MASK....2240
Table 9-327. VIP_INT1_LIST0_INT_STAT....2240
Table 9-328. Register Call Summary for Register VIP_INT1_LIST0_INT_STAT....2243
Table 9-329. VIP_INT1_LIST0_INT_MASK....2243
Table 9-330. Register Call Summary for Register VIP_INT1_LIST0_INT_MASK....2246
Table 9-331. VIP_PERF_MON0....2246
Table 9-332. Register Call Summary for Register VIP_PERF_MON0....2246
Table 9-333. VIP_PERF_MON1....2246
Table 9-334. Register Call Summary for Register VIP_PERF_MON1....2247
Table 9-335. VIP_PERF_MON2....2247
Table 9-336. Register Call Summary for Register VIP_PERF_MON2....2248
Table 9-337. VIP_PERF_MON3....2248
Table 9-338. Register Call Summary for Register VIP_PERF_MON3....2249
Table 9-339. VIP_PERF_MON4....2249
Table 9-340. Register Call Summary for Register VIP_PERF_MON4....2249
Table 9-341. VIP_PERF_MON5....2249
Table 9-342. Register Call Summary for Register VIP_PERF_MON5....2250
Table 9-343. VIP_PERF_MON6....2250
Table 9-344. Register Call Summary for Register VIP_PERF_MON6....2251
Table 9-345. VIP_PERF_MON7....2251
Table 9-346. Register Call Summary for Register VIP_PERF_MON7....2251
Table 9-347. VIP_PERF_MON8....2252
Table 9-348. Register Call Summary for Register VIP_PERF_MON8....2252
Table 9-349. VIP_PERF_MON9....2252
Table 9-350. Register Call Summary for Register VIP_PERF_MON9....2253
Table 9-351. VIP_PERF_MON10....2253
Table 9-352. Register Call Summary for Register VIP_PERF_MON10....2254
Table 9-353. VIP_PERF_MON11....2254
Table 9-354. Register Call Summary for Register VIP_PERF_MON11....2254
Table 9-355. VIP_PERF_MON12....2254
Table 9-356. Register Call Summary for Register VIP_PERF_MON12....2255
Table 9-357. VIP_PERF_MON13....2255
Table 9-358. Register Call Summary for Register VIP_PERF_MON13....2256
Table 9-359. VIP_PERF_MON14....2256
Table 9-360. Register Call Summary for Register VIP_PERF_MON14....2257
Table 9-361. VIP_PERF_MON15....2257
Table 9-362. Register Call Summary for Register VIP_PERF_MON15....2257
Table 9-363. VIP_PERF_MON16....2257
Table 9-364. Register Call Summary for Register VIP_PERF_MON16....2258
Table 9-365. VIP_PERF_MON17....2258
Table 9-366. Register Call Summary for Register VIP_PERF_MON17....2259
Table 9-367. VIP_PERF_MON18....2259

Table 9-368. Register Call Summary for Register VIP_PERF_MON18....2260
Table 9-369. VIP_PERF_MON19....2260
Table 9-370. Register Call Summary for Register VIP_PERF_MON19....2260
Table 9-371. VIP_PERF_MON20....2260
Table 9-372. Register Call Summary for Register VIP_PERF_MON20....2261
Table 9-373. VIP_PERF_MON21....2261
Table 9-374. Register Call Summary for Register VIP_PERF_MON21....2262
Table 9-375. VIP_PERF_MON22....2262
Table 9-376. Register Call Summary for Register VIP_PERF_MON22....2262
Table 9-377. VIP_PERF_MON23....2263
Table 9-378. Register Call Summary for Register VIP_PERF_MON23....2263
Table 9-379. VIP_PERF_MON24....2263
Table 9-380. Register Call Summary for Register VIP_PERF_MON24....2264
Table 9-381. VIP_PERF_MON25....2264
Table 9-382. Register Call Summary for Register VIP_PERF_MON25....2265
Table 9-383. VIP_PERF_MON26....2265
Table 9-384. Register Call Summary for Register VIP_PERF_MON26....2265
Table 9-385. VIP_PERF_MON27....2265
Table 9-386. Register Call Summary for Register VIP_PERF_MON27....2266
Table 9-387. VIP_PERF_MON28....2266
Table 9-388. Register Call Summary for Register VIP_PERF_MON28....2267
Table 9-389. VIP_PERF_MON29....2267
Table 9-390. Register Call Summary for Register VIP_PERF_MON29....2268
Table 9-391. VIP_PERF_MON30....2268
Table 9-392. Register Call Summary for Register VIP_PERF_MON30....2268
Table 9-393. VIP_PERF_MON31....2268
Table 9-394. Register Call Summary for Register VIP_PERF_MON31....2269
Table 9-395. VIP_PERF_MON32....2269
Table 9-396. Register Call Summary for Register VIP_PERF_MON32....2270
Table 9-397. VIP_PERF_MON33....2270
Table 9-398. Register Call Summary for Register VIP_PERF_MON33....2271
Table 9-399. VIP_PERF_MON34....2271
Table 9-400. Register Call Summary for Register VIP_PERF_MON34....2271
Table 9-401. VIP_PERF_MON35....2271
Table 9-402. Register Call Summary for Register VIP_PERF_MON35....2272
Table 9-403. VIP_PERF_MON36....2272
Table 9-404. Register Call Summary for Register VIP_PERF_MON36....2273
Table 9-405. VIP_PERF_MON37....2273
Table 9-406. Register Call Summary for Register VIP_PERF_MON37....2274
Table 9-407. VIP_PERF_MON38....2274
Table 9-408. Register Call Summary for Register VIP_PERF_MON38....2274
Table 9-409. VIP_PERF_MON39....2274
Table 9-410. Register Call Summary for Register VIP_PERF_MON39....2275
Table 9-411. VIP_PERF_MON40....2275
Table 9-412. Register Call Summary for Register VIP_PERF_MON40....2276
Table 9-413. VIP_PERF_MON41....2276
Table 9-414. Register Call Summary for Register VIP_PERF_MON41....2277
Table 9-415. VIP_PERF_MON42....2277
Table 9-416. Register Call Summary for Register VIP_PERF_MON42....2277
Table 9-417. VIP_PERF_MON43....2277
Table 9-418. Register Call Summary for Register VIP_PERF_MON43....2278
Table 9-419. VIP_PERF_MON44....2278
Table 9-420. Register Call Summary for Register VIP_PERF_MON44....2279
Table 9-421. VIP_PERF_MON45....2279
Table 9-422. Register Call Summary for Register VIP_PERF_MON45....2280
Table 9-423. VIP_PERF_MON46....2280
Table 9-424. Register Call Summary for Register VIP_PERF_MON46....2280
Table 9-425. VIP_PERF_MON47....2280
Table 9-426. Register Call Summary for Register VIP_PERF_MON47....2281
Table 9-427. VIP_PERF_MON48....2281
Table 9-428. Register Call Summary for Register VIP_PERF_MON48....2282

Table 9-429. VIP_PERF_MON49 2282
Table 9-430. Register Call Summary for Register VIP_PERF_MON49 2283
Table 9-431. VIP_PERF_MON50 2283
Table 9-432. Register Call Summary for Register VIP_PERF_MON50 2283
Table 9-433. VIP_PERF_MON51 2283
Table 9-434. Register Call Summary for Register VIP_PERF_MON51 2284
Table 9-435. VIP_PERF_MON52 2284
Table 9-436. Register Call Summary for Register VIP_PERF_MON52 2285
Table 9-437. VIP_PERF_MON53 2285
Table 9-438. Register Call Summary for Register VIP_PERF_MON53 2285
Table 9-439. VIP_PERF_MON54 2286
Table 9-440. Register Call Summary for Register VIP_PERF_MON54 2286
Table 9-441. VIP_PERF_MON55 2286
Table 9-442. Register Call Summary for Register VIP_PERF_MON55 2287
Table 9-443. VIP_PERF_MON56 2287
Table 9-444. Register Call Summary for Register VIP_PERF_MON56 2288
Table 9-445. VIP_PERF_MON57 2288
Table 9-446. Register Call Summary for Register VIP_PERF_MON57 2288
Table 9-447. VIP_PERF_MON58 2288
Table 9-448. Register Call Summary for Register VIP_PERF_MON58 2289
Table 9-449. VIP_PERF_MON59 2289
Table 9-450. Register Call Summary for Register VIP_PERF_MON59 2290
Table 9-451. VIP_PERF_MON60 2290
Table 9-452. Register Call Summary for Register VIP_PERF_MON60 2291
Table 9-453. VIP_PERF_MON61 2291
Table 9-454. Register Call Summary for Register VIP_PERF_MON61 2291
Table 9-455. VIP0_LO_Y_CSTAT 2291
Table 9-456. Register Call Summary for Register VIP0_LO_Y_CSTAT 2292
Table 9-457. VIP0_LO_UV_CSTAT 2292
Table 9-458. Register Call Summary for Register VIP0_LO_UV_CSTAT 2293
Table 9-459. VIP0_UP_Y_CSTAT 2293
Table 9-460. Register Call Summary for Register VIP0_UP_Y_CSTAT 2294
Table 9-461. VIP0_UP_UV_CSTAT 2294
Table 9-462. Register Call Summary for Register VIP0_UP_UV_CSTAT 2295
Table 9-463. VIP1_LO_Y_CSTAT 2295
Table 9-464. Register Call Summary for Register VIP1_LO_Y_CSTAT 2296
Table 9-465. VIP1_LO_UV_CSTAT 2296
Table 9-466. Register Call Summary for Register VIP1_LO_UV_CSTAT 2297
Table 9-467. VIP1_UP_Y_CSTAT 2297
Table 9-468. Register Call Summary for Register VIP1_UP_Y_CSTAT 2298
Table 9-469. VIP1_UP_UV_CSTAT 2298
Table 9-470. Register Call Summary for Register VIP1_UP_UV_CSTAT 2298
Table 9-471. VPI_CTL_CSTAT 2299
Table 9-472. Register Call Summary for Register VPI_CTL_CSTAT 2299
Table 9-473. VIP0_ANC_A_CSTAT 2299
Table 9-474. Register Call Summary for Register VIP0_ANC_A_CSTAT 2300
Table 9-475. VIP0_ANC_B_CSTAT 2300
Table 9-476. Register Call Summary for Register VIP0_ANC_B_CSTAT 2301
Table 9-477. VIP1_ANC_A_CSTAT 2301
Table 9-478. Register Call Summary for Register VIP1_ANC_A_CSTAT 2302
Table 9-479. VIP1_ANC_B_CSTAT 2302
Table 9-480. Register Call Summary for Register VIP1_ANC_B_CSTAT 2303
Table 10-1. VPE Integration Attributes 2307
Table 10-2. VPE Clocks and Resets 2307
Table 10-3. VPE Hardware Requests 2307
Table 10-4. Parameter Description 2320
Table 10-5. Vertical Scaler Configuration Parameters 2322
Table 10-6. Register Group 1 2326
Table 10-7. Register Group 2 2326
Table 10-8. Register Group 3 2326
Table 10-9. Scaler Configuration 2327

Table 10-10. Vertical Scaler Configuration....2327
Table 10-11. Coefficient Data Files....2330
Table 10-12. Quantized Coefficients of HDTV Application with Video Data Range....2351
Table 10-13. Quantized Coefficients of HDTV Application with Graphics Data Range....2351
Table 10-14. Quantized Coefficients of SDTV Application with Video Data Range....2353
Table 10-15. Quantized Coefficients of SDTV Application with Graphics Data Range....2354
Table 10-16. VPDMA Modes of Operation....2357
Table 10-17. VPDMA Client Buffering and Functionality....2361
Table 10-18. VPDMA Channels Assignment....2362
Table 10-19. VPDMA Interrupt Events....2363
Table 10-20. VPE Interrupt Sources....2364
Table 10-21. Data Packet Descriptor Word 0 Field Descriptions....2369
Table 10-22. Common ARGB in Memory (Byte Order)....2370
Table 10-23. Common ARGB in 32-bit Memory/CPU Register....2370
Table 10-24. VPDMA ARGB in Memory (Byte Order)....2371
Table 10-25. VPDMA ARGB in 32-bit Memory/CPU Register....2371
Table 10-26. VPDMA Descriptor RGB Data Type Mapping....2371
Table 10-27. VPDMA Descriptor YUV Data Type Mapping....2372
Table 10-28. Data Packet Descriptor Word 1 Field Description....2373
Table 10-29. Data Packet Descriptor Word 2 Field Descriptions....2373
Table 10-30. Data Packet Descriptor Word 3 Field Descriptions....2374
Table 10-31. Data Packet Descriptor Word 4 Inbound Data Field Descriptions....2374
Table 10-32. Data Packet Descriptor Word 4 Outbound Data Field Descriptions....2375
Table 10-33. Data Packet Descriptor Word 5 Outbound Data Field Descriptions....2376
Table 10-34. Configuration Descriptor Header Word0 Field Descriptions....2377
Table 10-35. Configuration Descriptor Header Word1 Field Descriptions....2377
Table 10-36. Configuration Descriptor Header Word2 Field Descriptions....2377
Table 10-37. Configuration Descriptor Header Word3 Field Descriptions....2378
Table 10-38. Address Data Block Format Field Descriptions....2378
Table 10-39. Destination Field Description....2379
Table 10-40. Control Descriptor Header Description....2379
Table 10-41. Control Descriptor Types Summary....2379
Table 10-42. Sync on Client Field Descriptions (Word - 1)....2380
Table 10-43. Sync on Client Field Descriptions (Word - 3)....2380
Table 10-44. Sync on List Field Descriptions (Word - 3)....2380
Table 10-45. Sync on External Event Field Descriptions (Word - 3)....2381
Table 10-46. Sync on Channel Field Descriptions (Word - 3)....2381
Table 10-47. Change Client Interrupt Field Descriptions (Word - 1)....2381
Table 10-48. Change Client Interrupt Field Descriptions (Word - 2)....2381
Table 10-49. Change Client Interrupt Field Descriptions (Word - 3)....2382
Table 10-50. Send Interrupt Field Descriptions (Word - 3)....2382
Table 10-51. Reload List Field Descriptions (Word - 0)....2382
Table 10-52. Reload List Field Descriptions (Word - 1)....2382
Table 10-53. Reload List Field Descriptions (Word - 3)....2382
Table 10-54. Abort Channel Field Descriptions (Word - 3)....2383
Table 10-55. VPE Instance Summary....2400
Table 10-56. VPE_CSC Registers Mapping Summary....2400
Table 10-57. VPE_CSC00....2400
Table 10-58. Register Call Summary for Register VPE_CSC00....2401
Table 10-59. VPE_CSC01....2401
Table 10-60. Register Call Summary for Register VPE_CSC01....2402
Table 10-61. VPE_CSC02....2402
Table 10-62. Register Call Summary for Register VPE_CSC02....2402
Table 10-63. VPE_CSC03....2402
Table 10-64. Register Call Summary for Register VPE_CSC03....2403
Table 10-65. VPE_CSC04....2403
Table 10-66. Register Call Summary for Register VPE_CSC04....2403
Table 10-67. VPE_CSC05....2403
Table 10-68. Register Call Summary for Register VPE_CSC05....2404
Table 10-69. VPE_SC Registers Mapping Summary....2404
Table 10-70. VPE_CFG_SC0....2405

Table 10-71. Register Call Summary for Register VPE_CFG_SC0..2407
Table 10-72. VPE_CFG_SC1..2407
Table 10-73. Register Call Summary for Register VPE_CFG_SC1..2407
Table 10-74. VPE_CFG_SC2..2408
Table 10-75. Register Call Summary for Register VPE_CFG_SC2..2408
Table 10-76. VPE_CFG_SC3..2408
Table 10-77. Register Call Summary for Register VPE_CFG_SC3..2408
Table 10-78. VPE_CFG_SC4..2408
Table 10-79. Register Call Summary for Register VPE_CFG_SC4..2409
Table 10-80. VPE_CFG_SC5..2409
Table 10-81. Register Call Summary for Register VPE_CFG_SC5..2410
Table 10-82. VPE_CFG_SC6..2410
Table 10-83. Register Call Summary for Register VPE_CFG_SC6..2410
Table 10-84. VPE_CFG_SC8..2410
Table 10-85. Register Call Summary for Register VPE_CFG_SC8..2411
Table 10-86. VPE_CFG_SC9..2411
Table 10-87. Register Call Summary for Register VPE_CFG_SC9..2411
Table 10-88. VPE_CFG_SC10..2411
Table 10-89. Register Call Summary for Register VPE_CFG_SC10..2412
Table 10-90. VPE_CFG_SC11..2412
Table 10-91. Register Call Summary for Register VPE_CFG_SC11..2412
Table 10-92. VPE_CFG_SC12..2412
Table 10-93. Register Call Summary for Register VPE_CFG_SC12..2413
Table 10-94. VPE_CFG_SC13..2413
Table 10-95. Register Call Summary for Register VPE_CFG_SC13..2413
Table 10-96. VPE_CFG_SC18..2413
Table 10-97. Register Call Summary for Register VPE_CFG_SC18..2414
Table 10-98. VPE_CFG_SC19..2414
Table 10-99. Register Call Summary for Register VPE_CFG_SC19..2414
Table 10-100. VPE_CFG_SC20..2414
Table 10-101. Register Call Summary for Register VPE_CFG_SC20..2415
Table 10-102. VPE_CFG_SC21..2415
Table 10-103. Register Call Summary for Register VPE_CFG_SC21..2415
Table 10-104. VPE_CFG_SC22..2415
Table 10-105. Register Call Summary for Register VPE_CFG_SC22..2416
Table 10-106. VPE_CFG_SC24..2416
Table 10-107. Register Call Summary for Register VPE_CFG_SC24..2416
Table 10-108. VPE_CFG_SC25..2416
Table 10-109. Register Call Summary for Register VPE_CFG_SC25..2417
Table 10-110. VPE_CHR_US Registers Mapping Summary..2417
Table 10-111. VPE_PID..2417
Table 10-112. Register Call Summary for Register VPE_PID..2417
Table 10-113. VPE_REG0..2417
Table 10-114. Register Call Summary for Register VPE_REG0..2418
Table 10-115. VPE_REG1..2418
Table 10-116. Register Call Summary for Register VPE_REG1..2418
Table 10-117. VPE_REG2..2418
Table 10-118. Register Call Summary for Register VPE_REG2..2419
Table 10-119. VPE_REG3..2419
Table 10-120. Register Call Summary for Register VPE_REG3..2419
Table 10-121. VPE_REG4..2419
Table 10-122. Register Call Summary for Register VPE_REG4..2420
Table 10-123. VPE_REG5..2420
Table 10-124. Register Call Summary for Register VPE_REG5..2420
Table 10-125. VPE_REG6..2420
Table 10-126. Register Call Summary for Register VPE_REG6..2421
Table 10-127. VPE_REG7..2421
Table 10-128. Register Call Summary for Register VPE_REG7..2421
Table 10-129. VPE_DEI Registers Mapping Summary..2421
Table 10-130. VPE_DEI_REG0..2422
Table 10-131. Register Call Summary for Register VPE_DEI_REG0..2422

Table 10-132. VPE_DEI_REG1....2423
Table 10-133. Register Call Summary for Register VPE_DEI_REG1....2423
Table 10-134. VPE_DEI_REG2....2423
Table 10-135. Register Call Summary for Register VPE_DEI_REG2....2424
Table 10-136. VPE_DEI_REG3....2424
Table 10-137. Register Call Summary for Register VPE_DEI_REG3....2425
Table 10-138. VPE_DEI_REG4....2425
Table 10-139. Register Call Summary for Register VPE_DEI_REG4....2425
Table 10-140. VPE_DEI_REG5....2425
Table 10-141. Register Call Summary for Register VPE_DEI_REG5....2426
Table 10-142. VPE_DEI_REG6....2426
Table 10-143. Register Call Summary for Register VPE_DEI_REG6....2426
Table 10-144. VPE_DEI_REG7....2426
Table 10-145. Register Call Summary for Register VPE_DEI_REG7....2427
Table 10-146. VPE_DEI_REG8....2427
Table 10-147. Register Call Summary for Register VPE_DEI_REG8....2427
Table 10-148. VPE_DEI_REG9....2428
Table 10-149. Register Call Summary for Register VPE_DEI_REG9....2428
Table 10-150. VPE_DEI_REG10....2428
Table 10-151. Register Call Summary for Register VPE_DEI_REG10....2429
Table 10-152. VPE_DEI_REG11....2429
Table 10-153. Register Call Summary for Register VPE_DEI_REG11....2429
Table 10-154. VPE_DEI_REG12....2429
Table 10-155. Register Call Summary for Register VPE_DEI_REG12....2430
Table 10-156. VPE_DEI_REG13....2430
Table 10-157. Register Call Summary for Register VPE_DEI_REG13....2430
Table 10-158. VPE_DEI_REG14....2430
Table 10-159. Register Call Summary for Register VPE_DEI_REG14....2431
Table 10-160. VPE_VPDMA Registers Mapping Summary....2432
Table 10-161. VPE_VPDMA_PID....2435
Table 10-162. Register Call Summary for Register VPE_VPDMA_PID....2436
Table 10-163. VPE_LIST_ADDR....2436
Table 10-164. Register Call Summary for Register VPE_LIST_ADDR....2436
Table 10-165. VPE_LIST_ATTR....2437
Table 10-166. Register Call Summary for Register VPE_LIST_ATTR....2437
Table 10-167. VPE_LIST_STAT_SYNC....2437
Table 10-168. Register Call Summary for Register VPE_LIST_STAT_SYNC....2439
Table 10-169. VPE_BG_RGB....2439
Table 10-170. Register Call Summary for Register VPE_BG_RGB....2439
Table 10-171. VPE_BG_YUV....2439
Table 10-172. Register Call Summary for Register VPE_BG_YUV....2440
Table 10-173. VPE_VPDMA_SETUP....2440
Table 10-174. Register Call Summary for Register VPE_VPDMA_SETUP....2440
Table 10-175. VPE_MAX_SIZE1....2440
Table 10-176. Register Call Summary for Register VPE_MAX_SIZE1....2441
Table 10-177. VPE_MAX_SIZE2....2441
Table 10-178. Register Call Summary for Register VPE_MAX_SIZE2....2441
Table 10-179. VPE_MAX_SIZE3....2441
Table 10-180. Register Call Summary for Register VPE_MAX_SIZE3....2442
Table 10-181. VPE_INT0_CHANNEL0_INT_STAT....2442
Table 10-182. Register Call Summary for Register VPE_INT0_CHANNEL0_INT_STAT....2444
Table 10-183. VPE_INT0_CHANNEL0_INT_MASK....2444
Table 10-184. Register Call Summary for Register VPE_INT0_CHANNEL0_INT_MASK....2445
Table 10-185. VPE_INT0_CHANNEL1_INT_STAT....2445
Table 10-186. Register Call Summary for Register VPE_INT0_CHANNEL1_INT_STAT....2450
Table 10-187. VPE_INT0_CHANNEL1_INT_MASK....2450
Table 10-188. Register Call Summary for Register VPE_INT0_CHANNEL1_INT_MASK....2452
Table 10-189. VPE_INT0_CHANNEL2_INT_STAT....2452
Table 10-190. Register Call Summary for Register VPE_INT0_CHANNEL2_INT_STAT....2457
Table 10-191. VPE_INT0_CHANNEL2_INT_MASK....2457
Table 10-192. Register Call Summary for Register VPE_INT0_CHANNEL2_INT_MASK....2460

Table 10-193. VPE_INT0_CHANNEL3_INT_STAT... 2460
Table 10-194. Register Call Summary for Register VPE_INT0_CHANNEL3_INT_STAT..2465
Table 10-195. VPE_INT0_CHANNEL3_INT_MASK...2465
Table 10-196. Register Call Summary for Register VPE_INT0_CHANNEL3_INT_MASK.. 2468
Table 10-197. VPE_INT0_CHANNEL4_INT_STAT... 2468
Table 10-198. Register Call Summary for Register VPE_INT0_CHANNEL4_INT_STAT..2473
Table 10-199. VPE_INT0_CHANNEL4_INT_MASK...2473
Table 10-200. Register Call Summary for Register VPE_INT0_CHANNEL4_INT_MASK.. 2476
Table 10-201. VPE_INT0_CHANNEL5_INT_STAT... 2476
Table 10-202. Register Call Summary for Register VPE_INT0_CHANNEL5_INT_STAT..2481
Table 10-203. VPE_INT0_CHANNEL5_INT_MASK...2481
Table 10-204. Register Call Summary for Register VPE_INT0_CHANNEL5_INT_MASK.. 2484
Table 10-205. VPE_INT0_CLIENT0_INT_STAT...2484
Table 10-206. Register Call Summary for Register VPE_INT0_CLIENT0_INT_STAT.. 2486
Table 10-207. VPE_INT0_CLIENT0_INT_MASK... 2487
Table 10-208. Register Call Summary for Register VPE_INT0_CLIENT0_INT_MASK...2488
Table 10-209. VPE_INT0_CLIENT1_INT_STAT...2488
Table 10-210. Register Call Summary for Register VPE_INT0_CLIENT1_INT_STAT.. 2493
Table 10-211. VPE_INT0_CLIENT1_INT_MASK...2493
Table 10-212. Register Call Summary for Register VPE_INT0_CLIENT1_INT_MASK...2495
Table 10-213. VPE_INT0_LIST0_INT_STAT..2495
Table 10-214. Register Call Summary for Register VPE_INT0_LIST0_INT_STAT... 2498
Table 10-215. VPE_INT0_LIST0_INT_MASK.. 2498
Table 10-216. Register Call Summary for Register VPE_INT0_LIST0_INT_MASK...2501
Table 10-217. VPE_PERF_MON0...2501
Table 10-218. Register Call Summary for Register VPE_PERF_MON0.. 2502
Table 10-219. VPE_PERF_MON1...2502
Table 10-220. Register Call Summary for Register VPE_PERF_MON1.. 2503
Table 10-221. VPE_PERF_MON2...2503
Table 10-222. Register Call Summary for Register VPE_PERF_MON2.. 2504
Table 10-223. VPE_PERF_MON3...2504
Table 10-224. Register Call Summary for Register VPE_PERF_MON3.. 2506
Table 10-225. VPE_PERF_MON4...2506
Table 10-226. Register Call Summary for Register VPE_PERF_MON4.. 2507
Table 10-227. VPE_PERF_MON5...2507
Table 10-228. Register Call Summary for Register VPE_PERF_MON5.. 2508
Table 10-229. VPE_PERF_MON6...2508
Table 10-230. Register Call Summary for Register VPE_PERF_MON6.. 2510
Table 10-231. VPE_PERF_MON7...2510
Table 10-232. Register Call Summary for Register VPE_PERF_MON7...2511
Table 10-233. VPE_PERF_MON8.. 2511
Table 10-234. Register Call Summary for Register VPE_PERF_MON8.. 2512
Table 10-235. VPE_PERF_MON9...2512
Table 10-236. Register Call Summary for Register VPE_PERF_MON9.. 2514
Table 10-237. VPE_PERF_MON10...2514
Table 10-238. Register Call Summary for Register VPE_PERF_MON10.. 2515
Table 10-239. VPE_PERF_MON11.. 2515
Table 10-240. Register Call Summary for Register VPE_PERF_MON11..2516
Table 10-241. VPE_PERF_MON12...2516
Table 10-242. Register Call Summary for Register VPE_PERF_MON12.. 2518
Table 10-243. VPE_PERF_MON13...2518
Table 10-244. Register Call Summary for Register VPE_PERF_MON13.. 2519
Table 10-245. VPE_PERF_MON14...2519
Table 10-246. Register Call Summary for Register VPE_PERF_MON14.. 2520
Table 10-247. VPE_PERF_MON15...2520
Table 10-248. Register Call Summary for Register VPE_PERF_MON15...2522
Table 10-249. VPE_PERF_MON16...2522
Table 10-250. Register Call Summary for Register VPE_PERF_MON16.. 2523
Table 10-251. VPE_PERF_MON17...2523
Table 10-252. Register Call Summary for Register VPE_PERF_MON17.. 2524
Table 10-253. VPE_PERF_MON18...2524

Table 10-254. Register Call Summary for Register VPE_PERF_MON18........ 2526
Table 10-255. VPE_PERF_MON19........ 2526
Table 10-256. Register Call Summary for Register VPE_PERF_MON19........ 2527
Table 10-257. VPE_PERF_MON20........ 2527
Table 10-258. Register Call Summary for Register VPE_PERF_MON20........ 2528
Table 10-259. VPE_PERF_MON21........ 2528
Table 10-260. Register Call Summary for Register VPE_PERF_MON21........ 2530
Table 10-261. VPE_PERF_MON22........ 2530
Table 10-262. Register Call Summary for Register VPE_PERF_MON22........ 2531
Table 10-263. VPE_PERF_MON23........ 2531
Table 10-264. Register Call Summary for Register VPE_PERF_MON23........ 2532
Table 10-265. VPE_PERF_MON24........ 2532
Table 10-266. Register Call Summary for Register VPE_PERF_MON24........ 2534
Table 10-267. VPE_PERF_MON25........ 2534
Table 10-268. Register Call Summary for Register VPE_PERF_MON25........ 2535
Table 10-269. VPE_PERF_MON26........ 2535
Table 10-270. Register Call Summary for Register VPE_PERF_MON26........ 2536
Table 10-271. VPE_PERF_MON27........ 2536
Table 10-272. Register Call Summary for Register VPE_PERF_MON27........ 2538
Table 10-273. VPE_PERF_MON28........ 2538
Table 10-274. Register Call Summary for Register VPE_PERF_MON28........ 2539
Table 10-275. VPE_PERF_MON29........ 2539
Table 10-276. Register Call Summary for Register VPE_PERF_MON29........ 2540
Table 10-277. VPE_PERF_MON30........ 2540
Table 10-278. Register Call Summary for Register VPE_PERF_MON30........ 2542
Table 10-279. VPE_PERF_MON31........ 2542
Table 10-280. Register Call Summary for Register VPE_PERF_MON31........ 2543
Table 10-281. VPE_PERF_MON32........ 2543
Table 10-282. Register Call Summary for Register VPE_PERF_MON32........ 2544
Table 10-283. VPE_PERF_MON33........ 2544
Table 10-284. Register Call Summary for Register VPE_PERF_MON33........ 2546
Table 10-285. VPE_PERF_MON34........ 2546
Table 10-286. Register Call Summary for Register VPE_PERF_MON34........ 2547
Table 10-287. VPE_PERF_MON35........ 2547
Table 10-288. Register Call Summary for Register VPE_PERF_MON35........ 2548
Table 10-289. VPE_PERF_MON36........ 2548
Table 10-290. Register Call Summary for Register VPE_PERF_MON36........ 2550
Table 10-291. VPE_PERF_MON37........ 2550
Table 10-292. Register Call Summary for Register VPE_PERF_MON37........ 2551
Table 10-293. VPE_PERF_MON38........ 2551
Table 10-294. Register Call Summary for Register VPE_PERF_MON38........ 2552
Table 10-295. VPE_PERF_MON39........ 2552
Table 10-296. Register Call Summary for Register VPE_PERF_MON39........ 2554
Table 10-297. VPE_PERF_MON40........ 2554
Table 10-298. Register Call Summary for Register VPE_PERF_MON40........ 2555
Table 10-299. VPE_PERF_MON41........ 2555
Table 10-300. Register Call Summary for Register VPE_PERF_MON41........ 2556
Table 10-301. VPE_PERF_MON42........ 2556
Table 10-302. Register Call Summary for Register VPE_PERF_MON42........ 2557
Table 10-303. VPE_PERF_MON43........ 2557
Table 10-304. Register Call Summary for Register VPE_PERF_MON43........ 2558
Table 10-305. VPE_PERF_MON44........ 2558
Table 10-306. Register Call Summary for Register VPE_PERF_MON44........ 2559
Table 10-307. VPE_PERF_MON45........ 2559
Table 10-308. Register Call Summary for Register VPE_PERF_MON45........ 2560
Table 10-309. VPE_PERF_MON46........ 2560
Table 10-310. Register Call Summary for Register VPE_PERF_MON46........ 2561
Table 10-311. VPE_PERF_MON47........ 2561
Table 10-312. Register Call Summary for Register VPE_PERF_MON47........ 2562
Table 10-313. VPE_PERF_MON48........ 2563
Table 10-314. Register Call Summary for Register VPE_PERF_MON48........ 2564

Table 10-315. VPE_PERF_MON49....2564
Table 10-316. Register Call Summary for Register VPE_PERF_MON49....2565
Table 10-317. VPE_PERF_MON50....2565
Table 10-318. Register Call Summary for Register VPE_PERF_MON50....2566
Table 10-319. VPE_PERF_MON51....2566
Table 10-320. Register Call Summary for Register VPE_PERF_MON51....2567
Table 10-321. VPE_PERF_MON52....2567
Table 10-322. Register Call Summary for Register VPE_PERF_MON52....2568
Table 10-323. VPE_PRI_CHROMA_CSTAT....2568
Table 10-324. Register Call Summary for Register VPE_PRI_CHROMA_CSTAT....2570
Table 10-325. VPE_PRI_LUMA_CSTAT....2570
Table 10-326. Register Call Summary for Register VPE_PRI_LUMA_CSTAT....2570
Table 10-327. VPE_PRI_FLD1_LUMA_CSTAT....2571
Table 10-328. Register Call Summary for Register VPE_PRI_FLD1_LUMA_CSTAT....2571
Table 10-329. VPE_PRI_FLD1_CHROMA_CSTAT....2572
Table 10-330. Register Call Summary for Register VPE_PRI_FLD1_CHROMA_CSTAT....2573
Table 10-331. VPE_PRI_FLD2_LUMA_CSTAT....2573
Table 10-332. Register Call Summary for Register VPE_PRI_FLD2_LUMA_CSTAT....2574
Table 10-333. VPE_PRI_FLD2_CHROMA_CSTAT....2574
Table 10-334. Register Call Summary for Register VPE_PRI_FLD2_CHROMA_CSTAT....2575
Table 10-335. VPE_PRI_MV0_CSTAT....2575
Table 10-336. Register Call Summary for Register VPE_PRI_MV0_CSTAT....2576
Table 10-337. VPE_PRI_MV_OUT_CSTAT....2576
Table 10-338. Register Call Summary for Register VPE_PRI_MV_OUT_CSTAT....2577
Table 10-339. VPE_VIP0_UP_Y_CSTAT....2577
Table 10-340. Register Call Summary for Register VPE_VIP0_UP_Y_CSTAT....2578
Table 10-341. VPE_VIP0_UP_UV_CSTAT....2578
Table 10-342. Register Call Summary for Register VPE_VIP0_UP_UV_CSTAT....2579
Table 10-343. VPE_VPI_CTL_CSTAT....2579
Table 10-344. Register Call Summary for Register VPE_VPI_CTL_CSTAT....2580
Table 10-345. VPE Registers Mapping Summary....2580
Table 10-346. VPE_CLKC_PID....2580
Table 10-347. Register Call Summary for Register VPE_CLKC_PID....2581
Table 10-348. VPE_SYSCONFIG....2581
Table 10-349. Register Call Summary for Register VPE_SYSCONFIG....2581
Table 10-350. VPE_INTC_INTR0_STATUS_RAW0....2582
Table 10-351. Register Call Summary for Register VPE_INTC_INTR0_STATUS_RAW0....2583
Table 10-352. VPE_INTC_INTR0_STATUS_RAW1....2583
Table 10-353. Register Call Summary for Register VPE_INTC_INTR0_STATUS_RAW1....2584
Table 10-354. VPE_INTC_INTR0_STATUS_ENA0....2584
Table 10-355. Register Call Summary for Register VPE_INTC_INTR0_STATUS_ENA0....2586
Table 10-356. VPE_INTC_INTR0_STATUS_ENA1....2586
Table 10-357. Register Call Summary for Register VPE_INTC_INTR0_STATUS_ENA1....2587
Table 10-358. VPE_INTC_INTR0_ENA_SET0....2587
Table 10-359. Register Call Summary for Register VPE_INTC_INTR0_ENA_SET0....2589
Table 10-360. VPE_INTC_INTR0_ENA_SET1....2589
Table 10-361. Register Call Summary for Register VPE_INTC_INTR0_ENA_SET1....2591
Table 10-362. VPE_INTC_INTR0_ENA_CLR0....2591
Table 10-363. Register Call Summary for Register VPE_INTC_INTR0_ENA_CLR0....2592
Table 10-364. VPE_INTC_INTR0_ENA_CLR1....2592
Table 10-365. Register Call Summary for Register VPE_INTC_INTR0_ENA_CLR1....2594
Table 10-366. VPE_INTC_EOI....2594
Table 10-367. Register Call Summary for Register VPE_INTC_EOI....2594
Table 10-368. VPE_CLKC_CLKEN....2594
Table 10-369. Register Call Summary for Register VPE_CLKC_CLKEN....2595
Table 10-370. VPE_CLKC_RST....2595
Table 10-371. Register Call Summary for Register VPE_CLKC_RST....2595
Table 10-372. VPE_CLKC_DPS....2595
Table 10-373. Register Call Summary for Register VPE_CLKC_DPS....2597
Table 10-374. VPE_RANGE_MAP....2597
Table 10-375. Register Call Summary for Register VPE_RANGE_MAP....2597

Table 11-1. Display Subsystem DPI1 Interface Signals Mapping....2602
Table 11-2. Display Subsystem DPI2 Interface Signals Mapping....2602
Table 11-3. Display Subsystem DPI3 Interface Signals Mapping....2602
Table 11-4. Display Subsystem TV Parallel Interface Signals Mapping....2603
Table 11-5. Display Subsystem Hardware Requests....2604
Table 11-6. Display Subsystem Clocks....2605
Table 11-7. Display Subsystem Modules Clock Sources....2606
Table 11-8. Display Subsystem Resets....2607
Table 11-9. Display Subsystem Power Domains....2608
Table 11-10. VIDEO PLL Operation Modes When Not Locked....2614
Table 11-11. DPLL_VIDEO Recommended Programming Values....2618
Table 11-12. DPLL_HDMI Operation Modes When Not Locked....2621
Table 11-13. DPLL_HDMI Register Call Summary for HDMI PLL Programming Sequence....2624
Table 11-14. DPLL_HDMI Recommended Programming Values....2625
Table 11-15. DSS Initialization Sequence....2627
Table 11-16. Display Subsystem Instance Summary....2628
Table 11-17. DSS Registers Mapping Summary....2628
Table 11-18. DSS_REVISION....2628
Table 11-19. DSS_SYSSTATUS....2628
Table 11-20. DSS_CTRL....2629
Table 11-21. DSS_STATUS....2630
Table 11-22. OCP2SCP2 Registers Mapping Summary....2631
Table 11-23. OCP2SCP_REVISION....2632
Table 11-24. OCP2SCP_SYSCONFIG....2632
Table 11-25. OCP2SCP_SYSSTATUS....2632
Table 11-26. OCP2SCP_TIMING....2633
Table 11-27. DPLL_VIDEO1 Registers Mapping Summary....2633
Table 11-28. DPLL_VIDEO2 Registers Mapping Summary....2633
Table 11-29. PLL_CONTROL....2634
Table 11-30. PLL_STATUS....2635
Table 11-31. PLL_GO....2636
Table 11-32. PLL_CONFIGURATION1....2637
Table 11-33. PLL_CONFIGURATION2....2637
Table 11-34. PLL_CONFIGURATION3....2639
Table 11-35. PLL_SSC_CONFIGURATION1....2640
Table 11-36. PLL_SSC_CONFIGURATION2....2640
Table 11-37. PLL_CONFIGURATION4....2641
Table 11-38. DPLL_HDMI Registers Mapping Summary....2641
Table 11-39. PLLCTRL_HDMI_CONTROL....2641
Table 11-40. PLLCTRL_HDMI_STATUS....2642
Table 11-41. PLLCTRL_HDMI_GO....2643
Table 11-42. PLLCTRL_HDMI_CONFIGURATION1....2643
Table 11-43. PLLCTRL_HDMI_CONFIGURATION2....2644
Table 11-44. PLLCTRL_HDMI_CONFIGURATION3....2645
Table 11-45. PLLCTRL_HDMI_SSC_CONFIGURATION1....2646
Table 11-46. PLLCTRL_HDMI_SSC_CONFIGURATION2....2646
Table 11-47. PLLCTRL_HDMI_CONFIGURATION4....2647
Table 11-48. HDMI_WP Registers Mapping Summary....2647
Table 11-49. HDMI_WP_PWR_CTRL....2648
Table 11-50. HDMI_WP_CLK....2648
Table 11-51. DSI1_A Registers Mapping Summary....2649
Table 11-52. DSI1_C Registers Mapping Summary....2649
Table 11-53. DSI_CLK_CTRL....2649
Table 11-54. DISPC Parallel Interface Signals....2656
Table 11-55. DSS Output Data Signals to RGB Color Components Mapping....2659
Table 11-56. DISPC Programmable Fields in Bypass Mode....2660
Table 11-57. DISPC Integration Attributes....2665
Table 11-58. DISPC Clocks and Resets....2666
Table 11-59. DISPC Hardware Requests....2666
Table 11-60. DISPC Interrupts....2671
Table 11-61. DISPC DMA Buffer Size....2673

Table 11-62. DISPC Register Settings for Accessing Image in Internal Memory....2675
Table 11-63. DISPC Register Settings for Rotation Using TILER....2683
Table 11-64. DISPC Rotation Mode Definition....2684
Table 11-65. DISPC Rotation Orientation Definition....2684
Table 11-66. DISPC Memory Formats Supported....2684
Table 11-67. DISPC Replication Enabled: RGB Pixel Formats Remapping Into ARGB40-10.10.10.10....2688
Table 11-68. DISPC Replication Disabled: RGB Pixel Formats Remapping Into ARGB40-10.10.10.10....2688
Table 11-69. DISPC Replication Enabled: RGB Pixel Formats Remapping Into ARGB32-8888....2691
Table 11-70. DISPC Replication Disabled: RGB Pixel Formats Remapping Into ARGB32-8888....2691
Table 11-71. DISPC Color Space Conversion YUV to RGB Bit Field Setting....2692
Table 11-72. DISPC Line Buffer Width for Scaler Unit....2696
Table 11-73. DISPC Register Bit Field Associated to Coefficient for ARGB and Y Configuration in VIDp Scaler....2698
Table 11-74. DISPC Register Bit Field Associated to Coefficient for Cb and Cr Configuration in VIDp Scaler....2699
Table 11-75. DISPC Vertical and Horizontal Accumulator Phase....2700
Table 11-76. DISPC Pixel Clock Frequency Limitations (Any Pixel Format) – Active Matrix Display....2702
Table 11-77. DISPC CSC RGB to YUV Bit Field Setting....2703
Table 11-78. DISPC Register Bit Field Associated With Coefficient for ARGB and Y Configuration in WB Scaler....2706
Table 11-79. DISPC Register Bit Field Associated With Coefficient for Cb and Cr Configuration in WB Scaler....2707
Table 11-80. DISPC Vertical/Horizontal Accumulator Phase....2708
Table 11-81. DISPC Truncation Logic....2709
Table 11-82. DISPC Pipeline Connection to LCD, TV, or WB Output....2711
Table 11-83. DISPC Z-Order Register Settings and Default Configuration....2711
Table 11-84. DISPC Alpha Blending – ARGB....2715
Table 11-85. DISPC CPR or RGB to YUV Coefficients With Associated Bit Fields....2720
Table 11-86. Bit function....2723
Table 11-87. Status of protection bits as F, V and H vary....2724
Table 11-88. DISPC LCD Data Output (Spatial/Temporal Dithering Disabled)....2725
Table 11-89. DISPC Programming Rules....2730
Table 11-90. DISPC PPL and LLP Value for HD Standard....2733
Table 11-91. DISPC Shadow Registers....2739
Table 11-92. DISPC Global Initialization of Surrounding Modules....2745
Table 11-93. DISPC Configuration....2745
Table 11-94. DISPC Configure the GFX DMA Channel....2746
Table 11-95. DISPC Configure the Video DMA Channel....2746
Table 11-96. DISPC Configure the WB DMA Channel....2747
Table 11-97. DISPC Configure the GFX Pipeline....2748
Table 11-98. DISPC Configure the GFX Window....2748
Table 11-99. DISPC Configure the GFX Pipeline Processing....2748
Table 11-100. DISPC Configure the GFX Pipeline Layer Output....2749
Table 11-101. DISPC Configure the Video Pipeline....2749
Table 11-102. DISPC Configure the Video Window....2750
Table 11-103. DISPC Configure the Video Pipeline Processing....2750
Table 11-104. DISPC Configure the VC-1 Range Mapping....2751
Table 11-105. DISPC Configure the Video Color Space Conversion....2751
Table 11-106. DISPC Configure the Video Scaler Unit for RGB Pixel Formats or Y Component....2752
Table 11-107. DISPC Configure the Video Scaler Unit for Cb and Cr Components....2752
Table 11-108. DISPC Configure the Video Pipeline Layer Output....2753
Table 11-109. DISPC Configure the WB Pipeline....2753
Table 11-110. DISPC Configure the Capture Window....2754
Table 11-111. DISPC Configure the WB Scaler Unit for RGB Pixel Formats or Y Component....2754
Table 11-112. DISPC Configure the WB Scaler Unit for Cb and Cr Components....2754
Table 11-113. DISPC Configure the WB Color Space Conversion Unit....2755
Table 11-114. DISPC Configure the LCD Output....2755
Table 11-115. DISPC Configure the LCD Overlay Manager....2755
Table 11-116. DISPC Configure the Gamma Table for LCD1....2756
Table 11-117. DISPC Configure the Gamma Table for LCD2....2756
Table 11-118. DISPC Configure the Gamma Table for LCD3....2756
Table 11-119. DISPC Configure the Color Phase Rotation....2756
Table 11-120. DISPC Configure the LCD Panel Timings and Parameters....2757
Table 11-121. DISPC Configure BT.656 or BT.1120 Mode....2758
Table 11-122. DISPC Configure BT.656 or BT.1120 Blanking Values....2758

Table 11-123. DISPC Configure the TV Output....2759
Table 11-124. DISPC Configure the TV Overlay Manager....2759
Table 11-125. DISPC Configure the Gamma Table for TV Output....2759
Table 11-126. DISPC Configure the TV Panel Timings and Parameters....2759
Table 11-127. DISPC Instance Summary....2760
Table 11-128. DISPC_VIDp_BA_j Logical Register Mapping....2760
Table 11-129. DISPC_VIDp_BA_UV_j Logical Register Mapping....2760
Table 11-130. DISPC_VIDp_POSITION Logical Register Mapping....2760
Table 11-131. DISPC_VIDp_SIZE Logical Register Mapping....2760
Table 11-132. DISPC_VIDp_ATTRIBUTES Logical Register Mapping....2760
Table 11-133. DISPC_VIDp_ATTRIBUTES2 Logical Register Mapping....2760
Table 11-134. DISPC_VIDp_BUF_THRESHOLD Logical Register Mapping....2760
Table 11-135. DISPC_VIDp_ROW_INC Logical Register Mapping....2761
Table 11-136. DISPC_VIDp_PIXEL_INC Logical Register Mapping....2761
Table 11-137. DISPC_VIDp_FIR Logical Register Mapping....2761
Table 11-138. DISPC_VIDp_PICTURE_SIZE Logical Register Mapping....2761
Table 11-139. DISPC_VIDp_ACCU_j Logical Register Mapping....2761
Table 11-140. DISPC_VIDp_FIR_COEF_H_i Logical Register Mapping....2761
Table 11-141. DISPC_VIDp_FIR_COEF_HV_i Logical Register Mapping....2761
Table 11-142. DISPC_VIDp_FIR_COEF_V_i Logical Register Mapping....2761
Table 11-143. DISPC_VIDp_FIR_COEF_H2_i Logical Register Mapping....2762
Table 11-144. DISPC_VIDp_FIR_COEF_HV2_i Logical Register Mapping....2762
Table 11-145. DISPC_VIDp_FIR_COEF_V2_i Logical Register Mapping....2762
Table 11-146. DISPC_VIDp_CONV_COEF0 Logical Register Mapping....2762
Table 11-147. DISPC_VIDp_CONV_COEF1 Logical Register Mapping....2762
Table 11-148. DISPC_VIDp_CONV_COEF2 Logical Register Mapping....2762
Table 11-149. DISPC_VIDp_CONV_COEF3 Logical Register Mapping....2763
Table 11-150. DISPC_VIDp_CONV_COEF4 Logical Register Mapping....2763
Table 11-151. DISPC_CONTROLo Logical Register Mapping....2763
Table 11-152. DISPC_CONFIGo Logical Register Mapping....2763
Table 11-153. DISPC_DEFAULT_COLORo Logical Register Mapping....2763
Table 11-154. DISPC_TRANS_COLORo Logical Register Mapping....2763
Table 11-155. DISPC_GAMMA_TABLEo Logical Register Mapping....2764
Table 11-156. DISPC_TIMING_Ho Logical Register Mapping....2764
Table 11-157. DISPC_TIMING_Vo Logical Register Mapping....2764
Table 11-158. DISPC_POL_FREQo Logical Register Mapping....2764
Table 11-159. DISPC_DIVISORo Logical Register Mapping....2764
Table 11-160. DISPC_SIZE_LCDo Logical Register Mapping....2764
Table 11-161. DISPC_SIZE Logical Register Mapping....2764
Table 11-162. DISPC_DATAo_CYCLE1 Logical Register Mapping....2764
Table 11-163. DISPC_DATAo_CYCLE2 Logical Register Mapping....2765
Table 11-164. DISPC_DATAo_CYCLE3 Logical Register Mapping....2765
Table 11-165. DISPC_CPRo_COEF_R Logical Register Mapping....2765
Table 11-166. DISPC_CPRo_COEF_G Logical Register Mapping....2765
Table 11-167. DISPC_CPRo_COEF_B Logical Register Mapping....2765
Table 11-168. DISPC Registers Mapping Summary....2765
Table 11-169. DISPC_REVISION....2771
Table 11-170. DISPC_SYSCONFIG....2771
Table 11-171. DISPC_SYSSTATUS....2772
Table 11-172. DISPC_IRQSTATUS....2772
Table 11-173. DISPC_IRQENABLE....2777
Table 11-174. DISPC_CONTROL1....2781
Table 11-175. DISPC_CONFIG1....2784
Table 11-176. DISPC_DEFAULT_COLOR0....2788
Table 11-177. DISPC_DEFAULT_COLOR1....2788
Table 11-178. DISPC_TRANS_COLOR0....2789
Table 11-179. DISPC_TRANS_COLOR1....2789
Table 11-180. DISPC_LINE_STATUS....2789
Table 11-181. DISPC_LINE_NUMBER....2790
Table 11-182. DISPC_TIMING_H1....2790
Table 11-183. DISPC_TIMING_V1....2790

Table 11-184. DISPC_POL_FREQ1..2791
Table 11-185. DISPC_DIVISOR1..2792
Table 11-186. DISPC_GLOBAL_ALPHA..2792
Table 11-187. DISPC_SIZE_TV..2793
Table 11-188. DISPC_SIZE_LCD1..2793
Table 11-189. DISPC_GFX_BA_j..2794
Table 11-190. DISPC_GFX_POSITION..2794
Table 11-191. DISPC_GFX_SIZE..2795
Table 11-192. DISPC_GFX_ATTRIBUTES..2795
Table 11-193. DISPC_GFX_BUF_THRESHOLD..2798
Table 11-194. DISPC_GFX_BUF_SIZE_STATUS..2799
Table 11-195. DISPC_GFX_ROW_INC..2799
Table 11-196. DISPC_GFX_PIXEL_INC..2799
Table 11-197. DISPC_GFX_TABLE_BA..2800
Table 11-198. DISPC_VID1_BA_j..2800
Table 11-199. DISPC_VID1_POSITION..2801
Table 11-200. DISPC_VID1_SIZE..2801
Table 11-201. DISPC_VID1_ATTRIBUTES..2802
Table 11-202. DISPC_VID1_BUF_THRESHOLD..2805
Table 11-203. DISPC_VID1_BUF_SIZE_STATUS..2806
Table 11-204. DISPC_VID1_ROW_INC..2806
Table 11-205. DISPC_VID1_PIXEL_INC..2806
Table 11-206. DISPC_VID1_FIR..2807
Table 11-207. DISPC_VID1_PICTURE_SIZE..2807
Table 11-208. DISPC_VID1_ACCU_j..2808
Table 11-209. DISPC_VID1_FIR_COEF_H_i..2808
Table 11-210. DISPC_VID1_FIR_COEF_HV_i..2809
Table 11-211. DISPC_VID1_CONV_COEF0..2809
Table 11-212. DISPC_VID1_CONV_COEF1..2809
Table 11-213. DISPC_VID1_CONV_COEF2..2810
Table 11-214. DISPC_VID1_CONV_COEF3..2810
Table 11-215. DISPC_VID1_CONV_COEF4..2811
Table 11-216. DISPC_VID2_BA_j..2811
Table 11-217. DISPC_VID2_POSITION..2812
Table 11-218. DISPC_VID2_SIZE..2812
Table 11-219. DISPC_VID2_ATTRIBUTES..2813
Table 11-220. DISPC_VID2_BUF_THRESHOLD..2816
Table 11-221. DISPC_VID2_BUF_SIZE_STATUS..2816
Table 11-222. DISPC_VID2_ROW_INC..2817
Table 11-223. DISPC_VID2_PIXEL_INC..2817
Table 11-224. DISPC_VID2_FIR..2818
Table 11-225. DISPC_VID2_PICTURE_SIZE..2818
Table 11-226. DISPC_VID2_ACCU_j..2819
Table 11-227. DISPC_VID2_FIR_COEF_H_i..2819
Table 11-228. DISPC_VID2_FIR_COEF_HV_i..2820
Table 11-229. DISPC_VID2_CONV_COEF0..2820
Table 11-230. DISPC_VID2_CONV_COEF1..2820
Table 11-231. DISPC_VID2_CONV_COEF2..2821
Table 11-232. DISPC_VID2_CONV_COEF3..2821
Table 11-233. DISPC_VID2_CONV_COEF4..2822
Table 11-234. DISPC_DATA1_CYCLE1..2822
Table 11-235. DISPC_DATA1_CYCLE2..2822
Table 11-236. DISPC_DATA1_CYCLE3..2823
Table 11-237. DISPC_VID1_FIR_COEF_V_i..2823
Table 11-238. DISPC_VID2_FIR_COEF_V_i..2824
Table 11-239. DISPC_CPR1_COEF_R..2824
Table 11-240. DISPC_CPR1_COEF_G..2825
Table 11-241. DISPC_CPR1_COEF_B..2825
Table 11-242. DISPC_GFX_PRELOAD..2825
Table 11-243. DISPC_VID1_PRELOAD..2826
Table 11-244. DISPC_VID2_PRELOAD..2826

Table 11-245. DISPC_CONTROL2....2826
Table 11-246. DISPC_GFX_POSITION2....2829
Table 11-247. DISPC_VID1_POSITION2....2830
Table 11-248. DISPC_VID2_POSITION2....2830
Table 11-249. DISPC_VID3_POSITION2....2831
Table 11-250. DISPC_VID3_ACCU_j....2831
Table 11-251. DISPC_VID3_BA_j....2832
Table 11-252. DISPC_VID3_FIR_COEF_H_i....2832
Table 11-253. DISPC_VID3_FIR_COEF_HV_i....2833
Table 11-254. DISPC_VID3_FIR_COEF_V_i....2833
Table 11-255. DISPC_VID3_ATTRIBUTES....2834
Table 11-256. DISPC_VID3_CONV_COEF0....2837
Table 11-257. DISPC_VID3_CONV_COEF1....2837
Table 11-258. DISPC_VID3_CONV_COEF2....2838
Table 11-259. DISPC_VID3_CONV_COEF3....2838
Table 11-260. DISPC_VID3_CONV_COEF4....2839
Table 11-261. DISPC_VID3_BUF_SIZE_STATUS....2839
Table 11-262. DISPC_VID3_BUF_THRESHOLD....2839
Table 11-263. DISPC_VID3_FIR....2840
Table 11-264. DISPC_VID3_PICTURE_SIZE....2840
Table 11-265. DISPC_VID3_PIXEL_INC....2841
Table 11-266. DISPC_VID3_POSITION....2841
Table 11-267. DISPC_VID3_PRELOAD....2842
Table 11-268. DISPC_VID3_ROW_INC....2842
Table 11-269. DISPC_VID3_SIZE....2842
Table 11-270. DISPC_DEFAULT_COLOR2....2843
Table 11-271. DISPC_TRANS_COLOR2....2843
Table 11-272. DISPC_CPR2_COEF_B....2843
Table 11-273. DISPC_CPR2_COEF_G....2844
Table 11-274. DISPC_CPR2_COEF_R....2844
Table 11-275. DISPC_DATA2_CYCLE1....2845
Table 11-276. DISPC_DATA2_CYCLE2....2845
Table 11-277. DISPC_DATA2_CYCLE3....2846
Table 11-278. DISPC_SIZE_LCD2....2846
Table 11-279. DISPC_TIMING_H2....2847
Table 11-280. DISPC_TIMING_V2....2847
Table 11-281. DISPC_POL_FREQ2....2847
Table 11-282. DISPC_DIVISOR2....2848
Table 11-283. DISPC_WB_ACCU_j....2849
Table 11-284. DISPC_WB_BA_j....2849
Table 11-285. DISPC_WB_FIR_COEF_H_i....2850
Table 11-286. DISPC_WB_FIR_COEF_HV_i....2851
Table 11-287. DISPC_WB_FIR_COEF_V_i....2851
Table 11-288. DISPC_WB_ATTRIBUTES....2852
Table 11-289. DISPC_WB_CONV_COEF0....2855
Table 11-290. DISPC_WB_CONV_COEF1....2856
Table 11-291. DISPC_WB_CONV_COEF2....2856
Table 11-292. DISPC_WB_CONV_COEF3....2857
Table 11-293. DISPC_WB_CONV_COEF4....2857
Table 11-294. DISPC_WB_BUF_SIZE_STATUS....2858
Table 11-295. DISPC_WB_BUF_THRESHOLD....2858
Table 11-296. DISPC_WB_FIR....2858
Table 11-297. DISPC_WB_PICTURE_SIZE....2859
Table 11-298. DISPC_WB_PIXEL_INC....2859
Table 11-299. DISPC_WB_ROW_INC....2860
Table 11-300. DISPC_WB_SIZE....2860
Table 11-301. DISPC_VID1_BA_UV_j....2861
Table 11-302. DISPC_VID2_BA_UV_j....2861
Table 11-303. DISPC_VID3_BA_UV_j....2862
Table 11-304. DISPC_WB_BA_UV_j....2862
Table 11-305. DISPC_CONFIG2....2863

Table 11-306. DISPC_VID1_ATTRIBUTES2....... 2865
Table 11-307. DISPC_VID2_ATTRIBUTES2....... 2866
Table 11-308. DISPC_VID3_ATTRIBUTES2....... 2867
Table 11-309. DISPC_GAMMA_TABLE0....... 2868
Table 11-310. DISPC_GAMMA_TABLE1....... 2868
Table 11-311. DISPC_GAMMA_TABLE2....... 2868
Table 11-312. DISPC_VID1_FIR2....... 2869
Table 11-313. DISPC_VID1_ACCU2_j....... 2869
Table 11-314. DISPC_VID1_FIR_COEF_H2_i....... 2870
Table 11-315. DISPC_VID1_FIR_COEF_HV2_i....... 2870
Table 11-316. DISPC_VID1_FIR_COEF_V2_i....... 2871
Table 11-317. DISPC_VID2_FIR2....... 2871
Table 11-318. DISPC_VID2_ACCU2_j....... 2872
Table 11-319. DISPC_VID2_FIR_COEF_H2_i....... 2872
Table 11-320. DISPC_VID2_FIR_COEF_HV2_i....... 2873
Table 11-321. DISPC_VID2_FIR_COEF_V2_i....... 2873
Table 11-322. DISPC_VID3_FIR2....... 2874
Table 11-323. DISPC_VID3_ACCU2_j....... 2874
Table 11-324. DISPC_VID3_FIR_COEF_H2_i....... 2875
Table 11-325. DISPC_VID3_FIR_COEF_HV2_i....... 2875
Table 11-326. DISPC_VID3_FIR_COEF_V2_i....... 2876
Table 11-327. DISPC_WB_FIR2....... 2876
Table 11-328. DISPC_WB_ACCU2_j....... 2877
Table 11-329. DISPC_WB_FIR_COEF_H2_i....... 2877
Table 11-330. DISPC_WB_FIR_COEF_HV2_i....... 2878
Table 11-331. DISPC_WB_FIR_COEF_V2_i....... 2879
Table 11-332. DISPC_GLOBAL_BUFFER....... 2879
Table 11-333. DISPC_DIVISOR....... 2880
Table 11-334. DISPC_WB_ATTRIBUTES2....... 2881
Table 11-335. DISPC_DEFAULT_COLOR3....... 2881
Table 11-336. DISPC_TRANS_COLOR3....... 2882
Table 11-337. DISPC_CPR3_COEF_B....... 2882
Table 11-338. DISPC_CPR3_COEF_G....... 2882
Table 11-339. DISPC_CPR3_COEF_R....... 2883
Table 11-340. DISPC_DATA3_CYCLE1....... 2883
Table 11-341. DISPC_DATA3_CYCLE2....... 2884
Table 11-342. DISPC_DATA3_CYCLE3....... 2884
Table 11-343. DISPC_SIZE_LCD3....... 2885
Table 11-344. DISPC_DIVISOR3....... 2886
Table 11-345. DISPC_POL_FREQ3....... 2886
Table 11-346. DISPC_TIMING_H3....... 2887
Table 11-347. DISPC_TIMING_V3....... 2888
Table 11-348. DISPC_CONTROL3....... 2888
Table 11-349. DISPC_CONFIG3....... 2891
Table 11-350. DISPC_GAMMA_TABLE3....... 2893
Table 11-351. DISPC_BA0_FLIPIMMEDIATE_EN....... 2893
Table 11-352. DISABLE_MSTANDBY_ENHANCEMENT....... 2893
Table 11-353. DISPC_GLOBAL_MFLAG_ATTRIBUTE....... 2894
Table 11-354. DISPC_GFX_MFLAG_THRESHOLD....... 2895
Table 11-355. DISPC_VID1_MFLAG_THRESHOLD....... 2895
Table 11-356. DISPC_VID2_MFLAG_THRESHOLD....... 2895
Table 11-357. DISPC_VID3_MFLAG_THRESHOLD....... 2896
Table 11-358. DISPC_WB_MFLAG_THRESHOLD....... 2896
Table 11-359. HDMI Video Timings (CEA-861-D)....... 2898
Table 11-360. HDMI Video timings (VESA DMT)....... 2899
Table 12-1. GPU Integration Attributes....... 2905
Table 12-2. GPU Clocks and Resets....... 2906
Table 12-3. GPU Hardware Requests....... 2906
Table 12-4. GPU Instance Summary....... 2910
Table 12-5. GPU_WRAPPER Registers Mapping Summary....... 2910
Table 12-6. REVISION....... 2910

Table 12-7. Register Call Summary for Register REVISION....2911
Table 12-8. HWINFO....2911
Table 12-9. Register Call Summary for Register HWINFO....2911
Table 12-10. SYSCONFIG....2911
Table 12-11. Register Call Summary for Register SYSCONFIG....2912
Table 12-12. IRQSTATUS_RAW_0....2912
Table 12-13. Register Call Summary for Register IRQSTATUS_RAW_0....2912
Table 12-14. IRQSTATUS_RAW_1....2912
Table 12-15. Register Call Summary for Register IRQSTATUS_RAW_1....2913
Table 12-16. IRQSTATUS_RAW_2....2913
Table 12-17. Register Call Summary for Register IRQSTATUS_RAW_2....2913
Table 12-18. IRQSTATUS_0....2913
Table 12-19. Register Call Summary for Register IRQSTATUS_0....2914
Table 12-20. IRQSTATUS_1....2914
Table 12-21. Register Call Summary for Register IRQSTATUS_1....2915
Table 12-22. IRQSTATUS_2....2915
Table 12-23. Register Call Summary for Register IRQSTATUS_2....2915
Table 12-24. IRQENABLE_SET_0....2915
Table 12-25. Register Call Summary for Register IRQENABLE_SET_0....2916
Table 12-26. IRQENABLE_SET_1....2916
Table 12-27. Register Call Summary for Register IRQENABLE_SET_1....2916
Table 12-28. IRQENABLE_SET_2....2916
Table 12-29. Register Call Summary for Register IRQENABLE_SET_2....2917
Table 12-30. IRQENABLE_CLR_0....2917
Table 12-31. Register Call Summary for Register IRQENABLE_CLR_0....2917
Table 12-32. IRQENABLE_CLR_1....2917
Table 12-33. Register Call Summary for Register IRQENABLE_CLR_1....2918
Table 12-34. IRQENABLE_CLR_2....2918
Table 12-35. Register Call Summary for Register IRQENABLE_CLR_2....2919
Table 12-36. PAGE_CONFIG....2919
Table 12-37. Register Call Summary for Register PAGE_CONFIG....2919
Table 12-38. INTERRUPT_EVENT....2919
Table 12-39. Register Call Summary for Register INTERRUPT_EVENT....2921
Table 12-40. DEBUG_CONFIG....2921
Table 12-41. Register Call Summary for Register DEBUG_CONFIG....2922
Table 12-42. DEBUG_STATUS_0....2922
Table 12-43. Register Call Summary for Register DEBUG_STATUS_0....2924
Table 12-44. DEBUG_STATUS_1....2924
Table 12-45. Register Call Summary for Register DEBUG_STATUS_1....2925
Table 13-1. BB2D Integration Attributes....2929
Table 13-2. BB2D Clocks and Resets....2929
Table 13-3. BB2D Hardware Requests....2930
Table 13-4. BB2D Instance Summary....2933
Table 13-5. BB2D Registers Mapping Summary....2933
Table 13-6. AQHICLOCKCONTROL....2935
Table 13-7. AQHIIDLE....2935
Table 13-8. AQAXICONFIG....2936
Table 13-9. AQAXISTATUS....2936
Table 13-10. AQINTRACKNOWLEDGE....2937
Table 13-11. AQINTRENBL....2937
Table 13-12. AQIDENT....2937
Table 13-13. GCFEATURES....2938
Table 13-14. GCCHIPID....2939
Table 13-15. GCCHIPREV....2939
Table 13-16. GCCHIPDATE....2939
Table 13-17. GCCHIPTIME....2940
Table 13-18. GCCHIPCUSTOMER....2940
Table 13-19. GCMINORFEATURES0....2940
Table 13-20. GCRESETMEMCOUNTERS....2941
Table 13-21. GCTOTALREADS....2942
Table 13-22. GCTOTALWRITES....2942

Table 13-23. GCCHIPSPECS....2942
Table 13-24. GCTOTALWRITEBURSTS....2943
Table 13-25. GCTOTALWRITEREQS....2943
Table 13-26. GCTOTALWRITELASTS....2943
Table 13-27. GCTOTALREADBURSTS....2943
Table 13-28. GCTOTALREADREQS....2944
Table 13-29. GCTOTALREADLASTS....2944
Table 13-30. GCGPOUT0....2944
Table 13-31. GCAXICONTROL....2944
Table 13-32. GCMINORFEATURES1....2945
Table 13-33. GCTOTALCYCLES....2946
Table 13-34. GCTOTALIDLECYCLES....2946
Table 13-35. GCCHIPSPECS2....2947
Table 13-36. GCMINORFEATURES2....2947
Table 13-37. GCMODULEPOWERCONTROLS....2948
Table 13-38. GCMODULEPOWERMODULECONTROL....2949
Table 13-39. GCMODULEPOWERMODULESTATUS....2950
Table 13-40. GCREGMMUSTATUS....2950
Table 13-41. GCREGMMUCONTROL....2951
Table 13-42. GCREGMMUEXCEPTION0....2951
Table 13-43. GCREGMMUEXCEPTION1....2951
Table 13-44. GCREGMMUEXCEPTION2....2952
Table 13-45. GCREGMMUEXCEPTION3....2952
Table 13-46. AQMEMORYDEBUG....2952
Table 13-47. AQREGISTERTIMINGCONTROL....2953
Table 13-48. GCDISPLAYPRIORITY....2953
Table 13-49. GCDBGCYCLECOUNTER....2954
Table 13-50. GCOUTSTANDINGREADS0....2954
Table 13-51. GCOUTSTANDINGREADS1....2954
Table 13-52. GCOUTSTANDINGWRITES....2955
Table 13-53. GCDEBUGSIGNALSRA....2955
Table 13-54. GCDEBUGSIGNALSTX....2955
Table 13-55. GCDEBUGSIGNALSFE....2956
Table 13-56. GCDEBUGSIGNALSPE....2956
Table 13-57. GCDEBUGSIGNALSDE....2956
Table 13-58. GCDEBUGSIGNALSSH....2957
Table 13-59. GCDEBUGSIGNALSPA....2957
Table 13-60. GCDEBUGSIGNALSSE....2958
Table 13-61. GCDEBUGSIGNALSMC....2958
Table 13-62. GCDEBUGSIGNALSHI....2959
Table 13-63. GCDEBUGCONTROL0....2959
Table 13-64. GCDEBUGCONTROL1....2959
Table 13-65. GCDEBUGCONTROL2....2960
Table 13-66. GCDEBUGCONTROL3....2960
Table 13-67. GCBUSCONTROL....2961
Table 13-68. GCREGENDIANNESS0....2961
Table 13-69. GCREGENDIANNESS1....2962
Table 13-70. GCREGENDIANNESS2....2962
Table 13-71. GCREGDRAWPRIMITIVESTARTTIMESTAMP....2962
Table 13-72. GCREGDRAWPRIMITIVEENDTIMESTAMP....2962
Table 13-73. GCREGCONTROL0....2963
Table 13-74. AQCMDBUFFERADDR....2963
Table 13-75. AQCMDBUFFERCTRL....2964
Table 13-76. AQFESTATUS....2964
Table 13-77. AQFEDEBUGCURCMDADR....2964
Table 14-1. MCmd Qualifier Description....2967
Table 14-2. MReqInfo Qualifier Description....2968
Table 14-3. SResp Qualifier Description....2968
Table 14-4. L3_MAIN Integration Attributes....2970
Table 14-5. L3_MAIN Clocks and Resets....2971
Table 14-6. L3_MAIN Hardware Requests....2971

Table 14-7. Master NIUs....2972
Table 14-8. Slave NIUs....2973
Table 14-9. L3_MAIN Clock Domains and Elements....2974
Table 14-10. ConnID Values....2978
Table 14-11. L3 Time-out Flag Mapping....2983
Table 14-12. L3_MAIN Firewall Exported Reset Values....2985
Table 14-13. L3_MAIN Firewall Exported Values Mapping....2986
Table 14-14. L3_MAIN Firewall Exported Values Mapping....2986
Table 14-15. MReqInfo Values....2987
Table 14-16. L3_MAIN ReqInfo Mapping....2988
Table 14-17. Slave NIU Firewall and Region Configuration....2990
Table 14-18. L3_MAIN Firewall Read/Write Permission-Setting Register....2993
Table 14-19. L3_MAIN Firewall Permission-Setting Register....2994
Table 14-20. L3 Firewall Error-Logging Registers....2995
Table 14-21. L3_MAIN Firewalls Default Configurations....2997
Table 14-22. Control Module Register – Factorization....2997
Table 14-23. L3_MAIN Connectivity and Holes Error Routing....2999
Table 14-24. Interconnect Flag Mapping....3001
Table 14-25. Global Initialization of Surrounding Modules....3003
Table 14-26. Subprocess Call Summary for Main Sequence – Error Analysis Mode....3004
Table 14-27. Custom Error Identification....3004
Table 14-28. L3_MAIN Protection Violation Error Identification....3004
Table 14-29. L3_MAIN Standard Error Identification....3005
Table 14-30. FLAGMUX Configuration....3005
Table 14-31. FLAGMUX Time-out Configuration (for CLK1_1 and CLK1_2)....3005
Table 14-32. FLAGMUX Time-out Configuration (for CLK2_1)....3005
Table 14-33. L3_MAIN Firewall Instance Summary....3006
Table 14-34. L3_MAIN Firewall Registers Summary....3007
Table 14-35. L3_MAIN Firewall Registers Mapping Summary....3007
Table 14-36. L3_MAIN Firewall Registers Mapping Summary....3008
Table 14-37. L3_MAIN Firewall Registers Mapping Summary....3008
Table 14-38. L3_MAIN Firewall Registers Mapping Summary....3008
Table 14-39. L3_MAIN Firewall Registers Mapping Summary....3009
Table 14-40. L3_MAIN Firewall Registers Mapping Summary....3009
Table 14-41. L3_MAIN Firewall Registers Mapping Summary....3010
Table 14-42. L3_MAIN Firewall Registers Mapping Summary....3010
Table 14-43. ERROR_LOG_k....3012
Table 14-44. Register Call Summary for Register ERROR_LOG_k....3012
Table 14-45. LOGICAL_ADDR_ERRLOG_k....3012
Table 14-46. Register Call Summary for Register LOGICAL_ADDR_ERRLOG_k....3013
Table 14-47. REGUPDATE_CONTROL....3013
Table 14-48. Register Call Summary for Register REGUPDATE_CONTROL....3013
Table 14-49. START_REGION_i....3013
Table 14-50. Register Call Summary for Register START_REGION_i....3014
Table 14-51. Size of START_REGION_i[] START_REGION Bit Field....3014
Table 14-52. END_REGION_i....3014
Table 14-53. Register Call Summary for Register END_REGION_i....3015
Table 14-54. Size of END_REGION_i[] END_REGION Bit Field....3015
Table 14-55. MRM_PERMISSION_REGION_HIGH_j....3016
Table 14-56. Register Call Summary for Register MRM_PERMISSION_REGION_HIGH_j....3017
Table 14-57. MRM_PERMISSION_REGION_LOW_j....3017
Table 14-58. Register Call Summary for Register MRM_PERMISSION_REGION_LOW_j....3017
Table 14-59. Reset Value for MRM_PERMISSION_REGION_LOW_j....3018
Table 14-60. HOST Instance Summary....3018
Table 14-61. HOST Registers Summary....3019
Table 14-62. L3_HOST_STDHOSTHDR_COREREG....3020
Table 14-63. Register Call Summary for Register L3_HOST_STDHOSTHDR_COREREG....3020
Table 14-64. L3_HOST_STDHOSTHDR_VERSIONREG....3020
Table 14-65. Register Call Summary for Register L3_HOST_STDHOSTHDR_VERSIONREG....3021
Table 14-66. L3_HOST_STDHOSTHDR_MAINCTLREG....3021
Table 14-67. Register Call Summary for Register L3_HOST_STDHOSTHDR_MAINCTLREG....3021

Table 14-68. L3_HOST_STDERRLOG_SVRTSTDLVL....3021
Table 14-69. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTSTDLVL....3022
Table 14-70. L3_HOST_STDERRLOG_SVRTCUSTOMLVL....3022
Table 14-71. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTCUSTOMLVL....3022
Table 14-72. L3_HOST_STDERRLOG_MAIN....3023
Table 14-73. Register Call Summary for Register L3_HOST_STDERRLOG_MAIN....3023
Table 14-74. L3_HOST_STDERRLOG_HDR....3023
Table 14-75. Register Call Summary for Register L3_HOST_STDERRLOG_HDR....3025
Table 14-76. L3_HOST_STDERRLOG_MSTADDR....3025
Table 14-77. Register Call Summary for Register L3_HOST_STDERRLOG_MSTADDR....3025
Table 14-78. L3_HOST_STDERRLOG_SLVADDR....3025
Table 14-79. Register Call Summary for Register L3_HOST_STDERRLOG_SLVADDR....3025
Table 14-80. L3_HOST_STDERRLOG_INFO....3025
Table 14-81. Register Call Summary for Register L3_HOST_STDERRLOG_INFO....3026
Table 14-82. L3_HOST_STDERRLOG_SLVOFSLSB....3026
Table 14-83. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSLSB....3026
Table 14-84. L3_HOST_STDERRLOG_SLVOFSMSB....3026
Table 14-85. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSMSB....3027
Table 14-86. L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR....3027
Table 14-87. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR....3027
Table 14-88. L3_HOST_STDERRLOG_CUSTOMINFO_INFO....3027
Table 14-89. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_INFO....3027
Table 14-90. L3_HOST_STDERRLOG_CUSTOMINFO_WR....3027
Table 14-91. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_WR....3028
Table 14-92. L3_HOST_STDERRLOG_CUSTOMINFO_ADDR....3028
Table 14-93. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_ADDR....3028
Table 14-94. L3_HOST_STDERRLOG_CUSTOMINFO_DECERR....3028
Table 14-95. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_DECERR....3029
Table 14-96. L3_MAIN TARG Instance Summary....3029
Table 14-97. L3_MAIN TARG Register Summary....3031
Table 14-98. L3_MAIN TARG Register Summary....3031
Table 14-99. L3_MAIN TARG Register Summary....3032
Table 14-100. L3_MAIN TARG Register Summary....3033
Table 14-101. L3_MAIN TARG Register Summary....3033
Table 14-102. L3_MAIN TARG Register Summary....3034
Table 14-103. L3_MAIN TARG Register Summary....3035
Table 14-104. L3_MAIN TARG Register Summary....3035
Table 14-105. L3_MAIN TARG Register Summary....3036
Table 14-106. L3_MAIN TARG Register Summary....3037
Table 14-107. L3_MAIN TARG Register Summary....3038
Table 14-108. L3_TARG_STDHOSTHDR_COREREG....3040
Table 14-109. Register Call Summary for Register L3_TARG_STDHOSTHDR_COREREG....3040
Table 14-110. L3_TARG_STDHOSTHDR_VERSIONREG....3040
Table 14-111. Register Call Summary for Register L3_TARG_STDHOSTHDR_VERSIONREG....3041
Table 14-112. L3_TARG_STDHOSTHDR_MAINCTLREG....3041
Table 14-113. Register Call Summary for Register L3_TARG_STDHOSTHDR_MAINCTLREG....3041
Table 14-114. L3_TARG_STDHOSTHDR_NTTPADDR_0....3041
Table 14-115. Register Call Summary for Register L3_TARG_STDHOSTHDR_NTTPADDR_0....3042
Table 14-116. L3_TARG_STDERRLOG_SVRTSTDLVL....3042
Table 14-117. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTSTDLVL....3042
Table 14-118. L3_TARG_STDERRLOG_SVRTCUSTOMLVL....3042
Table 14-119. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTCUSTOMLVL....3043
Table 14-120. L3_TARG_STDERRLOG_MAIN....3043
Table 14-121. Register Call Summary for Register L3_TARG_STDERRLOG_MAIN....3044
Table 14-122. L3_TARG_STDERRLOG_HDR....3044
Table 14-123. Register Call Summary for Register L3_TARG_STDERRLOG_HDR....3045
Table 14-124. L3_TARG_STDERRLOG_MSTADDR....3045
Table 14-125. Register Call Summary for Register L3_TARG_STDERRLOG_MSTADDR....3046
Table 14-126. L3_TARG_STDERRLOG_SLVADDR....3046
Table 14-127. Register Call Summary for Register L3_TARG_STDERRLOG_SLVADDR....3046
Table 14-128. L3_TARG_STDERRLOG_INFO....3046

Table 14-129. Register Call Summary for Register L3_TARG_STDERRLOG_INFO....3046
Table 14-130. L3_TARG_STDERRLOG_SLVOFSLSB....3046
Table 14-131. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSLSB....3047
Table 14-132. L3_TARG_STDERRLOG_SLVOFSMSB....3047
Table 14-133. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSMSB....3047
Table 14-134. L3_TARG_STDERRLOG_CUSTOMINFO_INFO....3047
Table 14-135. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_INFO....3048
Table 14-136. L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR....3048
Table 14-137. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR....3048
Table 14-138. L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE....3048
Table 14-139. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE....3049
Table 14-140. L3_TARG_ADDRSPACESIZELOG....3049
Table 14-141. Register Call Summary for Register L3_TARG_ADDRSPACESIZELOG....3049
Table 14-142. FLAGMUX Instance Summary....3049
Table 14-143. FLAGMUX Registers Summary....3049
Table 14-144. L3_FLAGMUX_STDHOSTHDR_COREREG....3050
Table 14-145. Register Call Summary for Register L3_FLAGMUX_STDHOSTHDR_COREREG....3050
Table 14-146. L3_FLAGMUX_STDHOSTHDR_VERSIONREG....3050
Table 14-147. Register Call Summary for Register L3_FLAGMUX_STDHOSTHDR_VERSIONREG....3051
Table 14-148. L3_FLAGMUX_MASK0....3051
Table 14-149. Register Call Summary for Register L3_FLAGMUX_MASK0....3051
Table 14-150. L3_FLAGMUX_REGERR0....3051
Table 14-151. Register Call Summary for Register L3_FLAGMUX_REGERR0....3052
Table 14-152. L3_FLAGMUX_MASK1....3052
Table 14-153. Register Call Summary for Register L3_FLAGMUX_MASK1....3052
Table 14-154. L3_FLAGMUX_REGERR1....3052
Table 14-155. Register Call Summary for Register L3_FLAGMUX_REGERR1....3052
Table 14-156. Instance Summary....3053
Table 14-157. FLAGMUX CLK1MERGE Registers Summary....3053
Table 14-158. L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG....3053
Table 14-159. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG....3054
Table 14-160. L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG....3054
Table 14-161. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG....3054
Table 14-162. L3_FLAGMUX_CLK1MERGE_MASK0....3054
Table 14-163. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_MASK0....3054
Table 14-164. L3_FLAGMUX_CLK1MERGE_REGERR0....3055
Table 14-165. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_REGERR0....3055
Table 14-166. L3_FLAGMUX_CLK1MERGE_MASK1....3055
Table 14-167. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_MASK1....3055
Table 14-168. L3_FLAGMUX_CLK1MERGE_REGERR1....3055
Table 14-169. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_REGERR1....3056
Table 14-170. FLAGMUX TIMEOUT Instance Summary....3056
Table 14-171. FLAGMUX Registers Summary....3056
Table 14-172. L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG....3056
Table 14-173. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG....3057
Table 14-174. L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG....3057
Table 14-175. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG....3058
Table 14-176. L3_FLAGMUX_TIMEOUT_MASK0....3058
Table 14-177. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_MASK0....3058
Table 14-178. L3_FLAGMUX_TIMEOUT_REGERR0....3058
Table 14-179. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_REGERR0....3058
Table 14-180. L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG....3058
Table 14-181. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG....3059
Table 14-182. L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG....3059
Table 14-183. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG....3060
Table 14-184. L3_FLAGMUX_TIMEOUT1_MASK0....3060
Table 14-185. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_MASK0....3060
Table 14-186. L3_FLAGMUX_TIMEOUT1_REGERR0....3060
Table 14-187. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_REGERR0....3060
Table 14-188. L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG....3060
Table 14-189. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG....3061

Table 14-190. L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG....... 3061
Table 14-191. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG....... 3062
Table 14-192. L3_FLAGMUX_TIMEOUT2_MASK0....... 3062
Table 14-193. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_MASK0....... 3062
Table 14-194. L3_FLAGMUX_TIMEOUT2_REGERR0....... 3062
Table 14-195. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_REGERR0....... 3062
Table 14-196. BW_REGULATOR Instance Summary....... 3062
Table 14-197. BW_REGULATOR Register Summary....... 3063
Table 14-198. BW_REGULATOR Register Summary....... 3063
Table 14-199. BW_REGULATOR Register Summary....... 3063
Table 14-200. BW_REGULATOR Register Summary....... 3064
Table 14-201. BW_REGULATOR Register Summary....... 3064
Table 14-202. BW_REGULATOR Register Summary....... 3064
Table 14-203. BW_REGULATOR Register Summary....... 3065
Table 14-204. L3_BW_REGULATOR_STDHOSTHDR_COREREG....... 3065
Table 14-205. Register Call Summary for Register L3_BW_REGULATOR_STDHOSTHDR_COREREG....... 3067
Table 14-206. L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG....... 3067
Table 14-207. Register Call Summary for Register L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG....... 3068
Table 14-208. L3_BW_REGULATOR_BANDWIDTH....... 3068
Table 14-209. Register Call Summary for Register L3_BW_REGULATOR_BANDWIDTH....... 3068
Table 14-210. L3_BW_REGULATOR_WATERMARK....... 3068
Table 14-211. Register Call Summary for Register L3_BW_REGULATOR_WATERMARK....... 3069
Table 14-212. L3_BW_REGULATOR_PRESS....... 3069
Table 14-213. Register Call Summary for Register L3_BW_REGULATOR_PRESS....... 3070
Table 14-214. L3_BW_REGULATOR_CLEARHISTORY....... 3070
Table 14-215. Register Call Summary for Register L3_BW_REGULATOR_CLEARHISTORY....... 3071
Table 14-216. BW_LIMITER Instance Summary....... 3071
Table 14-217. BW_LIMITER Register Summary....... 3072
Table 14-218. BW_LIMITER Register Summary....... 3072
Table 14-219. BW_LIMITER Register Summary....... 3073
Table 14-220. BW_LIMITER Register Summary....... 3073
Table 14-221. L3_BW_LIMITER_STDHOSTHDR_COREREG....... 3073
Table 14-222. Register Call Summary for Register L3_BW_LIMITER_STDHOSTHDR_COREREG....... 3074
Table 14-223. L3_BW_LIMITER_STDHOSTHDR_VERSIONREG....... 3074
Table 14-224. Register Call Summary for Register L3_BW_LIMITER_STDHOSTHDR_VERSIONREG....... 3075
Table 14-225. L3_BW_LIMITER_BANDWIDTH_FRACTIONAL....... 3075
Table 14-226. Register Call Summary for Register L3_BW_LIMITER_BANDWIDTH_FRACTIONAL....... 3075
Table 14-227. L3_BW_LIMITER_BANDWIDTH_INTEGER....... 3075
Table 14-228. Register Call Summary for Register L3_BW_LIMITER_BANDWIDTH_INTEGER....... 3076
Table 14-229. L3_BW_LIMITER_WATERMARK_0....... 3076
Table 14-230. Register Call Summary for Register L3_BW_LIMITER_WATERMARK_0....... 3077
Table 14-231. L3_BW_LIMITER_CLEARHISTORY....... 3077
Table 14-232. Register Call Summary for Register L3_BW_LIMITER_CLEARHISTORY....... 3077
Table 14-233. STATCOLL Instance Summary....... 3077
Table 14-234. STATCOLL Register Summary....... 3078
Table 14-235. STATCOLL Register Summary....... 3078
Table 14-236. STATCOLL Register Summary....... 3082
Table 14-237. L3_STCOL_STDHOSTHDR_COREREG....... 3085
Table 14-238. Register Call Summary for Register L3_STCOL_STDHOSTHDR_COREREG....... 3086
Table 14-239. L3_STCOL_STDHOSTHDR_VERSIONREG....... 3086
Table 14-240. Register Call Summary for Register L3_STCOL_STDHOSTHDR_VERSIONREG....... 3087
Table 14-241. L3_STCOL_MASK0....... 3087
Table 14-242. Register Call Summary for Register L3_STCOL_MASK0....... 3087
Table 14-243. L3_STCOL_REGERR0....... 3087
Table 14-244. Register Call Summary for Register L3_STCOL_REGERR0....... 3088
Table 14-245. L3_STCOL_EN....... 3088
Table 14-246. Register Call Summary for Register L3_STCOL_EN....... 3088
Table 14-247. L3_STCOL_SOFTEN....... 3088
Table 14-248. Register Call Summary for Register L3_STCOL_SOFTEN....... 3089
Table 14-249. L3_STCOL_IGNORESUSPEND....... 3089
Table 14-250. Register Call Summary for Register L3_STCOL_IGNORESUSPEND....... 3089

Table 14-251. L3_STCOL_TRIGEN 3089
Table 14-252. Register Call Summary for Register L3_STCOL_TRIGEN 3090
Table 14-253. L3_STCOL_REQEVT 3090
Table 14-254. Register Call Summary for Register L3_STCOL_REQEVT 3091
Table 14-255. L3_STCOL_RSPEVT 3091
Table 14-256. Register Call Summary for Register L3_STCOL_RSPEVT 3092
Table 14-257. L3_STCOL_EVTMUX_SEL0 3092
Table 14-258. Register Call Summary for Register L3_STCOL_EVTMUX_SEL0 3092
Table 14-259. L3_STCOL_EVTMUX_SEL1 3092
Table 14-260. Register Call Summary for Register L3_STCOL_EVTMUX_SEL1 3093
Table 14-261. L3_STCOL_EVTMUX_SEL2 3093
Table 14-262. Register Call Summary for Register L3_STCOL_EVTMUX_SEL2 3093
Table 14-263. L3_STCOL_EVTMUX_SEL3 3093
Table 14-264. Register Call Summary for Register L3_STCOL_EVTMUX_SEL3 3094
Table 14-265. L3_STCOL_EVTMUX_SEL4 3094
Table 14-266. Register Call Summary for Register L3_STCOL_EVTMUX_SEL4 3094
Table 14-267. L3_STCOL_EVTMUX_SEL5 3094
Table 14-268. Register Call Summary for Register L3_STCOL_EVTMUX_SEL5 3095
Table 14-269. L3_STCOL_EVTMUX_SEL6 3095
Table 14-270. Register Call Summary for Register L3_STCOL_EVTMUX_SEL6 3095
Table 14-271. L3_STCOL_EVTMUX_SEL7 3095
Table 14-272. Register Call Summary for Register L3_STCOL_EVTMUX_SEL7 3095
Table 14-273. L3_STCOL_DUMP_IDENTIFIER 3095
Table 14-274. Register Call Summary for Register L3_STCOL_DUMP_IDENTIFIER 3096
Table 14-275. L3_STCOL_DUMP_COLLECTTIME 3096
Table 14-276. Register Call Summary for Register L3_STCOL_DUMP_COLLECTTIME 3096
Table 14-277. L3_STCOL_DUMP_SLVADDR 3096
Table 14-278. Register Call Summary for Register L3_STCOL_DUMP_SLVADDR 3097
Table 14-279. L3_STCOL_DUMP_MSTADDR 3097
Table 14-280. Register Call Summary for Register L3_STCOL_DUMP_MSTADDR 3097
Table 14-281. L3_STCOL_DUMP_SLVOFS 3097
Table 14-282. Register Call Summary for Register L3_STCOL_DUMP_SLVOFS 3098
Table 14-283. L3_STCOL_DUMP_MODE 3098
Table 14-284. Register Call Summary for Register L3_STCOL_DUMP_MODE 3099
Table 14-285. L3_STCOL_DUMP_SEND 3099
Table 14-286. Register Call Summary for Register L3_STCOL_DUMP_SEND 3099
Table 14-287. L3_STCOL_DUMP_DISABLE 3099
Table 14-288. Register Call Summary for Register L3_STCOL_DUMP_DISABLE 3100
Table 14-289. L3_STCOL_DUMP_ALARM_TRIG 3100
Table 14-290. Register Call Summary for Register L3_STCOL_DUMP_ALARM_TRIG 3101
Table 14-291. L3_STCOL_DUMP_ALARM_MINVAL 3101
Table 14-292. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MINVAL 3101
Table 14-293. L3_STCOL_DUMP_ALARM_MAXVAL 3101
Table 14-294. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MAXVAL 3102
Table 14-295. L3_STCOL_DUMP_ALARM_MODE0 3102
Table 14-296. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE0 3102
Table 14-297. L3_STCOL_DUMP_ALARM_MODE1 3102
Table 14-298. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE1 3103
Table 14-299. L3_STCOL_DUMP_ALARM_MODE2 3103
Table 14-300. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE2 3104
Table 14-301. L3_STCOL_DUMP_ALARM_MODE3 3104
Table 14-302. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE3 3104
Table 14-303. L3_STCOL_DUMP_ALARM_MODE4 3104
Table 14-304. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE4 3105
Table 14-305. L3_STCOL_DUMP_ALARM_MODE5 3105
Table 14-306. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE5 3105
Table 14-307. L3_STCOL_DUMP_ALARM_MODE6 3105
Table 14-308. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE6 3106
Table 14-309. L3_STCOL_DUMP_ALARM_MODE7 3106
Table 14-310. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE7 3106
Table 14-311. L3_STCOL_DUMP_CNT0 3106

Table 14-312. Register Call Summary for Register L3_STCOL_DUMP_CNT0 3107
Table 14-313. L3_STCOL_DUMP_CNT1 3107
Table 14-314. Register Call Summary for Register L3_STCOL_DUMP_CNT1 3107
Table 14-315. L3_STCOL_DUMP_CNT2 3107
Table 14-316. Register Call Summary for Register L3_STCOL_DUMP_CNT2 3108
Table 14-317. L3_STCOL_DUMP_CNT3 3108
Table 14-318. Register Call Summary for Register L3_STCOL_DUMP_CNT3 3108
Table 14-319. L3_STCOL_DUMP_CNT4 3108
Table 14-320. Register Call Summary for Register L3_STCOL_DUMP_CNT4 3108
Table 14-321. L3_STCOL_DUMP_CNT5 3108
Table 14-322. Register Call Summary for Register L3_STCOL_DUMP_CNT5 3109
Table 14-323. L3_STCOL_DUMP_CNT6 3109
Table 14-324. Register Call Summary for Register L3_STCOL_DUMP_CNT6 3109
Table 14-325. L3_STCOL_DUMP_CNT7 3109
Table 14-326. Register Call Summary for Register L3_STCOL_DUMP_CNT7 3109
Table 14-327. L3_STCOL_FILTER_i_GLOBALEN 3109
Table 14-328. Register Call Summary for Register L3_STCOL_FILTER_i_GLOBALEN 3110
Table 14-329. L3_STCOL_FILTER_i_ADDRMIN 3110
Table 14-330. Register Call Summary for Register L3_STCOL_FILTER_i_ADDRMIN 3111
Table 14-331. L3_STCOL_FILTER_i_ADDRMAX 3111
Table 14-332. Register Call Summary for Register L3_STCOL_FILTER_i_ADDRMAX 3111
Table 14-333. L3_STCOL_FILTER_i_ADDREN 3111
Table 14-334. Register Call Summary for Register L3_STCOL_FILTER_i_ADDREN 3112
Table 14-335. L3_STCOL_FILTER_i_EN_k 3112
Table 14-336. Register Call Summary for Register L3_STCOL_FILTER_i_EN_k 3112
Table 14-337. L3_STCOL_FILTER_i_MASK_m_MSTADDR 3112
Table 14-338. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_MSTADDR 3113
Table 14-339. L3_STCOL_FILTER_i_MASK_m_RD 3113
Table 14-340. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_RD 3114
Table 14-341. L3_STCOL_FILTER_i_MASK_m_WR 3114
Table 14-342. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_WR 3115
Table 14-343. L3_STCOL_FILTER_i_MASK_m_ERR 3115
Table 14-344. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_ERR 3115
Table 14-345. L3_STCOL_FILTER_i_MASK_m_USERINFO 3115
Table 14-346. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_USERINFO 3116
Table 14-347. L3_STCOL_FILTER_i_MASK_m_SLVADDR 3116
Table 14-348. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_SLVADDR 3116
Table 14-349. L3_STCOL_FILTER_i_MASK_m_REQUSERINFO 3116
Table 14-350. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_REQUSERINFO 3117
Table 14-351. L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO 3117
Table 14-352. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO 3118
Table 14-353. L3_STCOL_FILTER_i_MATCH_m_MSTADDR 3118
Table 14-354. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_MSTADDR 3118
Table 14-355. L3_STCOL_FILTER_i_MATCH_m_SLVADDR 3118
Table 14-356. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_SLVADDR 3119
Table 14-357. L3_STCOL_FILTER_i_MATCH_m_RD 3119
Table 14-358. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_RD 3120
Table 14-359. L3_STCOL_FILTER_i_MATCH_m_WR 3120
Table 14-360. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_WR 3121
Table 14-361. L3_STCOL_FILTER_i_MATCH_m_ERR 3121
Table 14-362. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_ERR 3121
Table 14-363. L3_STCOL_FILTER_i_MATCH_m_USERINFO 3121
Table 14-364. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_USERINFO 3122
Table 14-365. L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO 3122
Table 14-366. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO 3122
Table 14-367. L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO 3122
Table 14-368. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO 3123
Table 14-369. L3_STCOL_OP_i_THRESHOLD_MINVAL 3123
Table 14-370. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MINVAL 3124
Table 14-371. L3_STCOL_OP_i_THRESHOLD_MAXVAL 3124
Table 14-372. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MAXVAL 3124

Table 14-373. L3_STCOL_OP_i_EVTINFOSEL....3124
Table 14-374. Register Call Summary for Register L3_STCOL_OP_i_EVTINFOSEL....3125
Table 14-375. L3_STCOL_OP_i_SEL....3125
Table 14-376. Register Call Summary for Register L3_STCOL_OP_i_SEL....3125
Table 14-377. Integration Attributes....3127
Table 14-378. Clocks and Resets....3127
Table 14-379. L4_PER1 TAs....3128
Table 14-380. L4_PER1 IAs....3129
Table 14-381. L4_PER2 TAs....3129
Table 14-382. L4_PER2 IAs....3130
Table 14-383. L4_PER3 TAs....3130
Table 14-384. L4_PER3 IAs....3131
Table 14-385. L4_CFG TAs....3131
Table 14-386. L4_CFG IAs....3132
Table 14-387. L4_WKUP TAs....3132
Table 14-388. L4_WKUP IAs....3132
Table 14-389. L4 ConnID Definition....3134
Table 14-390. L4_PER1 Firewall Default Configuration....3136
Table 14-391. L4_PER2 Firewall Default Configuration....3136
Table 14-392. L4_PER3 Firewall Default Configuration....3137
Table 14-393. L4_CFG Firewall Default Configuration....3137
Table 14-394. L4_WKUP Firewall Default Configuration....3138
Table 14-395. Region Allocations for L4_PER1 Interconnect....3140
Table 14-396. Region Allocations for L4_PER2 Interconnect....3142
Table 14-397. Region Allocations for L4_PER3 Interconnect....3143
Table 14-398. Region Allocations for L4_CFG Interconnect....3145
Table 14-399. Region Allocations for L4_WKUP Interconnect....3147
Table 14-400. L4 Firewall Register Description Overview....3148
Table 14-401. L4 CODE Bit Field Definition....3149
Table 14-402. L4 Time-out Link and TA Programming....3150
Table 14-403. Global Initialization of Surrounding Modules....3153
Table 14-404. Protection Violation Error Identification....3154
Table 14-405. Unsupported Command/Address Hole Error Identification....3155
Table 14-406. Reset TA and Module....3155
Table 14-407. Time-Out Configuration....3155
Table 14-408. Firewall Configuration....3156
Table 14-409. L4_PER1 Instance Summary....3157
Table 14-410. L4_PER2 Instance Summary....3158
Table 14-411. L4_PER3 Instance Summary....3158
Table 14-412. L4_CFG Instance Summary....3159
Table 14-413. L4_WKUP Instance Summary....3160
Table 14-414. IA Registers Mapping Summary....3160
Table 14-415. IA Registers Mapping Summary....3161
Table 14-416. IA Registers Mapping Summary....3161
Table 14-417. IA Registers Mapping Summary....3162
Table 14-418. L4_IA_COMPONENT_L....3162
Table 14-419. Register Call Summary for Register L4_IA_COMPONENT_L....3163
Table 14-420. L4_IA_COMPONENT_H....3163
Table 14-421. Register Call Summary for Register L4_IA_COMPONENT_H....3163
Table 14-422. L4_IA_CORE_L....3163
Table 14-423. Register Call Summary for Register L4_IA_CORE_L....3164
Table 14-424. L4_IA_CORE_H....3164
Table 14-425. Register Call Summary for Register L4_IA_CORE_H....3164
Table 14-426. L4_IA_AGENT_CONTROL_L....3164
Table 14-427. Register Call Summary for Register L4_IA_AGENT_CONTROL_L....3165
Table 14-428. L4_IA_AGENT_CONTROL_H....3165
Table 14-429. Register Call Summary for Register L4_IA_AGENT_CONTROL_H....3166
Table 14-430. L4_IA_AGENT_STATUS_L....3166
Table 14-431. Register Call Summary for Register L4_IA_AGENT_STATUS_L....3166
Table 14-432. L4_IA_AGENT_STATUS_H....3166
Table 14-433. Register Call Summary for Register L4_IA_AGENT_STATUS_H....3167

Table 14-434. L4_IA_ERROR_LOG_L....3167
Table 14-435. Register Call Summary for Register L4_IA_ERROR_LOG_L....3168
Table 14-436. L4_IA_ERROR_LOG_H....3168
Table 14-437. Register Call Summary for Register L4_IA_ERROR_LOG_H....3168
Table 14-438. L4_IA_ERROR_LOG_ADDR_L....3168
Table 14-439. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_L....3169
Table 14-440. L4_IA_ERROR_LOG_ADDR_H....3169
Table 14-441. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_H....3169
Table 14-442. CFG_TA Register Mapping Summary 1....3170
Table 14-443. CFG_TA Register Mapping Summary 2....3170
Table 14-444. CFG_TA Register Mapping Summary 3....3170
Table 14-445. CFG_TA Register Mapping Summary 4....3170
Table 14-446. CFG_TA Register Mapping Summary 5....3171
Table 14-447. CFG_TA Register Mapping Summary 6....3171
Table 14-448. CFG_TA Register Mapping Summary 7....3171
Table 14-449. CFG_TA Register Mapping Summary 8....3172
Table 14-450. CFG_TA Register Mapping Summary 9....3172
Table 14-451. CFG_TA Register Mapping Summary 10....3172
Table 14-452. CFG_TA Register Mapping Summary 11....3172
Table 14-453. CFG_TA Register Mapping Summary 12....3173
Table 14-454. CFG_TA Register Mapping Summary 13....3173
Table 14-455. CFG_TA Register Mapping Summary 14....3173
Table 14-456. PER1_TA Register Mapping Summary 1....3173
Table 14-457. PER1_TA Register Mapping Summary 2....3174
Table 14-458. PER1_TA Register Mapping Summary 3....3174
Table 14-459. PER1_TA Register Mapping Summary 4....3174
Table 14-460. PER1_TA Register Mapping Summary 5....3175
Table 14-461. PER1_TA Register Mapping Summary 6....3175
Table 14-462. PER1_TA Register Mapping Summary 7....3175
Table 14-463. PER1_TA Register Mapping Summary 8....3176
Table 14-464. PER1_TA Register Mapping Summary 9....3176
Table 14-465. PER2_TA Register Mapping Summary 1....3176
Table 14-466. PER2_TA Register Mapping Summary 2....3177
Table 14-467. PER2_TA Register Mapping Summary 3....3177
Table 14-468. PER2_TA Register Mapping Summary 4....3177
Table 14-469. PER2_TA Register Mapping Summary 5....3178
Table 14-470. PER2_TA Register Mapping Summary 6....3178
Table 14-471. PER2_TA Register Mapping Summary 7....3178
Table 14-472. PER3_TA Register Mapping Summary 1....3179
Table 14-473. PER3_TA Register Mapping Summary 2....3179
Table 14-474. PER3_TA Register Mapping Summary 3....3179
Table 14-475. PER3_TA Register Mapping Summary 4....3180
Table 14-476. PER3_TA Register Mapping Summary 5....3180
Table 14-477. PER3_TA Register Mapping Summary 6....3180
Table 14-478. PER3_TA Register Mapping Summary 7....3181
Table 14-479. PER3_TA Register Mapping Summary 8....3181
Table 14-480. PER3_TA Register Mapping Summary 9....3181
Table 14-481. WKUP_TA Register Mapping Summary 1....3182
Table 14-482. WKUP_TA Register Mapping Summary 2....3182
Table 14-483. WKUP_TA Register Mapping Summary 3....3182
Table 14-484. L4_TA_COMPONENT_H....3183
Table 14-485. Register Call Summary for Register L4_TA_COMPONENT_H....3183
Table 14-486. L4_TA_COMPONENT_L....3183
Table 14-487. Register Call Summary for Register L4_TA_COMPONENT_L....3183
Table 14-488. L4_TA_CORE_L....3183
Table 14-489. Register Call Summary for Register L4_TA_CORE_L....3184
Table 14-490. L4_TA_CORE_H....3184
Table 14-491. Register Call Summary for Register L4_TA_CORE_H....3184
Table 14-492. L4_TA_AGENT_CONTROL_L....3184
Table 14-493. Register Call Summary for Register L4_TA_AGENT_CONTROL_L....3185
Table 14-494. L4_TA_AGENT_CONTROL_H....3185

Table 14-495. Register Call Summary for Register L4_TA_AGENT_CONTROL_H....3185
Table 14-496. L4_TA_AGENT_STATUS_L....3186
Table 14-497. Register Call Summary for Register L4_TA_AGENT_STATUS_L....3186
Table 14-498. L4_TA_AGENT_STATUS_H....3186
Table 14-499. Register Call Summary for Register L4_TA_AGENT_STATUS_H....3186
Table 14-500. LA Register Mapping Summary....3187
Table 14-501. LA Register Mapping Summary....3187
Table 14-502. L4_LA_COMPONENT_L....3187
Table 14-503. Register Call Summary for Register L4_LA_COMPONENT_L....3188
Table 14-504. L4_LA_COMPONENT_H....3188
Table 14-505. Register Call Summary for Register L4_LA_COMPONENT_H....3188
Table 14-506. L4_LA_NETWORK_L....3188
Table 14-507. Register Call Summary for Register L4_LA_NETWORK_L....3189
Table 14-508. L4_LA_NETWORK_H....3189
Table 14-509. Register Call Summary for Register L4_LA_NETWORK_H....3189
Table 14-510. L4_LA_INITIATOR_INFO_L....3189
Table 14-511. Register Call Summary for Register L4_LA_INITIATOR_INFO_L....3190
Table 14-512. Reset value for L4_LA_INITIATOR_INFO_L....3190
Table 14-513. L4_LA_INITIATOR_INFO_H....3190
Table 14-514. Register Call Summary for Register L4_LA_INITIATOR_INFO_H....3191
Table 14-515. Reset value for L4_LA_INITIATOR_INFO_H....3191
Table 14-516. L4_LA_NETWORK_CONTROL_L....3191
Table 14-517. Register Call Summary for Register L4_LA_NETWORK_CONTROL_L....3192
Table 14-518. L4_LA_NETWORK_CONTROL_H....3192
Table 14-519. Register Call Summary for Register L4_LA_NETWORK_CONTROL_H....3192
Table 14-520. L4_LA_FLAG_MASK_j_L....3192
Table 14-521. Register Call Summary for Register L4_LA_FLAG_MASK_j_L....3193
Table 14-522. Reset Value for L4_LA_FLAG_MASK_j_L....3193
Table 14-523. L4_LA_FLAG_MASK_j_H....3193
Table 14-524. Register Call Summary for Register L4_LA_FLAG_MASK_j_H....3193
Table 14-525. L4_LA_FLAG_STATUS_j_L....3193
Table 14-526. Register Call Summary for Register L4_LA_FLAG_STATUS_j_L....3194
Table 14-527. L4_LA_FLAG_STATUS_j_H....3194
Table 14-528. Register Call Summary for Register L4_LA_FLAG_STATUS_j_H....3194
Table 14-529. L4 AP Register Summary....3194
Table 14-530. L4 AP Register Summary....3195
Table 14-531. L4_AP_COMPONENT_L....3195
Table 14-532. Register Call Summary for Register L4_AP_COMPONENT_L....3196
Table 14-533. L4_AP_COMPONENT_H....3196
Table 14-534. Register Call Summary for Register L4_AP_COMPONENT_H....3196
Table 14-535. L4_AP_SEGMENT_i_L....3196
Table 14-536. Register Call Summary for Register L4_AP_SEGMENT_i_L....3196
Table 14-537. L4_AP_SEGMENT_i_H....3196
Table 14-538. Register Call Summary for Register L4_AP_SEGMENT_i_H....3197
Table 14-539. Reset Value for L4_AP_SEGMENT_i....3197
Table 14-540. L4_AP_PROT_GROUP_MEMBERS_k_L....3197
Table 14-541. Register Call Summary for Register L4_AP_PROT_GROUP_MEMBERS_k_L....3198
Table 14-542. L4_AP_PROT_GROUP_MEMBERS_k_H....3198
Table 14-543. Register Call Summary for Register L4_AP_PROT_GROUP_MEMBERS_k_H....3198
Table 14-544. L4_AP_PROT_GROUP_ROLES_k_L....3198
Table 14-545. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_L....3198
Table 14-546. Reset Value for L4_AP_PROT_GROUP_ROLES_k....3199
Table 14-547. L4_AP_PROT_GROUP_ROLES_k_H....3199
Table 14-548. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_H....3199
Table 14-549. L4_AP_REGION_l_L....3199
Table 14-550. Register Call Summary for Register L4_AP_REGION_l_L....3200
Table 14-551. L4_AP_REGION_l_H....3200
Table 14-552. Register Call Summary for Register L4_AP_REGION_l_H....3200
Table 14-553. L4_AP_REGION_l Reset Value for L4 PER1....3201
Table 14-554. L4_AP_REGION_l Reset Value for L4 PER2....3203
Table 14-555. L4_AP_REGION_l Reset Value for L4 PER3....3204

Table 14-556. L4_AP_REGION_l Reset Value for L4 CFG....3206
Table 14-557. L4_AP_REGION_l Reset values for L4 WKUP....3209
Table 15-1. DMM Integration Attributes....3218
Table 15-2. DMM Clocks and Resets....3218
Table 15-3. DMM Hardware Requests....3219
Table 15-4. DMM TILER Container Geometry....3219
Table 15-5. DMM Clocks....3221
Table 15-6. DMM Local Power-Management Features....3221
Table 15-7. DMM Hardware Status Features....3221
Table 15-8. Events....3221
Table 15-9. ConnIDs vs PEG Priority Register Fields....3231
Table 15-10. Well-Formed Tiled Mode 2D Block Requests....3235
Table 15-11. Coding and Description of TILER Modes....3240
Table 15-12. Coding and Description of TILER Orientations....3240
Table 15-13. Tiled Mode Container Characteristics....3241
Table 15-14. TILER Aliased View in the L3 Interconnect Mapping....3251
Table 15-15. Address Format in the TILER-Specific Address Map....3251
Table 15-16. Address Format of the TILER Aliased View in the System Address Map....3251
Table 15-17. Minimum Buffer Size to Efficiently Handle Lines of up to N Elements....3253
Table 15-18. Memory Data Payload for Buffered Raster-Based Initiators on 32-Bit DDR2 or DDR3....3254
Table 15-19. Address Definition....3259
Table 15-20. Configuration....3259
Table 15-21. Address Definition....3260
Table 15-22. Configuration....3260
Table 15-23. Address Upper Bits Shifting....3262
Table 15-24. Global Initialization of Surrounding Modules....3263
Table 15-25. DMM Global Initialization....3263
Table 15-26. Coding and Description of TILER Modes....3263
Table 15-27. Configuration Settings and LUT Refill....3263
Table 15-28. Interleaving Settings....3264
Table 15-29. Aliased Tiled View Orientation Settings and LUT Refill....3264
Table 15-30. Priority Settings....3264
Table 15-31. Error Handling....3265
Table 15-32. 29-Bit View Address Offset and 33-Bit Full TILER Address for an 8-Bit Frame-Buffer....3266
Table 15-33. 29-Bit View Address Offset and 33-Bit Full TILER Address for a 1D Object....3267
Table 15-34. DMM Instance Summary....3269
Table 15-35. DMM Registers Mapping Summary....3269
Table 15-36. DMM_REVISION....3270
Table 15-37. DMM_HWINFO....3270
Table 15-38. DMM_LISA_HWINFO....3270
Table 15-39. DMM_SYSCONFIG....3270
Table 15-40. DMM_LISA_LOCK....3271
Table 15-41. DMM_EMERGENCY....3271
Table 15-42. DMM_LISA_MAP_i....3272
Table 15-43. DMM_TILER_HWINFO....3272
Table 15-44. DMM_TILER_OR0....3273
Table 15-45. DMM_TILER_OR1....3274
Table 15-46. DMM_PAT_HWINFO....3275
Table 15-47. DMM_PAT_GEOMETRY....3275
Table 15-48. DMM_PAT_CONFIG....3276
Table 15-49. DMM_PAT_VIEW0....3277
Table 15-50. DMM_PAT_VIEW1....3278
Table 15-51. DMM_PAT_VIEW_MAP_i....3279
Table 15-52. DMM_PAT_VIEW_MAP_BASE....3280
Table 15-53. DMM_PAT_IRQ_EOI....3281
Table 15-54. DMM_PAT_IRQSTATUS_RAW....3281
Table 15-55. DMM_PAT_IRQSTATUS....3285
Table 15-56. DMM_PAT_IRQENABLE_SET....3289
Table 15-57. DMM_PAT_IRQENABLE_CLR....3294
Table 15-58. DMM_PAT_STATUS_i....3298
Table 15-59. DMM_PAT_DESCR_i....3299

Table 15-60. DMM_PAT_AREA_i........ 3299
Table 15-61. DMM_PAT_CTRL_i........ 3300
Table 15-62. DMM_PAT_DATA_i........ 3300
Table 15-63. DMM_PEG_HWINFO........ 3300
Table 15-64. DMM_PEG_PRIO_k........ 3301
Table 15-65. DMM_PEG_PRIO_PAT........ 3302
Table 15-66. EMIF Module I/O signals........ 3306
Table 15-67. EMIF Integration Attributes........ 3308
Table 15-68. EMIF Clocks and Resets........ 3309
Table 15-69. EMIF Hardware Requests........ 3309
Table 15-70. MAddrSpace Mapping........ 3311
Table 15-71. FIFO Allocation........ 3312
Table 15-72. Events........ 3317
Table 15-73. Load Value For The EMR(1) Register During DDR2 SDRAM Initialization........ 3319
Table 15-74. Load Value For The MR Register During DDR2 SDRAM Initialization........ 3319
Table 15-75. Another Load Value For The MR Register During DDR2 SDRAM Initialization........ 3320
Table 15-76. Load Value For The MR2 Register During DDR3 SDRAM Initialization........ 3320
Table 15-77. Load Value For The MR1 Register During DDR3 SDRAM Initialization........ 3321
Table 15-78. Load Value For The MR0 Register During DDR3 SDRAM Initialization........ 3321
Table 15-79. 64-Byte Linear Read Starting at Address 0x0 (DDR2 and DDR3)........ 3324
Table 15-80. 64-Byte Linear Read Starting at Address 0x8 (DDR2)........ 3324
Table 15-81. 64-Byte Linear Read Starting at Address 0x10 (DDR2 and DDR3)........ 3324
Table 15-82. 64-Byte Linear Read Starting at Address 0x18 (DDR2 and DDR3)........ 3324
Table 15-83. Turnaround Time........ 3324
Table 15-84. SDRAM Addressing Space........ 3325
Table 15-85. Local Address to SDRAM Address Mapping for IBANK_POS = 0 and EBANK_POS = 0........ 3326
Table 15-86. Local Address to SDRAM Address Mapping for IBANK_POS = 1 and EBANK_POS = 0........ 3326
Table 15-87. Local Address to SDRAM Address Mapping for IBANK_POS = 2 and EBANK_POS = 0........ 3326
Table 15-88. Local Address to SDRAM Address Mapping for IBANK_POS = 3 and EBANK_POS = 0........ 3327
Table 15-89. Local Address to SDRAM Address Mapping for IBANK_POS = 0 and EBANK_POS = 1........ 3327
Table 15-90. Local Address to SDRAM Address Mapping for IBANK_POS = 1 and EBANK_POS = 1........ 3328
Table 15-91. Local Address to SDRAM Address Mapping for IBANK_POS = 2 and EBANK_POS = 1........ 3328
Table 15-92. Local Address to SDRAM Address Mapping for IBANK_POS = 3 and EBANK_POS = 1........ 3329
Table 15-93. Performance Counter Filter Configuration........ 3332
Table 15-94. Global Initialization of Surrounding Modules........ 3335
Table 15-95. EMIF Configuration Sequence........ 3335
Table 15-96. EMIF Output Impedance Calibration Mode........ 3342
Table 15-97. EMIF SDRAM Self-Refresh Entering........ 3342
Table 15-98. EMIF SDRAM Self-Refresh Exiting........ 3342
Table 15-99. EMIF SDRAM Power-Down Mode Entering........ 3342
Table 15-100. EMIF SDRAM Power-Down Mode Exiting........ 3342
Table 15-101. EMIF ECC Configuration........ 3343
Table 15-102. EMIF Instance Summary........ 3344
Table 15-103. EMIF Registers Mapping Summary........ 3344
Table 15-104. EMIF_REVISION........ 3348
Table 15-105. Register Call Summary for Register EMIF_REVISION........ 3348
Table 15-106. EMIF_STATUS........ 3348
Table 15-107. Register Call Summary for Register EMIF_STATUS........ 3349
Table 15-108. EMIF_SDRAM_CONFIG........ 3349
Table 15-109. Register Call Summary for Register EMIF_SDRAM_CONFIG........ 3351
Table 15-110. EMIF_SDRAM_CONFIG_2........ 3351
Table 15-111. Register Call Summary for Register EMIF_SDRAM_CONFIG_2........ 3352
Table 15-112. EMIF_SDRAM_REFRESH_CONTROL........ 3352
Table 15-113. Register Call Summary for Register EMIF_SDRAM_REFRESH_CONTROL........ 3353
Table 15-114. EMIF_SDRAM_REFRESH_CONTROL_SHADOW........ 3353
Table 15-115. Register Call Summary for Register EMIF_SDRAM_REFRESH_CONTROL_SHADOW........ 3354
Table 15-116. EMIF_SDRAM_TIMING_1........ 3354
Table 15-117. Register Call Summary for Register EMIF_SDRAM_TIMING_1........ 3355
Table 15-118. EMIF_SDRAM_TIMING_1_SHADOW........ 3355
Table 15-119. Register Call Summary for Register EMIF_SDRAM_TIMING_1_SHADOW........ 3355
Table 15-120. EMIF_SDRAM_TIMING_2........ 3355

Table 15-121. Register Call Summary for Register EMIF_SDRAM_TIMING_2 3356
Table 15-122. EMIF_SDRAM_TIMING_2_SHADOW 3356
Table 15-123. Register Call Summary for Register EMIF_SDRAM_TIMING_2_SHADOW 3357
Table 15-124. EMIF_SDRAM_TIMING_3 3357
Table 15-125. Register Call Summary for Register EMIF_SDRAM_TIMING_3 3358
Table 15-126. EMIF_SDRAM_TIMING_3_SHADOW 3358
Table 15-127. Register Call Summary for Register EMIF_SDRAM_TIMING_3_SHADOW 3358
Table 15-128. EMIF_LPDDR2_NVM_TIMING 3358
Table 15-129. Register Call Summary for Register EMIF_LPDDR2_NVM_TIMING 3359
Table 15-130. EMIF_LPDDR2_NVM_TIMING_SHADOW 3359
Table 15-131. Register Call Summary for Register EMIF_LPDDR2_NVM_TIMING_SHADOW 3359
Table 15-132. EMIF_POWER_MANAGEMENT_CONTROL 3359
Table 15-133. Register Call Summary for Register EMIF_POWER_MANAGEMENT_CONTROL 3360
Table 15-134. EMIF_POWER_MANAGEMENT_CONTROL_SHADOW 3360
Table 15-135. Register Call Summary for Register EMIF_POWER_MANAGEMENT_CONTROL_SHADOW 3361
Table 15-136. EMIF_OCP_CONFIG 3361
Table 15-137. Register Call Summary for Register EMIF_OCP_CONFIG 3361
Table 15-138. EMIF_OCP_CONFIG_VALUE_1 3361
Table 15-139. Register Call Summary for Register EMIF_OCP_CONFIG_VALUE_1 3362
Table 15-140. EMIF_OCP_CONFIG_VALUE_2 3362
Table 15-141. Register Call Summary for Register EMIF_OCP_CONFIG_VALUE_2 3362
Table 15-142. EMIF_IODFT_TLGC 3362
Table 15-143. Register Call Summary for Register EMIF_IODFT_TLGC 3363
Table 15-144. EMIF_PERFORMANCE_COUNTER_1 3363
Table 15-145. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_1 3363
Table 15-146. EMIF_PERFORMANCE_COUNTER_2 3363
Table 15-147. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_2 3364
Table 15-148. EMIF_PERFORMANCE_COUNTER_CONFIG 3364
Table 15-149. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_CONFIG 3365
Table 15-150. EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT 3365
Table 15-151. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT 3365
Table 15-152. EMIF_PERFORMANCE_COUNTER_TIME 3365
Table 15-153. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_TIME 3366
Table 15-154. EMIF_MISC_REG 3366
Table 15-155. Register Call Summary for Register EMIF_MISC_REG 3366
Table 15-156. EMIF_DLL_CALIB_CTRL 3366
Table 15-157. Register Call Summary for Register EMIF_DLL_CALIB_CTRL 3367
Table 15-158. EMIF_DLL_CALIB_CTRL_SHADOW 3367
Table 15-159. Register Call Summary for Register EMIF_DLL_CALIB_CTRL_SHADOW 3367
Table 15-160. EMIF_END_OF_INTERRUPT 3367
Table 15-161. Register Call Summary for Register EMIF_END_OF_INTERRUPT 3368
Table 15-162. EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS 3368
Table 15-163. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS 3369
Table 15-164. EMIF_SYSTEM_OCP_INTERRUPT_STATUS 3369
Table 15-165. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_STATUS 3369
Table 15-166. EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET 3369
Table 15-167. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET 3370
Table 15-168. EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR 3370
Table 15-169. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR 3371
Table 15-170. EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG 3371
Table 15-171. Register Call Summary for Register EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG 3372
Table 15-172. EMIF_TEMP_ALERT_CONFIG 3372
Table 15-173. Register Call Summary for Register EMIF_TEMP_ALERT_CONFIG 3373
Table 15-174. EMIF_OCP_ERROR_LOG 3373
Table 15-175. Register Call Summary for Register EMIF_OCP_ERROR_LOG 3373
Table 15-176. EMIF_READ_WRITE_LEVELING_RAMP_WINDOW 3373
Table 15-177. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_RAMP_WINDOW 3374
Table 15-178. EMIF_READ_WRITE_LEVELING_RAMP_CONTROL 3374
Table 15-179. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_RAMP_CONTROL 3375
Table 15-180. EMIF_READ_WRITE_LEVELING_CONTROL 3375
Table 15-181. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_CONTROL 3376

Table 15-182. EMIF_DDR_PHY_CONTROL_1......3376
Table 15-183. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_1......3377
Table 15-184. EMIF_DDR_PHY_CONTROL_1_SHADOW......3377
Table 15-185. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_1_SHADOW......3378
Table 15-186. EMIF_DDR_PHY_CONTROL_2......3378
Table 15-187. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_2......3378
Table 15-188. EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING......3378
Table 15-189. Register Call Summary for Register EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING......3379
Table 15-190. EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING......3379
Table 15-191. Register Call Summary for Register EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING..3380
Table 15-192. EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING......3380
Table 15-193. Register Call Summary for Register EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING..3381
Table 15-194. EMIF_ECC_CTRL_REG......3381
Table 15-195. Register Call Summary for Register EMIF_ECC_CTRL_REG......3382
Table 15-196. EMIF_ECC_ADDRESS_RANGE_1......3382
Table 15-197. Register Call Summary for Register EMIF_ECC_ADDRESS_RANGE_1......3382
Table 15-198. EMIF_ECC_ADDRESS_RANGE_2......3383
Table 15-199. Register Call Summary for Register EMIF_ECC_ADDRESS_RANGE_2......3383
Table 15-200. EMIF_READ_WRITE_EXECUTION_THRESHOLD......3383
Table 15-201. Register Call Summary for Register EMIF_READ_WRITE_EXECUTION_THRESHOLD......3384
Table 15-202. EMIF_COS_CONFIG......3384
Table 15-203. Register Call Summary for Register EMIF_COS_CONFIG......3384
Table 15-204. EMIF_1B_ECC_ERR_CNT......3384
Table 15-205. Register Call Summary for Register EMIF_1B_ECC_ERR_CNT......3385
Table 15-206. EMIF_1B_ECC_ERR_THRSH......3385
Table 15-207. Register Call Summary for Register EMIF_1B_ECC_ERR_THRSH......3385
Table 15-208. EMIF_1B_ECC_ERR_DIST_1......3385
Table 15-209. Register Call Summary for Register EMIF_1B_ECC_ERR_DIST_1......3386
Table 15-210. EMIF_1B_ECC_ERR_ADDR_LOG......3386
Table 15-211. Register Call Summary for Register EMIF_1B_ECC_ERR_ADDR_LOG......3386
Table 15-212. EMIF_2B_ECC_ERR_ADDR_LOG......3386
Table 15-213. Register Call Summary for Register EMIF_2B_ECC_ERR_ADDR_LOG......3387
Table 15-214. EMIF_PHY_STATUS_1......3387
Table 15-215. Register Call Summary for Register EMIF_PHY_STATUS_1......3387
Table 15-216. EMIF_PHY_STATUS_2......3387
Table 15-217. Register Call Summary for Register EMIF_PHY_STATUS_2......3387
Table 15-218. EMIF_PHY_STATUS_3......3388
Table 15-219. Register Call Summary for Register EMIF_PHY_STATUS_3......3388
Table 15-220. EMIF_PHY_STATUS_4......3388
Table 15-221. Register Call Summary for Register EMIF_PHY_STATUS_4......3388
Table 15-222. EMIF_PHY_STATUS_5......3388
Table 15-223. Register Call Summary for Register EMIF_PHY_STATUS_5......3389
Table 15-224. EMIF_PHY_STATUS_6......3389
Table 15-225. Register Call Summary for Register EMIF_PHY_STATUS_6......3389
Table 15-226. EMIF_PHY_STATUS_7......3389
Table 15-227. Register Call Summary for Register EMIF_PHY_STATUS_7......3390
Table 15-228. EMIF_PHY_STATUS_8......3390
Table 15-229. Register Call Summary for Register EMIF_PHY_STATUS_8......3390
Table 15-230. EMIF_PHY_STATUS_9......3390
Table 15-231. Register Call Summary for Register EMIF_PHY_STATUS_9......3390
Table 15-232. EMIF_PHY_STATUS_10......3390
Table 15-233. Register Call Summary for Register EMIF_PHY_STATUS_10......3391
Table 15-234. EMIF_PHY_STATUS_11......3391
Table 15-235. Register Call Summary for Register EMIF_PHY_STATUS_11......3391
Table 15-236. EMIF_PHY_STATUS_12......3391
Table 15-237. Register Call Summary for Register EMIF_PHY_STATUS_12......3392
Table 15-238. EMIF_PHY_STATUS_13......3392
Table 15-239. Register Call Summary for Register EMIF_PHY_STATUS_13......3392
Table 15-240. EMIF_PHY_STATUS_14......3392
Table 15-241. Register Call Summary for Register EMIF_PHY_STATUS_14......3393
Table 15-242. EMIF_PHY_STATUS_15......3393

Table 15-243. Register Call Summary for Register EMIF_PHY_STATUS_15....3393
Table 15-244. EMIF_PHY_STATUS_16....3393
Table 15-245. Register Call Summary for Register EMIF_PHY_STATUS_16....3393
Table 15-246. EMIF_PHY_STATUS_17....3393
Table 15-247. Register Call Summary for Register EMIF_PHY_STATUS_17....3394
Table 15-248. EMIF_PHY_STATUS_18....3394
Table 15-249. Register Call Summary for Register EMIF_PHY_STATUS_18....3394
Table 15-250. EMIF_PHY_STATUS_19....3394
Table 15-251. Register Call Summary for Register EMIF_PHY_STATUS_19....3395
Table 15-252. EMIF_PHY_STATUS_20....3395
Table 15-253. Register Call Summary for Register EMIF_PHY_STATUS_20....3395
Table 15-254. EMIF_PHY_STATUS_21....3395
Table 15-255. Register Call Summary for Register EMIF_PHY_STATUS_21....3395
Table 15-256. EMIF_PHY_STATUS_22....3395
Table 15-257. Register Call Summary for Register EMIF_PHY_STATUS_22....3396
Table 15-258. EMIF_PHY_STATUS_23....3396
Table 15-259. Register Call Summary for Register EMIF_PHY_STATUS_23....3396
Table 15-260. EMIF_PHY_STATUS_24....3396
Table 15-261. Register Call Summary for Register EMIF_PHY_STATUS_24....3397
Table 15-262. EMIF_PHY_STATUS_25....3397
Table 15-263. Register Call Summary for Register EMIF_PHY_STATUS_25....3397
Table 15-264. EMIF_PHY_STATUS_26....3397
Table 15-265. Register Call Summary for Register EMIF_PHY_STATUS_26....3398
Table 15-266. EMIF_PHY_STATUS_27....3398
Table 15-267. Register Call Summary for Register EMIF_PHY_STATUS_27....3398
Table 15-268. EMIF_PHY_STATUS_28....3398
Table 15-269. Register Call Summary for Register EMIF_PHY_STATUS_28....3399
Table 15-270. EMIF_EXT_PHY_CONTROL_1....3399
Table 15-271. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_1....3400
Table 15-272. EMIF_EXT_PHY_CONTROL_1_SHADOW....3400
Table 15-273. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_1_SHADOW....3401
Table 15-274. EMIF_EXT_PHY_CONTROL_2....3401
Table 15-275. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_2....3402
Table 15-276. EMIF_EXT_PHY_CONTROL_2_SHADOW....3402
Table 15-277. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_2_SHADOW....3402
Table 15-278. EMIF_EXT_PHY_CONTROL_3....3402
Table 15-279. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_3....3403
Table 15-280. EMIF_EXT_PHY_CONTROL_3_SHADOW....3403
Table 15-281. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_3_SHADOW....3403
Table 15-282. EMIF_EXT_PHY_CONTROL_4....3404
Table 15-283. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_4....3404
Table 15-284. EMIF_EXT_PHY_CONTROL_4_SHADOW....3404
Table 15-285. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_4_SHADOW....3405
Table 15-286. EMIF_EXT_PHY_CONTROL_5....3405
Table 15-287. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_5....3405
Table 15-288. EMIF_EXT_PHY_CONTROL_5_SHADOW....3405
Table 15-289. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_5_SHADOW....3406
Table 15-290. EMIF_EXT_PHY_CONTROL_6....3406
Table 15-291. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_6....3406
Table 15-292. EMIF_EXT_PHY_CONTROL_6_SHADOW....3406
Table 15-293. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_6_SHADOW....3407
Table 15-294. EMIF_EXT_PHY_CONTROL_7....3407
Table 15-295. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_7....3408
Table 15-296. EMIF_EXT_PHY_CONTROL_7_SHADOW....3408
Table 15-297. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_7_SHADOW....3408
Table 15-298. EMIF_EXT_PHY_CONTROL_8....3408
Table 15-299. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_8....3409
Table 15-300. EMIF_EXT_PHY_CONTROL_8_SHADOW....3409
Table 15-301. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_8_SHADOW....3409
Table 15-302. EMIF_EXT_PHY_CONTROL_9....3409
Table 15-303. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_9....3410

Table 15-304. EMIF_EXT_PHY_CONTROL_9_SHADOW.......3410
Table 15-305. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_9_SHADOW.......3411
Table 15-306. EMIF_EXT_PHY_CONTROL_10.......3411
Table 15-307. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_10.......3411
Table 15-308. EMIF_EXT_PHY_CONTROL_10_SHADOW.......3411
Table 15-309. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_10_SHADOW.......3412
Table 15-310. EMIF_EXT_PHY_CONTROL_11.......3412
Table 15-311. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_11.......3412
Table 15-312. EMIF_EXT_PHY_CONTROL_11_SHADOW.......3412
Table 15-313. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_11_SHADOW.......3413
Table 15-314. EMIF_EXT_PHY_CONTROL_12.......3413
Table 15-315. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_12.......3413
Table 15-316. EMIF_EXT_PHY_CONTROL_12_SHADOW.......3413
Table 15-317. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_12_SHADOW.......3414
Table 15-318. EMIF_EXT_PHY_CONTROL_13.......3414
Table 15-319. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_13.......3415
Table 15-320. EMIF_EXT_PHY_CONTROL_13_SHADOW.......3415
Table 15-321. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_13_SHADOW.......3415
Table 15-322. EMIF_EXT_PHY_CONTROL_14.......3415
Table 15-323. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_14.......3416
Table 15-324. EMIF_EXT_PHY_CONTROL_14_SHADOW.......3416
Table 15-325. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_14_SHADOW.......3416
Table 15-326. EMIF_EXT_PHY_CONTROL_15.......3416
Table 15-327. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_15.......3417
Table 15-328. EMIF_EXT_PHY_CONTROL_15_SHADOW.......3417
Table 15-329. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_15_SHADOW.......3418
Table 15-330. EMIF_EXT_PHY_CONTROL_16.......3418
Table 15-331. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_16.......3418
Table 15-332. EMIF_EXT_PHY_CONTROL_16_SHADOW.......3418
Table 15-333. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_16_SHADOW.......3419
Table 15-334. EMIF_EXT_PHY_CONTROL_17.......3419
Table 15-335. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_17.......3419
Table 15-336. EMIF_EXT_PHY_CONTROL_17_SHADOW.......3419
Table 15-337. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_17_SHADOW.......3420
Table 15-338. EMIF_EXT_PHY_CONTROL_18.......3420
Table 15-339. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_18.......3421
Table 15-340. EMIF_EXT_PHY_CONTROL_18_SHADOW.......3421
Table 15-341. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_18_SHADOW.......3421
Table 15-342. EMIF_EXT_PHY_CONTROL_19.......3421
Table 15-343. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_19.......3422
Table 15-344. EMIF_EXT_PHY_CONTROL_19_SHADOW.......3422
Table 15-345. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_19_SHADOW.......3422
Table 15-346. EMIF_EXT_PHY_CONTROL_20.......3422
Table 15-347. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_20.......3423
Table 15-348. EMIF_EXT_PHY_CONTROL_20_SHADOW.......3423
Table 15-349. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_20_SHADOW.......3423
Table 15-350. EMIF_EXT_PHY_CONTROL_21.......3423
Table 15-351. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_21.......3424
Table 15-352. EMIF_EXT_PHY_CONTROL_21_SHADOW.......3424
Table 15-353. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_21_SHADOW.......3425
Table 15-354. EMIF_EXT_PHY_CONTROL_22.......3425
Table 15-355. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_22.......3425
Table 15-356. EMIF_EXT_PHY_CONTROL_22_SHADOW.......3425
Table 15-357. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_22_SHADOW.......3426
Table 15-358. EMIF_EXT_PHY_CONTROL_23.......3426
Table 15-359. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_23.......3426
Table 15-360. EMIF_EXT_PHY_CONTROL_23_SHADOW.......3426
Table 15-361. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_23_SHADOW.......3426
Table 15-362. EMIF_EXT_PHY_CONTROL_24.......3427
Table 15-363. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_24.......3427
Table 15-364. EMIF_EXT_PHY_CONTROL_24_SHADOW.......3427

Table 15-365. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_24_SHADOW..3428
Table 15-366. EMIF_EXT_PHY_CONTROL_25.. 3428
Table 15-367. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_25..3429
Table 15-368. EMIF_EXT_PHY_CONTROL_25_SHADOW.. 3429
Table 15-369. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_25_SHADOW..3430
Table 15-370. EMIF_EXT_PHY_CONTROL_26.. 3430
Table 15-371. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_26..3431
Table 15-372. EMIF_EXT_PHY_CONTROL_26_SHADOW.. 3431
Table 15-373. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_26_SHADOW..3431
Table 15-374. EMIF_EXT_PHY_CONTROL_27.. 3431
Table 15-375. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_27..3432
Table 15-376. EMIF_EXT_PHY_CONTROL_27_SHADOW.. 3432
Table 15-377. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_27_SHADOW..3433
Table 15-378. EMIF_EXT_PHY_CONTROL_28.. 3433
Table 15-379. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_28..3433
Table 15-380. EMIF_EXT_PHY_CONTROL_28_SHADOW.. 3433
Table 15-381. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_28_SHADOW..3434
Table 15-382. EMIF_EXT_PHY_CONTROL_29.. 3434
Table 15-383. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_29..3435
Table 15-384. EMIF_EXT_PHY_CONTROL_29_SHADOW.. 3435
Table 15-385. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_29_SHADOW..3435
Table 15-386. EMIF_EXT_PHY_CONTROL_30.. 3435
Table 15-387. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_30..3436
Table 15-388. EMIF_EXT_PHY_CONTROL_30_SHADOW.. 3436
Table 15-389. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_30_SHADOW..3436
Table 15-390. EMIF_EXT_PHY_CONTROL_31.. 3437
Table 15-391. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_31..3437
Table 15-392. EMIF_EXT_PHY_CONTROL_31_SHADOW.. 3437
Table 15-393. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_31_SHADOW..3438
Table 15-394. EMIF_EXT_PHY_CONTROL_32.. 3438
Table 15-395. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_32..3438
Table 15-396. EMIF_EXT_PHY_CONTROL_32_SHADOW.. 3438
Table 15-397. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_32_SHADOW..3439
Table 15-398. EMIF_EXT_PHY_CONTROL_33.. 3439
Table 15-399. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_33..3440
Table 15-400. EMIF_EXT_PHY_CONTROL_33_SHADOW.. 3440
Table 15-401. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_33_SHADOW..3440
Table 15-402. EMIF_EXT_PHY_CONTROL_34.. 3440
Table 15-403. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_34..3441
Table 15-404. EMIF_EXT_PHY_CONTROL_34_SHADOW.. 3441
Table 15-405. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_34_SHADOW..3441
Table 15-406. EMIF_EXT_PHY_CONTROL_35.. 3442
Table 15-407. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_35..3442
Table 15-408. EMIF_EXT_PHY_CONTROL_35_SHADOW.. 3442
Table 15-409. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_35_SHADOW..3443
Table 15-410. EMIF_EXT_PHY_CONTROL_36.. 3443
Table 15-411. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_36..3444
Table 15-412. EMIF_EXT_PHY_CONTROL_36_SHADOW.. 3444
Table 15-413. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_36_SHADOW..3445
Table 15-414. GPMC I/O Description.. 3449
Table 15-415. GPMC Pin Multiplexing Options...3449
Table 15-416. GPMC Integration Attributes...3452
Table 15-417. GPMC Clocks and Resets.. 3452
Table 15-418. GPMC Hardware Requests.. 3453
Table 15-419. GPMC Clocks... 3456
Table 15-420. gpmc_clk Configuration.. 3456
Table 15-421. GPMC Local Power-Management Features...3456
Table 15-422. GPMC Interrupt Events...3457
Table 15-423. Boot Control Interface Input Signals Description.. 3459
Table 15-424. Idle Cycle Insertion Configuration...3470
Table 15-425. Chip-Select Configuration for NAND Interfacing...3499

Table 15-426. ECC Enable Settings..3507
Table 15-427. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits)...3512
Table 15-428. Aligned Message Byte Mapping in 8-Bit NAND...3513
Table 15-429. Aligned Message Byte Mapping in 16-Bit NAND...3513
Table 15-430. Aligned Nibble Mapping of Message in 8-Bit NAND..3513
Table 15-431. Misaligned Nibble Mapping of Message in 8-Bit NAND...3513
Table 15-432. Aligned Nibble Mapping of Message in 16-Bit NAND..3514
Table 15-433. Misaligned Nibble Mapping of Message in 16-Bit NAND (1 Unused Nibble).......................3514
Table 15-434. Misaligned Nibble Mapping of Message in 16-Bit NAND (2 Unused Nibbles).....................3514
Table 15-435. Misaligned Nibble Mapping of Message in 16-Bit NAND (3 Unused Nibbles).....................3514
Table 15-436. Prefetch Mode Configuration...3525
Table 15-437. Write-Posting Mode Configuration...3526
Table 15-438. GPMC Initialization..3532
Table 15-439. GPMC Configuration in NOR Mode..3532
Table 15-440. GPMC Configuration in NAND Mode..3532
Table 15-441. Reset GPMC...3532
Table 15-442. NOR Memory Type..3532
Table 15-443. NOR Chip-Select Configuration..3533
Table 15-444. NOR Timings Configuration...3533
Table 15-445. Wait Pin Configuration...3533
Table 15-446. Enable Chip-Select..3533
Table 15-447. NAND Memory Type..3533
Table 15-448. NAND Chip-Select Configuration..3533
Table 15-449. Asynchronous Read and Write Operations...3533
Table 15-450. ECC Engine...3534
Table 15-451. Prefetch and Write-Posting Engine...3534
Table 15-452. Wait Pin Configuration...3534
Table 15-453. Enable Chip-Select..3534
Table 15-454. Mode Parameters Check List...3535
Table 15-455. Access Type Parameters Check List..3535
Table 15-456. Timing Parameters..3536
Table 15-457. NAND Formulas Description..3538
Table 15-458. Synchronous NOR Formulas Description..3539
Table 15-459. Asynchronous NOR Formulas Description..3545
Table 15-460. GPMC Signals...3548
Table 15-461. Useful Timing Parameters on the Memory Side..3549
Table 15-462. Calculating GPMC Timing Parameters..3550
Table 15-463. AC Characteristics for Asynchronous Read Access...3551
Table 15-464. GPMC Timing Parameters for Asynchronous Read Access..3552
Table 15-465. AC Characteristics for Asynchronous Single Write (Memory Side).....................................3552
Table 15-466. GPMC Timing Parameters for Asynchronous Single Write..3553
Table 15-467. Supported Memories Interfaces...3554
Table 15-468. NAND Interface Bus Operations Summary..3555
Table 15-469. NOR Interface Bus Operations Summary..3556
Table 15-470. GPMC Instance Summary..3558
Table 15-471. GPMC Registers Mapping Summary...3558
Table 15-472. GPMC_REVISION...3559
Table 15-473. GPMC_SYSCONFIG..3559
Table 15-474. GPMC_SYSSTATUS...3559
Table 15-475. GPMC_IRQSTATUS..3560
Table 15-476. GPMC_IRQENABLE..3561
Table 15-477. GPMC_TIMEOUT_CONTROL..3562
Table 15-478. GPMC_ERR_ADDRESS..3562
Table 15-479. GPMC_ERR_TYPE...3562
Table 15-480. GPMC_CONFIG..3563
Table 15-481. GPMC_STATUS..3564
Table 15-482. GPMC_CONFIG1_i...3564
Table 15-483. GPMC_CONFIG2_i...3567
Table 15-484. GPMC_CONFIG3_i...3567
Table 15-485. GPMC_CONFIG4_i...3568
Table 15-486. GPMC_CONFIG5_i...3569

Table 15-487. GPMC_CONFIG6_i....3570
Table 15-488. GPMC_CONFIG7_i....3571
Table 15-489. GPMC_NAND_COMMAND_i....3572
Table 15-490. GPMC_NAND_ADDRESS_i....3572
Table 15-491. GPMC_NAND_DATA_i....3572
Table 15-492. GPMC_PREFETCH_CONFIG1....3572
Table 15-493. GPMC_PREFETCH_CONFIG2....3574
Table 15-494. GPMC_PREFETCH_CONTROL....3574
Table 15-495. GPMC_PREFETCH_STATUS....3575
Table 15-496. GPMC_ECC_CONFIG....3576
Table 15-497. GPMC_ECC_CONTROL....3576
Table 15-498. GPMC_ECC_SIZE_CONFIG....3577
Table 15-499. GPMC_ECCj_RESULT....3578
Table 15-500. GPMC_BCH_RESULT0_i....3579
Table 15-501. GPMC_BCH_RESULT1_i....3580
Table 15-502. GPMC_BCH_RESULT2_i....3580
Table 15-503. GPMC_BCH_RESULT3_i....3580
Table 15-504. GPMC_BCH_RESULT4_i....3580
Table 15-505. GPMC_BCH_RESULT5_i....3580
Table 15-506. GPMC_BCH_RESULT6_i....3581
Table 15-507. GPMC_BCH_SWDATA....3581
Table 15-508. ELM Integration Attributes....3583
Table 15-509. ELM Clocks and Resets....3584
Table 15-510. ELM Hardware Requests....3584
Table 15-511. Local Power-Management Features....3585
Table 15-512. Events....3585
Table 15-513. ELM_LOCATION_STATUS_i Value Decoding....3586
Table 15-514. ELM Processing Initialization....3587
Table 15-515. ELM Processing Completion for Continuous Mode....3588
Table 15-516. ELM Processing Completion for Page Mode....3588
Table 15-517. Use Case: Continuous Mode....3589
Table 15-518. 16-Bit NAND Sector Buffer Address Map....3590
Table 15-519. Use Case: Page Mode....3590
Table 15-520. ELM Instance Summary....3593
Table 15-521. ELM Registers Mapping Summary....3593
Table 15-522. ELM_REVISION....3594
Table 15-523. ELM_SYSCONFIG....3594
Table 15-524. ELM_SYSSTATUS....3595
Table 15-525. ELM_IRQSTATUS....3595
Table 15-526. ELM_IRQENABLE....3596
Table 15-527. ELM_LOCATION_CONFIG....3597
Table 15-528. ELM_PAGE_CTRL....3597
Table 15-529. ELM_SYNDROME_FRAGMENT_0_i....3598
Table 15-530. ELM_SYNDROME_FRAGMENT_1_i....3598
Table 15-531. ELM_SYNDROME_FRAGMENT_2_i....3598
Table 15-532. ELM_SYNDROME_FRAGMENT_3_i....3598
Table 15-533. ELM_SYNDROME_FRAGMENT_4_i....3599
Table 15-534. ELM_SYNDROME_FRAGMENT_5_i....3599
Table 15-535. ELM_SYNDROME_FRAGMENT_6_i....3599
Table 15-536. ELM_LOCATION_STATUS_i....3599
Table 15-537. ELM_ERROR_LOCATION_0_i....3600
Table 15-538. ELM_ERROR_LOCATION_1_i....3600
Table 15-539. ELM_ERROR_LOCATION_2_i....3600
Table 15-540. ELM_ERROR_LOCATION_3_i....3601
Table 15-541. ELM_ERROR_LOCATION_4_i....3601
Table 15-542. ELM_ERROR_LOCATION_5_i....3601
Table 15-543. ELM_ERROR_LOCATION_6_i....3601
Table 15-544. ELM_ERROR_LOCATION_7_i....3602
Table 15-545. ELM_ERROR_LOCATION_8_i....3602
Table 15-546. ELM_ERROR_LOCATION_9_i....3602
Table 15-547. ELM_ERROR_LOCATION_10_i....3602

Table 15-548. ELM_ERROR_LOCATION_11_i....3603
Table 15-549. ELM_ERROR_LOCATION_12_i....3603
Table 15-550. ELM_ERROR_LOCATION_13_i....3603
Table 15-551. ELM_ERROR_LOCATION_14_i....3604
Table 15-552. ELM_ERROR_LOCATION_15_i....3604
Table 15-553. OCM Subsystem Integration Attributes....3606
Table 15-554. OCM Subsystem Clocks and Resets....3606
Table 15-555. OCM Subsystem Hardware Requests....3607
Table 15-556. OCM Subsystem Events....3610
Table 15-557. OCMC Error Handling In Case Of Different Error Types....3616
Table 15-558. OCMC ECC Configuration....3616
Table 15-559. OCM Subsystem Instance Summary....3622
Table 15-560. OCM Subsystem Registers Mapping Summary....3622
Table 15-561. OCMC_ECC_PID....3623
Table 15-562. Register Call Summary for Register OCMC_ECC_PID....3624
Table 15-563. OCMC_SYSCONFIG_PM....3624
Table 15-564. Register Call Summary for Register OCMC_SYSCONFIG_PM....3625
Table 15-565. OCMC_SYSCONFIG_RST....3625
Table 15-566. Register Call Summary for Register OCMC_SYSCONFIG_RST....3625
Table 15-567. OCMC_MEM_SIZE_READ....3625
Table 15-568. Register Call Summary for Register OCMC_MEM_SIZE_READ....3626
Table 15-569. INTR0_STATUS_RAW_SET....3626
Table 15-570. Register Call Summary for Register INTR0_STATUS_RAW_SET....3627
Table 15-571. INTR0_STATUS_ENABLED_CLEAR....3627
Table 15-572. Register Call Summary for Register INTR0_STATUS_ENABLED_CLEAR....3628
Table 15-573. INTR0_ENABLE_SET....3629
Table 15-574. Register Call Summary for Register INTR0_ENABLE_SET....3630
Table 15-575. INTR0_ENABLE_CLEAR....3630
Table 15-576. Register Call Summary for Register INTR0_ENABLE_CLEAR....3631
Table 15-577. OCMC_INTR0_EOI....3631
Table 15-578. Register Call Summary for Register OCMC_INTR0_EOI....3631
Table 15-579. INTR1_STATUS_RAW_SET....3631
Table 15-580. Register Call Summary for Register INTR1_STATUS_RAW_SET....3633
Table 15-581. INTR1_STATUS_ENABLED_CLEAR....3633
Table 15-582. Register Call Summary for Register INTR1_STATUS_ENABLED_CLEAR....3634
Table 15-583. INTR1_ENABLE_SET....3634
Table 15-584. Register Call Summary for Register INTR1_ENABLE_SET....3635
Table 15-585. INTR1_ENABLE_CLEAR....3635
Table 15-586. Register Call Summary for Register INTR1_ENABLE_CLEAR....3636
Table 15-587. OCMC_INTR1_EOI....3637
Table 15-588. Register Call Summary for Register OCMC_INTR1_EOI....3637
Table 15-589. CFG_OCMC_ECC....3637
Table 15-590. Register Call Summary for Register CFG_OCMC_ECC....3638
Table 15-591. CFG_OCMC_ECC_MEM_BLK....3638
Table 15-592. Register Call Summary for Register CFG_OCMC_ECC_MEM_BLK....3639
Table 15-593. CFG_OCMC_ECC_ERROR....3639
Table 15-594. Register Call Summary for Register CFG_OCMC_ECC_ERROR....3639
Table 15-595. CFG_OCMC_ECC_CLEAR_HIST....3639
Table 15-596. Register Call Summary for Register CFG_OCMC_ECC_CLEAR_HIST....3640
Table 15-597. STATUS_ERROR_CNT....3640
Table 15-598. Register Call Summary for Register STATUS_ERROR_CNT....3641
Table 15-599. STATUS_SEC_ERROR_TRACE....3641
Table 15-600. Register Call Summary for Register STATUS_SEC_ERROR_TRACE....3641
Table 15-601. STATUS_DED_ERROR_TRACE....3642
Table 15-602. Register Call Summary for Register STATUS_DED_ERROR_TRACE....3642
Table 15-603. STATUS_ADDR_TRANSLATION_ERROR_TRACE....3642
Table 15-604. Register Call Summary for Register STATUS_ADDR_TRANSLATION_ERROR_TRACE....3642
Table 15-605. STATUS_SEC_ERROR_DISTR_0....3642
Table 15-606. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_0....3643
Table 15-607. STATUS_SEC_ERROR_DISTR_1....3643
Table 15-608. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_1....3643

Table 15-609. STATUS_SEC_ERROR_DISTR_2 3643
Table 15-610. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_2 3644
Table 15-611. STATUS_SEC_ERROR_DISTR_3 3644
Table 15-612. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_3 3644
Table 15-613. STATUS_SEC_ERROR_DISTR_4 3644
Table 15-614. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_4 3645
Table 15-615. CFG_OCMC_CBUF_EN 3645
Table 15-616. Register Call Summary for Register CFG_OCMC_CBUF_EN 3646
Table 15-617. CFG_OCMC_CBUF_RESET 3646
Table 15-618. Register Call Summary for Register CFG_OCMC_CBUF_RESET 3647
Table 15-619. CFG_OCMC_CBUF_ERR_HANDLER 3647
Table 15-620. Register Call Summary for Register CFG_OCMC_CBUF_ERR_HANDLER 3649
Table 15-621. STATUS_CBUF_WR_OUT_OF_RANGE_ERR 3649
Table 15-622. Register Call Summary for Register STATUS_CBUF_WR_OUT_OF_RANGE_ERR 3649
Table 15-623. STATUS_CBUF_WR_VBUF_START_ERR 3649
Table 15-624. Register Call Summary for Register STATUS_CBUF_WR_VBUF_START_ERR 3650
Table 15-625. STATUS_CBUF_WR_ADDR_SEQ_ERROR 3650
Table 15-626. Register Call Summary for Register STATUS_CBUF_WR_ADDR_SEQ_ERROR 3650
Table 15-627. STATUS_CBUF_RD_OUT_OF_RANGE_ERROR 3650
Table 15-628. Register Call Summary for Register STATUS_CBUF_RD_OUT_OF_RANGE_ERROR 3651
Table 15-629. STATUS_CBUF_VBUF_RD_START_ERROR 3651
Table 15-630. Register Call Summary for Register STATUS_CBUF_VBUF_RD_START_ERROR 3651
Table 15-631. STATUS_CBUF_RD_ADDR_SEQ_ERROR 3651
Table 15-632. Register Call Summary for Register STATUS_CBUF_RD_ADDR_SEQ_ERROR 3652
Table 15-633. STATUS_CBUF_OVERFLOW_MID 3652
Table 15-634. Register Call Summary for Register STATUS_CBUF_OVERFLOW_MID 3652
Table 15-635. STATUS_CBUF_OVERFLOW_WRAP 3652
Table 15-636. Register Call Summary for Register STATUS_CBUF_OVERFLOW_WRAP 3653
Table 15-637. STATUS_CBUF_UNDERFLOW 3653
Table 15-638. Register Call Summary for Register STATUS_CBUF_UNDERFLOW 3653
Table 15-639. STATUS_CBUF_SHORT_FRAME_DETECT 3653
Table 15-640. Register Call Summary for Register STATUS_CBUF_SHORT_FRAME_DETECT 3654
Table 15-641. CBUF_i_VBUF_START_ADDR 3654
Table 15-642. Register Call Summary for Register CBUF_i_VBUF_START_ADDR 3654
Table 15-643. CBUF_i_VBUF_END_ADDR 3654
Table 15-644. Register Call Summary for Register CBUF_i_VBUF_END_ADDR 3655
Table 15-645. CBUF_i_OCMC_START_ADDR 3655
Table 15-646. Register Call Summary for Register CBUF_i_OCMC_START_ADDR 3655
Table 15-647. CBUF_i_OCMC_BUF_SIZE 3655
Table 15-648. Register Call Summary for Register CBUF_i_OCMC_BUF_SIZE 3655
Table 15-649. CBUF_k_LAST_WR_ADDR 3655
Table 15-650. Register Call Summary for Register CBUF_k_LAST_WR_ADDR 3656
Table 15-651. CBUF_k_LAST_RD_ADDR 3656
Table 15-652. Register Call Summary for Register CBUF_k_LAST_RD_ADDR 3656
Table 15-653. LAST_ILLEGAL_OCMC_ADDR 3656
Table 15-654. Register Call Summary for Register LAST_ILLEGAL_OCMC_ADDR 3657
Table 16-1. External DMA_SYSTEM Request Signals 3661
Table 16-2. DMA_SYSTEM Integration Attributes 3664
Table 16-3. DMA_SYSTEM Clocks and Resets 3664
Table 16-4. DMA_SYSTEM Hardware Requests 3665
Table 16-5. DMA_SYSTEM Default Request Mapping 3667
Table 16-6. Connection of The Device DREQs to The DMA_CROSSBAR Inputs 3673
Table 16-7. Local Power-Management Features 3678
Table 16-8. Clock Activity Settings 3678
Table 16-9. Logical DMA Channel Events 3679
Table 16-10. Parameter Values for Addressing Mode Examples (a), (b), and (c) 3684
Table 16-11. Equations for Rotation 3684
Table 16-12. Example Parameter Values for a 90-Degree Clockwise Image Rotation 3686
Table 16-13. Buffering Disable 3689
Table 16-14. Type 1 3696
Table 16-15. Type 2 With Source and Destination Address Updates 3696

Table 16-16. Type 2 With Source or Destination Address Update....3697
Table 16-17. Type 3 With Source and Destination Address Updates....3697
Table 16-18. Type 3 With Source or Destination Address Update....3697
Table 16-19. DMA_SYSTEM Instance Summary....3709
Table 16-20. DMA_SYSTEM Registers Mapping Summary....3709
Table 16-21. DMA4_REVISION....3710
Table 16-22. DMA4_IRQSTATUS_Lj....3710
Table 16-23. DMA4_IRQENABLE_Lj....3710
Table 16-24. DMA4_SYSSTATUS....3711
Table 16-25. DMA4_OCP_SYSCONFIG....3711
Table 16-26. DMA4_CAPS_0....3712
Table 16-27. DMA4_CAPS_2....3713
Table 16-28. DMA4_CAPS_3....3714
Table 16-29. DMA4_CAPS_4....3715
Table 16-30. DMA4_GCR....3717
Table 16-31. DMA4_CCRi....3718
Table 16-32. DMA4_CLNK_CTRLi....3721
Table 16-33. DMA4_CICRi....3721
Table 16-34. DMA4_CSRi....3722
Table 16-35. DMA4_CSDPi....3724
Table 16-36. DMA4_CENi....3726
Table 16-37. DMA4_CFNi....3726
Table 16-38. DMA4_CSSAi....3726
Table 16-39. DMA4_CDSAi....3726
Table 16-40. DMA4_CSEIi....3727
Table 16-41. DMA4_CSFIi....3727
Table 16-42. DMA4_CDEIi....3727
Table 16-43. DMA4_CDFIi....3728
Table 16-44. DMA4_CSACi....3728
Table 16-45. DMA4_CDACi....3728
Table 16-46. DMA4_CCENi....3728
Table 16-47. DMA4_CCFNi....3729
Table 16-48. DMA4_COLORi....3729
Table 16-49. DMA4_CDPi....3729
Table 16-50. DMA4_CNDPi....3730
Table 16-51. DMA4_CCDNi....3731
Table 16-52. EDMA Channel Controllers Configuration....3734
Table 16-53. EDMA Transfer Controllers Configuration....3735
Table 16-54. External EDMA Request Signals....3736
Table 16-55. EDMA Integration Attributes....3739
Table 16-56. EDMA Clocks and Resets....3739
Table 16-57. EDMA Hardware Requests....3739
Table 16-58. EDMA Default Request Mapping....3741
Table 16-59. EDMA Parameter RAM Contents....3751
Table 16-60. EDMA Channel Parameter Description....3753
Table 16-61. Dummy and Null Transfer Request....3756
Table 16-62. Parameter Updates in EDMA_TPCC (for Non-Null, Non-Dummy PaRAM Set)....3758
Table 16-63. Expected Number of Transfers for Non-Null Transfer....3765
Table 16-64. Shadow Region Registers....3769
Table 16-65. Chain Event Triggers....3771
Table 16-66. EDMA Transfer Completion Interrupts....3772
Table 16-67. EDMA Error Interrupts....3772
Table 16-68. Transfer Complete Code (TCC) to EDMA_TPCC Interrupt Mapping....3773
Table 16-69. Number of Interrupts....3773
Table 16-70. Allowed Accesses....3781
Table 16-71. MPPA Registers to Region Assignment....3781
Table 16-72. Example Access Denied....3782
Table 16-73. Example Access Allowed....3784
Table 16-74. Read/Write Command Optimization Rules....3788
Table 16-75. EDMA Transfer Controller Configurations....3790
Table 16-76. Debug Checklist....3816

Table 16-77. EDMA Instance Summary....3819
Table 16-78. DSP Private Access EDMA Instance Summary....3819
Table 16-79. EVE EDMA Instance Summary (Private Access)....3819
Table 16-80. System EDMA_TPCC Registers Mapping Summary....3820
Table 16-81. DSP1 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)....3822
Table 16-82. DSP2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)....3825
Table 16-83. DSP EDMA_TPCC Registers Mapping Summary (Private Access)....3828
Table 16-84. EVE1 and EVE2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)....3830
Table 16-85. System EDMA TPTC0 and EDMA TPTC1 Registers Mapping Summary....3834
Table 16-86. DSP1 EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)....3835
Table 16-87. DSP2 EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)....3835
Table 16-88. DSP EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (Private Access)....3836
Table 16-89. EVE1 and EVE2 EDMA_TPTC0 Registers Mapping Summary (L3_MAIN Access)....3837
Table 16-90. EVE1 and EVE2 EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)....3838
Table 16-91. EDMA_TPCC_PID....3839
Table 16-92. EDMA_TPCC_CCCFG....3840
Table 16-93. EDMA_TPCC_CLKGDIS....3841
Table 16-94. EDMA_TPCC_DCHMAPN_m....3842
Table 16-95. EDMA_TPCC_QCHMAPN_j....3842
Table 16-96. EDMA_TPCC_DMAQNUMN_k....3842
Table 16-97. EDMA_TPCC_QDMAQNUM....3843
Table 16-98. EDMA_TPCC_QUETCMAP....3844
Table 16-99. EDMA_TPCC_QUEPRI....3844
Table 16-100. EDMA_TPCC_EMR....3845
Table 16-101. EDMA_TPCC_EMRH....3846
Table 16-102. EDMA_TPCC_EMCR....3847
Table 16-103. EDMA_TPCC_EMCRH....3848
Table 16-104. EDMA_TPCC_QEMR....3849
Table 16-105. EDMA_TPCC_QEMCR....3849
Table 16-106. EDMA_TPCC_CCERR....3850
Table 16-107. EDMA_TPCC_CCERRCLR....3852
Table 16-108. EDMA_TPCC_EEVAL....3853
Table 16-109. EDMA_TPCC_DRAEM_k....3854
Table 16-110. EDMA_TPCC_DRAEHM_k....3855
Table 16-111. EDMA_TPCC_QRAEN_k....3856
Table 16-112. EDMA_TPCC_Q0E_p....3856
Table 16-113. EDMA_TPCC_Q1E_p....3857
Table 16-114. EDMA_TPCC_QSTATN_i....3857
Table 16-115. EDMA_TPCC_QWMTHRA....3858
Table 16-116. EDMA_TPCC_QWMTHRB....3859
Table 16-117. EDMA_TPCC_CCSTAT....3859
Table 16-118. EDMA_TPCC_AETCTL....3861
Table 16-119. EDMA_TPCC_AETSTAT....3862
Table 16-120. EDMA_TPCC_AETCMD....3862
Table 16-121. EDMA_TPCC_MPFAR....3863
Table 16-122. EDMA_TPCC_MPFSR....3863
Table 16-123. EDMA_TPCC_MPFCR....3864
Table 16-124. EDMA_TPCC_MPPAG....3865
Table 16-125. EDMA_TPCC_MPPAN_k....3866
Table 16-126. EDMA_TPCC_ER....3868
Table 16-127. EDMA_TPCC_ERH....3869
Table 16-128. EDMA_TPCC_ECR....3870
Table 16-129. EDMA_TPCC_ECRH....3871
Table 16-130. EDMA_TPCC_ESR....3872
Table 16-131. EDMA_TPCC_ESRH....3873
Table 16-132. EDMA_TPCC_CER....3874
Table 16-133. EDMA_TPCC_CERH....3875
Table 16-134. EDMA_TPCC_EER....3876
Table 16-135. EDMA_TPCC_EERH....3877
Table 16-136. EDMA_TPCC_EECR....3879
Table 16-137. EDMA_TPCC_EECRH....3880

Table 16-138. EDMA_TPCC_EESR.......3881
Table 16-139. EDMA_TPCC_EESRH.......3882
Table 16-140. EDMA_TPCC_SER.......3883
Table 16-141. EDMA_TPCC_SERH.......3884
Table 16-142. EDMA_TPCC_SECR.......3885
Table 16-143. EDMA_TPCC_SECRH.......3886
Table 16-144. EDMA_TPCC_IER.......3887
Table 16-145. EDMA_TPCC_IERH.......3888
Table 16-146. EDMA_TPCC_IECR.......3889
Table 16-147. EDMA_TPCC_IECRH.......3890
Table 16-148. EDMA_TPCC_IESR.......3891
Table 16-149. EDMA_TPCC_IESRH.......3892
Table 16-150. EDMA_TPCC_IPR.......3893
Table 16-151. EDMA_TPCC_IPRH.......3894
Table 16-152. EDMA_TPCC_ICR.......3895
Table 16-153. EDMA_TPCC_ICRH.......3896
Table 16-154. EDMA_TPCC_IEVAL.......3897
Table 16-155. EDMA_TPCC_QER.......3898
Table 16-156. EDMA_TPCC_QEER.......3898
Table 16-157. EDMA_TPCC_QEECR.......3899
Table 16-158. EDMA_TPCC_QEESR.......3900
Table 16-159. EDMA_TPCC_QSER.......3900
Table 16-160. EDMA_TPCC_QSECR.......3901
Table 16-161. EDMA_TPCC_ER_RN_k.......3901
Table 16-162. EDMA_TPCC_ERH_RN_k.......3902
Table 16-163. EDMA_TPCC_ECR_RN_k.......3904
Table 16-164. EDMA_TPCC_ECRH_RN_k.......3905
Table 16-165. EDMA_TPCC_ESR_RN_k.......3906
Table 16-166. EDMA_TPCC_ESRH_RN_k.......3907
Table 16-167. EDMA_TPCC_CER_RN_k.......3908
Table 16-168. EDMA_TPCC_CERH_RN_k.......3909
Table 16-169. EDMA_TPCC_EER_RN_k.......3910
Table 16-170. EDMA_TPCC_EERH_RN_k.......3911
Table 16-171. EDMA_TPCC_EECR_RN_k.......3912
Table 16-172. EDMA_TPCC_EECRH_RN_k.......3913
Table 16-173. EDMA_TPCC_EESR_RN_k.......3914
Table 16-174. EDMA_TPCC_EESRH_RN_k.......3915
Table 16-175. EDMA_TPCC_SER_RN_k.......3916
Table 16-176. EDMA_TPCC_SERH_RN_k.......3917
Table 16-177. EDMA_TPCC_SECR_RN_k.......3918
Table 16-178. EDMA_TPCC_SECRH_RN_k.......3919
Table 16-179. EDMA_TPCC_IER_RN_k.......3921
Table 16-180. EDMA_TPCC_IERH_RN_k.......3922
Table 16-181. EDMA_TPCC_IECR_RN_k.......3923
Table 16-182. EDMA_TPCC_IECRH_RN_k.......3924
Table 16-183. EDMA_TPCC_IESR_RN_k.......3925
Table 16-184. EDMA_TPCC_IESRH_RN_k.......3926
Table 16-185. EDMA_TPCC_IPR_RN_k.......3927
Table 16-186. EDMA_TPCC_IPRH_RN_k.......3928
Table 16-187. EDMA_TPCC_ICR_RN_k.......3929
Table 16-188. EDMA_TPCC_ICRH_RN_k.......3930
Table 16-189. EDMA_TPCC_IEVAL_RN_k.......3931
Table 16-190. EDMA_TPCC_QER_RN_k.......3931
Table 16-191. EDMA_TPCC_QEER_RN_k.......3932
Table 16-192. EDMA_TPCC_QEECR_RN_k.......3933
Table 16-193. EDMA_TPCC_QEESR_RN_k.......3933
Table 16-194. EDMA_TPCC_QSER_RN_k.......3934
Table 16-195. EDMA_TPCC_QSECR_RN_k.......3934
Table 16-196. EDMA_TPCC_OPT_n.......3935
Table 16-197. EDMA_TPCC_SRC_n.......3937
Table 16-198. EDMA_TPCC_ABCNT_n.......3937

Table 16-199. EDMA_TPCC_DST_n....3938
Table 16-200. EDMA_TPCC_BIDX_n....3939
Table 16-201. EDMA_TPCC_LNK_n....3939
Table 16-202. EDMA_TPCC_CIDX_n....3940
Table 16-203. EDMA_TPCC_CCNT_n....3941
Table 16-204. EDMA_TPTCn_PID....3941
Table 16-205. EDMA_TPTCn_TCCFG....3942
Table 16-206. EDMA_TPTCn_TCSTAT....3943
Table 16-207. EDMA_TPTCn_INTSTAT....3944
Table 16-208. EDMA_TPTCn_INTEN....3945
Table 16-209. EDMA_TPTCn_INTCLR....3945
Table 16-210. EDMA_TPTCn_INTCMD....3946
Table 16-211. EDMA_TPTCn_ERRSTAT....3946
Table 16-212. EDMA_TPTCn_ERREN....3947
Table 16-213. EDMA_TPTCn_ERRCLR....3948
Table 16-214. EDMA_TPTCn_ERRDET....3949
Table 16-215. EDMA_TPTCn_ERRCMD....3950
Table 16-216. EDMA_TPTCn_RDRATE....3950
Table 16-217. EDMA_TPTCn_POPT....3951
Table 16-218. EDMA_TPTCn_PSRC....3952
Table 16-219. EDMA_TPTCn_PCNT....3952
Table 16-220. EDMA_TPTCn_PDST....3953
Table 16-221. EDMA_TPTCn_PBIDX....3953
Table 16-222. EDMA_TPTCn_PMPPRXY....3954
Table 16-223. EDMA_TPTCn_SAOPT....3955
Table 16-224. EDMA_TPTCn_SASRC....3956
Table 16-225. EDMA_TPTCn_SACNT....3957
Table 16-226. EDMA_TPTCn_SADST....3957
Table 16-227. EDMA_TPTCn_SABIDX....3958
Table 16-228. EDMA_TPTCn_SAMPPRXY....3958
Table 16-229. EDMA_TPTCn_SACNTRLD....3959
Table 16-230. EDMA_TPTCn_SASRCBREF....3960
Table 16-231. EDMA_TPTCn_SADSTBREF....3960
Table 16-232. EDMA_TPTCn_DFCNTRLD....3961
Table 16-233. EDMA_TPTCn_DFSRCBREF....3961
Table 16-234. EDMA_TPTCn_DFDSTBREF....3962
Table 16-235. EDMA_TPTCn_DFOPTi....3962
Table 16-236. EDMA_TPTCn_DFSRCi....3963
Table 16-237. EDMA_TPTCn_DFCNTi....3964
Table 16-238. EDMA_TPTCn_DFDSTi....3964
Table 16-239. EDMA_TPTCn_DFBIDXi....3965
Table 16-240. EDMA_TPTCn_DFMPPRXYi....3965
Table 17-1. Interrupts From External Devices....3970
Table 17-2. MPU_INTC Default Interrupt Mapping....3972
Table 17-3. DSP1_INTC Default Interrupt Mapping....3981
Table 17-4. DSP2_INTC Default Interrupt Mapping....3987
Table 17-5. IPU1_Cx_INTC Default Interrupt Mapping....3993
Table 17-6. IPU2_Cx_INTC Default Interrupt Mapping....3997
Table 17-7. EVE1_INTC1 Default Interrupt Mapping....4001
Table 17-8. EVE2_INTC1 Default Interrupt Mapping....4003
Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs....4005
Table 18-1. Control Module I/O Description....4018
Table 18-2. Control Module Integration Attributes....4019
Table 18-3. Control Module Clocks and Resets....4019
Table 18-4. Control Module Hardware Requests....4020
Table 18-5. Description Of The Pad Configuration Register Bits....4022
Table 18-6. Pull Selection....4024
Table 18-7. Thermal Management Signals Description....4028
Table 18-8. FIFOs Generic Description....4032
Table 18-9. Summary of the Thermal Management Related Registers....4033
Table 18-10. ADC Values Versus Temperature....4034

Table 18-11. Control Bits For the PBIAS and MMC1 I/O Cells....4038
Table 18-12. PBIAS Cell Voltage Configuration....4039
Table 18-13. PBIAS Cell Error Signal Truth Table....4040
Table 18-14. Generic Description of the CTRL_CORE_X_IRQ_B_A IRQ_CROSSBAR Control Registers....4040
Table 18-15. Interrupt Lines Associated With The IRQ_CROSSBAR Control Registers....4043
Table 18-16. Generic Description of the CTRL_CORE_DMA_X_DREQ_B_A DMA_CROSSBAR Control Registers....4043
Table 18-17. DREQ Lines Associated With The DMA_CROSSBAR Control Registers....4046
Table 18-18. Memory Region Lock Registers....4047
Table 18-19. Output Impedance Controls - I[2:0]....4048
Table 18-20. Slew Rate Controls - SR[2:0]....4048
Table 18-21. Weak Driver Controls - WD[1:0]....4048
Table 18-22. Software Group Controls for the DDR2/DDR3 Pads....4049
Table 18-23. Vref Cell Load Current Selection....4053
Table 18-24. Vref Cell Coupling Capacitor Selection....4053
Table 18-25. Controls for the Vref-Generation Cells Versus DDR2/DDR3 Receiver Pads....4053
Table 18-26. Registers Associated With AVS Class 0 Voltage....4055
Table 18-27. CONTROL MODULE Instance Summary....4059
Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary....4059
Table 18-29. CTRL_CORE_STATUS....4078
Table 18-30. CTRL_CORE_SEC_ERR_STATUS_FUNC_1....4079
Table 18-31. CTRL_CORE_SEC_ERR_STATUS_DEBUG_1....4080
Table 18-32. CTRL_CORE_MPU_FORCEWRNP....4082
Table 18-33. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_0....4082
Table 18-34. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_1....4083
Table 18-35. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_2....4083
Table 18-36. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_3....4083
Table 18-37. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_4....4083
Table 18-38. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_5....4084
Table 18-39. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_0....4084
Table 18-40. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_1....4084
Table 18-41. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_2....4084
Table 18-42. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_3....4085
Table 18-43. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_4....4085
Table 18-44. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_5....4085
Table 18-45. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_6....4086
Table 18-46. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_7....4086
Table 18-47. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_0....4086
Table 18-48. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_1....4086
Table 18-49. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_2....4087
Table 18-50. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_3....4087
Table 18-51. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_4....4087
Table 18-52. CTRL_CORE_STD_FUSE_OPP_BGAP_GPU....4087
Table 18-53. CTRL_CORE_STD_FUSE_OPP_BGAP_MPU....4088
Table 18-54. CTRL_CORE_STD_FUSE_OPP_BGAP_CORE....4088
Table 18-55. CTRL_CORE_STD_FUSE_OPP_BGAP_MPU23....4089
Table 18-56. CTRL_CORE_STD_FUSE_MPK_0....4089
Table 18-57. CTRL_CORE_STD_FUSE_MPK_1....4089
Table 18-58. CTRL_CORE_STD_FUSE_MPK_2....4089
Table 18-59. CTRL_CORE_STD_FUSE_MPK_3....4090
Table 18-60. CTRL_CORE_STD_FUSE_MPK_4....4090
Table 18-61. CTRL_CORE_STD_FUSE_MPK_5....4090
Table 18-62. CTRL_CORE_STD_FUSE_MPK_6....4090
Table 18-63. CTRL_CORE_STD_FUSE_MPK_7....4091
Table 18-64. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_0....4091
Table 18-65. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_1....4091
Table 18-66. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_2....4091
Table 18-67. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_3....4092
Table 18-68. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_4....4092
Table 18-69. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_5....4092
Table 18-70. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_0....4093
Table 18-71. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_1....4093

Table 18-72. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_2....4093
Table 18-73. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_3....4093
Table 18-74. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_4....4094
Table 18-75. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_5....4094
Table 18-76. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_6....4094
Table 18-77. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_7....4094
Table 18-78. CTRL_CORE_CUST_FUSE_SWRV_0....4095
Table 18-79. CTRL_CORE_CUST_FUSE_SWRV_1....4095
Table 18-80. CTRL_CORE_CUST_FUSE_SWRV_2....4095
Table 18-81. CTRL_CORE_CUST_FUSE_SWRV_3....4095
Table 18-82. CTRL_CORE_CUST_FUSE_SWRV_4....4096
Table 18-83. CTRL_CORE_CUST_FUSE_SWRV_5....4096
Table 18-84. CTRL_CORE_CUST_FUSE_SWRV_6....4096
Table 18-85. CTRL_CORE_DEV_CONF....4096
Table 18-86. CTRL_CORE_TEMP_SENSOR_MPU....4097
Table 18-87. CTRL_CORE_TEMP_SENSOR_GPU....4097
Table 18-88. CTRL_CORE_TEMP_SENSOR_CORE....4098
Table 18-89. CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR....4098
Table 18-90. CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR....4099
Table 18-91. CTRL_CORE_HWOBS_CONTROL....4099
Table 18-92. CTRL_CORE_PHY_POWER_USB....4100
Table 18-93. CTRL_CORE_PHY_POWER_SATA....4101
Table 18-94. CTRL_CORE_BANDGAP_MASK_1....4101
Table 18-95. CTRL_CORE_BANDGAP_THRESHOLD_MPU....4103
Table 18-96. CTRL_CORE_BANDGAP_THRESHOLD_GPU....4103
Table 18-97. CTRL_CORE_BANDGAP_THRESHOLD_CORE....4103
Table 18-98. CTRL_CORE_BANDGAP_TSHUT_MPU....4104
Table 18-99. CTRL_CORE_BANDGAP_TSHUT_GPU....4104
Table 18-100. CTRL_CORE_BANDGAP_TSHUT_CORE....4105
Table 18-101. CTRL_CORE_BANDGAP_STATUS_1....4106
Table 18-102. CTRL_CORE_SATA_EXT_MODE....4106
Table 18-103. CTRL_CORE_DTEMP_MPU_0....4107
Table 18-104. CTRL_CORE_DTEMP_MPU_1....4107
Table 18-105. CTRL_CORE_DTEMP_MPU_2....4107
Table 18-106. CTRL_CORE_DTEMP_MPU_3....4108
Table 18-107. CTRL_CORE_DTEMP_MPU_4....4108
Table 18-108. CTRL_CORE_DTEMP_GPU_0....4108
Table 18-109. CTRL_CORE_DTEMP_GPU_1....4108
Table 18-110. CTRL_CORE_DTEMP_GPU_2....4109
Table 18-111. CTRL_CORE_DTEMP_GPU_3....4109
Table 18-112. CTRL_CORE_DTEMP_GPU_4....4109
Table 18-113. CTRL_CORE_DTEMP_CORE_0....4110
Table 18-114. CTRL_CORE_DTEMP_CORE_1....4110
Table 18-115. CTRL_CORE_DTEMP_CORE_2....4110
Table 18-116. CTRL_CORE_DTEMP_CORE_3....4110
Table 18-117. CTRL_CORE_DTEMP_CORE_4....4111
Table 18-118. CTRL_CORE_SMA_SW_0....4111
Table 18-119. CTRL_CORE_SEC_ERR_STATUS_FUNC_2....4112
Table 18-120. CTRL_CORE_SEC_ERR_STATUS_DEBUG_2....4113
Table 18-121. CTRL_CORE_EMIF_INITIATOR_PRIORITY_1....4115
Table 18-122. CTRL_CORE_EMIF_INITIATOR_PRIORITY_2....4116
Table 18-123. CTRL_CORE_EMIF_INITIATOR_PRIORITY_3....4117
Table 18-124. CTRL_CORE_EMIF_INITIATOR_PRIORITY_4....4117
Table 18-125. CTRL_CORE_EMIF_INITIATOR_PRIORITY_5....4118
Table 18-126. CTRL_CORE_EMIF_INITIATOR_PRIORITY_6....4119
Table 18-127. CTRL_CORE_L3_INITIATOR_PRESSURE_1....4120
Table 18-128. CTRL_CORE_L3_INITIATOR_PRESSURE_2....4120
Table 18-129. CTRL_CORE_L3_INITIATOR_PRESSURE_4....4121
Table 18-130. CTRL_CORE_L3_INITIATOR_PRESSURE_5....4121
Table 18-131. CTRL_CORE_L3_INITIATOR_PRESSURE_6....4122
Table 18-132. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_0....4122

Table 18-133. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_1....4122
Table 18-134. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_2....4123
Table 18-135. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_3....4123
Table 18-136. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_4....4123
Table 18-137. CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL....4124
Table 18-138. CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL....4124
Table 18-139. CTRL_CORE_CUST_FUSE_UID_0....4125
Table 18-140. CTRL_CORE_CUST_FUSE_UID_1....4125
Table 18-141. CTRL_CORE_CUST_FUSE_UID_2....4125
Table 18-142. CTRL_CORE_CUST_FUSE_UID_3....4126
Table 18-143. CTRL_CORE_CUST_FUSE_UID_4....4126
Table 18-144. CTRL_CORE_CUST_FUSE_UID_5....4126
Table 18-145. CTRL_CORE_CUST_FUSE_UID_6....4126
Table 18-146. CTRL_CORE_CUST_FUSE_PCIE_ID_0....4127
Table 18-147. CTRL_CORE_CUST_FUSE_USB_ID_0....4127
Table 18-148. CTRL_CORE_MAC_ID_SW_0....4127
Table 18-149. CTRL_CORE_MAC_ID_SW_1....4127
Table 18-150. CTRL_CORE_MAC_ID_SW_2....4128
Table 18-151. CTRL_CORE_MAC_ID_SW_3....4128
Table 18-152. CTRL_CORE_SMA_SW_1....4128
Table 18-153. CTRL_CORE_DSS_PLL_CONTROL....4130
Table 18-154. CTRL_CORE_MMR_LOCK_1....4131
Table 18-155. CTRL_CORE_MMR_LOCK_2....4132
Table 18-156. CTRL_CORE_MMR_LOCK_3....4132
Table 18-157. CTRL_CORE_MMR_LOCK_4....4132
Table 18-158. CTRL_CORE_MMR_LOCK_5....4132
Table 18-159. CTRL_CORE_CONTROL_IO_1....4133
Table 18-160. CTRL_CORE_CONTROL_IO_2....4134
Table 18-161. CTRL_CORE_CONTROL_DSP1_RST_VECT....4135
Table 18-162. CTRL_CORE_CONTROL_DSP2_RST_VECT....4135
Table 18-163. CTRL_CORE_STD_FUSE_OPP_BGAP_DSPEVE....4135
Table 18-164. CTRL_CORE_STD_FUSE_OPP_BGAP_IVA....4136
Table 18-165. CTRL_CORE_LDOSRAM_DSPEVE_VOLTAGE_CTRL....4136
Table 18-166. CTRL_CORE_LDOSRAM_IVA_VOLTAGE_CTRL....4137
Table 18-167. CTRL_CORE_TEMP_SENSOR_DSPEVE....4138
Table 18-168. CTRL_CORE_TEMP_SENSOR_IVA....4139
Table 18-169. CTRL_CORE_BANDGAP_MASK_2....4139
Table 18-170. CTRL_CORE_BANDGAP_THRESHOLD_DSPEVE....4140
Table 18-171. CTRL_CORE_BANDGAP_THRESHOLD_IVA....4141
Table 18-172. CTRL_CORE_BANDGAP_TSHUT_DSPEVE....4141
Table 18-173. CTRL_CORE_BANDGAP_TSHUT_IVA....4142
Table 18-174. CTRL_CORE_BANDGAP_STATUS_2....4142
Table 18-175. CTRL_CORE_DTEMP_DSPEVE_0....4143
Table 18-176. CTRL_CORE_DTEMP_DSPEVE_1....4143
Table 18-177. CTRL_CORE_DTEMP_DSPEVE_2....4143
Table 18-178. CTRL_CORE_DTEMP_DSPEVE_3....4144
Table 18-179. CTRL_CORE_DTEMP_DSPEVE_4....4144
Table 18-180. CTRL_CORE_DTEMP_IVA_0....4144
Table 18-181. CTRL_CORE_DTEMP_IVA_1....4144
Table 18-182. CTRL_CORE_DTEMP_IVA_2....4145
Table 18-183. CTRL_CORE_DTEMP_IVA_3....4145
Table 18-184. CTRL_CORE_DTEMP_IVA_4....4145
Table 18-185. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_5....4146
Table 18-186. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_2....4146
Table 18-187. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_3....4147
Table 18-188. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_4....4147
Table 18-189. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5....4147
Table 18-190. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2....4148
Table 18-191. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3....4148
Table 18-192. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4....4149
Table 18-193. CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2....4149

Table 18-194. CTRL_CORE_LDOSRAM_CORE_2_VOLTAGE_CTRL....... 4150
Table 18-195. CTRL_CORE_LDOSRAM_CORE_3_VOLTAGE_CTRL....... 4150
Table 18-196. CTRL_CORE_NMI_DESTINATION_1....... 4151
Table 18-197. CTRL_CORE_NMI_DESTINATION_2....... 4152
Table 18-198. CTRL_CORE_IP_PRESSURE....... 4152
Table 18-199. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_0....... 4153
Table 18-200. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_1....... 4153
Table 18-201. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_2....... 4153
Table 18-202. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_3....... 4154
Table 18-203. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_4....... 4154
Table 18-204. CTRL_CORE_CUST_FUSE_SWRV_7....... 4154
Table 18-205. CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0....... 4154
Table 18-206. CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1....... 4155
Table 18-207. CTRL_CORE_PCIE_POWER_STATE....... 4155
Table 18-208. CTRL_CORE_BOOTSTRAP....... 4155
Table 18-209. CTRL_CORE_MLB_SIG_IO_CTRL....... 4156
Table 18-210. CTRL_CORE_MLB_DAT_IO_CTRL....... 4157
Table 18-211. CTRL_CORE_MLB_CLK_BG_CTRL....... 4158
Table 18-212. CTRL_CORE_EVE1_IRQ_0_1....... 4159
Table 18-213. CTRL_CORE_EVE1_IRQ_2_3....... 4159
Table 18-214. CTRL_CORE_EVE1_IRQ_4_5....... 4159
Table 18-215. CTRL_CORE_EVE1_IRQ_6_7....... 4159
Table 18-216. CTRL_CORE_EVE2_IRQ_0_1....... 4160
Table 18-217. CTRL_CORE_EVE2_IRQ_2_3....... 4160
Table 18-218. CTRL_CORE_EVE2_IRQ_4_5....... 4160
Table 18-219. CTRL_CORE_EVE2_IRQ_6_7....... 4161
Table 18-220. CTRL_CORE_IPU1_IRQ_23_24....... 4161
Table 18-221. CTRL_CORE_IPU1_IRQ_25_26....... 4161
Table 18-222. CTRL_CORE_IPU1_IRQ_27_28....... 4161
Table 18-223. CTRL_CORE_IPU1_IRQ_29_30....... 4162
Table 18-224. CTRL_CORE_IPU1_IRQ_31_32....... 4162
Table 18-225. CTRL_CORE_IPU1_IRQ_33_34....... 4162
Table 18-226. CTRL_CORE_IPU1_IRQ_35_36....... 4163
Table 18-227. CTRL_CORE_IPU1_IRQ_37_38....... 4163
Table 18-228. CTRL_CORE_IPU1_IRQ_39_40....... 4163
Table 18-229. CTRL_CORE_IPU1_IRQ_41_42....... 4164
Table 18-230. CTRL_CORE_IPU1_IRQ_43_44....... 4164
Table 18-231. CTRL_CORE_IPU1_IRQ_45_46....... 4164
Table 18-232. CTRL_CORE_IPU1_IRQ_47_48....... 4164
Table 18-233. CTRL_CORE_IPU1_IRQ_49_50....... 4165
Table 18-234. CTRL_CORE_IPU1_IRQ_51_52....... 4165
Table 18-235. CTRL_CORE_IPU1_IRQ_53_54....... 4165
Table 18-236. CTRL_CORE_IPU1_IRQ_55_56....... 4166
Table 18-237. CTRL_CORE_IPU1_IRQ_57_58....... 4166
Table 18-238. CTRL_CORE_IPU1_IRQ_59_60....... 4166
Table 18-239. CTRL_CORE_IPU1_IRQ_61_62....... 4167
Table 18-240. CTRL_CORE_IPU1_IRQ_63_64....... 4167
Table 18-241. CTRL_CORE_IPU1_IRQ_65_66....... 4167
Table 18-242. CTRL_CORE_IPU1_IRQ_67_68....... 4167
Table 18-243. CTRL_CORE_IPU1_IRQ_69_70....... 4168
Table 18-244. CTRL_CORE_IPU1_IRQ_71_72....... 4168
Table 18-245. CTRL_CORE_IPU1_IRQ_73_74....... 4168
Table 18-246. CTRL_CORE_IPU1_IRQ_75_76....... 4169
Table 18-247. CTRL_CORE_IPU1_IRQ_77_78....... 4169
Table 18-248. CTRL_CORE_IPU1_IRQ_79_80....... 4169
Table 18-249. CTRL_CORE_IPU2_IRQ_23_24....... 4169
Table 18-250. CTRL_CORE_IPU2_IRQ_25_26....... 4170
Table 18-251. CTRL_CORE_IPU2_IRQ_27_28....... 4170
Table 18-252. CTRL_CORE_IPU2_IRQ_29_30....... 4170
Table 18-253. CTRL_CORE_IPU2_IRQ_31_32....... 4171
Table 18-254. CTRL_CORE_IPU2_IRQ_33_34....... 4171

Table 18-255. CTRL_CORE_IPU2_IRQ_35_36....4171
Table 18-256. CTRL_CORE_IPU2_IRQ_37_38....4172
Table 18-257. CTRL_CORE_IPU2_IRQ_39_40....4172
Table 18-258. CTRL_CORE_IPU2_IRQ_41_42....4172
Table 18-259. CTRL_CORE_IPU2_IRQ_43_44....4172
Table 18-260. CTRL_CORE_IPU2_IRQ_45_46....4173
Table 18-261. CTRL_CORE_IPU2_IRQ_47_48....4173
Table 18-262. CTRL_CORE_IPU2_IRQ_49_50....4173
Table 18-263. CTRL_CORE_IPU2_IRQ_51_52....4174
Table 18-264. CTRL_CORE_IPU2_IRQ_53_54....4174
Table 18-265. CTRL_CORE_IPU2_IRQ_55_56....4174
Table 18-266. CTRL_CORE_IPU2_IRQ_57_58....4174
Table 18-267. CTRL_CORE_IPU2_IRQ_59_60....4175
Table 18-268. CTRL_CORE_IPU2_IRQ_61_62....4175
Table 18-269. CTRL_CORE_IPU2_IRQ_63_64....4175
Table 18-270. CTRL_CORE_IPU2_IRQ_65_66....4176
Table 18-271. CTRL_CORE_IPU2_IRQ_67_68....4176
Table 18-272. CTRL_CORE_IPU2_IRQ_69_70....4176
Table 18-273. CTRL_CORE_IPU2_IRQ_71_72....4177
Table 18-274. CTRL_CORE_IPU2_IRQ_73_74....4177
Table 18-275. CTRL_CORE_IPU2_IRQ_75_76....4177
Table 18-276. CTRL_CORE_IPU2_IRQ_77_78....4177
Table 18-277. CTRL_CORE_IPU2_IRQ_79_80....4178
Table 18-278. CTRL_CORE_DSP1_IRQ_32_33....4178
Table 18-279. CTRL_CORE_DSP1_IRQ_34_35....4178
Table 18-280. CTRL_CORE_DSP1_IRQ_36_37....4179
Table 18-281. CTRL_CORE_DSP1_IRQ_38_39....4179
Table 18-282. CTRL_CORE_DSP1_IRQ_40_41....4179
Table 18-283. CTRL_CORE_DSP1_IRQ_42_43....4180
Table 18-284. CTRL_CORE_DSP1_IRQ_44_45....4180
Table 18-285. CTRL_CORE_DSP1_IRQ_46_47....4180
Table 18-286. CTRL_CORE_DSP1_IRQ_48_49....4180
Table 18-287. CTRL_CORE_DSP1_IRQ_50_51....4181
Table 18-288. CTRL_CORE_DSP1_IRQ_52_53....4181
Table 18-289. CTRL_CORE_DSP1_IRQ_54_55....4181
Table 18-290. CTRL_CORE_DSP1_IRQ_56_57....4182
Table 18-291. CTRL_CORE_DSP1_IRQ_58_59....4182
Table 18-292. CTRL_CORE_DSP1_IRQ_60_61....4182
Table 18-293. CTRL_CORE_DSP1_IRQ_62_63....4182
Table 18-294. CTRL_CORE_DSP1_IRQ_64_65....4183
Table 18-295. CTRL_CORE_DSP1_IRQ_66_67....4183
Table 18-296. CTRL_CORE_DSP1_IRQ_68_69....4183
Table 18-297. CTRL_CORE_DSP1_IRQ_70_71....4184
Table 18-298. CTRL_CORE_DSP1_IRQ_72_73....4184
Table 18-299. CTRL_CORE_DSP1_IRQ_74_75....4184
Table 18-300. CTRL_CORE_DSP1_IRQ_76_77....4185
Table 18-301. CTRL_CORE_DSP1_IRQ_78_79....4185
Table 18-302. CTRL_CORE_DSP1_IRQ_80_81....4185
Table 18-303. CTRL_CORE_DSP1_IRQ_82_83....4185
Table 18-304. CTRL_CORE_DSP1_IRQ_84_85....4186
Table 18-305. CTRL_CORE_DSP1_IRQ_86_87....4186
Table 18-306. CTRL_CORE_DSP1_IRQ_88_89....4186
Table 18-307. CTRL_CORE_DSP1_IRQ_90_91....4187
Table 18-308. CTRL_CORE_DSP1_IRQ_92_93....4187
Table 18-309. CTRL_CORE_DSP1_IRQ_94_95....4187
Table 18-310. CTRL_CORE_DSP2_IRQ_32_33....4188
Table 18-311. CTRL_CORE_DSP2_IRQ_34_35....4188
Table 18-312. CTRL_CORE_DSP2_IRQ_36_37....4188
Table 18-313. CTRL_CORE_DSP2_IRQ_38_39....4188
Table 18-314. CTRL_CORE_DSP2_IRQ_40_41....4189
Table 18-315. CTRL_CORE_DSP2_IRQ_42_43....4189

Table 18-316. CTRL_CORE_DSP2_IRQ_44_45 ... 4189
Table 18-317. CTRL_CORE_DSP2_IRQ_46_47 ... 4190
Table 18-318. CTRL_CORE_DSP2_IRQ_48_49 ... 4190
Table 18-319. CTRL_CORE_DSP2_IRQ_50_51 ... 4190
Table 18-320. CTRL_CORE_DSP2_IRQ_52_53 ... 4190
Table 18-321. CTRL_CORE_DSP2_IRQ_54_55 ... 4191
Table 18-322. CTRL_CORE_DSP2_IRQ_56_57 ... 4191
Table 18-323. CTRL_CORE_DSP2_IRQ_58_59 ... 4191
Table 18-324. CTRL_CORE_DSP2_IRQ_60_61 ... 4192
Table 18-325. CTRL_CORE_DSP2_IRQ_62_63 ... 4192
Table 18-326. CTRL_CORE_DSP2_IRQ_64_65 ... 4192
Table 18-327. CTRL_CORE_DSP2_IRQ_66_67 ... 4193
Table 18-328. CTRL_CORE_DSP2_IRQ_68_69 ... 4193
Table 18-329. CTRL_CORE_DSP2_IRQ_70_71 ... 4193
Table 18-330. CTRL_CORE_DSP2_IRQ_72_73 ... 4193
Table 18-331. CTRL_CORE_DSP2_IRQ_74_75 ... 4194
Table 18-332. CTRL_CORE_DSP2_IRQ_76_77 ... 4194
Table 18-333. CTRL_CORE_DSP2_IRQ_78_79 ... 4194
Table 18-334. CTRL_CORE_DSP2_IRQ_80_81 ... 4195
Table 18-335. CTRL_CORE_DSP2_IRQ_82_83 ... 4195
Table 18-336. CTRL_CORE_DSP2_IRQ_84_85 ... 4195
Table 18-337. CTRL_CORE_DSP2_IRQ_86_87 ... 4196
Table 18-338. CTRL_CORE_DSP2_IRQ_88_89 ... 4196
Table 18-339. CTRL_CORE_DSP2_IRQ_90_91 ... 4196
Table 18-340. CTRL_CORE_DSP2_IRQ_92_93 ... 4196
Table 18-341. CTRL_CORE_DSP2_IRQ_94_95 ... 4197
Table 18-342. CTRL_CORE_MPU_IRQ_4_7 ... 4197
Table 18-343. CTRL_CORE_MPU_IRQ_8_9 ... 4197
Table 18-344. CTRL_CORE_MPU_IRQ_10_11 ... 4198
Table 18-345. CTRL_CORE_MPU_IRQ_12_13 ... 4198
Table 18-346. CTRL_CORE_MPU_IRQ_14_15 ... 4198
Table 18-347. CTRL_CORE_MPU_IRQ_16_17 ... 4199
Table 18-348. CTRL_CORE_MPU_IRQ_18_19 ... 4199
Table 18-349. CTRL_CORE_MPU_IRQ_20_21 ... 4199
Table 18-350. CTRL_CORE_MPU_IRQ_22_23 ... 4199
Table 18-351. CTRL_CORE_MPU_IRQ_24_25 ... 4200
Table 18-352. CTRL_CORE_MPU_IRQ_26_27 ... 4200
Table 18-353. CTRL_CORE_MPU_IRQ_28_29 ... 4200
Table 18-354. CTRL_CORE_MPU_IRQ_30_31 ... 4201
Table 18-355. CTRL_CORE_MPU_IRQ_32_33 ... 4201
Table 18-356. CTRL_CORE_MPU_IRQ_34_35 ... 4201
Table 18-357. CTRL_CORE_MPU_IRQ_36_37 ... 4201
Table 18-358. CTRL_CORE_MPU_IRQ_38_39 ... 4202
Table 18-359. CTRL_CORE_MPU_IRQ_40_41 ... 4202
Table 18-360. CTRL_CORE_MPU_IRQ_42_43 ... 4202
Table 18-361. CTRL_CORE_MPU_IRQ_44_45 ... 4203
Table 18-362. CTRL_CORE_MPU_IRQ_46_47 ... 4203
Table 18-363. CTRL_CORE_MPU_IRQ_48_49 ... 4203
Table 18-364. CTRL_CORE_MPU_IRQ_50_51 ... 4204
Table 18-365. CTRL_CORE_MPU_IRQ_52_53 ... 4204
Table 18-366. CTRL_CORE_MPU_IRQ_54_55 ... 4204
Table 18-367. CTRL_CORE_MPU_IRQ_56_57 ... 4204
Table 18-368. CTRL_CORE_MPU_IRQ_58_59 ... 4205
Table 18-369. CTRL_CORE_MPU_IRQ_60_61 ... 4205
Table 18-370. CTRL_CORE_MPU_IRQ_62_63 ... 4205
Table 18-371. CTRL_CORE_MPU_IRQ_64_65 ... 4206
Table 18-372. CTRL_CORE_MPU_IRQ_66_67 ... 4206
Table 18-373. CTRL_CORE_MPU_IRQ_68_69 ... 4206
Table 18-374. CTRL_CORE_MPU_IRQ_70_71 ... 4207
Table 18-375. CTRL_CORE_MPU_IRQ_72_73 ... 4207
Table 18-376. CTRL_CORE_MPU_IRQ_74_75 ... 4207

Table 18-377. CTRL_CORE_MPU_IRQ_76_77....4207
Table 18-378. CTRL_CORE_MPU_IRQ_78_79....4208
Table 18-379. CTRL_CORE_MPU_IRQ_80_81....4208
Table 18-380. CTRL_CORE_MPU_IRQ_82_83....4208
Table 18-381. CTRL_CORE_MPU_IRQ_84_85....4209
Table 18-382. CTRL_CORE_MPU_IRQ_86_87....4209
Table 18-383. CTRL_CORE_MPU_IRQ_88_89....4209
Table 18-384. CTRL_CORE_MPU_IRQ_90_91....4209
Table 18-385. CTRL_CORE_MPU_IRQ_92_93....4210
Table 18-386. CTRL_CORE_MPU_IRQ_94_95....4210
Table 18-387. CTRL_CORE_MPU_IRQ_96_97....4210
Table 18-388. CTRL_CORE_MPU_IRQ_98_99....4211
Table 18-389. CTRL_CORE_MPU_IRQ_100_101....4211
Table 18-390. CTRL_CORE_MPU_IRQ_102_103....4211
Table 18-391. CTRL_CORE_MPU_IRQ_104_105....4212
Table 18-392. CTRL_CORE_MPU_IRQ_106_107....4212
Table 18-393. CTRL_CORE_MPU_IRQ_108_109....4212
Table 18-394. CTRL_CORE_MPU_IRQ_110_111....4212
Table 18-395. CTRL_CORE_MPU_IRQ_112_113....4213
Table 18-396. CTRL_CORE_MPU_IRQ_114_115....4213
Table 18-397. CTRL_CORE_MPU_IRQ_116_117....4213
Table 18-398. CTRL_CORE_MPU_IRQ_118_119....4214
Table 18-399. CTRL_CORE_MPU_IRQ_120_121....4214
Table 18-400. CTRL_CORE_MPU_IRQ_122_123....4214
Table 18-401. CTRL_CORE_MPU_IRQ_124_125....4215
Table 18-402. CTRL_CORE_MPU_IRQ_126_127....4215
Table 18-403. CTRL_CORE_MPU_IRQ_128_129....4215
Table 18-404. CTRL_CORE_MPU_IRQ_130_133....4215
Table 18-405. CTRL_CORE_MPU_IRQ_134_135....4216
Table 18-406. CTRL_CORE_MPU_IRQ_136_137....4216
Table 18-407. CTRL_CORE_MPU_IRQ_138_139....4216
Table 18-408. CTRL_CORE_MPU_IRQ_140_141....4217
Table 18-409. CTRL_CORE_MPU_IRQ_142_143....4217
Table 18-410. CTRL_CORE_MPU_IRQ_144_145....4217
Table 18-411. CTRL_CORE_MPU_IRQ_146_147....4217
Table 18-412. CTRL_CORE_MPU_IRQ_148_149....4218
Table 18-413. CTRL_CORE_MPU_IRQ_150_151....4218
Table 18-414. CTRL_CORE_MPU_IRQ_152_153....4218
Table 18-415. CTRL_CORE_MPU_IRQ_154_155....4219
Table 18-416. CTRL_CORE_MPU_IRQ_156_157....4219
Table 18-417. CTRL_CORE_MPU_IRQ_158_159....4219
Table 18-418. CTRL_CORE_DMA_SYSTEM_DREQ_0_1....4220
Table 18-419. CTRL_CORE_DMA_SYSTEM_DREQ_2_3....4220
Table 18-420. CTRL_CORE_DMA_SYSTEM_DREQ_4_5....4220
Table 18-421. CTRL_CORE_DMA_SYSTEM_DREQ_6_7....4221
Table 18-422. CTRL_CORE_DMA_SYSTEM_DREQ_8_9....4221
Table 18-423. CTRL_CORE_DMA_SYSTEM_DREQ_10_11....4221
Table 18-424. CTRL_CORE_DMA_SYSTEM_DREQ_12_13....4222
Table 18-425. CTRL_CORE_DMA_SYSTEM_DREQ_14_15....4222
Table 18-426. CTRL_CORE_DMA_SYSTEM_DREQ_16_17....4222
Table 18-427. CTRL_CORE_DMA_SYSTEM_DREQ_18_19....4223
Table 18-428. CTRL_CORE_DMA_SYSTEM_DREQ_20_21....4223
Table 18-429. CTRL_CORE_DMA_SYSTEM_DREQ_22_23....4223
Table 18-430. CTRL_CORE_DMA_SYSTEM_DREQ_24_25....4224
Table 18-431. CTRL_CORE_DMA_SYSTEM_DREQ_26_27....4224
Table 18-432. CTRL_CORE_DMA_SYSTEM_DREQ_28_29....4224
Table 18-433. CTRL_CORE_DMA_SYSTEM_DREQ_30_31....4225
Table 18-434. CTRL_CORE_DMA_SYSTEM_DREQ_32_33....4225
Table 18-435. CTRL_CORE_DMA_SYSTEM_DREQ_34_35....4225
Table 18-436. CTRL_CORE_DMA_SYSTEM_DREQ_36_37....4226
Table 18-437. CTRL_CORE_DMA_SYSTEM_DREQ_38_39....4226

Table 18-438. CTRL_CORE_DMA_SYSTEM_DREQ_40_41 4226
Table 18-439. CTRL_CORE_DMA_SYSTEM_DREQ_42_43 4227
Table 18-440. CTRL_CORE_DMA_SYSTEM_DREQ_44_45 4227
Table 18-441. CTRL_CORE_DMA_SYSTEM_DREQ_46_47 4227
Table 18-442. CTRL_CORE_DMA_SYSTEM_DREQ_48_49 4228
Table 18-443. CTRL_CORE_DMA_SYSTEM_DREQ_50_51 4228
Table 18-444. CTRL_CORE_DMA_SYSTEM_DREQ_52_53 4228
Table 18-445. CTRL_CORE_DMA_SYSTEM_DREQ_54_55 4229
Table 18-446. CTRL_CORE_DMA_SYSTEM_DREQ_56_57 4229
Table 18-447. CTRL_CORE_DMA_SYSTEM_DREQ_58_59 4230
Table 18-448. CTRL_CORE_DMA_SYSTEM_DREQ_60_61 4230
Table 18-449. CTRL_CORE_DMA_SYSTEM_DREQ_62_63 4230
Table 18-450. CTRL_CORE_DMA_SYSTEM_DREQ_64_65 4231
Table 18-451. CTRL_CORE_DMA_SYSTEM_DREQ_66_67 4231
Table 18-452. CTRL_CORE_DMA_SYSTEM_DREQ_68_69 4231
Table 18-453. CTRL_CORE_DMA_SYSTEM_DREQ_70_71 4232
Table 18-454. CTRL_CORE_DMA_SYSTEM_DREQ_72_73 4232
Table 18-455. CTRL_CORE_DMA_SYSTEM_DREQ_74_75 4232
Table 18-456. CTRL_CORE_DMA_SYSTEM_DREQ_76_77 4233
Table 18-457. CTRL_CORE_DMA_SYSTEM_DREQ_78_79 4233
Table 18-458. CTRL_CORE_DMA_SYSTEM_DREQ_80_81 4233
Table 18-459. CTRL_CORE_DMA_SYSTEM_DREQ_82_83 4234
Table 18-460. CTRL_CORE_DMA_SYSTEM_DREQ_84_85 4234
Table 18-461. CTRL_CORE_DMA_SYSTEM_DREQ_86_87 4234
Table 18-462. CTRL_CORE_DMA_SYSTEM_DREQ_88_89 4235
Table 18-463. CTRL_CORE_DMA_SYSTEM_DREQ_90_91 4235
Table 18-464. CTRL_CORE_DMA_SYSTEM_DREQ_92_93 4235
Table 18-465. CTRL_CORE_DMA_SYSTEM_DREQ_94_95 4236
Table 18-466. CTRL_CORE_DMA_SYSTEM_DREQ_96_97 4236
Table 18-467. CTRL_CORE_DMA_SYSTEM_DREQ_98_99 4236
Table 18-468. CTRL_CORE_DMA_SYSTEM_DREQ_100_101 4237
Table 18-469. CTRL_CORE_DMA_SYSTEM_DREQ_102_103 4237
Table 18-470. CTRL_CORE_DMA_SYSTEM_DREQ_104_105 4237
Table 18-471. CTRL_CORE_DMA_SYSTEM_DREQ_106_107 4238
Table 18-472. CTRL_CORE_DMA_SYSTEM_DREQ_108_109 4238
Table 18-473. CTRL_CORE_DMA_SYSTEM_DREQ_110_111 4239
Table 18-474. CTRL_CORE_DMA_SYSTEM_DREQ_112_113 4239
Table 18-475. CTRL_CORE_DMA_SYSTEM_DREQ_114_115 4239
Table 18-476. CTRL_CORE_DMA_SYSTEM_DREQ_116_117 4240
Table 18-477. CTRL_CORE_DMA_SYSTEM_DREQ_118_119 4240
Table 18-478. CTRL_CORE_DMA_SYSTEM_DREQ_120_121 4240
Table 18-479. CTRL_CORE_DMA_SYSTEM_DREQ_122_123 4241
Table 18-480. CTRL_CORE_DMA_SYSTEM_DREQ_124_125 4241
Table 18-481. CTRL_CORE_DMA_SYSTEM_DREQ_126_127 4241
Table 18-482. CTRL_CORE_DMA_EDMA_DREQ_0_1 4242
Table 18-483. CTRL_CORE_DMA_EDMA_DREQ_2_3 4242
Table 18-484. CTRL_CORE_DMA_EDMA_DREQ_4_5 4242
Table 18-485. CTRL_CORE_DMA_EDMA_DREQ_6_7 4243
Table 18-486. CTRL_CORE_DMA_EDMA_DREQ_8_9 4243
Table 18-487. CTRL_CORE_DMA_EDMA_DREQ_10_11 4243
Table 18-488. CTRL_CORE_DMA_EDMA_DREQ_12_13 4244
Table 18-489. CTRL_CORE_DMA_EDMA_DREQ_14_15 4244
Table 18-490. CTRL_CORE_DMA_EDMA_DREQ_16_17 4244
Table 18-491. CTRL_CORE_DMA_EDMA_DREQ_18_19 4245
Table 18-492. CTRL_CORE_DMA_EDMA_DREQ_20_21 4245
Table 18-493. CTRL_CORE_DMA_EDMA_DREQ_22_23 4245
Table 18-494. CTRL_CORE_DMA_EDMA_DREQ_24_25 4246
Table 18-495. CTRL_CORE_DMA_EDMA_DREQ_26_27 4246
Table 18-496. CTRL_CORE_DMA_EDMA_DREQ_28_29 4246
Table 18-497. CTRL_CORE_DMA_EDMA_DREQ_30_31 4247
Table 18-498. CTRL_CORE_DMA_EDMA_DREQ_32_33 4247

Table 18-499. CTRL_CORE_DMA_EDMA_DREQ_34_35 4247
Table 18-500. CTRL_CORE_DMA_EDMA_DREQ_36_37 4248
Table 18-501. CTRL_CORE_DMA_EDMA_DREQ_38_39 4248
Table 18-502. CTRL_CORE_DMA_EDMA_DREQ_40_41 4248
Table 18-503. CTRL_CORE_DMA_EDMA_DREQ_42_43 4249
Table 18-504. CTRL_CORE_DMA_EDMA_DREQ_44_45 4249
Table 18-505. CTRL_CORE_DMA_EDMA_DREQ_46_47 4249
Table 18-506. CTRL_CORE_DMA_EDMA_DREQ_48_49 4250
Table 18-507. CTRL_CORE_DMA_EDMA_DREQ_50_51 4250
Table 18-508. CTRL_CORE_DMA_EDMA_DREQ_52_53 4250
Table 18-509. CTRL_CORE_DMA_EDMA_DREQ_54_55 4251
Table 18-510. CTRL_CORE_DMA_EDMA_DREQ_56_57 4251
Table 18-511. CTRL_CORE_DMA_EDMA_DREQ_58_59 4251
Table 18-512. CTRL_CORE_DMA_EDMA_DREQ_60_61 4252
Table 18-513. CTRL_CORE_DMA_EDMA_DREQ_62_63 4252
Table 18-514. CTRL_CORE_DMA_DSP1_DREQ_0_1 4252
Table 18-515. CTRL_CORE_DMA_DSP1_DREQ_2_3 4253
Table 18-516. CTRL_CORE_DMA_DSP1_DREQ_4_5 4253
Table 18-517. CTRL_CORE_DMA_DSP1_DREQ_6_7 4253
Table 18-518. CTRL_CORE_DMA_DSP1_DREQ_8_9 4254
Table 18-519. CTRL_CORE_DMA_DSP1_DREQ_10_11 4254
Table 18-520. CTRL_CORE_DMA_DSP1_DREQ_12_13 4254
Table 18-521. CTRL_CORE_DMA_DSP1_DREQ_14_15 4255
Table 18-522. CTRL_CORE_DMA_DSP1_DREQ_16_17 4255
Table 18-523. CTRL_CORE_DMA_DSP1_DREQ_18_19 4255
Table 18-524. CTRL_CORE_DMA_DSP2_DREQ_0_1 4256
Table 18-525. CTRL_CORE_DMA_DSP2_DREQ_2_3 4256
Table 18-526. CTRL_CORE_DMA_DSP2_DREQ_4_5 4256
Table 18-527. CTRL_CORE_DMA_DSP2_DREQ_6_7 4257
Table 18-528. CTRL_CORE_DMA_DSP2_DREQ_8_9 4257
Table 18-529. CTRL_CORE_DMA_DSP2_DREQ_10_11 4257
Table 18-530. CTRL_CORE_DMA_DSP2_DREQ_12_13 4258
Table 18-531. CTRL_CORE_DMA_DSP2_DREQ_14_15 4258
Table 18-532. CTRL_CORE_DMA_DSP2_DREQ_16_17 4258
Table 18-533. CTRL_CORE_DMA_DSP2_DREQ_18_19 4259
Table 18-534. CTRL_CORE_OVS_DMARQ_IO_MUX 4259
Table 18-535. CTRL_CORE_OVS_IRQ_IO_MUX 4259
Table 18-536. CTRL_CORE_CONTROL_PBIAS 4259
Table 18-537. CTRL_CORE_CONTROL_HDMI_TX_PHY 4260
Table 18-538. CTRL_CORE_CONTROL_USB2PHYCORE 4261
Table 18-539. CTRL_CORE_CONTROL_HDMI_1 4263
Table 18-540. CTRL_CORE_CONTROL_DDRCACH1_0 4264
Table 18-541. CTRL_CORE_CONTROL_DDRCACH2_0 4266
Table 18-542. CTRL_CORE_CONTROL_DDRCH1_0 4269
Table 18-543. CTRL_CORE_CONTROL_DDRCH1_1 4271
Table 18-544. CTRL_CORE_CONTROL_DDRCH2_0 4274
Table 18-545. CTRL_CORE_CONTROL_DDRCH2_1 4276
Table 18-546. CTRL_CORE_CONTROL_DDRCH1_2 4279
Table 18-547. CTRL_CORE_CONTROL_DDRIO_0 4280
Table 18-548. CTRL_CORE_CONTROL_DDRIO_1 4281
Table 18-549. CTRL_CORE_CONTROL_HYST_1 4283
Table 18-550. CTRL_CORE_CONTROL_SPARE_RW 4283
Table 18-551. CTRL_CORE_SRCOMP_NORTH_SIDE 4284
Table 18-552. CTRL_CORE_SRCOMP_SOUTH_SIDE 4286
Table 18-553. CTRL_CORE_PAD_GPMC_AD0 4286
Table 18-554. CTRL_CORE_PAD_GPMC_AD1 4287
Table 18-555. CTRL_CORE_PAD_GPMC_AD2 4289
Table 18-556. CTRL_CORE_PAD_GPMC_AD3 4290
Table 18-557. CTRL_CORE_PAD_GPMC_AD4 4291
Table 18-558. CTRL_CORE_PAD_GPMC_AD5 4292
Table 18-559. CTRL_CORE_PAD_GPMC_AD6 4293

Table 18-560. CTRL_CORE_PAD_GPMC_AD7 4294
Table 18-561. CTRL_CORE_PAD_GPMC_AD8 4296
Table 18-562. CTRL_CORE_PAD_GPMC_AD9 4297
Table 18-563. CTRL_CORE_PAD_GPMC_AD10 4298
Table 18-564. CTRL_CORE_PAD_GPMC_AD11 4299
Table 18-565. CTRL_CORE_PAD_GPMC_AD12 4300
Table 18-566. CTRL_CORE_PAD_GPMC_AD13 4301
Table 18-567. CTRL_CORE_PAD_GPMC_AD14 4303
Table 18-568. CTRL_CORE_PAD_GPMC_AD15 4304
Table 18-569. CTRL_CORE_PAD_GPMC_A0 4305
Table 18-570. CTRL_CORE_PAD_GPMC_A1 4306
Table 18-571. CTRL_CORE_PAD_GPMC_A2 4308
Table 18-572. CTRL_CORE_PAD_GPMC_A3 4309
Table 18-573. CTRL_CORE_PAD_GPMC_A4 4310
Table 18-574. CTRL_CORE_PAD_GPMC_A5 4312
Table 18-575. CTRL_CORE_PAD_GPMC_A6 4313
Table 18-576. CTRL_CORE_PAD_GPMC_A7 4314
Table 18-577. CTRL_CORE_PAD_GPMC_A8 4316
Table 18-578. CTRL_CORE_PAD_GPMC_A9 4317
Table 18-579. CTRL_CORE_PAD_GPMC_A10 4318
Table 18-580. CTRL_CORE_PAD_GPMC_A11 4319
Table 18-581. CTRL_CORE_PAD_GPMC_A12 4321
Table 18-582. CTRL_CORE_PAD_GPMC_A13 4322
Table 18-583. CTRL_CORE_PAD_GPMC_A14 4323
Table 18-584. CTRL_CORE_PAD_GPMC_A15 4325
Table 18-585. CTRL_CORE_PAD_GPMC_A16 4326
Table 18-586. CTRL_CORE_PAD_GPMC_A17 4327
Table 18-587. CTRL_CORE_PAD_GPMC_A18 4328
Table 18-588. CTRL_CORE_PAD_GPMC_A19 4329
Table 18-589. CTRL_CORE_PAD_GPMC_A20 4330
Table 18-590. CTRL_CORE_PAD_GPMC_A21 4332
Table 18-591. CTRL_CORE_PAD_GPMC_A22 4333
Table 18-592. CTRL_CORE_PAD_GPMC_A23 4334
Table 18-593. CTRL_CORE_PAD_GPMC_A24 4335
Table 18-594. CTRL_CORE_PAD_GPMC_A25 4336
Table 18-595. CTRL_CORE_PAD_GPMC_A26 4338
Table 18-596. CTRL_CORE_PAD_GPMC_A27 4339
Table 18-597. CTRL_CORE_PAD_GPMC_CS1 4340
Table 18-598. CTRL_CORE_PAD_GPMC_CS0 4341
Table 18-599. CTRL_CORE_PAD_GPMC_CS2 4342
Table 18-600. CTRL_CORE_PAD_GPMC_CS3 4343
Table 18-601. CTRL_CORE_PAD_GPMC_CLK 4345
Table 18-602. CTRL_CORE_PAD_GPMC_ADVN_ALE 4346
Table 18-603. CTRL_CORE_PAD_GPMC_OEN_REN 4347
Table 18-604. CTRL_CORE_PAD_GPMC_WEN 4348
Table 18-605. CTRL_CORE_PAD_GPMC_BEN0 4350
Table 18-606. CTRL_CORE_PAD_GPMC_BEN1 4351
Table 18-607. CTRL_CORE_PAD_GPMC_WAIT0 4352
Table 18-608. CTRL_CORE_PAD_VIN1A_CLK0 4353
Table 18-609. CTRL_CORE_PAD_VIN1B_CLK1 4355
Table 18-610. CTRL_CORE_PAD_VIN1A_DE0 4356
Table 18-611. CTRL_CORE_PAD_VIN1A_FLD0 4357
Table 18-612. CTRL_CORE_PAD_VIN1A_HSYNC0 4358
Table 18-613. CTRL_CORE_PAD_VIN1A_VSYNC0 4360
Table 18-614. CTRL_CORE_PAD_VIN1A_D0 4361
Table 18-615. CTRL_CORE_PAD_VIN1A_D1 4362
Table 18-616. CTRL_CORE_PAD_VIN1A_D2 4363
Table 18-617. CTRL_CORE_PAD_VIN1A_D3 4364
Table 18-618. CTRL_CORE_PAD_VIN1A_D4 4366
Table 18-619. CTRL_CORE_PAD_VIN1A_D5 4367
Table 18-620. CTRL_CORE_PAD_VIN1A_D6 4368

Table 18-621. CTRL_CORE_PAD_VIN1A_D7.. 4369
Table 18-622. CTRL_CORE_PAD_VIN1A_D8.. 4370
Table 18-623. CTRL_CORE_PAD_VIN1A_D9.. 4371
Table 18-624. CTRL_CORE_PAD_VIN1A_D10.. 4373
Table 18-625. CTRL_CORE_PAD_VIN1A_D11.. 4374
Table 18-626. CTRL_CORE_PAD_VIN1A_D12.. 4375
Table 18-627. CTRL_CORE_PAD_VIN1A_D13.. 4376
Table 18-628. CTRL_CORE_PAD_VIN1A_D14.. 4378
Table 18-629. CTRL_CORE_PAD_VIN1A_D15.. 4379
Table 18-630. CTRL_CORE_PAD_VIN1A_D16.. 4380
Table 18-631. CTRL_CORE_PAD_VIN1A_D17.. 4381
Table 18-632. CTRL_CORE_PAD_VIN1A_D18.. 4382
Table 18-633. CTRL_CORE_PAD_VIN1A_D19.. 4384
Table 18-634. CTRL_CORE_PAD_VIN1A_D20.. 4385
Table 18-635. CTRL_CORE_PAD_VIN1A_D21.. 4386
Table 18-636. CTRL_CORE_PAD_VIN1A_D22.. 4387
Table 18-637. CTRL_CORE_PAD_VIN1A_D23.. 4388
Table 18-638. CTRL_CORE_PAD_VIN2A_CLK0..4390
Table 18-639. CTRL_CORE_PAD_VIN2A_DE0..4391
Table 18-640. CTRL_CORE_PAD_VIN2A_FLD0..4392
Table 18-641. CTRL_CORE_PAD_VIN2A_HSYNC0.. 4393
Table 18-642. CTRL_CORE_PAD_VIN2A_VSYNC0.. 4394
Table 18-643. CTRL_CORE_PAD_VIN2A_D0.. 4396
Table 18-644. CTRL_CORE_PAD_VIN2A_D1.. 4397
Table 18-645. CTRL_CORE_PAD_VIN2A_D2.. 4398
Table 18-646. CTRL_CORE_PAD_VIN2A_D3.. 4399
Table 18-647. CTRL_CORE_PAD_VIN2A_D4.. 4401
Table 18-648. CTRL_CORE_PAD_VIN2A_D5.. 4402
Table 18-649. CTRL_CORE_PAD_VIN2A_D6.. 4403
Table 18-650. CTRL_CORE_PAD_VIN2A_D7.. 4404
Table 18-651. CTRL_CORE_PAD_VIN2A_D8.. 4405
Table 18-652. CTRL_CORE_PAD_VIN2A_D9.. 4407
Table 18-653. CTRL_CORE_PAD_VIN2A_D10.. 4408
Table 18-654. CTRL_CORE_PAD_VIN2A_D11.. 4409
Table 18-655. CTRL_CORE_PAD_VIN2A_D12.. 4410
Table 18-656. CTRL_CORE_PAD_VIN2A_D13.. 4411
Table 18-657. CTRL_CORE_PAD_VIN2A_D14.. 4413
Table 18-658. CTRL_CORE_PAD_VIN2A_D15.. 4414
Table 18-659. CTRL_CORE_PAD_VIN2A_D16.. 4415
Table 18-660. CTRL_CORE_PAD_VIN2A_D17.. 4416
Table 18-661. CTRL_CORE_PAD_VIN2A_D18.. 4417
Table 18-662. CTRL_CORE_PAD_VIN2A_D19.. 4419
Table 18-663. CTRL_CORE_PAD_VIN2A_D20.. 4420
Table 18-664. CTRL_CORE_PAD_VIN2A_D21.. 4421
Table 18-665. CTRL_CORE_PAD_VIN2A_D22.. 4423
Table 18-666. CTRL_CORE_PAD_VIN2A_D23.. 4424
Table 18-667. CTRL_CORE_PAD_VOUT1_CLK.. 4425
Table 18-668. CTRL_CORE_PAD_VOUT1_DE.. 4427
Table 18-669. CTRL_CORE_PAD_VOUT1_FLD.. 4428
Table 18-670. CTRL_CORE_PAD_VOUT1_HSYNC...4429
Table 18-671. CTRL_CORE_PAD_VOUT1_VSYNC...4430
Table 18-672. CTRL_CORE_PAD_VOUT1_D0...4431
Table 18-673. CTRL_CORE_PAD_VOUT1_D1...4433
Table 18-674. CTRL_CORE_PAD_VOUT1_D2...4434
Table 18-675. CTRL_CORE_PAD_VOUT1_D3...4435
Table 18-676. CTRL_CORE_PAD_VOUT1_D4...4436
Table 18-677. CTRL_CORE_PAD_VOUT1_D5...4438
Table 18-678. CTRL_CORE_PAD_VOUT1_D6...4439
Table 18-679. CTRL_CORE_PAD_VOUT1_D7...4440
Table 18-680. CTRL_CORE_PAD_VOUT1_D8...4441
Table 18-681. CTRL_CORE_PAD_VOUT1_D9...4442

Table 18-682. CTRL_CORE_PAD_VOUT1_D10....4443
Table 18-683. CTRL_CORE_PAD_VOUT1_D11....4445
Table 18-684. CTRL_CORE_PAD_VOUT1_D12....4446
Table 18-685. CTRL_CORE_PAD_VOUT1_D13....4447
Table 18-686. CTRL_CORE_PAD_VOUT1_D14....4449
Table 18-687. CTRL_CORE_PAD_VOUT1_D15....4450
Table 18-688. CTRL_CORE_PAD_VOUT1_D16....4451
Table 18-689. CTRL_CORE_PAD_VOUT1_D17....4453
Table 18-690. CTRL_CORE_PAD_VOUT1_D18....4454
Table 18-691. CTRL_CORE_PAD_VOUT1_D19....4455
Table 18-692. CTRL_CORE_PAD_VOUT1_D20....4456
Table 18-693. CTRL_CORE_PAD_VOUT1_D21....4458
Table 18-694. CTRL_CORE_PAD_VOUT1_D22....4459
Table 18-695. CTRL_CORE_PAD_VOUT1_D23....4460
Table 18-696. CTRL_CORE_PAD_MDIO_MCLK....4462
Table 18-697. CTRL_CORE_PAD_MDIO_D....4463
Table 18-698. CTRL_CORE_PAD_RMII_MHZ_50_CLK....4464
Table 18-699. CTRL_CORE_PAD_UART3_RXD....4465
Table 18-700. CTRL_CORE_PAD_UART3_TXD....4466
Table 18-701. CTRL_CORE_PAD_RGMII0_TXC....4468
Table 18-702. CTRL_CORE_PAD_RGMII0_TXCTL....4469
Table 18-703. CTRL_CORE_PAD_RGMII0_TXD3....4470
Table 18-704. CTRL_CORE_PAD_RGMII0_TXD2....4472
Table 18-705. CTRL_CORE_PAD_RGMII0_TXD1....4473
Table 18-706. CTRL_CORE_PAD_RGMII0_TXD0....4474
Table 18-707. CTRL_CORE_PAD_RGMII0_RXC....4476
Table 18-708. CTRL_CORE_PAD_RGMII0_RXCTL....4477
Table 18-709. CTRL_CORE_PAD_RGMII0_RXD3....4478
Table 18-710. CTRL_CORE_PAD_RGMII0_RXD2....4480
Table 18-711. CTRL_CORE_PAD_RGMII0_RXD1....4481
Table 18-712. CTRL_CORE_PAD_RGMII0_RXD0....4482
Table 18-713. CTRL_CORE_PAD_USB1_DRVVBUS....4483
Table 18-714. CTRL_CORE_PAD_USB2_DRVVBUS....4484
Table 18-715. CTRL_CORE_PAD_GPIO6_14....4486
Table 18-716. CTRL_CORE_PAD_GPIO6_15....4487
Table 18-717. CTRL_CORE_PAD_GPIO6_16....4488
Table 18-718. CTRL_CORE_PAD_XREF_CLK0....4490
Table 18-719. CTRL_CORE_PAD_XREF_CLK1....4491
Table 18-720. CTRL_CORE_PAD_XREF_CLK2....4492
Table 18-721. CTRL_CORE_PAD_XREF_CLK3....4493
Table 18-722. CTRL_CORE_PAD_MCASP1_ACLKX....4495
Table 18-723. CTRL_CORE_PAD_MCASP1_FSX....4496
Table 18-724. CTRL_CORE_PAD_MCASP1_ACLKR....4497
Table 18-725. CTRL_CORE_PAD_MCASP1_FSR....4498
Table 18-726. CTRL_CORE_PAD_MCASP1_AXR0....4500
Table 18-727. CTRL_CORE_PAD_MCASP1_AXR1....4501
Table 18-728. CTRL_CORE_PAD_MCASP1_AXR2....4502
Table 18-729. CTRL_CORE_PAD_MCASP1_AXR3....4503
Table 18-730. CTRL_CORE_PAD_MCASP1_AXR4....4504
Table 18-731. CTRL_CORE_PAD_MCASP1_AXR5....4506
Table 18-732. CTRL_CORE_PAD_MCASP1_AXR6....4507
Table 18-733. CTRL_CORE_PAD_MCASP1_AXR7....4508
Table 18-734. CTRL_CORE_PAD_MCASP1_AXR8....4509
Table 18-735. CTRL_CORE_PAD_MCASP1_AXR9....4510
Table 18-736. CTRL_CORE_PAD_MCASP1_AXR10....4512
Table 18-737. CTRL_CORE_PAD_MCASP1_AXR11....4513
Table 18-738. CTRL_CORE_PAD_MCASP1_AXR12....4514
Table 18-739. CTRL_CORE_PAD_MCASP1_AXR13....4515
Table 18-740. CTRL_CORE_PAD_MCASP1_AXR14....4517
Table 18-741. CTRL_CORE_PAD_MCASP1_AXR15....4518
Table 18-742. CTRL_CORE_PAD_MCASP2_ACLKX....4519

Table 18-743. CTRL_CORE_PAD_MCASP2_FSX....4520
Table 18-744. CTRL_CORE_PAD_MCASP2_ACLKR....4521
Table 18-745. CTRL_CORE_PAD_MCASP2_FSR....4522
Table 18-746. CTRL_CORE_PAD_MCASP2_AXR0....4523
Table 18-747. CTRL_CORE_PAD_MCASP2_AXR1....4525
Table 18-748. CTRL_CORE_PAD_MCASP2_AXR2....4526
Table 18-749. CTRL_CORE_PAD_MCASP2_AXR3....4527
Table 18-750. CTRL_CORE_PAD_MCASP2_AXR4....4528
Table 18-751. CTRL_CORE_PAD_MCASP2_AXR5....4529
Table 18-752. CTRL_CORE_PAD_MCASP2_AXR6....4531
Table 18-753. CTRL_CORE_PAD_MCASP2_AXR7....4532
Table 18-754. CTRL_CORE_PAD_MCASP3_ACLKX....4533
Table 18-755. CTRL_CORE_PAD_MCASP3_FSX....4534
Table 18-756. CTRL_CORE_PAD_MCASP3_AXR0....4535
Table 18-757. CTRL_CORE_PAD_MCASP3_AXR1....4537
Table 18-758. CTRL_CORE_PAD_MCASP4_ACLKX....4538
Table 18-759. CTRL_CORE_PAD_MCASP4_FSX....4539
Table 18-760. CTRL_CORE_PAD_MCASP4_AXR0....4540
Table 18-761. CTRL_CORE_PAD_MCASP4_AXR1....4542
Table 18-762. CTRL_CORE_PAD_MCASP5_ACLKX....4543
Table 18-763. CTRL_CORE_PAD_MCASP5_FSX....4544
Table 18-764. CTRL_CORE_PAD_MCASP5_AXR0....4546
Table 18-765. CTRL_CORE_PAD_MCASP5_AXR1....4547
Table 18-766. CTRL_CORE_PAD_MMC1_CLK....4548
Table 18-767. CTRL_CORE_PAD_MMC1_CMD....4549
Table 18-768. CTRL_CORE_PAD_MMC1_DAT0....4550
Table 18-769. CTRL_CORE_PAD_MMC1_DAT1....4551
Table 18-770. CTRL_CORE_PAD_MMC1_DAT2....4552
Table 18-771. CTRL_CORE_PAD_MMC1_DAT3....4553
Table 18-772. CTRL_CORE_PAD_MMC1_SDCD....4554
Table 18-773. CTRL_CORE_PAD_MMC1_SDWP....4556
Table 18-774. CTRL_CORE_PAD_GPIO6_10....4557
Table 18-775. CTRL_CORE_PAD_GPIO6_11....4558
Table 18-776. CTRL_CORE_PAD_MMC3_CLK....4559
Table 18-777. CTRL_CORE_PAD_MMC3_CMD....4561
Table 18-778. CTRL_CORE_PAD_MMC3_DAT0....4562
Table 18-779. CTRL_CORE_PAD_MMC3_DAT1....4563
Table 18-780. CTRL_CORE_PAD_MMC3_DAT2....4564
Table 18-781. CTRL_CORE_PAD_MMC3_DAT3....4566
Table 18-782. CTRL_CORE_PAD_MMC3_DAT4....4567
Table 18-783. CTRL_CORE_PAD_MMC3_DAT5....4568
Table 18-784. CTRL_CORE_PAD_MMC3_DAT6....4570
Table 18-785. CTRL_CORE_PAD_MMC3_DAT7....4571
Table 18-786. CTRL_CORE_PAD_SPI1_SCLK....4572
Table 18-787. CTRL_CORE_PAD_SPI1_D1....4573
Table 18-788. CTRL_CORE_PAD_SPI1_D0....4574
Table 18-789. CTRL_CORE_PAD_SPI1_CS0....4575
Table 18-790. CTRL_CORE_PAD_SPI1_CS1....4576
Table 18-791. CTRL_CORE_PAD_SPI1_CS2....4577
Table 18-792. CTRL_CORE_PAD_SPI1_CS3....4579
Table 18-793. CTRL_CORE_PAD_SPI2_SCLK....4580
Table 18-794. CTRL_CORE_PAD_SPI2_D1....4581
Table 18-795. CTRL_CORE_PAD_SPI2_D0....4582
Table 18-796. CTRL_CORE_PAD_SPI2_CS0....4583
Table 18-797. CTRL_CORE_PAD_DCAN1_TX....4584
Table 18-798. CTRL_CORE_PAD_DCAN1_RX....4585
Table 18-799. CTRL_CORE_PAD_UART1_RXD....4587
Table 18-800. CTRL_CORE_PAD_UART1_TXD....4588
Table 18-801. CTRL_CORE_PAD_UART1_CTSN....4589
Table 18-802. CTRL_CORE_PAD_UART1_RTSN....4590
Table 18-803. CTRL_CORE_PAD_UART2_RXD....4591

Table 18-804. CTRL_CORE_PAD_UART2_TXD....4593
Table 18-805. CTRL_CORE_PAD_UART2_CTSN....4594
Table 18-806. CTRL_CORE_PAD_UART2_RTSN....4595
Table 18-807. CTRL_CORE_PAD_I2C1_SDA....4596
Table 18-808. CTRL_CORE_PAD_I2C1_SCL....4597
Table 18-809. CTRL_CORE_PAD_I2C2_SDA....4598
Table 18-810. CTRL_CORE_PAD_I2C2_SCL....4598
Table 18-811. CTRL_CORE_PAD_WAKEUP0....4599
Table 18-812. CTRL_CORE_PAD_WAKEUP1....4600
Table 18-813. CTRL_CORE_PAD_WAKEUP2....4601
Table 18-814. CTRL_CORE_PAD_WAKEUP3....4602
Table 18-815. CTRL_CORE_PAD_ON_OFF....4603
Table 18-816. CTRL_CORE_PAD_RTC_PORZ....4604
Table 18-817. CTRL_CORE_PAD_TMS....4604
Table 18-818. CTRL_CORE_PAD_TDI....4605
Table 18-819. CTRL_CORE_PAD_TDO....4606
Table 18-820. CTRL_CORE_PAD_TCLK....4607
Table 18-821. CTRL_CORE_PAD_TRSTN....4607
Table 18-822. CTRL_CORE_PAD_RTCK....4608
Table 18-823. CTRL_CORE_PAD_EMU0....4609
Table 18-824. CTRL_CORE_PAD_EMU1....4610
Table 18-825. CTRL_CORE_PAD_RESETN....4611
Table 18-826. CTRL_CORE_PAD_NMIN_DSP....4612
Table 18-827. CTRL_CORE_PAD_RSTOUTN....4612
Table 18-828. CTRL_CORE_PADCONF_WAKEUPEVENT_0....4613
Table 18-829. CTRL_CORE_PADCONF_WAKEUPEVENT_1....4614
Table 18-830. CTRL_CORE_PADCONF_WAKEUPEVENT_2....4616
Table 18-831. CTRL_CORE_PADCONF_WAKEUPEVENT_3....4618
Table 18-832. CTRL_CORE_PADCONF_WAKEUPEVENT_4....4619
Table 18-833. CTRL_CORE_PADCONF_WAKEUPEVENT_5....4621
Table 18-834. CTRL_CORE_PADCONF_WAKEUPEVENT_6....4623
Table 18-835. CTRL_CORE_PADCONF_WAKEUPEVENT_7....4624
Table 18-836. CTRL_CORE_PADCONF_WAKEUPEVENT_8....4626
Table 18-837. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_2....4627
Table 18-838. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_3....4628
Table 18-839. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_4....4628
Table 18-840. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_5....4629
Table 18-841. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_1....4629
Table 18-842. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_2....4630
Table 18-843. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_3....4630
Table 18-844. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_4....4631
Table 18-845. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_5....4631
Table 18-846. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_0....4632
Table 18-847. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_1....4632
Table 18-848. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_2....4632
Table 18-849. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_3....4632
Table 18-850. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_4....4633
Table 18-851. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_0....4633
Table 18-852. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_1....4633
Table 18-853. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_2....4633
Table 18-854. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_3....4634
Table 18-855. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_4....4634
Table 18-856. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_0....4634
Table 18-857. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_1....4634
Table 18-858. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_2....4635
Table 18-859. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_3....4635
Table 18-860. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_4....4635
Table 18-861. CTRL_CORE_LDOSRAM_CORE_4_VOLTAGE_CTRL....4636
Table 18-862. CTRL_CORE_LDOSRAM_CORE_5_VOLTAGE_CTRL....4636
Table 18-863. CTRL_CORE_LDOSRAM_DSPEVE_2_VOLTAGE_CTRL....4637
Table 18-864. CTRL_CORE_SMA_SW_2....4638

Table 18-865. CTRL_CORE_SMA_SW_3....4638
Table 18-866. CTRL_CORE_SMA_SW_6....4639
Table 18-867. CTRL_CORE_SMA_SW_7....4640
Table 18-868. CTRL_CORE_SMA_SW_8....4641
Table 18-869. CTRL_CORE_SMA_SW_9....4641
Table 18-870. CTRL_CORE_PCIESS1_PCS1....4641
Table 18-871. CTRL_CORE_PCIESS1_PCS2....4641
Table 18-872. CTRL_CORE_PCIESS2_PCS1....4642
Table 18-873. CTRL_CORE_PCIESS2_PCS2....4643
Table 18-874. CTRL_CORE_PCIE_PCS....4643
Table 18-875. CTRL_CORE_PCIE_PCS_REVISION....4644
Table 18-876. CTRL_CORE_PCIE_CONTROL....4644
Table 18-877. CTRL_CORE_PHY_POWER_PCIESS1....4645
Table 18-878. CTRL_CORE_PHY_POWER_PCIESS2....4645
Table 18-879. CTRL_MODULE_WKUP Registers Mapping Summary....4646
Table 18-880. CTRL_WKUP_SEC_CTRL....4648
Table 18-881. CTRL_WKUP_SEC_TAP....4648
Table 18-882. CTRL_WKUP_OCPREG_SPARE....4649
Table 18-883. CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG....4650
Table 18-884. CTRL_WKUP_SECURE_EMIF2_SDRAM_CONFIG....4651
Table 18-885. CTRL_WKUP_STD_FUSE_USB_CONF....4652
Table 18-886. CTRL_WKUP_STD_FUSE_CONF....4653
Table 18-887. CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT....4654
Table 18-888. CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT....4655
Table 18-889. CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1....4657
Table 18-890. CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT_2....4657
Table 18-891. CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL....4657
Table 18-892. CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL....4658
Table 18-893. CTRL_WKUP_LDOSRAM_GPU_VOLTAGE_CTRL....4659
Table 18-894. CTRL_WKUP_LDOSRAM_MPU_VOLTAGE_CTRL....4659
Table 18-895. CTRL_WKUP_LDOSRAM_CORE_VOLTAGE_CTRL....4660
Table 18-896. CTRL_WKUP_LDOSRAM_MPU_2_VOLTAGE_CTRL....4661
Table 18-897. CTRL_WKUP_STD_FUSE_DIE_ID_0....4662
Table 18-898. CTRL_WKUP_ID_CODE....4662
Table 18-899. CTRL_WKUP_STD_FUSE_DIE_ID_1....4663
Table 18-900. CTRL_WKUP_STD_FUSE_DIE_ID_2....4663
Table 18-901. CTRL_WKUP_STD_FUSE_DIE_ID_3....4663
Table 18-902. CTRL_WKUP_STD_FUSE_PROD_ID_0....4663
Table 18-903. CTRL_WKUP_CONTROL_XTAL_OSCILLATOR....4664
Table 18-904. CTRL_WKUP_EFUSE_1....4664
Table 18-905. CTRL_WKUP_EFUSE_2....4666
Table 18-906. CTRL_WKUP_EFUSE_3....4667
Table 18-907. CTRL_WKUP_EFUSE_4....4668
Table 18-908. CTRL_WKUP_EFUSE_13....4670
Table 18-909. IODELAYCONFIG Integration Attributes....4671
Table 18-910. IODELAYCONFIG Clocks and Resets....4671
Table 18-911. IODELAYCONFIG Module Instance Summary....4672
Table 18-912. IODELAYCONFIG Registers Mapping Summary....4672
Table 18-913. CONFIG_REG_0....4689
Table 18-914. Register Call Summary for Register CONFIG_REG_0....4689
Table 18-915. CONFIG_REG_2....4690
Table 18-916. Register Call Summary for Register CONFIG_REG_2....4690
Table 18-917. CONFIG_REG_3....4690
Table 18-918. Register Call Summary for Register CONFIG_REG_3....4690
Table 18-919. CONFIG_REG_4....4690
Table 18-920. Register Call Summary for Register CONFIG_REG_4....4691
Table 18-921. CONFIG_REG_8....4691
Table 18-922. Register Call Summary for Register CONFIG_REG_8....4691
Table 18-923. CFG_RMII_MHZ_50_CLK_IN....4691
Table 18-924. Register Call Summary for Register CFG_RMII_MHZ_50_CLK_IN....4692
Table 18-925. CFG_RMII_MHZ_50_CLK_OEN....4692

Table 18-926. Register Call Summary for Register CFG_RMII_MHZ_50_CLK_OEN....4692
Table 18-927. CFG_RMII_MHZ_50_CLK_OUT....4692
Table 18-928. Register Call Summary for Register CFG_RMII_MHZ_50_CLK_OUT....4693
Table 18-929. CFG_WAKEUP0_IN....4693
Table 18-930. Register Call Summary for Register CFG_WAKEUP0_IN....4693
Table 18-931. CFG_WAKEUP0_OEN....4693
Table 18-932. Register Call Summary for Register CFG_WAKEUP0_OEN....4694
Table 18-933. CFG_WAKEUP0_OUT....4694
Table 18-934. Register Call Summary for Register CFG_WAKEUP0_OUT....4694
Table 18-935. CFG_WAKEUP1_IN....4694
Table 18-936. Register Call Summary for Register CFG_WAKEUP1_IN....4695
Table 18-937. CFG_WAKEUP1_OEN....4695
Table 18-938. Register Call Summary for Register CFG_WAKEUP1_OEN....4695
Table 18-939. CFG_WAKEUP1_OUT....4695
Table 18-940. Register Call Summary for Register CFG_WAKEUP1_OUT....4696
Table 18-941. CFG_WAKEUP2_IN....4696
Table 18-942. Register Call Summary for Register CFG_WAKEUP2_IN....4696
Table 18-943. CFG_WAKEUP2_OEN....4696
Table 18-944. Register Call Summary for Register CFG_WAKEUP2_OEN....4697
Table 18-945. CFG_WAKEUP2_OUT....4697
Table 18-946. Register Call Summary for Register CFG_WAKEUP2_OUT....4697
Table 18-947. CFG_WAKEUP3_IN....4697
Table 18-948. Register Call Summary for Register CFG_WAKEUP3_IN....4698
Table 18-949. CFG_WAKEUP3_OEN....4698
Table 18-950. Register Call Summary for Register CFG_WAKEUP3_OEN....4698
Table 18-951. CFG_WAKEUP3_OUT....4698
Table 18-952. Register Call Summary for Register CFG_WAKEUP3_OUT....4699
Table 18-953. CFG_DCAN1_RX_IN....4699
Table 18-954. Register Call Summary for Register CFG_DCAN1_RX_IN....4699
Table 18-955. CFG_DCAN1_RX_OEN....4699
Table 18-956. Register Call Summary for Register CFG_DCAN1_RX_OEN....4700
Table 18-957. CFG_DCAN1_RX_OUT....4700
Table 18-958. Register Call Summary for Register CFG_DCAN1_RX_OUT....4700
Table 18-959. CFG_DCAN1_TX_IN....4700
Table 18-960. Register Call Summary for Register CFG_DCAN1_TX_IN....4701
Table 18-961. CFG_DCAN1_TX_OEN....4701
Table 18-962. Register Call Summary for Register CFG_DCAN1_TX_OEN....4701
Table 18-963. CFG_DCAN1_TX_OUT....4701
Table 18-964. Register Call Summary for Register CFG_DCAN1_TX_OUT....4702
Table 18-965. CFG_DCAN2_RX_IN....4702
Table 18-966. Register Call Summary for Register CFG_DCAN2_RX_IN....4702
Table 18-967. CFG_DCAN2_RX_OEN....4702
Table 18-968. Register Call Summary for Register CFG_DCAN2_RX_OEN....4703
Table 18-969. CFG_DCAN2_RX_OUT....4703
Table 18-970. Register Call Summary for Register CFG_DCAN2_RX_OUT....4703
Table 18-971. CFG_DCAN2_TX_IN....4703
Table 18-972. Register Call Summary for Register CFG_DCAN2_TX_IN....4704
Table 18-973. CFG_DCAN2_TX_OEN....4704
Table 18-974. Register Call Summary for Register CFG_DCAN2_TX_OEN....4704
Table 18-975. CFG_DCAN2_TX_OUT....4704
Table 18-976. Register Call Summary for Register CFG_DCAN2_TX_OUT....4705
Table 18-977. CFG_EMU0_IN....4705
Table 18-978. Register Call Summary for Register CFG_EMU0_IN....4705
Table 18-979. CFG_EMU0_OEN....4705
Table 18-980. Register Call Summary for Register CFG_EMU0_OEN....4706
Table 18-981. CFG_EMU0_OUT....4706
Table 18-982. Register Call Summary for Register CFG_EMU0_OUT....4706
Table 18-983. CFG_EMU1_IN....4706
Table 18-984. Register Call Summary for Register CFG_EMU1_IN....4707
Table 18-985. CFG_EMU1_OEN....4707
Table 18-986. Register Call Summary for Register CFG_EMU1_OEN....4707

Table 18-987. CFG_EMU1_OUT....4707
Table 18-988. Register Call Summary for Register CFG_EMU1_OUT....4708
Table 18-989. CFG_EMU2_IN....4708
Table 18-990. Register Call Summary for Register CFG_EMU2_IN....4708
Table 18-991. CFG_EMU2_OEN....4708
Table 18-992. Register Call Summary for Register CFG_EMU2_OEN....4709
Table 18-993. CFG_EMU2_OUT....4709
Table 18-994. Register Call Summary for Register CFG_EMU2_OUT....4709
Table 18-995. CFG_EMU3_IN....4709
Table 18-996. Register Call Summary for Register CFG_EMU3_IN....4710
Table 18-997. CFG_EMU3_OEN....4710
Table 18-998. Register Call Summary for Register CFG_EMU3_OEN....4710
Table 18-999. CFG_EMU3_OUT....4710
Table 18-1000. Register Call Summary for Register CFG_EMU3_OUT....4711
Table 18-1001. CFG_EMU4_IN....4711
Table 18-1002. Register Call Summary for Register CFG_EMU4_IN....4711
Table 18-1003. CFG_EMU4_OEN....4711
Table 18-1004. Register Call Summary for Register CFG_EMU4_OEN....4712
Table 18-1005. CFG_EMU4_OUT....4712
Table 18-1006. Register Call Summary for Register CFG_EMU4_OUT....4712
Table 18-1007. CFG_GPIO6_10_IN....4712
Table 18-1008. Register Call Summary for Register CFG_GPIO6_10_IN....4713
Table 18-1009. CFG_GPIO6_10_OEN....4713
Table 18-1010. Register Call Summary for Register CFG_GPIO6_10_OEN....4713
Table 18-1011. CFG_GPIO6_10_OUT....4713
Table 18-1012. Register Call Summary for Register CFG_GPIO6_10_OUT....4714
Table 18-1013. CFG_GPIO6_11_IN....4714
Table 18-1014. Register Call Summary for Register CFG_GPIO6_11_IN....4714
Table 18-1015. CFG_GPIO6_11_OEN....4714
Table 18-1016. Register Call Summary for Register CFG_GPIO6_11_OEN....4715
Table 18-1017. CFG_GPIO6_11_OUT....4715
Table 18-1018. Register Call Summary for Register CFG_GPIO6_11_OUT....4715
Table 18-1019. CFG_GPIO6_14_IN....4715
Table 18-1020. Register Call Summary for Register CFG_GPIO6_14_IN....4716
Table 18-1021. CFG_GPIO6_14_OEN....4716
Table 18-1022. Register Call Summary for Register CFG_GPIO6_14_OEN....4716
Table 18-1023. CFG_GPIO6_14_OUT....4716
Table 18-1024. Register Call Summary for Register CFG_GPIO6_14_OUT....4717
Table 18-1025. CFG_GPIO6_15_IN....4717
Table 18-1026. Register Call Summary for Register CFG_GPIO6_15_IN....4717
Table 18-1027. CFG_GPIO6_15_OEN....4717
Table 18-1028. Register Call Summary for Register CFG_GPIO6_15_OEN....4718
Table 18-1029. CFG_GPIO6_15_OUT....4718
Table 18-1030. Register Call Summary for Register CFG_GPIO6_15_OUT....4718
Table 18-1031. CFG_GPIO6_16_IN....4718
Table 18-1032. Register Call Summary for Register CFG_GPIO6_16_IN....4719
Table 18-1033. CFG_GPIO6_16_OEN....4719
Table 18-1034. Register Call Summary for Register CFG_GPIO6_16_OEN....4719
Table 18-1035. CFG_GPIO6_16_OUT....4719
Table 18-1036. Register Call Summary for Register CFG_GPIO6_16_OUT....4720
Table 18-1037. CFG_GPMC_A0_IN....4720
Table 18-1038. Register Call Summary for Register CFG_GPMC_A0_IN....4720
Table 18-1039. CFG_GPMC_A0_OEN....4720
Table 18-1040. Register Call Summary for Register CFG_GPMC_A0_OEN....4721
Table 18-1041. CFG_GPMC_A0_OUT....4721
Table 18-1042. Register Call Summary for Register CFG_GPMC_A0_OUT....4721
Table 18-1043. CFG_GPMC_A10_IN....4721
Table 18-1044. Register Call Summary for Register CFG_GPMC_A10_IN....4722
Table 18-1045. CFG_GPMC_A10_OEN....4722
Table 18-1046. Register Call Summary for Register CFG_GPMC_A10_OEN....4722
Table 18-1047. CFG_GPMC_A10_OUT....4722

Table 18-1048. Register Call Summary for Register CFG_GPMC_A10_OUT....4723
Table 18-1049. CFG_GPMC_A11_IN....4723
Table 18-1050. Register Call Summary for Register CFG_GPMC_A11_IN....4723
Table 18-1051. CFG_GPMC_A11_OEN....4723
Table 18-1052. Register Call Summary for Register CFG_GPMC_A11_OEN....4724
Table 18-1053. CFG_GPMC_A11_OUT....4724
Table 18-1054. Register Call Summary for Register CFG_GPMC_A11_OUT....4724
Table 18-1055. CFG_GPMC_A12_IN....4724
Table 18-1056. Register Call Summary for Register CFG_GPMC_A12_IN....4725
Table 18-1057. CFG_GPMC_A12_OEN....4725
Table 18-1058. Register Call Summary for Register CFG_GPMC_A12_OEN....4725
Table 18-1059. CFG_GPMC_A12_OUT....4725
Table 18-1060. Register Call Summary for Register CFG_GPMC_A12_OUT....4726
Table 18-1061. CFG_GPMC_A13_IN....4726
Table 18-1062. Register Call Summary for Register CFG_GPMC_A13_IN....4726
Table 18-1063. CFG_GPMC_A13_OEN....4726
Table 18-1064. Register Call Summary for Register CFG_GPMC_A13_OEN....4727
Table 18-1065. CFG_GPMC_A13_OUT....4727
Table 18-1066. Register Call Summary for Register CFG_GPMC_A13_OUT....4727
Table 18-1067. CFG_GPMC_A14_IN....4727
Table 18-1068. Register Call Summary for Register CFG_GPMC_A14_IN....4728
Table 18-1069. CFG_GPMC_A14_OEN....4728
Table 18-1070. Register Call Summary for Register CFG_GPMC_A14_OEN....4728
Table 18-1071. CFG_GPMC_A14_OUT....4728
Table 18-1072. Register Call Summary for Register CFG_GPMC_A14_OUT....4729
Table 18-1073. CFG_GPMC_A15_IN....4729
Table 18-1074. Register Call Summary for Register CFG_GPMC_A15_IN....4729
Table 18-1075. CFG_GPMC_A15_OEN....4729
Table 18-1076. Register Call Summary for Register CFG_GPMC_A15_OEN....4730
Table 18-1077. CFG_GPMC_A15_OUT....4730
Table 18-1078. Register Call Summary for Register CFG_GPMC_A15_OUT....4730
Table 18-1079. CFG_GPMC_A16_IN....4730
Table 18-1080. Register Call Summary for Register CFG_GPMC_A16_IN....4731
Table 18-1081. CFG_GPMC_A16_OEN....4731
Table 18-1082. Register Call Summary for Register CFG_GPMC_A16_OEN....4731
Table 18-1083. CFG_GPMC_A16_OUT....4731
Table 18-1084. Register Call Summary for Register CFG_GPMC_A16_OUT....4732
Table 18-1085. CFG_GPMC_A17_IN....4732
Table 18-1086. Register Call Summary for Register CFG_GPMC_A17_IN....4732
Table 18-1087. CFG_GPMC_A17_OEN....4732
Table 18-1088. Register Call Summary for Register CFG_GPMC_A17_OEN....4733
Table 18-1089. CFG_GPMC_A17_OUT....4733
Table 18-1090. Register Call Summary for Register CFG_GPMC_A17_OUT....4733
Table 18-1091. CFG_GPMC_A18_IN....4733
Table 18-1092. Register Call Summary for Register CFG_GPMC_A18_IN....4734
Table 18-1093. CFG_GPMC_A18_OEN....4734
Table 18-1094. Register Call Summary for Register CFG_GPMC_A18_OEN....4734
Table 18-1095. CFG_GPMC_A18_OUT....4734
Table 18-1096. Register Call Summary for Register CFG_GPMC_A18_OUT....4735
Table 18-1097. CFG_GPMC_A19_IN....4735
Table 18-1098. Register Call Summary for Register CFG_GPMC_A19_IN....4735
Table 18-1099. CFG_GPMC_A19_OEN....4735
Table 18-1100. Register Call Summary for Register CFG_GPMC_A19_OEN....4736
Table 18-1101. CFG_GPMC_A19_OUT....4736
Table 18-1102. Register Call Summary for Register CFG_GPMC_A19_OUT....4736
Table 18-1103. CFG_GPMC_A1_IN....4736
Table 18-1104. Register Call Summary for Register CFG_GPMC_A1_IN....4737
Table 18-1105. CFG_GPMC_A1_OEN....4737
Table 18-1106. Register Call Summary for Register CFG_GPMC_A1_OEN....4737
Table 18-1107. CFG_GPMC_A1_OUT....4737
Table 18-1108. Register Call Summary for Register CFG_GPMC_A1_OUT....4738

Table 18-1109. CFG_GPMC_A20_IN....4738
Table 18-1110. Register Call Summary for Register CFG_GPMC_A20_IN....4738
Table 18-1111. CFG_GPMC_A20_OEN....4738
Table 18-1112. Register Call Summary for Register CFG_GPMC_A20_OEN....4739
Table 18-1113. CFG_GPMC_A20_OUT....4739
Table 18-1114. Register Call Summary for Register CFG_GPMC_A20_OUT....4739
Table 18-1115. CFG_GPMC_A21_IN....4739
Table 18-1116. Register Call Summary for Register CFG_GPMC_A21_IN....4740
Table 18-1117. CFG_GPMC_A21_OEN....4740
Table 18-1118. Register Call Summary for Register CFG_GPMC_A21_OEN....4740
Table 18-1119. CFG_GPMC_A21_OUT....4740
Table 18-1120. Register Call Summary for Register CFG_GPMC_A21_OUT....4741
Table 18-1121. CFG_GPMC_A22_IN....4741
Table 18-1122. Register Call Summary for Register CFG_GPMC_A22_IN....4741
Table 18-1123. CFG_GPMC_A22_OEN....4741
Table 18-1124. Register Call Summary for Register CFG_GPMC_A22_OEN....4742
Table 18-1125. CFG_GPMC_A22_OUT....4742
Table 18-1126. Register Call Summary for Register CFG_GPMC_A22_OUT....4742
Table 18-1127. CFG_GPMC_A23_IN....4742
Table 18-1128. Register Call Summary for Register CFG_GPMC_A23_IN....4743
Table 18-1129. CFG_GPMC_A23_OEN....4743
Table 18-1130. Register Call Summary for Register CFG_GPMC_A23_OEN....4743
Table 18-1131. CFG_GPMC_A23_OUT....4743
Table 18-1132. Register Call Summary for Register CFG_GPMC_A23_OUT....4744
Table 18-1133. CFG_GPMC_A24_IN....4744
Table 18-1134. Register Call Summary for Register CFG_GPMC_A24_IN....4744
Table 18-1135. CFG_GPMC_A24_OEN....4744
Table 18-1136. Register Call Summary for Register CFG_GPMC_A24_OEN....4745
Table 18-1137. CFG_GPMC_A24_OUT....4745
Table 18-1138. Register Call Summary for Register CFG_GPMC_A24_OUT....4745
Table 18-1139. CFG_GPMC_A25_IN....4745
Table 18-1140. Register Call Summary for Register CFG_GPMC_A25_IN....4746
Table 18-1141. CFG_GPMC_A25_OEN....4746
Table 18-1142. Register Call Summary for Register CFG_GPMC_A25_OEN....4746
Table 18-1143. CFG_GPMC_A25_OUT....4746
Table 18-1144. Register Call Summary for Register CFG_GPMC_A25_OUT....4747
Table 18-1145. CFG_GPMC_A26_IN....4747
Table 18-1146. Register Call Summary for Register CFG_GPMC_A26_IN....4747
Table 18-1147. CFG_GPMC_A26_OEN....4747
Table 18-1148. Register Call Summary for Register CFG_GPMC_A26_OEN....4748
Table 18-1149. CFG_GPMC_A26_OUT....4748
Table 18-1150. Register Call Summary for Register CFG_GPMC_A26_OUT....4748
Table 18-1151. CFG_GPMC_A27_IN....4748
Table 18-1152. Register Call Summary for Register CFG_GPMC_A27_IN....4749
Table 18-1153. CFG_GPMC_A27_OEN....4749
Table 18-1154. Register Call Summary for Register CFG_GPMC_A27_OEN....4749
Table 18-1155. CFG_GPMC_A27_OUT....4749
Table 18-1156. Register Call Summary for Register CFG_GPMC_A27_OUT....4750
Table 18-1157. CFG_GPMC_A2_IN....4750
Table 18-1158. Register Call Summary for Register CFG_GPMC_A2_IN....4750
Table 18-1159. CFG_GPMC_A2_OEN....4750
Table 18-1160. Register Call Summary for Register CFG_GPMC_A2_OEN....4751
Table 18-1161. CFG_GPMC_A2_OUT....4751
Table 18-1162. Register Call Summary for Register CFG_GPMC_A2_OUT....4751
Table 18-1163. CFG_GPMC_A3_IN....4751
Table 18-1164. Register Call Summary for Register CFG_GPMC_A3_IN....4752
Table 18-1165. CFG_GPMC_A3_OEN....4752
Table 18-1166. Register Call Summary for Register CFG_GPMC_A3_OEN....4752
Table 18-1167. CFG_GPMC_A3_OUT....4752
Table 18-1168. Register Call Summary for Register CFG_GPMC_A3_OUT....4753
Table 18-1169. CFG_GPMC_A4_IN....4753

Table 18-1170. Register Call Summary for Register CFG_GPMC_A4_IN........ 4753
Table 18-1171. CFG_GPMC_A4_OEN........ 4753
Table 18-1172. Register Call Summary for Register CFG_GPMC_A4_OEN........ 4754
Table 18-1173. CFG_GPMC_A4_OUT........ 4754
Table 18-1174. Register Call Summary for Register CFG_GPMC_A4_OUT........ 4754
Table 18-1175. CFG_GPMC_A5_IN........ 4754
Table 18-1176. Register Call Summary for Register CFG_GPMC_A5_IN........ 4755
Table 18-1177. CFG_GPMC_A5_OEN........ 4755
Table 18-1178. Register Call Summary for Register CFG_GPMC_A5_OEN........ 4755
Table 18-1179. CFG_GPMC_A5_OUT........ 4755
Table 18-1180. Register Call Summary for Register CFG_GPMC_A5_OUT........ 4756
Table 18-1181. CFG_GPMC_A6_IN........ 4756
Table 18-1182. Register Call Summary for Register CFG_GPMC_A6_IN........ 4756
Table 18-1183. CFG_GPMC_A6_OEN........ 4756
Table 18-1184. Register Call Summary for Register CFG_GPMC_A6_OEN........ 4757
Table 18-1185. CFG_GPMC_A6_OUT........ 4757
Table 18-1186. Register Call Summary for Register CFG_GPMC_A6_OUT........ 4757
Table 18-1187. CFG_GPMC_A7_IN........ 4757
Table 18-1188. Register Call Summary for Register CFG_GPMC_A7_IN........ 4758
Table 18-1189. CFG_GPMC_A7_OEN........ 4758
Table 18-1190. Register Call Summary for Register CFG_GPMC_A7_OEN........ 4758
Table 18-1191. CFG_GPMC_A7_OUT........ 4758
Table 18-1192. Register Call Summary for Register CFG_GPMC_A7_OUT........ 4759
Table 18-1193. CFG_GPMC_A8_IN........ 4759
Table 18-1194. Register Call Summary for Register CFG_GPMC_A8_IN........ 4759
Table 18-1195. CFG_GPMC_A8_OEN........ 4759
Table 18-1196. Register Call Summary for Register CFG_GPMC_A8_OEN........ 4760
Table 18-1197. CFG_GPMC_A8_OUT........ 4760
Table 18-1198. Register Call Summary for Register CFG_GPMC_A8_OUT........ 4760
Table 18-1199. CFG_GPMC_A9_IN........ 4760
Table 18-1200. Register Call Summary for Register CFG_GPMC_A9_IN........ 4761
Table 18-1201. CFG_GPMC_A9_OEN........ 4761
Table 18-1202. Register Call Summary for Register CFG_GPMC_A9_OEN........ 4761
Table 18-1203. CFG_GPMC_A9_OUT........ 4761
Table 18-1204. Register Call Summary for Register CFG_GPMC_A9_OUT........ 4762
Table 18-1205. CFG_GPMC_AD0_IN........ 4762
Table 18-1206. Register Call Summary for Register CFG_GPMC_AD0_IN........ 4762
Table 18-1207. CFG_GPMC_AD0_OEN........ 4762
Table 18-1208. Register Call Summary for Register CFG_GPMC_AD0_OEN........ 4763
Table 18-1209. CFG_GPMC_AD0_OUT........ 4763
Table 18-1210. Register Call Summary for Register CFG_GPMC_AD0_OUT........ 4763
Table 18-1211. CFG_GPMC_AD10_IN........ 4763
Table 18-1212. Register Call Summary for Register CFG_GPMC_AD10_IN........ 4764
Table 18-1213. CFG_GPMC_AD10_OEN........ 4764
Table 18-1214. Register Call Summary for Register CFG_GPMC_AD10_OEN........ 4764
Table 18-1215. CFG_GPMC_AD10_OUT........ 4764
Table 18-1216. Register Call Summary for Register CFG_GPMC_AD10_OUT........ 4765
Table 18-1217. CFG_GPMC_AD11_IN........ 4765
Table 18-1218. Register Call Summary for Register CFG_GPMC_AD11_IN........ 4765
Table 18-1219. CFG_GPMC_AD11_OEN........ 4765
Table 18-1220. Register Call Summary for Register CFG_GPMC_AD11_OEN........ 4766
Table 18-1221. CFG_GPMC_AD11_OUT........ 4766
Table 18-1222. Register Call Summary for Register CFG_GPMC_AD11_OUT........ 4766
Table 18-1223. CFG_GPMC_AD12_IN........ 4766
Table 18-1224. Register Call Summary for Register CFG_GPMC_AD12_IN........ 4767
Table 18-1225. CFG_GPMC_AD12_OEN........ 4767
Table 18-1226. Register Call Summary for Register CFG_GPMC_AD12_OEN........ 4767
Table 18-1227. CFG_GPMC_AD12_OUT........ 4767
Table 18-1228. Register Call Summary for Register CFG_GPMC_AD12_OUT........ 4768
Table 18-1229. CFG_GPMC_AD13_IN........ 4768
Table 18-1230. Register Call Summary for Register CFG_GPMC_AD13_IN........ 4768

Table 18-1231. CFG_GPMC_AD13_OEN....4768
Table 18-1232. Register Call Summary for Register CFG_GPMC_AD13_OEN....4769
Table 18-1233. CFG_GPMC_AD13_OUT....4769
Table 18-1234. Register Call Summary for Register CFG_GPMC_AD13_OUT....4769
Table 18-1235. CFG_GPMC_AD14_IN....4769
Table 18-1236. Register Call Summary for Register CFG_GPMC_AD14_IN....4770
Table 18-1237. CFG_GPMC_AD14_OEN....4770
Table 18-1238. Register Call Summary for Register CFG_GPMC_AD14_OEN....4770
Table 18-1239. CFG_GPMC_AD14_OUT....4770
Table 18-1240. Register Call Summary for Register CFG_GPMC_AD14_OUT....4771
Table 18-1241. CFG_GPMC_AD15_IN....4771
Table 18-1242. Register Call Summary for Register CFG_GPMC_AD15_IN....4771
Table 18-1243. CFG_GPMC_AD15_OEN....4771
Table 18-1244. Register Call Summary for Register CFG_GPMC_AD15_OEN....4772
Table 18-1245. CFG_GPMC_AD15_OUT....4772
Table 18-1246. Register Call Summary for Register CFG_GPMC_AD15_OUT....4772
Table 18-1247. CFG_GPMC_AD1_IN....4772
Table 18-1248. Register Call Summary for Register CFG_GPMC_AD1_IN....4773
Table 18-1249. CFG_GPMC_AD1_OEN....4773
Table 18-1250. Register Call Summary for Register CFG_GPMC_AD1_OEN....4773
Table 18-1251. CFG_GPMC_AD1_OUT....4773
Table 18-1252. Register Call Summary for Register CFG_GPMC_AD1_OUT....4774
Table 18-1253. CFG_GPMC_AD2_IN....4774
Table 18-1254. Register Call Summary for Register CFG_GPMC_AD2_IN....4774
Table 18-1255. CFG_GPMC_AD2_OEN....4774
Table 18-1256. Register Call Summary for Register CFG_GPMC_AD2_OEN....4775
Table 18-1257. CFG_GPMC_AD2_OUT....4775
Table 18-1258. Register Call Summary for Register CFG_GPMC_AD2_OUT....4775
Table 18-1259. CFG_GPMC_AD3_IN....4775
Table 18-1260. Register Call Summary for Register CFG_GPMC_AD3_IN....4776
Table 18-1261. CFG_GPMC_AD3_OEN....4776
Table 18-1262. Register Call Summary for Register CFG_GPMC_AD3_OEN....4776
Table 18-1263. CFG_GPMC_AD3_OUT....4776
Table 18-1264. Register Call Summary for Register CFG_GPMC_AD3_OUT....4777
Table 18-1265. CFG_GPMC_AD4_IN....4777
Table 18-1266. Register Call Summary for Register CFG_GPMC_AD4_IN....4777
Table 18-1267. CFG_GPMC_AD4_OEN....4777
Table 18-1268. Register Call Summary for Register CFG_GPMC_AD4_OEN....4778
Table 18-1269. CFG_GPMC_AD4_OUT....4778
Table 18-1270. Register Call Summary for Register CFG_GPMC_AD4_OUT....4778
Table 18-1271. CFG_GPMC_AD5_IN....4778
Table 18-1272. Register Call Summary for Register CFG_GPMC_AD5_IN....4779
Table 18-1273. CFG_GPMC_AD5_OEN....4779
Table 18-1274. Register Call Summary for Register CFG_GPMC_AD5_OEN....4779
Table 18-1275. CFG_GPMC_AD5_OUT....4779
Table 18-1276. Register Call Summary for Register CFG_GPMC_AD5_OUT....4780
Table 18-1277. CFG_GPMC_AD6_IN....4780
Table 18-1278. Register Call Summary for Register CFG_GPMC_AD6_IN....4780
Table 18-1279. CFG_GPMC_AD6_OEN....4780
Table 18-1280. Register Call Summary for Register CFG_GPMC_AD6_OEN....4781
Table 18-1281. CFG_GPMC_AD6_OUT....4781
Table 18-1282. Register Call Summary for Register CFG_GPMC_AD6_OUT....4781
Table 18-1283. CFG_GPMC_AD7_IN....4781
Table 18-1284. Register Call Summary for Register CFG_GPMC_AD7_IN....4782
Table 18-1285. CFG_GPMC_AD7_OEN....4782
Table 18-1286. Register Call Summary for Register CFG_GPMC_AD7_OEN....4782
Table 18-1287. CFG_GPMC_AD7_OUT....4782
Table 18-1288. Register Call Summary for Register CFG_GPMC_AD7_OUT....4783
Table 18-1289. CFG_GPMC_AD8_IN....4783
Table 18-1290. Register Call Summary for Register CFG_GPMC_AD8_IN....4783
Table 18-1291. CFG_GPMC_AD8_OEN....4783

Table 18-1292. Register Call Summary for Register CFG_GPMC_AD8_OEN....................4784
Table 18-1293. CFG_GPMC_AD8_OUT....................4784
Table 18-1294. Register Call Summary for Register CFG_GPMC_AD8_OUT....................4784
Table 18-1295. CFG_GPMC_AD9_IN....................4784
Table 18-1296. Register Call Summary for Register CFG_GPMC_AD9_IN....................4785
Table 18-1297. CFG_GPMC_AD9_OEN....................4785
Table 18-1298. Register Call Summary for Register CFG_GPMC_AD9_OEN....................4785
Table 18-1299. CFG_GPMC_AD9_OUT....................4785
Table 18-1300. Register Call Summary for Register CFG_GPMC_AD9_OUT....................4786
Table 18-1301. CFG_GPMC_ADVN_ALE_IN....................4786
Table 18-1302. Register Call Summary for Register CFG_GPMC_ADVN_ALE_IN....................4786
Table 18-1303. CFG_GPMC_ADVN_ALE_OEN....................4786
Table 18-1304. Register Call Summary for Register CFG_GPMC_ADVN_ALE_OEN....................4787
Table 18-1305. CFG_GPMC_ADVN_ALE_OUT....................4787
Table 18-1306. Register Call Summary for Register CFG_GPMC_ADVN_ALE_OUT....................4787
Table 18-1307. CFG_GPMC_BEN0_IN....................4787
Table 18-1308. Register Call Summary for Register CFG_GPMC_BEN0_IN....................4788
Table 18-1309. CFG_GPMC_BEN0_OEN....................4788
Table 18-1310. Register Call Summary for Register CFG_GPMC_BEN0_OEN....................4788
Table 18-1311. CFG_GPMC_BEN0_OUT....................4788
Table 18-1312. Register Call Summary for Register CFG_GPMC_BEN0_OUT....................4789
Table 18-1313. CFG_GPMC_BEN1_IN....................4789
Table 18-1314. Register Call Summary for Register CFG_GPMC_BEN1_IN....................4789
Table 18-1315. CFG_GPMC_BEN1_OEN....................4789
Table 18-1316. Register Call Summary for Register CFG_GPMC_BEN1_OEN....................4790
Table 18-1317. CFG_GPMC_BEN1_OUT....................4790
Table 18-1318. Register Call Summary for Register CFG_GPMC_BEN1_OUT....................4790
Table 18-1319. CFG_GPMC_CLK_IN....................4790
Table 18-1320. Register Call Summary for Register CFG_GPMC_CLK_IN....................4791
Table 18-1321. CFG_GPMC_CLK_OEN....................4791
Table 18-1322. Register Call Summary for Register CFG_GPMC_CLK_OEN....................4791
Table 18-1323. CFG_GPMC_CLK_OUT....................4791
Table 18-1324. Register Call Summary for Register CFG_GPMC_CLK_OUT....................4792
Table 18-1325. CFG_GPMC_CS0_IN....................4792
Table 18-1326. Register Call Summary for Register CFG_GPMC_CS0_IN....................4792
Table 18-1327. CFG_GPMC_CS0_OEN....................4792
Table 18-1328. Register Call Summary for Register CFG_GPMC_CS0_OEN....................4793
Table 18-1329. CFG_GPMC_CS0_OUT....................4793
Table 18-1330. Register Call Summary for Register CFG_GPMC_CS0_OUT....................4793
Table 18-1331. CFG_GPMC_CS1_IN....................4793
Table 18-1332. Register Call Summary for Register CFG_GPMC_CS1_IN....................4794
Table 18-1333. CFG_GPMC_CS1_OEN....................4794
Table 18-1334. Register Call Summary for Register CFG_GPMC_CS1_OEN....................4794
Table 18-1335. CFG_GPMC_CS1_OUT....................4794
Table 18-1336. Register Call Summary for Register CFG_GPMC_CS1_OUT....................4795
Table 18-1337. CFG_GPMC_CS2_IN....................4795
Table 18-1338. Register Call Summary for Register CFG_GPMC_CS2_IN....................4795
Table 18-1339. CFG_GPMC_CS2_OEN....................4795
Table 18-1340. Register Call Summary for Register CFG_GPMC_CS2_OEN....................4796
Table 18-1341. CFG_GPMC_CS2_OUT....................4796
Table 18-1342. Register Call Summary for Register CFG_GPMC_CS2_OUT....................4796
Table 18-1343. CFG_GPMC_CS3_IN....................4796
Table 18-1344. Register Call Summary for Register CFG_GPMC_CS3_IN....................4797
Table 18-1345. CFG_GPMC_CS3_OEN....................4797
Table 18-1346. Register Call Summary for Register CFG_GPMC_CS3_OEN....................4797
Table 18-1347. CFG_GPMC_CS3_OUT....................4797
Table 18-1348. Register Call Summary for Register CFG_GPMC_CS3_OUT....................4798
Table 18-1349. CFG_GPMC_OEN_REN_IN....................4798
Table 18-1350. Register Call Summary for Register CFG_GPMC_OEN_REN_IN....................4798
Table 18-1351. CFG_GPMC_OEN_REN_OEN....................4798
Table 18-1352. Register Call Summary for Register CFG_GPMC_OEN_REN_OEN....................4799

Table 18-1353. CFG_GPMC_OEN_REN_OUT....4799
Table 18-1354. Register Call Summary for Register CFG_GPMC_OEN_REN_OUT....4799
Table 18-1355. CFG_GPMC_WAIT0_IN....4799
Table 18-1356. Register Call Summary for Register CFG_GPMC_WAIT0_IN....4800
Table 18-1357. CFG_GPMC_WAIT0_OEN....4800
Table 18-1358. Register Call Summary for Register CFG_GPMC_WAIT0_OEN....4800
Table 18-1359. CFG_GPMC_WAIT0_OUT....4800
Table 18-1360. Register Call Summary for Register CFG_GPMC_WAIT0_OUT....4801
Table 18-1361. CFG_GPMC_WEN_IN....4801
Table 18-1362. Register Call Summary for Register CFG_GPMC_WEN_IN....4801
Table 18-1363. CFG_GPMC_WEN_OEN....4801
Table 18-1364. Register Call Summary for Register CFG_GPMC_WEN_OEN....4802
Table 18-1365. CFG_GPMC_WEN_OUT....4802
Table 18-1366. Register Call Summary for Register CFG_GPMC_WEN_OUT....4802
Table 18-1367. CFG_MCASP1_ACLKR_IN....4802
Table 18-1368. Register Call Summary for Register CFG_MCASP1_ACLKR_IN....4803
Table 18-1369. CFG_MCASP1_ACLKR_OEN....4803
Table 18-1370. Register Call Summary for Register CFG_MCASP1_ACLKR_OEN....4803
Table 18-1371. CFG_MCASP1_ACLKR_OUT....4803
Table 18-1372. Register Call Summary for Register CFG_MCASP1_ACLKR_OUT....4804
Table 18-1373. CFG_MCASP1_ACLKX_IN....4804
Table 18-1374. Register Call Summary for Register CFG_MCASP1_ACLKX_IN....4804
Table 18-1375. CFG_MCASP1_ACLKX_OEN....4804
Table 18-1376. Register Call Summary for Register CFG_MCASP1_ACLKX_OEN....4805
Table 18-1377. CFG_MCASP1_ACLKX_OUT....4805
Table 18-1378. Register Call Summary for Register CFG_MCASP1_ACLKX_OUT....4805
Table 18-1379. CFG_MCASP1_AXR0_IN....4805
Table 18-1380. Register Call Summary for Register CFG_MCASP1_AXR0_IN....4806
Table 18-1381. CFG_MCASP1_AXR0_OEN....4806
Table 18-1382. Register Call Summary for Register CFG_MCASP1_AXR0_OEN....4806
Table 18-1383. CFG_MCASP1_AXR0_OUT....4806
Table 18-1384. Register Call Summary for Register CFG_MCASP1_AXR0_OUT....4807
Table 18-1385. CFG_MCASP1_AXR10_IN....4807
Table 18-1386. Register Call Summary for Register CFG_MCASP1_AXR10_IN....4807
Table 18-1387. CFG_MCASP1_AXR10_OEN....4807
Table 18-1388. Register Call Summary for Register CFG_MCASP1_AXR10_OEN....4808
Table 18-1389. CFG_MCASP1_AXR10_OUT....4808
Table 18-1390. Register Call Summary for Register CFG_MCASP1_AXR10_OUT....4808
Table 18-1391. CFG_MCASP1_AXR11_IN....4808
Table 18-1392. Register Call Summary for Register CFG_MCASP1_AXR11_IN....4809
Table 18-1393. CFG_MCASP1_AXR11_OEN....4809
Table 18-1394. Register Call Summary for Register CFG_MCASP1_AXR11_OEN....4809
Table 18-1395. CFG_MCASP1_AXR11_OUT....4809
Table 18-1396. Register Call Summary for Register CFG_MCASP1_AXR11_OUT....4810
Table 18-1397. CFG_MCASP1_AXR12_IN....4810
Table 18-1398. Register Call Summary for Register CFG_MCASP1_AXR12_IN....4810
Table 18-1399. CFG_MCASP1_AXR12_OEN....4810
Table 18-1400. Register Call Summary for Register CFG_MCASP1_AXR12_OEN....4811
Table 18-1401. CFG_MCASP1_AXR12_OUT....4811
Table 18-1402. Register Call Summary for Register CFG_MCASP1_AXR12_OUT....4811
Table 18-1403. CFG_MCASP1_AXR13_IN....4811
Table 18-1404. Register Call Summary for Register CFG_MCASP1_AXR13_IN....4812
Table 18-1405. CFG_MCASP1_AXR13_OEN....4812
Table 18-1406. Register Call Summary for Register CFG_MCASP1_AXR13_OEN....4812
Table 18-1407. CFG_MCASP1_AXR13_OUT....4812
Table 18-1408. Register Call Summary for Register CFG_MCASP1_AXR13_OUT....4813
Table 18-1409. CFG_MCASP1_AXR14_IN....4813
Table 18-1410. Register Call Summary for Register CFG_MCASP1_AXR14_IN....4813
Table 18-1411. CFG_MCASP1_AXR14_OEN....4813
Table 18-1412. Register Call Summary for Register CFG_MCASP1_AXR14_OEN....4814
Table 18-1413. CFG_MCASP1_AXR14_OUT....4814

Table 18-1414. Register Call Summary for Register CFG_MCASP1_AXR14_OUT....4814
Table 18-1415. CFG_MCASP1_AXR15_IN....4814
Table 18-1416. Register Call Summary for Register CFG_MCASP1_AXR15_IN....4815
Table 18-1417. CFG_MCASP1_AXR15_OEN....4815
Table 18-1418. Register Call Summary for Register CFG_MCASP1_AXR15_OEN....4815
Table 18-1419. CFG_MCASP1_AXR15_OUT....4815
Table 18-1420. Register Call Summary for Register CFG_MCASP1_AXR15_OUT....4816
Table 18-1421. CFG_MCASP1_AXR1_IN....4816
Table 18-1422. Register Call Summary for Register CFG_MCASP1_AXR1_IN....4816
Table 18-1423. CFG_MCASP1_AXR1_OEN....4816
Table 18-1424. Register Call Summary for Register CFG_MCASP1_AXR1_OEN....4817
Table 18-1425. CFG_MCASP1_AXR1_OUT....4817
Table 18-1426. Register Call Summary for Register CFG_MCASP1_AXR1_OUT....4817
Table 18-1427. CFG_MCASP1_AXR2_IN....4817
Table 18-1428. Register Call Summary for Register CFG_MCASP1_AXR2_IN....4818
Table 18-1429. CFG_MCASP1_AXR2_OEN....4818
Table 18-1430. Register Call Summary for Register CFG_MCASP1_AXR2_OEN....4818
Table 18-1431. CFG_MCASP1_AXR2_OUT....4818
Table 18-1432. Register Call Summary for Register CFG_MCASP1_AXR2_OUT....4819
Table 18-1433. CFG_MCASP1_AXR3_IN....4819
Table 18-1434. Register Call Summary for Register CFG_MCASP1_AXR3_IN....4819
Table 18-1435. CFG_MCASP1_AXR3_OEN....4819
Table 18-1436. Register Call Summary for Register CFG_MCASP1_AXR3_OEN....4820
Table 18-1437. CFG_MCASP1_AXR3_OUT....4820
Table 18-1438. Register Call Summary for Register CFG_MCASP1_AXR3_OUT....4820
Table 18-1439. CFG_MCASP1_AXR4_IN....4820
Table 18-1440. Register Call Summary for Register CFG_MCASP1_AXR4_IN....4821
Table 18-1441. CFG_MCASP1_AXR4_OEN....4821
Table 18-1442. Register Call Summary for Register CFG_MCASP1_AXR4_OEN....4821
Table 18-1443. CFG_MCASP1_AXR4_OUT....4821
Table 18-1444. Register Call Summary for Register CFG_MCASP1_AXR4_OUT....4822
Table 18-1445. CFG_MCASP1_AXR5_IN....4822
Table 18-1446. Register Call Summary for Register CFG_MCASP1_AXR5_IN....4822
Table 18-1447. CFG_MCASP1_AXR5_OEN....4822
Table 18-1448. Register Call Summary for Register CFG_MCASP1_AXR5_OEN....4823
Table 18-1449. CFG_MCASP1_AXR5_OUT....4823
Table 18-1450. Register Call Summary for Register CFG_MCASP1_AXR5_OUT....4823
Table 18-1451. CFG_MCASP1_AXR6_IN....4823
Table 18-1452. Register Call Summary for Register CFG_MCASP1_AXR6_IN....4824
Table 18-1453. CFG_MCASP1_AXR6_OEN....4824
Table 18-1454. Register Call Summary for Register CFG_MCASP1_AXR6_OEN....4824
Table 18-1455. CFG_MCASP1_AXR6_OUT....4824
Table 18-1456. Register Call Summary for Register CFG_MCASP1_AXR6_OUT....4825
Table 18-1457. CFG_MCASP1_AXR7_IN....4825
Table 18-1458. Register Call Summary for Register CFG_MCASP1_AXR7_IN....4825
Table 18-1459. CFG_MCASP1_AXR7_OEN....4825
Table 18-1460. Register Call Summary for Register CFG_MCASP1_AXR7_OEN....4826
Table 18-1461. CFG_MCASP1_AXR7_OUT....4826
Table 18-1462. Register Call Summary for Register CFG_MCASP1_AXR7_OUT....4826
Table 18-1463. CFG_MCASP1_AXR8_IN....4826
Table 18-1464. Register Call Summary for Register CFG_MCASP1_AXR8_IN....4827
Table 18-1465. CFG_MCASP1_AXR8_OEN....4827
Table 18-1466. Register Call Summary for Register CFG_MCASP1_AXR8_OEN....4827
Table 18-1467. CFG_MCASP1_AXR8_OUT....4827
Table 18-1468. Register Call Summary for Register CFG_MCASP1_AXR8_OUT....4828
Table 18-1469. CFG_MCASP1_AXR9_IN....4828
Table 18-1470. Register Call Summary for Register CFG_MCASP1_AXR9_IN....4828
Table 18-1471. CFG_MCASP1_AXR9_OEN....4828
Table 18-1472. Register Call Summary for Register CFG_MCASP1_AXR9_OEN....4829
Table 18-1473. CFG_MCASP1_AXR9_OUT....4829
Table 18-1474. Register Call Summary for Register CFG_MCASP1_AXR9_OUT....4829

Table 18-1475. CFG_MCASP1_FSR_IN....4829
Table 18-1476. Register Call Summary for Register CFG_MCASP1_FSR_IN....4830
Table 18-1477. CFG_MCASP1_FSR_OEN....4830
Table 18-1478. Register Call Summary for Register CFG_MCASP1_FSR_OEN....4830
Table 18-1479. CFG_MCASP1_FSR_OUT....4830
Table 18-1480. Register Call Summary for Register CFG_MCASP1_FSR_OUT....4831
Table 18-1481. CFG_MCASP1_FSX_IN....4831
Table 18-1482. Register Call Summary for Register CFG_MCASP1_FSX_IN....4831
Table 18-1483. CFG_MCASP1_FSX_OEN....4831
Table 18-1484. Register Call Summary for Register CFG_MCASP1_FSX_OEN....4832
Table 18-1485. CFG_MCASP1_FSX_OUT....4832
Table 18-1486. Register Call Summary for Register CFG_MCASP1_FSX_OUT....4832
Table 18-1487. CFG_MCASP2_ACLKR_IN....4832
Table 18-1488. Register Call Summary for Register CFG_MCASP2_ACLKR_IN....4833
Table 18-1489. CFG_MCASP2_ACLKR_OEN....4833
Table 18-1490. Register Call Summary for Register CFG_MCASP2_ACLKR_OEN....4833
Table 18-1491. CFG_MCASP2_ACLKR_OUT....4833
Table 18-1492. Register Call Summary for Register CFG_MCASP2_ACLKR_OUT....4834
Table 18-1493. CFG_MCASP2_ACLKX_IN....4834
Table 18-1494. Register Call Summary for Register CFG_MCASP2_ACLKX_IN....4834
Table 18-1495. CFG_MCASP2_ACLKX_OEN....4834
Table 18-1496. Register Call Summary for Register CFG_MCASP2_ACLKX_OEN....4835
Table 18-1497. CFG_MCASP2_ACLKX_OUT....4835
Table 18-1498. Register Call Summary for Register CFG_MCASP2_ACLKX_OUT....4835
Table 18-1499. CFG_MCASP2_AXR0_IN....4835
Table 18-1500. Register Call Summary for Register CFG_MCASP2_AXR0_IN....4836
Table 18-1501. CFG_MCASP2_AXR0_OEN....4836
Table 18-1502. Register Call Summary for Register CFG_MCASP2_AXR0_OEN....4836
Table 18-1503. CFG_MCASP2_AXR0_OUT....4836
Table 18-1504. Register Call Summary for Register CFG_MCASP2_AXR0_OUT....4837
Table 18-1505. CFG_MCASP2_AXR1_IN....4837
Table 18-1506. Register Call Summary for Register CFG_MCASP2_AXR1_IN....4837
Table 18-1507. CFG_MCASP2_AXR1_OEN....4837
Table 18-1508. Register Call Summary for Register CFG_MCASP2_AXR1_OEN....4838
Table 18-1509. CFG_MCASP2_AXR1_OUT....4838
Table 18-1510. Register Call Summary for Register CFG_MCASP2_AXR1_OUT....4838
Table 18-1511. CFG_MCASP2_AXR2_IN....4838
Table 18-1512. Register Call Summary for Register CFG_MCASP2_AXR2_IN....4839
Table 18-1513. CFG_MCASP2_AXR2_OEN....4839
Table 18-1514. Register Call Summary for Register CFG_MCASP2_AXR2_OEN....4839
Table 18-1515. CFG_MCASP2_AXR2_OUT....4839
Table 18-1516. Register Call Summary for Register CFG_MCASP2_AXR2_OUT....4840
Table 18-1517. CFG_MCASP2_AXR3_IN....4840
Table 18-1518. Register Call Summary for Register CFG_MCASP2_AXR3_IN....4840
Table 18-1519. CFG_MCASP2_AXR3_OEN....4840
Table 18-1520. Register Call Summary for Register CFG_MCASP2_AXR3_OEN....4841
Table 18-1521. CFG_MCASP2_AXR3_OUT....4841
Table 18-1522. Register Call Summary for Register CFG_MCASP2_AXR3_OUT....4841
Table 18-1523. CFG_MCASP2_AXR4_IN....4841
Table 18-1524. Register Call Summary for Register CFG_MCASP2_AXR4_IN....4842
Table 18-1525. CFG_MCASP2_AXR4_OEN....4842
Table 18-1526. Register Call Summary for Register CFG_MCASP2_AXR4_OEN....4842
Table 18-1527. CFG_MCASP2_AXR4_OUT....4842
Table 18-1528. Register Call Summary for Register CFG_MCASP2_AXR4_OUT....4843
Table 18-1529. CFG_MCASP2_AXR5_IN....4843
Table 18-1530. Register Call Summary for Register CFG_MCASP2_AXR5_IN....4843
Table 18-1531. CFG_MCASP2_AXR5_OEN....4843
Table 18-1532. Register Call Summary for Register CFG_MCASP2_AXR5_OEN....4844
Table 18-1533. CFG_MCASP2_AXR5_OUT....4844
Table 18-1534. Register Call Summary for Register CFG_MCASP2_AXR5_OUT....4844
Table 18-1535. CFG_MCASP2_AXR6_IN....4844

Table 18-1536. Register Call Summary for Register CFG_MCASP2_AXR6_IN....4845
Table 18-1537. CFG_MCASP2_AXR6_OEN....4845
Table 18-1538. Register Call Summary for Register CFG_MCASP2_AXR6_OEN....4845
Table 18-1539. CFG_MCASP2_AXR6_OUT....4845
Table 18-1540. Register Call Summary for Register CFG_MCASP2_AXR6_OUT....4846
Table 18-1541. CFG_MCASP2_AXR7_IN....4846
Table 18-1542. Register Call Summary for Register CFG_MCASP2_AXR7_IN....4846
Table 18-1543. CFG_MCASP2_AXR7_OEN....4846
Table 18-1544. Register Call Summary for Register CFG_MCASP2_AXR7_OEN....4847
Table 18-1545. CFG_MCASP2_AXR7_OUT....4847
Table 18-1546. Register Call Summary for Register CFG_MCASP2_AXR7_OUT....4847
Table 18-1547. CFG_MCASP2_FSR_IN....4847
Table 18-1548. Register Call Summary for Register CFG_MCASP2_FSR_IN....4848
Table 18-1549. CFG_MCASP2_FSR_OEN....4848
Table 18-1550. Register Call Summary for Register CFG_MCASP2_FSR_OEN....4848
Table 18-1551. CFG_MCASP2_FSR_OUT....4848
Table 18-1552. Register Call Summary for Register CFG_MCASP2_FSR_OUT....4849
Table 18-1553. CFG_MCASP2_FSX_IN....4849
Table 18-1554. Register Call Summary for Register CFG_MCASP2_FSX_IN....4849
Table 18-1555. CFG_MCASP2_FSX_OEN....4849
Table 18-1556. Register Call Summary for Register CFG_MCASP2_FSX_OEN....4850
Table 18-1557. CFG_MCASP2_FSX_OUT....4850
Table 18-1558. Register Call Summary for Register CFG_MCASP2_FSX_OUT....4850
Table 18-1559. CFG_MCASP3_ACLKX_IN....4850
Table 18-1560. Register Call Summary for Register CFG_MCASP3_ACLKX_IN....4851
Table 18-1561. CFG_MCASP3_ACLKX_OEN....4851
Table 18-1562. Register Call Summary for Register CFG_MCASP3_ACLKX_OEN....4851
Table 18-1563. CFG_MCASP3_ACLKX_OUT....4851
Table 18-1564. Register Call Summary for Register CFG_MCASP3_ACLKX_OUT....4852
Table 18-1565. CFG_MCASP3_AXR0_IN....4852
Table 18-1566. Register Call Summary for Register CFG_MCASP3_AXR0_IN....4852
Table 18-1567. CFG_MCASP3_AXR0_OEN....4852
Table 18-1568. Register Call Summary for Register CFG_MCASP3_AXR0_OEN....4853
Table 18-1569. CFG_MCASP3_AXR0_OUT....4853
Table 18-1570. Register Call Summary for Register CFG_MCASP3_AXR0_OUT....4853
Table 18-1571. CFG_MCASP3_AXR1_IN....4853
Table 18-1572. Register Call Summary for Register CFG_MCASP3_AXR1_IN....4854
Table 18-1573. CFG_MCASP3_AXR1_OEN....4854
Table 18-1574. Register Call Summary for Register CFG_MCASP3_AXR1_OEN....4854
Table 18-1575. CFG_MCASP3_AXR1_OUT....4854
Table 18-1576. Register Call Summary for Register CFG_MCASP3_AXR1_OUT....4855
Table 18-1577. CFG_MCASP3_FSX_IN....4855
Table 18-1578. Register Call Summary for Register CFG_MCASP3_FSX_IN....4855
Table 18-1579. CFG_MCASP3_FSX_OEN....4855
Table 18-1580. Register Call Summary for Register CFG_MCASP3_FSX_OEN....4856
Table 18-1581. CFG_MCASP3_FSX_OUT....4856
Table 18-1582. Register Call Summary for Register CFG_MCASP3_FSX_OUT....4856
Table 18-1583. CFG_MCASP4_ACLKX_IN....4856
Table 18-1584. Register Call Summary for Register CFG_MCASP4_ACLKX_IN....4857
Table 18-1585. CFG_MCASP4_ACLKX_OEN....4857
Table 18-1586. Register Call Summary for Register CFG_MCASP4_ACLKX_OEN....4857
Table 18-1587. CFG_MCASP4_ACLKX_OUT....4857
Table 18-1588. Register Call Summary for Register CFG_MCASP4_ACLKX_OUT....4858
Table 18-1589. CFG_MCASP4_AXR0_IN....4858
Table 18-1590. Register Call Summary for Register CFG_MCASP4_AXR0_IN....4858
Table 18-1591. CFG_MCASP4_AXR0_OEN....4858
Table 18-1592. Register Call Summary for Register CFG_MCASP4_AXR0_OEN....4859
Table 18-1593. CFG_MCASP4_AXR0_OUT....4859
Table 18-1594. Register Call Summary for Register CFG_MCASP4_AXR0_OUT....4859
Table 18-1595. CFG_MCASP4_AXR1_IN....4859
Table 18-1596. Register Call Summary for Register CFG_MCASP4_AXR1_IN....4860

Table 18-1597. CFG_MCASP4_AXR1_OEN... 4860
Table 18-1598. Register Call Summary for Register CFG_MCASP4_AXR1_OEN.. 4860
Table 18-1599. CFG_MCASP4_AXR1_OUT... 4860
Table 18-1600. Register Call Summary for Register CFG_MCASP4_AXR1_OUT.. 4861
Table 18-1601. CFG_MCASP4_FSX_IN... 4861
Table 18-1602. Register Call Summary for Register CFG_MCASP4_FSX_IN.. 4861
Table 18-1603. CFG_MCASP4_FSX_OEN... 4861
Table 18-1604. Register Call Summary for Register CFG_MCASP4_FSX_OEN.. 4862
Table 18-1605. CFG_MCASP4_FSX_OUT... 4862
Table 18-1606. Register Call Summary for Register CFG_MCASP4_FSX_OUT.. 4862
Table 18-1607. CFG_MCASP5_ACLKX_IN... 4862
Table 18-1608. Register Call Summary for Register CFG_MCASP5_ACLKX_IN.. 4863
Table 18-1609. CFG_MCASP5_ACLKX_OEN... 4863
Table 18-1610. Register Call Summary for Register CFG_MCASP5_ACLKX_OEN.. 4863
Table 18-1611. CFG_MCASP5_ACLKX_OUT... 4863
Table 18-1612. Register Call Summary for Register CFG_MCASP5_ACLKX_OUT.. 4864
Table 18-1613. CFG_MCASP5_AXR0_IN... 4864
Table 18-1614. Register Call Summary for Register CFG_MCASP5_AXR0_IN.. 4864
Table 18-1615. CFG_MCASP5_AXR0_OEN... 4864
Table 18-1616. Register Call Summary for Register CFG_MCASP5_AXR0_OEN.. 4865
Table 18-1617. CFG_MCASP5_AXR0_OUT... 4865
Table 18-1618. Register Call Summary for Register CFG_MCASP5_AXR0_OUT.. 4865
Table 18-1619. CFG_MCASP5_AXR1_IN... 4865
Table 18-1620. Register Call Summary for Register CFG_MCASP5_AXR1_IN.. 4866
Table 18-1621. CFG_MCASP5_AXR1_OEN... 4866
Table 18-1622. Register Call Summary for Register CFG_MCASP5_AXR1_OEN.. 4866
Table 18-1623. CFG_MCASP5_AXR1_OUT... 4866
Table 18-1624. Register Call Summary for Register CFG_MCASP5_AXR1_OUT.. 4867
Table 18-1625. CFG_MCASP5_FSX_IN... 4867
Table 18-1626. Register Call Summary for Register CFG_MCASP5_FSX_IN.. 4867
Table 18-1627. CFG_MCASP5_FSX_OEN... 4867
Table 18-1628. Register Call Summary for Register CFG_MCASP5_FSX_OEN.. 4868
Table 18-1629. CFG_MCASP5_FSX_OUT... 4868
Table 18-1630. Register Call Summary for Register CFG_MCASP5_FSX_OUT.. 4868
Table 18-1631. CFG_MDIO_D_IN... 4868
Table 18-1632. Register Call Summary for Register CFG_MDIO_D_IN.. 4869
Table 18-1633. CFG_MDIO_D_OEN... 4869
Table 18-1634. Register Call Summary for Register CFG_MDIO_D_OEN.. 4869
Table 18-1635. CFG_MDIO_D_OUT... 4869
Table 18-1636. Register Call Summary for Register CFG_MDIO_D_OUT.. 4870
Table 18-1637. CFG_MDIO_MCLK_IN... 4870
Table 18-1638. Register Call Summary for Register CFG_MDIO_MCLK_IN.. 4870
Table 18-1639. CFG_MDIO_MCLK_OEN... 4870
Table 18-1640. Register Call Summary for Register CFG_MDIO_MCLK_OEN.. 4871
Table 18-1641. CFG_MDIO_MCLK_OUT... 4871
Table 18-1642. Register Call Summary for Register CFG_MDIO_MCLK_OUT.. 4871
Table 18-1643. CFG_MLBP_CLK_N_IN... 4871
Table 18-1644. Register Call Summary for Register CFG_MLBP_CLK_N_IN.. 4872
Table 18-1645. CFG_MLBP_CLK_N_OEN... 4872
Table 18-1646. Register Call Summary for Register CFG_MLBP_CLK_N_OEN.. 4872
Table 18-1647. CFG_MLBP_CLK_N_OUT... 4872
Table 18-1648. Register Call Summary for Register CFG_MLBP_CLK_N_OUT.. 4873
Table 18-1649. CFG_MLBP_CLK_P_IN... 4873
Table 18-1650. Register Call Summary for Register CFG_MLBP_CLK_P_IN.. 4873
Table 18-1651. CFG_MLBP_CLK_P_OEN... 4873
Table 18-1652. Register Call Summary for Register CFG_MLBP_CLK_P_OEN.. 4874
Table 18-1653. CFG_MLBP_CLK_P_OUT... 4874
Table 18-1654. Register Call Summary for Register CFG_MLBP_CLK_P_OUT.. 4874
Table 18-1655. CFG_MLBP_DAT_N_IN... 4874
Table 18-1656. Register Call Summary for Register CFG_MLBP_DAT_N_IN.. 4875
Table 18-1657. CFG_MLBP_DAT_N_OEN... 4875

Table 18-1658. Register Call Summary for Register CFG_MLBP_DAT_N_OEN....4875
Table 18-1659. CFG_MLBP_DAT_N_OUT....4875
Table 18-1660. Register Call Summary for Register CFG_MLBP_DAT_N_OUT....4876
Table 18-1661. CFG_MLBP_DAT_P_IN....4876
Table 18-1662. Register Call Summary for Register CFG_MLBP_DAT_P_IN....4876
Table 18-1663. CFG_MLBP_DAT_P_OEN....4876
Table 18-1664. Register Call Summary for Register CFG_MLBP_DAT_P_OEN....4877
Table 18-1665. CFG_MLBP_DAT_P_OUT....4877
Table 18-1666. Register Call Summary for Register CFG_MLBP_DAT_P_OUT....4877
Table 18-1667. CFG_MLBP_SIG_N_IN....4877
Table 18-1668. Register Call Summary for Register CFG_MLBP_SIG_N_IN....4878
Table 18-1669. CFG_MLBP_SIG_N_OEN....4878
Table 18-1670. Register Call Summary for Register CFG_MLBP_SIG_N_OEN....4878
Table 18-1671. CFG_MLBP_SIG_N_OUT....4878
Table 18-1672. Register Call Summary for Register CFG_MLBP_SIG_N_OUT....4879
Table 18-1673. CFG_MLBP_SIG_P_IN....4879
Table 18-1674. Register Call Summary for Register CFG_MLBP_SIG_P_IN....4879
Table 18-1675. CFG_MLBP_SIG_P_OEN....4879
Table 18-1676. Register Call Summary for Register CFG_MLBP_SIG_P_OEN....4880
Table 18-1677. CFG_MLBP_SIG_P_OUT....4880
Table 18-1678. Register Call Summary for Register CFG_MLBP_SIG_P_OUT....4880
Table 18-1679. CFG_MMC1_CLK_IN....4880
Table 18-1680. Register Call Summary for Register CFG_MMC1_CLK_IN....4881
Table 18-1681. CFG_MMC1_CLK_OEN....4881
Table 18-1682. Register Call Summary for Register CFG_MMC1_CLK_OEN....4881
Table 18-1683. CFG_MMC1_CLK_OUT....4881
Table 18-1684. Register Call Summary for Register CFG_MMC1_CLK_OUT....4882
Table 18-1685. CFG_MMC1_CMD_IN....4882
Table 18-1686. Register Call Summary for Register CFG_MMC1_CMD_IN....4882
Table 18-1687. CFG_MMC1_CMD_OEN....4882
Table 18-1688. Register Call Summary for Register CFG_MMC1_CMD_OEN....4883
Table 18-1689. CFG_MMC1_CMD_OUT....4883
Table 18-1690. Register Call Summary for Register CFG_MMC1_CMD_OUT....4883
Table 18-1691. CFG_MMC1_DAT0_IN....4883
Table 18-1692. Register Call Summary for Register CFG_MMC1_DAT0_IN....4884
Table 18-1693. CFG_MMC1_DAT0_OEN....4884
Table 18-1694. Register Call Summary for Register CFG_MMC1_DAT0_OEN....4884
Table 18-1695. CFG_MMC1_DAT0_OUT....4884
Table 18-1696. Register Call Summary for Register CFG_MMC1_DAT0_OUT....4885
Table 18-1697. CFG_MMC1_DAT1_IN....4885
Table 18-1698. Register Call Summary for Register CFG_MMC1_DAT1_IN....4885
Table 18-1699. CFG_MMC1_DAT1_OEN....4885
Table 18-1700. Register Call Summary for Register CFG_MMC1_DAT1_OEN....4886
Table 18-1701. CFG_MMC1_DAT1_OUT....4886
Table 18-1702. Register Call Summary for Register CFG_MMC1_DAT1_OUT....4886
Table 18-1703. CFG_MMC1_DAT2_IN....4886
Table 18-1704. Register Call Summary for Register CFG_MMC1_DAT2_IN....4887
Table 18-1705. CFG_MMC1_DAT2_OEN....4887
Table 18-1706. Register Call Summary for Register CFG_MMC1_DAT2_OEN....4887
Table 18-1707. CFG_MMC1_DAT2_OUT....4887
Table 18-1708. Register Call Summary for Register CFG_MMC1_DAT2_OUT....4888
Table 18-1709. CFG_MMC1_DAT3_IN....4888
Table 18-1710. Register Call Summary for Register CFG_MMC1_DAT3_IN....4888
Table 18-1711. CFG_MMC1_DAT3_OEN....4888
Table 18-1712. Register Call Summary for Register CFG_MMC1_DAT3_OEN....4889
Table 18-1713. CFG_MMC1_DAT3_OUT....4889
Table 18-1714. Register Call Summary for Register CFG_MMC1_DAT3_OUT....4889
Table 18-1715. CFG_MMC1_SDCD_IN....4889
Table 18-1716. Register Call Summary for Register CFG_MMC1_SDCD_IN....4890
Table 18-1717. CFG_MMC1_SDCD_OEN....4890
Table 18-1718. Register Call Summary for Register CFG_MMC1_SDCD_OEN....4890

Table 18-1719. CFG_MMC1_SDCD_OUT....4890
Table 18-1720. Register Call Summary for Register CFG_MMC1_SDCD_OUT....4891
Table 18-1721. CFG_MMC1_SDWP_IN....4891
Table 18-1722. Register Call Summary for Register CFG_MMC1_SDWP_IN....4891
Table 18-1723. CFG_MMC1_SDWP_OEN....4891
Table 18-1724. Register Call Summary for Register CFG_MMC1_SDWP_OEN....4892
Table 18-1725. CFG_MMC1_SDWP_OUT....4892
Table 18-1726. Register Call Summary for Register CFG_MMC1_SDWP_OUT....4892
Table 18-1727. CFG_MMC3_CLK_IN....4892
Table 18-1728. Register Call Summary for Register CFG_MMC3_CLK_IN....4893
Table 18-1729. CFG_MMC3_CLK_OEN....4893
Table 18-1730. Register Call Summary for Register CFG_MMC3_CLK_OEN....4893
Table 18-1731. CFG_MMC3_CLK_OUT....4893
Table 18-1732. Register Call Summary for Register CFG_MMC3_CLK_OUT....4894
Table 18-1733. CFG_MMC3_CMD_IN....4894
Table 18-1734. Register Call Summary for Register CFG_MMC3_CMD_IN....4894
Table 18-1735. CFG_MMC3_CMD_OEN....4894
Table 18-1736. Register Call Summary for Register CFG_MMC3_CMD_OEN....4895
Table 18-1737. CFG_MMC3_CMD_OUT....4895
Table 18-1738. Register Call Summary for Register CFG_MMC3_CMD_OUT....4895
Table 18-1739. CFG_MMC3_DAT0_IN....4895
Table 18-1740. Register Call Summary for Register CFG_MMC3_DAT0_IN....4896
Table 18-1741. CFG_MMC3_DAT0_OEN....4896
Table 18-1742. Register Call Summary for Register CFG_MMC3_DAT0_OEN....4896
Table 18-1743. CFG_MMC3_DAT0_OUT....4896
Table 18-1744. Register Call Summary for Register CFG_MMC3_DAT0_OUT....4897
Table 18-1745. CFG_MMC3_DAT1_IN....4897
Table 18-1746. Register Call Summary for Register CFG_MMC3_DAT1_IN....4897
Table 18-1747. CFG_MMC3_DAT1_OEN....4897
Table 18-1748. Register Call Summary for Register CFG_MMC3_DAT1_OEN....4898
Table 18-1749. CFG_MMC3_DAT1_OUT....4898
Table 18-1750. Register Call Summary for Register CFG_MMC3_DAT1_OUT....4898
Table 18-1751. CFG_MMC3_DAT2_IN....4898
Table 18-1752. Register Call Summary for Register CFG_MMC3_DAT2_IN....4899
Table 18-1753. CFG_MMC3_DAT2_OEN....4899
Table 18-1754. Register Call Summary for Register CFG_MMC3_DAT2_OEN....4899
Table 18-1755. CFG_MMC3_DAT2_OUT....4899
Table 18-1756. Register Call Summary for Register CFG_MMC3_DAT2_OUT....4900
Table 18-1757. CFG_MMC3_DAT3_IN....4900
Table 18-1758. Register Call Summary for Register CFG_MMC3_DAT3_IN....4900
Table 18-1759. CFG_MMC3_DAT3_OEN....4900
Table 18-1760. Register Call Summary for Register CFG_MMC3_DAT3_OEN....4901
Table 18-1761. CFG_MMC3_DAT3_OUT....4901
Table 18-1762. Register Call Summary for Register CFG_MMC3_DAT3_OUT....4901
Table 18-1763. CFG_MMC3_DAT4_IN....4901
Table 18-1764. Register Call Summary for Register CFG_MMC3_DAT4_IN....4902
Table 18-1765. CFG_MMC3_DAT4_OEN....4902
Table 18-1766. Register Call Summary for Register CFG_MMC3_DAT4_OEN....4902
Table 18-1767. CFG_MMC3_DAT4_OUT....4902
Table 18-1768. Register Call Summary for Register CFG_MMC3_DAT4_OUT....4903
Table 18-1769. CFG_MMC3_DAT5_IN....4903
Table 18-1770. Register Call Summary for Register CFG_MMC3_DAT5_IN....4903
Table 18-1771. CFG_MMC3_DAT5_OEN....4903
Table 18-1772. Register Call Summary for Register CFG_MMC3_DAT5_OEN....4904
Table 18-1773. CFG_MMC3_DAT5_OUT....4904
Table 18-1774. Register Call Summary for Register CFG_MMC3_DAT5_OUT....4904
Table 18-1775. CFG_MMC3_DAT6_IN....4904
Table 18-1776. Register Call Summary for Register CFG_MMC3_DAT6_IN....4905
Table 18-1777. CFG_MMC3_DAT6_OEN....4905
Table 18-1778. Register Call Summary for Register CFG_MMC3_DAT6_OEN....4905
Table 18-1779. CFG_MMC3_DAT6_OUT....4905

Table 18-1780. Register Call Summary for Register CFG_MMC3_DAT6_OUT....4906
Table 18-1781. CFG_MMC3_DAT7_IN....4906
Table 18-1782. Register Call Summary for Register CFG_MMC3_DAT7_IN....4906
Table 18-1783. CFG_MMC3_DAT7_OEN....4906
Table 18-1784. Register Call Summary for Register CFG_MMC3_DAT7_OEN....4907
Table 18-1785. CFG_MMC3_DAT7_OUT....4907
Table 18-1786. Register Call Summary for Register CFG_MMC3_DAT7_OUT....4907
Table 18-1787. CFG_RGMII0_RXC_IN....4907
Table 18-1788. Register Call Summary for Register CFG_RGMII0_RXC_IN....4908
Table 18-1789. CFG_RGMII0_RXC_OEN....4908
Table 18-1790. Register Call Summary for Register CFG_RGMII0_RXC_OEN....4908
Table 18-1791. CFG_RGMII0_RXC_OUT....4908
Table 18-1792. Register Call Summary for Register CFG_RGMII0_RXC_OUT....4909
Table 18-1793. CFG_RGMII0_RXCTL_IN....4909
Table 18-1794. Register Call Summary for Register CFG_RGMII0_RXCTL_IN....4909
Table 18-1795. CFG_RGMII0_RXCTL_OEN....4909
Table 18-1796. Register Call Summary for Register CFG_RGMII0_RXCTL_OEN....4910
Table 18-1797. CFG_RGMII0_RXCTL_OUT....4910
Table 18-1798. Register Call Summary for Register CFG_RGMII0_RXCTL_OUT....4910
Table 18-1799. CFG_RGMII0_RXD0_IN....4910
Table 18-1800. Register Call Summary for Register CFG_RGMII0_RXD0_IN....4911
Table 18-1801. CFG_RGMII0_RXD0_OEN....4911
Table 18-1802. Register Call Summary for Register CFG_RGMII0_RXD0_OEN....4911
Table 18-1803. CFG_RGMII0_RXD0_OUT....4911
Table 18-1804. Register Call Summary for Register CFG_RGMII0_RXD0_OUT....4912
Table 18-1805. CFG_RGMII0_RXD1_IN....4912
Table 18-1806. Register Call Summary for Register CFG_RGMII0_RXD1_IN....4912
Table 18-1807. CFG_RGMII0_RXD1_OEN....4912
Table 18-1808. Register Call Summary for Register CFG_RGMII0_RXD1_OEN....4913
Table 18-1809. CFG_RGMII0_RXD1_OUT....4913
Table 18-1810. Register Call Summary for Register CFG_RGMII0_RXD1_OUT....4913
Table 18-1811. CFG_RGMII0_RXD2_IN....4913
Table 18-1812. Register Call Summary for Register CFG_RGMII0_RXD2_IN....4914
Table 18-1813. CFG_RGMII0_RXD2_OEN....4914
Table 18-1814. Register Call Summary for Register CFG_RGMII0_RXD2_OEN....4914
Table 18-1815. CFG_RGMII0_RXD2_OUT....4914
Table 18-1816. Register Call Summary for Register CFG_RGMII0_RXD2_OUT....4915
Table 18-1817. CFG_RGMII0_RXD3_IN....4915
Table 18-1818. Register Call Summary for Register CFG_RGMII0_RXD3_IN....4915
Table 18-1819. CFG_RGMII0_RXD3_OEN....4915
Table 18-1820. Register Call Summary for Register CFG_RGMII0_RXD3_OEN....4916
Table 18-1821. CFG_RGMII0_RXD3_OUT....4916
Table 18-1822. Register Call Summary for Register CFG_RGMII0_RXD3_OUT....4916
Table 18-1823. CFG_RGMII0_TXC_IN....4916
Table 18-1824. Register Call Summary for Register CFG_RGMII0_TXC_IN....4917
Table 18-1825. CFG_RGMII0_TXC_OEN....4917
Table 18-1826. Register Call Summary for Register CFG_RGMII0_TXC_OEN....4917
Table 18-1827. CFG_RGMII0_TXC_OUT....4917
Table 18-1828. Register Call Summary for Register CFG_RGMII0_TXC_OUT....4918
Table 18-1829. CFG_RGMII0_TXCTL_IN....4918
Table 18-1830. Register Call Summary for Register CFG_RGMII0_TXCTL_IN....4918
Table 18-1831. CFG_RGMII0_TXCTL_OEN....4918
Table 18-1832. Register Call Summary for Register CFG_RGMII0_TXCTL_OEN....4919
Table 18-1833. CFG_RGMII0_TXCTL_OUT....4919
Table 18-1834. Register Call Summary for Register CFG_RGMII0_TXCTL_OUT....4919
Table 18-1835. CFG_RGMII0_TXD0_IN....4919
Table 18-1836. Register Call Summary for Register CFG_RGMII0_TXD0_IN....4920
Table 18-1837. CFG_RGMII0_TXD0_OEN....4920
Table 18-1838. Register Call Summary for Register CFG_RGMII0_TXD0_OEN....4920
Table 18-1839. CFG_RGMII0_TXD0_OUT....4920
Table 18-1840. Register Call Summary for Register CFG_RGMII0_TXD0_OUT....4921

Table 18-1841. CFG_RGMII0_TXD1_IN....4921
Table 18-1842. Register Call Summary for Register CFG_RGMII0_TXD1_IN....4921
Table 18-1843. CFG_RGMII0_TXD1_OEN....4921
Table 18-1844. Register Call Summary for Register CFG_RGMII0_TXD1_OEN....4922
Table 18-1845. CFG_RGMII0_TXD1_OUT....4922
Table 18-1846. Register Call Summary for Register CFG_RGMII0_TXD1_OUT....4922
Table 18-1847. CFG_RGMII0_TXD2_IN....4922
Table 18-1848. Register Call Summary for Register CFG_RGMII0_TXD2_IN....4923
Table 18-1849. CFG_RGMII0_TXD2_OEN....4923
Table 18-1850. Register Call Summary for Register CFG_RGMII0_TXD2_OEN....4923
Table 18-1851. CFG_RGMII0_TXD2_OUT....4923
Table 18-1852. Register Call Summary for Register CFG_RGMII0_TXD2_OUT....4924
Table 18-1853. CFG_RGMII0_TXD3_IN....4924
Table 18-1854. Register Call Summary for Register CFG_RGMII0_TXD3_IN....4924
Table 18-1855. CFG_RGMII0_TXD3_OEN....4924
Table 18-1856. Register Call Summary for Register CFG_RGMII0_TXD3_OEN....4925
Table 18-1857. CFG_RGMII0_TXD3_OUT....4925
Table 18-1858. Register Call Summary for Register CFG_RGMII0_TXD3_OUT....4925
Table 18-1859. CFG_RTCK_IN....4925
Table 18-1860. Register Call Summary for Register CFG_RTCK_IN....4926
Table 18-1861. CFG_RTCK_OEN....4926
Table 18-1862. Register Call Summary for Register CFG_RTCK_OEN....4926
Table 18-1863. CFG_RTCK_OUT....4926
Table 18-1864. Register Call Summary for Register CFG_RTCK_OUT....4927
Table 18-1865. CFG_SPI1_CS0_IN....4927
Table 18-1866. Register Call Summary for Register CFG_SPI1_CS0_IN....4927
Table 18-1867. CFG_SPI1_CS0_OEN....4927
Table 18-1868. Register Call Summary for Register CFG_SPI1_CS0_OEN....4928
Table 18-1869. CFG_SPI1_CS0_OUT....4928
Table 18-1870. Register Call Summary for Register CFG_SPI1_CS0_OUT....4928
Table 18-1871. CFG_SPI1_CS1_IN....4928
Table 18-1872. Register Call Summary for Register CFG_SPI1_CS1_IN....4929
Table 18-1873. CFG_SPI1_CS1_OEN....4929
Table 18-1874. Register Call Summary for Register CFG_SPI1_CS1_OEN....4929
Table 18-1875. CFG_SPI1_CS1_OUT....4929
Table 18-1876. Register Call Summary for Register CFG_SPI1_CS1_OUT....4930
Table 18-1877. CFG_SPI1_CS2_IN....4930
Table 18-1878. Register Call Summary for Register CFG_SPI1_CS2_IN....4930
Table 18-1879. CFG_SPI1_CS2_OEN....4930
Table 18-1880. Register Call Summary for Register CFG_SPI1_CS2_OEN....4931
Table 18-1881. CFG_SPI1_CS2_OUT....4931
Table 18-1882. Register Call Summary for Register CFG_SPI1_CS2_OUT....4931
Table 18-1883. CFG_SPI1_CS3_IN....4931
Table 18-1884. Register Call Summary for Register CFG_SPI1_CS3_IN....4932
Table 18-1885. CFG_SPI1_CS3_OEN....4932
Table 18-1886. Register Call Summary for Register CFG_SPI1_CS3_OEN....4932
Table 18-1887. CFG_SPI1_CS3_OUT....4932
Table 18-1888. Register Call Summary for Register CFG_SPI1_CS3_OUT....4933
Table 18-1889. CFG_SPI1_D0_IN....4933
Table 18-1890. Register Call Summary for Register CFG_SPI1_D0_IN....4933
Table 18-1891. CFG_SPI1_D0_OEN....4933
Table 18-1892. Register Call Summary for Register CFG_SPI1_D0_OEN....4934
Table 18-1893. CFG_SPI1_D0_OUT....4934
Table 18-1894. Register Call Summary for Register CFG_SPI1_D0_OUT....4934
Table 18-1895. CFG_SPI1_D1_IN....4934
Table 18-1896. Register Call Summary for Register CFG_SPI1_D1_IN....4935
Table 18-1897. CFG_SPI1_D1_OEN....4935
Table 18-1898. Register Call Summary for Register CFG_SPI1_D1_OEN....4935
Table 18-1899. CFG_SPI1_D1_OUT....4935
Table 18-1900. Register Call Summary for Register CFG_SPI1_D1_OUT....4936
Table 18-1901. CFG_SPI1_SCLK_IN....4936

Table 18-1902. Register Call Summary for Register CFG_SPI1_SCLK_IN........ 4936
Table 18-1903. CFG_SPI1_SCLK_OEN........ 4936
Table 18-1904. Register Call Summary for Register CFG_SPI1_SCLK_OEN........ 4937
Table 18-1905. CFG_SPI1_SCLK_OUT........ 4937
Table 18-1906. Register Call Summary for Register CFG_SPI1_SCLK_OUT........ 4937
Table 18-1907. CFG_SPI2_CS0_IN........ 4937
Table 18-1908. Register Call Summary for Register CFG_SPI2_CS0_IN........ 4938
Table 18-1909. CFG_SPI2_CS0_OEN........ 4938
Table 18-1910. Register Call Summary for Register CFG_SPI2_CS0_OEN........ 4938
Table 18-1911. CFG_SPI2_CS0_OUT........ 4938
Table 18-1912. Register Call Summary for Register CFG_SPI2_CS0_OUT........ 4939
Table 18-1913. CFG_SPI2_D0_IN........ 4939
Table 18-1914. Register Call Summary for Register CFG_SPI2_D0_IN........ 4939
Table 18-1915. CFG_SPI2_D0_OEN........ 4939
Table 18-1916. Register Call Summary for Register CFG_SPI2_D0_OEN........ 4940
Table 18-1917. CFG_SPI2_D0_OUT........ 4940
Table 18-1918. Register Call Summary for Register CFG_SPI2_D0_OUT........ 4940
Table 18-1919. CFG_SPI2_D1_IN........ 4940
Table 18-1920. Register Call Summary for Register CFG_SPI2_D1_IN........ 4941
Table 18-1921. CFG_SPI2_D1_OEN........ 4941
Table 18-1922. Register Call Summary for Register CFG_SPI2_D1_OEN........ 4941
Table 18-1923. CFG_SPI2_D1_OUT........ 4941
Table 18-1924. Register Call Summary for Register CFG_SPI2_D1_OUT........ 4942
Table 18-1925. CFG_SPI2_SCLK_IN........ 4942
Table 18-1926. Register Call Summary for Register CFG_SPI2_SCLK_IN........ 4942
Table 18-1927. CFG_SPI2_SCLK_OEN........ 4942
Table 18-1928. Register Call Summary for Register CFG_SPI2_SCLK_OEN........ 4943
Table 18-1929. CFG_SPI2_SCLK_OUT........ 4943
Table 18-1930. Register Call Summary for Register CFG_SPI2_SCLK_OUT........ 4943
Table 18-1931. CFG_TDI_IN........ 4943
Table 18-1932. Register Call Summary for Register CFG_TDI_IN........ 4944
Table 18-1933. CFG_TDI_OEN........ 4944
Table 18-1934. Register Call Summary for Register CFG_TDI_OEN........ 4944
Table 18-1935. CFG_TDI_OUT........ 4944
Table 18-1936. Register Call Summary for Register CFG_TDI_OUT........ 4945
Table 18-1937. CFG_TDO_IN........ 4945
Table 18-1938. Register Call Summary for Register CFG_TDO_IN........ 4945
Table 18-1939. CFG_TDO_OEN........ 4945
Table 18-1940. Register Call Summary for Register CFG_TDO_OEN........ 4946
Table 18-1941. CFG_TDO_OUT........ 4946
Table 18-1942. Register Call Summary for Register CFG_TDO_OUT........ 4946
Table 18-1943. CFG_TMS_IN........ 4946
Table 18-1944. Register Call Summary for Register CFG_TMS_IN........ 4947
Table 18-1945. CFG_TMS_OEN........ 4947
Table 18-1946. Register Call Summary for Register CFG_TMS_OEN........ 4947
Table 18-1947. CFG_TMS_OUT........ 4947
Table 18-1948. Register Call Summary for Register CFG_TMS_OUT........ 4948
Table 18-1949. CFG_TRSTN_IN........ 4948
Table 18-1950. Register Call Summary for Register CFG_TRSTN_IN........ 4948
Table 18-1951. CFG_TRSTN_OEN........ 4948
Table 18-1952. Register Call Summary for Register CFG_TRSTN_OEN........ 4949
Table 18-1953. CFG_TRSTN_OUT........ 4949
Table 18-1954. Register Call Summary for Register CFG_TRSTN_OUT........ 4949
Table 18-1955. CFG_UART1_CTSN_IN........ 4949
Table 18-1956. Register Call Summary for Register CFG_UART1_CTSN_IN........ 4950
Table 18-1957. CFG_UART1_CTSN_OEN........ 4950
Table 18-1958. Register Call Summary for Register CFG_UART1_CTSN_OEN........ 4950
Table 18-1959. CFG_UART1_CTSN_OUT........ 4950
Table 18-1960. Register Call Summary for Register CFG_UART1_CTSN_OUT........ 4951
Table 18-1961. CFG_UART1_RTSN_IN........ 4951
Table 18-1962. Register Call Summary for Register CFG_UART1_RTSN_IN........ 4951

Table 18-1963. CFG_UART1_RTSN_OEN....4951
Table 18-1964. Register Call Summary for Register CFG_UART1_RTSN_OEN....4952
Table 18-1965. CFG_UART1_RTSN_OUT....4952
Table 18-1966. Register Call Summary for Register CFG_UART1_RTSN_OUT....4952
Table 18-1967. CFG_UART1_RXD_IN....4952
Table 18-1968. Register Call Summary for Register CFG_UART1_RXD_IN....4953
Table 18-1969. CFG_UART1_RXD_OEN....4953
Table 18-1970. Register Call Summary for Register CFG_UART1_RXD_OEN....4953
Table 18-1971. CFG_UART1_RXD_OUT....4953
Table 18-1972. Register Call Summary for Register CFG_UART1_RXD_OUT....4954
Table 18-1973. CFG_UART1_TXD_IN....4954
Table 18-1974. Register Call Summary for Register CFG_UART1_TXD_IN....4954
Table 18-1975. CFG_UART1_TXD_OEN....4954
Table 18-1976. Register Call Summary for Register CFG_UART1_TXD_OEN....4955
Table 18-1977. CFG_UART1_TXD_OUT....4955
Table 18-1978. Register Call Summary for Register CFG_UART1_TXD_OUT....4955
Table 18-1979. CFG_UART2_CTSN_IN....4955
Table 18-1980. Register Call Summary for Register CFG_UART2_CTSN_IN....4956
Table 18-1981. CFG_UART2_CTSN_OEN....4956
Table 18-1982. Register Call Summary for Register CFG_UART2_CTSN_OEN....4956
Table 18-1983. CFG_UART2_CTSN_OUT....4956
Table 18-1984. Register Call Summary for Register CFG_UART2_CTSN_OUT....4957
Table 18-1985. CFG_UART2_RTSN_IN....4957
Table 18-1986. Register Call Summary for Register CFG_UART2_RTSN_IN....4957
Table 18-1987. CFG_UART2_RTSN_OEN....4957
Table 18-1988. Register Call Summary for Register CFG_UART2_RTSN_OEN....4958
Table 18-1989. CFG_UART2_RTSN_OUT....4958
Table 18-1990. Register Call Summary for Register CFG_UART2_RTSN_OUT....4958
Table 18-1991. CFG_UART2_RXD_IN....4958
Table 18-1992. Register Call Summary for Register CFG_UART2_RXD_IN....4959
Table 18-1993. CFG_UART2_RXD_OEN....4959
Table 18-1994. Register Call Summary for Register CFG_UART2_RXD_OEN....4959
Table 18-1995. CFG_UART2_RXD_OUT....4959
Table 18-1996. Register Call Summary for Register CFG_UART2_RXD_OUT....4960
Table 18-1997. CFG_UART2_TXD_IN....4960
Table 18-1998. Register Call Summary for Register CFG_UART2_TXD_IN....4960
Table 18-1999. CFG_UART2_TXD_OEN....4960
Table 18-2000. Register Call Summary for Register CFG_UART2_TXD_OEN....4961
Table 18-2001. CFG_UART2_TXD_OUT....4961
Table 18-2002. Register Call Summary for Register CFG_UART2_TXD_OUT....4961
Table 18-2003. CFG_UART3_RXD_IN....4961
Table 18-2004. Register Call Summary for Register CFG_UART3_RXD_IN....4962
Table 18-2005. CFG_UART3_RXD_OEN....4962
Table 18-2006. Register Call Summary for Register CFG_UART3_RXD_OEN....4962
Table 18-2007. CFG_UART3_RXD_OUT....4962
Table 18-2008. Register Call Summary for Register CFG_UART3_RXD_OUT....4963
Table 18-2009. CFG_UART3_TXD_IN....4963
Table 18-2010. Register Call Summary for Register CFG_UART3_TXD_IN....4963
Table 18-2011. CFG_UART3_TXD_OEN....4963
Table 18-2012. Register Call Summary for Register CFG_UART3_TXD_OEN....4964
Table 18-2013. CFG_UART3_TXD_OUT....4964
Table 18-2014. Register Call Summary for Register CFG_UART3_TXD_OUT....4964
Table 18-2015. CFG_USB1_DRVVBUS_IN....4964
Table 18-2016. Register Call Summary for Register CFG_USB1_DRVVBUS_IN....4965
Table 18-2017. CFG_USB1_DRVVBUS_OEN....4965
Table 18-2018. Register Call Summary for Register CFG_USB1_DRVVBUS_OEN....4965
Table 18-2019. CFG_USB1_DRVVBUS_OUT....4965
Table 18-2020. Register Call Summary for Register CFG_USB1_DRVVBUS_OUT....4966
Table 18-2021. CFG_USB2_DRVVBUS_IN....4966
Table 18-2022. Register Call Summary for Register CFG_USB2_DRVVBUS_IN....4966
Table 18-2023. CFG_USB2_DRVVBUS_OEN....4966

Table 18-2024. Register Call Summary for Register CFG_USB2_DRVVBUS_OEN........ 4967
Table 18-2025. CFG_USB2_DRVVBUS_OUT........ 4967
Table 18-2026. Register Call Summary for Register CFG_USB2_DRVVBUS_OUT........4967
Table 18-2027. CFG_VIN1A_CLK0_IN........ 4967
Table 18-2028. Register Call Summary for Register CFG_VIN1A_CLK0_IN........4968
Table 18-2029. CFG_VIN1A_CLK0_OEN........ 4968
Table 18-2030. Register Call Summary for Register CFG_VIN1A_CLK0_OEN........4968
Table 18-2031. CFG_VIN1A_CLK0_OUT........ 4968
Table 18-2032. Register Call Summary for Register CFG_VIN1A_CLK0_OUT........4969
Table 18-2033. CFG_VIN1A_D0_IN........4969
Table 18-2034. Register Call Summary for Register CFG_VIN1A_D0_IN........ 4969
Table 18-2035. CFG_VIN1A_D0_OEN........4969
Table 18-2036. Register Call Summary for Register CFG_VIN1A_D0_OEN........ 4970
Table 18-2037. CFG_VIN1A_D0_OUT........4970
Table 18-2038. Register Call Summary for Register CFG_VIN1A_D0_OUT........ 4970
Table 18-2039. CFG_VIN1A_D10_IN........4970
Table 18-2040. Register Call Summary for Register CFG_VIN1A_D10_IN........ 4971
Table 18-2041. CFG_VIN1A_D10_OEN........4971
Table 18-2042. Register Call Summary for Register CFG_VIN1A_D10_OEN........ 4971
Table 18-2043. CFG_VIN1A_D10_OUT........4971
Table 18-2044. Register Call Summary for Register CFG_VIN1A_D10_OUT........ 4972
Table 18-2045. CFG_VIN1A_D11_IN........ 4972
Table 18-2046. Register Call Summary for Register CFG_VIN1A_D11_IN........4972
Table 18-2047. CFG_VIN1A_D11_OEN........4972
Table 18-2048. Register Call Summary for Register CFG_VIN1A_D11_OEN........ 4973
Table 18-2049. CFG_VIN1A_D11_OUT........ 4973
Table 18-2050. Register Call Summary for Register CFG_VIN1A_D11_OUT........4973
Table 18-2051. CFG_VIN1A_D12_IN........4973
Table 18-2052. Register Call Summary for Register CFG_VIN1A_D12_IN........ 4974
Table 18-2053. CFG_VIN1A_D12_OEN........4974
Table 18-2054. Register Call Summary for Register CFG_VIN1A_D12_OEN........4974
Table 18-2055. CFG_VIN1A_D12_OUT........4974
Table 18-2056. Register Call Summary for Register CFG_VIN1A_D12_OUT........ 4975
Table 18-2057. CFG_VIN1A_D13_IN........4975
Table 18-2058. Register Call Summary for Register CFG_VIN1A_D13_IN........ 4975
Table 18-2059. CFG_VIN1A_D13_OEN........4975
Table 18-2060. Register Call Summary for Register CFG_VIN1A_D13_OEN........4976
Table 18-2061. CFG_VIN1A_D13_OUT........4976
Table 18-2062. Register Call Summary for Register CFG_VIN1A_D13_OUT........ 4976
Table 18-2063. CFG_VIN1A_D14_IN........4976
Table 18-2064. Register Call Summary for Register CFG_VIN1A_D14_IN........ 4977
Table 18-2065. CFG_VIN1A_D14_OEN........4977
Table 18-2066. Register Call Summary for Register CFG_VIN1A_D14_OEN........4977
Table 18-2067. CFG_VIN1A_D14_OUT........4977
Table 18-2068. Register Call Summary for Register CFG_VIN1A_D14_OUT........ 4978
Table 18-2069. CFG_VIN1A_D15_IN........4978
Table 18-2070. Register Call Summary for Register CFG_VIN1A_D15_IN........ 4978
Table 18-2071. CFG_VIN1A_D15_OEN........4978
Table 18-2072. Register Call Summary for Register CFG_VIN1A_D15_OEN........4979
Table 18-2073. CFG_VIN1A_D15_OUT........4979
Table 18-2074. Register Call Summary for Register CFG_VIN1A_D15_OUT........ 4979
Table 18-2075. CFG_VIN1A_D16_IN........4979
Table 18-2076. Register Call Summary for Register CFG_VIN1A_D16_IN........ 4980
Table 18-2077. CFG_VIN1A_D16_OEN........4980
Table 18-2078. Register Call Summary for Register CFG_VIN1A_D16_OEN........4980
Table 18-2079. CFG_VIN1A_D16_OUT........4980
Table 18-2080. Register Call Summary for Register CFG_VIN1A_D16_OUT........ 4981
Table 18-2081. CFG_VIN1A_D17_IN........4981
Table 18-2082. Register Call Summary for Register CFG_VIN1A_D17_IN........ 4981
Table 18-2083. CFG_VIN1A_D17_OEN........4981
Table 18-2084. Register Call Summary for Register CFG_VIN1A_D17_OEN........4982

Table 18-2085. CFG_VIN1A_D17_OUT....4982
Table 18-2086. Register Call Summary for Register CFG_VIN1A_D17_OUT....4982
Table 18-2087. CFG_VIN1A_D18_IN....4982
Table 18-2088. Register Call Summary for Register CFG_VIN1A_D18_IN....4983
Table 18-2089. CFG_VIN1A_D18_OEN....4983
Table 18-2090. Register Call Summary for Register CFG_VIN1A_D18_OEN....4983
Table 18-2091. CFG_VIN1A_D18_OUT....4983
Table 18-2092. Register Call Summary for Register CFG_VIN1A_D18_OUT....4984
Table 18-2093. CFG_VIN1A_D19_IN....4984
Table 18-2094. Register Call Summary for Register CFG_VIN1A_D19_IN....4984
Table 18-2095. CFG_VIN1A_D19_OEN....4984
Table 18-2096. Register Call Summary for Register CFG_VIN1A_D19_OEN....4985
Table 18-2097. CFG_VIN1A_D19_OUT....4985
Table 18-2098. Register Call Summary for Register CFG_VIN1A_D19_OUT....4985
Table 18-2099. CFG_VIN1A_D1_IN....4985
Table 18-2100. Register Call Summary for Register CFG_VIN1A_D1_IN....4986
Table 18-2101. CFG_VIN1A_D1_OEN....4986
Table 18-2102. Register Call Summary for Register CFG_VIN1A_D1_OEN....4986
Table 18-2103. CFG_VIN1A_D1_OUT....4986
Table 18-2104. Register Call Summary for Register CFG_VIN1A_D1_OUT....4987
Table 18-2105. CFG_VIN1A_D20_IN....4987
Table 18-2106. Register Call Summary for Register CFG_VIN1A_D20_IN....4987
Table 18-2107. CFG_VIN1A_D20_OEN....4987
Table 18-2108. Register Call Summary for Register CFG_VIN1A_D20_OEN....4988
Table 18-2109. CFG_VIN1A_D20_OUT....4988
Table 18-2110. Register Call Summary for Register CFG_VIN1A_D20_OUT....4988
Table 18-2111. CFG_VIN1A_D21_IN....4988
Table 18-2112. Register Call Summary for Register CFG_VIN1A_D21_IN....4989
Table 18-2113. CFG_VIN1A_D21_OEN....4989
Table 18-2114. Register Call Summary for Register CFG_VIN1A_D21_OEN....4989
Table 18-2115. CFG_VIN1A_D21_OUT....4989
Table 18-2116. Register Call Summary for Register CFG_VIN1A_D21_OUT....4990
Table 18-2117. CFG_VIN1A_D22_IN....4990
Table 18-2118. Register Call Summary for Register CFG_VIN1A_D22_IN....4990
Table 18-2119. CFG_VIN1A_D22_OEN....4990
Table 18-2120. Register Call Summary for Register CFG_VIN1A_D22_OEN....4991
Table 18-2121. CFG_VIN1A_D22_OUT....4991
Table 18-2122. Register Call Summary for Register CFG_VIN1A_D22_OUT....4991
Table 18-2123. CFG_VIN1A_D23_IN....4991
Table 18-2124. Register Call Summary for Register CFG_VIN1A_D23_IN....4992
Table 18-2125. CFG_VIN1A_D23_OEN....4992
Table 18-2126. Register Call Summary for Register CFG_VIN1A_D23_OEN....4992
Table 18-2127. CFG_VIN1A_D23_OUT....4992
Table 18-2128. Register Call Summary for Register CFG_VIN1A_D23_OUT....4993
Table 18-2129. CFG_VIN1A_D2_IN....4993
Table 18-2130. Register Call Summary for Register CFG_VIN1A_D2_IN....4993
Table 18-2131. CFG_VIN1A_D2_OEN....4993
Table 18-2132. Register Call Summary for Register CFG_VIN1A_D2_OEN....4994
Table 18-2133. CFG_VIN1A_D2_OUT....4994
Table 18-2134. Register Call Summary for Register CFG_VIN1A_D2_OUT....4994
Table 18-2135. CFG_VIN1A_D3_IN....4994
Table 18-2136. Register Call Summary for Register CFG_VIN1A_D3_IN....4995
Table 18-2137. CFG_VIN1A_D3_OEN....4995
Table 18-2138. Register Call Summary for Register CFG_VIN1A_D3_OEN....4995
Table 18-2139. CFG_VIN1A_D3_OUT....4995
Table 18-2140. Register Call Summary for Register CFG_VIN1A_D3_OUT....4996
Table 18-2141. CFG_VIN1A_D4_IN....4996
Table 18-2142. Register Call Summary for Register CFG_VIN1A_D4_IN....4996
Table 18-2143. CFG_VIN1A_D4_OEN....4996
Table 18-2144. Register Call Summary for Register CFG_VIN1A_D4_OEN....4997
Table 18-2145. CFG_VIN1A_D4_OUT....4997

Table 18-2146. Register Call Summary for Register CFG_VIN1A_D4_OUT........ 4997
Table 18-2147. CFG_VIN1A_D5_IN........ 4997
Table 18-2148. Register Call Summary for Register CFG_VIN1A_D5_IN........ 4998
Table 18-2149. CFG_VIN1A_D5_OEN........ 4998
Table 18-2150. Register Call Summary for Register CFG_VIN1A_D5_OEN........ 4998
Table 18-2151. CFG_VIN1A_D5_OUT........ 4998
Table 18-2152. Register Call Summary for Register CFG_VIN1A_D5_OUT........ 4999
Table 18-2153. CFG_VIN1A_D6_IN........ 4999
Table 18-2154. Register Call Summary for Register CFG_VIN1A_D6_IN........ 4999
Table 18-2155. CFG_VIN1A_D6_OEN........ 4999
Table 18-2156. Register Call Summary for Register CFG_VIN1A_D6_OEN........ 5000
Table 18-2157. CFG_VIN1A_D6_OUT........ 5000
Table 18-2158. Register Call Summary for Register CFG_VIN1A_D6_OUT........ 5000
Table 18-2159. CFG_VIN1A_D7_IN........ 5000
Table 18-2160. Register Call Summary for Register CFG_VIN1A_D7_IN........ 5001
Table 18-2161. CFG_VIN1A_D7_OEN........ 5001
Table 18-2162. Register Call Summary for Register CFG_VIN1A_D7_OEN........ 5001
Table 18-2163. CFG_VIN1A_D7_OUT........ 5001
Table 18-2164. Register Call Summary for Register CFG_VIN1A_D7_OUT........ 5002
Table 18-2165. CFG_VIN1A_D8_IN........ 5002
Table 18-2166. Register Call Summary for Register CFG_VIN1A_D8_IN........ 5002
Table 18-2167. CFG_VIN1A_D8_OEN........ 5002
Table 18-2168. Register Call Summary for Register CFG_VIN1A_D8_OEN........ 5003
Table 18-2169. CFG_VIN1A_D8_OUT........ 5003
Table 18-2170. Register Call Summary for Register CFG_VIN1A_D8_OUT........ 5003
Table 18-2171. CFG_VIN1A_D9_IN........ 5003
Table 18-2172. Register Call Summary for Register CFG_VIN1A_D9_IN........ 5004
Table 18-2173. CFG_VIN1A_D9_OEN........ 5004
Table 18-2174. Register Call Summary for Register CFG_VIN1A_D9_OEN........ 5004
Table 18-2175. CFG_VIN1A_D9_OUT........ 5004
Table 18-2176. Register Call Summary for Register CFG_VIN1A_D9_OUT........ 5005
Table 18-2177. CFG_VIN1A_DE0_IN........ 5005
Table 18-2178. Register Call Summary for Register CFG_VIN1A_DE0_IN........ 5005
Table 18-2179. CFG_VIN1A_DE0_OEN........ 5005
Table 18-2180. Register Call Summary for Register CFG_VIN1A_DE0_OEN........ 5006
Table 18-2181. CFG_VIN1A_DE0_OUT........ 5006
Table 18-2182. Register Call Summary for Register CFG_VIN1A_DE0_OUT........ 5006
Table 18-2183. CFG_VIN1A_FLD0_IN........ 5006
Table 18-2184. Register Call Summary for Register CFG_VIN1A_FLD0_IN........ 5007
Table 18-2185. CFG_VIN1A_FLD0_OEN........ 5007
Table 18-2186. Register Call Summary for Register CFG_VIN1A_FLD0_OEN........ 5007
Table 18-2187. CFG_VIN1A_FLD0_OUT........ 5007
Table 18-2188. Register Call Summary for Register CFG_VIN1A_FLD0_OUT........ 5008
Table 18-2189. CFG_VIN1A_HSYNC0_IN........ 5008
Table 18-2190. Register Call Summary for Register CFG_VIN1A_HSYNC0_IN........ 5008
Table 18-2191. CFG_VIN1A_HSYNC0_OEN........ 5008
Table 18-2192. Register Call Summary for Register CFG_VIN1A_HSYNC0_OEN........ 5009
Table 18-2193. CFG_VIN1A_HSYNC0_OUT........ 5009
Table 18-2194. Register Call Summary for Register CFG_VIN1A_HSYNC0_OUT........ 5009
Table 18-2195. CFG_VIN1A_VSYNC0_IN........ 5009
Table 18-2196. Register Call Summary for Register CFG_VIN1A_VSYNC0_IN........ 5010
Table 18-2197. CFG_VIN1A_VSYNC0_OEN........ 5010
Table 18-2198. Register Call Summary for Register CFG_VIN1A_VSYNC0_OEN........ 5010
Table 18-2199. CFG_VIN1A_VSYNC0_OUT........ 5010
Table 18-2200. Register Call Summary for Register CFG_VIN1A_VSYNC0_OUT........ 5011
Table 18-2201. CFG_VIN1B_CLK1_IN........ 5011
Table 18-2202. Register Call Summary for Register CFG_VIN1B_CLK1_IN........ 5011
Table 18-2203. CFG_VIN1B_CLK1_OEN........ 5011
Table 18-2204. Register Call Summary for Register CFG_VIN1B_CLK1_OEN........ 5012
Table 18-2205. CFG_VIN1B_CLK1_OUT........ 5012
Table 18-2206. Register Call Summary for Register CFG_VIN1B_CLK1_OUT........ 5012

Table 18-2207. CFG_VIN2A_CLK0_IN.. 5012
Table 18-2208. Register Call Summary for Register CFG_VIN2A_CLK0_IN..5013
Table 18-2209. CFG_VIN2A_CLK0_OEN.. 5013
Table 18-2210. Register Call Summary for Register CFG_VIN2A_CLK0_OEN..5013
Table 18-2211. CFG_VIN2A_CLK0_OUT..5013
Table 18-2212. Register Call Summary for Register CFG_VIN2A_CLK0_OUT..5014
Table 18-2213. CFG_VIN2A_D0_IN..5014
Table 18-2214. Register Call Summary for Register CFG_VIN2A_D0_IN.. 5014
Table 18-2215. CFG_VIN2A_D0_OEN..5014
Table 18-2216. Register Call Summary for Register CFG_VIN2A_D0_OEN..5015
Table 18-2217. CFG_VIN2A_D0_OUT..5015
Table 18-2218. Register Call Summary for Register CFG_VIN2A_D0_OUT... 5015
Table 18-2219. CFG_VIN2A_D10_IN..5015
Table 18-2220. Register Call Summary for Register CFG_VIN2A_D10_IN.. 5016
Table 18-2221. CFG_VIN2A_D10_OEN..5016
Table 18-2222. Register Call Summary for Register CFG_VIN2A_D10_OEN... 5016
Table 18-2223. CFG_VIN2A_D10_OUT..5016
Table 18-2224. Register Call Summary for Register CFG_VIN2A_D10_OUT... 5017
Table 18-2225. CFG_VIN2A_D11_IN.. 5017
Table 18-2226. Register Call Summary for Register CFG_VIN2A_D11_IN...5017
Table 18-2227. CFG_VIN2A_D11_OEN..5017
Table 18-2228. Register Call Summary for Register CFG_VIN2A_D11_OEN... 5018
Table 18-2229. CFG_VIN2A_D11_OUT.. 5018
Table 18-2230. Register Call Summary for Register CFG_VIN2A_D11_OUT..5018
Table 18-2231. CFG_VIN2A_D12_IN..5018
Table 18-2232. Register Call Summary for Register CFG_VIN2A_D12_IN.. 5019
Table 18-2233. CFG_VIN2A_D12_OEN..5019
Table 18-2234. Register Call Summary for Register CFG_VIN2A_D12_OEN..5019
Table 18-2235. CFG_VIN2A_D12_OUT..5019
Table 18-2236. Register Call Summary for Register CFG_VIN2A_D12_OUT... 5020
Table 18-2237. CFG_VIN2A_D13_IN..5020
Table 18-2238. Register Call Summary for Register CFG_VIN2A_D13_IN.. 5020
Table 18-2239. CFG_VIN2A_D13_OEN..5020
Table 18-2240. Register Call Summary for Register CFG_VIN2A_D13_OEN..5021
Table 18-2241. CFG_VIN2A_D13_OUT..5021
Table 18-2242. Register Call Summary for Register CFG_VIN2A_D13_OUT... 5021
Table 18-2243. CFG_VIN2A_D14_IN..5021
Table 18-2244. Register Call Summary for Register CFG_VIN2A_D14_IN.. 5022
Table 18-2245. CFG_VIN2A_D14_OEN..5022
Table 18-2246. Register Call Summary for Register CFG_VIN2A_D14_OEN..5022
Table 18-2247. CFG_VIN2A_D14_OUT..5022
Table 18-2248. Register Call Summary for Register CFG_VIN2A_D14_OUT... 5023
Table 18-2249. CFG_VIN2A_D15_IN..5023
Table 18-2250. Register Call Summary for Register CFG_VIN2A_D15_IN.. 5023
Table 18-2251. CFG_VIN2A_D15_OEN..5023
Table 18-2252. Register Call Summary for Register CFG_VIN2A_D15_OEN..5024
Table 18-2253. CFG_VIN2A_D15_OUT..5024
Table 18-2254. Register Call Summary for Register CFG_VIN2A_D15_OUT... 5024
Table 18-2255. CFG_VIN2A_D16_IN..5024
Table 18-2256. Register Call Summary for Register CFG_VIN2A_D16_IN.. 5025
Table 18-2257. CFG_VIN2A_D16_OEN..5025
Table 18-2258. Register Call Summary for Register CFG_VIN2A_D16_OEN..5025
Table 18-2259. CFG_VIN2A_D16_OUT..5025
Table 18-2260. Register Call Summary for Register CFG_VIN2A_D16_OUT... 5026
Table 18-2261. CFG_VIN2A_D17_IN..5026
Table 18-2262. Register Call Summary for Register CFG_VIN2A_D17_IN.. 5026
Table 18-2263. CFG_VIN2A_D17_OEN..5026
Table 18-2264. Register Call Summary for Register CFG_VIN2A_D17_OEN..5027
Table 18-2265. CFG_VIN2A_D17_OUT..5027
Table 18-2266. Register Call Summary for Register CFG_VIN2A_D17_OUT... 5027
Table 18-2267. CFG_VIN2A_D18_IN..5027

Table 18-2268. Register Call Summary for Register CFG_VIN2A_D18_IN.......... 5028
Table 18-2269. CFG_VIN2A_D18_OEN.......... 5028
Table 18-2270. Register Call Summary for Register CFG_VIN2A_D18_OEN.......... 5028
Table 18-2271. CFG_VIN2A_D18_OUT.......... 5028
Table 18-2272. Register Call Summary for Register CFG_VIN2A_D18_OUT.......... 5029
Table 18-2273. CFG_VIN2A_D19_IN.......... 5029
Table 18-2274. Register Call Summary for Register CFG_VIN2A_D19_IN.......... 5029
Table 18-2275. CFG_VIN2A_D19_OEN.......... 5029
Table 18-2276. Register Call Summary for Register CFG_VIN2A_D19_OEN.......... 5030
Table 18-2277. CFG_VIN2A_D19_OUT.......... 5030
Table 18-2278. Register Call Summary for Register CFG_VIN2A_D19_OUT.......... 5030
Table 18-2279. CFG_VIN2A_D1_IN.......... 5030
Table 18-2280. Register Call Summary for Register CFG_VIN2A_D1_IN.......... 5031
Table 18-2281. CFG_VIN2A_D1_OEN.......... 5031
Table 18-2282. Register Call Summary for Register CFG_VIN2A_D1_OEN.......... 5031
Table 18-2283. CFG_VIN2A_D1_OUT.......... 5031
Table 18-2284. Register Call Summary for Register CFG_VIN2A_D1_OUT.......... 5032
Table 18-2285. CFG_VIN2A_D20_IN.......... 5032
Table 18-2286. Register Call Summary for Register CFG_VIN2A_D20_IN.......... 5032
Table 18-2287. CFG_VIN2A_D20_OEN.......... 5032
Table 18-2288. Register Call Summary for Register CFG_VIN2A_D20_OEN.......... 5033
Table 18-2289. CFG_VIN2A_D20_OUT.......... 5033
Table 18-2290. Register Call Summary for Register CFG_VIN2A_D20_OUT.......... 5033
Table 18-2291. CFG_VIN2A_D21_IN.......... 5033
Table 18-2292. Register Call Summary for Register CFG_VIN2A_D21_IN.......... 5034
Table 18-2293. CFG_VIN2A_D21_OEN.......... 5034
Table 18-2294. Register Call Summary for Register CFG_VIN2A_D21_OEN.......... 5034
Table 18-2295. CFG_VIN2A_D21_OUT.......... 5034
Table 18-2296. Register Call Summary for Register CFG_VIN2A_D21_OUT.......... 5035
Table 18-2297. CFG_VIN2A_D22_IN.......... 5035
Table 18-2298. Register Call Summary for Register CFG_VIN2A_D22_IN.......... 5035
Table 18-2299. CFG_VIN2A_D22_OEN.......... 5035
Table 18-2300. Register Call Summary for Register CFG_VIN2A_D22_OEN.......... 5036
Table 18-2301. CFG_VIN2A_D22_OUT.......... 5036
Table 18-2302. Register Call Summary for Register CFG_VIN2A_D22_OUT.......... 5036
Table 18-2303. CFG_VIN2A_D23_IN.......... 5036
Table 18-2304. Register Call Summary for Register CFG_VIN2A_D23_IN.......... 5037
Table 18-2305. CFG_VIN2A_D23_OEN.......... 5037
Table 18-2306. Register Call Summary for Register CFG_VIN2A_D23_OEN.......... 5037
Table 18-2307. CFG_VIN2A_D23_OUT.......... 5037
Table 18-2308. Register Call Summary for Register CFG_VIN2A_D23_OUT.......... 5038
Table 18-2309. CFG_VIN2A_D2_IN.......... 5038
Table 18-2310. Register Call Summary for Register CFG_VIN2A_D2_IN.......... 5038
Table 18-2311. CFG_VIN2A_D2_OEN.......... 5038
Table 18-2312. Register Call Summary for Register CFG_VIN2A_D2_OEN.......... 5039
Table 18-2313. CFG_VIN2A_D2_OUT.......... 5039
Table 18-2314. Register Call Summary for Register CFG_VIN2A_D2_OUT.......... 5039
Table 18-2315. CFG_VIN2A_D3_IN.......... 5039
Table 18-2316. Register Call Summary for Register CFG_VIN2A_D3_IN.......... 5040
Table 18-2317. CFG_VIN2A_D3_OEN.......... 5040
Table 18-2318. Register Call Summary for Register CFG_VIN2A_D3_OEN.......... 5040
Table 18-2319. CFG_VIN2A_D3_OUT.......... 5040
Table 18-2320. Register Call Summary for Register CFG_VIN2A_D3_OUT.......... 5041
Table 18-2321. CFG_VIN2A_D4_IN.......... 5041
Table 18-2322. Register Call Summary for Register CFG_VIN2A_D4_IN.......... 5041
Table 18-2323. CFG_VIN2A_D4_OEN.......... 5041
Table 18-2324. Register Call Summary for Register CFG_VIN2A_D4_OEN.......... 5042
Table 18-2325. CFG_VIN2A_D4_OUT.......... 5042
Table 18-2326. Register Call Summary for Register CFG_VIN2A_D4_OUT.......... 5042
Table 18-2327. CFG_VIN2A_D5_IN.......... 5042
Table 18-2328. Register Call Summary for Register CFG_VIN2A_D5_IN.......... 5043

Table 18-2329. CFG_VIN2A_D5_OEN....5043
Table 18-2330. Register Call Summary for Register CFG_VIN2A_D5_OEN....5043
Table 18-2331. CFG_VIN2A_D5_OUT....5043
Table 18-2332. Register Call Summary for Register CFG_VIN2A_D5_OUT....5044
Table 18-2333. CFG_VIN2A_D6_IN....5044
Table 18-2334. Register Call Summary for Register CFG_VIN2A_D6_IN....5044
Table 18-2335. CFG_VIN2A_D6_OEN....5044
Table 18-2336. Register Call Summary for Register CFG_VIN2A_D6_OEN....5045
Table 18-2337. CFG_VIN2A_D6_OUT....5045
Table 18-2338. Register Call Summary for Register CFG_VIN2A_D6_OUT....5045
Table 18-2339. CFG_VIN2A_D7_IN....5045
Table 18-2340. Register Call Summary for Register CFG_VIN2A_D7_IN....5046
Table 18-2341. CFG_VIN2A_D7_OEN....5046
Table 18-2342. Register Call Summary for Register CFG_VIN2A_D7_OEN....5046
Table 18-2343. CFG_VIN2A_D7_OUT....5046
Table 18-2344. Register Call Summary for Register CFG_VIN2A_D7_OUT....5047
Table 18-2345. CFG_VIN2A_D8_IN....5047
Table 18-2346. Register Call Summary for Register CFG_VIN2A_D8_IN....5047
Table 18-2347. CFG_VIN2A_D8_OEN....5047
Table 18-2348. Register Call Summary for Register CFG_VIN2A_D8_OEN....5048
Table 18-2349. CFG_VIN2A_D8_OUT....5048
Table 18-2350. Register Call Summary for Register CFG_VIN2A_D8_OUT....5048
Table 18-2351. CFG_VIN2A_D9_IN....5048
Table 18-2352. Register Call Summary for Register CFG_VIN2A_D9_IN....5049
Table 18-2353. CFG_VIN2A_D9_OEN....5049
Table 18-2354. Register Call Summary for Register CFG_VIN2A_D9_OEN....5049
Table 18-2355. CFG_VIN2A_D9_OUT....5049
Table 18-2356. Register Call Summary for Register CFG_VIN2A_D9_OUT....5050
Table 18-2357. CFG_VIN2A_DE0_IN....5050
Table 18-2358. Register Call Summary for Register CFG_VIN2A_DE0_IN....5050
Table 18-2359. CFG_VIN2A_DE0_OEN....5050
Table 18-2360. Register Call Summary for Register CFG_VIN2A_DE0_OEN....5051
Table 18-2361. CFG_VIN2A_DE0_OUT....5051
Table 18-2362. Register Call Summary for Register CFG_VIN2A_DE0_OUT....5051
Table 18-2363. CFG_VIN2A_FLD0_IN....5051
Table 18-2364. Register Call Summary for Register CFG_VIN2A_FLD0_IN....5052
Table 18-2365. CFG_VIN2A_FLD0_OEN....5052
Table 18-2366. Register Call Summary for Register CFG_VIN2A_FLD0_OEN....5052
Table 18-2367. CFG_VIN2A_FLD0_OUT....5052
Table 18-2368. Register Call Summary for Register CFG_VIN2A_FLD0_OUT....5053
Table 18-2369. CFG_VIN2A_HSYNC0_IN....5053
Table 18-2370. Register Call Summary for Register CFG_VIN2A_HSYNC0_IN....5053
Table 18-2371. CFG_VIN2A_HSYNC0_OEN....5053
Table 18-2372. Register Call Summary for Register CFG_VIN2A_HSYNC0_OEN....5054
Table 18-2373. CFG_VIN2A_HSYNC0_OUT....5054
Table 18-2374. Register Call Summary for Register CFG_VIN2A_HSYNC0_OUT....5054
Table 18-2375. CFG_VIN2A_VSYNC0_IN....5054
Table 18-2376. Register Call Summary for Register CFG_VIN2A_VSYNC0_IN....5055
Table 18-2377. CFG_VIN2A_VSYNC0_OEN....5055
Table 18-2378. Register Call Summary for Register CFG_VIN2A_VSYNC0_OEN....5055
Table 18-2379. CFG_VIN2A_VSYNC0_OUT....5055
Table 18-2380. Register Call Summary for Register CFG_VIN2A_VSYNC0_OUT....5056
Table 18-2381. CFG_VOUT1_CLK_IN....5056
Table 18-2382. Register Call Summary for Register CFG_VOUT1_CLK_IN....5056
Table 18-2383. CFG_VOUT1_CLK_OEN....5056
Table 18-2384. Register Call Summary for Register CFG_VOUT1_CLK_OEN....5057
Table 18-2385. CFG_VOUT1_CLK_OUT....5057
Table 18-2386. Register Call Summary for Register CFG_VOUT1_CLK_OUT....5057
Table 18-2387. CFG_VOUT1_D0_IN....5057
Table 18-2388. Register Call Summary for Register CFG_VOUT1_D0_IN....5058
Table 18-2389. CFG_VOUT1_D0_OEN....5058

Table 18-2390. Register Call Summary for Register CFG_VOUT1_D0_OEN.......... 5058
Table 18-2391. CFG_VOUT1_D0_OUT.......... 5058
Table 18-2392. Register Call Summary for Register CFG_VOUT1_D0_OUT.......... 5059
Table 18-2393. CFG_VOUT1_D10_IN.......... 5059
Table 18-2394. Register Call Summary for Register CFG_VOUT1_D10_IN.......... 5059
Table 18-2395. CFG_VOUT1_D10_OEN.......... 5059
Table 18-2396. Register Call Summary for Register CFG_VOUT1_D10_OEN.......... 5060
Table 18-2397. CFG_VOUT1_D10_OUT.......... 5060
Table 18-2398. Register Call Summary for Register CFG_VOUT1_D10_OUT.......... 5060
Table 18-2399. CFG_VOUT1_D11_IN.......... 5060
Table 18-2400. Register Call Summary for Register CFG_VOUT1_D11_IN.......... 5061
Table 18-2401. CFG_VOUT1_D11_OEN.......... 5061
Table 18-2402. Register Call Summary for Register CFG_VOUT1_D11_OEN.......... 5061
Table 18-2403. CFG_VOUT1_D11_OUT.......... 5061
Table 18-2404. Register Call Summary for Register CFG_VOUT1_D11_OUT.......... 5062
Table 18-2405. CFG_VOUT1_D12_IN.......... 5062
Table 18-2406. Register Call Summary for Register CFG_VOUT1_D12_IN.......... 5062
Table 18-2407. CFG_VOUT1_D12_OEN.......... 5062
Table 18-2408. Register Call Summary for Register CFG_VOUT1_D12_OEN.......... 5063
Table 18-2409. CFG_VOUT1_D12_OUT.......... 5063
Table 18-2410. Register Call Summary for Register CFG_VOUT1_D12_OUT.......... 5063
Table 18-2411. CFG_VOUT1_D13_IN.......... 5063
Table 18-2412. Register Call Summary for Register CFG_VOUT1_D13_IN.......... 5064
Table 18-2413. CFG_VOUT1_D13_OEN.......... 5064
Table 18-2414. Register Call Summary for Register CFG_VOUT1_D13_OEN.......... 5064
Table 18-2415. CFG_VOUT1_D13_OUT.......... 5064
Table 18-2416. Register Call Summary for Register CFG_VOUT1_D13_OUT.......... 5065
Table 18-2417. CFG_VOUT1_D14_IN.......... 5065
Table 18-2418. Register Call Summary for Register CFG_VOUT1_D14_IN.......... 5065
Table 18-2419. CFG_VOUT1_D14_OEN.......... 5065
Table 18-2420. Register Call Summary for Register CFG_VOUT1_D14_OEN.......... 5066
Table 18-2421. CFG_VOUT1_D14_OUT.......... 5066
Table 18-2422. Register Call Summary for Register CFG_VOUT1_D14_OUT.......... 5066
Table 18-2423. CFG_VOUT1_D15_IN.......... 5066
Table 18-2424. Register Call Summary for Register CFG_VOUT1_D15_IN.......... 5067
Table 18-2425. CFG_VOUT1_D15_OEN.......... 5067
Table 18-2426. Register Call Summary for Register CFG_VOUT1_D15_OEN.......... 5067
Table 18-2427. CFG_VOUT1_D15_OUT.......... 5067
Table 18-2428. Register Call Summary for Register CFG_VOUT1_D15_OUT.......... 5068
Table 18-2429. CFG_VOUT1_D16_IN.......... 5068
Table 18-2430. Register Call Summary for Register CFG_VOUT1_D16_IN.......... 5068
Table 18-2431. CFG_VOUT1_D16_OEN.......... 5068
Table 18-2432. Register Call Summary for Register CFG_VOUT1_D16_OEN.......... 5069
Table 18-2433. CFG_VOUT1_D16_OUT.......... 5069
Table 18-2434. Register Call Summary for Register CFG_VOUT1_D16_OUT.......... 5069
Table 18-2435. CFG_VOUT1_D17_IN.......... 5069
Table 18-2436. Register Call Summary for Register CFG_VOUT1_D17_IN.......... 5070
Table 18-2437. CFG_VOUT1_D17_OEN.......... 5070
Table 18-2438. Register Call Summary for Register CFG_VOUT1_D17_OEN.......... 5070
Table 18-2439. CFG_VOUT1_D17_OUT.......... 5070
Table 18-2440. Register Call Summary for Register CFG_VOUT1_D17_OUT.......... 5071
Table 18-2441. CFG_VOUT1_D18_IN.......... 5071
Table 18-2442. Register Call Summary for Register CFG_VOUT1_D18_IN.......... 5071
Table 18-2443. CFG_VOUT1_D18_OEN.......... 5071
Table 18-2444. Register Call Summary for Register CFG_VOUT1_D18_OEN.......... 5072
Table 18-2445. CFG_VOUT1_D18_OUT.......... 5072
Table 18-2446. Register Call Summary for Register CFG_VOUT1_D18_OUT.......... 5072
Table 18-2447. CFG_VOUT1_D19_IN.......... 5072
Table 18-2448. Register Call Summary for Register CFG_VOUT1_D19_IN.......... 5073
Table 18-2449. CFG_VOUT1_D19_OEN.......... 5073
Table 18-2450. Register Call Summary for Register CFG_VOUT1_D19_OEN.......... 5073

Table 18-2451. CFG_VOUT1_D19_OUT... 5073
Table 18-2452. Register Call Summary for Register CFG_VOUT1_D19_OUT...5074
Table 18-2453. CFG_VOUT1_D1_IN.. 5074
Table 18-2454. Register Call Summary for Register CFG_VOUT1_D1_IN..5074
Table 18-2455. CFG_VOUT1_D1_OEN...5074
Table 18-2456. Register Call Summary for Register CFG_VOUT1_D1_OEN.. 5075
Table 18-2457. CFG_VOUT1_D1_OUT... 5075
Table 18-2458. Register Call Summary for Register CFG_VOUT1_D1_OUT...5075
Table 18-2459. CFG_VOUT1_D20_IN.. 5075
Table 18-2460. Register Call Summary for Register CFG_VOUT1_D20_IN..5076
Table 18-2461. CFG_VOUT1_D20_OEN...5076
Table 18-2462. Register Call Summary for Register CFG_VOUT1_D20_OEN.. 5076
Table 18-2463. CFG_VOUT1_D20_OUT... 5076
Table 18-2464. Register Call Summary for Register CFG_VOUT1_D20_OUT...5077
Table 18-2465. CFG_VOUT1_D21_IN.. 5077
Table 18-2466. Register Call Summary for Register CFG_VOUT1_D21_IN..5077
Table 18-2467. CFG_VOUT1_D21_OEN...5077
Table 18-2468. Register Call Summary for Register CFG_VOUT1_D21_OEN.. 5078
Table 18-2469. CFG_VOUT1_D21_OUT... 5078
Table 18-2470. Register Call Summary for Register CFG_VOUT1_D21_OUT...5078
Table 18-2471. CFG_VOUT1_D22_IN.. 5078
Table 18-2472. Register Call Summary for Register CFG_VOUT1_D22_IN..5079
Table 18-2473. CFG_VOUT1_D22_OEN...5079
Table 18-2474. Register Call Summary for Register CFG_VOUT1_D22_OEN.. 5079
Table 18-2475. CFG_VOUT1_D22_OUT... 5079
Table 18-2476. Register Call Summary for Register CFG_VOUT1_D22_OUT...5080
Table 18-2477. CFG_VOUT1_D23_IN.. 5080
Table 18-2478. Register Call Summary for Register CFG_VOUT1_D23_IN..5080
Table 18-2479. CFG_VOUT1_D23_OEN...5080
Table 18-2480. Register Call Summary for Register CFG_VOUT1_D23_OEN.. 5081
Table 18-2481. CFG_VOUT1_D23_OUT... 5081
Table 18-2482. Register Call Summary for Register CFG_VOUT1_D23_OUT...5081
Table 18-2483. CFG_VOUT1_D2_IN.. 5081
Table 18-2484. Register Call Summary for Register CFG_VOUT1_D2_IN..5082
Table 18-2485. CFG_VOUT1_D2_OEN...5082
Table 18-2486. Register Call Summary for Register CFG_VOUT1_D2_OEN.. 5082
Table 18-2487. CFG_VOUT1_D2_OUT... 5082
Table 18-2488. Register Call Summary for Register CFG_VOUT1_D2_OUT...5083
Table 18-2489. CFG_VOUT1_D3_IN.. 5083
Table 18-2490. Register Call Summary for Register CFG_VOUT1_D3_IN..5083
Table 18-2491. CFG_VOUT1_D3_OEN...5083
Table 18-2492. Register Call Summary for Register CFG_VOUT1_D3_OEN.. 5084
Table 18-2493. CFG_VOUT1_D3_OUT... 5084
Table 18-2494. Register Call Summary for Register CFG_VOUT1_D3_OUT...5084
Table 18-2495. CFG_VOUT1_D4_IN.. 5084
Table 18-2496. Register Call Summary for Register CFG_VOUT1_D4_IN..5085
Table 18-2497. CFG_VOUT1_D4_OEN...5085
Table 18-2498. Register Call Summary for Register CFG_VOUT1_D4_OEN.. 5085
Table 18-2499. CFG_VOUT1_D4_OUT... 5085
Table 18-2500. Register Call Summary for Register CFG_VOUT1_D4_OUT...5086
Table 18-2501. CFG_VOUT1_D5_IN.. 5086
Table 18-2502. Register Call Summary for Register CFG_VOUT1_D5_IN..5086
Table 18-2503. CFG_VOUT1_D5_OEN...5086
Table 18-2504. Register Call Summary for Register CFG_VOUT1_D5_OEN.. 5087
Table 18-2505. CFG_VOUT1_D5_OUT... 5087
Table 18-2506. Register Call Summary for Register CFG_VOUT1_D5_OUT...5087
Table 18-2507. CFG_VOUT1_D6_IN.. 5087
Table 18-2508. Register Call Summary for Register CFG_VOUT1_D6_IN..5088
Table 18-2509. CFG_VOUT1_D6_OEN...5088
Table 18-2510. Register Call Summary for Register CFG_VOUT1_D6_OEN.. 5088
Table 18-2511. CFG_VOUT1_D6_OUT...5088

Table 18-2512. Register Call Summary for Register CFG_VOUT1_D6_OUT....5089
Table 18-2513. CFG_VOUT1_D7_IN....5089
Table 18-2514. Register Call Summary for Register CFG_VOUT1_D7_IN....5089
Table 18-2515. CFG_VOUT1_D7_OEN....5089
Table 18-2516. Register Call Summary for Register CFG_VOUT1_D7_OEN....5090
Table 18-2517. CFG_VOUT1_D7_OUT....5090
Table 18-2518. Register Call Summary for Register CFG_VOUT1_D7_OUT....5090
Table 18-2519. CFG_VOUT1_D8_IN....5090
Table 18-2520. Register Call Summary for Register CFG_VOUT1_D8_IN....5091
Table 18-2521. CFG_VOUT1_D8_OEN....5091
Table 18-2522. Register Call Summary for Register CFG_VOUT1_D8_OEN....5091
Table 18-2523. CFG_VOUT1_D8_OUT....5091
Table 18-2524. Register Call Summary for Register CFG_VOUT1_D8_OUT....5092
Table 18-2525. CFG_VOUT1_D9_IN....5092
Table 18-2526. Register Call Summary for Register CFG_VOUT1_D9_IN....5092
Table 18-2527. CFG_VOUT1_D9_OEN....5092
Table 18-2528. Register Call Summary for Register CFG_VOUT1_D9_OEN....5093
Table 18-2529. CFG_VOUT1_D9_OUT....5093
Table 18-2530. Register Call Summary for Register CFG_VOUT1_D9_OUT....5093
Table 18-2531. CFG_VOUT1_DE_IN....5093
Table 18-2532. Register Call Summary for Register CFG_VOUT1_DE_IN....5094
Table 18-2533. CFG_VOUT1_DE_OEN....5094
Table 18-2534. Register Call Summary for Register CFG_VOUT1_DE_OEN....5094
Table 18-2535. CFG_VOUT1_DE_OUT....5094
Table 18-2536. Register Call Summary for Register CFG_VOUT1_DE_OUT....5095
Table 18-2537. CFG_VOUT1_FLD_IN....5095
Table 18-2538. Register Call Summary for Register CFG_VOUT1_FLD_IN....5095
Table 18-2539. CFG_VOUT1_FLD_OEN....5095
Table 18-2540. Register Call Summary for Register CFG_VOUT1_FLD_OEN....5096
Table 18-2541. CFG_VOUT1_FLD_OUT....5096
Table 18-2542. Register Call Summary for Register CFG_VOUT1_FLD_OUT....5096
Table 18-2543. CFG_VOUT1_HSYNC_IN....5096
Table 18-2544. Register Call Summary for Register CFG_VOUT1_HSYNC_IN....5097
Table 18-2545. CFG_VOUT1_HSYNC_OEN....5097
Table 18-2546. Register Call Summary for Register CFG_VOUT1_HSYNC_OEN....5097
Table 18-2547. CFG_VOUT1_HSYNC_OUT....5097
Table 18-2548. Register Call Summary for Register CFG_VOUT1_HSYNC_OUT....5098
Table 18-2549. CFG_VOUT1_VSYNC_IN....5098
Table 18-2550. Register Call Summary for Register CFG_VOUT1_VSYNC_IN....5098
Table 18-2551. CFG_VOUT1_VSYNC_OEN....5098
Table 18-2552. Register Call Summary for Register CFG_VOUT1_VSYNC_OEN....5099
Table 18-2553. CFG_VOUT1_VSYNC_OUT....5099
Table 18-2554. Register Call Summary for Register CFG_VOUT1_VSYNC_OUT....5099
Table 18-2555. CFG_XREF_CLK0_IN....5099
Table 18-2556. Register Call Summary for Register CFG_XREF_CLK0_IN....5100
Table 18-2557. CFG_XREF_CLK0_OEN....5100
Table 18-2558. Register Call Summary for Register CFG_XREF_CLK0_OEN....5100
Table 18-2559. CFG_XREF_CLK0_OUT....5100
Table 18-2560. Register Call Summary for Register CFG_XREF_CLK0_OUT....5101
Table 18-2561. CFG_XREF_CLK1_IN....5101
Table 18-2562. Register Call Summary for Register CFG_XREF_CLK1_IN....5101
Table 18-2563. CFG_XREF_CLK1_OEN....5101
Table 18-2564. Register Call Summary for Register CFG_XREF_CLK1_OEN....5102
Table 18-2565. CFG_XREF_CLK1_OUT....5102
Table 18-2566. Register Call Summary for Register CFG_XREF_CLK1_OUT....5102
Table 18-2567. CFG_XREF_CLK2_IN....5102
Table 18-2568. Register Call Summary for Register CFG_XREF_CLK2_IN....5103
Table 18-2569. CFG_XREF_CLK2_OEN....5103
Table 18-2570. Register Call Summary for Register CFG_XREF_CLK2_OEN....5103
Table 18-2571. CFG_XREF_CLK2_OUT....5103
Table 18-2572. Register Call Summary for Register CFG_XREF_CLK2_OUT....5104

Table 18-2573. CFG_XREF_CLK3_IN..5104
Table 18-2574. Register Call Summary for Register CFG_XREF_CLK3_IN..5104
Table 18-2575. CFG_XREF_CLK3_OEN..5104
Table 18-2576. Register Call Summary for Register CFG_XREF_CLK3_OEN..5105
Table 18-2577. CFG_XREF_CLK3_OUT..5105
Table 18-2578. Register Call Summary for Register CFG_XREF_CLK3_OUT..5105
Table 19-1. Mailbox Configuration in the Device..5107
Table 19-2. MAILBOX Integration Attributes..5108
Table 19-3. MAILBOX Clocks and Resets...5108
Table 19-4. MAILBOX Hardware Requests..5109
Table 19-5. IVA_MBOX Integration Attributes...5111
Table 19-6. IVA_MBOX Clocks and Resets..5111
Table 19-7. IVA_MBOX Hardware Requests..5111
Table 19-8. EVEx_MBOX Integration Attributes...5112
Table 19-9. EVEx_MBOX Clocks and Resets..5113
Table 19-10. EVEx_MBOX Hardware Requests..5113
Table 19-11. Mailbox Users in the Device..5115
Table 19-12. Local Power Management Features..5116
Table 19-13. Interrupt Events...5117
Table 19-14. Global Initialization of Surrounding Modules for MAILBOX...5121
Table 19-15. Global Initialization of Surrounding Modules for IVA_MBOX...5121
Table 19-16. Global Initialization of Surrounding Modules for EVEx_MBOX...5121
Table 19-17. Mailbox Global Initialization..5121
Table 19-18. Sending a Message (Polling Method)..5122
Table 19-19. Sending a Message (Interrupt Method)..5122
Table 19-20. Receiving a Message (Polling Method)...5122
Table 19-21. Receiving a Message (Interrupt Method)...5122
Table 19-22. Events Servicing in Sending Mode..5122
Table 19-23. Events Servicing in Receiving Mode..5123
Table 19-24. Mailbox Instance Summary..5124
Table 19-25. MAILBOX Registers Mapping Summary (1/5)..5124
Table 19-26. MAILBOX Registers Mapping Summary (2/5)..5125
Table 19-27. MAILBOX Registers Mapping Summary (3/5)..5125
Table 19-28. MAILBOX Registers Mapping Summary (4/5)..5125
Table 19-29. MAILBOX Registers Mapping Summary (5/5)..5126
Table 19-30. IVA_MBOX Registers Mapping Summary..5126
Table 19-31. EVE1_MBOX Registers Mapping Summary..5127
Table 19-32. EVE2_MBOX Registers Mapping Summary..5127
Table 19-33. MAILBOX_REVISION...5128
Table 19-34. MAILBOX_SYSCONFIG..5128
Table 19-35. MAILBOX_MESSAGE_m...5129
Table 19-36. MAILBOX_FIFOSTATUS_m...5129
Table 19-37. MAILBOX_MSGSTATUS_m..5129
Table 19-38. MAILBOX_IRQSTATUS_RAW_u...5130
Table 19-39. MAILBOX_IRQSTATUS_CLR_u..5134
Table 19-40. MAILBOX_IRQENABLE_SET_u..5138
Table 19-41. MAILBOX_IRQENABLE_CLR_u..5142
Table 19-42. MAILBOX_IRQ_EOI...5146
Table 20-1. MMU Integration Attributes...5152
Table 20-2. MMU Clocks and Resets...5152
Table 20-3. MMU Hardware Requests...5152
Table 20-4. First-Level Descriptor Format..5156
Table 20-5. Second-Level Descriptor Format..5159
Table 20-6. MMU Local Power Management Features...5163
Table 20-7. MMU Events...5163
Table 20-8. Error Handling...5164
Table 20-9. Global Initialization of Surrounding Modules..5165
Table 20-10. Configure a TLB Entry..5166
Table 20-11. MMU Writing TLB Entries Statically...5167
Table 20-12. Protecting TLB Entries..5167
Table 20-13. Deleting TLB Entries..5167

Table 20-14. Read TLB Entries....5167
Table 20-15. MMU Instance Summary....5168
Table 20-16. System MMU Register Mapping Summary....5168
Table 20-17. DSP1 MMU Register Mapping Summary....5169
Table 20-18. DSP2 MMU Register Mapping Summary....5170
Table 20-19. EVE1 MMU Register Mapping Summary....5171
Table 20-20. EVE2 MMU Register Mapping Summary....5172
Table 20-21. IPU MMU Register Mapping Summary....5172
Table 20-22. MMU_REVISION....5173
Table 20-23. MMU_SYSCONFIG....5173
Table 20-24. MMU_SYSSTATUS....5174
Table 20-25. MMU_IRQSTATUS....5174
Table 20-26. MMU_IRQENABLE....5175
Table 20-27. MMU_WALKING_ST....5176
Table 20-28. MMU_CNTL....5177
Table 20-29. MMU_FAULT_AD....5177
Table 20-30. MMU_TTB....5177
Table 20-31. MMU_LOCK....5178
Table 20-32. MMU_LD_TLB....5178
Table 20-33. MMU_CAM....5178
Table 20-34. MMU_RAM....5179
Table 20-35. MMU_GFLUSH....5179
Table 20-36. MMU_FLUSH_ENTRY....5179
Table 20-37. MMU_READ_CAM....5180
Table 20-38. MMU_READ_RAM....5180
Table 20-39. MMU_EMU_FAULT_AD....5181
Table 20-40. MMU_FAULT_PC....5181
Table 20-41. MMU_FAULT_STATUS....5181
Table 20-42. MMU_GPR....5182
Table 20-43. MMU_BYPASS_REGION1_ADDR....5182
Table 20-44. MMU_BYPASS_REGION1_SIZE....5182
Table 20-45. MMU_BYPASS_REGION2_ADDR....5183
Table 20-46. MMU_BYPASS_REGION2_SIZE....5183
Table 20-47. MMU_BYPASS_REGION3_ADDR....5184
Table 20-48. MMU_BYPASS_REGION3_SIZE....5184
Table 20-49. MMU_BYPASS_REGION4_ADDR....5185
Table 20-50. MMU_BYPASS_REGION4_SIZE....5185
Table 21-1. Spinlock Integration Attributes....5189
Table 21-2. Spinlock Clocks and Resets....5189
Table 21-3. Spinlock Local Power Management Features....5190
Table 21-4. Global Initialization of Surrounding Modules....5192
Table 21-5. Spinlock System Bug Recovery....5192
Table 21-6. Register Call Summary....5193
Table 21-7. Subprocess Call Summary....5193
Table 21-8. Spinlock Instance Summary....5194
Table 21-9. Spinlock Registers Mapping Summary....5194
Table 21-10. SPINLOCK_REVISION....5194
Table 21-11. SPINLOCK_SYSCONFIG....5194
Table 21-12. SPINLOCK_SYSTATUS....5195
Table 21-13. SPINLOCK_LOCK_REG_i....5196
Table 22-1. Input/Output Description....5202
Table 22-2. Integration Attributes....5205
Table 22-3. Clocks and Resets....5205
Table 22-4. GP Timers Hardware Requests....5206
Table 22-5. IDLEMODE Settings....5211
Table 22-6. IDLEMODE Settings....5212
Table 22-7. Value Loaded in TCRR to Generate 1-ms Tick....5215
Table 22-8. Prescaler/Timer Reload Values Versus Contexts....5218
Table 22-9. Prescaler Clock Ratio Values....5220
Table 22-10. Value and Corresponding Interrupt Period....5221
Table 22-11. Global Initialization of Surrounding Modules....5226

Table 22-12. GP Timer Module Global Initialization....5226
Table 22-13. GP Timer Mode Configuration....5226
Table 22-14. GP Timer Compare Mode Configuration....5227
Table 22-15. GP Timer Capture Mode Configuration....5227
Table 22-16. Initialize Capture Mode....5227
Table 22-17. Detect Event....5227
Table 22-18. GP Timer PWM Mode Configuration....5228
Table 22-19. GP Timer Instance Summary....5229
Table 22-20. TIMER1, TIMER2, and TIMER10 Register Mapping Summary....5230
Table 22-21. TIMER3 and TIMER4 Register Mapping Summary....5230
Table 22-22. TIMER5, TIMER6 and TIMER7 Register Mapping Summary....5231
Table 22-23. TIMER8, TIMER9 and TIMER11 Register Mapping Summary....5231
Table 22-24. TIMER13, TIMER14 and TIMER15 Register Mapping Summary....5232
Table 22-25. TIMER16 and TIMER12 Register Mapping Summary....5232
Table 22-26. TIDR....5232
Table 22-27. TIOCP_CFG....5233
Table 22-28. IRQ_EOI....5234
Table 22-29. IRQSTATUS_RAW....5235
Table 22-30. IRQSTATUS....5236
Table 22-31. IRQENABLE_SET....5236
Table 22-32. IRQENABLE_CLR....5237
Table 22-33. IRQWAKEEN....5238
Table 22-34. TCLR....5239
Table 22-35. TCRR....5240
Table 22-36. TLDR....5241
Table 22-37. TTGR....5241
Table 22-38. TWPS....5242
Table 22-39. TMAR....5243
Table 22-40. TCAR1....5243
Table 22-41. TSICR....5244
Table 22-42. TCAR2....5245
Table 22-43. TPIR....5245
Table 22-44. TNIR....5246
Table 22-45. TCVR....5246
Table 22-46. TOCR....5246
Table 22-47. TOWR....5246
Table 22-48. IRQSTATUS_SET....5247
Table 22-49. IRQSTATUS_CLR....5247
Table 22-50. COUNTER_32K Integration Attributes....5250
Table 22-51. COUNTER_32K Clocks and Resets....5250
Table 22-52. COUNTER_32K Hardware Requests....5250
Table 22-53. COUNTER_32K Timer Instance Summary....5253
Table 22-54. COUNTER_32K Timer Register Summary....5253
Table 22-55. REVISION....5254
Table 22-56. Register Call Summary for Register REVISION....5254
Table 22-57. SYSCONFIG....5254
Table 22-58. Register Call Summary for Register SYSCONFIG....5254
Table 22-59. CR....5255
Table 22-60. Register Call Summary for Register CR....5255
Table 22-61. Watchdog Timer Default State....5256
Table 22-62. Watchdog Timer Integration Attributes....5258
Table 22-63. Watchdog Timer Clocks and Resets....5258
Table 22-64. Watchdog Timer Hardware Requests....5258
Table 22-65. IDLEMODE Settings....5260
Table 22-66. Watchdog Timer Events....5260
Table 22-67. Count and Prescaler Default Reset Values....5261
Table 22-68. Prescaler Clock Ratio Values....5262
Table 22-69. Reset Period Examples....5263
Table 22-70. Default Watchdog Timer Reset Periods....5263
Table 22-71. Global Initialization of Surrounding Modules....5266
Table 22-72. Watchdog Timer Module Global Initialization....5267

Table 22-73. Watchdog Timer Basic Configuration....5267
Table 22-74. Disable the Watchdog Timer....5267
Table 22-75. Enable the Watchdog Timer....5267
Table 22-76. Watchdog Timer Instance Summary....5268
Table 22-77. WD_TIMER2 Register Mapping Summary....5268
Table 22-78. WIDR....5270
Table 22-79. WDSC....5270
Table 22-80. WDST....5271
Table 22-81. WISR....5271
Table 22-82. WIER....5272
Table 22-83. WWER....5272
Table 22-84. WCLR....5273
Table 22-85. WCRR....5273
Table 22-86. WLDR....5273
Table 22-87. WTGR....5274
Table 22-88. WWPS....5274
Table 22-89. WDLY....5274
Table 22-90. WSPR....5275
Table 22-91. WIRQEOI....5275
Table 22-92. WIRQSTATRAW....5275
Table 22-93. WIRQSTAT....5276
Table 22-94. WIRQENSET....5276
Table 22-95. WIRQENCLR....5277
Table 22-96. WIRQWAKEEN....5277
Table 23-1. RTC External Signals....5282
Table 23-2. RTC Integration Attributes....5283
Table 23-3. RTC Clocks and Resets....5283
Table 23-4. RTC Hardware Requests....5283
Table 23-5. Interrupt Trigger Events....5286
Table 23-6. RTC Register Names and Values....5288
Table 23-7. RTC Idle Configuration....5292
Table 23-8. Global Initialization of Surrounding Modules....5293
Table 23-9. RTC Module Global Initialization....5293
Table 23-10. RTC Instance Summary....5294
Table 23-11. RTC_SS Registers Mapping Summary....5294
Table 23-12. RTC_SECONDS_REG....5294
Table 23-13. Register Call Summary for Register RTC_SECONDS_REG....5295
Table 23-14. RTC_MINUTES_REG....5295
Table 23-15. Register Call Summary for Register RTC_MINUTES_REG....5295
Table 23-16. RTC_HOURS_REG....5295
Table 23-17. Register Call Summary for Register RTC_HOURS_REG....5296
Table 23-18. RTC_DAYS_REG....5296
Table 23-19. Register Call Summary for Register RTC_DAYS_REG....5296
Table 23-20. RTC_MONTHS_REG....5297
Table 23-21. Register Call Summary for Register RTC_MONTHS_REG....5297
Table 23-22. RTC_YEARS_REG....5297
Table 23-23. Register Call Summary for Register RTC_YEARS_REG....5297
Table 23-24. RTC_WEEKS_REG....5298
Table 23-25. Register Call Summary for Register RTC_WEEKS_REG....5298
Table 23-26. RTC_ALARM_SECONDS_REG....5298
Table 23-27. Register Call Summary for Register RTC_ALARM_SECONDS_REG....5298
Table 23-28. RTC_ALARM_MINUTES_REG....5298
Table 23-29. Register Call Summary for Register RTC_ALARM_MINUTES_REG....5299
Table 23-30. RTC_ALARM_HOURS_REG....5299
Table 23-31. Register Call Summary for Register RTC_ALARM_HOURS_REG....5299
Table 23-32. RTC_ALARM_DAYS_REG....5300
Table 23-33. Register Call Summary for Register RTC_ALARM_DAYS_REG....5300
Table 23-34. RTC_ALARM_MONTHS_REG....5300
Table 23-35. Register Call Summary for Register RTC_ALARM_MONTHS_REG....5300
Table 23-36. RTC_ALARM_YEARS_REG....5301
Table 23-37. Register Call Summary for Register RTC_ALARM_YEARS_REG....5301

Table 23-38. RTC_CTRL_REG....5301
Table 23-39. Register Call Summary for Register RTC_CTRL_REG....5302
Table 23-40. RTC_STATUS_REG....5302
Table 23-41. Register Call Summary for Register RTC_STATUS_REG....5303
Table 23-42. RTC_INTERRUPTS_REG....5303
Table 23-43. Register Call Summary for Register RTC_INTERRUPTS_REG....5303
Table 23-44. RTC_COMP_LSB_REG....5304
Table 23-45. Register Call Summary for Register RTC_COMP_LSB_REG....5304
Table 23-46. RTC_COMP_MSB_REG....5304
Table 23-47. Register Call Summary for Register RTC_COMP_MSB_REG....5304
Table 23-48. RTC_OSC_REG....5304
Table 23-49. Register Call Summary for Register RTC_OSC_REG....5305
Table 23-50. RTC_SCRATCH0_REG....5305
Table 23-51. Register Call Summary for Register RTC_SCRATCH0_REG....5306
Table 23-52. RTC_SCRATCH1_REG....5306
Table 23-53. Register Call Summary for Register RTC_SCRATCH1_REG....5306
Table 23-54. RTC_SCRATCH2_REG....5306
Table 23-55. Register Call Summary for Register RTC_SCRATCH2_REG....5306
Table 23-56. RTC_KICK0_REG....5306
Table 23-57. Register Call Summary for Register RTC_KICK0_REG....5307
Table 23-58. RTC_KICK1_REG....5307
Table 23-59. Register Call Summary for Register RTC_KICK1_REG....5307
Table 23-60. RTC_REVISION_REG....5307
Table 23-61. Register Call Summary for Register RTC_REVISION_REG....5308
Table 23-62. RTC_SYSCONFIG_REG....5308
Table 23-63. Register Call Summary for Register RTC_SYSCONFIG_REG....5308
Table 23-64. RTC_IRQWAKEEN....5308
Table 23-65. Register Call Summary for Register RTC_IRQWAKEEN....5309
Table 23-66. RTC_ALARM2_SECONDS_REG....5309
Table 23-67. Register Call Summary for Register RTC_ALARM2_SECONDS_REG....5309
Table 23-68. RTC_ALARM2_MINUTES_REG....5310
Table 23-69. Register Call Summary for Register RTC_ALARM2_MINUTES_REG....5310
Table 23-70. RTC_ALARM2_HOURS_REG....5310
Table 23-71. Register Call Summary for Register RTC_ALARM2_HOURS_REG....5310
Table 23-72. RTC_ALARM2_DAYS_REG....5311
Table 23-73. Register Call Summary for Register RTC_ALARM2_DAYS_REG....5311
Table 23-74. RTC_ALARM2_MONTHS_REG....5311
Table 23-75. Register Call Summary for Register RTC_ALARM2_MONTHS_REG....5311
Table 23-76. RTC_ALARM2_YEARS_REG....5312
Table 23-77. Register Call Summary for Register RTC_ALARM2_YEARS_REG....5312
Table 23-78. RTC_PMIC_REG....5312
Table 23-79. Register Call Summary for Register RTC_PMIC_REG....5313
Table 23-80. RTC_RTL_DEBOUNCE_REG....5313
Table 23-81. Register Call Summary for Register RTC_RTL_DEBOUNCE_REG....5314
Table 24-1. HS I^{2}C Input/Output....5319
Table 24-2. HS I^{2}C Integration Attributes....5326
Table 24-3. HS I^{2}C Clocks and Resets....5326
Table 24-4. HS I^{2}C Hardware Requests....5326
Table 24-5. HS I^{2}C Operation Mode Selection....5329
Table 24-6. HS I^{2}C t_{LOW} and t_{high}Values of the I^{2}C Clock....5330
Table 24-7. HS I^{2}C Register Values for Maximum I^{2}C Bit Rates in I^{2}C F/S, I^{2}C HS Modes....5330
Table 24-8. HS I^{2}C Local Power-Management Features....5332
Table 24-9. HS I^{2}C Clock Activity Settings....5332
Table 24-10. HS I^{2}C Events....5333
Table 24-11. HS I^{2}C DMA Requests....5334
Table 24-12. HS I^{2}C List of Tests....5339
Table 24-13. Subprocess Call Summary for Sequence – I^{2}C Setup Procedure....5344
Table 24-14. HS I^{2}C Register Call Summary for Sequence – Setup Procedure....5344
Table 24-15. HS I^{2}C Register Call Summary for Sequence – Master Transmitter Mode, Polling Method, in F/S and HS Modes....5345

Table 24-16. HS I^{2}C Register Call Summary for Sequence – Master Receiver Mode, Polling Method, in F/S and HS Modes........ 5346
Table 24-17. HS I^{2}C Register Call Summary for Sequence – Master Transmitter Mode, Interrupt Method, in F/S and HS Modes........ 5348
Table 24-18. HS I^{2}C Register Call Summary for Sequence – Master Receiver Mode, Interrupt Method, in F/S and HS Modes........ 5349
Table 24-19. HS I^{2}C Register Call Summary for Sequence – Master Transmitter Mode, DMA Method in F/S and HS Modes........ 5351
Table 24-20. HS I^{2}C Register Call Summary for Sequence – Master Receiver Mode, DMA Method in F/S and HS Modes 5353
Table 24-21. HS I^{2}C Register Call Summary for Sequence – Slave Transmitter/Receiver Mode, Polling........ 5353
Table 24-22. HS I^{2}C Register Call Summary for Sequence – Slave Transmitter/Receiver Mode, Interrupt........ 5354
Table 24-23. HS I^{2}C Instance Summary........ 5355
Table 24-24. HS I^{2}C Registers Mapping Summary 1........ 5355
Table 24-25. HS I^{2}C Registers Mapping Summary 2........ 5356
Table 24-26. I2C_REVNB_LO........ 5357
Table 24-27. I2C_REVNB_HI........ 5357
Table 24-28. I2C_SYSC........ 5357
Table 24-29. I2C_EOI........ 5358
Table 24-30. I2C_IRQSTATUS_RAW........ 5358
Table 24-31. I2C_IRQSTATUS........ 5360
Table 24-32. I2C_IRQENABLE_SET........ 5362
Table 24-33. I2C_IRQENABLE_CLR........ 5365
Table 24-34. I2C_WE........ 5367
Table 24-35. I2C_DMARXENABLE_SET........ 5369
Table 24-36. I2C_DMATXENABLE_SET........ 5369
Table 24-37. I2C_DMARXENABLE_CLR........ 5369
Table 24-38. I2C_DMATXENABLE_CLR........ 5370
Table 24-39. I2C_DMARXWAKE_EN........ 5370
Table 24-40. I2C_DMATXWAKE_EN........ 5371
Table 24-41. I2C_SYSS........ 5372
Table 24-42. I2C_BUF........ 5373
Table 24-43. I2C_CNT........ 5373
Table 24-44. I2C_DATA........ 5374
Table 24-45. I2C_CON........ 5374
Table 24-46. I2C_OA........ 5375
Table 24-47. I2C_SA........ 5375
Table 24-48. I2C_PSC........ 5376
Table 24-49. I2C_SCLL........ 5376
Table 24-50. I2C_SCLH........ 5376
Table 24-51. I2C_SYSTEST........ 5377
Table 24-52. I2C_BUFSTAT........ 5378
Table 24-53. I2C_OA1........ 5379
Table 24-54. I2C_OA2........ 5379
Table 24-55. I2C_OA3........ 5379
Table 24-56. I2C_ACTOA........ 5380
Table 24-57. I2C_SBLOCK........ 5380
Table 24-58. Functional Modes........ 5382
Table 24-59. I/O Description........ 5383
Table 24-60. HDQ/1-Wire Command Byte........ 5385
Table 24-61. HDQ1W Integration Attributes........ 5386
Table 24-62. HDQ1W Clocks and Resets........ 5386
Table 24-63. HDQ1W Hardware Requests........ 5386
Table 24-64. Local Power-Management Features........ 5390
Table 24-65. Events........ 5391
Table 24-66. Global Initialization of Surrounding Modules........ 5394
Table 24-67. HDQ1W Module Global Initialization........ 5394
Table 24-68. HDQ Mode Selection........ 5394
Table 24-69. HDQ Write Operation Mode........ 5394
Table 24-70. HDQ Read Operation Mode........ 5395
Table 24-71. Initialize HDQ Slave........ 5395
Table 24-72. 1-Wire Mode Selection........ 5395

Table 24-73. 1-Wire Write Operation Mode....5395
Table 24-74. 1-Wire Read Operation Mode....5395
Table 24-75. Initialize 1-Wire Slave....5396
Table 24-76. HDQ1W Instance Summary....5396
Table 24-77. HDQ1W Registers Mapping Summary....5396
Table 24-78. HDQ_REVISION....5397
Table 24-79. HDQ_TX_DATA....5397
Table 24-80. HDQ_RX_DATA....5397
Table 24-81. HDQ_CTRL_STATUS....5397
Table 24-82. HDQ_INT_STATUS....5398
Table 24-83. HDQ_SYSCONFIG....5399
Table 24-84. HDQ_SYSSTATUS....5399
Table 24-85. UART Interface Signals....5402
Table 24-86. IrDA I/O Signals....5404
Table 24-87. UART_EFR[1:0] IR Address Checking Options....5408
Table 24-88. 4-PPM Format....5410
Table 24-89. FIR Preamble, Start Flag, and Stop Flag....5410
Table 24-90. FIR Data Byte Transmission Order Example....5410
Table 24-91. CIR I/O Signals....5411
Table 24-92. UART Integration Attributes....5417
Table 24-93. UART Clocks and Resets....5417
Table 24-94. UART Hardware Requests....5418
Table 24-95. Local Power-Management Features....5423
Table 24-96. UART Mode Interrupts....5423
Table 24-97. IrDA Mode Interrupts....5424
Table 24-98. CIR Mode Interrupts....5425
Table 24-99. TX FIFO Trigger Level Setting Summary....5426
Table 24-100. RX FIFO Trigger-Level Setting Summary....5427
Table 24-101. UART/IrDA/CIR Register Access Mode Programming (Using UART_LCR)....5435
Table 24-102. Subconfiguration Mode A Summary....5436
Table 24-103. Subconfiguration Mode B Summary....5436
Table 24-104. Suboperational Mode Summary....5436
Table 24-105. UART/IrDA/CIR Register Access Mode Overview....5436
Table 24-106. UART Mode Selection....5437
Table 24-107. UART Mode Register Overview....5437
Table 24-108. IrDA Mode Register Overview....5438
Table 24-109. CIR Mode Register Overview....5440
Table 24-110. UART Baud Rate Settings (48-MHz Clock)....5441
Table 24-111. UART Baud Rate Settings (192-MHz Clock)....5442
Table 24-112. UART Parity Bit Encoding....5442
Table 24-113. UART_EFR[3:0] Software Flow Control Options....5443
Table 24-114. IrDA Baud Rate Settings....5448
Table 24-115. Duty Cycle....5451
Table 24-116. Global Initialization of Surrounding Modules for UART/IrDA/CIR....5453
Table 24-117. UART/IrDA/CIR Global Initialization....5453
Table 24-118. Configure Register Access Mode....5454
Table 24-119. Configure Register Access Submode TCR_TLR....5454
Table 24-120. Configure Register Access Submode MSR_SPR....5454
Table 24-121. Configure Register Access Submode XOFF....5454
Table 24-122. DMA Mode Settings....5454
Table 24-123. Load FIFO Triggers Defined by the FCR....5455
Table 24-124. Load FIFO Triggers Defined by the TLR....5455
Table 24-125. Load FIFO Triggers Defined by the Concatenated Value....5455
Table 24-126. Baud Rate Settings....5455
Table 24-127. Interrupt Settings....5455
Table 24-128. Protocol Settings....5456
Table 24-129. UART/IrDA/CIR Mode Selection....5456
Table 24-130. Hardware Flow Control Configuration....5456
Table 24-131. Software Flow Control Configuration....5456
Table 24-132. SIR Mode Receive Settings....5457
Table 24-133. SIR Mode Transmit Settings....5457

Table 24-134. MIR Mode Receive Settings....5458
Table 24-135. MIR Mode Transmit Settings....5458
Table 24-136. FIR Mode Receive Settings....5458
Table 24-137. FIR Mode Transmit Settings....5459
Table 24-138. UART/IrDA/CIR Instance Summary....5460
Table 24-139. UART/IrDA/CIR Registers Mapping Summary 1....5460
Table 24-140. UART/IrDA/CIR Registers Mapping Summary 2....5461
Table 24-141. UART/IrDA/CIR Registers Mapping Summary 3....5463
Table 24-142. UART/IrDA/CIR Registers Mapping Summary 4....5464
Table 24-143. UART_THR....5465
Table 24-144. UART_RHR....5466
Table 24-145. UART_DLL....5466
Table 24-146. UART_IER....5466
Table 24-147. UART_IER_IRDA....5467
Table 24-148. UART_IER_CIR....5468
Table 24-149. UART_DLH....5469
Table 24-150. UART_IIR....5470
Table 24-151. UART_IIR_IRDA....5470
Table 24-152. UART_IIR_CIR....5471
Table 24-153. UART_FCR....5472
Table 24-154. UART_EFR....5473
Table 24-155. UART_LCR....5474
Table 24-156. UART_XON1_ADDR1....5475
Table 24-157. UART_MCR....5476
Table 24-158. UART_LSR....5477
Table 24-159. UART_LSR_IRDA....5478
Table 24-160. UART_LSR_CIR....5479
Table 24-161. UART_XON2_ADDR2....5480
Table 24-162. UART_TCR....5480
Table 24-163. UART_XOFF1....5481
Table 24-164. UART_MSR....5481
Table 24-165. UART_SPR....5482
Table 24-166. UART_TLR....5482
Table 24-167. UART_XOFF2....5483
Table 24-168. UART_MDR1....5483
Table 24-169. UART_MDR2....5484
Table 24-170. UART_SFLSR....5486
Table 24-171. UART_TXFLL....5486
Table 24-172. UART_RESUME....5487
Table 24-173. UART_TXFLH....5487
Table 24-174. UART_SFREGL....5488
Table 24-175. UART_RXFLL....5488
Table 24-176. UART_SFREGH....5489
Table 24-177. UART_RXFLH....5489
Table 24-178. UART_BLR....5490
Table 24-179. UART_UASR....5491
Table 24-180. UART_ACREG....5492
Table 24-181. UART_SCR....5493
Table 24-182. UART_SSR....5494
Table 24-183. UART_EBLR....5495
Table 24-184. UART_MVR....5495
Table 24-185. UART_SYSC....5496
Table 24-186. UART_SYSS....5497
Table 24-187. UART_WER....5497
Table 24-188. UART_CFPS....5499
Table 24-189. UART_RXFIFO_LVL....5499
Table 24-190. UART_TXFIFO_LVL....5500
Table 24-191. UART_IER2....5500
Table 24-192. UART_ISR2....5501
Table 24-193. UART_FREQ_SEL....5501
Table 24-194. UART_MDR3....5502

Table 24-195. UART_TX_DMA_THRESHOLD....5502
Table 24-196. McSPI I/O Description (Master Mode)....5505
Table 24-197. McSPI I/O Description (Slave Mode)....5506
Table 24-198. SPI Master Clock Rates....5507
Table 24-199. Phase and Polarity Combinations....5507
Table 24-200. McSPI Integration Attributes....5514
Table 24-201. McSPI Clocks and Resets....5514
Table 24-202. McSPI Hardware Requests....5514
Table 24-203. SPI Master Clock Rates....5522
Table 24-204. CLKSPIO High/Low Time Computation....5523
Table 24-205. Clock Granularity Examples....5524
Table 24-206. FIFO Writes, Word Length Relationship....5529
Table 24-207. Smart-Idle Mode and Wake-Up Capabilities....5536
Table 24-208. Global Initialization of Surrounding Modules....5538
Table 24-209. McSPI Global Initialization....5538
Table 24-210. McSPI Receive Mode Initialization....5538
Table 24-211. McSPI Transmit Mode Initialization....5539
Table 24-212. McSPI Transmit-and-Receive Mode Initialization....5539
Table 24-213. Common Transfer Sequence (Main Process)....5539
Table 24-214. End of Transfer Sequences....5540
Table 24-215. Transmit-and-Receive (Master and Slave) (Main Process)....5540
Table 24-216. Transmit-and-Receive (Master and Slave) (Interrupt Routine)....5540
Table 24-217. Transmit-Only With Interrupts (Master and Slave) (Main Process)....5541
Table 24-218. Transmit-Only With Interrupts (Master and Slave) (Interrupt Routine)....5541
Table 24-219. Transmit-Only With DMA (Master and Slave) (Main Process)....5541
Table 24-220. Transmit-Only With DMA (Master and Slave) (Interrupt Routine)....5541
Table 24-221. Receive-Only With Interrupt (Master Normal) (Main Process)....5541
Table 24-222. Receive-Only With Interrupt (Master Normal) (Interrupt Routine)....5542
Table 24-223. Receive-Only With DMA (Master Normal) (Main Process)....5542
Table 24-224. Receive-Only With DMA (Master Normal) (Interrupt Routine)....5542
Table 24-225. Receive-Only With Interrupt (Master Turbo) (Main Process)....5542
Table 24-226. Receive-Only With Interrupt (Master Turbo) (Interrupt Routine)....5543
Table 24-227. Receive-Only With DMA (Master Turbo) (Main Process)....5543
Table 24-228. Receive-Only With DMA (Master Turbo) (Interrupt Routine)....5543
Table 24-229. Receive-Only (Slave) (Main Process)....5544
Table 24-230. Receive-Only (Slave) (Interrupt Routine)....5544
Table 24-231. FIFO Mode Common Sequence (Master) (Main Process)....5544
Table 24-232. End of Transfer Sequences in FIFO Mode....5545
Table 24-233. Receive-Only Procedure – Polling Method....5550
Table 24-234. Receive-Only Procedure – Interrupt Method....5551
Table 24-235. Transmit-Only Procedure – Polling Method....5551
Table 24-236. Transmit-and-Receive Procedure – Polling Method....5551
Table 24-237. McSPI Instance Summary....5552
Table 24-238. McSPI Register Summary....5552
Table 24-239. MCSPI_HL_REV....5553
Table 24-240. Register Call Summary for Register MCSPI_HL_REV....5553
Table 24-241. MCSPI_HL_HWINFO....5553
Table 24-242. Register Call Summary for Register MCSPI_HL_HWINFO....5554
Table 24-243. MCSPI_HL_SYSCONFIG....5554
Table 24-244. Register Call Summary for Register MCSPI_HL_SYSCONFIG....5555
Table 24-245. MCSPI_REVISION....5555
Table 24-246. Register Call Summary for Register MCSPI_REVISION....5555
Table 24-247. MCSPI_SYSCONFIG....5555
Table 24-248. Register Call Summary for Register MCSPI_SYSCONFIG....5556
Table 24-249. MCSPI_SYSSTATUS....5556
Table 24-250. Register Call Summary for Register MCSPI_SYSSTATUS....5557
Table 24-251. MCSPI_IRQSTATUS....5557
Table 24-252. Register Call Summary for Register MCSPI_IRQSTATUS....5560
Table 24-253. MCSPI_IRQENABLE....5560
Table 24-254. Register Call Summary for Register MCSPI_IRQENABLE....5562
Table 24-255. MCSPI_WAKEUPENABLE....5562

Table 24-256. Register Call Summary for Register MCSPI_WAKEUPENABLE.... 5562
Table 24-257. MCSPI_SYST.... 5562
Table 24-258. Register Call Summary for Register MCSPI_SYST.... 5564
Table 24-259. MCSPI_MODULCTRL.... 5564
Table 24-260. Register Call Summary for Register MCSPI_MODULCTRL.... 5566
Table 24-261. MCSPI_CHxCONF.... 5566
Table 24-262. Register Call Summary for Register MCSPI_CHxCONF.... 5570
Table 24-263. MCSPI_CHxSTAT.... 5570
Table 24-264. Register Call Summary for Register MCSPI_CHxSTAT.... 5571
Table 24-265. MCSPI_CHxCTRL.... 5571
Table 24-266. Register Call Summary for Register MCSPI_CHxCTRL.... 5572
Table 24-267. MCSPI_TXx.... 5572
Table 24-268. Register Call Summary for Register MCSPI_TXx.... 5573
Table 24-269. MCSPI_RXx.... 5573
Table 24-270. Register Call Summary for Register MCSPI_RXx.... 5574
Table 24-271. MCSPI_XFERLEVEL.... 5574
Table 24-272. Register Call Summary for Register MCSPI_XFERLEVEL.... 5575
Table 24-273. MCSPI_DAFTX.... 5575
Table 24-274. Register Call Summary for Register MCSPI_DAFTX.... 5576
Table 24-275. MCSPI_DAFRX.... 5576
Table 24-276. Register Call Summary for Register MCSPI_DAFRX.... 5576
Table 24-277. QSPI I/O Signals.... 5578
Table 24-278. QSPI Integration Attributes.... 5580
Table 24-279. QSPI Clocks and Resets.... 5580
Table 24-280. QSPI Hardware Requests.... 5580
Table 24-281. SPI Clock Modes Definition.... 5585
Table 24-282. QSPI Events.... 5588
Table 24-283. QSPI Instance Summary.... 5590
Table 24-284. QSPI Registers Mapping Summary.... 5590
Table 24-285. QSPI_PID.... 5590
Table 24-286. Register Call Summary for Register QSPI_PID.... 5590
Table 24-287. QSPI_SYSCONFIG.... 5591
Table 24-288. Register Call Summary for Register QSPI_SYSCONFIG.... 5591
Table 24-289. QSPI_INTR_STATUS_RAW_SET.... 5591
Table 24-290. Register Call Summary for Register QSPI_INTR_STATUS_RAW_SET.... 5592
Table 24-291. QSPI_INTR_STATUS_ENABLED_CLEAR.... 5592
Table 24-292. Register Call Summary for Register QSPI_INTR_STATUS_ENABLED_CLEAR.... 5593
Table 24-293. QSPI_INTR_ENABLE_SET_REG.... 5593
Table 24-294. Register Call Summary for Register QSPI_INTR_ENABLE_SET_REG.... 5593
Table 24-295. QSPI_INTR_ENABLE_CLEAR_REG.... 5593
Table 24-296. Register Call Summary for Register QSPI_INTR_ENABLE_CLEAR_REG.... 5594
Table 24-297. QSPI_INTC_EOI_REG.... 5594
Table 24-298. Register Call Summary for Register QSPI_INTC_EOI_REG.... 5594
Table 24-299. QSPI_SPI_CLOCK_CNTRL_REG.... 5595
Table 24-300. Register Call Summary for Register QSPI_SPI_CLOCK_CNTRL_REG.... 5595
Table 24-301. QSPI_SPI_DC_REG.... 5595
Table 24-302. Register Call Summary for Register QSPI_SPI_DC_REG.... 5597
Table 24-303. QSPI_SPI_CMD_REG.... 5597
Table 24-304. Register Call Summary for Register QSPI_SPI_CMD_REG.... 5599
Table 24-305. QSPI_SPI_STATUS_REG.... 5599
Table 24-306. Register Call Summary for Register QSPI_SPI_STATUS_REG.... 5599
Table 24-307. QSPI_SPI_DATA_REG.... 5599
Table 24-308. Register Call Summary for Register QSPI_SPI_DATA_REG.... 5600
Table 24-309. QSPI_SPI_SETUP0_REG.... 5600
Table 24-310. Register Call Summary for Register QSPI_SPI_SETUP0_REG.... 5601
Table 24-311. QSPI_SPI_SETUP1_REG.... 5601
Table 24-312. Register Call Summary for Register QSPI_SPI_SETUP1_REG.... 5602
Table 24-313. QSPI_SPI_SETUP2_REG.... 5602
Table 24-314. Register Call Summary for Register QSPI_SPI_SETUP2_REG.... 5602
Table 24-315. QSPI_SPI_SETUP3_REG.... 5602
Table 24-316. Register Call Summary for Register QSPI_SPI_SETUP3_REG.... 5603

Table 24-317. QSPI_SPI_SWITCH_REG....5603
Table 24-318. Register Call Summary for Register QSPI_SPI_SWITCH_REG....5604
Table 24-319. QSPI_SPI_DATA_REG_1....5604
Table 24-320. Register Call Summary for Register QSPI_SPI_DATA_REG_1....5604
Table 24-321. QSPI_SPI_DATA_REG_2....5604
Table 24-322. Register Call Summary for Register QSPI_SPI_DATA_REG_2....5605
Table 24-323. QSPI_SPI_DATA_REG_3....5605
Table 24-324. Register Call Summary for Register QSPI_SPI_DATA_REG_3....5605
Table 24-325. McASP I/O Signals....5610
Table 24-326. Biphase-Mark Encoder....5619
Table 24-327. Preamble Codes....5620
Table 24-328. McASP Integration Attributes....5623
Table 24-329. McASP Clocks and Resets....5623
Table 24-330. McASP Hardware Requests....5624
Table 24-331. McASP TFU TDM Mode Settings....5638
Table 24-332. McASP TFU DIT-Mode Example Settings....5639
Table 24-333. McASP RFU Settings....5641
Table 24-334. Local Power-Management Features....5642
Table 24-335. Channel Status and User Data for Each DIT Block....5647
Table 24-336. TX Events....5656
Table 24-337. RX Events....5657
Table 24-338. Global Initialization of Surrounding Modules....5666
Table 24-339. McASP Transmitters Global Initialization for DIT-Mode Operation....5666
Table 24-340. Transmit Format Unit Configuration for DIT-Transmission....5667
Table 24-341. Transmit Frame-Synchronization Generator Configuration for DIT-Transmission....5668
Table 24-342. Transmit Clock Generator Configuration in DIT-Mode....5668
Table 24-343. McASP Pins Functional Configuration....5668
Table 24-344. DIT-Specific Subframe Fields Configuration....5668
Table 24-345. McASP Receivers Global Initialization for TDM-Mode Operation....5669
Table 24-346. Receive Format Unit Configuration for TDM-Reception....5670
Table 24-347. Receive Frame-Synchronization Generator Configuration for TDM-Reception....5670
Table 24-348. Receive Clock Generator Configuration....5671
Table 24-349. McASP Receiver Pins Functional Configuration....5672
Table 24-350. McASP Transmitters Global Initialization for TDM-Mode Operation....5672
Table 24-351. Transmit Format Unit Configuration for TDM-Transmission....5673
Table 24-352. Transmit Frame-Synchronization Generator Configuration for TDM-Transmission....5674
Table 24-353. Transmit Clock Generator Configuration for TDM Cases....5674
Table 24-354. McASP Transmit Pins Functional Configuration....5675
Table 24-355. Register Call Summary for Main Sequence – McASP DIT-/TDM- Transmission Polling Method....5677
Table 24-356. Subprocess Call Summary for Main Sequence – McASP DIT-/TDM- Transmission Polling Method....5677
Table 24-357. McASP DIT-/TDM- Interrupt Transmission Model....5679
Table 24-358. Register Call Summary for Subsequence – McASP DIT-/TDM- Transmission Startup Procedure....5679
Table 24-359. McASP DMA Transmission Model with Interrupt Events Servicing....5679
Table 24-360. Register Call Summary for Main Sequence – McASP Reception Polling Method....5681
Table 24-361. Subprocess Call Summary for Main Sequence – McASP Reception Polling Method....5682
Table 24-362. McASP TDM- Interrupt Reception Model....5684
Table 24-363. Register Call Summary for Subsequence – McASP TDM- Reception Startup Procedure....5684
Table 24-364. McASP DMA Reception Model with Interrupt Events Servicing....5684
Table 24-365. Register Call Summary for McASP Receive Interrupt Events Servicing....5688
Table 24-366. Subprocess Call Summary for Receive Interrupt Events Servicing....5688
Table 24-367. Register Call Summary for McASP Transmit Error Handling....5689
Table 24-368. Register Call Summary for McASP Receive Error Handling....5690
Table 24-369. McASP Instance Summary....5691
Table 24-370. MCASP_CFG Register Summary 1....5691
Table 24-371. MCASP_CFG Register Summary 2....5692
Table 24-372. MCASP_CFG Register Summary 3....5694
Table 24-373. MCASP_CFG Register Summary 4....5695
Table 24-374. MCASP_PID....5696
Table 24-375. PWRIDLESYSCONFIG....5697
Table 24-376. MCASP_PFUNC....5697
Table 24-377. MCASP_PDIR....5699

Table 24-378. MCASP_PDOUT....5701
Table 24-379. MCASP_PDIN....5704
Table 24-380. MCASP_PDSET....5706
Table 24-381. MCASP_PDCLR....5708
Table 24-382. MCASP_GBLCTL....5710
Table 24-383. MCASP_AMUTE....5712
Table 24-384. MCASP_LBCTL....5712
Table 24-385. MCASP_TXDITCTL....5713
Table 24-386. MCASP_GBLCTLR....5714
Table 24-387. MCASP_RXMASK....5715
Table 24-388. MCASP_RXFMT....5715
Table 24-389. MCASP_RXFMCTL....5717
Table 24-390. MCASP_ACLKRCTL....5718
Table 24-391. MCASP_AHCLKRCTL....5719
Table 24-392. MCASP_RXTDM....5720
Table 24-393. MCASP_EVTCTLR....5721
Table 24-394. MCASP_RXSTAT....5722
Table 24-395. MCASP_RXTDMSLOT....5723
Table 24-396. MCASP_RXCLKCHK....5724
Table 24-397. MCASP_REVTCTL....5725
Table 24-398. MCASP_GBLCTLX....5725
Table 24-399. MCASP_TXMASK....5726
Table 24-400. MCASP_TXFMT....5727
Table 24-401. MCASP_TXFMCTL....5728
Table 24-402. MCASP_ACLKXCTL....5729
Table 24-403. MCASP_AHCLKXCTL....5730
Table 24-404. MCASP_TXTDM....5731
Table 24-405. MCASP_EVTCTLX....5732
Table 24-406. MCASP_TXSTAT....5733
Table 24-407. MCASP_TXTDMSLOT....5734
Table 24-408. MCASP_TXCLKCHK....5735
Table 24-409. MCASP_XEVTCTL....5735
Table 24-410. MCASP_CLKADJEN....5736
Table 24-411. MCASP_DITCSRAi....5736
Table 24-412. MCASP_DITCSRBi....5737
Table 24-413. MCASP_DITUDRAi....5737
Table 24-414. MCASP_DITUDRBi....5738
Table 24-415. MCASP_XRSRCTLn....5738
Table 24-416. MCASP_TXBUFn....5739
Table 24-417. MCASP_RXBUFn....5739
Table 24-418. MCASP_AFIFO Register Summary 1....5740
Table 24-419. MCASP_AFIFO Register Summary 2....5740
Table 24-420. MCASP_AFIFO Register Summary 3....5740
Table 24-421. MCASP_AFIFO Register Summary 4....5740
Table 24-422. WFIFOCTL....5740
Table 24-423. WFIFOSTS....5741
Table 24-424. RFIFOCTL....5742
Table 24-425. RFIFOSTS....5743
Table 24-426. MCASP_DAT Register Summary 1....5743
Table 24-427. MCASP_DAT Register Summary 2....5744
Table 24-428. MCASP_DAT Register Summary 3....5744
Table 24-429. MCASP_DAT Register Summary 4....5744
Table 24-430. MCASP_RXBUF....5744
Table 24-431. MCASP_TXBUF....5745
Table 24-432. USB1 Input/Output Description....5752
Table 24-433. USB2 Input/Output Description....5752
Table 24-434. USB3 Input/Output Description....5752
Table 24-435. USB4 Input/Output Description....5752
Table 24-436. SuperSpeed USB Subsystem Integration Attributes....5758
Table 24-437. SuperSpeed USB Subsystem Clocks and Resets....5758
Table 24-438. SuperSpeed USB Subsystem Hardware Requests....5759

Table 24-439. SATA Controller I/O Signals....5765
Table 24-440. SATA Controller Integration Attributes....5767
Table 24-441. SATA Controller Clocks and Resets....5768
Table 24-442. SATA Controller Hardware Requests....5768
Table 24-443. SATA Primitives Summary....5771
Table 24-444. Local Power-Management Features....5776
Table 24-445. Interrupt Events Summary....5781
Table 24-446. Summary of SATA Host Subsystem Functional Registers....5788
Table 24-447. Global Initialization of Surrounding Modules....5789
Table 24-448. SATA Controller Global Initialization....5789
Table 24-449. Firmware Capability Writes....5791
Table 24-450. Prepare and Issue a Command....5791
Table 24-451. FIS Reception....5792
Table 24-452. SATA Instance Summary....5792
Table 24-453. DWC_ahsata Registers Mapping Summary....5792
Table 24-454. SATA_CAP....5793
Table 24-455. SATA_GHC....5795
Table 24-456. SATA_IS....5795
Table 24-457. SATA_PI....5796
Table 24-458. SATA_VS....5796
Table 24-459. SATA_CCC_CTL....5796
Table 24-460. SATA_CCC_PORTS....5797
Table 24-461. SATA_CAP2....5798
Table 24-462. SATA_BISTAFR....5798
Table 24-463. SATA_BISTCR....5799
Table 24-464. SATA_BISTFCTR....5801
Table 24-465. SATA_BISTSR....5801
Table 24-466. SATA_BISTDECR....5802
Table 24-467. SATA_OOBR....5802
Table 24-468. SATA_TIMER1MS....5803
Table 24-469. SATA_GPARAM1R....5803
Table 24-470. SATA_GPARAM2R....5804
Table 24-471. SATA_PPARAMR....5805
Table 24-472. SATA_TESTR....5806
Table 24-473. SATA_VERSIONR....5807
Table 24-474. SATA_IDR....5807
Table 24-475. SATA_PxCLB....5808
Table 24-476. SATA_PxCLBU....5808
Table 24-477. SATA_PxFB....5808
Table 24-478. SATA_PxFBU....5808
Table 24-479. SATA_PxIS....5809
Table 24-480. SATA_PxIE....5813
Table 24-481. SATA_PxCMD....5814
Table 24-482. SATA_PxTFD....5818
Table 24-483. SATA_PxSIG....5818
Table 24-484. SATA_PxSSTS....5819
Table 24-485. SATA_PxSCTL....5820
Table 24-486. SATA_PxSERR....5821
Table 24-487. SATA_PxSACT....5822
Table 24-488. SATA_PxCI....5822
Table 24-489. SATA_PxSNTF....5823
Table 24-490. SATA_PxDMACR....5823
Table 24-491. SATAMAC_wrapper Registers Mapping Summary....5824
Table 24-492. SATA_SYSCONFIG....5824
Table 24-493. SATA_CDRLOCK....5826
Table 24-494. PCIe_SS I/O Signals....5830
Table 24-495. PCIe_SS Port Configuration....5830
Table 24-496. PCIe Controllers Integration Attributes....5832
Table 24-497. PCIe Controllers Clocks and Resets....5832
Table 24-498. PCIe Controllers Hardware Requests....5832
Table 24-499. Power States Summary....5842

Table 24-500. Local Power-Management Features....5845
Table 24-501. PCIe Main Hardware Interrupt Events Summary....5849
Table 24-502. MSI Message Data Encoding (RC)....5853
Table 24-503. MSI Hardware Interrupt Events Summary....5854
Table 24-504. PCIe_SS Address Space Map....5854
Table 24-505. Address Translation Unit Region Access Summary....5855
Table 24-506. Summary of Access Types related to PCIe Controller Local Resources....5861
Table 24-507. PCIe Default BAR Configuration....5862
Table 24-508. BARn Register Access Methods (Mem BAR)....5863
Table 24-509. BARn Register Access Methods (IO BAR)....5863
Table 24-510. Global Initialization of the PCIe Surrounding Modules....5864
Table 24-511. Main Sequence PCIe Controller Global Initialization....5864
Table 24-512. Local EP (Type 0) PCIe Standard and Port Logic Registers Initialization Subsequence....5867
Table 24-513. RC (Type 1) PCIe Standard and Port Logic Registers Initialization Subsequence....5867
Table 24-514. PCIe Type-0 (EP) Configuration Standard Capability Registers vs PCIe Controller Hardware Registers Mapping....5868
Table 24-515. PCIe Type-0 (EP) Standard Configuration Header Registers vs PCIe Controller Hardware Registers Mapping....5868
Table 24-516. PCIe Type-1 (RC) Configuration Standard Capability Registers vs PCIe Controller Hardware Registers Mapping....5869
Table 24-517. PCIe Type-1 (RC) Standard Configuration Header Registers vs PCIe Controller Hardware Registers Mapping....5869
Table 24-518. PCIe_SS1 Instance Summary....5870
Table 24-519. PCIe_SS2 Instance Summary....5870
Table 24-520. PCIe_SS1_EP_CFG_PCIe Registers Mapping Summary....5870
Table 24-521. PCIe_SS2_EP_CFG_PCIe Registers Mapping Summary....5872
Table 24-522. PCIECTRL_EP_PCIEWIRE_DEVICE_VENDORID....5873
Table 24-523. PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_REGISTER....5873
Table 24-524. PCIECTRL_EP_PCIEWIRE_CLASSCODE_REVISIONID....5874
Table 24-525. PCIECTRL_EP_PCIEWIRE_BIST_HEAD_LAT_CACH....5874
Table 24-526. PCIECTRL_EP_PCIEWIRE_BAR0....5875
Table 24-527. PCIECTRL_EP_PCIEWIRE_BAR1....5876
Table 24-528. PCIECTRL_EP_PCIEWIRE_BAR2....5876
Table 24-529. PCIECTRL_EP_PCIEWIRE_BAR3....5877
Table 24-530. PCIECTRL_EP_PCIEWIRE_BAR4....5878
Table 24-531. PCIECTRL_EP_PCIEWIRE_BAR5....5878
Table 24-532. PCIECTRL_EP_PCIEWIRE_CARDBUS_CIS_POINTER....5879
Table 24-533. PCIECTRL_EP_PCIEWIRE_SUBID_SUBVENDORID....5879
Table 24-534. PCIECTRL_EP_PCIEWIRE_EXPANSION_ROM_BAR....5880
Table 24-535. PCIECTRL_EP_PCIEWIRE_CAPPTR....5880
Table 24-536. PCIECTRL_EP_PCIEWIRE_INTERRUPT....5880
Table 24-537. PCIECTRL_EP_PCIEWIRE_PM_CAP....5881
Table 24-538. PCIECTRL_EP_PCIEWIRE_PM_CSR....5881
Table 24-539. PCIECTRL_EP_PCIEWIRE_PCIE_CAP....5882
Table 24-540. PCIECTRL_EP_PCIEWIRE_DEV_CAP....5883
Table 24-541. PCIECTRL_EP_PCIEWIRE_DEV_CAS....5883
Table 24-542. PCIECTRL_EP_PCIEWIRE_LNK_CAP....5884
Table 24-543. PCIECTRL_EP_PCIEWIRE_LNK_CAS....5885
Table 24-544. PCIECTRL_EP_PCIEWIRE_DEV_CAP_2....5886
Table 24-545. PCIECTRL_EP_PCIEWIRE_DEV_CAS_2....5887
Table 24-546. PCIECTRL_EP_PCIEWIRE_LNK_CAP_2....5887
Table 24-547. PCIECTRL_EP_PCIEWIRE_LNK_CAS_2....5888
Table 24-548. PCIe_SS1_EP_CFG_DBICS Registers Mapping Summary....5889
Table 24-549. PCIe_SS2_EP_CFG_DBICS Registers Mapping Summary....5889
Table 24-550. PCIECTRL_EP_DBICS_DEVICE_VENDORID....5891
Table 24-551. PCIECTRL_EP_DBICS_STATUS_COMMAND_REGISTER....5891
Table 24-552. PCIECTRL_EP_DBICS_CLASSCODE_REVISIONID....5892
Table 24-553. PCIECTRL_EP_DBICS_BIST_HEAD_LAT_CACH....5892
Table 24-554. PCIECTRL_EP_DBICS_BAR0....5893
Table 24-555. PCIECTRL_EP_DBICS_BAR1....5893
Table 24-556. PCIECTRL_EP_DBICS_BAR2....5894

Table 24-557. PCIECTRL_EP_DBICS_BAR3........ 5895
Table 24-558. PCIECTRL_EP_DBICS_BAR4........ 5895
Table 24-559. PCIECTRL_EP_DBICS_BAR5........ 5896
Table 24-560. PCIECTRL_EP_DBICS_CARDBUS_CIS_POINTER........ 5897
Table 24-561. PCIECTRL_EP_DBICS_SUBID_SUBVENDORID........ 5897
Table 24-562. PCIECTRL_EP_DBICS_EXPANSION_ROM_BAR........ 5897
Table 24-563. PCIECTRL_EP_DBICS_CAPPTR........ 5898
Table 24-564. PCIECTRL_EP_DBICS_INTERRUPT........ 5898
Table 24-565. PCIECTRL_EP_DBICS_PM_CAP........ 5898
Table 24-566. PCIECTRL_EP_DBICS_PM_CSR........ 5899
Table 24-567. PCIECTRL_EP_DBICS_MSI_CAP........ 5900
Table 24-568. PCIECTRL_EP_DBICS_MSI_ADD_L32........ 5900
Table 24-569. PCIECTRL_EP_DBICS_MSI_ADD_U32........ 5900
Table 24-570. PCIECTRL_EP_DBICS_MSI_DATA........ 5901
Table 24-571. PCIECTRL_EP_DBICS_PCIE_CAP........ 5901
Table 24-572. PCIECTRL_EP_DBICS_DEV_CAP........ 5901
Table 24-573. PCIECTRL_EP_DBICS_DEV_CAS........ 5902
Table 24-574. PCIECTRL_EP_DBICS_LNK_CAP........ 5903
Table 24-575. PCIECTRL_EP_DBICS_LNK_CAS........ 5904
Table 24-576. PCIECTRL_EP_DBICS_DEV_CAP_2........ 5905
Table 24-577. PCIECTRL_EP_DBICS_DEV_CAS_2........ 5905
Table 24-578. PCIECTRL_EP_DBICS_LNK_CAP_2........ 5906
Table 24-579. PCIECTRL_EP_DBICS_LNK_CAS_2........ 5906
Table 24-580. PCIe_SS1_RC_CFG_DBICS Registers Mapping Summary........ 5908
Table 24-581. PCIe_SS2_RC_CFG_DBICS Registers Mapping Summary........ 5908
Table 24-582. PCIECTRL_RC_DBICS_DEVICE_VENDORID........ 5909
Table 24-583. PCIECTRL_RC_DBICS_STATUS_COMMAND_REGISTER........ 5909
Table 24-584. PCIECTRL_RC_DBICS_CLASSCODE_REVISIONID........ 5910
Table 24-585. PCIECTRL_RC_DBICS_BIST_HEAD_LAT_CACH........ 5911
Table 24-586. PCIECTRL_RC_DBICS_BAR0........ 5911
Table 24-587. PCIECTRL_RC_DBICS_BAR1........ 5912
Table 24-588. PCIECTRL_RC_DBICS_BUS_NUM_REG........ 5913
Table 24-589. PCIECTRL_RC_DBICS_IOBASE_LIMIT_SEC_STATUS........ 5913
Table 24-590. PCIECTRL_RC_DBICS_MEM_BASE_LIMIT........ 5914
Table 24-591. PCIECTRL_RC_DBICS_PREF_MEM_BASE_LIMIT........ 5914
Table 24-592. PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_BASEADDR........ 5915
Table 24-593. PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_LIMITADDR........ 5915
Table 24-594. PCIECTRL_RC_DBICS_IO_BASE_LIMIT........ 5915
Table 24-595. PCIECTRL_RC_DBICS_CAPPTR........ 5915
Table 24-596. PCIECTRL_RC_DBICS_EXPANSION_ROM_BAR........ 5916
Table 24-597. PCIECTRL_RC_DBICS_BRIDGE_INT........ 5916
Table 24-598. PCIECTRL_RC_DBICS_PCIE_CAP........ 5917
Table 24-599. PCIECTRL_RC_DBICS_DEV_CAP........ 5917
Table 24-600. PCIECTRL_RC_DBICS_DEV_CAS........ 5918
Table 24-601. PCIECTRL_RC_DBICS_LNK_CAP........ 5919
Table 24-602. PCIECTRL_RC_DBICS_LNK_CAS........ 5919
Table 24-603. PCIECTRL_RC_DBICS_SLOT_CAP........ 5920
Table 24-604. PCIECTRL_RC_DBICS_SLOT_CAS........ 5921
Table 24-605. PCIECTRL_RC_DBICS_ROOT_CAC........ 5922
Table 24-606. PCIECTRL_RC_DBICS_ROOT_STS........ 5922
Table 24-607. PCIECTRL_RC_DBICS_DEV_CAP_2........ 5922
Table 24-608. PCIECTRL_RC_DBICS_DEV_CAS_2........ 5923
Table 24-609. PCIECTRL_RC_DBICS_LNK_CAP_2........ 5924
Table 24-610. PCIECTRL_RC_DBICS_LNK_CAS_2........ 5924
Table 24-611. PCIe_SS1_PL_CONF Registers Mapping Summary........ 5926
Table 24-612. PCIe_SS2_PL_CONF Registers Mapping Summary........ 5927
Table 24-613. PCIECTRL_PL_LAT_REL_TIM........ 5928
Table 24-614. PCIECTRL_PL_VENDOR_SPECIFIC_DLLP........ 5929
Table 24-615. PCIECTRL_PL_PT_LNK_R........ 5929
Table 24-616. PCIECTRL_PL_ACK_FREQ_ASPM........ 5930
Table 24-617. PCIECTRL_PL_PT_LNK_CTRL_R........ 5930

Table 24-618. PCIECTRL_PL_LN_SKW_R.......... 5931
Table 24-619. PCIECTRL_PL_SYMB_N_R.......... 5932
Table 24-620. PCIECTRL_PL_SYMB_T_R.......... 5932
Table 24-621. PCIECTRL_PL_FL_MSK_R2.......... 5932
Table 24-622. PCIECTRL_PL_OBNP_SUBREQ_CTRL.......... 5932
Table 24-623. PCIECTRL_PL_TR_P_STS_R.......... 5933
Table 24-624. PCIECTRL_PL_TR_NP_STS_R.......... 5933
Table 24-625. PCIECTRL_PL_TR_C_STS_R.......... 5933
Table 24-626. PCIECTRL_PL_Q_STS_R.......... 5934
Table 24-627. PCIECTRL_PL_VC_TR_A_R1.......... 5934
Table 24-628. PCIECTRL_PL_VC_TR_A_R2.......... 5934
Table 24-629. PCIECTRL_PL_VC0_PR_Q_C.......... 5935
Table 24-630. PCIECTRL_PL_VC0_NPR_Q_C.......... 5935
Table 24-631. PCIECTRL_PL_VC0_CR_Q_C.......... 5936
Table 24-632. PCIECTRL_PL_WIDTH_SPEED_CTL.......... 5936
Table 24-633. PCIECTRL_PL_PHY_STS_R.......... 5937
Table 24-634. PCIECTRL_PL_PHY_CTRL_R.......... 5937
Table 24-635. PCIECTRL_PL_MSI_CTRL_ADDRESS.......... 5937
Table 24-636. PCIECTRL_PL_MSI_CTRL_UPPER_ADDRESS.......... 5938
Table 24-637. PCIECTRL_PL_MSI_CTRL_INT_ENABLE_N.......... 5938
Table 24-638. PCIECTRL_PL_MSI_CTRL_INT_MASK_N.......... 5938
Table 24-639. PCIECTRL_PL_MSI_CTRL_INT_STATUS_N.......... 5938
Table 24-640. PCIECTRL_PL_MSI_CTRL_GPIO.......... 5939
Table 24-641. PCIECTRL_PL_PIPE_LOOPBACK.......... 5939
Table 24-642. PCIECTRL_PL_DBI_RO_WR_EN.......... 5939
Table 24-643. PCIECTRL_PL_AXIS_SLV_ERR_RESP.......... 5940
Table 24-644. PCIECTRL_PL_AXIS_SLV_TIMEOUT.......... 5940
Table 24-645. PCIECTRL_PL_IATU_INDEX.......... 5941
Table 24-646. PCIECTRL_PL_IATU_REG_CTRL_1.......... 5941
Table 24-647. PCIECTRL_PL_IATU_REG_CTRL_2.......... 5942
Table 24-648. PCIECTRL_PL_IATU_REG_LOWER_BASE.......... 5944
Table 24-649. PCIECTRL_PL_IATU_REG_UPPER_BASE.......... 5944
Table 24-650. PCIECTRL_PL_IATU_REG_LIMIT.......... 5944
Table 24-651. PCIECTRL_PL_IATU_REG_LOWER_TARGET.......... 5944
Table 24-652. PCIECTRL_PL_IATU_REG_UPPER_TARGET.......... 5945
Table 24-653. PCIECTRL_PL_IATU_REG_CTRL_3.......... 5945
Table 24-654. PCIe_SS1_EP_CFG_DBI_CS2 Registers Mapping Summary.......... 5946
Table 24-655. PCIe_SS2_EP_CFG_DBI_CS2 Registers Mapping Summary.......... 5946
Table 24-656. PCIECTRL_EP_DBICS2_DEVICE_VENDORID.......... 5948
Table 24-657. PCIECTRL_EP_DBICS2_STATUS_COMMAND_REGISTER.......... 5948
Table 24-658. PCIECTRL_EP_DBICS2_CLASSCODE_REVISIONID.......... 5949
Table 24-659. PCIECTRL_EP_DBICS2_BIST_HEAD_LAT_CACH.......... 5949
Table 24-660. PCIECTRL_EP_DBICS2_BAR0_MASK.......... 5950
Table 24-661. PCIECTRL_EP_DBICS2_BAR1_MASK.......... 5950
Table 24-662. PCIECTRL_EP_DBICS2_BAR2_MASK.......... 5950
Table 24-663. PCIECTRL_EP_DBICS2_BAR3_MASK.......... 5951
Table 24-664. PCIECTRL_EP_DBICS2_BAR4_MASK.......... 5951
Table 24-665. PCIECTRL_EP_DBICS2_BAR5_MASK.......... 5951
Table 24-666. PCIECTRL_EP_DBICS2_CARDBUS_CIS_POINTER.......... 5952
Table 24-667. PCIECTRL_EP_DBICS2_SUBID_SUBVENDORID.......... 5952
Table 24-668. PCIECTRL_EP_DBICS2_EXPANSION_ROM_BAR.......... 5952
Table 24-669. PCIECTRL_EP_DBICS2_CAPPTR.......... 5953
Table 24-670. PCIECTRL_EP_DBICS2_INTERRUPT.......... 5953
Table 24-671. PCIECTRL_EP_DBICS2_PM_CAP.......... 5953
Table 24-672. PCIECTRL_EP_DBICS2_PM_CSR.......... 5954
Table 24-673. PCIECTRL_EP_DBICS2_PCIE_CAP.......... 5955
Table 24-674. PCIECTRL_EP_DBICS2_DEV_CAP.......... 5955
Table 24-675. PCIECTRL_EP_DBICS2_DEV_CAS.......... 5956
Table 24-676. PCIECTRL_EP_DBICS2_LNK_CAP.......... 5957
Table 24-677. PCIECTRL_EP_DBICS2_LNK_CAS.......... 5957
Table 24-678. PCIECTRL_EP_DBICS2_DEV_CAP_2.......... 5958

Table 24-679. PCIECTRL_EP_DBICS2_DEV_CAS_2....5959
Table 24-680. PCIECTRL_EP_DBICS2_LNK_CAP_2....5960
Table 24-681. PCIECTRL_EP_DBICS2_LNK_CAS_2....5960
Table 24-682. PCIe_SS1_RC_CFG_DBI_CS2 Registers Mapping Summary....5962
Table 24-683. PCIe_SS2_RC_CFG_DBI_CS2 Registers Mapping Summary....5962
Table 24-684. PCIECTRL_RC_DBICS2_DEVICE_VENDORID....5963
Table 24-685. PCIECTRL_RC_DBICS2_STATUS_COMMAND_REGISTER....5963
Table 24-686. PCIECTRL_RC_DBICS2_CLASSCODE_REVISIONID....5964
Table 24-687. PCIECTRL_RC_DBICS2_BIST_HEAD_LAT_CACH....5965
Table 24-688. PCIECTRL_RC_DBICS2_BAR0_MASK....5965
Table 24-689. PCIECTRL_RC_DBICS2_BAR1_MASK....5965
Table 24-690. PCIECTRL_RC_DBICS2_BUS_NUM_REG....5966
Table 24-691. PCIECTRL_RC_DBICS2_IOBASE_LIMIT_SEC_STATUS....5966
Table 24-692. PCIECTRL_RC_DBICS2_MEM_BASE_LIMIT....5967
Table 24-693. PCIECTRL_RC_DBICS2_PREF_MEM_BASE_LIMIT....5967
Table 24-694. PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_BASEADDR....5968
Table 24-695. PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_LIMITADDR....5968
Table 24-696. PCIECTRL_RC_DBICS2_IO_BASE_LIMIT....5969
Table 24-697. PCIECTRL_RC_DBICS2_CAPPTR....5969
Table 24-698. PCIECTRL_RC_DBICS2_EXPANSION_ROM_BAR....5969
Table 24-699. PCIECTRL_RC_DBICS2_BRIDGE_INT....5969
Table 24-700. PCIECTRL_RC_DBICS2_PCIE_CAP....5970
Table 24-701. PCIECTRL_RC_DBICS2_DEV_CAP....5971
Table 24-702. PCIECTRL_RC_DBICS2_DEV_CAS....5971
Table 24-703. PCIECTRL_RC_DBICS2_LNK_CAP....5972
Table 24-704. PCIECTRL_RC_DBICS2_LNK_CAS....5973
Table 24-705. PCIECTRL_RC_DBICS2_SLOT_CAP....5974
Table 24-706. PCIECTRL_RC_DBICS2_SLOT_CAS....5974
Table 24-707. PCIECTRL_RC_DBICS2_ROOT_CAC....5975
Table 24-708. PCIECTRL_RC_DBICS2_ROOT_STS....5975
Table 24-709. PCIECTRL_RC_DBICS2_DEV_CAP_2....5976
Table 24-710. PCIECTRL_RC_DBICS2_DEV_CAS_2....5977
Table 24-711. PCIECTRL_RC_DBICS2_LNK_CAP_2....5977
Table 24-712. PCIECTRL_RC_DBICS2_LNK_CAS_2....5978
Table 24-713. PCIe_SS1_TI_CONF Registers Mapping Summary....5979
Table 24-714. PCIe_SS2_TI_CONF Registers Mapping Summary....5979
Table 24-715. PCIECTRL_TI_CONF_REVISION....5980
Table 24-716. PCIECTRL_TI_CONF_SYSCONFIG....5980
Table 24-717. PCIECTRL_TI_CONF_IRQ_EOI....5981
Table 24-718. PCIECTRL_TI_CONF_IRQSTATUS_RAW_MAIN....5982
Table 24-719. PCIECTRL_TI_CONF_IRQSTATUS_MAIN....5983
Table 24-720. PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN....5985
Table 24-721. PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN....5986
Table 24-722. PCIECTRL_TI_CONF_IRQSTATUS_RAW_MSI....5988
Table 24-723. PCIECTRL_TI_CONF_IRQSTATUS_MSI....5989
Table 24-724. PCIECTRL_TI_CONF_IRQENABLE_SET_MSI....5990
Table 24-725. PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI....5990
Table 24-726. PCIECTRL_TI_CONF_DEVICE_TYPE....5991
Table 24-727. PCIECTRL_TI_CONF_DEVICE_CMD....5992
Table 24-728. PCIECTRL_TI_CONF_PM_CTRL....5993
Table 24-729. PCIECTRL_TI_CONF_PHY_CS....5994
Table 24-730. PCIECTRL_TI_CONF_INTX_ASSERT....5994
Table 24-731. PCIECTRL_TI_CONF_INTX_DEASSERT....5995
Table 24-732. PCIECTRL_TI_CONF_MSI_XMT....5995
Table 24-733. PCIECTRL_TI_CONF_DEBUG_CFG....5996
Table 24-734. PCIECTRL_TI_CONF_DEBUG_DATA....5996
Table 24-735. PCIECTRL_TI_CONF_DIAG_CTRL....5996
Table 24-736. DCAN2 I/O Description....6000
Table 24-737. DCAN1 I/O Description....6000
Table 24-738. DCAN Integration Attributes....6003
Table 24-739. DCAN Clocks and Resets....6003

Table 24-740. DCAN Hardware Requests....6004
Table 24-741. Initialization of a Transmit Object....6013
Table 24-742. Initialization of a single Receive Object for Data Frames....6014
Table 24-743. Initialization of a Single Receive Object for Remote Frames....6014
Table 24-744. Parameters of the CAN Bit Time....6021
Table 24-745. Example For Bit Timing....6028
Table 24-746. Structure of a Message Object....6030
Table 24-747. Message Object Field Descriptions....6030
Table 24-748. Message RAM Addressing in Debug/Suspend and RDA Modes....6032
Table 24-749. Message RAM Representation in Debug/Suspend Mode....6033
Table 24-750. Message RAM Representation in RAM Direct Access Mode....6034
Table 24-751. DCAN Instance Summary....6042
Table 24-752. DCAN Registers Mapping Summary....6042
Table 24-753. DCAN_CTL....6043
Table 24-754. Register Call Summary for Register DCAN_CTL....6046
Table 24-755. DCAN_ES....6046
Table 24-756. Register Call Summary for Register DCAN_ES....6048
Table 24-757. DCAN_ERRC....6048
Table 24-758. Register Call Summary for Register DCAN_ERRC....6049
Table 24-759. DCAN_BTR....6049
Table 24-760. Register Call Summary for Register DCAN_BTR....6050
Table 24-761. DCAN_INT....6050
Table 24-762. Register Call Summary for Register DCAN_INT....6051
Table 24-763. DCAN_TEST....6051
Table 24-764. Register Call Summary for Register DCAN_TEST....6052
Table 24-765. DCAN_PERR....6052
Table 24-766. Register Call Summary for Register DCAN_PERR....6053
Table 24-767. DCAN_REL....6053
Table 24-768. Register Call Summary for Register DCAN_REL....6053
Table 24-769. DCAN_ABOTR....6053
Table 24-770. Register Call Summary for Register DCAN_ABOTR....6053
Table 24-771. DCAN_TXRQ_X....6053
Table 24-772. Register Call Summary for Register DCAN_TXRQ_X....6054
Table 24-773. DCAN_TXRQ12....6054
Table 24-774. Register Call Summary for Register DCAN_TXRQ12....6055
Table 24-775. DCAN_TXRQ34....6055
Table 24-776. Register Call Summary for Register DCAN_TXRQ34....6055
Table 24-777. DCAN_TXRQ56....6055
Table 24-778. Register Call Summary for Register DCAN_TXRQ56....6056
Table 24-779. DCAN_TXRQ78....6056
Table 24-780. Register Call Summary for Register DCAN_TXRQ78....6056
Table 24-781. DCAN_NWDAT_X....6056
Table 24-782. Register Call Summary for Register DCAN_NWDAT_X....6057
Table 24-783. DCAN_NWDAT12....6057
Table 24-784. Register Call Summary for Register DCAN_NWDAT12....6057
Table 24-785. DCAN_NWDAT34....6058
Table 24-786. Register Call Summary for Register DCAN_NWDAT34....6058
Table 24-787. DCAN_NWDAT56....6058
Table 24-788. Register Call Summary for Register DCAN_NWDAT56....6058
Table 24-789. DCAN_NWDAT78....6058
Table 24-790. Register Call Summary for Register DCAN_NWDAT78....6059
Table 24-791. DCAN_INTPND_X....6059
Table 24-792. Register Call Summary for Register DCAN_INTPND_X....6060
Table 24-793. DCAN_INTPND12....6060
Table 24-794. Register Call Summary for Register DCAN_INTPND12....6060
Table 24-795. DCAN_INTPND34....6060
Table 24-796. Register Call Summary for Register DCAN_INTPND34....6061
Table 24-797. DCAN_INTPND56....6061
Table 24-798. Register Call Summary for Register DCAN_INTPND56....6061
Table 24-799. DCAN_INTPND78....6061
Table 24-800. Register Call Summary for Register DCAN_INTPND78....6062

Table 24-801. DCAN_MSGVAL_X....6062
Table 24-802. Register Call Summary for Register DCAN_MSGVAL_X....6062
Table 24-803. DCAN_MSGVAL12....6062
Table 24-804. Register Call Summary for Register DCAN_MSGVAL12....6063
Table 24-805. DCAN_MSGVAL34....6063
Table 24-806. Register Call Summary for Register DCAN_MSGVAL34....6063
Table 24-807. DCAN_MSGVAL56....6063
Table 24-808. Register Call Summary for Register DCAN_MSGVAL56....6064
Table 24-809. DCAN_MSGVAL78....6064
Table 24-810. Register Call Summary for Register DCAN_MSGVAL78....6064
Table 24-811. DCAN_INTMUX12....6064
Table 24-812. Register Call Summary for Register DCAN_INTMUX12....6065
Table 24-813. DCAN_INTMUX34....6065
Table 24-814. Register Call Summary for Register DCAN_INTMUX34....6065
Table 24-815. DCAN_INTMUX56....6065
Table 24-816. Register Call Summary for Register DCAN_INTMUX56....6066
Table 24-817. DCAN_INTMUX78....6066
Table 24-818. Register Call Summary for Register DCAN_INTMUX78....6066
Table 24-819. DCAN_IF1CMD....6066
Table 24-820. Register Call Summary for Register DCAN_IF1CMD....6069
Table 24-821. DCAN_IF1MSK....6069
Table 24-822. Register Call Summary for Register DCAN_IF1MSK....6070
Table 24-823. DCAN_IF1ARB....6070
Table 24-824. Register Call Summary for Register DCAN_IF1ARB....6071
Table 24-825. DCAN_IF1MCTL....6071
Table 24-826. Register Call Summary for Register DCAN_IF1MCTL....6073
Table 24-827. DCAN_IF1DATA....6073
Table 24-828. Register Call Summary for Register DCAN_IF1DATA....6073
Table 24-829. DCAN_IF1DATB....6073
Table 24-830. Register Call Summary for Register DCAN_IF1DATB....6074
Table 24-831. DCAN_IF2CMD....6074
Table 24-832. Register Call Summary for Register DCAN_IF2CMD....6077
Table 24-833. DCAN_IF2MSK....6077
Table 24-834. Register Call Summary for Register DCAN_IF2MSK....6078
Table 24-835. DCAN_IF2ARB....6078
Table 24-836. Register Call Summary for Register DCAN_IF2ARB....6079
Table 24-837. DCAN_IF2MCTL....6079
Table 24-838. Register Call Summary for Register DCAN_IF2MCTL....6080
Table 24-839. DCAN_IF2DATA....6080
Table 24-840. Register Call Summary for Register DCAN_IF2DATA....6081
Table 24-841. DCAN_IF2DATB....6081
Table 24-842. Register Call Summary for Register DCAN_IF2DATB....6081
Table 24-843. DCAN_IF3OBS....6081
Table 24-844. Register Call Summary for Register DCAN_IF3OBS....6083
Table 24-845. DCAN_IF3MSK....6083
Table 24-846. Register Call Summary for Register DCAN_IF3MSK....6084
Table 24-847. DCAN_IF3ARB....6084
Table 24-848. Register Call Summary for Register DCAN_IF3ARB....6085
Table 24-849. DCAN_IF3MCTL....6085
Table 24-850. Register Call Summary for Register DCAN_IF3MCTL....6086
Table 24-851. DCAN_IF3DATA....6086
Table 24-852. Register Call Summary for Register DCAN_IF3DATA....6087
Table 24-853. DCAN_IF3DATB....6087
Table 24-854. Register Call Summary for Register DCAN_IF3DATB....6087
Table 24-855. DCAN_IF3UPD12....6087
Table 24-856. Register Call Summary for Register DCAN_IF3UPD12....6088
Table 24-857. DCAN_IF3UPD34....6088
Table 24-858. Register Call Summary for Register DCAN_IF3UPD34....6088
Table 24-859. DCAN_IF3UPD56....6088
Table 24-860. Register Call Summary for Register DCAN_IF3UPD56....6089
Table 24-861. DCAN_IF3UPD78....6089

Table 24-862. Register Call Summary for Register DCAN_IF3UPD78 ... 6089
Table 24-863. DCAN_TIOC ... 6089
Table 24-864. Register Call Summary for Register DCAN_TIOC ... 6091
Table 24-865. DCAN_RIOC ... 6091
Table 24-866. Register Call Summary for Register DCAN_RIOC ... 6092
Table 24-867. MII I/O Description ... 6096
Table 24-868. RMII I/O Description ... 6097
Table 24-869. RGMII I/O Description ... 6098
Table 24-870. GMAC_SW Integration Attributes ... 6101
Table 24-871. GMAC_SW Clocks and Resets ... 6101
Table 24-872. GMAC_SW Hardware Requests ... 6101
Table 24-873. Learned Address Control Bits ... 6113
Table 24-874. Free (Unused) Address Table Entry Bit Values ... 6113
Table 24-875. Multicast Address Table Entry Bit Values ... 6113
Table 24-876. VLAN/Multicast Address Table Entry Bit Values ... 6114
Table 24-877. Unicast Address Table Entry Bit Values ... 6115
Table 24-878. OUI Unicast Address Table Entry Bit Values ... 6116
Table 24-879. Unicast Address Table Entry Bit Values ... 6116
Table 24-880. VLAN Table Entry ... 6117
Table 24-881. Example of TX Configuration ... 6126
Table 24-882. Example of RX Configuration ... 6126
Table 24-883. Example of Rate-limit Configurations ... 6126
Table 24-884. In-Band Data ... 6130
Table 24-885. Embedded Memories ... 6131
Table 24-886. Switch Latency ... 6134
Table 24-887. Emulation Control Input ... 6134
Table 24-888. Rx Statistics Summary ... 6137
Table 24-889. Tx Statistics Summary ... 6138
Table 24-890. Format for TCP/UDP Packets ... 6145
Table 24-891. Format for ICMP Packets ... 6145
Table 24-892. Rule Engine Format ... 6145
Table 24-893. Instruction Format ... 6146
Table 24-894. Packet Data Operand ... 6147
Table 24-895. Constant Operand ... 6147
Table 24-896. 32-bit Register Operand ... 6148
Table 24-897. 1-bit Register Operand ... 6148
Table 24-898. Base Register Operand ... 6148
Table 24-899. End of Packet Operand ... 6148
Table 24-900. Octet Count Operand ... 6149
Table 24-901. Immediate Data Operands ... 6149
Table 24-902. Immediate Operand Masks ... 6149
Table 24-903. Operand Field With Four Operands ... 6150
Table 24-904. Operand Field With Three Operands ... 6150
Table 24-905. Operand Field With Two Operands ... 6150
Table 24-906. Operand Field With One Operand ... 6151
Table 24-907. Arithmetic/Logical Instruction Field ... 6151
Table 24-908. Arithmetic/Logical Operation Codes ... 6151
Table 24-909. Operation Fields ... 6152
Table 24-910. Limit Operation ... 6152
Table 24-911. Save Operation ... 6153
Table 24-912. Jump Operation ... 6153
Table 24-913. No Operation ... 6154
Table 24-914. Values of Message Type Field ... 6160
Table 24-915. MDIO Read Frame Format ... 6166
Table 24-916. MDIO Write Frame Format ... 6166
Table 24-917. GMAC_SW Instance Summary ... 6173
Table 24-918. SS Registers Mapping Summary ... 6173
Table 24-919. CPSW_ID_VER ... 6173
Table 24-920. CPSW_CONTROL ... 6174
Table 24-921. CPSW_SOFT_RESET ... 6174
Table 24-922. CPSW_STAT_PORT_EN ... 6175

Table 24-923. CPSW_PTYPE....6175
Table 24-924. CPSW_SOFT_IDLE....6176
Table 24-925. CPSW_THRU_RATE....6177
Table 24-926. CPSW_GAP_THRESH....6177
Table 24-927. CPSW_TX_START_WDS....6177
Table 24-928. CPSW_FLOW_CONTROL....6178
Table 24-929. CPSW_VLAN_LTYPE....6178
Table 24-930. CPSW_TS_LTYPE....6178
Table 24-931. CPSW_DLR_LTYPE....6179
Table 24-932. CPSW_EEE_PRESCALE....6179
Table 24-933. PORT Registers Mapping Summary....6179
Table 24-934. P0_CONTROL....6181
Table 24-935. Register Call Summary for Register P0_CONTROL....6182
Table 24-936. P0_MAX_BLKS....6182
Table 24-937. Register Call Summary for Register P0_MAX_BLKS....6182
Table 24-938. P0_BLK_CNT....6182
Table 24-939. Register Call Summary for Register P0_BLK_CNT....6183
Table 24-940. P0_TX_IN_CTL....6183
Table 24-941. Register Call Summary for Register P0_TX_IN_CTL....6183
Table 24-942. P0_PORT_VLAN....6183
Table 24-943. Register Call Summary for Register P0_PORT_VLAN....6184
Table 24-944. P0_TX_PRI_MAP....6184
Table 24-945. Register Call Summary for Register P0_TX_PRI_MAP....6184
Table 24-946. P0_CPDMA_TX_PRI_MAP....6184
Table 24-947. Register Call Summary for Register P0_CPDMA_TX_PRI_MAP....6185
Table 24-948. P0_CPDMA_RX_CH_MAP....6185
Table 24-949. Register Call Summary for Register P0_CPDMA_RX_CH_MAP....6186
Table 24-950. P0_RX_DSCP_PRI_MAP0....6186
Table 24-951. Register Call Summary for Register P0_RX_DSCP_PRI_MAP0....6187
Table 24-952. P0_RX_DSCP_PRI_MAP1....6187
Table 24-953. Register Call Summary for Register P0_RX_DSCP_PRI_MAP1....6188
Table 24-954. P0_RX_DSCP_PRI_MAP2....6188
Table 24-955. Register Call Summary for Register P0_RX_DSCP_PRI_MAP2....6188
Table 24-956. P0_RX_DSCP_PRI_MAP3....6188
Table 24-957. Register Call Summary for Register P0_RX_DSCP_PRI_MAP3....6189
Table 24-958. P0_RX_DSCP_PRI_MAP4....6189
Table 24-959. Register Call Summary for Register P0_RX_DSCP_PRI_MAP4....6190
Table 24-960. P0_RX_DSCP_PRI_MAP5....6190
Table 24-961. Register Call Summary for Register P0_RX_DSCP_PRI_MAP5....6191
Table 24-962. P0_RX_DSCP_PRI_MAP6....6191
Table 24-963. Register Call Summary for Register P0_RX_DSCP_PRI_MAP6....6192
Table 24-964. P0_RX_DSCP_PRI_MAP7....6192
Table 24-965. Register Call Summary for Register P0_RX_DSCP_PRI_MAP7....6192
Table 24-966. P0_IDLE2LPI....6192
Table 24-967. Register Call Summary for Register P0_IDLE2LPI....6193
Table 24-968. P0_LPI2WAKE....6193
Table 24-969. Register Call Summary for Register P0_LPI2WAKE....6193
Table 24-970. P1_CONTROL....6193
Table 24-971. Register Call Summary for Register P1_CONTROL....6195
Table 24-972. P1_MAX_BLKS....6195
Table 24-973. Register Call Summary for Register P1_MAX_BLKS....6196
Table 24-974. P1_BLK_CNT....6196
Table 24-975. Register Call Summary for Register P1_BLK_CNT....6196
Table 24-976. P1_TX_IN_CTL....6196
Table 24-977. Register Call Summary for Register P1_TX_IN_CTL....6197
Table 24-978. P1_PORT_VLAN....6197
Table 24-979. Register Call Summary for Register P1_PORT_VLAN....6197
Table 24-980. P1_TX_PRI_MAP....6197
Table 24-981. Register Call Summary for Register P1_TX_PRI_MAP....6198
Table 24-982. P1_TS_SEQ_MTYPE....6198
Table 24-983. Register Call Summary for Register P1_TS_SEQ_MTYPE....6199

Table 24-984. P1_SA_LO.... 6199
Table 24-985. Register Call Summary for Register P1_SA_LO.... 6199
Table 24-986. P1_SA_HI.... 6199
Table 24-987. Register Call Summary for Register P1_SA_HI....6199
Table 24-988. P1_SEND_PERCENT.... 6199
Table 24-989. Register Call Summary for Register P1_SEND_PERCENT....6200
Table 24-990. P1_RX_DSCP_PRI_MAP0....6200
Table 24-991. Register Call Summary for Register P1_RX_DSCP_PRI_MAP0.... 6201
Table 24-992. P1_RX_DSCP_PRI_MAP1....6201
Table 24-993. Register Call Summary for Register P1_RX_DSCP_PRI_MAP1.... 6202
Table 24-994. P1_RX_DSCP_PRI_MAP2....6202
Table 24-995. Register Call Summary for Register P1_RX_DSCP_PRI_MAP2.... 6203
Table 24-996. P1_RX_DSCP_PRI_MAP3....6203
Table 24-997. Register Call Summary for Register P1_RX_DSCP_PRI_MAP3.... 6203
Table 24-998. P1_RX_DSCP_PRI_MAP4....6204
Table 24-999. Register Call Summary for Register P1_RX_DSCP_PRI_MAP4.... 6204
Table 24-1000. P1_RX_DSCP_PRI_MAP5....6204
Table 24-1001. Register Call Summary for Register P1_RX_DSCP_PRI_MAP5.... 6205
Table 24-1002. P1_RX_DSCP_PRI_MAP6....6205
Table 24-1003. Register Call Summary for Register P1_RX_DSCP_PRI_MAP6.... 6206
Table 24-1004. P1_RX_DSCP_PRI_MAP7....6206
Table 24-1005. Register Call Summary for Register P1_RX_DSCP_PRI_MAP7.... 6207
Table 24-1006. P1_IDLE2LPI.... 6207
Table 24-1007. Register Call Summary for Register P1_IDLE2LPI....6207
Table 24-1008. P1_LPI2WAKE....6207
Table 24-1009. Register Call Summary for Register P1_LPI2WAKE.... 6208
Table 24-1010. P2_CONTROL....6208
Table 24-1011. Register Call Summary for Register P2_CONTROL....6209
Table 24-1012. P2_MAX_BLKS....6209
Table 24-1013. Register Call Summary for Register P2_MAX_BLKS....6210
Table 24-1014. P2_BLK_CNT.... 6210
Table 24-1015. Register Call Summary for Register P2_BLK_CNT....6210
Table 24-1016. P2_TX_IN_CTL....6211
Table 24-1017. Register Call Summary for Register P2_TX_IN_CTL.... 6211
Table 24-1018. P2_PORT_VLAN....6211
Table 24-1019. Register Call Summary for Register P2_PORT_VLAN....6212
Table 24-1020. P2_TX_PRI_MAP.... 6212
Table 24-1021. Register Call Summary for Register P2_TX_PRI_MAP....6212
Table 24-1022. P2_TS_SEQ_MTYPE.... 6212
Table 24-1023. Register Call Summary for Register P2_TS_SEQ_MTYPE.... 6213
Table 24-1024. P2_SA_LO....6213
Table 24-1025. Register Call Summary for Register P2_SA_LO.... 6213
Table 24-1026. P2_SA_HI.... 6213
Table 24-1027. Register Call Summary for Register P2_SA_HI....6214
Table 24-1028. P2_SEND_PERCENT.... 6214
Table 24-1029. Register Call Summary for Register P2_SEND_PERCENT....6214
Table 24-1030. P2_RX_DSCP_PRI_MAP0....6214
Table 24-1031. Register Call Summary for Register P2_RX_DSCP_PRI_MAP0.... 6215
Table 24-1032. P2_RX_DSCP_PRI_MAP1....6215
Table 24-1033. Register Call Summary for Register P2_RX_DSCP_PRI_MAP1.... 6216
Table 24-1034. P2_RX_DSCP_PRI_MAP2....6216
Table 24-1035. Register Call Summary for Register P2_RX_DSCP_PRI_MAP2.... 6217
Table 24-1036. P2_RX_DSCP_PRI_MAP3....6217
Table 24-1037. Register Call Summary for Register P2_RX_DSCP_PRI_MAP3.... 6218
Table 24-1038. P2_RX_DSCP_PRI_MAP4....6218
Table 24-1039. Register Call Summary for Register P2_RX_DSCP_PRI_MAP4.... 6218
Table 24-1040. P2_RX_DSCP_PRI_MAP5....6218
Table 24-1041. Register Call Summary for Register P2_RX_DSCP_PRI_MAP5.... 6219
Table 24-1042. P2_RX_DSCP_PRI_MAP6....6219
Table 24-1043. Register Call Summary for Register P2_RX_DSCP_PRI_MAP6.... 6220
Table 24-1044. P2_RX_DSCP_PRI_MAP7....6220

Table 24-1045. Register Call Summary for Register P2_RX_DSCP_PRI_MAP7 6221
Table 24-1046. P2_IDLE2LPI 6221
Table 24-1047. Register Call Summary for Register P2_IDLE2LPI 6221
Table 24-1048. P2_LPI2WAKE 6221
Table 24-1049. Register Call Summary for Register P2_LPI2WAKE 6222
Table 24-1050. CPDMA Registers Mapping Summary 6222
Table 24-1051. CPDMA_TX_IDVER 6223
Table 24-1052. Register Call Summary for Register CPDMA_TX_IDVER 6223
Table 24-1053. CPDMA_TX_CONTROL 6223
Table 24-1054. Register Call Summary for Register CPDMA_TX_CONTROL 6224
Table 24-1055. CPDMA_TX_TEARDOWN 6224
Table 24-1056. Register Call Summary for Register CPDMA_TX_TEARDOWN 6224
Table 24-1057. CPDMA_RX_IDVER 6224
Table 24-1058. Register Call Summary for Register CPDMA_RX_IDVER 6225
Table 24-1059. CPDMA_RX_CONTROL 6225
Table 24-1060. Register Call Summary for Register CPDMA_RX_CONTROL 6225
Table 24-1061. CPDMA_RX_TEARDOWN 6225
Table 24-1062. Register Call Summary for Register CPDMA_RX_TEARDOWN 6225
Table 24-1063. CPDMA_SOFT_RESET 6226
Table 24-1064. Register Call Summary for Register CPDMA_SOFT_RESET 6226
Table 24-1065. CPDMA_DMACONTROL 6226
Table 24-1066. Register Call Summary for Register CPDMA_DMACONTROL 6227
Table 24-1067. CPDMA_DMASTATUS 6227
Table 24-1068. Register Call Summary for Register CPDMA_DMASTATUS 6228
Table 24-1069. CPDMA_RX_BUFFER_OFFSET 6229
Table 24-1070. Register Call Summary for Register CPDMA_RX_BUFFER_OFFSET 6229
Table 24-1071. CPDMA_EMCONTROL 6229
Table 24-1072. Register Call Summary for Register CPDMA_EMCONTROL 6229
Table 24-1073. CPDMA_TX_PRI0_RATE 6229
Table 24-1074. Register Call Summary for Register CPDMA_TX_PRI0_RATE 6230
Table 24-1075. CPDMA_TX_PRI1_RATE 6230
Table 24-1076. Register Call Summary for Register CPDMA_TX_PRI1_RATE 6230
Table 24-1077. CPDMA_TX_PRI2_RATE 6230
Table 24-1078. Register Call Summary for Register CPDMA_TX_PRI2_RATE 6231
Table 24-1079. CPDMA_TX_PRI3_RATE 6231
Table 24-1080. Register Call Summary for Register CPDMA_TX_PRI3_RATE 6231
Table 24-1081. CPDMA_TX_PRI4_RATE 6231
Table 24-1082. Register Call Summary for Register CPDMA_TX_PRI4_RATE 6231
Table 24-1083. CPDMA_TX_PRI5_RATE 6231
Table 24-1084. Register Call Summary for Register CPDMA_TX_PRI5_RATE 6232
Table 24-1085. CPDMA_TX_PRI6_RATE 6232
Table 24-1086. Register Call Summary for Register CPDMA_TX_PRI6_RATE 6232
Table 24-1087. CPDMA_TX_PRI7_RATE 6232
Table 24-1088. Register Call Summary for Register CPDMA_TX_PRI7_RATE 6233
Table 24-1089. CPDMA_TX_INTSTAT_RAW 6233
Table 24-1090. Register Call Summary for Register CPDMA_TX_INTSTAT_RAW 6233
Table 24-1091. CPDMA_TX_INTSTAT_MASKED 6233
Table 24-1092. Register Call Summary for Register CPDMA_TX_INTSTAT_MASKED 6234
Table 24-1093. CPDMA_TX_INTMASK_SET 6234
Table 24-1094. Register Call Summary for Register CPDMA_TX_INTMASK_SET 6234
Table 24-1095. CPDMA_TX_INTMASK_CLEAR 6234
Table 24-1096. Register Call Summary for Register CPDMA_TX_INTMASK_CLEAR 6235
Table 24-1097. CPDMA_IN_VECTOR 6235
Table 24-1098. Register Call Summary for Register CPDMA_IN_VECTOR 6235
Table 24-1099. CPDMA_EOI_VECTOR 6235
Table 24-1100. Register Call Summary for Register CPDMA_EOI_VECTOR 6236
Table 24-1101. CPDMA_RX_INTSTAT_RAW 6236
Table 24-1102. Register Call Summary for Register CPDMA_RX_INTSTAT_RAW 6237
Table 24-1103. CPDMA_RX_INTSTAT_MASKED 6237
Table 24-1104. Register Call Summary for Register CPDMA_RX_INTSTAT_MASKED 6237
Table 24-1105. CPDMA_RX_INTMASK_SET 6237

Table 24-1106. Register Call Summary for Register CPDMA_RX_INTMASK_SET....6239
Table 24-1107. CPDMA_RX_INTMASK_CLEAR....6239
Table 24-1108. Register Call Summary for Register CPDMA_RX_INTMASK_CLEAR....6240
Table 24-1109. CPDMA_DMA_INTSTAT_RAW....6240
Table 24-1110. Register Call Summary for Register CPDMA_DMA_INTSTAT_RAW....6240
Table 24-1111. CPDMA_DMA_INTSTAT_MASKED....6240
Table 24-1112. Register Call Summary for Register CPDMA_DMA_INTSTAT_MASKED....6241
Table 24-1113. CPDMA_DMA_INTMASK_SET....6241
Table 24-1114. Register Call Summary for Register CPDMA_DMA_INTMASK_SET....6241
Table 24-1115. CPDMA_DMA_INTMASK_CLEAR....6241
Table 24-1116. Register Call Summary for Register CPDMA_DMA_INTMASK_CLEAR....6242
Table 24-1117. CPDMA_RX0_PENDTHRESH....6242
Table 24-1118. Register Call Summary for Register CPDMA_RX0_PENDTHRESH....6242
Table 24-1119. CPDMA_RX1_PENDTHRESH....6242
Table 24-1120. Register Call Summary for Register CPDMA_RX1_PENDTHRESH....6242
Table 24-1121. CPDMA_RX2_PENDTHRESH....6242
Table 24-1122. Register Call Summary for Register CPDMA_RX2_PENDTHRESH....6243
Table 24-1123. CPDMA_RX3_PENDTHRESH....6243
Table 24-1124. Register Call Summary for Register CPDMA_RX3_PENDTHRESH....6243
Table 24-1125. CPDMA_RX4_PENDTHRESH....6243
Table 24-1126. Register Call Summary for Register CPDMA_RX4_PENDTHRESH....6244
Table 24-1127. CPDMA_RX5_PENDTHRESH....6244
Table 24-1128. Register Call Summary for Register CPDMA_RX5_PENDTHRESH....6244
Table 24-1129. CPDMA_RX6_PENDTHRESH....6244
Table 24-1130. Register Call Summary for Register CPDMA_RX6_PENDTHRESH....6244
Table 24-1131. CPDMA_RX7_PENDTHRESH....6244
Table 24-1132. Register Call Summary for Register CPDMA_RX7_PENDTHRESH....6245
Table 24-1133. CPDMA_RX0_FREEBUFFER....6245
Table 24-1134. Register Call Summary for Register CPDMA_RX0_FREEBUFFER....6245
Table 24-1135. CPDMA_RX1_FREEBUFFER....6245
Table 24-1136. Register Call Summary for Register CPDMA_RX1_FREEBUFFER....6246
Table 24-1137. CPDMA_RX2_FREEBUFFER....6246
Table 24-1138. Register Call Summary for Register CPDMA_RX2_FREEBUFFER....6246
Table 24-1139. CPDMA_RX3_FREEBUFFER....6246
Table 24-1140. Register Call Summary for Register CPDMA_RX3_FREEBUFFER....6247
Table 24-1141. CPDMA_RX4_FREEBUFFER....6247
Table 24-1142. Register Call Summary for Register CPDMA_RX4_FREEBUFFER....6247
Table 24-1143. CPDMA_RX5_FREEBUFFER....6247
Table 24-1144. Register Call Summary for Register CPDMA_RX5_FREEBUFFER....6248
Table 24-1145. CPDMA_RX6_FREEBUFFER....6248
Table 24-1146. Register Call Summary for Register CPDMA_RX6_FREEBUFFER....6248
Table 24-1147. CPDMA_RX7_FREEBUFFER....6249
Table 24-1148. Register Call Summary for Register CPDMA_RX7_FREEBUFFER....6249
Table 24-1149. STATS Registers Mapping Summary....6249
Table 24-1150. GOOD_RX_FRAMES....6250
Table 24-1151. Register Call Summary for Register GOOD_RX_FRAMES....6250
Table 24-1152. BROADCAST_RX_FRAMES....6250
Table 24-1153. Register Call Summary for Register BROADCAST_RX_FRAMES....6251
Table 24-1154. MULTICAST_RX_FRAMES....6251
Table 24-1155. Register Call Summary for Register MULTICAST_RX_FRAMES....6251
Table 24-1156. PAUSE_RX_FRAMES....6251
Table 24-1157. Register Call Summary for Register PAUSE_RX_FRAMES....6252
Table 24-1158. RX_CRC_ERRORS....6252
Table 24-1159. Register Call Summary for Register RX_CRC_ERRORS....6252
Table 24-1160. RX_ALIGN_CODE_ERRORS....6252
Table 24-1161. Register Call Summary for Register RX_ALIGN_CODE_ERRORS....6253
Table 24-1162. OVERSIZE_RX_FRAMES....6253
Table 24-1163. Register Call Summary for Register OVERSIZE_RX_FRAMES....6253
Table 24-1164. RX_JABBERS....6253
Table 24-1165. Register Call Summary for Register RX_JABBERS....6253
Table 24-1166. UNDERSIZE_RX_FRAMES....6254

Table 24-1167. Register Call Summary for Register UNDERSIZE_RX_FRAMES....6254
Table 24-1168. RX_FRAGMENTS....6254
Table 24-1169. Register Call Summary for Register RX_FRAGMENTS....6254
Table 24-1170. RX_OCTETS....6254
Table 24-1171. Register Call Summary for Register RX_OCTETS....6255
Table 24-1172. GOOD_TX_FRAMES....6255
Table 24-1173. Register Call Summary for Register GOOD_TX_FRAMES....6255
Table 24-1174. BROADCAST_TX_FRAMES....6255
Table 24-1175. Register Call Summary for Register BROADCAST_TX_FRAMES....6255
Table 24-1176. MULTICAST_TX_FRAMES....6256
Table 24-1177. Register Call Summary for Register MULTICAST_TX_FRAMES....6256
Table 24-1178. PAUSE_TX_FRAMES....6256
Table 24-1179. Register Call Summary for Register PAUSE_TX_FRAMES....6256
Table 24-1180. DEFERRED_TX_FRAMES....6256
Table 24-1181. Register Call Summary for Register DEFERRED_TX_FRAMES....6257
Table 24-1182. COLLISIONS....6257
Table 24-1183. Register Call Summary for Register COLLISIONS....6257
Table 24-1184. SINGLE_COLLISION_TX_FRAMES....6257
Table 24-1185. Register Call Summary for Register SINGLE_COLLISION_TX_FRAMES....6257
Table 24-1186. MULTIPLE_COLLISION_TX_FRAMES....6258
Table 24-1187. Register Call Summary for Register MULTIPLE_COLLISION_TX_FRAMES....6258
Table 24-1188. EXCESSIVE_COLLISIONS....6258
Table 24-1189. Register Call Summary for Register EXCESSIVE_COLLISIONS....6258
Table 24-1190. LATE_COLLISIONS....6258
Table 24-1191. Register Call Summary for Register LATE_COLLISIONS....6259
Table 24-1192. TX_UNDERRUN....6259
Table 24-1193. Register Call Summary for Register TX_UNDERRUN....6259
Table 24-1194. CARRIER_SENSE_ERRORS....6259
Table 24-1195. Register Call Summary for Register CARRIER_SENSE_ERRORS....6259
Table 24-1196. TX_OCTETS....6259
Table 24-1197. Register Call Summary for Register TX_OCTETS....6260
Table 24-1198. RX_TX_64_OCTET_FRAMES....6260
Table 24-1199. Register Call Summary for Register RX_TX_64_OCTET_FRAMES....6260
Table 24-1200. RX_TX_65_127_OCTET_FRAMES....6260
Table 24-1201. Register Call Summary for Register RX_TX_65_127_OCTET_FRAMES....6261
Table 24-1202. RX_TX_128_255_OCTET_FRAMES....6261
Table 24-1203. Register Call Summary for Register RX_TX_128_255_OCTET_FRAMES....6261
Table 24-1204. RX_TX_256_511_OCTET_FRAMES....6261
Table 24-1205. Register Call Summary for Register RX_TX_256_511_OCTET_FRAMES....6261
Table 24-1206. RX_TX_512_1023_OCTET_FRAMES....6261
Table 24-1207. Register Call Summary for Register RX_TX_512_1023_OCTET_FRAMES....6262
Table 24-1208. RX_TX_1024_UP_OCTET_FRAMES....6262
Table 24-1209. Register Call Summary for Register RX_TX_1024_UP_OCTET_FRAMES....6262
Table 24-1210. NET_OCTETS....6262
Table 24-1211. Register Call Summary for Register NET_OCTETS....6263
Table 24-1212. RX_START_OF_FRAME_OVERRUNS....6263
Table 24-1213. Register Call Summary for Register RX_START_OF_FRAME_OVERRUNS....6263
Table 24-1214. RX_MIDDLE_OF_FRAME_OVERRUNS....6263
Table 24-1215. Register Call Summary for Register RX_MIDDLE_OF_FRAME_OVERRUNS....6264
Table 24-1216. RX_DMA_OVERRUNS....6264
Table 24-1217. Register Call Summary for Register RX_DMA_OVERRUNS....6264
Table 24-1218. STATERAM Registers Mapping Summary....6264
Table 24-1219. TX0_HDP....6265
Table 24-1220. Register Call Summary for Register TX0_HDP....6265
Table 24-1221. TX1_HDP....6265
Table 24-1222. Register Call Summary for Register TX1_HDP....6266
Table 24-1223. TX2_HDP....6266
Table 24-1224. Register Call Summary for Register TX2_HDP....6266
Table 24-1225. TX3_HDP....6266
Table 24-1226. Register Call Summary for Register TX3_HDP....6267
Table 24-1227. TX4_HDP....6267

Table 24-1228. Register Call Summary for Register TX4_HDP....... 6267
Table 24-1229. TX5_HDP....... 6267
Table 24-1230. Register Call Summary for Register TX5_HDP....... 6267
Table 24-1231. TX6_HDP....... 6267
Table 24-1232. Register Call Summary for Register TX6_HDP....... 6268
Table 24-1233. TX7_HDP....... 6268
Table 24-1234. Register Call Summary for Register TX7_HDP....... 6268
Table 24-1235. RX0_HDP....... 6268
Table 24-1236. Register Call Summary for Register RX0_HDP....... 6269
Table 24-1237. RX1_HDP....... 6269
Table 24-1238. Register Call Summary for Register RX1_HDP....... 6269
Table 24-1239. RX2_HDP....... 6269
Table 24-1240. Register Call Summary for Register RX2_HDP....... 6269
Table 24-1241. RX3_HDP....... 6269
Table 24-1242. Register Call Summary for Register RX3_HDP....... 6270
Table 24-1243. RX4_HDP....... 6270
Table 24-1244. Register Call Summary for Register RX4_HDP....... 6270
Table 24-1245. RX5_HDP....... 6270
Table 24-1246. Register Call Summary for Register RX5_HDP....... 6271
Table 24-1247. RX6_HDP....... 6271
Table 24-1248. Register Call Summary for Register RX6_HDP....... 6271
Table 24-1249. RX7_HDP....... 6271
Table 24-1250. Register Call Summary for Register RX7_HDP....... 6271
Table 24-1251. TX0_CP....... 6271
Table 24-1252. Register Call Summary for Register TX0_CP....... 6272
Table 24-1253. TX1_CP....... 6272
Table 24-1254. Register Call Summary for Register TX1_CP....... 6272
Table 24-1255. TX2_CP....... 6272
Table 24-1256. Register Call Summary for Register TX2_CP....... 6272
Table 24-1257. TX3_CP....... 6272
Table 24-1258. Register Call Summary for Register TX3_CP....... 6273
Table 24-1259. TX4_CP....... 6273
Table 24-1260. Register Call Summary for Register TX4_CP....... 6273
Table 24-1261. TX5_CP....... 6273
Table 24-1262. Register Call Summary for Register TX5_CP....... 6274
Table 24-1263. TX6_CP....... 6274
Table 24-1264. Register Call Summary for Register TX6_CP....... 6274
Table 24-1265. TX7_CP....... 6274
Table 24-1266. Register Call Summary for Register TX7_CP....... 6274
Table 24-1267. RX0_CP....... 6274
Table 24-1268. Register Call Summary for Register RX0_CP....... 6275
Table 24-1269. RX1_CP....... 6275
Table 24-1270. Register Call Summary for Register RX1_CP....... 6275
Table 24-1271. RX2_CP....... 6275
Table 24-1272. Register Call Summary for Register RX2_CP....... 6276
Table 24-1273. RX3_CP....... 6276
Table 24-1274. Register Call Summary for Register RX3_CP....... 6276
Table 24-1275. RX4_CP....... 6276
Table 24-1276. Register Call Summary for Register RX4_CP....... 6277
Table 24-1277. RX5_CP....... 6277
Table 24-1278. Register Call Summary for Register RX5_CP....... 6277
Table 24-1279. RX6_CP....... 6277
Table 24-1280. Register Call Summary for Register RX6_CP....... 6278
Table 24-1281. RX7_CP....... 6278
Table 24-1282. Register Call Summary for Register RX7_CP....... 6278
Table 24-1283. CPTS Registers Mapping Summary....... 6278
Table 24-1284. CPTS_IDVER....... 6279
Table 24-1285. Register Call Summary for Register CPTS_IDVER....... 6279
Table 24-1286. CPTS_CONTROL....... 6279
Table 24-1287. Register Call Summary for Register CPTS_CONTROL....... 6280
Table 24-1288. CPTS_TS_PUSH....... 6280

Table 24-1289. Register Call Summary for Register CPTS_TS_PUSH....... 6280
Table 24-1290. CPTS_TS_LOAD_VAL....... 6280
Table 24-1291. Register Call Summary for Register CPTS_TS_LOAD_VAL....... 6281
Table 24-1292. CPTS_TS_LOAD_EN....... 6281
Table 24-1293. Register Call Summary for Register CPTS_TS_LOAD_EN....... 6281
Table 24-1294. CPTS_INTSTAT_RAW....... 6281
Table 24-1295. Register Call Summary for Register CPTS_INTSTAT_RAW....... 6282
Table 24-1296. CPTS_INTSTAT_MASKED....... 6282
Table 24-1297. Register Call Summary for Register CPTS_INTSTAT_MASKED....... 6282
Table 24-1298. CPTS_INT_ENABLE....... 6282
Table 24-1299. Register Call Summary for Register CPTS_INT_ENABLE....... 6283
Table 24-1300. CPTS_EVENT_POP....... 6283
Table 24-1301. Register Call Summary for Register CPTS_EVENT_POP....... 6283
Table 24-1302. CPTS_EVENT_LOW....... 6283
Table 24-1303. Register Call Summary for Register CPTS_EVENT_LOW....... 6284
Table 24-1304. CPTS_EVENT_HIGH....... 6284
Table 24-1305. Register Call Summary for Register CPTS_EVENT_HIGH....... 6284
Table 24-1306. ALE Registers Mapping Summary....... 6284
Table 24-1307. ALE_IDVER....... 6285
Table 24-1308. ALE_CONTROL....... 6285
Table 24-1309. ALE_PRESCALE....... 6286
Table 24-1310. ALE_UNKNOWN_VLAN....... 6287
Table 24-1311. ALE_TBLCTL....... 6287
Table 24-1312. ALE_TBLW2....... 6288
Table 24-1313. ALE_TBLW1....... 6288
Table 24-1314. ALE_TBLW0....... 6288
Table 24-1315. ALE_PORTCTL0....... 6288
Table 24-1316. ALE_PORTCTL1....... 6289
Table 24-1317. ALE_PORTCTL2....... 6290
Table 24-1318. ALE_PORTCTL3....... 6291
Table 24-1319. ALE_PORTCTL4....... 6292
Table 24-1320. ALE_PORTCTL5....... 6293
Table 24-1321. SL Registers Mapping Summary....... 6294
Table 24-1322. SL_IDVER....... 6295
Table 24-1323. Register Call Summary for Register SL_IDVER....... 6295
Table 24-1324. SL_MACCONTROL....... 6295
Table 24-1325. Register Call Summary for Register SL_MACCONTROL....... 6298
Table 24-1326. SL_MACSTATUS....... 6298
Table 24-1327. Register Call Summary for Register SL_MACSTATUS....... 6298
Table 24-1328. SL_SOFT_RESET....... 6298
Table 24-1329. Register Call Summary for Register SL_SOFT_RESET....... 6299
Table 24-1330. SL_RX_MAXLEN....... 6299
Table 24-1331. Register Call Summary for Register SL_RX_MAXLEN....... 6299
Table 24-1332. SL_BOFFTEST....... 6299
Table 24-1333. Register Call Summary for Register SL_BOFFTEST....... 6300
Table 24-1334. SL_RX_PAUSE....... 6300
Table 24-1335. Register Call Summary for Register SL_RX_PAUSE....... 6301
Table 24-1336. SL_TX_PAUSE....... 6301
Table 24-1337. Register Call Summary for Register SL_TX_PAUSE....... 6301
Table 24-1338. SL_EMCONTROL....... 6301
Table 24-1339. Register Call Summary for Register SL_EMCONTROL....... 6302
Table 24-1340. SL_RX_PRI_MAP....... 6302
Table 24-1341. Register Call Summary for Register SL_RX_PRI_MAP....... 6303
Table 24-1342. SL_TX_GAP....... 6303
Table 24-1343. Register Call Summary for Register SL_TX_GAP....... 6303
Table 24-1344. MDIO Registers Mapping Summary....... 6303
Table 24-1345. MDIO_VER....... 6303
Table 24-1346. Register Call Summary for Register MDIO_VER....... 6304
Table 24-1347. MDIO_CONTROL....... 6304
Table 24-1348. Register Call Summary for Register MDIO_CONTROL....... 6305
Table 24-1349. MDIO_ALIVE....... 6305

Table 24-1350. Register Call Summary for Register MDIO_ALIVE....6305
Table 24-1351. MDIO_LINK....6305
Table 24-1352. Register Call Summary for Register MDIO_LINK....6306
Table 24-1353. MDIO_LINKINTRAW....6306
Table 24-1354. Register Call Summary for Register MDIO_LINKINTRAW....6306
Table 24-1355. MDIO_LINKINTMASKED....6306
Table 24-1356. Register Call Summary for Register MDIO_LINKINTMASKED....6307
Table 24-1357. MDIO_USERINTRAW....6307
Table 24-1358. Register Call Summary for Register MDIO_USERINTRAW....6307
Table 24-1359. MDIO_USERINTMASKED....6307
Table 24-1360. Register Call Summary for Register MDIO_USERINTMASKED....6308
Table 24-1361. MDIO_USERINTMASKSET....6308
Table 24-1362. Register Call Summary for Register MDIO_USERINTMASKSET....6309
Table 24-1363. MDIO_USERINTMASKCLR....6309
Table 24-1364. Register Call Summary for Register MDIO_USERINTMASKCLR....6309
Table 24-1365. MDIO_USERACCESS0....6309
Table 24-1366. Register Call Summary for Register MDIO_USERACCESS0....6310
Table 24-1367. MDIO_USERPHYSEL0....6310
Table 24-1368. Register Call Summary for Register MDIO_USERPHYSEL0....6311
Table 24-1369. MDIO_USERACCESS1....6311
Table 24-1370. Register Call Summary for Register MDIO_USERACCESS1....6311
Table 24-1371. MDIO_USERPHYSEL1....6311
Table 24-1372. Register Call Summary for Register MDIO_USERPHYSEL1....6312
Table 24-1373. WR Registers Mapping Summary....6312
Table 24-1374. WR_IDVER....6312
Table 24-1375. Register Call Summary for Register WR_IDVER....6313
Table 24-1376. WR_SOFT_RESET....6313
Table 24-1377. Register Call Summary for Register WR_SOFT_RESET....6313
Table 24-1378. WR_CONTROL....6313
Table 24-1379. Register Call Summary for Register WR_CONTROL....6314
Table 24-1380. WR_INT_CONTROL....6314
Table 24-1381. Register Call Summary for Register WR_INT_CONTROL....6315
Table 24-1382. WR_C0_RX_THRESH_EN....6315
Table 24-1383. Register Call Summary for Register WR_C0_RX_THRESH_EN....6315
Table 24-1384. WR_C0_RX_EN....6315
Table 24-1385. Register Call Summary for Register WR_C0_RX_EN....6315
Table 24-1386. WR_C0_TX_EN....6315
Table 24-1387. Register Call Summary for Register WR_C0_TX_EN....6316
Table 24-1388. WR_C0_MISC_EN....6316
Table 24-1389. Register Call Summary for Register WR_C0_MISC_EN....6316
Table 24-1390. WR_C0_RX_THRESH_STAT....6316
Table 24-1391. Register Call Summary for Register WR_C0_RX_THRESH_STAT....6317
Table 24-1392. WR_C0_RX_STAT....6317
Table 24-1393. Register Call Summary for Register WR_C0_RX_STAT....6317
Table 24-1394. WR_C0_TX_STAT....6317
Table 24-1395. Register Call Summary for Register WR_C0_TX_STAT....6317
Table 24-1396. WR_C0_MISC_STAT....6317
Table 24-1397. Register Call Summary for Register WR_C0_MISC_STAT....6318
Table 24-1398. WR_C0_RX_IMAX....6318
Table 24-1399. Register Call Summary for Register WR_C0_RX_IMAX....6318
Table 24-1400. WR_C0_TX_IMAX....6318
Table 24-1401. Register Call Summary for Register WR_C0_TX_IMAX....6319
Table 24-1402. WR_RGMII_CTL....6319
Table 24-1403. Register Call Summary for Register WR_RGMII_CTL....6319
Table 24-1404. WR_STATUS....6320
Table 24-1405. Register Call Summary for Register WR_STATUS....6320
Table 24-1406. SPF Registers Mapping Summary....6320
Table 24-1407. SPF_IDVER....6321
Table 24-1408. Register Call Summary for Register SPF_IDVER....6321
Table 24-1409. SPF_STATUS....6321
Table 24-1410. Register Call Summary for Register SPF_STATUS....6322

Table 24-1411. SPF_CONTROL....6322
Table 24-1412. Register Call Summary for Register SPF_CONTROL....6322
Table 24-1413. SPF_DROPCOUNT....6322
Table 24-1414. Register Call Summary for Register SPF_DROPCOUNT....6323
Table 24-1415. SPF_SWRESET....6323
Table 24-1416. Register Call Summary for Register SPF_SWRESET....6323
Table 24-1417. SPF_PRESCALE....6323
Table 24-1418. Register Call Summary for Register SPF_PRESCALE....6324
Table 24-1419. SPF_RATELIMi....6324
Table 24-1420. Register Call Summary for Register SPF_RATELIMi....6324
Table 24-1421. SPF_CONSTj....6324
Table 24-1422. Register Call Summary for Register SPF_CONSTj....6325
Table 24-1423. SPF_INSTRW2....6325
Table 24-1424. Register Call Summary for Register SPF_INSTRW2....6325
Table 24-1425. SPF_INSTRW1....6325
Table 24-1426. Register Call Summary for Register SPF_INSTRW1....6325
Table 24-1427. SPF_INSTRW0....6325
Table 24-1428. Register Call Summary for Register SPF_INSTRW0....6326
Table 24-1429. SPF_INSTR_CTL....6326
Table 24-1430. Register Call Summary for Register SPF_INSTR_CTL....6326
Table 24-1431. SPF_LOG_BEGIN....6326
Table 24-1432. Register Call Summary for Register SPF_LOG_BEGIN....6327
Table 24-1433. SPF_LOG_END....6327
Table 24-1434. Register Call Summary for Register SPF_LOG_END....6327
Table 24-1435. SPF_LOG_HWPTR....6327
Table 24-1436. Register Call Summary for Register SPF_LOG_HWPTR....6328
Table 24-1437. SPF_LOG_SWPTR....6328
Table 24-1438. Register Call Summary for Register SPF_LOG_SWPTR....6328
Table 24-1439. SPF_LOG_MAP0....6328
Table 24-1440. Register Call Summary for Register SPF_LOG_MAP0....6328
Table 24-1441. SPF_LOG_MAP1....6328
Table 24-1442. Register Call Summary for Register SPF_LOG_MAP1....6329
Table 24-1443. SPF_LOG_THRESHk....6329
Table 24-1444. Register Call Summary for Register SPF_LOG_THRESHk....6329
Table 24-1445. SPF_INTCNT....6329
Table 24-1446. Register Call Summary for Register SPF_INTCNT....6330
Table 24-1447. SPF_INT_RAW....6330
Table 24-1448. Register Call Summary for Register SPF_INT_RAW....6330
Table 24-1449. SPF_INT_MASKED....6330
Table 24-1450. Register Call Summary for Register SPF_INT_MASKED....6331
Table 24-1451. SPF_MASK_SET....6331
Table 24-1452. Register Call Summary for Register SPF_MASK_SET....6331
Table 24-1453. SPF_MASK_CLR....6331
Table 24-1454. Register Call Summary for Register SPF_MASK_CLR....6332
Table 24-1455. MLB Sub System I/O Description....6334
Table 24-1456. MLB Integration Attributes....6338
Table 24-1457. MLB Clocks and Resets....6338
Table 24-1458. MLB Hardware Requests....6339
Table 24-1459. MediaLB Channel Address to Logical Channel Mapping....6340
Table 24-1460. Channel Table RAM Address Mapping....6341
Table 24-1461. Channel Allocation Table Entry Map....6342
Table 24-1462. Format Of Channel Allocation Table Entry....6343
Table 24-1463. Field Descriptions Of Channel Allocation Table Entry....6343
Table 24-1464. Format Of Synchronous Channel Descriptor Table Entry....6344
Table 24-1465. Field Descriptions Of Synchronous Channel Descriptor Table Entry....6344
Table 24-1466. Format Of Isochronous Channel Descriptor Table Entry....6345
Table 24-1467. Field Descriptions Of Isochronous Channel Descriptor Table Entry....6345
Table 24-1468. Format Of Asynchronous and Control Channel Descriptor Table Entry....6347
Table 24-1469. Field Descriptions Of Asynchronous And Control Channel Descriptor Table Entry....6347
Table 24-1470. Field Descriptions Of DMA Descriptor Table....6348
Table 24-1471. Format Of Synchronous DMA Descriptor Table Entry....6350

Table 24-1472. Format Of Isochronous DMA Descriptor Table Entry....6351
Table 24-1473. Format Of Single-Packet Asynchronous and Control DMA Descriptor Table Entry....6352
Table 24-1474. Format Of Multiple-Packet Asynchronous And Control DMA Descriptor Table Entry....6353
Table 24-1475. Global Initialization of Surrounding Modules....6356
Table 24-1476. Channel Initialization Steps....6356
Table 24-1477. Configuring The Hardware....6356
Table 24-1478. Programming The Routing Fabric Block....6357
Table 24-1479. Subsequence - Program The Channel Descriptor Table....6357
Table 24-1480. Subsequence - Write To The Data Buffer RAM....6357
Table 24-1481. Subsequence - Read From The Data Buffer RAM....6358
Table 24-1482. Programming The DMA Block....6358
Table 24-1483. Subsequence - Program The DMA Block Ping Page....6359
Table 24-1484. Subsequence - Program The DMA Block Pong Page....6360
Table 24-1485. Synchronizing And Unmuting The Synchronous Channel....6360
Table 24-1486. Channel Servicing Steps....6361
Table 24-1487. Servicing the DMA Interrupts....6361
Table 24-1488. Subsequence - Reading The DMA Descriptor Table Entry....6362
Table 24-1489. Servicing MLB Interrupts....6362
Table 24-1490. Polling For MediaLB System Commands....6363
Table 24-1491. Direct Channel Table RAM Writes....6363
Table 24-1492. Direct Channel Table RAM Reads....6363
Table 24-1493. MLB Instance Summary....6363
Table 24-1494. MLB Registers Mapping Summary....6364
Table 24-1495. MLB_MLBSSREV....6364
Table 24-1496. MLB_MLBSSPWR....6364
Table 24-1497. MLB_MLBSSPRF....6365
Table 24-1498. MLB_MLBC0....6367
Table 24-1499. MLB_MS0....6368
Table 24-1500. MLB_MS1....6368
Table 24-1501. MLB_MSS....6369
Table 24-1502. MLB_MSD....6370
Table 24-1503. MLB_MIEN....6370
Table 24-1504. MLB_MLBC1....6373
Table 24-1505. MLB_MDAT0....6373
Table 24-1506. MLB_MDAT1....6373
Table 24-1507. MLB_MDAT2....6374
Table 24-1508. MLB_MDAT3....6374
Table 24-1509. MLB_MDWE0....6374
Table 24-1510. MLB_MDWE1....6374
Table 24-1511. MLB_MDWE2....6375
Table 24-1512. MLB_MDWE3....6375
Table 24-1513. MLB_MCTL....6375
Table 24-1514. MLB_MADR....6375
Table 24-1515. MLB_DIENR....6376
Table 24-1516. MLB_DICER0....6376
Table 24-1517. MLB_DICER1....6376
Table 24-1518. MLB_DCTL....6377
Table 24-1519. MLB_DCSR0....6377
Table 24-1520. MLB_DCSR1....6378
Table 24-1521. MLB_DCMR0....6378
Table 24-1522. MLB_DCMR1....6378
Table 25-1. Standard 4.5 Supported Features....6383
Table 25-2. MMCi Supported Transfer Rates and Functionalities....6384
Table 25-3. Description of eMMC/SD/SDIOi host controller I/O's (where i = 1 to 4)....6385
Table 25-4. Relationship Between Configuration and Name of Response Type....6390
Table 25-5. MMC Integration Attributes....6394
Table 25-6. MMC Clocks and Resets....6395
Table 25-7. MMC Hardware Requests....6396
Table 25-8. Smart-Idle Mode and Wake-Up Capabilities....6401
Table 25-9. Local Power-Management Features....6402
Table 25-10. Clock Activity Settings....6403

Table 25-11. Events....6404
Table 25-12. Descriptor Line Overview....6409
Table 25-13. Available Actions of a Descriptor Line....6409
Table 25-14. Additional Parameters of a Descriptor Line....6409
Table 25-15. ADMA2 States Description....6411
Table 25-16. ADMA FSM Symbol Definition....6411
Table 25-17. Memory Size, BLEN, and Buffer Relationship....6417
Table 25-18. MMC, SD, SDIO Responses in the MMCHS_RSPxx Registers....6418
Table 25-19. CC and TC Values Upon Error Detected....6419
Table 25-20. eMMC/SD/SDIOi Controller Transfer Stop Command Summary....6424
Table 25-21. eMMC/SD/SDIO Hardware Status Features....6428
Table 25-22. eMMC/SD/SDIO Preset Value Registers....6429
Table 25-23. Global Initialization of Surrounding Modules....6430
Table 25-24. eMMC/SD/SDIO Controller Meta Initialization Steps....6430
Table 25-25. Register Call Summary for Main Sequence – Software Reset Flow....6431
Table 25-26. eMMC/SD/SDIO Controller Wake-Up Configuration....6431
Table 25-27. Register Call Summary for Main Sequence – Bus Configuration....6432
Table 25-28. Register Call Summary for Main Sequence – Card Identification and Selection....6434
Table 25-29. Subprocess Call Summary for Main Sequence – Card Identification and Selection....6435
Table 25-30. CMD Line Reset....6435
Table 25-31. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With interrupt....6436
Table 25-32. Subprocess Call Summary for Main Sequence – eMMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Mode With Interrupt....6436
Table 25-33. DATA Lines Reset....6437
Table 25-34. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling....6437
Table 25-35. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling....6438
Table 25-36. Register Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling....6440
Table 25-37. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling....6440
Table 25-38. Register Call Summary for Main Sequence – Read/Write in CE-ATA Mode....6441
Table 25-39. Subprocess Call Summary for Main Sequence – Read/Write in CE-ATA Mode....6441
Table 25-40. Register Call Summary for Main Sequence – Suspend Flow....6442
Table 25-41. Subprocess Call Summary for Main Sequence – Suspend Flow....6443
Table 25-42. Register Call Summary for Main Sequence - Resume Flow....6443
Table 25-43. Subprocess Call Summary for Main Sequence - Resume Flow....6443
Table 25-44. Register Call Summary for Main Sequence – Command Transfer Flow With Polling....6444
Table 25-45. Subprocess Call Summary for Main Sequence – Command Transfer Flow With Polling....6445
Table 25-46. Register Call Summary for Main Sequence – Command Transfer Flow With Interrupts....6446
Table 25-47. Subprocess Call Summary for Main Sequence – Command Transfer Flow With Interrupts....6447
Table 25-48. Register Call Summary for Main Sequence – Clock Frequency Change Flow....6447
Table 25-49. Register Call Summary for Main Sequence – Bus Width Configuration Flow....6448
Table 25-50. Subprocess Call Summary for Main Sequence – Bus Width Configuration Flow....6448
Table 25-51. Register Call Summary for Main Sequence – Power Switching Procedure....6450
Table 25-52. Subprocess Call Summary for Main Sequence – Power Switching Procedure....6450
Table 25-53. Register Call Summary for Main Sequence – Boot Using CMD0....6451
Table 25-54. Subprocess Call Summary for Main Sequence – Boot Using CMD0....6451
Table 25-55. Register Call Summary for Main Sequence – Boot Using CMD0....6452
Table 25-56. Subprocess Call Summary for Main Sequence – Boot Using CMD0....6453
Table 25-57. eMMC/SD/SDIO Instance Summary....6455
Table 25-58. MMC Registers Mapping Summary....6455
Table 25-59. MMC Registers Mapping Summary....6456
Table 25-60. MMCHS_HL_REV....6456
Table 25-61. MMCHS_HL_HWINFO....6457
Table 25-62. MMCHS_HL_SYSCONFIG....6458
Table 25-63. MMCHS_SYSCONFIG....6459
Table 25-64. MMCHS_SYSSTATUS....6461
Table 25-65. MMCHS_CSRE....6461
Table 25-66. MMCHS_SYSTEST....6461
Table 25-67. MMCHS_CON....6465
Table 25-68. MMCHS_PWCNT....6470
Table 25-69. MMCHS_DLL....6470
Table 25-70. MMCHS_SDMASA....6472

Table 25-71. MMCHS_BLK....6472
Table 25-72. MMCHS_ARG....6473
Table 25-73. MMCHS_CMD....6474
Table 25-74. MMCHS_RSP10....6478
Table 25-75. MMCHS_RSP32....6478
Table 25-76. MMCHS_RSP54....6478
Table 25-77. MMCHS_RSP76....6479
Table 25-78. MMCHS_DATA....6479
Table 25-79. MMCHS_PSTATE....6480
Table 25-80. MMCHS_HCTL....6483
Table 25-81. MMCHS_SYSCTL....6486
Table 25-82. MMCHS_STAT....6489
Table 25-83. MMCHS_IE....6495
Table 25-84. MMCHS_ISE....6497
Table 25-85. MMCHS_AC12....6499
Table 25-86. MMCHS_CAPA....6503
Table 25-87. MMCHS_CAPA2....6506
Table 25-88. Supported Data Rate Modes....6509
Table 25-89. MMCHS_CUR_CAPA....6509
Table 25-90. MMCHS_FE....6509
Table 25-91. MMCHS_ADMAES....6511
Table 25-92. MMCHS_ADMASAL....6512
Table 25-93. MMCHS_PVINITSD....6513
Table 25-94. MMCHS_PVHSSDR12....6514
Table 25-95. MMCHS_PVSDR25SDR50....6515
Table 25-96. MMCHS_PVSDR104DDR50....6516
Table 25-97. MMCHS_REV....6517
Table 26-1. SATA PHY I/O Signals....6522
Table 26-2. SATA PHY Integration Attributes....6525
Table 26-3. SATA PHY Clocks....6525
Table 26-4. SATA PHY Resets....6525
Table 26-5. Register Call Summary for SATA PLL Programming Sequence....6536
Table 26-6. Recommended Programming Values....6536
Table 26-7. SATA PHY Subsystem Low-Level Programming Sequence....6537
Table 26-8. DPLL CLKDCOLDO Recommended Settings....6537
Table 26-9. SATA PHY Tuning Table....6537
Table 26-10. USB3_PHY I/O Signals....6541
Table 26-11. Integration Attributes....6543
Table 26-12. Clocks....6543
Table 26-13. Resets....6544
Table 26-14. Register Call Summary for USB3_PHY PLL Programming Sequence....6555
Table 26-15. Recommended Programming Values....6555
Table 26-16. USB3_PHY Subsystem Low-Level Programming Sequence....6556
Table 26-17. USB3_PHY Tuning Table....6556
Table 26-18. USB3 PHY and SATA PHY Instance Summary....6558
Table 26-19. USB3_PHY_RX Registers Mapping Summary....6558
Table 26-20. USB3PHYRX_ANA_PROGRAMMABILITY_REG1....6559
Table 26-21. USB3PHYRX_TRIM_REG4....6559
Table 26-22. USB3PHYRX_DLL_REG1....6560
Table 26-23. USB3PHYRX_DIGITAL_MODES_REG1....6560
Table 26-24. USB3PHYRX_EQUALIZER_REG1....6561
Table 26-25. USB3_PHY_TX Registers Mapping Summary....6563
Table 26-26. USB3PHYTX_FUNC_CONFIG_REG....6563
Table 26-27. USB3PHYTX_TEST_CONFIG_REG....6563
Table 26-28. USB3PHYTX_PATTGEN_PRELOAD....6564
Table 26-29. SATA_PHY_RX Registers Mapping Summary....6565
Table 26-30. SATAPHYRX_ANA_PROGRAMMABILITY_REG1....6565
Table 26-31. SATAPHYRX_TRIM_REG4....6566
Table 26-32. SATAPHYRX_DLL_REG1....6566
Table 26-33. SATAPHYRX_DIGITAL_MODES_REG1....6567
Table 26-34. SATAPHYRX_EQUALIZER_REG1....6568

Table 26-35. SATAPHYRX_IO_AND_A2D_OVERRIDES_REG1 ... 6569
Table 26-36. SATA_PHY_TX Registers Mapping Summary ... 6570
Table 26-37. SATAPHYTX_FUNC_CONFIG_REG ... 6570
Table 26-38. SATAPHYTX_TEST_CONFIG_REG ... 6570
Table 26-39. SATAPHYTX_PATTGEN_PRELOAD ... 6571
Table 26-40. DPLLCTRL Registers Mapping Summary ... 6572
Table 26-41. PLL_STATUS ... 6572
Table 26-42. PLL_GO ... 6573
Table 26-43. PLL_CONFIGURATION1 ... 6573
Table 26-44. PLL_CONFIGURATION2 ... 6574
Table 26-45. PLL_CONFIGURATION3 ... 6574
Table 26-46. PLL_SSC_CONFIGURATION1 ... 6575
Table 26-47. PLL_SSC_CONFIGURATION2 ... 6575
Table 26-48. PLL_CONFIGURATION4 ... 6575
Table 26-49. PCIe PHY I/O Signals ... 6580
Table 26-50. Integration Attributes ... 6583
Table 26-51. Clocks ... 6584
Table 26-52. Resets ... 6584
Table 26-53. DPLL_PCIE_REF Recommended Configuration ... 6595
Table 26-54. Register Call Summary for DPLL_PCIE_REF Programming Sequence ... 6596
Table 26-55. Recommended Programming Values ... 6596
Table 26-56. PCIePHY Subsystem Low-Level Programming Sequence ... 6600
Table 26-57. Preferred PCIe_PHY_RX SCP Register Settings ... 6602
Table 26-58. PCIe PHY Subsystem Instance Summary ... 6603
Table 26-59. PCIe1_PHY_RX Registers Mapping Summary ... 6603
Table 26-60. PCIe2_PHY_RX Registers Mapping Summary ... 6603
Table 26-61. PCIEPHYRX_ANA_PROGRAMMABILITY_REG1 ... 6604
Table 26-62. PCIEPHYRX_TRIM_REG4 ... 6605
Table 26-63. PCIEPHYRX_DLL_REG1 ... 6605
Table 26-64. PCIEPHYRX_DIGITAL_MODES_REG1 ... 6606
Table 26-65. PCIEPHYRX_EQUALIZER_REG1 ... 6607
Table 26-66. PCIe1_PHY_TX Registers Mapping Summary ... 6609
Table 26-67. PCIe2_PHY_TX Registers Mapping Summary ... 6609
Table 26-68. PCIEPHYTX_FUNC_CONFIG_REG ... 6609
Table 26-69. PCIEPHYTX_DRIVER_DATA_CONFIG1 ... 6610
Table 26-70. PCIEPHYTX_TEST_CONFIG_REG ... 6610
Table 26-71. PCIEPHYTX_PATTGEN_PRELOAD ... 6611
Table 26-72. OCP2SCP Registers Mapping Summary ... 6612
Table 26-73. OCP2SCP_REVISION ... 6612
Table 26-74. OCP2SCP_SYSCONFIG ... 6612
Table 26-75. OCP2SCP_SYSSTATUS ... 6613
Table 26-76. OCP2SCP_TIMING ... 6613
Table 27-1. I/O Description ... 6619
Table 27-2. GPIO Integration Attributes ... 6623
Table 27-3. GPIO Clocks and Resets ... 6623
Table 27-4. GPIO Hardware Requests ... 6625
Table 27-5. Functional Clock Configuration ... 6633
Table 27-6. Local Power-Management Features ... 6636
Table 27-7. Clock Activity Settings ... 6636
Table 27-8. Events ... 6637
Table 27-9. Wake-Up Signals ... 6639
Table 27-10. GPIO Channels Description ... 6639
Table 27-11. Global Initialization of Surrounding Modules ... 6643
Table 27-12. General-Purpose Interface Global Initialization ... 6643
Table 27-13. General-Purpose Interface Read Input Register ... 6644
Table 27-14. General-Purpose Interface Set Bit Function ... 6644
Table 27-15. General-Purpose Interface Clear Bit Function ... 6644
Table 27-16. Instance Summary ... 6645
Table 27-17. General-Purpose Interface GPIO2, GPIO7 and GPIO8 Registers Summary ... 6645
Table 27-18. General-Purpose Interface GPIO3 to GPIO5 Registers Summary ... 6646
Table 27-19. General-Purpose Interface GPIO6 and GPIO1 Registers Summary ... 6647

Table 27-20. GPIO_REVISION....6649
Table 27-21. GPIO_SYSCONFIG....6649
Table 27-22. GPIO_EOI....6650
Table 27-23. GPIO_IRQSTATUS_RAW_0....6650
Table 27-24. GPIO_IRQSTATUS_RAW_1....6651
Table 27-25. GPIO_IRQSTATUS_0....6651
Table 27-26. GPIO_IRQSTATUS_1....6651
Table 27-27. GPIO_IRQSTATUS_SET_0....6652
Table 27-28. GPIO_IRQSTATUS_SET_1....6652
Table 27-29. GPIO_IRQSTATUS_CLR_0....6652
Table 27-30. GPIO_IRQSTATUS_CLR_1....6653
Table 27-31. GPIO_IRQWAKEN_0....6653
Table 27-32. GPIO_IRQWAKEN_1....6654
Table 27-33. GPIO_SYSSTATUS....6654
Table 27-34. GPIO_CTRL....6654
Table 27-35. GPIO_OE....6655
Table 27-36. GPIO_DATAIN....6655
Table 27-37. GPIO_DATAOUT....6656
Table 27-38. GPIO_LEVELDETECT0....6656
Table 27-39. GPIO_LEVELDETECT1....6657
Table 27-40. GPIO_RISINGDETECT....6657
Table 27-41. GPIO_FALLINGDETECT....6657
Table 27-42. GPIO_DEBOUNCENABLE....6658
Table 27-43. GPIO_DEBOUNCINGTIME....6658
Table 27-44. GPIO_CLEARDATAOUT....6658
Table 27-45. GPIO_SETDATAOUT....6659
Table 28-1. I/O External Keyboard Signals....6664
Table 28-2. Keyboard Controller Integration Attributes....6666
Table 28-3. Keyboard Controller Clocks and Resets....6666
Table 28-4. Keyboard Controller Hardware Requests....6667
Table 28-5. Local Power-Management Features....6669
Table 28-6. Events....6669
Table 28-7. Keyboard Controller Functional Modes....6670
Table 28-8. Keyboard Controller Timer Values....6670
Table 28-9. Timer Prescale Values....6671
Table 28-10. State-Machine Values....6672
Table 28-11. Global Initialization of Surrounding Modules....6677
Table 28-12. Keyboard Controller Global Initialization....6677
Table 28-13. Keyboard Controller Hardware Mode....6678
Table 28-14. Keyboard Controller Software Mode....6679
Table 28-15. Keyboard Controller Event Servicing....6679
Table 28-16. Keyboard Controller Instance Summary....6681
Table 28-17. Keyboard Controller Register Mapping Summary....6681
Table 28-18. KBD_REVISION....6681
Table 28-19. KBD_SYSCONFIG....6682
Table 28-20. KBD_EOI....6682
Table 28-21. KBD_IRQSTATUS_RAW....6683
Table 28-22. KBD_IRQSTATUS....6683
Table 28-23. KBD_IRQENABLE_SET....6684
Table 28-24. KBD_IRQENABLE_CLR....6685
Table 28-25. KBD_IRQWAKEEN....6685
Table 28-26. KBD_PENDING....6686
Table 28-27. KBD_CTRL....6687
Table 28-28. KBD_DEBOUNCINGTIME....6688
Table 28-29. KBD_KEYLONGTIME....6688
Table 28-30. KBD_TIMEOUT....6688
Table 28-31. KBD_STATEMACHINE....6688
Table 28-32. KBD_ROWINPUTS....6689
Table 28-33. KBD_COLUMNOUTPUTS....6689
Table 28-34. KBD_FULLCODE31_0....6689
Table 28-35. KBD_FULLCODE63_32....6689

Table 28-36. KBD_FULLCODE17_0........ 6690
Table 28-37. KBD_FULLCODE35_18........ 6690
Table 28-38. KBD_FULLCODE53_36........ 6690
Table 28-39. KBD_FULLCODE71_54........ 6691
Table 28-40. KBD_FULLCODE80_72........ 6691
Table 29-1. PWMSS Unsupported Features........ 6694
Table 29-2. PWM Subsystems I/O Signals........ 6695
Table 29-3. PWMSS Integration Attributes........ 6699
Table 29-4. PWMSS Clocks and Resets........ 6699
Table 29-5. PWMSS Hardware Requests........ 6699
Table 29-6. Device Limitations for the eHRPWM and eQEP Functional Interfaces of PWMSSn........ 6701
Table 29-7. Local IDLE Clock Management Features........ 6703
Table 29-8. Local Module Clock Control and Status Features........ 6703
Table 29-9. PWMSS_CFG Instance Summary........ 6704
Table 29-10. PWMSSn_CFG Registers Mapping Summary........ 6704
Table 29-11. PWMSS_IDVER........ 6704
Table 29-12. PWMSS_SYSCONFIG........ 6704
Table 29-13. PWMSS_CLKCONFIG........ 6705
Table 29-14. PWMSS_CLKSTATUS........ 6706
Table 29-15. Submodule Configuration Parameters........ 6712
Table 29-16. ePWM Time-Base Submodule Registers........ 6717
Table 29-17. ePWM Key Time-Base Signals........ 6717
Table 29-18. ePWM Counter-Compare Submodule Registers........ 6727
Table 29-19. ePWM Counter-Compare Submodule Key Signals........ 6727
Table 29-20. Action-Qualifier Submodule Registers........ 6731
Table 29-21. ePWM Action-Qualifier Submodule Possible Input Events........ 6732
Table 29-22. ePWM Action-Qualifier Event Priority for Up-Down-Count Mode........ 6734
Table 29-23. ePWM Action-Qualifier Event Priority for Up-Count Mode........ 6734
Table 29-24. ePWM Action-Qualifier Event Priority for Down-Count Mode........ 6734
Table 29-25. Behavior if CMPA/CMPB is Greater than the Period........ 6735
Table 29-26. EPWMx Initialization for Figure 29-26 6738
Table 29-27. EPWMx Run Time Changes for Figure 29-26 6738
Table 29-28. EPWMx Initialization for Figure 29-27 6740
Table 29-29. EPWMx Run Time Changes for Figure 29-27 6740
Table 29-30. EPWMx Initialization for Figure 29-28 6742
Table 29-31. EPWMx Run Time Changes for Figure 29-28 6742
Table 29-32. EPWMx Initialization for Figure 29-29 6744
Table 29-33. EPWMx Run Time Changes for Figure 29-29 6744
Table 29-34. EPWMx Initialization for Figure 29-30 6746
Table 29-35. EPWMx Run Time Changes for Figure 29-30 6746
Table 29-36. EPWMx Initialization for Figure 29-31 6748
Table 29-37. EPWMx Run Time Changes for Figure 29-31 6748
Table 29-38. Dead-Band Generator Submodule Registers........ 6749
Table 29-39. Classical Dead-Band Operating Modes........ 6751
Table 29-40. PWM-Chopper Submodule Registers........ 6753
Table 29-41. ePWM Trip-Zone Submodule Registers........ 6758
Table 29-42. Possible Actions On an ePWM Trip Event........ 6759
Table 29-43. Event-Trigger Submodule Registers........ 6762
Table 29-44. Resolution for PWM and HRPWM........ 6767
Table 29-45. HRPWM Submodule Registers........ 6768
Table 29-46. Relationship Between MEP Steps, PWM Frequency and Resolution........ 6769
Table 29-47. CMPA vs Duty (left), and [CMPA:CMPAHR] vs Duty (right)........ 6770
Table 29-48. ePWM/HRPWM Module Control and Status Registers Grouped by Submodule........ 6772
Table 29-49. PWMSS_EPWM Instance Summary........ 6774
Table 29-50. PWMSSn_EPWM Registers Mapping Summary........ 6774
Table 29-51. EPWM_TBCTL........ 6775
Table 29-52. Register Call Summary for Register EPWM_TBCTL........ 6777
Table 29-53. EPWM_TBSTS........ 6777
Table 29-54. Register Call Summary for Register EPWM_TBSTS........ 6778
Table 29-55. HRPWM_TBPHSHR........ 6778
Table 29-56. Register Call Summary for Register HRPWM_TBPHSHR........ 6778

Table 29-57. EPWM_TBPHS....6778
Table 29-58. Register Call Summary for Register EPWM_TBPHS....6779
Table 29-59. EPWM_TBCNT....6779
Table 29-60. Register Call Summary for Register EPWM_TBCNT....6779
Table 29-61. EPWM_TBPRD....6779
Table 29-62. Register Call Summary for Register EPWM_TBPRD....6780
Table 29-63. EPWM_CMPCTL....6780
Table 29-64. Register Call Summary for Register EPWM_CMPCTL....6782
Table 29-65. HRPWM_CMPAHR....6782
Table 29-66. Register Call Summary for Register HRPWM_CMPAHR....6782
Table 29-67. EPWM_CMPA....6782
Table 29-68. Register Call Summary for Register EPWM_CMPA....6783
Table 29-69. EPWM_CMPB....6783
Table 29-70. Register Call Summary for Register EPWM_CMPB....6784
Table 29-71. EPWM_AQCTLA....6784
Table 29-72. Register Call Summary for Register EPWM_AQCTLA....6786
Table 29-73. EPWM_AQCTLB....6786
Table 29-74. Register Call Summary for Register EPWM_AQCTLB....6787
Table 29-75. EPWM_AQSFRC....6787
Table 29-76. Register Call Summary for Register EPWM_AQSFRC....6788
Table 29-77. EPWM_AQCSFRC....6788
Table 29-78. Register Call Summary for Register EPWM_AQCSFRC....6789
Table 29-79. EPWM_DBCTL....6789
Table 29-80. Register Call Summary for Register EPWM_DBCTL....6790
Table 29-81. EPWM_DBRED....6790
Table 29-82. Register Call Summary for Register EPWM_DBRED....6791
Table 29-83. EPWM_DBFED....6791
Table 29-84. Register Call Summary for Register EPWM_DBFED....6791
Table 29-85. EPWM_TZSEL....6791
Table 29-86. Register Call Summary for Register EPWM_TZSEL....6792
Table 29-87. EPWM_TZCTL....6792
Table 29-88. Register Call Summary for Register EPWM_TZCTL....6793
Table 29-89. EPWM_TZEINT....6793
Table 29-90. Register Call Summary for Register EPWM_TZEINT....6793
Table 29-91. EPWM_TZFLG....6793
Table 29-92. Register Call Summary for Register EPWM_TZFLG....6794
Table 29-93. EPWM_TZCLR....6794
Table 29-94. Register Call Summary for Register EPWM_TZCLR....6795
Table 29-95. EPWM_TZFRC....6795
Table 29-96. Register Call Summary for Register EPWM_TZFRC....6796
Table 29-97. EPWM_ETSEL....6796
Table 29-98. Register Call Summary for Register EPWM_ETSEL....6796
Table 29-99. EPWM_ETPS....6797
Table 29-100. Register Call Summary for Register EPWM_ETPS....6798
Table 29-101. EPWM_ETFLG....6798
Table 29-102. Register Call Summary for Register EPWM_ETFLG....6798
Table 29-103. EPWM_ETCLR....6798
Table 29-104. Register Call Summary for Register EPWM_ETCLR....6799
Table 29-105. EPWM_ETFRC....6799
Table 29-106. Register Call Summary for Register EPWM_ETFRC....6799
Table 29-107. EPWM_PCCTL....6799
Table 29-108. Register Call Summary for Register EPWM_PCCTL....6800
Table 29-109. HRPWM_HRCTL....6800
Table 29-110. Register Call Summary for Register HRPWM_HRCTL....6801
Table 29-111. eCAP Control and Status Functional Registers....6812
Table 29-112. PWMSS_ECAP Instance Summary....6813
Table 29-113. PWMSSn_ECAP Registers Mapping Summary....6813
Table 29-114. PWMSS_ECAP_TSCNT....6813
Table 29-115. Register Call Summary for Register PWMSS_ECAP_TSCNT....6813
Table 29-116. PWMSS_ECAP_CNTPHS....6814
Table 29-117. Register Call Summary for Register PWMSS_ECAP_CNTPHS....6814

Table 29-118. PWMSS_ECAP_CAP1....6814
Table 29-119. Register Call Summary for Register PWMSS_ECAP_CAP1....6814
Table 29-120. PWMSS_ECAP_CAP2....6815
Table 29-121. Register Call Summary for Register PWMSS_ECAP_CAP2....6815
Table 29-122. PWMSS_ECAP_CAP3....6815
Table 29-123. Register Call Summary for Register PWMSS_ECAP_CAP3....6815
Table 29-124. PWMSS_ECAP_CAP4....6815
Table 29-125. Register Call Summary for Register PWMSS_ECAP_CAP4....6816
Table 29-126. PWMSS_ECAP_ECCTL1....6816
Table 29-127. Register Call Summary for Register PWMSS_ECAP_ECCTL1....6817
Table 29-128. PWMSS_ECAP_ECCTL2....6817
Table 29-129. Register Call Summary for Register PWMSS_ECAP_ECCTL2....6819
Table 29-130. PWMSS_ECAP_ECEINT....6819
Table 29-131. Register Call Summary for Register PWMSS_ECAP_ECEINT....6820
Table 29-132. PWMSS_ECAP_ECFLG....6820
Table 29-133. Register Call Summary for Register PWMSS_ECAP_ECFLG....6821
Table 29-134. PWMSS_ECAP_ECCLR....6821
Table 29-135. Register Call Summary for Register PWMSS_ECAP_ECCLR....6822
Table 29-136. PWMSS_ECAP_ECFRC....6822
Table 29-137. Register Call Summary for Register PWMSS_ECAP_ECFRC....6823
Table 29-138. PWMSS_ECAP_PID....6823
Table 29-139. Register Call Summary for Register PWMSS_ECAP_PID....6823
Table 29-140. Quadrature Decoder Truth Table....6831
Table 29-141. eQEP Control and Status Functional Registers....6846
Table 29-142. PWMSS_EQEP Instance Summary....6846
Table 29-143. PWMSSn_EQEP Registers Mapping Summary....6847
Table 29-144. EQEP_QPOSCNT....6848
Table 29-145. Register Call Summary for Register EQEP_QPOSCNT....6848
Table 29-146. EQEP_QPOSINIT....6848
Table 29-147. Register Call Summary for Register EQEP_QPOSINIT....6848
Table 29-148. EQEP_QPOSMAX....6848
Table 29-149. Register Call Summary for Register EQEP_QPOSMAX....6849
Table 29-150. EQEP_QPOSCMP....6849
Table 29-151. Register Call Summary for Register EQEP_QPOSCMP....6849
Table 29-152. EQEP_QPOSILAT....6849
Table 29-153. Register Call Summary for Register EQEP_QPOSILAT....6850
Table 29-154. EQEP_QPOSSLAT....6850
Table 29-155. Register Call Summary for Register EQEP_QPOSSLAT....6850
Table 29-156. EQEP_QPOSLAT....6850
Table 29-157. Register Call Summary for Register EQEP_QPOSLAT....6850
Table 29-158. EQEP_QUTMR....6850
Table 29-159. Register Call Summary for Register EQEP_QUTMR....6851
Table 29-160. EQEP_QUPRD....6851
Table 29-161. Register Call Summary for Register EQEP_QUPRD....6851
Table 29-162. EQEP_QWDTMR....6851
Table 29-163. Register Call Summary for Register EQEP_QWDTMR....6852
Table 29-164. EQEP_QWDPRD....6852
Table 29-165. Register Call Summary for Register EQEP_QWDPRD....6852
Table 29-166. EQEP_QDECCTL....6852
Table 29-167. Register Call Summary for Register EQEP_QDECCTL....6853
Table 29-168. EQEP_QEPCTL....6853
Table 29-169. Register Call Summary for Register EQEP_QEPCTL....6855
Table 29-170. EQEP_QCAPCTL....6855
Table 29-171. Register Call Summary for Register EQEP_QCAPCTL....6856
Table 29-172. EQEP_QPOSCTL....6856
Table 29-173. Register Call Summary for Register EQEP_QPOSCTL....6857
Table 29-174. EQEP_QEINT....6857
Table 29-175. Register Call Summary for Register EQEP_QEINT....6858
Table 29-176. EQEP_QFLG....6858
Table 29-177. Register Call Summary for Register EQEP_QFLG....6859
Table 29-178. EQEP_QCLR....6859

Table 29-179. Register Call Summary for Register EQEP_QCLR....... 6860
Table 29-180. EQEP_QFRC....... 6860
Table 29-181. Register Call Summary for Register EQEP_QFRC....... 6862
Table 29-182. EQEP_QEPSTS....... 6862
Table 29-183. Register Call Summary for Register EQEP_QEPSTS....... 6863
Table 29-184. EQEP_QCTMR....... 6863
Table 29-185. Register Call Summary for Register EQEP_QCTMR....... 6863
Table 29-186. EQEP_QCPRD....... 6863
Table 29-187. Register Call Summary for Register EQEP_QCPRD....... 6863
Table 29-188. EQEP_QCTMRLAT....... 6863
Table 29-189. Register Call Summary for Register EQEP_QCTMRLAT....... 6864
Table 29-190. EQEP_QCPRDLAT....... 6864
Table 29-191. Register Call Summary for Register EQEP_QCPRDLAT....... 6864
Table 29-192. EQEP_REVID....... 6864
Table 29-193. Register Call Summary for Register EQEP_REVID....... 6865
Table 30-1. VCP Integration Attributes....... 6869
Table 30-2. VCP Clocks and Resets....... 6870
Table 30-3. VCP Hardware Requests....... 6870
Table 30-4. Local Power-Management Features....... 6872
Table 30-5. VCP1 and VCP2 Reset Description....... 6873
Table 30-6. VCP1 and VCP2 Interrupts Description....... 6873
Table 30-7. Traceback Soft Decision Sliding Window Limits....... 6878
Table 30-8. Traceback Hard Decision Sliding Window Limits....... 6878
Table 30-9. Traceback Hard Decision Sliding Window Limits....... 6879
Table 30-10. Branch Metrics for Rate 1/2....... 6881
Table 30-11. Branch Metrics for Rate 1/3....... 6881
Table 30-12. Branch Metrics for Rate 1/4....... 6881
Table 30-13. VCP Soft Inputs Quantization....... 6882
Table 30-14. VCP Memory Sleep Mode Truth Table....... 6882
Table 30-15. Branch Metrics....... 6883
Table 30-16. Branch Metrics in DSP Memory (BM = 1)....... 6883
Table 30-17. Branch Metrics in DSP Memory (BM = 0)....... 6884
Table 30-18. Soft Decision Organization....... 6886
Table 30-19. E_SYMR 0 (No error for SMAR)....... 6888
Table 30-20. E_SYMX 0 (No error for SMAX)....... 6888
Table 30-21. MAXIMINERR 0 (Error Check)....... 6889
Table 30-22. Required EDMA Links Per User Channel....... 6890
Table 30-23. EDMA Transfer Options Parameters....... 6891
Table 30-24. Configuration EDMA Transfer to the Branch Metrics FIFO....... 6892
Table 30-25. Configuration EDMA Hard-Decisions Mode....... 6893
Table 30-26. Configuration EDMA Soft-Decisions Mode....... 6894
Table 30-27. Configuration EDMA Output Parameters Transfer....... 6895
Table 30-28. VCP1 and VCP2 Instance Summary....... 6897
Table 30-29. VCP DATA Registers Summary....... 6897
Table 30-30. VCP Configuration Registers Summary....... 6897
Table 30-31. VCP1 and VCP2 Memories Register Summary....... 6898
Table 30-32. VCP_VCPIC0....... 6898
Table 30-33. Register Call Summary for Register VCP_VCPIC0....... 6899
Table 30-34. VCP_VCPIC1....... 6899
Table 30-35. Register Call Summary for Register VCP_VCPIC1....... 6899
Table 30-36. VCP_VCPIC2....... 6899
Table 30-37. Register Call Summary for Register VCP_VCPIC2....... 6900
Table 30-38. VCP_VCPIC3....... 6900
Table 30-39. Register Call Summary for Register VCP_VCPIC3....... 6900
Table 30-40. VCP_VCPIC4....... 6901
Table 30-41. Register Call Summary for Register VCP_VCPIC4....... 6901
Table 30-42. VCP_VCPIC5....... 6901
Table 30-43. Register Call Summary for Register VCP_VCPIC5....... 6902
Table 30-44. VCP_VCPOUT0....... 6902
Table 30-45. Register Call Summary for Register VCP_VCPOUT0....... 6902
Table 30-46. VCP_VCPOUT1....... 6903

Table 30-47. Register Call Summary for Register VCP_VCPOUT1........6903
Table 30-48. VCP_VCPWBM........6903
Table 30-49. Register Call Summary for Register VCP_VCPWBM........6904
Table 30-50. VCP_VCPRDECS........6904
Table 30-51. Register Call Summary for Register VCP_VCPRDECS........6904
Table 30-52. REVISION........6904
Table 30-53. Register Call Summary for Register REVISION........6904
Table 30-54. VCP_SYSCONFIG........6904
Table 30-55. Register Call Summary for Register VCP_SYSCONFIG........6905
Table 30-56. VCP_IRQ_EOI........6905
Table 30-57. Register Call Summary for Register VCP_IRQ_EOI........6905
Table 30-58. VCP_IRQSTATUS........6905
Table 30-59. Register Call Summary for Register VCP_IRQSTATUS........6906
Table 30-60. VCP_IRQENABLE_SET........6906
Table 30-61. Register Call Summary for Register VCP_IRQENABLE_SET........6906
Table 30-62. VCP_IRQENABLE_CLR........6907
Table 30-63. Register Call Summary for Register VCP_IRQENABLE_CLR........6907
Table 30-64. VCP_DEBUG........6907
Table 30-65. Register Call Summary for Register VCP_DEBUG........6908
Table 30-66. VCP_VCPEXE........6908
Table 30-67. Register Call Summary for Register VCP_VCPEXE........6908
Table 30-68. VCP_VCPEND........6908
Table 30-69. Register Call Summary for Register VCP_VCPEND........6909
Table 30-70. VCP_VCPSTAT0........6909
Table 30-71. Register Call Summary for Register VCP_VCPSTAT0........6910
Table 30-72. VCP_VCPSTAT1........6910
Table 30-73. Register Call Summary for Register VCP_VCPSTAT1........6911
Table 30-74. VCP_VCPERR........6911
Table 30-75. Register Call Summary for Register VCP_VCPERR........6912
Table 30-76. VCP_VCPEMU........6912
Table 30-77. Register Call Summary for Register VCP_VCPEMU........6912
Table 31-1. I/O Description........6915
Table 31-2. Integration Attributes........6917
Table 31-3. Clocks and Resets........6917
Table 31-4. L4_PER2 TAs........6917
Table 31-5. Region Allocations for L4_PER2 Interconnect........6917
Table 31-6. ATL Instance Summary........6921
Table 31-7. ATL Registers Mapping Summary........6921
Table 31-8. ATL_REVID........6922
Table 31-9. ATL_PPMR0........6922
Table 31-10. ATL_BBSR0........6922
Table 31-11. ATL_ATLCR0........6923
Table 31-12. ATL_SWEN0........6923
Table 31-13. ATL_BWSMUX0........6923
Table 31-14. ATL_AWSMUX0........6924
Table 31-15. ATL_PCLKMUX0........6925
Table 31-16. ATL_PPMR1........6925
Table 31-17. ATL_BBSR1........6926
Table 31-18. ATL_ATLCR1........6926
Table 31-19. ATL_SWEN1........6926
Table 31-20. ATL_BWSMUX1........6927
Table 31-21. ATL_AWSMUX1........6927
Table 31-22. ATL_PCLKMUX1........6928
Table 31-23. ATL_PPMR2........6928
Table 31-24. ATL_BBSR2........6929
Table 31-25. ATL_ATLCR2........6929
Table 31-26. ATL_SWEN2........6929
Table 31-27. ATL_BWSMUX2........6930
Table 31-28. ATL_AWSMUX2........6930
Table 31-29. ATL_PCLKMUX2........6931
Table 31-30. ATL_PPMR3........6932

Table 31-31. ATL_BBSR3....6932
Table 31-32. ATL_ATLCR3....6932
Table 31-33. ATL_SWEN3....6933
Table 31-34. ATL_BWSMUX3....6933
Table 31-35. ATL_AWSMUX3....6934
Table 31-36. ATL_PCLKMUX3....6934
Table 32-1. Device Power Balls....6941
Table 32-2. Mapping for Input Sources....6944
Table 32-3. PMIC Clock Requirements....6944
Table 32-4. Sysboot Pads Description....6947
Table 32-5. MMC2 Configuration (SR2.0)....6947
Table 32-6. GPMC for XIP/NAND Configuration....6947
Table 32-7. System Clock (SYS_CLK1) Speed Selection....6948
Table 32-8. Offset Between Redundant Images....6948
Table 32-9. Booting Devices Order....6949
Table 32-10. Pin Multiplexing According to Boot Peripheral....6950
Table 32-11. ROM Exception Vectors....6953
Table 32-12. Dead Loops....6954
Table 32-13. RAM Exception Vectors....6955
Table 32-14. Tracing Data....6955
Table 32-15. Control Module Registers Modified by ROM Code at Each Startup....6960
Table 32-16. PRCM Module Mode Registers Modified by ROM Code....6960
Table 32-17. ROM Code Default Clock Settings....6961
Table 32-18. ASIC ID Structure....6964
Table 32-19. Items....6965
Table 32-20. ID Subblock....6965
Table 32-21. Checksum Subblock....6965
Table 32-22. Booting Messages....6965
Table 32-23. Device Descriptor....6970
Table 32-24. Device-Qualifier Descriptor....6970
Table 32-25. Configuration Descriptor....6971
Table 32-26. Other Speed Configuration Descriptor....6971
Table 32-27. Interface Descriptor....6971
Table 32-28. BULK IN Endpoint Descriptor....6972
Table 32-29. BULK OUT Endpoint Descriptor....6972
Table 32-30. Language ID String Descriptor....6973
Table 32-31. Manufacturer ID String Descriptor....6973
Table 32-32. Product ID String Descriptor....6973
Table 32-33. Configuration String Descriptor....6973
Table 32-34. Interface String Descriptor....6973
Table 32-35. USB ROM Standard Device Descriptor....6973
Table 32-36. USB ROM Descriptor Strings....6974
Table 32-37. Standard Device Requests Supported....6974
Table 32-38. XIP Timing Parameters....6981
Table 32-39. NAND Timing Parameters....6982
Table 32-40. ONFI Parameters Page Description....6982
Table 32-41. Supported NAND Devices....6983
Table 32-42. Fourth NAND ID Data Byte....6984
Table 32-43. Master Boot Record Structure....6998
Table 32-44. Partition Table Entry....6999
Table 32-45. FAT Directory Entry....7002
Table 32-46. FAT Entry Description....7003
Table 32-47. CH TOC Item....7010
Table 32-48. TOC Filenames....7011
Table 32-49. CHSETTINGS Item....7011
Table 32-50. Clocking Settings....7011
Table 32-51. CHFLASH Item....7013
Table 32-52. CHMMCSD Item....7013
Table 32-53. CHQSPI Item....7014
Table 32-54. GP Header Image Format....7015
Table 32-55. Booting Parameter Structure....7015

Table 32-56. Tracing Vector 1 ... 7017
Table 32-57. Tracing Vector 2 ... 7018
Table 32-58. Tracing Vector 3 ... 7018
Table 32-59. Tracing Vector 4 ... 7019
Table 32-60. Start Hypervisor ... 7020
Table 32-61. Clean L1 and/or L2 cache ... 7020
Table 32-62. Write L2 Cache Auxiliary Control ... 7020
Table 32-63. Write Tag and Data RAM Latency Control Register ... 7021
Table 32-64. Write L2 Cache Prefetch Control Register ... 7021
Table 32-65. Write Auxiliary Control Register ... 7021
Table 32-66. Write AMBA IF Register ... 7021
Table 32-67. Set Timer CNTFRQ Register ... 7022
Table 33-1. IEEE1149.1 Signals ... 7028
Table 33-2. Trace Port Signals ... 7029
Table 33-3. ICEPick Boot Modes at POR ... 7031
Table 33-4. ICEPick Secondary Debug TAPs Mapping ... 7032
Table 33-5. ICEPick Debug Core Mapping ... 7033
Table 33-6. Device Cross-Triggering ... 7037
Table 33-7. Debug Subsystem Suspend Input Lines ... 7038
Table 33-8. Debug Subsystem Suspend Output Lines ... 7038
Table 33-9. Emulation Interrupts ... 7040
Table 33-10. L2 Cache Events ... 7044
Table 33-11. IPU SCTM Counters Repartition ... 7044
Table 33-12. SCTM Events for IPU Subsystem ... 7045
Table 33-13. EVE SCTM Counters Configuration ... 7046
Table 33-14. SCTM Events for EVE Subsystem ... 7047
Table 33-15. L3_MAIN Interconnect Functional Probe Mapping ... 7056
Table 33-16. Master-ID Mapping (Debug View) ... 7056
Table 33-17. Performance Monitoring Events Detection ... 7059
Table 33-18. Performance Filtering Options ... 7059
Table 33-19. Statistics Collector Master Address Mapping ... 7060
Table 33-20. Statistics Collector Slave Address Mapping ... 7061
Table 33-21. Aggregation Modes ... 7062
Table 33-22. SC_SDRAM Port Mapping ... 7063
Table 33-23. SC_LAT0 Port Mapping ... 7064
Table 33-24. SC_LAT1 Port Mapping ... 7065
Table 33-25. SC_LAT2 Port Mapping ... 7066
Table 33-26. SC_LAT3 Port Mapping ... 7067
Table 33-27. SC_LAT4 Port Mapping ... 7068
Table 33-28. SC_LAT5 Port Mapping ... 7069
Table 33-29. SC_LAT6 Port Mapping ... 7070
Table 33-30. SC_LAT7 Port Mapping ... 7071
Table 33-31. SC_LAT8 Port Mapping ... 7072
Table 33-32. STM Message Software Masters ... 7074
Table 33-33. STM Message Hardware Masters ... 7074
Table 33-34. Trace Port Configuration ... 7076
Table 33-35. Concurrent Debug and Trace ... 7076
Table 33-36. DRM Instance Summary ... 7077
Table 33-37. DRM Registers Mapping Summary ... 7077
Table 33-38. DRM_SUSPEND_CTRL0 ... 7078
Table 33-39. DRM_SUSPEND_CTRL1 ... 7079
Table 33-40. DRM_SUSPEND_CTRL2 ... 7080
Table 33-41. DRM_SUSPEND_CTRL3 ... 7080
Table 33-42. DRM_SUSPEND_CTRL4 ... 7081
Table 33-43. DRM_SUSPEND_CTRL5 ... 7082
Table 33-44. DRM_SUSPEND_CTRL6 ... 7083
Table 33-45. DRM_SUSPEND_CTRL7 ... 7084
Table 33-46. DRM_SUSPEND_CTRL8 ... 7084
Table 33-47. DRM_SUSPEND_CTRL9 ... 7085
Table 33-48. DRM_SUSPEND_CTRL10 ... 7086
Table 33-49. DRM_SUSPEND_CTRL11 ... 7087

Table 33-50. DRM_SUSPEND_CTRL12 ... 7088
Table 33-51. DRM_SUSPEND_CTRL13 ... 7088
Table 33-52. DRM_SUSPEND_CTRL14 ... 7089
Table 33-53. DRM_SUSPEND_CTRL15 ... 7090
Table 33-54. DRM_SUSPEND_CTRL16 ... 7091
Table 33-55. DRM_SUSPEND_CTRL17 ... 7092
Table 33-56. DRM_SUSPEND_CTRL18 ... 7092
Table 33-57. DRM_SUSPEND_CTRL19 ... 7093
Table 33-58. DRM_SUSPEND_CTRL20 ... 7094
Table 33-59. DRM_SUSPEND_CTRL21 ... 7095
Table 33-60. DRM_SUSPEND_CTRL22 ... 7096
Table 33-61. DRM_SUSPEND_CTRL23 ... 7096
Table 33-62. DRM_SUSPEND_CTRL24 ... 7097
Table 33-63. DRM_SUSPEND_CTRL25 ... 7098
Table 33-64. DRM_SUSPEND_CTRL26 ... 7099
Table 33-65. DRM_SUSPEND_CTRL27 ... 7100
Table 33-66. DRM_SUSPEND_CTRL28 ... 7100
Table 33-67. DRM_SUSPEND_CTRL29 ... 7101
Table 33-68. DRM_SUSPEND_CTRL30 ... 7102
Table 33-69. DRM_SUSPEND_CTRL31 ... 7103
Table 33-70. DRM_SUSPEND_CTRL32 ... 7104
Table 33-71. DRM_SUSPEND_CTRL33 ... 7104
Table 33-72. DRM_SUSPEND_CTRL34 ... 7105
Table 33-73. DRM_SUSPEND_CTRL35 ... 7106
Table 33-74. DRM_SUSPEND_CTRL36 ... 7107
Table 33-75. DRM_SUSPEND_CTRL37 ... 7108

This page intentionally left blank.

Preface

Read This First

Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

Glossary

TI Glossary	This glossary lists and explains terms, acronyms, and definitions.

About This Manual

Information About Cautions and Warnings

This book may contain cautions and warnings.

> **CAUTION**
>
> This is an example of a caution statement.
>
> A caution statement describes a situation that could potentially damage your software or equipment.

> **WARNING**
>
> This is an example of a warning statement.
>
> A warning statement describes a situation that could potentially cause harm to you.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

Register, Field, and Bit Calls

The naming convention applied for a call consists of:

- For a register call: *<Module name>.<Register name>*; for example: UART.UASR
- For a bit field call:
  - *<Module name>.<Register name>[End:Start] <Field name> field*; for example, UART.UASR[4:0] SPEED bit field
  - *<Field name> field <Module name>.<Register name>[End:Start]*; for example, SPEED bit field UART.UASR[4:0]
- For a bit call:
  - *<Module name>.<Register name>[pos] <Bit name> bit*; for example, UART.UASR[5] BIT_BY_CHAR bit
  - *<Bit name> bit <Module name>.<Register name>[pos]*; for example, BIT_BY_CHAR bit UART.UASR[5]

To help the reader navigate the document, each register call is hyperlinked to its register description in the register manual section. After each register description, a table summarizes all hyperlinked register calls.

Coding Rules

The programming models or code listings follow the rules:

Type	Definition	Example
File	Starts with the module name	PRCM_test1.c MCBSP1_init.h
Variable	Global variables are prefixed by "g_" Pointers are prefixed by "p" Global pointers are prefixed by "g_p"	g_SDMA_LogicalChan pAddrCounter g_pSDMA_LogicalChan
Function	Starts with the module name	PRCM_SetupClocks() ArmIntC_MaskInterrupts()
Typedef	Ends with "_t"	PRCM_Struct_t
Definition	Starts with the module name and is followed by the register name	#define SMS_ERR_TYPE *((volatileUint32*)0x680080F4) #define MCBSP2_RCR1_REG *((volatile Uint32*)0x4807401C)
Enumeration	Starts with the module name	Typedef enum DMA_Mode_Label { INPUT_MODE OUTPUT_MODE } DMA_Mode_t;

Flow Chart Rules

Flow charts follow the following rules:

Shape	Name	Definition
	Process	Any computational steps or processing function of a program; defined operation(s) causing change in value, form, or location of information
	Decision	A decision or switching-type operation that determines which of a number of alternate paths is followed
	Predefined process or sub-process	One or more named operations or program steps specified in a subroutine or another set of flow charts
	Data or I/O	General I/O function; information available for processing (input) or recording of processed information (output)
Start/Stop	Terminator	Terminal point in a flow chart: start, stop, halt, delay, or interrupt; may show exit from a closed subroutine
The height of the text box and its associated line increases or decreases as you add text. To change the width of the comment, drag the side handle.	Annotation	Additional descriptive clarification, comment
	On page connector (reference)	Exit to, or entry from, another part of chart in the same page
	Off page connector (reference)	The flow continues on a different page.
	Summing Junction	Logical AND
	Or	Logical OR
	Parallel mode (ISO)	Beginning or end of two or more simultaneous operations
	Flow Line	Lines indicate the sequence of steps and the direction of flow.

Export Control Notice

Recipient agrees to not knowingly export or re-export, directly or indirectly, any product or technical data (as defined by the U.S., EU, and other Export Administration Regulations) including software, or any controlled product restricted by other applicable national regulations, received from disclosing party under nondisclosure obligations (if any), or any direct product of such technology, to any destination to which such export or re-export is restricted or prohibited by U.S. or other applicable laws, without obtaining prior authorization from U.S. Department of Commerce and other competent Government authorities to the extent required by those laws.

DRA75x, DRA74x MIPI® Disclaimer

The material contained herein is not a license, either expressly or impliedly, to any IPR owned or controlled by any of the authors or developers of this material or MIPI. The material contained herein is provided on an "AS IS" basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and MIPI hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence.

ALSO, THERE IS NO WARRANTY OF CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT. IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT OR MIPI BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT, SPECIFICATION OR DOCUMENT RELATING TO THIS MATERIAL, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Without limiting the generality of this Disclaimer stated above, the user of the contents of this Document is further notified that MIPI: (a) does not evaluate, test or verify the accuracy, soundness or credibility of the contents of this Document; (b) does not monitor or enforce compliance with the contents of this Document; and (c) does not certify, test, or in any manner investigate products or services or any claims of compliance with the contents of this Document. The use or implementation of the contents of this Document may involve or require the use of intellectual property rights ("IPR") including (but not limited to) patents, patent applications, or copyrights owned by one or more parties, whether or not Members of MIPI. MIPI does not make any search or investigation for IPR, nor does MIPI require or request the disclosure of any IPR or claims of IPR as respects the contents of this Document or otherwise. Questions pertaining to this document, or the terms or conditions of its provision, should be addressed to:

MIPI Alliance, Inc. c/o IEEE-ISTO 445 Hoes Lane Piscataway, NJ 08854 Attn: Board Secretary

Trademarks

OMAP, ICECrusher and SmartReflex are trademarks of Texas Instruments.

Arm, Jazelle, Thumb, Cortex, and AMBA are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Arm9, CoreSight, and Neon are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Bluetooth is a registered trademark of Bluetooth SIG, Inc. and is licensed to Texas Instruments.

HDQ is a trademark of Benchmarq.

1-Wire is a registered trademark of Dallas Semiconductor.

Windows, Direct3D, DirectDraw, and DirectX are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries.

USSE and PowerVR are trademarks or registered trademarks of Imagination Technologies Limited.

OpenVG and OpenWF are trademarks of Khronos Group, Inc.

Arteris is a trademark of Arteris, Inc.

RealVideo is a registered trademark of RealNetworks, Inc.

Secure Digital and SD are registered trademarks of SD Card Association.

MMC and eMMC are trademarks of MultiMediaCard Association.

SonicsLX, Sonics3220 are trademarks or registered trademarks of Sonics, Inc.

HD Radio is a trademark of iBiquity Digital Corporation.

Linux is a registered trademark of Linus Torvalds.

On2 is property of Google, Inc.

MediaLB is a register trademark of Microchip Technology Inc.

MIPI is a registered trademark of the MIPI Alliance, Inc.

OneNAND is a trademark of SAMSUNG Electronics, Corporation.

All other trademarks are the property of their respective owners.

Chapter 1
Introduction

This chapter introduces the features, subsystems, and architecture of the DRA75x (Jacinto 6 Ex and Jacinto 6 EP) and DRA74x (Jacinto 6) families of high-performance infotainment processors.

1.1 DRA75x, DRA74x Overview 257
1.2 DRA75x, DRA74x Environment 259
1.3 DRA75x, DRA74x Description 260
1.4 DRA75x, DRA74x Family 268
1.5 DRA75x, DRA74x Device Identification 268
1.6 DRA75x, DRA74x Package Characteristics Overview 270

1.1 DRA75x, DRA74x Overview

The DRA75x, DRA74x is a high-performance, infotainment application device, based on enhanced OMAP™ architecture integrated on a 28-nm technology.

- The architecture is designed for advanced graphical HMI and Navigation, Digital and Analog Radio, Rear Seat Entertainment and Multimedia playback, providing Advanced Driver Assistance integration capabilities with Video analytics support, and best-in-class CPU performance, video, image, and graphics processing sufficient to support, among others:
  - Streaming video up to full high definition (Full-HD) (1920×1080p, 60 fps)
  - 2-dimensional (2D) and 3-dimensional (3D) graphics and composition
  - Decode of digital radio standards (DAB, HD Radio™), and analog AM/FM/RDS radio
  - Efficient web browsing

Note

The supported set of features and peripherals is device part number dependent. Refer to device-specific Data Manual, for more information.

Note

TI limits support for this family of SoCs to features that are supported via Software Development Kits (SDK). The SDK “build sheet” is available for download as part of each SDK and should be referenced to understand the subset of SoC hardware functionality that is available in software:

https://www.ti.com/tool/PROCESSOR-SDK-TDAX

- The device is composed of the following subsystems:
  - Cortex®-A15 microprocessor unit (MPU) subsystem, including two ARM® Cortex-A15 cores
  - Two Digital Signal Processor (DSP) C66x subsystems
  - Image and video accelerator high-definition (IVA-HD) subsystem
  - Two Cortex®-M4 image processing unit (IPU) subsystems, each including two ARM Cortex-M4 microprocessors available for general purpose usage
  - Two Embedded Vision Engine (EVE) subsystems
  - Display subsystem (DSS)
  - Video Processing subsystem (VPE)
  - Video Input Capture (VIP)
  - 3D-graphics processing unit (GPU) subsystem, including POWERVR™ SGX544 dual-core
  - 2D-graphics accelerator (BB2D) subsystem, including Vivante™ GC320 core
  - Three pulse-width modulation (PWM) subsystems
  - Real-time clock (RTC) subsystem
  - Debug subsystem
- The device provides a rich set of connectivity peripherals, including among others:
  - One USB3.0 and three USB2.0 subsystems
  - SATA 2 subsystem
  - Two PCI Express Gen2 subsystems
  - 3-port Gigabit Ethernet Switch subsystem
  - Two Controller Area Network (DCAN) subsystems
- The device includes support for:
  - Error Detection and Correction:
    - Parity bit per byte on C66x DSP L1 program cache and Single-Error Correction Dual-Error Detection (SECDED) on L2 memories on the DSP
    - SECDED on Large L3 memory

    - EVE: Error detection for all internal data memories (DMEM, WBUF, IBUFLA, IBUFLB, IBUFHA, IBUFHB):
      - Single bit error detection (parity bit per byte) on DMEM, WBUF and IBUFs
      - Double bit error detection (distance 3, 10-bit hamming code) and parity on VCOP accesses to memory and working/image buffers in EVE
    - SECDED on external DDR memory interface (EMIF1 only)
  - MMU/MPU
    - MMU used for key masters (Cortex-A15 MPU, Cortex-M4 IPU, C66x DSP, EVE, EDMA)
    - Memory protection of C66x cores
    - MMU inside the Dynamic Memory Manager

- The device includes state-of-the-art integrated power-management techniques required for high-performance infotainment products.
- The device also integrates:
  - On-chip memory
  - External memory interfaces
  - Memory management
  - Level 3 (L3) and level 4 (L4) interconnects
  - System peripherals
  - Car, audio and media peripherals
  - Radio accelerators

1.2 DRA75x, DRA74x Environment

This section provides an overview of the DRA75x, DRA74x environment.

Figure 1-1 is an example of a non-exhaustive environment for the DRA75x, DRA74x device.

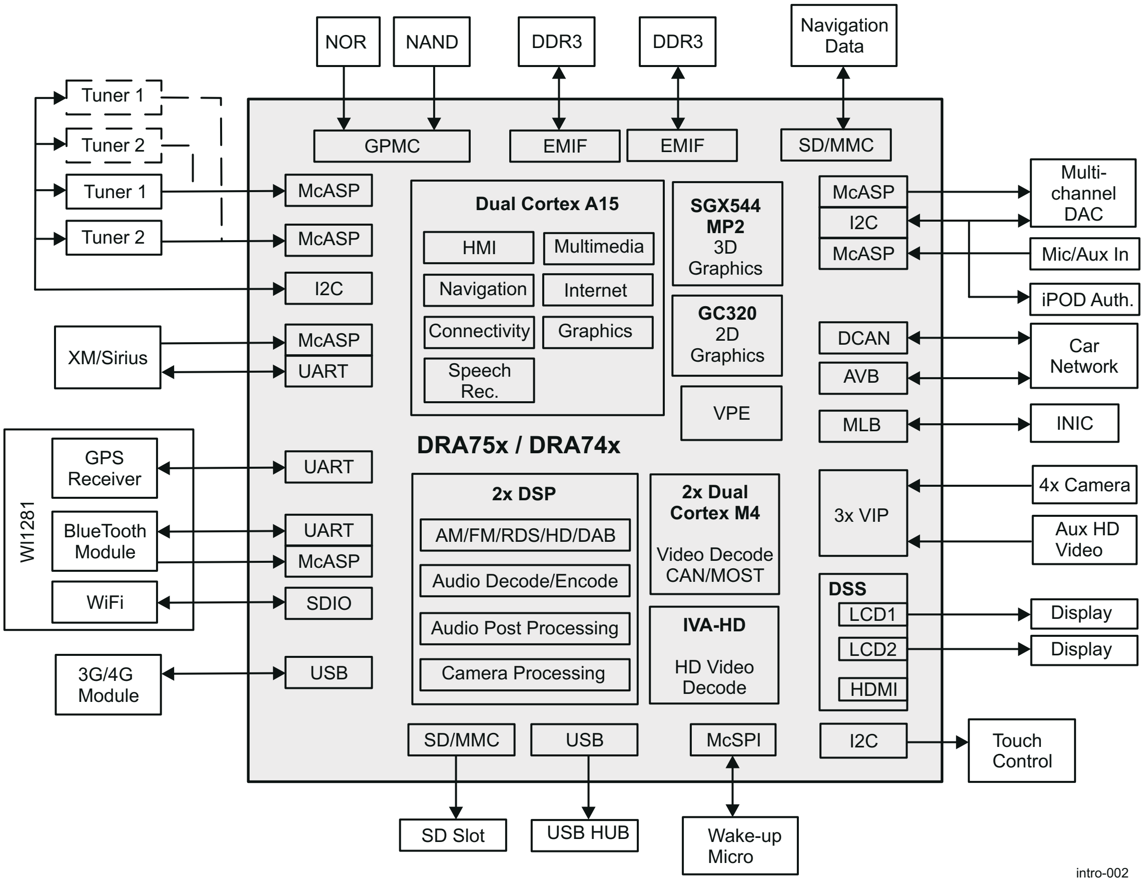


Figure 1-1. DRA75x, DRA74x Sample Environment Diagram

Note

The supported set of features and peripherals is device part number dependent. Refer to device-specific Data Manual, for more information.

1.3 DRA75x, DRA74x Description

The DRA75x, DRA74x device is offered in a 760-ball, 23×23-mm, 0.8-mm ball pitch with Via Channel™ Array (VCA) technology, ball grid array (BGA) package.

Figure 1-2 is the block diagram of the DRA75x, DRA74x device.

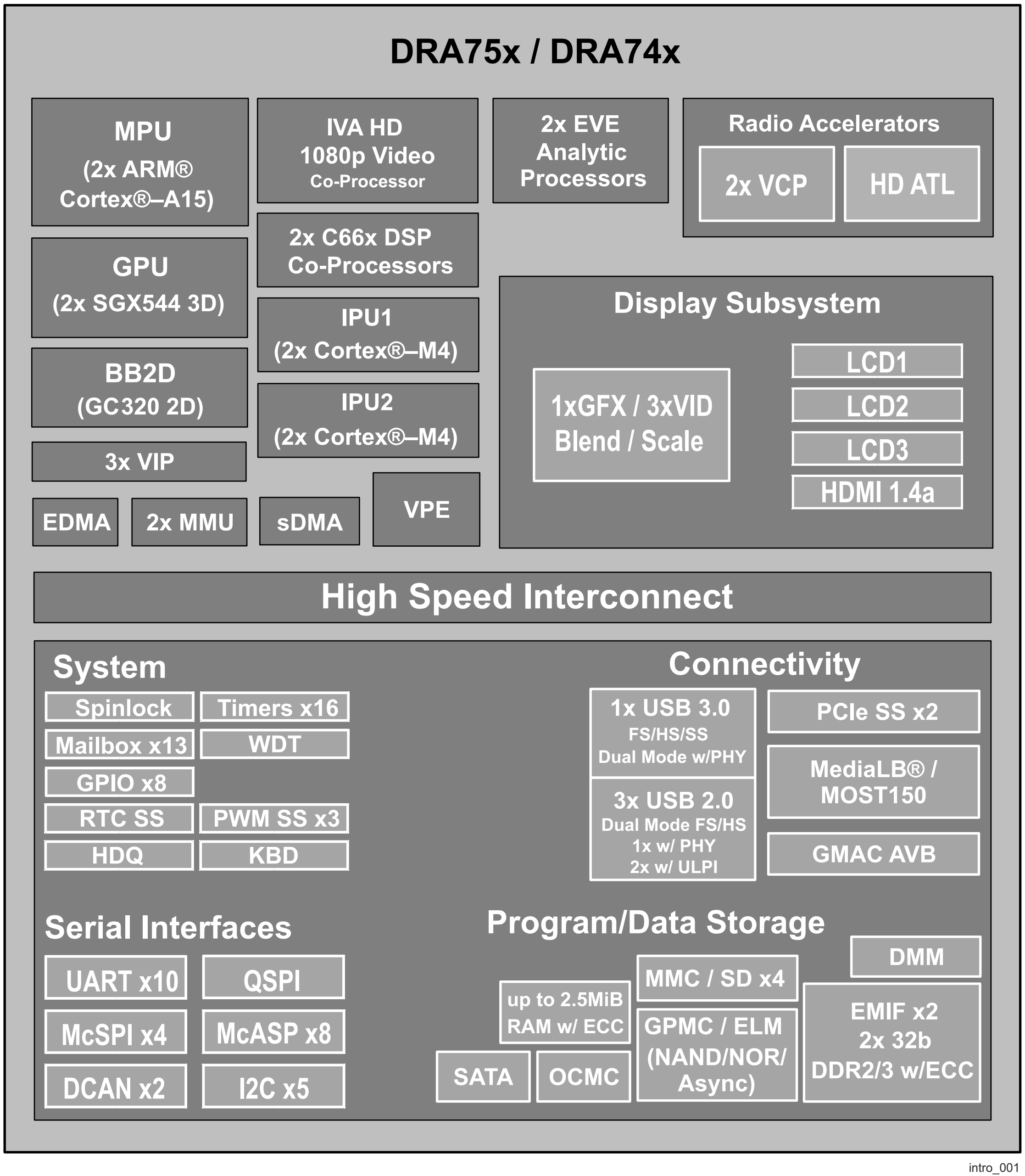


Figure 1-2. DRA75x, DRA74x Block Diagram

Note

The supported set of features and peripherals is device part number dependent. Refer to device-specific Data Manual, for more information.

1.3.1 MPU Subsystem

The Cortex-A15 MPU subsystem integrates the following submodules:

- ARM Cortex-A15 MPCore
  - Two central processing units (CPUs)
  - ARM Version 7 ISA: Standard ARM instruction set plus Thumb®-2, Jazelle® RCT Java™ accelerator, hardware virtualization support, and large physical address extensions (LPAE)
  - Neon™ SIMD coprocessor and VFPv4 per CPU
  - Interrupt controller with up to 160 interrupt requests
  - One general-purpose timer and one watchdog timer per CPU
  - Debug and trace features
  - 32-KiB instruction and 32-KiB data level 1 (L1) cache per CPU
- Shared 2-MiB level 2 (L2) cache
- 48-KiB bootable ROM
- Local power, reset, and clock management (PRCM) module
- Emulation features
- Digital phase-locked loop (DPLL)

1.3.2 DSP Subsystems

There are two DSP subsystems in the device. Each DSP subsystem contains the following submodules:

- TMS320C66x™ VLIW DSP core for audio processing, and general-purpose imaging and video processing. It extends the performance of existing C64x+™ and C647x™ DSPs through enhancements and new features.
  - 32-KiB L1D cache , and 32-KiB L1P cache or addressable SRAM
  - 288-KiB L2 cache
    - 256-KiB configurable as cache or SRAM
    - 32-KiB SRAM
- Enhanced direct memory access (EDMA) engine for video and audio data transfer
- Memory management units (MMU) for address management.
- Interrupt controller (INTC)
- Emulation capabilities

1.3.3 EVE Subsystems

The Embedded Vision Engine (EVE) module is a programmable imaging and vision processing engine, intended to be used in devices that serve consumer electronics imaging and vision applications. Its programmability allows late in-development or post-silicon processing requirements to be met by addition of differentiating features in imaging and vision products.

There are two EVE subsystems in the device, each consisting of:

- ARP32 scalar core, with 32-KiB direct mapped program cache, and 32-KiB data memory
- VCOP vector core, with 16 processing elements, 32-KiB working buffer, and four 16-KiB image buffers
- Internal Mailboxes for EVE to EVE, and EVE to MPU/IPU/DSP communication
- Integrated EDMA controller and memory management units

1.3.4 IPU Subsystems

There are two Cortex-M4 IPU subsystems in the device available for general purpose usage.

Each IPU subsystem includes the following components:

- Two Cortex-M4 CPUs
- ARMv7E-M and Thumb-2 instruction set architectures
- Hardware division and single-cycle multiplication acceleration
- Dedicated INTC with up to 63 physical interrupt events with 16-level priority
- Two-level memory subsystem hierarchy
  - L1

    32-KiB shared cache memory

  - L2 ROM + RAM
    - 64-KiB RAM
    - 16-KiB bootable ROM
- MMU for address translation
- Integrated power management
- Emulation feature embedded in the Cortex-M4

1.3.5 IVA-HD Subsystem

The IVA-HD subsystem is a set of video encoder and decoder hardware accelerators.

The list of supported codecs can be found in the software development kit (SDK) documentation.

1.3.6 Display Subsystem

The display subsystem provides the control signals required to interface the device system memory frame buffer (SDRAM) directly to the displays. It supports hardware cursor, independent gamma curve on all interfaces, multiple-buffer, and programmable color phase rotation. The display subsystem allows low-power display refresh and arbitration between normal and low-priority pipelines.

The display subsystem consists of the following components:

- **Display controller:** Reads and displays the encoded pixel data stored in memory and writes the output of one of the overlays or one of the pipelines into the system memory. The display controller supports the following components:
  - Three video pipelines, one graphic pipeline, and one write-back pipeline. The graphic pipeline supports pixel formats such as: ARGB16-4444, RGB16-565, ARGB16-1555, ARGB32-8888, RGBA32-8888, RGB24-888. It allows selection of the color-depth expansion.
  - Write-back pipeline: Uses poly-phase filtering for independent horizontal and vertical resampling (upsampling and downsampling). It allows programmable color space conversion of RGB24 into YUV4:2:2-UYVY, YUV4:2:2-YUV2, or YUV4:2:0-NV12 or NV21, and selection of color-depth reduction from RGB24 to RGB16.
  - Three LCD outputs, each one with dedicated overlay manager, for support of active matrix color displays (up to 24-bit interface). Maximum listed resolutions are not supported concurrently on all outputs.
    - First main LCD output delivered on MIPI® DPI 1.0 LCD pixel interface, supporting up to WUXGA (1920 x 1200) with reduced blanking periods.
    - Second and third LCD outputs delivered on MIPI DPI 2.0 LCD pixel interfaces, supporting up to WUXGA (1920 x 1200) with reduced blanking periods.
  - One TV output with dedicated overlay manager to support HDMI v1.4a interface (1080p @ 60 fps video and multichannel audio)
  - Own direct memory access (DMA) engine
- **High-definition multimedia interface (HDMI) encoder** with the following main features:
  - HDMI 1.4a and DVI 1.0 compliant

1.3.7 Video Processing Subsystem

The video processing engine (VPE) module provides support for the following memory-to-memory operations:

- Reads of raster or tiled YUV420 coplanar, YUV422 coplanar, or YUV422 interleaved video
- Deinterlacing up to two 1080i video streams
- Scaling up to 1080p (1920x1080 resolution) of the input video
- Chroma up- and downsampling
- VC-1 Range Mapping and Range Reduction
- Color space conversion
- Writes of the resulting video in YUV420 coplanar (raster or tiled), YUV422 coplanar (raster or tiled), YUV422 interleaved (raster or tiled), YUV444 single plane (raster only), or RGB888 (raster only).

1.3.8 Video Capture

There are three Video Input Port (VIP) modules in the device, providing video capture functions.

VIP1 and VIP2 modules support each up to:
- Two separate 24-bit video ports for parallel RGB/YUV/RAW (or BT656/1120) data, up to 165 MHz
- Two separate 8-bit video ports for YUV/RAW (or BT656) data, up to 165 MHz

VIP3 module supports up to two separate 16-bit video ports for parallel RGB/YUV/RAW (or BT656/1120) data, up to 165MHz.

Each VIP module supports:
- Embedded Sync (multiplexed sources) and Discrete Sync (single source) data interface modes
- Color space conversion and scaling:
  – Up to 2047 pixels wide input with scaling
  – Up to 3840 pixels wide input - when chroma up/down sampling, without scaling
  – Up to 4095 pixels wide input - without scaling and chroma up/down sampling
  – Maximum supported input resolution is further limited by:
    - Pixel clock and feature-dependent constraints
    - For RGB24-bit format (RAW data), the maximum frame width is limited to 2730 pixels
- Embedded DMA engine, supporting tiled (2D) and raster addressing

1.3.9 3D GPU Subsystem

The 3D graphics processing unit (GPU) subsystem is based on POWERVR® SGX544 subsystem from Imagination Technologies. It supports phone, PDA, and handheld gaming applications. The GPU can process different data types simultaneously, such as: pixel data, vertex data, video data, and general-purpose data.

The GPU subsystem has the following features:

- Multicore GPU architecture: two SGX544 cores. Shared system level cache of 128 KiB
- Tile-based deferred rendering architecture
- Second-generation universal scalable shader engines (USSE2), multithreaded engines incorporating pixel and vertex shader functionality
- Present and texture load accelerators
  – Enables to move, rotate, twiddle, and scale texture surfaces.
  – Supports RGB, ARGB, YUV422, and YUV420 surface formats.
  – Supports bilinear upscale.
  – Supports source colorkey.

- Industry-standard API supports DirectX® 9, OpenVG™ 1.1, and OpenCL™1.1 Embedded Profile
- Fine-grained task switching, load balancing, and power management
- Programmable high-quality image antialiasing
- Bilinear, trilinear, anisotropic texture filtering
- Advanced geometry DMA driven operation for minimum CPU interaction
- Fully virtualized memory addressing for OS operation in a unified memory architecture (MMU)

1.3.10 BB2D Subsystem

The 2D BitBlt (BB2D) graphics accelerator subsystem is based on the GC320 core from Vivante Corporation and has the following features:

- API support:
  – OpenWF™, DirectFB
  – GDI/DirectDraw
  – Adobe® Flash®
- BB2D architecture:

  - BitBlt and StretchBlt
  - DirectFB hardware acceleration
  - ROP2, ROP3, ROP4 full alpha blending and transparency
  - Clipping rectangle support
  - Alpha blending includes Java 2 Porter-Duff compositing rules
  - 90-, 180-, 270-degree rotation on every primitive
  - YUV-to-RGB color space conversion
  - Programmable display format conversion with 14 source and 7 destination formats
  - High-quality, 9-tap, 32-phase filter for image and video scaling at 1080p
  - Monochrome expansion for text rendering
  - 32 K × 32 K coordinate system

1.3.11 On-Chip Debug Support

The on-chip debug support has the following features:

- Multiprocessor debugging lets users control multiple CPU cores embedded in the device, such as:
  - Global starting and stopping of individual or multiple processors
  - Each processor can generate triggers that can be used to alter the execution flow of other processors
  - System clocking and power down
  - Interconnection of multiple devices
  - Channel triggering
- Target debugging, using IEEE1149.1 (JTAG®)
- Reduction of power consumption in normal operating mode

The debug subsystem includes:

- Generic TAP for emulation and test control (ICEPick-D™)
- Debug access port (DAP)
- Embedded Trace Macro (ETM)
- Trace Port Interface Unit (TPIU)
- Embedded Trace Buffer (ETM)
- Emulation Pin Manager (EPM)
- Cross triggering (XTRIG)

The debug subsystem provides also:

- ICEMelter, for controlling the wake-up and power-down of the emulation power domain
- L3_INSTR CORE instrumentation interconnect
- OCP watch-point (OCP-WP), for monitoring L3 interconnect transaction when target transaction attributes match the user-defined attributes or trigger on external debug event
- Power-management events profiler (PM instrumentation)
- Clock-management events profiler (CM instrumentation)
- Statistics collector (performance probes)

1.3.12 Power, Reset, and Clock Management

The PRCM module allows efficient control of clocks and power according to the required performance, and reduction of power consumption.

The PRCM module is divided into:

- Power and reset management (PRM) with the following features:
  - Dynamic clock gating
  - Dynamic voltage and frequency scaling (DVFS)
  - Dynamic power switching (DPS)
  - Static leakage management (SLM)
- Clock management (CM) for clock generation and distribution, allowing reduction of dynamic consumption.

1.3.13 On-Chip Memory

- The device can include up to three instantiations of an On-Chip Memory Controller (OCMC) with associated RAM with ECC, with total size up to 2.5 MiB. OCMC_RAM2 (1MiB) and OCMC_RAM3 (1MiB) are not present on DRA74x devices, but are included in some of the DRA75x devices. For details, see the device data manual.
- Circular buffer feature for each OCMC RAM (8-MiB virtual address space required) allowing on-the-fly processing of VIP data by EVE
- Save and Restore Memory / Scratch Pad in the wake-up domain

1.3.14 Memory Management

The memory management is performed from:

- System DMA controller with up to 128 hardware requests, 32 prioritizable logical channels, and 256 × 64-bit FIFO dynamically allocable between active channels.
- Enhanced DMA controller supporting two simultaneous read and two simultaneous write physical channels, and up to 64 programmable logical channels.
- Dynamic memory management (DMM) module, which performs global address translation, address rotation (tiling), and access interleaving between the two EMIF channels.
- Two memory management units (MMU), with 4KiB, 64KiB, 1MiB, 16MiB programmable page sizes, and 32 entries TLB.
  - MMU1 dedicated to EDMA
  - MMU2 dedicated to PCIe_SS1 and PCIe_SS2

1.3.15 External Memory Interfaces

- Two 32-bit DDR2 / DDR3 EMIF controllers, each of which with the following features:
  - Dual-port controller for efficient memory sharing between applications
  - 32-bit data path, one chip-select per memory controller
  - Memory density up to 2 GiB supported over one chip-select providing a total SDRAM space of 4 GiB addressable by the MPU extended address range.
  - EMIF1 controller supports Single bit Error Correction and Dual Error Detection (SECDED)
    - SECDED supported in both 32-bit and 16-bit/Narrow mode
    - Programmable address ranges to define SECDED protected region
    - Parity bits calculated and stored on all writes to SECDED protected address region
    - Parity bits verified on all reads from SECDED protected address region
    - Statistics for 1-bit and 2-bit errors
- General-purpose memory controller (GPMC) supporting connection with:
  - Asynchronous SRAM memories
  - Asynchronous and synchronous NOR flash memories
  - NAND flash memories, with up to 16-bit ECC via the Error Location Module (ELM)
  - Pseudo-SRAM devices
- Quad SPI module, supporting 1 to 4 address bytes for SPI NOR flash, and up to 66-MHz single data rate

1.3.16 System and Connectivity Peripherals

The device supports a comprehensive set of peripherals to provide flexible and high-speed interfacing and on-chip programming resources.

1.3.16.1 System Peripherals

- Sixteen general-purpose timers (two timer modules supporting 1-ms tick generation)
- One watchdog timer (WDT)
- One 32-kHz synchronization timer
- System control module, which contains registers for the following functions:
  - Static device configuration
  - Debug and observability
  - Status

  - Pad configuration
  - I/O configuration
  - eFuse logic
  - Analog function control
  - System boot decoding logic
- Thirteen system Mailboxes for communication between MPU, DSP and IPU subsystems.
- SpinLock module for hardware semaphore between the MPU, DSP, and IPU subsystems
- Inter-processor Communication Register
- Three Pulse Width Modulation Subsystems (PWMSS), each containing Enhanced High Resolution Pulse Width Modulator (eHRPWM), Enhanced Capture (eCAP), and Enhanced Quadrature Encoded Pulse (eQEP) modules.
- Real-Time-Clock Subsystem (RTCSS), supporting four external wake-up inputs and one power enable output, all of which are 3.3- or 1.8-V multivoltage I/Os.
- Eight general-purpose input/output (GPIO) modules with 32 I/Os each. One GPIO module supporting wake-up request generation.
- HDQ/ 1-Wire® – HDQ and Maxim Integrated Products 1-Wire protocols interface
- Keyboard controller, supporting up to 9 × 9 matrix keypads

1.3.16.2 Media Connectivity Peripherals

- Four HS-MMC/SD/SDIO modules:
  - Two modules acting as HS-MMC/SD initiator controllers, supporting JEDEC JESD84 v4.5-A441 and SD3.0 physical layer with SDA3.00 standards
    - One controller with 8-bit interface for JESD84 memories with dual voltage IOs (1.8 or 3.3 V). Another controller with 4-bit interface for external card support with embedded dual voltage I/Os (1.8 or 3 V) and supporting UHS-I rates.
    - Each controller including its DMA controller compliant to ADMA2 (SDA3.00 Part A2 DMA controller)
  - Two modules acting as SDIO interface controllers. One controller supporting 4-bit data bus width. Another controller supporting up to 8-bit data bus width.
- One SuperSpeed Universal Serial Bus (USB) Dual-Role-Device (DRD) subsystem with embedded HS and SS PHYs, compatible with the USB2.0 (up to 480 Mbps) and USB3.0 (5 Gbps) standards
- Three High Speed USB subsystems:
  - One HS USB subsystem with embedded HS PHY, supporting up to 480 Mbps.
  - Two HS USB subsystems, each with ULPI (SDR) interface to external HS PHY, supporting up to 480 Mbps.
- One SATA subsystem, providing interface for solid-state drive (SSD) or hard-disk drive (HDD) mass storage. Supports one HBA port with SATA-2 generation speed of 3 Gbps.

1.3.16.3 Car Connectivity Peripherals

- One Media Local Bus (MLB) subsystem for connection to MOST® optical ring. The module supports both 3-pin (up to MOST50, 1024×Fs) and 6-pin (up to MOST150, 2048 ×Fs) versions of MediaLB® Physical Layer Specification v4.0, and all types of transfer (Synchronous, Isochronous, Asynchronous/Packet, Control) over 64 logical channels.
- One 3-port Gigabit Ethernet Switch subsystem (10, 100, or 1000 Mbps). The switch provides two external Ethernet ports and one internal CPPI interface port with AVB/Industrial Ethernet and 802.1ae support. Included support for 3.3-V RMII/MII and 1.8- or 3.3-V RGMII.
- Two DCAN controllers, supporting bitrates up to 1 Mbit/s and compliant to the Controller Area Network (CAN) 2.0B protocol specification.
- Two PCI Express subsystems, one providing Gen2 compliant 2-lane port, and the other providing Gen2 compliant 1-lane port, up to 5.0 Gbps per lane. Both PCIe subsystems provide support for either Root Complex or Endpoint. The two PCIe subsystems share common 2-lane PCIe PHY, configurable to operate either as 2-lane to one controller (PCIe_SS1) or two separate lanes to two controllers (PCIe_SS1 and PCIe_SS2).

1.3.16.4 Audio Connectivity Peripherals

- Eight multichannel audio serial ports (McASP):
  - Two McASP supporting up to 16 channels each and independent TX/RX clock/sync domains
  - Six McASP supporting up to 4 channels each and unified clock/sync domain
  - Features list of the McASP include:
    - Independent transmit and receive modules, each including programmable clock and frame sync generator, TDM streams from 2 to 32, support for time slot sizes up to 32 bits, data formatter for bit manipulation
    - Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components.
    - Wide variety of I2S and similar bit-stream formats
    - Integrated digital audio interface transmitter (DIT) supporting S/PDIF, IEC60958-1, AES-3 formats, and enhanced channel status/user data RAM
    - 384-slot TDM with external digital audio interface receiver (DIR) device
    - Extensive error checking and recovery

1.3.16.5 Serial Control Peripherals

- Ten universal asynchronous receiver/transmitter (UART) modules as serial-communication interfaces, 16C750 compatible.
  - One UART module supporting extended modem control signals
  - One UART module with IrDA features
- Four general purpose multichannel serial peripheral interface (McSPI) modules
- Five multimaster HS I^2C controller modules, compliant with Philips I^2C specification version 2.1.
  - I2C1 and I2C2 controllers suppot Fast-mode, with rates up to 400 Kbps
  - I2C3, I2C4 and I2C5 controllers support High-speed mode, with rates up to 3.4 Mbps

1.3.16.6 Radio Accelerators

- One Audio Tracking Logic (ATL) module, containing four ATL instances, for HD-Radio support and asynchronous sample rate conversion assistance. Each ATL instance supports error tracking between two reference signals, and generation of modulated clock (using software controlled cycle stealing)
- Two Viterbi Co-Processor (VCP) modules, providing acceleration of digital radio processing (decoding of convolutional encoded data).

1.4 DRA75x, DRA74x Family

The DRA75x, DRA74x family is composed of the following:
- DRA75x (Jacinto6 Ex family of devices):
  - DRA756
  - DRA755
  - DRA754
- DRA75x (Jacinto6 EP family of devices):
  - DRA752
  - DRA751
  - DRA750
- DRA74x (Jacinto6 family of devices):
  - DRA746
  - DRA745
  - DRA744

The supported set of features and peripherals is device part number dependent. Refer to device Data Manual, for more information.

1.5 DRA75x, DRA74x Device Identification

Table 1-1 describes the identification registers.

The identification registers include the data registers listed in Table 1-2 and Table 1-4. These registers are read-only accessed ports that are programmed into eFuses FARM FROM.

Table 1-1. Device Identification Register Fields

Register Field	Alias Name	Physical Address	Address Offset
CTRL_CORE_STATUS[8:6] DEVICE_TYPE	DEVICE_TYPE	0x4A00 2134	0x134
CTRL_WKUP_STD_FUSE_DIE_ID_0[31:0] STD_FUSE_DIE_ID_0	DIE_ID[31:0]	0x4AE0 C200	0x200
CTRL_WKUP_ID_CODE[31:0] STD_FUSE_IDCODE	ID_CODE	0x4AE0 C204	0x204
CTRL_WKUP_STD_FUSE_DIE_ID_1[31:0] STD_FUSE_DIE_ID_1	DIE_ID[63:32]	0x4AE0 C208	0x208
CTRL_WKUP_STD_FUSE_DIE_ID_2[31:0] STD_FUSE_DIE_ID_2	DIE_ID[95:64]	0x4AE0 C20C	0x20C
CTRL_WKUP_STD_FUSE_DIE_ID_3[31:0] STD_FUSE_DIE_ID_3	DIE_ID[127:96]	0x4AE0 C210	0x210
CTRL_WKUP_STD_FUSE_PROD_ID_0[31:0] STD_FUSE_PROD_ID	PROD_ID	0x4AE0 C214	0x214

Table 1-2. DIE_ID

Register Field	Alias Name	Value
CTRL_WKUP_STD_FUSE_DIE_ID_0[31:0] STD_FUSE_DIE_ID_0	DIE_ID[31:0]	This register is for internal-use only.
CTRL_WKUP_STD_FUSE_DIE_ID_1[31:0] STD_FUSE_DIE_ID_1	DIE_ID[63:32]	This register is for internal-use only.
CTRL_WKUP_STD_FUSE_DIE_ID_2[31:0] STD_FUSE_DIE_ID_2	DIE_ID[95:64]	Part number identifier. See *DRA75x, DRA74x Part Number Identifier* for more information.
CTRL_WKUP_STD_FUSE_DIE_ID_3[31:0] STD_FUSE_DIE_ID_3	DIE_ID[127:96]	Reserved

The part number identification data can be read in the [31:0] STD_FUSE_DIE_ID_2 bit-field of the CTRL_WKUP_STD_FUSE_DIE_ID_2 register. See Table 1-3 for more information.

Table 1-3. DRA75x, DRA74x Part Number Identifier

CTRL_WKUP_STD_FUSE_DIE_ID_2 [31:0] STD_FUSE_DIE_ID_2	Value and Description	Comment
[31:24] Base PN	Refer to the Device Comparison section of a device-specific Data Manual (DM) for the Base PN value of a given part number.	Refer to device DM for details on the supported features for a given part number.
[23:19] Speed	0 (0x00) = A speed designator 1 (0x01) = B speed designator ... 24 (0x18) = Y speed designator 25 (0x19) = Z speed designator	Refer to device DM for supported speed grades for a given device, and the definition for supported speed grades.
[18] Temperature	0 = Reserved 1 = Automotive, -40°C to 125°C	Junction temperature.
[17:16] Package	2 = ABC 23×23 Solder Flip Chip Others = Reserved	Refer to device DM for details on packaging.
[15:0] Reserved	Reserved	Reserved

The product type can be read in the value of the RAMP_SYSTEM bit field of the IC_CODE register (see Table 1-4). The silicon revision can be read in the value of the VERSION bit field of the ID_CODE register (see Table 1-4).

Table 1-4. ID_CODE

Register Field	Value	Comment
CTRL_WKUP_ID_CODE[31:28] VERSION	See Table 1-5	Revision number
CTRL_WKUP_ID_CODE[27:12] RAMP_SYSTEM	See Table 1-5	Ramp system number
CTRL_WKUP_ID_CODE[11:1] TI_IDM	0x17	Manufacturer identity (TI)
CTRL_WKUP_ID_CODE[0] ONE	0x1	Always set to 1

Table 1-5, *DRA75x, DRA74x ID_CODE Values* lists the ramp system and revision number values.

Table 1-5. DRA75x, DRA74x ID_CODE Values

Silicon Type	VERSION	RAMP_SYSTEM	ID_CODE
DRA75x / DRA74x SR1.0	0x0	0xB990	0x0B99002F
DRA75x / DRA74x SR1.1	0x1	0xB990	0x1B99002F
DRA75x / DRA74x SR2.0	0x2	0xB990	0x2B99002F

The device type can be read in the PROD_ID register (see Table 1-6).

Table 1-6. PROD_ID

Register Field	Value	Comment
CTRL_WKUP_STD_FUSE_PROD_ID_0 [7:0] DEVICE_TYPE	0xF0	Reads 0xF0 when device is a general-purpose (GP) device

The device type can be read also in the CTRL_CORE_STATUS register (see Table 1-7).

Table 1-7. DEVICE_TYPE

Register Field	Value	Comment
CTRL_CORE_STATUS [8:6] DEVICE_TYPE	0x3	Reads 0x3 when device is a general-purpose (GP) device

1.6 DRA75x, DRA74x Package Characteristics Overview

The DRA75x, DRA74x die will be offered with the following characteristics:

- Body: 23 × 23mm
- Technology: Ball Grid Array (BGA) package
- Ball pitch: 0.8-mm ball pitch with Via Channel® Array (VCA) technology
- Pattern: partial grid
- Pins: 760 total device pins

For more information refer to device Data Manual.

Chapter 2

Memory Mapping

This chapter describes the memory mapping in the device.

2.1 Introduction 272
2.2 L3_MAIN Memory Map 273
2.3 L4 Memory Map 278
2.4 L4_PER Memory Map 283
2.5 MPU Memory Map 290
2.6 IPU Memory Map 292
2.7 DSP Memory Map 294
2.8 EVE Memory Map 295
2.9 TILER View Memory Map 296

2.1 Introduction

The microprocessor unit (MPU) has a 32-bit address port, which allows it to handle a 4-GiB space divided into several regions, depending on the target type.

The memory map has the following features that are shared among the initiators, such as the MPU subsystem:

- Memory space: General-purpose memory controller (GPMC)
- Dynamic memory management (DMM) controller
- Register spaces: Level 3 (L3) and level 4 (L4) interconnects
- Dedicated spaces:EVE/IPU/DSP subsystem.

The GPMC and DMM are dedicated to memory connection. The GPMC is used for NOR and NAND flash and static random access memories (SRAMs). The DMM is used for synchronous dynamic random access memories (SDRAMs), such as DDR. For more information, see Section 15.2, *Dynamic Memory Manager*, and Section 15.3, *EMIF Controller*.

The L3 interconnect allows the sharing of resources, such as peripherals and external or on-chip memories, among all the initiators of the platform. The L4 interconnects control access to the peripherals.

Transfers across the platform between initiators and targets are physically conditioned by the chip interconnect and can be logically conditioned by firewalls. For more information about the intercommunication (L3 and L4 interconnects) and protection mechanisms implemented in the device, see Section 14.2, *L3 Interconnect*, and Section 14.3, *L4 Interconnect*.

Figure 2-1 shows the interconnect of the device and the main modules and subsystems in the platform.

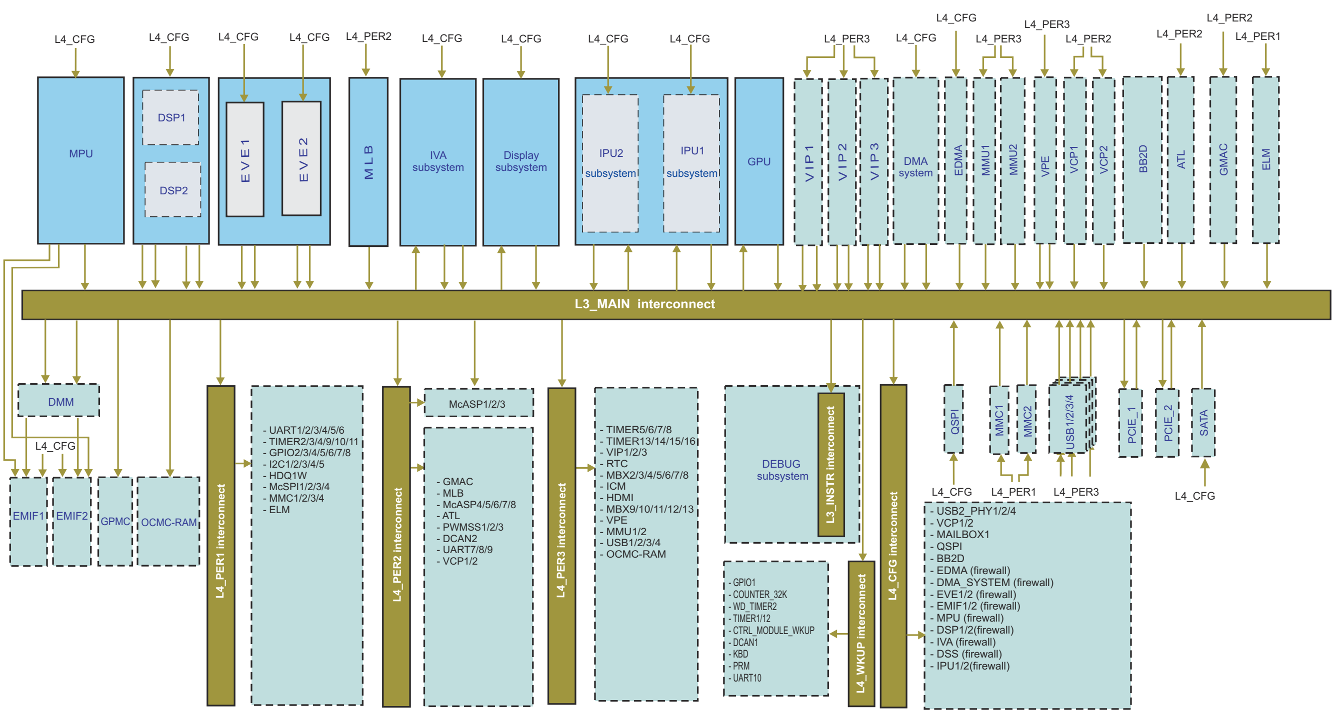


Figure 2-1. Interconnect Overview

2.2 L3_MAIN Memory Map

The memory space system is hierarchical: level 1 (L1), level 2 (L2), L3_MAIN, and L4. L1 and L2 are memories in the MPU, IPU, and digital signal processor (DSP) subsystems. L3_MAIN handles many types of data transfers, including data exchange with system on-chip/external memories. The chip-level interconnect, which consists of one L3_MAIN and five L4s, enables communication among all modules and subsystems.

This section provides a global view of the memory mapping of the device at the L3_MAIN interconnect and describes the boot, GPMC, and SDRAM controller (SDRC) (EMIF/DMM) spaces.

The system memory mapping is flexible, with two levels of granularity for target address space allocation:

- L1: The four quarters are labeled Q0, Q1, Q2, and Q3. Each quarter corresponds to a 1-GiB address space (the total low-address space is 4 GiB, 32-bit). The CPU extended address range is labeled as high memory (Q8 – Q15) and provides a total of 8 GiB.
- L2: Each quarter is divided into eight blocks of 32 MiB, with target spaces mapped in the blocks.

This organization allows the decoding of all target spaces based on the 7 most-significant bits (MSBs) of the 32-bit address ([31:25]).

- Boot space:

  When booting from the on-chip ROM with the appropriate external sys_boot pin configuration, the lowest 1-MiB memory space [0x0000 0000–0x000F FFFF] is redirected to the on-chip boot ROM address space [0x4000 0000–0x400F FFFF].

  When booting from the GPMC, the memory space is part of the GPMC address space. At reset, the 0x0000 0000 address is available on chip-select 0 (CS0) for a memory size of 16 MiB.

  For more information about the sys_boot pins configuration, see Section 15.4, *General-Purpose Memory Controller*, and Chapter 32, *Initialization*.
- GPMC space:

  Eight independent GPMC chip-selects (CS0 to CS7) are available in the first quarter (Q0) of the addressing space to access NOR/NAND flash and SRAM. The chip-selects have a programmable start address and programmable size (up to 128 MiB) in a total memory space of (Q0) 1GiB, but limited now to 512 MiB.
- EMIF1/EMIF2 CS0 space:

  Q2 addressing space is interleaved on two DDR-memory controllers (EMIF1 and EMIF2), each activating its CS0 line. These chip-selects can be programmed to 64, 128, 256, 512, 1024, and 2048 MiB. Interleaving occurs at 128-byte granularity.

  The EMIF1-CS0 base address is always 0x8000 0000 at reset, and occupies a 1-GiB address space at reset (interleaving is disabled at reset).

  Q3 addressing space is interleaved on two SDRAM controllers (EMIF1 and EMIF2), each activating its CS0 line. These chip-selects can be programmed to 64, 128, 256, 512, and 1024 MiB. Interleaving occurs at 128-byte granularity.

  EMIF1-CS0 and EMIF2-CS0 in Q3 space are disabled at reset. Their base address is programmable to achieve a continuous address space with the respective CS0 in Q2 space, regardless of the address range programmed.
- TILER space:

  Q3 addressing space is also used to access the TILER system. This space is visible only for the Display Subsystem (DSS) . See Table 2-12.
- 8 GiB of SDRAM virtualization (only 4 GiB are physically available; the other 4 GiB are reserved):

  This is a high address range (Q8 – Q15) that requires an address greater than 32 bits. This space is visible only for the MPU Subsystem. See Table 2-8.

Table 2-1 describes the global memory map.

Table 2-1. L3_MAIN Memory Map

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
Q0 (1GiB)	GPMC[1]	0x0000_0000	0x1FFF_FFFF	512 MiB	8/16 Ex[1]/R/W
	PCIE_SS1	0x2000_0000	0x2FFF_FFFF	256 MiB	PCIe_SS1 configuration space
	PCIE_SS2	0x3000_0000	0x3FFF_FFFF	256 MiB	PCIe_SS2 configuration space
Q1 (1GiB)	Reserved	0x4000_0000	0x402F_FFFF	3 MiB	Reserved
	OCMC_RAM1	0x4030_0000	0x4037_FFFF	512 KiB	32bit Ex[1]/R/W
	Reserved	0x4038_0000	0x403F_FFFF	512 KiB	Reserved
	OCMC_RAM2[2]	0x4040_0000	0x404F_FFFF	1 MiB	32bit Ex[1]/R/W
	OCMC_RAM3[2]	0x4050_0000	0x405F_FFFF	1 MiB	32bit Ex[1]/R/W
	Reserved	0x4060_0000	0x407F_FFFF	2 MiB	Reserved
	DSP1_L2_SRAM	0x4080_0000	0x4084_7FFF	288 KiB	DSP1 L2 SRAM and cache. See Table 2-10.
	Reserved	0x4084_8000	0x40CF_FFFF	4832 KiB	Reserved
	DSP1_SYSTEM	0x40D0_0000	0x40D0_0FFF	4 KiB	DSP1 System MMR block
	DSP1_MMU0CFG	0x40D0_1000	0x40D0_1FFF	4 KiB	DSP1 MMU0 configuration
	DSP1_MMU1CFG	0x40D0_2000	0x40D0_2FFF	4 KiB	DSP1 MMU1 configuration
	DSP1_FW0CFG	0x40D0_3000	0x40D0_3FFF	4 KiB	DSP1 Firewall 0 config
	DSP1_FW1CFG	0x40D0_4000	0x40D0_4FFF	4 KiB	DSP1 Firewall 1 config
	DSP1_EDMA_TC0	0x40D0_5000	0x40D0_5FFF	4 KiB	DSP1 EDMA Transfer Controller 0
	DSP1_EDMA_TC1	0x40D0_6000	0x40D0_6FFF	4 KiB	DSP1 EDMA Transfer Controller 1
	DSP1_NoC	0x40D0_7000	0x40D0_7FFF	4 KiB	DSP1 interconnect registers
	Reserved	0x40D0_8000	0x40D0_FFFF	32 KiB	Reserved
	DSP1_EDMA_CC	0x40D1_0000	0x40D1_7FFF	32 KiB	DSP1 EDMA Channel Controller
	Reserved	0x40D1_8000	0x40DF_FFFF	928 KiB	Reserved
	DSP1_L1P_SRAM	0x40E0_0000	0x40E0_7FFF	32 KiB	DSP1 L1P Cache/RAM
	Reserved	0x40E0_8000	0x40EF_FFFF	992 KiB	Reserved
	DSP1_L1D_SRAM	0x40F0_0000	0x40F0_7FFF	32 KiB	DSP1 L1D Cache/RAM
	Reserved	0x40F0_8000	0x40FF_FFFF	992 KiB	Reserved
	DSP2_L2_SRAM	0x4100_0000	0x4104_7FFF	288 KiB	DSP2 L2 SRAM and cache. See Table 2-10.
	Reserved	0x4104_8000	0x414F_FFFF	4832 KiB	Reserved
	DSP2_SYSTEM	0x4150_0000	0x4150_0FFF	4 KiB	DSP2 System MMR block
	DSP2_MMU0CFG	0x4150_1000	0x4150_1FFF	4 KiB	DSP2 MMU0 configuration
	DSP2_MMU1CFG	0x4150_2000	0x4150_2FFF	4 KiB	DSP2 MMU1 configuration
	DSP2_FW0CFG	0x4150_3000	0x4150_3FFF	4 KiB	DSP2 Firewall 0 config
	DSP2_FW1CFG	0x4150_4000	0x4150_4FFF	4 KiB	DSP2 Firewall 1 config
	DSP2_EDMA_TC0	0x4150_5000	0x4150_5FFF	4 KiB	DSP2 EDMA Transfer Controller 0
	DSP2_EDMA_TC1	0x4150_6000	0x4150_6FFF	4 KiB	DSP2 EDMA Transfer Controller 1
	DSP2_NoC	0x4150_7000	0x4150_7FFF	4 KiB	DSP2 interconnect registers
	Reserved	0x4150_8000	0x4150_FFFF	32 KiB	Reserved
	DSP2_EDMA_CC	0x4151_0000	0x4151_7FFF	32 KiB	DSP2 EDMA Channel Controller
	Reserved	0x4151_8000	0x415F_FFFF	928 KiB	Reserved
	DSP2_L1P_SRAM	0x4160_0000	0x4160_7FFF	32 KiB	DSP2 L1P Cache/RAM

Table 2-1. L3_MAIN Memory Map (continued)

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
	Reserved	0x4160_8000	0x416F_FFFF	992 KiB	Reserved
	DSP2_L1D_SRAM	0x4170_0000	0x4170_7FFF	32 KiB	DSP2 L1D Cache/RAM
	Reserved	0x4170_8000	0x417F_FFFF	992 KiB	Reserved
	OCMC_RAM1_CBUF	0x4180_0000	0x41FF_FFFF	8 MiB	OCMC RAM1 CBUF virtual address space (Bit 31 needs to be set on the OCMC data interface)
	EVE1	0x4200_0000	0x420F_FFFF	1 MiB	EVE1 configuration space
	EVE2	0x4210_0000	0x421F_FFFF	1 MiB	EVE2 configuration space
	Reserved	0x4220_0000	0x432F_FFFF	17 MiB	Reserved
	EDMA_TPCC	0x4330_0000	0x433F_FFFF	1 MiB	EDMA TPCC configuration space
	EDMA_TC0	0x4340_0000	0x434F_FFFF	1 MiB	EDMA TPTC1 configuration space
	EDMA_TC1	0x4350_0000	0x435F_FFFF	1 MiB	EDMA TPTC2 configuration space
	Reserved	0x4360_0000	0x43FF_FFFF	10 MiB	Reserved
	L3_MAIN_SN	0x4400_0000	0x457F_FFFF	24 MiB	L3 configuration registers (Service Network)
	McASP1	0x4580_0000	0x45BF_FFFF	4 MiB	McASP1 data port
	McASP2	0x45C0_0000	0x45FF_FFFF	4 MiB	McASP2 data port
	McASP3	0x4600_0000	0x463F_FFFF	4 MiB	McASP3 data port
	VCP1	0x4640_0000	0x4640_FFFF	64 KiB	VCP1 configuration space
	Reserved	0x4641_0000	0x467F_FFFF	4032 KiB	Reserved
	VCP2	0x4680_0000	0x4680_FFFF	64 KiB	VCP2 configuration space
	Reserved	0x4681_0000	0x47FF_FFFF	24 MiB	Reserved
	L4_PER1	0x4800_0000	0x481F_FFFF	2 MiB	L4_PER1 domain. See Table 2-5
	Reserved	0x4820_0000	0x483F_FFFF	2 MiB	MPU private memory space. See Table 2-8
	L4_PER2	0x4840_0000	0x487F_FFFF	4 MiB	L4_PER2 domain. See Table 2-6
	L4_PER3	0x4880_0000	0x48FF_FFFF	8 MiB	L4_PER3 domain. See Table 2-7
	OCMC_RAM2_CBUF	0x4900_0000	0x497F_FFFF	8 MiB	OCMC RAM2 CBUF virtual address space (Bit 31 needs to be set on the OCMC data interface)
	OCMC_RAM3_CBUF	0x4980_0000	0x49FF_FFFF	8 MiB	OCMC RAM3 CBUF virtual address space (Bit 31 needs to be set on the OCMC data interface)
	L4_CFG	0x4A00_0000	0x4ADF_FFFF	14 MiB	L4_CFG domain. See Table 2-3
	L4_WKUP	0x4AE0_0000	0x4AFF_FFFF	2 MiB	L4_WKUP domain. See Table 2-4
	Reserved	0x4B00_0000	0x4B2F_FFFF	3 MiB	Reserved
	QSPI_ADDRSP0	0x4B30_0000	0x4B3F_FFFF	1 MiB	QSPI MMR space (Maddrspace 0)
	Reserved	0x4B40_0000	0x4BFF_FFFF	12 MiB	Reserved
	EMIF1	0x4C00_0000	0x4CFF_FFFF	16 MiB	EMIF1 configuration registers
	EMIF2	0x4D00_0000	0x4DFF_FFFF	16 MiB	EMIF2 configuration registers
	DMM	0x4E00_0000	0x4FFF_FFFF	32 MiB	DMM configuration registers
	GPMC	0x5000_0000	0x50FF_FFFF	16 MiB	GPMC configuration registers
	PCIE_SS1	0x5100_0000	0x517F_FFFF	8 MiB	PCIE_SS1 configuration registers
	PCIE_SS2	0x5180_0000	0x51FF_FFFF	8 MiB	PCIE_SS2 configuration registers
	Reserved	0x5200_0000	0x53FF_FFFF	32 MiB	Reserved
	L3_INSTR	0x5400_0000	0x547F_FFFF	8 MiB	Emulation domain. See Table 2-2
	CT_TBR	0x5480_0000	0x54FF_FFFF	8 MiB	Emulation domain. See Table 2-2
	IPU2_ROM	0x5500_0000	0x5500_3FFF	16 KiB	IPU2_ROM
	Reserved	0x5500_4000	0x5501_FFFF	112 KiB	Reserved
	IPU2_RAM	0x5502_0000	0x5502_FFFF	64 KiB	IPU2_RAM

Table 2-1. L3_MAIN Memory Map (continued)

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
	Reserved	0x5503_0000	0x5507_FFFF	320 KiB	Reserved
	Reserved	0x5508_0000	0x5508_07FF	2KiB	Reserved
	IPU2_UNICACHE_MMU	0x5508_0800	0x5508_0FFF	2KiB	IPU2_UNICACHE_MMU config registers
	Reserved	0x5508_1000	0x5508_1FFF	4KiB	Reserved
	IPU2_MMU	0x5508_2000	0x5508_2FFF	4 KiB	IPU2_MMU config registers
	Reserved	0x5508_3000	0x55FF_FFFF	16 MiB	Reserved
	GPU	0x5600_0000	0x57FF_FFFF	32 MiB	3D GPU domain
	DSS	0x5800_0000	0x587F_FFFF	8 MiB	DSS domain
	IPU1_ROM	0x5880_0000	0x58FF_FFFF	16 KiB	IPU1_ROM
	Reserved	0x5880_4000	0x5881_FFFF	112 KiB	Reserved
	IPU1_RAM	0x5882_0000	0x5882_FFFF	64 KiB	IPU1_RAM
	Reserved	0x5883_0000	0x5887_FFFF	320 KiB	Reserved
	Reserved	0x5888_0000	0x5888_07FF	2KiB	Reserved
	IPU1_UNICACHE_MMU	0x5888_0800	0x5888_0FFF	2KiB	IPU1_UNICACHE_MMU config registers
	Reserved	0x5888_1000	0x5888_1FFF	4KiB	Reserved
	IPU1_MMU	0x5888_2000	0x5888_2FFF	4 KiB	IPU1_MMU config registers
	Reserved	0x5888_3000	0x58FF_FFFF	8 MiB	Reserved
	BB2D	0x5900_0000	0x59FF_FFFF	16 MiB	2D graphics accelerator
	IVA_CONFIG	0x5A00_0000	0x5A3F_FFFF	4 MiB	IVA CONFIG domain
	Reserved	0x5A40_0000	0x5AFF_FFFF	12 MiB	Reserved
	IVA_SL2IF	0x5B00_0000	0x5B3F_FFFF	4 MiB	IVA SL2IF domain
	Reserved	0x5B40_0000	0x5BFF_FFFF	12 MiB	Reserved
	QSPI_ADDRSP1	0x5C00_0000	0x5FFF_FFFF	64 MiB	QSPI CS0/CS1/CS2/CS3 space (Maddrspace 1)
	TILER	0x6000_0000	0x7FFF_FFFF	512 MiB	SDRAM addressing through DMM with TILER off
Q2[(3)] (1GiB)	DDR-SDRAM address space				
	EMIF1_SDRAM_CS0	0x8000_0000	0xBFFF_FFFF	1 GiB	EMIF1 CS0: Access to DDR
	EMIF2_SDRAM_CS0	0x8000_0000	0xBFFF_FFFF	1 GiB	EMIF2 CS0: Access to DDR
Q3[(3)] (1GiB)	EMIF1_SDRAM_CS0	0xC000_0000	0xFFFF_FFFF	1 GiB	EMIF1 CS0: Access to DDR
	EMIF2_SDRAM_CS0	0xC000_0000	0xFFFF_FFFF	1 GiB	EMIF2 CS0: Access to DDR

(1) Ex = Executable
(2) OCMC_RAM2 and OCMC_RAM3 banks are not present on DRA74x devices, but are included in some of the DRA75x devices. For details, see the device data manual.
(3) Depending on the DMM_LISA_MAP_i settings the Q2 address space can be configured in the following ways:
- Allocated only to EMIF1_SDRAM_CS0
- Allocated only to EMIF2_SDRAM_CS0
- Shared between EMIF1_SDRAM_CS0 and EMIF2_SDRAM_CS0 but not interleaved
- Interleaved between EMIF1_SDRAM_CS0 and EMIF2_SDRAM_CS0

The same applies to the Q3 address space.

2.2.1 L3_INSTR Memory Map

The L3_INSTR interconnect is a 8-MiB space composed of the L3_INSTR interconnect configuration registers and module registers.

Table 2-2 describes the mapping of the registers for the L3_INSTR interconnect.

Table 2-2. L3_INSTR Memory Map

Region name	Start_address (hex)	End_address (hex)	Size	Description
CT_STM_ADD_SP_0	0x5400_0000	0x540F_FFFF	1MiB	MIPI_STM - System Trace (address space 0)
CT_STM_ADD_SP_1	0x5410_0000	0x5413_FFFF	256KiB	MIPI_STM - System Trace (address space 1)
MPU_C0_DEBUG	0x5414_0000	0x5414_1FFF	8KiB	MPU_C0 Debug Performance Monitoring Unit
MPU_C1_DEBUG	0x5414_2000	0x5414_3FFF	8KiB	MPU_C1 Debug Performance Monitoring Unit
Reserved	0x5414_4000	0x5414_7FFF	16KiB	Reserved
MPU_C0_CS_CTI_MPU	0x5414_8000	0x5414_8FFF	4KiB	Cross Triggering Interface (CTI0 component)
MPU_C1_CS_CTI_MPU	0x5414_9000	0x5414_9FFF	4KiB	Cross Triggering Interface (CTI1 component)
Reserved	0x5414_A000	0x5414_BFFF	8KiB	Reserved
MPU_C0_CS_PTM_MPU	0x5414_C000	0x5414_CFFF	4KiB	Processor Trace Macrocell Component 0
MPU_C1_CS_PTM_MPU	0x5414_D000	0x5414_DFFF	4KiB	Processor Trace Macrocell Component 1
Reserved	0x5414_E000	0x5415_7FFF	40KiB	Reserved
MPU_CS_TF	0x5415_8000	0x5415_8FFF	4KiB	CS_TF (APBv3) - Trace Funnel for MPU
DAP_PC	0x5415_9000	0x5415_9FFF	4KiB	DAP_PC
MPU_CS_STM	0x5415_A000	0x5415_AFFF	4KiB	CoreSight™ System Trace Module
ATB_FIFO_SGU	0x5415_B000	0x5415_BFFF	4KiB	AMBA® Trace Buffer Static Gathering Unit
Reserved	0x5415_C000	0x5415_EFFF	12KiB	Reserved
T2ASYNC_APB_MPU_DEBUG _MPU_MPU	0x5415_F000	0x5415_FFFF	4KiB	APB Bridge control and time-out register
DRM	0x5416_0000	0x5416_0FFF	4KiB	DRM (OCP) - Debug Register Mapping
CT_STM_CONF_PORT	0x5416_1000	0x5416_1FFF	4KiB	MIPI_STM(OCP) configuration port - System Trace
Reserved	0x5416_2000	0x5416_2FFF	4KiB	Reserved
CS_TPIU	0x5416_3000	0x5416_3FFF	4KiB	CS_TPIU (APBv3) - Trace Port Interface Unit
DEBUGSS_CS_TF_1	0x5416_4000	0x5416_4FFF	4KiB	CS_TF (APBv3) - Trace Funnel for DEBUGSS
Reserved	0x5416_5000	0x5416_6FFF	8KiB	Reserved
CT_TBR	0x5416_7000	0x5416_7FFF	4KiB	C-Tools Trace Buffer
CT_UART	0x5416_8000	0x5416_8FFF	4KiB	C-Tools UART
DEBUGSS_CS_CTI	0x5416_9000	0x5416_9FFF	4KiB	Cross Triggering Interface
DEBUGSS_CS_CTM	0x5416_A000	0x5416_AFFF	4KiB	Core Sight -System Trace Module
MASTER_TIMESTAMP	0x5416_B000	0x5416_BFFF	4KiB	Master Time Stamp
Reserved	0x5416_C000	0x5417_0FFF	20KiB	Reserved
DEBUGSS_OCP2SCP	0x5417_1000	0x5417_1FFF	4KiB	Interconnect registers
L4_CFG_EMU	0x5417_2000	0x5417_2FFF	4KiB	Interconnect registers
Reserved	0x5417_3000	0x5417_FFFF	52KiB	Reserved
L3_INSTR_EMU	0x5418_0000	0x5418_0FFF	4KiB	Interconnect registers
Reserved	0x5418_1000	0x547F_FFFF	6652KiB	Reserved

2.3 L4 Memory Map

The L4 interconnects handle transfers with peripherals. The L4 interconnect comprises the following interconnects:

- L4_CFG
- L4_WKUP
- L4_PER1
- L4_PER2
- L4_PER3

The L4 interconnect can be configured to tune the access according to the characteristics of each module.

The following sections describe the register mapping of the L4 interconnect. Software configures these registers.

2.3.1 L4_CFG Memory Map

The L4_CFG interconnect is a 12-MiB space composed of the L4_CFG interconnect configuration registers and the module registers.

Table 2-3 describes the mapping of the registers for the L4_CFG interconnect.

Note

All memory spaces described as modules provide direct access to module registers outside the L4_CFG interconnect. All other accesses are internal to the L4_CFG interconnect.

Table 2-3. L4_CFG Memory Map

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
L4_CFG	L4_CFG_AP	0x4A00_0000	0x4A00_07FF	2KiB	Address protection
	L4_CFG_LA	0x4A00_0800	0x4A00_0FFF	2KiB	Link agent
	L4_CFG_IA_IP0	0x4A00_1000	0x4A00_1FFF	4KiB	Initiator port
CTRL_MODULE_CORE	TP_CTRL_MODULE_CORE_TARG	0x4A00_2000	0x4A00_3FFF	8KiB	Module target port
	TA_CTRL_MODULE_CORE_TARG	0x4A00_4000	0x4A00_4FFF	4KiB	L4 target agent
CM_CORE_AON	TP_CM_CORE_AON_TARG	0x4A00_5000	0x4A00_5FFF	4KiB	Module target port
	TA_CM_CORE_AON_TARG	0x4A00_6000	0x4A00_6FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A00_7000	0x4A00_7FFF	4KiB	Reserved
CM_CORE	TP_CM_CORE_TARG	0x4A00_8000	0x4A00_9FFF	8KiB	Module target port
	TA_CM_CORE_TARG	0x4A00_A000	0x4A00_AFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A00_B000	0x4A05_5FFF	148KiB	Reserved
DMA_SYSTEM	TP_DMA_SYSTEM_TARG	0x4A05_6000	0x4A05_6FFF	4KiB	Module target port
	TA_DMA_SYSTEM_TARG	0x4A05_7000	0x4A05_7FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A05_8000	0x4A07_FFFF	156KiB	Reserved

Table 2-3. L4_CFG Memory Map (continued)

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
OCP2SCP1	TP_OCP2SCP1_TARG	0x4A08_0000	0x4A08_3FFF	16KiB	Module target port - OCP2SCP module registers
	TP_OCP2SCP1_USB _PHY1_CORE_TARG	0x4A08_4000	0x4A08_43FF	1KiB	Module target port - OCP2SCP target - USB2PHY1
	Reserved	0x4A08_4400	0x4A08_47FF	1KiB	Reserved
	Reserved	0x4A08_4800	0x4A08_4BFF	1KiB	Reserved
	TP_OCP2SCP1_DPLLCTRL _USB_OTG_SS_TARG	0x4A08_4C00	0x4A08_4FFF	1KiB	Module target port - OCP2SCP target- DPLLCTRL_USB_OTG_SS
	TP_OCP2SCP1_USB _PHY2_CORE_TARG	0x4A08_5000	0x4A08_53FF	1KiB	Module target port - OCP2SCP target - USB2PHY2
Reserved	Reserved	0x4A08_5400	0x4A08_7FFF	11KiB	Reserved
L4_CFG	TA_OCP2SCP1_TARG	0x4A08_8000	0x4A08_8FFF	4Kib	L4 target agent
Reserved	Reserved	0x4A08_9000	0x4A08_FFFF	28KiB	Reserved
OCP2SCP3	TP_OCP2SCP3_TARG	0x4A09_0000	0x4A09_3FFF	16KiB	Module target port - OCP2SCP module registers
	Reserved	0x4A09_4000	0x4A09_43FF	1KiB	Reserved
	Reserved	0x4A09_4400	0x4A09_47FF	1KiB	Reserved
	TP_OCP2SCP3_DPLLCTRL _PCIE1_TARG	0x4A09_4800	0x4A09_4BFF	1KiB	Module target port - OCP2SCP target- DPLLCTRL_PCIE
Reserved	Reserved	0x4A09_4C00	0x4A09_4FFF	1KiB	Reserved
OCP2SCP3	Reserved	0x4A09_5000	0x4A09_53FF	1KiB	Reserved
	Reserved	0x4A09_5400	0x4A09_57FF	1KiB	Reserved
Reserved	Reserved	0x4A09_5800	0x4A09_5FFF	2KiB	Reserved
OCP2SCP3	Reserved	0x4A09_6000	0x4A09_63FF	1KiB	Reserved
	Reserved	0x4A09_6400	0x4A09_67FF	1KiB	Reserved
	TP_OCP2SCP3_DPLLCTRL _SATA_TARG	0x4A09_6800	0x4A09_6BFF	1KiB	Module target port - OCP2SCP target- DPLLCTRL_SATA
Reserved	Reserved	0x4A09_6C00	0x4A09_7FFF	5KiB	Reserved
L4_CFG	TA_OCP2SCP3_TARG	0x4A09_8000	0x4A09_8FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A09_9000	0x4A09_FFFF	28KiB	Reserved
OCP2SCP2	TP_OCP2SCP2_TARG	0x4A0A_0000	0x4A0A_3FFF	16KiB	Module target port - OCP2SCP module registers
	TP_OCP2SCP2_DPLLCTRL _VIDEO1_TARG	0x4A0A_4000	0x4A0A_43FF	1KiB	Module target port - OCP2SCP target - DPLLCTRL_VIDEO1
Reserved	Reserved	0x4A0A_4400	0x4A0A_4FFF	3KiB	Reserved
OCP2SCP2	TP_OCP2SCP2_DPLLCTRL _VIDEO2_TARG	0x4A0A_5000	0x4A0A_53FF	1KiB	Module target port - OCP2SCP target - DPLLCTRL_VIDEO2
Reserved	Reserved	0x4A0A_5400	0x4A0A_5FFF	3KiB	Reserved
OCP2SCP2	TP_OCP2SCP2_DPLLCTRL _HDMI_TARG	0x4A0A_6000	0x4A0A_63FF	1KiB	Module target port - OCP2SCP target - DPLLCTRL_HDMI
Reserved	Reserved	0x4A0A_6400	0x4A0A_7FFF	7KiB	Reserved
L4_CFG	TA_OCP2SCP2_TARG	0x4A0A_8000	0x4A0A_8FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A0A_9000	0x4A0F_3FFF	300KiB	Reserved
MAILBOX1	TP_MAILBOX1_TARG	0x4A0F_4000	0x4A0F_4FFF	4KiB	Module target port
	TA_MAILBOX1_TARG	0x4A0F_5000	0x4A0F_5FFF	4KiB	L4 target agent
SPINLOCK	TP_SPINLOCK_TARG	0x4A0F_6000	0x4A0F_6FFF	4KiB	Module target port
	TA_SPINLOCK_TARG	0x4A0F_7000	0x4A0F_7FFF	4KiB	L4 target agent

Table 2-3. L4_CFG Memory Map (continued)

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	Reserved	0x4A0F_8000	0x4A10_1FFF	40KiB	Reserved
OCP_WP_NOC	TP_OCP_WP_NOC_TARG	0x4A10_2000	0x4A10_2FFF	4KiB	Module target port
	TA_OCP_WP_NOC_TARG	0x4A10_3000	0x4A10_3FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A10_4000	0x4A13_FFFF	240KiB	Reserved
SATA	TP_SATA_TARG	0x4A14_0000	0x4A14_FFFF	64KiB	Module target port
	TA_SATA_TARG	0x4A15_0000	0x4A15_0FFF	4KiB	L4 target agent
EVE1	TP_EVE1_FW_CFG_TARG	0x4A15_1000	0x4A15_1FFF	4KiB	Module target port
	TA_EVE1_FW_CFG_TARG	0x4A15_2000	0x4A15_2FFF	4KiB	L4 target agent
EVE2	TP_EVE2_FW_CFG_TARG	0x4A15_3000	0x4A15_3FFF	4KiB	Module target port
	TA_EVE2_FW_CFG_TARG	0x4A15_4000	0x4A15_4FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A15_5000	0x4A15_8FFF	16KiB	Reserved
PCIE_SS2	TP_PCIE_SS2_FW _CFG_TARG	0x4A15_9000	0x4A15_9FFF	4KiB	Module target port
	TA_PCIE_SS2_FW _CFG_TARG	0x4A15_A000	0x4A15_AFFF	4KiB	L4 target agent
IPU1	TP_IPU1_FW_CFG_TARG	0x4A15_B000	0x4A15_BFFF	4KiB	Module target port
	TA_IPU1_FW_CFG_TARG	0x4A15_C000	0x4A15_CFFF	4KiB	L4 target agent
VCP1	TP_VCP1_FW_CFG_TARG	0x4A15_D000	0x4A15_DFFF	4KiB	Module target port
	TA_VCP1_FW_CFG_TARG	0x4A15_E000	0x4A15_EFFF	4KiB	L4 target agent
VCP2	TP_VCP2_FW_CFG_TARG	0x4A15_F000	0x4A15_FFFF	4KiB	Module target port
	TA_VCP2_FW_CFG_TARG	0x4A16_0000	0x4A16_0FFF	4KiB	L4 target agent
EDMA_TPCC	TP_EDMA_TPCC_FW_CFG _TARG	0x4A16_1000	0x4A16_1FFF	4KiB	Module target port
	TA_EDMA_TPCC_FW _CFG_TARG	0x4A16_2000	0x4A16_2FFF	4KiB	L4 target agent
EDMA_TC0	TP_EDMA_TC0_FW _CFG_TARG	0x4A16_3000	0x4A16_3FFF	4KiB	Module target port
	TA_EDMA_TC0_FW _CFG_TARG	0x4A16_4000	0x4A16_4FFF	4KiB	L4 target agent
PCIE_SS1	TP_PCIE_SS1_FW _CFG_TARG	0x4A16_5000	0x4A16_5FFF	4KiB	Module target port
	TA_PCIE_SS1_FW _CFG_TARG	0x4A16_6000	0x4A16_6FFF	4KiB	L4 target agent
McASP1	TP_MCASP1_FW _CFG_TARG	0x4A16_7000	0x4A16_7FFF	4KiB	Module target port
	TA_MCASP1_FW _CFG_TARG	0x4A16_8000	0x4A16_8FFF	4KiB	L4 target agent
McASP2	TP_MCASP2_FW _CFG_TARG	0x4A16_9000	0x4A16_9FFF	4KiB	Module target port
	TA_MCASP2_FW_CFG _TARG	0x4A16_A000	0x4A16_AFFF	4KiB	L4 target agent
McASP3	TP_MCASP3_FW _CFG_TARG	0x4A16_B000	0x4A16_BFFF	4KiB	Module target port
	TA_MCASP3_FW _CFG_TARG	0x4A16_C000	0x4A16_CFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A16_D000	0x4A17_0FFF	16KiB	Reserved
DSP1	TP_DSP1_FW_CFG_TARG	0x4A17_1000	0x4A17_1FFF	4KiB	Module target port
	TA_DSP1_FW_CFG_TARG	0x4A17_2000	0x4A17_2FFF	4KiB	L4 target agent
DSP2	TP_DSP2_FW_CFG_TARG	0x4A17_3000	0x4A17_3FFF	4KiB	Module target port

Table 2-3. L4_CFG Memory Map (continued)

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
	TA_DSP2_FW_CFG_TARG	0x4A17_4000	0x4A17_4FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A17_5000	0x4A17_8FFF	16KiB	Reserved
QSPI	TP_QSPI_FW_CFG_TARG	0x4A17_9000	0x4A17_9FFF	4KiB	Module target port
	TA_QSPI_FW_CFG_TARG	0x4A17_A000	0x4A17_AFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A17_B000	0x4A20_9FFF	572KiB	Reserved
MA_MPU _NTTP	TP_MA_MPU_NTTP_FW _CFG_TARG	0x4A20_A000	0x4A20_AFFF	4KiB	Module target port
	TA_MA_MPU_NTTP_FW _CFG_TARG	0x4A20_B000	0x4A20_BFFF	4KiB	L4 target agent
EMIF_OCP_FW	TP_EMIF_OCP_FW _CFG_TARG	0x4A20_C000	0x4A20_CFFF	4KiB	Module target port
	TA_EMIF_OCP_FW _CFG_TARG	0x4A20_D000	0x4A20_DFFF	4KiB	L4 target agent
OCMC_RAM2	TP_OCMC_RAM2_FW _CFG_TARG	0x4A20_E000	0x4A20_EFFF	4KiB	Module target port
	TA_OCMC_RAM2_FW _CFG_TARG	0x4A20_F000	0x4A20_FFFF	4KiB	L4 target agent
GPMC	TP_GPMC_FW _CFG_TARG	0x4A21_0000	0x4A21_0FFF	4KiB	Module target port
	TA_GPMC_FW _CFG_TARG	0x4A21_1000	0x4A21_1FFF	4KiB	L4 target agent
OCMC_RAM1	TP_OCMC_RAM1_FW_CFG_T ARG	0x4A21_2000	0x4A21_2FFF	4KiB	Module target port
	TA_OCMC_RAM1_FW_CFG_T ARG	0x4A21_3000	0x4A21_3FFF	4KiB	L4 target agent
GPU	TP_GPU_FW_CFG_TARG	0x4A21_4000	0x4A21_4FFF	4KiB	Module target port
	TA_GPU_FW_CFG_TARG	0x4A21_5000	0x4A21_5FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A21_6000	0x4A21_7FFF	8KiB	Reserved
IPU2	TP_IPU2_FW_CFG_TARG	0x4A21_8000	0x4A21_8FFF	4KiB	Module target port
	TA_IPU2_FW_CFG_TARG	0x4A21_9000	0x4A21_9FFF	4KiB	L4 target agent
BB2D	TP_BB2D_FW_CFG_TARG	0x4A21_A000	0x4A21_AFFF	4KiB	Module target port
	TA_BB2D_FW_CFG_TARG	0x4A21_B000	0x4A21_BFFF	4KiB	L4 target agent
DSS	TP_DSS_FW_CFG_TARG	0x4A21_C000	0x4A21_CFFF	4KiB	Module target port
	TA_DSS_FW_CFG_TARG	0x4A21_D000	0x4A21_DFFF	4KiB	L4 target agent
IVA	TP_IVA_SL2IF_FW _CFG_TARG	0x4A21_E000	0x4A21_EFFF	4KiB	Module target port
	TA_IVA_SL2IF_FW _CFG_TARG	0x4A21_F000	0x4A21_FFFF	4KiB	L4 target agent
IVA	TP_IVA_CONFIG_FW _CFG_TARG	0x4A22_0000	0x4A22_0FFF	4KiB	Module target port
	TA_IVA_CONFIG_FW _CFG_TARG	0x4A22_1000	0x4A22_1FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A22_2000	0x4A22_3FFF	8KiB	Reserved
DEBUGSS	TP_DEBUGSS_CT_TBR _FW_CFG_TARG	0x4A22_4000	0x4A22_4FFF	4KiB	Module target port. See Table 2-2
	TA_DEBUGSS_CT_TBR _FW_CFG_TARG	0x4A22_5000	0x4A22_5FFF	4KiB	L4 target agent

Table 2-3. L4_CFG Memory Map (continued)

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
L3_INSTR	TP_L3_INSTR_FW _CFG_TARG	0x4A22_6000	0x4A22_6FFF	4KiB	Module target port
	TA_L3_INSTR_FW _CFG_TARG	0x4A22_7000	0x4A22_7FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A22_8000	0x4A22_9FFF	8MiB	Reserved
OCMC_RAM3	TP_OCMC_RAM3_FW _CFG_TARG	0x4A22_A000	0x4A22_AFFF	4MiB	Module target port
	TA_OCMC_RAM3_FW _CFG_TARG	0x4A22_B000	0x4A22_BFFF	4MiB	L4 target agent
Reserved	Reserved	0x4A22_C000	0x4ADF_FFFF	12112MiB	Reserved

2.3.2 L4_WKUP Memory Map

The L4_WKUP interconnect is a 256-KiB space composed of the L4_WKUP interconnect configuration registers and the module registers.

Table 2-4 describes the mapping of the registers for the L4_WKUP interconnect.

Note

All memory spaces described as modules provide direct access to module registers outside the L4_WKUP interconnect. All other accesses are internal to the L4_WKUP interconnect.

Table 2-4. L4_WKUP Memory Map

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
L4_WKUP	L4_WKUP_AP	0x4AE0_0000	0x4AE0_07FF	2 KiB	Address protection
	L4_WKUP_LA	0x4AE0_0800	0x4AE0_0FFF	2 KiB	Link agent
	L4_WKUP_IA_IP0	0x4AE0_1000	0x4AE0_1FFF	4 KiB	Initiator port
Reserved	Reserved	0x4AE0_2000	0x4AE0_3FFF	8 KiB	Reserved
COUNTER_32K	TP_COUNTER_32K _TARG	0x4AE0_4000	0x4AE0_4FFF	4 KiB	Module target port
	TA_COUNTER_32K _TARG	0x4AE0_5000	0x4AE0_5FFF	4 KiB	L4 target agent
PRM	TP_PRM_TARG	0x4AE0_6000	0x4AE0_7FFF	8 KiB	Module target port
	TA_PRM_TARG	0x4AE0_8000	0x4AE0_8FFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE0_9000	0x4AE0_BFFF	12 KiB	Reserved
CTRL_MODULE _WKUP	TP_CTRL_MODULE _WKUP_TARG	0x4AE0_C000	0x4AE0_CFFF	4 KiB	Module target port
	TA_CTRL_MODULE _WKUP_TARG	0x4AE0_D000	0x4AE0_DFFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE0_E000	0x4AE0_FFFF	8 KiB	Reserved
GPIO1	TP_GPIO1_TARG	0x4AE1_0000	0x4AE1_0FFF	4 KiB	Module target port
	TP_GPIO1_TARG	0x4AE1_1000	0x4AE1_1FFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE1_2000	0x4AE1_3FFF	8 KiB	Reserved
WD_TIMER2	TP_WD_TIMER2_TARG	0x4AE1_4000	0x4AE1_4FFF	4 KiB	Module target port
	TA_WD_TIMER2_TARG	0x4AE1_5000	0x4AE1_5FFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE1_6000	0x4AE1_7FFF	8 KiB	Reserved

Table 2-4. L4_WKUP Memory Map (continued)

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
TIMER1	TP_TIMER1_TARG	0x4AE1_8000	0x4AE1_8FFF	4 KiB	Module target port
	TA_TIMER1_TARG	0x4AE1_9000	0x4AE1_9FFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE1_A000	0x4AE1_BFFF	8 KiB	Reserved
KBD	TP_KBD_TARG	0x4AE1_C000	0x4AE1_CFFF	4 KiB	Module target port
	TA_KBD_TARG	0x4AE1_D000	0x4AE1_DFFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE1_E000	0x4AE1_FFFF	8 KiB	Reserved
TIMER12	TP_TIMER12_TARG	0x4AE2_0000	0x4AE2_0FFF	4 KiB	Module target port
	TA_TIMER12_TARG	0x4AE2_1000	0x4AE2_1FFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE2_2000	0x4AE2_AFFF	36 KiB	Reserved
UART10	TP_UART10_TARG	0x4AE2_B000	0x4AE2_BFFF	4 KiB	Module target port
	TA_UART10_TARG	0x4AE2_C000	0x4AE2_CFFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE2_D000	0x4AE3_BFFF	60 KiB	Reserved
DCAN1	TP_DCAN1_TARG	0x4AE3_C000	0x4AE3_DFFF	8 KiB	Module target port
	TA_DCAN1_TARG	0x4AE3_E000	0x4AE3_EFFF	4 KiB	L4 target agent
Reserved	Reserved	0x4AE3_F000	0x4AFF_FFFF	1796 KiB	Reserved

Note

8- and 16-bit peripherals are aligned on 32-bit address boundaries.

2.4 L4_PER Memory Map

The L4_PER interconnect has three memory spaces:

- L4_PER1 memory space (Table 2-5)
- L4_PER2 memory space (Table 2-6)
- L4_PER3 memory space (Table 2-7)

The L4_PER interconnects are composed of the L4_PER interconnect confguration registers and the module registers.

Note

All memory spaces described as modules provide direct access to the module registers outside the L4_PER interconnects. All other accesses are internal to the L4_PERs interconnects.

2.4.1 L4_PER1 Memory Space Mapping

Table 2-5 describes the mapping of the registers for the L4_PER1 interconnect. DRA75

Table 2-5. L4_PER1 Memory Map

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
L4_PER1 interconnect	L4_PER1_AP	0x4800_0000	0x4800_07FF	2KiB	Address protection
	L4_PER1_LA	0x4800_0800	0x4800_0FFF	2KiB	Link agent
	L4_PER1_IA_IP0	0x4800_1000	0x4800_13FF	1KiB	Initiator port 0
	L4_PER1_IA_IP1	0x4800_1400	0x4800_17FF	1KiB	Initiator port 1
	L4_PER1_IA_IP2	0x4800_1800	0x4800_1BFF	1KiB	Initiator port 2
	L4_PER1_IA_IP3	0x4800_1C00	0x4800_1FFF	1KiB	Initiator port 3
Reserved	Reserved	0x4800_1C00	0x4801_FFFF	121KiB	Reserved

Table 2-5. L4_PER1 Memory Map (continued)

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
UART3	TP_UART3_TARG	0x4802_0000	0x4802_0FFF	4KiB	Module target port
	TA_UART3_TARG	0x4802_1000	0x4802_1FFF	4KiB	L4 target agent
Reserved	Reserved	0x4802_2000	0x4803_1FFF	64KiB	Reserved
TIMER2	TP_TIMER2_TARG	0x4803_2000	0x4803_2FFF	4KiB	Module target port
	TA_TIMER2_TARG	0x4803_3000	0x4803_3FFF	4KiB	L4 target agent
TIMER3	TP_TIMER3_TARG	0x4803_4000	0x4803_4FFF	4KiB	Module target port
	TA_TIMER3_TARG	0x4803_5000	0x4803_5FFF	4KiB	L4 target agent
TIMER4	TP_TIMER4_TARG	0x4803_6000	0x4803_6FFF	4KiB	Module target port
	TA_TIMER4_TARG	0x4803_7000	0x4803_7FFF	4KiB	L4 target agent
Reserved	Reserved	0x4803_8000	0x4803_DFFF	24KiB	Reserved
TIMER9	TP_TIMER9_TARG	0x4803_E000	0x4803_EFFF	4KiB	Module target port
	TA_TIMER9_TARG	0x4803_F000	0x4803_FFFF	4KiB	L4 target agent
Reserved	Reserved	0x4804_0000	0x4805_0FFF	68KiB	Reserved
GPIO7	TP_GPIO7_TARG	0x4805_1000	0x4805_1FFF	4KiB	Module target port
	TA_GPIO7_TARG	0x4805_2000	0x4805_2FFF	4KiB	L4 target agent
GPIO8	TP_GPIO8_TARG	0x4805_3000	0x4805_3FFF	4KiB	Module target port
	TA_GPIO8_TARG	0x4805_4000	0x4805_4FFF	4KiB	L4 target agent
GPIO2	TP_GPIO2_TARG	0x4805_5000	0x4805_5FFF	4KiB	Module target port
	TA_GPIO2_TARG	0x4805_6000	0x4805_6FFF	4KiB	L4 target agent
GPIO3	TP_GPIO3_TARG	0x4805_7000	0x4805_7FFF	4KiB	Module target port
	TA_GPIO3_TARG	0x4805_8000	0x4805_8FFF	4KiB	L4 target agent
GPIO4	TP_GPIO4_TARG	0x4805_9000	0x4805_9FFF	4KiB	Module target port
	TA_GPIO4_TARG	0x4805_A000	0x4805_AFFF	4KiB	L4 target agent
GPIO5	TP_GPIO5_TARG	0x4805_B000	0x4805_BFFF	4KiB	Module target port
	TA_GPIO5_TARG	0x4805_C000	0x4805_CFFF	4KiB	L4 target agent
GPIO6	TP_GPIO6_TARG	0x4805_D000	0x4805_DFFF	4KiB	Module target port
	TA_GPIO6_TARG	0x4805_E000	0x4805_EFFF	4KiB	L4 target agent
Reserved	Reserved	0x4805_F000	0x4805_FFFF	4KiB	Reserved
I2C3	TP_I2C3_TARG	0x4806_0000	0x4806_0FFF	4KiB	Module target port
	TA_I2C3_TARG	0x4806_1000	0x4806_1FFF	4KiB	L4 target agent
Reserved	Reserved	0x4806_2000	0x4806_5FFF	16KiB	L4 interconnect target agent
UART5	TP_UART5_TARG	0x4806_6000	0x4806_6FFF	4KiB	Module target port
	TA_UART5_TARG	0x4806_7000	0x4806_7FFF	4KiB	L4 target agent
UART6	TP_UART6_TARG	0x4806_8000	0x4806_8FFF	4KiB	Module target port
	TA_UART6_TARG	0x4806_9000	0x4806_9FFF	4KiB	L4 target agent
UART1	TP_UART1_TARG	0x4806_A000	0x4806_AFFF	4KiB	Module target port
	TA_UART1_TARG	0x4806_B000	0x4806_BFFF	4KiB	L4 target agent
UART2	TP_UART2_TARG	0x4806_C000	0x4806_CFFF	4KiB	Module target port
	TA_UART2_TARG	0x4806_D000	0x4806_DFFF	4KiB	L4 target agent
UART4	TP_UART4_TARG	0x4806_E000	0x4806_EFFF	4KiB	Module target port
	TA_UART4_TARG	0x4806_F000	0x4806_FFFF	4KiB	L4 target agent
I2C1	TP_I2C1_TARG	0x4807_0000	0x4807_0FFF	4KiB	Module target port
	TA_I2C1_TARG	0x4807_1000	0x4807_1FFF	4KiB	L4 target agent

Table 2-5. L4_PER1 Memory Map (continued)

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
I2C2	TP_I2C2_TARG	0x4807_2000	0x4807_2FFF	4KiB	Module target port
	TA_I2C2_TARG	0x4807_3000	0x4807_3FFF	4KiB	L4 target agent
Reserved	Reserved	0x4807_4000	0x4807_7FFF	16KiB	Reserved
ELM	TP_ELM_TARG	0x4807_8000	0x4807_8FFF	4KiB	Module target port
	TA_ELM_TARG	0x4807_9000	0x4807_9FFF	4KiB	L4 target agent
I2C4	TP_I2C4_TARG	0x4807_A000	0x4807_AFFF	4KiB	Module target port
	TA_I2C4_TARG	0x4807_B000	0x4807_BFFF	4KiB	L4 target agent
I2C5	TP_I2C5_TARG	0x4807_C000	0x4807_CFFF	4KiB	Module target port
	TA_I2C5_TARG	0x4807_D000	0x4807_DFFF	4KiB	L4 target agent
Reserved	Reserved	0x4807_E000	0x4808_5FFF	32KiB	Reserved
TIMER10	TP_TIMER10_TARG	0x4808_6000	0x4808_6FFF	4KiB	Module target port
	TA_TIMER10_TARG	0x4808_7000	0x4808_7FFF	4KiB	L4 target agent
TIMER11	TP_TIMER11_TARG	0x4808_8000	0x4808_8FFF	4KiB	Module target port
	TA_TIMER11_TARG	0x4808_9000	0x4808_9FFF	4KiB	L4 target agent
Reserved	Reserved	0x4808_A000	0x4809_7FFF	56KiB	Reserved
McSPI1	TP_MCSPI1_TARG	0x4809_8000	0x4809_8FFF	4KiB	Module target port
	TA_MCSPI1_TARG	0x4809_9000	0x4809_9FFF	4KiB	L4 target agent
McSPI2	TP_MCSPI2_TARG	0x4809_A000	0x4809_AFFF	4KiB	Module target port
	TA_MCSPI2_TARG	0x4809_B000	0x4809_BFFF	4KiB	L4 target agent
MMC1	TP_MMC1_TARG	0x4809_C000	0x4809_CFFF	4KiB	Module target port
	TA_MMC1_TARG	0x4809_D000	0x4809_DFFF	4KiB	L4 target agent
Reserved	Reserved	0x4809_E000	0x480A_CFFF	60KiB	Reserved
MMC3	TP_MMC3_TARG	0x480A_D000	0x480A_DFFF	4KiB	Module target port
	TA_MMC3_TARG	0x480A_E000	0x480A_EFFF	4KiB	L4 target agent
Reserved	Reserved	0x480A_F000	0x480B_1FFF	12KiB	Reserved
HDQ1W	TP_HDQ1W_TARG	0x480B_2000	0x480B_2FFF	4KiB	Module target port
	TA_HDQ1W_TARG	0x480B_3000	0x480B_3FFF	4KiB	L4 target agent
MMC2	TP_MMC2_TARG	0x480B_4000	0x480B_4FFF	4KiB	Module target port
	TA_MMC2_TARG	0x480B_5000	0x480B_5FFF	4KiB	L4 target agent
Reserved	Reserved	0x480B_6000	0x480B_7FFF	8KiB	Reserved
McSPI3	TP_MCSPI3_TARG	0x480B_8000	0x480B_8FFF	4KiB	Module target port
	TA_MCSPI3_TARG	0x480B_9000	0x480B_9FFF	4KiB	L4 target agent
McSPI4	TP_MCSPI4_TARG	0x480B_A000	0x480B_AFFF	4KiB	Module target port
	TA_MCSPI4_TARG	0x480B_B000	0x480B_BFFF	4KiB	L4 target agent
Reserved	Reserved	0x480B_C000	0x480D_0FFF	84KiB	Reserved
MMC4	TP_MMC4_TARG	0x480D_1000	0x480D_1FFF	4KiB	Module target port
	TA_MMC4_TARG	0x480D_2000	0x480D_2FFF	4KiB	L4 target agent
Reserved	Reserved	0x480D_3000	0x483F_FFFF	3252KiB	Reserved

2.4.2 L4_PER2 Memory Map

Table 2-6 describes the mapping of the register for the L4_PER2 interconnect.

Table 2-6. L4_PER2 Memory Map

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
L4_PER2 interconnect	L4_PER2_AP	0x4840_0000	0x4840_07FF	2KiB	Address protection
	L4_PER2_LA	0x4840_0800	0x4840_0FFF	2KiB	Link Agent
	L4_PER2_IA_IP0	0x4840_1000	0x4840_13FF	1KiB	Initiator Port 0
	L4_PER2_IA_IP1	0x4840_1400	0x4840_17FF	1KiB	Initiator Port 1
	L4_PER2_IA_IP2	0x4840_1800	0x4840_1BFF	1KiB	Initiator Port 2
Reserved	Reserved	0x4840_1C00	0x4841_FFFF	121KiB	Reserved
UART7	TP_UART7_TARG	0x4842_0000	0x4842_0FFF	4KiB	Module target port
	TA_UART7_TARG	0x4842_1000	0x4842_1FFF	4KiB	L4 interconnect target agent
UART8	TP_UART8_TARG	0x4842_2000	0x4842_2FFF	4KiB	Module target port
	TA_UART8_TARG	0x4842_3000	0x4842_3FFF	4KiB	L4 interconnect target agent
UART9	TP_UART9_TARG	0x4842_4000	0x4842_4FFF	4KiB	Module target port
	TA_UART9_TARG	0x4842_5000	0x4842_5FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4842_6000	0x4842_BFFF	24KiB	Reserved
MLB	TP_MLB_TARG	0x4842_C000	0x4842_CFFF	4KiB	Module target port
	TA_MLB_TARG	0x4842_D000	0x4842_DFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4842_E000	0x4843_5FFF	32KiB	Reserved
McASP4	TP_MCASP4_DAT_TARG	0x4843_6000	0x4843_6FFF	4KiB	Module target port
	TA_MCASP4_DAT_TARG	0x4843_7000	0x4843_7FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4843_8000	0x4843_9FFF	8KiB	Reserved
McASP5	TP_MCASP5_DAT_TARG	0x4843_A000	0x4843_AFFF	4KiB	Module target port
	TA_MCASP5_DAT_TARG	0x4843_B000	0x4843_BFFF	4KiB	L4 interconnect target agent
ATL	TP_ATL_TARG	0x4843_C000	0x4843_CFFF	4KiB	Module target port
	TA_ATL_TARG	0x4843_D000	0x4843_DFFF	4KiB	L4 interconnect target agent
PWMSS1	TP_PWMSS1_TARG	0x4843_E000	0x4843_EFFF	4KiB	Module target port
	TA_PWMSS1_TARG	0x4843_F000	0x4843_FFFF	4KiB	L4 interconnect target agent
PWMSS2	TP_PWMSS2_TARG	0x4844_0000	0x4844_0FFF	4KiB	Module target port
	TA_PWMSS2_TARG	0x4844_1000	0x4844_1FFF	4KiB	L4 interconnect target agent
PWMSS3	TP_PWMSS3_TARG	0x4844_2000	0x4844_2FFF	4KiB	Module target port
	TA_PWMSS3_TARG	0x4844_3000	0x4844_3FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4844_4000	0x4844_5FFF	8KiB	Reserved
VCP1	TP_VCP1_CFG_TARG	0x4844_6000	0x4844_6FFF	4KiB	Module target port
	TA_VCP1_CFG_TARG	0x4844_7000	0x4844_7FFF	4KiB	L4 interconnect target agent
VCP2	TP_VCP2_CFG_TARG	0x4844_8000	0x4844_8FFF	4KiB	Module target port
	TA_VCP2_CFG_TARG	0x4844_9000	0x4844_9FFF	4KiB	L4 interconnect target agent
IODELAYCONFIG	TP_DELAYLINE_TARG	0x4844_A000	0x4844_AFFF	4KiB	Module target port
Reserved	Reserved	0x4844_B000	0x4844_BFFF	4KiB	Reserved
McASP6	TP_MCASP6_DAT_TARG	0x4844_C000	0x4844_CFFF	4KiB	Module target port
	TA_MCASP6_DAT_TARG	0x4844_D000	0x4844_DFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4844_E000	0x4844_FFFF	8KiB	Reserved
McASP7	TP_MCASP7_DAT_TARG	0x4845_0000	0x4845_0FFF	4KiB	Module target port
	TA_MCASP7_DAT_TARG	0x4845_1000	0x4845_1FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4845_2000	0x4845_3FFF	8KiB	Reserved
McASP8	TP_MCASP8_DAT_TARG	0x4845_4000	0x4845_4FFF	4KiB	Module target port
	TA_MCASP8_DAT_TARG	0x4845_5000	0x4845_5FFF	4KiB	L4 interconnect target agent

Table 2-6. L4_PER2 Memory Map (continued)

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	Reserved	0x4845_6000	0x4845_FFFF	40KiB	Reserved
McASP1	TP_MCASP1_CFG_TARG	0x4846_0000	0x4846_1FFF	8KiB	Module target port
	TA_MCASP1_CFG_TARG	0x4846_2000	0x4846_2FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4846_3000	0x4846_3FFF	4KiB	Reserved
McASP2	TP_MCASP2_CFG_TARG	0x4846_4000	0x4846_5FFF	8KiB	Module target port
	TA_MCASP2_CFG_TARG	0x4846_6000	0x4846_6FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4846_7000	0x4846_7FFF	4KiB	Reserved
McASP3	TP_MCASP3_CFG_TARG	0x4846_8000	0x4846_9FFF	8KiB	Module target port
	TA_MCASP3_CFG_TARG	0x4846_A000	0x4846_AFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4846_B000	0x4846_BFFF	4KiB	Reserved
McASP4	TP_MCASP4_CFG_TARG	0x4846_C000	0x4846_DFFF	8KiB	Module target port
	TA_MCASP4_CFG_TARG	0x4846_E000	0x4846_EFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4846_F000	0x4846_FFFF	4KiB	Reserved
McASP5	TP_MCASP5_CFG_TARG	0x4847_0000	0x4847_1FFF	8KiB	Module target port
	TA_MCASP5_CFG_TARG	0x4847_2000	0x4847_2FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4847_3000	0x4847_3FFF	4KiB	Reserved
McASP6	TP_MCASP6_CFG_TARG	0x4847_4000	0x4847_5FFF	8KiB	Module target port
	TA_MCASP6_CFG_TARG	0x4847_6000	0x4847_6FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4847_7000	0x4847_7FFF	4KiB	Reserved
McASP7	TP_MCASP7_CFG_TARG	0x4847_8000	0x4847_9FFF	8KiB	Module target port
	TA_MCASP7_CFG_TARG	0x4847_A000	0x4847_AFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4847_B000	0x4847_BFFF	4KiB	Reserved
McASP8	TP_MCASP8_CFG_TARG	0x4847_C000	0x4847_DFFF	8KiB	Module target port
	TA_MCASP8_CFG_TARG	0x4847_E000	0x4847_EFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4847_F000	0x4847_FFFF	4KiB	Reserved
DCAN2	TP_DCAN2_TARG	0x4848_0000	0x4848_1FFF	8KiB	Module target port
	TA_DCAN2_TARG	0x4848_2000	0x4848_2FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4848_3000	0x4848_3FFF	4KiB	Reserved
GMAC_SW	TP_GMAC_SW_TARG	0x4848_4000	0x4848_7FFF	16KiB	Module target port
	TA_GMAC_SW_TARG	0x4848_8000	0x4848_8FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4848_9000	0x487F_FFFF	3548KiB	Reserved

2.4.3 L4_PER3 Memory Map

Table 2-7 describes the mapping of the register for the L4_PER3 interconnect.

Table 2-7. L4_PER3 Memory Map

Module name	Region name tag	Start_address (hex)	End_address (hex)	Size	Description
L4_PER3 interconnect	L4_PER3_AP	0x4880_0000	0x4880_07FF	2KiB	Address protection
	L4_PER3_LA	0x4880_0800	0x4880_0FFF	2KiB	Link agent
	L4_PER3_IA_IP0	0x4880_1000	0x4880_13FF	1KiB	Initiator Port 0
	L4_PER3_IA_IP1	0x4880_1400	0x4880_17FF	1KiB	Initiator Port 1
	L4_PER3_IA_IP2	0x4880_1800	0x4880_1BFF	1KiB	Initiator Port 2
Reserved	Reserved	0x4880_1C00	0x4880_1FFF	1KiB	Reserved

Table 2-7. L4_PER3 Memory Map (continued)

Module name	Region name tag	Start_address (hex)	End_address (hex)	Size	Description
MAILBOX13	TP_MAILBOX13_TARG	0x4880_2000	0x4880_2FFF	4KiB	Module target port
	TA_MAILBOX13_TARG	0x4880_3000	0x4880_3FFF	4KiB	L4 interconnect target agent
OCMC_RAM1	TP_OCMC_RAM1 _CFG_TARG	0x4880_4000	0x4880_4FFF	4KiB	Module target port
	TA_OCMC_RAM1 _CFG_TARG	0x4880_5000	0x4880_5FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4880_6000	0x4880_9FFF	16KiB	Reserved
OCMC_RAM2	TP_OCMC_RAM2 _CFG_TARG	0x4880_A000	0x4880_AFFF	4KiB	Module target port
	TA_OCMC_RAM2 _CFG_TARG	0x4880_B000	0x4880_BFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4880_C000	0x4880_FFFF	16KiB	Reserved
OCMC_RAM3	TP_OCMC_RAM3 _CFG_TARG	0x4881_0000	0x4881_0FFF	4KiB	Module target port
	TA_OCMC_RAM3 _CFG_TARG	0x4881_1000	0x4881_1FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4881_2000	0x4881_BFFF	40KiB	Reserved
MMU1	TP_MMU1_TARG	0x4881_C000	0x4881_CFFF	4KiB	Module target port
	TA_MMU1_TARG	0x4881_D000	0x4881_DFFF	4KiB	L4 interconnect target agent
MMU2	TP_MMU2_TARG	0x4881_E000	0x4881_EFFF	4KiB	Module target port
	TA_MMU2_TARG	0x4881_F000	0x4881_FFFF	4KiB	L4 interconnect target agent
TIMER5	TP_TIMER5_TARG	0x4882_0000	0x4882_0FFF	4KiB	Module target port
	TA_TIMER5_TARG	0x4882_1000	0x4882_1FFF	4KiB	L4 interconnect target agent
TIMER6	TP_TIMER6_TARG	0x4882_2000	0x4882_2FFF	4KiB	Module target port
	TA_TIMER6_TARG	0x4882_3000	0x4882_3FFF	4KiB	L4 interconnect target agent
TIMER7	TP_TIMER7_TARG	0x4882_4000	0x4882_4FFF	4KiB	Module target port
	TA_TIMER7_TARG	0x4882_5000	0x4882_5FFF	4KiB	L4 interconnect target agent
TIMER8	TP_TIMER8_TARG	0x4882_6000	0x4882_6FFF	4KiB	Module target port
	TA_TIMER8_TARG	0x4882_7000	0x4882_7FFF	4KiB	L4 interconnect target agent
TIMER13	TP_TIMER13_TARG	0x4882_8000	0x4882_8FFF	4KiB	Module target port
	TA_TIMER13_TARG	0x4882_9000	0x4882_9FFF	4KiB	L4 interconnect target agent
TIMER14	TP_TIMER14_TARG	0x4882_A000	0x4882_AFFF	4KiB	Module target port
	TA_TIMER14_TARG	0x4882_B000	0x4882_BFFF	4KiB	L4 interconnect target agent
TIMER15	TP_TIMER15_TARG	0x4882_C000	0x4882_CFFF	4KiB	Module target port
	TA_TIMER15_TARG	0x4882_D000	0x4882_DFFF	4KiB	L4 interconnect target agent
TIMER16	TP_TIMER16_TARG	0x4882_E000	0x4882_EFFF	4KiB	Module target port
	TA_TIMER16_TARG	0x4882_F000	0x4882_FFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4883_0000	0x4883_7FFF	32KiB	Reserved
RTC_SS	TP_RTC_SS_TARG	0x4883_8000	0x4883_8FFF	4KiB	Module target port
	TA_RTC_SS_TARG	0x4883_9000	0x4883_9FFF	4KiB	L4 interconnect target agent
MAILBOX2	TP_MAILBOX2_TARG	0x4883_A000	0x4883_AFFF	4KiB	Module target port
	TA_MAILBOX2_TARG	0x4883_B000	0x4883_BFFF	4KiB	L4 interconnect target agent
MAILBOX3	TP_MAILBOX3_TARG	0x4883_C000	0x4883_CFFF	4KiB	Module target port
	TA_MAILBOX3_TARG	0x4883_D000	0x4883_DFFF	4KiB	L4 interconnect target agent
MAILBOX4	TP_MAILBOX4_TARG	0x4883_E000	0x4883_EFFF	4KiB	Module target port
	TA_MAILBOX4_TARG	0x4883_F000	0x4883_FFFF	4KiB	L4 interconnect target agent

Table 2-7. L4_PER3 Memory Map (continued)

Module name	Region name tag	Start_address (hex)	End_address (hex)	Size	Description
MAILBOX5	TP_MAILBOX5_TARG	0x4884_0000	0x4884_0FFF	4KiB	Module target port
	TA_MAILBOX5_TARG	0x4884_1000	0x4884_1FFF	4KiB	L4 interconnect target agent
MAILBOX6	TP_MAILBOX6_TARG	0x4884_2000	0x4884_2FFF	4KiB	Module target port
	TA_MAILBOX6_TARG	0x4884_3000	0x4884_3FFF	4KiB	L4 interconnect target agent
MAILBOX7	TP_MAILBOX7_TARG	0x4884_4000	0x4884_4FFF	4KiB	Module target port
	TA_MAILBOX7_TARG	0x4884_5000	0x4884_5FFF	4KiB	L4 interconnect target agent
MAILBOX8	TP_MAILBOX8_TARG	0x4884_6000	0x4884_6FFF	4KiB	Module target port
	TA_MAILBOX8_TARG	0x4884_7000	0x4884_7FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4884_8000	0x4885_DFFF	88KiB	Reserved
MAILBOX9	TP_MAILBOX9_TARG	0x4885_E000	0x4885_EFFF	4KiB	Module target port
	TA_MAILBOX9_TARG	0x4885_F000	0x4885_FFFF	4KiB	L4 interconnect target agent
MAILBOX10	TP_MAILBOX10_TARG	0x4886_0000	0x4886_0FFF	4KiB	Module target port
	TA_MAILBOX10_TARG	0x4886_1000	0x4886_1FFF	4KiB	L4 interconnect target agent
MAILBOX11	TP_MAILBOX11_TARG	0x4886_2000	0x4886_2FFF	4KiB	Module target port
	TA_MAILBOX11_TARG	0x4886_3000	0x4886_3FFF	4KiB	L4 interconnect target agent
MAILBOX12	TP_MAILBOX12_TARG	0x4886_4000	0x4886_4FFF	4KiB	Module target port
	TA_MAILBOX12_TARG	0x4886_5000	0x4886_5FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4886_6000	0x4887_FFFF	104KiB	Reserved
USB1	TP_USB1_CFG_TARG	0x4888_0000	0x4889_FFFF	128KiB	Module target port
	TA_USB1_CFG_TARG	0x488A_0000	0x488A_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x488A_1000	0x488B_FFFF	124KiB	Reserved
USB2	TP_USB2_CFG_TARG	0x488C_0000	0x488D_FFFF	128KiB	Module target port
	TA_USB2_CFG_TARG	0x488E_0000	0x488E_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x488E_1000	0x488F_FFFF	124KiB	Reserved
USB3	TP_USB3_CFG_TARG	0x4890_0000	0x4891_FFFF	128KiB	Module target port
	TA_USB3_CFG_TARG	0x4892_0000	0x4892_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4892_1000	0x4893_FFFF	124KiB	Reserved
USB4	TP_USB4_CFG_TARG	0x4894_0000	0x4895_FFFF	128KiB	Module target port
	TA_USB4_CFG_TARG	0x4896_0000	0x4896_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4896_1000	0x4896_FFFF	60KiB	Reserved
VIP1	TP_VIP1_TARG	0x4897_0000	0x4897_FFFF	64KiB	Module target port
	TA_VIP1_TARG	0x4898_0000	0x4898_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4898_1000	0x4898_FFFF	60KiB	Reserved
VIP2	TP_VIP2_TARG	0x4899_0000	0x4899_FFFF	64KiB	Module target port
	TA_VIP2_TARG	0x489A_0000	0x489A_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x489A_1000	0x489A_FFFF	60KiB	Reserved
VIP3	TP_VIP3_TARG	0x489B_0000	0x489B_FFFF	64KiB	Module target port
	TA_VIP3_TARG	0x489C_0000	0x489C_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x489C_1000	0x489C_FFFF	60KiB	Reserved
VPE	TP_VPE_TARG	0x489D_0000	0x489D_FFFF	64KiB	Module target port
	TA_VPE_TARG	0x489E_0000	0x489E_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x489E_1000	0x48FF_FFFF	6268KiB	Reserved

2.5 MPU Memory Map

Table 2-8 describes the MPU memory mapping.

Table 2-8. MPU Memory Map

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
Q0	L3_MAIN map	0x00_0000_0000	0x00_3FFF_FFFF	1GiB	See Table 2-1
Q1	Reserved	0x00_4000_0000	0x00_4003_7FFF	224KiB	Reserved
	MPU_ROM[(1)]	0x00_4003_8000	0x00_4004_3FFF	48KiB	MPU internal boot ROM: 32bit Ex[(2)]/R
	Reserved	0x00_4004_4000	0x00_402F_FFFF	2800KiB	Reserved
	L3_MAIN map	0x00_4030_0000	0x00_46FF_FFFF	114MiB	See Table 2-1
	MPU_CS_STM	0x00_4700_0000	0x00_47FF_FFFF	16MiB	MPU_CS_STM config registers
	L3_MAIN map	0x00_4800_0000	0x00_481F_FFFF	2MiB	See Table 2-1
	MPU_INTC	0x00_4821_0000	0x00_4821_7FFF	32KiB	MPU_INTC config registers
	Reserved	0x00_4821_8000	0x00_4824_2FFF	172KiB	Reserved
	MPU_PRCM	0x00_4824_3000	0x00_4824_3FFF	4KiB	MPU_PRCM config registers
	Reserved	0x00_4824_4000	0x00_4828_0FFF	242KiB	Reserved
	MPU_CMU	0x00_4829_0000	0x00_4829_FFFF	64KiB	MPU_CMU config registers
	MPU_AXI2OCP	0x00_482A_0000	0x00_482A_EFFF	60KiB	MPU_AXI2OCP config registers
	MPU_MA	0x00_482A_F000	0x00_482A_FFFF	4KiB	MPU_MA config registers
	Reserved	0x00_482B_0000	0x00_483F_FFFF	1344KiB	Reserved
	L3_MAIN map	0x00_4840_0000	0x00_7FFF_FFFF	892MiB	See Table 2-1
Q2, Q3	L3_MAIN map	0x00_8000_0000	0x00_FFFF_FFFF	2GiB	See Table 2-1
	8 GiB[(3)] of SDRAM virtualization when interleaving[(4)] is disabled.				
Q8	EMIF1_SDRAM_CS0	0x02_0000_0000	0x02_3FFF_FFFF	1GiB	EMIF1 CS0: Access to DDR
Q9	Reserved	0x02_4000_0000	0x02_7FFF_FFFF	1GiB	Reserved
Q10[(5)]	EMIF1_SDRAM_CS0	0x02_8000_0000	0x02_BFFF_FFFF	1GiB	EMIF1 CS0: Access to DDR. Alias of Q2 (see Table 2-1).
	EMIF2_SDRAM_CS0	0x02_8000_0000	0x02_BFFF_FFFF	1GiB	EMIF2 CS0: Access to DDR. Alias of Q2 (see Table 2-1).
Q11[(5)]	EMIF1_SDRAM_CS0	0x02_C000_0000	0x02_FFFF_FFFF	1GiB	EMIF1 CS0: Access to DDR. Alias of Q3 (see Table 2-1).
	EMIF2_SDRAM_CS0	0x02_C000_0000	0x02_FFFF_FFFF	1GiB	EMIF2 CS0: Access to DDR. Alias of Q3 (see Table 2-1).
Q12	Reserved	0x03_0000_0000	0x03_3FFF_FFFF	1GiB	Reserved
Q13	Reserved	0x03_4000_0000	0x03_7FFF_FFFF	1GiB	Reserved
Q14	Reserved	0x03_8000_0000	0x03_BFFF_FFFF	1GiB	Reserved
Q15	EMIF2_SDRAM_CS0	0x03_C000_0000	0x03_FFFF_FFFF	1GiB	EMIF2 CS0: Access to DDR
	8 GiB[(3)] of SDRAM virtualization when interleaving[(4)] is enabled.				
Q8	EMIF1_SDRAM_CS0	0x02_0000_0000	0x02_3FFF_FFFF	1GiB[(6)]	EMIF1 CS0: Access to DDR
	EMIF2_SDRAM_CS0	0x02_0000_0000	0x02_3FFF_FFFF	1GiB[(6)]	EMIF2 CS0: Access to DDR
Q9	EMIF1_SDRAM_CS0	0x02_4000_0000	0x02_7FFF_FFFF	1GiB[(6)]	EMIF1 CS0: Access to DDR
	EMIF2_SDRAM_CS0	0x02_4000_0000	0x02_7FFF_FFFF	1GiB[(6)]	EMIF2 CS0: Access to DDR
Q10[(5)]	EMIF1_SDRAM_CS0	0x02_8000_0000	0x02_BFFF_FFFF	1GiB	EMIF1 CS0: Access to DDR. Alias of Q2 (see Table 2-1).
	EMIF2_SDRAM_CS0	0x02_8000_0000	0x02_BFFF_FFFF	1GiB	EMIF2 CS0: Access to DDR. Alias of Q2 (see Table 2-1).

Table 2-8. MPU Memory Map (continued)

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
Q11[(5)]	EMIF1_SDRAM_CS0	0x02_C000_0000	0x02_FFFF_FFFF	1GiB	EMIF1 CS0: Access to DDR. Alias of Q3 (see Table 2-1).
	EMIF2_SDRAM_CS0	0x02_C000_0000	0x02_FFFF_FFFF	1GiB	EMIF2 CS0: Access to DDR. Alias of Q3 (see Table 2-1).
Q12	Reserved	0x03_0000_0000	0x03_3FFF_FFFF	1GiB	Reserved
Q13	Reserved	0x03_4000_0000	0x03_7FFF_FFFF	1GiB	Reserved
Q14	Reserved	0x03_8000_0000	0x03_BFFF_FFFF	1GiB	Reserved
Q15	Reserved	0x03_C000_0000	0x03_FFFF_FFFF	1GiB	Reserved

Legend:		= MPU private memory space
		= Reserved memory space

(1) Boot space location depends on the external sys_boot [5:0] pins.
(2) Ex = Executable
(3) Only 4 GiB are physically available within this 8-GiB address range.
(4) Interleaving is configurable with one setting in the MPU memory adapter (MPU_MA) for all 8 GiB of SDRAM high memory. For more information about interleaving, see Section 4.3.4 , *Memory Adapter*.
(5) As Q10 is alias of Q2, depending on the DMM_LISA_MAP_i settings the Q10 address space can be configured in the following ways:

- Allocated only to EMIF1_SDRAM_CS0
- Allocated only to EMIF2_SDRAM_CS0
- Shared between EMIF1_SDRAM_CS0 and EMIF2_SDRAM_CS0 but not interleaved
- Interleaved between EMIF1_SDRAM_CS0 and EMIF2_SDRAM_CS0

The same applies to the Q11 address space, as it is alias of Q3.
(6) Because of the fixed MPU-to-EMIF address mapping with high-order interleaving enabled only 512MiB of this address space are used by EMIF1_SDRAM_CS0. The other 512MiB are used by EMIF2_SDRAM_CS0.

2.6 IPU Memory Map

The device implements two IPU subsystems (IPU1, IPU2). For more information about IPU, see Chapter 7.

Table 2-9 describes the IPU memory mapping.

Note

Some of the system (L3) resources, such as EVE and EDMA, are not directly accessible by IPU, as they are overlapping with IPU's own resources in its memory space. In such cases, software must properly configure IPU AMMU / L2 MMU so that IPU can access these system resources.

Table 2-9. IPU Memory Map

Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
IPU_BOOT_SPACE(1)	0x0000_0000	0x0000_3FFF	16KiB	IPU boot space
L3_MAIN map	0x0000_0000	0x1FFF_FFFF	512MiB	See Table 2-1
IPU_BITBAND_REGION1	0x2000_0000	0x200F_FFFF	1MiB	IPU bit-band region 1
Reserved	0x2010_0000	0x21FF_FFFF	31MiB	Reserved
IPU_BITBAND_ALIAS1	0x2200_0000	0x23FF_FFFF	32MiB	IPU bit-band alias 1
L3_MAIN map	0x2400_0000	0x3FFF_FFFF	448MiB	See Table 2-1
IPU_BITBAND_REGION2	0x4000_0000	0x400F_FFFF	1MiB	IPU bit-band region 2
Reserved	0x4010_0000	0x402F_FFFF	2MiB	Reserved
L3_MAIN map	0x4030_0000	0x41FF_FFFF	30MiB	See Table 2-1
IPU_BITBAND_ALIAS2	0x4200_0000	0x43FF_FFFF	32MiB	IPU bit-band alias 2
L3_MAIN map	0x4400_0000	0x54FF_FFFF	285MiB	See Table 2-1
IPU_ROM(2)	0x5500_0000	0x5500_3FFF	16KiB	IPU_ROM
IPU_RAM(2)	0x5502_0000	0x5502_FFFF	64KiB	IPU_RAM
IPU_UNICACHE_CFG	0x5508_0000	0x5508_00FF	256B	IPU_UNICACHE config registers
Reserved	0x5508_0100	0x5508_03FF	768B	Reserved
IPU_UNICACHE_SCTM	0x5508_0400	0x5508_07FF	1KiB	IPU_UNICACHE_SCTM config registers
IPU_UNICACHE_MMU (2)	0x5508_0800	0x5508_0FFF	2KiB	IPU_UNICACHE_MMU config registers
IPU_WUGEN	0x5508_1000	0x5508_1FFF	4KiB	IPU_WUGEN config registers
IPU_MMU(2)	0x5508_2000	0x5508_2FFF	4KiB	IPU_MMU config registers
Reserved	0x5508_3000	0x55FF_FFFF	16MiB	Reserved
L3_MAIN map	0x5600_0000	0xDFFF_FFFF	2,3GiB	See Table 2-1
Reserved	0xE000_0000	0xE000_0FFF	4KiB	Reserved
IPU_C0_DWT	0xE000_1000	0xE000_1FFF	4KiB	IPU_C0_DWT config registers
IPU_C0_FPB	0xE000_2000	0xE000_2FFF	4KiB	IPU_C0_FPB config registers
IPU_C0_INTC	0xE000_E000	0xE000_EFFF	4KiB	IPU_C0_INTC config registers
IPU_C0_ICECRUSHER	0xE004_2000	0xE004_2FFF	4KiB	IPU_C0_ICECRUSHER config registers
IPU_C0_RW_TABLE	0xE00F_E000	0xE00F_EFFF	4KiB	IPU_C0 RW table
IPU_C0_ROM_TABLE	0xE00F_F000	0xE00F_FFFF	4KiB	IPU_C0 ROM table
IPU_C1_DWT	0xE000_1000	0xE000_1FFF	4KiB	IPU_C1_DWT config registers
IPU_C1_FPB	0xE000_2000	0xE000_2FFF	4KiB	IPU_C1_FPB config registers

Table 2-9. IPU Memory Map (continued)

Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
IPU_C1_INTC	0xE000_E000	0xE000_EFFF	4KiB	IPU_C1_INTC config registers
IPU_C1_ICECRUSHER	0xE004_2000	0xE004_2FFF	4KiB	IPU_C1_ICECRUSHER config registers
IPU_C1_RW_TABLE	0xE00F_E000	0xE00F_EFFF	4KiB	IPU_C1 RW table
IPU_C1_ROM_TABLE	0xE00F_F000	0xE00F_FFFF	4KiB	IPU_C1 ROM table
L3_MAIN map	0xE010_0000	0xFFFF_FFFF	511MiB	See Table 2-1

Legend:		= IPU private memory space
		= Reserved memory space

(1) See Section 7.3.8, *IPU Boot Options*.
(2) Can also be accessed from L3_MAIN (by other initiators, such as: MPU, DSP, etc).

2.7 DSP Memory Map

The device implements two DSP subsystems (DSP1, DSP2). For more information about DSP, see Chapter 5.

Table 2-10 describes the DSP memory mapping.

Table 2-10. DSP Memory Map

Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
Reserved	0x0000_0000	0x007F_FFFF	8MiB	Reserved
DSP_L2[(1)]	0x0080_0000	0x0084_7FFF	288KiB	DSP L2 SRAM and cache. The L2 SRAM starts at 0x0080_0000 address.
Reserved	0x0084_8000	0x00DF_FFFF	5856KiB	Reserved
DSP_L1P[(1)]	0x00E0_0000	0x00E0_7FFF	32KiB	DSP L1P SRAM
Reserved	0x00E0_8000	0x00EF_FFFF	992KiB	Reserved
DSP_L1D[(1)]	0x00F0_0000	0x00F0_7FFF	32KiB	DSP L1D SRAM
Reserved	0x00F0_8000	0x00FF_FFFF	992KiB	Reserved
DSP_ICFG[(1)]	0x0100_0000	0x01BF_FFFF	12MiB	DSP internal CFG
Reserved	0x01C0_0000	0x01CF_FFFF	1MiB	Reserved
DSP_SYSTEM[(1)]	0x01D0_0000	0x01D0_0FFF	4KiB	DSP system registers block
DSP_MMU0CFG[(1)]	0x01D0_1000	0x01D0_1FFF	4KiB	DSP MMU0 configuration
DSP_MMU1CFG[(1)]	0x01D0_2000	0x01D0_2FFF	4KiB	DSP MMU1 configuration
DSP_FW0CFG[(1)]	0x01D0_3000	0x01D0_3FFF	4KiB	DSP firewall 0 config
DSP_FW1CFG[(1)]	0x01D0_4000	0x01D0_4FFF	4KiB	DSP firewall 1 config
DSP_EDMA_TC0[(1)]	0x01D0_5000	0x01D0_5FFF	4KiB	DSP EDMA transfer controller 0
DSP_EDMA_TC1[(1)]	0x01D0_6000	0x01D0_6FFF	4KiB	DSP EDMA transfer controller 1
DSP_NoC[(1)]	0x01D0_7000	0x01D0_7FFF	4KiB	DSP interconnect registers
Reserved	0x01D0_8000	0x01D0_FFFF	32KiB	Reserved
DSP_EDMA_CC[(1)]	0x01D1_0000	0x01D1_7FFF	32KiB	DSP EDMA channel controller
Reserved	0x01D1_8000	0x01FF_FFFF	2976KiB	Reserved
EVE1	0x0200_0000	0x020F_FFFF	1MiB	EVE1 configuration space
EVE2	0x0210_0000	0x021F_FFFF	1MiB	EVE2 configuration space
Reserved	0x0220_0000	0x032F_FFFF	17MiB	Reserved
EDMA_TPCC	0x0330_0000	0x033F_FFFF	1MiB	EDMA_TPCC configuration space
EDMA_TC0	0x0340_0000	0x034F_FFFF	1MiB	EDMA_TC0 configuration space
EDMA_TC1	0x0350_0000	0x035F_FFFF	1MiB	EDMA_TC1 configuration space
Reserved	0x0360_0000	0x07FF_FFFF	74MiB	Reserved
DSP_XMC_CTRL[(1)]	0x0800_0000	0x0800_FFFF	64KiB	DSP XMC control registers
DSP_EDI[(1)]	0x0801_0000	0x0801_FFFF	64KiB	DSP internal EDI translation region
L3_MAIN map	0x1400_0000	0xFFFF_FFFF	3,8GiB	See Table 2-1

Legend:		= DSP private memory space
		= Reserved memory space

(1) DSP subsystem internal resources. DSP accesses in ranges [0x0080_0000 – 0x01D1_7FFF] and [0x0800_0000 – 0x0801_FFFF] are performed locally within the DSP subsystem.

2.8 EVE Memory Map

The device implements two EVE subsystems (EVE1, EVE2). For more information about EVE, see Section 8.1.

Table 2-11 describes the EVE memory mapping.

Table 2-11. EVE Memory Map

Region name	Start_Address (hex)	End_Address (hex)	Size	Description
L3_MAIN map	0x0000_0000	0x3FFF_FFFF	1GiB	See Table 2-1
Reserved	0x4000_0000	0x4001_FFFF	128KiB	Reserved
EVE_DMEM	0x4002_0000	0x4002_7FFF	32KiB	EVE ARP32 data memory (DMEM)
Reserved	0x4002_8000	0x4003_FFFF	96KiB	Reserved
EVE_WBUF	0x4004_0000	0x4004_7FFF	32KiB	EVE VCOP working buffer (WBUF)
Reserved	0x4004_8000	0x4004_FFFF	32KiB	Reserved
EVE_IBUF_LA	0x4005_0000	0x4005_3FFF	16KiB	EVE image buffer low copy A (IBUFLA)
EVE_IBUF_HA	0x4005_4000	0x4005_7FFF	16KiB	EVE image buffer high copy A (IBUFHA)
Reserved	0x4005_8000	0x4006_FFFF	96KiB	Reserved
EVE_IBUF_LB	0x4007_0000	0x4007_3FFF	16KiB	EVE image buffer low copy B (IBUFLB)
EVE_IBUF_HB	0x4007_4000	0x4007_7FFF	16KiB	EVE image buffer high copy B (IBUFHB)
Reserved	0x4007_8000	0x4007_FFFF	32KiB	Reserved
EVE_SYSTEM	0x4008_0000	0x4008_0FFF	4KiB	EVE int, reset, clk, pwr, buffswc (MEMSWITCH_CTL)
EVE_MMU0_CFG	0x4008_1000	0x4008_1FFF	4KiB	EVE MMU0 configuration
EVE_MMU1_CFG	0x4008_2000	0x4008_2FFF	4KiB	EVE MMU1 configuration
EVE_T16C	0x4008_3000	0x4008_3FFF	4KiB	EVE T16 control
EVE_VCOPC	0x4008_4000	0x4008_4FFF	4KiB	EVE VCOP control
EVE_SCTM	0x4008_5000	0x4008_5FFF	4KiB	EVE Counter and Timer module (SCTM)
EVE_EDMA_TC0	0x4008_6000	0x4008_6FFF	4KiB	EVE EDMA Transfer controller 0 (EDMA TC0)
EVE_EDMA_TC1	0x4008_7000	0x4008_7FFF	4KiB	EVE EDMA Transfer controller 1 (EDMA TC1)
EVE_SMSET_CFG	0x4008_8000	0x4008_8FFF	4KiB	EVE SMSET configuration interface
EVE_SMSET_MSG	0x4008_9000	0x4008_9FFF	4KiB	EVE SMSET messaging interface
EVE_NoC	0x4008_A000	0x4008_AFFF	4KiB	EVE interconnect registers
EVE_MBX0	0x4008_B000	0x4008_BFFF	4KiB	EVE Mailbox 0
EVE_MBX1	0x4008_C000	0x4008_CFFF	4KiB	EVE Mailbox 1
EVE_MBX2	0x4008_D000	0x4008_DFFF	4KiB	EVE Mailbox 2
EVE_MBX3	0x4008_E000	0x4008_EFFF	4KiB	EVE Mailbox 3
EVE_MBX4	0x4008_F000	0x4008_FFFF	4KiB	EVE Mailbox 4
EVE_PCACHE_RAW	0x4009_0000	0x4009_7FFF	32KiB	EVE Program Cache Raw
EVE_PCACHE_Tags	0x4009_8000	0x4009_FFFF	32KiB	EVE Program Cache Tag
EVE_EDMA_CC	0x400A_0000	0x400A_7FFF	32KiB	EVE EDMA Channel Controller (EDMA CC)
Reserved	0x400A_8000	0x402F_FFFF	2.4MiB	Reserved
L3_MAIN map	0x4030_0000	0xFFFF_FFFF	3GiB	See Table 2-1

Legend:		= EVE private memory space
		= Reserved memory space

2.9 TILER View Memory Map

Table 2-12 describes the TILER view memory mapping.

Table 2-12. TILER View Memory Map

Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
TILER_VIEW_0	0x01_0000_0000	0x01_1FFF_FFFF	512MiB	Natural view
TILER_VIEW_1	0x01_2000_0000	0x01_3FFF_FFFF	512MiB	0° view with vertical mirror
TILER_VIEW_2	0x01_4000_0000	0x01_5FFF_FFFF	512MiB	0° view with horizontal mirror
TILER_VIEW_3	0x01_6000_0000	0x01_7FFF_FFFF	512MiB	180° view
TILER_VIEW_4	0x01_8000_0000	0x01_9FFF_FFFF	512MiB	90° view with vertical mirror
TILER_VIEW_5	0x01_A000_0000	0x01_BFFF_FFFF	512MiB	270° view
TILER_VIEW_6	0x01_C000_0000	0x01_DFFF_FFFF	512MiB	90° view
TILER_VIEW_7	0x01_E000_0000	0x01_FFFF_FFFF	512MiB	90° view with horizontal mirror

Note

TILER view memory space is only visible for Display Subsystem (DSS) .

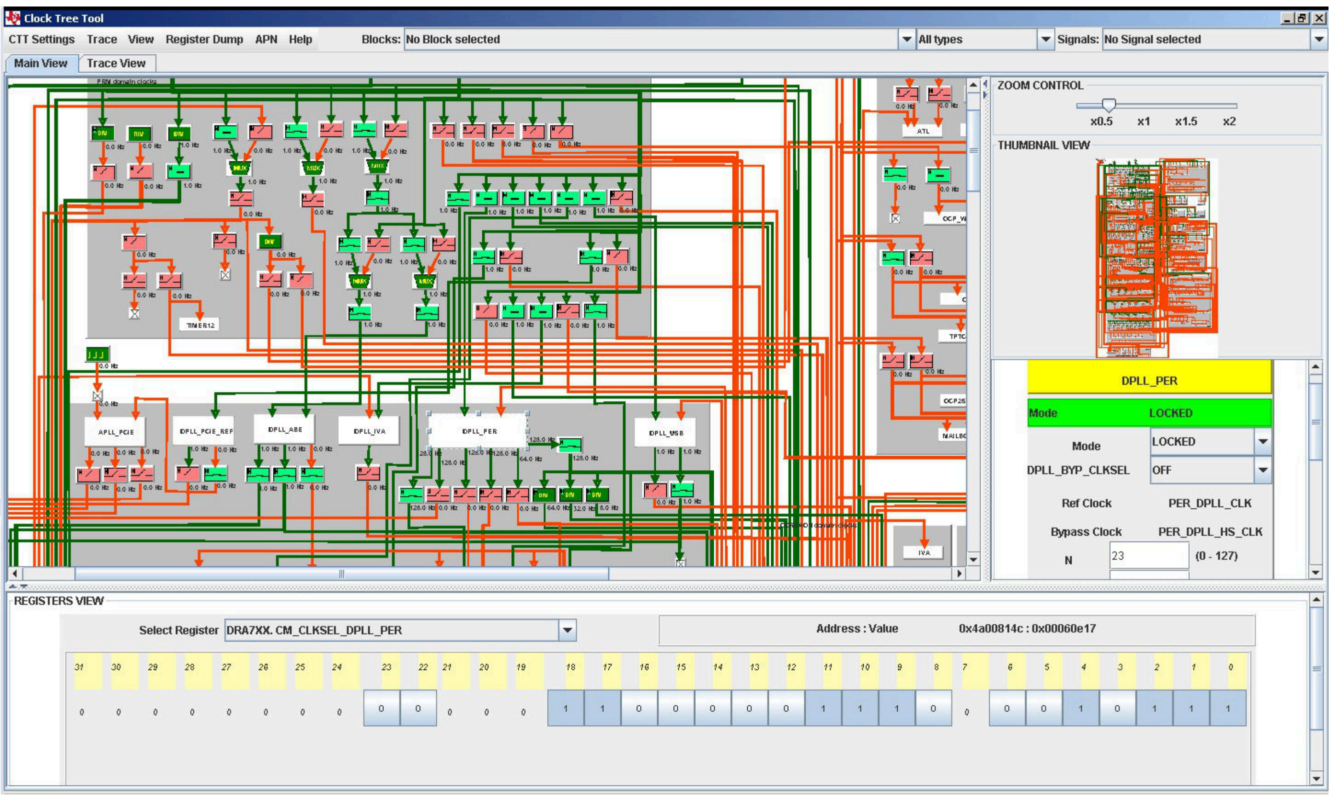
Chapter 3

Power, Reset, and Clock Management

This chapter describes the power, reset, and clock management in the device.

Note

For a detailed visual representation of the distribution and management of the device clocks at the PRCM level, see the clock device interactive software for the device CLOCKTREETOOL-AUTOMOTIVE.



prcm-095

Figure 3-1. Clock Tree Tool (CTT)

The Clock Tree Tool (CTT) is a Java™-based, stand-alone application. The CTT is interactive clock tree configuration software for the device. The CTT lets the user:

- Visualize the device clock tree
- Interact with clock tree elements and view the effect on PRCM registers
- Interact with the PRCM registers and view the effect on the device clock tree
- View a trace of all the device registers affected by the user interaction with clock tree
- Extract registers and register dumps for Code Composer Studio™ and Lauterbach

The advantage of the CTT is that the user can visualize the clock tree state of the device on power-on reset and then customize the configuration of the clock tree for the specific use case and identify the device register settings associated to that configuration. Furthermore, the user can dump and read in register settings for the specific scenarios.

Being an interactive visual tool, the CTT gives the user a global view of the device clock tree architecture and allows determining the exact register settings to obtain the specific configuration.

3.1 Device Power Management Introduction 299
3.2 PRCM Subsystem Overview 328
3.3 PRCM Subsystem Environment 330
3.4 PRCM Subsystem Integration 334
3.5 Reset Management Functional Description 340
3.6 Clock Management Functional Description 383
3.7 Power Management Functional Description 565
3.8 Voltage-Management Functional Description 597
3.9 Device Low-Power States 602
3.10 PRCM Module Programming Guide 607
3.11 612
3.12 PRCM Software Configuration for OPP_PLUS 613
3.13 PRCM Register Manual 613

3.1 Device Power Management Introduction

Power management is one of the most important design aspects of any system.

The device power-management architecture ensures maximum performance and operation time for user satisfaction (audio/video support) while offering versatile power-management techniques for maximum design flexibility, depending on application requirements.

This introduction contains the following information:

- Power-management architecture building blocks for the device
- State-of-the-art power-management techniques supported by the power-management architecture of the device

3.1.1 Device Power-Management Architecture Building Blocks

To provide a versatile architecture that supports multiple power-management techniques, the power-management framework is built with three levels of resource management: clock, power, and voltage.

These management levels are enforced by defining the managed entities or building blocks of the power-management architecture, called the clock, power, and voltage domains.

A domain is a group of modules or subsections of the device that share a common entity (for example, common clock source, common voltage source, or common power switch). The group forming the domain is managed by a policy manager. For example, a clock for a clock domain is managed by a dedicated clock manager within the power, reset, and clock management (PRCM) module. The clock manager considers the joint clocking constraints of all the modules belonging to that clock domain (and, hence, receiving that clock).

Note

In the following sections, the term <*module*> is used to represent the device IPs (that is, modules or subsystems), other than the PRCM module, that receive clock, reset, or power signals from the PRCM module.

3.1.1.1 Clock Management

The PRCM module manages the gating (that is, switching off) and enabling of the clocks to the device modules. The clocks are managed based on the requirement constraints of the associated modules. The following sections identify the module clock characteristics, management policy, clock domains, and clock domain management.

3.1.1.1.1 Module Interface and Functional Clocks

Each module within the device has specific clock input characteristics requirements. Based on the characteristics of the clocks delivered to the modules, the clocks are divided into two categories: interface clocks and functional clocks (see Figure 3-2).

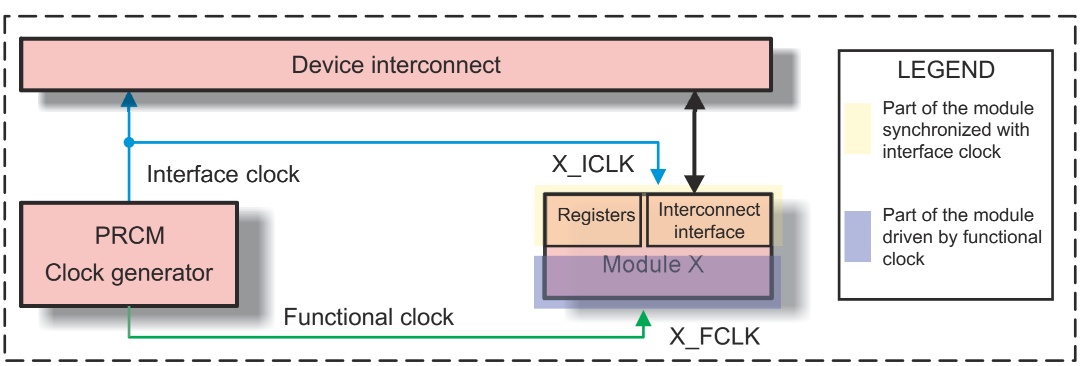


Figure 3-2. Functional and Interface Clocks

The interface clocks have the following characteristics:

- They ensure proper communication between any module/subsystem and the interconnect.
- In most cases, they supply the system interconnect interface and registers of the module.
- A typical module has one interface clock, but modules with multiple interface clocks may also exist (that is, when connected to multiple interconnect buses).
- Interface clock management is done at the device level.
- From the standpoint of the PRCM module, an interface clock is identified with an _ICLK suffix.

Functional clocks have the following characteristics:

- They supply the functional part of a module or subsystem.
- A module can have one or more functional clocks. Some functional clocks are mandatory, while others are optional. A module needs its mandatory clock(s) to be operational. The optional clocks are used for specific features and can be shut down without stopping the module activity (for example, the clock for the camera).
- From the standpoint of the PRCM module, a functional clock is distributed directly to the related modules through a dedicated clock tree. It is identified with an _FCLK suffix.

Some clocks are qualified as permanent clocks. They are functional clocks, that can stay active while the corresponding entity manages them.

3.1.1.1.2

Note

At the module level, the interface clocks are always fed by the interface clock outputs of the PRCM module. The functional clocks are fed by a PRCM module functional clock output or a PRCM module interface clock output. In the latter case, the functional and interface module clocks inherit the clock-management features (autoidle features) of the PRCM module interface clock.

3.1.1.1.3 Module-Level Clock Management

Each module in the device may also have specific clock requirements. Certain module clocks must be active when operating in specific modes, or they may be gated. Globally, the activation and gating of the module clocks are managed by the PRCM module. Hence, the PRCM module must be aware of when to activate and when to gate the module clocks.

The PRCM module differentiates the clock-management behavior for device modules based on whether the module can initiate transactions on the device interconnect (called master module) or it cannot initiate transactions and only responds to the transactions initiated by the master (called slave module). Thus, two hardware-based power-management protocols are used:

- Master standby protocol: Clock-management protocol between the PRCM and master modules
- Slave idle protocol: Clock-management protocol between the PRCM and slave modules

Master standby protocol

This protocol is used to indicate that a master module must initiate a transaction on the device interconnect and requests specific (functional and interface) clocks for that purpose. The PRCM module ensures that the required clocks are active when the master module requests the PRCM module to enable them. This is called a module wake-up transition and the module is said to be functional after this transition completes.

Similarly, when the master module no longer requires the clocks, it informs the PRCM module, which can then gate the clocks to the module. The master module is then said to be in standby mode.

Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the module. This is done by setting the <*Module*>_SYSCONFIG. MIDLEMODE or <*Module*>_SYSCONFIG. STANDBYMODE bit fields, as described in Table 3-1. The behavior, identified in the STANDBYMODE Bit Value column, must be configured.

Table 3-1. Master Module Standby Mode Settings

STANDBYMODE Bit Value	Selected Mode	Description
0x0	Force-standby	The module unconditionally asserts the standby request to the PRCM module, regardless of its internal operations. The PRCM module may gate the functional and interface clocks to the module. This mode must be used carefully because it does not prevent loss of data at the time the clocks are gated.
0x1	No-standby	The module never asserts the standby request to the PRCM module. This mode is safe from a module point of view because it ensures that the clocks remain active; however, it is not efficient from a power-saving perspective because it never allows the PRCM module output clocks to be gated.
0x2	Smart-standby	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idled. The PRCM module can then gate the clocks to the module.
0x3	Smart-standby wake-up-capable mode	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idle. The PRCM module can then gate the clocks to the module. The module may generate (master-related) wake-up events when in standby state. The mode is relevant only if the appropriate module mwakeup output is implemented.

Note

- Smart-standby mode is the preferred mode of operation, while force-standby and no-standby modes are intended for debugging purposes.
- A master module may support all or some of the standby modes listed in Table 3-1. See the power-management section in the module chapter to identify the supported standby mode.

The standby status of a master module is indicated by the CM_*<Power domain>*_*<Module>*_CLKCTRL[x]. STBYST bit in the PRCM module. Table 3-2 describes the master module standby status.

Table 3-2. Master Module Standby Status

STBYST Bit Value	Description
0x0	The module is functional.
0x1	The module is in standby mode.

Table 3-3 lists the enabling conditions for the master module clocks managed by the standby protocol.

Table 3-3. Master Module Clock Enabling Conditions

Relation		Condition
AND		Clock domain is ready.
	OR	Master module standby request is deasserted.
		Master module wake-up request is asserted.

Slave idle protocol

This hardware protocol allows the PRCM module to control the state of a slave module. The PRCM module informs the slave module, through assertion of an IDLE request, when its clocks (interface and functional) can be gated. The slave can then acknowledge the request from the PRCM module, and the PRCM module is then allowed to gate the clocks to the module. A slave module is said to be in IDLE state when its clocks are gated by the PRCM module.

Similarly, an idled slave module may need to be woken up because of a service request from a master module or because the slave module receives a wake-up event (for example, an interrupt or a direct memory access [DMA] request). In this situation the PRCM module enables the clocks for the module, and then signals the module to wake up by deasserting the IDLE request.

Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the slave module. This is done by setting the <*Module*>_SYSCONFIG. SIDLEMODE or <*Module*>_SYSCONFIG. IDLEMODE bit field, as described in Table 3-4. The behavior, identified in the IDLEMODE Bit Value column, must be configured by software.

Table 3-4. Module Idle Mode Settings

IDLEMODE Bit Value	Selected Mode	Description
0x0	Force-idle	The module unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully because it does not prevent loss of data at the time the clock is switched off.
0x1	No-idle	The module never acknowledges any IDLE request from the PRCM module. This mode is safe from a module point of view because it ensures the clocks remain active; however, it is not efficient from a power-saving perspective because it does not allow the PRCM module output clock to be shut off, and thus the power domain to be set to a lower power state.
0x2	Smart-idle	The module acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management.
0x3	Smart-idle wake-up-capable mode	The module acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management. The module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. The mode is relevant only if the appropriate module swakeup output(s) is implemented.

Note

- Smart-idle mode is the preferred mode of operation, while force-idle and no-idle modes are intended for debugging purposes.
- A slave module may support all or some of the idle modes listed in Table 3-4. See the power-management section in the module chapter to identify the supported idle modes.

The idle status of a slave module is indicated by the CM_<*Power domain*>_<*Module*>_CLKCTRL[x] IDLEST bit field in the PRCM module. Table 3-5 lists the possible idle statuses for a slave module.

Table 3-5. Slave Module Idle Status

IDLEST Bit Value	Idle Status	Description
0x0	Functional	The module is fully functional. The interface and functional clocks are active.
0x1	In transition	The module is performing a wake-up or a sleep transition.
0x2	Interface idle	The module interface clock is idled. The module may remain functional if using a separate functional clock.
0x3	Full idle	The module is fully idle. The interface and functional clocks are gated.

When configured in smart-idle mode, the slave module may acknowledge the IDLE request of the PRCM module based on the activity of its interface and/or functional clocks. To define which module clocks (that is, interface and/or functional) should be considered when responding to the PRCM module request, software must configure the <*Module*>_SYSCONFIG[x] CLOCKACTIVITY bit field.

The CLOCKACTIVITY setting is used internally by the module to determine the part of the module on which the conditions to acknowledge the PRCM module IDLE request are tested. As an example, if the functional clock must remain active when the module is in idle mode, the module must acknowledge a PRCM module IDLE request by considering only the interface clock gating conditions (that is, there is no pending activity on the interconnect).

Note

See the power-management section in the module chapter to identify whether this feature is configurable.

Using the CLOCKACTIVITY setting along with smart-idle mode ensures that the clock remains active for the module features that must remain available during the module idle mode. Table 3-6 describes the possible CLOCKACTIVITY settings for a module.

Table 3-6. Slave Module Clock Activity Settings

CLOCKACTIVITY Bit Value	Module Interface Clock	Module Functional Clock	Description
0x0	Gated	Gated	The interface and functional clocks are considered when generating the acknowledgment. This setting also means both clocks may be gated upon a PRCM module IDLE request.
0x1	Active	Gated	The interface clock is not shut down and is not considered when generating the acknowledgment to the PRCM module IDLE request. Only the functional clock is considered.
0x2	Gated	Active	The functional clock is not shut down and is not considered when generating the acknowledgment to the PRCM module IDLE request. Only the interface clock is considered.
0x3	Active	Active	The interface and functional clocks are not shut down. The module can acknowledge the IDLE request without checking the internal functions linked to its clocks.

Note

- The software configuration of the CLOCKACTIVITY settings may not be available for a given module. For some modules, the CLOCKACTIVITY settings can be hardwired.
- A slave module may support all or some of the CLOCKACTIVITY settings listed in Table 3-6.

See the power-management section in the specific module chapter to identify the supported idle feature and settings.

CAUTION

The PRCM module does not have any hardware means to read the CLOCKACTIVITY settings of the module. Software must ensure consistent programming between the CLOCKACTIVITY settings of the module and the clock-gating control bits in the PRCM module. The PRCM module must be configured (where software control is available) not to gate the module clock, which should remain active according to the CLOCKACTIVITY settings of the module.

For idle protocol management on the PRCM module side, the behavior of the PRCM module is configured in the CM_<Clock domain>_<module>_CLKCTRL[1:0] MODULEMODE bit field. Based on the configured behavior, the PRCM module asserts the IDLE request to the module unconditionally (that is, immediately when software requests) or through hardware control when the module idle conditions are satisfied. Table 3-7 describes the configurable behavior of MODULEMODE.

Table 3-7. Slave Module Mode Settings in PRCM

MODULEMODE Bit Value	Selected Mode	Description
0x0	Disabled	The PRCM module unconditionally asserts the module IDLE request. This request applies to the gating of the functional and interface clocks to the module. If acknowledged by the module, the PRCM module can gate all clocks to the module (that is, the module is completely disabled). It can react only to an asynchronous wake-up event (that is, a wake-up event that does not require the module functional clock to be active).

Table 3-7. Slave Module Mode Settings in PRCM (continued)

MODULEMODE Bit Value	Selected Mode	Description
0x1	Auto	This mode applies to a module when the PRCM module manages only its interface clock and not its functional clock. The PRCM module automatically asserts/deasserts the module IDLE request based on the clock domain transitions. If acknowledged by the module, the PRCM module can gate the interface clock to the module.
0x2	Enabled	This mode applies to a module when the PRCM module manages its interface and functional clocks. The functional clock to the module remains active unconditionally, while the PRCM module automatically asserts/deasserts the module IDLE request based on the clock domain transitions. If acknowledged by the module, the PRCM module can gate only the interface clock to the module.
0x3	Reserved	Not available

Note

The PRCM module may support all or some of the MODULEMODE module settings listed in Table 3-7. See the CM_<Clock domain>_<module>_CLKCTRL[1:0] MODULEMODE bit field description for the module to identify the supported settings.

Note

Modules, which can be configured with DISABLED or AUTO values of ModuleMode, cannot go from IDLE to DISABLED state if ModuleMode is changed from AUTO to DISABLED while module is in IDLE state. Once the CM_<Clock domain>_CLKSTCTRL[1:0] CLKTRCTRL has been set to SW_WKUP, then SW has to poll for PM_<Clock_domain>_PWRSTST[1:0] PowerStateSt = 0x03 and PM_<Clock_domain>_PWRSTST[20] InTransition = 0x00 before update MODULEMODE bit field.

Table 3-8 and Table 3-9 list the enabling conditions for the slave module clocks managed by the idle protocol.

Table 3-8. Slave Module Interface Clock Enabling Conditions

Relation		Condition
AND		Clock domain is ready.
	OR	Slave module idle status is 0x0 (fully functional).
		Slave module idle status is 0x1 (in transition).
		Slave module wake-up request is asserted.

Table 3-9. Slave Module Functional Clock Enabling Conditions

Relation		Condition
AND		Clock domain is ready.
	OR	Slave module idle status is 0x0 (fully functional).
		Slave module idle status is 0x1 (in transition).
		Slave module idle status is 0x2 (interface clock is idled).
		Slave module wake-up request is asserted.

The module clock domain must be ready for the optional clocks to the module, and any associated clock-enable control is asserted.

Note

A given clock can be used by more that one module. Clock-enabling conditions are then ORed together (that is, the clock is provided as soon as one of the enabling conditions is true). As a consequence, the clock is disabled only when all related enabling conditions are false.

Module wake-up request

In IDLE state, a slave module may have to wake up to generate an interrupt or a DMA request. This can be the result of an external request (for example, to the input/output [I/O] port of a general-purpose input/output [GPIO] module) or an internally generated event (for example, WD_TIMER time up). The slave module, with wake-up capability, sends a wake-up request to the PRCM module. The PRCM module then activates the module clocks and acknowledges the module wake-up request.

In IDLE state, some slave modules may require functional clock(s) to generate a wake-up event. Such requests are called synchronous wake-up events on the PRCM module side, while the events generated when the functional or interface module clocks are gated are called asynchronous wake-up events.

Note

See the power-management section in the module chapter to identify whether its wake-up event is synchronous or asynchronous.

The standby and idle clock-management protocols allow the configuration of the module-level clock-management interaction between the PRCM module and individual modules of the device. However, the PRCM module may not necessarily gate the clock to the module immediately after the module switches to standby or idle mode at the end of this interaction. This is because the same clock can be shared by other modules that are active and need this shared clock to complete their activity. As a result, the PRCM module provides a second level of clock management called the clock-domain level, as explained in Section 3.1.1.1.4, *Clock Domain*.

3.1.1.1.4 Clock Domain

A clock domain is a group of modules fed by clock signals controlled by the same clock manager in the PRCM module (see Figure 3-3). By gating the clocks in a clock domain, the clocks to all the modules belonging to that clock domain can be cut to lower their active power consumption (that is, the device is on and the clocks to the modules are dynamically switched to ACTIVE or INACTIVE [gated] state). Thus, a clock domain allows control of the dynamic power consumption of the device.

The device is partitioned into multiple clock domains and each clock domain is controlled by an associated clock manager within the PRCM module. This allows the PRCM module to activate and gate individually each device clock domain.

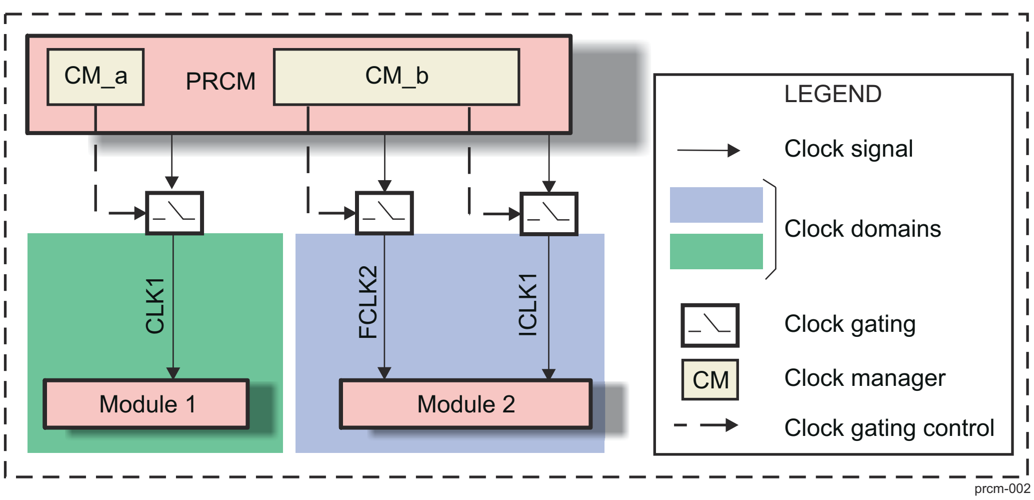


Figure 3-3. Generic Clock Domain

Figure 3-3 is an example of two clock managers: CM_a and CM_b. Each clock manager manages a clock domain. The clock domain of CM_b is composed of two clocks, a functional clock (FCLK2) and an interface clock (ICLK1), while that of CM_a consists of a clock (CLK1) that is used by the module as functional and interface clock. The clocks to Module 2 can be gated independently of the clock to Module 1, thus ensuring power savings when Module 2 is not in use.

The PRCM module lets software check the status of the clock domain functional clocks. The CM_<Clock domain>_CLKSTCTRL[x] CLKACTIVITY_FCLK/<Clock name>_FCLK bit in the PRCM module identifies the state of the functional clock(s) within the clock domain. Table 3-10 lists the two possible states of the functional clock.

Table 3-10. Clock Domain Functional Clock States

CLKACTIVITY Bit Value	Status	Description
0x0	Gated	The functional clock of the clock domain is inactive.
0x1	Active	The functional clock of the clock domain is running.

Table 3-11. Clock Domain Interface Clock States

CLKACTIVITY Bit Value	Status	Description
0x0	Gated	The interface clock of the clock domain is inactive.
0x1	Active	The interface clock of the clock domain is running.

All Clock Domains are summarized and described in Section 3.6.4, *Clock Domains.*

3.1.1.1.5 Clock Domain-Level Clock Management

The domain clock manager can automatically (that is, based on hardware conditions) and jointly manage the interface clocks within the clock domain. The functional clocks within the clock domain are managed through software settings.

A clock domain can switch between three possible states: ACTIVE, IDLE_TRANSITION (IDLEREQ), and INACTIVE (IDLE). Figure 3-4 shows the sleep and wake-up transitions of the clock domain between ACTIVE and INACTIVE states.

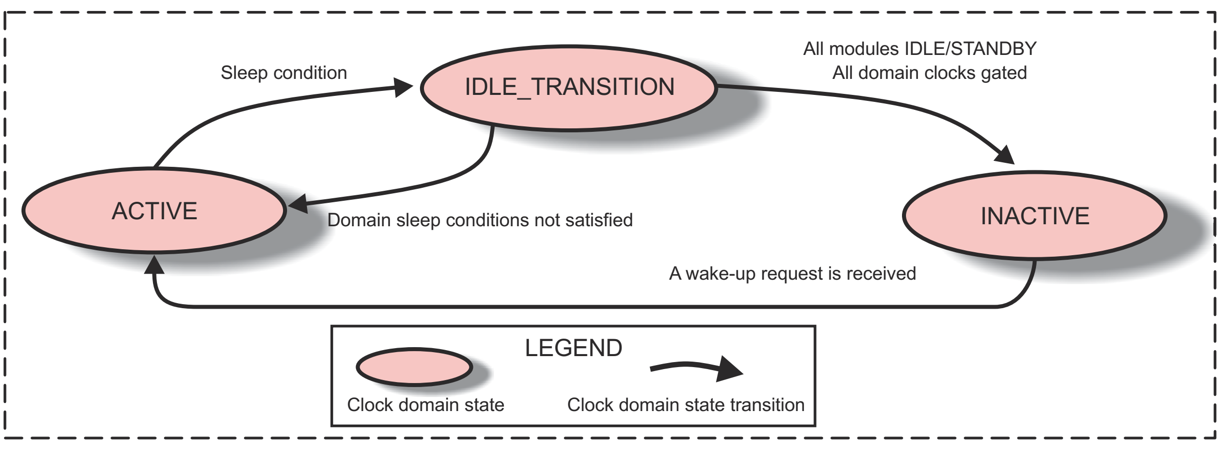


Figure 3-4. Clock Domain State Transitions

Table 3-12 defines the clock domain states.

Table 3-12. Clock Domain Clock States

State	Description
ACTIVE	• Every nondisabled slave module (that is, those whose MODULEMODE value is not set to disabled) is put out of IDLE state. • All mandatory functional clocks to the active slave modules (that is, not idled) of the clock domain are provided. • All interface clocks to the nondisabled slave modules in the clock domain are provided. • All mandatory functional and interface clocks to the active master modules (that is, not in STANDBY state) in the clock domain are provided. • Every enabled optional clock to the modules in the clock domain is provided.
IDLE_TRANSITION	This is a transitory state. • Every master module in the clock domain is in STANDBY state. • Every IDLE request to all the slave modules in the clock domain is asserted. • The functional clocks to the slave module in enabled state (that is, those whose MODULEMODE values are set to enabled) remain active. • Every enabled optional clock to the modules in the clock domain is provided.
INACTIVE	All clocks within the clock domain are gated. • Every slave module in the clock domain (that is, those whose MODULEMODE value is set to disabled or auto) is in IDLE state and set to disabled or auto mode. • Every master module in the clock domain is in STANDBY state. • Every optional functional clock in the clock domain is gated.

Each clock domain transition behavior is managed by an associated register bit field in the CM_<Clock domain>_CLKSTCTRL[x] CLKTRCTRL PRCM module.

Table 3-13 describes the clock transition mode settings of the clock domain.

Table 3-13. Clock Domain Clock Transition Mode Settings

CLKTRCTRL Bit Value	Selected Mode	Description
0x0	NO_SLEEP	A clock domain sleep transition is never initiated, regardless of the hardware conditions.
0x1	SW_SLEEP	A software-forced sleep transition. The transition is initiated when the associated hardware conditions are satisfied (see Table 3-15).
0x2	SW_WKUP	A software-forced clock domain wake-up transition is initiated, regardless of the hardware conditions identified in Table 3-14.
0x3	HW_AUTO	Hardware-controlled automatic sleep and wake-up transition is initiated by the PRCM module when the associated hardware conditions are satisfied (see Table 3-14 and Table 3-15).

Note

Depending on its characteristics, a clock domain may or may not support all the clock transition mode settings described in Table 3-13. See the clock domain clock management section of the specific clock domain to identify the supported clock transition mode settings.

3.1.1.1.6 Clock Domain HW_AUTO Mode Sequences

The sequence diagrams in Figure 3-5 through Figure 3-7 identify the PRCM module hardware-controlled enabling and gating of the functional and interface clocks to the module. They show the changes in the state of the module based on the changes to the clock domain state and module mode settings.

Figure 3-5 shows the behavior of a slave module receiving the interface and functional clocks and having two configurable module modes: disabled and enabled.

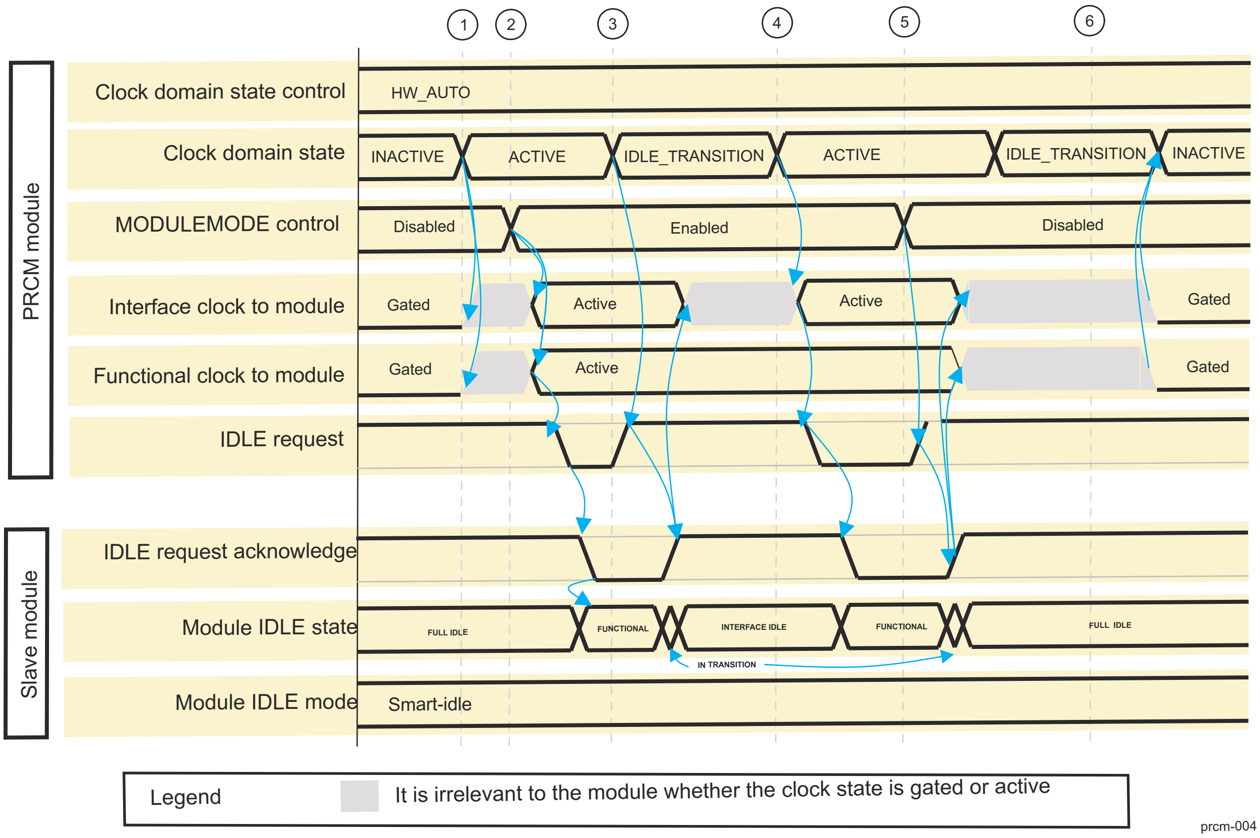


Figure 3-5. Clock Domain/Slave Module Clock-Management Interaction Sequence 1

Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL IDLE state and its functional and interface clocks are gated.

1. The clock domain wakes up and changes its state to ACTIVE. Because the MODULEMODE control is still disabled, this event has no effect on the state of the module. The functional and interface clocks can still be restarted automatically, based on the requirements of other modules sharing these clocks.
2. Software changes the module mode to ENABLED. The clocks to the module are automatically restarted. The PRCM module then deasserts a hardware IDLE request signal to the module. The module sends an IDLE request acknowledge to the PRCM module. The module is now effectively awake. The module IDLE state is functional.
3. The clock domain switches to IDLE_TRANSITION state. In turn, the PRCM module requests the module to go into IDLE state by asserting the IDLE request signal. When acknowledged, the clock to the module can be gated, depending on other modules sharing the same clock. The functional clock of the module remains enabled because the module is in enabled mode.
4. The clock domain does not have all conditions to complete the sleep transition and wakes up again. In turn, the interface clock to the module is automatically restarted, and then the module is put out of IDLE state.
5. Software disables the module. The PRCM module requests the module to go to IDLE state by asserting the IDLE request signal. When acknowledged, the clock to the module can be gated, depending on other modules sharing the same clock.
6. The clock domain switches to IDLE_TRANSITION state. When the sleep transition conditions of the clock domain are satisfied, the clocks (functional and interface) are gated. The clock domain then switches to INACTIVE state. This has no effect on the module, which is already in FULL IDLE state.

Figure 3-6 shows the behavior of the same slave module, receiving the interface and functional clocks, when the module mode is changed while the clock domain state is IDLE_TRANSITION.

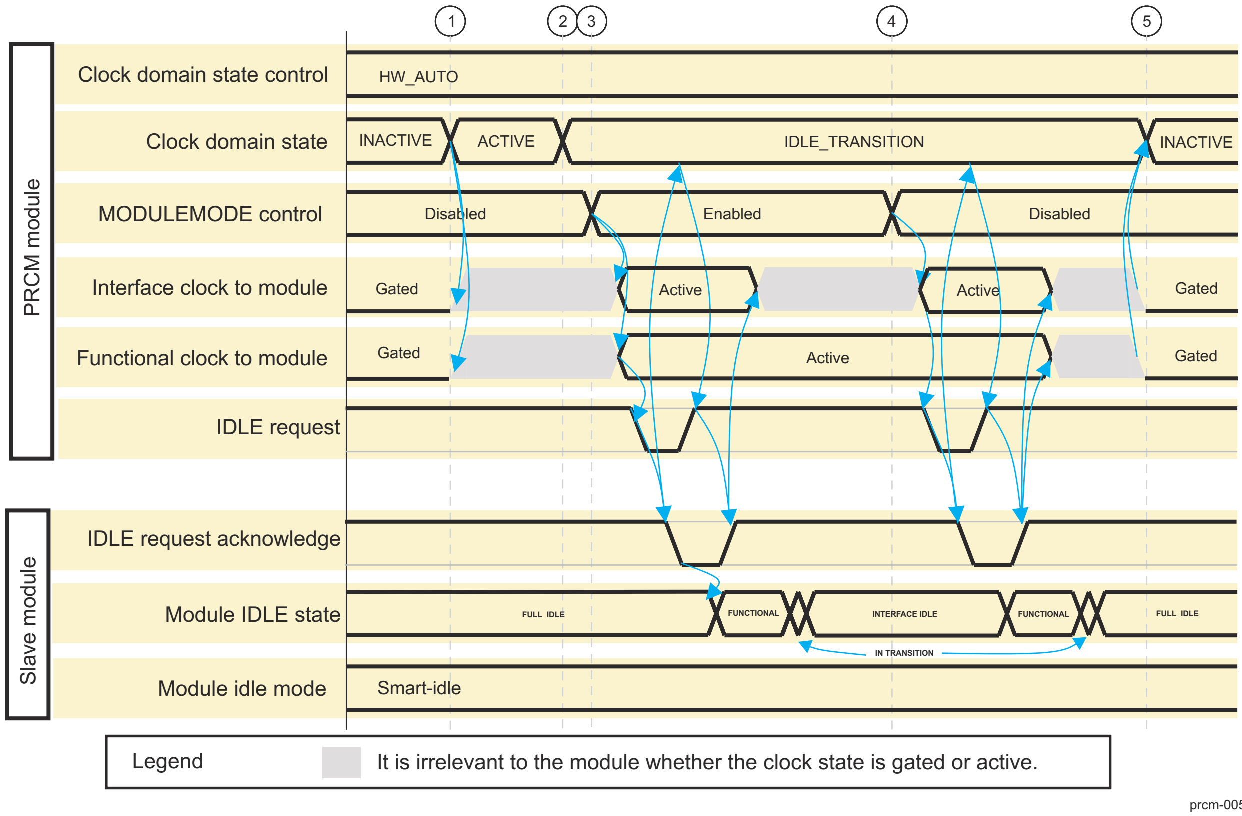


Figure 3-6. Clock Domain/Slave Module Clock-Management Interaction Sequence 2

Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL IDLE state and its functional and interface clocks are gated.

1. The clock domain wakes up and changes its state to ACTIVE. Because the MODULEMODE control is disabled, this event has no effect on the state of the module. The functional and interface clocks can still be restarted automatically, based on the requirements of other modules sharing these clocks.
2. The clock domain goes into the IDLE_TRANSITION state. Because the module mode control is disabled, this event has no effect on the module state.
3. Software changes the module mode to ENABLED. The clocks to the module are restarted automatically and then the module is put out of IDLE state. As soon as acknowledged, the module is requested to go back to IDLE state with gating of the interface clock only (that is, the INTERFACE IDLE state). The interface clock to the module can be gated, depending on other modules sharing the same clocks.
4. Software disables the module. The interface clock to the module is restarted automatically. The PRCM module requests the module to go out of IDLE state by asserting the IDLE request signal. As IDLE request acknowledge is de-asseted, PRCM set back the module to IDLE state. When acknowledged, the interface and the functional clocks to the module can be gated, depending on other modules sharing the same clock.
5. When the clock domain sleep transition conditions are satisfied and the functional and interface clocks are gated, the clock domain switches to INACTIVE state. This has no effect on the module, which is already in FULL IDLE state.

Figure 3-7 shows the behavior of a slave module receiving only interface clock and supporting the configurable auto module mode.

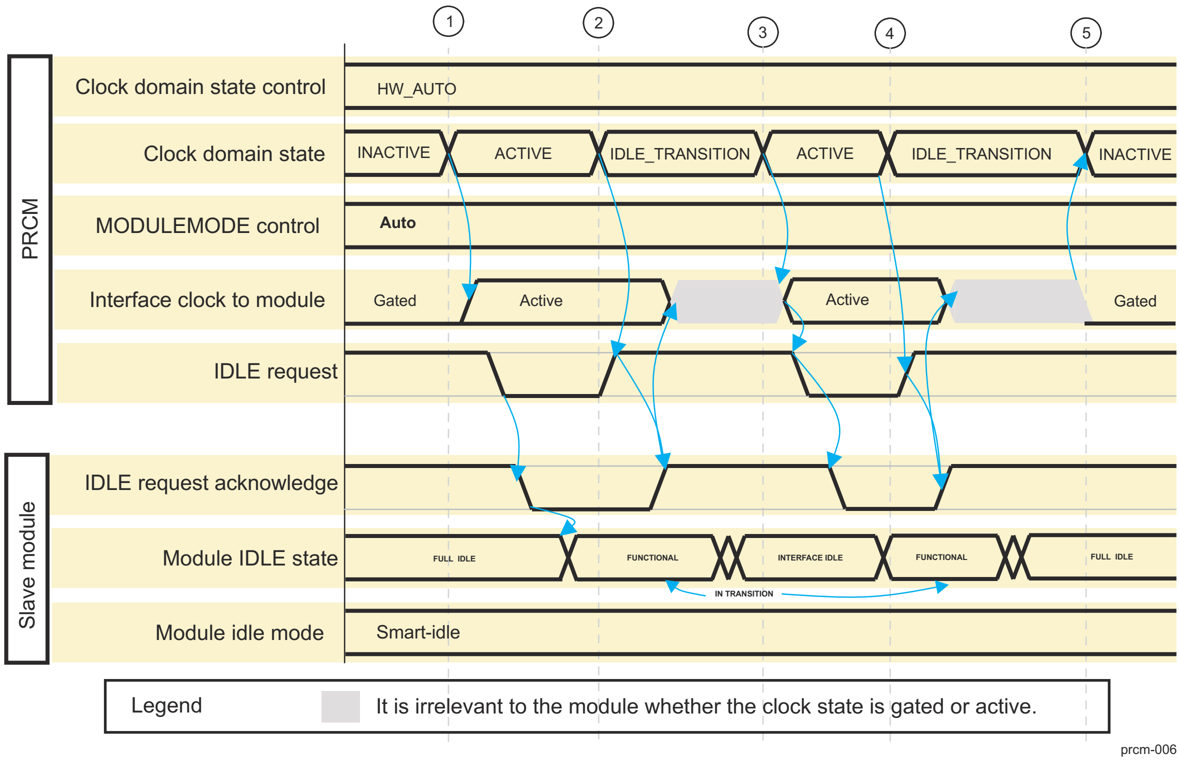


Figure 3-7. Clock Domain/Slave Module Clock-Management Interaction Sequence 3

Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL IDLE state and its interface clock is gated.

1. The clock domain wakes up and changes its state to ACTIVE. In turn, the interface clock to the module is automatically restarted. The PRCM module then deasserts a hardware IDLE request signal to the module. The module sends an IDLE request acknowledge to the PRCM module. The module is now effectively awake; that is, the module IDLE state is functional.
2. The clock domain switches to IDLE_TRANSITION state. The PRCM module requests the module to go to IDLE state by asserting the IDLE request signal. When acknowledged, the interface clock to the module can be gated, depending on other modules sharing the same clock.
3. The clock domain does not have all conditions to complete the sleep transition and wakes up again. In turn, the interface clock to the module is automatically restarted, and then the module goes out of IDLE state.
4. This step is the same as Step 2.
5. The clock domain has all conditions to complete the sleep transition. The module is in IDLE state and its clock is gated.

3.1.1.1.7 Clock Domain Sleep/Wake-up

The clock domain manager initiates a domain wake-up transition when the conditions listed in Table 3-14 are satisfied.

Table 3-14. Clock Domain Wake-Up Conditions

Relation	Condition
OR	The SW_WKUP clock transition mode for the clock domain is set (CLKTRCTRL = 0x2).
	At least one wake-up request is asserted by one of the modules of the clock domain.
	At least one dynamic dependency[(1)] from another clock domain is active.
	At least one static dependency[(1)] from another clock domain is active.
	At least one wake-up dependency[(1)] from a module in another clock domain is active.

(1) The dynamic, static, and wake-up dependencies are explained in Section 3.1.1.1.8, *Clock Domain Dependency*.

The clock domain manager initiates a domain sleep transition when the conditions listed in Table 3-15 are satisfied.

Table 3-15. Clock Domain Sleep Conditions

Relation		Condition
AND		All master modules in the clock domain are in STANDBY state.
		No wake-up request is asserted by any module of the clock domain.
		No dynamic domain dependency(1) from any other domain is active.
		No wake-up dependency(1) from any module in another domain is active.
		No static domain dependency(1) from any other domain is active.
	OR	The SW_SLEEP clock transition mode is set for the clock domain (CLKTRCTRL = 0x1).
		The HW_AUTO clock transition mode is set for the clock domain (CLKTRCTRL = 0x3).

(1) The dynamic, static, and wake-up dependencies are explained in Section 3.1.1.1.8, *Clock Domain Dependency*.

3.1.1.1.8 Clock Domain Dependency

A domain dependency is a binary relationship between two clock domains. A clock domain A is said to depend on a clock domain B when a module in clock domain B provides services to a module in clock domain A. As a result, clock domain B must be active when clock domain A is active so that the module in clock domain B is accessible by the module in clock domain A.

Dependency between two clock domains can also exist if one clock domain serves to ensure communication between two modules (for example, the clock domain of the device interconnect).

Thus, a clock domain can support the types of clock domain dependencies described in the following sections.

Table 3-16 and Table 3-17 detail all the domain dependencies:

- **NA/NA**: if no dependency can exist because no corresponding interconnect path exists in the device
- When cell is different than NA/NA, the cell contains two attributes:
  - First attribute for the presence and control method of a static dependency:
    - **SW**: Static dependency is controlled by a software bit. This is the most generic way of control. Depending on the use case and on latency requirement for accessing target domain, software can enable or not the static dependency. When not enabled, access to those domains is performed using dynamic dependencies.
    - **1**: Static dependency is always enabled (hard-wired). This is relevant for a domain that has an "exact standby" system initiator with the target domain being the domain containing the interconnect module to which the initiator is connected.
    - **0**: Static dependency is never enabled (hard-wired). This is relevant for domains that do not have strong access latency requirements, or for which a static dependency is not desired for specific reasons (for example, dependencies with emulation domain). Access to those domains is performed using dynamic dependencies.
    - **NA**: Nonapplicable (means domain has no system initiator able to access the other domain)
  - Second attribute for the presence and control method of a dynamic dependency:
    - **1..n**: Dynamic dependency is always enabled (hard-wired). The number of corresponding interconnect interfaces is also specified. This is relevant for most of the domain-to-domain direct interconnect connection.
    - **0**: Dynamic dependency is never enabled (hard-wired). This is relevant only when a static dependency is always enabled between same domains, or when the target domain cannot support the wakeup on access feature.
    - **NA**: Nonapplicable (means that both domains are not directly linked by an OCP interface)

Table 3-16. Device Domain Dependencies (Table 1)

Static/ dynamic dependencies from below domains to right-side domains	L4CFG	ATL	DMA	IPU2	L3INSTR	L3MAIN1	COREAON	CUSTEFUSE	DSP1	DSP2	DSS	EVE1	EVE2	GPU
VIP	0/na	0/na	0/na	0/na	0/na	1/0	0/na	0/na	0/na	0/na	0/na	0/na	0/na	0/na
L4CFG	na/n1a	na/na	0/1	na/na	0/0	0/35	0/6	0/1	0/na	0/na	na/na	na/na	na/na	na/na
DMA	SW/na	0/na	na/na	SW/na	0/na	1/0	0/na	0/na	0/na	0/na	SW/na	0/na	0/na	0/na
IPU2	SW/na	SW/na	0/na	na/na	0/na	SW/1	0/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na
DSP1	0/na	SW/na	0/na	SW/na	0/na	SW/3	0/na	0/na	na/na	SW/na	SW/na	SW/na	SW/na	SW/na
DSP2	0/na	SW/na	0/na	SW/na	0/na	SW/3	0/na	0/na	SW/na	na/na	SW/na	SW/na	SW/na	SW/na
L3INSTR	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na
L3MAIN1	0/1	0/na	0/na	0/1	0/0	na/na	0/na	0/na	0/1	0/1	0/2	0/1	0/1	0/1
DSS	0/na	0/na	0/na	0/na	0/na	1/0	0/na	0/na	0/na	0/na	na/na	na/na	na/na	0/na
EMU	0/na	0/na	0/na	0/na	0/na	0/1	0/na	0/na	0/na	0/na	0/na	0/na	0/na	0/na
EVE1	0/na	0/na	0/na	0/na	0/na	1/0	0/na	0/na	0/na	0/na	0/na	na/na	SW/na	0/na
EVE2	0/na	0/na	0/na	0/na	0/na	1/0	0/na	0/na	0/na	0/na	0/na	SW/na	na/na	0/na
GPU	0/na	0/na	0/na	0/na	0/na	1/0	0/na	0/na	0/na	0/na	0/na	na/na	na/na	na/na
IPU1	SW/na	SW/na	na/na	SW/na	na/na	SW/1	0/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na
IVA	0/na	0/na	0/na	0/na	0/0	1/0	0/na	0/na	0/na	0/na	0/na	na/na	na/na	0/na
GMAC	na/na	0/na	na/na	0/na	na/na	1/0	0/na	0/na	0/na	0/na	na/na	na/na	na/na	na/na
L3INIT	SW/na	0/na	0/na	0/na	0/na	1/0	0/na	0/na	0/na	0/na	0/na	na/na	na/na	0/na
L4PER1	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	0/na	0/1	na/na	na/na	na/na
L4PER2	na/na	0/1	na/na	SW/na	na/na	1/0	na/na	na/na	SW/na	SW/na	na/na	na/na	na/na	na/na
L4PER3	na/na	na/na	na/na	0/na	na/na	0/4	na/na	na/na	0/na	0/na	na/na	na/na	na/na	na/na
L4SEC	0/na	na/na	0/na	0/na	0/na	1/0	0/na	0/na	0/na	0/na	0/na	na/na	na/na	0/na
MPU	SW/na	na/na	0/na	SW/na	0/na	SW/1	0/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na
VPE	na/na	0/na	na/na	0/na	na/na	1/0	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na
PCIE	SW/na	SW/na	na/na	0/na	na/na	1/0	0/na	na/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na

Table 3-17. Device Domain Dependencies (Table 2)

Static/ dynamic dependencies from below domains to right-side domains	IPU1	CAM	EMU	IPU	IVA	GMAC	L3INIT	L4PER1	L4PER2	L4PER3	L4SEC	MPU	RTC	EMIF	VPE	WKUPAON	PCIe
VIP	0/na	na/na	0/na	0/na	0/na	0/na	0/na	0/na	0/na	0/na	0/na	0/na	0/na	SW/na	0/na	0/na	0/na
L4CFG	na/na	0/na	na/na	na/na	na/na	na/na	0/2	na/na	na/na	na/na	na/na	0/1	na/na	0/1	na/na	na/na	na/na
DMA	SW/na	0/na	0/na	SW/na	SW/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	0/na	na/na	SW/na	0/na	SW/na	SW/na
IPU2	SW/na	SW/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	0/na	na/na	SW/na	SW/na	SW/na	SW/na
DSP1	0/na	SW/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	0/na	na/na	SW/na	SW/na	SW/na	SW/na
DSP2	0/na	SW/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	0/na	na/na	SW/na	SW/na	SW/na	SW/na
L3INSTR	na/na	na/na	0/0	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na
L3MAIN1	0/1	0/0	0/na	0/na	0/2	0/na	0/na	0/1	0/4	0/1	0/4	0/na	na/na	0/2	na/na	0/1	0/2

Table 3-17. Device Domain Dependencies (Table 2) (continued)

Static/dynamic dependencies from below domains to right-side domains	IPU1	CAM	EMU	IPU	IVA	GMAC	L3INIT	L4PER1	L4PER2	L4PER3	L4SEC	MPU	RTC	EMIF	VPE	WKUPAON	PCIe
DSS	0/na	0/na	0/na	na/na	SW/na	na/na	0/na	0/na	na/na	na/na	0/na	0/na	na/na	SW/na	na/na	0/na	na/na
EMU	0/na	0/na	na/na	0/na	0/na	na/na	0/na	0/na	na/na	na/na	0/na	0/0	na/na	0/na	na/na	0/na	na/na
EVE1	0/na	na/na	na/na	na/na	SW/na	na/na	na/na	na/na	na/na	na/na	na/na	0/na	na/na	SW/na	na/na	na/na	na/na
EVE2	0/na	na/na	na/na	na/na	SW/na	na/na	na/na	na/na	na/na	na/na	na/na	0/na	na/na	SW/na	na/na	na/na	na/na
GPU	0/na	0/na	0/na	na/na	SW/na	na/na	0/na	0/na	na/na	na/na	0/na	0/na	na/na	SW/na	na/na	0/na	na/na
IPU1	na/na	SW/na	na/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	na/na	na/na	SW/na	SW/na	SW/na	SW/na
IVA	0/na	0/na	0/na	na/na	na/na	na/na	0/na	0/na	na/na	na/na	0/na	0/na	na/na	SW/na	na/na	0/na	na/na
GMAC	0/na	na/na	na/na	0/na	na/na	na/na	na/na	na/na	SW/na	na/na	na/na	0/na	na/na	SW/na	na/na	na/na	na/na
L3INIT	0/na	0/na	0/na	na/na	SW/na	na/na	na/na	SW/na	na/na	SW/na	SW/na	0/na	na/na	SW/na	na/na	SW/na	na/na
L4PER1	0/na	na/na	na/na	0/2	na/na	na/na	0/2	na/na	na/na	na/na	0/4	na/na	na/na	na/na	na/na	na/na	na/na
L4PER2	SW/na	na/na	na/na	0/1	na/na	0/1	0/1	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na
L4PER3	0/na	0/3	na/na	0/4	na/na	na/na	0/8	na/na	na/na	na/na	na/na	na/na	0/1	na/na	0/1	na/na	na/na
L4SEC	0/na	0/na	0/na	na/na	0/na	na/na	0/na	SW/na	na/na	na/na	na/na	0/na	na/na	SW/na	na/na	0/na	na/na
MPU	SW/na	SW/na	0/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	na/na	na/na	SW/2	SW/na	SW/na	SW/na
VPE	0/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	na/na	SW/na	na/na	na/na	na/na	SW/na	na/na	na/na	na/na
PCIE	SW/na	SW/na	na/na	SW/na	SW/nac	SW/na	SW/na	SW/na	SW/na	SW/na	SW/na	0/na	na/na	SW/na	SW/na	na/na	na/na

When a static dependency is hardware-coded between two domains directly linked by one or several interconnect interfaces, then the corresponding dynamic dependency is useless.

Emulation domain (DAP initiator) has no static dependency with any other domain. It has, however, a dynamic dependency with the L3_MAIN2 interconnect. A domain that can access an emulation domain does not have static or dynamic dependency with it.

Domain dependencies are chosen such that any access (even from EMU/DAP) towards a nondisabled target is always completed normally. Using static dependencies allows having minimal access latencies by keeping necessary domains on whenever the initiator is not standby. This may be at the expense of additional power consumption because some domains may stay on while not in use for a long time. By disabling static dependencies, applicative access is still completed normally by waking up, if applicable, the necessary domain on the path from the initiator to the target. Power consumption can be optimized at the expense of additional access latencies.

Note

Once the CM_<Clock domain>_CLKSTCTRL[1:0] CLKTRCTRL has been set to SW_WKUP, then SW has to poll for PM_<Clock_domain>_PWRSTST[1:0] PowerStateSt = 0x03 and PM_<Clock_domain>_PWRSTST[20] InTransition = 0x00 before update MODULEMODE bit field.

3.1.1.1.8.1 Static Dependency

If clock domain A has a master module that can access a slave module in clock domain B, then clock domain A can have a static dependency with clock domain B. Similarly, a static dependency can also exist between domain A and B if domain B conveys the transactions from a domain A module toward a module in any other domain. For example, CD_DSP can have a static dependency with CD_L3_MAIN1 because this domain has a level 3 (L3) interconnect to carry the transactions from the digital signal processor (DSP) module.

This static dependency consists of forcing clock domain B to stay active as long as there is at least one master module of clock domain A that is not in STANDBY state. If clock domains A and B are initially in GATED state, then clock domain B becomes active as soon as clock domain A becomes active when a wake-up request from the master module is received by the PRCM module.

Similarly, as a result of the static dependency, clock domain B can be gated only if all the master modules of clock domain A that can access the slave modules in clock domain B are in STANDBY state.

The static dependency between a source clock domain and a destination clock domain is configured in the PRCM module by setting the CM_<Source Clock domain>_STATICDEP[x] <Destination Clock domain>_STATDEP bit. As a result, the source clock domain forces the destination clock domain to become active and stay active as long as the source clock domain is active.

The destination domain must be put into forced wake-up (CM_<X>_CLKSTCTRL[1:0] CLKTRCTRL = SW_WKUP) before changing a configurable static dependency.

3.1.1.1.8.2 Dynamic Dependency

When clock domains A and B contain modules directly linked to a common device interconnect, these clock domains can have a dynamic dependency.

A dynamic dependency consists of forcing clock domain B to stay active as long as a module from clock domain A is communicating with the module in clock domain B through the interconnect. Clock domain B becomes active as soon as the communication is initiated. This is automatically managed by the PRCM module by monitoring the communication on the interconnect between the modules of the two clock domains.

Similarly, the inverse condition of this dependency can be stated: Clock domain B can be inactive only if there has not been transactions from clock domain A to clock domain B, identified as a sliding window duration on the interconnect activity status.

The size of the sliding window is based on the number of cycles of a prescaled level 4 (L4) clock whose frequency is configured by setting the CM_DYN_DEP_PRESCAL[5:0] PRESCAL bit field. The prescaled clock frequency is given as:

Prescaled clock frequency = L4 interface clock frequency / (PRESCAL + 1)

The size of the sliding window is fixed by setting the CM_<Clock domain>_DYNAMICDEP[27:24] WINDOWSIZE bit field. It is given as:

Sliding window duration = WINDOWSIZE × Period of Prescaled clock cycle

Figure 3-8 is an example of the sliding window duration equal to eight clock cycles of an L4 clock when PRESCAL is set to 3 and WINDOWSIZE is set to 2.

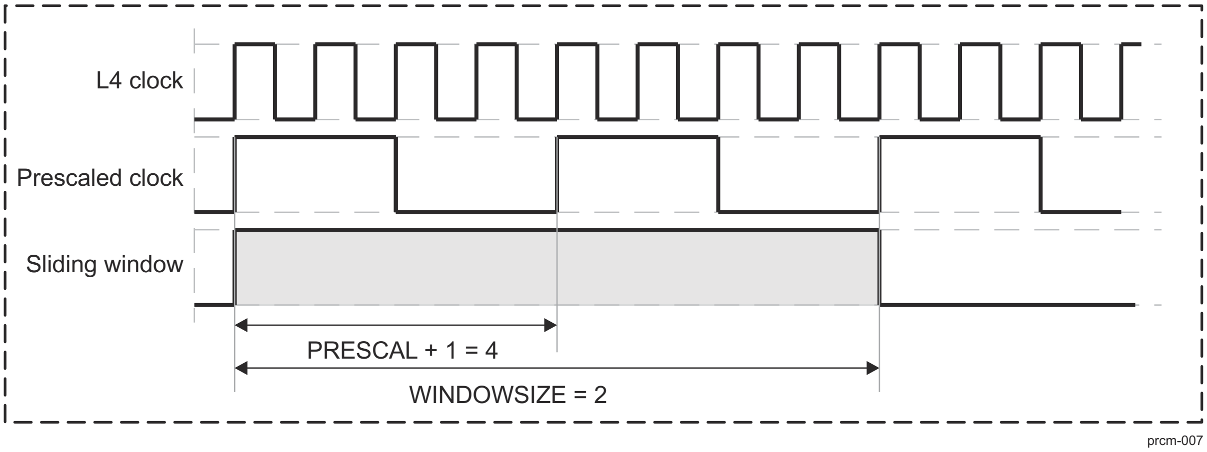


Figure 3-8. Sliding Window for Dynamic Dependency

This dynamic dependency is also referred to as the autosleep/autowakeup feature.

Note

- The static dependency between two clock domains can be configured by software (PRCM module registers) or hardwired in the PRCM module.
- The dynamic dependency between two clock domains is hardwired in the PRCM module.

A dynamic dependency is said to be active when both of the following conditions are met:

- There has been one or more transaction on the interconnect within the sliding window duration.

Otherwise, a dynamic dependency is said to be inactive.

The dynamic dependency between a source clock domain and a destination clock domain can be read in the PRCM module from the corresponding read-only CM_<Source Clock domain>_DYNAMICDEP[x] <Destination Clock domain>_DYNDEP bit.

Note

It is recommended to use dynamic dependencies. They give better power results. Static dependencies should be rarely used (in some cases they can be used as they give shorter latency for a system initiator to access slave).

3.1.1.1.8.3 Wake-Up Dependency

A wake-up dependency is a dependency between the clock domain of a module that owns one or several wake-up signals toward the clock domain of another module needed to service the associated wake-up event. As a result of this dependency, the wake-up event to a module activates not only its clock domain but also the clock domain of the servicing module.

Note

To ensure that the clock domain of the servicing module remains active, the wake-up signal that triggers a wake-up dependency stays active as long as the source of the event is not serviced.

Wake-up dependencies allow acceleration of the wake-up transition of multiple domains needed to service the wake-up event by initiating their transition in parallel. The static and dynamic dependencies can allow the wake-up of related domains, but the complete wake-up transition of all the associated domains is slower because of the sequential cascading of their wake-up transitions.

In the device, the source event of the wake-up signal to a slave module can be either of following types:

- Interrupt request to the microprocessor unit (MPU), DSP, or image processor unit (IPU) interrupt controller (INTC)
- DMA request to a DMA controller

Upon wake-up by these types of wake-up events, and for as long as they remain asserted, the PRCM module takes the following actions:

- The power domain of the servicing module (for example, MPU, DSP, IPU, or DMA) is forced to POWER ON state and the clock domain becomes active.
- The power domain of the device interconnect between the servicing module and the module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.
- The power domain of the slave module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.
- The slave module originator of the wake-up event is switched from IDLE to ACTIVE state.

On assertion of a wake-up event of a stand-alone master module, and as long as it remains asserted, the PRCM module takes the following action:

- The power domain of the master module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.

Note

For slave modules, the static and dynamic dependencies of a clock domain are not affected by its wake-up dependency settings. For master modules, the static dependencies are not affected.

Hence, in addition to the activation of the clock domains previously described, all clock domains associated by static dependencies are also activated.

However, the clock domains associated with the wake-up clock domain through dynamic dependencies are activated only if a transaction is initiated to these clock domains.

For each wake-up signal coming from a slave module, the type of the corresponding event can be configured in the PM_<Power domain>_<Originator Module> _WKDEP[x] WKUPDEP_<Originator Module>_[IRQ/DMA]_<Servicing Module> bit of the PRCM module, where <Power domain> is the name of the power domain of the originator module of the wake-up event identified as <Originator Module>. Servicing Module refers to the module servicing the wake-up event.

Note

When only one event type is associated with the wake-up signal of a slave module, the wake-up dependency (WKUPDEP) for the module clock domain is not configurable and may be hardwired in the PRCM module.

For the master modules, there is no configurable wake-up dependency. Their power domain is switched on and their clock domain is activated by the PRCM module when they assert their wake-up signal.

3.1.1.2 Power Management

The PRCM module manages the switching on and off of the power supply to the device modules. To minimize device power consumption, the power to the modules can be switched off when they are not in use. Independent power control of sections of the device lets the PRCM module turn on and off specific sections of the device without affecting other sections.

3.1.1.2.1 Power Domain

A power domain is a section (that is, a group of modules) of the device with an independent and dedicated power manager (see Figure 3-9). A power domain can be turned on and off without affecting other parts of the device.

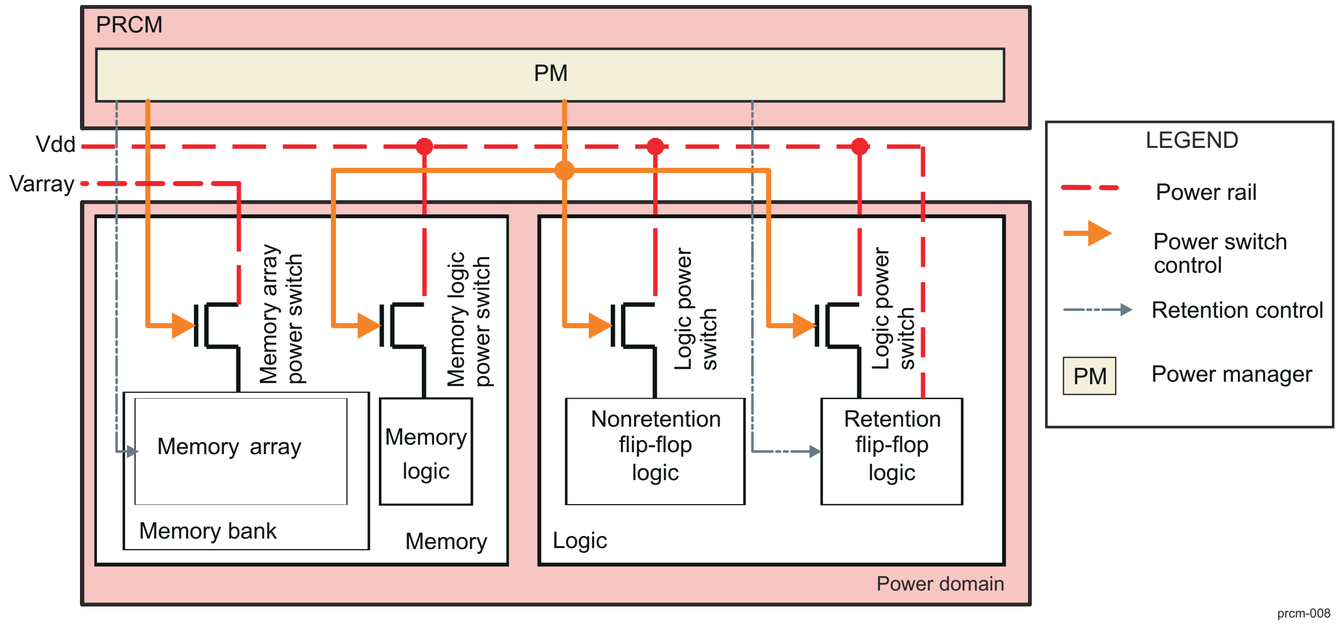


Figure 3-9. Generic Power Domain

To minimize device power consumption, the modules are grouped into power domains. A power domain can be split into logic and memory areas.

The memory area contains two entities:

- Memory bank: Composed of memory arrays. It is powered by a dedicated voltage rail and an associated power switch (for example, Varray and memory array power switches).
- Memory logic: Powered by the same voltage source as the logic area of the power domain, but has its dedicated power switch (for example, Vdd and memory logic power switches)

The logic area in the power domain can also be split between retention flip-flops (RFFs) and nonretention flip-flops (DFFs).

Table 3-18 lists the possible states and substates of the logic area in a power domain.

Table 3-18. States of a Logic Area in a Power Domain

State	Substate	Description
ON	ON-ACTIVE	Logic is fully powered and at least one enclosed clock domain is active.
	ON-INACTIVE	Logic is fully powered and all enclosed clock domains are idled.
RETENTION	CSWR (close switch retention)	Logic is fully powered and all enclosed clock domains are idled.
OFF		Logic power switches are off. All the logic (DFF and RFF) is lost except for the context, which has been saved in the scratchpad memory of an always-on power domain. Vdd can be set to 0 V if all associated power domains are in this state.

RETENTION state is useful for quickly switching to low-power idle mode (in which the domain clocks are gated and the domain voltage is less than the on-voltage level) without losing the context, and then quickly switching back to ON-ACTIVE state when necessary. In RETENTION state, power consumption is less than in ON power state.

The behavior of the memory array power switch and memory logic power switch can be selected through software settings in the PRCM module or can be hardwired. Once the behavior is selected, the PRCM module hardware automatically handles these elements to ensure correct power transition sequencing between the power domain states.

Software can also initiate power state changes of the memory array when the associated power domain is in ON power state. This allows the memory array to be turned off and on as needed.

The memory area can be configured to any of the power states listed in Table 3-19.

Table 3-19. States of a Memory Area in a Power Domain

State	Description
ON	The memory array is powered and fully functional.
RETENTION	The memory array is fully powered, but memory is not accessible. The array can be put into retention through an applicable direct retention control signal. Data in memory are always retained.
OFF	The memory array is powered down. Data in memory are lost.

3.1.1.2.2 Module Logic and Memory Context

In case of a power state transition in the logic or memory areas, the context of the module may no longer be valid. This can also be the case when the domain resets are asserted by the device. A specific RM_<Clock Domain Name>_<Module Name>_CONTEXT register provides the status of the device logic and memory context.

- The module logic context consists of simple flip-flops (DFFs) if the module has no logic RFFs.
- If the module has logic retention (full or partial), it is assumed that the context consists of: only RFFs, only DFFs or both RFFs and DFFs.

Note

The display subsystem is an exception where the status of DFF and RFF context is given, because only HMDI keys are retained, while most of the display subsystem is not retained.

These context status bits must be cleared by software.

3.1.1.2.3 Power Domain Management

The power manager associated with each power domain is assigned the task of managing the domain power transitions. It ensures that all hardware conditions are satisfied before it can initiate a power domain transition from a source to a target power state (for example, from ON-ACTIVE state to CSWR RETENTION state).

The hardware condition for power domain transition from ON-ACTIVE to any other transition state is:

- All clock domain managers in the power domain are in IDLE state.

Figure 3-10 shows all possible power domain state transitions.

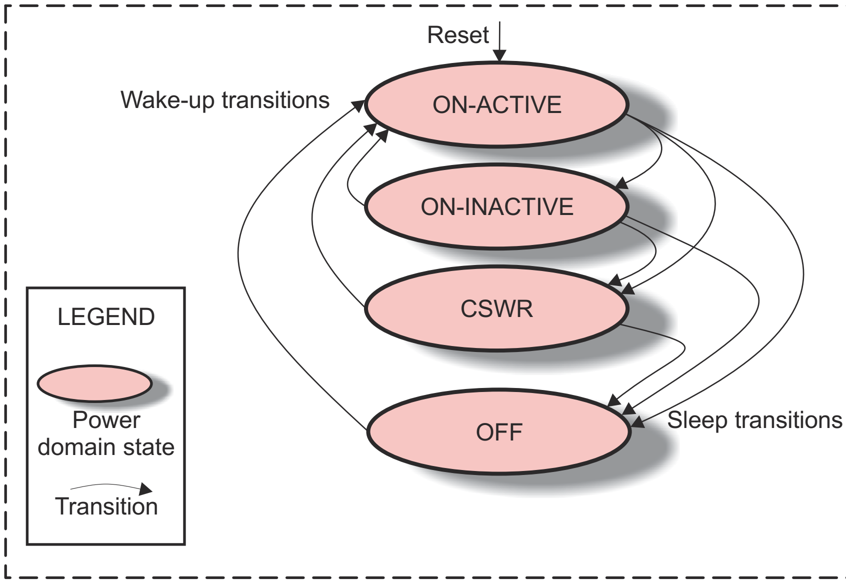


Figure 3-10. Power Domain Transitions

Successive power-down transitions can be performed by lowering the power state from ON-ACTIVE to ON-INACTIVE to RETENTION, and then to OFF and LOWPOWERSTATECHANGE, as long as the hardware condition is satisfied.

However, the power domain wake-up transition from any low-power state (ON-INACTIVE or CSWR) to ON-ACTIVE state is always direct.

The power domain manager initiates a power domain wake-up transition when the conditions listed in Table 3-20 are satisfied.

Table 3-20. Power Domain Wake-Up Conditions

Relation		Condition
AND		Voltage domain is on.
	OR	There is at least a wake-up condition for one enclosed functional clock domain.
		There is a request for clock generation or distribution enclosed in the power domain.
		There is a PRCM module service request (applicable only to power domains, including PRCM module logic).

The power domain manager initiates a domain sleep transition when the conditions listed in Table 3-21 are satisfied.

Table 3-21. Power Domain Sleep Conditions

Relation	Condition
AND	All functional clock domains enclosed in the power domain are idled.
	All clock generation or distribution enclosed in the power domain is quiet, and corresponding input clocks are gated. For example, DPLL, if present, must be in stop mode.
	There is no PRCM module service request (applicable only to power domains, including PRCM module logic).

Table 3-22 lists the control and status features of the PRCM module power domain.

Table 3-22. Power Domain Control and Status Registers

Register/Bit Field	Type	Description
PM_<Power domain>_PWRSTCTRL[1:0] POWERSTATE	Control	Selects the target power state of the power domain among OFF, ON-ACTIVE, ON-INACTIVE, and RETENTION
PM_<Power domain>_PWRSTCTRL[x] LOWPOWERSTATECHANGE	Control	Power state change request when domain has already performed a sleep transition. Allows going into deeper low-power state without waking up the power domain.
PM_<Power domain>_PWRSTCTRL[2] LOGICRETSTATE	Control	Selects whether the power domain logic is in CSWR RETENTION state when the domain transitions to RETENTION state
PM_<Power domain>_PWRSTCTRL[x] <memory bank>_RETSTATE	Control	Selects whether the memory bank in the power domain is in ON, or RETENTION state when the power domain is in RETENTION state. The memory bank cannot be in ON state when the power domain is in RETENTION state.
PM_<Power domain>_PWRSTCTRL[x] <memory bank>_ONSTATE	Control	Selects whether the memory bank is in ON, RETENTION or OFF state when the power domain is in ON state
PM_<Power domain>_PWRSTST[1:0] POWERSTATEST	Status	Identifies the current state of the power domain. It can be OFF, RETENTION, ON-INACTIVE, or ON-ACTIVE.
PM_<Power domain>_PWRSTST[2] LOGICSTATEST	Status	Identifies the current state of the logic area in the power domain. It can be OFF or ON.
PM_<Power domain>_PWRSTST[20] INTRANSITION	Status	Identifies whether a power state transition in the power domain is in progress or there is no ongoing transition
PM_<Power domain>_PWRSTST[x] <memory bank>_STATEST	Status	Identifies the current power state of the memory bank in the power domain. It can be OFF, RETENTION, or ON.
PM_<Power domain>_PWRSTST[25:24] LASTPOWERSTATEENTERED	Status	Identifies the last (previous) power state of the power domain. It can be OFF, RETENTION, ON-INACTIVE, or ON-ACTIVE.

3.1.1.3 Voltage Management

The PRCM module do not provide controls over the Voltage management. All OPP changes will be handled by Application software directly without any PRCM intervention.

3.1.1.3.1 Voltage Domain

A voltage domain is a section of the device supplied by a dedicated voltage source (that is, an internal LDOs or external switch mode power supply [SMPS]). A voltage domain may or may not be controlled by the PRCM module.

The voltage managers in the PRCM module is one type:

- Dynamically configurable by software to scale the domain voltage level to specific values within the operational voltage range of the device. This is called adaptive voltage scaling (AVS).

Figure 3-11 shows a voltage domain.

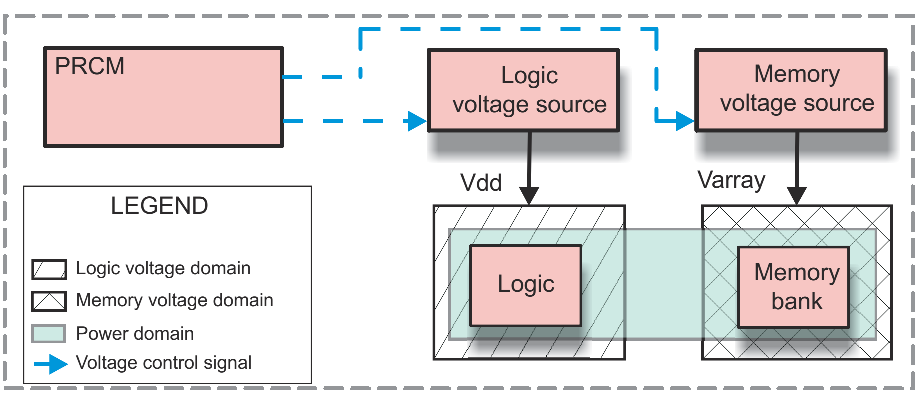


Figure 3-11. Generic Voltage Domain

By partitioning the device into independent voltage domains, different operating voltages can be assigned to the different sections of the device (that is, a group of modules or memory banks). The independent voltage control allows voltage scaling of device subsections to ensure that each module or memory bank operates at the optimized operating voltage level based on the application performance requirements. Similarly, when a memory bank is not in use, it can be switched to retention voltage levels to ensure power savings.

3.1.1.3.2 Voltage Domain Management

Figure 3-12 shows the different voltage control paths available within a generic logic voltage management block to control the voltage supply to the logic voltage domains of the device.

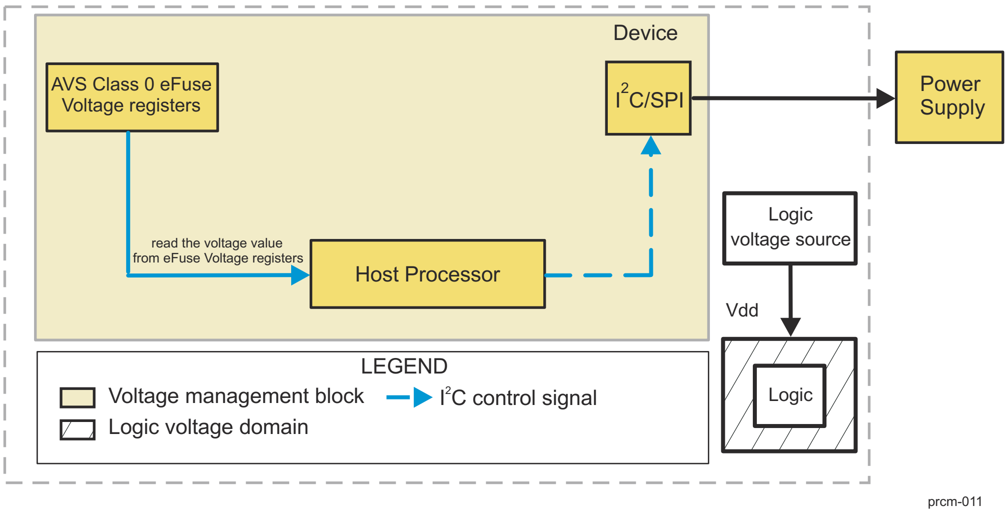


Figure 3-12. Generic Logic Voltage Management

Figure 3-13 shows the voltage control path available within a generic memory voltage management block to control the voltage supply to the memory voltage domains of the device.

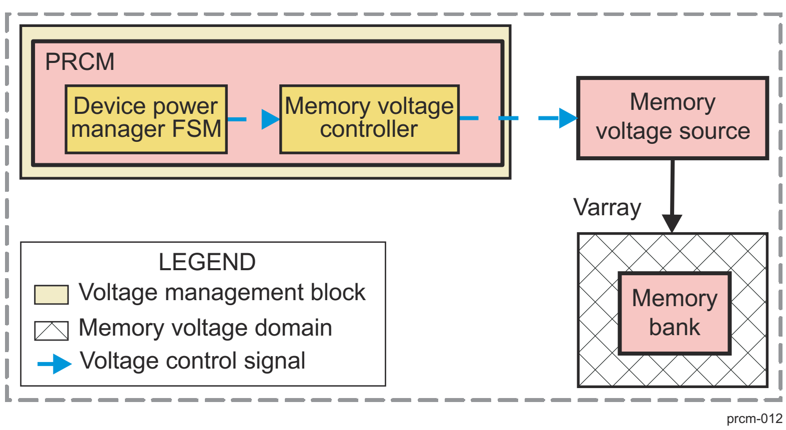


Figure 3-13. Generic Memory Voltage Management

The PRCM hardware supports automatic scaling down of the memory array supply whenever the memory domains transition to RETENTION power state. The device power manager FSM manages the voltage scaling of memory voltage domains through the memory voltage controller (or LDO).

3.1.1.3.3 AVS Overview

SmartReflex is a power-management technique used to control the operating voltage of a device to reduce its active power consumption.

With SmartReflex, the power supply voltage is adapted to the silicon performance in one way:

- Statically adapted to the manufacturing process of a given device

SmartReflex achieves optimal performance/power trade-off for all devices across the technology process spectrum.

The static correction of the device voltage level (see Figure 3-14) is based on the desired performance level and silicon performance characteristics of the device. As a result of process dispersion, each die has its specific silicon performance. The range of the process distribution defines the weak devices (low-performance silicon) and the strong devices (high-performance silicon).

A weak device is a device with the lowest performance tolerated for a process distribution; that is, at the typical voltage, the inherent maximum frequency is the lowest frequency of the chip distribution. Considered as the worst case, weak devices are used to constrain the target frequency of all the chips (OPP definition).

A strong device is a device with the highest performance tolerated for a process distribution. The inherent maximum frequency at the typical voltage is greater than the targeted frequency.

Figure 3-14 shows that with the SmartReflex voltage-control architecture, it is possible to compensate for the device silicon characteristics and obtain optimal performance characteristics. Based on the device characteristics, the device voltage level can be adjusted for specific performance level.

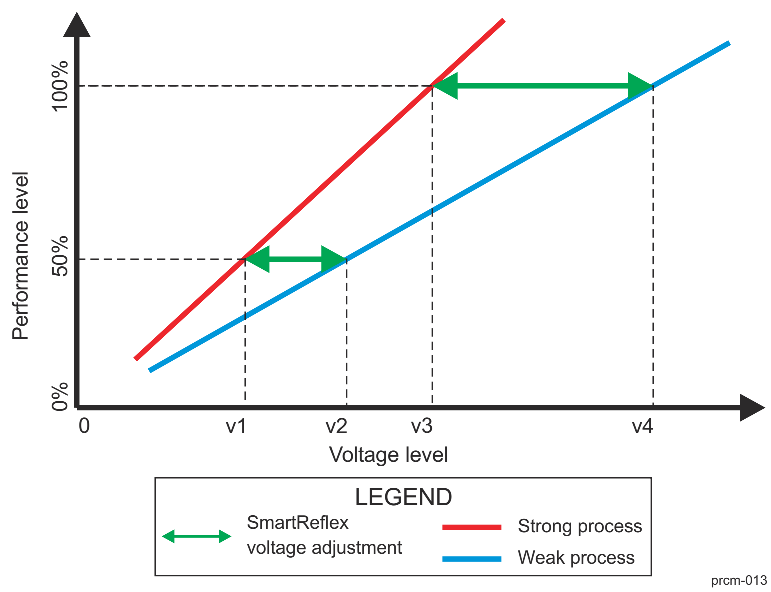


Figure 3-14. SmartReflex Static Voltage Adjustment

Figure 3-15 is a functional overview of the SmartReflex voltage-control architecture of the device connected to an external power IC.

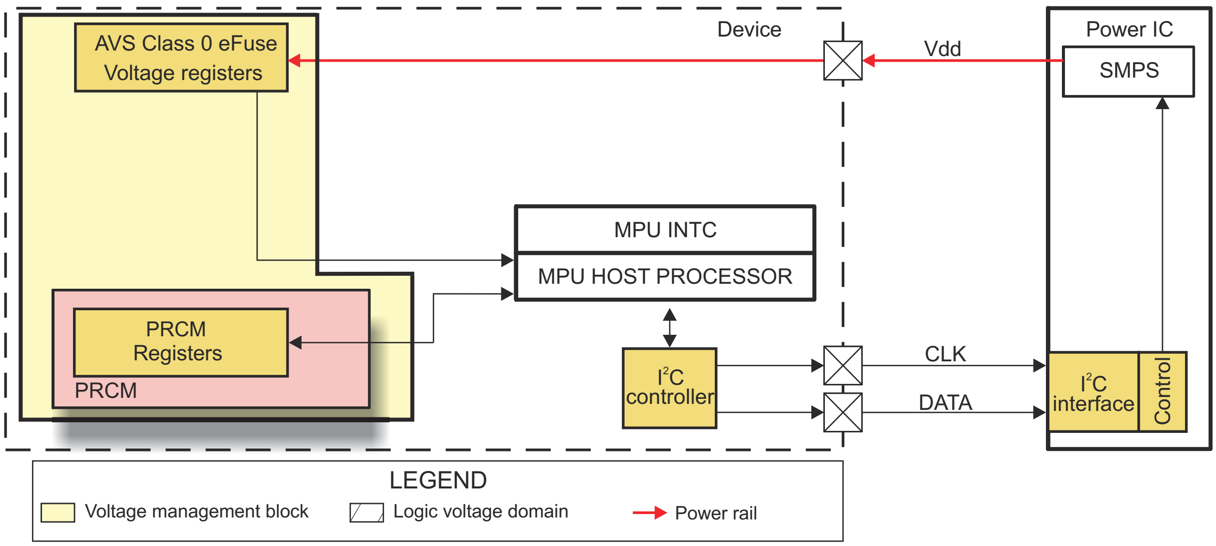


Figure 3-15. SmartReflex Voltage Control Functional Overview

SmartReflex voltage control consists of the following modules:

- MPU HOST Processor
- I^{2}C interface
- SMPS

The SmartReflex module senses the frequency of internal ring oscillator and generates an error value that identifies the difference between the desired optimal voltage and the actual value and outputs this directly on dedicated hardware interface. This error value is set in an internal register (for software read, if necessary) and is also generated an interrupt to indicate error values outside acceptable limits.

The processor converts the error value to a command that defines the change in the output voltage required to bring it to the desired voltage level.

The device supports one operational mode for SmartReflex voltage control:

- Class 0: Manufacturing Test Calibration

3.1.1.3.3.1 AVS Class 0 (SmartReflex™) Voltage Control

Adaptive Voltage Scaling (AVS) Class 0 (also referred to as SmartReflex™) is a procedure for lowering the voltage on certain device power rails. AVS Class 0 attempts to normalize the power consumption across all devices. The optimal voltage for each AVS supported rail of each device is determined after analysis in the factory. This value is written in the device eFuse where it can be read through dedicated registers. These registers reside in the control module. For more information, see Section 18.4.6.12, *AVS Class 0 Associated Registers*, in Chapter 18, *Control Module*.

3.1.2 Power-Management Techniques

The following sections describe the power-management techniques supported by the device.

Note

The values in Figure 3-16 through Figure 3-18, which show the power-management techniques, are hypothetical. They do not represent valid test results on the device.

3.1.2.1 Standby Leakage Management

Standby leakage management (SLM) is a power-management technique that reduces standby power consumption by reducing power leakage.

With SLM, the device switches into low-power system modes automatically or in response to user requests during system standby (that is, in situations when no application is started and system activity is negligible or limited).

When applying SLM, the system remains in the lowest static power mode compatible with the system response time requirement.

This technique trades static power consumption for wake-up latency.

3.1.2.2 Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling (DVFS) consists of minimizing the idle time of the system. The DVFS technique uses dynamic selection of the optimal operating frequency and voltage to allow a task to be performed in the required length of time. This reduces the active power consumption (power consumed while executing a task) of the device while still meeting task requirements.

Note

The values in Figure 3-16 are hypothetical. They are meant only to clarify the concept and do not represent valid test results on the device.

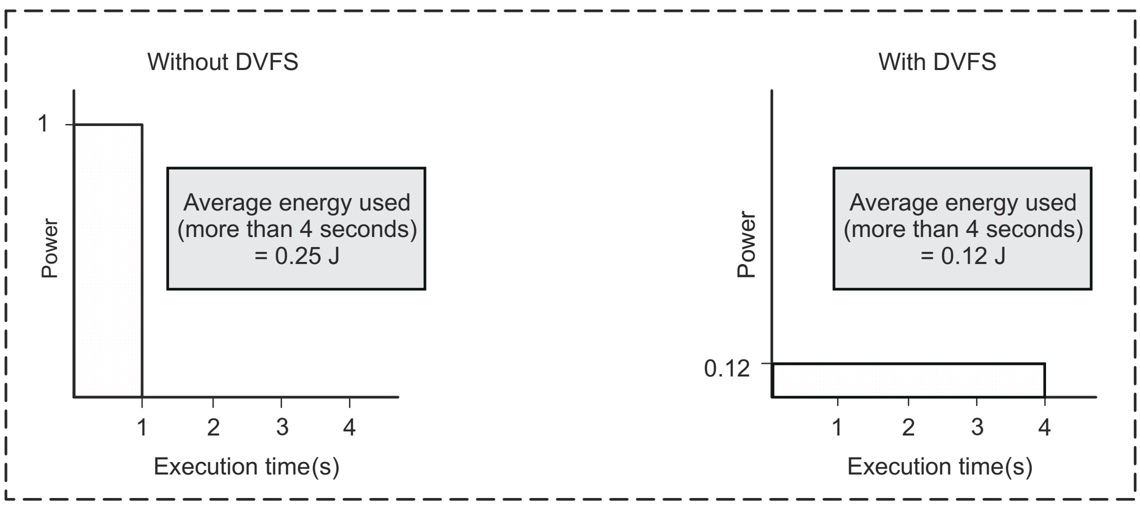


Figure 3-16. Comparison of Energy Consumed With/Without DVFS

Figure 3-16 shows the DVFS technique by comparing a process executed at maximum frequency and operating voltage without applying DVFS to the same process executed at optimal frequency and voltage using DVFS, based on the task requirements. If a task that must terminate in 4 seconds is performed at maximum operating frequency (see the left side of the figure), it terminates in 1 second, and the remaining 3 seconds are spent in idle mode.

With DVFS (see the right side of the figure), the operating frequency is reduced to optimal level; the task takes the full 4 seconds to complete, but power consumption is reduced. In addition, the voltage can be reduced further to save power so the dynamic and leakage power consumption are reduced.

DVFS requires control over the clock frequency and the operating voltage of the device elements. By intelligently switching the individual elements of the device to their OPPs, the power consumption of the device for a given task can be minimized.

For practical reasons related to the development of the device (flow, tools), DVFS can be used only for a few discrete steps, not over a continuum of voltage and frequency values. Each step, or OPP, is composed of a voltage (V) and frequency (F) pair. For an OPP, the frequency corresponds to the maximum frequency allowed at a voltage, or reciprocally; the voltage corresponds to the minimum voltage allowed for a frequency.

When applying DVFS, a processor or system always runs at the lowest OPP that meets the performance requirement at a given time. The user determines the optimal OPP for a given task and then switches to that OPP to save power.

3.1.2.3 Dynamic Power Switching

Like DVFS, dynamic power switching (DPS) is a power-management technique intended to reduce the active power consumption of a device. However, whereas DVFS reduces dynamic and leakage power consumption, DPS reduces only leakage power consumption, at the expense of a slight overhead in dynamic power consumption.

With DPS, the system switches dynamically between high- and low-consumption system power modes during system active time. When DPS is applied, a processor or system runs at the highest OPP (maximum frequency and voltage) to complete its tasks quickly, followed by an automatic switch to a low-power mode for minimum power consumption. DPS is useful when a real-time application is waiting for an event. The system can switch into a low-power system mode if the wake-up latency conditions allow it.

This technique consists of maximizing the idle period of the system to reduce its power consumption.

Note

The values in Figure 3-17 are hypothetical. They are meant only to clarify the concept and do not represent valid test results on the device.

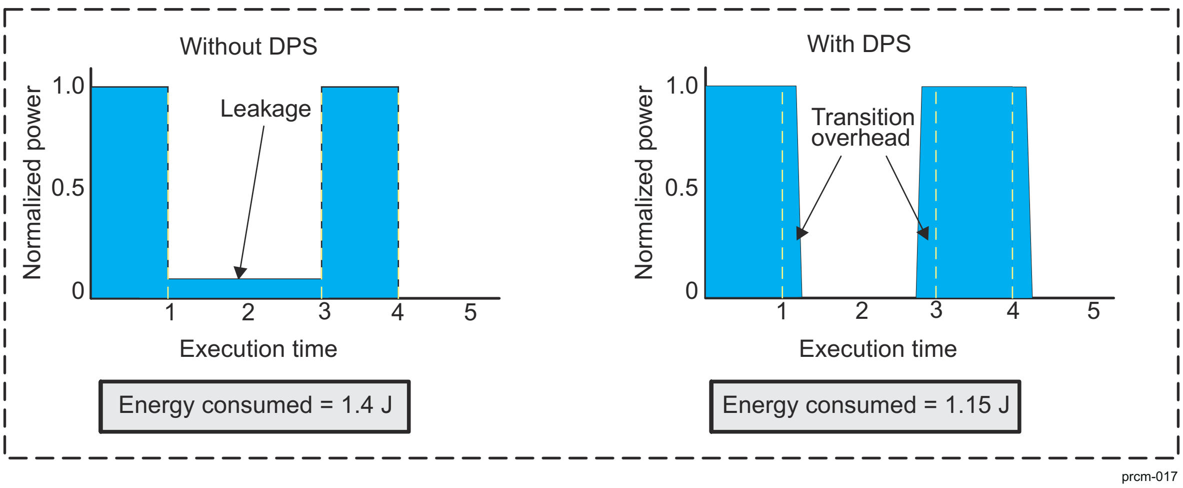


Figure 3-17. Comparison of Energy Consumed With/Without DPS

Figure 3-17 compares the behavior of power consumption for the same operation of the device without DPS (see the left side of the figure) and with DPS (see the right side of the figure). When operating without DPS, the device has a constant leakage current in idle mode. By using DPS, the system reduces the leakage current to 0. However, the transitions between system power modes may require storing the information before entering a low-power inactive state and restoring the information after a wake-up event (see Figure 3-17). This results in additional dynamic power consumption, referred to as transition overhead (see Figure 3-17). Transition overhead must be considered for a DPS operation.

For efficient deployment of DPS techniques, it is necessary to predict dynamically the performance requirement of the applications running on the processor. The DPS controller must account for the overhead of wake-up latencies related to domain switching and ensure that they do not significantly affect the performance of the device. Even with transition overhead, the user can identify an optimal idle-time limit, after which the DPS is useful for dynamic power saving.

3.1.2.4 Adaptive Voltage Scaling

With SmartReflex, power-supply voltage is adapted to silicon performance, statically (based on performance points predefined in the manufacturing process of a given device).

AVS achieves the optimal performance/power trade-off for all devices across the technology process spectrum. This ensures optimal power consumption for a given OPP.

3.1.2.5 Adaptive Body Bias

The device implements transistor body bias techniques for forward body bias (FBB) to boost the operating clock frequency for operation at higher OPPs.

Adaptive body bias (ABB) is based on the process corner and the current OPP. This is configured in the EFUSE_CTRL_CUST bit field at the device characterization and is not continuously updated. A dedicated LDO (VBBLDO) is used to produce the voltage bias.

ABB is supported only for MPU, IVAHD, DSPEVE, GPU voltage domains.

3.1.2.6 SR3-APG (Automatic Power Gating)

In addition to power-management techniques supported in the device, the MPU subsystem also employs SR3-APG power-management technology to reduce leakage. This technology allows for full logic and memory retention on MPU_C0 and MPU_C1 when required conditions are satisfied. It is controlled by the PRCM_MPU. For more information, see Chapter 4, *Dual Cortex-A15 MPU Subsystem*.

3.1.2.7 Combining Power-Management Techniques

The power-management techniques previously described have specific features and are most effective when used under the specific operating conditions of the device. Hence, the best active power savings are obtained by combining the DVFS, DPS, SLM, and AVS techniques. For a given operating state, one or more of the power-saving techniques can be applied to ensure optimal operation with maximum power saving.

AVS must be used at boot time to adapt the voltage to the process characteristics (strong/weak) of the device and then be used continuously to compensate temperature variations. AVS can also ensure the maximum available application performance of the device at a given OPP.

When medium application performance is required, or when application performance requirements vary, DVFS can be applied. The voltage and frequency can be scaled to match the closest OPP that meets the performance requirement.

When application performance requirements fall between two OPPs, or when a low application performance is required that is below the lowest performance OPP, DPS can be applied to switch to low-power mode.

When combining DVFS and DPS, the operating frequency must not be scaled to match the performance requirement without scaling the voltage. Lower operating frequency increases task completion time and reduces idle time. This prevents DPS or reduces its efficiency (DPS becomes more effective as idle time increases). Unless DPS cannot be applied for other reasons, for a given operating point of DVFS the operating frequency must always be set to the maximum allowed at a given voltage. This ensures optimal process completion time and application of DPS.

If DPS cannot be applied in a given context, scaling the frequency while keeping the voltage constant does not save energy; it does, however, reduce peak power consumption. This can have a positive effect on temperature dissipation and battery life.

SLM must be used when no applications are running and performance requirement drops to 0.

Note

The OPPs shown in Figure 3-18 are only for indication and clarity of text. They do not correspond to validated OPPs of the device.

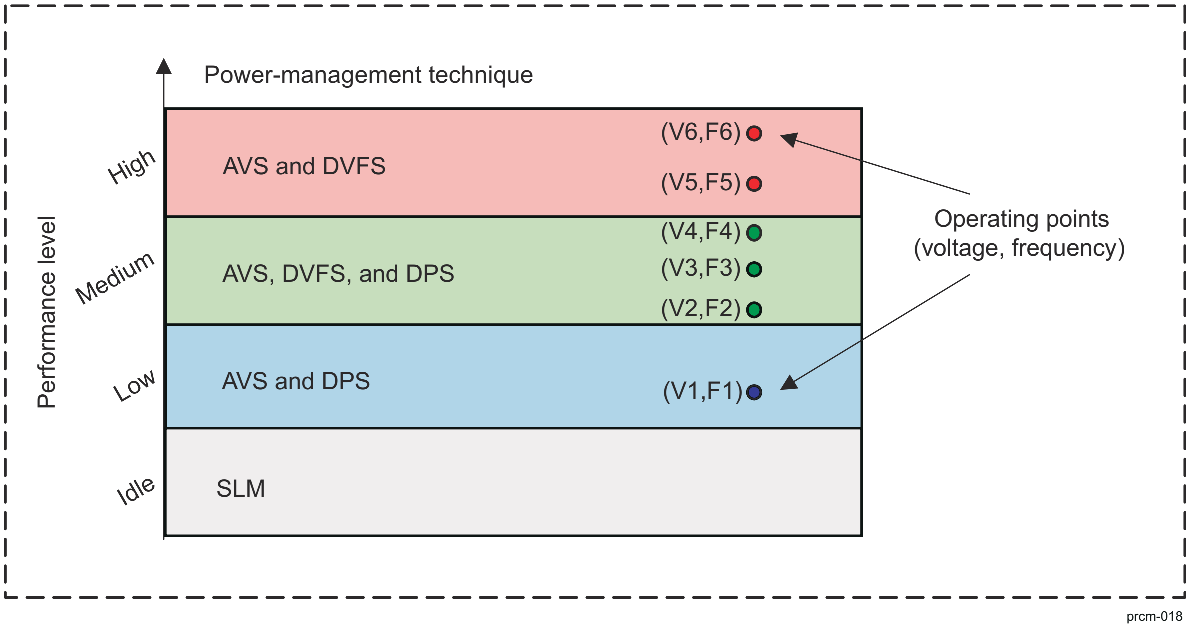


Figure 3-18. Performance Level and Applied Power-Management Techniques

3.1.2.7.1 DPS Versus SLM

DPS and SLM are similar concepts: they consist of switching the system between high- and low-power consumption modes. However, their operating timescales differ, principally in the latency allowed for mode transitions.

DPS is generally used in an applicative context (tasks are started). Therefore, mode transitions are related to system performance requirements or the processor load. DPS transition latencies must be small (typically between 10 μs and 100 μs) compared to the time constraints or deadlines of the application so that they do not degrade application performance. DPS requires performance prediction to ensure that transition latencies do not deteriorate device performance to the point that real-time application deadlines are missed or the user experience degrades too much for an interactive application.

SLM is not used in an applicative context (no task started). Mode transitions are related more to system responsiveness, and the transition latencies must be small compared to user sensitivity so that they do not degrade the user experience. For SLM, transition latencies are typically 1–10 ms or more.

DPS and SLM also differ in the type of wake-up event used to exit low-power idle mode. For DPS, wake-up events are application-related (timer, DMA request, peripheral interrupt, etc.); for SLM, wake-up events are user-related (touch screen, key pressed, peripheral connections, etc.).

3.2 PRCM Subsystem Overview

3.2.1 Introduction

The power-management framework of the device significantly reduces dynamic power consumption and static leakage current. This framework incorporates support for state-of-the-art power-management techniques. It ensures optimal device operation with significantly reduced power consumption. The power-management framework (PMFW) of the device is handled by the PRCM which is a logical module composed of the following three physical submodules:

- PRM: Handles device-level power and reset management. It also handles some clocks in the device. This module always remains on, unless no power is supplied to the device pads.
- CM_CORE_AON: Handles device-level clock management of the MPU, DSP1, DSP2, COREAON, CORE, IPU, L4PER, DSS, VPE, L3INIT, CAM, IVAHD, CUSTEFUSE, EVE1, EVE2, RTC and GPU power domains. This module always remains on, unless no power is supplied to the device pads.
- CM_CORE: Handles device-level clock management of the MMR block.

The device supports the power-management techniques with the following features:

- Partitioning of the device into voltage, power, clock, and reset domains
- Domain isolation that allows any combination of domain ON/OFF states
- Clock tree with selective clock-gating conditions
- Hardware-controlled reset sequencing management
- Power, reset, and clock control hardware mechanism to manage sleep and wake-up dependencies of the power domains
- Support for hardware-controlled autogating of module clocks
- Memory retention capability for preserving memory contents in low-power sleep mode
- DVFS support for the processor and peripherals

The PMFW interfaces with all the components of the device for power, clock, and reset management through associated control signals. It integrates enhanced features to let the device adapt energy consumption dynamically according to changing application and performance requirements. The innovative hardware architecture allows a substantial reduction in leakage current.

The power-management modules are fully configurable through their L4 interface ports.

Figure 3-19 is an overview of the power-management modules and their internal connections with a generic power domain.

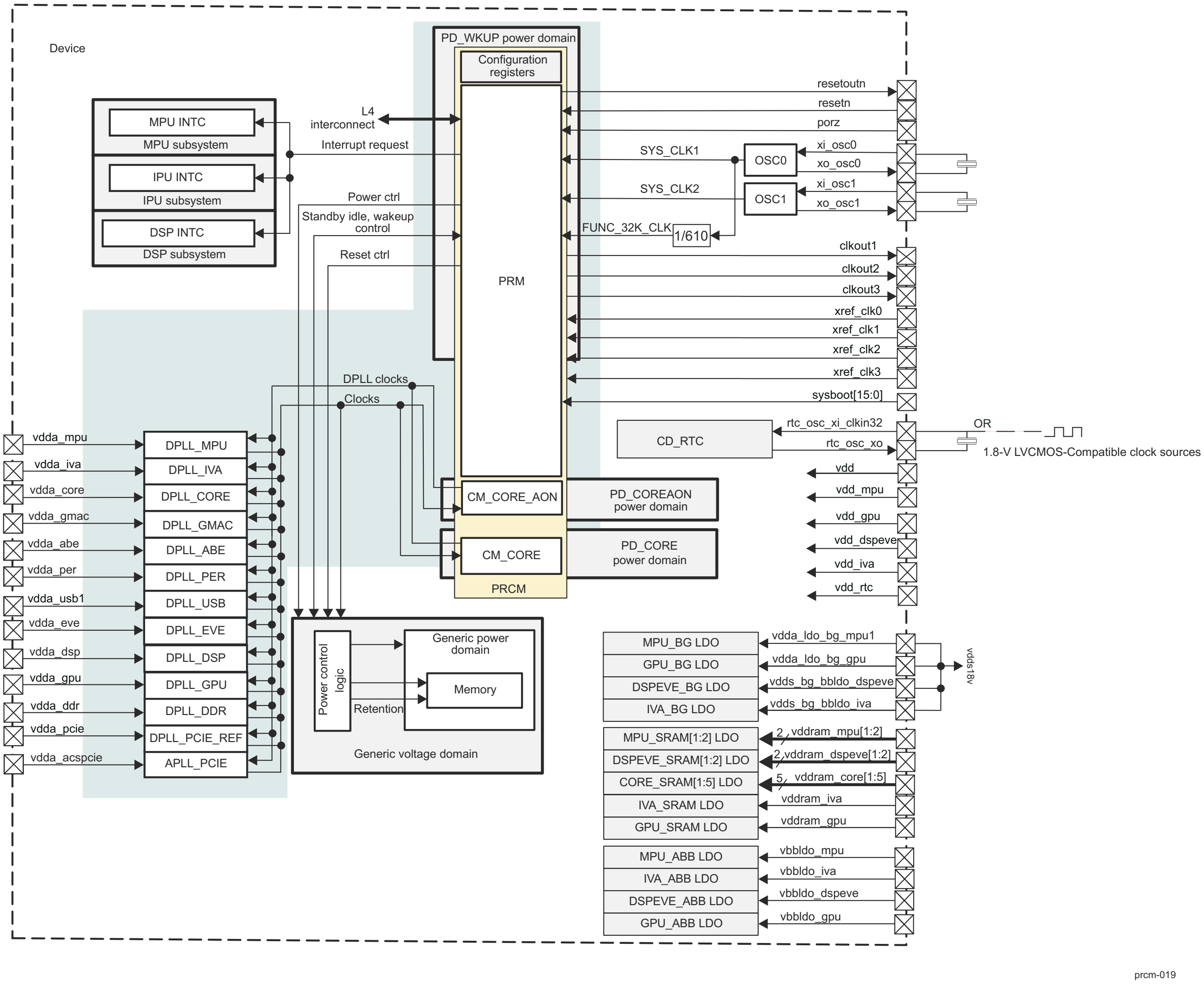


Figure 3-19. PMFW Overview

Note

The device I/O logic maps the module signals to the different pads of the device. For more information, see the *Pad Configuration Registers* and sections in Section 18.1, *Control Module*.

3.2.2 Power-Management Framework Features

The power-management modules:

- Manage independent power domains
- Control scalable logic voltage domains and selectable voltage modes for memory voltage domains
- Handle standby, idle, and wake-up procedures for the modules of the device
- Allow software and partial hardware control
- Monitor and handle wake-up events
- Control system clock and reset input sources
- Manage and distribute clocks and resets with high control granularity
- Handle power-up sequences
- Have debug and emulation features
- Control RFFs of device modules to support DPS

3.3 PRCM Subsystem Environment

The modules of the PMFW receive the external reset, clock, and power signals. See Figure 3-19.

The following sections describe the interfaces for external clock, reset, and power sources.

3.3.1 External Clock Signals

Table 3-23 lists the external clock pins, signal names, their direction, associated module, and description of the signals. If the signal is input to the device, the module is the destination module for the signal; if the signal is an output from the device, the module is the source module of the signal.

Table 3-23. External Clock Signals

Pin	Signal3	I/O(1)	PMFW Module	Description
rtc_osc_xi_clkin32	RTC_32K_CLK	I	Primary Input to SoC	RTC clock input. Optional external 32k clock
rtc_osc_xo	32K_CTRL	O	Primary Output	External 32k oscilator clock control (optional)
xi_osc0	XI_OSC0	I	Primary Input to SoC	System Oscillator OSC0 Crystal Input.This is the main system clock of the device.
xo_osc0	XO_OSC0	O	Primary Output	System Oscillator OSC0 Crystal output.
xi_osc1	XI_OSC1	I	Primary Input to SoC	Auxiliary Oscillator OSC1 Crystal input .
xo_osc1	XO_OSC1	O	Primary Output	Auxiliary Oscillator OSC1 Crystal output .
clkout1	CLKOUTMUX1_CLK	O	Primary Output	Device Clock output 1.Can be used as a system clock for other devices.
clkout2	CLKOUTMUX2_CLK	O	Primary Output	Device Clock output 2.Can be used as a system clock for other devices.
clkout3	CLKOUTMUX0_CLK	O	Primary Output	Device Clock output 3.Can be used as a system clock for other devices.
xref_clk0	XREF_CLK0	I	Primary Input to SoC	External Reference Clock 0.For Audio and other Peripherals.
xref_clk1	XREF_CLK1	I	Primary Input to SoC	External Reference Clock 1.For Audio and other Peripherals.
xref_clk2	XREF_CLK2	I	Primary Input to SoC	External Reference Clock 2.For Audio and other Peripherals.
xref_clk3	XREF_CLK3	I	Primary Input to SoC	External Reference Clock 3.For Audio and other Peripherals.

(1) I = Input; O = Output; I/O = Bidirectional

3.3.2 External Boot Signals

The PRM receives SYSBOOT[15:0] information from external pins sys_boot[15:0]. They are used to select interfaces or devices for the booting list. For more information, see Chapter 32, *Initialization*, Section 32.2.4, *Sysboot Configuration* .

Table 3-24 lists the external boot pins, signal names, their direction, the associated modules, description, and reset values of the signals. If the signal is input to the device, the module is the destination module for the signal; if the signal is an output from the device, the module is the source module of the signal.

Table 3-24. External Boot Signals

Pin	Signal	I/O (1)	PMWF Module	Description	Module Reset Value
sys_boot0	SYSBOOT0	I	PRM	System boot configuration pin 0: Latched at power-on reset (POR)	Z
sys_boot1	SYSBOOT1	I	PRM	System boot configuration pin 1: Latched at POR	Z
sys_boot2	SYSBOOT2	I	PRM	System boot configuration pin 2: Latched at POR	Z

Table 3-24. External Boot Signals (continued)

Pin	Signal	I/O [(1)]	PMWF Module	Description	Module Reset Value
sys_boot3	SYSBOOT3	I	PRM	System boot configuration pin 3: Latched at POR	Z
sys_boot4	SYSBOOT4	I	PRM	System boot configuration pin 4: Latched at POR	Z
sys_boot5	SYSBOOT5	I	PRM	System boot configuration pin 5: Latched at POR	Z
sys_boot6	SYSBOOT6	I	PRM	System boot configuration pin6: Latched at POR	Z
sys_boot7	SYSBOOT7	I	PRM	System boot configuration pin 7: Latched at POR	Z
sys_boot8	SYSBOOT8	I	PRM	System boot configuration pin 8: Latched at POR	Z
sys_boot9	SYSBOOT9	I	PRM	System boot configuration pin 9: Latched at POR	Z
sys_boot10	SYSBOOT10	I	PRM	System boot configuration pin 10: Latched at POR	Z
sys_boot11	SYSBOOT11	I	PRM	System boot configuration pin 11: Latched at POR	Z
sys_boot12	SYSBOOT12	I	PRM	System boot configuration pin12: Latched at POR	Z
sys_boot13	SYSBOOT13	I	PRM	System boot configuration pin 13: Latched at POR	Z
sys_boot14	SYSBOOT14	I	PRM	System boot configuration pin 14: Latched at POR	Z
sys_boot15	SYSBOOT15	I	PRM	System boot configuration pin15: Latched at POR	Z

(1) I = Input; O = Output; I/O = bidirectional

Note

For proper device operation, sysboot14 must be tied to vss . For SR1.1, sysboot15 must be tied to vdd, but for SR2.0 it is configurable. For more information, see Section 18.4.6.1.1.1, *Permanent PU/PD disabling (SR 2.0 only)* in Chapter 18, *Control Module*.

3.3.3 External Reset Signals

The PRM drives the SYS_WARM_OUT_RST reset output signal which is directly mapped as RESETOUT device pin.

The PRM asserts SYS_WARM_OUT_RST output upon the assertion of any source of global reset (warm or cold). SYS_WARM_OUT_RST assertion results in the assertion of device pin RESETOUT. This reset signal keeps external peripherals under reset while the device is under reset.

Table 3-25 lists the external reset pins, signal names, their direction, associated module, and description of the signals. If the signal is input to the device, the module is the destination module for the signal; if the signal is an output from the device, the module is the source module of the signal.

Table 3-25. External Reset Signals

Pin	Signal	I/O[(1)]	PMFW Module	Description	Module Reset Value
resetn	SYS_WARM_IN_RST	I	PRCM	Device Reset Input	Z

Table 3-25. External Reset Signals (continued)

Pin	Signal	I/O[(1)]	PMFW Module	Description	Module Reset Value
rstoutn	SYS_WARM_OUT_RST	O	PRCM	Reset out	Z
porz	SYS_PWRON_RST	I	PRCM	Power on Reset	Z

(1) I = Input; O = Output; I/O = bidirectional

3.3.4 External Voltage Inputs

Table 3-26 lists the external voltage sources related to the PRCM module.

Note

Table 3-26 lists only the voltage sources that are directly managed by the PRCM module or received by parts of the PRCM module (for example, DPLLs, LDOs, etc.). It does not give the device voltage sources not directly associated with the PRCM module.

Table 3-26. Voltage Sources

Pin	Signal	Managed by the PRCM	Description
vdd	VDD_CORE_L	Yes (AVS)	Supplies VDD_CORE_L logic voltage domain
vdd_mpu	VDD_MPU_L	Yes (AVS, ABB and DVFS)	Supplies VDD_MPU_L logic voltage domain
vdd_iva	VDD_IVAHD_L	Yes (AVS, ABB and DVFS)	Supplies VDD_IVAHD_L logic voltage domain
vdd_gpu	VDD_GPU_L	Yes (AVS, ABB, and DVFS)	Supplies VDD_GPU_L logic voltage domain
vdd_dspeve	VDD_DSPEVE_L	Yes (AVS, ABB, and DVFS)	Supplies VDD_DSPEVE_L logic voltage domain
vdda_gmac	VDDA_DPLL_GMAC	No (fixed)	DPLL_GMAC and HSDIVIDER analog power supply
vdda_core	VDDA_DPLL_CORE	No (fixed)	DPLL_CORE and HSDIVIDER analog power supply
vdda_mpu	VDDA_DPLL_MPU	No (fixed)	DPLL_MPU analog power supply
vdda_iva	VDDA_DPLL_IVA	No (fixed)	DPLL_IVA analog power supply
vdda_abe	VDDA_DPLL_ABE	No (fixed)	DPLL_ABE analog power supply
vdda_per	VDDA_DPLL_PER,	No (fixed)	DPLL_PER and PER HSDIVIDER analog power supply
vdda_usb1	VDDA_DPLL_USB	No (fixed)	DPLL_USB and HS USB1 analog power supply
vdda_dsp	VDDA_DPLL_DSP	No (fixed)	DPLL_DSP analog power supply
vdda_eve	VDDA_DPLL_EVE	No (fixed)	DPLL_EVE analog power supply
vdda_gpu	VDDA_DPLL_GPU	No (fixed)	DPLL_GPU analog power supply
vdda_ddr	VDDA_DPLL_DDR	No (fixed)	DPLL_DDR and DDR HSDIVIDER analog power supply
vdda_pcie	VDDA_DPLL_PCIE	No (fixed)	DPLL_PCIe_REF and PCIe analog power supply
vdda_acspcie	VDDA_APLL_PCIE	No (fixed)	APLL_PCIe analog power supply
vdd_rtc	VDD_RTC	No (fixed)	RTC voltage domain supply
vdda_ldo_bg_mpu1	VDDA_LDO_BG_MPU1	No	Supplies LDO_MPU and BG
vdda_ldo_bg_gpu	VDDA_LDO_BG_GPU	No	Supplies LDO_GPU and BG
vddram_mpu[1:2]	VDDRAM_MPU[1:2]	Yes (AVS)	Supplies LDO_MPU for MPU SRAM
vdds_bg_bbldo_dspeve	VDDS_LDO_BG_BBLDO_DSPEVE	No	Supplies LDO_DSPEVE for DSPEVE SRAM
vddram_dspeve[1:2]	VDDRAM_DSPEVE[1:2]	Yes (AVS)	Supplies LDO_DSPEVE for EVE-DSP SRAM

Table 3-26. Voltage Sources (continued)

Pin	Signal	Managed by the PRCM	Description
vdds_bg_bbldo_iva	VDDS_BG_BBLDO_IVA	No	Supplies Bandgap near IVA and BBLDO
vddram_iva	VDDRAM_IVA	Yes (AVS)	Supplies LDO_IVA for IVA SRAM
vddram_core[1:5]	VDDRAM_CORE[1:5]	Yes (AVS)	Supplies LDO_CORE for CORE SRAM
vddram_gpu	VDDRAM_GPU	Yes (AVS)	Supplies LDO_GPU for GPU SRAM
vbbldo_mpu	VBBLDO_MPU	Yes (AVS)	MPU Back bias supply
vbbldo_iva	VBBLDO_IVA	Yes (AVS)	IVA Back bias supply
vbbldo_dspeve	VBBLDO_DSPEVE	Yes (AVS)	EVE-DSP Back bias supply
vbbldo_gpu	VBBLDO_GPU	Yes (AVS)	GPU Back bias supply

3.4 PRCM Subsystem Integration

The internal configuration registers of the power-management modules can be accessed for configuration and control through their respective L4_WKUP interconnect. In addition to the L4 interconnect, the internal module interface contains the following interfaces and signals:

- A set of signals for idle/wake-up control for each module
- Clocks and reset signals for the modules
- Power control signals (switches and memories) to the power domains
- Interrupts to the MPU, IPU, and DSP INTCs
- Phase-locked loop (PLL) control commands for recalibration and bypass of the digital phase-locked loops (DPLLs)

Figure 3-19 shows details of the control interface to a generic power domain.

3.4.1 Device Power-Management Layout

The PMFW sees the device split into voltage domains, power domains, and clock domains. Table 3-27 provides the device-level view with module association to the clock, power, and voltage domains.

Table 3-27. PMFW Device-Level Layout

Voltage Domain	Power Domain	Clock Domain	Module
VD_CORE	PD_WKUPAON	CD_WKUPAON	CTRL_MODULE_WKUP
			L4_WKUP interconnect
			GPIO1
			TIMER1
			TIMER12
			WD_TIMER2
			PRCM_MPU
			COUNTER_32K
			KBD
			DCAN1
			UART10
		N/A (the PRCM module)	PRM
	PD_COREAON	CD_COREAON	CM_CORE_AON
			APLL_PCIE
			WUGEN_DMA_SYSTEM
		N/A	DPLL_ABE, DPLLCTRL_ABE
			DPLL_CORE, DPLLCTRL_CORE
			DPLL_PER, DPLLCTRL_PER
			DPLL_GPU, DPLLCTRL_GPU
			DPLL_IVA, DPLLCTRL_IVA
			DPLL_GMAC, DPLLCTRL_GMAC
			DPLL_DDR, DPLLCTRL_DDR
			DPLL_PCIE_REF
			DPLL_SATA, DPLLCTRL_SATA
			DPLL_USB, DPLL_USB_OTG_SS, DPLLCTRL_USB, DPLLCTRL_USB_OTG_SS
			DPLL_HDMI, DPLLCTRL_HDMI
			DPLL_VIDEO1, DPLLCTRL_VIDEO1

Table 3-27. PMFW Device-Level Layout (continued)

Voltage Domain	Power Domain	Clock Domain	Module
			DPLL_VIDEO2, DPLLCTRL_VIDEO2
			DPLL_DSP, DPLLCTRL_DSP
			DPLL_EVE, DPLLCTRL_EVE
			WUGEN_IPU
		CD_IPU	McASP1
			TIMER5
			TIMER6
			TIMER7
			TIMER8
			UART6
			I2C5
		CD_EMU	DEBUGSS, DPLL_DEBUG, DPLLCTRL_DEBUG, MPU_EMU_DEBUG
		CD_L4_PER1	TIMER10
			TIMER11
			TIMER2
			TIMER3
			TIMER4
			TIMER9
			ELM
			GPIO2
			GPIO3
			GPIO4
			GPIO5
			GPIO6
			GPIO7
			GPIO8
			HDQ1W
			I2C1
			I2C2
			I2C3
			I2C4
			L4_PER1 interconnect
			McSPI1
			McSPI2
			McSPI3
			McSPI4
			MMC3
			MMC4
			UART1
			UART2
			UART3
			UART4
			UART5

Table 3-27. PMFW Device-Level Layout (continued)

Voltage Domain	Power Domain	Clock Domain	Module
		CD_L4PER2	L4_PER2 interconnect
			McASP2
			McASP3
			McASP4
			McASP5
			McASP6
			McASP7
			McASP8
			DCAN2
			QSPI
			PWMSS1
			PWMSS2
			PWMSS3
			UART7
			UART8
			UART9
		CD_L4PER3	L4_PER3 interconnect
			TIMER13
			TIMER14
			TIMER15
			TIMER16
		CD_L4SEC	AES1
			AES2
			SHA2MD5_1
			SHA2MD5_2
			RNG
			DMA_CRYPTO
			DES3DES
			FPKA
		CD_L4_CFG	CTRL_MODULE_CORE
			SPINLOCK
			L4_CFG interconnect
			MAILBOX1, MAILBOX2, MAILBOX3, MAILBOX4, MAILBOX5, MAILBOX6, MAILBOX7, MAILBOX8, MAILBOX9, MAILBOX10, MAILBOX11, MAILBOX12, MAILBOX13
			OCP2SCP2
		CD_EMIF	DMM
			EMIF1
			EMIF2
			EMIF_OCP_FW
			DLL (EMIFDLL1, EMIFDLL2)
			EMIF_PHY1, EMIF_PHY2

Table 3-27. PMFW Device-Level Layout (continued)

Voltage Domain	Power Domain	Clock Domain	Module
		CD_L3_MAIN1	L3_MAIN_1 interconnect
			GPMC
			OCMC_RAM1
			OCMC_RAM2
			OCMC_RAM3
			MMU1
			MMU2
			VCP1
			VCP2
			EDMA_TC0, EDMA_TC1, EDMA_TPCC
		CD_ATL	ATL
		CD_DMA	DMA_SYSTEM
		CD_L3_INSTR	OCP_WP_NOC
			L3_INSTR interconnect
			L3_MAIN_2 interconnect
			CTRL_MODULE_BANDGAP
			DLL_AGING
		CD_L3INIT	MMC1
			MMC2
			OCP2SCP1
			OCP2SCP3
			IEEE1500_2_OCP
			MLB
		CD_GMAC	GMAC
	PD_CAM	CD_CAM	VIP1
			VIP2
			VIP3
	PD_CORE	CD_IPU2	IPU2
	PD_IPU	CD_IPU1	IPU1
	PD_DSS	CD_DSS	BB2D, DSS, HDMI, HDMI_PHY
	PD_CUSTEFUSE	CD_CUSTEFUSE	EFUSE_CTRL_CUST
	PD_L3INIT	CD_L3INIT	USB1
			USB2
			USB3, USB3_PHY, USB3_PHY_RX, USB3_PHY_TX, USB2PHY1, USB2PHY2
			USB4
			SATA, SATA_PHY_RX, SATA_PHY_TX
		CD_PCIE	PCIe_SS1, PCIe_SS2
	PD_VPE	CD_VPE	VPE
VD_MPU	PD_MPU	CD_MPU	CPU0, CPU1
	PD_MPUAON	CD_MPUAON	MPU_L2CACHE
		N/A	DPLL_MPU, DPLLCTRL_MPU, INTC_MPU

Table 3-27. PMFW Device-Level Layout (continued)

Voltage Domain	Power Domain	Clock Domain	Module
VD_IVAHD	PD_IVA	CD_IVA	IVAHD
			SL2
VD_DSPEVE	PD_DSP1	CD_DSP1	DSP1
	PD_DSP2	CD_DSP2	DSP2
	PD_EVE1	CD_EVE1	EVE1
	PD_EVE2	CD_EVE2	EVE2
	PD_MMAON	N/A (not clocked by PRCM)	DSP SYS Wakeup Logic
VD_GPU	PD_GPU	CD_GPU	GPU
VD_RTC	PD_RTC	CD_RTC	RTC_SS

3.4.2 Power-Management Scheme, Reset, and Interrupt Requests

3.4.2.1 Power Domain

Table 3-28 lists the PMFW modules and their associated power domains.

Table 3-28. PMFW Module Power Domains

PMFW Module	Power Domain
PRM	PD_WKUPAON
CM_CORE_AON	PD_COREAON
CM_CORE	PD_CORE

The PRM part of the PRCM module is in the PD_WKUPAON power domain, which is continuously active. It is composed of the logic that must be permanently supplied to manage domain power-state transitions and detect wake-up events.

The CM_CORE_AON part of the PRCM module is in the PD_COREAON power domain, which is an always-on power domain, while the CM_CORE part of the PRCM module is in the PD_CORE power domain, which can be activated and deactivated according to the requirements of the executing applications.

3.4.2.2 Resets

The PMFW modules are reset by independent reset signals (see Table 3-29).

Table 3-29. PMFW Module Reset Signals

PMFW Module	Reset Signal
PRM	SYS_PWRON_RST_IN
CM_CORE_AON	CM_CORE_AON_PWRON_RST
	CM_CORE_AON_RST
CM_CORE	CM_CORE_PWRON_RET_RST
	CM_CORE_RET_RST

Note

For more information about the reset trigger sources and assertion conditions, see Section 3.5.3, *Reset Sources*.

3.4.2.3 PRCM Interrupt Requests

The PMFW modules can generate the interrupts listed in Table 3-30.

Table 3-30. PMFW Hardware Requests

Interrupt Requests

Table 3-30. PMFW Hardware Requests (continued)

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
PRM	PRM_IRQ_MPU	IRQ_CROSSBAR_6	MPU_IRQ_11 DSP1_IRQ_37 DSP2_IRQ_37	PRM MPU interrupt request
	PRM_IRQ_IPU1	IRQ_CROSSBAR_133	IPU1_IRQ_47 IPU2_IRQ_47	PRM IPU1 interrupt request
	PRM_IRQ_IPU2	IRQ_CROSSBAR_386	-	This IRQ source signal is not mapped by default to eny device INTC.

3.4.2.4

Note

The "**Default Mapping**" column in Table 3-30 *PMFW Hardware Requests* shows the default mapping of module IRQ source signals. These module IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module, respectively.
For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.
For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

3.5 Reset Management Functional Description

3.5.1 Overview

In the device, the reset scheme is managed by the PRM, device-level reset manager.

3.5.1.1 PRCM Reset Management Functional Description

The PRM handles the device power-on and warm reset pads: porz, resetn, and rstoutn. The PRM:

- Extends the reset duration beyond the porz pad release
- Provides the reset duration period after the resetn pad assertion
- Routes the device pad resets

The PRM is functionally sensitive to the device POR.

3.5.1.1.1 Power-On Reset

The PRM receives the device POR on the porz reset pad of the device. It extends the duration of the POR module until at least stable 32-kHz clock and SYS_CLK1 are provided to it. The 32-kHz clock and SYS_CLK1 of the device are propagated once the PRCM releases the internal POR.

3.5.1.1.2 Warm Reset

The device warm reset can be received from an external source or the device PRM module as a result of an internal event.

The PRCM can generate the global reset used by the domain Reset Managers from these warm reset input sources:

- SYS_WARM_IN_RST
- MPU_WDT_RST
- GLOBAL_SW_WARM_RST

3.5.1.2 PRM Reset Management Functional Description

The PRM module manages the resets to all power domains inside the device.

Note

The PRM module has no knowledge or control over resets generated locally within a module (for example, through the <Module name>_SYSCONFIG[x] SOFTRESET configuration register bit). A software reset has the same effect on the module logic as a hardware reset.

All PRM reset outputs are asynchronously asserted, while the deassertion is synchronous to the SYS_CLK1 clock. The reset managers in PRM use this clock to stall, or delay, deassertion of reset upon source deactivation.

In each power domain one or more reset domains are defined by a unique reset signal that originates from the reset manager and is connected to one or more modules of the device. All the connected modules of the reset domain are reset simultaneously when the reset signal is asserted. Independent control of these reset domains allows sequencing of the release of resets and ensures a safe reset of the entire power domain.

3.5.2 General Characteristics of Reset Signals

Reset signals can be categorized based on four criteria:

- Scope: Global or local reset
- Occurrence: Cold or warm reset
- Source type: Software-controlled or hardware-triggered reset
- Retention type: Retention reset or nonretention reset

3.5.2.1 Scope

A reset signal can be categorized according to its scope (the area of the device affected by the reset):

- Global reset:Affects the entire logic of the device; all modules are reset. Generally, occurs when the device powers up (POR) or an abnormal operation is detected (watchdog timeout, thermal shutdown, etc.)
- Local reset: Affects one power domain, reset domain, or module.

3.5.2.2 Occurrence

A reset signal can be categorized depending on when the reset occurs:

- Cold reset: Occurs on device power up (POR), or in certain emulation modes. Also, it can be software-initiated. Upon cold reset, it is assumed the device is being powered up, and therefore, everything in the device is being reset. That is, cold resets must be considered as global resets.
- Warm reset: Warm reset types are not necessarily applied globally within device. Also, a module can use a warm reset to reset a subset of its logic. This is often done to speed up reset recovery time; that is, the time to transition to a safe operating state, compared to the time required upon receipt of a cold reset. Warm reset events include software-triggered reset per power domain, watchdog time-out, externally triggered and emulation initiated.

Modules that behave differently in cold reset and warm reset have two reset signals: RST and PWRON_RST. These reset signals reconstruct warm reset and cold reset in modules that require them.

The following modules are reset upon global cold reset events and not upon global warm reset events:

- All DPLLs associated with the PRCM module
- EMIF
- Control module: CTRL_MODULE_CORE, CTRL_MODULE_WKUP

Note

The DPLLs state will change after warm reset even though they are not warm reset-sensitive, refer to Section 3.5.6.18, *Global Warm Reset Sequence* for more details.

Note

For information about the PRCM module registers affected by the global warm reset, see the register description in Section 3.13, *PRCM Register Manual*.

3.5.2.3 Source Type

A reset can be categorized depending on whether it is software-controlled or hardware-triggered:

- Software reset: Triggered by setting a bit in a configuration register of the PRCM module
- Hardware reset: Triggered by a signal from a hardware module inside or outside the PRCM module

3.5.2.4 Retention Type

The power domain manager in PRM controls the assertion of two types of local cold reset sources, retention and nonretention reset. They are identified by the naming convention <PowerDomain>_DOM_RET_RST and <PowerDomain>_DOM_RST, respectively.

Upon transitioning a power domain to the ON-ACTIVE power state from the OFF power state:

- Retention type reset source is always asserted.
- Nonretention type reset source is always asserted.

Upon transitioning a power domain to the ON-ACTIVE power state from the RETENTION power state:

- Retention type reset source is never asserted.
- Nonretention type reset source is optionally asserted. The software-selectable option enables the PRM to support CSWRET mechanism. Nonretention type reset source has the following behavior:
  - It is not asserted if the power domain state transitions from the CSWR-RETENTION state.

3.5.3 Reset Sources

The reset sources triggering the reset managers in the PRM are described in this section.

3.5.3.1 Global Reset Sources

Table 3-31 lists the global reset sources of the device. The global reset source signals received by the reset manager trigger the reset of all the device modules. For all hardware reset signals, the source of the reset is identified; for the software reset signals, the bit triggering the reset is identified.

Table 3-31. Global Reset Sources

Type(1)	Name	Source/Control	Description
H/C	SYS_PWRON_RST	porz input pin	The entire device is held in reset during porz pin assertion. porz pin is typically asserted by external PMIC at POR. PMIC keeps porz asserted until all voltage rails are ramped-up and stable. PRM extends device internal reset after porz release. Refer to Power Supply Sequences in Device Data Manual.
H/W	SYS_WARMIN_RST	resetn input pin	Warm reset from external device. Internal warm reset is triggered by the active (falling) edge of this signal and is extended with RstTime1.
S/C	GLOBAL_COLD_SW_RST	PRM_RSTCTRL[1] RST_GLOBAL_COLD_SW	Global software cold reset
S/W	GLOBAL_WARM_SW_RST	PRM_RSTCTRL[0] RST_GLOBAL_WARM_SW	Global software warm reset
H/W	TSHUT_MPU_RST	MPU voltage domain thermal sensor	Asserted when measured temperature is greater than shutdown temperature threshold
H/W	TSHUT_IVA_RST	IVAHD voltage domain thermal sensor	
H/W	TSHUT_CORE_RST	Device thermal sensor, placed close to the ISS, DSP, and IVA	
H/W	TSHUT_DSPEVE_RST	DSPEVE voltage domain thermal sensor	
H/W	TSHUT_GPU_RST	GPU voltage domain thermal sensor	
H/W	ICEPICK_RST	ICEPick™ module	ICEPick warm reset. It is used only in emulation mode.
H/C	ICEPICK_POR_RST	ICEPick module	ICEPick cold reset. It is used only in emulation mode.
H/W	MPU_WDT_RST	WD_TIMER2 or MPU subsystem WDT	It is triggered by a time-out event.

(1) H = Hardware reset, S = Software reset, C = Cold reset, W = Warm reset

3.5.3.2 Local Reset Sources

In addition to the global reset sources the device can have a number of local reset sources for each power domain. The local reset sources can be cold or warm reset sources. They can be software-controlled or hardware-triggered. Table 3-32 identifies the possible types of hardware-triggered local cold reset sources. Some power domains can support one or both of these local cold reset sources. The table does not list the software-triggered local warm reset sources that are listed in the reset management section of the respective power domains. A local reset source signal received by the reset manager resets only a specific part of the device (for example, some modules/subsystems within the power domain).

Table 3-32. Local Reset Sources

Type(1)	Name	Source/Control	Description
H/C	<Power domain>_RET_RST	PRCM	Asserted only for a power domain state transition from OFF to ON-ACTIVE state
H/C	<Power domain>_RST	PRCM	Asserted for any power domain transition from OFF to ON-ACTIVE state

(1) H = Hardware reset, C = Cold reset

3.5.4 Reset Logging

The reset status registers RM_<power domain>_RSTST and PRM_RSTST are reset asynchronously on assertion of a global cold reset. However, a reset status bit is always logged when the reset is released to the domain.

For this reason, after the assertion of a global cold reset, the reset status register is cleared to 0. When the domain reset is released, the register bit to log the global cold reset (the PRM_RSTST[0] GLOBAL_COLD_RST bit) is updated to 1. For the same reason, the reset status register of domains released from reset by software is updated only when software releases the domain reset.

The assertion of a global cold reset prevents logging any other source of reset until after the release of the domain reset. This is valid in the following situations:

- A source of reset other than global cold reset is asserted before, during, or after the active period of a global cold source of reset and before the release of the domain reset signal.
- A source of reset other than global cold reset is asserted and then released, but a global cold reset source is asserted before the release of the domain reset signal.

3.5.5 Reset Domains

A power domain can receive power-on reset (PWRON_RST) and/or normal reset (RST) signals. These signals reset nonretention logic and behave as follows:

- On any global or local cold reset, RST and PWRON_RST are asserted.
- On any global or local warm reset, only RST is asserted.
- On wakeup from OFF, PWRON_RST is asserted.

A power domain can receive two additional retention logic reset signals: power-on retention reset (PWRON_RET_RST) and/or retention reset (RET_RST). These signals behave as follows:

- On any global cold reset, RET_RST and PWRON_RET_RST are asserted.
- On any global cold reset or wakeup from an off (no power supply) state to the ON-ACTIVE state, RET_RST and PWRON_RET_RST are asserted.
- On any global warm reset, only RET_RST is asserted.

This section discusses the trigger sources and attributes for all reset domains of the device. For an explanation of each reset trigger source of the device, see Section 3.5.3, *Reset Sources.*

Table 3-33 identifies the associated power and rest domains for each module.

Note

The DPLLs state will change after warm reset even though their registers are not warm reset-sensitive, refer to Section 3.5.6.18, *Global Warm Reset Sequence* for more details.

Table 3-33. Modules, Power Domains, and Reset Domains Association

Module	Power Domain	Reset Domains
CM_CORE_AON	PD_COREAON	CM_CORE_AON_PWRON_RST, CM_CORE_AON_RST
APLL_PCIE	PD_COREAON	COREAON_PWRON_RST
DPLL_ABE	PD_COREAON	COREAON_PWRON_RST
DPLL_CORE	PD_COREAON	COREAON_PWRON_RST
DPLL_PER	PD_COREAON	COREAON_PWRON_RST
DPLL_PCIE_REF	PD_COREAON	COREAON_PWRON_RST
DPLL_IVA	PD_COREAON	DPLL_IVA_PRWON_RST
DPLL_GMAC	PD_COREAON	COREAON_PWRON_RST
DPLL_DDR	PD_COREAON	COREAON_PWRON_RST
DPLL_GPU	PD_COREAON	COREAON_PWRON_RST

Table 3-33. Modules, Power Domains, and Reset Domains Association (continued)

Module	Power Domain	Reset Domains
DPLL_USB	PD_COREAON	COREAON_PWRON_RST
WUGEN_IPU	PD_COREAON	None
WUGEN_DMA_SYSTEM	PD_COREAON	None
SPINNER	PD_COREAON	None
DPLL_DSP	PD_COREAON	DPLL_DSP_PWRON_RST
DPLL_EVE	PD_COREAON	DPLL_EVE_PWRON_RST
DPLL_MPU	PD_MPUAON	DPLL_MPU_PWRON_RST
INTC_MPU	PD_MPUAON	MPUAON_RST
MPU	PD_MPU	MPU_PWRON_RST, MPU_RST, MPU_MA_PWRON_RET_RST, MPU_MA_RET_RST, MPU_MA_RST
McASP1	PD_COREAON	IPU_RST
TIMER5	PD_COREAON	IPU_RST
TIMER6	PD_COREAON	IPU_RST
TIMER7	PD_COREAON	IPU_RST
TIMER8	PD_COREAON	IPU_RST
UART6	PD_COREAON	IPU_RET_RST
I2C5	PD_COREAON	IPU_RST
IPU1	PD_IPU	IPU1_PWRON_RST, IPU1_RET_RST, IPU1_CPU0_RST, IPU1_CPU1_RST, IPU1_RST
VIP1	PD_CAM	CAM_RST
VIP2	PD_CAM	CAM_RST
VIP3	PD_CAM	CAM_RST
EFUSE_CTRL_CUST	PD_CUSTEFUSE	CUSTEFUSE_RST
CM_CORE	PD_CORE	CM_CORE_PWRON_RET_RST, CM_CORE_RET_RST
CTRL_MODULE_CORE	PD_COREAON	CORE_PWRON_RET_RST
CTRL_MODULE_BANDGAP	PD_COREAON	CORE_PWRON_RET_RST
EMIFPHY	PD_CORE	EMIF_DDR_PHY_PWRON_RST
DLL	PD_CORE	DLL_RST
DLL_AGING	PD_CORE	CORE_RST
DMM	PD_COREAON	CORE_RST, CORE_RET_RST
IPU2	PD_COREAON	IPU2_PWRON_RST, IPU2_RET_RST, IPU2_CPU0_RST, IPU2_CPU1_RST,IPU2_RST
EMIF1	PD_COREAON	CORE_PWRON_RET_RST, CORE_PWRON_RST
EMIF2	PD_COREAON	CORE_PWRON_RET_RST, CORE_PWRON_RST
EMIF_OCP_FW	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
GPMC	PD_COREAON	CORE_RET_RST
SPINLOCK	PD_COREAON	CORE_RET_RST
L3_MAIN_2 interconnect	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
L3_MAIN_1 interconnect	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
L3_INSTR interconnect	PD_COREAON	CORE_RST
OCP_WP_NOC	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
L4_CFG interconnect	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST

Table 3-33. Modules, Power Domains, and Reset Domains Association (continued)

Module	Power Domain	Reset Domains
MAILBOX1	PD_COREAON	CORE_RET_RST
MAILBOX10	PD_COREAON	CORE_RET_RST
MAILBOX11	PD_COREAON	CORE_RET_RST
MAILBOX12	PD_COREAON	CORE_RET_RST
MAILBOX13	PD_COREAON	CORE_RET_RST
MAILBOX2	PD_COREAON	CORE_RET_RST
MAILBOX3	PD_COREAON	CORE_RET_RST
MAILBOX4	PD_COREAON	CORE_RET_RST
MAILBOX5	PD_COREAON	CORE_RET_RST
MAILBOX6	PD_COREAON	CORE_RET_RST
MAILBOX7	PD_COREAON	CORE_RET_RST
MAILBOX8	PD_COREAON	CORE_RET_RST
MAILBOX9	PD_COREAON	CORE_RET_RST
MMU1	PD_COREAON	CORE_RET_RST
MMU2	PD_COREAON	CORE_RET_RST
VCP1	PD_COREAON	CORE_RST
VCP2	PD_COREAON	CORE_RST
EDMA_TPCC	PD_COREAON	CORE_RET_RST
EDMA_TC0	PD_COREAON	CORE_RET_RST
EDMA_TC1	PD_COREAON	CORE_RET_RST
OCMC_RAM1	PD_COREAON	CORE_RST
OCMC_RAM2	PD_COREAON	CORE_RST
OCMC_RAM3	PD_COREAON	CORE_RST
DMA_SYSTEM	PD_COREAON	DMA_RET_RST
OCP2SCP2	PD_COREAON	CORE_RST
ATL	PD_COREAON	CORE_RST
DSS	PD_DSS	DSS_RET_RST, DSS_RST
BB2D	PD_DSS	DSS_RST
VPE	PD_VPE	VPE_RST
CM_EMU	PD_COREAON	EMU_PWRON_RST
DEBUGSS	PD_COREAON	EMU_EARLY_PWRON_RST, EMU_PWRON_RST, EMU_RST
GPU	PD_GPU	GPU_RST
IVAHD	PD_IVA	IVA_PWRON_RST, IVA_RST, IVA_SEQ1_RST, IVA_SEQ2_RST
SL2	PD_IVA	IVA_RST
IEEE1500_2_OCP	PD_COREAON	L3INIT_RST
MMC1	PD_COREAON	L3INIT_RET_RST
MMC2	PD_COREAON	L3INIT_RET_RST
USB1	PD_L3INIT	L3INIT_RET_RST
USB2	PD_L3INIT	L3INIT_RET_RST
USB3	PD_L3INIT	L3INIT_RET_RST
USB4	PD_L3INIT	L3INIT_RET_RST
OCP2SCP1	PD_COREAON	L3INIT_RST
OCP2SCP3	PD_COREAON	L3INIT_RST
SATA	PD_L3INIT	L3INIT_RST

Table 3-33. Modules, Power Domains, and Reset Domains Association (continued)

Module	Power Domain	Reset Domains
PCIe_SS1	PD_L3INIT	L3INIT_PWRON_RST, L3INIT_RST
PCIe_SS2	PD_L3INIT	L3INIT_PWRON_RST, L3INIT_RST
MLB_SS	PD_L3INIT	L3INIT_RST
GMAC_SW	PD_COREAON	L3INIT_RST
TIMER10	PD_COREAON	L4PER_RST
TIMER11	PD_COREAON	L4PER_RST
TIMER13	PD_COREAON	L4PER_RST
TIMER14	PD_COREAON	L4PER_RST
TIMER15	PD_COREAON	L4PER_RST
TIMER16	PD_COREAON	L4PER_RST
TIMER2	PD_COREAON	L4PER_RST
TIMER3	PD_COREAON	L4PER_RST
TIMER4	PD_COREAON	L4PER_RST
TIMER9	PD_COREAON	L4PER_RST
ELM	PD_COREAON	L4PER_RST
GPIO2	PD_COREAON	L4PER_RET_RST
GPIO3	PD_COREAON	L4PER_RET_RST
GPIO4	PD_COREAON	L4PER_RET_RST
GPIO5	PD_COREAON	L4PER_RET_RST
GPIO6	PD_COREAON	L4PER_RET_RST
GPIO7	PD_COREAON	L4PER_RET_RST
GPIO8	PD_COREAON	L4PER_RET_RST
HDQ1W	PD_COREAON	L4PER_RST
I2C1	PD_COREAON	L4PER_RET_RST
I2C2	PD_COREAON	L4PER_RST
I2C3	PD_COREAON	L4PER_RST
I2C4	PD_COREAON	L4PER_RST
L4_PER1 interconnect	PD_COREAON	L4PER_PWRON_RET_RST, L4_PER_RST
L4_PER2 interconnect	PD_COREAON	L4PER_PWRON_RET_RST, L4_PER_RST
L4_PER3 interconnect	PD_COREAON	L4PER_PWRON_RET_RST, L4_PER_RST
McASP2	PD_COREAON	L4PER_RST
McASP3	PD_COREAON	L4PER_RST
McASP4	PD_COREAON	L4PER_RST
McASP5	PD_COREAON	L4PER_RST
McASP6	PD_COREAON	L4PER_RST
McASP7	PD_COREAON	L4PER_RST
McASP8	PD_COREAON	L4PER_RST
McSPI1	PD_COREAON	L4PER_RST
McSPI2	PD_COREAON	L4PER_RST
McSPI3	PD_COREAON	L4PER_RST
McSPI4	PD_COREAON	L4PER_RST
MMC3	PD_COREAON	L4PER_RST
MMC4	PD_COREAON	L4PER_RST
DCAN2	PD_COREAON	L4PER_RST
UART1	PD_COREAON	L4PER_RET_RST

Table 3-33. Modules, Power Domains, and Reset Domains Association (continued)

Module	Power Domain	Reset Domains
UART2	PD_COREAON	L4PER_RET_RST
UART3	PD_COREAON	L4PER_RET_RST
UART4	PD_COREAON	L4PER_RET_RST
UART5	PD_COREAON	L4PER_RET_RST
UART7	PD_COREAON	L4PER_RET_RST
UART8	PD_COREAON	L4PER_RET_RST
UART9	PD_COREAON	L4PER_RET_RST
DMA_CRYPTO	PD_COREAON	L4PER_RET_RST
AES1	PD_COREAON	L4PER_RET_RST
AES2	PD_COREAON	L4PER_RET_RST
SHA2MD5_1	PD_COREAON	L4PER_RET_RST
SHA2MD5_2	PD_COREAON	L4PER_RET_RST
RNG	PD_COREAON	L4PER_RET_RST
QSPI	PD_COREAON	L4PER_RST
PWMSS1	PD_COREAON	L4PER_RST
PWMSS2	PD_COREAON	L4PER_RST
PWMSS3	PD_COREAON	L4PER_RST
DES3DES	PD_COREAON	L4PER_RET_RST
FPKA	PD_COREAON	L4PER_RST
DSP1	PD_DSP1	DSP1_RST, DSP1_PWRON_RST, DSP1_RET_RST, DSP1_SYS_RST
DSP2	PD_DSP2	DSP2_RST, DSP2_PWRON_RST, DSP2_RET_RST, DSP2_SYS_RST
CTRL_MODULE_WKUP	PD_WKUPAON	WKUPAON_PWRON_RST
PRM	PD_WKUPAON	PRM_PWRON_RST, PRM_RST
PRCM_MPU	PD_WKUPAON	LPRM_PWRON_RST, LPRM_RST
GPIO1	PD_WKUPAON	WKUPAON_RST
KBD	PD_WKUPAON	WKUPAON_RST
COUNTER_32K	PD_WKUPAON	WKUPAON_RST, WKUPAON_SYS_PWRON_RST
TIMER1	PD_WKUPAON	WKUPAON_RST
TIMER12	PD_WKUPAON	WKUPAON_RST
WD_TIMER2	PD_WKUPAON	WKUPAON_RST
L4_WKUP interconnect	PD_WKUPAON	WKUPAON_RST
UART10	PD_WKUPAON	WKUPAON_RST
DCAN1	PD_WKUPAON	WKUPAON_RST
RTC_SS	PD_RTC	RTC_RST
EVE1	PD_EVE1	EVE1_CPU_RST, EVE1_PWRON_RST, EVE1_RST
EVE2	PD_EVE2	EVE2_CPU_RST, EVE2_PWRON_RST, EVE2_RST

Table 3-34 lists the reset sources that trigger the reset domains of the device.

Table 3-34. Reset Sources for the Reset Domains

Reset Domain	Reset Source	Reset Source Type
CM_CORE_AON_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
CM_CORE_AON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVAHD_RST	Global warm
	TSHUT_MPU_RST	Global warm
COREAON_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
COREAON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DPLL_IVA_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DPLL_DSP_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DPLL_EVE_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
MMAON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
MPUAON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
MPU_L2RSTDISABLE	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
CAM_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
CM_CORE_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
CM_CORE_RET_RST	GLOBAL_COLD_SW_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
CORE_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	SYS_PWRON_RST	Global cold
	ICEPICKPOR_RST	Global cold
CORE_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	SYS_PWRON_RST	Global cold
	ICEPICKPOR_RST	Global cold
CORE_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
CORE_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
CUSTEFUSE_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DLL_RST	DLL_FREQCHANGE_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DPLL_IVA_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DPLL_L3INIT_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DPLL_MPU_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSS_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DSS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU1_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
IPU1_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU1_RSTCTRL[2] RST_IPU	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU1_CPU0_RST	IPU1_ICECRUSHER0_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU1_RSTCTRL[0] RST_CPU0	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU1_CPU1_RST	IPU1_ICECRUSHER1_RST	Local Warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU1_RSTCTRL[1] RST_CPU1	Local warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU1_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU1_RSTCTRL[2] RST_IPU	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
IPU2_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
IPU2_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU2_RSTCTRL[2] RST_IPU	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU2_CPU0_RST	IPU2_ICECRUSHER0_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU2_RSTCTRL[0] RST_CPU0	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU2_CPU1_RST	IPU2_ICECRUSHER1_RST	Local Warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU2_RSTCTRL[1] RST_CPU1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IPU2_RST	GLOBAL_COLD_SW_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IPU2_RSTCTRL[2] RST_IPU	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
EMU_EARLY_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	SYS_PWRON_RST	Global cold
EMU_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
EMU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
GPU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IVA_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
IVA_RST	GLOBAL_COLD_SW_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_IVA_RSTCTRL[2] RST_LOGIC	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IVA_SEQ1_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	IVA_ICECRUSHER1_RST	Local warm
	MPU_WDT_RST	Global warm
	RM_IVA_RSTCTRL[0] RST_SEQ1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
IVA_SEQ2_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	IVA_ICECRUSHER2_RST	Local warm
	MPU_WDT_RST	Global warm
	RM_IVA_RSTCTRL[1] RST_SEQ2	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
L3INIT_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
L3INIT_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
L3INIT_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
MPU_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
MPU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
MPU_MA_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
MPU_MA_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
MPU_MA_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
SDMA_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SDMA_RESTORE_RST	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DSP1_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_DSP1_RSTCTRL[0] RST_DSP1_LRST	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	DSP1_EMU_RESET_REQ_TR	Local warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	ICEPICK_RST	Global warm
DSP1_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSP1_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_DSP1_RSTCTRL[1] RST_DSP1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DSP1_SYS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_DSP1_RSTCTRL[1] RST_DSP1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DSP2_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_DSP2_RSTCTRL[0] RST_DSP2_LRST	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	DSP2_EMU_RESET_REQ_TR	Local warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DSP2_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSP2_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_DSP2_RSTCTRL[1] RST_DSP2	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DSP2_SYS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	RM_DSP2_RSTCTRL[1] RST_DSP2	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
WKUPAON_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
WKUPAON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
WKUPAON_SYS_PWRON_RST	SYS_PWRON_RST	Global cold
PRM_RST	GLOBAL_COLD_SW_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
PRM_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
LPRM_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
LPRM_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
VPE_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
RTC_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
L4PER_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
L4PER_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
L4PER_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
EVE1_CPU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	EVE1_EMU_RESET_REQ_TR	Local warm
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	RM_EVE1_RSTCTRL[0] RST_EVE1_LRST	Local warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
EVE1_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	RM_EVE1_RSTCTRL[1] RST_EVE1	Local warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
EVE1_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
EVE2_CPU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	EVE2_EMU_RESET_REQ_TR	Local warm
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	RM_EVE2_RSTCTRL[0] RST_EVE2_LRST	Local warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
EVE2_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	RM_EVE2_RSTCTRL[1] RST_EVE2	Local warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
EVE2_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	SYS_PWRON_RST	Global cold
DSS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm
DSS_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	MPU_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	TSHUT_DSPEVE_RST	Global warm
	TSHUT_GPU_RST	Global warm
	TSHUT_IVA_RST	Global warm
	TSHUT_MPU_RST	Global warm
	ICEPICK_RST	Global warm

Table 3-35 lists the attributes of the reset manager associated with the reset domains. The clock to the reset manager, the delay count before release of reset, and the reset release stall conditions for the reset domains are listed.

Table 3-35. Reset Domains Attributes

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
CM_CORE_AON_PWRON_RST	WKUPAON_GCLK	0x2	None
CM_CORE_AON_RST	WKUPAON_GCLK	0x2	L4_ROOT_CLK clock is not active.
COREAON_PWRON_RST	WKUPAON_GCLK	0x0	None
COREAON_RST	WKUPAON_GCLK	0x0	None
DPLL_IVA_PWRON_RST	WKUPAON_GCLK	0x0	None
MMAON_RST	WKUPAON_GCLK	ResetTime2[(1)]	DSP_GFCLK is not active.
DPLL_DSP_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	None
DPLL_EVE_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	None
MPUAON_RST	WKUPAON_GCLK	ResetTime2[(1)]	MPU_GCLK is not active.
CUSTEFUSE_RST	WKUPAON_GCLK	ResetTime2[(1)]	CUSTEFUSE_SYS_GFCLK is not active.
CAM_RST	WKUPAON_GCLK	0x0	None
CM_CORE_PWRON_RET_RST	WKUPAON_GCLK	0x2	None
CM_CORE_RET_RST	WKUPAON_GCLK	0x2	L4_ICLK clock is not active.

Table 3-35. Reset Domains Attributes (continued)

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
CORE_PWRON_RET_RST	WKUPAON_GCLK	0x0	None
CORE_PWRON_RST	WKUPAON_GCLK	0x0	None
CORE_RET_RST	WKUPAON_GCLK	0x0	None
CORE_RST	WKUPAON_GCLK	0x0	None
DLL_RST	WKUPAON_GCLK	0x3	None
DMA_RET_RST	WKUPAON_GCLK	0x0	None
DPLL_IVA_PWRON_RST	WKUPAON_GCLK	0x0	None
DPLL_L3INIT_PWRON_RET_RST	WKUPAON_GCLK	0x0	None
DPLL_MPU_PWRON_RST	WKUPAON_GCLK	0x0	None
DSS_RET_RST	WKUPAON_GCLK	0x0	None
DSS_RST	WKUPAON_GCLK	0x0	None
IPU1_PWRON_RST	WKUPAON_GCLK	ResetTime2[1]	IPU1_GFCLK clock is not active, RM_IPU1_RSTCTRL[2] RST_IPU bit is set, and automatic restore is complete.
IPU1_RET_RST	WKUPAON_GCLK	ResetTime2[1]	IPU1_GFCLK clock is not active and the subsystem is reset.
IPU1_CPU0_RST	WKUPAON_GCLK	ResetTime2[1]	IPU1_GFCLK clock is not active and the subsystem is reset.
IPU1_CPU1_RST	WKUPAON_GCLK	ResetTime2[1]	IPU1_GFCLK clock is not active.
IPU1_RST	WKUPAON_GCLK	ResetTime2[1]	IPU1_GFCLK clock is not active and the subsystem is reset.
IPU_RET_RST	WKUPAON_GCLK	ResetTime2[1]	None
IPU_RST	WKUPAON_GCLK	0X0	None

Table 3-35. Reset Domains Attributes (continued)

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
IPU2_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	IPU2_GCLK clock is not active, RM_IPU2_RSTCTRL[2] RST_IPU bit is set, and automatic restore is complete.
IPU2_RET_RST	WKUPAON_GCLK	ResetTime2[(1)]	IPU2_GFCLK clock is not active and the subsystem is reset.
IPU2_CPU0_RST	WKUPAON_GCLK	ResetTime2[(1)]	IPU2_GFCLK clock is not active and the subsystem is reset.
IPU2_CPU1_RST	WKUPAON_GCLK	ResetTime2[(1)]	IPU2_GFCLK clock is not active.
IPU2_RST	WKUPAON_GCLK	ResetTime2[(1)]	IPU2_GFCLK clock is not active and the subsystem is reset.
EMU_EARLY_PWRON_RST	WKUPAON_ICLK	0x20	None
EMU_PWRON_RST	WKUPAON_ICLK	ResetTime2[(1)]	EMU_SYS_CLK clock is not active.
EMU_RST	WKUPAON_ICLK	ResetTime2[(1)]	EMU_SYS_CLK clock is not active.
GPU_RST	WKUPAON_GCLK	0x0	None
IVA_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	IVA_GCLK clock is not active and RM_IVA_RSTCTRL[2] RST_LOGIC bit is set.
IVA_RST	WKUPAON_GCLK	ResetTime2[(1)]	IVA_GCLK clock is not active.
IVA_SEQ1_RST	WKUPAON_GCLK	ResetTime2[(1)]	IVA_GCLK clock is not active and IVA and SL2 are idle.
IVA_SEQ2_RST	WKUPAON_GCLK	ResetTime2[(1)]	IVA_GCLK clock is not active and IVA and SL2 are idle.
L3INIT_PWRON_RST	WKUPAON_GCLK	0x0	None
L3INIT_RET_RST	WKUPAON_GCLK	0x0	None
L3INIT_RST	WKUPAON_GCLK	0x0	None
MPU_L2RSTDISABLE	WKUPAON_GCLK	ResetTime2 + 32[(1)]	MPU_RST is asserted high or PD_MPU is OFF.
MPU_MA_PWRON_RET_RST	WKUPAON_GCLK	ResetTime2[(1)]	MPU_DPLL_CLK clock is not active.
MPU_MA_RET_RST	WKUPAON_GCLK	ResetTime2[(1)]	MPU_DPLL_CLK clock is not active.
MPU_MA_RST	WKUPAON_GCLK	ResetTime2[(1)]	MPU_DPLL_CLK clock is not active.
MPU_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	MPU_DPLL_CLK clock is not active.
MPU_RST	WKUPAON_GCLK	ResetTime2[(1)]	MPU_DPLL_CLK clock is not active, the subsystem is reset, and automatic restore is complete.
DMA_RET_RST	WKUPAON_GCLK	0x0	None
DSP1_RST	WKUPAON_GCLK	ResetTime2[(1)]	DSP1_GFCLK clock is not active and the subsystem is reset.
DSP1_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	DSP1_GFCLK clock is not active, RM_DSP1_RSTCTRL[1] RST_DSP1 bit is cleared, and automatic restore is complete.

Table 3-35. Reset Domains Attributes (continued)

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
DSP1_RET_RST	WKUPAON_GCLK	ResetTime2(1)	DSP1_GFCLK clock is not active and the subsystem is reset.
DSP1_SYS_RST	WKUPAON_GCLK	ResetTime2(1)	DSP1_GFCLK clock is not active and the subsystem is reset.
DSP2_RST	WKUPAON_GCLK	ResetTime2(1)	DSP2_GFCLK clock is not active and the subsystem is reset.
DSP2_PWRON_RST	WKUPAON_GCLK	ResetTime2(1)	DSP2_GFCLK clock is not active, RM_DSP2_RSTCTRL[1] RST_DSP2 bit is cleared, and automatic restore is complete.
DSP2_RET_RST	WKUPAON_GCLK	ResetTime2(1)	DSP2_GFCLK clock is not active and the subsystem is reset.
DSP2_SYS_RST	WKUPAON_GCLK	ResetTime2(1)	DSP2_GFCLK clock is not active and the subsystem is reset.
WKUPAON_PWRON_RST	WKUPAON_GCLK	0x0	None
WKUPAON_RST	WKUPAON_GCLK	0x0	None
WKUPAON_SYS_PWRON_RST	WKUPAON_GCLK	0x0	None
PRM_PWRON_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
PRM_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
LPRM_PWRON_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
LPRM_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
VPE_RST	WKUPAON_GCLK	ResetTime2(1)	None
RTC_RST	WKUPAON_GCLK	ResetTime2(1)	None
L4PER_PWRON_RET_RST	WKUPAON_GCLK	0x0	None
L4PER_RET_RST	WKUPAON_GCLK	0x0	None
L4PER_RST	WKUPAON_GCLK	0x0	None
EVE1_CPU_RST	WKUPAON_GCLK	ResetTime2(1)	None
EVE1_PWRON_RST	WKUPAON_GCLK	ResetTime2(1)	EVE1_GFCLK is not active.
EVE1_RST	WKUPAON_GCLK	ResetTime2(1)	None
EVE2_CPU_RST	WKUPAON_GCLK	ResetTime2(1)	None
EVE2_PWRON_RST	WKUPAON_GCLK	ResetTime2(1)	EVE2_GFCLK is not active.
EVE2_RST	WKUPAON_GCLK	ResetTime2(1)	None

(1) ResetTime2 is set in the PRM_RSTTIME[14:10] RSTTIME2 bit field.

Note

WKUPAON_SYS_PWRON_RST is connected directly to the SYS_PWRON_RST source reset.

3.5.6 Reset Sequences

3.5.6.1 MPU Subsystem Power-On Reset Sequence

Figure 3-20 shows the POR sequence of the MPU subsystem.

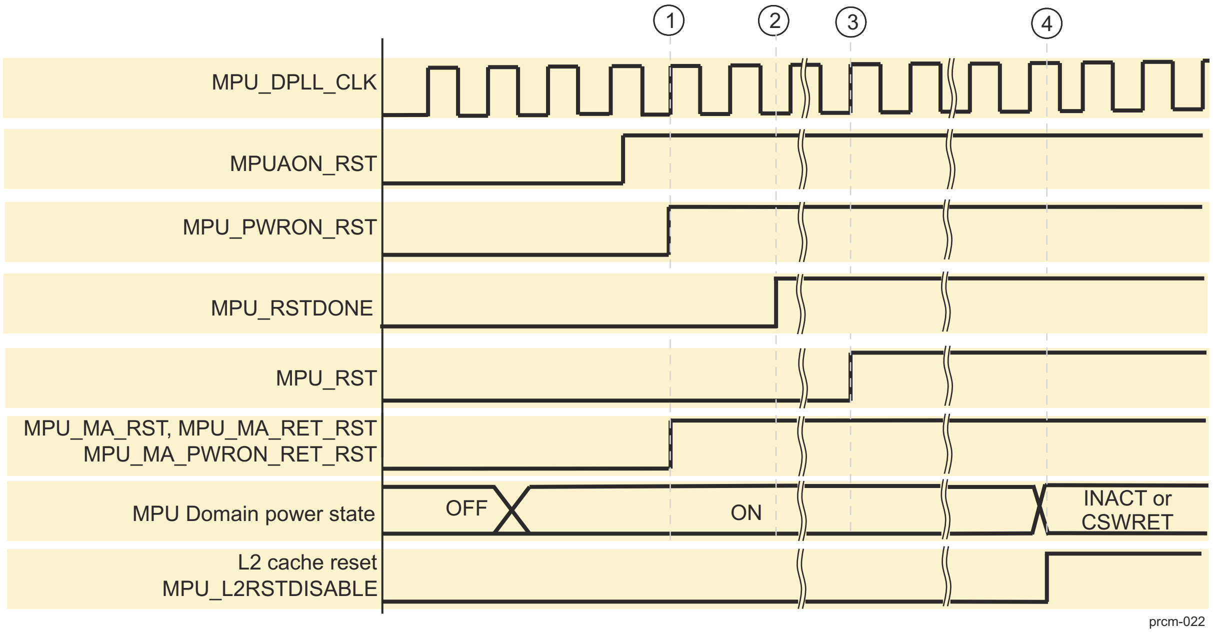


Figure 3-20. MPU Power-On Reset Sequence

The assumptions after power-on reset assertion are:

- The PRCM module provides the DPLL_MPU reference clock and the bypass clock.
- The PRCM module has released DPLL_MPU reset and DPLL_MPU is in bypass mode providing the clock (that is, bypass clock) to all the modules in the MPU subsystem.

The POR sequence is:

1. The PRCM module releases asynchronously in PD_MPUAON the MPUAON_RST reset to the INTC_MPU module in the MPU subsystem.
2. The PRCM module releases in PD_MPU the MPU_PWRON_RST (only after MPU_DPLL_CLK is active) to the MPU subsystem and waits until the subsystem completes its internal reset sequence. MA_MPU (Memory-adapter) dedicated reset signals (MPU_MA_RST, MPU_MA_RET_RST, and MPU_MA_PWRON_RET_RST) are released when MPU_DPLL_CLK is running (at the same time as MPU_PWRON_RST). When PRCM receive active MPU_RSTDONE signal from MPU, it de-asserts MPU_RST.
3. When the MPU subsystem internal reset sequence completes, the PRCM module releases in PD_MPU the MPU_RST signal and the MPU starts booting.
4. The PRCM module drives the MPU_L2RSTDISABLE signal low a minimum of 32 SYS_CLK cycles after MPU_RST is deasserted. The PRCM module keeps the MPU_L2RSTDISABLE signal low until the next power domain transition.

Note

- The reset to the L2 cache memory (MPU_L2RSTDISABLE) in the MPU subsystem is asserted during initial POR (that is, when the PD_MPU wakes up from OFF state). It is also asserted during local or global warm reset. However, it is not asserted when the PD_MPU wakes up from RETENTION state (that is, when the logic is OFF and L2 memory is in RETENTION state). This ensures that the L2 cache is retained on wakeup.
- The L1 cache memory in the MPU subsystem is not retained on PD_MPU wakeup.

3.5.6.2 MPU Subsystem Warm Reset Sequence

Figure 3-21 shows the warm reset sequence of the MPU subsystem.

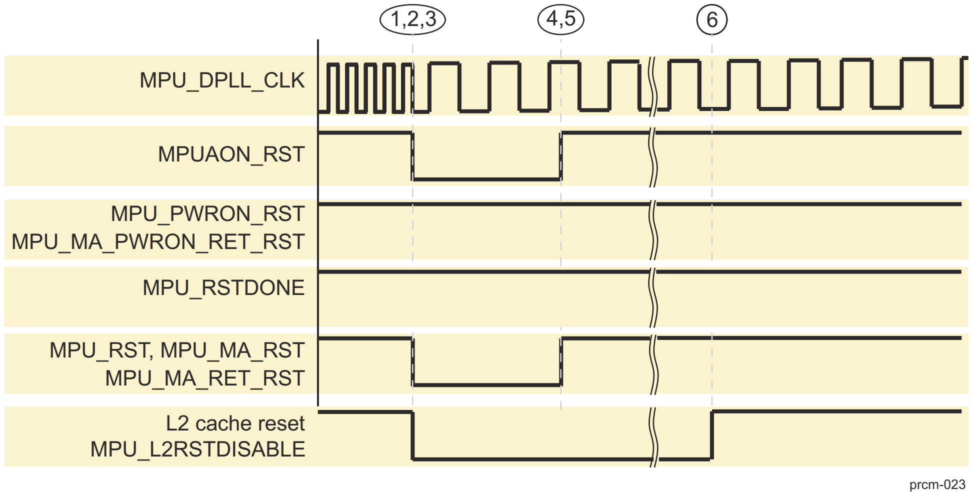


Figure 3-21. MPU Warm Reset Sequence

The assumptions are:

- The DPLL_MPU is locked and is providing the clock to the MPU subsystem.
- A global warm reset to the MPU subsystem is asserted.

The warm reset sequence is:

1. The PRCM module asserts in PD_MPUAON the MPUAON_RST reset to the INTC_MPU module in the MPU subsystem. The MPU DPLL is locked.
2. The PRCM module asserts in PD_MPU the MPU_RST reset to the MPU subsystem and also asserts MPU_MA_RST and MPU_MA_RET_RST resets to the MA_MPU module.
3. The PRCM module resets the L2 cache memory in the MPU subsystem by asserting its reset.
4. The PRCM module releases the MPUAON_RST reset to the INTC_MPU module. The MPU DPLL is in bypass mode.
5. The PRCM module releases MPU_RST, MPU_MA_RST, and MPU_MA_RET_RST only after DPLL_MPU is in bypass mode and MPU_DPLL_CLK is stable and active.
6. The PRCM module drives the MPU_L2RSTDISABLE signal low a minimum of 32 SYS_CLK cycles after MPU_RST is deasserted. The PRCM keeps the MPU_L2RSTDISABLE signal low until the next power domain transition.

3.5.6.3 MPU Subsystem Reset Sequence on Sleep and Wake-Up Transitions From RETENTION State

Figure 3-21 shows the sleep and wake-up transitions reset sequence from the RETENTION state of the MPU subsystem.

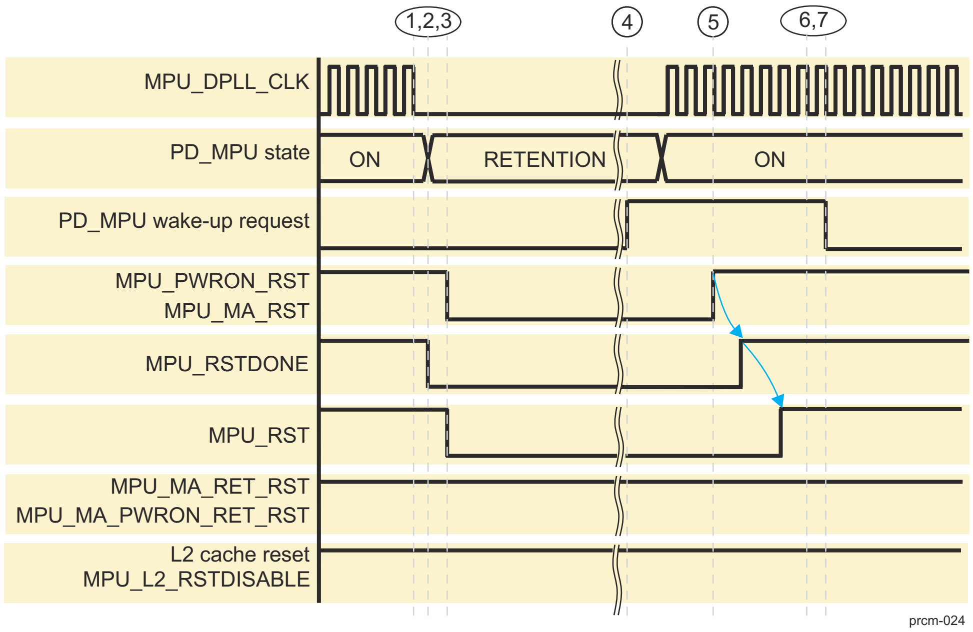


Figure 3-22. MPU Reset Sequence on Sleep and Wake-Up Transition

The assumption is:

- The DPLL_MPU is locked and is providing the clock to the MPU subsystem.

The sleep and wake-up transitions reset sequence is:

1. The PRCM module gates MPU_DPLL_CLK to the MPU subsystem.
2. The PRCM module switches PD_MPU to OSWR state.
3. The PRCM module asserts MPU_PWRON_RST and MPU_RST resets to the MPU subsystem, and asserts MPU_MA_RST to the MA_MPU module. The entire logic in the PD_MPU is held in reset. The reset to the L2 cache memory in the MPU subsystem is not asserted if the logic in PD_MPU is held in reset.
4. The PRCM module resets the L2 cache memory in the MPU subsystem by asserting its reset.
5. The PRCM module releases MPU_RST when the MPU_DPLL_CLK is stable and active. The PRCM deasserts the MPU_PWRON_RST, MPU_MA_RST. When PRCM receives active MPU_RSTDONE signal from MPU, it de-asserts MPU_RST.
6. During the wakeup from RET state, the MPU_L2RSTDISABLE signal must be maintained at active high so the L2 cache will not be reset when the MPU is reset.

3.5.6.4 IVA Subsystem Power-On Reset Sequence

Figure 3-23 shows the power-on reset sequence of the IVA subsystem.

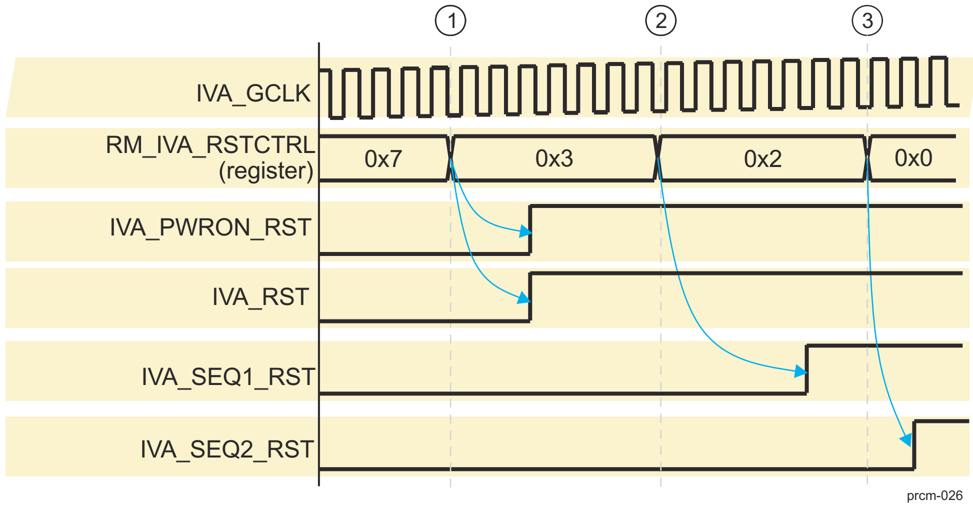


Figure 3-23. IVA Power-On Reset Sequence

The power-on reset to IVA is applied when PD_IVA is powered. The assumptions after power-on reset assertion are:

- The PRCM module provides the IVA_GCLK functional clock to the IVA subsystem, and it has been enabled by MPU software control.

The power-on reset sequence is:

1. Software clears the RM_IVA_RSTCTRL[2] RST_LOGIC bit. This causes the PRCM module to release the IVA_PWRON_RST reset used inside IVA mainly to reset the emulation logic and the IVA_RST reset used to reset all logic inside IVA. Then software can download data into TCM memory while keeping the sequencer CPUs under reset.
2. When the TCM memory is initialized, software clears the RM_IVA_RSTCTRL[0] RST_SEQ1 bit. This releases IVA_SEQ1_RST reset to the Sequencer1 CPU.
3. Similarly, software can clear the RM_IVA_RSTCTRL[1] RST_SEQ2 bit. This releases IVA_SEQ2_RST reset to the Sequencer2 CPU.

3.5.6.5 IVA Subsystem Software Warm Reset Sequence

Figure 3-24 shows the software warm reset sequence of the IVA subsystem.

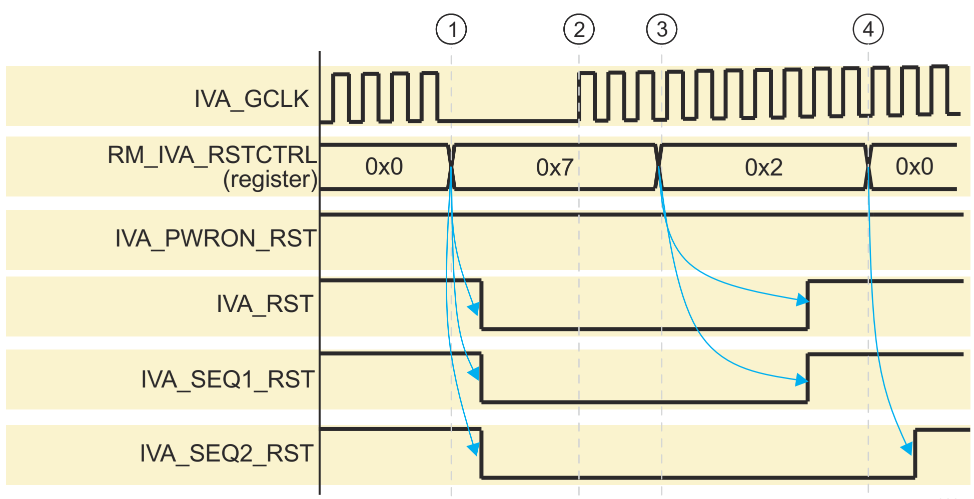


Figure 3-24. IVA Software Warm Reset Sequence

Before asserting the software reset to the IVA subsystem the MPU software must ensure that:

- IVA sequencer CPUs are in IDLE state (CM_IVA_IVA_CLKCTRL[17:16] IDLEST).
- The IVA subsystem is in STANDBY state (CM_IVA_IVA_CLKCTRL[18] STBYST).
- The functional clock to the IVA subsystem has been gated by the PRCM module (CM_IVA_CLKSTCTRL[8] CLKACTIVITY_IVA_GCLK).

The software reset sequence is:

1. The MPU software sets the RM_IVA_RSTCTRL[2] RST_LOGIC, RM_IVA_RSTCTRL[1] RST_SEQ2, and RM_IVA_RSTCTRL[0] RST_SEQ1 bits. This causes the PRCM module to assert the IVA_RST, IVA_SEQ1_RST, and IVA_SEQ2_RST resets to the IVA subsystem. The IVA_PWRON_RST remains deasserted.
2. The MPU software enables the functional clock to the IVA subsystem.
3. The MPU software clears the RM_IVA_RSTCTRL[2] RST_LOGIC and RM_IVA_RSTCTRL[0] RST_SEQ1 bits. This causes the PRCM module to release the IVA_RST and IVA_SEQ1_RST resets to the IVA subsystem.
4. The MPU software clears the RM_IVA_RSTCTRL[1] RST_SEQ2 bit. This releases the IVA_SEQ2_RST reset to the Sequencer2 CPU.

3.5.6.6 DSP1 Subsystem Power-On Reset Sequence

Figure 3-25 shows the power-on reset sequence of the DSP1 subsystem.

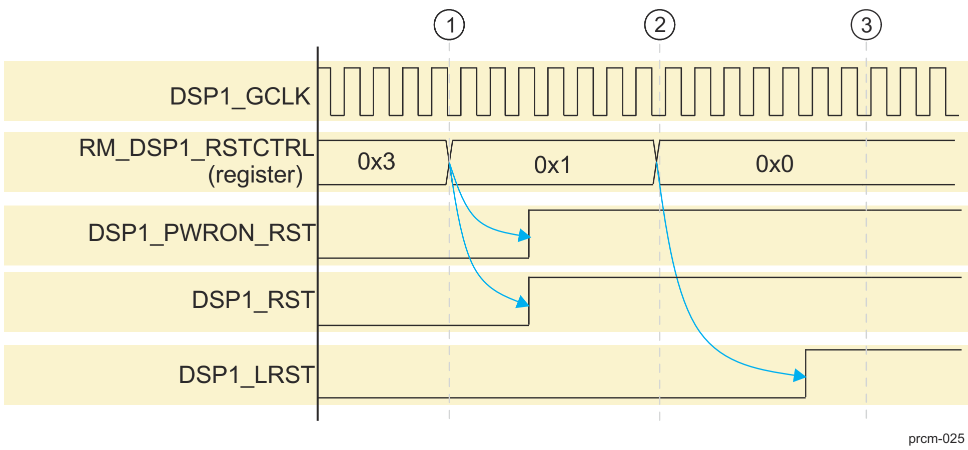


Figure 3-25. DSP1 Subsystem Power-On Reset Sequence

The power-on reset to DSP1 is applied when PD_DSP1 is powered. The assumptions ofter power-on reset assertion are:

- PD_DSP1 is on.
- The PRCM module provides DSP1_GCLK functional clock to the DSP subsystem, and it has been enabled by MPU software control.

The Power-On Reset sequence is:

1. Software clears the RM_DSP1_RSTCTRL[1] RST_DSP1 bit. This causes the PRCM module to release the DSP1_PWRON_RST used inside DSP1 mainly to reset the emulation logic and the DSP1_RST used to reset all logic inside DSP1. Then software can download data into TCM memory while keeping the CPU under reset.
2. When the memory is initialized, software clears the RM_DSP1_RSTCTRL[0] RST_DSP1_LRST bit. This release DSP1_LRST to the local CPU inside DSP subsystem.

3.5.6.7 DSP1 Subsystem Software Warm Reset Sequence

Figure 3-26 shows the software warm reset sequence of the DSP1 subsystem.

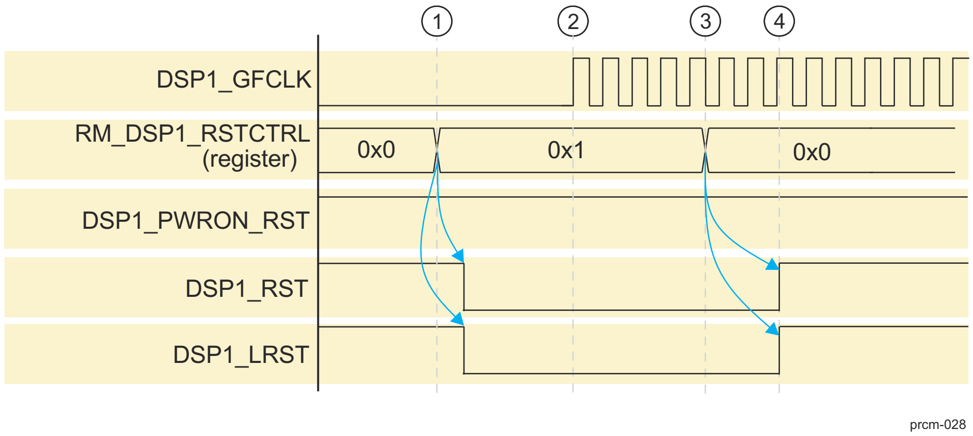


Figure 3-26. DSP1 Subsystem Software Warm Reset Sequence

Before asserting the software reset to the DSP subsystem, the MPU software must ensure that:

- The DSP CPUs are in IDLE state (CM_DSP1_DSP1_CLKCTRL[17:16] IDLEST).
- The DSP subsystem is in STANDBY state (CM_DSP1_DSP1_CLKCTRL[18] STBYST).
- The functional clock to the DSP subsystem has been gated by the PRCM module (CM_DSP1_CLKSTCTRL[8] CLKACTIVITY_DSP1_GFCLK)

The software reset sequence is:

1. MPU software sets the RM_DSP1_RSTCTRL[1] RST_DSP1 = 0 and RM_DSP1_RSTCTRL[0] RST_DSP1_LRST = 1. This causes the PRCM module to assert DSP1_RST, DSP1_LRST to the DSP subsystem. The DSP1_PWRON_RST remains deasserted.
2. The MPU software enables the functional clock to the DSP subsystem.
3. The MPU software cleares theRM_DSP1_RSTCTRL[1] RST_DSP1 and RM_DSP1_RSTCTRL[0] RST_DSP1_LRST bits. This causes the PRCM module to release DSP1_RST and DSP1_LRST to the DSP subsystem.

3.5.6.8 DSP2 Subsystem Power-On Reset Sequence

Figure 3-27 shows the power-on reset sequence of the DSP2 subsystem.

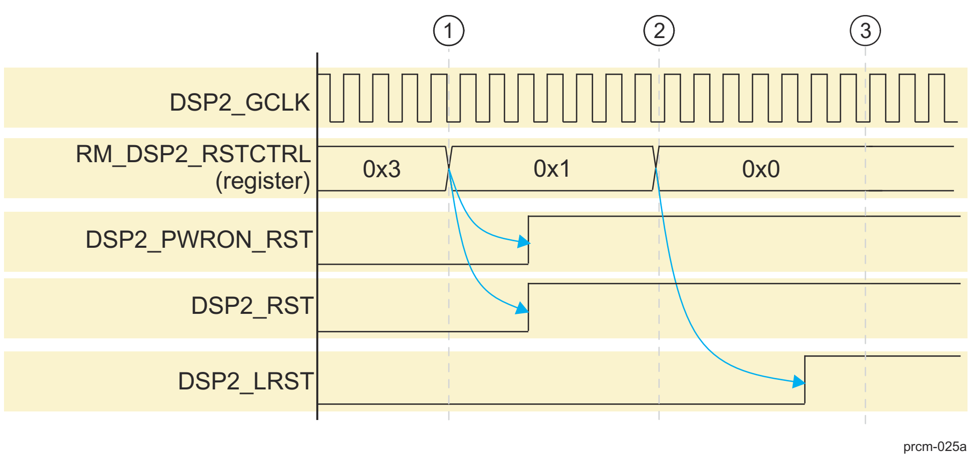


Figure 3-27. DSP2 Subsystem Power-On Reset Sequence

The power-on reset to DSP2 is applied when PD_DSP2 is powered. The assumptions ofter power-on reset assertion are:

- PD_DSP2 is on.

- The PRCM module provides DSP2_GCLK functional clock to the DSP subsystem, and it has been enabled by MPU software control.

The Power-On Reset sequence is:

1. Software clears the RM_DSP2_RSTCTRL[1] RST_DSP2 bit. This causes the PRCM module to release the DSP2_PWRON_RST used inside DSP2 mainly to reset the emulation logic and the DSP2_RST used to reset all logic inside DSP2. Then software can download data into TCM memory while keeping the CPU under reset.
2. When the memory is initialized, software clears the RM_DSP2_RSTCTRL[0] RST_DSP2_LRST bit. This release DSP2_LRST to the local CPU inside DSP subsystem.

3.5.6.9 DSP2 Subsystem Software Warm Reset Sequence

Figure 3-26 shows the software warm reset sequence of the DSP2 subsystem.

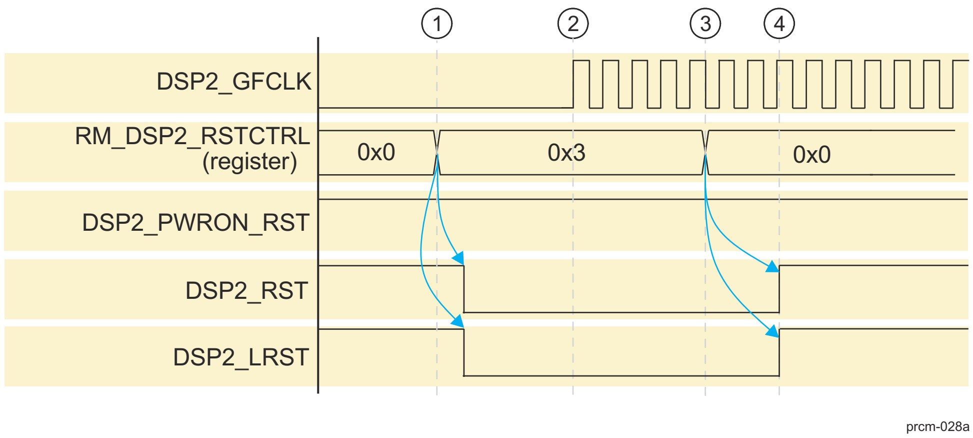


Figure 3-28. DSP2 Subsystem Software Warm Reset Sequence

Before asserting the software reset to the DSP subsystem, the MPU software must ensure that:

- The DSP CPUs are in IDLE state (CM_DSP2_DSP2_CLKCTRL[17:16] IDLEST).
- The DSP subsystem is in STANDBY state (CM_DSP2_DSP2_CLKCTRL[18] STBYST).
- The functional clock to the DSP subsystem has been gated by the PRCM module (CM_DSP2_CLKSTCTRL[8] CLKACTIVITY_DSP2_GFCLK)

The software reset sequence is:

1. MPU software sets the RM_DSP2_RSTCTRL[1] RST_DSP2 = 0 and RM_DSP2_RSTCTRL[0] RST_DSP2_LRST = 1. This causes the PRCM module to assert DSP2_RST, DSP2_LRST to the DSP subsystem. The DSP2_PWRON_RST remains deasserted.
2. The MPU software enables the functional clock to the DSP subsystem.
3. The MPU software cleares theRM_DSP2_RSTCTRL[1] RST_DSP2 and RM_DSP2_RSTCTRL[0] RST_DSP2_LRST bits. This causes the PRCM module to release DSP2_RST and DSP2_LRST to the DSP subsystem.

3.5.6.10 IPU1 Subsystem Power-On Reset Sequence

Figure 3-29 shows the power-on reset sequence of the IPU1 subsystem.

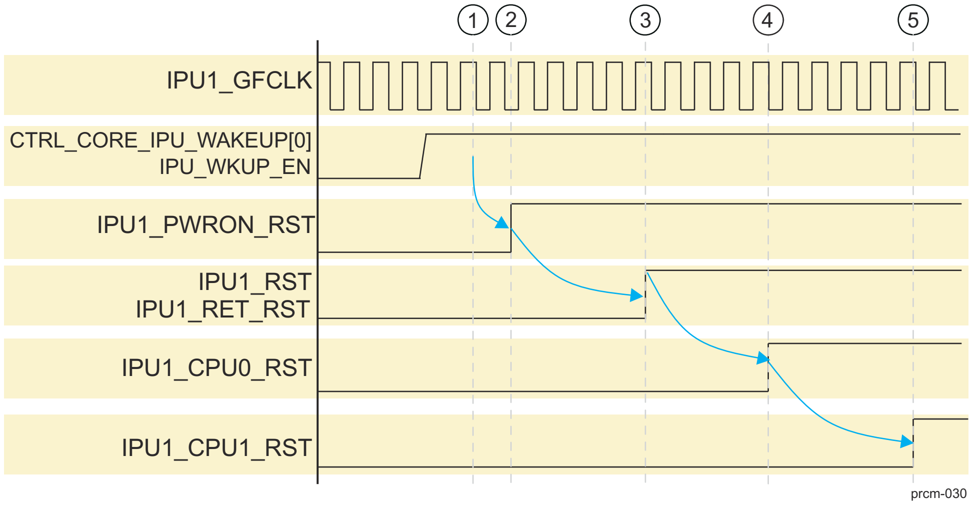


Figure 3-29. IPU1 Power-On Reset Sequence

The assumptions on power-on reset assertion are:

- The IPU subsystem is held in reset by the PRCM module and the following are asserted:
  - IPU1_PWRON_RST
  - IPU1_RET_RST
  - IPU1_CPU0_RST
  - IPU1_CPU1_RST
  - IPU1_RST

The power-on reset sequence is:

1. Software clears the RM_IPU1_RSTCTRL[2] RST_IPU bit in the PRCM module register to release the IPU shared cache and CACHE_MMU_IPU from reset.
2. The PRCM module releases IPU1_PWRON_RST once the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) reaches its limit and the IPU1_GFCLK is running. Upon deassertion of the POR signal, the IPU subsystem starts the CPU and CACHE_MMU_IPU initialization sequence.
3. When the reset sequence of Step 2 completes and the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the IPU1_RST and IPU1_RET_RST signals.
4. MPU software must configure CACHE_MMU_IPU once CACHE_MMU_IPU is out of reset.
   After CACHE_MMU_IPU configuration and cache initialization is done, MPU software clears the RM_IPU1_RSTCTRL[0] RST_CPU0 bit in the PRCM module register.
5. The PRCM module releases IPU_CPU0_RST, which causes IPU_C0 to start booting.
6. MPU software can clear the RM_IPU1_RSTCTRL[1] RST_CPU1 bit in the PRCM module register so that the PRCM module releases the IPU_CPU1_RST to IPU_C1.

3.5.6.11 IPU1 Subsystem Software Warm Reset Sequence

Figure 3-30 shows the software warm reset sequence of the IPU1 subsystem.

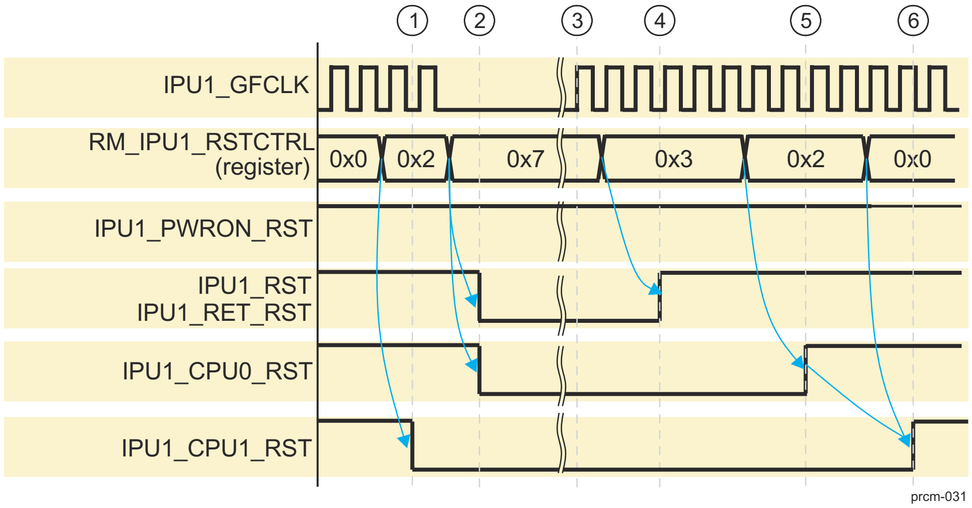


Figure 3-30. IPU1 Subsystem Software Warm Reset Sequence

For doing the software reset of IPU1, the MPU software must ensure that IPU CPUs (IPU1_C0 and IPU1_C1) are in IDLE state and clock is gated

The software warm reset sequence is:

1. When IPU1_C1 is in IDLE state, IPU1_C0 software or MPU software sets the RM_IPU1_RSTCTRL[1] RST_CPU1 bit. The PRCM module asserts the IPU1_CPU1_RST reset signal to IPU1_C1.
2. When IPU1_C0 is in IDLE state, the MPU software sets the RM_IPU1_RSTCTRL[2] RST_IPU and RM_IPU1_RSTCTRL[0] RST_CPU0 bits.
3. The PRCM module asserts the IPU1_RST, IPU1_RET_RST, and IPU1_CPU0_RST reset signals. The IPU1_PWRON_RST remains deasserted in this case.
4. The MPU software reenables the IPU1_GFCLK and the initialization sequence starts inside the IPU subsystem. Software clears the RM_IPU1_RSTCTRL[2] RST_IPU bit and RM_IPU1_RSTCTRL[0] RST_CPU0 bit in the PRCM module register.
5. When the reset sequence of Step 4 completes and the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the IPU1_RST, IPU1_RET_RST, and IPU1_CPU0_RST reset signals. IPU1_C0 starts rebooting.
6. Software can then clear the RM_IPU1_RSTCTRL[1] RST_CPU1 bit in the PRCM module register. The PRCM module releases the IPU1_CPU1_RST reset signal to IPU1_C1 to start booting.

3.5.6.12 IPU2 Subsystem Power-On Reset Sequence

Figure 3-31 shows the power-on reset sequence of the IPU2 subsystem.

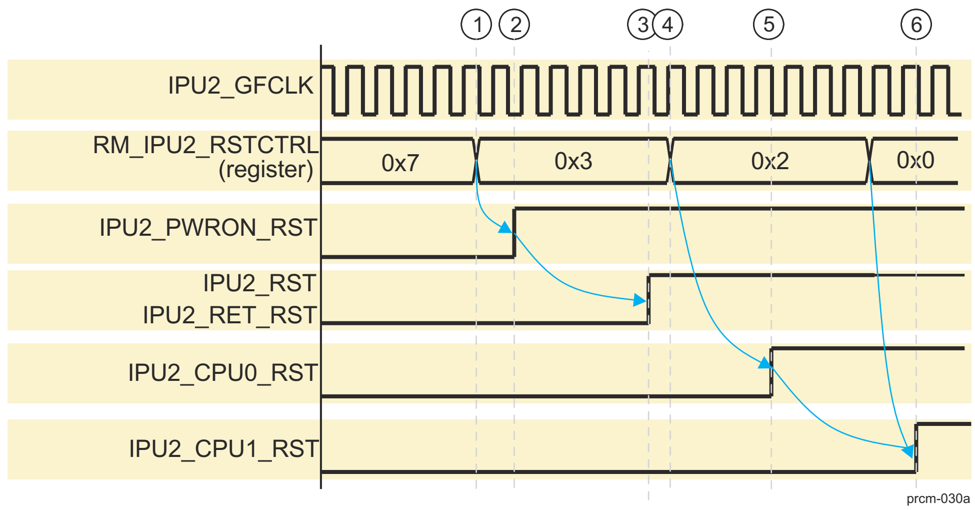


Figure 3-31. IPU2 Power-On Reset Sequence

The assumptions on power-on reset assertion are:

- The IPU subsystem is held in reset by the PRCM module and the following are asserted:
  - IPU2_PWRON_RST
  - IPU2_RET_RST
  - IPU2_CPU0_RST
  - IPU2_CPU1_RST
  - IPU2_RST

The power-on reset sequence is:

1. Software clears the RM_IPU2_RSTCTRL[2] RST_IPU bit in the PRCM module register to release the IPU shared cache and CACHE_MMU_IPU from reset.
2. The PRCM module releases IPU2_PWRON_RST once the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) reaches its limit and the IPU2_GFCLK is running. Upon deassertion of the POR signal, the IPU subsystem starts the CPU and CACHE_MMU_IPU initialization sequence.
3. When the reset sequence of Step 2 completes and the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the IPU_MMU_CACHE_RST and IPU2_RET_RST signals.
4. MPU software must configure CACHE_MMU_IPU once CACHE_MMU_IPU is out of reset.
   After CACHE_MMU_IPU configuration and cache initialization is done, MPU software clears the RM_IPU2_RSTCTRL[0] RST_CPU0 bit in the PRCM module register.
5. The PRCM module releases IPU2_CPU0_RST, which causes IPU_C0 to start booting.
6. MPU software can clear the RM_IPU2_RSTCTRL[1] RST_CPU1 bit in the PRCM module register so that the PRCM module releases the IPU2_CPU1_RST to IPU_C1.

3.5.6.13 IPU2 Subsystem Software Warm Reset Sequence

Figure 3-32 shows the software warm reset sequence of the IPU2 subsystem.

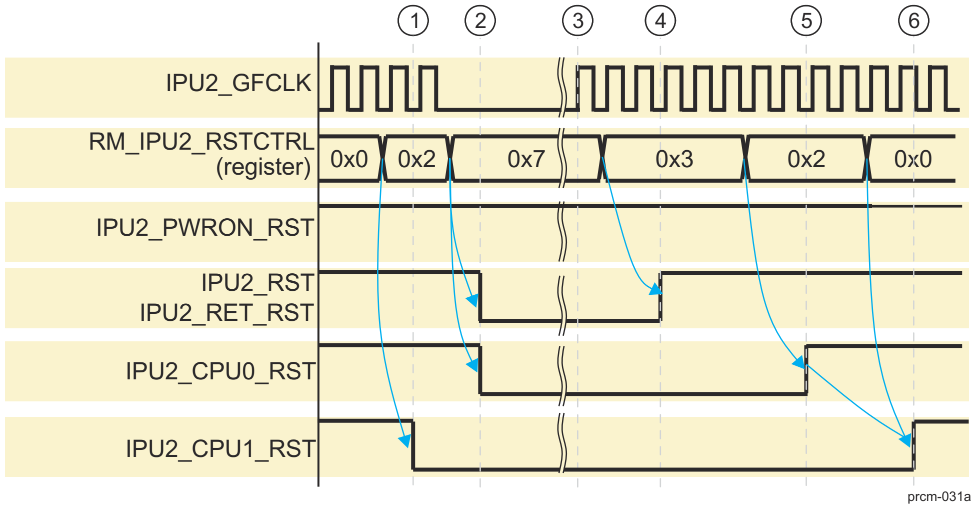


Figure 3-32. IPU2 Subsystem Software Warm Reset Sequence

For doing the software reset of IPU2, the MPU software must ensure that IPU CPUs (IPU2_C0 and IPU2_C1) are in IDLE state and clock is gated

The software warm reset sequence is:

1. When IPU2_C1 is in IDLE state, IPU2_C0 software or MPU software sets the RM_IPU2_RSTCTRL[1] RST_CPU1 bit. The PRCM module asserts the IPU2_CPU1_RST reset signal to IPU2_C1.
2. When IPU2_C0 is in IDLE state, the MPU software sets the RM_IPU2_RSTCTRL[2] RST_IPU and RM_IPU2_RSTCTRL[0] RST_CPU0 bits.
3. The PRCM module asserts the IPU2_RST, IPU2_RET_RST, and IPU2_CPU0_RST reset signals. The IPU2_PWRON_RST remains deasserted in this case.
4. The MPU software reenables the IPU2_GFCLK and the initialization sequence starts inside the IPU subsystem. Software clears the RM_IPU2_RSTCTRL[2] RST_IPU bit and RM_IPU2_RSTCTRL[0] RST_CPU0 bit in the PRCM module register.
5. When the reset sequence of Step 4 completes and the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the IPU2_RST, IPU2_RET_RST, and IPU2_CPU0_RST reset signals. IPU2_C0 starts rebooting.
6. Software can then clear the RM_IPU2_RSTCTRL[1] RST_CPU1 bit in the PRCM module register. The PRCM module releases the IPU2_CPU1_RST reset signal to IPU2_C1 to start booting.

3.5.6.14 EVE1 Subsystem Power-On Reset Sequence

Figure 3-33 shows the power-on reset sequence of the EVE1 subsystem.

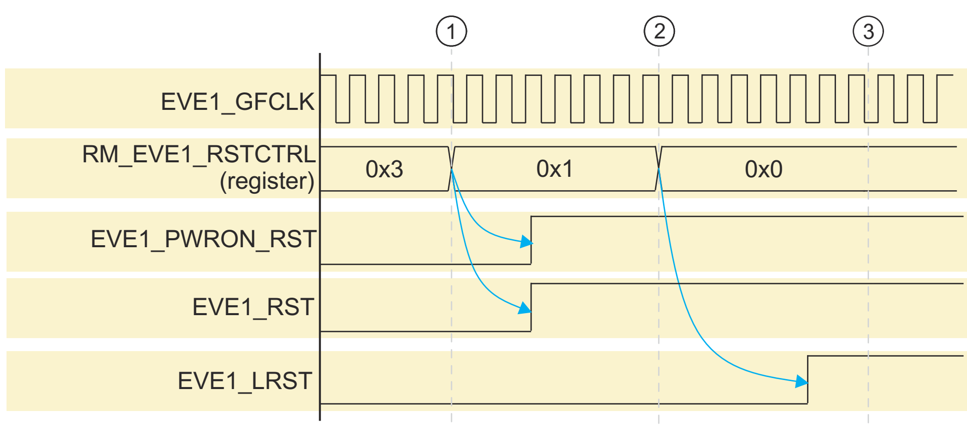


Figure 3-33. EVE1 Subsystem Power-On Reset Sequence

The power-on reset to EVE1 is applied when PD_EVE1 is powered. The assumptions on power-on reset assertion are:

- The PRCM module provides the EVE1_GFCLK functional clock to the EVE subsystem, and it has been enabled by MPU software control.

The power-on reset sequence is:

1. Software clears the RM_EVE1_RSTCTRL[1] RST_EVE1 bit. This causes the PRCM module to release the EVE1_PWRON_RST and the EVE1_RST signals. Then software can download data into TCM memory while keeping the sequencer CPUs under reset.
2. When the TCM memory is initialized, software clears the RM_EVE1_RSTCTRL[0] RST_EVE1_LRST bit .This release EVE1_LRST to the local CPU inside EVE subsystem.

3.5.6.15 EVE1 Subsystem Software Warm Reset Sequence

Figure 3-34 shows the software warm reset sequence of the EVE1 subsystem.

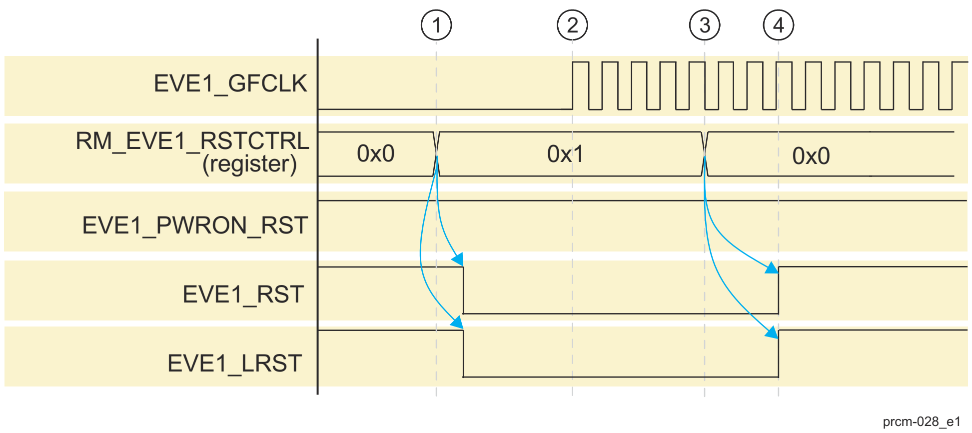


Figure 3-34. EVE1 Subsystem Software Warm Reset Sequence

For doing the software reset of EVE1 the MPU software must ensure that EVE1 CPU is in IDLE state and EVE1 is in STANDBY state and the functional clock to EVE1 has been gated.

The software warm reset sequence is:

1. MPU software sets the RM_EVE1_RSTCTRL register to 0x1. This causes the PRCM module to asset the EVE1_RST, EVE1_LRST to the EVE subsystem. The EVE1_PWRON_RST remains deasserted.
2. The MPU software enables the functional clock to the EVE1 subsystem.
3. The MPU software cleares the RM_EVE1_RSTCTRL[1] RST_EVE1 and RM_EVE1_RSTCTRL[0] RST_EVE1_LRST bits. This causes the PRCM module to release EVE1_RST and EVE1_LRST to the EVE subsystem.

3.5.6.16 EVE2 Subsystem Power-On Reset Sequence

Figure 3-35 shows the power-on reset sequence of the EVE2 subsystem.

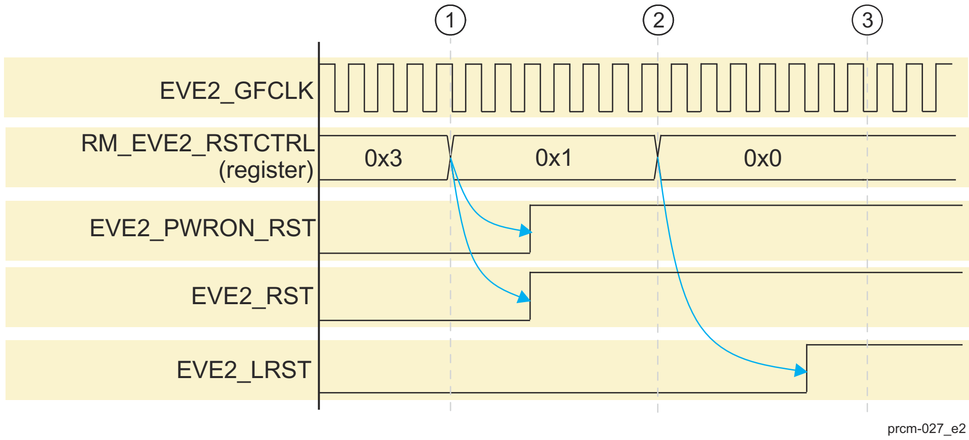


Figure 3-35. EVE2 Subsystem Power-On Reset Sequence

The power-on reset to EVE2 is applied when PD_EVE2 is powered. The assumptions on power-on reset assertion are:

- The PRCM module provides the EVE2_GFCLK functional clock to the EVE subsystem, and it has been enabled by MPU software control.

The power-on reset sequence is:

1. Software clears the RM_EVE2_RSTCTRL[1] RST_EVE2 bit.This causes the PRCM module to release the EVE2_PWRON_RST used inside EVE2 to reset mainly the emulation logic and EVE2_RST used to reset all logic inside EVE2. Then software can download data into TCM memory while keeping the CPU under reset.
2. When the TCM memory is initialized, software clears the RM_EVE2_RSTCTRL[0] RST_EVE2_LRST bit. This releases EVE2_LRST to the local CPU inside EVE subsystem.

3.5.6.17 EVE2 Subsystem Software Warm Reset Sequence

Figure 3-36 shows the software warm reset sequence of the EVE2 subsystem.

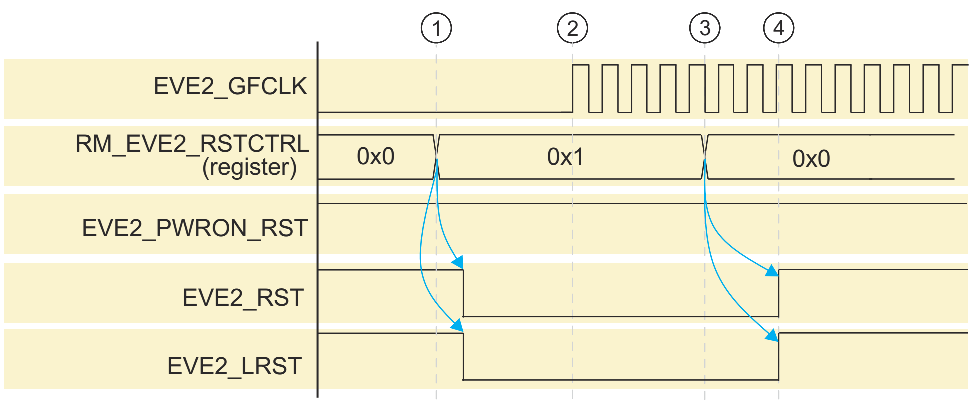


Figure 3-36. EVE2 Subsystem Software Warm Reset Sequence

For doing the software reset of EVE2 the MPU software must ensure that EVE2 CPU is in IDLE state and EVE2 is in STANDBY state and the functional clock to EVE2 has been gated.

The software warm reset sequence is:

1. MPU software sets the RM_EVE2_RSTCTRL register to 0x1. This causes the PRCM module to asset the EVE2_RST, EVE2_LRST to the EVE subsystem. The EVE2_PWRON_RST remains deasserted.
2. The MPU software enables the functional clock to the EVE2 subsystem.

3. The MPU software cleares the RM_EVE2_RSTCTRL[1] RST_EVE2 and RM_EVE2_RSTCTRL[0] RST_EVE2_LRST bits. This causes the PRCM module to release EVE2_RST and EVE2_LRST to the EVE subsystem.

3.5.6.18 Global Warm Reset Sequence

This section describes the global warm reset sequence.

Figure 3-37 shows the global warm reset sequence.

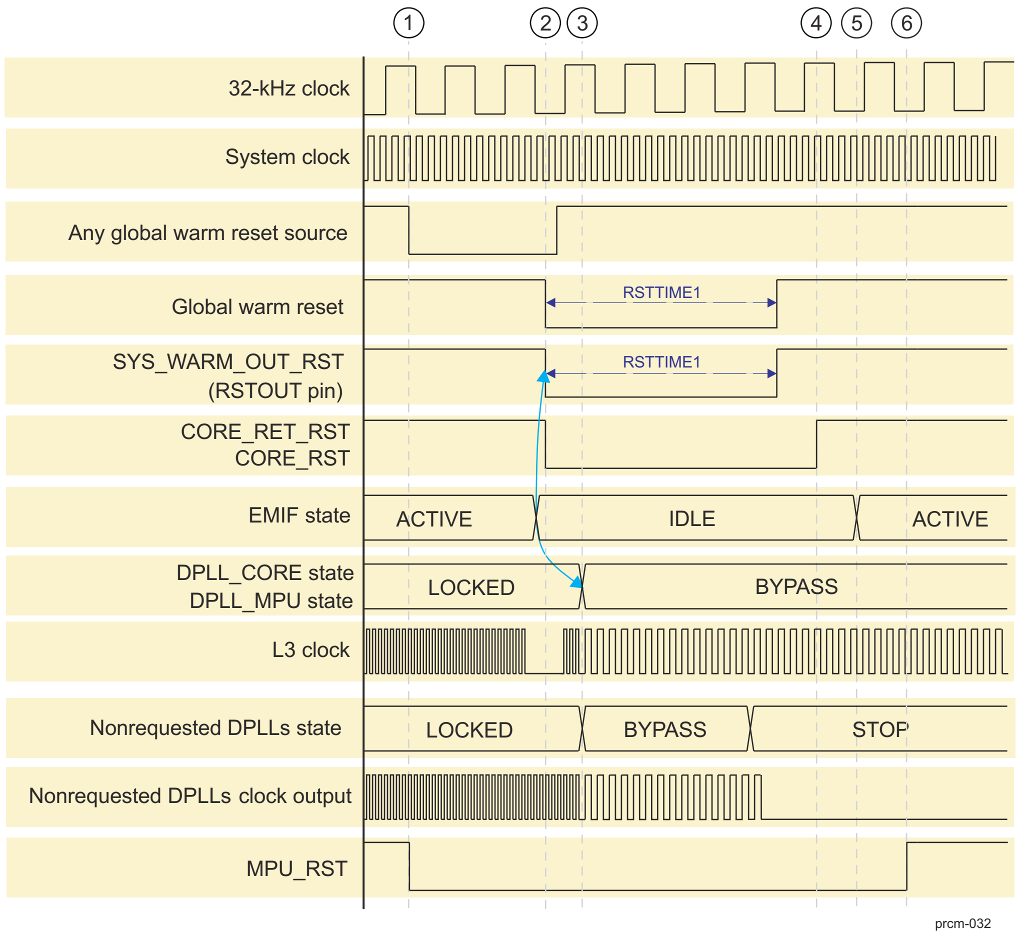


Figure 3-37. Global Warm Reset Sequence

The assumptions are:

- All the logic and memory voltage sources are at their nominal voltage levels.
- The device is active:
  - All resets are released.
  - MPU, CORE, and IVA DPLLs are locked.

The steps of a global warm reset sequence are:

1. On assertion of any global warm reset source the PRM signals the EMIF that a global warm reset event has occurred. The EMIF initiates the transition to IDLE state. The PRCM module delays global warm reset to the device for a minimum of 16 L3 clock cycles so that the EMIF switches to IDLE state and switches the external SDRAM to self-refresh mode.
2. The reset managers in the PRM assert the following resets:
   - The external warm reset SYS_WARM_RST (rstoutn pin).
   - All power domain warm resets are asserted.

- The PRM and CM registers, sensitive to warm reset, are asynchronously reset.
- DPLL hardware resets are not asserted.
  - DPLL_MPU transitions to bypass mode.
  - DPLL_CORE transitions to bypass mode once the EMIF switches to IDLE state.
  - DPLL_IVA, DPLL_PER, DPLL_USB, DPLL_DSP, DPLL_EVE, DPLL_GPU and DPLL_DDR transition to idle bypass low-power mode. Then, as clock signals are no more requested, they are gated and these DPLLs go to stop mode.
  - DPLL_ABE configuration is not changed.
  - DPLL_GMAC configuration is not changed.
- CM gates the clocks that are not needed, as per their default reset setting in the associated registers.

3. The device warm reset (internal and rstoutn pin) duration is set up by the PRM_RSTTIME[9:0] RSTTIME1 bit field. It defines the global warm reset duration in number of FUNC_32K_CLK clock cycles. Default value loaded after POR is 6 clock cycles. During this time, the DPLL_ABE control registers are reset and DPLL_ABE transitions to bypass mode when the system clock restarts and the DPLL_ABE outputs are no longer used.
4. The CORE power domain is released from reset (that is, warm reset-sensitive modules in the CORE power domain).
5. The PRCM module switches the EMIF from IDLE state back to ACTIVE state.
6. PD_MPU is released from reset when the clocks to the MPU subsystem are active. The MPU reboots.

Note

- The PD_DSP and PD_IVA, PD_EVE, PD_IPU2 power domains are held under reset after global warm reset by assertion of the software source of the reset.
- The following are held under reset after global warm reset until the PRCM module enables their interface clock:
  - PD_L4PER
  - PD_L3INIT
  - PD_DSS
  - PD_GPU
  - PD_CAM

3.6 Clock Management Functional Description

3.6.1 Overview

The PRCM module provides the control for clock generation, division, distribution, synchronization, and gating. It distributes the clock sources to all modules in the device. For information about the clock-management functional architecture of the device, see Section 3.1.1.1, *Clock Management*.

The PRCM module provides clocks to the internal DPLLs for internal high-frequency clock generation. Clock division and gating are handled by the PRM, CM_CORE_AON, and CM_CORE sections of the PRCM module. Figure 3-38 shows the high-level clock-management scheme in the device.

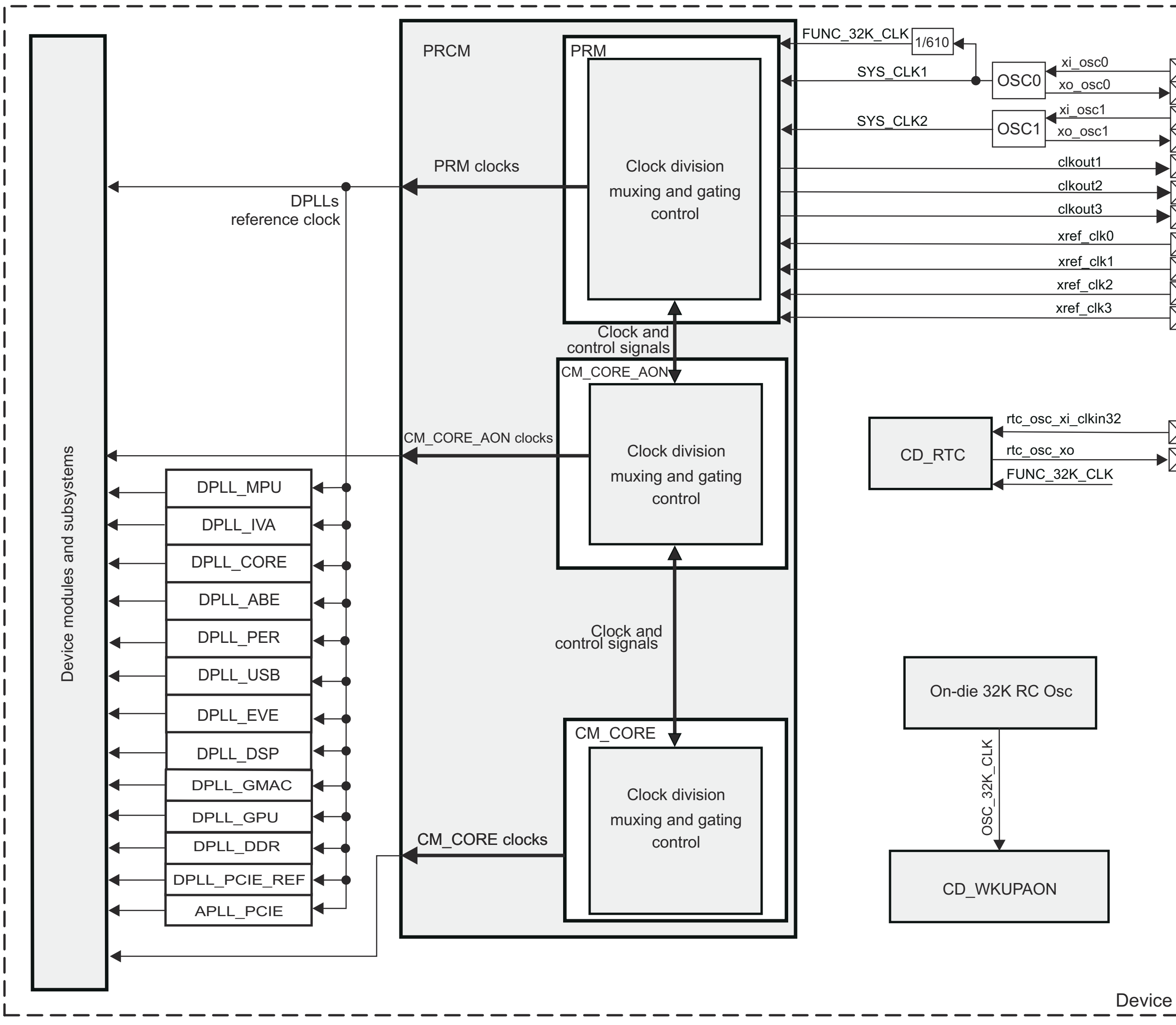


Figure 3-38. PRCM Module Clock Manager Overview

3.6.2 External Clock Inputs

3.6.2.1 FUNC_32K_CLK Clock

The 32-kHz frequency is used for low-frequency operation. FUNC_32K_CLK is derived from SYS_CLK1. It supplies the always-on wake-up domain for operation in lowest power mode and as the clock source to the DPLL_ABE.

3.6.2.2 High-Frequency System Clock Input

The system clocks SYS_CLK1 and SYS_CLK2, are the main source clocks of the device. SYS_CLK1 and SYS_CLK2 are received directly from internal oscillators (OSC0 and OSC1) of the PRCM module. They are supplied as the reference clock to the DPLLs and as functional clock to several modules.

3.6.2.3 External Reference Clock Input

The external reference clocks xref_clk0, xref_clk1, xref_clk2, and xref_clk3 are received directly from external reference clock source for the device. They are supplied as the functional clock to the TIMERs, and as reference clock to the McASP and other peripherals.

3.6.3 Internal Clock Sources and Generators

The PRCM module clock sources/generators are split into the following parts:

- PRM clock source that receives system clocks SYS_CLK1 and SYS_CLK2 inputs.
- CM_CORE_AON and CM_CORE clock sources that distribute high-frequency clocks
- DPLL clock generators that synthesize high-frequency clocks for the device

Table 3-36. Internal Clock Sources

Clock Name	Source	Frequency	Note
FUNC_32K_CLK	SYS_CLK1/610	32 KHz	See Section 3.6.3.1, *PRM Clock Source*
SYS_CLK1	xi_osc0	19.2 MHz; 20 MHz; 27 MHz	See Section 3.6.3.1, *PRM Clock Source*
SYS_CLK2	xi_osc1	(19.2 - 32) MHz	See Section 3.6.3.1, *PRM Clock Source*
OSC_32K_CLK(1)	On-die 32K RC Osc	32 KHz	See Section 3.6.3.2.2, *CM_CORE_AON_CLKOUTMUX Overview*
FUNC_96M_AON_CLK	DPLL_PER	96 MHz	See Section 3.6.3.4, *DPLL_PER Description*
FUNC_192M_CLK	DPLL_PER	192 MHz	See Section 3.6.3.4, *DPLL_PER Description*
FUNC_128M_CLK	DPLL_PER	128 MHz	See Section 3.6.3.4, *DPLL_PER Description*
CORE_CLK	DPLL_CORE	532 MHz	See Figure 3-10, *DPLL_CORE Description*

(1) The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

3.6.3.1 PRM Clock Source

Note

The audio back end (ABE) module is not supported for this family of devices, but the ABE name is still present in some clock or DPLL names.

The PRM clock source receives the SYS_CLK1 and SYS_CLK2 clocks from external input pins. Along with these clocks, it receives a clock ABE_LP_CLK, divided version of DPLL_ABE_X2_CLK, which is generated by DPLL_ABE. The PRM manages the low-frequency clocks associated with these four (counting also FUNC_32K_CLK as an input clock) input clocks. The PRM sources various versions (through gating controls) of these externally sourced clocks to supply:

- PRCM-managed DPLLs with a reference clock, which is permanently supplied with always-on buffers
- The DSS with a reference clock, which is permanently supplied with always-on buffers

- The various timers and watchdog timers with clocks supplied by always-on buffers
- The clocks for the CM clock generator and the CORE power domain
- Timer functional clocks
- The bandgap and control module for thermal sensor feature
- Reference clock for various modules:
  – USB_OTG_SS
  – SATA

Figure 3-39 is a logical representation of the PRM clock source.

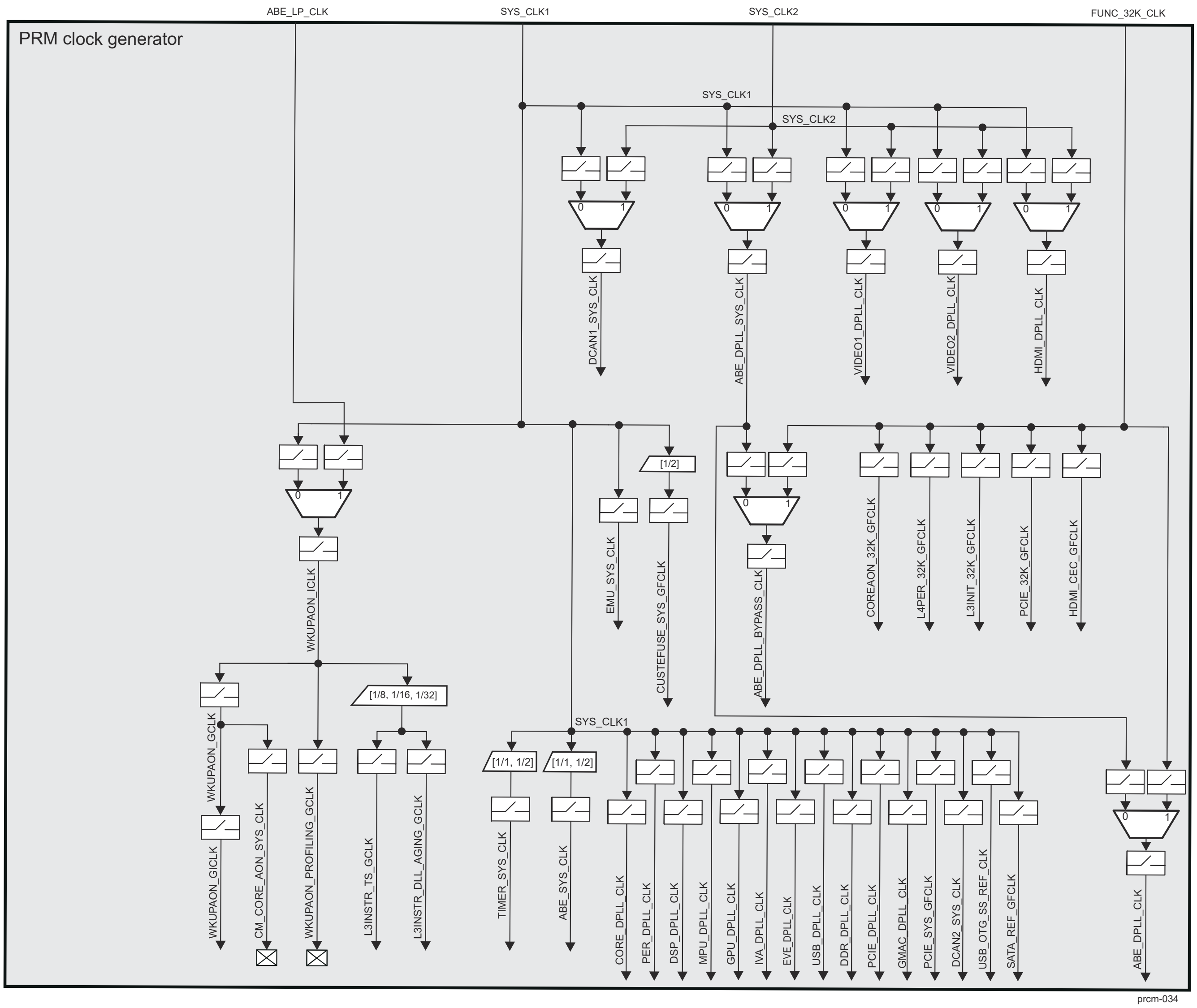


Figure 3-39. PRM Clock Manager Overview

Table 3-37 identifies controls for clock dividers or muxes in the PRM.

Table 3-37. PRM Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux ABE_DPLL_CLK	CM_CLKSEL_ABE_PLL_REF[0] CLKSEL
Mux WKUPAON_ICLK	CM_CLKSEL_WKUPAON[0] CLKSEL
Mux ABE_DPLL_BYPASS_CLK	CM_CLKSEL_ABE_PLL_BYPAS[0] CLKSEL
Mux DCAN1_SYS_CLK	CM_WKUPAON_DCAN1_CLKCTRL[24] CLKSEL

Table 3-37. PRM Clock Division and Muxing Control (continued)

Divider/Mux	Control Bit Field
Mux ABE_DPLL_SYS_CLK	CM_CLKSEL_ABE_PLL_SYS[0] CLKSEL
Mux VIDEO1_DPLL_CLK	CM_CLKSEL_VIDEO1_PLL_SYS[0] CLKSEL
Mux VIDEO2_DPLL_CLK	CM_CLKSEL_VIDEO2_PLL_SYS[0] CLKSEL
Mux HDMI_DPLL_CLK	CM_CLKSEL_HDMI_PLL_SYS[0] CLKSEL
Divider ABE_SYS_CLK	CM_CLKSEL_ABE_SYS[0] CLKSEL
Divider TIMER_SYS_CLK	CM_CLKSEL_TIMER_SYS[0] CLKSEL
Divider MPU_DPLL_CLK _ABE	CM_MPU_MPU_CLKCTRL[26] CLKSEL_ABE_DIV_MODE
Divider MPU_DPLL_CLK_EMIF	CM_MPU_MPU_CLKCTRL[25:24] CLKSEL_EMIF_DIV_MODE
Divider L3INSTR_TS_GCLK and L3INSTR_DLL_AGING_GCLK	CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL[25:24] CLKSEL

Note

For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

The PRM provides a 32-kHz gated and ungated clock for use by portions of the PD_WKUPAON power domain and some peripherals outside the PD_WKUPAON power domain.

The PRM creates COREAON_32K_GFCLK (gated version of FUNC_32K_CLK) to provide 32kHz clock to peripherals outside of PD_WKUPAON.

It also provides the system clock to the DSS, a gated and buffered version to the WKUPAON_GICLK interconnect clock, and the DPLLs controlled by the PRCM module.

3.6.3.2 CM Clock Source

The clock manager (CM) is primarily responsible for generating interface and functional clocks from the internal clocks provided by DPLL_CORE and DPLL_PER. The CM is physically divided into two independent entities: CM_CORE_AON, which is placed in the PD_COREAON always-on power domain, and CM_CORE, which is placed in the PD_CORE switchable power domain. The split is done to provide control over various entities, such as modules, DPLLs, and clocks, during low-power use case scenarios when the PD_CORE power domain can be switched to RETENTION state.

3.6.3.2.1 CM_CORE_AON Clock Generator

CM_CORE_AON receives a system clocks (SYS_CLK1 and SYS_CLK2) from the PRM, which serves as its functional clock. CM_CORE_AON provides a gated clock to

- PD_CAM
- PD_CORE
- PD_CUSTEFUSE
- PD_DSP1
- PD_DSP2
- PD_DSS
- PD_EMU
- PD_EVE1
- PD_EVE2
- PD_GPU
- PD_IPU
- PD_IVA
- PD_L3INIT
- PD_L4PER
- PD_MPU
- PD_RTC

- PD_VPE
- PD_WKUPAON
- PD_COREAON
- PD_MMAON
- PD_MPUAON

and some DPLLs:

- DPLL_MPU
- DPLL_IVA
- DPLL_EVE
- DPLL_GPU
- DPLL_DDR
- DPLL_GMAC
- DPLL_DSP
- DPLL_PER
- DPLL_ABE
- DPLL_CORE

and its associated HSDivider.

Figure 3-40 shows the various functional and interface clocks generated by CM_CORE_AON.

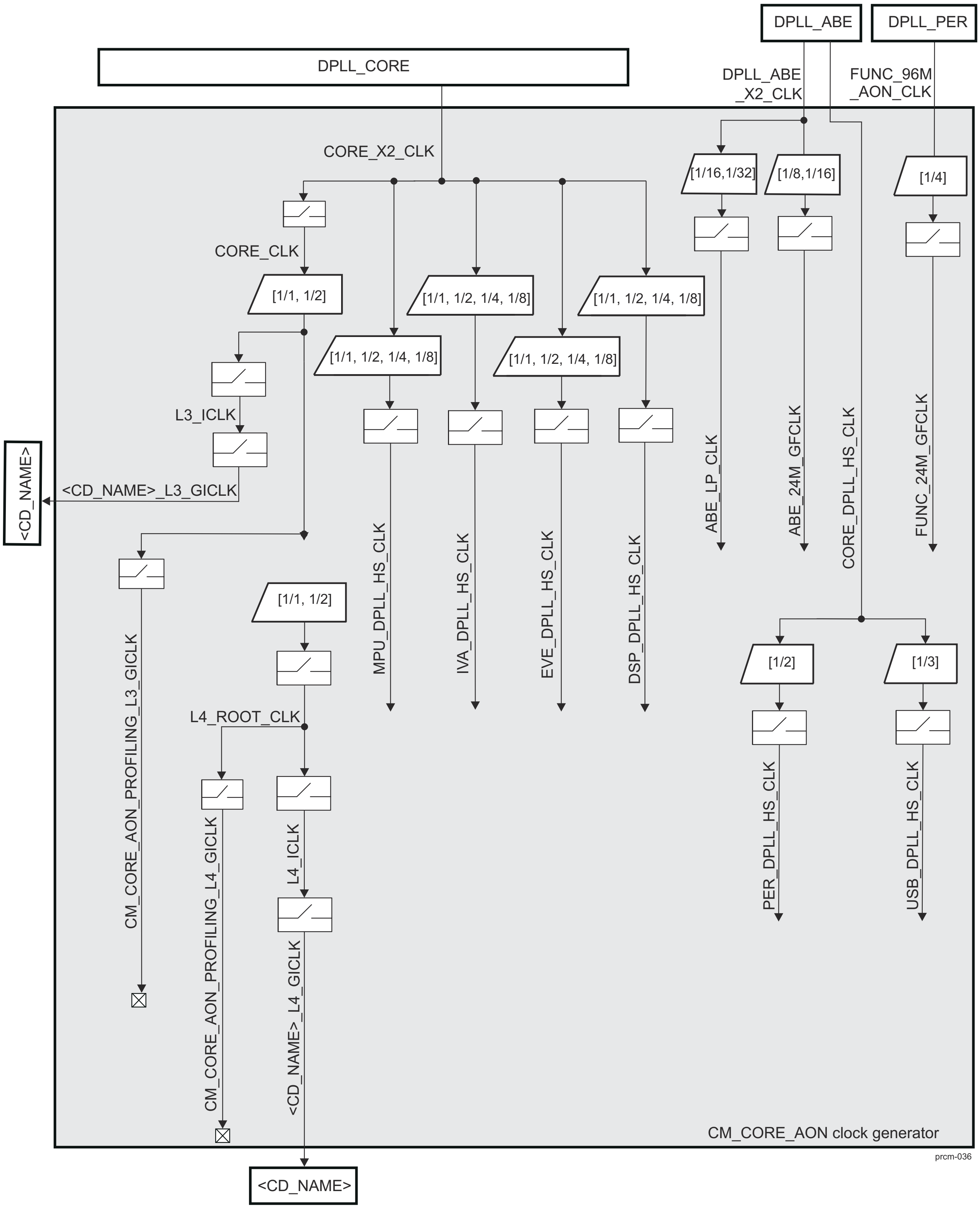


Figure 3-40. CM_CORE_AON Overview (a)

Table 3-38 identifies controls for clock dividers or muxes in the CM_CORE_AON (a) clock source.

Table 3-38. CM_CORE_AON (a) Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Divider L3_ICLK	CM_CLKSEL_CORE[4] CLKSEL_L3
Divider L4_ROOT_CLK	CM_CLKSEL_CORE[8] CLKSEL_L4
Divider MPU_DPLL_HS_CLK	CM_BYPCLK_DPLL_MPU[1:0] CLKSEL
Divider IVA_DPLL_HS_CLK	CM_BYPCLK_DPLL_IVA[1:0] CLKSEL
Divider EVE_DPLL_HS_CLK	CM_BYPCLK_DPLL_EVE[1:0] CLKSEL
Divider DSP_DPLL_HS_CLK	CM_BYPCLK_DPLL_DSP[1:0] CLKSEL
Divider ABE_LP_CLK	CM_CLKSEL_ABE_LP_CLK[1:0] CLKSEL
Divider ABE_24M_FCLK	CM_CLKSEL_ABE_24M[1:0] CLKSEL

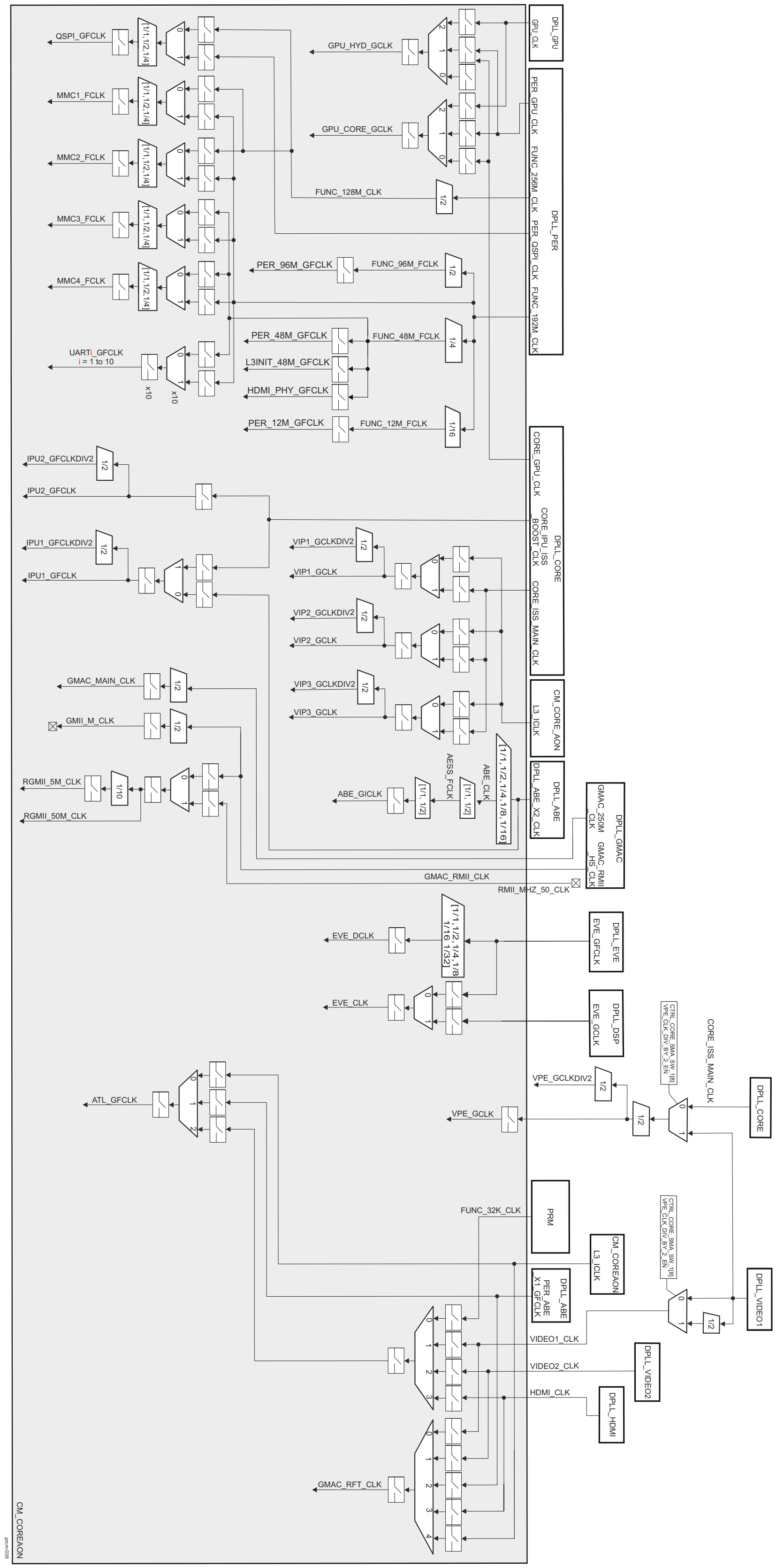


Figure 3-41. CM_CORE_AON Overview (b)

Note

VIDEO1_CLK, VIDEO2_CLK, and HDMI_CLK clocks and associated DPLL HSDIVIDERS are controlled by dedicated DPLL controllers (DPLL_VIDEO1, DPLL_VIDEO2 and DPLL_HDMI) in Display Subsystem, outside PRCM module. For more information, see Section 11.1.2.1, *Display Subsystem Clocks*, and Section 11.3.1, *HDMI Overview*.

Table 3-39 identifies controls for clock dividers or muxes in the CM_CORE_AON (b) clock source.

Table 3-39. CM_CORE_AON (b) Clock Division and Muxing Control

Divider/Mux/Switch	Control Bit Field
Mux MMC1_GFCLK	CM_L3INIT_MMC1_CLKCTRL[24] CLKSEL_SOURCE
Divider MMC1_GFCLK	CM_L3INIT_MMC1_CLKCTRL[26:25] CLKSEL_DIV
Mux MMC2_GFCLK	CM_L3INIT_MMC2_CLKCTRL[24] CLKSEL_SOURCE
Divider MMC2_GFCLK	CM_L3INIT_MMC2_CLKCTRL[26:25] CLKSEL_DIV
Mux MMC3_FCLK	CM_L4PER_MMC3_CLKCTRL[24] CLKSEL_MUX
Divider MMC3_FCLK	CM_L4PER_MMC3_CLKCTRL[26:25] CLKSEL_DIV
Mux MMC4_FCLK	CM_L4PER_MMC4_CLKCTRL[24] CLKSEL_MUX
Divider MMC4_FCLK	CM_L4PER_MMC4_CLKCTRL[26:25] CLKSEL_DIV
Mux UART1_GFCLK	CM_L4PER_UART1_CLKCTRL[24] CLKSEL
Mux UART2_GFCLK	CM_L4PER_UART2_CLKCTRL[24] CLKSEL
Mux UART3_GFCLK	CM_L4PER_UART3_CLKCTRL[24] CLKSEL
Mux UART4_GFCLK	CM_L4PER_UART4_CLKCTRL[24] CLKSEL
Mux UART5_GFCLK	CM_L4PER_UART5_CLKCTRL[24] CLKSEL
Mux UART6_GFCLK	CM_IPU_UART6_CLKCTRL[24] CLKSEL
Mux UART7_GFCLK	CM_L4PER2_UART7_CLKCTRL[24] CLKSEL
Mux UART8_GFCLK	CM_L4PER2_UART8_CLKCTRL[24] CLKSEL
Mux UART9_GFCLK	CM_L4PER2_UART9_CLKCTRL[24] CLKSEL
Mux UART10_GFCLK	CM_WKUPAON_UART10_CLKCTRL[24] CLKSEL
Mux GPU_HYD_GCLK	CM_GPU_GPU_CLKCTRL[27:26] CLKSEL_HYD_CLK
Mux GPU_CORE_GCLK	CM_GPU_GPU_CLKCTRL[25:24] CLKSEL_CORE_CLK
Mux QSPI_GFCLK	CM_L4PER2_QSPI_CLKCTRL[24] CLKSEL_SOURCE
Divider QSPI_GFCLK	CM_L4PER2_QSPI_CLKCTRL[26:25] CLKSEL_DIV
Mux VIP1_GCLK	CM_CAM_VIP1_CLKCTRL[24] CLKSEL
Mux VIP2_GCLK	CM_CAM_VIP2_CLKCTRL[24] CLKSEL
Mux VIP3_GCLK	CM_CAM_VIP3_CLKCTRL[24] CLKSEL
Divider ABE_CLK	CM_CLKSEL_ABE_CLK_DIV[2:0] CLKSEL
Divider AESS_FCLK	CM_CLKSEL_AESS_FCLK_DIV[0] CLKSEL
Divider ABE_GICLK	CM_CLKSEL_ABE_GICLK_DIV[0] CLKSEL
Mux IPU1_GFCLK	CM_IPU1_IPU1_CLKCTRL[24] CLKSEL
Mux RGMII_50M_CLK	CM_GMAC_GMAC_CLKCTRL[24] CLKSEL_REF
Mux ATL_GFCLK	CM_ATL_ATL_CLKCTRL[27:26] CLKSEL_SOURCE2
Mux ATL_SOURCE1	CM_ATL_ATL_CLKCTRL[25:24] CLKSEL_SOURCE1
Mux GMAC_RFT_CLK	CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT
Mux EVE_CLK	CM_CLKSEL_EVE_CLK[0] CLKSEL
Divider EVE_DCLK	CM_CLKSEL_EVE_GFCLK_CLKOUTMUX[2:0] CLKSEL

Note

For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

3.6.3.2.2 CM_CORE_AON_CLKOUTMUX Overview

Figure 3-42 is an overview of the CM_CORE_AON_CLKOUTMUX

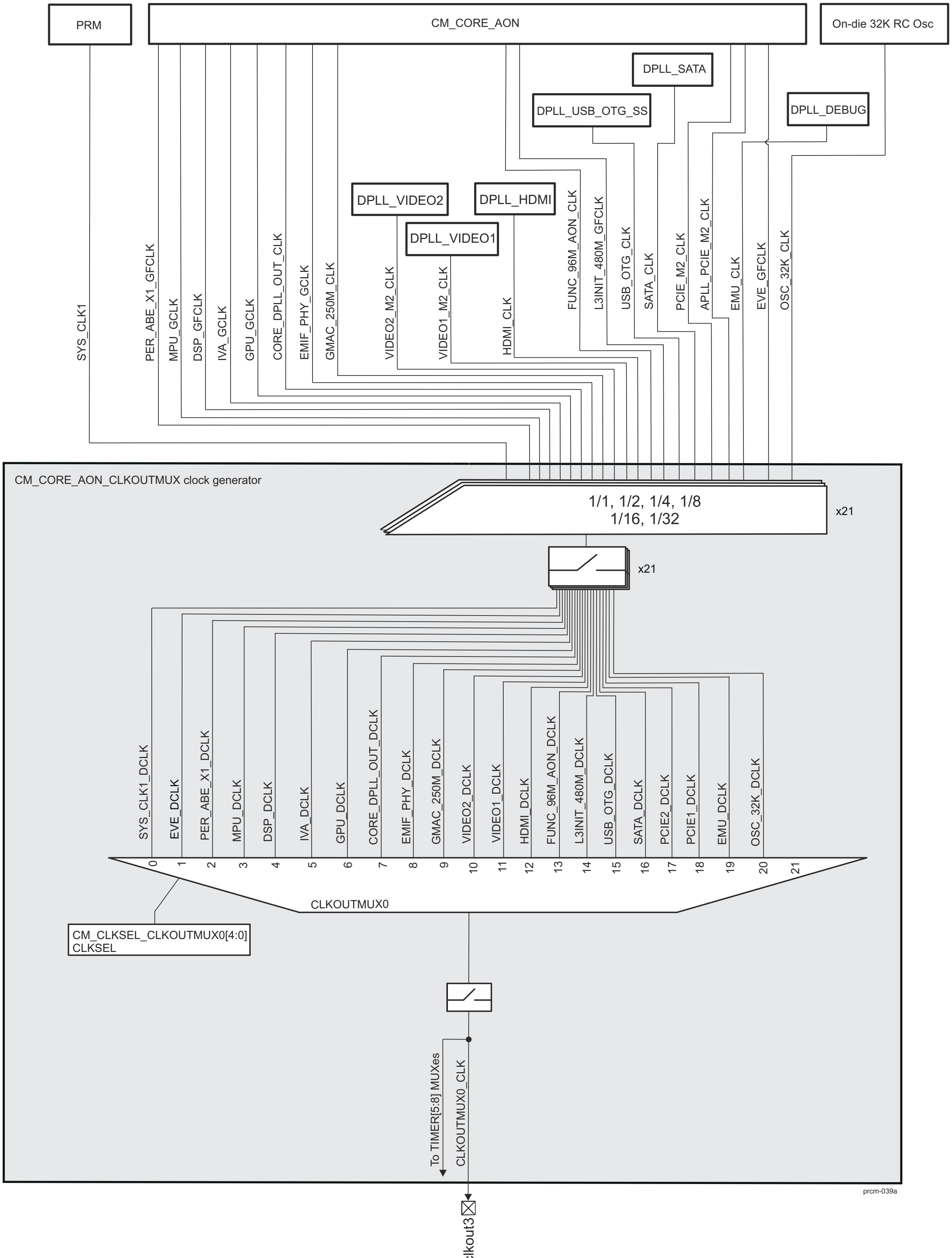


Figure 3-42. CM_CORE_AON_CLKOUTMUX Clock Manager Overview (CLKOUTMUX0)

Note

There is no dedicated register to enable CLKOUTMUX0_CLK clock to propagate to clkout3 device pad. CLKOUTMUX0_CLK will stay gated until one of the TIMER5/6/7/8 is enabled and CLKOUTMUX0 _CLK is selected as the source clock for the timer functional clock.

The CLKOUTMUX0_CLK clock is propagated to clkout3 device pad only when at least one of the following registers is configured: CM_IPU_TIMER5/6/7/8_CLKCTRL[27:24] CLKSEL (value: 0xB).

In addition, if there is a need to propagate MPU_GCLK to device pad (clkout3), the user needs to write 0x1 to the hardware observability control register: CTRL_CORE_HWOBS_CONTROL[0] HWOBS_MACRO_ENABLE.

Note

The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

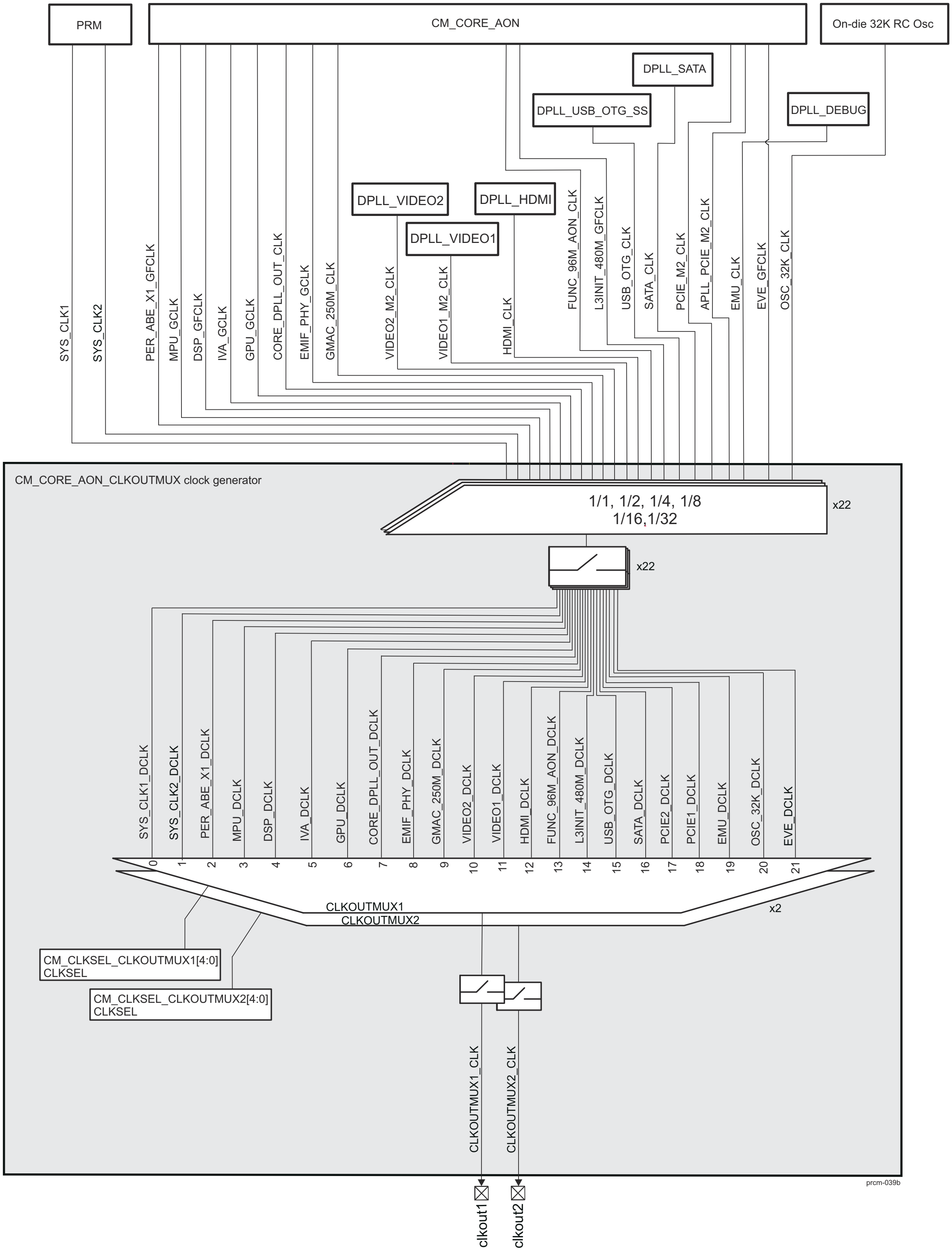


Figure 3-43. CM_CORE_AON_CLKOUTMUX Clock Manager Overview (CLKOUTMUX1 and CLKOUTMUX2)

Note

In order to enable CLKOUTMUX1_CLK and CLKOUTMUX2_CLK clocks to propagate to the corresponding device pads (clkout1 and clkout2), the user needs to configure registers: CM_COREAON_CLKOUTMUX1_CLKCTRL[8] OPTFCLKEN_CLKOUTMUX1_CLK (value: 0x1) and CM_COREAON_CLKOUTMUX2_CLKCTRL[8] OPTFCLKEN_CLKOUTMUX2_CLK (value: 0x1).

In addition, if there is a need to propagate MPU_GCLK to device pad (clkout1 or clkout2), the user needs to write 0x1 to the hardware observability control register: CTRL_CORE_HWOBS_CONTROL[0] HWOBS_MACRO_ENABLE.

Note

The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

Table 3-40 identifies controls for clock dividers or muxes in the CM_CORE_AON_CLKOUTMUX.

Table 3-40. CM_CORE_AON_CLKOUTMUX Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux CLKOUTMUX0_CLK	CM_CLKSEL_CLKOUTMUX0[4:0] CLKSEL
Mux CLKOUTMUX2_CLK	CM_CLKSEL_CLKOUTMUX2[4:0] CLKSEL
Mux CLKOUTMUX1_CLK	CM_CLKSEL_CLKOUTMUX1[4:0] CLKSEL
Divider SYS_CLK1_DCLK	CM_CLKSEL_SYS_CLK1_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK2_DCLK	CM_CLKSEL_SYS_CLK2_CLKOUTMUX[2:0] CLKSEL
Divider PER_ABE_X1_DCLK	CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX[2:0] CLKSEL
Divider MPU_DCLK	CM_CLKSEL_MPU_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider DSP_DCLK	CM_CLKSEL_DSP_GFCLK_CLKOUTMUX[2:0] CLKSEL
Divider IVA_DCLK	CM_CLKSEL_IVA_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider GPU_DCLK	CM_CLKSEL_GPU_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider CORE_DPLL_OUT_DCLK	CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX[2:0] CLKSEL
Divider EMIF_PHY_DCLK	CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider GMAC_250M_DCLK	CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX[2:0] CLKSEL
Divier VIDEO2_DCLK	CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX[2:0] CLKSEL
Divider VIDEO1_DCLK	CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX[2:0] CLKSEL
Divider HDMI_DCLK	CM_CLKSEL_HDMI_CLK_CLKOUTMUX[2:0] CLKSEL
Divider FUNC_96M_AON_DCLK	CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX[2:0] CLKSEL
Divider L3INIT_480M_DCLK	CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX[2:0] CLKSEL
Divider USB_OTG_DCLK	CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SATA_DCLK	CM_CLKSEL_SATA_CLK_CLKOUTMUX[2:0] CLKSEL
Divider PCIE2_DCLK	CM_CLKSEL_PCIE2_CLK_CLKOUTMUX[2:0] CLKSEL
Divider PCIE1_DCLK	CM_CLKSEL_PCIE1_CLK_CLKOUTMUX[2:0] CLKSEL
Divider EMU_DCLK	CM_CLKSEL_EMU_CLK_CLKOUTMUX[2:0] CLKSEL
Divider OSC_32K_DCLK	CM_CLKSEL_OSC_32K_CLK_CLKOUTMUX[2:0] CLKSEL
Divider EVE_DCLK	CM_CLKSEL_EVE_GFCLK_CLKOUTMUX[2:0] CLKSEL

3.6.3.2.3 CM_CORE_AON_TIMER Overview

Figure 3-44 through Figure 3-47 are an overview of the CM_CORE_AON_TIMER related to the device timers.

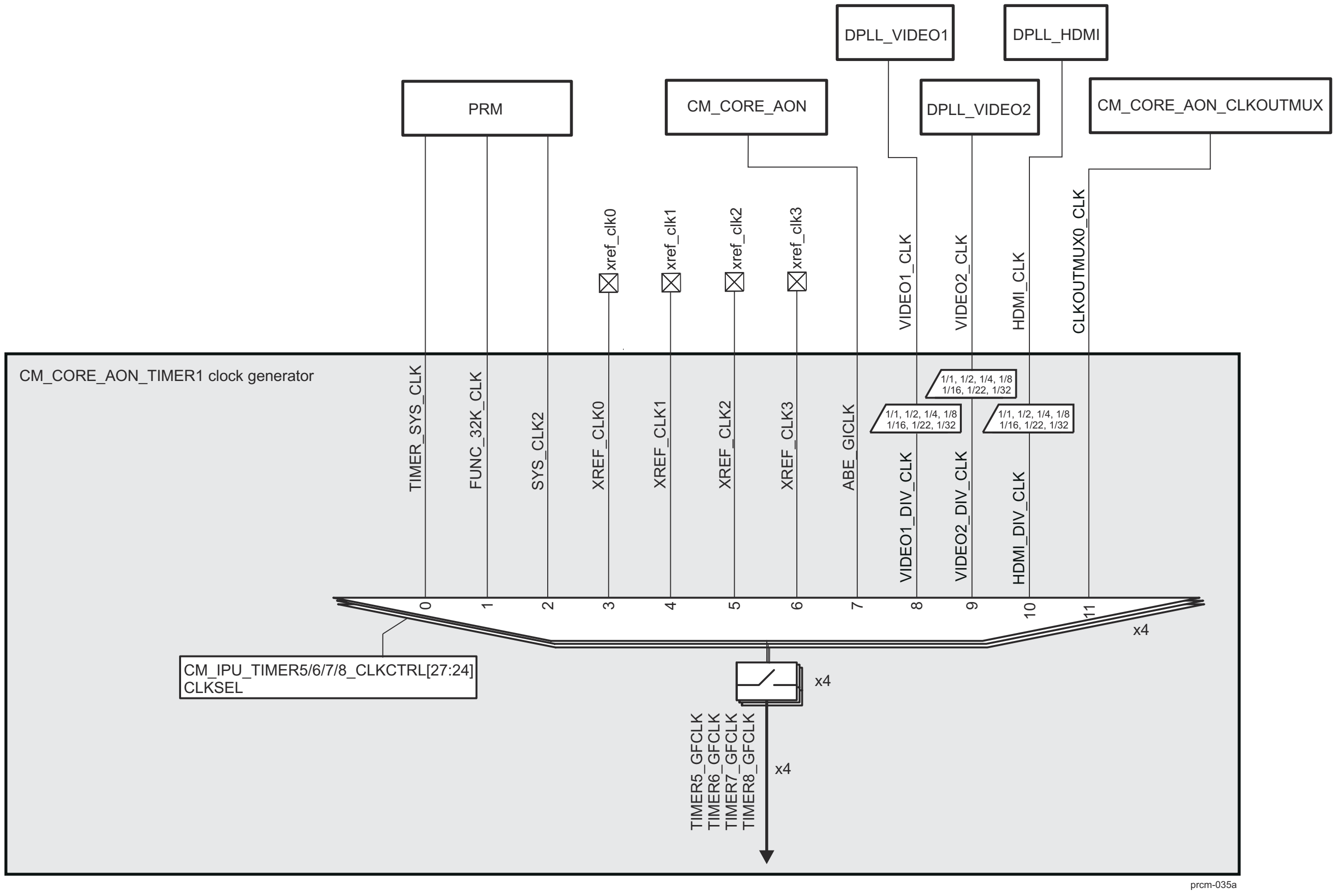


Figure 3-44. CM_CORE_AON_TIMER1 Clock Manager Overview

Note

VIDEO1_CLK, VIDEO2_CLK and HDMI_CLK clocks and associated DPLL HSDIVIDERS are controlled by dedicated DPLL controllers (DPLL_VIDEO1, DPLL_VIDEO2 and DPLL_HDMI) in Display Subsystem, outside PRCM module. For more information, see Section 11.1.2.1, *Display Subsystem Clocks* and Section 11.3.1, *HDMI Overview*.

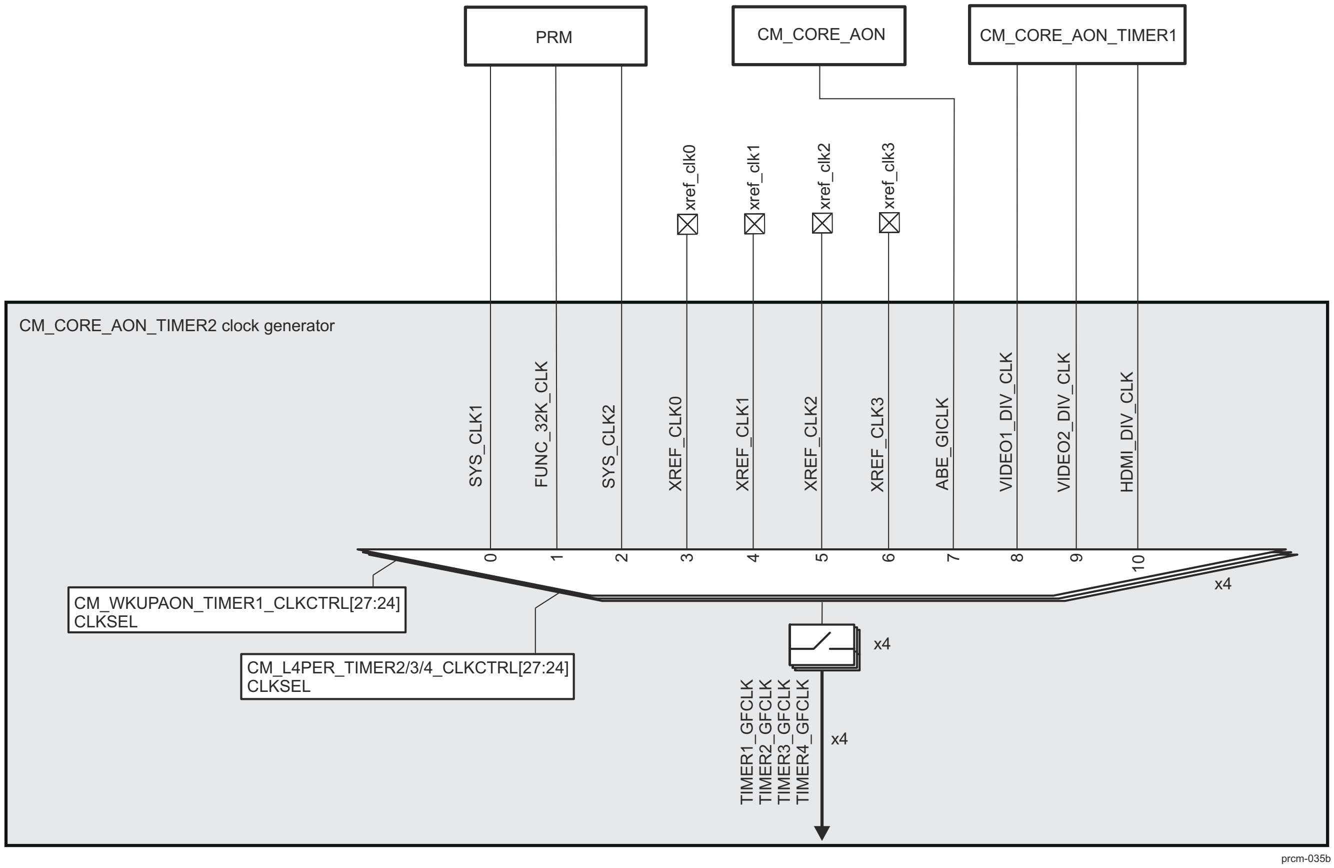


Figure 3-45. CM_CORE_AON_TIMER2 Clock Manager Overview

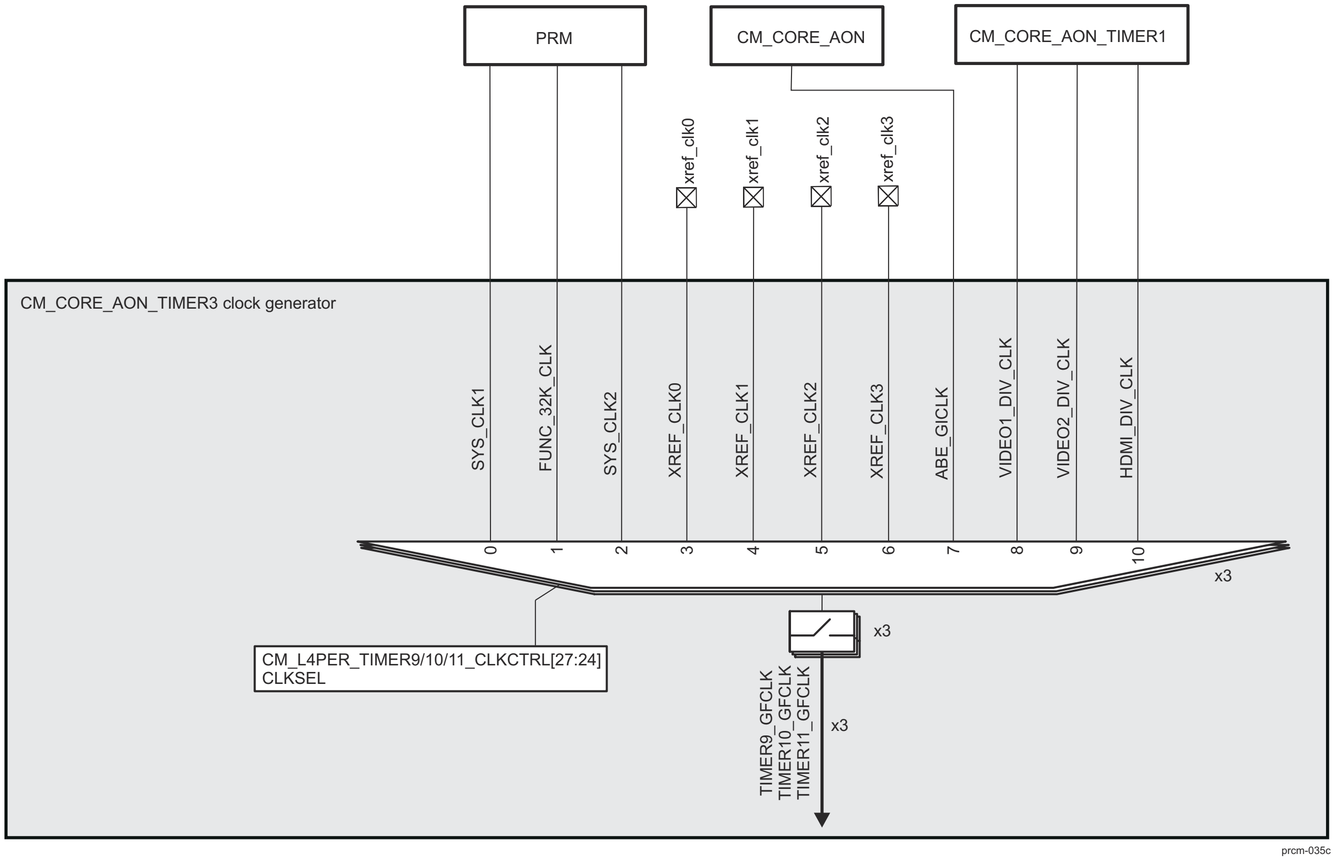


Figure 3-46. CM_CORE_AON_TIMER3 Clock Manager Overview

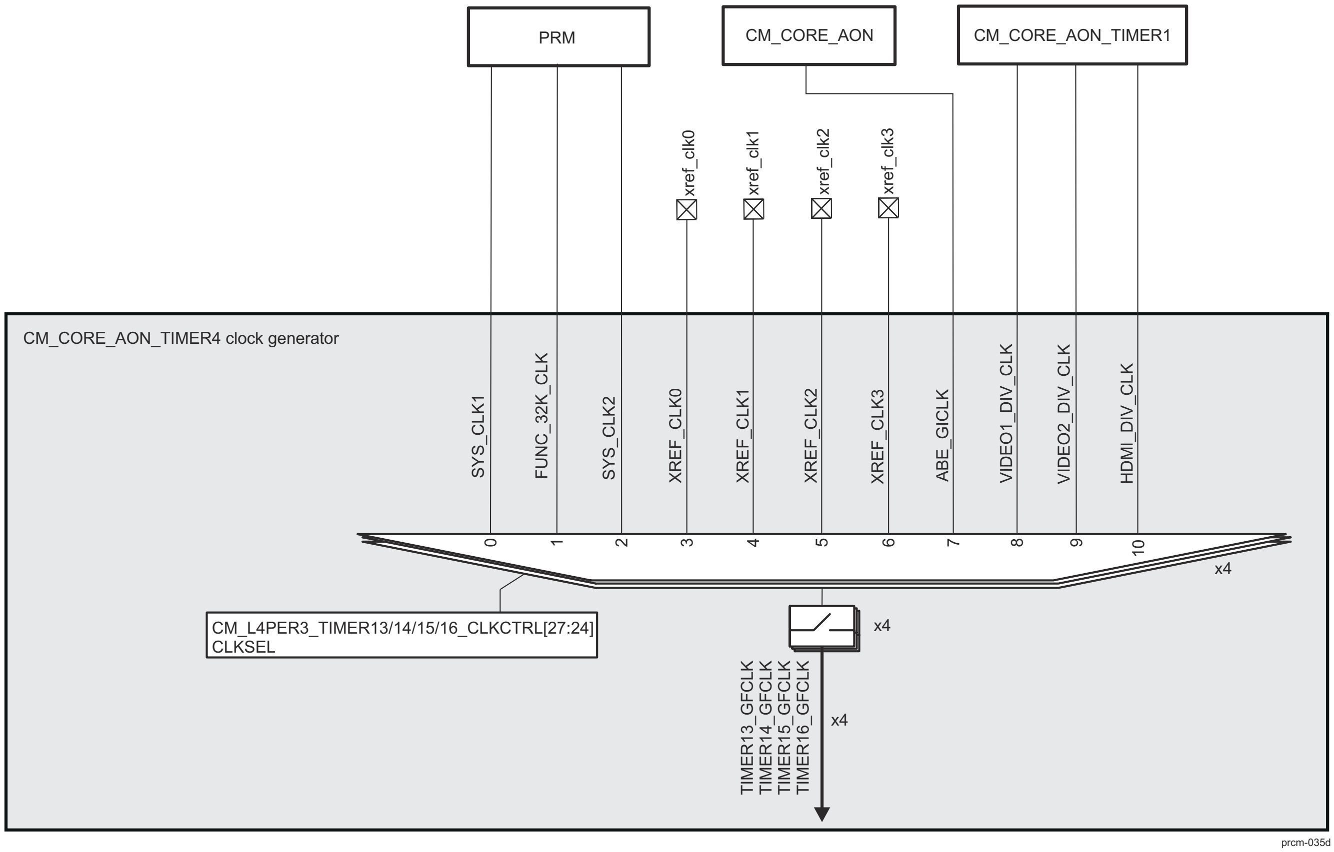


Figure 3-47. CM_CORE_AON_TIMER4 Clock Manager Overview

Table 3-41 identifies controls for clock dividers or muxes in the CM_CORE_AON_TIMER.

Table 3-41. CM_CORE_AON_TIMER Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux TIMER1_GFCLK	CM_WKUPAON_TIMER1_CLKCTRL[27:24] CLKSEL
Mux TIMER2_GFCLK	CM_L4PER_TIMER2_CLKCTRL[27:24] CLKSEL
Mux TIMER3_GFCLK	CM_L4PER_TIMER3_CLKCTRL[27:24] CLKSEL
Mux TIMER4_GFCLK	CM_L4PER_TIMER4_CLKCTRL[27:24] CLKSEL
Mux TIMER5_GFCLK	CM_IPU_TIMER5_CLKCTRL[27:24] CLKSEL
Mux TIMER6_GFCLK	CM_IPU_TIMER6_CLKCTRL[27:24] CLKSEL
Mux TIMER7_GFCLK	CM_IPU_TIMER7_CLKCTRL[27:24] CLKSEL
Mux TIMER8_GFCLK	CM_IPU_TIMER8_CLKCTRL[27:24] CLKSEL
Divider VIDEO1_CLK	CM_CLKSEL_VIDEO1_TIMER[2:0] CLKSEL
Divider VIDEO2_CLK	CM_CLKSEL_VIDEO2_TIMER[2:0] CLKSEL
Divider HDMI_CLK	CM_CLKSEL_HDMI_TIMER[2:0] CLKSEL
Mux TIMER9_GFCLK	CM_L4PER_TIMER9_CLKCTRL[27:24] CLKSEL
Mux TIMER10_GFCLK	CM_L4PER_TIMER10_CLKCTRL[27:24] CLKSEL
Mux TIMER11_GFCLK	CM_L4PER_TIMER11_CLKCTRL[27:24] CLKSEL
Mux TIMER13_GFCLK	CM_L4PER3_TIMER13_CLKCTRL[27:24] CLKSEL
Mux TIMER14_GFCLK	CM_L4PER3_TIMER14_CLKCTRL[27:24] CLKSEL
Mux TIMER15_GFCLK	CM_L4PER3_TIMER15_CLKCTRL[27:24] CLKSEL
Mux TIMER16_GFCLK	CM_L4PER3_TIMER16_CLKCTRL[27:24] CLKSEL

Note

For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

3.6.3.2.4 CM_CORE_AON_MCASP Overview

Figure 3-48 through Figure 3-50 are an overview of the CM_CORE_AON_MCASP related to the device McASP.

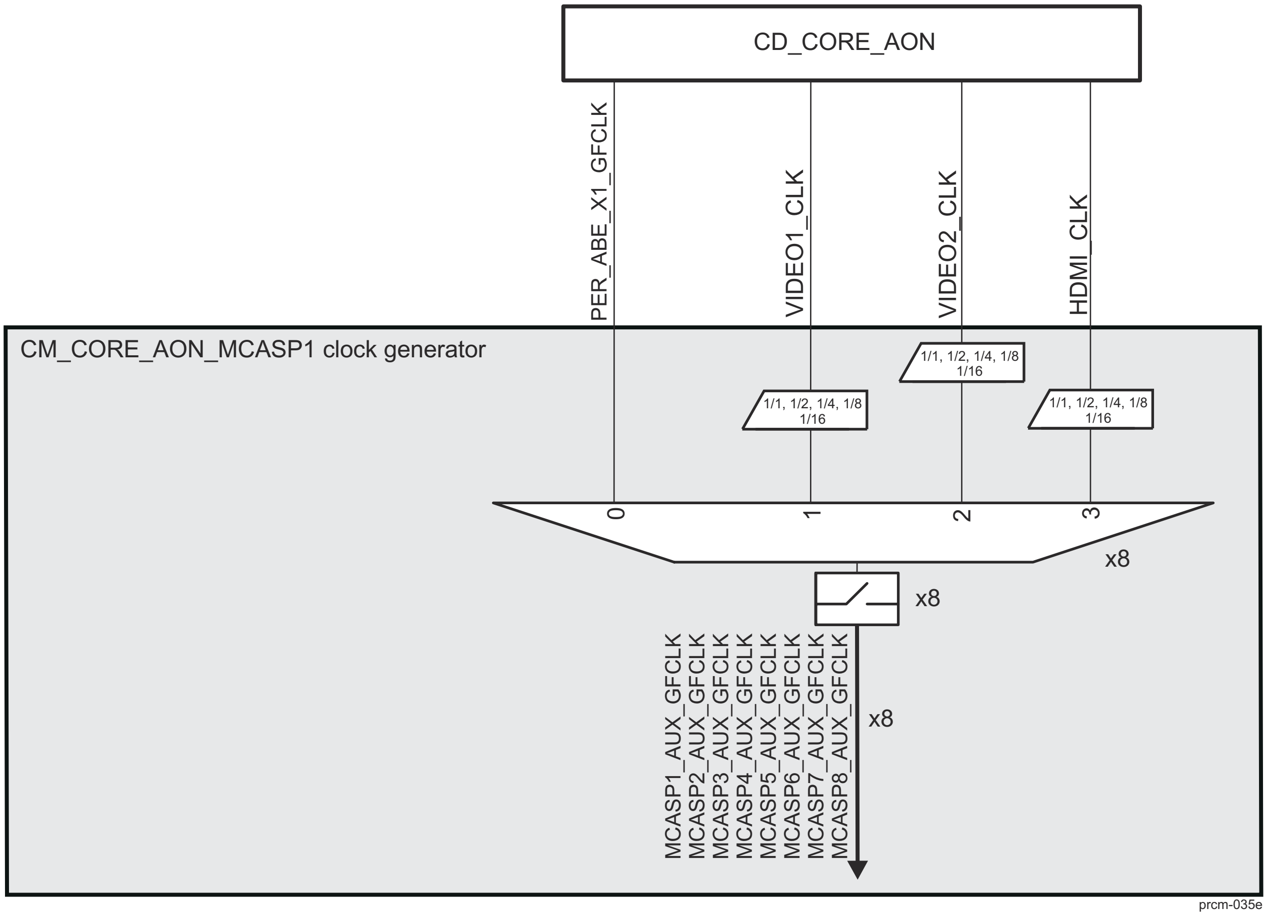


Figure 3-48. CM_CORE_AON_MCASP1 Clock Manager Overview

Note

VIDEO1_CLK, VIDEO2_CLK and HDMI_CLK clocks and associated DPLL HSDIVIDERS are controlled by dedicated DPLL controllers (DPLL_VIDEO1, DPLL_VIDEO2 and DPLL_HDMI) in Display Subsystem, outside PRCM module. For more information, see Section 11.1.2.1, *Display Subsystem Clocks* and Section 11.3.1, *HDMI Overview*.

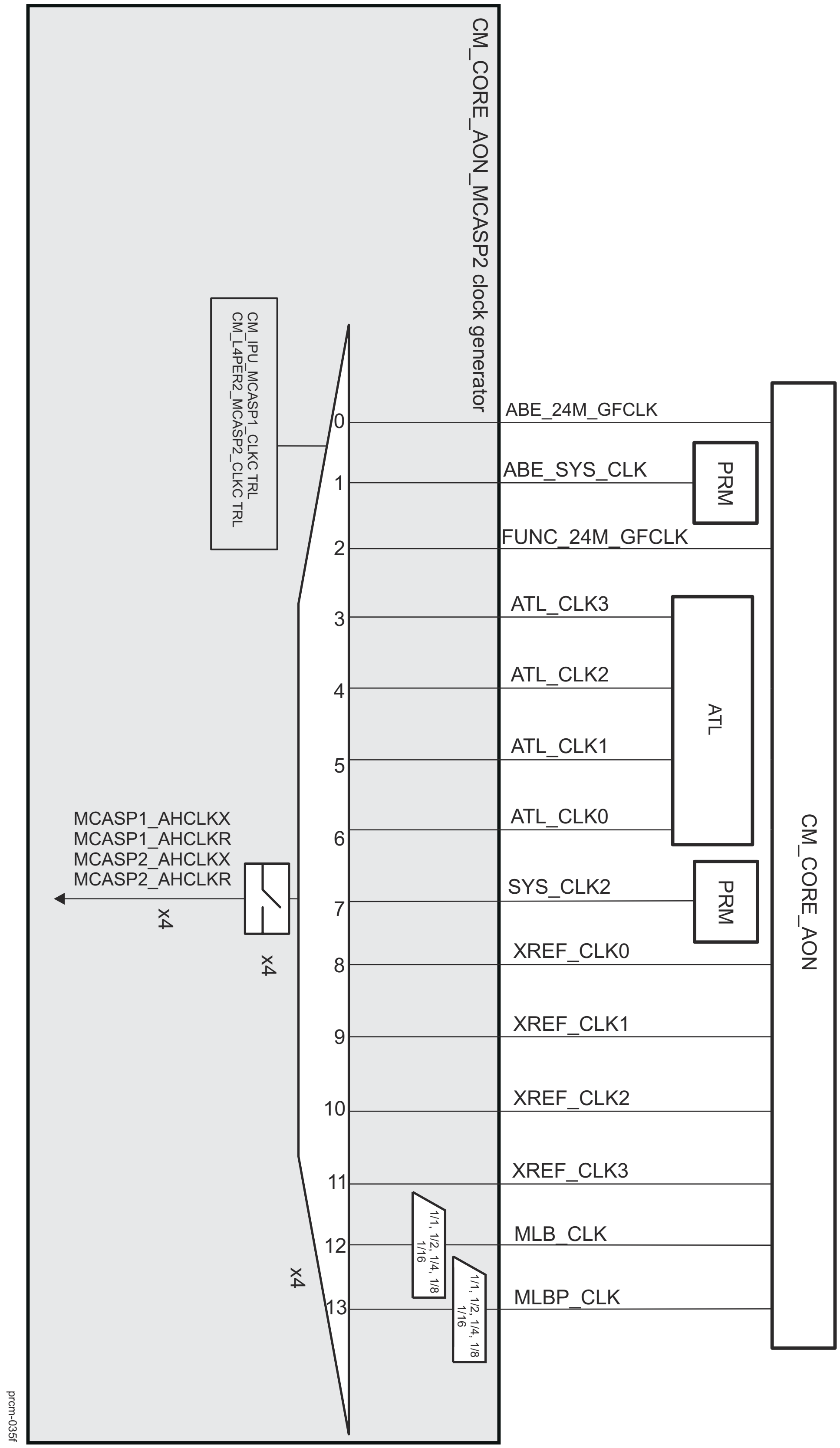


Figure 3-49. CM_CORE_AON_MCASP2 Clock Manager Overview

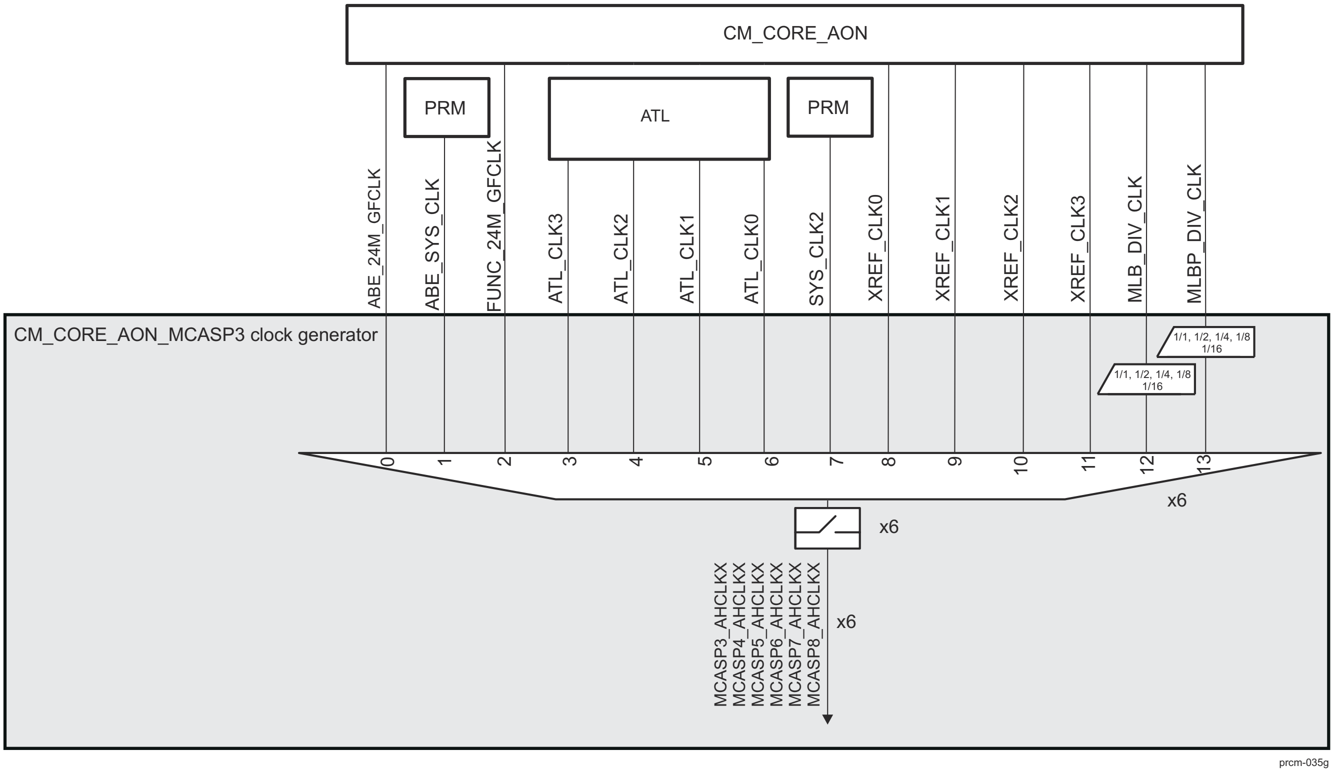


Figure 3-50. CM_CORE_AON_MCASP3 Clock Manager Overview

Table 3-42 identifies controls for clock dividers or muxes in the CM_CORE_AON_MCASP.

Table 3-42. CM_CORE_AON_MCASP Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux MCASP1_AUX_GFCLK	CM_IPU_MCASP1_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP2_AUX_GFCLK	CM_L4PER2_MCASP2_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP3_AUX_GFCLK	CM_L4PER2_MCASP3_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP4_AUX_GFCLK	CM_L4PER2_MCASP4_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP5_AUX_GFCLK	CM_L4PER2_MCASP5_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP6_AUX_GFCLK	CM_L4PER2_MCASP6_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP7_AUX_GFCLK	CM_L4PER2_MCASP7_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP8_AUX_GFCLK	CM_L4PER2_MCASP8_CLKCTRL[23:22] CLKSEL_AUX_CLK
Divider VIDEO1_CLK	CM_CLKSEL_VIDEO1_MCASP_AUX[2:0] CLKSEL
Divider VIDEO2_CLK	CM_CLKSEL_VIDEO2_MCASP_AUX[2:0] CLKSEL
Divider HDMI_CLK	CM_CLKSEL_HDMI_MCASP_AUX[2:0] CLKSEL
Divider PER_ABE_X1_GFCLK	CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX[2:0] CLKSEL
Mux MCASP1_AHCLKX	CM_IPU_MCASP1_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP1_AHCLKR	CM_IPU_MCASP1_CLKCTRL[31:28] CLKSEL_AHCLKR
Mux MCASP2_AHCLKX	CM_L4PER2_MCASP2_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP2_AHCLKR	CM_L4PER2_MCASP2_CLKCTRL[31:28] CLKSEL_AHCLKR
Divider MLB_CLK	CM_CLKSEL_MLB_MCASP[2:0] CLKSEL
Divider MLBP_CLK	CM_CLKSEL_MLBP_MCASP[2:0] CLKSEL
Mux MCASP3_AHCLKX	CM_L4PER2_MCASP3_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP4_AHCLKX	CM_L4PER2_MCASP4_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP5_AHCLKX	CM_L4PER2_MCASP5_CLKCTRL[27:24] CLKSEL_AHCLKX

Table 3-42. CM_CORE_AON_MCASP Clock Division and Muxing Control (continued)

Divider/Mux	Control Bit Field
Mux MCASP6_AHCLKX	CM_L4PER2_MCASP6_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP7_AHCLKX	CM_L4PER2_MCASP7_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP8_AHCLKX	CM_L4PER2_MCASP8_CLKCTRL[27:24] CLKSEL_AHCLKX

Note

For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

3.6.3.3 Generic DPLL Overview

To generate high-frequency clocks, the device supports multiple on-chip DPLLs controlled directly by the PRCM module. They are of two types: type A and type B DPLLs.

The following DPLLs belong to type A:

- DPLL_MPU
- DPLL_IVA
- DPLL_CORE
- DPLL_PER
- DPLL_ABE
- DPLL_DSP
- DPLL_EVE
- DPLL_GMAC
- DPLL_GPU
- DPLL_DDR

The following DPLLs belong to type B:

- DPLL_USB
- DPLL_PCIE_REF

All DPLLs (type A and type B) support the features described in the following sections, unless otherwise identified.

Note

This chapter discusses only the DPLLs that are directly controlled by the PRCM module. The other DPLLs embedded in and managed by other subsystems are described in their respective subsystems.

3.6.3.3.1 Generic APLL Overview

APLL_PCIE is an analog PLL which supplies high speed clock to PCI Express. APLL_PCIE is controlled directly by the PRCM module.

All the feature support by APLL_PCIE are described in Section 3.6.3.16, *APLL_PCIE Description*

3.6.3.3.2 DPLLs Output Clocks Parameters

Figure 3-51 shows the functional architecture of a generic DPLL.

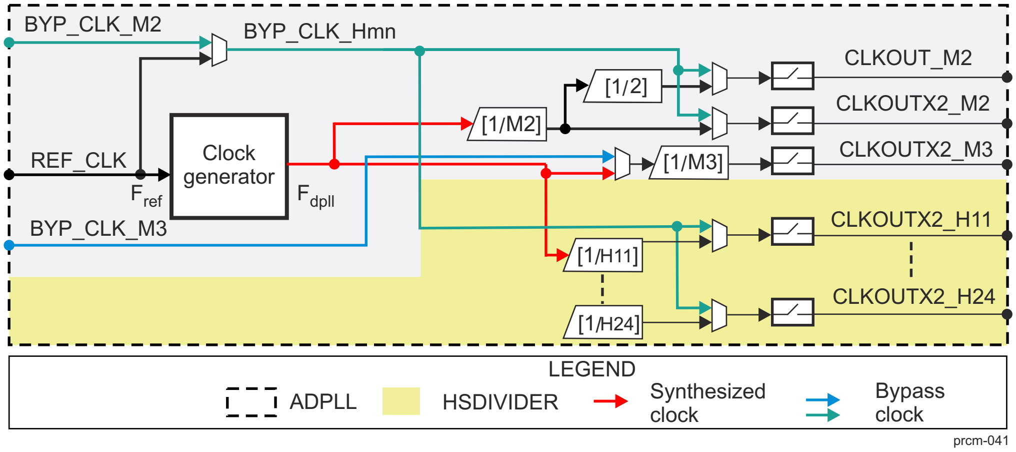


Figure 3-51. Generic DPLL Functional Diagram

The DPLL has three input clocks:

- REF_CLK: Used to generate the synthesized clock but can also be used as the bypass clock for some outputs of the DPLL whenever the DPLL enters bypass mode. It is mandatory for the DPLL clock synthesis.
- BYP_CLK_M2: Selectable bypass clock for the output of an M2 post-divider (optional)
- BYP_CLK_M3: Selectable bypass clock for the output of an M3 post-divider (optional)

The DPLL provides the bypass clock used by HSDIVIDER: BYP_CLK_Hmn, which is output by the multiplexer between the BYP_CLK_M2 and REF_CLK clock signals.

Note

The [1/2] divider located after the M2 post-divider is not valid when the CM_CLKSEL_DPLL_MPU[22] DCC_EN bit is set (applies only to DPLL_MPU when a frequency higher than 1.4 GHz is needed).

The DPLL can be programmed to be locked at any frequency given by one of the following equations:

- $F_{dpll} = F_{ref} \times 2 \times M / (N + 1)$
- $F_{dpll} = F_{ref} \times 2 \times (4 \times M / (N + 1))$ in case the CM_CLKMODE_DPLL_ABE[11] DPLL_REGM4XEN bit is set (applies only to DPLL_ABE)
- $F_{dpll} = F_{ref} \times (M / (N + 1))$ in case the CM_CLKSEL_DPLL_MPU[22] DCC_EN bit is set (applies only to DPLL_MPU when frequency higher than 1.4GHz is needed).

Where:

- F_{dpll} is the DPLL lock frequency.
- F_{ref} is the REF_CLK frequency. F_{ref} is also known as CLKINP.
- M is the software-configured multiplication ratio binary value.
- N is the software-configured division ratio binary value.

Note

It is preferred to minimize the value for N parameter (it minimizes lock time and jitter). Then M should be chosen to provide correct frequency (with lowest delta as possible).

It internally generates three main clocks: CLKOUT_M2, CLKOUTX2_M2, and CLKOUTX2_M3 as shown in Table 3-43.

Table 3-43. CLKOUT_M2, CLKOUTX2_M2, and CLKOUTX2_M3 Frequencies With DPLL State

Output	Equation	DPLL Mode
CLKOUT_M2	F_{dpll} / (2 x M2)	Locked (typical case)
	F_{ref} or BYP_CLK_M2	Before lock or during relock
	F_{dpll} / M2	DC corrector logic is used (DCC_EN bit is set) with M2 set to 0x1.
CLKOUTX2_M2	F_{dpll} / M2	Locked (typical case)
	F_{ref} or BYP_CLK_M2	Before lock or during relock
	F_{dpll} / M2	DC corrector logic is used (DCC_EN bit is set).
CLKOUTX2_M3	F_{dpll} / M3 or BYP_CLK_M3/M3	Locked (typical case)
	0	Before lock or during relock
	F_{dpll} / M3 or BYP_CLK_M3/M3	DC corrector logic is used (DCC_EN bit is set).

Where:

- M2 is the software-configured division ratio binary value.
- M3 is the software-configured division ratio binary value.
- CLKOUT_M2 and CLKOUTX2_M2 bypass clock input can be switched when the DPLL is not in locked state by using the M2 bypass clock select control bit.
- CLKOUTX2_M3 output clock can be switched when the DPLL is in locked state by using the M3 clock select control bit.

Note

- A value of 0 for M2 and M3 division ratios is not allowed. They are set to 1 after reset.
- CLKOUT_M2 is generated based on a fixed divide-by-2 ratio, except in bypass mode.

The DPLL can contain one or two HSDIVIDER modules to produce more clocks with divided ratio based on the DPLL synthesized clock frequency. HSDIVIDER1 provides four extra post-dividers from H11 to H14 (the output clocks are CLKOUTX2_H11 through CLKOUTX2_H14). HSDIVIDER2 provides four extra post-dividers from H21 through H24 (the output clocks are CLKOUTX2_H21 through CLKOUTX2_H24). The HSDIVIDER output clock frequency is given by the equations in Table 3-44.

Table 3-44. CLKOUTX2_Hmn Frequencies With DPLL State

Equation	DPLL Mode
CLKOUTX2_Hmn = F_{dpll} /Hmn	Locked
CLKOUTX2_Hmn = BYP_CLK_Hmn	Before lock or during relock

Where:

- F_{dpll} is the DPLL lock frequency.
- Hmn is the software-configured division ratio binary value.
- n is in the range from 1 to 4.
- m is equal to 1 or 2.

Note

Hmn division ratio is set to 1 after reset.

All clock outputs of the DPLL can be gated. The PRCM module provides the DPLL with a clock-gating control signal to enable or disable the clock. DPLL provides the PRCM module with a clock activity status signal to let the PRCM module hardware know when the clock is effectively running or effectively gated. The PRCM module provides a CM_IDLEST_DPLL_dpll_name[0] ST_DPLL_CLK status bit, which indicates the lock state or nonlock state of the DPLL.

The type B DPLL (DPLL_USB) has two outputs: CLKOUT and CLKDCOLDO. The DPLL can be programmed to be locked with the output clock frequencies given by the following equation:

- $F_{clkout} = F_{ref} \times (M / (N + 1)) \times (1 / M2)$
- $F_{clkdcoldo} = F_{ref} \times (M / (N + 1))$

Where:

- F_{clkout} is the frequency of the clock at output CLKOUT.
- $F_{clkdcoldo}$ is the frequency of the clock at output CLKDCOLDO.
- F_{ref} is the REF_CLK frequency.
- M is the software-configured multiplication ratio binary value.
- N is the software-configured division ratio binary value.
- M2 is the software-configured division ratio binary value.

CLKOUT supports the same bypass modes as previously identified. CLKDCOLDO does not support bypass mode.

The type B DPLL output clock is synthesized by an internal oscillator that is phase-locked to the REF_CLK. Two oscillators are built within a type B DPLL. The oscillators are user-selectable based on the synthesized output clock frequency requirement. If the required frequency is greater than or equal to 1500 MHz, the user must program a value of 1 in the CM_CLKSEL_DPLL_dpll_name[21] DPLL_SELFREQDCO bit. This drives a value of 100 on the SELFREQDCO input of the corresponding type B DPLL. If the required frequency is less than 1500 MHz, the user must program a value of 0 in the CM_CLKSEL_DPLL_dpll_name[21] DPLL_SELFREQDCO bit. This drives a value of 010 on the SELFREQDCO input of the corresponding type B DPLL.

3.6.3.3.3 Enable Control, Status, and Low-Power Operation Mode

The DPLL has a manual mode control bit field, which allows the setting of the different operating modes of the DPLL. When the DPLL is switched to lock mode, the current values of the multiplication ratio (M) and the division ratio (N) are latched in the DPLL. The DPLL then starts the lock or relock sequence to synthesize the corresponding output frequency clock.

The status of the synthesized clock output of the DPLL is represented by the CLKOUT status bit. It can be gated or active.

The type A DPLLs can be switched to low-power operation mode (also called LPMODE) to optimize DPLL power consumption when the input and output clock frequencies are low. This mode can be software-enabled using the low-power mode control bit of the DPLL.

It must be enabled only if both of the following operating conditions are satisfied:

- $F_{ref} / (N + 1)$ is less than or equal to 1 MHz.
- $F_{ref} \times M / (N + 1)$ is less than or equal to 100 MHz.

Where:

- F_{ref} is the REF_CLK frequency.
- M is the software-configured multiplication ratio binary value.
- N is the software-configured division ratio binary value.

3.6.3.3.4 DPLL Power Modes

DPLL supports several power modes for type A and only one power mode for type B. Each mode results in a tradeoff between power savings and relock time. The PRCM module allows only a few modes for each DPLL, depending on the use of the DPLL.

Table 3-45 lists the DPLL power modes.

Table 3-45. DPLL Power Modes

Power Mode	CLKOUT State	Logic Current (mA)	Analog Current (mA)	Freq Lock Time	Phase Lock Time
Low-power stop	Clock stopped	0.065 (leakage)	0.009 (leakage)	2.5 µs + (70 × (N+1) / F_{ref})	2.5 µs + (120 × (N+1) / F_{ref})
Fast-relock stop	Clock stopped	0.065 (leakage)	0.009 (leakage)	2.5 µs + (70 × (N+1) / F_{ref})	2.5 µs + (120 × (N+1) / F_{ref})
Low-power bypass	Bypass clock/ clock stopped	0.065 (leakage)	0.009 (leakage)	2.5 µs + (70 × (N+1) / F_{ref})	2.5 µs + (120 × (N+1) / F_{ref})
Fast-relock bypass	Bypass clock/ clock stopped	0.065 (leakage)	0.5 (leakage)	0.05 µs + (70 × (N+1) / F_{ref})	0.05 µs + (120 × (N+1) / F_{ref})
Lock	Synthesized clock	0.95 (active)	3 (active)	N/A	N/A

Where:

- F_{ref} is the REF_CLK frequency.

A DPLL power mode can be achieved on a software request (manual) and/or automatically (automatic), depending on the specific hardware conditions.

A DPLL can switch from one mode to another as a result of the following:

- Software-programmed transition (manual): Software configures the dedicated DPLL manual mode control feature for the next desired DPLL mode. It must ensure that the transition can be performed based on the clock activity on the device.
- Combined software-programmed and hardware-conditions-based transition (auto): This mode allows the DPLL to automatically transition to a low-power state (that is, any state other than the LOCK state) when the output clocks are gated or the destination clock domain is inactive, and to switch back to the LOCK state when the output clock is needed (that is, the clock is ungated or the clock domain becomes active). The desired low-power state for the automatic transition is configured in the dedicated Auto Mode Control parameter of the DPLL.

Note

With T_{ref} = 1 / F_{ref} = (N + 1) / CLKINP; (Fref = CLKINP / (N + 1)):

$T_{ref}$is the REF_CLK period.

This formula indicates that a smaller N divider value provides a smaller time for switching the clock after an M2 post-divider change.

A compromise is necessary between the clock switching latency and power consumption. Having a smaller N value:

– Requires a higher M2 post-divider value to obtain the same target frequency.
– Results in a higher DPLL lock frequency, and then higher power consumption.

Note

– A manual transition can be performed from any power mode to any other power mode.
– An automatic transition can be performed from lock mode to any low-power mode.

Note

When the DPLL is in Low-Power bypass mode, Auto-idle mode is disabled and no clock is requested, the DPLL makes a transition to Low-power stop mode.

3.6.3.3.5 DPLL Recalibration

The DPLL recalibration applies only to type-A DPLLs. Each time the DPLL is reset or performs a lock sequence (following a change in the value of multiplier M or divider N), it performs a recalibration of the output frequency, based on voltage and temperature conditions. In lock mode, the DPLL maintains a steady lock frequency output by compensating for voltage and temperature changes within a certain range. However, if the voltage or temperature drifts outside the range or shows a significant or fast change, the DPLL may not be able to track and compensate it. It would need a recalibration, which is signaled by assertion of a recalibration flag.

Note

- The recalibration mechanism is active only while the DPLL is in lock mode. When the DPLL is in off or bypass mode (low-power or fast-relock), it does not assert the recalibration flag.
- If the DPLL drifts out of the operating range limits while not locked, and then when it tries to relock, it fails to lock within the normal delay and recalibrates automatically before eventually locking. The only difference between this case and a standard relock is the recalibration delay.

During recalibration, the DPLL loses lock and output clock switches to the bypass clock.

The DPLL can automatically start recalibration when the recalibration flag is asserted, or recalibration can be triggered by software control. The trigger setting of the recalibration can be configured by the corresponding registers of the DPLL in the PRCM module. The software-controlled recalibration mode is selected by default.

Software-controlled recalibration: The DPLL continues its tracking mechanism as long as the recalibration is not triggered by software (that is, by enabling the recalibration-enable control parameter). If the DPLL reaches upper or lower bounds of the DCO control code and software has still not triggered recalibration, the DPLL stops its tracking mechanism. The output clock remains active, but frequency and jitter are not ensured to meet the requirement.

Automatic recalibration: The DPLL immediately starts the recalibration as soon as the recalibration flag is asserted.

Note

Automatic recalibration of the DPLL can start at any time. While relocking, the DPLL switches to bypass mode, which introduces a frequency change. For modules that are sensitive to frequency change while operating, this can introduce operational instability. For example, the external memory EMIF controller is sensitive to a frequency change on the DPLL because its embedded DLL relocks on a frequency change. Any EMIF access during this DLL relock period can be corrupted. It is, therefore, important to stall EMIF access during DPLL recalibration.

To allow software to recalibrate the DPLL at the correct time depending on the device activity, the PRCM module can generate a wake-up event on the processor power domain, followed by an interrupt on the processor subsystem when the DPLL recalibration flag is asserted.

Table 3-46 lists the DPLL recalibration and control parameters.

Table 3-46. DPLL Recalibration Control Parameters

Parameter	Register	Description
Recalibration-enable control	CM_CLKMODE_DPLL_<module>[8] DPLL_DRIFTGUARD_EN	Enable/disable the DPLL automatic recalibration feature.
Recalibration-interrupt mask control	PRM_IRQENABLE_MPU PRM_IRQENABLE_MPU_2 PRM_IRQENABLE_IPU1 PRM_IRQENABLE_IPU2 PRM_IRQENABLE_DSP1 PRM_IRQENABLE_DSP2	Mask/unmask the DPLL recalibration interrupt to processor.

Table 3-46. DPLL Recalibration Control Parameters (continued)

Parameter	Register	Description
Recalibration-interrupt status	PRM_IRQSTATUS_MPU PRM_IRQSTATUS_MPU_2 PRM_IRQSTATUS_IPU1 PRM_IRQSTATUS_IPU2 PRM_IRQSTATUS_DSP1 PRM_IRQSTATUS_DSP2	Status of the DPLL recalibration interrupt to processor

3.6.3.3.6 DPLL Output Power Down

The DCO clock LDO (DCOCLKLDO) of the DPLL can be powered down if all output dividers of the DPLL are powered down. The PRCM module automatically reenables the power to the LDO when an output divider is powered up or the DPLL switches to bypass mode.

Table 3-47. DPLL Power-Down Control Parameters

Parameter	Description
DCO clock LDO power down control	Enable/disable automatic power-down feature if all output dividers are powered down.

3.6.3.4 DPLL_PER Description

3.6.3.4.1 DPLL_PER Overview

Figure 3-52 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

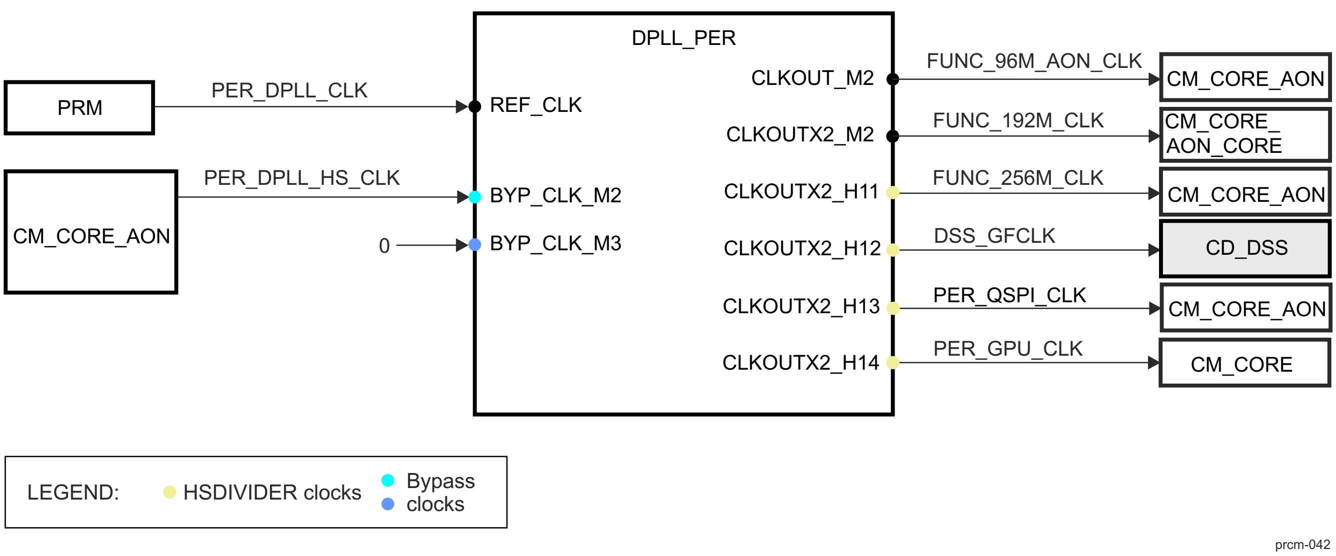


Figure 3-52. DPLL_PER Overview

3.6.3.4.2 DPLL_PER Synthesized Clock Parameters

This section describes the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*

Table 3-48 lists the clock synthesis parameters of the DPLL.

Table 3-48. DPLL_PER Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_PER[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_PER[6:0] DPLL_DIV

Table 3-49 lists the clock output divider parameters of the DPLL.

Table 3-49. DPLL_PER Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_PER[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_PER[4:0] DIVHS
CLKOUTX2_M2	Status	CM_DIV_M2_DPLL_PER[11] CLKX2ST
CLKOUTX2_H11	Status	CM_DIV_H11_DPLL_PER[9] CLKST
CLKOUTX2_H11	Divider control	CM_DIV_H11_DPLL_PER[5:0] DIVHS
CLKOUTX2_H12	Status	CM_DIV_H12_DPLL_PER[9] CLKST
CLKOUTX2_H12	Divider control	CM_DIV_H12_DPLL_PER[5:0] DIVHS
CLKOUTX2_H13	Status	CM_DIV_H13_DPLL_PER[9] CLKST
CLKOUTX2_H13	Divider control	CM_DIV_H13_DPLL_PER[5:0] DIVHS
CLKOUTX2_H14	Status	CM_DIV_H14_DPLL_PER[9] CLKST
CLKOUTX2_H14	Divider control	CM_DIV_H14_DPLL_PER[5:0] DIVHS

3.6.3.4.3 DPLL_PER Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and the associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode*, and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-50 lists the operating modes supported by the DPLL.

Table 3-50. DPLL_PER Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-51 lists the control bit fields for the operating mode control of the DPLL.

Table 3-51. DPLL_PER Mode Control Parameters

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_PER[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_PER[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_PER[2:0] AUTO_DPLL_MODE

3.6.3.4.4 DPLL_PER Recalibration

Table 3-52 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-52. DPLL_PER Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_PER[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[3] DPLL_PER_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[3] DPLL_PER_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[3] DPLL_PER_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[3] DPLL_PER_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[3] DPLL_PER_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[3] DPLL_PER_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[3] DPLL_PER_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[3] DPLL_PER_RECAL_EN

Table 3-52. DPLL_PER Recalibration Feature Parameters (continued)

Parameter Name	Control/Status Bit Field
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[3] DPLL_PER_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[3] DPLL_PER_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[3] DPLL_PER_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[3] DPLL_PER_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[3] DPLL_PER_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[3] DPLL_PER_RECAL_EN

3.6.3.5 DPLL_CORE Description

3.6.3.5.1 DPLL_CORE Overview

Figure 3-53 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

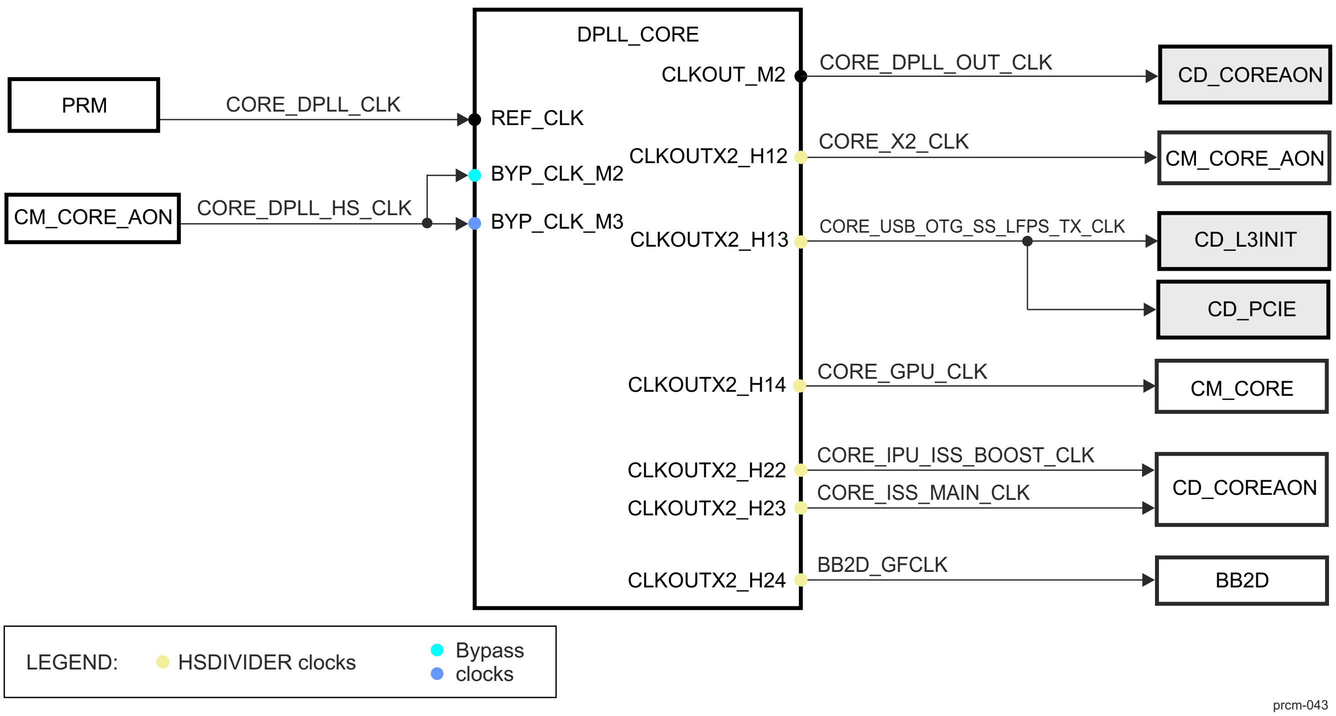


Figure 3-53. DPLL_CORE Overview

3.6.3.5.2 DPLL_CORE Synthesized Clock Parameters

This section describes the clock synthesis and clock-out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-53 lists the clock synthesis parameters of the DPLL.

Table 3-53. DPLL_CORE Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_CORE[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_CORE[6:0] DPLL_DIV
M (restore)	CM_CLKSEL_DPLL_CORE_RESTORE[18:8] DPLL_MULT
N (restore)	CM_CLKSEL_DPLL_CORE_RESTORE[6:0] DPLL_DIV

Table 3-54 lists the clock output divider parameters of the DPLL.

Table 3-54. DPLL_CORE Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_CORE[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_CORE[4:0] DIVHS
CLKOUTX2_H12	Status	CM_DIV_H12_DPLL_CORE[9] CLKST
CLKOUTX2_H12	Divider control	CM_DIV_H12_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H13	Status	CM_DIV_H13_DPLL_CORE[9] CLKST
CLKOUTX2_H13	Divider control	CM_DIV_H13_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H14	Status	CM_DIV_H14_DPLL_CORE[9] CLKST
CLKOUTX2_H14	Divider control	CM_DIV_H14_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H22	Status	CM_DIV_H22_DPLL_CORE[9] CLKST
CLKOUTX2_H22	Divider control	CM_DIV_H22_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H23	Status	CM_DIV_H23_DPLL_CORE[9] CLKST
CLKOUTX2_H23	Divider control	CM_DIV_H23_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H24	Status	CM_DIV_H24_DPLL_CORE[9]CLKST
CLKOUTX2_H24	Divider control	CM_DIV_H24_DPLL_CORE[5:0]DIVHS

3.6.3.5.3 DPLL_CORE Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-55 lists the operating modes supported by the DPLL.

Table 3-55. DPLL_CORE Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-56 lists the control bit fields for the operating mode control of the DPLL.

Table 3-56. DPLL_CORE Mode Control Parameters

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_CORE[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_CORE[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_CORE[2:0] AUTO_DPLL_MODE
Low-Power Mode Control (Restore)	CM_CLKMODE_DPLL_CORE_RESTORE[10] DPLL_LPMODE_EN
Manual Mode Control (Restore)	CM_CLKMODE_DPLL_CORE_RESTORE[2:0] DPLL_EN
Auto Mode Control (Restore)	CM_AUTOIDLE_DPLL_CORE_RESTORE[2:0] AUTO_DPLL_MODE

3.6.3.5.4 DPLL_CORE Recalibration

Table 3-57 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-57. DPLL_CORE Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_CORE[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[0] DPLL_CORE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[0] DPLL_CORE_RECAL_EN

Table 3-57. DPLL_CORE Recalibration Feature Parameters (continued)

Parameter Name	Control/Status Bit Field
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[0] DPLL_CORE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[0] DPLL_CORE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[0] DPLL_CORE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[0] DPLL_CORE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[0] DPLL_CORE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[0] DPLL_CORE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[0] DPLL_CORE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[0] DPLL_CORE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[0] DPLL_CORE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[0] DPLL_CORE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[0] DPLL_CORE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[0] DPLL_CORE_RECAL_EN

3.6.3.6 DPLL_ABE Description

3.6.3.6.1 DPLL_ABE Overview

Figure 3-54 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

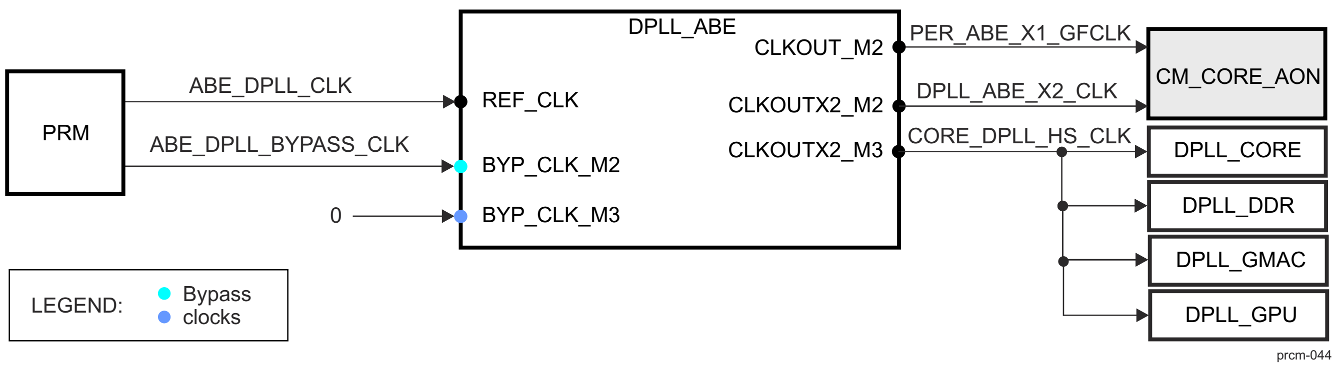


Figure 3-54. DPLL_ABE Overview

3.6.3.6.2 DPLL_ABE Synthesized Clock Parameters

This section describes the clock synthesis and clock-out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-58 lists the clock synthesis parameters of the DPLL.

Table 3-58. DPLL_ABE Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_ABE[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_ABE[6:0] DPLL_DIV
REGM4XEN	CM_CLKMODE_DPLL_ABE[11] DPLL_REGM4XEN

Table 3-59 lists the clock output divider parameters of the DPLL.

Table 3-59. DPLL_ABE Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_ABE[9] CLKST

Table 3-59. DPLL_ABE Clock Output Parameters (continued)

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Divider Control	CM_DIV_M2_DPLL_ABE[4:0] DIVHS
CLKOUTX2_M2	Status	CM_DIV_M2_DPLL_ABE[11] CLKX2ST
CLKOUTX2_M3	Status	CM_DIV_M3_DPLL_ABE[9] CLKST
CLKOUTX2_M3	Divider Control	CM_DIV_M3_DPLL_ABE [4:0] DIVHS

3.6.3.6.3 DPLL_ABE Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode*, and Section 3.6.3.3.4, *DPLL Power Mode*.

Table 3-60 lists the operating modes supported by the DPLL.

Table 3-60. DPLL_ABE Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-61 lists the control bit fields for the operating mode control of the DPLL.

Table 3-61. DPLL_ABE Mode Control Parameters

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_ABE[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_ABE[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_ABE[2:0] AUTO_DPLL_MODE

3.6.3.6.4 DPLL_ABE Recalibration

Table 3-62 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-62. DPLL_ABE Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_ABE[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[4] DPLL_ABE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[4] DPLL_ABE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[4] DPLL_ABE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[4] DPLL_ABE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[4] DPLL_ABE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[4] DPLL_ABE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[4] DPLL_ABE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[4] DPLL_ABE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[4] DPLL_ABE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[4] DPLL_ABE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[4] DPLL_ABE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[4] DPLL_ABE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[4] DPLL_ABE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[4] DPLL_ABE_RECAL_EN

3.6.3.7 DPLL_MPU Description

3.6.3.7.1 DPLL_MPU Overview

Figure 3-55 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

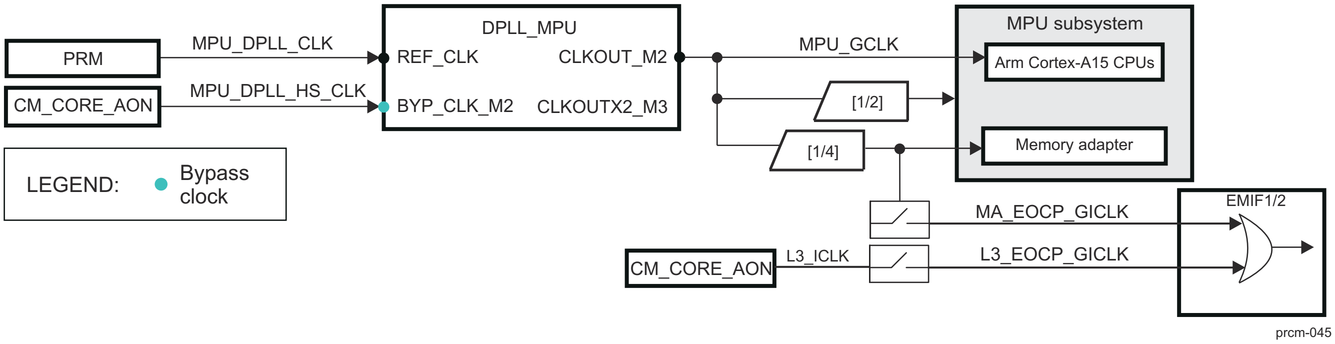


Figure 3-55. DPLL_MPU Overview

3.6.3.7.2 DPLL_MPU Tactical Clocking Adjustment

Figure 3-55 includes the clocking adjustment scheme for DPLL_MPU. Another clock is requested by the EMIF1/2 modules and this clock must be dynamically switched between L3_EOCP_GICLK clock and MA_EOCP_GICLK clock coming from the MPU subsystem (namely from Memory Adapter part of it), depending on the respective activity of MPU and EMIF clock domain.

3.6.3.7.3 DPLL_MPU Synthesized Clock Parameters

This section describes the clock synthesis and clock out divider parameters of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for an explanation of the clock synthesis and output divider parameters of the DPLL module.

Table 3-63 lists the clock synthesis parameters of the DPLL.

Table 3-63. DPLL_MPU Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_MPU[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_MPU[6:0] DPLL_DIV

Table 3-64 lists the clock output divider parameters of the DPLL.

Table 3-64. DPLL_MPU Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_MPU[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_MPU[4:0] DIVHS
CLKOUT_M2 - DCC	DCC feature control	CM_CLKSEL_DPLL_MPU[22] DCC_EN

3.6.3.7.4 DPLL_MPU Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes, and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode*, and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-65 lists the operating modes supported by the DPLL.

Table 3-65. DPLL_MPU Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-66 lists the control bit fields for the operating mode control of the DPLL.

Table 3-66. DPLL_MPU Mode Control Parameters

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_MPU[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_MPU[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_MPU[2:0] AUTO_DPLL_MODE

Note

The user software must ensure that the MPU voltage domain is on before programming DPLL_MPU. This can be ensured by performing a forced wakeup (CLKCTRL= SW_WKUP) on MPU domain. When software detects that the domain is "ON", DPLL_MPU can be programmed.

3.6.3.7.5 DPLL_MPU Recalibration

Table 3-67 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-67. DPLL_MPU Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_MPU[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[1] DPLL_MPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[1] DPLL_MPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[1] DPLL_MPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[1] DPLL_MPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[1] DPLL_MPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[1] DPLL_MPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[1] DPLL_MPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[1] DPLL_MPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[1] DPLL_MPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[1] DPLL_MPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[1] DPLL_MPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[1] DPLL_MPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[1] DPLL_MPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[1] DPLL_MPU_RECAL_EN

3.6.3.8 DPLL_IVA Description

3.6.3.8.1 DPLL_IVA Overview

Figure 3-56 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

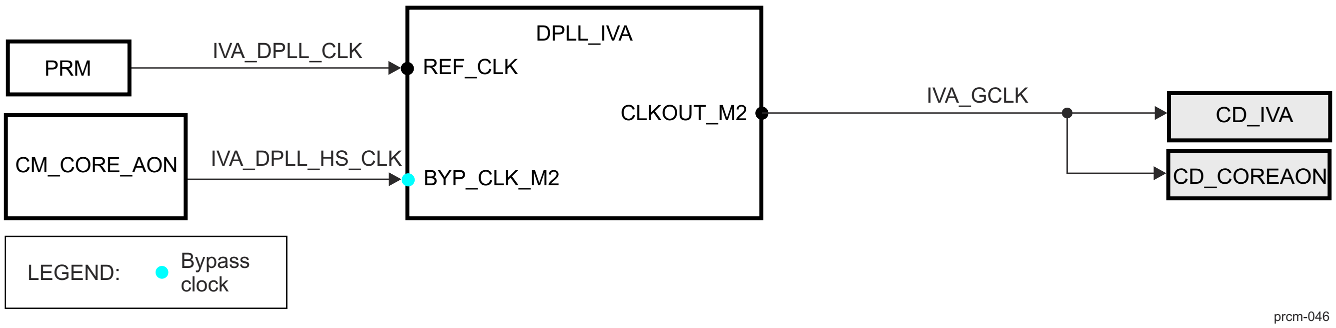


Figure 3-56. DPLL_IVA Overview

3.6.3.8.2 DPLL_IVA Synthesized Clock Parameters

This section describes the clock synthesis and clock-out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-68 lists the clock synthesis parameters of the DPLL.

Table 3-68. DPLL_IVA Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_IVA[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_IVA[6:0] DPLL_DIV

Table 3-69 lists the clock output divider parameters of the DPLL.

Table 3-69. DPLL_IVA Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_IVA[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_IVA[4:0] DIVHS

3.6.3.8.3 DPLL_IVA Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode*, and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-70 lists the operating modes supported by the DPLL.

Table 3-70. DPLL_IVA Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-71 lists the control bit fields for the operating mode control of the DPLL.

Table 3-71. DPLL_IVA Mode Control Parameters

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_IVA[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_IVA[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_IVA[2:0] AUTO_DPLL_MODE

3.6.3.8.4 DPLL_IVA Recalibration

Table 3-72 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-72. DPLL_IVA Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_IVA[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[2] DPLL_IVA_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[2] DPLL_IVA_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[2] DPLL_IVA_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[2] DPLL_IVA_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[2] DPLL_IVA_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[2] DPLL_IVA_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[2] DPLL_IVA_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[2] DPLL_IVA_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[2] DPLL_IVA_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[2] DPLL_IVA_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[2] DPLL_IVA_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[2] DPLL_IVA_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[2] DPLL_IVA_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[2] DPLL_IVA_RECAL_EN

3.6.3.9 DPLL_USB Description

3.6.3.9.1 DPLL_USB Overview

Figure 3-57 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

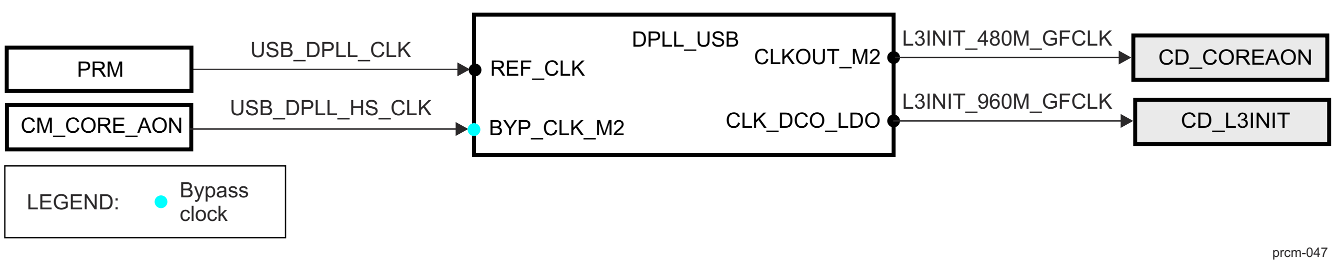


Figure 3-57. DPLL_USB Overview

3.6.3.9.2 DPLL_USB Synthesized Clock Parameters

This section describes the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-73 lists the clock synthesis parameters of the DPLL.

Table 3-73. DPLL_USB Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_USB[19:8] DPLL_MULT
N	CM_CLKSEL_DPLL_USB[7:0] DPLL_DIV

Table 3-74 lists the clock output divider parameters of the DPLL.

Table 3-74. DPLL_USB Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_USB[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_USB[6:0] DIVHS
CLK_DCO_LDO	DCO output control	CM_CLKSEL_DPLL_USB[21] DPLL_SELFREQDCO

3.6.3.9.3 DPLL_USB Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-75 lists the operating modes supported by the DPLL.

Table 3-75. DPLL_USB Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Available	Not available	Available	Not available	Available

Table 3-76 lists the control bit fields for the operating mode control of the DPLL.

Table 3-76. DPLL_USB Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_USB[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_USB[2:0] AUTO_DPLL_MODE

3.6.3.9.4 DPLL_USB Recalibration

Table 3-77 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-77. DPLL_USB Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[13] DPLL_USB_RECAL_EN(1)

(1) DPLL_USB recalibration feature is not supported in this family of devices.

3.6.3.10 DPLL_EVE Description

3.6.3.10.1 DPLL_EVE Overview

Figure 3-58 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

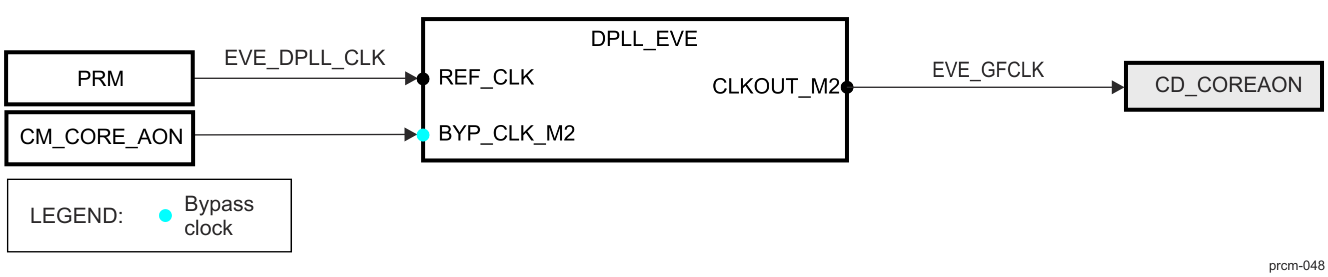


Figure 3-58. DPLL_EVE Overview

3.6.3.10.2 DPLL_EVE Synthesized Clock Parameters

This section describes the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-78 lists the clock synthesis parameters of the DPLL.

Table 3-78. DPLL_EVE Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_EVE[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_EVE[6:0] DPLL_DIV

Table 3-79 lists the clock output divider parameters of the DPLL.

Table 3-79. DPLL_EVE Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_EVE[9] CLKST
CLKOUT_M2	Divider Control	CM_DIV_M2_DPLL_EVE[4:0] DIVHS

3.6.3.10.3 DPLL_EVE Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-80 lists the operating modes supported by the DPLL.

Table 3-80. DPLL_EVE Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-81 lists the control bit fields for the operating mode control of the DPLL.

Table 3-81. DPLL_EVE Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_EVE[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_EVE[2:0] AUTO_DPLL_MODE

3.6.3.10.4 DPLL_EVE Recalibration

Table 3-82 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-82. DPLL_EVE Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_EVE[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[12] DPLL_EVE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[11] DPLL_EVE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[12] DPLL_EVE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[12] DPLL_EVE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[12] DPLL_EVE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[12] DPLL_EVE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[12] DPLL_EVE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[12] DPLL_EVE_RECAL_EN

Table 3-82. DPLL_EVE Recalibration Feature Parameters (continued)

Parameter Name	Control/Status Bit Field
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[12] DPLL_EVE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[12] DPLL_EVE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[12] DPLL_EVE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[12] DPLL_EVE_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[12] DPLL_EVE_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[12] DPLL_EVE_RECAL_EN

3.6.3.11 DPLL_DSP Description

3.6.3.11.1 DPLL_DSP Overview

Figure 3-59 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

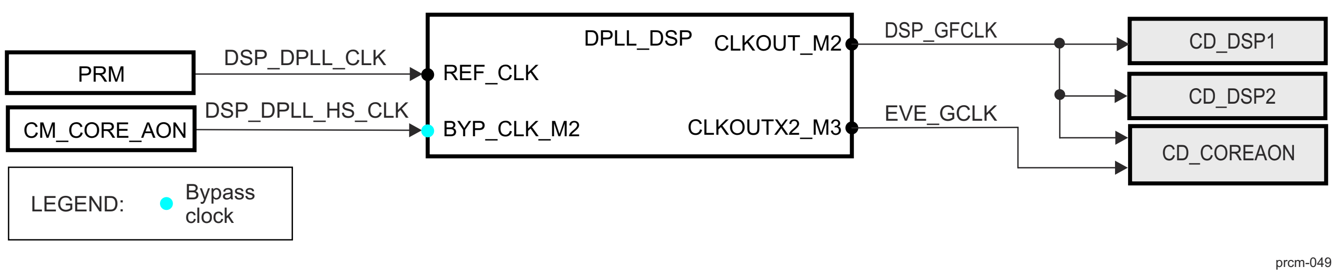


Figure 3-59. DPLL_DSP Overview

3.6.3.11.2 DPLL_DSP Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-83 lists the clock synthesis parameters of the DPLL.

Table 3-83. DPLL_DSP Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_DSP[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_DSP[6:0] DPLL_DIV

Table 3-84 lists the clock output divider parameters of the DPLL.

Table 3-84. DPLL_DSP Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_DSP[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_DSP[4:0] DIVHS
CLKOUTX2_M3	Status	CM_DIV_M3_DPLL_DSP[9] CLKST
CLKOUTX2_M3	Divider control	CM_DIV_M3_DPLL_DSP[4:0] DIVHS

3.6.3.11.3 DPLL_DSP Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-85 lists the operating modes supported by the DPLL.

Table 3-85. DPLL_DSP Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Available	Available	Available	Available	Not available

Table 3-86 lists the control bit fields for the operating mode control of the DPLL.

Table 3-86. DPLL_DSP Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_DSP[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_DSP[2:0] AUTO_DPLL_MODE

3.6.3.11.4 DPLL_DSP Recalibration

Table 3-87 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-87. DPLL_DSP Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_DSP[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[11] DPLL_DSP_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[10] DPLL_DSP_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[11] DPLL_DSP_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[11] DPLL_DSP_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[11] DPLL_DSP_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[11] DPLL_DSP_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[11] DPLL_DSP_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[11] DPLL_DSP_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[11] DPLL_DSP_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[11] DPLL_DSP_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[11] DPLL_DSP_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[11] DPLL_DSP_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[11] DPLL_DSP_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[11] DPLL_DSP_RECAL_EN

3.6.3.12 DPLL_GMAC Description

3.6.3.12.1 DPLL_GMAC Overview

Figure 3-60 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

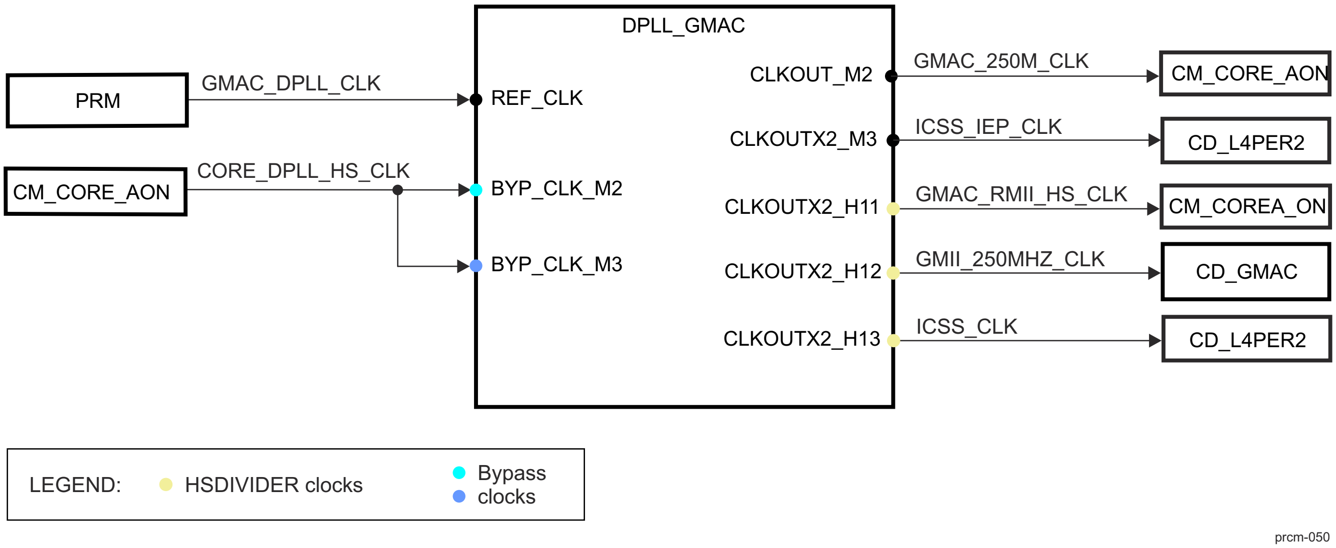


Figure 3-60. DPLL_GMAC Overview

Note

The PRU-ICSS1 and PRU-ICSS2 modules are not supported in this family of devices; therefore, the associated ICSS_CLK and ICSS_IEP_CLK clocks and related status bits are not functional.

3.6.3.12.2 DPLL_GMAC Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-88 lists the clock synthesis parameters of the DPLL.

Table 3-88. DPLL_GMAC Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_GMAC[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_GMAC[6:0] DPLL_DIV

Table 3-89 lists the clock output divider parameters of the DPLL.

Table 3-89. DPLL_GMAC Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_GMAC[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_GMAC[4:0] DIVHS
CLKOUT_M3	Status	CM_DIV_M3_DPLL_GMAC[9] CLKST
CLKOUT_M3	Divider control	CM_DIV_M3_DPLL_GMAC[4:0] DIVHS
CLKOUT_H11	Status	CM_DIV_H11_DPLL_GMAC[9] CLKST
CLKOUT_H11	Divider control	CM_DIV_H11_DPLL_GMAC[5:0] DIVHS
CLKOUT_H12	Status	CM_DIV_H12_DPLL_GMAC[9] CLKST
CLKOUT_H12	Divider control	CM_DIV_H12_DPLL_GMAC[5:0] DIVHS
CLKOUT_H13	Status	CM_DIV_H13_DPLL_GMAC[9] CLKST
CLKOUT_H13	Divider control	CM_DIV_H13_DPLL_GMAC[5:0] DIVHS

3.6.3.12.3 DPLL_GMAC Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-90 lists the operating modes supported by the DPLL.

Table 3-90. DPLL_GMAC Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-91 lists the control bit fields for the operating mode control of the DPLL.

Table 3-91. DPLL_GMAC Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_GMAC[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_GMAC[2:0] AUTO_DPLL_MODE

3.6.3.12.4 DPLL_GMAC Recalibration

Table 3-92 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-92. DPLL_GMAC Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_GMAC[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[5] DPLL_GMAC_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[5] DPLL_GMAC_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[5] DPLL_GMAC_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[5] DPLL_GMAC_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[5] DPLL_GMAC_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[5] DPLL_GMAC_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[5] DPLL_GMAC_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[5] DPLL_GMAC_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[5] DPLL_GMAC_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[5] DPLL_GMAC_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[5] DPLL_GMAC_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[5] DPLL_GMAC_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[5] DPLL_GMAC_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[5] DPLL_GMAC_RECAL_EN

3.6.3.13 DPLL_GPU Description

3.6.3.13.1 DPLL_GPU Overview

Figure 3-61 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

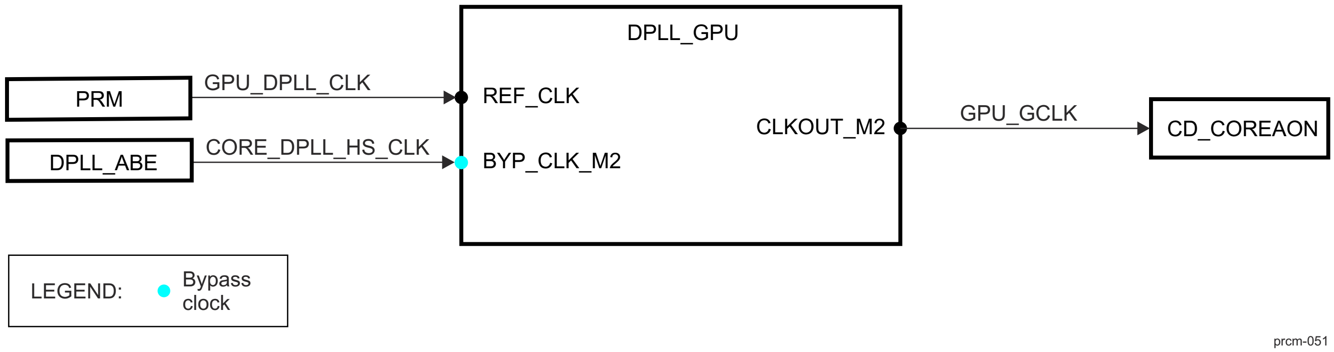


Figure 3-61. DPLL_GPU Overview

3.6.3.13.2 DPLL_GPU Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-93 lists the clock synthesis parameters of the DPLL.

Table 3-93. DPLL_GPU Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_GPU[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_GPU[6:0] DPLL_DIV

Table 3-94 lists the clock output divider parameters of the DPLL.

Table 3-94. DPLL_GPU Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_GPU[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_GPU[4:0] DIVHS

3.6.3.13.3 DPLL_GPU Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-95 lists the operating modes supported by the DPLL.

Table 3-95. DPLL_GPU Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-96 lists the control bit fields for the operating mode control of the DPLL.

Table 3-96. DPLL_GPU Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_GPU[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_GPU[2:0] AUTO_DPLL_MODE

3.6.3.13.4 DPLL_GPU Recalibration

Table 3-97 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-97. DPLL_GPU Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_GPU[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[6] DPLL_GPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[6] DPLL_GPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[6] DPLL_GPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[6] DPLL_GPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[6] DPLL_GPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[6] DPLL_GPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[6] DPLL_GPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[6] DPLL_GPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[6] DPLL_GPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[6] DPLL_GPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[6] DPLL_GPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[6] DPLL_GPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[6] DPLL_GPU_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[6] DPLL_GPU_RECAL_EN

3.6.3.14 DPLL_DDR Description

3.6.3.14.1 DPLL_DDR Overview

Figure 3-62 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

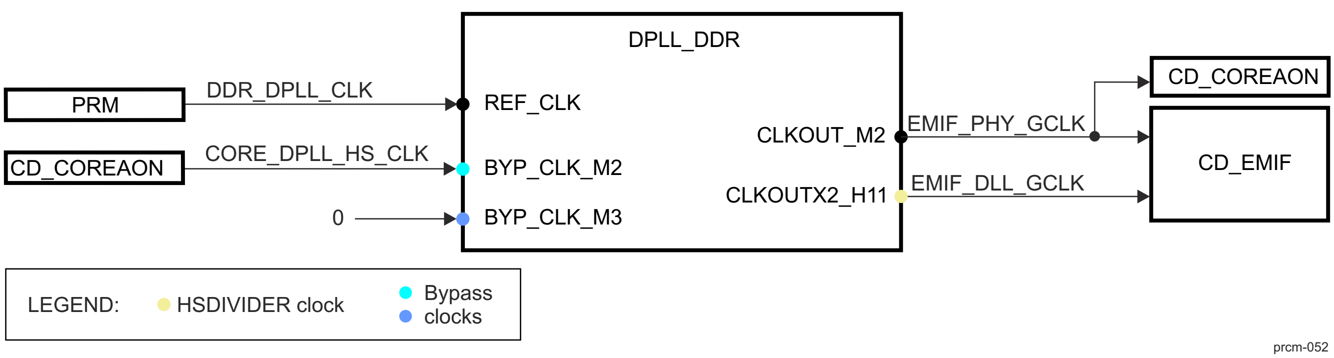


Figure 3-62. DPLL_DDR Overview

3.6.3.14.2 DPLL_DDR Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-98 lists the clock synthesis parameters of the DPLL.

Table 3-98. DPLL_DDR Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_DDR[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_DDR[6:0] DPLL_DIV

Table 3-99 lists the clock output divider parameters of the DPLL.

Table 3-99. DPLL_DDR Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_DDR[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_DDR[4:0] DIVHS
CLKOUTX2_H11	Status	CM_DIV_H11_DPLL_DDR[9] CLKST
CLKOUTX2_H11	Divider control	CM_DIV_H11_DPLL_DDR[5:0] DIVHS

3.6.3.14.3 DPLL_DDR Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-100 lists the operating modes supported by the DPLL.

Table 3-100. DPLL_DDR Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-101 lists the control bit fields for the operating mode control of the DPLL.

Table 3-101. DPLL_DDR Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_DDR[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_DDR[2:0] AUTO_DPLL_MODE

3.6.3.14.4 DPLL_DDR Recalibration

Table 3-102 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.3.5, *DPLL Recalibration*.

Table 3-102. DPLL_DDR Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration – Enable Control	CM_CLKMODE_DPLL_DDR[8] DPLL_DRIFTGUARD_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_MPU[7] DPLL_DDR_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_MPU[7] DPLL_DDR_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU1[7] DPLL_DDR_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU1[7] DPLL_DDR_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_IPU2[7] DPLL_DDR_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_IPU2[7] DPLL_DDR_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP1[7] DPLL_DDR_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP1[7] DPLL_DDR_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_DSP2[7] DPLL_DDR_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_DSP2[7] DPLL_GPU_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE1[7] DPLL_DDR_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE1[7] DPLL_DDR_RECAL_EN
Recalibration – Interrupt Status	PRM_IRQSTATUS_EVE2[7] DPLL_DDR_RECAL_ST
Recalibration – Interrupt Mask Control	PRM_IRQENABLE_EVE2[7] DPLL_DDR_RECAL_EN

3.6.3.15 DPLL_PCIE_REF Description

3.6.3.15.1 DPLL_PCIE_REF Overview

Figure 3-63 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

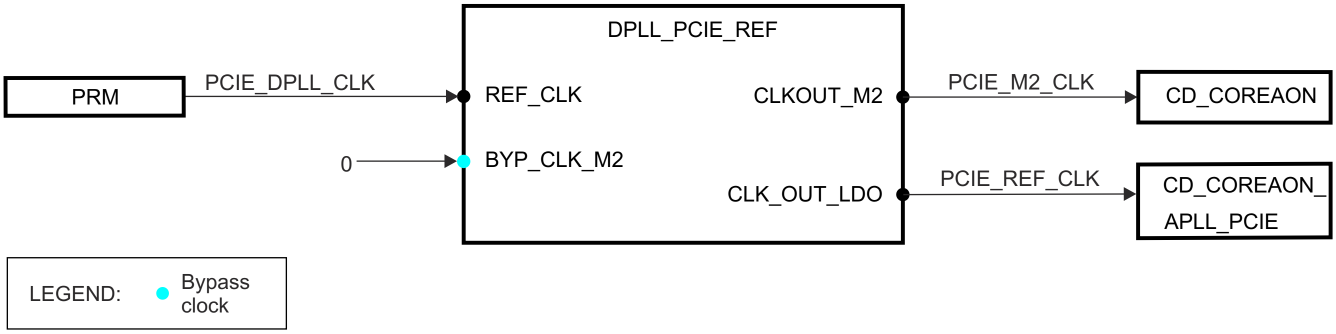


Figure 3-63. DPLL_PCIE_REF Overview

3.6.3.15.2 DPLL_PCIE_REF Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.3, *Generic DPLL Overview*.

Table 3-103 lists the clock synthesis parameters of the DPLL.

Table 3-103. DPLL_PCIE_REF Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_PCIE_REF[19:8] DPLL_MULT
N	CM_CLKSEL_DPLL_PCIE_REF[7:0] DPLL_DIV

Table 3-104 lists the clock output divider parameters of the DPLL.

Table 3-104. DPLL_PCIE_REF Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUTX2_M2	Status	CM_DIV_M2_DPLL_PCIE_REF[9] CLKST
CLKOUTX2_M2	Divider control	CM_DIV_M2_DPLL_PCIE_REF[6:0] DIVHS
CLK_OUT_LDO	Status	CM_DIV_M2_DPLL_PCIE_REF[10] CLKLDOST

3.6.3.15.3 DPLL_PCIE_REF Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.3.3, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.3.4, *DPLL Power Modes*.

Table 3-105 lists the operating modes supported by the DPLL.

Table 3-105. DPLL_PCIE_REF Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Not Available	Not Available	Available

Table 3-106 lists the control bit fields for the operating mode control of the DPLL.

Table 3-106. DPLL_PCIE_REF Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_PCIE_REF[2:0] AUTO_DPLL_MODE

3.6.3.16 APLL_PCIE Description

3.6.3.16.1 APLL_PCIE Overview

Figure 3-64 is an overview of the APLL. For a functional overview of a generic APLL module, see Section 3.6.3.3.1, *Generic APLL Overview*.

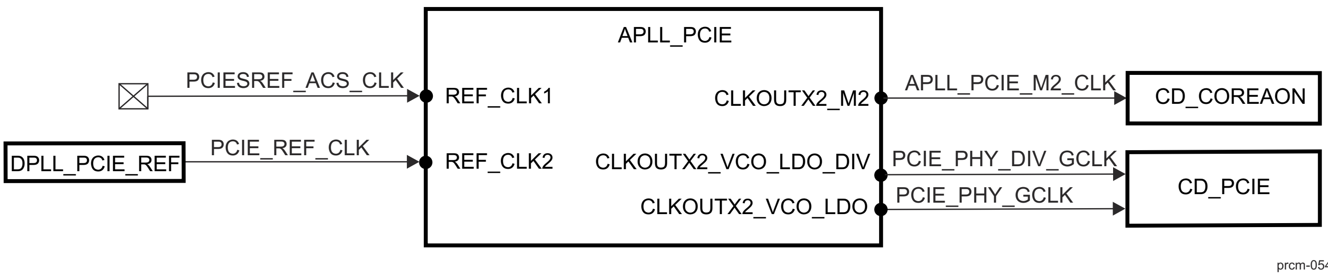


Figure 3-64. APLL_PCIE Overview

3.6.3.16.2 APLL_PCIE Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the APLL.

Table 3-107 lists the clock output divider parameters of the APLL.

Table 3-107. APLL_PCIE Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_APLL_PCIE[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_APLL_PCIE[6:0] DIVHS
CLKOUTX2_VCO_LDO_DIV	Status	CM_CLKVCOLDO_APLL_PCIE[10] CLK_DIVST
CLKOUTX2_VCO_LDO	Status	CM_CLKVCOLDO_APLL_PCIE[9] CLKST

3.6.3.16.3 APLL_PCIE Power Modes

This section identifies the operating modes supported by the APLL and the control bit fields to set its operating modes.

Note

In order to disable the APLL_PCIE, the user needs to disable PCIe_SSx (where x = 1 or 2) using the CM_PCIE_PCIESSx_CLKCTRL[1:0] MODULEMODE registers. When PCIe_SS is disabled, the PRCM module automatically disables the APLL_PCIE. Please note that setting CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT bitfield to 0x0 does not disable the APLL_PCIE.

Table 3-105 lists the operating modes supported by the DPLL.

Table 3-108. APLL_PCIE Modes

Auto Idle	Force Lock
Available	Available

Table 3-109 lists the control bit fields for the operating mode control of the APLL.

Table 3-109. APLL_PCIE Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT

3.6.4 Clock Domains

In this sections are summarized and described all Clock Domains in the device.

3.6.4.1 CD_WKUPAON Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.1.1 Overview

Figure 3-65 is an overview of the clock domain.

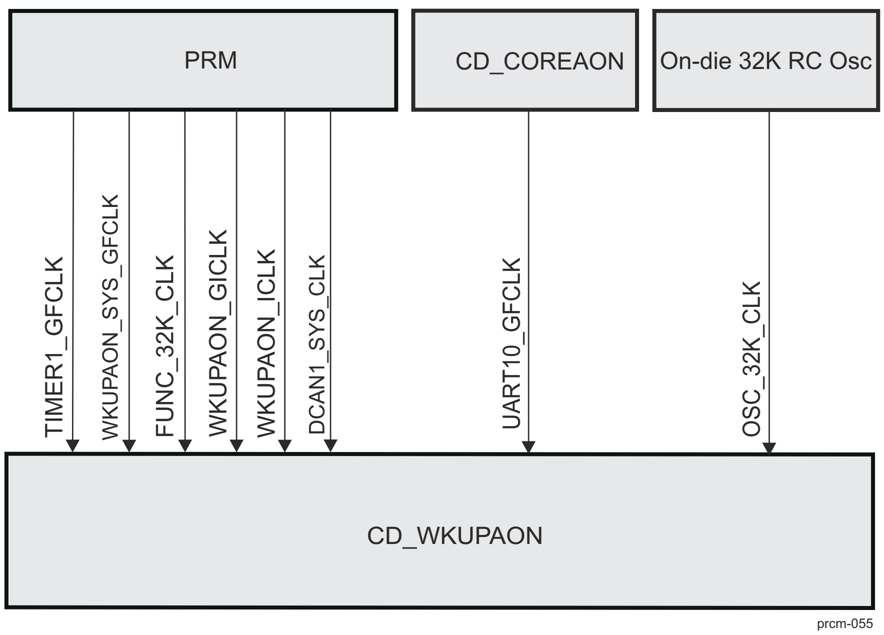


Figure 3-65. CD_WKUPAON Overview

Note

The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

3.6.4.1.2 Clock Domain Modes

Table 3-110 lists the clock domain modes supported by the clock domain.

Table 3-110. CD_WKUPAON Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-111 lists the clock domain state transition control and status bits for the clock in this clock domain

Table 3-111. CD_WKUPAON Control and Status Parameters

Parameter Name	Control/Status Bit Field
ABE_LP_CLK Clock Status	CM_WKUPAON_CLKSTCTRL[9] CLKACTIVITY_ABE_LP_CLK
WKUPAON_GICLK Clock Status	CM_WKUPAON_CLKSTCTRL[12] CLKACTIVITY_WKUPAON_GICLK

Table 3-111. CD_WKUPAON Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
SYS_CLK Clock Status; includes profiling EMU_SYS_CLK and all functional SYS_CLK	CM_WKUPAON_CLKSTCTRL[8] CLKACTIVITY_SYS_CLK
SYS_CLK Clock Status of functional branches, exclude activity of the EMU_SYS_GCLK clock	CM_WKUPAON_CLKSTCTRL[14] CLKACTIVITY_SYS_CLK_FUNC
WKUPAON_SYS_GFCLK Clock Status	CM_WKUPAON_CLKSTCTRL[11] CLKACTIVITY_WKUPAON_SYS_GFCLK
WKUPAON_32K_GFCLK Clock Control for GPIO1	CM_WKUPAON_GPIO1_CLKCTRL[8] OPTFCLKEN_DBCLK
DCAN1_SYS_CLK Clock Status	CM_WKUPAON_CLKSTCTRL[16] CLKACTIVITY_DCAN1_SYS_CLK
TIMER1_GFCLK Clock Status	CM_WKUPAON_CLKSTCTRL[17] CLKACTIVITY_TIMER1_GFCLK
UART10_GFCLK Clock Status	CM_WKUPAON_CLKSTCTRL[18] CLKACTIVITY_UART10_GFCLK
Clock Domain State Transition Control	CM_WKUPAON_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.1.3 Clock Domain Dependency

CD_WKUPAON has no static dependency or dynamic dependency with any other clock domain of the device.

3.6.4.1.3.1 Wake-Up Dependency

Table 3-112 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-112. CD_WKUPAON Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO1	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_GPIO1_WKDEP[2] WKUPDEP_GPIO1_IRQ1_DSP1	Read/write
GPIO1	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_GPIO1_WKDEP[5] WKUPDEP_GPIO1_IRQ1_DSP2	Read/write
GPIO1	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_GPIO1_WKDEP[12] WKUPDEP_GPIO1_IRQ2_DSP1	Read/write
GPIO1	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_GPIO1_WKDEP[15] WKUPDEP_GPIO1_IRQ2_DSP2	Read/write
GPIO1	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_GPIO1_WKDEP[4] WKUPDEP_GPIO1_IRQ1_IPU1	Read/write
GPIO1	CD_WKUPAON	CD_IPU2	Disabled	PM_WKUPAON_GPIO1_WKDEP[1] WKUPDEP_GPIO1_IRQ1_IPU2	Read/write
GPIO1	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_GPIO1_WKDEP[14] WKUPDEP_GPIO1_IRQ2_IPU1	Read/write
GPIO1	CD_WKUPAON	CD_IPU2	Disabled	PM_WKUPAON_GPIO1_WKDEP[11] WKUPDEP_GPIO1_IRQ2_IPU2	Read/write
GPIO1	CD_WKUPAON	CD_MPU	Disabled	PM_WKUPAON_GPIO1_WKDEP[0] WKUPDEP_GPIO1_IRQ1_MPU	Read/write

Table 3-112. CD_WKUPAON Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO1	CD_WKUPAON	CD_MPU	Disabled	PM_WKUPAON_GPIO1_WKDEP[10] WKUPDEP_GPIO1_IRQ2_MPU	Read/write
GPIO1	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_GPIO1_WKDEP[6] WKUPDEP_GPIO1_IRQ1_EVE1	Read/write
GPIO1	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_GPIO1_WKDEP[16] WKUPDEP_GPIO1_IRQ2_EVE1	Read/write
GPIO1	CD_WKUPAON	CD_EVE2	Disabled	PM_WKUPAON_GPIO1_WKDEP[7] WKUPDEP_GPIO1_IRQ1_EVE2	Read/write
GPIO1	CD_WKUPAON	CD_EVE2	Disabled	PM_WKUPAON_GPIO1_WKDEP[17] WKUPDEP_GPIO1_IRQ2_EVE2	Read/write
KBD	CD_WKUPAON	CD_MPU	Disabled	PM_WKUPAON_KBD_WKDEP[0] WKUPDEP_KBD_MPU	Read/write
KBD	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_KBD_WKDEP[2] WKUPDEP_KBD_DSP1	Read/write
KBD	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_KBD_WKDEP[5] WKUPDEP_KBD_DSP2	Read/write
KBD	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_KBD_WKDEP[4] WKUPDEP_KBD_IPU1	Read/write
KBD	CD_WKUPAON	CD_IPU2	Disabled	PM_WKUPAON_KBD_WKDEP[1] WKUPDEP_KBD_IPU2	Read/write
KBD	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_KBD_WKDEP[6] WKUPDEP_KBD_EVE1	Read/write
KBD	CD_WKUPAON	CD_EVE2	Disabled	PM_WKUPAON_KBD_WKDEP[7] WKUPDEP_KBD_EVE2	Read/write
TIMER1	CD_WKUPAON	CD_MPU	Disabled	PM_WKUPAON_TIMER1_WKDEP[0] WKUPDEP_TIMER1_MPU	Read/write
TIMER1	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_TIMER1_WKDEP[2] WKUPDEP_TIMER1_DSP1	Read/write

Table 3-112. CD_WKUPAON Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER1	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_TIMER1_WKDEP[5] WKUPDEP_TIMER1_DSP2	Read/write
TIMER1	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_TIMER1_WKDEP[4] WKUPDEP_TIMER1_IPU1	Read/write
TIMER1	CD_WKUPAON	CD_IPU2	Disabled	PM_WKUPAON_TIMER1_WKDEP[1] WKUPDEP_TIMER1_IPU2	Read/write
TIMER1	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_TIMER1_WKDEP[6] WKUPDEP_TIMER1_EVE1	Read/write
TIMER1	CD_WKUPAON	CD_EVE2	Disabled	PM_WKUPAON_TIMER1_WKDEP[7] WKUPDEP_TIMER1_EVE2	Read/write
TIMER12	CD_WKUPAON	CD_MPU	Disabled	PM_WKUPAON_TIMER12_WKDEP[0] WKUPDEP_TIMER12_MPU	Read/write
TIMER12	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_TIMER12_WKDEP[2] WKUPDEP_TIMER12_DSP1	Read/write
TIMER12	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_TIMER12_WKDEP[5] WKUPDEP_TIMER12_DSP2	Read/write
TIMER12	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_TIMER12_WKDEP[4] WKUPDEP_TIMER12_IPU1	Read/write
TIMER12	CD_WKUPAON	CD_IPU2	Disabled	PM_WKUPAON_TIMER12_WKDEP[1] WKUPDEP_TIMER12_IPU2	Read/write
TIMER12	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_TIMER12_WKDEP[6] WKUPDEP_TIMER12_EVE1	Read/write
TIMER12	CD_WKUPAON	CD_EVE2	Disabled	PM_WKUPAON_TIMER12_WKDEP[7] WKUPDEP_TIMER12_EVE2	Read/write
WD_TIMER2	CD_WKUPAON	CD_MPU	Disabled	PM_WKUPAON_WD_TIMER2_WKDEP[0] WKUPDEP_WD_TIMER2_MPU	Read/write
WD_TIMER2	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_WD_TIMER2_WKDEP[2] WKUPDEP_WD_TIMER2_DSP1	Read/write

Table 3-112. CD_WKUPAON Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
WD_TIMER2	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_WD_TIMER2_WKDEP[5] WKUPDEP_WD_TIMER2_DSP2	Read/write
WD_TIMER2	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_WD_TIMER2_WKDEP[4] WKUPDEP_WD_TIMER2_IPU1	Read/write
WD_TIMER2	CD_WKUPAON	CD_IPU2	Disabled	PM_WKUPAON_WD_TIMER2_WKDEP[1] WKUPDEP_WD_TIMER2_IPU2	Read/write
WD_TIMER2	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_WD_TIMER2_WKDEP[6] WKUPDEP_WD_TIMER2_EVE1	Read/write
WD_TIMER2	CD_WKUPAON	CD_EVE2	Disabled	PM_WKUPAON_WD_TIMER2_WKDEP[7] WKUPDEP_WD_TIMER2_EVE2	Read/write

3.6.4.1.4 Clock Domain Module Attributes

Table 3-113 lists for each module of the clock domain the clocks it receives and their role (that is, functional or interface clock).

Table 3-113. CD_WKUPAON Modules Clocks Association

Module	Clock	Clock Type
PRCM_MPU	FUNC_32K_CLK	Functional
	WKUPAON_ICLK	Interface
GPIO1	WKUPAON_GICLK	Interface
	WKUPAON_SYS_GFCLK	Functional
KBD	WKUPAON_GICLK	Interface
	WKUPAON_SYS_GFCLK	Functional
COUNTER_32K	FUNC_32K_CLK	Functional
	WKUPAON_GICLK	Interface
TIMER1	TIMER1_GFCLK	Functional
	WKUPAON_GICLK	Interface
TIMER12	WKUPAON_GICLK	Interface
	OSC_32K_CLK(1)	Functional
WD_TIMER2	WKUPAON_GICLK	Interface
	WKUPAON_SYS_GFCLK	Functional
CTRL_MODULE_WKUP	WKUPAON_GICLK	Interface
L4_WKUP interconnect	WKUPAON_GICLK	Interface
DCAN1	DCAN1_SYS_CLK	Functional
	WKUPAON_GICLK	Interface
UART10	UART10_GFCLK	Functional
	WKUPAON_GICLK	Interface

(1) The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

Table 3-114 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-114. CD_WKUPAON Modules Wake-Up Request

Module	Wake-Up Feature
PRCM_MPU	None
CTRL_MODULE_WKUP	None
DCAN1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ or DMA_SYSTEM-DMA)
GPIO1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
KBD	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
COUNTER_32K	None
TIMER1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
TIMER12	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
WD_TIMER2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
L4_WKUP interconnect	None
UART10	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ or DMA_SYSTEM-DMA)

Table 3-115 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-115. CD_WKUPAON Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
GPIO1	Slave	CM_WKUPAON_GPIO1_CLKCTRL[17:16] IDLEST	Idle status
KBD	Slave	CM_WKUPAON_KBD_CLKCTRL[17:16] IDLEST	Idle status
COUNTER_32K	Slave	CM_WKUPAON_COUNTER_32K_CLKCTRL[17:16] IDLEST	Idle status
TIMER1	Slave	CM_WKUPAON_TIMER1_CLKCTRL[17:16] IDLEST	Idle status
TIMER12	Slave	CM_WKUPAON_TIMER12_CLKCTRL[17:16] IDLEST	Idle status
WD_TIMER2	Slave	CM_WKUPAON_WD_TIMER2_CLKCTRL[17:16] IDLEST	Idle status
L4_WKUP interconnect	Slave	CM_WKUPAON_L4_WKUP_CLKCTRL[17:16] IDLEST	Idle status
DCAN1	Slave	CM_WKUPAON_DCAN1_CLKCTRL[17:16] IDLEST	Idle status
UART10	Slave	CM_WKUPAON_UART10_CLKCTRL[17:16] IDLEST	Idle status

Table 3-116 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

Table 3-116. CD_WKUPAON Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GPIO1	Available	Available	N/A	CM_WKUPAON_GPIO1_CLKCTRL[1:0] MODULEMODE	Read/write

Table 3-116. CD_WKUPAON Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
KBD	Available	N/A	Available	CM_WKUPAON_KBD_CLKCTRL[1:0] MODULEMODE	Read/write
COUNTER_32K	N/A	Available	N/A	CM_WKUPAON_COUNTER_32K_CLKCTRL[1:0] MODULEMODE	Read only
TIMER1	Available	N/A	Available	CM_WKUPAON_TIMER1_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER12	N/A	Available	N/A	CM_WKUPAON_TIMER12_CLKCTRL[1:0] MODULEMODE	Read only
WD_TIMER2	Available	N/A	Available	CM_WKUPAON_WD_TIMER2_CLKCTRL[1:0] MODULEMODE	Read/write
L4_WKUP interconnect	N/A	Available	N/A	CM_WKUPAON_L4_WKUP_CLKCTRL[1:0] MODULEMODE	Read only
DCAN1	Available	N/A	Available	CM_WKUPAON_DCAN1_CLKCTRL[1:0] MODULEMODE	Read/write
UART10	Available	N/A	Available	CM_WKUPAON_UART10_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.2 CD_DSP1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.2.1 Overview

Figure 3-66 is an overview of the clock domain.

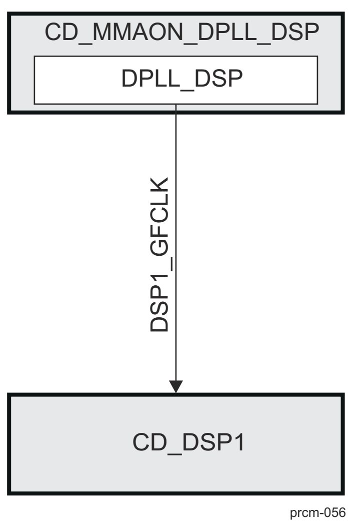


Figure 3-66. CD_DSP1 Overview

3.6.4.2.2 Clock Domain Modes

Table 3-117 lists the modes supported by the clock domain.

Table 3-117. CD_DSP1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-118 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-118. CD_DSP1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
DSP_GFCLK Clock Status	CM_DSP1_CLKSTCTRL[8] CLKACTIVITY_DSP1_GFCLK
Clock Domain State Transition Control	CM_DSP1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.2.3 Clock Domain Dependency

CD_DSP1 has no module wake-up dependency with any other clock domain of the device.

3.6.4.2.3.1 Static Dependency

Table 3-119 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-119. CD_DSP1 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CAM	Disabled	CM_DSP1_STATICDEP[9] CAM_STATDEP	Read write
CD_IVA	Disabled	CM_DSP1_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3MAIN1	Always enabled	CM_DSP1_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_L3INIT	Disabled	CM_DSP1_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Disabled	CM_DSP1_STATICDEP[12] L4CFG_STATDEP	Read only
CD_L4PER1	Disabled	CM_DSP1_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_DSP1_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_DSP1_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_L4SEC	Disabled	CM_DSP1_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_EMIF	Disabled	CM_DSP1_STATICDEP[4] EMIF_STATDEP	Read/write
CD_WKUPAON	Disabled	CM_DSP1_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Always disabled	CM_DSP1_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Always disabled	CM_DSP1_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_DSP2	Disabled	CM_DSP1_STATICDEP[18] DSP2_STATDEP	Read/write
CD_IPU	Disabled	CM_DSP1_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_DSP1_STATICDEP[23] IPU1_STATDEP	Read/write
CD_IPU2	Disabled	CM_DSP1_STATICDEP[0] IPU2_STATDEP	Read/write
CD_DSS	Disabled	CM_DSP1_STATICDEP[8] DSS_STATDEP	Read/write

Table 3-119. CD_DSP1 Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_GPU	Disabled	CM_DSP1_STATICDEP[10] GPU_STATDEP	Read/write
CD_GMAC	Disabled	CM_DSP1_STATICDEP[25] GMAC_STATDEP	Read/write
CD_VPE	Disabled	CM_DSP1_STATICDEP[28] VPE_STATDEP	Read/write
CD_PCIE	Disabled	CM_DSP1_STATICDEP[29] PCIE_STATDEP	Read/write
CD_ATL	Disabled	CM_DSP1_STATICDEP[30] ATL_STATDEP	Read/write
CD_EVE1	Disabled	CM_DSP1_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_DSP1_STATICDEP[20] EVE2_STATDEP	Read/write

3.6.4.2.3.2 Dynamic Dependency

Table 3-120 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-120. CD_DSP1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3MAIN1	Always enabled	CM_DSP1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.2.4 Clock Domain Module Attributes

Table 3-121 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-121. CD_DSP1 Modules Clocks Association

Module	Clock	Clock Type
DSP1	DSP1_GFCLK	Interface and functional

Table 3-122 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-122. CD_DSP1 Modules Wake-Up Request

Module	Wake-Up Feature
DSP1	Master wake-up request

Table 3-123 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-123. CD_DSP1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DSP1	Master/slave	CM_DSP1_DSP1_CLKCTRL[18] STBYST	Standby status
		CM_DSP1_DSP1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-124 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-124. CD_DSP1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSP	Available	Available	N/A	CM_DSP1_DSP1_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.3 CD_DSP2 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.3.1 Overview

Figure 3-67 is an overview of the clock domain.

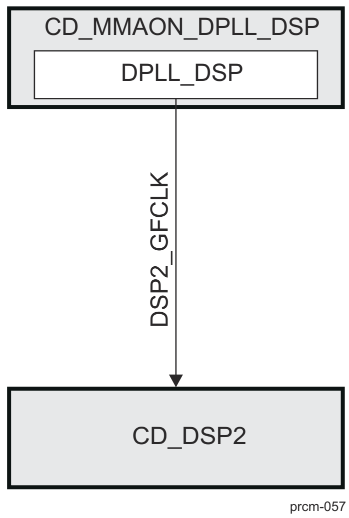


Figure 3-67. CD_DSP2 Overview

3.6.4.3.2 Clock Domain Modes

Table 3-125 lists the modes supported by the clock domain.

Table 3-125. CD_DSP2 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-126 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-126. CD_DSP2 Control and Status Parameters

Parameter Name	Control/Status Bit Field
DSP2_GFCLK Clock Status	CM_DSP2_CLKSTCTRL[8] CLKACTIVITY_DSP2_GFCLK
Clock Domain State Transition Control	CM_DSP2_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.3.3 Clock Domain Dependency

CD_DSP2 has no module wake-up dependency with any other clock domain of the device.

3.6.4.3.3.1 Static Dependency

Table 3-127 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-127. CD_DSP2 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CAM	Disabled	CM_DSP2_STATICDEP[9] CAM_STATDEP	Read write
CD_IVA	Disabled	CM_DSP2_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3MAIN1	Always enabled	CM_DSP2_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_L3INIT	Disabled	CM_DSP2_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Disabled	CM_DSP2_STATICDEP[12] L4CFG_STATDEP	Read only
CD_L4PER1	Disabled	CM_DSP2_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_DSP2_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_DSP2_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_L4SEC	Disabled	CM_DSP2_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_EMIF	Disabled	CM_DSP2_STATICDEP[4] EMIF_STATDEP	Read/write
CD_WKUPAON	Disabled	CM_DSP2_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Always disabled	CM_DSP2_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Always disabled	CM_DSP2_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_DSP1	Disabled	CM_DSP2_STATICDEP[1] DSP1_STATDEP	Read/write
CD_IPU	Disabled	CM_DSP2_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_DSP2_STATICDEP[23] IPU1_STATDEP	Read/write
CD_IPU2	Disabled	CM_DSP2_STATICDEP[0] IPU2_STATDEP	Read/write
CD_DSS	Disabled	CM_DSP2_STATICDEP[8] DSS_STATDEP	Read/write
CD_GPU	Disabled	CM_DSP2_STATICDEP[10] GPU_STATDEP	Read/write
CD_GMAC	Disabled	CM_DSP2_STATICDEP[25] GMAC_STATDEP	Read/write
CD_VPE	Disabled	CM_DSP2_STATICDEP[28] VPE_STATDEP	Read/write
CD_PCIE	Disabled	CM_DSP2_STATICDEP[29] PCIE_STATDEP	Read/write
CD_ATL	Disabled	CM_DSP2_STATICDEP[30] ATL_STATDEP	Read/write
CD_EVE1	Disabled	CM_DSP2_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_DSP2_STATICDEP[20] EVE2_STATDEP	Read/write

3.6.4.3.3.2 Dynamic Dependency

Table 3-128 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-128. CD_DSP2 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3MAIN1	Always enabled	CM_DSP1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.3.4 Clock Domain Module Attributes

Table 3-129 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-129. CD_DSP2 Modules Clocks Association

Module	Clock	Clock Type
DSP2	DSP2_GFCLK	Interface and functional

Table 3-130 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-130. CD_DSP2 Modules Wake-Up Request

Module	Wake-Up Feature
DSP2	Master wake-up request

Table 3-131 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-131. CD_DSP2 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DSP2	Master/slave	CM_DSP2_DSP2_CLKCTRL[18] STBYST	Standby status
		CM_DSP2_DSP2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-132 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-132. CD_DSP2 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSP2	Available	Available	N/A	CM_DSP2_DSP2_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.4 CD_CUSTEFUSE Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.4.1 Overview

Figure 3-68 is an overview of the clock domain.

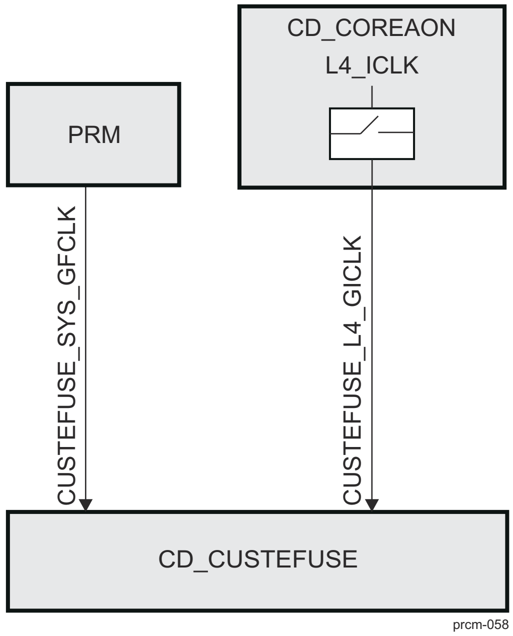


Figure 3-68. CD_CUSTEFUSE Overview

3.6.4.4.2 Clock Domain Modes

Table 3-133 lists the modes supported by the clock domain.

Table 3-133. CD_CUSTEFUSE Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	N/A	Available	Available

Table 3-134 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-134. CD_CUSTEFUSE Control and Status Parameters

Parameter Name	Control/Status Bit Field
CUSTEFUSE_SYS_GFCLK Clock Status	CM_CUSTEFUSE_CLKSTCTRL[9] CLKACTIVITY_CUSTEFUSE_SYS_GFCLK
CUSTEFUSE_L4_GICLK Clock Status	CM_CUSTEFUSE_CLKSTCTRL[8] CLKACTIVITY_CUSTEFUSE_L4_GICLK
Clock Domain State Transition Control	CM_CUSTEFUSE_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.4.3 Clock Domain Dependency

CD_CUSTEFUSE has no static or dynamic dependency with any other clock domain of the device.

3.6.4.4.4 Clock Domain Module Attributes

Table 3-135 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-135. CD_CUSTEFUSE Modules Clocks Association

Module	Clock	Clock Type
EFUSE_CTRL_CUST	CUSTEFUSE_SYS_GFCLK	Functional
	CUSTEFUSE_L4_GICLK	Interface

Table 3-136 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-136. CD_CUSTEFUSE Modules Wake-Up Request

Module	Wake-Up Feature
EFUSE_CTRL_CUST	None

Table 3-137 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-137. CD_CUSTEFUSE Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
EFUSE_CTRL_CUST	Idle	CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL[17:16] IDLEST	Idle status

Table 3-138 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-138. CD_CUSTEFUSE Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
EFUSE_CTRL_CUST	Available	N/A	Available	CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.5 CD_MPU Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.5.1 Overview

Figure 3-69 is an overview of the clock domain.

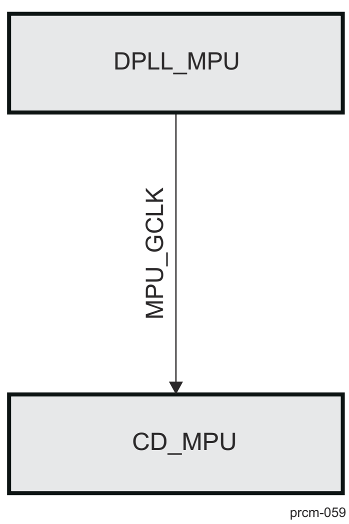


Figure 3-69. CD_MPU Overview

3.6.4.5.2 Clock Domain Modes

Table 3-139 lists the clock domain modes supported by the clock domain.

Table 3-139. CD_MPU Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-140 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-140. CD_MPU Control and Status Parameters

Parameter Name	Control/Status Bit Field
MPU_GCLK Clock Status	CM_MPU_CLKSTCTRL[8] CLKACTIVITY_MPU_GCLK
Clock Domain State Transition Control	CM_MPU_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.5.3 Clock Domain Dependency

CD_MPU has no module wake-up dependency with any other clock domain of the device.

3.6.4.5.3.1 Static Dependency

Table 3-141 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-141. CD_MPU Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CUSTEFUSE	Always disabled	CM_MPU_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_DSS	Disabled	CM_MPU_STATICDEP[8] DSS_STATDEP	Read/write
CD_IPU	Disabled	CM_MPU_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_MPU_STATICDEP[23] IPU1_STATDEP	Read/write
CD_IPU2	Disabled	CM_MPU_STATICDEP[0] IPU2_STATDEP	Read/write
CD_GPU	Disabled	CM_MPU_STATICDEP[10] GPU_STATDEP	Read/write
CD_CAM	Always disabled	CM_MPU_STATICDEP[9] CAM_STATDEP	Read only
CD_IVA	Disabled	CM_MPU_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3MAIN	Enabled	CM_MPU_STATICDEP[5] L3MAIN1_STATDEP	Read/write
CD_L3INIT	Enabled	CM_MPU_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Enabled	CM_MPU_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER1	Enabled	CM_MPU_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Enabled	CM_MPU_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Enabled	CM_MPU_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_L4SEC	Disabled	CM_MPU_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_EMIF	Enabled	CM_MPU_STATICDEP[4] EMIF_STATDEP	Read/write
CD_DMA	Always disabled	CM_MPU_STATICDEP[11] SDMA_STATDEP	Read only
CD_DSP1	Disabled	CM_MPU_STATICDEP[1] DSP1_STATDEP	Read/write
CD_DSP2	Disabled	CM_MPU_STATICDEP[1] DSP2_STATDEP	Read/write
CD_WKUPAON	Enabled	CM_MPU_STATICDEP[15] WKUPAON_STATDEP	Read/write

Table 3-141. CD_MPU Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_COREAON	Always disabled	CM_MPU_STATICDEP[16] COREAON_STATDEP	Read only
CD_GMAC	Disabled	CM_MPU_STATICDEP[25] GMAC_STATDEP	Read/write
CD_VPE	Disabled	CM_MPU_STATICDEP[28] VPE_STATDEP	Read/write
CD_PCIE	Enabled	CM_MPU_STATICDEP[29] PCIE_STATDEP	Read/write
CD_EVE1	Disabled	CM_MPU_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_MPU_STATICDEP[20] EVE2_STATDEP	Read/write

3.6.4.5.3.2 Dynamic Dependency

Table 3-142 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-142. CD_MPU Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3MAIN1	Always enabled	CM_MPU_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only
CD_EMIF	Always enabled	CM_MPU_DYNAMICDEP[4] EMIF_DYNDEP	Read only

3.6.4.5.4 Clock Domain Module Attributes

Table 3-143 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-143. CD_MPU Modules Clocks Association

Module	Clock	Clock Type
MPU	MPU_GCLK	Interface and functional

Table 3-144 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-144. CD_MPU Modules Wake-Up Request

Module	Wake-Up Feature
MPU	Master wake-up request

Table 3-145 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-145. CD_MPU Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
MPU	Master/slave	CM_MPU_MPU_CLKCTRL[18] STBYST	Standby status
		CM_MPU_MPU_CLKCTRL[17:16] IDLEST	Idle status

Table 3-146 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-146. CD_MPU Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
MPU	N/A	Available	N/A	CM_MPU_MPU_CLKCTRL[1:0] MODULEMODE	Read only

3.6.4.6 CD_L4PER1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.6.1 Overview

Figure 3-70 is an overview of the clock domain.

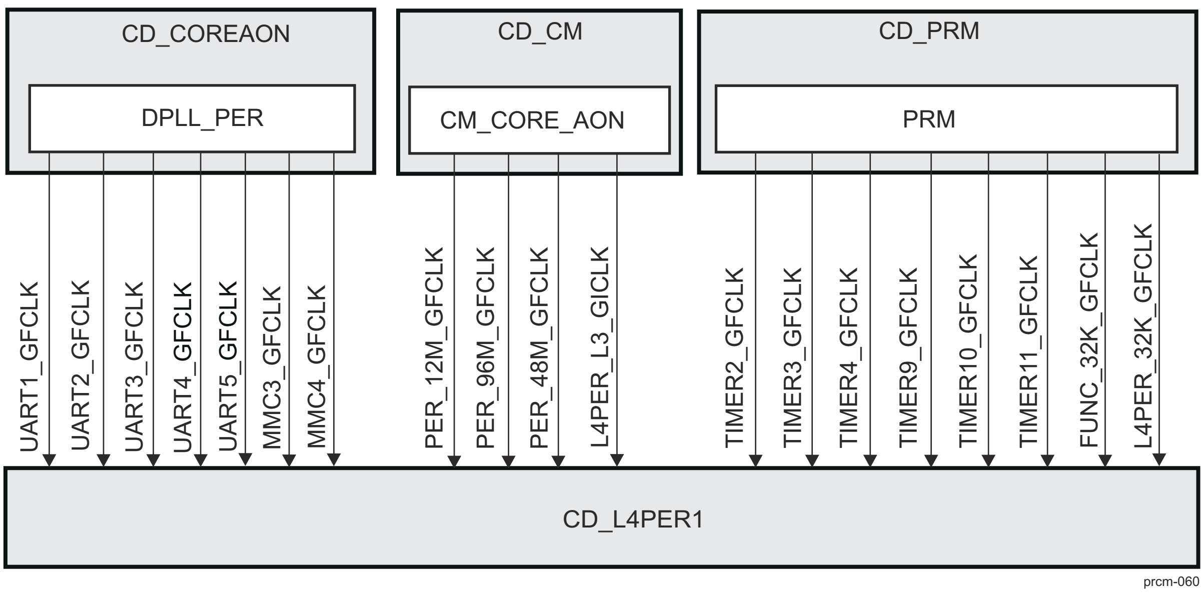


Figure 3-70. CD_L4PER1 Overview

3.6.4.6.2 Clock Domain Modes

Table 3-147 lists the clock domain modes supported by the clock domain.

Table 3-147. CD_L4PER1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-148 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-148. CD_L4PER1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
TIMER10_GFCLK clock status	CM_L4PER_CLKSTCTRL[9] CLKACTIVITY_TIMER10_GFCLK
TIMER11_GFCLK clock status	CM_L4PER_CLKSTCTRL[10] CLKACTIVITY_TIMER11_GFCLK
TIMER2_GFCLK clock status	CM_L4PER_CLKSTCTRL[11] CLKACTIVITY_TIMER2_GFCLK
TIMER3_GFCLK clock status	CM_L4PER_CLKSTCTRL[12] CLKACTIVITY_TIMER3_GFCLK
TIMER4_GFCLK clock status	CM_L4PER_CLKSTCTRL[13] CLKACTIVITY_TIMER4_GFCLK
TIMER9_GFCLK clock status	CM_L4PER_CLKSTCTRL[14] CLKACTIVITY_TIMER9_GFCLK
L4PER_L3_GICLK clock status	CM_L4PER_CLKSTCTRL[8] CLKACTIVITY_L4PER_L3_GICLK
PER_12M_GFCLK clock status	CM_L4PER_CLKSTCTRL[19] CLKACTIVITY_PER_12M_GFCLK

Table 3-148. CD_L4PER1 Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
L4PER_32K_GFCLK clock status	CM_L4PER_CLKSTCTRL[27] CLKACTIVITY_PER_32K_GFCLK
PER_48M_GFCLK clock status	CM_L4PER_CLKSTCTRL[20] CLKACTIVITY_PER_48M_GFCLK
PER_96M_GFCLK clock status	CM_L4PER_CLKSTCTRL[21] CLKACTIVITY_PER_96M_GFCLK
GPIO_GFCLK clock status	CM_L4PER_CLKSTCTRL[24] CLKACTIVITY_GPIO_GFCLK
UART1_GFCLK clock status	CM_L4PER_CLKSTCTRL[15] CLKACTIVITY_UART1_GFCLK
UART2_GFCLK clock status	CM_L4PER_CLKSTCTRL[16] CLKACTIVITY_UART2_GFCLK
UART3_GFCLK clock status	CM_L4PER_CLKSTCTRL[17] CLKACTIVITY_UART3_GFCLK
UART4_GFCLK clock status	CM_L4PER_CLKSTCTRL[18] CLKACTIVITY_UART4_GFCLK
UART5_GFCLK clock status	CM_L4PER_CLKSTCTRL[26] CLKACTIVITY_UART5_GFCLK
MMC3_GFCLK clock status	CM_L4PER_CLKSTCTRL[22] CLKACTIVITY_MMC3_GFCLK
MMC4_GFCLK clock status	CM_L4PER_CLKSTCTRL[23] CLKACTIVITY_MMC4_GFCLK
Clock Domain State Transition Control	CM_L4PER_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.6.3 Clock Domain Dependency

CD_L4PER1 has no static dependency with any other clock domain of the device.

3.6.4.6.3.1 Dynamic Dependency

Table 3-149 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-149. CD_L4PER1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4SEC	Always enabled	CM_L4PER_DYNAMICDEP[14] L4SEC_DYNDEP	Read only
CD_DSS	Always enabled	CM_L4PER_DYNAMICDEP[8] DSS_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4PER_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_IPU	Always enabled	CM_L4PER_DYNAMICDEP[3] IPU_DYNDEP	Read only

3.6.4.6.3.2 Wake-Up Dependency

Table 3-150 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER10	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER10_WKDEP[0] WKUPDEP_TIMER10_MPU	Read/write
TIMER10	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER10_WKDEP[2] WKUPDEP_TIMER10_DSP1	Read/write
TIMER10	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER10_WKDEP[5] WKUPDEP_TIMER10_DSP2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER10	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER10_WKDEP[4] WKUPDEP_TIMER10_IPU1	Read/write
TIMER10	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER10_WKDEP[1] WKUPDEP_TIMER10_IPU2	Read/write
TIMER10	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER10_WKDEP[6] WKUPDEP_TIMER10_EVE1	Read/write
TIMER10	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER10_WKDEP[7] WKUPDEP_TIMER10_EVE2	Read/write
TIMER11	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER11_WKDEP[4] WKUPDEP_TIMER11_IPU1	Read/write
TIMER11	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER11_WKDEP[1] WKUPDEP_TIMER11_IPU2	Read/write
TIMER11	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER11_WKDEP[0] WKUPDEP_TIMER11_MPU	Read/write
TIMER11	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER11_WKDEP[2] WKUPDEP_TIMER11_DSP1	Read/write
TIMER11	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER11_WKDEP[5] WKUPDEP_TIMER11_DSP2	Read/write
TIMER11	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER11_WKDEP[6] WKUPDEP_TIMER11_EVE1	Read/write
TIMER11	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER11_WKDEP[7] WKUPDEP_TIMER11_EVE2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER2	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[0] WKUPDEP_TIMER2_MPU	Read/write
TIMER2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[2] WKUPDEP_TIMER2_DSP1	Read/write
TIMER2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[5] WKUPDEP_TIMER2_DSP2	Read/write
TIMER2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[4] WKUPDEP_TIMER2_IPU1	Read/write
TIMER2	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[1] WKUPDEP_TIMER2_IPU2	Read/write
TIMER2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[6] WKUPDEP_TIMER2_EVE1	Read/write
TIMER2	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[7] WKUPDEP_TIMER2_EVE2	Read/write
TIMER3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[4] WKUPDEP_TIMER3_IPU1	Read/write
TIMER3	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[1] WKUPDEP_TIMER3_IPU2	Read/write
TIMER3	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[0] WKUPDEP_TIMER3_MPU	Read/write
TIMER3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[2] WKUPDEP_TIMER3_DSP1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[5] WKUPDEP_TIMER3_DSP2	Read/write
TIMER3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[6] WKUPDEP_TIMER3_EVE1	Read/write
TIMER3	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[7] WKUPDEP_TIMER3_EVE2	Read/write
TIMER4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[4] WKUPDEP_TIMER4_IPU1	Read/write
TIMER4	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[1] WKUPDEP_TIMER4_IPU2	Read/write
TIMER4	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[0] WKUPDEP_TIMER4_MPU	Read/write
TIMER4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[2] WKUPDEP_TIMER4_DSP1	Read/write
TIMER4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[5] WKUPDEP_TIMER4_DSP2	Read/write
TIMER4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[6] WKUPDEP_TIMER4_EVE1	Read/write
TIMER4	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[7] WKUPDEP_TIMER4_EVE2	Read/write
TIMER9	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER9_WKDEP[4] WKUPDEP_TIMER9_IPU1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER9	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER9_WKDEP[1] WKUPDEP_TIMER9_IPU2	Read/write
TIMER9	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER9_WKDEP[0] WKUPDEP_TIMER9_MPU	Read/write
TIMER9	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER9_WKDEP[2] WKUPDEP_TIMER9_DSP1	Read/write
TIMER9	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER9_WKDEP[5] WKUPDEP_TIMER9_DSP2	Read/write
TIMER9	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER9_WKDEP[6] WKUPDEP_TIMER9_EVE1	Read/write
TIMER9	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER9_WKDEP[7] WKUPDEP_TIMER9_EVE2	Read/write
GPIO2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[2] WKUPDEP_GPIO2_IRQ1_DSP1	Read/write
GPIO2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[5] WKUPDEP_GPIO2_IRQ1_DSP2	Read/write
GPIO2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[12] WKUPDEP_GPIO2_IRQ2_DSP1	Read/write
GPIO2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[15] WKUPDEP_GPIO2_IRQ2_DSP2	Read/write
GPIO2	CD_L4PER	CD_IPU1, CD_L3_MAIN1,CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[4] WKUPDEP_GPIO2_IRQ1_IPU1	Read/write
GPIO2	CD_L4PER	CD_IPU2, CD_L3_MAIN1,CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[1] WKUPDEP_GPIO2_IRQ1_IPU2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO2	CD_L4PER	CD_IPU1, CD_L3_MAIN1,CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[14] WKUPDEP_GPIO2_IRQ2_IPU1	Read/write
GPIO2	CD_L4PER	CD_IPU2, CD_L3_MAIN1,CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[11] WKUPDEP_GPIO2_IRQ2_IPU2	Read/write
GPIO2	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[0] WKUPDEP_GPIO2_IRQ1_MPU	Read/write
GPIO2	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[10] WKUPDEP_GPIO2_IRQ2_MPU	Read/write
GPIO2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[6] WKUPDEP_GPIO2_IRQ1_EVE1	Read/write
GPIO2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[16] WKUPDEP_GPIO2_IRQ2_EVE1	Read/write
GPIO2	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[7] WKUPDEP_GPIO2_IRQ1_EVE2	Read/write
GPIO2	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[17] WKUPDEP_GPIO2_IRQ2_EVE2	Read/write
GPIO3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[2] WKUPDEP_GPIO3_IRQ1_DSP1	Read/write
GPIO3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[5] WKUPDEP_GPIO3_IRQ1_DSP2	Read/write
GPIO3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[12] WKUPDEP_GPIO3_IRQ2_DSP1	Read/write
GPIO3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[15] WKUPDEP_GPIO3_IRQ2_DSP2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[4] WKUPDEP_GPIO3_IRQ1_IPU1	Read/write
GPIO3	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[1] WKUPDEP_GPIO3_IRQ1_IPU2	Read/write
GPIO3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[14] WKUPDEP_GPIO3_IRQ2_IPU1	Read/write
GPIO3	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[11] WKUPDEP_GPIO3_IRQ2_IPU2	Read/write
GPIO3	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[0] WKUPDEP_GPIO3_IRQ1_MPU	Read/write
GPIO3	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[10] WKUPDEP_GPIO3_IRQ2_MPU	Read/write
GPIO3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[6] WKUPDEP_GPIO3_IRQ1_EVE1	Read/write
GPIO3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[16] WKUPDEP_GPIO3_IRQ2_EVE1	Read/write
GPIO3	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[7] WKUPDEP_GPIO3_IRQ1_EVE2	Read/write
GPIO3	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[17] WKUPDEP_GPIO3_IRQ2_EVE2	Read/write
GPIO4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[2] WKUPDEP_GPIO4_IRQ1_DSP1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[5] WKUPDEP_GPIO4_IRQ1_DSP2	Read/write
GPIO4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[12] WKUPDEP_GPIO4_IRQ2_DSP1	Read/write
GPIO4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[15] WKUPDEP_GPIO4_IRQ2_DSP2	Read/write
GPIO4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[4] WKUPDEP_GPIO4_IRQ1_IPU1	Read/write
GPIO4	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[1] WKUPDEP_GPIO4_IRQ1_IPU2	Read/write
GPIO4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[14] WKUPDEP_GPIO4_IRQ2_IPU1	Read/write
GPIO4	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[11] WKUPDEP_GPIO4_IRQ2_IPU2	Read/write
GPIO4	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[0] WKUPDEP_GPIO4_IRQ1_MPU	Read/write
GPIO4	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[10] WKUPDEP_GPIO4_IRQ2_MPU	Read/write
GPIO4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[6] WKUPDEP_GPIO4_IRQ1_EVE1	Read/write
GPIO4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[16] WKUPDEP_GPIO4_IRQ2_EVE1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO4	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[7] WKUPDEP_GPIO4_IRQ1_EVE2	Read/write
GPIO4	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[17] WKUPDEP_GPIO4_IRQ2_EVE2	Read/write
GPIO5	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[2] WKUPDEP_GPIO5_IRQ1_DSP1	Read/write
GPIO5	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[5] WKUPDEP_GPIO5_IRQ1_DSP2	Read/write
GPIO5	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[12] WKUPDEP_GPIO5_IRQ2_DSP1	Read/write
GPIO5	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[15] WKUPDEP_GPIO5_IRQ2_DSP2	Read/write
GPIO5	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[4] WKUPDEP_GPIO5_IRQ1_IPU1	Read/write
GPIO5	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[1] WKUPDEP_GPIO5_IRQ1_IPU2	Read/write
GPIO5	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[14] WKUPDEP_GPIO5_IRQ2_IPU1	Read/write
GPIO5	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[11] WKUPDEP_GPIO5_IRQ2_IPU2	Read/write
GPIO5	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[0] WKUPDEP_GPIO5_IRQ1_MPU	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO5	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[10] WKUPDEP_GPIO5_IRQ2_MPU	Read/write
GPIO5	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[6] WKUPDEP_GPIO5_IRQ1_EVE1	Read/write
GPIO5	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[16] WKUPDEP_GPIO5_IRQ2_EVE1	Read/write
GPIO5	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[7] WKUPDEP_GPIO5_IRQ1_EVE2	Read/write
GPIO5	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO5_WKDEP[17] WKUPDEP_GPIO5_IRQ2_EVE2	Read/write
GPIO6	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[2] WKUPDEP_GPIO6_IRQ1_DSP1	Read/write
GPIO6	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[5] WKUPDEP_GPIO6_IRQ1_DSP2	Read/write
GPIO6	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[12] WKUPDEP_GPIO6_IRQ2_DSP1	Read/write
GPIO6	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[15] WKUPDEP_GPIO6_IRQ2_DSP2	Read/write
GPIO6	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[4] WKUPDEP_GPIO6_IRQ1_IPU1	Read/write
GPIO6	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[1] WKUPDEP_GPIO6_IRQ1_IPU2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO6	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[14] WKUPDEP_GPIO6_IRQ2_IPU1	Read/write
GPIO6	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[11] WKUPDEP_GPIO6_IRQ2_IPU2	Read/write
GPIO6	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[0] WKUPDEP_GPIO6_IRQ1_MPU	Read/write
GPIO6	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[10] WKUPDEP_GPIO6_IRQ2_MPU	Read/write
GPIO6	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[6] WKUPDEP_GPIO6_IRQ1_EVE1	Read/write
GPIO6	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[16] WKUPDEP_GPIO6_IRQ2_EVE1	Read/write
GPIO6	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[7] WKUPDEP_GPIO6_IRQ1_EVE2	Read/write
GPIO6	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO6_WKDEP[17] WKUPDEP_GPIO6_IRQ2_EVE2	Read/write
GPIO7	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[2] WKUPDEP_GPIO7_IRQ1_DSP1	Read/write
GPIO7	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[5] WKUPDEP_GPIO7_IRQ1_DSP2	Read/write
GPIO7	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[12] WKUPDEP_GPIO7_IRQ2_DSP1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO7	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[15] WKUPDEP_GPIO7_IRQ2_DSP2	Read/write
GPIO7	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[4] WKUPDEP_GPIO7_IRQ1_IPU1	Read/write
GPIO7	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[1] WKUPDEP_GPIO7_IRQ1_IPU2	Read/write
GPIO7	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[14] WKUPDEP_GPIO7_IRQ2_IPU1	Read/write
GPIO7	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[11] WKUPDEP_GPIO7_IRQ2_IPU2	Read/write
GPIO7	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[0] WKUPDEP_GPIO7_IRQ1_MPU	Read/write
GPIO7	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[10] WKUPDEP_GPIO7_IRQ2_MPU	Read/write
GPIO7	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[6] WKUPDEP_GPIO7_IRQ1_EVE1	Read/write
GPIO7	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[16] WKUPDEP_GPIO7_IRQ2_EVE1	Read/write
GPIO7	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[7] WKUPDEP_GPIO7_IRQ1_EVE2	Read/write
GPIO7	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO7_WKDEP[17] WKUPDEP_GPIO7_IRQ2_EVE2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO8	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[2] WKUPDEP_GPIO8_IRQ1_DSP1	Read/write
GPIO8	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[5] WKUPDEP_GPIO8_IRQ1_DSP2	Read/write
GPIO8	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[12] WKUPDEP_GPIO8_IRQ2_DSP1	Read/write
GPIO8	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[15] WKUPDEP_GPIO8_IRQ2_DSP2	Read/write
GPIO8	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[4] WKUPDEP_GPIO8_IRQ1_IPU1	Read/write
GPIO8	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[1] WKUPDEP_GPIO8_IRQ1_IPU2	Read/write
GPIO8	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[14] WKUPDEP_GPIO8_IRQ2_IPU1	Read/write
GPIO8	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[11] WKUPDEP_GPIO8_IRQ2_IPU2	Read/write
GPIO8	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[0] WKUPDEP_GPIO8_IRQ1_MPU	Read/write
GPIO8	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[10] WKUPDEP_GPIO8_IRQ2_MPU	Read/write
GPIO8	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[6] WKUPDEP_GPIO8_IRQ1_EVE1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO8	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[16] WKUPDEP_GPIO8_IRQ2_EVE1	Read/write
GPIO8	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[7] WKUPDEP_GPIO8_IRQ1_EVE2	Read/write
GPIO8	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO8_WKDEP[17] WKUPDEP_GPIO8_IRQ2_EVE2	Read/write
I2C1	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C1_WKDEP[13] WKUPDEP_I2C1_DMA_SDMA	Read/write
I2C1	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[4] WKUPDEP_I2C1_IRQ_IPU1	Read/write
I2C1	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[1] WKUPDEP_I2C1_IRQ_IPU2	Read/write
I2C1	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[0] WKUPDEP_I2C1_IRQ_MPU	Read/write
I2C1	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[2] WKUPDEP_I2C1_IRQ_DSP1	Read/write
I2C1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[5] WKUPDEP_I2C1_IRQ_DSP2	Read/write
I2C1	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C1_WKDEP[12] WKUPDEP_I2C1_DMA_DSP1	Read/write
I2C1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C1_WKDEP[15] WKUPDEP_I2C1_DMA_DSP2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
I2C1	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[6] WKUPDEP_I2C1_IRQ_EVE1	Read/write
I2C1	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[7] WKUPDEP_I2C1_IRQ_EVE2	Read/write
I2C2	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C2_WKDEP[13] WKUPDEP_I2C2_DMA_SDMA	Read/write
I2C2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[4] WKUPDEP_I2C2_IRQ_IPU1	Read/write
I2C2	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[1] WKUPDEP_I2C2_IRQ_IPU2	Read/write
I2C2	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[0] WKUPDEP_I2C2_IRQ_MPU	Read/write
I2C2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[2] WKUPDEP_I2C2_IRQ_DSP1	Read/write
I2C2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[5] WKUPDEP_I2C2_IRQ_DSP2	Read/write
I2C2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C2_WKDEP[12] WKUPDEP_I2C2_DMA_DSP1	Read/write
I2C2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C2_WKDEP[15] WKUPDEP_I2C2_DMA_DSP2	Read/write
I2C2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[6] WKUPDEP_I2C2_IRQ_EVE1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
I2C2	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[7] WKUPDEP_I2C2_IRQ_EVE2	Read/write
I2C3	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C3_WKDEP[13] WKUPDEP_I2C3_DMA_SDMA	Read/write
I2C3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C3_WKDEP[4] WKUPDEP_I2C3_IRQ_IPU1	Read/write
I2C3	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C3_WKDEP[1] WKUPDEP_I2C3_IRQ_IPU2	Read/write
I2C3	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C3_WKDEP[0] WKUPDEP_I2C3_IRQ_MPU	Read/write
I2C3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C3_WKDEP[2] WKUPDEP_I2C3_IRQ_DSP1	Read/write
I2C3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C3_WKDEP[5] WKUPDEP_I2C3_IRQ_DSP2	Read/write
I2C3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C3_WKDEP[12] WKUPDEP_I2C3_DMA_DSP1	Read/write
I2C3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C3_WKDEP[15] WKUPDEP_I2C3_DMA_DSP2	Read/write
I2C3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C3_WKDEP[6] WKUPDEP_I2C3_IRQ_EVE1	Read/write
I2C3	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C3_WKDEP[7] WKUPDEP_I2C3_IRQ_EVE2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
I2C4	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C4_WKDEP[13] WKUPDEP_I2C4_DMA_SDMA	Read/write
I2C4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C4_WKDEP[4] WKUPDEP_I2C4_IRQ_IPU1	Read/write
I2C4	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C4_WKDEP[1] WKUPDEP_I2C4_IRQ_IPU2	Read/write
I2C4	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C4_WKDEP[0] WKUPDEP_I2C4_IRQ_MPU	Read/write
I2C4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C4_WKDEP[2] WKUPDEP_I2C4_IRQ_DSP1	Read/write
I2C4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C4_WKDEP[5] WKUPDEP_I2C4_IRQ_DSP2	Read/write
I2C4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C4_WKDEP[12] WKUPDEP_I2C4_DMA_DSP1	Read/write
I2C4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C4_WKDEP[15] WKUPDEP_I2C4_DMA_DSP2	Read/write
I2C4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C4_WKDEP[6] WKUPDEP_I2C4_IRQ_EVE1	Read/write
I2C4	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C4_WKDEP[7] WKUPDEP_I2C4_IRQ_EVE2	Read/write
McSPI1	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[3] WKUPDEP_MCSPI1_SDMA	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McSPI1	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[2] WKUPDEP_MCSPI1_DSP1	Read/write
McSPI1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[5] WKUPDEP_MCSPI1_DSP2	Read/write
McSPI1	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[4] WKUPDEP_MCSPI1_IPU1	Read/write
McSPI1	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[1] WKUPDEP_MCSPI1_IPU2	Read/write
McSPI1	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[0] WKUPDEP_MCSPI1_MPU	Read/write
McSPI1	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[6] WKUPDEP_MCSPI1_EVE1	Read/write
McSPI1	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[7] WKUPDEP_MCSPI1_EVE2	Read/write
McSPI2	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[3] WKUPDEP_MCSPI2_SDMA	Read/write
McSPI2	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[2] WKUPDEP_MCSPI2_DSP1	Read/write
McSPI2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[5] WKUPDEP_MCSPI2_DSP2	Read/write
McSPI2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[4] WKUPDEP_MCSPI2_IPU1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McSPI2	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[1] WKUPDEP_MCSPI2_IPU2	Read/write
McSPI2	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[0] WKUPDEP_MCSPI2_MPU	Read/write
McSPI2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[6] WKUPDEP_MCSPI2_EVE1	Read/write
McSPI2	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[7] WKUPDEP_MCSPI2_EVE2	Read/write
McSPI3	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[3] WKUPDEP_MCSPI3_SDMA	Read/write
McSPI3	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[2] WKUPDEP_MCSPI3_DSP1	Read/write
McSPI3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[5] WKUPDEP_MCSPI3_DSP2	Read/write
McSPI3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[4] WKUPDEP_MCSPI3_IPU1	Read/write
McSPI3	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[1] WKUPDEP_MCSPI3_IPU2	Read/write
McSPI3	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[0] WKUPDEP_MCSPI3_MPU	Read/write
McSPI3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[6] WKUPDEP_MCSPI3_EVE1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McSPI3	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[7] WKUPDEP_MCSPI3_EVE2	Read/write
McSPI4	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[3] WKUPDEP_MCSPI4_SDMA	Read/write
McSPI4	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[2] WKUPDEP_MCSPI4_DSP1	Read/write
McSPI4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[5] WKUPDEP_MCSPI4_DSP2	Read/write
McSPI4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[4] WKUPDEP_MCSPI4_IPU1	Read/write
McSPI4	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[1] WKUPDEP_MCSPI4_IPU2	Read/write
McSPI4	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[0] WKUPDEP_MCSPI4_MPU	Read/write
McSPI4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[6] WKUPDEP_MCSPI4_EVE1	Read/write
McSPI4	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[7] WKUPDEP_MCSPI4_EVE2	Read/write
MMC3	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[3] WKUPDEP_MMC3_SDMA	Read/write
MMC3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[2] WKUPDEP_MMC3_DSP1	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MMC3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[5] WKUPDEP_MMC3_DSP2	Read/write
MMC3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[4] WKUPDEP_MMC3_IPU1	Read/write
MMC3	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[1] WKUPDEP_MMC3_IPU2	Read/write
MMC3	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[0] WKUPDEP_MMC3_MPU	Read/write
MMC3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[6] WKUPDEP_MMC3_EVE1	Read/write
MMC3	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC3_WKDEP[7] WKUPDEP_MMC3_EVE2	Read/write
MMC4	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[3] WKUPDEP_MMC4_SDMA	Read/write
MMC4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[2] WKUPDEP_MMC4_DSP1	Read/write
MMC4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[5] WKUPDEP_MMC4_DSP2	Read/write
MMC4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[4] WKUPDEP_MMC4_IPU1	Read/write
MMC4	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[1] WKUPDEP_MMC4_IPU2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MMC4	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[0] WKUPDEP_MMC4_MPU	Read/write
MMC4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[6] WKUPDEP_MMC4_EVE1	Read/write
MMC4	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[7] WKUPDEP_MMC4_EVE2	Read/write
UART1	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[3] WKUPDEP_UART1_SDMA	Read/write
UART1	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[2] WKUPDEP_UART1_DSP1	Read/write
UART1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[5] WKUPDEP_UART1_DSP2	Read/write
UART1	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[4] WKUPDEP_UART1_IPU1	Read/write
UART1	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[1] WKUPDEP_UART1_IPU2	Read/write
UART1	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[0] WKUPDEP_UART1_MPU	Read/write
UART1	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[6] WKUPDEP_UART1_EVE1	Read/write
UART1	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[7] WKUPDEP_UART1_EVE2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
UART2	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[3] WKUPDEP_UART2_SDMA	Read/write
UART2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[2] WKUPDEP_UART2_DSP1	Read/write
UART2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[5] WKUPDEP_UART2_DSP2	Read/write
UART2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[4] WKUPDEP_UART2_IPU1	Read/write
UART2	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[1] WKUPDEP_UART2_IPU2	Read/write
UART2	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[0] WKUPDEP_UART2_MPU	Read/write
UART2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[6] WKUPDEP_UART2_EVE1	Read/write
UART2	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[7] WKUPDEP_UART2_EVE2	Read/write
UART3	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[3] WKUPDEP_UART3_SDMA	Read/write
UART3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[2] WKUPDEP_UART3_DSP1	Read/write
UART3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[5] WKUPDEP_UART3_DSP2	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
UART3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[4] WKUPDEP_UART3_IPU1	Read/write
UART3	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[1] WKUPDEP_UART3_IPU2	Read/write
UART3	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[0] WKUPDEP_UART3_MPU	Read/write
UART3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[6] WKUPDEP_UART3_EVE1	Read/write
UART3	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[7] WKUPDEP_UART3_EVE2	Read/write
UART4	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[3] WKUPDEP_UART4_SDMA	Read/write
UART4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[2] WKUPDEP_UART4_DSP1	Read/write
UART4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[5] WKUPDEP_UART4_DSP2	Read/write
UART4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[4] WKUPDEP_UART4_IPU1	Read/write
UART4	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[1] WKUPDEP_UART4_IPU2	Read/write
UART4	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[0] WKUPDEP_UART4_MPU	Read/write

Table 3-150. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
UART4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[6] WKUPDEP_UART4_EVE1	Read/write
UART4	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART4_WKDEP[7] WKUPDEP_UART4_EVE2	Read/write
UART5	CD_L4PER	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[3] WKUPDEP_UART5_SDMA	Read/write
UART5	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[2] WKUPDEP_UART5_DSP1	Read/write
UART5	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[5] WKUPDEP_UART5_DSP2	Read/write
UART5	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[4] WKUPDEP_UART5_IPU1	Read/write
UART5	CD_L4PER	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[1] WKUPDEP_UART5_IPU2	Read/write
UART5	CD_L4PER	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[0] WKUPDEP_UART5_MPU	Read/write
UART5	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[6] WKUPDEP_UART5_EVE1	Read/write
UART5	CD_L4PER	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART5_WKDEP[7] WKUPDEP_UART5_EVE2	Read/write

3.6.4.6.4 Clock Domain Module Attributes

Table 3-151 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-151. CD_L4PER1 Modules Clocks Association

Module	Clock	Clock Type
TIMER10	TIMER10_GFCLK	Functional

Table 3-151. CD_L4PER1 Modules Clocks Association (continued)

Module	Clock	Clock Type
	L4PER_L3_GICLK	Interface[(1)]
TIMER11	TIMER11_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
TIMER2	TIMER2_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
TIMER3	TIMER3_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
TIMER4	TIMER4_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
TIMER9	TIMER9_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
ELM	L4PER_L3_GICLK	Interface[(1)]
GPIO2	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
GPIO3	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
GPIO4	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
GPIO5	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
GPIO6	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
GPIO7	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
GPIO8	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[(1)]
HDQ1W	L4PER_L3_GICLK	Interface[(1)]
	PER_12M_GFCLK	Functional
I2C1	L4PER_L3_GICLK	Interface[(1)]
	PER_96M_GFCLK	Functional
I2C2	L4PER_L3_GICLK	Interface[(1)]
	PER_96M_GFCLK	Functional
I2C3	L4PER_L3_GICLK	Interface[(1)]
	PER_96M_GFCLK	Functional
I2C4	L4PER_L3_GICLK	Interface[(1)]
	PER_96M_GFCLK	Functional
L4_PER1 interconnect	L4PER_L3_GICLK	Interface[(1)]
McSPI1	L4PER_L3_GICLK	Interface[(1)]
	PER_48M_GFCLK	Functional
McSPI2	L4PER_L3_GICLK	Interface[(1)]
	PER_48M_GFCLK	Functional
McSPI3	L4PER_L3_GICLK	Interface[(1)]
	PER_48M_GFCLK	Functional
McSPI4	L4PER_L3_GICLK	Interface[(1)]
	PER_48M_GFCLK	Functional

Table 3-151. CD_L4PER1 Modules Clocks Association (continued)

Module	Clock	Clock Type
MMC3	L4PER_32K_GFCLK	Functional
	L4PER_L3_GICLK	Interface(1)
	MMC3_GFCLK	Functional
MMC4	L4PER_32K_GFCLK	Functional
	L4PER_L3_GICLK	Interface(1)
	MMC4_GFCLK	Functional
UART1	L4PER_L3_GICLK	Interface(1)
	UART1_GFCLK	Functional
UART2	L4PER_L3_GICLK	Interface(1)
	UART2_GFCLK	Functional
UART3	L4PER_L3_GICLK	Interface(1)
	UART3_GFCLK	Functional
UART4	L4PER_L3_GICLK	Interface(1)
	UART4_GFCLK	Functional
UART5	L4PER_L3_GICLK	Interface(1)
	UART5_GFCLK	Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4PER_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-152 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-152. CD_L4PER1 Modules Wake-Up Request

Module	Wake-Up Feature
TIMER10	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
TIMER11	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
TIMER2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
TIMER3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
TIMER4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
TIMER9	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
ELM	None
GPIO2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
GPIO3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
GPIO4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
GPIO5	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
GPIO6	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
GPIO7	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
GPIO8	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)

Table 3-152. CD_L4PER1 Modules Wake-Up Request (continued)

Module	Wake-Up Feature
HDQ1W	None
I2C1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
I2C2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
I2C3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
I2C4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
L4_PER1 interconnect	None
McSPI1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McSPI2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McSPI3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McSPI4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
MMC3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
MMC4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
UART1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
UART2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
UART3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
UART4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
UART5	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)

Table 3-153 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-153. CD_L4PER1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
TIMER10	Slave	CM_L4PER_TIMER10_CLKCTRL[17:16] IDLEST	Idle status
TIMER11	Slave	CM_L4PER_TIMER11_CLKCTRL[17:16] IDLEST	Idle status
TIMER2	Slave	CM_L4PER_TIMER2_CLKCTRL[17:16] IDLEST	Idle status
TIMER3	Slave	CM_L4PER_TIMER3_CLKCTRL[17:16] IDLEST	Idle status
TIMER4	Slave	CM_L4PER_TIMER4_CLKCTRL[17:16] IDLEST	Idle status
TIMER9	Slave	CM_L4PER_TIMER9_CLKCTRL[17:16] IDLEST	Idle status
ELM	Slave	CM_L4PER_ELM_CLKCTRL[17:16] IDLEST	Idle status

Table 3-153. CD_L4PER1 Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
GPIO2	Slave	CM_L4PER_GPIO2_CLKCTRL[17:16] IDLEST	Idle status
GPIO3	Slave	CM_L4PER_GPIO3_CLKCTRL[17:16] IDLEST	Idle status
GPIO4	Slave	CM_L4PER_GPIO4_CLKCTRL[17:16] IDLEST	Idle status
GPIO5	Slave	CM_L4PER_GPIO5_CLKCTRL[17:16] IDLEST	Idle status
GPIO6	Slave	CM_L4PER_GPIO6_CLKCTRL[17:16] IDLEST	Idle status
GPIO7	Slave	CM_L4PER_GPIO7_CLKCTRL[17:16] IDLEST	Idle status
GPIO8	Slave	CM_L4PER_GPIO8_CLKCTRL[17:16] IDLEST	Idle status
HDQ1W	Slave	CM_L4PER_HDQ1W_CLKCTRL[17:16] IDLEST	Idle status
I2C1	Slave	CM_L4PER_I2C1_CLKCTRL[17:16] IDLEST	Idle status
I2C2	Slave	CM_L4PER_I2C2_CLKCTRL[17:16] IDLEST	Idle status
I2C3	Slave	CM_L4PER_I2C3_CLKCTRL[17:16] IDLEST	Idle status
I2C4	Slave	CM_L4PER_I2C4_CLKCTRL[17:16] IDLEST	Idle status
L4_PER1 interconnect	Slave	CM_L4PER_L4_PER1_CLKCTRL[17:16] IDLEST	Idle status
McSPI1	Slave	CM_L4PER_MCSPI1_CLKCTRL[17:16] IDLEST	Idle status
McSPI2	Slave	CM_L4PER_MCSPI2_CLKCTRL[17:16] IDLEST	Idle status
McSPI3	Slave	CM_L4PER_MCSPI3_CLKCTRL[17:16] IDLEST	Idle status
McSPI4	Slave	CM_L4PER_MCSPI4_CLKCTRL[17:16] IDLEST	Idle status
MMC3	Slave	CM_L4PER_MMC3_CLKCTRL[17:16] IDLEST	Idle status
MMC4	Slave	CM_L4PER_MMC4_CLKCTRL[17:16] IDLEST	Idle status
UART1	Slave	CM_L4PER_UART1_CLKCTRL[17:16] IDLEST	Idle status
UART2	Slave	CM_L4PER_UART2_CLKCTRL[17:16] IDLEST	Idle status
UART3	Slave	CM_L4PER_UART3_CLKCTRL[17:16] IDLEST	Idle status
UART4	Slave	CM_L4PER_UART4_CLKCTRL[17:16] IDLEST	Idle status
UART5	Slave	CM_L4PER_UART5_CLKCTRL[17:16] IDLEST	Idle status

Table 3-154 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-154. CD_L4PER1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
TIMER10	Available	N/A	Available	CM_L4PER_TIMER10_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER11	Available	N/A	Available	CM_L4PER_TIMER11_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER2	Available	N/A	Available	CM_L4PER_TIMER2_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER3	Available	N/A	Available	CM_L4PER_TIMER3_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER4	Available	N/A	Available	CM_L4PER_TIMER4_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER9	Available	N/A	Available	CM_L4PER_TIMER9_CLKCTRL[1:0] MODULEMODE	Read/write
ELM	N/A	Available	N/A	CM_L4PER_ELM_CLKCTRL[1:0] MODULEMODE	Read only
GPIO2	Available	Available	N/A	CM_L4PER_GPIO2_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO3	Available	Available	N/A	CM_L4PER_GPIO3_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO4	Available	Available	N/A	CM_L4PER_GPIO4_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO5	Available	Available	N/A	CM_L4PER_GPIO5_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO6	Available	Available	N/A	CM_L4PER_GPIO6_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO7	Available	Available	N/A	CM_L4PER_GPIO7_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO8	Available	Available	N/A	CM_L4PER_GPIO8_CLKCTRL[1:0] MODULEMODE	Read/write
HDQ1W	Available	N/A	Available	CM_L4PER_HDQ1W_CLKCTRL[1:0] MODULEMODE	Read/write
I2C1	Available	N/A	Available	CM_L4PER_I2C1_CLKCTRL[1:0] MODULEMODE	Read/write
I2C2	Available	N/A	Available	CM_L4PER_I2C2_CLKCTRL[1:0] MODULEMODE	Read/write
I2C3	Available	N/A	Available	CM_L4PER_I2C3_CLKCTRL[1:0] MODULEMODE	Read/write
I2C4	Available	N/A	Available	CM_L4PER_I2C4_CLKCTRL[1:0] MODULEMODE	Read/write

Table 3-154. CD_L4PER1 Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L4_PER1 interconnect	N/A	Available	N/A	CM_L4PER_L4_PER1_CLKCTRL[1:0] MODULEMODE	Read only
McSPI1	Available	N/A	Available	CM_L4PER_MCSPI1_CLKCTRL[1:0] MODULEMODE	Read/write
McSPI2	Available	N/A	Available	CM_L4PER_MCSPI2_CLKCTRL[1:0] MODULEMODE	Read/write
McSPI3	Available	N/A	Available	CM_L4PER_MCSPI3_CLKCTRL[1:0] MODULEMODE	Read/write
McSPI4	Available	N/A	Available	CM_L4PER_MCSPI4_CLKCTRL[1:0] MODULEMODE	Read/write
MMC3	Available	N/A	Available	CM_L4PER_MMC3_CLKCTRL[1:0] MODULEMODE	Read/write
MMC4	Available	N/A	Available	CM_L4PER_MMC4_CLKCTRL[1:0] MODULEMODE	Read/write
UART1	Available	N/A	Available	CM_L4PER_UART1_CLKCTRL[1:0] MODULEMODE	Read/write
UART2	Available	N/A	Available	CM_L4PER_UART2_CLKCTRL[1:0] MODULEMODE	Read/write
UART3	Available	N/A	Available	CM_L4PER_UART3_CLKCTRL[1:0] MODULEMODE	Read/write
UART4	Available	N/A	Available	CM_L4PER_UART4_CLKCTRL[1:0] MODULEMODE	Read/write
UART5	Available	N/A	Available	CM_L4PER_UART5_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.7 CD_L4PER2 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.7.1 Overview

Figure 3-71 is an overview of the clock domain.

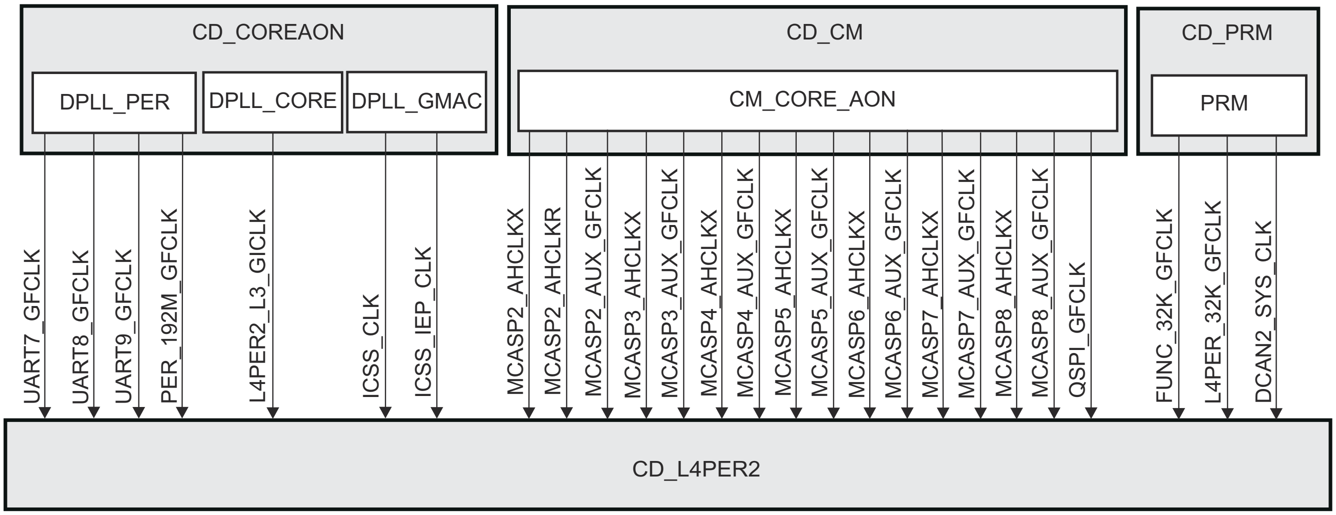


Figure 3-71. CD_L4PER2 Overview

Note

The PRU-ICSS1 and PRU-ICSS2 modules are not supported in this family of devices; therefore, the associated ICSS_CLK and ICSS_IEP_CLK clocks and related status bits are not functional.

3.6.4.7.2 Clock Domain Modes

Table 3-155 lists the clock domain modes supported by the clock domain.

Table 3-155. CD_L4PER2 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-156 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-156. CD_L4PER2 Control and Status Parameters

Parameter Name	Control/Status Bit Field
ICSS_CLK clock status	CM_L4PER2_CLKSTCTRL[8] CLKACTIVITY_ICSS_CLK
ICSS_IEP_CLK clock status	CM_L4PER2_CLKSTCTRL[14] CLKACTIVITY_ICSS_IEP_CLK
UART7_GFCLK clock status	CM_L4PER2_CLKSTCTRL[9] CLKACTIVITY_UART7_GFCLK
UART8_GFCLK clock status	CM_L4PER2_CLKSTCTRL[10] CLKACTIVITY_UART8_GFCLK
UART9_GFCLK clock status	CM_L4PER2_CLKSTCTRL[11] CLKACTIVITY_UART9_GFCLK
QSPI_GFCLK clock status	CM_L4PER2_CLKSTCTRL[12] CLKACTIVITY_QSPI_GFCLK
L4PER2_L3_GICLK clock status	CM_L4PER2_CLKSTCTRL[16] CLKACTIVITY_L4PER2_L3_GICLK
DCAN2_SYS_CLK clock status	CM_L4PER2_CLKSTCTRL[15] CLKACTIVITY_DCAN2_SYS_CLK
PER_192M_GFCLK clock status	CM_L4PER2_CLKSTCTRL[13] CLKACTIVITY_PER_192M_GFCLK
MCASP2_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[17] CLKACTIVITY_MCASP2_AHCLKX
MCASP2_AHCLKR clock status	CM_L4PER2_CLKSTCTRL[18] CLKACTIVITY_MCASP2_AHCLKR
MCASP2_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[19] CLKACTIVITY_MCASP2_AUX_GFCLK
MCASP3_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[20] CLKACTIVITY_MCASP3_AHCLKX
MCASP3_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[21] CLKACTIVITY_MCASP3_AUX_GFCLK
MCASP4_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[22] CLKACTIVITY_MCASP4_AHCLKX

Table 3-156. CD_L4PER2 Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
MCASP4_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[23] CLKACTIVITY_MCASP4_AUX_GFCLK
MCASP5_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[25] CLKACTIVITY_MCASP5_AHCLKX
MCASP5_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[24] CLKACTIVITY_MCASP5_AUX_GFCLK
MCASP6_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[26] CLKACTIVITY_MCASP6_AHCLKX
MCASP6_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[27] CLKACTIVITY_MCASP6_AUX_GFCLK
MCASP7_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[28] CLKACTIVITY_MCASP7_AHCLKX
MCASP7_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[29] CLKACTIVITY_MCASP7_AUX_GFCLK
MCASP8_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[30] CLKACTIVITY_MCASP8_AHCLKX
MCASP8_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[31] CLKACTIVITY_MCASP8_AUX_GFCLK
Clock Domain State Transition Control	CM_L4PER2_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.7.3 Clock Domain Dependency

CD_L4PER2 has no static dependency with any other clock domain of the device.

3.6.4.7.3.1 Dynamic Dependency

Table 3-157 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-157. CD_L4PER2 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4_CFG	Always enabled	CM_L4PER2_DYNAMICDEP[12] L4CFG_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4PER2_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_IPU	Always enabled	CM_L4PER2_DYNAMICDEP[3] IPU_DYNDEP	Read only
CD_ATL	Always enabled	CM_L4PER2_DYNAMICDEP[6] ATL_DYNDEP	Read only
CD_GMAC	Always enabled	CM_L4PER2_DYNAMICDEP[22] GMAC_DYNDEP	Read only

3.6.4.7.3.2 Wake-Up Dependency

Table 3-158 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
UART7	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[3] WKUPDEP_UART7_SDMA	Read/write
UART7	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[0] WKUPDEP_UART7_MPU	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
UART7	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[2] WKUPDEP_UART7_DSP1	Read/write
UART7	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[5] WKUPDEP_UART7_DSP2	Read/write
UART7	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[4] WKUPDEP_UART7_IPU1	Read/write
UART7	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[1] WKUPDEP_UART7_IPU2	Read/write
UART7	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[6] WKUPDEP_UART7_EVE1	Read/write
UART7	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_UART7_WKDEP[7] WKUPDEP_UART7_EVE2	Read/write
DCAN2	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[3] WKUPDEP_DCAN2_SDMA	Read/write
DCAN2	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[0] WKUPDEP_DCAN2_MPU	Read/write
DCAN2	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[2] WKUPDEP_DCAN2_DSP1	Read/write
DCAN2	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[5] WKUPDEP_DCAN2_DSP2	Read/write
DCAN2	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[4] WKUPDEP_DCAN2_IPU1	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DCAN2	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[1] WKUPDEP_DCAN2_IPU2	Read/write
DCAN2	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[6] WKUPDEP_DCAN2_EVE1	Read/write
DCAN2	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[7] WKUPDEP_DCAN2_EVE2	Read/write
QSPI	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[0] WKUPDEP_QSPI_MPU	Read/write
QSPI	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[2] WKUPDEP_QSPI_DSP1	Read/write
QSPI	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[5] WKUPDEP_QSPI_DSP2	Read/write
QSPI	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[4] WKUPDEP_QSPI_IPU1	Read/write
QSPI	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[1] WKUPDEP_QSPI_IPU2	Read/write
QSPI	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[6] WKUPDEP_QSPI_EVE1	Read/write
QSPI	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[7] WKUPDEP_QSPI_EVE2	Read/write
McASP2	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP2_WKDEP[13] WKUPDEP_MCASP2_DMA_SDMA	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McASP2	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP2_WKDEP[12] WKUPDEP_MCASP2_DMA_DSP1	Read/write
McASP2	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP2_WKDEP[15] WKUPDEP_MCASP2_DMA_DSP2	Read/write
McASP2	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP2_WKDEP[0] WKUPDEP_MCASP2_IRQ_MPU	Read/write
McASP2	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP2_WKDEP[2] WKUPDEP_MCASP2_IRQ_DSP1	Read/write
McASP2	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP2_WKDEP[5] WKUPDEP_MCASP2_IRQ_DSP2	Read/write
McASP2	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP2_WKDEP[4] WKUPDEP_MCASP2_IRQ_IPU1	Read/write
McASP2	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP2_WKDEP[1] WKUPDEP_MCASP2_IRQ_IPU2	Read/write
McASP2	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP2_WKDEP[6] WKUPDEP_MCASP2_IRQ_EVE1	Read/write
McASP2	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP2_WKDEP[7] WKUPDEP_MCASP2_IRQ_EVE2	Read/write
McASP3	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP3_WKDEP[13] WKUPDEP_MCASP3_DMA_SDMA	Read/write
McASP3	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP3_WKDEP[12] WKUPDEP_MCASP3_DMA_DSP1	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McASP3	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP3_WKDEP[15] WKUPDEP_MCASP3_DMA_DSP2	Read/write
McASP3	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP3_WKDEP[0] WKUPDEP_MCASP3_IRQ_MPU	Read/write
McASP3	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP3_WKDEP[2] WKUPDEP_MCASP3_IRQ_DSP1	Read/write
McASP3	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP3_WKDEP[5] WKUPDEP_MCASP3_IRQ_DSP2	Read/write
McASP3	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP3_WKDEP[4] WKUPDEP_MCASP3_IRQ_IPU1	Read/write
McASP3	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP3_WKDEP[1] WKUPDEP_MCASP3_IRQ_IPU2	Read/write
McASP3	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP3_WKDEP[6] WKUPDEP_MCASP3_IRQ_EVE1	Read/write
McASP3	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP3_WKDEP[7] WKUPDEP_MCASP3_IRQ_EVE2	Read/write
McASP4	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP4_WKDEP[13] WKUPDEP_MCASP4_DMA_SDMA	Read/write
McASP4	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP4_WKDEP[12] WKUPDEP_MCASP4_DMA_DSP1	Read/write
McASP4	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP4_WKDEP[15] WKUPDEP_MCASP4_DMA_DSP2	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McASP4	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[0] WKUPDEP_MCASP4_IRQ_MPU	Read/write
McASP4	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[2] WKUPDEP_MCASP4_IRQ_DSP1	Read/write
McASP4	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[5] WKUPDEP_MCASP4_IRQ_DSP2	Read/write
McASP4	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[4] WKUPDEP_MCASP4_IRQ_IPU1	Read/write
McASP4	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[1] WKUPDEP_MCASP4_IRQ_IPU2	Read/write
McASP4	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[6] WKUPDEP_MCASP4_IRQ_EVE1	Read/write
McASP4	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[7] WKUPDEP_MCASP4_IRQ_EVE2	Read/write
McASP5	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP5_WKDEP[13] WKUPDEP_MCASP5_DMA_SDMA	Read/write
McASP5	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP5_WKDEP[12] WKUPDEP_MCASP5_DMA_DSP1	Read/write
McASP5	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP5_WKDEP[15] WKUPDEP_MCASP5_DMA_DSP2	Read/write
McASP5	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[0] WKUPDEP_MCASP5_IRQ_MPU	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McASP5	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[2] WKUPDEP_MCASP5_IRQ_DSP1	Read/write
McASP5	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[5] WKUPDEP_MCASP5_IRQ_DSP2	Read/write
McASP5	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[4] WKUPDEP_MCASP5_IRQ_IPU1	Read/write
McASP5	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[1] WKUPDEP_MCASP5_IRQ_IPU2	Read/write
McASP5	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[6] WKUPDEP_MCASP5_IRQ_EVE1	Read/write
McASP5	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[7] WKUPDEP_MCASP5_IRQ_EVE2	Read/write
McASP6	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP6_WKDEP[13] WKUPDEP_MCASP6_DMA_SDMA	Read/write
McASP6	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP6_WKDEP[12] WKUPDEP_MCASP6_DMA_DSP1	Read/write
McASP6	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP6_WKDEP[15] WKUPDEP_MCASP6_DMA_DSP2	Read/write
McASP6	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP6_WKDEP[0] WKUPDEP_MCASP6_IRQ_MPU	Read/write
McASP6	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP6_WKDEP[2] WKUPDEP_MCASP6_IRQ_DSP1	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McASP6	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP6_WKDEP[5] WKUPDEP_MCASP6_IRQ_DSP2	Read/write
McASP6	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP6_WKDEP[4] WKUPDEP_MCASP6_IRQ_IPU1	Read/write
McASP6	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP6_WKDEP[1] WKUPDEP_MCASP6_IRQ_IPU2	Read/write
McASP6	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP6_WKDEP[6] WKUPDEP_MCASP6_IRQ_EVE1	Read/write
McASP6	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP6_WKDEP[7] WKUPDEP_MCASP6_IRQ_EVE2	Read/write
McASP7	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP7_WKDEP[13] WKUPDEP_MCASP7_DMA_SDMA	Read/write
McASP7	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP7_WKDEP[12] WKUPDEP_MCASP7_DMA_DSP1	Read/write
McASP7	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP7_WKDEP[15] WKUPDEP_MCASP7_DMA_DSP2	Read/write
McASP7	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP7_WKDEP[0] WKUPDEP_MCASP7_IRQ_MPU	Read/write
McASP7	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP7_WKDEP[2] WKUPDEP_MCASP7_IRQ_DSP1	Read/write
McASP7	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP7_WKDEP[5] WKUPDEP_MCASP7_IRQ_DSP2	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McASP7	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP7_WKDEP[4] WKUPDEP_MCASP7_IRQ_IPU1	Read/write
McASP7	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP7_WKDEP[1] WKUPDEP_MCASP7_IRQ_IPU2	Read/write
McASP7	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP7_WKDEP[6] WKUPDEP_MCASP7_IRQ_EVE1	Read/write
McASP7	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP7_WKDEP[7] WKUPDEP_MCASP7_IRQ_EVE2	Read/write
McASP8	CD_L4PER2	CD_SDMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP8_WKDEP[13] WKUPDEP_MCASP8_DMA_SDMA	Read/write
McASP8	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP8_WKDEP[12] WKUPDEP_MCASP8_DMA_DSP1	Read/write
McASP8	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP8_WKDEP[15] WKUPDEP_MCASP8_DMA_DSP2	Read/write
McASP8	CD_L4PER2	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP8_WKDEP[0] WKUPDEP_MCASP8_IRQ_MPU	Read/write
McASP8	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP8_WKDEP[2] WKUPDEP_MCASP8_IRQ_DSP1	Read/write
McASP8	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP8_WKDEP[5] WKUPDEP_MCASP8_IRQ_DSP2	Read/write
McASP8	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP8_WKDEP[4] WKUPDEP_MCASP8_IRQ_IPU1	Read/write

Table 3-158. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McASP8	CD_L4PER2	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP8_WKDEP[1] WKUPDEP_MCASP8_IRQ_IPU2	Read/write
McASP8	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP8_WKDEP[6] WKUPDEP_MCASP8_IRQ_EVE1	Read/write
McASP8	CD_L4PER2	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP8_WKDEP[7] WKUPDEP_MCASP8_IRQ_EVE2	Read/write

3.6.4.7.4 Clock Domain Module Attributes

Table 3-159 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-159. CD_L4PER2 Modules Clocks Association

Module	Clock	Clock Type
L4_PER2 interconnect	L4PER2_L3_GICLK	Interface(1)
DCAN2	L4PER2_L3_GICLK	Interface
	DCAN2_SYS_CLK	Functional
McASP2	L4PER2_L3_GICLK	Interface(1)
	MCASP2_AHCLKR	Functional
	MCASP2_AHCLKX	Functional
	MCASP2_AUX_GFCLK	Functional
McASP3	L4PER2_L3_GICLK	Interface(1)
	MCASP3_AHCLKX	Functional
	MCASP3_AUX_GFCLK	Functional
McASP4	L4PER2_L3_GICLK	Interface(1)
	MCASP4_AHCLKX	Functional
	MCASP4_AUX_GFCLK	Functional
McASP5	L4PER2_L3_GICLK	Interface(1)
	MCASP5_AHCLKX	Functional
	MCASP5_AUX_GFCLK	Functional
McASP6	L4PER2_L3_GICLK	Interface(1)
	MCASP6_AHCLKX	Functional
	MCASP6_AUX_GFCLK	Functional
McASP7	L4PER2_L3_GICLK	Interface(1)
	MCASP7_AHCLKX	Functional
	MCASP7_AUX_GFCLK	Functional
McASP8	L4PER2_L3_GICLK	Interface(1)
	MCASP8_AHCLKX	Functional
	MCASP8_AUX_GFCLK	Functional
QSPI	L4PER2_L3_GICLK	Interface
	QSPI_GFCLK	Functional

Table 3-159. CD_L4PER2 Modules Clocks Association (continued)

Module	Clock	Clock Type
PWMSS1	L4PER2_L3_GICLK	Interface(1) and Functional(2)
PWMSS2	L4PER2_L3_GICLK	Interface(1) and Functional(2)
PWMSS3	L4PER2_L3_GICLK	Interface(1) and Functional(2)
UART7	L4PER2_L3_GICLK	Interface(1)
	UART7_GFCLK	Functional
UART8	L4PER2_L3_GICLK	Interface(1)
	UART8_GFCLK	Functional
UART9	L4PER2_L3_GICLK	Interface(1)
	UART9_GFCLK	Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4PER2_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.
(2) The module's functional clock is equal to L4PER2_L3_GICLK/2.

Table 3-160 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-160. CD_L4PER2 Modules Wake-Up Request

Module	Wake-Up Feature
L4_PER2 interconnect	None
DCAN2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McASP2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McASP3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McASP4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McASP5	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McASP6	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McASP7	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
McASP8	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
QSPI	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
PWMSS1	None
PWMSS2	None
PWMSS3	None
UART7	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
UART8	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
UART9	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)

Table 3-161 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-161. CD_L4PER2 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L4_PER2 interconnect	Slave	CM_L4PER2_L4_PER2_CLKCTRL[17:16] IDLEST	Idle status
DCAN2	Slave	CM_L4PER2_DCAN2_CLKCTRL[17:16] IDLEST	Idle status
McASP2	Slave	CM_L4PER2_MCASP2_CLKCTRL[17:16] IDLEST	Idle status
McASP3	Slave	CM_L4PER2_MCASP3_CLKCTRL[17:16] IDLEST	Idle status
McASP4	Slave	CM_L4PER2_MCASP4_CLKCTRL[17:16] IDLEST	Idle status
McASP5	Slave	CM_L4PER2_MCASP5_CLKCTRL[17:16] IDLEST	Idle status
McASP6	Slave	CM_L4PER2_MCASP6_CLKCTRL[17:16] IDLEST	Idle status
McASP7	Slave	CM_L4PER2_MCASP7_CLKCTRL[17:16] IDLEST	Idle status
McASP8	Slave	CM_L4PER2_MCASP8_CLKCTRL[17:16] IDLEST	Idle status
QSPI	Slave	CM_L4PER2_QSPI_CLKCTRL[17:16] IDLEST	Idle status
PWMSS1	Slave	CM_L4PER2_PWMSS1_CLKCTRL[17:16] IDLEST	Idle status
PWMSS2	Slave	CM_L4PER2_PWMSS2_CLKCTRL[17:16] IDLEST	Idle status
PWMSS3	Slave	CM_L4PER2_PWMSS3_CLKCTRL[17:16] IDLEST	Idle status
UART7	Slave	CM_L4PER2_UART7_CLKCTRL[17:16] IDLEST	Idle status
UART8	Slave	CM_L4PER2_UART8_CLKCTRL[17:16] IDLEST	Idle status
UART9	Slave	CM_L4PER2_UART9_CLKCTRL[17:16] IDLEST	Idle status

Table 3-162 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-162. CD_L4PER2 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L4_PER2 interconnect	N/A	Available	N/A	CM_L4PER2_L4_PER2_CLKCTRL[1:0] MODULEMODE	Read only
DCAN2	Available	N/A	Available	CM_L4PER2_DCAN2_CLKCTRL[1:0] MODULEMODE	Read/write
McASP2	Available	N/A	Available	CM_L4PER2_MCASP2_CLKCTRL[1:0] MODULEMODE	Read/write
McASP3	Available	N/A	Available	CM_L4PER2_MCASP3_CLKCTRL[1:0] MODULEMODE	Read/write
McASP4	Available	N/A	Available	CM_L4PER2_MCASP4_CLKCTRL[1:0] MODULEMODE	Read/write

Table 3-162. CD_L4PER2 Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
McASP5	Available	N/A	Available	CM_L4PER2_MCASP5_CLKCTRL[1:0] MODULEMODE	Read/write
McASP6	Available	N/A	Available	CM_L4PER2_MCASP6_CLKCTRL[1:0] MODULEMODE	Read/write
McASP7	Available	N/A	Available	CM_L4PER2_MCASP7_CLKCTRL[1:0] MODULEMODE	Read/write
McASP8	Available	N/A	Available	CM_L4PER2_MCASP8_CLKCTRL[1:0] MODULEMODE	Read/write
QSPI	Available	N/A	Available	CM_L4PER2_QSPI_CLKCTRL[1:0] MODULEMODE	Read/write
PWMSS1	Available	N/A	Available	CM_L4PER2_PWMSS1_CLKCTRL[1:0] MODULEMODE	Read/write
PWMSS2	Available	N/A	Available	CM_L4PER2_PWMSS2_CLKCTRL[1:0] MODULEMODE	Read/write
PWMSS3	Available	N/A	Available	CM_L4PER2_PWMSS3_CLKCTRL[1:0] MODULEMODE	Read/write
UART7	Available	N/A	Available	CM_L4PER2_UART7_CLKCTRL[1:0] MODULEMODE	Read/write
UART8	Available	N/A	Available	CM_L4PER2_UART8_CLKCTRL[1:0] MODULEMODE	Read/write
UART9	Available	N/A	Available	CM_L4PER2_UART9_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.8 CD_L4PER3 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.8.1 Overview

Figure 3-72 is an overview of the clock domain.

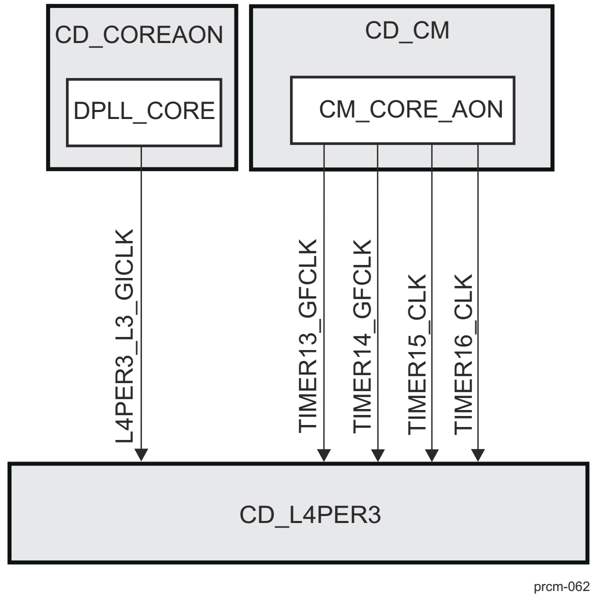


Figure 3-72. CD_L4PER3 Overview

3.6.4.8.2 Clock Domain Modes

Table 3-163 lists the clock domain modes supported by the clock domain.

Table 3-163. CD_L4PER3 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-164 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-164. CD_L4PER3 Control and Status Parameters

Parameter Name	Control/Status Bit Field
TIMER13_GFCLK clock status	CM_L4PER3_CLKSTCTRL[9] CLKACTIVITY_TIMER13_GFCLK
TIMER14_GFCLK clock status	CM_L4PER3_CLKSTCTRL[10] CLKACTIVITY_TIMER14_GFCLK
TIMER15_GFCLK clock status	CM_L4PER3_CLKSTCTRL[11] CLKACTIVITY_TIMER15_GFCLK
TIMER16_GFCLK clock status	CM_L4PER3_CLKSTCTRL[12] CLKACTIVITY_TIMER16_GFCLK
L4PER3_L3_GICLK clock status	CM_L4PER3_CLKSTCTRL[8] CLKACTIVITY_L4PER3_L3_GICLK
Clock Domain State Transition Control	CM_L4PER3_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.8.3 Clock Domain Dependency

CD_L4PER3 has no static dependency with any other clock domain of the device.

3.6.4.8.3.1 Dynamic Dependency

Table 3-165 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-165. CD_L4PER3 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4_CFG	Always enabled	CM_L4PER3_DYNAMICDEP[12] L4CFG_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4PER3_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_IPU	Always enabled	CM_L4PER3_DYNAMICDEP[3] IPU_DYNDEP	Read only

Table 3-165. CD_L4PER3 Dynamic Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3MAIN1	Always enabled	CM_L4PER3_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only
CD_RTC	Always enabled	CM_L4PER3_DYNAMICDEP[23] RTC_DYNDEP	Read only
CD_CAM	Always enabled	CM_L4PER3_DYNAMICDEP[9] CAM_DYNDEP	Read only
CD_VPE	Always enabled	CM_L4PER3_DYNAMICDEP[31] VPE_DYNDEP	Read only

3.6.4.8.3.2 Wake-Up Dependency

CD_L4PER3 has no module wake-up dependency with any other clock domain of the device.

3.6.4.8.4 Clock Domain Module Attributes

Table 3-166 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-166. CD_L4PER3 Modules Clocks Association

Module	Clock	Clock Type
L4_PER3 interconnect	L4PER3_L3_GICLK	Interface(1)
TIMER13	L4PER3_L3_GICLK	Interface(1)
	TIMER13_GFCLK	Functional
TIMER14	L4PER3_L3_GICLK	Interface(1)
	TIMER14_GFCLK	Functional
TIMER15	L4PER3_L3_GICLK	Interface(1)
	TIMER15_GFCLK	Functional
TIMER16	L4PER3_L3_GICLK	Interface(1)
	TIMER16_GFCLK	Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4PER3_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-167 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-167. CD_L4PER3 Modules Wake-Up Request

Module	Wake-Up Feature
L4_PER3 interconnect	None
TIMER13	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
TIMER14	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
TIMER15	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
TIMER16	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)

Table 3-168 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-168. CD_L4PER3 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L4_PER3 interconnect	Slave	CM_L4PER3_L4_PER3_CLKCTRL[17:16] IDLEST	Idle status

Table 3-168. CD_L4PER3 Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
TIMER13	Slave	CM_L4PER3_TIMER13_CLKCTRL[17:16] IDLEST	Idle status
TIMER14	Slave	CM_L4PER3_TIMER14_CLKCTRL[17:16] IDLEST	Idle status
TIMER15	Slave	CM_L4PER3_TIMER15_CLKCTRL[17:16] IDLEST	Idle status
TIMER16	Slave	CM_L4PER3_TIMER16_CLKCTRL[17:16] IDLEST	Idle status

Table 3-169 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-169. CD_L4PER3 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L4_PER3 interconnect	N/A	Available	N/A	CM_L4PER3_L4_PER3_CLKCTRL[1:0] MODULEMODE	Read only
TIMER13	Available	N/A	Available	CM_L4PER3_TIMER13_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER14	Available	N/A	Available	CM_L4PER3_TIMER14_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER15	Available	N/A	Available	CM_L4PER3_TIMER15_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER16	Available	N/A	Available	CM_L4PER3_TIMER16_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.9 CD_L4SEC Clock Domain

This section identifies the modes supported by the clock domain, the associated control, and status bits. It also identifies the dependencies of the domain with the other clock domains of the device.

3.6.4.9.1 Overview

Figure 3-73 shows an overview of the clock domain.

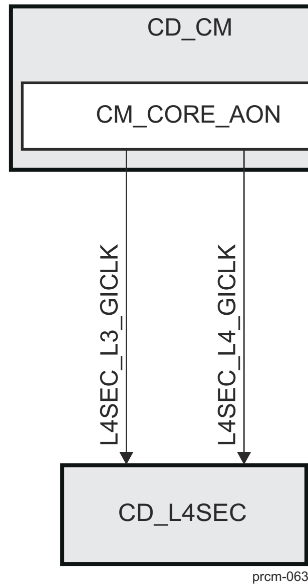


Figure 3-73. CD_L4SEC Overview

3.6.4.9.2 Clock Domain Modes

Table 3-170 lists the clock domain modes supported by the clock domain.

Table 3-170. CD_L4SEC Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-171 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-171. CD_L4SEC Control and Status Parameters

Parameter Name	Control/Status Bit Field
L4SEC_L3_GICLK clock status	CM_L4SEC_CLKSTCTRL[8] CLKACTIVITY_L4SEC_L3_GICLK
Clock Domain State Transition Control	CM_L4SEC_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.9.3 Clock Domain Dependency

CD_L4SEC has no module wake-up dependency with any other clock domain of the device.

3.6.4.9.3.1 Static Dependency

Table 3-172 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-172. CD_L4SEC Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4PER1	Disabled	CM_L4SEC_STATICDEP[13] L4PER_STATDEP	Read/Write
CD_L3_MAIN1	Always enabled	CM_L4SEC_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_L4SEC_STATICDEP[4] EMIF_STATDEP	Read/Write

3.6.4.9.3.2 Dynamic Dependency

Table 3-173 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-173. CD_L4SEC Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Disabled	CM_L4SEC_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.9.4 Clock Domain Module Attributes

Table 3-174 lists the clocks received by each module of the clock domain and the role (that is, functional clock or interface clock).

Table 3-174. CD_L4SEC Modules Clocks Association

Module	Clock	Clock Type
AES1	L4SEC_L3_GICLK	Interface
AES2	L4SEC_L3_GICLK	Interface
SHA2MD5_1	L4SEC_L3_GICLK	Interface
SHA2MD5_2	L4SEC_L3_GICLK	Interface
CryptoDMA	L4SEC_L3_GICLK	Interface(1) and Functional
DES3DES	L4SEC_L3_GICLK	Interface(1)
RNG	L4SEC_L3_GICLK	Interface(1)
FPKA	L4SEC_L3_GICLK	Interface(1) and Functional(2)

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4SEC_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.
(2) The module's functional clock is equal to L4SEC_L3_GICLK/2.

Table 3-175 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-175. CD_L4SEC Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
AES1	Slave	CM_L4SEC_AES1_CLKCTRL[17:16] IDLEST	Idle status
AES2	Slave	CM_L4SEC_AES2_CLKCTRL[17:16] IDLEST	Idle status
SHA2MD5_1	Slave	CM_L4SEC_SHA2MD51_CLKCTRL[17:16] IDLEST	Idle status
SHA2MD5_2	Slave	CM_L4SEC_SHA2MD52_CLKCTRL[17:16] IDLEST	Idle status
CryptoDMA	Master/Slave	CM_L4SEC_DMA_CRYPTO_CLKCTRL[18] STBYST	Standby status
		CM_L4SEC_DMA_CRYPTO_CLKCTRL [17:16] IDLEST	Idle status
DES3DES	Slave	CM_L4SEC_DES3DES_CLKCTRL[17:16] IDLEST	Idle status
RNG	Slave	CM_L4SEC_RNG_CLKCTRL[17:16] IDLEST	Idle status
FPKA	Slave	CM_L4SEC_FPKA_CLKCTRL[17:16] IDLEST	Idle status

3.6.4.9.5

Table 3-176 lists the supported slave clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-176. CD_L4SEC Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
AES1	Available	Available	N/A	CM_L4SEC_AES1_CLKCTRL[1:0] MODULEMODE	Read/Write
AES2	Available	Available	N/A	CM_L4SEC_AES2_CLKCTRL[1:0] MODULEMODE	Read/Write
SHA2MD5_1	Available	Available	N/A	CM_L4SEC_SHA2MD51_CLKCTRL[1:0] MODULEMODE	Read/Write
SHA2MD5_2	Available	Available	N/A	CM_L4SEC_SHA2MD52_CLKCTRL[1:0] MODULEMODE	Read/Write
DMA_CRYPTO	N/A	Available	N/A	CM_L4SEC_DMA_CRYPTO_CLKCTRL[1:0] MODULEMODE	Read only
DES3DES	Available	Available	N/A	CM_L4SEC_DES3DES_CLKCTRL[1:0] MODULEMODE	Read/Write
RNG	Available	Available	N/A	CM_L4SEC_RNG_CLKCTRL[1:0] MODULEMODE	Read/Write
FPKA	Available	N/A	Available	CM_L4SEC_FPKA_CLKCTRL[1:0] MODULEMODE	Read/Write

3.6.4.10 CD_L3INIT Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.10.1 Overview

Figure 3-74 is an overview of the clock domain.

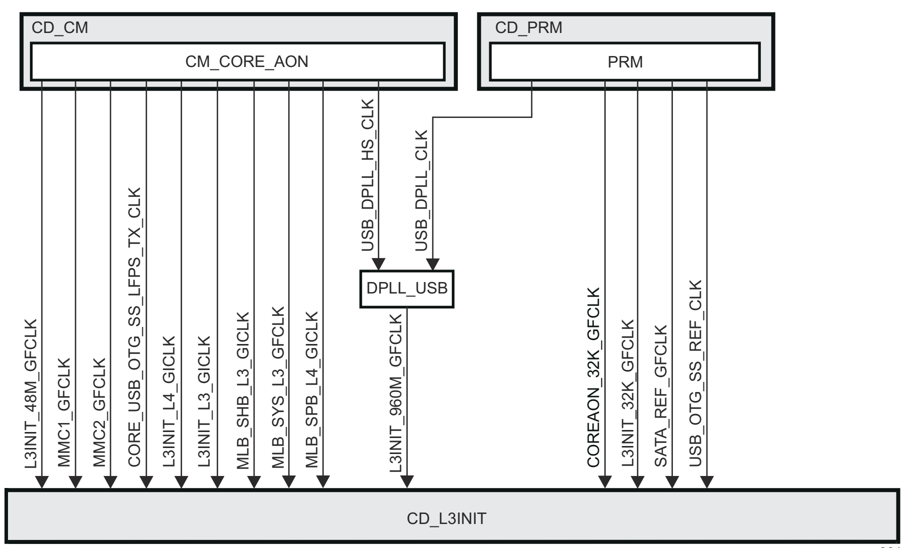


Figure 3-74. CD_L3INIT Overview

3.6.4.10.2 Clock Domain Modes

Table 3-177 lists the clock domain modes supported by the clock domain.

Table 3-177. CD_L3INIT Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-178 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-178. CD_L3INIT Control and Status Parameters

Parameter Name	Control/Status Bit Field
USB_OTG_SS_REF_CLK Clock Status	CM_L3INIT_CLKSTCTRL[20] CLKACTIVITY_USB_OTG_SS_REF_CLK
L3INIT_480M_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[21] CLKACTIVITY_L3INIT_480M_GFCLK
USB_DPLL_HS_CLK Clock Status	CM_L3INIT_CLKSTCTRL[13] CLKACTIVITY_USB_DPLL_HS_CLK
USB_DPLL_CLK Clock Status	CM_L3INIT_CLKSTCTRL[12] CLKACTIVITY_USB_DPLL_CLK
L3INIT_48M_GFCLK Clock Statuss	CM_L3INIT_CLKSTCTRL[11] CLKACTIVITY_L3INIT_48M_GFCLK
L3INIT_32K_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[23] CLKACTIVITY_L3INIT_32K_GFCLK
MMC1_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[15] CLKACTIVITY_MMC1_GFCLK
MMC1_32K_GFCLK Optional functional clock control	CM_L3INIT_MMC1_CLKCTRL[8] OPTFCLKEN_32K_CLK
MMC2_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[16] CLKACTIVITY_MMC2_GFCLK
MMC2_32K_GFCLK Optional functional clock control	CM_L3INIT_MMC2_CLKCTRL[8] OPTFCLKEN_32K_CLK
SATA_REF_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[24] CLKACTIVITY_SATA_REF_GFCLK
L3INIT_L3_GICLK Clock Status	CM_L3INIT_CLKSTCTRL[8] CLKACTIVITY_L3INIT_L3_GICLK
L3INIT_L4_GICLK Clock Status	CM_L3INIT_CLKSTCTRL[9] CLKACTIVITY_L3INIT_L4_GICLK
L3INIT_USB_LFPS_TX_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[10] CLKACTIVITY_L3INIT_USB_LFPS_TX_GFCLK
L3INIT_960M_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[22] CLKACTIVITY_L3INIT_960M_GFCLK
COREAON_32K_GFCLK Clock Status	CM_COREAON_CLKSTCTRL[12] CLKACTIVITY_COREAON_32K_GFCLK
COREAON_32K_GFCLK Clock Control	CM_COREAON_USB_PHY1_CORE_CLKCTRL[8] OPTFCLKEN_CLK32K
COREAON_32K_GFCLK Clock Control	CM_COREAON_USB_PHY2_CORE_CLKCTRL[8] OPTFCLKEN_CLK32K
COREAON_32K_GFCLK Clock Control	CM_COREAON_USB_PHY3_CORE_CLKCTRL[8] OPTFCLKEN_CLK32K
ABE_GICLK Clock Status	CM_COREAON_CLKSTCTRL[16] CLKACTIVITY_ABE_GICLK
Clock Domain State Transition Control	CM_COREAON_CLKSTCTRL[1:0] CLKTRCTRL
MLB_SHB_L3_GICLK Clock Status	CM_L3INIT_CLKSTCTRL[17] CLKACTIVITY_MLB_SHB_L3_GICLK
MLB_SPB_L4_GICLK Clock Status	CM_L3INIT_CLKSTCTRL[18] CLKACTIVITY_MLB_SPB_L4_GICLK
MLB_SYS_L3_GFCLK Clock Status	CM_L3INIT_CLKSTCTRL[19] CLKACTIVITY_MLB_SYS_L3_GFCLK
Clock Domain State Transition Control	CM_L3INIT_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.10.3 Clock Domain Dependency

3.6.4.10.3.1 Static Dependency

Table 3-179 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-179. CD_L3INIT Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IVA	Disabled	CM_L3INIT_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Always enabled	CM_L3INIT_STATICDEP[5] L3MAIN1_STATDEP	Read only

Table 3-179. CD_L3INIT Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4_CFG	Disabled	CM_L3INIT_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER	Disabled	CM_L3INIT_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4SEC	Disabled	CM_L3INIT_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_EMIF	Disabled	CM_L3INIT_STATICDEP[4] EMIF_STATDEP	Read/write
CD_WKUPAON	Disabled	CM_L3INIT_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_L4PER3	Disabled	CM_L3INIT_STATICDEP[27] L4PER3_STATDEP	Read/write

3.6.4.10.3.2 Dynamic Dependency

Table 3-180 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-180. CD_L3INIT Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always disabled	CM_L3INIT_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.10.3.3 Wake-Up Dependency

Table 3-181 lists the wake-up dependency settings for the modules of this clock in the clock domain.

Table 3-181. CD_L3INIT Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MMC1	CD_L3INIT	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[3] WKUPDEP_MMC1_SDMA	Read/write
MMC1	CD_L3INIT	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[2] WKUPDEP_MMC1_DSP1	Read/write
MMC1	CD_L3INIT	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[5] WKUPDEP_MMC1_DSP2	Read/write
MMC1	CD_L3INIT	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[4] WKUPDEP_MMC1_IPU1	Read/write
MMC1	CD_L3INIT	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[1] WKUPDEP_MMC1_IPU2	Read/write

Table 3-181. CD_L3INIT Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MMC1	CD_L3INIT	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[0] WKUPDEP_MMC1_MPU	Read/write
MMC1	CD_L3INIT	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[6] WKUPDEP_MMC1_EVE1	Read/write
MMC1	CD_L3INIT	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC1_WKDEP[7] WKUPDEP_MMC1_EVE2	Read/write
MMC2	CD_L3INIT	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[3] WKUPDEP_MMC2_SDMA	Read/write
MMC2	CD_L3INIT	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[2] WKUPDEP_MMC2_DSP1	Read/write
MMC2	CD_L3INIT	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[5] WKUPDEP_MMC2_DSP2	Read/write
MMC2	CD_L3INIT	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[4] WKUPDEP_MMC2_IPU1	Read/write
MMC2	CD_L3INIT	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[1] WKUPDEP_MMC2_IPU2	Read/write
MMC2	CD_L3INIT	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[0] WKUPDEP_MMC2_MPU	Read/write
MMC2	CD_L3INIT	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[6] WKUPDEP_MMC2_EVE1	Read/write
MMC2	CD_L3INIT	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_MMC2_WKDEP[7] WKUPDEP_MMC2_EVE2	Read/write

Table 3-181. CD_L3INIT Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
USB1	CD_L3INIT	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS1_WKDEP[4] WKUPDEP_USB_OTG_SS1_IPU1	Read/write
USB1	CD_L3INIT	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS1_WKDEP[1] WKUPDEP_USB_OTG_SS1_IPU2	Read/write
USB1	CD_L3INIT	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS1_WKDEP[2] WKUPDEP_USB_OTG_SS1_DSP1	Read/write
USB1	CD_L3INIT	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS1_WKDEP[5] WKUPDEP_USB_OTG_SS1_DSP2	Read/write
USB1	CD_L3INIT	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS1_WKDEP[0] WKUPDEP_USB_OTG_SS1_MPU	Read/write
USB1	CD_L3INIT	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS1_WKDEP[6] WKUPDEP_USB_OTG_SS1_EVE1	Read/write
USB1	CD_L3INIT	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS1_WKDEP[7] WKUPDEP_USB_OTG_SS1_EVE2	Read/write
USB2	CD_L3INIT	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS2_WKDEP[4] WKUPDEP_USB_OTG_SS2_IPU1	Read/write
USB2	CD_L3INIT	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS2_WKDEP[1] WKUPDEP_USB_OTG_SS2_IPU2	Read/write
USB2	CD_L3INIT	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS2_WKDEP[2] WKUPDEP_USB_OTG_SS2_DSP1	Read/write
USB2	CD_L3INIT	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS2_WKDEP[5] WKUPDEP_USB_OTG_SS2_DSP2	Read/write

Table 3-181. CD_L3INIT Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
USB2	CD_L3INIT	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS2_WKDEP[0] WKUPDEP_USB_OTG_SS2_MPU	Read/write
USB2	CD_L3INIT	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS2_WKDEP[6] WKUPDEP_USB_OTG_SS2_EVE1	Read/write
USB2	CD_L3INIT	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS2_WKDEP[7] WKUPDEP_USB_OTG_SS2_EVE2	Read/write
USB3	CD_L3INIT	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS3_WKDEP[4] WKUPDEP_USB_OTG_SS3_IPU1	Read/write
USB3	CD_L3INIT	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS3_WKDEP[1] WKUPDEP_USB_OTG_SS3_IPU2	Read/write
USB3	CD_L3INIT	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS3_WKDEP[2] WKUPDEP_USB_OTG_SS3_DSP1	Read/write
USB3	CD_L3INIT	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS3_WKDEP[5] WKUPDEP_USB_OTG_SS3_DSP2	Read/write
USB3	CD_L3INIT	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS3_WKDEP[0] WKUPDEP_USB_OTG_SS3_MPU	Read/write
USB3	CD_L3INIT	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS3_WKDEP[6] WKUPDEP_USB_OTG_SS3_EVE1	Read/write
USB3	CD_L3INIT	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS3_WKDEP[7] WKUPDEP_USB_OTG_SS3_EVE2	Read/write
USB4	CD_L3INIT	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS4_WKDEP[4] WKUPDEP_USB_OTG_SS4_IPU1	Read/write

Table 3-181. CD_L3INIT Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
USB4	CD_L3INIT	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS4_WKDEP[1] WKUPDEP_USB_OTG_SS4_IPU2	Read/write
USB4	CD_L3INIT	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS4_WKDEP[2] WKUPDEP_USB_OTG_SS4_DSP1	Read/write
USB4	CD_L3INIT	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS4_WKDEP[5] WKUPDEP_USB_OTG_SS4_DSP2	Read/write
USB4	CD_L3INIT	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS4_WKDEP[0] WKUPDEP_USB_OTG_SS4_MPU	Read/write
USB4	CD_L3INIT	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS4_WKDEP[6] WKUPDEP_USB_OTG_SS4_EVE1	Read/write
USB4	CD_L3INIT	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_USB_OTG_SS4_WKDEP[7] WKUPDEP_USB_OTG_SS4_EVE2	Read/write
SATA	CD_L3INIT	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_SATA_WKDEP[4] WKUPDEP_SATA_IPU1	Read/write
SATA	CD_L3INIT	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_SATA_WKDEP[1] WKUPDEP_SATA_IPU2	Read/write
SATA	CD_L3INIT	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_SATA_WKDEP[2] WKUPDEP_SATA_DSP1	Read/write
SATA	CD_L3INIT	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_SATA_WKDEP[5] WKUPDEP_SATA_DSP2	Read/write
SATA	CD_L3INIT	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_SATA_WKDEP[6] WKUPDEP_SATA_EVE1	Read/write

Table 3-181. CD_L3INIT Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
SATA	CD_L3INIT	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L3INIT_SATA_WKDEP[7] WKUPDEP_SATA_EVE2	Read/write

3.6.4.10.4 Clock Domain Module Attributes

Table 3-182 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-182. CD_L3INIT Modules Clocks Association

Module	Clock	Clock Type
MMC1	MMC1_GFCLK	Functional
	L3INIT_32K_GFCLK	Functional
	L3INIT_L3_GICLK	Interface(1)
MMC2	MMC2_GFCLK	Functional
	L3INIT_32K_GFCLK	Functional
	L3INIT_L3_GICLK	Interface(1)
USB1	L3INIT_960M_GFCLK	Functional
	L3INIT_L3_GICLK	Interface(1)
	USB_OTG_SS_REF_CLK	Reference clock for the DPLL_USB_OTG_SS (not managed by the PRCM module)
	USB_LFPS_TX_GFCLK	Interface
USB2	L3INIT_960M_GFCLK	Functional
	L3INIT_L3_GICLK	Interface(1)
	USB_OTG_SS_REF_CLK	Reference clock for the DPLL_USB_OTG_SS (not managed by the PRCM module)
USB3	L3INIT_L3_GICLK	Interface
	USB_OTG_SS_REF_CLK	Reference clock for the DPLL_USB_OTG_SS (not managed by the PRCM module)
USB4	L3INIT_L3_GICLK	Interface(1)
	USB_OTG_SS_REF_CLK	Reference clock for the DPLL_USB_OTG_SS (not managed by the PRCM module)
USB2PHY1	COREAON_32K_GFCLK	Functional
USB2PHY2	COREAON_32K_GFCLK	Functional
USB3_PHY	COREAON_32K_GFCLK	Functional
SATA	L3INIT_L3_GICLK	Interface
	L3INIT_48M_GFCLK	Functional
	SATA_REF_GFCLK	Functional
IEEE1500_2_OCP	L3INIT_L3_GICLK	Interface and functional
	L3INIT_L4_GICLK	Interface
OCP2SCP1	L3INIT_L4_GICLK	Interface
OCP2SCP3	L3INIT_L4_GICLK	Interface
MLB_SS	MLB_SHB_L3_GICLK	Interface
MLB_SS	MLB_SPB_L4_GICLK	Interface

Table 3-182. CD_L3INIT Modules Clocks Association (continued)

Module	Clock	Clock Type
MLB_SS	MLB_SYS_L3_GFCLK	Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4INIT_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-183 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-183. CD_L3INIT Modules Wake-Up Request

Module	Wake-Up Feature
MMC1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
MMC2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
OCP2SCP1, OCP2SCP3, GMAC	None
IEEE1500_2_OCP	Master wake-up request
USB1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
USB2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
USB3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
USB4	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
USB2PHY1	None
USB2PHY2	None
USB3_PHY	None
SATA	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)/ Master wake-up request
PCIe_SS1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)/ Master wake-up request
PCIe_SS2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)/ Master wake-up request
MLB_SS	Master wake-up request

Table 3-184 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-184. CD_L3INIT Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
IEEE1500_2_OCP	Slave/master	CM_L3INIT_IEEE1500_2_OCP_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_IEEE1500_2_OCP_CLKCTRL[17:16] IDLEST	Idle status
MMC1	Master/slave	CM_L3INIT_MMC1_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_MMC1_CLKCTRL[17:16] IDLEST	Idle status
MMC2	Master/slave	CM_L3INIT_MMC2_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_MMC2_CLKCTRL[17:16] IDLEST	Idle status
USB1	Slave/master	CM_L3INIT_USB_OTG_SS1_CLKCTRL[18] STBYST	Standby status

Table 3-184. CD_L3INIT Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
		CM_L3INIT_USB_OTG_SS1_CLKCTRL[17:16] IDLEST	Idle status
USB2	Slave/master	CM_L3INIT_USB_OTG_SS2_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_USB_OTG_SS2_CLKCTRL[17:16] IDLEST	Idle status
USB3	Slave/master	CM_L3INIT_USB_OTG_SS3_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_USB_OTG_SS3_CLKCTRL[17:16] IDLEST	Idle status
USB4	Slave/master	CM_L3INIT_USB_OTG_SS4_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_USB_OTG_SS4_CLKCTRL[17:16] IDLEST	Idle status
USB2PHY1	None	CM_COREAON_USB_PHY1_CORE_CLKCTRL[8] OPTFCLKEN_CLK32K	Optional functional clock control
USB2PHY2	None	CM_COREAON_USB_PHY2_CORE_CLKCTRL[8] OPTFCLKEN_CLK32K	Optional functional clock control
USB3_PHY	None	CM_COREAON_USB_PHY3_CORE_CLKCTRL[8] OPTFCLKEN_CLK32K	Optional functional clock control
SATA	Slave/master	CM_L3INIT_SATA_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_SATA_CLKCTRL[17:16] IDLEST	Idle status
OCP2SCP1	Slave	CM_L3INIT_OCP2SCP1_CLKCTRL[17:16] IDLEST	Standby status
OCP2SCP3	Slave	CM_L3INIT_OCP2SCP3_CLKCTRL[17:16] IDLEST	Standby status
MLB_SS		CM_L3INIT_MLB_SS_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_MLB_SS_CLKCTRL[17:16] IDLEST	Idle status

Table 3-185 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-185. CD_L3INIT Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
IEEE1500_2_OCP	N/A	Available	N/A	CM_L3INIT_IEEE1500_2_OCP_CLKCTRL[1:0] MODULEMODE	Read only
MMC1	Available	N/A	Available	CM_L3INIT_MMC1_CLKCTRL[1:0] MODULEMODE	Read/write
MMC2	Available	N/A	Available	CM_L3INIT_MMC2_CLKCTRL[1:0] MODULEMODE	Read/write
USB1	Available	Available	N/A	CM_L3INIT_USB_OTG_SS1_CLKCTRL[1:0] MODULEMODE	Read/write

Table 3-185. CD_L3INIT Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
USB2	Available	Available	N/A	CM_L3INIT_USB_OTG_SS2_CLKCTRL[1:0] MODULEMODE	Read/write
USB3	Available	Available	N/A	CM_L3INIT_USB_OTG_SS3_CLKCTRL[1:0] MODULEMODE	Read/write
USB4	Available	Available	N/A	CM_L3INIT_USB_OTG_SS4_CLKCTRL[1:0] MODULEMODE	Read/write
SATA	Available	N/A	Available	CM_L3INIT_SATA_CLKCTRL[1:0] MODULEMODE	Read/write
OCP2SCP1	Available	Available	N/A	CM_L3INIT_OCP2SCP1_CLKCTRL[1:0] MODULEMODE	Read/write
OCP2SCP3	Available	Available	N/A	CM_L3INIT_OCP2SCP3_CLKCTRL[1:0] MODULEMODE	Read/write
MLB_SS	Available	N/A	Available	CM_L3INIT_MLB_SS_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.11 CD_IVA Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.11.1 Overview

Figure 3-75 is an overview of the clock domain.

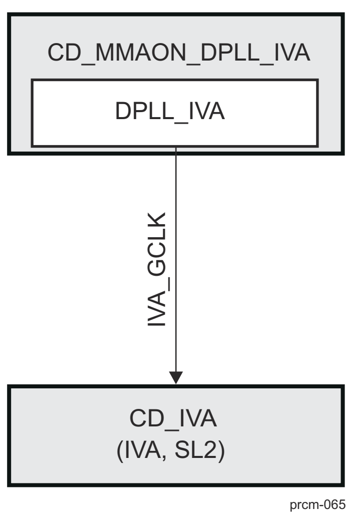


Figure 3-75. CD_IVA Overview

3.6.4.11.2 Clock Domain Modes

Table 3-186 lists the clock domain modes supported by the clock domain.

Table 3-186. CD_IVA Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-187 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-187. CD_IVA Control and Status Parameters

Parameter Name	Control/Status Bit Field
IVA_GCLK Clock Status	CM_IVA_CLKSTCTRL[8] CLKACTIVITY_IVA_GCLK
Clock Domain State Transition Control	CM_IVA_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.11.3 Clock Domain Dependency

CD_IVA has no module wake-up dependency with any other clock domain of the device.

3.6.4.11.3.1 Static Dependency

Table 3-188 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-188. CD_IVA Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_IVA_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_IVA_STATICDEP[4] EMIF_STATDEP	Read/write

3.6.4.11.3.2 Dynamic Dependency

Table 3-189 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-189. CD_IVA Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Disabled	CM_IVA_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.11.4 Clock Domain Module Attributes

Table 3-190 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-190. CD_IVA Modules Clocks Association

Module	Clock	Clock Type
IVA	IVA_GCLK	Interface and functional
SL2	IVA_GCLK	Interface

Table 3-191 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-191. CD_IVA Modules Wake-Up Request

Module	Wake-Up Feature
IVA	None
SL2	None

Table 3-192 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-192. CD_IVA Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
IVA	Master/Slave	CM_IVA_IVA_CLKCTRL[18] STBYST	Standby status
		CM_IVA_IVA_CLKCTRL[17:16] IDLEST	Idle status

Table 3-192. CD_IVA Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
SL2	Slave	CM_IVA_SL2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-193 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-193. CD_IVA Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
IVA	Available	Available	N/A	CM_IVA_IVA_CLKCTRL[1:0] MODULEMODE	Read/write
SL2	Available	Available	N/A	CM_IVA_SL2_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.12 CD_GPU Description

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.12.1 Overview

Figure 3-76 is an overview of the clock domain.

Figure 3-76. CD_GPU Overview

3.6.4.12.2 Clock Domain Modes

Table 3-194 lists the clock domain modes supported by the clock domain.

Table 3-194. CD_GPU Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-195 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-195. CD_GPU Control and Status Parameters

Parameter Name	Control/Status Bit Field
GPU_L3_GICLK clock status	CM_GPU_CLKSTCTRL[8] CLKACTIVITY_GPU_L3_GICLK
Select the source of HYD_CLK	CM_GPU_GPU_CLKCTRL[27:26] CLKSEL_HYD_CLK

Table 3-195. CD_GPU Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
GPU_CORE_GCLK clock status	CM_GPU_CLKSTCTRL[9] CLKACTIVITY_GPU_CORE_GCLK
Select the source of CORE_CLK	CM_GPU_GPU_CLKCTRL[25:24] CLKSEL_CORE_CLK
GPU_HYD_GCLK clock status	CM_GPU_CLKSTCTRL[10] CLKACTIVITY_GPU_HYD_GCLK
Clock Domain State Transition Control	CM_GPU_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.12.3 Clock Domain Dependency

CD_GPU has no module wake-up dependency with any other clock domain of the device.

3.6.4.12.3.1 Static Dependency

Table 3-196 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-196. CD_GPU Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IVA	Disabled	CM_GPU_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Always enabled	CM_GPU_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_GPU_STATICDEP[4] EMIF_STATDEP	Read/write

3.6.4.12.3.2 Dynamic Dependency

Table 3-197 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-197. CD_GPU Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Alway disabled	CM_GPU_DYNAMICDEP[6] L3MAIN1_DYNDEP	Read only

3.6.4.12.4 Clock Domain Module Attributes

Table 3-198 identifies for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-198. CD_GPU Modules Clocks Association

Module	Clock	Clock Type
GPU	GPU_L3_GICLK	Interface
	GPU_CORE_GCLK	Functional
	GPU_HYD_GCLK	Functional

Table 3-199 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-199. CD_GPU Modules Wake-Up Request

Module	Wake-Up Feature
GPU	None

Table 3-200 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-200. CD_GPU Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
GPU	Master/slave	CM_GPU_GPU_CLKCTRL[18] STBYST	Standby status

Table 3-200. CD_GPU Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
		CM_GPU_GPU_CLKCTRL[17:16] IDLEST	Idle status

Table 3-201 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-201. CD_GPU Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GPU	Available	N/A	Available	CM_GPU_GPU_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.13 CD_EMU Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.13.1 Overview

Figure 3-77 is an overview of the clock domain.

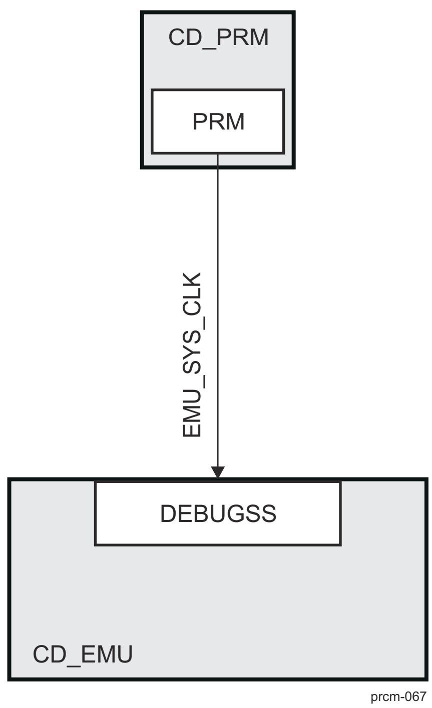


Figure 3-77. CD_EMU Overview

3.6.4.13.2 Clock Domain Modes

Table 3-202 lists the clock domain modes supported by the clock domain.

Table 3-202. CD_EMU Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Not available	Not available	Available	Available

Table 3-203 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-203. CD_EMU Control and Status Parameters

Parameter Name	Control/Status Bit Field
EMU_SYS_GCLK Clock Status	CM_EMU_CLKSTCTRL[8] CLKACTIVITY_EMU_SYS_CLK
Clock Domain State Transition Control	CM_EMU_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.13.3 Clock Domain Dependency

CD_EMU has no static or module wake-up dependency with any other clock domain of the device.

3.6.4.13.3.1 Dynamic Dependency

Table 3-204 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-204. CD_EMU Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_EMU_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.13.4 Clock Domain Module Attributes

Table 3-205 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-205. CD_EMU Modules Clocks Association

Module	Clock	Clock Type
DEBUGSS	EMU_SYS_CLK	Interface and functional

Table 3-206 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-206. CD_EMU Modules Wake-Up Request

Module	Wake-Up Feature
DEBUGSS	Master wake-up request

3.6.4.14 CD_DSS Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device. BB2D is a part of CD_DSS clock domain. BB2D_GFCLK is the functional clock for BB2D module. It is derived from H24 post divider of DPLL_CORE.

3.6.4.14.1 Overview

Figure 3-78 is an overview of the clock domain.

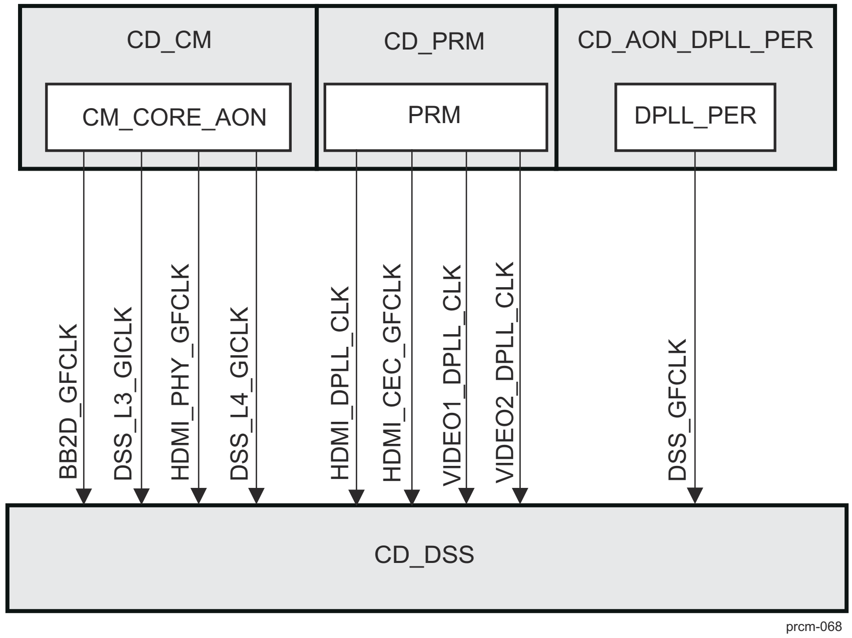


Figure 3-78. CD_DSS Overview

3.6.4.14.2 Clock Domain Modes

Table 3-207 lists the clock domain modes supported by the clock domain.

Table 3-207. CD_DSS Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-208 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-208. CD_DSS Control and Status Parameters

Parameter Name	Control/Status Bit Field
VIDEO1_DPLL_CLK Clock Status	CM_DSS_CLKSTCTRL[10] CLKACTIVITY_VIDEO1_DPLL_CLK
VIDEO1_CLK Clock Control	CM_DSS_DSS_CLKCTRL[12] OPTFCLKEN_VIDEO1_CLK
VIDEO2_DPLL_CLK Clock Status	CM_DSS_CLKSTCTRL[12] CLKACTIVITY_VIDEO2_DPLL_CLK
VIDEO2_CLK Clock Control	CM_DSS_DSS_CLKCTRL[13] OPTFCLKEN_VIDEO2_CLK
DSS_GFCLK Clock Status	CM_DSS_CLKSTCTRL[9] CLKACTIVITY_DSS_GFCLK
DSSCLK Clock Control	CM_DSS_DSS_CLKCTRL[8] OPTFCLKEN_DSSCLK
HDMI_DPLL_CLK Clock Status	CM_DSS_CLKSTCTRL[11] CLKACTIVITY_HDMI_DPLL_CLK
HDMI_CLK Clock Control	CM_DSS_DSS_CLKCTRL[10] OPTFCLKEN_HDMI_CLK
HDMI_CEC_GFCLK Clock Control	CM_DSS_DSS_CLKCTRL[11] OPTFCLKEN_32KHZ_CLK
HDMI_PHY_GFCLK Clock Control	CM_DSS_DSS_CLKCTRL[9] OPTFCLKEN_48MHZ_CLK
BB2D_GFCLK Clock Control	CM_DSS_CLKSTCTRL[13] CLKACTIVITY_BB2D_GFCLK
DSS_L3_GICLK Clock Status	CM_DSS_CLKSTCTRL[8] CLKACTIVITY_DSS_L3_GICLK
DSS_L4_GICLK Clock Status	CM_DSS_CLKSTCTRL[15] CLKACTIVITY_DSS_L4_GICLK
HDMI_CEC_GFCLK Clock Status	CM_DSS_CLKSTCTRL[17] CLKACTIVITY_HDMI_CEC_GFCLK
HDMI_PHY_GFCLK Clock Status	CM_DSS_CLKSTCTRL[18] CLKACTIVITY_HDMI_PHY_GFCLK
Clock Domain State Transition Control	CM_DSS_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.14.3 Clock Domain Dependency

3.6.4.14.3.1 Static Dependency

Table 3-209 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-209. CD_DSS Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IVA	Disabled	CM_DSS_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Always enabled	CM_DSS_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_DSS_STATICDEP[4] EMIF_STATDEP	Read/write

3.6.4.14.3.2 Dynamic Dependency

Table 3-210 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-210. CD_DSS Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Alway disabled	CM_DSS_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.14.3.3 Wake-Up Dependency

Table 3-211 lists the wake-up dependency settings for the modules of this clock in the clock domain

Table 3-211. CD_DSS Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS	CD_DSS	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[12] WKUPDEP_DSI1_A_DSP1	Read/write
DSS	CD_DSS	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[15] WKUPDEP_DSI1_A_DSP2	Read/write
DSS	CD_DSS	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[14] WKUPDEP_DSI1_A_IPU1	Read/write
DSS	CD_DSS	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[11] WKUPDEP_DSI1_A_IPU2	Read/write
DSS	CD_DSS	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[10] WKUPDEP_DSI1_A_MPU	Read/write
DSS	CD_DSS	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[13] WKUPDEP_DSI1_A_SDMA	Read/write

Table 3-211. CD_DSS Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS	CD_DSS	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[16] WKUPDEP_DSI1_A_EVE1	Read/write
DSS	CD_DSS	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[17] WKUPDEP_DSI1_A_EVE2	Read/write
DSS	CD_DSS	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[22] WKUPDEP_DSI1_B_DSP1	Read/write
DSS	CD_DSS	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[25] WKUPDEP_DSI1_B_DSP2	Read/write
DSS	CD_DSS	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[24] WKUPDEP_DSI1_B_IPU1	Read/write
DSS	CD_DSS	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[21] WKUPDEP_DSI1_B_IPU2	Read/write
DSS	CD_DSS	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[20] WKUPDEP_DSI1_B_MPU	Read/write
DSS	CD_DSS	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[23] WKUPDEP_DSI1_B_SDMA	Read/write
DSS-DSI1_B	CD_DSS	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[26] WKUPDEP_DSI1_B_EVE1	Read/write
DSS	CD_DSS	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[27] WKUPDEP_DSI1_B_EVE2	Read/write
DSS	CD_DSS	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[2] WKUPDEP_DISPC_DSP1	Read/write

Table 3-211. CD_DSS Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS	CD_DSS	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[5] WKUPDEP_DISPC_DSP2	Read/write
DSS	CD_DSS	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[4] WKUPDEP_DISPC_IPU1	Read/write
DSS	CD_DSS	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[1] WKUPDEP_DISPC_IPU2	Read/write
DSS	CD_DSS	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[0] WKUPDEP_DISPC_MPU	Read/write
DSS	CD_DSS	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[3] WKUPDEP_DISPC_SDMA	Read/write
DSS	CD_DSS	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[6] WKUPDEP_DISPC_EVE1	Read/write
DSS	CD_DSS	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[7] WKUPDEP_DISPC_EVE2	Read/write
DSS-HDMI	CD_DSS	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_DSS_DSS2_WKDEP[23] WKUPDEP_HDMIDMA_SDMA	Read/write
DSS-HDMI	CD_DSS	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_DSS_DSS2_WKDEP[22] WKUPDEP_HDMIDMA_DSP1	Read/write
DSS-HDMI	CD_DSS	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[25] WKUPDEP_HDMIDMA_DSP2	Read/write
DSS-HDMI	CD_DSS	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[4] WKUPDEP_HDMIIRQ_IPU1	Read/write

Table 3-211. CD_DSS Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS-HDMI	CD_DSS	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[1] WKUPDEP_HDMIIRQ_IPU2	Read/write
DSS-HDMI	CD_DSS	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[2] WKUPDEP_HDMIIRQ_DSP1	Read/write
DSS-HDMI	CD_DSS	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[5] WKUPDEP_HDMIIRQ_DSP2	Read/write
DSS-HDMI	CD_DSS	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[0] WKUPDEP_HDMIIRQ_MPU	Read/write
DSS-HDMI	CD_DSS	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[6] WKUPDEP_HDMIIRQ_EVE1	Read/write
DSS-HDMI	CD_DSS	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[7] WKUPDEP_HDMIIRQ_EVE2	Read/write
DSS	CD_DSS	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[12] WKUPDEP_DSI1_C_DSP1	Read/write
DSS	CD_DSS	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[15] WKUPDEP_DSI1_C_DSP2	Read/write
DSS	CD_DSS	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[14] WKUPDEP_DSI1_C_IPU1	Read/write
DSS	CD_DSS	CD_IPU2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[11] WKUPDEP_DSI1_C_IPU2	Read/write
DSS	CD_DSS	CD_MPU, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[10] WKUPDEP_DSI1_C_MPU	Read/write

Table 3-211. CD_DSS Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS	CD_DSS	CD_DMA, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[13] WKUPDEP_DSI1_C_SDMA	Read/write
DSS	CD_DSS	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[16] WKUPDEP_DSI1_C_EVE1	Read/write
DSS	CD_DSS	CD_EVE2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS2_WKDEP[17] WKUPDEP_DSI1_C_EVE2	Read/write

3.6.4.14.4 Clock Domain Module Attributes

Table 3-212 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-212. CD_DSS Modules Clocks Association

Module	Clock	Clock Type
DSS	HDMI_DPLL_CLK	Functional
	DSS_GFCLK	Functional
	HDMI_PHY_GFCLK	Functional
	HDMI_CEC_GFCLK	Functional
	VIDEO1_DPLL_CLK	Functional
	VIDEO2_DPLL_CLK	Functional
	DSS_L3_GICLK	Interface(1)
BB2D	BB2D_GFCLK	Functional
	DSS_L3_GICLK	Interface

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the DSS_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-213 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-213. CD_DSS Modules Wake-Up Request

Module	Wake-Up Feature
DSS	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)
BB2D	None

Table 3-214 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-214. CD_DSS Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DSS	Master/slave	CM_DSS_DSS_CLKCTRL[18] STBYST	Standby status
		CM_DSS_DSS_CLKCTRL[17:16] IDLEST	Idle status

Table 3-214. CD_DSS Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
BB2D	Master/slave	CM_DSS_BB2D_CLKCTRL[18] STBYST	Standby status
		CM_DSS_BB2D_CLKCTRL[17:16] IDLEST	Idle status

Table 3-215 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-215. CD_DSS Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSS	Available	N/A	Available	CM_DSS_DSS_CLKCTRL[1:0] MODULEMODE	Read/write
BB2D	Available	N/A	Available	CM_DSS_BB2D_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.15 CD_L4_CFG Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.15.1 Overview

Figure 3-79 is an overview of the clock domain.

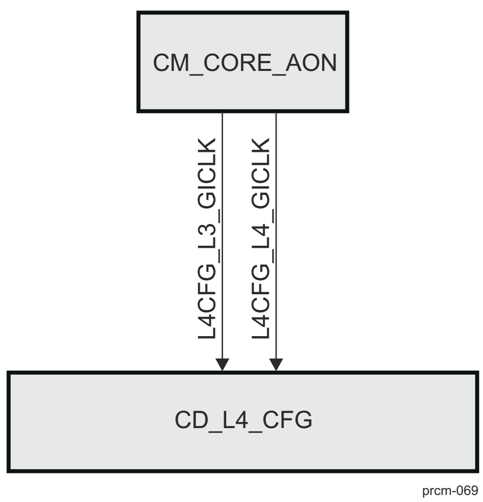


Figure 3-79. CD_L4_CFG Overview

3.6.4.15.2 Clock Domain Modes

Table 3-216 lists the clock domain modes supported by the clock domain.

Table 3-216. CD_L4_CFG Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-217 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-217. CD_L4_CFG Control and Status Parameters

Parameter Name	Control/Status Bit Field
L4CFG_L4_GICLK Clock Status	CM_L4CFG_CLKSTCTRL[8] CLKACTIVITY_L4CFG_L4_GICLK
L4CFG_L3_GICLK Clock Status	CM_L4CFG_CLKSTCTRL[9] CLKACTIVITY_L4CFG_L3_GICLK
Clock Domain State Transition Control	CM_L4CFG_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.15.3 Clock Domain Dependency

CD_L4_CFG has no static or module wake-up dependency with any other clock domain of the device.

3.6.4.15.3.1 Dynamic Dependency

Table 3-218 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-218. CD_L4_CFG Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CUSTEFUSE	Always enabled	CM_L4CFG_DYNAMICDEP[17] CUSTEFUSE_DYNDEP	Read only
CD_L3_MAIN1	Always enabled	CM_L4CFG_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4CFG_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_EMIF	Always enabled	CM_L4CFG_DYNAMICDEP[4] EMIF_DYNDEP	Read only
CD_DMA	Always enabled	CM_L4CFG_DYNAMICDEP[11] SDMA_DYNDEP	Read only
CD_MPU	Always enabled	CM_L4CFG_DYNAMICDEP[19] MPU_DYNDEP	Read only
CD_COREAON	Always enabled	CM_L4CFG_DYNAMICDEP[16] COREAON_DYNDEP	Read only

3.6.4.15.4 Clock Domain Module Attributes

Table 3-219 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-219. CD_L4_CFG Modules Clocks Association

Module	Clock	Clock Type
SPINLOCK	L4CFG_L3_GICLK	Interface(1)
L4_CFG interconnect	L4CFG_L3_GICLK	Interface(1)
MAILBOX(1 to 13)	L4CFG_L3_GICLK	Interface(1)
OCP2SCP2	L4CFG_L4_GICLK	Interface

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4CFG_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-220 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-220. CD_L4_CFG Modules Wake-Up Request

Module	Wake-Up Feature
SPINLOCK	None
L4_CFG interconnect	None
MAILBOX1	None
MAILBOX2	None
MAILBOX3	None
MAILBOX4	None
MAILBOX5	None

Table 3-220. CD_L4_CFG Modules Wake-Up Request (continued)

Module	Wake-Up Feature
MAILBOX6	None
MAILBOX7	None
MAILBOX8	None
MAILBOX9	None
MAILBOX10	None
MAILBOX11	None
MAILBOX12	None
MAILBOX13	None
OCP2SCP2	None

Table 3-221 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-221. CD_L4_CFG Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
SPINLOCK	Slave	CM_L4CFG_SPINLOCK_CLKCTRL[17:16] IDLEST	Idle status
L4_CFG interconnect	Slave	CM_L4CFG_L4_CFG_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX1	Slave	CM_L4CFG_MAILBOX1_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX2	Slave	CM_L4CFG_MAILBOX2_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX3	Slave	CM_L4CFG_MAILBOX3_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX4	Slave	CM_L4CFG_MAILBOX4_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX5	Slave	CM_L4CFG_MAILBOX5_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX6	Slave	CM_L4CFG_MAILBOX6_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX7	Slave	CM_L4CFG_MAILBOX7_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX8	Slave	CM_L4CFG_MAILBOX8_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX9	Slave	CM_L4CFG_MAILBOX9_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX10	Slave	CM_L4CFG_MAILBOX10_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX11	Slave	CM_L4CFG_MAILBOX11_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX12	Slave	CM_L4CFG_MAILBOX12_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX13	Slave	CM_L4CFG_MAILBOX13_CLKCTRL[17:16] IDLEST	Idle status
OCP2SCP2	Slave	CM_L4CFG_OCP2SCP2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-222 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-222. CD_L4_CFG Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
SPINLOCK	N/A	Available	N/A	CM_L4CFG_SPINLOCK_CLKCTRL[1:0] MODULEMODE	Read only
L4_CFG interconnect	N/A	Available	N/A	CM_L4CFG_L4_CFG_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX1	N/A	Available	N/A	CM_L4CFG_MAILBOX1_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX2	N/A	Available	N/A	CM_L4CFG_MAILBOX2_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX3	N/A	Available	N/A	CM_L4CFG_MAILBOX3_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX4	N/A	Available	N/A	CM_L4CFG_MAILBOX4_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX5	N/A	Available	N/A	CM_L4CFG_MAILBOX5_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX6	N/A	Available	N/A	CM_L4CFG_MAILBOX6_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX7	N/A	Available	N/A	CM_L4CFG_MAILBOX7_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX8	N/A	Available	N/A	CM_L4CFG_MAILBOX8_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX9	N/A	Available	N/A	CM_L4CFG_MAILBOX9_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX10	N/A	Available	N/A	CM_L4CFG_MAILBOX10_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX11	N/A	Available	N/A	CM_L4CFG_MAILBOX11_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX12	N/A	Available	N/A	CM_L4CFG_MAILBOX12_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX13	N/A	Available	N/A	CM_L4CFG_MAILBOX13_CLKCTRL[1:0] MODULEMODE	Read only
OCP2SCP2	N/A	Available	N/A	CM_L4CFG_OCP2SCP2_CLKCTRL[1:0] MODULEMODE	Read only

3.6.4.16 CD_L3_INSTR Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.16.1 Overview

Figure 3-80 is an overview of the clock domain.

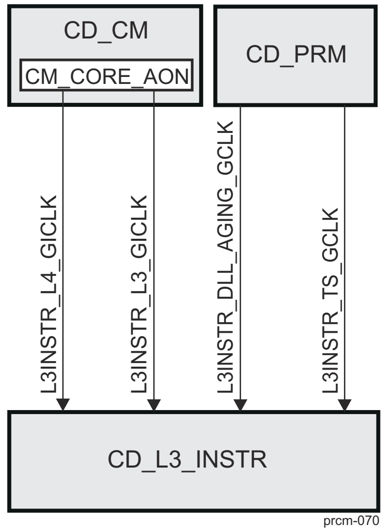


Figure 3-80. CD_L3_INSTR Overview

3.6.4.16.2 Clock Domain Modes

Table 3-223 lists the clock domain modes supported by the clock domain.

Table 3-223. CD_L3_INSTR Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Not available	Not available	Not available	Available

Table 3-224 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-224. CD_L3_INSTR Control and Status Parameters

Parameter Name	Control/Status Bit Field
L3INSTR_L3_GICLK Clock Status	CM_L3INSTR_CLKSTCTRL[8] CLKACTIVITY_L3INSTR_L3_GICLK
L3INSTR_DLL_AGING_GCLK Clock Status	CM_L3INSTR_CLKSTCTRL[9] CLKACTIVITY_L3INSTR_DLL_AGING_GCLK
L3INSTR_TS_GCLK Clock Status	CM_L3INSTR_CLKSTCTRL[10] CLKACTIVITY_L3INSTR_TS_GCLK
Clock Domain State Transition Control	CM_L3INSTR_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.16.3 Clock Domain Dependency

CD_L3_INSTR has no static, dynamic, or module wake-up dependency with any other clock domain of the device.

For aging, DLL requires a low-frequency clock that must run as long as the CORE power domain is on. To match this requirement, the DLL_AGING module has been created and instantiated in this clock domain, which is the last to transition in low-power state.

3.6.4.16.4 Clock Domain Module Attributes

Table 3-225 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-225. CD_L3_INSTR Modules Clocks Association

Module	Clock	Clock Type
L3_MAIN_2 interconnect	L3INSTR_L3_GICLK	Interface
	L3INSTR_L4_GICLK	Interface
L3_INSTR interconnect	L3INSTR_L3_GICLK	Interface

Table 3-225. CD_L3_INSTR Modules Clocks Association (continued)

Module	Clock	Clock Type
OCP_WP_NOC	L3INSTR_L3_GICLK	Interface
	L3INSTR_L4_GICLK	Interface
DLL_AGING	L3INSTR_DLL_AGING_GCLK	Interface
CTRL_MODULE_BANDGAP	L3INSTR_TS_GCLK	Interface

Table 3-226 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-226. CD_L3_INSTR Modules Wake-Up Request

Module	Wake-Up Feature
L3_MAIN_2 interconnect	None
L3_INSTR interconnect	None
OCP_WP_NOC	None
DLL_AGING	None
CTRL_MODULE_BANDGAP	None

Table 3-227 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-227. CD_L3_INSTR Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L3_MAIN_3 interconnect	Slave	CM_L3INSTR_L3_MAIN_2_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
L3_INSTR interconnect	Slave	CM_L3INSTR_L3_INSTR_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
OCP_WP_NOC	Slave	CM_L3INSTR_OCP_WP_NOC_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
DLL_AGING	Slave	CM_L3INSTR_DLL_AGING_CLKCTRL[17:16] IDLEST	Idle status
CTRL_MODULE_BANDGAP	Slave	CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL[17:16] IDLEST	Idle status

Table 3-228 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-228. CD_L3_INSTR Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L3_MAIN_2 interconnect	Available	Available	N/A	CM_L3INSTR_L3_MAIN_2_CLKCTRL[1:0] MODULEMODE	Read/write
L3_MAIN_2 interconnect	Available	Available	N/A	CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/write
L3_INSTR interconnect	Available	Available	N/A	CM_L3INSTR_L3_INSTR_CLKCTRL[1:0] MODULEMODE	Read/write

Table 3-228. CD_L3_INSTR Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L3_INSTR interconnect	Available	Available	N/A	CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/write
OCP_WP_NOC	Available	Available	N/A	CM_L3INSTR_OCP_WP_NOC_CLKCTRL[1:0] MODULEMODE	Read/write
OCP_WP_NOC	Available	Available	N/A	CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/write
DLL_AGING	N/A	Available	N/A	CM_L3INSTR_DLL_AGING_CLKCTRL[1:0] MODULEMODE	Read only
CTRL_MODULE_BANDGAP	N/A	Available	N/A	CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL[1:0] MODULEMODE	Read only

3.6.4.17 CD_L3_MAIN1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.17.1 Overview

Figure 3-81 is an overview of the clock domain.

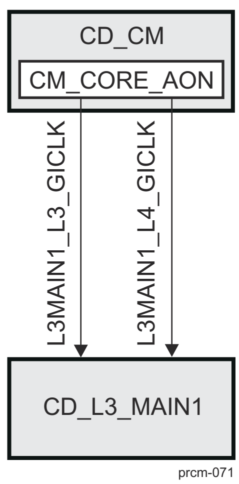


Figure 3-81. CD_L3_MAIN1 Overview

3.6.4.17.2 Clock Domain Modes

Table 3-229 lists the clock domain modes supported by the clock domain.

Table 3-229. CD_L3_MAIN1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-230 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-230. CD_L3_MAIN1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
L3MAIN1_L3_GICLK Clock Status	CM_L3MAIN1_CLKSTCTRL[8] CLKACTIVITY_L3MAIN1_L3_GICLK
L3MAIN1_L4_GICLK Clock Status	CM_L3MAIN1_CLKSTCTRL[9] CLKACTIVITY_L3MAIN1_L4_GICLK
Clock Domain State Transition Control	CM_L3MAIN1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.17.3 Clock Domain Dependency

CD_L3_MAIN1 has no static or module wake-up dependency with any other clock domain of the device.

3.6.4.17.3.1 Dynamic Dependency

Table 3-231 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-231. CD_L3_MAIN1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IPU	Always enabled	CM_L3MAIN1_DYNAMICDEP[3] IPU_DYNDEP	Read only
CD_IPU1	Always enabled	CM_L3MAIN1_DYNAMICDEP[18] IPU1_DYNDEP	Read only
CD_IPU2	Always enabled	CM_L3MAIN1_DYNAMICDEP[0] IPU2_DYNDEP	Read only
CD_DSP1	Always enabled	CM_L3MAIN1_DYNAMICDEP[1] DSP1_DYNDEP	Read only
CD_DSP2	Always enabled	CM_L3MAIN1_DYNAMICDEP[20] DSP2_DYNDEP	Read only
CD_IVA	Always enabled	CM_L3MAIN1_DYNAMICDEP[2] IVA_DYNDEP	Read only
CD_EMIF	Always enabled	CM_L3MAIN1_DYNAMICDEP[4] EMIF_DYNDEP	Read only
CD_DSS	Always enabled	CM_L3MAIN1_DYNAMICDEP[8] DSS_DYNDEP	Read only
CD_GPU	Always enabled	CM_L3MAIN1_DYNAMICDEP[10] GPU_DYNDEP	Read only
CD_L4PER	Always enabled	CM_L3MAIN1_DYNAMICDEP[13] L4PER_DYNDEP	Read only
CD_L4PER2	Always enabled	CM_L3MAIN1_DYNAMICDEP[22] L4PER2_DYNDEP	Read only
CD_L4PER3	Always enabled	CM_L3MAIN1_DYNAMICDEP[23] L4PER3_DYNDEP	Read only
CD_L4SEC	Always enabled	CM_L3MAIN1_DYNAMICDEP[14] L4SEC_DYNDEP	Read only
CD_L4_CFG	Always enabled	CM_L3MAIN1_DYNAMICDEP[12] L4CFG_DYNDEP	Read only
CD_PCIE	Always enabled	CM_L3MAIN1_DYNAMICDEP[21] PCIE_DYNDEP	Read only
CD_WKUPAON	Always enabled	CM_L3MAIN1_DYNAMICDEP[15] WKUPAON_DYNDEP	Read only
CD_EVE1	Always enabled	CM_L3MAIN1_DYNAMICDEP[28] EVE1_DYNDEP	Read only
CD_EVE2	Always enabled	CM_L3MAIN1_DYNAMICDEP[29] EVE2_DYNDEP	Read only

3.6.4.17.4 Clock Domain Module Attributes

Table 3-232 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-232. CD_L3_MAIN1 Modules Clocks Association

Module	Clock	Clock Type
L3_MAIN1 interconnect	L3MAIN1_L3_GICLK	Interface
	L3MAIN1_L4_GICLK	Interface
GPMC	L3MAIN1_L3_GICLK	Interface
OCMC_RAM1	L3MAIN1_L3_GICLK	Interface(1)
OCMC_RAM2	L3MAIN1_L3_GICLK	Interface(1)
OCMC_RAM3	L3MAIN1_L3_GICLK	Interface(1)
VCP1	L3MAIN1_L3_GICLK	Interface
VCP2	L3MAIN1_L3_GICLK	Interface
MMU_EDMA	L3MAIN1_L3_GICLK	Interface(1)
MMU_PCIESS	L3MAIN1_L3_GICLK	Interface(1)
EDMA_TPCC	L3MAIN1_L3_GICLK	Interface
EDMA_TC0	L3MAIN1_L3_GICLK	Interface
EDMA_TC1	L3MAIN1_L3_GICLK	Interface

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L3MAIN1_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-233 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-233. CD_L3_MAIN1 Modules Wake-Up Request

Module	Wake-Up Feature
L3_MAIN1 interconnect	None
GPMC	None
OCMC_RAM1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
OCMC_RAM2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
OCMC_RAM3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
VCP1	None
VCP2	None
MMU_EDMA	None
MMU_PCIESS	None
EDMA_TPCC	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)
EDMA_TC0	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)/ Master wake-up request
EDMA_TC1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)/ Master wake-up request

Table 3-234 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-234. CD_L3_MAIN1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L3_MAIN1 interconnect	Slave	CM_L3MAIN1_L3_MAIN_1_CLKCTRL[17:16] IDLEST	Idle status
GPMC	Slave	CM_L3MAIN1_GPMC_CLKCTRL [17:16] IDLEST	Idle status
OCMC_RAM1	Slave	CM_L3MAIN1_OCMC_RAM1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-234. CD_L3_MAIN1 Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
OCMC_RAM2	Slave	CM_L3MAIN1_OCMC_RAM2_CLKCTRL[17:16] IDLEST	Idle status
OCMC_RAM3	Slave	CM_L3MAIN1_OCMC_RAM3_CLKCTRL[17:16] IDLEST	Idle status
VCP1	Slave	CM_L3MAIN1_VCP1_CLKCTRL[17:16] IDLEST	Idle status
VCP2	Slave	CM_L3MAIN1_VCP2_CLKCTRL[17:16] IDLEST	Idle status
MMU_EDMA	Slave	CM_L3MAIN1_MMU_EDMA_CLKCTRL[17:16] IDLEST	Idle status
MMU_PCIESS	Slave	CM_L3MAIN1_MMU_PCIESS_CLKCTRL[17:16] IDLEST	Idle status
EDMA_TPCC	Slave	CM_L3MAIN1_TPCC_CLKCTRL[17:16] IDLEST	Idle status
EDMA_TC0	Master/slave	CM_L3MAIN1_TPTC1_CLKCTRL[18] STBYST	Standby status
		CM_L3MAIN1_TPTC1_CLKCTRL[17:16] IDLEST	Idle status
EDMA_TC1	Master/slave	CM_L3MAIN1_TPTC2_CLKCTRL[18] STBYST	Standby status
		CM_L3MAIN1_TPTC2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-235 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-235. CD_L3_MAIN1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L3_MAIN1 interconnect	N/A	Available	N/A	CM_L3MAIN1_L3_MAIN_1_CLKCTRL[1:0] MODULEMODE	Read only
GPMC	Available	Available	N/A	CM_L3MAIN1_GPMC_CLKCTRL [1:0] MODULEMODE	Read/write
OCMC_RAM1	N/A	Available	N/A	CM_L3MAIN1_OCMC_RAM1_CLKCTRL [1:0] MODULEMODE	Read only
OCMC_RAM2	N/A	Available	N/A	CM_L3MAIN1_OCMC_RAM2_CLKCTRL [1:0] MODULEMODE	Read only
OCMC_RAM3	N/A	Available	N/A	CM_L3MAIN1_OCMC_RAM3_CLKCTRL [1:0] MODULEMODE	Read only
VCP1	N/A	Available	N/A	CM_L3MAIN1_VCP1_CLKCTRL[1:0] MODULEMODE	Read only
VCP2	N/A	Available	N/A	CM_L3MAIN1_VCP2_CLKCTRL[1:0] MODULEMODE	Read only
MMU_EDMA	N/A	Available	N/A	CM_L3MAIN1_MMU_EDMA_CLKCTRL[1:0] MODULEMODE	Read only
MMU_PCIESS	N/A	Available	N/A	CM_L3MAIN1_MMU_PCIESS_CLKCTRL[1:0] MODULEMODE	Read only

Table 3-235. CD_L3_MAIN1 Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
EDMA_TPCC	N/A	Available	N/A	CM_L3MAIN1_TPCC_CLKCTRL[1:0] MODULEMODE	Read only
EDMA_TC0	Available	Available	N/A	CM_L3MAIN1_TPTC1_CLKCTRL[1:0] MODULEMODE	Read/write
EDMA_TC1	Available	Available	N/A	CM_L3MAIN1_TPTC2_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.18 CD_EMIF Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.18.1 Overview

Figure 3-82 is an overview of the clock domain.

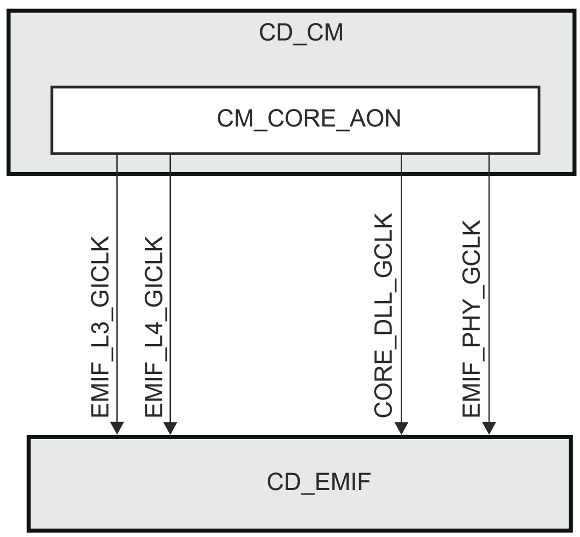


Figure 3-82. CD_EMIF Overview

3.6.4.18.2 Clock Domain Modes

Table 3-236 lists the clock domain modes supported by the clock domain.

Table 3-236. CD_EMIF Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-237 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-237. CD_EMIF Control and Status Parameters

Parameter Name	Control/Status Bit Field
CORE_DLL_GCLK Clock Status	CM_EMIF_CLKSTCTRL[9] CLKACTIVITY_EMIF_DLL_GCLK
CORE_DLL_GCLK Clock Control	CM_EMIF_EMIF_DLL_CLKCTRL[8] OPTFCLKEN_DLL_CLK
EMIF_L3_GICLK Clock Status	CM_EMIF_CLKSTCTRL[8] CLKACTIVITY_EMIF_L3_GICLK
EMIF_PHY_GCLK Clock Status	CM_EMIF_CLKSTCTRL[10] CLKACTIVITY_EMIF_PHY_GCLK

Table 3-237. CD_EMIF Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
Clock Domain State Transition Control	CM_EMIF_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.18.3 Clock Domain Dependency

CD_EMIF has no static, dynamic, or module wake-up dependency with any other clock domain of the device.

3.6.4.18.4 Clock Domain Module Attributes

Table 3-238 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-238. CD_EMIF Modules Clocks Association

Module	Clock	Clock Type
DLL	EMIF_DLL_GCLK	Functional
DMM	EMIF_L3_GICLK	Interface
EMIF1	EMIF_L3_GICLK	Interface
	MA_EOCP_GICLK(1)	Interface
	L3_EOCP_GICLK(1)	Interface
	EMIF_PHY_GCLK	Interface
EMIF2	EMIF_L3_GICLK	Interface
	MA_EOCP_GICLK(1)	Interface
	L3_EOCP_GICLK(1)	Interface
	EMIF_PHY_GCLK	Interface
EMIF_OCP_FW	EMIF_L3_GICLK	Interface
	EMIF_L4_GICLK	Interface

(1) EMIF modules are clocked by MA_EOCP_GICLK when MPU is active; otherwise, the L3_EOCP_GICLK clock is used.

Table 3-239 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-239. CD_EMIF Modules Wake-Up Request

Module	Wake-Up Feature
DLL	None
DMM	None
EMIF1	None
EMIF2	None
EMIF_OCP_FW	None

Table 3-240 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-240. CD_EMIF Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DMM	Slave	CM_EMIF_DMM_CLKCTRL[17:16] IDLEST	Idle status
EMIF1	Slave	CM_EMIF_EMIF1_CLKCTRL[17:16] IDLEST	Idle status
EMIF2	Slave	CM_EMIF_EMIF2_CLKCTRL[17:16] IDLEST	Idle status
EMIF_OCP_FW	Slave	CM_EMIF_EMIF_OCP_FW_CLKCTRL[17:16] IDLEST	Idle status

Table 3-241 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-241. CD_EMIF Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DMM	N/A	Available	N/A	CM_EMIF_DMM_CLKCTRL[1:0] MODULEMODE	Read only
EMIF1	Available	Available	N/A	CM_EMIF_EMIF1_CLKCTRL[1:0] MODULEMODE	Read/write
EMIF2	Available	Available	N/A	CM_EMIF_EMIF2_CLKCTRL[1:0] MODULEMODE	Read/write
EMIF_OCP_FW	N/A	Available	N/A	CM_EMIF_EMIF_OCP_FW_CLKCTRL[1:0] MODULEMODE	Read only

3.6.4.19 CD_IPU Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.19.1 Overview

Figure 3-83 is an overview of the clock domain.

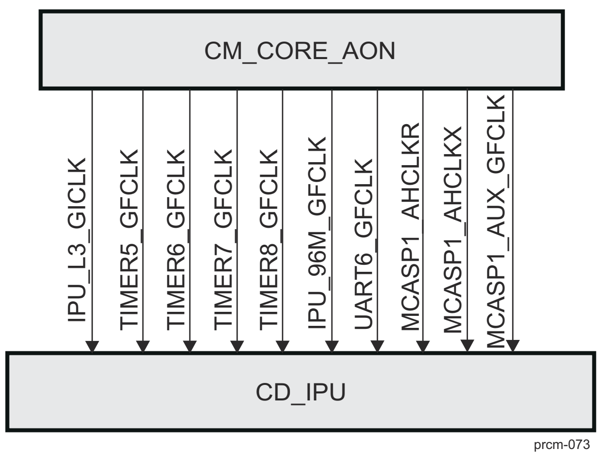


Figure 3-83. CD_IPU Overview

3.6.4.19.2 Clock Domain Modes

Table 3-242 lists the clock domain modes supported by the clock domain.

Table 3-242. CD_IPU Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-243 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-243. CD_IPU Control and Status Parameters

Parameter Name	Control/Status Bit Field
IPU_L3_GICLK Clock Status	CM_IPU_CLKSTCTRL[8] CLKACTIVITY_IPU_L3_GICLK
IPU_96M_GFCLK Clock Status	CM_IPU_CLKSTCTRL[13] CLKACTIVITY_IPU_96M_GFCLK

Table 3-243. CD_IPU Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
UART6_GFCLK Clock Status	CM_IPU_CLKSTCTRL[14] CLKACTIVITY_UART6_GFCLK
TIMER5_GFCLK Clock Status	CM_IPU_CLKSTCTRL[9] CLKACTIVITY_TIMER5_GFCLK
TIMER6_GFCLK Clock Status	CM_IPU_CLKSTCTRL[10] CLKACTIVITY_TIMER6_GFCLK
TIMER7_GFCLK Clock Status	CM_IPU_CLKSTCTRL[11] CLKACTIVITY_TIMER7_GFCLK
TIMER8_GFCLK Clock Status	CM_IPU_CLKSTCTRL[12] CLKACTIVITY_TIMER8_GFCLK
MCASP1_AHCLKR Clock Status	CM_IPU_CLKSTCTRL[18] CLKACTIVITY_MCASP1_AHCLKR
MCASP1_AHCLKX Clock Status	CM_IPU_CLKSTCTRL[17] CLKACTIVITY_MCASP1_AHCLKX
MCASP1_AUX_GFCLK Clock Status	CM_IPU_CLKSTCTRL[16] CLKACTIVITY_MCASP1_AUX_GFCLK
Clock Domain State Transition Control	CM_IPU_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.19.3 Clock Domain Dependency

CD_IPU has no module wake-up dependency with any other clock domain of the device.

3.6.4.19.3.1 Static Dependency

Table 3-244 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-244. CD_IPU Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_DSS	Disabled	CM_IPU1_STATICDEP[8] DSS_STATDEP	Read/write
CD_GPU	Disabled	CM_IPU1_STATICDEP[10] GPU_STATDEP	Read/write
CD_CAM	Alway disabled	CM_IPU1_STATICDEP[9] CAM_STATDEP	Read only
CD_IVA	Disabled	CM_IPU1_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Enabled	CM_IPU1_STATICDEP[5] L3MAIN1_STATDEP	Read/write
CD_L3INIT	Disabled	CM_IPU1_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Enabled	CM_IPU1_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER1	Disabled	CM_IPU1_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_IPU1_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_IPU1_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_L4SEC	Disabled	CM_IPU1_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_EMIF	Enabled	CM_IPU1_STATICDEP[4] EMIF_STATDEP	Read/write
CD_DMA	Alway disabled	CM_IPU1_STATICDEP[11] SDMA_STATDEP	Read only
CD_IPU	Disabled	CM_IPU1_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU2	Disabled	CM_IPU1_STATICDEP[0] IPU2_STATDEP	Read/write
CD_DSP	Disabled	CM_IPU1_STATICDEP[1] DSP1_STATDEP	Read/write
CD_DSP2	Disabled	CM_IPU1_STATICDEP[18] DSP2_STATDEP	Read/write

Table 3-244. CD_IPU Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_WKUPAON	Enabled	CM_IPU1_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Alway disabled	CM_IPU1_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Alway disabled	CM_IPU1_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_GMAC	Disabled	CM_IPU1_STATICDEP[25] GMAC_STATDEP	Read/write
CD_VPE	Enabled	CM_IPU1_STATICDEP[28] VPE_STATDEP	Read/write
CD_PCIE	Enabled	CM_IPU1_STATICDEP[29] PCIE_STATDEP	Read/write
CD_ATL	Enabled	CM_IPU1_STATICDEP[30] ATL_STATDEP	Read/write
CD_EVE1	Disabled	CM_IPU1_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_IPU1_STATICDEP[20] EVE2_STATDEP	Read/write

3.6.4.19.3.2 Dynamic Dependency

Table 3-245 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-245. CD_IPU Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_IPU1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.19.4 Clock Domain Module Attributes

Table 3-246 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-246. CD_IPU Modules Clocks Association

Module	Clock	Clock Type
TIMER5	IPU_L3_GICLK	Interface[(1)]
	TIMER5_GFCLK	Functional
TIMER6	IPU_L3_GICLK	Interface[(1)]
	TIMER6_GFCLK	Functional
TIMER7	IPU_L3_GICLK	Interface[(1)]
	TIMER7_GFCLK	Functional
TIMER8	IPU_L3_GICLK	Interface[(1)]
	TIMER8_GFCLK	Functional
UART6	IPU_L3_GICLK	Interface[(1)]
	UART6_GFCLK	Functional
I2C5	IPU_L3_GICLK	Interface[(1)]
	IPU_96M_GFCLK	Functional
McASP1	IPU_L3_GICLK	Interface[(1)]
	MCASP1_AHCLKR	Functional
	MCASP1_AHCLKX	Functional

Table 3-246. CD_IPU Modules Clocks Association (continued)

Module	Clock	Clock Type
	MCASP1_AUX_GFCLK	Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the IPU_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-247 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-247. CD_IPU Modules Wake-Up Request

Module	Wake-Up Feature
TIMER5	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ,DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
TIMER6	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
TIMER7	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
TIMER8	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ,)
UART6	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, , DMA_SYSTEM-DMA)
I2C5	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, , DMA_SYSTEM-DMA)
McASP1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, , DMA_SYSTEM-DMA)

Table 3-248 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-248. CD_IPU Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
TIMER5	Slave	CM_IPU_TIMER5_CLKCTRL[17:16] IDLEST	Idle status
TIMER6	Slave	CM_IPU_TIMER6_CLKCTRL[17:16] IDLEST	Idle status
TIMER7	Slave	CM_IPU_TIMER7_CLKCTRL[17:16] IDLEST	Idle status
TIMER8	Slave	CM_IPU_TIMER8_CLKCTRL[17:16] IDLEST	Idle status
UART6	Slave	CM_IPU_UART6_CLKCTRL[17:16] IDLEST	Idle status
I2C5	Slave	CM_IPU_I2C5_CLKCTRL[17:16] IDLEST	Idle status
McASP1	Slave	CM_IPU_MCASP1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-249 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-249. CD_IPU Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
McASP1	Available	N/A	Available	CM_IPU_MCASP1_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER5	Available	N/A	Available	CM_IPU_TIMER5_CLKCTRL[1:0] MODULEMODE	Read/write

Table 3-249. CD_IPU Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
TIMER6	Available	N/A	Available	CM_IPU_TIMER6_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER7	Available	N/A	Available	CM_IPU_TIMER7_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER8	Available	N/A	Available	CM_IPU_TIMER8_CLKCTRL[1:0] MODULEMODE	Read/write
I2C5	Available	N/A	Available	CM_IPU_I2C5_CLKCTRL[1:0] MODULEMODE	Read/write
UART6	Available	N/A	Available	CM_IPU_UART6_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.20 CD_IPU1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.20.1 Overview

Figure 3-84 is an overview of the clock domain.

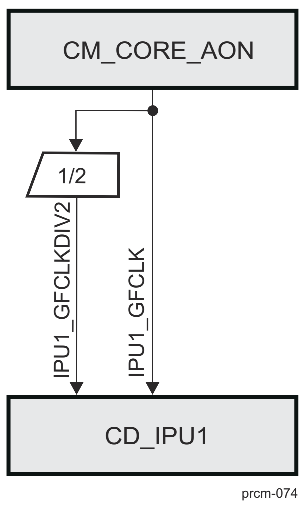


Figure 3-84. CD_IPU1 Overview

3.6.4.20.2 Clock Domain Modes

Table 3-250 lists the clock domain modes supported by the clock domain.

Table 3-250. CD_IPU1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-251 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-251. CD_IPU1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
IPU1_GFCLK Clock Status	CM_IPU1_CLKSTCTRL[8] CLKACTIVITY_IPU1_GFCLK

Table 3-251. CD_IPU1 Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
Clock Domain State Transition Control	CM_IPU1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.20.3 Clock Domain Dependency

CD_IPU1 has no module wake-up dependency with any other clock domain of the device.

3.6.4.20.3.1 Static Dependency

Table 3-252 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-252. CD_IPU1 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_DSS	Disabled	CM_IPU1_STATICDEP[8] DSS_STATDEP	Read/write
CD_GPU	Disabled	CM_IPU1_STATICDEP[10] GPU_STATDEP	Read/write
CD_CAM	Alway disabled	CM_IPU1_STATICDEP[9] CAM_STATDEP	Read only
CD_IVA	Disabled	CM_IPU1_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Enabled	CM_IPU1_STATICDEP[5] L3MAIN1_STATDEP	Read/write
CD_L3INIT	Disabled	CM_IPU1_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Enabled	CM_IPU1_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER1	Disabled	CM_IPU1_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_IPU1_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_IPU1_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_L4SEC	Disabled	CM_IPU1_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_EMIF	Enabled	CM_IPU1_STATICDEP[4] EMIF_STATDEP	Read/write
CD_DMA	Alway disabled	CM_IPU1_STATICDEP[11] SDMA_STATDEP	Read only
CD_IPU	Disabled	CM_IPU1_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU2	Disabled	CM_IPU1_STATICDEP[0] IPU2_STATDEP	Read/write
CD_DSP	Disabled	CM_IPU1_STATICDEP[1] DSP1_STATDEP	Read/write
CD_DSP2	Disabled	CM_IPU1_STATICDEP[18] DSP2_STATDEP	Read/write
CD_WKUPAON	Enabled	CM_IPU1_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Alway disabled	CM_IPU1_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Alway disabled	CM_IPU1_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_GMAC	Disabled	CM_IPU1_STATICDEP[25] GMAC_STATDEP	Read/write

Table 3-252. CD_IPU1 Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_VPE	Enabled	CM_IPU1_STATICDEP[28] VPE_STATDEP	Read/write
CD_PCIE	Enabled	CM_IPU1_STATICDEP[29] PCIE_STATDEP	Read/write
CD_ATL	Enabled	CM_IPU1_STATICDEP[30] ATL_STATDEP	Read/write
CD_EVE1	Disabled	CM_IPU1_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_IPU1_STATICDEP[20] EVE2_STATDEP	Read/write

3.6.4.20.3.2 Dynamic Dependency

Table 3-253 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-253. CD_IPU1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_IPU1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.20.4 Clock Domain Module Attributes

Table 3-254 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-254. CD_IPU1 Modules Clocks Association

Module	Clock	Clock Type
IPU1	IPU1_GFCLK	Interface and Functional
	IPU1_GFCLKDIV2(1)	Interface and Functional

(1) IPU1_GFCLKDIV2 is divided by 2 version of the IPU1_GFCLK. For more information about the IPU1_GFCLK additional division by 2, see Section 7.2.1 *Dual Cortex-M4 IPU Subsystem Clock and Reset Distribution*.

Table 3-255 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-255. CD_IPU1 Modules Wake-Up Request

Module	Wake-Up Feature
IPU1	Master wake-up request

Table 3-256 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-256. CD_IPU1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
IPU1	Master/slave	CM_IPU1_IPU1_CLKCTRL[18] STBYST	Standby status
		CM_IPU1_IPU1_CLKCTRL[17:16] IDLEST	Idle status
		CM_IPU1_IPU1_CLKCTRL[24] CLKSEL	Select the timer functional clock

Table 3-257 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-257. CD_IPU1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
IPU1	Available	Available	N/A	CM_IPU1_IPU1_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.21 CD_IPU2 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.21.1 Overview

Figure 3-85 is an overview of the clock domain.

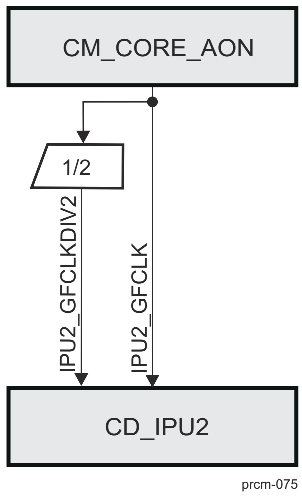


Figure 3-85. CD_IPU2 Overview

3.6.4.21.2 Clock Domain Modes

Table 3-258 lists the clock domain modes supported by the clock domain.

Table 3-258. CD_IPU2 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-259 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-259. CD_IPU2 Control and Status Parameters

Parameter Name	Control/Status Bit Field
IPU2_GFCLK Clock Status	CM_IPU2_CLKSTCTRL[8] CLKACTIVITY_IPU2_GFCLK
Clock Domain State Transition Control	CM_IPU2_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.21.3 Clock Domain Dependency

CD_IPU2 has no module wake-up dependency with any other clock domain of the device.

3.6.4.21.3.1 Static Dependency

Table 3-260 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-260. CD_IPU2 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_DSS	Disabled	CM_IPU2_STATICDEP[8] DSS_STATDEP	Read/write

Table 3-260. CD_IPU2 Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_GPU	Disabled	CM_IPU2_STATICDEP[10] GPU_STATDEP	Read/write
CD_CAM	Alway disabled	CM_IPU2_STATICDEP[9] CAM_STATDEP	Read only
CD_IVA	Disabled	CM_IPU2_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Enabled	CM_IPU2_STATICDEP[5] L3MAIN1_STATDEP	Read/write
CD_L3INIT	Disabled	CM_IPU2_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Enabled	CM_IPU2_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER1	Disabled	CM_IPU2_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_IPU2_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_IPU2_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_L4SEC	Disabled	CM_IPU2_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_EMIF	Enabled	CM_IPU2_STATICDEP[4] EMIF_STATDEP	Read/write
CD_DMA	Alway disabled	CM_IPU2_STATICDEP[11] SDMA_STATDEP	Read only
CD_IPU	Disabled	CM_IPU2_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_IPU2_STATICDEP[23] IPU1_STATDEP	Read/write
CD_DSP	Disabled	CM_IPU2_STATICDEP[1] DSP1_STATDEP	Read/write
CD_DSP2	Disabled	CM_IPU2_STATICDEP[18] DSP2_STATDEP	Read/write
CD_WKUPAON	Enabled	CM_IPU2_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Alway disabled	CM_IPU2_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Alway disabled	CM_IPU2_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_GMAC	Disabled	CM_IPU2_STATICDEP[25] GMAC_STATDEP	Read/write
CD_VPE	Enabled	CM_IPU2_STATICDEP[28] VPE_STATDEP	Read/write
CD_PCIE	Enabled	CM_IPU2_STATICDEP[29] PCIE_STATDEP	Read/write
CD_ATL	Enabled	CM_IPU2_STATICDEP[30] ATL_STATDEP	Read/write
CD_EVE1	Disabled	CM_IPU2_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_IPU2_STATICDEP[20] EVE2_STATDEP	Read/write

3.6.4.21.3.2 Dynamic Dependency

Table 3-261 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-261. CD_IPU2 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_IPU2_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only
CD_CAM	Disabled	CM_IPU2_DYNAMICDEP[9] CAM_DYNDEP	Read only

3.6.4.21.4 Clock Domain Module Attributes

Table 3-262 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-262. CD_IPU2 Modules Clocks Association

Module	Clock	Clock Type
IPU2	IPU2_GFCLK	Interface and Functional
	IPU2_GFCLKDIV2(1)	Interface and Functional

(1) IPU2_GFCLKDIV2 is divided by 2 version of the IPU2_GFCLK. For more information about the IPU2_GFCLK additional division by 2, see Section 7.2.1 *Dual Cortex-M4 IPU Subsystem Clock and Reset Distribution.*

Table 3-263 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-263. CD_IPU2 Modules Wake-Up Request

Module	Wake-Up Feature
IPU2	Master wake-up request

Table 3-264 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-264. CD_IPU2 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
IPU2	Master/slave	CM_IPU2_IPU2_CLKCTRL[18] STBYST	Standby status
		CM_IPU2_IPU2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-265 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-265. CD_IPU2 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
IPU2	Available	Available	N/A	CM_IPU2_IPU2_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.22 CD_DMA Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.22.1 Overview

Figure 3-86 is an overview of the clock domain.

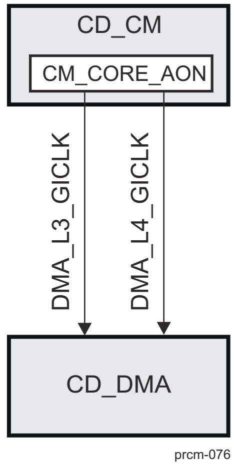


Figure 3-86. CD_DMA Overview

3.6.4.22.2 Clock Domain Modes

Table 3-266 lists the clock domain modes supported by the clock domain.

Table 3-266. CD_DMA Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-267 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-267. CD_DMA Control and Status Parameters

Parameter Name	Control/Status Bit Field
DMA_L3_GICLK Clock Status	CM_DMA_CLKSTCTRL[8] CLKACTIVITY_DMA_L3_GICLK
Clock Domain State Transition Control	CM_DMA_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.22.3 Clock Domain Dependency

CD_DMA has no module wake-up dependency with any other clock domain of the device.

3.6.4.22.3.1 Static Dependency

Table 3-268 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-268. CD_DMA Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_EMIF	Disabled	CM_DMA_STATICDEP[4] EMIF_STATDEP	Read/write
CD_DSS	Disabled	CM_DMA_STATICDEP[8] DSS_STATDEP	Read/write
CD_CAM	Alway disabled	CM_DMA_STATICDEP[9] CAM_STATDEP	Read only
CD_IVA	Disabled	CM_DMA_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Enabled	CM_DMA_STATICDEP[5] L3MAIN1_STATDEP	Read/write
CD_L3INIT	Enabled	CM_DMA_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Enabled	CM_DMA_STATICDEP[12] L4CFG_STATDEP	Read/write

Table 3-268. CD_DMA Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4PER	Enabled	CM_DMA_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4SEC	Disabled	CM_DMA_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_WKUPAON	Enabled	CM_DMA_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_IPU	Disabled	CM_DMA_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_DMA_STATICDEP[23] IPU1_STATDEP	Read/write
CD_L4PER2	Enabled	CM_DMA_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Enabled	CM_DMA_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_PCIE	Enabled	CM_DMA_STATICDEP[29] PCIE_STATDEP	Read/write
CD_IPU2	Disabled	CM_DMA_STATICDEP[0] IPU2_STATDEP	Read/write

3.6.4.22.3.2 Dynamic Dependency

Table 3-269 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-269. CD_DMA Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always disabled	CM_DMA_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.22.4 Clock Domain Module Attributes

Table 3-270 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-270. CD_DMA Modules Clocks Association

Module	Clock	Clock Type
DMA_SYSTEM	DMA_L3_GICLK	Interface(1) and functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the DMA_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-271 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-271. CD_DMA Modules Wake-Up Request

Module	Wake-Up Feature
DMA_SYSTEM	Master wake-up request

Table 3-272 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-272. CD_DMA Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DMA_SYSTEM	Master/slave	CM_DMA_DMA_SYSTEM_CLKCTRL[18] STBYST	Standby status
		CM_DMA_DMA_SYSTEM_CLKCTRL[17:16] IDLEST	Idle status

Table 3-273 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-273. CD_DMA Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DMA_SYSTEM	N/A	Available	N/A	CM_DMA_DMA_SYSTEM_CLKCTRL[1:0] MODULEMODE	Read only

3.6.4.23 CD_ATL Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.23.1 Overview

Figure 3-87 is an overview of the clock domain.

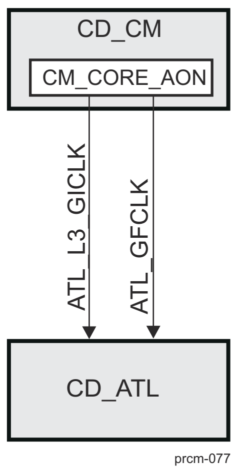


Figure 3-87. CD_ATL Overview

3.6.4.23.2 Clock Domain Modes

Table 3-274 lists the clock domain modes supported by the clock domain.

Table 3-274. CD_ATL Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-275 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-275. CD_ATL Control and Status Parameters

Parameter Name	Control/Status Bit Field
ATL_L3_GICLK Clock Status	CM_ATL_CLKSTCTRL[8] CLKACTIVITY_ATL_L3_GICLK
ATL_GFCLK Clock Status	CM_ATL_CLKSTCTRL[9] CLKACTIVITY_ATL_GFCLK
Clock Domain State Transition Control	CM_ATL_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.23.3 Clock Domain Module Attributes

Table 3-276 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-276. CD_ATL Modules Clocks Association

Module	Clock	Clock Type
ATL	ATL_L3_GICLK	Interface(1)
	ATL_GFCLK	Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the ATL_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-277 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-277. CD_ATL Modules Wake-Up Request

Module	Wake-Up Feature
ATL	None

Table 3-278 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-278. CD_ATL Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
ATL	Slave	CM_ATL_ATL_CLKCTRL[17:16] IDLEST	Idle status
		CM_ATL_ATL_CLKCTRL[27:26] CLKSEL_SOURCE2	Select source for ATL clock
		CM_ATL_ATL_CLKCTRL[25:24] CLKSEL_SOURCE1	Select source for ATL clock

Table 3-279 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-279. CD_ATL Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
ATL	Available	N/A	Available	CM_ATL_ATL_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.24 CD_CAM Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.24.1 Overview

Figure 3-88 is an overview of the clock domain.

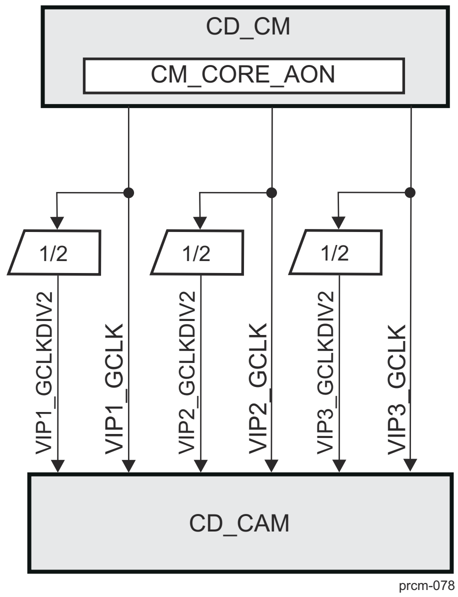


Figure 3-88. CD_CAM Overview

3.6.4.24.2 Clock Domain Modes

Table 3-280 lists the clock domain modes supported by the clock domain.

Table 3-280. CD_CAM Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-281 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-281. CD_CAM Control and Status Parameters

Parameter Name	Control/Status Bit Field
VIP1_GCLK Clock Status	CM_CAM_CLKSTCTRL[8] CLKACTIVITY_VIP1_GCLK
VIP2_GCLK Clock Status	CM_CAM_CLKSTCTRL[9] CLKACTIVITY_VIP2_GCLK
VIP3_GCLK Clock Status	CM_CAM_CLKSTCTRL[10] CLKACTIVITY_VIP3_GCLK
Clock Domain State Transition Control	CM_CAM_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.24.3 Clock Domain Dependency

CD_CAM has no module wake-up dependency with any other clock domain of the device.

3.6.4.24.3.1 Static Dependency

Table 3-282 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-282. CD_CAM Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IVA	Disabled	CM_CAM_STATICDEP[2] IVA_STATDEP	Read/write
CD_L3_MAIN1	Always enabled	CM_CAM_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_L4_CFG	Disabled	CM_CAM_STATICDEP[12] L4CFG_STATDEP	Read only
CD_EMIF	Always enabled	CM_CAM_STATICDEP[4] EMIF_STATDEP	Read/write

Table 3-282. CD_CAM Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_EVE1	Disabled	CM_CAM_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_CAM_STATICDEP[20] EVE2_STATDEP	Read/write
CD_GMAC	Disabled	CM_CAM_STATICDEP[25] GMAC_STATDEP	Read/write
CD_L4PER3	Disabled	CM_CAM_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_VPE	Disabled	CM_CAM_STATICDEP[28] VPE_STATDEP	Read/write

3.6.4.24.3.2 Dynamic Dependency

CD_CAM has no dynamic dependency with any other clock domain of the device.

3.6.4.24.4 Clock Domain Module Attributes

Table 3-283 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-283. CD_CAM Modules Clocks Association

Module	Clock	Clock Type
VIP1	VIP1_GCLK	Functional
	VIP1_GCLKDIV2[(1)]	Interface
VIP2	VIP2_GCLK	Functional
	VIP2_GCLKDIV2[(2)]	Interface
VIP3	VIP3_GCLK	Functional
	VIP3_GCLKDIV2[(3)]	Interface

(1) VIP1_GCLKDIV2 is divided by 2 version of the VIP1_GCLK. For more information about the VIP1_GCLK additional division by 2, please see Section 9.3 *VIP Integration*
(2) VIP2_GCLKDIV2 is divided by 2 version of the VIP2_GCLK. For more information about the VIP2_GCLK additional division by 2, please see Section 9.3 *VIP Integration*
(3) VIP3_GCLKDIV2 is divided by 2 version of the VIP3_GCLK. For more information about the VIP3_GCLK additional division by 2, please see Section 9.3 *VIP Integration*

3.6.4.24.5

Table 3-284 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-284. CD_CAM Modules Wake-Up Request

Module	Wake-Up Feature
VIP1	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)/ Master wake-up request
VIP2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)/ Master wake-up request
VIP3	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)/ Master wake-up request

Table 3-285 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-285. CD_CAM Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
VIP1	Master/slave	CM_CAM_VIP1_CLKCTRL[18] STBYST	Standby status

Table 3-285. CD_CAM Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
		CM_CAM_VIP1_CLKCTRL[17:16] IDLEST	Idle status
VIP2	Master/slave	CM_CAM_VIP2_CLKCTRL[18] STBYST	Standby status
		CM_CAM_VIP2_CLKCTRL[17:16] IDLEST	Idle status
VIP3	Master/slave	CM_CAM_VIP3_CLKCTRL[18] STBYST	Standby status
		CM_CAM_VIP3_CLKCTRL[17:16] IDLEST	Idle status

Table 3-286 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-286. CD_CAM Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
VIP1	Available	Available	N/A	CM_CAM_VIP1_CLKCTRL[1:0] MODULEMODE	Read/write
VIP2	Available	Available	N/A	CM_CAM_VIP2_CLKCTRL[1:0] MODULEMODE	Read/write
VIP3	Available	Available	N/A	CM_CAM_VIP3_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.25 CD_GMAC Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.25.1 Overview

Figure 3-89 is an overview of the clock domain.

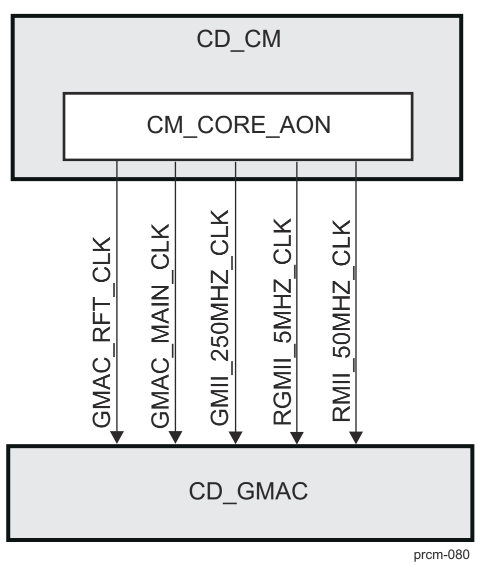


Figure 3-89. CD_GMAC Overview

3.6.4.25.2 Clock Domain Modes

Table 3-287 lists the clock domain modes supported by the clock domain.

Table 3-287. CD_GMAC Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-288 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-288. CD_CAM Control and Status Parameters

Parameter Name	Control/Status Bit Field
GMII_250MHZ_CLK Clock Status	CM_GMAC_CLKSTCTRL[8] CLKACTIVITY_GMII_250MHZ_CLK
RGMII_5MHZ_CLK Clock Status	CM_GMAC_CLKSTCTRL[9] CLKACTIVITY_RGMII_5MHZ_CLK
RMII_50MHZ_CLK Clock Status	CM_GMAC_CLKSTCTRL[10] CLKACTIVITY_RMII_50MHZ_CLK
GMAC_MAIN_CLK Clock Status	CM_GMAC_CLKSTCTRL[12] CLKACTIVITY_GMAC_MAIN_CLK
GMAC_RFT_CLK Clock Status	CM_GMAC_CLKSTCTRL[11] CLKACTIVITY_GMAC_RFT_CLK
Clock Domain State Transition Control	CM_GMAC_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.25.3 Clock Domain Dependency

3.6.4.25.3.1 Static Dependency

Table 3-289 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-289. CD_GMAC Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_GMAC_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_GMAC_STATICDEP[4] EMIF_STATDEP	Read/write
CD_L4PER2	Disabled	CM_GMAC_STATICDEP[26] L4PER2_STATDEP	Read/write

3.6.4.25.3.2 Dynamic Dependency

Table 3-290 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-290. CD_GMAC Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always disabled	CM_GMAC_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.25.4 Clock Domain Module Attributes

Table 3-291 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-291. CD_GMAC Modules Clocks Association“

Module	Clock	Clock Type
CPGMAC	GMAC_MAIN_CLK	Interface
	GMAC_RFT_CLK	Functional
	GMII_250MHZ_CLK	Functional
	RGMII_5MHZ_CLK	Functional
	RMII_50MHZ_CLK	Functional

Table 3-292 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-292. CD_GMAC Modules Wake-Up Request

Module	Wake-Up Feature
GMAC	None

Table 3-293 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-293. CD_GMAC Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
GMAC	Master/slave	CM_GMAC_GMAC_CLKCTRL[18] STBYST	Standby status
		CM_GMAC_GMAC_CLKCTRL[17:16] IDLEST	Idle status
		CM_GMAC_GMAC_CLKCTRL[24] CLKSEL_REF	Select the source of the functional clock
		CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT	Select the source of the CPTS_RFT_CLK

Table 3-294 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-294. CD_GMAC Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GMAC	Available	N/A	Available	CM_GMAC_GMAC_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.26 CD_VPE Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.26.1 CD_VPE Overview

Figure 3-90 is an overview of the clock domain.

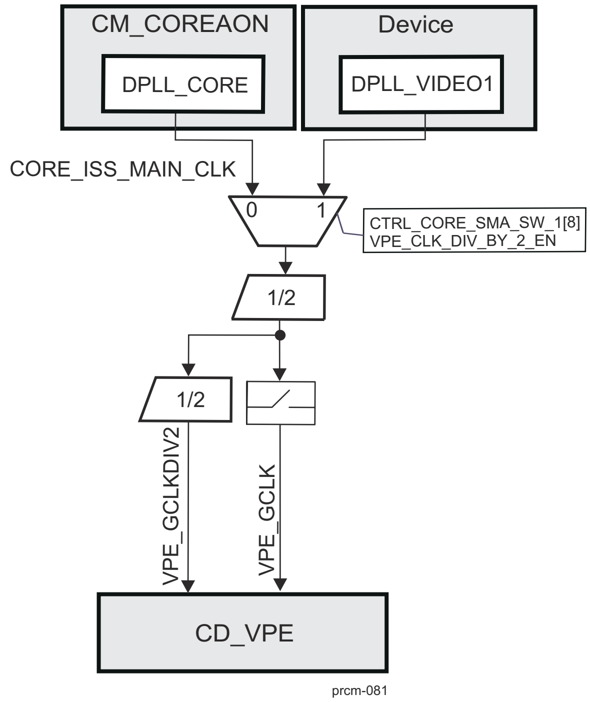


Figure 3-90. CD_VPE Overview

3.6.4.26.2 Clock Domain Modes

Table 3-295 lists the clock domain modes supported by the clock domain.

Table 3-295. CD_VPE Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-296 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-296. CD_VPE Control and Status Parameters

Parameter Name	Control/Status Bit Field
CLKACTIVITY_VPE_GCLK Clock Status	CM_VPE_CLKSTCTRL[8] CLKACTIVITY_VPE_GCLK
Clock Domain State Transition Control	CM_VPE_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.26.3 Clock Domain Dependency

Table 3-297 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-297. CD_VPE Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_VPE_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_VPE_STATICDEP[4] EMIF_STATDEP	Read/write
CD_L4PER3	Disabled	CM_VPE_STATICDEP[27] L4PER3_STATDEP	Read/write

3.6.4.26.3.1 Wake-Up Dependency

Table 3-298 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-298. CD_VPE Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
VPE	CD_VPE	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_VPE_VPE_WKDEP[4] WKUPDEP_VPE_IPU1	Read/ write
	CD_VPE	CD_IPU2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_VPE_VPE_WKDEP[1] WKUPDEP_VPE_IPU2	Read/ write
	CD_VPE	CD_MPU, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_VPE_VPE_WKDEP[0] WKUPDEP_VPE_MPU	Read/ write
	CD_VPE	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_VPE_VPE_WKDEP[2] WKUPDEP_VPE_DSP1	Read/ write
	CD_VPE	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_VPE_VPE_WKDEP[5] WKUPDEP_VPE_DSP2	Read/ write
	CD_VPE	CD_EVE1,CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_VPE_VPE_WKDEP[6] WKUPDEP_VPE_EVE1	Read/ write
	CD_VPE	CD_EVE2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_VPE_VPE_WKDEP[7] WKUPDEP_VPE_EVE2	Read/ write

3.6.4.26.4 Clock Domain Module Attributes

Table 3-299 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-299. CD_VPE Modules Clocks Association

Module	Clock	Clock Type
VPE	VPE_GCLK	Functional
	VPE_GCLKDIV2(1)	Interface

(1) VPE_GCLKDIV2 is divided by 2 version of the VPE_GCLK. For more information about the VPE_GCLK additional division by 2, please see Section 10.2 *VPE Integration*.

Table 3-300 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-300. CD_VPE Modules Wake-Up Request

Module	Wake-Up Feature
VPE	Master wake-up request
	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)

Table 3-301 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-301. CD_VPE Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
VPE	Master/Slave	CM_VPE_VPE_CLKCTRL[18] STBYST	Standby status

Table 3-301. CD_VPE Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
		CM_VPE_VPE_CLKCTRL[17:16] IDLEST	Idle status

Table 3-302 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-302. CD_VPE Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
VPE	Available	Available	N/A	CM_VPE_VPE_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.27 CD_EVE1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.27.1 CD_EVE1 Overview

Figure 3-91 is an overview of the clock domain.

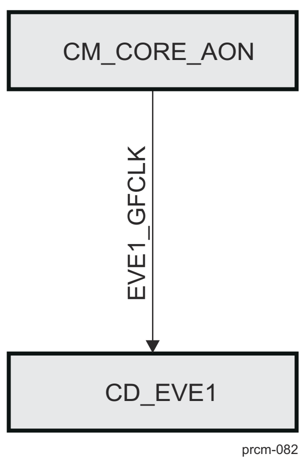


Figure 3-91. CD_EVE1 Overview

3.6.4.27.2 Clock Domain Modes

Table 3-303 lists the clock domain modes supported by the clock domain.

Table 3-303. CD_EVE1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-304 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-304. CD_EVE1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
EVE1_GFCLK Clock Status	CM_EVE1_CLKSTCTRL[8] CLKACTIVITY_EVE1_GFCLK
Clock Domain State Transition Control	CM_EVE1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.27.3 Clock Domain Dependency

Table 3-305 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-305. CD_EVE1 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_EVE1_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Always enabled	CM_EVE1_STATICDEP[4] EMIF_STATDEP	Read/write
CD_IVA	Disabled	CM_EVE1_STATICDEP[2] IVA_STATDEP	Read/write
CD_EVE2	Disabled	CM_EVE1_STATICDEP[20] EVE2_STATDEP	Read/write

3.6.4.27.3.1 Wake-Up Dependency

Table 3-306 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-306. CD_EVE1 Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
EVE1	CD_EVE1	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[4] WKUPDEP_EVE1_IPU1	Read/ write
	CD_EVE1	CD_IPU2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[1] WKUPDEP_EVE1_IPU2	Read/ write
	CD_EVE1	CD_MPU, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[0] WKUPDEP_EVE1_MPU	Read/ write
	CD_EVE1	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[2] WKUPDEP_EVE1_DSP1	Read/ write
	CD_EVE1	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[5] WKUPDEP_EVE1_DSP2	Read/ write
	CD_EVE1	CD_DMA, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[3] WKUPDEP_EVE1_SDMA	Read/ write
	CD_EVE1	CD_EVE2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[7] WKUPDEP_EVE1_EVE2	Read/ write

3.6.4.27.4 Clock Domain Module Attributes

Table 3-307 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-307. CD_EVE1 Modules Clocks Association

Module	Clock	Clock Type
EVE1	EVE1_GFCLK	Interface

Table 3-308 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-308. CD_EVE1 Modules Wake-Up Request

Module	Wake-Up Feature
EVE1	Slave wake-up request (MPU-IRQ, IPU1-IRQ,IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ DMA_SYSTEM-DMA)

Table 3-309 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-309. CD_EVE1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
EVE1	Master/Slave	CM_EVE1_EVE1_CLKCTRL[18] STBYST	Standby status
		CM_EVE1_EVE1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-310 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-310. CD_EVE1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
EVE1	Available	Available	N/A	CM_EVE1_EVE1_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.28 CD_EVE2 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.28.1 CD_EVE2 Overview

Figure 3-92 is an overview of the clock domain.

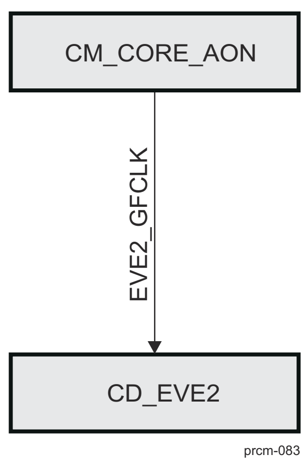


Figure 3-92. CD_EVE2 Overview

3.6.4.28.2 Clock Domain Modes

Table 3-311 lists the clock domain modes supported by the clock domain.

Table 3-311. CD_EVE2 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-312 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-312. CD_EVE2 Control and Status Parameters

Parameter Name	Control/Status Bit Field
EVE2_GFCLK Clock Status	CM_EVE2_CLKSTCTRL[8] CLKACTIVITY_EVE2_GFCLK

Table 3-312. CD_EVE2 Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
Clock Domain State Transition Control	CM_EVE2_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.28.3 Clock Domain Dependency

Table 3-313 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-313. CD_EVE2 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_EVE2_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Always enabled	CM_EVE2_STATICDEP[4] EMIF_STATDEP	Read/write
CD_IVA	Disabled	CM_EVE2_STATICDEP[2] IVA_STATDEP	Read/write
CD_EVE1	Disabled	CM_EVE2_STATICDEP[19] EVE1_STATDEP	Read/write

3.6.4.28.3.1 Wake-Up Dependency

Table 3-314 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-314. CD_EVE2 Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
EVE2	CD_EVE2	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE2_EVE2_WKDEP[4] WKUPDEP_EVE2_IPU1	Read/write
	CD_EVE2	CD_IPU2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE2_EVE2_WKDEP[1] WKUPDEP_EVE2_IPU2	Read/write
	CD_EVE2	CD_MPU, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE2_EVE2_WKDEP[0] WKUPDEP_EVE2_MPU	Read/write
	CD_EVE2	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE2_EVE2_WKDEP[2] WKUPDEP_EVE2_DSP1	Read/write
	CD_EVE2	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE2_EVE2_WKDEP[5] WKUPDEP_EVE2_DSP2	Read/write
	CD_EVE2	CD_DMA, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE2_EVE2_WKDEP[3] WKUPDEP_EVE2_SDMA	Read/write
	CD_EVE2	CD_EVE1,CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE2_EVE2_WKDEP[6] WKUPDEP_EVE2_EVE1	Read/write

3.6.4.28.4 Clock Domain Module Attributes

Table 3-315 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-315. CD_EVE2 Modules Clocks Association

Module	Clock	Clock Type
EVE2	EVE2_GFCLK	Interface

Table 3-316 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-316. CD_EVE2 Modules Wake-Up Request

Module	Wake-Up Feature
EVE2	Slave wake-up request (MPU-IRQ, IPU1-IRQ,IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ DMA_SYSTEM-DMA)

Table 3-317 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-317. CD_EVE2 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
EVE2	Master/Slave	CM_EVE2_EVE2_CLKCTRL[18] STBYST	Standby status
		CM_EVE2_EVE2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-318 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-318. CD_EVE2 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
EVE2	Available	Available	N/A	CM_EVE2_EVE2_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.29 CD_RTC Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.29.1 CD_RTC Overview

Figure 3-93 is an overview of the clock domain.

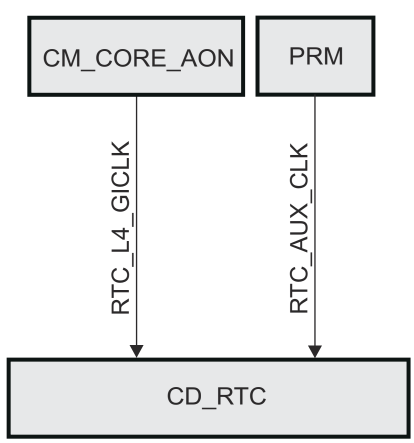


Figure 3-93. CD_RTC Overview

3.6.4.29.2 Clock Domain Modes

Table 3-319 lists the clock domain modes supported by the clock domain.

Table 3-319. CD_RTC Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	N/A	Available	Available

Table 3-320 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-320. CD_RTC Control and Status Parameters

Parameter Name	Control/Status Bit Field
RTC_L4_GICLK Clock Status	CM_RTC_CLKSTCTRL[8] CLKACTIVITY_RTC_L4_GICLK
RTC_AUX_CLK Clock Status	CM_RTC_CLKSTCTRL[10] CLKACTIVITY_RTC_AUX_CLK
Clock Domain State Transition Control	CM_RTC_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.29.3 Clock Domain Dependency

CD_RTC has no static or dynamic dependency with any other clock domain of the device.

3.6.4.29.3.1 Wake-Up Dependency

Table 3-321 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-321. CD_RTC Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
RTC	CD_RTC	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[4] WKUPDEP_RTC_IRQ1_IPU1	Read/write
	CD_RTC	CD_IPU2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[1] WKUPDEP_RTC_IRQ1_IPU2	Read/write
	CD_RTC	CD_MPU, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[0] WKUPDEP_RTC_IRQ1_MPU	Read/write
	CD_RTC	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[2] WKUPDEP_RTC_IRQ1_DSP1	Read/write
	CD_RTC	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[5] WKUPDEP_RTC_IRQ1_DSP2	Read/write
	CD_RTC	CD_EVE1,CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[6] WKUPDEP_RTC_IRQ1_EVE1	Read/write
	CD_RTC	CD_EVE2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[7] WKUPDEP_RTC_IRQ1_EVE2	Read/write
	CD_RTC	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[14] WKUPDEP_RTC_IRQ2_IPU1	Read/write

Table 3-321. CD_RTC Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
	CD_RTC	CD_IPU2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[11] WKUPDEP_RTC_IRQ2_IPU2	Read/write
	CD_RTC	CD_MPU, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[10] WKUPDEP_RTC_IRQ2_MPU	Read/write
	CD_RTC	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[12] WKUPDEP_RTC_IRQ2_DSP1	Read/write
	CD_RTC	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[15] WKUPDEP_RTC_IRQ2_DSP2	Read/write
	CD_RTC	CD_EVE1,CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[16] WKUPDEP_RTC_IRQ2_EVE1	Read/write
	CD_RTC	CD_EVE2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_RTC_RTCSS_WKDEP[17] WKUPDEP_RTC_IRQ2_EVE2	Read/write

3.6.4.29.4 Clock Domain Module Attributes

Table 3-322 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-322. CD_RTC Modules Clocks Association

Module	Clock	Clock Type
RTC	RTC_L4_GICLK	Interface
	RTC_AUX_CLK	Functional

Table 3-323 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-323. CD_RTC Modules Wake-Up Request

Module	Wake-Up Feature
RTC	Slave wake-up request (MPU-IRQ, IPU1-IRQ,IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ, DMA_SYSTEM-DMA)

Table 3-324 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-324. CD_RTC Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
RTC	Slave	CM_RTC_RTCSS_CLKCTRL[17:16] IDLEST	Idle status

Table 3-325 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-325. CD_RTC Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
RTC	Available	N/A	Available	CM_RTC_RTCSS_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.30 CD_PCIE Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.30.1 CD_PCIE Overview

Figure 3-94 is an overview of the clock domain.

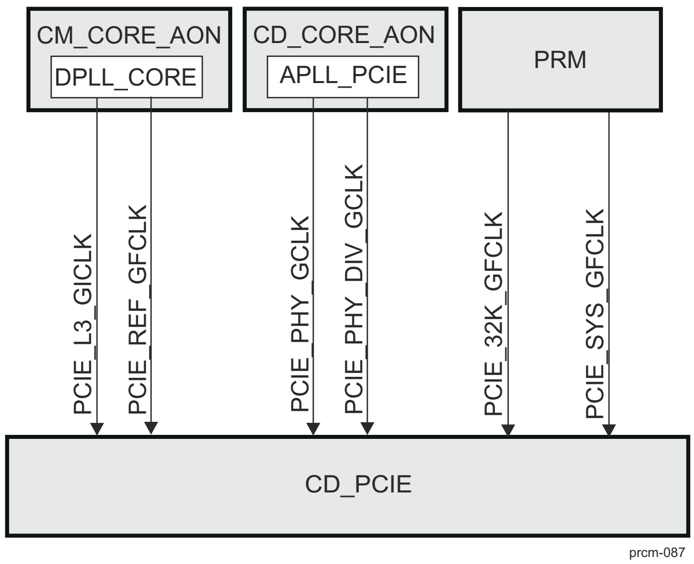


Figure 3-94. CD_PCIE Overview

3.6.4.30.2 Clock Domain Modes

Table 3-326 lists the clock domain modes supported by the clock domain.

Table 3-326. CD_PCIE Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-327 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-327. CD_PCIE Control and Status Parameters

Parameter Name	Control/Status Bit Field
PCIE_L3_GICLK Clock Status	CM_PCIE_CLKSTCTRL[8] CLKACTIVITY_PCIE_L3_GICLK
PCIE_PHY_GCLK Clock Status	CM_PCIE_CLKSTCTRL[9] CLKACTIVITY_PCIE_PHY_GCLK
PCIE_PHY_DIV_GCLK Clock Status	CM_PCIE_CLKSTCTRL[10] CLKACTIVITY_PCIE_PHY_DIV_GCLK
PCIE_REF_GFCLK Clock Status	CM_PCIE_CLKSTCTRL[11] CLKACTIVITY_PCIE_REF_GFCLK
PCIE_SYS_GFCLK Clock Status	CM_PCIE_CLKSTCTRL[12] CLKACTIVITY_PCIE_SYS_GFCLK
PCIE_32K_GFCLK Clock Status	CM_PCIE_CLKSTCTRL[13] CLKACTIVITY_PCIE_32K_GFCLK
Clock Domain State Transition Control	CM_PCIE_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.30.3 Clock Domain Dependency

Table 3-328 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-328. CD_PCIE Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_EMIF	Disabled	CM_PCIE_STATICDEP[4] EMIF_STATDEP	Read/write
CD_IVA	Disabled	CM_PCIE_STATICDEP[2] IVA_STATDEP	Read/write
CD_DSP1	Disabled	CM_PCIE_STATICDEP[1] DSP1_STATDEP	Read/write
CD_DSP2	Disabled	CM_PCIE_STATICDEP[18] DSP2_STATDEP	Read/write
CD_IPU	Disabled	CM_PCIE_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_PCIE_STATICDEP[23] IPU1_STATDEP	Read/write
CD_L3INIT	Disabled	CM_PCIE_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_DSS	Disabled	CM_PCIE_STATICDEP[8] DSS_STATDEP	Read/write
CD_CAM	Disabled	CM_PCIE_STATICDEP[9] CAM_STATDEP	Read/write
CD_GPU	Disabled	CM_PCIE_STATICDEP[10] GPU_STATDEP	Read/write
CD_DMA	Disabled	CM_PCIE_STATICDEP[11] SDMA_STATDEP	Read only
CD_L4_CFG	Disabled	CM_PCIE_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER	Disabled	CM_PCIE_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4SEC	Disabled	CM_PCIE_STATICDEP[14] L4SEC_STATDEP	Read/write
CD_COREAON	Disabled	CM_PCIE_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Disabled	CM_PCIE_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_EVE1	Disabled	CM_PCIE_STATICDEP[19] EVE1_STATDEP	Read/write
CD_EVE2	Disabled	CM_PCIE_STATICDEP[20] EVE2_STATDEP	Read/write
CD_GMAC	Disabled	CM_PCIE_STATICDEP[25] GMAC_STATDEP	Read/write
CD_L4PER2	Disabled	CM_PCIE_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Enabled	CM_PCIE_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_VPE	Disabled	CM_PCIE_STATICDEP[28] VPE_STATDEP	Read/write
CD_ATL	Disabled	CM_PCIE_STATICDEP[30] ATL_STATDEP	Read/write

3.6.4.30.3.1 Wake-Up Dependency

Table 3-329 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-329. CD_PCIE Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
PCIE-PCIe_SS1	CD_PCIE	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS1_WKDEP[4] WKUPDEP_PCIESS1_IPU1	Read/write
	CD_PCIE	CD_IPU2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS1_WKDEP[1] WKUPDEP_PCIESS1_IPU2	Read/write
	CD_PCIE	CD_MPU, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS1_WKDEP[0] WKUPDEP_PCIESS1_MPU	Read/write
	CD_PCIE	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS1_WKDEP[2] WKUPDEP_PCIESS1_DSP1	Read/write
	CD_PCIE	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS1_WKDEP[5] WKUPDEP_PCIESS1_DSP2	Read/write
	CD_PCIE	CD_EVE1,CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS1_WKDEP[6] WKUPDEP_PCIESS1_EVE1	Read/write
	CD_PCIE	CD_EVE2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS1_WKDEP[7] WKUPDEP_PCIESS1_EVE2	Read/write
PCIE-PCIe_SS2	CD_PCIE	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS2_WKDEP[4] WKUPDEP_PCIESS2_IPU1	Read/write
	CD_PCIE	CD_IPU2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS2_WKDEP[1] WKUPDEP_PCIESS2_IPU2	Read/write
	CD_PCIE	CD_MPU, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS2_WKDEP[0] WKUPDEP_PCIESS2_MPU	Read/write
	CD_PCIE	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS2_WKDEP[2] WKUPDEP_PCIESS2_DSP1	Read/write
	CD_PCIE	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS2_WKDEP[5] WKUPDEP_PCIESS2_DSP2	Read/write
	CD_PCIE	CD_EVE1,CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS2_WKDEP[6] WKUPDEP_PCIESS2_EVE1	Read/write
	CD_PCIE	CD_EVE2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_PCIE_PCIESS2_WKDEP[7] WKUPDEP_PCIESS2_EVE2	Read/write

3.6.4.30.4 Clock Domain Module Attributes

Table 3-330 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-330. CD_PCIE Modules Clocks Association

Module	Clock	Clock Type
PCIe_SS1	PCIE_L3_GICLK	Interface
	PCIE_REF_GFCLK	Functional
	PCIE_32K_GFCLK	Functional
	PCIE_PHY_GCLK	Functional
	PCIE_PHY_DIV_GCLK	Functional
PCIe_SS2	PCIE_L3_GICLK	Interface
	PCIE_REF_GFCLK	Functional
	PCIE_32K_GFCLK	Functional
	PCIE_PHY_GCLK	Functional
	PCIE_PHY_DIV_GCLK	Functional

Table 3-331 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-331. CD_PCIE Modules Wake-Up Request

Module	Wake-Up Feature
PCIe_SS1	Slave wake-up request (MPU-IRQ, IPU1-IRQ,IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)/ Master wake-up request
PCIe_SS2	Slave wake-up request (MPU-IRQ, IPU1-IRQ, IPU2-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)/ Master wake-up request

Table 3-332 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-332. CD_PCIE Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
PCIe_SS1	Master/Slave	CM_PCIE_PCIESS1_CLKCTRL[18] STBYST	Standby status
		CM_PCIE_PCIESS1_CLKCTRL[17:16] IDLEST	Idle status
		CM_PCIE_PCIESS1_CLKCTRL[10] OPTFCLKEN_PCIEPHY_CLK_DIV	PCIE PHY optional clock control
		CM_PCIE_PCIESS1_CLKCTRL[9] OPTFCLKEN_PCIEPHY_CLK	PCIE PHY optional clock control
		CM_PCIE_PCIESS1_CLKCTRL[8] OPTFCLKEN_32KHZ	PCIE PHY optional clock control
PCIe_SS2	Master/Slave	CM_PCIE_PCIESS2_CLKCTRL[18] STBYST	Standby status
		CM_PCIE_PCIESS2_CLKCTRL[17:16] IDLEST	Idle status
		CM_PCIE_PCIESS2_CLKCTRL[10] OPTFCLKEN_PCIEPHY_CLK_DIV	PCIE PHY optional clock control
		CM_PCIE_PCIESS2_CLKCTRL[9] OPTFCLKEN_PCIEPHY_CLK	PCIE PHY optional clock control
		CM_PCIE_PCIESS2_CLKCTRL[8] OPTFCLKEN_32KHZ	PCIE PHY optional clock control

Note

In order to disable the APLL_PCIE, the user needs to disable PCIe_SSx (where x = 1 or 2) using the CM_PCIE_PCIESSx_CLKCTRL[1:0] MODULEMODE registers. When PCIe_SS is disabled, the PRCM module automatically disables the APLL_PCIE. Please note that setting CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT bitfield to 0x0 does not disable the APLL_PCIE.

Table 3-333 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-333. CD_PCIE Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
PCIe_SS1	Available	N/A	Available	CM_PCIE_PCIESS1_CLKCTRL[1:0] MODULEMODE	Read/write
PCIe_SS2	Available	N/A	Available	CM_PCIE_PCIESS2_CLKCTRL[1:0] MODULEMODE	Read/write

3.7 Power Management Functional Description

Note

Only the MPU Subsystem supports memory retention.

MPU subsystem does not support OFF state. Only CPU1 supports FORCED_OFF state with no subsequent recovery to ON/active state - this is very application specific and may not be available in all TI standard software offerings.

Note

In the L3INIT power domain OFF state is only allowed in systems where Ethernet RGMII is NOT used in the system - this is very application specific and may not be available in all TI standard software offerings.

This section describes the functional concepts of power management at the power domain level in the device.

The following power domains support dynamic power switching (DPS) with switching times of less than 5 μs.

- PD_CORE
- PD_MPU

3.7.1 PD_WKUPAON Description

PD_WKUPAON contains the following reset domains:

- WKUPAON_PWRON_RST
- WKUPAON_RST
- WKUPAON_SYS_PWRON_RST
- PRM_PWRON_RST
- PRM_RST
- LPRM_PWRON_RST
- LPRM_RST

PD_WKUPAON contains the CD_WKUPAON clock domain.

Table 3-334 lists the logic retention capability for each module of the power domain.

Table 3-334. PD_WKUPAON Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
CTRL_MODULE_WKUP	No	None	None
GPIO1	No	RM_WKUPAON_GPIO1_CONTEXT[0] LOSTCONTEXT_DFF	None
KBD	No	RM_WKUPAON_KBD_CONTEXT[0] LOSTCONTEXT_DFF	None
PRCM_MPU	No	None	None
COUNTER_32K	No	RM_WKUPAON_COUNTER_32K_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER1	No	RM_WKUPAON_TIMER1_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER12	No	RM_WKUPAON_TIMER12_CONTEXT[0] LOSTCONTEXT_DFF	None
WD_TIMER2	No	RM_WKUPAON_WD_TIMER2_CONTEXT[0] LOSTCONTEXT_DFF	None
L4_WKUP interconnect	No	RM_WKUPAON_L4_WKUP_CONTEXT[0] LOSTCONTEXT_DFF	None

Table 3-334. PD_WKUPAON Modules Power Attributes (continued)

Module	Logic Retention	DFF Context Status	RFF Context Status
PRM	No	None	None
DCAN1	No	RM_WKUPAON_DCAN1_CONTEXT[0] LOSTCONTEXT_DFF	None
UART10	No	RM_WKUPAON_UART10_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.1.1 Power Domain Modes

The PD_WKUPAON power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power state control or status bit field for this power domain.

The PD_WKUPAON power domain has two memory banks(UART_MEM, DCAN_MEM) which are related to UART10 and DCAN1.

3.7.1.1.1 Logic and Memory Area Power Modes

Table 3-335 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-335. PD_WKUPAON Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
WKUP_BANK		ON	RETENTION	
	UART10 - UART_MEM	always_on	always_retention	
	DCAN1 - DCAN_MEM	always_on	always_retention	

3.7.2 PD_DSP1 Description

PD_DSP1 contains the following reset domains:

- DSP1_RST
- DSP1_PWRON_RST
- DSP1_RET_RST
- DSP1_SYS_RST

PD_DSP1 contains the CD_DSP1 clock domain.

Table 3-336 lists the logic retention capability for each module of the power domain.

Table 3-336. PD_DSP1 Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DSP1	No	RM_DSP1_DSP1_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.2.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.2.1.1 Logic and Memory Area Power Modes

Table 3-337 lists the power modes supported by the logic area of the power domain.

Table 3-337. PD_DSP1 Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-338 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On and Logic Retention columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-338. PD_DSP1 Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
DSP1_EDMA		ON	RETENTION	OFF
	DSP – DSP_DMA	always_on	always_retention	
DSP1_L1		ON	OFF, RETENTION	OFF
	DSP – DSP_L1	always_on	software_control	
DSP1_L2		ON	OFF, RETENTION	OFF
	DSP – DSP_L2	always_on	software_control	

3.7.2.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-339 lists the power mode controls for the power domain.

Table 3-339. PD_DSP1 Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain – Low-Power State Change Control		PM_DSP1_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area – State Control (logic in ON state)	DSP1_EDMA	PM_DSP1_PWRSTCTRL[21:20] DSP1_EDMA_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	DSP1_L2	PM_DSP1_PWRSTCTRL[19:18] DSP1_L2_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	DSP1_L1	PM_DSP1_PWRSTCTRL[17:16] DSP1_L1_ONSTATE	Read only
Power Domain – State Transition Control		PM_DSP1_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-340 lists the power mode status for the power domain.

Table 3-340. PD_DSP1 Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_DSP1_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	DSP1_EDMA	PM_DSP1_PWRSTST[9:8] DSP1_EDMA_STATEST
Memory Area – State Status	DSP1_L2	PM_DSP1_PWRSTST[7:6] DSP1_L2_STATEST
Memory Area – State Status	DSP1_L1	PM_DSP1_PWRSTST[5:4] DSP1_L1_STATEST
Power Domain – State Transition Status		PM_DSP1_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_DSP1_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_DSP1_PWRSTST[1:0] POWERSTATEST

3.7.3 PD_DSP2 Description

PD_DSP2 contains the following reset domains:

- DSP2_RST
- DSP2_PWRON_RST
- DSP2_RET_RST
- DSP2_SYS_RST

PD_DSP2 contains the CD_DSP2 clock domain.

Table 3-341 lists the logic retention capability for each module of the power domain.

Table 3-341. PD_DSP2 Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DSP2	No	RM_DSP2_DSP2_CONTEXT[1] LOSTCONTEXT_DFF	None

3.7.3.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.3.1.1 Logic and Memory Area Power Modes

Table 3-342 lists the power modes supported by the logic area of the power domain.

Table 3-342. PD_DSP2 Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-343 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-343. PD_DSP2 Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
DSP2_EDMA		ON	RETENTION	OFF
	DSP – DSP_DMA	always_on	always_retention	
DSP2_L1		ON	OFF, RETENTION	OFF
	DSP – DSP_L1	always_on	software_control	
DSP2_L2		ON	OFF, RETENTION	OFF
	DSP – DSP_L2	always_on	software_control	

3.7.3.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-344 lists the power mode controls for the power domain.

Table 3-344. PD_DSP2 Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain – Low-Power State Change Control		PM_DSP2_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area – State Control (logic in ON state)	DSP2_EDMA	PM_DSP2_PWRSTCTRL[21:20] DSP2_EDMA_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	DSP2_L2	PM_DSP2_PWRSTCTRL[19:18] DSP2_L2_ONSTATE	Read only

Table 3-344. PD_DSP2 Power Modes Control Parameters (continued)

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (logic in ON state)	DSP2_L1	PM_DSP2_PWRSTCTRL[17:16] DSP2_L1_ONSTATE	Read only
Power Domain – State Transition Control		PM_DSP2_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-345 lists the power mode status for the power domain.

Table 3-345. PD_DSP2 Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power domain transition status		PM_DSP2_PWRSTST[20] INTRANSITION
Memory Area – State Status	DSP2_L1	PM_DSP2_PWRSTST[5:4] DSP2_L1_STATEST
Memory Area – State Status	DSP2_L2	PM_DSP2_PWRSTST[7:6] DSP2_L2_STATEST
Memory Area – State Status	DSP2_EDMA	PM_DSP2_PWRSTST[9:8] DSP2_EDMA_STATEST
Logic Area – Retention State Control		PM_DSP2_PWRSTST[2] LOGICSTATEST
Power Domain – State Transition Control		PM_DSP2_PWRSTST[1:0] POWERSTATE
Last low power state entered		PM_DSP2_PWRSTST[25:24] LASTPOWERSTATEENTERED

3.7.4 PD_CUSTEFUSE Description

PD_CUSTEFUSE contains the following reset domains:

- CUSTEFUSE_RST

PD_CUSTEFUSE contains the CD_CUSTEFUSE clock domain.

Table 3-336 lists the logic retention capability for each module of the power domain.

Table 3-346. PD_CUSTEFUSE Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
EFUSE_CTRL_CUST	No	RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.4.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.4.1.1 Logic and Memory Area Power Modes

Table 3-337 lists the power modes supported by the logic area of the power domain.

Table 3-347. PD_CUSTEFUSE Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-338 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

There is no memory bank implemented for the PD_CUSTEFUSE.

3.7.4.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-339 lists the power mode controls for the power domain.

Table 3-348. PD_CUSTEFUSE Power Modes Control Parameters

Parameter Name	Control Bit Field	Access Type
Power Domain – Low-Power State Change Control	PM_CUSTEFUSE_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Power Domain – State Transition Control	PM_CUSTEFUSE_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-340 lists the power mode status for the power domain.

Table 3-349. PD_CUSTEFUSE Power Modes Status Parameters

Parameter Name	Status Bit Field
Power Domain – Last Power State Entered Status	PM_CUSTEFUSE_PWRSTST[25:24] LASTPOWERSTATEENTERED
Power Domain – State Transition Status	PM_CUSTEFUSE_PWRSTST[20] INTRANSITION
Logic Area – State Status	PM_CUSTEFUSE_PWRSTST[2] LOGICSTATEST
Power Domain – State Status	PM_CUSTEFUSE_PWRSTST[1:0] POWERSTATEST

3.7.5 PD_MPU Description

PD_MPU contains the following reset domains:

- MPU_PWRON_RST
- MPU_RST
- MPU_MA_RST
- MPU_MA_RET_RST
- MPU_MA_PWRON_RET_RST

PD_MPU contains the CD_MPU clock domain.

Table 3-350 lists the logic retention capability for each module of the power domain.

Table 3-350. PD_MPU Module Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
MPU	Partial	RM_MPU_MPU_CONTEXT[0] LOSTCONTEXT_DFF	RM_MPU_MPU_CONTEXT[1] LOSTCONTEXT_RFF

3.7.5.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.5.1.1 Logic and Memory Area Power Modes

Table 3-351 lists the power modes supported by the logic area of the power domain.

Table 3-351. PD_MPU Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
N/A	Available	Available	Available

Table 3-352 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-352. PD_MPU Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
MPU_L2		ON	OFF or RETENTION	
	MPU – MPU_L2	always_on	always_retention	
MPU_RAM		ON	RETENTION	
	MPU - MPU_RAM	always_on	always_retention	

3.7.5.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-353 lists the power mode controls for the power domain.

Table 3-353. PD_MPU Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (logic in RETENTION state)	MPU_L2	PM_MPU_PWRSTCTRL[9] MPU_L2_RETSTATE	Read/write
Memory Area – State Control (logic in RETENTION state)	MPU_RAM	PM_MPU_PWRSTCTRL[10] MPU_RAM_RETSTATE	Read only
Power Domain – Low-Power State Change Control		PM_MPU_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read only
Logic Area – Retention State Control		PM_MPU_PWRSTCTRL[2] LOGICRETSTATE	Read/write
Memory Area – State Control (logic in ON state)	MPU_L2	PM_MPU_PWRSTCTRL[19:18] MPU_L2_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	MPU_RAM	PM_MPU_PWRSTCTRL[21:20] MPU_RAM_ONSTATE	Read only
Power Domain – State Transition Control		PM_MPU_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-354 lists the status of the power modes for the power domain.

Table 3-354. PD_MPU Power Mode Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_MPU_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	MPU_L2	PM_MPU_PWRSTST[7:6] MPU_L2_STATEST
Memory Area – State Status	MPU_RAM	PM_MPU_PWRSTST[9:8] MPU_RAM_STATEST
Power Domain – State Transition Status		PM_MPU_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_MPU_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_MPU_PWRSTST[1:0] POWERSTATEST

3.7.5.1.3 Power State Override

The PRCM module controls the power state of the MPU subsystem, whereas the PRCM_MPU controls the power state of each CPU (CPU0 and CPU1). The MPU subsystem requires that the MPU power domain can transition only to a low-power mode when CPU0 and CPU1 are in a lower power mode than is specified in the PM_MPU_PWRSTCTRL register. The power mode of the CPUs is communicated to the PRCM module by the PRCM_MPU through two dedicated internal signals. These two signals specify the lowest power state that the MPU can enter and, if necessary, override the low-power state programmed in the PM_MPU_PWRSTCTRL register.

Table 3-355 lists the low-power modes allowed by the MPU.

Table 3-355. MPU Allowed Low-Power Mode

MPU Allowed Low-Power Mode	Comment
CSWRET	CPUs are in SR3-APG mode and L1$ is in RETENTION state.
INACTIVE	CPUs are in SR3-APG mode and L1$ is in ON state, or CPUs are in INACTIVE state.
ON	CPUs are in ON state.

If PM_MPU_PWRSTCTRL is programmed to a lower power state than allowed (specified by the two signals), then it is overwritten by the hardware. The valued of the PM_MPU_PWRSTCTRL register is not changed (that is, not overwritten).

3.7.6 PD_IPU Description

PD_IPU contains the following reset domains:

- IPU1_PWRON_RST
- IPU1_RST
- IPU1_RET_RST
- IPU_RST
- IPU_RET_RST
- IPU1_CPU0_RST
- IPU1_CPU1_RST

PD_IPU contains the CD_IPU and CD_IPU1 clock domain.

Table 3-356 lists the logic retention capability for each module of the power domain.

Table 3-356. PD_IPU Module Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
IPU1	Partial	RM_IPU1_IPU1_CONTEXT[0] LOSTCONTEXT_DFF	RM_IPU1_IPU1_CONTEXT[1] LOSTCONTEXT_RFF
McASP1	No	RM_IPU_MCASP1_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER5	No	RM_IPU_TIMER5_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER6	No	RM_IPU_TIMER6_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER7	No	RM_IPU_TIMER7_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER8	No	RM_IPU_TIMER8_CONTEXT[0] LOSTCONTEXT_DFF	None
UART6	Full	None	RM_IPU_UART6_CONTEXT[1] LOSTCONTEXT_RFF
I2C5	No	RM_IPU_I2C5_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.6.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.6.1.1 Logic and Memory Area Power Modes

Table 3-357 lists the power modes supported by the logic area of the power domain.

Table 3-357. PD_IPU Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-358 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-358. PD_IPU Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
AESSMEM		ON	OFF or RETENTION	
	IPU1 – IPU_L2RAM_MEM	always_on	software_control	
PERIPHMEM		ON	OFF or RETENTION	
	IPU1 - IPU_UNICACHE_MEM	always_on	always_retention	
	UART6 - UART_MEM	always_on	always_retention	

3.7.6.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-359 lists the power mode controls for the power domain.

Table 3-359. PD_IPU Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (logic in RETENTION state)	AESSMEM	PM_IPU_PWRSTCTRL[8] AESSMEM_RETSTATE	Read/write
Memory Area – State Control (logic in RETENTION state)	PERIPHMEM	PM_IPU_PWRSTCTRL[10] PERIPHMEM_RETSTATE	Read/write
Power Domain – Low-Power State Change Control		PM_IPU_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area – Retention State Control		PM_IPU_PWRSTCTRL[2] LOGICRETSTATE	Read only
Memory Area – State Control (logic in ON state)	AESSMEM	PM_IPU_PWRSTCTRL[17:16] AESSMEM_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	PERIPHMEM	PM_IPU_PWRSTCTRL[21:20] PERIPHMEM_ONSTATE	Read only
Power Domain – State Transition Control		PM_IPU_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-360 lists the status of the power modes for the power domain.

Table 3-360. PD_IPU Power Mode Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_IPU_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	AESSMEM	PM_IPU_PWRSTST[5:4] AESSMEM_STATEST
Memory Area – State Status	PERIPHMEM	PM_IPU_PWRSTST[9:8] PERIPHMEM_STATEST
Power Domain – State Transition Status		PM_IPU_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_IPU_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_IPU_PWRSTST[1:0] POWERSTATEST

3.7.7 PD_L3INIT Description

PD_L3INIT contains the following reset domains:

- L3INIT_PWRON_RST
- L3INIT_RET_RST
- L3INIT_RST

PD_L3INIT contains the CD_L3INIT clock domain.

Table 3-361 lists the logic retention capability for each module of the power domain.

Table 3-361. PD_L3INIT Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
IEEE1500_2_OCP	No	RM_L3INIT_IEEE1500_2_OCP_CONTEXT[0] LOSTCONTEXT_DFF	None
MMC1	Full	None	RM_L3INIT_MMC1_CONTEXT[1] LOSTCONTEXT_RFF
MMC2	Full	None	RM_L3INIT_MMC2_CONTEXT[1] LOSTCONTEXT_RFF
MLB_SS	No	RM_L3INIT_MLB_SS_CONTEXT[0] LOSTCONTEXT_DFF	None
USB1	Full	None	RM_L3INIT_USB_OTG_SS1_CONTEXT[1] LOSTCONTEXT_RFF
USB2	Full	None	RM_L3INIT_USB_OTG_SS2_CONTEXT[1] LOSTCONTEXT_RFF
USB3	Full	None	RM_L3INIT_USB_OTG_SS3_CONTEXT[1] LOSTCONTEXT_RFF
USB4	Full	None	RM_L3INIT_USB_OTG_SS4_CONTEXT[1] LOSTCONTEXT_RFF
SATA	No	RM_L3INIT_SATA_CONTEXT[0] LOSTCONTEXT_DFF	None
OCP2SCP1	No	RM_L3INIT_OCP2SCP1_CONTEXT[0] LOSTCONTEXT_DFF	None
OCP2SCP3	No	RM_L3INIT_OCP2SCP3_CONTEXT[0] LOSTCONTEXT_DFF	None
PCIe_SS1	No	RM_PCIE_PCIESS1_CONTEXT[0] LOSTCONTEXT_DFF	None
PCIe_SS2	No	RM_PCIE_PCIESS2_CONTEXT[0] LOSTCONTEXT_DFF	None
CPGMAC	No	RM_GMAC_GMAC_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.7.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.7.1.1 Logic and Memory Area Power Modes

Table 3-362 lists the power modes supported by the logic area of the power domain.

Table 3-362. PD_L3INIT Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available

Table 3-363 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-363. PD_L3INIT Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
L3INIT_BANK1		ON	OFF	OFF
	MLB - MLB_MEM	always_on	always_retention	
	MMC1 – MMC_RAM	always_on	always_off	
	MMC2 – MMC_RAM	always_on	always_off	
	PCIe_SS1 – PCIE_MEM	always_on	always_off	
	PCIe_SS2 – PCIE_MEM	always_on	always_off	
	SATA – sata_bank	always_on	always_off	
L3INIT_BANK2		ON	RETENTION	OFF
	USB_OTG_SS1 – USB_MEM	always_on	always_retention	
	USB_OTG_SS2– USB_MEM	always_on	always_retention	
	USB_OTG_SS3 – USB_MEM	always_on	always_retention	
	USB_OTG_SS4 – USB_MEM	always_on	always_retention	
GMAC_BANK		ON	RETENTION	OFF
	CPGMAC – cpgmacss_r.ram_tx_db	always_on	always_retention	

3.7.7.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-364 lists the power modes controls for the power domain.

Table 3-364. PD_L3INIT Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (logic in RETENTION state)	L3INIT_BANK1	PM_L3INIT_PWRSTCTRL[8] L3INIT_BANK1_RETSTATE	Read only
Memory Area – State Control (logic in RETENTION state)	L3INIT_BANK2	PM_L3INIT_PWRSTCTRL[9] L3INIT_BANK2_RETSTATE	Read only
Memory Area -State Control (logic in RETENTION state)	GMAC_BANK	PM_L3INIT_PWRSTCTRL[10] GMAC_BANK_RETSTATE	Read only
Power Domain – Low-Power State Change Control		PM_L3INIT_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area – RETENTION State Control		PM_L3INIT_PWRSTCTRL[2] LOGICRETSTATE	Read/write
Memory Area – State Control (logic in ON state)	L3INIT_BANK1	PM_L3INIT_PWRSTCTRL[15:14] L3INIT_BANK1_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	L3INIT_BANK2	PM_L3INIT_PWRSTCTRL[17:16] L3INIT_BANK2_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	GMAC_BANK	PM_L3INIT_PWRSTCTRL[19:18] GMAC_BANK_ONSTATE	Read only
Power Domain – State Transition Control		PM_L3INIT_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-365 lists the status of the power modes for the power domain.

Table 3-365. PD_L3INIT Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_L3INIT_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	L3INIT_BANK1	PM_L3INIT_PWRSTST[5:4] L3INIT_BANK1_STATEST
Memory Area – State Status	L3INIT_BANK2	PM_L3INIT_PWRSTST[7:6] L3INIT_BANK2_STATEST
Memory Area – State Status	L3INIT_GMAC	PM_L3INIT_PWRSTST[9:8] L3INIT_GMAC_STATEST
Power Domain – State Transition Status		PM_L3INIT_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_L3INIT_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_L3INIT_PWRSTST[1:0] POWERSTATEST

3.7.8 PD_L4PER Description

PD_L4PER contains the following reset domains:

- L4PER_PWRON_RET_RST
- L4PER_RET_RST
- L4PER_RST

PD_L4PER contains the CD_L4PER1, CD_L4PER2, CD_L4PER3,CD_L4SEC clock domains.

Table 3-366 lists the logic retention capability for each module of the power domain.

Table 3-366. PD_L4PER Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
AES1	Full	None	None
AES2	Full	None	None
DCAN2	No	RM_L4PER2_DCAN2_CONTEXT[0] LOSTCONTEXT_DFF	None
DES3DES	Full	None	RM_L4SEC_DES3DES_CONTEXT[1] LOSTCONTEXT_RFF
DMA_CRYPTO	Full	None	RM_L4SEC_DMA_CRYPTO_CONTEXT[1] LOSTCONTEXT_RFF
ELM	No	RM_L4PER_ELM_CONTEXT[0] LOSTCONTEXT_DFF	No
FPKA	No	RM_L4SEC_FPKA_CONTEXT[0] LOSTCONTEXT_DFF	No
GPIO2	Full	None	RM_L4PER_GPIO2_CONTEXT[1] LOSTCONTEXT_RFF
GPIO3	Full	None	RM_L4PER_GPIO3_CONTEXT[1] LOSTCONTEXT_RFF
GPIO4	Full	None	RM_L4PER_GPIO4_CONTEXT[1] LOSTCONTEXT_RFF
GPIO5	Full	None	RM_L4PER_GPIO5_CONTEXT[1] LOSTCONTEXT_RFF
GPIO6	Full	None	RM_L4PER_GPIO6_CONTEXT[1] LOSTCONTEXT_RFF
GPIO7	Full	None	RM_L4PER_GPIO7_CONTEXT[1] LOSTCONTEXT_RFF
GPIO8	Full	None	RM_L4PER_GPIO8_CONTEXT[1] LOSTCONTEXT_RFF
HDQ1W	No	RM_L4PER_HDQ1W_CONTEXT[0] LOSTCONTEXT_DFF	None

Table 3-366. PD_L4PER Modules Power Attributes (continued)

Module	Logic Retention	DFF Context Status	RFF Context Status
I2C1	Full	None	RM_L4PER_I2C1_CONTEXT[1] LOSTCONTEXT_RFF
I2C2	No	RM_L4PER_I2C2_CONTEXT[0] LOSTCONTEXT_DFF	None
I2C3	No	RM_L4PER_I2C3_CONTEXT[0] LOSTCONTEXT_DFF	None
I2C4	No	RM_L4PER_I2C4_CONTEXT[0] LOSTCONTEXT_DFF	None
L4_PER1	Partial	RM_L4PER_L4PER1_CONTEXT[0] LOSTCONTEXT_DFF	RM_L4PER_L4PER1_CONTEXT[1] LOSTCONTEXT_RFF
L4PER2	Partial	RM_L4PER2_L4PER2_CONTEXT[0] LOSTCONTEXT_DFF	RM_L4PER2_L4PER2_CONTEXT[1] LOSTCONTEXT_RFF
L4PER3	Partial	RM_L4PER3_L4PER3_CONTEXT[0] LOSTCONTEXT_DFF	RM_L4PER3_L4PER3_CONTEXT[1] LOSTCONTEXT_RFF
McASP2	No	RM_L4PER2_MCASP2_CONTEXT[0] LOSTCONTEXT_DFF	None
McASP3	No	RM_L4PER2_MCASP3_CONTEXT[0] LOSTCONTEXT_DFF	None
McASP4	No	RM_L4PER2_MCASP4_CONTEXT[0] LOSTCONTEXT_DFF	None
McASP5	No	RM_L4PER2_MCASP5_CONTEXT[0] LOSTCONTEXT_DFF	None
McASP6	No	RM_L4PER2_MCASP6_CONTEXT[0] LOSTCONTEXT_DFF	None
McASP7	No	RM_L4PER2_MCASP7_CONTEXT[0] LOSTCONTEXT_DFF	None
McASP8	No	RM_L4PER2_MCASP8_CONTEXT[0] LOSTCONTEXT_DFF	None
McSPI1	No	RM_L4PER_MCSPI1_CONTEXT[0] LOSTCONTEXT_DFF	None
McSPI2	No	RM_L4PER_MCSPI2_CONTEXT[0] LOSTCONTEXT_DFF	None
McSPI3	No	RM_L4PER_MCSPI3_CONTEXT[0] LOSTCONTEXT_DFF	None
McSPI4	No	RM_L4PER_MCSPI4_CONTEXT[0] LOSTCONTEXT_DFF	None
MMC3	No	RM_L4PER_MMC3_CONTEXT[0] LOSTCONTEXT_DFF	None
MMC4	No	RM_L4PER_MMC4_CONTEXT[0] LOSTCONTEXT_DFF	None
PWMSS1	No	RM_L4PER2_PWMSS1_CONTEXT[0] LOSTCONTEXT_DFF	None
PWMSS2	No	RM_L4PER2_PWMSS2_CONTEXT[0] LOSTCONTEXT_DFF	None
PWMSS3	No	RM_L4PER2_PWMSS3_CONTEXT[0] LOSTCONTEXT_DFF	None
QSPI	No	RM_L4PER2_QSPI_CONTEXT[0] LOSTCONTEXT_DFF	None
RNG	Full	None	RM_L4SEC_RNG_CONTEXT[1] LOSTCONTEXT_RFF
SHA2MD5_1	Full	None	RM_L4SEC_SHA2MD51_CONTEXT[1] LOSTCONTEXT_RFF

Table 3-366. PD_L4PER Modules Power Attributes (continued)

Module	Logic Retention	DFF Context Status	RFF Context Status
SHA2MD5_2	Full	None	RM_L4SEC_SHA2MD52_CONTEXT[1] LOSTCONTEXT_RFF
TIMER2	No	RM_L4PER_TIMER2_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER3	No	RM_L4PER_TIMER3_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER4	No	RM_L4PER_TIMER4_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER9	No	RM_L4PER_TIMER9_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER10	No	RM_L4PER_TIMER10_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER11	No	RM_L4PER_TIMER11_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER13	No	RM_L4PER3_TIMER13_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER14	No	RM_L4PER3_TIMER14_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER15	No	RM_L4PER3_TIMER15_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER16	No	RM_L4PER3_TIMER16_CONTEXT[0] LOSTCONTEXT_DFF	None
UART1	Full	None	RM_L4PER_UART1_CONTEXT[1] LOSTCONTEXT_RFF
UART2	Full	None	RM_L4PER_UART2_CONTEXT[1] LOSTCONTEXT_RFF
UART3	Full	None	RM_L4PER_UART3_CONTEXT[1] LOSTCONTEXT_RFF
UART4	Full	None	RM_L4PER_UART4_CONTEXT[1] LOSTCONTEXT_RFF
UART5	Full	None	RM_L4PER_UART5_CONTEXT[1] LOSTCONTEXT_RFF
UART7	Full	None	RM_L4PER2_UART7_CONTEXT[1] LOSTCONTEXT_RFF
UART8	Full	None	RM_L4PER2_UART8_CONTEXT[1] LOSTCONTEXT_RFF
UART9	Full	None	RM_L4PER2_UART9_CONTEXT[1] LOSTCONTEXT_RFF

3.7.8.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.8.1.1 Logic and Memory Area Power Modes

Table 3-367 lists the power modes supported by the logic area of the power domain.

Table 3-367. PD_L4PER Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
N/A	Available	Available	Available

Table 3-368 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-368. PD_L4PER Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
NONRETAINED_ BANK		ON	OFF	OFF
	MMC3 - MMC_RAM	always_on	always_off	
	MMC4 – MMC_RAM	always_on	always_off	
	FPKA – pka_mem	always_on	always_off	
RETAINED_BANK		ON	RETENTION	OFF
	DCAN2 – DCAN_MEM	always_on	always_retention	
	DMA_CRYPTO – DMA_CRYPTO_MEM	always_on	always_retention	
	UART1 – UART_MEM	always_on	always_retention	
	UART2 – UART_MEM	always_on	always_retention	
	UART3 – UART_MEM	always_on	always_retention	
	UART4 – UART_MEM	always_on	always_retention	
	UART5 – UART_MEM	always_on	always_retention	
	UART7 – UART_MEM	always_on	always_retention	
	UART8 – UART_MEM	always_on	always_retention	
	UART9 – UART_MEM	always_on	always_retention	

3.7.8.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-369 lists the power modes controls for the power domain.

Table 3-369. PD_L4PER Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (logic in RETENTION state)	RETAINED_BANK	PM_L4PER_PWRSTCTRL[8] RETAINED_BANK_RETSTATE	Read only
Memory Area – State Control (logic in RETENTION state)	NONRETAINED_BANK	PM_L4PER_PWRSTCTRL[9] NONRETAINED_BANK_RETSTATE	Read only
Power Domain – Low-Power State Change Control		PM_L4PER_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area – RETENTION State Control		PM_L4PER_PWRSTCTRL[2] LOGICRETSTATE	Read/write
Memory Area – State Control (logic in ON state)	RETAINED_BANK	PM_L4PER_PWRSTCTRL[17:16] RETAINED_BANK_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	NONRETAINED_BANK	PM_L4PER_PWRSTCTRL[19:18] NONRETAINED_BANK_ONSTATE	Read only
Power Domain – State Transition Control		PM_L4PER_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-370 lists the status of the power modes for the power domain.

Table 3-370. PD_L4PER Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_L4PER_PWRSTST[25:24] LASTPOWERSTATEENTERED

Table 3-370. PD_L4PER Power Modes Status Parameters (continued)

Parameter Name	Memory Bank	Status Bit Field
Memory Area – State Status	RETAINED_BANK	PM_L4PER_PWRSTST[5:4] RETAINED_BANK_STATEST
Memory Area – State Status	L3INIT_BANK2	PM_L4PER_PWRSTST[7:6] NONRETAINED_BANK_STATEST
Power Domain – State Transition Status		PM_L4PER_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_L4PER_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_L4PER_PWRSTST[1:0] POWERSTATEST

3.7.9 PD_IVA Description

PD_IVA contains the following reset domains:

- IVA_PWRON_RST
- IVA_RST
- IVA_SEQ1_RST
- IVA_SEQ2_RST

PD_IVA contains the CD_IVA clock domain.

Table 3-371 lists the logic retention capability for each module of the power domain.

Table 3-371. PD_IVA Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
IVA	No	RM_IVA_IVA_CONTEXT[0] LOSTCONTEXT_DFF	None
SL2	No	RM_IVA_SL2_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.9.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.9.1.1 Logic and Memory Area Power Modes

Table 3-372 lists the power modes supported by the logic area of the power domain.

Table 3-372. PD_IVA Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Available	Available

Table 3-373 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-373. PD_IVA Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
HWA_MEM		ON	OFF	OFF
	IVA – HWA_MEM	always_on	always_off	
SL2_MEM		ON	OFF or RETENTION	OFF
	SL2 – SL2MEM	always_on	software_control	
TCM_1_MEM		ON	OFF or RETENTION	OFF

Table 3-373. PD_IVA Memory Area Power Modes (continued)

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
	IVA – TCM_1	always_on	software_control	
TCM_2_MEM		ON	OFF or RETENTION	OFF
	IVA – TCM_2	always_on	software_control	

3.7.9.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-374 lists the power mode controls for the power domain.

Table 3-374. PD_IVA Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (logic in RETENTION state)	SL2_MEM	PM_IVA_PWRSTCTRL[9] SL2_MEM_RETSTATE	Read/write
Memory Area – State Control (logic in RETENTION state)	HWA_MEM	PM_IVA_PWRSTCTRL[8] HWA_MEM_RETSTATE	Read only
Power Domain – Low-Power State Change Control		PM_IVA_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area – State Control (logic in ON state)	TCM_2_MEM	PM_IVA_PWRSTCTRL[23:22] TCM2_MEM_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	TCM_1_MEM	PM_IVA_PWRSTCTRL[21:20] TCM1_MEM_ONSTATE	Read only
Logic Area – Retention State Control		PM_IVA_PWRSTCTRL[2] LOGICRETSTATE	Read only
Memory Area – State Control (logic in ON state)	SL2_MEM	PM_IVA_PWRSTCTRL[19:18] SL2_MEM_ONSTATE	Read only
Memory Area – State Control (logic in ON state)	HWA_MEM	PM_IVA_PWRSTCTRL[17:16] HWA_MEM_ONSTATE	Read only
Memory Area – State Control (logic in RETENTION state)	TCM_2_MEM	PM_IVA_PWRSTCTRL[11] TCM2_MEM_RETSTATE	Read/write
Memory Area – State Control (logic in RETENTION state)	TCM_1_MEM	PM_IVA_PWRSTCTRL[10] TCM1_MEM_RETSTATE	Read/write
Power Domain – State Transition Control		PM_IVA_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-375 lists the status of the power modes for the power domain.

Table 3-375. PD_IVA Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_IVA_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	TCM_1_MEM	PM_IVA_PWRSTST[9:8] TCM1_MEM_STATEST
Memory Area – State Status	TCM_2_MEM	PM_IVA_PWRSTST[11:10] TCM2_MEM_STATEST
Memory Area – State Status	SL2_MEM	PM_IVA_PWRSTST[7:6] SL2_MEM_STATEST
Memory Area – State Status	HWA_MEM	PM_IVA_PWRSTST[5:4] HWA_MEM_STATEST
Power Domain – State Transition Status		PM_IVA_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_IVA_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_IVA_PWRSTST[1:0] POWERSTATEST

3.7.10 PD_GPU Description

PD_GPU contains the GPU_RST reset domain.

PD_GPU contains the CD_GPU clock domain.

Table 3-376 lists the logic retention capability for each module of the power domain.

Table 3-376. PD_GPU Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
GPU	No	RM_GPU_GPU_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.10.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.10.1.1 Logic and Memory Area Power Modes

Table 3-377 lists the power modes supported by the logic area of the power domain.

Table 3-377. PD_GPU Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Available	Available

Table 3-378 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-378. PD_GPU Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
GPU_MEM		ON		OFF
	GPU – GPU_MEM	always_on	always_off	

3.7.10.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-379 lists the power mode controls for the power domain.

Table 3-379. PD_GPU Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain – Low-Power State Change Control		PM_GPU_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area – State Control (logic in ON state)	GPU_MEM	PM_GPU_PWRSTCTRL[17:16] GPU_MEM_ONSTATE	Read only
Power Domain – State Transition Control		PM_GPU_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-380 lists the status of the power modes for the power domain.

Table 3-380. PD_GPU Power Mode Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Memory Area – State Status	GPU_MEM	PM_GPU_PWRSTST[5:4] GPU_MEM_STATEST
Power Domain – Last Power State Entered Status		PM_GPU_PWRSTST[25:24] LASTPOWERSTATEENTERED

Table 3-380. PD_GPU Power Mode Status Parameters (continued)

Parameter Name	Memory Bank	Status Bit Field
Power Domain – State Transition Status		PM_GPU_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_GPU_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_GPU_PWRSTST[1:0] POWERSTATEST

3.7.11 PD_EMU Description

PD_EMU contains the following reset domains:

- EMU_EARLY_PWRON_RST
- EMU_PWRON_RST
- EMU_RST

PD_EMU contains the CD_EMU clock domain.

Table 3-381 lists the logic retention capability for each module of the power domain.

Table 3-381. PD_EMU Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DEBUGSS logic	No	RM_EMU_DEBUGSS_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.11.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.11.1.1 Logic and Memory Area Power Modes

Table 3-382 lists the power modes supported by the logic area of the power domain.

Table 3-382. PD_EMU Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Available	Available

Table 3-383 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-383. PD_EMU Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
EMU_BANK		ON		OFF
	DEBUGSS – DebugSS_MEM	always_on	always_off	

3.7.11.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-384 lists the power modes controls for the power domain.

Table 3-384. PD_EMU Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (logic in ON state)	EMU_BANK	PM_EMU_PWRSTCTRL[17:16] EMU_BANK_ONSTATE	Read only

Table 3-384. PD_EMU Power Modes Control Parameters (continued)

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain - State Transition Control		PM_EMU_PWRSTCTRL[1:0] POWERSTATE	Read only

Table 3-385 lists the status of the power modes for the power domain.

Table 3-385. PD_EMU Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Memory Area – State Status	EMU_BANK	PM_EMU_PWRSTST[5:4] EMU_BANK_STATEST
Power Domain – Last Power State Entered Status		PM_EMU_PWRSTST[25:24] LASTPOWERSTATEENTERED
Power Domain – State Transition Status		PM_EMU_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_EMU_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_EMU_PWRSTST[1:0] POWERSTATEST

3.7.12 PD_DSS Description

PD_DSS contains the following reset domains:

- DSS_RET_RST
- DSS_RST

PD_DSS contains the CD_DSS clock domain.

Table 3-386 lists the logic retention capability for each module of the power domain.

Table 3-386. PD_DSS Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DSS	Partial	RM_DSS_DSS_CONTEXT[0] LOSTCONTEXT_DFF	RM_DSS_DSS_CONTEXT[1] LOSTCONTEXT_RFF
BB2D	None	RM_DSS_BB2D_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.12.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.12.1.1 Logic and Memory Area Power Modes

Table 3-387 lists the power modes supported by the logic area of the power domain.

Table 3-387. PD_DSS Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-388 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-388. PD_DSS Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
DSS_MEM		ON	OFF	OFF

Table 3-388. PD_DSS Memory Area Power Modes (continued)

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
	DSS – DSSMEM	always_on	always_off	
	BB2D - BB2D_MEM	always_on	always_off	

3.7.12.1.2 Logic and Memory Area Power Mode Control and Status

Table 3-389 lists the power modes controls for the power domain.

Table 3-389. PD_DSS Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (logic in RETENTION state)	DSS_MEM	PM_DSS_PWRSTCTRL[8] DSS_MEM_RETSTATE	Read only
Power Domain – Low-Power State Change Control		PM_DSS_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area – RETENTION State Control		PM_DSS_PWRSTCTRL[2] LOGICRETSTATE	Read only
Memory Area – State Control (logic in ON state)	DSS_MEM	PM_DSS_PWRSTCTRL[17:16] DSS_MEM_ONSTATE	Read only
Power Domain – State Transition Control		PM_DSS_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-390 lists the status of the power modes for the power domain.

Table 3-390. PD_DSS Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Memory Area – State Status	DSS_MEM	PM_DSS_PWRSTST[5:4] DSS_MEM_STATEST
Power Domain – Last Low Power State Entered Status		PM_DSS_PWRSTST[25:24] LASTPOWERSTATEENTERED
Power Domain – State Transition Status		PM_DSS_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_DSS_PWRSTST[2] LOGICSTATEST
Power Domain – Current Power State Status		PM_DSS_PWRSTST[1:0] POWERSTATEST

3.7.13 PD_CORE Description

PD_CORE contains the following reset domains:

- CM_CORE_PWRON_RET_RST
- CM_CORE_RET_RST
- CORE_PWRON_RET_RST
- CORE_PWRON_RST
- CORE_RET_RST
- CORE_RST
- DLL_RST
- IPU2_PWRON_RST
- IPU2_RET_RST
- IPU2_CPU0_RST
- IPU2_CPU1_RST
- IPU2_RST
- DMA_RET_RST

PD_CORE contains the following clock domains:

- CD_ATL
- CD_DMA
- CD_IPU2
- CD_EMIF

- CD_L3_MAIN1
- CD_L3_INSTR
- CD_L4_CFG

Table 3-391 lists the logic retention capability for each module of the power domain.

Table 3-391. PD_CORE Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
CONTROL_MODULE_CORE	Full	None	None
CONTROL_MODULE_BANDGAP	No	None	None
CM_CORE	Full	None	None
ATL	No	RM_ATL_ATL_CONTEXT[0] LOSTCONTEXT_DFF	None
DLL	No	RM_EMIF_EMIF_DLL_CONTEXT[0] LOSTCONTEXT_DFF	None
DLL_AGING	No	None	None
DMM	Partial	RM_EMIF_DMM_CONTEXT[0] LOSTCONTEXT_DFF	RM_EMIF_DMM_CONTEXT[1] LOSTCONTEXT_RFF
IPU2	Partial	RM_IPU2_IPU2_CONTEXT[0] LOSTCONTEXT_DFF	RM_IPU2_IPU2_CONTEXT[1] LOSTCONTEXT_RFF
EMIF1	Partial	RM_EMIF_EMIF1_CONTEXT[0] LOSTCONTEXT_DFF	RM_EMIF_EMIF1_CONTEXT[1] LOSTCONTEXT_RFF
EMIF2	Partial	RM_EMIF_EMIF2_CONTEXT[0] LOSTCONTEXT_DFF	RM_EMIF_EMIF2_CONTEXT[1] LOSTCONTEXT_RFF
EMIF_OCP_FW	Partial	RM_EMIF_EMIF_OCP_FW_CONTEXT[0] LOSTCONTEXT_DFF	RM_EMIF_EMIF_OCP_FW_CONTEXT[1] LOSTCONTEXT_RFF
GPMC	Full	None	RM_L3MAIN1_GPMC_CONTEXT[1] LOSTCONTEXT_RFF
SPINLOCK	Full	None	RM_L4CFG_SPINLOCK_CONTEXT[1] LOSTCONTEXT_RFF
L3_MAIN_1 interconnect	Partial	RM_L3MAIN1_L3_MAIN_1_CONTEXT[0] LOSTCONTEXT_DFF	RM_L3MAIN1_L3_MAIN_1_CONTEXT[1] LOSTCONTEXT_RFF
L3_MAIN_2 interconnect	Partial	RM_L3INSTR_L3_MAIN_2_CONTEXT[0] LOSTCONTEXT_DFF	RM_L3INSTR_L3_MAIN_2_CONTEXT[1] LOSTCONTEXT_RFF
L3_INSTR interconnect	No	RM_L3INSTR_L3_INSTR_CONTEXT[0] LOSTCONTEXT_DFF	None
L4_CFG interconnect	Partial	RM_L4CFG_L4_CFG_CONTEXT[0] LOSTCONTEXT_DFF	RM_L4CFG_L4_CFG_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX1	Full	None	RM_L4CFG_MAILBOX1_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX2	Full	None	RM_L4CFG_MAILBOX2_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX3	Full	None	RM_L4CFG_MAILBOX3_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX4	Full	None	RM_L4CFG_MAILBOX4_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX5	Full	None	RM_L4CFG_MAILBOX5_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX6	Full	None	RM_L4CFG_MAILBOX6_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX7	Full	None	RM_L4CFG_MAILBOX7_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX8	Full	None	RM_L4CFG_MAILBOX8_CONTEXT[1] LOSTCONTEXT_RFF

Table 3-391. PD_CORE Modules Power Attributes (continued)

Module	Logic Retention	DFF Context Status	RFF Context Status
MAILBOX9	Full	None	RM_L4CFG_MAILBOX9_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX10	Full	None	RM_L4CFG_MAILBOX10_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX11	Full	None	RM_L4CFG_MAILBOX11_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX12	Full	None	RM_L4CFG_MAILBOX12_CONTEXT[1] LOSTCONTEXT_RFF
MAILBOX13	Full	None	RM_L4CFG_MAILBOX13_CONTEXT[1] LOSTCONTEXT_RFF
OCMC_RAM1	No	RM_L3MAIN1_OCMC_RAM1_CONTEXT[0] LOSTCONTEXT_DFF	None
OCMC_RAM2	No	RM_L3MAIN1_OCMC_RAM2_CONTEXT[0] LOSTCONTEXT_DFF	None
OCMC_RAM3	No	RM_L3MAIN1_OCMC_RAM3_CONTEXT[0] LOSTCONTEXT_DFF	None
DMA_SYSTEM	Full	None	RM_DMA_DMA_SYSTEM_CONTEXT[1] LOSTCONTEXT_RFF
OCP2SCP2	No	RM_L4CFG_OCP2SCP2_CONTEXT[0] LOSTCONTEXT_DFF	None
OCP_WP_NOC	Partial	RM_L3INSTR_OCP_WP_NOC_CONTEXT[0] LOSTCONTEXT_DFF	RM_L3INSTR_OCP_WP_NOC_CONTEXT[1] LOSTCONTEXT_RFF
MMU_EDMA	No	None	RM_L3MAIN1_MMU_EDMA_CONTEXT[1] LOSTCONTEXT_RFF
MMU_PCIES	Full	None	RM_L3MAIN1_MMU_PCIESS_CONTEXT[1] LOSTCONTEXT_RFF
TPCC	Full	None	RM_L3MAIN1_TPCC_CONTEXT[1] LOSTCONTEXT_RFF
TPTC1	Full	None	RM_L3MAIN1_TPTC1_CONTEXT[1] LOSTCONTEXT_RFF
TPTC2	Full	None	RM_L3MAIN1_TPTC2_CONTEXT[1] LOSTCONTEXT_RFF
VCP1	No	RM_L3MAIN1_VCP1_CONTEXT[0] LOSTCONTEXT_DFF	None
VCP2	No	RM_L3MAIN1_VCP2_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.13.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.13.1.1 Logic and Memory Area Power Modes

Table 3-392 lists the power modes supported by the logic area of the power domain.

Table 3-392. PD_CORE Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
N/A	Available	Available	Available

Table 3-393 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-393. PD_CORE Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
CORE_NRET_BANK		ON	OFF	
	DMM – DMM_MEMBANK1	always_on	always_retention	
	OCP_WP_NOC – OCPWP_BANK	always_on	always_off	
CORE_OCMRAM		ON	RETENTION	
	OCMC_RAM1 – OCMC_RAM_Bank1	always_on	always_retention	
	OCMC_RAM2 – OCMC_RAM_Bank1	always_on	always_retention	
	OCMC_RAM3 – OCMC_RAM_Bank1	always_on	always_retention	
CORE_OTHER_BANK		ON	RETENTION	
	ATL – ATL_MEM	always_on	always_retention	
	DMA_SYSTEM – DMA_MEM	always_on	always_retention	
	DMM – DMM_MEMBANK2	always_on	always_off	
	TPCC – TPCC_MEM	software_control	software_control	
	TPTC1 – TPTC_MEM	software_control	software_control	
	TPTC2 – TPTC_MEM	software_control	software_control	
	VCP1 - vcp_mem	always_on	always_retention	
	VCP2 - vcp_mem	always_on	always_retention	
IPU_L2RAM		ON	OFF or RETENTION	
	IPU2 – IPU_L2RAM_MEM	always_on	software_control	
IPU_UNICACHE		ON	OFF or RETENTION	
	IPU2 – IPU_UNICACHE_mem	always_on	software_control	

3.7.13.1.2 Logic and Memory Area Power Mode Control and Status

Table 3-394 lists the power mode controls for the power domain.

Table 3-394. PD_CORE Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (Logic in RETENTION state)	CORE_OCMRAM	PM_CORE_PWRSTCTRL[9] CORE_OCMRAM_RETSTATE	Read only
Memory Area – State Control (Logic in RETENTION state)	CORE_OTHER_BANK	PM_CORE_PWRSTCTRL[8] CORE_OTHER_BANK_RETSTATE	Read only
Memory Area – State Control (Logic in ON state)	CORE_NRET_BANK	PM_CORE_PWRSTCTRL[25:24] OCP_NRET_BANK_ONSTATE	Read only
Memory Area – State Control (Logic in ON state)	IPU_UNICACHE	PM_CORE_PWRSTCTRL[23:22] IPU_UNICACHE_ONSTATE	Read only
Memory Area – State Control (Logic in ON state)	IPU_L2RAM	PM_CORE_PWRSTCTRL[21:20] IPU_L2RAM_ONSTATE	Read only
Memory Area – State Control (Logic in ON state)	CORE_OCMRAM	PM_CORE_PWRSTCTRL[19:18] CORE_OCMRAM_ONSTATE	Read only

Table 3-394. PD_CORE Power Modes Control Parameters (continued)

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (Logic in ON state)	CORE_OTHER_BANK	PM_CORE_PWRSTCTRL[17:16] CORE_OTHER_BANK_ONSTATE	Read only
Memory Area – State Control (Logic in RETENTION state)	CORE_NRET_BANK	PM_CORE_PWRSTCTRL[12] OCP_NRET_BANK_RETSTATE	Read only
Memory Area – State Control (Logic in RETENTION state)	IPU_UNICACHE	PM_CORE_PWRSTCTRL[11] IPU_UNICACHE_RETSTATE	Read/write
Memory Area – State Control (Logic in RETENTION state)	IPU_L2RAM	PM_CORE_PWRSTCTRL[10] IPU_L2RAM_RETSTATE	Read/write
Logic Area – RETENTION State Control	N/A	PM_CORE_PWRSTCTRL[2] LOGICRETSTATE	Read/write
Power Domain – Low-Power State Change Control	N/A	PM_CORE_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Power Domain – State Transition Control	N/A	PM_CORE_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-395 lists the status of the power modes for the power domain.

Table 3-395. PD_CORE Power Mode Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_CORE_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	IPU_L2RAM	PM_CORE_PWRSTST[9:8] IPU_L2RAM_STATEST
Memory Area – State Status	CORE_OCMRAM	PM_CORE_PWRSTST[7:6] CORE_OCMRAM_STATEST
Memory Area – State Status	CORE_OTHER_BANK	PM_CORE_PWRSTST[5:4] CORE_OTHER_BANK_STATEST
Memory Area – State Status	CORE_NRET_BANK	PM_CORE_PWRSTST[13:12] OCP_NRET_BANK_STATEST
Memory Area – State Status	IPU_UNICACHE	PM_CORE_PWRSTST[11:10] IPU_UNICACHE_STATEST
Logic Area – State Status		PM_CORE_PWRSTST[2] LOGICSTATEST
Power Domain – State Transition Status		PM_CORE_PWRSTST[20] INTRANSITION
Power Domain – State Status		PM_CORE_PWRSTST[1:0] POWERSTATEST

3.7.14 PD_CAM Description

PD_CAM contains the CAM_RST reset domain.

PD_CAM contains the CD_CAM clock domain.

Table 3-396 lists the logic retention capability for each module of the power domain.

Table 3-396. PD_CAM Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
VIP1	No	RM_CAM_VIP1_CONTEXT[0] LOSTCONTEXT_DFF	None
VIP2	No	RM_CAM_VIP2_CONTEXT[0] LOSTCONTEXT_DFF	None
VIP3	No	RM_CAM_VIP3_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.14.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.14.1.1 Logic and Memory Area Power Modes

Table 3-397 lists the power modes supported by the logic area of the power domain.

Table 3-397. PD_CAM Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Available	Available

Table 3-398 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-398. PD_CAM Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention
VIP_BANK		ON	OFF
	VIP1 – VIP_MEM	always_on	always_retention
	VIP2 – VIP_MEM	always_on	always_retention
	VIP3 – VIP_MEM	always_on	always_retention

3.7.14.1.2 Logic and Memory Area Power Mode Control and Status

Table 3-399 lists the power mode controls for the power domain.

Table 3-399. PD_CAM Power Mode Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain – Low-Power State Change Control		PM_CAM_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area – State Control (Logic in ON state)	VIP_BANK	PM_CAM_PWRSTCTRL[17:16] VIP_BANK_ONSTATE	Read only
Power Domain – State Transition Control		PM_CAM_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-400 lists the status of the power modes for the power domain.

Table 3-400. PD_CAM Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Memory Area – State Status	VIP_BANK	PM_CAM_PWRSTST[5:4] VIP_BANK_STATEST
Power Domain – Last Power State Entered Status		PM_CAM_PWRSTST[25:24] LASTPOWERSTATEENTERED
Power Domain – State Transition Status		PM_CAM_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_CAM_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_CAM_PWRSTST[1:0] POWERSTATEST

3.7.15 PD_MPUAON Description

PD_MPUAON contains the following reset domains:

- MPUON_RST

- DPLL_MPU_PWRON_RST

PD_MPUAON has no associated clock domains.

Table 3-401 lists the logic retention capability for each module of the power domain.

Table 3-401. PD_MPUAON Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DPLL_MPU	No	None	None

3.7.15.1 Power Domain Modes

The PD_MPUAON power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power-state control or status bit field for this power domain.

The PD_MPUAON power domain has no memory banks.

3.7.16 PD_MMAON Description

PD_MMAON contains the following reset domains:

- MMAON_RST
- DPLL_DSP_PWRON_RST
- DPLL_EVE_PWRON_RST

PD_MMAON has no associated clock domains.

Table 3-402 lists the logic retention capability for each module of the power domain.

Table 3-402. PD_MMAON Module Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DPLL_DSP	No	None	None
DPLL_EVE	No	None	None

3.7.16.1 Power Domain Modes

The PD_MMAON power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power-state control or status bit field for this power domain.

The PD_MMAON power domain has no memory banks.

3.7.17 PD_COREAON Description

PD_COREAON contains the following reset domains:

- CM_CORE_AON_PWRON_RST
- CM_CORE_AON_RST
- COREAON_PWRON_RST
- COREAON_RST
- DPLL_IVA_PWRON_RST

PD_COREAON contains the CD_COREAON_L4 clock domain.

Table 3-403 lists the logic retention capability for each module of the power domain.

Table 3-403. PD_COREAON Module Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
CM_CORE_AON	No	None	None
DPLL_ABE	No	None	None
DPLL_CORE	No	None	None
DPLL_PER	No	None	None
DPLL_DDR	No	None	None

Table 3-403. PD_COREAON Module Power Attributes (continued)

Module	Logic Retention	DFF Context Status	RFF Context Status
DPLL_GMAC	No	None	None
DPLL_GPU	No	None	None
DPLL_IVA	No	None	None
DPLL_PCIE_REF	No	None	None
APLL_PCIE	No	None	None
DPLL_USB	No	None	None
WUGEN_IPU	No	None	None
WUGEN_DMA_SYSTEM	No	None	None
SPINNER	No	None	None

3.7.17.1 Power Domain Modes

The PD_COREAON power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power-state control or status bit field for this power domain.

The PD_COREAON power domain has no memory banks.

3.7.18 PD_VPE Description

PD_VPE contains the following reset domains:

- VPE_RST

PD_VPE contains the CD_VPE clock domain.

Table 3-404 lists the logic retention capability for each module of the power domain.

Table 3-404. PD_VPE Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
VPE	No	RM_VPE_VPE_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.18.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.18.1.1 Logic and Memory Area Power Modes

Table 3-405 lists the power modes supported by the logic area of the power domain.

Table 3-405. PD_VPE Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available

Table 3-406 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-406. PD_VPE Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
VPE_BANK		ON	OFF or RETENTION	OFF
	VPE – VPE_MEM	always_on	always_retention	

3.7.18.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-407 lists the power mode controls for the power domain.

Table 3-407. PD_VPE Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (Logic in ON state)	VPE_BANK	PM_VPE_PWRSTCTRL[17:16] VPE_BANK_ONSTATE	Read only
Memory Area – State Control (logic in RETENTION state)	VPE_BANK	PM_VPE_PWRSTCTRL[8] VPE_BANK_RETSTATE	Read/write
Power Domain – Low-Power State Change Control		PM_VPE_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area – Retention State Control		PM_VPE_PWRSTCTRL[2] LOGICRETSTATE	Read/write
Power Domain – State Transition Control		PM_VPE_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-408 lists the status of the power modes for the power domain.

Table 3-408. PD_VPE Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_VPE_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	VPE_BANK	PM_VPE_PWRSTST[5:4] VPE_BANK_STATEST
Power Domain – State Transition Status		PM_VPE_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_VPE_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_VPE_PWRSTST[1:0] POWERSTATEST

3.7.19 PD_EVE1 Description

PD_EVE1 contains the following reset domains:

- EVE1_CPU_RST
- EVE1_PWRON_RST
- EVE1_RST

PD_EVE1 contains the CD_EVE1 clock domain.

Table 3-409 lists the logic retention capability for each module of the power domain.

Table 3-409. PD_EVE1 Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
EVE1	No	RM_EVE1_EVE1_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.19.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.19.1.1 Logic and Memory Area Power Modes

Table 3-410 lists the power modes supported by the logic area of the power domain.

Table 3-410. PD_EVE1 Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-411 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-411. PD_EVE1 Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
EVE1_BANK		ON	OFF or RETENTION	OFF
	EVE1 – EVE_BANK	always_on	always_off	

3.7.19.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-412 lists the power mode controls for the power domain.

Table 3-412. PD_EVE1 Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (Logic in ON state)	EVE1_BANK	PM_EVE1_PWRSTCTRL[17:16] EVE1_BANK_ONSTATE	Read only
Power Domain – Low-Power State Change Control		PM_EVE1_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Power Domain – State Transition Control		PM_EVE1_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-413 lists the status of the power modes for the power domain.

Table 3-413. PD_EVE1 Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_EVE1_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	EVE1_BANK	PM_EVE1_PWRSTST[5:4] EVE1_BANK_STATEST
Power Domain – State Transition Status		PM_EVE1_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_EVE1_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_EVE1_PWRSTST[1:0] POWERSTATEST

3.7.20 PD_EVE2 Description

PD_EVE2 contains the following reset domains:

- EVE2_CPU_RST
- EVE2_PWRON_RST
- EVE2_RST

PD_EVE2 contains the CD_EVE2 clock domain.

Table 3-414 lists the logic retention capability for each module of the power domain.

Table 3-414. PD_EVE2 Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
EVE2	No	RM_EVE2_EVE2_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.20.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.20.1.1 Logic and Memory Area Power Modes

Table 3-415 lists the power modes supported by the logic area of the power domain.

Table 3-415. PD_EVE2 Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-416 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module – Memory column).

Table 3-416. PD_EVE2 Memory Area Power Modes

Memory Bank	Module – Memory	Logic On	Logic Retention	Logic Off
EVE2_BANK		ON	OFF or RETENTION	OFF
	EVE2 – EVE_BANK	always_on	always_off	

3.7.20.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-417 lists the power mode controls for the power domain.

Table 3-417. PD_EVE2 Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area – State Control (Logic in ON state)	EVE2_BANK	PM_EVE2_PWRSTCTRL[17:16] EVE2_BANK_ONSTATE	Read only
Power Domain – Low-Power State Change Control		PM_EVE2_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Power Domain – State Transition Control		PM_EVE2_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-418 lists the status of the power modes for the power domain.

Table 3-418. PD_EVE2 Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain – Last Power State Entered Status		PM_EVE2_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area – State Status	EVE2_BANK	PM_EVE2_PWRSTST[5:4] EVE2_BANK_STATEST
Power Domain – State Transition Status		PM_EVE2_PWRSTST[20] INTRANSITION
Logic Area – State Status		PM_EVE2_PWRSTST[2] LOGICSTATEST
Power Domain – State Status		PM_EVE2_PWRSTST[1:0] POWERSTATEST

3.7.21 PD_RTC Description

PD_RTC contains the following reset domains:

- RTC_RST

PD_RTC contains the CD_RTC clock domain.

Table 3-419 lists the logic retention capability for each module of the power domain.

Table 3-419. PD_RTC Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
RTC_SS	No	RM_RTC_RTCSS_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.21.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.21.1.1 Logic and Memory Area Power Modes

Table 3-420 lists the power modes supported by the logic area of the power domain.

Table 3-420. PD_RTC Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
N/A	N/A	N/A	Available

There is no memory bank implemented for the PD_RTC.

3.8 Voltage-Management Functional Description

This section describes the voltage domains and voltage control architecture. It also explains the interactions between the device and the external power IC.

3.8.1 Overview

The voltage-management architecture of the device is based on voltage sources managed by the PRCM module. They define the voltage domains within the device (see Section 3.1.1.3, *Voltage Management*). This partition of the voltage domains ensures independent voltage control of each voltage domain through dedicated LDO. The following voltage domains are managed by the PRCM module:

- VDD_CORE_L
- VDD_MPU_L
- VDD_IVAHD_L
- VDD_DSPEVE_L
- VDD_GPU_L
- VDD_RTC_L

Note

For the association of the device power supply pin to the power domain, see Table 3-26.

The PRCM module supports the AVS technique on the VDD_MPU_L, VDD_CORE_L, and VDD_IVAHD_L, VDD_DSPEVE_L, VDD_GPU_L, VDD_RTC_L voltage domains.

The PRCM module also supports the ABB technique on the MPU, IVAHD, DSPEVE and GPU voltage domains through the VDD_MPU_ABB and VDD_IVA_ABB, VDD_DSPEVE_ABB, VDD_GPU_ABB biasing voltages.

At boot time, the device is set with all five voltage domains (VDD_MPU_L, VDD_GPU_L, VDD_CORE_L, VDD_DSPEVE_L, VDD_IVAHD_L) at OPP_NOM.

3.8.2 Voltage-Control Architecture

The PRM is split over several blocks that manage the different voltage sources.

- A device PRCM for managing the I/O wake-up control and system clock control sequencing during device sleep and wake-up transitions.
- LDO regulator controllers for ABB management, memory arrays voltage management, and WAKEUP logic.
- BANDGAPs reference voltage sleep control

Figure 3-95 shows the architecture for PRM voltage control. This figure represents an example in which the I^2C interface is used to control the PMIC.

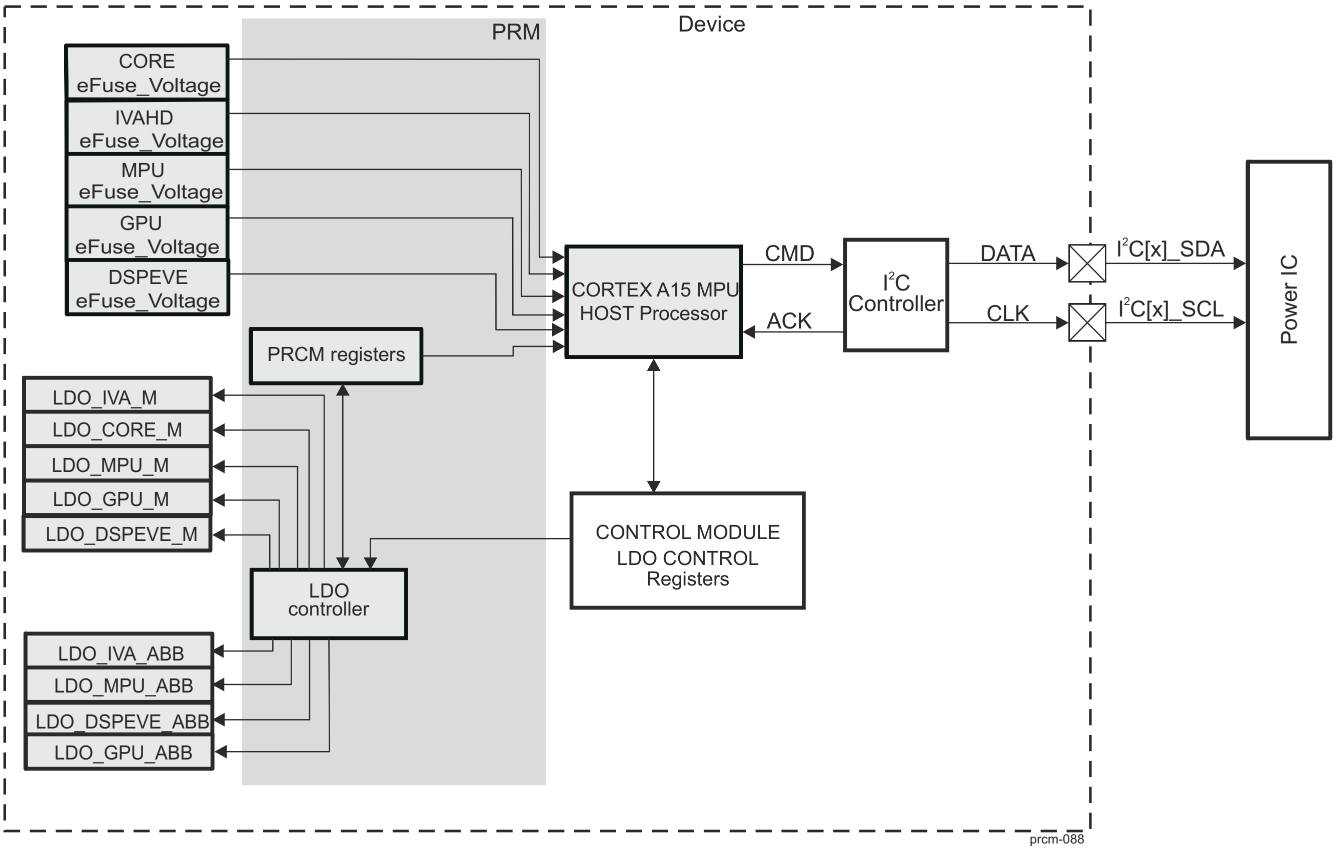


Figure 3-95. PRM Voltage Control Architecture

3.8.3 Internal LDOs Control

3.8.3.1 VDD_MPU_L, VDD_CORE_L, and VDD_IVAHD_L, VDD_GPU_L, VDD_DSPEVE_L Control

3.8.3.1.1 Adaptive Voltage Scaling

As explained in Section 3.1.2.4, *Adaptive Voltage Scaling* with the SmartReflex technology the power supply voltage can be adapted to the silicon performance statically (for example, adapted to the manufacturing process of a given device).

3.8.3.1.1.1 SmartReflex in the Device

Figure 3-96 shows the SmartReflex integration.

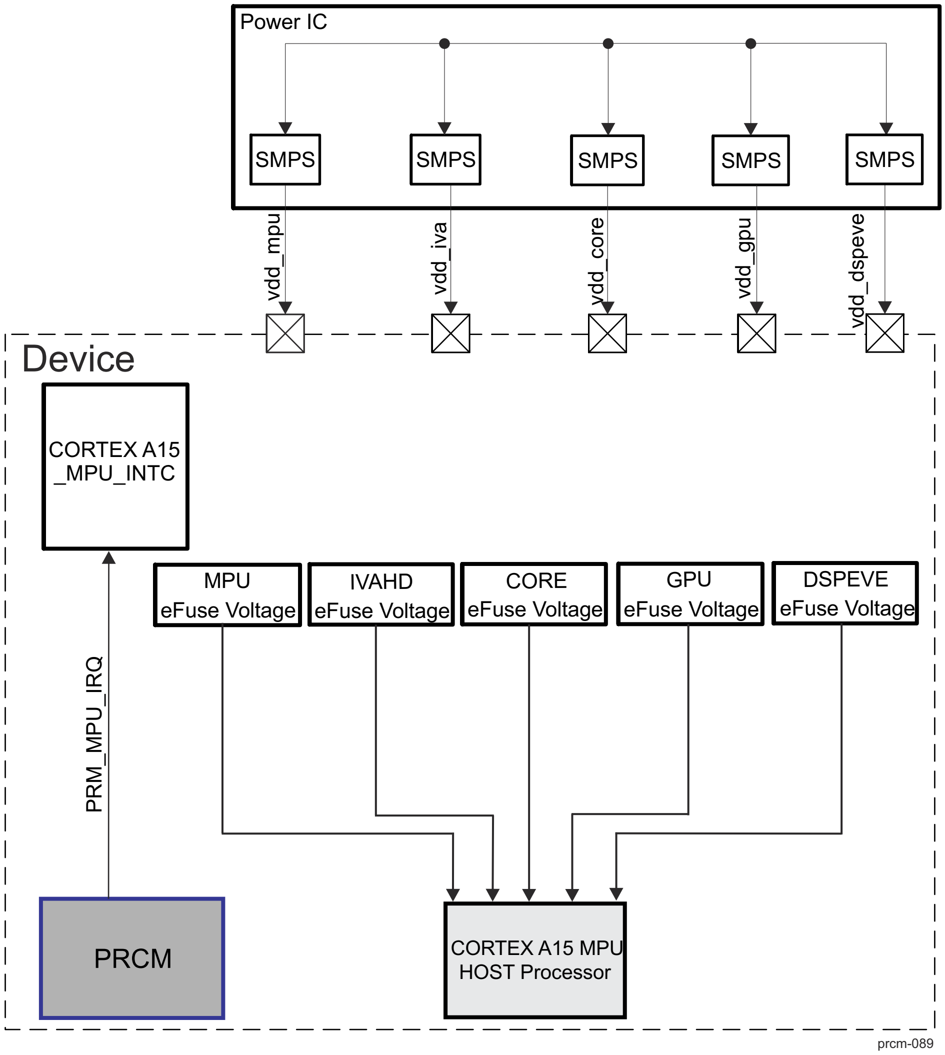


Figure 3-96. SmartReflex Integration

3.8.3.2 Memory LDOs

Embedded SRAM LDOs are used to supply power to the split-rail memory arrays. The PRM generates the controls used to select LDO operating mode: on-active, on-retention, or off.

Split-rail type SRAMs are used in the device to implement the larger memories. These SRAMs feature memory array and periphery logic, which are on separate supplies to allow independent power management. Proper memory operation, however, requires the SRAM array voltage never to be operated at a level lower than the SRAM periphery logic.

Memory LDO can switch to on and retention mode:

- On (active) mode: 1.15 V is the normal voltage reference used through all functional OPPs whenever memories must be functional. When logic voltage level VDD_x_L (where x can be MPU, CORE, or IVAHD) becomes higher than the associated memory voltage level VDD_x_M, the LDO operates in tracking mode and follows its respective VDD_x_M or VDD_x_ABB voltage level.
- Retention mode: 0.6 V is set when software allows and when all memory banks belonging to the LDO memory voltage are in RETENTION state. In this mode, the output voltage is generated from the corresponding VDD_WKUP_L logic voltage source.

Memory LDOs Transitions describes the state transition conditions and sequences for the memory LDOs.

Note

The voltage levels associated with the different modes may depend on the device characteristics.

3.8.3.3 ABB LDOs Control

The ABBLDO supports three voltage modes:

- Bypass mode: In this mode, the ABBLDO_x is bypassed and outputs the VDD_x_L voltages (x refers to the MPU and IVA). This mode is activated when FBB is not required, or when the voltage domain enters low-power mode.
- FBB mode is enabled when the device is a weak process device at the highest OPP.

The PRCM module provides the PRM_ABBLDO_MPU_SETUP and PRM_ABBLDO_GPU_SETUP registers for configuration with the following controls:

- SR2EN: To enable or bypass the ABB power management
- ACTIVE_FBB_SEL: To enable or bypass FBB mode
- SR2_WTCNT_VALUE: LDO settling delay on OPP change. The delay is in the number of system clock cycles.

The PRCM module provides the PRM_ABBLDO_MPU_CTRL and PRM_ABBLDO_GPU_CTRL registers for control:

- OPP_SEL: Current operational OPP
- OPP_CHANGE: Initiate an OPP-based ABBLDO setting change
- SR2_STATUS°: Current mode of operation of ABBLDO
- SR2_IN_TRANSITION: ABBLDO in transition

3.8.3.4 ABB LDO Programming Sequence

Anytime the user enters or leaves one of the ABB required OPPs, the appropriate procedure must be followed to enable or disable the ABB LDO.

3.8.3.4.1 ABB LDO Enable Sequence

The following steps turn on the ABB LDO after the voltage change when entering an OPP that requires ABB (FBB):

1. If set, clear the status of the PRCM interrupt for the ABB LDO transition completion:
   - The PRM_IRQSTATUS_MPU[x] ABB<Voltage_Domain_Name>_DONE_ST bit should be set to 0x1.
2. Based on which ABB voltage mode (Bypass or FBB) is selected for the OPP (single write), configure the ABB transition:
   - To enter in Forward Body Bias (FBB) mode, set the PRM_ABBLDO_<Voltage_Domain_Name>_SETUP[2] ACTIVE_FBB_SEL bit to 0x1.
3. The user should set the PRM_ABBLDO_<Voltage_Domain_Name>_CTRL[2] OPP_ CHANGE bit to 0x0.
4. PRM_ABBLDO_MPU_CTRL[1:0] OPP_SEL bits should be set to 0x1, which select ‘Fast’ or Forward Body Bias mode.
5. The user should set the ABB voltage value in CTRL_<CORE|WKUP>_LDOVBB_<Voltage_ Domain_Name>_VOLTAGE_CTRL[4:0] LDOVBB<Voltage_Domain_Name>_FBB_VSET_ OUT.
   - To use the override value, set the CTRL_<CORE|WKUP>_LDOVBB_<Voltage_ Domain_Name>_VOLTAGE_CTRL[10] LDOVBB<Voltage_Domain_Name>_FBB_MUX_ CTRL bit to 0x1.
6. To enable the ABB LDO OPP change, set the PRM_ABBLDO_<Voltage_ Domain_Name>_CTRL[2] OPP_CHANGE bit to 0x1.
   - Wait until the transaction is complete: the PRM_IRQSTATUS_MPU[x] ABB_<Voltage_Domain _Name>_DONE_ST bit should be set to 0x1.
   - Clear interrupt status: the PRM_IRQSTATUS_MPU[x] ABB_<Voltage_Domain _Name>_ DONE_ ST bit should be set to 0x0.

3.8.3.4.2 ABB LDO Disable Sequence (Entering in Bypass Mode)

The following steps turn off the ABB LDO after the voltage change when entering an OPP that requires ABB (FBB):

1. If set, clear the status of the PRCM interrupt for the ABB LDO transition completion:
   - The PRM_IRQSTATUS_MPU[x] ABB<Voltage_Domain_Name>_DONE_ST bit should be set to 0x1.
2. Based on which ABB voltage mode (Bypass) is selected for the OPP (single write), configure the ABB transition:
   - To enter nominal (Bypass) mode, set the PRM_ABBLDO_<Voltage _Domain_Name>_SETUP[2] ACTIVE_FBB_SEL bit to 0x0.
3. The user should set the PRM_ABBLDO_<Voltage_Domain_Name>_CTRL[2] OPP_ CHANGE bit to 0x0.
4. PRM_ABBLDO_MPU_CTRL[1:0] OPP_SEL bits should be set to 0x0, which select Bypass mode.
5. To enable the ABB LDO OPP change, set the PRM_ABBLDO_<Voltage_ Domain_Name>_CTRL[2] OPP_CHANGE bit to 0x1.
   - Wait until the transaction is complete: the PRM_IRQSTATUS_MPU[x] ABB_<Voltage_Domain _Name>_DONE_ST bit should be set to 0x1.
   - Clear interrupt status: the PRM_IRQSTATUS_MPU[x] ABB_<Voltage_Domain _Name>_DONE_ ST bit should be set to 0x1.
6. The user should set ABB voltage value in CTRL_<CORE|WKUP>_LDOVBB_<Voltage _Domain_Name>_VOLTAGE_CTRL[4:0] LDOVBB<Voltage_Domain_Name>_VSET_OUT to 0x0.

To use the eFuse voltage value, the user should set the CTRL_CORE|WKUP>_LDOVBB _<Voltage_Domain_Name>_VOLTAGE_CTRL[10] LDOVBB<Voltage_Domain_Name>_FBB_MUX_ CTRL bit to 0x0.

3.8.3.5 BANDGAPs Control

BANDGAPs provides voltage reference for internal LDOs. The PRCM module automatically controls the switching between ON and OFF states of the BANDGAPs, based on the power state of the device. It is completely transparent to user software.

BANDGAPs startup time is 100 μs. The PRM_BANDGAP_SETUP[7:0] STARTUP_COUNT bit field must be set accordingly.

3.8.4 DVFS

Dynamic voltage and frequency scaling is a technique that can be used on the logic voltage domains (VDD_MPU_L, VDD_IVAHD_L, VDD_DSPEVE_L and VDD_GPU_L), independently of one another. Upon a current or predictive performance request, determined by software according to ad hoc algorithms or heuristics, software can configure the PRCM module to change the OPP of a voltage domain. For an increase in performance, the voltage is first raised, and then the frequency is increased. For a decrease in performance, the frequency is first decreased, and then the voltage is dropped.

3.9 Device Low-Power States

The device low-power states are the result of any valid combination of power domain states in which all the power domains are no longer in ACTIVE state. In such a situation the PRCM module hardware can trigger events to further lower the consumption of the device and the system.

These device low-power states are characterized by the system power consumption, wake-up latency, and required functionality.

The low-power states are:

- "**RTC mode**" : All logic voltage domains (VD_CORE, VD_MPU, VD_GPU, VD_IVAHD, VD_DSPEVE) are into OFF state except for the VD_RTC domain which is ON.

Note

The VD_RTC remains always ON regardless of the low power state. RTC_SS is the only wake-up event source in the lowest power mode - "RTC mode".

- "**STANDBY**" : Any combination of logic voltage domain states (SLEEP or RETENTION) other than ACTIVE state.

For more details on the wakeup-sources from the "RTC mode" and the "STANDBY" low-power modes, refer to the Table 3-421.

Once the PRCM module hardware detects any valid combination of power domain states, and if a proper programming model of the PRCM module is set, the PRCM module automatically triggers the transition into the device low-power mode.

3.9.1 Device Wake-Up Source Summary

The wake-up events can be asynchronous or synchronous. Synchronous wake-up events require the 32-kHz clock or the system clock to be active, while asynchronous wake-up events do not require an active clock.

The *Modules Attributes* subsection of each clock domain in Section 3.6, *Clock Management Functional Description*, describes the wake-up capability support for each module of the corresponding power domain.

While the device is in STANDBY mode, additional asynchronous wakeup events from other domains are able to wake up the device.

Table 3-421 identifies which modules in which power domains can be configured to generate a wake-up while the device is in a low-power mode.

Table 3-421. Wake-Up Sources During Device Low Power Mode

Device Power Mode name	VD_RTC VOLTAGE STATE	VD_CORE VOLTAGE STATE	VD_MPU VOLTAGE STATE	VD_GPU VOLTAGE STATE	VD_IVAHD VOLTAGE STATE	VD_DSPEVE VOLTAGE STATE	Domain containing wake-up source	Wakeup sources
Really OFF	OFF	OFF	OFF	OFF	OFF	OFF	N/A	Application of Power Supply and Power on Reset Sequence
RTC Mode	ON	OFF	OFF	OFF	OFF	OFF	RTC SS sends signal to power supply to apply power voltage to all other voltage domains	RTC ("alarm" and "timer" slave wake-up capabilities), EXT_WKUP0, EXT_WKUP1, EXT_WKUP2, EXT_WKUP3
STANDBY		Active Voltage Level					PD_IPU (1)	I2C5, IPU1, McASP1, TIMER5, TIMER6, TIMER7, TIMER8, UART6
							PD_CORE	DMA_SYSTEM, IPU2, OCMC_RAM1, OCMC_RAM2, OCMC_RAM3, EDMA.TPCC, EDMA.TPTC1, EDMA.TPTC2
							PD_L3INIT (1)	IEEE1500_2_OCP, MLB_SS, MMC1, MMC2, PCIe_SS1, PCIe_SS2, SATA, USB1, USB2, USB3, USB4
							PD_L4PER	DCAN2, GPIO2 - GPIO8, I2C1 - I2C4, McASP2 - McASP8, McSPI1 - McSPI4, MMC3, MMC4, QSPI, TIMER2 - TIMER4, TIMER9 - TIMER11, TIMER13 - TIMER16, UART1 - UART9
							PD_MPU	MPU
							PD_DSP1	DSP1
							PD_DSP2	DSP2
							PD_EVE1	EVE1
							PD_EVE2	EVE2
							PD_WKUPAON	DCAN1, GPIO1, KBD, TIMER1, TIMER12, UART10,WD_TIMER2

Table 3-421. Wake-Up Sources During Device Low Power Mode (continued)

Device Power Mode name	VD_RTC VOLTAGE STATE	VD_CORE VOLTAGE STATE	VD_MPU VOLTAGE STATE	VD_GPU VOLTAGE STATE	VD_IVAHD VOLTAGE STATE	VD_DSPEVE VOLTAGE STATE	Domain containing wake-up source	Wakeup sources
							PD_EMU	(Debug_logic) Any ForceActive directive Dynamic depedency towards L3_MAIN

(1) Only modules able to generate asynchronous wake-up have to be taken into account, and only when domain is in INACT or CSWRET state.

3.9.2 Wakeup Upon Global Warm Reset

When global warm reset is the source of the device wakeup, the sequence is modified as follows:

- The PRCM module releases its reset line. In parallel, the device reset manager counts for global reset extension (set up by the PRM_RSTTIME[9:0] RSTTIME1 bit field). The PRCM module holds the other asserted resets until the global reset counter overflows.

The hardware blocks and modifies the CM_SHADOW_FREQ_CONFIG1[0] FREQ_UPDATE and CM_MPU_CLKSTCTRL[1:0] CLKTRCTRL bit fields.

3.9.3 Global Warm Reset During a Device Wake-Up Sequence

If a global warm reset occurs before the PRCM module completes the voltage stabilization count, the global warm reset is applied immediately. As a consequence, the sequence described in Section 3.9.2, *Wakeup Upon Global Warm Reset*, is performed.

If the global warm reset occurs after the PRCM module completes voltage stabilization (during phase 1 of automatic restore):

1. Global warm reset is delayed and applied after phase 1 restore completes.
2. Phase 2 of restore is discarded.
3. MPU boots after the warm reset sequence completes.

If the global warm reset occurs during phase 2 of automatic restore:

1. Global warm reset is applied immediately.
2. Phase 2 of restore is discarded.
3. MPU boots after the warm reset sequence completes.

3.9.4 I/O Management

Figure 3-97 is an overview of the power-management modules and their internal connections with a generic power domain.

ISOIN, ISOCLKIN, WUCLKIN, WUIN, ISOOUT, ISOCLKOUT, WUCLKOUT and WUOUT connections are built with daisy chaining approach and are all VDD_WKUP_L level signals driven by PRM. WUEN and WUEVNT are VDD level signals driven by the control module and/or IO control module. ISOOVR and ISOBYPASS (for DDR IO only) are VDD level signals driven by the PRM.

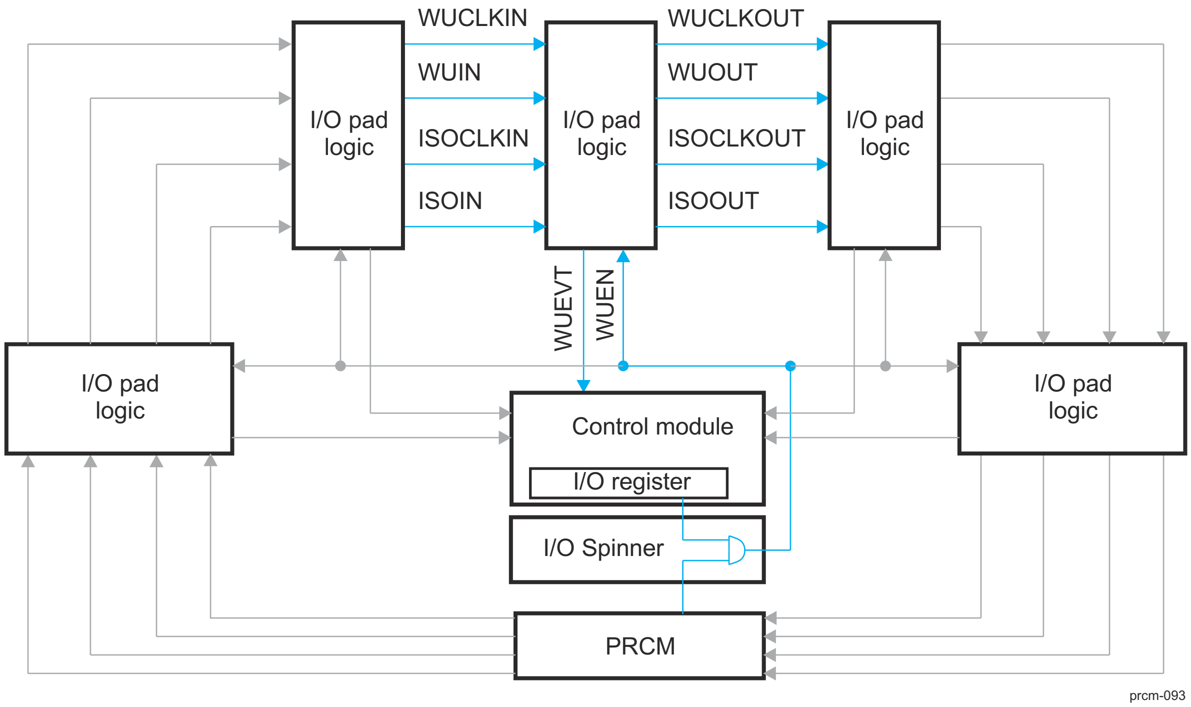


Figure 3-97. I/O Pads Daisy-Chain Configuration

Note

ISOBYPASS control from PRM is routed to all the EMIF i/f IOs and ISOOVR control from PRM is routed to all the remaining IOs. These control signals are routed using ALWON VDD buffers. The ISOOVR pin on EMIF IOs is tied to 0 using ALWON tie-off cells. PRM also provides an output status "IO_ISO_ACTIVE" which is set when EMIF IOs are isolated or transitioning between isolation and functional mode, and which is cleared when EMIF IOs are functional.

---

it is required that the ISOCLKIN is forced to 1 during global power-on reset (PRM_PWRON_RST_n = 0).

3.9.4.1 Isolation / Wakeup Sequence

Enabling Wake-Up Feature:

- Program Control module MMR to assert "WKEN" for each IO (To enable Wakeup feature of IO)
- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 1 to assert high the signal WUCLKIN.
- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 0 to assert low the signal WUCLKIN.
- This will latch WKEN, Latch the current pad input value.
- The PRCM register PRM_IO_PMCTRL[9] WUCLK_STATUS logs the signal WUCLK of the last pad of the IO ring.(Should be 0).

Device goes into sleep mode (There is no need to put IO in ISOLATION and hence no need to toggle ISOCLKIN and ISOIN (As core supply is still ON)) :

- WKUP event is generated by one of the IOs
- WUOUT of the last IO is asserted HIGH.
- Because of #9, PRM interrupt is generated towards MPU/Host processor
- MPU/Host processor disables the WUKP feature of each IO and power up the required domains.

DISABLING WKUP feature:

- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 1 to assert high the signal WUCLKIN.
- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 0 to assert low the signal WUCLKIN.

- The PRCM register PRM_IO_PMCTRL[9] WUCLK_STATUS logs the signal WUOUT of the last pad of the IO ring.

3.9.4.1.1 Software-Controlled I/O Isolation

The PRM_IO_PMCTRL[4] ISOOVR_EXTEND bit allows extending the non-EMIF I/O isolation. This feature can be used by software to restore modules driving output, such as GPIO, while non-EMIF I/Os are still isolated. Once software completes the relevant module restore, it clears the bit and hardware performs full-isolation-to-EMIF on the hardware-controlled I/O transition.

The PRM_IO_PMCTRL[5] IO_ON_STATUS bit is available for software to check completion of the EMIF on transition.

3.10 PRCM Module Programming Guide

3.10.1 DPLLs Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

3.10.1.1 Global Initialization

3.10.1.1.1 Surrounding Module Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DPLLs.

Table 3-422 describes the global initialization of the surrounding modules.

Table 3-422. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Ensure that the DPLL reference clock (gated version of system clock) is active.

3.10.1.1.2 DPLL Global Initialization

3.10.1.1.2.1 Main Sequence – DPLL Global Initialization

This procedure initializes the DPLL after a POR or software reset and then locks it to the desired synthesized clock frequency.

Table 3-423. DPLL Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure recalibration parameters.	See Section 3.10.1.1.2.2.	
Set DPLL automatic idle mode.	CM_AUTOIDLE_<DPLL name>[2:0] AUTO_DPLL_MODE	xx(1)
Configure synthesized clock parameters.	See Section 3.10.1.1.2.3.	
Configure output clocks parameters.	See Section 3.10.1.1.2.4.	
Lock DPLL.	CM_CLKMODE_<DPLL name>[2:0] DPLL_EN	0x7

(1) It depends on the desired auto idle mode. See Section 3.6.3.3.4, *DPLL Power Modes*.

3.10.1.1.2.2 Subsequence – Recalibration Parameter Configuration

This procedure enables the recalibration feature and the associated processor interrupt flag.

Table 3-424. DPLL Recalibration Parameter Configuration

Step	Register/Bit Field/Programming Model	Value
Clear recalibration interrupt status.	PRM_IRQSTATUS _<Processor name>[x] <DPLL name>_RECAL_ST	0x0
Unmask recalibration interrupt flag.	PRM_IRQENABLE _<Processor name>[x] <DPLL name>_RECAL_EN	0x1
Enable recalibration feature.	CM_CLKMODE_<DPLL name>[8] DPLL_DRIFTGUARD_EN	0x1

3.10.1.1.2.3 Subsequence – Synthesized Clock Parameter Configuration

This procedure configures the settings for the synthesized clock of the DPLL.

Table 3-425. DPLL Synthesized Clock Parameter Configuration

Step	Register/Bit Field/Programming Model	Value
Set DPLL clock synthesis multiplier.	CM_CLKSEL_<DPLL name>[18:8] DPLL_MULT	xx(1)
Set DPLL clock synthesis divider.	CM_CLKSEL_<DPLL name>[6:0] DPLL_DIV	xx(1)
IF : Low-power mode operation conditions satisfied?	Software test condition. See Section 3.6.3.3.3.	

Table 3-425. DPLL Synthesized Clock Parameter Configuration (continued)

Step	Register/Bit Field/Programming Model	Value
Enable DPLL low-power operation mode.	CM_CLKMODE_<DPLL name>[10] DPLL_LPMODE_EN	0x1
ENDIF		

(1) It depends on the desired synthesized clock frequency. See notes-list-normal, *Generic DPLL Overview*.

3.10.1.1.2.4 Subsequence – Output Clock Parameter Configuration

This procedure configures the settings for the output clocks of the DPLL.

Table 3-426. DPLL Output Clock Parameter Configuration

Step	Register/Bit Field/Programming Model	Value
Set output clock dividers (that is, M2, M3, and Hmn), where m is 1 or 2, and n is from 1 to 4. It depends on the available clock output of the DPLL.	CM_DIV_M2_<DPLL name>[4:0] DIVHS CM_DIV_M3_<DPLL name>[4:0] DIVHS CM_DIV_Hmn_<DPLL name>[5:0] DIVHS	xx(1)

(1) It depends on the desired output clock frequency. See Section 3.6.3.3, *Generic DPLL Overview*.

3.10.1.2 DPLL Output Frequency Change

Figure 3-98 shows the DPLL output-frequency change.

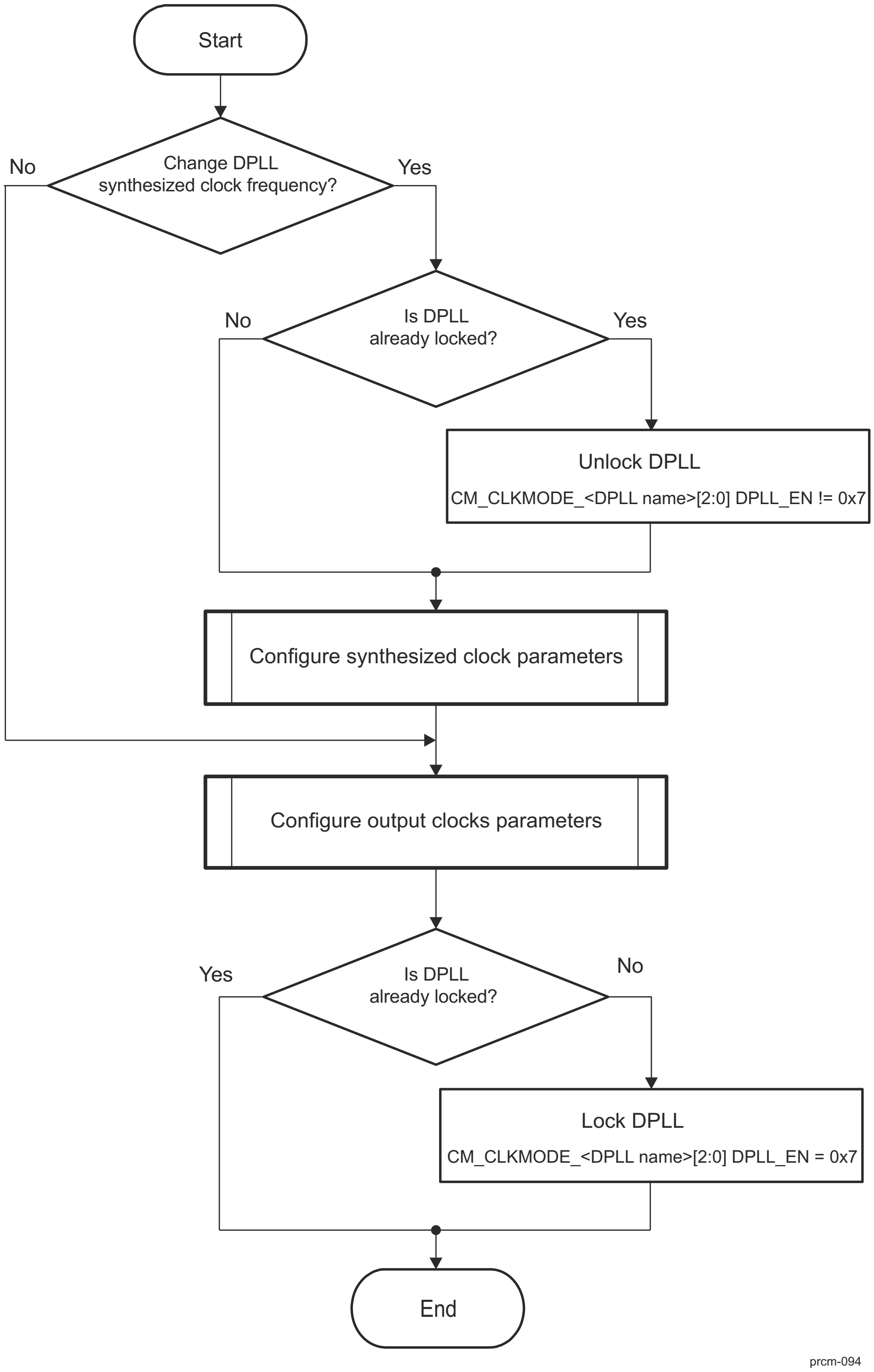


Figure 3-98. DPLL Output-Frequency Change

To unlock a DPLL, a mode different from the Lock Mode (0x7) should be programmed in the CM_CLKMODE_<DPLL NAME>[2:0] DPLL_EN bit field. The modes that can be programmed in the DPLL_EN bit field and can unlock the DPLL are:

- For type A DPLLs: Idle Low Power bypass mode (0x5) and Idle Fast Relock bypass mode (0x6)
- For type B DPLLs: Low Power Stop mode (0x1) and Idle Low Power bypass mode (0x5)

Table 3-427 and Table 3-428 summarize register and subprocess call sequences for DPLL output frequency changes.

Table 3-427. Register Call Summary for Sequence – DPLL Output Frequency Change

Register Name
CM_CLKMODE_DPLL name

Table 3-428. Subprocess Call Summary for Sequence – DPLL Output Frequency Change

Subprocess Name	Cross-Reference
Configure synthesized clock parameters.	See Section 3.10.1.1.2.3.
Configure output clocks parameters.	See Section 3.10.1.1.2.4.

3.10.2 Clock Management Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the clocks in the device.

3.10.2.1 Global Initialization

3.10.2.1.1 Surrounding Module Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DPLLs.

Table 3-429 describes the global initialization of the surrounding modules.

Table 3-429. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM DPLLs	Ensure that the DPLLS managed by the PRCM module are initialized.

3.10.2.1.2 Clock Management Global Initialization

3.10.2.1.2.1 Main Sequence – Clock Domain Global Initialization

This procedure initializes the clock domain of the device after a POR or software reset.

Table 3-430. Clock Domain Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure module clock-management feature of master modules in the clock domain.	<Module name>_SYSCONFIG[x] MIDLEMODE <Module name>_SYSCONFIG[x] STANDBYMODE	xx(1)
Configure module clock-management feature of slave modules in the clock domain.	See Section 3.10.3.3.	
Enable/disable static sleep dependency with other clock domains (that is, destination clock domains). Not all dependencies are configurable.	CM_<Clock Domain name>_STATICDEP[x] <Destination Clock Domain name>_STATDEP	0x0: Disable 0x1: Enable
Set dynamic dependency window size.	CM_<Clock Domain name>_DYNAMICDEP[27:24] WINDOWSIZE	xx(2)
Enable/disable module wake-up dependency for the modules of the clock domain. It is available when the module can generate an interrupt or a DMA request to a service provider module (for example, a processor or DMA).	PM_<Clock Domain name>_<Module name>_WKDEP[x] WKUPDEP_<Module name>_<DMA/IRQ reques>t_<DMA/Processor name>	0x0: Disable 0x1: Enable
Set clock domain state transition feature.	CM_<Clock Domain name>_CLKSTCTRL[1:0] CLKTRCTRL	xx(3)

(1) See the module register for valid modes.
(2) It depends on the desired size of the window. See Section 3.1.1.1.8.2, *Clock Domain Dependency*.
(3) It depends on the desired state of the clock domain. See Table 3-12.

3.10.2.1.2.2 Subsequence – Slave Module Clock-Management Parameters Configuration

This procedure configures the SSC parameters for the DPLL and enables the SSC feature.

Table 3-431. Slave Module Clock-Management Parameter Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module idle mode feature.	<Module name>_SYSCONFIG[x] SIDLEMODE <Module name>_SYSCONFIG[x] IDLEMODE	xx[(1)]
IF : Smart-idle mode is selected	<Module name>_SYSCONFIG[x] SIDLEMODE <Module name>_SYSCONFIG[x] IDLEMODE	0x10
Configure module clock requirement feature.	<Module name>_SYSCONFIG[x] CLOCKACTIVITY	x[(1)]
ENDIF		
Configure module management behavior on the PRCM module side.	CM_<Clock Domain name>_<Module name>_CLKCTRL[1:0] MODULEMODE	xx[(2)]

(1) See the module register for valid settings.
(2) The selected value depends on the desired clock-management behavior.

3.10.2.2 Clock Domain Sleep Transition and Troubleshooting

This procedure initiates a sleep transition on a clock domain and allows debugging if the transition does not occur.

Table 3-432. Clock Domain Sleep Transition and Troubleshooting

Step	Register/Bit Field/Programming Model	Value
Set clock domain sleep transition state.	CM_<Clock Domain name>_CLKSTCTRL[1:0] CLKTRCTRL	0x1: SW_SLEEP 0x3: HW_AUTO
IF : Clock domain sleep transition not initiated?		
Check that all clock domain master modules are in standby mode.	CM_<Clock Domain name>_<Module name>_CLKCTRL[18] STBYST	0x1: Module in standby
Check that all clock domain slave modules are in idle mode.	CM_<Clock Domain name>_<Module name>_CLKCTRL[17:16] IDLEST	0x1: In transition 0x2: Interface clock idled 0x3: Module idled
ENDIF		

3.10.2.3 Enable/Disable Software-Programmable Static Dependency

To change the setting of a software-programmable static dependency, use the procedure described in Table 3-433.

Table 3-433. Enable/Disable Software-Programmable Static Dependency

Step	Register/Bit Field/Programming Model	Value
Force destination domain to be awake (SW_WKUP).	CM_<Dest_CDname>_CLKSTCTRL[1:0] CLKTRCTRL	0x2
Wait until power domain that encloses the destination domain is ON.	PM_<Dest_PDname>_PWRSTST	=0x3
Change the static dependency.	CM_<Src_CDname>_STATICDEP[x] Dest_CDname_STATDEP	0x1
Put destination domain back to automatic transition (HW_AUTO).	CM_<Dest_CDname> _CLKSTCTRL[1:0] CLKTRCTRL	0x3

3.10.3 Power Management Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and control of the power domain in the device.

3.10.3.1 Global Initialization

3.10.3.1.1 Surrounding Module Global Initialization

Initialization of any surrounding modules within the device is not required. The external power IC and the device clocks should be active.

3.10.3.1.2 Power Management Global Initialization

3.10.3.1.2.1 Main Sequence – Power Domain Global Initialization and Setting

This procedure initializes the power domain of the device after a POR or software reset.

Table 3-434. Power Domain Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure memory area power state when the power domain is on. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_ONSTATE	0x1: RETAINED 0x3: ON
Configure memory area power state when the power domain transitions to RETENTION state. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_RETSTATE	0x1: RETAINED
Configure logic area RETENTION power state when the power domain transitions to RETENTION state.	PM_<Power Domain name>_PWRSTCTRL[2] LOGICRETSTATE	0x1: CSWR
Select target power state of the power domain. Not all states are programmable.	PM_<Power Domain name>_PWRSTCTRL[1:0] POWERSTATE	0x1: RETENTION 0x2: ON-INACTIVE 0x3: ON-ACTIVE
Wait until power state change is complete.	PM_<Power Domain name>_PWRSTST[1:0] POWERSTATEST	0x1: RETENTION 0x2: ON-INACTIVE 0x3: ON-ACTIVE

3.10.3.2 Forced Memory Area State Change With Power Domain ON

This procedure initiates a forced memory area state change while the power domain is ON.

Table 3-435. Forced Memory Area State Change With Power Domain ON

Step	Register/Bit Field/Programming Model	Value
Configure memory area target power state. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_ONSTATE	0x1: RETAINED 0x3: ON
Get memory area current state.	PM_<Power Domain name>_PWRSTST[x] <Memory Bank name>_STATEST	0x1: RETAINED 0x3: ON

3.10.3.3 Forced Power Domain Low-Power State Transition

Table 3-436. Forced Power Domain Low-Power State Transition

Step	Register/Bit Field/Programming Model	Value
Select target low-power state of the power domain. Not all states are programmable.	PM_<Power Domain name>_PWRSTCTRL[1:0] POWERSTATE	0x1: RETENTION
Force power domain low-power state transition.	PM_<Power Domain name>_PWRSTCTRL[4] LOWPOWERSTATECHANGE	0x1: Force change
Wait until state change is complete.	PM_<Power Domain name>_PWRSTCTRL[4] LOWPOWERSTATECHANGE	0x0: Change complete
Get current power state.	PM_<Power Domain name>_PWRSTST[1:0] POWERSTATEST	0x1: RETENTION

3.11

3.12 PRCM Software Configuration for OPP_PLUS

Note

For more information about the support of OPP_PLUS, see the "Operating Performance Points" section of the device Data Manual.

Note

Contact your TI software support representative for details on OPP_PLUS configuration.

3.13 PRCM Register Manual

3.13.1 PRCM Instance Summary

Table 3-437. PRCM L4_CFG Instance Summary

Module Name	Base Address L4_CFG Interconnect	Size
CM_CORE_AON__OCP_SOCKET	0x4A00 5000	256 Bytes
CM_CORE_AON__CKGEN	0x4A00 5100	504 Bytes
CM_CORE_AON__MPU	0x4A00 5300	44 Bytes
CM_CORE_AON__DSP1	0x4A00 5400	36 Bytes
CM_CORE_AON__IPU	0x4A00 5500	132 Bytes
CM_CORE_AON__DSP2	0x4A00 5600	36 Bytes
CM_CORE_AON__EVE1	0x4A00 5640	36 Bytes
CM_CORE_AON__EVE2	0x4A00 5680	36 Bytes
RESERVED	0x4A00 56C0	36 Bytes
RESERVED	0x4A00 5700	36 Bytes
CM_CORE_AON__RTC	0x4A00 5740	8 Bytes
CM_CORE_AON__VPE	0x4A00 5760	12 Bytes
CM_CORE_AON__RESTORE	0x4A00 5E00	84 Bytes
CM_CORE_AON__INSTR	0x4A00 5F00	52 Bytes
CM_CORE__OCP_SOCKET	0x4A00 8000	244 Bytes
CM_CORE__CKGEN	0x4A00 8100	296 Bytes
CM_CORE__COREAON	0x4A00 8600	212 Bytes
CM_CORE__CORE	0x4A00 8700	1876 Bytes
CM_CORE__IVA	0x4A00 8F00	44 Bytes
CM_CORE__CAM	0x4A00 9000	76 Bytes
CM_CORE__DSS	0x4A00 9100	64 Bytes
CM_CORE__GPU	0x4A00 9200	36 Bytes
CM_CORE__L3INIT	0x4A00 9300	244 Bytes
CM_CORE__CUSTEFUSE	0x4A00 9600	36 Bytes
CM_CORE__L4PER	0x4A00 9700	536 Bytes
CM_CORE__RESTORE	0x4A00 9E00	88 Bytes
RESERVED	0x4A0D 9000	80 Bytes
RESERVED	0x4A0D D000	80 Bytes
RESERVED	0x4A18 3000	80 Bytes
RESERVED	0x4A18 5000	80 Bytes
RESERVED	0x4A18 7000	80 Bytes

Table 3-438. PRCM L4_WKUP Instance Summary

Module Name	Base Address L4_WKUP Interconnect	Size
OCP_SOCKET_PRM	0x4AE0 6000	248 Bytes
CKGEN_PRM	0x4AE0 6100	228 Bytes
MPU_PRM	0x4AE0 6300	40 Bytes
DSP1_PRM	0x4AE0 6400	40 Bytes
IPU_PRM	0x4AE0 6500	136 Bytes
RESERVED	0x4AE0 6600	184 Bytes
CORE_PRM	0x4AE0 6700	1864 Bytes
IVA_PRM	0x4AE0 6F00	48 Bytes
CAM_PRM	0x4AE0 7000	80 Bytes
DSS_PRM	0x4AE0 7100	64 Bytes
GPU_PRM	0x4AE0 7200	40 Bytes
L3INIT_PRM	0x4AE0 7300	248 Bytes
L4PER_PRM	0x4AE0 7400	512 Bytes
CUSTEFUSE_PRM	0x4AE0 7600	40 Bytes
WKUPAON_PRM	0x4AE0 7700	188 Bytes
WKUPAON_CM	0x4AE0 7800	220 Bytes
EMU_PRM	0x4AE0 7900	40 Bytes
EMU_CM	0x4AE0 7A00	16 Bytes
DSP2_PRM	0x4AE0 7B00	40 Bytes
EVE1_PRM	0x4AE0 7B40	40 Bytes
EVE2_PRM	0x4AE0 7B80	40 Bytes
RESERVED	0x4AE0 7BC0	40 Bytes
RESERVED	0x4AE0 7C00	40 Bytes
RTC_PRM	0x4AE0 7C40	8 Bytes
VPE_PRM	0x4AE0 7C80	40 Bytes
DEVICE_PRM	0x4AE0 7D00	312 Bytes
INSTR_PRM	0x4AE0 7F00	52 Bytes

3.13.2 CM_CORE_AON__CKGEN Registers

3.13.2.1 CM_CORE_AON__CKGEN Register Summary

Table 3-439. CM_CORE_AON__CKGEN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__CKGEN Physical Address L4_CFG Interconnect
CM_CLKSEL_CORE	RW	32	0x0000 0000	0x4A00 5100
CM_CLKSEL_ABE	RW	32	0x0000 0008	0x4A00 5108
CM_DLL_CTRL	RW	32	0x0000 0010	0x4A00 5110
CM_CLKMODE_DPLL_CORE	RW	32	0x0000 0020	0x4A00 5120
CM_IDLEST_DPLL_CORE	R	32	0x0000 0024	0x4A00 5124
CM_AUTOIDLE_DPLL_CORE	RW	32	0x0000 0028	0x4A00 5128
CM_CLKSEL_DPLL_CORE	RW	32	0x0000 002C	0x4A00 512C
CM_DIV_M2_DPLL_CORE	RW	32	0x0000 0030	0x4A00 5130
RESERVED	RW	32	0x0000 0034	0x4A00 5134
RESERVED	RW	32	0x0000 0038	0x4A00 5138
CM_DIV_H12_DPLL_CORE	RW	32	0x0000 003C	0x4A00 513C
CM_DIV_H13_DPLL_CORE	RW	32	0x0000 0040	0x4A00 5140
CM_DIV_H14_DPLL_CORE	RW	32	0x0000 0044	0x4A00 5144

Table 3-439. CM_CORE_AON__CKGEN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__CKGEN Physical Address L4_CFG Interconnect
RESERVED	R	32	0x0000 0048	0x4A00 5148
RESERVED	R	32	0x0000 004C	0x4A00 514C
RESERVED	RW	32	0x0000 0050	0x4A00 5150
CM_DIV_H22_DPLL_CORE	RW	32	0x0000 0054	0x4A00 5154
CM_DIV_H23_DPLL_CORE	RW	32	0x0000 0058	0x4A00 5158
CM_DIV_H24_DPLL_CORE	RW	32	0x0000 005C	0x4A00 515C
CM_CLKMODE_DPLL_MPU	RW	32	0x0000 0060	0x4A00 5160
CM_IDLEST_DPLL_MPU	R	32	0x0000 0064	0x4A00 5164
CM_AUTOIDLE_DPLL_MPU	RW	32	0x0000 0068	0x4A00 5168
CM_CLKSEL_DPLL_MPU	RW	32	0x0000 006C	0x4A00 516C
CM_DIV_M2_DPLL_MPU	RW	32	0x0000 0070	0x4A00 5170
RESERVED	R	32	0x0000 0088	0x4A00 5188
RESERVED	R	32	0x0000 008C	0x4A00 518C
CM_BYPCLK_DPLL_MPU	RW	32	0x0000 009C	0x4A00 519C
CM_CLKMODE_DPLL_IVA	RW	32	0x0000 00A0	0x4A00 51A0
CM_IDLEST_DPLL_IVA	R	32	0x0000 00A4	0x4A00 51A4
CM_AUTOIDLE_DPLL_IVA	RW	32	0x0000 00A8	0x4A00 51A8
CM_CLKSEL_DPLL_IVA	RW	32	0x0000 00AC	0x4A00 51AC
CM_DIV_M2_DPLL_IVA	RW	32	0x0000 00B0	0x4A00 51B0
RESERVED	RW	32	0x0000 00B4	0x4A00 51B4
RESERVED	R	32	0x0000 00C8	0x4A00 51C8
RESERVED	R	32	0x0000 00CC	0x4A00 51CC
CM_BYPCLK_DPLL_IVA	RW	32	0x0000 00DC	0x4A00 51DC
CM_CLKMODE_DPLL_ABE	RW	32	0x0000 00E0	0x4A00 51E0
CM_IDLEST_DPLL_ABE	R	32	0x0000 00E4	0x4A00 51E4
CM_AUTOIDLE_DPLL_ABE	RW	32	0x0000 00E8	0x4A00 51E8
CM_CLKSEL_DPLL_ABE	RW	32	0x0000 00EC	0x4A00 51EC
CM_DIV_M2_DPLL_ABE	RW	32	0x0000 00F0	0x4A00 51F0
CM_DIV_M3_DPLL_ABE	RW	32	0x0000 00F4	0x4A00 51F4
RESERVED	R	32	0x0000 0108	0x4A00 5208
RESERVED	R	32	0x0000 010C	0x4A00 520C
CM_CLKMODE_DPLL_DDR	RW	32	0x0000 0110	0x4A00 5210
CM_IDLEST_DPLL_DDR	R	32	0x0000 0114	0x4A00 5214
CM_AUTOIDLE_DPLL_DDR	RW	32	0x0000 0118	0x4A00 5218
CM_CLKSEL_DPLL_DDR	RW	32	0x0000 011C	0x4A00 521C
CM_DIV_M2_DPLL_DDR	RW	32	0x0000 0120	0x4A00 5220
RESERVED	RW	32	0x0000 0124	0x4A00 5224
CM_DIV_H11_DPLL_DDR	RW	32	0x0000 0128	0x4A00 5228
RESERVED	R	32	0x0000 012C	0x4A00 522C
RESERVED	R	32	0x0000 0130	0x4A00 5230
CM_CLKMODE_DPLL_DSP	RW	32	0x0000 0134	0x4A00 5234
CM_IDLEST_DPLL_DSP	R	32	0x0000 0138	0x4A00 5238
CM_AUTOIDLE_DPLL_DSP	RW	32	0x0000 013C	0x4A00 523C
CM_CLKSEL_DPLL_DSP	RW	32	0x0000 0140	0x4A00 5240
CM_DIV_M2_DPLL_DSP	RW	32	0x0000 0144	0x4A00 5244
CM_DIV_M3_DPLL_DSP	RW	32	0x0000 0148	0x4A00 5248
RESERVED	R	32	0x0000 014C	0x4A00 524C
RESERVED	R	32	0x0000 0150	0x4A00 5250
CM_BYPCLK_DPLL_DSP	RW	32	0x0000 0154	0x4A00 5254

Table 3-439. CM_CORE_AON__CKGEN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__CKGEN Physical Address L4_CFG Interconnect
CM_SHADOW_FREQ_CONFIG1	RW	32	0x0000 0160	0x4A00 5260
CM_SHADOW_FREQ_CONFIG2	RW	32	0x0000 0164	0x4A00 5264
CM_DYN_DEP_PRESCAL	RW	32	0x0000 0170	0x4A00 5270
RESERVED	R	32	0x0000 0180	0x4A00 5280
CM_CLKMODE_DPLL_EVE	RW	32	0x0000 0184	0x4A00 5284
CM_IDLEST_DPLL_EVE	R	32	0x0000 0188	0x4A00 5288
CM_AUTOIDLE_DPLL_EVE	RW	32	0x0000 018C	0x4A00 528C
CM_CLKSEL_DPLL_EVE	RW	32	0x0000 0190	0x4A00 5290
CM_DIV_M2_DPLL_EVE	RW	32	0x0000 0194	0x4A00 5294
RESERVED	R	32	0x0000 0198	0x4A00 5298
RESERVED	R	32	0x0000 019C	0x4A00 529C
RESERVED	R	32	0x0000 01A0	0x4A00 52A0
CM_BYPCLK_DPLL_EVE	RW	32	0x0000 01A4	0x4A00 52A4
CM_CLKMODE_DPLL_GMAC	RW	32	0x0000 01A8	0x4A00 52A8
CM_IDLEST_DPLL_GMAC	R	32	0x0000 01AC	0x4A00 52AC
CM_AUTOIDLE_DPLL_GMAC	RW	32	0x0000 01B0	0x4A00 52B0
CM_CLKSEL_DPLL_GMAC	RW	32	0x0000 01B4	0x4A00 52B4
CM_DIV_M2_DPLL_GMAC	RW	32	0x0000 01B8	0x4A00 52B8
CM_DIV_M3_DPLL_GMAC	RW	32	0x0000 01BC	0x4A00 52BC
CM_DIV_H11_DPLL_GMAC	RW	32	0x0000 01C0	0x4A00 52C0
CM_DIV_H12_DPLL_GMAC	RW	32	0x0000 01C4	0x4A00 52C4
CM_DIV_H13_DPLL_GMAC	RW	32	0x0000 01C8	0x4A00 52C8
RESERVED	R	32	0x0000 01CC	0x4A00 52CC
RESERVED	R	32	0x0000 01D0	0x4A00 52D0
RESERVED	R	32	0x0000 01D4	0x4A00 52D4
CM_CLKMODE_DPLL_GPU	RW	32	0x0000 01D8	0x4A00 52D8
CM_IDLEST_DPLL_GPU	R	32	0x0000 01DC	0x4A00 52DC
CM_AUTOIDLE_DPLL_GPU	RW	32	0x0000 01E0	0x4A00 52E0
CM_CLKSEL_DPLL_GPU	RW	32	0x0000 01E4	0x4A00 52E4
CM_DIV_M2_DPLL_GPU	RW	32	0x0000 01E8	0x4A00 52E8
RESERVED	R	32	0x0000 01EC	0x4A00 52EC
RESERVED	R	32	0x0000 01F0	0x4A00 52F0
RESERVED	R	32	0x0000 01F4	0x4A00 52F4

3.13.2.2 CM_CORE_AON__CKGEN Register Description

Table 3-440. CM_CLKSEL_CORE

Address Offset	0x0000 0000		
Physical Address	0x4A00 5100	**Instance**	CM_CORE_AON__CKGEN
Description	CORE module clock selection.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>CLKSEL_L4</td><td colspan="3">RESERVED</td><td>CLKSEL_L3</td><td colspan="4">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKSEL_L4	Selects L4 interconnect clock (L4_clk) 0x0: RESERVED 0x1: L4_CLK is L3_CLK divided by 2	R	0x1
7:5	RESERVED		R	0x0
4	CLKSEL_L3	Selects L3 interconnect clock (L3_clk) 0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0x0
3:0	RESERVED		R	0x0

Table 3-441. CM_CLKSEL_ABE

Address Offset	0x0000 0008		
Physical Address	0x4A00 5108	**Instance**	CM_CORE_AON__CKGEN
Description	ABE module clock selection.		
Type	RW		

31–2	1–0
RESERVED	CLKSEL_OPP

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1:0	CLKSEL_OPP	Selects the OPP divider ABE domain 0x0: ABE_CLK is divide by 1 of DPLL_ABE_X2_CLK 0x1: ABE_CLK is divide by 2 of DPLL_ABE_X2_CLK 0x2: ABE_CLK is divide by 4 of DPLL_ABE_X2_CLK 0x3: Reserved	RW	0x0

Table 3-442. CM_DLL_CTRL

Address Offset	0x0000 0010		
Physical Address	0x4A00 5110	**Instance**	CM_CORE_AON__CKGEN
Description	Special register for DLL control		
Type	RW		

31–1	0
RESERVED	DLL_OVERRIDE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	DLL_OVERRIDE	Control if DLL lock and code outputs are overriden or not 0x0: NO_OVR 0x1: OVR	RW	0x1

Table 3-443. CM_CLKMODE_DPLL_CORE

Address Offset	0x0000 0020

Table 3-443. CM_CLKMODE_DPLL_CORE (continued)

Physical Address	0x4A00 5120	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31:12	11	10	9	8	7:3	2:0
RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-444. CM_IDLEST_DPLL_CORE

Address Offset	0x0000 0024		
Physical Address	0x4A00 5124	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL_NOTINIT 0x1: DPLL_INIT	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose).	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL_UNLOCKED 0x1: DPLL_LOCKED	R	0x0

Table 3-445. CM_AUTOIDLE_DPLL_CORE

Address Offset	0x0000 0028		
Physical Address	0x4A00 5128	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31:3	2:0
RESERVED	AUTO_DPLL_MODE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control. 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-446. CM_CLKSEL_DPLL_CORE

Address Offset	0x0000 002C		
Physical Address	0x4A00 512C	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21	RESERVED		R	0x0
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL 0x0: CLKOUTHIF is generated from the DPLL oscillator (DCO) 0x1: CLKOUTHIF is generated from CLKINPHIF	RW	0x0
19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-447. CM_DIV_M2_DPLL_CORE

Address Offset	0x0000 0030		
Physical Address	0x4A00 5130	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31–10	9	8–5	4–0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_CORE. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-448. CM_DIV_H12_DPLL_CORE

Address Offset	0x0000 003C		
Physical Address	0x4A00 513C	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED			DIVHS					

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED	Reserved	R	0x0
5:0	DIVHS	This field programs the H12 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H12 = /1 0x2: H12 = /2 ... 0x3F: H12 = /63	RW	0x1

Table 3-449. CM_DIV_H13_DPLL_CORE

Address Offset	0x0000 0040		
Physical Address	0x4A00 5140	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED			DIVHS					

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
5:0	DIVHS	This field programs the H13 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H13 = /1 0x2: H13 = /2 ... 0x3F: H13 = /63	RW	0x1

Table 3-450. CM_DIV_H14_DPLL_CORE

Address Offset	0x0000 0044		
Physical Address	0x4A00 5144	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT4 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT4 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H14 post-divider factor (1 to 63) of DPLL_CORE. When a value of 63 is programmed in this register, HS divider will perform division by 2.5 that is divided by 2 at top level. 0x0: Reserved 0x1: H14 = /1 0x2: H14 = /2 ... 0x3F: H14 = /63	RW	0x1

Table 3-451. CM_DIV_H22_DPLL_CORE

Address Offset	0x0000 0054		
Physical Address	0x4A00 5154	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT2 o/p of the 2nd HSDIVIDER.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER2 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0

Bits	Field Name	Description	Type	Reset
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H22 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H22 = /1 0x2: H22 = /2 ... 0x3F: H22 = /63	RW	0x1

Table 3-452. CM_DIV_H23_DPLL_CORE

Address Offset	0x0000 0058		
Physical Address	0x4A00 5158	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT3 o/p of the 2nd HSDIVIDER.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER2 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H23 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H23 = /1 0x2: H23 = /2 ... 0x3F: H23 = /63	RW	0x1

Table 3-453. CM_DIV_H24_DPLL_CORE

Address Offset	0x0000 005C		
Physical Address	0x4A00 515C	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT4 o/p of the 2nd HSDIVIDER.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER2 CLKOUT4 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
5:0	DIVHS	This field programs the H24 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H24 = /1 0x2: H24 = /2 ... 0x3F: H24 = /63	RW	0x1

Table 3-454. CM_CLKMODE_DPLL_MPU

Address Offset	0x0000 0060		
Physical Address	0x4A00 5160	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31–12	11	10	9	8	7–3	2–0
RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		RW	0x0

Bits	Field Name	Description	Type	Reset
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-455. CM_IDLEST_DPLL_MPU

Address Offset	0x0000 0064		
Physical Address	0x4A00 5164	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31–5	4	3–1	0
RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-456. CM_AUTOIDLE_DPLL_MPU

Address Offset	0x0000 0068		
Physical Address	0x4A00 5168	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		

Table 3-456. CM_AUTOIDLE_DPLL_MPU (continued)

Type	RW

31–3	2–0
RESERVED	AUTO_DPLL_MODE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-457. CM_CLKSEL_DPLL_MPU

Address Offset	0x0000 006C		
Physical Address	0x4A00 516C	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31–24	23	22	21–19	18–8	7	6–0
RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Only CLKINPULOW bypass clock supported for this PLL	R	0x1
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled 0x1: Duty-cycle corrector is enabled	RW	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-458. CM_DIV_M2_DPLL_MPU

Address Offset	0x0000 0070		
Physical Address	0x4A00 5170	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31:10	9	8:5	4:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_MPU. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-459. CM_BYPCLK_DPLL_MPU

Address Offset	0x0000 009C		
Physical Address	0x4A00 519C	**Instance**	CM_CORE_AON__CKGEN
Description	Control MPU PLL BYPASS clock. [warm reset insensitive]		
Type	RW		

31:2	1:0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL MPU bypass clock	RW	0x0

Table 3-460. CM_CLKMODE_DPLL_IVA

Address Offset	0x0000 00A0		
Physical Address	0x4A00 51A0	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DPLL_RESGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_RESGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-461. CM_IDLEST_DPLL_IVA

Address Offset	0x0000 00A4		
Physical Address	0x4A00 51A4	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose) 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-462. CM_AUTOIDLE_DPLL_IVA

Address Offset	0x0000 00A8		
Physical Address	0x4A00 51A8	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													AUTO_DPLL_MODE		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-463. CM_CLKSEL_DPLL_IVA

Address Offset	0x0000 00AC		
Physical Address	0x4A00 51AC	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED			DPLL_MULT											RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-464. CM_DIV_M2_DPLL_IVA

Address Offset	0x0000 00B0		
Physical Address	0x4A00 51B0	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED				DIVHS				

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_IVA. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-465. CM_BYPCLK_DPLL_IVA

Address Offset	0x0000 00DC		
Physical Address	0x4A00 51DC	**Instance**	CM_CORE_AON__CKGEN
Description	Control IVA PLL BYPASS clock. [warm reset insensitive]		
Type	RW		

31–2	1–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL IVA bypass clock 0x0: DPLL_IVA bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_IVA bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_IVA bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_IVA bypass clock is CORE_X2_CLK divided by 8	RW	0x0

Table 3-466. CM_CLKMODE_DPLL_ABE

Address Offset	0x0000 00E0		
Physical Address	0x4A00 51E0	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31–12	11	10	9	8	7–3	2–0
RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled 0x1: REGM4XEN mode of the DPLL is enabled	RW	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-467. CM_IDLEST_DPLL_ABE

Address Offset	0x0000 00E4		
Physical Address	0x4A00 51E4	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31:5	4	3:1	0
RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0

Bits	Field Name	Description	Type	Reset
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-468. CM_AUTOIDLE_DPLL_ABE

Address Offset	0x0000 00E8		
Physical Address	0x4A00 51E8	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31–3	2–0
RESERVED	AUTO_DPLL_MODE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control;	RW	0x0

Table 3-469. CM_CLKSEL_DPLL_ABE

Address Offset	0x0000 00EC		
Physical Address	0x4A00 51EC	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31–24	23	22	21–19	18–8	7	6–0
RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Only CLKINPULOW bypass clock supported for this PLL	R	0x1
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0

Bits	Field Name	Description	Type	Reset
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-470. CM_DIV_M2_DPLL_ABE

Address Offset	0x0000 00F0		
Physical Address	0x4A00 51F0	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="20">RESERVED</td><td>CLKX2ST</td><td>RESERVED</td><td>CLKST</td><td colspan="4">RESERVED</td><td colspan="5">DIVHS</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	CLKX2ST	DPLL CLKOUTX2 status 0x0: CLK_GATED 0x1: CLK_ENABLED	R	0x0
10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: CLK_GATED 0x1: CLK_ENABLED	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_ABE. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-471. CM_DIV_M3_DPLL_ABE

Address Offset	0x0000 00F4		
Physical Address	0x4A00 51F4	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td>CLKST</td><td colspan="4">RESERVED</td><td colspan="5">DIVHS</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_ABE. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-472. CM_CLKMODE_DPLL_DDR

Address Offset	0x0000 0110		
Physical Address	0x4A00 5210	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31:12	11	10	9	8	7:3	2:0
RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-473. CM_IDLEST_DPLL_DDR

Address Offset	0x0000 0114		
Physical Address	0x4A00 5214	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31–5	4	3–1	0
RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-474. CM_AUTOIDLE_DPLL_DDR

Address Offset	0x0000 0118		
Physical Address	0x4A00 5218	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		

Table 3-474. CM_AUTOIDLE_DPLL_DDR (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													AUTO_DPLL_MODE		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-475. CM_CLKSEL_DPLL_DDR

Address Offset	0x0000 011C		
Physical Address	0x4A00 521C	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED			DPLL_MULT											RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled 0x1: Duty-cycle corrector is enabled	RW	0x0
21:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-476. CM_DIV_M2_DPLL_DDR

Address Offset	0x0000 0120		
Physical Address	0x4A00 5220	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31:10	9	8:5	4:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_DDR. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-477. CM_DIV_H11_DPLL_DDR

Address Offset	0x0000 0128		
Physical Address	0x4A00 5228	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT1 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
5:0	DIVHS	This field programs the H11 post-divider factor (1 to 63) of DPLL_DDR. 0x0: Reserved 0x1: H11 = /1 0x2: H11 = /2 ... 0x3F: H11 = /63	RW	0x1

Table 3-478. CM_CLKMODE_DPLL_DSP

Address Offset	0x0000 0134		
Physical Address	0x4A00 5234	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31:12	11	10	9	8	7:3	2:0
RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-479. CM_IDLEST_DPLL_DSP

Address Offset	0x0000 0138		
Physical Address	0x4A00 5238	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose) 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-480. CM_AUTOIDLE_DPLL_DSP

Address Offset	0x0000 013C		
Physical Address	0x4A00 523C	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		

Table 3-480. CM_AUTOIDLE_DPLL_DSP (continued)

Type	RW

31–3	2–0
RESERVED	AUTO_DPLL_MODE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-481. CM_CLKSEL_DPLL_DSP

Address Offset	0x0000 0140		
Physical Address	0x4A00 5240	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31–24	23	22	21–19	18–8	7	6–0
RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-482. CM_DIV_M2_DPLL_DSP

Address Offset	0x0000 0144		
Physical Address	0x4A00 5244	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31:10	9	8:5	4:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: CLK_GATED 0x1: CLK_ENABLED	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_DSP. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-483. CM_DIV_M3_DPLL_DSP

Address Offset	0x0000 0148		
Physical Address	0x4A00 5248	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

31:10	9	8:5	4:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_DSP. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-484. CM_BYPCLK_DPLL_DSP

Address Offset	0x0000 0154		
Physical Address	0x4A00 5254	**Instance**	CM_CORE_AON__CKGEN
Description	Control IVA PLL BYPASS clock. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL IVA bypass clock 0x0: DPLL_IVA bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_IVA bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_IVA bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_IVA bypass clock is CORE_X2_CLK divided by 8	RW	0x0

Table 3-485. CM_SHADOW_FREQ_CONFIG1

Address Offset	0x0000 0160		
Physical Address	0x4A00 5260	**Instance**	CM_CORE_AON__CKGEN
Description	Shadow register to program new DPLL configuration affecting EMIF and GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DPLL_DDR_DPLL_EN			DPLL_DDR_M2_DIV					RESERVED							DLL_RESET	DLL_OVERRIDE	RESERVED	FREQ_UPDATE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	DPLL_DDR_DPLL_EN	Shadow register for CM_CLKMODE_DPLL_DDR.DPLL_EN. The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5
15:11	DPLL_DDR_M2_DIV	Shadow register for CM_DIV_M2_DPLL_DDR.DIVHS. The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. Divide value from 1 to 31. 0x0: Reserved	RW	0x1
10:4	RESERVED		R	0x0
3	DLL_RESET	Specify if DLL should be reset or not during the frequency change hardware sequence. 0x0: DLL is not reset during the frequency change hardware sequence 0x1: DLL is reset automatically during the frequency change hardware sequence	RW	0x1
2	DLL_OVERRIDE	Shadow register for CM_DLL_CTRL.DLL_OVERRIDE.The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. 0x0: Lock and code outputs are not overriden 0x1: Lock output is overriden to '1' and code output is overriden with a value coming from control module.	RW	0x1
1	RESERVED		R	0x0
0	FREQ_UPDATE	Writing '1' indicates that a new configuration is available. It is automatically cleared by h/w after the configuration has been applied.	RW	0x0

Table 3-486. CM_SHADOW_FREQ_CONFIG2

Address Offset	0x0000 0164		
Physical Address	0x4A00 5264	**Instance**	CM_CORE_AON__CKGEN
Description	Shadow register to program new DPLL configuration affecting GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DPLL_CORE_H12_DIV	CLKSEL_L3	GPMC_FREQ_UPDATE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:2	DPLL_CORE_H12_DIV	Shadow register for CM_DIV_H12_DPLL_CORE.DIVHS. The main register is automatically loaded with the shadow register value after GPMC IDLE if the CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE field is set to '1' and GPMC_FREQ_UPDATE is set to '1'. Divide value from 1 to 31. 0x0: Reserved	RW	0x1
1	CLKSEL_L3	Shadow register for CM_CLKSEL_CORE.CLKSEL_L3. The main register is automatically loaded with the shadow register value after GPMC IDLE if the CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE field is set to '1' and GPMC_FREQ_UPDATE is set to '1'. 0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0x0
0	GPMC_FREQ_UPDATE	Controls whether or not GPMC has to be put automatically into idle during the frequency change operation. 0x0: GPMC is not put automatically into idle during frequency change operation. 0x1: GPMC is put automatically into idle during frequency change operation.	RW	0x0

Table 3-487. CM_DYN_DEP_PRESCAL

Address Offset	0x0000 0170		
Physical Address	0x4A00 5270	**Instance**	CM_CORE_AON__CKGEN
Description	Control the time unit of the sliding window for dynamic dependencies (auto-sleep feature).		
Type	RW		

31–6	5–0
RESERVED	PRESCAL

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	PRESCAL	Time unit is equal to (PRESCAL + 1) L4 clock cycles.	RW	0x20

Table 3-488. CM_CLKMODE_DPLL_EVE

Address Offset	0x0000 0184		
Physical Address	0x4A00 5284	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-489. CM_IDLEST_DPLL_EVE

Address Offset	0x0000 0188		
Physical Address	0x4A00 5288	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose) 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-490. CM_AUTOIDLE_DPLL_EVE

Address Offset	0x0000 018C		
Physical Address	0x4A00 528C	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31–3	2–0
RESERVED	AUTO_DPLL_MODE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-491. CM_CLKSEL_DPLL_EVE

Address Offset	0x0000 0190		
Physical Address	0x4A00 5290	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31-24	23	22	21-19	18-8	7	6-0
RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-492. CM_DIV_M2_DPLL_EVE

Address Offset	0x0000 0194		
Physical Address	0x4A00 5294	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31-10	9	8-5	4-0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_EVE. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-493. CM_BYPCLK_DPLL_EVE

Address Offset	0x0000 01A4		
Physical Address	0x4A00 52A4	**Instance**	CM_CORE_AON__CKGEN
Description	Control IVA PLL BYPASS clock. [warm reset insensitive]		
Type	RW		

31:2	1:0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL IVA bypass clock	RW	0x0

Table 3-494. CM_CLKMODE_DPLL_GMAC

Address Offset	0x0000 01A8		
Physical Address	0x4A00 52A8	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31:12	11	10	9	8	7:3	2:0
RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-495. CM_IDLEST_DPLL_GMAC

Address Offset	0x0000 01AC		
Physical Address	0x4A00 52AC	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31:5	4	3:1	0
RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-496. CM_AUTOIDLE_DPLL_GMAC

Address Offset	0x0000 01B0		
Physical Address	0x4A00 52B0	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31–3	2–0
RESERVED	AUTO_DPLL_MODE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-497. CM_CLKSEL_DPLL_GMAC

Address Offset	0x0000 01B4		
Physical Address	0x4A00 52B4	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31–24	23	22	21	20	19	18–8	7	6–0
RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled 0x1: Duty-cycle corrector is enabled	RW	0x0
21	RESERVED		R	0x0
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL 0x0: CLKOUTHIF is generated from the DPLL oscillator (DCO) 0x1: CLKOUTHIF is generated from CLKINPHIF	RW	0x0
19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-498. CM_DIV_M2_DPLL_GMAC

Address Offset	0x0000 01B8		
Physical Address	0x4A00 52B8	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31:10	9	8:5	4:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_GMAC. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-499. CM_DIV_M3_DPLL_GMAC

Address Offset	0x0000 01BC		
Physical Address	0x4A00 52BC	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

31:10	9	8:5	4:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_GMAC. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-500. CM_DIV_H11_DPLL_GMAC

Address Offset	0x0000 01C0		
Physical Address	0x4A00 52C0	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT1 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H11 post-divider factor (1 to 63) of DPLL_GMAC. 0x0: Reserved 0x1: H11 = /1 0x2: H11 = /2 ... 0x3F: H11 = /63	RW	0x1

Table 3-501. CM_DIV_H12_DPLL_GMAC

Address Offset	0x0000 01C4

Table 3-501. CM_DIV_H12_DPLL_GMAC (continued)

Physical Address	0x4A00 52C4	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED	Reserved	R	0x0
5:0	DIVHS	This field programs the H12 post-divider factor (1 to 63) of DPLL_GMAC. 0x0: Reserved 0x1: H12 = /1 0x2: H12 = /2 ... 0x3F: H12 = /63	RW	0x1

Table 3-502. CM_DIV_H13_DPLL_GMAC

Address Offset	0x0000 01C8		
Physical Address	0x4A00 52C8	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED	Reserved	R	0x0
5:0	DIVHS	This field programs the H13 post-divider factor (1 to 63) of DPLL_GMAC. 0x0: Reserved 0x1: H13 = /1 0x2: H13 = /2 ... 0x3F: H13 = /63	RW	0x1

Table 3-503. CM_CLKMODE_DPLL_GPU

Address Offset	0x0000 01D8		
Physical Address	0x4A00 52D8	**Instance**	CM_CORE_AON__CKGEN

Table 3-503. CM_CLKMODE_DPLL_GPU (continued)

Description	This register allows controlling the DPLL modes.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED					DPLL_EN		

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-504. CM_IDLEST_DPLL_GPU

Address Offset	0x0000 01DC		
Physical Address	0x4A00 52DC	**Instance**	CM_CORE_AON__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL_NOTINIT 0x1: DPLL_INIT	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose).	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL_UNLOCKED 0x1: DPLL_LOCKED	R	0x0

Table 3-505. CM_AUTOIDLE_DPLL_GPU

Address Offset	0x0000 01E0		
Physical Address	0x4A00 52E0	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													AUTO_DPLL_MODE		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control. 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-506. CM_CLKSEL_DPLL_GPU

Address Offset	0x0000 01E4		
Physical Address	0x4A00 52E4	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: DISABLED	R	0x0
21	RESERVED		R	0x0
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL 0x0: SEL_DCO 0x1: SEL_CLKINPHIF	RW	0x0
19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: RESERVED_0 0x1: RESERVED_1	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-507. CM_DIV_M2_DPLL_GPU

Address Offset	0x0000 01E8		
Physical Address	0x4A00 52E8	**Instance**	CM_CORE_AON__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31–10	9	8–5	4–0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_GPU. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 **...** 0x1F: M2 = /31	RW	0x1

3.13.3 CM_CORE_AON__DSP1 Registers

3.13.3.1 CM_CORE_AON__DSP1 Register Summary

Table 3-508. CM_CORE_AON__DSP1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__DSP1 Physical Address L4_CFG Interconnect
CM_DSP1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5400
CM_DSP1_STATICDEP	RW	32	0x0000 0004	0x4A00 5404
CM_DSP1_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5408
CM_DSP1_DSP1_CLKCTRL	RW	32	0x0000 0020	0x4A00 5420

3.13.3.2 CM_CORE_AON__DSP1 Register Description

Table 3-509. CM_DSP1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5400	**Instance**	CM_CORE_AON__DSP1
Description	This register enables the DSP domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>CLKACTIVITY_DSP1_GFCLK</td><td colspan="6">RESERVED</td><td colspan="2">CLKTRCTRL</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_DSP1_GFCLK	This field indicates the state of the DSP_ROOT_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-510. Register Call Summary for Register CM_DSP1_CLKSTCTRL

Reset Management Functional Description

- DSP1 Subsystem Software Warm Reset Sequence: [0]

Clock Management Functional Description

- Clock Domain Modes: [1] [2]

PRCM Register Manual

- CM_CORE_AON__DSP1 Register Summary: [3]

Table 3-511. CM_DSP1_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5404	**Instance**	CM_CORE_AON__DSP1
Description	This register controls the static domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	ATL_STATDEP	PCIE_STATDEP	VPE_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	RESERVED	RESERVED	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ATL_STATDEP	Static dependency towards L3INIT Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
29	PCIE_STATDEP	Static dependency towards PCIE Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
28	VPE_STATDEP	Static dependency towards VPE Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
27	L4PER3_STATDEP	Static dependency towards L4PER3 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE Clock Domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON Clock Domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	L4SEC_STATDEP	Static dependency towards L4SEC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG Clock Domain 0x0: Dependency is disabled	R	0x0
11	RESERVED		R	0x0
10	GPU_STATDEP	Static dependency towards GPU Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
9	CAM_STATDEP	Static dependency towards CAM Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	DSS_STATDEP	Static dependency towards DSS Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 Clock Domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	RESERVED		R	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-512. Register Call Summary for Register CM_DSP1_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24]

PRCM Register Manual

- CM_CORE_AON__DSP1 Register Summary: [27]

Table 3-513. CM_DSP1_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5408	**Instance**	CM_CORE_AON__DSP1
Description	This register controls the dynamic domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31:28	27:24	23:6	5	4:0
RESERVED	WINDOWSIZE	RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-514. Register Call Summary for Register CM_DSP1_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]
- Clock Domain Dependency: [1]

PRCM Register Manual

- CM_CORE_AON__DSP1 Register Summary: [2]

Table 3-515. CM_DSP1_DSP1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5420	**Instance**	CM_CORE_AON__DSP1
Description	This register manages the DSP clocks.		
Type	RW		

31–19	18	17–16	15–2	1–0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-516. Register Call Summary for Register CM_DSP1_DSP1_CLKCTRL

Reset Management Functional Description

- DSP1 Subsystem Software Warm Reset Sequence: [0] [1]

Clock Management Functional Description

- Clock Domain Module Attributes: [2] [3] [4]

PRCM Register Manual

- CM_CORE_AON__DSP1 Register Summary: [5]

3.13.4 CM_CORE_AON__DSP2 Registers

3.13.4.1 CM_CORE_AON__DSP2 Register Summary

Table 3-517. CM_CORE_AON__DSP2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__DSP2 Physical Address L4_CFG Interconnect
CM_DSP2_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5600
CM_DSP2_STATICDEP	RW	32	0x0000 0004	0x4A00 5604
CM_DSP2_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5608
CM_DSP2_DSP2_CLKCTRL	RW	32	0x0000 0020	0x4A00 5620

3.13.4.2 CM_CORE_AON__DSP2 Register Description

Table 3-518. CM_DSP2_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5600	**Instance**	CM_CORE_AON__DSP2
Description	This register enables the DSP domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:9	8	7:2	1:0
RESERVED	CLKACTIVITY_DSP2_GFCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_DSP2_GFCLK	This field indicates the state of the DSP_ROOT_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-519. Register Call Summary for Register CM_DSP2_CLKSTCTRL

Reset Management Functional Description

- DSP2 Subsystem Software Warm Reset Sequence: [0]

Clock Management Functional Description

- Clock Domain Modes: [1] [2]

PRCM Register Manual

- CM_CORE_AON__DSP2 Register Summary: [3]

Table 3-520. CM_DSP2_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5604	**Instance**	CM_CORE_AON__DSP2
Description	This register controls the static domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	ATL_STATDEP	PCIE_STATDEP	VPE_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	RESERVED	RESERVED	EVE2_STATDEP	EVE1_STATDEP	RESERVED	CUSTEFUSE_STATDEP	COREAON_STATDEP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ATL_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
29	PCIE_STATDEP	Static dependency towards PCIE cLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
28	VPE_STATDEP	Static dependency towards VPE Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
27	L4PER3_STATDEP	Static dependency towards L4PER3 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPUClock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2CLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 CLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	RESERVED		R	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	L4SEC_STATDEP	Static dependency towards L4SEC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled	R	0x0
11	RESERVED		R	0x0
10	GPU_STATDEP	Static dependency towards GPU Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-521. Register Call Summary for Register CM_DSP2_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24]

PRCM Register Manual

- CM_CORE_AON__DSP2 Register Summary: [27]

Table 3-522. CM_DSP2_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5608	**Instance**	CM_CORE_AON__DSP2
Description	This register controls the dynamic domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31:28	27:24	23:6	5	4:0
RESERVED	WINDOWSIZE	RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4

Bits	Field Name	Description	Type	Reset
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-523. Register Call Summary for Register CM_DSP2_DYNAMICDEP

PRCM Register Manual

- CM_CORE_AON__DSP2 Register Summary: [0]

Table 3-524. CM_DSP2_DSP2_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5620	**Instance**	CM_CORE_AON__DSP2
Description	This register manages the DSP clocks.		
Type	RW		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-525. Register Call Summary for Register CM_DSP2_DSP2_CLKCTRL

Reset Management Functional Description

- DSP2 Subsystem Software Warm Reset Sequence: [0] [1]

Clock Management Functional Description

- Clock Domain Module Attributes: [2] [3] [4]

Table 3-525. Register Call Summary for Register CM_DSP2_DSP2_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE_AON__DSP2 Register Summary: [5]

3.13.5 CM_CORE_AON__EVE1 Registers

3.13.5.1 CM_CORE_AON__EVE1 Register Summary

Table 3-526. CM_CORE_AON__EVE1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__EVE1 Physical Address L4_CFG Interconnect
CM_EVE1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5640
CM_EVE1_STATICDEP	RW	32	0x0000 0004	0x4A00 5644
CM_EVE1_EVE1_CLKCTRL	RW	32	0x0000 0020	0x4A00 5660

3.13.5.2 CM_CORE_AON__EVE1 Register Description

Table 3-527. CM_EVE1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5640	**Instance**	CM_CORE_AON__EVE1
Description	This register enables the EVE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:9	8	7:2	1:0
RESERVED	CLKACTIVITY_EVE1_GFCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EVE1_GFCLK	This field indicates the state of the EVE1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-528. Register Call Summary for Register CM_EVE1_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

PRCM Register Manual

- CM_CORE_AON__EVE1 Register Summary: [2]

Table 3-529. CM_EVE1_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5644	**Instance**	CM_CORE_AON__EVE1
Description	This register controls the static domain depedencies from EVE1 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									RESERVED	RESERVED	EVE2_STATDEP	RESERVED														L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	RESERVED		R	0x0

Table 3-530. Register Call Summary for Register CM_EVE1_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3]

PRCM Register Manual

- CM_CORE_AON__EVE1 Register Summary: [6]

Table 3-531. CM_EVE1_EVE1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5660	**Instance**	CM_CORE_AON__EVE1
Description	This register manages the EVE clocks.		
Type	RW		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-532. Register Call Summary for Register CM_EVE1_EVE1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE_AON__EVE1 Register Summary: [3]

3.13.6 CM_CORE_AON__EVE2 Registers

3.13.6.1 CM_CORE_AON__EVE2 Register Summary

Table 3-533. CM_CORE_AON__EVE2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__EVE2 Physical Address L4_CFG Interconnect
CM_EVE2_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5680
CM_EVE2_STATICDEP	RW	32	0x0000 0004	0x4A00 5684
CM_EVE2_EVE2_CLKCTRL	RW	32	0x0000 0020	0x4A00 56A0

3.13.6.2 CM_CORE_AON__EVE2 Register Description

Table 3-534. CM_EVE2_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5680	**Instance**	CM_CORE_AON__EVE2
Description	This register enables the EVE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31–9	8	7–2	1–0
RESERVED	CLKACTIVITY_EVE2_GFCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EVE2_GFCLK	This field indicates the state of the EVE2_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-535. Register Call Summary for Register CM_EVE2_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

Table 3-535. Register Call Summary for Register CM_EVE2_CLKSTCTRL (continued)

PRCM Register Manual

- CM_CORE_AON__EVE2 Register Summary: [2]

Table 3-536. CM_EVE2_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5684	**Instance**	CM_CORE_AON__EVE2
Description	This register controls the static domain depedencies from EVE2 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									RESERVED	RESERVED	RESERVED	EVE1_STATDEP	RESERVED													L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	RESERVED		R	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-537. Register Call Summary for Register CM_EVE2_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3]

PRCM Register Manual

- CM_CORE_AON__EVE2 Register Summary: [6]

Table 3-538. CM_EVE2_EVE2_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 56A0	**Instance**	CM_CORE_AON__EVE2
Description	This register manages the EVE clocks.		

Table 3-538. CM_EVE2_EVE2_CLKCTRL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-539. Register Call Summary for Register CM_EVE2_EVE2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE_AON__EVE2 Register Summary: [3]

3.13.7 CM_CORE_AON__INSTR Registers

3.13.7.1 CM_CORE_AON__INSTR Register Summary

Table 3-540. CM_CORE_AON__INSTR Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__INSTR Physical Address L4_CFG Interconnect
CMI_IDENTICATION	R	32	0x0000 0000	0x4A00 5F00
CMI_SYS_CONFIG	RW	32	0x0000 0010	0x4A00 5F10
CMI_STATUS	R	32	0x0000 0014	0x4A00 5F14
CMI_CONFIGURATION	RW	32	0x0000 0024	0x4A00 5F24
CMI_CLASS_FILTERING	RW	32	0x0000 0028	0x4A00 5F28
CMI_TRIGGERING	RW	32	0x0000 002C	0x4A00 5F2C

Table 3-540. CM_CORE_AON__INSTR Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__INSTR Physical Address L4_CFG Interconnect
CMI_SAMPLING	RW	32	0x0000 0030	0x4A00 5F30

3.13.7.2 CM_CORE_AON__INSTR Register Description

Table 3-541. CMI_IDENTICATION

Address Offset	0x0000 0000		
Physical Address	0x4A00 5F00	**Instance**	CM_CORE_AON__INSTR
Description	CM profiling identification register		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	0x- (1)

(1) TI internal data

Table 3-542. Register Call Summary for Register CMI_IDENTICATION

PRCM Register Manual

- CM_CORE_AON__INSTR Register Summary: [0]

Table 3-543. CMI_SYS_CONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A00 5F10	**Instance**	CM_CORE_AON__INSTR
Description	CM profiling system configuartion register		
Type	RW		

31–6	5–4	3–2	1	0
RESERVED	RESERVED	IDLEMODE	RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local tartget state management mode	RW	0x2
1	RESERVED		R	0x0
0	SOFTRESET	Software reset	RW	0x0

Table 3-544. Register Call Summary for Register CMI_SYS_CONFIG

PRCM Register Manual

- CM_CORE_AON__INSTR Register Summary: [0]

Table 3-545. CMI_STATUS

Address Offset	0x0000 0014		
Physical Address	0x4A00 5F14	**Instance**	CM_CORE_AON__INSTR
Description	CM profiling status register		

Table 3-545. CMI_STATUS (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							FIFOEMPTY	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	FIFOEMPTY	PM Profiling buffer empty	R	0x1
7:0	RESERVED		R	0x0

Table 3-546. Register Call Summary for Register CMI_STATUS

PRCM Register Manual

- CM_CORE_AON__INSTR Register Summary: [0]

Table 3-547. CMI_CONFIGURATION

Address Offset	0x0000 0024		
Physical Address	0x4A00 5F24	**Instance**	CM_CORE_AON__INSTR
Description	CM profiling configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLAIM_3		CLAIM_2	CLAIM_1	RESERVED				RESERVED	RESERVED							MOD_ACT_EN	RESERVED							EVT_CAPT_EN	RESERVED						

Bits	Field Name	Description	Type	Reset
31:30	CLAIM_3	Ownership	RW	0x0
29	CLAIM_2	Debugger override qualifier	RW	0x1
28	CLAIM_1	Current owner	R	0x0
27:24	RESERVED		R	0x0
23	RESERVED		R	0x0
22:16	RESERVED		R	0x0
15	MOD_ACT_EN	When HIGH the CM Module Activity collection is enabled	RW	0x0
14:8	RESERVED		R	0x0
7	EVT_CAPT_EN	When HIGH the CM events capture is enabled	RW	0x0
6:0	RESERVED		R	0x0

Table 3-548. Register Call Summary for Register CMI_CONFIGURATION

PRCM Register Manual

- CM_CORE_AON__INSTR Register Summary: [0]

Table 3-549. CMI_CLASS_FILTERING

Address Offset	0x0000 0028		
Physical Address	0x4A00 5F28	**Instance**	CM_CORE_AON__INSTR

Table 3-549. CMI_CLASS_FILTERING (continued)

Description	CM profiling class filtering register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SNAP_CAPT_EN_1F	SNAP_CAPT_EN_1E	SNAP_CAPT_EN_1D	SNAP_CAPT_EN_1C	SNAP_CAPT_EN_1B	SNAP_CAPT_EN_1A	SNAP_CAPT_EN_19	SNAP_CAPT_EN_18	SNAP_CAPT_EN_17	SNAP_CAPT_EN_16	SNAP_CAPT_EN_15	SNAP_CAPT_EN_14	SNAP_CAPT_EN_13	SNAP_CAPT_EN_12	SNAP_CAPT_EN_11	SNAP_CAPT_EN_10	RESERVED												SNAP_CAPT_EN_03	SNAP_CAPT_EN_02	SNAP_CAPT_EN_01	SNAP_CAPT_EN_00

Bits	Field Name	Description	Type	Reset
31	SNAP_CAPT_EN_1F	Snapshot capture enable - Class-ID = 0x1F	RW	0x0
30	SNAP_CAPT_EN_1E		RW	0x0
29	SNAP_CAPT_EN_1D		RW	0x0
28	SNAP_CAPT_EN_1C		RW	0x0
27	SNAP_CAPT_EN_1B		RW	0x0
26	SNAP_CAPT_EN_1A		RW	0x0
25	SNAP_CAPT_EN_19		RW	0x0
24	SNAP_CAPT_EN_18		RW	0x0
23	SNAP_CAPT_EN_17		RW	0x0
22	SNAP_CAPT_EN_16		RW	0x0
21	SNAP_CAPT_EN_15		RW	0x0
20	SNAP_CAPT_EN_14		RW	0x0
19	SNAP_CAPT_EN_13		RW	0x0
18	SNAP_CAPT_EN_12		RW	0x0
17	SNAP_CAPT_EN_11		RW	0x0
16	SNAP_CAPT_EN_10	Snapshot capture enable - Class-ID = 0x10	RW	0x0
15:4	RESERVED		R	0x0
3	SNAP_CAPT_EN_03	Snapshot capture enable - Class-ID = 0x03 [0x23]	RW	0x0
2	SNAP_CAPT_EN_02	Snapshot capture enable - Class-ID = 0x02 [0x22]	RW	0x0
1	SNAP_CAPT_EN_01	Snapshot capture enable - Class-ID = 0x01 [0x21]	RW	0x0
0	SNAP_CAPT_EN_00	Snapshot capture enable - Class-ID = 0x00 [0x20]	RW	0x0

Table 3-550. Register Call Summary for Register CMI_CLASS_FILTERING

PRCM Register Manual

- CM_CORE_AON__INSTR Register Summary: [0]

Table 3-551. CMI_TRIGGERING

Address Offset	0x0000 002C		
Physical Address	0x4A00 5F2C	**Instance**	CM_CORE_AON__INSTR
Description	CM profiling triggering control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	TRIG_STOP_EN / TRIG_START_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	TRIG_STOP_EN	Enable stop capturing CM events from external trigger detection	RW	0x0
0	TRIG_START_EN	Enable start capturing CM events from external trigger detection	RW	0x0

Table 3-552. Register Call Summary for Register CMI_TRIGGERING

PRCM Register Manual

- CM_CORE_AON__INSTR Register Summary: [0]

Table 3-553. CMI_SAMPLING

Address Offset	0x0000 0030		
Physical Address	0x4A00 5F30	**Instance**	CM_CORE_AON__INSTR
Description	CM profiling sampling window register		
Type	RW		

31–20	19–16	15–8	7–0
RESERVED	FCLK_DIV_FACOR	RESERVED	SAMP_WIND_SIZE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:16	FCLK_DIV_FACOR	FunClk divide factor ranging from 1 to 16	RW	0x0
15:8	RESERVED		R	0x0
7:0	SAMP_WIND_SIZE	CM events sampling window size	RW	0x0

Table 3-554. Register Call Summary for Register CMI_SAMPLING

PRCM Register Manual

- CM_CORE_AON__INSTR Register Summary: [0]

3.13.8 CM_CORE_AON__IPU Registers

3.13.8.1 CM_CORE_AON__IPU Register Summary

Table 3-555. CM_CORE_AON__IPU Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__IPU Physical Address L4_CFG Interconnect
CM_IPU1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5500
CM_IPU1_STATICDEP	RW	32	0x0000 0004	0x4A00 5504
CM_IPU1_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5508
CM_IPU1_IPU1_CLKCTRL	RW	32	0x0000 0020	0x4A00 5520
CM_IPU_CLKSTCTRL	RW	32	0x0000 0040	0x4A00 5540
CM_IPU_MCASP1_CLKCTRL	RW	32	0x0000 0050	0x4A00 5550
CM_IPU_TIMER5_CLKCTRL	RW	32	0x0000 0058	0x4A00 5558
CM_IPU_TIMER6_CLKCTRL	RW	32	0x0000 0060	0x4A00 5560
CM_IPU_TIMER7_CLKCTRL	RW	32	0x0000 0068	0x4A00 5568

Table 3-555. CM_CORE_AON__IPU Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__IPU Physical Address L4_CFG Interconnect
CM_IPU_TIMER8_CLKCTRL	RW	32	0x0000 0070	0x4A00 5570
CM_IPU_I2C5_CLKCTRL	RW	32	0x0000 0078	0x4A00 5578
CM_IPU_UART6_CLKCTRL	RW	32	0x0000 0080	0x4A00 5580

3.13.8.2 CM_CORE_AON__IPU Register Description

Table 3-556. CM_IPU1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5500	**Instance**	CM_CORE_AON__IPU
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:9	8	7:2	1:0
RESERVED	CLKACTIVITY_IPU1_GFCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_IPU1_GFCLK	This field indicates the state of the IPU1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the IPU1 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-557. Register Call Summary for Register CM_IPU1_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [2]

Table 3-558. CM_IPU1_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5504	**Instance**	CM_CORE_AON__IPU
Description	This register controls the static domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	ATL_STATDEP	PCIE_STATDEP	VPE_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	RESERVED	RESERVED	RESERVED	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ATL_STATDEP	Static dependency towards ATL clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	VPE_STATDEP	Static dependency towards VPE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	RESERVED		R	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11	SDMA_STATDEP	Static dependency towards DMA clock domain 0x0: Dependency is disabled	R	0x0
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-559. Register Call Summary for Register CM_IPU1_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25]
- Clock Domain Dependency: [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [56]

Table 3-560. CM_IPU1_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5508	**Instance**	CM_CORE_AON__IPU
Description	This register controls the dynamic domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31:28	27:24	23:6	5	4:0
RESERVED	WINDOWSIZE	RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-561. Register Call Summary for Register CM_IPU1_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]
- Clock Domain Dependency: [1]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [2]

Table 3-562. CM_IPU1_IPU1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5520	**Instance**	CM_CORE_AON__IPU
Description	This register manages the IPU1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects the timer functional clock 0x0: Selects DPLL_ABE_X2_CLK as the functional clock 0x1: Selects CORE_IPU_ISS_BOOST_CLK as the functional clock	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-563. Register Call Summary for Register CM_IPU1_IPU1_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2] [3] [4]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [5]

Table 3-564. CM_IPU_CLKSTCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 5540	**Instance**	CM_CORE_AON__IPU
Description	This register enables the ABE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLKACTIVITY_MCASP1_AHCLKR	CLKACTIVITY_MCASP1_AHCLKX	CLKACTIVITY_MCASP1_AUX_GFCLK	RESERVED	CLKACTIVITY_UART6_GFCLK	CLKACTIVITY_IPU_96M_GFCLK	CLKACTIVITY_TIMER8_GFCLK	CLKACTIVITY_TIMER7_GFCLK	CLKACTIVITY_TIMER6_GFCLK	CLKACTIVITY_TIMER5_GFCLK	CLKACTIVITY_IPU_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	CLKACTIVITY_MCASP1_AHCLKR	This field indicates the state of the MCASP1_AHCLKR clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_MCASP1_AHCLKX	This field indicates the state of the MCASP1_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_MCASP1_AUX_GFCLK	This field indicates the state of the MCASP1_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	RESERVED		R	0x0
14	CLKACTIVITY_UART6_GFCLK	This field indicates the state of the UART6_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_IPU_96M_GFCLK	This field indicates the state of the IPU_96M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_TIMER8_GFCLK	This field indicates the state of the TIMER8_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_TIMER7_GFCLK	This field indicates the state of the TIMER7_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_TIMER6_GFCLK	This field indicates the state of the TIMER6_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_TIMER5_GFCLK	This field indicates the state of the TIMER5_GFCLK functional clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_IPU_L3_GICLK	This field indicates the state of the IPU_L3_GICLK interface clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the ABE clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-565. Register Call Summary for Register CM_IPU_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [11]

Table 3-566. CM_IPU_MCASP1_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 5550	**Instance**	CM_CORE_AON__IPU
Description	This register manages the McASP clocks.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">CLKSEL_AHCLKR</td><td colspan="4">CLKSEL_AHCLKX</td><td colspan="2">CLKSEL_AUX_CLK</td><td colspan="4">RESERVED</td><td colspan="2">IDLEST</td><td colspan="14">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	CLKSEL_AHCLKR	Selects reference clock for AHCLKR 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-567. Register Call Summary for Register CM_IPU_MCASP1_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1] [2]
- Clock Domain Module Attributes: [3] [4]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [5]

Table 3-568. CM_IPU_TIMER5_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 5558	**Instance**	CM_CORE_AON__IPU
Description	This register manages the TIMER5 clocks.		
Type	RW		

31:28	27:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Selects the timer functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB: Selects CLKOUTMUX0_CLK 0xC-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-569. Register Call Summary for Register CM_IPU_TIMER5_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [3]

Table 3-570. CM_IPU_TIMER6_CLKCTRL

Address Offset	0x0000 0060		
Physical Address	0x4A00 5560	**Instance**	CM_CORE_AON__IPU
Description	This register manages the TIMER6 clocks.		
Type	RW		

31–28	27–24	23–18	17–16	15–2	1–0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27:24	CLKSEL	Selects the timer functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV _CLK 0xB: Selects CLKOUTMUX0_CLK 0xC-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-571. Register Call Summary for Register CM_IPU_TIMER6_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [3]

Table 3-572. CM_IPU_TIMER7_CLKCTRL

Address Offset	0x0000 0068		
Physical Address	0x4A00 5568	**Instance**	CM_CORE_AON__IPU
Description	This register manages the TIMER7 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Selects the timer functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV _CLK 0xB: Selects CLKOUTMUX0_CLK 0xC-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-573. Register Call Summary for Register CM_IPU_TIMER7_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [3]

Table 3-574. CM_IPU_TIMER8_CLKCTRL

Address Offset	0x0000 0070		
Physical Address	0x4A00 5570	**Instance**	CM_CORE_AON__IPU
Description	This register manages the TIMER8 clocks.		

Table 3-574. CM_IPU_TIMER8_CLKCTRL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Selects the timer functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV _CLK 0xB: Selects CLKOUTMUX0_CLK 0xC-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-575. Register Call Summary for Register CM_IPU_TIMER8_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [3]

Table 3-576. CM_IPU_I2C5_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 5578	**Instance**	CM_CORE_AON__IPU
Description	This register manages the I2C5 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-577. Register Call Summary for Register CM_IPU_I2C5_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE_AON__IPU Register Summary: [2]

Table 3-578. CM_IPU_UART6_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4A00 5580	**Instance**	CM_CORE_AON__IPU
Description	This register manages the UART6 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-579. Register Call Summary for Register CM_IPU_UART6_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual
- CM_CORE_AON__IPU Register Summary: [3]

3.13.9 CM_CORE_AON__MPU Registers

3.13.9.1 CM_CORE_AON__MPU Register Summary

Table 3-580. CM_CORE_AON__MPU Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__MPU Physical Address L4_CFG Interconnect
CM_MPU_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5300
CM_MPU_STATICDEP	RW	32	0x0000 0004	0x4A00 5304
CM_MPU_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5308
CM_MPU_MPU_CLKCTRL	RW	32	0x0000 0020	0x4A00 5320
CM_MPU_MPU_MPU_DBG_CLKCTRL	R	32	0x0000 0028	0x4A00 5328

3.13.9.2 CM_CORE_AON__MPU Register Description

Table 3-581. CM_MPU_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5300	**Instance**	CM_CORE_AON__MPU
Description	This register enables the MPU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		

Table 3-581. CM_MPU_CLKSTCTRL (continued)

Type	RW

31–9	8	7–2	1–0
RESERVED	CLKACTIVITY_MPU_GCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_MPU_GCLK	This field indicates the state of the MPU_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the MPU clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-582. CM_MPU_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5304	**Instance**	CM_CORE_AON__MPU
Description	This register controls the static domain depedencies from MPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31–30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	PCIE_STATDEP	VPE_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	RESERVED	RESERVED	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	VPE_STATDEP	Static dependency towards VPE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11	SDMA_STATDEP	Static dependency towards SDMA clock domain 0x0: Dependency is disabled	R	0x0
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-583. CM_MPU_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5308	**Instance**	CM_CORE_AON__MPU
Description	This register controls the dynamic domain depedencies from MPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	WINDOWSIZE	RESERVED	L3MAIN1_DYNDEP	EMIF_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_DYNDEP	Dynamic dependency towards EMIF clock domain 0x1: Dependency is enabled	R	0x1
3:0	RESERVED		R	0x0

Table 3-584. CM_MPU_MPU_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5320	**Instance**	CM_CORE_AON__MPU
Description	This register manages the MPU clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_ABE_DIV_MODE	CLKSEL_EMIF_DIV_MODE		RESERVED					STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	CLKSEL_ABE_DIV_MODE	Selects the ratio for MPU - ABE async bridge versus MPU DPLL clock 0x0: MPU DPLL clock divided by 8 0x1: MPU DPLL clock divided by 16	RW	0x0
25:24	CLKSEL_EMIF_DIV_MODE	Selects the ratio for MPU - L3 async bridge versus MPU DPLL clock 0x0: MPU DPLL clock divided by 4 0x1: MPU DPLL clock divided by 4 0x2: MPU DPLL clock divided by 8 0x3: MPU DPLL clock divided by 8	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-585. CM_MPU_MPU_MPU_DBG_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 5328	**Instance**	CM_CORE_AON__MPU
Description	This register manages the MPU_MPU_DBG clocks. [warm reset insensitive]		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

3.13.10 CM_CORE_AON__OCP_SOCKET Registers

3.13.10.1 CM_CORE_AON__OCP_SOCKET Register Summary

Table 3-586. CM_CORE_AON__OCP_SOCKET Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__OCP_SOCKET Physical Address L4_CFG Interconnect
REVISION_CM_CORE_AON	R	32	0x0000 0000	0x4A00 5000
CM_CM_CORE_AON_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4A00 5040
CM_CORE_AON_DEBUG_OUT	R	32	0x0000 00EC	0x4A00 50EC

Table 3-586. CM_CORE_AON__OCP_SOCKET Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__OCP_SOCKET Physical Address L4_CFG Interconnect
CM_CORE_AON_DEBUG_CFG0	RW	32	0x0000 00F0	0x4A00 50F0
CM_CORE_AON_DEBUG_CFG1	RW	32	0x0000 00F4	0x4A00 50F4
CM_CORE_AON_DEBUG_CFG2	RW	32	0x0000 00F8	0x4A00 50F8
CM_CORE_AON_DEBUG_CFG3	RW	32	0x0000 00FC	0x4A00 50FC

3.13.10.2 CM_CORE_AON__OCP_SOCKET Register Description

Table 3-587. REVISION_CM_CORE_AON

Address Offset	0x0000 0000		
Physical Address	0x4A00 5000	**Instance**	CM_CORE_AON__OCP_SOCKET
Description	This register contains the IP revision code for the CM_CORE_AON part of the PRCM		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision number	R	0x- [1]

(1) TI internal data

Table 3-588. CM_CM_CORE_AON_PROFILING_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 5040	**Instance**	CM_CORE_AON__OCP_SOCKET
Description	This register manages the CM_CORE_AON_PROFILING clock. [warm reset insensitive]		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. OCP configuration port is not accessible. 0x1: Module is managed automatically by HW along with CM_CORE_AON and EMU domain. OCP configuration port is accessible only when EMU domain is on. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-589. CM_CORE_AON_DEBUG_OUT

Address Offset	0x0000 00EC		
Physical Address	0x4A00 50EC	**Instance**	CM_CORE_AON__OCP_SOCKET
Description	This register is used to monitor the CM_COREAON's 32 bit HEDEBUG BUS [warm reset insensitive]		
Type	R		

31–0
OUTPUT

Bits	Field Name	Description	Type	Reset
31:0	OUTPUT	HW DEBUG OUTPUT	R	0x0

Table 3-590. CM_CORE_AON_DEBUG_CFG0

Address Offset	0x0000 00F0		
Physical Address	0x4A00 50F0	**Instance**	CM_CORE_AON__OCP_SOCKET
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31–10	9–0
RESERVED	SEL0

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	SEL0	Internal signal block select for debug word byte-0	RW	0x0

Table 3-591. CM_CORE_AON_DEBUG_CFG1

Address Offset	0x0000 00F4		
Physical Address	0x4A00 50F4	**Instance**	CM_CORE_AON__OCP_SOCKET
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31–10	9–0
RESERVED	SEL1

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	SEL1	Internal signal block select for debug word byte-1	RW	0x0

Table 3-592. CM_CORE_AON_DEBUG_CFG2

Address Offset	0x0000 00F8		
Physical Address	0x4A00 50F8	**Instance**	CM_CORE_AON__OCP_SOCKET
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31:10	9:0
RESERVED	SEL2

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	SEL2	Internal signal block select for debug word byte-2	RW	0x0

Table 3-593. CM_CORE_AON_DEBUG_CFG3

Address Offset	0x0000 00FC		
Physical Address	0x4A00 50FC	**Instance**	CM_CORE_AON__OCP_SOCKET
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31:10	9:0
RESERVED	SEL3

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	SEL3	Internal signal block select for debug word byte-3	RW	0x0

3.13.11 CM_CORE_AON__RESTORE Registers

3.13.11.1 CM_CORE_AON__RESTORE Register Summary

Table 3-594. CM_CORE_AON__RESTORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__RESTORE Physical Address L4_CFG Interconnect
CM_CLKSEL_CORE_RESTORE	RW	32	0x0000 0000	0x4A00 5E00
CM_DIV_M2_DPLL_CORE_RESTORE	RW	32	0x0000 0004	0x4A00 5E04
RESERVED	RW	32	0x0000 0008	0x4A00 5E08
RESERVED	RW	32	0x0000 000C	0x4A00 5E0C
CM_DIV_H12_DPLL_CORE_RESTORE	RW	32	0x0000 0010	0x4A00 5E10
CM_DIV_H13_DPLL_CORE_RESTORE	RW	32	0x0000 0014	0x4A00 5E14
CM_DIV_H14_DPLL_CORE_RESTORE	RW	32	0x0000 0018	0x4A00 5E18
RESERVED	RW	32	0x0000 001C	0x4A00 5E1C
CM_DIV_H22_DPLL_CORE_RESTORE	RW	32	0x0000 0020	0x4A00 5E20
CM_DIV_H23_DPLL_CORE_RESTORE	RW	32	0x0000 0024	0x4A00 5E24
CM_DIV_H24_DPLL_CORE_RESTORE	RW	32	0x0000 0028	0x4A00 5E28

Table 3-594. CM_CORE_AON__RESTORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__RESTORE Physical Address L4_CFG Interconnect
CM_CLKSEL_DPLL_CORE_RESTORE	RW	32	0x0000 002C	0x4A00 5E2C
RESERVED	R	32	0x0000 0030	0x4A00 5E30
RESERVED	R	32	0x0000 0034	0x4A00 5E34
CM_CLKMODE_DPLL_CORE_RESTORE	RW	32	0x0000 0038	0x4A00 5E38
CM_SHADOW_FREQ_CONFIG2_RESTORE	RW	32	0x0000 003C	0x4A00 5E3C
CM_SHADOW_FREQ_CONFIG1_RESTORE	RW	32	0x0000 0040	0x4A00 5E40
CM_AUTOIDLE_DPLL_CORE_RESTORE	RW	32	0x0000 0044	0x4A00 5E44
CM_MPU_CLKSTCTRL_RESTORE	RW	32	0x0000 0048	0x4A00 5E48
CM_CM_CORE_AON_PROFILING_CLKCTRL_RESTORE	RW	32	0x0000 004C	0x4A00 5E4C
CM_DYN_DEP_PRESCAL_RESTORE	RW	32	0x0000 0050	0x4A00 5E50

3.13.11.2 CM_CORE_AON__RESTORE Register Description

Table 3-595. CM_CLKSEL_CORE_RESTORE

Address Offset	0x0000 0000		
Physical Address	0x4A00 5E00	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_CLKSEL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CLKSEL_CORE register.	RW	0x0

Table 3-596. CM_DIV_M2_DPLL_CORE_RESTORE

Address Offset	0x0000 0004		
Physical Address	0x4A00 5E04	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DIV_M2_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_M2_DPLL_CORE register.	RW	0x1

Table 3-597. CM_DIV_H12_DPLL_CORE_RESTORE

Address Offset	0x0000 0010		
Physical Address	0x4A00 5E10	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DIV_H12_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H12_DPLL_CORE register.	RW	0x4

Table 3-598. CM_DIV_H13_DPLL_CORE_RESTORE

Address Offset	0x0000 0014		
Physical Address	0x4A00 5E14	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DIV_H13_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H13_DPLL_CORE register.	RW	0x4

Table 3-599. CM_DIV_H14_DPLL_CORE_RESTORE

Address Offset	0x0000 0018		
Physical Address	0x4A00 5E18	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DIV_H14_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H14_DPLL_CORE register.	RW	0x4

Table 3-600. CM_DIV_H22_DPLL_CORE_RESTORE

Address Offset	0x0000 0020		
Physical Address	0x4A00 5E20	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DIV_H22_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H22_DPLL_CORE register.	RW	0x4

Table 3-601. CM_DIV_H23_DPLL_CORE_RESTORE

Address Offset	0x0000 0024		
Physical Address	0x4A00 5E24	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DIV_H23_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H23_DPLL_CORE register.	RW	0x8

Table 3-602. CM_DIV_H24_DPLL_CORE_RESTORE

Address Offset	0x0000 0028		
Physical Address	0x4A00 5E28	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DIV_H24_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H24_DPLL_CORE register.	RW	0x8

Table 3-603. CM_CLKSEL_DPLL_CORE_RESTORE

Address Offset	0x0000 002C		
Physical Address	0x4A00 5E2C	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_CLKSEL_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CLKSEL_DPLL_CORE register.	RW	0x0

Table 3-604. CM_CLKMODE_DPLL_CORE_RESTORE

Address Offset	0x0000 0038		
Physical Address	0x4A00 5E38	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_CLKMODE_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CLKMODE_DPLL_CORE register.	RW	0x4

Table 3-605. CM_SHADOW_FREQ_CONFIG2_RESTORE

Address Offset	0x0000 003C		
Physical Address	0x4A00 5E3C	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_SHADOW_FREQ_CONFIG2. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_SHADOW_FREQ_CONFIG2 register.	RW	0x20

Table 3-606. CM_SHADOW_FREQ_CONFIG1_RESTORE

Address Offset	0x0000 0040		
Physical Address	0x4A00 5E40	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_SHADOW_FREQ_CONFIG1. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_SHADOW_FREQ_CONFIG1 register.	RW	0xc0c

Table 3-607. CM_AUTOIDLE_DPLL_CORE_RESTORE

Address Offset	0x0000 0044		
Physical Address	0x4A00 5E44	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_AUTOIDLE_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_AUTOIDLE_DPLL_COREregister.	RW	0x0

Table 3-608. CM_MPU_CLKSTCTRL_RESTORE

Address Offset	0x0000 0048		
Physical Address	0x4A00 5E48	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_MPU_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_MPU_CLKSTCTRL register.	RW	0x0

Table 3-609. CM_CM_CORE_AON_PROFILING_CLKCTRL_RESTORE

Address Offset	0x0000 004C		
Physical Address	0x4A00 5E4C	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_CM_CORE_AON_PROFILING_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CM_CORE_AON_PROFILING_CLKCTRL register.	RW	0x30001

Table 3-610. CM_DYN_DEP_PRESCAL_RESTORE

Address Offset	0x0000 0050		
Physical Address	0x4A00 5E50	**Instance**	CM_CORE_AON__RESTORE
Description	Second address map for register CM_DYN_DEP_PRESCAL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31–0
RESTORE

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CCM_DYN_DEP_PRESCAL register.	RW	0x20

3.13.12 CM_CORE_AON__RTC Registers

3.13.12.1 CM_CORE_AON__RTC Register Summary

Table 3-611. CM_CORE_AON__RTC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__RTC Physical Address L4_CFG Interconnect
CM_RTC_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5740
CM_RTC_RTCSS_CLKCTRL	RW	32	0x0000 0004	0x4A00 5744

3.13.12.2 CM_CORE_AON__RTC Register Description

Table 3-612. CM_RTC_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5740	**Instance**	CM_CORE_AON__RTC
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31–11	10	9	8	7–2	1–0
RESERVED	CLKACTIVITY_RTC_AUX_CLK	RESERVED	CLKACTIVITY_RTC_L4_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_RTC_AUX_CLK	This field indicates the state of the RTC_AUX_CLK in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	RESERVED		R	0x0
8	CLKACTIVITY_RTC_L4_GICLK	This field indicates the state of the RTC_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the WKUPAON clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-613. Register Call Summary for Register CM_RTC_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2]

PRCM Register Manual

- CM_CORE_AON__RTC Register Summary: [3]

Table 3-614. CM_RTC_RTCSS_CLKCTRL

Address Offset	0x0000 0004		
Physical Address	0x4A00 5744	**Instance**	CM_CORE_AON__RTC
Description	This register manages the RTC clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-615. Register Call Summary for Register CM_RTC_RTCSS_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE_AON__RTC Register Summary: [2]

3.13.13 CM_CORE_AON__VPE Registers

3.13.13.1 CM_CORE_AON__VPE Register Summary

Table 3-616. CM_CORE_AON__VPE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE_AON__VPE Physical Address L4_CFG Interconnect
CM_VPE_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5760
CM_VPE_VPE_CLKCTRL	RW	32	0x0000 0004	0x4A00 5764
CM_VPE_STATICDEP	RW	32	0x0000 0008	0x4A00 5768

3.13.13.2 CM_CORE_AON__VPE Register Description

Table 3-617. CM_VPE_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5760	**Instance**	CM_CORE_AON__VPE
Description	This register enables the VPE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>CLKACTIVITY_VPE_GCLK</td><td colspan="6">RESERVED</td><td colspan="2">CLKTRCTRL</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	CLKACTIVITY_VPE_GCLK	This field indicates the state of the VPE_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-618. Register Call Summary for Register CM_VPE_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

PRCM Register Manual

- CM_CORE_AON__VPE Register Summary: [2]

Table 3-619. CM_VPE_VPE_CLKCTRL

Address Offset	0x0000 0004		
Physical Address	0x4A00 5764	**Instance**	CM_CORE_AON__VPE
Description	This register manages the VPE clocks.		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19	18	17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2	1 0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-620. Register Call Summary for Register CM_VPE_VPE_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE_AON__VPE Register Summary: [3]

Table 3-621. CM_VPE_STATICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5768	**Instance**	CM_CORE_AON__VPE
Description	This register controls the static domain depedencies from VPE domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>L4PER3_STATDEP</td><td colspan="21">RESERVED</td><td>L3MAIN1_STATDEP</td><td>EMIF_STATDEP</td><td colspan="4">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3:0	RESERVED		R	0x0

Table 3-622. Register Call Summary for Register CM_VPE_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2]

Table 3-622. Register Call Summary for Register CM_VPE_STATICDEP (continued)

PRCM Register Manual

- CM_CORE_AON__VPE Register Summary: [3]

3.13.14 CM_CORE__CAM Registers

3.13.14.1 CM_CORE__CAM Register Summary

Table 3-623. CM_CORE__CAM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__CAM Physical Address L4_CFG Interconnect
CM_CAM_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9000
CM_CAM_STATICDEP	RW	32	0x0000 0004	0x4A00 9004
CM_CAM_VIP1_CLKCTRL	RW	32	0x0000 0020	0x4A00 9020
CM_CAM_VIP2_CLKCTRL	RW	32	0x0000 0028	0x4A00 9028
CM_CAM_VIP3_CLKCTRL	RW	32	0x0000 0030	0x4A00 9030
RESERVED	R	32	0x0000 0038	0x4A00 9038
RESERVED	R	32	0x0000 0040	0x4A00 9040
RESERVED	R	32	0x0000 0048	0x4A00 9048

3.13.14.2 CM_CORE__CAM Register Description

Table 3-624. CM_CAM_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9000	**Instance**	CM_CORE__CAM
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:13	12	11	10	9	8	7:2	1:0
RESERVED	RESERVED	RESERVED	CLKACTIVITY_VIP3_GCLK	CLKACTIVITY_VIP2_GCLK	CLKACTIVITY_VIP1_GCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	RESERVED		R	0x0
11	RESERVED		R	0x0
10	CLKACTIVITY_VIP3_GCLK	This field indicates the state of the VIP3_GCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
9	CLKACTIVITY_VIP2_GCLK	This field indicates the state of the VIP2_GCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_VIP1_GCLK	This field indicates the state of the VIP1_GCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the CAM clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-625. CM_CAM_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9004	**Instance**	CM_CORE__CAM
Description	This register controls the static domain depedencies from CAM domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31:29	28	27	26	25	24:23	22	21	20	19	18:13	12	11:6	5	4	3	2	1:0
RESERVED	VPE_STATDEP	L4PER3_STATDEP	RESERVED	GMAC_STATDEP	RESERVED	RESERVED	RESERVED	EVE2_STATDEP	EVE1_STATDEP	RESERVED	L4CFG_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	VPE_STATDEP	Static dependency towards VPE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	RESERVED		R	0x0
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24:23	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18:13	RESERVED		R	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled	R	0x0
11:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-626. CM_CAM_VIP1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9020	**Instance**	CM_CORE__CAM
Description	This register manages the VIP1 clocks.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>CLKSEL</td><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">IDLEST</td><td colspan="14">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for VIP between L3_ICLK and CORE_ISS_MAIN_CLK 0x0: Selects L3_ICLK 0x1: Selects CORE_ISS_MAIN_CLK	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-627. CM_CAM_VIP2_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 9028	**Instance**	CM_CORE__CAM
Description	This register manages the VIP2 clocks.		
Type	RW		

31:25	24	23:19	18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for VIP between L3_ICLK and CORE_ISS_MAIN_CLK 0x0: Selects L3_ICLK 0x1: Selects CORE_ISS_MAIN_CLK	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-628. CM_CAM_VIP3_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9030	**Instance**	CM_CORE__CAM
Description	This register manages the VIP3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for VIP between L3_ICLK and CORE_ISS_MAIN_CLK 0x0: Selects L3_ICLK 0x1: Selects CORE_ISS_MAIN_CLK	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-629. CM_CAM_CSI1_CLKCTRL

Address Offset	0x0000 0040	
Physical Address		**Instance**
Description	This register manages the CSI1 clocks.	
Type	R	

31–19	18	17–16	15–2	1–0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x3: Module is disabled and cannot be accessed 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-630. CM_CAM_CSI2_CLKCTRL

Address Offset	0x0000 0048	
Physical Address		**Instance**
Description	This register manages the CSI2 clocks.	
Type	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x3: Module is disabled and cannot be accessed 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

3.13.15 CM_CORE__CKGEN Registers

3.13.15.1 CM_CORE__CKGEN Register Summary

Table 3-631. CM_CORE__CKGEN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__CKGEN Physical Address L4_CFG Interconnect
CM_CLKSEL_USB_60MHZ	RW	32	0x0000 0004	0x4A00 8104
CM_CLKMODE_DPLL_PER	RW	32	0x0000 0040	0x4A00 8140
CM_IDLEST_DPLL_PER	R	32	0x0000 0044	0x4A00 8144
CM_AUTOIDLE_DPLL_PER	RW	32	0x0000 0048	0x4A00 8148
CM_CLKSEL_DPLL_PER	RW	32	0x0000 004C	0x4A00 814C
CM_DIV_M2_DPLL_PER	RW	32	0x0000 0050	0x4A00 8150
RESERVED	R	32	0x0000 0054	0x4A00 8154
CM_DIV_H11_DPLL_PER	RW	32	0x0000 0058	0x4A00 8158
CM_DIV_H12_DPLL_PER	RW	32	0x0000 005C	0x4A00 815C
CM_DIV_H13_DPLL_PER	RW	32	0x0000 0060	0x4A00 8160
CM_DIV_H14_DPLL_PER	RW	32	0x0000 0064	0x4A00 8164
RESERVED	R	32	0x0000 0068	0x4A00 8168
RESERVED	R	32	0x0000 006C	0x4A00 816C
CM_CLKMODE_DPLL_USB	RW	32	0x0000 0080	0x4A00 8180
CM_IDLEST_DPLL_USB	R	32	0x0000 0084	0x4A00 8184
CM_AUTOIDLE_DPLL_USB	RW	32	0x0000 0088	0x4A00 8188
CM_CLKSEL_DPLL_USB	RW	32	0x0000 008C	0x4A00 818C
CM_DIV_M2_DPLL_USB	RW	32	0x0000 0090	0x4A00 8190
RESERVED	R	32	0x0000 00A8	0x4A00 81A8
RESERVED	R	32	0x0000 00AC	0x4A00 81AC
CM_CLKDCOLDO_DPLL_USB	R	32	0x0000 00B4	0x4A00 81B4
CM_CLKMODE_DPLL_PCIE_REF	RW	32	0x0000 0100	0x4A00 8200
CM_IDLEST_DPLL_PCIE_REF	R	32	0x0000 0104	0x4A00 8204

Table 3-631. CM_CORE__CKGEN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__CKGEN Physical Address L4_CFG Interconnect
CM_AUTOIDLE_DPLL_PCIE_REF	RW	32	0x0000 0108	0x4A00 8208
CM_CLKSEL_DPLL_PCIE_REF	RW	32	0x0000 010C	0x4A00 820C
CM_DIV_M2_DPLL_PCIE_REF	RW	32	0x0000 0110	0x4A00 8210
RESERVED	R	32	0x0000 0114	0x4A00 8214
RESERVED	R	32	0x0000 0118	0x4A00 8218
CM_CLKMODE_APLL_PCIE	RW	32	0x0000 011C	0x4A00 821C
CM_IDLEST_APLL_PCIE	R	32	0x0000 0120	0x4A00 8220
CM_DIV_M2_APLL_PCIE	R	32	0x0000 0124	0x4A00 8224
CM_CLKVCOLDO_APLL_PCIE	R	32	0x0000 0128	0x4A00 8228

3.13.15.2 CM_CORE__CKGEN Register Description

Table 3-632. CM_CLKSEL_USB_60MHZ

Address Offset	0x0000 0004		
Physical Address	0x4A00 8104	**Instance**	CM_CORE__CKGEN
Description	Selects the configuration of the divider generating 60MHz clock for USB from the DPLL_USB o/p.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the configuration of the divider 0x0: Set the divider in bypass mode to support bypass clock from DPLL_USB to pass through without division. 0x1: Set the divider to divide the DPLL o/p (480MHz typical) by 8 to generate 60MHz clock.	RW	0x1

Table 3-633. Register Call Summary for Register CM_CLKSEL_USB_60MHZ

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [0]

Table 3-634. CM_CLKMODE_DPLL_PER

Address Offset	0x0000 0040		
Physical Address	0x4A00 8140	**Instance**	CM_CORE__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	RESERVED		R	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-635. Register Call Summary for Register CM_CLKMODE_DPLL_PER

Clock Management Functional Description

- DPLL_PER Power Modes: [0] [1]
- DPLL_PER Recalibration: [2]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [5]

Table 3-636. CM_IDLEST_DPLL_PER

Address Offset	0x0000 0044		
Physical Address	0x4A00 8144	**Instance**	CM_CORE__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-637. Register Call Summary for Register CM_IDLEST_DPLL_PER

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [0]

Table 3-638. CM_AUTOIDLE_DPLL_PER

Address Offset	0x0000 0048		
Physical Address	0x4A00 8148	**Instance**	CM_CORE__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													AUTO_DPLL_MODE		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-639. Register Call Summary for Register CM_AUTOIDLE_DPLL_PER

Clock Management Functional Description

- DPLL_PER Power Modes: [0]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [1]

Table 3-640. CM_CLKSEL_DPLL_PER

Address Offset	0x0000 004C		
Physical Address	0x4A00 814C	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31:24	23	22	21:19	18:8	7	6:0
RESERVED	DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_MULT	RESERVED	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-641. Register Call Summary for Register CM_CLKSEL_DPLL_PER

Clock Management Functional Description

- DPLL_PER Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-642. CM_DIV_M2_DPLL_PER

Address Offset	0x0000 0050		
Physical Address	0x4A00 8150	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31–12	11	10	9	8–5	4–0
RESERVED	CLKX2ST	RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	CLKX2ST	DPLL CLKOUTX2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_PER. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-643. Register Call Summary for Register CM_DIV_M2_DPLL_PER

Clock Management Functional Description

- DPLL_PER Synthesized Clock Parameters: [0] [1] [2]

Table 3-643. Register Call Summary for Register CM_DIV_M2_DPLL_PER (continued)

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [3]

Table 3-644. CM_DIV_H11_DPLL_PER

Address Offset	0x0000 0058		
Physical Address	0x4A00 8158	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT1 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H11 post-divider factor (1 to 63) of DPLL_PER. 0x0: Reserved 0x1: H11 = /1 0x2: H11 = /2 ... 0x3F: H11 = /63	RW	0x1

Table 3-645. Register Call Summary for Register CM_DIV_H11_DPLL_PER

Clock Management Functional Description

- DPLL_PER Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-646. CM_DIV_H12_DPLL_PER

Address Offset	0x0000 005C		
Physical Address	0x4A00 815C	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT2 status 0x0: CLK_GATED 0x1: CLK_ENABLED	R	0x0

Bits	Field Name	Description	Type	Reset
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H12 post-divider factor (1 to 63) of DPLL_PER. 0x0: Reserved 0x1: H12 = /1 0x2: H12 = /2 ... 0x3F: H12 = /63	RW	0x1

Table 3-647. Register Call Summary for Register CM_DIV_H12_DPLL_PER

Clock Management Functional Description

- DPLL_PER Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-648. CM_DIV_H13_DPLL_PER

Address Offset	0x0000 0060		
Physical Address	0x4A00 8160	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER1.		
Type	RW		

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H13 post-divider factor (1 to 63) of DPLL_PER. 0x0: Reserved 0x1: H13 = /1 0x2: H13 = /2 ... 0x3F: H13 = /63	RW	0x1

Table 3-649. Register Call Summary for Register CM_DIV_H13_DPLL_PER

Clock Management Functional Description

- DPLL_PER Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-650. CM_DIV_H14_DPLL_PER

Address Offset	0x0000 0064		
Physical Address	0x4A00 8164	**Instance**	CM_CORE__CKGEN

Table 3-650. CM_DIV_H14_DPLL_PER (continued)

Description	This register provides controls over the CLKOUT4 o/p of the HSDIVIDER1.
Type	RW

31:10	9	8:6	5:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT4 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H14 post-divider factor (1 to 63) of DPLL_PER. When a value of 63 is programmed in this register, HS divider will perform division by 2.5 that is divided by 2 at top level. 0x0: Reserved 0x2: 2 0x4: 4	RW	0x1

Table 3-651. Register Call Summary for Register CM_DIV_H14_DPLL_PER

Clock Management Functional Description

- DPLL_PER Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-652. CM_CLKMODE_DPLL_USB

Address Offset	0x0000 0080		
Physical Address	0x4A00 8180	**Instance**	CM_CORE__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31:3	2:0
RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Put the DPLL in Low Power Stop mode 0x2: Reserved2 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-653. Register Call Summary for Register CM_CLKMODE_DPLL_USB

Clock Management Functional Description

- DPLL_USB Power Modes: [0]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [1]

Table 3-654. CM_IDLEST_DPLL_USB

Address Offset	0x0000 0084		
Physical Address	0x4A00 8184	**Instance**	CM_CORE__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31–5	4	3–1	0
RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-655. Register Call Summary for Register CM_IDLEST_DPLL_USB

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [0]

Table 3-656. CM_AUTOIDLE_DPLL_USB

Address Offset	0x0000 0088		
Physical Address	0x4A00 8188	**Instance**	CM_CORE__CKGEN
Description	This register provides automatic control over the DPLL activity.		

Table 3-656. CM_AUTOIDLE_DPLL_USB (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													AUTO_DPLL_MODE		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-657. Register Call Summary for Register CM_AUTOIDLE_DPLL_USB

Clock Management Functional Description

- DPLL_USB Power Modes: [0]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [1]

Table 3-658. CM_CLKSEL_DPLL_USB

Address Offset	0x0000 008C		
Physical Address	0x4A00 818C	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPLL_SD_DIV								DPLL_BYP_CLKSEL	DCC_EN	DPLL_SELFREQDCO	RESERVED	DPLL_MULT												DPLL_DIV							

Bits	Field Name	Description	Type	Reset
31:24	DPLL_SD_DIV	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0x0: Reserved 0x1: Reserved	RW	0x4
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21	DPLL_SELFREQDCO	select DCO output according to required frequency. 0x0: DCO clock is 1500MHz SELFREQDCO input of DPLL is set to '010' 0x1: DCO clock is 1250MHz SELFREQDCO input of DPLL is set to '100'	RW	0x0
20	RESERVED		R	0x0
19:8	DPLL_MULT	DPLL multiplier factor (2 to 4095). (equal to input M of DPLL; M=2 to 4095 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7:0	DPLL_DIV	DPLL divider factor (0 to 255) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-659. Register Call Summary for Register CM_CLKSEL_DPLL_USB

Clock Management Functional Description

- DPLL_USB Synthesized Clock Parameters: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [8]

Table 3-660. CM_DIV_M2_DPLL_USB

Address Offset	0x0000 0090		
Physical Address	0x4A00 8190	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31:10	9	8:7	6:0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0

Bits	Field Name	Description	Type	Reset
8:7	RESERVED		R	0x0
6:0	DIVHS	This field programs the M2 post-divider factor (1 to 127) of DPLL_USB. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x7F: M2 = /127	RW	0x1

Table 3-661. Register Call Summary for Register CM_DIV_M2_DPLL_USB

Clock Management Functional Description

- DPLL_USB Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-662. CM_CLKDCOLDO_DPLL_USB

Address Offset	0x0000 00B4		
Physical Address	0x4A00 81B4	**Instance**	CM_CORE__CKGEN
Description	This register provides status over CLKDCOLDO output of the DPLL.		
Type	R		

31:10	9	8:0
RESERVED	ST_DPLL_CLKDCOLDO	RESERVED

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	ST_DPLL_CLKDCOLDO	DPLL CLKDCOLDO status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:0	RESERVED		R	0x0

Table 3-663. Register Call Summary for Register CM_CLKDCOLDO_DPLL_USB

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [0]

Table 3-664. CM_CLKMODE_DPLL_PCIE_REF

Address Offset	0x0000 0100		
Physical Address	0x4A00 8200	**Instance**	CM_CORE__CKGEN
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DPLL_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Put the DPLL in Low Power Stop mode 0x2: Reserved2 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-665. Register Call Summary for Register CM_CLKMODE_DPLL_PCIE_REF

Clock Management Functional Description

- DPLL_PCIE_REF Power Modes: [0]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [1]

Table 3-666. CM_IDLEST_DPLL_PCIE_REF

Address Offset	0x0000 0104		
Physical Address	0x4A00 8204	**Instance**	CM_CORE__CKGEN
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31:5	4	3:1	0
RESERVED	ST_DPLL_INIT	ST_DPLL_MODE	ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0

Bits	Field Name	Description	Type	Reset
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-667. Register Call Summary for Register CM_IDLEST_DPLL_PCIE_REF

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [0]

Table 3-668. CM_AUTOIDLE_DPLL_PCIE_REF

Address Offset	0x0000 0108		
Physical Address	0x4A00 8208	**Instance**	CM_CORE__CKGEN
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31–3	2–0
RESERVED	AUTO_DPLL_MODE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-669. Register Call Summary for Register CM_AUTOIDLE_DPLL_PCIE_REF

Clock Management Functional Description

- DPLL_PCIE_REF Power Modes: [0]

Table 3-669. Register Call Summary for Register CM_AUTOIDLE_DPLL_PCIE_REF (continued)

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [1]

Table 3-670. CM_CLKSEL_DPLL_PCIE_REF

Address Offset	0x0000 010C		
Physical Address	0x4A00 820C	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the DPLL.		
Type	RW		

31:24	23	22	21	20	19:8	7:0
DPLL_SD_DIV	DPLL_BYP_CLKSEL	DCC_EN	DPLL_SELFREQDCO	RESERVED	DPLL_MULT	DPLL_DIV

Bits	Field Name	Description	Type	Reset
31:24	DPLL_SD_DIV	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0x0: Reserved 0x1: Reserved	RW	0x4
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT	R	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21	DPLL_SELFREQDCO	select DCO output according to required frequency. 0x0: DCO clock is 1500MHz SELFREQDCO input of DPLL is set to '010' 0x1: DCO clock is 1250MHz SELFREQDCO input of DPLL is set to '100'	RW	0x0
20	RESERVED		R	0x0
19:8	DPLL_MULT	DPLL multiplier factor (2 to 4095). (equal to input M of DPLL; M=2 to 4095 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7:0	DPLL_DIV	DPLL divider factor (0 to 255) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-671. Register Call Summary for Register CM_CLKSEL_DPLL_PCIE_REF

Clock Management Functional Description

- DPLL_PCIE_REF Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-672. CM_DIV_M2_DPLL_PCIE_REF

Address Offset	0x0000 0110		
Physical Address	0x4A00 8210	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31:11	10	9	8:7	6:0
RESERVED	CLKLDOST	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKLDOST	DPLL CLKOUTLDO status 0x0: Output clock is gated 0x1: Output clock is enabled	R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:7	RESERVED		R	0x0
6:0	DIVHS	This field programs the M2 post-divider factor (1 to 127) of DPLL_PCIE_REF. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x7F: M2 = /127	RW	0x1

Table 3-673. Register Call Summary for Register CM_DIV_M2_DPLL_PCIE_REF

Clock Management Functional Description

- DPLL_PCIE_REF Synthesized Clock Parameters: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [3]

Table 3-674. CM_CLKMODE_APLL_PCIE

Address Offset	0x0000 011C		
Physical Address	0x4A00 821C	**Instance**	CM_CORE__CKGEN
Description	This register allows controlling the APLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLKDIV_BYPASS	REFSEL	RESERVED	INPSEL	MODE	MODE_SELECT

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKDIV_BYPASS	0x0: Division of CLKVCOLDO_DIV clock is controlled by OUTSEL pin driven by PCIE controlleur. If OUTSEL is '0', CLKVCOLDO_DIV is at same frequency than CLKVCOLDO output If OUTSEL is '1', CLKVCOLDO_DIV is at CLKVCOLDO divide by 2 frequency 0x1: CLKVCOLDO_DIV clock is not divided by 2 (CLKVCOLDO_DIV is at same frequency than CLKVCOLDO output)	RW	0x0
7	REFSEL	Select source of reference input clock 0x0: APLL reference input clock is from ADPLL 0x1: APLL reference input clock is from ACSPCIE	RW	0x0
6	RESERVED		R	0x0
5:3	INPSEL	Reference clock is 100MHz.	R	0x0
2	MODE	APLLPCIE Mode Status 0x0: APLLPCIE Mode Status	R	0x0
1:0	MODE_SELECT	Control APLL mode. **Note:** Please note that setting CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT bitfield to 0x0 does not disable the APLL_PCIE. In order to disable the APLL_PCIE, the user needs to disable PCIe_SSx (where x = 1 or 2) using the CM_PCIE_ PCIESSx_CLKCTRL[1:0] MODULEMODE registers. When PCIe_SS is disabled, the PRCM module automatically disables the APLL_PCIE. 0x0: RESERVED 0x1: Put the APLL in Force Lock mode 0x2: Put the APLL in Auto Idle mode 0x3: RESERVED	RW	0x0

Table 3-675. Register Call Summary for Register CM_CLKMODE_APLL_PCIE

Clock Management Functional Description
- APLL_PCIE Power Modes: [0] [1]
- Clock Domain Module Attributes: [2]

PRCM Register Manual
- CM_CORE__CKGEN Register Summary: [3]
- CM_CORE__CKGEN Register Description: [4]
- CM_CORE__L3INIT Register Description: [5] [6]

Table 3-676. CM_IDLEST_APLL_PCIE

Address Offset	0x0000 0120		
Physical Address	0x4A00 8220	**Instance**	CM_CORE__CKGEN
Description	This register allows monitoring APLL activity. This register is read only and automatically updated. [warm reset insensitive]		

Table 3-676. CM_IDLEST_APLL_PCIE (continued)

Type	R

31–1	0
RESERVED	ST_APLL_CLK

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ST_APLL_CLK	APLL lock status 0x0: APLL is either in bypass mode or in stop mode. 0x1: APLL is LOCKED	R	0x0

Table 3-677. Register Call Summary for Register CM_IDLEST_APLL_PCIE

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [0]

Table 3-678. CM_DIV_M2_APLL_PCIE

Address Offset	0x0000 0124		
Physical Address	0x4A00 8224	**Instance**	CM_CORE__CKGEN
Description	This register provides controls over the M2 divider of the DPLL.		
Type	R		

31–10	9	8–7	6–0
RESERVED	CLKST	RESERVED	DIVHS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:7	RESERVED		R	0x0
6:0	DIVHS	DPLL M2 post-divider factor (1 to 127). (RESERVED) 0x0: Reserved	R	0x1

Table 3-679. Register Call Summary for Register CM_DIV_M2_APLL_PCIE

Clock Management Functional Description

- APLL_PCIE Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

Table 3-680. CM_CLKVCOLDO_APLL_PCIE

Address Offset	0x0000 0128		
Physical Address	0x4A00 8228	**Instance**	CM_CORE__CKGEN
Description	This register provides status over CLKVCOLDO and CLKVCOLDO_DIV outputs of the APLL.		

Table 3-680. CM_CLKVCOLDO_APLL_PCIE (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					CLK_DIVST	CLKST	RESERVED								

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLK_DIVST	APLL CLKVCOLDO_DIV status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
9	CLKST	APLL CLKVCOLDO status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:0	RESERVED		R	0x0

Table 3-681. Register Call Summary for Register CM_CLKVCOLDO_APLL_PCIE

Clock Management Functional Description

- APLL_PCIE Synthesized Clock Parameters: [0] [1]

PRCM Register Manual

- CM_CORE__CKGEN Register Summary: [2]

3.13.16 CM_CORE__COREAON Registers

3.13.16.1 CM_CORE__COREAON Register Summary

Table 3-682. CM_CORE__COREAON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__COREAON Physical Address L4_CFG Interconnect
CM_COREAON_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 8600
RESERVED	R	32	0x0000 0028	0x4A00 8628
RESERVED	R	32	0x0000 0038	0x4A00 8638
CM_COREAON_USB_PHY1_CORE_CLKCTRL	RW	32	0x0000 0040	0x4A00 8640
RESERVED	R	32	0x0000 0050	0x4A00 8650
RESERVED	R	32	0x0000 0058	0x4A00 8658
RESERVED	R	32	0x0000 0068	0x4A00 8668
RESERVED	R	32	0x0000 0078	0x4A00 8678
CM_COREAON_USB_PHY2_CORE_CLKCTRL	RW	32	0x0000 0088	0x4A00 8688
CM_COREAON_USB_PHY3_CORE_CLKCTRL	RW	32	0x0000 0098	0x4A00 8698
CM_COREAON_CLKOUTMUX1_CLKCTRL	RW	32	0x0000 00A0	0x4A00 86A0
CM_COREAON_CLKOUTMUX2_CLKCTRL	RW	32	0x0000 00B0	0x4A00 86B0
CM_COREAON_L3INIT_60M_GFCLK_CLKCTRL	RW	32	0x0000 00C0	0x4A00 86C0
CM_COREAON_ABE_GICLK_CLKCTRL	RW	32	0x0000 00D0	0x4A00 86D0

3.13.16.2 CM_CORE__COREAON Register Description

Table 3-683. CM_COREAON_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 8600	**Instance**	CM_CORE__COREAON

Table 3-683. CM_COREAON_CLKSTCTRL (continued)

Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">RESERVED</td><td>CLKACTIVITY_ABE_GICLK</td><td colspan="3">RESERVED</td><td>CLKACTIVITY_COREAON_32K_GFCLK</td><td colspan="10">RESERVED</td><td colspan="2">CLKTRCTRL</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CLKACTIVITY_ABE_GICLK	This field indicates the state of the ABE_GICLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15:13	RESERVED		R	0x0
12	CLKACTIVITY_COREAON_32K_GFCLK	This field indicates the state of the COREAON_32K_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the COREAON clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-684. Register Call Summary for Register CM_COREAON_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [3]

Table 3-685. CM_COREAON_USB_PHY1_CORE_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 8640	**Instance**	CM_CORE__COREAON
Description	This register manages the USB PHY 32KHz clock.		
Type	RW		

31–9	8	7–0
RESERVED	OPTFCLKEN_CLK32K	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-686. Register Call Summary for Register CM_COREAON_USB_PHY1_CORE_CLKCTRL

Clock Management Functional Description

- Clock Domain Modes: [0]
- Clock Domain Module Attributes: [1]

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [2]

Table 3-687. CM_COREAON_USB_PHY2_CORE_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 8688	**Instance**	CM_CORE__COREAON
Description	This register manages the USB PHY 32KHz clock.		
Type	RW		

31–9	8	7–0
RESERVED	OPTFCLKEN_CLK32K	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-688. Register Call Summary for Register CM_COREAON_USB_PHY2_CORE_CLKCTRL

Clock Management Functional Description

- Clock Domain Modes: [0]
- Clock Domain Module Attributes: [1]

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [2]

Table 3-689. CM_COREAON_USB_PHY3_CORE_CLKCTRL

Address Offset	0x0000 0098		
Physical Address	0x4A00 8698	**Instance**	CM_CORE__COREAON
Description	This register manages the USB PHY 32KHz clock.		
Type	RW		

31:9	8	7:0
RESERVED	OPTFCLKEN_CLK32K	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-690. Register Call Summary for Register CM_COREAON_USB_PHY3_CORE_CLKCTRL

Clock Management Functional Description

- Clock Domain Modes: [0]
- Clock Domain Module Attributes: [1]

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [2]

Table 3-691. CM_COREAON_CLKOUTMUX1_CLKCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4A00 86A0	**Instance**	CM_CORE__COREAON
Description	Used for controlling the CLKOUTMUX 1 gate.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	OPTFCLKEN_CLKOUTMUX1_CLK	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLKOUTMUX1_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-692. Register Call Summary for Register CM_COREAON_CLKOUTMUX1_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [1]

Table 3-693. CM_COREAON_CLKOUTMUX2_CLKCTRL

Address Offset	0x0000 00B0		
Physical Address	0x4A00 86B0	**Instance**	CM_CORE__COREAON
Description	Used for controlling the CLKOUTMUX 2 gate.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_CLKOUTMUX2_CLK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLKOUTMUX2_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-694. Register Call Summary for Register CM_COREAON_CLKOUTMUX2_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [1]

Table 3-695. CM_COREAON_L3INIT_60M_GFCLK_CLKCTRL

Address Offset	0x0000 00C0		
Physical Address	0x4A00 86C0	**Instance**	CM_CORE__COREAON
Description	Used for controlling the L3INIT_60M_GFCLK gate.		
Type	RW		

31:9	8	7:0
RESERVED	OPTFCLKEN_L3INIT_60M_GFCLK	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_L3INIT_60M_GFCLK	Optional functional clock control; used to control the clock of USB2PHY2. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-696. Register Call Summary for Register CM_COREAON_L3INIT_60M_GFCLK_CLKCTRL

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [0]

Table 3-697. CM_COREAON_ABE_GICLK_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4A00 86D0	**Instance**	CM_CORE__COREAON
Description	Used for controlling ABE_GICLK gate.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	OPTFCLKEN_ABE_GICLK	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_ABE_GICLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-698. Register Call Summary for Register CM_COREAON_ABE_GICLK_CLKCTRL

PRCM Register Manual

- CM_CORE__COREAON Register Summary: [0]

3.13.17 CM_CORE__CORE Registers

3.13.17.1 CM_CORE__CORE Register Summary

Table 3-699. CM_CORE__CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__CORE Physical Address L4_CFG Interconnect
CM_L3MAIN1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 8700
CM_L3MAIN1_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 8708
CM_L3MAIN1_L3_MAIN_1_CLKCTRL	R	32	0x0000 0020	0x4A00 8720
CM_L3MAIN1_GPMC_CLKCTRL	RW	32	0x0000 0028	0x4A00 8728
CM_L3MAIN1_MMU_EDMA_CLKCTRL	R	32	0x0000 0030	0x4A00 8730
CM_L3MAIN1_MMU_PCIESS_CLKCTRL	R	32	0x0000 0048	0x4A00 8748
CM_L3MAIN1_OCMC_RAM1_CLKCTRL	R	32	0x0000 0050	0x4A00 8750
CM_L3MAIN1_OCMC_RAM2_CLKCTRL	R	32	0x0000 0058	0x4A00 8758
CM_L3MAIN1_OCMC_RAM3_CLKCTRL	R	32	0x0000 0060	0x4A00 8760
CM_L3MAIN1_TPCC_CLKCTRL	R	32	0x0000 0070	0x4A00 8770
CM_L3MAIN1_TPTC1_CLKCTRL	RW	32	0x0000 0078	0x4A00 8778
CM_L3MAIN1_TPTC2_CLKCTRL	RW	32	0x0000 0080	0x4A00 8780
CM_L3MAIN1_VCP1_CLKCTRL	R	32	0x0000 0088	0x4A00 8788
CM_L3MAIN1_VCP2_CLKCTRL	R	32	0x0000 0090	0x4A00 8790
RESERVED	R	32	0x0000 0098	0x4A00 8798
RESERVED	R	32	0x0000 00A0	0x4A00 87A0
RESERVED	R	32	0x0000 00A8	0x4A00 87A8
RESERVED	R	32	0x0000 00B0	0x4A00 87B0
RESERVED	R	32	0x0000 00B8	0x4A00 87B8
RESERVED	R	32	0x0000 00C0	0x4A00 87C0
RESERVED	R	32	0x0000 00C8	0x4A00 87C8
RESERVED	R	32	0x0000 00D0	0x4A00 87D0
RESERVED	R	32	0x0000 00D8	0x4A00 87D8
RESERVED	R	32	0x0000 00F0	0x4A00 87F0
RESERVED	R	32	0x0000 00F8	0x4A00 87F8
CM_IPU2_CLKSTCTRL	RW	32	0x0000 0200	0x4A00 8900

Table 3-699. CM_CORE__CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__CORE Physical Address L4_CFG Interconnect
CM_IPU2_STATICDEP	RW	32	0x0000 0204	0x4A00 8904
CM_IPU2_DYNAMICDEP	RW	32	0x0000 0208	0x4A00 8908
CM_IPU2_IPU2_CLKCTRL	RW	32	0x0000 0220	0x4A00 8920
CM_DMA_CLKSTCTRL	RW	32	0x0000 0300	0x4A00 8A00
CM_DMA_STATICDEP	RW	32	0x0000 0304	0x4A00 8A04
CM_DMA_DYNAMICDEP	R	32	0x0000 0308	0x4A00 8A08
CM_DMA_DMA_SYSTEM_CLKCTRL	R	32	0x0000 0320	0x4A00 8A20
CM_EMIF_CLKSTCTRL	RW	32	0x0000 0400	0x4A00 8B00
CM_EMIF_DMM_CLKCTRL	R	32	0x0000 0420	0x4A00 8B20
CM_EMIF_EMIF_OCP_FW_CLKCTRL	R	32	0x0000 0428	0x4A00 8B28
CM_EMIF_EMIF1_CLKCTRL	RW	32	0x0000 0430	0x4A00 8B30
CM_EMIF_EMIF2_CLKCTRL	RW	32	0x0000 0438	0x4A00 8B38
CM_EMIF_EMIF_DLL_CLKCTRL	RW	32	0x0000 0440	0x4A00 8B40
CM_ATL_ATL_CLKCTRL	RW	32	0x0000 0500	0x4A00 8C00
CM_ATL_CLKSTCTRL	RW	32	0x0000 0520	0x4A00 8C20
CM_L4CFG_CLKSTCTRL	RW	32	0x0000 0600	0x4A00 8D00
CM_L4CFG_DYNAMICDEP	RW	32	0x0000 0608	0x4A00 8D08
CM_L4CFG_L4_CFG_CLKCTRL	R	32	0x0000 0620	0x4A00 8D20
CM_L4CFG_SPINLOCK_CLKCTRL	R	32	0x0000 0628	0x4A00 8D28
CM_L4CFG_MAILBOX1_CLKCTRL	R	32	0x0000 0630	0x4A00 8D30
CM_L4CFG_SAR_ROM_CLKCTRL	R	32	0x0000 0638	0x4A00 8D38
CM_L4CFG_OCP2SCP2_CLKCTRL	R	32	0x0000 0640	0x4A00 8D40
CM_L4CFG_MAILBOX2_CLKCTRL	R	32	0x0000 0648	0x4A00 8D48
CM_L4CFG_MAILBOX3_CLKCTRL	R	32	0x0000 0650	0x4A00 8D50
CM_L4CFG_MAILBOX4_CLKCTRL	R	32	0x0000 0658	0x4A00 8D58
CM_L4CFG_MAILBOX5_CLKCTRL	R	32	0x0000 0660	0x4A00 8D60
CM_L4CFG_MAILBOX6_CLKCTRL	R	32	0x0000 0668	0x4A00 8D68
CM_L4CFG_MAILBOX7_CLKCTRL	R	32	0x0000 0670	0x4A00 8D70
CM_L4CFG_MAILBOX8_CLKCTRL	R	32	0x0000 0678	0x4A00 8D78
CM_L4CFG_MAILBOX9_CLKCTRL	R	32	0x0000 0680	0x4A00 8D80
CM_L4CFG_MAILBOX10_CLKCTRL	R	32	0x0000 0688	0x4A00 8D88
CM_L4CFG_MAILBOX11_CLKCTRL	R	32	0x0000 0690	0x4A00 8D90
CM_L4CFG_MAILBOX12_CLKCTRL	R	32	0x0000 0698	0x4A00 8D98
CM_L4CFG_MAILBOX13_CLKCTRL	R	32	0x0000 06A0	0x4A00 8DA0
RESERVED	R	32	0x0000 06A8	0x4A00 8DA8
RESERVED	R	32	0x0000 06B0	0x4A00 8DB0
RESERVED	R	32	0x0000 06B8	0x4A00 8DB8
RESERVED	R	32	0x0000 06C0	0x4A00 8DC0
CM_L3INSTR_CLKSTCTRL	R	32	0x0000 0700	0x4A00 8E00
CM_L3INSTR_L3_MAIN_2_CLKCTRL	RW	32	0x0000 0720	0x4A00 8E20
CM_L3INSTR_L3_INSTR_CLKCTRL	RW	32	0x0000 0728	0x4A00 8E28
CM_L3INSTR_OCP_WP_NOC_CLKCTRL	RW	32	0x0000 0740	0x4A00 8E40
CM_L3INSTR_DLL_AGING_CLKCTRL	R	32	0x0000 0748	0x4A00 8E48
CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL	RW	32	0x0000 0750	0x4A00 8E50

3.13.17.2 CM_CORE__CORE Register Description

Table 3-700. CM_L3MAIN1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 8700	**Instance**	CM_CORE__CORE

Table 3-700. CM_L3MAIN1_CLKSTCTRL (continued)

Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.
Type	RW

31–10	9	8	7–2	1–0
RESERVED	CLKACTIVITY_L3MAIN1_L4_GICLK	CLKACTIVITY_L3MAIN1_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_L3MAIN1_L4_GICLK	This field indicates the state of the L3MAIN1_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L3MAIN1_L3_GICLK	This field indicates the state of the L3MAIN1_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L3MAIN1 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-701. Register Call Summary for Register CM_L3MAIN1_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0] [1] [2]

PRCM Register Manual
- CM_CORE__CORE Register Summary: [3]
- CM_CORE__RESTORE Register Description: [4] [5]

Table 3-702. CM_L3MAIN1_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 8708	**Instance**	CM_CORE__CORE

Table 3-702. CM_L3MAIN1_DYNAMICDEP (continued)

Description	This register controls the dynamic domain depedencies from L3MAIN1 domain towards 'target' domains. It is relevant only for domain having OCP master port(s).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	EVE2_DYNDEP	EVE1_DYNDEP	WINDOWSIZE				L4PER3_DYNDEP	L4PER2_DYNDEP	PCIE_DYNDEP	DSP2_DYNDEP	RESERVED	IPU1_DYNDEP	RESERVED		WKUPAON_DYNDEP	L4SEC_DYNDEP	L4PER_DYNDEP	L4CFG_DYNDEP	RESERVED	GPU_DYNDEP	RESERVED	DSS_DYNDEP	RESERVED			EMIF_DYNDEP	IPU_DYNDEP	IVA_DYNDEP	DSP1_DYNDEP	IPU2_DYNDEP

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	RESERVED		R	0x0
29	EVE2_DYNDEP	Dynamic dependency towards EVE2 clock domain 0x1: Dependency is enabled	R	0x1
28	EVE1_DYNDEP	Dynamic dependency towards EVE1 clock domain 0x1: Dependency is enabled	R	0x1
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23	L4PER3_DYNDEP	Dynamic dependency towards L4PER3 clock domain 0x1: Dependency is enabled	R	0x1
22	L4PER2_DYNDEP	Dynamic dependency towards L4PER2 clock domain 0x1: Dependency is enabled	R	0x1
21	PCIE_DYNDEP	Dynamic dependency towards PCIE clock domain 0x1: Dependency is enabled	R	0x1
20	DSP2_DYNDEP	Dynamic dependency towards DSP2 clock domain 0x1: Dependency is enabled	R	0x1
19	RESERVED		R	0x0
18	IPU1_DYNDEP	Dynamic dependency towards IPU1 clock domain 0x1: Dependency is enabled	R	0x1
17:16	RESERVED		R	0x0
15	WKUPAON_DYNDEP	Dynamic dependency towards WKUPAON clock domain 0x1: Dependency is enabled	R	0x1
14	L4SEC_DYNDEP	Dynamic dependency towards L4SEC clock domain 0x1: Dependency is enabled	R	0x1
13	L4PER_DYNDEP	Dynamic dependency towards L4PER1 clock domain 0x1: Dependency is enabled	R	0x1
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain 0x1: Dependency is enabled	R	0x1
11	RESERVED		R	0x0
10	GPU_DYNDEP	Dynamic dependency towards GPU clock domain 0x1: Dependency is enabled	R	0x1
9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain 0x1: Dependency is enabled	R	0x1
7:5	RESERVED		R	0x0
4	EMIF_DYNDEP	Dynamic dependency towards EMIF clock domain 0x1: Dependency is enabled	R	0x1
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2	IVA_DYNDEP	Dynamic dependency towards IVA clock domain 0x1: Dependency is enabled	R	0x1
1	DSP1_DYNDEP	Dynamic dependency towards DSP1 clock domain 0x1: Dependency is enabled	R	0x1
0	IPU2_DYNDEP	Dynamic dependency towards IPU2 clock domain 0x1: Dependency is enabled	R	0x1

Table 3-703. Register Call Summary for Register CM_L3MAIN1_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [20]
- CM_CORE__RESTORE Register Description: [21] [22]

Table 3-704. CM_L3MAIN1_L3_MAIN_1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 8720	**Instance**	CM_CORE__CORE
Description	This register manages the L3_MAIN_1 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-705. Register Call Summary for Register CM_L3MAIN1_L3_MAIN_1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-706. CM_L3MAIN1_GPMC_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 8728	**Instance**	CM_CORE__CORE
Description	This register manages the GPMC clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by SW. OCP access to module are stalled. Can be used to change timing parameter of GPMC module. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-707. Register Call Summary for Register CM_L3MAIN1_GPMC_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-708. CM_L3MAIN1_MMU_EDMA_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 8730	**Instance**	CM_CORE__CORE
Description	This register manages the MMU_L4_EDMA clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-709. Register Call Summary for Register CM_L3MAIN1_MMU_EDMA_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-710. CM_L3MAIN1_MMU_PCIESS_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4A00 8748	**Instance**	CM_CORE__CORE
Description	This register manages the MMU_L4_PCIESS clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-711. Register Call Summary for Register CM_L3MAIN1_MMU_PCIESS_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-712. CM_L3MAIN1_OCMC_RAM1_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 8750	**Instance**	CM_CORE__CORE
Description	This register manages the OCMC_RAM1 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-713. Register Call Summary for Register CM_L3MAIN1_OCMC_RAM1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-714. CM_L3MAIN1_OCMC_RAM2_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 8758	**Instance**	CM_CORE__CORE

Table 3-714. CM_L3MAIN1_OCMC_RAM2_CLKCTRL (continued)

Description	This register manages the OCMC_RAM2 clocks.
Type	R

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-715. Register Call Summary for Register CM_L3MAIN1_OCMC_RAM2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-716. CM_L3MAIN1_OCMC_RAM3_CLKCTRL

Address Offset	0x0000 0060		
Physical Address	0x4A00 8760	**Instance**	CM_CORE__CORE
Description	This register manages the OCMC_RAM3 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-717. Register Call Summary for Register CM_L3MAIN1_OCMC_RAM3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-718. CM_L3MAIN1_TPCC_CLKCTRL

Address Offset	0x0000 0070		
Physical Address	0x4A00 8770	**Instance**	CM_CORE__CORE
Description	This register manages the TPCC clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-719. Register Call Summary for Register CM_L3MAIN1_TPCC_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-720. CM_L3MAIN1_TPTC1_CLKCTRL

Address Offset	0x0000 0078

Table 3-720. CM_L3MAIN1_TPTC1_CLKCTRL (continued)

Physical Address	0x4A00 8778	**Instance**	CM_CORE__CORE
Description	This register manages the TPTC1 clocks.		
Type	RW		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-721. Register Call Summary for Register CM_L3MAIN1_TPTC1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [3]

Table 3-722. CM_L3MAIN1_TPTC2_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4A00 8780	**Instance**	CM_CORE__CORE
Description	This register manages the TPTC2 clocks.		
Type	RW		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-723. Register Call Summary for Register CM_L3MAIN1_TPTC2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [3]

Table 3-724. CM_L3MAIN1_VCP1_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 8788	**Instance**	CM_CORE__CORE
Description	This register manages the VCP1 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-725. Register Call Summary for Register CM_L3MAIN1_VCP1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-726. CM_L3MAIN1_VCP2_CLKCTRL

Address Offset	0x0000 0090		
Physical Address	0x4A00 8790	**Instance**	CM_CORE__CORE
Description	This register manages the VCP2 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-727. Register Call Summary for Register CM_L3MAIN1_VCP2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-728. CM_IPU2_CLKSTCTRL

Address Offset	0x0000 0200

Table 3-728. CM_IPU2_CLKSTCTRL (continued)

Physical Address	0x4A00 8900	**Instance**	CM_CORE__CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:9	8	7:2	1:0
RESERVED	CLKACTIVITY_IPU2_GFCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_IPU2_GFCLK	This field indicates the state of the IPU2_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the IPU2 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-729. Register Call Summary for Register CM_IPU2_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-730. CM_IPU2_STATICDEP

Address Offset	0x0000 0204		
Physical Address	0x4A00 8904	**Instance**	CM_CORE__CORE
Description	This register controls the static domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ATL_STATDEP	PCIE_STATDEP	VPE_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	RESERVED	RESERVED	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP	WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ATL_STATDEP	Static dependency towards ATL clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	VPE_STATDEP	Static dependency towards VPE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0

Bits	Field Name	Description	Type	Reset
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11	SDMA_STATDEP	Static dependency towards DMA clock domain 0x0: Dependency is disabled	R	0x0
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	RESERVED		R	0x0

Table 3-731. Register Call Summary for Register CM_IPU2_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25]

Table 3-731. Register Call Summary for Register CM_IPU2_STATICDEP (continued)

PRCM Register Manual

- CM_CORE__CORE Register Summary: [28]

Table 3-732. CM_IPU2_DYNAMICDEP

Address Offset	0x0000 0208		
Physical Address	0x4A00 8908	**Instance**	CM_CORE__CORE
Description	This register controls the dynamic domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31:28	27:24	23:10	9	8:6	5	4:0
RESERVED	WINDOWSIZE	RESERVED	CAM_DYNDEP	RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:10	RESERVED		R	0x0
9	CAM_DYNDEP	Dynamic dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0
8:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-733. Register Call Summary for Register CM_IPU2_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-734. CM_IPU2_IPU2_CLKCTRL

Address Offset	0x0000 0220		
Physical Address	0x4A00 8920	**Instance**	CM_CORE__CORE
Description	This register manages the IPU2 clocks.		
Type	RW		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-735. Register Call Summary for Register CM_IPU2_IPU2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [3]

Table 3-736. CM_DMA_CLKSTCTRL

Address Offset	0x0000 0300		
Physical Address	0x4A00 8A00	**Instance**	CM_CORE__CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:9	8	7:2	1:0
RESERVED	CLKACTIVITY_DMA_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	CLKACTIVITY_DMA_L3_GICLK	This field indicates the state of the DMA_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DMA clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-737. Register Call Summary for Register CM_DMA_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-738. CM_DMA_STATICDEP

Address Offset	0x0000 0304		
Physical Address	0x4A00 8A04	**Instance**	CM_CORE__CORE
Description	This register controls the static domain depedencies from DMA domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PCIE_STATDEP	RESERVED	L4PER3_STATDEP	L4PER2_STATDEP	RESERVED	IPU_STATDEP	IPU1_STATDEP	RESERVED							WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED		CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	RESERVED		R	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
25	RESERVED		R	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22:16	RESERVED		R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	RESERVED		R	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-739. Register Call Summary for Register CM_DMA_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [16]
- CM_CORE__RESTORE Register Description: [17] [18]

Table 3-740. CM_DMA_DYNAMICDEP

Address Offset	0x0000 0308		
Physical Address	0x4A00 8A08	**Instance**	CM_CORE__CORE
Description	This register controls the dynamic domain depedencies from SDMA domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31:6	5	4:0
RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-741. Register Call Summary for Register CM_DMA_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [1]

Table 3-742. CM_DMA_DMA_SYSTEM_CLKCTRL

Address Offset	0x0000 0320		
Physical Address	0x4A00 8A20	**Instance**	CM_CORE__CORE
Description	This register manages the DMA_SYSTEM clocks.		
Type	R		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-743. Register Call Summary for Register CM_DMA_DMA_SYSTEM_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [3]

Table 3-744. CM_EMIF_CLKSTCTRL

Address Offset	0x0000 0400		
Physical Address	0x4A00 8B00	**Instance**	CM_CORE__CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:11	10	9	8	7:2	1:0
RESERVED	CLKACTIVITY_EMIF_PHY_GCLK	CLKACTIVITY_EMIF_DLL_GCLK	CLKACTIVITY_EMIF_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKACTIVITY_EMIF_PHY_GCLK	This field indicates the state of the EMIF_PHY_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
9	CLKACTIVITY_EMIF_DLL_GCLK	This field indicates the state of the DLL_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_EMIF_L3_GICLK	This field indicates the state of the EMIF_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the EMIF clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-745. Register Call Summary for Register CM_EMIF_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [4]

Table 3-746. CM_EMIF_DMM_CLKCTRL

Address Offset	0x0000 0420		
Physical Address	0x4A00 8B20	**Instance**	CM_CORE__CORE
Description	This register manages the DMM clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-747. Register Call Summary for Register CM_EMIF_DMM_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-748. CM_EMIF_EMIF_OCP_FW_CLKCTRL

Address Offset	0x0000 0428		
Physical Address	0x4A00 8B28	**Instance**	CM_CORE__CORE
Description	This register manages the EMIF_OCP_FW clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-749. Register Call Summary for Register CM_EMIF_EMIF_OCP_FW_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-750. CM_EMIF_EMIF1_CLKCTRL

Address Offset	0x0000 0430		
Physical Address	0x4A00 8B30	**Instance**	CM_CORE__CORE

Table 3-750. CM_EMIF_EMIF1_CLKCTRL (continued)

Description	This register manages the EMIF1 clocks.
Type	RW

31:25	24	23:18	17:16	15:2	1:0
RESERVED	RESERVED	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		RO	0x0
24	RESERVED		RO	0x0
23:18	RESERVED		RO	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	RO	0x3
15:2	RESERVED		RO	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by SW. OCP access to module are stalled. Can be used to change timing parameter of EMIF_1 module. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-751. Register Call Summary for Register CM_EMIF_EMIF1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-752. CM_EMIF_EMIF2_CLKCTRL

Address Offset	0x0000 0438		
Physical Address	0x4A00 8B38	**Instance**	CM_CORE__CORE
Description	This register manages the EMIF2 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by SW. OCP access to module are stalled. Can be used to change timing parameter of EMIF_2 module. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-753. Register Call Summary for Register CM_EMIF_EMIF2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-754. CM_EMIF_EMIF_DLL_CLKCTRL

Address Offset	0x0000 0440		
Physical Address	0x4A00 8B40	**Instance**	CM_CORE__CORE
Description	This register manages the DLL clock.		
Type	RW		

31:9	8	7:0
RESERVED	OPTFCLKEN_DLL_CLK	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_DLL_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled. DLL_CLK can be gated when EMIF domain performs sleep transition 0x1: Optional functional clock is enabled. DLL_CLK is garantied to not be gated if already running.	RW	0x0
7:0	RESERVED		R	0x0

Table 3-755. Register Call Summary for Register CM_EMIF_EMIF_DLL_CLKCTRL

Clock Management Functional Description

- Clock Domain Modes: [0]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [1]

Table 3-756. CM_ATL_ATL_CLKCTRL

Address Offset	0x0000 0500		
Physical Address	0x4A00 8C00	**Instance**	CM_CORE__CORE
Description	This register manages the ATL clocks.		
Type	RW		

31:28	27:26	25:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL_SOURCE2	CLKSEL_SOURCE1	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_SOURCE2	Selects source for ATL clock 0x0: Selects L3_ICLK 0x1: Selects PER_ABE_X1_CLK 0x2: Selects DPLL_CLK from SOURCE1 0x3: RESERVED	RW	0x0
25:24	CLKSEL_SOURCE1	Selects source for ATL clock 0x0: Selects FUNC_32K_CLK 0x1: Selects VIDEO1_CLK 0x2: Selects VIDEO2_CLK 0x3: Selects HDMI_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-757. Register Call Summary for Register CM_ATL_ATL_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Module Attributes: [2] [3] [4] [5]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [6]

Table 3-758. CM_ATL_CLKSTCTRL

Address Offset	0x0000 0520		
Physical Address	0x4A00 8C20	**Instance**	CM_CORE__CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:10	9	8	7:2	1:0
RESERVED	CLKACTIVITY_ATL_GFCLK	CLKACTIVITY_ATL_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_ATL_GFCLK	This field indicates the state of the ATL_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_ATL_L3_GICLK	This field indicates the state of the ATL_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the C2C clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-759. Register Call Summary for Register CM_ATL_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [3]

Table 3-760. CM_L4CFG_CLKSTCTRL

Address Offset	0x0000 0600		
Physical Address	0x4A00 8D00	**Instance**	CM_CORE__CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31–10	9	8	7–2	1–0
RESERVED	CLKACTIVITY_L4CFG_L3_GICLK	CLKACTIVITY_L4CFG_L4_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_L4CFG_L3_GICLK	This field indicates the state of the L4CFG_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L4CFG_L4_GICLK	This field indicates the state of the L4CFG_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L4CFG clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-761. Register Call Summary for Register CM_L4CFG_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2]

Table 3-761. Register Call Summary for Register CM_L4CFG_CLKSTCTRL (continued)

PRCM Register Manual

- CM_CORE__CORE Register Summary: [3]
- CM_CORE__RESTORE Register Description: [4]

Table 3-762. CM_L4CFG_DYNAMICDEP

Address Offset	0x0000 0608		
Physical Address	0x4A00 8D08	**Instance**	CM_CORE__CORE
Description	This register controls the dynamic domain depedencies from L4CFG domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED				MPU_DYNDEP	RESERVED	CUSTEFUSE_DYNDEP	COREAON_DYNDEP	RESERVED				SDMA_DYNDEP	RESERVED			L3INIT_DYNDEP	RESERVED	L3MAIN1_DYNDEP	EMIF_DYNDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:20	RESERVED		R	0x0
19	MPU_DYNDEP	Dynamic dependency towards MPU clock domain 0x1: Dependency is enabled	R	0x1
18	RESERVED		R	0x0
17	CUSTEFUSE_DYNDEP	Dynamic dependency towards CUSTEFUSE clock domain 0x1: Dependency is enabled	R	0x1
16	COREAON_DYNDEP	Dynamic dependency towards COREAON clock domain 0x1: Dependency is enabled	R	0x1
15:12	RESERVED		R	0x0
11	SDMA_DYNDEP	Dynamic dependency towards DMA clock domain 0x1: Dependency is enabled	R	0x1
10:8	RESERVED		R	0x0
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_DYNDEP	Dynamic dependency towards EMIF clock domain 0x1: Dependency is enabled	R	0x1
3:0	RESERVED		R	0x0

Table 3-763. Register Call Summary for Register CM_L4CFG_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [7]
- CM_CORE__RESTORE Register Description: [8] [9]

Table 3-764. CM_L4CFG_L4_CFG_CLKCTRL

Address Offset	0x0000 0620		
Physical Address	0x4A00 8D20	**Instance**	CM_CORE__CORE
Description	This register manages the L4_CFG clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-765. Register Call Summary for Register CM_L4CFG_L4_CFG_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-766. CM_L4CFG_SPINLOCK_CLKCTRL

Address Offset	0x0000 0628		
Physical Address	0x4A00 8D28	**Instance**	CM_CORE__CORE
Description	This register manages the SPINLOCK clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-767. Register Call Summary for Register CM_L4CFG_SPINLOCK_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-768. CM_L4CFG_MAILBOX1_CLKCTRL

Address Offset	0x0000 0630		
Physical Address	0x4A00 8D30	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX1 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-769. Register Call Summary for Register CM_L4CFG_MAILBOX1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-770. CM_L4CFG_SAR_ROM_CLKCTRL

Address Offset	0x0000 0638		
Physical Address	0x4A00 8D38	**Instance**	CM_CORE__CORE
Description	This register manages the SAR_ROM clocks. **NOTE: This register is NOT supported on this device.**		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-771. Register Call Summary for Register CM_L4CFG_SAR_ROM_CLKCTRL

PRCM Register Manual

- CM_CORE__CORE Register Summary: [0]

Table 3-772. CM_L4CFG_OCP2SCP2_CLKCTRL

Address Offset	0x0000 0640		
Physical Address	0x4A00 8D40	**Instance**	CM_CORE__CORE
Description	This register manages the OCP2SCP2 clocks and the optional clock of USB PHY.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-773. Register Call Summary for Register CM_L4CFG_OCP2SCP2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-774. CM_L4CFG_MAILBOX2_CLKCTRL

Address Offset	0x0000 0648		
Physical Address	0x4A00 8D48	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX2 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-775. Register Call Summary for Register CM_L4CFG_MAILBOX2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-776. CM_L4CFG_MAILBOX3_CLKCTRL

Address Offset	0x0000 0650		
Physical Address	0x4A00 8D50	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX3 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-777. Register Call Summary for Register CM_L4CFG_MAILBOX3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-778. CM_L4CFG_MAILBOX4_CLKCTRL

Address Offset	0x0000 0658		
Physical Address	0x4A00 8D58	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX4 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-779. Register Call Summary for Register CM_L4CFG_MAILBOX4_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-780. CM_L4CFG_MAILBOX5_CLKCTRL

Address Offset	0x0000 0660		
Physical Address	0x4A00 8D60	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX5 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-781. Register Call Summary for Register CM_L4CFG_MAILBOX5_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-782. CM_L4CFG_MAILBOX6_CLKCTRL

Address Offset	0x0000 0668		
Physical Address	0x4A00 8D68	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX6 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-783. Register Call Summary for Register CM_L4CFG_MAILBOX6_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-784. CM_L4CFG_MAILBOX7_CLKCTRL

Address Offset	0x0000 0670		
Physical Address	0x4A00 8D70	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX7 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-785. Register Call Summary for Register CM_L4CFG_MAILBOX7_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-786. CM_L4CFG_MAILBOX8_CLKCTRL

Address Offset	0x0000 0678		
Physical Address	0x4A00 8D78	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX8 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-787. Register Call Summary for Register CM_L4CFG_MAILBOX8_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-788. CM_L4CFG_MAILBOX9_CLKCTRL

Address Offset	0x0000 0680		
Physical Address	0x4A00 8D80	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX9 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-789. Register Call Summary for Register CM_L4CFG_MAILBOX9_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-790. CM_L4CFG_MAILBOX10_CLKCTRL

Address Offset	0x0000 0688		
Physical Address	0x4A00 8D88	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX10 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-791. Register Call Summary for Register CM_L4CFG_MAILBOX10_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-792. CM_L4CFG_MAILBOX11_CLKCTRL

Address Offset	0x0000 0690		
Physical Address	0x4A00 8D90	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX11 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-793. Register Call Summary for Register CM_L4CFG_MAILBOX11_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual
- CM_CORE__CORE Register Summary: [2]

Table 3-794. CM_L4CFG_MAILBOX12_CLKCTRL

Address Offset	0x0000 0698		
Physical Address	0x4A00 8D98	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX12 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-795. Register Call Summary for Register CM_L4CFG_MAILBOX12_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual
- CM_CORE__CORE Register Summary: [2]

Table 3-796. CM_L4CFG_MAILBOX13_CLKCTRL

Address Offset	0x0000 06A0		
Physical Address	0x4A00 8DA0	**Instance**	CM_CORE__CORE
Description	This register manages the MAILBOX13 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-797. Register Call Summary for Register CM_L4CFG_MAILBOX13_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-798. CM_L3INSTR_CLKSTCTRL

Address Offset	0x0000 0700		
Physical Address	0x4A00 8E00	**Instance**	CM_CORE__CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	R		

31:11	10	9	8	7:2	1:0
RESERVED	CLKACTIVITY_L3INSTR_TS_GCLK	CLKACTIVITY_L3INSTR_DLL_AGING_GCLK	CLKACTIVITY_L3INSTR_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_L3INSTR_TS_GCLK	This field indicates the state of the L3INSTR_TS_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_L3INSTR_DLL_AGING_GCLK	This field indicates the state of the L3INSTR_DLL_AGING_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L3INSTR_L3_GICLK	This field indicates the state of the L3INSTR_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L3INSTR clock domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	R	0x3

Table 3-799. Register Call Summary for Register CM_L3INSTR_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [4]

Table 3-800. CM_L3INSTR_L3_MAIN_2_CLKCTRL

Address Offset	0x0000 0720		
Physical Address	0x4A00 8E20	**Instance**	CM_CORE__CORE
Description	This register manages the L3_MAIN_2 clocks. [warm reset insensitive]		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-801. Register Call Summary for Register CM_L3INSTR_L3_MAIN_2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]
- CM_CORE__RESTORE Register Description: [3]

Table 3-802. CM_L3INSTR_L3_INSTR_CLKCTRL

Address Offset	0x0000 0728		
Physical Address	0x4A00 8E28	**Instance**	CM_CORE__CORE
Description	This register manages the L3 INSTRUMENTATION clocks. [warm reset insensitive]		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-803. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]
- CM_CORE__RESTORE Register Description: [3] [4]

Table 3-804. CM_L3INSTR_OCP_WP_NOC_CLKCTRL

Address Offset	0x0000 0740		
Physical Address	0x4A00 8E40	**Instance**	CM_CORE__CORE
Description	This register manages the OCP_WP_NOC clocks. [warm reset insensitive]		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-805. Register Call Summary for Register CM_L3INSTR_OCP_WP_NOC_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]
- CM_CORE__RESTORE Register Description: [3] [4]

Table 3-806. CM_L3INSTR_DLL_AGING_CLKCTRL

Address Offset	0x0000 0748		
Physical Address	0x4A00 8E48	**Instance**	CM_CORE__CORE
Description	This register manages the DLL_AGING clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-807. Register Call Summary for Register CM_L3INSTR_DLL_AGING_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

Table 3-807. Register Call Summary for Register CM_L3INSTR_DLL_AGING_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE__CORE Register Summary: [2]

Table 3-808. CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL

Address Offset	0x0000 0750		
Physical Address	0x4A00 8E50	**Instance**	CM_CORE__CORE
Description	This register manages the CTRL_MODULE_BANDGAP clock.		
Type	RW		

31:26	25:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	CLKSEL	Selects the divider value for generating the Thermal Sensor clock from WKUPAON_ICLK source. The divider has to be selected so as to guarantee a frequency between 1MHz and 2MHz. 0x0: Divide by 8 0x1: Divide by 16 0x2: Divide by 32 0x3: Reserved	RW	0x2
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-809. Register Call Summary for Register CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL

Clock Management Functional Description

- PRM Clock Source: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__CORE Register Summary: [3]

3.13.18 CM_CORE__CUSTEFUSE Registers

3.13.18.1 CM_CORE__CUSTEFUSE Register Summary

Table 3-810. CM_CORE__CUSTEFUSE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__CUSTEFUSE Physical Address L4_CFG Interconnect
CM_CUSTEFUSE_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9600
CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL	RW	32	0x0000 0020	0x4A00 9620

3.13.18.2 CM_CORE__CUSTEFUSE Register Description

Table 3-811. CM_CUSTEFUSE_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9600	**Instance**	CM_CORE__CUSTEFUSE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:10	9	8	7:2	1:0
RESERVED	CLKACTIVITY_CUSTEFUSE_SYS_GFCLK	CLKACTIVITY_CUSTEFUSE_L4_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_CUSTEFUSE_SYS_GFCLK	This field indicates the state of the CUSTEFUSE_SYS_CLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_CUSTEFUSE_L4_GICLK	This field indicates the state of the L4_CUSTEFUSE_GICLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the CUSTEFUSE clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-812. Register Call Summary for Register CM_CUSTEFUSE_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__CUSTEFUSE Register Summary: [3]

Table 3-813. CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9620	**Instance**	CM_CORE__CUSTEFUSE
Description	This register manages the CUSTEFUSE clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-814. Register Call Summary for Register CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__CUSTEFUSE Register Summary: [2]

3.13.19 CM_CORE__DSS Registers

3.13.19.1 CM_CORE__DSS Register Summary

Table 3-815. CM_CORE__DSS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__DSS Physical Address L4_CFG Interconnect
CM_DSS_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9100
CM_DSS_STATICDEP	RW	32	0x0000 0004	0x4A00 9104
CM_DSS_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9108
CM_DSS_DSS_CLKCTRL	RW	32	0x0000 0020	0x4A00 9120
CM_DSS_BB2D_CLKCTRL	RW	32	0x0000 0030	0x4A00 9130
RESERVED	R	32	0x0000 003C	0x4A00 913C

3.13.19.2 CM_CORE__DSS Register Description

Table 3-816. CM_DSS_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9100	**Instance**	CM_CORE__DSS
Description	This register enables the DSS domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

Bits	Field
31-19	RESERVED
18	CLKACTIVITY_HDMI_PHY_GFCLK
17	CLKACTIVITY_HDMI_CEC_GFCLK
16	RESERVED
15	CLKACTIVITY_DSS_L4_GICLK
14	RESERVED
13	CLKACTIVITY_BB2D_GFCLK
12	CLKACTIVITY_VIDEO2_DPLL_CLK
11	CLKACTIVITY_HDMI_DPLL_CLK
10	CLKACTIVITY_VIDEO1_DPLL_CLK
9	CLKACTIVITY_DSS_GFCLK
8	CLKACTIVITY_DSS_L3_GICLK
7-2	RESERVED
1-0	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	CLKACTIVITY_HDMI_PHY_GFCLK	This field indicates the state of the HDMI_PHY_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
17	CLKACTIVITY_HDMI_CEC_GFCLK	This field indicates the state of the HDMI_CEC_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	RESERVED		R	0x0
15	CLKACTIVITY_DSS_L4_GICLK	This field indicates the state of the DSS_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	RESERVED		R	0x0
13	CLKACTIVITY_BB2D_GFCLK	This field indicates the state of the BB2D_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_VIDEO2_DPLL_CLK	This field indicates the state of the VIDEO2_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_HDMI_DPLL_CLK	This field indicates the state of the HDMI_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_VIDEO1_DPLL_CLK	This field indicates the state of the VIDEO1_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_DSS_GFCLK	This field indicates the state of the DSS_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_DSS_L3_GICLK	This field indicates the state of the DSS_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the DSS clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-817. Register Call Summary for Register CM_DSS_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5] [7] [8] [9] [10]

PRCM Register Manual

- CM_CORE__DSS Register Summary: [11]

Table 3-818. CM_DSS_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9104	**Instance**	CM_CORE__DSS
Description	This register controls the static domain depedencies from DSS domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31:6	5	4	3	2	1:0
RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-819. Register Call Summary for Register CM_DSS_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2]

Table 3-819. Register Call Summary for Register CM_DSS_STATICDEP (continued)

PRCM Register Manual

- CM_CORE__DSS Register Summary: [3]

Table 3-820. CM_DSS_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9108	**Instance**	CM_CORE__DSS
Description	This register controls the dynamic domain depedencies from DSS domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

Bits	Field
31–6	RESERVED
5	L3MAIN1_DYNDEP
4–0	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-821. Register Call Summary for Register CM_DSS_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]

PRCM Register Manual

- CM_CORE__DSS Register Summary: [1]

Table 3-822. CM_DSS_DSS_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9120	**Instance**	CM_CORE__DSS
Description	This register manages the DSS clocks.		
Type	RW		

Bits	Field
31–19	RESERVED
18	STBYST
17–16	IDLEST
15–14	RESERVED
13	OPTFCLKEN_VIDEO2_CLK
12	OPTFCLKEN_VIDEO1_CLK
11	OPTFCLKEN_32KHZ_CLK
10	OPTFCLKEN_HDMI_CLK
9	OPTFCLKEN_48MHZ_CLK
8	OPTFCLKEN_DSSCLK
7–2	RESERVED
1–0	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:14	RESERVED		R	0x0
13	OPTFCLKEN_VIDEO2_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
12	OPTFCLKEN_VIDEO1_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
11	OPTFCLKEN_32KHZ_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
10	OPTFCLKEN_HDMI_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
9	OPTFCLKEN_48MHZ_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
8	OPTFCLKEN_DSSCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-823. Register Call Summary for Register CM_DSS_DSS_CLKCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5]
- Clock Domain Module Attributes: [6] [7] [8]

PRCM Register Manual

- CM_CORE__DSS Register Summary: [9]

Table 3-824. CM_DSS_BB2D_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9130	**Instance**	CM_CORE__DSS
Description	This register manages the BB2D clocks.		
Type	RW		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-825. Register Call Summary for Register CM_DSS_BB2D_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__DSS Register Summary: [3]

3.13.20 CM_CORE__GPU Registers

3.13.20.1 CM_CORE__GPU Register Summary

Table 3-826. CM_CORE__GPU Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__GPU Physical Address L4_CFG Interconnect
CM_GPU_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9200
CM_GPU_STATICDEP	RW	32	0x0000 0004	0x4A00 9204
CM_GPU_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9208
CM_GPU_GPU_CLKCTRL	RW	32	0x0000 0020	0x4A00 9220

3.13.20.2 CM_CORE__GPU Register Description

Table 3-827. CM_GPU_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9200	**Instance**	CM_CORE__GPU
Description	This register enables the GPU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:11	10	9	8	7:2	1:0
RESERVED	CLKACTIVITY_GPU_HYD_GCLK	CLKACTIVITY_GPU_CORE_GCLK	CLKACTIVITY_GPU_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKACTIVITY_GPU_HYD_GCLK	This field indicates the state of the GPU_HYD_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_GPU_CORE_GCLK	This field indicates the state of the GPU_CORE_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_GPU_L3_GICLK	This field indicates the state of the GPU_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the GPU clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-828. Register Call Summary for Register CM_GPU_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3]

PRCM Register Manual

- CM_CORE__GPU Register Summary: [4]

Table 3-829. CM_GPU_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9204	**Instance**	CM_CORE__GPU
Description	This register controls the static domain depedencies from GPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31:6	5	4	3	2	1:0
RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-830. Register Call Summary for Register CM_GPU_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2]

PRCM Register Manual

- CM_CORE__GPU Register Summary: [3]

Table 3-831. CM_GPU_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9208	**Instance**	CM_CORE__GPU
Description	This register controls the dynamic domain depedencies from GPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
5:0	RESERVED		R	0x0

Table 3-832. Register Call Summary for Register CM_GPU_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]

PRCM Register Manual

- CM_CORE__GPU Register Summary: [1]

Table 3-833. CM_GPU_GPU_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9220	**Instance**	CM_CORE__GPU
Description	This register manages the GPU clocks.		
Type	RW		

31:28	27:26	25:24	23:19	18	17:16	15:2	1:0
RESERVED	CLKSEL_HYD_CLK	CLKSEL_CORE_CLK	RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_HYD_CLK	Select the source of the functional clock 0x0: Selects the CORE_GPU_CLK as the source 0x1: Selects the PER_GPU_CLK 0x2: Selects GPU_GCLK 0x3: RESERVED	RW	0x0
25:24	CLKSEL_CORE_CLK	Select the source of the functional clock 0x0: Selects the CORE_GPU_CLK as the source 0x1: Selects the PER_GPU_CLK 0x2: Selects GPU_GCLK 0x3: RESERVED	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-834. Register Call Summary for Register CM_GPU_GPU_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Modes: [2] [3]
- Clock Domain Module Attributes: [4] [5] [6]

PRCM Register Manual

- CM_CORE__GPU Register Summary: [7]

3.13.21 CM_CORE__IVA Registers

3.13.21.1 CM_CORE__IVA Register Summary

Table 3-835. CM_CORE__IVA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__IVA Physical Address L4_CFG Interconnect
CM_IVA_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 8F00
CM_IVA_STATICDEP	RW	32	0x0000 0004	0x4A00 8F04
CM_IVA_DYNAMICDEP	R	32	0x0000 0008	0x4A00 8F08
CM_IVA_IVA_CLKCTRL	RW	32	0x0000 0020	0x4A00 8F20
CM_IVA_SL2_CLKCTRL	RW	32	0x0000 0028	0x4A00 8F28

3.13.21.2 CM_CORE__IVA Register Description

Table 3-836. CM_IVA_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 8F00	**Instance**	CM_CORE__IVA
Description	This register enables the IVA domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLKACTIVITY_IVA_GCLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_IVA_GCLK	This field indicates the state of the IVA_ROOT_CLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the IVA clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-837. Register Call Summary for Register CM_IVA_CLKSTCTRL

Reset Management Functional Description

- IVA Subsystem Software Warm Reset Sequence: [0]

Clock Management Functional Description

- Clock Domain Modes: [1] [2]

PRCM Register Manual

- CM_CORE__IVA Register Summary: [3]

Table 3-838. CM_IVA_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 8F04	**Instance**	CM_CORE__IVA
Description	This register controls the static domain depedencies from IVA domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31–6	5	4	3–0
RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3:0	RESERVED		R	0x0

Table 3-839. Register Call Summary for Register CM_IVA_STATICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0] [1]

PRCM Register Manual
- CM_CORE__IVA Register Summary: [2]

Table 3-840. CM_IVA_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 8F08	**Instance**	CM_CORE__IVA
Description	This register controls the dynamic domain depedencies from IVA domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31:6	5	4:0
RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-841. Register Call Summary for Register CM_IVA_DYNAMICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0]

PRCM Register Manual
- CM_CORE__IVA Register Summary: [1]

Table 3-842. CM_IVA_IVA_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 8F20	**Instance**	CM_CORE__IVA
Description	This register manages the IVA clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-843. Register Call Summary for Register CM_IVA_IVA_CLKCTRL

Reset Management Functional Description

- IVA Subsystem Software Warm Reset Sequence: [0] [1]

Clock Management Functional Description

- Clock Domain Module Attributes: [2] [3] [4]

PRCM Register Manual

- CM_CORE__IVA Register Summary: [5]

Table 3-844. CM_IVA_SL2_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 8F28	**Instance**	CM_CORE__IVA
Description	This register manages the SL2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-845. Register Call Summary for Register CM_IVA_SL2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__IVA Register Summary: [2]

3.13.22 CM_CORE__L3INIT Registers

3.13.22.1 CM_CORE__L3INIT Register Summary

Table 3-846. CM_CORE__L3INIT Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__L3INIT Physical Address L4_CFG Interconnect
CM_L3INIT_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9300
CM_L3INIT_STATICDEP	RW	32	0x0000 0004	0x4A00 9304
CM_L3INIT_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9308
CM_L3INIT_MMC1_CLKCTRL	RW	32	0x0000 0028	0x4A00 9328
CM_L3INIT_MMC2_CLKCTRL	RW	32	0x0000 0030	0x4A00 9330
CM_L3INIT_USB_OTG_SS2_CLKCTRL	RW	32	0x0000 0040	0x4A00 9340
CM_L3INIT_USB_OTG_SS3_CLKCTRL	RW	32	0x0000 0048	0x4A00 9348
CM_L3INIT_USB_OTG_SS4_CLKCTRL	RW	32	0x0000 0050	0x4A00 9350
CM_L3INIT_MLB_SS_CLKCTRL	RW	32	0x0000 0058	0x4A00 9358
CM_L3INIT_IEEE1500_2_OCP_CLKCTRL	R	32	0x0000 0078	0x4A00 9378
CM_L3INIT_SATA_CLKCTRL	RW	32	0x0000 0088	0x4A00 9388
CM_PCIE_CLKSTCTRL	RW	32	0x0000 00A0	0x4A00 93A0
CM_PCIE_STATICDEP	RW	32	0x0000 00A4	0x4A00 93A4
CM_PCIE_PCIESS1_CLKCTRL	RW	32	0x0000 00B0	0x4A00 93B0
CM_PCIE_PCIESS2_CLKCTRL	RW	32	0x0000 00B8	0x4A00 93B8
CM_GMAC_CLKSTCTRL	RW	32	0x0000 00C0	0x4A00 93C0
CM_GMAC_STATICDEP	RW	32	0x0000 00C4	0x4A00 93C4
CM_GMAC_DYNAMICDEP	R	32	0x0000 00C8	0x4A00 93C8
CM_GMAC_GMAC_CLKCTRL	RW	32	0x0000 00D0	0x4A00 93D0

Table 3-846. CM_CORE__L3INIT Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__L3INIT Physical Address L4_CFG Interconnect
CM_L3INIT_OCP2SCP1_CLKCTRL	RW	32	0x0000 00E0	0x4A00 93E0
CM_L3INIT_OCP2SCP3_CLKCTRL	RW	32	0x0000 00E8	0x4A00 93E8
CM_L3INIT_USB_OTG_SS1_CLKCTRL	RW	32	0x0000 00F0	0x4A00 93F0

3.13.22.2 CM_CORE__L3INIT Register Description

Table 3-847. CM_L3INIT_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9300	**Instance**	CM_CORE__L3INIT
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

Bits	Field
31-25	RESERVED
24	CLKACTIVITY_SATA_REF_GFCLK
23	CLKACTIVITY_L3INIT_32K_GFCLK
22	CLKACTIVITY_L3INIT_960M_GFCLK
21	CLKACTIVITY_L3INIT_480M_GFCLK
20	CLKACTIVITY_USB_OTG_SS_REF_CLK
19	CLKACTIVITY_MLB_SYS_L3_GFCLK
18	CLKACTIVITY_MLB_SPB_L4_GICLK
17	CLKACTIVITY_MLB_SHB_L3_GICLK
16	CLKACTIVITY_MMC2_GFCLK
15	CLKACTIVITY_MMC1_GFCLK
14	RESERVED
13	CLKACTIVITY_USB_DPLL_HS_CLK
12	CLKACTIVITY_USB_DPLL_CLK
11	CLKACTIVITY_L3INIT_48M_GFCLK
10	CLKACTIVITY_L3INIT_USB_LFPS_TX_GFCLK
9	CLKACTIVITY_L3INIT_L4_GICLK
8	CLKACTIVITY_L3INIT_L3_GICLK
7-2	RESERVED
1-0	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKACTIVITY_SATA_REF_GFCLK	This field indicates the state of the SATA_REF_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
23	CLKACTIVITY_L3INIT_32K_GFCLK	This field indicates the state of the L3INIT_32K_FCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
22	CLKACTIVITY_L3INIT_960M_GFCLK	This field indicates the state of the L3INIT_960M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
21	CLKACTIVITY_L3INIT_480M_GFCLK	This field indicates the state of the L3INIT_480M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
20	CLKACTIVITY_USB_OTG_SS_REF_CLK	This field indicates the state of the USB_OTG_SS_REF_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
19	CLKACTIVITY_MLB_SYS_L3_GFCLK	This field indicates the state of the MLB_SYS_L3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
18	CLKACTIVITY_MLB_SPB_L4_GICLK	This field indicates the state of the MLB_SPB_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_MLB_SHB_L3_GICLK	This field indicates the state of the MLB_SHB_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_MMC2_GFCLK	This field indicates the state of the MMC2 clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_MMC1_GFCLK	This field indicates the state of the MMC1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	RESERVED		R	0x0
13	CLKACTIVITY_USB_DPLL_HS_CLK	This field indicates the state of the USB_DPLL_HS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_USB_DPLL_CLK	This field indicates the state of the USB_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_L3INIT_48M_GFCLK	This field indicates the state of the INIT_48M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_L3INIT_USB_LFPS_TX_GFCLK	This field indicates the state of the L3INIT_USB_LFPS_TX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_L3INIT_L4_GICLK	This field indicates the state of the L3INIT_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L3INIT_L3_GICLK	This field indicates the state of the L3INIT_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L3INIT clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-848. Register Call Summary for Register CM_L3INIT_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [17]
- CM_CORE__RESTORE Register Description: [18] [19]

Table 3-849. CM_L3INIT_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9304	**Instance**	CM_CORE__L3INIT
Description	This register controls the static domain depedencies from L3INIT domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				L4PER3_STATDEP	RESERVED											WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED						L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26:16	RESERVED		R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-850. Register Call Summary for Register CM_L3INIT_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [8]

Table 3-851. CM_L3INIT_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9308	**Instance**	CM_CORE__L3INIT
Description	This register controls the dynamic domain depedencies from L3INIT domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is Disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-852. Register Call Summary for Register CM_L3INIT_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [1]

Table 3-853. CM_L3INIT_MMC1_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 9328	**Instance**	CM_CORE__L3INIT
Description	This register manages the MMC1 clocks.		
Type	RW		

31:27	26:25	24	23:19	18	17:16	15:9	8	7:2	1:0
RESERVED	CLKSEL_DIV	CLKSEL_SOURCE	RESERVED	STBYST	IDLEST	RESERVED	OPTFCLKEN_CLK32K	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	MMC1 clock divide ratio. 0x0: MMC1 clock is divided by 1. 0x1: MMC1 clock is divided by 2. 0x2: MMC1 clock is divided by 4. 0x3: RESERVED	RW	0x0
24	CLKSEL_SOURCE	Selects the source of the functional clock. 0x0: 128MHz clock derived from DPLL_PER is selected 0x1: 192MHz clock derived from DPLL_PER is selected	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-854. Register Call Summary for Register CM_L3INIT_MMC1_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Modes: [2]
- Clock Domain Module Attributes: [3] [4] [5]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [6]

Table 3-855. CM_L3INIT_MMC2_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9330	**Instance**	CM_CORE__L3INIT
Description	This register manages the MMC2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_DIV		CLKSEL_SOURCE	RESERVED					STBYST	IDLEST		RESERVED							OPTFCLKEN_CLK32K	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:25	CLKSEL_DIV	MMC2 clock divide ratio 0x0: MMC2 clock is divided by 1. 0x1: MMC2 clock is divided by 2. 0x2: MMC2 clock is divided by 4. 0x3: RESERVED	RW	0x0
24	CLKSEL_SOURCE	Selects the source of the functional clock. 0x0: 128MHz clock derived from DPLL_PER is selected 0x1: 192MHz clock derived from DPLL_PER is selected	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-856. Register Call Summary for Register CM_L3INIT_MMC2_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Modes: [2]
- Clock Domain Module Attributes: [3] [4] [5]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [6]

Table 3-857. CM_L3INIT_USB_OTG_SS2_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 9340	**Instance**	CM_CORE__L3INIT
Description	This register manages the USB_OTG_SS2 clocks.		
Type	RW		

31:19	18	17:16	15:9	8	7:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	OPTFCLKEN_REFCLK960M	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_REFCLK960M	USB_OTG_SS optional clock control: REFCLK960M (960MHz clock) 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-858. Register Call Summary for Register CM_L3INIT_USB_OTG_SS2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

Table 3-859. CM_L3INIT_USB_OTG_SS3_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4A00 9348	**Instance**	CM_CORE__L3INIT
Description	This register manages the USB_OTG_SS3 clocks.		

Table 3-859. CM_L3INIT_USB_OTG_SS3_CLKCTRL (continued)

Type	RW

31–19	18	17–16	15–2	1–0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-860. Register Call Summary for Register CM_L3INIT_USB_OTG_SS3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

Table 3-861. CM_L3INIT_USB_OTG_SS4_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 9350	**Instance**	CM_CORE__L3INIT
Description	This register manages the USB_OTG_SS4 clocks.		
Type	RW		

31–19	18	17–16	15–2	1–0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-862. Register Call Summary for Register CM_L3INIT_USB_OTG_SS4_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

Table 3-863. CM_L3INIT_MLB_SS_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 9358	**Instance**	CM_CORE__L3INIT
Description	This register manages the MLBSS clocks.		
Type	RW		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-864. Register Call Summary for Register CM_L3INIT_MLB_SS_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

Table 3-865. CM_L3INIT_IEEE1500_2_OCP_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 9378	**Instance**	CM_CORE__L3INIT
Description	This register manages the IEE1500_2_OCP clocks.		
Type	R		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved	R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-866. Register Call Summary for Register CM_L3INIT_IEEE1500_2_OCP_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

Table 3-867. CM_L3INIT_SATA_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 9388	**Instance**	CM_CORE__L3INIT
Description	This register manages the SATA clocks.		
Type	RW		

31:19	18	17:16	15:9	8	7:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	OPTFCLKEN_REF_CLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved	R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_REF_CLK	SATA optional clock control: REF_CLK (from SYS_CLK clock) 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-868. Register Call Summary for Register CM_L3INIT_SATA_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

Table 3-869. CM_PCIE_CLKSTCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4A00 93A0	**Instance**	CM_CORE__L3INIT
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:14	13	12	11	10	9	8	7:2	1:0
RESERVED	CLKACTIVITY_PCIE_32K_GFCLK	CLKACTIVITY_PCIE_SYS_GFCLK	CLKACTIVITY_PCIE_REF_GFCLK	CLKACTIVITY_PCIE_PHY_DIV_GCLK	CLKACTIVITY_PCIE_PHY_GCLK	CLKACTIVITY_PCIE_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	CLKACTIVITY_PCIE_32K_GFCLK	This field indicates the state of the PCIE_32K_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_PCIE_SYS_GFCLK	This field indicates the state of the PCIE_SYS_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
11	CLKACTIVITY_PCIE_REF_GFCLK	This field indicates the state of the PCIE_REF_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_PCIE_PHY_DIV_GCLK	This field indicates the state of the PCIE_PHY_DIV_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_PCIE_PHY_GCLK	This field indicates the state of the PCIE_PHY_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_PCIE_L3_GICLK	This field indicates the state of the PCIE_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L3INIT clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-870. Register Call Summary for Register CM_PCIE_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [7]

Table 3-871. CM_PCIE_STATICDEP

Address Offset	0x0000 00A4		
Physical Address	0x4A00 93A4	**Instance**	CM_CORE__L3INIT
Description	This register controls the static domain depedencies from PCIE domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ATL_STATDEP	RESERVED	VPE_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	RESERVED	RESERVED	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP	RESERVED	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ATL_STATDEP	Static dependency towards ATL clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
29	RESERVED		R	0x0
28	VPE_STATDEP	Static dependency towards VPE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	RESERVED		R	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	SDMA_STATDEP	Static dependency towards SDMA clock domain 0x0: Dependency is disabled	R	0x0
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6:5	RESERVED		R	0x0
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	RESERVED		R	0x0

Table 3-872. Register Call Summary for Register CM_PCIE_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [20] [21] [22] [23] [24]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [25]

Table 3-873. CM_PCIE_PCIESS1_CLKCTRL

Address Offset	0x0000 00B0		
Physical Address	0x4A00 93B0	**Instance**	CM_CORE__L3INIT
Description	This register manages the PCESS1 clocks.		

Table 3-873. CM_PCIE_PCIESS1_CLKCTRL (continued)

Type	RW

31:19	18	17:16	15:11	10	9	8	7:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	OPTFCLKEN_PCIEPHY_CLK_DIV	OPTFCLKEN_PCIEPHY_CLK	OPTFCLKEN_32KHZ	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:11	RESERVED		R	0x0
10	OPTFCLKEN_PCIEPHY_CLK_DIV	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
9	OPTFCLKEN_PCIEPHY_CLK	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
8	OPTFCLKEN_32KHZ	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. **Note:** In order to disable the APLL_PCIE, the user needs to disable PCIe_SSx (where x = 1 or 2) using the CM_PCIE_ PCIESSx_CLKCTRL[1:0] MODULEMODE registers. When PCIe_SS is disabled, the PRCM module automatically disables the APLL_PCIE. Please note that setting CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT bitfield to 0x0 does not disable the APLL_PCIE. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-874. Register Call Summary for Register CM_PCIE_PCIESS1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2] [3] [4] [5]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [6]

Table 3-875. CM_PCIE_PCIESS2_CLKCTRL

Address Offset	0x0000 00B8		
Physical Address	0x4A00 93B8	**Instance**	CM_CORE__L3INIT
Description	This register manages the PCESS2 clocks.		
Type	RW		

31:19	18	17:16	15:11	10	9	8	7:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	OPTFCLKEN_PCIEPHY_CLK_DIV	OPTFCLKEN_PCIEPHY_CLK	OPTFCLKEN_32KHZ	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:11	RESERVED		R	0x0
10	OPTFCLKEN_PCIEPHY_CLK_DIV	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
9	OPTFCLKEN_PCIEPHY_CLK	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
8	OPTFCLKEN_32KHZ	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. **Note:** In order to disable the APLL_PCIE, the user needs to disable PCIe_SSx (where x = 1 or 2) using the CM_PCIE_ PCIESSx_CLKCTRL[1:0] MODULEMODE registers. When PCIe_SS is disabled, the PRCM module automatically disables the APLL_PCIE. Please note that setting CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT bitfield to 0x0 does not disable the APLL_PCIE. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-876. Register Call Summary for Register CM_PCIE_PCIESS2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2] [3] [4] [5]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [6]

Table 3-877. CM_GMAC_CLKSTCTRL

Address Offset	0x0000 00C0		
Physical Address	0x4A00 93C0	**Instance**	CM_CORE__L3INIT
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13	12	11	10	9	8	7 6 5 4 3 2	1 0
RESERVED	CLKACTIVITY_GMAC_MAIN_CLK	CLKACTIVITY_GMAC_RFT_CLK	CLKACTIVITY_RMII_50MHZ_CLK	CLKACTIVITY_RGMII_5MHZ_CLK	CLKACTIVITY_GMII_250MHZ_CLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	CLKACTIVITY_GMAC_MAIN_CLK	This field indicates the state of the GMAC_MAIN_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_GMAC_RFT_CLK	This field indicates the state of the GMAC_RFT_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_RMII_50MHZ_CLK	This field indicates the state of the RMII_50MHZ_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_RGMII_5MHZ_CLK	This field indicates the state of the RGMII_5MHZ_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_GMII_250MHZ_CLK	This field indicates the state of the GMII_250MHZ_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	**WARNING**: This bit field must not be programmed for SW_SLEEP or HW_AUTO for EEE mode. Controls the clock state transition of the GMAC clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-878. Register Call Summary for Register CM_GMAC_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [6]

Table 3-879. CM_GMAC_STATICDEP

Address Offset	0x0000 00C4		
Physical Address	0x4A00 93C4	**Instance**	CM_CORE__L3INIT
Description	This register controls the static domain depedencies from GMAC domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31:27	26	25:6	5	4	3:0
RESERVED	L4PER2_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3:0	RESERVED		R	0x0

Table 3-880. Register Call Summary for Register CM_GMAC_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

Table 3-881. CM_GMAC_DYNAMICDEP

Address Offset	0x0000 00C8		
Physical Address	0x4A00 93C8	**Instance**	CM_CORE__L3INIT
Description	This register controls the dynamic domain depedencies from GMAC domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:6	5	4:0
RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-882. Register Call Summary for Register CM_GMAC_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [1]

Table 3-883. CM_GMAC_GMAC_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4A00 93D0	**Instance**	CM_CORE__L3INIT
Description	This register manages the GMAC clocks.		
Type	RW		

31:28	27:25	24	23:19	18	17:16	15:2	1:0
RESERVED	CLKSEL_RFT	CLKSEL_REF	RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:25	CLKSEL_RFT	Selects the source of the GMAC_RFT_CLK. [warm reset insensitive] 0x0: Selects VIDEO1_CLK derived from DPLL_VIDEO1 0x1: Selects VIDEO2_CLK derived from DPLL_VIDEO2 0x2: Selects PER_ABE_X1_GFCLK derived from DPLL_ABE 0x3: Selects HDMI_CLK derived from DPLL_HDMI 0x4: Selects L3_ICLK 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x4
24	CLKSEL_REF	Selects the source of the RMII_50MHZ_CLK functional clock. [warm reset insensitive] 0x0: Selects GMAC_RMII_HS_CLK 0x1: Selects GMAC_RMII_CLK	RW	0x0
23:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0
1:0	MODULEMODE	Controsl how mandatory clocks are managed. [warm reset insensitive] 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-884. Register Call Summary for Register CM_GMAC_GMAC_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Module Attributes: [2] [3] [4] [5] [6]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [7]

Table 3-885. CM_L3INIT_OCP2SCP1_CLKCTRL

Address Offset	0x0000 00E0		
Physical Address	0x4A00 93E0	**Instance**	CM_CORE__L3INIT
Description	This register manages the OCP2SCP1 clocks and the optional clock of USB PHY.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-886. Register Call Summary for Register CM_L3INIT_OCP2SCP1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [2]

Table 3-887. CM_L3INIT_OCP2SCP3_CLKCTRL

Address Offset	0x0000 00E8		
Physical Address	0x4A00 93E8	**Instance**	CM_CORE__L3INIT
Description	This register manages the OCP2SCP3 clocks and the optional clock of USB PHY.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-888. Register Call Summary for Register CM_L3INIT_OCP2SCP3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [2]

Table 3-889. CM_L3INIT_USB_OTG_SS1_CLKCTRL

Address Offset	0x0000 00F0		
Physical Address	0x4A00 93F0	**Instance**	CM_CORE__L3INIT
Description	This register manages the USB_OTG_SS1 clocks.		
Type	RW		

31:19	18	17:16	15:9	8	7:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	OPTFCLKEN_REFCLK960M	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_REFCLK960M	USB_OTG_SS optional clock control: REFCLK960M (960MHz clock) 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-890. Register Call Summary for Register CM_L3INIT_USB_OTG_SS1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L3INIT Register Summary: [3]

3.13.23 CM_CORE__L4PER Registers

3.13.23.1 CM_CORE__L4PER Register Summary

Table 3-891. CM_CORE__L4PER Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__L4PER Physical Address L4_CFG Interconnect
CM_L4PER_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9700
CM_L4PER_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 9708
CM_L4PER2_L4_PER2_CLKCTRL	R	32	0x0000 000C	0x4A00 970C
CM_L4PER3_L4_PER3_CLKCTRL	R	32	0x0000 0014	0x4A00 9714
RESERVED	R	32	0x0000 0018	0x4A00 9718
RESERVED	R	32	0x0000 0020	0x4A00 9720
CM_L4PER_TIMER10_CLKCTRL	RW	32	0x0000 0028	0x4A00 9728
CM_L4PER_TIMER11_CLKCTRL	RW	32	0x0000 0030	0x4A00 9730
CM_L4PER_TIMER2_CLKCTRL	RW	32	0x0000 0038	0x4A00 9738
CM_L4PER_TIMER3_CLKCTRL	RW	32	0x0000 0040	0x4A00 9740
CM_L4PER_TIMER4_CLKCTRL	RW	32	0x0000 0048	0x4A00 9748
CM_L4PER_TIMER9_CLKCTRL	RW	32	0x0000 0050	0x4A00 9750
CM_L4PER_ELM_CLKCTRL	R	32	0x0000 0058	0x4A00 9758
CM_L4PER_GPIO2_CLKCTRL	RW	32	0x0000 0060	0x4A00 9760
CM_L4PER_GPIO3_CLKCTRL	RW	32	0x0000 0068	0x4A00 9768
CM_L4PER_GPIO4_CLKCTRL	RW	32	0x0000 0070	0x4A00 9770
CM_L4PER_GPIO5_CLKCTRL	RW	32	0x0000 0078	0x4A00 9778
CM_L4PER_GPIO6_CLKCTRL	RW	32	0x0000 0080	0x4A00 9780
CM_L4PER_HDQ1W_CLKCTRL	RW	32	0x0000 0088	0x4A00 9788
CM_L4PER2_PWMSS2_CLKCTRL	RW	32	0x0000 0090	0x4A00 9790
CM_L4PER2_PWMSS3_CLKCTRL	RW	32	0x0000 0098	0x4A00 9798
CM_L4PER_I2C1_CLKCTRL	RW	32	0x0000 00A0	0x4A00 97A0
CM_L4PER_I2C2_CLKCTRL	RW	32	0x0000 00A8	0x4A00 97A8
CM_L4PER_I2C3_CLKCTRL	RW	32	0x0000 00B0	0x4A00 97B0
CM_L4PER_I2C4_CLKCTRL	RW	32	0x0000 00B8	0x4A00 97B8
CM_L4PER_L4_PER1_CLKCTRL	R	32	0x0000 00C0	0x4A00 97C0
CM_L4PER2_PWMSS1_CLKCTRL	RW	32	0x0000 00C4	0x4A00 97C4

Table 3-891. CM_CORE__L4PER Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__L4PER Physical Address L4_CFG Interconnect
CM_L4PER3_TIMER13_CLKCTRL	RW	32	0x0000 00C8	0x4A00 97C8
CM_L4PER3_TIMER14_CLKCTRL	RW	32	0x0000 00D0	0x4A00 97D0
CM_L4PER3_TIMER15_CLKCTRL	RW	32	0x0000 00D8	0x4A00 97D8
CM_L4PER_MCSPI1_CLKCTRL	RW	32	0x0000 00F0	0x4A00 97F0
CM_L4PER_MCSPI2_CLKCTRL	RW	32	0x0000 00F8	0x4A00 97F8
CM_L4PER_MCSPI3_CLKCTRL	RW	32	0x0000 0100	0x4A00 9800
CM_L4PER_MCSPI4_CLKCTRL	RW	32	0x0000 0108	0x4A00 9808
CM_L4PER_GPIO7_CLKCTRL	RW	32	0x0000 0110	0x4A00 9810
CM_L4PER_GPIO8_CLKCTRL	RW	32	0x0000 0118	0x4A00 9818
CM_L4PER_MMC3_CLKCTRL	RW	32	0x0000 0120	0x4A00 9820
CM_L4PER_MMC4_CLKCTRL	RW	32	0x0000 0128	0x4A00 9828
CM_L4PER3_TIMER16_CLKCTRL	RW	32	0x0000 0130	0x4A00 9830
CM_L4PER2_QSPI_CLKCTRL	RW	32	0x0000 0138	0x4A00 9838
CM_L4PER_UART1_CLKCTRL	RW	32	0x0000 0140	0x4A00 9840
CM_L4PER_UART2_CLKCTRL	RW	32	0x0000 0148	0x4A00 9848
CM_L4PER_UART3_CLKCTRL	RW	32	0x0000 0150	0x4A00 9850
CM_L4PER_UART4_CLKCTRL	RW	32	0x0000 0158	0x4A00 9858
CM_L4PER2_MCASP2_CLKCTRL	RW	32	0x0000 0160	0x4A00 9860
CM_L4PER2_MCASP3_CLKCTRL	RW	32	0x0000 0168	0x4A00 9868
CM_L4PER_UART5_CLKCTRL	RW	32	0x0000 0170	0x4A00 9870
CM_L4PER2_MCASP5_CLKCTRL	RW	32	0x0000 0178	0x4A00 9878
CM_L4SEC_CLKSTCTRL	RW	32	0x0000 0180	0x4A00 9880
CM_L4SEC_STATICDEP	RW	32	0x0000 0184	0x4A00 9884
CM_L4SEC_DYNAMICDEP	R	32	0x0000 0188	0x4A00 9888
CM_L4PER2_MCASP8_CLKCTRL	RW	32	0x0000 0190	0x4A00 9890
CM_L4PER2_MCASP4_CLKCTRL	RW	32	0x0000 0198	0x4A00 9898
CM_L4SEC_AES1_CLKCTRL	RW	32	0x0000 01A0	0x4A00 98A0
CM_L4SEC_AES2_CLKCTRL	RW	32	0x0000 01A8	0x4A00 98A8
CM_L4SEC_DES3DES_CLKCTRL	RW	32	0x0000 01B0	0x4A00 98B0
CM_L4SEC_FPKA_CLKCTRL	RW	32	0x0000 01B8	0x4A00 98B8
CM_L4SEC_RNG_CLKCTRL	RW	32	0x0000 01C0	0x4A00 98C0
CM_L4SEC_SHA2MD51_CLKCTRL	RW	32	0x0000 01C8	0x4A00 98C8
CM_L4PER2_UART7_CLKCTRL	RW	32	0x0000 01D0	0x4A00 98D0
CM_L4SEC_DMA_CRYPTO_CLKCTRL	R	32	0x0000 01D8	0x4A00 98D8
CM_L4PER2_UART8_CLKCTRL	RW	32	0x0000 01E0	0x4A00 98E0
CM_L4PER2_UART9_CLKCTRL	RW	32	0x0000 01E8	0x4A00 98E8
CM_L4PER2_DCAN2_CLKCTRL	RW	32	0x0000 01F0	0x4A00 98F0
CM_L4SEC_SHA2MD52_CLKCTRL	RW	32	0x0000 01F8	0x4A00 98F8
CM_L4PER2_CLKSTCTRL	RW	32	0x0000 01FC	0x4A00 98FC
CM_L4PER2_DYNAMICDEP	RW	32	0x0000 0200	0x4A00 9900
CM_L4PER2_MCASP6_CLKCTRL	RW	32	0x0000 0204	0x4A00 9904
CM_L4PER2_MCASP7_CLKCTRL	RW	32	0x0000 0208	0x4A00 9908
CM_L4PER2_STATICDEP	RW	32	0x0000 020C	0x4A00 990C
CM_L4PER3_CLKSTCTRL	RW	32	0x0000 0210	0x4A00 9910
CM_L4PER3_DYNAMICDEP	RW	32	0x0000 0214	0x4A00 9914

3.13.23.2 CM_CORE__L4PER Register Description

Table 3-892. CM_L4PER_CLKSTCTRL

Address Offset	0x0000 0000

Table 3-892. CM_L4PER_CLKSTCTRL (continued)

Physical Address	0x4A00 9700	**Instance**	CM_CORE__L4PER
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>CLKACTIVITY_L4PER_32K_GFCLK</td><td>CLKACTIVITY_UART5_GFCLK</td><td>RESERVED</td><td>CLKACTIVITY_GPIO_GFCLK</td><td>CLKACTIVITY_MMC4_GFCLK</td><td>CLKACTIVITY_MMC3_GFCLK</td><td>CLKACTIVITY_PER_96M_GFCLK</td><td>CLKACTIVITY_PER_48M_GFCLK</td><td>CLKACTIVITY_PER_12M_GFCLK</td><td>CLKACTIVITY_UART4_GFCLK</td><td>CLKACTIVITY_UART3_GFCLK</td><td>CLKACTIVITY_UART2_GFCLK</td><td>CLKACTIVITY_UART1_GFCLK</td><td>CLKACTIVITY_TIMER9_GFCLK</td><td>CLKACTIVITY_TIMER4_GFCLK</td><td>CLKACTIVITY_TIMER3_GFCLK</td><td>CLKACTIVITY_TIMER2_GFCLK</td><td>CLKACTIVITY_TIMER11_GFCLK</td><td>CLKACTIVITY_TIMER10_GFCLK</td><td>CLKACTIVITY_L4PER_L3_GICLK</td><td colspan="6">RESERVED</td><td colspan="2">CLKTRCTRL</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	CLKACTIVITY_L4PER_32K_GFCLK	This field indicates the state of the L4PER_32K_FCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
26	CLKACTIVITY_UART5_GFCLK	This field indicates the state of the UART5_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
25	RESERVED		R	0x0
24	CLKACTIVITY_GPIO_GFCLK	This field indicates the state of the GPIO_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
23	CLKACTIVITY_MMC4_GFCLK	This field indicates the state of the MMC4_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
22	CLKACTIVITY_MMC3_GFCLK	This field indicates the state of the MMC3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
21	CLKACTIVITY_PER_96M_GFCLK	This field indicates the state of the PER_96M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
20	CLKACTIVITY_PER_48M_GFCLK	This field indicates the state of the PER_48M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
19	CLKACTIVITY_PER_12M_GFCLK	This field indicates the state of the PER_12M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
18	CLKACTIVITY_UART4_GFCLK	This field indicates the state of the UART4_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_UART3_GFCLK	This field indicates the state of the UART3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_UART2_GFCLK	This field indicates the state of the UART2_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_UART1_GFCLK	This field indicates the state of the UART1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_TIMER9_GFCLK	This field indicates the state of the DMT9_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_TIMER4_GFCLK	This field indicates the state of the DMT4_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_TIMER3_GFCLK	This field indicates the state of the DMT3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_TIMER2_GFCLK	This field indicates the state of the DMT2_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_TIMER11_GFCLK	This field indicates the state of the DMT11_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_TIMER10_GFCLK	This field indicates the state of the DMT10_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L4PER_L3_GICLK	This field indicates the state of the L4PER_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-893. Register Call Summary for Register CM_L4PER_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [20]
- CM_CORE__RESTORE Register Description: [21] [22]

Table 3-894. CM_L4PER_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9708	**Instance**	CM_CORE__L4PER
Description	This register controls the dynamic domain depedencies from L4PER domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31:28	27:24	23:15	14	13:9	8	7	6:4	3	2:0
RESERVED	WINDOWSIZE	RESERVED	L4SEC_DYNDEP	RESERVED	DSS_DYNDEP	L3INIT_DYNDEP	RESERVED	IPU_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:15	RESERVED		R	0x0
14	L4SEC_DYNDEP	Dynamic dependency towards L4SEC clock domain 0x1: Dependency is enabled	R	0x1
13:9	RESERVED		R	0x0
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain 0x1: Dependency is enabled	R	0x1
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6:4	RESERVED		R	0x0
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2:0	RESERVED		R	0x0

Table 3-895. Register Call Summary for Register CM_L4PER_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]
- CM_CORE__RESTORE Register Description: [5] [6]

Table 3-896. CM_L4PER2_L4_PER2_CLKCTRL

Address Offset	0x0000 000C		
Physical Address	0x4A00 970C	**Instance**	CM_CORE__L4PER
Description	This register manages the L4_PER2 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-897. Register Call Summary for Register CM_L4PER2_L4_PER2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-898. CM_L4PER3_L4_PER3_CLKCTRL

Address Offset	0x0000 0014		
Physical Address	0x4A00 9714	**Instance**	CM_CORE__L4PER
Description	This register manages the L4_PER3 clocks.		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-899. Register Call Summary for Register CM_L4PER3_L4_PER3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-900. CM_L4PER_TIMER10_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 9728	**Instance**	CM_CORE__L4PER

Table 3-900. CM_L4PER_TIMER10_CLKCTRL (continued)

Description	This register manages the TIMER10 clocks.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-901. Register Call Summary for Register CM_L4PER_TIMER10_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-902. CM_L4PER_TIMER11_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9730	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER11 clocks.		
Type	RW		

31:28	27:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-903. Register Call Summary for Register CM_L4PER_TIMER11_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

Table 3-903. Register Call Summary for Register CM_L4PER_TIMER11_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-904. CM_L4PER_TIMER2_CLKCTRL

Address Offset	0x0000 0038		
Physical Address	0x4A00 9738	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER2 clocks.		
Type	RW		

31:28	27:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-905. Register Call Summary for Register CM_L4PER_TIMER2_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-906. CM_L4PER_TIMER3_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 9740	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER3 clocks.		
Type	RW		

31:28	27:24	23:18	17:16	15:1	0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-907. Register Call Summary for Register CM_L4PER_TIMER3_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual
- CM_CORE__L4PER Register Summary: [3]

Table 3-908. CM_L4PER_TIMER4_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4A00 9748	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER4 clocks.		
Type	RW		

31:28	27:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-909. Register Call Summary for Register CM_L4PER_TIMER4_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-910. CM_L4PER_TIMER9_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 9750	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER9 clocks.		
Type	RW		

31:28	27:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-911. Register Call Summary for Register CM_L4PER_TIMER9_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-912. CM_L4PER_ELM_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 9758	**Instance**	CM_CORE__L4PER
Description	This register manages the ELM clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-913. Register Call Summary for Register CM_L4PER_ELM_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual
- CM_CORE__L4PER Register Summary: [2]

Table 3-914. CM_L4PER_GPIO2_CLKCTRL

Address Offset	0x0000 0060		
Physical Address	0x4A00 9760	**Instance**	CM_CORE__L4PER
Description	This register manages the GPIO2 clocks.		
Type	RW		

31–18	17–16	15–9	8	7–2	1–0
RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-915. Register Call Summary for Register CM_L4PER_GPIO2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-916. CM_L4PER_GPIO3_CLKCTRL

Address Offset	0x0000 0068		
Physical Address	0x4A00 9768	**Instance**	CM_CORE__L4PER
Description	This register manages the GPIO3 clocks.		
Type	RW		

31–18	17–16	15–9	8	7–2	1–0
RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-917. Register Call Summary for Register CM_L4PER_GPIO3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-918. CM_L4PER_GPIO4_CLKCTRL

Address Offset	0x0000 0070		
Physical Address	0x4A00 9770	**Instance**	CM_CORE__L4PER
Description	This register manages the GPIO4 clocks.		
Type	RW		

31–18	17–16	15–9	8	7–2	1–0
RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-919. Register Call Summary for Register CM_L4PER_GPIO4_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-920. CM_L4PER_GPIO5_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 9778	**Instance**	CM_CORE__L4PER
Description	This register manages the GPIO5 clocks.		
Type	RW		

31:18	17:16	15:9	8	7:2	1:0
RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-921. Register Call Summary for Register CM_L4PER_GPIO5_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-922. CM_L4PER_GPIO6_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4A00 9780	**Instance**	CM_CORE__L4PER
Description	This register manages the GPIO6 clocks.		
Type	RW		

31:18	17:16	15:9	8	7:2	1:0
RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	odule idle status. [warm reset insensitive]	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: FCLK_DIS 0x1: FCLK_EN	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed.	RW	0x0

Table 3-923. Register Call Summary for Register CM_L4PER_GPIO6_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-924. CM_L4PER_HDQ1W_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 9788	**Instance**	CM_CORE__L4PER
Description	This register manages the HDQ1W clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-925. Register Call Summary for Register CM_L4PER_HDQ1W_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-926. CM_L4PER2_PWMSS2_CLKCTRL

Address Offset	0x0000 0090		
Physical Address	0x4A00 9790	**Instance**	CM_CORE__L4PER
Description	This register manages the PWMSS1 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-927. Register Call Summary for Register CM_L4PER2_PWMSS2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-928. CM_L4PER2_PWMSS3_CLKCTRL

Address Offset	0x0000 0098		
Physical Address	0x4A00 9798	**Instance**	CM_CORE__L4PER
Description	This register manages the PWMSS2 clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-929. Register Call Summary for Register CM_L4PER2_PWMSS3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-930. CM_L4PER_I2C1_CLKCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4A00 97A0	**Instance**	CM_CORE__L4PER
Description	This register manages the I2C1 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-931. Register Call Summary for Register CM_L4PER_I2C1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-932. CM_L4PER_I2C2_CLKCTRL

Address Offset	0x0000 00A8		
Physical Address	0x4A00 97A8	**Instance**	CM_CORE__L4PER
Description	This register manages the I2C2 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-933. Register Call Summary for Register CM_L4PER_I2C2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-934. CM_L4PER_I2C3_CLKCTRL

Address Offset	0x0000 00B0		
Physical Address	0x4A00 97B0	**Instance**	CM_CORE__L4PER
Description	This register manages the I2C3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-935. Register Call Summary for Register CM_L4PER_I2C3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-936. CM_L4PER_I2C4_CLKCTRL

Address Offset	0x0000 00B8		
Physical Address	0x4A00 97B8	**Instance**	CM_CORE__L4PER
Description	This register manages the I2C4 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-937. Register Call Summary for Register CM_L4PER_I2C4_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-938. CM_L4PER_L4_PER1_CLKCTRL

Address Offset	0x0000 00C0		
Physical Address	0x4A00 97C0	**Instance**	CM_CORE__L4PER
Description	This register manages the L4_PER1 clocks.		
Type	R		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-939. Register Call Summary for Register CM_L4PER_L4_PER1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

Table 3-939. Register Call Summary for Register CM_L4PER_L4_PER1_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-940. CM_L4PER2_PWMSS1_CLKCTRL

Address Offset	0x0000 00C4		
Physical Address	0x4A00 97C4	**Instance**	CM_CORE__L4PER
Description	This register manages the PWMSS0 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-941. Register Call Summary for Register CM_L4PER2_PWMSS1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-942. CM_L4PER3_TIMER13_CLKCTRL

Address Offset	0x0000 00C8		
Physical Address	0x4A00 97C8	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER13 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-943. Register Call Summary for Register CM_L4PER3_TIMER13_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual
- CM_CORE__L4PER Register Summary: [3]

Table 3-944. CM_L4PER3_TIMER14_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4A00 97D0	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER14 clocks.		

Table 3-944. CM_L4PER3_TIMER14_CLKCTRL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-945. Register Call Summary for Register CM_L4PER3_TIMER14_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-946. CM_L4PER3_TIMER15_CLKCTRL

Address Offset	0x0000 00D8		
Physical Address	0x4A00 97D8	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER15 clocks.		
Type	RW		

31:28	27:24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-947. Register Call Summary for Register CM_L4PER3_TIMER15_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

Table 3-947. Register Call Summary for Register CM_L4PER3_TIMER15_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-948. CM_L4PER_MCSPI1_CLKCTRL

Address Offset	0x0000 00F0		
Physical Address	0x4A00 97F0	**Instance**	CM_CORE__L4PER
Description	This register manages the McSPI1 clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-949. Register Call Summary for Register CM_L4PER_MCSPI1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-950. CM_L4PER_MCSPI2_CLKCTRL

Address Offset	0x0000 00F8		
Physical Address	0x4A00 97F8	**Instance**	CM_CORE__L4PER
Description	This register manages the McSPI2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-951. Register Call Summary for Register CM_L4PER_MCSPI2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-952. CM_L4PER_MCSPI3_CLKCTRL

Address Offset	0x0000 0100		
Physical Address	0x4A00 9800	**Instance**	CM_CORE__L4PER
Description	This register manages the McSPI3 clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-953. Register Call Summary for Register CM_L4PER_MCSPI3_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-954. CM_L4PER_MCSPI4_CLKCTRL

Address Offset	0x0000 0108		
Physical Address	0x4A00 9808	**Instance**	CM_CORE__L4PER
Description	This register manages the McSPI4 clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-955. Register Call Summary for Register CM_L4PER_MCSPI4_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual
- CM_CORE__L4PER Register Summary: [2]

Table 3-956. CM_L4PER_GPIO7_CLKCTRL

Address Offset	0x0000 0110		
Physical Address	0x4A00 9810	**Instance**	CM_CORE__L4PER
Description	This register manages the GPIO7 clocks.		
Type	RW		

31:18	17:16	15:9	8	7:2	1:0
RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-957. Register Call Summary for Register CM_L4PER_GPIO7_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0] [1]

Table 3-957. Register Call Summary for Register CM_L4PER_GPIO7_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-958. CM_L4PER_GPIO8_CLKCTRL

Address Offset	0x0000 0118		
Physical Address	0x4A00 9818	**Instance**	CM_CORE__L4PER
Description	This register manages the GPIO8 clocks.		
Type	RW		

31:18	17:16	15:9	8	7:2	1:0
RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-959. Register Call Summary for Register CM_L4PER_GPIO8_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-960. CM_L4PER_MMC3_CLKCTRL

Address Offset	0x0000 0120		
Physical Address	0x4A00 9820	**Instance**	CM_CORE__L4PER
Description	This register manages the MMC3 clocks.		
Type	RW		

31:27	26:25	24	23:18	17:16	15:9	8	7:2	1:0
RESERVED	CLKSEL_DIV	CLKSEL_MUX	RESERVED	IDLEST	RESERVED	OPTFCLKEN_CLK32K	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	Selects the divider value 0x0: Select MMC CLK divided by 1 0x1: Select MMC CLK divided by 2 0x2: Selects MMC CLK divided by 4 0x3: RESERVED	RW	0x0
24	CLKSEL_MUX	Select the clock for the MMC from DPLL_PER. 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-961. Register Call Summary for Register CM_L4PER_MMC3_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-962. CM_L4PER_MMC4_CLKCTRL

Address Offset	0x0000 0128		
Physical Address	0x4A00 9828	**Instance**	CM_CORE__L4PER
Description	This register manages the MMC4 clocks.		
Type	RW		

31:27	26:25	24	23:18	17:16	15:9	8	7:2	1:0
RESERVED	CLKSEL_DIV	CLKSEL_MUX	RESERVED	IDLEST	RESERVED	OPTFCLKEN_CLK32K	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	Selects the divider value 0x0: Select MMC CLK divided by 1 0x1: Select MMC CLK divided by 2 0x2: Selects MMC CLK divided by 4 0x3: RESERVED	RW	0x0
24	CLKSEL_MUX	Select the clock for the MMC from DPLL_PER. 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-963. Register Call Summary for Register CM_L4PER_MMC4_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-964. CM_L4PER3_TIMER16_CLKCTRL

Address Offset	0x0000 0130		
Physical Address	0x4A00 9830	**Instance**	CM_CORE__L4PER
Description	This register manages the TIMER16 clocks.		
Type	RW		

31:28	27:24	23:18	17	16:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-965. Register Call Summary for Register CM_L4PER3_TIMER16_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-966. CM_L4PER2_QSPI_CLKCTRL

Address Offset	0x0000 0138		
Physical Address	0x4A00 9838	**Instance**	CM_CORE__L4PER
Description	This register manages the QSPI clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_DIV		CLKSEL_SOURCE	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	QSPI clock divide ratio. 0x0: QSPI clock is divided by 1. 0x1: QSPI clock is divided by 2. 0x2: QSPI clock is divided by 4. 0x3: RESERVED	RW	0x0

Bits	Field Name	Description	Type	Reset
24	CLKSEL_SOURCE	Selects the source of the functional clock. 0x0: 128MHz clock derived from DPLL_PER is selected 0x1: Selects PER_QSPI_CLK from DPLL_PER	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-967. Register Call Summary for Register CM_L4PER2_QSPI_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-968. CM_L4PER_UART1_CLKCTRL

Address Offset	0x0000 0140		
Physical Address	0x4A00 9840	**Instance**	CM_CORE__L4PER
Description	This register manages the UART1 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-969. Register Call Summary for Register CM_L4PER_UART1_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-970. CM_L4PER_UART2_CLKCTRL

Address Offset	0x0000 0148		
Physical Address	0x4A00 9848	**Instance**	CM_CORE__L4PER
Description	This register manages the UART2 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-971. Register Call Summary for Register CM_L4PER_UART2_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-972. CM_L4PER_UART3_CLKCTRL

Address Offset	0x0000 0150		
Physical Address	0x4A00 9850	**Instance**	CM_CORE__L4PER
Description	This register manages the UART3 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-973. Register Call Summary for Register CM_L4PER_UART3_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-974. CM_L4PER_UART4_CLKCTRL

Address Offset	0x0000 0158		
Physical Address	0x4A00 9858	**Instance**	CM_CORE__L4PER
Description	This register manages the UART4 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-975. Register Call Summary for Register CM_L4PER_UART4_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-976. CM_L4PER2_MCASP2_CLKCTRL

Address Offset	0x0000 0160		
Physical Address	0x4A00 9860	**Instance**	CM_CORE__L4PER
Description	This register manages the McASP2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKSEL_AHCLKR				CLKSEL_AHCLKX				CLKSEL_AUX_CLK		RESERVED				IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	CLKSEL_AHCLKR	Selects reference clock for AHCLKR 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0

Bits	Field Name	Description	Type	Reset
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-977. Register Call Summary for Register CM_L4PER2_MCASP2_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1] [2]
- Clock Domain Module Attributes: [3] [4]

Table 3-977. Register Call Summary for Register CM_L4PER2_MCASP2_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [5]

Table 3-978. CM_L4PER2_MCASP3_CLKCTRL

Address Offset	0x0000 0168		
Physical Address	0x4A00 9868	**Instance**	CM_CORE__L4PER
Description	This register manages the McASP3 clocks.		
Type	RW		

31:28	27:24	23:22	21:18	17:16	15:2	1:0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-979. Register Call Summary for Register CM_L4PER2_MCASP3_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-980. CM_L4PER_UART5_CLKCTRL

Address Offset	0x0000 0170		
Physical Address	0x4A00 9870	**Instance**	CM_CORE__L4PER
Description	This register manages the UART5 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-981. Register Call Summary for Register CM_L4PER_UART5_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-982. CM_L4PER2_MCASP5_CLKCTRL

Address Offset	0x0000 0178		
Physical Address	0x4A00 9878	**Instance**	CM_CORE__L4PER
Description	This register manages the McASP5 clocks.		
Type	RW		

31:28	27:24	23	22:19	18	17:2	1:0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0

Bits	Field Name	Description	Type	Reset
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-983. Register Call Summary for Register CM_L4PER2_MCASP5_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-984. CM_L4SEC_CLKSTCTRL

Address Offset	0x0000 0180		
Physical Address	0x4A00 9880	**Instance**	CM_CORE__L4PER
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLKACTIVITY_L4SEC_L3_GICLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_L4SEC_L3_GICLK	This field indicates the state of the L3_SECURE_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-985. Register Call Summary for Register CM_L4SEC_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-986. CM_L4SEC_STATICDEP

Address Offset	0x0000 0184		
Physical Address	0x4A00 9884	**Instance**	CM_CORE__L4PER
Description	This register controls the static domain depedencies from L4SEC domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31:14	13	12:6	5	4	3:0
RESERVED	L4PER_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3:0	RESERVED		R	0x0

Table 3-987. Register Call Summary for Register CM_L4SEC_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-988. CM_L4SEC_DYNAMICDEP

Address Offset	0x0000 0188		
Physical Address	0x4A00 9888	**Instance**	CM_CORE__L4PER
Description	This register controls the dynamic domain depedencies from L4SEC domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31:7	6	5	4:0
RESERVED	RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-989. Register Call Summary for Register CM_L4SEC_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [1]

Table 3-990. CM_L4PER2_MCASP8_CLKCTRL

Address Offset	0x0000 0190

Table 3-990. CM_L4PER2_MCASP8_CLKCTRL (continued)

Physical Address	0x4A00 9890	**Instance**	CM_CORE__L4PER
Description	This register manages the McASP8 clocks.		
Type	RW		

31:28	27:24	23:22	21:18	17:16	15:1	0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-991. Register Call Summary for Register CM_L4PER2_MCASP8_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-992. CM_L4PER2_MCASP4_CLKCTRL

Address Offset	0x0000 0198		
Physical Address	0x4A00 9898	**Instance**	CM_CORE__L4PER
Description	This register manages the McASP4 clocks.		
Type	RW		

31:28	27:24	23:22	21:18	17:16	15:2	1:0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0

Bits	Field Name	Description	Type	Reset
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-993. Register Call Summary for Register CM_L4PER2_MCASP4_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-994. CM_L4SEC_AES1_CLKCTRL

Address Offset	0x0000 01A0		
Physical Address	0x4A00 98A0	**Instance**	CM_CORE__L4PER
Description	This register manages the AES1 clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-995. Register Call Summary for Register CM_L4SEC_AES1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-996. CM_L4SEC_AES2_CLKCTRL

Address Offset	0x0000 01A8		
Physical Address	0x4A00 98A8	**Instance**	CM_CORE__L4PER
Description	This register manages the AES2 clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-997. Register Call Summary for Register CM_L4SEC_AES2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-998. CM_L4SEC_DES3DES_CLKCTRL

Address Offset	0x0000 01B0		
Physical Address	0x4A00 98B0	**Instance**	CM_CORE__L4PER
Description	This register manages the DES3DES clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-999. Register Call Summary for Register CM_L4SEC_DES3DES_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-1000. CM_L4SEC_FPKA_CLKCTRL

Address Offset	0x0000 01B8		
Physical Address	0x4A00 98B8	**Instance**	CM_CORE__L4PER
Description	This register manages the FPKA clocks.		
Type	RW		

31–18	17–16	15–2	1–0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1001. Register Call Summary for Register CM_L4SEC_FPKA_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-1002. CM_L4SEC_RNG_CLKCTRL

Address Offset	0x0000 01C0		
Physical Address	0x4A00 98C0	**Instance**	CM_CORE__L4PER
Description	This register manages the RNG clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-1003. Register Call Summary for Register CM_L4SEC_RNG_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-1004. CM_L4SEC_SHA2MD51_CLKCTRL

Address Offset	0x0000 01C8		
Physical Address	0x4A00 98C8	**Instance**	CM_CORE__L4PER
Description	This register manages the SHA2MD51 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-1005. Register Call Summary for Register CM_L4SEC_SHA2MD51_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-1006. CM_L4PER2_UART7_CLKCTRL

Address Offset	0x0000 01D0		
Physical Address	0x4A00 98D0	**Instance**	CM_CORE__L4PER
Description	This register manages the UART7 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1007. Register Call Summary for Register CM_L4PER2_UART7_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-1008. CM_L4SEC_DMA_CRYPTO_CLKCTRL

Address Offset	0x0000 01D8		
Physical Address	0x4A00 98D8	**Instance**	CM_CORE__L4PER
Description	This register manages the DMA_CRYPTO clocks.		
Type	R		

31:19	18	17:16	15:2	1:0
RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1009. Register Call Summary for Register CM_L4SEC_DMA_CRYPTO_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-1010. CM_L4PER2_UART8_CLKCTRL

Address Offset	0x0000 01E0		
Physical Address	0x4A00 98E0	**Instance**	CM_CORE__L4PER
Description	This register manages the UART8 clocks.		
Type	RW		

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1011. Register Call Summary for Register CM_L4PER2_UART8_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-1012. CM_L4PER2_UART9_CLKCTRL

Address Offset	0x0000 01E8		
Physical Address	0x4A00 98E8	**Instance**	CM_CORE__L4PER
Description	This register manages the UART9 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1013. Register Call Summary for Register CM_L4PER2_UART9_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [3]

Table 3-1014. CM_L4PER2_DCAN2_CLKCTRL

Address Offset	0x0000 01F0		
Physical Address	0x4A00 98F0	**Instance**	CM_CORE__L4PER
Description	This register manages the DCAN2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1015. Register Call Summary for Register CM_L4PER2_DCAN2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-1016. CM_L4SEC_SHA2MD52_CLKCTRL

Address Offset	0x0000 01F8		
Physical Address	0x4A00 98F8	**Instance**	CM_CORE__L4PER
Description	This register manages the SHA2MD52 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-1017. Register Call Summary for Register CM_L4SEC_SHA2MD52_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [2]

Table 3-1018. CM_L4PER2_CLKSTCTRL

Address Offset	0x0000 01FC		
Physical Address	0x4A00 98FC	**Instance**	CM_CORE__L4PER
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLKACTIVITY_MCASP8_AUX_GFCLK	CLKACTIVITY_MCASP8_AHCLKX	CLKACTIVITY_MCASP7_AUX_GFCLK	CLKACTIVITY_MCASP7_AHCLKX	CLKACTIVITY_MCASP6_AUX_GFCLK	CLKACTIVITY_MCASP6_AHCLKX	CLKACTIVITY_MCASP5_AHCLKX	CLKACTIVITY_MCASP5_AUX_GFCLK	CLKACTIVITY_MCASP4_AUX_GFCLK	CLKACTIVITY_MCASP4_AHCLKX	CLKACTIVITY_MCASP3_AUX_GFCLK	CLKACTIVITY_MCASP3_AHCLKX	CLKACTIVITY_MCASP2_AUX_GFCLK	CLKACTIVITY_MCASP2_AHCLKR	CLKACTIVITY_MCASP2_AHCLKX	CLKACTIVITY_L4PER2_L3_GICLK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKACTIVITY_DCAN2_SYS_CLK	CLKACTIVITY_ICSS_IEP_CLK	CLKACTIVITY_PER_192M_GFCLK	CLKACTIVITY_QSPI_GFCLK	CLKACTIVITY_UART9_GFCLK	CLKACTIVITY_UART8_GFCLK	CLKACTIVITY_UART7_GFCLK	CLKACTIVITY_ICSS_CLK	RESERVED	RESERVED				RESERVED	CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31	CLKACTIVITY_MCASP8_AUX_GFCLK	This field indicates the state of the MCASP8_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
30	CLKACTIVITY_MCASP8_AHCLKX	This field indicates the state of the MCASP8_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
29	CLKACTIVITY_MCASP7_AUX_GFCLK	This field indicates the state of the MCASP7_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
28	CLKACTIVITY_MCASP7_AHCLKX	This field indicates the state of the MCASP7_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
27	CLKACTIVITY_MCASP6_AUX_GFCLK	This field indicates the state of the MCASP6_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
26	CLKACTIVITY_MCASP6_AHCLKX	This field indicates the state of the MCASP6_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
25	CLKACTIVITY_MCASP5_AHCLKX	This field indicates the state of the MCASP5_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
24	CLKACTIVITY_MCASP5_AUX_GFCLK	This field indicates the state of the MCASP5_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
23	CLKACTIVITY_MCASP4_AUX_GFCLK	This field indicates the state of the MCASP4_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
22	CLKACTIVITY_MCASP4_AHCLKX	This field indicates the state of the MCASP4_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
21	CLKACTIVITY_MCASP3_AUX_GFCLK	This field indicates the state of the MCASP3_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
20	CLKACTIVITY_MCASP3_AHCLKX	This field indicates the state of the MCASP3_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
19	CLKACTIVITY_MCASP2_AUX_GFCLK	This field indicates the state of the MCASP2_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
18	CLKACTIVITY_MCASP2_AHCLKR	This field indicates the state of the MCASP2_AHCLKR clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_MCASP2_AHCLKX	This field indicates the state of the MCASP2_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_L4PER2_L3_GICLK	This field indicates the state of the L4PER2_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_DCAN2_SYS_CLK	This field indicates the state of the DCAN2_SYS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_ICSS_IEP_CLK	This field indicates the state of the ICSS_IEP_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_PER_192M_GFCLK	This field indicates the state of the PER_192M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_QSPI_GFCLK	This field indicates the state of the QSPI_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_UART9_GFCLK	This field indicates the state of the UART9_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_UART8_GFCLK	This field indicates the state of the UART8_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_UART7_GFCLK	This field indicates the state of the UART7_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_ICSS_CLK	This field indicates the state of the ICSS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7	RESERVED		R	0x0
6:3	RESERVED		R	0x0
2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-1019. Register Call Summary for Register CM_L4PER2_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [25]

Table 3-1020. CM_L4PER2_DYNAMICDEP

Address Offset	0x0000 0200		
Physical Address	0x4A00 9900	**Instance**	CM_CORE__L4PER
Description	This register controls the dynamic domain depedencies from L4PER2 domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31 30 29 28	27 26 25 24	23	22	21 20 19 18 17 16 15 14 13	12	11 10 9 8	7	6	5 4	3	2 1 0
RESERVED	WINDOWSIZE	RESERVED	GMAC_DYNDEP	RESERVED	L4CFG_DYNDEP	RESERVED	L3INIT_DYNDEP	ATL_DYNDEP	RESERVED	IPU_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23	RESERVED		R	0x0
22	GMAC_DYNDEP	Dynamic dependency towards GMAC clock domain 0x1: Dependency is enabled	R	0x1
21:13	RESERVED		R	0x0
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain 0x1: Dependency is enabled	R	0x1
11:8	RESERVED		R	0x0
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6	ATL_DYNDEP	Dynamic dependency towards ATL clock domain 0x1: Dependency is enabled	R	0x1
5:4	RESERVED		R	0x0
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2:0	RESERVED		R	0x0

Table 3-1021. Register Call Summary for Register CM_L4PER2_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [5]

Table 3-1022. CM_L4PER2_MCASP6_CLKCTRL

Address Offset	0x0000 0204		
Physical Address	0x4A00 9904	**Instance**	CM_CORE__L4PER
Description	This register manages the McASP6 clocks.		
Type	RW		

31:28	27:24	23	22:20	19:18	17:1	0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1023. Register Call Summary for Register CM_L4PER2_MCASP6_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1]
- Clock Domain Module Attributes: [2] [3]

Table 3-1023. Register Call Summary for Register CM_L4PER2_MCASP6_CLKCTRL (continued)

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-1024. CM_L4PER2_MCASP7_CLKCTRL

Address Offset	0x0000 0208		
Physical Address	0x4A00 9908	**Instance**	CM_CORE__L4PER
Description	This register manages the McASP7 clocks.		
Type	RW		

31:28	27:24	23:22	21:18	17:16	15:2	1:0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1025. Register Call Summary for Register CM_L4PER2_MCASP7_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_MCASP Overview: [0] [1]
- Clock Domain Module Attributes: [2] [3]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [4]

Table 3-1026. CM_L4PER2_STATICDEP

Address Offset	0x0000 020C		
Physical Address	0x4A00 990C	**Instance**	CM_CORE__L4PER
Description	This register controls the static domain depedencies from L4PER2 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31:24	23	22:19	18	17:6	5	4:2	1	0
RESERVED	IPU1_STATDEP	RESERVED	DSP2_STATDEP	RESERVED	L3MAIN1_STATDEP	RESERVED	DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22:19	RESERVED		R	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:2	RESERVED		R	0x0
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1027. Register Call Summary for Register CM_L4PER2_STATICDEP

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [0]

Table 3-1028. CM_L4PER3_CLKSTCTRL

Address Offset	0x0000 0210		
Physical Address	0x4A00 9910	**Instance**	CM_CORE__L4PER
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31:13	12	11	10	9	8	7:6	5:2	1:0
RESERVED	CLKACTIVITY_TIMER16_GFCLK	CLKACTIVITY_TIMER15_GFCLK	CLKACTIVITY_TIMER14_GFCLK	CLKACTIVITY_TIMER13_GFCLK	CLKACTIVITY_L4PER3_L3_GICLK	RESERVED	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	CLKACTIVITY_TIMER16_GFCLK	This field indicates the state of the DMT16_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_TIMER15_GFCLK	This field indicates the state of the DMT15_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_TIMER14_GFCLK	This field indicates the state of the DMT14_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_TIMER13_GFCLK	This field indicates the state of the DMT13_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
8	CLKACTIVITY_L4PER3_L3_GICLK	This field indicates the state of the L4PER2_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:6	RESERVED		R	0x0
5:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-1029. Register Call Summary for Register CM_L4PER3_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [5]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [6]

Table 3-1030. CM_L4PER3_DYNAMICDEP

Address Offset	0x0000 0214		
Physical Address	0x4A00 9914	**Instance**	CM_CORE__L4PER
Description	This register controls the dynamic domain depedencies from L4PER3 domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>VPE_DYNDEP</td><td colspan="3">RESERVED</td><td colspan="4">WINDOWSIZE</td><td>RTC_DYNDEP</td><td colspan="10">RESERVED</td><td>L4CFG_DYNDEP</td><td colspan="2">RESERVED</td><td>CAM_DYNDEP</td><td>RESERVED</td><td>L3INIT_DYNDEP</td><td>RESERVED</td><td>L3MAIN1_DYNDEP</td><td>RESERVED</td><td>IPU_DYNDEP</td><td colspan="3">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	VPE_DYNDEP	Dynamic dependency towards VPE clock domain 0x1: Dependency is enabled	R	0x1
30:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23	RTC_DYNDEP	Dynamic dependency towards RTC clock domain 0x1: Dependency is enabled	R	0x1
22:13	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain 0x1: Dependency is enabled	R	0x1
11:10	RESERVED		R	0x0
9	CAM_DYNDEP	Dynamic dependency towards CAM clock domain 0x1: Dependency is enabled	R	0x1
8	RESERVED		R	0x0
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	RESERVED		R	0x0
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2:0	RESERVED		R	0x0

Table 3-1031. Register Call Summary for Register CM_L4PER3_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- CM_CORE__L4PER Register Summary: [7]

3.13.24 CM_CORE__OCP_SOCKET Registers

3.13.24.1 CM_CORE__OCP_SOCKET Register Summary

Table 3-1032. CM_CORE__OCP_SOCKET Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__OCP_SOCKET Physical Address L4_CFG Interconnect
REVISION_CM_CORE	R	32	0x0000 0000	0x4A00 8000
CM_CM_CORE_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4A00 8040
CM_CORE_DEBUG_CFG	RW	32	0x0000 00F0	0x4A00 80F0

3.13.24.2 CM_CORE__OCP_SOCKET Register Description

Table 3-1033. REVISION_CM_CORE

Address Offset	0x0000 0000		
Physical Address	0x4A00 8000	**Instance**	CM_CORE__OCP_SOCKET
Description	This register contains the IP revision code for the CM_CORE part of the PRCM		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision number	R	0x-[(1)]

(1) TI internal data

Table 3-1034. Register Call Summary for Register REVISION_CM_CORE

PRCM Register Manual

- CM_CORE__OCP_SOCKET Register Summary: [0]

Table 3-1035. CM_CM_CORE_PROFILING_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 8040	**Instance**	CM_CORE__OCP_SOCKET
Description	This register manages the CM_CORE_PROFILING clocks. [warm reset insensitive]		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. OCP configuration port is not accessible. 0x1: Module is managed automatically by HW along with L3INSTR domain. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-1036. Register Call Summary for Register CM_CM_CORE_PROFILING_CLKCTRL

PRCM Register Manual

- CM_CORE__OCP_SOCKET Register Summary: [0]
- CM_CORE__RESTORE Register Description: [1] [2]

Table 3-1037. CM_CORE_DEBUG_CFG

Address Offset	0x0000 00F0		
Physical Address	0x4A00 80F0	**Instance**	CM_CORE__OCP_SOCKET
Description	This register is used to configure the CM_CORE's 32-bit debug output. There is one 8-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31:24	23:16	15:8	7:0
SEL3	SEL2	SEL1	SEL0

Bits	Field Name	Description	Type	Reset
31:24	SEL3	Internal signal block select for debug word byte-3	RW	0x3
23:16	SEL2	Internal signal block select for debug word byte-2	RW	0x2
15:8	SEL1	Internal signal block select for debug word byte-1	RW	0x1

Bits	Field Name	Description	Type	Reset
7:0	SEL0	Internal signal block select for debug word byte-0	RW	0x0

Table 3-1038. Register Call Summary for Register CM_CORE_DEBUG_CFG

PRCM Register Manual

- CM_CORE__OCP_SOCKET Register Summary: [0]

3.13.25 CM_CORE__RESTORE Registers

3.13.25.1 CM_CORE__RESTORE Register Summary

Table 3-1039. CM_CORE__RESTORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CM_CORE__RESTORE Physical Address L4_CFG Interconnect
CM_L3MAIN1_CLKSTCTRL_RESTORE	RW	32	0x0000 0018	0x4A00 9E18
CM_L4CFG_CLKSTCTRL_RESTORE	RW	32	0x0000 0020	0x4A00 9E20
CM_L4PER_CLKSTCTRL_RESTORE	RW	32	0x0000 0028	0x4A00 9E28
CM_L3INIT_CLKSTCTRL_RESTORE	RW	32	0x0000 002C	0x4A00 9E2C
CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE	RW	32	0x0000 0030	0x4A00 9E30
CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE	RW	32	0x0000 0034	0x4A00 9E34
CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE	RW	32	0x0000 0038	0x4A00 9E38
CM_CM_CORE_PROFILING_CLKCTRL_RESTORE	RW	32	0x0000 003C	0x4A00 9E3C
CM_L3MAIN1_DYNAMICDEP_RESTORE	RW	32	0x0000 0048	0x4A00 9E48
CM_L4CFG_DYNAMICDEP_RESTORE	RW	32	0x0000 0058	0x4A00 9E58
CM_L4PER_DYNAMICDEP_RESTORE	RW	32	0x0000 005C	0x4A00 9E5C
RESERVED	R	32	0x0000 0060	0x4A00 9E60
CM_DMA_STATICDEP_RESTORE	RW	32	0x0000 006C	0x4A00 9E6C

3.13.25.2 CM_CORE__RESTORE Register Description

Table 3-1040. CM_L3MAIN1_CLKSTCTRL_RESTORE

Address Offset	0x0000 0018		
Physical Address	0x4A00 9E18	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L3MAIN1_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3MAIN1_CLKSTCTRL register.	RW	0x0

Table 3-1041. Register Call Summary for Register CM_L3MAIN1_CLKSTCTRL_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1042. CM_L4CFG_CLKSTCTRL_RESTORE

Address Offset	0x0000 0020		
Physical Address	0x4A00 9E20	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L4CFG_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4CFG_CLKSTCTRLregister.	RW	0x0

Table 3-1043. Register Call Summary for Register CM_L4CFG_CLKSTCTRL_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1044. CM_L4PER_CLKSTCTRL_RESTORE

Address Offset	0x0000 0028		
Physical Address	0x4A00 9E28	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L4PER_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4PER_CLKSTCTRL register.	RW	0x0

Table 3-1045. Register Call Summary for Register CM_L4PER_CLKSTCTRL_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1046. CM_L3INIT_CLKSTCTRL_RESTORE

Address Offset	0x0000 002C		
Physical Address	0x4A00 9E2C	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L3INIT_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INIT_CLKSTCTRL register.	RW	0x0

Table 3-1047. Register Call Summary for Register CM_L3INIT_CLKSTCTRL_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1048. CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE

Address Offset	0x0000 0030		
Physical Address	0x4A00 9E30	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L3INSTR_L3_MAIN_2_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESTORE

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INSTR_L3_MAIN_3_CLKCTRL register.	RW	0x30001

Table 3-1049. Register Call Summary for Register CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [2]

Table 3-1050. CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE

Address Offset	0x0000 0034		
Physical Address	0x4A00 9E34	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L3INSTR_L3_INSTR_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INSTR_L3_INSTR_CLKCTRL register.	RW	0x30001

Table 3-1051. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [2]

Table 3-1052. CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE

Address Offset	0x0000 0038		
Physical Address	0x4A00 9E38	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L3INSTR_OCP_WP_NOC_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INSTR_OCP_WP_NOC_CLKCTRL register.	RW	0x30001

Table 3-1053. Register Call Summary for Register CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [2]

Table 3-1054. CM_CM_CORE_PROFILING_CLKCTRL_RESTORE

Address Offset	0x0000 003C

Table 3-1054. CM_CM_CORE_PROFILING_CLKCTRL_RESTORE (continued)

Physical Address	0x4A00 9E3C	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_CM_CORE_PROFILING_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31–0
RESTORE

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CM_CORE_PROFILING_CLKCTRL register.	RW	0x30001

Table 3-1055. Register Call Summary for Register CM_CM_CORE_PROFILING_CLKCTRL_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1056. CM_L3MAIN1_DYNAMICDEP_RESTORE

Address Offset	0x0000 0048		
Physical Address	0x4A00 9E48	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L3MAIN1_DYNAMICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31–0
RESTORE

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3MAIN1_DYNAMICDEP register.	RW	0x4001058

Table 3-1057. Register Call Summary for Register CM_L3MAIN1_DYNAMICDEP_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1058. CM_L4CFG_DYNAMICDEP_RESTORE

Address Offset	0x0000 0058		
Physical Address	0x4A00 9E58	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L4CFG_DYNAMICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31–0
RESTORE

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4CFG_DYNAMICDEP register.	RW	0x40789f2

Table 3-1059. Register Call Summary for Register CM_L4CFG_DYNAMICDEP_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1060. CM_L4PER_DYNAMICDEP_RESTORE

Address Offset	0x0000 005C

Table 3-1060. CM_L4PER_DYNAMICDEP_RESTORE (continued)

Physical Address	0x4A00 9E5C	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_L4PER_DYNAMICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31–0
RESTORE

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4PER_DYNAMICDEP register.	RW	0x4004180

Table 3-1061. Register Call Summary for Register CM_L4PER_DYNAMICDEP_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

Table 3-1062. CM_DMA_STATICDEP_RESTORE

Address Offset	0x0000 006C		
Physical Address	0x4A00 9E6C	**Instance**	CM_CORE__RESTORE
Description	Second address map for register CM_DMA_STATICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31–0
RESTORE

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DMA_STATICDEP register.	RW	0xb0f0

Table 3-1063. Register Call Summary for Register CM_DMA_STATICDEP_RESTORE

PRCM Register Manual

- CM_CORE__RESTORE Register Summary: [0]

3.13.26 CAM_PRM Registers

3.13.26.1 CAM_PRM Register Summary

Table 3-1064. CAM_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CAM_PRM Physical Address L4_WKUP Interconnect
PM_CAM_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7000
PM_CAM_PWRSTST	RW	32	0x0000 0004	0x4AE0 7004
PM_CAM_VIP1_WKDEP	RW	32	0x0000 0020	0x4AE0 7020
RM_CAM_VIP1_CONTEXT	RW	32	0x0000 0024	0x4AE0 7024
PM_CAM_VIP2_WKDEP	RW	32	0x0000 0028	0x4AE0 7028
RM_CAM_VIP2_CONTEXT	RW	32	0x0000 002C	0x4AE0 702C
PM_CAM_VIP3_WKDEP	RW	32	0x0000 0030	0x4AE0 7030
RM_CAM_VIP3_CONTEXT	RW	32	0x0000 0034	0x4AE0 7034
RESERVED	R	32	0x0000 003C	0x4AE0 703C
RESERVED	R	32	0x0000 0044	0x4AE0 7044
RESERVED	R	32	0x0000 004C	0x4AE0 704C

3.13.26.2 CAM_PRM Register Description

Table 3-1065. PM_CAM_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7000	**Instance**	CAM_PRM
Description	This register controls the CAM power state to reach upon a domain sleep transition		
Type	RW		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	VIP_BANK_ONSTATE	VIP_BANK memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x0

Table 3-1066. PM_CAM_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7004	**Instance**	CAM_PRM
Description	This register provides a status on the current CAM power domain state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	VIP_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	VIP_BANK_STATEST	VIP_BANK memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1067. PM_CAM_VIP1_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7020	**Instance**	CAM_PRM
Description	This register controls wakeup dependency based on VIP1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_VIP1_EVE2	WKUPDEP_VIP1_EVE1	WKUPDEP_VIP1_DSP2	WKUPDEP_VIP1_IPU1	RESERVED	WKUPDEP_VIP1_DSP1	WKUPDEP_VIP1_IPU2	WKUPDEP_VIP1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_VIP1_EVE2	Wakeup dependency from VIP1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_VIP1_EVE1	Wakeup dependency from VIP1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_VIP1_DSP2	Wakeup dependency from VIP1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_VIP1_IPU1	Wakeup dependency from VIP1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_VIP1_DSP1	Wakeup dependency from VIP1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_VIP1_IPU2	Wakeup dependency from vip1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_VIP1_MPU	Wakeup dependency from VIP1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1068. RM_CAM_VIP1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7024	**Instance**	CAM_PRM
Description	This register contains dedicated VIP1 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_VIP_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VIP_BANK	Specify if memory-based context in VIP_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1069. PM_CAM_VIP2_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7028	**Instance**	CAM_PRM
Description	This register controls wakeup dependency based on VIP2 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_VIP2_EVE2	WKUPDEP_VIP2_EVE1	WKUPDEP_VIP2_DSP2	WKUPDEP_VIP2_IPU1	RESERVED	WKUPDEP_VIP2_DSP1	WKUPDEP_VIP2_IPU2	WKUPDEP_VIP2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_VIP2_EVE2	Wakeup dependency from VIP2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_VIP2_EVE1	Wakeup dependency from VIP2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_VIP2_DSP2	Wakeup dependency from VIP1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_VIP2_IPU1	Wakeup dependency from VIP2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_VIP2_DSP1	Wakeup dependency from VIP2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_VIP2_IPU2	Wakeup dependency from VIP2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_VIP2_MPU	Wakeup dependency from VIP2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1070. RM_CAM_VIP2_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 702C	**Instance**	CAM_PRM
Description	This register contains dedicated VIP2 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_VIP_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VIP_BANK	Specify if memory-based context in VIP_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1071. PM_CAM_VIP3_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7030	**Instance**	CAM_PRM
Description	This register controls wakeup dependency based on VIP3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	WKUPDEP_VIP3_EVE2	WKUPDEP_VIP3_EVE1	WKUPDEP_VIP3_DSP2	WKUPDEP_VIP3_IPU1	RESERVED	WKUPDEP_VIP3_DSP1	WKUPDEP_VIP3_IPU2	WKUPDEP_VIP3_MPU

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	WKUPDEP_VIP3_EVE2	Wakeup dependency from VIP3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_VIP3_EVE1	Wakeup dependency from VIP3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_VIP3_DSP2	Wakeup dependency from VIP3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_VIP3_IPU1	Wakeup dependency from VIP3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_VIP3_DSP1	Wakeup dependency from VIP3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_VIP3_IPU2	Wakeup dependency from vip3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_VIP3_MPU	Wakeup dependency from VIP3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1072. RM_CAM_VIP3_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7034	**Instance**	CAM_PRM
Description	This register contains dedicated VIP3 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_VIP_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VIP_BANK	Specify if memory-based context in VIP_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.13.27 CKGEN_PRM Registers

3.13.27.1 CKGEN_PRM Register Summary

Table 3-1073. CKGEN_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_PRM Physical Address L4_WKUP Interconnect
CM_CLKSEL_SYSCLK1	RW	32	0x0000 0000	0x4AE0 6100
CM_CLKSEL_WKUPAON	RW	32	0x0000 0008	0x4AE0 6108
CM_CLKSEL_ABE_PLL_REF	RW	32	0x0000 000C	0x4AE0 610C
CM_CLKSEL_SYS	RW	32	0x0000 0010	0x4AE0 6110
CM_CLKSEL_ABE_PLL_BYPAS	RW	32	0x0000 0014	0x4AE0 6114
CM_CLKSEL_ABE_PLL_SYS	RW	32	0x0000 0018	0x4AE0 6118
CM_CLKSEL_ABE_24M	RW	32	0x0000 001C	0x4AE0 611C
CM_CLKSEL_ABE_SYS	RW	32	0x0000 0020	0x4AE0 6120
CM_CLKSEL_HDMI_MCASP_AUX	RW	32	0x0000 0024	0x4AE0 6124
CM_CLKSEL_HDMI_TIMER	RW	32	0x0000 0028	0x4AE0 6128
CM_CLKSEL_MCASP_SYS	RW	32	0x0000 002C	0x4AE0 612C
CM_CLKSEL_MLBP_MCASP	RW	32	0x0000 0030	0x4AE0 6130
CM_CLKSEL_MLB_MCASP	RW	32	0x0000 0034	0x4AE0 6134
CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX	RW	32	0x0000 0038	0x4AE0 6138
RESERVED	RW	32	0x0000 0040	0x4AE0 6140
CM_CLKSEL_TIMER_SYS	RW	32	0x0000 0044	0x4AE0 6144
CM_CLKSEL_VIDEO1_MCASP_AUX	RW	32	0x0000 0048	0x4AE0 6148
CM_CLKSEL_VIDEO1_TIMER	RW	32	0x0000 004C	0x4AE0 614C
CM_CLKSEL_VIDEO2_MCASP_AUX	RW	32	0x0000 0050	0x4AE0 6150
CM_CLKSEL_VIDEO2_TIMER	RW	32	0x0000 0054	0x4AE0 6154
CM_CLKSEL_CLKOUTMUX0	RW	32	0x0000 0058	0x4AE0 6158
CM_CLKSEL_CLKOUTMUX1	RW	32	0x0000 005C	0x4AE0 615C
CM_CLKSEL_CLKOUTMUX2	RW	32	0x0000 0060	0x4AE0 6160

Table 3-1073. CKGEN_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_PRM Physical Address L4_WKUP Interconnect
CM_CLKSEL_HDMI_PLL_SYS	RW	32	0x0000 0064	0x4AE0 6164
CM_CLKSEL_VIDEO1_PLL_SYS	RW	32	0x0000 0068	0x4AE0 6168
CM_CLKSEL_VIDEO2_PLL_SYS	RW	32	0x0000 006C	0x4AE0 616C
CM_CLKSEL_ABE_CLK_DIV	RW	32	0x0000 0070	0x4AE0 6170
CM_CLKSEL_ABE_GICLK_DIV	RW	32	0x0000 0074	0x4AE0 6174
CM_CLKSEL_AESS_FCLK_DIV	RW	32	0x0000 0078	0x4AE0 6178
CM_CLKSEL_EVE_CLK	RW	32	0x0000 0080	0x4AE0 6180
CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX	RW	32	0x0000 0084	0x4AE0 6184
CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX	RW	32	0x0000 0088	0x4AE0 6188
CM_CLKSEL_DSP_GFCLK_CLKOUTMUX	RW	32	0x0000 008C	0x4AE0 618C
CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX	RW	32	0x0000 0090	0x4AE0 6190
CM_CLKSEL_EMU_CLK_CLKOUTMUX	RW	32	0x0000 0094	0x4AE0 6194
CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX	RW	32	0x0000 0098	0x4AE0 6198
CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX	RW	32	0x0000 009C	0x4AE0 619C
CM_CLKSEL_GPU_GCLK_CLKOUTMUX	RW	32	0x0000 00A0	0x4AE0 61A0
CM_CLKSEL_HDMI_CLK_CLKOUTMUX	RW	32	0x0000 00A4	0x4AE0 61A4
CM_CLKSEL_IVA_GCLK_CLKOUTMUX	RW	32	0x0000 00A8	0x4AE0 61A8
CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX	RW	32	0x0000 00AC	0x4AE0 61AC
CM_CLKSEL_MPU_GCLK_CLKOUTMUX	RW	32	0x0000 00B0	0x4AE0 61B0
CM_CLKSEL_PCIE1_CLK_CLKOUTMUX	RW	32	0x0000 00B4	0x4AE0 61B4
CM_CLKSEL_PCIE2_CLK_CLKOUTMUX	RW	32	0x0000 00B8	0x4AE0 61B8
CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX	RW	32	0x0000 00BC	0x4AE0 61BC
CM_CLKSEL_SATA_CLK_CLKOUTMUX	RW	32	0x0000 00C0	0x4AE0 61C0
CM_CLKSEL_OSC_32K_CLK_CLKOUTMUX	RW	32	0x0000 00C4	0x4AE0 61C4
CM_CLKSEL_SYS_CLK1_CLKOUTMUX	RW	32	0x0000 00C8	0x4AE0 61C8
CM_CLKSEL_SYS_CLK2_CLKOUTMUX	RW	32	0x0000 00CC	0x4AE0 61CC
CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX	RW	32	0x0000 00D0	0x4AE0 61D0
CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX	RW	32	0x0000 00D4	0x4AE0 61D4
CM_CLKSEL_ABE_LP_CLK	RW	32	0x0000 00D8	0x4AE0 61D8
RESERVED	R	32	0x0000 00DC	0x4AE0 61DC
CM_CLKSEL_EVE_GFCLK_CLKOUTMUX	RW	32	0x0000 00E0	0x4AE0 61E0

3.13.27.2 CKGEN_PRM Register Description

Table 3-1074. CM_CLKSEL_SYSCLK1

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6100	**Instance**	CKGEN_PRM
Description	Select the SYS CLK for SYSCLK1_32K_CLK. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 6 0x1: Select SYS_CLK divided by 10	RW	0x0

Table 3-1075. CM_CLKSEL_WKUPAON

Address Offset	0x0000 0008		
Physical Address	0x4AE0 6108	**Instance**	CKGEN_PRM
Description	Control the functional clock source of WKUPAON, PRM and Smart Reflex functional clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the clock source for WKUPAON_ICLK clock 0x0: Selects SYS_CLK for WKUPAON_ICLK 0x1: Selects ABE_LP_CLK for WKUPAON_ICLK	RW	0x0

Table 3-1076. CM_CLKSEL_ABE_PLL_REF

Address Offset	0x0000 000C		
Physical Address	0x4AE0 610C	**Instance**	CKGEN_PRM
Description	Control the source of the reference clock for DPLL_ABE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the source for the DPLL_ABE reference clock. 0x0: Selects ABE_DPLL_SYS_CLK for ABE_DPLL_CLK 0x1: Selects FUNC_32K_CLK for ABE_DPLL_CLK	RW	0x0

Table 3-1077. CM_CLKSEL_SYS

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6110	**Instance**	CKGEN_PRM
Description	ROM code sets the SYS_CLK configuration corresponding to the frequency of SYS_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													SYS_CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	SYS_CLKSEL	System clock input selection. 0x0: Uninitialized 0x1: Reserved 0x2: Input clock is 20 MHz 0x3: Reserved 0x4: Input clock is 19.2 MHz 0x5: Reserved 0x6: Input clock is 27 MHz 0x7: Reserved	RW	0x0

Table 3-1078. CM_CLKSEL_ABE_PLL_BYPAS

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6114	**Instance**	CKGEN_PRM
Description	Control the source of the bypass clock for DPLL_ABE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Control the source of the bypass clock for DPLL_ABE 0x0: Selects ABE_DPLL_SYS_CLK for ABE_DPLL_BYPASS_CLK 0x1: Selects FUNC_32K_CLK for ABE_DPLL_BYPASS_CLK	RW	0x0

Table 3-1079. CM_CLKSEL_ABE_PLL_SYS

Address Offset	0x0000 0018		
Physical Address	0x4AE0 6118	**Instance**	CKGEN_PRM
Description	Control the source of the SYS clock for DPLL_ABE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_ABE reference and bypass clock. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1080. CM_CLKSEL_ABE_24M

Address Offset	0x0000 001C		
Physical Address	0x4AE0 611C	**Instance**	CKGEN_PRM
Description	Select the ABE_24M_FCLK for TIMERS subsystems. [warm reset insensitive]		

Table 3-1080. CM_CLKSEL_ABE_24M (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 8 0x1: Select SYS_CLK divided by 16	RW	0x0

Table 3-1081. CM_CLKSEL_ABE_SYS

Address Offset	0x0000 0020		
Physical Address	0x4AE0 6120	**Instance**	CKGEN_PRM
Description	Select the SYS CLK for IPU subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 1 0x1: Select SYS_CLK divided by 2 Must be used for SYS_CLK >26MHz	RW	0x0

Table 3-1082. CM_CLKSEL_HDMI_MCASP_AUX

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6124	**Instance**	CKGEN_PRM
Description	Select the HDMI_CLK for McASP subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select HDMI_CLK divided by 1 0x1: Select HDMI_CLK divided by 2 0x2: Select HDMI_CLK divided by 4 0x3: Select HDMI_CLK divided by 8 0x4: Select HDMI_CLK divided by 16 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-1083. CM_CLKSEL_HDMI_TIMER

Address Offset	0x0000 0028		
Physical Address	0x4AE0 6128	**Instance**	CKGEN_PRM
Description	Select the HDMI_CLK for TIMER subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select HDMI_CLK divided by 1 0x1: Select HDMI_CLK divided by 2 0x2: Select HDMI_CLK divided by 4 0x3: Select HDMI_CLK divided by 8 0x4: Select HDMI_CLK divided by 16 0x5: Select HDMI_CLK divided by 22 0x6: Select HDMI_CLK divided by 32 0x7: Reserved	RW	0x0

Table 3-1084. CM_CLKSEL_MCASP_SYS

Address Offset	0x0000 002C		
Physical Address	0x4AE0 612C	**Instance**	CKGEN_PRM
Description	Select the SYS CLK for ABE_24M_FCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 8 0x1: Select SYS_CLK divided by 16	RW	0x0

Table 3-1085. CM_CLKSEL_MLBP_MCASP

Address Offset	0x0000 0030

Table 3-1085. CM_CLKSEL_MLBP_MCASP (continued)

Physical Address	0x4AE0 6130	**Instance**	CKGEN_PRM
Description	Select the MLBP_CLK for McASP subsystems. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select MLBP_CLK divided by 1 0x1: Select MLBP_CLK divided by 2 0x2: Select MLBP_CLK divided by 4 0x3: Select MLBP_CLK divided by 8 0x4: Select MLBP_CLK divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1086. CM_CLKSEL_MLB_MCASP

Address Offset	0x0000 0034		
Physical Address	0x4AE0 6134	**Instance**	CKGEN_PRM
Description	Select the MLB_CLK for McASP subsystems. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select MLB_CLK divided by 1 0x1: Select MLB_CLK divided by 2 0x2: Select MLB_CLK divided by 4 0x3: Select MLB_CLK divided by 8 0x4: Select MLB_CLK divided by 16 0x5: RESERVED 0x6: RESERVED1 0x7: RESERVED	RW	0x0

Table 3-1087. CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX

Address Offset	0x0000 0038		
Physical Address	0x4AE0 6138	**Instance**	CKGEN_PRM
Description	Select the PER_ABE_X1_GFCLK_CLK for McASP subsystems. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select PER_ABE_X1_GFCLK divided by 1 0x1: Select PER_ABE_X1_GFCLK divided by 2 0x2: Select PER_ABE_X1_GFCLK divided by 4 0x3: Select PER_ABE_X1_GFCLK divided by 8 0x4: Select PER_ABE_X1_GFCLK divided by 16 0x5: Reserved 0x6: Reserved 0x7: RESERVED	RW	0x0

Table 3-1088. CM_CLKSEL_TIMER_SYS

Address Offset	0x0000 0044		
Physical Address	0x4AE0 6144	**Instance**	CKGEN_PRM
Description	Select the SYS CLK for TIMERS subsystems. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 1 0x1: Select SYS_CLK divided by 2	RW	0x0

Table 3-1089. CM_CLKSEL_VIDEO1_MCASP_AUX

Address Offset	0x0000 0048		
Physical Address	0x4AE0 6148	**Instance**	CKGEN_PRM
Description	Select the VIDEO1_CLK for McASP subsystems. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select VIDEO1_CLK divided by 1 0x1: Select VIDEO1_CLK divided by 2 0x2: Select VIDEO1_CLK divided by 4 0x3: Select VIDEO1_CLK divided by 8 0x4: Select VIDEO1_CLK divided by 16 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-1090. CM_CLKSEL_VIDEO1_TIMER

Address Offset	0x0000 004C		
Physical Address	0x4AE0 614C	**Instance**	CKGEN_PRM
Description	Select the VIDEO1_CLK for TIMER subsystems. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select VIDEO1_CLK divided by 1 0x1: Select VIDEO1_CLK divided by 2 0x2: Select VIDEO1_CLK divided by 4 0x3: Select VIDEO1_CLK divided by 8 0x4: Select VIDEO1_CLK divided by 16 0x5: Select VIDEO1_CLK divided by 22 0x6: Select VIDEO1_CLK divided by 32 0x7: RESERVED	RW	0x0

Table 3-1091. CM_CLKSEL_VIDEO2_MCASP_AUX

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6150	**Instance**	CKGEN_PRM
Description	Select the VIDEO2_CLK for McASP subsystems. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select VIDEO2_CLK divided by 1 0x1: Select VIDEO2_CLK divided by 2 0x2: Select VIDEO2_CLK divided by 4 0x3: Select VIDEO2_CLK divided by 8 0x4: Select VIDEO2_CLK divided by 16 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-1092. CM_CLKSEL_VIDEO2_TIMER

Address Offset	0x0000 0054		
Physical Address	0x4AE0 6154	**Instance**	CKGEN_PRM
Description	Select the VIDEO2_CLK for TIMER subsystems. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select VIDEO2_CLK divided by 1 0x1: Select VIDEO2_CLK divided by 2 0x2: Select VIDEO2_CLK divided by 4 0x3: Select VIDEO2_CLK divided by 8 0x4: Select VIDEO2_CLK divided by 16 0x5: Select VIDEO2_CLK divided by 22 0x6: Select VIDEO2_CLK divided by 32 0x7: RESERVED	RW	0x0

Table 3-1093. CM_CLKSEL_CLKOUTMUX0

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6158	**Instance**	CKGEN_PRM
Description	Control the source of the CLKOUTMUX0_CLK.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CLKSEL				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	CLKSEL	Select the source clock for CLKOUTMUX0_CLK. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_SYS_CLK1_CLKOUTMUX 0x1: Selects divided version of EVE_GFCLK. See CM_CLKSEL_EVE_GFCLK_CLKOUTMUX 0x2: Selects divided version of PER_ABE_X1_GFCLK. See CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX 0x3: Selects divided version of MPU_GCLK. See CM_CLKSEL_MPU_GCLK_CLKOUTMUX 0x4: Selects divided version of DSP_GFCLK. See CM_CLKSEL_DSP_GFCLK_CLKOUTMUX 0x5: Selects divided version of IVA_GCLK. See CM_CLKSEL_IVA_GCLK_CLKOUTMUX 0x6: Selects divided version of GPU_GCLK. See CM_CLKSEL_GPU_GCLK_CLKOUTMUX 0x7: Selects divided version of CORE_DPLL_OUT_CLK. See CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX 0x8: Selects divided version of EMIF_PHY_GCLK. See CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX 0x9: Selects divided version of GMAC_250M_CLK. See CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX 0xA: Selects divided version of VIDEO2_CLK. See CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX 0xB: Selects divided version of VIDEO1_CLK. See CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX 0xC: Selects divided version of HDMI_CLK. See CM_CLKSEL_HDMI_CLK_CLKOUTMUX 0xD: Selects divided version of FUNC_96M_AON_CLK. See CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX 0xE: Selects divided version of L3INIT_480M_GFCLK. See CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX 0xF: Selects divided version of USB_OTG_CLK. See CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX 0x10: Selects divided version of SATA_CLK. See CM_CLKSEL_SATA_CLK_CLKOUTMUX 0x11: Selects divided version of PCIE_M2_CLK. See CM_CLKSEL_PCIE2_CLK_CLKOUTMUX 0x12: Selects divided version of APLL_PCIE_M2_CLK. See CM_CLKSEL_PCIE1_CLK_CLKOUTMUX 0x13: Selects divided version of EMU_CLK. See CM_CLKSEL_EMU_CLK_CLKOUTMUX 0x14: Selects divided version of OSC_32K_CLK. See CM_CLKSEL_OSC_32K_CLK_CLKOUTMUX Note: The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics. 0x15-0x1F: RESERVED	RW	0x0

Table 3-1094. CM_CLKSEL_CLKOUTMUX1

Address Offset	0x0000 005C		
Physical Address	0x4AE0 615C	**Instance**	CKGEN_PRM
Description	Control the source of the CLKOUTMUX1_CLK.		

Table 3-1094. CM_CLKSEL_CLKOUTMUX1 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CLKSEL				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	CLKSEL	Select the source clock for CLKOUTMUX1_CLK. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_SYS_CLK1_CLKOUTMUX 0x1: Selects divided version of SYS_CLK2. See CM_CLKSEL_SYS_CLK2_CLKOUTMUX 0x2: Selects divided version of PER_ABE_X1_GFCLK. See CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX 0x3: Selects divided version of MPU_GCLK. See CM_CLKSEL_MPU_GCLK_CLKOUTMUX 0x4: Selects divided version of DSP_GFCLK. See CM_CLKSEL_DSP_GFCLK_CLKOUTMUX 0x5: Selects divided version of IVA_GCLK. See CM_CLKSEL_IVA_GCLK_CLKOUTMUX 0x6: Selects divided version of GPU_GCLK. See CM_CLKSEL_GPU_GCLK_CLKOUTMUX 0x7: Selects divided version of CORE_DPLL_OUT_CLK. See CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX 0x8: Selects divided version of EMIF_PHY_GCLK. See CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX 0x9: Selects divided version of GMAC_250M_CLK. See CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX 0xA: Selects divided version of VIDEO2_CLK. See CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX 0xB: Selects divided version of VIDEO1_CLK. See CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX 0xC: Selects divided version of HDMI_CLK. See CM_CLKSEL_HDMI_CLK_CLKOUTMUX 0xD: Selects divided version of FUNC_96M_AON_CLK. See CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX 0xE: Selects divided version of L3INIT_480M_GFCLK. See CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX 0xF: Selects divided version of USB_OTG_CLK. See CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX 0x10: Selects divided version of SATA_CLK. See CM_CLKSEL_SATA_CLK_CLKOUTMUX 0x11: Selects divided version of PCIE_M2_CLK. See CM_CLKSEL_PCIE2_CLK_CLKOUTMUX 0x12: Selects divided version of APLL_PCIE_M2_CLK. See CM_CLKSEL_PCIE1_CLK_CLKOUTMUX 0x13: Selects divided version of EMU_CLK. See CM_CLKSEL_EMU_CLK_CLKOUTMUX 0x14: Selects divided version of OSC_32K_CLK. See CM_CLKSEL_OSC_32K_CLK_CLKOUTMUX Note: The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics. 0x15: Selects divided version of EVE_GFCLK. See CM_CLKSEL_EVE_GFCLK_CLKOUTMUX 0x16-0x1F: RESERVED	RW	0x0

Table 3-1095. CM_CLKSEL_CLKOUTMUX2

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6160	**Instance**	CKGEN_PRM

Table 3-1095. CM_CLKSEL_CLKOUTMUX2 (continued)

Description	Control the source of the CLKOUTMUX2_CLK.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CLKSEL				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	CLKSEL	Select the source clock for CLKOUTMUX2_CLK. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_SYS_CLK1_CLKOUTMUX 0x1: Selects divided version of SYS_CLK2. See CM_CLKSEL_SYS_CLK2_CLKOUTMUX 0x2: Selects divided version of PER_ABE_X1_GFCLK. See CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX 0x3: Selects divided version of MPU_GCLK. See CM_CLKSEL_MPU_GCLK_CLKOUTMUX 0x4: Selects divided version of DSP_GFCLK. See CM_CLKSEL_DSP_GFCLK_CLKOUTMUX 0x5: Selects divided version of IVA_GCLK. See CM_CLKSEL_IVA_GCLK_CLKOUTMUX 0x6: Selects divided version of GPU_GCLK. See CM_CLKSEL_GPU_GCLK_CLKOUTMUX 0x7: Selects divided version of CORE_DPLL_OUT_CLK. See CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX 0x8: Selects divided version of EMIF_PHY_GCLK. See CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX 0x9: Selects divided version of GMAC_250M_CLK. See CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX 0xA: Selects divided version of VIDEO2_CLK. See CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX 0xB: Selects divided version of VIDEO1_CLK. See CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX 0xC: Selects divided version of HDMI_CLK. See CM_CLKSEL_HDMI_CLK_CLKOUTMUX 0xD: Selects divided version of FUNC_96M_AON_CLK. See CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX 0xE: Selects divided version of L3INIT_480M_GFCLK. See CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX 0xF: Selects divided version of USB_OTG_CLK. See CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX 0x10: Selects divided version of SATA_CLK. See CM_CLKSEL_SATA_CLK_CLKOUTMUX 0x11: Selects divided version of PCIE_M2_CLK. See CM_CLKSEL_PCIE2_CLK_CLKOUTMUX 0x12: Selects divided version of APLL_PCIE_M2_CLK. See CM_CLKSEL_PCIE1_CLK_CLKOUTMUX 0x13: Selects divided version of EMU_CLK. See CM_CLKSEL_EMU_CLK_CLKOUTMUX 0x14: Selects divided version of OSC_32K_CLK. See CM_CLKSEL_OSC_32K_CLK_CLKOUTMUX Note: The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics. 0x15: Selects divided version of EVE_GFCLK. See CM_CLKSEL_EVE_GFCLK_CLKOUTMUX 0x16-0x1F: RESERVED	RW	0x0

Table 3-1096. CM_CLKSEL_HDMI_PLL_SYS

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6164	**Instance**	CKGEN_PRM

Table 3-1096. CM_CLKSEL_HDMI_PLL_SYS (continued)

Description	Control the source of the SYS clock for DPLL_HDMI
Type	RW

31–1	0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_HDMI 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1097. CM_CLKSEL_VIDEO1_PLL_SYS

Address Offset	0x0000 0068		
Physical Address	0x4AE0 6168	**Instance**	CKGEN_PRM
Description	Control the source of the SYS clock for DPLL_VIDEO1		
Type	RW		

31–1	0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_VIDEO1. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1098. CM_CLKSEL_VIDEO2_PLL_SYS

Address Offset	0x0000 006C		
Physical Address	0x4AE0 616C	**Instance**	CKGEN_PRM
Description	Control the source of the SYS clock for DPLL_VIDEO1		
Type	RW		

31–1	0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_VIDEO2. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1099. CM_CLKSEL_ABE_CLK_DIV

Address Offset	0x0000 0070		
Physical Address	0x4AE0 6170	**Instance**	CKGEN_PRM
Description	Select the ABE_CLK. [warm reset insensitive]		

Table 3-1099. CM_CLKSEL_ABE_CLK_DIV (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1100. CM_CLKSEL_ABE_GICLK_DIV

Address Offset	0x0000 0074		
Physical Address	0x4AE0 6174	**Instance**	CKGEN_PRM
Description	Select the ABE_GICLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2	RW	0x0

Table 3-1101. CM_CLKSEL_AESS_FCLK_DIV

Address Offset	0x0000 0078		
Physical Address	0x4AE0 6178	**Instance**	CKGEN_PRM
Description	Select the AESS_FCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2	RW	0x0

Table 3-1102. CM_CLKSEL_EVE_CLK

Address Offset	0x0000 0080		
Physical Address	0x4AE0 6180	**Instance**	CKGEN_PRM
Description	Control the source of the EVE_CLK for EVE1, EVE2		
Type	RW		

31:1	0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the EVE_CLK for EVE1, EVE2 0x0: Selects clock from DPLL_EVE 0x1: Selects clock from DPLL_DSP	RW	0x0

Table 3-1103. CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX

Address Offset	0x0000 0084		
Physical Address	0x4AE0 6184	**Instance**	CKGEN_PRM
Description	Select the USB_OTG_CLK. [warm reset insensitive]		
Type	RW		

31:3	2:0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1104. CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX

Address Offset	0x0000 0088		
Physical Address	0x4AE0 6188	**Instance**	CKGEN_PRM
Description	Select the CORE_DPLL_OUT_CLK. [warm reset insensitive]		
Type	RW		

31:3	2:0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1105. CM_CLKSEL_DSP_GFCLK_CLKOUTMUX

Address Offset	0x0000 008C		
Physical Address	0x4AE0 618C	**Instance**	CKGEN_PRM
Description	Select the DSP_GFCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1106. CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX

Address Offset	0x0000 0090		
Physical Address	0x4AE0 6190	**Instance**	CKGEN_PRM
Description	Select the EMIF_PHY_GCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1107. CM_CLKSEL_EMU_CLK_CLKOUTMUX

Address Offset	0x0000 0094		
Physical Address	0x4AE0 6194	**Instance**	CKGEN_PRM
Description	Select the EMU_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1108. CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX

Address Offset	0x0000 0098		
Physical Address	0x4AE0 6198	**Instance**	CKGEN_PRM
Description	Select the FUNC_96M_AON_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1109. CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX

Address Offset	0x0000 009C		
Physical Address	0x4AE0 619C	**Instance**	CKGEN_PRM
Description	Select the GMAC_250M_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1110. CM_CLKSEL_GPU_GCLK_CLKOUTMUX

Address Offset	0x0000 00A0		
Physical Address	0x4AE0 61A0	**Instance**	CKGEN_PRM
Description	Select the GPU_GCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1111. CM_CLKSEL_HDMI_CLK_CLKOUTMUX

Address Offset	0x0000 00A4		
Physical Address	0x4AE0 61A4	**Instance**	CKGEN_PRM
Description	Select the HDMI_CLK. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1112. CM_CLKSEL_IVA_GCLK_CLKOUTMUX

Address Offset	0x0000 00A8		
Physical Address	0x4AE0 61A8	**Instance**	CKGEN_PRM
Description	Select the IVA_GCLK. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1113. CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX

Address Offset	0x0000 00AC		
Physical Address	0x4AE0 61AC	**Instance**	CKGEN_PRM
Description	Select the L3INIT_480M_GFCLK. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1114. CM_CLKSEL_MPU_GCLK_CLKOUTMUX

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 61B0	**Instance**	CKGEN_PRM
Description	Select the MPU_GCLK. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divide d by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1115. CM_CLKSEL_PCIE1_CLK_CLKOUTMUX

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 61B4	**Instance**	CKGEN_PRM
Description	Select the PCIE1_DCLK, where APLL_PCIE_M2_CLK is the source clock of PCIE1_DCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select APLL_PCIE_M2_CLK divided by 1 0x1: Select APLL_PCIE_M2_CLK divided by 2 0x2: Select APLL_PCIE_M2_CLK divided by 4 0x3: Select APLL_PCIE_M2_CLK divided by 8 0x4: Select APLL_PCIE_M2_CLK divided by 16 0x5: Select APLL_PCIE_M2_CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1116. CM_CLKSEL_PCIE2_CLK_CLKOUTMUX

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 61B8	**Instance**	CKGEN_PRM
Description	Select the PCIE2_DCLK, where PCIE_M2_CLK is the source clock of PCIE2_DCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select PCIE_M2_CLK divided by 1 0x1: Select PCIE_M2_CLK divided by 2 0x2: Select PCIE_M2_CLK divided by 4 0x3: Select PCIE_M2_CLK divided by 8 0x4: Select PCIE_M2_CLK divided by 16 0x5: Select PCIE_M2_CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1117. CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 61BC	**Instance**	CKGEN_PRM
Description	Select the PER_ABE_X1_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1118. CM_CLKSEL_SATA_CLK_CLKOUTMUX

Address Offset	0x0000 00C0		
Physical Address	0x4AE0 61C0	**Instance**	CKGEN_PRM
Description	Select the SATA_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1119. CM_CLKSEL_OSC_32K_CLK_CLKOUTMUX

Address Offset	0x0000 00C4		
Physical Address	0x4AE0 61C4	**Instance**	CKGEN_PRM
Description	Select the OSC_32K_CLK. [warm reset insensitive] Note: The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1120. CM_CLKSEL_SYS_CLK1_CLKOUTMUX

Address Offset	0x0000 00C8		
Physical Address	0x4AE0 61C8	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1121. CM_CLKSEL_SYS_CLK2_CLKOUTMUX

Address Offset	0x0000 00CC		
Physical Address	0x4AE0 61CC	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK2. [warm reset insensitive]		
Type	RW		

31:3	2:0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1122. CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX

Address Offset	0x0000 00D0		
Physical Address	0x4AE0 61D0	**Instance**	CKGEN_PRM
Description	Select the VIDEO1_CLK. [warm reset insensitive]		
Type	RW		

31:3	2:0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1123. CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 61D4	**Instance**	CKGEN_PRM
Description	Select the VIDEO2_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1124. CM_CLKSEL_ABE_LP_CLK

Address Offset	0x0000 00D8		
Physical Address	0x4AE0 61D8	**Instance**	CKGEN_PRM
Description	Select the ABE_LP_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CL KS EL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 16 0x1: Select CLK divided by 32	RW	0x0

Table 3-1125. CM_CLKSEL_EVE_GFCLK_CLKOUTMUX

Address Offset	0x0000 00E0

Table 3-1125. CM_CLKSEL_EVE_GFCLK_CLKOUTMUX (continued)

Physical Address	0x4AE0 61E0	**Instance**	CKGEN_PRM
Description	Select the EVE_GFCLK. [warm reset insensitive]		
Type	RW		

31–3	2–0
RESERVED	CLKSEL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

3.13.28 CORE_PRM Registers

3.13.28.1 CORE_PRM Register Summary

Table 3-1126. CORE_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM Physical Address L4_WKUP Interconnect
PM_CORE_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6700
PM_CORE_PWRSTST	RW	32	0x0000 0004	0x4AE0 6704
RM_L3MAIN1_L3_MAIN_1_CONTEXT	RW	32	0x0000 0024	0x4AE0 6724
RM_L3MAIN1_GPMC_CONTEXT	RW	32	0x0000 002C	0x4AE0 672C
RM_L3MAIN1_MMU_EDMA_CONTEXT	RW	32	0x0000 0034	0x4AE0 6734
RM_L3MAIN1_MMU_PCIESS_CONTEXT	RW	32	0x0000 004C	0x4AE0 674C
PM_L3MAIN1_OCMC_RAM1_WKDEP	RW	32	0x0000 0050	0x4AE0 6750
RM_L3MAIN1_OCMC_RAM1_CONTEXT	RW	32	0x0000 0054	0x4AE0 6754
PM_L3MAIN1_OCMC_RAM2_WKDEP	RW	32	0x0000 0058	0x4AE0 6758
RM_L3MAIN1_OCMC_RAM2_CONTEXT	RW	32	0x0000 005C	0x4AE0 675C
PM_L3MAIN1_OCMC_RAM3_WKDEP	RW	32	0x0000 0060	0x4AE0 6760
RM_L3MAIN1_OCMC_RAM3_CONTEXT	RW	32	0x0000 0064	0x4AE0 6764
RESERVED	R	32	0x0000 006C	0x4AE0 676C
PM_L3MAIN1_TPCC_WKDEP	RW	32	0x0000 0070	0x4AE0 6770
RM_L3MAIN1_TPCC_CONTEXT	RW	32	0x0000 0074	0x4AE0 6774
PM_L3MAIN1_TPTC1_WKDEP	RW	32	0x0000 0078	0x4AE0 6778
RM_L3MAIN1_TPTC1_CONTEXT	RW	32	0x0000 007C	0x4AE0 677C
PM_L3MAIN1_TPTC2_WKDEP	RW	32	0x0000 0080	0x4AE0 6780
RM_L3MAIN1_TPTC2_CONTEXT	RW	32	0x0000 0084	0x4AE0 6784
RM_L3MAIN1_VCP1_CONTEXT	RW	32	0x0000 008C	0x4AE0 678C
RM_L3MAIN1_VCP2_CONTEXT	RW	32	0x0000 0094	0x4AE0 6794
RESERVED	RW	32	0x0000 009C	0x4AE0 679C
RESERVED	RW	32	0x0000 00A4	0x4AE0 67A4
RESERVED	RW	32	0x0000 00AC	0x4AE0 67AC
RESERVED	RW	32	0x0000 00B4	0x4AE0 67B4
RESERVED	RW	32	0x0000 00BC	0x4AE0 67BC

Table 3-1126. CORE_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM Physical Address L4_WKUP Interconnect
RESERVED	RW	32	0x0000 00C4	0x4AE0 67C4
RESERVED	RW	32	0x0000 00CC	0x4AE0 67CC
RESERVED	RW	32	0x0000 00D4	0x4AE0 67D4
RESERVED	RW	32	0x0000 00DC	0x4AE0 67DC
RESERVED	RW	32	0x0000 00F4	0x4AE0 67F4
RESERVED	RW	32	0x0000 00FC	0x4AE0 67FC
RM_IPU2_RSTCTRL	RW	32	0x0000 0210	0x4AE0 6910
RM_IPU2_RSTST	RW	32	0x0000 0214	0x4AE0 6914
RM_IPU2_IPU2_CONTEXT	RW	32	0x0000 0224	0x4AE0 6924
RM_DMA_DMA_SYSTEM_CONTEXT	RW	32	0x0000 0324	0x4AE0 6A24
RM_EMIF_DMM_CONTEXT	RW	32	0x0000 0424	0x4AE0 6B24
RM_EMIF_EMIF_OCP_FW_CONTEXT	RW	32	0x0000 042C	0x4AE0 6B2C
RM_EMIF_EMIF1_CONTEXT	RW	32	0x0000 0434	0x4AE0 6B34
RM_EMIF_EMIF2_CONTEXT	RW	32	0x0000 043C	0x4AE0 6B3C
RM_EMIF_EMIF_DLL_CONTEXT	RW	32	0x0000 0444	0x4AE0 6B44
RM_ATL_ATL_CONTEXT	RW	32	0x0000 0524	0x4AE0 6C24
RM_L4CFG_L4_CFG_CONTEXT	RW	32	0x0000 0624	0x4AE0 6D24
RM_L4CFG_SPINLOCK_CONTEXT	RW	32	0x0000 062C	0x4AE0 6D2C
RM_L4CFG_MAILBOX1_CONTEXT	RW	32	0x0000 0634	0x4AE0 6D34
RM_L4CFG_SAR_ROM_CONTEXT	RW	32	0x0000 063C	0x4AE0 6D3C
RM_L4CFG_OCP2SCP2_CONTEXT	RW	32	0x0000 0644	0x4AE0 6D44
RM_L4CFG_MAILBOX2_CONTEXT	RW	32	0x0000 064C	0x4AE0 6D4C
RM_L4CFG_MAILBOX3_CONTEXT	RW	32	0x0000 0654	0x4AE0 6D54
RM_L4CFG_MAILBOX4_CONTEXT	RW	32	0x0000 065C	0x4AE0 6D5C
RM_L4CFG_MAILBOX5_CONTEXT	RW	32	0x0000 0664	0x4AE0 6D64
RM_L4CFG_MAILBOX6_CONTEXT	RW	32	0x0000 066C	0x4AE0 6D6C
RM_L4CFG_MAILBOX7_CONTEXT	RW	32	0x0000 0674	0x4AE0 6D74
RM_L4CFG_MAILBOX8_CONTEXT	RW	32	0x0000 067C	0x4AE0 6D7C
RM_L4CFG_MAILBOX9_CONTEXT	RW	32	0x0000 0684	0x4AE0 6D84
RM_L4CFG_MAILBOX10_CONTEXT	RW	32	0x0000 068C	0x4AE0 6D8C
RM_L4CFG_MAILBOX11_CONTEXT	RW	32	0x0000 0694	0x4AE0 6D94
RM_L4CFG_MAILBOX12_CONTEXT	RW	32	0x0000 069C	0x4AE0 6D9C
RM_L4CFG_MAILBOX13_CONTEXT	RW	32	0x0000 06A4	0x4AE0 6DA4
RESERVED	RW	32	0x0000 06AC	0x4AE0 6DAC
RESERVED	RW	32	0x0000 06B4	0x4AE0 6DB4
RESERVED	RW	32	0x0000 06BC	0x4AE0 6DBC
RESERVED	RW	32	0x0000 06C4	0x4AE0 6DC4
RM_L3INSTR_L3_MAIN_2_CONTEXT	RW	32	0x0000 0724	0x4AE0 6E24
RM_L3INSTR_L3_INSTR_CONTEXT	RW	32	0x0000 072C	0x4AE0 6E2C
RM_L3INSTR_OCP_WP_NOC_CONTEXT	RW	32	0x0000 0744	0x4AE0 6E44

3.13.28.2 CORE_PRM Register Description

Table 3-1127. PM_CORE_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6700	**Instance**	CORE_PRM
Description	This register controls the CORE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	OCP_NRET_BANK_ONSTATE	IPU_UNICACHE_ONSTATE	IPU_L2RAM_ONSTATE	CORE_OCMRAM_ONSTATE	CORE_OTHER_BANK_ONSTATE	RESERVED	OCP_NRET_BANK_RETSTATE	IPU_UNICACHE_RETSTATE	IPU_L2RAM_RETSTATE	CORE_OCMRAM_RETSTATE	CORE_OTHER_BANK_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	OCP_NRET_BANK_ONSTATE	OCP_WP bank and DMM bank2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
23:22	IPU_UNICACHE_ONSTATE	IPU UNICACHE bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
21:20	IPU_L2RAM_ONSTATE	IPU L2 bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	CORE_OCMRAM_ONSTATE	OCMRAM bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	CORE_OTHER_BANK_ONSTATE	DMA/ICR bank and DMM bank1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:13	RESERVED		R	0x0
12	OCP_NRET_BANK_RETSTATE	Note: Not supported on this device.	R	0x0
11	IPU_UNICACHE_RETSTATE	Note: Not supported on this device.	R	0x0
10	IPU_L2RAM_RETSTATE	Note: Not supported on this device.	R	0x0
9	CORE_OCMRAM_RETSTATE	Note: Not supported on this device.	R	0x0
8	CORE_OTHER_BANK_RETSTATE	Note: Not supported on this device.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Note: Not supported on this device.	R	0x0
1:0	POWERSTATE	Power state control 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1128. PM_CORE_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6704	**Instance**	CORE_PRM

Table 3-1128. PM_CORE_PWRSTST (continued)

Description	This register provides a status on the current CORE power domain state. [warm reset insensitive]
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED						OCP_NRET_BANK_STATEST		IPU_UNICACHE_STATEST		IPU_L2RAM_STATEST		CORE_OCMRAM_STATEST		CORE_OTHER_BANK_STATEST		RESERVED	LOGICSTATEST	POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:14	RESERVED		R	0x0
13:12	OCP_NRET_BANK_STATEST	OCP_WP bank and DMM bank2 state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
11:10	IPU_UNICACHE_STATEST	IPU UNICACHE bank state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
9:8	IPU_L2RAM_STATEST	IPU L2 bank state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	CORE_OCMRAM_STATEST	OCMRAM bank state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
5:4	CORE_OTHER_BANK_STATEST	DMA/ICR bank and DMM bank1 state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1129. RM_L3MAIN1_L3_MAIN_1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6724	**Instance**	CORE_PRM
Description	This register contains dedicated L3_MAIN_1 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1130. RM_L3MAIN1_GPMC_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 672C	**Instance**	CORE_PRM
Description	This register contains dedicated GPMC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1131. RM_L3MAIN1_MMU_EDMA_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 6734	**Instance**	CORE_PRM
Description	This register contains dedicated MMU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1132. RM_L3MAIN1_MMU_PCIESS_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 674C	**Instance**	CORE_PRM
Description	This register contains dedicated MMU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1133. PM_L3MAIN1_OCMC_RAM1_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6750	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on OCMC_RAM1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_OCMC_RAM1_EVE2	WKUPDEP_OCMC_RAM1_EVE1	WKUPDEP_OCMC_RAM1_DSP2	WKUPDEP_OCMC_RAM1_IPU1	RESERVED	WKUPDEP_OCMC_RAM1_DSP1	WKUPDEP_OCMC_RAM1_IPU2	WKUPDEP_OCMC_RAM1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_OCMC_RAM1_EVE2	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_OCMC_RAM1_EVE1	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_OCMC_RAM1_DSP2	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_OCMC_RAM1_IPU1	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_OCMC_RAM1_DSP1	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_OCMC_RAM1_IPU2	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_OCMC_RAM1_MPU	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1134. RM_L3MAIN1_OCMC_RAM1_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 6754	**Instance**	CORE_PRM
Description	This register contains dedicated OCMC_RAM context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_CORE_OCMRAM	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	LOSTMEM_CORE_OCMRAM	Specify if memory-based context in CORE_OCMRAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1135. PM_L3MAIN1_OCMC_RAM2_WKDEP

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6758	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on OCMC_RAM2 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_OCMC_RAM2_EVE2	WKUPDEP_OCMC_RAM2_EVE1	WKUPDEP_OCMC_RAM2_DSP2	WKUPDEP_OCMC_RAM2_IPU1	RESERVED	WKUPDEP_OCMC_RAM2_DSP1	WKUPDEP_OCMC_RAM2_IPU2	WKUPDEP_OCMC_RAM2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_OCMC_RAM2_EVE2	Wakeup dependency from OCMC_RAM2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_OCMC_RAM2_EVE1	Wakeup dependency from OCMC_RAM2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_OCMC_RAM2_DSP2	Wakeup dependency from OCMC_RAM2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_OCMC_RAM2_IPU1	Wakeup dependency from OCMC_RAM2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_OCMC_RAM2_DSP1	Wakeup dependency from OCMC_RAM2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_OCMC_RAM2_IPU2	Wakeup dependency from OCMC_RAM2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_OCMC_RAM2_MPU	Wakeup dependency from OCMC_RAM2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1136. RM_L3MAIN1_OCMC_RAM2_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 675C	**Instance**	CORE_PRM
Description	This register contains dedicated OCMC_RAM2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CORE_OCMRAM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OCMRAM	Specify if memory-based context in CORE_OCMRAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1137. PM_L3MAIN1_OCMC_RAM3_WKDEP

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6760	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on OCMC_RAM3 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_OCMC_RAM3_EVE2	WKUPDEP_OCMC_RAM3_EVE1	WKUPDEP_OCMC_RAM3_DSP2	WKUPDEP_OCMC_RAM3_IPU1	RESERVED	WKUPDEP_OCMC_RAM3_DSP1	WKUPDEP_OCMC_RAM3_IPU2	WKUPDEP_OCMC_RAM3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_OCMC_RAM3_EVE2	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_OCMC_RAM3_EVE1	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_OCMC_RAM3_DSP2	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_OCMC_RAM3_IPU1	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_OCMC_RAM3_DSP1	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_OCMC_RAM3_IPU2	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_OCMC_RAM3_MPU	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1138. RM_L3MAIN1_OCMC_RAM3_CONTEXT

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6764	**Instance**	CORE_PRM
Description	This register contains dedicated OCMC_RAM3 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_CORE_OCMRAM	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OCMRAM	Specify if memory-based context in CORE_OCMRAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1139. PM_L3MAIN1_TPCC_WKDEP

Address Offset	0x0000 0070		
Physical Address	0x4AE0 6770	**Instance**	CORE_PRM

Table 3-1139. PM_L3MAIN1_TPCC_WKDEP (continued)

Description	This register controls wakeup dependency based on TPCC service requests.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_TPCC_EVE2	WKUPDEP_TPCC_EVE1	WKUPDEP_TPCC_DSP2	WKUPDEP_TPCC_IPU1	RESERVED	WKUPDEP_TPCC_DSP1	WKUPDEP_TPCC_IPU2	WKUPDEP_TPCC_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TPCC_EVE2	Wakeup dependency from TPCC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TPCC_EVE1	Wakeup dependency from TPCC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TPCC_DSP2	Wakeup dependency from TPCC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TPCC_IPU1	Wakeup dependency from TPCC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TPCC_DSP1	Wakeup dependency from TPCC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TPCC_IPU2	Wakeup dependency from TPCC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_TPCC_MPU	Wakeup dependency from TPCC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1140. RM_L3MAIN1_TPCC_CONTEXT

Address Offset	0x0000 0074		
Physical Address	0x4AE0 6774	**Instance**	CORE_PRM
Description	This register contains dedicated TPCC context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_TPCC_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_TPCC_BANK	Specify if memory-based context in TPCC_MEM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1141. PM_L3MAIN1_TPTC1_WKDEP

Address Offset	0x0000 0078		
Physical Address	0x4AE0 6778	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on TPTC service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_TPTC1_EVE2	WKUPDEP_TPTC1_EVE1	WKUPDEP_TPTC1_DSP2	WKUPDEP_TPTC1_IPU1	RESERVED	WKUPDEP_TPTC1_DSP1	WKUPDEP_TPTC1_IPU2	WKUPDEP_TPTC1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TPTC1_EVE2	Wakeup dependency from TPTC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TPTC1_EVE1	Wakeup dependency from TPTC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TPTC1_DSP2	Wakeup dependency from TPTC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TPTC1_IPU1	Wakeup dependency from TPTC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TPTC1_DSP1	Wakeup dependency from TPTC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TPTC1_IPU2	Wakeup dependency from TPTC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TPTC1_MPU	Wakeup dependency from TPTC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1142. RM_L3MAIN1_TPTC1_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 677C	**Instance**	CORE_PRM

Table 3-1142. RM_L3MAIN1_TPTC1_CONTEXT (continued)

Description	This register contains dedicated TPTC1 context statuses. [warm reset insensitive]
Type	RW

31-9	8	7-2	1	0
RESERVED	LOSTMEM_TPTC_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_TPTC_BANK	Specify if memory-based context in TPTC_MEM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1143. PM_L3MAIN1_TPTC2_WKDEP

Address Offset	0x0000 0080		
Physical Address	0x4AE0 6780	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on TPTC service requests.		
Type	RW		

31-10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TPTC2_EVE2	WKUPDEP_TPTC2_EVE1	WKUPDEP_TPTC2_DSP2	WKUPDEP_TPTC2_IPU1	RESERVED	WKUPDEP_TPTC2_DSP1	WKUPDEP_TPTC2_IPU2	WKUPDEP_TPTC2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_TPTC2_EVE2	Wakeup dependency from TPTC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TPTC2_EVE1	Wakeup dependency from TPTC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TPTC2_DSP2	Wakeup dependency from TPTC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TPTC2_IPU1	Wakeup dependency from TPTC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TPTC2_DSP1	Wakeup dependency from TPTC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TPTC2_IPU2	Wakeup dependency from TPTC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TPTC2_MPU	Wakeup dependency from TPTC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1144. RM_L3MAIN1_TPTC2_CONTEXT

Address Offset	0x0000 0084		
Physical Address	0x4AE0 6784	**Instance**	CORE_PRM
Description	This register contains dedicated TPTC2 context statuses. [warm reset insensitive]		
Type	RW		

31–9	8	7–2	1	0
RESERVED	LOSTMEM_TPTC_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_TPTC_BANK	Specify if memory-based context in TPTC_MEM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1145. RM_L3MAIN1_VCP1_CONTEXT

Address Offset	0x0000 008C		
Physical Address	0x4AE0 678C	**Instance**	CORE_PRM
Description	This register contains dedicated VCP1 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_VCP_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VCP_BANK	Specify if memory-based context in VCP memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1146. RM_L3MAIN1_VCP2_CONTEXT

Address Offset	0x0000 0094		
Physical Address	0x4AE0 6794	**Instance**	CORE_PRM
Description	This register contains dedicated VCP2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_VCP_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VCP_BANK	Specify if memory-based context in VCP memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1147. RM_IPU2_RSTCTRL

Address Offset	0x0000 0210		
Physical Address	0x4AE0 6910	**Instance**	CORE_PRM
Description	This register controls the release of the IPU2 sub-system resets.		
Type	RW		

31–3	2	1	0
RESERVED	RST_IPU	RST_CPU1	RST_CPU0

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	RST_IPU	IPU system reset control. 0x0: Reset is cleared for IPU CACHE MMU 0x1: Reset is asserted for the IPU CACHE MMU	RW	0x1
1	RST_CPU1	IPU Cortex M4 CPU1 reset control 0x0: Reset is cleared for the IPU Cortex M4 CPU1 0x1: Reset is asserted for the IPU Cortex M4 CPU1	RW	0x1
0	RST_CPU0	IPU Cortex M4 CPU0 reset control. 0x0: Reset is cleared for the IPU Cortex M4 CPU0 0x1: Reset is asserted for the IPU Cortex M4 CPU0	RW	0x1

Table 3-1148. RM_IPU2_RSTST

Address Offset	0x0000 0214		
Physical Address	0x4AE0 6914	**Instance**	CORE_PRM
Description	This register logs the different reset sources of the IPU2 SS. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									RST_ICECRUSHER_CPU1	RST_ICECRUSHER_CPU0	RST_EMULATION_CPU1	RST_EMULATION_CPU0	RST_IPU	RST_CPU1	RST_CPU0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	RST_ICECRUSHER_CPU1	Cortex M4 CPU1 has been reset due to IPU ICECRUSHER1 reset source 0x0: No icecrusher reset 0x1: CPU1 has been reset upon icecrusher reset	RW	0x0
5	RST_ICECRUSHER_CPU0	Cortex M4 CPU0 has been reset due to IPU ICECRUSHER0 reset source 0x0: No icecrusher reset 0x1: CPU0 has been reset upon icecrusher reset	RW	0x0
4	RST_EMULATION_CPU1	Cortex M4 CPU1 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: CPU1 has been reset upon emulation reset	RW	0x0
3	RST_EMULATION_CPU0	Cortex M4 CPU0 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: CPU0 has been reset upon emulation reset	RW	0x0
2	RST_IPU	IPU system SW reset status 0x0: No SW reset occurred 0x1: IPU MMU and CACHE interface has been reset upon SW reset	RW	0x0
1	RST_CPU1	IPU Cortex-M4 CPU1 SW reset status 0x0: No SW reset occurred 0x1: Cortex M4 CPU1 has been reset upon SW reset	RW	0x0
0	RST_CPU0	IPU Cortex-M4 CPU0 SW reset status 0x0: No SW reset occurred 0x1: Cortex M4 CPU0 has been reset upon SW reset	RW	0x0

Table 3-1149. RM_IPU2_IPU2_CONTEXT

Address Offset	0x0000 0224		
Physical Address	0x4AE0 6924	**Instance**	CORE_PRM
Description	This register contains dedicated IPU2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_IPU_L2RAM	LOSTMEM_IPU_UNICACHE	RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	LOSTMEM_IPU_L2RAM	Specify if memory-based context in IPU_L2RAM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_IPU_UNICACHE	Specify if memory-based context in IPU_UNICACHE memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1150. RM_DMA_DMA_SYSTEM_CONTEXT

Address Offset	0x0000 0324		
Physical Address	0x4AE0 6A24	**Instance**	CORE_PRM
Description	This register contains dedicated SDMA context statuses. [warm reset insensitive]		
Type	RW		

31–9	8	7–2	1	0
RESERVED	LOSTMEM_CORE_OTHER_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OTHER_BANK	Specify if memory-based context in CORE_OTHER_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DMA_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1151. RM_EMIF_DMM_CONTEXT

Address Offset	0x0000 0424		
Physical Address	0x4AE0 6B24	**Instance**	CORE_PRM
Description	This register contains dedicated DMM context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: MAINTAINED 0x1: LOST	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal) 0x0: MAINTAINED 0x1: LOST	RW	0x1

Table 3-1152. RM_EMIF_EMIF_OCP_FW_CONTEXT

Address Offset	0x0000 042C		
Physical Address	0x4AE0 6B2C	**Instance**	CORE_PRM
Description	This register contains dedicated EMIF_OCP_FW context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1153. RM_EMIF_EMIF1_CONTEXT

Address Offset	0x0000 0434		
Physical Address	0x4AE0 6B34	**Instance**	CORE_PRM
Description	This register contains dedicated EMIF_1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1154. RM_EMIF_EMIF2_CONTEXT

Address Offset	0x0000 043C		
Physical Address	0x4AE0 6B3C	**Instance**	CORE_PRM
Description	This register contains dedicated EMIF_2 context statuses. [warm reset insensitive]		

Table 3-1154. RM_EMIF_EMIF2_CONTEXT (continued)

Type	RW

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1155. RM_EMIF_EMIF_DLL_CONTEXT

Address Offset	0x0000 0444		
Physical Address	0x4AE0 6B44	**Instance**	CORE_PRM
Description	This register contains dedicated DLL context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DLL_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1156. RM_ATL_ATL_CONTEXT

Address Offset	0x0000 0524		
Physical Address	0x4AE0 6C24	**Instance**	CORE_PRM
Description	This register contains dedicated ATL context statuses. [warm reset insensitive]		

Table 3-1156. RM_ATL_ATL_CONTEXT (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_ATL_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_ATL_BANK	Specify if memory-based context in ATL_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1157. RM_L4CFG_L4_CFG_CONTEXT

Address Offset	0x0000 0624		
Physical Address	0x4AE0 6D24	**Instance**	CORE_PRM
Description	This register contains dedicated L4_CFG context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1158. RM_L4CFG_SPINLOCK_CONTEXT

Address Offset	0x0000 062C		
Physical Address	0x4AE0 6D2C	**Instance**	CORE_PRM
Description	This register contains dedicated HW_SEM context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1159. RM_L4CFG_MAILBOX1_CONTEXT

Address Offset	0x0000 0634		
Physical Address	0x4AE0 6D34	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX1 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1160. RM_L4CFG_SAR_ROM_CONTEXT

Address Offset	0x0000 063C		
Physical Address	0x4AE0 6D3C	**Instance**	CORE_PRM
Description	This register contains dedicated SAR_ROM context statuses. [warm reset insensitive] **NOTE: This register is NOT supported on this device.**		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1161. RM_L4CFG_OCP2SCP2_CONTEXT

Address Offset	0x0000 0644		
Physical Address	0x4AE0 6D44	**Instance**	CORE_PRM
Description	This register contains dedicated OCP2SCP2 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1162. RM_L4CFG_MAILBOX2_CONTEXT

Address Offset	0x0000 064C		
Physical Address	0x4AE0 6D4C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX2 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1163. RM_L4CFG_MAILBOX3_CONTEXT

Address Offset	0x0000 0654		
Physical Address	0x4AE0 6D54	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX3 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: MAINTAINED 0x1: LOST	RW	0x1
0	RESERVED		R	0x0

Table 3-1164. RM_L4CFG_MAILBOX4_CONTEXT

Address Offset	0x0000 065C		
Physical Address	0x4AE0 6D5C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX4 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: MAINTAINED 0x1: LOST	RW	0x1
0	RESERVED		R	0x0

Table 3-1165. RM_L4CFG_MAILBOX5_CONTEXT

Address Offset	0x0000 0664		
Physical Address	0x4AE0 6D64	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX5 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1166. RM_L4CFG_MAILBOX6_CONTEXT

Address Offset	0x0000 066C		
Physical Address	0x4AE0 6D6C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX6 context statuses. [warm reset insensitive]		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="30">RESERVED</td><td>LOSTCONTEXT_RFF</td><td>RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1167. RM_L4CFG_MAILBOX7_CONTEXT

Address Offset	0x0000 0674		
Physical Address	0x4AE0 6D74	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX7 context statuses. [warm reset insensitive]		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="30">RESERVED</td><td>LOSTCONTEXT_RFF</td><td>RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1168. RM_L4CFG_MAILBOX8_CONTEXT

Address Offset	0x0000 067C		
Physical Address	0x4AE0 6D7C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX8 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1169. RM_L4CFG_MAILBOX9_CONTEXT

Address Offset	0x0000 0684		
Physical Address	0x4AE0 6D84	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX9 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1170. RM_L4CFG_MAILBOX10_CONTEXT

Address Offset	0x0000 068C		
Physical Address	0x4AE0 6D8C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX10 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1171. RM_L4CFG_MAILBOX11_CONTEXT

Address Offset	0x0000 0694		
Physical Address	0x4AE0 6D94	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX11 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1172. RM_L4CFG_MAILBOX12_CONTEXT

Address Offset	0x0000 069C		
Physical Address	0x4AE0 6D9C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX12 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1173. RM_L4CFG_MAILBOX13_CONTEXT

Address Offset	0x0000 06A4		
Physical Address	0x4AE0 6DA4	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX13 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1174. RM_L3INSTR_L3_MAIN_2_CONTEXT

Address Offset	0x0000 0724		
Physical Address	0x4AE0 6E24	**Instance**	CORE_PRM
Description	This register contains dedicated L3_3 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1175. RM_L3INSTR_L3_INSTR_CONTEXT

Address Offset	0x0000 072C		
Physical Address	0x4AE0 6E2C	**Instance**	CORE_PRM
Description	This register contains dedicated L3_INSTR context statuses. [warm reset insensitive]		
Type	RW		

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1176. RM_L3INSTR_OCP_WP_NOC_CONTEXT

Address Offset	0x0000 0744		
Physical Address	0x4AE0 6E44	**Instance**	CORE_PRM
Description	This register contains dedicated OCP_WP1 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_CORE_NRET_BANK	RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_NRET_BANK	Specify if memory-based context in CORE_NRET_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.13.29 CUSTEFUSE_PRM Registers

3.13.29.1 CUSTEFUSE_PRM Register Summary

Table 3-1177. CUSTEFUSE_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CUSTEFUSE_PRM Physical Address L4_WKUP Interconnect
PM_CUSTEFUSE_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7600
PM_CUSTEFUSE_PWRSTST	RW	32	0x0000 0004	0x4AE0 7604

Table 3-1177. CUSTEFUSE_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CUSTEFUSE_PRM Physical Address L4_WKUP Interconnect
RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT	RW	32	0x0000 0024	0x4AE0 7624

3.13.29.2 CUSTEFUSE_PRM Register Description

Table 3-1178. PM_CUSTEFUSE_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7600	**Instance**	CUSTEFUSE_PRM
Description	This register controls the CUSTEFUSE power state to reach upon a domain sleep transition		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="27">RESERVED</td><td>LOWPOWERSTATECHANGE</td><td colspan="2">RESERVED</td><td colspan="2">POWERSTATE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x0

Table 3-1179. Register Call Summary for Register PM_CUSTEFUSE_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1]

PRCM Register Manual

- CUSTEFUSE_PRM Register Summary: [2]

Table 3-1180. PM_CUSTEFUSE_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7604	**Instance**	CUSTEFUSE_PRM
Description	This register provides a status on the current CUSTEFUSE power domain state. [warm reset insensitive]		

Table 3-1180. PM_CUSTEFUSE_PWRSTST (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED																	LOGICSTATEST	POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1181. Register Call Summary for Register PM_CUSTEFUSE_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3]

PRCM Register Manual

- CUSTEFUSE_PRM Register Summary: [4]

Table 3-1182. RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7624	**Instance**	CUSTEFUSE_PRM
Description	This register contains dedicated CUSTEFUSE module context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CUSTEFUSE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1183. Register Call Summary for Register RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT

Power Management Functional Description

- PD_CUSTEFUSE Description: [0]

PRCM Register Manual

- CUSTEFUSE_PRM Register Summary: [1]

3.13.30 DEVICE_PRM Registers

3.13.30.1 DEVICE_PRM Register Summary

Table 3-1184. DEVICE_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM Physical Address L4_WKUP Interconnect
PRM_RSTCTRL	RW	32	0x0000 0000	0x4AE0 7D00
PRM_RSTST	RW	32	0x0000 0004	0x4AE0 7D04
PRM_RSTTIME	RW	32	0x0000 0008	0x4AE0 7D08
RESERVED	RW	32	0x0000 000C	0x4AE0 7D0C
RESERVED	RW	32	0x0000 0010	0x4AE0 7D10
RESERVED	RW	32	0x0000 0014	0x4AE0 7D14
PRM_PSCON_COUNT	RW	32	0x0000 0018	0x4AE0 7D18
PRM_IO_COUNT	RW	32	0x0000 001C	0x4AE0 7D1C
PRM_IO_PMCTRL	RW	32	0x0000 0020	0x4AE0 7D20
RESERVED_v (v = 0 to 37)	R	32	0x0000 0024 + (v*4)	0x4AE0 7D24 + (v*4)
PRM_SRAM_COUNT	RW	32	0x0000 00BC	0x4AE0 7DBC
RESERVED	RW	32	0x0000 00C0	0x4AE0 7DC0
PRM_SLDO_CORE_SETUP	RW	32	0x0000 00C4	0x4AE0 7DC4
PRM_SLDO_CORE_CTRL	R	32	0x0000 00C8	0x4AE0 7DC8
PRM_SLDO_MPU_SETUP	RW	32	0x0000 00CC	0x4AE0 7DCC
PRM_SLDO_MPU_CTRL	RW	32	0x0000 00D0	0x4AE0 7DD0
PRM_SLDO_GPU_SETUP	RW	32	0x0000 00D4	0x4AE0 7DD4
PRM_SLDO_GPU_CTRL	RW	32	0x0000 00D8	0x4AE0 7DD8
PRM_ABBLDO_MPU_SETUP	RW	32	0x0000 00DC	0x4AE0 7DDC
PRM_ABBLDO_MPU_CTRL	RW	32	0x0000 00E0	0x4AE0 7DE0
PRM_ABBLDO_GPU_SETUP	RW	32	0x0000 00E4	0x4AE0 7DE4

Table 3-1184. DEVICE_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM Physical Address L4_WKUP Interconnect
PRM_ABBLDO_GPU_CTRL	RW	32	0x0000 00E8	0x4AE0 7DE8
PRM_BANDGAP_SETUP	RW	32	0x0000 00EC	0x4AE0 7DEC
PRM_DEVICE_OFF_CTRL	RW	32	0x0000 00F0	0x4AE0 7DF0
PRM_PHASE1_CNDP	R	32	0x0000 00F4	0x4AE0 7DF4
PRM_PHASE2A_CNDP	R	32	0x0000 00F8	0x4AE0 7DF8
PRM_PHASE2B_CNDP	R	32	0x0000 00FC	0x4AE0 7DFC
RESERVED	R	32	0x0000 0100	0x4AE0 7E00
RESERVED	R	32	0x0000 0110	0x4AE0 7E10
RESERVED	R	32	0x0000 0114	0x4AE0 7E14
PRM_SLDO_DSPEVE_SETUP	RW	32	0x0000 0118	0x4AE0 7E18
PRM_SLDO_IVA_SETUP	RW	32	0x0000 011C	0x4AE0 7E1C
PRM_ABBLDO_DSPEVE_CTRL	RW	32	0x0000 0120	0x4AE0 7E20
PRM_ABBLDO_IVA_CTRL	RW	32	0x0000 0124	0x4AE0 7E24
PRM_SLDO_DSPEVE_CTRL	RW	32	0x0000 0128	0x4AE0 7E28
PRM_SLDO_IVA_CTRL	RW	32	0x0000 012C	0x4AE0 7E2C
PRM_ABBLDO_DSPEVE_SETUP	RW	32	0x0000 0130	0x4AE0 7E30
PRM_ABBLDO_IVA_SETUP	RW	32	0x0000 0134	0x4AE0 7E34

3.13.30.2 DEVICE_PRM Register Description

Table 3-1185. PRM_RSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7D00	**Instance**	DEVICE_PRM
Description	Global software cold and warm reset control. This register is auto-cleared. Only write 1 is possible. A read returns 0 only.		
Type	RW		

31–2	1	0
RESERVED	RST_GLOBAL_COLD_SW	RST_GLOBAL_WARM_SW

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_GLOBAL_COLD_SW	Global COLD software reset control. This bit is reset only upon a global cold source of reset. 0x0: Global COLD software reset is cleared. 0x1: Triggers a global COLD software reset. The software must ensure the SDRAM is properly put in sef-refresh mode before applying this reset.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	RST_GLOBAL_WARM_SW	Global WARM software reset control. This bit is reset upon any global source of reset (warm and cold). 0x0: Global warm software reset is cleared. 0x1: Triggers a global warm software reset.	RW	0x0

Table 3-1186. PRM_RSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7D04	**Instance**	DEVICE_PRM
Description	This register logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31:17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	TSHUT_IVA_RST	TSHUT_DSPEVE_RST	RESERVED	TSHUT_CORE_RST	TSHUT_MM_RST	TSHUT_MPU_RST	RESERVED	ICEPICK_RST	RESERVED	RESERVED	RESERVED	EXTERNAL_WARM_RST	RESERVED	MPU_WDT_RST	RESERVED	GLOBAL_WARM_SW_RST	GLOBAL_COLD_RST

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	TSHUT_IVA_RST	TSHUT_IVA warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MM reset. 0x1: TSHUT_MM reset has occurred.	RW	0x0
15	TSHUT_DSPEVE_RST	TSHUT_DSPEVE warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MM reset. 0x1: TSHUT_MM reset has occurred.	RW	0x0
14	RESERVED		R	0x0
13	TSHUT_CORE_RST	TSHUT_CORE warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_CORE reset. 0x1: TSHUT_CORE reset has occurred.	RW	0x0
12	TSHUT_MM_RST	TSHUT_GPU warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MM reset. 0x1: TSHUT_MM reset has occurred.	RW	0x0
11	TSHUT_MPU_RST	TSHUT_MPU warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MPU reset. 0x1: TSHUT_MPU reset has occurred.	RW	0x0
10	RESERVED		R	0x0
9	ICEPICK_RST	IcePick reset event. This is a source of global warm reset initiated by the emulation. 0x0: No ICEPICK reset. 0x1: IcePick reset has occurred.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	RESERVED		R	0x0
7	RESERVED		R	0x0
6	RESERVED		R	0x0
5	EXTERNAL_WARM_RST	External warm reset event 0x0: No global warm reset. 0x1: Global external warm reset has occurred.	RW	0x0
4	RESERVED		R	0x0
3	MPU_WDT_RST	WD_TIMER2 and MPU subsystem watchdog reset event. This is a source of global WARM reset. 0x0: No reset. 0x1: Reset has occurred.	RW	0x0
2	RESERVED		R	0x0
1	GLOBAL_WARM_SW_RST	Global warm software reset event 0x0: No global warm SW reset 0x1: Global warm SW reset has occurred.	RW	0x0
0	GLOBAL_COLD_RST	Power-on (cold) reset event 0x0: No power-on reset. 0x1: Power-on reset has occurred.	RW	0x1

Table 3-1187. PRM_RSTTIME

Address Offset	0x0000 0008		
Physical Address	0x4AE0 7D08	**Instance**	DEVICE_PRM
Description	Reset duration control. [warm reset insensitive]		
Type	RW		

31–15	14–10	9–0
RESERVED	RSTTIME2	RSTTIME1

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:10	RSTTIME2	Power domain reset duration 2 in number of RM.SYSCLK clock cycles. 0x0: Reserved	RW	0x10
9:0	RSTTIME1	Global reset duration 1 in number of Func_32k_clk clock cycles. This bit-field is only sensitive to the external power-on reset (WKUPAON_SYS_PWRON_RST reset line) 0x0: Reserved	RW	0x6

Table 3-1188. PRM_PSCON_COUNT

Address Offset	0x0000 0018		
Physical Address	0x4AE0 7D18	**Instance**	DEVICE_PRM
Description	This register allows controlling 2 parameters for power state controller. [warm reset insensitive]		
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	HG_PONOUT_2_PGOODIN_TIME	PONOUT_2_PGOODIN_TIME	PCHARGE_TIME

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	HG_PONOUT_2_PGOODIN_TIME	The value 'NbCycles' set in this field determines the duration of the PONOUT to PGOODIN transition for power domain without DPS. The duration is computed as 8 x NbCycles of system clock cycles. Target is 10us.	RW	0x30
15:8	PONOUT_2_PGOODIN_TIME	The value 'NbCycles' set in this field determines the duration of the PONOUT to PGOODIN transition for power domain without DPS. The duration is computed as 8 x NbCycles of system clock cycles. Target is 10us.	RW	0x30
7:0	PCHARGE_TIME	Number of system clock cycles for the SRAM pre-charge duration. Target is 600ns.	RW	0x17

Table 3-1189. PRM_IO_COUNT

Address Offset	0x0000 001C		
Physical Address	0x4AE0 7D1C	**Instance**	DEVICE_PRM
Description	This register allows controlling DDR IO isolation removal setup. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ISO_2_ON_TIME							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ISO_2_ON_TIME	Determines the setup time of the DDR IOs going out of isolation. Counting on the system clock. Target is 1.5us.	RW	0x3a

Table 3-1190. PRM_IO_PMCTRL

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7D20	**Instance**	DEVICE_PRM
Description	This register allows controlling power management features of the IOs.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															GLOBAL_WUEN	RESERVED						WUCLK_STATUS	WUCLK_CTRL	RESERVED		IO_ON_STATUS	ISOOVR_EXTEND	RESERVED		ISOCLK_STATUS	ISOCLK_OVERRIDE

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	GLOBAL_WUEN	Global IO wakeup enable. This is a gating condition to all individual IO WUEN coming from control module. Gating is done in the Spinner logic. 0x0: All individual IO WUEN are gated in the Spinner logic (overriden to 0). 0x1: All individual IO WUEN from control module are going to IOs.	RW	0x0
15:10	RESERVED		R	0x0
9	WUCLK_STATUS	Gives value of WUCLKOUT signal coming back from IO pad ring.	R	0x0

Bits	Field Name	Description	Type	Reset
8	WUCLK_CTRL	Direct control on WUCLKIN signal to IO pad ring. 0x0: WUCLKIN signal is driven to 0. IO wakeup daisy chain is functional as well as IO whose wakeup feature is enabled. 0x1: WUCLKIN signal is driven to 1. IO wakeup daisy chain is reset and is latching current pad states and WUEN inputs.	RW	0x0
7:6	RESERVED		R	0x0
5	IO_ON_STATUS	Gives the functional status of the IO ring. 0x0: Part or all of the IOs are not in the ON state, that is are in isolation state. 0x1: All IOs are in the ON state.	R	0x1
4	ISOOVR_EXTEND	Control non-EMIF IO isolation extension upon a device wakeup from OFF mode. 0x0: Non-EMIF IO isolation is not extended. 'EMIF_ON' IO transition happens as soon as automatic restore is completed. 0x1: Non-EMIF IO isolation is extended. 'EMIF_ON' IO transition is stalled.	RW	0x0
3:2	RESERVED		R	0x0
1	ISOCLK_STATUS	Gives value of ISOCLKOUT signal coming back from IO pad ring.	R	0x0
0	ISOCLK_OVERRIDE	Override control on ISOCLKIN signal to IO pad ring. Used at boot time when it is needed to change the mode of an IO from 1.8V default mode to 1.2V mode. When not overriden, this signal is controlled by hardware only. 0x0: ISOCLKIN signal is not overriden. 0x1: ISOCLKIN signal is overriden to active value ('1').	RW	0x0

Table 3-1191. PRM_SRAM_COUNT

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 7DBC	**Instance**	DEVICE_PRM
Description	Common setup for SRAM LDO transition counters. Applies to all voltage domains. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STARTUP_COUNT								SLPCNT_VALUE								VSETUPCNT_VALUE								RESERVED		PCHARGECNT_VALUE					

Bits	Field Name	Description	Type	Reset
31:24	STARTUP_COUNT	Determines the start-up duration of SRAM and ABB LDO. The duration is computed as 16 x NbCycles of system clock cycles. Target is 50us.	RW	0x78
23:16	SLPCNT_VALUE	Delay between retention/off assertion of last SRAM bank and SRAMALLRET signal to LDO is driven high. Counting on system clock. Target is 2us.	RW	0x0
15:8	VSETUPCNT_VALUE	SRAM LDO rampup time from retention to active mode. The duration is computed as 8 x NbCycles of system clock cycles. Target is 30us.	RW	0x0
7:6	RESERVED		R	0x0
5:0	PCHARGECNT_VALUE	Delay between de-assertion of standby_rta_ret_on and standby_rta_ret_good. Counting on system clock. Target is 600ns.	RW	0x17

Table 3-1192. PRM_SLDO_CORE_SETUP

Address Offset	0x0000 00C4		
Physical Address	0x4AE0 7DC4	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	RW		

31:9	8	7	6	5	4	3	2	1	0
RESERVED	AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0

Bits	Field Name	Description	Type	Reset
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1193. PRM_SLDO_CORE_CTRL

Address Offset	0x0000 00C8		
Physical Address	0x4AE0 7DC8	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	R		

31:10	9	8	7:1	0
RESERVED	SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED	RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode	R	0x0

Table 3-1194. PRM_SLDO_MPU_SETUP

Address Offset	0x0000 00CC		
Physical Address	0x4AE0 7DCC	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for MPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0

Bits	Field Name	Description	Type	Reset
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1195. PRM_SLDO_MPU_CTRL

Address Offset	0x0000 00D0		
Physical Address	0x4AE0 7DD0	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for MPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED							RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1196. PRM_SLDO_GPU_SETUP

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 7DD4	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for GPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1197. PRM_SLDO_GPU_CTRL

Address Offset	0x0000 00D8		
Physical Address	0x4AE0 7DD8	**Instance**	DEVICE_PRM

Table 3-1197. PRM_SLDO_GPU_CTRL (continued)

Description	Control and status of the SRAM LDO for GPU voltage domain. [warm reset insensitive]
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td>SRAM_IN_TRANSITION</td><td>SRAMLDO_STATUS</td><td colspan="7">RESERVED</td><td>RETMODE_ENABLE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1198. PRM_ABBLDO_MPU_SETUP

Address Offset	0x0000 00DC		
Physical Address	0x4AE0 7DDC	**Instance**	DEVICE_PRM
Description	Selects the MPU_ABB LDO mode.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="8">SR2_WTCNT_VALUE</td><td colspan="3">RESERVED</td><td>RESERVED</td><td>RESERVED</td><td>ACTIVE_FBB_SEL</td><td>RESERVED</td><td>SR2EN</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	RESERVED		R	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

Table 3-1199. PRM_ABBLDO_MPU_CTRL

Address Offset	0x0000 00E0		
Physical Address	0x4AE0 7DE0	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on MPU voltage domain. [warm reset insensitive]		
Type	RW		

31:7	6	5	4:3	2	1:0
RESERVED	SR2_IN_TRANSITION	RESERVED	SR2_STATUS	OPP_CHANGE	OPP_SEL

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	To control the ABB LDO (FBB/RBB) at a given OPP, set to 0x1. Refer to section ABB LDO Programming sequence. 0x0: default : Nominal 0x1: Fast OPP	RW	0x0

Table 3-1200. PRM_ABBLDO_GPU_SETUP

Address Offset	0x0000 00E4		
Physical Address	0x4AE0 7DE4	**Instance**	DEVICE_PRM
Description	Selects the GPU_ABB LDO mode.		
Type	RW		

31-16	15-8	7-5	4	3	2	1	0
RESERVED	SR2_WTCNT_VALUE	RESERVED	RESERVED	RESERVED	ACTIVE_FBB_SEL	RESERVED	SR2EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	RESERVED		R	0x0
3	RESERVED		R	0x0
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

Table 3-1201. PRM_ABBLDO_GPU_CTRL

Address Offset	0x0000 00E8		
Physical Address	0x4AE0 7DE8	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on GPU voltage domain. [warm reset insensitive]		
Type	RW		

31-7	6	5	4-3	2	1-0
RESERVED	SR2_IN_TRANSITION	RESERVED	SR2_STATUS	OPP_CHANGE	OPP_SEL

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	To control the ABB LDO (FBB/RBB) at a given OPP, set to 0x1. Refer to section ABB LDO Programming sequence. 0x0: default : Nominal 0x1: Fast OPP	RW	0x0

Table 3-1202. PRM_BANDGAP_SETUP

Address Offset	0x0000 00EC		
Physical Address	0x4AE0 7DEC	**Instance**	DEVICE_PRM
Description	Setup of the bandgap. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STARTUP_COUNT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	STARTUP_COUNT	Determines the start-up duration of BANDGAP. The duration is computed as 32 x NbCycles of system clock cycles. Target is 100us.	RW	0x78

Table 3-1203. PRM_DEVICE_OFF_CTRL

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 7DF0	**Instance**	DEVICE_PRM
Description	This register is used to control device OFF transition.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:10	9	8	7:1	0
RESERVED	EMIF2_OFFWKUP_DISABLE	EMIF1_OFFWKUP_DISABLE	RESERVED	DEVICE_OFF_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	EMIF2_OFFWKUP_DISABLE	Controls the EMIF2_DEVICE_OFFWKUP_CORESRTACTST notifier sent to EMIF1 upon a device wakeup from OFF mode. [warm reset insensitive] 0x0: Notifier is activated. 0x1: Notifier is not activated - stays low	RW	0x0
8	EMIF1_OFFWKUP_DISABLE	Controls the EMIF1_DEVICE_OFFWKUP_CORESRTACTST notifier sent to EMIF2 upon a device wakeup from OFF mode. [warm reset insensitive] 0x0: Notifier is activated. 0x1: Notifier is not activated - stays low	RW	0x0
7:1	RESERVED		R	0x0
0	DEVICE_OFF_ENABLE	Controls transition to device OFF mode. 0x0: Device is not allowed to perform transition to OFF mode 0x1: Device is allowed to perform transition to OFF mode as soon as all power domains in MPU, MM and CORE voltage are in OFF or OSWRET state (open switch retention)	RW	0x0

Table 3-1204. PRM_PHASE1_CNDP

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 7DF4	**Instance**	DEVICE_PRM
Description	This register stores the start descriptor address of automatic restore phase1. [warm reset insensitive] **NOTE: This register is NOT supported on this device.**		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE1_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE1_CNDP	Start descriptor address of automatic restore phase1. Hard-coded to SAR_ROM base address.	R	0x4a05e000

Table 3-1205. PRM_PHASE2A_CNDP

Address Offset	0x0000 00F8		
Physical Address	0x4AE0 7DF8	**Instance**	DEVICE_PRM
Description	This register stores the start descriptor address of automatic restore phase2A. [warm reset insensitive] **NOTE: This register is NOT supported on this device.**		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE2A_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE2A_CNDP	Start descriptor address of automatic restore phase2A. Hard-coded to SAR_ROM base address + 0x30.	R	0x4a05e030

Table 3-1206. PRM_PHASE2B_CNDP

Address Offset	0x0000 00FC		
Physical Address	0x4AE0 7DFC	**Instance**	DEVICE_PRM
Description	This register stores the start descriptor address of automatic restore phase2B. [warm reset insensitive] **NOTE: This register is NOT supported on this device.**		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE2B_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE2B_CNDP	Start descriptor address of automatic restore phase2B. Hard-coded to SAR_ROM base address + 0x60.	R	0x4a05e060

Table 3-1207. PRM_SLDO_DSPEVE_SETUP

Address Offset	0x0000 0118		
Physical Address	0x4AE0 7E18	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for DSPEVE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1208. PRM_SLDO_IVA_SETUP

Address Offset	0x0000 011C		
Physical Address	0x4AE0 7E1C	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for IVA voltage domain. [warm reset insensitive]		
Type	RW		

31–9	8	7	6	5	4	3	2	1	0
RESERVED	AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1209. PRM_ABBLDO_DSPEVE_CTRL

Address Offset	0x0000 0120		
Physical Address	0x4AE0 7E20	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on DSPEVE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SR2_IN_TRANSITION	RESERVED	SR2_STATUS	OPP_CHANGE	OPP_SEL

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	To control the ABB LDO (FBB/RBB) at a given OPP, set to 0x1. Refer to section ABB LDO Programming sequence. 0x0: default : Nominal 0x1: Fast OPP	RW	0x0

Table 3-1210. PRM_ABBLDO_IVA_CTRL

Address Offset	0x0000 0124		
Physical Address	0x4AE0 7E24	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on IVA voltage domain. [warm reset insensitive]		
Type	RW		

31–7	6	5	4:3	2	1:0
RESERVED	SR2_IN_TRANSITION	RESERVED	SR2_STATUS	OPP_CHANGE	OPP_SEL

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	To control the ABB LDO (FBB/RBB) at a given OPP, set to 0x1. Refer to section ABB LDO Programming sequence. 0x0: default : Nominal 0x1: Fast OPP	RW	0x0

Table 3-1211. PRM_SLDO_DSPEVE_CTRL

Address Offset	0x0000 0128		
Physical Address	0x4AE0 7E28	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	RW		

31:10	9	8	7:1	0
RESERVED	SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED	RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1212. PRM_SLDO_IVA_CTRL

Address Offset	0x0000 012C		
Physical Address	0x4AE0 7E2C	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED							RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1213. PRM_ABBLDO_DSPEVE_SETUP

Address Offset	0x0000 0130		
Physical Address	0x4AE0 7E30	**Instance**	DEVICE_PRM
Description	Selects the GPU_ABB LDO mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_WTCNT_VALUE								RESERVED			RESERVED	RESERVED	ACTIVE_FBB_SEL	RESERVED	SR2EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	RESERVED		R	0x0
3	RESERVED		R	0x0
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

Table 3-1214. PRM_ABBLDO_IVA_SETUP

Address Offset	0x0000 0134		
Physical Address	0x4AE0 7E34	**Instance**	DEVICE_PRM
Description	Selects the GPU_ABB LDO mode.		
Type	RW		

31:16	15:8	7:5	4	3	2	1	0
RESERVED	SR2_WTCNT_VALUE	RESERVED	RESERVED	RESERVED	ACTIVE_FBB_SEL	RESERVED	SR2EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	RESERVED		R	0x0
3	RESERVED		R	0x0
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

3.13.31 DSP1_PRM Registers

3.13.31.1 DSP1_PRM Register Summary

Table 3-1215. DSP1_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_PRM Physical Address L4_WKUP Interconnect
PM_DSP1_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6400
PM_DSP1_PWRSTST	RW	32	0x0000 0004	0x4AE0 6404
RM_DSP1_RSTCTRL	RW	32	0x0000 0010	0x4AE0 6410
RM_DSP1_RSTST	RW	32	0x0000 0014	0x4AE0 6414
RM_DSP1_DSP1_CONTEXT	RW	32	0x0000 0024	0x4AE0 6424

3.13.31.2 DSP1_PRM Register Description

Table 3-1216. PM_DSP1_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6400	**Instance**	DSP1_PRM
Description	This register controls the DSP power state to reach upon a domain sleep transition		
Type	RW		

31:22	21:20	19:18	17:16	15:5	4	3:2	1:0
RESERVED	DSP1_EDMA_ONSTATE	DSP1_L2_ONSTATE	DSP1_L1_ONSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	DSP1_EDMA_ONSTATE	DSP_EDMA state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	DSP1_L2_ONSTATE	DSP_L2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	DSP1_L1_ONSTATE	DSP_L1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1217. PM_DSP1_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6404	**Instance**	DSP1_PRM
Description	This register provides a status on the DSP domain current power state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:10	9:8	7:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	DSP1_EDMA_STATEST	DSP1_L2_STATEST	DSP1_L1_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	DSP1_EDMA_STATEST	DSP_EDMA memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	DSP1_L2_STATEST	DSP_L2 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	DSP1_L1_STATEST	DSP_L1 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1218. RM_DSP1_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6410	**Instance**	DSP1_PRM
Description	This register controls the release of the DSP sub-system resets.		
Type	RW		

31–2	1	0
RESERVED	RST_DSP1	RST_DSP1_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_DSP1	DSP reset control 0x0: Reset is cleared for the MMU, cache and slave interface 0x1: Reset is asserted for the MMU, cache and slave interface	RW	0x1
0	RST_DSP1_LRST	DSP Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1219. RM_DSP1_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6414	**Instance**	DSP1_PRM
Description	This register logs the different reset sources of the DSP domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31–4	3	2	1	0
RESERVED	RST_DSP1_EMU_REQ	RST_DSP1_EMU	RST_DSP1	RST_DSP1_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_DSP1_EMU_REQ	DSP processor has been reset due to DSP emulation reset request driven from DSP-SS 0x0: No emulation reset 0x1: DSP DSP has been reset upon emulation reset request	RW	0x0
2	RST_DSP1_EMU	DSP domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: DSP has been reset upon emulation reset	RW	0x0
1	RST_DSP1	DSP SW reset status 0x0: No SW reset occurred 0x1: MMU, cache and slave interface has been reset upon SW reset	RW	0x0
0	RST_DSP1_LRST	DSP Local SW reset 0x0: No SW reset occurred 0x1: DSP has been reset upon SW reset	RW	0x0

Table 3-1220. RM_DSP1_DSP1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6424	**Instance**	DSP1_PRM
Description	This register contains dedicated DSP context statuses. [warm reset insensitive]		
Type	RW		

31:11	10	9	8	7:1	0
RESERVED	LOSTMEM_DSP_EDMA	LOSTMEM_DSP_L2	LOSTMEM_DSP_L1	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LOSTMEM_DSP_EDMA	Specify if memory-based context in DSP_EDMA memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
9	LOSTMEM_DSP_L2	Specify if memory-based context in DSP_L2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_DSP_L1	Specify if memory-based context in DSP_L1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSP_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.13.32 DSP2_PRM Registers

3.13.32.1 DSP2_PRM Register Summary

Table 3-1221. DSP2_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_PRM Physical Address L4_WKUP Interconnect
PM_DSP2_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7B00
PM_DSP2_PWRSTST	RW	32	0x0000 0004	0x4AE0 7B04
RM_DSP2_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7B10
RM_DSP2_RSTST	RW	32	0x0000 0014	0x4AE0 7B14
RM_DSP2_DSP2_CONTEXT	RW	32	0x0000 0024	0x4AE0 7B24

3.13.32.2 DSP2_PRM Register Description

Table 3-1222. PM_DSP2_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7B00	**Instance**	DSP2_PRM
Description	This register controls the DSP power state to reach upon a domain sleep transition		
Type	RW		

31:22	21:20	19:18	17:16	15:5	4	3:2	1:0
RESERVED	DSP2_EDMA_ONSTATE	DSP2_L2_ONSTATE	DSP2_L1_ONSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	DSP2_EDMA_ONSTATE	DSP_EDMA state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	DSP2_L2_ONSTATE	DSP_L2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	DSP2_L1_ONSTATE	DSP_L1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1223. PM_DSP2_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7B04	**Instance**	DSP2_PRM
Description	This register provides a status on the DSP domain current power state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:10	9:8	7:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	DSP2_EDMA_STATEST	DSP2_L2_STATEST	DSP2_L1_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	DSP2_EDMA_STATEST	DSP_EDMA memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
7:6	DSP2_L2_STATEST	DSP_L2 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	DSP2_L1_STATEST	DSP_L1 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1224. RM_DSP2_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7B10	**Instance**	DSP2_PRM
Description	This register controls the release of the DSP sub-system resets.		
Type	RW		

31:2	1	0
RESERVED	RST_DSP2	RST_DSP2_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_DSP2	DSP SW reset control 0x0: Reset is cleared for the MMU, cache and slave interface 0x1: Reset is asserted for the MMU, cache and slave interface	RW	0x1
0	RST_DSP2_LRST	DSP Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1225. RM_DSP2_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7B14	**Instance**	DSP2_PRM
Description	This register logs the different reset sources of the DSP domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		

Table 3-1225. RM_DSP2_RSTST (continued)

Type	RW

31:4	3	2	1	0
RESERVED	RST_DSP2_EMU_REQ	RST_DSP2_EMU	RST_DSP2	RST_DSP2_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_DSP2_EMU_REQ	DSP processor has been reset due to DSP emulation reset request driven from DSP-SS 0x0: No emulation reset 0x1: DSP DSP has been reset upon emulation reset request	RW	0x0
2	RST_DSP2_EMU	DSP domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: DSP has been reset upon emulation reset	RW	0x0
1	RST_DSP2	DSP SW reset status 0x0: No SW reset occurred 0x1: MMU, cache and slave interface has been reset upon SW reset	RW	0x0
0	RST_DSP2_LRST	DSP Local SW reset 0x0: No SW reset occurred 0x1: DSP has been reset upon SW reset	RW	0x0

Table 3-1226. RM_DSP2_DSP2_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7B24	**Instance**	DSP2_PRM
Description	This register contains dedicated DSP context statuses. [warm reset insensitive]		
Type	RW		

31:11	10	9	8	7:1	0
RESERVED	LOSTMEM_DSP_EDMA	LOSTMEM_DSP_L2	LOSTMEM_DSP_L1	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10	LOSTMEM_DSP_EDMA	Specify if memory-based context in DSP_EDMA memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
9	LOSTMEM_DSP_L2	Specify if memory-based context in DSP_L2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_DSP_L1	Specify if memory-based context in DSP_L1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSP_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.13.33 DSS_PRM Registers

3.13.33.1 DSS_PRM Register Summary

Table 3-1227. DSS_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSS_PRM Physical Address L4_WKUP Interconnect
PM_DSS_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7100
PM_DSS_PWRSTST	RW	32	0x0000 0004	0x4AE0 7104
PM_DSS_DSS_WKDEP	RW	32	0x0000 0020	0x4AE0 7120
RM_DSS_DSS_CONTEXT	RW	32	0x0000 0024	0x4AE0 7124
PM_DSS_DSS2_WKDEP	RW	32	0x0000 0028	0x4AE0 7128
RM_DSS_BB2D_CONTEXT	RW	32	0x0000 0034	0x4AE0 7134
RESERVED	RW	32	0x0000 003C	0x4AE0 713C

3.13.33.2 DSS_PRM Register Description

Table 3-1228. PM_DSS_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7100	**Instance**	DSS_PRM
Description	This register controls the DSS power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DSS_MEM_ONSTATE	RESERVED	DSS_MEM_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	DSS_MEM_ONSTATE	DSS_MEM state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:9	RESERVED		R	0x0
8	DSS_MEM_RETSTATE	Note: Not supported on this device.	R	0x0
7:5	RESERVED			
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Note: Not supported on this device.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x0

Table 3-1229. Register Call Summary for Register PM_DSS_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Mode Control and Status: [0] [1] [2] [3] [4]

PRCM Register Manual

- DSS_PRM Register Summary: [5]

Table 3-1230. PM_DSS_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7104	**Instance**	DSS_PRM
Description	This register provides a status on the current DSS power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	DSS_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	DSS_MEM_STATEST	DSS_MEM state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1231. Register Call Summary for Register PM_DSS_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Mode Control and Status: [0] [1] [2] [3] [4]

PRCM Register Manual

- DSS_PRM Register Summary: [5]

Table 3-1232. PM_DSS_DSS_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7120	**Instance**	DSS_PRM
Description	This register controls wakeup dependency based on DSS service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_DSI1_B_EVE2	WKUPDEP_DSI1_B_EVE1	WKUPDEP_DSI1_B_DSP2	WKUPDEP_DSI1_B_IPU1	WKUPDEP_DSI1_B_SDMA	WKUPDEP_DSI1_B_DSP1	WKUPDEP_DSI1_B_IPU2	WKUPDEP_DSI1_B_MPU	RESERVED	RESERVED	WKUPDEP_DSI1_A_EVE2	WKUPDEP_DSI1_A_EVE1	WKUPDEP_DSI1_A_DSP2	WKUPDEP_DSI1_A_IPU1	WKUPDEP_DSI1_A_SDMA	WKUPDEP_DSI1_A_DSP1	WKUPDEP_DSI1_A_IPU2	WKUPDEP_DSI1_A_MPU	RESERVED	RESERVED	WKUPDEP_DISPC_EVE2	WKUPDEP_DISPC_EVE1	WKUPDEP_DISPC_DSP2	WKUPDEP_DISPC_IPU1	WKUPDEP_DISPC_SDMA	WKUPDEP_DISPC_DSP1	WKUPDEP_DISPC_IPU2	WKUPDEP_DISPC_MPU

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	RESERVED		R	0x0
28	RESERVED		R	0x0
27	WKUPDEP_DSI1_B_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	WKUPDEP_DSI1_B_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	WKUPDEP_DSI1_B_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	WKUPDEP_DSI1_B_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	WKUPDEP_DSI1_B_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	WKUPDEP_DSI1_B_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	WKUPDEP_DSI1_B_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	WKUPDEP_DSI1_B_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_DSI1_A_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_DSI1_A_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_DSI1_A_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_DSI1_A_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	WKUPDEP_DSI1_A_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	WKUPDEP_DSI1_A_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_DSI1_A_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_DSI1_A_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_DISPC_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_DISPC_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DISPC_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DISPC_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_DISPC_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_DISPC_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DISPC_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DISPC_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1233. Register Call Summary for Register PM_DSS_DSS_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [10] [11] [12] [13] [14] [15] [16] [17] [20] [21] [22] [23] [24] [25] [26] [27]

PRCM Register Manual

- DSS_PRM Register Summary: [30]

Table 3-1234. RM_DSS_DSS_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7124	**Instance**	DSS_PRM
Description	This register contains dedicated DSS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_DSS_MEM	RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_DSS_MEM	Specify if memory-based context in DSS_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1235. Register Call Summary for Register RM_DSS_DSS_CONTEXT

Power Management Functional Description

- PD_DSS Description: [0] [1]

PRCM Register Manual

- DSS_PRM Register Summary: [2]

Table 3-1236. PM_DSS_DSS2_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7128	**Instance**	DSS_PRM
Description	This register controls wakeup dependency based on DSS service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						WKUPDEP_HDMIDMA_DSP2	RESERVED	WKUPDEP_HDMIDMA_SDMA	WKUPDEP_HDMIDMA_DSP1	RESERVED		RESERVED	RESERVED	WKUPDEP_DSI1_C_EVE2	WKUPDEP_DSI1_C_EVE1	WKUPDEP_DSI1_C_DSP2	WKUPDEP_DSI1_C_IPU1	WKUPDEP_DSI1_C_SDMA	WKUPDEP_DSI1_C_DSP1	WKUPDEP_DSI1_C_IPU2	WKUPDEP_DSI1_C_MPU	RESERVED	RESERVED	WKUPDEP_HDMIIRQ_EVE2	WKUPDEP_HDMIIRQ_EVE1	WKUPDEP_HDMIIRQ_DSP2	WKUPDEP_HDMIIRQ_IPU1	RESERVED	WKUPDEP_HDMIIRQ_DSP1	WKUPDEP_HDMIIRQ_IPU2	WKUPDEP_HDMIIRQ_MPU

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	WKUPDEP_HDMIDMA_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	RESERVED		R	0x0
23	WKUPDEP_HDMIDMA_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	WKUPDEP_HDMIDMA_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_DSI1_C_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_DSI1_C_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_DSI1_C_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_DSI1_C_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	WKUPDEP_DSI1_C_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	WKUPDEP_DSI1_C_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
11	WKUPDEP_DSI1_C_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_DSI1_C_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_HDMIIRQ_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_HDMIIRQ_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_HDMIIRQ_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_HDMIIRQ_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_HDMIIRQ_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_HDMIIRQ_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_HDMIIRQ_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1237. Register Call Summary for Register PM_DSS_DSS2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [12] [13] [14] [15] [16] [17] [18] [19]

Table 3-1237. Register Call Summary for Register PM_DSS_DSS2_WKDEP (continued)

PRCM Register Manual

- DSS_PRM Register Summary: [22]

Table 3-1238. RM_DSS_BB2D_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7134	**Instance**	DSS_PRM
Description	This register contains dedicated BB2B context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_DSS_MEM	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_DSS_MEM	Specify if memory-based context in DSS_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: MAINTAINED 0x1: LOST	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RST signal) 0x0: MAINTAINED 0x1: LOST	RW	0x1

Table 3-1239. Register Call Summary for Register RM_DSS_BB2D_CONTEXT

Power Management Functional Description

- PD_DSS Description: [0]

PRCM Register Manual

- DSS_PRM Register Summary: [1]

3.13.34 EMU_CM Registers

3.13.34.1 EMU_CM Register Summary

Table 3-1240. EMU_CM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMU_CM Physical Address L4_WKUP Interconnect
CM_EMU_CLKSTCTRL	RW	32	0x0000 0000	0x4AE0 7A00
CM_EMU_DEBUGSS_CLKCTRL	R	32	0x0000 0004	0x4AE0 7A04
CM_EMU_DYNAMICDEP	RW	32	0x0000 0008	0x4AE0 7A08
CM_EMU_MPU_EMU_DBG_CLKCTRL	R	32	0x0000 000C	0x4AE0 7A0C

3.13.34.2 EMU_CM Register Description

Table 3-1241. CM_EMU_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7A00	**Instance**	EMU_CM
Description	This register enables the EMU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1:0
RESERVED	CLKACTIVITY_EMU_SYS_CLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EMU_SYS_CLK	This field indicates the state of the EMU_SYS_CLK clock in the domain. 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the EMU clock domain. 0x0: Reserved 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x2

Table 3-1242. Register Call Summary for Register CM_EMU_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1]

PRCM Register Manual

- EMU_CM Register Summary: [2]

Table 3-1243. CM_EMU_DEBUGSS_CLKCTRL

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7A04	**Instance**	EMU_CM
Description	This register manages the DEBUGSS clocks. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STBYST	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1244. Register Call Summary for Register CM_EMU_DEBUGSS_CLKCTRL

PRCM Register Manual

- EMU_CM Register Summary: [0]

Table 3-1245. CM_EMU_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4AE0 7A08	**Instance**	EMU_CM
Description	This register controls the dynamic domain depedencies from EMU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31:28	27:24	23:6	5	4:0
RESERVED	WINDOWSIZE	RESERVED	L3MAIN1_DYNDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1

Bits	Field Name	Description	Type	Reset
4:0	RESERVED		R	0x0

Table 3-1246. Register Call Summary for Register CM_EMU_DYNAMICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0]

PRCM Register Manual

- EMU_CM Register Summary: [1]

Table 3-1247. CM_EMU_MPU_EMU_DBG_CLKCTRL

Address Offset	0x0000 000C		
Physical Address	0x4AE0 7A0C	**Instance**	EMU_CM
Description	This register manages the MPU_EMU_DBG clocks. [warm reset insensitive]		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1248. Register Call Summary for Register CM_EMU_MPU_EMU_DBG_CLKCTRL

PRCM Register Manual

- EMU_CM Register Summary: [0]

3.13.35 EMU_PRM Registers

3.13.35.1 EMU_PRM Register Summary

Table 3-1249. EMU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMU_PRM Physical Address L4_WKUP Interconnect
PM_EMU_PWRSTCTRL	R	32	0x0000 0000	0x4AE0 7900
PM_EMU_PWRSTST	RW	32	0x0000 0004	0x4AE0 7904
RM_EMU_DEBUGSS_CONTEXT	RW	32	0x0000 0024	0x4AE0 7924

3.13.35.2 EMU_PRM Register Description

Table 3-1250. PM_EMU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7900	**Instance**	EMU_PRM
Description	This register controls the EMU power state to reach upon a domain sleep transition		
Type	R		

31:18	17:16	15:2	1:0
RESERVED	EMU_BANK_ONSTATE	RESERVED	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EMU_BANK_ONSTATE	EMU memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state	R	0x0

Table 3-1251. Register Call Summary for Register PM_EMU_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1]

PRCM Register Manual

- EMU_PRM Register Summary: [2]

Table 3-1252. PM_EMU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7904	**Instance**	EMU_PRM
Description	This register provides a status on the EMU domain current power state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	EMU_BANKSTATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	EMU_BANK_STATEST	EMU memory bank state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON It is supplied by WKUP LDO	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1253. Register Call Summary for Register PM_EMU_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4]

PRCM Register Manual

- EMU_PRM Register Summary: [5]

Table 3-1254. RM_EMU_DEBUGSS_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7924	**Instance**	EMU_PRM
Description	This register contains dedicated DEBUGSS context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_EMU_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EMU_BANK	Specify if memory-based context in EMU_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EMU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1255. Register Call Summary for Register RM_EMU_DEBUGSS_CONTEXT

Power Management Functional Description

- PD_EMU Description: [0]

PRCM Register Manual

- EMU_PRM Register Summary: [1]

3.13.36 EVE1_PRM Registers

3.13.36.1 EVE1_PRM Register Summary

Table 3-1256. EVE1_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_PRM Physical Address L4_WKUP Interconnect
PM_EVE1_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7B40
PM_EVE1_PWRSTST	RW	32	0x0000 0004	0x4AE0 7B44
RM_EVE1_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7B50
RM_EVE1_RSTST	RW	32	0x0000 0014	0x4AE0 7B54
PM_EVE1_EVE1_WKDEP	RW	32	0x0000 0020	0x4AE0 7B60
RM_EVE1_EVE1_CONTEXT	RW	32	0x0000 0024	0x4AE0 7B64

3.13.36.2 EVE1_PRM Register Description

Table 3-1257. PM_EVE1_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7B40	**Instance**	EVE1_PRM
Description	This register controls the EVE power state to reach upon a domain sleep transition		
Type	RW		

31:18	17:16	15:5	4	3:2	1:0
RESERVED	EVE1_BANKONSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EVE1_BANK_ONSTATE	EVE1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1258. PM_EVE1_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7B44	**Instance**	EVE1_PRM
Description	This register provides a status on the EVE domain current power state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:10	9:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	RESERVED	EVE1_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:6	RESERVED		R	0x0
5:4	EVE1_BANK_STATEST	EVE0 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1259. RM_EVE1_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7B50	**Instance**	EVE1_PRM
Description	This register controls the release of the EVE sub-system resets.		
Type	RW		

31:2	1	0
RESERVED	RST_EVE1	RST_EVE1_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_EVE1	EVE reset control 0x0: Reset is cleared 0x1: Local Reset is asserted	RW	0x1
0	RST_EVE1_LRST	EVE Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1260. RM_EVE1_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7B54	**Instance**	EVE1_PRM
Description	This register logs the different reset sources of the EVE domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31:4	3	2	1	0
RESERVED	RST_EVE1_EMU_REQ	RST_EVE1_EMU	RST_EVE1	RST_EVE1_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3	RST_EVE1_EMU_REQ	EVE1 processor has been reset due to EVE emulation reset request driven from EVE1-SS 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset request	RW	0x0
2	RST_EVE1_EMU	EVE1 domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset	RW	0x0
1	RST_EVE1	EVE0 SW reset status 0x0: No SW reset occurred 0x1: Local reset upon SW reset	RW	0x0
0	RST_EVE1_LRST	EVE0 Local SW reset 0x0: No SW reset occurred 0x1: EVE has been reset upon SW reset	RW	0x0

Table 3-1261. PM_EVE1_EVE1_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7B60	**Instance**	EVE1_PRM
Description	This register controls wakeup dependency based on EVE1 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_EVE1_EVE2	RESERVED	WKUPDEP_EVE1_DSP2	WKUPDEP_EVE1_IPU1	WKUPDEP_EVE1_SDMA	WKUPDEP_EVE1_DSP1	WKUPDEP_EVE1_IPU2	WKUPDEP_EVE1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_EVE1_EVE2	Wakeup dependency from EVE1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	WKUPDEP_EVE1_DSP2	Wakeup dependency from EVE1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_EVE1_IPU1	Wakeup dependency from EVE1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_EVE1_SDMA	Wakeup dependency from EVE1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_EVE1_DSP1	Wakeup dependency from EVE1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_EVE1_IPU2	Wakeup dependency from EVE1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_EVE1_MPU	Wakeup dependency from EVE1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1262. RM_EVE1_EVE1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7B64	**Instance**	EVE1_PRM
Description	This register contains dedicated EVE context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_EVE_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EVE_BANK	Specify if memory-based context in EVE1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EVE0_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.13.37 EVE2_PRM Registers

3.13.37.1 EVE2_PRM Register Summary

Table 3-1263. EVE2_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE2_PRM Physical Address L4_WKUP Interconnect
PM_EVE2_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7B80
PM_EVE2_PWRSTST	RW	32	0x0000 0004	0x4AE0 7B84
RM_EVE2_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7B90
RM_EVE2_RSTST	RW	32	0x0000 0014	0x4AE0 7B94
PM_EVE2_EVE2_WKDEP	RW	32	0x0000 0020	0x4AE0 7BA0
RM_EVE2_EVE2_CONTEXT	RW	32	0x0000 0024	0x4AE0 7BA4

3.13.37.2 EVE2_PRM Register Description

Table 3-1264. PM_EVE2_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7B80	**Instance**	EVE2_PRM
Description	This register controls the EVE power state to reach upon a domain sleep transition		
Type	RW		

31:18	17:16	15:5	4	3:2	1:0
RESERVED	EVE2_BANK_ONSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EVE2_BANK_ONSTATE	EVE2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1265. PM_EVE2_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7B84	**Instance**	EVE2_PRM
Description	This register provides a status on the EVE domain current power state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	EVE2_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	EVE2_BANK_STATEST	EVE2 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1266. RM_EVE2_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7B90	**Instance**	EVE2_PRM
Description	This register controls the release of the EVE sub-system resets.		
Type	RW		

31:2	1	0
RESERVED	RST_EVE2	RST_EVE2_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_EVE2	EVE SW reset control 0x0: Reset is cleared 0x1: Local Reset is asserted	RW	0x1
0	RST_EVE2_LRST	EVE Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1267. RM_EVE2_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7B94	**Instance**	EVE2_PRM
Description	This register logs the different reset sources of the EVE domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31:4	3	2	1	0
RESERVED	RST_EVE2_EMU_REQ	RST_EVE2_EMU	RST_EVE2	RST_EVE2_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_EVE2_EMU_REQ	EVE2 processor has been reset due to EVE emulation reset request driven from EVE2-SS 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset request	RW	0x0
2	RST_EVE2_EMU	EVE2 domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset	RW	0x0

Bits	Field Name	Description	Type	Reset
1	RST_EVE2	EVE SW reset status 0x0: No SW reset occurred 0x1: Local reset upon SW reset	RW	0x0
0	RST_EVE2_LRST	EVE Local SW reset 0x0: No SW reset occurred 0x1: EVE has been reset upon SW reset	RW	0x0

Table 3-1268. PM_EVE2_EVE2_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7BA0	**Instance**	EVE2_PRM
Description	This register controls wakeup dependency based on EVE2 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	WKUPDEP_EVE2_EVE1	WKUPDEP_EVE2_DSP2	WKUPDEP_EVE2_IPU1	WKUPDEP_EVE2_SDMA	WKUPDEP_EVE2_DSP1	WKUPDEP_EVE2_IPU2	WKUPDEP_EVE2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	RESERVED		R	0x0
6	WKUPDEP_EVE2_EVE1	Wakeup dependency from EVE2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: DISABLED 0x1: ENABLED	RW	0x0
5	WKUPDEP_EVE2_DSP2	Wakeup dependency from EVE2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: DISABLED 0x1: ENABLED	RW	0x0
4	WKUPDEP_EVE2_IPU1	Wakeup dependency from EVE2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: DISABLED 0x1: ENABLED	RW	0x0
3	WKUPDEP_EVE2_SDMA	Wakeup dependency from EVE2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: DISABLED 0x1: ENABLED	RW	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_EVE2_DSP1	Wakeup dependency from EVE2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: DISABLED 0x1: ENABLED	RW	0x0
1	WKUPDEP_EVE2_IPU2	Wakeup dependency from EVE2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: DISABLED 0x1: ENABLED	RW	0x0
0	WKUPDEP_EVE2_MPU	Wakeup dependency from EVE2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: DISABLED 0x1: ENABLED	RW	0x0

Table 3-1269. RM_EVE2_EVE2_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7BA4	**Instance**	EVE2_PRM
Description	This register contains dedicated EVE context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_EVE_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EVE_BANK	Specify if memory-based context in EVE2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EVE1_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.13.38 GPU_PRM Registers

3.13.38.1 GPU_PRM Register Summary

Table 3-1270. GPU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPU_PRM Physical Address L4_WKUP Interconnect
PM_GPU_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7200

Table 3-1270. GPU_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	GPU_PRM Physical Address L4_WKUP Interconnect
PM_GPU_PWRSTST	RW	32	0x0000 0004	0x4AE0 7204
RM_GPU_GPU_CONTEXT	RW	32	0x0000 0024	0x4AE0 7224

3.13.38.2 GPU_PRM Register Description

Table 3-1271. PM_GPU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7200	**Instance**	GPU_PRM
Description	This register controls the GPU power state to reach upon a domain sleep transition		
Type	RW		

31:18	17:16	15:5	4	3:2	1:0
RESERVED	GPU_MEM_ONSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	GPU_MEM_ONSTATE	GPU_MEM memory bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x0

Table 3-1272. Register Call Summary for Register PM_GPU_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2]

PRCM Register Manual

- GPU_PRM Register Summary: [3]

Table 3-1273. PM_GPU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7204	**Instance**	GPU_PRM
Description	This register provides a status on the current GPU power domain state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	GPU_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	GPU_MEM_STATEST	GPU_MEM memory bank state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x0
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x0

Table 3-1274. Register Call Summary for Register PM_GPU_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4]

PRCM Register Manual

- GPU_PRM Register Summary: [5]

Table 3-1275. RM_GPU_GPU_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7224	**Instance**	GPU_PRM
Description	This register contains dedicated GPU context statuses. [warm reset insensitive]		
Type	RW		

31–9	8	7–1	0
RESERVED	LOSTMEM_GPU_MEM	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_GPU_MEM	Specify if memory-based context in GPU_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of GPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1276. Register Call Summary for Register RM_GPU_GPU_CONTEXT

Power Management Functional Description

- PD_GPU Description: [0]

PRCM Register Manual

- GPU_PRM Register Summary: [1]

3.13.39 INSTR_PRM Registers

3.13.39.1 INSTR_PRM Register Summary

Table 3-1277. INSTR_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	INSTR_PRM Physical Address L4_WKUP Interconnect
PMI_IDENTICATION	R	32	0x0000 0000	0x4AE0 7F00
PMI_SYS_CONFIG	RW	32	0x0000 0010	0x4AE0 7F10
PMI_STATUS	R	32	0x0000 0014	0x4AE0 7F14
PMI_CONFIGURATION	RW	32	0x0000 0024	0x4AE0 7F24
PMI_CLASS_FILTERING	RW	32	0x0000 0028	0x4AE0 7F28
PMI_TRIGGERING	RW	32	0x0000 002C	0x4AE0 7F2C
PMI_SAMPLING	RW	32	0x0000 0030	0x4AE0 7F30

3.13.39.2 INSTR_PRM Register Description

Table 3-1278. PMI_IDENTICATION

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7F00	**Instance**	INSTR_PRM
Description	PM profiling identification register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	0x- (1)

(1) TI internal data

Table 3-1279. Register Call Summary for Register PMI_IDENTICATION

PRCM Register Manual

- INSTR_PRM Register Summary: [0]

Table 3-1280. PMI_SYS_CONFIG

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7F10	**Instance**	INSTR_PRM
Description	PM profiling system configuartion register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										RESERVED		IDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local tartget state management mode	RW	0x2
1	RESERVED		R	0x0
0	SOFTRESET	Software reset	RW	0x0

Table 3-1281. Register Call Summary for Register PMI_SYS_CONFIG

PRCM Register Manual

- INSTR_PRM Register Summary: [0]

Table 3-1282. PMI_STATUS

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7F14	**Instance**	INSTR_PRM
Description	PM profiling status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:9	8	7:0
RESERVED	FIFOEMPTY	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	FIFOEMPTY	PM Profiling buffer empty	R	0x1
7:0	RESERVED		R	0x0

Table 3-1283. Register Call Summary for Register PMI_STATUS

PRCM Register Manual

- INSTR_PRM Register Summary: [0]

Table 3-1284. PMI_CONFIGURATION

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7F24	**Instance**	INSTR_PRM
Description	PM profiling configuration register		
Type	RW		

31:30	29	28	27:24	23	22:16	15	14:8	7	6:0
CLAIM_3	CLAIM_2	CLAIM_1	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	EVT_CAPT_EN	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	CLAIM_3	Ownership	RW	0x0
29	CLAIM_2	Debugger override qualifier	RW	0x1
28	CLAIM_1	Current owner	R	0x0
27:24	RESERVED		R	0x0
23	RESERVED		R	0x0
22:16	RESERVED		R	0x0
15	RESERVED		R	0x0
14:8	RESERVED		R	0x0
7	EVT_CAPT_EN	When HIGH the PM events capture is enabled	RW	0x0
6:0	RESERVED		R	0x0

Table 3-1285. Register Call Summary for Register PMI_CONFIGURATION

PRCM Register Manual

- INSTR_PRM Register Summary: [0]

Table 3-1286. PMI_CLASS_FILTERING

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7F28	**Instance**	INSTR_PRM
Description	PM profiling class filtering register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SNAP_CAPT_EN_03	SNAP_CAPT_EN_02	SNAP_CAPT_EN_01	SNAP_CAPT_EN_00

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	SNAP_CAPT_EN_03	Snapshot capture enable - Class-ID = 0x03	RW	0x0
2	SNAP_CAPT_EN_02	Snapshot capture enable - Class-ID = 0x02	RW	0x0
1	SNAP_CAPT_EN_01	Snapshot capture enable - Class-ID = 0x01	RW	0x0
0	SNAP_CAPT_EN_00	Snapshot capture enable - Class-ID = 0x00	RW	0x0

Table 3-1287. Register Call Summary for Register PMI_CLASS_FILTERING

PRCM Register Manual

- INSTR_PRM Register Summary: [0]

Table 3-1288. PMI_TRIGGERING

Address Offset	0x0000 002C		
Physical Address	0x4AE0 7F2C	**Instance**	INSTR_PRM
Description	PM profiling triggering control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TRIG_STOP_EN	TRIG_START_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	TRIG_STOP_EN	Enable stop capturing PM events from external trigger detection	RW	0x0
0	TRIG_START_EN	Enable start capturing PM events from external trigger detection	RW	0x0

Table 3-1289. Register Call Summary for Register PMI_TRIGGERING

PRCM Register Manual

- INSTR_PRM Register Summary: [0]

Table 3-1290. PMI_SAMPLING

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7F30	**Instance**	INSTR_PRM
Description	PM profiling sampling window register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FCLK_DIV_FACOR	RESERVED	SAMP_WIND_SIZE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:16	FCLK_DIV_FACOR	FunClk divide factor ranging from 1 to 16	RW	0x0
15:8	RESERVED		R	0x0
7:0	SAMP_WIND_SIZE	PM events sampling window size	RW	0x0

Table 3-1291. Register Call Summary for Register PMI_SAMPLING

PRCM Register Manual

- INSTR_PRM Register Summary: [0]

3.13.40 IPU_PRM Registers

3.13.40.1 IPU_PRM Register Summary

Table 3-1292. IPU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU_PRM Physical Address L4_WKUP Interconnect
PM_IPU_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6500
PM_IPU_PWRSTST	RW	32	0x0000 0004	0x4AE0 6504
RM_IPU1_RSTCTRL	RW	32	0x0000 0010	0x4AE0 6510
RM_IPU1_RSTST	RW	32	0x0000 0014	0x4AE0 6514
RM_IPU1_IPU1_CONTEXT	RW	32	0x0000 0024	0x4AE0 6524
PM_IPU_MCASP1_WKDEP	RW	32	0x0000 0050	0x4AE0 6550
RM_IPU_MCASP1_CONTEXT	RW	32	0x0000 0054	0x4AE0 6554
PM_IPU_TIMER5_WKDEP	RW	32	0x0000 0058	0x4AE0 6558
RM_IPU_TIMER5_CONTEXT	RW	32	0x0000 005C	0x4AE0 655C
PM_IPU_TIMER6_WKDEP	RW	32	0x0000 0060	0x4AE0 6560
RM_IPU_TIMER6_CONTEXT	RW	32	0x0000 0064	0x4AE0 6564
PM_IPU_TIMER7_WKDEP	RW	32	0x0000 0068	0x4AE0 6568
RM_IPU_TIMER7_CONTEXT	RW	32	0x0000 006C	0x4AE0 656C
PM_IPU_TIMER8_WKDEP	RW	32	0x0000 0070	0x4AE0 6570
RM_IPU_TIMER8_CONTEXT	RW	32	0x0000 0074	0x4AE0 6574
PM_IPU_I2C5_WKDEP	RW	32	0x0000 0078	0x4AE0 6578
RM_IPU_I2C5_CONTEXT	RW	32	0x0000 007C	0x4AE0 657C
PM_IPU_UART6_WKDEP	RW	32	0x0000 0080	0x4AE0 6580
RM_IPU_UART6_CONTEXT	RW	32	0x0000 0084	0x4AE0 6584

3.13.40.2 IPU_PRM Register Description

Table 3-1293. PM_IPU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6500	**Instance**	IPU_PRM
Description	This register controls the IPU domain power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PERIPHMEM_ONSTATE	RESERVED	AESSMEM_ONSTATE	RESERVED	PERIPHMEM_RETSTATE	RESERVED	AESSMEM_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	PERIPHMEM_ONSTATE	PERIPHMEM memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	RESERVED		R	0x0
17:16	AESSMEM_ONSTATE	AESSMEM memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:11	RESERVED		R	0x0
10	PERIPHMEM_RETSTATE	Note: Not supported on this device.	R	0x0
9	RESERVED		R	0x0
8	AESSMEM_RETSTATE	Note: Not supported on this device.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Note: Not supported on this device.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x0

Table 3-1294. PM_IPU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6504	**Instance**	IPU_PRM
Description	This register provides a status on the IPU domain current power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED										PERIPHMEM_STATEST		RESERVED		AESSMEM_STATEST		RESERVED	LOGICSTATEST	POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	PERIPHMEM_STATEST	PERIPHMEM memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	RESERVED		R	0x0
5:4	AESSMEM_STATEST	AESSMEM memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1295. RM_IPU1_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6510	**Instance**	IPU_PRM
Description	This register controls the release of the IPU1 sub-system resets.		
Type	RW		

31–3	2	1	0
RESERVED	RST_IPU	RST_CPU1	RST_CPU0

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	RST_IPU	IPU system reset control. 0x0: Reset is cleared for IPU CACHE MMU 0x1: Reset is asserted for the IPU CACHE MMU	RW	0x1
1	RST_CPU1	IPU Cortex M4 CPU1 reset control 0x0: Reset is cleared for the IPU Cortex M4 CPU1 0x1: Reset is asserted for the IPU Cortex M4 CPU1	RW	0x1
0	RST_CPU0	IPU Cortex M4 CPU0 reset control. 0x0: Reset is cleared for the IPU Cortex M4 CPU0 0x1: Reset is asserted for the IPU Cortex M4 CPU0	RW	0x1

Table 3-1296. RM_IPU1_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6514	**Instance**	IPU_PRM
Description	This register logs the different reset sources of the IPU1 SS. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31:7	6	5	4	3	2	1	0
RESERVED	RST_ICECRUSHER_CPU1	RST_ICECRUSHER_CPU0	RST_EMULATION_CPU1	RST_EMULATION_CPU0	RST_IPU	RST_CPU1	RST_CPU0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	RST_ICECRUSHER_CPU1	Cortex M4 CPU1 has been reset due to IPU ICECRUSHER1 reset source Read 0x0: No icecrusher reset Read 0x1: CPU1 has been reset upon icecrusher reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
5	RST_ICECRUSHER_CPU0	Cortex M4 CPU0 has been reset due to IPU ICECRUSHER0 reset source Read 0x0: No icecrusher reset Read 0x1: CPU0 has been reset upon icecrusher reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
4	RST_EMULATION_CPU1	Cortex M4 CPU1 has been reset due to emulation reset source, for example, assert reset command initiated by the icepick module Read 0x0: No emulation reset Read 0x1: CPU1 has been reset upon emulation reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RST_EMULATION_CPU0	Cortex M4 CPU0 has been reset due to emulation reset source, for example assert reset command initiated by the icepick module Read 0x0: No emulation reset Read 0x1: CPU0 has been reset upon emulation reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
2	RST_IPU	IPU system software reset status Read 0x0: No software reset occurred Read 0x1: IPU MMU and CACHE interface has been reset upon SW reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
1	RST_CPU1	IPU Cortex-M4 CPU1 software reset status Read 0x0: No software reset occurred Read 0x1: Cortex M4 CPU1 has been reset upon software reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
0	RST_CPU0	IPU Cortex-M4 CPU0 software reset status Read 0x0: No software reset occurred Read 0x1: Cortex M4 CPU0 has been reset upon software reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0

Table 3-1297. RM_IPU1_IPU1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6524	**Instance**	IPU_PRM
Description	This register contains dedicated IPU1 context statuses. [warm reset insensitive]		
Type	RW		

31:10	9	8	7:2	1	0
RESERVED	LOSTMEM_IPU_L2RAM	LOSTMEM_IPU_UNICACHE	RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	LOSTMEM_IPU_L2RAM	Specify if memory-based context in IPU_L2RAM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
8	LOSTMEM_IPU_UNICACHE	Specify if memory-based context in IPU_UNICACHE memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1298. PM_IPU_MCASP1_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6550	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on McASP1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP1_DMA_DSP2	RESERVED	WKUPDEP_MCASP1_DMA_SDMA	WKUPDEP_MCASP1_DMA_DSP1	RESERVED		RESERVED	RESERVED	WKUPDEP_MCASP1_IRQ_EVE2	WKUPDEP_MCASP1_IRQ_EVE1	WKUPDEP_MCASP1_IRQ_DSP2	WKUPDEP_MCASP1_IRQ_IPU1	RESERVED	WKUPDEP_MCASP1_IRQ_DSP1	WKUPDEP_MCASP1_IRQ_IPU2	WKUPDEP_MCASP1_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP1_DMA_DSP2	Wakeup dependency from McASP1 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP1_DMA_SDMA	Wakeup dependency from McASP1 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
12	WKUPDEP_MCASP1_DMA_DSP1	Wakeup dependency from McASP1 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCASP1_IRQ_EVE2	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP1_IRQ_EVE1	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP1_IRQ_DSP2	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP1_IRQ_IPU1	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP1_IRQ_DSP1	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP1_IRQ_IPU2	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP1_IRQ_MPU	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1299. RM_IPU_MCASP1_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 6554	**Instance**	IPU_PRM
Description	This register contains dedicated McASP context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1300. PM_IPU_TIMER5_WKDEP

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6558	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER5 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER5_EVE2	WKUPDEP_TIMER5_EVE1	WKUPDEP_TIMER5_DSP2	WKUPDEP_TIMER5_IPU1	RESERVED	WKUPDEP_TIMER5_DSP1	WKUPDEP_TIMER5_IPU2	WKUPDEP_TIMER5_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER5_EVE2	Wakeup dependency from TIMER5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER5_EVE1	Wakeup dependency from TIMER5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER5_DSP2	Wakeup dependency from TIMER5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_TIMER5_IPU1	Wakeup dependency from TIMER5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER5_DSP1	Wakeup dependency from TIMER5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER5_IPU2	Wakeup dependency from TIMER5 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER5_MPU	Wakeup dependency from TIMER5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1301. RM_IPU_TIMER5_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 655C	**Instance**	IPU_PRM
Description	This register contains dedicated TIMER5 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1302. PM_IPU_TIMER6_WKDEP

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6560	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER6 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER6_EVE2	WKUPDEP_TIMER6_EVE1	WKUPDEP_TIMER6_DSP2	WKUPDEP_TIMER6_IPU1	RESERVED	WKUPDEP_TIMER6_DSP1	WKUPDEP_TIMER6_IPU2	WKUPDEP_TIMER6_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER6_EVE2	Wakeup dependency from TIMER6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER6_EVE1	Wakeup dependency from TIMER6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER6_DSP2	Wakeup dependency from TIMER6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER6_IPU1	Wakeup dependency from TIMER6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER6_DSP1	Wakeup dependency from TIMER6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER6_IPU2	Wakeup dependency from TIMER6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER6_MPU	Wakeup dependency from TIMER6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1303. RM_IPU_TIMER6_CONTEXT

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6564	**Instance**	IPU_PRM

Table 3-1303. RM_IPU_TIMER6_CONTEXT (continued)

Description	This register contains dedicated TIMER6 context statuses. [warm reset insensitive]
Type	RW

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1304. PM_IPU_TIMER7_WKDEP

Address Offset	0x0000 0068		
Physical Address	0x4AE0 6568	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER7 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER7_EVE2	WKUPDEP_TIMER7_EVE1	WKUPDEP_TIMER7_DSP2	WKUPDEP_TIMER7_IPU1	RESERVED	WKUPDEP_TIMER7_DSP1	WKUPDEP_TIMER7_IPU2	WKUPDEP_TIMER7_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER7_EVE2	Wakeup dependency from TIMER7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER7_EVE1	Wakeup dependency from TIMER7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_TIMER7_DSP2	Wakeup dependency from TIMER7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER7_IPU1	Wakeup dependency from TIMER7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER7_DSP1	Wakeup dependency from TIMER7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER7_IPU2	Wakeup dependency from TIMER7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER7_MPU	Wakeup dependency from TIMER7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1305. RM_IPU_TIMER7_CONTEXT

Address Offset	0x0000 006C		
Physical Address	0x4AE0 656C	**Instance**	IPU_PRM
Description	This register contains dedicated TIMER7 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1306. PM_IPU_TIMER8_WKDEP

Address Offset	0x0000 0070

Table 3-1306. PM_IPU_TIMER8_WKDEP (continued)

Physical Address	0x4AE0 6570	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER8 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER8_EVE2	WKUPDEP_TIMER8_EVE1	WKUPDEP_TIMER8_DSP2	WKUPDEP_TIMER8_IPU1	RESERVED	WKUPDEP_TIMER8_DSP1	WKUPDEP_TIMER8_IPU2	WKUPDEP_TIMER8_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER8_EVE2	Wakeup dependency from TIMER8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER8_EVE1	Wakeup dependency from TIMER8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER8_DSP2	Wakeup dependency from TIMER8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER8_IPU1	Wakeup dependency from TIMER8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER8_DSP1	Wakeup dependency from TIMER8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER8_IPU2	Wakeup dependency from TIMER8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_TIMER8_MPU	Wakeup dependency from TIMER8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1307. RM_IPU_TIMER8_CONTEXT

Address Offset	0x0000 0074		
Physical Address	0x4AE0 6574	**Instance**	IPU_PRM
Description	This register contains dedicated TIMER8 context statuses. [warm reset insensitive]		
Type	RW		

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1308. PM_IPU_I2C5_WKDEP

Address Offset	0x0000 0078		
Physical Address	0x4AE0 6578	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on I2C5 service requests.		
Type	RW		

31:16	15	14	13	12	11:10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_I2C5_DMA_DSP2	RESERVED	WKUPDEP_I2C5_DMA_SDMA	WKUPDEP_I2C5_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_I2C5_IRQ_EVE2	WKUPDEP_I2C5_IRQ_EVE1	WKUPDEP_I2C5_IRQ_DSP2	WKUPDEP_I2C5_IRQ_IPU1	RESERVED	WKUPDEP_I2C5_IRQ_DSP1	WKUPDEP_I2C5_IRQ_IPU2	WKUPDEP_I2C5_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
15	WKUPDEP_I2C5_DMA_DSP2	Wakeup dependency from I2C5 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C5_DMA_SDMA	Wakeup dependency from I2C5 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C5_DMA_DSP1	Wakeup dependency from I2C5 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_I2C5_IRQ_EVE2	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C5_IRQ_EVE1	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C5_IRQ_DSP2	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C5_IRQ_IPU1	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C5_IRQ_DSP1	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C5_IRQ_IPU2	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_I2C5_IRQ_MPU	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1309. RM_IPU_I2C5_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 657C	**Instance**	IPU_PRM
Description	This register contains dedicated I2C5 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1310. PM_IPU_UART6_WKDEP

Address Offset	0x0000 0080		
Physical Address	0x4AE0 6580	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on UART6 service requests.		
Type	RW		

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_UART6_EVE2	WKUPDEP_UART6_EVE1	WKUPDEP_UART6_DSP2	WKUPDEP_UART6_IPU1	WKUPDEP_UART6_SDMA	WKUPDEP_UART6_DSP1	WKUPDEP_UART6_IPU2	WKUPDEP_UART6_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_UART6_EVE2	Wakeup dependency from UART6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART6_EVE1	Wakeup dependency from UART6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART6_DSP2	Wakeup dependency from UART6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART6_IPU1	Wakeup dependency from UART6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART6_SDMA	Wakeup dependency from UART6 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART6_DSP1	Wakeup dependency from UART6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART6_IPU2	Wakeup dependency from UART6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART6_MPU	Wakeup dependency from UART6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1311. RM_IPU_UART6_CONTEXT

Address Offset	0x0000 0084		
Physical Address	0x4AE0 6584	**Instance**	IPU_PRM
Description	This register contains dedicated UART6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

3.13.41 IVA_PRM Registers

3.13.41.1 IVA_PRM Register Summary

Table 3-1312. IVA_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IVA_PRM Physical Address L4_WKUP Interconnect
PM_IVA_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6F00
PM_IVA_PWRSTST	RW	32	0x0000 0004	0x4AE0 6F04
RM_IVA_RSTCTRL	RW	32	0x0000 0010	0x4AE0 6F10
RM_IVA_RSTST	RW	32	0x0000 0014	0x4AE0 6F14
RM_IVA_IVA_CONTEXT	RW	32	0x0000 0024	0x4AE0 6F24
RM_IVA_SL2_CONTEXT	RW	32	0x0000 002C	0x4AE0 6F2C

3.13.41.2 IVA_PRM Register Description

Table 3-1313. PM_IVA_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6F00	**Instance**	IVA_PRM
Description	This register controls the IVA power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	TCM2_MEM_ONSTATE	TCM1_MEM_ONSTATE	SL2_MEM_ONSTATE	HWA_MEM_ONSTATE	RESERVED	TCM2_MEM_RETSTATE	TCM1_MEM_RETSTATE	SL2_MEM_RETSTATE	HWA_MEM_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:22	TCM2_MEM_ONSTATE	TCM_CORE memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
21:20	TCM1_MEM_ONSTATE	TCM1 memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	SL2_MEM_ONSTATE	SL2 memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	HWA_MEM_ONSTATE	HWA memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:12	RESERVED		R	0x0
11	TCM2_MEM_RETSTATE	Note: Not supported on this device.	R	0x0
10	TCM1_MEM_RETSTATE	Note: Not supported on this device.	R	0x0
9	SL2_MEM_RETSTATE	Note: Not supported on this device.	R	0x0
8	HWA_MEM_RETSTATE	Note: Not supported on this device.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Note: Not supported on this device.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1314. PM_IVA_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6F04	**Instance**	IVA_PRM
Description	This register provides a status on the current IVA power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	TCM2_MEM_STATEST	TCM1_MEM_STATEST	SL2_MEM_STATEST	HWA_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:12	RESERVED		R	0x0
11:10	TCM2_MEM_STATEST	TCM2 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
9:8	TCM1_MEM_STATEST	TCM1 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	SL2_MEM_STATEST	SL2 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	HWA_MEM_STATEST	HWA memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1315. RM_IVA_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6F10	**Instance**	IVA_PRM
Description	This register controls the release of the IVA sub-system resets.		
Type	RW		

31:3	2	1	0
RESERVED	RST_LOGIC	RST_SEQ2	RST_SEQ1

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	RST_LOGIC	IVA logic and SL2 reset control 0x0: Reset is cleared for the IVA logic and SL2 0x1: Reset is asserted for IVA logic and SL2	RW	0x1
1	RST_SEQ2	IVA Sequencer2 reset control 0x0: Reset is cleared for IVA Sequencer CPU2 0x1: Reset is asserted for IVA Sequencer CPU2	RW	0x1
0	RST_SEQ1	IVA sequencer1 reset control 0x0: Reset is cleared for the IVA Sequencer CPU1 0x1: Reset is asserted for the IVA sequencer CPU1	RW	0x1

Table 3-1316. RM_IVA_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6F14	**Instance**	IVA_PRM
Description	This register logs the different reset sources of the IVA domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31:7	6	5	4	3	2	1	0
RESERVED	RST_ICECRUSHER_SEQ2	RST_ICECRUSHER_SEQ1	RST_EMULATION_SEQ2	RST_EMULATION_SEQ1	RST_LOGIC	RST_SEQ2	RST_SEQ1

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6	RST_ICECRUSHER_SEQ2	Sequencer2 CPU has been reset due to IVA ICECRUSHER2 reset event 0x0: No icecrusher reset 0x1: Sequencer2 has been reset upon icecrusher reset	RW	0x0
5	RST_ICECRUSHER_SEQ1	Sequencer1 CPU has been reset due to IVA ICECRUSHER1 reset event 0x0: No icecrusher reset 0x1: Sequencer1 has been reset upon icecrusher reset	RW	0x0
4	RST_EMULATION_SEQ2	Sequencer2 CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: Sequencer2 has been reset upon emulation reset	RW	0x0
3	RST_EMULATION_SEQ1	Sequencer1 CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: Sequencer1 has been reset upon emulation reset	RW	0x0
2	RST_LOGIC	IVA logic and SL2 SW reset 0x0: No SW reset occurred 0x1: IVA logic and SL2 has been reset upon SW reset	RW	0x0
1	RST_SEQ2	IVA Sequencer2 CPU SW reset 0x0: No SW reset occurred 0x1: Sequencer2 has been reset upon SW reset	RW	0x0
0	RST_SEQ1	IVA Sequencer1 CPU SW reset 0x0: No SW reset occurred 0x1: Sequencer1 has been reset upon SW reset	RW	0x0

Table 3-1317. RM_IVA_IVA_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6F24	**Instance**	IVA_PRM
Description	This register contains dedicated IVA context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LOSTMEM_HWA_MEM	LOSTMEM_TCM2_MEM	LOSTMEM_TCM1_MEM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10	LOSTMEM_HWA_MEM	Specify if memory-based context in HWA_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
9	LOSTMEM_TCM2_MEM	Specify if memory-based context in TCM2_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_TCM1_MEM	Specify if memory-based context in TCM1_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IVA_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1318. RM_IVA_SL2_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 6F2C	**Instance**	IVA_PRM
Description	This register contains dedicated SL2 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_SL2_MEM	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_SL2_MEM	Specify if memory-based context in SL2_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IVA_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.13.42 L3INIT_PRM Registers

3.13.42.1 L3INIT_PRM Register Summary

Table 3-1319. L3INIT_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L3INIT_PRM Physical Address L4_WKUP Interconnect
PM_L3INIT_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7300
PM_L3INIT_PWRSTST	RW	32	0x0000 0004	0x4AE0 7304
RM_PCIESS_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7310
RM_PCIESS_RSTST	RW	32	0x0000 0014	0x4AE0 7314
PM_L3INIT_MMC1_WKDEP	RW	32	0x0000 0028	0x4AE0 7328
RM_L3INIT_MMC1_CONTEXT	RW	32	0x0000 002C	0x4AE0 732C
PM_L3INIT_MMC2_WKDEP	RW	32	0x0000 0030	0x4AE0 7330
RM_L3INIT_MMC2_CONTEXT	RW	32	0x0000 0034	0x4AE0 7334
PM_L3INIT_USB_OTG_SS2_WKDEP	RW	32	0x0000 0040	0x4AE0 7340
RM_L3INIT_USB_OTG_SS2_CONTEXT	RW	32	0x0000 0044	0x4AE0 7344
PM_L3INIT_USB_OTG_SS3_WKDEP	RW	32	0x0000 0048	0x4AE0 7348
RM_L3INIT_USB_OTG_SS3_CONTEXT	RW	32	0x0000 004C	0x4AE0 734C
PM_L3INIT_USB_OTG_SS4_WKDEP	RW	32	0x0000 0050	0x4AE0 7350
RM_L3INIT_USB_OTG_SS4_CONTEXT	RW	32	0x0000 0054	0x4AE0 7354
RM_L3INIT_MLB_SS_CONTEXT	RW	32	0x0000 005C	0x4AE0 735C
RM_L3INIT_IEEE1500_2_OCP_CONTEXT	RW	32	0x0000 007C	0x4AE0 737C
PM_L3INIT_SATA_WKDEP	RW	32	0x0000 0088	0x4AE0 7388
RM_L3INIT_SATA_CONTEXT	RW	32	0x0000 008C	0x4AE0 738C
PM_PCIE_PCIESS1_WKDEP	RW	32	0x0000 00B0	0x4AE0 73B0
RM_PCIE_PCIESS1_CONTEXT	RW	32	0x0000 00B4	0x4AE0 73B4
PM_PCIE_PCIESS2_WKDEP	RW	32	0x0000 00B8	0x4AE0 73B8
RM_PCIE_PCIESS2_CONTEXT	RW	32	0x0000 00BC	0x4AE0 73BC
RM_GMAC_GMAC_CONTEXT	RW	32	0x0000 00D4	0x4AE0 73D4
RM_L3INIT_OCP2SCP1_CONTEXT	RW	32	0x0000 00E4	0x4AE0 73E4
RM_L3INIT_OCP2SCP3_CONTEXT	RW	32	0x0000 00EC	0x4AE0 73EC
PM_L3INIT_USB_OTG_SS1_WKDEP	RW	32	0x0000 00F0	0x4AE0 73F0
RM_L3INIT_USB_OTG_SS1_CONTEXT	RW	32	0x0000 00F4	0x4AE0 73F4

3.13.42.2 L3INIT_PRM Register Description

Table 3-1320. PM_L3INIT_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7300	**Instance**	L3INIT_PRM
Description	This register controls the L3INIT power state to reach upon a domain sleep transition Note: In the L3INIT power domain OFF state is only allowed in systems where Ethernet RGMII is NOT used in the system - this is very application specific and may not be available in all TI standard software offerings.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GMAC_BANK_ONSTATE	L3INIT_BANK2_ONSTATE	L3INIT_BANK1_ONSTATE	RESERVED	GMAC_BANK_RETSTATE	L3INIT_BANK2_RETSTATE	L3INIT_BANK1_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:18	GMAC_BANK_ONSTATE	GMAC BANK state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	L3INIT_BANK2_ONSTATE	L3INIT BANK2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:14	L3INIT_BANK1_ONSTATE	L3INIT BANK1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
13:11	RESERVED		R	0x0
10	GMAC_BANK_RETSTATE	Note: Not supported on this device.	R	0x0
9	L3INIT_BANK2_RETSTATE	Note: Not supported on this device.	R	0x0
8	L3INIT_BANK1_RETSTATE	Note: Not supported on this device.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Note: Not supported on this device.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x0

Table 3-1321. Register Call Summary for Register PM_L3INIT_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4] [5] [6] [7] [8]

PRCM Register Manual

- L3INIT_PRM Register Summary: [9]

Table 3-1322. PM_L3INIT_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7304	**Instance**	L3INIT_PRM
Description	This register provides a status on the current L3INIT power domain state. [warm reset insensitive]		

Table 3-1322. PM_L3INIT_PWRSTST (continued)

Type RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">LASTPOWERSTATEENTERED</td><td colspan="3">RESERVED</td><td>INTRANSITION</td><td colspan="10">RESERVED</td><td colspan="2">L3INIT_GMAC_STATEST</td><td colspan="2">L3INIT_BANK2_STATEST</td><td colspan="2">L3INIT_BANK1_STATEST</td><td>RESERVED</td><td>LOGICSTATEST</td><td colspan="2">POWERSTATEST</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	L3INIT_GMAC_STATEST	L3INIT GMAC state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	L3INIT_BANK2_STATEST	L3INIT BANK2 state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	L3INIT_BANK1_STATEST	L3INIT BANK1 state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1323. Register Call Summary for Register PM_L3INIT_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L3INIT_PRM Register Summary: [7]

Table 3-1324. RM_PCIESS_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7310	**Instance**	L3INIT_PRM
Description	This register controls the release of the PCIESS local reset.		
Type	RW		

31-2	1	0
RESERVED	RST_LOCAL_PCIE2	RST_LOCAL_PCIE1

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_LOCAL_PCIE2	PCIESS2 local reset control 0x0: Reset is cleared for the PCIE2 0x1: Reset is asserted for the PCIE2	RW	0x1
0	RST_LOCAL_PCIE1	PCIESS1 local reset control 0x0: Reset is cleared for the PCIE1 0x1: Reset is asserted for the PCIE1	RW	0x1

Table 3-1325. Register Call Summary for Register RM_PCIESS_RSTCTRL

PRCM Register Manual

- L3INIT_PRM Register Summary: [0]

Table 3-1326. RM_PCIESS_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7314	**Instance**	L3INIT_PRM
Description	This register logs the different reset sources of the PCIESS domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31-2	1	0
RESERVED	RST_LOCAL_PCIE2	RST_LOCAL_PCIE1

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	RST_LOCAL_PCIE2	PCIESS2 local SW reset 0x0: No SW reset occurred 0x1: PCIE2 has been reset upon SW reset	RW	0x0
0	RST_LOCAL_PCIE1	PCIESS1 local SW reset 0x0: No SW reset occurred 0x1: PCIE1 has been reset upon SW reset	RW	0x0

Table 3-1327. Register Call Summary for Register RM_PCIESS_RSTST

PRCM Register Manual

- L3INIT_PRM Register Summary: [0]

Table 3-1328. PM_L3INIT_MMC1_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7328	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on MMC1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_MMC1_EVE2	WKUPDEP_MMC1_EVE1	WKUPDEP_MMC1_DSP2	WKUPDEP_MMC1_IPU1	WKUPDEP_MMC1_SDMA	WKUPDEP_MMC1_DSP1	WKUPDEP_MMC1_IPU2	WKUPDEP_MMC1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MMC1_EVE2	Wakeup dependency from MMC1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC1_EVE1	Wakeup dependency from MMC1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MMC1_DSP2	Wakeup dependency from MMC1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_MMC1_IPU1	Wakeup dependency from MMC1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC1_SDMA	Wakeup dependency from MMC1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC1_DSP1	Wakeup dependency from MMC1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC1_IPU2	Wakeup dependency from MMC1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MMC1_MPU	Wakeup dependency from MMC1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1329. Register Call Summary for Register PM_L3INIT_MMC1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L3INIT_PRM Register Summary: [10]

Table 3-1330. RM_L3INIT_MMC1_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 732C	**Instance**	L3INIT_PRM
Description	This register contains dedicated MMC1 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1331. Register Call Summary for Register RM_L3INIT_MMC1_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1332. PM_L3INIT_MMC2_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7330	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on MMC2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_MMC2_EVE2	WKUPDEP_MMC2_EVE1	WKUPDEP_MMC2_DSP2	WKUPDEP_MMC2_IPU1	WKUPDEP_MMC2_SDMA	WKUPDEP_MMC2_DSP1	WKUPDEP_MMC2_IPU2	WKUPDEP_MMC2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MMC2_EVE2	Wakeup dependency from MMC2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC2_EVE1	Wakeup dependency from MMC2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_MMC2_DSP2	Wakeup dependency from MMC2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MMC2_IPU1	Wakeup dependency from MMC2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC2_SDMA	Wakeup dependency from MMC2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC2_DSP1	Wakeup dependency from MMC2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC2_IPU2	Wakeup dependency from MMC2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MMC2_MPU	Wakeup dependency from MMC2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1333. Register Call Summary for Register PM_L3INIT_MMC2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L3INIT_PRM Register Summary: [10]

Table 3-1334. RM_L3INIT_MMC2_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7334	**Instance**	L3INIT_PRM
Description	This register contains dedicated MMC2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1335. Register Call Summary for Register RM_L3INIT_MMC2_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1336. PM_L3INIT_USB_OTG_SS2_WKDEP

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7340	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS2 service requests.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td>RESERVED</td><td>RESERVED</td><td>WKUPDEP_USB_OTG_SS2_EVE2</td><td>WKUPDEP_USB_OTG_SS2_EVE1</td><td>WKUPDEP_USB_OTG_SS2_DSP2</td><td>WKUPDEP_USB_OTG_SS2_IPU1</td><td>RESERVED</td><td>WKUPDEP_USB_OTG_SS2_DSP1</td><td>WKUPDEP_USB_OTG_SS2_IPU2</td><td>WKUPDEP_USB_OTG_SS2_MPU</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RESERVED		R	0x0
7	WKUPDEP_USB_OTG_SS2_EVE2	Wakeup dependency from USB2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS2_EVE1	Wakeup dependency from USB2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS2_DSP2	Wakeup dependency from USB2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_USB_OTG_SS2_IPU1	Wakeup dependency from USB2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS2_DSP1	Wakeup dependency from USB2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS2_IPU2	Wakeup dependency from USB2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS2_MPU	Wakeup dependency from USB2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1337. Register Call Summary for Register PM_L3INIT_USB_OTG_SS2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L3INIT_PRM Register Summary: [9]

Table 3-1338. RM_L3INIT_USB_OTG_SS2_CONTEXT

Address Offset	0x0000 0044		
Physical Address	0x4AE0 7344	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:9	8	7:2	1	0
RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1339. Register Call Summary for Register RM_L3INIT_USB_OTG_SS2_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1340. PM_L3INIT_USB_OTG_SS3_WKDEP

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7348	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS3 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_USB_OTG_SS3_EVE2	WKUPDEP_USB_OTG_SS3_EVE1	WKUPDEP_USB_OTG_SS3_DSP2	WKUPDEP_USB_OTG_SS3_IPU1	RESERVED	WKUPDEP_USB_OTG_SS3_DSP1	WKUPDEP_USB_OTG_SS3_IPU2	WKUPDEP_USB_OTG_SS3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RESERVED		R	0x0
7	WKUPDEP_USB_OTG_SS3_EVE2	Wakeup dependency from USB3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS3_EVE1	Wakeup dependency from USB3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS3_DSP2	Wakeup dependency from USB3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_USB_OTG_SS3_IPU1	Wakeup dependency from USB3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS3_DSP1	Wakeup dependency from USB3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS3_IPU2	Wakeup dependency from USB3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS3_MPU	Wakeup dependency from USB3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1341. Register Call Summary for Register PM_L3INIT_USB_OTG_SS3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L3INIT_PRM Register Summary: [9]

Table 3-1342. RM_L3INIT_USB_OTG_SS3_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 734C	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1343. Register Call Summary for Register RM_L3INIT_USB_OTG_SS3_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1344. PM_L3INIT_USB_OTG_SS4_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 7350	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS4 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_USB_OTG_SS4_EVE2	WKUPDEP_USB_OTG_SS4_EVE1	WKUPDEP_USB_OTG_SS4_DSP2	WKUPDEP_USB_OTG_SS4_IPU1	RESERVED	WKUPDEP_USB_OTG_SS4_DSP1	WKUPDEP_USB_OTG_SS4_IPU2	WKUPDEP_USB_OTG_SS4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RESERVED		R	0x0
7	WKUPDEP_USB_OTG_SS4_EVE2	Wakeup dependency from USB4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS4_EVE1	Wakeup dependency from USB4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS4_DSP2	Wakeup dependency from USB4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_USB_OTG_SS4_IPU1	Wakeup dependency from USB4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS4_DSP1	Wakeup dependency from USB4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS4_IPU2	Wakeup dependency from USB4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS4_MPU	Wakeup dependency from USB4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1345. Register Call Summary for Register PM_L3INIT_USB_OTG_SS4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L3INIT_PRM Register Summary: [9]

Table 3-1346. RM_L3INIT_USB_OTG_SS4_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 7354	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1347. Register Call Summary for Register RM_L3INIT_USB_OTG_SS4_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1348. RM_L3INIT_MLB_SS_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 735C	**Instance**	L3INIT_PRM
Description	This register contains dedicated MLBSS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_MLB_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	LOSTMEM_MLB_BANK	Specify if memory-based context in MLB_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1349. Register Call Summary for Register RM_L3INIT_MLB_SS_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1350. RM_L3INIT_IEEE1500_2_OCP_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 737C	**Instance**	L3INIT_PRM
Description	This register contains dedicated IEEE1500_2_OCP context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1351. Register Call Summary for Register RM_L3INIT_IEEE1500_2_OCP_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1352. PM_L3INIT_SATA_WKDEP

Address Offset	0x0000 0088		
Physical Address	0x4AE0 7388	**Instance**	L3INIT_PRM

Table 3-1352. PM_L3INIT_SATA_WKDEP (continued)

Description	This register controls wakeup dependency based on SATA service requests.
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td>RESERVED</td><td>RESERVED</td><td>WKUPDEP_SATA_EVE2</td><td>WKUPDEP_SATA_EVE1</td><td>WKUPDEP_SATA_DSP2</td><td>WKUPDEP_SATA_IPU1</td><td>RESERVED</td><td>WKUPDEP_SATA_DSP1</td><td>WKUPDEP_SATA_IPU2</td><td>WKUPDEP_SATA_MPU</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_SATA_EVE2	Wakeup dependency from SATA module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_SATA_EVE1	Wakeup dependency from SATA module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_SATA_DSP2	Wakeup dependency from SATA module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_SATA_IPU1	Wakeup dependency from SATA module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_SATA_DSP1	Wakeup dependency from SATA module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_SATA_IPU2	Wakeup dependency from SATA module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_SATA_MPU	Wakeup dependency from SATA module (SWakeup signal) towards MPU + L3MAIN1 + L4CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1353. Register Call Summary for Register PM_L3INIT_SATA_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5]

PRCM Register Manual

- L3INIT_PRM Register Summary: [8]

Table 3-1354. RM_L3INIT_SATA_CONTEXT

Address Offset	0x0000 008C		
Physical Address	0x4AE0 738C	**Instance**	L3INIT_PRM
Description	This register contains dedicated SATA context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1355. Register Call Summary for Register RM_L3INIT_SATA_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1356. PM_PCIE_PCIESS1_WKDEP

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 73B0	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on PCIESS1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_PCIESS1_EVE2	WKUPDEP_PCIESS1_EVE1	WKUPDEP_PCIESS1_DSP2	WKUPDEP_PCIESS1_IPU1	RESERVED	WKUPDEP_PCIESS1_DSP1	WKUPDEP_PCIESS1_IPU2	WKUPDEP_PCIESS1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_PCIESS1_EVE2	Wakeup dependency from PCIESS1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_PCIESS1_EVE1	Wakeup dependency from PCIESS1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_PCIESS1_DSP2	Wakeup dependency from PCIESS1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_PCIESS1_IPU1	Wakeup dependency from PCIESS1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_PCIESS1_DSP1	Wakeup dependency from PCIESS1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_PCIESS1_IPU2	Wakeup dependency from PCIESS1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_PCIESS1_MPU	Wakeup dependency from PCIESS1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1357. Register Call Summary for Register PM_PCIE_PCIESS1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L3INIT_PRM Register Summary: [9]

Table 3-1358. RM_PCIE_PCIESS1_CONTEXT

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 73B4	**Instance**	L3INIT_PRM
Description	This register contains dedicated PCIESS1 context statuses. [warm reset insensitive]		
Type	RW		

31–9	8	7–1	0
RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in PCIESS1_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1359. Register Call Summary for Register RM_PCIE_PCIESS1_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1360. PM_PCIE_PCIESS2_WKDEP

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 73B8	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on PCIESS2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_PCIESS2_EVE2	WKUPDEP_PCIESS2_EVE1	WKUPDEP_PCIESS2_DSP2	WKUPDEP_PCIESS2_IPU1	RESERVED	WKUPDEP_PCIESS2_DSP1	WKUPDEP_PCIESS2_IPU2	WKUPDEP_PCIESS2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_PCIESS2_EVE2	Wakeup dependency from PCIESS2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_PCIESS2_EVE1	Wakeup dependency from PCIESS2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_PCIESS2_DSP2	Wakeup dependency from PCIESS2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_PCIESS2_IPU1	Wakeup dependency from PCIESS2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_PCIESS2_DSP1	Wakeup dependency from PCIESS2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_PCIESS2_IPU2	Wakeup dependency from PCIESS2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_PCIESS2_MPU	Wakeup dependency from PCIESS2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1361. Register Call Summary for Register PM_PCIE_PCIESS2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L3INIT_PRM Register Summary: [9]

Table 3-1362. RM_PCIE_PCIESS2_CONTEXT

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 73BC	**Instance**	L3INIT_PRM
Description	This register contains dedicated PCIESS2 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in PCIESS1_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1363. Register Call Summary for Register RM_PCIE_PCIESS2_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1364. RM_GMAC_GMAC_CONTEXT

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 73D4	**Instance**	L3INIT_PRM
Description	This register contains dedicated GMAC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_GMAC_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_GMAC_BANK	Specify if memory-based context in GMAC_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1365. Register Call Summary for Register RM_GMAC_GMAC_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1366. RM_L3INIT_OCP2SCP1_CONTEXT

Address Offset	0x0000 00E4		
Physical Address	0x4AE0 73E4	**Instance**	L3INIT_PRM
Description	This register contains dedicated OCP2SCP1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1367. Register Call Summary for Register RM_L3INIT_OCP2SCP1_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1368. RM_L3INIT_OCP2SCP3_CONTEXT

Address Offset	0x0000 00EC		
Physical Address	0x4AE0 73EC	**Instance**	L3INIT_PRM
Description	This register contains dedicated OCP2SCP3 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1369. Register Call Summary for Register RM_L3INIT_OCP2SCP3_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

Table 3-1370. PM_L3INIT_USB_OTG_SS1_WKDEP

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 73F0	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_USB_OTG_SS1_EVE2	WKUPDEP_USB_OTG_SS1_EVE1	WKUPDEP_USB_OTG_SS1_DSP2	WKUPDEP_USB_OTG_SS1_IPU1	RESERVED	WKUPDEP_USB_OTG_SS1_DSP1	WKUPDEP_USB_OTG_SS1_IPU2	WKUPDEP_USB_OTG_SS1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_USB_OTG_SS1_EVE2	Wakeup dependency from USB1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS1_EVE1	Wakeup dependency from USB1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS1_DSP2	Wakeup dependency from USB1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_USB_OTG_SS1_IPU1	Wakeup dependency from USB1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS1_DSP1	Wakeup dependency from USB1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS1_IPU2	Wakeup dependency from USB1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS1_MPU	Wakeup dependency from USB1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1371. Register Call Summary for Register PM_L3INIT_USB_OTG_SS1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L3INIT_PRM Register Summary: [9]

Table 3-1372. RM_L3INIT_USB_OTG_SS1_CONTEXT

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 73F4	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS1 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_L3INIT_BANK1	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1373. Register Call Summary for Register RM_L3INIT_USB_OTG_SS1_CONTEXT

Power Management Functional Description

- PD_L3INIT Description: [0]

PRCM Register Manual

- L3INIT_PRM Register Summary: [1]

3.13.43 L4PER_PRM Registers

3.13.43.1 L4PER_PRM Register Summary

Table 3-1374. L4PER_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM Physical Address L4_WKUP Interconnect
PM_L4PER_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7400

Table 3-1374. L4PER_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM Physical Address L4_WKUP Interconnect
PM_L4PER_PWRSTST	RW	32	0x0000 0004	0x4AE0 7404
RM_L4PER2_L4PER2_CONTEXT	RW	32	0x0000 000C	0x4AE0 740C
RM_L4PER3_L4PER3_CONTEXT	RW	32	0x0000 0014	0x4AE0 7414
RESERVED	RW	32	0x0000 001C	0x4AE0 741C
RESERVED	RW	32	0x0000 0024	0x4AE0 7424
PM_L4PER_TIMER10_WKDEP	RW	32	0x0000 0028	0x4AE0 7428
RM_L4PER_TIMER10_CONTEXT	RW	32	0x0000 002C	0x4AE0 742C
PM_L4PER_TIMER11_WKDEP	RW	32	0x0000 0030	0x4AE0 7430
RM_L4PER_TIMER11_CONTEXT	RW	32	0x0000 0034	0x4AE0 7434
PM_L4PER_TIMER2_WKDEP	RW	32	0x0000 0038	0x4AE0 7438
RM_L4PER_TIMER2_CONTEXT	RW	32	0x0000 003C	0x4AE0 743C
PM_L4PER_TIMER3_WKDEP	RW	32	0x0000 0040	0x4AE0 7440
RM_L4PER_TIMER3_CONTEXT	RW	32	0x0000 0044	0x4AE0 7444
PM_L4PER_TIMER4_WKDEP	RW	32	0x0000 0048	0x4AE0 7448
RM_L4PER_TIMER4_CONTEXT	RW	32	0x0000 004C	0x4AE0 744C
PM_L4PER_TIMER9_WKDEP	RW	32	0x0000 0050	0x4AE0 7450
RM_L4PER_TIMER9_CONTEXT	RW	32	0x0000 0054	0x4AE0 7454
RM_L4PER_ELM_CONTEXT	RW	32	0x0000 005C	0x4AE0 745C
PM_L4PER_GPIO2_WKDEP	RW	32	0x0000 0060	0x4AE0 7460
RM_L4PER_GPIO2_CONTEXT	RW	32	0x0000 0064	0x4AE0 7464
PM_L4PER_GPIO3_WKDEP	RW	32	0x0000 0068	0x4AE0 7468
RM_L4PER_GPIO3_CONTEXT	RW	32	0x0000 006C	0x4AE0 746C
PM_L4PER_GPIO4_WKDEP	RW	32	0x0000 0070	0x4AE0 7470
RM_L4PER_GPIO4_CONTEXT	RW	32	0x0000 0074	0x4AE0 7474
PM_L4PER_GPIO5_WKDEP	RW	32	0x0000 0078	0x4AE0 7478
RM_L4PER_GPIO5_CONTEXT	RW	32	0x0000 007C	0x4AE0 747C
PM_L4PER_GPIO6_WKDEP	RW	32	0x0000 0080	0x4AE0 7480
RM_L4PER_GPIO6_CONTEXT	RW	32	0x0000 0084	0x4AE0 7484
RM_L4PER_HDQ1W_CONTEXT	RW	32	0x0000 008C	0x4AE0 748C
RM_L4PER2_PWMSS2_CONTEXT	RW	32	0x0000 0094	0x4AE0 7494
RM_L4PER2_PWMSS3_CONTEXT	RW	32	0x0000 009C	0x4AE0 749C
PM_L4PER_I2C1_WKDEP	RW	32	0x0000 00A0	0x4AE0 74A0
RM_L4PER_I2C1_CONTEXT	RW	32	0x0000 00A4	0x4AE0 74A4
PM_L4PER_I2C2_WKDEP	RW	32	0x0000 00A8	0x4AE0 74A8
RM_L4PER_I2C2_CONTEXT	RW	32	0x0000 00AC	0x4AE0 74AC
PM_L4PER_I2C3_WKDEP	RW	32	0x0000 00B0	0x4AE0 74B0
RM_L4PER_I2C3_CONTEXT	RW	32	0x0000 00B4	0x4AE0 74B4
PM_L4PER_I2C4_WKDEP	RW	32	0x0000 00B8	0x4AE0 74B8
RM_L4PER_I2C4_CONTEXT	RW	32	0x0000 00BC	0x4AE0 74BC
RM_L4PER_L4PER1_CONTEXT	RW	32	0x0000 00C0	0x4AE0 74C0
RM_L4PER2_PWMSS1_CONTEXT	RW	32	0x0000 00C4	0x4AE0 74C4
PM_L4PER_TIMER13_WKDEP	RW	32	0x0000 00C8	0x4AE0 74C8
RM_L4PER3_TIMER13_CONTEXT	RW	32	0x0000 00CC	0x4AE0 74CC
PM_L4PER_TIMER14_WKDEP	RW	32	0x0000 00D0	0x4AE0 74D0
RM_L4PER3_TIMER14_CONTEXT	RW	32	0x0000 00D4	0x4AE0 74D4
PM_L4PER_TIMER15_WKDEP	RW	32	0x0000 00D8	0x4AE0 74D8
RM_L4PER3_TIMER15_CONTEXT	RW	32	0x0000 00DC	0x4AE0 74DC
PM_L4PER_MCSPI1_WKDEP	RW	32	0x0000 00F0	0x4AE0 74F0
RM_L4PER_MCSPI1_CONTEXT	RW	32	0x0000 00F4	0x4AE0 74F4

Table 3-1374. L4PER_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM Physical Address L4_WKUP Interconnect
PM_L4PER_MCSPI2_WKDEP	RW	32	0x0000 00F8	0x4AE0 74F8
RM_L4PER_MCSPI2_CONTEXT	RW	32	0x0000 00FC	0x4AE0 74FC
PM_L4PER_MCSPI3_WKDEP	RW	32	0x0000 0100	0x4AE0 7500
RM_L4PER_MCSPI3_CONTEXT	RW	32	0x0000 0104	0x4AE0 7504
PM_L4PER_MCSPI4_WKDEP	RW	32	0x0000 0108	0x4AE0 7508
RM_L4PER_MCSPI4_CONTEXT	RW	32	0x0000 010C	0x4AE0 750C
PM_L4PER_GPIO7_WKDEP	RW	32	0x0000 0110	0x4AE0 7510
RM_L4PER_GPIO7_CONTEXT	RW	32	0x0000 0114	0x4AE0 7514
PM_L4PER_GPIO8_WKDEP	RW	32	0x0000 0118	0x4AE0 7518
RM_L4PER_GPIO8_CONTEXT	RW	32	0x0000 011C	0x4AE0 751C
PM_L4PER_MMC3_WKDEP	RW	32	0x0000 0120	0x4AE0 7520
RM_L4PER_MMC3_CONTEXT	RW	32	0x0000 0124	0x4AE0 7524
PM_L4PER_MMC4_WKDEP	RW	32	0x0000 0128	0x4AE0 7528
RM_L4PER_MMC4_CONTEXT	RW	32	0x0000 012C	0x4AE0 752C
PM_L4PER_TIMER16_WKDEP	RW	32	0x0000 0130	0x4AE0 7530
RM_L4PER3_TIMER16_CONTEXT	RW	32	0x0000 0134	0x4AE0 7534
PM_L4PER2_QSPI_WKDEP	RW	32	0x0000 0138	0x4AE0 7538
RM_L4PER2_QSPI_CONTEXT	RW	32	0x0000 013C	0x4AE0 753C
PM_L4PER_UART1_WKDEP	RW	32	0x0000 0140	0x4AE0 7540
RM_L4PER_UART1_CONTEXT	RW	32	0x0000 0144	0x4AE0 7544
PM_L4PER_UART2_WKDEP	RW	32	0x0000 0148	0x4AE0 7548
RM_L4PER_UART2_CONTEXT	RW	32	0x0000 014C	0x4AE0 754C
PM_L4PER_UART3_WKDEP	RW	32	0x0000 0150	0x4AE0 7550
RM_L4PER_UART3_CONTEXT	RW	32	0x0000 0154	0x4AE0 7554
PM_L4PER_UART4_WKDEP	RW	32	0x0000 0158	0x4AE0 7558
RM_L4PER_UART4_CONTEXT	RW	32	0x0000 015C	0x4AE0 755C
PM_L4PER2_MCASP2_WKDEP	RW	32	0x0000 0160	0x4AE0 7560
RM_L4PER2_MCASP2_CONTEXT	RW	32	0x0000 0164	0x4AE0 7564
PM_L4PER2_MCASP3_WKDEP	RW	32	0x0000 0168	0x4AE0 7568
RM_L4PER2_MCASP3_CONTEXT	RW	32	0x0000 016C	0x4AE0 756C
PM_L4PER_UART5_WKDEP	RW	32	0x0000 0170	0x4AE0 7570
RM_L4PER_UART5_CONTEXT	RW	32	0x0000 0174	0x4AE0 7574
PM_L4PER2_MCASP5_WKDEP	RW	32	0x0000 0178	0x4AE0 7578
RM_L4PER2_MCASP5_CONTEXT	RW	32	0x0000 017C	0x4AE0 757C
PM_L4PER2_MCASP6_WKDEP	RW	32	0x0000 0180	0x4AE0 7580
RM_L4PER2_MCASP6_CONTEXT	RW	32	0x0000 0184	0x4AE0 7584
PM_L4PER2_MCASP7_WKDEP	RW	32	0x0000 0188	0x4AE0 7588
RM_L4PER2_MCASP7_CONTEXT	RW	32	0x0000 018C	0x4AE0 758C
PM_L4PER2_MCASP8_WKDEP	RW	32	0x0000 0190	0x4AE0 7590
RM_L4PER2_MCASP8_CONTEXT	RW	32	0x0000 0194	0x4AE0 7594
PM_L4PER2_MCASP4_WKDEP	RW	32	0x0000 0198	0x4AE0 7598
RM_L4PER2_MCASP4_CONTEXT	RW	32	0x0000 019C	0x4AE0 759C
RM_L4SEC_AES1_CONTEXT	RW	32	0x0000 01A4	0x4AE0 75A4
RM_L4SEC_AES2_CONTEXT	RW	32	0x0000 01AC	0x4AE0 75AC
RM_L4SEC_DES3DES_CONTEXT	RW	32	0x0000 01B4	0x4AE0 75B4
RM_L4SEC_FPKA_CONTEXT	RW	32	0x0000 01BC	0x4AE0 75BC
RM_L4SEC_RNG_CONTEXT	RW	32	0x0000 01C4	0x4AE0 75C4
RM_L4SEC_SHA2MD51_CONTEXT	RW	32	0x0000 01CC	0x4AE0 75CC
PM_L4PER2_UART7_WKDEP	RW	32	0x0000 01D0	0x4AE0 75D0

Table 3-1374. L4PER_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM Physical Address L4_WKUP Interconnect
RM_L4PER2_UART7_CONTEXT	RW	32	0x0000 01D4	0x4AE0 75D4
RM_L4SEC_DMA_CRYPTO_CONTEXT	RW	32	0x0000 01DC	0x4AE0 75DC
PM_L4PER2_UART8_WKDEP	RW	32	0x0000 01E0	0x4AE0 75E0
RM_L4PER2_UART8_CONTEXT	RW	32	0x0000 01E4	0x4AE0 75E4
PM_L4PER2_UART9_WKDEP	RW	32	0x0000 01E8	0x4AE0 75E8
RM_L4PER2_UART9_CONTEXT	RW	32	0x0000 01EC	0x4AE0 75EC
PM_L4PER2_DCAN2_WKDEP	RW	32	0x0000 01F0	0x4AE0 75F0
RM_L4PER2_DCAN2_CONTEXT	RW	32	0x0000 01F4	0x4AE0 75F4
RM_L4SEC_SHA2MD52_CONTEXT	RW	32	0x0000 01FC	0x4AE0 75FC

3.13.43.2 L4PER_PRM Register Description

Table 3-1375. PM_L4PER_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7400	**Instance**	L4PER_PRM
Description	This register controls the L4PER power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												NONRETAINED_BANK_ONSTATE		RETAINED_BANK_ONSTATE		RESERVED						NONRETAINED_BANK_RETSTATE	RETAINED_BANK_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:18	NONRETAINED_BANK_ONSTATE	NONRETAINED_BANK state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	RETAINED_BANK_ONSTATE	RETAINED_BANK state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:10	RESERVED		R	0x0
9	NONRETAINED_BANK_RETSTATE	Note: Not supported on this device.	R	0x0
8	RETAINED_BANK_RETSTATE	Note: Not supported on this device.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	0x1
1:0	POWERSTATE	Power state control 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1376. Register Call Summary for Register PM_L4PER_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L4PER_PRM Register Summary: [7]

Table 3-1377. PM_L4PER_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7404	**Instance**	L4PER_PRM
Description	This register provides a status on the current L4PER power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED												NONRETAINED_BANK_STATEST		RETAINED_BANK_STATEST		RESERVED	LOGICSTATEST	POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:6	NONRETAINED_BANK_STATEST	NONRETAINED_BANK state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	RETAINED_BANK_STATEST	RETAINED_BANK state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1378. Register Call Summary for Register PM_L4PER_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4] [5]

PRCM Register Manual

- L4PER_PRM Register Summary: [6]

Table 3-1379. RM_L4PER2_L4PER2_CONTEXT

Address Offset	0x0000 000C		
Physical Address	0x4AE0 740C	**Instance**	L4PER_PRM
Description	This register contains dedicated L4_PER2 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1380. Register Call Summary for Register RM_L4PER2_L4PER2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0] [1]

PRCM Register Manual

- L4PER_PRM Register Summary: [2]

Table 3-1381. RM_L4PER3_L4PER3_CONTEXT

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7414	**Instance**	L4PER_PRM
Description	This register contains dedicated L4_PER3 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1382. Register Call Summary for Register RM_L4PER3_L4PER3_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0] [1]

PRCM Register Manual

- L4PER_PRM Register Summary: [2]

Table 3-1383. PM_L4PER_TIMER10_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7428	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER10 service requests.		

Table 3-1383. PM_L4PER_TIMER10_WKDEP (continued)

Type	RW

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER10_EVE2	WKUPDEP_TIMER10_EVE1	WKUPDEP_TIMER10_DSP2	WKUPDEP_TIMER10_IPU1	RESERVED	WKUPDEP_TIMER10_DSP1	WKUPDEP_TIMER10_IPU2	WKUPDEP_TIMER10_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER10_EVE2	Wakeup dependency from TIMER10 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER10_EVE1	Wakeup dependency from TIMER10 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER10_DSP2	Wakeup dependency from TIMER10 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER10_IPU1	Wakeup dependency from TIMER10 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER10_DSP1	Wakeup dependency from TIMER10 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER10_IPU2	Wakeup dependency from TIMER10 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_TIMER10_MPU	Wakeup dependency from TIMER10 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1384. Register Call Summary for Register PM_L4PER_TIMER10_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L4PER_PRM Register Summary: [9]

Table 3-1385. RM_L4PER_TIMER10_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 742C	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER10 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1386. Register Call Summary for Register RM_L4PER_TIMER10_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1387. PM_L4PER_TIMER11_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7430	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER11 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER11_EVE2	WKUPDEP_TIMER11_EVE1	WKUPDEP_TIMER11_DSP2	WKUPDEP_TIMER11_IPU1	RESERVED	WKUPDEP_TIMER11_DSP1	WKUPDEP_TIMER11_IPU2	WKUPDEP_TIMER11_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER11_EVE2	Wakeup dependency from TIMER11 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER11_EVE1	Wakeup dependency from TIMER11 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER11_DSP2	Wakeup dependency from TIMER11 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER11_IPU1	Wakeup dependency from TIMER11 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER11_DSP1	Wakeup dependency from TIMER11 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER11_IPU2	Wakeup dependency from TIMER11 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER11_MPU	Wakeup dependency from TIMER11 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1388. Register Call Summary for Register PM_L4PER_TIMER11_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L4PER_PRM Register Summary: [9]

Table 3-1389. RM_L4PER_TIMER11_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7434	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER11 context statuses. [warm reset insensitive]		
Type	RW		

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1390. Register Call Summary for Register RM_L4PER_TIMER11_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1391. PM_L4PER_TIMER2_WKDEP

Address Offset	0x0000 0038		
Physical Address	0x4AE0 7438	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER2 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER2_EVE2	WKUPDEP_TIMER2_EVE1	WKUPDEP_TIMER2_DSP2	WKUPDEP_TIMER2_IPU1	RESERVED	WKUPDEP_TIMER2_DSP1	WKUPDEP_TIMER2_IPU2	WKUPDEP_TIMER2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER2_EVE2	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER2_EVE1	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER2_DSP2	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER2_IPU1	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER2_DSP1	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER2_IPU2	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER2_MPU	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1392. Register Call Summary for Register PM_L4PER_TIMER2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L4PER_PRM Register Summary: [9]

Table 3-1393. RM_L4PER_TIMER2_CONTEXT

Address Offset	0x0000 003C		
Physical Address	0x4AE0 743C	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER2 context statuses. [warm reset insensitive]		

Table 3-1393. RM_L4PER_TIMER2_CONTEXT (continued)

Type RW

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1394. Register Call Summary for Register RM_L4PER_TIMER2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1395. PM_L4PER_TIMER3_WKDEP

Address Offset 0x0000 0040

Physical Address 0x4AE0 7440 **Instance** L4PER_PRM

Description This register controls wakeup dependency based on TIMER3 service requests.

Type RW

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER3_EVE2	WKUPDEP_TIMER3_EVE1	WKUPDEP_TIMER3_DSP2	WKUPDEP_TIMER3_IPU1	RESERVED	WKUPDEP_TIMER3_DSP1	WKUPDEP_TIMER3_IPU2	WKUPDEP_TIMER3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER3_EVE2	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_TIMER3_EVE1	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER3_DSP2	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER3_IPU1	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER3_DSP1	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER3_IPU2	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER3_MPU	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1396. Register Call Summary for Register PM_L4PER_TIMER3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L4PER_PRM Register Summary: [9]

Table 3-1397. RM_L4PER_TIMER3_CONTEXT

Address Offset	0x0000 0044		
Physical Address	0x4AE0 7444	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1398. Register Call Summary for Register RM_L4PER_TIMER3_CONTEXT

Power Management Functional Description
- PD_L4PER Description: [0]

PRCM Register Manual
- L4PER_PRM Register Summary: [1]

Table 3-1399. PM_L4PER_TIMER4_WKDEP

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7448	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER4 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER4_EVE2	WKUPDEP_TIMER4_EVE1	WKUPDEP_TIMER4_DSP2	WKUPDEP_TIMER4_IPU1	RESERVED	WKUPDEP_TIMER4_DSP1	WKUPDEP_TIMER4_IPU2	WKUPDEP_TIMER4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER4_EVE2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER4_EVE1	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_TIMER4_DSP2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER4_IPU1	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER4_DSP1	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER4_IPU2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER4_MPU	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1400. Register Call Summary for Register PM_L4PER_TIMER4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L4PER_PRM Register Summary: [9]

Table 3-1401. RM_L4PER_TIMER4_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 744C	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER4 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1402. Register Call Summary for Register RM_L4PER_TIMER4_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1403. PM_L4PER_TIMER9_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 7450	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER9 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER9_EVE2	WKUPDEP_TIMER9_EVE1	WKUPDEP_TIMER9_DSP2	WKUPDEP_TIMER9_IPU1	RESERVED	WKUPDEP_TIMER9_DSP1	WKUPDEP_TIMER9_IPU2	WKUPDEP_TIMER9_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER9_EVE2	Wakeup dependency from TIMER9 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER9_EVE1	Wakeup dependency from TIMER9 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER9_DSP2	Wakeup dependency from TIMER9 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_TIMER9_IPU1	Wakeup dependency from TIMER9 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER9_DSP1	Wakeup dependency from TIMER9 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER9_IPU2	Wakeup dependency from TIMER9 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER9_MPU	Wakeup dependency from TIMER9 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1404. Register Call Summary for Register PM_L4PER_TIMER9_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- L4PER_PRM Register Summary: [9]

Table 3-1405. RM_L4PER_TIMER9_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 7454	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER9 context statuses. [warm reset insensitive]		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="31">RESERVED</td><td>LOSTCONTEXT_DFF</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1406. Register Call Summary for Register RM_L4PER_TIMER9_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1407. RM_L4PER_ELM_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 745C	**Instance**	L4PER_PRM
Description	This register contains dedicated ELM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1408. Register Call Summary for Register RM_L4PER_ELM_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1409. PM_L4PER_GPIO2_WKDEP

Address Offset	0x0000 0060		
Physical Address	0x4AE0 7460	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_GPIO2_IRQ2_EVE2	WKUPDEP_GPIO2_IRQ2_EVE1	WKUPDEP_GPIO2_IRQ2_DSP2	WKUPDEP_GPIO2_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO2_IRQ2_DSP1	WKUPDEP_GPIO2_IRQ2_IPU2	WKUPDEP_GPIO2_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO2_IRQ1_EVE2	WKUPDEP_GPIO2_IRQ1_EVE1	WKUPDEP_GPIO2_IRQ1_DSP2	WKUPDEP_GPIO2_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO2_IRQ1_DSP1	WKUPDEP_GPIO2_IRQ1_IPU2	WKUPDEP_GPIO2_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO2_IRQ2_EVE2	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO2_IRQ2_EVE1	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO2_IRQ2_DSP2	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO2_IRQ2_IPU1	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO2_IRQ2_DSP1	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO2_IRQ2_IPU2	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO2_IRQ2_MPU	Wakeup dependency from GPIO2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_GPIO2_IRQ1_EVE2	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO2_IRQ1_EVE1	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO2_IRQ1_DSP2	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO2_IRQ1_IPU1	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO2_IRQ1_DSP1	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO2_IRQ1_IPU2	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO2_IRQ1_MPU	Wakeup dependency from GPIO2 module (SWakeup signal for POROCPSINTERRUPT1) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1410. Register Call Summary for Register PM_L4PER_GPIO2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

PRCM Register Manual

- L4PER_PRM Register Summary: [18]

Table 3-1411. RM_L4PER_GPIO2_CONTEXT

Address Offset	0x0000 0064		
Physical Address	0x4AE0 7464	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1412. Register Call Summary for Register RM_L4PER_GPIO2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1413. PM_L4PER_GPIO3_WKDEP

Address Offset	0x0000 0068		
Physical Address	0x4AE0 7468	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO3 service requests.		
Type	RW		

31:20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_GPIO3_IRQ2_EVE2	WKUPDEP_GPIO3_IRQ2_EVE1	WKUPDEP_GPIO3_IRQ2_DSP2	WKUPDEP_GPIO3_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO3_IRQ2_DSP1	WKUPDEP_GPIO3_IRQ2_IPU2	WKUPDEP_GPIO3_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO3_IRQ1_EVE2	WKUPDEP_GPIO3_IRQ1_EVE1	WKUPDEP_GPIO3_IRQ1_DSP2	WKUPDEP_GPIO3_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO3_IRQ1_DSP1	WKUPDEP_GPIO3_IRQ1_IPU2	WKUPDEP_GPIO3_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO3_IRQ2_EVE2	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
16	WKUPDEP_GPIO3_IRQ2_EVE1	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO3_IRQ2_DSP2	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO3_IRQ2_IPU1	Wakeup dependency from GPIO3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO3_IRQ2_DSP1	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO3_IRQ2_IPU2	Wakeup dependency from GPIO3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO3_IRQ2_MPU	Wakeup dependency from GPIO3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_GPIO3_IRQ1_EVE2	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO3_IRQ1_EVE1	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO3_IRQ1_DSP2	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO3_IRQ1_IPU1	Wakeup dependency from GPIO3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	WKUPDEP_GPIO3_IRQ1_DSP1	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO3_IRQ1_IPU2	3Wakeup dependency from GPIO3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO3_IRQ1_MPU	Wakeup dependency from GPIO3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1414. Register Call Summary for Register PM_L4PER_GPIO3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

PRCM Register Manual

- L4PER_PRM Register Summary: [18]

Table 3-1415. RM_L4PER_GPIO3_CONTEXT

Address Offset	0x0000 006C		
Physical Address	0x4AE0 746C	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO3 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1416. Register Call Summary for Register RM_L4PER_GPIO3_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

Table 3-1416. Register Call Summary for Register RM_L4PER_GPIO3_CONTEXT (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1417. PM_L4PER_GPIO4_WKDEP

Address Offset	0x0000 0070		
Physical Address	0x4AE0 7470	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED	RESERVED	WKUPDEP_GPIO4_IRQ2_EVE2	WKUPDEP_GPIO4_IRQ2_EVE1	WKUPDEP_GPIO4_IRQ2_DSP2	WKUPDEP_GPIO4_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO4_IRQ2_DSP1	WKUPDEP_GPIO4_IRQ2_IPU2	WKUPDEP_GPIO4_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO4_IRQ1_EVE2	WKUPDEP_GPIO4_IRQ1_EVE1	WKUPDEP_GPIO4_IRQ1_DSP2	WKUPDEP_GPIO4_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO4_IRQ1_DSP1	WKUPDEP_GPIO4_IRQ1_IPU2	WKUPDEP_GPIO4_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO4_IRQ2_EVE2	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO4_IRQ2_EVE1	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO4_IRQ2_DSP2	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO4_IRQ2_IPU1	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO4_IRQ2_DSP1	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
11	WKUPDEP_GPIO4_IRQ2_IPU2	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO4_IRQ2_MPU	Wakeup dependency from GPIO4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_GPIO4_IRQ1_EVE2	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO4_IRQ1_EVE1	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO4_IRQ1_DSP2	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO4_IRQ1_IPU1	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO4_IRQ1_DSP1	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO4_IRQ1_IPU2	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO4_IRQ1_MPU	Wakeup dependency from GPIO4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1418. Register Call Summary for Register PM_L4PER_GPIO4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

Table 3-1418. Register Call Summary for Register PM_L4PER_GPIO4_WKDEP (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [18]

Table 3-1419. RM_L4PER_GPIO4_CONTEXT

Address Offset	0x0000 0074		
Physical Address	0x4AE0 7474	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO4 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1420. Register Call Summary for Register RM_L4PER_GPIO4_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1421. PM_L4PER_GPIO5_WKDEP

Address Offset	0x0000 0078		
Physical Address	0x4AE0 7478	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_GPIO5_IRQ2_EVE2	WKUPDEP_GPIO5_IRQ2_EVE1	WKUPDEP_GPIO5_IRQ2_DSP2	WKUPDEP_GPIO5_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO5_IRQ2_DSP1	WKUPDEP_GPIO5_IRQ2_IPU2	WKUPDEP_GPIO5_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO5_IRQ1_EVE2	WKUPDEP_GPIO5_IRQ1_EVE1	WKUPDEP_GPIO5_IRQ1_DSP2	WKUPDEP_GPIO5_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO5_IRQ1_DSP1	WKUPDEP_GPIO5_IRQ1_IPU2	WKUPDEP_GPIO5_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO5_IRQ2_EVE2	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO5_IRQ2_EVE1	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO5_IRQ2_DSP2	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO5_IRQ2_IPU1	Wakeup dependency from GPIO5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO5_IRQ2_DSP1	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO5_IRQ2_IPU2	Wakeup dependency from GPIO5 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO5_IRQ2_MPU	Wakeup dependency from GPIO5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_GPIO5_IRQ1_EVE2	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO5_IRQ1_EVE1	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO5_IRQ1_DSP2	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO5_IRQ1_IPU1	Wakeup dependency from GPIO5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO5_IRQ1_DSP1	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO5_IRQ1_IPU2	5Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO5_IRQ1_MPU	Wakeup dependency from GPIO5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1422. Register Call Summary for Register PM_L4PER_GPIO5_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

PRCM Register Manual

- L4PER_PRM Register Summary: [18]

Table 3-1423. RM_L4PER_GPIO5_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 747C	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1424. Register Call Summary for Register RM_L4PER_GPIO5_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1425. PM_L4PER_GPIO6_WKDEP

Address Offset	0x0000 0080		
Physical Address	0x4AE0 7480	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO6 service requests.		
Type	RW		

31:20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_GPIO6_IRQ2_EVE2	WKUPDEP_GPIO6_IRQ2_EVE1	WKUPDEP_GPIO6_IRQ2_DSP2	WKUPDEP_GPIO6_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO6_IRQ2_DSP1	WKUPDEP_GPIO6_IRQ2_IPU2	WKUPDEP_GPIO6_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO6_IRQ1_EVE2	WKUPDEP_GPIO6_IRQ1_EVE1	WKUPDEP_GPIO6_IRQ1_DSP2	WKUPDEP_GPIO6_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO6_IRQ1_DSP1	WKUPDEP_GPIO6_IRQ1_IPU2	WKUPDEP_GPIO6_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO6_IRQ2_EVE2	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
16	WKUPDEP_GPIO6_IRQ2_EVE1	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO6_IRQ2_DSP2	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO6_IRQ2_IPU1	Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO6_IRQ2_DSP1	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO6_IRQ2_IPU2	Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO6_IRQ2_MPU	Wakeup dependency from GPIO6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_GPIO6_IRQ1_EVE2	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO6_IRQ1_EVE1	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO6_IRQ1_DSP2	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO6_IRQ1_IPU1	Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	WKUPDEP_GPIO6_IRQ1_DSP1	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO6_IRQ1_IPU2	5Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO6_IRQ1_MPU	Wakeup dependency from GPIO6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1426. Register Call Summary for Register PM_L4PER_GPIO6_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

PRCM Register Manual

- L4PER_PRM Register Summary: [18]

Table 3-1427. RM_L4PER_GPIO6_CONTEXT

Address Offset	0x0000 0084		
Physical Address	0x4AE0 7484	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO6 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1428. Register Call Summary for Register RM_L4PER_GPIO6_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

Table 3-1428. Register Call Summary for Register RM_L4PER_GPIO6_CONTEXT (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1429. RM_L4PER_HDQ1W_CONTEXT

Address Offset	0x0000 008C		
Physical Address	0x4AE0 748C	**Instance**	L4PER_PRM
Description	This register contains dedicated HDQ1W context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1430. Register Call Summary for Register RM_L4PER_HDQ1W_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1431. RM_L4PER2_PWMSS2_CONTEXT

Address Offset	0x0000 0094		
Physical Address	0x4AE0 7494	**Instance**	L4PER_PRM
Description	This register contains dedicated PWMSS2 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1432. Register Call Summary for Register RM_L4PER2_PWMSS2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1433. RM_L4PER2_PWMSS3_CONTEXT

Address Offset	0x0000 009C		
Physical Address	0x4AE0 749C	**Instance**	L4PER_PRM
Description	This register contains dedicated PWMSS3 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1434. Register Call Summary for Register RM_L4PER2_PWMSS3_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1435. PM_L4PER_I2C1_WKDEP

Address Offset	0x0000 00A0		
Physical Address	0x4AE0 74A0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	WKUPDEP_I2C1_DMA_DSP2	RESERVED	WKUPDEP_I2C1_DMA_SDMA	WKUPDEP_I2C1_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_I2C1_IRQ_EVE2	WKUPDEP_I2C1_IRQ_EVE1	WKUPDEP_I2C1_IRQ_DSP2	WKUPDEP_I2C1_IRQ_IPU1	RESERVED	WKUPDEP_I2C1_IRQ_DSP1	WKUPDEP_I2C1_IRQ_IPU2	WKUPDEP_I2C1_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C1_DMA_DSP2	Wakeup dependency from I2C1 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C1_DMA_SDMA	Wakeup dependency from I2C1 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C1_DMA_DSP1	Wakeup dependency from I2C1 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_I2C1_IRQ_EVE2	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C1_IRQ_EVE1	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C1_IRQ_DSP2	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C1_IRQ_IPU1	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_I2C1_IRQ_DSP1	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C1_IRQ_IPU2	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C1_IRQ_MPU	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1436. Register Call Summary for Register PM_L4PER_I2C1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1437. RM_L4PER_I2C1_CONTEXT

Address Offset	0x0000 00A4		
Physical Address	0x4AE0 74A4	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C1 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1438. Register Call Summary for Register RM_L4PER_I2C1_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

Table 3-1438. Register Call Summary for Register RM_L4PER_I2C1_CONTEXT (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1439. PM_L4PER_I2C2_WKDEP

Address Offset	0x0000 00A8		
Physical Address	0x4AE0 74A8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C2 service requests.		
Type	RW		

31:16	15	14	13	12	11:10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_I2C2_DMA_DSP2	RESERVED	WKUPDEP_I2C2_DMA_SDMA	WKUPDEP_I2C2_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_I2C2_IRQ_EVE2	WKUPDEP_I2C2_IRQ_EVE1	WKUPDEP_I2C2_IRQ_DSP2	WKUPDEP_I2C2_IRQ_IPU1	RESERVED	WKUPDEP_I2C2_IRQ_DSP1	WKUPDEP_I2C2_IRQ_IPU2	WKUPDEP_I2C2_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C2_DMA_DSP2	Wakeup dependency from I2C2 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C2_DMA_SDMA	Wakeup dependency from I2C2 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C2_DMA_DSP1	Wakeup dependency from I2C2 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_I2C2_IRQ_EVE2	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C2_IRQ_EVE1	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_I2C2_IRQ_DSP2	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C2_IRQ_IPU1	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C2_IRQ_DSP1	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C2_IRQ_IPU2	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C2_IRQ_MPU	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1440. Register Call Summary for Register PM_L4PER_I2C2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1441. RM_L4PER_I2C2_CONTEXT

Address Offset	0x0000 00AC		
Physical Address	0x4AE0 74AC	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C2 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1442. Register Call Summary for Register RM_L4PER_I2C2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1443. PM_L4PER_I2C3_WKDEP

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 74B0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C3 service requests.		
Type	RW		

31-16	15	14	13	12	11-10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_I2C3_DMA_DSP2	RESERVED	WKUPDEP_I2C3_DMA_SDMA	WKUPDEP_I2C3_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_I2C3_IRQ_EVE2	WKUPDEP_I2C3_IRQ_EVE1	WKUPDEP_I2C3_IRQ_DSP2	WKUPDEP_I2C3_IRQ_IPU1	RESERVED	WKUPDEP_I2C3_IRQ_DSP1	WKUPDEP_I2C3_IRQ_IPU2	WKUPDEP_I2C3_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C3_DMA_DSP2	Wakeup dependency from I2C3 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C3_DMA_SDMA	Wakeup dependency from I2C3 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C3_DMA_DSP1	Wakeup dependency from I2C3 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_I2C3_IRQ_EVE2	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C3_IRQ_EVE1	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C3_IRQ_DSP2	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C3_IRQ_IPU1	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C3_IRQ_DSP1	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C3_IRQ_IPU2	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C3_IRQ_MPU	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1444. Register Call Summary for Register PM_L4PER_I2C3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1445. RM_L4PER_I2C3_CONTEXT

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 74B4	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1446. Register Call Summary for Register RM_L4PER_I2C3_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1447. PM_L4PER_I2C4_WKDEP

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 74B8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C4 service requests.		
Type	RW		

31:16	15	14	13	12	11:10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_I2C4_DMA_DSP2	RESERVED	WKUPDEP_I2C4_DMA_SDMA	WKUPDEP_I2C4_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_I2C4_IRQ_EVE2	WKUPDEP_I2C4_IRQ_EVE1	WKUPDEP_I2C4_IRQ_DSP2	WKUPDEP_I2C4_IRQ_IPU1	RESERVED	WKUPDEP_I2C4_IRQ_DSP1	WKUPDEP_I2C4_IRQ_IPU2	WKUPDEP_I2C4_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C4_DMA_DSP2	Wakeup dependency from I2C4 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13	WKUPDEP_I2C4_DMA_SDMA	Wakeup dependency from I2C4 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C4_DMA_DSP1	Wakeup dependency from I2C4 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_I2C4_IRQ_EVE2	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C4_IRQ_EVE1	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C4_IRQ_DSP2	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C4_IRQ_IPU1	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C4_IRQ_DSP1	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C4_IRQ_IPU2	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C4_IRQ_MPU	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1448. Register Call Summary for Register PM_L4PER_I2C4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1449. RM_L4PER_I2C4_CONTEXT

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 74BC	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C4 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1450. Register Call Summary for Register RM_L4PER_I2C4_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1451. RM_L4PER_L4PER1_CONTEXT

Address Offset	0x0000 00C0		
Physical Address	0x4AE0 74C0	**Instance**	L4PER_PRM
Description	This register contains dedicated L4_PER1 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1452. Register Call Summary for Register RM_L4PER_L4PER1_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0] [1]

PRCM Register Manual

- L4PER_PRM Register Summary: [2]

Table 3-1453. RM_L4PER2_PWMSS1_CONTEXT

Address Offset	0x0000 00C4		
Physical Address	0x4AE0 74C4	**Instance**	L4PER_PRM
Description	This register contains dedicated PWMSS1 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1454. Register Call Summary for Register RM_L4PER2_PWMSS1_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1455. PM_L4PER_TIMER13_WKDEP

Address Offset	0x0000 00C8		
Physical Address	0x4AE0 74C8	**Instance**	L4PER_PRM

Table 3-1455. PM_L4PER_TIMER13_WKDEP (continued)

Description This register controls wakeup dependency based on TIMER13 service requests.

Type RW

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER13_EVE2	WKUPDEP_TIMER13_EVE1	WKUPDEP_TIMER13_DSP2	WKUPDEP_TIMER13_IPU1	RESERVED	WKUPDEP_TIMER13_DSP1	WKUPDEP_TIMER13_IPU2	WKUPDEP_TIMER13_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER13_EVE2	Wakeup dependency from TIMER13 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER13_EVE1	Wakeup dependency from TIMER13 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER13_DSP2	Wakeup dependency from TIMER13 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER13_IPU1	Wakeup dependency from TIMER13 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER13_DSP1	Wakeup dependency from TIMER13 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER13_IPU2	Wakeup dependency from TIMER13 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_TIMER13_MPU	Wakeup dependency from TIMER13 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1456. Register Call Summary for Register PM_L4PER_TIMER13_WKDEP

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1457. RM_L4PER3_TIMER13_CONTEXT

Address Offset	0x0000 00CC		
Physical Address	0x4AE0 74CC	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER13 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1458. Register Call Summary for Register RM_L4PER3_TIMER13_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1459. PM_L4PER_TIMER14_WKDEP

Address Offset	0x0000 00D0		
Physical Address	0x4AE0 74D0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER14 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER14_EVE2	WKUPDEP_TIMER14_EVE1	WKUPDEP_TIMER14_DSP2	WKUPDEP_TIMER14_IPU1	RESERVED	WKUPDEP_TIMER14_DSP1	WKUPDEP_TIMER14_IPU2	WKUPDEP_TIMER14_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER14_EVE2	Wakeup dependency from TIMER14 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER14_EVE1	Wakeup dependency from TIMER14 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER14_DSP2	Wakeup dependency from TIMER14 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER14_IPU1	Wakeup dependency from TIMER14 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER14_DSP1	Wakeup dependency from TIMER14 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER14_IPU2	Wakeup dependency from TIMER14 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER14_MPU	Wakeup dependency from TIMER14 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1460. Register Call Summary for Register PM_L4PER_TIMER14_WKDEP

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1461. RM_L4PER3_TIMER14_CONTEXT

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 74D4	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER14 context statuses. [warm reset insensitive]		
Type	RW		

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1462. Register Call Summary for Register RM_L4PER3_TIMER14_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1463. PM_L4PER_TIMER15_WKDEP

Address Offset	0x0000 00D8		
Physical Address	0x4AE0 74D8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER15 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER15_EVE2	WKUPDEP_TIMER15_EVE1	WKUPDEP_TIMER15_DSP2	WKUPDEP_TIMER15_IPU1	RESERVED	WKUPDEP_TIMER15_DSP1	WKUPDEP_TIMER15_IPU2	WKUPDEP_TIMER15_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER15_EVE2	Wakeup dependency from TIMER15 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER15_EVE1	Wakeup dependency from TIMER15 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER15_DSP2	Wakeup dependency from TIMER15 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER15_IPU1	Wakeup dependency from TIMER15 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER15_DSP1	Wakeup dependency from TIMER15 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER15_IPU2	Wakeup dependency from TIMER15 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER15_MPU	Wakeup dependency from TIMER15 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1464. Register Call Summary for Register PM_L4PER_TIMER15_WKDEP

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1465. RM_L4PER3_TIMER15_CONTEXT

Address Offset	0x0000 00DC		
Physical Address	0x4AE0 74DC	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER15 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1466. Register Call Summary for Register RM_L4PER3_TIMER15_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1467. PM_L4PER_MCSPI1_WKDEP

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 74F0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI1 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_MCSPI1_EVE2	WKUPDEP_MCSPI1_EVE1	WKUPDEP_MCSPI1_DSP2	WKUPDEP_MCSPI1_IPU1	WKUPDEP_MCSPI1_SDMA	WKUPDEP_MCSPI1_DSP1	WKUPDEP_MCSPI1_IPU2	WKUPDEP_MCSPI1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCSPI1_EVE2	Wakeup dependency from McSPI1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_MCSPI1_EVE1	Wakeup dependency from McSPI1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCSPI1_DSP2	Wakeup dependency from McSPI1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCSPI1_IPU1	Wakeup dependency from McSPI1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI1_SDMA	Wakeup dependency from McSPI1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI1_DSP1	Wakeup dependency from McSPI1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI1_IPU2	Wakeup dependency from McSPI1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI1_MPU	Wakeup dependency from McSPI1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1468. Register Call Summary for Register PM_L4PER_MCSPI1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1469. RM_L4PER_MCSPI1_CONTEXT

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 74F4	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1470. Register Call Summary for Register RM_L4PER_MCSPI1_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1471. PM_L4PER_MCSPI2_WKDEP

Address Offset	0x0000 00F8		
Physical Address	0x4AE0 74F8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI2 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_MCSPI2_EVE2	WKUPDEP_MCSPI2_EVE1	WKUPDEP_MCSPI2_DSP2	WKUPDEP_MCSPI2_IPU1	WKUPDEP_MCSPI2_SDMA	WKUPDEP_MCSPI2_DSP1	WKUPDEP_MCSPI2_IPU2	WKUPDEP_MCSPI2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCSPI2_EVE2	Wakeup dependency from McSPI2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_MCSPI2_EVE1	Wakeup dependency from McSPI2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCSPI2_DSP2	Wakeup dependency from McSPI2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCSPI2_IPU1	Wakeup dependency from McSPI2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI2_SDMA	Wakeup dependency from McSPI2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI2_DSP1	Wakeup dependency from McSPI2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI2_IPU2	Wakeup dependency from McSPI2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI2_MPU	Wakeup dependency from McSPI2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1472. Register Call Summary for Register PM_L4PER_MCSPI2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1473. RM_L4PER_MCSPI2_CONTEXT

Address Offset	0x0000 00FC		
Physical Address	0x4AE0 74FC	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1474. Register Call Summary for Register RM_L4PER_MCSPI2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1475. PM_L4PER_MCSPI3_WKDEP

Address Offset	0x0000 0100		
Physical Address	0x4AE0 7500	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_MCSPI3_EVE2	WKUPDEP_MCSPI3_EVE1	WKUPDEP_MCSPI3_DSP2	WKUPDEP_MCSPI3_IPU1	WKUPDEP_MCSPI3_SDMA	WKUPDEP_MCSPI3_DSP1	WKUPDEP_MCSPI3_IPU2	WKUPDEP_MCSPI3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCSPI3_EVE2	Wakeup dependency from McSPI3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_MCSPI3_EVE1	Wakeup dependency from McSPI3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCSPI3_DSP2	Wakeup dependency from McSPI3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCSPI3_IPU1	Wakeup dependency from McSPI3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI3_SDMA	Wakeup dependency from McSPI3 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI3_DSP1	Wakeup dependency from McSPI3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI3_IPU2	Wakeup dependency from McSPI3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI3_MPU	Wakeup dependency from McSPI3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1476. Register Call Summary for Register PM_L4PER_MCSPI3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1477. RM_L4PER_MCSPI3_CONTEXT

Address Offset	0x0000 0104		
Physical Address	0x4AE0 7504	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1478. Register Call Summary for Register RM_L4PER_MCSPI3_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1479. PM_L4PER_MCSPI4_WKDEP

Address Offset	0x0000 0108		
Physical Address	0x4AE0 7508	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI4 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_MCSPI4_EVE2	WKUPDEP_MCSPI4_EVE1	WKUPDEP_MCSPI4_DSP2	WKUPDEP_MCSPI4_IPU1	WKUPDEP_MCSPI4_SDMA	WKUPDEP_MCSPI4_DSP1	WKUPDEP_MCSPI4_IPU2	WKUPDEP_MCSPI4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCSPI4_EVE2	Wakeup dependency from McSPI4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_MCSPI4_EVE1	Wakeup dependency from McSPI4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCSPI4_DSP2	Wakeup dependency from McSPI4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCSPI4_IPU1	Wakeup dependency from McSPI4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI4_SDMA	Wakeup dependency from McSPI4 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI4_DSP1	Wakeup dependency from McSPI4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI4_IPU2	Wakeup dependency from McSPI4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI4_MPU	Wakeup dependency from McSPI4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1480. Register Call Summary for Register PM_L4PER_MCSPI4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1481. RM_L4PER_MCSPI4_CONTEXT

Address Offset	0x0000 010C		
Physical Address	0x4AE0 750C	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1482. Register Call Summary for Register RM_L4PER_MCSPI4_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1483. PM_L4PER_GPIO7_WKDEP

Address Offset	0x0000 0110		
Physical Address	0x4AE0 7510	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO7 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED	RESERVED	WKUPDEP_GPIO7_IRQ2_EVE2	WKUPDEP_GPIO7_IRQ2_EVE1	WKUPDEP_GPIO7_IRQ2_DSP2	WKUPDEP_GPIO7_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO7_IRQ2_DSP1	WKUPDEP_GPIO7_IRQ2_IPU2	WKUPDEP_GPIO7_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO7_IRQ1_EVE2	WKUPDEP_GPIO7_IRQ1_EVE1	WKUPDEP_GPIO7_IRQ1_DSP2	WKUPDEP_GPIO7_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO7_IRQ1_DSP1	WKUPDEP_GPIO7_IRQ1_IPU2	WKUPDEP_GPIO7_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO7_IRQ2_EVE2	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
16	WKUPDEP_GPIO7_IRQ2_EVE1	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO7_IRQ2_DSP2	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO7_IRQ2_IPU1	Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO7_IRQ2_DSP1	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO7_IRQ2_IPU2	Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO7_IRQ2_MPU	Wakeup dependency from GPIO7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_GPIO7_IRQ1_EVE2	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO7_IRQ1_EVE1	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO7_IRQ1_DSP2	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO7_IRQ1_IPU1	Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	WKUPDEP_GPIO7_IRQ1_DSP1	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO7_IRQ1_IPU2	5Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO7_IRQ1_MPU	Wakeup dependency from GPIO7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1484. Register Call Summary for Register PM_L4PER_GPIO7_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

PRCM Register Manual

- L4PER_PRM Register Summary: [18]

Table 3-1485. RM_L4PER_GPIO7_CONTEXT

Address Offset	0x0000 0114		
Physical Address	0x4AE0 7514	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO7 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1486. Register Call Summary for Register RM_L4PER_GPIO7_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

Table 3-1486. Register Call Summary for Register RM_L4PER_GPIO7_CONTEXT (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1487. PM_L4PER_GPIO8_WKDEP

Address Offset	0x0000 0118		
Physical Address	0x4AE0 7518	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO8 service requests.		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_GPIO8_IRQ2_EVE2	WKUPDEP_GPIO8_IRQ2_EVE1	WKUPDEP_GPIO8_IRQ2_DSP2	WKUPDEP_GPIO8_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO8_IRQ2_DSP1	WKUPDEP_GPIO8_IRQ2_IPU2	WKUPDEP_GPIO8_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO8_IRQ1_EVE2	WKUPDEP_GPIO8_IRQ1_EVE1	WKUPDEP_GPIO8_IRQ1_DSP2	WKUPDEP_GPIO8_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO8_IRQ1_DSP1	WKUPDEP_GPIO8_IRQ1_IPU2	WKUPDEP_GPIO8_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO8_IRQ2_EVE2	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO8_IRQ2_EVE1	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO8_IRQ2_DSP2	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO8_IRQ2_IPU1	Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO8_IRQ2_DSP1	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
11	WKUPDEP_GPIO8_IRQ2_IPU2	Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO8_IRQ2_MPU	Wakeup dependency from GPIO8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_GPIO8_IRQ1_EVE2	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO8_IRQ1_EVE1	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO8_IRQ1_DSP2	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO8_IRQ1_IPU1	Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO8_IRQ1_DSP1	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO8_IRQ1_IPU2	5Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO8_IRQ1_MPU	Wakeup dependency from GPIO8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1488. Register Call Summary for Register PM_L4PER_GPIO8_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

Table 3-1488. Register Call Summary for Register PM_L4PER_GPIO8_WKDEP (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [18]

Table 3-1489. RM_L4PER_GPIO8_CONTEXT

Address Offset	0x0000 011C		
Physical Address	0x4AE0 751C	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO8 context statuses. [warm reset insensitive]		
Type	RW		

31:2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1490. Register Call Summary for Register RM_L4PER_GPIO8_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1491. PM_L4PER_MMC3_WKDEP

Address Offset	0x0000 0120		
Physical Address	0x4AE0 7520	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on MMC3 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_MMC3_EVE2	WKUPDEP_MMC3_EVE1	WKUPDEP_MMC3_DSP2	WKUPDEP_MMC3_IPU1	WKUPDEP_MMC3_SDMA	WKUPDEP_MMC3_DSP1	WKUPDEP_MMC3_IPU2	WKUPDEP_MMC3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MMC3_EVE2	Wakeup dependency from MMC3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC3_EVE1	Wakeup dependency from MMC3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MMC3_DSP2	Wakeup dependency from MMC3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MMC3_IPU1	Wakeup dependency from MMC3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC3_SDMA	Wakeup dependency from MMC3 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC3_DSP1	Wakeup dependency from MMC3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC3_IPU2	Wakeup dependency from MMC3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MMC3_MPU	Wakeup dependency from MMC3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1492. Register Call Summary for Register PM_L4PER_MMC3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1493. RM_L4PER_MMC3_CONTEXT

Address Offset	0x0000 0124

Table 3-1493. RM_L4PER_MMC3_CONTEXT (continued)

Physical Address	0x4AE0 7524	**Instance**	L4PER_PRM
Description	This register contains dedicated MMC3 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_NONRETAINED_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1494. Register Call Summary for Register RM_L4PER_MMC3_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1495. PM_L4PER_MMC4_WKDEP

Address Offset	0x0000 0128		
Physical Address	0x4AE0 7528	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on MMC4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_MMC4_EVE2	WKUPDEP_MMC4_EVE1	WKUPDEP_MMC4_DSP2	WKUPDEP_MMC4_IPU1	WKUPDEP_MMC4_SDMA	WKUPDEP_MMC4_DSP1	WKUPDEP_MMC4_IPU2	WKUPDEP_MMC4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MMC4_EVE2	Wakeup dependency from MMC4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC4_EVE1	Wakeup dependency from MMC4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MMC4_DSP2	Wakeup dependency from MMC4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MMC4_IPU1	Wakeup dependency from MMC4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC4_SDMA	Wakeup dependency from MMC4 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC4_DSP1	Wakeup dependency from MMC4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC4_IPU2	Wakeup dependency from MMC4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_MMC4_MPU	Wakeup dependency from MMC4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1496. Register Call Summary for Register PM_L4PER_MMC4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1497. RM_L4PER_MMC4_CONTEXT

Address Offset	0x0000 012C		
Physical Address	0x4AE0 752C	**Instance**	L4PER_PRM
Description	This register contains dedicated MMC4 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_NONRETAINED_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1498. Register Call Summary for Register RM_L4PER_MMC4_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

Table 3-1498. Register Call Summary for Register RM_L4PER_MMC4_CONTEXT (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1499. PM_L4PER_TIMER16_WKDEP

Address Offset	0x0000 0130		
Physical Address	0x4AE0 7530	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER16 service requests.		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER16_EVE2	WKUPDEP_TIMER16_EVE1	WKUPDEP_TIMER16_DSP2	WKUPDEP_TIMER16_IPU1	RESERVED	WKUPDEP_TIMER16_DSP1	WKUPDEP_TIMER16_IPU2	WKUPDEP_TIMER16_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER16_EVE2	Wakeup dependency from TIMER16 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER16_EVE1	Wakeup dependency from TIMER16 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER16_DSP2	Wakeup dependency from TIMER16 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER16_IPU1	Wakeup dependency from TIMER16 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER16_DSP1	Wakeup dependency from TIMER16 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_TIMER16_IPU2	Wakeup dependency from TIMER16 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER16_MPU	6Wakeup dependency from TIMER16 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1500. Register Call Summary for Register PM_L4PER_TIMER16_WKDEP

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1501. RM_L4PER3_TIMER16_CONTEXT

Address Offset	0x0000 0134		
Physical Address	0x4AE0 7534	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER16 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1502. Register Call Summary for Register RM_L4PER3_TIMER16_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1503. PM_L4PER2_QSPI_WKDEP

Address Offset	0x0000 0138		
Physical Address	0x4AE0 7538	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on QSPI service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_QSPI_EVE2	WKUPDEP_QSPI_EVE1	WKUPDEP_QSPI_DSP2	WKUPDEP_QSPI_IPU1	RESERVED	WKUPDEP_QSPI_DSP1	WKUPDEP_QSPI_IPU2	WKUPDEP_QSPI_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_QSPI_EVE2	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_QSPI_EVE1	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_QSPI_DSP2	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_QSPI_IPU1	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_QSPI_DSP1	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_QSPI_IPU2	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_QSPI_MPU	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1504. Register Call Summary for Register PM_L4PER2_QSPI_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

Table 3-1504. Register Call Summary for Register PM_L4PER2_QSPI_WKDEP (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [9]

Table 3-1505. RM_L4PER2_QSPI_CONTEXT

Address Offset	0x0000 013C		
Physical Address	0x4AE0 753C	**Instance**	L4PER_PRM
Description	This register contains dedicated QSPI context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1506. Register Call Summary for Register RM_L4PER2_QSPI_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1507. PM_L4PER_UART1_WKDEP

Address Offset	0x0000 0140		
Physical Address	0x4AE0 7540	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_UART1_EVE2	WKUPDEP_UART1_EVE1	WKUPDEP_UART1_DSP2	WKUPDEP_UART1_IPU1	WKUPDEP_UART1_SDMA	WKUPDEP_UART1_DSP1	WKUPDEP_UART1_IPU2	WKUPDEP_UART1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART1_EVE2	Wakeup dependency from UART1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART1_EVE1	Wakeup dependency from UART1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART1_DSP2	Wakeup dependency from UART1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART1_IPU1	Wakeup dependency from UART1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART1_SDMA	Wakeup dependency from UART1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART1_DSP1	Wakeup dependency from UART1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART1_IPU2	Wakeup dependency from UART1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART1_MPU	Wakeup dependency from UART1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1508. Register Call Summary for Register PM_L4PER_UART1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1509. RM_L4PER_UART1_CONTEXT

Address Offset	0x0000 0144		
Physical Address	0x4AE0 7544	**Instance**	L4PER_PRM

Table 3-1509. RM_L4PER_UART1_CONTEXT (continued)

Description	This register contains dedicated UART1 context statuses. [warm reset insensitive]
Type	RW

31–9	8	7–2	1	0
RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1510. Register Call Summary for Register RM_L4PER_UART1_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1511. PM_L4PER_UART2_WKDEP

Address Offset	0x0000 0148		
Physical Address	0x4AE0 7548	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_UART2_EVE2	WKUPDEP_UART2_EVE1	WKUPDEP_UART2_DSP2	WKUPDEP_UART2_IPU1	WKUPDEP_UART2_SDMA	WKUPDEP_UART2_DSP1	WKUPDEP_UART2_IPU2	WKUPDEP_UART2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART2_EVE2	Wakeup dependency from UART2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART2_EVE1	Wakeup dependency from UART2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART2_DSP2	Wakeup dependency from UART2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART2_IPU1	Wakeup dependency from UART2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART2_SDMA	2Wakeup dependency from UART2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART2_DSP1	Wakeup dependency from UART2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART2_IPU2	Wakeup dependency from UART2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_UART2_MPU	Wakeup dependency from UART2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1512. Register Call Summary for Register PM_L4PER_UART2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1513. RM_L4PER_UART2_CONTEXT

Address Offset	0x0000 014C		
Physical Address	0x4AE0 754C	**Instance**	L4PER_PRM
Description	This register contains dedicated UART2 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1514. Register Call Summary for Register RM_L4PER_UART2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1515. PM_L4PER_UART3_WKDEP

Address Offset	0x0000 0150		
Physical Address	0x4AE0 7550	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART3 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_UART3_EVE2	WKUPDEP_UART3_EVE1	WKUPDEP_UART3_DSP2	WKUPDEP_UART3_IPU1	WKUPDEP_UART3_SDMA	WKUPDEP_UART3_DSP1	WKUPDEP_UART3_IPU2	WKUPDEP_UART3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART3_EVE2	Wakeup dependency from UART3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART3_EVE1	Wakeup dependency from UART3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART3_DSP2	Wakeup dependency from UART3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART3_IPU1	Wakeup dependency from UART3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART3_SDMA	Wakeup dependency from UART3 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART3_DSP1	Wakeup dependency from UART3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_UART3_IPU2	Wakeup dependency from UART3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART3_MPU	Wakeup dependency from UART3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1516. Register Call Summary for Register PM_L4PER_UART3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1517. RM_L4PER_UART3_CONTEXT

Address Offset	0x0000 0154		
Physical Address	0x4AE0 7554	**Instance**	L4PER_PRM
Description	This register contains dedicated UART3 context statuses. [warm reset insensitive]		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>LOSTMEM_RETAINED_BANK</td><td colspan="6">RESERVED</td><td>LOSTCONTEXT_RFF</td><td>RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1518. Register Call Summary for Register RM_L4PER_UART3_CONTEXT

Power Management Functional Description • PD_L4PER Description: [0]
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1519. PM_L4PER_UART4_WKDEP

Address Offset	0x0000 0158		
Physical Address	0x4AE0 7558	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART4 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_UART4_EVE2	WKUPDEP_UART4_EVE1	WKUPDEP_UART4_DSP2	WKUPDEP_UART4_IPU1	WKUPDEP_UART4_SDMA	WKUPDEP_UART4_DSP1	WKUPDEP_UART4_IPU2	WKUPDEP_UART4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART4_EVE2	Wakeup dependency from UART4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART4_EVE1	Wakeup dependency from UART4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART4_DSP2	Wakeup dependency from UART4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART4_IPU1	Wakeup dependency from UART4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART4_SDMA	Wakeup dependency from UART4 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_UART4_DSP1	Wakeup dependency from UART4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART4_IPU2	Wakeup dependency from UART4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART4_MPU	Wakeup dependency from UART4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1520. Register Call Summary for Register PM_L4PER_UART4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1521. RM_L4PER_UART4_CONTEXT

Address Offset	0x0000 015C		
Physical Address	0x4AE0 755C	**Instance**	L4PER_PRM
Description	This register contains dedicated UART4 context statuses. [warm reset insensitive]		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>LOSTMEM_RETAINED_BANK</td><td colspan="6">RESERVED</td><td>LOSTCONTEXT_RFF</td><td>RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1522. Register Call Summary for Register RM_L4PER_UART4_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1523. PM_L4PER2_MCASP2_WKDEP

Address Offset	0x0000 0160		
Physical Address	0x4AE0 7560	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP2 service requests.		
Type	RW		

31:16	15	14	13	12	11:10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_MCASP2_DMA_DSP2	RESERVED	WKUPDEP_MCASP2_DMA_SDMA	WKUPDEP_MCASP2_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_MCASP2_IRQ_EVE2	WKUPDEP_MCASP2_IRQ_EVE1	WKUPDEP_MCASP2_IRQ_DSP2	WKUPDEP_MCASP2_IRQ_IPU1	RESERVED	WKUPDEP_MCASP2_IRQ_DSP1	WKUPDEP_MCASP2_IRQ_IPU2	WKUPDEP_MCASP2_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP2_DMA_DSP2	Wakeup dependency from McASP2 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP2_DMA_SDMA	Wakeup dependency from McASP2 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP2_DMA_DSP1	Wakeup dependency from McASP2 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCASP2_IRQ_EVE2	Wakeup dependency from McASP2 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP2_IRQ_EVE1	Wakeup dependency from McASP2 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP2_IRQ_DSP2	Wakeup dependency from McASP2 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP2_IRQ_IPU1	Wakeup dependency from McASP2 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP2_IRQ_DSP1	Wakeup dependency from McASP2 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP2_IRQ_IPU2	Wakeup dependency from McASP2 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP2_IRQ_MPU	Wakeup dependency from McASP2 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1524. Register Call Summary for Register PM_L4PER2_MCASP2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1525. RM_L4PER2_MCASP2_CONTEXT

Address Offset	0x0000 0164		
Physical Address	0x4AE0 7564	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP2 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1526. Register Call Summary for Register RM_L4PER2_MCASP2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1527. PM_L4PER2_MCASP3_WKDEP

Address Offset	0x0000 0168		
Physical Address	0x4AE0 7568	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP3 service requests.		
Type	RW		

31–16	15	14	13	12	11–10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_MCASP3_DMA_DSP2	RESERVED	WKUPDEP_MCASP3_DMA_SDMA	WKUPDEP_MCASP3_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_MCASP3_IRQ_EVE2	WKUPDEP_MCASP3_IRQ_EVE1	WKUPDEP_MCASP3_IRQ_DSP2	WKUPDEP_MCASP3_IRQ_IPU1	RESERVED	WKUPDEP_MCASP3_IRQ_DSP1	WKUPDEP_MCASP3_IRQ_IPU2	WKUPDEP_MCASP3_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP3_DMA_DSP2	Wakeup dependency from McASP3 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13	WKUPDEP_MCASP3_DMA_SDMA	Wakeup dependency from McASP3 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP3_DMA_DSP1	3Wakeup dependency from McASP3 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCASP3_IRQ_EVE2	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP3_IRQ_EVE1	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP3_IRQ_DSP2	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP3_IRQ_IPU1	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP3_IRQ_DSP1	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP3_IRQ_IPU2	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP3_IRQ_MPU	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1528. Register Call Summary for Register PM_L4PER2_MCASP3_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1529. RM_L4PER2_MCASP3_CONTEXT

Address Offset	0x0000 016C		
Physical Address	0x4AE0 756C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP3 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1530. Register Call Summary for Register RM_L4PER2_MCASP3_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1531. PM_L4PER_UART5_WKDEP

Address Offset	0x0000 0170		
Physical Address	0x4AE0 7570	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_UART5_EVE2	WKUPDEP_UART5_EVE1	WKUPDEP_UART5_DSP2	WKUPDEP_UART5_IPU1	WKUPDEP_UART5_SDMA	WKUPDEP_UART5_DSP1	WKUPDEP_UART5_IPU2	WKUPDEP_UART5_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART5_EVE2	Wakeup dependency from UART5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART5_EVE1	Wakeup dependency from UART5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART5_DSP2	Wakeup dependency from UART5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART5_IPU1	Wakeup dependency from UART5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART5_SDMA	Wakeup dependency from UART5 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART5_DSP1	Wakeup dependency from UART5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART5_IPU2	Wakeup dependency from UART5 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_UART5_MPU	Wakeup dependency from UART5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1532. Register Call Summary for Register PM_L4PER_UART5_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1533. RM_L4PER_UART5_CONTEXT

Address Offset	0x0000 0174		
Physical Address	0x4AE0 7574	**Instance**	L4PER_PRM
Description	This register contains dedicated UART5 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1534. Register Call Summary for Register RM_L4PER_UART5_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1535. PM_L4PER2_MCASP5_WKDEP

Address Offset	0x0000 0178		
Physical Address	0x4AE0 7578	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP5 service requests.		
Type	RW		

31:16	15	14	13	12	11:10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_MCASP5_DMA_DSP2	RESERVED	WKUPDEP_MCASP5_DMA_SDMA	WKUPDEP_MCASP5_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_MCASP5_IRQ_EVE2	WKUPDEP_MCASP5_IRQ_EVE1	WKUPDEP_MCASP5_IRQ_DSP2	WKUPDEP_MCASP5_IRQ_IPU1	RESERVED	WKUPDEP_MCASP5_IRQ_DSP1	WKUPDEP_MCASP5_IRQ_IPU2	WKUPDEP_MCASP5_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP5_DMA_DSP2	Wakeup dependency from McASP5 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP5_DMA_SDMA	Wakeup dependency from McASP5 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP5_DMA_DSP1	Wakeup dependency from McASP5 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCASP5_IRQ_EVE2	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP5_IRQ_EVE1	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_MCASP5_IRQ_DSP2	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP5_IRQ_IPU1	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP5_IRQ_DSP1	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP5_IRQ_IPU2	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP5_IRQ_MPU	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1536. Register Call Summary for Register PM_L4PER2_MCASP5_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1537. RM_L4PER2_MCASP5_CONTEXT

Address Offset	0x0000 017C		
Physical Address	0x4AE0 757C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP5 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1538. Register Call Summary for Register RM_L4PER2_MCASP5_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1539. PM_L4PER2_MCASP6_WKDEP

Address Offset	0x0000 0180		
Physical Address	0x4AE0 7580	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP6 service requests.		
Type	RW		

31-16	15	14	13	12	11-10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_MCASP6_DMA_DSP2	RESERVED	WKUPDEP_MCASP6_DMA_SDMA	WKUPDEP_MCASP6_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_MCASP6_IRQ_EVE2	WKUPDEP_MCASP6_IRQ_EVE1	WKUPDEP_MCASP6_IRQ_DSP2	WKUPDEP_MCASP6_IRQ_IPU1	RESERVED	WKUPDEP_MCASP6_IRQ_DSP1	WKUPDEP_MCASP6_IRQ_IPU2	WKUPDEP_MCASP6_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP6_DMA_DSP2	Wakeup dependency from McASP6 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP6_DMA_SDMA	Wakeup dependency from McASP6 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP6_DMA_DSP1	Wakeup dependency from McASP6 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RESERVED		R	0x0
7	WKUPDEP_MCASP6_IRQ_EVE2	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP6_IRQ_EVE1	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP6_IRQ_DSP2	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP6_IRQ_IPU1	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP6_IRQ_DSP1	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP6_IRQ_IPU2	Wakeup dependency from McASP6 (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP6_IRQ_MPU	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1540. Register Call Summary for Register PM_L4PER2_MCASP6_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1541. RM_L4PER2_MCASP6_CONTEXT

Address Offset	0x0000 0184		
Physical Address	0x4AE0 7584	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1542. Register Call Summary for Register RM_L4PER2_MCASP6_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1543. PM_L4PER2_MCASP7_WKDEP

Address Offset	0x0000 0188		
Physical Address	0x4AE0 7588	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP7 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP7_DMA_DSP2	RESERVED	WKUPDEP_MCASP7_DMA_SDMA	WKUPDEP_MCASP7_DMA_DSP1	RESERVED		RESERVED	RESERVED	WKUPDEP_MCASP7_IRQ_EVE2	WKUPDEP_MCASP7_IRQ_EVE1	WKUPDEP_MCASP7_IRQ_DSP2	WKUPDEP_MCASP7_IRQ_IPU1	RESERVED	WKUPDEP_MCASP7_IRQ_DSP1	WKUPDEP_MCASP7_IRQ_IPU2	WKUPDEP_MCASP7_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP7_DMA_DSP2	Wakeup dependency from McASP7 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13	WKUPDEP_MCASP7_DMA_SDMA	Wakeup dependency from McASP7 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP7_DMA_DSP1	Wakeup dependency from McASP7 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCASP7_IRQ_EVE2	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP7_IRQ_EVE1	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP7_IRQ_DSP2	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP7_IRQ_IPU1	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP7_IRQ_DSP1	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP7_IRQ_IPU2	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP7_IRQ_MPU	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1544. Register Call Summary for Register PM_L4PER2_MCASP7_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1545. RM_L4PER2_MCASP7_CONTEXT

Address Offset	0x0000 018C		
Physical Address	0x4AE0 758C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP7 context statuses. [warm reset insensitive]		
Type	RW		

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1546. Register Call Summary for Register RM_L4PER2_MCASP7_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1547. PM_L4PER2_MCASP8_WKDEP

Address Offset	0x0000 0190		
Physical Address	0x4AE0 7590	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP8 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	WKUPDEP_MCASP8_DMA_DSP2	RESERVED	WKUPDEP_MCASP8_DMA_SDMA	WKUPDEP_MCASP8_DMA_DSP1	RESERVED	RESERVED	RESERVED	WKUPDEP_MCASP8_IRQ_EVE2	WKUPDEP_MCASP8_IRQ_EVE1	WKUPDEP_MCASP8_IRQ_DSP2	WKUPDEP_MCASP8_IRQ_IPU1	RESERVED	WKUPDEP_MCASP8_IRQ_DSP1	WKUPDEP_MCASP8_IRQ_IPU2	WKUPDEP_MCASP8_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP8_DMA_DSP2	Wakeup dependency from McASP8 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP8_DMA_SDMA	Wakeup dependency from McASP8 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP8_DMA_DSP1	Wakeup dependency from McASP8 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCASP8_IRQ_EVE2	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP8_IRQ_EVE1	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP8_IRQ_DSP2	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP8_IRQ_IPU1	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	WKUPDEP_MCASP8_IRQ_DSP1	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP8_IRQ_IPU2	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP8_IRQ_MPU	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1548. Register Call Summary for Register PM_L4PER2_MCASP8_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1549. RM_L4PER2_MCASP8_CONTEXT

Address Offset	0x0000 0194		
Physical Address	0x4AE0 7594	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP8 context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1550. Register Call Summary for Register RM_L4PER2_MCASP8_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

Table 3-1550. Register Call Summary for Register RM_L4PER2_MCASP8_CONTEXT (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1551. PM_L4PER2_MCASP4_WKDEP

Address Offset	0x0000 0198		
Physical Address	0x4AE0 7598	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP4 service requests.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td>WKUPDEP_MCASP4_DMA_DSP2</td><td>RESERVED</td><td>WKUPDEP_MCASP4_DMA_SDMA</td><td>WKUPDEP_MCASP4_DMA_DSP1</td><td colspan="2">RESERVED</td><td>RESERVED</td><td>RESERVED</td><td>WKUPDEP_MCASP4_IRQ_EVE2</td><td>WKUPDEP_MCASP4_IRQ_EVE1</td><td>WKUPDEP_MCASP4_IRQ_DSP2</td><td>WKUPDEP_MCASP4_IRQ_IPU1</td><td>RESERVED</td><td>WKUPDEP_MCASP4_IRQ_DSP1</td><td>WKUPDEP_MCASP4_IRQ_IPU2</td><td>WKUPDEP_MCASP4_IRQ_MPU</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP4_DMA_DSP2	Wakeup dependency from McASP4 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP4_DMA_SDMA	Wakeup dependency from McASP4 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP4_DMA_DSP1	Wakeup dependency from McASP4 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_MCASP4_IRQ_EVE2	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_MCASP4_IRQ_EVE1	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP4_IRQ_DSP2	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP4_IRQ_IPU1	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP4_IRQ_DSP1	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP4_IRQ_IPU2	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP4_IRQ_MPU	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1552. Register Call Summary for Register PM_L4PER2_MCASP4_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

PRCM Register Manual

- L4PER_PRM Register Summary: [12]

Table 3-1553. RM_L4PER2_MCASP4_CONTEXT

Address Offset	0x0000 019C		
Physical Address	0x4AE0 759C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1554. Register Call Summary for Register RM_L4PER2_MCASP4_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1555. RM_L4SEC_AES1_CONTEXT

Address Offset	0x0000 01A4		
Physical Address	0x4AE0 75A4	**Instance**	L4PER_PRM
Description	This register contains dedicated AES1 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1556. Register Call Summary for Register RM_L4SEC_AES1_CONTEXT

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1557. RM_L4SEC_AES2_CONTEXT

Address Offset	0x0000 01AC

Table 3-1557. RM_L4SEC_AES2_CONTEXT (continued)

Physical Address	0x4AE0 75AC	**Instance**	L4PER_PRM
Description	This register contains dedicated AES2 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1558. Register Call Summary for Register RM_L4SEC_AES2_CONTEXT

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1559. RM_L4SEC_DES3DES_CONTEXT

Address Offset	0x0000 01B4		
Physical Address	0x4AE0 75B4	**Instance**	L4PER_PRM
Description	This register contains dedicated DES3DES context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1560. Register Call Summary for Register RM_L4SEC_DES3DES_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1561. RM_L4SEC_FPKA_CONTEXT

Address Offset	0x0000 01BC		
Physical Address	0x4AE0 75BC	**Instance**	L4PER_PRM
Description	This register contains dedicated FPKA context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_NONRETAINED_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1562. Register Call Summary for Register RM_L4SEC_FPKA_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1563. RM_L4SEC_RNG_CONTEXT

Address Offset	0x0000 01C4		
Physical Address	0x4AE0 75C4	**Instance**	L4PER_PRM
Description	This register contains dedicated RNG context statuses. [warm reset insensitive]		

Table 3-1563. RM_L4SEC_RNG_CONTEXT (continued)

Type	RW

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1564. Register Call Summary for Register RM_L4SEC_RNG_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1565. RM_L4SEC_SHA2MD51_CONTEXT

Address Offset	0x0000 01CC		
Physical Address	0x4AE0 75CC	**Instance**	L4PER_PRM
Description	This register contains dedicated SHA2MD51 context statuses. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1566. Register Call Summary for Register RM_L4SEC_SHA2MD51_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1567. PM_L4PER2_UART7_WKDEP

Address Offset	0x0000 01D0		
Physical Address	0x4AE0 75D0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART7 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_UART7_EVE2	WKUPDEP_UART7_EVE1	WKUPDEP_UART7_DSP2	WKUPDEP_UART7_IPU1	WKUPDEP_UART7_SDMA	WKUPDEP_UART7_DSP1	WKUPDEP_UART7_IPU2	WKUPDEP_UART7_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART7_EVE2	Wakeup dependency from UART7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART7_EVE1	Wakeup dependency from UART7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART7_DSP2	Wakeup dependency from UART7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART7_IPU1	Wakeup dependency from UART7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART7_SDMA	Wakeup dependency from UART7 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_UART7_DSP1	Wakeup dependency from UART7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART7_IPU2	Wakeup dependency from UART7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART7_MPU	Wakeup dependency from UART7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1568. Register Call Summary for Register PM_L4PER2_UART7_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1569. RM_L4PER2_UART7_CONTEXT

Address Offset	0x0000 01D4		
Physical Address	0x4AE0 75D4	**Instance**	L4PER_PRM
Description	This register contains dedicated UART7 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1570. Register Call Summary for Register RM_L4PER2_UART7_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1571. RM_L4SEC_DMA_CRYPTO_CONTEXT

Address Offset	0x0000 01DC		
Physical Address	0x4AE0 75DC	**Instance**	L4PER_PRM
Description	This register contains dedicated DMA_CRYPTO context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1572. Register Call Summary for Register RM_L4SEC_DMA_CRYPTO_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

Table 3-1572. Register Call Summary for Register RM_L4SEC_DMA_CRYPTO_CONTEXT (continued)

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1573. PM_L4PER2_UART8_WKDEP

Address Offset	0x0000 01E0		
Physical Address	0x4AE0 75E0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART8 service requests.		
Type	RW		

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_UART8_EVE2	WKUPDEP_UART8_EVE1	WKUPDEP_UART8_DSP2	WKUPDEP_UART8_IPU1	WKUPDEP_UART8_SDMA	WKUPDEP_UART8_DSP1	WKUPDEP_UART8_IPU2	WKUPDEP_UART8_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART8_EVE2	Wakeup dependency from UART8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART8_EVE1	Wakeup dependency from UART8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART8_DSP2	Wakeup dependency from UART8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART8_IPU1	Wakeup dependency from UART8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART8_SDMA	Wakeup dependency from UART8 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_UART8_DSP1	Wakeup dependency from UART8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART8_IPU2	Wakeup dependency from UART8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART8_MPU	Wakeup dependency from UART8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1574. Register Call Summary for Register PM_L4PER2_UART8_WKDEP

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1575. RM_L4PER2_UART8_CONTEXT

Address Offset	0x0000 01E4		
Physical Address	0x4AE0 75E4	**Instance**	L4PER_PRM
Description	This register contains dedicated UART8 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:2	1	0
RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
0	RESERVED		R	0x0

Table 3-1576. Register Call Summary for Register RM_L4PER2_UART8_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1577. PM_L4PER2_UART9_WKDEP

Address Offset	0x0000 01E8		
Physical Address	0x4AE0 75E8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART9 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_UART9_EVE2	WKUPDEP_UART9_EVE1	WKUPDEP_UART9_DSP2	WKUPDEP_UART9_IPU1	WKUPDEP_UART9_SDMA	WKUPDEP_UART9_DSP1	WKUPDEP_UART9_IPU2	WKUPDEP_UART9_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART9_EVE2	Wakeup dependency from UART9 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART9_EVE1	Wakeup dependency from UART9 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART9_DSP2	Wakeup dependency from UART9 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART9_IPU1	Wakeup dependency from UART9 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3	WKUPDEP_UART9_SDMA	Wakeup dependency from UART9 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART9_DSP1	Wakeup dependency from UART9 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART9_IPU2	Wakeup dependency from UART9 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART9_MPU	Wakeup dependency from UART9 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1578. Register Call Summary for Register PM_L4PER2_UART9_WKDEP

PRCM Register Manual

- L4PER_PRM Register Summary: [0]

Table 3-1579. RM_L4PER2_UART9_CONTEXT

Address Offset	0x0000 01EC		
Physical Address	0x4AE0 75EC	**Instance**	L4PER_PRM
Description	This register contains dedicated UART9 context statuses. [warm reset insensitive]		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>LOSTMEM_RETAINED_BANK</td><td colspan="6">RESERVED</td><td>LOSTCONTEXT_RFF</td><td>RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1580. Register Call Summary for Register RM_L4PER2_UART9_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1581. PM_L4PER2_DCAN2_WKDEP

Address Offset	0x0000 01F0		
Physical Address	0x4AE0 75F0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCAN2 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_DCAN2_EVE2	WKUPDEP_DCAN2_EVE1	WKUPDEP_DCAN2_DSP2	WKUPDEP_DCAN2_IPU1	WKUPDEP_DCAN2_SDMA	WKUPDEP_DCAN2_DSP1	WKUPDEP_DCAN2_IPU2	WKUPDEP_DCAN2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_DCAN2_EVE2	Wakeup dependency from DCAN2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCAN2_EVE1	Wakeup dependency from DCAN2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCAN2_DSP2	Wakeup dependency from DCAN2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_DCAN2_IPU1	Wakeup dependency from DCAN2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_DCAN2_SDMA	Wakeup dependency from DCAN2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_DCAN2_DSP1	Wakeup dependency from DCAN2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCAN2_IPU2	Wakeup dependency from DCAN2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCAN2_MPU	Wakeup dependency from DCAN2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1582. Register Call Summary for Register PM_L4PER2_DCAN2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7]

PRCM Register Manual

- L4PER_PRM Register Summary: [10]

Table 3-1583. RM_L4PER2_DCAN2_CONTEXT

Address Offset	0x0000 01F4		
Physical Address	0x4AE0 75F4	**Instance**	L4PER_PRM
Description	This register contains dedicated DCAN2 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_DCAN_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	LOSTMEM_DCAN_BANK	Specify if memory-based context in DCAN memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1584. Register Call Summary for Register RM_L4PER2_DCAN2_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1585. RM_L4SEC_SHA2MD52_CONTEXT

Address Offset	0x0000 01FC		
Physical Address	0x4AE0 75FC	**Instance**	L4PER_PRM
Description	This register contains dedicated SHA2MD52 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1586. Register Call Summary for Register RM_L4SEC_SHA2MD52_CONTEXT

Power Management Functional Description

- PD_L4PER Description: [0]

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

3.13.44 MPU_PRM Registers

3.13.44.1 MPU_PRM Register Summary

Table 3-1587. MPU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRM Physical Address L4_WKUP Interconnect
PM_MPU_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6300
PM_MPU_PWRSTST	RW	32	0x0000 0004	0x4AE0 6304
RM_MPU_MPU_CONTEXT	RW	32	0x0000 0024	0x4AE0 6324

3.13.44.2 MPU_PRM Register Description

Table 3-1588. PM_MPU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6300	**Instance**	MPU_PRM
Description	This register controls the MPU domain power state to reach upon a domain sleep transition. If the value programmed in this register correspond to a lower power state than the one programmed in MPU-SS for CPU0 and/or CPU1, then value of this register is overwritten in PRCM logic to limit the power state to enter. Note: Even if value of this register is overwritten in PRCM logic, value of this register remains unchanged. - If user programs MPU power domain to go to CSWRET, then he can not program L2$ to OFF mode. Note: Only the MPU Subsystem supports memory retention. MPU subsystem does not support OFF state. Only CPU1 supports FORCED_OFF state with no subsequent recovery to ON/active state - this is very application specific and may not be available in all TI standard software offerings.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td colspan="2">MPU_RAM_ONSTATE</td><td colspan="2">MPU_L2_ONSTATE</td><td colspan="7">RESERVED</td><td>MPU_RAM_RETSTATE</td><td>MPU_L2_RETSTATE</td><td colspan="4">RESERVED</td><td>LOWPOWERSTATECHANGE</td><td>RESERVED</td><td>LOGICRETSTATE</td><td colspan="2">POWERSTATE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	MPU_RAM_ONSTATE	MPU_RAM memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	MPU_L2_ONSTATE	MPU_L2 memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:11	RESERVED		R	0x0
10	MPU_RAM_RETSTATE	MPU_RAM memory state when domain is RETENTION. 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1

Bits	Field Name	Description	Type	Reset
9	MPU_L2_RETSTATE	MPU_L2 memory state when domain is RETENTION. Should always be same as or higher than LogicRETState bit-field. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
8:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change.	R	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	0x1
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1589. Register Call Summary for Register PM_MPU_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4] [5] [6]
- Power State Override: [7] [8] [9] [10]

PRCM Register Manual

- MPU_PRM Register Summary: [11]

Table 3-1590. PM_MPU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6304	**Instance**	MPU_PRM
Description	This register provides a status on the MPU domain current power state. [warm reset insensitive]		
Type	RW		

31–26	25–24	23–21	20	19–10	9–8	7–6	5–3	2	1–0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	MPU_RAMSTATEST	MPU_L2_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	MPU_RAM_STATEST	MPU_RAM memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	MPU_L2_STATEST	MPU_L2 memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
5:3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1591. Register Call Summary for Register PM_MPU_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4] [5]

PRCM Register Manual

- MPU_PRM Register Summary: [6]

Table 3-1592. RM_MPU_MPU_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6324	**Instance**	MPU_PRM
Description	This register contains dedicated MPU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTMEM_MPU_RAM	LOSTMEM_MPU_L2	RESERVED	LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LOSTMEM_MPU_RAM	Specify if memory-based context in MPU_RAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
9	LOSTMEM_MPU_L2	Specify if memory-based context in MPU_L2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_MA_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1593. Register Call Summary for Register RM_MPU_MPU_CONTEXT

Power Management Functional Description

- PD_MPU Description: [0] [1]

PRCM Register Manual

- MPU_PRM Register Summary: [2]

3.13.45 OCP_SOCKET_PRM Registers

3.13.45.1 OCP_SOCKET_PRM Register Summary

Table 3-1594. OCP_SOCKET_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCP_SOCKET_PRM Physical Address L4_WKUP Interconnect
REVISION_PRM	R	32	0x0000 0000	0x4AE0 6000
PRM_IRQSTATUS_MPU	RW	32	0x0000 0010	0x4AE0 6010
PRM_IRQSTATUS_MPU_2	RW	32	0x0000 0014	0x4AE0 6014
PRM_IRQENABLE_MPU	RW	32	0x0000 0018	0x4AE0 6018
PRM_IRQENABLE_MPU_2	RW	32	0x0000 001C	0x4AE0 601C
PRM_IRQSTATUS_IPU2	RW	32	0x0000 0020	0x4AE0 6020
PRM_IRQENABLE_IPU2	RW	32	0x0000 0028	0x4AE0 6028

Table 3-1594. OCP_SOCKET_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	OCP_SOCKET_PRM Physical Address L4_WKUP Interconnect
PRM_IRQSTATUS_DSP1	RW	32	0x0000 0030	0x4AE0 6030
PRM_IRQENABLE_DSP1	RW	32	0x0000 0038	0x4AE0 6038
CM_PRM_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4AE0 6040
PRM_IRQENABLE_DSP2	RW	32	0x0000 0044	0x4AE0 6044
PRM_IRQENABLE_EVE1	RW	32	0x0000 0048	0x4AE0 6048
PRM_IRQENABLE_EVE2	RW	32	0x0000 004C	0x4AE0 604C
RESERVED	RW	32	0x0000 0050	0x4AE0 6050
RESERVED	RW	32	0x0000 0054	0x4AE0 6054
PRM_IRQENABLE_IPU1	RW	32	0x0000 0058	0x4AE0 6058
PRM_IRQSTATUS_DSP2	RW	32	0x0000 005C	0x4AE0 605C
PRM_IRQSTATUS_EVE1	RW	32	0x0000 0060	0x4AE0 6060
PRM_IRQSTATUS_EVE2	RW	32	0x0000 0064	0x4AE0 6064
RESERVED	RW	32	0x0000 0068	0x4AE0 6068
RESERVED	RW	32	0x0000 006C	0x4AE0 606C
PRM_IRQSTATUS_IPU1	RW	32	0x0000 0070	0x4AE0 6070
PRM_DEBUG_CFG1	RW	32	0x0000 00E4	0x4AE0 60E4
PRM_DEBUG_CFG2	RW	32	0x0000 00E8	0x4AE0 60E8
PRM_DEBUG_CFG3	RW	32	0x0000 00EC	0x4AE0 60EC
PRM_DEBUG_CFG	RW	32	0x0000 00F0	0x4AE0 60F0
PRM_DEBUG_OUT	R	32	0x0000 00F4	0x4AE0 60F4

3.13.45.2 OCP_SOCKET_PRM Register Description

Table 3-1595. REVISION_PRM

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6000	**Instance**	OCP_SOCKET_PRM
Description	This register contains the IP revision code for the PRM part of the PRCM		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision number	R	0x- [1]

(1) TI internal data

Table 3-1596. Register Call Summary for Register REVISION_PRM

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1597. PRM_IRQSTATUS_MPU

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6010	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on MPU interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED	DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	RESERVED	IO_ST	TRANSITION_ST	DPLL_DDR_RECALL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	RESERVED		R	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1598. Register Call Summary for Register PRM_IRQSTATUS_MPU

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

Voltage-Management Functional Description

- ABB LDO Enable Sequence: [11] [12] [13]
- ABB LDO Disable Sequence (Entering in Bypass Mode): [14] [15] [16]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [17]

Table 3-1599. PRM_IRQSTATUS_MPU_2

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6014	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on MPU interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31:8	7	6:0
RESERVED	ABB_MPU_DONE_ST	RESERVED

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MPU_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6:0	RESERVED		R	0x0

Table 3-1600. Register Call Summary for Register PRM_IRQSTATUS_MPU_2

Clock Management Functional Description

- DPLL Recalibration: [0]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [1]

Table 3-1601. PRM_IRQENABLE_MPU

Address Offset	0x0000 0018		
Physical Address	0x4AE0 6018	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable MPU interrupt activation.		
Type	RW		

31	30	29	28:12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED	DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	IO_EN	TRANSITION_EN	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
29	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28:12	RESERVED		R	0x0
11	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain) 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1602. Register Call Summary for Register PRM_IRQENABLE_MPU

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1603. PRM_IRQENABLE_MPU_2

Address Offset	0x0000 001C		
Physical Address	0x4AE0 601C	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable MPU interrupt activation.		
Type	RW		

31:8	7	6:0
RESERVED	ABB_MPU_DONE_EN	RESERVED

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6:0	RESERVED		R	0x0

Table 3-1604. Register Call Summary for Register PRM_IRQENABLE_MPU_2

Clock Management Functional Description

- DPLL Recalibration: [0]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [1]

Table 3-1605. PRM_IRQSTATUS_IPU2

Address Offset	0x0000 0020		
Physical Address	0x4AE0 6020	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on IPU2 interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		

Table 3-1605. PRM_IRQSTATUS_IPU2 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	TRANSITION_ST	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1606. Register Call Summary for Register PRM_IRQSTATUS_IPU2

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

Table 3-1606. Register Call Summary for Register PRM_IRQSTATUS_IPU2 (continued)

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1607. PRM_IRQENABLE_IPU2

Address Offset	0x0000 0028		
Physical Address	0x4AE0 6028	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable IPU2 interrupt activationt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	TRANSITION_EN	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain) 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1608. Register Call Summary for Register PRM_IRQENABLE_IPU2

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1609. PRM_IRQSTATUS_DSP1

Address Offset	0x0000 0030		
Physical Address	0x4AE0 6030	**Instance**	OCP_SOCKET_PRM

Table 3-1609. PRM_IRQSTATUS_DSP1 (continued)

Description	This register provides status on DSP1 interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1610. Register Call Summary for Register PRM_IRQSTATUS_DSP1

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

Table 3-1610. Register Call Summary for Register PRM_IRQSTATUS_DSP1 (continued)

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1611. PRM_IRQENABLE_DSP1

Address Offset	0x0000 0038		
Physical Address	0x4AE0 6038	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable DSP1 interrupt activation.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1612. Register Call Summary for Register PRM_IRQENABLE_DSP1

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_USB Recalibration: [6]
- DPLL_EVE Recalibration: [7]
- DPLL_DSP Recalibration: [8]
- DPLL_GMAC Recalibration: [9]
- DPLL_GPU Recalibration: [10]
- DPLL_DDR Recalibration: [11]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [12]

Table 3-1613. CM_PRM_PROFILING_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4AE0 6040	**Instance**	OCP_SOCKET_PRM
Description	This register manages the PRM_PROFILING clock. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. OCP configuration port is not accessible. 0x1: Module is managed automatically by HW along with EMU domain. OCP configuration port is accessible only when EMU domain is on. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-1614. Register Call Summary for Register CM_PRM_PROFILING_CLKCTRL

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1615. PRM_IRQENABLE_DSP2

Address Offset	0x0000 0044		
Physical Address	0x4AE0 6044	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable DSP2 interrupt activation.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1616. Register Call Summary for Register PRM_IRQENABLE_DSP2

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1617. PRM_IRQENABLE_EVE1

Address Offset	0x0000 0048		
Physical Address	0x4AE0 6048	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable EVE1 interrupt activation.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1618. Register Call Summary for Register PRM_IRQENABLE_EVE1

Clock Management Functional Description

- DPLL_PER Recalibration: [0]
- DPLL_CORE Recalibration: [1]
- DPLL_ABE Recalibration: [2]
- DPLL_MPU Recalibration: [3]
- DPLL_IVA Recalibration: [4]
- DPLL_EVE Recalibration: [5]
- DPLL_DSP Recalibration: [6]
- DPLL_GMAC Recalibration: [7]
- DPLL_GPU Recalibration: [8]
- DPLL_DDR Recalibration: [9]

Table 3-1618. Register Call Summary for Register PRM_IRQENABLE_EVE1 (continued)

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [10]

Table 3-1619. PRM_IRQENABLE_EVE2

Address Offset	0x0000 004C		
Physical Address	0x4AE0 604C	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable EVE2 interrupt activation.		
Type	RW		

31	30	29	28	27–13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED	DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1620. Register Call Summary for Register PRM_IRQENABLE_EVE2

Clock Management Functional Description

- DPLL_PER Recalibration: [0]
- DPLL_CORE Recalibration: [1]
- DPLL_ABE Recalibration: [2]
- DPLL_MPU Recalibration: [3]
- DPLL_IVA Recalibration: [4]
- DPLL_EVE Recalibration: [5]
- DPLL_DSP Recalibration: [6]
- DPLL_GMAC Recalibration: [7]
- DPLL_GPU Recalibration: [8]
- DPLL_DDR Recalibration: [9]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [10]

Table 3-1621. PRM_IRQENABLE_IPU1

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6058	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable IPU1 interrupt activation.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED	DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	TRANSITION_EN	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain) 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1622. Register Call Summary for Register PRM_IRQENABLE_IPU1

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1623. PRM_IRQSTATUS_DSP2

Address Offset	0x0000 005C		
Physical Address	0x4AE0 605C	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on DSP interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1624. Register Call Summary for Register PRM_IRQSTATUS_DSP2

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1625. PRM_IRQSTATUS_EVE1

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6060	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on EVE interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27–13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED	DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1626. Register Call Summary for Register PRM_IRQSTATUS_EVE1

Clock Management Functional Description
- DPLL_PER Recalibration: [0]
- DPLL_CORE Recalibration: [1]
- DPLL_ABE Recalibration: [2]
- DPLL_MPU Recalibration: [3]
- DPLL_IVA Recalibration: [4]
- DPLL_EVE Recalibration: [5]
- DPLL_DSP Recalibration: [6]
- DPLL_GMAC Recalibration: [7]
- DPLL_GPU Recalibration: [8]
- DPLL_DDR Recalibration: [9]

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [10]

Table 3-1627. PRM_IRQSTATUS_EVE2

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6064	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on EVE interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1628. Register Call Summary for Register PRM_IRQSTATUS_EVE2

Clock Management Functional Description

- DPLL_PER Recalibration: [0]
- DPLL_CORE Recalibration: [1]
- DPLL_ABE Recalibration: [2]
- DPLL_MPU Recalibration: [3]
- DPLL_IVA Recalibration: [4]
- DPLL_EVE Recalibration: [5]
- DPLL_DSP Recalibration: [6]
- DPLL_GMAC Recalibration: [7]
- DPLL_GPU Recalibration: [8]
- DPLL_DDR Recalibration: [9]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [10]

Table 3-1629. PRM_IRQSTATUS_IPU1

Address Offset	0x0000 0070		
Physical Address	0x4AE0 6070	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on IPU1 interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>ABB_MPU_DONE_ST</td><td>ABB_IVA_DONE_ST</td><td>ABB_DSPEVE_DONE_ST</td><td>ABB_GPU_DONE_ST</td><td colspan="15">RESERVED</td><td>DPLL_EVE_RECAL_ST</td><td>DPLL_DSP_RECAL_ST</td><td>FORCEWKUP_ST</td><td>IO_ST</td><td>TRANSITION_ST</td><td>DPLL_DDR_RECAL_ST</td><td>DPLL_GPU_RECAL_ST</td><td>DPLL_GMAC_RECAL_ST</td><td>DPLL_ABE_RECAL_ST</td><td>DPLL_PER_RECAL_ST</td><td>DPLL_IVA_RECAL_ST</td><td>DPLL_MPU_RECAL_ST</td><td>DPLL_CORE_RECAL_ST</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1630. Register Call Summary for Register PRM_IRQSTATUS_IPU1

Clock Management Functional Description

- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_ABE Recalibration: [3]
- DPLL_MPU Recalibration: [4]
- DPLL_IVA Recalibration: [5]
- DPLL_EVE Recalibration: [6]
- DPLL_DSP Recalibration: [7]
- DPLL_GMAC Recalibration: [8]
- DPLL_GPU Recalibration: [9]
- DPLL_DDR Recalibration: [10]

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [11]

Table 3-1631. PRM_DEBUG_CFG1

Address Offset	0x0000 00E4		
Physical Address	0x4AE0 60E4	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									SEL1						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL1	Internal signal block select for debug word byte-1	RW	0x0

Table 3-1632. Register Call Summary for Register PRM_DEBUG_CFG1

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1633. PRM_DEBUG_CFG2

Address Offset	0x0000 00E8		
Physical Address	0x4AE0 60E8	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SEL2								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL2	Internal signal block select for debug word byte-2	RW	0x0

Table 3-1634. Register Call Summary for Register PRM_DEBUG_CFG2

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1635. PRM_DEBUG_CFG3

Address Offset	0x0000 00EC		
Physical Address	0x4AE0 60EC	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SEL3								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL3	Internal signal block select for debug word byte-3	RW	0x0

Table 3-1636. Register Call Summary for Register PRM_DEBUG_CFG3

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1637. PRM_DEBUG_CFG

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 60F0	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		

Table 3-1637. PRM_DEBUG_CFG (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SEL0								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL0	Internal signal block select for debug word byte-0	RW	0x0

Table 3-1638. Register Call Summary for Register PRM_DEBUG_CFG

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1639. PRM_DEBUG_OUT

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 60F4	**Instance**	OCP_SOCKET_PRM
Description	This register is used to monitor the PRM's 32 bit HEDEBUG BUS [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUT																															

Bits	Field Name	Description	Type	Reset
31:0	OUTPUT	HW DEBUG OUTPUT	R	0x0

Table 3-1640. Register Call Summary for Register PRM_DEBUG_OUT

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [0]

3.13.46 RTC_PRM Registers

3.13.46.1 RTC_PRM Register Summary

Table 3-1641. RTC_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	RTC_PRM Physical Address L4_WKUP Interconnect
PM_RTC_RTCSS_WKDEP	RW	32	0x0000 0020	0x4AE0 7C60
RM_RTC_RTCSS_CONTEXT	RW	32	0x0000 0024	0x4AE0 7C64

3.13.46.2 RTC_PRM Register Description

Table 3-1642. PM_RTC_RTCSS_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7C60	**Instance**	RTC_PRM
Description	This register controls wakeup dependency based on RTCSS service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_RTC_IRQ2_EVE2	WKUPDEP_RTC_IRQ2_EVE1	WKUPDEP_RTC_IRQ2_DSP2	WKUPDEP_RTC_IRQ2_IPU1	RESERVED	WKUPDEP_RTC_IRQ2_DSP1	WKUPDEP_RTC_IRQ2_IPU2	WKUPDEP_RTC_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_RTC_IRQ1_EVE2	WKUPDEP_RTC_IRQ1_EVE1	WKUPDEP_RTC_IRQ1_DSP2	WKUPDEP_RTC_IRQ1_IPU1	RESERVED	WKUPDEP_RTC_IRQ1_DSP1	WKUPDEP_RTC_IRQ1_IPU2	WKUPDEP_RTC_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_RTC_IRQ2_EVE2	Wakeup dependency from RTCSS module (timer_swakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_RTC_IRQ2_EVE1	Wakeup dependency from RTCSS module (timer_swakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_RTC_IRQ2_DSP2	Wakeup dependency from RTCSS module (timer_swakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_RTC_IRQ2_IPU1	Wakeup dependency from RTCSS module (timer_swakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_RTC_IRQ2_DSP1	Wakeup dependency from RTCSS module (timer_swakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_RTC_IRQ2_IPU2	Wakeup dependency from RTCSS module (timer_swakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_RTC_IRQ2_MPU	Wakeup dependency from RTCSS module (timer_swakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_RTC_IRQ1_EVE2	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_RTC_IRQ1_EVE1	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_RTC_IRQ1_DSP2	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_RTC_IRQ1_IPU1	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_RTC_IRQ1_DSP1	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_RTC_IRQ1_IPU2	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_RTC_IRQ1_MPU	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1643. Register Call Summary for Register PM_RTC_RTCSS_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [9] [10] [11] [12] [13] [14] [15]

PRCM Register Manual

- RTC_PRM Register Summary: [18]

Table 3-1644. RM_RTC_RTCSS_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7C64	**Instance**	RTC_PRM
Description	This register contains dedicated RTCSS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1645. Register Call Summary for Register RM_RTC_RTCSS_CONTEXT

Power Management Functional Description

- PD_RTC Description: [0]

PRCM Register Manual

- RTC_PRM Register Summary: [1]

3.13.47 VPE_PRM Registers

3.13.47.1 VPE_PRM Register Summary

Table 3-1646. VPE_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_PRM Physical Address L4_WKUP Interconnect
PM_VPE_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7C80
PM_VPE_PWRSTST	RW	32	0x0000 0004	0x4AE0 7C84
PM_VPE_VPE_WKDEP	RW	32	0x0000 0020	0x4AE0 7CA0
RM_VPE_VPE_CONTEXT	RW	32	0x0000 0024	0x4AE0 7CA4

3.13.47.2 VPE_PRM Register Description

Table 3-1647. PM_VPE_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7C80	**Instance**	VPE_PRM
Description	This register controls the VPE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VPE_BANK_ONSTATE	RESERVED	VPE_BANK_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	VPE_BANK_ONSTATE	DSP_L1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:9	RESERVED	Reserved	R	0x0
8	VPE_BANK_RETSTATE	Note: Not supported on this device.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED	Reserved	R	0x0
2	LOGICRETSTATE	Note: Not supported on this device.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: Reserved 0x3: ON State	RW	0x3

Table 3-1648. Register Call Summary for Register PM_VPE_PWRSTCTRL

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4]

PRCM Register Manual

- VPE_PRM Register Summary: [5]

Table 3-1649. PM_VPE_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7C84	**Instance**	VPE_PRM
Description	This register provides a status on the VPE domain current power state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	VPE_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	VPE_BANK_STATEST	VPE_BANK memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Reserved 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Reserved 0x2: Reserved 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1650. Register Call Summary for Register PM_VPE_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0] [1] [2] [3] [4]

PRCM Register Manual

- VPE_PRM Register Summary: [5]

Table 3-1651. PM_VPE_VPE_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7CA0	**Instance**	VPE_PRM
Description	This register controls wakeup dependency based on VPE service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	RESERVED	WKUPDEP_VPE_EVE2	WKUPDEP_VPE_EVE1	WKUPDEP_VPE_DSP2	WKUPDEP_VPE_IPU1	RESERVED	WKUPDEP_VPE_DSP1	WKUPDEP_VPE_IPU2	WKUPDEP_VPE_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_VPE_EVE2	Wakeup dependency from VPE module (Swakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_VPE_EVE1	Wakeup dependency from VPE module (Swakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_VPE_DSP2	Wakeup dependency from VPE module (Swakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_VPE_IPU1	Wakeup dependency from VPE module (Swakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_VPE_DSP1	Wakeup dependency from VPE module (Swakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_VPE_IPU2	Wakeup dependency from VPE module (Swakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_VPE_MPU	Wakeup dependency from VPE module (Swakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1652. Register Call Summary for Register PM_VPE_VPE_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

Table 3-1652. Register Call Summary for Register PM_VPE_VPE_WKDEP (continued)

PRCM Register Manual

- VPE_PRM Register Summary: [9]

Table 3-1653. RM_VPE_VPE_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7CA4	**Instance**	VPE_PRM
Description	This register contains dedicated VPE context statuses. [warm reset insensitive]		
Type	RW		

31–9	8	7–1	0
RESERVED	LOSTMEM_VPE_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VPE_BANK	Specify if memory-based context in VPE memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of VPE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1654. Register Call Summary for Register RM_VPE_VPE_CONTEXT

Power Management Functional Description

- PD_VPE Description: [0]

PRCM Register Manual

- VPE_PRM Register Summary: [1]

3.13.48 WKUPAON_CM Registers

3.13.48.1 WKUPAON_CM Register Summary

Table 3-1655. WKUPAON_CM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WKUPAON_CM Physical Address L4_WKUP Interconnect
CM_WKUPAON_CLKSTCTRL	RW	32	0x0000 0000	0x4AE0 7800
CM_WKUPAON_L4_WKUP_CLKCTRL	R	32	0x0000 0020	0x4AE0 7820
RESERVED	R	32	0x0000 0028	0x4AE0 7828
CM_WKUPAON_WD_TIMER2_CLKCTRL	RW	32	0x0000 0030	0x4AE0 7830
CM_WKUPAON_GPIO1_CLKCTRL	RW	32	0x0000 0038	0x4AE0 7838
CM_WKUPAON_TIMER1_CLKCTRL	RW	32	0x0000 0040	0x4AE0 7840

Table 3-1655. WKUPAON_CM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	WKUPAON_CM Physical Address L4_WKUP Interconnect
CM_WKUPAON_TIMER12_CLKCTRL	R	32	0x0000 0048	0x4AE0 7848
CM_WKUPAON_COUNTER_32K_CLKCTRL	R	32	0x0000 0050	0x4AE0 7850
RESERVED	R	32	0x0000 0060	0x4AE0 7860
CM_WKUPAON_KBD_CLKCTRL	RW	32	0x0000 0078	0x4AE0 7878
CM_WKUPAON_UART10_CLKCTRL	RW	32	0x0000 0080	0x4AE0 7880
CM_WKUPAON_DCAN1_CLKCTRL	RW	32	0x0000 0088	0x4AE0 7888
RESERVED	RW	32	0x0000 0090	0x4AE0 7890
RESERVED	RW	32	0x0000 0098	0x4AE0 7898
RESERVED	RW	32	0x0000 00A0	0x4AE0 78A0
RESERVED	R	32	0x0000 00B0	0x4AE0 78B0
RESERVED	R	32	0x0000 00B8	0x4AE0 78B8
RESERVED	R	32	0x0000 00C0	0x4AE0 78C0
RESERVED	R	32	0x0000 00C8	0x4AE0 78C8
RESERVED	R	32	0x0000 00D0	0x4AE0 78D0
RESERVED	R	32	0x0000 00D8	0x4AE0 78D8

3.13.48.2 WKUPAON_CM Register Description

Table 3-1656. CM_WKUPAON_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7800	**Instance**	WKUPAON_CM
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31–20	19	18	17	16	15	14	13	12	11	10	9	8	7–2	1–0
RESERVED	RESERVED	CLKACTIVITY_UART10_GFCLK	CLKACTIVITY_TIMER1_GFCLK	CLKACTIVITY_DCAN1_SYS_CLK	CLKACTIVITY_SYS_CLK_ALL	CLKACTIVITY_SYS_CLK_FUNC	RESERVED	CLKACTIVITY_WKUPAON_GICLK	CLKACTIVITY_WKUPAON_SYS_GFCLK	RESERVED	CLKACTIVITY_ABE_LP_CLK	CLKACTIVITY_SYS_CLK	RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	CLKACTIVITY_UART10_GFCLK	This field indicates the state of the UART10_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
17	CLKACTIVITY_TIMER1_GFCLK	This field indicates the state of the TIMER1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_DCAN1_SYS_CLK	This field indicates the state of the DCAN1_SYS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_SYS_CLK_ALL	This field indicates the state of the SYS_CLK runing at SCRM level because of any SCRM clock request. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_SYS_CLK_FUNC	This field indicates the state of the functional SYS_CLK clocks in the domain (this exclude activity of EMU_GCLK clock). [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	RESERVED		R	0x0
12	CLKACTIVITY_WKUPAON_GICLK	This field indicates the state of the WKUPAON_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_WKUPAON_SYS_GFCLK	This field indicates the state of the WKUPAON_SYS_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	RESERVED		R	0x0
9	CLKACTIVITY_ABE_LP_CLK	This field indicates the state of the ABE_LP_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_SYS_CLK	This field indicates the state of the SYS_CLK clock in the domain(it includes profiling, EMU_SYS_GCLK and all functional SYS_CLK. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the WKUPAON clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-1657. Register Call Summary for Register CM_WKUPAON_CLKSTCTRL

Clock Management Functional Description

- Clock Domain Modes: [0] [1] [2] [3] [4] [7] [8] [9] [11]

PRCM Register Manual

- WKUPAON_CM Register Summary: [12]

Table 3-1658. CM_WKUPAON_L4_WKUP_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7820	**Instance**	WKUPAON_CM
Description	This register manages the WKUPAON clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1659. Register Call Summary for Register CM_WKUPAON_L4_WKUP_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- WKUPAON_CM Register Summary: [2]

Table 3-1660. CM_WKUPAON_WD_TIMER2_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7830	**Instance**	WKUPAON_CM
Description	This register manages the WD_TIMER2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1661. Register Call Summary for Register CM_WKUPAON_WD_TIMER2_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- WKUPAON_CM Register Summary: [2]

Table 3-1662. CM_WKUPAON_GPIO1_CLKCTRL

Address Offset	0x0000 0038		
Physical Address	0x4AE0 7838	**Instance**	WKUPAON_CM
Description	This register manages the GPIO1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEST	RESERVED	OPTFCLKEN_DBCLK	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-1663. Register Call Summary for Register CM_WKUPAON_GPIO1_CLKCTRL

Clock Management Functional Description

- Clock Domain Modes: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- WKUPAON_CM Register Summary: [3]

Table 3-1664. CM_WKUPAON_TIMER1_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7840	**Instance**	WKUPAON_CM
Description	This register manages the TIMER1 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2	1 0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1_DIV_CLK 0x9: Selects VIDEO2_DIV_CLK 0xA: Selects HDMI_DIV_CLK 0xB-0xF: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1665. Register Call Summary for Register CM_WKUPAON_TIMER1_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual
- WKUPAON_CM Register Summary: [3]

Table 3-1666. CM_WKUPAON_TIMER12_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7848	**Instance**	WKUPAON_CM
Description	This register manages the TIMER12 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1667. Register Call Summary for Register CM_WKUPAON_TIMER12_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- WKUPAON_CM Register Summary: [2]

Table 3-1668. CM_WKUPAON_COUNTER_32K_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4AE0 7850	**Instance**	WKUPAON_CM
Description	This register manages the COUNTER_32K clocks.		
Type	R		

31 30 29 28 27 26 25 24 23 22 21 20 19 18	17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2	1 0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1669. Register Call Summary for Register CM_WKUPAON_COUNTER_32K_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

PRCM Register Manual

- WKUPAON_CM Register Summary: [2]

Table 3-1670. CM_WKUPAON_KBD_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4AE0 7878	**Instance**	WKUPAON_CM
Description	This register manages the KBD clocks.		
Type	RW		

31:18	17:16	15:2	1:0
RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1671. Register Call Summary for Register CM_WKUPAON_KBD_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0] [1]

Table 3-1671. Register Call Summary for Register CM_WKUPAON_KBD_CLKCTRL (continued)

PRCM Register Manual

- WKUPAON_CM Register Summary: [2]

Table 3-1672. CM_WKUPAON_UART10_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4AE0 7880	**Instance**	WKUPAON_CM
Description	This register manages the UART10 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_FCLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_FCLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1673. Register Call Summary for Register CM_WKUPAON_UART10_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- WKUPAON_CM Register Summary: [3]

Table 3-1674. CM_WKUPAON_DCAN1_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4AE0 7888	**Instance**	WKUPAON_CM

Table 3-1674. CM_WKUPAON_DCAN1_CLKCTRL (continued)

Description	This register manages the DCAN1 clocks.
Type	RW

31:25	24	23:18	17:16	15:2	1:0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects SYS clock for DCAN1 between SYS_CLK1 and SYS_CLK2 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1675. Register Call Summary for Register CM_WKUPAON_DCAN1_CLKCTRL

Clock Management Functional Description

- PRM Clock Source: [0]
- Clock Domain Module Attributes: [1] [2]

PRCM Register Manual

- WKUPAON_CM Register Summary: [3]

3.13.49 WKUPAON_PRM Registers

3.13.49.1 WKUPAON_PRM Register Summary

Table 3-1676. WKUPAON_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WKUPAON_PRM Physical Address L4_WKUP Interconnect
RM_WKUPAON_L4_WKUP_CONTEXT	RW	32	0x0000 0024	0x4AE0 7724
RESERVED	RW	32	0x0000 0028	0x4AE0 7728
RESERVED	RW	32	0x0000 002C	0x4AE0 772C
PM_WKUPAON_WD_TIMER2_WKDEP	RW	32	0x0000 0030	0x4AE0 7730

Table 3-1676. WKUPAON_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	WKUPAON_PRM Physical Address L4_WKUP Interconnect
RM_WKUPAON_WD_TIMER2_CONTEXT	RW	32	0x0000 0034	0x4AE0 7734
PM_WKUPAON_GPIO1_WKDEP	RW	32	0x0000 0038	0x4AE0 7738
RM_WKUPAON_GPIO1_CONTEXT	RW	32	0x0000 003C	0x4AE0 773C
PM_WKUPAON_TIMER1_WKDEP	RW	32	0x0000 0040	0x4AE0 7740
RM_WKUPAON_TIMER1_CONTEXT	RW	32	0x0000 0044	0x4AE0 7744
PM_WKUPAON_TIMER12_WKDEP	RW	32	0x0000 0048	0x4AE0 7748
RM_WKUPAON_TIMER12_CONTEXT	RW	32	0x0000 004C	0x4AE0 774C
RM_WKUPAON_COUNTER_32K_CONTEXT	RW	32	0x0000 0054	0x4AE0 7754
RESERVED	RW	32	0x0000 0064	0x4AE0 7764
PM_WKUPAON_KBD_WKDEP	RW	32	0x0000 0078	0x4AE0 7778
RM_WKUPAON_KBD_CONTEXT	RW	32	0x0000 007C	0x4AE0 777C
PM_WKUPAON_UART10_WKDEP	RW	32	0x0000 0080	0x4AE0 7780
RM_WKUPAON_UART10_CONTEXT	RW	32	0x0000 0084	0x4AE0 7784
PM_WKUPAON_DCAN1_WKDEP	RW	32	0x0000 0088	0x4AE0 7788
RM_WKUPAON_DCAN1_CONTEXT	RW	32	0x0000 008C	0x4AE0 778C
RESERVED	RW	32	0x0000 00A0	0x4AE0 77A0
RESERVED	RW	32	0x0000 00A4	0x4AE0 77A4
RESERVED	RW	32	0x0000 00B4	0x4AE0 77B4
RESERVED	RW	32	0x0000 00BC	0x4AE0 77BC
RESERVED	RW	32	0x0000 00C4	0x4AE0 77C4
RESERVED	RW	32	0x0000 00CC	0x4AE0 77CC
RESERVED	RW	32	0x0000 00D4	0x4AE0 77D4
RESERVED	RW	32	0x0000 00DC	0x4AE0 77DC

3.13.49.2 WKUPAON_PRM Register Description

Table 3-1677. RM_WKUPAON_L4_WKUP_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7724	**Instance**	WKUPAON_PRM
Description	This register contains dedicated L4_WKUP context statuses. [warm reset insensitive]		
Type	RW		

31–1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1678. Register Call Summary for Register RM_WKUPAON_L4_WKUP_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1679. PM_WKUPAON_WD_TIMER2_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7730	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on WD_TIMER2 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_WD_TIMER2_EVE2	WKUPDEP_WD_TIMER2_EVE1	WKUPDEP_WD_TIMER2_DSP2	WKUPDEP_WD_TIMER2_IPU1	RESERVED	WKUPDEP_WD_TIMER2_DSP1	WKUPDEP_WD_TIMER2_IPU2	WKUPDEP_WD_TIMER2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_WD_TIMER2_EVE2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_WD_TIMER2_EVE1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_WD_TIMER2_DSP2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_WD_TIMER2_IPU1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_WD_TIMER2_DSP1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_WD_TIMER2_IPU2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_WD_TIMER2_MPU	Wakeup dependency from WDT2 module (SWakeup signal) towards MPU + L3MAIN1 + L4CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1680. Register Call Summary for Register PM_WKUPAON_WD_TIMER2_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [9]

Table 3-1681. RM_WKUPAON_WD_TIMER2_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7734	**Instance**	WKUPAON_PRM
Description	This register contains dedicated WD_TIMER2 context statuses. [warm reset insensitive]		
Type	RW		

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1682. Register Call Summary for Register RM_WKUPAON_WD_TIMER2_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1683. PM_WKUPAON_GPIO1_WKDEP

Address Offset	0x0000 0038		
Physical Address	0x4AE0 7738	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on GPIO1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED	RESERVED	WKUPDEP_GPIO1_IRQ2_EVE2	WKUPDEP_GPIO1_IRQ2_EVE1	WKUPDEP_GPIO1_IRQ2_DSP2	WKUPDEP_GPIO1_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO1_IRQ2_DSP1	WKUPDEP_GPIO1_IRQ2_IPU2	WKUPDEP_GPIO1_IRQ2_MPU	RESERVED	RESERVED	WKUPDEP_GPIO1_IRQ1_EVE2	WKUPDEP_GPIO1_IRQ1_EVE1	WKUPDEP_GPIO1_IRQ1_DSP2	WKUPDEP_GPIO1_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO1_IRQ1_DSP1	WKUPDEP_GPIO1_IRQ1_IPU2	WKUPDEP_GPIO1_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	WKUPDEP_GPIO1_IRQ2_EVE2	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO1_IRQ2_EVE1	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO1_IRQ2_DSP2	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO1_IRQ2_IPU1	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO1_IRQ2_DSP1	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO1_IRQ2_IPU2	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
10	WKUPDEP_GPIO1_IRQ2_MPU	Wakeup dependency from GPIO1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_GPIO1_IRQ1_EVE2	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO1_IRQ1_EVE1	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO1_IRQ1_DSP2	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO1_IRQ1_IPU1	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO1_IRQ1_DSP1	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO1_IRQ1_IPU2	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO1_IRQ1_MPU	Wakeup dependency from GPIO1 module (SWakeup signal for POROCPSINTERRUPT1) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1684. Register Call Summary for Register PM_WKUPAON_GPIO1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [18]

Table 3-1685. RM_WKUPAON_GPIO1_CONTEXT

Address Offset	0x0000 003C

Table 3-1685. RM_WKUPAON_GPIO1_CONTEXT (continued)

Physical Address	0x4AE0 773C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated GPIO1 context statuses. [warm reset insensitive]		
Type	RW		

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1686. Register Call Summary for Register RM_WKUPAON_GPIO1_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1687. PM_WKUPAON_TIMER1_WKDEP

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7740	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on TIMER1 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER1_EVE2	WKUPDEP_TIMER1_EVE1	WKUPDEP_TIMER1_DSP2	WKUPDEP_TIMER1_IPU1	RESERVED	WKUPDEP_TIMER1_DSP1	WKUPDEP_TIMER1_IPU2	WKUPDEP_TIMER1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_TIMER1_EVE2	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER1_EVE1	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER1_DSP2	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER1_IPU1	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER1_DSP1	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER1_IPU2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER1_MPU	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1688. Register Call Summary for Register PM_WKUPAON_TIMER1_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [9]

Table 3-1689. RM_WKUPAON_TIMER1_CONTEXT

Address Offset	0x0000 0044		
Physical Address	0x4AE0 7744	**Instance**	WKUPAON_PRM
Description	This register contains dedicated TIMER1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:1	0
RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1690. Register Call Summary for Register RM_WKUPAON_TIMER1_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1691. PM_WKUPAON_TIMER12_WKDEP

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7748	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on TIMER12 service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_TIMER12_EVE2	WKUPDEP_TIMER12_EVE1	WKUPDEP_TIMER12_DSP2	WKUPDEP_TIMER12_IPU1	RESERVED	WKUPDEP_TIMER12_DSP1	WKUPDEP_TIMER12_IPU2	WKUPDEP_TIMER12_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_TIMER12_EVE2	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_TIMER12_EVE1	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER12_DSP2	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER12_IPU1	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER12_DSP1	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER12_IPU2	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER12_MPU	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1692. Register Call Summary for Register PM_WKUPAON_TIMER12_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [7]

Table 3-1693. RM_WKUPAON_TIMER12_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 774C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated TIMER12 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1694. Register Call Summary for Register RM_WKUPAON_TIMER12_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1695. RM_WKUPAON_COUNTER_32K_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 7754	**Instance**	WKUPAON_PRM
Description	This register contains dedicated COUNTER_32K context statuses. This bit-field is only sensitive to the external power-on reset (SYS_PWRON_RST reset line)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_SYS_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1696. Register Call Summary for Register RM_WKUPAON_COUNTER_32K_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

Table 3-1696. Register Call Summary for Register RM_WKUPAON_COUNTER_32K_CONTEXT (continued)

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1697. PM_WKUPAON_KBD_WKDEP

Address Offset	0x0000 0078		
Physical Address	0x4AE0 7778	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on KBD service requests.		
Type	RW		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_KBD_EVE2	WKUPDEP_KBD_EVE1	WKUPDEP_KBD_DSP2	WKUPDEP_KBD_IPU1	RESERVED	WKUPDEP_KBD_DSP1	WKUPDEP_KBD_IPU2	WKUPDEP_KBD_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_KBD_EVE2	Wakeup dependency from KBD module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_KBD_EVE1	Wakeup dependency from KBD module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_KBD_DSP2	Wakeup dependency from KBD module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_KBD_IPU1	Wakeup dependency from KBD module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_KBD_DSP1	Wakeup dependency from KBD module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_KBD_IPU2	Wakeup dependency from KBD module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_KBD_MPU	Wakeup dependency from KBD module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1698. Register Call Summary for Register PM_WKUPAON_KBD_WKDEP

Clock Management Functional Description

- Clock Domain Dependency: [0] [1] [2] [3] [4] [5] [6]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [9]

Table 3-1699. RM_WKUPAON_KBD_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 777C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated KBD context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1700. Register Call Summary for Register RM_WKUPAON_KBD_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1701. PM_WKUPAON_UART10_WKDEP

Address Offset	0x0000 0080		
Physical Address	0x4AE0 7780	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on UART10 service requests.		

Table 3-1701. PM_WKUPAON_UART10_WKDEP (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RESERVED	RESERVED	WKUPDEP_UART10_EVE2	WKUPDEP_UART10_EVE1	WKUPDEP_UART10_DSP2	WKUPDEP_UART10_IPU1	WKUPDEP_UART10_SDMA	WKUPDEP_UART10_DSP1	WKUPDEP_UART10_IPU2	WKUPDEP_UART10_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_UART10_EVE2	Wakeup dependency from UART10 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART10_EVE1	Wakeup dependency from UART10 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART10_DSP2	Wakeup dependency from UART10 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART10_IPU1	Wakeup dependency from UART10 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART10_SDMA	Wakeup dependency from UART10 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART10_DSP1	Wakeup dependency from UART10 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART10_IPU2	Wakeup dependency from UART10 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_UART10_MPU	Wakeup dependency from UART10 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1702. Register Call Summary for Register PM_WKUPAON_UART10_WKDEP

PRCM Register Manual

- WKUPAON_PRM Register Summary: [0]

Table 3-1703. RM_WKUPAON_UART10_CONTEXT

Address Offset	0x0000 0084		
Physical Address	0x4AE0 7784	**Instance**	WKUPAON_PRM
Description	This register contains dedicated UART10 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_RETAINED_BANK	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in UART memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1704. Register Call Summary for Register RM_WKUPAON_UART10_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Table 3-1705. PM_WKUPAON_DCAN1_WKDEP

Address Offset	0x0000 0088		
Physical Address	0x4AE0 7788	**Instance**	WKUPAON_PRM

Table 3-1705. PM_WKUPAON_DCAN1_WKDEP (continued)

Description	This register controls wakeup dependency based on DCAN1 service requests.
Type	RW

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	WKUPDEP_DCAN1_EVE2	WKUPDEP_DCAN1_EVE1	WKUPDEP_DCAN1_DSP2	WKUPDEP_DCAN1_IPU1	WKUPDEP_DCAN1_SDMA	WKUPDEP_DCAN1_DSP1	WKUPDEP_DCAN1_IPU2	WKUPDEP_DCAN1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	RESERVED		R	0x0
7	WKUPDEP_DCAN1_EVE2	Wakeup dependency from DCAN1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCAN1_EVE1	Wakeup dependency from DCAN1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCAN1_DSP2	Wakeup dependency from DCAN1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCAN1_IPU1	Wakeup dependency from DCAN1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_DCAN1_SDMA	Wakeup dependency from DCAN1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_DCAN1_DSP1	Wakeup dependency from DCAN1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCAN1_IPU2	Wakeup dependency from DCAN1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_DCAN1_MPU	Wakeup dependency from DCAN1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1706. Register Call Summary for Register PM_WKUPAON_DCAN1_WKDEP

PRCM Register Manual

- WKUPAON_PRM Register Summary: [0]

Table 3-1707. RM_WKUPAON_DCAN1_CONTEXT

Address Offset	0x0000 008C		
Physical Address	0x4AE0 778C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated DCAN1 context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_DCAN_MEM	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_DCAN_MEM	Specify if memory-based context in DCAN memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1708. Register Call Summary for Register RM_WKUPAON_DCAN1_CONTEXT

Power Management Functional Description

- PD_WKUPAON Description: [0]

PRCM Register Manual

- WKUPAON_PRM Register Summary: [1]

Chapter 4

Dual Cortex-A15 MPU Subsystem

This chapter describes the dual Cortex®-A15 MPU subsystem.

4.1 Dual Cortex-A15 MPU Subsystem Overview 1270
4.2 Dual Cortex-A15 MPU Subsystem Integration 1274
4.3 Dual Cortex-A15 MPU Subsystem Functional Description 1278
4.4 Dual Cortex-A15 MPU Subsystem Register Manual 1301

4.1 Dual Cortex-A15 MPU Subsystem Overview

4.1.1 Introduction

The dual Cortex®-A15 microprocessor unit (MPU) subsystem serves the applications processing role by running the high-level operating system (HLOS) and application code. The MPU subsystem is based on the symmetric multiprocessor (SMP) architecture, and thus it delivers high performance and optimal power management, debug, and emulation capabilities.

The MPU subsystem incorporates two Cortex-A15 MPU cores (MPU_C0 and MPU_C1), individual level 1 (L1) caches, level 2 (L2) cache (MPU_L2CACHE) shared between them, and various other shared peripherals. To aid software development, the processor cores can be kept cache-coherent with each other and with the L2 cache.

The MPU subsystem provides a high-performance computing platform with high peak-computing performance and low memory latency, while also supporting a configuration to shut off one core and run the other at low voltage and low frequency to achieve low-power operation.

Note

Only MPU_C1 can be put in off (FORCED_OFF) power state. The lowest power state of MPU_C0 is CSWR.

Figure 4-1 is a high-level block diagram of the MPU subsystem.

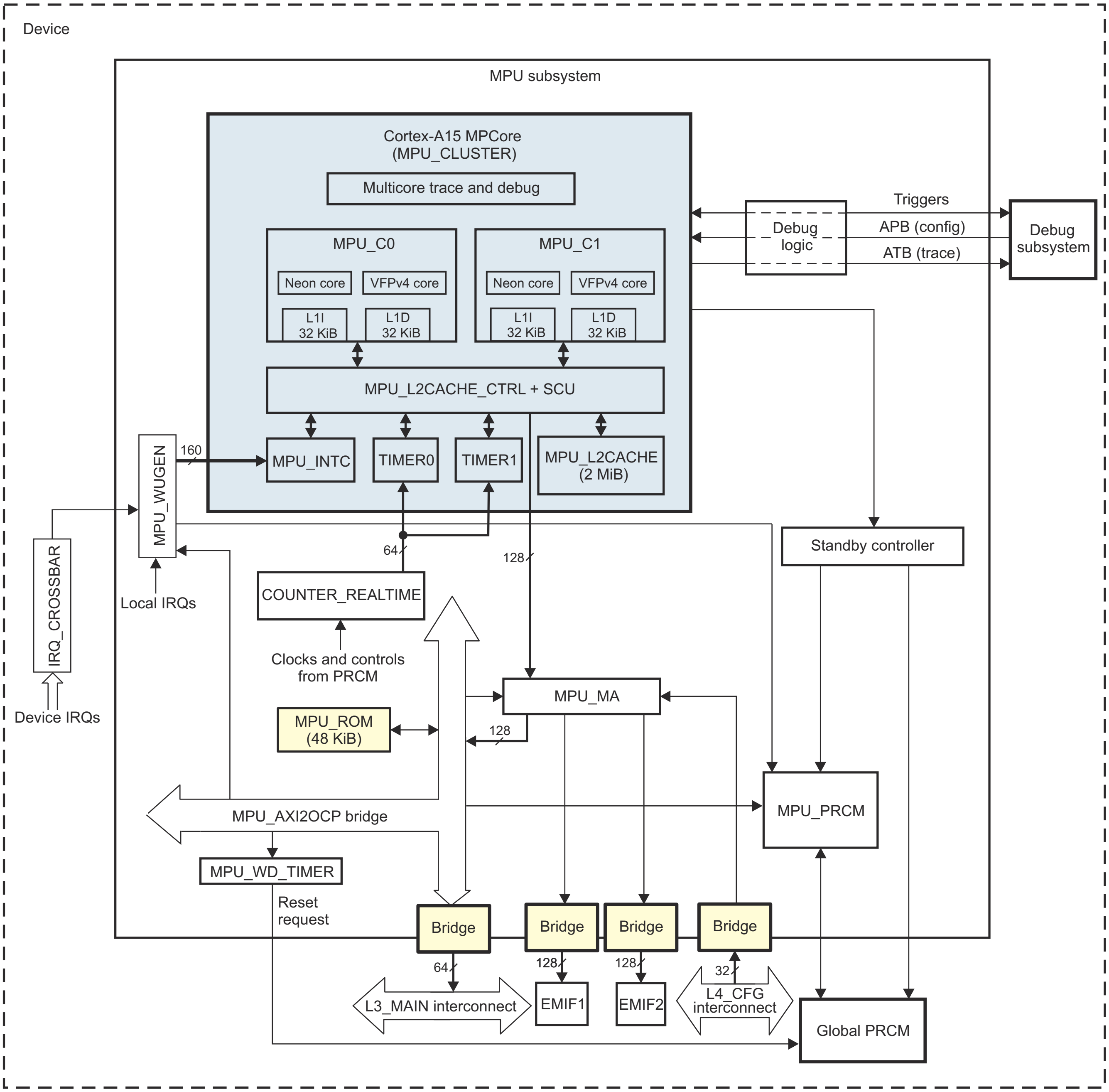


Figure 4-1. MPU Subsystem Overview

4.1.2 Features

The MPU subsystem integrates the following:

- Arm®Cortex-A15 MPCore (MPU_CLUSTER)
  - Two Cortex-A15 MPU cores (revision r2p2, SMP architecture), each of them having the following features:
    - Superscalar, dynamic multi-issue technology
      - Out-of-order (OoO) instruction dispatch and completion
      - Dynamic branch prediction with branch target buffer (BTB), global history buffer (GHB), and 48-entry return stack
      - Continuous fetch and decoding of three instructions per clock cycle
      - Dispatch of up to four instructions and completion of eight instructions per clock cycle
      - Provides optimal performance from binaries compiled for previous Arm processors
      - Five execution units handle simple instructions, branch instructions, Neon™ and floating point instructions, multiply instructions, and load and store instructions.
      - Simple instructions take two cycles from dispatch, while complex instructions take up to 11 cycles.
      - Can issue two simple instructions in a cycle
      - Can issue a load and a store instruction in the same cycle
    - Integrated Neon processing engine to include the Arm Neon Advanced SIMD (single instruction, multiple data) support for accelerated media and signal processing computation
    - Includes VFPv4-compatible hardware to support single- and double-precision add, subtract, divide, multiply and accumulate, and square root operations
    - Extensive support to accelerate virtualization using a hypervisor
    - 32-KiB L1 instruction (L1I) and 32-KiB L1 data (L1D) cache:
      - 64-byte line size
      - 2-way set associative
    - Memory management unit (MMU):
      - Two-level translation lookaside buffer (TLB) organization
        - First level is an 32-entry, fully associative micro-TLB implemented for each of instruction fetch, load, and store.
        - Second level is a unified, 4-way associative, 512-entry main TLB
      - Supports hardware TLB table-walk for backward-compatible and new 64-bit entry page table formats
      - New page table format can produce 40-bit physical addresses
      - Two-stage translation where first stage is HLOS-controlled and the second level may be controlled by a hypervisor. Second stage always uses the new page table format
  - Integrated L2 cache (MPU_L2CACHE) and snoop control unit (SCU):
    - 2-MiB of unified (instructions and data) cache organized as 16 ways of 2048 sets of 64-byte lines
    - Redundant L1 data (cache) tags to perform snoop filtering (L1 instruction cache tags are not duplicated)
    - Operates at Cortex-A15 MPU core clock rate
    - Integrated L2 cache controller (MPU_L2CACHE_CTRL):
      - Sixteen 64-byte line buffers that handle evictions, line fills and snoop transfers
      - One 128-bit AMBA4 Coherent Bus (AXI4-ACE) port
      - Auto-prefetch buffer for up to 16 streams per core and detecting forward and backward strides
      - L1 cache coherency between the MPU cores is maintained by the L2 cache controller, which also serves as a snoop controller.
    - SCU ensures memory coherency between the two MPU cores
  - Generalized interrupt controller (GIC, also referred to as MPU_INTC): An interrupt controller supplied by Arm. The single GIC in the MPU_CLUSTER routes interrupts to each of the MPU cores. The GIC supports:
    - Number of shared peripheral interrupts (SPI): 160
    - Number of software generated interrupts (SGI): 16
    - Number of CPU interfaces: 2

    - Virtual CPU interface for virtualization support. This allows the majority of guest operating system (OS) interactions with the GIC to be handled in hardware, but with physical interrupts still requiring hypervisor intervention to assign them to the appropriate virtual machine.
  - Integrated timer counter and one timer block per MPU core
  - Arm CoreSight ™ debug and trace modules. For more information, see Chapter 33, *On-Chip Debug Support*.
- MPU_AXI2OCP bridge (local interconnect):
  - Connected to Memory Adapter (MPU_MA), which routes the non-EMIF address space transactions to MPU_AXI2OCP
  - Single request multiple data (SRMD) protocol on L3_MAIN port
  - Multiple targets:
    - 64-bit port to the L3_MAIN interconnect. Interface frequency is 1/4 or 1/8 of core frequency
    - MPU_ROM
    - Internal MPU subsystem peripheral targets, including Memory Adapter LISA Section Manager (MA_LSM), wake-up generator (MPU_WUGEN), watchdog timer (MPU_WD_TIMER), and local PRCM module (MPU_PRCM) configuration
    - Internal AXI target, CoreSight System Trace Module (CS_STM)
- Memory adapter (MPU_MA): Helps decrease the latency of accesses between the MPU_L2CACHE and the two EMIFs (EMIF0 and EMIF1) by providing a direct path between the MPU subsystem and the EMIFs:
  - Connected to 128-bit AMBA4 interface of MPU_CLUSTER
  - Direct 128-bit interface to each of EMIF0 and EMIF1
  - Interface speed between MPU_CLUSTER and MPU_MA is at half-speed of MPU_CLUSTER internal core frequency
  - Quarter-speed interface to EMIF
  - Performs interleaving functions to optimize EMIF interface bandwidth
  - Uses firewall logic to check access rights of incoming addresses
- Local PRCM (MPU_PRCM):
  - Handles MPU_C0 and MPU_C1 power domains
  - Supports SR3-APG (SmartReflex3 Automatic Power Gating) power management technology inside the MPU_CLUSTER
  - MPU subsystem has six power domains
- Wake-up generator (MPU_WUGEN)
  - Responsible for waking up the MPU cores
  - Used by the ROM code and OS during SMP boot
- Standby controller: Handles the power transitions inside the MPU subsystem
- Realtime (master) counter (COUNTER_REALTIME): Produces the count used by the private timer peripherals in the MPU_CLUSTER
- Watchdog timer (MPU_WD_TIMER): Used to generate a chip-level watchdog reset request to global PRCM
- On-chip ROM (MPU_ROM): Both MPU_C0 and MPU_C1 can boot from this memory. The MPU_ROM size is 48-KiB, and the address range is from 0x4003 8000 to 0x4004 3FFF. For more information, see Chapter 32, *Initialization*.
- Interfaces:
  - 128-bit interface to each of EMIF0 and EMIF1
  - 64-bit master port to the L3_MAIN interconnect
  - 32-bit slave port from the L4_CFG_EMU interconnect (debug subsystem) for configuration of the MPU subsystem debug modules
  - 32-bit slave port from the L4_CFG interconnect for memory adapter firewall (MPU_MA_NTTP_FW) configuration
  - 32-bit ATB output for transmitting debug and trace data
  - 160 peripheral interrupt inputs

4.2 Dual Cortex-A15 MPU Subsystem Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 4-2 shows the MPU subsystem integration.

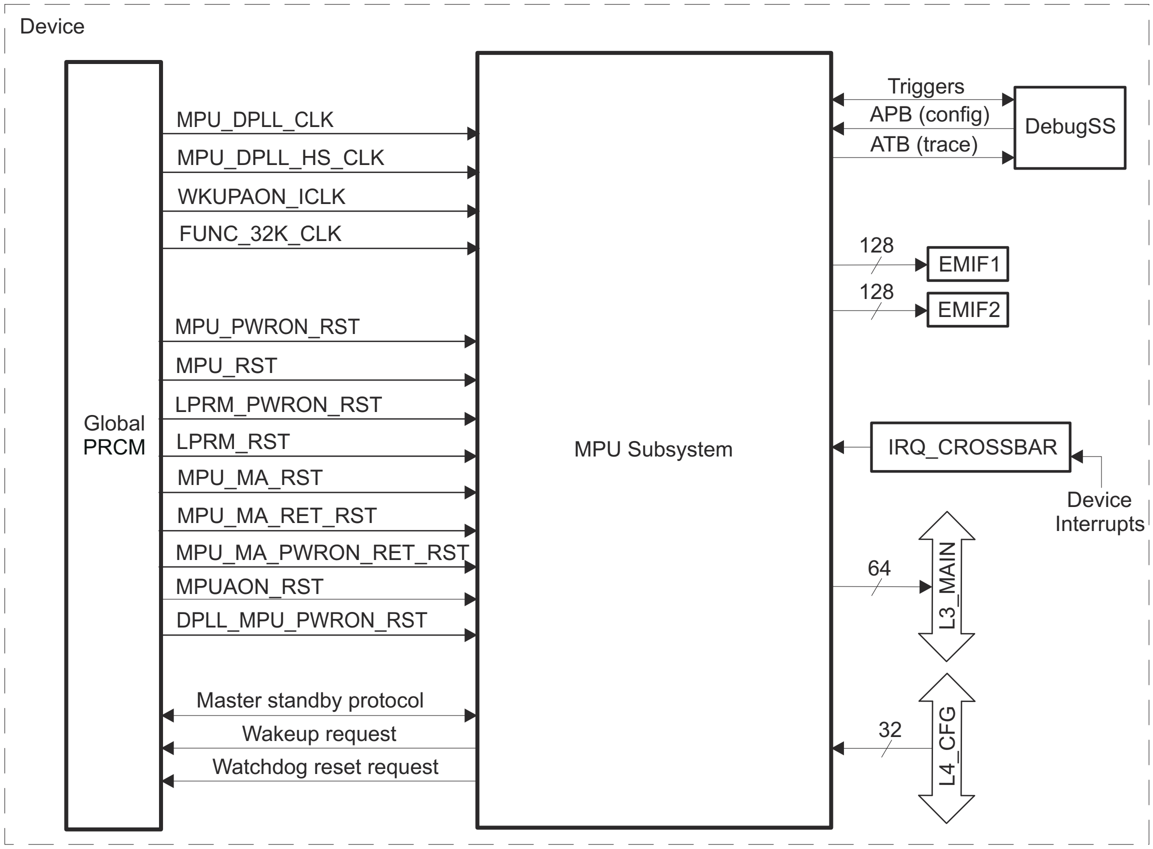


Figure 4-2. MPU Subsystem Integration

Note

For more information about:

- Clock distribution inside MPU subsystem, see *Clock Distribution*.
- Reset distribution inside MPU subsystem, see *Reset Distribution*.
- MPU_INTC default interrupt mapping, and IRQ_CROSSBAR mapping, see *Interrupt Controllers*.
- MPU watchdog timer reset request, see Section 4.3.6, *MPU Watchdog Timer*.
- MPU trace and cross-triggering with Debug Subsystem, see Chapter 33, *On-Chip Debug Support*.
- Master standby protocol and wakeup request, see *Module-Level Clock Management* in *Power, Reset, and Clock Management*.

4.2.1 Clock Distribution

The Cortex-A15 MPU clock generator is fed by the MPU digital phase-locked loop (DPLL), which can be gated off by the global power, reset, and clock management (PRCM) module when system power domain is in a low-power state. There is a global clock gating for each MPU core. Because of the DPLL_MPU, the MPU subsystem is asynchronous from the rest of the device.

Figure 4-3 shows the MPU subsystem clocking scheme.

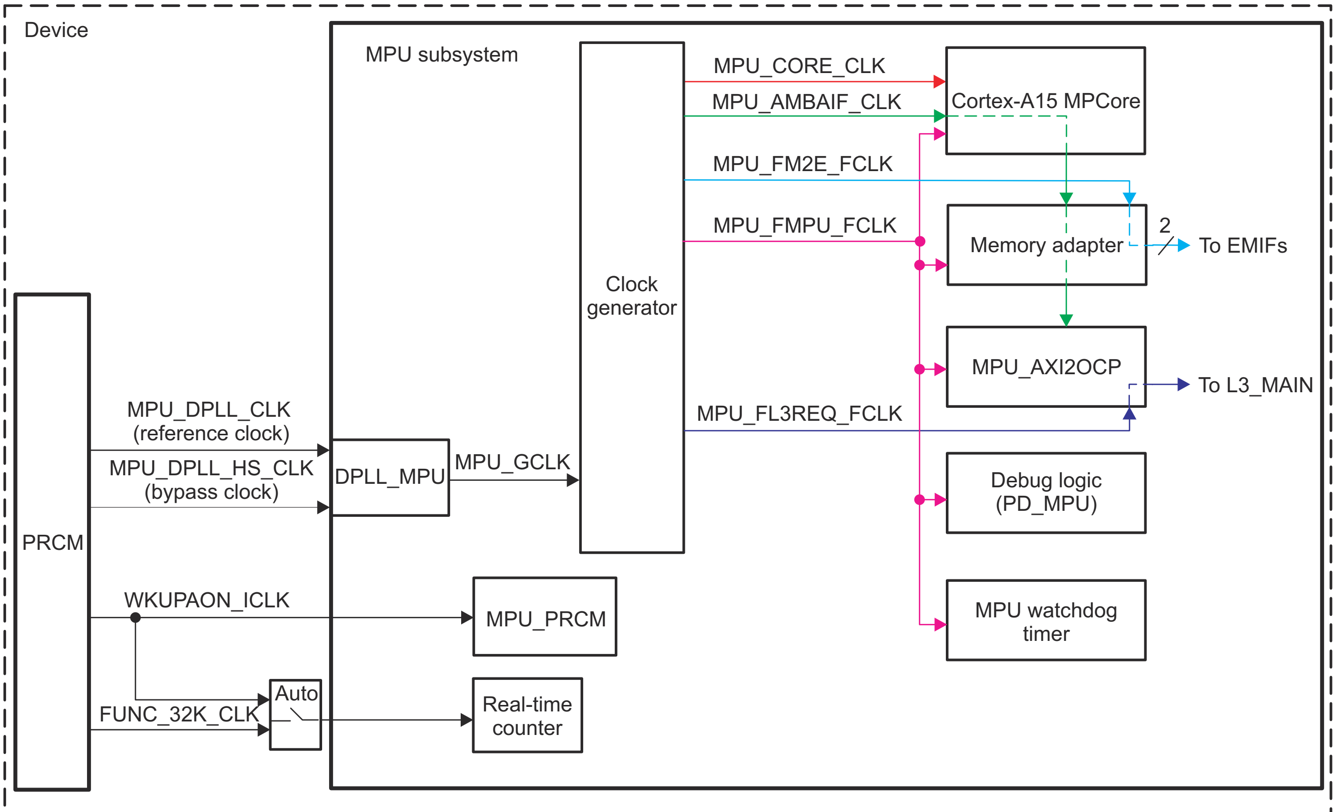


Figure 4-3. MPU Subsystem Clocking Scheme

The clock generator generates the following clocks from the DPLL_MPU output clock (MPU_GCLK):

- MPU_CORE_CLK
- MPU_AMBAIF_CLK
- MPU_FMPU_FCLK
- MPU_FL3REQ_FCLK
- MPU_FM2E_FCLK

MPU_CORE_CLK is directly derived (no hardware dividing) from the DPLL_MPU output clock (that is, MPU_CORE_CLK = MPU_GCLK). The other clocks are derived via dividers.

Table 4-1 shows the supported frequency values for MPU subsystem clocks at different OPPs.

Table 4-1. MPU Subsystem Clocks Frequency Value Versus OPP

Clocks (Derived From DPLL_MPU)	OPP_LOW	OPP_NOM	OPP_OD	OPP_HIGH	OPP_PLUS
MPU_CORE_CLK (f)	See (1)	See (1)	See (1)	See (1)	See (1)
MPU_AMBAIF_CLK	f/2	f/2	f/2	f/2	f/2
MPU_FMPU_FCLK	f/4	f/4	f/4	f/4	f/4
MPU_FL3REQ_FCLK	f/4	f/4	f/4	f/8	f/8
MPU_FM2E_FCLK	f/4	f/4	f/4	f/4	f/4

(1) See device *Data Manual* for information about frequency value

Some of the OPPs listed in Table 4-1 may not be supported for some devices.

Note

For more information about the supported OPPs, see the "*Operating Performance Points*" section of the device Data Manual.

Note

For more information about the DPLL_MPU, see *DPLL_MPU Description*, in *Power, Reset, and Clock Management*.

The two MPU cores cannot be clocked at different frequencies.

The realtime counter (COUNTER_REALTIME) is clocked by either WKUPAON_ICLK or FUNC_32K_CLK clocks, provided by the global PRCM module. By default, WKUPAON_ICLK is used as a COUNTER_REALTIME clock. When the MPU subsystem goes to standby mode, the counter automatically switches to a low-power mode using the FUNC_32K_CLK (SYS_CLK1/610).

4.2.2 Reset Distribution

Resets to the MPU subsystem are provided by the global PRCM module and controlled by the local MPU_PRCM module.

Figure 4-4 shows the reset scheme of the MPU subsystem.

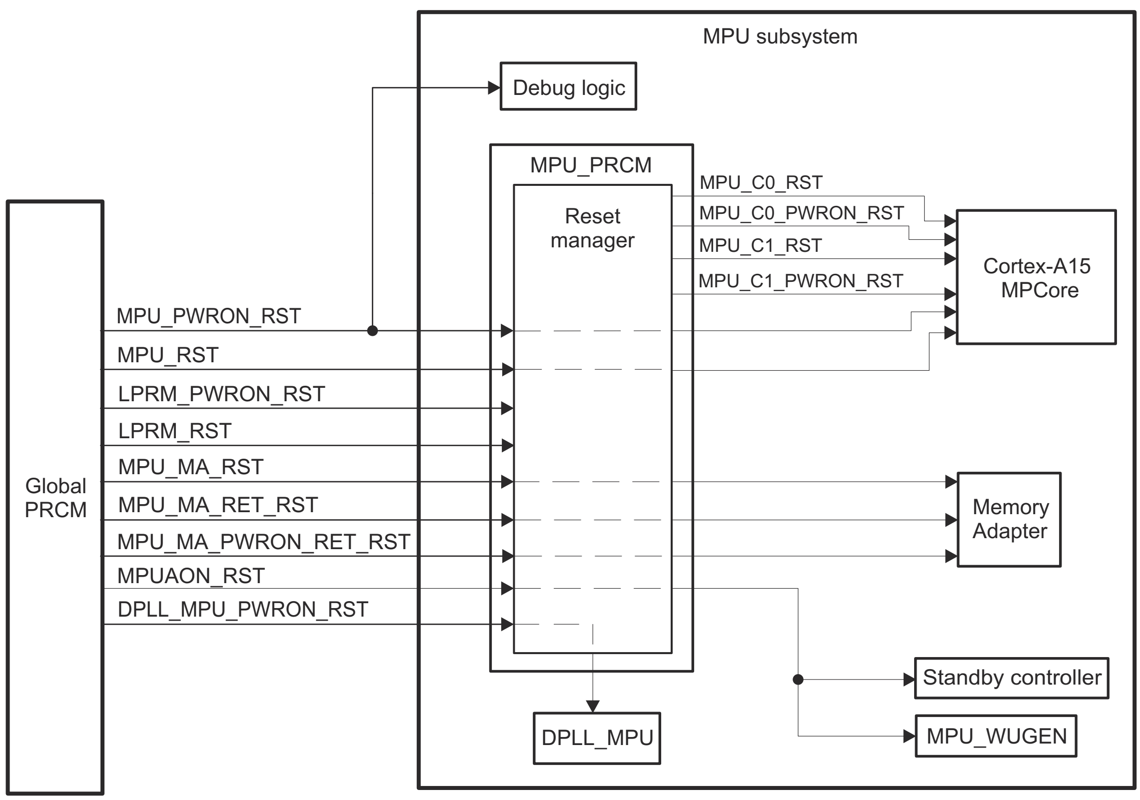


Figure 4-4. MPU Subsystem Reset Scheme

All external resets that are input to the local PRCM signals are active low. All external reset input signals are driven by the global PRCM. Four internal reset signals are generated by the local PRCM module.

The LPRM_PWRON_RST reset signal is a global cold reset for the wake-up logic and resets the wake-up domain logic (the PSCON modules) in the local PRCM module. A cold reset is typically asserted when power is initially applied to the system. The user can check whether this reset event has occurred by reading the PRM_RSTST[0] GLOBAL_COLD_RST bit.

The LPRM_RST reset signal is a global warm reset that resets the wake-up domain logic (the PSCON modules) in the local PRCM module. Warm reset is typically used to reset a system that has been operating for some time. The user can check whether this reset event has occurred by reading the PRM_RSTST[1] GLOBAL_WARM_RST bit.

The DPLL_MPU_PWRON_RST reset signal resets the DPLL_MPU.

The MPUAON_RST reset signal resets the MPU always-on domain: the standby controller and the MPU_WUGEN. The user can check whether the reset has occurred by reading the WKG_CONTROL_0[15] DOMAIN_RST bit for MPU_C0 and the WKG_CONTROL_1[15] DOMAIN_RST bit for MPU_C1.

The MPU_MA has three incoming reset signals:

- MPU_MA_RST
- MPU_MA_RET_RST
- MPU_MA_PWRON_RET_RST

The local PRCM module provides two reset signals for each CPU:

- The MPU_C0_RST and MPU_C1_RST reset signals are warm reset events. These reset signals initialize most of the Arm MPU cores, except the debug logic (breakpoints and watchpoints are retained during this reset). The user can check whether these reset events have occurred by reading the RM_CPUx_RSTCTRL[0] RST bit.
- The MPU_C0_PWRON_RST and MPU_C1_PWRON_RST reset signals are cold and debug reset events.

For more information about clocks, resets, and power domains, and the MPU_PWRON_RST and MPU_RST reset signals, see *Power, Reset, and Clock Management*.

4.3 Dual Cortex-A15 MPU Subsystem Functional Description

4.3.1 MPU Subsystem Block Diagram

Figure 4-5 shows the block diagram of the MPU subsystem.

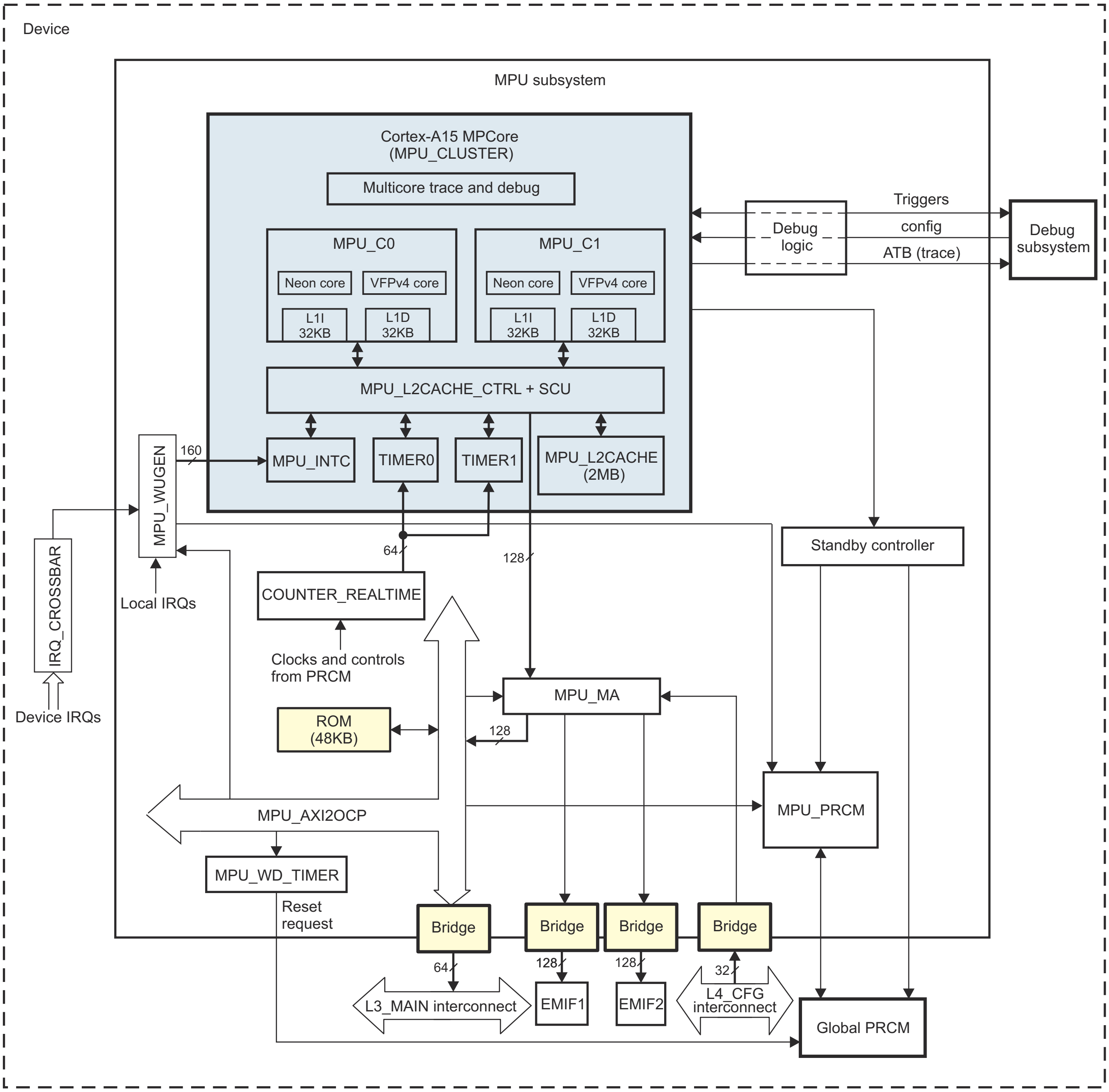


Figure 4-5. MPU Subsystem Block Diagram

4.3.2 Cortex-A15 MPCore (MPU_CLUSTER)

The MPU_CLUSTER consists of:

- Two CPUs (dual-core configuration), including:
  – Arm version 7 ISA: Standard Arm instruction set plus Thumb®-2 , Jazelle® RCT, and Jazelle DBX Java™ accelerator

  - Neon SIMD coprocessor and VFPv4
  - 12-stage in-order MPU core pipeline
  - 128-bit-wide instruction fetch allows fetching up to four instructions/cycle
  - 32 KiB/32 KiB instruction and data cache for each MPU core
  - Complex Execution Unit (FPU) per MPU core
  - 32-entry fully-associative micro-TLB each for instruction and data per MPU core
  - 512-entry 4-way set-associative unified TLB per MPU core
- Unified L2 cache control including tags
- Interrupt controller (MPU_INTC)
  - Supports 160 hardware interrupts. For more information about interrupt mapping, see *Interrupt Controllers*.
- One timer and watchdog timer per MPU core
- Internal APB bridge that connects to the APB port of each MPU core

The major interfaces of the MPU_CLUSTER are:

- Single AXI master supporting 128-bit interface
- System coherency supported through the AXI4 ACE interface
- An ATB port (for processor trace)
- An APB port
- Interrupt request lines

Note

The following hardware restrictions/limitations are applied to the Cortex-A15 MPCore in this device:

- Not implemented features:
  - ACP port to support hardware coherency with external master
  - ECC/parity support for the L1 or L2 caches
- Supported ACE configurations:
  - **AXI3** mode (default). In this configuration, barrier transactions are not issued on the AMBA4 interface.
  - **ACE non-coherent, no L3** mode. In this configuration, barrier transactions are issued on the AMBA4 interface.

For more information about AMBA4 interface configuration, see Section 4.3.8, *MPU Subsystem AMBA Interface Configuration*.

For more information about Cortex-A15 MPCore, see the Arm *Cortex-A15 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

4.3.2.1 MPU L2 Cache Memory System

The MPU subsystem implements an L2 memory system. This memory system consists of an L2 cache (MPU_L2CACHE) and associated L2 cache controller (MPU_L2CACHE_CTRL). The MPU L2 cache controller runs at full-CPU speed and is configured to have one 128-bit master port. The L2 cache controller is configurable via CP15 registers and is tightly coupled to the L1 memory system. The MPU L2 memory system supports Arm Instruction Set Architecture (v7).

MPU L2 supports hardware cache coherency, but is used in a limited way in the device. Because the rest of the system does not support coherency with the MPU L2 cache, software coherence is required.

The L2 cache size on the MPU subsystem is 2 MiB. The cache is configured as 16-way set associative, with 64-B line size. The L2 cache controller performs critical word-first-refilling with a random or pseudo-random cache replacement policy.

The L2 includes logic to support cache event monitoring. The events being monitored are routed to the hardware debug (MPUHWDBGOUT[31:0]) port. The mapping of these events to the MPUHWDBGOUT[31:0] port is described in the *Control Module* chapter.

The L2 can be configured to generate interrupts on error conditions or event counter overflow/increment. The L2 interrupt (MPU_CLUSTER_IRQ_AXIERR) is mapped to interrupt line MPU_IRQ_3. When an interrupt occurs, software may look at corresponding interrupt register to determine the source of the interrupt.

4.3.2.1.1 MPU L2 Cache Architecture

The main features of the MPU_L2CACHE are:

- 2048-KiB 16-way set-associative unified instruction/data cache
- 9-stage L2 pipeline
- L2 hit latency of 12 cycles
- Fixed line length of 64-bytes (16 words)
- L1 inclusive
- Physically indexed and tagged
- 16-way set-associative (maximum)
- Bank partitions to support streaming Neon loads and simultaneous (two) L2 requests
- Exclusive-D L2 cache fill policy
- Global-random replacement strategy
- Four dirty bits per cache line to minimize traffic to L3
- Low-leakage sleep mode (retention until accessed)
- Cache redundancy and repair

4.3.2.1.2 MPU L2 Cache Controller

The main features of the MPU_L2CACHE_CTRL are:

- Single 128-bit AXI4 master port interface
- Tightly coupled L2/SCU for better L2 hit latency
- Nonblocking: Supports hit-under-miss and miss-under-miss for Neon requests
- Performs critical data first refilling
- Supports the following cache modes:
  - Write-through, read allocate
  - Write-back, read allocate, and write allocate
- Supports write-combining to two independently tagged quad-words
- 12 × 128-bit write buffers to buffer subblock writes (per MPU core)
- 12 × 128-bit ACP write buffer (shared across MPU cores)
- 16-entry × 512-bit victim buffers (shared across MPU cores)
- Separate 128-bit interfaces to the I-side and the D-side:
  - Four beat (128-bit) transfers to refill L1 from L2 (or 128-bit AXI)
  - Eight beat (64-bit) transfers to refill L1 from AXI
- 128-bit subblock write and copy-back interface on the D-side
- Outstanding transactions on the AXI4 master port:
  - 16 read
  - 16 write
- Performs hardware table walk using the L2 unified TLB

4.3.2.1.3

Note

The Cortex-A15 processor memory system treats all write-through (WT) accesses as 'write-through, no-allocate'. This means that no cache line from any write-through page allocates in any L1 data or L2 cache. This implies that write-through lines are implemented as non-cacheable in Cortex-A15. Memory requests for write-through cache lines are not looked-up in the L1D or L2, and are sent directly to the AXI master interface. Not caching WT lines was done primarily to avoid back-and-forth L1/L2 snoop invalidations and increasing interconnect traffic when more than one CPU attempts to write to the same WT line. Coherency traffic is greatly reduced in an MP configuration when WT data is treated as 'write-through, no-allocate'.

For non-cacheable and WT memory, all Cortex-A15 memory read requests are 64 bytes. A15 over-reads and then allows forwarding of the data to multiple load instructions. If there is only a single load request for that line, there is no latency hit (since Cortex-A15 does critical word first) although there is potentially an increase in bus power. If there are multiple hits due to consecutive loads of data within the same line, there is a possibility of significant performance gains.

4.3.3 MPU_AXI2OCP

MPU_AXI2OCP provides a protocol bridge between buses and also serves as a small local interconnect to:

- Handle traffic to the L3_MAIN interconnect
- Configure local configuration registers in the MPU subsystem

Main features of MPU_AXI2OCP:

- Connects to the L3_MAIN interconnect through a 64-bit port. The interface frequency is configurable between one fourth (default value) and one eighth of the MPU_DPLL_CLK clock signal frequency. This is programmable in the global PRCM register CM_MPU_MPU_CLKCTRL[25:24] CLKSEL_EMIF_DIV_MODE bit. For CM_MPU_MPU_CLKCTRL register description, see *Power, Reset, and Clock Management*
- Connects to the CS_STM module through a 32-bit AXI interface (for software instrumentation)
- Connection to the following modules for register configuration:
  – MPU_MA
  – MPU_PRCM
  – MPU_WUGEN
  – MPU_WD_TIMER
- Contains internal configuration register related to the MPU_MA function (MA_PRIORITY)
- Supports memory barrier instruction
- Supports single-request-multiple-data (data handshaking) burst mode to pipeline requests
- Supports multiple outstanding requests
- Supports posted and nonposted write transactions, based on the attributes of the transactions coming from the Cortex-A15 processor. Software can override all writes from the MPU_AXI2OCP to the L3_MAIN interconnect to be nonposted, regardless of the attributes of the transactions coming from the Arm Cortex-A15 processor, by setting the Control Module register CTRL_CORE_MPU_FORCEWRNP[0] MPU_FORCEWRNP bit to 0x1. This bit must not be changed until the transfer completes. For CTRL_CORE_MPU_FORCEWRNP register description, see *Control Module*.

4.3.4 Memory Adapter

4.3.4.1 MPU_MA Overview

To decrease the latency of accesses between the MPU L2 cache, and the two EMIF modules, a direct path between the MPU subsystem and the EMIFs is created. This direct path is supported by a memory adapter (MPU_MA) module. The MPU_MA splits the incoming (from L2 cache) AXI4 traffic into MPU_AXI2OCP and EMIF accesses. The MPU_AXI2OCP accesses are sent to the memory adapter A2O ports. The EMIF accesses are optionally interleaved between the two EMIF ports. Mandatory firewall checks are performed on all accesses to the EMIFs.

The main features of the MPU_MA are:

- Splits accesses between the MPU_AXI2OCP and the EMIF
- Input from L2 cache and output to MPU_AXI2OCP is AMBA4-compatible and runs at half the Cortex-A15 CPU frequency
- Supported read response interleaving on A2O port
- Parallel processing of reads and writes
- Support for narrow bursts
- Supports 4 × 128-bit line fills and eviction with critical word first
- Supports barrier instructions on normal read and write channels
- Direct 128-bit interfaces to each of EMIF1 and EMIF2:
  - Single request multiple data
  - No write response on posted writes
- Performs interleaving functions to load-balance the activities across the two EMIFs
- Uses firewall logic to check access rights of incoming addresses. The firewall on both EMIFs supports:
  - Configurable number of regions with fixed priority
  - Access support for up to eight execution domains
  - Busy indicator during reconfiguration
- Blocked read and write access to the EMIF for all accesses failing authorization checks
- Burst wrap for single cache line fills
- Supports boot from EMIF space
- Supports 8 GiB of memory. 6GiB accessible only by the MPU (MPU-only memory) and 2GiB shared among the system. Only 4 GiB are physically available on this device. 2GiB accessible only by the MPU and 2 GiB shared among the system.
- MA_LSM supports programmable interleaving for 2 GiB of shared memory:
  - Programmable multizone DRAM mapping and interleaving configuration
  - Supports four prioritized sections for defining configuration regions within the external memory
  - Supports interleaving at 128-, 256-, and 512-byte boundaries
- Fixed interleaving for extended 6 GiB of MPU-only memory
- Supports standard disconnect and idle protocols for independent powering down of the MPU_MA and both EMIFs (the EMIF must be powered down or up as a pair)
- Probe interface for performance monitoring of the EMIF ports
- 11 outstanding reads and 16 outstanding writes
- Supports exclusive accesses used for MPU internal synchronization
- Provides watchpoint capability on AXI bus. For more information, see Section 4.3.4.7.

Figure 4-6 shows the integration of the MPU_MA in the device.

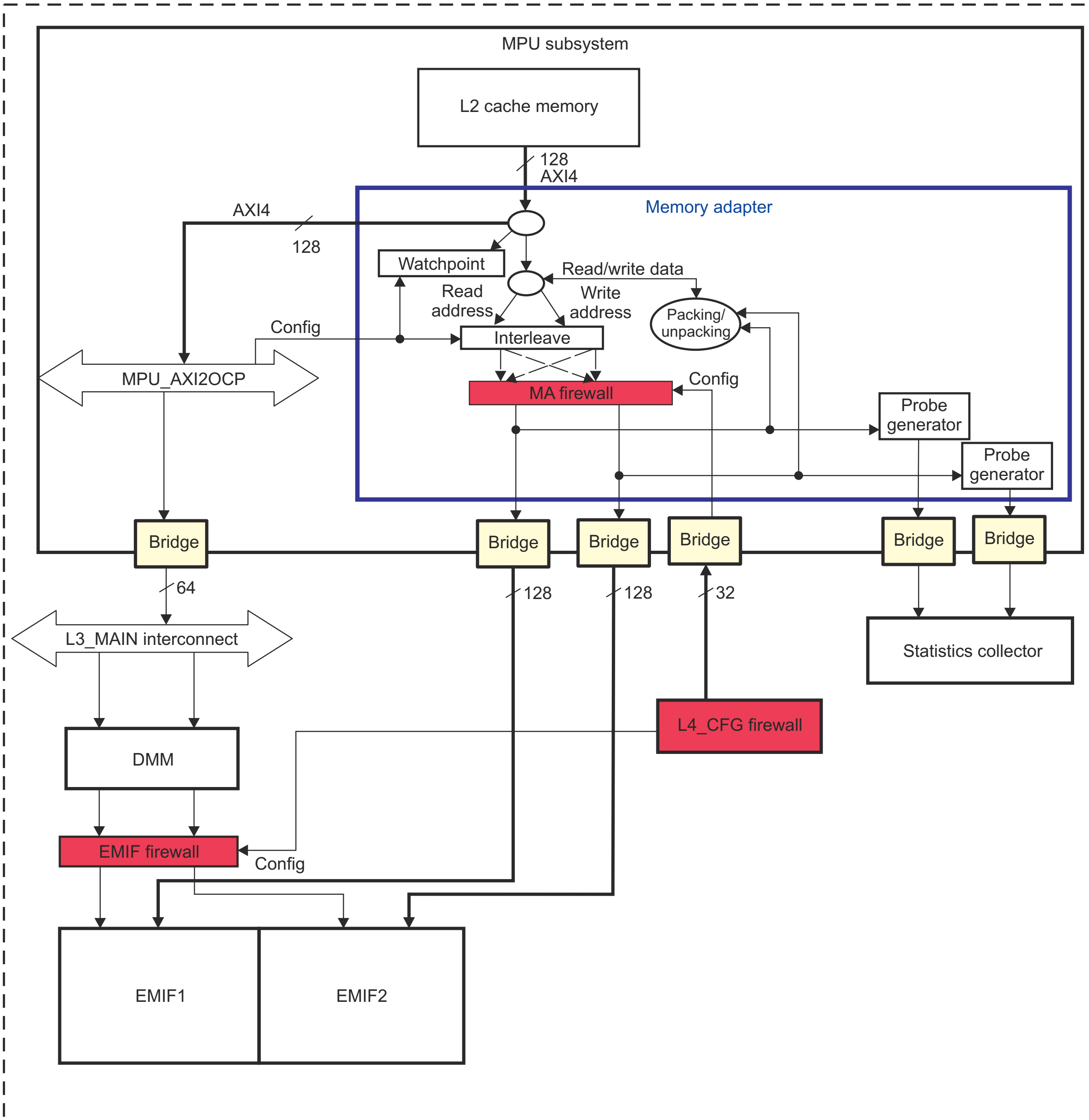


Figure 4-6. MPU_MA Overview

4.3.4.2 AXI Input Interface

The AXI input is driven from the single external port of the L2 cache. The incoming accesses are directed to the MPU_AXI2OCP module through the A2O port or begin their processing for a potential EMIF access. The routing of accesses is based on the AXI address and the EMIF boot input (pi_emifboot).

Table 4-2 lists the AXI access memory mapping. If the accesses are targeted for the MPU_AXI2OCP, then they are registered and sent to the A2O port without any alterations in its attributes (size, ID, etc.). The reserved space in the lower order memory area is a result of providing 8 GiB of continuous space for all the nonaliased regions.

The EMIF memory space is broken into eight 1-GiB sections, which are denoted *emif(a)* through *emif(h)*. This partitioning helps to specify the aliasing of several memory ranges. It also aids in the discussion of how the MPU memory space is mapped into the physical EMIF space.

Note

Although the MPU has 8-GiB EMIF dedicated memory space, only 4 GiB are used. 4 GiB is the total of physically available SDRAM for this device.

It is expected that the OS uses either the lower 2-GiB space and the lower aliased address of emif(a) and (b), or the continuous 8-GiB space and the upper aliasing of the emif(a) and (b) for all EMIF accesses.

Table 4-2. AXI Access Memory Mapping

Region Name	Start Address	End Address	Interleaving	MPU_MA Action
Boot space	0x00 0000 0000	0x00 000F FFFF	Default	Sent to Firewall/EMIF, if pi_emifboot = 1
			N/A	Sent to A2O port, if pi_emifboot = 0
Non-emif	0x00 0010 0000	0x00 7FFF FFFF	N/A	Sent to A2O port
Emif(a)	0x00 8000 0000	0x00 BFFF FFFF	Programmable	Aliased to address range 0x02 8000 0000–0x02 FFFF FFFF
Emif(b)	0x00 C000 0000	0x00 FFFF FFFF	Programmable	
Reserved	0x01 0000 0000	0x01 FFFF FFFF	N/A	DECERR returned
Emif(c)	0x02 0000 0000	0x02 3FFF FFFF	Fixed	Sent to Firewall/EMIF
Emif(d)	0x02 4000 0000	0x02 7FFF FFFF	Fixed	Sent to Firewall/EMIF
Emif(a)	0x02 8000 0000	0x02 BFFF FFFF	Programmable	Sent to Firewall/EMIF
Emif(b)	0x02 C000 0000	0x02 FFFF FFFF	Programmable	Sent to Firewall/EMIF
Emif(g)	0x03 0000 0000	0x03 3FFF FFFF	Fixed	Sent to Firewall/EMIF
Emif(h)	0x03 4000 0000	0x03 7FFF FFFF	Fixed	Sent to Firewall/EMIF
Emif(e)	0x03 8000 0000	0x03 BFFF FFFF	Fixed	Sent to Firewall/EMIF
Emif(f)	0x03 C000 0000	0x03 FFFF FFFF	Fixed	Sent to Firewall/EMIF
Reserved	0x04 0000 0000	0xFF FFFF FFFF	N/A	DECERR returned

4.3.4.3 Interleaving

To load-balance the activities across the two EMIFs, interleaving is used. For the lower 2-GiB address space, which is shared between the system and the MPU, a wide range of interleaving options are provided using the MA_LISA_MAP_i registers. Because the EMIF memories are accessible by the MPU subsystem through MPU_MA and by the L3_MAIN interconnect through the DMM, the same interleaving scheme must be used in both paths. To ensure compatibility, the interleaving function is implemented by a scaled down version of the LISA Section Manager (LSM), which implements the interleaving function in the DMM.

The high-order memory space, which is accessible only from the MPU subsystem, uses a simple fixed-interleaving scheme, which can be disabled. Heavy use in high memory space under noninterleaved configuration affects the balancing of the system access in lower-order memory.

4.3.4.3.1 High-Order Fixed Interleaving Model

The memory address ranges *emif(c)* through *emif(h)* can be interleaved between the two EMIFs by setting the MA_PRIORITY[8] HIMEM_INTERLEAVE_UN bit to 0x1. If enabled, this high-order interleaving occurs on a fixed 256-byte boundary, unlike emif(a) and emif(b) programmable interleaving which can be configured to 128-, 256-, and 512-byte boundary through the MA_LISA_MAP_i[19:18] SDRC_INTL bit field and is enabled through the MA_LISA_MAP_i [9:8] SDRC_MAP bit field.

Figure 4-7 and Figure 4-8 show how sections emif(a) through emif(h) are mapped into the EMIF1 and EMIF2 address spaces. The MA_LISA_MAP_i registers can be programmed to allow emif(a) and emif(b) address ranges to be remapped anywhere in EMIF1 or EMIF2 address spaces. Restricting EMIF1 and EMIF2 address spaces to the locations 0x0000 0000–0x7FFF FFFF allows the fully programmable emif(a) and emif(b) regions to co-exist with the fixed interleaving regions (emif(c) through emif(h)) without overlap. Although it is not recommended, the user can program the interleaver to map the emif(a) and emif(b) address ranges to the EMIF 0x8000 0000–0xFFFF FFFF location. If this is done, then additional address aliasing exists between emif sections (a/b) and sections (c) through (h).

Note

In fixed interleaving, memory address space 0 is always used.

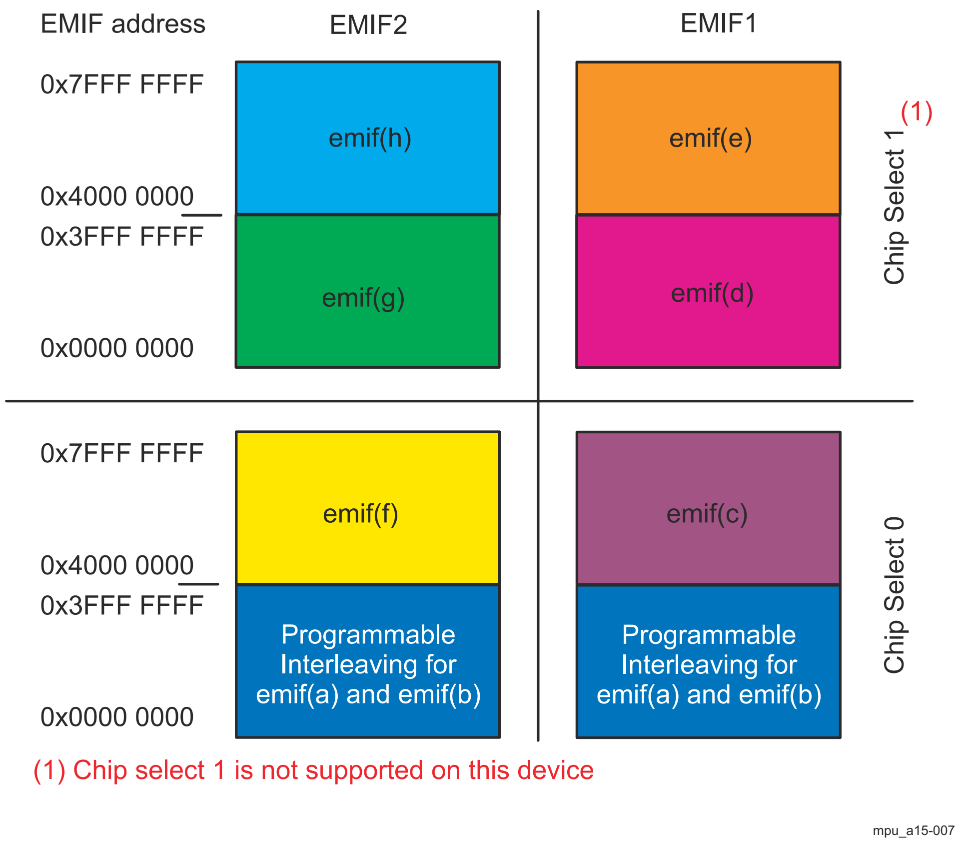


Figure 4-7. Fixed MPU-to-EMIF Address Mapping Without High-Order Interleaving

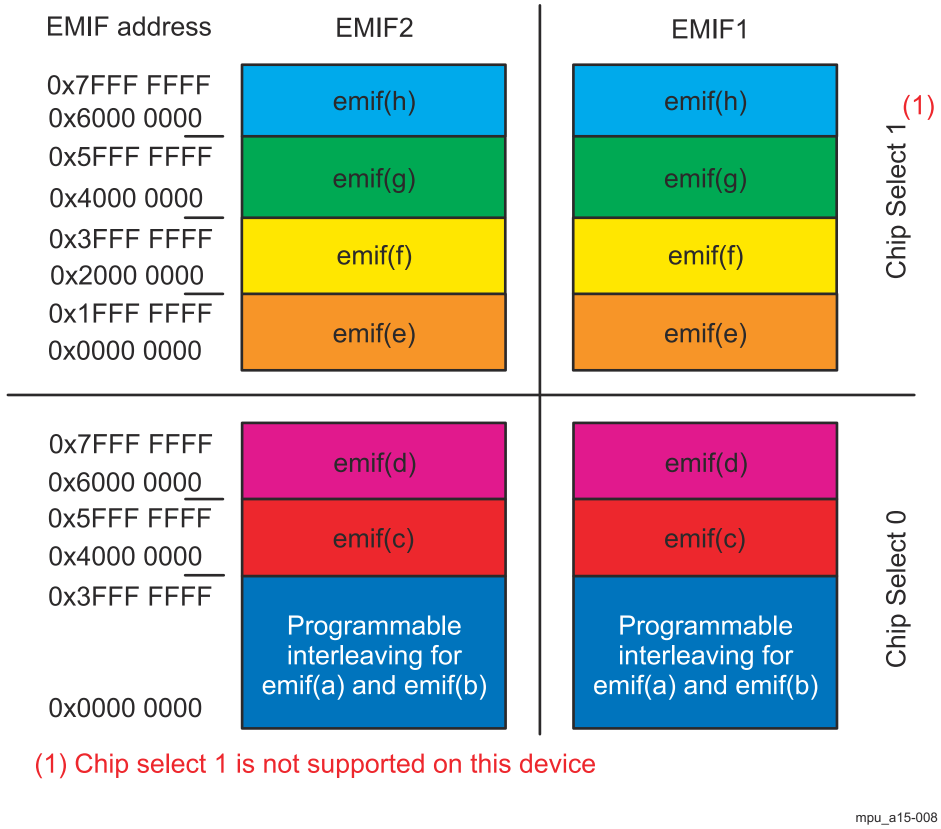


Figure 4-8. Fixed MPU-to-EMIF Address Mapping With High-Order Interleaving

4.3.4.3.2 Lower 2-GiB Programmable Interleaving Model

For more information about the lower 2-GiB programmable interleaving model of the MA_LSM, see the *Dynamic Mapping*, and *Address Mapping* sections in Section 15.2, *Dynamic Memory Manager*. When reading, replace DMM with MA_LSM.

Note

The MA_LSM is associated with the lower 2GiB shared memory space, that is, emif(a) and emif(b). In this case there is programmable interleaving (128B, 256B and 512B) and programmable MPU-to-EMIF address mapping configured trough the MA_LISA_MAP_i registers.

The MA_LSM is NOT associated with the 6GiB of MPU-only memory, that is, Emif(c) through Emif(h). In this case there is fixed interleaving (256B only) and fixed MPU-to-EMIF address mapping which cannot be programmed using the MA_LISA_MAP_i registers because MA_LSM is not used with address ranges greater than 2GiB. The fixed MPU-to-EMIF address mapping is shown in Figure 4-7 and Figure 4-8.

4.3.4.3.3 Local Interconnect and Synchronization Agent (LISA) Section Manager

The LSM module in the MPU_MA contains one set of registers and two ports. Each port can handle one address translation, provide the interface size, and determine the EMIF mapping. The two ports process read and write addresses simultaneously. The MA_LSM supports four sections.

4.3.4.3.4 MA_LSM Registers

Table 4-3 lists the DMM registers that are duplicated locally in the MPU_MA. These registers have the same names except that DMM is replaced by MPU_MA. Software must keep the content of these registers equal to the content of their counterparts in the DMM.

Table 4-3. MPU_MA Registers Duplicated From the DMM Register Map

Register	Type	Width (Bits)	Base Address
MA_LISA_LOCK	RW	32	0x482A F01C
MA_LISA_MAP_i (where i = 0 to 3)	RW	32	0x482A F040–0x482A F04C

For descriptions of the registers, see Section 15.2, *Dynamic Memory Manager*. The only differences are the following reset values of the MA_LISA_MAP_0 register:

- MA_LISA_MAP_0[22:20] SYS_SIZE = 0x1 (32-MiB section)
- MA_LISA_MAP_0[17:16] SDRC_ADDRSPC = 0x1
- MA_LISA_MAP_0[9:8] SDRC_MAP = 0x1

The configuration of the MPU_MA must be programmed by the Cortex-A15 MPU to match the configuration of the DMM during boot. The registers must be locked after initial programming to avoid corruption by application software. When the SM configuration registers are being programmed, there are no interlocks or safeguards to prevent accesses from being processed at the same time. The MA_LISA_LOCK and MA_LISA_MAP_i registers are retained during power down of retention mode.

If an access is made to a region marked as having unused address space or an unmapped SDRC map, no access is made to the EMIF and an error is returned. If an access is made to a region marked as accesses-SDRC-internal-registers, no access occurs to the EMIF and an error is returned. This is because the internal registers are not accessible from this port.

If an access is made to a region that is not mapped to any LISA section, or selects a section with one of the following programmations, then no access is made to the EMIF and an error is returned on the AXI port:

- SDRC_MAP field equal to UNMAPPED
- SDRC_ADDRSPC field equal to UNUSED

4.3.4.3.5 Posted and Nonposted Writes

On posted writes, a response is returned after all write data is accepted by the MPU_MA and one of the following occurs:

- The request is canceled, because of an interleaving error.
- The request is pushed into the firewall FIFO.

On nonposted writes, a response is not returned until after all write data is accepted by the MPU_MA and one of the following occurs:

- The request is canceled, because of an interleaving error.
- The request has a firewall violation.
- The associated response is received from the EMIF (access is completed in the EMIF).

When the FORCEWRNP[0] MPUFORCEWRNP bit is set to 0x1, all writes are forced to be converted to nonposted writes (prevents posted writes).

4.3.4.3.6 Errors

If the region is marked as unmapped, o targets EMIF internal registers or reserved space, access is blocked and an error response is returned. Such an error occurs on the first half of a potential split access and aborts the entire access at that time. DECERR errors ultimately generate Arm aborts. See the Arm Fault Status Register for more information.

To avoid undefined results, the following program rules must be followed:

- A section cannot be mapped as interleaving (MA_LISA_MAP_i[9:8] SDRC_MAP = 0x3), and the MA_LISA_MAP_i[19:18] SDRC_INTL bit field must be equal to 0x0.
- If the section maps to a single EMIF (MA_LISA_MAP_i[9:8] SDRC_MAP = 0x1 or 0x2), the MA_LISA_MAP_i[7:0] SDRC_ADDR bit field must be aligned to the MA_LISA_MAP_i[22:20] SYS_SIZE bit field (that is, the lower SDRC_ADDR bits must be 0).

- If the section maps to both EMIFs (MA_LISA_MAP_i[9:8] SDRC_MAP = 0x3), the MA_LISA_MAP_i[7:0] SDRC_ADDR bit field must be aligned to MA_LISA_MAP_i[22:20] SYS_SIZE – 1, because the addressable space for each EMIF is half.
- If the section maps to both EMIFs (MA_LISA_MAP_i[9:8] SDRC_MAP = 0x3), the MA_LISA_MAP_i[22:20] SYS_SIZE bit field cannot be 0.

4.3.4.4 Statistics Collector Probe Ports

To enable performance monitoring by the debug subsystem, a subset of interconnect signals must be monitored and sent to a statistics collector in the CORE domain. For more information about the statistics collector, see Chapter 33, *On-Chip Debug Support* chapter.

4.3.4.5 MPU_MA Firewall

A firewall (MA_MPU_NTTP_FW) exists between the interleaving logic and each EMIF port. The firewall checks the address and access qualifiers against the permission attributes for the highest priority region to which the address belongs. If the access fails to get permission, a violation occurs and the fw_func_error or fw_debug_error output is asserted. The failing access is blocked and a slave error response is returned on the MPU_MA port for read and nonposted accesses. For more information about the MPU_MA firewall, see *Interconnect*.

4.3.4.6 MPU_MA Power and Reset Management

The MA_LSM supports only smart-idle mode.

4.3.4.7 MPU_MA Watchpoint

The MPU_MA implements a Watchpoint unit (MPU_MA_WP) attached to its AXI input interface and thus allowing the user to track certain AXI transactions. The MPU_MA_WP can be programmed to generate a trigger when a specified AXI transaction (or a chain of AXI transactions) satisfies a set of user-defined attributes. Upon triggering (that is, upon transaction match), additional information about the target transaction is captured in dedicated log registers.

4.3.4.7.1 Watchpoint Types

The MPU_MA_WP supports three types of watchpoints:

- Data watchpoint
- Memory barrier watchpoint
- Chained watchpoint (combination of data watchpoint and memory barrier)

Two types of chaining are supported based on the configuration of the DBG_HWWP0_CHAIN_CNTL[1] CHAIN_TYPE bit:

- Data watchpoint match *followed* by memory barrier match
- Data watchpoint match *preceded* by memory barrier match

Chaining is enabled by setting the DBG_HWWP0_CHAIN_CNTL[0] CHAIN_WP_EN bit to '1'. Note that both the data watchpoint and the memory barrier must also be subsequently enabled (by setting the DBG_HWWP0_MAIN_CNTL[0] WP_EN and DBG_HWWP0_MEM_CNTL[0] MEM_BAR_WP_EN bits, respectively) for chaining to work. To avoid race conditions and partial matches, no chained match may occur until the complete chain of watchpoints has been fully enabled.

4.3.4.7.2 Transaction Filtering Options

The following filtering options can be specified for a **data watchpoint match** condition (via the DBG_HWWP0_MAIN_CNTL and DBG_HWWP0_AUX_CNTL registers):

- Address within or outside a specific range
- Various access types:
  - Read or Write (non-posted/posted/either) or any (no preference)
  - Supervisor or User or any (no preference)
  - Data or Instruction or any (no preference)
- MPU_MA target: AXI2OCP bridge or EMIF or any (no preference)

- Initiator: CPU0 or CPU1 or unknown source (ACP, FEQ, etc.) or any (no preference)

The following filtering options can be specified for a **memory barrier match** condition (via the DBG_HWWP0_MEM_CNTL register):

- Type of memory barrier instruction: DSB or DMB or any (no preference)
- Type of memory barrier access: Read or Write or any (no preference)

4.3.4.7.3 Transaction Match Effects

The following information concerns transaction attributes logging depending on the transaction match:

- **Data watchpoint match**: When a data watchpoint matches, the following transaction attributes are logged irrespective of the target of the transaction (not all attributes may be applicable for all targets):
  – 40-bit AXI physical address of the transaction
    - Related log registers: DBG_HWWP0_HG_ADDR0_LOG and DBG_HWWP0_LW_ADDR0_LOG
  – 128-bit data associated to the *n*-th beat of the burst transaction, where *n* is configured by the user (before enabling the watchpoint) in the DBG_HWWP0_MAIN_CNTL[23:20] BEAT_SEL bit field
    - Related log registers: DBG_HWWP0_DATA0_LOG, DBG_HWWP0_DATA1_LOG, DBG_HWWP2_DATA0_LOG, and DBG_HWWP3_DATA0_LOG
  – Byte Enable: Specifies the 16 data byte enables associated with the up to 128-bit data captured
    - Related log register: DBG_HWWP0_DATA_TRANS_ATTR0_LOG[15:0] BYTE_EN
  – Initiator Info: Specifies which initiator generated the transaction
    - Related log register: DBG_HWWP0_TRANS_ATTR0_LOG[22:20] INIT_INFO
  – Response Info: Specifies whether the response indicates transaction success or failure
    - Related log register: DBG_HWWP0_TRANS_ATTR0_LOG[25:24] RESP_INFO
  – Target Info: Specifies the target of the access
    - Related log register: DBG_HWWP0_TRANS_ATTR0_LOG[18:16] TARGET_INFO
  – Transaction Type: Specifies the type of the transaction which matches
    - Related log register: DBG_HWWP0_TRANS_ATTR0_LOG[12:10] TRANS_TYPE
  – Burst Length (1..63)
    - Related log register: DBG_HWWP0_TRANS_ATTR0_LOG[9:4] BURST_LENGTH
  – Burst Type: Specifies the type of the burst (only relevant if Burst Length > 1)
    - Related log register: DBG_HWWP0_TRANS_ATTR0_LOG[2:0] BURST_TYPE
  – Additional attributes, such as:
    - Data/Instruction access (DBG_HWWP0_TRANS_ATTR1_LOG[2] DATA)
    - Supervisor/User access (DBG_HWWP0_TRANS_ATTR1_LOG[1] SUPERVISOR)
- **Memory barrier watchpoint match**: When a memory barrier watchpoint matches, transaction attributes logged are those of the transaction immediately following the memory barrier
- **Chained watchpoint match**: When a chained watchpoint matches (which implies that both memory barrier and data watchpoint have matched), transaction attributes logged are those of the transaction of the matching data watchpoint

4.3.4.7.4 Trigger Generation

The user can configure the MPU_MA_WP unit to generate a trigger upon watchpoint match. The MPU_MA_WP has a single trigger output (MA_WP_TRIGGER) which is mapped to the CTITRIGIN[6] input of the CS_CTI_S module and is shared by the following trigger sources:

- **Data watchpoint trigger**: Upon data watchpoint match, the DBG_HWWP0_MAIN_CNTL[31] TRIG bit is set (if trigger generation is enabled). This status bit is cleared upon 0->1 transition of the DBG_HWWP0_MAIN_CNTL[0] WP_EN bit
- **Memory barrier watchpoint trigger**: Upon memory barrier match, the DBG_HWWP0_MEM_CNTL[31] MEM_BAR_TRIG bit is set (if trigger generation is enabled). This status bit is cleared upon 0->1 transition of the DBG_HWWP0_MEM_CNTL[0] MEM_BAR_WP_EN bit

- **Chained watchpoint trigger**: Upon chained watchpoint match, the DBG_HWWP0_CHAIN_CNTL[31] CHAIN_WP_TRIG bit is set (if trigger generation is enabled). This status bit is cleared upon 0->1 transition of the DBG_HWWP0_CHAIN_CNTL[0] CHAIN_WP_EN bit

Note that the MPU_MA_WP module supports only a global enable (by setting the TRIG_CTRL[0] TRIG_EN bit) for all three sources listed above. It is SW responsibility to identify the source of the trigger generation by checking the corresponding trigger status bits.

4.3.4.7.5 Programming Options Summary

Below is a summary of the MPU_MA_WP programmable options:

- Match on a data watchpoint only:
  - Configure the DBG_HWWP0_MAIN_CNTL register
  - Configure the DBG_HWWP0_AUX_CNTL register
  - Set the DBG_HWWP0_MAIN_CNTL[0] WP_EN bit to enable data watchpoint
- Match on a memory barrier only:
  - Configure the DBG_HWWP0_MEM_CNTL register
  - Set the DBG_HWWP0_MEM_CNTL[0] MEM_BAR_WP_EN bit to enable memory barrier
- Match on a data watchpoint chained with a memory barrier:
  - Configure the DBG_HWWP0_CHAIN_CNTL[1] CHAIN_TYPE bit to chain data watchpoint *before* memory barrier or data watchpoint *after* memory barrier.
  - Set the DBG_HWWP0_CHAIN_CNTL[0] CHAIN_WP_EN bit to enable the chain (but both the memory and data watchpoints must also be subsequently enabled)
  - Configure the DBG_HWWP0_MAIN_CNTL register
  - Configure the DBG_HWWP0_AUX_CNTL register
  - Set the DBG_HWWP0_MAIN_CNTL[0] WP_EN bit to enable data watchpoint
  - Configure the DBG_HWWP0_MEM_CNTL register
  - Set the DBG_HWWP0_MEM_CNTL[0] MEM_BAR_WP_EN bit to enable memory barrier

4.3.5 Realtime Counter (Master Counter)

4.3.5.1 Counter Operation

The real-time counter (COUNTER_REALTIME) is a free-running counter which produces the count used by the private timer peripherals in the MPU_CLUSTER. For this reason, it is also called Master Counter.

COUNTER_REALTIME supports two modes of operation:

- **Active (FUNC)**: This is the normal mode in which the counter runs
- **Sleep / Low-Power (LP)**: The counter automatically switches to this mode when the MPU subsystem goes to standby mode

COUNTER_REALTIME has two clock inputs:

- **32K (slow) clock**: This is the FUNC_32K_CLK (SYSCLK1/610) clock
- **System (fast) clock**: This is the WKUPAON_ICLK clock (coming from the global PRCM). The two sources for this clock are:
  - SYS_CLK1: Main system clock of the SoC. Supported frequencies are: 19.2, 20, and 27 MHz
  - ABE_LP_CLK: Clock derived from DPLL_ABE, used in some low-power use cases; runs at 12.288 MHz (DPLL cascading)

COUNTER_REALTIME implements two internal counters: a 40-bit coarse counter, and a 48-bit system counter.

The coarse counter is driven by the FUNC_32K_CLK clock. The coarse counter is the only one running in LP mode (the system counter is off in LP mode) and it is also used in FUNC mode as a timebase reference for the system counter. During active mode, the coarse counter value is multiplied by 375/2. This effectively shifts left the base counter value. This initial value is output to the CPUs and is also fed back into the circuit.

The system clock is running in parallel to the 32K clock and it drives the active mode circuit of COUNTER_REALTIME, which includes the following basic blocks:

- 48-bit system counter
- 12-bit fractional incrementor. It has the following input parameters:
  – Numerator (N) for SYS mode: PRM_FRAC_INCREMENTER_NUMERATOR [11:0] SYS_MODE_NUMERATOR
  – Numerator for ABE_LP mode: PRM_FRAC_INCREMENTER_NUMERATOR [27:16] ABE_LP_MODE_NUMERATOR
  – Common Denominator (D): PRM_FRAC_INCREMENTER_DENUMERATOR_RELOAD [11:0] DENOMINATOR
- Fine alignment circuit

At each system clock rising edge, the system counter value is compared to the shifted (x375/2) coarse counter value and based on this comparison, the system counter is either incremented or unchanged.

Note

COUNTER_REALTIME is designed for a count rate of 6.144 million counts per second (32768Hz x 375/2). Since the SoC does not use a true 32768Hz source (instead uses SYS_CLK1/610, which is ~32787Hz for 20MHz SYS_CLK1), the resulting timer rate is not 6.144 and timer drift can accumulate. To minimize error in active mode, the adjustable fine increment rate MUST equal the fixed coarse increment rate.

Table 4-4 shows suggested COUNTER_REALTIME increment values.

Table 4-4. COUNTER_REALTIME Increment Values

COUNTER_REALTIME Clock Mode	COUNTER_REALTIME Clock Name	COUNTER_REALTIME Fast Source Rate (Hz)	Numerator (N) for Fast Source	Denominator (D) for Fast Source	COUNTER_REALTIME Fine Rate (Hz)	COUNTER_REALTIME 32K Source Rate (Hz)	COUNTER_REALTIME Fixed Coarse Rate (Hz)
			Target 32K*(375/2)		(Sys clock)*(N/D)	(SYSCLK1/610)	32K*(375/2)
System (SYS)	SYS_CLK1	19,200,000	75	244	5,901,639.3	31,475	5,901,639.3
		20,000,000	75	244	6,147,541.0	32,787	6,147,541.0
		27,000,000	75	244	8,299,180.3	44,262	8,299,180.3
ABE Low-Power (ABE_LP)	ABE_LP_CLK	12,288,000	1875	3904	5,901,639.3	31,475	5,901,639.3
			868	1735	6,147,541.2	32,787	6,147,541.0
			955	1414	8,299,179.6	44,262	8,299,180.3
Low-Power (LP)	FUNC_32K_CLK	NA	NA	NA	NA	31,475	5,901,639.3
		NA	NA	NA	NA	32,787	6,147,541.0
		NA	NA	NA	NA	44,262	8,299,180.3

4.3.5.2 Frequency Change Procedure

The change between the system clock (SYS_CLK1) and the ABE low-power clock (ABE_LP_CLK) is performed within global PRCM. A software-controllable bit (CM_CLKSEL_WKUPAON[0] CLKSEL) controls this change. A glitch-free clock multiplexer produces the resulting clock (WKUPAON_ICLK), which is used as the functional clock of the master counter. The CM_CLKSEL_WKUPAON[0] CLKSEL bit is exported from global PRCM to the master counter to control simultaneously the mode (SYS or ABE_LP).

The transition from SYS to ABE_LP mode (DPLL cascading entry) is done under software control as follows:

- DPLL_ABE must be locked before (under software control, global PRCM register CM_CLKMODE_DPLL_ABE[2:0] DPLL_EN = 0x7).
- Software writes 1 in the global PRCM register CM_CLKSEL_WKUPAON[0] CLKSEL bit to choose DPLL_ABE clock.

The transition from ABE_LP to SYS mode (DPLL cascading exit) is done under software control as follows:

- Software writes 0 in the global PRCM register CM_CLKSEL_WKUPAON[0] CLKSEL bit to choose SYS CLK.

- DPLL_ABE may be unlocked after.

Transition from FUNC (ABE_LP or SYS) to LP mode is done by hardware upon MPU subsystem standby entry sequence.

Transition from LP to FUNC (ABE_LP or SYS) mode is done by hardware upon MPU subsystem standby exit sequence.

The frequency change involves some uncertainty on the counter due to a glitch-free clock MUX in global PRCM. In order not to accumulate them, a free-running coarse counter on the FUNC_32K_CLK clock holds the reference count, used for realignment when necessary. To load and reload the reference count value (x375/2) from the coarse counter, the user must write 1 in the PRM_FRAC_INCREMENTER_DENUMERATOR_RELOAD[16] RELOAD bit (but it must be 0 prior to that).

Once configured (at first boot time), the counter is running in all modes without any action required by software.

4.3.6 MPU Watchdog Timer

The MPU watchdog timer (MPU_WD_TIMER) implements two channels, one per MPU core (MPU_WD_TIMER_C0 and MPU_WD_TIMER_C1, respectively; an unified name MPU_WD_TIMER_Cx is used hereafter in the chapter). The MPU_WD_TIMER operates on MPU subsystem clock (MPU_DPLL_CLK).

Each MPU_WD_TIMER_Cx channel implements:

- A 32-bit decrementing counter which has a period set by the value loaded into the counter (via the WDT_LOAD_REGISTER_i[31:0] NEWCOUNT bit field) and the prescaler ratio (set via the WDT_PRESCALER_REGISTER_i[9:0] PRESCALER bit field). The period is calculated as follows: $T_{MPU\_WD\_TIMER\_Cx}$ = (PRESCALER + 1) x (NEWCOUNT + 1) / f(MPU_DPLL_CLK).
- Two interrupt output signals (WARN, INTR)
- One reset request output (MPUSSRST)

The counter starts decrementing when the WDT_CONTROL_REGISTER_i[0] ENABLE bit is set to 0x1. The current count value can be monitored by reading the WDT_COUNT_REGISTER_i[31:0] CURRENTCOUNT bit field. When the counter reaches zero, a timeout condition occurs. In the timeout condition, the counter stops counting and:

- MPU_WD_TIMER_Cx_IRQ interrupt is generated to the MPU_INTC, if enabled by setting the WDT_CONTROL_REGISTER_i[1] INTREN bit to 0x1.
- Reset request is generated to the global PRCM, if enabled by setting the WDT_CONTROL_REGISTER_i[3] MPUSSRSTEN bit to 0x1.

Note

If the MPU core corresponding to the MPU_WD_TIMER_Cx channel is in debug state, the counter does not decrement until the MPU core returns to non-debug state. Debug state is inferred by monitoring the DBGACK signal corresponding to this core.

Additionally, the user can also setup a warning condition which can be used to signal an interrupt that gives software a notice when the MPU_WD_TIMER_Cx is getting close to a timeout. The threshold value is set in the WDT_WARNING_REGISTER_i[31:0] WARNING_WATERMARK bit field. The current count value is then compared to the threshold (warning watermark) level value and when CURRENTCOUNT = WARNING_WATERMARK, a warning interrupt (MPU_WD_TIMER_Cx_IRQ_WARN) is generated to the MPU_INTC (if enabled by setting the WDT_CONTROL_REGISTER_i[1] WARNEN bit to 0x1).

The mapping of the four MPU_WD_TIMER interrupts is as follows:

- MPU_WD_TIMER_C0_IRQ_WARN mapped to MPU_IRQ_5
- MPU_WD_TIMER_C1_IRQ_WARN mapped to MPU_IRQ_6
- MPU_WD_TIMER_C0_IRQ mapped to MPU_IRQ_139
- MPU_WD_TIMER_C1_IRQ mapped to MPU_IRQ_140

The user can also poll the following status bits:

- WDT_RESET_STATUS_REGISTER_i[0] TO, to know when the timeout condition has occured. This bit might also be used to figure out which MPU_WD_TIMER_Cx signalled a reset request.
- WDT_RESET_STATUS_REGISTER_i[1] WARN, to know when the warning condition has occured.

The following programming guidelines should be taken into account:

- The WDT_PRESCALER_REGISTER_i register should be written (if needed) before the WDT_LOAD_REGISTER_i register is written. This is because when the WDT_LOAD_REGISTER_i register is written, the WDT_COUNT_REGISTER_i register is immediately updated with this value and at the same time, the PRESCALER value is sampled to be used by the decrement logic which controls the WDT_COUNT_REGISTER_i register.
- The WDT_WARNING_REGISTER_i and WDT_LOAD_REGISTER_i registers should be written before the MPU_WD_TIMER_Cx is enabled (WDT_CONTROL_REGISTER_i[0] ENABLE = 0x1). Otherwise, interrupts and reset request may be asserted immediately depending on the state of these registers. For example, after reset these registers have '0' and if the WDT_CONTROL_REGISTER register is configured to enable the corresponding interrupts and reset request, and then MPU_WD_TIMER_Cx is enabled, interrupts and reset request are immediately asserted.

The suggested programming order is as follows:

1. Set the warning watermark level (WDT_WARNING_REGISTER_i), if needed
2. Set the prescaler ratio (WDT_PRESCALER_REGISTER_i[9:0] PRESCALER)
3. Set the new count value (WDT_LOAD_REGISTER_i)
4. Enable corresponding interrupts and reset request in WDT_CONTROL_REGISTER_i, if needed
5. Enable MPU_WD_TIMER_Cx (WDT_CONTROL_REGISTER_i[0] ENABLE = 0x1)

Note

When the MPU cores are going to low power state, the MPU_WD_TIMER may need to be disabled. If it is not disabled, then the MPU_WD_TIMER may timeout (since the MPU core is not refreshing the timeout counters) and will generate MPUSS reset request which will reset the MPU domain, including both MPU cores.

4.3.7 MPU Subsystem Power Management

The MPU subsystem implements a local PRCM (MPU_PRCM) module to handle the local Cortex-A15 CPU power domains, along with the corresponding L1 cache. The MPU_PRCM module includes two power-management control (PSCON) modules to control the power chains for MPU_C0 and MPU_C1. The PRM_PSCON_COUNT register is used for that control purpose.

In addition to the standard power-management technique supported in the device, the MPU subsystem also employs an SR3-APG (SmartReflex3 automatic power gating) power-management technology to reduce leakage. This technology allows for full logic and memories retention on MPU_C0 and MPU_C1 and is controlled by the MPU_PRCM. The SR3-APG power-management can be enabled by setting the PRM_PSCON_COUNT[24] HG_EN bit. For more information about how to enable SR3-APG fast-wakeup, see Section 4.3.7.6, *SR3-APG Technology Fail-Safe Mode*.

4.3.7.1 Power Domains

The MPU subsystem is divided into six power domains controlled by the local or global PRCM module, as shown in Figure 4-9.

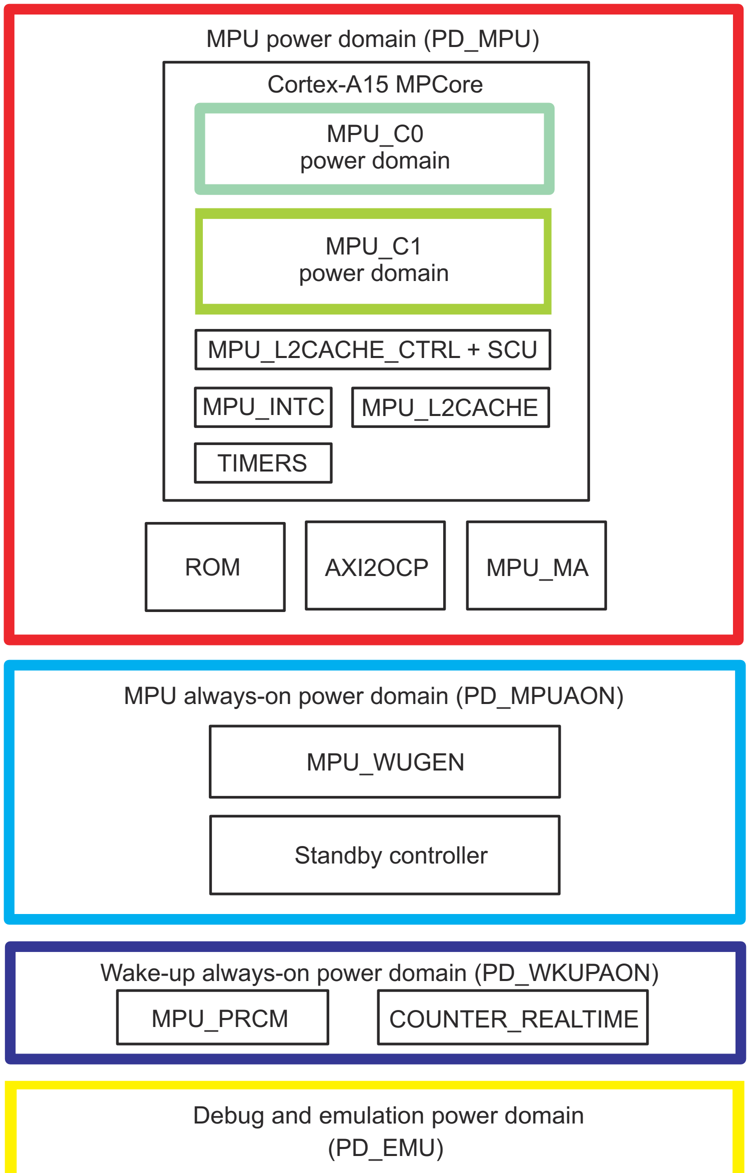


Figure 4-9. MPU Subsystem Power Domains Overview

Note

The MPU debug and trace modules are implemented in various power domains. For more information, see Chapter 33, *On-Chip Debug Support*.

The device-level power domains are directly aligned with voltage domains and thus can be represented as a cross-reference to the different voltage domains.

For information about the physical power domains (PD_MPU, PD_MPUAON, PD_WKUPAON, and PD_EMU) and the related voltage domains, see *Power, Reset, and Clock Management*.

4.3.7.2 Power States of MPU_Cx

MPU_Cx (where x = 0 or 1) changes power states only when the StandbyWFI signal is asserted. There is no signal coming from the MPU cores, or the MPU_CLUSTER to define in which power state MPU_Cx can go. Software must program such information by writing to the PM_CPUx_PWRSTCTRL[1:0] POWERSTATE bit field before executing the WFI instruction. Table 4-5 provides the software requirements before executing the WFI instruction, and the condition to return to RUN mode.

Table 4-5. MPU_Cx State Transitions

Low-Power State	Software Sequence Before Executing WFI	Wakeup (Transition Back to Run Mode)
WFI/ON Logic ON L1$ ON	Execute a Data Synchronization Barrier (DSB) instruction.	Managed locally to MPU_Cx upon one of the following sources: • An interrupt, masked or unmasked (important: masked does not mean disabled) • An imprecise data abort, regardless of the value of the CPSRA bit • A debug request generated by writing 1 to the HALTING DEBUG MODE bit of the Cortex-A15 DBGSCR register, or the assertion of the Cortex-A15 EDBGRQ pin • Global device cold or warm reset source
WFI/INACT Logic ON L1$ ON	Execute a DSB instruction.	Managed locally by MPU_PRCM upon following source: • An interrupt, enabled at MPU_WUGEN level • A forced wakeup (sw_wkup) • A full debug wake-up request sequence: ForceActive from the DAP_PC, a debug request (as explained previously), assert block reset, wait for processor to halt and clear the block reset • Global device cold or warm reset source
WFI/RETENTION Logic RET L1$ ON L1$ peripheral OFF	1. Set the PM_CPUx_PWRSTCTRL[1:0] POWERSTATE bit field to 0x1 (RETENTION state). 2. Execute a DSB instruction.	Managed locally by MPU_PRCM upon following source: • An interrupt, enabled at MPU_WUGEN level • A forced wakeup (sw_wkup) • A full debug wake-up request sequence: ForceActive from the DAP_PC, a debug request (as explained previously), assert block reset, wait for processor to halt and clear the block reset • Global device cold or warm reset source MPU_PRCM does not need to reset MPU_Cx.

Table 4-5. MPU_Cx State Transitions (continued)

Low-Power State	Software Sequence Before Executing WFI	Wakeup (Transition Back to Run Mode)
WFI/FORCED_OFF[(1)] Logic OFF L1$ OFF	1. Save MPU_INTC configuration for MPU_C1 interrupts. 2. Disable SGI and PPI toward MPU_C1 in MPU_INTC. 3. Save all Arm registers, including CPSR and SPSR. 4. Save all CP15 registers. 5. Save all debug-related states. 6. Set the PM_CPU1_PWRSTCTRL[1:0] POWERSTATE bit field to 0x0 (OFF state). 7. Set the PM_CPU1_PWRSTCTRL[7] FORCED_OFF bit to 0x1. 8. Clear the C-bit in the SCTLR to prevent further data cache allocation. 9. Clean/invalidate L1$ content on an as-needed basis. 10. Clear SMP bit to take CPU out of coherency. 11. Execute an ISB instruction to ensure that all of the above CP15 register changes have been committed. 12. Execute a DSB instruction.	Managed locally by MPU_PRCM upon following source: • A forced wakeup (sw_wkup) • A full debug wake-up request sequence (as previously mentioned). Block reset is not supported. • Global device cold or warm reset source MPU_PRCM must reset MPU_C1 to make it reboot and restore its context, including MPU_INTC.

(1) Applies to MPU_C1 only.

Note

For the description of the Cortex-A15 CPU registers and the DSB/ISB instructions, see the Arm *Cortex-A15 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

The RETENTION low-power state is not natively supported by the MPU_CLUSTER. This mode is implemented with SR3-APG power-management technology. The MPU subsystem power-management hardware is designed to ensure that the system does not have an L1 cache coherency problem when putting both MPU cores in retention mode. In this mode, the MPU_Cx logic is in full retention with all memory content preserved by keeping the array of memories fully powered and the logic of the memory peripheries shut down. In slow wake-up mode, memories are put into retention to prevent more leakage.

In FORCED_OFF and RETENTION low-power states, the standby controller gates the clock to the MPU_CLUSTER by deasserting the CLKEN signal before signaling the MPU_PRCM to perform a power transition. In these low-power states, the MPU core can be wakened only by the MPU_PRCM. A number of important actions must be performed by software before entering such a state.

FORCED_OFF mode applies only to MPU_C1. In this mode, it is critically important for software to clear the SMP bit of the targeted MPU core to take that MPU core out of coherency and to prevent TLB, BTB, or instruction cache maintenance operations from other MPU core in the cluster being issued to this MPU core. The SMP bit is part of the Auxiliary Control Register (ACTLR); there is one ACTLR per MPU core. The ACTLR is a CP15 register. For more information about this register, see the Arm *Cortex-A15 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

A wakeup from FORCED_OFF low-power state must always happen through a MPU core (MPU_C1) reset so that the MPU core (MPU_C1) restores its state to the same level as before the power transition before handling the wake-up event itself (the interrupt). A wakeup from RETENTION low-power state does not need to happen through a MPU core reset because MPU core logics are fully retained.

Note

The private peripheral interrupts (PPIs) cannot wake up an MPU core in INACT, RETENTION, or FORCED_OFF low-power state. Software-generated interrupts (SGI) cannot wake up an MPU core in the same low-power state. However, it is possible to work around the issue by applying the following method:

- When MPU_Cx wants to send an SGI to MPU_Cy (that is, the other MPU core), it must first systematically set the CM_CPUy_CLKSTCTRL[1:0] CLKTRCTRL bit field to 0x2 (software-forced wake-up transition),
- The MPU_Cy corresponding ISR must write back the CM_CPUy_CLKSTCTRL[1:0] CLKTRCTRL bit field to 0x3 (HW_AUTO, automatic transition).

Table 4-6 gives details of the power state of the supported MPU_Cx and the corresponding values of the MPU_PRCM register.

Table 4-6. MPU_Cx Supported Power States

Hardware Conditions		MPU_Cx Programming Model		Resulting MPU_Cx State			
StandbyWFI StandbyWFE	State of L2/Other MPU core	MPU_PRCM Power State PM_CPUx_PWRSTCTRL[1:0] POWERSTATE	MPU_PRCM Clock Transition Control CM_CPUx_CLKSTCTRL[1:0] CLKTRCTRL	Logic	L1 Cache	Arm Cortex-A15 Internal Clock	Power State at MPU_PRCM
MPU_Cx running	Any	Any	Any	ON	ON	ON	ON
MPU_Cx in WFE	Any	Any	Any	ON	ON	OFF	ON
MPU_Cx in WFI	Any	Any	NO_SLEEP/ SW_WKUP	ON	ON	OFF	ON
MPU_Cx in WFI	Any	ON	HW_AUTO	ON	ON	OFF	ON
MPU_C1 in WFI	Any	FORCED_OFF(1)	HW_AUTO	OFF	OFF	OFF	OFF
MPU_Cx in WFI	!= (IDLE/WFI/ WFI)	Any	HW_AUTO	ON	ON	OFF	ON
MPU_Cx in WFI	IDLE/WFI/ WFI	INACT	HW_AUTO	ON	ON	OFF	INACT
MPU_Cx in WFI	IDLE/WFI/ WFI	RETENTION	HW_AUTO	SR3-APG/ ON(2)	ON/ RETENTION (3)	OFF	CSWRET

(1) FORCED_OFF mode applies only to MPU_C1. See Table 4-5 for the guidelines about how to enter FORCED_OFF mode.
(2) If PRM_PSCON_COUNT[24] HG_EN = 1, MPU core logic is put in SR3-APG state. If not, MPU core logic is kept ON.
(3) L1$ state depends on the values of the PRM_PSCON_COUNT[25] HG_RAMPUP and PRM_PSCON_COUNT[24] HG_EN bits. When CPU PD is in the CSWRET state, if HG_ENABLE = 0x1 and HG_RAMPUP = 0x1, then L1$ memory is in ON state, else L1$ memory is in the RETENTION state. For more information, see Section 4.3.7.6, *SR3-APG Technology Fail-Safe Mode*.

The PM_CPUx_PWRSTCTRL register is static over any power transition. That is, software programs it before executing the WFI instruction and does not change it until MPU_Cx is again in running mode. In other words, when MPU_Cx reaches a low-power state, it cannot move to another low-power state. It must be woken up to reach another low-power state. To wake up MPU_Cx, the user must:

1. Execute a forced wake-up transition to the MPU_Cx: CM_CPUx_CLKSTCTRL[1:0] CLKTRCTRL = 0x2.
2. The MPU_Cx interrupt handler must write back the automatic hardware transition: CM_CPUx_CLKSTCTRL[1:0] CLKTRCTRL = 0x3.

Table 4-7 shows the available MPU_Cx power states in single and coherency mode. For coherency, software must ensure that both MPU cores are in the same power state.

Table 4-7. Available MPU_Cx Power States in Single and Coherency Mode

MPU_C0 Power State	MPU_C1 Power State	Mode
Running/ON or WFI/ON	Running/ON or WFI/ON	SMP mode (coherent mode)
WFI/INACT	WFI/INACT	
WFI/CSWRET	WFI/CSWRET	
Running/ON or WFI/ON	FORCED_OFF	Single mode (MPU_C1 is out of coherency)
WFI/INACT	FORCED_OFF	
WFI/CSWRET	FORCED_OFF	

4.3.7.3 Power States of MPU Subsystem

The MPU subsystem power domain (PD_MPU) must be at a higher or equal power state (a state that consumes more power) than the higher of the two MPU cores. For example, it is illegal for the MPU subsystem power state to be RETENTION, while the power state of one or both of the MPU cores is ON.

Software must ensure that only legal power states are programmed. When an illegal state is entered, the behavior of the hardware is unpredictable.

Table 4-8 lists the MPU subsystem legal power states.

Table 4-8. MPU Subsystem Legal Power States

Hardware Conditions		MPU/System Programming Model				Resulting MPU/System State			
MPU_Cx States	State of L2	PRCM Power State PM_MPU_PWRSTCTRL[1:0] POWERSTATE	PRCM Logic Retention State PM_MPU_PWRSTCTRL[2] LOGICRETSTATE	PRCM L2 Memory Retention State PM_MPU_PWRSTCTRL[9] MPU_L2_RETSTATE	PRCM Clock Transition Control CM_MPU_CLKSTCTRL[1:0] CLKTRCTRL	Logic	L2 Cache	DPLL Clock	Power State (at PRCM)
At least one is ON	Any	Any	Any	Any	Any	ON	ON	ON	ON
Any	!= (IDLE / WFI)	Any	Any	Any	Any	ON	ON	ON	ON
Power state of both MPU cores is less than or equal to INACT	IDLE / WFI	Any	Any	Any	NO_SLEEP/ SW_WKUP	ON	ON	ON	ON
	IDLE / WFI	ON	Any	Any	HW_AUTO	ON	ON	ON	ON
	IDLE / WFI	INACT	Any	Any	HW_AUTO	ON	ON	OFF	INACT
Power state of both MPU cores is less than or equal to CSWRET	IDLE / WFI	RETENTION	CSWRET	RETENTION	HW_AUTO	ON	RETENTION	OFF	CSWRET
	IDLE / WFI	RETENTION	CSWRET	OFF	HW_AUTO	ON	OFF	OFF	CSWRET

Note

All the transitions presented in Table 4-8, except for the first line, are handled by the MPU_PRCM and the global PRCM.

Once the MPU subsystem reaches a low-power state (INACT, CSWRET), it cannot move to another low-power state. It must be woken up to reach another low-power state.

4.3.7.4 MPU_WUGEN

The MPU_WUGEN belongs to the MPU always-on power domain (PD_MPUAON) and is responsible for generating wake-up events from the incoming interrupts (external and local to the MPU subsystem) according to the MPU_WUGEN 160-bit enable field (from WKG_ENB_A_x to WKG_ENB_E_x for MPU_Cx, where x = 0 or 1), which defines the interrupt that wakes up the MPU cores.

All interrupts are enabled after reset, except MPU_IRQ_8. The Cortex-A15 MPU can access the MPU_WUGEN internal configuration registers through the MPU_AXI2OCP.

Software must program interrupt enabling and disabling coherently in the MPU_INTC and in the MPU_WUGEN enable registers. That is, a given interrupt for a given MPU core is either enabled at both MPU_INTC and MPU_WUGEN, or disabled at both; no combination is allowed.

MPU_WUGEN includes two registers (AUX_CORE_BOOT_0 and AUX_CORE_BOOT_1) which can be used by the ROM code and OS during SMP boot. The AUX_CORE_BOOT_0 register is intended to indicate boot status to MPU_C1, the second core (also known as aux core); the AUX_CORE_BOOT_1 register can be used to store execution start address of the MPU_C1. For more information, see Section 32.3.4, *Startup and Configuration*, in Chapter 32, *Initialization*.

Figure 4-10 is a functional overview of the MPU_WUGEN in the MPU subsystem.

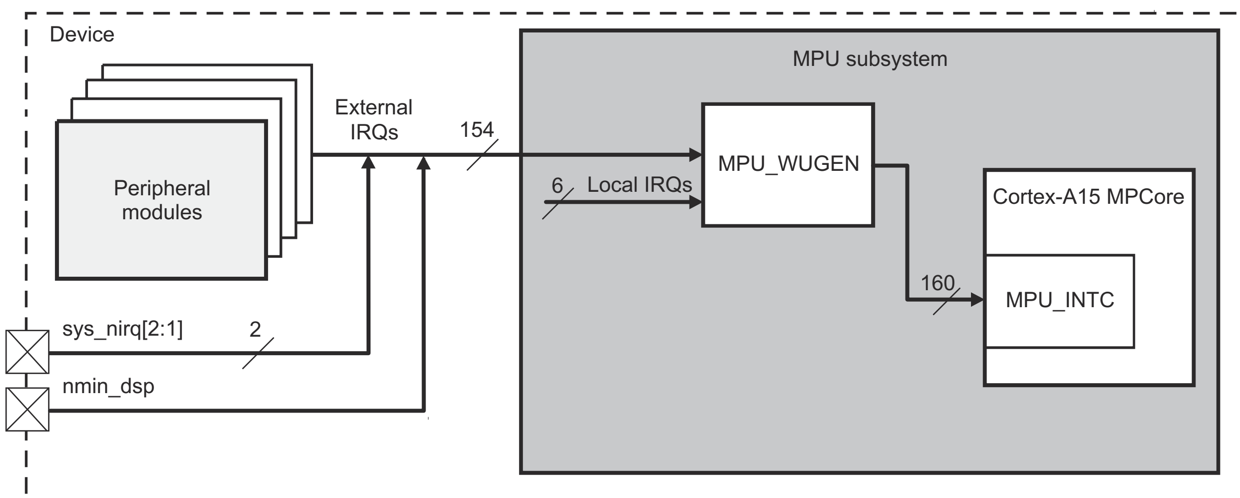


Figure 4-10. MPU_WUGEN Overview

4.3.7.5 Power Transition Sequence

At power-on-reset (POR), or global warm reset, both MPU cores are powered on. Software can determine the appropriate power mode for both MPU cores and program the MPU_PRCM accordingly.

There are three types of power transitions:

- Power transitions that do not involve the local or global PRCM module

- MPU_Cx transition to WFE/ON - corresponds to line 2 of Table 4-6
- MPU_Cx transition to WFI/ON - corresponds to lines 3, 4, and 6 of Table 4-6

- Power transitions handled by the local PRCM module
  - MPU_C1 transition to WFI/FORCED_OFF - corresponds to line 5 of Table 4-6. Requires software to program MPU_PRCM with FORCED_OFF mode.
  - MPU_Cx transition to WFI/INACT, or WFI/RET - corresponds to line 7, 8 and 9 of Table 4-6. Requires software to program the MPU_PRCM with INACT/RET mode.
- Power transitions handled by the local and global PRCM modules

4.3.7.6 SR3-APG Technology Fail-Safe Mode

The SR3-APG power-management technology implements a fast power ramp-up technology. To take advantage of the fast ramp-up feature in SR3-APG, software must first enable it by setting the PRM_PSCON_COUNT[25] HG_RAMPUP bit to 1.

A fail-safe mechanism is put in place to revert back to the standard power ramp-up time by setting the PRM_PSCON_COUNT[25] HG_RAMPUP bit to 0.

The slow ramp-up time can be set through the PRM_PSCON_COUNT[23:16] HG_PONOUT_2_PGDOODIN_TIME bit field when the HG weak chain is used (in other words, the PRM_PSCON_COUNT[25] HG_RAMPUP bit is set to 0x0).

This applies only when SR3-APG is enabled (PRM_PSCON_COUNT[24] HG_EN = 1).

The L1 cache memory state depends on the values of the PRM_PSCON_COUNT[25] HG_RAMPUP and PRM_PSCON_COUNT[24] HG_EN bits. When CPU PD is in CSWRET state and HG_ENABLE and HG_RAMPUP are set to 0x1, L1 cache is in the ON state.

If SR3-APG is disabled (PRM_PSCON_COUNT[24] HG_EN = 0), the L1 cache array can be put in RETENTION during CSWRET regardless of the PRM_PSCON_COUNT[25] HG_RAMPUP bit.

4.3.8 MPU Subsystem AMBA Interface Configuration

There are some inputs at the MPU core (MPU_Cx) boundary which can be tied off to disable certain kind of transactions appearing on the AMBA4 interface:

- SYSBARDISABLE (SBD): This controls whether:
  - Barriers are handled internal to MPU core (SBD = 1), or
  - Barriers are issued on the AMBA4 interface (SBD = 0).
- BROADCASTINNER (BI), BROADCASTOUTER (BO), BROADDCASTMAINTENANCE (BCM): These control whether external snoops are issued on AMBA4 interface or not:
  - BI=BO=BCM=0: No external snoops issued.
  - BI=BO=BCM=1: External snoops can be issued. Not supported in the device.

For detailed description on these inputs, see the Arm *Cortex-A15 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

To give flexibility and mitigate risk, a programmable register (AMBA_IF_MODE) is added to control the tie-off value of these MPU_Cx inputs. Once this register is programmed, MPU power domain has to go through OSWRET transition for the programmed values in bits AMBA_IF_MODE [3:0] to take effect. This is because the MPU_Cx non-cpu logic latches the value on the corresponding inputs when it is coming out of reset.

These are the only legal combinations of [BI, BO, BCM, SBD] allowed:

- BI, BO, BCM, SBD = 0001 – Corresponds to **AXI3** mode. Barriers not issued on AMBA4 interface. External snoops not issued. This is the default mode.
- BI, BO, BCM, SBD = 0000 – Corresponds to **ACE non-coherent, no L3** mode. Barriers issued on AMBA4 interface. External snoops not issued.

The ROM code provides specific API for configuring the AMBA_IF_MODE register. For more information, see section *Wakeup Generator*, in Chapter 32, *Initialization*.

4.4 Dual Cortex-A15 MPU Subsystem Register Manual

4.4.1 Dual Cortex-A15 MPU Subsystem Instance Summary

Table 4-9. Dual Cortex-A15 MPU Subsystem Instance Summary

Module Name	Base Address	Size
MPU_CS_STM	0x4700 0000	16 MiB
MPU_INTC	0x4821 0000	32 KiB
MPU_PRCM_OCP_SOCKET	0x4824 3000	512 bytes
MPU_PRCM_DEVICE	0x4824 3200	512 bytes
MPU_PRCM_PRM_C0	0x4824 3400	512 bytes
MPU_PRCM_CM_C0	0x4824 3600	512 bytes
MPU_PRCM_PRM_C1	0x4824 3800	512 bytes
MPU_PRCM_CM_C1	0x4824 3A00	512 bytes
MPU_WUGEN	0x4828 1000	4 KiB
Reserved	0x4829 0000	12 KiB
MPU_WD_TIMER	0x482A 0000	4 KiB
MPU_AXI2OCP_MISC	0x482A 2000	4 KiB
MPU_MA_LSM	0x482A F000	256 bytes
MPU_MA_WP	0x482A F200	256 bytes

4.4.2 MPU_CS_STM Registers

For information about the MPU_CS_STM registers and their description, see the Arm *CoreSight STM Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

4.4.3 MPU_INTC Registers

For information about the MPU_INTC registers and their description, see the Arm *Cortex-A15 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

4.4.4 MPU_PRCM_OCP_SOCKET Registers

4.4.4.1 MPU_PRCM_OCP_SOCKET Register Summary

Table 4-10. Local PRCM Revision Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRCM_OCP_SOCKET Physical Address
REVISION_PRCM_MPU	R	32	0x0000 0000	0x4824 3000

4.4.4.2 MPU_PRCM_OCP_SOCKET Register Description

Table 4-11. REVISION_PRCM_MPU

Address Offset	0x0000 0000		
Physical Address	0x4824 3000	**Instance**	MPU_PRCM_OCP_SOCKET
Description	IP Revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 4-12. Register Call Summary for Register REVISION_PRCM_MPU

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_OCP_SOCKET Register Summary: [0]

4.4.5 MPU_PRCM_DEVICE Registers

4.4.5.1 MPU_PRCM_DEVICE Register Summary

Table 4-13. MPU_PRCM_DEVICE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRCM_DEVICE Physical Address
PRM_RSTST	RW	32	0x0000 0000	0x4824 3200
PRM_PSCON_COUNT	RW	32	0x0000 0004	0x4824 3204
PRM_FRAC_INCREMENTER_NUMERATOR	RW	32	0x0000 0010	0x4824 3210
PRM_FRAC_INCREMENTER_DENUMERATOR_RELOAD	RW	32	0x0000 0014	0x4824 3214

4.4.5.2 MPU_PRCM_DEVICE Register Description

Table 4-14. PRM_RSTST

Address Offset	0x0000 0000		
Physical Address	0x4824 3200	**Instance**	MPU_PRCM_DEVICE
Description	This register logs the global reset sources, thus contains information regarding the cold/warm reset events generated by global PRCM. Each bit is set upon release of the domain reset signal. Must be cleared by software. This register is insensitive to global warm reset.		
Type	RW		

31–2	1	0
RESERVED	GLOBAL_WARM_RST	GLOBAL_COLD_RST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	GLOBAL_WARM_RST	Global warm reset event generated by global PRCM 0x0: No global warm reset. 0x1: Global external warm reset has occurred.	RW (W1toClr)	0x0
0	GLOBAL_COLD_RST	Power-on (cold) reset event generated by global PRCM 0x0: No power-on reset. 0x1: Power-on reset has occurred.	RW (W1toClr)	0x1

Table 4-15. PRM_PSCON_COUNT

Address Offset	0x0000 0004		
Physical Address	0x4824 3204	**Instance**	MPU_PRCM_DEVICE
Description	Programmable precharge count for L1cache		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	HG_RAMPUP	HG_EN	HG_PONOUT_2_PGDOODIN_TIME	RESERVED	PCHARGE_TIME

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x00
25	HG_RAMPUP	Ramp-up mode selection of HG power chain switch 0x0: Slow ramp-up mode – HG weak chain is used. The delay between PONOUTHG and PGOODINHG is defined by the HG_PONOUT_2_PGDOODIN_TIME bit field. 0x1: Fast ramp-up mode – HG weak chain is not used	RW	0x0
24	HG_EN	HG power chain switch enable 0x0: HG power chain switch is disabled 0x1: HG power chain switch is enabled	RW	0x0
23:16	HG_PONOUT_2_PGDOODIN_TIME	The value set in this field determines the slow ramp-up time and the duration (number of cycles) of the PONOUTHG to PGOODINHG (transition for power domain without DPS). The duration is computed as 8 x HG_PONOUT_2_PGDOODIN_TIME of system clock cycles. Target is 10us.	RW	0x30
15:8	RESERVED	Reserved	R	0x00
7:0	PCHARGE_TIME	Programmable precharge count during retention	RW	0x17

Table 4-16. PRM_FRAC_INCREMENTER_NUMERATOR

Address Offset	0x0000 0010		
Physical Address	0x4824 3210	**Instance**	MPU_PRCM_DEVICE
Description	Fractional incrementor		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="12">ABE_LP_MODE_NUMERATOR</td><td colspan="4">RESERVED</td><td colspan="12">SYS_MODE_NUMERATOR</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	R	0x0
27:16	ABE_LP_MODE_NUMERATOR	Numerator to be used in fractional incrementor when ABE_LP_CLK clock is used as PRCM clock. Reset value corresponds to ABE_LP_CLK clock = 12.288 MHz.	RW	0x659
15:12	RESERVED	Reserved	R	0x0
11:0	SYS_MODE_NUMERATOR	Numerator to be used in fractional incrementor when SYS_CLK1 is used as PRCM clock. **NOTE**: The reset value corresponds to SYS_CLK1 = 38.4 MHz. Because the device does not support such SYS_CLK1 frequency, it is SW responsibility to set a value corresponding to one of the SYS_CLK1 frequencies listed in Table 4-4.	RW	0x208

Table 4-17. PRM_FRAC_INCREMENTER_DENUMERATOR_RELOAD

Address Offset	0x0000 0014		
Physical Address	0x4824 3214	**Instance**	MPU_PRCM_DEVICE
Description	Reload command and denominator to be used in fractional incrementor		

Table 4-17. PRM_FRAC_INCREMENTER_DENUMERATOR_RELOAD (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															RELOAD	RESERVED				DENOMINATOR											

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000
16	RELOAD	Reload counter value from coarse counter. 0->1 transition in this field is used to load the coarse counter into counter.	RW	0x0
15:12	RESERVED	Reserved	R	0x0
11:0	DENOMINATOR	Denominator to be used in fractional incrementor when when SYS_CLK1 is used as PRCM clock. **NOTE**: The reset value corresponds to SYS_CLK1 = 38.4 MHz. Because the device does not support such SYS_CLK1 frequency, it is SW responsibility to set a value corresponding to one of the SYS_CLK1 frequencies listed in Table 4-4.	RW	0xCB2

4.4.6 MPU_PRCM_PRM_C0 Registers

4.4.6.1 MPU_PRCM_PRM_C0 Register Summary

Table 4-18. MPU_PRCM_PRM_C0 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRCM_PRM_C0 Physical Address
PM_CPU0_PWRSTCTRL	RW	32	0x0000 0000	0x4824 3400
PM_CPU0_PWRSTST	RW	32	0x0000 0004	0x4824 3404
RM_CPU0_CPU0_RSTCTRL	RW	32	0x0000 0010	0x4824 3410
RM_CPU0_CPU0_RSTST	RW	32	0x0000 0014	0x4824 3414
RM_CPU0_CPU0_CONTEXT	RW	32	0x0000 0024	0x4824 3424

4.4.6.2 MPU_PRCM_PRM_C0 Register Description

Table 4-19. PM_CPU0_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4824 3400	**Instance**	MPU_PRCM_PRM_C0
Description	This register controls the CPU domain power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														L1_BANK_ONSTATE		RESERVED							L1_BANK_RETSTATE	RESERVED					LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:16	L1_BANK_ONSTATE	CPU_L1 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R	0x3

Bits	Field Name	Description	Type	Reset
15:9	RESERVED	Reserved	R	0x00
8	L1_BANK_RETSTATE	CPU_L1 memory state when domain is RETENTION. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
7:3	RESERVED	Reserved	R	0x00
2	LOGICRETSTATE	Logic state when power domain is RETENTION Read 0x1: Whole logic is retained when domain is in RETENTION state.	R	0x1
1:0	POWERSTATE	Power state control 0x0: OFF state **NOTE: OFF state for MPU_C0 is NOT supported in this device.** 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON state	RW	0x3

Table 4-20. Register Call Summary for Register PM_CPU0_PWRSTCTRL

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_PRM_C0 Register Summary: [0]

Table 4-21. PM_CPU0_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4824 3404	**Instance**	MPU_PRCM_PRM_C0
Description	This register provides a status on the CPU domain current power state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	L1_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered 0x0: Power domain was previously in OFF **NOTE: OFF state for MPU_C0 is NOT supported in this device.** 0x1: Power domain was previously RETENTION 0x2: Power domain was previously INACTIVE 0x3: Power domain was previously ON	RW	0x0
23:21	RESERVED	Reserved	R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED	Reserved	R	0x0000

Bits	Field Name	Description	Type	Reset
5:4	L1_BANK_STATEST	CPU_L1 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RET Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED	Reserved	R	0x0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF **NOTE: OFF state for MPU_C0 is NOT supported in this device.** Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

Table 4-22. Register Call Summary for Register PM_CPU0_PWRSTST

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_PRM_C0 Register Summary: [0]

Table 4-23. RM_CPU0_CPU0_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4824 3410	**Instance**	MPU_PRCM_PRM_C0
Description	This register controls the assertion/release of the CPU CORE reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RST	CPU warm local reset control 0x0: Reset is cleared 0x1: Reset is asserted	RW	0x0

Table 4-24. Register Call Summary for Register RM_CPU0_CPU0_RSTCTRL

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_PRM_C0 Register Summary: [0]

Table 4-25. RM_CPU0_CPU0_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4824 3414	**Instance**	MPU_PRCM_PRM_C0
Description	This register logs the different reset sources of the MPU domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DBGRST_REQ_RSTST	RSTST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	DBGRST_REQ_RSTST	MPU_C0 processor has been reset due to MPU_C0 emulation reset request driven from MPUSS. Read 0x0: No emulation reset Read 0x1: MPU_C0 has been reset upon emulation request.	RW (W1toClr)	0x0
0	RSTST	MPU_C0 software reset Read 0x0: No software reset occurred. Read 0x1: MPU_C0 has been reset upon software reset.	RW (W1toClr)	0x0

Table 4-26. Register Call Summary for Register RM_CPU0_CPU0_RSTST

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_PRM_C0 Register Summary: [0]

Table 4-27. RM_CPU0_CPU0_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4824 3424	**Instance**	MPU_PRCM_PRM_C0
Description	This register contains dedicated CPU context statuses. [warm reset insensitive]		
Type	RW		

31–9	8	7–1	0
RESERVED	LOSTMEM_CPU_L1	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x00 0000
8	LOSTMEM_CPU_L1	Specify if memory-based context in CPU_L1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW (W1toClr)	0x1
7:1	RESERVED	Reserved	R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW (W1toClr)	0x1

Table 4-28. Register Call Summary for Register RM_CPU0_CPU0_CONTEXT

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_PRM_C0 Register Summary: [0]

4.4.7 MPU_PRCM_CM_C0 Registers

4.4.7.1 MPU_PRCM_CM_C0 Register Summary

Table 4-29. MPU_PRCM_CM_C0 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRCM_CM_C0 Physical Address
CM_CPU0_CLKSTCTRL	RW	32	0x0000 0000	0x4824 3600
CM_CPU0_CPU0_CLKCTRL	R	32	0x0000 0020	0x4824 3620

4.4.7.2 MPU_PRCM_CM_C0 Register Description

Table 4-30. CM_CPU0_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4824 3600	**Instance**	MPU_PRCM_CM_C0
Description	This register enables the CPU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
Type	RW		

31:2	1:0
RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	CLKTRCTRL	Controls the full domain transition of the CPU domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may, however, occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wakeup transition on the domain.0x1: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 4-31. Register Call Summary for Register CM_CPU0_CLKSTCTRL

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_CM_C0 Register Summary: [0]

Table 4-32. CM_CPU0_CPU0_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4824 3620	**Instance**	MPU_PRCM_CM_C0
Description	This register manages the CPU clocks.		
Type	R		

31:1	0
RESERVED	STBYST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby). Read 0x1: Module is in standby.	R	0x1

Table 4-33. Register Call Summary for Register CM_CPU0_CPU0_CLKCTRL

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_CM_C0 Register Summary: [0]

4.4.8 MPU_PRCM_PRM_C1 Registers

4.4.8.1 MPU_PRCM_PRM_C1 Register Summary

Table 4-34. MPU_PRCM_PRM_C1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRCM_PRM_C1 Physical Address
PM_CPU1_PWRSTCTRL	RW	32	0x0000 0000	0x4824 3800
PM_CPU1_PWRSTST	RW	32	0x0000 0004	0x4824 3804
RM_CPU1_CPU1_RSTCTRL	RW	32	0x0000 0010	0x4824 3810
RM_CPU1_CPU1_RSTST	RW	32	0x0000 0014	0x4824 3814
RM_CPU1_CPU1_CONTEXT	RW	32	0x0000 0024	0x4824 3824

4.4.8.2 MPU_PRCM_PRM_C1 Register Description

Table 4-35. PM_CPU1_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4824 3800	**Instance**	MPU_PRCM_PRM_C1
Description	This register controls the CPU domain power state to reach upon a domain sleep transition		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="2">L1_BANK_ONSTATE</td><td colspan="7">RESERVED</td><td>L1_BANK_RETSTATE</td><td>FORCED_OFF</td><td colspan="4">RESERVED</td><td>LOGICRETSTATE</td><td colspan="2">POWERSTATE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:16	L1_BANK_ONSTATE	CPU_L1 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R	0x3
15:9	RESERVED	Reserved	R	0x00
8	L1_BANK_RETSTATE	CPU L1 memory state when domain is RETENTION. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
7	FORCED_OFF	Selects if logic must be forced in OFF state. 0x0: Whole logic is not forced in OFF state. 0x1: Whole logic is forced in OFF state.	RW	0x0
6:3	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
2	LOGICRETSTATE	Logic state when power domain is RETENTION Read 0x1: Whole logic is retained when domain is in RETENTION state.	R	0x1
1:0	POWERSTATE	Power state control 0x0: OFF state **NOTE: Must always be followed by FORCED_OFF = 0x1.** 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON state	RW	0x3

Table 4-36. PM_CPU1_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4824 3804	**Instance**	MPU_PRCM_PRM_C1
Description	This register provides a status on the CPU domain current power state. [warm reset insensitive]		
Type	RW		

31:26	25:24	23:21	20	19:6	5:4	3	2	1:0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	L1_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x00
25:24	LASTPOWERSTATEENTERED	Last low-power state entered 0x0: Power domain was previously FORCED_OFF. 0x1: Power domain was previously in RETENTION. 0x2: Power domain previously was INACTIVE. 0x3: Power domain was previously ON.	RW	0x0
23:21	RESERVED	Reserved	R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED	Reserved	R	0x0000
5:4	L1_BANK_STATEST	CPU_L1 memory state status Read 0x0: Memory is OFF. Read 0x1: Memory is RET. Read 0x2: Reserved Read 0x3: Memory is ON.	R	0x3
3	RESERVED	Reserved	R	0x0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF. Read 0x1: Logic in domain is ON.	R	0x1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is FORCED_OFF. Read 0x1: Power domain is in RETENTION. Read 0x2: Power domain is ON-INACTIVE. Read 0x3: Power domain is ON-ACTIVE.	R	0x3

Table 4-37. RM_CPU1_CPU1_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4824 3810	**Instance**	MPU_PRCM_PRM_C1
Description	This register controls the assertion/release of the CPU CORE reset.		
Type	RW		

31–1	0
RESERVED	RST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RST	CPU warm local reset control 0x0: Reset is cleared. 0x1: Reset is asserted.	RW	0x0

Table 4-38. RM_CPU1_CPU1_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4824 3814	**Instance**	MPU_PRCM_PRM_C1
Description	This register logs the different reset sources of the MPU domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31–2	1	0
RESERVED	DBGRST_REQ_RSTST	RSTST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	DBGRST_REQ_RSTST	MPU_C1 processor has been reset due to MPU_C0 emulation reset request driven from MPUSS Read 0x0: No emulation reset. Read 0x1: MPU_C1 has been reset upon emulation request.	RW (W1toClr)	0x0
0	RSTST	MPU_C1 software reset Read 0x0: No software reset occurred. Read 0x1: MPU_C1 has been reset upon software reset.	RW (W1toClr)	0x0

Table 4-39. RM_CPU1_CPU1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4824 3824	**Instance**	MPU_PRCM_PRM_C1
Description	This register contains dedicated CPU context statuses. [warm reset insensitive]		
Type	RW		

31:9	8	7:1	0
RESERVED	LOSTMEM_CPU_L1	RESERVED	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x00 0000
8	LOSTMEM_CPU_L1	Specify if memory-based context in CPU_L1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained. 0x1: Context has been lost.	RW (W1toClr)	0x1
7:1	RESERVED	Reserved	R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained. 0x1: Context has been lost.	RW (W1toClr)	0x1

4.4.9 MPU_PRCM_CM_C1 Registers

4.4.9.1 MPU_PRCM_CM_C1 Register Summary

Table 4-40. MPU_PRCM_CM_C1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRCM_CM_C1 Physical Address
CM_CPU1_CLKSTCTRL	RW	32	0x0000 0000	0x4824 3A00
CM_CPU1_CPU1_CLKCTRL	R	32	0x0000 0020	0x4824 3A20

4.4.9.2 MPU_PRCM_CM_C1 Register Description

Table 4-41. CM_CPU1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4824 3A00	**Instance**	MPU_PRCM_CM_C1
Description	This register enables the MPU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
Type	RW		

31:2	1:0
RESERVED	CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the full domain transition of the CPU domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may, however, occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wake-up transition are based upon hardware conditions.	RW	0x0

Table 4-42. Register Call Summary for Register CM_CPU1_CLKSTCTRL

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_CM_C1 Register Summary: [0]

Table 4-43. CM_CPU1_CPU1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4824 3A20	**Instance**	MPU_PRCM_CM_C1
Description	This register manages the MPU clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STBYST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby). Read 0x1: Module is in standby.	R	0x1

Table 4-44. Register Call Summary for Register CM_CPU1_CPU1_CLKCTRL

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_PRCM_CM_C1 Register Summary: [0]

4.4.10 MPU_WUGEN Registers

4.4.10.1 MPU_WUGEN Register Summary

Table 4-45. MPU_WUGEN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_WUGEN Physical Address
WKG_CONTROL_0	R	32	0x0000 0000	0x4828 1000
WKG_ENB_A_0	RW	32	0x0000 0010	0x4828 1010
WKG_ENB_B_0	RW	32	0x0000 0014	0x4828 1014
WKG_ENB_C_0	RW	32	0x0000 0018	0x4828 1018
WKG_ENB_D_0	RW	32	0x0000 001C	0x4828 101C
WKG_ENB_E_0	RW	32	0x0000 0020	0x4828 1020
RESERVED	R	32	0x0000 0024	0x4828 1024
RESERVED	R	32	0x0000 0028	0x4828 1028
RESERVED	R	32	0x0000 002C	0x4828 102C
RESERVED	R	32	0x0000 0030	0x4828 1030
RESERVED	R	32	0x0000 0034	0x4828 1034

Table 4-45. MPU_WUGEN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MPU_WUGEN Physical Address
WKG_CONTROL_1	R	32	0x0000 0400	0x4828 1400
WKG_ENB_A_1	RW	32	0x0000 0410	0x4828 1410
WKG_ENB_B_1	RW	32	0x0000 0414	0x4828 1414
WKG_ENB_C_1	RW	32	0x0000 0418	0x4828 1418
WKG_ENB_D_1	RW	32	0x0000 041C	0x4828 141C
WKG_ENB_E_1	RW	32	0x0000 0420	0x4828 1420
RESERVED	R	32	0x0000 0424	0x4828 1424
RESERVED	R	32	0x0000 0428	0x4828 1428
RESERVED	R	32	0x0000 042C	0x4828 142C
RESERVED	R	32	0x0000 0430	0x4828 1430
RESERVED	R	32	0x0000 0434	0x4828 1434
AUX_CORE_BOOT_0	RW	32	0x0000 0800	0x4828 1800
AUX_CORE_BOOT_1	RW	32	0x0000 0804	0x4828 1804
STM_HWEVENTS_INV	RW	32	0x0000 0808	0x4828 1808
AMBA_IF_MODE	RW	32	0x0000 080C	0x4828 180C
RESERVED	R	32	0x0000 0C00	0x4828 1C00
RESERVED	R	32	0x0000 0C04	0x4828 1C04
TIMESTAMPCYCLELO	R	32	0x0000 0C08	0x4828 1C08
TIMESTAMPCYCLEHI	R	32	0x0000 0C0C	0x4828 1C0C

4.4.10.2 MPU_WUGEN Register Description

Table 4-46. WKG_CONTROL_0

Address Offset	0x0000 0000		
Physical Address	0x4828 1000	**Instance**	MPU_WUGEN
Description	Wake-up generator control and status register for MPU_C0		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td>DOMAINRESET</td><td>MPU_WARM_RESET</td><td>MPU_COLD_RESET</td><td colspan="2">RESERVED</td><td>EVENTO</td><td>STANDBYWFE</td><td>STANDBYWFI</td><td colspan="8">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x00000
15	DOMAINRESET	MPU always-on power domain (PD_MPUAON) reset status bit. It shows if the reset occurred previously. 0x0: no reset occur 0x1: reset occur	R	0x0
14	MPU_WARM_RESET	This bit is set when the MPU_WARM_RESET signal is asserted. 0x0: MPU_WARM_RESET reset signal has not been asserted 0x1: MPU_WARM_RESET reset request has been asserted	RW	0x0

Bits	Field Name	Description	Type	Reset
13	MPU_COLD_RESET	This bit is set when the MPU_COLD_RESET signal is asserted. 0x0: MPU_COLD_RESET reset signal has not been asserted 0x1: MPU_COLD_RESET reset request has been asserted	RW	0x0
12:11	RESERVED	Reserved	R	0x0
10	EVENTO	EVENTO status bit. The event output signal is active, when one SEV instruction is executed. This bit is set when a rising edge of EVENTO from CPU is detected. 0x0: Rising edge of EVENTO is not detected 0x1: Rising edge of EVENTO is detected	RW	0x0
9	STANDBYWFE	This bit gives software the visibility to track whether WFE mode have been entered. 0x0: WFE mode has not been entered 0x1: WFE mode has been entered	RW	0x0
8	STANDBYWFI	This bit gives software the visibility to track whether WFI mode have been entered. 0x0: WFI mode has not been entered 0x1: WFI mode has been entered	RW	0x0
7:0	RESERVED	Reserved	R	0x0

Table 4-47. WKG_ENB_A_0

Address Offset	0x0000 0010		
Physical Address	0x4828 1010	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C0 (interrupts MPU_IRQ_0 to MPU_IRQ_31). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR31	WKG_ENB_FOR_INTR30	WKG_ENB_FOR_INTR29	WKG_ENB_FOR_INTR28	WKG_ENB_FOR_INTR27	WKG_ENB_FOR_INTR26	WKG_ENB_FOR_INTR25	WKG_ENB_FOR_INTR24	WKG_ENB_FOR_INTR23	WKG_ENB_FOR_INTR22	WKG_ENB_FOR_INTR21	WKG_ENB_FOR_INTR20	WKG_ENB_FOR_INTR19	WKG_ENB_FOR_INTR18	WKG_ENB_FOR_INTR17	WKG_ENB_FOR_INTR16	WKG_ENB_FOR_INTR15	WKG_ENB_FOR_INTR14	WKG_ENB_FOR_INTR13	WKG_ENB_FOR_INTR12	WKG_ENB_FOR_INTR11	WKG_ENB_FOR_INTR10	WKG_ENB_FOR_INTR9	WKG_ENB_FOR_INTR8	WKG_ENB_FOR_INTR7	WKG_ENB_FOR_INTR6	WKG_ENB_FOR_INTR5	WKG_ENB_FOR_INTR4	WKG_ENB_FOR_INTR3	WKG_ENB_FOR_INTR2	WKG_ENB_FOR_INTR1	WKG_ENB_FOR_INTR0

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR31	Wakeup enable for interrupt line MPU_IRQ_31	RW	0x1
30	WKG_ENB_FOR_INTR30	Wakeup enable for interrupt line MPU_IRQ_30	RW	0x1
29	WKG_ENB_FOR_INTR29	Wakeup enable for interrupt line MPU_IRQ_29	RW	0x1
28	WKG_ENB_FOR_INTR28	Wakeup enable for interrupt line MPU_IRQ_28	RW	0x1
27	WKG_ENB_FOR_INTR27	Wakeup enable for interrupt line MPU_IRQ_27	RW	0x1
26	WKG_ENB_FOR_INTR26	Wakeup enable for interrupt line MPU_IRQ_26	RW	0x1
25	WKG_ENB_FOR_INTR25	Wakeup enable for interrupt line MPU_IRQ_25	RW	0x1
24	WKG_ENB_FOR_INTR24	Wakeup enable for interrupt line MPU_IRQ_24	RW	0x1
23	WKG_ENB_FOR_INTR23	Wakeup enable for interrupt line MPU_IRQ_23	RW	0x1
22	WKG_ENB_FOR_INTR22	Wakeup enable for interrupt line MPU_IRQ_22	RW	0x1
21	WKG_ENB_FOR_INTR21	Wakeup enable for interrupt line MPU_IRQ_21	RW	0x1
20	WKG_ENB_FOR_INTR20	Wakeup enable for interrupt line MPU_IRQ_20	RW	0x1

Bits	Field Name	Description	Type	Reset
19	WKG_ENB_FOR_INTR19	Wakeup enable for interrupt line MPU_IRQ_19	RW	0x1
18	WKG_ENB_FOR_INTR18	Wakeup enable for interrupt line MPU_IRQ_18	RW	0x1
17	WKG_ENB_FOR_INTR17	Wakeup enable for interrupt line MPU_IRQ_17	RW	0x1
16	WKG_ENB_FOR_INTR16	Wakeup enable for interrupt line MPU_IRQ_16	RW	0x1
15	WKG_ENB_FOR_INTR15	Wakeup enable for interrupt line MPU_IRQ_15	RW	0x1
14	WKG_ENB_FOR_INTR14	Wakeup enable for interrupt line MPU_IRQ_14	RW	0x1
13	WKG_ENB_FOR_INTR13	Wakeup enable for interrupt line MPU_IRQ_13	RW	0x1
12	WKG_ENB_FOR_INTR12	Wakeup enable for interrupt line MPU_IRQ_12	RW	0x1
11	WKG_ENB_FOR_INTR11	Wakeup enable for interrupt line MPU_IRQ_11	RW	0x1
10	WKG_ENB_FOR_INTR10	Wakeup enable for interrupt line MPU_IRQ_10	RW	0x1
9	WKG_ENB_FOR_INTR9	Wakeup enable for interrupt line MPU_IRQ_9	RW	0x1
8	WKG_ENB_FOR_INTR8	Wakeup enable for interrupt line MPU_IRQ_8	RW	0x0
7	WKG_ENB_FOR_INTR7	Wakeup enable for interrupt line MPU_IRQ_7	RW	0x1
6	WKG_ENB_FOR_INTR6	Wakeup enable for interrupt line MPU_IRQ_6	RW	0x1
5	WKG_ENB_FOR_INTR5	Wakeup enable for interrupt line MPU_IRQ_5	RW	0x1
4	WKG_ENB_FOR_INTR4	Wakeup enable for interrupt line MPU_IRQ_4	RW	0x1
3	WKG_ENB_FOR_INTR3	Wakeup enable for interrupt line MPU_IRQ_3	RW	0x1
2	WKG_ENB_FOR_INTR2	Wakeup enable for interrupt line MPU_IRQ_2	RW	0x1
1	WKG_ENB_FOR_INTR1	Wakeup enable for interrupt line MPU_IRQ_1	RW	0x1
0	WKG_ENB_FOR_INTR0	Wakeup enable for interrupt line MPU_IRQ_0	RW	0x1

Table 4-48. WKG_ENB_B_0

Address Offset	0x0000 0014		
Physical Address	0x4828 1014	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C0 (interrupts MPU_IRQ_32 to MPU_IRQ_63). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR63	WKG_ENB_FOR_INTR62	WKG_ENB_FOR_INTR61	WKG_ENB_FOR_INTR60	WKG_ENB_FOR_INTR59	WKG_ENB_FOR_INTR58	WKG_ENB_FOR_INTR57	WKG_ENB_FOR_INTR56	WKG_ENB_FOR_INTR55	WKG_ENB_FOR_INTR54	WKG_ENB_FOR_INTR53	WKG_ENB_FOR_INTR52	WKG_ENB_FOR_INTR51	WKG_ENB_FOR_INTR50	WKG_ENB_FOR_INTR49	WKG_ENB_FOR_INTR48	WKG_ENB_FOR_INTR47	WKG_ENB_FOR_INTR46	WKG_ENB_FOR_INTR45	WKG_ENB_FOR_INTR44	WKG_ENB_FOR_INTR43	WKG_ENB_FOR_INTR42	WKG_ENB_FOR_INTR41	WKG_ENB_FOR_INTR40	WKG_ENB_FOR_INTR39	WKG_ENB_FOR_INTR38	WKG_ENB_FOR_INTR37	WKG_ENB_FOR_INTR36	WKG_ENB_FOR_INTR35	WKG_ENB_FOR_INTR34	WKG_ENB_FOR_INTR33	WKG_ENB_FOR_INTR32

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR63	Wakeup enable for interrupt line MPU_IRQ_63	RW	0x1
30	WKG_ENB_FOR_INTR62	Wakeup enable for interrupt line MPU_IRQ_62	RW	0x1
29	WKG_ENB_FOR_INTR61	Wakeup enable for interrupt line MPU_IRQ_61	RW	0x1
28	WKG_ENB_FOR_INTR60	Wakeup enable for interrupt line MPU_IRQ_60	RW	0x1
27	WKG_ENB_FOR_INTR59	Wakeup enable for interrupt line MPU_IRQ_59	RW	0x1
26	WKG_ENB_FOR_INTR58	Wakeup enable for interrupt line MPU_IRQ_58	RW	0x1
25	WKG_ENB_FOR_INTR57	Wakeup enable for interrupt line MPU_IRQ_57	RW	0x1
24	WKG_ENB_FOR_INTR56	Wakeup enable for interrupt line MPU_IRQ_56	RW	0x1
23	WKG_ENB_FOR_INTR55	Wakeup enable for interrupt line MPU_IRQ_55	RW	0x1

Bits	Field Name	Description	Type	Reset
22	WKG_ENB_FOR_INTR54	Wakeup enable for interrupt line MPU_IRQ_54	RW	0x1
21	WKG_ENB_FOR_INTR53	Wakeup enable for interrupt line MPU_IRQ_53	RW	0x1
20	WKG_ENB_FOR_INTR52	Wakeup enable for interrupt line MPU_IRQ_52	RW	0x1
19	WKG_ENB_FOR_INTR51	Wakeup enable for interrupt line MPU_IRQ_51	RW	0x1
18	WKG_ENB_FOR_INTR50	Wakeup enable for interrupt line MPU_IRQ_50	RW	0x1
17	WKG_ENB_FOR_INTR49	Wakeup enable for interrupt line MPU_IRQ_49	RW	0x1
16	WKG_ENB_FOR_INTR48	Wakeup enable for interrupt line MPU_IRQ_48	RW	0x1
15	WKG_ENB_FOR_INTR47	Wakeup enable for interrupt line MPU_IRQ_47	RW	0x1
14	WKG_ENB_FOR_INTR46	Wakeup enable for interrupt line MPU_IRQ_46	RW	0x1
13	WKG_ENB_FOR_INTR45	Wakeup enable for interrupt line MPU_IRQ_45	RW	0x1
12	WKG_ENB_FOR_INTR44	Wakeup enable for interrupt line MPU_IRQ_44	RW	0x1
11	WKG_ENB_FOR_INTR43	Wakeup enable for interrupt line MPU_IRQ_43	RW	0x1
10	WKG_ENB_FOR_INTR42	Wakeup enable for interrupt line MPU_IRQ_42	RW	0x1
9	WKG_ENB_FOR_INTR41	Wakeup enable for interrupt line MPU_IRQ_41	RW	0x1
8	WKG_ENB_FOR_INTR40	Wakeup enable for interrupt line MPU_IRQ_40	RW	0x1
7	WKG_ENB_FOR_INTR39	Wakeup enable for interrupt line MPU_IRQ_39	RW	0x1
6	WKG_ENB_FOR_INTR38	Wakeup enable for interrupt line MPU_IRQ_38	RW	0x1
5	WKG_ENB_FOR_INTR37	Wakeup enable for interrupt line MPU_IRQ_37	RW	0x1
4	WKG_ENB_FOR_INTR36	Wakeup enable for interrupt line MPU_IRQ_36	RW	0x1
3	WKG_ENB_FOR_INTR35	Wakeup enable for interrupt line MPU_IRQ_35	RW	0x1
2	WKG_ENB_FOR_INTR34	Wakeup enable for interrupt line MPU_IRQ_34	RW	0x1
1	WKG_ENB_FOR_INTR33	Wakeup enable for interrupt line MPU_IRQ_33	RW	0x1
0	WKG_ENB_FOR_INTR32	Wakeup enable for interrupt line MPU_IRQ_32	RW	0x1

Table 4-49. WKG_ENB_C_0

Address Offset	0x0000 0018		
Physical Address	0x4828 1018	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C0 (interrupts MPU_IRQ_64 to MPU_IRQ_95). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR95	WKG_ENB_FOR_INTR94	WKG_ENB_FOR_INTR93	WKG_ENB_FOR_INTR92	WKG_ENB_FOR_INTR91	WKG_ENB_FOR_INTR90	WKG_ENB_FOR_INTR89	WKG_ENB_FOR_INTR88	WKG_ENB_FOR_INTR87	WKG_ENB_FOR_INTR86	WKG_ENB_FOR_INTR85	WKG_ENB_FOR_INTR84	WKG_ENB_FOR_INTR83	WKG_ENB_FOR_INTR82	WKG_ENB_FOR_INTR81	WKG_ENB_FOR_INTR80	WKG_ENB_FOR_INTR79	WKG_ENB_FOR_INTR78	WKG_ENB_FOR_INTR77	WKG_ENB_FOR_INTR76	WKG_ENB_FOR_INTR75	WKG_ENB_FOR_INTR74	WKG_ENB_FOR_INTR73	WKG_ENB_FOR_INTR72	WKG_ENB_FOR_INTR71	WKG_ENB_FOR_INTR70	WKG_ENB_FOR_INTR69	WKG_ENB_FOR_INTR68	WKG_ENB_FOR_INTR67	WKG_ENB_FOR_INTR66	WKG_ENB_FOR_INTR65	WKG_ENB_FOR_INTR64

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR95	Wakeup enable for interrupt line MPU_IRQ_95	RW	0x1
30	WKG_ENB_FOR_INTR94	Wakeup enable for interrupt line MPU_IRQ_94	RW	0x1
29	WKG_ENB_FOR_INTR93	Wakeup enable for interrupt line MPU_IRQ_93	RW	0x1
28	WKG_ENB_FOR_INTR92	Wakeup enable for interrupt line MPU_IRQ_92	RW	0x1
27	WKG_ENB_FOR_INTR91	Wakeup enable for interrupt line MPU_IRQ_91	RW	0x1
26	WKG_ENB_FOR_INTR90	Wakeup enable for interrupt line MPU_IRQ_90	RW	0x1

Bits	Field Name	Description	Type	Reset
25	WKG_ENB_FOR_INTR89	Wakeup enable for interrupt line MPU_IRQ_89	RW	0x1
24	WKG_ENB_FOR_INTR88	Wakeup enable for interrupt line MPU_IRQ_88	RW	0x1
23	WKG_ENB_FOR_INTR87	Wakeup enable for interrupt line MPU_IRQ_87	RW	0x1
22	WKG_ENB_FOR_INTR86	Wakeup enable for interrupt line MPU_IRQ_86	RW	0x1
21	WKG_ENB_FOR_INTR85	Wakeup enable for interrupt line MPU_IRQ_85	RW	0x1
20	WKG_ENB_FOR_INTR84	Wakeup enable for interrupt line MPU_IRQ_84	RW	0x1
19	WKG_ENB_FOR_INTR83	Wakeup enable for interrupt line MPU_IRQ_83	RW	0x1
18	WKG_ENB_FOR_INTR82	Wakeup enable for interrupt line MPU_IRQ_82	RW	0x1
17	WKG_ENB_FOR_INTR81	Wakeup enable for interrupt line MPU_IRQ_81	RW	0x1
16	WKG_ENB_FOR_INTR80	Wakeup enable for interrupt line MPU_IRQ_80	RW	0x1
15	WKG_ENB_FOR_INTR79	Wakeup enable for interrupt line MPU_IRQ_79	RW	0x1
14	WKG_ENB_FOR_INTR78	Wakeup enable for interrupt line MPU_IRQ_78	RW	0x1
13	WKG_ENB_FOR_INTR77	Wakeup enable for interrupt line MPU_IRQ_77	RW	0x1
12	WKG_ENB_FOR_INTR76	Wakeup enable for interrupt line MPU_IRQ_76	RW	0x1
11	WKG_ENB_FOR_INTR75	Wakeup enable for interrupt line MPU_IRQ_75	RW	0x1
10	WKG_ENB_FOR_INTR74	Wakeup enable for interrupt line MPU_IRQ_74	RW	0x1
9	WKG_ENB_FOR_INTR73	Wakeup enable for interrupt line MPU_IRQ_73	RW	0x1
8	WKG_ENB_FOR_INTR72	Wakeup enable for interrupt line MPU_IRQ_72	RW	0x1
7	WKG_ENB_FOR_INTR71	Wakeup enable for interrupt line MPU_IRQ_71	RW	0x1
6	WKG_ENB_FOR_INTR70	Wakeup enable for interrupt line MPU_IRQ_70	RW	0x1
5	WKG_ENB_FOR_INTR69	Wakeup enable for interrupt line MPU_IRQ_69	RW	0x1
4	WKG_ENB_FOR_INTR68	Wakeup enable for interrupt line MPU_IRQ_68	RW	0x1
3	WKG_ENB_FOR_INTR67	Wakeup enable for interrupt line MPU_IRQ_67	RW	0x1
2	WKG_ENB_FOR_INTR66	Wakeup enable for interrupt line MPU_IRQ_66	RW	0x1
1	WKG_ENB_FOR_INTR65	Wakeup enable for interrupt line MPU_IRQ_65	RW	0x1
0	WKG_ENB_FOR_INTR64	Wakeup enable for interrupt line MPU_IRQ_64	RW	0x1

Table 4-50. WKG_ENB_D_0

Address Offset	0x0000 001C		
Physical Address	0x4828 101C	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C0 (interrupts MPU_IRQ_96 to MPU_IRQ_127). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKG_ENB_FOR_INTR127	WKG_ENB_FOR_INTR126	WKG_ENB_FOR_INTR125	WKG_ENB_FOR_INTR124	WKG_ENB_FOR_INTR123	WKG_ENB_FOR_INTR122	WKG_ENB_FOR_INTR121	WKG_ENB_FOR_INTR120	WKG_ENB_FOR_INTR119	WKG_ENB_FOR_INTR118	WKG_ENB_FOR_INTR117	WKG_ENB_FOR_INTR116	WKG_ENB_FOR_INTR115	WKG_ENB_FOR_INTR114	WKG_ENB_FOR_INTR113	WKG_ENB_FOR_INTR112

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR111	WKG_ENB_FOR_INTR110	WKG_ENB_FOR_INTR109	WKG_ENB_FOR_INTR108	WKG_ENB_FOR_INTR107	WKG_ENB_FOR_INTR106	WKG_ENB_FOR_INTR105	WKG_ENB_FOR_INTR104	WKG_ENB_FOR_INTR103	WKG_ENB_FOR_INTR102	WKG_ENB_FOR_INTR101	WKG_ENB_FOR_INTR100	WKG_ENB_FOR_INTR99	WKG_ENB_FOR_INTR98	WKG_ENB_FOR_INTR97	WKG_ENB_FOR_INTR96

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR127	Wakeup enable for interrupt line MPU_IRQ_127	RW	0x1
30	WKG_ENB_FOR_INTR126	Wakeup enable for interrupt line MPU_IRQ_126	RW	0x1

Bits	Field Name	Description	Type	Reset
29	WKG_ENB_FOR_INTR125	Wakeup enable for interrupt line MPU_IRQ_125	RW	0x1
28	WKG_ENB_FOR_INTR124	Wakeup enable for interrupt line MPU_IRQ_124	RW	0x1
27	WKG_ENB_FOR_INTR123	Wakeup enable for interrupt line MPU_IRQ_123	RW	0x1
26	WKG_ENB_FOR_INTR122	Wakeup enable for interrupt line MPU_IRQ_122	RW	0x1
25	WKG_ENB_FOR_INTR121	Wakeup enable for interrupt line MPU_IRQ_121	RW	0x1
24	WKG_ENB_FOR_INTR120	Wakeup enable for interrupt line MPU_IRQ_120	RW	0x1
23	WKG_ENB_FOR_INTR119	Wakeup enable for interrupt line MPU_IRQ_119	RW	0x1
22	WKG_ENB_FOR_INTR118	Wakeup enable for interrupt line MPU_IRQ_118	RW	0x1
21	WKG_ENB_FOR_INTR117	Wakeup enable for interrupt line MPU_IRQ_117	RW	0x1
20	WKG_ENB_FOR_INTR116	Wakeup enable for interrupt line MPU_IRQ_116	RW	0x1
19	WKG_ENB_FOR_INTR115	Wakeup enable for interrupt line MPU_IRQ_115	RW	0x1
18	WKG_ENB_FOR_INTR114	Wakeup enable for interrupt line MPU_IRQ_114	RW	0x1
17	WKG_ENB_FOR_INTR113	Wakeup enable for interrupt line MPU_IRQ_113	RW	0x1
16	WKG_ENB_FOR_INTR112	Wakeup enable for interrupt line MPU_IRQ_112	RW	0x1
15	WKG_ENB_FOR_INTR111	Wakeup enable for interrupt line MPU_IRQ_111	RW	0x1
14	WKG_ENB_FOR_INTR110	Wakeup enable for interrupt line MPU_IRQ_110	RW	0x1
13	WKG_ENB_FOR_INTR109	Wakeup enable for interrupt line MPU_IRQ_109	RW	0x1
12	WKG_ENB_FOR_INTR108	Wakeup enable for interrupt line MPU_IRQ_108	RW	0x1
11	WKG_ENB_FOR_INTR107	Wakeup enable for interrupt line MPU_IRQ_107	RW	0x1
10	WKG_ENB_FOR_INTR106	Wakeup enable for interrupt line MPU_IRQ_106	RW	0x1
9	WKG_ENB_FOR_INTR105	Wakeup enable for interrupt line MPU_IRQ_105	RW	0x1
8	WKG_ENB_FOR_INTR104	Wakeup enable for interrupt line MPU_IRQ_104	RW	0x1
7	WKG_ENB_FOR_INTR103	Wakeup enable for interrupt line MPU_IRQ_103	RW	0x1
6	WKG_ENB_FOR_INTR102	Wakeup enable for interrupt line MPU_IRQ_102	RW	0x1
5	WKG_ENB_FOR_INTR101	Wakeup enable for interrupt line MPU_IRQ_101	RW	0x1
4	WKG_ENB_FOR_INTR100	Wakeup enable for interrupt line MPU_IRQ_100	RW	0x1
3	WKG_ENB_FOR_INTR99	Wakeup enable for interrupt line MPU_IRQ_99	RW	0x1
2	WKG_ENB_FOR_INTR98	Wakeup enable for interrupt line MPU_IRQ_98	RW	0x1
1	WKG_ENB_FOR_INTR97	Wakeup enable for interrupt line MPU_IRQ_97	RW	0x1
0	WKG_ENB_FOR_INTR96	Wakeup enable for interrupt line MPU_IRQ_96	RW	0x1

Table 4-51. WKG_ENB_E_0

Address Offset	0x0000 001C		
Physical Address	0x4828 1020	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C0 (interrupts MPU_IRQ_128 to MPU_IRQ_159). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

WKG_ENB_FOR_INTR159	WKG_ENB_FOR_INTR158	WKG_ENB_FOR_INTR157	WKG_ENB_FOR_INTR156	WKG_ENB_FOR_INTR155	WKG_ENB_FOR_INTR154	WKG_ENB_FOR_INTR153	WKG_ENB_FOR_INTR152	WKG_ENB_FOR_INTR151	WKG_ENB_FOR_INTR150	WKG_ENB_FOR_INTR149	WKG_ENB_FOR_INTR148	WKG_ENB_FOR_INTR147	WKG_ENB_FOR_INTR146	WKG_ENB_FOR_INTR145	WKG_ENB_FOR_INTR144	WKG_ENB_FOR_INTR143	WKG_ENB_FOR_INTR142	WKG_ENB_FOR_INTR141	WKG_ENB_FOR_INTR140	WKG_ENB_FOR_INTR139	WKG_ENB_FOR_INTR138	WKG_ENB_FOR_INTR137	WKG_ENB_FOR_INTR136	WKG_ENB_FOR_INTR135	WKG_ENB_FOR_INTR134	WKG_ENB_FOR_INTR133	WKG_ENB_FOR_INTR132	WKG_ENB_FOR_INTR131	WKG_ENB_FOR_INTR130	WKG_ENB_FOR_INTR129	WKG_ENB_FOR_INTR128

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR159	Wakeup enable for interrupt line MPU_IRQ_159	RW	0x1
30	WKG_ENB_FOR_INTR158	Wakeup enable for interrupt line MPU_IRQ_158	RW	0x1
29	WKG_ENB_FOR_INTR157	Wakeup enable for interrupt line MPU_IRQ_157	RW	0x1
28	WKG_ENB_FOR_INTR156	Wakeup enable for interrupt line MPU_IRQ_156	RW	0x1
27	WKG_ENB_FOR_INTR155	Wakeup enable for interrupt line MPU_IRQ_155	RW	0x1
26	WKG_ENB_FOR_INTR154	Wakeup enable for interrupt line MPU_IRQ_154	RW	0x1
25	WKG_ENB_FOR_INTR153	Wakeup enable for interrupt line MPU_IRQ_153	RW	0x1
24	WKG_ENB_FOR_INTR152	Wakeup enable for interrupt line MPU_IRQ_152	RW	0x1
23	WKG_ENB_FOR_INTR151	Wakeup enable for interrupt line MPU_IRQ_151	RW	0x1
22	WKG_ENB_FOR_INTR150	Wakeup enable for interrupt line MPU_IRQ_150	RW	0x1
21	WKG_ENB_FOR_INTR149	Wakeup enable for interrupt line MPU_IRQ_149	RW	0x1
20	WKG_ENB_FOR_INTR148	Wakeup enable for interrupt line MPU_IRQ_148	RW	0x1
19	WKG_ENB_FOR_INTR147	Wakeup enable for interrupt line MPU_IRQ_147	RW	0x1
18	WKG_ENB_FOR_INTR146	Wakeup enable for interrupt line MPU_IRQ_146	RW	0x1
17	WKG_ENB_FOR_INTR145	Wakeup enable for interrupt line MPU_IRQ_145	RW	0x1
16	WKG_ENB_FOR_INTR144	Wakeup enable for interrupt line MPU_IRQ_144	RW	0x1
15	WKG_ENB_FOR_INTR143	Wakeup enable for interrupt line MPU_IRQ_143	RW	0x1
14	WKG_ENB_FOR_INTR142	Wakeup enable for interrupt line MPU_IRQ_142	RW	0x1
13	WKG_ENB_FOR_INTR141	Wakeup enable for interrupt line MPU_IRQ_141	RW	0x1
12	WKG_ENB_FOR_INTR140	Wakeup enable for interrupt line MPU_IRQ_140	RW	0x1
11	WKG_ENB_FOR_INTR139	Wakeup enable for interrupt line MPU_IRQ_139	RW	0x1
10	WKG_ENB_FOR_INTR138	Wakeup enable for interrupt line MPU_IRQ_138	RW	0x1
9	WKG_ENB_FOR_INTR137	Wakeup enable for interrupt line MPU_IRQ_137	RW	0x1
8	WKG_ENB_FOR_INTR136	Wakeup enable for interrupt line MPU_IRQ_136	RW	0x1
7	WKG_ENB_FOR_INTR135	Wakeup enable for interrupt line MPU_IRQ_135	RW	0x1
6	WKG_ENB_FOR_INTR134	Wakeup enable for interrupt line MPU_IRQ_134	RW	0x1
5	WKG_ENB_FOR_INTR133	Wakeup enable for interrupt line MPU_IRQ_133	RW	0x1
4	WKG_ENB_FOR_INTR132	Wakeup enable for interrupt line MPU_IRQ_132	RW	0x1
3	WKG_ENB_FOR_INTR131	Wakeup enable for interrupt line MPU_IRQ_131	RW	0x1
2	WKG_ENB_FOR_INTR130	Wakeup enable for interrupt line MPU_IRQ_130	RW	0x1
1	WKG_ENB_FOR_INTR129	Wakeup enable for interrupt line MPU_IRQ_129	RW	0x1
0	WKG_ENB_FOR_INTR128	Wakeup enable for interrupt line MPU_IRQ_128	RW	0x1

Table 4-52. WKG_CONTROL_1

Address Offset	0x0000 0400		
Physical Address	0x4828 1400	**Instance**	MPU_WUGEN
Description	Wake-up generator control and status register for MPU_C1		

Table 4-52. WKG_CONTROL_1 (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DOMAINRESET	MPU_WARM_RESET	MPU_COLD_RESET	RESERVED		EVENTO	STANDBYWFE	STANDBYWFI	RESERVED							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x00000
15	DOMAINRESET	MPU always-on power domain (PD_MPUAON) reset status bit. It shows if the reset occurred previously. 0x0: No reset occurred. 0x1: Reset occurred.	R	0x0
14	MPU_WARM_RESET	This bit is set when the MPU_WARM_RESET signal is asserted. 0x0: MPU_WARM_RESET reset signal has not been asserted. 0x1: MPU_WARM_RESET reset request has been asserted.	RW	0x0
13	MPU_COLD_RESET	This bit is set when the MPU_COLD_RESET signal is asserted. 0x0: MPU_COLD_RESET reset signal has not been asserted. 0x1: MPU_COLD_RESET reset request has been asserted.	RW	0x0
12:11	RESERVED	Reserved	R	0x0
10	EVENTO	EVENTO status bit. The event output signal is active, when one SEV instruction is executed. This bit is set when a rising edge of EVENTO from CPU is detected. 0x0: Rising edge of EVENTO is not detected. 0x1: Rising edge of EVENTO is detected.	RW	0x0
9	STANDBYWFE	This bit gives software the visibility to track whether WFE mode have been entered. 0x0: WFE mode has not been entered. 0x1: WFE mode has been entered.	RW	0x0
8	STANDBYWFI	This bit gives software the visibility to track whether WFI mode have been entered. 0x0: WFI mode has not been entered. 0x1: WFI mode has been entered.	RW	0x0
7:0	RESERVED	Reserved	R	0x0

Table 4-53. WKG_ENB_A_1

Address Offset	0x0000 0410		
Physical Address	0x4828 1410	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C1 (interrupts MPU_IRQ_0 to MPU_IRQ_31). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

WKG_ENB_FOR_INTR31	WKG_ENB_FOR_INTR30	WKG_ENB_FOR_INTR29	WKG_ENB_FOR_INTR28	WKG_ENB_FOR_INTR27	WKG_ENB_FOR_INTR26	WKG_ENB_FOR_INTR25	WKG_ENB_FOR_INTR24	WKG_ENB_FOR_INTR23	WKG_ENB_FOR_INTR22	WKG_ENB_FOR_INTR21	WKG_ENB_FOR_INTR20	WKG_ENB_FOR_INTR19	WKG_ENB_FOR_INTR18	WKG_ENB_FOR_INTR17	WKG_ENB_FOR_INTR16	WKG_ENB_FOR_INTR15	WKG_ENB_FOR_INTR14	WKG_ENB_FOR_INTR13	WKG_ENB_FOR_INTR12	WKG_ENB_FOR_INTR11	WKG_ENB_FOR_INTR10	WKG_ENB_FOR_INTR9	WKG_ENB_FOR_INTR8	WKG_ENB_FOR_INTR7	WKG_ENB_FOR_INTR6	WKG_ENB_FOR_INTR5	WKG_ENB_FOR_INTR4	WKG_ENB_FOR_INTR3	WKG_ENB_FOR_INTR2	WKG_ENB_FOR_INTR1	WKG_ENB_FOR_INTR0

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR31	Wakeup enable for interrupt line MPU_IRQ_31	RW	0x1
30	WKG_ENB_FOR_INTR30	Wakeup enable for interrupt line MPU_IRQ_30	RW	0x1
29	WKG_ENB_FOR_INTR29	Wakeup enable for interrupt line MPU_IRQ_29	RW	0x1
28	WKG_ENB_FOR_INTR28	Wakeup enable for interrupt line MPU_IRQ_28	RW	0x1
27	WKG_ENB_FOR_INTR27	Wakeup enable for interrupt line MPU_IRQ_27	RW	0x1
26	WKG_ENB_FOR_INTR26	Wakeup enable for interrupt line MPU_IRQ_26	RW	0x1
25	WKG_ENB_FOR_INTR25	Wakeup enable for interrupt line MPU_IRQ_25	RW	0x1
24	WKG_ENB_FOR_INTR24	Wakeup enable for interrupt line MPU_IRQ_24	RW	0x1
23	WKG_ENB_FOR_INTR23	Wakeup enable for interrupt line MPU_IRQ_23	RW	0x1
22	WKG_ENB_FOR_INTR22	Wakeup enable for interrupt line MPU_IRQ_22	RW	0x1
21	WKG_ENB_FOR_INTR21	Wakeup enable for interrupt line MPU_IRQ_21	RW	0x1
20	WKG_ENB_FOR_INTR20	Wakeup enable for interrupt line MPU_IRQ_20	RW	0x1
19	WKG_ENB_FOR_INTR19	Wakeup enable for interrupt line MPU_IRQ_19	RW	0x1
18	WKG_ENB_FOR_INTR18	Wakeup enable for interrupt line MPU_IRQ_18	RW	0x1
17	WKG_ENB_FOR_INTR17	Wakeup enable for interrupt line MPU_IRQ_17	RW	0x1
16	WKG_ENB_FOR_INTR16	Wakeup enable for interrupt line MPU_IRQ_16	RW	0x1
15	WKG_ENB_FOR_INTR15	Wakeup enable for interrupt line MPU_IRQ_15	RW	0x1
14	WKG_ENB_FOR_INTR14	Wakeup enable for interrupt line MPU_IRQ_14	RW	0x1
13	WKG_ENB_FOR_INTR13	Wakeup enable for interrupt line MPU_IRQ_13	RW	0x1
12	WKG_ENB_FOR_INTR12	Wakeup enable for interrupt line MPU_IRQ_12	RW	0x1
11	WKG_ENB_FOR_INTR11	Wakeup enable for interrupt line MPU_IRQ_11	RW	0x1
10	WKG_ENB_FOR_INTR10	Wakeup enable for interrupt line MPU_IRQ_10	RW	0x1
9	WKG_ENB_FOR_INTR9	Wakeup enable for interrupt line MPU_IRQ_9	RW	0x1
8	WKG_ENB_FOR_INTR8	Wakeup enable for interrupt line MPU_IRQ_8	RW	0x0
7	WKG_ENB_FOR_INTR7	Wakeup enable for interrupt line MPU_IRQ_7	RW	0x1
6	WKG_ENB_FOR_INTR6	Wakeup enable for interrupt line MPU_IRQ_6	RW	0x1
5	WKG_ENB_FOR_INTR5	Wakeup enable for interrupt line MPU_IRQ_5	RW	0x1
4	WKG_ENB_FOR_INTR4	Wakeup enable for interrupt line MPU_IRQ_4	RW	0x1
3	WKG_ENB_FOR_INTR3	Wakeup enable for interrupt line MPU_IRQ_3	RW	0x1
2	WKG_ENB_FOR_INTR2	Wakeup enable for interrupt line MPU_IRQ_2	RW	0x1
1	WKG_ENB_FOR_INTR1	Wakeup enable for interrupt line MPU_IRQ_1	RW	0x1
0	WKG_ENB_FOR_INTR0	Wakeup enable for interrupt line MPU_IRQ_0	RW	0x1

Table 4-54. WKG_ENB_B_1

Address Offset	0x0000 0414		
Physical Address	0x4828 1414	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C1 (interrupts MPU_IRQ_32 to MPU_IRQ_63). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		

Table 4-54. WKG_ENB_B_1 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR63	WKG_ENB_FOR_INTR62	WKG_ENB_FOR_INTR61	WKG_ENB_FOR_INTR60	WKG_ENB_FOR_INTR59	WKG_ENB_FOR_INTR58	WKG_ENB_FOR_INTR57	WKG_ENB_FOR_INTR56	WKG_ENB_FOR_INTR55	WKG_ENB_FOR_INTR54	WKG_ENB_FOR_INTR53	WKG_ENB_FOR_INTR52	WKG_ENB_FOR_INTR51	WKG_ENB_FOR_INTR50	WKG_ENB_FOR_INTR49	WKG_ENB_FOR_INTR48	WKG_ENB_FOR_INTR47	WKG_ENB_FOR_INTR46	WKG_ENB_FOR_INTR45	WKG_ENB_FOR_INTR44	WKG_ENB_FOR_INTR43	WKG_ENB_FOR_INTR42	WKG_ENB_FOR_INTR41	WKG_ENB_FOR_INTR40	WKG_ENB_FOR_INTR39	WKG_ENB_FOR_INTR38	WKG_ENB_FOR_INTR37	WKG_ENB_FOR_INTR36	WKG_ENB_FOR_INTR35	WKG_ENB_FOR_INTR34	WKG_ENB_FOR_INTR33	WKG_ENB_FOR_INTR32

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR63	Wakeup enable for interrupt line MPU_IRQ_63	RW	0x1
30	WKG_ENB_FOR_INTR62	Wakeup enable for interrupt line MPU_IRQ_62	RW	0x1
29	WKG_ENB_FOR_INTR61	Wakeup enable for interrupt line MPU_IRQ_61	RW	0x1
28	WKG_ENB_FOR_INTR60	Wakeup enable for interrupt line MPU_IRQ_60	RW	0x1
27	WKG_ENB_FOR_INTR59	Wakeup enable for interrupt line MPU_IRQ_59	RW	0x1
26	WKG_ENB_FOR_INTR58	Wakeup enable for interrupt line MPU_IRQ_58	RW	0x1
25	WKG_ENB_FOR_INTR57	Wakeup enable for interrupt line MPU_IRQ_57	RW	0x1
24	WKG_ENB_FOR_INTR56	Wakeup enable for interrupt line MPU_IRQ_56	RW	0x1
23	WKG_ENB_FOR_INTR55	Wakeup enable for interrupt line MPU_IRQ_55	RW	0x1
22	WKG_ENB_FOR_INTR54	Wakeup enable for interrupt line MPU_IRQ_54	RW	0x1
21	WKG_ENB_FOR_INTR53	Wakeup enable for interrupt line MPU_IRQ_53	RW	0x1
20	WKG_ENB_FOR_INTR52	Wakeup enable for interrupt line MPU_IRQ_52	RW	0x1
19	WKG_ENB_FOR_INTR51	Wakeup enable for interrupt line MPU_IRQ_51	RW	0x1
18	WKG_ENB_FOR_INTR50	Wakeup enable for interrupt line MPU_IRQ_50	RW	0x1
17	WKG_ENB_FOR_INTR49	Wakeup enable for interrupt line MPU_IRQ_49	RW	0x1
16	WKG_ENB_FOR_INTR48	Wakeup enable for interrupt line MPU_IRQ_48	RW	0x1
15	WKG_ENB_FOR_INTR47	Wakeup enable for interrupt line MPU_IRQ_47	RW	0x1
14	WKG_ENB_FOR_INTR46	Wakeup enable for interrupt line MPU_IRQ_46	RW	0x1
13	WKG_ENB_FOR_INTR45	Wakeup enable for interrupt line MPU_IRQ_45	RW	0x1
12	WKG_ENB_FOR_INTR44	Wakeup enable for interrupt line MPU_IRQ_44	RW	0x1
11	WKG_ENB_FOR_INTR43	Wakeup enable for interrupt line MPU_IRQ_43	RW	0x1
10	WKG_ENB_FOR_INTR42	Wakeup enable for interrupt line MPU_IRQ_42	RW	0x1
9	WKG_ENB_FOR_INTR41	Wakeup enable for interrupt line MPU_IRQ_41	RW	0x1
8	WKG_ENB_FOR_INTR40	Wakeup enable for interrupt line MPU_IRQ_40	RW	0x1
7	WKG_ENB_FOR_INTR39	Wakeup enable for interrupt line MPU_IRQ_39	RW	0x1
6	WKG_ENB_FOR_INTR38	Wakeup enable for interrupt line MPU_IRQ_38	RW	0x1
5	WKG_ENB_FOR_INTR37	Wakeup enable for interrupt line MPU_IRQ_37	RW	0x1
4	WKG_ENB_FOR_INTR36	Wakeup enable for interrupt line MPU_IRQ_36	RW	0x1
3	WKG_ENB_FOR_INTR35	Wakeup enable for interrupt line MPU_IRQ_35	RW	0x1
2	WKG_ENB_FOR_INTR34	Wakeup enable for interrupt line MPU_IRQ_34	RW	0x1
1	WKG_ENB_FOR_INTR33	Wakeup enable for interrupt line MPU_IRQ_33	RW	0x1
0	WKG_ENB_FOR_INTR32	Wakeup enable for interrupt line MPU_IRQ_32	RW	0x1

Table 4-55. WKG_ENB_C_1

Address Offset	0x0000 0418

Table 4-55. WKG_ENB_C_1 (continued)

Physical Address	0x4828 1418	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C1 (interrupts MPU_IRQ_64 to MPU_IRQ_95). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKG_ENB_FOR_INTR95	WKG_ENB_FOR_INTR94	WKG_ENB_FOR_INTR93	WKG_ENB_FOR_INTR92	WKG_ENB_FOR_INTR91	WKG_ENB_FOR_INTR90	WKG_ENB_FOR_INTR89	WKG_ENB_FOR_INTR88	WKG_ENB_FOR_INTR87	WKG_ENB_FOR_INTR86	WKG_ENB_FOR_INTR85	WKG_ENB_FOR_INTR84	WKG_ENB_FOR_INTR83	WKG_ENB_FOR_INTR82	WKG_ENB_FOR_INTR81	WKG_ENB_FOR_INTR80

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR79	WKG_ENB_FOR_INTR78	WKG_ENB_FOR_INTR77	WKG_ENB_FOR_INTR76	WKG_ENB_FOR_INTR75	WKG_ENB_FOR_INTR74	WKG_ENB_FOR_INTR73	WKG_ENB_FOR_INTR72	WKG_ENB_FOR_INTR71	WKG_ENB_FOR_INTR70	WKG_ENB_FOR_INTR69	WKG_ENB_FOR_INTR68	WKG_ENB_FOR_INTR67	WKG_ENB_FOR_INTR66	WKG_ENB_FOR_INTR65	WKG_ENB_FOR_INTR64

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR95	Wakeup enable for interrupt line MPU_IRQ_95	RW	0x1
30	WKG_ENB_FOR_INTR94	Wakeup enable for interrupt line MPU_IRQ_94	RW	0x1
29	WKG_ENB_FOR_INTR93	Wakeup enable for interrupt line MPU_IRQ_93	RW	0x1
28	WKG_ENB_FOR_INTR92	Wakeup enable for interrupt line MPU_IRQ_92	RW	0x1
27	WKG_ENB_FOR_INTR91	Wakeup enable for interrupt line MPU_IRQ_91	RW	0x1
26	WKG_ENB_FOR_INTR90	Wakeup enable for interrupt line MPU_IRQ_90	RW	0x1
25	WKG_ENB_FOR_INTR89	Wakeup enable for interrupt line MPU_IRQ_89	RW	0x1
24	WKG_ENB_FOR_INTR88	Wakeup enable for interrupt line MPU_IRQ_88	RW	0x1
23	WKG_ENB_FOR_INTR87	Wakeup enable for interrupt line MPU_IRQ_87	RW	0x1
22	WKG_ENB_FOR_INTR86	Wakeup enable for interrupt line MPU_IRQ_86	RW	0x1
21	WKG_ENB_FOR_INTR85	Wakeup enable for interrupt line MPU_IRQ_85	RW	0x1
20	WKG_ENB_FOR_INTR84	Wakeup enable for interrupt line MPU_IRQ_84	RW	0x1
19	WKG_ENB_FOR_INTR83	Wakeup enable for interrupt line MPU_IRQ_83	RW	0x1
18	WKG_ENB_FOR_INTR82	Wakeup enable for interrupt line MPU_IRQ_82	RW	0x1
17	WKG_ENB_FOR_INTR81	Wakeup enable for interrupt line MPU_IRQ_81	RW	0x1
16	WKG_ENB_FOR_INTR80	Wakeup enable for interrupt line MPU_IRQ_80	RW	0x1
15	WKG_ENB_FOR_INTR79	Wakeup enable for interrupt line MPU_IRQ_79	RW	0x1
14	WKG_ENB_FOR_INTR78	Wakeup enable for interrupt line MPU_IRQ_78	RW	0x1
13	WKG_ENB_FOR_INTR77	Wakeup enable for interrupt line MPU_IRQ_77	RW	0x1
12	WKG_ENB_FOR_INTR76	Wakeup enable for interrupt line MPU_IRQ_76	RW	0x1
11	WKG_ENB_FOR_INTR75	Wakeup enable for interrupt line MPU_IRQ_75	RW	0x1
10	WKG_ENB_FOR_INTR74	Wakeup enable for interrupt line MPU_IRQ_74	RW	0x1
9	WKG_ENB_FOR_INTR73	Wakeup enable for interrupt line MPU_IRQ_73	RW	0x1
8	WKG_ENB_FOR_INTR72	Wakeup enable for interrupt line MPU_IRQ_72	RW	0x1
7	WKG_ENB_FOR_INTR71	Wakeup enable for interrupt line MPU_IRQ_71	RW	0x1
6	WKG_ENB_FOR_INTR70	Wakeup enable for interrupt line MPU_IRQ_70	RW	0x1
5	WKG_ENB_FOR_INTR69	Wakeup enable for interrupt line MPU_IRQ_69	RW	0x1
4	WKG_ENB_FOR_INTR68	Wakeup enable for interrupt line MPU_IRQ_68	RW	0x1
3	WKG_ENB_FOR_INTR67	Wakeup enable for interrupt line MPU_IRQ_67	RW	0x1
2	WKG_ENB_FOR_INTR66	Wakeup enable for interrupt line MPU_IRQ_66	RW	0x1
1	WKG_ENB_FOR_INTR65	Wakeup enable for interrupt line MPU_IRQ_65	RW	0x1

Bits	Field Name	Description	Type	Reset
0	WKG_ENB_FOR_INTR64	Wakeup enable for interrupt line MPU_IRQ_64	RW	0x1

Table 4-56. WKG_ENB_D_1

Address Offset	0x0000 041C		
Physical Address	0x4828 141C	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C1 (interrupts MPU_IRQ_96 to MPU_IRQ_127). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKG_ENB_FOR_INTR127	WKG_ENB_FOR_INTR126	WKG_ENB_FOR_INTR125	WKG_ENB_FOR_INTR124	WKG_ENB_FOR_INTR123	WKG_ENB_FOR_INTR122	WKG_ENB_FOR_INTR121	WKG_ENB_FOR_INTR120	WKG_ENB_FOR_INTR119	WKG_ENB_FOR_INTR118	WKG_ENB_FOR_INTR117	WKG_ENB_FOR_INTR116	WKG_ENB_FOR_INTR115	WKG_ENB_FOR_INTR114	WKG_ENB_FOR_INTR113	WKG_ENB_FOR_INTR112

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR111	WKG_ENB_FOR_INTR110	WKG_ENB_FOR_INTR109	WKG_ENB_FOR_INTR108	WKG_ENB_FOR_INTR107	WKG_ENB_FOR_INTR106	WKG_ENB_FOR_INTR105	WKG_ENB_FOR_INTR104	WKG_ENB_FOR_INTR103	WKG_ENB_FOR_INTR102	WKG_ENB_FOR_INTR101	WKG_ENB_FOR_INTR100	WKG_ENB_FOR_INTR99	WKG_ENB_FOR_INTR98	WKG_ENB_FOR_INTR97	WKG_ENB_FOR_INTR96

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR127	Wakeup enable for interrupt line MPU_IRQ_127	RW	0x1
30	WKG_ENB_FOR_INTR126	Wakeup enable for interrupt line MPU_IRQ_126	RW	0x1
29	WKG_ENB_FOR_INTR125	Wakeup enable for interrupt line MPU_IRQ_125	RW	0x1
28	WKG_ENB_FOR_INTR124	Wakeup enable for interrupt line MPU_IRQ_124	RW	0x1
27	WKG_ENB_FOR_INTR123	Wakeup enable for interrupt line MPU_IRQ_123	RW	0x1
26	WKG_ENB_FOR_INTR122	Wakeup enable for interrupt line MPU_IRQ_122	RW	0x1
25	WKG_ENB_FOR_INTR121	Wakeup enable for interrupt line MPU_IRQ_121	RW	0x1
24	WKG_ENB_FOR_INTR120	Wakeup enable for interrupt line MPU_IRQ_120	RW	0x1
23	WKG_ENB_FOR_INTR119	Wakeup enable for interrupt line MPU_IRQ_119	RW	0x1
22	WKG_ENB_FOR_INTR118	Wakeup enable for interrupt line MPU_IRQ_118	RW	0x1
21	WKG_ENB_FOR_INTR117	Wakeup enable for interrupt line MPU_IRQ_117	RW	0x1
20	WKG_ENB_FOR_INTR116	Wakeup enable for interrupt line MPU_IRQ_116	RW	0x1
19	WKG_ENB_FOR_INTR115	Wakeup enable for interrupt line MPU_IRQ_115	RW	0x1
18	WKG_ENB_FOR_INTR114	Wakeup enable for interrupt line MPU_IRQ_114	RW	0x1
17	WKG_ENB_FOR_INTR113	Wakeup enable for interrupt line MPU_IRQ_113	RW	0x1
16	WKG_ENB_FOR_INTR112	Wakeup enable for interrupt line MPU_IRQ_112	RW	0x1
15	WKG_ENB_FOR_INTR111	Wakeup enable for interrupt line MPU_IRQ_111	RW	0x1
14	WKG_ENB_FOR_INTR110	Wakeup enable for interrupt line MPU_IRQ_110	RW	0x1
13	WKG_ENB_FOR_INTR109	Wakeup enable for interrupt line MPU_IRQ_109	RW	0x1
12	WKG_ENB_FOR_INTR108	Wakeup enable for interrupt line MPU_IRQ_108	RW	0x1
11	WKG_ENB_FOR_INTR107	Wakeup enable for interrupt line MPU_IRQ_107	RW	0x1
10	WKG_ENB_FOR_INTR106	Wakeup enable for interrupt line MPU_IRQ_106	RW	0x1
9	WKG_ENB_FOR_INTR105	Wakeup enable for interrupt line MPU_IRQ_105	RW	0x1
8	WKG_ENB_FOR_INTR104	Wakeup enable for interrupt line MPU_IRQ_104	RW	0x1
7	WKG_ENB_FOR_INTR103	Wakeup enable for interrupt line MPU_IRQ_103	RW	0x1
6	WKG_ENB_FOR_INTR102	Wakeup enable for interrupt line MPU_IRQ_102	RW	0x1
5	WKG_ENB_FOR_INTR101	Wakeup enable for interrupt line MPU_IRQ_101	RW	0x1

Bits	Field Name	Description	Type	Reset
4	WKG_ENB_FOR_INTR100	Wakeup enable for interrupt line MPU_IRQ_100	RW	0x1
3	WKG_ENB_FOR_INTR99	Wakeup enable for interrupt line MPU_IRQ_99	RW	0x1
2	WKG_ENB_FOR_INTR98	Wakeup enable for interrupt line MPU_IRQ_98	RW	0x1
1	WKG_ENB_FOR_INTR97	Wakeup enable for interrupt line MPU_IRQ_97	RW	0x1
0	WKG_ENB_FOR_INTR96	Wakeup enable for interrupt line MPU_IRQ_96	RW	0x1

Table 4-57. WKG_ENB_E_1

Address Offset	0x0000 041C		
Physical Address	0x4828 1420	**Instance**	MPU_WUGEN
Description	Wake-up interrupt enable register for MPU_C1 (interrupts MPU_IRQ_128 to MPU_IRQ_159). Write 0x0: Disable interrupt. Write 0x1: Enable interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKG_ENB_FOR_INTR159	WKG_ENB_FOR_INTR158	WKG_ENB_FOR_INTR157	WKG_ENB_FOR_INTR156	WKG_ENB_FOR_INTR155	WKG_ENB_FOR_INTR154	WKG_ENB_FOR_INTR153	WKG_ENB_FOR_INTR152	WKG_ENB_FOR_INTR151	WKG_ENB_FOR_INTR150	WKG_ENB_FOR_INTR149	WKG_ENB_FOR_INTR148	WKG_ENB_FOR_INTR147	WKG_ENB_FOR_INTR146	WKG_ENB_FOR_INTR145	WKG_ENB_FOR_INTR144

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKG_ENB_FOR_INTR143	WKG_ENB_FOR_INTR142	WKG_ENB_FOR_INTR141	WKG_ENB_FOR_INTR140	WKG_ENB_FOR_INTR139	WKG_ENB_FOR_INTR138	WKG_ENB_FOR_INTR137	WKG_ENB_FOR_INTR136	WKG_ENB_FOR_INTR135	WKG_ENB_FOR_INTR134	WKG_ENB_FOR_INTR133	WKG_ENB_FOR_INTR132	WKG_ENB_FOR_INTR131	WKG_ENB_FOR_INTR130	WKG_ENB_FOR_INTR129	WKG_ENB_FOR_INTR128

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR159	Wakeup enable for interrupt line MPU_IRQ_159	RW	0x1
30	WKG_ENB_FOR_INTR158	Wakeup enable for interrupt line MPU_IRQ_158	RW	0x1
29	WKG_ENB_FOR_INTR157	Wakeup enable for interrupt line MPU_IRQ_157	RW	0x1
28	WKG_ENB_FOR_INTR156	Wakeup enable for interrupt line MPU_IRQ_156	RW	0x1
27	WKG_ENB_FOR_INTR155	Wakeup enable for interrupt line MPU_IRQ_155	RW	0x1
26	WKG_ENB_FOR_INTR154	Wakeup enable for interrupt line MPU_IRQ_154	RW	0x1
25	WKG_ENB_FOR_INTR153	Wakeup enable for interrupt line MPU_IRQ_153	RW	0x1
24	WKG_ENB_FOR_INTR152	Wakeup enable for interrupt line MPU_IRQ_152	RW	0x1
23	WKG_ENB_FOR_INTR151	Wakeup enable for interrupt line MPU_IRQ_151	RW	0x1
22	WKG_ENB_FOR_INTR150	Wakeup enable for interrupt line MPU_IRQ_150	RW	0x1
21	WKG_ENB_FOR_INTR149	Wakeup enable for interrupt line MPU_IRQ_149	RW	0x1
20	WKG_ENB_FOR_INTR148	Wakeup enable for interrupt line MPU_IRQ_148	RW	0x1
19	WKG_ENB_FOR_INTR147	Wakeup enable for interrupt line MPU_IRQ_147	RW	0x1
18	WKG_ENB_FOR_INTR146	Wakeup enable for interrupt line MPU_IRQ_146	RW	0x1
17	WKG_ENB_FOR_INTR145	Wakeup enable for interrupt line MPU_IRQ_145	RW	0x1
16	WKG_ENB_FOR_INTR144	Wakeup enable for interrupt line MPU_IRQ_144	RW	0x1
15	WKG_ENB_FOR_INTR143	Wakeup enable for interrupt line MPU_IRQ_143	RW	0x1
14	WKG_ENB_FOR_INTR142	Wakeup enable for interrupt line MPU_IRQ_142	RW	0x1
13	WKG_ENB_FOR_INTR141	Wakeup enable for interrupt line MPU_IRQ_141	RW	0x1
12	WKG_ENB_FOR_INTR140	Wakeup enable for interrupt line MPU_IRQ_140	RW	0x1
11	WKG_ENB_FOR_INTR139	Wakeup enable for interrupt line MPU_IRQ_139	RW	0x1
10	WKG_ENB_FOR_INTR138	Wakeup enable for interrupt line MPU_IRQ_138	RW	0x1
9	WKG_ENB_FOR_INTR137	Wakeup enable for interrupt line MPU_IRQ_137	RW	0x1

Bits	Field Name	Description	Type	Reset
8	WKG_ENB_FOR_INTR136	Wakeup enable for interrupt line MPU_IRQ_136	RW	0x1
7	WKG_ENB_FOR_INTR135	Wakeup enable for interrupt line MPU_IRQ_135	RW	0x1
6	WKG_ENB_FOR_INTR134	Wakeup enable for interrupt line MPU_IRQ_134	RW	0x1
5	WKG_ENB_FOR_INTR133	Wakeup enable for interrupt line MPU_IRQ_133	RW	0x1
4	WKG_ENB_FOR_INTR132	Wakeup enable for interrupt line MPU_IRQ_132	RW	0x1
3	WKG_ENB_FOR_INTR131	Wakeup enable for interrupt line MPU_IRQ_131	RW	0x1
2	WKG_ENB_FOR_INTR130	Wakeup enable for interrupt line MPU_IRQ_130	RW	0x1
1	WKG_ENB_FOR_INTR129	Wakeup enable for interrupt line MPU_IRQ_129	RW	0x1
0	WKG_ENB_FOR_INTR128	Wakeup enable for interrupt line MPU_IRQ_128	RW	0x1

Table 4-58. AUX_CORE_BOOT_0

Address Offset	0x800		
Physical Address	0x4828 1800	**Instance**	MPU_WUGEN
Description	This register is used by the ROM code and OS during SMP boot. It is used to indicate the boot status to MPU_C1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MPU_C1_STATUS				RESERVED			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x000000
7:4	MPU_C1_STATUS	MPU_C1 boot status. If ≠ 0x0, branch at the address specified in AUX_CORE_BOOT_1.	RW	0x0
3:0	RESERVED	Reserved	RW	0x0

Table 4-59. AUX_CORE_BOOT_1

Address Offset	0x804		
Physical Address	0x4828 1804	**Instance**	MPU_WUGEN
Description	This register is used by the ROM code and OS during SMP boot. It is intended to store the execution start address of MPU_C1. When needed, the SMP OS (executing on MPU_C0) stores the execution start address of MPU_C1 in AUX_CORE_BOOT_1, and then wakes up MPU_C1 by executing a SEV command. When MPU_C1 receives an event (caused by the SEV command), it executes the event handler in the ROM Code, which eventually branches to the address stored in AUX_CORE_BOOT_1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Field Name	Description	Type	Reset
31:0	AUX_CORE_BOOT_1	SMP boot register	RW	0x00000000

Table 4-60. STM_HWEVENTS_INV

Address Offset	0x0000 0808		
Physical Address	0x4828 1808	**Instance**	MPU_WUGEN
Description	Gives programmable control of inverting or not inverting MPUHWDBGOUT[31:0] going to HWEVENTS[31:0] input of CS_STM		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STM_HWEVENT_INV_31	STM_HWEVENT_INV_30	STM_HWEVENT_INV_29	STM_HWEVENT_INV_28	STM_HWEVENT_INV_27	STM_HWEVENT_INV_26	STM_HWEVENT_INV_25	STM_HWEVENT_INV_24	STM_HWEVENT_INV_23	STM_HWEVENT_INV_22	STM_HWEVENT_INV_21	STM_HWEVENT_INV_20	STM_HWEVENT_INV_19	STM_HWEVENT_INV_18	STM_HWEVENT_INV_17	STM_HWEVENT_INV_16	STM_HWEVENT_INV_15	STM_HWEVENT_INV_14	STM_HWEVENT_INV_13	STM_HWEVENT_INV_12	STM_HWEVENT_INV_11	STM_HWEVENT_INV_10	STM_HWEVENT_INV_9	STM_HWEVENT_INV_8	STM_HWEVENT_INV_7	STM_HWEVENT_INV_6	STM_HWEVENT_INV_5	STM_HWEVENT_INV_4	STM_HWEVENT_INV_3	STM_HWEVENT_INV_2	STM_HWEVENT_INV_1	STM_HWEVENT_INV_0

Bits	Field Name	Description		Type	Reset
31	STM_HWEVENT_INV_ 31	Polarity inversion control for MPUHWDBGOUT31 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
30	STM_HWEVENT_INV_ 30	Polarity inversion control for MPUHWDBGOUT30 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
29	STM_HWEVENT_INV_ 29	Polarity inversion control for MPUHWDBGOUT29 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
28	STM_HWEVENT_INV_ 28	Polarity inversion control for MPUHWDBGOUT28 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
27	STM_HWEVENT_INV_ 27	Polarity inversion control for MPUHWDBGOUT27 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
26	STM_HWEVENT_INV_ 26	Polarity inversion control for MPUHWDBGOUT26 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
25	STM_HWEVENT_INV_ 25	Polarity inversion control for MPUHWDBGOUT25 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
24	STM_HWEVENT_INV_ 24	Polarity inversion control for MPUHWDBGOUT24 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
23	STM_HWEVENT_INV_ 23	Polarity inversion control for MPUHWDBGOUT23 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
22	STM_HWEVENT_INV_ 22	Polarity inversion control for MPUHWDBGOUT22 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
21	STM_HWEVENT_INV_ 21	Polarity inversion control for MPUHWDBGOUT21 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
20	STM_HWEVENT_INV_ 20	Polarity inversion control for MPUHWDBGOUT20 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
19	STM_HWEVENT_INV_ 19	Polarity inversion control for MPUHWDBGOUT19 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		

Bits	Field Name	Description		Type	Reset
18	STM_HWEVENT_INV_ 18	Polarity inversion control for MPUHWDBGOUT18 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
17	STM_HWEVENT_INV_ 17	Polarity inversion control for MPUHWDBGOUT17 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
16	STM_HWEVENT_INV_ 16	Polarity inversion control for MPUHWDBGOUT16 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
15	STM_HWEVENT_INV_ 15	Polarity inversion control for MPUHWDBGOUT15 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
14	STM_HWEVENT_INV_ 14	Polarity inversion control for MPUHWDBGOUT14 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
13	STM_HWEVENT_INV_ 13	Polarity inversion control for MPUHWDBGOUT13 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
12	STM_HWEVENT_INV_ 12	Polarity inversion control for MPUHWDBGOUT12 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
11	STM_HWEVENT_INV_ 11	Polarity inversion control for MPUHWDBGOUT11 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
10	STM_HWEVENT_INV_ 10	Polarity inversion control for MPUHWDBGOUT10 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
9	STM_HWEVENT_INV_ 9	Polarity inversion control for MPUHWDBGOUT9 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
8	STM_HWEVENT_INV_ 8	Polarity inversion control for MPUHWDBGOUT8 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
7	STM_HWEVENT_INV_ 7	Polarity inversion control for MPUHWDBGOUT7 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
6	STM_HWEVENT_INV_ 6	Polarity inversion control for MPUHWDBGOUT6 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
5	STM_HWEVENT_INV_ 5	Polarity inversion control for MPUHWDBGOUT5 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		
4	STM_HWEVENT_INV_ 4	Polarity inversion control for MPUHWDBGOUT4 signal.		RW	0x0
		0x0:	Polarity unchanged		
		0x1	Polarity inverted		

Bits	Field Name	Description	Type	Reset
3	STM_HWEVENT_INV_ 3	Polarity inversion control for MPUHWDBGOUT3 signal. 0x0: Polarity unchanged 0x1 Polarity inverted	RW	0x0
2	STM_HWEVENT_INV_ 2	Polarity inversion control for MPUHWDBGOUT2 signal. 0x0: Polarity unchanged 0x1 Polarity inverted	RW	0x0
1	STM_HWEVENT_INV_ 1	Polarity inversion control for MPUHWDBGOUT1 signal. 0x0: Polarity unchanged 0x1 Polarity inverted	RW	0x0
0	STM_HWEVENT_INV_0	Polarity inversion control for MPUHWDBGOUT0 signal. 0x0: Polarity unchanged 0x1 Polarity inverted	RW	0x0

Table 4-61. AMBA_IF_MODE

Address Offset	0x0000 080C		
Physical Address	0x4828 180C	**Instance**	MPU_WUGEN
Description	This register controls the MPU core interface tie-off values for BI, BO, BCM and SBD. This register is located in MPU always-on domain and is reset by MPUAON_RST.		
Type	RW		

31:6	5	4	3	2	1	0
RESERVED	ES2_PM_MODE	APB_FENCE_EN	BI	BO	BCM	SBD

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved. Ignored on write and zero on read.	R	0x000 0000
5	ES2_PM_MODE	Enables OFF mode behavior. 0x0 : OFF Mode 1, CPUs would enter and exit OFF mode together. 0x1 : OFF Mode 2, CPUs are allowed to enter/exit OFF mode independently. **NOTE: This bit is NOT supported in this device.**	RW	0x0
4	APB_FENCE_EN	Enables APB fencing logic.	RW	0x1
3	BI	BROADCASTINNER input of MPU core.	RW	0x0
2	BO	BROADCASTOUTER input of MPU core.	RW	0x0
1	BCM	BROADCASTMAINTENANCE input of MPU core.	RW	0x0
0	SBD	SYSBARDISABLE input of MPU core.	RW	0x1

Table 4-62. TIMESTAMPCYCLELO

Address Offset	0x0000 0C08		
Physical Address	0x4828 1C08	**Instance**	MPU_WUGEN
Description	Lower 32 bits of the 48-bit timestamp counter value		
Type	R		

31:0
COUNTER_31_0

Bits	Field Name	Description	Type	Reset
31:0	COUNTER_31_0	Lower 32 bits of the 48-bit timestamp counter value.	R	0x000 0000

Table 4-63. TIMESTAMPCYCLEHI

Address Offset	0x0000 0C0C		
Physical Address	0x4828 1C0C	**Instance**	MPU_WUGEN
Description	Higher 16 bits of the 48-bit timestamp counter value		
Type	R		

31–16	15–0
RESERVED	COUNTER_47_32

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved. Ignored on write and zero on read.	R	0x0000
15:0	COUNTER_47_32	Higher 16 bits of the timestamp counter value.	R	0x0000

4.4.11 MPU_WD_TIMER Registers

4.4.11.1 MPU_WD_TIMER Register Summary

Table 4-64. MPU_WD_TIMER Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_WD_TIMER Physical Address
WDT_LOAD_REGISTER_i (1)	RW	32	0x0000 0000 + (0x20 * i)	0x482A 0000 + (0x20 * i)
WDT_COUNT_REGISTER_i (1)	R	32	0x0000 0004 + (0x20 * i)	0x482A 0004 + (0x20 * i)
WDT_WARNING_REGISTER_i (1)	RW	32	0x0000 0008 + (0x20 * i)	0x482A 0008 + (0x20 * i)
WDT_PRESCALER_REGISTER_i (1)	RW	32	0x0000 000C + (0x20 * i)	0x482A 000C + (0x20 * i)
WDT_CONTROL_REGISTER_i (1)	RW	32	0x0000 0010 + (0x20 * i)	0x482A 0010 + (0x20 * i)
WDT_RESET_STATUS_REGISTER_i (1)	RW	32	0x0000 0014 + (0x20 * i)	0x482A 0014 + (0x20 * i)

(1) i = 0 to 1

4.4.11.2 MPU_WD_TIMER Register Description

Table 4-65. WDT_LOAD_REGISTER_i

Address Offset	0x0000 0000 + (0x20 * i)	**Index**	i = 0 to 1
Physical Address	0x482A 0000 + (0x20 * i)	**Instance**	MPU_WD_TIMER
Description	When a new value is stored in this register, the WDT_COUNT_REGISTER_i is immediately loaded with this value and the prescaler state is cleared. This register is reset by warm reset of the corresponding CPU.		
Type	RW		

31–0
NEWCOUNT

Bits	Field Name	Description	Type	Reset
31:0	NEWCOUNT	New value to load into WDT_COUNT_REGISTER_i.	RW	0x0000 0000

Table 4-66. Register Call Summary for Register WDT_LOAD_REGISTER_i

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU Watchdog Timer: [0] [1] [2] [3] [4]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_WD_TIMER Register Summary: [5]
- MPU_WD_TIMER Register Description: [6]

Table 4-67. WDT_COUNT_REGISTER_i

Address Offset	0x0000 0004 + (0x20 * i)	**Index**	i = 0 to 1
Physical Address	0x482A 0004 + (0x20 * i)	**Instance**	MPU_WD_TIMER
Description	This register is a 32-bit decrementing counter. The decrement rate is programmed in the WDT_PRESCALER_REGISTER_i. The WDT_COUNT_REGISTER_i can be read to get the current count. It decrements if the MPU_WD_TIMER_Cx is enabled (WDT_CONTROL_REGISTER_i[0] ENABLE = 0x1). If the processor related to the corresponding watchdog channel is in debug state, the counter does not decrement until the processor returns to non-debug state. The WDT_COUNT_REGISTER_i decrements down to zero and stops. The only way to update the WDT_COUNT_REGISTER_i is to write to the WDT_LOAD_REGISTER_i. If a software failure prevents the WDT_COUNT_REGISTER_i from being refreshed, the WDT_COUNT_REGISTER_i reaches zero, the watchdog timeout status flag is set and all interrupt requests or reset requests enabled in the WDT_CONTROL_REGISTER_i are signalled. If a reset request is enabled, the global PRCM is then responsible for resetting the MPUSS. Debug state is inferred by monitoring the DBGACK signal corresponding to this core. This register is reset by warm reset of the corresponding MPU core.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRENTCOUNT																															

Bits	Field Name	Description	Type	Reset
31:0	CURRENTCOUNT	Current count of the MPU_WD_TIMER.	R	0x0000 0000

Table 4-68. Register Call Summary for Register WDT_COUNT_REGISTER_i

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU Watchdog Timer: [0] [1] [2]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_WD_TIMER Register Summary: [3]
- MPU_WD_TIMER Register Description: [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]

Table 4-69. WDT_WARNING_REGISTER_i

Address Offset	0x0000 0008 + (0x20 * i)	**Index**	i = 0 to 1
Physical Address	0x482A 0008 + (0x20 * i)	**Instance**	MPU_WD_TIMER
Description	The WDT_COUNT_REGISTER_i is compared to the WDT_WARNING_REGISTER_i. If WDT_COUNT_REGISTER_i is less than or equal to the WDT_WARNING_REGISTER_i and WDT_CONTROL_REGISTER_i[8] WARNEN = 0b1, a warning interrupt is signalled to the MPU_INTC. The warning condition can be used to signal an interrupt that gives software a notice that the MPU_WD_TIMER_Cx is getting close to a timeout, when a more serious action should be taken.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WARNING_WATERMARK																															

Bits	Field Name	Description	Type	Reset
31:0	WARNING_WATERMARK	A warning condition occurs when the WDT_COUNT_REGISTER_i value is less than or equal to the WDT_WARNING_REGISTER_i.	RW	0x0000 0000

Table 4-70. Register Call Summary for Register WDT_WARNING_REGISTER_i

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU Watchdog Timer: [0] [1] [2]

Table 4-70. Register Call Summary for Register WDT_WARNING_REGISTER_i (continued)

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_WD_TIMER Register Summary: [3]
- MPU_WD_TIMER Register Description: [4] [5] [6]

Table 4-71. WDT_PRESCALER_REGISTER_i

Address Offset	0x0000 000C + (0x20 * i)	**Index**	i = 0 to 1
Physical Address	0x482A 000C + (0x20 * i)	**Instance**	MPU_WD_TIMER
Description	This register is used to set the count rate of the MPU_WD_TIMER_Cx counter.		
Type	RW		

31–10	9–0
RESERVED	PRESCALER

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved. Ignored on writes. Reads return 0s.	R	0x00 0000
9:0	PRESCALER	Sets the prescaler ratio. WDT_COUNT_REGISTER_i decrements every (PRESCALER + 1) MPU_DPLL_CLK clocks. Note: If the prescaler is set to (MPU_DPLL_CLK [in MHz] - 1), the MPU_WD_TIMER_Cx counter counts at a 1 microsecond rate.	RW	0x000

Table 4-72. Register Call Summary for Register WDT_PRESCALER_REGISTER_i

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU Watchdog Timer: [0] [1] [2]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_WD_TIMER Register Summary: [3]
- MPU_WD_TIMER Register Description: [4]

Table 4-73. WDT_CONTROL_REGISTER_i

Address Offset	0x0000 0010 + (0x20 * i)	**Index**	i = 0 to 1
Physical Address	0x482A 0010 + (0x20 * i)	**Instance**	MPU_WD_TIMER
Description	This register controls the behavior of the MPU_WD_TIMER_Cx. This register is reset by warm reset of the corresponding MPU core.		
Type	RW		

31–9	8	7–4	3	2	1	0
RESERVED	WARNEN	RESERVED	MPUSSRSTEN	RESERVED	INTREN	ENABLE

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved. Ignored on writes. Reads return 0s.	R	0x00 0000
8	WARNEN	Warning Interrupt Enable. If this bit is set and the warning watermark test is true, a warning interrupt is generated to the MPU_INTC.	RW	0
7:4	RESERVED	Reserved. Ignored on writes. Reads return 0s.	R	0x0
3	MPUSSRSTEN	MPUSS Reset Enable. If this field is 0b1 when the timer reaches zero, a request is sent to the global PRCM to begin a global warm reset.	RW	0

Bits	Field Name	Description	Type	Reset
2	RESERVED	Reserved. Ignored on writes. Reads return 0s.	R	0
1	INTREN	Interrupt Enable. If this field is 0b1 when the timer reaches zero, an interrupt request is sent to the MPU_INTC.	RW	0
0	ENABLE	Enable for MPU_WD_TIMER_Cx. 0: MPU_WD_TIMER_Cx is disabled. It will not count down and it will not generate a reset request. All MPU_WD_TIMER_Cx registers may be accessed. 1: MPU_WD_TIMER_Cx is enabled. It will count down and generate a reset request if it reaches 0. This bit is reset by warm or power-on reset.	RW	0

Table 4-74. Register Call Summary for Register WDT_CONTROL_REGISTER_i

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU Watchdog Timer: [0] [1] [2] [3] [4] [5] [6]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_WD_TIMER Register Summary: [7]
- MPU_WD_TIMER Register Description: [8] [9] [10] [11] [12]

Table 4-75. WDT_RESET_STATUS_REGISTER_i

Address Offset	0x0000 0014 + (0x20 * i)	**Index**	i = 0 to 1
Physical Address	0x482A 0014 + (0x20 * i)	**Instance**	MPU_WD_TIMER
Description	The TO bit indicated that this MPU_WD_TIMER_Cx has timed out. This might be used to figure out which MPU_WD_TIMER_Cx signalled a reset. This register is not reset by warm reset, but only by cold reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														WARN	TO

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved. Ignored on writes. Reads return 0s.	R	0x0000 0000
1	WARN	Warning. Indicates that the count has passed the warning watermark level while the WDT_CONTROL_REGISTER_i[8] WARNEN bit was set. Write a '1' to this bit to reset it.	RW W1toClr	0
0	TO	Timeout. Indicates the WDT_COUNT_REGISTER_i has reached zero (timed out) and the signalling enabled in the WDT_CONTROL_REGISTER_i has occurred. Can be used to determine which MPU_WD_TIMER_Cx instance caused a reset. Write a '1' to this bit to reset it.	RW W1toClr	0

Table 4-76. Register Call Summary for Register WDT_RESET_STATUS_REGISTER_i

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU Watchdog Timer: [0] [1]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_WD_TIMER Register Summary: [2]

4.4.12 MPU_AXI2OCP_MISC Registers

4.4.12.1 MPU_AXI2OCP_MISC Register Summary

Table 4-77. MPU_AXI2OCP_MISC Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_AXI2OCP_MISC Physical Address
MA_PRIORITY	RW	32	0x0	0x482A 2000

4.4.12.2 MPU_AXI2OCP_MISC Register Description

Table 4-78. MA_PRIORITY

Address Offset	0x0000 0000		
Physical Address	0x482A 2000	**Instance**	MPU_AXI2OCP_MISC
Description	Memory adapter priority register. This register indicates the priority of memory access from MPU_MA to EMIF. This priority is used by EMIF in scheduling MPU_MA access to EMIF. 0x0 is highest priority and 0x7 is lowest priority.		
Type	RW		

31:9	8	7:3	2:0
RESERVED	HIMEM_INTERLEAVE_UN	RESERVED	PRIORITY

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x00 0000
8	HIMEM_INTERLEAVE_UN	HIMEM_INTERLEAVE_UN	RW	0
7:3	RESERVED	Reserved	R	0x00
2:0	PRIORITY	MPU_MA priority value	RW	0x4

4.4.13 MPU_MA_LSM Registers

4.4.13.1 MPU_MA_LSM Register Summary

Table 4-79. MPU_MA_LSM Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_MA_LSM Physical Address
RESERVED	R	32	0x0000 0010	0x482A F010
MA_LISA_LOCK	RW	32	0x0000 001C	0x482A F01C
MA_LISA_MAP_i [(1)]	RW	32	0x0000 0040 + (0x4 * i)	0x482A F040 + (0x4 * i)

(1) i = 0 to 3

4.4.13.2 MPU_MA_LSM Register Description

For the description of the MPU_MA_LSM registers, see Section 15.2, *Dynamic Memory Manager*, where MA_LISA_LOCK corresponds to the DMM_LISA_LOCK register description, and MA_LISA_MAP_i corresponds to the DMM_LISA_MAP_i register description.

4.4.14 MPU_MA_WP Registers

4.4.14.1 MPU_MA_WP Register Summary

Table 4-80. MPU_MA_WP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_MA_WP Physical Address
DBG_HWWP_CAP	R	32	0x0000 0000	0x482A F200
TRIG_CTRL	RW	32	0x0000 0004	0x482A F204
DBG_HWWP0_LW_ADDR0	RW	32	0x0000 0008	0x482A F208
DBG_HWWP0_HG_ADDR0	RW	32	0x0000 000C	0x482A F20C
DBG_HWWP0_MAIN_CNTL	RW	32	0x0000 0010	0x482A F210
DBG_HWWP0_AUX_CNTL	RW	32	0x0000 0014	0x482A F214
DBG_HWWP0_MEM_CNTL	RW	32	0x0000 0018	0x482A F218
DBG_HWWP0_CHAIN_CNTL	RW	32	0x0000 001C	0x482A F21C
DBG_HWWP0_LW_ADDR0_LOG	R	32	0x0000 0020	0x482A F220
DBG_HWWP0_HG_ADDR0_LOG	R	32	0x0000 0024	0x482A F224
DBG_HWWP0_DATA0_LOG	R	32	0x0000 0028	0x482A F228
DBG_HWWP0_DATA1_LOG	R	32	0x0000 002C	0x482A F22C
DBG_HWWP0_DATA2_LOG	R	32	0x0000 0030	0x482A F230
DBG_HWWP0_DATA3_LOG	R	32	0x0000 0034	0x482A F234
DBG_HWWP0_TRANS_ATTR0_LOG	R	32	0x0000 0038	0x482A F238
DBG_HWWP0_TRANS_ATTR1_LOG	R	32	0x0000 003C	0x482A F23C
DBG_HWWP0_DATA_TRANS_ATTR0_LOG	R	32	0x0000 0040	0x482A F240

Note

The user is required to set all trigger options and match criteria *before* enabling the watchpoint via setting the DBG_HWWP0_MAIN_CNTL[0] WP_EN bit. To change any match options or to re-enable the trigger, the WP_EN must first be cleared. The _LOG registers should normally be read only after the DBG_HWWP0_MAIN_CNTL[31] TRIG bit is verified to be '1'.

4.4.14.2 MPU_MA_WP Register Description

Table 4-81. DBG_HWWP_CAP

Address Offset	0x0000 0000		
Physical Address	0x482A F200	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Capabilities Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	HWWP_MEM_CHAIN_REG_PRESENT	HWWP_TRANS_ATTR1_REG_PRESENT	HWWP_TRANS_ATTR0_REG_PRESENT	HWWP_AUX_CNTL_REG_PRESENT	RESERVED	DATA_WIDTH	RESERVED	ADDR_WIDTH	NUM_WP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	HWWP_MEM_CHAIN_REG_PRESENT	Memory Barrier Chain Control Register implementation 0x0: Not present 0x1: Present	R	0x1
14	HWWP_TRANS_ATTR1_REG_PRESENT	Transaction Attribute 1 Register implementation 0x0: Not present 0x1: Present	R	0x1
13	HWWP_TRANS_ATTR0_REG_PRESENT	Transaction Attribute 0 Register implementation 0x0: Not present 0x1: Present	R	0x1
12	HWWP_AUX_CNTL_REG_PRESENT	Auxillary Control Register implementation 0x0: Not present 0x1: Present	R	0x1
11	RESERVED	Reserved	R	0x0
10:8	DATA_WIDTH	Data Bus Width 0x0: 8 bits 0x1: 16 bits 0x2: 32 bits 0x3: 64 bits 0x4: 128 bits All other values: Reserved	R	0x4
7	RESERVED	Reserved	R	0x0
6:4	ADDR_WIDTH	Address Bus Width 0x0: 8 bits 0x1: 16 bits 0x2: 24 bits 0x3: 32 bits 0x4: 36 bits 0x5: 40 bits 0x6: 64 bits 0x7: Reserved	R	0x5
3:0	NUM_WP	Number of Watchpoints supported (0-15)	R	0x1

Table 4-82. Register Call Summary for Register DBG_HWWP_CAP

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [0]

Table 4-83. TRIG_CTRL

Address Offset	0x0000 0004		
Physical Address	0x482A F204	**Instance**	MPU_MA_WP
Description	Trigger Control Register		

Table 4-83. TRIG_CTRL (continued)

Type	RW

31–1	0
RESERVED	TRIG_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	TRIG_EN	0x0: Trigger disabled. Trigger output (MA_WP_TRIGGER) will not fire 0x1: Trigger enabled. Trigger output (MA_WP_TRIGGER) will fire	RW	0x0

Table 4-84. Register Call Summary for Register TRIG_CTRL

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [1]

Table 4-85. DBG_HWWP0_LW_ADDR0

Address Offset	0x0000 0008		
Physical Address	0x482A F208	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Addr0 Register (lower order bits 31:0). This register should be written only when WP_EN=0.		
Type	RW		

31–0
LOWER_ORDER_WP_ADDR

Bits	Field Name	Description	Type	Reset
31:0	LOWER_ORDER_WP_ADDR	The byte-addressable lower order AXI-4 physical watchpoint address to monitor	RW	0x0000 0000

Table 4-86. Register Call Summary for Register DBG_HWWP0_LW_ADDR0

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [0]

Table 4-87. DBG_HWWP0_HG_ADDR0

Address Offset	0x0000 000C		
Physical Address	0x482A F20C	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Addr0 Register (higher order bits 39:32). This register should be written only when WP_EN=0.		
Type	RW		

31–8	7–0
RESERVED	HIGHER_ORDER_WP_ADDR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x00 0000

Bits	Field Name	Description	Type	Reset
7:0	HIGHER_ORDER_WP_ADDR	The byte-addressable higher order AXI-4 physical watchpoint address to monitor	RW	0x00

Table 4-88. Register Call Summary for Register DBG_HWWP0_HG_ADDR0

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [0]

Table 4-89. DBG_HWWP0_MAIN_CNTL

Address Offset	0x0000 0010		
Physical Address	0x482A F210	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Main Control Register		
Type	RW		

31	30:24	23:20	19:17	16	15:14	13:12	11	10:5	4	3:1	0
TRIG	RESERVED	BEAT_SEL	RESERVED	RESERVED	SUPERVISOR_USER_ACCESS	SECURE_ACCESS	RESERVED	WP_ADDR_MASK	WP_MATCH_CRITERIA	WP_LS_ACCESS	WP_EN

Bits	Field Name	Description	Type	Reset
31	TRIG	Watchpoint trigger 0x0: Watchpoint not triggered 0x1: Watchpoint has triggered (Reset upon 0->1 transition of WP_EN)	R	0x0
30:24	RESERVED	Reserved	R	0x00
23:20	BEAT_SEL	Beat Select (This parameter decides upon for which beat of the burst the data byte lanes should be captured data)	RW	0x0
19:17	RESERVED	Reserved	R	0x0
16	RESERVED	Reserved	R	0x0
15:14	SUPERVISOR_USER_ACCESS	Supervisor/User access 0x0: Reserved 0x1: User 0x2: Supervisor 0x3: No preference	RW	0x3
13:12	SECURE_ACCESS	Secure/Non-secure access 0x0: Reserved 0x1: Non-secure 0x2: Secure. Not supported on GP device 0x3: No preference	RW	0x3
11	RESERVED	Reserved	R	0x0
10:5	WP_ADDR_MASK	Watchpoint address mask (bits to ignore) 0x0: Ignore address bit 0 ..... 0x27: Ignore address bit 39 0x28 – 0x3F: Reserved	RW	0x00
4	WP_MATCH_CRITERIA	Watchpoint match criteria 0x0: Match if access within address range to include MIN and MAX 0x1: Match if access outside address range	RW	0x0

Bits	Field Name	Description	Type	Reset
3:1	WP_LS_ACCESS	Watchpoint Load/Store access 0x0: (Load) Load exclusive or swap 0x1: (Store) Store exclusive or swap (non-posted) 0x2: (Store) Store exclusive or swap (posted) 0x3: Any type of store 0x4, 0x5, 0x6: Reserved 0x7: No preference (valid only if CHAIN_WP_EN=0; otherwise, reserved) Note: In the case of CHAIN_WP_EN=1, both data and memory barrier watchpoints must have the same transaction type; that is, both must be read or both must be write	RW	0x7
0	WP_EN	Watchpoint enable 0x0: Disable the watchpoint 0x1: Enable the watchpoint	RW	0x0

Table 4-90. Register Call Summary for Register DBG_HWWP0_MAIN_CNTL

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0] [1] [2] [3] [4] [5] [6] [7] [8]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [9] [10] [11]

Table 4-91. DBG_HWWP0_AUX_CNTL

Address Offset	0x0000 0014		
Physical Address	0x482A F214	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Auxilliary Control Register. This register should be written only when WP_EN=0.		
Type	RW		

Bits	Field
31:16	RESERVED
15:14	MA_SPLIT_TARG
13:7	RESERVED
6:4	INITIATOR_ID
3:2	RESERVED
1:0	ACCESS_TYPE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:14	MA_SPLIT_TARG	MA splitter target 0x0: Reserved 0x1: AXI2OCP bridge 0x2: EMIF 0x3: No preference	RW	0x3
13:7	RESERVED	Reserved	R	0x00
6:4	INITIATOR_ID	Initiator ID 0x0: CPU_0 0x1: CPU_1 0x2: CPU_2. Not supported on this device 0x3: CPU_3. Not supported on this device 0x4: Unknown source (ACP, FEQ, etc) 0x5: CMU. Not supported on this device 0x6: Reserved 0x7: No preference	RW	0x7
3:2	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
1:0	ACCESS_TYPE	Access type 0x0: Reserved 0x1: Instructions 0x2: Data/others 0x3: No preference	RW	0x3

Table 4-92. Register Call Summary for Register DBG_HWWP0_AUX_CNTL

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0] [1] [2]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [3]

Table 4-93. DBG_HWWP0_MEM_CNTL

Address Offset	0x0000 0018		
Physical Address	0x482A F218	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Memory Barrier Control Register. This register should be written only when WP_EN=0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM_BAR_TRIG	RESERVED																										MEM_BAR_ACCESS_TYPE		MEM_BAR_TYPE		MEM_BAR_WP_EN

Bits	Field Name	Description	Type	Reset
31	MEM_BAR_TRIG	Memory barrier trigger 0x0: Memory Barrier Watchpoint not triggered 0x1: Memory Barrier Watchpoint has triggered (Reset upon 0->1 transition of MEM_BAR_WP_EN)	R	0x0
30:5	RESERVED	Reserved	R	0x000 0000
4:3	MEM_BAR_ACCESS_TYPE	Type of memory barrier access 0x0: Reserved 0x1: Read 0x2: Write 0x3: Don't care (only if CHAIN_WP_EN=0; otherwise, reserved) Note: In the case of CHAIN_WP_EN=1, both memory barrier and data watchpoint must have the same transaction types; that is, both must be read or both must be write	RW	0x3
2:1	MEM_BAR_TYPE	Memory barrier type 0x0: Reserved 0x1: DSB 0x2: DMB 0x3: No preference	RW	0x3
0	MEM_BAR_WP_EN	Memory barrier watchpoint enable 0x0: Disable the watchpoint 0x1: Enable the watchpoint	RW	0x0

Table 4-94. Register Call Summary for Register DBG_HWWP0_MEM_CNTL

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0] [1] [2] [3] [4] [5] [6] [7]

Table 4-94. Register Call Summary for Register DBG_HWWP0_MEM_CNTL (continued)

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [8]

Table 4-95. DBG_HWWP0_CHAIN_CNTL

Address Offset	0x0000 001C		
Physical Address	0x482A F21C	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Data/Memory Barrier Chain Control Register. This register should be written only when WP_EN=0.		
Type	RW		

31	30:2	1	0
CHAIN_WP_TRIG	RESERVED	CHAIN_TYPE	CHAIN_WP_EN

Bits	Field Name	Description	Type	Reset
31	CHAIN_WP_TRIG	Chained watchpoints (memory barrier and data watchpoint) trigger 0x0: Chained Watchpoints not triggered 0x1: Chained Watchpoints have triggered (Reset upon 0->1 transition of CHAIN_WP_EN)	R	0x0
30:2	RESERVED	Reserved	R	0x0000 0000
1	CHAIN_TYPE	Chain type 0x0: Watchpoint match then memory barrier match 0x1: Memory barrier match then watchpoint match	RW	0x0
0	CHAIN_WP_EN	Chained watchpoints (memory barrier and data watchpoint) enable 0x0: Disable the chained watchpoints 0x1: Enable the chained watchpoints Note: Both the memory barrier and data watchpoint should be enabled subsequent to this to avoid partial match/race conditions	RW	0x0

Table 4-96. Register Call Summary for Register DBG_HWWP0_CHAIN_CNTL

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0] [1] [2] [3] [4] [5]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [6]

Table 4-97. DBG_HWWP0_LW_ADDR0_LOG

Address Offset	0x0000 0020		
Physical Address	0x482A F220	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Addr0 Log Register (lower order bits 31:0). This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31:0
WP_ADDR_LOWER_ORDER_BITS

Bits	Field Name	Description	Type	Reset
31:0	WP_ADDR_LOWER_ORDER_BITS	Watchpoint address lower order bits (bits 31:0) (The byte-addressable lower order AXI-4 physical watchpoint address bits which results in a match)	R	0x0000 0000

Table 4-98. Register Call Summary for Register DBG_HWWP0_LW_ADDR0_LOG

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [1]

Table 4-99. DBG_HWWP0_HG_ADDR0_LOG

Address Offset	0x0000 0024		
Physical Address	0x482A F224	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Addr0 Log Register (higher order bits 39:32). This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31–8	7–0
RESERVED	WP_ADDR_HIGHER_ORDER_BITS

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x00 0000
7:0	WP_ADDR_HIGHER_ORDER_BITS	Watchpoint address higher order bits (bits 39:32) (The byte-addressable higher order AXI-4 physical watchpoint address bits which results in a match)	R	0x00

Table 4-100. Register Call Summary for Register DBG_HWWP0_HG_ADDR0_LOG

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [1]

Table 4-101. DBG_HWWP0_DATA0_LOG

Address Offset	0x0000 0028		
Physical Address	0x482A F228	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Data Log Register (bits 31:0). This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31–0
DATA0_CAPTURE

Bits	Field Name	Description	Type	Reset
31:0	DATA0_CAPTURE	Data capture (bits 31:0) (32-bit data associated with the access which results in a watchpoint match)	R	0x0000 0000

Table 4-102. Register Call Summary for Register DBG_HWWP0_DATA0_LOG

Dual Cortex-A15 MPU Subsystem Functional Description
- MPU_MA Watchpoint: [0]

Dual Cortex-A15 MPU Subsystem Register Manual
- MPU_MA_WP Register Summary: [1]

Table 4-103. DBG_HWWP0_DATA1_LOG

Address Offset	0x0000 002C		
Physical Address	0x482A F22C	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Data Log Register (bits 63:32). This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1_CAPTURE																															

Bits	Field Name	Description	Type	Reset
31:0	DATA1_CAPTURE	Data capture (bits 63:32) (32-bit data associated with the access which results in a watchpoint match)	R	0x0000 0000

Table 4-104. Register Call Summary for Register DBG_HWWP0_DATA1_LOG

Dual Cortex-A15 MPU Subsystem Functional Description
- MPU_MA Watchpoint: [0]

Dual Cortex-A15 MPU Subsystem Register Manual
- MPU_MA_WP Register Summary: [1]

Table 4-105. DBG_HWWP0_DATA2_LOG

Address Offset	0x0000 0030		
Physical Address	0x482A F230	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Data Log Register (bits 95:64). This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA2_CAPTURE																															

Bits	Field Name	Description	Type	Reset
31:0	DATA2_CAPTURE	Data capture (bits 95:64) (32-bit data associated with the access which results in a watchpoint match)	R	0x0000 0000

Table 4-106. Register Call Summary for Register DBG_HWWP0_DATA2_LOG

Dual Cortex-A15 MPU Subsystem Register Manual
- MPU_MA_WP Register Summary: [0]

Table 4-107. DBG_HWWP0_DATA3_LOG

Address Offset	0x0000 0034		
Physical Address	0x482A F234	**Instance**	MPU_MA_WP

Table 4-107. DBG_HWWP0_DATA3_LOG (continued)

Description	Debug Watchpoint Data Log Register (bits 127:96). This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA3_CAPTURE																															

Bits	Field Name	Description	Type	Reset
31:0	DATA3_CAPTURE	Data capture (bits 127:96) (32-bit data associated with the access which results in a watchpoint match)	R	0x0000 0000

Table 4-108. Register Call Summary for Register DBG_HWWP0_DATA3_LOG

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [0]

Table 4-109. DBG_HWWP0_TRANS_ATTR0_LOG

Address Offset	0x0000 0038		
Physical Address	0x482A F238	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Transaction Attributes 0 Log Register. This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						RESP_INFO		RESERVED	INIT_INFO			RESERVED	TARGET_INFO			RESERVED			TRANS_TYPE			BURST_LENGTH						RESERVED	BURST_TYPE		

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x00
25:24	RESP_INFO	Response info 0x0: Reserved 0x1: Okay 0x2: Request failed 0x3: Request error	R	0x0
23	RESERVED	Reserved	R	0x0
22:20	INIT_INFO	Initiator info 0x0: CPU_0 0x1: CPU_1 0x2: CPU_2. Not supported on this device 0x3: CPU_3. Not supported on this device 0x4: Unknown source (ACP, FEQ, etc) 0x5: CMU. Not supported on this device 0x6, 0x7: Reserved	R	0x0
19	RESERVED	Reserved	R	0x0
18:16	TARGET_INFO	Target info 0x0: AXI2OCP 0x1: EMIF All other values: Reserved	R	0x0
15:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12:10	TRANS_TYPE	Transaction type (The type of transaction which results in a watchpoint match and is protocol independent. Not all protocols support all transaction types) 0x0: Reserved 0x1: Write posted 0x2: Read 0x3: Read exclusive 0x4: Read linked 0x5: Write non-posted 0x6: Write conditional 0x7: Broadcast	R	0x0
9:4	BURST_LENGTH	Burst length (The length of the burst which results in a watchpoint match) 0x1: Burst length = 1 (min value) 0x2: Burst length = 2 ..... 0x3F: Burst length = 63 (max value)	R	0x00
3	RESERVED	Reserved	R	0x0
2:0	BURST_TYPE	Burst type 0x0: Incrementing 0x1: Wrapping 0x3: Fixed (streaming) All other values: Reserved	R	0x0

Table 4-110. Register Call Summary for Register DBG_HWWP0_TRANS_ATTR0_LOG

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0] [1] [2] [3] [4] [5]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [6]

Table 4-111. DBG_HWWP0_TRANS_ATTR1_LOG

Address Offset	0x0000 003C		
Physical Address	0x482A F23C	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Transaction Attributes 1 Log Register. This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31:3	2	1	0
RESERVED	DATA	SUPERVISOR	SECURE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	DATA	Data access/Instruction fetch 0x0: Other, data, PLE, eviction 0x1: Instruction	R	0x0
1	SUPERVISOR	Supervisor/User access 0x0: User 0x1: Supervisor	R	0x0

Bits	Field Name	Description	Type	Reset
0	SECURE	Secure/Non-secure access 0x0: Non-secure 0x1: Secure. Not supported on GP device	R	0x0

Table 4-112. Register Call Summary for Register DBG_HWWP0_TRANS_ATTR1_LOG

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0] [1]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [2]

Table 4-113. DBG_HWWP0_DATA_TRANS_ATTR0_LOG

Address Offset	0x0000 0040		
Physical Address	0x482A F240	**Instance**	MPU_MA_WP
Description	Debug Watchpoint Data Transaction Attributes 0 Log Register. This register should be read only when TRIG=1 or WP_EN=0. This register is reset upon 0->1 transition of WP_EN.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BYTE_EN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	BYTE_EN	Byte enable (Byte enables for the 128-bit of data captured for the transaction match)	R	0x0000

Table 4-114. Register Call Summary for Register DBG_HWWP0_DATA_TRANS_ATTR0_LOG

Dual Cortex-A15 MPU Subsystem Functional Description

- MPU_MA Watchpoint: [0]

Dual Cortex-A15 MPU Subsystem Register Manual

- MPU_MA_WP Register Summary: [1]

Chapter 5

DSP Subsystems

This chapter describes the features and functions of the device integrated digital processing subsystems.

Note

Devices may include up to two identical instances of DSP subsystem (DSP1 and DSP2). Refer to the device *Data Manual* for number of DSP instances on a specific device. This chapter describes the two DSP device superset. For devices with a single DSP, the information on DSP2 can be ignored.

5.1 DSP Subsystems Overview.................................... 1349
5.2 DSP Subsystem Integration.................................... 1354
5.3 DSP Subsystems Functional Description.................................... 1360
5.4 DSP Subsystem Register Manual.................................... 1396

5.1 DSP Subsystems Overview

The device includes two identical instances (DSP1 and DSP2) of a digital signal processor (DSP) subsystem, based on the TI's standard TMS320C66x DSP CorePac core.

The TMS320C66x DSP core enhances the TMS320C674x core, which merges the C674x floating point and the C64x+ fixed-point instruction set architectures. The C66x DSP is object-code compatible with the C64x+/C674x DSPs.

For more information on the TMS320C66x core CPU, see the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

Each of the two DSP subsystems integrated in the device includes the following components:

- A TMS320C66x CorePac DSP core that encompasses :
  - L1 program-dedicated (L1P) cacheable memory
  - L1 data-dedicated (L1D) cacheable memory
  - L2 (program and data) cacheable memory
  - Extended Memory Controller (XMC)
  - External Memory Controller (EMC)
  - DSP CorePac located interrupt controller (INTC)
  - DSP CorePac located power-down controller (PDC)
- Dedicated enhanced data memory access engine - EDMA, to transfer data from/to memories and peripherals external to the DSP subsystems and to local DSP memory (most commonly L2 SRAM). The external DMA requests are passed through DSP system level (SYS) wakeup logic, and collected from the DSP1 / DSP2 dedicated outputs of the device DMA Events Crossbar for each of the two subsystems.
- A level 2 (L2) interconnect network (DSP NoC) to allow connectivity between different modules of the subsystem or the remainder of the device via the device L3_MAIN interconnect.
- Two memory management units (on EDMA L2 interconnect and DSP MDMA paths) for accessing the device L3_MAIN interconnect address space
- Dedicated system control logic (DSP_SYSTEM) responsible for power management, clock generation, and connection to the device power, reset, and clock management (PRCM) module

Figure 5-1 is the DSP subsystem top-level architecture.

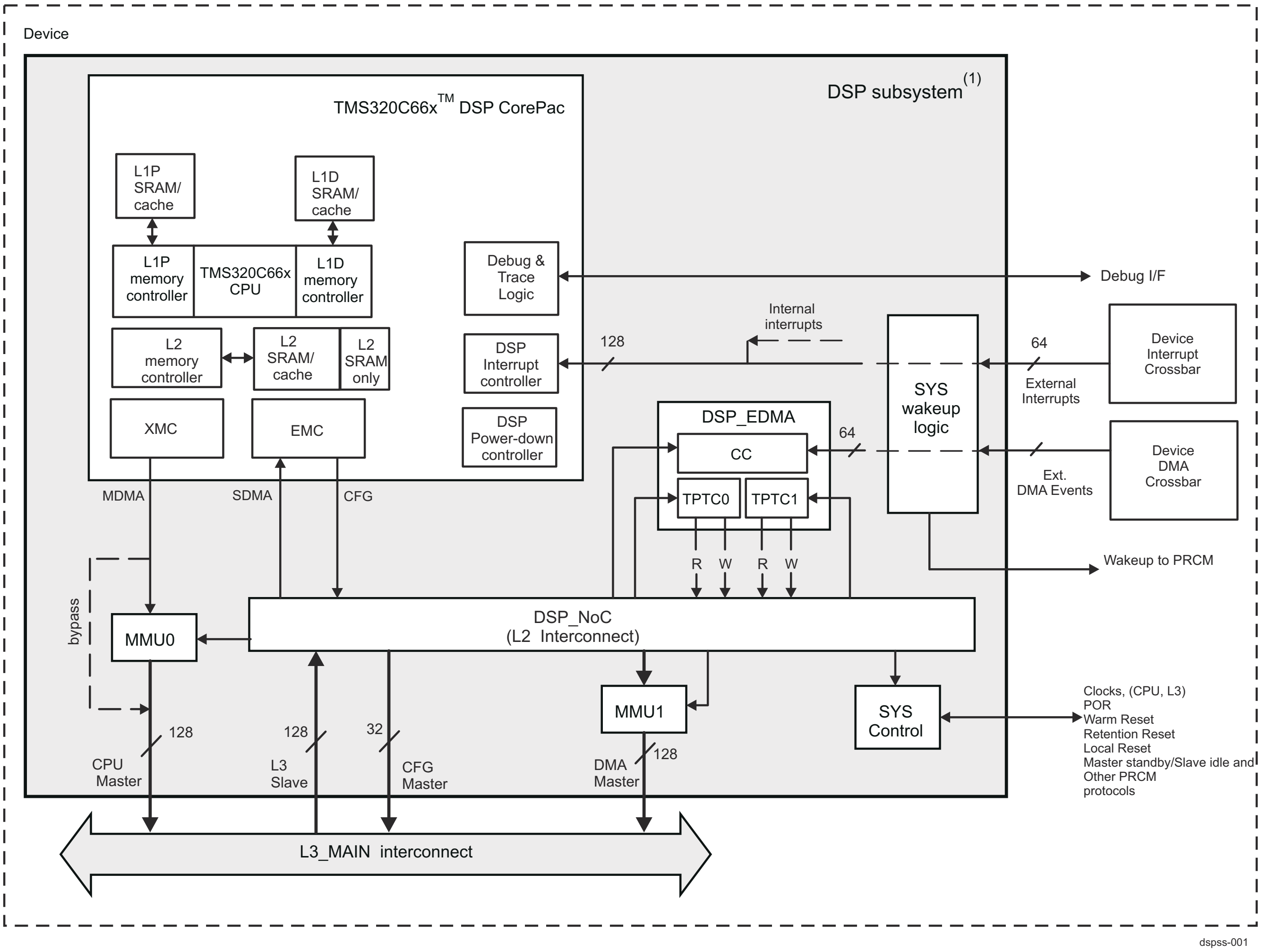


Figure 5-1. DSP Subsystem Highlight

Note

(1) : This diagram shows a single DSP instance. Each device may have **one or two identical DSP instances**, refer to corresponding *device Data Manual* for the specific device support.

5.1.1 DSP Subsystems Key Features

The TMS320C66x Instruction Set Architecture (ISA) is the latest for the C6000 family. As with its predecessors (C64x, C64x+ and C674x), the C66x is an advanced VLIW architecture with 8 functional units (two multiplier units and six arithmetic logic units) that operate in parallel. The C66x CPU has a total of 64 general-purpose 32-bit registers.

Some features of the DSP C6000 family devices are :

- Advanced VLIW CPU with eight functional units (two multipliers and six ALUs) which:
  - Executes up to eight instructions per cycle for up to ten times the performance of typical DSPs
  - Allows designers to develop highly effective RISC-like code for fast development time
- Instruction packing
  - Gives code size equivalence for eight instructions executed serially or in parallel
  - Reduces code size, program fetches, and power consumption
- Conditional execution of most instructions
  - Reduces costly branching

  - Increases parallelism for higher sustained performance
- Efficient code execution on independent functional units
  - Industry's most efficient C compiler on DSP benchmark suite
  - Industry's first assembly optimizer for fast development and improved parallelization
- 8-/16-/32-bit/64-bit data support, providing efficient memory support for a variety of applications
- 40-bit arithmetic options which add extra precision for vocoders and other computationally intensive applications
- Saturation and normalization to provide support for key arithmetic operations
- Field manipulation and instruction extract, set, clear, and bit counting support common operation found in control and data manipulation applications.

The C66x CPU has the following additional features :

- Each multiplier can perform two 16 × 16-bit or four 8 × 8 bit multiplies every clock cycle.
- Quad 8-bit and dual 16-bit instruction set extensions with data flow support
- Support for non-aligned 32-bit (word) and 64-bit (double word) memory accesses
- Special communication-specific instructions have been added to address common operations in error-correcting codes.
- Bit count and rotate hardware extends support for bit-level algorithms.
- Compact instructions: Common instructions (AND, ADD, LD, MPY) have 16-bit versions to reduce code size.
- Protected mode operation: A two-level system of privileged program execution to support higher-capability operating systems and system features such as memory protection.
- Exceptions support for error detection and program redirection to provide robust code execution
- Hardware support for modulo loop operation to reduce code size and allow interrupts during fully-pipelined code
- Each multiplier can perform 32 × 32 bit multiplies
- Additional instructions to support complex multiplies allowing up to eight 16-bit multiply/add/subtracts per clock cycle

The TMS320C66x has the following key improvements to the ISA :

- 4x Multiply Accumulate improvement for both fixed and floating point
- Improvement of the floating point arithmetic
- Enhancement of the vector processing capability for fixed and floating point
- Addition of domain-specific instructions for complex arithmetic and matrix operations

On the C66x ISA, the vector processing capability is improved by extending the width of the SIMD instructions. The C674x DSP supports 2-way SIMD operations for 16-bit data and 4-way SIMD operations for 8-bit data. C66x enhances this capabilities with the addition of SIMD instructions for 32-bit data allowing operation on 128-bit vectors. For example the QMPY32 instruction is able to perform the element to element multiplication between two vectors of four 32-bit data each.

C66x ISA includes a set of specific instructions to handle complex arithmetic and matrix operations.

- **TMS320C66x DSP CorePac memory components :**
  - A 32-KiB L1 program memory (L1P) configurable as cache and / or SRAM:
    - When configured as a cache, the L1P is a 1-way set-associative cache with a 32-byte cache line
    - The DSP CorePac L1P memory controller provides bandwidth management, memory protection, and power-down functions
    - The L1P is capable of cache block and global coherence operations
    - The L1P controller has an Error Detection (ED) mechanism, including necessary SRAM
    - The L1P memory can be fully configured as a cache or SRAM
    - Page size for L1P memory is 2KB
  - A 32-KiB L1 data memory (L1D) configurable as cache and / or SRAM:
    - When configured as a cache, the L1D is a 2-way set-associative cache with a 64-byte cache line
    - The DSP CorePac L1D memory controller provides bandwidth management, memory protection, and power-down functions
    - The L1D memory can be fully configured as a cache or SRAM

    - No support for error correction or detection
    - Page size for L1D memory is 2KB
  - A 288-KiB (program and data) L2 memory, only part of which is cacheable:
    - When configured as a cache, the L2 memory is a 4-way set associative cache with a 128-byte cache line
    - Only 256 KiB of L2 memory can be configured as cache or SRAM
    - 32 KiB of the L2 memory is always mapped as SRAM
    - The L2 memory controller has an Error Correction Code (ECC) and ED mechanism, including necessary SRAM
    - The L2 memory controller supports hardware prefetching and also provides bandwidth management, memory protection, and power-down functions.
    - Page size for L2 memory is 16KB
- The **External Memory Controller (EMC)** is a bridge from the C66x CorePac to the rest of the DSP subsystem and device. It has :
  - a 32-bit configuration port (CFG) providing access to local subsystem resources (like DSP_EDMA, DSP_SYSTEM, etc) or to L3_MAIN resources accessible via the CFG address range.
  - a 128-bit slave-DMA port (SDMA) which provides accesses of system masters outside the DSP subsystem to resources inside the DSP subsystem or C66x DSP CorePac memories, i.e. when the DSP subsystem is the slave in a transaction.
- The Extended Memory Controller (XMC) processes requests from the L2 Cache Controller (which are a result of CPU instruction fetches, load/store commands, cache operations) to device resources via the C66x DSP CorePac 128-bit master DMA (MDMA) port :
  - Memory protection for addresses outside C66x DSP CorePac generated over device L3_MAIN on the MDMA port
  - Prefetch, multi-in-flight requests
- A DSP local **Interrupt Controller (INTC)** in the DSP C66x CorePac, interfaces the system events to the DSP C66x core CPU interrupt and exceptions inputs. Each DSP subsystem C66x CorePac interrupt controller supports up to 128 system events of which 64 interrupts are external to DSP subsystems, collected from the DSP1 /DSP2 dedicated outputs of the device Interrupt Crossbar.
- **Local Enhanced Direct Memory Access (EDMA) controller feaures:**
  - Channel controller (CC) : 64-channel, 128 PaRAM, 2 Queues
  - 2 x Third-party Transfer Controllers (TPTC0 and TPTC1):
    - Each TC has a 128-bit read port and a 128-bit write port
    - 2KiB FIFOs on each TPTC
  - 1-dimensional/2-dimensional (1D/2D) addressing
  - Chaining capability
- **DSP subsystems integrated MMUs**:
  - Two MMUs are integrated:
    - The MMU0 is located between DSP MDMA master port and the device L3_MAIN interconnect and can be optionally bypassed
    - The MMU1 is located between the EDMA master port and the device L3_MAIN interconnect
- A DSP local **Power-Down Controller (PDC)** is responsible to power-down various parts of the DSP C66x CorePac, or the entire DSP C66x CorePac.
- The DSP subsystems **System Control logic** provides:
  - Slave idle and master standby protocols with device PRCM for powerdown
  - OCP Disconnect handshake for init and target busses
  - Asynchronous reset
  - Power-down modes :
    - "Clockstop" mode featuring wake-up on interrupt event. The DMA event wake-up is managed in software.
- The device DSP subsystems are supplied by a PRCM DPLL, but each DSP1/2 **has integrated its own PLL module** outside the C66x CorePac for clock gating and division.

- **Each of the two device DSP subsystems has following port instances** to connect to remaining part of the device. See also Figure 5-1:
  - A 128-bit initiator (DSP MDMA master) port for MDMA/Cache requests
  - A 128-bit initiator (DSP EDMA master) port for EDMA requests
  - A 32-bit initiator (DSP CFG master) port for configuration requests
  - A 128-bit target (DSP slave) port for requests to DSP memories and various peripherals
- **C66x DSP subsystems (DSPSS) safety aspects** :
  - Above mentioned memory ECC/ED mechanisms
  - MMUs enable mapping of only the necessary application space to the processor
  - Memory Protection Units internal to the DSPSS (in L1P,L1D and L2 memory controllers) and external to DSPSS (firewalls) to help define legal accesses and raise exceptions on illegal accesses
  - Exceptions: Memory errors, various DSP errors, MMU errors and some system errors are detected and cause exceptions. The exceptions could be handled by the DSP or by a designated safety processor at the chip level. Note that it may not be possible for the safety processor to completely handle some exceptions

Unsupported features on the C66x DSP core for the device are:

- The Extended Memory Controller MPAX (memory protection and address extension) 36-bit addressing is NOT supported

Known DSP subsystem powermode restrictions for the device are:

- "Full logic / RAM retention" mode featuring wake-up on both interrupt or DMA event (logic in “always on” domain). Only OFF mode is supported by DSP subsystem, **requiring full boot.**

5.2 DSP Subsystem Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 5-2 shows the integration of the DSP subsystem.

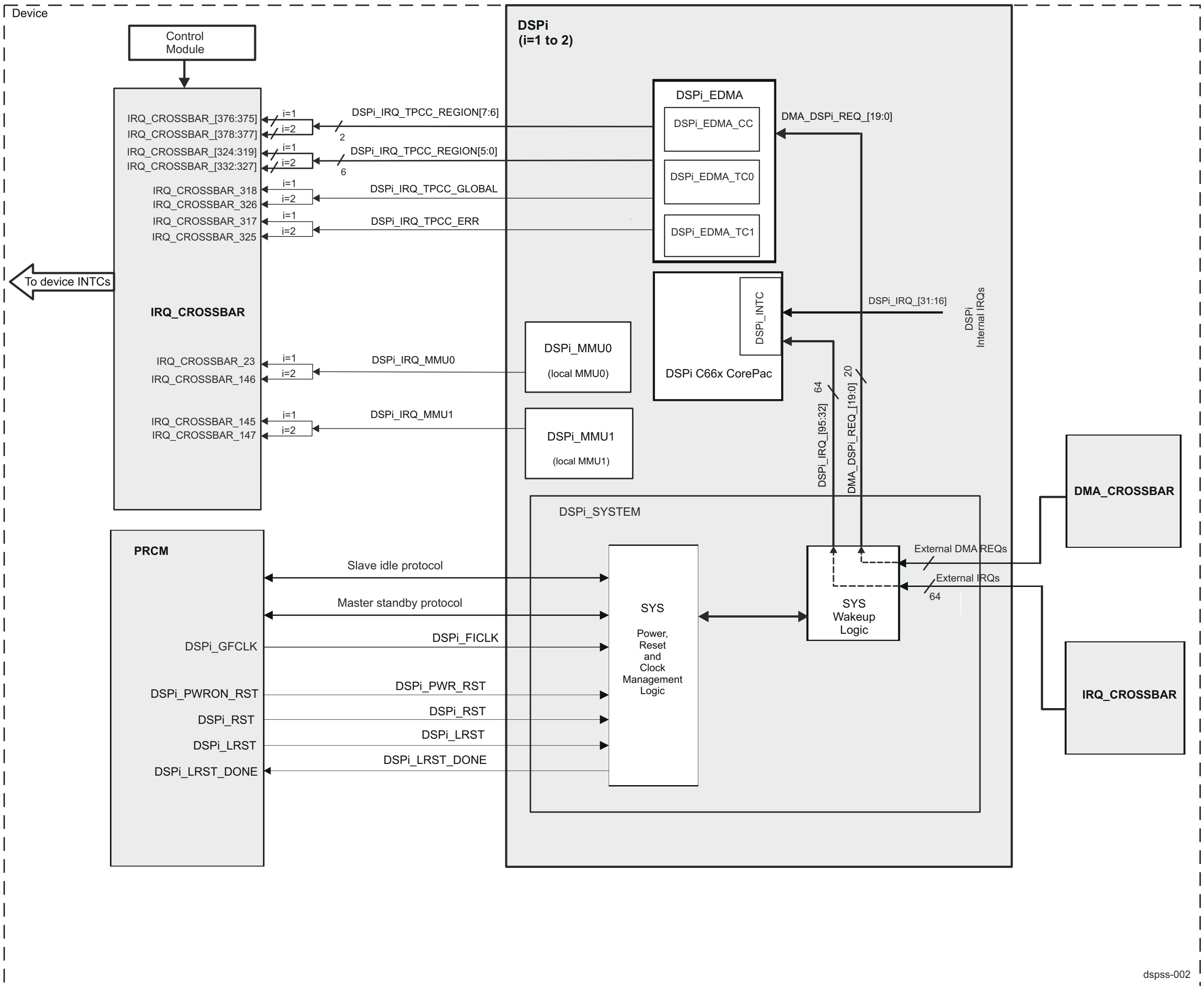


Figure 5-2. DSP Subsystem Integration

Note

For more information about the slave idle protocol and the wake-up request, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

Table 5-1 through Table 5-3 summarize the integration of the module in the device.

Table 5-1. DSP Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
DSP1	PD_DSP1	L3_MAIN

Table 5-1. DSP Integration Attributes (continued)

DSP2	PD_DSP2	L3_MAIN

Table 5-2. DSP Clocks and Resets

Clocks

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DSP1	DSP1_FICLK	DSP1_GFCLK	PRCM module	DSP1 subsystem gateable interface and functional clock.
DSP2	DSP2_FICLK	DSP2_GFCLK	PRCM module	DSP2 subsystem gateable interface and functional clock.

Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DSP1	DSP1_PWR_RST	DSP1_PWRON_RST	PRCM module	For information about PRCM reset sources and distribution, see *PD_DSP1 Description* in *Power, Reset, and Clock Management*. For DSP1 local reset details see also the Section 5.3.3.2.
	DSP1_RST	DSP1_RST	PRCM module	
	DSP1_LRST	DSP1_LRST	PRCM module	
	DSP1_LRST_DONE(1)	DSP1_LRST_DONE	DSP1	
DSP2	DSP2_PWR_RST	DSP2_PWRON_RST	PRCM module	For information about PRCM reset sources and distribution, see *PD_DSP2 Description* in *Power, Reset, and Clock Management*. For DSP2 local reset details see also the Section 5.3.3.2.
	DSP2_RST	DSP2_RST	PRCM module	
	DSP2_LRST	DSP2_LRST	PRCM module	
	DSP2_LRST_DONE(1)	DSP2_LRST_DONE	DSP2	

(1) Destination of this local reset monitoring signal is the PRCM

Note

For information about PRCM clock gating and management, see Section 3.7.2, *PD_DSP1 Description* and *PD_DSP2 Description* in *Power, Reset, and Clock Management*.

The DSP1 / DSP2 generates a number of interrupt requests (IRQs) mapped via the device IRQ_CROSBBAR to other device interrupt controllers (outside DSP subsystem). They are described in Table 5-3.

Table 5-3. DSP Hardware Requests

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
Interrupt Requests				
DSP1	DSP1_IRQ_MMU0	IRQ_CROSSBAR_23	MPU_IRQ_28 DSP1_IRQ_54 DSP2_IRQ_54	Interrupt from the DSP1 subsystem local MMU0 (DSP1_MMU0CFG).
	DSP1_IRQ_MMU1	IRQ_CROSSBAR_145	-	DSP1 subsystem local MMU1 (DSP1_MMUCFG1) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_ERR	IRQ_CROSSBAR_317	-	DSP1 subsystem aggregated ("OR-ed") error interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_GLOBAL	IRQ_CROSSBAR_318	-	DSP1 subsystem EDMA channel controller global interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION0	IRQ_CROSSBAR_319	-	DSP1 subsystem EDMA channel controller REGION0 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION1	IRQ_CROSSBAR_320	-	DSP1 subsystem EDMA channel controller REGION1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION2	IRQ_CROSSBAR_321	-	DSP1 subsystem EDMA channel controller REGION2 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 5-3. DSP Hardware Requests (continued)

	DSP1_IRQ_TPCC_REGION3	IRQ_CROSSBAR_322	-	DSP1 subsystem EDMA channel controller REGION3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION4	IRQ_CROSSBAR_323	-	DSP1 subsystem EDMA channel controller REGION4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION5	IRQ_CROSSBAR_324	-	DSP1 subsystem EDMA channel controller REGION5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION6	IRQ_CROSSBAR_375	-	DSP1 subsystem EDMA channel controller REGION6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION7	IRQ_CROSSBAR_376	-	DSP1 subsystem EDMA channel controller REGION7 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2	DSP2_IRQ_MMU0	IRQ_CROSSBAR_146	-	DSP2 subsystem local MMU0 (DSP2_MMU0CFG) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_MMU1	IRQ_CROSSBAR_147	-	DSP2 subsystem local MMU1 (DSP2_MMUCFG1) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_ERR	IRQ_CROSSBAR_325	-	DSP2 subsystem aggregated ("OR-ed") error interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_GLOBAL	IRQ_CROSSBAR_326	-	DSP2 subsystem EDMA channel controller global interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_REGION0	IRQ_CROSSBAR_327	-	DSP2 subsystem EDMA channel controller REGION0 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 5-3. DSP Hardware Requests (continued)

DSP2_IRQ_TPCC_REGION1	IRQ_CROSSBAR_328	-	DSP2 subsystem EDMA channel controller REGION1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION2	IRQ_CROSSBAR_329	-	DSP2 subsystem EDMA channel controller REGION2 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION3	IRQ_CROSSBAR_330	-	DSP2 subsystem EDMA channel controller REGION3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION4	IRQ_CROSSBAR_331	-	DSP2 subsystem EDMA channel controller REGION4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION5	IRQ_CROSSBAR_332	-	DSP2 subsystem EDMA channel controller REGION5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION6	IRQ_CROSSBAR_377	-	DSP2 subsystem EDMA channel controller REGION6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION7	IRQ_CROSSBAR_378	-	DSP2 subsystem EDMA channel controller REGION7 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Note

The DSP1 / DSP2 does NOT generate any DMA requests towards other device DMA controllers outside DSP1 / DSP2 (EDMA, DMA_SYSTEM, etc.).

Note

The “**Default Mapping**” column in Table 5-3, *DSP Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

- For a description of the interrupt source controls at DSP subsystem level, see Section 5.3.4, *DSP Interrupt Requests*.

DSP1/DSP2 subsystem external interrupt sources: The default interrupt sources mapped by the device IRQ_CROSSBAR to the DSP1 / DSP2 interrupt controller lines are described in the *Interrupt Controllers*. The programmable muxing of various external interrupt sources to the DSP1_INTC.DSP1_IRQ_x and DSP2_INTC.DSP2_IRQ_x input lines (where x=32 to 95) is covered in the Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, of the *Control Module*.

DSP1/DSP2 subsystem internal interrupt sources: The mapping of DSP subsystem internal IRQ sources to DSP1_IRQ_x / DSP2_IRQ_x lines (where x=0 to 31 and x=96 to 127 for DSP subsystem internal event sources) is described in the Section 5.3.4, *DSP Interrupt Requests*.

DSP1/DSP2 subsystem external DMA request sources: The Table 5-6 and Table 5-7 lists the default DSP1 and DSP2 external DMA request sources, respectively, routed via the device DMA_CROSSBAR to the DSP1_EDMA / DSP2_EDMA channel controller inputs (DMA_DSP1_DREQ_i / DMA_DSP2_DREQ_i).

5.3 DSP Subsystems Functional Description

5.3.1 DSP Subsystems Block Diagram

Each of the device two DSP subsystems - DSP1 and DSP2 is composed of a DSP C66x CorePac coupled with several other submodules that enable its integration in the device architecture. Device DSP subsystem provides :

- a 128-bit master data port (MDMA) on device L3_MAIN with a dedicated DSP subsystem local MMU (MMU0) on the path.
- a 32-bit master configuration port (CFG) on device L3_MAIN through which DSP host configures various device located peripherals (external to the DSP subsystem).
- a 128-bit slave DMA port (SDMA) on device L3_MAIN which allows external initiators (masters) to DSP to manipulate some portion of its config / status registers (those which are mapped in the L3_MAIN space) in the device
- a 128-bit master EDMA port - which allows the DSP_EDMA traffic controllers to initiate transfers on L3_MAIN.

The C66x DSP subsystem is illustrated in the Figure 5-1.

5.3.2 DSP Subsystem Components

5.3.2.1 C66x DSP Subsystem Introduction

The key component of the C66x DSP subsystem is built on the TI's high performance TMS320C66x DSP CorePac which consists of a single TMS320C66x CPU (DSP_C0) processor along with a level 1 (L1P and L1D) cacheable SRAM and a level 2 (L2) cacheable SRAM memories interfaced via associated local L1P, L1D and L2 memory controllers, respectively. A DSP C66x CorePac includes also some other internal peripheral components, see Section 5.3.2.2.3 for details.

This chapter provides an overview of the DSP subsystem and the following considerations associated with it :

- DSP C66x CorePac Core and L1 / L2 Memories
- DSP System control and configuration :
  - clock management
  - wake-up event generation
  - interrupt masking
- DSP Booting
- DSP subsystem internal memory and external memory (L3_MAIN) space views
- DSP INTC interrupts mapping, event combining and exception generation
- DMA requests mapping to EDMA channels and EDMA traffic routing
- Others

For more information on the TMS320C66x DSP CorePac, refer to the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C), the *TMS320C66x DSP Cache User Guide*, (SPRUGY8) and the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

5.3.2.2 DSP TMS320C66x CorePac

The TMS320C66x DSP CorePac is illustrated on Figure 5-1. It consists of a single DSP C66x CPU (DSP_C0) processor tightly coupled with level 1 - L1P (program), L1D (data) cacheable SRAM memories and level 2 (L2) cacheable SRAM memories. The C66x CorePac integrated memories are interfaced via associated local L1P, L1D and L2 memory controllers, respectively.

Additionally, the DSP C66x CorePac contains the following internal peripherals:
- an interrupt controller (DSP_INTC) to service DSP C66x CorePac internal and external interrupt events
- a power-down controller (DSP_PDC)
- an external memory controller - DSP_EMC
- an extended memory controller - DSP_XMC_CTRL
- a bandwidth manager - BWM with local controls to the L1P-, L1D- and L2-memories
- an internal direct memory access controller - IDMA

The C66x CorePac DSP also instantiates Debug and Trace logic, part of which is implemented in the DSP core C66x CPU. For more details, refer to the Chapter 33, *On-Chip Debug Support* .

5.3.2.2.1 DSP TMS320C66x CorePac CPU

The DSP C66x CorePac CPU includes following key components :

- A program fetch unit
- 16-/32-bit instruction dispatch unit, advanced instruction packing
- Instruction decode unit
- Two data paths, each with four functional units
- 64 x 32-bit general purpose registers
- Control logic and associated registers
- An internal interrupt and exception controller
- Test, emulation logic
- Internal DMA (IDMA) for transfers between internal memories

For more information on the TMS320C66x central processing unit, see the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

5.3.2.2.2 DSP TMS320C66x CorePac Internal Memory Controllers and Memories

The TMS320C66x DSP CorePac implements a two-level internal cache-based memory architecture.

5.3.2.2.2.1 Level 1 Memories

Level 1 memory (L1) is split into separate program memory (L1P memory) and data memory (L1D memory). Each of the memories can be split into static RAM (normal addressable on-chip memory) and cache.

L1P memory is dedicated to TMS320C66x CPU program words storage and is interfaced via a dedicated L1P memory controller in the DSP C66x CorePac. User can choose to initialize one part of the L1P memory as cache and the other as SRAM. The entire 32 KiB memory is cacheable. The L1P features a dynamically configurable cache size (4 KiB, 8 KiB, 16 KiB and 32 KiB) defined via L1P controller configuration register - L1PCFG[2:0] L1PMODE bitfield. **Note that, the L1P controller maps the cache space by starting at the top of the L1P memory map (i.e. from most significant address) and working downwards. The L1P mapped SRAM size is 32 KiB minus the configured cache size.**

Note

The L1P cache / SRAM is ONLY read-accessible by the C66x CPU processor. The DSP C66x CorePac external DMAs (SDMA and EDMA) and internal DMA (IDMA) are the only initiators which can write to the L1P memory. The CPU may however write access and modify certain L1P cache/ SRAM controller registers if such access is allowed for the register.

Note

In the device integrated DSP, at reset, the entire 32 KiB- L1P memory is initialized as a cache (reset value of L1PMODE=0x7).

For more information on the L1P cache/SRAM memories, refer to the *TMS320C66x DSP Cache User Guide*, (SPRUGY8).

L1D memory is used for level 1 CPU data storage and is interfaced via a dedicated L1D memory controller in the DSP C66x CorePac. User can choose to initialize one part of the L1D memory as cache and the other as SRAM. The entire 32 KiB memory is cacheable. The L1D features a dynamically configurable cache size (4 KiB, 8 KiB, 16 KiB and 32 KiB) defined via L1D configuration register - L1DCFG[2:0] L1DMODE bitfield. **Note that, the L1D controller maps the cache space by starting at the top of the L1D memory map (i.e. from most significant address) and working downwards. The L1D mapped SRAM size is 32 KiB minus the configured cache size.**

Note

In the device integrated DSP, at reset, the entire 32 KiB- L1D memory is initialized as a cache (i.e. reset value of L1DMODE = 0x7).

For more information on the L1D cache/SRAM memories, refer to the *TMS320C66x DSP Cache User Guide*, (SPRUGY8).

5.3.2.2.2.2 Level 2 Memory

The L2 memory can also be split into L2 RAM (normal addressable on-chip memory) and L2-cache for caching external to the DSP meagmodule memory locations. The on-chip integrated L2 memory total size is 288 KiB. The L2 memory is shared between data and program word sources within and outside the DSP C66x CorePac. The L2 memory is divided into two physical 128 bit-wide banks, accesses to which are interleaved on address LSB. Each of the two L2 banks is further split into 4 subbanks.

Note

Only 256 KiB of the L2 memory are cacheable in the device DSP. The remaining 32 KiB are always mapped as static RAM.

The L2 memory features a dynamically configurable cache size (32 KiB, 64 KiB, 128 KiB and 256 KiB) defined via L2 configuration register - L2CFG[2:0] L2MODE bitfield. The additional (to the 32KiB fixed SRAM L2) SRAM available is 256-KiB minus the cache size.

L2 memory controller is responsible for MDMA bus error events reporting. The DSP C66x CorePac MDMA bus error event is exported outside the C66x DSP CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"MDMAERREVT"** event in the Table 5-5.

Note

The MDMA bus error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

5.3.2.2.3 DSP C66x CorePac Internal Peripherals

The DSP C66x CorePac includes the following internal peripherals:

- DSP interrupt controller (DSP_INTC)
- DSP power-down controller (DSP_PDC)
- Bandwidth manager (DSP_BWM)
- Memory Protection Hardware
- Internal DMA (DSP_IDMA) controller
- External Memory Controller (DSP_EMC)
- Extended Memory Controller (DSP_XMC_CTRL) including prefetch buffer logic
- Error Detection logic for the L1P memory
- Error Detection and Correction (ECC) logic for the L2 memory

This section briefly describes the DSP_INTC, DSP_PDC, DSP_BWM, DSP_IDMA, DSP_EMC, DSP_XMC_CTRL controller, L1P Error detection and L2 ECC logic. For more information on the TMS320C66x DSP CorePac, see the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.1 DSP C66x CorePac Interrupt Controller (DSP INTC)

The DSP C66x CorePac includes an interrupt controller (DSP_INTC) and can receive a total of 128 system events as inputs. They include DSP-generated events and chip-level events.

In addition to these 128 events, a non-maskable (NMI) event (see the Section 5.3.4.1.1) and reset events are mapped to the DSP_INTC as well, and are routed straight through to the DSP CPU core.

For more details on the DSP_INTC functionalities and its corresponding control / status registers (part of the DSP_ICFG local configuration space), refer to the section *Interrupt Controller* within the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

For more details on input interrupt mappings and associated IRQ wake-up events, refer to the Section 5.3.4.1.

For more information about the device DSP_INTC, see *Interrupt Controllers*. For more information on chip level IRQ mapping via the device IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

5.3.2.2.3.2 DSP C66x CorePac Power-Down Controller (DSP PDC)

The DSP C66x CorePac includes a power-down controller (PDC). The PDC can power-down all of the following components of the DSP C66x CorePac and internal memories of the DSP subsystem:

- C66x CPU
- L1P Memory
- L2 Memory
- Cache controllers
- Entire TMS320C66x DSP CorePac

Refer to the *Power-Down Controller* in the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

5.3.2.2.3.3 DSP C66x CorePac Bandwidth Manager (BWM)

The DSP C66x CorePac implements a bandwidth manager (BWM) to assure that some requestors do NOT block resources in the C66x CorePac DSP for extended periods of time.

Refer to the *Bandiwdth Management Architecture* in the*TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

5.3.2.2.3.4 DSP C66x CorePac Memory Protection Hardware

The C66x Core Pac memory protection architecture introduces in the DSP a combination of DSP privilege levels and a memory system permission structure. This provides several benefits to the system, as follows:

The DSP C66x CorePac MP events are exported outside the DSP C66x CorePac in DSP subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **memory protection fault** events listed in the Table 5-5.

Note

The memory protection exception events are not exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and associated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

Note

IDMA, DMA or System initiators should not issue read/write requests to regions of DSP L1P, L1D, or L2 memory configured as cache. In such cases the corresponding MPPA register should be set to 0x0 to disallow external read/write accesses.

Refer to the section *Memory Protection* in the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

5.3.2.2.3.5 DSP C66x CorePac Internal DMA (IDMA) Controller

The IDMA controller performs fast block transfers between any two memory locations local to the DSP C66x CorePac. Local memory locations are defined as those in Level 1 program (L1P), Level 1 data (L1D), and Level 2 (L2) memories, or in the external peripheral configuration (CFG) port.

The IDMA configuration / status registers themselves are part of the DSP_ICFG and are visible only to C66x CPU.

Note

The IDMA cannot transfer data to or from the internal DSP memory-mapped register space (DSP_ICFG).

The IDMA exception is mapped to the system level ERRINT_IRQ interrupt event. For more details, refer to the Section 5.3.4.2.2 and the Table 5-5.

The DSP C66x CorePac IDMA error event is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"EMC_IDMAERR"** event in the Table 5-5.

Note

The IDMA exception error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

The IDMA is fully described in the section *Internal Direct Memory Access (IDMA) Controller* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.6 DSP C66x CorePac External Memory Controller

The DSP C66x CorePac has an embedded External Memory Controller which acts as a bridge between the DSP C66x CorePac CPU and the remaining part of DSP subsystem. It implements two ports interfacing the DSP C66x CorePac environment:

- An external peripheral 32-bit CFG port which acts as an (DSP CPU cfg) initiator on the DSP_NoC L2 interconnect. It is the root source of all C66x CPU external configuration traffic **(excluding the DSP_ICFG traffic which takes place only within the DSP C66x CorePac)** towards the subsystem.
- An SDMA slave port which is a target on the L2 DSP_NoC interconnect. It generally accepts traffic initiated on DSP_NoC by the:
  - DSP_EDMA_TC0 and DSP_EDMA_TC1
  - DSP external slave port on the L3_MAIN

In summary:

- The CPU CFG port provides access to the memory-mapped registers which control various peripherals and resources within the DSP subsystem, such as the MMUs, DSP_EDMA controllers, DSP system control and wakeup logic, DSP NoC itself, DSP external peripherals, etc.
- The DSP system masters found outside the DSP C66x CorePac such as the DSP_EDMA controllers (TC0 and TC1) or L3_MAIN masters (device MPU, IPU1, IPU2, etc.) access the SDMA slave port to reach resources inside the DSP C66x CorePac. In respect to the SDMA port, the DSP C66x CorePac is the slave in the transaction.

The DSP_EMC controller adds following functionalities to the DSP C66x CorePac:

- Reporting errors related to the C66x CPU external peripheral configuration bus (associated registers)

Note

Regarding PrivID versus AID mapping functionality of the EMC, the C66x DSP CorePac is able to distinguish ONLY "local" vs "external" requests. The device integrated C66x DSP CorePac has the SDMA PrivID input tied-off to a value of 0x0. On DSP C66x CorePac SDMA port, this means that no distinguishing can be made between external requests which come over DSP_NoC interconnect from local DSP_EDMA and those coming from other initiators accesing DSP via the device L3_MAIN interconnect.

This limits the funcitionality of the internal memory protection registers.

The DSP C66x CorePac EMC error event is exported outside the DSP C66x CorePac in the subsystem, and is capable to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **EMC_BUSERR** event in the Table 5-5.

Note

The EMC configuration bus error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

For various DSP_NoC initiator vs target mappings, refer to the Table 5-8.

Note that, there are DSP_NoC pressure (Mflag bus) controls in DSP_SYSTEM logic related to the C66x CPU CFG and DSP_NoC SDMA init traffic. They are described in the Section 5.3.8.3.

The EMC functionalities / registers are fully described in the section *External Memory Controller (EMC)* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.7 DSP C66x CorePac Extended Memory Controller

The DSP C66x CorePac located extended memory controller (DSP_XMC_CTRL) implements a local DMA master port (MDMA) which provides the primary path for C66x CPU and cache requests to the device level memories such as (DDR or L3 SRAM) and peripheral / memory mapped register space. Via some additional logic, including DSP_SYSTEM controls and a local DSP MMU - DSP_MMU0 on the path (with option to bypass), C66x local MDMA port is mapped to the DSP subsystem CPU master port (i.e. MDMA master port of the DSP CPU on L3_MAIN). The DSP C66x Corepac MDMA port is mapped to the DSP Subsystem CPU Master Port (with DSP_MMU0 involved or not involved) to allow fast accesses (DSP_NoC not involved) to the external SDRAM (via the L3_MAIN and DMM) or to L3 SRAM (via the L3_MAIN).

The memory protection settings in MPAX defines types of the memory accesses permitted on various address ranges within DSP C66x CorePac 32-bit address map.

The DSP_XMC_CTRL also instantiates program and data prefetch buffer logic to reduce time during servicing read requests from the L1D, L1P and L2 memory controllers. The aim is to buffer program and data fetches from external L3_MAIN memory locations. While the program prefetch buffer is organized as 4 entry x 32 byte, the data prefetch buffer is organized in 8 slots, with 128 bytes per slot. The DSP_XMC_CTRL prefetch reduces the penalty associated with accesses to the L3_MAIN SDRAM upon L1P, L1D and L2-cache read-misses.

Note

the DSP_XMC_CTRL registers are part of the DSP_ICFG space, hence they are not accessible outside the DSP C66x CorePac (visible only to the C66x CPU).

In summary the DSP_XMC_CTRL provides :

- a master path from the DSP C66x CorePac level 2 cache / SRAM memory to device memory such as SDRAM or L3 SRAM or peripheral / MMR address space.
- memory protection on external address ranges related to L3_MAIN RAMs (via MPAX unit) :
  - 16 user-defined address ranges (MPAX segments) can be used to divide external memory space
- Address translation
- Data and Program prefetch buffer logic
- 12- address candidate buffer that acts as a "stream detection filter"

Note

The device DSP subsystem does NOT use the DSP C66x CorePac multicore shared memory controller (MSMC) port to add more static RAM within subsystem boundaries, i.e. no additional SRAM is available in the DSP except for the L1P, L1D and L2 memories. Only the DSP_XMC_CTRL controller MDMA port on the L3_MAIN interconnect is used to extend DSP available RAM memory via a direct or DSP_MMU0 translated access to the device EMIF DDR memories and OCMC RAMs.

Note

Only the memory protection function of the DSP_XMC_CTRL MPAX unit is intergated and used in the device DSP subsystem. The MDMA port 32-bit to 36-bit address extension function is NOT used in the device DSP because L3_MAIN address bus width is 32-bit. The DSP_MMU0 does NOT perform an address size (32b -> 36b) extension as well.

The XMC functionalities / registers are fully described in the section *Extended Memory Controller (XMC)* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.7.1 XMC MDMA Accesses at DSP System Level

5.3.2.2.3.7.1.1 DSP System MPAX Logic

The default configuration of MPAX registers provides a 32-bit view of system memory on L3_MAIN.

In summary, each MPAX segment (mentioned above) is programmed with a starting virtual base address, segment sizes from 4 GiB down to 4 KiB, replacement address (i.e., physical address); and permission attributes. Provided that DSP_MMU0 can be used to perform address translation, in most cases the replacement address will equal the base address (i.e., virtual == physical from DSP C66x CorePac perespective).

The system level implementation of MPAX logic allows the C66x CPU to change permission without being required to flush the cache.

The C66x CPU subsystem relies on the MPAXn.PERM field to properly configure the permissions for remote address ranges. The MDMA.rperm[6:0] signals are tie-off to a fixed value of 0x7F on the DSP C66x CorePac boundary.

5.3.2.2.3.7.1.2 MDMA Non-Post Override Control

The C66x corepac submits writes denoted as either “cacheable” or non-cacheable. Write accesses that are non-cacheable will be submitted as interconnect (L3_MAIN) non-posted writes; whereas write accesses that are cacheable are submitted as interconnect posted writes. An exception for the cache writes to L3_MAIN is that in the case of a cache block write-back operation (when actual cache evict busrts are actually issued towards L3_MAIN connected memory), a non-posted write is submitted.

Note

In order to provide a safety net for interconnects that may do aggressive reordering, a memory-mapped register SW control is provided - DSP_SYS_BUS_CONFIG[24] NOPOSTOVERRIDE. When set, this results in all write commands being issued as non-posted. This bit defaults to set, and thus the default behavior is for non-posted writes to be used exclusively.

5.3.2.2.3.8 L1P Memory Error Detection Logic

The L1P memory detection logic (no correction is implemented) uses a 4-bit parity per 256-bit location (1-bit parity per 64-bit line quadrant).

The L1P error detection logic features:
- L1P error detection command, status and address controls (registers)
- L1P_ED error detection exception / interrupt to the DSP_INTC upon DMA / IDMA access
- a direct exception event to C66x CPU (DSP_INTC not involved) upon parity error during a program fetch from L1P-cache
- L1P-cache error recovery

The L1P parity error detection event is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"PMC_ED"** event in the Table 5-5.

Note

The L1P error detection event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

For more details on L1P error detection logic, refer to the section *L1P Error Detection*, of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.2.3.9 L2 Memory Error Detection and Correction Logic

The L2 Memory error detection and correction logic (ECC) implements a distance-3 “detect 2, correct 1” Hamming code based error correction / detection algorithm. A 12-bit hamming code per 256-bit is used.

The L2 error detection and correction logic features:
- L2 error detection command, status and address controls (registers)
- L2 EDC enable
- L2 error detection event counter

- 2x L2 EDC exception / interrupts mapped to the DSP_INTC :
  – L2_ED1 = "error corrected" event
  – L2_ED2 = "error-not-corrected" event

The two L2 memory error correction events are exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"UMC_ED1" and "UMC_ED2"** events in the Table 5-5.

Note

The L2 error detection events are not exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

For more details on L2 error detection and correction logic, refer to the section *L2 Error Detection and Correction* of the*TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.3.2.3 DSP Debug and Trace Support

The DSP subsystem offers full support for the native DSP C66x CorePac debug features. This includes Advanced Event Triggering (AET) and Trace.

5.3.2.3.1 DSP Advanced Event Triggering (AET)

AET capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- Hardware Program Breakpoints: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- Data Watchpoints: specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- Counters: count the occurrence of an event or cycles for performance monitoring.
- State Sequencing: allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

5.3.2.3.2 DSP Trace Support

Trace is a debug technology that provides a detailed, historical account of application code execution, timing, and data accesses. Trace collects, compresses, and exports debug information for analysis. Trace works in real-time and does not impact the execution of the system. Trace is supported via Code Composer Studio.

5.3.2.3.3

See also the Chapter 33, *On-Chip Debug Support* .

5.3.3 DSP System Control Logic

The DSP_SYSTEM module controls the following functions:

- Generation of the divided clocks (DSP_CLK2 or DSP_CLK3) to all components of the DSP subsystem
- Synchronization of the DSP divided clocks
- PRCM module power handshaking
- Reset input resynchronization of the active-to-inactive transition to the CD1_CLK clock
- DSP subsystem top level configuraiton registers and its access from the DSP core.

Figure 5-3 highlights the DSP_SYSTEM and its connectivities to the surrounding blocks within the subsystem and in the device.

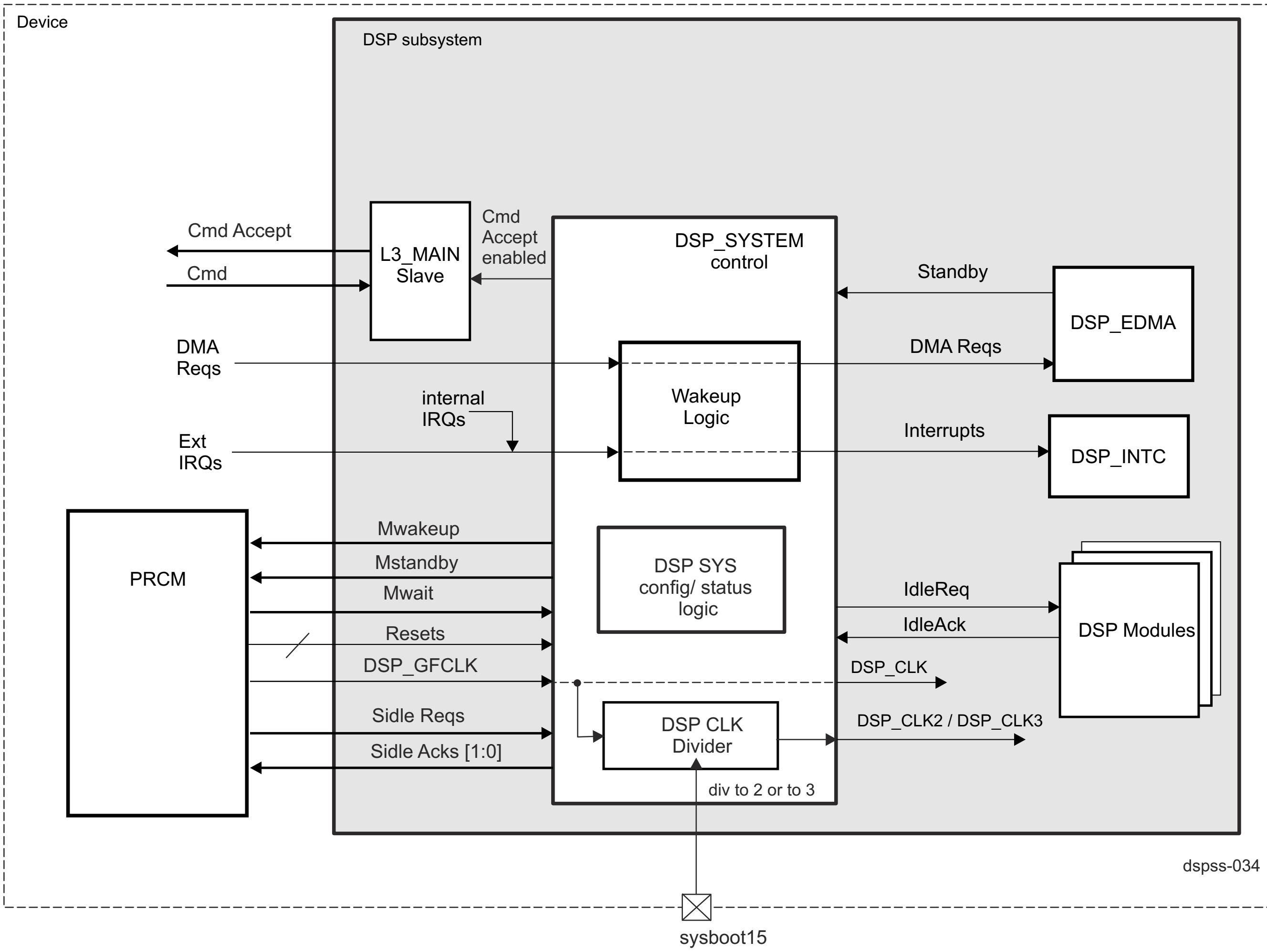


Figure 5-3. DSP_SYSTEM Block Diagram

5.3.3.1 DSP System Clocks

The DSP1 and DSP2 subsystems inputs a primary non-divided clock (DSP1_FICLK / DSP2_FICLK) and based on it (DSP_CLK1), internally generates either **a divided by 2 clock** (DSP_CLK2) version or **a divided by 3 clock** (DSP_CLK3). The divided clock determines the operation rate of the DSP subsystem logic and bus interfaces. The division is defined upon device boot time through signal level externally applied on the device **sysboot15** input. The actual bit configuration is latched upon power-on reset in Control Module register CTRL_CORE_BOOTSTRAP[15] SYS_BOOT_15_CLOCK_DIVIDER boot status bit. For more details, refer to the Section 18.4.6.14.1, *System Boot Status Settings* of the chapter, *Control Module*.

Note

Only DSP_CLK3 clock is supported on this SoC. Upon boot time, sysboot15 set at '1' selects a DSP_CLK3 divided clock version for the DSP subsystem logic and bus interfaces. For SR1.1, sysboot15 must be tied to vdd to select DSP_CLK3, but for SR2.0 it is configurable. For more information, see *Permanent PU/PD disabling (SR 2.0 only)* in *Control Module*.

The clock operating mode setting (DSP_CLK2 or DSP_CLK3) must be static just before and continually after reset deassertion. This signal will also drive the configuration to the DSP C66x CorePac for the XMC_MDMA_CLK, EMC_SDMA_CLK, and EMC_CFG_CLK configurations.

The DSPSS1 / DSPSS2 subsystem input clock frequency (DSP_CLK1) corresponds to the PRCM DSP1_GFCLK / DSP2_GFCLK frequency that is configured in the device PRCM registers.

Note

For valid DSP_CLK1 (and hence for DSP_CLK3 = DSP_CLK1 / 3) frequency range, see the Operating Performance Points section of the device Data Manual.

The Section 5.3.2 also shows the distribution of the different DSP subsystems blocks within the two DSP local clock domains CD0_CLK (running on DSP_CLK frequency) and CD1_CLK (running on DSP_CLK2 or DSP_CLK3 frequency).

5.3.3.2 DSP Hardware Resets

The DSP uses the same reset sources than those mapped to the DSP C66x CorePac; i.e. DSP C66x CorePac reset inputs will be pinned out as DSP system reset inputs.

The Table 5-4 summarizes the DSP hardware reset inputs and their functional descriptions.

Table 5-4. Summary of the DSP1 and DSP2 Hardware Resets

DSP1 reset input	DSP1 reset "done" output to PRCM	Description
DSP1_PWR_RST	-	This is power-on reset signal used inside DSP1 to reset mainly the emulation logic. It resets the entire DSP1 logic.
DSP1_RST	-	Reset signal used to reset all logic inside DSP1 except Emulation logic.
DSP1_LRST	-	Reset applied ONLY to the C66x CPU inside DSP1
-	DSP1_LRST_DONE	Indicates completion of the DSP1 local C66x CPU reset to device PRCM
DSP2 reset input	**DSP2 reset "done" output to PRCM**	**Description**
DSP2_PWR_RST	-	This is power-on reset signal used inside DSP2 to reset mainly the emulation logic. It resets the entire DSP2 logic.
DSP2_RST	-	Reset signal used to reset all logic inside DSP2 except Emulation logic.
DSP2_LRST	-	Reset applied ONLY to the C66x CPU inside DSP2
-	DSP2_LRST_DONE	Indicates completion of the DSP2 local C66x CPU reset to device PRCM

See also the Section 5.2 for more information on the PRCM reset sources to DSP reset inputs connectivity.

Refer to the *DSP1 Subsystem Power-on Reset Sequence* and the *DSP2 Subsystem Power-on Reset Sequence* in the chapter, *Power, Reset and Clock Manamgement* for more details on the DSP1 and DSP2 power-on reset sequence, respectively.

Note

In the case of DSP1 / DSP2 recovery from the “Powerdown-grid off”, a full power-on-reset sequence is required before re-booting and resuming functional operation.

Note

The DSP host (device MPU) software must ensure that the PRCM functional clock DSP1_GFCLK / DSP2_GFCLK is enabled to the DSP1 / DSP2, respectively, prior to starting the DSP1 / DSP2 power-on reset sequence.

5.3.3.3 DSP Software Resets

During a software reset on the DSP, all resets described in Table 5-4 are asserted, except for the power-on DSP_PWRON_RST signal which remains de-asserted in this case.

The DSP subsystem does NOT implement any local software reset controls. The software reset assertion and DSP_LRST completion monitoring is done in PRCM located registers (part of the DSP1_PRM / DSP2_PRM address space).

Refer to the *DSP1 Subsystem Software Warm Reset Sequence* and the *DSP2 Subsystem Software Warm Reset Sequence* in the chapter, *Power, Reset and Clock Manamgement* for more details on the DSP1 and DSP2 software reset sequence and related software controls, respectively.

5.3.3.4 DSP Power Management

The supported power-down modes are:

- Slave idle and master standby protocols for powerdown
- "Disconnect from interconnect" handshake for init and target busses
- Clock Stop mode - wakeup on interrupt or DMA event
- Grid OFF mode : No power supply is switched-on

Note

Powerdown-retention mode is NOT supported by DSP subsystem. **The DSP recovery from the Powerdown-grid OFF mode requires full boot.**

The DSP C66x CorePac natively supports “CLKSTOP/Static Powerdown” and “POWERDOWN (Grid off)” modes of operation. Once the device PRCM restores clocks and power supply, then the DSP C66x CorePac can exit static-powerdown.

The DSP_SYSTEM wakeup logic is implemented in the "always-on" clock / power supply domain. This logic monitors new interrupts / events and will drive the IDLE wakeup request when a new interrupt / event occurs.

5.3.3.4.1 DSP System Powerdown Protocols

For each of the powerdown modes – Static or Powerdown-Grid Off, the device PRCM will control whether clocks are gated, or whether supplies are reduced or removed.

Note

In the case of “Powerdown-grid off”, a full power-on-reset sequence is required before re-booting and resuming functional operation.

In the static powerdown modes - the DSP recognizes new level interrupts while in a clock-gated state (and drive wakeup request). During this powerdown mode all internal state will be retained, including DSP C66x CorePac, interconnect, EDMA, memories, etc.

The following protocols are implemented with PRCM:

- Slave idle protocol with device PRCM for powerdown (wake-up capable)
- Master standby protocol with device PRCM for powerdown
- Interconnect disconnect for master and slave ports

The Master standby and slave idle protocols behaviour is controlled in the DSP_SYS_SYSCONFIG register.

5.3.3.4.2 DSP Software and Hardware Power Down Sequence Overview

Figure 5-4 highlights the high level flow-chart for entry into any of the DSP powerdown modes. The system host (typically) first informs the DSP that it should enter a powerdown mode. The host (g.h. device MPU) sends a software message (normally via system level mailbox+interrupt). In parallel, the PRCM (via host or DSP programmation) will hardware assert an SIdleReq request to the DSP via the IDLE Protocol connection. At the next stage, the C66x CPU, in general, performs any software bookkeeping necessary to transition the DSP subsystem to a quiescent state. This may include : waiting for outstanding DMA transfers to complete, waiting for outstanding DMA transfers to complete, etc. The C66x processor should finally execute the IDLE instruction when it is ready to be powered-down. Assuming the DSP_SYS_SYSCONFIG[5:4] STANDBYMODE is enabled, then the hardware will transition to an idle state and notify to the system the intention to enter powerdown state to the system via the master standby and slave idle protocols. After IDLE and MSTANDBY handshake is completed, the DSP clocks are optionally gated; and supply rails are optionally reduced or turned off.

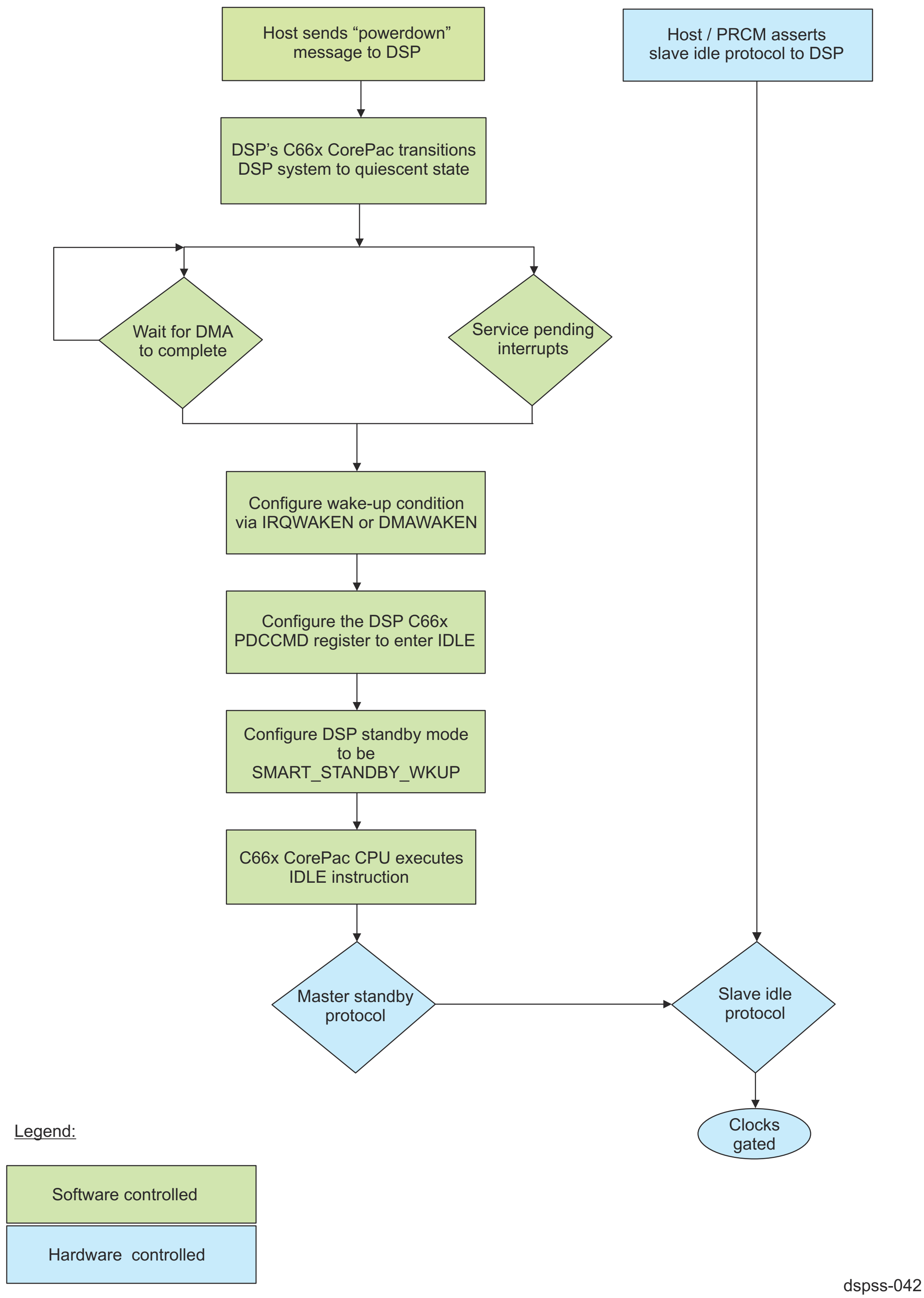


Figure 5-4. Extended Duration Sleep Software and Hardware Sequence

Note

The PM_DSPx_PWRSTCTRL[1:0] POWERSTATE bit field in device PRCM must be set to 0x3 (ON state) prior to performing the sequence shown in Figure 5-4 for the transition to be successful.

5.3.3.4.3 DSP IDLE Wakeup

In order to facilitate auto-wakeup of DSP C66x, the IDLE protocol's wakeup capability is used. Wakeup operation is enabled if DSP_SYS_SYSCONFIG[3:2] IDLEMODE is set to 0x3.

In this mode, while in IDLE state, if an external input interrupt source is asserted (if enabled via the DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 mask) or if an external DMA event source is asserted (if enabled via the DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 mask) or if the DSP subsystem NMI input is asserted **(note that there is no wake enable mask for the non-maskable interrupt)** then the Mwakeup signal is asserted to the PRCM which is expected to observe the Mwakeup. Upon such assertion the PRCM enables the clocks, exiting the "Standby" and "Idle" states. at this point the C66x CPU is able to branch to the pending interrupt service routine. The MWakeup is deasserted when all IRQ or DMA requests enabled in the DSP_SYS_IRQWAKEEN0/1 and DSP_SYS_DMAWAKEEN0/1 are deasserted.

The Wakeup logic controlling assertion of the MWakeup request is completely asynchronous because in IDLE mode the clock may not be present. It relies on level sensitive interrupts.

Note

The DSP_EDMA must be manually removed from IDLE / Standby state. During that time, it is possible that the EDMA input event is no longer pending and may not have been recognized/latched as an EVENT to the EDMA. In that case, the user SW can enable the DSP EDMA_WAKE_INT (in associated DSP_SYS_EDMAWAKE0_IRQENABLE_SET register) to recognize in the (DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW /DSP_SYS_EDMAWAKE0_IRQSTATUS) which specific EDMA event was asserted and caused the wakeup condition. The DSP software can then trigger the corresponding DSP_EDMA channel manually (by setting the ESR) or by servicing the interrupt/event manually via reads and writes. For more details, refer to the Section 5.3.5.1.

5.3.3.4.4 DSP SYSTEM IRQWAKEEN registers

The DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 masking bits must be appropriately set for any valid interrupt mapped from device IRQ_CROSSBAR to the DSP subsystem boundary, to enable its path (passing through the DSP_SYSTEM wakeup logic) to the DSP local interrupt controller - DSP_INTC.

CAUTION

In order for a given interrupt to be serviced by the DSP (even when the Idle Instruction is NOT being executed), the Interrupt must be enabled in the coresponding DSP_SYS_IRQWAKEEN0 or DSP_SYS_IRQWAKEEN1 register.

5.3.3.4.5 DSP Automatic Power Transition

This section provides register details for configuring the DSP1 subsystem in automatic power transition mode. The same should be considered for DSP2 in corresponding PRCM and DSP2 related registers.

The DSP1 module is supposed to be configured to automatic management in PRCM.DSP1_CM_CORE_AON via setting the register CM_DSP1_DSP1_CLKCTRL[1:0] MODULEMODE bitfield to 0x1. The DSP1 clock domain is supposed to be configured in automatic "HW_AUTO" transition (setting bitfield CM_DSP1_CLKSTCTRL[1:0] CLKTRCTRL=0x3).
The power state (controls are in the PRCM.DSP1_PRM instance) to reach upon a sleep transition is configured in the PM_DSP1_PWRSTCTRL[1:0]POWERSTATE bitfield.

5.3.4 DSP Interrupt Requests

The DSP subsystem relies on the DSP C66x CorePac local interrupt controller - DSP_INTC for mapping the various input interrupts to the C66x CPU, that are :

- generated outside the DSP, by the device intergated modules and subsystems
- generated within the DSP subsystems but outside the DSP C66x CorePac
- generated by different components within the DSP C66x CorePac

In addition, a non-maskable input interrupt, direct mapped on a C66x processor NMI input is implemented. It is mapped via a register that resides within the device Control Module. Both the maskable interrupts and the non-maskable interrupts are synchronized internally.

Part of the DSP subsystem module generated interrupts which are output as follows:

- DSP_EDMA interrupts
- DSP_MMU0 and DSP_MMU1 interrupts
- Error interrupts

Figure 5-5 shows how are the interrupt sources organized. To manage and expand the interrupt capabilities of the DSP C66x CorePac (internal and external interrupt requests), the DSP subsystem includes two levels of interrupt control :

- The DSP C66x CorePac local Interrupt controller - DSP_INTC
- The System control logic - DSP_SYSTEM

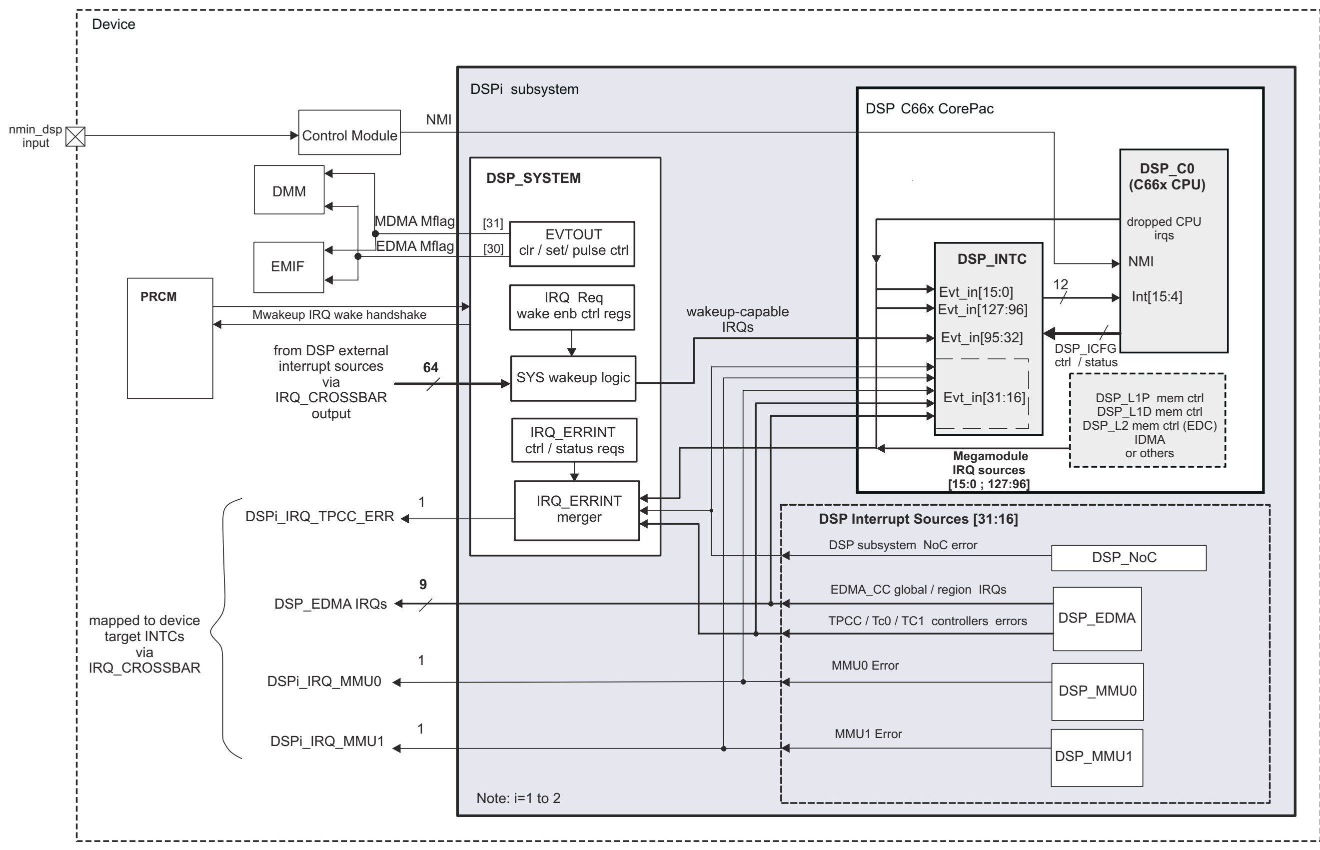


Figure 5-5. DSP Subsystem Interrupt Management

5.3.4.1 DSP Input Interrupts

In summary, the DSP_INTC accepts up to 124 event inputs, and flexibly maps those down to 12 interrupt inputs to the DSP. The mapping can be 1:1 (input:output), or can use the event combiner to map multiple interrupts (within a 32-bit group) to one of the DSP interrupt inputs. In general, many of the 124 interrupt controller inputs are collected within the DSP C66x CorePac, and are NOT available at the DSP C66x CorePac boundaries.

The C66x CPU dropped event is exported outside the C66x CorePac in DSP subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"INTERR" event** listed in the Table 5-5.

Note

The dropped CPU event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

Part of the input interrupts, generated by DSP peripherals that are located outside the DSP C66x CorePac - DSP_EDMA (DSP_EDMA_CC, DSP_EDMA_TC0, DSP_EDMA_TC1) , DSP_MMU0, DSP_MMU1 and DSP_NoC, are also mapped to outputs at the DSP subsystem boundary, such that they can be exported to system hosts (MPU, etc.) via the device IRQ_CROSSBAR.

Of particular interest, MMUs (DSP_MMU0 and DSP_MMU1) interrupts will typically be serviced by the device MPU instead of by the local DSP core.

Any interrupt input at DSP subsystem boundaries (i.e. excluding the DSP subsystem internal IRQ sources that reside in and outside the DSP C66x CorePac) can be used to wake-up the DSP subsystem from an IDLE state. This is described in the Section 5.3.3.4.3 and is controlled by the DSP_SYSTEM logic register DSP_SYS_SYSCONFIG [3:2] IDLEMODE bitfield along with the DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 registers.

CAUTION

The DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 bits MUST be enabled for externally mapped interrupts (DSP_INTC[95:32]) to be serviced by the DSP regardless of the DSP power state (IDLE or non-IDLE).

The DSP C66x CorePac DSP_INTC registers are NOT readable by any entity other than the C66x CPU, because they are part of the DSP_ICFG C66x CorePac internal configuration space (see also the Section 5.3.10). Hereby, only the C66x itself is able to service these interrupt events. The only way for these interrupts to be cleared is for the DSP CPU to clear the state in the EVTFLAGi (where i=0 to 3) register, or via reset assertion.

Note

For cases where the DSP maps an interrupt directly, the DSP is not strictly required to clear the EVTFLAGi register. User software must take the extra step of clearing the EVTFLAGi to cause the corresponding output interrupt to be cleared and re-asserted upon a new input event assertion.

5.3.4.1.1 DSP Non-maskable Interrupt Input

The device DSP also supports a non-maskable interrupt (NMI) directly mapped to the NMI input of the C66x CPU. This line is also mapped to the NMEVT input of the DSP local INTC, and can be used as an exception signal, too. At system level, the NMI interrupt mapping to the DSP_INTC is controlled via the **device core Control Module** register as follows:

- CTRL_CORE_NMI_DESTINATION_2 [15:8] DSP1 = 0x1 enables the DSP1 to receive the NMI coming from the device **nmin_dsp** input.
- CTRL_CORE_NMI_DESTINATION_2 [23:26] DSP2 = 0x1 enables the DSP2 to receive the NMI coming from the device **nmin_dsp** input.

For more details on the NMI receive enable bit mapping, refer to the *Control Module Register Manual* in the chapter, *Control Module*.

5.3.4.2 DSP Event and Interrupt Generation Outputs

5.3.4.2.1 DSP MDMA and DSP EDMA Mflag Event Outputs

The Mflag events generated by DSP subsystem EVTOUT bus are represented in the Figure 5-5.

A couple of the DSP EVTOUT bus outputs - EVTOUT[31] and EVTOUT[30] are used for generation of MFLAGs dedicated to the DSP MDMA and EDMA ports, respectively. DSP MFLAGs are connected directly to DMM and EMIF. The DSP MFLAGs participate in the DMM Emergency and EMIF MFLAG prioritization schemes. At the L3 Level Bandwidth regulators connected to the DSP MDMA and EDMA ports can be used to control DSP traffic versus other device traffic.

The device DSP subsystem is able to generate the 2 output Mflag events via the following DSP_SYSTEM module located registers :

- DSP_SYS_EVTOUT_SET[31:30]
- DSP_SYS_EVTOUT_CLR[31:30]

The current state of the outputs can be detected by reading any of these "pseudo" register bits.

Note

Only EVTOUT[31:30] outputs are implemented in the device, hence only bits [31:30] of the mentioned DSP_SYS_EVTOUT_x registers are used.

The DSP_SYS_EVTOUT_SET register unconditionally drives the corresponding output event to '1'. The DSP_SYS_EVTOUT_CLR register unconditionally drives the corresponding output event to a '0'.

5.3.4.2.2 DSP Aggregated Error Interrupt Output

The aggregated error interrupt of the DSP subsystem is shown in the Figure 5-5.

The subset of those events that correspond to: **DSP C66x CorePac generated error events**, **DSP_EDMA error interrupts** and **L2 DSP_NoC interconnect error interrupt**, is reduced by an OR-schematic to a single ERRINT_IRQ output interrupt which is made available on DSP subsystem boundary. It is expected that one of the DSP system hosts monitors the interrupts/error conditions in safety conscious systems.

Figure 5-6 shows a functional representation of the DSP error interrupt "OR"-reduction logic. In summary, there exists **an unmasked status** (DSP_SYS_ERRINT_IRQSTATUS_RAW) register, two complementary enable bit-vector registers (DSP_SYS_ERRINT_IRQENABLE_SET / DSP_SYS_ERRINT_IRQENABLE_CLR), and a masked status register (DSP_SYS_ERRINT_IRQSTATUS). The ERRINT event is asserted when any enabled error interrupt input is asserted.

Note

The ERRINT_IRQ output can be programmatically mapped as the DSPi_IRQ_TPCC_ERR (where i=1 to 2) interrupt to all device (dsp hosts) interrupt controllers via the device IRQ_CROSSBAR. For more information on the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

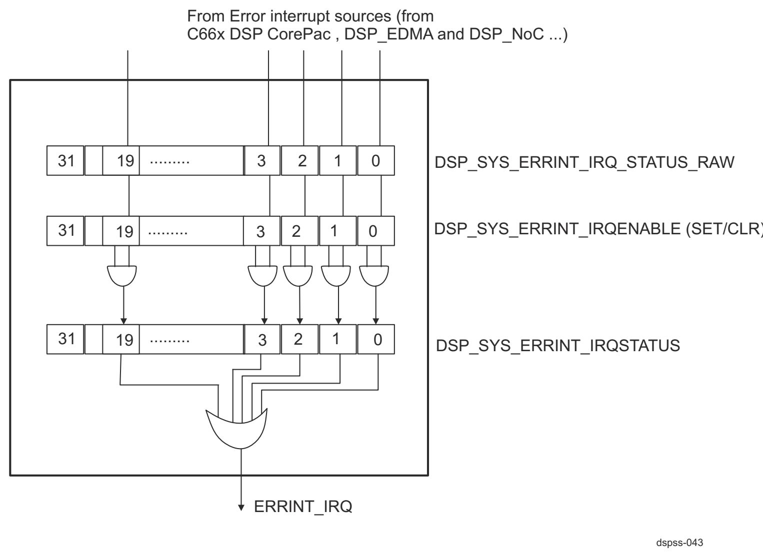


Figure 5-6. ERRINT Diagram

The Table 5-5 details the mapping of error event output sources to the bit positions within the following DSP error event related registers :

- DSP_SYS_ERRINT_IRQSTATUS_RAW
- DSP_SYS_ERRINT_IRQSTATUS
- DSP_SYS_ERRINT_IRQENABLE_SET
- DSP_SYS_ERRINT_IRQENABLE_CLR

Following functional descriptions are valid for the above registers :

- **IRQ status raw register -** This register provides a per-event raw interrupt status vector. The Raw status is set even if the corresponding event is not enabled. Software can write '1' to set the (raw) status for debug purposes.
- **IRQ status register -** This register provides a per-event enabled interrupt status vector. The Enabled status is set if the corresponding event is enabled and the raw status is set. Software can write 1 to clear the (raw) status after the interrupt has been serviced. The clear takes effect even if the interrupt is not enabled.
- **IRQ enable register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to set (i.e., enable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Clear" - register)
- **IRQ clear register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to clear (i.e., disable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Set" register)

Note

A DSP_SYS_ERRINT_IRQSTATUS_RAW bit is set even if the corresponding event is NOT enabled in the DSP_SYS_ERRINT_IRQENABLE_SET.

Table 5-5. DSP ERRINT Interrupt Mapping

Interrupt Number	Name	Description
0	tpcc_errint_level	DSP EDMA CC error interrupt
1	tptc_errint0_level	DSP EDMA TC0 error interrupt
2	tptc_errint1_level	DSP EDMA TC1 error interrupt
3	noc_errint_level	DSP L2 Interconnect (DSP_NoC) error interrupt
4	INTERR	DSP C66x CorePac Dropped CPU Interrupt event

Table 5-5. DSP ERRINT Interrupt Mapping (continued)

Interrupt Number	Name	Description
5	EMC_IDMAERR	DSP C66x CorePac Invalid IDMA Parameters
6	MDMAERREVT	DSP C66x CorePac VbusM Error Event
7	PMC_ED	DSP C66x CorePac Single bit error detected during DMA read
8	UMC_ED1	DSP C66x CorePac Corrected bit error detected
9	UMC_ED2	DSP C66x CorePac Uncorrected bit error detected
10	SYS_CMPA	DSP C66x CorePac CPU memory protection fault
11	PMC_CMPA	DSP C66x CorePac CPU memory protection fault
12	PMC_DMPA	DSP C66x CorePac DMA memory protection fault
13	DMC_CMPA	DSP C66x CorePac CPU memory protection fault
14	DMC_DMPA	DSP C66x CorePac DMA memory protection fault
15	UMC_CMPA	DSP C66x CorePac CPU memory protection fault
16	UMC_DMPA	DSP C66x CorePac DMA memory protection fault
17	EMC_CMPA	DSP C66x CorePac CPU memory protection fault
18	EMC_BUSERR	DSP C66x CorePac Bus Error Interrupt
19	Reserved	-
20	Reserved	-
21	Reserved	-
22	Reserved	-

Note that neither of the events, listed in Table 5-5 , is exported as a separate hardware interrupt off the DSP boundary.

5.3.4.2.3 Non-DSP C66x CorePac Generated Peripheral Interrupt Outputs

The non-DSP C66x CorePac interrupts generated by peripherals within the DSP subsystem are also summarized in the Figure 5-5.

Besides the aggregated error event ERRINT_IRQ interrupts – DSPi_IRQ_TPCC_ERR (where i=1 to 2), interrupts (see also Figure 5-5) **generated individually by DSPSS peripherals located outside the DSP C66x CorePac**, are mapped as separate IRQ outputs at DSP boundaries. They are sourced by the DSP_EDMA_CC, DSP_EDMA_TC0, DSP_EDMA_TC1, DSP_MMU0, DSP_MMU1 and DSP_NoC and exported to other host INTCs via the device IRQ_CROSSBAR. Refer to the Section 5.2, for more information on these DSP interrupt outputs mapping.

5.3.5 DSP DMA Requests

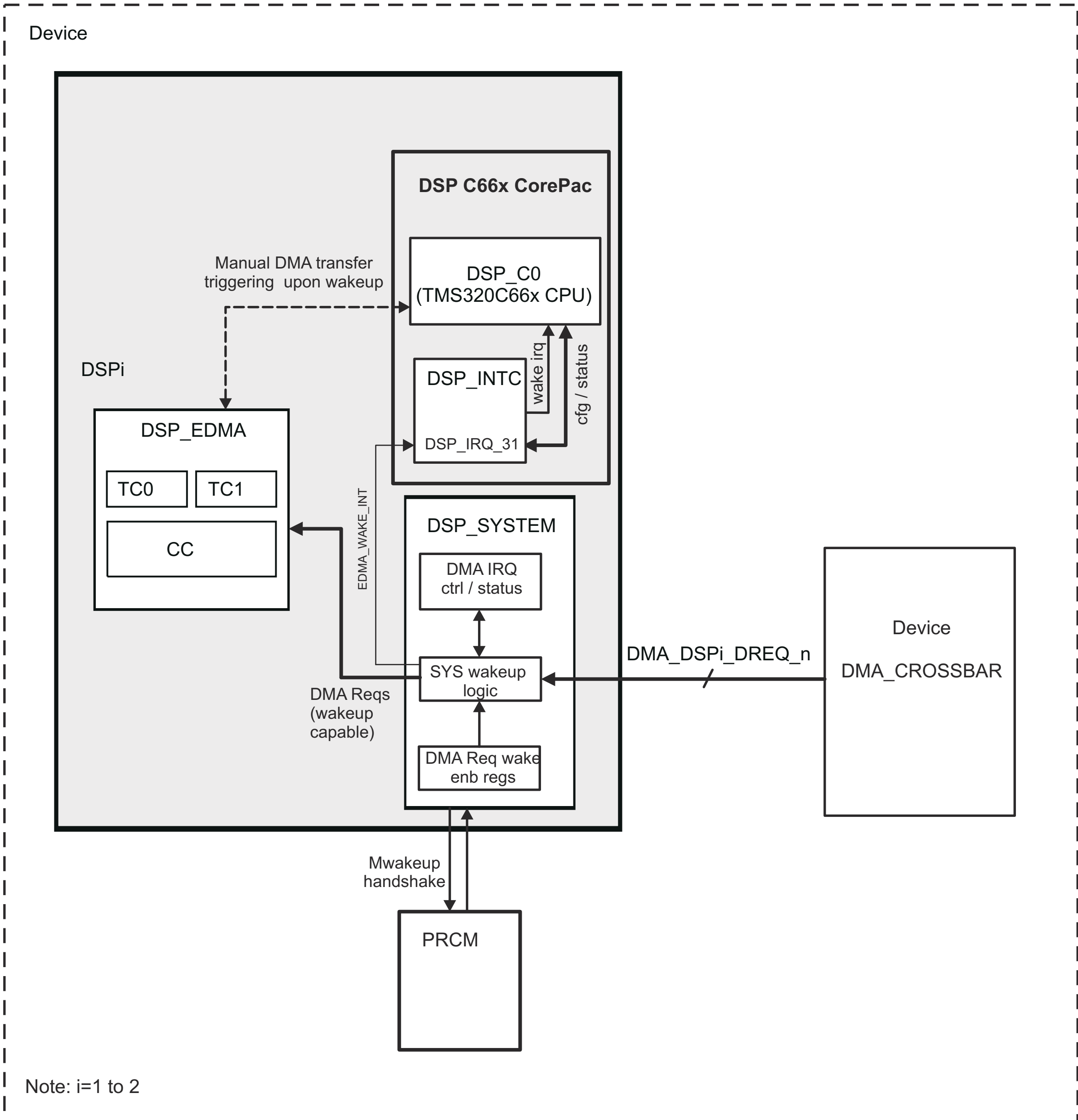
The DSP_EDMA_CC (channel controller) supports 64 hardware event inputs, that can be used to synchronize the 64 DMA channels. These event inputs are provided at the DSP subsystem boundary via the device DMA_CROSSBAR and can be mapped to sources within the device.

The DSP subsystem receives DMA requests from certain peripherals, such as the McASP modules. The DMA requests path through the DSP logic is shown in Figure 5-7.

Similar to the interrupts received at DSP subsystem boundary, the DSP EDMA requests are first routed through the wakeup generation logic of the DSP_SYSTEM module, hence, each DMA request received by the DSP subsystem can wakeup the system from DSP low power modes (including wakeup from DSP OFF mode). To enable the DMA requests mapped via the DMA_CROSSBAR to **DSP_EDMA_CC [19:0]** inputs, corresponding bits in range [19:0] of the register DSP_SYS_DMAWAKEEN0 must be enabled in software.

CAUTION

The DMA request corresponding DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 MUST be enabled, for the DMA requests to be serviced by the DSP regardless of the DSP being in IDLE or active state.



dspss-005

Figure 5-7. DSP DMA Requests

Table 5-6 and Table 5-7list the default DMA sources for the DSP1_EDMA and DSP2_EDMA controllers. In addition, DSP1_EDMA / DSP2_EDMA inputs (DMA_DSP1_DREQ_[19:0] / DMA_DSP2_DREQ_[19:0]) can alternatively be sourced through the associated DMA_CROSSBAR from one of the 256 multiplexed device DMA sources listed in Table 16-6. The CTRL_CORE_DMA_DSP1_DREQ_y_z / CTRL_CORE_DMA_DSP2_DREQ_y_z registers (where *y* and *z* are indexes of DSP1_EDMA / DSP2_EDMA input lines) in the Control Module are used to select between the default DMA sources and the multiplexed DMA sources.

For more details on the device DMA_CROSSBAR multiplexing registers structure, refer to *DMA_CROSSBAR Module Functional Description* of chapter, *Control Module*.

Table 5-6. DSP1_EDMA Default Request Mapping

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP1_DREQ_0	1	CTRL_CORE_DMA_DSP1_DREQ_0_1[7:0]	128	McASP1_DREQ_RX	McASP1 receive event
DMA_DSP1_DREQ_1	2	CTRL_CORE_DMA_DSP1_DREQ_0_1[23:16]	129	McASP1_DREQ_TX	McASP1 transmit event
DMA_DSP1_DREQ_2	3	CTRL_CORE_DMA_DSP1_DREQ_2_3[7:0]	130	McASP2_DREQ_RX	McASP2 receive event
DMA_DSP1_DREQ_3	4	CTRL_CORE_DMA_DSP1_DREQ_2_3[23:16]	131	McASP2_DREQ_TX	McASP2 transmit event
DMA_DSP1_DREQ_4	5	CTRL_CORE_DMA_DSP1_DREQ_4_5[7:0]	132	McASP3_DREQ_RX	McASP3 receive event
DMA_DSP1_DREQ_5	6	CTRL_CORE_DMA_DSP1_DREQ_4_5[23:16]	133	McASP3_DREQ_TX	McASP3 transmit event
DMA_DSP1_DREQ_6	7	CTRL_CORE_DMA_DSP1_DREQ_6_7[7:0]	134	McASP4_DREQ_RX	McASP4 receive event
DMA_DSP1_DREQ_7	8	CTRL_CORE_DMA_DSP1_DREQ_6_7[23:16]	135	McASP4_DREQ_TX	McASP4 transmit event
DMA_DSP1_DREQ_8	9	CTRL_CORE_DMA_DSP1_DREQ_8_9[7:0]	136	McASP5_DREQ_RX	McASP5 receive event
DMA_DSP1_DREQ_9	10	CTRL_CORE_DMA_DSP1_DREQ_8_9[23:16]	137	McASP5_DREQ_TX	McASP5 transmit event
DMA_DSP1_DREQ_10	11	CTRL_CORE_DMA_DSP1_DREQ_10_11[7:0]	138	McASP6_DREQ_RX	McASP6 receive event
DMA_DSP1_DREQ_11	12	CTRL_CORE_DMA_DSP1_DREQ_10_11[23:16]	139	McASP6_DREQ_TX	McASP6 transmit event
DMA_DSP1_DREQ_12	13	CTRL_CORE_DMA_DSP1_DREQ_12_13[7:0]	140	McASP7_DREQ_RX	McASP7 receive event
DMA_DSP1_DREQ_13	14	CTRL_CORE_DMA_DSP1_DREQ_12_13[23:16]	141	McASP7_DREQ_TX	McASP7 transmit event
DMA_DSP1_DREQ_14	15	CTRL_CORE_DMA_DSP1_DREQ_14_15[7:0]	142	McASP8_DREQ_RX	McASP8 receive event
DMA_DSP1_DREQ_15	16	CTRL_CORE_DMA_DSP1_DREQ_14_15[23:16]	143	McASP8_DREQ_TX	McASP8 transmit event
DMA_DSP1_DREQ_16	17	CTRL_CORE_DMA_DSP1_DREQ_16_17[7:0]	154	VCP1_DREQ_RX	VCP1 RX event(1)
DMA_DSP1_DREQ_17	18	CTRL_CORE_DMA_DSP1_DREQ_16_17[23:16]	155	VCP1_DREQ_TX	VCP1 TX event (1)
DMA_DSP1_DREQ_18	19	CTRL_CORE_DMA_DSP1_DREQ_18_19[7:0]	156	VCP2_DREQ_RX	VCP2 RX event(1)
DMA_DSP1_DREQ_19	20	CTRL_CORE_DMA_DSP1_DREQ_18_19[23:16]	157	VCP2_DREQ_TX	VCP2 TX event (1)
DMA_DSP1_DREQ_20 - DMA_DSP1_DREQ_63	N/A	N/A	N/A	Reserved	Reserved

(1) VCP does not support Const/FIFO mode DMA transfers. The DSP1_EDMA should be configured for AB-Synchronized transfer with ACNT = 8, BCNT = number of elements.

Table 5-7. DSP2_EDMA Default Request Mapping

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP2_DREQ_0	1	CTRL_CORE_DMA_DSP2_DREQ_0_1[7:0]	128	McASP1_DREQ_RX	McASP1 receive event
DMA_DSP2_DREQ_1	2	CTRL_CORE_DMA_DSP2_DREQ_0_1[23:16]	129	McASP1_DREQ_TX	McASP1 transmit event
DMA_DSP2_DREQ_2	3	CTRL_CORE_DMA_DSP2_DREQ_2_3[7:0]	130	McASP2_DREQ_RX	McASP2 receive event
DMA_DSP2_DREQ_3	4	CTRL_CORE_DMA_DSP2_DREQ_2_3[23:16]	131	McASP2_DREQ_TX	McASP2 transmit event
DMA_DSP2_DREQ_4	5	CTRL_CORE_DMA_DSP2_DREQ_4_5[7:0]	132	McASP3_DREQ_RX	McASP3 receive event
DMA_DSP2_DREQ_5	6	CTRL_CORE_DMA_DSP2_DREQ_4_5[23:16]	133	McASP3_DREQ_TX	McASP3 transmit event
DMA_DSP2_DREQ_6	7	CTRL_CORE_DMA_DSP2_DREQ_6_7[7:0]	134	McASP4_DREQ_RX	McASP4 receive event
DMA_DSP2_DREQ_7	8	CTRL_CORE_DMA_DSP2_DREQ_6_7[23:16]	135	McASP4_DREQ_TX	McASP4 transmit event
DMA_DSP2_DREQ_8	9	CTRL_CORE_DMA_DSP2_DREQ_8_9[7:0]	136	McASP5_DREQ_RX	McASP5 receive event
DMA_DSP2_DREQ_9	10	CTRL_CORE_DMA_DSP2_DREQ_8_9[23:16]	137	McASP5_DREQ_TX	McASP5 transmit event
DMA_DSP2_DREQ_10	11	CTRL_CORE_DMA_DSP2_DREQ_10_11[7:0]	138	McASP6_DREQ_RX	McASP6 receive event
DMA_DSP2_DREQ_11	12	CTRL_CORE_DMA_DSP2_DREQ_10_11[23:16]	139	McASP6_DREQ_TX	McASP6 transmit event
DMA_DSP2_DREQ_12	13	CTRL_CORE_DMA_DSP2_DREQ_12_13[7:0]	140	McASP7_DREQ_RX	McASP7 receive event
DMA_DSP2_DREQ_13	14	CTRL_CORE_DMA_DSP2_DREQ_12_13[23:16]	141	McASP7_DREQ_TX	McASP7 transmit event
DMA_DSP2_DREQ_14	15	CTRL_CORE_DMA_DSP2_DREQ_14_15[7:0]	142	McASP8_DREQ_RX	McASP8 receive event
DMA_DSP2_DREQ_15	16	CTRL_CORE_DMA_DSP2_DREQ_14_15[23:16]	143	McASP8_DREQ_TX	McASP8 transmit event
DMA_DSP2_DREQ_16	17	CTRL_CORE_DMA_DSP2_DREQ_16_17[7:0]	154	VCP1_DREQ_RX	VCP1 RX event [(1)]
DMA_DSP2_DREQ_17	18	CTRL_CORE_DMA_DSP2_DREQ_16_17[23:16]	155	VCP1_DREQ_TX	VCP1 TX event [(1)]
DMA_DSP2_DREQ_18	19	CTRL_CORE_DMA_DSP2_DREQ_18_19[7:0]	156	VCP2_DREQ_RX	VCP2 RX event [(1)]
DMA_DSP2_DREQ_19	20	CTRL_CORE_DMA_DSP2_DREQ_18_19[23:16]	157	VCP2_DREQ_TX	VCP2 TX event [(1)]
DMA_DSP2_DREQ_20 - DMA_DSP2_DREQ_63	N/A	N/A	N/A	Reserved	Reserved

(1) VCP does not support Const/FIFO mode DMA transfers. The DSP2_EDMA should be configured for AB-Synchronized transfer with ACNT = 8, BCNT = number of elements.

5.3.5.1 DSP EDMA Wakeup Interrupt

This section provides description of the registers used for the **EDMA wakeup interrupt** functionality, including the EDMA_WAKE_INT IRQ status and enable fields. The EDMA Wakeup Interrupt allows incoming EDMA events to be latched and an interrupt sent to the DSP (if enabled). This interrupt is generated in the DSP_SYSTEM as an single "OR-ed" output of all external DMA requests latched in DSP subsystem. This output is further synchronized to DSP_FCLK and mapped as the EDMA_WAKE_INT event to the DSP_IRQ_31 input of the C66x DSP CorePac DSP_INTC. The C66x CPU is expected to service the interrupt by triggering the corresponding EDMA channel manually, or by servicing the request via normal reads and writes (instead of using the EDMA). This functionality is required since the EDMA is not capable of following the smart wakeup protocol.

Note

The DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 registers are used for enabling the assertion of the 'Mwakeup' asynchronous wakeup request to the device PRCM upon DMA requests reception. The interrupt functionality of the registers: DSP_SYS_EDMAWAKE0_x covered in this subsection is specifically for generating **an wake interrupt** to the DSP. In most cases, the enable mask for the two sets of registers should be set to the same value.

The EDMAWAKE0 registers corresponding to the EDMA Events 19 thru 0 (msbit to lsbit) are as follows:

- DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW[19:0]
- DSP_SYS_EDMAWAKE0_IRQSTATUS[19:0]
- DSP_SYS_EDMAWAKE0_IRQENABLE_SET[19:0]
- DSP_SYS_EDMAWAKE0_IRQENABLE_CLR[19:0]

Following functional descriptions are valid for the above registers :

- **IRQ status raw register -** This register provides a per-event raw interrupt status vector. The Raw status is set even if the corresponding event is not enabled. Software can write '1' to set the (raw) status for debug purposes.
- **IRQ status register -** This register provides a per-event enabled interrupt status vector. The Enabled status is set if the corresponding event is enabled and the raw status is set. Software can write 1 to clear the (raw) status after the interrupt has been serviced. The clear takes effect even if the interrupt is not enabled.
- **IRQ enable register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to set (i.e., enable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Clear" - register)
- **IRQ clear register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to clear (i.e., disable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Set" register)

Note

A DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW bit is set even if the corresponding event is NOT enabled in the DSP_SYS_EDMAWAKE0_IRQENABLE_SET[19:0].

5.3.6 DSP Intergated Memory Management Units

5.3.6.1 DSP MMUs Overview

A standalone memory management unit (DSP_MMU0) is included within the DSP1 (DSP1_MMU0) and DSP2 (DSP2_MMU0) subsystems boundaries. The DSP_MMU0 is integrated on the C66x CPU MDMA path to the device L3_MAIN interconnect. This provides several benefits including protection of the system memories from corruption by DSP1 and DSP2 accidental accesses.

A standalone memory management unit (DSP_MMU1) is included within the DSP1 (DSP1_MMU1) and DSP2 (DSP2_MMU1) subsystems boundaries. The DSP_MMU1 is integrated on the EDMA data path which starts from the L2 DSP_NoC interconnect and leaves the DSP subsystem on the DSP EDMA master port. This provides several benefits including protection of the device L3_MAIN memory space from corruption by DSP1 and DSP2 DMA (DSP1_EDMA and DSP2_EDMA, respectively) accidental accesses.

Both DSP MMUs generate interrupts which are internally mapped to the DSP C66x CorePac DSP_INTC and output to the device IRQ_CROSSBAR. See also the Section 5.2 and Section 5.3.4.

CAUTION

In the case of a page fault, a DSP C66x CorePac CPU is unable to service it's own DSP_MMU0 and DSP_MMU1 interrupts . The device MPU (Cortex-A15) is expected to manage any TLB patches as necessary.

Both DSP MMUs (on MDMA and EDMA paths respectively) have identical functionalities.

- 32-bit input and ouput address width (to match L3_MAIN address width)
- 32 TLB cache entries
- 32 + 1 tags
- 128-bit data bus for MDMA and EDMA

5.3.6.2 Routing MDMA Traffic through DSP MMU0

DSP C66x CPU traffic initiated on the DSP MDMA port can be optionally routed through the DSP_MMU0 on the 32-bit MDMA address path. This is controlled in two levels :

- **global** by the DSP_SYSTEM register DSP_SYS_MMU_CONFIG [0] MMU0_EN bit. This bit acts as a mux-select : setting it to 0b1 enables requests to use the DSP_MMU0; clearing this bit to 0b0 disables MMU table lookup and causes accesses to use the non-translated address (MMU bypass). By default the DSP_MMU0 is disabled in DSP_SYSTEM and MDMA port traffic bypasses the DSP_MMU0.
- **local** by MMU enable control in a dedicated DSP_MMU0 memory mapped register. It is used to enable the MMU functionality after the page tables are programmed for MMU operation. For details, refer to the Chapter 20, *Memory Management Units*.

Note

For the DSP_MMU0 to operate, SW should enable it both at the DSP_SYSTEM global level and DSP_MMU0 local register level.

When enabling the DSP_MMU0, software must take care that no transactions are in flight through that MMU. This is typically handled by issueing a DSP "MFENCE" instruction operation. Note that the local enable bit inside the DSP_MMU0 must be configured as normal (refer to the Chapter 20, *Memory Management Units*.)

For more information on the MFENCE operation, refer to the section, *C66x CPU Instruction Set* of the *TMS320C66x DSP CPU and Instruction Set*).

In addition, the DSP_MMU0 traffic can be aborted in case of a lockup via the DSP_SYS_MMU_CONFIG [8] MMU0_ABORT bit. In other words, this bit can be used to clear a hang condition that may occur if the DSP_MMU0 encounters a page fault that cannot be serviced.

For more information on device DSP_MMU0 functionality and register settings, refer to the Section 20.3, *MMU Functional Description* and Section 20.5, *MMU Register Manual*, in the chapter, *Memory Management Units*, respectively.

5.3.6.3 Routing EDMA Traffic thorugh DSP MMU1

The DSP_EDMA traffics initiated on the DSP EDMA master port can be optionally routed through the DSP_MMU1 on the EDMA address path. This is controlled in two levels :

- **global** by the DSP_SYSTEM register DSP_SYS_MMU_CONFIG [4] MMU1_EN bit. This bit acts as a mux-select : setting it to 0b1 enables requests to use the DSP_MMU1; clearing this bit to 0b0 disables MMU table lookup and causes accesses to use the non-translated address (MMU bypass). By default the DSP_MMU1 is disabled in DSP_SYSTEM and EDMA traffic bypasses the DSP_MMU1.

- **local** by MMU enable control in a dedicated DSP_MMU1 memory mapped register. It is used to enable the MMU functionality after the page tables are programmed for MMU operation. For details, refer to the Section 20.5, *MMU Register Manual*, in the Chapter 20,*Memory Management Units*.

Note

For the DSP_MMU1 to operate, SW should enable it both at the DSP_SYSTEM global level and DSP_MMU1 local register level.

When enabling the DSP_MMU1, software must take care that no transactions are in flight through that MMU. This is typically handled by disabling any EDMA transactions prior to enabling the MMU. Note that the local enable bit inside the DSP_MMU1 must be configured as normal (refer to the Chapter 20,*Memory Management Units*.)

For more information on the MFENCE operation, refer to the section, *C66x CPU Instruction Set* of the *TMS320C66x DSP CPU and Instruction Set*).

In addition, the DSP_MMU1 traffic can be aborted in case of a lockup via the DSP_SYS_MMU_CONFIG [12] MMU1_ABORT bit. In other words, this bit can be used to clear a hang condition that may occur if the DSP_MMU1 encounters a page fault that cannot be serviced.

For more information on device DSP_MMU1 functionality and settings, refer to the Section 20.3, *MMU Functional Description* and the Section 20.5, *MMU Register Manual*, in the chapter, *Memory Management Units*, respectively.

5.3.7 DSP Integrated EDMA Subsystem

This section represents an overview of the DSP integrated EDMA functionalities, as well as the subsystem level and device related register controls. For more details on the EDMA functionalities and programming registers, refer to the *Enhanced DMA*.

5.3.7.1 DSP EDMA Overview

The enhanced-DMA subsystem which is part of the DSP1 (DSP1_EDMA) and the DSP2 (DSP2_EDMA) subsystems is the primary DMA engine for transfers between system memory (DDR and/or L3_MAIN SRAM) and DSP internal memories (L1s and L2).

The Channel Controller - DSP_EDMA_CC serves as the “user interface” of the DSP_EDMA. The two Transfer Controllers - DSP_EDMA_TC0 and DSP_EDMA_TC1 serve as the data transfer engines of the DSP_EDMA. The C66x CPU tipically programs the Channel Controller, which in turn submits Transfer Requests (TR) to the appropriate Transfer Controller. Interrupts are posted in the DSP_EDMA_CC upon transfer completion (if requested), and signaled to the C66x. The EDMA TC completion interrupt is not supported/connected.

The DSP_EDMA is primarily used to perform block transfers between DSP C66x CorePac memories (mostly L2 memory) and system memory (mostly DDR or L3 SRAM).

The DSP_EDMA is configured with 2 Queues (in the CC). Two DSP_EDMA traffic controllers (TC) offer high performance and preemptability of transfers. For typical use cases, it is expected that low latency/small payload transfers) use Queue0/TC0 and high bandwidth/large payload transfers (e.g., DDR on L3_MAIN or DSP local L2 SRAM) will use Queue1/TC1.

DSP_EDMA_CC configuration in the device features :

- 64x EDMA channels
- 8x QDMA channels
- 64x interrupt channels
- 128x PaRAM entries
- 2x Event Queues
- 2x Traffic controllers

- memory protection support
- channel mapping capability
- 8x memory protected and Shadow Regions

DSP_EDMA_TC0/TC1 configuration in the device features :

- 2048 Byte FIFO support
- multitag support
- 16-bit data bus
- 4-level destination register depth
- 7-bit address for internal FIFOs
- 16 IDs for Read commands
- 16 IDs for Write commands

Note

The device DSP integrated EDMA controller instances (DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1) are functionally identical with the device EDMA controller instances (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1). The only difference is that the DSP_EDMA instances are located at different physical addresses.

For more details on the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 controllers functionalities, refer to *EDMA Controller Functional Description*, in *Enhanced DMA*.

The DSP_EDMA instances, their corresponding registers summary and descriptions are covered in the *EDMA Register Manual* of the *Enhanced DMA*.

5.3.7.2 DSP System and Device Level Settings of DSP EDMA

DSP_EDMA traffic TC0 and TC1 controllers "Active" or "Idle" status: can be monitored in DSP_SYSTEM located :

- DSP_SYS_STAT[1] TC0_STAT bit
- DSP_SYS_STAT[2] TC1_STAT bit

The **default** burst size for both the DSP_EDMA_TC0 and DSP_EDMA_TC1 can be defined in DSP_SYSTEM register. This is achieved via programming :

- DSP_SYS_BUS_CONFIG [1:0] TC0_DBS
- DSP_SYS_BUS_CONFIG [5:4] TC1_DBS

There are also other DSP_EDMA controls associated with DSP_NoC interconnect pressure settings. For more details, refer to the Section 5.3.8.

The 3 error events associated with the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 are exported outside the DSP C66x CorePac in the subsystem, and are able to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"tpcc_errint_level", "tptc_errint0_level" and "tptc_errint1_level"** events, respectively in the Table 5-5.

Note

The DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 events are NOT exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

The DSP_SYSTEM logic is assigned to route the external DMA requests to the EDMA hardware request inputs. Additionally, EDMA events can conditionally wake-up the DSP system from a low power mode, via software enabling DSP_SYSTEM MWakeup handshake with the device PRCM. This mechanism is described in the Section 5.3.5.

The programmable muxing of various external DMA request sources to the DSP EDMA. DMA_DSP1_DREQ_x and DMA_DSP2_DREQ_x input lines (where x=0 to 19) is covered in the *DMA_CROSSBAR Module Functional Description*, of the *Control Module*.

DSP1/DSP2 subsystem external DMA request sources : For the default DSP1 / DSP2 external DMA request sources, routed via the device DMA_CROSSBAR to the DSP1_EDMA / DSP2_EDMA channel controller inputs (DMA_DSP1_DREQ_i / DMA_DSP2_DREQ_i), respectively, refer to the Section 5.3.5.

5.3.8 DSP L2 interconnect Network

A 128-bit level 2 (L2) Interconnect from Arteris - FlexNoC ® is instantiated in the DSP subsystem, outside the DSP C66x CorePac. It is signified as "DSP_NoC" throughout this chapter.

Note

The C66x master MDMA data does NOT flow through the DSP_NoC.

The system and local initiators on DSP_NoC are as follows :

- local C66x CPU 32-bit CFG master port which traffic is split via DSP_NoC fabric into several configuration target traffics inside and outside the DSP subsystem.
- SDMA initiator port on DSP_NoC which conveys accesses towards DSP Memories and memory-mapped registers initiated outside the DSP subsystem via the L3_MAIN interconnect.
- EDMA traffic controllers - TC0 read / write initiator ports
- EDMA traffic controllers - TC1 read / write initiator ports

Note

The DSP_ICFG space is not visible to SDMA initiators (DSP_EDMA or DSP hosts on L3_MAIN) with CFG traffic.

The targets on the DSP_NoC are as follows :

- DSP C66x CorePac SDMA port
- Internal CFG targets on the DSP_NoC :
  - DSP_MMU0 Cfg
  - DSP_MMU1 Cfg
  - DSP_SYSTEM Cfg
  - DSP_EDMA_CC Cfg
  - DSP_EDMA_TC0 Cfg
  - DSP_EDMA_TC1 Cfg
- 32-bit CFG port on L3_MAIN (it acts as master on the L3_MAIN)
- EDMA Target port which conveys EDMA bidi transfers outside the DSP (through or bypassing DSP_MMU1).

The Table 5-8 summarizes the interconnections which can be established between DSP initiators and targets over the L2 DSP_NoC in the device. In this table HW implemented interconnections are marked with an asterics.

Table 5-8. DSP_NoC Defined Connectivities

		DSP_NoC Initiators			
		DSP C66x CorePac CFG init	EDMA_TC0 init	EDMA_TC1 init	SDMA init (mapped to SDMA port on L3_MAIN)
DSP_NoC Targets	DSP C66x CorePac SDMA (slave) port	n.a.	*	*	*
	DSP_MMU0 Cfg	*	n.a.	n.a.	*
	DSP_MMU1 Cfg	*	n.a.	n.a.	*
	DSP_SYSTEM Cfg	*	n.a.	n.a.	*
	DSP_EDMA_CC Cfg	*	n.a.	n.a.	*
	DSP_EDMA_TC0 Cfg	*	n.a.	n.a.	*
	DSP_EDMA_TC1 Cfg	*	n.a.	n.a.	*
	DSP_NoC Cfg	*	n.a.	n.a.	*
	Cfg port (Cfg Init on L3_MAIN)	*	n.a.	n.a.	n.a.

Table 5-8. DSP_NoC Defined Connectivities (continued)

		DSP_NoC Initiators			
	Master DMA port (DSP DMA init on L3_MAIN)	n.a.	*	*	n.a.

A DSP_NoC error event (combination of several local to the interconnect events) is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"noc_errint_level"** event in the Table 5-5.

Note

The DSP_NoC event is NOT exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 5.3.4.2.2.

5.3.8.1 DSP Public Firewall Settings

The DSP1 and DSP2 L2 Interconnect (DSP1_NoC and DSP2_NoC, respectively) implements two firewalls – dsp firewall0 (DSP_FW0) is used to protect DSP_MMU0's configuration space (which includes the TLB) and dsp firewall1 is used to protect DSP_MMU1's configuration space (which includes the TLB). Access permission is based on the privilege level, domain, ConnID, and access types of a request.

The default value of 0xFFFF_FFFF in the MRM region 0 permission registers:

- L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0
- L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0
- L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0
- L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

permits any requestor to access the DSP_MMU0 and DSP_MMU1 configuration space.

For more information on the access region definitions, public privilege access, public user access and initiator permission settings, which are identical between DSP_NoC firewalls and L3_MAIN interconnect firewalls, refer to the *L3_MAIN Firewall Functionality*, in the, *L3_MAIN Interconnect*.

There are also several other DSP_NoC registers
- L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0, L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL used for error handling, firewall reset and other purposes. These are functionally identical with the corresponding L3_MAIN interconnect registers, described in the *L3_MAIN Interconnect Error Handling*, in the *L3_MAIN Interconnect*.

The various firewall access control registers are part of the C66x CPU local accessible - DSP_FW_L2_NOC_CFG address space, and L3_MAIN initiators accessible DSP1/2_FW_L2_NOC_CFG configuration space. The corresponding MMU0 and MMU1 configuration space firewall registers (DSP_FW0 starting at offset **0x0000_0000** , and DSP_FW1 starting at offset **0x0000_1000**) are summarized and described in the Section 5.4.4.

5.3.8.2 DSP NoC Flag Mux and Error Log Registers

The DSP_NoC registers (starting at offset >= 0x0000_4000) are used for error logging and flag muxing purposes. The status information stored in there can be used for example to resolve issues related to DSP_NoC access conflicts, for debug purposes, etc.

For more information, refer to the *Flag Muxing*, in the *L3_MAIN Interconnect*.

5.3.8.3 DSP NoC Arbitration

A pressure based arbitration is implemented for the DSP_NoC interconnect.

A DSP_NoC local MFlag mechanism is used but it is SW controlled in the DSP_SYSTEM configuration space.

This is done via the register DSP_SYS_BUS_CONFIG bitfields :

- TC0_L2PRES - for DSP_EDMA_TC0 pressure control
- TC1_L2PRES - for DSP_EDMA_TC1 pressure control
- CFG_L2PRES - for the DSP C66x CorePac 32-bit CFG pressure control
- SDMA_L2PRES - for pressure control of the DSP system and L3_MAIN accesses targetting the DSP C66x CorePac SDMA slave port

The pressure for each port is signaled on a MFlag[1:0] bus and conveys a value of 0 (lowest), 1 (medium), or 3 (highest). A value of 0x2 is reserved/undefined.

Note

The default pressure level for all ports is 0x0 and is recommended for most systems. This results in round-robin arbitration across active requests.

5.3.9 DSP Boot Configuration

DSP1 subsystem boot vector input which defines the 22-bit DSP1 Boot Address is mapped to the device core control module register CTRL_CORE_CONTROL_DSP1_RST_VECT[21:0] DSP1_RST_VECT bitfield. DSP2 subsystem boot vector input which defines the 22-bit DSP2 Boot Address is mapped to the device core control module register CTRL_CORE_CONTROL_DSP2_RST_VECT[21:0] DSP2_RST_VECT bitfield.In general, the device MPU (Cortex-A15) host loads code to a given address location in the device system memory, sets the DSP1_RST_VECT / DSP2_RST_VECT bitfield to the address value, and then release the DSP1 / DSP2 from reset. At that point, the DSP1 / DSP2 will begin fetching code from that location.

Note

If the values of the control core module CTRL_CORE_CONTROL_DSP1_RST_VECT[21:0] / CTRL_CORE_CONTROL_DSP2_RST_VECT [21:0] register change, the values will be taken into account by DSP upon the next reset.

Note

Upon device boot time (a power-on reset applied), the device "sysboot15" input latched in the Control Module bootstrap register defines the value of DSP functional clock divider (2 or 3). For more details, refer to the Section 5.3.3.1

5.3.10 DSP Internal and External Memory Views

5.3.10.1 C66x CPU View of the Address Space

The **C66x CPU View** represents the view from the DSP, which result from program fetches, or load / store instructions. Accesses to DSP memories (L1P, L1D, L2) and to DSP Internal configuration space (DSP_ICFG) are intercepted within the DSP C66x CorePac (whether using local or global addresses).

The DSP C66x CorePac CFG (C66x CPU 32-bit master port) interface is strictly for non-cacheable loads and stores, and is intended to be used for I/O space or memory mapped registers (MMR) space. DSP C66x CorePac CFG accesses are routed / arbitrated by the DSP_NoC L2 inteconnect. DSP C66x CorePac CFG accesses are mapped between 0x01C0_0000 and 0x0FFF_FFFF.

Note

The DSP C66x CorePac CFG initiator interface is strictly for non-cacheable loads and stores within MMR and I/O spaces.

DSP configuration accesses to **external to DSP subsystem** peripherals can be issued on either the DSP subsystem CFG Master port which is mapped to the device L3_MAIN or DSP 32-bit CFG "system" interface is connected to the chip level L3_MAIN interconnect and used to access L3_MAIN addressses that are mapped between 0x01C0_0000 and 0x0FFF_FFFF.

DSP **accesses (to non-DSP memories like SDRAM on L3_MAIN) for addresses above 0x1000_0000 are handled via the DSP (XMC) MDMA 128-bit master interface and are routed to the DSP subsystem MDMA Initiator port either through DSP_ MMU0 or bypassing MMU.**

Note

In some cases, L3_MAIN peripherals may be mapped to both the MDMA bus and the CFG bus. In that case, there may be a latency advantage of using the CFG address for those peripherals

---

Table 5-9 shows the DSP C66x CPU memory view of the various DSP C66x CorePac internal and external resources.

Table 5-9. C66x CPU View Map

C66x CPU View (DSP C66x CorePac Internal only, MDMA or CFG init ports)[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x0080_0000	0x0084_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x00E0_0000	0x00E0_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D	DSP L1D SRAM (local)
0x0180_0000	0x01BF_FFFF	4096 KiB	DSP_ICFG	DSP Internal CFG [2]
0x01D0_0000	0x01D0_0FFF	4 KiB	DSP_SYSTEM	DSP_SYSTEM Memory Mapped Registers block
0x01D0_1000	0x01D0_1FFF	4 KiB	DSP_MMU0CFG	DSP MMU0 configuration / registers
0x01D0_2000	0x01D0_2FFF	4 KiB	DSP_MMU1CFG	DSP MMU1 configuration / registers
0x01D0_5000	0x01D0_5FFF	4 KiB	DSP_EDMA_TC0	DSP_EDMA Transfer Controller 0
0x01D0_6000	0x01D0_6FFF	4 KiB	DSP_EDMA_TC1	DSP_EDMA Transfer Controller 1
0x01D0_7000	0x01D0_7FFF	4 KiB	DSP_NoC	DSP L2 interconnect registers
0x01D1_0000	0x01D1_7FFF	32 KiB	DSP_EDMA_CC	DSP_EDMA Channel Controller
0x0200_0000	0x020F_FFFF	1 MiB	EVE1	DSP configuration traffic to the EVE1 (mapped on the DSP CFG interface)
0x0210_0000	0x021F_FFFF	1 MiB	EVE2	DSP configuration traffic to the EVE2 (mapped on the DSP CFG interface)
0x0220_0000	0x023F_FFFF	2 MiB	Reserved	Reserved
0x0330_0000	0x033F_FFFF	1 MiB	EDMA_TPCC	DSP configuration traffic to the EDMA_TPCC (mapped on the DSP CFG interface)
0x0340_0000	0x034F_FFFF	1 MiB	EDMA_TC0	DSP configuration traffic to the EDMA_TC0 (mapped on the DSP CFG interface)
0x0350_0000	0x035F_FFFF	1 MiB	EDMA_TC1	DSP configuration traffic to the EDMA_TC1 (mapped on the DSP CFG interface)
0x0800_0000	0x0800_FFFF	64 KiB	DSP_XMC_CTRL MMRs	DSP internal MMRs for XMC controller (non-cache)
0x0802_0000	0x080F_FFFF	896 KiB	MDMA non-cache	MDMA initiator (non-cache) to L3_MAIN (DSP_MMU0)
0x0810_0000	0x0BBF_FFFF	59 MiB		

Table 5-9. C66x CPU View Map (continued)

C66x CPU View (DSP C66x CorePac Internal only, MDMA or CFG init ports)[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x1000_0000	0x10FF_FFFF	16 MiB	DSP1 L1P, L1D and L2 memories	An image of DSP1 C66x CorePac internal space - only L1P, L1D and L2 memories (global) [3]
0x1000_0000	0x10FF_FFFF	16 MiB	DSP2 MDMA (cached)	DSP2 MDMA initiator (cached) to L3_MAIN (through DSP2_MMU0)
0x1100_0000	0x11FF_FFFF	16 MiB	DSP2 L1P, L1D and L2 memories	An image of DSP2 C66x CorePac internal space - only L1P, L1D and L2 memories (global) [4]
			DSP1 MDMA (cached)	MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0)
0x1200_0000	0x1FFF_FFFF	224 MiB	MDMA (cached)	DSP1 and DSP2 MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0 / DSP2_MMU0, respectively)
0x2000_0000	0xFFFF_FFFF	3584 MiB	MDMA (cached)	DSP1 and DSP2 MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0 / DSP2_MMU0, respectively)

(1) Only the C66x CPU view of device implemented functional memory regions are shown. The remaining regions are reserved.
(2) The internal configuration space registers are visible ONLY to the C66x CPU (DSP_C0) within the DSP C66x CorePac, i.e. they are NOT visible to initiators outside the DSP C66x CorePac.
(3) The DSP1 CPU sees an image of its own memories in the 0x1000_0000 - 0x10FF_FFFF address range (the same mapped also at lower addresses 0x0080_0000 - 0x00F0_7FFF). On the other side, DSP2 CPU addresses, generated in the same range (with DSP2_MMU0 involved), are mapped to the DSP2 MDMA port (cached transfers to/from L3_MAIN connected system memories).
(4) The DSP2 CPU sees an image of its own memories in the 0x1100_0000 - 0x11FF_FFFF address range (the same mapped also at lower addresses 0x0080_0000 - 0x00F0_7FFF). On the other side, DSP1 CPU addresses, generated in the same range (with DSP1_MMU0 involved), are mapped to the DSP1 MDMA port (cached transfers to / from L3_MAIN connected system memories).

Refer to the *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

5.3.10.2 DSP_EDMA View of the Address Space

EDMA is able to initiate internal accesses directly to the DSP memories via the DSP C66x CorePac SDMA bus. The access is conducted to the DSP C66x CorePac internal memories over the L2 DSP_NoC interconnect.

Table 5-10 shows the DSP integrated EDMA controller memory view of the various DSP C66x CorePac internal and external resources.

Table 5-10. DSP EDMA Controller View Map

DSP_EDMA Controller View (EDMA master internal / external port)		Size	DSP Memory Region	Function
Start Address	End Address			
0x0080_0000	0x0084_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x00E0_0000	0x00E0_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D	DSP L1D SRAM (local)
0x0802_0000	0x0BBF_FFFF	59 MiB	EDMA to L3_MAIN	EDMA initiator (DSP_MMU1)
0x1000_0000	0x10FF_FFFF	16 MiB	DSP L1/L2	An image of DSP C66x CorePac internal space - only L1P, L1D and L2 memories (global) [1]

Table 5-10. DSP EDMA Controller View Map (continued)

DSP_EDMA Controller View (EDMA master internal / external port)		Size	DSP Memory Region	Function
Start Address	End Address			
0x2000_0000	0xFFFF_FFFF	3584 MiB	DMA OCP	L3_MAIN interconnect memory via MMU1 / DMA OCP Initiator

(1) The internal configuration space registers DSP_ICFG are visible ONLY to the C66x CPU (DSP_C0) within the DSP C66x CorePac , i.e. they are NOT visible to the DSP_EDMA.

Access from EDMA to external resources on L3_MAIN are routed via DSP subsystem **EDMA initiator port**. **Note that these accesses are transferred through the DSP_MMU1 memory management unit.**

Note

The DSP_EDMA can NOT access the DSP_ICFG (DSP C66x CorePac internal) addresses.

With the DSP_MMU1 disabled, the subset of the memory map used for DSP_EDMA internal accesses will NOT be visible. Thus only addresse which equal 0x2000_0000 and above will be considered as valid 32-bit addresses (i.e. L3_MAIN space accesses only).

Refer to the *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

5.3.10.3 L3_MAIN View of the DSP Address Space

System initiated accesses (i.e. external-to-DSP accesses over device L3_MAIN) to DSP are issued over the DSP SDMA target port.

Note

The MSB-bits of the address are truncated to only provide an 8 MiB view of the memory map within the DSP subsystem. Notice that the relative offsets of the DSP CFG space is different for the OCP SDMA target port relative to the DSP internal initiators.

The SDMA target bus is able to access internal subsystem address space (such as DSP_EDMA, DSP_MMU0, DPS_MMU1, etc.), or the DSP local memory address space. The SDMA target bus **is NOT able to access the DSP ICFG space (such as DSP_INTC, DSP_BWM, etc)** or the other initiator ports on the DSP subsystem boundary (i.e., accesses cannot go through DSPSS to get to the DSP_EDMA initiator port, L3_MAIN CFG initiator port).

The DSP slave DMA port memory map - Table 5-11 shows an 8 MiB window (23-bit address) both from the SDMA Target bus (0x0000_0000 through 0x007F_FFFF), as well as the EDMA (0x0080_0000 through 0x00FF_FFFF). The DSP C66x CorePac internally views itself as a 16 MiB window where 0x0000_0000 through 0x007F_FFFF is reserved, L2 SRAM starts at 0x0080_0000, L1P SRAM starts at 0x00E0_0000, and L1D SRAM starts at 0x00F0_0000).

Table 5-11. SDMA Target Port Memory Map

System L3_MAIN View(1)		Size	DSP Memory Region	Function
Start Address	End Address			
0x0000_0000	0x0004_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x0050_0000	0x0050_0FFF	4 KiB	DSP_SYSTEM	DSP SYSTEM MMR Block
0x0050_1000	0x0050_1FFF	4 KiB	DSP_MMU0CFG	DSP MMU0 configuration / regs
0x0050_2000	0x0050_2FFF	4 KiB	DSP_MMU1CFG	DSP MMU1 configuration / regs
0x0050_5000	0x0050_5FFF	4 KiB	DSP_EDMA_TC0	DSP EDMA Transfer Controller 0
0x0050_6000	0x0050_6FFF	4 KiB	DSP_EDMA_TC1	DSP EDMA Transfer Controller 1

Table 5-11. SDMA Target Port Memory Map (continued)

System L3_MAIN View[(1)]		Size	DSP Memory Region	Function
Start Address	End Address			
0x0050_7000	0x0050_7FFF	4 KiB	DSP_NoC	DSP L2 Interconnect registers
0x0051_0000	0x0051_7FFF	32 KiB	DSP_EDMA_CC	DSP EDMA Channel Controller
0x0060_0000	0x0060_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x0070_0000	0x0070_7FFF	32 KiB	DSP_L1D	DSP L1D SRAM (local)

(1) Only system (L3_MAIN) view over functionally used regions are shown. The remaining regions are reserved.

Refer to the *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

5.4 DSP Subsystem Register Manual

This section describes the DSP Subsystem instances registers.

5.4.1 DSP Subsystem Instance Summary

Table 5-12. DSP Subsystem Instance Summary

Module Name	Module Base Address	Size
DSP_ICFG	0x0180 0000[(1)]	4 KiB
DSP_SYSTEM	0x01D0 0000[(1)]	256 Bytes
DSP_FW_L2_NOC_CFG	0x01D0 3000[(1)]	8576 Bytes
DSP1_SYSTEM	0x40D0 0000	256 Bytes
DSP1_FW_L2_NOC_CFG	0x40D0 3000	8576 Bytes
DSP2_SYSTEM	0x4150 0000	256 Bytes
DSP2_FW_L2_NOC_CFG	0x4150 3000	8576 Bytes

(1) The registers of DSP subsystem instances prefixed only with DSP in the name, and NOT DSP1 or DSP2, are NOT visible on the device L3_MAIN. They are visible only within the DSP_ICFG internal configuration space hence accessible only by the DSP C66x CPU.

Note

For more details on the DSP_MMU0 and DSP_MMU1 registers, as well as their :

- DSP_MMU0CFG and DSP_MMU1CFG physical addresses accesible only by DSP_C0 CPU core in the DSP subsystem
- DSP1_MMU0CFG and DSP1_MMU1CFG physical addresses visible on L3_MAIN
- DSP2_MMU0CFG and DSP2_MMU1CFG physical addresses visible on L3_MAIN

refer to Section 20.5, *MMU Register Manual*, in Chapter 20, *Memory Management Units*.

Note

For more details on the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 registers, as well as their:

- DSP1_EDMA_CC, DSP1_EDMA_TC0 and DSP1_EDMA_TC1 physical addresses visible on L3_MAIN
- DSP2_EDMA_CC, DSP2_EDMA_TC0 and DSP2_EDMA_TC1 physical addresses visible on L3_MAIN

refer to *EDMA Register Manual*, in *Enhanced DMA*.

CAUTION

The L1P, L1D and L2 memory controller registers mapped in the L3_MAIN are limited to 32-bit data access; 16- and 8-bit access are not allowed and can corrupt register content.

5.4.2 DSP_ICFG Registers

5.4.2.1 DSP_ICFG Register Summary

Note

The DSP_ICFG addresses are visible only within the DSP core internal configuration space.

Table 5-13. DSP_ICFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[(4)]
EVTFLAG0	R	32	0x0000 0000	0x0180 0000	0x0000 0000
EVTFLAG1	R	32	0x0000 0004	0x0180 0004	0x0000 0000
EVTFLAG2	R	32	0x0000 0008	0x0180 0008	0x0000 0000
EVTFLAG3	R	32	0x0000 000C	0x0180 000C	0x0000 0000
EVTSET0	W	32	0x0000 0020	0x0180 0020	0x0000 0000
EVTSET1	W	32	0x0000 0024	0x0180 0024	0x0000 0000
EVTSET2	W	32	0x0000 0028	0x0180 0028	0x0000 0000
EVTSET3	W	32	0x0000 002C	0x0180 002C	0x0000 0000
EVTCLR0	W	32	0x0000 0040	0x0180 0040	0x0000 0000
EVTCLR1	W	32	0x0000 0044	0x0180 0044	0x0000 0000
EVTCLR2	W	32	0x0000 0048	0x0180 0048	0x0000 0000
EVTCLR3	W	32	0x0000 004C	0x0180 004C	0x0000 0000
EVTMASK0	RW	32	0x0000 0080	0x0180 0080	0x0000 0000
EVTMASK1	RW	32	0x0000 0084	0x0180 0084	0x0000 0000
EVTMASK2	RW	32	0x0000 0088	0x0180 0088	0x0000 0000
EVTMASK3	RW	32	0x0000 008C	0x0180 008C	0x0000 0000
MEVTFLAG0	R	32	0x0000 00A0	0x0180 00A0	0x0000 0000
MEVTFLAG1	R	32	0x0000 00A4	0x0180 00A4	0x0000 0000
MEVTFLAG2	R	32	0x0000 00A8	0x0180 00A8	0x0000 0000
MEVTFLAG3	R	32	0x0000 00AC	0x0180 00AC	0x0000 0000
EXPMASK0	RW	32	0x0000 00C0	0x0180 00C0	0xFFFF FFFF
EXPMASK1	RW	32	0x0000 00C4	0x0180 00C4	0xFFFF FFFF
EXPMASK2	RW	32	0x0000 00C8	0x0180 00C8	0xFFFF FFFF
EXPMASK3	RW	32	0x0000 00CC	0x0180 00CC	0xFFFF FFFF
MEXPFLAG0	R	32	0x0000 00E0	0x0180 00E0	0x0000 0000
MEXPFLAG1	R	32	0x0000 00E4	0x0180 00E4	0x0000 0000
MEXPFLAG2	R	32	0x0000 00E8	0x0180 00E8	0x0000 0000
MEXPFLAG3	R	32	0x0000 00EC	0x0180 00EC	0x0000 0000
INTMUX1	RW	32	0x0000 0104	0x0180 0104	0x0706 0504
INTMUX2	RW	32	0x0000 0108	0x0180 0108	0x0B0A 0908
INTMUX3	RW	32	0x0000 010C	0x0180 010C	0x0F0E 0D0C
AEGMUX0	RW	32	0x0000 0140	0x0180 0140	0x0302 0100
AEGMUX1	RW	32	0x0000 0144	0x0180 0144	0x0706 0504
INTXSTAT	RW	32	0x0000 0180	0x0180 0180	0x0000 0000
INTXCLR	RW	32	0x0000 0184	0x0180 0184	0x0000 0000

Table 5-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[(4)]
INTDMASK	RW	32	0x0000 0188	0x0180 0188	0x0000 0000
EVTASRT	RW	32	0x0000 01C0	0x0180 01C0	0x0302 0100
PDCCMD	RW	32	0x0001 0000	0x0181 0000	0x0000 0000
MM_REVID	RW	32	0x0001 2000	0x0181 2000	0x0000 0000
IDMA0_STAT	RW	32	0x0002 0000	0x0182 0000	0x0000 0000
IDMA0_MASK	RW	32	0x0002 0004	0x0182 0004	0x0000 0000
IDMA0_SOURCE	RW	32	0x0002 0008	0x0182 0008	0x0000 0000
IDMA0_DEST	RW	32	0x0002 000C	0x0182 000C	0x0000 0000
IDMA0_COUNT	RW	32	0x0002 0010	0x0182 0010	0x0000 0000
IDMA1_STAT	RW	32	0x0002 0100	0x0182 0100	0x0000 0000
IDMA1_SOURCE	RW	32	0x0002 0108	0x0182 0108	0x0000 0000
IDMA1_DEST	RW	32	0x0002 010C	0x0182 010C	0x0000 0000
IDMA1_COUNT	RW	32	0x0002 0110	0x0182 0110	0x0000 0000
CPUARBE	RW	32	0x0002 0200	0x0182 0200	0x0001 0010
IDMAARBE	RW	32	0x0002 0204	0x0182 0204	0x0000 0010
SDMAARBE	RW	32	0x0002 0208	0x0182 0208	0x0000 0001
ECFGARBE	RW	32	0x0002 0210	0x0182 0210	0x0007 0000
ICFGMPFAR	R	32	0x0002 0300	0x0182 0300	0x0000 0000
ICFGMPFSR	RW	32	0x0002 0304	0x0182 0304	0x0000 0000
ICFGMPFCR	RW	32	0x0002 0308	0x0182 0308	0x0000 0000
ECFGERR	RW	32	0x0002 0408	0x0182 0408	0x0000 0000
ECFGERRCLR	RW	32	0x0002 040C	0x0182 040C	0x0000 0000
PAMAP0	RW	32	0x0002 0500	0x0182 0500	0x0000 0000
PAMAP1	RW	32	0x0002 0504	0x0182 0504	0x0000 0001
PAMAP2	RW	32	0x0002 0508	0x0182 0508	0x0000 0002
PAMAP3	RW	32	0x0002 050C	0x0182 050C	0x0000 0003
PAMAP4	RW	32	0x0002 0510	0x0182 0510	0x0000 0004
PAMAP5	RW	32	0x0002 0514	0x0182 0514	0x0000 0005
PAMAP6	RW	32	0x0002 0518	0x0182 0518	0x0000 0006
PAMAP7	RW	32	0x0002 051C	0x0182 051C	0x0000 0007
PAMAP8	RW	32	0x0002 0520	0x0182 0520	0x0000 0007
PAMAP9	RW	32	0x0002 0524	0x0182 0524	0x0000 0007
PAMAP10	RW	32	0x0002 0528	0x0182 0528	0x0000 0007
PAMAP11	RW	32	0x0002 052C	0x0182 052C	0x0000 0007
PAMAP12	RW	32	0x0002 0530	0x0182 0530	0x0000 0007
PAMAP13	RW	32	0x0002 0534	0x0182 0534	0x0000 0007
PAMAP14	RW	32	0x0002 0538	0x0182 0538	0x0000 0007
PAMAP15	RW	32	0x0002 053C	0x0182 053C	0x0000 0007
L2CFG	RW	32	0x0004 0000	0x0184 0000	0x0100 0000
L1PCFG	RW	32	0x0004 0020	0x0184 0020	0x0000 0007
L1PCC	RW	32	0x0004 0024	0x0184 0024	0x0000 0000
L1DCFG	RW	32	0x0004 0040	0x0184 0040	0x0000 0007
L1DCC	RW	32	0x0004 0044	0x0184 0044	0x0000 0000
CPUARBU	RW	32	0x0004 1000	0x0184 1000	0x0001 0010
IDMAARBU	RW	32	0x0004 1004	0x0184 1004	0x0000 0010

Table 5-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[(4)]
SDMAARBU	RW	32	0x0004 1008	0x0184 1008	0x0000 0001
UCARBU	RW	32	0x0004 100C	0x0184 100C	0x0000 0020
MDMAARBU	RW	32	0x0004 1010	0x0184 1010	0x0607 0000
CPUARBD	RW	32	0x0004 1040	0x0184 1040	0x0001 0010
IDMAARBD	RW	32	0x0004 1044	0x0184 1044	0x0000 0010
SDMAARBD	RW	32	0x0004 1048	0x0184 1048	0x0000 0001
UCARBD	RW	32	0x0004 104C	0x0184 104C	0x0000 0020
L2WBAR	W	32	0x0004 4000	0x0184 4000	0x0000 0000
L2WWC	RW	32	0x0004 4004	0x0184 4004	0x0000 0000
L2WIBAR	W	32	0x0004 4010	0x0184 4010	0x0000 0000
L2WIWC	RW	32	0x0004 4014	0x0184 4014	0x0000 0000
L2IBAR	W	32	0x0004 4018	0x0184 4018	0x0000 0000
L2IWC	RW	32	0x0004 401C	0x0184 401C	0x0000 0000
L1PIBAR	W	32	0x0004 4020	0x0184 4020	0x0000 0000
L1PIWC	RW	32	0x0004 4024	0x0184 4024	0x0000 0000
L1DWIBAR	W	32	0x0004 4030	0x0184 4030	0x0000 0000
L1DWIWC	RW	32	0x0004 4034	0x0184 4034	0x0000 0000
L1DWBAR	W	32	0x0004 4040	0x0184 4040	0x0000 0000
L1DWWC	RW	32	0x0004 4044	0x0184 4044	0x0000 0000
L1DIBAR	W	32	0x0004 4048	0x0184 4048	0x0000 0000
L1DIWC	RW	32	0x0004 404C	0x0184 404C	0x0000 0000
L2WB	RW	32	0x0004 5000	0x0184 5000	0x0000 0000
L2WBINV	RW	32	0x0004 5004	0x0184 5004	0x0000 0000
L2INV	RW	32	0x0004 5008	0x0184 5008	0x0000 0000
L1PINV	RW	32	0x0004 5028	0x0184 5028	0x0000 0000
L1DWB	RW	32	0x0004 5040	0x0184 5040	0x0000 0000
L1DWBINV	RW	32	0x0004 5044	0x0184 5044	0x0000 0000
L1DINV	RW	32	0x0004 5048	0x0184 5048	0x0000 0000
L2EDSTAT	RW	32	0x0004 6004	0x0184 6004	0x0000 0001
L2EDCMD	RW	32	0x0004 6008	0x0184 6008	0x0000 0001
L2EDADDR	RW	32	0x0004 600C	0x0184 600C	0x0000 0000
L2EDCPEC	RW	32	0x0004 6018	0x0184 6018	0x0000 0000
L2EDCNEC	RW	32	0x0004 601C	0x0184 601C	0x0000 0000
MDMAERR	RW	32	0x0004 6020	0x0184 6020	0x0000 0000
MDMAERRCLR	RW	32	0x0004 6024	0x0184 6024	0x0000 0000
L2EDCEN	RW	32	0x0004 6030	0x0184 6030	0x0000 001F
L1PEDSTAT	RW	32	0x0004 6404	0x0184 6404	0x0000 0000
L1PEDCMD	RW	32	0x0004 6408	0x0184 6408	0x0000 0000
L1PEDADDR	RW	32	0x0004 640C	0x0184 640C	0x0000 0000
MARk[(1)]	RW	32	0x0004 8000 + (0x4*k)	0x0184 8000 + (0x4*k)	See [(3)]
L2MPFAR	R	32	0x0004 A000	0x0184 A000	0x0000 0000
L2MPFSR	RW	32	0x0004 A004	0x0184 A004	0x0000 0000
L2MPFCR	RW	32	0x0004 A008	0x0184 A008	0x0000 0000
L2MPPAm[(2)]	RW	32	0x0004 A200 + (0x4*m)	0x0184 A200 + (0x4*m)	0x0000 FFFF
L1PMPFAR	R	32	0x0004 A400	0x0184 A400	0x0000 0000

Table 5-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value(4)
L1PMPFSR	RW	32	0x0004 A404	0x0184 A404	0x0000 0000
L1PMPFCR	RW	32	0x0004 A408	0x0184 A408	0x0000 0000
L1PMPPA16	RW	32	0x0004 A640	0x0184 A640	0x0000 FFFF
L1PMPPA17	RW	32	0x0004 A644	0x0184 A644	0x0000 FFFF
L1PMPPA18	RW	32	0x0004 A648	0x0184 A648	0x0000 FFFF
L1PMPPA19	RW	32	0x0004 A64C	0x0184 A64C	0x0000 FFFF
L1PMPPA20	RW	32	0x0004 A650	0x0184 A650	0x0000 FFFF
L1PMPPA21	RW	32	0x0004 A654	0x0184 A654	0x0000 FFFF
L1PMPPA22	RW	32	0x0004 A658	0x0184 A658	0x0000 FFFF
L1PMPPA23	RW	32	0x0004 A65C	0x0184 A65C	0x0000 FFFF
L1PMPPA24	RW	32	0x0004 A660	0x0184 A660	0x0000 FFFF
L1PMPPA25	RW	32	0x0004 A664	0x0184 A664	0x0000 FFFF
L1PMPPA26	RW	32	0x0004 A668	0x0184 A668	0x0000 FFFF
L1PMPPA27	RW	32	0x0004 A66C	0x0184 A66C	0x0000 FFFF
L1PMPPA28	RW	32	0x0004 A670	0x0184 A670	0x0000 FFFF
L1PMPPA29	RW	32	0x0004 A674	0x0184 A674	0x0000 FFFF
L1PMPPA30	RW	32	0x0004 A678	0x0184 A678	0x0000 FFFF
L1PMPPA31	RW	32	0x0004 A67C	0x0184 A67C	0x0000 FFFF
L1DMPFAR	R	32	0x0004 AC00	0x0184 AC00	0x0000 0000
L1DMPFSR	RW	32	0x0004 AC04	0x0184 AC04	0x0000 0000
L1DMPFCR	RW	32	0x0004 AC08	0x0184 AC08	0x0000 0000
MPLK0	W	32	0x0004 AD00	0x0184 AD00	0x0000 0000
MPLK1	W	32	0x0004 AD04	0x0184 AD04	0x0000 0000
MPLK2	W	32	0x0004 AD08	0x0184 AD08	0x0000 0000
MPLK3	W	32	0x0004 AD0C	0x0184 AD0C	0x0000 0000
MPLKCMD	RW	32	0x0004 AD10	0x0184 AD10	0x0000 0000
MPLKSTAT	RW	32	0x0004 AD14	0x0184 AD14	0x0000 0002
L1DMPPA16	RW	32	0x0004 AE40	0x0184 AE40	0x0000 FFF6
L1DMPPA17	RW	32	0x0004 AE44	0x0184 AE44	0x0000 FFF6
L1DMPPA18	RW	32	0x0004 AE48	0x0184 AE48	0x0000 FFF6
L1DMPPA19	RW	32	0x0004 AE4C	0x0184 AE4C	0x0000 FFF6
L1DMPPA20	RW	32	0x0004 AE50	0x0184 AE50	0x0000 FFF6
L1DMPPA21	RW	32	0x0004 AE54	0x0184 AE54	0x0000 FFF6
L1DMPPA22	RW	32	0x0004 AE58	0x0184 AE58	0x0000 FFF6
L1DMPPA23	RW	32	0x0004 AE5C	0x0184 AE5C	0x0000 FFF6
L1DMPPA24	RW	32	0x0004 AE60	0x0184 AE60	0x0000 FFF6
L1DMPPA25	RW	32	0x0004 AE64	0x0184 AE64	0x0000 FFF6
L1DMPPA26	RW	32	0x0004 AE68	0x0184 AE68	0x0000 FFF6
L1DMPPA27	RW	32	0x0004 AE6C	0x0184 AE6C	0x0000 FFF6
L1DMPPA28	RW	32	0x0004 AE70	0x0184 AE70	0x0000 FFF6
L1DMPPA29	RW	32	0x0004 AE74	0x0184 AE74	0x0000 FFF6
L1DMPPA30	RW	32	0x0004 AE78	0x0184 AE78	0x0000 FFF6
L1DMPPA31	RW	32	0x0004 AE7C	0x0184 AE7C	0x0000 FFF6

(1) k = 0 to 255
(2) m = 0 to 31
(3) MAR0 = 0x0000 0001;

MAR1…MAR11 = 0x0000 0000;
MAR12…MAR15 = 0x0000 000D;
MAR16…MAR255 = 0x0000 000C.

(4) This column considers not ONLY the DSP C66x CorePac specific reset values, but also the device level specific reset values.

5.4.2.2 DSP_ICFG Register Description

For bitfield descriptions of all registers which reside within the DSP_ICFG configuration address space, refer to the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

5.4.3 DSP_SYSTEM Registers

5.4.3.1 DSP_SYSTEM Register Summary

Note

While the DSP1_SYSTEM and DSP2_SYSTEM addresses are part of the device L3_MAIN memory space, **the DSP_SYSTEM addresses are visible only within the DSP_ICFG internal configuration space (visible only to C66x CPU and debug logic).**

Table 5-14. DSP_SYSTEM and DSP1_SYSTEM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_SYSTEM Physical Address DSP1 and DSP2 private	DSP1_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_REVISION	R	32	0x0000 0000	0x01D0 0000	0x40D0 0000
DSP_SYS_HWINFO	R	32	0x0000 0004	0x01D0 0004	0x40D0 0004
DSP_SYS_SYSCONFIG	RW	32	0x0000 0008	0x01D0 0008	0x40D0 0008
DSP_SYS_STAT	R	32	0x0000 000C	0x01D0 000C	0x40D0 000C
DSP_SYS_DISC_CONFIG	RW	32	0x0000 0010	0x01D0 0010	0x40D0 0010
DSP_SYS_BUS_CONFIG	RW	32	0x0000 0014	0x01D0 0014	0x40D0 0014
DSP_SYS_MMU_CONFIG	RW	32	0x0000 0018	0x01D0 0018	0x40D0 0018
DSP_SYS_IRQWAKEEN0	RW	32	0x0000 0020	0x01D0 0020	0x40D0 0020
DSP_SYS_IRQWAKEEN1	RW	32	0x0000 0024	0x01D0 0024	0x40D0 0024
DSP_SYS_DMAWAKEEN0	RW	32	0x0000 0030	0x01D0 0030	0x40D0 0030
DSP_SYS_DMAWAKEEN1	RW	32	0x0000 0034	0x01D0 0034	0x40D0 0034
DSP_SYS_EVTOUT_SET	RW	32	0x0000 0040	0x01D0 0040	0x40D0 0040
DSP_SYS_EVTOUT_CLR	RW	32	0x0000 0044	0x01D0 0044	0x40D0 0044
RESERVED	R	32	0x0000 0048	0x01D0 0048	0x40D0 0048
DSP_SYS_ERRINT_IRQSTATUS_RAW	RW	32	0x0000 0050	0x01D0 0050	0x40D0 0050
DSP_SYS_ERRINT_IRQSTATUS	RW	32	0x0000 0054	0x01D0 0054	0x40D0 0054
DSP_SYS_ERRINT_IRQENABLE_SET	RW	32	0x0000 0058	0x01D0 0058	0x40D0 0058
DSP_SYS_ERRINT_IRQENABLE_CLR	RW	32	0x0000 005C	0x01D0 005C	0x40D0 005C
DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW	RW	32	0x0000 0060	0x01D0 0060	0x40D0 0060
DSP_SYS_EDMAWAKE0_IRQSTATUS	RW	32	0x0000 0064	0x01D0 0064	0x40D0 0064
DSP_SYS_EDMAWAKE0_IRQENABLE_SET	RW	32	0x0000 0068	0x01D0 0068	0x40D0 0068
DSP_SYS_EDMAWAKE0_IRQENABLE_CLR	RW	32	0x0000 006C	0x01D0 006C	0x40D0 006C
DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW	RW	32	0x0000 0070	0x01D0 0070	0x40D0 0070
DSP_SYS_EDMAWAKE1_IRQSTATUS	RW	32	0x0000 0074	0x01D0 0074	0x40D0 0074
DSP_SYS_EDMAWAKE1_IRQENABLE_SET	RW	32	0x0000 0078	0x01D0 0078	0x40D0 0078
DSP_SYS_EDMAWAKE1_IRQENABLE_CLR	RW	32	0x0000 007C	0x01D0 007C	0x40D0 007C
RESERVED	R	32	0x0000 00E0	0x01D0 00E0	0x40D0 00E0
RESERVED	R	32	0x0000 00E4	0x01D0 00E4	0x40D0 00E4
DSP_SYS_HW_DBGOUT_SEL	RW	32	0x0000 00F8	0x01D0 00F8	0x40D0 00F8

Table 5-14. DSP_SYSTEM and DSP1_SYSTEM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_SYSTEM Physical Address DSP1 and DSP2 private	DSP1_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_HW_DBGOUT_VAL	R	32	0x0000 00FC	0x01D0 00FC	0x40D0 00FC

Table 5-15. DSP2_SYSTEM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_REVISION	R	32	0x0000 0000	0x4150 0000
DSP_SYS_HWINFO	R	32	0x0000 0004	0x4150 0004
DSP_SYS_SYSCONFIG	RW	32	0x0000 0008	0x4150 0008
DSP_SYS_STAT	R	32	0x0000 000C	0x4150 000C
DSP_SYS_DISC_CONFIG	RW	32	0x0000 0010	0x4150 0010
DSP_SYS_BUS_CONFIG	RW	32	0x0000 0014	0x4150 0014
DSP_SYS_MMU_CONFIG	RW	32	0x0000 0018	0x4150 0018
DSP_SYS_IRQWAKEEN0	RW	32	0x0000 0020	0x4150 0020
DSP_SYS_IRQWAKEEN1	RW	32	0x0000 0024	0x4150 0024
DSP_SYS_DMAWAKEEN0	RW	32	0x0000 0030	0x4150 0030
DSP_SYS_DMAWAKEEN1	RW	32	0x0000 0034	0x4150 0034
DSP_SYS_EVTOUT_SET	RW	32	0x0000 0040	0x4150 0040
DSP_SYS_EVTOUT_CLR	RW	32	0x0000 0044	0x4150 0044
RESERVED	R	32	0x0000 0048	0x4150 0048
DSP_SYS_ERRINT_IRQSTATUS_RAW	RW	32	0x0000 0050	0x4150 0050
DSP_SYS_ERRINT_IRQSTATUS	RW	32	0x0000 0054	0x4150 0054
DSP_SYS_ERRINT_IRQENABLE_SET	RW	32	0x0000 0058	0x4150 0058
DSP_SYS_ERRINT_IRQENABLE_CLR	RW	32	0x0000 005C	0x4150 005C
DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW	RW	32	0x0000 0060	0x4150 0060
DSP_SYS_EDMAWAKE0_IRQSTATUS	RW	32	0x0000 0064	0x4150 0064
DSP_SYS_EDMAWAKE0_IRQENABLE_SET	RW	32	0x0000 0068	0x4150 0068
DSP_SYS_EDMAWAKE0_IRQENABLE_CLR	RW	32	0x0000 006C	0x4150 006C
DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW	RW	32	0x0000 0070	0x4150 0070
DSP_SYS_EDMAWAKE1_IRQSTATUS	RW	32	0x0000 0074	0x4150 0074
DSP_SYS_EDMAWAKE1_IRQENABLE_SET	RW	32	0x0000 0078	0x4150 0078
DSP_SYS_EDMAWAKE1_IRQENABLE_CLR	RW	32	0x0000 007C	0x4150 007C
RESERVED	R	32	0x0000 00E0	0x4150 00E0
RESERVED	R	32	0x0000 00E4	0x4150 00E4
DSP_SYS_HW_DBGOUT_SEL	RW	32	0x0000 00F8	0x4150 00F8
DSP_SYS_HW_DBGOUT_VAL	R	32	0x0000 00FC	0x4150 00FC

5.4.3.2 DSP_SYSTEM Register Description

Table 5-16. DSP_SYS_REVISION

Address Offset	0x0000 0000		
Physical Address	0x01D0 0000 0x40D0 0000 0x4150 0000	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			

Table 5-16. DSP_SYS_REVISION (continued)

Type	R

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-[(1)]

(1) TI Internal Data

Table 5-17. DSP_SYS_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x01D0 0004 0x40D0 0004 0x4150 0004	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
Type	R		

31–4	3–0
INFO	NUM

Bits	Field Name	Description	Type	Reset
31:4	INFO	0x0: No configurable options in subsystem.	R	0x0
3:0	NUM	Instance Number Set by subsystem input. In a multi-DSP system, provides a unique/incrementing values for each DSP.	R	0x0

Table 5-18. DSP_SYS_SYSCONFIG

Address Offset	0x0000 0008		
Physical Address	0x01D0 0008 0x40D0 0008 0x4150 0008	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
Type	RW		

31–9	8	7–6	5–4	3–2	1–0
RESERVED	RESERVED	RESERVED	STANDBYMODE	IDLEMODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved. Read returns 0.	R	0x00 0000
8	RESERVED	Reserved. User must write 0.	RW	0x0
7:6	RESERVED	Reserved. Read returns 0.	R	0x0

Bits	Field Name	Description	Type	Reset
5:4	STANDBYMODE	0x0: FORCE_STANDBY This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the SAF asserts with minimal hardware condition the "STANDBY" status. It is the responsibility of the software to ensure that the SAF is in a correct quiet state before programming this mode. Additionally when in this mode, the SAF is not allowed to generate wakeup request. 0x1: NO_STANDBY This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the SAF in C66xOSS asserts the "STANDBY" status. 0x2: SMART_STANDBY default. C66xOSS generates the standby status based upon all hardware internal status, namely after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode, the SAF is not allowed to generate wakeup request. 0x3: SMART_STANDBY_WKUP Same as Smart-Standby. (C66xOSS generates the standby status based upon all hardware internal status, namely after having performed all hardware operations necessary to be in a correct quiet state). . Additionally when in this mode, the SAF is allowed to generate wakeup request	RW	0x2
3:2	IDLEMODE	0x0: FORCE_IDLE This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the IAF acknowledges a request to go idle from the power manager with minimal hardware condition. It is the responsibility of the software to ensure that the IAF are in a correct quiet state before requesting a force-idle transition. Additionally when in this mode, the IAF is not allowed to generate any wakeup request. 0x1: NO_IDLE When in this mode, the IAF disregards any request to go idle from the power manager. 0x2: SMART_IDLE default. When in this mode, the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode, the IAF is not allowed to generate any wakeup request 0x3: SMARTIDLEWKUP When in this mode, the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary for the IAF to be in a correct quiet state. Additionally when in this mode, the IAF is allowed to generate wakeup request.	RW	0x2
1:0	RESERVED	Reserved	R	0x0

Table 5-19. DSP_SYS_STAT

Address Offset	0x0000 000C		
Physical Address	0x01D0 000C 0x40D0 000C 0x4150 000C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is intended to provide indication to software (including a remote host) as to whether the DSP is able to enter a low power mode.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	OCPI_DISC_STAT	RESERVED	TC1_STAT	TC0_STAT	C66X_STAT

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5:4	OCPI_DISC_STAT	L3_MAIN (OCP) Initiator(s) Disconnect Status Read 0x0 : OCP inititiator ports are disconnected Read 0x1 : OCP initiator ports are attempting to disconnect. Read 0x2 : OCP initiator ports are active, no request to disconnect is pending.	R	0x2
3	RESERVED		R	0
2	TC1_STAT	EDMA TC1 Status 0x0: IDLE 0x1: ACTIVE - Active, based on inverse of tptc1_mstandby	R	1
1	TC0_STAT	EDMA TC0 Status 0x0: IDLE 0x1: ACTIVE - Active, based on inverse of tptc0_mstandby	R	1
0	C66X_STAT	C66x Status 0x0: IDLE C66x core is idle 0x1: ACTIVE C66x core is active.	R	1

Table 5-20. DSP_SYS_DISC_CONFIG

Address Offset	0x0000 0010		
Physical Address	0x01D0 0010 0x40D0 0010 0x4150 0010	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to manually disconnect the OCP busses.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															OCPI_DISC

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	OCPI_DISC	OCP Initiator (on L3_MAIN) Disconnect request Read 0: Disconnect not in progress, or has completed. Write 0: No effect. Read 1: Disconnect request is in progress. Write 1: Request for OCP Initiator to disconnect and mask write byte enable signals.	RW	0

Table 5-21. DSP_SYS_BUS_CONFIG

Address Offset	0x0000 0014

Table 5-21. DSP_SYS_BUS_CONFIG (continued)

Physical Address	0x01D0 0014 0x40D0 0014 0x4150 0014	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register controls the burst and priority settings for the internal initiators.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	SDMA_PRI			RESERVED			NOPOSTOVERRIDE	RESERVED		SDMA_L2PRES		RESERVED		CFG_L2PRES		RESERVED		TC1_L2PRES		RESERVED		TC0_L2PRES		RESERVED		TC1_DBS		RESERVED		TC0_DBS	

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:28	SDMA_PRI	Sets the CBA/VBusM Priority for the DSP C66x CorePac SDMA port. Can typically be left at default value. 0x0 is highest, ..., 0x7 is lowest priority.	RW	0x4
27:25	RESERVED		R	0x0
24	NOPOSTOVERRIDE	OCP Posted Write vs Non-Posted Write override 0x0: MIX Posted writes are used for cacheable write transactions. Non-posted writes are used for non-cacheable write transactions. 0x1: NOPOST Non-posted writes are used exclusively.	RW	1
23:22	RESERVED		R	0x0
21:20	SDMA_L2PRES	OCP Target port L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect. 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
19:18	RESERVED		R	0x0
17:16	CFG_L2PRES	DSP CFG L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
15:14	RESERVED		R	0x0
13:12	TC1_L2PRES	TC1 L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
11:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:8	TC0_L2PRES	TC0 L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect. 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
7:6	RESERVED		R	0x0
5:4	TC1_DBS	TC1 Default Burst size. 0x0: BYTE_16 - "16-Byte" bursts 0x1: BYTE_32 - "32-Byte" bursts 0x2: BYTE_64 - "64-Byte" bursts 0x3: BYTE_128 - "128-Byte" bursts	RW	0x3
3:2	RESERVED		R	0x0
1:0	TC0_DBS	TC0 Default Burst size 0x0: BYTE_16 - "16-Byte" bursts 0x1: BYTE_32 - "32-Byte" bursts 0x2: BYTE_64 - "64-Byte" bursts 0x3: BYTE_128 - "128-Byte" bursts	RW	0x3

Table 5-22. DSP_SYS_MMU_CONFIG

Address Offset	0x0000 0018		
Physical Address	0x01D0 0018 0x40D0 0018 0x4150 0018	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to enable the subsystem MMUs.		
Type	RW		

31:13	12	11:9	8	7:5	4	3:1	0
RESERVED	MMU1_ABORT	RESERVED	MMU0_ABORT	RESERVED	MMU1_EN	RESERVED	MMU0_EN

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	MMU1_ABORT	MMU1 Abort 0x0: NOABORT = Abort not requested. 0x1: ABORT = MMU abort requested. Can be used in case of page translation failure or lockup.	RW	0
11:9	RESERVED		R	0x0
8	MMU0_ABORT	MMU0 Abort 0x0: NOABORT = Abort not requested. 0x1: ABORT = MMU abort requested. Can be used in case of page translation failure or lockup.	RW	0
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	MMU1_EN	MMU1 Enable 0x0: DISABLED = MMU is disabled and the MMU IP is bypassed. 0x1: ENABLED = MMU is enabled. (The MMU mmrs (including Enable bit) need to be set in addition to this bit.)	RW	0
3:1	RESERVED		R	0x0
0	MMU0_EN	MMU0 Enable 0x0: DISABLED = MMU is disabled and the MMU IP is bypassed. 0x1: ENABLED = MMU is enabled. (The MMU mmrs (including Enable bit) need to be set in addition to this bit.)	RW	0

Table 5-23. DSP_SYS_IRQWAKEEN0

Address Offset	0x0000 0020		
Physical Address	0x01D0 0020 0x40D0 0020 0x4150 0020	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global interrupt wakeup enable bit vector that defines which input interrupts are used to cause a wake from powerdown state (via the slave idle protocol). IRQWAKEEN0 is for interrupt inputs 63 thru 32, and IRQWAKEEN1 is for interrupt inputs 95 thru 64. Internal interrupts cannot cause a wake condition since in this state there is no guaranteed clock and all sub-modules should be in idle state. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding interrupt for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable bit vector for interrupt #n+32 0x0: DISABLE = Interrupt #n+32 disabled for wakeup 0x1: ENABLE = Interrupt #n+32 enabled for wakeup	RW	0x0

Table 5-24. DSP_SYS_IRQWAKEEN1

Address Offset	0x0000 0024		
Physical Address	0x01D0 0024 0x40D0 0024 0x4150 0024	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global interrupt wakeup enable bit vector that defines which input interrupts are used to cause a wake from powerdown state (via the slave idle protocol). IRQWAKEEN0 is for interrupt inputs 63 thru 32, and IRQWAKEEN1 is for interrupt inputs 95 thru 64. Internal interrupts cannot cause a wake condition since in this state there is no guaranteed clock and all sub-modules should be in idle state. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding interrupt for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable bit vector for interrupt #n+64 0x0: DISABLE = Interrupt #n+64 disabled for wakeup 0x1: ENABLE = Interrupt #n+64 enabled for wakeup	RW	0x0

Table 5-25. DSP_SYS_DMAWAKEEN0

Address Offset	0x0000 0030		
Physical Address	0x01D0 0030 0x40D0 0030 0x4150 0030	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global dma event wakeup enable bit vector that defines which input dma events are used to cause a wake from powerdown state (via the slave idle protocol). DMAWAKEEN0 is for dma event inputs 31 thru 0, and DMAWAKEEN1 is for dma event inputs 63 thru 32. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding dma event for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable for event #n 0x0: DISABLE = Interrupt #n disabled for wakeup 0x1: ENABLE = Interrupt #n enabled for wakeup	RW	0x0

Table 5-26. DSP_SYS_DMAWAKEEN1

Address Offset	0x0000 0034		
Physical Address	0x01D0 0034 0x40D0 0034 0x4150 0034	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global dma event wakeup enable bit vector that defines which input dma events are used to cause a wake from powerdown state (via the slave idle protocol). DMAWAKEEN0 is for dma event inputs 31 thru 0, and DMAWAKEEN1 is for dma event inputs 63 thru 32. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding dma event for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable for event #n+32 0x0: DISABLE = Interrupt #n+32 disabled for wakeup 0x1: ENABLE = Interrupt #n+32 enabled for wakeup	RW	0x0

Table 5-27. DSP_SYS_EVTOUT_SET

Address Offset	0x0000 0040		
Physical Address	0x01D0 0040 0x40D0 0040 0x4150 0040	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	These registers can be used to drive event outputs from the DSP subsystem to a desired state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EVENT

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Output Event for event #n Write 0x00 0001: Drive output event #n high/1. Read 0x00 0000: Event #n is low/0. Read 0x00 0001 : Event #n is high/1. Write 0x00 0000 : No action.	RW	0x0

Table 5-28. DSP_SYS_EVTOUT_CLR

Address Offset	0x0000 0044		
Physical Address	0x01D0 0044 0x40D0 0044 0x4150 0044	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	These registers can be used to drive event outputs from the DSP subsystem to a desired state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Output Event for event #n Read 0x00 0000: Event #n is low/0. Write 0x00 0001: Drive output event #n low/0. Read 0x00 0001 : Event #n is high/1. Write 0x00 0000 : No action.	RW	0x0

Table 5-29. DSP_SYS_ERRINT_IRQSTATUS_RAW

Address Offset	0x0000 0050		
Physical Address	0x01D0 0050 0x40D0 0050 0x4150 0050	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													EVENT																		

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	EVENT	Settable raw status for event #n Read 0x00 0000 : No event pending Write 0x00 0000 : No action Read 0x00 0001: Event pending Write 0x00 0001: Set event (for debug)	RW	0x0

Table 5-30. DSP_SYS_ERRINT_IRQSTATUS

Address Offset	0x0000 0054		
Physical Address	0x01D0 0054 0x40D0 0054 0x4150 0054	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		

Table 5-30. DSP_SYS_ERRINT_IRQSTATUS (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													EVENT																		

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	EVENT	Clearable, enabled status for event #n Read 0x00 0000 : No enabled event pending Write 0x00 0000 : No action Read 0x00 0001 : Enabled Event pending Write 0x00 0001 : Clear raw event	RW	0x0

Table 5-31. DSP_SYS_ERRINT_IRQENABLE_SET

Address Offset	0x0000 0058		
Physical Address	0x01D0 0058 0x40D0 0058 0x4150 0058	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													ENABLE																		

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	ENABLE	Enable for event #n Read 0x00 0000 : Interrupt disabled Write 0x00 0000 : No action Read 0x00 0001 : Interrupt enabled Write 0x00 0001 : Enable interrupt	RW	0x0

Table 5-32. DSP_SYS_ERRINT_IRQENABLE_CLR

Address Offset	0x0000 005C		
Physical Address	0x01D0 005C 0x40D0 005C 0x4150 005C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													ENABLE																		

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:0	ENABLE	Enable for event #n Read 0x00 0000 : Interrupt disabled Write 0x00 0000 : No action Read 0x00 0001 : Interrupt enabled Write 0x00 0001 : Disable interrupt (i.e., clear ENABLEn bit)	RW	0x0

Table 5-33. DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW

Address Offset	0x0000 0060		
Physical Address	0x01D0 0060 0x40D0 0060 0x4150 0060	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Settable raw status for event #n Read 0x0000 0000 : No event pending Write 0x0000 0001 : Set event (for debug) Read 0x0000 0001 : Event pending Write 0x0000 0000 : No action	RW	0x0

Table 5-34. DSP_SYS_EDMAWAKE0_IRQSTATUS

Address Offset	0x0000 0064		
Physical Address	0x01D0 0064 0x40D0 0064 0x4150 0064	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable, enabled status for event #n Read 0x0000 0000 : No enabled event pending Write 0x0000 0001 : Clear raw event Read 0x0000 0001 : Enabled Event pending Write 0x0000 0000 : No action	RW	0x0

Table 5-35. DSP_SYS_EDMAWAKE0_IRQENABLE_SET

Address Offset	0x0000 0068		
Physical Address	0x01D0 0068 0x40D0 0068 0x4150 0068	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ENABLE

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Enable interrupt Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0000 0000

Table 5-36. DSP_SYS_EDMAWAKE0_IRQENABLE_CLR

Address Offset	0x0000 006C		
Physical Address	0x01D0 006C 0x40D0 006C 0x4150 006C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Disable interrupt (i.e., clear ENABLEn bit) Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0

Table 5-37. DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW

Address Offset	0x0000 0070		
Physical Address	0x01D0 0070 0x40D0 0070 0x4150 0070	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Settable raw status for event #n+32 Read 0x0000 0000: No event pending Write 0x0000 0001: Set event (for debug) Read 0x0000 0001: Event pending Write 0x0000 0000 : No action	RW	0x0

Table 5-38. DSP_SYS_EDMAWAKE1_IRQSTATUS

Address Offset	0x0000 0074		
Physical Address	0x01D0 0074 0x40D0 0074 0x4150 0074	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		

Table 5-38. DSP_SYS_EDMAWAKE1_IRQSTATUS (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable, enabled status for event #n+32 Read 0x0000 0000: No enabled event pending Write 0x0000 0001: Clear raw event Read 0x0000 0001: Enabled Event pending Write 0x0000 0000: No action	RW	0x0

Table 5-39. DSP_SYS_EDMAWAKE1_IRQENABLE_SET

Address Offset	0x0000 0078		
Physical Address	0x01D0 0078 0x40D0 0078 0x4150 0078	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n+32 Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Enable interrupt Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0

Table 5-40. DSP_SYS_EDMAWAKE1_IRQENABLE_CLR

Address Offset	0x0000 007C		
Physical Address	0x01D0 007C 0x40D0 007C 0x4150 007C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n+32 Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Disable interrupt (i.e., clear ENABLEn bit) Read 0x0000 0001: Interrupt enabled Write 0x0000 0000 : No action	RW	0x0

Table 5-41. DSP_SYS_HW_DBGOUT_SEL

Address Offset	0x0000 00F8

Table 5-41. DSP_SYS_HW_DBGOUT_SEL (continued)

Physical Address	0x01D0 00F8 0x40D0 00F8 0x4150 00F8	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	THis register is used to select which group of internal signals are mapped to the hw_dbgout output bus.		
Type	RW		

31–4	3–0
RESERVED	GROUP

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	GROUP	Debug Group output control mux select 0x0 : Disabled, debug outputs driven to 0x0. 0x1 : G1 = select output group 1 0x2 : G2 = select output group 2 N: GN = select output group N	RW	0x0

Table 5-42. DSP_SYS_HW_DBGOUT_VAL

Address Offset	0x0000 00FC		
Physical Address	0x01D0 00FC 0x40D0 00FC 0x4150 00FC	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to read the value of the currently selected debug output group.		
Type	R		

31–0
VALUE

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Read returns state of hw_dbgout bus	R	0x0

5.4.4 DSP_FW_L2_NOC_CFG Registers

This section covers the DSPSS level defined public firewall (DSP_MMU0, DSP_MMU1) and L2 interconnect (DSP_NoC) functional registers.

5.4.4.1 DSP_FW_L2_NOC_CFG Register Summary

Note

While the DSP1_FW_L2_NOC_CFG and DSP2_FW_L2_NOC_CFG addresses are part of the device L3_MAIN memory space, **the DSP_FW_L2_NOC_CFG addresses are visible only within the DSP_ICFG internal configuration space (visible only to C66x CPU and debug logic).**

Table 5-43. DSP_FW_L2_NOC_CFG and DSP1_FW_L2_NOC_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_FW_L2_NOC_CFG DSP1 and DSP2 private	DSP1_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_50 3000_ERROR_LOG_0	RW	32	0x0000 0000	0x01D0 3000	0x40D0 3000
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_50 3000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 0004	0x01D0 3004	0x40D0 3004
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_50 3000_REGUPDATE_CONTROL	RW	32	0x0000 0040	0x01D0 3040	0x40D0 3040

Table 5-43. DSP_FW_L2_NOC_CFG and DSP1_FW_L2_NOC_CFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_FW_L2_NOC_CFG DSP1 and DSP2 private	DSP1_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 0088	0x01D0 3088	0x40D0 3088
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 008C	0x01D0 308C	0x40D0 308C
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1	RW	32	0x0000 0090	0x01D0 3090	0x40D0 3090
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1	RW	32	0x0000 0094	0x01D0 3094	0x40D0 3094
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1	RW	32	0x0000 0098	0x01D0 3098	0x40D0 3098
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1	RW	32	0x0000 009C	0x01D0 309C	0x40D0 309C
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0	RW	32	0x0000 1000	0x01D0 4000	0x40D0 4000
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 1004	0x01D0 4004	0x40D0 4004
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL	RW	32	0x0000 1040	0x01D0 4040	0x40D0 4040
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 1088	0x01D0 4088	0x40D0 4088
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 108C	0x01D0 408C	0x40D0 408C
DSPNOC_FLAGMUX_ID_COREID	R	32	0x0000 4000	0x01D0 7000	0x40D0 7000
DSPNOC_FLAGMUX_ID_REVISIONID	R	32	0x0000 4004	0x01D0 7004	0x40D0 7004
DSPNOC_FLAGMUX_FAULTEN	RW	32	0x0000 4008	0x01D0 7008	0x40D0 7008
DSPNOC_FLAGMUX_FAULTSTATUS	R	32	0x0000 400C	0x01D0 700C	0x40D0 700C
DSPNOC_FLAGMUX_FLAGINEN0	RW	32	0x0000 4010	0x01D0 7010	0x40D0 7010
DSPNOC_FLAGMUX_FLAGINSTATUS0	R	32	0x0000 4014	0x01D0 7014	0x40D0 7014
DSPNOC_ERRORLOG_ID_COREID	R	32	0x0000 4200	0x01D0 7200	0x40D0 7200
DSPNOC_ERRORLOG_ID_REVISIONID	R	32	0x0000 4204	0x01D0 7204	0x40D0 7204
DSPNOC_ERRORLOG_FAULTEN	RW	32	0x0000 4208	0x01D0 7208	0x40D0 7208
DSPNOC_ERRORLOG_ERRVLD	R	32	0x0000 420C	0x01D0 720C	0x40D0 720C
DSPNOC_ERRORLOG_ERRCLR	RW	32	0x0000 4210	0x01D0 7210	0x40D0 7210
DSPNOC_ERRORLOG_ERRLOG0	R	32	0x0000 4214	0x01D0 7214	0x40D0 7214
DSPNOC_ERRORLOG_ERRLOG1	R	32	0x0000 4218	0x01D0 7218	0x40D0 7218
DSPNOC_ERRORLOG_ERRLOG3	R	32	0x0000 4220	0x01D0 7220	0x40D0 7220
DSPNOC_ERRORLOG_ERRLOG5	R	32	0x0000 4228	0x01D0 7228	0x40D0 7228

Table 5-44. DSP2_FW_L2_NOC_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0	RW	32	0x0000 0000	0x4150 3000
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 0004	0x4150 3004
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4150 3040
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 0088	0x4150 3088
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 008C	0x4150 308C

Table 5-44. DSP2_FW_L2_NOC_CFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1	RW	32	0x0000 0090	0x4150 3090
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1	RW	32	0x0000 0094	0x4150 3094
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1	RW	32	0x0000 0098	0x4150 3098
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1	RW	32	0x0000 009C	0x4150 309C
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0	RW	32	0x0000 1000	0x4150 4000
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 1004	0x4150 4004
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL	RW	32	0x0000 1040	0x4150 4040
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 1088	0x4150 4088
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 108C	0x4150 408C
DSPNOC_FLAGMUX_ID_COREID	R	32	0x0000 4000	0x4150 7000
DSPNOC_FLAGMUX_ID_REVISIONID	R	32	0x0000 4004	0x4150 7004
DSPNOC_FLAGMUX_FAULTEN	RW	32	0x0000 4008	0x4150 7008
DSPNOC_FLAGMUX_FAULTSTATUS	R	32	0x0000 400C	0x4150 700C
DSPNOC_FLAGMUX_FLAGINEN0	RW	32	0x0000 4010	0x4150 7010
DSPNOC_FLAGMUX_FLAGINSTATUS0	R	32	0x0000 4014	0x4150 7014
DSPNOC_ERRORLOG_ID_COREID	R	32	0x0000 4200	0x4150 7200
DSPNOC_ERRORLOG_ID_REVISIONID	R	32	0x0000 4204	0x4150 7204
DSPNOC_ERRORLOG_FAULTEN	RW	32	0x0000 4208	0x4150 7208
DSPNOC_ERRORLOG_ERRVLD	R	32	0x0000 420C	0x4150 720C
DSPNOC_ERRORLOG_ERRCLR	RW	32	0x0000 4210	0x4150 7210
DSPNOC_ERRORLOG_ERRLOG0	R	32	0x0000 4214	0x4150 7214
DSPNOC_ERRORLOG_ERRLOG1	R	32	0x0000 4218	0x4150 7218
DSPNOC_ERRORLOG_ERRLOG3	R	32	0x0000 4220	0x4150 7220
DSPNOC_ERRORLOG_ERRLOG5	R	32	0x0000 4228	0x4150 7228

5.4.4.2 DSP_FW_L2_NOC_CFG Register Description

Table 5-45. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0

Address Offset	0x0000 0000		
Physical Address	0x01D0 3000 0x40D0 3000 0x4150 3000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Error log register		
Type	RW		

31–28	27	26	25–22	21–18	17–0
RESERVED	BLK_BURST_VIOLATION	RESERVED	REGION_START_ERRLOG	REGION_END_ERRLOG	REQINFO_ERRLOG

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	BLK_BURST_VIOLATION	2D burst not allowed or exceeding allowed size	RW	0x0
26	RESERVED		R	0x0
25:21	REGION_START_ERRLOG	Wrong access hit this region number	RW	0x0
20:16	REGION_END_ERRLOG	Wrong access hit this region number	RW	0x0
15:0	REQINFO_ERRLOG	Error in reqinfo vector	RW	0x0

Table 5-46. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0

Address Offset	0x0000 0004		
Physical Address	0x01D0 3004 0x40D0 3004 0x4150 3004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Logical Physical Address Error log register		
Type	R		

31:28	27:0
RESERVED	SLVOFS_LOGICAL

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	SLVOFS_LOGICAL	Address generated by the Arm before being translated	R	0x0

Table 5-47. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL

Address Offset	0x0000 0040		
Physical Address	0x01D0 3040 0x40D0 3040 0x4150 3040	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Register update control register		
Type	RW		

31	30:2	1	0
RESERVED	RESERVED	FW_LOAD_REQ	FW_UPDATE_REQ

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:2	RESERVED	Reserved	R	0x0
1	FW_LOAD_REQ	HW set/SW clear	RW	0x1
0	FW_UPDATE_REQ	HW set/SW clear	RW	0x1

Table 5-48. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0

Address Offset	0x0000 0088		
Physical Address	0x01D0 3088 0x40D0 3088 0x4150 3088	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	MRM_PERMISSION_REGION_0_LOW register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 5-49. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0

Address Offset	0x0000 008C		
Physical Address	0x01D0 308C 0x40D0 308C 0x4150 308C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_HIGH register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1

Bits	Field Name	Description	Type	Reset
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 5-50. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1

Address Offset	0x0000 0090		
Physical Address	0x01D0 3090 0x40D0 3090 0x4150 3090	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Start physical address of region 1		
Type	RW		

31–4	3–0
RESERVED	START_REGION_1

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	START_REGION_1	Physical target start address of firewall region 1	RW	0x0

Table 5-51. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1

Address Offset	0x0000 0094		
Physical Address	0x01D0 3094 0x40D0 3094 0x4150 3094	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	End physical address of region 1		
Type	RW		

31	30–4	3–0
END_REGION_1_ENABLE	RESERVED	END_REGION_1

Bits	Field Name	Description	Type	Reset
31	END_REGION_1_ENABLE	End Region 1 enable	RW	0x0
30:4	RESERVED		R	0x0
3:0	END_REGION_1	Physical target end address of firewall region 1	RW	0x0

Table 5-52. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1

Address Offset	0x0000 0098		
Physical Address	0x01D0 3098	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_1_LOW register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td>PUB_PRV_DEBUG</td><td>PUB_USR_DEBUG</td><td colspan="2">RESERVED</td><td>PUB_PRV_READ</td><td>PUB_PRV_WRITE</td><td>PUB_PRV_EXE</td><td>PUB_USR_READ</td><td>PUB_USR_WRITE</td><td>PUB_USR_EXE</td><td colspan="6">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 5-53. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1

Address Offset	0x0000 009C		
Physical Address	0x01D0 309C 0x40D0 309C 0x4150 309C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_1_HIGH register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1

Bits	Field Name	Description	Type	Reset
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 5-54. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0

Address Offset	0x0000 1000		
Physical Address	0x01D0 4000 0x40D0 4000 0x4150 4000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Error log register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>BLK_BURST_VIOLATION</td><td>RESERVED</td><td colspan="5">REGION_START_ERRLOG</td><td colspan="5">REGION_END_ERRLOG</td><td colspan="16">REQINFO_ERRLOG</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27	BLK_BURST_VIOLATION	2D burst not allowed or exceeding allowed size	RW	0x0
26	RESERVED		R	0x0
25:21	REGION_START_ERRLOG	Wrong access hit this region number	RW	0x0
20:16	REGION_END_ERRLOG	Wrong access hit this region number	RW	0x0
15:0	REQINFO_ERRLOG	Error in reqinfo vector	RW	0x0

Table 5-55. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0

Address Offset	0x0000 1004		
Physical Address	0x01D0 4004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Logical Physical Address Error log register		
Type	R		

31:28	27:0
RESERVED	SLVOFS_LOGICAL

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	SLVOFS_LOGICAL	Address generated by the Arm before being translated	R	0x0

Table 5-56. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL

Address Offset	0x0000 1040		
Physical Address	0x01D0 4040	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Register update control register		
Type	RW		

31	30:2	1	0
RESERVED	RESERVED	FW_LOAD_REQ	FW_UPDATE_REQ

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:2	RESERVED	Reserved	R	0x0
1	FW_LOAD_REQ	HW set/SW clear	RW	0x1
0	FW_UPDATE_REQ	HW set/SW clear	RW	0x1

Table 5-57. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0

Address Offset	0x0000 1088		
Physical Address	0x01D0 4088 0x40D0 4088 0x4150 4088	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_LOW register		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

RESERVED	PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED	PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 5-58. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

Address Offset	0x0000 108C		
Physical Address	0x01D0 408C 0x40D0 408C 0x4150 408C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_HIGH register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1

Bits	Field Name	Description	Type	Reset
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 5-59. DSPNOC_FLAGMUX_ID_COREID

Address Offset	0x0000 4000		
Physical Address	0x01D0 7000 0x40D0 7000 0x4150 7000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31–8	7–0
CORECHECKSUM	CORETYPEID

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0xff71d7
7:0	CORETYPEID	Field identifying the type of IP.	R	0xb

Table 5-60. DSPNOC_FLAGMUX_ID_REVISIONID

Address Offset	0x0000 4004		
Physical Address	0x01D0 7004 0x40D0 7004 0x4150 7004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision.	R	0x-(1)

(1) TI internal data

Table 5-61. DSPNOC_FLAGMUX_FAULTEN

Address Offset	0x0000 4008		
Physical Address	0x01D0 7008 0x40D0 7008 0x4150 7008	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			

Table 5-61. DSPNOC_FLAGMUX_FAULTEN (continued)

Type	RW

31–1	0
RESERVED	FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Global Fault Enable register	RW	0x1

Table 5-62. DSPNOC_FLAGMUX_FAULTSTATUS

Address Offset	0x0000 400C		
Physical Address	0x01D0 700C 0x40D0 700C 0x4150 700C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31–1	0
RESERVED	FAULTSTATUS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTSTATUS	Global Fault Status register	R	0x0

Table 5-63. DSPNOC_FLAGMUX_FLAGINEN0

Address Offset	0x0000 4010		
Physical Address	0x01D0 7010 0x40D0 7010 0x4150 7010	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		

31–1	0
RESERVED	FLAGINEN0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINEN0	FlagIn Enable register #0	RW	0x1

Table 5-64. DSPNOC_FLAGMUX_FLAGINSTATUS0

Address Offset	0x0000 4014		
Physical Address	0x01D0 7014 0x40D0 7014 0x4150 7014	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			

Table 5-64. DSPNOC_FLAGMUX_FLAGINSTATUS0 (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FLAGINSTATUS0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINSTATUS0	FlagIn Status register #0	R	0x0

Table 5-65. DSPNOC_ERRORLOG_ID_COREID

Address Offset	0x0000 4200		
Physical Address	0x01D0 7200 0x40D0 7200 0x4150 7200	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORECHECKSUM																								CORETYPEID							

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0xaf434
7:0	CORETYPEID	Field identifying the type of IP.	R	0xd

Table 5-66. DSPNOC_ERRORLOG_ID_REVISIONID

Address Offset	0x0000 4204		
Physical Address	0x01D0 7204 0x40D0 7204 0x4150 7204	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision.	R	0x-(1)

(1) TI Internal data

Table 5-67. DSPNOC_ERRORLOG_FAULTEN

Address Offset	0x0000 4208		
Physical Address	0x01D0 7208 0x40D0 7208 0x4150 7208	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Enable Fault output	RW	0x1

Table 5-68. DSPNOC_ERRORLOG_ERRVLD

Address Offset	0x0000 420C		
Physical Address	0x01D0 720C 0x40D0 720C 0x4150 720C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRVLD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRVLD	Error logged Valid	R	0x0

Table 5-69. DSPNOC_ERRORLOG_ERRCLR

Address Offset	0x0000 4210		
Physical Address	0x01D0 7210 0x40D0 7210 0x4150 7210	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRCLR	Clr ErrVld status	RW	0x0

Table 5-70. DSPNOC_ERRORLOG_ERRLOG0

Address Offset	0x0000 4214		
Physical Address	0x01D0 7214 0x40D0 7214 0x4150 7214	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Header: Lock, Opcode, Len1, ErrCode values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FORMAT	RESERVED	LEN1	RESERVED	ERRCODE	RESERVED	OPC	LOCK

Bits	Field Name	Description	Type	Reset
31	FORMAT	Format of ErrLog0 register	R	0x1
30:28	RESERVED		R	0x0
27:16	LEN1	Header: Len1 value	R	0x0
15:11	RESERVED		R	0x0
10:8	ERRCODE	Header: Error Code value	R	0x0
7:5	RESERVED		R	0x0
4:1	OPC	Header: Opcode value	R	0x0
0	LOCK	Header: Lock bit value	R	0x0

Table 5-71. DSPNOC_ERRORLOG_ERRLOG1

Address Offset	0x0000 4218		
Physical Address	0x01D0 7218 0x40D0 7218 0x4150 7218	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31:15	14:0
RESERVED	ERRLOG1

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:0	ERRLOG1	Header: RouteId lsb value	R	0x0

Table 5-72. DSPNOC_ERRORLOG_ERRLOG3

Address Offset	0x0000 4220		
Physical Address	0x01D0 7220 0x40D0 7220 0x4150 7220	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30:0
RESERVED	ERRLOG3

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:0	ERRLOG3	Header: Addr lsb value	R	0x0

Table 5-73. DSPNOC_ERRORLOG_ERRLOG5

Address Offset	0x0000 4228		
Physical Address	0x01D0 7228 0x40D0 7228 0x4150 7228	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			

Table 5-73. DSPNOC_ERRORLOG_ERRLOG5 (continued)

Type								R																							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										ERRLOG5																					

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:0	ERRLOG5	Header: User lsb value	R	0x0

Chapter 6

IVA Subsystem

Note

The IVA-HD subsystem is a set of video encoder and decoder hardware accelerators. The list of supported codecs can be found in the software development kit (SDK) documentation.

The IVA-HD subsystem availability is device part number dependent. Refer to device *Data Manual*, for more information.

Chapter 7

Dual Cortex-M4 IPU Subsystem

This chapter describes the dual Cortex-M4 image processor unit (IPU) subsystem.

7.1 Dual Cortex-M4 IPU Subsystem Overview 1433
7.2 Dual Cortex-M4 IPU Subsystem Integration 1436
7.3 Dual Cortex-M4 IPU Subsystem Functional Description 1441
7.4 Dual Cortex-M4 IPU Subsystem Register Manual 1454

7.1 Dual Cortex-M4 IPU Subsystem Overview

7.1.1 Introduction

The device instantiates two dual Cortex®-M4 image processor unit (IPU) subsystems available for general purpose usage.

Note

The two IPU subsystems are identical from functional point of view. Thus, a unified name **IPUx** shall be used throughout the chapter for simplification.

Each IPU subsystem contains two Arm® Cortex-M4 processors (IPUx_C0 and IPUx_C1) that share a common level 1 (L1) cache (called unicache [IPUx_UNICACHE]). The two Cortex-M4 cores are completely homogeneous to one another. Any task possible using one Cortex-M4 core is also possible using the other Cortex-M4 core. It is software responsibility to distribute the various tasks between each Cortex-M4 core for optimal performance.

The integrated interrupt handling of the IPUx subsystem allows it to function as an efficient control unit.

Figure 7-1 is a high-level block diagram of the IPUx subsystem.

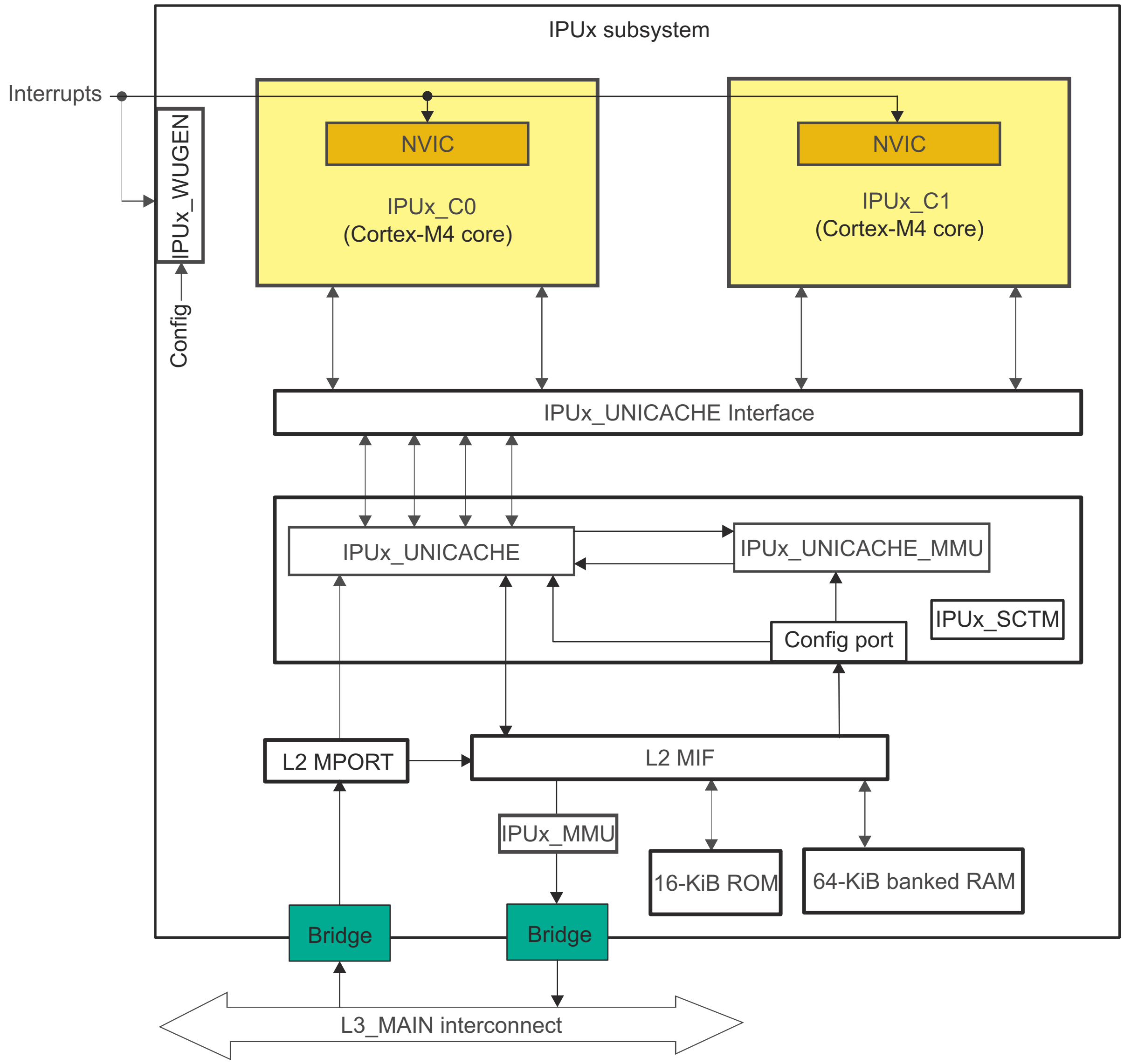


Figure 7-1. IPUx Subsystem Overview

7.1.2 Features

Each IPU subsystem integrates the following:

- Two Arm Cortex-M4 microprocessors (IPUx_C0 and IPUx_C1):
  - Armv7-M and Thumb®-2 instruction set architecture (ISA)
  - Armv6 SIMD and digital signal processor (DSP) extensions
  - Single-cycle MAC
  - Integrated nested vector interrupt controller (NVIC) (also called IPUx_Cx_INTC, where x = 0, 1)
  - Integrated bus matrix
  - Registers:
    - Thirteen general-purpose 32-bit registers
    - Link register (LR)
    - Program counter (PC)
    - Program status register, xPSR
    - Two banked SP registers
  - Integrated power management
  - Extensive debug capabilities
- Unicache interface:
  - Instruction and data interface
  - Supports paralleled accesses
- Level 2 (L2) master interface (MIF) splitter for access to memory or configuration port
- Configuration port: Used for unicache maintenance and unicache memory management unit (IPUx_UNICACHE_MMU) configuration
- Unicache:
  - 32 KiB divided into 16 banks
  - 4-way
  - Cache configuration lock/freeze/preload
  - Internal MMU:
    - 16-entry region-based address translation
    - Read/write control and access type control
    - Execute Never (XN) MMU protection policy
    - Little-endian format
- Subsystem counter timer module (IPUx_UNICACHE_SCTM, or just SCTM)
- On-chip ROM (IPUx_ROM) and banked RAM (IPUx_RAM) memory
- Emulation/debug: Emulation feature embedded in Cortex-M4
- L2 MMU (IPUx_MMU): 32 entries with table walking logic
- Wake-up generator (IPUx_WUGEN): Generates wake-up request from external interrupts
- Power management:
  - Local power-management control: Configurable through the IPUx_WUGEN registers.
  - Three sleep modes supported, controlled by the local power-management module.
  - IPUx is clock-gated in all sleep modes.
  - IPUx_Cx_INTC interrupt interface stays awake.

7.2 Dual Cortex-M4 IPU Subsystem Integration

Figure 7-2 and Figure 7-3 show the integration of IPU1 and IPU2 in the device.

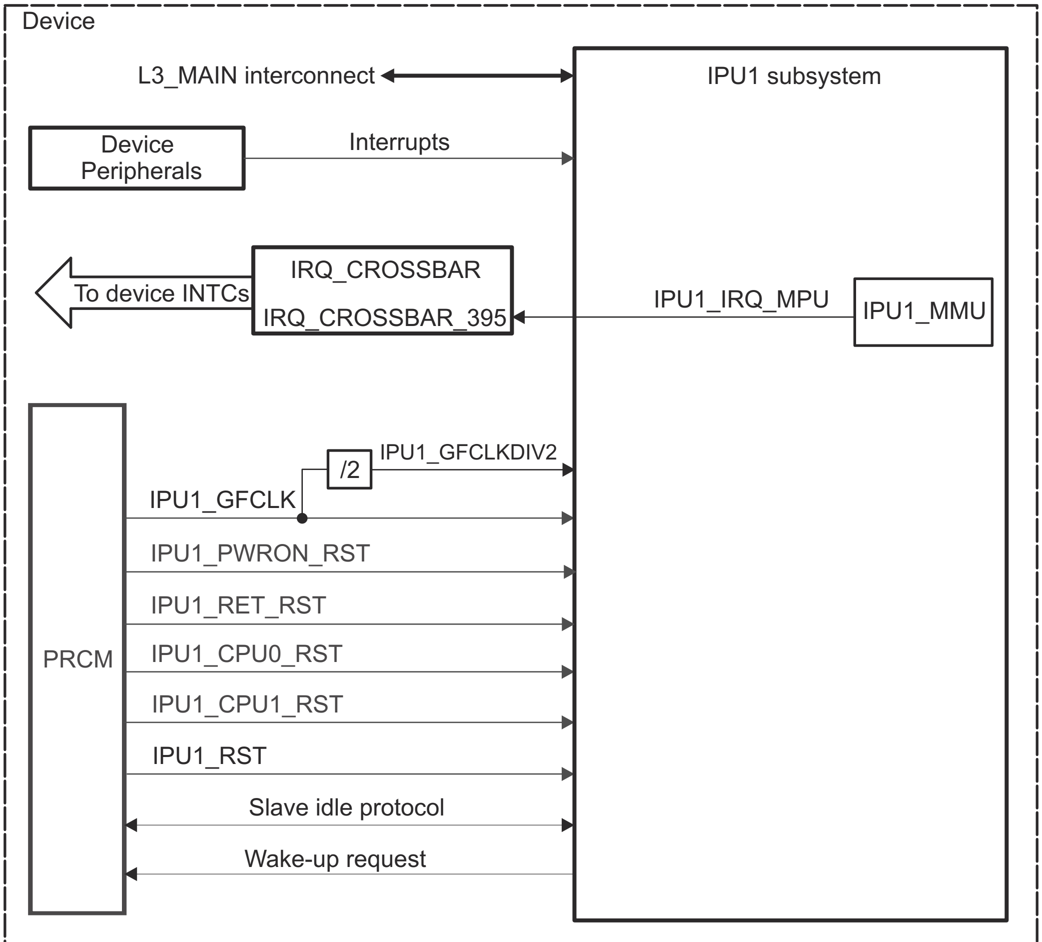


Figure 7-2. IPU1 Subsystem Integration

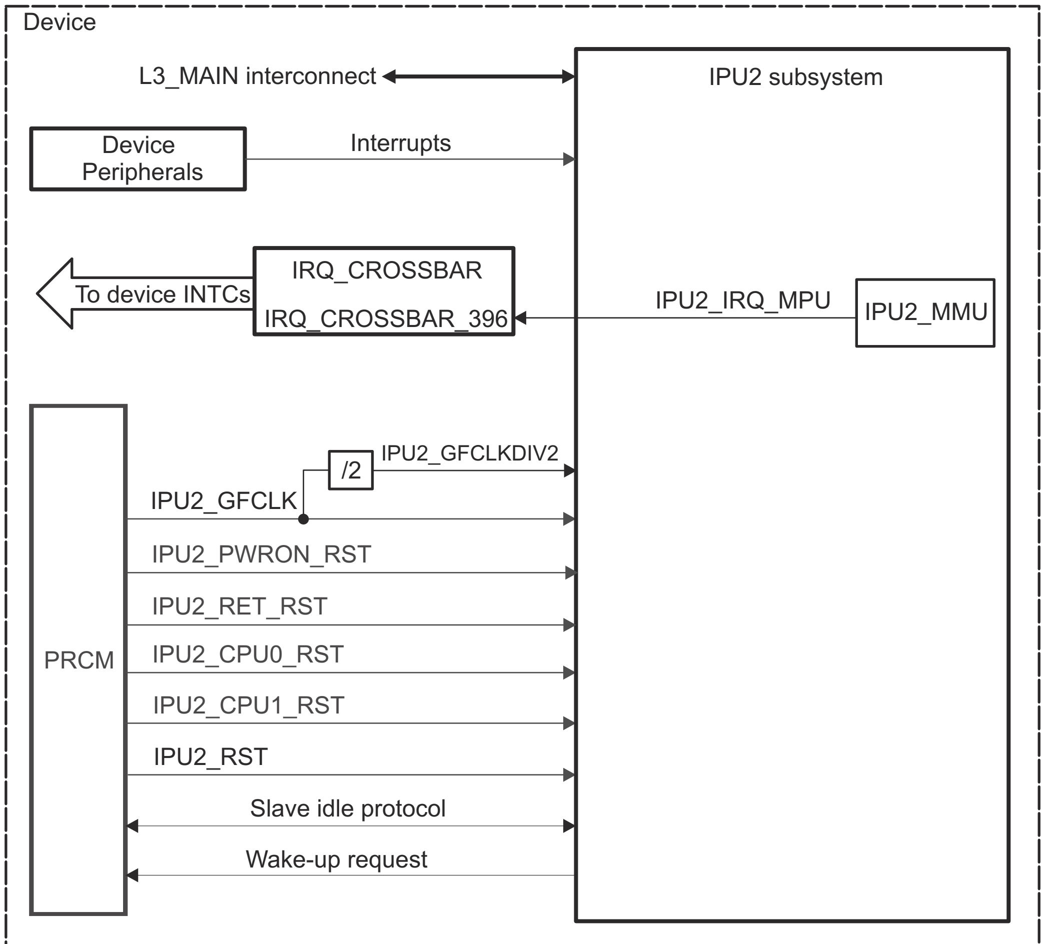


Figure 7-3. IPU2 Subsystem Integration

Table 7-1 through Table 7-3 summarize the integration of the module in the device.

Table 7-1. IPUx Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
IPU1	PD_IPU	L3_MAIN
IPU2	PD_CORE	L3_MAIN

Table 7-2. IPUx Hardware Requests

Interrupt Requests				
Module Instance	Interrupt Name (Source)	IRQ_CROSSBAR Input (Destination)	Default Mapping	Description
IPU1	IPU1_IRQ_MPU	IRQ_CROSSBAR_395	MPU_IRQ_100	IPU1_MMU fault interrupt.
IPU2	IPU2_IRQ_MPU	IRQ_CROSSBAR_396	–	IPU2_MMU fault interrupt. This interrupt is not mapped by default to any device INTC.

Table 7-3. IPUx Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
IPU1	IPU1_GFCLK	IPU1_GFCLK	PRCM module	IPU1 interface and functional clock(s). See Section 7.2.1.1 for details.

Table 7-3. IPUx Clocks and Resets (continued)

	IPU1_GFCLKDIV2	IPU1_GFCLK		
IPU2	IPU2_GFCLK	IPU2_GFCLK	PRCM module	IPU2 interface and functional clock(s). See Section 7.2.1.1 for details.
	IPU2_GFCLKDIV2	IPU2_GFCLK		

Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
IPU1	IPU1_PWRON_RST	IPU1_PWRON_RST	PRCM module	Power-on reset, used to reset the whole IPU1 subsystem
	IPU1_RET_RST	IPU1_RET_RST	PRCM module	Retention reset to few retention logic inside the IPU1_UNICACHE
	IPU1_CPU0_RST	IPU1_CPU0_RST	PRCM module	Reset signal to IPU1_C0
	IPU1_CPU1_RST	IPU1_CPU1_RST	PRCM module	Reset signal to IPU1_C1
	IPU1_RST	IPU1_RST	PRCM module	Reset signal to the IPU1_UNICACHE and the IPU1_MMU
IPU2	IPU2_PWRON_RST	IPU2_PWRON_RST	PRCM module	Power-on reset, used to reset the whole IPU2 subsystem
	IPU2_RET_RST	IPU2_RET_RST	PRCM module	Retention reset to few retention logic inside the IPU2_UNICACHE
	IPU2_CPU0_RST	IPU2_CPU0_RST	PRCM module	Reset signal to IPU2_C0
	IPU2_CPU1_RST	IPU2_CPU1_RST	PRCM module	Reset signal to IPU2_C1
	IPU2_RST	IPU2_RST	PRCM module	Reset signal to the IPU2_UNICACHE and the IPU2_MMU

Note

The “**Default Mapping**” column in Table 7-2 shows the default mapping of module IRQ source signal. This IRQ source signal can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.

For more information about the IRQ_CROSSBAR module, see *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

For more information about clocks, resets, and power domains, see *Power, Reset, and Clock Management*.

7.2.1 Dual Cortex-M4 IPU Subsystem Clock and Reset Distribution

7.2.1.1 Clock Distribution

The IPUx subsystem has two clock inputs:

- **IPUx_GFCLK**: Main source clock for IPUx; feeds most of IPUx internal modules:
  - IPUx Unicache & MMU, L2 MIF & MPORT (all running on (1x) IPUx_GFCLK)
  - Cortex-M4 cores, IPUx_RAM, IPUx_ROM (all running on (1/2x) IPUx_GFCLK via internal divider)
- **IPUx_GFCLKDIV2**: This is IPUx_GFCLK externally divided by two; feeds IPUx L2 MMU

The IPUx_GFCLK itself can be derived from either:

- CORE_IPU_ISS_BOOST_CLK – main clock for IPUx from DPLL_CORE, or
- DPLL_ABE_X2_CLK – alternative clock from DPLL_DDR

For more information, see *Power, Reset, and Clock Management*.

Figure 7-4 shows the clocking scheme of the IPUx subsystem.

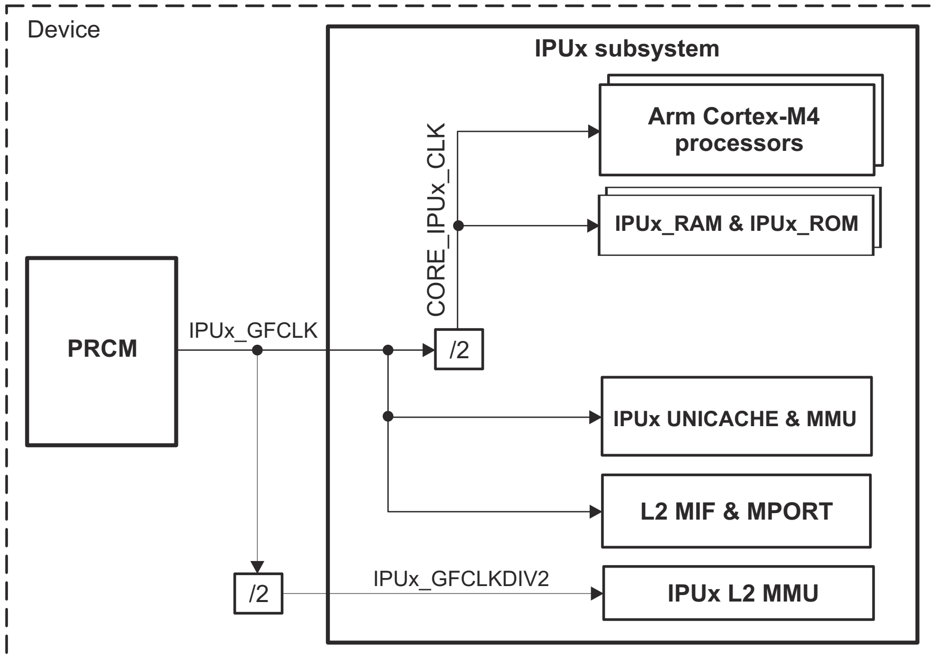


Figure 7-4. IPUx Subsystem Clocking Scheme

7.2.1.2 Reset Distribution

Three reset signals controlled by the power, reset, and clock management (PRCM) module let the two Arm Cortex-M4 processors and the rest of the IPUx subsystem be reset independently. These three reset signals are: IPUx_CPU0_RST, IPUx_CPU1_RST, and IPUx_RST. The Arm Cortex-M4 processors must come out of reset one at a time:

- At IPUx_CPU0_RST, IPUx_C0 comes out of reset, but IPUx_C1 is held in reset.
- IPUx_C0 controls the reset for IPUx_C1 (through the PRCM RM_IPUx_RSTCTRL[1] RST2 bit).

Because IPUx_C0 controls the reset for IPUx_C1 (through the PRCM registers), the code running on IPUx_C0 decides the mode of operation, whether it is:

- Mode 1: One Arm Cortex-M4 processor is running, and the other processor is held on reset.
- Mode 2: The two Arm Cortex-M4 processors are running.

This decision of which mode to use is driven by the use case. If the software partitioning and performance requirement for a use case requires two Cortex-M4 processors to run simultaneously, the user must go to mode 2. If IPUx_C1 is not required for a particular use case, the user can remain in mode 1.

Figure 7-5 shows the reset scheme of the IPUx subsystem.

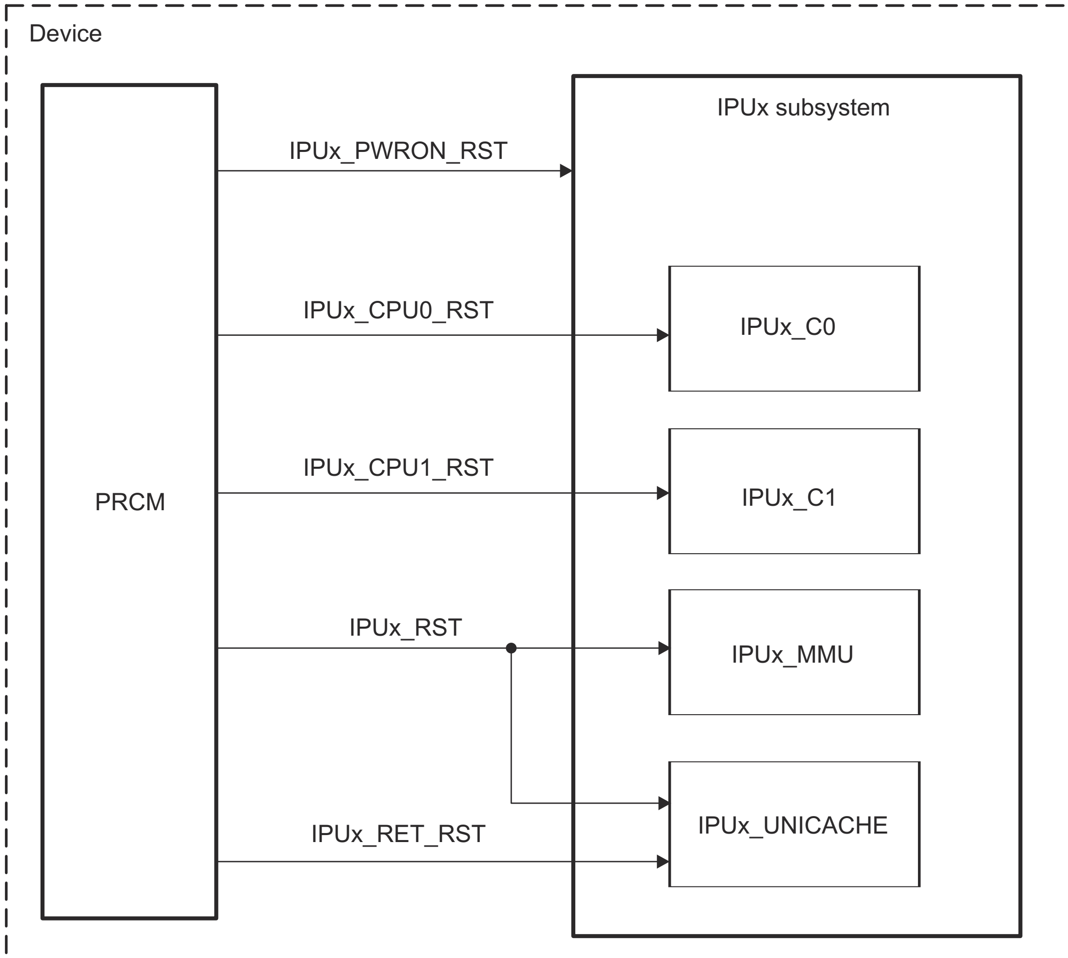


Figure 7-5. IPUx Subsystem Reset Scheme

7.3 Dual Cortex-M4 IPU Subsystem Functional Description

7.3.1 IPUx Subsystem Block Diagram

The IPUx subsystem integrates the following group of submodules:

- Two Arm Cortex-M4 processors: Two cores (r0p1 revision), IPUx_C0 and IPUx_C1. For a description of the Arm Cortex-M4 processor, see the Arm *Cortex-M4 Technical Reference Manual*, available at infocenter.arm.com/help/index.jsp.
- Interrupt controller (IPUx_Cx_INTC): To facilitate parallel processing, the interrupt mapping is the same for the two cores. Each Cortex-M4 processor receives the same interrupts, except for a few internal interrupts. Every IRQ line is shared between the two Arm processors. By properly configuring the IPUx_Cx_INTC registers inside each Arm processor, it can be ensured that the shared IRQ is taken by only one of the Arm processors (for more information, see *Interrupt Controllers*).
- IPUx_UNICACHE interface: The cache interface converts the data between the different protocols in the subsystem. Four ports are required to support the four buses from the Arm Cortex-M4 processors (two for each processor).The instruction and data connections from each Arm Cortex-M4 are multiplexed, but the Arm Cortex-M4 prevents conflicts on this connection. Default cache policies are provided through the sideband signals and are not used to access the cache. Cacheability is provided through the MMU inside the cache.
- IPUx_UNICACHE: Allows basic maintenance operations, which are performed through a dedicated interface: preload, lock, clean (write out dirty lines, but do not invalidate directly), and invalidate.
- IPUx_UNICACHE_MMU: Serves the role of an attribute MMU (AMMU) for the unicache. It provides the multi-access cache with region-based address translation, read/write control, access type control, and multilevel cache maintenance. Access to the IPUx_UNICACHE_MMU is done only under privilege mode. The IPUx_UNICACHE_MMU can be programmed by the dual Cortex-A15 microprocessor unit (MPU) subsystem through the IPUx subsystem slave port.
- IPUx_UNICACHE_SCTM: Embedded in the IPUx_UNICACHE
- Interconnect configuration port: Cache maintenance and MMU configuration are done through an interconnect slave port. Accesses must be performed to a noncacheable area that must be defined within the IPUx_UNICACHE_MMU. Interconnect accesses are generated from the L2 MIF.
- IPUx_MMU: Provides address translation for all the accesses done from the IPUx subsystem to the level 3 (L3_MAIN) interconnect. The IPUx_MMU can be programmed by the MPU subsystem through the IPUx subsystem slave port.
- L3_MAIN interconnect port: Allows access to the system memories and peripherals. For the address mapping of the modules in the IPUx subsystem, see *Memory Mapping*.
- On-chip IPUx_ROM and banked IPUx_RAM memory. The IPUx_ROM memory is used for boot/initialization purposes. For more information about the initialization of the device, see *Initialization*.

Figure 7-6 is a block diagram of the IPUx subsystem.

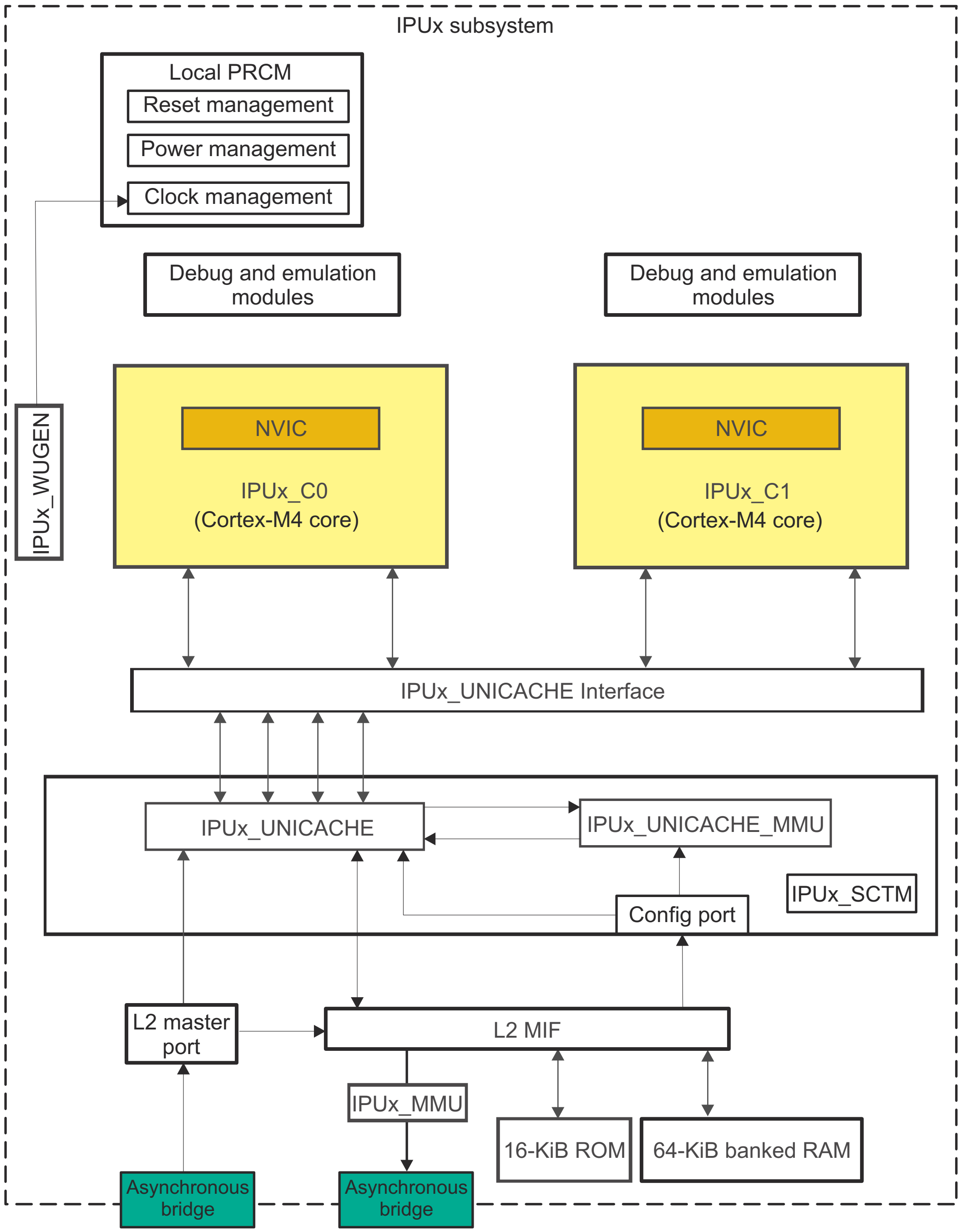


Figure 7-6. IPUx Subsystem Block Diagram

7.3.2 Power Management

7.3.2.1 Local Power Management

The user can configure the local power management through the STANDBY_CORE_SYSCONFIG, IDLE_CORE_SYSCONFIG, and WUGEN_IRQ_EN registers. The user can:

- Configure the different standby modes (the default is smart wake-up standby mode) through the STANDBY_CORE_SYSCONFIG[1:0] STANDBYMODE bit field
- Configure the different idle modes (the default is smart wake-up idle mode) through the IDLE_CORE_SYSCONFIG[1:0] IDLEMODE bit field

- Control which interrupts will cause a wakeup by appropriately configuring the WUGEN_MEVT0 and WUGEN_MEVT1 registers (the default is ALL_MASKED)

The IPUx subsystem provides three sleep modes:

- On mode: Sleep on exit (wait for the interrupt service routine [ISR] to complete)
- Sleep mode: Wait for interrupt to wake up the processor
- Deep-sleep mode: Long duration sleep; phase-locked loop (PLL) can be stopped

During sleep mode, the system clock can be stopped, but the free-running clock input must still be running to allow the processor to be wakened by an interrupt or an event. The sleep modes are invoked by wait for interrupt (WFI) or wait for event (WFE) instructions. The processor clock is automatically stopped, waiting for an interrupt or an event. Deep-sleep mode also stops the processor clock, but this can also be supported by the PRCM module. A combined signal is generated from the two Cortex-M4 processors in deep-sleep mode to initiate another power state and let the PRCM module handle the next power states. At this time, software must ensure that all IPUx_UNICACHE background operations (for example, maintenance) are complete.

Table 7-4 describes local clock gating.

Table 7-4. Local Clock Gating

Cortex-M4 CPU Mode	IPUx_C1 On	IPUx_C1 Sleep	IPUx_C1 Deep Sleep
IPUx_C0 On	On	Functional clock 2 stopped locally	Functional clock 2 stopped locally
IPUx_C0 Sleep	Functional clock 1 stopped locally	Functional clock 1 and clock 2 stopped locally	Functional clock 1 and clock 2 stopped locally
IPUx_C0 Deep Sleep	Functional clock 1 stopped locally	Functional clock 1 and clock 2 stopped locally	Standby request to power-management module

Note

For information about source clock gating and for a description of the sleep/wake-up transitions, see *Power, Reset, and Clock Management*.

7.3.2.2 Power Domains

The IPUx subsystem is divided into two power domains (PD_IPU and PD_COREAON), which are controlled by the PRCM module.

The PD_IPU power domain is the main power domain and includes all the IPUx subsystem components (two Arm Cortex-M4 processors, IPUx_UNICACHE, IPUx_ROM and IPUx_RAM memories, and emulation\debug modules) except the IPUx_WUGEN.

The PD_COREAON power domain is an always-on power domain. The PD_COREAON power domain contains the IPUx_WUGEN, which generates a wake-up request from external interrupts. By this separate PD_COREAON power domain, the wake-up request can be generated even when the PD_IPU power domain is in OFF or RET state.

7.3.2.3

For information about the PD_IPU and PD_COREAON power domains, see *Power, Reset, and Clock Management*.

7.3.2.4 Voltage Domain

The IPUx subsystem is located within the CORE voltage domain (VD_CORE). All IPU logic (Cortex-M4 cores, IPUx_UNICACHE/MMU, IPUx_WUGEN, etc...) is fed by VD_CORE. All IPU memory arrays are fed by on-chip memory (SRAM) LDO dedicated to the CORE domain – SLDO_CORE.

For information about VD_CORE and SLDO_CORE, see *Power, Reset, and Clock Management*.

7.3.2.5 Power States and Modes

Table 7-5 lists the different power modes and the expected states for each power domain.

Table 7-5. IPUx Subsystem Power Modes

		Activity			Power Status	
	Functional Domain	**Core**		**IPUx subsystem**	**PD_DSP Power Domain**	**PD_COREAON Power Domain**
	Modules included	IPUx_C0	IPUx_C1	IPUx_WUGEN		
Power modes	Active	Active	Active	Active	ON	ON
	IPUx_C0 idle	Idle	Active	Active	ON	ON
	IPUx_C1 idle	Active	Idle	Active	ON	ON
	Core standby	Idle	Idle	Active	ON	ON
	Full idle	Idle	Idle	Idle	ON	ON
	CSWR	Idle	Idle	Idle	ON/LOWV	ON/LOWV
	OSWR	Idle	Idle	Idle	RET	ON
	Power off	Idle	Idle	Idle	OFF	OFF

Table 7-6 lists the power mode transitions in the IPUx subsystem.

Table 7-6. Power Mode Transitions

	Active	IPUx_C0 Idle	IPUx_C1 Idle	Standby	Full-Idle	Retention	Power Off
Active		WFE/WFI instruction	WFE/WFI instruction				
IPUx_C0 idle	Events/ interrupts			Deep sleep			
IPUx_C1 idle	Events/ interrupts			Deep sleep			
Standby	Wake-up IRQ				L1/L2/ IPUx_WUGEN functional domain idle		
Full idle	Wake-up IRQ			Wake-up (through interconnect)		PRCM module	PRCM module
CSWR/OSWR					PRCM module or wakeup		PRCM module
Power off					PRCM module or wakeup	PRCM module	

The different power modes and their features are:

- Active mode: All function domains are operative.
- IPUx_C0 and IPUx_C1 idle mode:
  - Only the CPU core is idled (when running WFE/WFI instructions).
  - Only one Cortex-M4 core can be in this mode. Interrupts or events can waken the core.
  - Can go into sleep or deep-sleep mode. Potentially, both cores can be in sleep mode.
  - When both cores are in deep-sleep mode, a standby request is sent to the PRCM module.
  - Software must ensure that all IPUx_UNICACHE background operations (for example, maintenance) are complete before the PRCM module asserts an IdleReq.
- Core standby mode
  - Both cores in the CORE functional domain are in idle mode (an interrupt cannot wake up either of the cores).
  - The PRCM module must have acknowledged its acceptance by an MWait signal.
  - After this handshake, all power management is under the control of the PRCM module.

- Full-idle mode:
  – The IPUx subsystem functional domain is also idled.
  – After coming to this mode, power states can be moved deeper.
- Retention mode:
  – The voltage of the logic supply is lowered to reduce static power consumption by leakage current. The logic power switch in the PD_IPU power domain is still closed (ON); thus, all logic states are retained.
  – L1 and/or L2 memories can go independently into retention depending on the settings done at the PRCM level.
- Power off mode:
  – The voltage source is shut down. The logic states, including the retention logic, are lost.
  – The IPUx_WUGEN is not operating and only the PRCM module can trigger the IPUx subsystem wakeup.
  – Reset must be applied to the IPUx subsystem to restart the two Arm Cortex-M4 processors and the memory subsystem.

7.3.2.6 Wake-Up Generator (IPUx_WUGEN)

The IPUx_WUGEN in the IPUx subsystem enables efficient power management. The IPUx_WUGEN generates a wake-up signal to the PRCM module to enable the IPUx subsystem to recover its functional clocks, which are gated by the PRCM module when at least one request is active. The IPUx_WUGEN can be configured in standby mode or idle mode through the STANDBY_CORE_SYSCONFIG[1:0] STANDBYMODE and IDLE_CORE_SYSCONFIG[1:0] IDLEMODE bit fields.

7.3.2.6.1 IPUx_WUGEN Main Features

The IPUx_WUGEN allows:

- Gating of the IPUx subsystem clock dynamically, thus reducing power consumption
- Simplifying of dependencies in the PRCM module

Figure 7-7 is an overview of the IPUx_WUGEN.

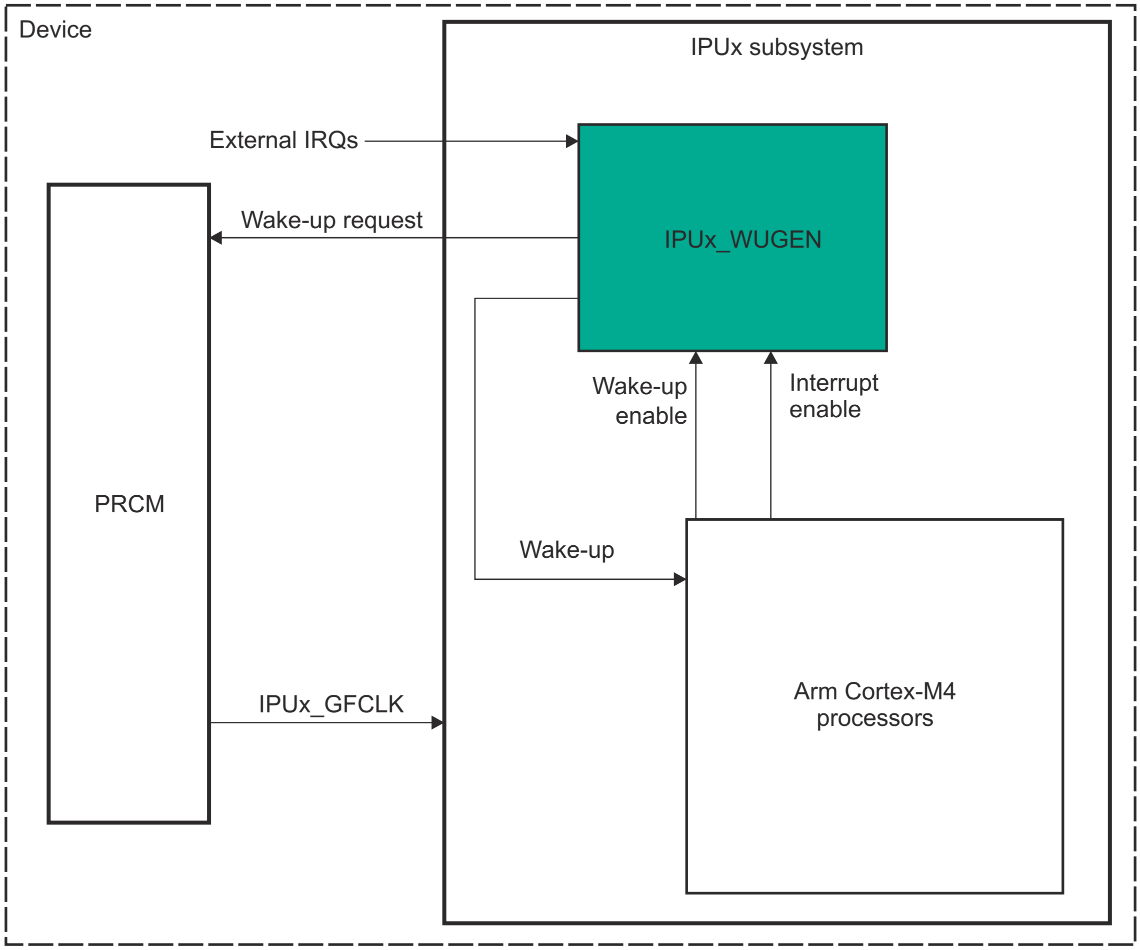


Figure 7-7. IPUx_WUGEN Overview

The wake-up signal to the PRCM module requests the IPUx subsystem functional clock (IPUx_GFCLK). The wake-up signal to the Arm Cortex-M4 processors indicates to them that at least one enabled request is active.

7.3.3 IPUx_UNICACHE

Table 7-7 describes the IPUx_UNICACHE configuration in the IPUx subsystem platform.

Table 7-7. IPUx_UNICACHE Configuration

Parameter	Value
Way	4
Size	32 KiB
Bank elements	32 bits
Bank number	16
Slave interface data size	32 bits
Master interface data size	64 bits
Line size	256 bits
MMU lookup	Included
Number of slaves	4
Number of masters	1
Number of fill/prefetch buffers	Four prefetch buffers
Slave types	IPUx_UNICACHE interface

IPUx_UNICACHE allows basic maintenance operations, which are performed through a dedicated interface:

1. Preload
2. Lock
3. Clean
4. Invalidate

Maintenance of the cache is performed between the start and end addresses. This allows for direct control of memory regions. All maintenance operations occur in the background and can generate an interrupt when they complete. Such operations are protected by software semaphore, because only one operation at a time can be performed. The maintenance operations can also be performed using MMU small entries.

7.3.4 IPUx_UNICACHE_MMU

The IPUx_UNICACHE_MMU serves the role of an attribute MMU (AMMU) for the unicache. It provides the multi-access cache with region-based address translation, read/write control, access type control, and multi-level cache maintenance. Table 7-8 describes the IPUx_UNICACHE_MMU configuration in the device.

Table 7-8. IPUx_UNICACHE_MMU Configuration

Parameter	Values
Number of large pages	4 entries
Size of large pages	512 MiB or 32 MiB (configurable)
Number of medium pages	2 entries
Size of medium pages	256 KiB or 128 KiB (configurable)
Number of small pages	10 entries
Size of small pages	16 KiB or 4 KiB (configurable)
Number of patch pages	Not included
Size of line pages	256-bit
Number of comparison interfaces	4
Number of comparator sets	1
Write pipeline data comparison	Disabled
Number of IPUx_UNICACHE maintenance interfaces	3
Size of entry address	32-bit

As can be seen in Table 7-8, IPUx_UNICACHE_MMU supports different page sizes: large, medium, and small. The number of large pages, number of medium pages, etc., is defined at design time. The size of the pages is configurable in the following IPUx_UNICACHE_MMU registers:

- CACHE_MMU_LARGE_POLICY_i[1] SIZE
- CACHE_MMU_MED_POLICY_j[1] SIZE
- CACHE_MMU_SMALL_POLICY_k[1] SIZE

The different MMU page sizes can be used to create smaller policies within a larger region.

The logical source address is configured in:

- CACHE_MMU_LARGE_ADDR_i[31:25] ADDRESS
- CACHE_MMU_MED_ADDR_j[31:17] ADDRESS
- CACHE_MMU_SMALL_ADDR_k[31:12] ADDRESS

The logical source translated address is configured in:

- CACHE_MMU_LARGE_XLTE_i[31:25] ADDRESS
- CACHE_MMU_MED_XLTE_j[31:17] ADDRESS
- CACHE_MMU_SMALL_XLTE_k[31:12] ADDRESS

When the SIZE bit is set to 0, all the bits in the ADDRESS bit field of the corresponding logical source address and logical source translated address can be used.

When the SIZE bit is set to 1, the ADDRESS bit field of the corresponding logical source address and logical source translated address must be programmed only with addresses that can address the size of a second possible page.

7.3.5 IPUx_UNICACHE_SCTM

The Subsystem Counter Timer Module (SCTM) is a generic profile counter and timer module that provides the following functions:

- Counter functions:
  - Input events counting
  - Two counter modes:
    - Event counting
    - Duration counting
  - Counter chaining
- Timer functions:
  - Periodic intervals generation
  - Two timer modes:
    - Run-once mode
    - Restart mode
  - Events and/or interrupt generation

The SCTM has eight counters (two of which have timer functions). There are input events going into the SCTM and interrupt events going to the IPU INTC. For more information about the counter configuration and the SCTM input events, see *IPU Subsystem Performance Monitoring*, in *On-Chip Debug Support*. For more information about the mapping of the SCTM interrupt event signals to IPU INTC inputs, see *Interrupt Requests to IPU1_Cx_INTC*, in *Interrupt Controllers*.

Figure 7-8 shows the SCTM block diagram.

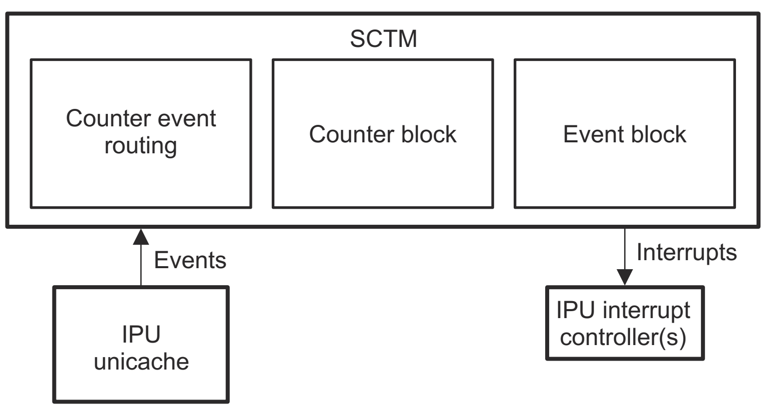


Figure 7-8. SCTM Block Diagram

7.3.5.1 Counter Functions

7.3.5.1.1 Input Events

Signals from within the subsystem are routed to the SCTM and used to control the counters and timers in the module. The routing of the input events from the module boundary to an individual counter is accomplished through an input event multiplexer and is controlled by the CACHE_SCTM_CTCR_WT_i[20:16] INPSEL and CACHE_SCTM_CTCR_WOT_j[20:16] INPSEL bit fields.

Note

For more information about input events to the module boundary, see *Cache Events*, in *On-Chip Debug Support*.

7.3.5.1.2 Counters

There are individual 32-bit counters in the SCTM. The counters count when the input event signals are asserted.

7.3.5.1.2.1 Counting Modes

The counters in the SCTM support two mutually exclusive counting modes:

- Event mode: The counter increments each time a rising edge is detected on the designated input event signal.
- Duration mode: The counter continually increments when the event input is asserted.

7.3.5.1.2.2 Counter Overflow

When the counter reaches the terminal value (0xFFFFFFF) it wraps and continues to increment. This is considered a timer overflow condition. The CACHE_SCTM_CTCR_WT_i[6] OVRFLW and CACHE_SCTM_CTCR_WOT_j[6] OVRFLW bits indicate that overflow has occurred. The overflow bit can be cleared by reading it. When chained, only the high-order counter overflows.

7.3.5.1.2.3 Counters and Processor State

The counters can be configured to alter their behavior based on the state of the CPU. The CACHE_SCTM_CTCR_WT_i[4] FREE and CACHE_SCTM_CTCR_WOT_j[4] FREE bits determine whether the counter will continue to operate when the processor enters debug halt state. When the FREE bit is set to 0, the counter stops incrementing while the debug halt input from the CPU is asserted. Normal operation resumes when the processor exits the debug halt state and the debug halt input is deasserted. When the FREE bit is set to 1, the state of the debug halt input is not used to control counter operation.

The CACHE_SCTM_CTCR_WT_i[5] IDLE and CACHE_SCTM_CTCR_WOT_j[5] IDLE bits determine whether the counter will continue to operate when the processor enters idle state (the processor is no longer executing instructions and is waiting for a wake-up event). When the IDLE bit is set to 0, the counter stops incrementing while the idle input from the CPU is asserted. Normal operation resumes when the processor exits the idle state and the idle input is deasserted. When the IDLE bit is set to 1, the state of the idle input is not used to control counter operation.

7.3.5.1.2.4 Chaining Counters

The individual 32-bit counters in the SCTM can be chained with an adjacent counter to form a 64-bit counter. Counters chained to a counter across an even-odd index boundary with the even counter contain the least-significant 32 bits of the 64-bit pairing. For example, counters 1 and 0 can be paired and counter 1 will contain bits 63:32 and counter 0 will contain bits 31:0. The high-order counter increments by 1 each time the low-order counter wraps.

Counters are chained by setting the CACHE_SCTM_CTCR_WT_i[2] CHAIN or CACHE_SCTM_CTCR_WOT_j[2] CHAIN bit for both counters. When chained, the counter control for both counters is taken from the CACHE_SCTM_CTCR_WT_i or CACHE_SCTM_CTCR_WOT_j register of the low-order counter. Other than the CHAIN bit, all other bits in the high-order CACHE_SCTM_CTCR_WT_i or CACHE_SCTM_CTCR_WOT_j register are ignored.

Chained counters can function only in counter mode. Timer mode is not supported.

The CACHE_SCTM_CTCR_WT_i[7] CHNSDW and CACHE_SCTM_CTCR_WOT_j[7] CHNSDW bits are used to indicate that a counter can provide atomics accessed when chained. These bits are valid only for counters with even indexes (the lower half of a 64-bit counter pair). When these bits are set, the counter can shadow the value of the lower half of the chained counter value at the same time the upper half of the counter is read. The shadowed value (not the current value) is returned when the value of the low-order counter is read.

Therefore, when a chained counter has atomic read capability, an atomic counter value can be obtained simply by reading the high-order counter first, followed by the low-order counter. This order must always be observed to prevent reading stale counter values from the low-order counter. When counters are functioning independently, the shadow feature is deactivated and a read of the counter always returns the current value.

7.3.5.1.2.5 Enabling and Disabling Counters

After the counter is correctly configured, it can be started by setting the CACHE_SCTM_CTCR_WT_i[0] ENBL or CACHE_SCTM_CTCR_WOT_j[0] ENBL bit. At this point, the counter begins incrementing under the control of the configured event input. The counter can be disabled (counting stops) at any time by clearing the ENBL bit. Counters can be enabled and disabled dynamically during application flow.

Counters can also be enabled and disabled as groups through the CACHE_SCTM_CTGNBL register. This register provides control of the individual counter-enable in groups. This allows an application to enable or disable groups of counters in lockstep by setting corresponding bits to 1 (bit number corresponds to counter number).

7.3.5.1.2.6 Resetting Counters

The counters can be reset to their initial value (0x00000000) by writing 1 to the CACHE_SCTM_CTCR_WT_i[1] RESET or CACHE_SCTM_CTCR_WOT_j[1] RESET bit. If the counter is chained, the high-order and low-order counters are reset when the RESET bit is written for the low-order counter.

Counters can also be reset as groups through the CACHE_SCTM_CTGRST registers. These registers provide control of the individual counter reset in groups. This allows an application to reset groups of counters in lockstep.

7.3.5.2 Timer Functions

Counters 0 and 1 in the SCTM can function as timers. When operating as timers, interrupts and/or debug input events are generated when the value of the counter reaches a designated interval.

7.3.5.2.1 Periodic Intervals

The interval for a timer is contained in the CACHE_SCTM_TINTVLR_i register. There is a CACHE_SCTM_TINTVLR_i register for every timer-capable counter in the SCTM. Timers are initialized to 0. When the corresponding CACHE_SCTM_CTCNTR_k increments and matches the values designated in CACHE_SCTM_TINTVLR_i, the timer is considered to be triggered and events configured in CACHE_SCTM_CTCR_WT_i are generated.

Timers can function in one of two mutually exclusive modes:

- Run-once mode: The timer stops after the first interval match and is not re-enabled until the timer is reset to the initial value (0) by setting the CACHE_SCTM_CTCR_WT_i[1] RESET bit to 1.
- Restart mode: The timer automatically resets to the initial value (0) each time the designated interval is reached.

The CACHE_SCTM_CTCR_WT_i[10] RESTART bit is used to configure the timer mode.

7.3.5.2.2 Event Generation

Timers can generate interrupts and debug events. Interrupts are routed from the module boundary to the interrupt controller(s) in the subsystem. Debug events are routed as triggers to debug logic within the subsystem.

The generation of the interrupts is controlled by the CACHE_SCTM_CTCR_WT_i[8] INT bit. The generation of debug events is controlled by the CACHE_SCTM_CTCR_WT_i[9] DBG bit. The INT and DBG bits can be set simultaneously and both signals are generated on interval match.

If neither INT nor DBG is set, the timer function is disabled and the counter functions as a regular counter.

7.3.6 IPUx_MMU

An additional MMU provides address translation for the accesses done from the IPUx subsystem to the L3_MAIN interconnect. The main characteristics of this MMU are:

- 32 entries
- Compatible with Armv6 architecture MMU translation tables (protection bits not used)
- Page-based or access-based endianness conversion
- Two-level descriptor hierarchy
- One intermediate page table
- Four page sizes (16 MiB, 1 MiB, 64 KiB, 4 KiB)
- Page table alignment on 128-byte boundary for Arm11® compatibility

The configuration of the MMU can be done from one of the Cortex-M4 cores or from the L3_MAIN interconnect slave port. The accesses done to configure the MMU cannot be part of a burst access.

For more information about the IPUx_MMU, see Chapter 20, *Memory Management Units*.

7.3.6.1 IPUx_MMU Behavior on Page-Fault in IPUx Subsystem

Table 7-9. IPUx_MMU Behavior on Page-Fault

Application		Debug	
Table-Walker Enabled	**Table-Walker Disabled**	**Table-Walker Enabled**	**Table-Walker Disabled**
Use Table-Walker to find translation. Update TLB cache if successful, set TRANSLATIONFAULT bit and interrupt if not. The following bits are used for the purpose: MMU_IRQENABLE[1] TRANSLATIONFAULT and MMU_IRQSTATUS[1] TRANSLATIONFAULT.	Set TLBMISS bit and interrupt and stall. The following registers are used for the purpose: MMU_IRQENABLE[0] TLBMISS and MMU_IRQSTATUS[0] TLBMISS.	Use Table-Walker to find translation. Update TLB cache if successful (only if the MMU_CNTL[3] EMUTLBUPDATE bit is set), generate in-band bus error if not.	Set EMUMISS bit and interrupt and stall. The following bits are used for the purpose: MMU_IRQENABLE[2] EMUMISS and MMU_IRQSTATUS[2] EMUMISS.

The MMU fault interrupt line is connected to both Cortex-M4 cores (at IPUx_IRQ_16 interrupt line) and is also propagated outside of the IPUx subsystem and connected to an IRQ_CROSSBAR input (IRQ_CROSSBAR_395 for the IPU1 MMU interrupt, IRQ_CROSSBAR_396 for the IPU2 MMU interrupt). The user can route this interrupt to any device host processor by programming properly the corresponding Control Module registers. The host processor (typically, a Cortex-A15) receives the MMU fault and must clean up the fault to resume the execution of the code (or reset IPUx subsystem). It is not possible for one of the Cortex-M4 CPUs to clean up the fault caused by the other Cortex-M4 CPU. This is because both the slave port of the IPUx_MMU (which is stalled) and the configuration port of IPUx_MMU are connected (through a splitter) to the same IPUx_UNICACHE master port.

The default behavior of the IPUx_MMU previously described can be overridden by setting the MMU_GP_REG[0] BUS_ERR_BACK_EN bit to 1. Once this bit is set, all MMU faults (including TLB miss) return a bus error to the IPUx subsystem (interrupt event XLATE_MMU_FAULT). This allows the end user to quickly establish the cause of the MMU fault by having appropriate code in the ISR.

7.3.7 Interprocessor Communication (IPC)

7.3.7.1 Use of WFE and SEV

The IPUx subsystem provides a multiprocessor communication interface for synchronizing tasks. The Arm processors have one output signal, TXEV (transmit event), for sending events and one input signal, RXEV (receive event), for receiving events. Figure 7-9 shows how TXEV and RXEV are connected in the IPUx subsystem.

When a WFE instruction is executed, the processor enters into sleep mode waiting for an event and continues instruction execution when an external event is received. With an SEV (send event) instruction, one processor can wake up the other processor, which is in sleep mode.

The WFE and SEV instructions can help reduce the number of iterations around a lock acquire loop (a spinlock), and thereby reduce power consumption. The basic mechanism involves an observer that is in a spinlock executing a WFE instruction, which suspends execution on that observer until an asynchronous exception or an explicit event (sent by an observer using the SEV instruction) is seen by that observer. The observer that holds the lock uses the SEV instruction to send an event after a lock is released.

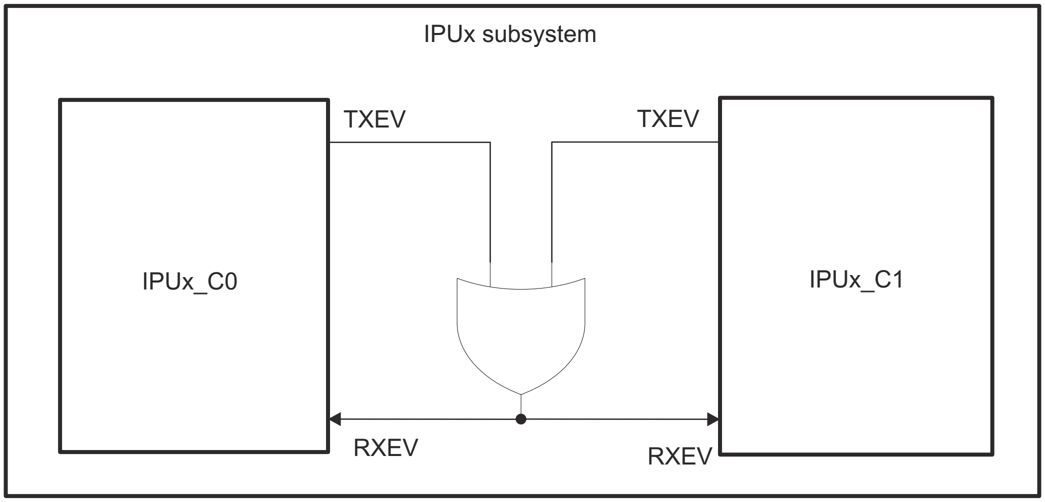


Figure 7-9. Event Communication Connection in IPUx Subsystem

7.3.7.2 Use of Interrupt for IPC

Each Cortex-M4 core can interrupt the other Cortex-M4 core by setting up an interrupt register (CORTEXM4_CTRL_REG). This register is used to trigger the corresponding 'per core' HWSEM_M4_IRQ interrupt (interrupt number 19). Because the priority level for that interrupt can be defined, it is possible to choose the task level at which the interrupt will run. For example, if IPUx_C0 was active and IPUx_C1 was idle (WFI state), when IPUx_C0 completes its task it sets the bit for IPUx_C1 in the control register (CORTEXM4_CTRL_REG[16] INT_CORTEX_2) and goes into sleep mode. IPUx_C1 wakes up seeing this interrupt, and starts running its task. After the completion of its task, IPUx_C1 sets the interrupt for IPUx_C0 (CORTEXM4_CTRL_REG[0] INT_CORTEX_1), and then goes into WFI state. This kind of handshake ensures that if IPUx_C0 and IPUx_C1 are accessing the same resources (memory, registers etc.), only one of the CPUs at a time is active.

7.3.7.3 Use of the Bit-Band Feature for Semaphore Operations

The two Cortex-M4 cores share the same memory system, and it is possible to use the bit-band feature to carry semaphore operations. Because the bit-band alias writes are locked read-modify-write transfers, provided that all tasks changed only the lock bit representing themselves, the lock bits of other tasks are not lost, even if two tasks try to write to the same memory location at the same time.

Each Cortex-M4 core supports two bit-band regions.

Bit-band 1 applies to the virtual address space 0x2000 0000–0x200F FFFF (1 MiB). This virtual address space can be mapped to any physical address and bit-banding will apply to that region. It is recommended that the user map the L2 IPUx_RAM (64 KiB) to this virtual space and use it only for bit-banding operations. If required, the user can define other available small and medium pages over and above the L2 IPUx_RAM virtual space and further extend the use of the bit-band feature.

Bit-band 2 applies to the virtual address space 0x4000 0000–0x400F FFFF (1 MiB). The first 16 KiB of this space (0x4000 0000–0x4000 3FFF) are already reserved for small (one page) pages and cannot be remapped by software. The bit-band alias that corresponds to this 16-KiB region (0x4200 0000–0x4207 FFFF) must also be treated as reserved and no access should be made. The rest of bit-band 2 can be used by appropriately defining the available small and medium pages. In this device, because it is likely that during normal AMMU programming all of L3_MAIN is mapped to this region, it is highly recommended that the user use only bit-band 1 for all purposes and bit-band 2 only if it is necessary.

7.3.7.4 Private Memory Space

Each Arm Cortex-M4 processor has its own memory space, inaccessible by the other processor. In the private memory space are the IPUx_Cx_INTC and RW table registers: CORTEXM4_RW_PID1 and CORTEXM4_RW_PID2. CORTEXM4_RW_PID1 and CORTEXM4_RW_PID2 are accessible only by the respective Cortex-M4 cores (CORTEXM4_RW_PID1 is accessible only by IPUx_C0, while CORTEXM4_RW_PID2 is accessible only by IPUx_C1) . These registers are not accessible from the Cortex-A15 MPU. Because they are not shared, they do not require the bit-band feature (semaphore) to read and write to them.

7.3.8 IPU Boot Options

The IPU boot location is controlled via two Control Module registers:

- CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR[19:0] CORTEX_M4_MMUADDRTRANSLTR: Used to set the physical translated address for IPU AMMU
- CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR[19:0] CORTEX_M4_MMUADDRLOGICTR: Used to set the logical source address for IPU AMMU

By default, two AMMU pages are enabled:

- Small page-0: Translates the 16KB address range from CORTEX_M4_MMUADDRLOGICTR to (CORTEX_M4_MMUADDRLOGICTR + 0x3FFF). If CORTEX_M4_MMUADDRLOGICTR is set to 0x00000, page-0 will control the boot location. If the boot location needs to be mapped to the L2 RAM (0x5502_0000), then CORTEX_M4_MMUADDRTRANSLTR needs to be set to 0x55020. This page is set as non-cacheable at reset.
- Small page-1: Loaded with the physical address of the IPU AMMU configuration registers (0x5508_0000 - 0x5508_0FFF), which is mapped to the virtual address range from 0x4000_0000 to 0x4000_0FFF. This page is also set as non-cacheable at reset.

Note

Small page-1 is by default 4KB. Software has to modify it to 16KB to cover L2MMU/WUGEN masks.

For IPU to boot from any location in L3:

1. Provide boot address through CORTEX_M4_MMUADDRTRANSLTR to AMMU page-0 (CORTEX_M4_MMUADDRLOGICTR set to 0x00000). Keep L2 MMU disabled (or enable L2 MMU but keep the same translation for 0x0; otherwise there will be L2-MMU page-walks / page-faults).
2. Set CORTEX_M4_MMUADDRTRANSLTR to 0x00000 (or any value). Host CPU re-programs AMMU page to map 0x0 virtual address to a physical L2 RAM / L3 location. Only after the programming is complete, Cortex-M4 reset is released. L2 MMU as described above.
3. Set CORTEX_M4_MMUADDRTRANSLTR to 0x00000 (no translation). Host CPU programs L2 MMU to do the address translation for 0x0.

For IPU to boot from L2 RAM:

1. This must be done through AMMU page-0. Use either (1) or (2) as described above.

7.4 Dual Cortex-M4 IPU Subsystem Register Manual

7.4.1 IPUx Subsystem Instance Summary

Table 7-10 and Table 7-11 summarize the IPU1 and IPU2 subsystem instances, respectively.

Table 7-10. IPU1 Subsystem Instance Summary

Module Name	Base Address (IPU Private Access)	Base Address (L3_MAIN Interconnect)	Size
IPU1_UNICACHE_CFG	0x5508 0000	N/A	256 B
IPU1_UNICACHE_SCTM	0x5508 0400	N/A	1 KiB
IPU1_UNICACHE_MMU (AMMU)	0x5508 0800	0x5888 0800	2 KiB
IPU1_WUGEN	0x5508 1000	N/A	4 KiB
IPU1_MMU	0x5508 2000	0x5888 2000	4 KiB
IPU1_C0_INTC[(1)]	0xE000 E000	N/A	4 KiB
IPU1_C1_INTC[(1)]	0xE000 E000	N/A	4 KiB
IPU1_C0_RW_TABLE[(1)]	0xE00F E000	N/A	4 KiB
IPU1_C1_RW_TABLE[(1)]	0xE00F E000	N/A	4 KiB

(1) Different view from each Cortex-M4

Table 7-11. IPU2 Subsystem Instance Summary

Module Name	Base Address (IPU Private Access)	Base Address (L3_MAIN Interconnect)	Size
IPU2_UNICACHE_CFG	0x5508 0000	N/A	256 B
IPU2_UNICACHE_SCTM	0x5508 0400	N/A	1 KiB
IPU2_UNICACHE_MMU (AMMU)	0x5508 0800	0x5508 0800	2 KiB
IPU2_WUGEN	0x5508 1000	N/A	4 KiB
IPU2_MMU	0x5508 2000	0x5508 2000	4 KiB
IPU2_C0_INTC[(1)]	0xE000 E000	N/A	4 KiB
IPU2_C1_INTC[(1)]	0xE000 E000	N/A	4 KiB
IPU2_C0_RW_TABLE[(1)]	0xE00F E000	N/A	4 KiB
IPU2_C1_RW_TABLE[(1)]	0xE00F E000	N/A	4 KiB

(1) Different view from each Cortex-M4

7.4.2 IPUx_UNICACHE_CFG Registers

7.4.2.1 IPUx_UNICACHE_CFG Register Summary

Table 7-12. IPU1_UNICACHE_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU1 Private Access)
CACHE_CONFIG	RW	32	0x0000 0004	0x5508 0004
CACHE_INT	RW	32	0x0000 0008	0x5508 0008
CACHE_OCP	RW	32	0x0000 000C	0x5508 000C
CACHE_MAINT	RW	32	0x0000 0010	0x5508 0010
CACHE_MTSTART	RW	32	0x0000 0014	0x5508 0014
CACHE_MTEND	RW	32	0x0000 0018	0x5508 0018
CACHE_CTADDR	RW	32	0x0000 001C	0x5508 001C

Table 7-12. IPU1_UNICACHE_CFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU1 Private Access)
CACHE_CTDATA	RW	32	0x0000 0020	0x5508 0020

Table 7-13. IPU2_UNICACHE_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU2 Private Access)
CACHE_CONFIG	RW	32	0x0000 0004	0x5508 0004
CACHE_INT	RW	32	0x0000 0008	0x5508 0008
CACHE_OCP	RW	32	0x0000 000C	0x5508 000C
CACHE_MAINT	RW	32	0x0000 0010	0x5508 0010
CACHE_MTSTART	RW	32	0x0000 0014	0x5508 0014
CACHE_MTEND	RW	32	0x0000 0018	0x5508 0018
CACHE_CTADDR	RW	32	0x0000 001C	0x5508 001C
CACHE_CTDATA	RW	32	0x0000 0020	0x5508 0020

7.4.2.2 IPUx_UNICACHE_CFG Register Description

Table 7-14. CACHE_CONFIG

Address Offset	0x0000 0004		
Physical Address	0x5508 0004 0x5888 0004 0x5508 0004 0x5508 0004	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Configuration Register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="27">RESERVED</td><td>LOCK_MAIN</td><td>LOCK_PORT</td><td>LOCK_INT</td><td>BYPASS</td><td>CACHE_LOCK</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4	LOCK_MAIN	Lock access to maintenance registers 0x0: Locked 0x1: Not locked	RW	1
3	LOCK_PORT	Lock access to interface registers 0x0: Locked 0x1: Not locked	RW	1
2	LOCK_INT	Lock access to interrupt registers 0x0: Locked 0x1: Not locked	RW	1
1	BYPASS	Bypass cache 0x0: Everything is non-cacheable. 0x1: Everything is cacheable.	RW	0
0	CACHE_LOCK	Unicache lock. Once this bit is set only debugger or hardware reset can clear. 0x0: No effect	RW	0

Bits	Field Name	Description	Type	Reset
		0x1: Only debug accesses allowed		

Table 7-15. Register Call Summary for Register CACHE_CONFIG

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]

Table 7-16. CACHE_INT

Address Offset	0x0000 0008		
Physical Address	0x5508 0008 0x5888 0008 0x5508 0008 0x5508 0008	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Interrupt Register		
Type	RW		

31:9	8:5	4	3	2	1	0
RESERVED	PORT	READ	WRITE	MAINT	PAGEFAULT	CONFIG

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved.	R	0x000000
8:5	PORT	Slave interface number that has recorded an error	RW W1toClr	0x0
4	READ	Interface read response error	RW W1toClr	0
3	WRITE	Interface write response error	RW W1toClr	0
2	MAINT	Maintenance is completed	RW W1toClr	0
1	PAGEFAULT	Unicache MMU page fault	RW W1toClr	0
0	CONFIG	Configuration error	RW W1toClr	0

Table 7-17. Register Call Summary for Register CACHE_INT

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]

Table 7-18. CACHE_OCP

Address Offset	0x0000 000C		
Physical Address	0x5508 000C 0x5888 000C 0x5508 000C 0x5508 000C	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Interface Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLEANBUF	PREFETCH	CACHED	WRALLOCATE	WRBUFFER	WRAP

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved.	R	0x0000000
5	CLEANBUF	Clean write and prefetch buffers in cache 0x0: Do not clean 0x1: Clean	RW	0
4	PREFETCH	Always prefetch data 0x0: Follow MMU policies 0x1: Always prefetch	RW	0
3	CACHED	Follow cacheable sideband signals 0x0: Reads always not allocated, writes write through if cached 0x1: Slave sideband signals determine policy	RW	1
2	WRALLOCATE	Follow write allocate sideband signals 0x0: No writes are allocated independent to sideband 0x1: Follow sideband	RW	0
1	WRBUFFER	Write throughs and write back no allocate are buffered 0x0: Write throughs and write back no allocated are not buffered 0x1: Write throughs and write back no allocated are buffered	RW	0
0	WRAP	OCP wrap mode (critical word first) 0x0: Disabled 0x1: Enabled	RW	0

Table 7-19. Register Call Summary for Register CACHE_OCP

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]

Table 7-20. CACHE_MAINT

Address Offset	0x0000 0010		
Physical Address	0x5508 0010 0x5888 0010 0x5508 0010 0x5508 0010	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Maintenance Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	INTERRUPT	INVALIDATE	CLEAN	UNLOCK	LOCK	PRELOAD

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved.	R	0x0000000
5	INTERRUPT	Generate interrupt when maintenance operation is complete 0x0: Do not generate interrupt 0x1: Generate interrupt Note: This bit is cleared by HW when maintenance is complete.	RW	0
4	INVALIDATE	Invalidate lines in region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Invalidate Note: This bit is cleared by HW when maintenance is complete.	RW	0
3	CLEAN	Evict dirty lines in region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Clean Note: This bit is cleared by HW when maintenance is complete.	RW	0
2	UNLOCK	Unlock region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Unlock Note: This bit is cleared by HW when maintenance is complete.	RW	0
1	LOCK	Lock region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Lock Note: This bit is cleared by HW when maintenance is complete.	RW	0
0	PRELOAD	Preload region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Preload Note: This bit is cleared by HW when maintenance is complete.	RW	0

Table 7-21. Register Call Summary for Register CACHE_MAINT

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]

Table 7-22. CACHE_MTSTART

Address Offset	0x0000 0014		
Physical Address	0x5508 0014 0x5888 0014 0x5508 0014 0x5508 0014	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Maintenance Start Configuration Register		

Table 7-22. CACHE_MTSTART (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	START_ADDR	Start address of maintenance operations, reset to 0x0000 0000 when finished	RW	0x0000 0000

Table 7-23. Register Call Summary for Register CACHE_MTSTART

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]

Table 7-24. CACHE_MTEND

Address Offset	0x0000 0018		
Physical Address	0x5508 0018 0x5888 0018 0x5508 0018 0x5508 0018	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Maintenance End Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
END_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	END_ADDR	End address of maintenance operations, reset to 0x0000 0000 when finished	RW	0x0000 0000

Table 7-25. Register Call Summary for Register CACHE_MTEND

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]

Table 7-26. CACHE_CTADDR

Address Offset	0x0000 001C		
Physical Address	0x5508 001C 0x5888 001C 0x5508 001C 0x5508 001C	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Cache Test Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	ADDRESS	Address of cache visibility when read CACHE_CTDATA register, autoincrements	RW	0x0000 0000

Table 7-27. Register Call Summary for Register CACHE_CTADDR

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]
- IPUx_UNICACHE_CFG Register Description: [2] [3]

Table 7-28. CACHE_CTDATA

Address Offset	0x0000 0020		
Physical Address	0x5508 0020 0x5888 0020 0x5508 0020 0x5508 0020	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Cache Test Data Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Cache data at address of CACHE_CTADDR register, CACHE_CTADDR autoincrements each time CACHE_CTDATA is read	RW	0x0000 0000

Table 7-29. Register Call Summary for Register CACHE_CTDATA

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_UNICACHE_CFG Register Summary: [0] [1]
- IPUx_UNICACHE_CFG Register Description: [2] [3]

7.4.3 IPUx_UNICACHE_SCTM Registers

7.4.3.1 IPUx_UNICACHE_SCTM Register Summary

Table 7-30. IPU1_UNICACHE_SCTM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU1 Private Access)
CACHE_SCTM_CTCNTL	RW	32	0x0000 0000	0x5508 0400
RESERVED	R	32	0x0000 0020	0x5508 0420
RESERVED	R	32	0x0000 0024	0x5508 0424
RESERVED	R	32	0x0000 0028	0x5508 0428
RESERVED	R	32	0x0000 002C	0x5508 042C
CACHE_SCTM_TINTVLR_i (1)	RW	32	0x0000 0040 + (0x4 * i)	0x5508 0440 + (0x4 * i)
CACHE_SCTM_CTDBGNUM	R	32	0x0000 007C	0x5508 047C
CACHE_SCTM_CTGNBL	RW	32	0x0000 00F0	0x5508 04F0
CACHE_SCTM_CTGRST	RW	32	0x0000 00F8	0x5508 04F8
CACHE_SCTM_CTCR_WT_i (1)	RW	32	0x0000 0100 + (0x4 * i)	0x5508 0500 + (0x4 * i)
CACHE_SCTM_CTCR_WOT_j (3)	RW	32	0x0000 0108 + (0x4 * j)	0x5508 0508 + (0x4 * j)
CACHE_SCTM_CTCNTR_k (2)	R	32	0x0000 0180 + (0x4 * k)	0x5508 0580 + (0x4 * k)

(1) i = 0 to 1
(2) k = 0 to 7
(3) j = 0 to 5

Table 7-31. IPU2_UNICACHE_SCTM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU2 Private Access)
CACHE_SCTM_CTCNTL	RW	32	0x0000 0000	0x5508 0400
RESERVED	R	32	0x0000 0020	0x5508 0420
RESERVED	R	32	0x0000 0024	0x5508 0424
RESERVED	R	32	0x0000 0028	0x5508 0428
RESERVED	R	32	0x0000 002C	0x5508 042C
CACHE_SCTM_TINTVLR_i (1)	RW	32	0x0000 0040 + (0x4 * i)	0x5508 0440 + (0x4 * i)
CACHE_SCTM_CTDBGNUM	R	32	0x0000 007C	0x5508 047C
CACHE_SCTM_CTGNBL	RW	32	0x0000 00F0	0x5508 04F0
CACHE_SCTM_CTGRST	RW	32	0x0000 00F8	0x5508 04F8
CACHE_SCTM_CTCR_WT_i (1)	RW	32	0x0000 0100 + (0x4 * i)	0x5508 0500 + (0x4 * i)
CACHE_SCTM_CTCR_WOT_j (3)	RW	32	0x0000 0108 + (0x4 * j)	0x5508 0508 + (0x4 * j)
CACHE_SCTM_CTCNTR_k (2)	R	32	0x0000 0180 + (0x4 * k)	0x5508 0580 + (0x4 * k)

(1) i = 0 to 1
(2) k = 0 to 7
(3) j = 0 to 5

7.4.3.2 IPUx_UNICACHE_SCTM Register Description

Table 7-32. CACHE_SCTM_CTCNTL

Address Offset	0x0000 0000		
Physical Address	0x5508 0400 0x5888 0400 0x5508 0400 0x5508 0400	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description			
Type	RW		

31:26	25:18	17:13	12:7	6:3	2:1	0
NUMSTM	NUMINPT	NUMTIMR	NUMCNTR	REVISION	IDLEMODE	ENBL

Bits	Field Name	Description	Type	Reset
31:26	NUMSTM	Number of timers that can export via STM	R	0x00
25:18	NUMINPT	Number of event input signals	R	0x1F
17:13	NUMTIMR	Number of timers in the module	R	0x02
12:7	NUMCNTR	Number of counters in the module	R	0x08
6:3	REVISION	Revision ID of SCTM	R	0x- TI internal data
2:1	IDLEMODE	Idle mode control 0x0: Force Idle mode 0x1: Ths SCTM will acknoledge the idle request, but never transition to the idle state 0x2: Ths SCTM uses the smart idle protocol. This is the default mode 0x3: Since the SCTM does not support internal wakeup, this mode is identical to smart_idle	RW	0x2

Bits	Field Name	Description	Type	Reset
0	ENBL	SCTM global enable 0x0: This module is disabled. Only the configuration interface is functional. All other logic is reset 0x1: The module is enabled and individual counter/timers can be configured	RW	0

Table 7-33. CACHE_SCTM_TINTVLR_i

Address Offset	0x0000 0040 + (0x4 * i)		
Physical Address	0x5508 0440 + (0x4 * i) 0x5888 0440 + (0x4 * i) 0x5508 0440 + (0x4 * i) 0x5508 0440 + (0x4 * i)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	These registers contain the interval match value for the corresponding timers in the SCTM		
Type	RW		

31–0
INTERVAL

Bits	Field Name	Description	Type	Reset
31:0	INTERVAL	Interval match value for the timers in the SCTM	RW	0x0000 0000

Table 7-34. CACHE_SCTM_CTDBGNUM

Address Offset	0x0000 007C		
Physical Address	0x5508 047C 0x5888 047C 0x5508 047C 0x5508 047C	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Counter Timer Number Debug Event Register		
Type	R		

31–3	2–0
RESERVED	NUMEVT

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved.	R	0x0000 0000
2:0	NUMEVT	Number of input selectors for debug events	R	0x0

Table 7-35. CACHE_SCTM_CTGNBL

Address Offset	0x0000 00F0		
Physical Address	0x5508 04F0 0x5888 04F0 0x5508 04F0 0x5508 04F0	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	These registers provide for simultaneous enable/disable of 32 counters		
Type	RW		

31–8	7–0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved.	R	0x000000
7:0	ENABLE	The counter enable bit field	RW	0x00

Table 7-36. CACHE_SCTM_CTGRST

Address Offset	0x0000 00F8		
Physical Address	0x5508 04F8 0x5888 04F8 0x5508 04F8 0x5508 04F8	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	These registers provide for simultaneous reset of 32 counters		
Type	RW		

31:8	7:0
RESERVED	RESET

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved.	R	0x000000
7:0	RESET	The counter reset bit field	RW	0x00

Table 7-37. CACHE_SCTM_CTCR_WT_i

Address Offset	0x0000 0100 + (0x4 * i)		
Physical Address	0x5508 0500 + (0x4 * i) 0x5888 0500 + (0x4 * i) 0x5508 0500 + (0x4 * i) 0x5508 0500 + (0x4 * i)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WT: with timer)		
Type	RW		

31:21	20:16	15:11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	INPSEL	RESERVED	RESTART	DBG	INT	CHNSDW	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved.	R	0x000
20:16	INPSEL	Counter Timer input selection 0: Constantly asserted input that results in a free-running counter/timer 1-31: Index of event input signal selected	RW	0x00
15:11	RESERVED	Reserved.	R	0x00
10	RESTART	Restart the timer after an interval match 0: The timer stops after the first interval match. It must be manually reset by software before it starts counting again (run-once timer mode). 1: The timer immediately resets to 0 and begins incrementing again based on the current input configuration (restart timer mode).	RW	0
9	DBG	Signal debug logic on interval match 0: No debug event is generated. 1: Upon interval match, generates a debug event on the corresponding debug output event signal	RW	0
8	INT	Generate interrupt on interval match 0: No interrupt is generated.	RW	0

Bits	Field Name	Description	Type	Reset
		1: Upon interval match, generates an interrupt on the corresponding interrupt output event signal		
7	CHNSDW	Counter has a shadow register for chain reads. 0: The read of the corresponding counter register returns the current value. 1: Read of the high-order counter register, simultaneously loads the current value of the 32 LSBs into a shadow register. The read of the counter register that corresponds to this counter returns the value of the shadow register. This is applicable only when the counter is chained.	R	0
6	OVRFLW	Counter has wrapped since it was last read 0: The counter has not wrapped since the last read. 1: The counter has wrapped since the last read.	R	0
5	IDLE	Counter ignores processor IDLE state 0: The counter does not increment during IDLE state. 1: The counter continues to function during IDLE state; applicable if IDLEMODE = 1.	RW	0
4	FREE	Counter ignores processor debug halt state 0: The counter does not increment (decrement) during the debug halt state. 1: The counter continues to function during debug halt state.	RW	0
3	DURMODE	Counter is in duration or occurrence mode 0: The counter operates in event mode. The counter increments by 1 each time a rising edge is seen on the designated input event signal. 1: The counter operates in duration mode. The counter increments every time a clock cycle is seen and the corresponding input event signal is asserted.	RW	0
2	CHAIN	Counter is chained to an adjacent counter 0: The counter is not chained. 1: Reserved	RW	0
1	RESET	Counter reset control 0: No effect 1: The corresponding counter is reset to the initial value and the OVERFLW bit is cleared. It continues to function if it is still enabled.	RW	0
0	ENBL	Counter enable control 0: The counter does not increment. 1: The counter increments as configured.	RW	0

Table 7-38. CACHE_SCTM_CTCR_WOT_j

Address Offset	0x0000 0108 + (0x4 * j)		
Physical Address	0x5508 0508 + (0x4 * j) 0x5888 0508 + (0x4 * j) 0x5508 0508 + (0x4 * j) 0x5508 0508 + (0x4 * j)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WOT: without timer)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	INPSEL	RESERVED	CHNSDW	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved.	R	0x000
20:16	INPSEL	Counter input selection 0: Constant low signal on the output interface 1–31: Index of event input signal selected	RW	0x000
15:8	RESERVED	Reserved.	R	0x00
7	CHNSDW	Counter has a shadow register for chain reads. 0: The read of the corresponding counter register returns the current value. 1: Read of the high-order counter register, simultaneously loads the current value of the 32 LSBs into a shadow register. The read of the counter register that corresponds to this counter returns the value of the shadow register. This is applicable only when the counter is chained.	R	1
6	OVRFLW	Counter has wrapped since it was last read 0: The counter has not wrapped since the last read. 1: The counter has wrapped since the last read.	R	0
5	IDLE	Counter ignores processor IDLE state 0: The counter does not increment during IDLE state. 1: The counter continues to function during IDLE state; applicable if IDLEMODE = 1.	RW	0
4	FREE	Counter ignores processor debug halt state 0: The counter does not increment (decrement) during the debug halt state. 1: The counter continues to function during debug halt state.	RW	0
3	DURMODE	Counter is in duration or occurrence mode 0: The counter operates in event mode. The counter increments by 1 each time a rising edge is seen on the designated input event signal. 1: The counter operates in duration mode. The counter increments every time a clock cycle is seen and the corresponding input event signal is asserted.	RW	0
2	CHAIN	Counter is chained to an adjacent counter 0: The counter is not chained. 1: The counter is chained to its partner.	RW	0
1	RESET	Counter reset control 0: No effect 1: The corresponding counter is reset to the initial value and the OVERFLW bit is cleared. It continues to function if it is still enabled.	RW	0
0	ENBL	Counter enable control 0: The counter does not increment. 1: The counter increments as configured.	RW	0

Table 7-39. CACHE_SCTM_CTCNTR_k

Address Offset	0x0000 0180 + (0x4 * k)		
Physical Address	0x5508 0580 + (0x4 * k) 0x5888 0580 + (0x4 * k) 0x5508 0580 + (0x4 * k) 0x5508 0580 + (0x4 * k)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	These registers contain the value of an individual counter in the module. There will be a CTCNTR for every counter in the module		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Counter value	R	0x0000 0000

7.4.4 IPUx_UNICACHE_MMU (AMMU) Registers

7.4.4.1 IPUx_UNICACHE_MMU (AMMU) Register Summary

Table 7-40. IPU1_UNICACHE_MMU (AMMU) Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU1 Private Access)	Physical Address (L3_MAIN Access)
CACHE_MMU_LARGE_ADDR_i (1)	RW	32	0x0000 0000 + (0x4 * i)	0x5508 0800 + (0x4 * i)	0x5888 0800 + (0x4 * i)
CACHE_MMU_LARGE_XLTE_i (1)	RW	32	0x0000 0020 + (0x4 * i)	0x5508 0820 + (0x4 * i)	0x5888 0820 + (0x4 * i)
CACHE_MMU_LARGE_POLICY_i (1)	RW	32	0x0000 0040 + (0x4 * i)	0x5508 0840 + (0x4 * i)	0x5888 0840 + (0x4 * i)
CACHE_MMU_MED_ADDR_j (2)	RW	32	0x0000 0060 + (0x4 * j)	0x5508 0860 + (0x4 * j)	0x5888 0860 + (0x4 * j)
CACHE_MMU_MED_XLTE_j (2)	RW	32	0x0000 00A0 + (0x4 * j)	0x5508 08A0 + (0x4 * j)	0x5888 08A0 + (0x4 * j)
CACHE_MMU_MED_POLICY_j (2)	RW	32	0x0000 00E0 + (0x4 *j)	0x5508 08E0 + (0x4 * j)	0x5888 08E0 + (0x4 * j)
CACHE_MMU_SMALL_ADDR_k (3)	RW	32	0x0000 0120 + (0x4 * k)	0x5508 0920 + (0x4 * k)	0x5888 0920 + (0x4 * k)
CACHE_MMU_SMALL_XLTE_k (3)	RW	32	0x0000 01A0 + (0x4 * k)	0x5508 09A0 + (0x4 * k)	0x5888 09A0 + (0x4 * k)
CACHE_MMU_SMALL_POLICY_k (3)	RW	32	0x0000 0220 + (0x4 * k)	0x5508 0A20 + (0x4 * k)	0x5888 0A20 + (0x4 * k)
CACHE_MMU_SMALL_MAINT_k (3)	RW	32	0x0000 02A0 + (0x4 * k)	0x5508 0AA0 + (0x4 * k)	0x5888 0AA0 + (0x4 * k)
Reserved	RW	32	0x0000 04A8	0x5508 0CA8	0x5888 0CA8
Reserved	RW	32	0x0000 04AC	0x5508 0CAC	0x5888 0CAC
Reserved	RW	32	0x0000 04B0	0x5508 0CB0	0x5888 0CB0
Reserved	R	32	0x0000 04B4	0x5508 0CB4	0x5888 0CB4
CACHE_MMU_MMUCONFIG	RW	32	0x0000 04B8	0x5508 0CB8	0x5888 0CB8

(1) i = 0 to 3
(2) j = 0 to 1
(3) k = 0 to 9

Table 7-41. IPU2_UNICACHE_MMU (AMMU) Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU2 Private Access)	Physical Address (L3_MAIN Access)
CACHE_MMU_LARGE_ADDR_i [(1)]	RW	32	0x0000 0000 + (0x4 * i)	0x5508 0800 + (0x4 * i)	0x5508 0800 + (0x4 * i)
CACHE_MMU_LARGE_XLTE_i [(1)]	RW	32	0x0000 0020 + (0x4 * i)	0x5508 0820 + (0x4 * i)	0x5508 0820 + (0x4 * i)
CACHE_MMU_LARGE_POLICY_i [(1)]	RW	32	0x0000 0040 + (0x4 * i)	0x5508 0840 + (0x4 * i)	0x5508 0840 + (0x4 * i)
CACHE_MMU_MED_ADDR_j [(2)]	RW	32	0x0000 0060 + (0x4 * j)	0x5508 0860 + (0x4 * j)	0x5508 0860 + (0x4 * j)
CACHE_MMU_MED_XLTE_j [(2)]	RW	32	0x0000 00A0 + (0x4 * j)	0x5508 08A0 + (0x4 * j)	0x5508 08A0 + (0x4 * j)
CACHE_MMU_MED_POLICY_j [(2)]	RW	32	0x0000 00E0 + (0x4 *j)	0x5508 08E0 + (0x4 * j)	0x5508 08E0 + (0x4 * j)
CACHE_MMU_SMALL_ADDR_k [(3)]	RW	32	0x0000 0120 + (0x4 * k)	0x5508 0920 + (0x4 * k)	0x5508 0920 + (0x4 * k)
CACHE_MMU_SMALL_XLTE_k [(3)]	RW	32	0x0000 01A0 + (0x4 * k)	0x5508 09A0 + (0x4 * k)	0x5508 09A0 + (0x4 * k)
CACHE_MMU_SMALL_POLICY_k [(3)]	RW	32	0x0000 0220 + (0x4 * k)	0x5508 0A20 + (0x4 * k)	0x5508 0A20 + (0x4 * k)
CACHE_MMU_SMALL_MAINT_k [(3)]	RW	32	0x0000 02A0 + (0x4 * k)	0x5508 0AA0 + (0x4 * k)	0x5508 0AA0 + (0x4 * k)
Reserved	RW	32	0x0000 04A8	0x5508 0CA8	0x5508 0CA8
Reserved	RW	32	0x0000 04AC	0x5508 0CAC	0x5508 0CAC
Reserved	RW	32	0x0000 04B0	0x5508 0CB0	0x5508 0CB0
Reserved	R	32	0x0000 04B4	0x5508 0CB4	0x5508 0CB4
CACHE_MMU_MMUCONFIG	RW	32	0x0000 04B8	0x5508 0CB8	0x5508 0CB8

(1) i = 0 to 3
(2) j = 0 to 1
(3) k = 0 to 9

7.4.4.2 IPUx_UNICACHE_MMU (AMMU) Register Description

Table 7-42. CACHE_MMU_LARGE_ADDR_i

Address Offset	0x0000 0000 + (0x4 * i)		
Physical Address	0x5508 0800 + (0x4 * i) 0x5888 0800 + (0x4 * i) 0x5508 0800 + (0x4 * i) 0x5508 0800 + (0x4 * i)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Large page address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS							RESERVED																								

Bits	Field Name	Description	Type	Reset
31:25	ADDRESS	Logical source address	RW	0x00
24:0	RESERVED	Reserved.	R	0x0000000

Table 7-43. CACHE_MMU_LARGE_XLTE_i

Address Offset	0x0000 0020 + (0x4 * i)

Table 7-43. CACHE_MMU_LARGE_XLTE_i (continued)

Physical Address	0x5508 0820 + (0x4 * i) 0x5888 0820 + (0x4 * i) 0x5508 0820 + (0x4 * i) 0x5508 0820 + (0x4 * i)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Large page translated address		
Type	RW		

Bits 31:25	Bits 24:1	Bit 0
ADDRESS	RESERVED	IGNORE

Bits	Field Name	Description	Type	Reset
31:25	ADDRESS	Logical source translated address	RW	0x00
24:1	RESERVED	Reserved	R	0x000000
0	IGNORE	Do not use translated address.	RW	0

Table 7-44. CACHE_MMU_LARGE_POLICY_i

Address Offset	0x0000 0040 + (0x4 * i)		
Physical Address	0x5508 0840 + (0x4 * i) 0x5888 0840 + (0x4 * i) 0x5508 0840 + (0x4 * i) 0x5508 0840 + (0x4 * i)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Large page policy		
Type	RW		

Bits 31:20	19	18	17	16	15:7	6	5	4	3:2	1	0
RESERVED	L1_WR_POLICY	L1_ALLOCATE	L1_POSTED	L1_CACHEABLE	RESERVED	PRELOAD	READ	EXECUTE	RESERVED	SIZE	ENABLE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19	L1_WR_POLICY	L1 write policy 0x0: Write through 0x1: Write back	RW	0
18	L1_ALLOCATE	L1 allocate policy 0x0: No writes are allocated 0x1: Follow sideband	RW	0
17	L1_POSTED	L1 posted policy 0x0: Not posted 0x1: Posted	RW	0
16	L1_CACHEABLE	L1 cache policy 0x0: Non-cacheable 0x1: Cacheable	RW	0
15:7	RESERVED	Reserved	R	0x000

Bits	Field Name	Description	Type	Reset
6	PRELOAD	Preload region 0x0: Do not preload 0x1: Preload	RW	0
5	READ	Read only	RW	0
4	EXECUTE	Execute only	RW	0
3:2	RESERVED	Reserved	R	0x0
1	SIZE	Size of page 0x0: 32 MiB 0x1: 512 MiB	RW	0
0	ENABLE	Enable page 0x0: Page not enabled 0x1: Page enabled	RW	0

Table 7-45. CACHE_MMU_MED_ADDR_j

Address Offset	0x0000 0060 + (0x4 * j)		
Physical Address	0x5508 0860 + (0x4 * j) 0x5888 0860 + (0x4 * j) 0x5508 0860 + (0x4 * j) 0x5508 0860 + (0x4 * j)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Medium page address		
Type	RW		

31–17	16–0
ADDRESS	RESERVED

Bits	Field Name	Description	Type	Reset
31:17	ADDRESS	Logical source address	RW	0x0000
16:0	RESERVED	Reserved	R	0x00000

Table 7-46. CACHE_MMU_MED_XLTE_j

Address Offset	0x0000 00A0 + (0x4 * j)		
Physical Address	0x5508 08A0 + (0x4 * j) 0x5888 08A0 + (0x4 * j) 0x5508 08A0 + (0x4 * j) 0x5508 08A0 + (0x4 * j)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Medium page translated address		
Type	RW		

31–17	16–1	0
ADDRESS	RESERVED	IGNORE

Bits	Field Name	Description	Type	Reset
31:17	ADDRESS	Logical source translated address	RW	0x0000
16:1	RESERVED	Reserved.	R	0x0000
0	IGNORE	Do not use translated address.	RW	0

Table 7-47. CACHE_MMU_MED_POLICY_j

Address Offset	0x0000 00E0 + (0x4 * j)

Table 7-47. CACHE_MMU_MED_POLICY_j (continued)

Physical Address	0x5508 08E0 + (0x4 * j) 0x5888 08E0 + (0x4 * j) 0x5508 08E0 + (0x4 * j) 0x5508 08E0 + (0x4 * j)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Medium page policy		
Type	RW		

31:20	19	18	17	16	15:7	6	5	4	3:2	1	0
RESERVED	L1_WR_POLICY	L1_ALLOCATE	L1_POSTED	L1_CACHEABLE	RESERVED	PRELOAD	READ	EXECUTE	RESERVED	SIZE	ENABLE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19	L1_WR_POLICY	L1 write policy 0x0: Write through 0x1: Write back	RW	0
18	L1_ALLOCATE	L1 allocate policy 0x0: No writes are allocated 0x1: Follow sideband	RW	0
17	L1_POSTED	L1 posted policy 0x0: Non-posted 0x1: Posted	RW	0
16	L1_CACHEABLE	L1 cache policy 0x0: Non-cacheable 0x1: Cacheable	RW	0
15:7	RESERVED	Reserved	R	0x000
6	PRELOAD	Preload region 0x0: Do not preload 0x1: Preload	RW	0
5	READ	Read only	RW	0
4	EXECUTE	Execute only	RW	0
3:2	RESERVED	Reserved	R	0x0
1	SIZE	Size of page 0x0: 128 KiB 0x1: 256 KiB	RW	0
0	ENABLE	Enable page 0x0: Page not enabled 0x1: Page enabled	RW	0

Table 7-48. CACHE_MMU_SMALL_ADDR_k

Address Offset	0x0000 0120 + (0x4 * k)		
Physical Address	0x5508 0920 + (0x4 * k) 0x5888 0920 + (0x4 * k) 0x5508 0920 + (0x4 * k) 0x5508 0920 + (0x4 * k)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Small page address		

Table 7-48. CACHE_MMU_SMALL_ADDR_k (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																				RESERVED											

Bits	Field Name	Description	Type	Reset
31:12	ADDRESS	Logical source address	RW	See Table 7-49.
11:0	RESERVED	Reserved.	R	0x000

Table 7-49. Reset Value for CACHE_MMU_SMALL_ADDR_k[31:12] ADDRESS

Instance	Reset Value
CACHE_MMU_SMALL_ADDR_0	Takes the value of CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR[19:0] shifted 12-bit left
CACHE_MMU_SMALL_ADDR_1	0x40000
CACHE_MMU_SMALL_ADDR_[2..9]	0x00000

Table 7-50. CACHE_MMU_SMALL_XLTE_k

Address Offset	0x0000 01A0 + (0x4 * k)		
Physical Address	0x5508 09A0 + (0x4 * k) 0x5888 09A0 + (0x4 * k) 0x5508 09A0 + (0x4 * k) 0x5508 09A0 + (0x4 * k)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Small page translated address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																				RESERVED											IGNORE

Bits	Field Name	Description	Type	Reset
31:12	ADDRESS	Physical translated address	RW	See Table 7-51.
11:1	RESERVED	Reserved	R	0x000
0	IGNORE	Do not use translated address.	RW	0

Table 7-51. Reset Value for CACHE_MMU_SMALL_XLTE_k[31:12] ADDRESS

Instance	Reset Value
CACHE_MMU_SMALL_XLTE_0	Takes the value of CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR[19:0] shifted 12-bit left
CACHE_MMU_SMALL_XLTE_1	0x55080
CACHE_MMU_SMALL_XLTE_[2..9]	0x00000

Table 7-52. CACHE_MMU_SMALL_POLICY_k

Address Offset	0x0000 0220 + (0x4 * k)		
Physical Address	0x5508 0A20 + (0x4 * k) 0x5888 0A20 + (0x4 * k) 0x5508 0A20 + (0x4 * k) 0x5508 0A20 + (0x4 * k)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Small page policy		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	L1_WR_POLICY	L1_ALLOCATE	L1_POSTED	L1_CACHEABLE	RESERVED	COHERENCY	RESERVED	PRELOAD	READ	EXECUTE	RESERVED	SIZE	ENABLE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x00
19	L1_WR_POLICY	L1 write policy 0x0: Write through 0x1: Write back	RW	0
18	L1_ALLOCATE	L1 allocate policy 0x0: No writes are allocated 0x1: Follow sideband	RW	0
17	L1_POSTED	L1 posted policy 0x0: Non-posted 0x1: Posted	RW	0
16	L1_CACHEABLE	L1 cache policy 0x0: Non-cacheable 0x1: Cacheable	RW	0
15:9	RESERVED	Reserved	R	0x00
8	COHERENCY	Coherency	R	0
7	RESERVED	Reserved	R	0
6	PRELOAD	Preload region 0x0: Do not preload 0x1: Preload	RW	0
5	READ	Read only	RW	0
4	EXECUTE	Execute only	RW	0
3:2	RESERVED	Reserved	R	0x0
1	SIZE	Size of page 0x0: 4 KiB 0x1: 16 KiB	RW	0
0	ENABLE	Enable page 0x0: Page not enabled 0x1: Page enabled	RW	0

Table 7-53. CACHE_MMU_SMALL_MAINT_k

Address Offset	0x0000 02A0 + (0x4 * k)		
Physical Address	0x5508 0AA0 + (0x4 * k) 0x5888 0AA0 + (0x4 * k) 0x5508 0AA0 + (0x4 * k) 0x5508 0AA0 + (0x4 * k)	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Small page maintenance configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	INTERRUPT	INVALIDATE	CLEAN	LOCK	PRELOAD

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved.	R	0x0000000
4	INTERRUPT	Generate interrupt when maintenance operation is complete	RW	0
3	INVALIDATE	Invalidate page	RW	0
2	CLEAN	Evict page	RW	0
1	LOCK	Lock page	RW	0
0	PRELOAD	Preload page	RW	0

Table 7-54. CACHE_MMU_MMUCONFIG

Address Offset	0x0000 04B8		
Physical Address	0x5508 0CB8 0x5888 0CB8 0x5508 0CB8 0x5508 0CB8	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	MMU configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														PRIVILEGE	MMU_LOCK

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved.	R	0x0000 0000
1	PRIVILEGE	Privilege bit. Once this bit is set, only global flush, debugger, or hardware reset can clear. 0x0: CPU can access everything. 0x1: CPU can only access maintenance, and DMA cannot access MMU at all.	RW	0
0	MMU_LOCK	MMU lock. Once this bit is set only a global flush, debugger, or hardware reset can clear. 0x0: CPU can access everything. 0x1: CPU can only access maintenance, and DMA cannot access the MMU.	RW	0

7.4.5 IPUx_MMU Registers

For information about the IPUx_MMU registers and their description, see Chapter 20, *Memory Management Units*.

7.4.6 IPUx_Cx_INTC Registers

For information about the IPUx_Cx_INTC (NVICs) registers and their description, see the *Arm Cortex-M4 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

7.4.7 IPUx_WUGEN Registers

7.4.7.1 IPUx_WUGEN Register Summary

Table 7-55 summarizes the IPU1_WUGEN, and IPU2_WUGEN register mapping, respectively..

Table 7-55. IPU1_WUGEN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU1 Private Access)
CORTEXM4_CTRL_REG	RW	32	0x0000 0000	0x5508 1000
STANDBY_CORE_SYSCONFIG	RW	32	0x0000 0004	0x5508 1004
IDLE_CORE_SYSCONFIG	RW	32	0x0000 0008	0x5508 1008
WUGEN_MEVT0	RW	32	0x0000 000C	0x5508 100C
WUGEN_MEVT1	RW	32	0x0000 0010	0x5508 1010
RESERVED	R	32	0x0000 0014	0x5508 1014

Table 7-56. IPU2_WUGEN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU2 Private Access)
CORTEXM4_CTRL_REG	RW	32	0x0000 0000	0x5508 1000
STANDBY_CORE_SYSCONFIG	RW	32	0x0000 0004	0x5508 1004
IDLE_CORE_SYSCONFIG	RW	32	0x0000 0008	0x5508 1008
WUGEN_MEVT0	RW	32	0x0000 000C	0x5508 100C
WUGEN_MEVT1	RW	32	0x0000 0010	0x5508 1010
RESERVED	R	32	0x0000 0014	0x5508 1014

7.4.7.2 IPUx_WUGEN Register Description

Table 7-57. CORTEXM4_CTRL_REG

Address Offset	0x0000 0000		
Physical Address	0x5508 1000 0x5888 1000 0x5508 1000 0x5508 1000	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	The register is used by one CPU to interrupt the other, thus used as a handshake between the two CPUs 0x0: Interrupt is cleared; 0x1: Interrupt is set.		
Type	RW		

31:17	16	15:1	0
RESERVED	INT_CORTEX_2	RESERVED	INT_CORTEX_1

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	RW	0x0000 0000
16	INT_CORTEX_2	Interrupt to IPUx_C1	RW	0
15:1	RESERVED	Reserved	RW	0x0000 0000
0	INT_CORTEX_1	Interrupt to IPUx_C0	RW	0

Table 7-58. STANDBY_CORE_SYSCONFIG

Address Offset	0x0000 0004

Table 7-58. STANDBY_CORE_SYSCONFIG (continued)

Physical Address	0x5508 1004 0x5888 1004 0x5508 1004 0x5508 1004	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Standby protocol		
Type	RW		

31–2	1–0
RESERVED	STANDBYMODE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	RW	0x0000 0000
1:0	STANDBYMODE	0x0: Force-standby mode 0x1: No-standby mode 0x2: Smart-standby mode 0x3: Smart-standby wake-up mode – normal mode to be used	RW	0x3

Table 7-59. IDLE_CORE_SYSCONFIG

Address Offset	0x0000 0008		
Physical Address	0x5508 1008 0x5888 1008 0x5508 1008 0x5508 1008	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	Idle protocol		
Type	RW		

31–2	1–0
RESERVED	IDLEMODE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	RW	0x0000 0000
1:0	IDLEMODE	0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode 0x3: Smart-idle wake-up mode – normal mode to be used	RW	0x3

Table 7-60. WUGEN_MEVT0

Address Offset	0x0000 000C		
Physical Address	0x5508 100C 0x5888 100C 0x5508 100C 0x5508 100C	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	This register contains the interrupt mask (LSB) wake-up enable bit per interrupt request: 0x0: Interrupt is disabled; 0x1: Interrupt is enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MIRQ31	MIRQ30	MIRQ29	MIRQ28	MIRQ27	MIRQ26	MIRQ25	MIRQ24	MIRQ23	MIRQ22	MIRQ21	MIRQ20	MIRQ19	MIRQ18	MIRQ17	MIRQ16	MIRQ15	MIRQ14	MIRQ13	MIRQ12	MIRQ11	MIRQ10	MIRQ9	MIRQ8	MIRQ7	MIRQ6	MIRQ5	MIRQ4	MIRQ3	MIRQ2	MIRQ1	MIRQ0

Bits	Field Name	Description	Type	Reset
31	MIRQ31	Interrupt Mask bit 31	RW	0
30	MIRQ30	Interrupt Mask bit 30	RW	0
29	MIRQ29	Interrupt Mask bit 29	RW	0
28	MIRQ28	Interrupt Mask bit 28	RW	0
27	MIRQ27	Interrupt Mask bit 27	RW	0
26	MIRQ26	Interrupt Mask bit 26	RW	0
25	MIRQ25	Interrupt Mask bit 25	RW	0
24	MIRQ24	Interrupt Mask bit 24	RW	0
23	MIRQ23	Interrupt Mask bit 23	RW	0
22	MIRQ22	Interrupt Mask bit 22	RW	0
21	MIRQ21	Interrupt Mask bit 21	RW	0
20	MIRQ20	Interrupt Mask bit 20	RW	0
19	MIRQ19	Interrupt Mask bit 19	RW	0
18	MIRQ18	Interrupt Mask bit 18	RW	0
17	MIRQ17	Interrupt Mask bit 17	RW	0
16	MIRQ16	Interrupt Mask bit 16	RW	0
15	MIRQ15	Interrupt Mask bit 15	RW	0
14	MIRQ14	Interrupt Mask bit 14	RW	0
13	MIRQ13	Interrupt Mask bit 13	RW	0
12	MIRQ12	Interrupt Mask bit 12	RW	0
11	MIRQ11	Interrupt Mask bit 11	RW	0
10	MIRQ10	Interrupt Mask bit 10	RW	0
9	MIRQ9	Interrupt Mask bit 9	RW	0
8	MIRQ8	Interrupt Mask bit 8	RW	0
7	MIRQ7	Interrupt Mask bit 7	RW	0
6	MIRQ6	Interrupt Mask bit 6	RW	0
5	MIRQ5	Interrupt Mask bit 5	RW	0
4	MIRQ4	Interrupt Mask bit 4	RW	0
3	MIRQ3	Interrupt Mask bit 3	RW	0
2	MIRQ2	Interrupt Mask bit 2	RW	0
1	MIRQ1	Interrupt Mask bit 1	RW	0
0	MIRQ0	Interrupt Mask bit 0	RW	0

Table 7-61. WUGEN_MEVT1

Address Offset	0x0000 0010		
Physical Address	0x5508 1010 0x5888 1010 0x5508 1010 0x5508 1010	**Instance**	IPU1_WUGEN_IPU IPU1_WUGEN_MAIN_L3 IPU2_WUGEN_IPU IPU2_WUGEN_MAIN_L3
Description	This register contains the interrupt mask (MSB) wake-up enable bit per interrupt request: 0x0: Interrupt is disabled; 0x1: Interrupt is enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MIRQ63	MIRQ62	MIRQ61	MIRQ60	MIRQ59	MIRQ58	MIRQ57	MIRQ56	MIRQ55	MIRQ54	MIRQ53	MIRQ52	MIRQ51	MIRQ50	MIRQ49	MIRQ48	MIRQ47	MIRQ46	MIRQ45	MIRQ44	MIRQ43	MIRQ42	MIRQ41	MIRQ40	MIRQ39	MIRQ38	MIRQ37	MIRQ36	MIRQ35	MIRQ34	MIRQ33	MIRQ32

Bits	Field Name	Description	Type	Reset
31	MIRQ63	Interrupt Mask bit 63	RW	0
30	MIRQ62	Interrupt Mask bit 62	RW	0
29	MIRQ61	Interrupt Mask bit 61	RW	0
28	MIRQ60	Interrupt Mask bit 60	RW	0
27	MIRQ59	Interrupt Mask bit 59	RW	0
26	MIRQ58	Interrupt Mask bit 58	RW	0
25	MIRQ57	Interrupt Mask bit 57	RW	0
24	MIRQ56	Interrupt Mask bit 56	RW	0
23	MIRQ55	Interrupt Mask bit 55	RW	0
22	MIRQ54	Interrupt Mask bit 54	RW	0
21	MIRQ53	Interrupt Mask bit 53	RW	0
20	MIRQ52	Interrupt Mask bit 52	RW	0
19	MIRQ51	Interrupt Mask bit 51	RW	0
18	MIRQ50	Interrupt Mask bit 50	RW	0
17	MIRQ49	Interrupt Mask bit 49	RW	0
16	MIRQ48	Interrupt Mask bit 48	RW	0
15	MIRQ47	Interrupt Mask bit 47	RW	0
14	MIRQ46	Interrupt Mask bit 46	RW	0
13	MIRQ45	Interrupt Mask bit 45	RW	0
12	MIRQ44	Interrupt Mask bit 44	RW	0
11	MIRQ43	Interrupt Mask bit 43	RW	0
10	MIRQ42	Interrupt Mask bit 42	RW	0
9	MIRQ41	Interrupt Mask bit 41	RW	0
8	MIRQ40	Interrupt Mask bit 40	RW	0
7	MIRQ39	Interrupt Mask bit 39	RW	0
6	MIRQ38	Interrupt Mask bit 38	RW	0
5	MIRQ37	Interrupt Mask bit 37	RW	0
4	MIRQ36	Interrupt Mask bit 36	RW	0
3	MIRQ35	Interrupt Mask bit 35	RW	0
2	MIRQ34	Interrupt Mask bit 34	RW	0
1	MIRQ33	Interrupt Mask bit 33	RW	0
0	MIRQ32	Interrupt Mask bit 32	RW	0

7.4.8 IPUx_Cx_RW_TABLE Registers

7.4.8.1 IPUx_Cx_RW_TABLE Register Summary

Table 7-62. IPU1_Cx_RW_TABLE Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU1 Private Access)	Physical Address (L3_MAIN Access)
CORTEXM4_RW_PID1	RW	32	0x0000 0000	0xE00F E000	N/A
CORTEXM4_RW_PID2	RW	32	0x0000 0004	0xE00F E004	N/A
RESERVED	R	32	0x0000 0008	0xE00F E008	N/A

Table 7-63. IPU2_Cx_RW_TABLE Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU2 Private Access)	Physical Address (L3_MAIN Access)
CORTEXM4_RW_PID1	RW	32	0x0000 0000	0xE00F E000	N/A
CORTEXM4_RW_PID2	RW	32	0x0000 0004	0xE00F E004	N/A
RESERVED	R	32	0x0000 0008	0xE00F E008	N/A

7.4.8.2 IPUx_Cx_RW_TABLE Register Description

Table 7-64. CORTEXM4_RW_PID1

Address Offset	0x0000 0000		
Physical Address	0xE00F E000 0xE00F E000	**Instance**	IPU1_CX_RW_TABLE_IPU IPU2_CX_RW_TABLE_IPU
Description	Peripheral Identification register– allows the user software to differentiate between the two Arm Cortex-M4 processors (two CPUs). The same piece of code running on the two CPUs can result in different execution (for example, branch to different location) depending on the address stored in the register. The address is stored by the BIOS code. The register cannot be accessed when the BIOS code is running (used).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADD1																															

Bits	Field Name	Description	Type	Reset
31:0	BASEADD1	IPUx_ROM memory address	RW	0x0000 0000

Table 7-65. Register Call Summary for Register CORTEXM4_RW_PID1

Dual Cortex-M4 IPU Subsystem Functional Description

- Private Memory Space: [0] [1] [2]

Dual Cortex-M4 IPU Subsystem Register Manual

- IPUx_Cx_RW_TABLE Register Summary: [3] [4]

Table 7-66. CORTEXM4_RW_PID2

Address Offset	0x0000 0004		
Physical Address	0xE00F E004 0xE00F E004	**Instance**	IPU1_CX_RW_TABLE_IPU IPU2_CX_RW_TABLE_IPU
Description	Peripheral Identification register – allows the user software to differentiate between the two Arm Cortex-M4 processors (two CPUs). The same piece of code running on the two CPUs can result in different execution (for example, branch to different location) depending on the address stored in the register. The address is stored by the BIOS code. The register cannot be accessed when the BIOS code is running (used).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADD2																															

Bits	Field Name	Description	Type	Reset
31:0	BASEADD2	IPUx_ROM memory address	RW	0x0000 0000

Table 7-67. Register Call Summary for Register CORTEXM4_RW_PID2

Dual Cortex-M4 IPU Subsystem Functional Description

- Private Memory Space: [0] [1] [2]

Table 7-67. Register Call Summary for Register CORTEXM4_RW_PID2 (continued)

Dual Cortex-M4 IPU Subsystem Register Manual • IPUx_Cx_RW_TABLE Register Summary: [3] [4]

Chapter 8

Embedded Vision Engine

This chapter describes the Embedded Vision Engine (EVE) for the device.

Note

Devices may include up to two identical instances of EVE subsystem (EVE1 and EVE2). Refer to the device *Data Manual* for the number of EVE instances on a specific device. This chapter describes the two EVE device superset. For devices without EVE subsystem integrated, the information can be ignored.

8.1 Embedded Vision Engine (EVE) Subsystem 1481
8.2 ARP32 CPU and Instruction Set 1625
8.3 VCOP CPU and Instruction Set 1820

8.1 Embedded Vision Engine (EVE) Subsystem

8.1.1 EVE Overview

The Embedded Vision Engine (EVE) module is a programmable imaging and vision processing engine, intended for use in devices that serve customer electronics imaging and vision applications. Its programmability meets late-in-development or post-silicon processing requirements, and lets third parties or customers add differentiating features in imaging and vision products.

The device includes two instantiations of the EVE engine. A single EVE module consists of an ARP32 scalar core, a vector coprocessor (VCOP) vector core, and an Enhanced DMA (EDMA3) controller.

Figure 8-1 is a high-level block diagram of the EVE module.

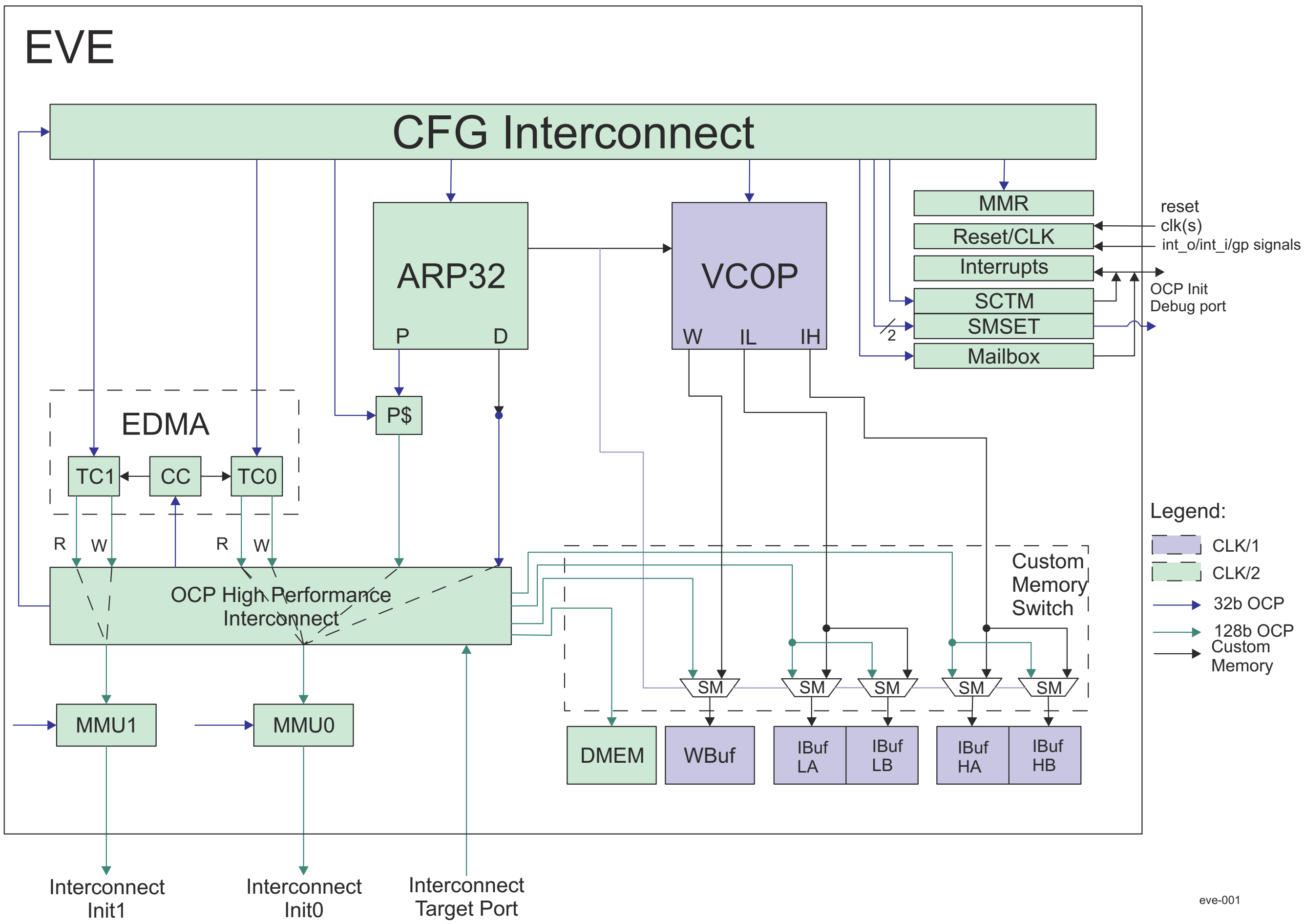


Figure 8-1. EVE Overview

The ARP32 scalar core is the subsystem controller. It coordinates internal EVE interaction, as well as interaction with other components in the system on chip (SoC), like host processor and the DSP. The VCOP is a SIMD engine with built-in loop control and address generation.

The EDMA block is the local DMA. It is used for transferring data between system memories (typically SDRAM and/or L3 SRAM) and internal EVE memories.

Module interconnect is conceptually broken into two parts: A high-performance interconnect and a configuration interconnect. The high-performance interconnect serves as primary high bandwidth (partial) crossbar connection between the EDMA, ARP32, DMA, OCP initiator/target buses and EVE memories. The configuration interconnect provides connectivity to the various memory-mapped registers (MMR) within the EVE module.

The custom memory switch provides a statically muxed low-latency and high-bandwidth connection between the VCOP and the internal EVE memories. The custom memory switch also provides any accesses from the high-performance interconnect and EVE memories.

The interrupt controller (INTC) handles incoming interrupts, merging them with internal interrupt sources to drive the interrupt inputs of the ARP32 .

The MMR block includes control and status bits for general EVE functions that do not reside in other blocks, including reset and clock controller registers and memory static mux ownership.

The SCTM block includes debug logic that supports the counting and timing of various EVE level events, namely VCOP stall counts, VCOP access counts, ARP32 cache miss counts, and so on. The SMSET block provides software message tracing, as well as event tracking.

The reset-clock controller handles subsystem reset and power-management functions.

The EVE engine includes the following main features:

- Two 128-bit interconnect initiator ports used for:
  - Paging between system-level memory (L3 SRAM/DDR) and EVE memory (primarily IBUF, WBUF)
  - ARP32 program fetches to system memory (through program cache)
  - ARP32 load or store requests to system memory
  - ARP32 program cache-related read requests, including prefetch/preload requests
- 128-bit interconnect target port used for system-level host or DMA access to EVE memory or MMR space
- Scalar core (ARP32) with the following features:
  - 32KB program cache (direct mapped and prefetch)
  - 32KB data memory (DMEM)
- Vector core (VCOP):
  - 32KB working buffer (WBUF)
  - 16KB image buffer low copy A (IBUFLA)
  - 16KB image buffer low copy B (IBUFLB)
  - 16KB image buffer high copy A (IBUFHA)
  - 16KB image buffer high copy B (IBUFHB)
- EDMA channel controller (EDMACC): 128 PaRAM entries, 2 Queues
- EDMA transfer controllers: two instances, 2k FIFO each
- Memory Management Units (MMUs):
  - 32-entry TLB per MMU
  - Page walking with hardware
  - EDMA accesses and ARP32 program or data accesses to system memory space
  - Can limit EVE accesses to desired subset of system addresses
- Configuration interconnect for MMR and debug accesses
- High-performance interconnect for high throughput and high concurrency data transfers between connected endpoints
- Multiple interrupts for interrupt mapping, DMA event mapping, and interprocessor handshaking
- Support for Idle and Standby handshake for clock gating
- No support for retention and memory array off modes
- Error detection on all memories:
  - Single bit error detect on DMEM, WBUF, IBUFLA, IBUFLB, IBUFHA, and IBUFHB
  - Double bit error detect on program cache
- Invalid instruction detection in the two processor units (ARP32 and VCOP)
- Debug support
  - Subsystem Counter Timer Module (SCTM) for counting and measuring of VCOP, EVE program cache, and EDMA performance-related state
  - Software Messaging System Event Trace (SMSET) for trace of software messages and hardware events
  - ARP32 debug support: State visibility, breakpoint, run control, cross-triggering
  - VCOP debug support: State visibility and run control

- Interprocessor communication: Internal Mailbox for DSP/EVE communication

8.1.1.1 EVE Memories

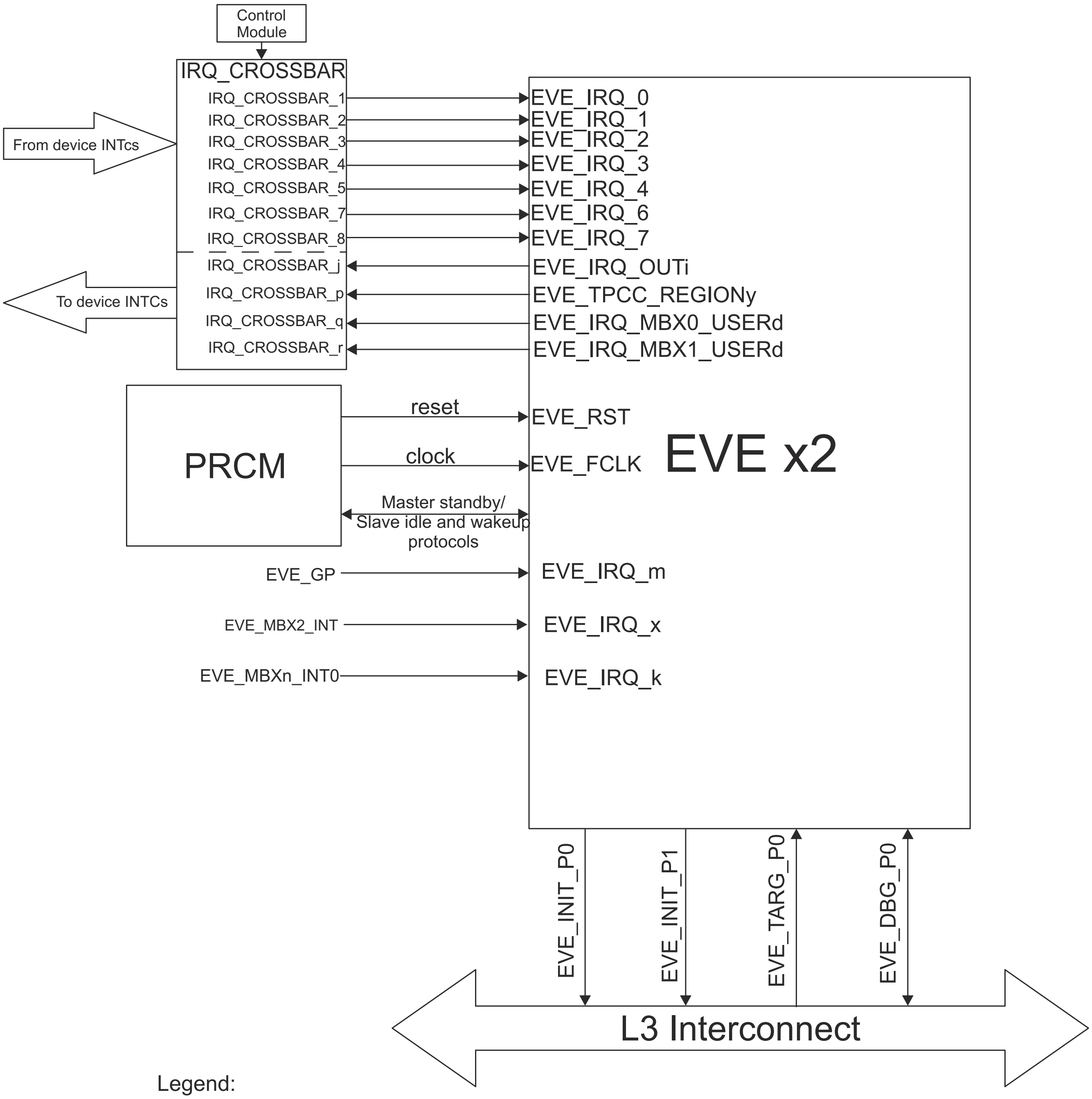
Table 8-1 lists the memories inside the EVE subsystem. See Section 8.1.3.3, *Internal Memory Overview*, for an overview of each memory and its organization, see Section 8.1.3.16, *Memory Map*, for the subsystem memory mapping.

Table 8-1. EVE Subsystem Memory Blocks

Memory	Organization	Size	Function
PMEM	1024 × 256	32KB	ARP32 program cache
DMEM	2048 × 128	32KB	ARP32 data memory
WBUF	8 × 1024 × 32	32KB	VCOP working buffer
IBUFLA	8 × 512 × 32	16KB	Image buffer low copy A
IBUFLB	8 × 512 × 32	16KB	Image buffer low copy B
IBUFHA	8 × 512 × 32	16KB	Image buffer high copy A
IBUFHB	8 × 512 × 32	16KB	Image buffer high copy B

8.1.2 EVE Integration

Figure 8-2 shows the EVE integration inside the device. EVE includes a number of mailboxes, interrupts, DMA events, and general-purpose I/Os (GPIOs) to facilitate handshaking between the EVE modules and also between other initiators or targets such as DSP, MPU, IPU, VIP, VPE and the PRCM module.



Legend:
i = 0, 1, 2, 3
j = 168 to 171 for EVE1
j = 172 to 175 for EVE2
k = 28, 29, 30
m = 8 for EVE1, 9 for EVE2
d = 1 to 3;
y = 1 to 4

x = 16 for EVE1, 17 for EVE2
n = 2, 3, 4
p = 281 to 283 for EVE1
p = 290 to 292 for EVE2
q = 284 to 289 for EVE1; 293 to 298 for EVE2

Figure 8-2. EVE Integration

Note

Figure 8-2 is an overview of the EVE integration; for detailed explanation of the interrupt, mailbox, and general-purpose signals coming out or going in each EVE subsystem, see Figure 8-3 and Figure 8-4.

Table 8-2 through Table 8-4 summarize the integration of the EVE subsystem in the device.

Table 8-2. EVE Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
EVE1	PD_EVE1	L3_MAIN
EVE2	PD_EVE2	L3_MAIN

Table 8-3. EVE Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
EVE1	EVE1_FCLK	EVE_GFCLK	PRCM	Embedded vision engine 1 functional clock, gated. For more information about clock gating and management, refer to *Power, Reset, and Clock Management (PRCM)*.
EVE2	EVE2_FCLK	EVE_GFCLK	PRCM	EVE2 functional clock, gated. For more information about clock gating and management refer to *Power, Reset, and Clock Management (PRCM)*.
Resets				
EVE1	EVE_RST	EVE1_RST EVE1_CPU_RST EVE1_PWRON_RST	PRCM	For information about PRCM reset sources and distribution, see *Reset Management Functional Description*, in *Power, Reset, and Clock Management (PRCM)*.
EVE2	EVE_RST	EVE2_RST EVE2_CPU_RST EVE2_PWRON_RST	PRCM	For information about PRCM reset sources and distribution, see *Reset Management Functional Description*, in *Power, Reset, and Clock Management (PRCM)*.

Table 8-4. EVE Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
EVE1	ELM_IRQ	IRQ_CROSSBAR_1	EVE1_IRQ_0	Error location process completion
	EXT_SYS_IRQ_1	IRQ_CROSSBAR_2	EVE1_IRQ_1	External interrupt (active low)
	CTRL_MODULE_CORE_IRQ_SEC_EVTS	IRQ_CROSSBAR_3	EVE1_IRQ_2	Firewall out-band error
	L3_MAIN_IRQ_DBG_ERR	IRQ_CROSSBAR_4	EVE1_IRQ_3	Interconnect debug error
	L3_MAIN_IRQ_APP_ERR	IRQ_CROSSBAR_5	EVE1_IRQ_4	Interconnect application or non-attributes errors on L3
	PRM_IRQ_MPU	IRQ_CROSSBAR_6	EVE1_IRQ_5	PRCM module interrupt
	DMA_SYSTEM_IRQ_0	IRQ_CROSSBAR_7	EVE1_IRQ_6	SDMA interrupt to EVE1
	DMA_SYSTEM_IRQ_1	IRQ_CROSSBAR_8	EVE1_IRQ_7	SDMA interrupt to EVE1
	EVE2_GP8	N/A	EVE1_IRQ_9	EVE general-purpose interrupt
	EVE2_MBX2_INT1	N/A	EVE1_IRQ_17	EVE mailbox interrupt
	MAILBOX2_INTERRUPT0	N/A	EVE1_IRQ_28	Mailbox 2 interrupt 0
	RESERVED	N/A	EVE1_IRQ_29	Reserved

Table 8-4. EVE Hardware Requests (continued)

Interrupt Requests				
	RESERVED	N/A	EVE1_IRQ_30	Reserved
	EVE1_IRQ_OUT0	IRQ_CROSSBAR_168	N/A	EVE interrupt output
	EVE1_IRQ_OUT1	IRQ_CROSSBAR_169	N/A	EVE interrupt output
	EVE1_IRQ_OUT2	IRQ_CROSSBAR_170	N/A	EVE interrupt output
	EVE1_IRQ_OUT3	IRQ_CROSSBAR_171	N/A	EVE interrupt output
	EVE1_IRQ_TPCC_REGION1	IRQ_CROSSBAR_281	N/A	EVE TPCC region interrupt1
	EVE1_IRQ_TPCC_REGION2	IRQ_CROSSBAR_282	N/A	EVE TPCC region interrupt2
	EVE1_IRQ_TPCC_REGION3	IRQ_CROSSBAR_283	N/A	EVE TPCC region interrupt3
	EVE1_IRQ_MBX0_USER1	IRQ_CROSSBAR_284	N/A	EVE Mailbox 0 user 1
	EVE1_IRQ_MBX0_USER2	IRQ_CROSSBAR_285	N/A	EVE Mailbox 0 user 2
	EVE1_IRQ_MBX0_USER3	IRQ_CROSSBAR_286	N/A	EVE Mailbox 0 user 3
	EVE1_IRQ_MBX1_USER1	IRQ_CROSSBAR_287	N/A	EVE Mailbox 1 user 1
	EVE1_IRQ_MBX1_USER2	IRQ_CROSSBAR_288	N/A	EVE Mailbox 1 user 2
	EVE1_IRQ_MBX1_USER3	IRQ_CROSSBAR_289	N/A	EVE Mailbox 1 user 3
EVE2	ELM_IRQ	IRQ_CROSSBAR_1	EVE2_IRQ_0	Error location process completion
	EXT_SYS_IRQ_1	IRQ_CROSSBAR_2	EVE2_IRQ_1	External interrupt (active low)
	CTRL_MODULE_CORE_IRQ_SEC_EVTS	IRQ_CROSSBAR_3	EVE2_IRQ_2	Firewall out-band error
	L3_MAIN_IRQ_DBG_ERR	IRQ_CROSSBAR_4	EVE2_IRQ_3	Interconnect debug error
	L3_MAIN_IRQ_APP_ERR	IRQ_CROSSBAR_5	EVE2_IRQ_4	Interconnect application or non-attributes errors on L3
	PRM_IRQ_MPU	IRQ_CROSSBAR_6	EVE2_IRQ_5	PRCM module interrupt
	DMA_SYSTEM_IRQ_0	IRQ_CROSSBAR_7	EVE2_IRQ_6	SDMA interrupt to EVE2
	DMA_SYSTEM_IRQ_1	IRQ_CROSSBAR_8	EVE2_IRQ_7	SDMA interrupt to EVE2
	EVE1_GP1	N/A	EVE2_IRQ_8	EVE general-purpose interrupt
	EVE1_MBX2_INT2	N/A	EVE2_IRQ_16	EVE mailbox interrupt
	MAILBOX2_INTERRUPT0	N/A	EVE2_IRQ_28	Mailbox 2 interrupt 0
	RESERVED	N/A	EVE2_IRQ_29	Reserved
	RESERVED	N/A	EVE2_IRQ_30	Reserved
	EVE2_IRQ_OUT0	IRQ_CROSSBAR_172	N/A	EVE interrupt output
	EVE2_IRQ_OUT1	IRQ_CROSSBAR_173	N/A	EVE interrupt output
	EVE2_IRQ_OUT2	IRQ_CROSSBAR_174	N/A	EVE interrupt output
	EVE2_IRQ_OUT3	IRQ_CROSSBAR_175	N/A	EVE interrupt output
	EVE2_IRQ_TPCC_REGION1	IRQ_CROSSBAR_290	N/A	EVE TPCC region interrupt1
	EVE2_IRQ_TPCC_REGION2	IRQ_CROSSBAR_291	N/A	EVE TPCC region interrupt2
	EVE2_IRQ_TPCC_REGION3	IRQ_CROSSBAR_292	N/A	EVE TPCC region interrupt3

Table 8-4. EVE Hardware Requests (continued)

Interrupt Requests				
	EVE2_IRQ_MBX0_USER 1	IRQ_CROSSBAR_293	N/A	EVE Mailbox 0 user 1
	EVE2_IRQ_MBX0_USER 2	IRQ_CROSSBAR_294	N/A	EVE Mailbox 0 user 2
	EVE2_IRQ_MBX0_USER 3	IRQ_CROSSBAR_295	N/A	EVE Mailbox 0 user 3
	EVE2_IRQ_MBX1_USER 1	IRQ_CROSSBAR_296	N/A	EVE Mailbox 1 user 1
	EVE2_IRQ_MBX1_USER 2	IRQ_CROSSBAR_297	N/A	EVE Mailbox 1 user 2
	EVE2_IRQ_MBX1_USER 3	IRQ_CROSSBAR_298	N/A	EVE Mailbox 1 user 3
No DMA Requests				

Note

The **Default Mapping** column in Table 8-4 shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device INTC through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

8.1.2.1 Multi-EVE Recommended Connections

Figure 8-3 and Figure 8-4 show the interrupts, general purpose signals, and DMA requests for each EVE instance.

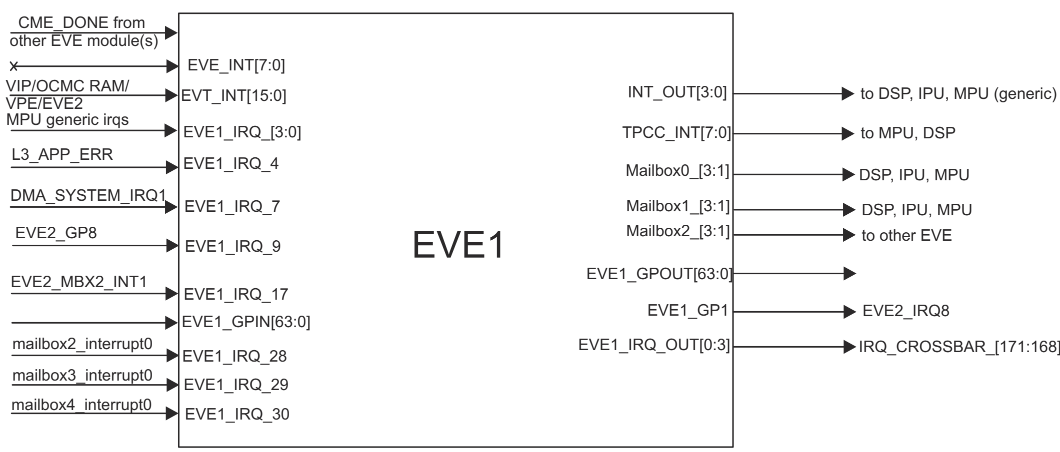


Figure 8-3. EVE1 Signals

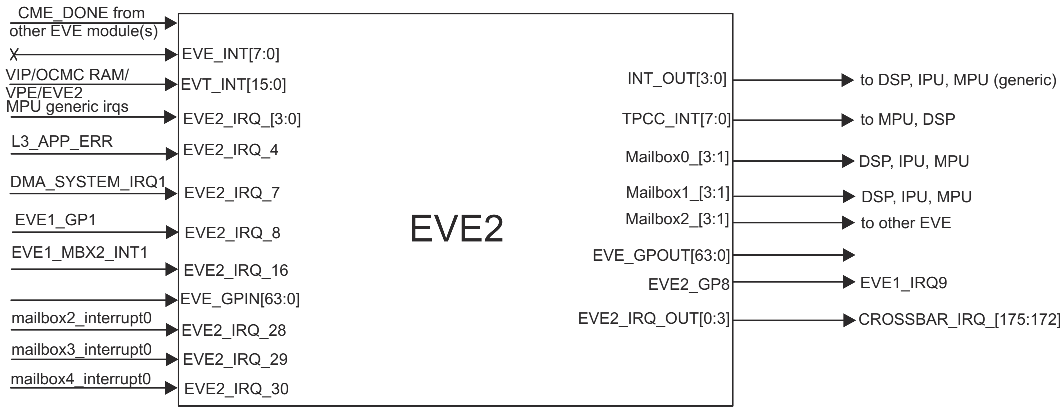


Figure 8-4. EVE2 Signals

EVE INT[7:0] Usage:

EVE_INT[7:0] are not connected to any chip-level or inter-EVE interrupt. These interrupts can be tied to any chip-level mailboxes or any other interrupts that are intended as wake-up interrupts.

TPCC Interrupts Usage:

TPCC interrupts are used as region-based interrupts. The transfers for different processors are divided into separate regions and on completion of a transfer the associated TPCC_INT interrupts the corresponding processor. Region 0 interrupt is reserved for the internal ARP32 processor. Region 1 is routed to the main processor unit (MPU) of the system. Region 2 is routed to DSP1, region 3 to DSP2, and region 4 to IPU.

GPIO Usage:

There are total of 64 general-purpose outputs (GPOs) and 64 general-purpose inputs (GPIs) in each EVE instance. The GPOs of each EVE are connected to the GPIs of all other EVEs, as shown in Figure 8-3 and Figure 8-4. The EVEs communicate their internal processing/task status to the other EVEs with these GPIO signals.

In addition, for each EVE instance, one GPO pin is connected to a DMA event of every EVE. This connection facilitates direct triggering of DMA from the targeted EVE. The EVE that is triggering the DMA transfer to another EVE must be aware of the status of the other EVE in order to safely initiate the DMA transfer. This should be managed in software if desired. For more information about general-purpose signals uses, see Section 8.1.3.6, *General-Purpose Inputs/Outputs*.

8.1.3 EVE Functional Description

8.1.3.1 EVE Connection ID (ConnID) Mapping

The ConnID value is used for error reporting in the memory switch error register (EVE_MSW_ERR[24:16] CONNID). Table 1-5 lists the EVE internal initiators and their assigned 9-bit ConnID values.

Table 8-5. EVE Internal ConnID Mapping

Initiator	CONNID Value
System initiators (through OCP target bus)	0 to FFh
ARP32	100h
VCOP	101h
EDMA TC0	102h
EDMA TC1	103h

8.1.3.2 EVE Processors Overview

8.1.3.2.1 Scalar Core (ARP32)

The scalar core (ARP32) executes conventional RISC instructions. The scalar core is based on the ARP32 processor, which contains the following five functional units:

- The L-unit performs the logical, shift, rotation, extraction, reverse, clear, set, and equal operations.
- The S-unit performs the move operations.
- The D-unit performs arithmetic operations that include compare less than and greater than instructions, address calculation for load and store instructions, PC calculations for branch instructions and stack pointer increment/decrement for PUSH/POP and CALL/RETURN instructions.
- The M-unit performs multiplication.
- The DIV-unit is used for bit serial divide logic.

The ARP32 includes 32-bit program fetch and 32-bit data access bus. For additional information about ARP32 core see Section 8.2, *ARP32 CPU and Instruction Set*.

8.1.3.2.2 VCOP

The VCOP is an SIMD engine with built-in loop control and address generation. The VCOP is programmed in array of 2D block processing level. The vector core has the following resources:

- 4 nested for loops, with loop variables i1, i2, i3, and i4
- 8 address generators, each capable of 4-dimensional addressing; address pattern is *base + i1*const1 + i2*const2 + i3*const3 + i4*const4*
- 16-entry vector register file, each entry is 8-way SIMD × 40-bit signed (sign-extended or zero-padded from 8/16/32-bit signed/unused memory data, or zero-padded from operation upon register data)
- Two general-purpose functional units, each N-way SIMD, N = 2, 4, 8, 16, 32
- Table lookup unit supporting up to N parallel histogram operations
- 8 load units
- 8 store units
- Flat versus Aliased view of EVE memory

In addition, the VCOP supports the following functions:

- Generic compute
- Table lookup
- Histogram and weighted histogram

For additional information about VCOP, see Section 8.3, *VCOP CPU and Instruction Set*.

8.1.3.2.3 Scalar-Vector Interaction

The ARP32 scalar and vector cores share one program memory, but can execute in parallel on separate threads. The scalar core has sole control of the program memory, so all vector instructions are accessed by the scalar core. When a vector instruction is recognized as such it is relayed through the scalar-vector interface to the vector core for execution.

8.1.3.3 Internal Memory Overview

This section provides a functional overview for each of the memory endpoints listed in Table 8-6.

Table 8-6. Internal Memory Blocks

Memory	Organization	Size	Function
PMEM	1024 × 256	32KB	ARP32 program cache
DMEM	2048 × 128	32KB	ARP32 data memory
WBUF	8 × 1024 × 32	32KB	VCOP working buffer
IBUFLA	8 × 512 × 32	16KB	Image buffer low copy A
IBUFLB	8 × 512 × 32	16KB	Image buffer low copy B
IBUFHA	8 × 512 × 32	16KB	Image buffer high copy A
IBUFHB	8 × 512 × 32	16KB	Image buffer high copy B

Note

EVE internal memories, PMEM, DMEM, WBUFs and IBUFs, are visible in the SoC. Device host processors like Cortex-A15, Cortex-M4, and DSP access the EVE memories through its OCP ports. For EVE memory mapping, see Table 8-26. For information about the address space of EVE subsystem modules and memories from device point of view, see *Memory Mapping*.

8.1.3.3.1 Program Cache/Memory

For detailed explanation of the program cache operation, see Section 8.1.3.4, *Program Cache Architecture*.

8.1.3.3.2 ARP32 Data Memory (DMEM)

The ARP32 data memory is logically organized as four banks of 32-bit memories. It is the primary location for storing run-time variables and stack for the software execution environment for the ARP32 core. This memory is byte addressable, and is accessed with byte granularity by all valid initiators. DMEM is accessed at peak rate of up to 4B/cycle by the ARP32 core, or up to 16B/cycle by the system components (EVE EDMA, DMA target bus). The vector core cannot access this memory.

The arbitration between the ARP32 and DMA/System accesses to DMEM is done on round-robin scheme. That is a continuous stream of requests from two (or more) initiator ports results in each requestor gaining access on alternating burst boundaries, therefore DMA/System accesses to DMEM should be minimized in order to avoid impacting the ARP32 performance. Only a single access (ARP32 or DMA/System access) occurs in a single cycle.

8.1.3.3.3 WBUF

VCOP WBUF is the primary location for VCOP look-up tables (LUTs) and other relatively long-lived data buffers. This memory is byte addressable. On a time-slot basis (as controlled by the ARP32 core) WBUF ownership is assigned to either the system (ARP32, EDMA, DMA target bus) or the VCOP.

WBUF is accessible by VCOP, ARP32, and system components (EDMA, DMA target bus) depending on the programmed ownership. The primary master of working buffer is the VCOP.

As shown in Figure 8-5, VCOP can access each of the eight banks of memory independently, giving a total of 256-bits per cycle. ARP32 core can access up to 32-bits per cycle in a given bank and EDMA (and other system components) can access up to 128-bits per cycle of contiguous banks of memory within a 128-bit aligned window (this is a case when there are no independent bank accesses, and single access does not cross 128-bit boundary).

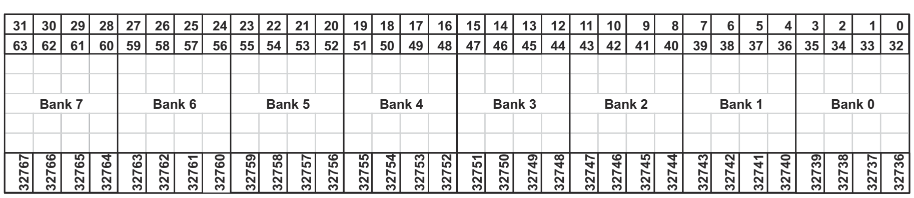


Figure 8-5. WBUF Bank Organization

EDMA versus system accesses (through OCP target bus) are arbitrated dynamically, on OCP bus boundaries, with round-robin policy within the OCP high performance Interconnect. EDMA/System versus ARP32 accesses are arbitrated at dataphase boundaries.

The master that owns the WBUF is pseudostatically defined in the EVE_MSW_CTL[16] WBUF bit. When EVE_MSW_CTL[16] WBUF = 0x0 the static mux (SM) provides a connection from the OCP high-performance interconnect (EDMA, ARP32 or external-to-EVE initiated access) to the WBUF memory, and VCOP accesses are not allowed. Otherwise, (EVE_MSW_CTL[16] WBUF = 0x1) SM provides a connection from VCOP to the WBUF memory and system accesses are not allowed.

In case VCOP or system initiators access the WBUF address location while WBUF is not owned, an error is captured in the EVE memory switch error register (EVE_MSW_ERR) and EVE memory switch error the address register (EVE_MSW_ERRADDR) and the corresponding flag in EVE_MSW_ERR_IRQSTATUS_RAW register is set. Debug accesses do not set the error registers or interrupt, but result in OCP ERR response.

8.1.3.3.4 Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB

EVE image buffers are the primary location for the time-slot based data buffers. These buffers are built to facilitate ping-pong ownership and accesses by the EDMA and the VCOP. On a time-slot bases (as controlled by the ARP32 core), each of the respective image buffers can be assigned ownership to either the System (ARP32, EDMA, OCP target bus) or the VCOP.

IBUFLx and IBUFHx (x = A, B) are typically used as output buffers, and as such VCOP can concurrently access up to eight banks of memory in the VCOP-owned IBUFLA and IBUFLB banks and up to eight banks of memory in one of the VCOP owned IBUFHA and IBUFHB (see Figure 8-6). ARP32 accesses 32-bits per cycle in one of the system-owned IBUF banks.

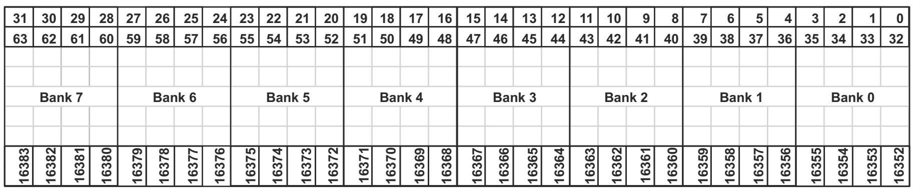


Figure 8-6. IBUF Bank Organization

EDMA (and other EVE system components) access up to 128 bits per cycle of contiguous banks of memory (in case there are no independent bank accesses) within a 128-bit window (a single access does not cross 128-bit boundary) for each system-owned corresponding IBUF bank. For example, if all banks are system owned, EDMA TC0 (or any other EVE system resource) accesses IBUFL at 128-bits per cycle in parallel with EDMA TC1 accessing IBUFH at 128-bits per cycle, giving a total throughput of 256-bits per cycle.

EDMA versus system versus ARP32 accesses are arbitrated dynamically on OCP burst boundaries. The arbitration is done with round-robin policy inside the EVE high-performance interconnect.

The ownership of each IBUF is pseudostatically controlled by the EVE_MSW_CTL [12]IBUFHB, EVE_MSW_CTL [8]IBUFLB, EVE_MSW_CTL [4]IBUFHA, EVE_MSW_CTL [0]IBUFLA bits. Ownership is also internally changed by ARP32 custom instruction or through writes to the memory-mapped address.

By setting EVE_MSW_CTL [12/8/4/0]IBUF = 0x1 the ownership of the IBUF memory is granted to VCOP, otherwise (EVE_MSW_CTL:IBUF = 0x0) the memory is owned by the EVE system resources. If IBUF is system owned, then the SM provides a connection from the EVE high-performance interconnect (for EDMA, ARP32, or external-to-EVE initiated access) to the corresponding IBUF memory, and VCOP accesses are not allowed. If IBUF is VCOP owned, the static mux provides a connection from VCOP to the IBUF memory, and system accesses are not allowed.

Ownership of each of the four IBUF regions (IBUFLA, IBUFLB, IBUFHA, and IBUFHB) is independently programmable with no restrictions. Aliased versus nonaliased mode is controlled by setting the EVE_MEMMAP register. In nonaliased mode it is possible for the VCOP core to own all four IBUFs, or for the system resources to own all four IBUFs or any combination. In aliased mode VCOP owns only one of the IBUFHx regions and one of the IBUFLx regions (x = A, B) because both A and B copies of the IBUFs are mapped to the same address.

In case VCOP or system initiators access the WBUF address location while IBUF is not owned, an error is captured in EVE memory switch error register (EVE_MSW_ERR) and EVE memory switch error address register (EVE_MSW_ERRADDR) and the corresponding flag in the EVE_MSW_ERR_IRQSTATUS_RAW register is set. Debug accesses do not cause the error registers or interrupt to be set, but result in OCP ERR response.

Figure 8-7 shows the typical ownership patterns that offer different amounts of memory/bandwidth to VCOP (V) versus system (S) accesses.

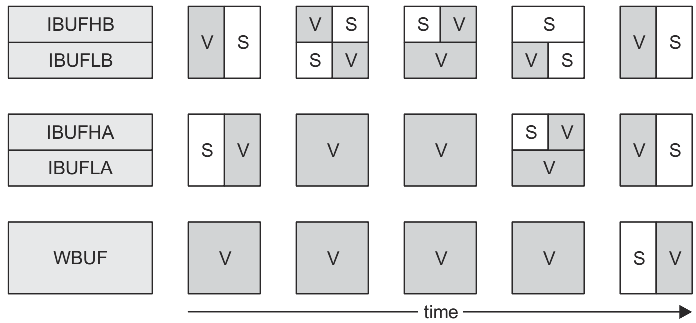


Figure 8-7. VCOP VersusSystem Memory Ownership Examples

8.1.3.3.5 Memory Switch Error Registers

The memory switch error registers capture the first occurrence of an error caused by a functional access. Software must service the error and clear the MMR to capture subsequent errors. The registers capture the identity of the specific requester through the OCP ConnID (see Table 8-5) field and the specific address that is requested.

The MSBs of the address is always 0x400, which matches the internal (local) address view. For system accesses, this does not necessarily match the system view base address. For IBUFL and IBUFH, the physical address offset is captured even in aliased mode (for example, IBUFLB address at 0x4007 0000 is used in both aliased and nonaliased mode).

Unlike the parity/error detect registers, that provide unique error registers per memory type, the memory switch error registers capture errors regardless of which specific memory is accessed.

Debug accesses do not cause the error or interrupt registers to be set, but result in OCP error response.

The memory switch error registers are:

- EVE_MSW_ERR– indicates whether the error is system-, EDMA-, VCOP-, or ARP32-initiated buffer ownership error and captures the corresponding ConnID
- EVE_MSW_ERRADDR– Captures the physical address of memory switch error

Associated error interrupts are mapped on:

- EVE_MSW_ERR_IRQSTATUS_RAW - indicates error interrupt
- EVE_MSW_ERR_IRQSTATUS - indicates error interrupt if event is enabled
- EVE_MSW_ERR_IRQENABLE_SET - enables the corresponding error interrupt
- EVE_MSW_ERR_IRQENABLE_CLR - disables the corresponding error interrupt

8.1.3.3.6 Memory Error Detection

EVE supports parity-based error detection for all internal data memories (including DMEM, WBUF, IBUFLA, IBUFLB, IBUFHA, and IBUFHB) on the minimum access size granularity of 8 bits (that is, there is 1 bit of parity per byte of memory). EVE subsystem also implements double bit error detection for program cache SRAM through a distance 3, 10-bit Hamming code. The 10-bit Hamming code is also applied to the tag and address for a particular cache line.

When parity is enabled, write accesses to any memory cause the parity/encoding bit(s) to be calculated for the specific write data (along with tag and address value for program cache) and written to the corresponding encoding location. For all read accesses from any data memory, the parity bit is read and compared against previously calculated parity for the current pattern in the memory. For program cache, only ARP32 program fetches perform stored versus expected encoding comparison (accesses through program cache OCP debug target port do not result in code calculation). If a mismatch occurs the error details are recorded in parity error MMR (different for each memory - IBUF, WBUF, DMEM) and the associated interrupt is asserted. At the same time, an OCP error response is generated.

The registers that capture program cache-related parity errors, ARP32 DMEM-related parity errors, WBUF-related parity errors, and IBUF-related parity errors are:

- EVE_PMEM_ED_CTL– Enables and disables error PMEM detection logic. Also lets the user invert error detection logic.
- EVE_PMEM_ED_STAT–Status register for the detected PMEM error
- EVE_PMEM_EDADDR– Physical address of parity error
- EVE_DMEM_ED_CTL– Enables/Disables DMEM error detection logic. Also lets the user invert error detection logic.
- EVE_DMEM_ED_STAT– Status register for the detected DMEM error
- EVE_DMEM_EDADDR– Physical address of parity error
- EVE_DMEM_EDADDR_BO– Address byte offset for parity error
- EVE_WBUF_ED_CTL– Enables/Disables WBUF error detection logic. Also lets the user invert error detection logic.
- EVE_WBUF_ED_STAT– Status register for the detected WBUF error
- EVE_WBUF_EDADDR– Physical address of parity error
- EVE_WBUF_EDADDR_BO– Address byte offset for parity error
- EVE_IBUF_ED_CTL– Enables/Disables IBUF error detection logic. Also lets the user invert error detection logic.
- EVE_IBUF_ED_STAT– Status register for the detected IBUF error
- EVE_IBUF_EDADDR– Physical address of parity error
- EVE_IBUF_EDADDR_BO– Address byte offset for parity error

Data returned to the requestor is not modified even in the case of error. The requestor consumes the data potentially causing execution or propagation of bad code and/or data. For a description of the recovery mechanism, see Section 8.1.3.3.6.4, *Parity Error Recovery*.

Each MMR captures the first occurrence of an error. Software then services the error and clears the MMR to capture subsequent errors (generation of OCP error responses continue if additional errors are detected). The parity error MMRs include bit fields to describe which initiator generated the memory access and the specific byte address. Debug accesses do not cause the error or interrupt registers to be set, but result in OCP error response.

Error interrupts are routed to both ARP32 and as EVE outputs to the device level host. ARP32 services interrupts caused by VCOP, EDMA, or OCP target accesses. However, for errors caused by ARP32 accesses, the ARP32 core is unable to service interrupt, and either the DSP or device local CPU detects and services those errors. This detect and service action is described in Section 8.1.3.10.1, *EVE Interrupt Sources – Memory Switch and Parity Error Interrrupts*.

8.1.3.3.6.1 Captured Address – EDADDR and EDADDR_BO

The EDADDR registers (EVE_PMEM_EDADDR, EVE_DMEM_EDADDR, EVE_WBUF_EDADDR, and EVE_IBUF_EDADDR) capture the memory width aligned address for the faulting address. A byte offset bit array for the corresponding memory (EVE_PMEM_EDADDR_BO, EVE_DMEM_EDADDR_BO, EVE_WBUF_EDADDR_BO, and EVE_IBUF_EDADDR_BO) is used to indicate which specific bytes relative to the aligned address is at fault. For most requestors this gives an accurate view of the specific byte for which parity error was detected.

For VCOP, because independent word addresses are generated to the eight banks of IBUF and WBUF, the BANK0 address is latched in the EDADDR register even if that bank is not at fault. In the case of linear addressing, BANK0 + byte_offset mask gives an accurate view. In the case of table-base loads (each bank uses a different address) the reported BANK0 address and the byte offset are used to post-mortem determine which specific address have the error.

The MSBs of the address are always 0x400, which matches the internal (local) address view. For system accesses this does not necessarily match the system view base address. For IBUFs the physical address offset is captured in both aliased and nonaliased mode (for example, IBUFLB address at 0x4007 is used in both aliased and nonaliased modes).

Note

EDADDR_BO does not exist for PMEM memory. PMEM only supports Hamming code-based error detection. PMEM does not support parity per 8-bit detection (unlike the data memories).

8.1.3.3.6.2 Modes of Operation

The EVE error detection includes two basic modes of operation by setting the EN bit in the EVE_<MEMORY>_ED_CTRL (EVE_PMEM_ED_CTL, EVE_DMEM_ED_CTL, EVE_WBUF_ED_CTL and EVE_IBUF_ED_CTL) register, as listed in Table 8-7.

Table 8-7. Error Detection Modes

Mode	Description
EN	Error detection logic is enabled. Writes update parity. Reads check parity.
DIS	Error detection logic is disabled and can be clock gated. Writes do not modify parity. Reads do not check parity.

Note

No parity valid bit exists in addition to a parity state bit. After an existing reset, the ED logic is in disabled state. In this mode the parity RAMs are clock gated to minimize power consumption (though state must be retained). When transitioning to enable state, the underlying SRAM + Parity is in a state that may result in parity mismatch.

For ARP32 program cache, the program cache automatically goes through an Invalidate All sequence; during this time the ARP32 program cache fetch path is stalled. After all lines are invalidated, any subsequent ARP32 request is serviced as cache miss. The newly fetched cacheline is written to PMEM along with valid hamming code bits, and subsequent cache hits check the parity for each instruction fetch. For debug and configuration accesses, the parity and encoding bits are not checked. Debug software uses the tag state to produce an appropriate visual for debug purposes.

For data memory (DMEM, WBUF, and IBUF), it is possible (though not recommended for functional and nontest software) for ARP32, VCOP, or DMA to read a memory location for which the parity bit is not set. After enabling EVE_<DMEM, WBUF, or IBUF>_ED_CTL for the corresponding memory, software must initialize all of the memory, which results in setting a valid parity bit. After this, any parity mismatch results in error signaling, as summarized in the previous sections.

8.1.3.3.6.3 Parity Error Testability

For the software to test their parity error interrupt service routine (ISRs), parity errors can be forced to occur. The EVE subsystem does this by providing an INVERT mode of operation per memory endpoint (by setting the INV bit in the EVE_<PMEM, DMEM, WBUF, or IBUF>_ED_CTL register). When INVERT mode is set (EVE_<MEMORY>_ED_CTL[1] INV = 0x1), and parity is enabled, all read accesses to the corresponding memory return the inverse value of the stored parity bit before the parity check comparison. As a result a parity error is latched and reported through an interrupt, as described in Section 8.1.3.3.5, *Memory Switch Error Registers.* For program memory, the INV bit causes the encoding calculation to return the inverted value for pass or fail.

For testing the ARP32 parity error ISR, these registers can be set asynchronously by the host processor or system EDMA controller. This setting mimics the random nature of radiation-induced soft error.

Parity or Hamming errors can be injected to a specific address after the initialization sequence. Parity or error detection is placed in disable mode (EVE_<MEMORY>_ED_CTL[0] EN = 0x0), and software can write an incorrect value to the desired address. Writes during disable mode do not modify the parity bit or encoding bits, thus the stored parity or encoding bits are considered incorrect. The memory then can be enabled. Any subsequent read access to that memory location results in a parity error mismatch.

8.1.3.3.6.4 Parity Error Recovery

When a parity error occurs, the faulty data is consumed by the requester. VCOP may fetch and process bad data from IBUF or WBUF. ARP32 may try to execute a faulty instruction from the program cache, or read a faulty pointer from DMEM. To avoid these problems, the subsystem parity error interrupt scheme notifies ARP32 or the host of any parity errors (see Section 8.1.3.10.1 , *EVE Interrupt Sources – Memory Switch and Parity Error Interrupts*).

To minimize corruption of the system, and to let the system host query the error detection registers, a user selectable parity error disconnect scheme is provided. This mechanism lets the user disconnect the ARP32 from the neighboring EVE system, and/or disconnect the OCP initiator port from the device interconnect. Each of these disconnects can be manually initiated by setting the EVE_DISC_CONFIG register, or EVE_DISC_CONFIG can be initiated when a specific parity error is detected based on requestor and memory. The disconnect sequence is enabled in the EVE_ED_OCPI_DISC_EN and EVE_ED_ARP32_DISC_EN registers. Figure 8-8 shows an overview of the disconnection request.

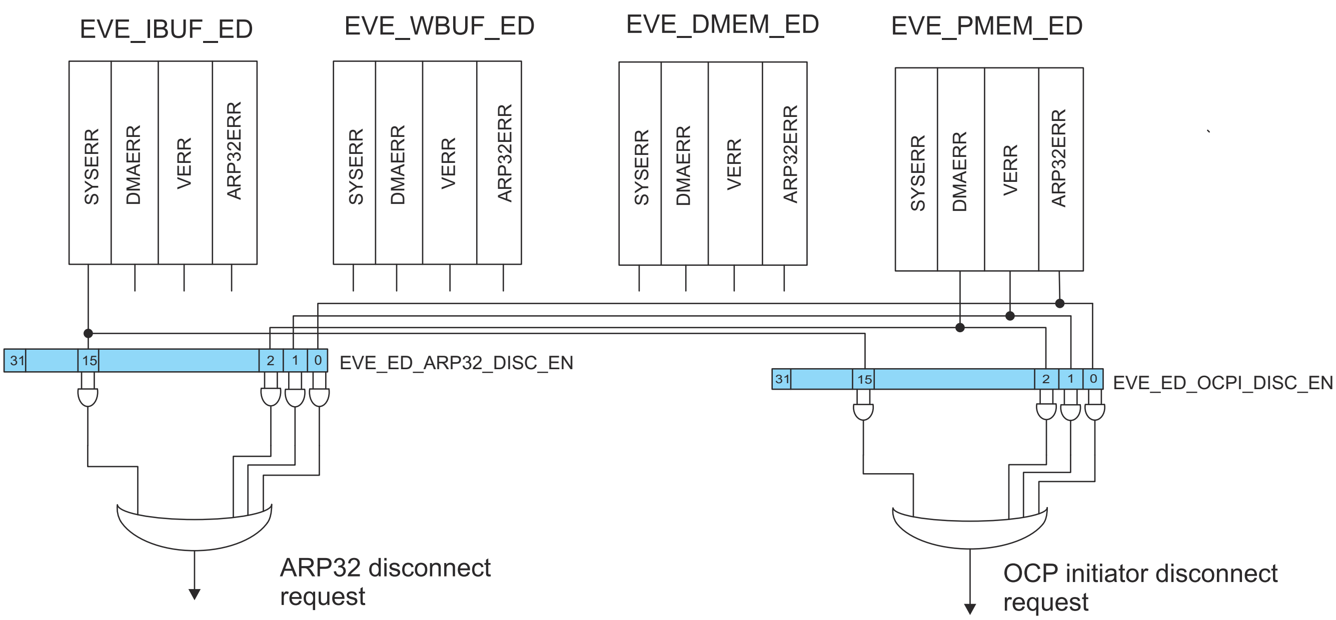


Figure 8-8. Parity Error ARP32/OCP Disconnect

8.1.3.3.7 VCOP System Error Halt Conditions

VCOP has several internal error conditions that force the core to halt execution and freeze its internal state so that ARP32 or debugger can inspect the state of the VCOP. The vector core also provides an error interrupt that causes VCOP to halt based on detection of EVE-level errors. These are mapped to the VCOP memory switch error source status (EVE_MSW_ERR[1] VERR) and VCOP parity error source status (EVE_WBUF_ED_STAT[1] VERR or EVE_IBUF_ED_STAT[1] VERR). By default the halt signals are enabled. Halt signal generation can be modified by writing to the EVE_VCOP_HALT_CONFIG[1] MSW_EN and EVE_VCOP_HALT_CONFIG[0] ED_EN bits.

In addition, the EVE_VCOP_HALT_CONFIG[2] FORCE_ABORT bit is used to directly force the vector core to halt based on any unforseen error condition. The error is detected by ARP32 through a watchdog timer at which point ARP32 sets the FORCE_ABORT signal to halt or abort the current loop occurring in the VCOP core.

8.1.3.4 Program Cache Architecture

The main features of ARP32 program cache are:

- 32-KB direct mapped cache
- 32-B line size
- Hit throughput: 32-bits per cycle, 100% throughput
- Read-only (from ARP32 perspective), all accesses are cacheable
- Read and write by debug OCP port
- Software directed preload
- Demand based prefetch
- Software directed invalidate
  - Global
  - Range based
- Debug support:
  - Single register one-line invalidate (for breakpoint support)
  - Tags memory mapped
  - SRAM memory mapped
  - Cache miss signaling to SCTM
- Hamming code based error detection: 10-bits per 256-bit location + tag + address

Program cache is a direct-mapped cache, therefore software controls the linking of the functions in external memory to minimize conflict misses. The program cache supports error detection encoding and can correct two errors per 256 bits of program code.

8.1.3.4.1 Basic Operation

The program cache is always enabled and treats all ARP32 accesses as cacheable.

For cache hits the interface between ARP32 and program cache can handle back-to-back requests with 0 cycle latency to provide full throughput and program execution for the ARP32. For cache misses, the cache controller stalls ARP32, and internally submits a 32-bit OCP request (through the demand-based prefetch (DBP) block). After the 32-bit cacheline returns, the program cache returns the appropriate data to the ARP32 and starts the CPU.

For CPU-generated miss requests (not software-direct preload), the DBP block snoops the internal request and either services the request directly or schedules a miss+prefetch operation to the system.

Figure 8-9 shows the program cache block diagram.

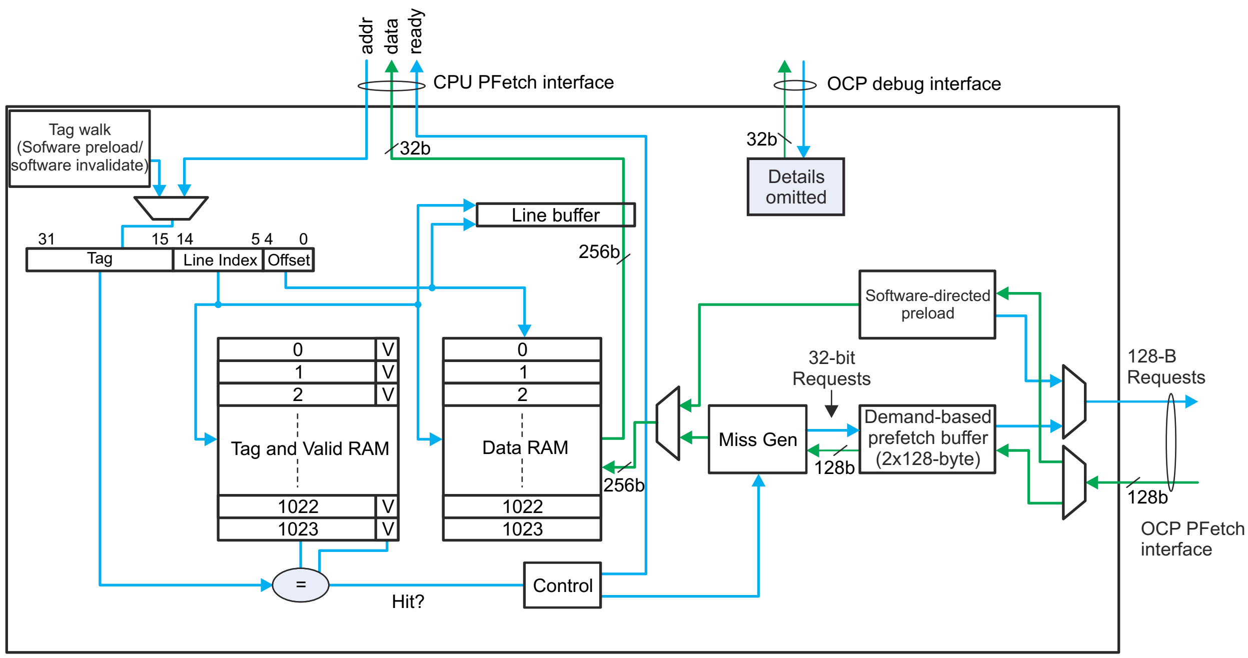


Figure 8-9. EVE Program Cache Architecture

8.1.3.4.2 Line Buffer

The program cache includes a software and performance transparent 256-bit line buffer. The line buffer minimizes accesses to the underlying cache/SRAM in the case of back-to-back hits to the same line; doing so minimizes power consumption. The line buffer is allocated on any program cache access to a new line. As long as subsequent program fetches are to the same line, the data returns to ARP32 directly from the line buffer.

8.1.3.4.3 Software Direct Preload

Software Direct Preload SDP allows preloading of a range of system memory into the program cache. Software sets the preload base address register (EVE_PC_PBAR), along with a preload byte counter (EVE_PC_PBC) register (maximum = 0x8000). Hardware will issue a cache line fill request to the system for the associated line for the programmed address range. When the data is returned for each line, it is written into the cache (32-bits per write, to minimize program stalls) and the tag is marked as valid for the new address. This process is repeated for every cache line in the requested range. When the operation completes, the EVE_PC_PBC register is set to 0.

Note

Software must verify that the previous operation is completed before issuing another SDP operation.

Functional ARP32 requests are given a lower priority relative to software preload requests. The normal operation of the software preload results in many idle cycles while preload requests are in flight through the system. During that time ARP32 CPU accesses that result in cache hits proceed unhindered. If a CPU cache miss occurs while SDP operation is active, the two operations proceed in parallel.

The start address can be any arbitrary byte address, whereas the preload operation occurs on cache lines (32-bit aligned).Thus, the range preloaded is effectively rounded down to the nearest cache-line address relative to the start address, and rounded up to include the entire cache line relative to the end address.

Preload requests operate in parallel with the DBP block and do not use the buffering provided by the DBP block. Instead, the returned data for software direct preload is written directly to the program cache as highest priority, thus minimizing the required buffering and minimizing the impact on the system.

8.1.3.4.4 User Coherence Operation

The memory spaces cached in ARP32 program cache are not always coherent with the external memory. For example, if the copy of the code in external memory is modified, it is possible that the copy of the code that is cached in L1P is out of date with respect to the corresponding location contents in system memory. This may happen, for example, if software is implemented using code overlays. The application controlling such paging should take the appropriate steps to invalidate a cache that is rendered incoherent.

In addition, the ARP32 breakpoint methodology involves emulation and debug software inserting a breakpoint instruction in the program image in system memory (by replacing a valid instruction). Thus, there is a mechanism to update the cache to reflect the newly inserted breakpoint instruction. This is accomplished with the same software or hardware mechanism.

To allow for such software-directed invalidate, the cache controller supports three methods to invalidate the content of the cache:

- Global invalidate
- Range-based invalidate
- Single-address invalidate

8.1.3.4.4.1 Global Invalidate

The program cache clear register must be set (EVE_PC_INV[0] I = 0x1) to invalidate all the lines in the program cache. This is done by resetting all valid bits. Because EVE implements the valid bits with memory, the operation takes approximately a number of cycles equal to the number of tags. During invalidate-all operation, any CPU fetches are stalled until the invalidate-all operation completes, resulting in subsequent misses in the program cache.

8.1.3.4.4.2 Range-Based Invalidate

The program cache controller supports a programmable invalidate mechanism, with which the starting address and the number of words to invalidate can be specified to start an invalidation sequence. The programming model for the invalidation mechanism consist of two memory mapped registers: the start address register (EVE_PC_IBAR[31:0] ADDR) that holds the start address and the byte count register (EVE_PC_IBC[15:0] BC) that holds the number of bytes to be invalidated.

The invalidate operation begins immediately with the writing into EVE_PC_IBC[15:0] BC (maximum = 0x8000). The application first sets the EVE_PC_IBAR[31:0] ADDR bit field and then the byte count register to ensure correct operation. The operation involves cycling through addresses starting from the value of EVE_PC_IBAR[31:0] ADDR in increments of the cache line size, doing tag lookups to check if the line exists in the cache. If the line exists in cache, the corresponding valid bit is reset. The. EVE_PC_IBC[15:0] BC bit field is reset to 0 when the invalidate completes. As the byte count register is readable, a check for 0 provides a synchronization event for the application to execute from the region being invalidated or before issuing another range-based invalidate.

Note

EVE_PC_IBAR[31:0] ADDR can be any arbitrary byte address; whereas the invalidate operation occurs on cache-lines (that is, 32-bit aligned). Thus the range invalidated is effectively rounded down to the nearest cache-line address relative to the start address, and rounded up to include the entire cache line relative to the end address.

As in the global invalidate case, any CPU program fetches issued during the invalidate-range operation are stalled until the invalidation completes.

8.1.3.4.4.3 Single-Address Invalidate – For Breakpoint Operation

Because ARP32 supports breakpoint insertion, a single register version of the range-based invalidate is provided through the invalidate single address register (EVE_PC_ISAR). This is distinct from the previously defined range based invalidate, though the underlying hardware mechanism is the same. It is legal for the range-based invalidate and single-address invalidate to occur at the same time. Both commands must complete, though the order of completion is not explicitly defined.

Emulation software replaces the appropriate instruction in system memory with the breakpoint instruction. To ensure that future ARP32 execution of that line of code is fetched from system memory (as opposed to directly serviced from the cache), emulation software writes the address to EVE_PC_ISAR[31:0] ADDR.

The invalidate operation begins immediately on writing into the EVE_PC_ISAR register. The hardware operation involves checking the tag for the specified address. If the line exists in the cache, the corresponding valid bit is reset. Upon completion, the EVE_PC_ISAR[31:0] ADDR bit field is set to 0, thus informing the emulation software driver that the command is complete.

Operation is undefined if the application of emulation software tries to write to this register before the previous command completes.

8.1.3.4.5 Demand-Based Prefetch

The DBP block is used to improve performance for straight line cache misses, by snooping the program cache miss/software preload requests and issuing larger/pipelined burst requests to the system. The DBP block includes a 256-byte buffer that is logically composed of two 128-byte buffers.

When EVE is reset, the buffers are reset to an idle/empty state and DBP defaults to enabled (EVE_BUS_CONFIG[4] DBP_ENABLE = 0x1). When a new cache-line request is detected, the DBP initiates 128-byte burst to the system for the current 128-byte aligned range, and another 128-byte burst for the next 128-byte aligned range. DBP requests never cross the 128-byte boundary. Depending on the alignment of the new cache request, for initial requests with LSB = 0, 128-byte request is issued; for LSB = 0x20, a 96-byte burst is issued; for LSB = 0x40 a 64-byte burst is issued; and for LSB = 0x60, a 32-byte burst is issued. This last case does not allocate space in the prefetch buffer.

Any subsequent cache miss requests that match addresses for the previously prefetched data are returned directly from the prefetch buffer. When a cache request is received for address with LSB = 0x60 (last 32 bytes of a 128-byte aligned buffer), then that 128-byte buffer segment is marked invalid and the next sequential 128 bytes of data is prefetched. The prefetch buffer is always 256 bytes ahead of the program cache miss requests, and operates in ping-pong fashion with each of the 128-bytes buffer segments.

Example:
1. Buffer 0 holds addresses from 0x0 through 0x7F and buffer 1 holds addresses 0x80 through 0xFF. If the next miss request is to any location in buffer 1 (for example 0xA0), then buffer 0 is invalidated and prefetches range 0x100 through 0x17F.
2. Buffer 0 holds addresses from 0x0 through 0x7F and buffer 1 holds addresses 0x80 through 0xFF. If the next miss request is to the final quadrant of buffer 1 (0xE0), then return data from 0xE0 is returned. Both buffer 1 and buffer 0 are invalidated and addresses 0x100 and 0x180 are prefetched.

When a discontinuous cache-line request is received (that is, the request is not allocated [or in flight] in the prefetch buffer), then the entire prefetch buffer is invalidated, and the sequence begins from the start.

The prefetch buffer is invalidated when any user coherence operation is initiated.

DBP is disabled by clearing the EVE_BUS_CONFIG[4] DBP_ENABLE bit. When disabled, all cache miss requests bypass the DBP buffers and are issued to the system as 32-byte demand cache misses.

8.1.3.4.6 Debug Support

8.1.3.4.6.1 Read/Write Accessibility through OCP Debug Target Port

The OCP debug target port into the program cache provides full read and write of the cache SRAM contents, both for the data as well as tag contents, including parity/ecc and valid bits.

Writability is provided for memory self-test and parity error injection. ARP32 is in halted (for example in IDLE or debug halt) or reset state while write accesses are issued to program cache SRAM or tags. This is not intended as a code-preload feature. As such, if writes are performed to the tag value, 0 must be written to the valid bit before ARP32 can resume operation.

Relative to the EVE level base address, the data RAM resides at address 0x0 through 0x7FFF. The tag + ECC + valid bits reside at word-aligned addresses in the range of 0x8000 through 0x8FFF (base relative).

Table 8-8. EVE Tag Visibility

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="17">TAG</td><td colspan="4">Reserved</td><td colspan="10">HC</td><td>V</td></tr>
</table>

Bit	Name	Description
31:15	TAG	Tag. When valid bit = 1, this bit represents the 17 MSBs of the cached address. Reads return tag; writes have no effect. Read access when LB = 0.
14:11	Reserved	Reserved. Read return 0s.
10:1	HC	Hamming code. Written by hardware based on computed Hamming code value calculated based on tag, address, and program cache data contents for this line. Read and compared by hardware to recalculated value when cache-line hit occurs. Reads return Hamming code; writes have no effect. Read accesses when LB = 0.
0	V	Valid bit. Set to 1 by hardware when tag + data line is valid/allocated. Cleared to 0 by hardware when tag + data line is invalid. Reads return valid bit; writes have no effect. Read access when LB = 0.

8.1.3.4.6.2 Breakpoint Support

Breakpoints are supported through the single-address invalidate operation summarized in Section 8.1.3.4.4.3, *Single-Address Invalidate – For Breakpoint Operation*. The emulation software replaces a functional instruction in system memory with the breakpoint opcode, and then invalidates that address in program cache through the range-based invalidate operation.

8.1.3.4.6.3 Cache Profiling

The program cache provides several output signals to the EVE SCTM block for the purpose of profiling and debugging cache performance issues.

Table 8-9 lists the cache profiling signals. The Type column of the table is either pulse or duration. The pulse signals are driven active one cycle for each occurrence. A sequence of consecutive active cycles represent multiple occurrences. It is possible that the duration signal stays active for multiple cycles on a given occurrence. These are stall signals that determine the total number of cycles in the stalled state. The SCTM Mode column describes the counter mode that is programmable in the SCTM module for a given counter. The SCTM Mode column is either event or duration. Event signals result in incrementing the counter once per low-to-high transition on a given input. As such, the event mode is not overly useful for cache profiling. The duration mode measures the number of cycles during which a given signal is active. This is the mode used for most Cache profiling. In the case of counting pulses, this reports the total number of occurrences of a given state/signal (that is, cache hits). In the case of counting duration, this reports the total time in a given state.

Table 8-9. Cache Profiling Signal List

Name	Type	SCTM Mode	Description
cache_miss_count	Pulse	Duration	Per cache miss request occurrence
cache_hit_count	Pulse	Duration	Per cache hit request occurrence
line_buffer_hit_count	Pulse	Duration	Line buffer hit occurrence
cache_miss_stall	Duration	Duration (or event)	Driven high for the duration of a cache miss. Used by SCTM to measure the time the ARP32 is stalled due to a cache miss.
prefetch_compulsory_count	Pulse	Duration	Pulse on each or every compulsory DBP request to the system (that is, for the first 128-bit request after a buffer flush, or from IDLE state).
prefetch_lookahead_count	Pulse	Duration	Pulse on each or every lookahead DBP request to the system (that is, for the second 128-bit request after a flush).

Table 8-9. Cache Profiling Signal List (continued)

Name	Type	SCTM Mode	Description
prefetch_hit_count	Pulse	Duration	Pulse on each or every cache request that hits an already requested prefetch cache line, whether resident in the prefetch buffer or already requested but still in flight, and thus does not generate an additional prefetch request.
prefetch_line_count	Pulse	Duration	Pulse on every 32-bit cacheline that is requested/allocated into the prefetch buffer (including those that bypass the buffer due to starting at address 0x60). The difference between prefetch_line_count and prefetch_hit_count is equivalent to a prefetch_flush_line_count.
prefetch_discard_stall	Duration	Duration (or event)	Represents the amount of time that previously issued prefetch requests data is in flight (and is discarded) while a new compulsatory prefetch request begins.

8.1.3.4.7 Error Detection

For a description of the error detection mechanism, see Section 8.1.3.3.6, *Memory Error Detection*.

8.1.3.5 EDMA

Note

The device EVE integrated EDMA controller instance (EDMA3_0_CC0) which is functionally identical with the device EDMA controller instances (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1). The only difference is that the EDMA3_0_CC0 instance is located at different physical addresses.

EDMA is a DMA engine for transfers between the system memory (DDR and L3 SRAM) and EVE internal memories. For more detailed description of EDMA controller, see *Enhanced DMA*, in *Enhanced DMA*.

Figure 8-10 is a high-level overview figure of the EDMA controller embedded in the EVE subsystem. The figure shows two major components:

- The Channel Controller (EDMACC or CC), which serves as the user interface of EDMA
- The Transfer Controllers (EDMATC or TC), which serve as the data transfer engine of EDMA

The ARP32 submits requests to the channel controller, which in turn submits transfer requests (TRs) to the appropriate transfer controller. Interrupts are posted in the EDMA CC upon transfer completion (if requested) and signaled to ARP32 core.

The EDMA is configured with two queues (in the CC) and two TC instances. The two TC instances offer maximum performance and concurrency for the case where simultaneous transfers occur between two different system-level endpoints, and two different EVE internal memories. For typical use cases, it is expected that transfers involving EMIF or DDR and L3 SRAM are mapped to two different TCs (TC0 for EMIF-related requests and TC1 for L3 SRAM related requests). For example, EMIF or DDR to IBUFL in parallel with IBUFH to L3 SRAM can be processed with maximum performance.

Table 8-10 shows the specific EDMA configuration for the available resources on EVE.

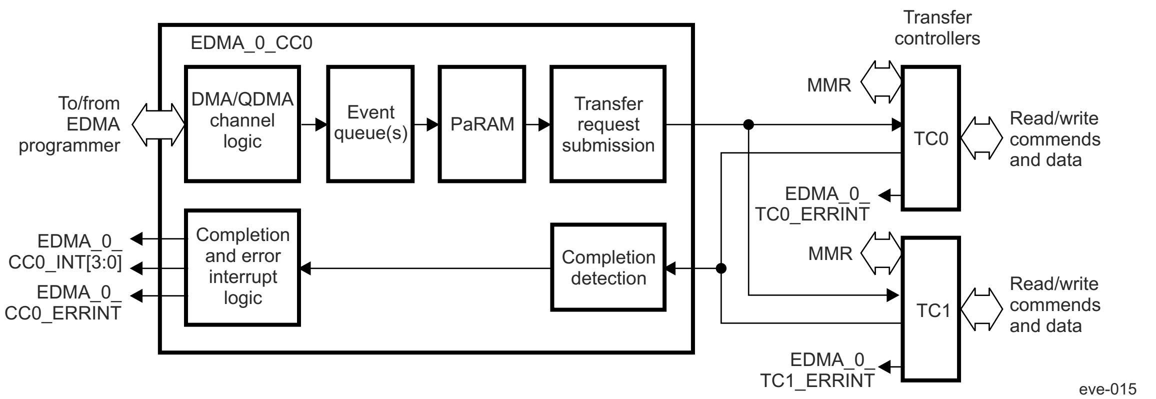


Figure 8-10. EDMA Block Diagram

Table 8-10. EDMA Configuration

Name	Description	Configuration
NUM_DMACH	Number of EDMA channels	16
NUM_QDMACH	Number of QDMA channels	8
NUM_INTCH	Number of interrupt channels	16
NUM_PARAMENTRY	Number of PaRAM entries	128
NUM_EVQUE	Number of event queues	2
NUM_TC	Number of TPTC interfaces	2
MPEXIST	Memory protection exitsts	No
NUM_REGIONS	Number of MP and shadow regions	8
CHMAPEXIST	Channel mapping exists	Yes

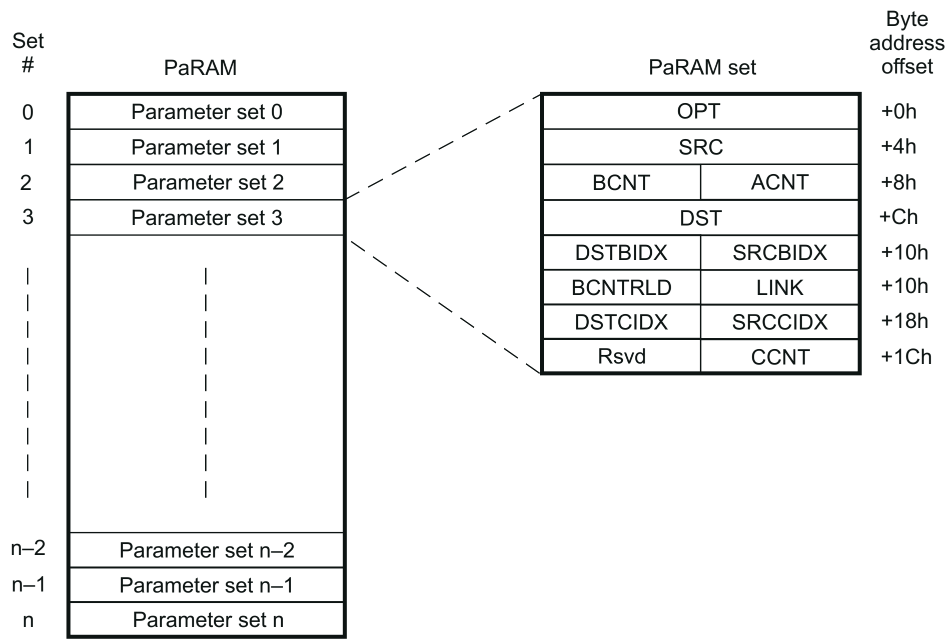
8.1.3.5.1 DMA Channel Events

The EDMA input events/signals are driven by the evt_int[15:0] input signals. Those input interrupts are also connected to INTC1 of the EVE. A subset of these event inputs is mapped to ECM start events. Software must enable individual input interrupts as either events or ARP32 interrupts.

The EDMA channels can also be used under software control. In this case the DMA channels are triggered by writes to the event set register, or through internal chaining feature.

8.1.3.5.2 DMA Parameter Set

Figure 8-11 shows the layout of parameter RAM sets as a sequence of 128 entries which are adjacent in memory, and a detailed look into the contents of each parameter RAM.



Note: *n* is the number of PaRAM sets supported in the EDMACC for a specific device.

Figure 8-11. Structure of Parameter RAM Sets and Contents

Table 8-11. Fields in Parameter RAM

Acronym	Parameter
OPT	Channel Options
SRC	Channel Source Address
ACNT	Count for 1st Dimension
BCNT	Count for 2nd Dimension
DST	Channel Destination Address
SRCBIDX	Source BCNT Index
DSTBIDX	Destination BCNT Index
LINK	Link Address
BCNTRLD	BCNT Reload
SRCCIDX	Source CCNT Index
DSTCCIDX	Destination CCNT Index
CCNT	Count for 3rd Dimension
RSVD	Reserved

For a detailed explanation of the configuration of DMA PARAM set, see the *EVE Programmers Guide*, available through a TI representative.

8.1.3.5.3 Channel Controller

This section reviews various hardware resources of the channel controller that are programmed from the ARP32 controller of the EVE subsystem.

Parameter set: A parameter set is the program memory of the channel controller that describes the attributes of a transfer. Each parameter set describes one data transfer from a source address to a destination address. A parameter set is a group of eight contiguous 32-bit words and has an address that is aligned on a 256-bit boundary. There are 64 of these parameter sets, allowing the description of 64 transfers ahead of time.

Individual parameter sets are used to describe isolated transfers, or they are linked together to describe a group of related transfers.

There are two types of DMA channels in the channel controller (CC) interface:

- EDMA channel: a data transfer channel that takes a parameter set as an input and requires that an event is triggered by a write to the corresponding event of the event set register (ESR), or by detecting assertion of a hardware event (Event Triggered), or by chain triggering. Software can designate a specific event as the trigger event by programming the mapping between the parameter set and the event in the DCHMAP register, and by mapping a DMA channel to a specific hardware event queue in the DMAQNUM register. EVE has 16 EDMA channels that must be assigned to 1 of 16 events of the ESR register, and mapped to 1 of 2 TC queues using DMAQNUM registers.
- QDMA channel: a data transfer channel that is similar to an EDMA channel, but is automatically triggered by the write to a specific word of parameter set (typically the eighth word). Eight QDMA channels are mapped to one of the two TC queues using the QDMAQNUM register.

An EDMA channel is used to describe a regular sequence of data transfers, in which the amount of data sent or received, and source and destination data address increments between successive transfers does not vary dynamically, but by regular amounts. A QDMA channel is used when it is needed to program in new values for the source and destination addresses. The amount of data to transferred varies from one transfer to another, and for other dynamic situations.

EVENT queue: data transfer events are different physical DMA channels (ESR writes for EDMA channels and parameter set trigger word writes for QDMA channels) and are detected simultaneously. The events are queued based on a fixed priority arbitration scheme; the EDMA channels are higher-priority events than the QDMA channels. Among the two groups of channels, the lowest-numbered channel is the highest priority.

Figure 8-12 shows the block diagram of the channel controller.

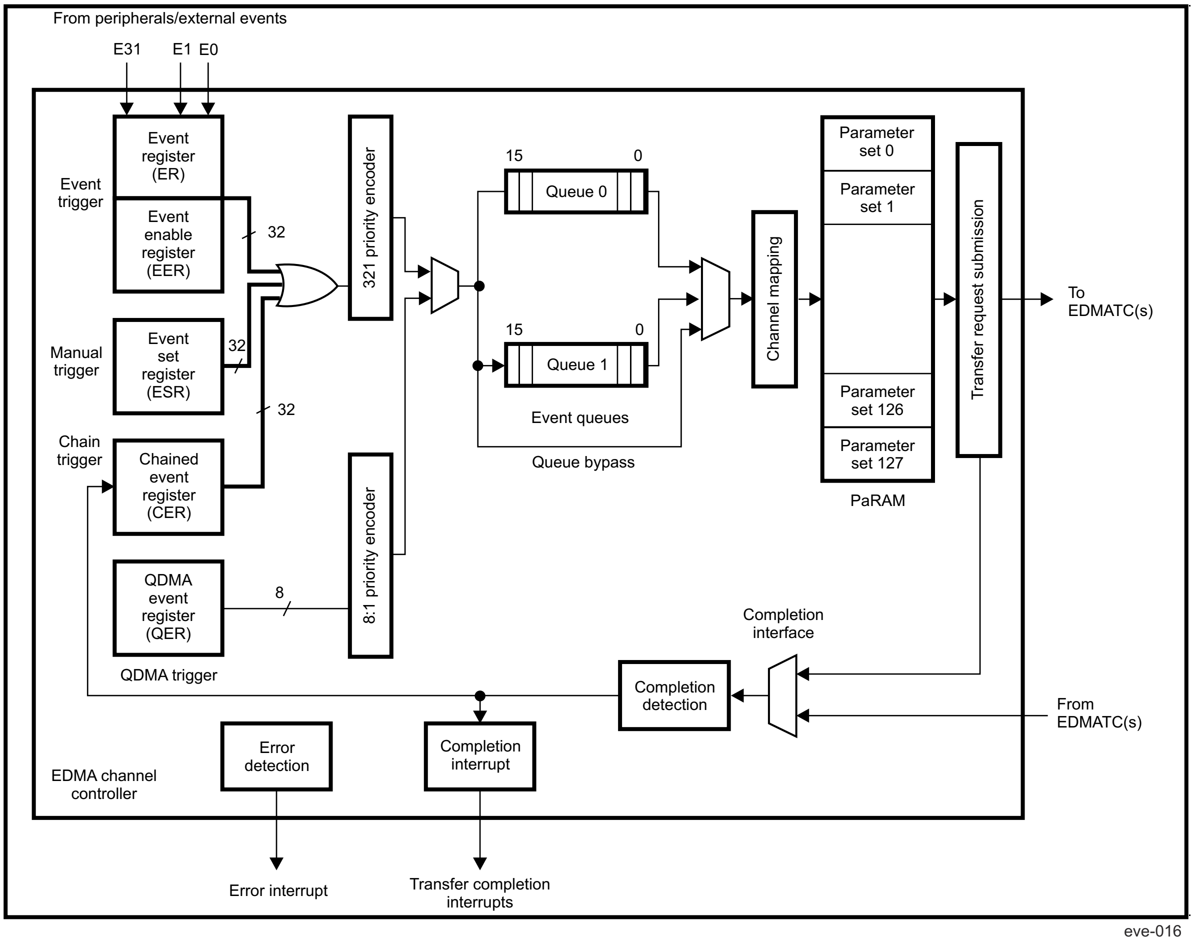


Figure 8-12. Channel Controller Block Diagram

8.1.3.5.4 EVE-Level Bus Width and Throughput

Table 8-12 shows the EVE-level effective bus width and the resulting peak bandwidth. In general, the effective bus width is the minimum of the bus width values between a given initiator and given target. For example, the DMEM port is effectively 4 bytes wide when accessed by ARP32 and is effectively 16 bytes wide when accessed by DMA. Similarly, VCOP memories are 4 bytes, 32 bytes and 16 bytes wide when accessed by ARP32, VCOP, and TCn, respectively.

Table 8-12. EVE-Level Effective Bus Width

		INITIATORS									
		ARP32		VCOP			TC0		TC1		OCP Target Port
		Program	Data	WBUF	IBUFL	IBUFH	Read	Write	Read	Write	
TARGETS	Program cache hit	4									
	Program cache OCP Target		4				4	4	4	4	4
	DMEM		4				16	16	16	16	16
	WBUF		4	32			16	16	16	16	16
	IBUFL		4		32		16	16	16	16	16
	IBUFH		4			32	16	16	16	16	16
	CC MMRs		4				4	4	4	4	4
	TC MMRs		4				4	4	4	4	4

Table 8-12. EVE-Level Effective Bus Width (continued)

	CFGMMRs		4				4	4	4	4	4
	OCP Init0	16	4				16	16			
	OCP Init1								16	16	

8.1.3.5.4.1 Concurrent Transfer Requirements

The two OCP ports are independent from one another to allow full bandwidth and nonblock requests, when EVE TC0 accesses one system-level memory at the same time as EVE TC1 accesses a different system-level memory and both internally access two different IBUF memories. The overall throughput is limited to only 128 bits per cycle in the case when the two TCs are accessing:

- The same system-level memory (EMIF, L3 SRAM)
- The same internal memory buffer

8.1.3.6 General-Purpose Inputs/Outputs

The general-purpose input signal state can be read directly through EVE_GPIN0 (for inputs 31:0) and EVE_GPIN1 (for inputs 63:32). These registers/GPIN signals are also mapped to the ARP32 INTC2 and INTC3 registers respectively.

The GPIs are used to convey state information (readable in the GPIN registers), or they are enabled as interrupts through the corresponding ARP32_INTn_IRQENABLE_SET register. A rising edge on the GP input is latched as a new interrupt in the corresponding ARP32_INTn_IRQSTATUS register.

The EVE subsystem also provides 64 GPOs. These outputs are manipulated by EVE_GPOUTm, EVE_GPOUTm_SET, EVE_GPOUTm_CLR, and EVE_GPOUTm_PULSE. The EVE_GPOUT0 register group controls eve_gpout[31:0] signals and the EVE_GPOUT1 register group controls eve_gpout[63:32] signals. The current state of the output is detected by reading any of the EVE_GPOUTm, EVE_GPOUTm_SET, EVE_GPOUTm_CLR, or EVE_GPOUTm_PULSE registers. The 64 GPOs are used to drive interrupts or state information to other EVEs in a multi-EVE system.

The GPOUTn register can be explicitly set with the corresponding write data word. Byte or halfword writes are permitted in this case (though support may depend on the specific initiator). A read-modify-write sequence can be required if different tasks own different bits within the GPOUT bank. In this case, access for the read-modify-write sequence must be assured. The EVE_GPOUTm_SET register unconditionally drives the corresponding output event to 1. GPIOm_CLR unconditionally drives the corresponding output event to 0. The EVE_GPOUTm_PULSE register unconditionally drives the output to one for four cycles, then deasserts the signal to 0. If the signal is 1, then the EVE_GPOUTm_PULSE simply deasserts the output signal. Software must ensure the output is low before using the EVE_GPOUTm_PULSE feature.

There are several GPOUT signals connected to arbitration/pressure bits (MFlag bits) in the device.

For EVE1 port 1 and EVE1 port 2 (EVE1 TC0 and EVE1 TC1) MFlag is driven by eve1_gpout[63] and eve1_gpout[62] respectively; eve1_gpout[63] is connected to DMM_P1 and EMIF and eve1_gpout[62] is connected to DMM_P2 and EMIF.

EVE2 MFlag is driven by eve2_gpout[63] and eve2_gpout[62] for EVE2_P1 and EVE2_P2 respectively; gpout[63] is connected to DMM_P2 and EMIF and gpout[62] is connected to DMM_P1 and EMIF.

EVE signals are described in Figure 8-3 through Figure 8-4.

EVE MFlag signals are not used inside L3_MAIN. Within the device main interconnect bandwidth regulators are used to control pressure/arbitration of EVE traffic.

8.1.3.7 CME Signaling

The credit management engine (CME) protocol is a credit-based data transfer scheme. A central agent asserts a START signal to a DMA engine or host processor. This signal indicates that a fixed-size data buffer is available to be read or written. Then the DMA engine or host processor is responsible for asserting a DONE signal back to the CME when the data buffer is read or written. The Central Agent uses this information to update its credit accounting (there is a consumer end and producer end for any buffer that is managed by CME). The EVE subsystem generically supports START and DONE signaling, and software must service the request accordingly.

CME START signal:

In a multi-EVE system the CME START signals are broadcast to the eve_evt_int[n:0] input signals of each EVE. These signals are mapped on both EDMA events or ARP32 interrupts. Software enables the appropriate event or input in the appropriate EVE, based on the overall system goals.

For a given EVE, a START signal must either be used to trigger an EDMA transfer directly for a known or agreed quantity of data, or it must trigger an ARP32 interrupt, which then manually sets up an EDMA transfer as necessary.

CME DONE signal:

EVE provides eight CME DONE signals that can be triggered from multiple internal conditions or sources. For any EVE in the system, the DONE output signals are driven either based on any one of the eight EDMA region interrupt outputs, or based on the EVE_CME_DONE_GPOUT register.

The EDMA region interrupt outputs are statically allocated in the device or host. As such, it is desirable to connect higher-numbered "done" bits to the CME to avoid resource allocation issues. The basic logic is essentially a mux select for each respective bit (see Figure 8-13). The selection is done through the EVE_CME_DONE_SEL[31:0] register.

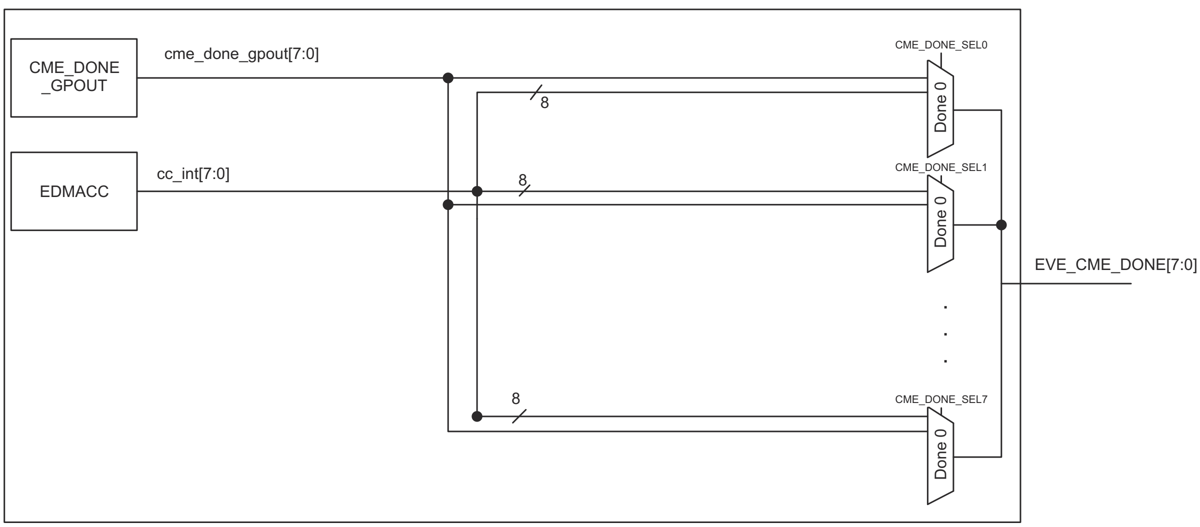


Figure 8-13. CME Done Logic

CME registers in the EVE subsystem are:

- EVE_CME_DONE_GPOUT
- EVE_CME_DONE_GPOUT_SET
- EVE_CME_DONE_GPOUT_CLR
- EVE_CME_DONE_GPOUT_PULSE
- EVE_CME_DONE_SEL
- EVE_CME_DONE_EN

The DONE GPOUT registers are used to directly drive the state of the internal CME DONE output signals. The EVE_CME_DONE_SEL register sets whether the EVE output signals are driven from the EDMA region interrupts, the DONE GPOUT internal value, or a pass through from the eve_passthru signals.

Note

The device does not support hardware CME.

8.1.3.8 Multi-EVE and VIP Usage Models

8.1.3.8.1 Data Partitioning

In this usage model, the data received from the VIP is partitioned between EVEs.

Figure 8-14 shows data partitioning usage model. In this case VIPx is the producer, and EVE1 and EVE2 are consumers. Also at the other end, EVE1 and EVE2 are producers, while VIPx+1 is the consumer.

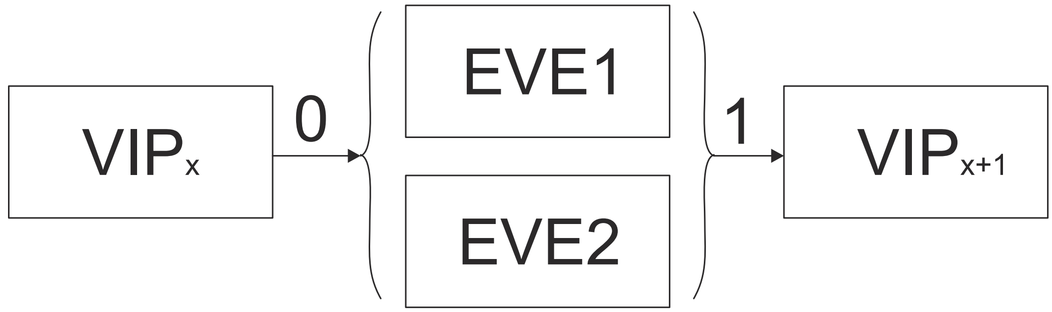


Figure 8-14. Data Partitioning

The CME issues producer start to VIPx, and VIPx returns done when it is done. The CME then broadcasts a consumer START to all the EVEs and combines the consumer DONE it receives from all the EVEs.

The CME issues producer start to EVEs and combines the producer done from all of them before it issues a consumer START to VIPx+1. VIPx+1 indicates a consumer DONE when it is done.

8.1.3.8.2 Task Partitioning

In this usage model, task is partitioned between the EVEs.

Figure 8-15 shows the task partitioning usage model.

In this case EVE1 is consumer for VIPx, EVE2 is the consumer for EVE1, and VIPx+1 is the consumer for EVE2.

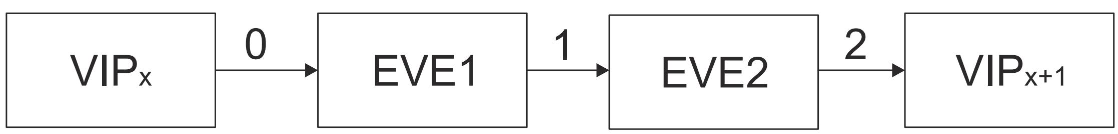


Figure 8-15. Task Partitioning

The CME issues producer start to VIPx, and VIPx returns done when it is done. The CME then indicates a consumer start to EVE1 and receives the consumer done EVE1.

EVEs can use the following methods to communicate the START and DONE signaling between them:

1. **Mailbox**: Each EVE indicates START and DONE to the respective EVE using the Send Remote and Receive Local method. This method gives the most flexible time relationship for the task partitioning, but also adds most load on ARP32

2. **GPIOs**: In this case, each EVE uses GPIOs to communicate Start and Done to each EVE. This method adds the complexity of determining change of state of GPIOs to make sure a change of state is correctly determined.

3. **EDMA events**: Each EVE can trigger an EDMA and indicate DONE directly by using EDMA events and GPOuts for DONE functionality. The event trigger is completely hardware-based handshaking that does not include the ARP32 in the scheme. The DONE generation involves ARP32 as GPOut is used for this functionality. This scheme can only be used where there is deterministic timing relationship between each event and the processing needs in each EVE. In this case, the EVE triggering the EDMA uses its GPOUT signal connected to the INTC1 of the corresponding EVE to trigger the EDMA.

The last EVE in the usage model communicates with VIPx using the standard CME protocol.

8.1.3.8.3

For recommended Multi-EVE Interrupts/Events/Mailboxes/GPIOs recommended connections see Section 8.1.2.1.

8.1.3.9 Memory Management Unit

There are two MMUs. Each of the two EDMA TCs is mapped to a different MMU. One of the MMUs is also shared with ARP32 program and data accesses. The ARP32 program cache and data accesses share an MMU to provide a convenient mechanism whereby ARP32 debug accesses have a single and consistent view of the system, which is equivalent to ARP32 software view. For the EDMA paths, the two MMUs provide maximum concurrency for each TC and its respective accesses to system memory. See Table 8-13 for MMU Configurations

Table 8-13. MMU Configuration

Name	Description	Configuration
MMU_Entries_Number	The number of entries in the TLB cache	32 entries
MMU_Command_Bypass	Bypass command register	2 cycle latency
MMU_Response_Bypass	Bypass response register(depends on clock speed versus technology)	2 cycle latency
MMU_LRU_Enable	LRU algorithm for TLB endry replacement	Disabled
MMU_RRB_Enable	Determines if Generic RRB module is instantiated (Zero only for DSPSS)	Present
LOG_DATA_WIDTH	data width configuration	128-bit data bus

Each MMU allows multiple EVEs to have the same software even when the EVEs are communicating with each other. This can be accomplished by using virtual addresses for each EVE address space. The predominant use of multiple EVEs occurs when adjacent EVEs communicate with each other and send or receive data from adjacent neighbors. In this case, it is suggested that a consistent right neighbor and left neighbor address is defined with MMU for the EVEs. Keeping the right neighbor and left neighbor address the same across different platforms and devices helps with software reuse strategy.

The suggested address mapping for the left neighbor is 0x4100 0000, and for the right neighbor is 0x4200 0000, where the left neighbor for EVEn is EVEn-1 and the right neighbor for EVEn is EVEn+1. For EVE1, the left neighbor is the last EVE implemented in the SoC, and for the last EVE in the SoC, EVE1 is the right neighbor.

For more information about MMU configuration, see Chapter 20, *Memory Management Unit* .

8.1.3.10 Interrupt Control

Each standard block (Interconnect, EDMA, SCTM) provides active high interrupts directly to the ARP32 INTC. In some cases, the submodule IP generated interrupts are mapped as outputs on the EVE boundary.

EVE interrupts (parity error interrupts and memory switch error interrupts) are summarized in Section 8.1.3.10.1, *EVE Interrupt Sources – Memory Switch and Parity Error Interrupts*. EVE accepts active-high input interrupts from the system and maps them to system-level mailbox interrupts.

Figure 8-16 is an overview of EVE interrupt topology.

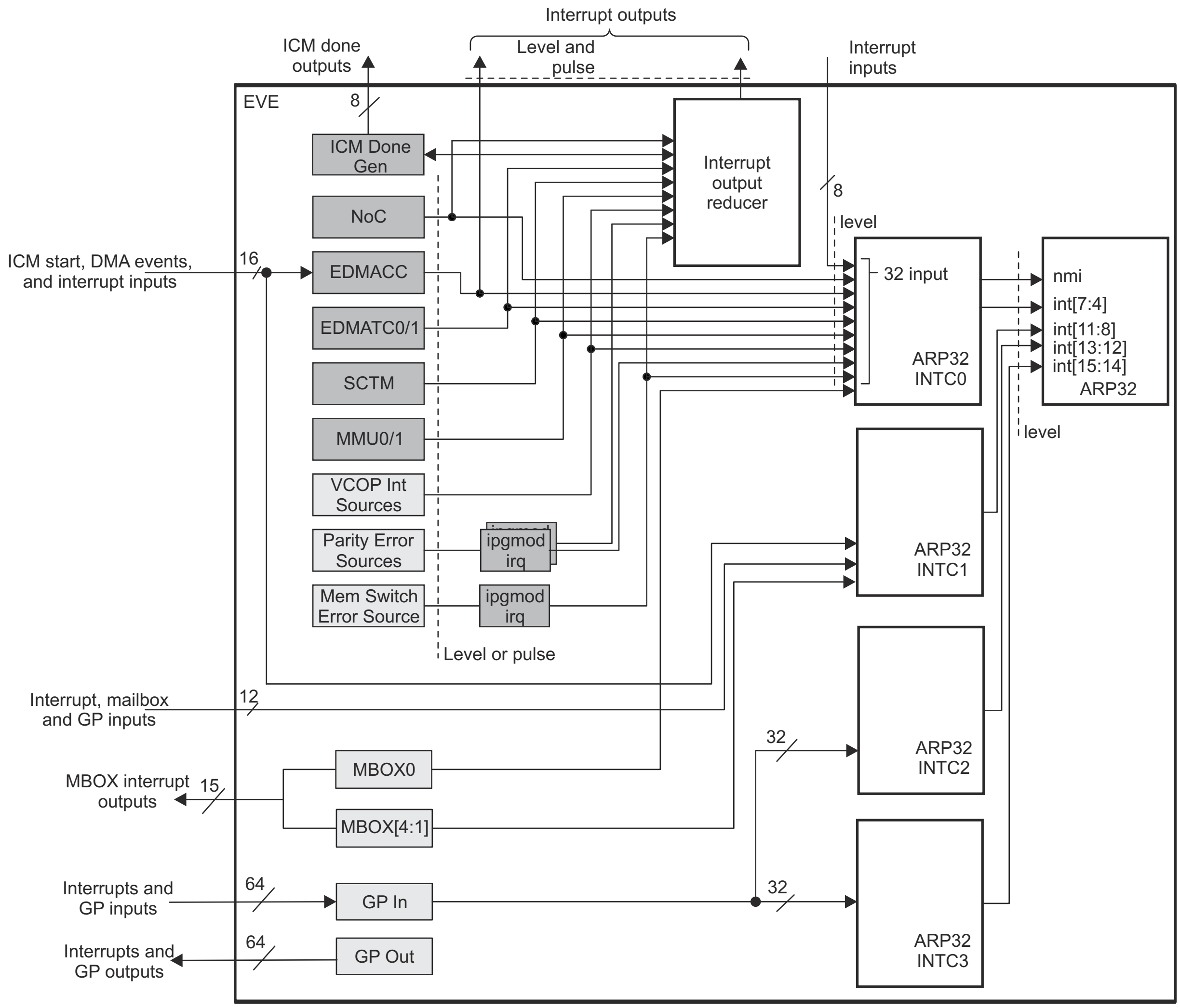


Figure 8-16. EVE Interrupt Block Diagram

8.1.3.10.1 EVE Interrupt Sources – Memory Switch and Parity Error Interrupts

Figure 8-17 illustrates the mapping of EVE memory switch errors to a single interrupt output (EVE_MSW_ERR_INT). Figure 8-18 shows the mapping of EVE parity and error detect error interrupt outputs (EVE_ED_LCL_IRQ and EVE_ED_OUT_IRQ).

The errors are captured in the appropriate functional error bit field along with the additional error details (such as ConnID and specific address). Error status interrupts are readable in the EVE_MSW_ERR_IRQSTATUS_RAW registers. Software enables (or disables) interrupt generation for that source by writing to the appropriate bit in the interrupt enable set or clear register. When any enabled EVE_MSW_ERR_IRQSTATUS_RAW bit is set, then an interrupt is generated to the ARP32 INTC on the corresponding IRQ output (enabling of IRQSTATUS_RAW event is done through EVE_MSW_ERR_IRQENABLE_SET[3:0]ENABLE and EVE_MSW_ERR_IRQENABLE_CLR[3:0] ENABLE registers).

When the interrupt is served, software clears the interrupt status register fields and the source error register (through the EVE_MSW_ERR register or the EVE_<MEM>_ED register). New interrupt is latched into the EVE_MSW_ERR_IRQSTATUS register only after the register is cleared and new error occurs on the corresponding source error bit or signal. The appropriate bit in the EVE_MSW_ERR_IRQSTATUS_RAW register must be set on a low-to-high transition of the corresponding source error bit or signal. The EVE_MSW_ERR_IRQSTATUS register is routed to EVE-level interrupt output (active-high level and pulse signals are output) and as an active-high level interrupt to the ARP32 INTC.

Two interrupts are provided for EVE parity or error detect interrupt. The first is the local version (EVE_ED_LCL_IRQ) provided as an active-high level interrupt to the ARP32 INTC. The second is an output version (EVE_ED_OUT_IRQ) provided as an output on the EVE boundary as both active-high level and pulse version. This lets software selectively control which parity error sources are serviced by the ARP32, and which parity error sources are serviced by the external host. Typically, ARP32 services parity error interrupts generated by VCOP, EDMA, or system accesses and the system host services parity error interrupts generated by ARP32 accesses. The following registers control the detection and servicing of those interrupts (EVE_ED_LCL_IRQ and EVE_ED_OUT_IRQ):

- EVE_ED_LCL_IRQSTATUS_RAW
- EVE_ED_LCL_IRQSTATUS
- EVE_ED_LCL_IRQENABLE_SET
- EVE_ED_LCL_IRQENABLE_CLR
- EVE_ED_OUT_IRQSTATUS_RAW
- EVE_ED_OUT_IRQSTATUS
- EVE_ED_OUT_IRQENABLE_SET
- EVE_ED_OUT_IRQENABLE_CLR

The mechanism for detecting LCL and OUT interrupts is the same as the sequence described for the memory switch error interrupts. Any interrupt condition is captured in the IRQSTATUS_RAW registers (EVE_ED_LCL_IRQSTATUS_RAW[31:0] EVENT and EVE_ED_OUT_IRQSTATUS_RAW[32:0]EVENT), but only after enabling the appropriate IRQSTATUS_RAW bits (through EVE_ED_LCL_IRQENABLE_SET[3:0] ENABLE and EVE_ED_LCL_IRQENABLE_SET[3:0] ENABLE bit fields) the event is outputted for further processing (to the EVE parity or error detect error interrupts).

See Table 8-14, Figure 8-18, and Figure 8-18 for more information.

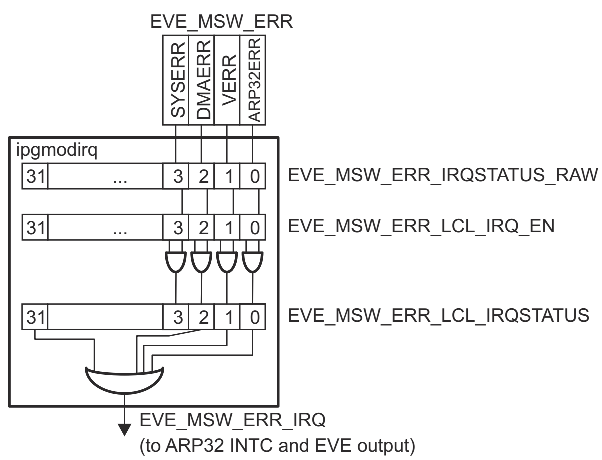


Figure 8-17. EVE Memory Switch Error Interrupt

Table 8-14. EVE_MSW_ERR Register Interrupt Mapping

Bit Position	Source	Error
0	ARP32ERR	ARP32 initiated buffer ownership error
1	VERR	VCOP initiated buffer ownership error
2	DMAERR	EDMA initiated buffer ownership error
3	SYSERR	System-initiated ownership error

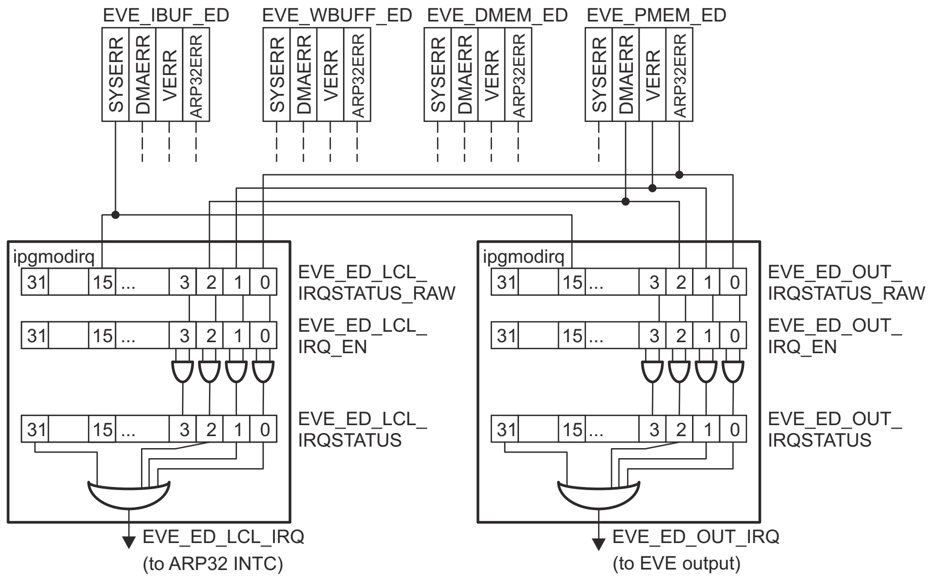


Figure 8-18. EVE Parity/Error Detect Interrupt

Table 8-15. EVE Local and Output Error Detect Error Interrupt Mapping

Bit Position	Source	Error
0	EVE_PMEM_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
1	Reserved	Reserved
2	Reserved	Reserved
3	Reserved	Reserved
4	EVE_DMEM_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
5	Reserved	Reserved
6	EVE_DMEM_ED_STAT[2] DMAERR	EDMA-initiated parity error
7	EVE_DMEM_ED_STAT[3] SYSERR	System-initiated parity error
8	EVE_WBUF_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
9	EVE_WBUF_ED_STAT[1] VERR	VCOP-initiated parity error
10	EVE_WBUF_ED_STAT[2] DMAERR	EDMA-initiated parity error
11	EVE_WBUF_ED_STAT[3] SYSERR	System-initiated parity error
12	EVE_IBUF_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
13	EVE_IBUF_ED_STAT[1] VERR	VCOP-initiated parity error
14	EVE_IBUF_ED_STAT[2] DMAERR	EDMA-initiated parity error
15	EVE_IBUF_ED_STAT[3] SYSERR	System-initiated parity error

8.1.3.10.2 ARP32 INTC

The EVE subsystem instantiates four INTC subblocks. Each ARP32 INTC supports up to 32 active-high level interrupt inputs, and outputs up to five active-high level interrupt outputs. NTC0 maps to NMI and INT[7:4]; INTC1 maps to INT[11:8]; INTC2 maps to INT[13:12]; and INTC3 maps to INT[15:14].

The ARP32_INTn_IRQSTATUS_RAW or ARP32_NMI_IRQSTATUS_RAW register is set when the input signal transitions from a low-to-high state. Software clears the ARP32_INTn_IRQSTATUS_RAW or ARP32_NMI_IRQSTATUS_RAW register by writing 1 into the appropriate bit position. To latch a new level interrupt, software must first clear the source so that a new low-to-high transition is generated on the INTC input. This is compatible with both level and source interrupt sources.

Upon clearing any bit in a given ARP32_[INTn(j)|NMI]_IRQSTATUS_RAW register, if any enabled interrupts are still set, that is ARP32_[INTn(j)|NMI] output signal pulse is low for two clock cycles and is about to transition back to high state. This results in resetting the ARP32 IFR and thus triggers a new ISR, protecting against race conditions. Any new interrupts must occur after the initial read of the ARP32_INTn_IRQSTATUS or ARP32_NMI_IRQSTATUS state.

Upon entering an ISR, software must first read from the EVE_INTk_OUT_IRQSTATUS register, which may show multiple enabled interrupts pending. Software clears the state of those enabled interrupts that are set (by writing 1 to EVE_INTk_OUT_IRQENABLE_CLR and IRQSTATUS bit) and that are dispatched for servicing. Software determines which interrupt source to service first through whatever scheme is convenient. Upon servicing the selected interrupt, software clears the original source interrupt status.

Figure 8-19 shows the INTC for ARP32.

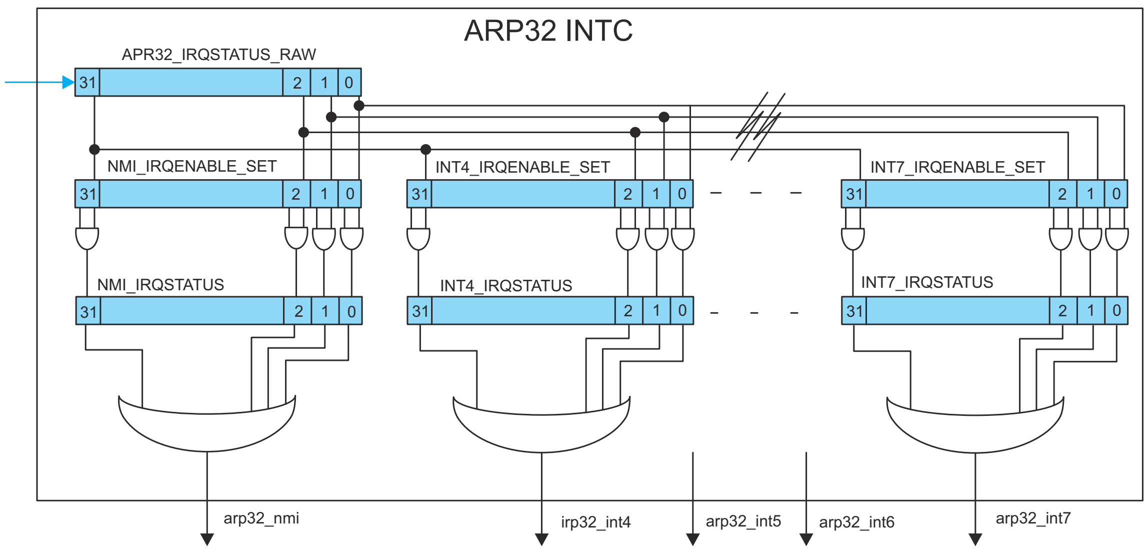


Figure 8-19. EVE INTC for ARP32

Table 8-16, Table 8-17, Table 8-18, and Table 8-19 summarize the ARP32 interrupt mapping.

Table 8-16. EVE ARP32 Interrupt Event Mapping Group0/INTC0

Interrupt	Name	Description	Source
0	eve_int00 (1)	Mapping to Mailbox 0	EVE input
1	eve_int01 (1)	Mapping to Mailbox 1	EVE input
2	eve_int02 (1)	Mapping to Mailbox 2	EVE input
3	eve_int03 (1)	Reserved	EVE input
4	eve_int04 (1)	Reserved	EVE input
5	eve_int05 (1)	Reserved	EVE input
6	eve_int06 (1)	Reserved	EVE input
7	eve_int07 (1)	Reserved	EVE input
8	tpcc_intg	EDMA CC global interrupt	EDMA CC
9	tpcc_int0	EDMA CC region 0 interrupt	EDMA CC
10	tpcc_int1	EDMA CC region 1 interrupt	EDMA CC
11	tpcc_int2	EDMA CC region 2 interrupt	EDMA CC
12	tpcc_int3	EDMA CC region 3 interrupt	EDMA CC
13	tpcc_int4	EDMA CC region 4 interrupt	EDMA CC
14	tpcc_int5	EDMA CC region 5 interrupt	EDMA CC

Table 8-16. EVE ARP32 Interrupt Event Mapping Group0/INTC0 (continued)

Interrupt	Name	Description	Source
15	tpcc_int6	EDMA CC region 6 interrupt	EDMA CC
16	tpcc_int7	EDMA CC region 7 interrupt	EDMA CC
17	SCTM_TIMEVNTINT0	SCTM timer interrupt 0	SCTM
18	SCTM_TIMEVNTINT1	SCTM timer interrupt 1	SCTM
19	vcop_done [(2)]	VCOP done	VCOP
20	vcop_err_intn	VCOP error	VCOP
21	mmu0_int	MMU0 interrupt	MMU0
22	mmu1_int	MMU1 interrupt	MMU1
23	tpcc_errint	EDMA CC error interrupt	EDMA CC
24	tptc_errint0	EDMA TC0 error interrupt	EDMA TC0
25	tptc_errint1	EDMA TC1 error interrupt	EDMA TC1
26	noc_errint	Interconnect error interrupt	Interconnect
27	EVE_MSW_ERR_INT	Buffer error interrupt	EVE top
28	EVE_ED_LCL_ERR_INT or EVE_ED_OUT_ERR_INT [(3)]	Parity error interrupt	EVE top
29	mailbox0_interrupt0	Mailbox 0 interrupt 0	Mailbox 0
30	mailbox1_interrupt0	Mailbox 1 interrupt 0	Mailbox 1
31	Reserved	Reserved	Reserved

(1) eve_int[7:0] are available only to the ARP32 interrupt controller. They are unavailable and reserved to the output interrupt reducer.
(2) Vcop_done will deassert when new instructions are issued. Using the interrupt service routine, the interrupt is cleared in ARP32 INTC. Clearing of the actual source is with the functional mechanism of sending new instructions.
(3) EVE_ED_LCL_ERR_INT (EVE_ED_LCL_IRQSTATUS_RAW, EVE_ED_LCL_IRQSTATUS) is used by the ARP32 INTC and EVE_ED_OUT_ERR_INT (EVE_ED_OUT_IRQSTATUS_RAW, EVE_ED_OUT_IRQSTATUS) is used for the output interrupts reducer.

Table 8-17. EVE ARP32 Interrupt Event Mapping Group1/INTC1

Interrupt	Name	INTC1 Mapping	Description	Source
0	eve_evt_int[0]	eve_intc1[00]	Require mapping to ICM_cstart0 signal or VIP and VPE interrupt	EVE input
1	eve_evt_int[1]	eve_intc1[01]	Require mapping to ICM_cstart1 signal or VIP and VPE interrupt	EVE input
2	eve_evt_int[2]	eve_intc1[02]	Require mapping to ICM_cstart2 signal or VIP and VPE interrupt	EVE input
3	eve_evt_int[3]	eve_intc1[03]	Require mapping to ICM_cstar3 signal or VIP and VPE interrupt	EVE input
4	eve_evt_int[4]	eve_intc1[04]	Require mapping to ICM_pstart0 signal or VIP and VPE interrupt	EVE input
5	eve_evt_int[5]	eve_intc1[05]	Require mapping to ICM_pstart1 signal or VIP and VPE interrupt	EVE input
6	eve_evt_int[6]	eve_intc1[06]	Require mapping to ICM_pstart2 signal or VIP and VPE interrupt	EVE input
7	eve_evt_int[7]	eve_intc1[07]	Require mapping to ICM_pstart3 signal or VIP and VPE interrupt	EVE input
8	eve_evt_int[8]	eve_intc1[08]	General purpose interrupt and EDMA event from EVE1	EVE input
9	eve_evt_int[9]	eve_intc1[09]	General purpose interrupt and EDMA event from EVE2	EVE input
10	Reserved	Reserved	Reserved	Reserved
11	Reserved	Reserved	Reserved	Reserved
12	eve_evt_int[12]	eve_intc1[12]	Not used	Not used
13	eve_evt_int[13]	eve_intc1[13]	Not used	Not used

Table 8-17. EVE ARP32 Interrupt Event Mapping Group1/INTC1 (continued)

Interrupt	Name	INTC1 Mapping	Description	Source
14	eve_evt_int[14]	eve_intc1[14]	Not used	Not used
15	eve_evt_int[15]	eve_intc1[15]	Not used	Not used
16	eve_int1[0]	eve_intc1[16]	Require mapping to remote EVE1 Mailbox interrupt. Reserved.	EVE input
17	eve_int1[1]	eve_intc1[17]	Require mapping to remote EVE2 Mailbox interrupt. Reserved.	EVE input
18	Reserved	Reserved	Reserved	Reserved
19	Reserved	Reserved	Reserved	Reserved
20	eve_int1[4]	eve_intc1[20]	Not used	Not Used
21	eve_int1[5]	eve_intc1[21]	Not used	Not Used
22	eve_int1[6]	eve_intc1[22]	Not used	Not Used
23	eve_int1[7]	eve_intc1[23]	Not used	Not Used
24	eve_int1[8]	eve_intc1[24]	General-purpose interrupt	EVE input
25	eve_int1[9]	eve_intc1[25]	General-purpose interrupt	EVE input
26	eve_int1[10]	eve_intc1[26]	General-purpose interrupt	EVE input
27	eve_int1[11]	eve_intc1[27]	General-purpose interrupt	EVE input
28	mailbox2_interrupt0	eve_intc1[28]	Mailbox 2 interrupt 0	Mailbox 2
29	Reserved	eve_intc1[29]	Reserved	Reserved
30	Reserved	eve_intc1[30]	Reserved	Reserved
31	Reserved	Reserved	Reserved	Reserved

Table 8-18. ARP32 Interrupt Mapping for Group2/INTC2

Interrupt	Name	Description	Source
0	eve_gpin[00]	GP input 00	GPI register
1	eve_gpin[01]	GP input 01	GPI register
2	eve_gpin[02]	GP input 02	GPI register
...	...	...	...
31	eve_gpin[31]	GP input 31	GPI register

Table 8-19. ARP32 Interrupt Mapping for Group3/INTC3

Interrupt	Name	Description	Source
0	eve_gpin[32]	GP input 32	GPI register
1	eve_gpin[33]	GP input 33	GPI register
2	eve_gpin[34]	GP input 34	GPI register
...	...	...	...
31	eve_gpin[63]	GP input 63	GPI register

8.1.3.10.3 Output Interrupt Reduction

EVE provides four output interrupts based on the same approach shown previously for the ARP32 INTC. The EDMA region interrupts, as well as the four output interrupts, are provided at the EVE boundary as shown in Figure 8-16.

For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

Table 8-16 shows the interrupt mapping for the output interrupt reducer.

8.1.3.10.4 End of Interrupt Mapping

End of interrupt (EOI) is a feature used for software handshake with pulsed interrupts.

EOI MMR functionality is provided for EVE-generated interrupts that are mapped as outputs to the system. This feature is necessary when the external system uses the pulsed version of a given interrupt. Because all interrupts inside EVE are level interrupts, the EOI feature is not required for those interrupts.

Note

The EVE subsystem does not condition interrupts for internal submodules (EDMA, interconnect, MMU, or SCTM). Those interrupts are mapped directly as outputs to the system and no EOI functionality is provided for them.

Table 8-20 lists the EOI mapping for the required interrupts.

Table 8-20. EVE EOI Mapping

EVE MMR Value	EVE Interrupt Overview
1	eve_int0_out
2	eve_int1_out
3	eve_int2_out
4	eve_int3_out

8.1.3.11 Interprocessor Communication

This kind of communication is managed through mailboxes. The mailbox function supports 2-way communication between a maximum of four users. This function relies on internal submodules, each supporting 1-way communication between one user referred to as the sender and another user referred to as the receiver. Software allocates the mailbox submodule to the communication between two users. EVE includes internal mailbox to support synchronization and messages passing between EVE-, ARP32-, and system-level hosts.

8.1.3.11.1 Mailbox Configuration

The mailbox submodule includes support for four users and, therefore, provides four interrupt outputs. MAILBOX_Interrupt0 is mapped to the EVE INTC and MAILBOX_Interrupt[3:1] are mapped to three external host processors.

Six submailboxes are provided, allowing bidirectional communication between the ARP32 and the three external hosts:

- MAILBOX_MESSAGE0: HOST0 to EVE
- MAILBOX_MESSAGE1: HOST1 to EVE
- MAILBOX_MESSAGE2: HOST2 to EVE
- MAILBOX_MESSAGE3: EVE to HOST0
- MAILBOX_MESSAGE4: EVE to HOST1
- MAILBOX_MESSAGE5: EVE to HOST2

In the case of EVE-to-EVE communication, a send-remote-receive-local scheme is used, and thus software can leave MESSAGE[5:3] unused.

The listed allocation in this section is just an example. For more information about the Mailbox module, see *Mailbox*.

8.1.3.11.1.1 Mailbox 0 – EVE to DSP1, DSP2 and MPU

Mailbox 0 is dedicated to bidirectional communication between local EVE, two DSPs (DSP1 & DSP2), and a host MPU. In this mailbox, 6 out of 16, submailboxes are used, 3 for senders and 3 for receivers. Additional mailboxes are available, if needed, to separate message traffic based on type or content between each sender and receiver pair (see Table 8-21).

Table 8-21. EVE to DSP1, DSP2 and MPU Mapping

	Receiver			
	EVEn	DSP1	DSP2	MPU

Table 8-21. EVE to DSP1, DSP2 and MPU Mapping (continued)

		Receiver			
Sender	EVEn	-	En_M0	En_M0	En_M0
	DSP1	En_M0	-	-	-
	DSP2	En_M0	-	-	-
	MPU	En_M0	-	-	-

8.1.3.11.1.2 Mailbox 1 – EVE to Other Hosts

Mailbox 1 is dedicated to bidirectional communication with other system-level hosts (such as additional MPU cores). The actual usage and mapping varies on the different devices.

In a multicore system, additional mailbox modules are included to support generic and lower-frequency, higher-latency communication. This mailbox is used by host that needs tightly coupled interaction with the local EVE.

This mailbox has a similar usage as mailbox 0. Six submailboxes are used, three for senders and three for receivers (see Table 8-22).

Table 8-22. EVE to Other Hosts Mapping

		Receiver			
		EVEn	Host X	Host Y	Host Z
Sender	EVEn	-	En_M1	En_M1	En_M1
	Host X	En_M1	-	-	-
	Host Y	En_M1	-	-	-
	Host Z	En_M1	-	-	-

8.1.3.11.1.3 Mailbox 2 – EVE to EVE in a 2x EVE System

In a system with 2x EVEs, Mailbox2 in each EVE is reserved for "receive" messages from other EVEs. Table 8-23 shows the connection between different EVEs. Since EVE implements send remote, receive local methodology for mailboxes, each EVE receives messages locally, in its own mailbox, from other EVEs. While it sends messages to other EVEs to their mailboxes respectively. This reduces the latency incurred in accessing the mailboxes.

Table 8-23. EVE to EVE Mapping

		Receive Local	
		EVE1	EVE2
Sender	EVE1	-	E1_M2
	EVE2	E0_M2	-

8.1.3.12 Powerdown

The EVE subsystem supports two methods of power-minimization and clock gating:

- Active mode power minimization
- Extended duration sleep

8.1.3.12.1 Extended Duration Sleep

Extended duration sleep is an explicitly requested powerdown mode handshaking with the device-level PRCM. As a result, the EVE input clocks are gated. The EVE subsystem is able to latch new interrupts and wake-up requests while in this clock gated state. Internal state is retained, including ARP32 registers, ARP32 data SRAM and program cache contents, VCOP state, EDMA registers, PaRAM, EVE level registers, and so forth.

Since the EVE subsystem has only one logical clock domain (even though divided clocks are used) the PRCM only gates clocks to the EVE subsystem after all of the appropriate handshakes (that is IDLE, STANDBY, and OCP Disconnect) are concluded and the EVE subsystem is in a completely IDLE, STANDBY, and DISCONNECTED state.

8.1.3.12.1.1 Sequence Overview

To enter extended duration sleep mode, the system host first signals the EVE subsystem (through systemlevel mailbox and interrupt) that it is entering this mode. In parallel, the PRCM asserts an SIdleReq request to the EVE subsystem. ARP32 performs any software book-keeping necessary to transition the EVE subsystem to a quiescent state. This can include:

- Waiting for outstanding DMA transfers to complete
- Waiting for VCOP to finish the current task
- Servicing pending interrupts

If the EVE_SYSCONFIG[5:4] STANDBYMODE bit field is set accordingly, then the EVE hardware transitions to IDLE state and signals the desire to enter powerdown state to the device through the STANDBY and Idle protocols. After handshakes are completed internal EVE clocks are gated.

Figure 8-20 depicts the extended duration sleep software/hardware sequence.

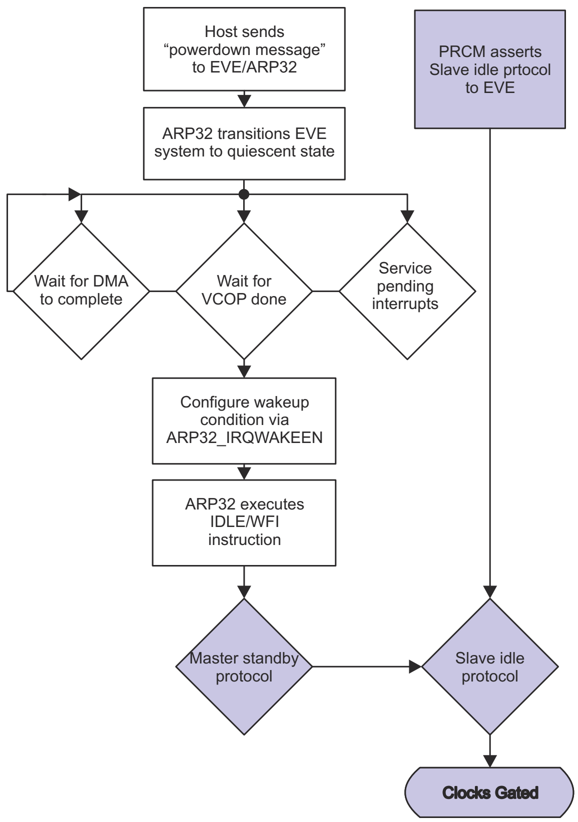


Figure 8-20. Extended Duration Sleep Software/Hardware Sequence

8.1.3.12.1.2 Idle Protocol Overview

EVE IDLE protocol is associated with the EVE OCP target port. The configuration of the IDLE state is done by setting the EVE_SYSCONFIG[3:2] IDLEMODE bit field.

When the EVE subsystem is configured in Smart-Idle or in Smart-Idle-Wakeup, the following sequence must be performed upon an IDLE request from the PRCM:

- Acknowledge the receipt of the request by going into sleeptrans state.
- Wait for MSTANDBY state.
- Disconnect all OCP target ports using the OCP disconnect protocol.
- Gate EVE internal clocks.
- Acknowledge the IDLE request by going into IDLE state.

Externally generated OCP requests (through DMA OCP target bus) to EVE memory space are gated only after the MStandby state is entered. This is required because explicit messaging with either the MPU or the DMA may be necessary before ARP32 software can reach a state that allows entrance to MStandby.

8.1.3.12.1.3 Mstandby Protocol Overview

MSTANDBY protocol is initiated based on ARP32 executing idle instruction provided the other internal EVE masters are in idle state (assuming in Smart-Standby mode).

ARP32 software must wait until all internal operations are finished, before executing idle instruction.

The standby sequence follows:
1. ARP32 software waits for all activity to stop.
2. ARP32 issues IDLE instruction and all EVE submodules are in IDLE state.
3. Using the OCP disconnect protocol, deactivate all OCP initiator ports.
4. Assert the MStandby signal.

8.1.3.12.1.4 IDLE Wakeup

To facilitate auto-idle wakeup of the EVE subsystem, the wakeup capability of the idle protocol is used. A wake-up operation is executed when SYSCONFIG[3:2] IDLEMODE = 3h (SmartIdle-Wakeup).

In this mode, while in IDLE state, if an external input interrupt source that is enabled through the ARP32_IRQWAKEEN mask is active, then the SWakeup signal is asserted to the system. The device is expected to monitor the SWakeup request and enable clocks and exit the STANDBY and IDLE states. At this point, ARP32 can branch to the pending ISR.

The SWakeup signal is deasserted when all interrupt signals that are enabled in the IRQWAKEEN register are deasserted.

The logic controlling the assertion of SWakeup is completely asynchronous and does not rely on any input clock.

ARP32_IRQWAKEEN is not replicated per NMI, INTx. Any interrupt enabled in the ARP32_IRQWAKEEN mask must be enabled in one of the corresponding ARP32_INT_IRQENABLE registers.

8.1.3.13 Hardware-Assisted Software Self-Test – MISRs

To facilitate software self-test, MISRs are instantiated on address and data buses at key points in the system, such as ARP32 interfaces and the interconnect-WBUF interface. ARP32 is covered because it is the key control engine. WBUF coverage is provided as a convenient central destination that is used to indirectly provide coverage for a majority of EVE logic.

The MISRs monitor the address and data buses and calculate a signature based on the address or data pattern on a valid address or data phases. The signature registers reset to 0. A different speed value can be manually written through software to each signature register: MISR0_A, MISR0_D, MISR1_A, MISR1_D, and MISR2_Dk. Based on a known memory access data pattern, the MISR signature can be predicted or calculated and used as a reference for subsequent tests that occur at boot time or during runtime in a safety critical application.

The MISR calculation is a shift-register/XOR tree calculation using classical CRC algorithms.

The MISR calculation is initiated by setting the MISR_CTL[2:0] ENABLE bit field:
- bit 0: enable MISR 0 – ARP32 PMEM path
- bit 1: enable MISR 1 – ARP32 DMEM path
- bit 2: enable MISR2 – Interconnect WBUF path

The MISR_CLEAR[2:0] CLEAR register clears the MISR 0 and MISR 1 calculations for the corresponding path.

Table 8-24 describes the location and width of the various MISRs in the system and is shown as thick blue lines in the block diagram in Figure 8-1.

Table 8-24. MISR Mapping

		Address	Read Data	Write Data	Bus Width	MISR algorithm
MISR0_A	ARP32/PMEM	x			32	CRC-32

Table 8-24. MISR Mapping (continued)

		Address	Read Data	Write Data	Bus Width	MISR algorithm
MISR0_D	ARP32/PMEM		x		32	CRC-32
MISR1_A	ARP32/DMEM	x			32	CRC-32
MISR1_D	ARP32/DMEM			x	32	CRC-32
MISR2_Dk (k = 0:3)	Interconnect/WBUF			x	128	CRC-32 (*4)

8.1.3.13.1 Mapping of MISRs to Different Width Buses

For WBUF data MISR, four instances of CRC-32 MISRs are instantiated in parallel to populate the width of the data bus. For ARP32 interfaces, a single instance of CRC-32 MISR is instantiated on each interface.

Each write data bus MISR (MISR1_D and MISR2_Dk) is activated based on the byte-enable pattern for the corresponding 32-bit word of memory. If any single byte-enable signal (of the 4 for the 4-bytes in the word) is active, then that MISR is updated. Byte-enables that are active use the corresponding data bus value. To avoid "don't care" signals from corrupting the MISR calculation, byte-enables that are inactive use a value of 0 on the input to the 32-bit MISR.

8.1.3.13.2 Detection of Valid Address and Data Cycles

For each MISR in the EVE subsystem, the update cycle occurs for functional accesses, once per read or write request or address phase for address based MISRs, and once per data phase for data-bus based MISRs. If a phase is extended due to "not ready" condition on the target, then the MISR is not updated on that cycle. MISR is updated only on the cycle where the REQUEST and READY signals are active.

8.1.3.13.3 Creating a Unique Signature – Software Self-Test Implications

To create tests that produce a unique signature, software must take care to ensure that the order of transactions (and resulting data/address phases) to a given MISR are consistent. Varying time between transactions does not affect the signature, because it is not updated on idle or not-ready cycles.

For cases that use a single initiator to access a given MISR, crossing asynchronous boundaries is acceptable. This is always true for ARP MISRs, but not necessarily for the WBUF MISRs.

For example if TC0 reads the OCP initiator bus and writes to WBUF MISR, the result is a consistent signature. And when TC0 reads EMIF address space (which is asynchronous) and writes to WBUF, while TC1 is also writing to WBUF results in different order of transactions.

For cases where all transactions/interactions are synchronous, using multiple initiators results in a consistent signature.

8.1.3.13.4 Multipass Tests Using WBUF MISR

The WBUF MISR is used to indirectly provide coverage for a majority of EVE logic. The EVE-level EDMA can be used to perform memcopy operation between a given source memory range and a WBUF memory address range and thus the WBUF MISR. The source buffer contains known values that are either the interim results of a VCOP self-test, or simply initialized to a known data pattern to provide interconnect level coverage. This is repeated for the necessary paths in the EVE subsystem to provide indirect coverage for those paths.

It is possible to reserve a portion of WBUF memory for the write data accesses during the self-test operation. The necessary reserved address space can be minimized by using EDMA constant mode transfers, which require a minimum of 32 bits of reserved space.

8.1.3.14 Error Recovery – ARP32 and OCP Disconnect

To prevent runaway code from corrupting the remainder of the system, and provide a clean reset/recovery mechanism, the EVE subsystem provides a mechanism to disconnect ARP32 from the remainder of the EVE subsystem, and to disconnect L3 initiator buses from the remainder of the device.

When ARP32 or OCP initiator buses are disconnected, the MPU or debugger can detect the EVE MMRs and memories through the interconnect target bus.

When the ARP32 and OCP buses are disconnected, a full reset and reboot cycle are issued in order to resume normal ARP32<->EVE operation. This is required in order to avoid any asynchronous timing paths due to asynchronous reset assertion to ARP32.

8.1.3.14.1 ARP32 Disconnect

ARP32 disconnect occurs when a specific parity error is detected and is enabled in the corresponding EVE_ED_ARP32_DISC_EN register. A disconnect request can also be issued by setting the EVE_DISC_CONFIG[0] ARP32_DISC bit to 1.

When the ARP32 core is in the process of disconnecting (waiting on in-flight request and response) the EVE_STAT[17:16] ARP32_DISC_STATUS bit field is set to 0x1, meaning "attempting to disconnect". When ARP32 is disconnected on both the program and data interfaces, then this bit field is set to 0x0 (disconnected).

Note

Software must wait for disconnected state before issuing a reset to the ARP32 core. This status ensures the neighboring system is not in a corrupted state

---

8.1.3.14.2 OCP Initiator Disconnect

The OCP initiator disconnect request occurs based on a specific parity error that is enabled in the corresponding EVE_ED_OCPI_DISC_EN register, or when the EVE_DISC_CONFIG[4] OCPI_DISC bit is set to 0x1.

When a disconnect request is issued both OCP initiator buses are disconnected. The disconnect logic forces the OCP initiator buses to stall at a clean boundary, such that EVE reset can be issued without violating the OCP protocol detected by the neighboring L3 interconnect. The disconnect logic also tries to fence any new requests, by allowing the current request handshake to complete, but gating any new OCP requests to the system. When the previously issued requests have received all responses, the interface is drained; at this point, the OCP disconnect protocol enters the M_OFF state.

When the OCP initiators are disconnecting, waiting on in-flight request/responses to complete, the EVE_STAT[21:20] OCPI_DISC_STATUS read value is 0x1 or "attempting to disconnect". Otherwise, when both initiators are disconnected, then the bit field contains the value 0, meaning that initiators are disconnected.
In order to guarantee that the neighboring system is not left in a corrupted state, software must wait for disconnected state before issuing a reset to EVE

Note

Any requests into the OCP target bus must be stopped.

8.1.3.15 Lock and Unlock Feature

The EVE subsystem provides a Lock/Unlock mechanism that helps to prevent unintended access to the various control registers of the EVE control module, or EVE subcomponents. Ten lock and unlock registers are defined, where each register is used to lock and unlock access to a specific area of the memory map. Table 8-25 summarizes the logical groupings.

Table 8-25. Lock Register Mapping

Register Name	Mapping	Locked Registers
MMR_LOCK0	EVE memory configuration and standard EVE registers	EVE_REVISION, EVE_HWINFO, EVE_SYSCONFIG, EVE_STAT, EVE_DISC_CONFIG, EVE_BUS_CONFIG, EVE_VCOP_HALT_CONFIG, EVE_MMU_CONFIG
MMR_LOCK1	Memory switch registers and error registers	EVE_MEMMAP, EVE_MSW_CTL, EVE_MSW_ERR, EVE_MSW_ERRADDR

Table 8-25. Lock Register Mapping (continued)

Register Name	Mapping	Locked Registers
MMR_LOCK2	Program cache registers	EVE_PC_INV, EVE_PC_IBAR, EVE_PC_IBC, EVE_PC_ISAR, EVE_PC_ISAR_DONE, EVE_PC_PBAR, EVE_PC_PBC
MMR_LOCK3	Memory error detection registers	EVE_<Memory>_ED_CTL, EVE_<Memory>_ED_STAT, EVE_<Memory>_EDADDR, EVE_<Memory>_EDADDR_BO, EVE_ED_ARP32_DISC_EN, EVE_ED_OCPI_DISC_EN
MMR_LOCK4	Interrupt registers and GPout signals and CME signaling	EVE_MSW_ERR_X, EVE_ED_LCL_X, ARP32_NMI_X, ARP32_INTi_X (see [1], [2]), EVE_GPOUTm, EVE_GPOUTm_SET/CLR/PULSE, EVE_GPIN0, EVE_GPIN1, EVE_CME_DONE
MMR_LOCK5	MISR registers	See Section 8.1.5.
MMR_LOCK6	MMU, ARP32, VCOP	See appropriate chapter, MMU, ARP32 Reference Guide, VCOP Reference guide.
MMR_LOCK7	Debug	–
MMR_LOCK8	EDMA channel controller	–
MMR_LOCK9	EVE output interrupts	EVE_ED_OUT_IRQSTATUS_RAW, EVE_ED_OUT_IRQSTATUS, EVE_ED_OUT_IRQENABLE_SET, EVE_ED_OUT_IRQENABLE_CLR, EVE_INTk_OUT_IRQSTATUS_RAW, EVE_INTk_OUT_IRQSTATUS, EVE_INTk_OUT_IRQENABLE_SET, EVE_INTk_OUT_IRQENABLE_CLR [3]

(1) X = IRQSTATUS_RAW, IRQSTATUS, IRQENABLE_SET, or IRQENABLE_CLR
(2) i = 4 to 7 and 8 to 15
(3) k = 0 to 3

8.1.3.16 EVE Memory Map

For ARP32/EDMA view, the entire 32-bit address space is defined. Accesses to addresses less than 0x4000 0000 and greater than or equal to 0x4010 0000 are used to generate accesses to the external system. These accesses are managed through the MMU module. Accesses to addresses 0x4000 0000 to 0x400F FFFF that are generated directly by ARP32/EDMA are used to access the internal EVE address space. The MMU initiator is only able to access external-to-EVE address space, thus EVE logically remaps a virtual address (address generated by ARP32/EDMA below 0x4000 0000 and above 0x4010 0000) to any 32-bit physical address in the system, such that addresses mapped to 0x4000 0000 still occur. In other words, the EVE address space coexists with the same address space mapped for a different function at system memory through the MMU translation.

EDMA views a compressed/aliased version of the VCOP memories. This is dependent on the EVE_MEMMAP[4] LCL_EDMA_ALIAS bit.

VCOP address space is limited to the VCOP memories. The VCOP memory map depends on the setting of the EVE_MEMMAP[0] VCOP_ALIAS bit.

The system address space represents accesses to the external system through the OCP target bus. At the system level, this memory range is mapped to a fixed base address. Table 8-26 lists the address that represents the offset relative to that base address.

Table 8-26 shows the EVE subsystem memory map, with separate columns to show the ARP32/EDMA, VCOP, and SYS views.

Table 8-26. EVE Subsystem Memory Map

ARP32/EDMA View		VCOP View		SYS View		Size	Region	Function
Start Address	End Address	Start Address	End Address	Start Address	End Address			
0h	3FFF FFFFh					1024MB	MMU0/MMU1	System Memory through MMU
4000 0000h	4001 FFFFh			0	1 FFFFh	128KB	Reserved	–
4002 0000h	4002 7FFFh			2 0000h	2 7FFFh	32KB	DMEM	Data memory
4002 8000h	4003 FFFFh			2 8000h	3 FFFFh	96KB	Reserved	–
4004 0000h	4004 7FFFh	0	7FFFh	4 0000h	4 7FFFh	32KB	WBUF	VCOP working buffer
4004 8000h	4004 FFFFh	8000h	FFFFh	4 8000h	4 FFFFh	32KB	Reserved	–
4005 0000h	4005 3FFFh	1 0000h	1 3FFFh	5 0000h	5 3FFFh	16KB	IBUFLA (or IBUFLB)	Image buffer low copy A (or B for VCOP)
4005 4000h	4005 7FFFh	1 4000h	1 7FFFh	5 4000h	5 7FFFh	16KB	IBUFHA (or IBUFHB)	Image buffer high copy A (or B for VCOP)
4005 8000h	4006 FFFFh	1 8000h	2 FFFFh	5 8000h	6 FFFFh	96KB	Reserved	–
4007 0000h	4007 3FFFh	3 0000h	3 3FFFh	7 0000h	7 3FFFh	16KB	IBUFLB	Image buffer low copy B
4007 4000h	4007 7FFFh	3 4000h	3 7FFFh	7 4000h	7 7FFFh	16KB	IBUFHB	Image buffer high copy B
4007 8000h	4007 FFFFh	3 8000h	3 FFFFh	7 8000h	7 FFFFh	32KB	Reserved	–
4008 0000h	4008 0FFFh			8 0000h	8 0FFFh	4KB	SYSTEM	Interrupt, reset, clock, power, buffer switch
4008 1000h	4008 1FFFh			8 1000h	8 1FFFh	4KB	MMU0 CFG	MMU0 configuration and registers
4008 2000h	4008 2FFFh			8 2000h	8 2FFFh	4KB	MMU1 CFG	MMU1 configuration and registers
4008 3000h	4008 3FFFh			8 3000h	8 3FFFh	4KB	ARP32	ARP32 control, including debug
4008 4000h	4008 4FFFh			8 4000h	8 4FFFh	4KB	VCOPC	VCOP control, including debug
4008 5000h	4008 5FFFh			8 5000h	8 5FFFh	4KB	SCTM	Subsystem counter/timer module
4008 6000h	4008 6FFFh			8 6000h	8 6FFFh	4KB	EDMA_TC0	EDMA transfer controller 0
4008 7000h	4008 7FFFh			8 7000h	8 7FFFh	4KB	EDMA_TC1	EDMA transfer controller 1
4008 8000h	4008 8FFFh			8 8000h	8 8FFFh	4KB	SMSET CFG	SMSET configuration interface
4008 9000h	4008 9FFFh			8 9000h	8 9FFFh	4KB	SMSET MSG	SMSET messaging interface
4008 A000h	4008 AFFFh			8 A000h	8 AFFFh	4KB	NoC	Interconnect registers
4008 B000h	4008 BFFFh			8 B000h	8 BFFFh	4KB	MBOX	Mailbox
4008 C000h	4008 FFFFh			8 C000h	8 FFFFh	16KB	Reserved	–
4009 0000h	4009 7FFFh			9 0000h	9 7FFFh	32KB	P$ RAW	Program cache raw
4009 8000h	4009 FFFFh			9 8000h	9 FFFFh	32KB	P$ Tags	Program cache tag
400A 0000h	400A 7FFFh			A 0000h	A 7FFFh	32KB	EDMA_CC	EDMA channel controller
400A 8000h	400A FFFFh			A 8000h	A FFFFh	32KB	Reserved	–
400B 0000h	400B FFFFh			B 0000h	B FFFFh	64KB	Reserved	–
4010 0000h	FFFF FFFFh					3071MB	MMU0/MMU1	System memory through MMU

8.1.3.16.1 VCOP and Local EDMA: IBUF Memory Map Aliasing

To facilitate ping-pong buffer management, EVE allows the local EDMA and VCOP view of IBUFLA versus IBUFLB, and IBUFHA versus IBUFHB memories to be aliased into the same address ranges. ARP32 and System accesses (through OCP target port) always use the unique/256-KB address map for IBUF memory accesses.

Having a VCOP accesses and EVE_MEMMAP[0] VCOP_ALIAS = 0x1 limits VCOP data memory view to 128KB, with A and B sets aliased to the same address range, depending on the MMR ownership bits

(EVE_MSW_CTL). When EVE_MEMMAP[0]VCOP_ALIAS = 0x0, VCOP uses the full 256 KB address. The EVE_MEMMAP[0]VCOP_ALIAS bit has no effect on how other initiators (ARP32, EDMA or SYS) see the VCOP memories. The VCOP view truth table is shown in Table 8-27.If IBUFLA/IBUFHA is set (while VCOP owned), then the memory switch points to the IBUFHA/IBUFLA memory and IBUFHB/IBUFLB is a "don't care". If VCOP accesses address that is shown in gray in Table 8-27, that is reported as an error.

Having local EDMA accesses, when EVE_MEMMAP[4] LCL_EDMA_ALIAS = 0x1, limits local EDMA data memory view to 128KB, with A and B sets aliased to the same address range, dependent on the MMR ownership bits (EVE_MSW_CTL). When EVE_MEMMAP[4] LCL_EDMA_ALIAS = 0x0, EDMA uses the full 256-KB address. The EVE_MEMMAP[4] LCL_EDMA_ALIAS bit has no effect on how other initiators detect the VCOP memories. Table 8-28 is the local EDMA truth table. If IBUFHA or IBUFLA is cleared (while EDMA/system owned), then the memory switch points to the IBUFHA or IBUFLA memory and IBUFHB/LB bit is a don't care bit. If ARP32, EDMA, or system accesses an address shown in gray, then that is reported as an error.

Note

The ALIAS and ownership register are modified when the system is initializated, or between major modes of operation when no memory transactions are ongoing. If the mode changes while accesses are in progress, then a given address may change from valid to reserved (or viceversa), or a given address can change from one memory to another.

Note

ARP32 and system accesses (including the system EDMA) always view the expanded 256-KB unique addresses for access to IBUF memories.

When a memory switch error is detected, the physical address is captured. IBUFLB address is 0x40007 0000 even if in aliased mode.

8.1.3.16.2 ARP32 Write Model – Avoiding Race Conditions

ARP32 uses posted writes exclusively. This implies that writes are "fire-and-forget" and that writes on a given path complete some variable number of cycles after being issued by ARP32. If ARP32 issues a write followed by either a read or write to the same endpoint, then those accesses complete in order. However, if ARP32 issues a write to a given endpoint target (for example MSW_CTL) and then issues a read or write to a different endpoint target (for example IBUF), then the IBUF access may occur before the MSW_CTL write. To assure that a write has landed, that ARP32 must issues a read to the same address range.

In case of memory switch ownership or aliasing modifications, software must ensure that the mode change is established. This is done by reading back the value of the ownership or alias registers.

Table 8-27. VCOP IBUF Aliasing Truth Table

EVE_MEMMAP:	EVE_MSW_CTL Bit				VCOP Address			
VCOP_ALIAS Bit	IBUFLA	IBUFHA	IBUFLB	IBUFHB	10000h	14000h	30000h	34000h
0	0	0	0	0				
0	0	0	0	1				IBUFHB
0	0	0	1	0			IBUFLB	
0	0	0	1	1			IBUFLB	IBUFHB
0	0	1	0	0		IBUFHA		
0	0	1	0	1		IBUFHA		IBUFHB
0	0	1	1	0		IBUFHA	IBUFLB	
0	0	1	1	1		IBUFHA	IBUFLB	IBUFHB
0	1	0	0	0	IBUFLA			
0	1	0	0	1	IBUFLA			IBUFHB
0	1	0	1	0	IBUFLA		IBUFLB	

Table 8-27. VCOP IBUF Aliasing Truth Table (continued)

EVE_MEMMAP: VCOP_ALIAS Bit	EVE_MSW_CTL Bit				VCOP Address			
	IBUFLA	IBUFHA	IBUFLB	IBUFHB	10000h	14000h	30000h	34000h
0	1	0	1	1	IBUFLA		IBUFLB	IBUFHB
0	1	1	0	0	IBUFLA	IBUFHA		
0	1	1	0	1	IBUFLA	IBUFHA		IBUFHB
0	1	1	1	0	IBUFLA	IBUFHA	IBUFLB	
0	1	1	1	1	IBUFLA	IBUFHA	IBUFLB	IBUFHB
1	0	0	0	0			Reserved	Reserved
1	0	0	0	1		IBUFHB	Reserved	Reserved
1	0	0	1	0	IBUFLB		Reserved	Reserved
1	0	0	1	1	IBUFLB	IBUFHB	Reserved	Reserved
1	0	1	0	X		IBUFHA	Reserved	Reserved
1	0	1	1	X	IBUFLB	IBUFHA	Reserved	Reserved
1	1	0	X	0	IBUFLA		Reserved	Reserved
1	1	0	X	1	IBUFLA	IBUFHB	Reserved	Reserved
1	1	1	X	X	IBUFLA	IBUFHA	Reserved	Reserved

Table 8-28. Local EDMA IBUF Aliasing Truth Table

EVE_MEMMAP: LCL_EDMA_ALIAS Bit	EVE_MSW_CTL Bit				EDMA Address			
	IBUFLA	IBUFHA	IBUFLB	IBUFHB	10000h	14000h	30000h	34000h
0	0	0	0	0	IBUFLA	IBUFHA	IBUFLB	IBUFHB
0	0	0	0	1	IBUFLA	IBUFHA	IBUFLB	
0	0	0	1	0	IBUFLA	IBUFHA		IBUFHB
0	0	0	1	1	IBUFLA	IBUFHA		
0	0	1	0	0	IBUFLA		IBUFLB	IBUFHB
0	0	1	0	1	IBUFLA		IBUFLB	
0	0	1	1	0	IBUFLA			IBUFHB
0	0	1	1	1	IBUFLA			
0	1	0	0	0		IBUFHA	IBUFLB	IBUFHB
0	1	0	0	1		IBUFHA	IBUFLB	
0	1	0	1	0		IBUFHA		IBUFHB
0	1	0	1	1		IBUFHA		
0	1	1	0	0			IBUFLB	IBUFHB
0	1	1	0	1			IBUFLB	
0	1	1	1	0				IBUFHB
0	1	1	1	1				
1	0	0	X	X	IBUFLA	IBUFHA	Reserved	Reserved
1	0	1	X	0	IBUFLA	IBUFHB	Reserved	Reserved
1	0	1	X	1	IBUFLA		Reserved	Reserved
1	1	0	0	X	IBUFLB	IBUFHA	Reserved	Reserved
1	1	0	1	X		IBUFHA	Reserved	Reserved
1	1	1	0	0	IBUFLB	IBUFHB	Reserved	Reserved
1	1	1	0	1	IBUFLB		Reserved	Reserved
1	1	1	1	0		IBUFHB	Reserved	Reserved
1	1	1	1	1			Reserved	Reserved

8.1.3.17 Debug Support

The SCTM and SMSET modules facilitate application and kernel-level performance tuning, as well as general debug. The EVE subsystem minimizes debug intrusion.

8.1.3.17.1 ARP32 Debug Support

The EVE subsystem provides the entire IcePick™ and cross-triggering interfaces at the EVE boundary for use at the system level. EVE ARP32 configuration supports two hardware watchpoints. There are no hardware breakpoints.

The ARP32 core supports the following debug features:
- A 32-bit OCP slave port as the debug interface
- Memory-mapped registers showing and controlling debug status of the core
- Ownership mechanism for managing use of all debug features by application code or debug software
- View of CPU resources (program memory, data memory, CPU architectural register, CPU control registers)
- Real-time debug access to CPU resources, if the CPU does not need to be halted to make a debug access
- Unlimited number of simultaneous software breakpoints using the BKPT instruction supported by the core
- Limited number of simultaneous hardware watchpoint (HWWP) units
- Run control:
  – Halting cpu at SWBP, HWBP, HWWP, incoming external Trigger
  – Single stepping in processor - instruction by instruction
  – Resuming CPU from halted state
- Cross triggering:
  – Halting the CPU on an incoming external trigger
  – Sending a trigger pulse out when the CPU halts as a result of SWBP, HWBP, HWWP trigger/match
  – Sending a trigger pulse out when HWBP, HWWP match is detected, even if the CPU is not configured to halt
- Debug control and status registers:
  – Shows the status of debug access, CPU core execution status
  – Shows miscellaneous status like RESET and IDLE state

8.1.3.17.2 SCTM

The SCTM provides general-purpose counters and timers. It also allows counting and measuring of specific internal EVE signals (for example, stall signals). The general-purpose counter/timer functionality is used by both BIOS software and application software. The signal-measuring functions are typically used for performance-related debug. See Section 8.1.6 for more details concerning SCTM.

8.1.3.17.2.1 SCTM Configuration

Table 8-29 shows the SCTM configuration for EVE. The SCTM module is configured for eight 32-bit counters, two of which can be configured as timers. Timers are counters that include a threshold MMR value and an interrupt. The interrupt is pulsed when the counter reaches the programmed threshold. It is possible for any two even and odd counter pairs to be chained (through MMR) to operate as a 64-bit counter. The SCTM configuration in EVE supports the atomic read feature on counter[3:0] pair and counter[5:6] pair.

Table 8-29. SCTM Configuration in EVE

Generic	SCTM Feature/Parameter Details	Value	Number on Device
CTM_NUMINPT	Number of event input signals supported	0 to 127	31
CTM_NUMCNTR	Number of counters in the module	1 to 32	8
CTM_NUMTIMR	Number of timers	0 to 8	2
CTM_TIMINTPOLARITY	Timer interrupt polarity	0 = Active low 1 = Active high	1
CTM_TIMINTWIDTH	Timer interrupt pulse width	1 or more	2
CTM_NUMSTM	Number of counters for STM export	0 to 32	0
CTM_CCMAVAIL	CCM frame export available	0 = No 1 = Yes	0

Table 8-29. SCTM Configuration in EVE (continued)

Generic	SCTM Feature/Parameter Details	Value	Number on Device
CTM_NUMDBGSGL	Number of debug event signals	0 to 8	0
CTM_ASYNCIDLEREQ	Whether IDLE request is asynchronous signal	0 = No 1 = Yes	0
CTM_CNTR_ CHAINSHADOW	Atomicity feature for even-indexed counters	One per counter. 0 = No chain shadow feature 1 = Chain shadow feature in counter chain mode	Counter numbers 2 and 4 have the chaining feature.

8.1.3.17.2.2 SCTM Resources Reserved for BIOS

Timer 0 (and associated interrupt) is exclusively reserved for BIOS software for driving the BIOS tick.

Timer 1 (and associated interrupt) is free to be used by application software.

BIOS also requires a 64-bit free-running counter with low-latency accessibility that is used during application logging. This is provided through the ARP32 free-running 64-bit counter.

8.1.3.17.2.3 SCTM Event Mapping

The SCTM events include low-level stall and duration signals sourced by various components of EVE, mainly program cache-related signals and VCOP-related signals. Table 8-30 summarized the event mapping. Event inputs start numbering at 1, because event0 is reserved and SCTM internally uses the functional clock as event0.

For more details on the functionality of each signal listed in the table, see *EVE Programmer's Guide*. Contact your TI representative for instructions how to access this document.

The SCTM module operates at half the clock rate (EVEx_GFCLK), CLK2 = 0.5 × CLK1. The event sources include CLK2 and CLK1 signals. The CLK2 relative signals are connected directly to SCTM. The CLK1 relative signals are conditioned by EVE-level logic to scale from CLK1 to CLK2. Because these signals are Duration type signals, EVE logic asserts a CLK2 pulse for every two CLK1 pulses detected. This results in, at most, 1 CLK1 cycle of inaccuracy in the CLK2 duration reported by the SCTM.

The SCTM allows measurement of active-high signals. The Type column represents the functional characteristics of the source signals, which can be Pulse, Duration or Edge. The SCTM Mode column describes the typical mode for the SCTM counter when used with a particular signal. The options are event or duration mode. The pulse signals are driven active one cycle for each occurrence, a sequence of consecutive active cycles represents multiple occurrences. For a type of signal where the duration implies the number of occurrences in particular state (for example, the number of cache hits) the furation mode of SCTM is used. The Duration type represents a signal that may stay active for multiple cycles on a given occurrence. These are stall signals, where software tried to determine the number of cycles in the stalled state. Duration mode indicates the time stalled. Event mode indicates the number of unique occurrences of stalls. Event is useful in this case since there is always a deassertion between occurrences. The Edge type goes active for undefined periods of time but must go inactive before the next occurrence. Event mode of the SCTM is used for these signals.

Table 8-30. SCTM Events

SCTM Event	Name	Source	Type	SCTM Mode	Clock
1	cache_miss_count	ARP32_Pcache	Pulse	Duration	CLK2
2	cache_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
3	cache_miss_stall	ARP32_Pcache	Duration	Duration or Event	CLK2
4	prefetch_compulsory_count	ARP32_Pcache	Pulse	Duration	CLK2
5	Prefetch_lookahead_count	ARP32_Pcache	Pulse	Duration	CLK2
6	prefetch_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
7	line_buffer_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
8	Prefetch_line_count	ARP32_Pcache	Pulse	Duration	CLK2

Table 8-30. SCTM Events (continued)

SCTM Event	Name	Source	Type	SCTM Mode	Clock
9	prefetch_discard_stall	ARP32_Pcache	Duration	Duration or event	CLK2
10	tpcc_aet	EDMA	Duration	Duration or event	CLK2
11	arp32_int4	INTC	Duration	Duration or event	CLK2
12	arp32_int5	INTC	Duration	Duration or event	CLK2
13	arp32_int6	INTC	Duration	Duration or event	CLK2
14	arp32_int7	INTC	Duration	Duration or event	CLK2
15	vcop_busy	VCOP	Pulse	Duration	CLK2
16	vcop_idle_and_done	VCOP	Pulse	Duration	CLK2
17	vcop_wait_for_arp32	VCOP	Pulse	Duration	CLK2
18	vcop_arp32_awaits	VCOP	Pulse	Duration	CLK2
19	vcop_overhead	VCOP	Pulse	Duration	CLK1
20	vcop_ld_stall_by_st	VCOP	Pulse	Duration	CLK1
21	vcop_op_stall_by_ldst	VCOP	Pulse	Duration	CLK1
22	vcop_op_stall_by_dependency	VCOP	Pulse	Duration	CLK1
23	vcop_rd_ibufl	VCOP	Pulse	Duration	CLK1
24	vcop_rd_ibufh	VCOP	Pulse	Duration	CLK1
25	vcop_rd_wbuf	VCOP	Pulse	Duration	CLK1
26	vcop_wr_ibufl	VCOP	Pulse	Duration	CLK1
27	vcop_wr_ibufh	VCOP	Pulse	Duration	CLK1
28	vcop_wr_wbuf	VCOP	Pulse	Duration	CLK1
29	vcop_loop_start	VCOP	Edge	Event	CLK2
30	vcop_done	VCOP	Edge	Event	CLK2
31	arp32_nmi	INTC	Duration	Duration or event	CLK2
32	arp32_int8	INTC	Duration	Duration of event	CLK2
33	arp32_int9	INTC	Duration	Duration of event	CLK2
34	arp32_int10	INTC	Duration	Duration of event	CLK2
35	arp32_int11	INTC	Duration	Duration of event	CLK2
36	arp32_int12	INTC	Duration	Duration of event	CLK2
37	arp32_int13	INTC	Duration	Duration of event	CLK2
38	arp32_int14	INTC	Duration	Duration of event	CLK2
39	arp32_int15	INTC	Duration	Duration of event	CLK2

8.1.3.17.2.4 SCTM Halt and Idle Modes

The SCTM module has a method for enabling and disabling counters based on whether the CPU is in debug-halt (SUSPEND) state or idle state (IDLE). This state information is provided from the ARP32 CPU to the SCTM module through the SUSPEND and IDLE inputs.

8.1.3.17.3 SMSET

The SMSET allows tracing of key EVE subsystem events, as well as software messages from the ARP32 CPU. SMSET accepts software messages through its OCP target port, and accepts key system events through the system event input. These messages or events are queued locally in SMSET and written to the chip-level software trace module through the SMSET and EVE OCP debug initiator port. The system trace macrocell (STM) then traces these messages along with trace content with other chip-level agents. For details, see Chapter 33, *On-Chip Debug Support* .

8.1.3.17.3.1 SMSET Configuration

Table 8-31 summarizes the SMSET configuration in EVEN. The SMSET module is configured to support both event and software message tracing, and supports cross-triggering. System events (that are not stallable and can overflow) include a 4-deep buffer; and software events (that can be stalled and never overflow) support a 2-deep buffer.

Table 8-31. SMSET Configuration in EVE

Generic	SCTM Feature/Parameter Details	Value	In EVE Subsystem	Number on Device
NB_EVENTS	Number of system events	1 to 255	52	9
SW_MESSAGE	Software messages support	0 to 1	1	1
TRIG_MIN_WIDTH	Minimum trigger input pulse (L4 cycles)	1 to 8	2	2
EVENTS_BUF_DEPTH	System events buffer depth	1 to 16	4	4
SW_BUF_DEPTH	Software message buffer depth	0 to 16	2	2

8.1.3.17.3.2 SMSET Event Mapping

The SMSET events include higher-level start and end task events. Table 8-32 summarize the event mapping. Refer to the *EVE Programmer's Guide* for details on the functionality of each signal listed. Contact your TI representative for instructions how to access the *EVE Programmer's Guide*.

The SMSET module operates at the CLK2 rate.

The SMSET traces events on either a high-to-low transition or low-to-high transition, but not both. Thus, the EDMAs tpcc_aet signal is not directly compatible with SMSET tracing. The tpcc_aet is an active-high signal that is used to measure the duration of a specific EDMA transfer. The tpcc_aet signal is programmed (in EDMA CC MMRs) to go high when a specific trigger is detected and to go low when a specific completion code is received. EVE logic implements custom tpcc_aet_start and tpcc_aet_end signals. The tpcc_aet_start signal pulses high (for one cycle) when tpcc_aet transitions to high, and the tpcc_aet_end signal pulses high (for one cycle) when tpcc_aet transitions to low.

The Type column in Table 8-32 lists the functional characteristics of the source signal (Pulse, Duration, or Edge).

The Pulse signals are expected to be driven active one cycle for each occurrence. A sequence of consecutive active cycles represent multiple occurrences. Duration mode of SCTM makes sense for this type of signal where the duration implies the number of occurrences in a particular state (for example, the number of cache hits).

The Duration type represents a signal that may stay active for multiple cycles on a given occurrence. These are typically stall signals where the user is trying to determine the number of cycles in the stalled state. Duration mode indicates the total time stalled. Event mode will indicate the number of unique occurrences of stalls. "Event" is useful in this case because there will always be a deassertion between occurrences.

An Edge signal goes active for undefined period of time but is goes inactive before the next occurrence. Event mode of the SCTM must be used for these signals.

Table 8-32. List of SMSET Events

SMSET Event	Name	Source	Type	SMSET Mode	Clock
0	tpcc_aet_start	EDMA	Pulse	Duration	CLK2
1	tpcc_aet_stop	EDMA	Pulse	Duration	CLK2
2	arp32_int4	INTC	Duration	Duration or event	CLK2
3	arp32_int5	INTC	Duration	Duration or event	CLK2
4	arp32_int6	INTC	Duration	Duration or event	CLK2
5	arp32_int7	INTC	Duration	Duration or event	CLK2
6	vcop_loop_start	VCOP	Edge	Event	CLK2
7	vcop_done	VCOP	Edge	Event	CLK2

Table 8-32. List of SMSET Events (continued)

SMSET Event	Name	Source	Type	SMSET Mode	Clock
8	arp32_nmi	INTC	Duration	Duration or event	CLK2
9	arp32_int8	INTC	Duration	Duration or event	CLK2
10	arp32_int9	INTC	Duration	Duration or event	CLK2
11	arp32_int10	INTC	Duration	Duration or event	CLK2
12	arp32_int11	INTC	Duration	Duration or event	CLK2
13	arp32_int12	INTC	Duration	Duration or event	CLK2
14	arp32_int13	INTC	Duration	Duration or event	CLK2
15	arp32_int14	INTC	Duration	Duration or event	CLK2
16	arp32_int15	INTC	Duration	Duration or event	CLK2

8.1.3.18 EVE L2_FNOC Interconnect

A level 2 (L2) Interconnect from Arteris - FlexNoC® is instantiated in the EVE subsystem.

8.1.3.18.1 EVE L2_FNOC Flag Mux and Error Log Registers

The EVE_L2_FNOC flagmux and error log registers are used for error logging and flag muxing purposes. The status information stored in them can be used to resolve issues related to EVE_NoC access coflicts, debugging, and so forth. The related registers are described in Section 8.1.5.3 *EVE L2_FNOC Register Summary and Description*. For more information on flag muxing and error logging, see *L3_MAIN Interconnect Error Handling* in *L3_MAIN Interconnect*.

8.1.4 EVE Programming Model

8.1.4.1 Boot

The EVE boot process follows:

1. EVEn_RST and EVEn_CPU_RST (n = 1, 2) are asserted for a minimum of 16 cycles.
2. Host processor (through PRCM) deasserts eve_rst_n (eve_arp32vcop_rst_n remains asserted). This lets the system access EVE-level memories/MMRs through the OCP target bus while ARP32 remains in reset.
3. Host processor initializes EVE state:
    a. Enable parity generation for EVE memories: clear DMEM, WBUF, IBUF memories to initialize parity state (optional and can be handled at runtime by ARP32).
    b. Load ARP32 code and data into desired memories:
        i. Code is typically in EMIF/DDR memory, but can be any valid chip-level memory.
        ii. Data is typically in EVE DMEM, but can also be in EMIF/DDR or other chip level memory
    c. Initialize MMU TLB and related page tables: A translation must exist for address 0h because ARP32 boot vector is always at 0h (for MMU programming model see Chapter 20, *Memory Management Unit*).
    d. Load VCOP memories: Ownership defaults to non-VCOP so no extra steps are required (optional and can be handled at runtime by ARP32).
    e. Initialize EDMA, INTC (optional and can be handled at runtime by ARP32).
4. Host processor (through PRCM) deasserts eve_arp32vcop_rst_n
5. ARP32 initializes EVE state, for steps not handled by host processor:
    a. Configure EDMA.
    b. Configure interrupt controller: Enable all error-related interrupts.
    c. Enable parity on all memories: Scrub memories to initialize parity state.
    d. Initialize VCOP working buffers.
6. ARP32 manages ping-pong processing:
    a. Page in input buffers through EDMA.
    b. Assign memory ownership of I/O buffers to VCOP or EDMA/SYS.
    c. Page out output buffers through EDMA.
    d. Initialize VCOP program.
    e. Wait for VCOP completion, EDMA input paging completion, and EDMA output paging completion.

8.1.4.2 Task Change and Program Cache Prefetch

The ARP32 switches tasks in a number of ways. In this context, a task represents a unit of program flow that typically executes for many ping-pong time slices and represents a code section that is typically 16k to 32k long.

8.1.4.2.1 Simple or Unoptimized Branch to New Task

In the simplest case, the ARP32 branches to the new code section. The existence of program cache provides reasonable performance, especially if the new task program code is cache friendly and is executed a number of times before the next task switch occurs.

8.1.4.2.2 Prefetch, Wait, then Branch to New Task

To optimize the task switch efficiency, use the software-directed preload operation. In this case:

- Complete with the current task.
- Execute a Software directed preload command (as described in Section 8.1.3.4.3, Software Direct Preload).
- Wait for preload completion.
- Branch to the new address.

Given that the cache view and system view of the program address are still the same, there are no additional overhead and constraints.

8.1.4.2.3 Hidden Prefetch

If any two tasks are small enough to fit in program cache, software uses SDP to fetch the contents of the next tasks into program cache while operating on the current task. If that program cache is direct mapped, extra care must be taken to ensure that any two tasks do not reside in the same modulo-32k range of memory.

8.1.4.3 Interrupts

The following pseudocode describes a typical and simple workflow for ARP32 to service pending interrupts. This flow is not the only possible sequence and is not meant to represent a mandated software sequence:

1. ARP32 enables and maps interrupts:
    a. Enable ARP32 global interrupt enable (ARP32.CSR.GIE).
    b. Enable ARP32 nonmaskable interrupts (ARP32.IER.NMIE).
    c. Enable ARP32 interrupt inputs (ARP32.IER[*n*]).
    d. Map EVE-level interrupts to the appropriate ARP32 interrupt input (ARP32_INTn_IRQENABLE_SET).
2. When interrupt is asserted, ARP32 branches into interrupt vector code:
    a. ARP32 hardware clears the GIE bit, new interrupts are ignored
    b. ARP32 software reads ARP32_INTn_IRQSTATUS
    c. Clear to be serviced interrupts in ARP32_INTn_IRQSTATUS (by issuing write of 1)
    d. For each pending interrupt in ARP32_INTn_IRQSTATUS:
        i. ARP32 software reads IP interrupt status (<IP_INT>_IRQSTATUS)
        ii. For each pending interrupt in <IP_INT>_IRQSTATUS:
            1. May not be necessary for pulsed or timer type interrupts.
            2. Perform functional operations necessary to service interrupt.
            3. Clear IP source interrupt status (by setting correct bit in <IP_INT>_IRQSTATUS to 1).
            4. Ideally, only exit loop after <IP_INT>_IRQSTATUS is all 0s.
        iii. Only exit loop after <IP_INT>_IRQSTATUS is all 0s
    e. ARP32 software returns from interrupt through BIRP (branch interrupt return pointer) instruction: ARP32 hardware sets the GIE bit (by copy of SCSR to CSR, new interrupts are enabled) and clears IFR*n*.

For ARP32 register and state details (especially for CSR, GIE, and SCSR functionalities), see the *ARP32 CPU and Instruction Set Reference Guide*.

8.1.4.4 Safety Considerations

This section highlights the known hardware that must be enabled in a safety-conscious system, and specific software techniques that increases the level of safety of EVE.

8.1.4.4.1 Memory Error Detection

Memory error detection provides hardware to detect single-bit errors in data memories (DMEM, WBUF, and IBUF) and detect double-bit errors in program memories. This hardware can capable detect stuck-at faults, as well as radiation-induced soft errors.

- For data memories, to avoid false error signaling, memories must be initialized with known values before reading those memories. See Section 8.1.3.3.6, *Memory Error Detection*.
- All sources of error detection error interrupts must be enabled. See Section 8.1.3.3.6, *Memory Error Detection,* and Section 8.1.3.10.1, *EVE Interrupt Sources – Memory Switch and Parity Error Interrupts*.
- Software must be tested at development time to ensure robust response to random errors. See Section 8.1.3.3.6.3, *Parity Error Testability*.
- Software must be tested at runtime to ensure hardware is properly operating. See Section 8.1.3.3.6.3, *Parity Error Testability*.

8.1.4.4.2 MMU

The MMU is configured to minimize the view of system memory that ARP32 and EDMA are allowed to access. This prevents either coding bugs or faulty hardware from impacting other critical areas of the system, such as the system host code and data sections. For detailed explanation of the MMU and its programming, see Chapter 20, *Memory Management Unit (MMU)* .

8.1.4.4.3 Firewall

The system firewall is used to prevent access by unintended system initiators from accidentally modifying EVE memory; or accesses from intended system initiators to unintended EVE memory. For firewall configuration see *Interconnect*.

8.1.4.4.4 Interconnect

The EVE detects accesses to reserved locations in the memory-map and informs the system.

8.1.4.4.5 Application Stability/Sequencing

The Mailbox module or the local EVE SMSET module can be leveraged to report either heartbeat or sequence messages to the system host. By sending deterministic messages, the host ARM verifies that EVE is still active and following a reasonable path of code execution.

8.1.4.4.6 Interrupt Servicing

ISRs are implemented to count the expected number of interrupts (for example, EDMA, mailbox, timer) and check for extra or missing interrupts where possible. For example, if the ARP32_INTn_IRQSTATUS[31:0] EVENT bit field is set, then at least one of the source registers is set as well. Or, if an EDMA interrupt is received, then at least one of the CIPR bits is set. For EDMA interrupts, software tracks DMAs submitted versus DMAs completed.

8.1.5 EVE Subsystem Register Manual

8.1.5.1 EVE Instance Summary

Table 8-33 lists EVE instances.

Table 8-33. EVE Instance Summary

Module Name	Base Address	Size
EVE1	0x4200 0000	4KiB
EVE1_DSP[(1)]	0x0200 0000	4KiB
EVE2	0x4210 0000	4KiB
EVE2_DSP[(1)]	0x0210 0000	4KiB

Table 8-33. EVE Instance Summary (continued)

Module Name	Base Address	Size
EVE1_L2_FNOC [(2)]	0x0208 A000	1024 Bytes
EVE2_L2_FNOC	0x4218 A000	1024 Bytes

(1) Address space is NOT visible on L3_MAIN for the device. It is visible only within the DSP_ICFG internal configuration space, hence accessible only by the DSP C66x CPU.
(2) Address space is NOT visible on L3_MAIN for the device.

8.1.5.2 EVE Register Summary and Description

8.1.5.2.1 EVE Register Summary

Table 8-34 summarizes the EVE registers.

Table 8-34. EVE Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1 Physical Address L3 Interconnect	EVE1_DSP Physical Address DSP private	EVE2 Physical Address L3 Interconnect	EVE2_DSP Physical Address DSP private
EVE_REVISION	R	32	0x8 0000	0x4208 0000	0x0208 0000	0x4218 0000	0x0218 0000
EVE_HWINFO	R	32	0x8 0004	0x4208 0004	0x0208 0004	0x4218 0004	0x0218 0004
EVE_SYSCONFIG	RW	32	0x8 0008	0x4208 0008	0x0208 0008	0x4218 0008	0x0218 0008
EVE_STAT	R	32	0x8 000C	0x4208 000C	0x0208 000C	0x4218 000C	0x0218 000C
EVE_DISC_CONFIG	RW	32	0x8 0010	0x4208 0010	0x0208 0010	0x4218 0010	0x0218 0010
EVE_BUS_CONFIG	RW	32	0x8 0014	0x4208 0014	0x0208 0014	0x4218 0014	0x0218 0014
EVE_VCOP_HALT_CONFIG	RW	32	0x8 0018	0x4208 0018	0x0208 0018	0x4218 0018	0x0218 0018
EVE_MMU_CONFIG	RW	32	0x8 001C	0x4208 001C	0x0208 001C	0x4218 001C	0x0218 001C
EVE_MEMMAP	RW	32	0x8 0020	0x4208 0020	0x0208 0020	0x4218 0020	0x0218 0020
EVE_MSW_CTL	RW	32	0x8 0024	0x4208 0024	0x0208 0024	0x4218 0024	0x0218 0024
EVE_MSW_ERR	RW	32	0x8 0028	0x4208 0028	0x0208 0028	0x4218 0028	0x0218 0028
EVE_MSW_ERRADDR	R	32	0x8 002C	0x4208 002C	0x0208 002C	0x4218 002C	0x0218 002C
EVE_PC_INV	RW	32	0x8 0040	0x4208 0040	0x0208 0040	0x4218 0040	0x0218 0040
EVE_PC_IBAR	RW	32	0x8 0050	0x4208 0050	0x0208 0050	0x4218 0050	0x0218 0050
EVE_PC_IBC	RW	32	0x8 0054	0x4208 0054	0x0208 0054	0x4218 0054	0x0218 0054
EVE_PC_ISAR	RW	32	0x8 0058	0x4208 0058	0x0208 0058	0x4218 0058	0x0218 0058
EVE_PC_ISAR_DONE	R	32	0x8 005C	0x4208 005C	0x0208 005C	0x4218 005C	0x0218 005C
EVE_PC_PBAR	RW	32	0x8 0060	0x4208 0060	0x0208 0060	0x4218 0060	0x0218 0060
EVE_PC_PBC	RW	32	0x8 0064	0x4208 0064	0x0208 0064	0x4218 0064	0x0218 0064
EVE_PMEM_ED_CTL	RW	32	0x8 0080	0x4208 0080	0x0208 0080	0x4218 0080	0x0218 0080
EVE_PMEM_ED_STAT	RW	32	0x8 0084	0x4208 0084	0x0208 0084	0x4218 0084	0x0218 0084
EVE_PMEM_EDADDR	R	32	0x8 0088	0x4208 0088	0x0208 0088	0x4218 0088	0x0218 0088
EVE_DMEM_ED_CTL	RW	32	0x8 0090	0x4208 0090	0x0208 0090	0x4218 0090	0x0218 0090
EVE_DMEM_ED_STAT	RW	32	0x8 0094	0x4208 0094	0x0208 0094	0x4218 0094	0x0218 0094
EVE_DMEM_EDADDR	R	32	0x8 0098	0x4208 0098	0x0208 0098	0x4218 0098	0x0218 0098
EVE_DMEM_EDADDR_BO	R	32	0x8 009C	0x4208 009C	0x0208 009C	0x4218 009C	0x0218 009C
EVE_WBUF_ED_CTL	RW	32	0x8 00A0	0x4208 00A0	0x0208 00A0	0x4218 00A0	0x0218 00A0
EVE_WBUF_ED_STAT	RW	32	0x8 00A4	0x4208 00A4	0x0208 00A4	0x4218 00A4	0x0218 00A4
EVE_WBUF_EDADDR	R	32	0x8 00A8	0x4208 00A8	0x0208 00A8	0x4218 00A8	0x0218 00A8
EVE_WBUF_EDADDR_BO	R	32	0x8 00AC	0x4208 00AC	0x0208 00AC	0x4218 00AC	0x0218 00AC
EVE_IBUF_ED_CTL	RW	32	0x8 00B0	0x4208 00B0	0x0208 00B0	0x4218 00B0	0x0218 00B0
EVE_IBUF_ED_STAT	RW	32	0x8 00B4	0x4208 00B4	0x0208 00B4	0x4218 00B4	0x0218 00B4
EVE_IBUF_EDADDR	R	32	0x8 00B8	0x4208 00B8	0x0208 00B8	0x4218 00B8	0x0218 00B8

Table 8-34. EVE Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1 Physical Address L3 Interconnect	EVE1_DSP Physical Address DSP private	EVE2 Physical Address L3 Interconnect	EVE2_DSP Physical Address DSP private
EVE_IBUF_EDADDR_BO	R	32	0x8 00BC	0x4208 00BC	0x0208 00BC	0x4218 00BC	0x0218 00BC
EVE_ED_ARP32_DISC_EN	RW	32	0x8 00F8	0x4208 00F8	0x0208 00F8	0x4218 00F8	0x0218 00F8
EVE_ED_OCPI_DISC_EN	RW	32	0x8 00FC	0x4208 00FC	0x0208 00FC	0x4218 00FC	0x0218 00FC
EVE_MSW_ERR_IRQSTATUS_RAW	RW	32	0x8 0110	0x4208 0110	0x0208 0110	0x4218 0110	0x0218 0110
EVE_MSW_ERR_IRQSTATUS	RW	32	0x8 0114	0x4208 0114	0x0208 0114	0x4218 0114	0x0218 0114
EVE_MSW_ERR_IRQENABLE_SET	RW	32	0x8 0118	0x4208 0118	0x0208 0118	0x4218 0118	0x0218 0118
EVE_MSW_ERR_IRQENABLE_CLR	RW	32	0x8 011C	0x4208 011C	0x0208 011C	0x4218 011C	0x0218 011C
EVE_ED_LCL_IRQSTATUS_RAW	RW	32	0x8 0120	0x4208 0120	0x0208 0120	0x4218 0120	0x0218 0120
EVE_ED_LCL_IRQSTATUS	RW	32	0x8 0124	0x4208 0124	0x0208 0124	0x4218 0124	0x0218 0124
EVE_ED_LCL_IRQENABLE_SET	RW	32	0x8 0128	0x4208 0128	0x0208 0128	0x4218 0128	0x0218 0128
EVE_ED_LCL_IRQENABLE_CLR	RW	32	0x8 012C	0x4208 012C	0x0208 012C	0x4218 012C	0x0218 012C
ARP32_NMI_IRQSTATUS_RAW	RW	32	0x8 0200	0x4208 0200	0x0208 0200	0x4218 0200	0x0218 0200
ARP32_NMI_IRQSTATUS	W	32	0x8 0204	0x4208 0204	0x0208 0204	0x4218 0204	0x0218 0204
ARP32_NMI_IRQENABLE_SET	RW	32	0x8 0208	0x4208 0208	0x0208 0208	0x4218 0208	0x0218 0208
ARP32_NMI_IRQENABLE_CLR	W	32	0x8 020C	0x4208 020C	0x0208 020C	0x4218 020C	0x0218 020C
ARP32_INTn_IRQSTATUS_RAW (1)	RW	32	0x8 01D0 + (0x10*n)	0x4208 01D0 + (0x10*n)	0x0208 01D0 + (0x10*n)	0x4218 01D0 + (0x10*n)	0x0218 01D0 + (0x10*n)
ARP32_INTn_IRQSTATUS (1)	W	32	0x8 01D4 + (0x10*n)	0x4208 01D4 + (0x10*n)	0x0208 01D4 + (0x10*n)	0x4218 01D4 + (0x10*n)	0x0218 01D4 + (0x10*n)
ARP32_INTn_IRQENABLE_SET (1)	RW	32	0x8 01D8 + (0x10*n)	0x4208 01D8 + (0x10*n)	0x0208 01D8 + (0x10*n)	0x4218 01D8 + (0x10*n)	0x0218 01D8 + (0x10*n)
ARP32_INTn_IRQENABLE_CLR (1)	W	32	0x8 01DC + (0x10*n)	0x4208 01DC + (0x10*n)	0x0208 01DC + (0x10*n)	0x4218 01DC + (0x10*n)	0x0218 01DC + (0x10*n)
ARP32_IRQWAKEEN	RW	32	0x8 02FC	0x4208 02FC	0x0208 02FC	0x4218 02FC	0x0218 02FC
MMR_LOCKi (4)	RW	32	0x8 0300 + (0x4*i)	0x4208 0300 + (0x4*i)	0x0208 0300 + (0x4*i)	0x4218 0300 + (0x4*i)	0x0218 0300 + (0x4*i)
MISR_CTL	RW	32	0x8 0400	0x4208 0400	0x0208 0400	0x4218 0400	0x0218 0400
MISR_CLEAR	RW	32	0x8 0404	0x4208 0404	0x0208 0404	0x4218 0404	0x0218 0404
MISR0_A	RW	32	0x8 0410	0x4208 0410	0x0208 0410	0x4218 0410	0x0218 0410
MISR0_D	RW	32	0x8 0414	0x4208 0414	0x0208 0414	0x4218 0414	0x0218 0414
MISR1_A	RW	32	0x8 0418	0x4208 0418	0x0208 0418	0x4218 0418	0x0218 0418
MISR1_D	RW	32	0x8 041C	0x4208 041C	0x0208 041C	0x4218 041C	0x0218 041C
MISR2_Dk (3)	RW	32	0x8 0420 + (0x4*k)	0x4208 0420 + (0x4*k)	0x0208 0420 + (0x4*k)	0x4218 0420 + (0x4*k)	0x0218 0420 + (0x4*k)
EVE_IRQ_EOI	RW	32	0x8 0500	0x4208 0500	0x0208 0500	0x4218 0500	0x0218 0500
EVE_ED_OUT_IRQSTATUS_RAW	RW	32	0x8 0510	0x4208 0510	0x0208 0510	0x4218 0510	0x0218 0510
EVE_ED_OUT_IRQSTATUS	RW	32	0x8 0514	0x4208 0514	0x0208 0514	0x4218 0514	0x0218 0514

Table 8-34. EVE Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1 Physical Address L3 Interconnect	EVE1_DSP Physical Address DSP private	EVE2 Physical Address L3 Interconnect	EVE2_DSP Physical Address DSP private
EVE_ED_OUT_IRQENABLE_SET	RW	32	0x8 0518	0x4208 0518	0x0208 0518	0x4218 0518	0x0218 0518
EVE_ED_OUT_IRQENABLE_CLR	RW	32	0x8 051C	0x4208 051C	0x0208 051C	0x4218 051C	0x0218 051C
EVE_INTk_OUT_IRQSTATUS_RAW (3)	RW	32	0x8 0520 + (0x10*k)	0x4208 0520 + (0x10*k)	0x0208 0520 + (0x10*k)	0x4218 0520 + (0x10*k)	0x0218 0520 + (0x10*k)
EVE_INTk_OUT_IRQSTATUS (3)	RW	32	0x8 0524 + (0x10*k)	0x4208 0524 + (0x10*k)	0x0208 0524 + (0x10*k)	0x4218 0524 + (0x10*k)	0x0218 0524 + (0x10*k)
EVE_INTk_OUT_IRQENABLE_SET (3)	RW	32	0x8 0528 + (0x10*k)	0x4208 0528 + (0x10*k)	0x0208 0528 + (0x10*k)	0x4218 0528 + (0x10*k)	0x0218 0528 + (0x10*k)
EVE_INTk_OUT_IRQENABLE_CLR (3)	RW	32	0x8 052C + (0x10*k)	0x4208 052C + (0x10*k)	0x0208 052C + (0x10*k)	0x4218 052C + (0x10*k)	0x0218 052C + (0x10*k)
ARP32_INTj_IRQSTATUS_RAW (2)	RW	32	0x8 0580 + (0x10*j)	0x4208 0580 + (0x10*j)	0x0208 0580 + (0x10*j)	0x4218 0580 + (0x10*j)	0x0218 0580 + (0x10*j)
ARP32_INTj_IRQSTATUS (2)	RW	32	0x8 0584 + (0x10*j)	0x4208 0584 + (0x10*j)	0x0208 0584 + (0x10*j)	0x4218 0584 + (0x10*j)	0x0218 0584 + (0x10*j)
ARP32_INTj_IRQENABLE_SET (2)	RW	32	0x8 0588 + (0x10*j)	0x4208 0588 + (0x10*j)	0x0208 0588 + (0x10*j)	0x4218 0588 + (0x10*j)	0x0218 0588 + (0x10*j)
ARP32_INTj_IRQENABLE_CLR (2)	RW	32	0x8 058C + (0x10*j)	0x4208 058C + (0x10*j)	0x0208 058C + (0x10*j)	0x4218 058C + (0x10*j)	0x0218 058C + (0x10*j)
ARP32_INT14_IRQSTATUS_RAW	RW	32	0x8 0680	0x4208 0680	0x0208 0680	0x4218 0680	0x0218 0680
ARP32_INT14_IRQSTATUS	RW	32	0x8 0684	0x4208 0684	0x0208 0684	0x4218 0684	0x0218 0684
ARP32_INT14_IRQENABLE_SET	RW	32	0x8 0688	0x4208 0688	0x0208 0688	0x4218 0688	0x0218 0688
ARP32_INT14_IRQENABLE_CLR	RW	32	0x8 068C	0x4208 068C	0x0208 068C	0x4218 068C	0x0218 068C
ARP32_INT15_IRQSTATUS_RAW	RW	32	0x8 0690	0x4208 0690	0x0208 0690	0x4218 0690	0x0218 0690
ARP32_INT15_IRQSTATUS	RW	32	0x8 0694	0x4208 0694	0x0208 0694	0x4218 0694	0x0218 0694
ARP32_INT15_IRQENABLE_SET	RW	32	0x8 0698	0x4208 0698	0x0208 0698	0x4218 0698	0x0218 0698
ARP32_INT15_IRQENABLE_CLR	RW	32	0x8 069C	0x4208 069C	0x0208 069C	0x4218 069C	0x0218 069C
EVE_GPOUTm (5)	RW	32	0x8 0700 + (0x10*m)	0x4208 0700 + (0x10*m)	0x0208 0700 + (0x10*m)	0x4218 0700 + (0x10*m)	0x0218 0700 + (0x10*m)
EVE_GPOUTm_SET (5)	RW	32	0x8 0704 + (0x10*m)	0x4208 0704 + (0x10*m)	0x0208 0704 + (0x10*m)	0x4218 0704 + (0x10*m)	0x0218 0704 + (0x10*m)
EVE_GPOUTm_CLR (5)	RW	32	0x8 0708 + (0x10*m)	0x4208 0708 + (0x10*m)	0x0208 0708 + (0x10*m)	0x4218 0708 + (0x10*m)	0x0218 0708 + (0x10*m)
EVE_GPOUTm_PULSE (5)	RW	32	0x8 070C + (0x10*m)	0x4208 070C + (0x10*m)	0x0208 070C + (0x10*m)	0x4218 070C + (0x10*m)	0x0218 070C + (0x10*m)
EVE_GPIN0	R	32	0x8 0740	0x4208 0740	0x0208 0740	0x4218 0740	0x0218 0740
EVE_GPIN1	R	32	0x8 0744	0x4208 0744	0x0208 0744	0x4218 0744	0x0218 0744
EVE_CME_DONE_GPOUT	RW	32	0x8 0780	0x4208 0780	0x0208 0780	0x4218 0780	0x0218 0780
EVE_CME_DONE_GPOUT_SET	RW	32	0x8 0784	0x4208 0784	0x0208 0784	0x4218 0784	0x0218 0784
EVE_CME_DONE_GPOUT_CLR	RW	32	0x8 0788	0x4208 0788	0x0208 0788	0x4218 0788	0x0218 0788

Table 8-34. EVE Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1 Physical Address L3 Interconnect	EVE1_DSP Physical Address DSP private	EVE2 Physical Address L3 Interconnect	EVE2_DSP Physical Address DSP private
EVE_CME_DONE_GPOUT_PULSE	RW	32	0x8 078C	0x4208 078C	0x0208 078C	0x4218 078C	0x0218 078C
EVE_CME_DONE_SEL	RW	32	0x8 0790	0x4208 0790	0x0208 0790	0x4218 0790	0x0218 0790
EVE_CME_DONE_EN	RW	32	0x8 0794	0x4208 0794	0x0208 0794	0x4218 0794	0x0218 0794
EVE_PM_STAT0	R	32	0x8 0FE0	0x4208 0FE0	0x0208 0FE0	0x4218 0FE0	0x0218 0FE0
EVE_PM_STAT1	R	32	0x8 0FE4	0x4208 0FE4	0x0208 0FE4	0x4218 0FE4	0x0218 0FE4
EVE_DBGOUT	RW	32	0x8 0FE8	0x4208 0FE8	0x0208 0FE8	0x4218 0FE8	0x0218 0FE8
EVE_RSVD0	RW	32	0x8 0FF4	0x4208 0FF4	0x0208 0FF4	0x4218 0FF4	0x0218 0FF4
EVE_RSVD1	RW	32	0x8 0FF8	0x4208 0FF8	0x0208 0FF8	0x4218 0FF8	0x0218 0FF8
EVE_TEST	RW	32	0x8 0FFC	0x4208 0FFC	0x0208 0FFC	0x4218 0FFC	0x0218 0FFC

(1) n = 4 to 7 for EVE1
n = 4 to 7 for EVE1_DSP
n = 4 to 7 for EVE2
n = 4 to 7 for EVE2_DSP

(2) j = 8 to 13 for EVE1
j = 8 to 13 for EVE1_DSP
j = 8 to 13 for EVE2
j = 8 to 13 for EVE2_DSP

(3) k = 0 to 3 for EVE1
k = 0 to 3 for EVE1_DSP
k = 0 to 3 for EVE2
k = 0 to 3 for EVE2_DSP

(4) i = 0 to 9 for EVE1
i = 0 to 9 for EVE1_DSP
i = 0 to 9 for EVE2
i = 0 to 9 for EVE2_DSP

(5) m = 0 to 1 for EVE1
m = 0 to 1 for EVE1_DSP
m = 0 to 1 for EVE2
m = 0 to 1 for EVE2_DSP

8.1.5.2.2 EVE Register Description

Table 8-35 through Table 8-233 describe the individual register bits.

Table 8-35. EVE_REVISION

Address Offset	0x0008 0000		
Physical Address	0x4208 0000 0x0208 0000 0x4218 0000 0x0218 0000	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision	R	0x0

Table 8-36. Register Call Summary for Register EVE_REVISION

Embedded Vision Engine (EVE) Subsystem

- Lock/Unlock feature: [0]
- EVE Register Summary: [1]

Table 8-37. EVE_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x4208 0004 0x0208 0004 0x4218 0004 0x0218 0004	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–4	3–0
INFO	EVENUM

Bits	Field Name	Description	Type	Reset
31: 4	INFO	0x0: No configurable options in EVE	R	0x00
3:0	EVENUM	EVE instance number set by eve_num inputs. In a multi-EVE system must be set to unique/incrementing values for each EVE.	R	0x0

Table 8-38. Register Call Summary for Register EVE_HWINFO

Embedded Vision Engine (EVE) Subsystem

- Lock/Unlock feature: [0]
- EVE Register Summary: [1]

Table 8-39. EVE_SYSCONFIG

Address Offset	0x0000 0008		
Physical Address	0x4208 0008 0x0208 0008 0x4218 0008 0x0218 0008	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–6	5–4	3–2	1	0
RESERVED	STANDBYMODE	IDLEMODE	FREEEMU	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved. Read returns 0s	RW	0x0000

Bits	Field Name	Description	Type	Reset
5:4	STANDBYMODE	00: Force-Standby mode: This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode / the SAF asserts with minimal hardware condition the status saying :I am in standby:. It is the responsibility of the software to ensure that the SAF is in a correct quiet state before programming this mode. Additionally when in this mode / the SAF is not allowed to generate wakeup reques . 01: No-Standby: This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode / the SAF never asserts the status declaring that the system is in standby. 10: Smart-Standby: default. EVE generates the standby status based upon all hardware internal status / namely after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode / the SAF is not allowed to generate wakeup request. 11: Smart-Standby-Wkup: Same as Smart-Standby. (EVE generates the standby status based upon all hardware internal status / namely after having performed all hardware operations necessary to be in a correct quiet state). . Additionally when in this mode / the SAF is allowed to generate wakeup request	RW	0x0
3:2	IDLEMODE	00: Force-idle: This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode the IAF acknowledges a request to go idle from the power manager with minimal hardware condition. It is the responsibility of the software to ensure that the IAF are in a correct quiet state before requesting a force-idle transition. Additionaly when in this mode the IAF is not allowed to generate any wakeup request. 01: No-idle: When in this mode the IAF disregards any request to go idle from the power manager. 10: Smart-idle: default mode. default. When in this mode / the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode / the IAF is not allowed to generate any wakeup reques 11: SmartIdleWkup : When in this mode / the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary for the IAF to be in a correct quiet state. Additionally when in this mode / the IAF is allowed to generate wakeup request	RW	0x0
1	FREEEMU	Resered. Note that SCTM has free control bit to define the timer operation during ARP32 debug halt mode	R	0x0
0	SOFTRESET	Reserved	R	0x0

Table 8-40. Register Call Summary for Register EVE_SYSCONFIG

Embedded Vision Engine (EVE) Subsystem

- Extended Duration Sleep: [0] [1]
- Lock/Unlock feature: [2]
- EVE Register Summary: [3]

Table 8-41. EVE_STAT

Address Offset	0x0008 000C		
Physical Address	0x4208 000C 0x0208 000C 0x4218 000C 0x0218 000C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP

Table 8-41. EVE_STAT (continued)

Description	
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										OCPI_DISC_STAT		RESERVED		ARP32_DISC_STATUS		RESERVED							INT_OUT_STAT	ARP32_INTC_STAT	RESERVED	TC1_STAT	TC0_STAT	RESERVED	PC_STAT	VCOP_STAT	ARP32_STAT

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0000
21:20	OCPI_DISC_STAT	OCP Initiator(s) Disconnect status2: 2: One or both initiators are active, no request to disconnect is pending 1: One or both initiators are attempting to disconnect 0: EVEs OCP initiators are disconnected	R	0x0000
19:18	RESERVED		R	0
17:16	ARP32_DISC_STATUS	ARP32 Program/Data Bus Disconnect Status 2: 2: ARP32 program and data busses are active, no request to disconnect is pending 1: ARP32 program and data buses are attempting to disconnect 0: ARP32 program and data buses are disconnected	R	0
15:9	RESERVED		R	0
8	INT_OUT_STAT	Interrupt Output status: 0: No edabled interrupts pending 1: Active (at least one enabled interrupt source is pending in the output reducer)	R	0
7	ARP32_INTC_STAT	Interrupt Controller Status: 0: No enabled interrupts pending 1: Active (at least one enabled interrupt source is pending in the ARP32 interrupt controller)	R	0
6	RESERVED		R	0
5	TC1_STAT	Transfer Controller1 Status: 0: Idle 1: Active	R	0
4	TC0_STAT	Transfer Controller0 Status: 0: Idle 1: Active	R	0
3	RESERVED		R	0
2	PC_STAT	Program Cache Status: 0: Idle 1: Active (Program cache is either performing prefetch/ preload/invalidation, or is servicing a CPU program fetch request (hit or miss)).	R	0

Bits	Field Name	Description	Type	Reset
1	VCOP_STAT	VCOP Status : 0: Idle 1: Active (Program execution in progress. Based on inverse of vcop_done. Does not account for activity on VCOP OCP debug interface)	R	0
0	ARP32_STAT	Program Cache Status: 0: Idle 1: Active (based on inverse of arp32_stanby).	R	0

Table 8-42. Register Call Summary for Register EVE_STAT

Embedded Vision Engine (EVE) Subsystem

- ARP32 Disconnect: [0]
- OCP Initiators Disconnect: [1]
- Lock/Unlock feature: [2]
- EVE Register Summary: [3]

Table 8-43. EVE_DISC_CONFIG

Address Offset	0x0008 0010		
Physical Address	0x4208 0010 0x0208 0010 0x4218 0010 0x0218 0010	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Color 0 noise threshold		
Type	RW		

31–5	4	3–1	0
RESERVED	OCPI_DISC	RESERVED	ARP32_DISC

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	OCPI_DISC	OCP Initiator Disconnect request: Write 1: request for OCP initiator to disconnect and mask write byte enable signals. Writing 0 has no effect Read 0: Disconnect not in progress of has completed. Read 1: Disconnect request in progress.	RW	0x0
3:1	RESERVED		R	0x0
0	ARP32_DISC	ARP32 Initiator Disconnect request. Write 1: Request for ARP32 program and data buses to disconnect and mask write byte enable signals. Writing 0 has no effect. Read 1: disconnect request in progress. Read 0: disconnect not in progress or has completed.	RW	0x0

Table 8-44. Register Call Summary for Register EVE_DISC_CONFIG

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- ARP32 Disconnect: [1]
- OCP Initiators Disconnect: [2]
- Lock/Unlock feature: [3]
- EVE Register Summary: [4]

Table 8-45. EVE_BUS_CONFIG

Address Offset	0x0008 0014		
Physical Address	0x4208 0014 0x0208 0014 0x4218 0014 0x0218 0014	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Color 0 noise threshold		
Type	RW		

31:14	13:12	11:10	9:8	7:5	4	3:0
RESERVED	TC1_DBS	RESERVED	TC0_DBS	RESERVED	DBP_ENABLE	MAX_IN_FLIGHT

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	TC1_DBS	TC1 default burst size. 00: 16 byte, 01: 32 byte, 10: 64 byte, 11: 128 byte (recommended)	RW	0x3
11:10	RESERVED		R	0x0
9:8	TC0_DBS	TC0 default burst size. 00: 16 byte, 01: 32 byte, 10: 64 byte, 11: 128 byte (recommended)	RW	0x3
7:5	RESERVED		R	
4	DBP_ENABLE	Program Cache Demand Based Prefetch enable: 0: DBP disabled 1: DBP enabled	RW	0x0
3:0	MAX_IN_FLIGHT	Defines maximum number of OCP requests in flight. Can be reduced to limit the peak bandwidth for software direct preload, which in turn may provide advantage to other EVE level (e.g. EDMA) or system-level initiators. 0: Reserved. 1: 1 request in flight allowed. 2: 2 requests in flight allowed.. F: 15 requests in flight allowed.	RW	0x4

Table 8-46. Register Call Summary for Register EVE_BUS_CONFIG

Embedded Vision Engine (EVE) Subsystem

- Demand Based Prefetch: [0] [1]
- Lock/Unlock feature: [3]
- EVE Register Summary: [4]

Table 8-47. EVE_VCOP_HALT_CONFIG

Address Offset	0x0008 0018		
Physical Address	0x4208 0018 0x0208 0018 0x4218 0018 0x0218 0018	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			

Table 8-47. EVE_VCOP_HALT_CONFIG (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													FORCE_ABORT	MSW_EN	ED_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	FORCE_ABORT	VCOP Force Abort Write: Read always returns 0s Write 0 has no effect. Write 1: issues force_abort command to VCOP (through pulse on vcop force abort input	RW	0x0000
1	MSW_EN	VCOP Memory Seitch Error Halt Enable: 0: Disabled 1: Enabled. VCOP halts on VCOP initiated memory swithc error	RW	0x00
0	ED_EN	VCOP Parity Error Detect Halt Enable: 0: Disabled 1: Enabled. VCOP halts on VCOP initiated parity error.	RW	0x000

Table 8-48. Register Call Summary for Register EVE_VCOP_HALT_CONFIG

Embedded Vision Engine (EVE) Subsystem

- VCOP System Error Halt Conditions: [0] [1] [2]
- Lock/Unlock feature: [3]
- EVE Register Summary: [4]

Table 8-49. EVE_MMU_CONFIG

Address Offset	0x0008 001C		
Physical Address	0x4208 001C 0x0208 001C 0x4218 001C 0x0218 001C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			MMU1_ABORT	RESERVED			MMU0_ABORT	RESERVED			MMU1_EN	RESERVED			MMU0_EN

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00
12	MMU1_ABORT	Causes the MMU to abort the current operation in case of lockup	RW	0x000
11:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MMU0_ABORT	Causes the MMU to abort the current operation in case of lockup	RW	0x000
7:5	RESERVED		R	0x0
4	MMU1_EN	Clearing this bit disables MMU table lookup and causes accesses to use the non-translated address. This bit defaults to enabled but an identical bit within an MMU configuration register defaults to disabled and must be set after the page tables are programmed for MMU operation	RW	0x0
3:1	RESERVED		R	0x0
0	MMU0_EN	Clearing this bit disables MMU table lookup and causes accesses to use the non-translated address. This bit defaults to enabled but an identical bit within an MMU configuration register defaults to disabled and must be set after the page tables are programmed for MMU operation	RW	0x0

Table 8-50. Register Call Summary for Register EVE_MMU_CONFIG

Embedded Vision Engine (EVE) Subsystem

- Lock/Unlock feature: [0]
- EVE Register Summary: [1]

Table 8-51. EVE_MEMMAP

Address Offset	0x0008 0020		
Physical Address	0x4208 0020 0x0208 0020 0x4218 0020 0x0218 0020	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–5	4	3–1	0
RESERVED	LCL_EDMA_ALIAS	RESERVED	VCOP_ALIAS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	LCL_EDMA_ALIAS	0: Local EDMA views full memory map 1: VCOP vies aliased memory map. In this mode VCOP views IBUFLA and IBUFLB at the same address and views IBUFHA and IBUFHB at the same address. In this mode only one of IBUFLA or IBUFLB and IBUFHA or IBUFHB can be owned by the system.Refer to Error: Reference source not found for full truth table. Software must poll for updated value to ensure mode change has taken effect.	RW	0x000
3:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	VCOP_ALIAS	0: VCOP views full memory map 1: VCOP views Aliased memory map. In this mode / VCOP views IBUFLA and IBUFLB at the same address / and views IBUFHA and IBUFHB at the same address. In this mode / only one of IBUFLA or IBUFLB can be :owned: by VCOP; and only one of IBUFHA or IBUFHB can be :owned: by VCOP. Refer to Error: Reference source not found for full truth table. Software must poll for updated value to ensure mode change has taken effect	RW	0x0000

Table 8-52. Register Call Summary for Register EVE_MEMMAP

Embedded Vision Engine (EVE) Subsystem

- Image Buffers - IBUFLA, IBUFLB, IBUFHA, IBUFHB: [0]
- Lock/Unlock feature: [1]
- EVE Memory Map: [2] [3]
- VCOP and Local EDMA: IBUF Memory Map Aliasing: [4] [5] [6] [7] [8] [9]
- ARP32 Write Model - Avoiding Race Conditions: [10] [11]
- EVE Register Summary: [12]

Table 8-53. EVE_MSW_CTL

Address Offset	0x0008 0024		
Physical Address	0x4208 0024 0x0208 0024 0x4218 0024 0x0218 0024	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Memory switch control register		
Type	RW		

31:17	16	15:13	12	11:9	8	7:5	4	3:1	0
RESERVED	WBUF	RESERVED	IBUFHB	RESERVED	IBUFLB	RESERVED	IBUFHA	RESERVED	IBUFLA

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	WBUF	Working buffer onwership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20). 0: System owned 1: VCOP owned	RW	0x0000
15:13	RESERVED		R	0x00
12	IBUFHB	Image buffer high B ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x000
11:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	IBUFLB	Image buffer low B ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x0
7:5	RESERVED		R	0x0
4	IBUFHA	Image buffer high A ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x0
3:1	RESERVED		R	0x0
0	IBUFLA	Image buffer low A ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x0

Table 8-54. Register Call Summary for Register EVE_MSW_CTL

Embedded Vision Engine (EVE) Subsystem

- VCOP Working Buffer (WBUF): [0] [1] [2]
- Image Buffers - IBUFLA, IBUFLB, IBUFHA, IBUFHB: [3] [4] [5] [6] [7] [8]
- Lock/Unlock feature: [9]
- VCOP and Local EDMA: IBUF Memory Map Aliasing: [10] [11]
- ARP32 Write Model - Avoiding Race Conditions: [12] [13]
- EVE Register Summary: [14]

Table 8-55. EVE_MSW_ERR

Address Offset	0x0008 0028		
Physical Address	0x4208 0028 0x0208 0028 0x4218 0028 0x0218 0028	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Memory Switch Error register		
Type	RW		

31:25	24:16	15:4	3	2	1	0
RESERVED	CONNID	RESERVED	SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
24:16	CONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. Undefined when no error status bits are set. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to ERR bit field	R	0x000
15:4	RESERVED		R	0x0
3	SYSERR	0 - System initiated buffer ownership error not recorded 1 - System initiated buffer ownership error detected/recorded. Write 1 to clear.	WO	0x000
2	DMAERR	0 : EDMA initiated buffer ownership error not recorded 1 : EDMA buffer ownership error detected/recorded Write 1 to clear.	WO	0x0
1	VERR	0 : VCOP initiated buffer ownership error not recorded 1 : VCOP initiated buffer ownership error detected/recorded Write 1 to clear.	WO	0x0
0	ARP32ERR	0 : ARP32 initiated buffer ownership error not recorded 1 : ARP32 initiated buffer ownership error detected/recorded Write 1 to clear.	WO	0x0

Table 8-56. Register Call Summary for Register EVE_MSW_ERR

Embedded Vision Engine (EVE) Subsystem

- EVE Connection ID (ConnID) Mapping: [0]
- VCOP Working Buffer (WBUF): [1]
- Image Buffers - IBUFLA, IBUFLB, IBUFHA, IBUFHB: [2]
- Memory Switch Error Registers: [3]
- VCOP System Error Halt Conditions: [4]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [5]
- Lock/Unlock feature: [6]
- EVE Register Summary: [7]

Table 8-57. EVE_MSW_ERRADDR

Address Offset	0x0008 002C		
Physical Address	0x4208 002C 0x0208 002C 0x4218 002C 0x0218 002C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Memory switch error address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of memory switch error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address. Value is undefined when no ERR bits set.	R	0x0000 0000

Table 8-58. Register Call Summary for Register EVE_MSW_ERRADDR

Embedded Vision Engine (EVE) Subsystem
- VCOP Working Buffer (WBUF): [0]
- Image Buffers - IBUFLA, IBUFLB, IBUFHA, IBUFHB: [1]
- Memory Switch Error Registers: [2]
- Lock/Unlock feature: [3]
- EVE Register Summary: [4]

Table 8-59. EVE_PC_INV

Address Offset	0x0008 0040		
Physical Address	0x4208 0040 0x0208 0040 0x4218 0040 0x0218 0040	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Invalidate all register		
Type	RW		

31–1	0
RESERVED	I

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	I	Invalidate all: Write 0: no effet. Write 1: initiate invalidate all command Read: 0 : Invalidate operation complete / or not in progress 1 : Invalidate operation still in progress	RW	0x0

Table 8-60. Register Call Summary for Register EVE_PC_INV

Embedded Vision Engine (EVE) Subsystem
- User Coherence Operation: [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-61. EVE_PC_IBAR

Address Offset	0x0008 0050		
Physical Address	0x4208 0050 0x0208 0050 0x4218 0050 0x0218 0050	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Invalidate Base Address register		
Type	RW		

31–0
ADDR

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Invalidate Base Address register. Defines system byte address base to be invalidated from L1P. The entire range to be invalidated is (loosely speaking) from the base address to the base address + byte count. The actual range is inclusive of cache line (32-B aligned) addresses containing the start address and end address	RW	0x0000 0000

Table 8-62. Register Call Summary for Register EVE_PC_IBAR

Embedded Vision Engine (EVE) Subsystem

- User Coherence Operation: [0] [1] [2] [3]
- Lock/Unlock feature: [4]
- EVE Register Summary: [5]

Table 8-63. EVE_PC_IBC

Address Offset	0x0008 0054		
Physical Address	0x4208 0054 0x0208 0054 0x4218 0054 0x0218 0054	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Invalidate byte count register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	BC	Invalidate Byte Count register. Defines number of bytes relative to Invalidate Base Address (IBAR.ADDR) to be invalidated from L1P. Maximum of 32k (0x8000). Reads return 0x0 when invalidate range operation is complete.	RW	0x0000

Table 8-64. Register Call Summary for Register EVE_PC_IBC

Embedded Vision Engine (EVE) Subsystem

- User Coherence Operation: [0] [1] [2]
- Lock/Unlock feature: [3]
- EVE Register Summary: [4]

Table 8-65. EVE_PC_ISAR

Address Offset	0x0008 0058		
Physical Address	0x4208 0058 0x0208 0058 0x4218 0058 0x0218 0058	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Invalidate single address register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Invalidate Single Address register. Defines system byte address (1 line only) to be invalidated from L1P. Reads return 0x0 when the invalidate operation is complete.	RW	0x0000 0000

Table 8-66. Register Call Summary for Register EVE_PC_ISAR

Embedded Vision Engine (EVE) Subsystem

- User Coherence Operation: [0] [1] [2] [3]
- Lock/Unlock feature: [4]
- EVE Register Summary: [5]
- EVE Register Description: [7]

Table 8-67. EVE_PC_ISAR_DONE

Address Offset	0x0008 005C		
Physical Address	0x4208 005C 0x0208 005C 0x4218 005C 0x0218 005C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Invalidate single address done register		
Type	R		

31–1	0
RESERVED	DONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	DONE	Reads return 0x1 when the invalidate operation is complete. Cleared on the next write to the EVE_PC_ISAR register	R	0x0

Table 8-68. Register Call Summary for Register EVE_PC_ISAR_DONE

Embedded Vision Engine (EVE) Subsystem

- Lock/Unlock feature: [0]
- EVE Register Summary: [1]

Table 8-69. EVE_PC_PBAR

Address Offset	0x0008 0060		
Physical Address	0x4208 0060 0x0208 0060 0x4218 0060 0x0218 0060	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Program cache preload base address register		
Type	RW		

31–0
ADDR

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Preload Base Address register. Defines system byte address base to be Preloaded into L1P. The entire range to be Preloaded is (loosely speaking) from the base address to the base address + byte count. The actual range is inclusive of cache line (32-B aligned) addresses containing the start address and end address	RW	0x0000 0000

Table 8-70. Register Call Summary for Register EVE_PC_PBAR

Embedded Vision Engine (EVE) Subsystem

- Software Direct Preload (SDP): [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-71. EVE_PC_PBC

Address Offset	0x0008 0064		
Physical Address	0x4208 0064 0x0208 0064 0x4218 0064 0x0218 0064	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31:16	15:0
RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BC	Preload Byte Count register. Defines number of bytes relative to Preload Base Address (PBAR.ADDR) to be Preloaded into L1P. Maximum of 32k (0x8000). Reads return 0x0 when Preload range operation is complete	RW	0x0000

Table 8-72. Register Call Summary for Register EVE_PC_PBC

Embedded Vision Engine (EVE) Subsystem

- Software Direct Preload (SDP): [0] [1]
- Lock/Unlock feature: [2]
- EVE Register Summary: [3]

Table 8-73. EVE_PMEM_ED_CTL

Address Offset	0x0008 0080		
Physical Address	0x4208 0080 0x0208 0080 0x4218 0080 0x0218 0080	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Program Memory Error Detection Control register		
Type	RW		

31:2	1	0
RESERVED	INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reds from meory return inverse of parity bit. Write 0 to clear, write 1 to set. Must be set when EN is set	RW	0x0

Bits	Field Name	Description	Type	Reset
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 8-74. Register Call Summary for Register EVE_PMEM_ED_CTL

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-75. EVE_PMEM_ED_STAT

Address Offset	0x0008 0084		
Physical Address	0x4208 0084 0x0208 0084 0x4218 0084 0x0218 0084	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Error detection status register		
Type	RW		

31:25	24:16	15:4	3	2	1	0
RESERVED	SYSCONNID	RESERVED	SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 8-76. Register Call Summary for Register EVE_PMEM_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [1]
- EVE Register Summary: [2]

Table 8-77. EVE_PMEM_EDADDR

Address Offset	0x0008 0088		
Physical Address	0x4208 0088 0x0208 0088 0x4218 0088 0x0218 0088	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Program memory error detection address		
Type	R		

31–0
ADDR

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 8-78. Register Call Summary for Register EVE_PMEM_EDADDR

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1] [2]
- EVE Register Summary: [3]

Table 8-79. EVE_DMEM_ED_CTL

Address Offset	0x0008 0090		
Physical Address	0x4208 0090 0x0208 0090 0x4218 0090 0x0218 0090	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	DMEM error detection control		
Type	RW		

31–2	1	0
RESERVED	INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reads from meory return inverse of parity bit. Write 0 to clear, write 1 to set. Must be set when EN is set	RW	0x0
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 8-80. Register Call Summary for Register EVE_DMEM_ED_CTL

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-81. EVE_DMEM_ED_STAT

Address Offset	0x0008 0094		
Physical Address	0x4208 0094 0x0208 0094 0x4218 0094 0x0218 0094	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	DMEM error detection status register		
Type	RW		

31:25	24:16	15:4	3	2	1	0
RESERVED	SYSCONNID	RESERVED	SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 8-82. Register Call Summary for Register EVE_DMEM_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [1] [2] [3]
- EVE Register Summary: [4]

Table 8-83. EVE_DMEM_EDADDR

Address Offset	0x0000 0098		
Physical Address	0x4208 0098 0x0208 0098 0x4218 0098 0x0218 0098	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	DMEM error detection address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ADDR

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 8-84. Register Call Summary for Register EVE_DMEM_EDADDR

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-85. EVE_DMEM_EDADDR_BO

Address Offset	0x0000 009C		
Physical Address	0x4208 009C 0x0208 009C 0x4218 009C 0x0218 009C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	DMEM error detection address byte offset register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	BO	Address byte offset for parity error. Indicates that an error has occured in byte offset #n. 0: No error occuredin byte offset #n; 1: Error occured in byte offset #n. Write to clear any of MEM.SYSERR/DMARR/ARP32/or VERR.	R	0x0

Table 8-86. Register Call Summary for Register EVE_DMEM_EDADDR_BO

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-87. EVE_WBUF_ED_CTL

Address Offset	0x0008 00A0		
Physical Address	0x4208 00A0 0x0208 00A0 0x4218 00A0 0x0218 00A0	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	WBUF error detection control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reads from meory return inverse of parity bit. Write 0 to clea Write 1 to set. Must be set when EN is set	RW	0x0
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 8-88. Register Call Summary for Register EVE_WBUF_ED_CTL

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-89. EVE_WBUF_ED_STAT

Address Offset	0x0008 00A4		
Physical Address	0x4208 00A4 0x0208 00A4 0x4218 00A4 0x0218 00A4	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	WBUF error detection status register		
Type	RW		

31:25	24:16	15:4	3	2	1	0
RESERVED	SYSCONNID	RESERVED	SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 8-90. Register Call Summary for Register EVE_WBUF_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- VCOP System Error Halt Conditions: [1]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [2] [3] [4] [5]
- EVE Register Summary: [6]

Table 8-91. EVE_WBUF_EDADDR

Address Offset	0x0008 00A8		
Physical Address	0x4208 00A8 0x0208 00A8 0x4218 00A8 0x0218 00A8	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	WBUF error detection address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 8-92. Register Call Summary for Register EVE_WBUF_EDADDR

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-93. EVE_WBUF_EDADDR_BO

Address Offset	0x0008 00AC		
Physical Address	0x4208 00AC 0x0208 00AC 0x4218 00AC 0x0218 00AC	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	WBUF error detection address byte offset register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	BO	Address byte offset for parity error. Indicates that an error has occured in byte offset #n. 0: No error occuredin byte offset #n; 1: Error occured in byte offset #n. Write to clear any of MEM.SYSERR/DMARR/ARP32/or VERR.	R	0x0

Table 8-94. Register Call Summary for Register EVE_WBUF_EDADDR_BO

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-95. EVE_IBUF_ED_CTL

Address Offset	0x0008 00B0		
Physical Address	0x4208 00B0 0x0208 00B0 0x4218 00B0 0x0218 00B0	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	IBUF error detection control		
Type	RW		

31–2	1	0
RESERVED	INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reads from meory return inverse of parity bit. Write 0 to clear, write 1 to set. Must be set when EN is set	RW	0x0
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 8-96. Register Call Summary for Register EVE_IBUF_ED_CTL

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-97. EVE_IBUF_ED_STAT

Address Offset	0x0008 00B4		
Physical Address	0x4208 00B4 0x0208 00B4 0x4218 00B4 0x0218 00B4	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	IBUF error detection status register		
Type	RW		

31–25	24–16	15–4	3	2	1	0
RESERVED	SYSCONNID	RESERVED	SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 8-98. Register Call Summary for Register EVE_IBUF_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- VCOP System Error Halt Conditions: [1]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [2] [3] [4] [5]
- EVE Register Summary: [6]

Table 8-99. EVE_IBUF_EDADDR

Address Offset	0x0008 00B8		
Physical Address	0x4208 00B8 0x0208 00B8 0x4218 00B8 0x0218 00B8	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	IBUF error detection address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 8-100. Register Call Summary for Register EVE_IBUF_EDADDR

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-101. EVE_IBUF_EDADDR_BO

Address Offset	0x0008 00BC		
Physical Address	0x4208 00BC 0x0208 00BC 0x4218 00BC 0x0218 00BC	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	IBUF error detection address byte offset register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	BO	Address byte offset for parity error. Indicates that an error has occured in byte offset #n. 0: No error occuredin byte offset #n; 1: Error occured in byte offset #n. Write to clear any of MEM.SYSERR/DMARR/ARP32/or VERR.	R	0x0

Table 8-102. Register Call Summary for Register EVE_IBUF_EDADDR_BO

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0] [1]
- EVE Register Summary: [2]

Table 8-103. EVE_ED_ARP32_DISC_EN

Address Offset	0x0008 00F8		
Physical Address	0x4208 00F8 0x0208 00F8 0x4218 00F8 0x0218 00F8	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	ARP32 disconnect enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Disconnect Enable for Event #n 0: Disconnect disabled 1: Disconnect enabled	RW	0x0000

Table 8-104. Register Call Summary for Register EVE_ED_ARP32_DISC_EN

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- ARP32 Disconnect: [1]
- Lock/Unlock feature: [2]
- EVE Register Summary: [3]

Table 8-105. EVE_ED_OCPI_DISC_EN

Address Offset	0x0008 00FC		
Physical Address	0x4208 00FC 0x0208 00FC 0x4218 00FC 0x0218 00FC	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	OCP interface disconnect enable register		

Table 8-105. EVE_ED_OCPI_DISC_EN (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Disconnect Enable for Event #n. 0: Disconnect disabled 1: Disconnect enabled	RW	0x0000

Table 8-106. Register Call Summary for Register EVE_ED_OCPI_DISC_EN

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- OCP Initiators Disconnect: [1]
- Lock/Unlock feature: [2]
- EVE Register Summary: [3]

Table 8-107. EVE_MSW_ERR_IRQSTATUS_RAW

Address Offset	0x0008 0110		
Physical Address	0x4208 0110 0x0208 0110 0x4218 0110 0x0218 0110	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Per event memory switch error interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EVENT			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0

Table 8-108. Register Call Summary for Register EVE_MSW_ERR_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- VCOP Working Buffer (WBUF): [0]
- Image Buffers - IBUFLA, IBUFLB, IBUFHA, IBUFHB: [1]
- Memory Switch Error Registers: [2]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [3] [4] [5]
- EVE Register Summary: [6]

Table 8-109. EVE_MSW_ERR_IRQSTATUS

Address Offset	0x0008 0114		
Physical Address	0x4208 0114 0x0208 0114 0x4218 0114 0x0218 0114	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Memory switch error interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	EVENT

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0

Table 8-110. Register Call Summary for Register EVE_MSW_ERR_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- Memory Switch Error Registers: [0]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [1] [2]
- EVE Register Summary: [3]

Table 8-111. EVE_MSW_ERR_IRQENABLE_SET

Address Offset	0x0008 0118		
Physical Address	0x4208 0118 0x0208 0118 0x4218 0118 0x0218 0118	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Memory switch error interrupt enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ENABLE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0

Table 8-112. Register Call Summary for Register EVE_MSW_ERR_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- Memory Switch Error Registers: [0]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [1]
- EVE Register Summary: [2]

Table 8-113. EVE_MSW_ERR_IRQENABLE_CLR

Address Offset	0x0008 011C		
Physical Address	0x4208 011C 0x0208 011C 0x4218 011C 0x0218 011C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Memory switch error interrupt clear register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ENABLE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0

Table 8-114. Register Call Summary for Register EVE_MSW_ERR_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- Memory Switch Error Registers: [0]
- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [1]
- EVE Register Summary: [2]

Table 8-115. EVE_ED_LCL_IRQSTATUS_RAW

Address Offset	0x0008 0120		
Physical Address	0x4208 0120 0x0208 0120 0x4218 0120 0x0218 0120	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Per event error detection local interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 8-116. Register Call Summary for Register EVE_ED_LCL_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0] [1]
- ARP32 Interrupt Controller: [2]
- EVE Register Summary: [3]

Table 8-117. EVE_ED_LCL_IRQSTATUS

Address Offset	0x0008 0124		
Physical Address	0x4208 0124 0x0208 0124 0x4218 0124 0x0218 0124	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Error detection local interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 8-118. Register Call Summary for Register EVE_ED_LCL_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0]
- ARP32 Interrupt Controller: [1]
- EVE Register Summary: [2]

Table 8-119. EVE_ED_LCL_IRQENABLE_SET

Address Offset	0x0008 0128		
Physical Address	0x4208 0128 0x0208 0128 0x4218 0128 0x0218 0128	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Error detection local interrupt enable register.		
Type	RW		

31:16	15:0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 8-120. Register Call Summary for Register EVE_ED_LCL_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0] [1] [2]
- EVE Register Summary: [3]

Table 8-121. EVE_ED_LCL_IRQENABLE_CLR

Address Offset	0x0008 012C		
Physical Address	0x4208 012C 0x0208 012C 0x4218 012C 0x0218 012C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Error detection local interrupt clear register.		
Type	RW		

31:16	15:0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 8-122. Register Call Summary for Register EVE_ED_LCL_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0]
- EVE Register Summary: [1]

Table 8-123. ARP32_NMI_IRQSTATUS_RAW

Address Offset	0x0008 0200

Table 8-123. ARP32_NMI_IRQSTATUS_RAW (continued)

Physical Address	0x4208 0200 0x0208 0200 0x4218 0200 0x0218 0200	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 8-124. Register Call Summary for Register ARP32_NMI_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- ARP32 Interrupt Controller: [0] [1]
- EVE Register Summary: [2]

Table 8-125. ARP32_NMI_IRQSTATUS

Address Offset	0x0008 0204		
Physical Address	0x4208 0204 0x0208 0204 0x4218 0204 0x0218 0204	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 8-126. Register Call Summary for Register ARP32_NMI_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- ARP32 Interrupt Controller: [0]
- EVE Register Summary: [1]

Table 8-127. ARP32_NMI_IRQENABLE_SET

Address Offset	0x0008 0208		
Physical Address	0x4208 0208 0x0208 0208 0x4218 0208 0x0218 0208	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 8-128. Register Call Summary for Register ARP32_NMI_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-129. ARP32_NMI_IRQENABLE_CLR

Address Offset	0x0008 020C		
Physical Address	0x4208 020C 0x0208 020C 0x4218 020C 0x0218 020C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 8-130. Register Call Summary for Register ARP32_NMI_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-131. ARP32_INTn_IRQSTATUS_RAW

Address Offset	0x0008 01D0 + (0x10*n)		
Physical Address	0x4208 01D0 + (0x10*n) 0x0208 01D0 + (0x10*n) 0x4218 01D0 + (0x10*n) 0x0218 01D0 + (0x10*n)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 8-132. Register Call Summary for Register ARP32_INTn_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- ARP32 Interrupt Controller: [0] [1]
- EVE Register Summary: [2]

Table 8-133. ARP32_INTn_IRQSTATUS

Address Offset	0x0008 01D4 + (0x10*n)

Table 8-133. ARP32_INTn_IRQSTATUS (continued)

Physical Address	0x4208 01D4 + (0x10*n) 0x0208 01D4 + (0x10*n) 0x4218 01D4 + (0x10*n) 0x0218 01D4 + (0x10*n)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 8-134. Register Call Summary for Register ARP32_INTn_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0]
- ARP32 Interrupt Controller: [1]
- Interrupts: [2] [3] [4]
- Interrupt Servicing: [5]
- EVE Register Summary: [6]

Table 8-135. ARP32_INTn_IRQENABLE_SET

Address Offset	0x0008 01D8 + (0x10*n)		
Physical Address	0x4208 01D8 + (0x10*n) 0x0208 01D8 + (0x10*n) 0x4218 01D8 + (0x10*n) 0x0218 01D8 + (0x10*n)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 8-136. Register Call Summary for Register ARP32_INTn_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0]
- Interrupts: [1]
- EVE Register Summary: [2]

Table 8-137. ARP32_INTn_IRQENABLE_CLR

Address Offset	0x0008 01DC + (0x10*n)		
Physical Address	0x4208 01DC + (0x10*n) 0x0208 01DC + (0x10*n) 0x4218 01DC + (0x10*n) 0x0218 01DC + (0x10*n)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			

Table 8-137. ARP32_INTn_IRQENABLE_CLR (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 8-138. Register Call Summary for Register ARP32_INTn_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-139. ARP32_IRQWAKEEN

Address Offset	0x0000 02FC		
Physical Address	0x4208 02FC 0x0208 02FC 0x4218 02FC 0x0218 02FC	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	Wake enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ENABLE																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	ENABLE	Wakeup Enable for event EVE_EVT_INT #n 0: Interrupt #n disabled for wakeup 1: Interrupt #n enabled for wakeup	RW	0x00 0000

Table 8-140. Register Call Summary for Register ARP32_IRQWAKEEN

Embedded Vision Engine (EVE) Subsystem

- Extended Duration Sleep: [0] [1] [2]
- EVE Register Summary: [3]

Table 8-141. MMR_LOCKi

Address Offset	0x0008 0300 + (0x4*i)		
Physical Address	0x4208 0300 + (0x4*i) 0x0208 0300 + (0x4*i) 0x4218 0300 + (0x4*i) 0x0218 0300 + (0x4*i)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description	MMR Lock/Unlock register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK	Lock/Unlock register for corresponding region	RW	0x0000 0000

Table 8-142. Register Call Summary for Register MMR_LOCKi

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-143. MISR_CTL

Address Offset	0x0008 0400		
Physical Address	0x4208 0400 0x0208 0400 0x4218 0400 0x0218 0400	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–3	2–0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED			
2:0	ENABLE	MISR Enable #N 0: MISR #n disabled 1: MISR #n enabled Bit 0: MISR 0 - ARP32 PMEM path Bit 1: MISR 1 - ARP32 DMEM path Bit 2: MISR 2 - INTC WBUF path	RW	0x0000

Table 8-144. Register Call Summary for Register MISR_CTL

Embedded Vision Engine (EVE) Subsystem

- Hardware Assisted software self test - MISRs: [0]
- EVE Register Summary: [1]

Table 8-145. MISR_CLEAR

Address Offset	0x0008 0404		
Physical Address	0x4208 0404 0x0208 0404 0x4218 0404 0x0218 0404	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–3	2–0
RESERVED	CLEAR

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLEAR	MISR Clear #N Write 0: no effect Write 1: Clear MISR #n Read 0: Previous MISR clear command has completed. Read 1: MISR Clear in progress (this state may never actually be readable)	RW	0x00

Table 8-146. Register Call Summary for Register MISR_CLEAR

Embedded Vision Engine (EVE) Subsystem

- Hardware Assisted software self test - MISRs: [0]
- EVE Register Summary: [1]

Table 8-147. MISR0_A

Address Offset	0x0008 0410

Table 8-147. MISR0_A (continued)

Physical Address	0x4208 0410 0x0208 0410 0x4218 0410 0x0218 0410	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 8-148. Register Call Summary for Register MISR0_A

Embedded Vision Engine (EVE) Subsystem

- Hardware Assisted software self test - MISRs: [0] [1]
- EVE Register Summary: [2]

Table 8-149. MISR0_D

Address Offset	0x0008 0414		
Physical Address	0x4208 0414 0x0208 0414 0x4218 0414 0x0218 0414	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 8-150. Register Call Summary for Register MISR0_D

Embedded Vision Engine (EVE) Subsystem

- Hardware Assisted software self test - MISRs: [0] [1]
- EVE Register Summary: [2]

Table 8-151. MISR1_A

Address Offset	0x0008 0418		
Physical Address	0x4208 0418 0x0208 0418 0x4218 0418 0x0218 0418	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 8-152. Register Call Summary for Register MISR1_A

Embedded Vision Engine (EVE) Subsystem

- Hardware Assisted software self test - MISRs: [0] [1]
- EVE Register Summary: [2]

Table 8-153. MISR1_D

Address Offset	0x0008 041C		
Physical Address	0x4208 041C 0x0208 041C 0x4218 041C 0x0218 041C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 8-154. Register Call Summary for Register MISR1_D

Embedded Vision Engine (EVE) Subsystem

- Hardware Assisted software self test - MISRs: [0] [1]
- Mapping of MISRs to Different Width buses: [2]
- EVE Register Summary: [3]

Table 8-155. MISR2_Dk

Address Offset	0x0008 0420 + (0x4*k)		
Physical Address	0x4208 0420 + (0x4*k) 0x0208 0420 + (0x4*k) 0x4218 0420 + (0x4*k) 0x0218 0420 + (0x4*k)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 8-156. Register Call Summary for Register MISR2_Dk

Embedded Vision Engine (EVE) Subsystem

- Hardware Assisted software self test - MISRs: [0] [1]
- Mapping of MISRs to Different Width buses: [2]
- EVE Register Summary: [3]

Table 8-157. EVE_IRQ_EOI

Address Offset	0x0008 0500		
Physical Address	0x4208 0500 0x0208 0500 0x4218 0500 0x0218 0500	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													LINE_NUMBER		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write number of interrupt output to force re-evaluation of associated pending interrupts. Refer to Section Error: Reference source not found for EOI mapping. Reads always return 0x0. Write n : EOI for Interrupt output associated w/ EOI #n.	RW	0x0

Table 8-158. Register Call Summary for Register EVE_IRQ_EOI

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-159. EVE_ED_OUT_IRQSTATUS_RAW

Address Offset	0x0008 0510		
Physical Address	0x4208 0510 0x0208 0510 0x4218 0510 0x0218 0510	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
15:0	EVENT	Settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 8-160. Register Call Summary for Register EVE_ED_OUT_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0] [1]
- ARP32 Interrupt Controller: [2]
- Lock/Unlock feature: [3]
- EVE Register Summary: [4]

Table 8-161. EVE_ED_OUT_IRQSTATUS

Address Offset	0x0008 0514		
Physical Address	0x4208 0514 0x0208 0514 0x4218 0514 0x0218 0514	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–16	15–0
RESERVED	EVENT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 8-162. Register Call Summary for Register EVE_ED_OUT_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0]
- ARP32 Interrupt Controller: [1]
- Lock/Unlock feature: [2]
- EVE Register Summary: [3]

Table 8-163. EVE_ED_OUT_IRQENABLE_SET

Address Offset	0x0008 0518		
Physical Address	0x4208 0518 0x0208 0518 0x4218 0518 0x0218 0518	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–16	15–0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 8-164. Register Call Summary for Register EVE_ED_OUT_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-165. EVE_ED_OUT_IRQENABLE_CLR

Address Offset	0x0008 051C		
Physical Address	0x4208 051C 0x0208 051C 0x4218 051C 0x0218 051C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–16	15–0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 8-166. Register Call Summary for Register EVE_ED_OUT_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources - Memory Switch and Parity Error Interrupts: [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-167. EVE_INTk_OUT_IRQSTATUS_RAW

Address Offset	0x0008 0520 + (0x10*k)		
Physical Address	0x4208 0520 + (0x10*k) 0x0208 0520 + (0x10*k) 0x4218 0520 + (0x10*k) 0x0218 0520 + (0x10*k)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–0
EVENT

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000 0000

Table 8-168. Register Call Summary for Register EVE_INTk_OUT_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- Lock/Unlock feature: [0]
- EVE Register Summary: [1]

Table 8-169. EVE_INTk_OUT_IRQSTATUS

Address Offset	0x0008 0524 + (0x10*k)

Table 8-169. EVE_INTk_OUT_IRQSTATUS (continued)

Physical Address	0x4208 0524 + (0x10*k) 0x0208 0524 + (0x10*k) 0x4218 0524 + (0x10*k) 0x0218 0524 + (0x10*k)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event.	RW	0x0000 0000

Table 8-170. Register Call Summary for Register EVE_INTk_OUT_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- ARP32 Interrupt Controller: [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-171. EVE_INTk_OUT_IRQENABLE_SET

Address Offset	0x0008 0528 + (0x10*k)		
Physical Address	0x4208 0528 + (0x10*k) 0x0208 0528 + (0x10*k) 0x4218 0528 + (0x10*k) 0x0218 0528 + (0x10*k)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000 0000

Table 8-172. Register Call Summary for Register EVE_INTk_OUT_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- Lock/Unlock feature: [0]
- EVE Register Summary: [1]

Table 8-173. EVE_INTk_OUT_IRQENABLE_CLR

Address Offset	0x0008 052C + (0x10*k)		
Physical Address	0x4208 052C + (0x10*k) 0x0208 052C + (0x10*k) 0x4218 052C + (0x10*k) 0x0218 052C + (0x10*k)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ENABLE

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000 0000

Table 8-174. Register Call Summary for Register EVE_INTk_OUT_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- ARP32 Interrupt Controller: [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-175. ARP32_INTj_IRQSTATUS_RAW

Address Offset	0x0008 0580 + (0x10*j)		
Physical Address	0x4208 0580 + (0x10*j) 0x0208 0580 + (0x10*j) 0x4218 0580 + (0x10*j) 0x0218 0580 + (0x10*j)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	R	0x0000 0000

Table 8-176. Register Call Summary for Register ARP32_INTj_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-177. ARP32_INTj_IRQSTATUS

Address Offset	0x0008 0584 + (0x10*j)		
Physical Address	0x4208 0584 + (0x10*j) 0x0208 0584 + (0x10*j) 0x4218 0584 + (0x10*j) 0x0218 0584 + (0x10*j)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000 0000

Table 8-178. Register Call Summary for Register ARP32_INTj_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-179. ARP32_INTj_IRQENABLE_SET

Address Offset	0x00008 0588 + (0x10*j)		
Physical Address	0x4208 0588 + (0x10*j) 0x0208 0588 + (0x10*j) 0x4218 0588 + (0x10*j) 0x0218 0588 + (0x10*j)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 8-180. Register Call Summary for Register ARP32_INTj_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-181. ARP32_INTj_IRQENABLE_CLR

Address Offset	0x00008 058C + (0x10*j)		
Physical Address	0x4208 058C + (0x10*j) 0x0208 058C + (0x10*j) 0x4218 058C + (0x10*j) 0x0218 058C + (0x10*j)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 8-182. Register Call Summary for Register ARP32_INTj_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-183. ARP32_INT14_IRQSTATUS_RAW

Address Offset	0x0008 0680		
Physical Address	0x4208 0680 0x0208 0680 0x4218 0680 0x0218 0680	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	R	0x00

Table 8-184. Register Call Summary for Register ARP32_INT14_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-185. ARP32_INT14_IRQSTATUS

Address Offset	0x0008 0684		
Physical Address	0x4208 0684 0x0208 0684 0x4218 0684 0x0218 0684	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000 0000

Table 8-186. Register Call Summary for Register ARP32_INT14_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-187. ARP32_INT14_IRQENABLE_SET

Address Offset	0x0008 0688		
Physical Address	0x4208 0688 0x0208 0688 0x4218 0688 0x0218 0688	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 8-188. Register Call Summary for Register ARP32_INT14_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-189. ARP32_INT14_IRQENABLE_CLR

Address Offset	0x0008 068C

Table 8-189. ARP32_INT14_IRQENABLE_CLR (continued)

Physical Address	0x4208 068C 0x0208 068C 0x4218 068C 0x0218 068C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0

Table 8-190. Register Call Summary for Register ARP32_INT14_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-191. ARP32_INT15_IRQSTATUS_RAW

Address Offset	0x0008 0690		
Physical Address	0x4208 0690 0x0208 0690 0x4218 0690 0x0218 0690	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	R	0x00

Table 8-192. Register Call Summary for Register ARP32_INT15_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-193. ARP32_INT15_IRQSTATUS

Address Offset	0x0008 0694		
Physical Address	0x4208 0694 0x0208 0694 0x4218 0694 0x0218 0694	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000 0000

Table 8-194. Register Call Summary for Register ARP32_INT15_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-195. ARP32_INT15_IRQENABLE_SET

Address Offset	0x0008 0698		
Physical Address	0x4208 0698 0x0208 0698 0x4218 0698 0x0218 0698	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 8-196. Register Call Summary for Register ARP32_INT15_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-197. ARP32_INT15_IRQENABLE_CLR

Address Offset	0x0008 069C		
Physical Address	0x4208 069C 0x0208 069C 0x4218 069C 0x0218 069C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0

Table 8-198. Register Call Summary for Register ARP32_INT15_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-199. EVE_GPOUTm

Address Offset	0x0008 0700 + (0x10*m)

Table 8-199. EVE_GPOUTm (continued)

Physical Address	0x4208 0700 + (0x10*m) 0x0208 0700 + (0x10*m) 0x4218 0700 + (0x10*m) 0x0218 0700 + (0x10*m)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: Drive GP Output #n low/1 Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1. Read 1: GP Output is high/1.	RW	0x0000 0000

Table 8-200. Register Call Summary for Register EVE_GPOUTm

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0] [1]
- Lock/Unlock feature: [2]
- EVE Register Summary: [3]

Table 8-201. EVE_GPOUTm_SET

Address Offset	0x0008 0704 + (0x10*m)		
Physical Address	0x4208 0704 + (0x10*m) 0x0208 0704 + (0x10*m) 0x4218 0704 + (0x10*m) 0x0218 0704 + (0x10*m)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: No action Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1. Read 1: GP Output is high/1.	RW	0x0000 0000

Table 8-202. Register Call Summary for Register EVE_GPOUTm_SET

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0] [1] [2]
- Lock/Unlock feature: [3]
- EVE Register Summary: [4]

Table 8-203. EVE_GPOUTm_CLR

Address Offset	0x0008 0708 + (0x10*m)		
Physical Address	0x4208 0708 + (0x10*m) 0x0208 0708 + (0x10*m) 0x4218 0708 + (0x10*m) 0x0218 0708 + (0x10*m)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EVENT

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: Drive GP Output #n low/1 Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1. Read 1: GP Output is high/1.	RW	0x0000 0000

Table 8-204. Register Call Summary for Register EVE_GPOUTm_CLR

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0] [1]
- EVE Register Summary: [2]

Table 8-205. EVE_GPOUTm_PULSE

Address Offset	0x0008 070C + (0x10*m)		
Physical Address	0x4208 070C + (0x10*m) 0x0208 070C + (0x10*m) 0x4218 070C + (0x10*m) 0x0218 070C + (0x10*m)	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: No action Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1 for four cycles / then drive low/0. Read 1: GP Output #n is high/1. Note: Writing to GPOUT registers when the four cycles for the previous write to GPOUT_PULSE register have not been completed can result in unpredictable output.	RW	0x0000 0000

Table 8-206. Register Call Summary for Register EVE_GPOUTm_PULSE

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0] [1] [2] [3] [4]
- EVE Register Summary: [5]

Table 8-207. EVE_GPIN0

Address Offset	0x0008 0740		
Physical Address	0x4208 0740 0x0208 0740 0x4218 0740 0x0218 0740	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Input #n Read 0: GP Input #n is low/0. Read 1: GP Input is high/1.	R	0x0000 0000

Table 8-208. Register Call Summary for Register EVE_GPIN0

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-209. EVE_GPIN1

Address Offset	0x0008 0744		
Physical Address	0x4208 0744 0x0208 0744 0x4218 0744 0x0218 0744	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	R		

31–0
EVENT

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Input #n Read 0: GP Input #n is low/0. Read 1: GP Input is high/1.	R	0x0000 0000

Table 8-210. Register Call Summary for Register EVE_GPIN1

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0]
- Lock/Unlock feature: [1]
- EVE Register Summary: [2]

Table 8-211. EVE_CME_DONE_GPOUT

Address Offset	0x0008 0780		
Physical Address	0x4208 0780 0x0208 0780 0x4218 0780 0x0218 0780	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–8	7–0
RESERVED	EVENT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done Output #n. Write 0: Drive Internal CME Done #n low/1 Read 0: Drive Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n high/1. Read 1: Internal CME Done is high/1.	RW	0x00

Table 8-212. Register Call Summary for Register EVE_CME_DONE_GPOUT

Embedded Vision Engine (EVE) Subsystem

- CME Signalling: [0] [1]
- EVE Register Summary: [2]

Table 8-213. EVE_CME_DONE_GPOUT_SET

Address Offset	0x0008 0784

Table 8-213. EVE_CME_DONE_GPOUT_SET (continued)

Physical Address	0x4208 0784 0x0208 0784 0x4218 0784 0x0218 0784	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVENT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done #n Write 0: No action Read 0: Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n high/1. Read 1: Internal CME Done is high/1.	RW	0x00

Table 8-214. Register Call Summary for Register EVE_CME_DONE_GPOUT_SET

Embedded Vision Engine (EVE) Subsystem

- CME Signalling: [0]
- EVE Register Summary: [1]

Table 8-215. EVE_CME_DONE_GPOUT_CLR

Address Offset	0x0008 0788		
Physical Address	0x4208 0788 0x0208 0788 0x4218 0788 0x0218 0788	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVENT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done #n Write 0: No action Read 0: Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n low/0. Read 1: Internal CME Done #n is high/1.	RW	0x00

Table 8-216. Register Call Summary for Register EVE_CME_DONE_GPOUT_CLR

Embedded Vision Engine (EVE) Subsystem

- CME Signalling: [0]
- EVE Register Summary: [1]

Table 8-217. EVE_CME_DONE_GPOUT_PULSE

Address Offset	0x0008 078C		
Physical Address	0x4208 078C 0x0208 078C 0x4218 078C 0x0218 078C	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			

Table 8-217. EVE_CME_DONE_GPOUT_PULSE (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVENT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done #n Write 0: No action Read 0: Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n high/1 for four cycles / then drive low/0. Read 1: Internal CME Done #n is high/1. Note: Writing to GPOUT registers when the four cycles for the previous write to GPOUT_PULSE register have not been completed can result in unpredictable output.	RW	0x00

Table 8-218. Register Call Summary for Register EVE_CME_DONE_GPOUT_PULSE

Embedded Vision Engine (EVE) Subsystem

- CME Signalling: [0]
- EVE Register Summary: [1]

Table 8-219. EVE_CME_DONE_SEL

Address Offset	0x0008 0790		
Physical Address	0x4208 0790 0x0208 0790 0x4218 0790 0x0218 0790	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL7				SEL6				SEL5				SEL4				SEL3				SEL2				SEL1				SEL0			

Bits	Field Name	Description	Type	Reset
31:28	SEL7	CME Done Output select for Bit #7 (n=7).	RW	0x0
27:24	SEL6	CME Done Output select for Bit #6 (n=6).	RW	0x0
23:20	SEL5	CME Done Output select for Bit #5 (n=5).	RW	0x0
19:16	SEL4	CME Done Output select for Bit #4 (n=4).	RW	0x0
15:12	SEL3	CME Done Output select for Bit #3 (n=3).	RW	0x0
11:8	SEL2	CME Done Output select for Bit #2 (n=2).	RW	0x0
7:4	SEL1	CME Done Output select for Bit #1 (n=1).	RW	0x0
3:0	SEL0	CME Done Output select for Bit #0 (n=0) 0: Driven by EDMA cc_int0 1: Driven by EDMA cc_int1 2: Driven by EDMA cc_int2 3: Driven by EDMA cc_int3 4: Driven by EDMA cc_int4 5: Driven by EDMA cc_int5 6: Driven by EDMA cc_int6 7: Driven by EDMA cc_int7 8: Driven by EVE_CME_DONE_GPOUTn 9: driven by eve_cme_done_gpout[0+n] (from EVE1) 10: driven by eve_cme_done_gpout[8+n] (from EVE2)	RW	0x0

Table 8-220. Register Call Summary for Register EVE_CME_DONE_SEL

Embedded Vision Engine (EVE) Subsystem

- CME Signalling: [0] [1] [2]
- EVE Register Summary: [3]

Table 8-221. EVE_CME_DONE_EN

Address Offset	0x0008 794		
Physical Address	0x4208 0794 0x0208 0794 0x4218 0794 0x0218 0794	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31–8	7–0
RESERVED	EN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0000 0000
7:0	EN	EVE CME Done EN #n Write 0: No action Read 0: EVE CME Done #n is disabled. Write 1: Enable EVE CME Done #n event. This allows the status of the CME Done #n to propagate to the output. Read 1: EVE CME Done #n is enabled.	RW	0x0

Table 8-222. Register Call Summary for Register EVE_CME_DONE_EN

Embedded Vision Engine (EVE) Subsystem

- CME Signalling: [0]
- EVE Register Summary: [1]

Table 8-223. EVE_PM_STAT0

Address Offset	0x0008 0FE0		
Physical Address	0x4208 0FE0 0x0208 0FE0 0x4218 0FE0 0x0218 0FE0	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	R		

31	30–28	27–26	25–24	23	22–20	19–18	17–16	15	14–12	11–10	9–8	7–6	5	4	3	2–1	0
RESERVED	OCPM1_SCONNECT	RESERVED	OCPM1_MCONNECT	RESERVED	OCPM0_SCONNECT	RESERVED	OCPM0_MCONNECT	RESERVED	OCPS_SCONNECT	RESERVED	OCPS_MCONNECT	RESERVED	MWAIT	MSTANDBY	SWAKEUP	SIDLEACK	SIDLEREQ

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	OCPM1_SCONNECT	Readable state of OCP Power management handshake	R	0x0
27:26	RESERVED		R	0x0
25:24	OCPM1_MCONNECT	Readable state of OCP Power management handshake	R	0x0
23	RESERVED		R	0x0
22:20	OCPM0_SCONNECT	Readable state of OCP Power management handshake	R	0x0
19:18	RESERVED		R	0x0
17:16	OCPM0_MCONNECT	Readable state of OCP Power management handshake	R	0x0
15	RESERVED		R	0x0
14:12	OCPS_SCONNECT	Readable state of OCP Power management handshake	R	0x0
11:10	RESERVED		R	0x0
9:8	OCPS_MCONNECT	Readable state of OCP Power management handshake	R	0x0

Bits	Field Name	Description	Type	Reset
7:6	RESERVED		R	0x0
5	MWAIT	Readable state of OCP Power management handshake	R	0x0
4	MSTANDBY	Readable state of OCP Power management handshake	R	0x0
3	SWAKEUP	Readable state of OCP Power management handshake	R	0x0
2:1	SIDLEACK	Readable state of OCP Power management handshake	R	0x0
0	SIDLEREQ	Readable state of OCP Power management handshake	R	0x0

Table 8-224. Register Call Summary for Register EVE_PM_STAT0

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-225. EVE_PM_STAT1

Address Offset	0x0008 0FE4		
Physical Address	0x4208 0FE4 0x0208 0FE4 0x4218 0FE4 0x0218 0FE4	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STBY_MDISCACK_OCPM1		STBY_MDISCACK_OCPM0		STBY_MDISCREQ_OCPM1	STBY_MDISCREQ_OCPM0	IDLE_SDISCONNECT_ACK	IDLE_SDISCONNECT_REQ	EVE_IDLE_INTR_DISABLE	TPTC1_MWAIT	TPTC0_MWAIT	EVE_PCACHE_OCP_BUSY	EVE_CONTROL_SIDLEACK	SMSET_SIDLEACK	L2_EVE_SIDLEACK	MMU1_CONFIG_SIDLEACK	MMU1_SIDLEACK	MMU0_CONFIG_SIDLEACK	MMU0_SIDLEACK	SCTM_SIDLEACK	TPCC_SIDLEACK	TPTC1_SIDLEACK	TPTC0_SIDLEACK	SUBMODULE_IDLE_REQ

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:22	STBY_MDISCACK_OCPM1	Readable state of internal power management handshake	R	0x0
21:20	STBY_MDISCACK_OCPM0	Readable state of internal power management handshake	R	0x0
19	STBY_MDISCREQ_OCPM1	Readable state of internal power management handshake	R	0x0
18	STBY_MDISCREQ_OCPM0	Readable state of internal power management handshake	R	0x0
17	IDLE_SDISCONNECT_ACK	Readable state of internal power management handshake	R	0x0
16	IDLE_SDISCONNECT_REQ	Readable state of internal power management handshake	R	0x0
15	EVE_IDLE_INTR_DISABLE	Readable state of internal power management handshake	R	0x0
14	TPTC1_MWAIT	Readable state of internal power management handshake	R	0x0
13	TPTC0_MWAIT	Readable state of internal power management handshake	R	0x0

Bits	Field Name	Description	Type	Reset
12	EVE_PCACHE_OCP_BUSY	Readable state of internal power management handshake	R	0x0
11	EVE_CONTROL_SIDLEACK	Readable state of internal power management handshake	R	0x0
10	SMSET_SIDLEACK	Readable state of internal power management handshake	R	0x0
9	L2_EVE_SIDLEACK	Readable state of internal power management handshake	R	0x0
8	MMU1_CONFIG_SIDLEACK	Readable state of internal power management handshake	R	0x0
7	MMU1_SIDLEACK	Readable state of internal power management handshake	R	0x0
6	MMU0_CONFIG_SIDLEACK	Readable state of internal power management handshake	R	0x0
5	MMU0_SIDLEACK	Readable state of internal power management handshake	R	0x0
4	SCTM_SIDLEACK	Readable state of internal power management handshake	R	0x0
3	TPCC_SIDLEACK	Readable state of internal power management handshake	R	0x0
2	TPTC1_SIDLEACK	Readable state of internal power management handshake	R	0x0
1	TPTC0_SIDLEACK	Readable state of internal power management handshake	R	0x0
0	SUBMODULE_IDLE_REQ			

Table 8-226. Register Call Summary for Register EVE_PM_STAT1

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-227. EVE_DBGOUT

Address Offset	0x0008 0FE8		
Physical Address	0x4208 0FE8 0x0208 0FE8 0x4218 0FE8 0x0218 0FE8	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																								RESERVED				GROUP			

Bits	Field Name	Description	Type	Reset
31:8	VALUE	Read returns state of eve_dbgout bus.	R	0x000
7:4	RESERVED		R	0x0
3:0	GROUP	Debug Group Output control : mux select 0x0 : disabled / all debug outputs driven to 0x0. 0x1 : select output group1 0x2 : select output group2 : 0xN : select output groupN Others - reserved	RW	0x0

Table 8-228. Register Call Summary for Register EVE_DBGOUT

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-229. EVE_RSVD0

Address Offset	0x0008 0FF4		
Physical Address	0x4208 0FF4 0x0208 0FF4 0x4218 0FF4 0x0218 0FF4	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															

Bits	Field Name	Description	Type	Reset
31:0	VAL	Value; This register is reserved for any necessary ECOs that may be required later in the design cycle.	RW	0x0000 0000

Table 8-230. Register Call Summary for Register EVE_RSVD0

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-231. EVE_RSVD1

Address Offset	0x0008 0FF8		
Physical Address	0x4208 0FF8 0x0208 0FF8 0x4218 0FF8 0x0218 0FF8	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	VAL	Value; This register is reserved for any necessary ECOs that may be required later in the design cycle.	RW	0x0000 0000

Table 8-232. Register Call Summary for Register EVE_RSVD1

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 8-233. EVE_TEST

Address Offset	0x0008 0FFC		
Physical Address	0x4208 0FFC 0x0208 0FFC 0x4218 0FFC 0x0218 0FFC	**Instance**	EVE1 EVE1_DSP EVE2 EVE2_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															

Bits	Field Name	Description	Type	Reset
31:0	VAL	Value; This register is reserved for any necessary ECOs that may be required later in the design cycle.	RW	0x0000 0000

Table 8-234. Register Call Summary for Register EVE_TEST

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

8.1.5.3 EVE L2_FNOC Register Summary and Description

8.1.5.3.1 EVE L2_FNOC Register Summary

Table 8-235 summarizes the EVE_L2FNOC registers.

Table 8-235. EVE L2_FNOC Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_L2_FNOC Physical Address	EVE2_L2_FNOC Physical Address
ERRLOGGER_i_ID_COREID (1)	R	32	0x0000 0000 + (0x200*i)	0x0208 A000 + (0x200*i)	0x4218 A000 + (0x200*i)
ERRLOGGER_i_ID_REVISIONID (1)	R	32	0x0000 0004 + (0x200*i)	0x0208 A004 + (0x200*i)	0x4218 A004 + (0x200*i)
ERRLOGGER_i_FAULTEN (1)	RW	32	0x0000 0008 + (0x200*i)	0x0208 A008 + (0x200*i)	0x4218 A008 + (0x200*i)
ERRLOGGER_i_ERRVLD (1)	R	32	0x0000 000C + (0x200*i)	0x0208 A00C + (0x200*i)	0x4218 A00C + (0x200*i)
ERRLOGGER_i_ERRCLR (1)	RW	32	0x0000 0010 + (0x200*i)	0x0208 A010 + (0x200*i)	0x4218 A010 + (0x200*i)
ERRLOGGER_i_ERRLOG0 (1)	R	32	0x0000 0014 + (0x200*i)	0x0208 A014 + (0x200*i)	0x4218 A014 + (0x200*i)
ERRLOGGER_i_ERRLOG1 (1)	R	32	0x0000 0018 + (0x200*i)	0x0208 A018 + (0x200*i)	0x4218 A018 + (0x200*i)
ERRLOGGER_i_ERRLOG3 (1)	R	32	0x0000 0020 + (0x200*i)	0x0208 A020 + (0x200*i)	0x4218 A020 + (0x200*i)
ERRLOGGER_i_ERRLOG5 (1)	R	32	0x0000 0028 + (0x200*i)	0x0208 A028 + (0x200*i)	0x4218 A028 + (0x200*i)
FLAGMUX_i_ID_COREID (1)	R	32	0x0000 0100 + (0x200*i)	0x0208 A100 + (0x200*i)	0x4218 A100 + (0x200*i)
FLAGMUX_i_ID_REVISIONID (1)	R	32	0x0000 0104 + (0x200*i)	0x0208 A104 + (0x200*i)	0x4218 A104 + (0x200*i)
FLAGMUX_i_FAULTEN (1)	RW	32	0x0000 0108 + (0x200*i)	0x0208 A108 + (0x200*i)	0x4218 A108 + (0x200*i)
FLAGMUX_i_FAULTSTATUS (1)	R	32	0x0000 010C + (0x200*i)	0x0208 A10C + (0x200*i)	0x4218 A10C + (0x200*i)
FLAGMUX_i_FLAGINEN0 (1)	RW	32	0x0000 0110 + (0x200*i)	0x0208 A110 + (0x200*i)	0x4218 A110 + (0x200*i)
FLAGMUX_i_FLAGINSTATUS0 (1)	R	32	0x0000 0114 + (0x200*i)	0x0208 A114 + (0x200*i)	0x4218 A114 + (0x200*i)

(1) i = 0 to 1 for EVE1_L2_FNOC
i = 0 to 1 for EVE2_L2_FNOC

8.1.5.3.2 EVE L2_FNOC Register Description

Table through describe the L2_FNOC registers in EVE.

Table 8-236. ERRLOGGER_i_ID_COREID

Address Offset	0x0000 0000		
Physical Address	0x0208 A000 + (0x200*i) 0x4218 A000 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC

Table 8-236. ERRLOGGER_i_ID_COREID (continued)

Description	
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORECHECKSUM																								CORETYPEID							

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0x220210[(1)]
7:0	CORETYPEID	Field identifying the type of IP.	R	0xd

(1) when i = 1 CHECKSUM reset value is 0x48443c

Table 8-237. Register Call Summary for Register ERRLOGGER_i_ID_COREID

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-238. ERRLOGGER_i_ID_REVISIONID

Address Offset	0x0000 0004		
Physical Address	0x0208 A004 + (0x200*i) 0x4218 A004 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLEXNOCID																								USERID							

Bits	Field Name	Description	Type	Reset
31:8	FLEXNOCID	Field containing the build revision of the software used to generate the IP HDL code.	R	0x10167
7:0	USERID	Field containing a user defined value, not used anywhere inside the IP itself.	R	0x0

Table 8-239. Register Call Summary for Register ERRLOGGER_i_ID_REVISIONID

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-240. ERRLOGGER_i_FAULTEN

Address Offset	0x0000 0008		
Physical Address	0x0208 A008 + (0x200*i) 0x4218 A008 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Enable Fault output	RW	0x0

Table 8-241. Register Call Summary for Register ERRLOGGER_i_FAULTEN

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-242. ERRLOGGER_i_ERRVLD

Address Offset	0x0000 000C		
Physical Address	0x0208 A00C + (0x200*i) 0x4218 A00C + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRVLD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRVLD	Error logged Valid	R	0x0

Table 8-243. Register Call Summary for Register ERRLOGGER_i_ERRVLD

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-244. ERRLOGGER_i_ERRCLR

Address Offset	0x0000 0010		
Physical Address	0x0208 A010 + (0x200*i) 0x4218 A010 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRCLR	Clr ErrVld status	RW	0x0

Table 8-245. Register Call Summary for Register ERRLOGGER_i_ERRCLR

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-246. ERRLOGGER_i_ERRLOG0

Address Offset	0x0000 0014		
Physical Address	0x0208 A014 + (0x200*i) 0x4218 A014 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description	Header: Lock, Opcode, Len1, ErrCode values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FORMAT	RESERVED	LEN1	RESERVED	ERRCODE	RESERVED	OPC	LOCK

Bits	Field Name	Description	Type	Reset
31	FORMAT	Format of ErrLog0 register	R	0x1
30:26	RESERVED		R	0x0
25:16	LEN1	Header: Len1 value	R	0x0
15:11	RESERVED		R	0x0
10:8	ERRCODE	Header: Error Code value	R	0x0
7:5	RESERVED		R	0x0
4:1	OPC	Header: Opcode value	R	0x0
0	LOCK	Header: Lock bit value	R	0x0

Table 8-247. Register Call Summary for Register ERRLOGGER_i_ERRLOG0

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-248. ERRLOGGER_i_ERRLOG1

Address Offset	0x0000 0018		
Physical Address	0x0208 A018 + (0x200*i) 0x4218 A018 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31:14	13:0
RESERVED	ERRLOG1

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	ERRLOG1	Header: RouteId lsb value	R	0x0

Table 8-249. Register Call Summary for Register ERRLOGGER_i_ERRLOG1

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-250. ERRLOGGER_i_ERRLOG3

Address Offset	0x0000 0020		
Physical Address	0x0208 A020 + (0x200*i) 0x4218 A020 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31:0
ERRLOG3

Bits	Field Name	Description	Type	Reset
31:0	ERRLOG3	Header: Addr lsb value	R	0x0

Table 8-251. Register Call Summary for Register ERRLOGGER_i_ERRLOG3

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-252. ERRLOGGER_i_ERRLOG5

Address Offset	0x0000 0028		
Physical Address	0x0208 A028 + (0x200*i) 0x4218 A028 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31–17	16–0
RESERVED	ERRLOG5

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16:0	ERRLOG5	Header: User lsb value	R	0x0

Table 8-253. Register Call Summary for Register ERRLOGGER_i_ERRLOG5

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-254. FLAGMUX_i_ID_COREID

Address Offset	0x0000 0100		
Physical Address	0x0208 A100 + (0x200*i) 0x4218 A100 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31–8	7–0
CORECHECKSUM	CORETYPEID

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0x2589ca (1)
7:0	CORETYPEID	Field identifying the type of IP.	R	0xb

(1) when i = 1 CHECKSUM reset value is 0x57c4a6

Table 8-255. Register Call Summary for Register FLAGMUX_i_ID_COREID

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-256. FLAGMUX_i_ID_REVISIONID

Address Offset	0x0000 0104		
Physical Address	0x0208 A104 + (0x200*i) 0x4218 A104 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31–8	7–0
FLEXNOCID	USERID

Bits	Field Name	Description	Type	Reset
31:8	FLEXNOCID	Field containing the build revision of the software used to generate the IP HDL code.	R	0x10167
7:0	USERID	Field containing a user defined value, not used anywhere inside the IP itself.	R	0x0

Table 8-257. Register Call Summary for Register FLAGMUX_i_ID_REVISIONID

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-258. FLAGMUX_i_FAULTEN

Address Offset	0x0000 0108		
Physical Address	0x0208 A108 + (0x200*i) 0x4218 A108 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	RW		

31:1	0
RESERVED	FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Global Fault Enable register	RW	0x0

Table 8-259. Register Call Summary for Register FLAGMUX_i_FAULTEN

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-260. FLAGMUX_i_FAULTSTATUS

Address Offset	0x0000 010C		
Physical Address	0x0208 A10C + (0x200*i) 0x4218 A10C + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31:1	0
RESERVED	FAULTSTATUS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTSTATUS	Global Fault Status register	R	0x0

Table 8-261. Register Call Summary for Register FLAGMUX_i_FAULTSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-262. FLAGMUX_i_FLAGINEN0

Address Offset	0x0000 0110		
Physical Address	0x0208 A110 + (0x200*i) 0x4218 A110 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			

Table 8-262. FLAGMUX_i_FLAGINEN0 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FLAGINEN0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINEN0	FlagIn Enable register #0	RW	0x0

Table 8-263. Register Call Summary for Register FLAGMUX_i_FLAGINEN0

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

Table 8-264. FLAGMUX_i_FLAGINSTATUS0

Address Offset	0x0000 0114		
Physical Address	0x0208 A114 + (0x200*i) 0x4218 A114 + (0x200*i)	**Instance**	EVE1_L2_FNOC EVE2_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FLAGINSTATUS0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINSTATUS0	FlagIn Status register #0	R	0x0

Table 8-265. Register Call Summary for Register FLAGMUX_i_FLAGINSTATUS0

Embedded Vision Engine (EVE) Subsystem

- EVE L2_FNOC Register Summary: [0]

8.1.6 Subsystem Counter Timer Module

8.1.6.1 Introduction

8.1.6.1.1 Overview

As the complexity of SoC architectures increases, system and subsystem optimization requires increased visibility into the frequency and duration of events within the system. One way to address these profiling requirements is to embed dedicated event or profile counters in all functional modules with profiling requirements. While this can address the fundamental requirements, there are several disadvantages to this approach:

- System architects must determine the detailed profiling requirements very early in the specification process.
- These dedicated profiling resources are generally quite similar across functional blocks so the redundant function introduces gate/area penalties in each functional module.
- Embedded profiling resources are harder to optimize out of subsequent system revisions when profiling requirements may be less prevalent.
- The embedded resources may have different configuration and control interfaces and this increases the difficulty of providing unified profiling data from all modules in a subsystem.

This section defines the functional specification for a generic counter timer module that can be instantiated within the processor subsystem and that also functions as a centralized profiling module for the entire subsystem. This module maps a large number of system and subsystem event signals to a smaller number of counter resources. Some of the counter resources in the module can be configured for timer functionality where system and/or debug events can be generated when designated intervals are matched.

In the EVE subsystem eight 32-bit counters are instantiated. Two of those eight counters can be configured as timers.

The generic centralized counter timer approach provides several advantages:

- Unified programmers view of all profiling resources within the subsystem
- Accessible by debug tools and application
- Parameterized synthesis for full configurability of the number of event inputs and counter resources
- A more flexible solution that lets designers tailor gate count to customer profiling requirements
- Standard configuration and control interfaces encourage reuse across subsystem modules within a complex SoC.
- In addition to profiling functions, resources in this module can be used to address normal application counter/ timer functions, such as OS timers.

8.1.6.1.2 Top-Level Requirements

The SCTM has the following top-level functional requirements:

- Counter timer resources:
  - A maximum of 32 counters can be instantiated in a single SCTM.
  - A maximum of eight of the counters can be instantiated with timer (event generation) functionality in addition to the base counter function.
- System event signal inputs:
  - A maximum of 127 event signal inputs can be supported by the SCTM.
  - Any of the system event signal inputs can be routed to any of the counter timer resources in the SCTM.
- Counter functionality:
  - The module contains a collection of 32-bit counters that can be chained to an adjacent counter for 64-bit capability.
  - The counters can be configured to operate with any one of the event signal inputs.
  - The counters can be configured to operate in a free-running mode that counts the total number of clock cycles.
  - The counters can operate in duration mode that counts the total duration in cycles that the assigned input signal is asserted.

  - The counters can operate in event mode that counts the total number of times the assigned event signal is asserted.
  - The counters can be configured to continue to run or to temporarily stop when the CPU enters the debug halt state.
  - The counters can be configured to continue to run or to temporarily stop when the CPU enters the IDLE state.
- Timer functionality:
  - Timers have all the functionality of counters.
  - Timers have a 32-bit interval match register.
  - Timers can be configured to generate an interrupt event when the counter matches the interval register.
  - Timers can be configured to generate a debug event when the counter matches the interval register.
  - The timers can be configured to run to the interval register once or to reinitialize each time the interval counter is matched.
- System Trace Interface:
  - The counter state dump can be configured as periodic or may be generated under application control.

8.1.6.1.3 Configuration

Table 8-266 lists the configuration of the SCTM in EVE.

Table 8-266. SCTM Configuration

Generic	SCTM Feature/Parameter Details	Value	Number on Device
CTM_NUMINPT	Number of event input signals supported	0 to 127	31
CTM_NUMCNTR	Number of counters in the module	1 to 32	8
CTM_NUMTIMR	Number of timers	0 to 8	2
CTM_TIMINTPOLARITY	Timer interrupt polarity	0 = Active low 1 = Active high	1
CTM_TIMINTWIDTH	Timer interrupt pulse width	1 or more	2
CTM_NUMSTM	Number of counters for STM export	0-32	0
CTM_CCMAVAIL	CCM frame export available	0 = No 1 = Yes	0
CTM_NUMDBGSGL	Number of debug event signals	0-8	0
CTM_ASYNCIDLEREQ	Whether IDLE request is asynchronous signal	0 = No 1 = Yes	0
CTM_CNTR_ CHAINSHADOW	Atomicity feature for even-indexed counters	One per counter. 0 = No chain shadow feature 1 = Chain shadow feature in counter chain mode	Counter numbers 2 and 4 have the chaining feature.

8.1.6.1.4 Block Diagram

Figure 8-21 represents a block diagram for the Counter Timer Module.

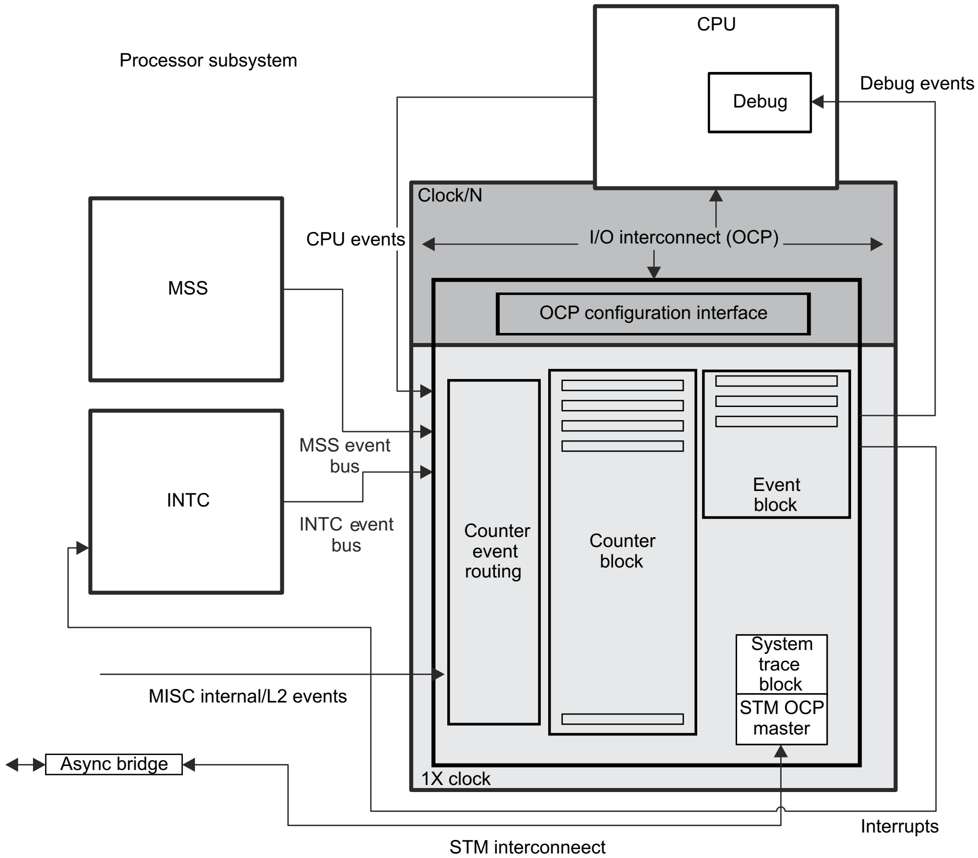


Figure 8-21. SCTM Block Diagram

8.1.6.2 Functional Description

The SCTM is a generic profile counter and timer module that can integrate these functions into a single module in a subsystem.

8.1.6.2.1 Configuration Interface

The SCTM has an OCP 2.1-compliant slave interface for configuring and controlling the module. For a detailed description of the register file and the individual registers, see Section 8.1.6.4, SCTM Register Manual. The configuration interface has the following basic functional requirements:

- Any unimplemented configuration memory space:
  - Generates a bus error on application read or write
  - Generates a bus error on debug right
  - Returns 0s and no bus error on debug read
- Any write to a read only register (application or debug) generates a bus error.

8.1.6.2.2 Counter Function

8.1.6.2.2.1 Input Events

Any signal of interest from within the subsystem can be routed to this module and used to control the counters and timers in the module. The routing of the input events from the module boundary to an individual counter is accomplished through an input event multiplexer and is controlled by the INPSEL bit field in the CTCR_WT_j or CTCR_WOT_j register.

The SCTM supports a maximum of 127 input events. The details of the input events for a particular device are detailed in a device-specific appendix to this specification. Application and debug software can determine the number of counters in the SCTM by reading the NUMINPT field in the SCTM_CTCNTL register.

8.1.6.2.2.2 Counters

The individual 32-bit counters in the SCTM count when the input event signals are asserted. The SCTM supports a maximum of 32 counters. The counter configuration for a particular device is detailed in a device-specific appendix to this specification. Application and debug software can determine the number of counters in the SCTM by reading the NUMCNTR field in the SCTM_CTCNTL register.

8.1.6.2.2.3 Counting Mode

Counters function in one of two mutually exclusive counting modes:

- Event mode – The counter increments each time a rising edge is detected on the designated input event signal.
- Duration mode – The counter continually increments when the event input is asserted.

The DURMOD bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register controls operating mode of the counter.

8.1.6.2.2.4 Counter Overflow

When the counter reaches the terminal value (FFFF FFFFh) it wraps and continues to increment. This condition is considered a timer overflow condition and the OVRFLW bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register indicates that overflow has occurred. The overflow bit can be cleared by reading the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register. When chained, only the high-order counter (bits 63:32) can overflow.

8.1.6.2.2.5 Counters and Processor State

The counters can be configured to alter their behavior based on the state of the CPU in the subsystem supported by the SCTM. The FREE bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register determines if the counter continues to operate when the processor enters the debug-halt state. When this bit is 0 (the default), the counter stops incrementing while the debug-halt input from the CPU is asserted. Normal operation resumes when the processor exits the debug halt state and the debug-halt input is deasserted. When the FREE bit is 1, the state of the debug-halt input is not used to control counter operation.

The IDLE bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register determines if the counter continues to operate when the processor enters the IDLE state (the processor is no longer executing instructions and is waiting for a wakeup event). When this bit is 0 (the default), the counter stops incrementing while the IDLE state input from the CPU is asserted. Normal operation resumes when the processor exits the idle state and the idle state INPUT is deasserted. When the IDLE bit is 1, the state of the idle input does not control counter operation.

Not all subsystems may contain a processor or the subsystem processor may not support debug halt and/or idle modes. In this case, the processor state inputs to the SCTM are tied to the inactive value.

When there are multiple CPUs in a subsystem, the processor state inputs can be sourced from a single processor, or they can be the logical OR of the processor state signals from all CPUs in the subsystem.

8.1.6.2.2.6 Chaining Counters

The individual 32-bit counters in the SCTM can be chained with an adjacent counter to form a 64-bit counter. Counters can be chained to a counter across an even-odd index boundary with the even counter containing the least-significant 32-bits of the 64-bit pairing. For example, when counters 1 and 0 are paired, counter 1 contains bits 63:32 and counter 0 contains bits 31:00. The higher-order counter increments by 1 each time the lower order counter wraps.

Counters are chained by setting the CHAIN bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register for both of the counters. When chained, the counter control for both counters is taken from the lower-order

SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register. Other than CHAIN, all the other bits in the higher-order SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register are ignored.

Chained counters can function in counter mode only. Timer mode is not supported.

8.1.6.2.2.6.1 Reading Chained Counters

Some applications may require atomic reads of the 64 bits of a chained counter pair. An SCTM instance can be configured at synthesis time to provide an atomic read capability for one or more 64-bit counter pairs. When this capability is present, the designated counter pair provides atomic access when the correct read sequence is performed.

So when a chained counter has atomic read capability, an atomic counter value can be obtained by first reading the high-order counter followed by the low-order counter. Always observe this ordering, to prevent reading stale counter values from the low-order counter.

The shadow feature for chained counters is active only when the low-order counter is chained to a high-order counter. When functioning independently, the shadow feature is deactivated and a read of the counter always returns the current value.

To prevent sync errors introduced by the debugger, only application accesses activate the shadow feature. Debug qualified accesses always return the current value of the low-order counter, regardless of configuration.

8.1.6.2.2.7 Enabling and Disabling Counters

After the counter is correctly configured, the counter can be started by setting the ENBL bit in the corresponding SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register . At this point, the counter begins incrementing under the control of the configured event input. The counter can be disabled (counting stops) at any time by clearing ENBL. Counters can be enabled and disabled dynamically during application flow.

Counters can also be enabled and disabled as groups through the SCTM_CTGNBL0 registers. These registers provide control of the individual counter enables in groups of 32. This allows an application to enable or disable groups of counter in lockstep.

8.1.6.2.2.8 Resetting Counters

The counters can be reset to their initial value (0000 0000h) by setting the RESET bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register to 1 . If the counter is chained, both the high-order and low-order counters are reset when the RESET bit is set in the CTCR*n* for the low-order counter.

Counters can also be reset as groups through the SCTM_CTGRSTL0 register. These registers provide control of the individual counter reset in groups of 32. This control letss an application reset groups of counters in lockstep.

8.1.6.2.3 Timer Function

A subset of the counters modules in the SCTM can function as timers. When operating as timers, interrupts and/or debug input events are generated when the counter value reaches a designated interval. The counters with timer capabilities always occupy the lowest indexed counters in the SCTM. A maximum of eight counters can function as timers in SCTM v1. The actual number for a particular implementation of the SCTM is configured at synthesis time. The number of timers in a particular device is detailed in a device-specific appendix to this specification. Application and debug software can determine the total number of counters with timer capabilities reading the NUMTIMR field in the SCTM_CTCNTL register.

8.1.6.2.3.1 Periodic Intervals

The interval for a timer is contained in the SCTM_TINTVLR_i register. There is a SCTM_TINTVLR_i register for every timer capable counter in the SCTM. Timers are initialized to 0. When the corresponding CTCNTRn increments and matches the values designated in SCTM_TINTVLR_i, the timer is considered to be triggered and events configured in SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j are generated.

Timers can function in one of two mutually exclusive modes:

- Run once mode – The timer stops after the first interval match and is not re-enabled until the timer is reset to the initial value (0) by setting the RESET bit in the SCTM_CTCR_WT_j and SCTM_CTCR_WOT_j register to 1.
- Restart mode – The timer automatically resets to the initial value (0) each time the designated interval is reached.

8.1.6.2.3.2 Event Generation

Timers are able to generate both interrupts and debug events. Interrupts are routed from the module boundary to INTC in the subsystem. Debug events are routed as triggers to debug logic within the subsystem. It is assumed that these debug trigger events are merged with other debug resources and used to control debug activity like CPU control or trace.

The generation of the interrupts is controlled by the INT bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register. The generation of debug events is controlled by the DBG in the CTCR*n*. Both INT and DBG can be set simultaneously and both signals are generated on interval match.

For debug events, the pulse width is one cycle and the active polarity is high (a high going pulse of one cycle). For interrupts, the polarity and duration of the pulse are synthesizable options that can be tailored to the requirements of the INTC in the subsystem.

If neither the INT nor the DBG bit is set, the timer function is disabled and the counter functions as a regular counter.

The SCTM events include low-level stall and duration signals sourced by various components of the EVE subsystem. The primary categories include program cache-related signals and VCOP-related signals. Table 8-267 summarized the event mapping. Event inputs start numbering at 1, because event0 is reserved and the SCTM internally uses the functional clock as event 0.

The SCTM module operates at the CLK2 rate (nominally 250 to 300 MHz). The event sources include CLK2 and CLK1 (CLK1 = 2 × CLK2) signals. The CLK2 relative signals are connected directly to the SCTM. The CLK1 relative signals are conditioned by EVE level logic to scale from CLK1 to CLK2. Because these are duration type signals, EVE logic asserts a CLK2 pulse for every two CLK1 pulses detected. This results in, at most, one CLK1 cycle of inaccuracy in the CLK2 duration reported by SCTM.

Table 8-267. List of SCTM Events

SCTM Event	Name	Source	Type	SCTM Mode	Clock
1	cache_miss_count	ARP32_Pcache	Pulse	Duration	CLK2
2	cache_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
3	cache_miss_stall	ARP32_Pcache	Duration	Duration or event	CLK2
4	prefetch_compulsory_count	ARP32_Pcache	Pulse	Duration	CLK2
5	Prefetch_lookahead_count	ARP32_Pcache	Pulse	Duration	CLK2
6	prefetch_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
7	line_buffer_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
8	Prefetch_line_count	ARP32_Pcache	Pulse	Duration	CLK2
9	prefetch_discard_stall	ARP32_Pcache	Duration	Duration or event	CLK2
10	tpcc_aet	EDMA	Duration	Duration or event	CLK2
11	arp32_int4	INTC	Duration	Duration or event	CLK2
12	arp32_int5	INTC	Duration	Duration or event	CLK2
13	arp32_int6	INTC	Duration	Duration or event	CLK2
14	arp32_int7	INTC	Duration	Duration or event	CLK2
15	vcop_busy	VCOP	Pulse	Duration	CLK2
16	vcop_idle_and_done	VCOP	Pulse	Duration	CLK2
17	vcop_wait_for_arp32	VCOP	Pulse	Duration	CLK2
18	vcop_arp32_awaits	VCOP	Pulse	Duration	CLK2
19	vcop_overhead	VCOP	Pulse	Duration	CLK1

Table 8-267. List of SCTM Events (continued)

SCTM Event	Name	Source	Type	SCTM Mode	Clock
20	vcop_ld_stall_by_st	VCOP	Pulse	Duration	CLK1
21	vcop_op_stall_by_ldst	VCOP	Pulse	Duration	CLK1
22	vcop_op_stall_by_dependency	VCOP	Pulse	Duration	CLK1
23	vcop_rd_ibufl	VCOP	Pulse	Duration	CLK1
24	vcop_rd_ibufh	VCOP	Pulse	Duration	CLK1
25	vcop_rd_wbuf	VCOP	Pulse	Duration	CLK1
26	vcop_wr_ibufl	VCOP	Pulse	Duration	CLK1
27	vcop_wr_ibufh	VCOP	Pulse	Duration	CLK1
28	vcop_wr_wbuf	VCOP	Pulse	Duration	CLK1
29	vcop_loop_start	VCOP	Edge	Event	CLK2
30	vcop_done	VCOP	Edge	Event	CLK2
31	arp32_nmi	INTC	Duration	Duration or event	CLK2
32	arp32_int8	INTC	Duration	Duration of event	CLK2
33	arp32_int9	INTC	Duration	Duration of event	CLK2
34	arp32_int10	INTC	Duration	Duration of event	CLK2
35	arp32_int11	INTC	Duration	Duration of event	CLK2
36	arp32_int12	INTC	Duration	Duration of event	CLK2
37	arp32_int13	INTC	Duration	Duration of event	CLK2
38	arp32_int14	INTC	Duration	Duration of event	CLK2
39	arp32_int15	INTC	Duration	Duration of event	CLK2

8.1.6.2.3.3 Watchdog Timer Function

A normally configured timer supports a software watchdog capability in which software can continually reset the timer by writing to the SCTM_CTCR_j[1] RESET bit or to the corresponding bit in the SCTM_CTGRST0 register before the interval expires (and the interrupt is generated).

The timers also support a hardware watchdog function where selected input events can start and reset the timers without software interaction.

When hardware watchdog mode is enabled, the SCTM_CTCR_WT/WOT_j[23:16] INPSEL bit field selects the event that starts a WD timer. Once the input event is selected, the timer increments using the functional clock. If it reaches the interval match value before being reset, the interrupt is generated.

The WD timer is reset when an input event is detected. This resets the counter to 0. The counter does not restart until another input event is detected. The state-machine shown in Figure 8-22 illustrated the functional operation of the timer when configured for hardware watchdog mode.

Some important notes about WD timer mode operation:

- The INPSEL field selects the WD start event. This is always a rising edge event.
- The DURMODE bit is overridden. Once started, the timer continues to increment using the functional clock.

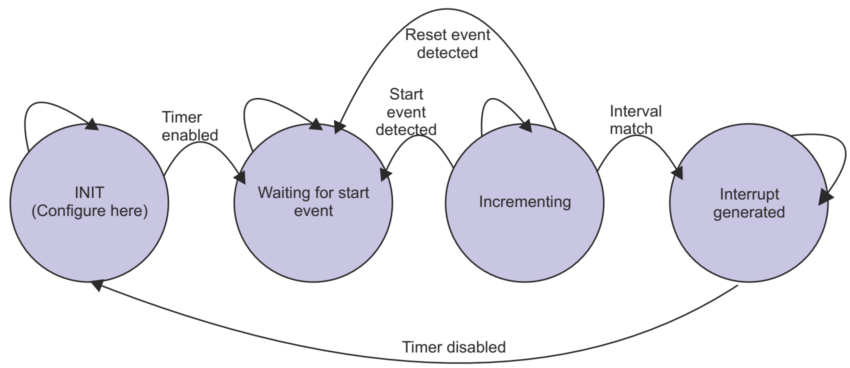


Figure 8-22. Watchdog Operation

8.1.6.2.4 System Trace Integration

8.1.6.2.4.1 Overview

Many devices have system-level trace capabilities. This capability is generally implemented in a top-level module called the STM. The STM collects software and hardware messages (the trace data) from various other system modules and subsystems through a dedicated STM bus (OCP compliant). The STM then formats these high-level trace messages into streams of MIPI® system trace protocol (STP) compliant messages. The STP messages are then transmitted to a dedicated STP receiver over a narrow (generally four pins), high-speed export interface.

The SCTM has an optional mode that provides an STM OCP interface and a mode where the state of the counters can be transferred through the STM.

Figure 8-23 shows a typical STM system configuration with the SCTM as a hardware master. The hardware master messages from the SCTM STM bus enter the L4 debug interconnect through an asynchronous bridge. Other system-level hardware masters are also writing messages to the STM through this interconnect. Software messages from applications running on the MCU or DSP to reach the STM message buffer through a bridge from L3 interconnect to the L4 interconnect. The STM orders and formats the high-level hardware and software messages into STP protocol messages and exports them to an off-chip trace receiver.

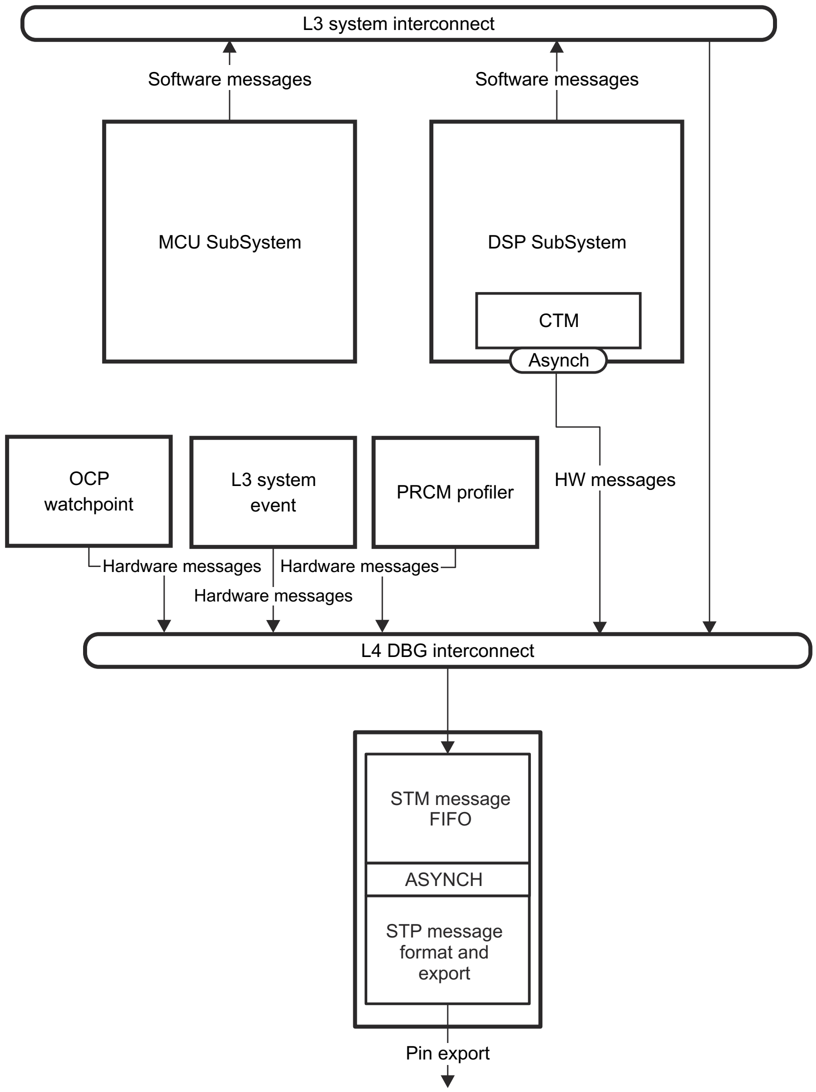


Figure 8-23. Typical STM System Integration

8.1.6.2.4.2 STM Configuration

Because the STM function of the SCTM is an optional feature, the NUMSTM bit in the SCTM_CTCCNTL register is read to determine if any counters are available for STM export:

- If the value in NUMSTM is 0, there is no STM capability in the module.
- If the value is greater than 0 (N ≤ the total counters in the module), then the first N counters can also be designated for STM export.

The STM capability of the SCTM is enabled by setting the ENBL bit in the SCTM_CTSTMCNTL register to 1.

8.1.6.2.4.2.1 Periodic Counter State Export

The general operating mode of the SCTM STM function is to periodically export a message frame that provides a snapshot of the state of each counter designated for inclusion in the STM counter state message. The length of the snapshot interval is controlled by a programmable counter. The interval for this counter is determined by the value in the SCTM_CTSTMINTVL_i register. A periodic counter is reloaded with this value. The counter decrements using the functional clock of the SCTM. When the counter decrements to 0, a snapshot of all the designated counters is taken for STM export and the counter is reloaded using the value in CTSTMINTVL. If the value in CTSTMINTVL is 0, periodic message export is disabled.

Because an entire snapshot of data must be captured when an interval expires, the SCTM must have temporary storage for all the counter values that are designated for export.

8.1.6.2.4.2.2 Application Control of Counter State Export

Application software can also independently trigger the export of a CSM frame bsetting the CSMXPORT bit in the CTSTMCNTL register to 1.

- When the bit is set to 1, a snapshot is captured and a CSM frame is exported.
- Software-triggered export is available only when periodic export is not active; writing 1 when periodic export is active has no effect.
- The SW bit in the CSM header designates that this frame is generated by software.

8.1.6.2.4.2.3 Application Control of the Counter Configuration Export

The STM system also supports an optional counter configuration message (CCM). The capability for generating this message is a synthesizable option. If the module has been synthesized to support CCM, the SCTM_CTSTMCNTL[3] CCMAVAIL bit is asserted.

Export of the CCM message frame is only triggered by setting the CCMXPORT bit in the CTSTMCNTL register to 1.

- When 1 is written, a snapshot is captured and a CCM frame is exported.
- Software-triggered export is available only when periodic export is not active; writing 1 when periodic export is active has no effect.

8.1.6.3 Use Case Examples

This section contains high-level examples of the programming sequences for common SCTM user scenarios. The following examples assume that the counter timer module is already enabled by setting the ENBL bit in the SCTM_CTCNTL register to 1. The examples also assume that the CTCR*n*:ENBL bit is cleared before any configuration writes other than RESET.

8.1.6.3.1 Counter Enable

8.1.6.3.1.1 Enabling a Single Counter

Perform the following procedure to enable a single counter:

1. Reset the counter by setting the SCTM_CTCR_WOT_j[1] RESET bit to 1.
2. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WOT_j[23:16]INPSEL bit field
3. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WOT_j [3]DURMODE bit (if required).
4. Set the behavior for system state by writing to the SCTM_CTCR_WOT_j[5]IDLE and SCTM_CTCR_WOT_j[4] FREE bits (if required).
5. Start the counter function by setting the SCTM_CTCR_WOT_j[0] ENBL bit to 1.

8.1.6.3.1.2 Reading a Single Counter

Perform the following procedure to read a single counter:

1. Read the value of the counter through the CTCNTR_k register.
2. Read the corresponding SCTM_CTCR_WOT_j[6] OVRFLW bit to determine if the counter has wrapped since the last read of the registers.
3. (Optional) The counter can then be reset by setting the SCTM_CTCR_WOT_j[1] RESET bit to 1.

8.1.6.3.1.3 Enabling a Group of Counters Simultaneously

Perform the following procedure to enable a group of counters:

1. For each counter in the group:

   a. Enable access to the SCTM by setting the SCTM_CTCNTL[0] ENBL bit to 1.
   b. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WOT_j [23:16] INPSEL bit field.
   c. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WOT_j [3] DURMODE bit (if required).
   d. Set the behavior for system state by writing to the SCTM_CTCR_WOT_j [5] IDLE and [4]FREE bits (if required).
2. Start all counters in lockstep by writing to the CTGNBL0 register for all counters that are set to 1 in the group.

8.1.6.3.1.4 Reading a Group of Counters Simultaneously

Perform the following procedure to read a group of counters:

1. Read the SCTM_CTGNBL0 register and save.
2. Clear the corresponding enable bit of each counter in the group to 0.
3. Write back the SCTM_CTGNBL0 register content.
4. For each counter in the group:
   a. Read the value of the counter through the SCTM_CTCNTR_k register.
   b. Read the corresponding SCTM_CTCR_WOT_j[6] OVRFLW bit to determine if the counter has wrapped since the last read of the SCTM_CTCR_WOT_j register.
5. (Optional) Resume the group count by writing back the saved value of the SCTM_CTGNBL0 register.

8.1.6.3.1.5 Configuring a Chained Counter

Perform the following procedure to configure a chained counter:

1. Select the counter pair to be used for chaining (N and N + 1).
2. Set the SCTM_CTCR_WOT_j [2]CHAIN bit of counter N + 1 to 1.
3. For the SCTM_CTCR_WOT_j register for counter N in the counter pair:
   a. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WOT_j[23:16]INPSEL bit field.
   b. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WOT_j [3]DURMODE bit (if required).
   c. Set the behavior for system state by writing to the SCTM_CTCR_WOT_j [5]IDLE and SCTM_CTCR_WOT_j [4]FREE bits (if required).
   d. Start the counter function by setting the SCTM_CTCR_WOT_j[0] ENBL bit to 1.

8.1.6.3.2 Timer Enable

This example assumes interrupt generation capabilities. Perform the following procedure to enable the timer:

1. Disable interrupts globally using the CPU INTM mask or some other global interrupt masking in the target subsystem.
2. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WT_j[23:16] INPSEL bit field.
3. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WT_j[3] DURMODE bit (if required).
4. Set the behavior for system state by writing to the SCTM_CTCR_WT_j [5]IDLE and [4] FREE bits (if required).
5. If the timer function is continuous, set the SCTM_CTCR_j[10] RESTART bit to 1.
6. Set the SCTM_CTCR_WT_j[8] INT bit to 1.
7. Set the interval match value in the corresponding SCTM_TINTVLR_i register.
8. Enable the corresponding interrupt in the subsystem INTC.
9. Enable interrupts globally.
10. Start the counter function by setting the SCTM_CTCR_WT_j[0] ENBL bit to 1.

8.1.6.3.3 Periodic STM Export Enable

Perform the following procedure to enable periodic STM export:

1. Configure, but do not enable, all the counters required for export per Section 8.1.6.3.1, Counter Enable.
2. Enable the STM configuration capability by setting the SCTM_CTSTMCNTL[0] ENBL bit to 1.
3. Disable periodic export by writing 0000 0000h to the SCTM_CTSTMINTVL register.
4. (Optional) Change the hardware master ID by writing the new value to the SCTM_CTSTMMSTID[6:0] MASTID bit.
5. Tag which counters are selected for export by setting the corresponding bits.
6. Set the total number of counters tagged for export through a write to the SCTM_CTSTMCNTL[11:6] NUMXPORT bit.
7. (Optional) To include overflow information in the CSM message, set the SCTM_CTSTMCNTL[1] SENDOVR bit to 1.
8. (Optional) To send a counter configuration message (CCM), set the SCTM_CTSTMCNTL[4] CCMXPORT bit to 1.
   a. This feature is only available if the SCTM_CTSTMCNTL[3]CCMAVAIL bit reads as 1.
   b. Test for completion of the CCM export through the SCTM_CTSTMCNTL[5] XPORTACT bit.
9. Enable periodic export by writing the interval value to the SCTM_CTSTMINTVL register.
10. Enable the counters per Section 8.1.6.3.1, *Counter Enable*.

8.1.6.3.4 Disabling the SCTM

Perform the following procedure to disable the SCTM:

1. When periodic STM export is enabled:
   a. Disable periodic export by writing 0000 0000h to the SCTM_CTSTMINTVL register.
   b. Test for completion of the CCM export through the SCTM_CTSTMCNTL[5] XPORTACT bit.
   c. Disable the STM function by writing 0 to the SCTM_CTSTMCNTL[0]ENBL bit.
2. Disable all counters by setting the SCTM_CTGNBL0 register to 0000 0000h.
3. Disable the SCTM by setting the SCTM_CTCNTL[0] ENBL bit to 0.

8.1.6.4 SCTM Register Manual

8.1.6.4.1 SCTM Instance Summary

Table 8-268. EVE_SCTM Instance Summary

Module Name	Physical Address	Size
EVE1_SCTM	0x4208 5000	512 bytes
EVE2_SCTM	0x4218 5000	512 bytes

8.1.6.4.2 SCTM Registers

8.1.6.4.2.1 SCTM Register Summary

Table 8-269 lists the SCTM registers

Table 8-269. EVE_SCTM Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_SCTM Base Address	EVE2_SCTM Base Address
SCTM_CTCNTL	RW	32	0x0000 0000	0x4208 5000	0x4218 5000
SCTM_CTSTMCNTL	RW	32	0x0000 0020	0x4208 5020	0x4218 5020
SCTM_CTSTMMSTID	RW	32	0x0000 0024	0x4208 5024	0x4218 5024
SCTM_CTSTMINTVL	RW	32	0x0000 0028	0x4208 5028	0x4218 5028
SCTM_CTSTMSEL	RW	32	0x0000 002C	0x4208 502C	0x4218 502C
SCTM_TINTVLR_i (1)	RW	32	0x0000 0040 + (0x4*i)	0x4208 5040 + (0x4*i)	0x4218 5040 + (0x4*i)
SCTM_CTDBGNUM	R	32	0x0000 007C	0x4208 507C	0x4218 507C
SCTM_CTDBGEVT	RW	32	0x0000 0080	0x4208 5080	0x4218 5080
SCTM_CTGNBL	RW	32	0x0000 00F0	0x4208 50F0	0x4218 50F0
SCTM_CTGRST	RW	32	0x0000 00F8	0x4208 50F8	0x4218 50F8
SCTM_CTCR_WT_m (3)	RW	32	0x0000 0100 + (0x4*m)	0x4208 5100 + (0x4*m)	0x4218 5100 + (0x4*m)
SCTM_CTCR_WOT_n (4)	RW	32	0x0000 0108 + (0x4*n)	0x4208 5108 + (0x4*n)	0x4218 5108 + (0x4*n)
SCTM_CTCNTR_k (2)	R	32	0x0000 0180 + (0x4*k)	0x4208 5180 + (0x4*k)	0x4218 5180 + (0x4*k)

(1) i = 0 to 7 for EVE1_SCTM
i = 0 to 7 for EVE2_SCTM
(2) k = 0 to 7 for EVE1_SCTM
k = 0 to 7 for EVE2_SCTM
(3) m = 0 to 1 for EVE1_SCTM
m = 0 to 1 for EVE2_SCTM
(4) n = 0 to 5 for EVE1_SCTM
n = 0 to 5 for EVE2_SCTM

8.1.6.4.2.2 SCTM Register Description

Table 8-270. SCTM_CTCNTL

Address Offset	0x0000 0000		
Physical Address	0x4208 5000 0x4218 5000	**Instance**	EVE1_SCTM EVE2_SCTM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMSTM						NUMINPT								NUMTIMR					NUMCNTR						REVID				IDLEMODE		ENBL

Bits	Field Name	Description	Type	Reset
31:26	NUMSTM	Number of timers that can export through STM	R	0x0
25:18	NUMINPT	Number of event input signals	R	0x20
17:13	NUMTIMR	Number of timers in the module	R	0x2
12:7	NUMCNTR	Number of counters in the module	R	0x8
6:3	REVID	Revision ID of SCTM	R	0x1
2:1	IDLEMODE	Idle mode control 0x0: Force Idle mode 0x1: Ths SCTM will acknoledge the idle request, but never transition to the idle state 0x3: Since the SCTM does not support internal wakeup, this mode is identical to smart_idle 0x2: Ths SCTM uses the smart idle protocol. This is the default mode	RW	0x2
0	ENBL	SCTM global enable 0x0: DISABLE 0x1: ENABLE	RW	0x0

Table 8-271. Register Call Summary for Register SCTM_CTCNTL

Embedded Vision Engine (EVE) Subsystem

- Counter Function: [0] [1]
- Timer Function: [2]
- Use Case Examples: [3]
- Counter Enable: [4]
- Disabling the SCTM: [5]
- SCTM Registers: [6]

Table 8-272. SCTM_CTSTMCNTL

Address Offset	0x0000 0020		
Physical Address	0x4208 5020 0x4218 5020	**Instance**	EVE1_SCTM EVE2_SCTM
Description	This register contains the control and status settings for STM export		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					XPORTACT	NUMXPORT					CCMXPORT	CCMVAIL	CSMXPORT	SENDOVR	ENBL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	XPORTACT	Indicates if a frame is currently being written to the STM	R	0x0
9:5	NUMXPORT	The total number of counters designated for export. this will be used as the count in the CSM and CCM headers. The value written should be the total number of counters designated for export -1	RW	0x0
4	CCMXPORT	SW control of CCM message export	RW	0x0
3	CCMVAIL	SCTM supports CCM export	R	0x0
2	CSMXPORT	SW control of CSM message export	RW	0x0
1	SENDOVR	Send overflow data in CSM frame	RW	0x1

Bits	Field Name	Description	Type	Reset
0	ENBL	STM global enable 0x0: DISABLE 0x1: ENABLE	RW	0x0

Table 8-273. Register Call Summary for Register SCTM_CTSTMCNTL

Embedded Vision Engine (EVE) Subsystem

- System Trace Integration: [0] [1]
- Periodic STM Export Enable: [2] [3] [4] [5] [6] [7]
- Disabling the SCTM: [8] [9]
- SCTM Registers: [10]

Table 8-274. SCTM_CTSTMMSTID

Address Offset	0x0000 0024		
Physical Address	0x4208 5024 0x4218 5024	**Instance**	EVE1_SCTM EVE2_SCTM
Description			
Type	RW		

31:7	6:0
RESERVED	MASTID

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6:0	MASTID	HW Master ID for this module.	RW	0x0

Table 8-275. Register Call Summary for Register SCTM_CTSTMMSTID

Embedded Vision Engine (EVE) Subsystem

- Periodic STM Export Enable: [0]
- SCTM Registers: [1]

Table 8-276. SCTM_CTSTMINTVL

Address Offset	0x0000 0028		
Physical Address	0x4208 5028 0x4218 5028	**Instance**	EVE1_SCTM EVE2_SCTM
Description			
Type	RW		

31:16	15:0
RESERVED	INTERVAL

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	INTERVAL	Periodic export interval	RW	0x0

Table 8-277. Register Call Summary for Register SCTM_CTSTMINTVL

Embedded Vision Engine (EVE) Subsystem

- Periodic STM Export Enable: [0] [1]
- Disabling the SCTM: [2]
- SCTM Registers: [3]

Table 8-278. SCTM_CTSTMSEL

Address Offset	0x0000 002C		
Physical Address	0x4208 502C 0x4218 502C	**Instance**	EVE1_SCTM EVE2_SCTM
Description	These registers mark the counters selected for export in the CSM		
Type	RW		

31–0
COUNTSEL

Bits	Field Name	Description	Type	Reset
31:0	COUNTSEL	The counter selection bit field	RW	0x0

Table 8-279. Register Call Summary for Register SCTM_CTSTMSEL

Embedded Vision Engine (EVE) Subsystem

- SCTM Registers: [0]

Table 8-280. SCTM_TINTVLR_i

Address Offset	0x0000 0040		
Physical Address	0x4208 5040 + (0x4*i) 0x4218 5040 + (0x4*i)	**Instance**	EVE1_SCTM EVE2_SCTM
Description	These registers contain the interval match value for the corresponding timers in the SCTM		
Type	RW		

31–0
INTERVAL

Bits	Field Name	Description	Type	Reset
31:0	INTERVAL	Interval match value for the timers in the SCTM	RW	0x0

Table 8-281. Register Call Summary for Register SCTM_TINTVLR_i

Embedded Vision Engine (EVE) Subsystem

- Timer Function: [0] [1] [2]
- Timer Enable: [3]
- SCTM Registers: [4]

Table 8-282. SCTM_CTDBGNUM

Address Offset	0x0000 007C		
Physical Address	0x4208 507C 0x4218 507C	**Instance**	EVE1_SCTM EVE2_SCTM
Description	Counter Timer Number Debug Event Register		
Type	R		

31–3	2–0
RESERVED	NUMEVT

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	NUMEVT	Number of input selectors for debug events	R	0x0

Table 8-283. Register Call Summary for Register SCTM_CTDBGNUM

Embedded Vision Engine (EVE) Subsystem

- SCTM Registers: [0]

Table 8-284. SCTM_CTDBGEVT

Address Offset	0x0000 0080		
Physical Address	0x4208 5080 0x4218 5080	**Instance**	EVE1_SCTM EVE2_SCTM
Description	Counter Timer Debug Event Register		
Type	RW		

31-8	7-0
RESERVED	INPSEL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	INPSEL	Index of event input signal on the module boundary	RW	0x0

Table 8-285. Register Call Summary for Register SCTM_CTDBGEVT

Embedded Vision Engine (EVE) Subsystem

- SCTM Registers: [0]

Table 8-286. SCTM_CTGNBL

Address Offset	0x0000 00F0		
Physical Address	0x4208 50F0 0x4218 50F0	**Instance**	EVE1_SCTM EVE2_SCTM
Description	These registers provide for simultaneous enable/disable of 32 counters		
Type	RW		

31-8	7-0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ENABLE	The counter enable bit field	RW	0x0

Table 8-287. Register Call Summary for Register SCTM_CTGNBL

Embedded Vision Engine (EVE) Subsystem

- Counter Function: [0]
- Counter Enable: [1] [2] [3]
- Disabling the SCTM: [4]
- SCTM Registers: [5]

Table 8-288. SCTM_CTGRST

Address Offset	0x0000 00F8		
Physical Address	0x4208 50F8 0x4218 50F8	**Instance**	EVE1_SCTM EVE2_SCTM
Description	These registers provide for simultaneous reset of 32 counters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESET

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RESET	The counter reset bit field	RW	0x0

Table 8-289. Register Call Summary for Register SCTM_CTGRST

Embedded Vision Engine (EVE) Subsystem

- Timer Function: [0]
- SCTM Registers: [1]

Table 8-290. SCTM_CTCR_WT_m

Address Offset	0x0000 0100		
Physical Address	0x4208 5100 + (0x4*m) 0x4218 5100 + (0x4*m)	**Instance**	EVE1_SCTM EVE2_SCTM
Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WT: with timer)		
Type	RW		

31:21	20:16	15:11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	INPSEL	RESERVED	RESTART	DBG	INT	RESERVED	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20:16	INPSEL	Counter Timer input selection	RW	0x0
15:11	RESERVED		R	0x0
10	RESTART	Restart the timer after an interval match	RW	0x0
9	DBG	Signal debug logic on interval match	RW	0x0
8	INT	Generate interrupt on interval match	RW	0x0
7	RESERVED		R	0x0
6	OVRFLW	Counter has wrapped since it was last read	R	0x0
5	IDLE	Counter ignores processor IDLE state	RW	0x0
4	FREE	Counter ignores processor debug halt state	RW	0x0
3	DURMODE	Counter is in duration or occurrence mode	RW	0x0
2	CHAIN	Counter is chained to an adjacent counter	RW	0x0
1	RESET	Counter reset control	RW	0x0
0	ENBL	Counter enable control	RW	0x0

Table 8-291. Register Call Summary for Register SCTM_CTCR_WT_m

Embedded Vision Engine (EVE) Subsystem

- SCTM Registers: [0]

Table 8-292. SCTM_CTCR_WOT_n

Address Offset	0x0000 0108		
Physical Address	0x4208 5108 + (0x4*n) 0x4218 5108 + (0x4*n)	**Instance**	EVE1_SCTM EVE2_SCTM

Table 8-292. SCTM_CTCR_WOT_n (continued)

Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WOT: without timer)
Type	RW

31:21	20:16	15:11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	INPSEL	RESERVED	RESTART	DBG	INT	RESERVED	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20:16	INPSEL	Counter Timer input selection	RW	0x0
15:11	RESERVED		R	0x0
10	RESTART	Restart the timer after an interval match	RW	0x0
9	DBG	Signal debug logic on interval match	RW	0x0
8	INT	Generate interrupt on interval match	RW	0x0
7	RESERVED		R	0x0
6	OVRFLW	Counter has wrapped since it was last read	R	0x0
5	IDLE	Counter ignores processor IDLE state	RW	0x0
4	FREE	Counter ignores processor debug halt state	RW	0x0
3	DURMODE	Counter is in duration or occurrence mode	RW	0x0
2	CHAIN	Counter is chained to an adjacent counter	RW	0x0
1	RESET	Counter reset control	RW	0x0
0	ENBL	Counter enable control	RW	0x0

Table 8-293. Register Call Summary for Register SCTM_CTCR_WOT_n

Embedded Vision Engine (EVE) Subsystem

- SCTM Registers: [0]

Table 8-294. SCTM_CTCNTR_k

Address Offset	0x0000 0180		
Physical Address	0x4208 5180 + (0x4*k) 0x4218 5180 + (0x4*k)	**Instance**	EVE1_SCTM EVE2_SCTM
Description	These registers contain the value of an individual counter in the moduel. There will be a CTCNTR for every counter in the module		
Type	R		

31:0
COUNT

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Counter value	R	0x0

Table 8-295. Register Call Summary for Register SCTM_CTCNTR_k

Embedded Vision Engine (EVE) Subsystem

- Counter Enable: [0]
- SCTM Registers: [1]

8.1.7 Software Message and System Event Trace

8.1.7.1 Introduction

8.1.7.1.1 Overview

The SMSET is a trace module used for monitoring key system events and transferring software messages. SMSET is performing arbitration between hardware messages and software messages written to the trace receiver through STM by the host processor. SMSET module provides various debug features such as: monitor system events, enable/disable event detection, Start/Stop event monitoring upon external trigger, time sampling, generating hardware message upon system event or every sampling window boundary and interleaving of hardware and software messages.

8.1.7.1.2 Configuration

Table 8-296 lists the configuration of the SMSET in the EVE subsystem.

Table 8-296. SMSET Configuration

Generic	SMSET Feature	Number on Device
NB_EVENTS	Number of system events	9
SW_MESSAGE	Software messages support	1
TRIG_MIN_WIDTH	Minimum trigger input pulse (L4 cycles)	2
EVENTS_BUF_DEPTH	System events buffer depth	4
SW_BUF_DEPTH	Software message buffer depth	2

8.1.7.1.3 Block Diagram

The block diagram shown in Figure 8-24 is a generic example to show the main concept of SMSET. Software messages are initiated by processors. Basically, a processor is writing a message to the trace receiver through STM. SMSET is performing arbitration between hardware messages and software messages.

Software messages and system event messages can be interleaved. Software messages do not require a sequence of write accesses; therefore, the SMSET is can switch to the system event detection as soon as the event is detected.

In case of conflicting accesses between software message and detection of system events, the SMSET can switch to the system event detection as soon as the event is detected. Software messages are never lost. Therefore, an SMSET buffer allows absorbing instrumentation accesses peaks, which prevents stalling the processor originating the message.

In case of a platform implementing several SMSET instances, the priority is arbitrated by the instrumentation NoC at SoC level.

It is possible to program the sampling window size to exchange STM bandwidth for time-stamping accuracy. All the events detected within the sampling window are considered close enough and packed in the same message as concurrent events and therefore reported in the trace log with the same time stamp. In case a selected event has been captured but not yet written to the STM queue, the system event trace module shall not wait for the sampling window boundary to write the pending packet to the STM. This results in two separate messages reporting the first and second pulse of the selected event. System events are considered synchronous to the SMSET functional clock. The sampling window size granularity is one SMSET cycle. The size ranges from 1 to 256 cycles.

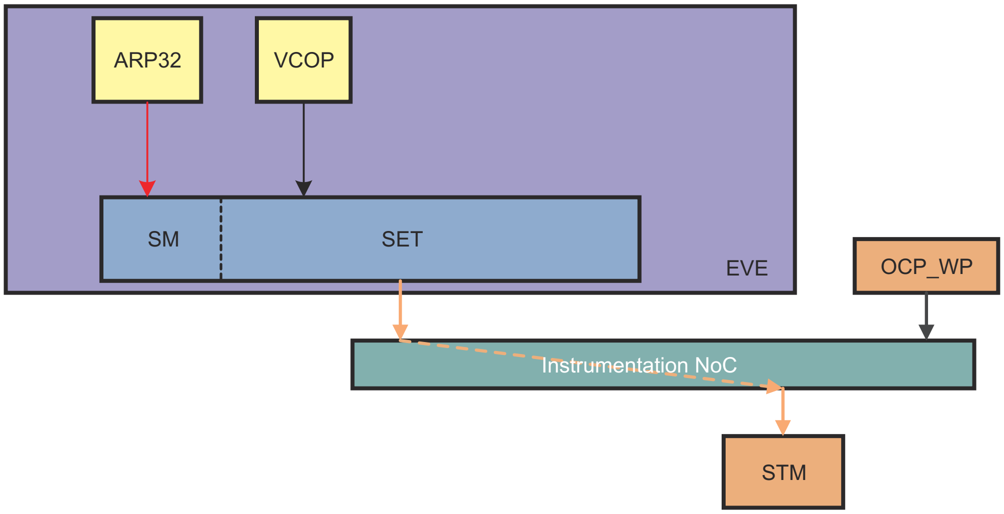


Figure 8-24. SMSET Block Diagram

8.1.7.2 Functional Description

8.1.7.2.1 Connectivity

The SMSET component is interfaced to the following:

- Configuration OCP slave port to configure the SMSET parameters
- System events [x N] for monitoring
- Processors supporting software messages (through OCP interface)
- Instrumentation OCP master port to write hardware and software messages into the STM queue

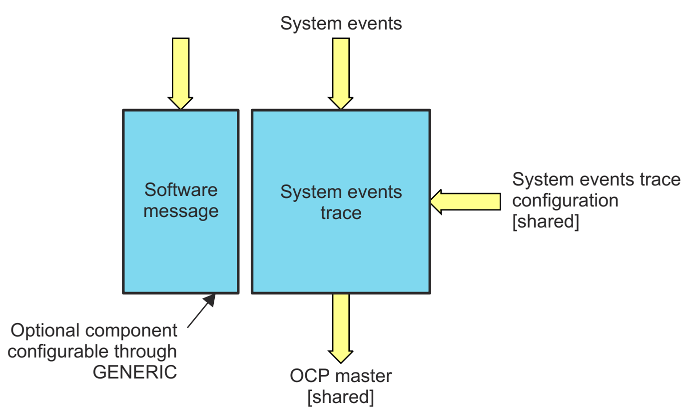


Figure 8-25. SMSET Interfaces

8.1.7.2.2 SMSET Event Mapping

The SMSET events include higher-level start and end task events. Table 8-297 summarize the event mapping.

The SMSET module operates at the CLK2 rate.

SMSET traces events on either a high-to-low transition or low-to-high transition, but not both. Thus, the EDMA's tpcc_aet signal is not directly compatible with SMSET tracing. The tpcc_aet is an active high signal that can be used to measure the duration of a specific edma transfer. The tpcc_aet signal can be programmed (in EDMA CC MMRs) to go high when a specific trigger is detected, and go low when a specific completion code is received. EVE logic will implement a custom tpcc_aet_start and tpcc_aet_end signals. The tpcc_aet_start signal will pulse high (for one cycle) when tpcc_aet transitions to high, and tpcc_aet_end signal will pulse high (for one cycle) when tpcc_aet_end signal when tpcc_aet transitions to low.

Table 8-297 details the various profiling signals.

The “pulse” signals are expected to be driven active one cycle for each occurrence. A sequence of consecutive active cycles represent multiple occurrences. "Duration" mode of the SMSET makes sense for this type of signal where the duration implies the number of occupancies in a particular state (for example the number of cache hits)

The “duration” type represents a signal that may stay active for multiple cycles on a given occurrence. These are typically stall signals where the user is attempting to determine the total number of cycles in the “stalled” state. “Duration” mode will indicate the total time stalled. “Event” mode will indicate the number of unique occurrences of stalls. “Event” is useful in this case since there will always be a deassertion between occurrences.

“Edge” type goes active for undefined period of time but is guaranteed to go inactive before the next occurrence. “Event” mode of the SCTM should be used for these signals.

Table 8-297. List of SMSET Events

SMSET Event	Name	Source	Type	SMSET Mode	Clock
0	tpcc_aet_start	EDMA	Pulse	Duration	CLK2
1	tpcc_aet_stop	EDMA	Pulse	Duration	CLK2
2	arp32_int4	INTC	Duration	Duration or event	CLK2
3	arp32_int5	INTC	Duration	Duration or event	CLK2
4	arp32_int6	INTC	Duration	Duration or event	CLK2
5	arp32_int7	INTC	Duration	Duration or event	CLK2
6	vcop_loop_start	VCOP	Edge	Event	CLK2
7	vcop_done	VCOP	Edge	Event	CLK2
8	arp32_nmi	INTC	Duration	Duration or event	CLK2
9	arp32_int8	INTC	Duration	Duration or event	CLK2
10	arp32_int9	INTC	Duration	Duration or event	CLK2
11	arp32_int10	INTC	Duration	Duration or event	CLK2
12	arp32_int11	INTC	Duration	Duration or event	CLK2
13	arp32_int12	INTC	Duration	Duration or event	CLK2
14	arp32_int13	INTC	Duration	Duration or event	CLK2
15	arp32_int14	INTC	Duration	Duration or event	CLK2
16	arp32_int15	INTC	Duration	Duration or event	CLK2

8.1.7.2.3 Software Messages

A GENERIC shall specify if software messages generated to STM through SMSET is supported. Software messages are initiated by processors. Basically a processor is writing a message to the trace receiver through STM. SMSET is just performing arbitration between hardware messages and software messages. Most of the subsystems cannot afford to have two OCP master ports for debug purpose. Interleaving is done in SMSET module.

8.1.7.2.4 SMSET Master Port

The SMSET master port is connected to an instrumentation OCP master port to export hardware and software messages through the STM module. The write accesses to STM are arbitrated at the instrumentation OCP interconnect level.

The SMSET master port supports burst transactions to reduce trace export surplus on the PTI interface; this surplus is due to instrumentation flows interleaving at instrumentation NoC level.

8.1.7.2.4.1 OCP Disconnect

The SMSET is supporting disconnect protocol on its master OCP interface, to make sure that software message transactions always complete. If software messages are initiated, while the instrumentation interconnect is disconnected, the access completes with a DVA response, and written data are lost.

8.1.7.2.5 SMSET Debug Features

The SMSET component provides the following system debug capabilities:

- Monitor events.
- Enable and disable event detection.
- Start event monitoring upon external trigger.
- Stop event monitoring upon external trigger.
- Time stamping.
- Generate hardware message upon system event.
- Generate hardware message every sampling window boundary.
- Interleave hardware messages and software messages.
- Program SMSET from debugger.
- Program SMSET from application.

8.1.7.2.6 Component Ownership

Some of the SMSET resources are owned either by the application or by the debugger. The ownership is required to configure or program SMSET. In other words, ownership determines if write access is granted to the SMSET control registers. The SMSET resource ownership is exclusive. Hence, simultaneous use of SMSET resources by both the debugger and the application is not permitted. However, the debugger can forcibly seize ownership of SMSET resources.

Note

A read access does not require ownership; therefore, either the debugger or the application can read any SMSET registers with or without ownership.

8.1.7.2.6.1 Ownership State

The ownership has three basic states:

- AVAILABLE: The unit has not been claimed.
- CLAIMED: The unit has been claimed.
- ENABLED: The unit is enabled by the owning party.

8.1.7.2.6.1.1 Available State

An SMSET is set to the AVAILABLE state when one of two conditions occurs:

- The state is available when power-on reset (POR) is asserted.
- The state is available if the SMSET is owned by the debugger and the emulator is disconnected.

Note

A warm reset has no effect on the operation of SMSET. An emulator disconnect has no effect when the application owns the unit.

8.1.7.2.6.1.2 Claimed State

An SMSET is in the CLAIMED state when either the debugger or the application has claimed it. The CLAIMED state provides exclusive access to the owner. The other party is not permitted write access to the unit until the owning party releases the claim. Once a unit is claimed, only the owner can write to the other resources controlled by the claim.

The debugger, however, is allowed to forcibly seize control of a unit, if it is already claimed by the application, by setting the DEBUGGEROVERRIDE bit to 1 in the SMSET configuration register (CFG).

8.1.7.2.6.1.3 Enabled State

An SMSET enters the ENABLED state when it is activated. The owner can continue writing to the other resources while SMSET is enabled.

8.1.7.2.6.2 Ownership Commands

Ownership of the SMSET resources is managed through the OWNERSHIP bit in the SMSET configuration register (CFG). Table 8-298 lists the ownership commands.

Table 8-298. Ownership Commands

OWNERSHIP Bits	Meaning	Description
00	Release ownership	Release ownership of system events trace. The Release command is accepted only from the owner.
01	Claim ownership	Claim access to system events trace. The claim command is successful only if the unit is available, or the requester is the debugger and the DEBUGGEROVERRIDE bit is high.
10	Enable unit	Activate system events trace for use. The enable command is accepted only from the owner.
11	No Operation	The NOP command does not affect ownership or claim state.

8.1.7.2.6.3 Claim Reset

When a resource is owned by the debugger, the resource is released if the debugger is disconnected. Debugger disconnection is accomplished by asserting PIDCON = 0.

If a resource is owned by the application and the processor enters the debug state, the resource continues to belong to the application. The SMSET is not sensitive to debug state.

If SMSET ownership is released while there are still data in the SMSET FIFO, the master port continues to export data to the STM until the FIFO is drained.

8.1.7.3 Use Case Examples

8.1.7.3.1 Procedure to Enable System Event Capture

Perform the following procedure to enable system event capture:

1. Claim the ownership by setting the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG) to 01b.
2. Check the ownership and the current owner by reading the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG).
3. Set the appropriate sampling window size value in the system event sampling window register (SMSET_SESW).
4. Enable the events needed to be detected by setting the appropriate bits in the programming of the system event detection enable registers (SMSET_SEDEN_i) to 1.
5. Select the polarity of the event detection by writing the appropriate value to the EventPolarity bit in the SMSET configuration register (SMSET_CFG).
6. Select the message event generation by writing the appropriate value to the MessageEventGeneration bit in the SMSET configuration register (SMSET_CFG).
7. Enable system event capture by setting the SystemEventCapture bit in the SMSET configuration register (SMSET_CFG) to 1.

8.1.7.3.2 Procedure to Start and Stop System Event Capture from External Trigger Detection

Perform the following procedure to start and stop system event capture from external trigger detection:

1. Claim the ownership by setting the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG) to 1.

2. Check the ownership and the current owner by reading the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG).
3. Set the appropriate sampling window size value in the system event sampling window register (SMSET_SESW).
4. Enable the events to be detected by setting the appropriate bits in the programming of the system event detection enable registers (SMSET_SEDEN_i) to 1.
5. Select the polarity of the event detection by writing the appropriate value to the EventPolarity bit in the SMSET configuration register (SMSET_CFG).
6. Select the message event generation by writing the appropriate value to the MessageEventGeneration bit in the SMSET configuration register (SMSET_CFG).
7. Enable capturing system events from external trigger detection by setting the StopCapturingSystemEvents bit in the SMSET configuration register to 1.
8. Enable capturing system events from external trigger detection by setting the StartCapturingSystemEvents bit in the SMSET configuration register to 1.
9. Enable system event capture by writing a 1 to the SystemEventCapture bit in the SMSET configuration register.

8.1.7.3.3 Procedure to Disable System Event Capture

Perform the following procedure to disable system event capture:

1. Disable system event capture by setting the SystemEventCapture bit in the SMSET configuration register (SMSET_CFG) to 0.
2. Release the ownership by setting the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG) to 0.
3. Check the ownership by reading the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG). It must be available (OWNERSHIP = 0).

8.1.7.4 SMSET Register Manual

8.1.7.4.1 SMSET Instance Summary

Table 8-299. SMSET Instance Summary

Module Name	Physical Address	Size
EVE1_SMSET	0x4208 8000	4k
EVE2_SMSET	0x4218 8000	4k

8.1.7.4.2 SMSET Register Summary

Table 8-300. EVE_SMSET Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_SMSET Base Address	EVE2_SMSET Base Address
SMSET_ID	R	32	0x0000 0000	0x4208 8000	0x4218 8000
SMSET_SCFG	RW	32	0x0000 0010	0x4208 8010	0x4218 8010
SMSET_SR	R	32	0x0000 0014	0x4208 8014	0x4218 8014
SMSET_CFG	RW	32	0x0000 0024	0x4208 8024	0x4218 8024
SMSET_SESW	RW	32	0x0000 0028	0x4208 8028	0x4218 8028
SMSET_SEDEN_i (1)	RW	32	0x0000 002C + (0x4*i)	0x4208 802C + (0x4*i)	0x4218 802C + (0x4*i)

(1) i = 1 to 8 for EVE1_SMSET
i = 1 to 8 for EVE2_SMSET

8.1.7.4.3 SMSET Register Description

Table 8-301. SMSET_ID

Address Offset	0x0000 0000		
Physical Address	0x4208 8000 0x4218 8000	**Instance**	EVE1_SMSET EVE2_SMSET
Description	SMSET identification register		

Table 8-301. SMSET_ID (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision	R	(1)

(1) TI internal information

Table 8-302. Register Call Summary for Register SMSET_ID

Embedded Vision Engine (EVE) Subsystem

- SMSET Register Summary: [0]

Table 8-303. SMSET_SCFG

Address Offset	0x0000 0010		
Physical Address	0x4208 8010 0x4218 8010	**Instance**	EVE1_SMSET EVE2_SMSET
Description	SMSET system configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode. 0x0: Force-idle; 0x1No-idle; 0x2 Smart-idle; 0x3 Smart-Idle wakeup-capable	R	0x2
1	RESERVED		R	0x0
0	SOFTRESET	Triggers System Event Trace module reset. This bit is automatically cleared by hardware	RW	0x0

Table 8-304. Register Call Summary for Register SMSET_SCFG

Embedded Vision Engine (EVE) Subsystem

- SMSET Register Summary: [0]

Table 8-305. SMSET_SR

Address Offset	0x0000 0014		
Physical Address	0x4208 8014 0x4218 8014	**Instance**	EVE1_SMSET EVE2_SMSET
Description	SMSET Status Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:10	9	8	7:1	0
RESERVED	SWFIFOEMPTY	HWFIFOEMPTY	RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SWFIFOEMPTY	SW message FIFO empty	R	0x1
8	HWFIFOEMPTY	System event trace FIFO empty	R	0x1
7:1	RESERVED		R	0x0
0	RESETDONE	Reset completed	R	0x1

Table 8-306. Register Call Summary for Register SMSET_SR

Embedded Vision Engine (EVE) Subsystem

- SMSET Register Summary: [0]

Table 8-307. SMSET_CFG

Address Offset	0x0000 0024		
Physical Address	0x4208 8024 0x4218 8024	**Instance**	EVE1_SMSET EVE2_SMSET
Description	SMSET Configuration register		
Type	RW		

31:30	29	28	27:8	7	6:5	4	3	2	1	0
OWNERSHIP	DEBUGGEROVERRIDE	CURRENTOWNER	RESERVED	CAPTUREEN	RESERVED	EVENTLEVEL	EVENTMSG	STOP	START	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	OWNERSHIP	Read to get current ownership status. The claim status encoding is (0=Available, 1=Claimed, 2=Enabled, 3=Reserved);Send command to modify ownership state: 00= Release ownership, 01 = Claim ownership, 10 = Enable unit, 11 = No operation	RW	0x0
29	DEBUGGEROVERRIDE	Reading from the DebuggerOverride bit returns a 1.	RW	0x1
28	CURRENTOWNER	This value reflects the SMSET ownership when the register is in a non-Available state.	R	0x0
27:8	RESERVED		R	0x0
7	CAPTUREEN	When high the sytem event capture is enabled	RW	0x0
6:5	RESERVED		R	0x0
4	EVENTLEVEL	This applies to all selected events: 0x0: low level event detection 0x1: high level evnet detection	RW	0x1
3	EVENTMSG	essage generated based on: 0x0: sampling window 0x1: event detection	RW	0x0

Bits	Field Name	Description	Type	Reset
2	STOP	Stop capturing system events from external trigger detection [EMU1 HIGH to LOW]	RW	0x0
1	START	Start capturing system events from external trigger detection [EMU0 HIGH to LOW]	RW	0x0
0	RESERVED		R	0x0

Table 8-308. Register Call Summary for Register SMSET_CFG

Embedded Vision Engine (EVE) Subsystem

- Procedure to Enable System Event Capture: [0] [1] [2] [3] [4]
- Procedure to Start/Stop System Event Capture from External Trigger Detection: [5] [6] [7] [8]
- Procedure to Disable System Event Capture: [9] [10] [11]
- SMSET Register Summary: [12]

Table 8-309. SMSET_SESW

Address Offset	0x0000 0028		
Physical Address	0x4208 8028 0x4218 8028	**Instance**	EVE1_SMSET EVE2_SMSET
Description	System Event Sampling Window register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">SAMPLINGWINDOWSIZE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	SAMPLINGWINDOWSIZE	System events sampling window size expressed as SMSET cycles	RW	0x0

Table 8-310. Register Call Summary for Register SMSET_SESW

Embedded Vision Engine (EVE) Subsystem

- Procedure to Enable System Event Capture: [0]
- Procedure to Start/Stop System Event Capture from External Trigger Detection: [1]
- SMSET Register Summary: [2]

Table 8-311. SMSET_SEDEN_i

Address Offset	0x0000 0030		
Physical Address	0x4208 802C + (0x4*i) 0x4218 802C + (0x4*i)	**Instance**	EVE1_SMSET EVE2_SMSET
Description	System Event Detection Enable register 1		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>EVENT32EN</td><td>EVENT31EN</td><td>EVENT30EN</td><td>EVENT29EN</td><td>EVENT28EN</td><td>EVENT27EN</td><td>EVENT26EN</td><td>EVENT25EN</td><td>EVENT24EN</td><td>EVENT23EN</td><td>EVENT22EN</td><td>EVENT21EN</td><td>EVENT20EN</td><td>EVENT19EN</td><td>EVENT18EN</td><td>EVENT17EN</td><td>EVENT16EN</td><td>EVENT15EN</td><td>EVENT14EN</td><td>EVENT13EN</td><td>EVENT12EN</td><td>EVENT11EN</td><td>EVENT10EN</td><td>EVENT9EN</td><td>EVENT8EN</td><td>EVENT7EN</td><td>EVENT6EN</td><td>EVENT5EN</td><td>EVENT4EN</td><td>EVENT3EN</td><td>EVENT2EN</td><td>EVENT1EN</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	EVENT32EN	Event 32 detection enable	RW	0x0
30	EVENT31EN	Event 31 detection enable	RW	0x0
29	EVENT30EN	Event 30 detection enable	RW	0x0

Bits	Field Name	Description	Type	Reset
28	EVENT29EN	Event 29 detection enable	RW	0x0
27	EVENT28EN	Event 28 detection enable	RW	0x0
26	EVENT27EN	Event 27 detection enable	RW	0x0
25	EVENT26EN	Event 26 detection enable	RW	0x0
24	EVENT25EN	Event 25 detection enable	RW	0x0
23	EVENT24EN	Event 24 detection enable	RW	0x0
22	EVENT23EN	Event 23 detection enable	RW	0x0
21	EVENT22EN	Event 22 detection enable	RW	0x0
20	EVENT21EN	Event 21 detection enable	RW	0x0
19	EVENT20EN	Event 20 detection enable	RW	0x0
18	EVENT19EN	Event 19 detection enable	RW	0x0
17	EVENT18EN	Event 18 detection enable	RW	0x0
16	EVENT17EN	Event 17 detection enable	RW	0x0
15	EVENT16EN	Event 16 detection enable	RW	0x0
14	EVENT15EN	Event 15 detection enable	RW	0x0
13	EVENT14EN	Event 14 detection enable	RW	0x0
12	EVENT13EN	Event 13 detection enable	RW	0x0
11	EVENT12EN	Event 12 detection enable	RW	0x0
10	EVENT11EN	Event 11 detection enable	RW	0x0
9	EVENT10EN	Event 10 detection enable	RW	0x0
8	EVENT9EN	Event 9 detection enable	RW	0x0
7	EVENT8EN	Event 8 detection enable	RW	0x0
6	EVENT7EN	Event 7 detection enable	RW	0x0
5	EVENT6EN	Event 6 detection enable	RW	0x0
4	EVENT5EN	Event 5 detection enable	RW	0x0
3	EVENT4EN	Event 4 detection enable	RW	0x0
2	EVENT3EN	Event 3 detection enable	RW	0x0
1	EVENT2EN	Event 2 detection enable	RW	0x0
0	EVENT1EN	Event 1 detection enable	RW	0x0

Table 8-312. Register Call Summary for Register SMSET_SEDEN_i

Embedded Vision Engine (EVE) Subsystem

- Procedure to Enable System Event Capture: [0]
- Procedure to Start/Stop System Event Capture from External Trigger Detection: [1]
- SMSET Register Summary: [2]

8.2 ARP32 CPU and Instruction Set

8.2.1 Overview

ARP32 refers to a family of customizable, embedded processors targeted for deeply embedded control applications that require high-performance at very-low system cost (die size) and dynamic power-consumption. Example of applications that can benefit using the ARP32 includes embedded video processing engine, computer vision, and video analytics processing engine.

Figure 8-26 illustrates the overlapping nature of the ARP32 CPU and feature set in different application contexts.

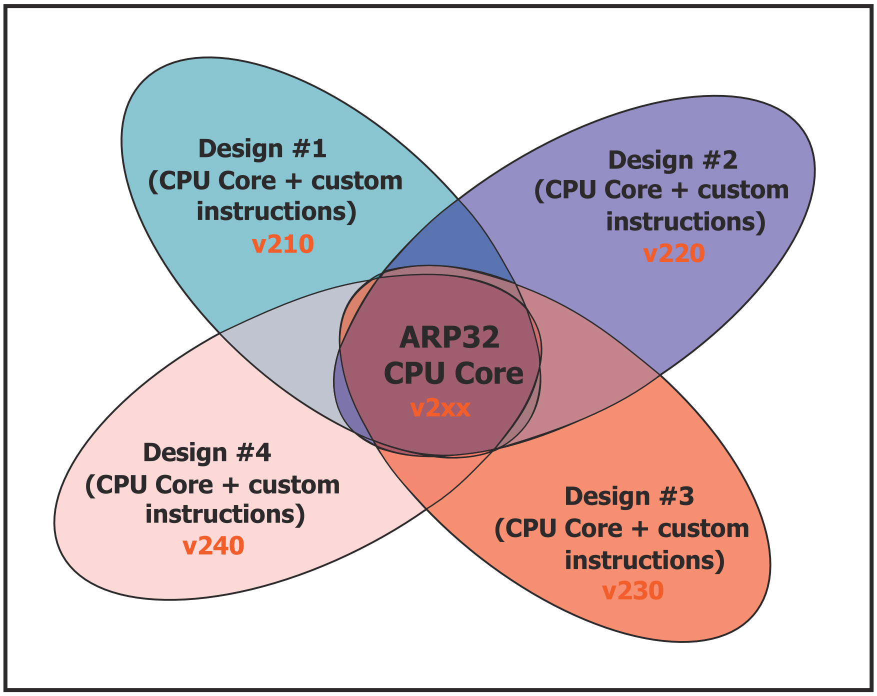


Figure 8-26. ARP32 Versions and ISA/Feature Space

8.2.2 Features

The main features of the ARP32 processor include:

- Single issue RISC processor with Harvard memory architecture
- Highly efficient integer pipeline delivering high performance at low area and power cost
- Provides customization of instruction set to provide acceleration of a target application
- Supported by fully optimizing C/C++ compiler including intrinsic-driven usage of special custom instructions

It is recommended that ARP32 CPU users only use C/C++ language with the supported TI compiler and tools; assembly level programming is highly discouraged and may break future compatibility with the ARP32 processor family or its variants.

8.2.3 Block Diagram

Figure 8-27 shows the ARP32 core interfaces.

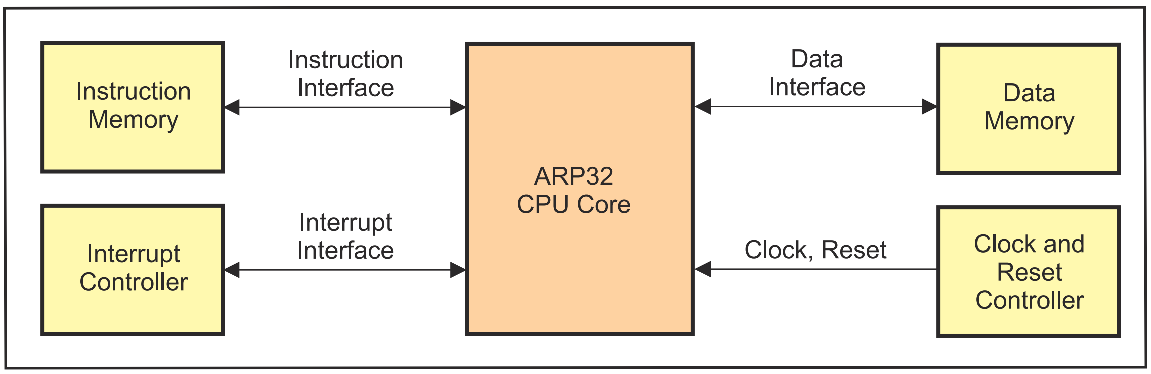


Figure 8-27. ARP32 CPU Block Diagram

8.2.4 Architecture

8.2.4.1 Interface Description

Table 8-313 provides the basic description of the interface signals of the core. The data and instruction memory interface implements a simple request-ready based handshake sufficient to directly interface TIs compiled SRAM memories.

There is no architectural clock-gating implemented within the ARP32 core. The clock control logic is implemented outside the ARP32 core assisted by signals driven by the CPU core via the **IDLE** instruction.

Table 8-313. Interface Signals

Signal	Direction	Width	Description
			Clock, Reset
cpu_fclk	In	1	Functional clock input
cpu_porz_i	In	1	Power On reset (active low)
cpu_resetz_i	In	1	Functional reset (active low)
cpu_standby_o	Out	1	CPU standby status (active high

Table 8-313. Interface Signals (continued)

Signal	Direction	Width	Description
Data Interface			
cpu_dmem_enz_o	Out	1	Data memory request, active low
cpu_dmem_dbg_o	Out	1	Debug access qualifier for Data memory request
cpu_dmem_wrz_o	Out	1	Data memory write enable, active low (0: write, 1:read)
cpu_dmem_bez[3:0]	Out	4	Data memory byte enables, active low
cpu_dmem_addr_o[31:2]	Out	30	Data memory address (carries a word address)
cpu_dmem_wdata_o[31:0]	Out	32	Data memory write data
cpu_dmem_rdata_i[31:0]	In	32	Data memory read data
cpu_dmem_rdy_i	In	1	Data memory ready, active high
cpu_dmem_err_i	In	1	Data memory access error, active high, valid along with cpu_dmem_rdy_i
Instruction Interface			
cpu_imem_enz_o	Out	1	Instruction memory request, active low
cpu_imem_dbg_o	Out	1	Debug access qualifier for Instruction memory request
cpu_imem_addr_o[31:2]	Out	30	Instruction memory address (carries a word address)
cpu_imem_rdata_i[31:0]	In	32	Instruction memory read data
cpu_imem_rdy_i	In	1	Instruction memory ready, active high
cpu_imem_err_i	In	1	Instruction memory access error, active high, valid along with cpu_imem_rdy_i
Interrupt Interface			
cpu_nmi_i	In	1	Non-maskable interrupt, active high
cpu_int4_i	In	1	Maskable interrupt 4, active high
cpu_int5_i	In	1	Maskable interrupt 5, active high
cpu_int6_i	In	1	Maskable interrupt 6, active high
cpu_int7_i	In	1	Maskable interrupt 7, active high
cpu_int8_i	In	1	Maskable interrupt 8, active high
cpu_int9_i	In	1	Maskable interrupt 9, active high
cpu_int10_i	In	1	Maskable interrupt 10, active high
cpu_int11_i	In	1	Maskable interrupt 11, active high
cpu_int12_i	In	1	Maskable interrupt 12, active high
cpu_int13_i	In	1	Maskable interrupt 13, active high
cpu_int14_i	In	1	Maskable interrupt 14, active high
cpu_int15_i	In	1	Maskable interrupt 15, active high
cpu_iack_o	Out	1	Interrupt acknowledge, active high
cpu_inum_o	Out	4	Identifier of acknowledged interrupt

8.2.4.1.1 Data Memory Interface

- The address sent out to data memory is always a word address. Access sizes smaller than word size are supported via byte enables.
- The CPU core asserts the memory request signal (active low, cpu_dmem_enz_o) along with corresponding address (cpu_dmem_addr_o) and read/write qualifier (active low, cpu_dmem_wrz_o) until a ready (active high, cpu_dmem_rdy_i) is received from the memory sub-system.
- For a zero-wait state access, the ready is expected next cycle to the request. Ready must be de-asserted to extend the access cycle for a higher wait state memory subsystem.
- For read accesses, the CPU samples the read data bus (cpu_dmem_rdata_i[31:0]) when ready is asserted. For write accesses, ready indicates write completion whereas read data bus is a don't care.
- For writes accesses, appropriate byte enables (active low, cpu_dmem_bez[3:0]) are asserted in request phase. This is the only way for the memory sub-system to identify access size of a write access. Byte steering is done by the core, hence the memory sub-system can use the ANY lanes of write data bus for easy selections/multiplexing of write data - for byte writes, the write data is duplicated on all four byte lanes of write data bus whereas for halfword writes, the write data is duplicated on both the halfword lanes of write data bus.
- For read accesses, access size is always 32b - the memory sub-system always returns 32b at the requested word address. CPU core selects the appropriate bytes based on access size and access address.

8.2.4.1.2 Instruction Memory Interface

- The address sent out to the instruction memory is always a word address, access size is always a word (32 bits) and writes are not supported. Hence, byte enables and read/write qualifiers are not present for the instruction memory interface.
- The CPU core asserts the memory request signal (active low, cpu_imem_enz_o) along with the corresponding address (cpu_imem_addr_o) until a ready (active high, cpu_imem_rdy_i) is received from the memory subsystem.
- For a zero-wait state access, the ready is expected the next cycle to the request. Ready must be de-asserted to extend the access cycle for a higher wait state memory subsystem.
- The CPU samples the read data bus (cpu_imem_rdata_i[31:0]) when ready is asserted.

8.2.4.2 Pipeline

8.2.4.2.1 Overview

The ARP32 CPU implements a three-deep pipeline (see Figure 8-28): merged fetch-decode (FD), Execute (EXE), and Writeback (WB). The fetch-decode and writeback stages are overlapped.

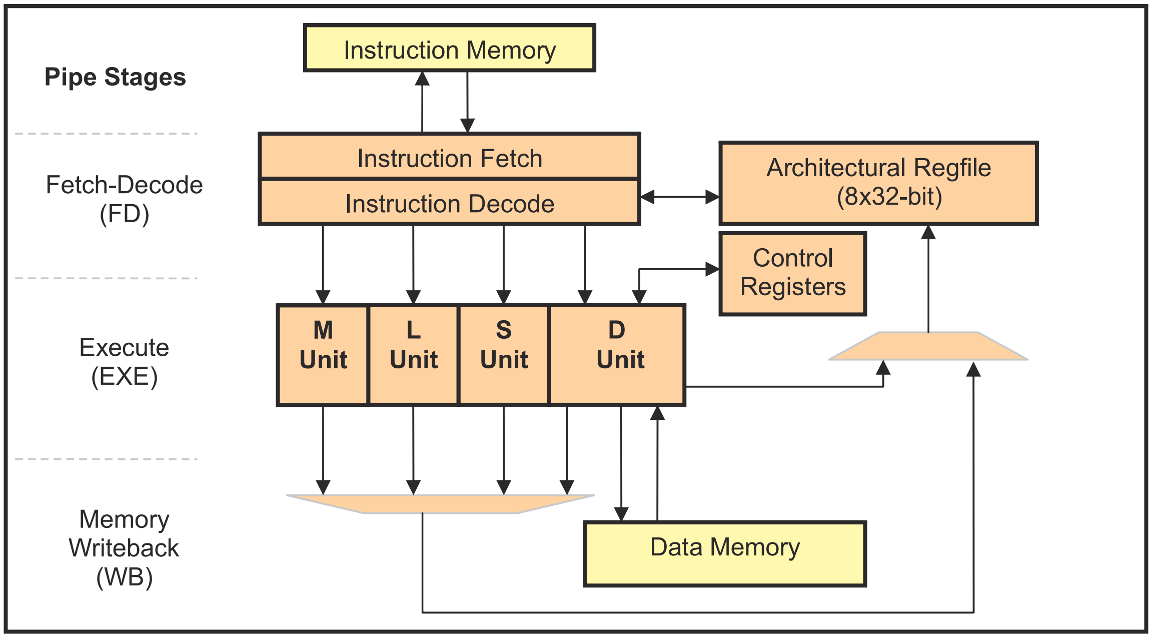


Figure 8-28. ARP32 CPU Pipeline

8.2.4.2.2 Pipeline Operation

The first stage (FetchDecode/FD) of the pipe operates directly on the instruction memory read data input. This stage consists of the following:

- Instructions are decoded to determine
  - size of the instruction (32 bit/16 bit)
  - source operand, destination operand (in architectural and control register file)
  - if it is a blocking multi-execute cycle instruction
- Source operands are read from architectural register file

In the second stage (Execute/EXE) of the pipe:

- All arithmetic/data processing instructions are executed.
- For a single execute cycle instruction, the results are available at the end of this stage and are written back to architectural register file and control register file (if required).
- Instructions requiring data memory access (load, store, call, return etc) compute the access address in this cycle and place the memory access request on data memory bus (along with all other qualifiers). For write accesses, write data is also provided in this stage.

In the last stage (MemoryWriteback/WB) of the pipe:

- All instructions having data memory access request issued in EXE stage waits for the read data (for read accesses) or write completion notification (for write accesses) via the data memory ‘ready’ signal.
- If memory subsystem asserts the ready signal, the read data is updated to architectural register file

Note

The Writeback stage is really outside the processor core and is used for load/store instructions only. All other instructions complete at EXE stage updating the result. Hence, in some sense the pipe depth of the ARP32 processor is really 2 (Fetch-Decode, Execute).

Consider the following example code sequence in Example 8-1. Figure 8-29 shows how the instructions flows through the pipe, highlighting the fact that only load/store instructions use the WB stage (others complete/retire at EXE).

Example 8-1. ARP32 CPU Pipeline Operation

```
MVK   0x100, R0          ; (I1) Move 0x100 into R0
MVKH  0x02000000, R0     ; (I2) Move 0x0200 into R0[31:16]
ADD   100, R0, R0        ; (I3) Add 100 to R0
LDW   *+R2(0), R3        ; (I4) Load a word to R3
ADD   4, R2, R2          ; (I5) Increment R2 to next word address
STW   R3, *+R4(0)        ; (I6) Store R3 to address pointed by R4
ADD   4, R4, R4          ; (I7) Increment R4 to next word address
```

8.2.4.2.2.1

Figure 8-29. ARP32 CPU Pipeline Operation

Cycle	1	2	3	4	5	6	7	8	9
FD	MVK	MVKH	ADD	LDW	ADD	STW	ADD		
	(I1)	(I2)	(I3)	(I4)	(I5)	(I6)	(I7)		
EXE		MVK	MVKH	ADD	LDW	ADD	STW	ADD	
		(I1)	(I2)	(I3)	(I4)	(I5)	(I6)	(I7)	
WB						LDW		STW	
						(I4)		(I6)	

8.2.4.2.3 Pipeline Interlocks

The pipeline is fully interlocked – the CPU stalls in case of source operand registers have pending loads. Data bypassing for read-after-write dependency is implemented at the end of EXE and WB stages to increase instructions-per-cycle (IPC).

The load data has a single-cycle load use penalty since the load data is written back to register file at WB stage. In the following example, the `ADD` instruction stalls for a cycle to allow load to complete:

```
LDW   *+R0(0), R0        ; Load a word into R0
ADD   4, R0, R0          ; Increment R0 to the next word address
MVK   100, R1            ; Move a value 100 into R1
```

Since the CPU allows a nondependent instruction to continue executing, this stall is avoided, if the MVK (no dependency on the load data) instruction is placed in the load delay slot – the CPU executes all three instructions without a stall.

```
LDW   *+R0(0), R0        ; Load a word into R0
MVK   100, R1            ; Move a value 100 into R1
ADD   4, R0, R0          ; Increment R0 to the next word address
```

8.2.4.3 Data Format

The ARP32 CPU is an integer machine supporting three different data width/formats: byte, halfword, and word. A byte is 8 bits, a halfword is 16 bits, and a word is 32 bits – they directly map to native C data types: char, short, and int, respectively.

Each type is aligned with the corresponding size, that is, byte aligned to byte boundary, halfword aligned to 2-byte boundary, word aligned to 4-byte boundary. It is the compiler/toolchain and the programmers responsibility to maintain the access alignment all the time (for global and/or local variables).

The ARP32 core does not support unaligned access. If an unaligned access is attempted by software (via an appropriate instruction), the CPU ignores the lower 2 bits of the data memory (byte) address, thus forcing an alignment to the (truncated, floored) word address.

8.2.4.4 Endian Support

The ARP32 CPU supports only little-endian byte ordering in the memory subsystem (see Figure 8-30). A word loaded from the memory is loaded into a CPU register as follows:

- a byte or half word at address A is the least significant byte or half word within the word at that address
- a byte at address A is the least significant byte within the half word at that address

Figure 8-30. ARP32 CPU Little Endianness

	31 24	23 16	15 8	7 0
Word at address A	Byte [Addr + 3]	Byte [Addr + 2]	Byte [Addr + 1]	Byte [Addr + 0]
Halfword at address A			Byte [Addr + 1]	Byte [Addr + 0]

8.2.4.5 Architectural Register File

The ARP32 CPU contains an 8-entry, 32-bit architectural register file named R0-R7.

The architectural registers are read (for source operands) at the end of Fetch-decode (2 dedicated read ports available at fetch-decode for this) and written back at the end of execute stage (1 dedicated write port available at EXE for this). Additionally, for load instructions, the memory read data is written back to architectural register file at the end of writeback stage (1 dedicated write port available at WB for this) and for store instructions, the architectural register file is read (for write data) at the execute stage (via a dedicated read port available in EXE phase).

The register file contains logic to detect matching of write address in the execute stage and two read addresses in the decode stage for bypassing of result data to read data buses. The read data is flopped in the register file and routed to the execution unit in the execute stage.

8.2.4.6 CPU Control Registers

The control register file (CREG) contains registers that control or report status for the ARP32 CPU. Table 8-314 lists the control registers contained in the control register file.

Control registers are accessed via **MVC**, **MVCH** instructions – this is the only mechanism for moving the contents of registers between the architectural register file and the control register file. Control registers are addressable via 5 bits in the *creg* field of the **MVC/MVCH** instructions. All control registers are defined as 32 bits – **MVC** + **MVCH** pair should be used to move immediate values to any control registers.

The following instructions are available in the **MVC/MVCH** family:

- **MVC** *ucst16, creg*
- **MVC** *areg, creg*
- **MVC** *creg, areg*
- **MVCH** *ucst16, creg*

Table 8-314. Control Registers

Acronym	Register Name	Section
CSR	Control Status Register	Section 8.2.4.6.1
IER	Interrupt Enable Register	Section 8.2.4.6.2
IFR	Interrupt Flag Register	Section 8.2.4.6.3
ISR	Interrupt Set Register	Section 8.2.4.6.4
ICR	Interrupt Clear Register	Section 8.2.4.6.5
NRP	Nonmaskable Interrupt Return Pointer Register	Section 8.2.4.6.6
IRP	Interrupt Return Pointer Register	Section 8.2.4.6.7
SP	Stack Pointer Register	Section 8.2.4.6.8
GDP	Global Data Pointer Register	Section 8.2.4.6.9
LR	Link Register	Section 8.2.4.6.10
LSA0	Loop 0 Start Address Register	Section 8.2.4.6.11
LEA0	Loop 0 End Address Register	Section 8.2.4.6.12
LCNT0	Loop 0 Iteration Count Register	Section 8.2.4.6.13
LSA1	Loop 1 Start Address Register	Section 8.2.4.6.14
LEA1	Loop 1 End Address Register	Section 8.2.4.6.15
LCNT1	Loop 1 Iteration Count Register	Section 8.2.4.6.16
LCNT0RLD	Loop 0 Iteration Count Reload Value Register	Section 8.2.4.6.17
SCSR	Shadow Control Status Register	Section 8.2.4.6.18
NMISCSR	NMI Shadow Control Status Register	Section 8.2.4.6.19
CPUID	CPU ID Register	Section 8.2.4.6.20
DPC	Decode Program Counter	Section 8.2.4.6.21
TSCH	Time-Stamp Counter (high 32 bits) Register	Section 8.2.4.6.22
TSCL	Time-Stamp Counter (low 32 bits) Register	Section 8.2.4.6.22

Direct modification of the CREG entries is limited to a few special case instructions. For example, some forms of the **ADD** and **SUB** instructions directly modify the stack pointer (SP) to improve code execution performance.

A read of a control register (by **MVC** *creg, areg*) can be immediately followed by any other instruction using the same *areg* without causing a stall – the read data is forwarded to the next instruction in the pipeline. The following example illustrates the behavior:

```
// Read from IRP and push it stack
MVC   IRP, R0            ; Read IRP and place it is R0
STRF  R0, R0             ; No stall, content of IRP is pushed to stack
```

A write to a control register is also bypassed to another instruction immediately following it and using the same *creg*. The following example illustrates the behavior:

```
// Update SP before a register push
MVC   5000, SP           ; Update SP to 5000
STRF  R0, R0             ; No stall, R0 is pushed @5000
```

However, a write to a control register (by **MVC** *areg, creg*) has one exposed effective delay slot for an immediately following read from the same control register (by **MVC** *creg, areg*). The following examples illustrate the behavior:

```
// Write to LCNT0 and read it back
MVC   2000, LCNT0        ; Update LCNT0 again
NOP                      ; without this NOP the next MVC will read UNPREDICTABLE value
MVC   LCNT0, R0          ; R0=2000
```

8.2.4.6.1 Control Status Register (CSR)

The control status register (CSR) contains control and status bits. The CSR is shown in Figure 8-31 and described in Table 8-315.

The condition code bits GT, LT, and EQ are not required to be one-hot. They may be set in any combinations using the **MVC** instruction or by combinations of **CMP** and instructions that update the EQ bit. However, execution of compare instructions enforces a one-hot condition for GT/LT/EQ.

Having more than one bit set does not affect conditional branch execution, as each branch compares only the respective condition bits, that is, the **BGE** instruction uses the CSR[4] and CSR[2] bits to determine if the branch is taken. The remaining condition bits have no effect on the **BGE** instruction.

Figure 8-31. Control Status Register (CSR)

31-16
Reserved
R-0

15-14	13	12	11-8	7	6	5	4	3	2	1	0
Reserved	DBZ	DBZE	INUM	V	SAT	C	GT	LT	EQ	Rsvd	GIE
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-315. Control Status Register (CSR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
13	DBZ	0-1	Divide-By-Zero. Status indicating if a Divide-By-Zero condition occurred since last time it was cleared. Once set, software MUST clear this bit. It is set only if the DBZE bit is set.
12	DBZE	0-1	Divide-By-Zero exception enable. When set to 1, CPU takes an UNDEF interrupt when a divide-by-zero condition is detected by **DIV, DIVU, MOD,** or **MODU** instructions. By default this bit is disabled.
11-8	INUM		Interrupt Number. Indicates the interrupt ID of the last taken interrupt.
		0	Reset
		1h	NMI
		2h	SWI
		3h	UNDEF
		4h	INT4
		5h	INT5
		6h	INT6
		7h	INT7
		8h	INT8
		9h	INT9
		Ah	INT10
		Bh	INT11
		Ch	INT12
		Dh	INT13
		Eh	INT14
		Fh	INT15
7	V	0-1	Overflow bit. Arithmetic operations that results in overflow or borrow set this bit. See individual instruction descriptions for instructions that modify the V bit.

Table 8-315. Control Status Register (CSR) Field Descriptions (continued)

Bit	Field	Value	Description
6	SAT	0-1	Saturation bit. Arithmetic operations whose results have been saturated set this bit. See individual instruction descriptions for instructions that modify the SAT bit.
5	C	0-1	Carry bit. Arithmetic operations that results in carry out or borrow set this bit. See individual instruction descriptions for instructions that modify the C bit.
4	GT	0-1	Greater-than bit. This bit is set or cleared based on the result of a **CMP** instruction. (GT = 1 if Rx > Ry; else GT = 0). See individual instruction descriptions for instructions that modify the GT bit.
3	LT	0-1	Less-than bit. This bit is set or cleared based on the result of a **CMP** instruction. (LT = 1 if Rx < Ry; else LT = 0). See individual instruction descriptions for instructions that modify the LT bit.
2	EQ	0-1	Equal bit. This bit is set to 1 if the result of an instruction execution results in a zero result or the result of a **CMP** instruction returns equality (EQ = 1 if Rx == Ry; else EQ = 0). See individual instruction descriptions for instructions that modify the EQ bit.
1	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GIE		Global interrupt enable.
		0	Disables all interrupts, except the reset interrupt and NMI (nonmaskable interrupt).
		1	Enables all interrupts.

8.2.4.6.2 Interrupt Enable Register (IER)

The interrupt enable register (IER) enables and disables individual interrupts. The IER is shown in Figure 8-32 and described in Table 8-316.

Figure 8-32. Interrupt Enable Register (IER)

31–16
Reserved
R-0

15	14	13	12	11	10	9	8	7	6	5	4	3–2	1	0
IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8	IE7	IE6	IE5	IE4	Reserved	NMIE	Rsvd
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-1

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-316. Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IE*n*		Interrupt enable. An interrupt triggers interrupt processing only if the corresponding bit is set to 1.
		0	Interrupt is disabled.
		1	Interrupt is enabled.
3-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	NMIE		Nonmaskable interrupt enable. An interrupt triggers interrupt processing only if the bit is set to 1. The NMIE bit is cleared at reset. After reset, software must set the NMIE bit to enable the NMI and to allow INT15-INT4 to be enabled by the GIE bit in CSR and the corresponding IER bit. The NMIE bit cannot be cleared manually; a write of 0 has no effect. The NMIE bit is also cleared by the occurrence of an NMI.
		0	All nonreset interrupts are disabled.
		1	All nonreset interrupts are enabled.
0	Reserved	1	Reserved. The reserved bit location is always read as 1. A value written to this field has no effect.

8.2.4.6.3 Interrupt Flag Register (IFR)

The interrupt flag register (IFR) contains the status of the maskable interrupts (INT15-INT4) and the NMI interrupt. Each corresponding bit in the IFR is set to 1 when that interrupt occurs; otherwise, the bits are cleared to 0. The IFR is shown in Figure 8-33 and described in Table 8-317.

The update to IFR via a write to the interrupt set register (ISR)/interrupt clear register (ICR) has one effective delay slot; IFR cannot be set/cleared using one **MVC** instruction (a write to ISR/ICR) and read the corresponding changed value of IFR via a very next **MVC** instruction. Software should use a **NOP** after a write to ISR/ICR, if the changed value has to be read immediately.

```
MVC    0x10, ISR          ; Set IFR[4]
NOP                       ; a NOP to fill the delay slot
MVC    IFR, R0            ; R0 would contain 0x10
```

Figure 8-33. Interrupt Flag Register (IFR)

31–16
Reserved
R-0

15	14	13	12	11	10	9	8	7	6	5	4	3–2	1	0
IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	Reserved	NMIF	Rsvd
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Readable by the **MVC** instruction; -*n* = value after reset

Table 8-317. Interrupt Flag Register (IFR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IF*n*		Interrupt flag. Indicates the status of the corresponding maskable interrupt. An interrupt flag may be manually set by setting the corresponding bit (IS*n*) in the interrupt set register (ISR) or manually cleared by setting the corresponding bit (IC*n*) in the interrupt clear register (ICR).
		0	Interrupt has not occurred.
		1	Interrupt has occurred.
3-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	NMIF		Nonmaskable interrupt flag.
		0	Interrupt has not occurred.
		1	Interrupt has occurred.
0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

8.2.4.6.4 Interrupt Set Register (ISR)

The interrupt set register (ISR) allows to manually set the maskable interrupts (INT15-INT4) in the interrupt flag register (IFR). Writing a 1 to any of the bits in ISR causes the corresponding interrupt flag (IF*n*) to be set in IFR. Writing a 0 to any bit in ISR has no effect. It is not possible to set any bit in ISR to affect NMI or reset. The ISR is shown in Figure 8-34 and described in Table 8-318.

Note

Any write to ISR (by the **MVC** instruction) effectively has one delay slot because the results cannot be read (by the **MVC** instruction) in IFR until two cycles after the write to ISR.

Any write to the interrupt clear register (ICR) is ignored by a simultaneous write to the same bit in ISR.

Figure 8-34. Interrupt Set Register (ISR)

31–16
Reserved
R-0

15	14	13	12	11	10	9	8	7	6	5	4	3–0
IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	Reserved
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	R-0

LEGEND: R = Read only; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-318. Interrupt Set Register (ISR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IS*n*		Interrupt set.
		0	Corresponding interrupt flag (IF*n*) in IFR is not set.
		1	Corresponding interrupt flag (IF*n*) in IFR is set.
3-0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

8.2.4.6.5 Interrupt Clear Register (ICR)

The interrupt clear register (ICR) allows to manually clear the maskable interrupts (INT15-INT4) in the interrupt flag register (IFR). Writing a 1 to any of the bits in ICR causes the corresponding interrupt flag (IF*n*) to be cleared in IFR. Writing a 0 to any bit in ICR has no effect. Incoming interrupts have priority and override any write to ICR. It is not possible to set any bit in ICR to affect NMI or reset. The ISR is shown in Figure 8-35 and described in Table 8-319.

Note

Any write to ICR (by the **MVC** instruction) effectively has one delay slot because the results cannot be read (by the **MVC** instruction) in IFR until two cycles after the write to ICR.

Any write to ICR is ignored by a simultaneous write to the same bit in the interrupt set register (ISR).

Figure 8-35. Interrupt Clear Register (ICR)

31	16
Reserved	
R-0	

15	14	13	12	11	10	9	8	7	6	5	4	3 0
IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8	IC7	IC6	IC5	IC4	Reserved
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	R-0

LEGEND: R = Read only; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-319. Interrupt Clear Register (ICR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IC*n*		Interrupt clear.
		0	Corresponding interrupt flag (IF*n*) in IFR is not cleared.
		1	Corresponding interrupt flag (IF*n*) in IFR is cleared.
3-0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

8.2.4.6.6 Nonmaskable Interrupt (NMI) Return Pointer Register (NRP)

The NMI return pointer register (NRP) contains the return pointer that directs the CPU to the proper location to continue program execution after NMI processing. A branch using the address in NRP (**BNRP**) in your interrupt service routine returns to the program flow when NMI servicing is complete. The NRP is shown in Figure 8-36 and described in Table 8-320.

Figure 8-36. NMI Return Pointer Register (NRP)

31	30 ... 0
Rsvd	Return Address
R-0	R/W-x

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -x = value is indeterminate after reset

Table 8-320. NMI Return Pointer Register (NRP) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Return Address	0-7FFF FFFFh	Contains the return address. It is written by the CPU while taking an NMI. It is read by the CPU to execute the **BNRP** instruction.

8.2.4.6.7 Interrupt Return Pointer Register (IRP)

The interrupt return pointer register (IRP) contains the return pointer that directs the CPU to the proper location to continue program execution after processing a maskable interrupt (INT15-INT4, SWI, or UNDEF). A branch using the address in IRP (**BIRP**) in your interrupt service routine returns to the program flow when interrupt servicing is complete. The IRP is shown in Figure 8-37 and described in Table 8-321.

Figure 8-37. Interrupt Return Pointer Register (IRP)

31	30 ... 0
Rsvd	Return Address
R-0	R/W-x

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -x = value is indeterminate after reset

Table 8-321. Interrupt Return Pointer Register (IRP) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Return Address	0-7FFF FFFFh	Contains the return address. It is written by the CPU while taking an interrupt. It is read by the CPU to execute the **BIRP** instruction.

8.2.4.6.8 Stack Pointer Register (SP)

The stack pointer register (SP) may be directly updated by software using the **MVC** instruction. The SP points to the top of the stack and always contains a byte address. The programmer is responsible for ensuring the correct alignment of the SP. See Section 8.2.4.14 for more details on stack pointer convention that must be followed. The SP is shown in Figure 8-38.

Figure 8-38. Stack Pointer Register (SP)

31 ... 0
Stack Pointer
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

8.2.4.6.9 Global Data Pointer Register (GDP)

The GDP may be directly updated by software using the **MVC/MVCH** instruction. The GDP points to the start of the global data section (.bss section) and always contains a byte address. The programmer is responsible for ensuring that the convention described in Section 8.2.4.15 is maintained. The GDP is shown in Figure 8-39.

Figure 8-39. Global Data Pointer Register (GDP)

31 ... 0
Global Data Pointer
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

8.2.4.6.10 Link Register (LR)

The link register (LR) contains the return address (a halfword address) for a **RET** instruction. A **CALL** instruction saves the last return address into the stack and updates LR with the return address. A **RET** instruction uses the current value of LR as the target address and restores the last return address into LR from the stack. The LR is shown in Figure 8-40 and described in Table 8-322.

Figure 8-40. Link Register (LR)

31	30 ... 0
Rsvd	Return Address
R-0	R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-322. Link Register (LR) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Return Address	0-7FFF FFFFh	Contains the return address (a halfword address) for a **RET** instruction.

8.2.4.6.11 Loop 0 Start Address Register (LSA0)

The loop 0 start address register (LSA0) contains the start address (byte address) of Loop 0. Note that Loop 0 is the inner loop. The LSA0 is shown in Figure 8-41 and described in Table 8-323.

See Section 8.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 8-41. Loop 0 Start Address Register (LSA0)

31 ... 0
LSA0
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-323. Loop 0 Start Address Register (LSA0) Field Descriptions

Bit	Field	Value	Description
31-0	LSA0	0-FFFF FFFFh	Start address of Loop 0 (byte address).

8.2.4.6.12 Loop 0 End Address Register (LEA0)

The loop 0 end address register (LEA0) contains the end address (byte address) of Loop 0. Note that Loop 0 is the inner loop. The LEA0 is shown in Figure 8-42 and described in Table 8-324.

See Section 8.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 8-42. Loop 0 End Address Register (LEA0)

31 ... 0
LEA0
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-324. Loop 0 End Address Register (LEA0) Field Descriptions

Bit	Field	Value	Description
31-0	LEA0	0-FFFF FFFFh	End address of Loop 0 (byte address).

8.2.4.6.13 Loop 0 Iteration Count Register (LCNT0)

The loop 0 iteration count register (LCNT0) contains the iteration count of Loop 0. Note that Loop 0 is the inner loop. The LCNT0 is shown in Figure 8-43 and described in Table 8-325.

See Section 8.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 8-43. Loop 0 Iteration Count Register (LCNT0)

31 ... 0
LCNT0
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-325. Loop 0 Iteration Count Register (LCNT0) Field Descriptions

Bit	Field	Value	Description
31-0	LCNT0	0-FFFF FFFFh	Iteration count of Loop 0.

8.2.4.6.14 Loop 1 Start Address Register (LSA1)

The loop 1 start address register (LSA1) contains the start address (byte address) of Loop 1. Note that Loop 1 is the outer loop. The LSA1 is shown in Figure 8-44 and described in Table 8-326.

See Section 8.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 8-44. Loop 1 Start Address Register (LSA1)

31 ... 0
LSA1
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-326. Loop 1 Start Address Register (LSA1) Field Descriptions

Bit	Field	Value	Description
31-0	LSA1	0-FFFF FFFFh	Start address of Loop 1 (byte address).

8.2.4.6.15 Loop 1 End Address Register (LEA1)

The loop 1 end address register (LEA1) contains the end address (byte address) of Loop 1. Note that Loop 1 is the outer loop. The LEA1 is shown in Figure 8-45 and described in Table 8-327.

See Section 8.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 8-45. Loop 1 End Address Register (LEA1)

31 ... 0
LEA1
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-327. Loop 1 End Address Register (LEA1) Field Descriptions

Bit	Field	Value	Description
31-0	LEA1	0-FFFF FFFFh	End address of Loop 1 (byte address).

8.2.4.6.16 Loop 1 Iteration Count Register (LCNT1)

The loop 1 iteration count register (LCNT1) contains the iteration count of Loop 1. Note that Loop 1 is the outer loop. The LCNT1 is shown in Figure 8-46 and described in Table 8-328.

See Section 8.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 8-46. Loop 1 Iteration Count Register (LCNT1)

31 ... 0
LCNT1
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-328. Loop 1 Iteration Count Register (LCNT1) Field Descriptions

Bit	Field	Value	Description
31-0	LCNT1	0-FFFF FFFFh	Iteration count of Loop 1.

8.2.4.6.17 Loop 0 Iteration Count Reload Value Register (LCNT0RLD)

The loop 0 iteration count reload value register (LCNT0RLD) contains the value that the loop 0 iteration count register (LCNT0) is reloaded during the outer loop rewind. LCNT0RLD is automatically updated by hardware when a write to LCNT0 is performed (via **MVC** instruction), thus LCNT0RLD does not need to be explicitly programmed during a loop setup. The LCNT0RLD is shown in Figure 8-47 and described in Table 8-329.

See Section 8.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 8-47. Loop 0 Iteration Count Reload Value Register (LCNT0RLD)

31 ... 0
LCNT0RLD
R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-329. Loop 0 Iteration Count Reload Value Register (LCNT0RLD) Field Descriptions

Bit	Field	Value	Description
31-0	LCNT0RLD	0-FFFF FFFFh	Reload value of the loop 0 iteration count register (LCNT0).

8.2.4.6.18 Shadow Control Status Register (SCSR)

The shadow control status register (SCSR) contains a copy of the control status register (CSR) (of background code) in the interrupt context (except for an NMI). The SCSR is shown in Figure 8-48 and described in Table 8-330.

On acceptance of an interrupt (except for an NMI), the current state of the control status register (CSR) is copied to SCSR. On execution of a **BIRP** instruction, SCSR is copied back to CSR. SCSR facilitates fast interrupt response. A write to SCSR has no effect on machine operation.

Figure 8-48. Shadow Control Status Register (SCSR)

31-16
Reserved
R-0

15-14	13	12	11-8	7	6	5	4	3	2	1	0
Reserved	DBZ	DBZE	INUM	V	SAT	C	GT	LT	EQ	Rsvd	GIE
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-330. Shadow Control Status Register (SCSR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
13	DBZ	0-1	Shadow copy of CSR:DBZ bit.
12	DBZE	0-1	Shadow copy of CSR:DBZE bit.
11-8	INUM	0-Fh	Shadow copy of CSR:INUM field.
7	V	0-1	Shadow copy of CSR:V bit.
6	SAT	0-1	Shadow copy of CSR:SAT bit.
5	C	0-1	Shadow copy of CSR:C bit.
4	GT	0-1	Shadow copy of CSR:GT bit.
3	LT	0-1	Shadow copy of CSR:LT bit.
2	EQ	0-1	Shadow copy of CSR:EQ bit.
1	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GIE	0-1	Shadow copy of CSR:GIE bit.

8.2.4.6.19 NMI Shadow Control Status Register (NMISCSR)

The NMI shadow control status register (NMISCSR) contains a copy of the control status register (CSR) (of background code) in the nonmaskable interrupt context. The NMISCSR is shown in Figure 8-49 and described in Table 8-331.

On acceptance of an NMI, the current state of the control status register (CSR) is copied to NMISCSR. On execution of a **BNRP** instruction, NMISCSR is copied back to CSR. NMISCSR facilitates correct servicing of an NMI even in a nested context. A write to NMISCSR has no effect on machine operation.

Figure 8-49. NMI Shadow Control Status Register (NMISCSR)

31–16
Reserved
R-0

15–14	13	12	11–8	7	6	5	4	3	2	1	0
Reserved	DBZ	DBZE	INUM	V	SAT	C	GT	LT	EQ	Rsvd	GIE
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 8-331. NMI Shadow Control Status Register (NMISCSR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
13	DBZ	0-1	Shadow copy of CSR:DBZ bit.
12	DBZE	0-1	Shadow copy of CSR:DBZE bit.
11-8	INUM	0-Fh	Shadow copy of CSR:INUM field.
7	V	0-1	Shadow copy of CSR:V bit.
6	SAT	0-1	Shadow copy of CSR:SAT bit.
5	C	0-1	Shadow copy of CSR:C bit.
4	GT	0-1	Shadow copy of CSR:GT bit.
3	LT	0-1	Shadow copy of CSR:LT bit.
2	EQ	0-1	Shadow copy of CSR:EQ bit.
1	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GIE	0-1	Shadow copy of CSR:GIE bit.

8.2.4.6.20 CPU Identification Register (CPUID)

The CPU identification register (CPUID) contains the major and minor version ID. The CPUID is shown in Figure 8-50 and described in Table 8-332.

Figure 8-50. CPU Identification Register (CPUID)

31–20	19–16
Reserved	CPUNUM
R-0	R-x

15–12	11–0
Y_MINOR	X_MAJOR
R-x	R-x

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset; -x = value is indeterminate after reset

Table 8-332. CPU Identification Register (CPUID) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
19-16	CPUNUM	0-Fh	Reflects the value on the t_cpunum_i[3:0] input port. The input port is sampled on reset release and kept constant – it is not sampled continuously.
15-12	Y_MINOR	0-Fh	Minor revision.
11-0	X_MAJOR	0-FFFh	Major revision.

8.2.4.6.21 Decode Program Counter Register (DPC)

The decode program counter register (DPC) contains the halfword address of the next instruction to be executed (or instruction currently at DEC stage of the pipe). The DPC is shown in Figure 8-51 and described in Table 8-333.

The DPC is a read-only register provided for debug purposes only. A read of DPC while the CPU is halted at a debug event shows the PC of the next instruction to be executed (after the CPU is resumed from a halted state). The debugger can use DPC to determine the halt location. Reading DPC from the debugger while the CPU is running has limited or no use.

An **MVC** instruction with DPC as write address is ignored by the CPU, no errors are generated.

A read of DPC via the **MVC** instruction returns the address of the **MVC** instruction itself. For example, in the following assembly instruction sequence, the read of DPC via the **MVC** instruction returns 0x101 in R1:

```
0x100    ADDD 1, R0, R0
0x101    MVC  DPC, R1    ; R1 gets 0x101
```

Figure 8-51. Decode Program Counter Register (DPC)

31	30 ... 0
Rsvd	Decode Program Counter
R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 8-333. Decode Program Counter Register (DPC) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Decode Program Counter	0-7FFF FFFFh	Contains the address of the next instruction to be executed.

8.2.4.6.22 Time Stamp Counter Registers (TSCL and TSCH)

The CPU contains a free running 64-bit counter that advances each CPU clock under normal operation. The counter is accessed as two 32-bit read-only control registers, TSCL (Figure 8-52) and TSCH (Figure 8-53).

Figure 8-52. Time Stamp Counter Register - Low Half (TSCL)

31 ... 0
CPU clock count (32 LSBs of 64-bit value)
R-0

LEGEND: R = Readable by the **MVC** instruction; -*n* = value after reset

Figure 8-53. Time Stamp Counter Register - High Half (TSCH)

31 ... 0
CPU clock count (32 MSBs of 64-bit value)
R-0

LEGEND: R = Readable by the **MVC** instruction; -*n* = value after reset

8.2.4.6.22.1 Initialization

The counter is cleared to 0 after reset, and counting is disabled. Subsequently, any write to TSCL initializes the counter to 0 (the write data is ignored) and enables the counter. A write to TSCH is completely ignored, without any side effect of enabling the time stamp counter (like that with TSCL). The counter remains enabled until the next CPU functional reset. Once enabled, there is no way to disable the counter or reinitialize the counter to 0.

8.2.4.6.22.2 Enabling Counting

The counter is enabled by writing to TSCL. The value written is ignored. Counting begins in the cycle after the **MVC** instruction executes. If executed with the count disabled, the following code sequence shows the timing of the count starting (assuming no stalls occur in the three cycles shown).

```
MVC    0, TSCL            ; Start the counter
MVC    TSCL, R0           ; R0 = 0
MVC    TSCL, R1           ; R1 = 1
```

8.2.4.6.22.3 Disabling Counting

Once enabled, counting cannot be disabled under program control. Counting is disabled in the following cases:

- After exiting the reset state.
- When the CPU is fully powered down.

8.2.4.6.22.4 Reading the Counter

Reading the full 64-bit count takes two sequential **MVC** instructions. A read from TSCL causes the upper 32 bits of the current count (the value of the counter when the corresponding **MVC** instruction is at Fetch/DEC stage) to be copied into TSCH. In normal operation, only this snapshot of the upper half of the 64-bit count is available to the programmer. The value read will always be the value copied at the cycle of the last **MVC TSCL,** *reg* instruction. If it is read with no TSCL reads having taken place since reset, then the reset value of 0 is read.

When reading the full 64-bit value, it must be ensured that no interrupts are serviced between the two **MVC** instructions if an ISR is allowed to make use of the time stamp counter. There is no way for an ISR to restore the previous value of TSCH (snapshot) if it reads TSCL, since a new snapshot is performed.

The following example shows the correct and recommended way of obtaining the full 64-bit value of the time stamp counter:

```
// Disable interrupts
MVC    CSR, R7            ; Get CSR into R7
CLR    0, 0, R7, R7       ; Clear R7[0]
MVC    R7, CSR            ; Copy R7 back to CSR (clears GIE)
// Read Counter
MVC    TSCL, R0           ; read lower 32b into R0
MVC    TSCH, R1           ; read upper 32b into R1
// Enable interrupts
MVC    CSR, R7            ; Get CSR into R7
SET    0, 0, R7, R7       ; Set R7[0]
MVC    R7, CSR            ; Copy R7 back to CSR (sets GIE)
```

8.2.4.7 CPU Shadow Registers

The ARP32 CPU implements a shadow copy for the following registers:

- All architectural registers: R0-R7
- Loop control registers: LSA*n*, LEA*n*, LCNT*n*

The shadow registers are used to save-off the main registers during an interrupt processing. When an interrupt is taken, the main registers are copied onto the corresponding shadow register by the CPU along with saving the interrupt return address in IRP/NRP. The original registers are restored back upon a return from interrupt – as a result of a **BIRP** or **BNRP** instruction execution.

As a result of this, as long as interrupts are not allowed to be nested, interrupt handler routines in the ARP32 CPU do not need to save of any registers on the stack, this significantly reduces the interrupt processing latency.

Table 8-334 provides the list of the shadow registers in the ARP32 CPU.

Table 8-334. CPU Shadow Registers

Acronym	Register Name	Description
SR0	Shadow R0	Shadow copy of R0
SR1	Shadow R1	Shadow copy of R1
SR2	Shadow R2	Shadow copy of R2
SR3	Shadow R3	Shadow copy of R3
SR4	Shadow R4	Shadow copy of R4
SR5	Shadow R5	Shadow copy of R5
SR6	Shadow R6	Shadow copy of R6
SR7	Shadow R7	Shadow copy of R7
SLSA0	Shadow Loop 0 Start Address	Shadow copy of LSA0
SLEA0	Shadow Loop 0 End Address	Shadow copy of LEA0
SLCNT0	Shadow Loop 0 Iteration Count	Shadow copy of LCNT0
SLSA1	Shadow Loop 1 Start Address	Shadow copy of LSA1
SLEA1	Shadow Loop 1 End Address	Shadow copy of LEA1
SLCNT1	Shadow Loop 1 Iteration Count	Shadow copy of LCNT1
SLCNT0RLD	Shadow Loop 0 Iteration Count Reload	Shadow copy of LCNT0RLD

Shadow registers are access via the **MVS** instruction:

- **MVS** *sreg, areg* - Read shadow register into an architectural register
- **MVS** *areg, sreg* - Write shadow register from an architectural register

The **MVS** instruction is executed by the S unit. Any write to the shadow registers (via a **MVS** *areg, sreg* instruction) has two exposed delay slots - a subsequent read from the same *sreg* (via a **MVS** *areg, sreg* instruction) must be separated by at least two instructions. Otherwise, the old value of *sreg* is read.

The read result (of a **MVS** *sreg, areg* instruction) is bypassed appropriately to a subsequent instruction using the same *areg* – there is no delay slot or stall in this case.

The following examples illustrate the behavior:

```
// Update SR0 and then read back (assume SR0 = 0x100, R0 = 0x200)
MVS   R0, SR0          ; Move R0 to SR0
NOP                    ; NOP to fill delay slot #1
MVS   SR0, R7          ; Move SR0 to R7, R7 = 0x100 (old value)
```

```
// Update SR0 and then read back (assume SR0 = 0x100, R0 = 0x200)
MVS   R0, SR0          ; Move R0 to SR0
NOP                    ; NOP to fill delay slot #1
```

```
NOP                    ; NOP to fill delay slot #2
MVS   SR0, R7          ; Move SR0 to R7, R7 = 0x200 (new value)
```

8.2.4.8 Functional Units

The ARP32 CPU contains the following main functional units:

- **L-Unit:** Performs all logical operations (AND/OR/XOR/NOT/shift and minimum/maximum), some bit level operations like bit rotation, signed/unsigned bit extraction, bit reversal, bit clear, bit set operations, left most bit detection, and saturation operation.
- **S-Unit:** Performs the move operations.
- **D-Unit:** Performs arithmetic operations that include compare less/greater than instructions, address calculation for load/store instructions, PC calculations for branch instructions, and stack pointer increment/ decrement for **PUSH/POP** and **CALL/RETURN** instructions.
- **M-Unit:** Performs multiplication, division, and modulo operations.

8.2.4.9 Instruction Fetch

The ARP32 ISA has instructions that are either 16 bit or 32 bit; the mix being chosen to achieve optimal code size without sacrificing the performance (of the amount of actual work done per instruction per cycle).

The ARP32 CPU allows 16/32-bit instructions to be mixed freely without any overhead or switching between instruction decoding mode. The program counter (PC) always points to a halfword address aligned with an instruction boundary. However, the program fetch is always aligned with a word boundary. The ARP32 CPU always requests a 32-bit word from a word-aligned address over its 32-bit wide program interface. Internally, the CPU maintains a 16 bit single entry fetch buffer (FetchBuffer[15:0]) to store an unused instruction word or part of an instruction word.

This allows a simpler program memory subsystem design and allows area, power efficient implementation of tightly coupled program memory subsystem without incurring any overhead on code density.

Since instructions in the ARP32 CPU are either 16 bit or 32 bit and they are freely mixable, any 32-bit word fetched from the program memory (FetchWord[31:0]) may have the following cases:

- **The FetchWord contains two 16-bit instructions:** the first instruction (FetchWord[15:0]) is sent to the instruction decoder and the second instruction (FetchWord[31:16]) is stored in the FetchBuffer. The stored instruction is then sent to the instruction decoder at the next decode cycle. No program request is sent for a cycle (as the fetch buffer already contains the next instruction to be executed) unless there is a branch.
- **The FetchWord contains one 32-bit instruction:** the entire FetchWord is sent to the instruction decoder and the FetchBuffer is invalidated.
- **The FetchWord contains one 16-bit instruction and the lower halfword of the next 32-bit instruction:** the valid 16-bit instruction (in FetchWord[15:0]) is sent to the instruction decoder and the lower halfword of the next 32-bit instruction is stored in the FetchBuffer. A second fetch must be performed to fetch the upper halfword of the 32-bit instruction before it is decoded. The request for this second fetch is sent in the current cycle. When the corresponding FetchWord is available (for example, at the next cycle), its lower halfword constitutes the full 32-bit instruction word and is sent to the instruction decoder. The unused upper halfword of the current FetchWord is stored back to the FetchBuffer again.

Thus, for a sequential program with an arbitrary mix of 16-bit and 32-bit instructions, the instruction is executed without a single cycle of stall (due to unavailability of a whole instruction word for decoding) in the pipeline. However, if the target address of a program discontinuity (due to branch, call, return interrupt) is not word aligned and there exists a 32-bit instruction at the target address, a single 32-bit aligned fetch is not sufficient to fetch a full 32-bit instruction word. In this case, there is a stall/bubble cycle in the CPU pipeline. The fetch engine of the CPU issues an additional instruction fetch request at the next cycle to fulfill the required 32-bit instruction fetch and the CPU pipeline continues normally.

8.2.4.10 Alignment of 32-bit Instructions

The fetch engine of the ARP32 CPU supports 16-bit and 32-bit instructions to be aligned at any halfword boundary. For a sequential execution, CPU fetches and executes from such instruction stream without incurring any stall cycles. However, if a program discontinuity target contains an unaligned 32-bit instruction, the CPU stalls for a cycle.

Note

The following requirements are for efficient code generation, not functionally correct code generation. The ARP32 CPU supports having a 32-bit instruction at an unaligned discontinuity target – with the associated stall overhead. There are cases where this cannot be avoided – for example, an interrupt return to an unaligned 32-bit instruction.

This requires that for the most efficient programming of the ARP32 CPU, all programs contain 32-bit instructions to be aligned on a 32-bit boundary in the program memory. For the ARP32 C/C++ compiler/toolchain, this means that a 32-bit alignment is maintained for the following cases:

- All assembly level basic blocks that start with a 32-bit instruction
- All functions that start with a 32-bit instruction
- All HLA loop rewind targets (content of LSA*n* registers) that contains a 32-bit instruction

If the previous conditions are met, the following most frequent cases where a program discontinuity happens, do not incur any additional stalls:

- Use of branch/call immediate instructions (**Bcc** *scst9*, **Bcc** *scst16*, or **CALL** *scst22*) for general program flow control
- Use of register call (**CALL** *src1*) for long calls to labels or function calls using function pointers
- Use of HLA for loop constructs and corresponding branch

8.2.4.11 Instruction Execution in Branch Delay Slot

The ARP32 CPU allows one delay slot for the following control flow instructions:

- branch (**Bcc** *scst9*, **Bcc** *scst16*, or **Bcc** *dst*)
- call (**CALL** *scst22* or **CALL** *src1*)
- return (**RET**)

There is no delay slot for SWI.

An instruction is placed in a branch delay slot as long as the restrictions mentioned in Section 8.2.A.3 are maintained.

8.2.4.12 Address Space

The ARP32 CPU uses the convention that both instruction and data memory is byte addressable. The ARP32 CPU is capable of supporting full 32-bit instruction and data memory address space.

The ARP32 core is a pure Harvard architecture, separate instruction and data bus access the program and data memory space. However, while integrating the core, it is possible to unify the program and data space to make it a unified (modified Harvard) memory architecture.

8.2.4.13 Program Counter Convention

In the ARP32 CPU, the PC (program counter register) always contains halfword addresses. Correspondingly, all PC related registers (IRP, NRP) also contain a halfword address. The PC conventions used in pseudo code and other program snippets presume this convention, PC = PC + 1 indicates advancement of the PC by 1 entry, with each entry containing 2 bytes.

However, since program fetch is always of word size and word aligned, a word address is sent over the instruction fetch address bus (cpu_imem_raddr_o[31:2]).

This convention is used in most of the cases in the ISA where a PC related operation is implied. However, there are some use cases where an absolute byte address is needed to reference a location in program memory. Based on these considerations, the following rules/conventions are applied to PC related instructions and operations; the ARP32 compiler/toolchain and assembly code must maintain the following rules/conventions:

- For immediate branch instructions (**Bcc** *scst9* or **Bcc** *scst16*), the branch offset is treated as halfword offsets. This offset is directly added to the PC to calculate effective branch address (see immediate branch instructions).
- For immediate call instructions (**CALL** *scst22*), the call address is treated as PC relative halfword offset. This offset is directly added to the PC to calculate effective call address (see immediate call instructions).
- For register branch or call instructions (**Bcc** *src1* or **CALL** *src1*), the register content is treated as the absolute byte address of the branch/call destination. This address is first converted to a halfword address and then loaded to the PC.
- For set loop address instruction (**SLA** *ucst16, creg*), the immediate offset is treated as a PC relative halfword offset. This offset is directly added to the PC to calculate effective branch address (see immediate branch instructions).
- While setting up address in loop address registers (LSA*n*, LEA*n*) using the **MVC/MVCH** instructions (immediate or register forms), the value set in the loop address register is the absolute halfword address (of loop start/end instruction).
- A call instruction (**CALL** *src1* or **CALL** *scst22*) saves off a return address to the stack that is a halfword address. A corresponding return instruction (**RET**) loads this address directly onto the PC to return to the correct location.
- An interrupt saves off a return address to the IRP/NRP register that is a halfword address. A corresponding return instruction (**BIRP** or **BNRP**) loads this address directly onto the PC to return to the correct location.
- Interrupt Service Table (IST) entries are treated as a halfword address to the interrupt handler routine.

The PC is an architectural register, that is, it contains machine state, but is not directly accessible through the instruction set. Instruction execution has an effect on the PC, but the current PC value can not be read or written explicitly.

8.2.4.14 Stack Pointer Convention

The ARP32 CPU has a 32-bit stack pointer register (SP) and its content is always assumed to be a byte address. Any address computation using the SP value considers content of the SP as a byte address while computing an effective address of a byte/halfword/word access.

The ARP32 CPU is a descending stack machine. The ARP32 CPU programming model requires that the programmer and/or the compiler/toolchain maintain the following conventions:

- Stack pointer (SP) is initialized to a high address, it reduces to a lower address as it proceeds through function calls during program execution.
- Stack pointer (SP) must be always aligned to a 4-byte word address. It is software responsibility to keep the SP always aligned to 4-byte word address when allocating/de-allocating stack – there is no hardware support to do this alignment automatically. Since the ARP32 CPU does not support unaligned access, a stack pointer value not aligned to 4-byte boundary may result in access to unaligned addresses and thus produce unexpected result.
- In all cases where the CPU itself modifies the SP (call, return, **LDRF/STRF**), it always increments/decrements it by 4 (word aligned).

- Stack pointer (SP) always points to the next free location where an entry is stacked. For pushing anything onto stack, the CPU writes to the location the SP is currently pointing to and then post-decrements the SP. Similarly, while popping anything from the stack, the CPU pre-decrements the SP and then reads from the location (pre-decremented) the SP points to. For example:
  – Every **CALL** saves the return address to the location the SP is currently pointing to and then post-decrements the SP.
  – Every **RET** pre-decrements the SP and then loads the return address from the location updated SP points to.
- To allocate stack in a function (in the function prolog), reduce the SP by an amount equal to local stack frame size.
- Local variables, input arguments to functions and spilled variables all are accessed via the SP relative positive offset load/store instructions.
- To de-allocate stack in a function (in the function epilog), increase the SP by an amount equal to local stack frame size.

The ARP32 core does not save any context on stack during interrupt processing. The same stack layout and convention is applied to the interrupt handler function/procedure – except for additional register save requirements.

8.2.4.15 Global Data Pointer Convention

Global data pointer register (GDP) contains a byte address of the start of global data section. It is computed by the linker and assigned to a linker symbol that the boot routine must be used to initialize GDP.

8.2.4.16 Conditional Execution

On the ARP32 CPU, conditional execution is available only by using conditional branches.

Most data processing instructions and all compare instructions update the condition flags in control status register: CSR[2]EQ, CSR[3]LT, CSR[4]GT. Some instructions update all flags and some instructions only update a subset. See the instruction descriptions for the flags they affect. Instructions update the status bit relevant to it; other status bits are left unchanged.

Conditional branch instructions are executed, based on the condition flags set in another instruction, either:

- immediately after the instruction that updated the flags
- after any number of intervening instructions that have not updated the flags

See conditional branch instruction for more details.

8.2.4.17 Hardware Loop Acceleration

8.2.4.17.1 Overview

The ARP32 CPU implements a hardware loop assist (HLA) function to reduce cycles in critical inner loops that are typically spent in loop control operations such as loop index increment, decrement, compare, branch operations. Using an HLA mechanism, a zero-overhead rewind to the top of the loop is achieved for up to two levels of nested loops. The structure and operation of the HLA is conformable for easy mapping of loop constructs (for, while, do, etc.) available in C/C++.

Up to two levels of nested loops are supported. Additional levels of nesting are possible through standard software techniques.

In a two level nested loop construct, the inner most level is termed as Loop 0, while the outer level is termed Loop 1. For a single-level nested loop, only Loop 0 is used. Control registers associated with Loop 0 and Loop 1 control the operation of each level.

8.2.4.17.2 Loop Registers

HLA operation is setup and controlled by six control registers listed in Table 8-335. Note that LSA*n* and LEA*n* registers contain/reflect the byte address.

Table 8-335. Hardware Loop Control Registers

Acronym	Register Name	Description
LSA0	Loop 0 Start Address Register	Contains the Start Address of Loop 0
LEA0	Loop 0 End Address Register	Contains the End Address of Loop 0
LCNT0	Loop 0 Iteration Count Register	Contains the Iteration Count of Loop 0
LSA1	Loop 1 Start Address Register	Contains the Start Address of Loop 1
LEA1	Loop 1 End Address Register	Contains the End Address of Loop 1
LCNT1	Loop 1 Iteration Count Register	Contains the Iteration Count of Loop 1
LCNT0RLD	Loop 0 Iteration Count Reload Value Register	Contains the LCNT0 reload value during outer loop rewind

8.2.4.17.3 Loop Setup Instructions

Loop control registers are accessed via the **MVC/MVCH** instructions like any other control registers.

To simplify and optimize loop setup in the most frequent compiler generated use cases, a separate instruction called set loop address (**SLA** *ucst16, creg*) is provided. The **SLA** instruction takes a PC-relative immediate positive halfword offset (*ucst16*) and writes the byte address of the location indicated by the PC + *ucst16* value to the LSA*n* and LEA*n* registers.

Loop setup occurs in the following order:

- For a single-level loop (only Loop 0):
  1. Setup LSA0
  2. Setup LEA0
  3. Setup LCNT0
- For a two-level loop (Loop 0 and Loop 1):
  1. Setup LSA1
  2. Setup LEA1
  3. Setup LCNT1
  4. Setup LSA0
  5. Setup LEA0
  6. Setup LCNT0

HLA is not assured to work if the above order is not maintained.

8.2.4.17.4 Loop Operation

The loop setup process for each level of nesting consists of the following (in the specified order):

1. Setup loop start address register (LSAn) with the byte address of the first instruction in the corresponding loop.
2. Setup loop end address register (LEAn) with the byte address of the last instruction in the corresponding loop.
3. Setup loop iteration count register (LCNTn) with the intended loop iteration count.

A loop becomes active as soon as a value greater than 1 is programmed in LCNTn. The LSAn and LEAn registers are written when the corresponding **MVC/SLA** instruction is executed but the values are not used by the processor (for detecting loop rewind condition) until the corresponding loop becomes active.

When a loop becomes active (LCNTn > 1), the PC of the instruction at DEC phase is checked with the LEAn register. If they match, the CPU takes a zero-cycle overhead branch to the top of the loop provided by the LSAn registers. The program fetch request to LSAn is placed when the last instruction of a loop is at DEC. Thus there is no delay slot for the loop rewind branch, neither any stall or null cycle.

For LCNTn = x, the loop rewinds x - 1 times, effectively the loop body executes total x times (x - 1 rewinds and the last fall through pass). For LCNTn = 1, the loop body executes once (zero rewind). For LCNTn = 0, the loop is not considered active – the CPU just linearly executes – however, as a result, the loop body executes once (linear fall through). Note that LCNTn = 0 does not mean loop body would not be executed.

During operation, the LCNTn registers are directly updated (decremented) with the remaining iteration count. The loop counters are updated along with the corresponding rewind-branch. At any time LCNTn is read to determine how many iterations of the corresponding loop are still to be executed. When each level completes, the corresponding LCNTn registers become zero, the LSAn/LEAn registers are left unchanged. Note that for LCNTn = 1, the loop body executes once and the loop counter is decremented when the last instruction of the corresponding loop is executed.

The LCNT0RLD register is updated automatically by the CPU whenever a value is written to LCNT0 (for example, during loop setup). For a two-level nested loop, once the inner most level iteration completes and the CPU rewinds to the top of the outer loop, the LCNT0 is reloaded with the content of LCNT0RLD. Note that even if LCNT0RLD is a programmer visible control register, it acts as a shadow register of LCNT0 (updated on any write to LCNT0) and must not be explicitly written during a loop setup.

There are no restrictions in embedding an HLA construct or instruction sequence, consisting of one or two levels of nested loops, into a larger level of nested loop implemented by normal software techniques.

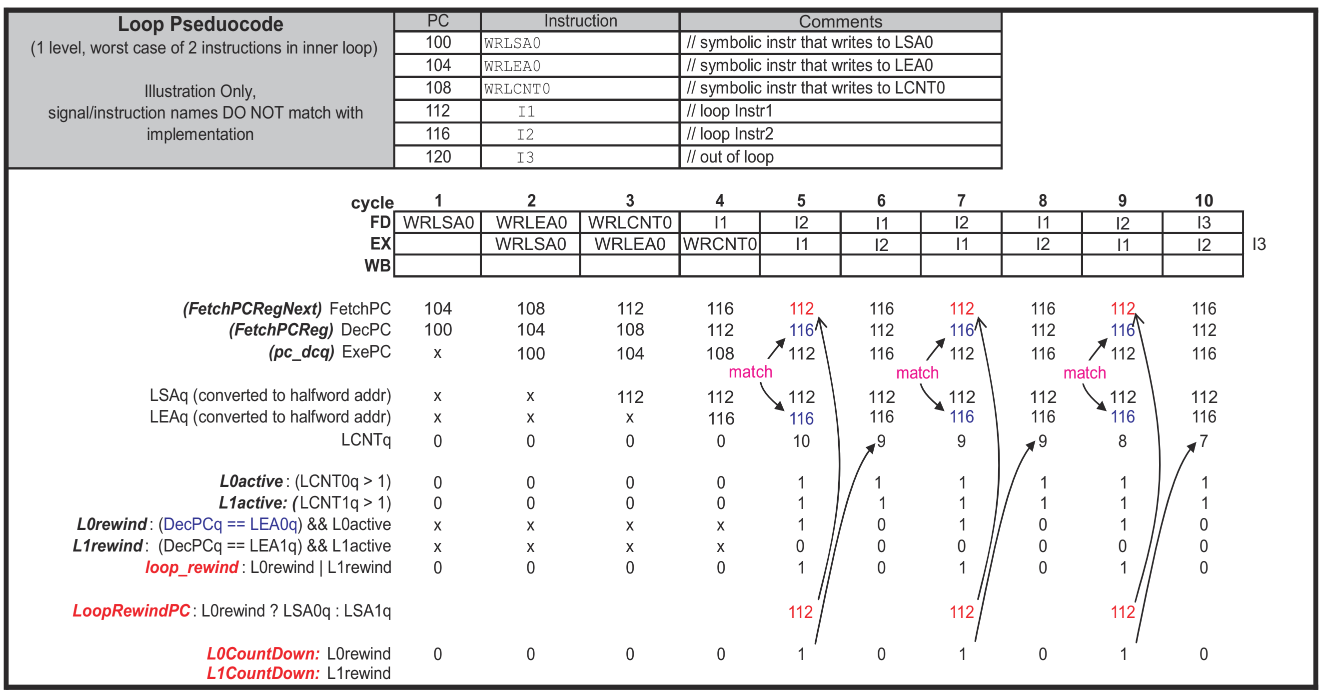


Figure 8-54. Loop Operation

8.2.4.17.5 Call and Branch within Loop Context

Branches (**Bcc** *scst9*, **Bcc** *scst16*, or **Bcc** *dst*) within the loop are allowed, provided the conditions explicitly mentioned in Section 8.2.A.3.2 are met.

Calls (**CALL** *scst22* or **CALL** *dst*) within the loop are generally not supported as the loop registers are not explicitly saved off when a CALL is executed within the loop context. However, software uses **CALL** instructions if the call tree does not contain any HLA usage within the caller code.

8.2.4.17.6 Dynamic Changes to Loop Iteration Count

A write to the LCNTn register within the loop body (of either of the loops) is used to dynamically change the iteration count, including pre-maturely terminating the corresponding level. Such a write to LCNTn instruction must be executed at least 2 instructions before the last (ending) instruction of the corresponding loop (x).

It is also possible to skip inner loop dynamically (per iteration of the outer loop) by simply branching across the inner loop within the outer loop context. It is not necessary to explicitly set the inner loop count to 0/1 (loop inactive values). Similarly, a branch out of the outer loop context is possible without setting the outer loop count to inactive values. This relies on the fact that even if the inner or outer loop count has not been set to an inactive value, a loop rewind des not happen unless the loop end PC (LEAn) is encountered.

These features (dynamically changing the loop iteration count, skipping inner/outer loop via branches) are used by the compiler to implement constructs such as last/continue/break and alike.

8.2.4.17.7 Interrupt Processing During HLA

Interrupts are not taken while the last instruction of a loop level is at DEC. Interrupts are taken after the loop rewind happens. The return address saved off during the interrupt processing points to the loop rewind address (LSAn). Consequently, a return from interrupt returns to the last rewind address (LSAn).

When taking an interrupt (irrespective if an HLA is active), the CPU saves off the six loop registers in the corresponding shadow registers. They are restored when interrupts are returned via the **BIRP/BNRP** instructions.

8.2.4.17.8 HLA Usage in Interrupt Context

Since loop context is fully saved by hardware while taking an interrupt, HLA is used within an ISR context freely without requiring any additional context save. For allowing nested interrupts, the shadow copies of HLA registers must be saved off to the stack explicitly along with other shadow registers. This helps achieve very-low interrupt latency in the ARP32 CPU under a single-level-nested-interrupt use model.

8.2.4.17.9 HLA Usage Restrictions

There are certain restrictions on HLA usage, see Section 8.2.A.3.2.

8.2.4.17.10 HLA Mapping Examples

8.2.4.17.10.1 Loops With Single Level of Nesting

Example 8-2. C memset-like Loop, Single Level, Minimum Instructions

Pseudo C code:

```
int   i, myarray[100];
for (i=0; i< 100; i++) {
    myarray[i] = 1;
}
```

8.2.4.17.10.1.1

Table 8-336. Example 1 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`MVKS 1, R0`	Write data
0x102	`SLA  6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA0 = 0x102 + hex(6 x 2) = 0x10E`
0x106	`SLA  5, LEA0`	Last instr of loop0 is at #5 halfword offset `LEA0 = 0x106 + hex(5 x 2) = 0x110`
0x10A	`MVC  100, LCNT0`	iter count = 100
0x10E	`STW  R0, *++R1(1)`	First instr of Loop 0
0x110	`NOP`	Last instr of Loop 0 (padding needed)

Table 8-337. Example 2 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`MVKS 1, R0`	Write data
0x102	`SLA  6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA0 = 0x102 + hex(6 x 2) = 0x10E`
0x106	`SLA  5, LEA0`	Last instr of loop0 is at #5 halfword offset `LEA0 = 0x106 + hex(5 x 2) = 0x110`
0x10A	`MVC  100, LCNT0`	iter count = 100
0x10E	`STW  R0, *+R1(0)`	First instr of Loop 0
0x110	`ADD  4, R1, R1`	Last instr of Loop 0 (padding needed)

Example 8-3. C memcpy-like Loop, Single Level, Minimum Instructions

Pseudo C code:

```
void MyMemcpyByWord (void *ptr1, void *ptr2, int size){
  int ii, dbwSize = size>>2;
  unsigned int *dst = ptr1;
  unsigned int *src = ptr2;
  for (ii=0 ; ii< dbwSize ; ii++)
  dst[ii] = src[ii];
}
```

8.2.4.17.10.1.2

Table 8-338. Example 1 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 5, LSA0`	First instr of loop0 is at #5 halfword offset `LSA0 = 0x100 + hex(5 x 2) = 0x10A`
0x104	`SLA 4, LEA0`	Last instr of loop0 is at #4 halfword offset `LEA0 = 0x104 + hex(4 x 2) = 0x10C`
0x108	`MVC R1, LCNT0`	iter count = “size” (passed via R1)
0x10A	`LDW *++R2(1), R3`	Load word from source (Loop 0, instr 1)
0x10C	`STW R3, *++R4(1)`	Store word to dest (Loop 0, instr 2)

Table 8-339. Example 2 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 5, LSA0`	First instr of loop0 is at #5 halfword offset `LSA0 = 0x100 + hex(5 x 2) = 0x10A`
0x104	`SLA 6, LEA0`	Last instr of loop0 is at #6 halfword offset `LEA0 = 0x104 + hex(6 x 2) = 0x110`
0x108	`MVC R1, LCNT0`	iter count = “size” (passed via R1)
0x10A	`LDW *+R2(0), R3`	Load word from source (Loop 0, instr 1)
0x10C	`ADD 4, R2, R2`	Increment src pointer
0x10E	`STW R3, *+R4(0)`	Store word to dest
0x110	`ADD 4, R4, R4`	Increment dest pointer (Loop 0, instr 4)

8.2.4.17.10.2 Loops With Two Levels of Nesting

Example 8-4. Two-level Nesting, Both Loops Ending at Same Instruction

Pseudo C code:

```
For (i=0; i<20; i++) {
   Instruction1      // size 16b
   Instruction2      // size 32b
   Instruction3      // size 32b
   Instruction4      // size 16b
   For (j=0; j<10; j++) {
      Instruction5   // size 16b
      Instruction6   // size 32b
      Instruction7   // size 16b
   }
}
```

8.2.4.17.10.2.1

Table 8-340. Example of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 18, LSA1`	First instr of loop1 is at #18 halfword offset `LSA1 = 0x100 + hex(18 x 2) = 0x124`
0x104	`SLA 19, LEA1`	Last instr of loop0 is at #19 halfword offset `LEA0 = 0x104 + hex(19 x 2) = 0x12A`
0x108	`MVC 20, LCNT1`	iter count (=20) of loop1
0x10C	`SLA 6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA1 = 0x10C + hex(6 x 2) = 0x118`
0x110	`SLA 9, LEA0`	First instr of loop0 is at #9 halfword offset `LEA0 = 0x110 + hex(9 x 2) = 0x122`
0x114	`MVC 10, LCNT0`	iter count (=10) of loop0
0x118	`Instruction1`	Loop1 start
0x11A	`Instruction2`	
0x11E	`Instruction3`	
0x122	`Instruction4`	
0x124	`Instruction5`	Loop0 start
0x126	`Instruction6`	
0x12A	`Instruction7`	Loop0 end, Loop1 end

Example 8-5. Two-level Nesting, Different Ending Instructions for Two Levels

Pseudo C code:

```
For (i=0; i<20; i++) {
   Instruction1      // size 16b
   Instruction2      // size 32b
   Instruction3      // size 32b
   Instruction4      // size 16b
   For (j=0; j<10; j++) {
      Instruction5   // size 16b
      Instruction6   // size 32b
      Instruction7   // size 16b
   }
   Instruction8      // size 32b
   Instruction9      // size 32b
   Instruction10     // size 16b
}
```

8.2.4.17.10.2.2

Table 8-341. Example of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 18, LSA1`	First instr of loop1 is at #18 halfword offset `LSA1 = 0x100 + hex(18 x 2) = 0x124`
0x104	`SLA 19, LEA1`	Last instr of loop1 is at #19 halfword offset `LEA0 = 0x104 + hex(19 x 2) = 0x12A`
0x108	`MVC 20, LCNT1`	iter count (=20) of loop1
0x10C	`SLA 6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA1 = 0x10C + hex(6 x 2) = 0x118`
0x110	`SLA 18, LEA0`	First instr of loop0 is at #18 halfword offset `LEA0 = 0x110 + hex(18 x 2) = 0x134`
0x114	`MVC 10, LCNT0`	iter count (=10) of loop0
0x118	`Instruction1`	Loop1 start
0x11A	`Instruction2`	
0x11E	`Instruction3`	
0x122	`Instruction4`	
0x124	`Instruction5`	Loop0 start
0x126	`Instruction6`	
0x12A	`Instruction7`	Loop0 end
0x12C	`Instruction8`	
0x130	`Instruction9`	
0x134	`Instruction10`	Loop1 end

8.2.4.18 Interrupts

8.2.4.18.1 Overview

The ARP32 CPU supports various types of interrupts that include: reset, a non-maskable interrupt (NMI), 12 maskable interrupts (INT15-INT4), an undefined instruction interrupt (UNDEF), and a software interrupt (SWI). The following registers control the CPU behavior on receipt of an interrupt:

- Control Status Register (CSR)
- Interrupt Enable Register (IER)
- Interrupt Flag Register (IFR)
- Interrupt Set Register (ISR)
- Interrupt Clear Register (ICR)
- Nonmaskable Interrupt Return Pointer Register (NRP)
- Interrupt Return Pointer Register (IRP)

The SWI and UNDEF interrupts are special cases that do not have an associated input pin. The SWI interrupt mechanism is activated by decoding of the **SWI** instruction. The UNDEF interrupt is activated by detection of an undefined instruction.

On acknowledge of an enabled interrupt, the ARP32 CPU loads the contents of the interrupt service table entry associated with the interrupt into the PC.

Each interrupt type is discussed in the following sections; Table 8-342 summarizes the ARP32 CPU interrupts.

Table 8-342. Interrupt Summary

Interrupt Name	Vector Address (byte)	Input Pin	Enable Control	Interrupt Return Pointer Register	Interrupt Return Instruction to be Used	Context Save and Restore Actions	
						By Hardware	By Software
Reset	00h	cpu_reset_i	Always Enabled	NA	NA	NA	NA
NMI	04h	cpu_nmi_i	IER:NMIE	NRP	BNRP	CSR to NMISCSR	R0-R7 to Stack, LSA*n* to Stack, LEA*n* to Stack, LCNT*n* to Stack
SWI	08h	None	Always Enabled	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
UNDEF	0Ch	None	Always Enabled	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT4	10h	cpu_int4_i	CSR:GIE; IER:NMIE; IER:IE4	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT5	14h	cpu_int5_i	CSR:GIE; IER:NMIE; IER:IE5	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT6	18h	cpu_int6_i	CSR:GIE; IER:NMIE; IER:IE6	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None

Table 8-342. Interrupt Summary (continued)

Interrupt Name	Vector Address (byte)	Input Pin	Enable Control	Interrupt Return Pointer Register	Interrupt Return Instruction to be Used	Context Save and Restore Actions	
						By Hardware	By Software
INT7	1Ch	cpu_int7_i	CSR:GIE; IER:NMIE; IER:IE7	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT8	20h	cpu_int8_i	CSR:GIE; IER:NMIE; IER:IE8	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT9	24h	cpu_int9_i	CSR:GIE; IER:NMIE; IER:IE9	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT10	28h	cpu_int10_i	CSR:GIE; IER:NMIE; IER:IE10	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT11	2Ch	cpu_int11_i	CSR:GIE; IER:NMIE; IER:IE11	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT12	30h	cpu_int12_i	CSR:GIE; IER:NMIE; IER:IE12	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT13	34h	cpu_int13_i	CSR:GIE; IER:NMIE; IER:IE13	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT14	38h	cpu_int14_i	CSR:GIE; IER:NMIE; IER:IE14	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT15	3Ch	cpu_int15_i	CSR:GIE; IER:NMIE; IER:IE15	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None

8.2.4.18.2 Interrupt Processing

Figure 8-55 provides an illustration of different actions and events of interest during an interrupt processing. In this case, both INT4 and INT5 were asserted simultaneously, INT4 was selected as the winner and was taken after few cycles when all enable conditions (CSR[0]GIE, IER[1]NMIE, IER[n]) were met.

Note that the CPU entity signals/registers are shown in **bold** font; internal (and sometimes just conceptual) signals are shown in normal font.

The conceptual signal “Interrupt Accept” shows when a particular interrupt (in this case, INT4) is accepted to be taken next. This is generated by considering that all set bits of IFR, the interrupt priority, the enable conditions, and a safe cycle boundary within the CPU, where accepting an interrupt is safe.

The conceptual signal “Interrupt Taken” shows when a particular interrupt (in this case, INT4) is actually taken. The actions corresponding to an interrupt processing (IST access, context save, etc.) starts to take place within the CPU after this condition is achieved.

The delay between ‘Interrupt Accept’ and ‘Interrupt Taken’ is due to pending memory transaction on CPU instruction and data interfaces. The CPU waits for all outstanding instruction and data memory requests to be serviced before staring interrupt context save or IST access process.

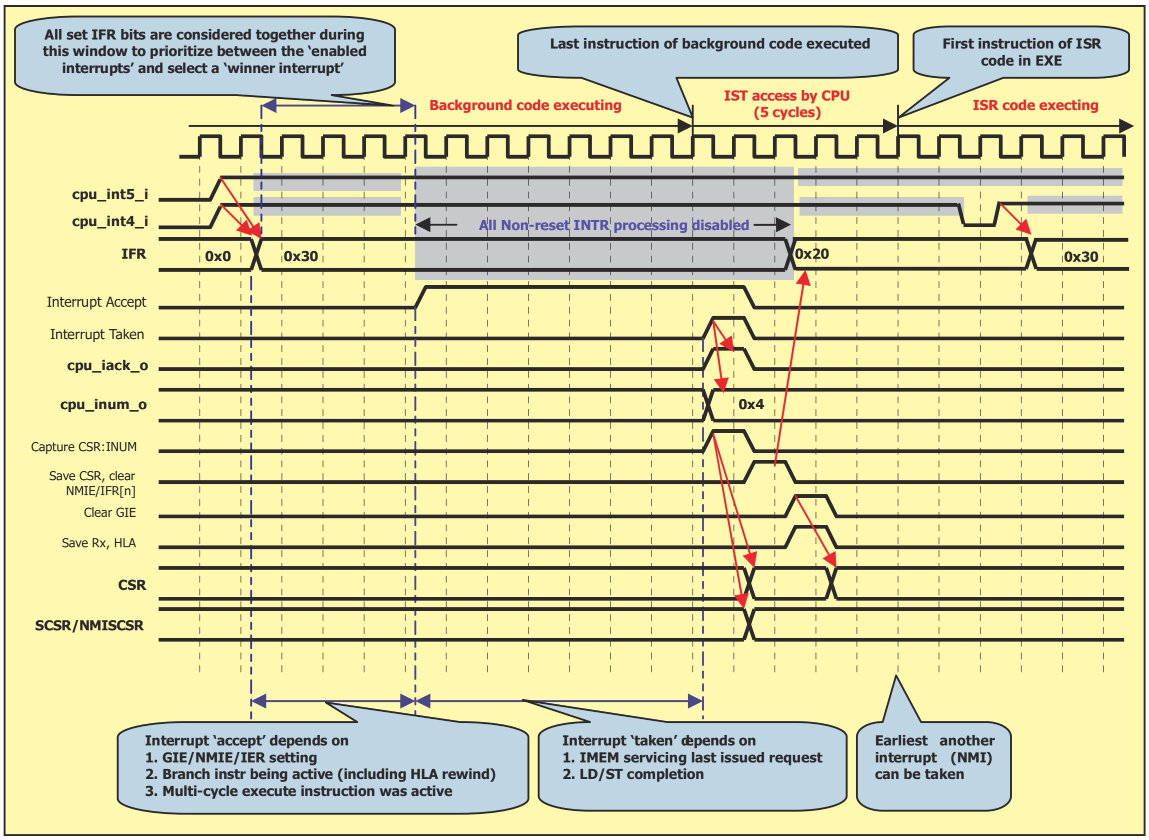


Figure 8-55. Interrupt Processing

8.2.4.18.3 Interrupt Acknowledgment

The ARP32 CPU interrupts are sensitive to rising edge detected on the input interrupt pins (detected synchronously detected on the rising edge of the input clock). All interrupt pins are active high, with the exception of reset, which is active low and asynchronous.

The cpu_iack_o output signal is asserted high for a single cycle to indicate (to hardware external to the CPU core) that the CPU has begun processing an interrupt. The cpu_inum_o signal indicates the number of the interrupt that is being processed. The value driven on cpu_inum_o signal is also captured in the INUM field of the control status register (CSR).

Note that cpu_iack_o and cpu_inum_o are asserted for all interrupts (including reset). Table 8-343 summarizes the cpu_inum_o values.

Table 8-343. cpu_inum_o Values

cpu_inum_o[3:0]	Interrupt
0h	Reset
1h	NMI
2h	SWI
3h	UNDEF
4h	INT4
5h	INT5
6h	INT6
7h	INT7
8h	INT8
9h	INT9
Ah	INT10
Bh	INT11
Ch	INT12
Dh	INT12
Eh	INT14
Fh	INT15

8.2.4.18.4 Interrupt Priorities

When two interrupts happen at the same time, they are serviced by the ARP32 in order of their priority. The priority level for each interrupt is listed in Table 8-344

Table 8-344. Interrupt Priorities

Priority	Interrupt
Highest	Reset
	NMI
	SWI
	INT4
	INT5
	INT6
	INT7
	INT8
	INT9
	INT10
	INT11
	INT12
	INT13
	INT14
	INT15
Lowest	UNDEF

8.2.4.18.5 Interrupt Service Table (IST)

Each entry of the interrupt service table (IST) contains a byte address that points to the beginning of the interrupt service routine for that interrupt. When the ARP32 CPU accepts an interrupt, the contents of the associated IST entry is read (IST[n]), converted to the corresponding halfword address, and then loaded into the PC creating an effective branch.

The starting address of the IST is always the address 00h in the instruction memory; the IST can not be relocated. The addresses (byte address) of each entry is shown in Table 8-345.

IST entries are byte address(s) of the interrupt handler (interrupt service routine) functions. Since 4 bytes are allocated for each IST entry, the interrupt handlers are placed anywhere in the 32-bit program memory space and the ARP32 CPU directly branches to that address. This reduces interrupt latency significantly since an instruction need not be fetched/decoded/executed from an IST location, nor an intermediate routine/call/branch needs to be processed for a branch to relatively large distance.

In fact, the IST is an “array of pointers (each entry 32 bits) to interrupt handler functions in C/C++” placed at the instruction memory address 0, which is remapped to a physical address by using the memory management unit (MMU0) that services program cache requests.

Table 8-345. Interrupt Service Table (IST)

Byte Address	IST Entry
00h	Reset
04h	NMI
08h	SWI
0Ch	UNDEF
10h	INT4
14h	INT5
18h	INT6
1Ch	INT7
20h	INT8
24h	INT9
28h	INT10
2Ch	INT11
30h	INT12
34h	INT13
38h	INT14
3Ch	INT15

8.2.4.18.6 Interrupt Flags

The ARP32 CPU maintains the pending status of all non-reset external interrupts (NMI, INT15-INT4) in the corresponding bit in the interrupt flag register (IFR). A 1 indicates the corresponding interrupt is pending, reading a 0 indicates the corresponding interrupt is not pending (either is being processed or it has not occurred).

When a rising edge is detected on the input pins of external interrupts (NMI, INT15-INT4), the corresponding bits in IFR are set unconditionally in the next CPU clock cycle. When the CPU actually takes a particular interrupt, the corresponding IFR bit is cleared by the CPU. Thus at any point, reading IFR using the **MVC** instruction shows which input external interrupts is asserted or is not yet processed by the CPU (or in pending state).

The CPU checks the status of IFR bits to understand that an interrupt is asserted, and starts processing this interrupt. Processing of an interrupt involves checking for all associated enable conditions, prioritization between simultaneously asserted interrupts and actions to perform interrupt service table look up to finally achieve a branch to the interrupt service routine of the corresponding interrupt. The exact behavior and actions taken on each kind of interrupts are described in Section 8.2.4.18.7.

It is possible to set one or more interrupt flags of maskable interrupts in IFR - by writing to the corresponding bit in the interrupt set register (ISR) - creating the same effect of an external interrupt assertion. This makes it possible for the software to raise a maskable interrupt. The following examples illustrate this:

Example 8-6. Setting Interrupt Flag

```
MVC   0x10, ISR      ; Set IFR[4] to raise INT4
```

Example 8-7. Setting Interrupt Flag

```
MVC   IFR,  R0       ; Get current state
SET   5,5,  R0, R0   ; Set R0:bit[5]
MVC   R0,   ISR      ; Set IFR[5] without changing other bits
```

8.2.4.18.6.1

Note

Setting of IFR by an external interrupt assertion has a higher priority than writing or clearing the IFR bits via the **MVC** instruction.

8.2.4.18.7 Interrupt Behavior

All maskable (INT15-INT4) and non-maskable (NMI) interrupt processing is triggered by the corresponding interrupt flag bit in the interrupt flag register (IFR). For SWI and UNDEF, a direct instruction decode (or non-decode) triggers interrupt processing. Once an interrupt is triggered, the CPU checks the corresponding enabling conditions (if any), does a priority resolution among simultaneously asserted interrupts, and then starts processing the interrupt.

The details of conditions for processing each type of interrupts and the actions taken by the CPU during and returning from the interrupts are described in the following subsections.

8.2.4.18.7.1 Reset Interrupt

Reset is the highest priority interrupt and is used to initialize the CPU to a known state. The reset interrupt is unique in a number of ways:

- cpu_resetz_i is an active-low signal and is treated asynchronously. All other interrupts are active-high signals, or activated via the instruction decoder.
- cpu_resetz_i must be held low for 4 clock cycles before it goes high again to reinitialize the CPU properly.
- cpu_resetz_i is not affected by branches or pending loads or any enable bits or any other condition applicable for processing other interrupts. The current instruction execution is aborted and the CPU registers are returned to their default state.
- Reset uses interrupt semantics, that is, loading of the IST table entry; however, it is not required to issue a **BIRP** instruction to exit reset processing.

8.2.4.18.7.2 Non-maskable Interrupt (NMI)

NMI is the second-highest priority interrupt and is generally used to alert the CPU of a serious hardware problem. For an NMI to be detected (and processed) by the ARP32 CPU, the non-maskable interrupt enable (NMIE) bit in the interrupt enable register (IER) must be set to 1. The NMIE bit is cleared to 0 at reset to prevent an NMI being taken prematurely (before system initialization). The NMI bit is set (to enable NMI) by software only after the system has been properly initialized and is ready to accept NMIs. The NMIE bit is also cleared at the occurrence of an NMI to prevent another NMI from being processed; it may be set again in the interrupt set register (ISR), to allow nested NMIs, by software after the CPU state has been saved properly in the previous NMI ISR. While the NMIE bit is cleared, all maskable interrupts (INT15–INT4) are also disabled.

When the NMIF bit is set, (as a result of an NMI assertion via cpu_nmi_i input pin), assuming the previous conditions are met, the ARP32 CPU is said to accept the interrupt and performs the following actions to process the NMI:

- CPU stops fetching further instruction from instruction memory
- CPU execute pipe is allowed to drain
  - Any load/store instructions (including **LDRF, STRF, CALL, RET**) currently executing are allowed to complete
  - Any instruction in the branch/call/return delay slot is allowed to complete
- The CSR[11:8]INUM field is updated with the NMI interrupt ID.
- The CSR content is copied to the NMISCSR.
- The IER[1]NMIE bit is cleared.
- The CSR[0]GIE bit is cleared.
- The IFR[1]NMIF bit in the IFR is cleared.
- The PC value of the next instruction to execute (after completion of the interrupt service routine) is stored to the NRP register. This is the interrupt return address.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to NMI IST location is sent and eventually the NMI IST entry is loaded into the PC. As a result, the CPU begins executing the NMI interrupt service routine.

To exit an NMI service routine, the **BNRP** instruction must be used. Execution of the **BNRP** instruction causes:

- The NMISCSR content is copied to the CSR.
- The PC loaded with the contents of the NRP.
- IER[1]NMIE bit is set.

8.2.4.18.7.3 SWI Interrupt

The **SWI** instruction is used to trigger the software interrupt. Software interrupts are always enabled. The SWI interrupt does not have a corresponding entry in the IFR, ICR, ISR, or IER, and it is not affected by the state of the CSR:GIE or IER[1]NMIE bits. An SWI does not have any delay slot. Decoding of SWI causes the following actions to be taken by the ARP32 CPU:

- The CSR:INUM field is updated with the SWI interrupt ID.
- The CSR content is copied to the SCSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are saved to the corresponding shadow registers.
- The CSR[0]GIE bit is cleared.
- The address of the next instruction (interrupt return address) to be stored to the IRP register.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to SWI IST location is sent and eventually the SWI IST entry is loaded into the PC. As a result, the CPU begins executing the SWI ISR.

To exit a SWI service routine, the **BIRP** instruction must be used. Execution of the **BIRP** instruction causes:

- The SCSR content is copied to the CSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are restored from the corresponding shadow registers.
- The PC loaded with the contents of the IRP.

8.2.4.18.7.4 Maskable Interrupts

The ARP32 CPU supports 12 maskable interrupts (INT15-INT4). For a maskable interrupt to be detected (and processed) by the ARP32 CPU:

- The global (maskable) interrupt enable (GIE) bit in the control status register (CSR) must be set to 1. The GIE bit is cleared to 0 at reset to prevent a maskable interrupt being taken prematurely (before system initialization). The GIE bit is set (to enable maskable interrupts) only after the system has been properly initialized and is ready to accept interrupts. The GIE bit is also cleared at the occurrence of an interrupt (NMI, INT15-INT4, SWI, and UNDEF) to prevent another interrupt from being processed; it may be set again in the ISR (to enable nested maskable interrupts) by software after the CPU state has been saved off properly.
- The IER:NMIE bit must be set.
- The corresponding enable bit in the interrupt enable register (IER) must also be set. IER bits allow enabling/disabling single maskable interrupt at a time.

When any of the IFR flag bits (IFR[15-4]) of maskable interrupts are set, (as a result of an maskable interrupt assertion via cpu_int[15-4]_i input pins), assuming the previous conditions are met, the ARP32 CPU accepts the highest priority interrupt and performs the following actions to process the interrupt:

- CPU stops fetching further instruction from instruction memory
- CPU execute pipe is allowed to drain
  - Any load/store instructions (including **LDRF, STRF, CALL, RET**) currently executing are allowed to complete
  - Any instruction in the branch/call/return delay slot is allowed to complete
- The CSR[11:8]INUM field is updated with the corresponding maskable interrupt ID.
- The CSR content is copied to the SCSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are saved to the corresponding shadow registers.
- The CSR[0]GIE bit is cleared.
- The associated bit in the IFR is cleared.
- The PC value of the next instruction to execute (after completion of the interrupt service routine) is stored to the IRP register. This is the interrupt return address.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to the corresponding IST location is sent and eventually the IST entry is loaded into the PC. As a result, the CPU begins executing the corresponding interrupt service routine.

To exit a maskable service routine, the **BIRP** instruction must be used. Execution of the **BIRP** instruction causes:

- The SCSR content is copied to the CSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are restored from the corresponding shadow registers.
- The PC loaded with the contents of the IRP.

8.2.4.18.7.5 UNDEF Interrupt

An UNDEF interrupt is triggered by either of the following conditions:

- The CPU decodes an undefined/unsupported instruction, an unmapped primary opcode in instruction word 7:0. This normally happen as a result of instruction memory getting corrupted. Note that not all kinds of invalid opcodes are detected by the CPU; for example, invalid fields other than the primary opcode of a valid instruction.
- The CPU executes a **DIV/DIVU/MOD/MODU** instruction where the divisor operand (*src1*) is 0.

An UNDEF interrupt causes the UNDEF IST entry to be loaded into the PC. UNDEF interrupts are always enabled, it does not have a corresponding entry in the IFR, ICR, ISR, or IER, and is unaffected by the state of the CSR[0]GIE or IER[1]NMIE bits.

The ARP32 CPU treats an UNDEF as an irrecoverable exception. It is not possible to correctly return from an UNDEF interrupt. The only way to recover from an UNDEF interrupt is to reset the CPU.

For example, when UNDEF is caused by an invalid instruction decode, the size of (undecoded) instruction under consideration is not determined unambiguously (since the ARP32 CPU supports freely mixable 16-bit and 32-bit instructions) and hence a recoverable return address is not saved to the IRP while taking an UNDEF interrupt.

The purpose of the UNDEF interrupt is to provide just the basic detection and diagnostic mechanism when such an irrecoverable fault occurs in the system. It is software responsibility to design such an UNDEF handler such that within the UNDEF handler context (without requiring to return from the UNDEF interrupt service routine), it can do diagnostic check(s), if any, and inform the external world (for example, a host CPU) that an irrecoverable fault/exception has happened within the ARP32 CPU that cannot be handled.

It is possible an **IDLE** instruction to be used in the UNDEF interrupt service routine to hold the processor in the UNDEF service routine, for debug purposes, and avoid returning to an incorrect instruction boundary (because of incorrect return address saved to IRP/NRP) causing further UNDEF(s).

Also a manipulation of the IRP to return to some known address (after clearing all machine state correctly, for example, CSR, SCSR), for example the C entry point (__cinit_00) with the expectation that the CPU practically re-executes from reset with all ISA initialization done freshly by the boot code at the entry point. However, it is still just a workaround considering that the root cause of instruction memory corrupt remains uncorrected.

When an undefined instruction is decoded by the ARP32 CPU it performs the following action:

- The CSR[11:8]INUM field is updated with the UNDEF interrupt ID.
- The CSR content is copied to the SCSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are saved to the corresponding shadow registers.
- The CSR[0]GIE bit is cleared.
- The address of the next instruction (interrupt return address) to be stored to the IRP register.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to the UNDEF IST location is sent and eventually the UNDEF IST entry is loaded into the PC. As a result, the CPU begins executing the UNDEF interrupt service routine.

To exit an UNDEF service routine, the **BIRP** instruction is used. Execution of the **BIRP** instruction causes:

- The SCSR content is copied to the CSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are restored from the corresponding shadow registers.
- The PC loaded with the contents of the IRP.

However, it is not assured to result in successfully returning to a correct instruction boundary. More UNDEF may follow immediately.

8.2.4.18.8 Interrupt Context Save and Restore

The ARP32 CPU supports extensive automatic context save and restore during interrupt processing to reduce the effective interrupt latency significantly.

While processing all maskable interrupts (INT15-INT4) , SWI and UNDEF, the ARP32 CPU saves off all key registers into shadow registers and during an interrupt return, via the **BIRP** instruction, restores them back:

- The Control Status Register (CSR) is saved to Shadow Control Status Register (SCSR)
- All architectural registers (R0-R7) are saved to corresponding shadow registers (SR0-SR7)
- The HLA setup registers (LSA*n*, LEA*n*, LCNT*n*, LCNT0RLD) are saved to the corresponding shadow registers (SLSAn, SLEAn, SLCNTn, SLCNT0RLD)

Both saving and restoring happens in a single cycle in parallel to interrupt vector fetch or the return branch execution. Also, as long as interrupt nesting is not enabled, there is absolutely no need of any additional context save (or restore) by the software at the start (or end) of the interrupt service routine. Hence, these interrupts have extremely low effective latency and are considered fast.

While processing the non-maskable interrupt (NMI), however, the CPU does not do extensive context save and restore. Only the CSR register is saved off to NMISCSR and during an NMI return, via the **BNRP** instruction, CSR is restored back. So, at the beginning of an NMI ISR, the software ensures to save off the complete CPU state to stack and restore them back before issuing a **BNRP** instruction (to accomplish a return from NMI):

- All architectural registers (R0-R7)
- The HLA setup registers (LSAn, LEAn, LCNTn, LCNT0RLD)

Thus, the interrupt behavior of the ARP32 CPU makes all maskable interrupts (INT15-INT4), SWI, and UNDEF as fast, and non-maskable interrupt (NMI) as just correct. The maskable interrupts (INT15-INT4) are used as the functional interrupt(s) of the system and the NMI are used to handle/service System Error, Exception scenarios (for example, a bus fault on program or data access).

The following example illustrates the process of context save and restore within an NMI service routine. In this example, the `NMIHandler()` is the actual software NMI handler function and a single unified stack across background and interrupt handler code is intended.

```
__NonMaskableInterruptHandlerWrapper:

        ; Save off Arch regs first
          STRF   R7, R0
        ; Save off HLA regs
          MVC    LSA0,     R0
          MVC    LEA0,     R1
          MVC    LCNT0,    R2
          MVC    LSA1,     R3
          MVC    LEA1,     R4
          MVC    LCNT1,    R5
          MVC    LCNT0RLD, R6
          STRF   R6, R0
        ; Now call the actual handler function
          CALL   NMIHandler
          NOP
        ; Restore HLA regs
          LDRF   R0, R6
          MVC    R0, LSA0
          MVC    R1, LEA0
          MVC    R2, LCNT0
          MVC    R3, LSA1
          MVC    R4, LEA1
          MVC    R5, LCNT1
          MVC    R6, LCNT0RLD
        ; Restore Arch regs
          LDRF   R0, R7
        ; Return from NMI ISR
          BNRP
          NOP
```

Note that in the example, during context restore, a write to LCNT0 also gets written to LCNT0RLD. Hence, the LCNT0RLD register is restored after LCNT0 is restored.

8.2.4.18.9 Nested Interrupts

When an interrupt (INT15–INT4, SWI, and UNDEF) is taken, the CSR[0]GIE bit is cleared by the hardware and subsequently the GIE bit is restored on interrupt return via the **BIRP** instruction. Thus all maskable interrupts (INT15–INT4) are disabled within the interrupt service routine of maskable interrupts.

Also, when an NMI is taken, the IER[1]NMIE bit is cleared by the hardware and subsequently the IER[1]NMIE bit is restored on interrupt return via the **BNRP** instruction. Thus NMI and all maskable interrupts (INT15–INT4) are disabled within the NMI service routine.

So in general, nesting of interrupts (except the case of NMI within a maskable interrupt routine) is disabled. However, under software control, it is possible that the ARP32 CPU is configured to take fully nested interrupts.

The ARP32 CPU offers two programming models with respect to usage of interrupts in a nested or non-nested manner.

8.2.4.18.9.1 Non-nested Interrupt Model

In this programming model, the CSR[0]GIE (or IER[1]NMIE) bit is not enabled within interrupts service routine of maskable interrupts (INT15–INT4), SWI, or UNDEF interrupt (or NMI). Hence, maskable interrupts (INT15–INT4), SWI, and UNDEF are not nested. However, an NMI is taken as a nested interrupt within a maskable interrupt (INT15–INT4), SWI, or UNDEF.

While adopting this programming model, the following points must be noted:

- It is the software responsibility to not use an **SWI** instruction within an ISR context. Otherwise, SWI corrupts the shadow registers and results in an unpredictable behavior.
- An UNDEF interrupts may be taken. UNDEF interrupts are irrecoverable exception and cannot be recovered from (see Section 8.2.4.18.7.5)
- NMI service routine follows the procedure described in Section 8.2.4.18.8. An NMIE must not be enabled within an NMI service routine as nesting of NMI itself is not supported.

Note that this programming model offers fast response time to system events via the maskable interrupt lines (INT15–INT4) by fully exploiting the extensive automatic context save/restore feature of the ARP32 CPU (see Section 8.2.4.18.8).

8.2.4.18.9.2 Nested Interrupt Model

In this programming model, the CSR[0]GIE bit is enabled within the interrupts service routine of maskable interrupts (INT15–INT4), SWI, or UNDEF interrupt. Hence, it is possible that maskable interrupts are nested. An NMI can always be taken with all other non-NMI interrupts.

Before enabling the CSR[0]GIE bit (in ISR), the software must ensure proper context save of all CPU registers onto the stack:

- Architectural registers (R0-R7)
- Shadow registers (SR0-SR7)
- Key control registers: SCSR, IRP/NRP
- Optional control register: SP, IER

While adopting this programming model, the following points must be noted:

- The SP needs to be saved, if a separate stack for the interrupt service routine is required
- IER needs to be saved, if a different interrupt mask is expected within an ISR
- Once the GIE bit is enabled, even the current interrupt (the once being serviced) becomes enabled to be re-entered. If reentrant behavior is not intended, IER must be appropriately modified after saving it off to the stack.
- NMI service routine must follow the procedure described in Section 8.2.4.18.8. An NMIE must not be enabled within an NMI service routine as nesting of NMI itself is not supported. All maskable interrupts (INT15–INT4) and NMI remain disabled within an NMI service routine.

The following example illustrates the process of context save and restore within a maskable interrupt service routine. In this example, it is assumed that the `InterruptHandler()` is the actual software interrupt handler function and a single unified stack across background and interrupt handler code is intended.

```
__NonMaskableInterruptHandlerWrapper:

        ; Save off Arch regs first
          STRF   R7, R0
        ; Save off HLA shadow regs
          MVS    SLSA0,     R0
          MVS    SLEA0,     R1
          MVS    SLCNT0,    R2
          MVS    SLSA1,     R3
          MVS    SLEA1,     R4
          MVS    SLCNT1,    R5
          MVS    SLCNT0RLD, R6
          STRF   R6, R0
        ; Save off SCSR, IRP
          MVC    SCSR,     R0
          MVC    IRP,      R1
          STRF   R1, R0
        ; Now call the actual handler function
          CALL   InterruptHandler
          NOP
        ; Restore SCSR, IRP
          LDRF   R0, R1
          MVC    R0, SCSR
          MVC    R1, IRP
        ; Restore HLA shadow regs
          LDRF   R0, R6
          MVS    R0, SLSA0
          MVS    R1, SLEA0
          MVS    R2, SLCNT0
          MVS    R3, SLSA1
          MVS    R4, SLEA1
          MVS    R5, SLCNT1
          MVS    R6, SLCNT0RLD
        ; Restore Arch regs into shadow copies
          LDRF   R0, R7
          MVS    R0, SR0
          MVS    R0, SR1
          MVS    R0, SR2
          MVS    R0, SR3
          MVS    R0, SR4
          MVS    R0, SR5
          MVS    R0, SR6
          MVS    R0, SR7
        ; Return from NMI ISR
          BIRP
          NOP
```

8.2.4.18.10 Non-nested Interrupt Latency

This section describes the best and worst case interrupt latencies in the non-nested interrupt programming model (described in Section 8.2.4.18.9.1).

8.2.4.18.10.1 Best Case Interrupt Latency

In the best case (and the most frequent case), the ARP32 CPU takes an interrupt as soon as (very next cycle) it is signaled at its boundary via the cpu_int[15-4]_i or cpu_nmi_i input ports. In this case, the ARP32 CPU takes as little as 5 cycles (from the cycle when the IFR bit gets set) to start executing the first instruction of the ISR:

- **Cycle 0:** cpu_int[15-4]_i or cpu_nmi_i assertion
- **Cycle 1:** corresponding IFR bit is set, interrupt enable conditions checked, interrupt priorities resolved, a winner interrupt is selected to be taken
- **Cycle 2:** cpu_iack_o, cpu_inum_o are asserted, program read request is sent to the corresponding IST address associate with the interrupt being serviced
- **Cycle 3:** program read data (IST content) is received and is loaded to PC
- **Cycle 4:** program read request is sent to the first instruction of the ISR routine
- **Cycle 5:** first instruction of the ISR routine is decoded
- **Cycle 6:** first instruction of the ISR routine is executed

Note that the ARP32 CPU saves off all architectural registers and other required CPU registers while taking the interrupt (in cycle 1) and as a result the ISR routine does not need to have any instructions to save off the machine state. The ISR routine directly starts executing code related to actual actions to be taken to service the interrupt.

8.2.4.18.10.2 Worst Case Interrupt Latency

Under certain cases, the ARP32 CPU blocks taking an interrupt, thus introducing variable interrupt latency. The following are the worst cases:

- When a load/store multiple (**LDRF, STRF**) instruction is at the EXE stage of the CPU pipeline and an interrupt is asserted. The interrupt is not taken until the **LDRF/STRF** instruction is completed, which takes several (variable) cycles.

  In compiler-generated code, **LDRF/STRF** is normally used as a function context save/restore across function calls; a maximum of three save-on-entry (SOE) registers as per the ARP32 CPU EABI function calling convention. Thus, three load/stores would need to finish before the CPU takes an interrupt. Moreover, if the stack pointer points to a wait-stated memory, this register save/restore takes even longer. Also, the program fetch request to the IST is stalled if the corresponding memory region is not zero-wait state, thus, increasing interrupt latency further.
- When a blocking, multicycle execute (**DIV/DIVU/MOD/MODU**) instruction is at the EXE stage of the CPU pipeline and an interrupt is asserted. The interrupt is not taken until the instruction at EXE is completed, which takes 13 (fixed) cycles.

8.2.A Instruction Set

This section provides a description of the instruction set. Unless otherwise explicitly mentioned, all instructions are a single execute cycle instruction. All instructions that are multicycle execute are mentioned explicitly.

8.2.A.1 Instruction Operation and Execution Notations

Table A-346 explains the symbols used in the instruction Pseudo Code descriptions.

All immediate values are unsigned (zero-extended) or signed (sign-extended) to match the relevant associated operand size. For example, zero (or sign) extended to 32 bits for operations with architectural registers (32 bit in size), zero-extended to 32 bits for operation with SP (32 bits in size), and zero-extended to the width of the PC for operations with the PC.

Although the instruction set architecture (ISA) does not contain a writable program counter (PC) register, in the description of some instructions (especially in the pseudo code), a conceptual register, PC, has been used. This is to simplify the description and does not imply existence of such a physical register.

Table A-346. Instruction Pseudo Code Notations

Symbol	Meaning
AND	bitwise AND
areg	architectural register (R0-R7)
baseR	base address register
creg	control register
dst	destination operand
GDP	global data pointer register
lmb0(x)	leftmost 0 bit search of x
lmb1(x)	leftmost 1 bit search of x
lsb*n*	n least-significant bits (for example, lsb32)
op*n*	field within opcode that specifies a unique instruction
OR	bitwise OR
PC	program counter
sat	saturate
SP	stack pointer register
src1	source operand 1
src2	source operand 2
sreg	shadow register
ucst*n*	n-bit unsigned immediate constant field (for example, ucst6)
XOR	bitwise exclusive-OR
+	addition
-	subtraction
×	multiplication
/	division
%	modulo
~	logical inverse
++	increment by 1
==	equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
<<	shift left
>>	shift right
&&	logical AND

8.2.A.2 Instruction Syntax and Opcode Notations

Table A-347 defines the syntaxes and opcode fields used in the instruction descriptions.

Table A-347. Instruction Syntax and Opcode Notations

Symbol	Meaning
areg	architectural register (R0-R7)
baseR	base address register
cc	condition code (LT, GT, EQ, LE, GE, NE)
creg	control register
dst	destination
GDP	global data pointer register
opn	opfield; field within opcode that specifies a unique instruction
scstn	n-bit signed constant field
sreg	shadow register
SP	stack pointer register
src1	source 1
src2	source 2
ucstn	n-bit unsigned constant field
x	don't care

8.2.A.3 Instruction Scheduling Restrictions

The ARP32 CPU architecture places the following restrictions on instruction scheduling that must be maintained. Whereas the compiler maintains these restrictions while generating assembly code for C/C++ programs, it is the responsibility of the programmer to maintain the restrictions for codes written directly in assembly.

8.2.A.3.1 Restrictions Applicable to a Branch Delay Slot

The following restrictions are applicable for an instruction in the delay slot of a branch (**Bcc** *scst9*, **Bcc** *scst16*, or **Bcc** *dst*), call (**CALL** *scst22* or **CALL** *src1*), and return (**RET**) instruction:

- 32-bit instructions are not allowed
- Another branch/call/return is not allowed
- **IDLE** instruction is not allowed
- **SWI** instruction is not allowed
- **LDRF/STRF** instructions with more than one register operand

Also, a branch delay slot cannot be the target of any branch/call/return.

8.2.A.3.2 Restrictions on Loops Using Hardware Loop Assist (HLA)

Loops setup using the HLA mechanism (Section 8.2.4.17) are required to maintain the following restrictions:

- Minimum number of instructions in Loop 0 is 2
- Branches (**Bcc/CALL/RET** + the delay slot instruction) cannot be placed at the end of a loop. This makes the loop rewind and branch target unambiguous.
- Last two instruction of any level cannot be a multi-slot load/store multiple (**LDRF/STRF**)
- Software cannot branch (**Bcc/CALL/RET**) to the last instruction of any loop, that is, the last instruction of any loop cannot be a branch target.

8.2.A.3.3 Restrictions on Other Types of Control Flow Instructions

The following restrictions are applicable for other types of control flow instructions:

- **SWI** instruction must be followed by a **NOP**
- **BIRP/BNRP** instructions must be followed by a **NOP**

8.2.A.3.4 Restrictions for Write Data Bypass to Control Register Reads

The following restrictions are applicable for write data bypass to control register reads:

- An **MVC** instruction that writes to a control register (**MVC** *areg, creg*; **MVC** *ucst16, creg*; or **MVCH** *ucst16, creg*), must not be followed by an **MVC** instruction (**MVC** *creg, areg*) that reads the same control register.
- An **SLA** instruction that writes to a control register (**SLA** *ucst16*, LSA*n* or **SLA** *ucst16*, LEA*n*) must not be followed by an **MVC** instruction (**MVC** *creg, areg*) that reads the same control register.
- An **SLA** instruction that writes to a control register (**SLA** *ucst16*, LSA*n* or **SLA** *ucst16*, LEA*n*) must not be followed by an **MVCH** instruction (**MVCH** *creg, areg*) that reads the same control register.

8.2.A.3.5 Restrictions for Write Data Bypass to Shadow Register Reads

The following restriction is applicable for write data bypass to shadow register reads:

- Thereare at least two instructions separating an **MVS** instruction writing to a shadow register (**MVS** *areg, sreg*) and another **MVS** instruction reading the same shadow register (**MVS** *sreg, areg*).

8.2.A.3.6 Restrictions for Link Register Update

The following restriction is applicable for link register update:

- Link registers cannot be read or written in the delay slot of a call (**CALL** *scst22* or **CALL** *src1*) and return (**RET**) instruction.

8.2.A.4 Instruction Set Encoding

The ARP32 ISA comprises of a mix of 16-bit and 32-bit instructions. An instruction packet is defined as a 32-bit field for a 32-bit instruction and a 16-bit field for a 16-bit instruction. The instruction decoder is sent an instruction packet every cycle.

8.2.A.5 Instruction Descriptions

Table A-348 lists the instructions for the ARP32 CPU.

Table A-348. Instruction Summary

Instruction	Description	Size
ABS src2, dst	Absolute value of a register	16-bit
ADD ucst16, src2, dst	Add 16-bit unsigned constant to a register	32-bit
ADD ucst16, SP	Add 16-bit unsigned constant to stack pointer, result to stack pointer	32-bit
ADD ucst16, SP, dst	Add 16-bit unsigned constant to stack pointer, result to register	32-bit
ADD ucst3, src2, dst	Add 3-bit unsigned constant to a register	16-bit
ADD src1, src2, dst	Signed addition of two register values	16-bit
AND ucst16, src2, dst	Bitwise AND 16-bit unsigned constant with a register	32-bit
AND src1, src2, dst	Bitwise AND two registers	16-bit
B dst	Indirect unconditional branch using a register	16-bit
BEQ dst	Indirect conditional branch using a register	16-bit
BGE dst	Indirect conditional branch using a register	16-bit
BGT dst	Indirect conditional branch using a register	16-bit
BLE dst	Indirect conditional branch using a register	16-bit
BLT dst	Indirect conditional branch using a register	16-bit
BNE dst	Indirect conditional branch using a register	16-bit
B scst9	Direct unconditional branch using a 9-bit signed constant offset	16-bit

Table A-348. Instruction Summary (continued)

Instruction	Description	Size
BEQ scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BGE scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BGT scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BLE scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BLT scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BNE scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
B scst16	Direct unconditional branch using a 16-bit signed constant offset	32-bit
BEQ scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BGE scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BGT scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BLE scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BLT scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BNE scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BIRP	Unconditional branch to interrupt return pointer	16-bit
BKPT	Software breakpoint	16-bit
BNRP	Unconditional branch to NMI return pointer	16-bit
CALL src1	Unconditional call using a register	16-bit
CALL scst22	Unconditional call using a 22-bit signed constant offset	32-bit
CLR ucst5_1, ucst5_2, src2, dst	Clear bit field bounded by two immediate values	32-bit
CLR src1, src2, dst	Clear bit field bounded by two register values	32-bit
CMP scst16, src2	Compare for equality, less than, greater than, 16-bit signed constant to a register	32-bit
CMP scst3, src2	Compare for equality, less than, greater than, 3-bit signed constant to a register	16-bit
CMP src1, src2	Signed compare for equality, less than, greater than, two register values	16-bit
CMPU ucst16, src2	Compare for equality, less than, greater than, 16-bit unsigned constant to a register	32-bit
CMPU ucst3, src2	Compare for equality, less than, greater than, 3-bit unsigned constant to a register	16-bit
CMPU src1, src2	Unsigned compare for equality, less than, greater than, two register values	16-bit
DIV src1, src2, dst	Signed division of two register values	32-bit
DIVU src1, src2, dst	Unsigned division of two register values	32-bit
EXT ucst5_1, ucst5_2, src2, dst	Extract and sign-extend a bit field bounded by two immediate values	32-bit
EXT src1, src2, dst	Extract and sign-extend a bit field bounded by two register values	32-bit
EXTU ucst5_1, ucst5_2, src2, dst	Extract and zero-extend a bit field bounded by two immediate values	32-bit
EXTU src1, src2, dst	Extract and zero-extend a bit field bounded by two register values	32-bit
IDLE	Idle until interrupt or reset	16-bit
LDB *+baseR[ucst3], dst	Load signed byte from memory with a 3-bit unsigned constant offset	16-bit
LDB *+baseR[ucst16], dst	Load signed byte from memory with a 16-bit unsigned constant offset	32-bit
LDB *+baseR[src1], dst	Load signed byte from memory with a register offset	16-bit
LDB *baseR++[ucst3], dst	Load signed byte from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDB *baseR++[src1], dst	Load signed byte from memory, postincrement memory with a register offset	16-bit
LDB *+SP[ucst6], dst	Load signed byte from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDB *+SP[ucst19], dst	Load signed byte from memory with a SP-relative 19-bit unsigned constant offset	32-bit

Table A-348. Instruction Summary (continued)

Instruction	Description	Size
LDB *+GDP[ucst19], dst	Load signed byte from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDBU *+baseR[ucst3], dst	Load unsigned byte from memory with a 3-bit unsigned constant offset	16-bit
LDBU *+baseR[ucst16], dst	Load unsigned byte from memory with a 16-bit unsigned constant offset	32-bit
LDBU *+baseR[src1], dst	Load unsigned byte from memory with a register offset	16-bit
LDBU *baseR++[ucst3], dst	Load unsigned byte from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDBU *baseR++[src1], dst	Load unsigned byte from memory, postincrement memory with a register offset	16-bit
LDBU *+SP[ucst6], dst	Load unsigned byte from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDBU *+SP[ucst19], dst	Load unsigned byte from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDBU *+GDP[ucst19], dst	Load unsigned byte from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDH *+baseR[ucst3], dst	Load signed halfword from memory with a 3-bit unsigned constant offset	16-bit
LDH *+baseR[ucst16], dst	Load signed halfword from memory with a 16-bit unsigned constant offset	32-bit
LDH *+baseR[src1], dst	Load signed halfword from memory with a register offset	16-bit
LDH *baseR++[ucst3], dst	Load signed halfword from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDH *baseR++[src1], dst	Load signed halfword from memory, postincrement memory with a register offset	16-bit
LDH *+SP[ucst6], dst	Load signed halfword from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDH *+SP[ucst19], dst	Load signed halfword from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDH *+GDP[ucst19], dst	Load signed halfword from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDHU *+baseR[ucst3], dst	Load unsigned halfword from memory with a 3-bit unsigned constant offset	16-bit
LDHU *+baseR[ucst16], dst	Load unsigned halfword from memory with a 16-bit unsigned constant offset	32-bit
LDHU *+baseR[src1], dst	Load unsigned halfword from memory with a register offset	16-bit
LDHU *baseR++[ucst3], dst	Load unsigned halfword from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDHU *baseR++[src1], dst	Load unsigned halfword from memory, postincrement memory with a register offset	16-bit
LDHU *+SP[ucst6], dst	Load unsigned halfword from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDHU *+SP[ucst19], dst	Load unsigned halfword from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDHU *+GDP[ucst19], dst	Load unsigned halfword from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDW *+baseR[ucst3], dst	Load word from memory with a 3-bit unsigned constant offset	16-bit
LDW *+baseR[ucst16], dst	Load word from memory with a 16-bit unsigned constant offset	32-bit
LDW *+baseR[src1], dst	Load word from memory with a register offset	16-bit
LDW *baseR++[ucst3], dst	Load word from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDW *baseR++[src1], dst	Load word from memory, postincrement memory with a register offset	16-bit
LDW *+SP[ucst6], dst	Load word from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDW *+SP[ucst19], dst	Load word from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDW *+GDP[ucst19], dst	Load word from memory with a GDP-relative 19-bit unsigned constant offset	32-bit

Table A-348. Instruction Summary (continued)

Instruction	Description	Size
LDRF op1, op2	Load register file from stack pointer	16-bit
LMBD ucst3, src2, dst	Left most bit detection	32-bit
MAX src1, src2, dst	Maximum of two signed register values	32-bit
MAXU src1, src2, dst	Maximum of two unsigned register values	32-bit
MIN src1, src2, dst	Minimum of two signed register values	32-bit
MINU src1, src2, dst	Minimum of two unsigned register values	32-bit
MOD src1, src2, dst	Signed modulo (32 % 32 = 32)	32-bit
MODU src1, src2, dst	Unsigned modulo (32 % 32 = 32)	32-bit
MPY src1, src2, dst	Signed multiplication of two register values (32-bit × 32-bit = 32-bit)	32-bit
MPYU src1, src2, dst	Unsigned multiplication of two register values (32-bit × 32-bit = 32-bit)	32-bit
MV src1, dst	Move register to register	16-bit
MVC areg, creg	Move architectural register to control register	16-bit
MVC creg, areg	Move control register to architectural register	16-bit
MVC ucst16, creg	Move 16-bit unsigned constant to control register	32-bit
MVCH ucst16, creg	Move 16-bit unsigned constant to upper bits of control register	32-bit
MVK ucst16, dst	Move 16-bit unsigned constant to register	32-bit
MVKH ucst16, dst	Move 16-bit unsigned constant to upper bits of register	32-bit
MVKLS scst16, dst	Move 16-bit signed constant to register	32-bit
MVKS scst6, dst	Move 6-bit signed constant to register	16-bit
MVS areg, sreg	Move architectural register to shadow register	16-bit
MVS sreg, areg	Move shadow register to architectural register	16-bit
NEG src2, dst	Negation	16-bit
NOP	No operation	16-bit
NOT src2, dst	Bitwise NOT	16-bit
OR ucst16, src2, dst	Bitwise OR 16-bit unsigned constant with a register	32-bit
OR src1, src2, dst	Bitwise OR two registers	16-bit
RET	Return from subroutine	16-bit
REV src1, src2, dst	Reverse a bit field bounded by two register values	32-bit
ROT src1, src2, dst	Rotate right	32-bit
ROTC src1, src2, dst	Rotate right through carry bit	32-bit
SADD src1, src2, dst	Signed addition of two register values with saturation	16-bit
SATN ucst3, src2, dst	Saturation to signed/unsigned *n*-bit value with sign/zero extend	32-bit
SET ucst5_1, ucst5_2, src2, dst	Set bit field bounded by two immediate values	32-bit
SET src1, src2, dst	Set bit field bounded by two register values	32-bit
SHL ucst5, src2, dst	Logical shift left by 5-bit unsigned constant	32-bit
SHL src1, src2, dst	Logical shift left by register value	16-bit
SHRA ucst5, src2, dst	Arithmetic shift right by 5-bit unsigned constant	32-bit
SHRA src1, src2, dst	Arithmetic shift right by register value	16-bit
SHRU ucst5, src2, dst	Logical shift right by 5-bit unsigned constant	32-bit
SHRU src1, src2, dst	Logical shift right by register value	16-bit
SLA ucst16, creg	Set loop address	32-bit
SSUB src1, src2, dst	Subtraction of two register values with saturation	16-bit
STB dst, *+baseR[ucst3]	Store byte to memory with a 3-bit unsigned constant offset	16-bit
STB dst, *+baseR[ucst16]	Store byte to memory with a 16-bit unsigned constant offset	32-bit
STB dst, *+baseR[src1]	Store byte to memory with a register offset	16-bit

Table A-348. Instruction Summary (continued)

Instruction	Description	Size
STB dst, *baseR++[ucst3]	Store byte to memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
STB dst, *baseR++[src1]	Store byte to memory, postincrement memory with a register offset	16-bit
STB dst, *+SP[ucst6]	Store byte to memory with a SP-relative 6-bit unsigned constant offset	16-bit
STB dst, *+SP[ucst19]	Store byte to memory with a SP-relative 19-bit unsigned constant offset	32-bit
STB dst, *+GDP[ucst19]	Store byte to memory with a GDP-relative 19-bit unsigned constant offset	32-bit
STH dst, *+baseR[ucst3]	Store halfword to memory with a 3-bit unsigned constant offset	16-bit
STH dst, *+baseR[ucst16]	Store halfword to memory with a 16-bit unsigned constant offset	32-bit
STH dst, *+baseR[src1]	Store halfword to memory with a register offset	16-bit
STH dst, *baseR++[ucst3]	Store halfword to memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
STH dst, *baseR++[src1]	Store halfword to memory, postincrement memory with a register offset	16-bit
STH dst, *+SP[ucst6]	Store halfword to memory with a SP-relative 6-bit unsigned constant offset	16-bit
STH dst, *+SP[ucst19]	Store halfword to memory with a SP-relative 19-bit unsigned constant offset	32-bit
STH dst, *+GDP[ucst19]	Store halfword to memory with a GDP-relative 19-bit unsigned constant offset	32-bit
STW dst, *+baseR[ucst3]	Store word to memory with a 3-bit unsigned constant offset	16-bit
STW dst, *+baseR[ucst16]	Store word to memory with a 16-bit unsigned constant offset	32-bit
STW dst, *+baseR[src1]	Store word to memory with a register offset	16-bit
STW dst, *baseR++[ucst3]	Store word to memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
STW dst, *baseR++[src1]	Store word to memory, postincrement memory with a register offset	16-bit
STW dst, *+SP[ucst6]	Store word to memory with a SP-relative 6-bit unsigned constant offset	16-bit
STW dst, *+SP[ucst19]	Store word to memory with a SP-relative 19-bit unsigned constant offset	32-bit
STW dst, *+GDP[ucst19]	Store word to memory with a GDP-relative 19-bit unsigned constant offset	32-bit
STHI ucst16, *+baseR[ucst6]	Store 16-bit halfword to memory with a 6-bit unsigned constant offset	32-bit
STRF op1, op2	Store register file to stack pointer	16-bit
SUB ucst16, src2, dst	Subtract 16-bit unsigned constant from a register	32-bit
SUB ucst16, SP	Subtract 16-bit unsigned constant from stack pointer, result to stack pointer	32-bit
SUB ucst16, SP, dst	Subtract 16-bit unsigned constant from stack pointer, result to register	32-bit
SUB ucst3, src2, dst	Subtract 3-bit unsigned constant from a register	16-bit
SUB src1, src2, dst	Signed subtraction of two register values	16-bit
SWI	Software interrupt	16-bit
XOR ucst16, src2, dst	Bitwise XOR 16-bit unsigned constant with a register	32-bit
XOR src1, src2, dst	Bitwise XOR two registers	16-bit

ABS

Absolute Value

Syntax

ABS *src2, dst*

Functional unit = L

Opcode

16 bit

15–13	12–10	9									0
dst	*src2*	0	0	0	1	0	0	0	0	0	0
3	3	*opcode*									

Description

The absolute value of *src2* is placed in *dst*. Note that *src2* is considered a signed 32-bit value.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = (src2 < 0)? -src2:src2
```

ADD

Add 16-Bit Unsigned Constant to Register

Syntax

ADD *ucst16, src2, dst*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*src2*	0	0	0	0	0	0	0	0	0	0
16	3	3	*opcode*									

Description

Addition of a 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (dst + {zero extend}ucst16)

CSR[7]V = {overflow} from (dst + {zero extend}ucst16)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 + {zero extend}ucst16
```

ADD

Add 16-Bit Unsigned Constant to Stack Pointer, Result to Stack Pointer

Syntax

ADD *ucst16,* **SP**

Functional unit = D

Opcode

32 bit

31 16	15	14	13	12	11	10	9	8	7							0
ucst16	0	0	1	1	1	x	x	x	0	0	0	0	0	0	0	1
16	*opcode*															

Description

Addition of a 16-bit unsigned constant (*ucst16*) with the stack pointer (SP) and store result to SP.

Condition Codes

None

Pseudo Code

```
SP = SP + ucst16
```

ADD

Add 16-Bit Unsigned Constant to Stack Pointer, Result to Register

Syntax

ADD *ucst16,* **SP** *, dst*

Functional unit = D

Opcode

32 bit

31		16	15	13	12	11	10	9	8	7						0
ucst16			*dst*		0	0	1	1	1	0	0	0	0	0	0	1	0
16			3		*opcode*											

Description

Addition of a 16-bit unsigned constant (*ucst16*) with the stack pointer (SP) and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (SP + ucst16)

CSR[7]V = {overflow} from (SP + ucst16)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = SP + ucst16
```

ADD

Add 3-Bit Unsigned Constant to Register

Syntax

ADD *ucst3, src2, dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*src2*	*ucst3*	0	1	1	0	0	0	0
3	3	3	*opcode*						

Description

Addition of a 3-bit unsigned constant (*ucst3*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (src2 + {zero extend}ucst3)

CSR[7]V = {overflow} from (src2 + {zero extend}ucst3)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 + {zero extend}ucst3
```

ADD

Signed Addition of Two Register Values

Syntax

ADD *src1, src2, dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	0	0	1	0
3	3	3	*opcode*						

Description

Signed addition of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (src2 + src1)

CSR[7]V = {overflow} from (src2 + src1)

Note that the carry (CSR[5]C) and overflow (CSR[7]V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

The status of the CSR[5]C bit is used to synthesize multiword (wider) addition. To synthesize a multiword addition, subsequent instructions use the CSR[5]C bit as a carry in operand, performing a normal addition if CSR[5]C == 0 and adding one more than usual if CSR[5]C == 1.

For example, if register pairs R0/R1 and R2/R3 hold 64-bit values (where R0 and R2 hold the least-significant words), the following instructions leave the 64-bit sum in the register pair R4/R5:

```
ADD   R2, R0, R4       ; R4=R2+R0, sets CSR[C]
MVC   CSR, R6          ; leaves CSR[C] untouched
EXTU  5, 5, R6, R6     ; if (CSR[C]) R6=0x1 else R6=0x0
ADD   R3, R1, R5       ; R5=R3+R1
ADD   R5, R6, R5       ; R5=R5+1 if CSR[C] was set, else R5=R5
```

Note that the above example needs appropriate modification in order to generate correct carry/overflow applicable for the total 64-bit result.

Pseudo Code

```
dst = src2 + src1
```

AND

Bitwise AND 16-Bit Unsigned Constant with Register

Syntax

AND *ucst16, src2, dst*

Functional unit = L

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*src2*	0	0	0	0	0	0	0	0	1	1
16	3	3	*opcode*									

Description

Bitwise AND of a zero-extended 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 AND {zero extend}ucst16
```

AND

Bitwise AND Two Registers

Syntax

AND *src1, src2, dst*

Functional unit = L

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*src2*	*src1*	0	1	1	1	0	1	0
3	3	3	*opcode*						

Description

Bitwise AND of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 AND src1
```

B(cc)

Indirect Branch Using a Register

Syntax

B(cc) *dst*

Functional unit = D

Opcode

16 bit

15 13	12	11	10	9 0
dst	x	x	x	*opcode*
3				10

Syntax	Opcode
B *dst*	00 0111 1110
BLT *dst*	01 0111 1110
BGT *dst*	10 1111 1110
BEQ *dst*	01 1111 1110
BLE *dst*	11 0111 1110
BGE *dst*	11 1111 1110
BNE *dst*	10 0111 1110

Description

Performs a branch to the address specified by the *dst* register. The contents of *dst* is treated as a byte address, it is converted to a halfword address before assigning to the PC. Note that the PC always contains a halfword address.

This instruction has one delay slot. See Section 8.2.A.3.1 for restrictions on scheduling an instruction in this delay slot.

For conditional branches, if the associated conditional (cc) bit is set in the control status register (CSR), the branch is taken; otherwise, the PC is incremented by 1. Valid values for ***cc*** are:

- <blank> = unconditional
- LT = CSR[3]LT bit is set to 1
- GT = CSR[4]GT bit is set to 1
- EQ = CSR[2]EQ bit is set to 1
- LE = CSR[3]LT and CSR:EQ bits are set to 1
- GE = CSR[4]GT and CSR:EQ bits are set to 1
- NE = CSR[2]EQ bit is cleared to 0

Condition Codes

None

Pseudo Code

```
if (cond)      PC = (*dst >> 1)
else           PC = PC + 1
```

B(cc)

Direct Branch Using a 9-Bit Signed Constant Offset

Syntax

B(cc) *scst9*

Functional unit = D

Opcode

16 bit

15 — 7	6 — 0
scst9	*opcode*
9	7

Syntax	Opcode
B *scst9*	100 1010
BLT *scst9*	100 1011
BGT *scst9*	100 1110
BEQ *scst9*	100 1100
BLE *scst9*	100 1111
BGE *scst9*	101 0000
BNE *scst9*	100 1101

Description

Performs a branch to the effective address formed by the signed addition of *scst9* with the current PC value. The offset is considered as a halfword offset. Note that the PC always contains a halfword address.

This instruction has one delay slot. See Section 8.2.A.3.1 for restrictions on scheduling an instruction in this delay slot.

For conditional branches, if the associated conditional (cc) bit is set in the control status register (CSR), the branch is taken; otherwise, the PC is incremented by 1. Valid values for ***cc*** are:

- <blank> = unconditional
- LT = CSR[3]LT bit is set to 1
- GT = CSR[4]GT bit is set to 1
- EQ = CSR[2]EQ bit is set to 1
- LE = CSR[3]LT and CSR:EQ bits are set to 1
- GE = CSR[4]GT and CSR:EQ bits are set to 1
- NE = CSR[2]EQ bit is cleared to 0

Condition Codes

None

Pseudo Code

```
if (cond)     PC = (PC + scst9)
else          PC = PC + 1
```

B(cc)

Direct Branch Using a 16-Bit Signed Constant Offset

Syntax

B(cc) *scst16*

Functional unit = D

Opcode

32 bit

31–16	15	14	13	12	11	10	9–0
scst16	x	x	x	x	x	x	*opcode*
16							10

Syntax	Opcode
B *scst16*	00 1000 1001
BLT *scst16*	01 0000 1001
BGT *scst16*	10 1000 1001
BEQ *scst16*	01 1000 1001
BLE *scst16*	11 0000 1001
BGE *scst16*	11 1000 1001
BNE *scst16*	10 0000 1001

Description

Performs a branch to the effective address formed by the signed addition of *scst16* with the current PC value. The offset is considered as a halfword offset. Note that the PC always contains a halfword address.

This instruction has one delay slot. See Section 8.2.A.3.1 for restrictions on scheduling an instruction in this delay slot.

For conditional branches, if the associated conditional (cc) bit is set in the control status register (CSR), the branch is taken; otherwise, the PC is incremented by 1. Valid values for ***cc*** are:

- <blank> = unconditional
- LT = CSR[3]LT bit is set to 1
- GT = CSR[4]GT bit is set to 1
- EQ = CSR[2]EQ bit is set to 1
- LE = CSR[3]LT and CSR:EQ bits are set to 1
- GE = CSR[4]GT and CSR:EQ bits are set to 1
- NE = CSR[2]EQ bit is cleared to 0

Condition Codes

None

Pseudo Code

```
if (cond)     PC = (PC + scst16)
else          PC = PC + 1
```

BIRP

Branch Using an Interrupt Return Pointer

Syntax

BIRP

Functional unit = D

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	1	0	0	1	1	1	1	1	1	1

opcode

Description

Performs an unconditional branch to the address in the interrupt return pointer register (IRP). The content of IRP is treated as a halfword address. The shadow control status register (SCSR) content is copied back to the control status register (CSR) as a result of this instruction.

This instruction has one delay slot. Only a NOP instruction is supported in a BIRP delay slot.

Condition Codes

For context restore: CSR = SCSR

Pseudo Code

```
PC = *IRP
CSR = SCSR, Rn = SRn, LSAn = SLSAn, LEAn = SLEAn, LCNTn = SLCNTn
```

BKPT

Software Breakpoint

Syntax

BKPT

Functional unit = S

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	0	1	1	1	1	1	1	1	1

opcode

Description

The **BKPT** instruction is equivalent to a single instruction endless loop. The ARP32 CPU re-fetches this same instruction again until released through the debug interface.

Condition Codes

None

Pseudo Code

None

BNRP

Branch Using NMI Return Pointer

Syntax

BNRP

Functional unit = D

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	1	0	1	1	1	1	1	1	1

opcode

Description

Performs an unconditional branch to the address in the nonmaskable interrupt return pointer register (NRP). The content of NRP is treated as a halfword address. The IER1 is set to 1 upon executing this instruction. The shadow control status register (SCSR) content is copied back to the control status register (CSR) as a result of this instruction.

This instruction has one delay slot. Only a NOP instruction is supported in a BNRP delay slot.

Condition Codes

For context restore: CSR = NMISCSR

Pseudo Code

```
PC = *NRP
CSR = NMISCSR
```

CALL

Call Using a Register

Syntax

CALL *src1*

Functional unit = D

Opcode

16 bit

15–13	12	11	10	9									0
src1	x	x	x	0	0	0	1	1	1	1	1	1	0
3	*opcode*												

Description

Store the link register (LR) to the location pointed to by the stack pointer (SP), decrement the SP, store PC + 2 into LR, then perform an unconditional branch to the address contained in *src1*. The contents of *src1* is treated as a byte address.

This instruction has one delay slot. See Section 8.2.A.3.1 for restrictions on scheduling an instruction in this delay slot.

Condition Codes

None

Pseudo Code

```
*(SP) = LR; SP -=4; LR = PC + 2; PC = *src1 >> 1
```

CALL

Call Using a 22-Bit Signed Constant Offset

Syntax

CALL *scst22*

Functional unit = D

Opcode

32 bit

31 … 10	9									0
scst22	0	0	0	0	0	0	1	0	0	1
22	*opcode*									

Description

Store the link register (LR) to the location pointed to by the stack pointer (SP), decrement the SP, store PC + 3 into LR, then perform an unconditional branch to the address specified by the PC relative immediate signed halfword offset (*scst22*).

This instruction has one delay slot. See Section 8.2.A.3.1 for restrictions on scheduling an instruction in this delay slot.

Condition Codes

None

Pseudo Code

```
*(SP) = LR; SP -=4; LR = PC + 3; PC = PC + scst22
```

CLR

Clear Bit Field Bounded by Two Immediate Values

Syntax

CLR *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25 21	20 16	15 13	12 10	9									0
x	x	x	x	x	x	*ucst5_1*	*ucst5_2*	*dst*	*src2*	1	1	1	0	0	0	0	0	0	0
						5	5	3	3	*opcode*									

Description

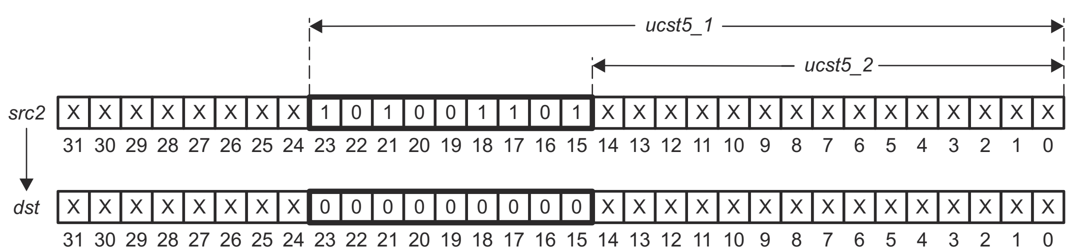
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is cleared to all 0s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be cleared to all 0s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 15 and *ucst5_1* is 23.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
 if ( i >= ucst5_2 && i <= ucst5_2) dst[i] = 0;
};
```

CLR

Clear Bit Field Bounded by Two Register Values

Syntax

CLR *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	1	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

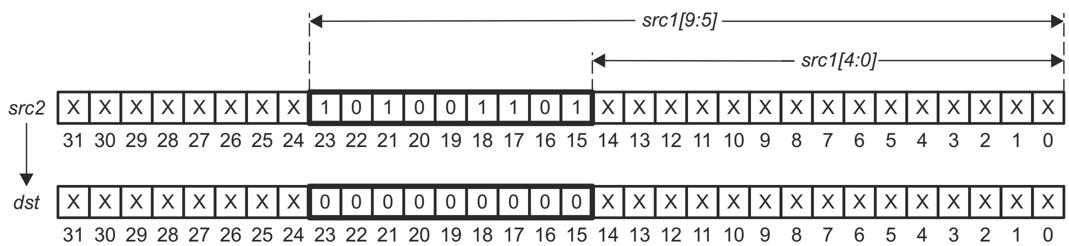
For *src1[9:5]* ≥ *src1[4:0*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is cleared to all 0s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be cleared to all 0s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 15 and *src1[9:5]* is 23.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
 if ( i >= src1[4:0] && i <= src1[9:5]) dst[i] = 0;
};
```

CMP

Compare for Equality, Less Than, Greater Than, 16-Bit Signed Constant to Register

Syntax

CMP *scst16, src2*

Functional unit = D

Opcode

32 bit

31 … 16	15	14	13	12 … 10	9									0
scst16	x	x	x	*src2*	0	1	0	0	0	0	0	0	0	0
16				3	*opcode*									

Description

Compares *src2* to a sign-extended 16-bit constant (*scst16*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison.

Condition Codes

CSR[2]EQ = (src2 == {sign extend}(scst16)

CSR[3]LT = (src2 < {sign extend}(scst16)

CSR[4]GT = (src2 > {sign extend}(scst16)

Pseudo Code

See Condition Codes

CMP

Compare for Equality, Less Than, Greater Than, 3-Bit Signed Constant to Register

Syntax

CMP *scst3, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12 – 10	9 – 7	6						0
x	x	x	*src2*	*scst3*	0	1	1	0	1	0	0
			3	3	*opcode*						

Description

Compares *src2* to a sign-extended 3-bit constant (*scst3*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison.

Condition Codes

CSR[2]EQ = (src2 == {sign extend}(scst3)

CSR[3]LT = (src2 < {sign extend}(scst3)

CSR[4]GT = (src2 > {sign extend}(scst3)

Pseudo Code

See Condition Codes

CMP

Signed Compare for Equality, Less Than, Greater Than, Two Register Values

Syntax

CMP *src1, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12–10	9–7	6						0
x	x	x	*src2*	*src1*	0	1	1	0	1	1	0
			3	3	*opcode*						

Description

Compares *src2* to *src1*. The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison.

Condition Codes

CSR[2]EQ = (src2 == src1)

CSR[3]LT = (src2 < src1)

CSR[4]GT = (src2 > src1)

Pseudo Code

See Condition Codes

CMPU

Compare for Equality, Less Than, Greater Than, 16-Bit Unsigned Constant to Register

Syntax

CMPU *ucst16, src2*

Functional unit = D

Opcode

32 bit

31	16	15	14	13	12	10	9									0
ucst16		x	x	x	*src2*		0	1	1	0	0	0	0	0	0	0
16					3		*opcode*									

Description

Compares *src2* to a zero-extended 16-bit constant (*ucst16*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison. This is an unsigned comparison.

Condition Codes

CSR[2]EQ = (src2 == {zero extend}(ucst16)

CSR[3]LT = (src2 < {zero extend}(ucst16)

CSR[4]GT = (src2 > {zero extend}(ucst16)

Pseudo Code

See Condition Codes

CMPU

Compare for Equality, Less Than, Greater Than, 3-Bit Unsigned Constant to Register

Syntax

CMPU *ucst3, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12–10	9–7	6						0
x	x	x	*src2*	*ucst3*	0	1	1	0	1	0	1
			3	3	*opcode*						

Description

Compares *src2* to a zero-extended 3-bit constant (*ucst3*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison. This is an unsigned comparison.

Condition Codes

CSR[2]EQ = (src2 == {zero extend}(ucst3)

CSR[3]LT = (src2 < {zero extend}(ucst3)

CSR[4]GT = (src2 > {zero extend}(ucst3)

Pseudo Code

See Condition Codes

CMPU

Unsigned Compare for Equality, Less Than, Greater Than, Two Register Values

Syntax

CMPU *src1, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12–10	9–7	6						0
x	x	x	*src2*	*src1*	0	1	1	0	1	1	1
			3	3	*opcode*						

Description

Compares *src2* to *src1*. The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison. This is an unsigned comparison.

Condition Codes

CSR[2]EQ = (src2 == src1)

CSR[3]LT = (src2 < src1)

CSR[4]GT = (src2 > src1)

Pseudo Code

See Condition Codes

DIV

Signed Division of Two Register Values

Syntax

DIV *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 – 13	12 – 10	9 – 7	6						0
x	x	x	x	x	x	x	x	x	x	x	0	1	0	0	1	*dst*	*src2*	*src1*	0	0	0	0	1	0	1
																3	3	3	*opcode*						

Description

Signed division of *src1* and *src2* and store result to *dst*. Divide by 0 raises UNDEF interrupt, *dst* is written with 0, CSR:EQ gets set.

Using **DIV** (integer division) and **SHRA** (arithmetic right shift) does not produce the same result for negative numbers. The quotient of **DIV** is rounded towards zero, whereas the quotient of **SHRA** is rounded towards negative infinity. For example, using the **DIV** instruction: -9/4 = -2, whereas using the **SHRA** instruction: -9/4 = -3.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 / src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

DIVU

Unsigned Division of Two Register Values

Syntax

DIVU *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 13 12 10 9 7 6 0

x	x	x	x	x	x	x	x	x	x	x	0	1	1	0	1	*dst*	*src2*	*src1*	0	0	0	0	1	0	1
																3	3	3				*opcode*			

Description

Unsigned division of *src1* and *src2* and store result to *dst*. Divide by 0 raises UNDEF interrupt, *dst* is written with 0, CSR:EQ gets set.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 / src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

EXT

Extract and Sign-Extend a Bit Field Bounded by Two Immediate Values

Syntax

EXT *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25–21	20–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	*ucst5_1*	*ucst5_2*	*dst*	*src2*	1	0	0	0	0	0	0	0	0	0
						5	5	3	3	*opcode*									

Description

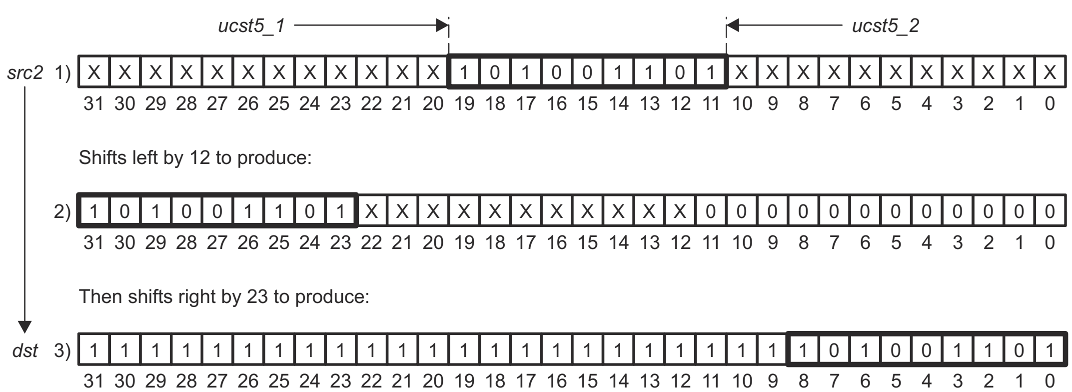
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is extracted and sign-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field to be extracted. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 11 and *ucst5_1* is 19.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= ucst5_2 && i <= ucst5_1) tmpdst[j++] = src2[i];
};
dst = tmpdst[j] ? signext(tmpdst) : tmpdst ;
```

EXT

Extract and Sign-Extend a Bit Field Bounded by Two Register Values

Syntax

EXT *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20				16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	1	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

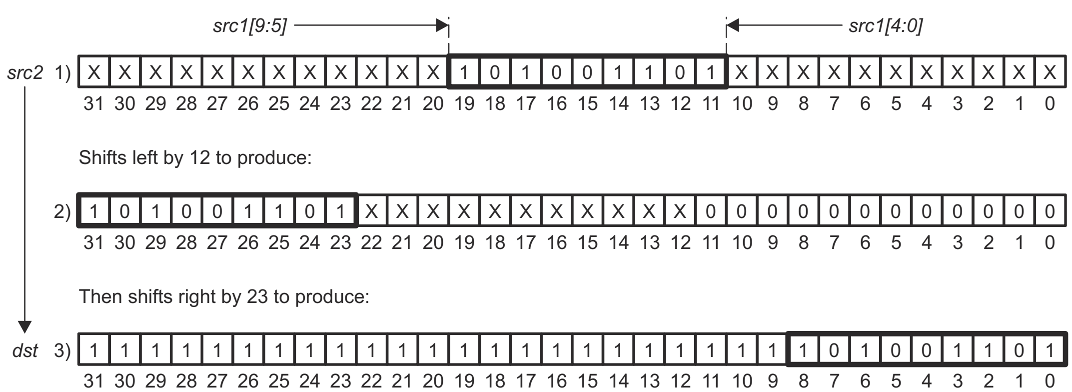
For *src1[9:5]* ≥ *src1[4:0]*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is extracted and sign-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field to be extracted. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5]* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 11 and *src1[9:5]* is 19.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= src1[4:0] && i <= src1[9:5]) tmpdst[j++] = src2[i];
};
dst = tmpdst[j] ? signext(tmpdst) : tmpdst ;
```

EXTU

Extract and Zero-Extend a Bit Field Bounded by Two Immediate Values

Syntax

EXTU *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25–21	20–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	*ucst5_1*	*ucst5_2*	*dst*	*src2*	1	0	1	0	0	0	0	0	0	0
						5	5	3	3	*opcode*									

Description

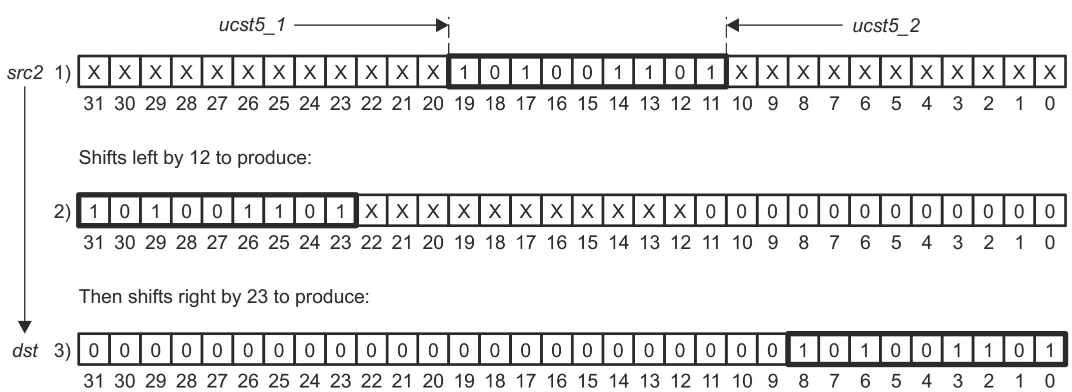
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is extracted and zero-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field to be extracted. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 11 and *ucst5_1* is 19.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= ucst5_2 && i <= ucst5_1) tmpdst[j++] = src2[i];
};
dst = tmpdst ;
```

EXTU

Extract and Zero-Extend a Bit Field Bounded by Two Register Values

Syntax

EXTU *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20				16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	1	1	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

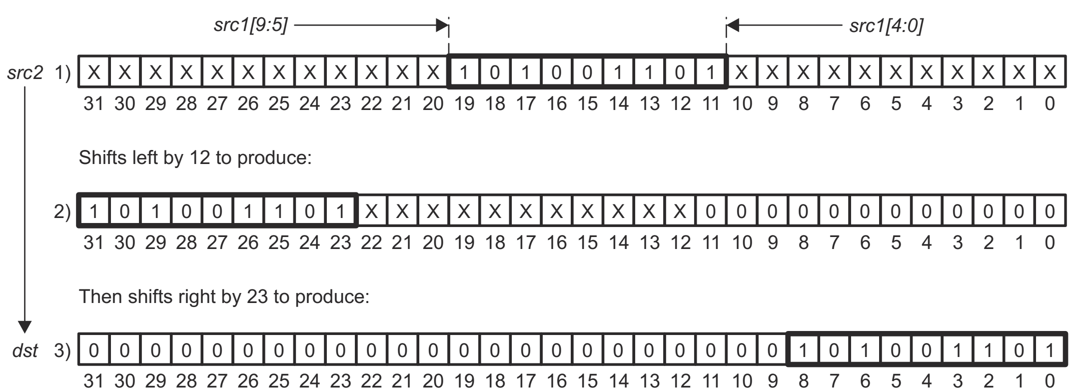
For *src1[9:5]* ≥ *src1[4:0]*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is extracted and zero-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field to be extracted. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5]* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 11 and *src1[9:5]* is 19.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= src1[4:0] && i <= src1[9:5]) tmpdst[j++] = src2[i];
};
dst = tmpdst ;
```

IDLE

Idle Until Interrupt or Reset

Syntax

IDLE

Functional unit = S

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	1	1	0	1	1	1	1	1	1	1

opcode

Description

The **IDLE** instruction causes the CPU to wait for all pending instruction and data memory transactions to complete and then go to an idle state. The idle state is the equivalent of an endless loop. Termination of this loop occurs on an interrupt or reset. However, the CPU does not actually continue fetch/executing any actual instructions. On acceptance of an interrupt, PC + 1 is stored into the interrupt return pointer register (IRP), subsequent instruction execution after return from the interrupt begins at the instruction following the **IDLE** instruction.

For more details on usage of IDLE instruction, see Section 8.2.B.3.

Condition Codes

None

Pseudo Code

None

LDB(U)

Load Byte from Memory with a 3-Bit Unsigned Constant Offset

Syntax

LDB **+baseR[ucst3], dst*

or

LDBU **+baseR[ucst3], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6 0
dst	*baseR*	*ucst3*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDB **+baseR[ucst3], dst*	101 1010	Load byte (sign-extended)
LDBU **+baseR[ucst3], dst*	101 1000	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero.There must be brackets, [ ], around the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 0 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst3])
```

LDB(U)

Load Byte from Memory with a 16-Bit Unsigned Constant Offset

Syntax

LDB **+baseR[ucst16], dst*

or

LDBU **+baseR[ucst16], dst*

Functional unit = D

Opcode

32 bit

31	16 15	13 12	10 9 0
ucst16	*dst*	*baseR*	*opcode*
16	3	3	10

Syntax	Opcode	Load Type
LDB **+baseR[ucst16], dst*	01 0000 1000	Load byte (sign-extended)
LDBU **+baseR[ucst16], dst*	00 0000 1000	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets,[], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 0 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst16])
```

LDB(U)

Load Byte from Memory with a Register Offset

Syntax

LDB **+baseR[src1], dst*

or

LDBU **+baseR[src1], dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6–0
dst	*baseR*	*src1*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDB **+baseR[src1], dst*	101 0101	Load byte (sign-extended)
LDBU **+baseR[src1], dst*	101 0011	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 0 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[src1])
```

LDB(U)

Load Byte from Memory, Postincrement with a 3-Bit Unsigned Constant Offset

Syntax

LDB **baseR++[ucst3], dst*

or

LDBU **baseR++[ucst3], dst*

Functional unit = D

Opcode

16 bit

15 — 13	12 — 10	9 — 7	6 — 0
dst	*baseR*	*ucst3*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDB **baseR++[ucst3], dst*	110 0100	Load byte (sign-extended)
LDBU **baseR++[ucst3], dst*	110 0010	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must be around the specified offset, if using the optional offset parameter.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[ucst3])
```

LDB(U)

Load Byte from Memory, Postincrement with a Register Offset

Syntax

LDB **baseR++[src1], dst*

or

LDBU **baseR++[src1], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6 0
dst	*baseR*	*src1*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDB **baseR++[src1], dst*	101 1111	Load byte (sign-extended)
LDBU **baseR++[src1], dst*	101 1101	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of bytes) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[src1])
```

LDB(U)

Load Byte from Memory with a SP-Relative 6-Bit Unsigned Constant Offset

Syntax

LDB *+**SP** *[ucst6], dst*

or

LDBU *+**SP** *[ucst6], dst*

Functional unit = D

Opcode

16 bit

15 13	12 7	6 0
dst	*ucst6*	*opcode*
3	6	7

Syntax	Opcode	Load Type
LDB *+**SP** *[ucst6], dst*	110 1001	Load byte (sign-extended)
LDBU *+**SP** *[ucst6], dst*	110 0111	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 0 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst6])
```

LDB(U)

Load Byte from Memory with a SP-Relative 19-Bit Unsigned Constant Offset

Syntax

LDB ***+SP** *[ucst19], dst*

or

LDBU ***+SP** *[ucst19], dst*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9–0
ucst19	*dst*	*ucst19*	*opcode*
16	3	3	10

Syntax	Opcode	Load Type
LDB ***+SP** *[ucst19], dst*	01 0000 0111	Load byte (sign-extended)
LDBU ***+SP** *[ucst19], dst*	00 0000 0111	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst19])
```

LDB(U)

Load Byte from Memory with a GDP-Relative 19-Bit Unsigned Constant Offset

Syntax

LDB ***+GDP** *[ucst19], dst*

or

LDBU ***+GDP** *[ucst19], dst*

Functional unit = D

Opcode

32 bit

31 ... 16	15 ... 13	12 ... 10	9 ... 0
ucst19	*dst*	*ucst19*	*opcode*
16	3	3	10

Syntax	Opcode	Load Type
LDB ***+GDP** *[ucst19], dst*	01 0000 0110	Load byte (sign-extended)
LDBU ***+GDP** *[ucst19], dst*	00 0000 0110	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the global data pointer register (GDP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(GDP[ucst19])
```

LDH(U)

Load Halfword from Memory with a 3-Bit Unsigned Constant Offset

Syntax

LDH **+baseR[ucst3], dst*

or

LDHU **+baseR[ucst3], dst*

Functional unit = D

Opcode

16 bit

15 – 13	12 – 10	9 – 7	6 – 0
dst	*baseR*	*ucst3*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDH **+baseR[ucst3], dst*	101 1011	Load halfword (sign-extended)
LDHU **+baseR[ucst3], dst*	101 1001	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 1 bit. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst3])
```

LDH(U)

Load Halfword from Memory with a 16-Bit Unsigned Constant Offset

Syntax

LDH **+baseR[ucst16], dst*

or

LDHU **+baseR[ucst16], dst*

Functional unit = D

Opcode

32 bit

31 ... 16	15 ... 13	12 ... 10	9 ... 0
ucst16	*dst*	*baseR*	*opcode*
16	3	3	10

Syntax	Opcode	Load Type
LDH **+baseR[ucst16], dst*	01 1000 1000	Load halfword (sign-extended)
LDHU **+baseR[ucst16], dst*	00 1000 1000	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 1 bit. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst16])
```

LDH(U)

Load Halfword from Memory with a Register Offset

Syntax

LDH **+baseR[src1], dst*

or

LDHU **+baseR[src1], dst*

Functional unit = D

Opcode

16 bit

15 — 13	12 — 10	9 — 7	6 — 0
dst	*baseR*	*src1*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDH **+baseR[src1], dst*	101 0110	Load halfword (sign-extended)
LDHU **+baseR[src1], dst*	101 0100	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 1 bit. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[src1])
```

LDH(U)

Load Halfword from Memory, Postincrement with a 3-Bit Unsigned Constant Offset

Syntax

LDH **baseR++[ucst3], dst*

or

LDHU **baseR++[ucst3], dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6–0
dst	*baseR*	*ucst3*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDH **baseR++[ucst3], dst*	110 0101	Load halfword (sign-extended)
LDHU **baseR++[ucst3], dst*	110 0011	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[ucst3])
```

LDH(U)

Load Halfword from Memory, Postincrement with a Register Offset

Syntax

LDH **baseR++[src1], dst*

or

LDHU **baseR++[src1], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6 0
dst	*baseR*	*src1*	*opcode*
3	3	3	7

Syntax	Opcode	Load Type
LDH **baseR++[src1], dst*	110 0000	Load halfword (sign-extended)
LDHU **baseR++[src1], dst*	101 1110	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of halfwords) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[src1])
```

LDH(U)

Load Halfword from Memory with a SP-Relative 6-Bit Unsigned Constant Offset

Syntax

LDH *+**SP** *[ucst6], dst*

or

LDHU *+**SP** *[ucst6], dst*

Functional unit = D

Opcode

16 bit

15 — 13	12 — 7	6 — 0
dst	*ucst6*	*opcode*
3	6	7

Syntax	Opcode	Load Type
LDH *+**SP** *[ucst6], dst*	110 1010	Load halfword (sign-extended)
LDHU *+**SP** *[ucst6], dst*	110 1000	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 1 bit. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst6])
```

LDH(U)

Load Halfword from Memory with a SP-Relative 19-Bit Unsigned Constant Offset

Syntax

LDH *+**SP** *[ucst19], dst*

or

LDHU *+**SP** *[ucst19], dst*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9–0
ucst19	*dst*	*ucst19*	*opcode*
16	3	3	10

Syntax	Opcode	Load Type
LDH *+**SP** *[ucst19], dst*	01 1000 0111	Load halfword (sign-extended)
LDHU *+**SP** *[ucst19], dst*	00 1000 0111	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst19])
```

LDH(U)

Load Halfword from Memory with a GDP-Relative 19-Bit Unsigned Constant Offset

Syntax

LDH ***+GDP** *[ucst19], dst*

or

LDHU ***+GDP** *[ucst19], dst*

Functional unit = D

Opcode

32 bit

31 ... 16	15 ... 13	12 ... 10	9 ... 0
ucst19	*dst*	*ucst19*	*opcode*
16	3	3	10

Syntax	Opcode	Load Type
LDH ***+GDP** *[ucst19], dst*	01 1000 0110	Load halfword (sign-extended)
LDHU ***+GDP** *[ucst19], dst*	00 1000 0110	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the global data pointer register (GDP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(GDP[ucst19])
```

LDW

Load Word from Memory with a 3-Bit Unsigned Constant Offset

Syntax

LDW **+baseR[ucst3], dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	0	1	1	1	0	0
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 2 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst3])
```

LDW

Load Word from Memory with a 16-Bit Unsigned Constant Offset

Syntax

LDW **+baseR[ucst16], dst*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*baseR*	1	0	0	0	0	0	1	0	0	0
16	3	3	*opcode*									

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 2 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst16])
```

LDW

Load Word from Memory with a Register Offset

Syntax

LDW **+baseR[src1], dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	0	1	0	1	1	1
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 2 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[src1])
```

LDW

Load Word from Memory, Postincrement with a 3-Bit Unsigned Constant Offset

Syntax

LDW **baseR++[ucst3], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*ucst3*	1	1	0	0	1	1	0
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of words) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[ucst3])
```

LDW

Load Word from Memory, Postincrement with a Register Offset

Syntax

LDW **baseR++[src1], dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	1	0	0	0	0	1
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of words) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[src1])
```

LDW

Load Word from Memory with a SP-Relative 6-Bit Unsigned Constant Offset

Syntax

LDW *+**SP** *[ucst6], dst*

Functional unit = D

Opcode

16 bit

15–13	12–7	6						0
dst	*ucst6*	1	1	0	1	0	1	1
3	6	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 2 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst6])
```

LDW

Load Word from Memory with a SP-Relative 19-Bit Unsigned Constant Offset

Syntax

LDW *+**SP** *[ucst19], dst*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst19	*dst*	*ucst19*	1	0	0	0	0	0	0	1	1	1
16	3	3	*opcode*									

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst19])
```

LDW

Load Word from Memory with a GDP-Relative 19-Bit Unsigned Constant Offset

Syntax

LDW *+**GDP** *[ucst19], dst*

Functional unit = D

Opcode

32 bit

31 ... 16	15 ... 13	12 ... 10	9									0
ucst19	*dst*	*ucst19*	1	0	0	0	0	0	0	1	1	0
16	3	3	*opcode*									

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from the global data pointer register (GDP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(GDP[ucst19])
```

LDRF

Load Register File from Stack Pointer

Syntax

LDRF *op1, op2*

Functional unit = D

Opcode

16 bit

15–13	12–10	9	8	7	6	5	4	3	2	1	0
op2	*op1*	x	x	x	1	0	1	0	0	0	1
3	3				*opcode*						

Description

The current base register file is loaded from the starting address specified by the preincremented value of the stack pointer (SP). *op1* specifies the ending register (last register loaded) and *op2* specifies the beginning register (first register loaded). All registers between *op1* and *op2* are loaded inclusive. The registers are loaded in order from highest register (*op2*) to lowest register (*op1*).

Valid values of *op1* and *op2* are: *op1* ≤ *op2*

For *op1* > *op2*, results in unspecified behavior.

This is a multicycle blocking instruction. The number of cycles is determined by the registers, for example, **LDRF** R0, R5 requires 6 cycles to complete.

Condition Codes

None

Pseudo Code

```
for ( i = op2; i <= op1 ; i--) {
SP += 4;
Ri = *SP;
}
```

LMBD

Left Most Bit Detection

Syntax

LMBD *ucst3, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18 16	15 13	12 10	9									0
x	x	x	x	x	x	x	x	x	x	x	x	x	*ucst3*	*dst*	*src2*	1	1	0	0	0	0	0	0	1	1
													3	3	3	*opcode*									

Description

The LSB of *ucst3* determines whether to search for a leftmost 1 or 0 in *src2*. The number of bits to the left of the first 1 or 0 when searching for a 1 or 0, respectively, is placed in *dst*.

- *ucst3* = 0, searches for leftmost 0 (count leading ones)
- *ucst3* = 1, searches for leftmost 1 (count leading zeros)
- Other values of *ucst3* are reserved. The result is undefined if reserved values are used.

The following diagrams illustrate the operation of **LMBD** for several cases.

When searching for 0 in *src2*, **LMBD** returns 0:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

When searching for 1 in *src2*, **LMBD** returns 4:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

When searching for 0 in *src2*, **LMBD** returns 32:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Condition Codes

CSR:EQ = (dst == 0)

LMBD (continued)

Left Most Bit Detection

Pseudo Code

```
dst = if (ucst3 == 0), lmb0(src2)
dst = if (ucst3 == 1), lmb1(src2)
```

MAX

Maximum of Two Signed Register Values

Syntax

MAX *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	13 12	10 9	7 6						0
x	x	x	x	x	x	x	x	x	x	x	1	1	0	0	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

Performs a maximum operation on the signed values in *src1* and *src2*. The larger value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 > src1) ? src2:src1
```

MAXU

Maximum of Two Unsigned Register Values

Syntax

MAXU *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	1	0	1	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3				*opcode*			

Description

Performs a maximum operation on the unsigned values in *src1* and *src2*. The larger value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 > src1) ? src2:src1
```

MIN

Minimum of Two Signed Register Values

Syntax

MIN *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	1	1	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

Performs a minimum operation on the signed values in *src1* and *src2*. The smaller value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 < src1) ? src2:src1
```

MINU

Minimum of Two Unsigned Register Values

Syntax

MINU *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	1	0	0	1	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

Performs a minimum operation on the unsigned values in *src1* and *src2*. The smaller value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 < src1) ? src2:src1
```

MOD

Signed Modulo

Syntax

MOD *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	0	1	0	1	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	1
																3	3	3	*opcode*						

Description

Performs a signed modulo on the values in *src1* and *src2* and the resulting value is written to *dst*. Sign of the result is the same as *src2* (the dividend). *src1* = 0 (that is, a modulo 0 case), raises the UNDEF interrupt, *dst* is written with 0, and CSR:EQ is set.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 % src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

MODU

Unsigned Modulo

Syntax

MODU *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–13	12–10	9–7	6						0
x	x	x	x	x	x	x	x	x	x	x	0	1	1	1	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	1
																3	3	3	*opcode*						

Description

Performs an unsigned modulo on the values in *src1* and *src2* and the resulting value is written to *dst*. *src1* = 0 (that is, a modulo 0 case), raises the UNDEF interrupt, *dst* is written with 0, and CSR:EQ is set.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 % src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

MPY

Signed Multiplication of Two Register Values

Syntax

MPY *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	0	1	0	0	0		*dst*	*src2*	*src1*	0	0	0	0	1	0	1
																	3	3	3				*opcode*			

Description

Performs a multiply on the signed values in *src1* and *src2*. Only the 32 least-significant bits of the result are retained and written to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = lsb32(src2 × src1)
```

MPYU

Unsigned Multiplication of Two Register Values

Syntax

MPYU *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	0	1	1	0	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	1
																3	3	3	*opcode*						

Description

Performs a multiply on the unsigned values in *src1* and *src2*. Only the 32 least-significant bits of the result are retained and written to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = lsb32(src2 × src1)
```

MV

Move Register to Register

Syntax

MV *src1, dst*

Functional unit = S

Opcode

16 bit

15–13	12	11	10	9–7	6						0
dst	x	x	x	*src1*	1	0	0	1	0	0	1
3				3	*opcode*						

Description

The contents of *src1* are copied to *dst*.

Condition Codes

None

Pseudo Code

```
dst = src1
```

MVC

Move Architectural Register to Control Register

Syntax

MVC *areg, creg*

Functional unit = S

Opcode

16 bit

15–11	10	9–7	6						0
creg	0	*areg*	1	0	0	1	0	0	0
5		3	*opcode*						

Description

The content of the architectural register (*areg*) is moved to the control register (*creg*).

Condition Codes

None

Pseudo Code

```
creg = areg
```

MVC

Move Control Register to Architectural Register

Syntax

MVC *creg, areg*

Functional unit = S

Opcode

16 bit

15 13	12 8	7							0
areg	*creg*	0	1	0	0	0	1	1	1
3	5	*opcode*							

Description

The content of the control register (*creg*) is moved to the architectural register (*areg*).

Condition Codes

None

Pseudo Code

```
areg = creg
```

MVC

Move 16-Bit Unsigned Constant to Control Register

Syntax

MVC *ucst16, creg*

Functional unit = S

Opcode

32 bit

31–16	15–11	10	9	8	7	6	5	4	3	2	1	0
ucst16	*creg*	x	1	0	0	0	0	0	1	0	1	0
16	5	*opcode*										

Description

A 16-bit unsigned constant (*ucst16*) is moved to the control register (*creg*).

Condition Codes

None

Pseudo Code

```
creg = {zero extend}ucst16
```

MVCH

Move 16-Bit Unsigned Constant to Upper Bits of Control Register

Syntax

MVCH *ucst16, creg*

Functional unit = S

Opcode

32 bit

31 – 16	15	14	13	12 – 8	7	6						0
ucst16	x	x	x	*creg*	x	0	0	0	1	1	0	0
16				5					*opcode*			

Description

A 16-bit unsigned constant (*ucst16*) is moved to the upper halfword (bits 31-16) of the control register (*creg*). The lower halfword (bits 15-0) of *creg* is left unchanged.

Condition Codes

None

Pseudo Code

```
creg[31:16] = ucst16
```

MVK

Move 16-Bit Unsigned Constant to Register

Syntax

MVK *ucst16, dst*

Functional unit = S

Opcode

32 bit

31 … 16	15 … 13	12	11	10	9									0
ucst16	*dst*	x	x	x	0	0	0	0	0	0	1	0	1	0
16	3				*opcode*									

Description

A 16-bit unsigned constant (*ucst16*) is moved to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = {zero extend}ucst16
```

MVKH

Move 16-Bit Unsigned Constant to Upper Bits of Register

Syntax

MVKH *ucst16, dst*

Functional unit = S

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*dst*	0	0	1	0	0	0	1	0	1	0
16	3	3	*opcode*									

Description

A 16-bit unsigned constant (*ucst16*) is moved to the upper halfword (bits 31-16) of *dst*. The lower halfword (bits 15-0) of *dst* is left unchanged.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst[31:16] = ucst16
```

MVKLS

Move 16-Bit Signed Constant to Register

Syntax

MVKLS *scst16, dst*

Functional unit = S

Opcode

32 bit

31	16	15	13	12	11	10	9									0
scst16		*dst*		x	x	x	0	1	0	0	0	0	1	0	1	0
16		3		*opcode*												

Description

A 16-bit constant (*scst16*) is sign extended and is moved to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = {sign extend}scst16
```

MVKS

Move 6-Bit Signed Constant to Register

Syntax

MVKS *scst6, dst*

Functional unit = S

Opcode

16 bit

15–13	12–7	6						0
dst	*scst6*	1	0	0	0	1	1	0
3	6	*opcode*						

Description

A 6-bit signed constant (*scst6*) is sign extended and is moved to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = {sign extend}scst6
```

MVS

Move Architectural Register to Shadow Register

Syntax

MVS *areg, sreg*

Functional unit = S

Opcode

16 bit

15 – 11	10	9 – 7	6						0
sreg	1	*areg*	1	0	0	1	0	0	0
5		3	*opcode*						

Description

The content of the architectural register (*areg*) is moved to the shadow register (*sreg*). This instruction has two exposed delay slots, the data written cannot be read back until the third cycle from the cycle when this instruction enters the EXE phase.

Condition Codes

None

Pseudo Code

```
sreg = areg
```

MVS

Move Shadow Register to Architectural Register

Syntax

MVS *sreg, areg*

Functional unit = S

Opcode

16 bit

15 13	12 8	7							0
areg	*sreg*	1	1	0	0	0	1	1	1
3	5	*opcode*							

Description

The content of the shadow register (*sreg*) is moved to the architectural register (*areg*).

Condition Codes

None

Pseudo Code

```
areg = sreg
```

NEG

Negation

Syntax

NEG *src2, dst*

Functional unit = L

Opcode

16 bit

15–13	12–10	9									0
dst	*src2*	0	1	0	1	0	0	0	0	0	0
3	3	*opcode*									

Description

src2 is subtracted from zero and the result is stored in *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = 0 - src2
```

NOP

No Operation

Syntax

NOP

Functional unit = None

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1
						opcode									

Description

No operation.

Condition Codes

None

Pseudo Code

None

NOT

Bitwise NOT

Syntax

NOT *src2, dst*

Functional unit = L

Opcode

16 bit

15 13	12 10	9									0
dst	*src2*	0	0	1	1	0	0	0	0	0	0
3	3	*opcode*									

Description

Performs a bitwise inversion of *src2* and stores the result in *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = ~src2
```

OR

Bitwise OR 16-Bit Unsigned Constant with Register

Syntax

OR *ucst16, src2, dst*

Functional unit = L

Opcode

32 bit

31 ... 16	15 ... 13	12 ... 10	9									0
ucst16	*dst*	*src2*	0	0	1	0	0	0	0	0	1	1
16	3	3	*opcode*									

Description

Bitwise OR of a zero-extended 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 OR {zero extend}ucst16
```

OR

Bitwise OR Two Registers

Syntax

OR *src1, src2, dst*

Functional unit = L

Opcode

16 bit

15 — 13	12 — 10	9 — 7	6						0
dst	*src2*	*src1*	0	1	1	1	0	1	1
3	3	3	*opcode*						

Description

Bitwise OR of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 OR src1
```

RET

Return from Subroutine

Syntax

RET

Functional unit = D

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	1	1	1	1	1	1	1	1	1

opcode

Description

Load the link register (LR) into the program counter (PC), increment the stack pointer (SP), then load the contents of the memory location pointed to by SP into LR. The content of LR and the content of the memory location pointed to by SP in this case is treated as an absolute halfword address.

This instruction has one delay slot. See Section 8.2.A.3.1 for restrictions on scheduling an instruction in this delay slot.

Condition Codes

None

Pseudo Code

```
PC = LR; SP += 4; LR = *(SP)
```

REV

Reverse a Bit Field Bounded by Two Register Values

Syntax

REV *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–13	12–10	9–7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	0	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

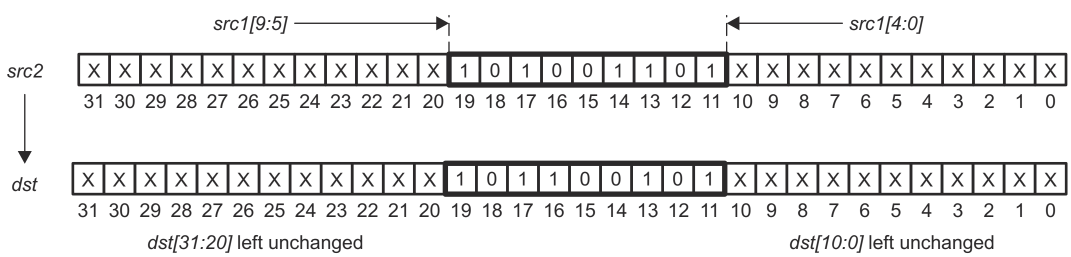
For *src1[9:5]* ≥ *src1[4:0]*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is bit reversed. The reversed bits are merged with the remaining (unchanged) bits in *dst* and are stored in *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field to be bit reversed. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be bit reversed. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5]* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 11 and *src1[9:5]* is 19.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
pntr = src1[9:5];
for( i = 0; i < 32; i++) {
  if ( i >= src1[4:0] && i <= src1[9:5]){
  dst[i] = dst[pntr]
  --pntr;
} else
  dst[i] = dst[i];
}
```

ROT

Rotate Right

Syntax

ROT *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–13	12–10	9–7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	0	1	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

src1[4:0] contains the number of bit positions that *src2* is to be rotated right. The new MSB value is defined by the previous LSB value of *src2*. The result is stored in *dst*. A *src1[4:0]* = 0, results in the **MV** *src1, dst* instruction.

In the following example, *src1[4:0]* is 9.

Rotate right

src1[4:0] = 9

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
src2	1	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1
dst	0	1	1	1	0	1	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 1; i < src1; i++) {
  dst = { dst[0], dst[32-2: 1] };
}
```

ROTC

Rotate Right Through Carry Bit

Syntax

ROTC *src1, src2, dst*

Functional unit = L

Opcode

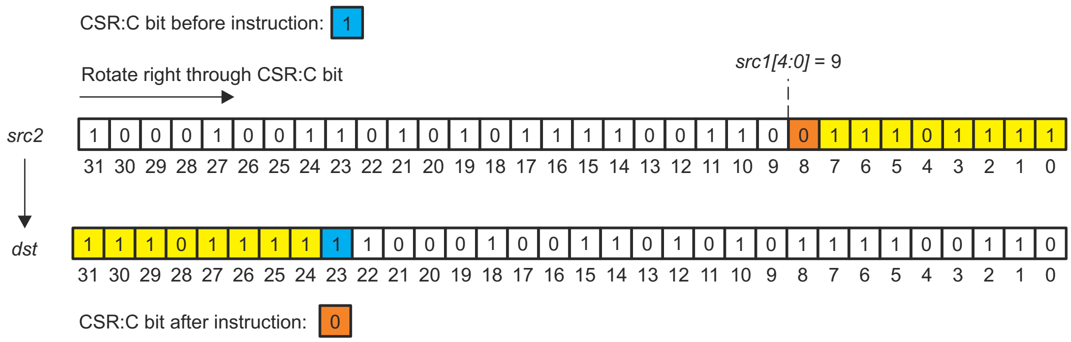
32 bit

31	30	29	28	27	26	25	24	23	22	21	20				16	15 13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	1	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																3	3	3	*opcode*						

Description

src1[4:0] contains the number of bit positions that *src2* is to be rotated right through the Carry bit (CSR[5]C). The new MSB value is defined by the current SAT bit in the control status register (CSR). The rotated LSB of *src2* is moved to CSR[5]C. The result is stored in *dst*. A *src1[4:0]* = 0, results in the **MV** *src1, dst* instruction.

In the following example, *src1[4:0]* is 9.



Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C: see the Pseudo Code

Pseudo Code

```
tmp = src2
for( i = 1; i < src1; i++) {
  tmp = { CSR[C], tmp[32-2: 1] };
  CSR[C] = tmp[0] ;
}
dst = tmp;
```

SADD

Signed Addition of Two Register Values with Saturation

Syntax

SADD *src1, src2, dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	1	0	0	0
3	3	3	*opcode*						

Description

Signed addition of *src1* with *src2* with result saturation and stored to *dst*. If a saturate occurs, the SAT bit in the control status register (CSR) is set after *dst* is written.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]SAT = satP OR satN

Pseudo Code

```
tmp = src2 + src1
satP = tmp > 2^31 - 1
satN = tmp < -2^31
dst = satP ? (2^31 - 1) : (satN ? -2^31 : tmp)
```

SATN

Saturation to Signed/Unsigned N-Bit Value with Sign/Zero Extend

Syntax

SATN *ucst3, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18 16	15 13	12 10	9									0
x	x	x	x	x	x	x	x	x	x	x	x	x	*ucst3[2:0]*	*dst*	*src2*	1	1	1	0	0	0	0	0	1	1
													3	3	3	*opcode*									

Description

Saturation to signed/unsigned N-bit value and sign/zero extend to 32 bits. *src2* is saturated based on *ucst3*:

- *ucst3[2]* = 1 indicates signed saturation
- *ucst3[2]* = 0 indicates unsigned saturation
- *ucst3[1:0]* = 0 indicates saturation to 8 bit value
- *ucst3[1:0]* = 1 indicates saturation to 16 bit value
- *ucst3[1:0]* = other values (2 or 3) are reserved

src2 is considered a signed value that needs to be saturated to a signed or unsigned range. The result is undefined, if reserved values in *ucst3* are used.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

dst = (src2 <= $-2^{(N-1)}$) ? $-2^{(N-1)}$: ((src2 >= $2^{(N-1)}$ - 1) ? $2^{(N-1)}$ - 1 : src2);

SET

Set Bit Field Bounded by Two Immediate Values

Syntax

SET *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25–21	20–16	15–13	12–10	9									0
x	x	x	x	x	x	*ucst5_1*	*ucst5_2*	*dst*	*src2*	1	1	0	0	0	0	0	0	0	0
						5	5	3	3	*opcode*									

Description

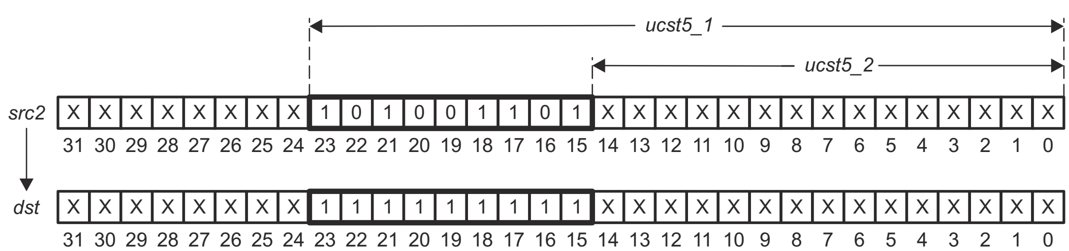
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is set to all 1s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be set to all 1s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 15 and *ucst5_1* is 23.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
  if ( i >= ucst5_2 && i <= ucst5_2) dst[i] = 1;
};
```

SET

Set Bit Field Bounded by Two Register Values

Syntax

SET *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	13	12 10	9 7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0		*dst*	*src2*	*src1*	0	0	0	0	1	0	0
																	3	3	3				*opcode*			

Description

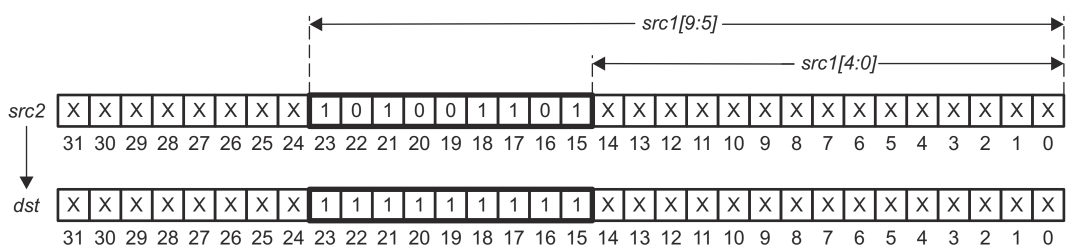
For *src1[9:5]* ≥ *src1[4:0*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is set to all 1s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be set to all 1s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 15 and *src1[9:5]* is 23.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
  if ( i >= src1[4:0] && i <= src1[9:5]) dst[i] = 1;
};
```

SHL

Logical Shift Left by 5-Bit Unsigned Constant

Syntax

SHL *ucst5, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20 16	15 13	12 10	9									0
x	x	x	x	x	x	x	x	x	x	x	*ucst5*	*dst*	*src2*	0	1	1	0	0	0	0	0	1	1
											5	3	3	*opcode*									

Description

src2 is left shifted by a 5-bit unsigned constant (*ucst5*) and the result is stored to *dst*. The shift in value is 0.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = src2 << ucst5
```

SHL

Logical Shift Left by Register Value

Syntax

SHL *src1, src2, dst*

Functional unit = L

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*src2*	*src1*	0	1	1	1	1	0	1
3	3	3	*opcode*						

Description

src2 is left shifted by an unsigned value (*src1*) and the result is stored to *dst*. The shift in value is 0. If the *src1* value is greater than 31, *dst* is cleared to 0.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = (src1 > 31) ? 0: (src2 << src1)
```

SHRA

Arithmetic Shift Right by 5-Bit Unsigned Constant

Syntax

SHRA *ucst5, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	*ucst5*	*dst*	*src2*	1	0	0	0	0	0	0	0	1	1
											5	3	3	*opcode*									

Description

src2 is right shifted by a 5-bit unsigned constant (*ucst5*) and the result is stored to *dst*. The shift in value is the sign bit (bit 31) of *src2*.

Using **DIV** (integer division) and **SHRA** (arithmetic right shift) does not produce the same result for negative numbers. The quotient of **DIV** is rounded towards zero, whereas the quotient of **SHRA** is rounded towards negative infinity. For example, using the **DIV** instruction: -9/4 = -2, whereas using the **SHRA** instruction: -9/4 = -3.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = {sign extend}{src2 >> ucst5}
```

SHRA

Arithmetic Shift Right by Register Value

Syntax

SHRA *src1, src2, dst*

Functional unit = L

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	1	1	1	0
3	3	3	*opcode*						

Description

src2 is right shifted by an unsigned value (*src1*) and the result is stored to *dst*. The shift in value is the sign bit (bit 31) of *src2* . If the *src1* value is greater than 31, *dst* is -1.

Using **DIV** (integer division) and **SHRA** (arithmetic right shift) does not produce the same result for negative numbers. The quotient of **DIV** is rounded towards zero, whereas the quotient of **SHRA** is rounded towards negative infinity. For example, using the **DIV** instruction: -9/4 = -2, whereas using the **SHRA** instruction: -9/4 = -3.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = (src1 > 31) ? FFFF FFFFh: {sign extend}{src2 >> src1}
```

SHRU

Logical Shift Right by 5-Bit Unsigned Constant

Syntax

SHRU *ucst5, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	*ucst5*	*dst*	*src2*	1	0	1	0	0	0	0	0	1	1
											5	3	3	*opcode*									

Description

src2 is right shifted by a 5-bit unsigned constant (*ucst5*) and the zero-extended result is stored to *dst*.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = {zero extend}{src2 >> ucst5}
```

SHRU

Logical Shift Right by Register Value

Syntax

SHRU *src1, src2, dst*

Functional unit = L

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	1	1	1	1
3	3	3	*opcode*						

Description

src2 is right shifted by an unsigned value (*src1*) and the result is stored to *dst*. The shift in value is 0. If the *src1* value is greater than 31, *dst* is cleared to 0.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = (src1 > 31) ? 0: {zero extend}{src2 >> src1}
```

SLA

Set Loop Address

Syntax

SLA *ucst16, creg*

Functional unit = S

Opcode

32 bit

31 ... 16	15 ... 11	10	9									0
ucst16	*creg*	x	1	0	1	0	0	0	1	0	1	0
16	5	*opcode*										

Description

Sets up loop bound address using a program counter (PC)-relative immediate positive offset. A zero-extended 16-bit unsigned immediate value (*ucst16*) is added to the PC of the instruction and is written to *creg*. Note that the offset is treated as a halfword offset. The only allowed *creg* registers for this instruction are: LSA0, LEA0, LSA1, and LEA1.

Condition Codes

None

Pseudo Code

```
creg = PC + {zero extend}ucst16
```

SSUB

Subtraction of Two Register Values with Saturation

Syntax

SSUB *src1, src2, dst*

Functional unit = D

Opcode

16 bit

15 – 13	12 – 10	9 – 7	6						0
dst	*src2*	*src1*	0	1	1	1	0	0	1
3	3	3	*opcode*						

Description

src1 is subtracted from *src2* with result saturation and stored to *dst*. If a saturate occurs, the SAT bit in the control status register (CSR) is set after *dst* is written.

Condition Codes

CSR[2] EQ = (dst == 0)

CSR[6] SAT = satP OR satN

Pseudo Code

```
tmp = src2 - src1
satP = tmp > 2^31 - 1
satN = tmp < -2^31
dst = satP ? (2^31 - 1) : (satN ? -2^31 : tmp)
```

STB

Store Byte to Memory with a 3-Bit Unsigned Constant Offset

Syntax

STB *dst, *+baseR[ucst3]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	1	0	1	1	1	1
3	3	3	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 0 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst3]) = *dst
```

STB

Store Byte to Memory with a 16-Bit Unsigned Constant Offset

Syntax

STB *dst, *+baseR[ucst16]*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*baseR*	1	0	1	0	0	0	1	0	0	0
16	3	3	*opcode*									

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 0 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst16]) = *dst
```

STB

Store Byte to Memory with a Register Offset

Syntax

STB *dst, *+baseR[src1]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	1	0	1	1	0	0
3	3	3	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 0 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(baseR[src1]) = *dst
```

STB

Store Byte to Memory, Postincrement with a 3-Bit Unsigned Constant Offset

Syntax

STB *dst, *baseR++[ucst3]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	1	0	1
3	3	3	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

Condition Codes

None

Pseudo Code

```
*(baseR++[ucst3]) = *dst
```

STB

Store Byte to Memory, Postincrement with a Register Offset

Syntax

STB *dst, *baseR++[src1]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	1	1	0	0	1	0
3	3	3	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of bytes) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

Condition Codes

None

Pseudo Code

```
*(baseR++[src1]) = *dst
```

STB

Store Byte to Memory with a SP-Relative 6-Bit Unsigned Constant Offset

Syntax

STB *dst,* *+**SP** *[ucst6]*

Functional unit = D

Opcode

16 bit

15 13	12 7	6						0
dst	*ucst6*	1	1	1	1	0	0	0
3	6	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 0 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(SP[ucst6]) = *dst
```

STB

Store Byte to Memory with a SP-Relative 19-Bit Unsigned Constant Offset

Syntax

STB *dst,* *+**SP** *[ucst19]*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst19	*dst*	*ucst19*	1	0	1	0	0	0	0	1	1	1
16	3	3	*opcode*									

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(SP[ucst19]) = *dst
```

STB

Store Byte to Memory with a GDP-Relative 19-Bit Unsigned Constant Offset

Syntax

STB *dst,* *+**GDP** *[ucst19]*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst19	*dst*	*ucst19*	1	0	1	0	0	0	0	1	1	0
16	3	3	*opcode*									

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the global data pointer register (GDP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(GDP[ucst19]) = *dst
```

STH

Store Halfword to Memory with a 3-Bit Unsigned Constant Offset

Syntax

STH *dst, *+baseR[ucst3]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	0	0	0
3	3	3	*opcode*						

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 1 bit. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst3]) = *dst
```

STH

Store Halfword to Memory with a 16-Bit Unsigned Constant Offset

Syntax

STH *dst, *+baseR[ucst16]*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*baseR*	1	1	0	0	0	0	1	0	0	0
16	3	3	*opcode*									

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 1 bit. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst16]) = *dst
```

STH

Store Halfword to Memory with a Register Offset

Syntax

STH *dst, *+baseR[src1]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	1	0	1	1	0	1
3	3	3	*opcode*						

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 1 bit. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[src1]) = *dst
```

STH

Store Halfword to Memory, Postincrement with a 3-Bit Unsigned Constant Offset

Syntax

STH *dst, *baseR++[ucst3]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	1	1	0
3	3	3	*opcode*						

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[ucst3]) = *dst
```

STH

Store Halfword to Memory, Postincrement with a Register Offset

Syntax

STH *dst, *baseR++[src1]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	1	1	0	0	1	1
3	3	3	*opcode*						

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[src1]) = *dst
```

STH

Store Halfword to Memory with a SP-Relative 6-Bit Unsigned Constant Offset

Syntax

STH *dst,* *+**SP** *[ucst6]*

Functional unit = D

Opcode

16 bit

15–13	12–7	6						0
dst	*ucst6*	1	1	1	1	0	0	1
3	6	*opcode*						

Description

The 16 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 1 bit. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst6]) = *dst
```

STH

Store Halfword to Memory with a SP-Relative 19-Bit Unsigned Constant Offset

Syntax

STH *dst,* *+**SP** *[ucst19]*

Functional unit = D

Opcode

32 bit

31-16	15-13	12-10	9	8	7	6	5	4	3	2	1	0
ucst19	*dst*	*ucst19*	1	1	0	0	0	0	0	1	1	1
16	3	3	*opcode*									

Description

The 16 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst19]) = *dst
```

STH

Store Halfword to Memory with a GDP-Relative 19-Bit Unsigned Constant Offset

Syntax

STH *dst,* *+**GDP** *[ucst19]*

Functional unit = D

Opcode

32 bit

31	16 15	13 12	10 9									0
ucst19	*dst*	*ucst19*	1	1	0	0	0	0	0	1	1	0
16	3	3	*opcode*									

Description

The 16 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the global data pointer register (GDP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(GDP[ucst19]) = *dst
```

STW

Store Word to Memory with a 3-Bit Unsigned Constant Offset

Syntax

STW *dst, *+baseR[ucst3]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	0	0	1
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 2 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst3]) = *dst
```

STW

Store Word to Memory with a 16-Bit Unsigned Constant Offset

Syntax

STW *dst, *+baseR[ucst16]*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9									0
ucst16	*dst*	*baseR*	1	1	1	0	0	0	1	0	0	0
16	3	3	*opcode*									

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 2 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst16]) = *dst
```

STW

Store Word to Memory with a Register Offset

Syntax

STW *dst, *+baseR[src1]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	1	0	1	1	1	0
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 2 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[src1]) = *dst
```

STW

Store Word to Memory, Postincrement with a 3-Bit Unsigned Constant Offset

Syntax

STW *dst, *baseR++[ucst3]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	1	1	1
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[ucst3]) = *dst
```

STW

Store Word to Memory, Postincrement with a Register Offset

Syntax

STW *dst, *baseR++[src1]*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*baseR*	*src1*	1	1	1	0	1	0	0
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[src1]) = *dst
```

STW

Store Word to Memory with a SP-Relative 6-Bit Unsigned Constant Offset

Syntax

STW *dst,* *+**SP** *[ucst6]*

Functional unit = D

Opcode

16 bit

15–13	12–7	6						0
dst	*ucst6*	1	1	1	1	0	1	0
3	6	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 2 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst6]) = *dst
```

STW

Store Word to Memory with a SP-Relative 19-Bit Unsigned Constant Offset

Syntax

STW *dst,* *+**SP** *[ucst19]*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst19	*dst*	*ucst19*	1	1	1	0	0	0	0	1	1	1
16	3	3	*opcode*									

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst19]) = *dst
```

STW

Store Word to Memory with a GDP-Relative 19-Bit Unsigned Constant Offset

Syntax

STW *dst,* *+**GDP** *[ucst19]*

Functional unit = D

Opcode

32 bit

31		16	15		13	12		10	9									0
ucst19			*dst*			*ucst19*			1	1	1	0	0	0	0	1	1	0
16			3			3			*opcode*									

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from the global data pointer register (GDP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(GDP[ucst19]) = *dst
```

STHI

Store 16-Bit Halfword to Memory with a 6-Bit Unsigned Constant Offset

Syntax

STHI *ucst16, *+baseR[ucst6]*

Functional unit = S

Opcode

32 bit

31–16	15–13	12–10	9–7	6	5	4	3	2	1	0
ucst19	*ucst6*	*baseR*	*ucst6*	0	0	0	1	0	1	1
16	3	3	3						*opcode*	

Description

The 16-bit unsigned constant (*ucst16*) is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 1 bit. After scaling, *ucst6* is added to *baseR*. The result of the calculation is the effective address in memory that contains *ucst16*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst6]) = ucst16
```

STRF

Store Register File to Stack Pointer

Syntax

STRF *op1, op2*

Functional unit = D

Opcode

16 bit

15–13	12–10	9	8	7	6						0
op2	*op1*	x	x	x	1	0	1	0	0	1	0
3	3				*opcode*						

Description

The current base register file is stored to the starting address specified by the postdecremented value of the stack pointer (SP). *op1* specifies the ending register (last register stored) and *op2* specifies the beginning register (first register stored). All registers between *op1* and *op2* are stored inclusive. The registers are stored in order from lowest register (*op2*) to highest register (*op1*).

Valid values of *op1* and *op2* are: *op1* ≥ *op2*

For *op1* < *op2*, results in unspecified behavior.

Condition Codes

None

Pseudo Code

```
for ( i = op2; i <= op1 ; i++) {
  *(SP) = Ri;
  SP -= 4;
}
```

SUB

Subtract 16-Bit Unsigned Constant from Register

Syntax

SUB *ucst16, src2, dst*

Functional unit = D

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*src2*	0	0	1	0	0	0	0	0	0	0
16	3	3	*opcode*									

Description

Subtract a 16-bit unsigned constant (*ucst16*) from *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (src2 - ucst16)

CSR[7]V = {overflow} from (src2 - ucst16)

Note that the carry (CSR[5]C) and overflow (CSR[7]V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 - ucst16
```

SUB

Subtract 16-Bit Unsigned Constant from Stack Pointer, Result to Stack Pointer

Syntax

SUB *ucst16,* **SP**

Functional unit = D

Opcode

32 bit

31–16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ucst16	0	0	1	1	1	x	x	x	1	0	0	0	0	0	0	1
16									*opcode*							

Description

Subtract a 16-bit unsigned constant (*ucst16*) from the stack pointer (SP) and store result to SP.

Condition Codes

None

Pseudo Code

```
SP = SP - ucst16
```

SUB

Subtract 16-Bit Unsigned Constant from Stack Pointer, Result to Register

Syntax

SUB *ucst16,* **SP** *, dst*

Functional unit = D

Opcode

32 bit

31–16	15–13	12	11	10	9	8	7							0
ucst16	*dst*	0	0	1	1	1	1	0	0	0	0	0	1	0
16	3	*opcode*												

Description

Subtract a 16-bit unsigned constant (*ucst16*) from the stack pointer (SP) and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (SP - ucst16)

CSR[7]V = {overflow} from (SP - ucst16)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = SP - ucst16
```

SUB

Subtract 3-Bit Unsigned Constant from Register

Syntax

SUB *ucst3, src2, dst*

Functional unit = D

Opcode

16 bit

15 – 13	12 – 10	9 – 7	6						0
dst	*src2*	*ucst3*	0	1	1	0	0	0	1
3	3	3	*opcode*						

Description

Subtract a 3-bit unsigned constant (*ucst3*) from *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (src2 - ucst3)

CSR[7]V = {overflow} from (src2 - ucst3)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 - ucst3
```

SUB

Signed Subtraction of Two Register Values

Syntax

SUB *src1, src2, dst*

Functional unit = D

Opcode

16 bit

15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	0	0	1	1
3	3	3	*opcode*						

Description

Subtract *src1* from *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (src2 - src1)

CSR[7]V = {overflow} from (src2 - src1)

Note that the carry (CSR[5]C) and overflow (CSR[7]V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

The status of the CSR[5]C bit can be used to synthesize multiword (wider) subtraction. With the **SUB** instruction, the CSR[5]C bit is set if no borrow occurs and the CSR[5]C bit is cleared if a borrow occurs. In other words, for **SUB** instructions, the CSR[5]C bit represents a not borrow. To synthesize multiword subtractions, subsequent instructions can use the CSR[5]C bit as a NOT(borrow) operand, performing a normal subtraction if CSR[5]C == 1 and subtracting one more than usual if CSR[5]C == 0.

For example, if register pairs R0/R1 and R2/R3 hold 64-bit values (where R0 and R2 hold the least-significant words), the following instructions leave the 64-bit difference in the register pair R4/R5:

```
     SUB   R2, R0, R4       ; R4=R0-R2, sets CSR[C]
     MVC   CSR, R6          ; leaves CSR[C] untouched
     EXTU  5, 5, R6, R6     ; if (CSR[C]) R6=0x1 else R6=0x0
     BNEQ  L1
     SUB   R3, R1, R5       ; R5=R1-R3
     SUB   1, R5, R5        ; R5=R5-1 if CSR[C] was NOT set
L1:  NOP                    ; else R5=R5
```

Note that the above example needs appropriate modification in order to generate correct carry/overflow applicable for the total 64-bit result.

Pseudo Code

```
dst = src2 - src1
```

SWI

Software Interrupt

Syntax

SWI

Functional unit = S

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	0	0	1	1	1	1	1	1	1

opcode

Description

Subtract *src1* from *src2* and store result to *dst*.

Condition Codes

SCSR = CSR; CSR[0] = 0

Pseudo Code

```
PC = *(IMEM[IST[SWI]]); IRP = PC + 1;
```

XOR

Bitwise Exclusive-OR Unsigned 16-Bit Constant with Register

Syntax

XOR *ucst16, src2, dst*

Functional unit = L

Opcode

32 bit

31–16	15–13	12–10	9	8	7	6	5	4	3	2	1	0
ucst16	*dst*	*src2*	0	1	0	0	0	0	0	0	1	1
16	3	3	*opcode*									

Description

Bitwise exclusive-OR (XOR) of a zero-extended 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 XOR {zero extend}ucst16
```

XOR

Bitwise Exclusive-OR Two Registers

Syntax

XOR *src1, src2, dst*

Functional unit = L

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*src2*	*src1*	0	1	1	1	1	0	0
3	3	3	*opcode*						

Description

Bitwise exclusive-OR (XOR) of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 XOR src1
```

8.2.B Clock, Reset, and Dynamic Power Management

8.2.B.1 Introduction

The ARP32 CPU has a single clock domain (cpu_fclk) and two reset domains:

- An asynchronous Power-On-Reset (cpu_porz_i)
- An asynchronous CPU Functional Reset (cpu_resetz_i)

Both of the reset signals are used asynchronously inside the design. Both the leading and trailing edges of these reset signals are synchronized with the CPU input clock by the external clock/reset management logic.

To reduce dynamic power consumption, the ARP32 CPU gates all of its internal clocks based on debugger connection status and CPU idle/standby status:

- If the ARP32 CPU is in the IDLE state and the CPU standby indication is asserted, all internal clocks of the ARP32 CPU remain gated.
- If debug connection is established, indicated via the input port cpu_dbgenable_i, the ARP32 CPU unconditionally enables all its internal clocks.

8.2.B.2 CPU Reset Modes

Table A-349 summarizes the two reset signals of the ARP32 CPU and their effects. Table A-350 summarizes the different application of the two resets.

Figure A-56 illustrates the ARP32 CPU power-on-reset sequence.

If warm reset is applied without Power on reset, to avoid meta-stability issues within the CPU core, the functional clock to the CPU core must be stopped by external clock generation logic prior to Warm reset assertion and the functional clock must be restored prior to Warm reset de-assertion.

Table A-349. CPU Reset Types

			Does CPU Reset?	
Reset Type	Signal Name	Reset Type	Debug Logic	Functional Logic
Power On Reset	cpu_porz_i	Async	Yes	No
Warm Reset	cpu_resetz_i	Async	No	Yes

Table A-350. CPU Reset Modes

				Does CPU Reset?		Need Synchronization?	
cpu_ porz_i	cpu_resetz_i	Reset Mode	Application	Debug Logic	Functional Logic	Assertion	De-assertion
0	0	Power On Reset	Reset at power up, full system reset	Yes	No	NA	Yes
0	1	Not supported	NA	NA	NA	NA	NA
1	0	Warm Reset	Reset of CPU core only, for example, watchdog reset	No	Yes	Yes	Yes
1	1	Normal	No reset, normal running mode	No	No	NA	NA

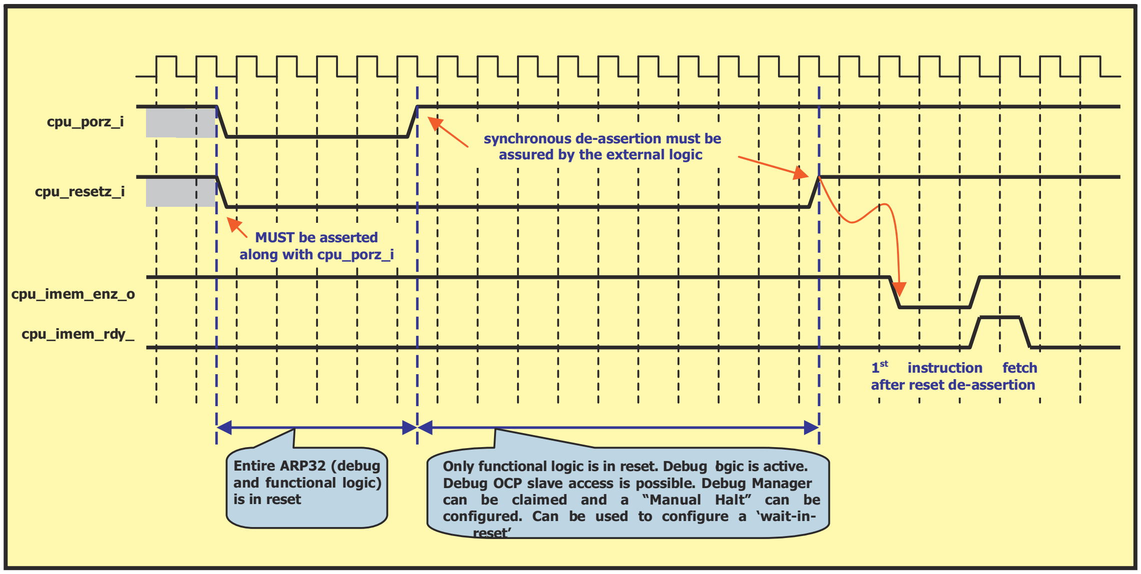


Figure A-56. Power-On-Reset

8.2.B.3 Dynamic Power Management

The ARP32 CPU supports instruction driven transition to a low-power state that is used to trigger a CPU system-wide low-power state. This is accomplished by the **IDLE** instruction. Upon execution of the **IDLE** instruction, the ARP32 CPU waits for any pending instruction or data memory transaction to complete and then goes to an endless wait state.

When in idle state, the CPU output signal cpu_standby_o is asserted high. During this window, the CPU gates clocks to all of its registers (except the interrupt flag register (IFR)) to achieve maximum saving of dynamic power. It is assured that the ARP32 CPU does not issue any new transactions on either instruction or data memory interfaces during this window. So, peripherals external to the CPU that don't need to be active while the CPU is in IDLE are also clock gated during this window.

The CPU comes out of this wait state only via an enabled external interrupt (NMI, INT4-INT7) or reset. The cpu_standby_o signal is de-asserted as soon as one of the following events occurs:

- any of the interrupt flag register (IFR) bits are set
- a debug connection is established (cpu_dbgenable_i is asserted high)

Interrupt enable conditions (like GIE, IER, etc.) are not considered when de-asserting cpu_standby_o. Thus, a disabled interrupt being asserted at the CPU boundary causes the cpu_standby_o signal to be de-asserted but the CPU still remains in idle state until an enabled interrupt (or reset) occurs.

Figure A-57 provides an illustrative waveform showing the CPU going into IDLE mode followed by a wakeup via INT4.

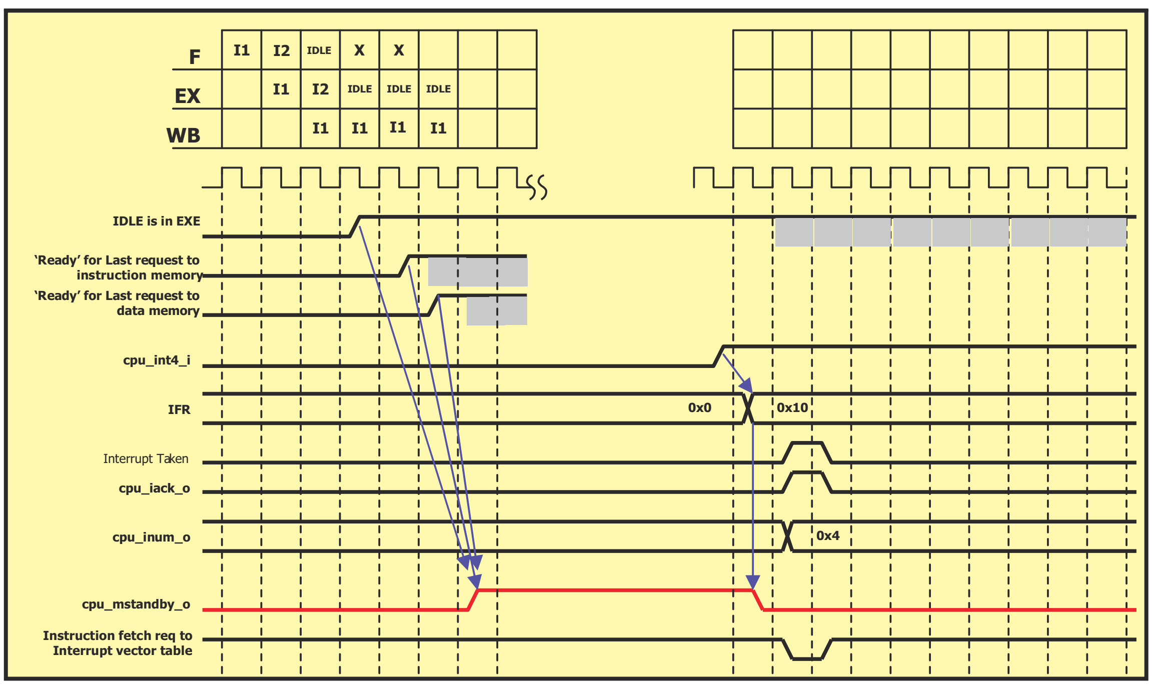


Figure A-57. CPU Standby and Wakeup Procedure

8.2.C Notes on Programming Model

8.2.C.1 Booting

The entry for reset in the ARP32 interrupt service table (IST) is setup such that the program flow after reset calls an initialization or boot code as soon as possible and performs at least the following functions:

- Initialize the stack pointer (SP) register.
- Initialize the global data pointer (GDP) register.
- Call ‘main’ function.

The following tasks are additionally performed if interrupt processing is needed from the very beginning; else, they are done as soon as the system is ready for (or expecting) interrupt processing:

- Initialize the interrupt enable register (IER) to enable the required maskable interrupts, as required.
- Initialize the interrupt enable register (IER) to enable the nonmaskable interrupt, if required.
- Initialize the control status register (CSR) to enable all the maskable interrupts globally.

8.2.C.2 Enabling and Disabling Interrupts

The ARP32 CPU provides a mechanism for easy and flexible interrupt control.

8.2.C.2.1 Globally Enabling or Disabling Maskable Interrupts

The global interrupt enable (GIE) bit in the control status register (CSR) allows programmers to enable or disable all maskable interrupts by controlling the value of a single bit. On the ARP32 CPU, programs must directly manipulate the GIE bit in CSR to disable and enable interrupts:

- CSR[0]GIE = 1 enables the maskable interrupts so that they are processed
- CSR[0]GIE = 0 disables the maskable interrupts so that they are not processed

For example, the following code sequence globally enables all maskable interrupts:

```
MVC   CSR, R0         ; Get CSR into R0
SET   0, 0, R0, R0    ; Set R0[0]
MVC   R0, CSR         ; Copy R0 back to CSR (sets GIE)
```

Similarly, the following code sequence globally disables all maskable interrupts:

```
MVC   CSR, R0         ; Get CSR into R0
CLR   0, 0, R0, R0    ; Clear R0[0]
MVC   R0, CSR         ; Copy R0 back to CSR (clears GIE)
```

As interrupt detection occurs in parallel with CPU execution, the CPU takes an interrupt in the cycle immediately following an **MVC** instruction that clears the GIE bit. However, CPU context save/restore behavior ensures that interrupts do not occur after subsequent instruction. Consider the following code example where the CPU takes an interrupt between instructions 1 and 2, between instructions 2 and 3, or between instructions 3 and 4. The CPU does not service an interrupt between instructions 4 and 5.

```
; assume GIE=1
MVC   CSR, R0         ;(1) Get CSR
AND   -2, R0, R0      ;(2) Get ready to clear GIE
MVC   R0, CSR         ;(3) Clear GIE
ADD   R0, R2, R3      ;(4)
ADD   R4, R4, R5      ;(5)
```

If the CPU services an interrupt between instructions 1 and 2 or between instructions 2 and 3, the SCSR[0]GIE bit holds the value 1 when arriving at the interrupt service routine. If the CPU services an interrupt between instructions 3 and 4, the SCSR[0]GIE bit holds the value 0. Thus, when the interrupt service routine resumes the interrupted code, it resumes with the GIE bit cleared (as a result of context restore of SCSR to CSR) as the interrupted code intended.

8.2.C.2.2 Enabling or Disabling Individual Interrupts

Software can enable and disable individual interrupts by setting and clearing the bits in the interrupt enable register (IER) that correspond to the individual interrupts. An interrupt can trigger interrupt processing only if the corresponding bit in the IER is set.

For example, the following code sequence enables INT4:

```
MVC    IER, R0         ; Get IER into R0
SET    4, 4, R0, R0    ; Set R0[4]
MVC    R0, IER         ; Copy R0 back to IER (sets IER[4])
```

Similarly, the following code sequence disables INT5:

```
MVC    IER, R0         ; Get IER into R0
CLR    5, 5, R0, R0    ; Clear R0[5]
MVC    R0, IER         ; Copy R0 back to IER (clears IER[5])
```

8.2.C.3 Stack Usage in Interrupt Service Routine

The ARP32 CPU has only one stack pointer and thus a unified stack is used for background and interrupt code. As a result, the programmer is responsible for maintaining a clean boundary of operations that modifies the stack pointer (for example, stack allocation and de-allocation for a function, context save onto stack via push/pop operations using the **LDRF/STRF** instructions).

Instructions/operations in the CPU that modify the stack pointer are designed so that an interrupt cannot disrupt the operations halfway through to completion. For example, a stack allocation and de-allocation for function can happen via the **ADD** *ucst16,* **SP** instruction that atomically modifies the stack. The return address save while taking a CALL or retrieving the return address while executing a RET is also atomic, and cannot be interrupted. The **LDRF/STRF** instructions used for single or multiple pop and push are also uninterruptable. The ARP32 C/++ compiler uses these instructions to maintain the C/C++ stack and thus the sanity of the stack is always maintained.

Under certain scenarios, to save additional user context onto the stack, it is done via the **LDRF/STRF** instruction. **LDRF/STRF** modifies the stack pointer and hence results in the stack layout of a function to be not valid anymore. Hence, such additional context save/restore to the stack must happen at function boundaries only.

These considerations are important for assembly level programmers doing additional context save/restore (beyond what the ARP32 compiler does) and also to new custom instruction design for the ARP32 CPU if they depend on stack pointer relative operations.

8.2.C.4 General Restrictions

The ARP32 CPU has the following restrictions on the programming model:

- **Long Long (as 64 bit) is not supported natively in the architecture.** 'long long' is supported via emulation in the Run Time Support (RTS) library. Since this is non-standard and sub-optimal, use of long may severely degrade performance. Hence, it is highly recommended not to use the C 'long long' datatype.
- **64-bit arithmetic.** Add/sub of more than 32-bit values (operand or results) requires **ADDC/SUBC/BC** (addition with carry, subtraction with borrow, branch on carry) instructions that are not supported in the ARP32 CPU. Again, it is possible to emulate this in RTS but it is a sub-optimal implementation and it is not recommended.
- **Multi-threading via mutex is not supported in the architecture.** In the ARP32 CPU, it does not implement an "EXCLUSIVE access instruction" to implement a 'proper' mutex in RTS. A polling based multi-threading is still possible.

8.3 VCOP CPU and Instruction Set

Note

Throughout this section, various sample assembly code pieces are used to illustrate VCOP architecture and functionality. Programmers are expected to write code in VCOP Kernel-C, instead of in assembly.

8.3.1 Module Overview

The Embedded Vision Engine (EVE) module is a programmable imaging and vision processing engine, intended to be used in devices that serves consumer electronics imaging and vision applications. Its programmability allows late-in-development or post-silicon processing requirements to be met, and allows third party or customers to add differentiating features in imaging and vision products.

The EVE Module consists of an ARP32 scalar core, a VCOP vector core, and a DMA controller.

8.3.2 Features

- Interfaces
  - CLK/2 MHz custom interface with the scalar core
- Functionality
  - CLK MHz functional clock
  - Based on VICP coprocessor
  - Dual-issue, N-way SIMD per instruction
  - Scalable in N ways of SIMD, N = {2 | 4 | 8 | 16 | 32}
  - For the first version of VCOP N = 8
  - 16 entries x N ways x 40-bit register file
  - Binary morphology acceleration
  - N-way table lookup and histogram acceleration (capable of N lookups per cycle or N histograms per 4 cycle)
  - Up to 20-bit vector data memory address space
- Debug interface with visibility of loop variables and load/store pointers (VCOP_LD_PTR_i/VCOP_ST_PTR_j).

8.3.3 Block Diagram

The EVE subsystem contains an ARP32 scalar core, VCOP vector core, EDMA, memory blocks, memory switch, local interconnect, reset/clock, interrupt controller, SCTM and SMSET instrumentation blocks. Figure 8-58 provides a block diagram.

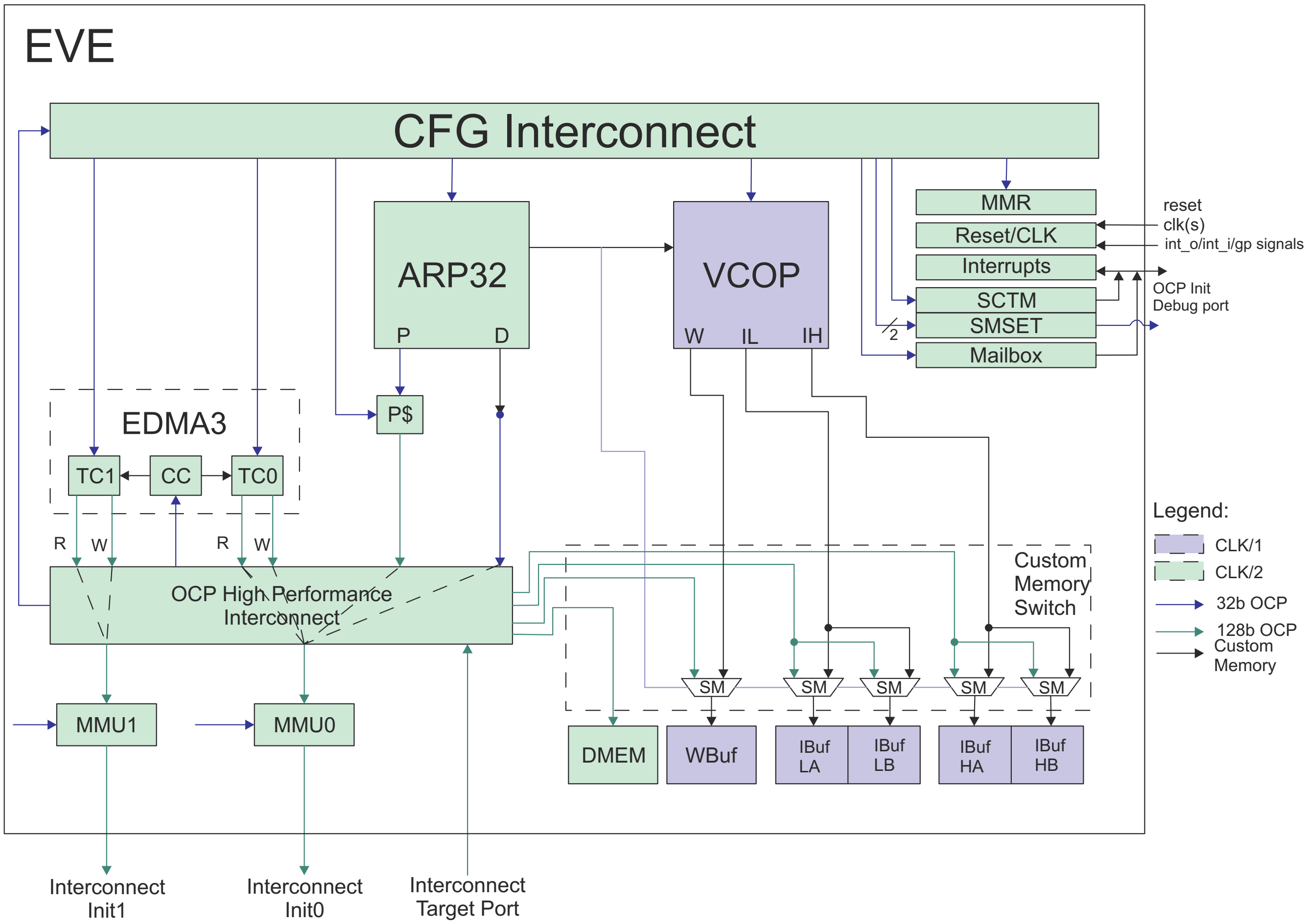


Figure 8-58. EVE Block Diagram

Figure 8-59 depicts the functional block diagram of EVE's Vector Coprocessor (VCOP).

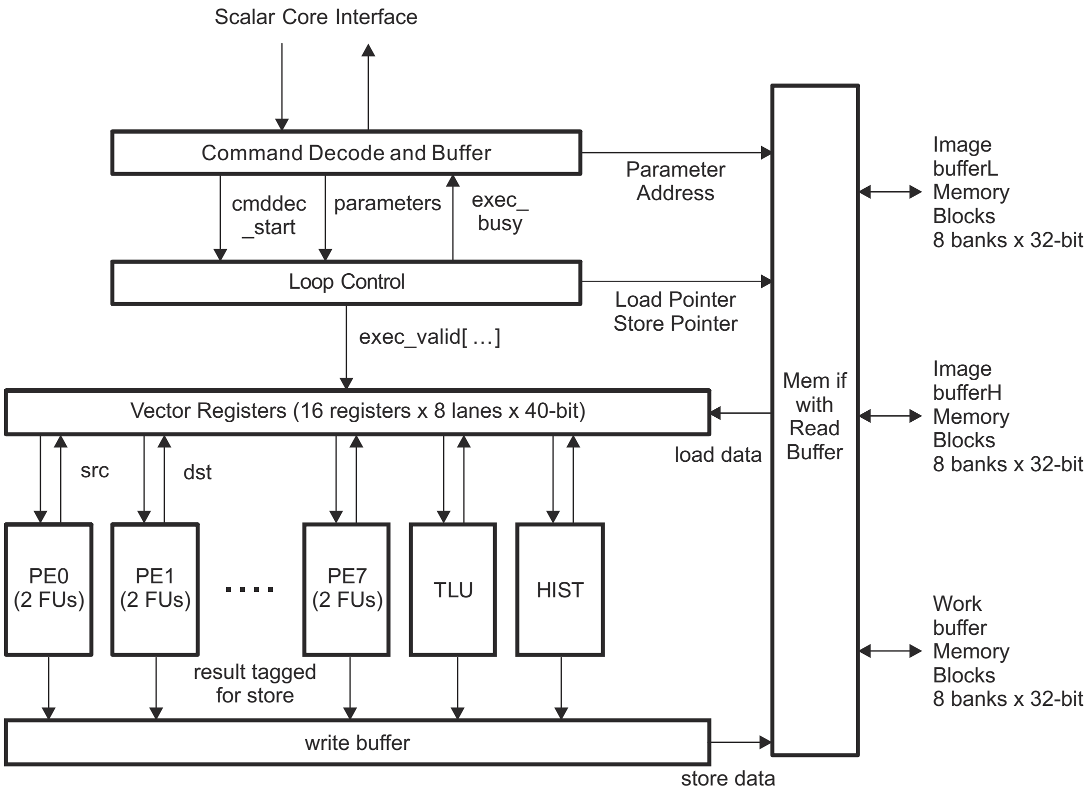


Figure 8-59. VCOP Block Diagram

8.3.4 System Interfaces

8.3.4.1 Interrupts

The vec_err_intn interrupt is driven by the vector core and indicates that the vector core has encountered an error condition, see Section 8.3.7.2.1 VCOP Register Manual for details.

8.3.4.2 Configuration Bus Slave Port

VCOP has a configuration bus slave interface that follows the OCP 32-bit slave protocol. The interface enables module configuration, status, processor state readout during debug.

The interface operates in the same clock domain as clk_eve_sca, the scalar core and scalar/vector interface clock.

The memory map of the configuration bus is specified in Section 8.3.7.2.1.

8.3.4.3 Performance Counter Interface

On the EVE level there is an SCTM module and an SMSET module to collect performance statistics.

SCTM is for counting signal active time and rise and fall transitions. SMSET is for conveying events to include into the system trace. Table 8-351 lists the signals that are exported to SCTM or SMSET as indicated. All SCTM signals are level signals operating at full vector core speed (clk_eve_vec), and all SMSET signals are rising-edge triggered signals.

Table 8-351. Performance Counter Signal List

Signal	Destination	Description
vcop_busy	SCTM	Indicate VCOP busy, for measuring VCOP loading
vcop_idle_and_done	SCTM	Indicate VCOP idle and done, basically not(vcop_busy) and vec_done, for measuring slack in real-time schedule

Table 8-351. Performance Counter Signal List (continued)

Signal	Destination	Description
vcop_wait_for_arp32	SCTM	Indicate VCOP idle and waiting for ARP32 to feed instruction, basically vec_rdy and not(vec_instr) and not(vcop_busy). This indicates VCOP under utilization that might be improved by reducing ARP32 interrupt services
vcop_arp32_awaits	SCTM	Indicate ARP32 has vector instruction ready and is waiting for VCOP to accept it, basically not(vec_rdy) and vec_valid. This indicates ARP32 idle time that might be utilized, however, not to the extent that vcop_wait_for_ARP32 time becomes significant.
vcop_overhead	SCTM	Indicate VCOP busy but not executing the loop, meaning exposed command decode (from decode-execute pipeline) and parameter access time. The former might be reduced by having more back-to-back loops, and the latter is unavoidable.
vcop_ld_stall_by_st	SCTM	Indicate LD stage is stalled by ST stage (forced write when write buffer is full). This means there is insufficient gaps in LD for ST to occur in the background, but this does not mean LD+ST is slowing down computation.
vcop_op_stall_by_ldst	SCTM	Indicate gaps in operation stage between iterations due to LD+ST, indicating extent of load/store bound (as opposed to compute-bound). This may be reduced by merging loops to reduce load/store.
vcop_op_stall_by_ dependency	SCTM	Idle cycle inserted in operation stage due to dependency. This may be minimized by scheduling.
vcop_rd_ibufl	SCTM	Indicate VCOP reading from IBUFL
vcop_rd_ibufh	SCTM	Indicate VCOP reading from IBUFH
vcop_rd_wbuf	SCTM	Indicate VCOP reading from WBUF
vcop_wr_ibufl	SCTM	Indicate VCOP writing to IBUFL
vcop_wr_ibufh	SCTM	Indicate VCOP writing to IBUFH
vcop_wr_wbuf	SCTM	Indicate VCOP writing to WBUF
vcop_loop_start	SMSET	Indicate VCOP getting a VLOOP instruction
vcop_done	SMSET	Indicate VCOP executing a VWDONE instruction, copy of vec_done

8.3.4.4 Data Memory Map

The first instance of EVE has a data memory configuration and a memory-map as listed in Table 8-352.

Memory maps for scalar and vector core are shown in Figure 8-60. An optional MMR field, VEC_MEM_128KB, wraps the VCOP memory map around 128KB, aliasing the A and B sets to the same address range.

Table 8-352. EVE Vector Data Memory Map

Byte Address	Name	Description
0x00000 ~ 0x07FFF	WBUF	Vector working buffer, 32Kbytes
0x08000 ~ 0x0FFFF	reserved	Reserved for future expansion
0x10000 ~ 0x13FFF	IBUFLA	Image buffer low copy A, 16 Kbytes
0x14000 ~ 0x17FFF	IBUFHA	Image buffer high copy A, 16 Kbytes
0x30000 ~ 0x33FFF	IBUFLB	Image buffer low copy B, 16 Kbytes
0x34000 ~ 0x37FFF	IBUFHB	Image buffer high copy B, 16 Kbytes

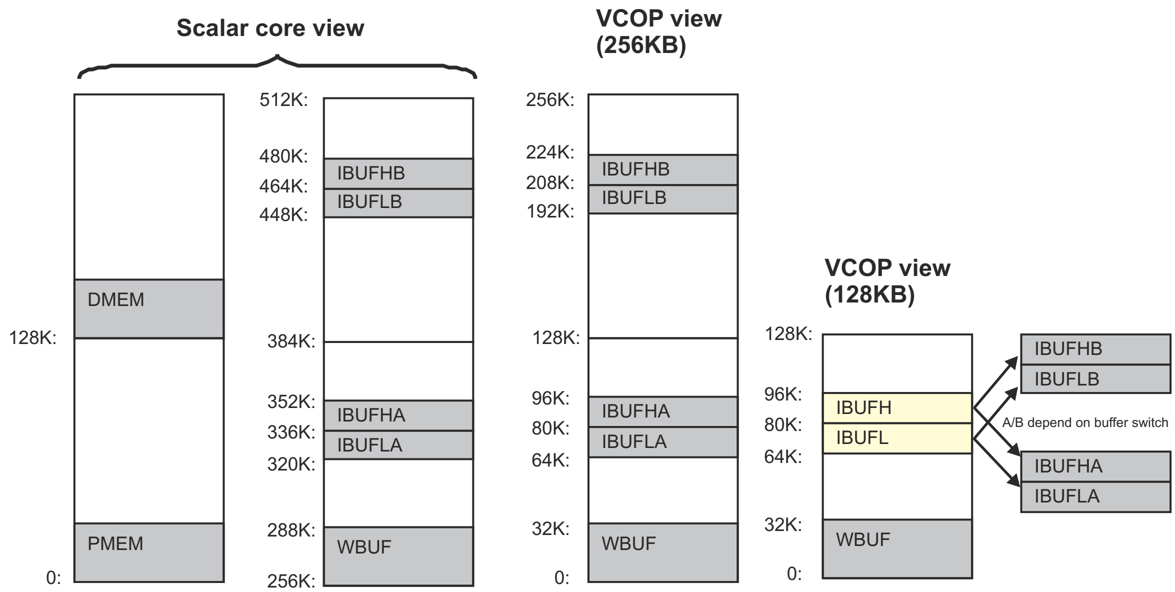


Figure 8-60. EVE Memory Map

8.3.5 Functional Description

8.3.5.1 Scalar-Vector Architecture

The EVE subsystem consists of a scalar core as the subsystem controller, a vector core as the high-throughput coprocessor, shared memories, DMA, clock/reset and interrupt controllers. See the *EVE Subsystem Reference Guide* for details.

8.3.5.1.1 Scalar Core

The scalar core executes conventional RISC instructions. The scalar core is based on the ARP32 processor.

See the *ARP32 CPU and Instruction Set Reference Guide* for more details.

8.3.5.1.2 Scalar-Vector Interaction

The EVE scalar and vector cores share one program memory, but can execute in parallel on separate threads. The scalar core has sole control of the program memory, so all vector instructions are accessed by the scalar core, recognized being a vector instruction, and relayed via the scalar/vector interface to the vector core for execution. The vector core executes a sequence of instructions (called a vector command) repeatedly, so typically can be busy executing and does not need further vector instructions for thousands of clock cycles at a time.

The vector core has temporary instruction storage. There is a pre-decode buffer of 64 words, plus a decoded buffer for two maximal-sized commands (see Figure 8-61). The vector core attempts to decode and fill the two commands first, and if vector instructions keep coming, fill the pre-decode buffer. Together the buffering holds 4 or more typical vector commands, and keep the vector core busy for a while, allowing the scalar core to switch context to serve interrupts when interrupts occur.

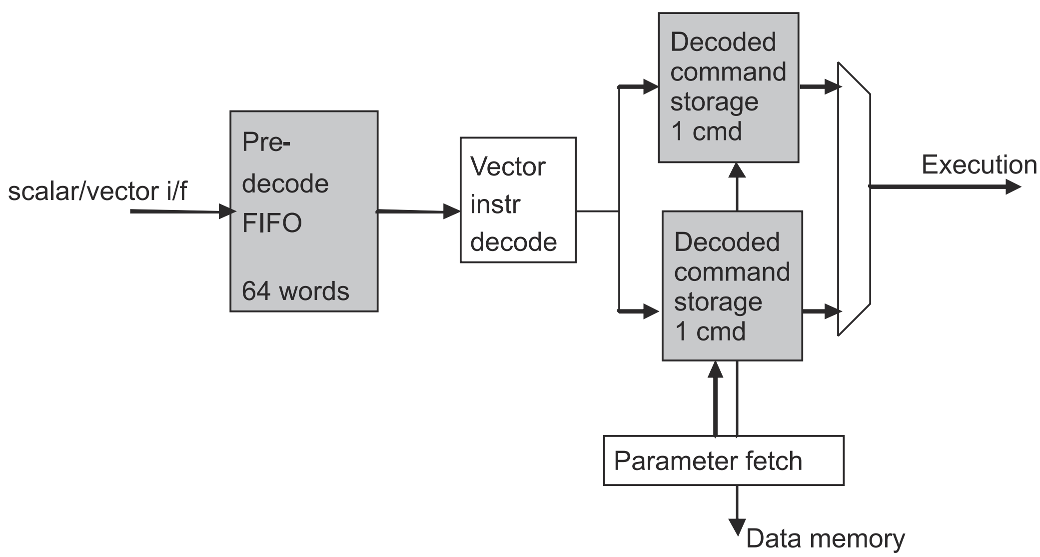


Figure 8-61. VCOP Instruction Buffering

8.3.5.2 Vector Core Overview

The vector core is a SIMD machine with built-in loop control and address generation. It is programmed in array or 2D block processing level. The vector core has the following resources:

- 4 nested for loops, with loop variables i1, i2, i3, and i4, plus an optional outer loop i0
- 8 address generators, each capable of 4-dimensional addressing, that is, the address pattern is: base + i1 × const1 + i2 × const2 + i3 × const3 + i4 × const4
- 16-entry vector register file, each entry is N-way SIMD × 40-bit signed (sign-extended or zero-padded from 8/16/32-bit signed/unsigned memory data, or from operation upon register data)
- Two general-purpose functional units, each N-way SIMD. Functional spec supports N = {2 | 4 | 8 | 16 | 32}, first instance N = 8.
- Table lookup unit supporting up to N parallel lookups.
- Histogram unit supporting up to N parallel histogram operations.
- 8 load units
- 8 store units

The vector core supports the following functions:

- Generic compute
- Table lookup
- Histogram and weighted histogram

8.3.5.2.1 Nested for Loop Model

EVE vector operation is controlled by a four-level nested for loop. Inside the loop, the behavior can be represented in sequential stages: load, arithmetic operation, store, and pointer update.

In each stage, a number of load, operation, stores, or pointer updates are carried out, with respect to the 16 × N × 40-bit vector register file. There is no dependency among loads, stores, or pointer updates; multiple loads unto the same register file is not allowed. Dependency among operations assumes 2 operations are carried out in parallel, with no delay slot for most operations, and 1 cycle of delay slot for some specific operations.

The vector register file, loop variables, address generators are replicated as needed in the hardware to support pipelining among the various stages.

Example 8-8 shows a skeleton of the nested loop model. There are four loop variables, i1, i2, i3, and i4. A snapshots of the loop variables are contained in the follwoing registers:

- VCOP_I0_I1[15:0] I0 - I0 loop variable
- VCOP_I0_I1[31:16] I1 - I1 loop variable
- VCOP_I2_I3[15:0] I2 - I2 loop variable
- VCOP_I2_I3[31:16] I3 - I3 loop variable
- VCOP_I4[15:0]I4 - I4 loop variable

Example 8-8. Nested Loop Model Skeleton

```
EVE_compute(...)
{
   for (i1=0; i1<=lpend1; i1++) {
      for (i2=0; i2<=lpend2; i2++) {
         for (i3=0; i3<=lpend3; i3++) {
            for (i4=0; i4<=lpend4; i4++) {
               for (k=0; k<num_inits; k++)
                  initialize_vreg_from_parameters(...);
               for (k=0; k<num_loads; k++)
                  load_vreg_from_local_memory(...);
               for (k=0; k<num_ops; k++)
                  op(...);         // 2 functional units, executing 2 ops per cycle
               for (k=0; k<num_stores; k++)
                  store_vreg_to_local_memory(...);
               for (k=0; k<num_agens; k++)
                  update_agen(...);
            }
         }
      }
   }
}
```

8.3.5.2.1.1

Each stage is described in more details in the following sections.

Note that loads and stores are predicated on the loop variables matching specific conditions and are thus not always carried out at every i4 iteration.

Each i4 iteration takes a number of cycles equal to the maximal number of cycles spent in loads, arithmetic operatoins, and stores. Cycle count in the arthmetic operations is constant for each loop, but cycle count in load and store can change depending on pointer update, loop level, and read/write memory contention.

For example, with lpend1 = 3, lpend2 = 1, lpend3 = 2, and lpend4 = 4, the nested for loop executes in exactly 4 × 2 × 3 × 5 = 120 i4 iterations. For the first few iterations the loop variables progress as:

i1	i2	i3	i4
0	0	0	0
0	0	0	1
0	0	0	2
0	0	0	3
0	0	0	4
0	0	1	0
0	0	1	1
0	0	1	2
...			

8.3.5.2.2 Instruction Organization

Vector instructions cannot appear in the program stream in arbitrary order. Instructions are classified as:

- Control
  - Repeat end count
  - Parameter pointer
  - Write buffer switch
- Synchronization
  - Wait for ready
  - Wait for done
- Computation
  - VLOOP
  - Register initialization
  - Address generator
  - Load
  - Operations
  - Store

The sequencing rule is that computation instructions that form a loop are in one contiguous block, starting with VCOP_VLOOP_PTR[31:0]VLOOP_PTR. This block of computation instructions is called a vector command. Control, synchronization, and scalar instructions can appear in any order, except that they cannot appear inside a vector command.

8.3.5.3 Vector Control

The VCTRL instruction is used to convery control information to the vector core. The instruction syntax is:

VCTRL	*scalar_register* **,** *control_register*

The control registers so far defined are 0: RPT_END and 1: PARAM_PTR (VCOP_PARAM.

8.3.5.3.1 Repeat End Count

A VCTRL instruction variation programs the repeat end count. The syntax is:

VCTRL	*scalar_register* **, RPT_END**

Subsequent VLOOP is executed RPT_END + 1 times. Reset default is 0, or repeating just once, and is the normal mode of operation. At end of loop execution, RPT_END is automatically reset to 0, so if the subsequent is non-repeating, which is most of the cases, there is no need to issue another VCTRL RPT_END to reset it.

RPT_END is a 12-bit value; the largest value supported is 0:4095, allowing for up to 4096 iterations of the repeat loop.

Note that the VLOOP instruction that follows can be of any type (computer, table lookup, histogram).

The repeat mechanism can be viewed as enclosing VLOOP by an outer loop, i0. The difference between this i0 loop and the i1, i2, i3, i4 loops inside VLOOP is that parameter pointer is advanced at the end of VLOOP. The repeat feature allows the same program code to be executed with a different set of parameters each time.

The repeat feature is useful for performing the same processing steps on a list of data regions that are not spaced regularly and/or not having the same dimension. Instead of having a scalar loop around the repeated VLOOP, using the repeat feature reduces scalar/vector interaction and resulting overhead.

Software must allocate memory appropriately to allow results of these data blocks to be stored (sequentially or otherwise is under programmer's control).

8.3.5.3.2 Parameter Pointer

In addition to the vector instructions in PMEM, the vector core must read a parameter file in WBUF or IBUFL/ IBUFH (depending on the pointer value) before it begins intended computation. The parameters include loop counts, address pointers to arrays, constants used in the computation, round/truncate shift count, saturation bounds, etc. The parameter pointer is located in VCOP_PARAM_PTR[31:0] PARAM_PTR.

Separation of configuration coded in the program memory versus configuration conveyed in data memory is essential, as this allows library functions to be constructed with reasonable code size and can serve a variety of need in applications. Loop counts and address pointers being conveyed via parameters is needed, so that, for example, one filter function works for various data/coefficient/output arrays and block and coefficient kernel sizes. Round/truncation shifts being convey via parameters means that, for example, the same filtering kernel program can work for filters with different radix point bit positions for filter coefficients.

Parameter registers are 16-bit per entry in the parameter file, up to 64 entries. The first 2 entries of the parameter file, P0 and P1, are hard-coded to constant values 0 and 1 respectively, as these are commonly used and it reduces code size by referring to these constants without taking up data memory space.

Parameters are referred to in the vector instructions either as single register, 16-bit parameter, or as a register pair, together make up 32-bit. Where a pair is used, the first parameter must be an even index, and contains the lower 16-bit of the 32-bit.

From an application development point of view, algorithm kernels are sometimes coded in one vector command per kernel, sometimes in several commands. Thus it is beneficial for the vector core to advance the parameter pointer automatically to reduce overhead. This requires a mechanism to specify or modify the parameter pointer when needed. This is addressed by the parameter pointer update feature through VCTRL.

A VCTRL instruction variation updates the parameter pointer for the subsequent VLOOP. The syntax is:

VCTRL *scalar_register* **, PARAM_PTR**

Unless updated again by VCTRL, the parameter pointer is advanced automatically at the end of VLOOP execution by the #PL field of VLOOP. Thus, parameter blocks for multiple consecutive VLOOPs can be placed consecutively in data memory, and no VCTRL VCOP_PARAM_PTR is needed except for the first VLOOP.

For efficiency of parameter access by VLOOP, the pointer value conveyed via VCTRL must be 32-bit aligned. The parameter length specified in VLOOP is also 32-bit aligned. Thus, at the beginning of any VLOOP, the pointer is 32-bit aligned.

Note that inside a VLOOP while parameters are pulled off to configure the loop, the parameter pointer is only 16-bit aligned.

For example, if 11 parameter registers are used in a vector command, P0..10, the encoded parameter length is ceiling((11 – 2)/2) = 5. The subsequent next vector command accesses parameter registers 5 32-bit words from the current vector command, leaving one 16-bit halfword in memory unused.

8.3.5.3.3 Switch Buffers

To minimize task-switching overhead, an instruction is added to write to buffer switch MMR. The instruction syntax is:

VSWITCHBUF *ucst20*

Where ucst20 is a 20-bit immediate value to be written to a dedicated MMR for buffer switches.

This instruction is carried out by the scalar core, and vector core just ignores it.

Change made to the buffer switch is visible in MMR as well, as both VSWITCHBUF and generic scalar core store to the buffer switch MMR are both valid ways to program the buffer switches, only that VSWITCHBUF achieves lower latency. Both mechanisms are kept consistent in the simulator/ debugger toolset.

8.3.5.4 Vector-Scalar Synchronization

8.3.5.4.1 Wait for Vector Core Done

The VWDONE instruction is used as a synchronization barrier between scalar and vector core, and affects the scalar/vector interface signal vec_done. Upon reset, vec_done is true. Upon execution of VWDONE in the scalar core, the scalar stalls and waits for vec_done. The vector core, upon exeucting this instruction, implying all previous submitted vector instructions are all done, asserts vec_done. Once asserted, vec_done stays true until receiving a subsequent VLOOP that turns the vector busy and vec_done false again.

The VWDONE instruction is typically used before the scalar core switches shared memory to itself or DMA. Before it changes the switch it makes sure that the vector core is idle and thus not accessing the memory.

VWDONE must appear between vector commands in the program, not inside a vector command.

8.3.5.4.2 Wait for Vector Core Ready

The VWRDY instruction is used to stall the scalar core until vector core is ready to accept additonal vector instructions. Vector core ignores this instruciton.

Normally the scalar core can simply issues vector instruction, and when the vector core is not ready, VCOP_STATUS[2]VEC_RDY = 0 the scalar core is automatically stalled. Having VWRDY allows inserting timer read before and after VWRDY to measure the time duration of vector core not being ready, for performance tuning purposes.

8.3.5.5 Vector Computation

8.3.5.5.1 Vector Loop

The VLOOP instruction marks the beginning of a vector command specifying a loop. Assembly format:

VLOOP *cmd_type* **,** *CL#* **:** *cmd_len* **,** *PL#* **:** *param_len*

The *cmd_type* field specifies whether it is the main compute (COMP), table lookup (TLU), or histogram (HIST).

Command length (*cmd_len*) and parameter length (*param_len*) in 32-bit words are encoded to facilitate quick parsing of the vector commands. The ending loop counts, lpend1/2/3/4, are encoded at fixed entries in the parameter file, P2 = lpend1, P3 = lpend2, P4 = lpend3, P5 = lpend4. This allows easy adaptation of EVE function to different array dimensions.

The first two entries of the parameter file have implicit values and are thus not conveyed in the data memory. P0 = 0 and P1 = 1. **The param_len does not include these two entries.**

The binary code of VLOOP includes a *version* field, which allows a compiler/assembler to indicate which version of EVE hardware the vector command is compiled for. The first version has *version* = 0, the next version, *version* = 1, and so on. This allows future version of EVE to offer backward compatibility; that is, execute previous-version binary as is. For example, if a future version of EVE has 16-way SIMD, it can execute 8-way SIMD binary by disabling half of the datapath.

Each vector command can have:

- Up to 8 address generators
- Up to 16 vector register initialization
- Up to 8 loads
- Up to 40 operations
- Up to 8 stores

Maximal command length is 81 instructions (VLOOP + 16 VINITs + 8 VAGENs + 8 VLDs + 40 operations + 8 VSTs).

8.3.5.5.1.1 Retention of State Between VLOOPs

From one VLOOP to the next VLOOP, the following state information is retained:

- Parameter pointer, advanced by parameter length in the first VLOOP

- Vector register contents, whatever left over from the last iteration of the first VLOOP

The following state information is not retained, thus must be configured within each VLOOP:

- Parameter register file (must be reloaded from data memory)
- Agen configuration, such as association of the four counts to parameter registers
- Agen address counter (reset at the beginning of each VLOOP execution)

8.3.5.5.2 Vector Register Initialization

It is useful to initialize vector register with constants provided in the parameter file. As parameter file is scalar (not NWAY SIMD), each value is broadcast to all NWAY of the destination vector register. Assembly syntax:

VINIT *type* **_** *init_loop preg* **,** *vreg*

The data *type* can be signed/unsigned halfword or word. In case of word, two parameter register entries (16-bit each) are used, first one being copied to the lower halfword, and the second one to the upper halfword.

For the word type initialization, *preg* specified is the lower halfword and must be even. For example, *VINITWU_ONCE P10, V2* initializes V2 using (P11 << 16) + P10.

The *init_loop* field can be {ONCE, I234_ZERO, I34_ZERO, I4_ZERO, ALWS }.

The VINIT instruction is equivalent to initializing a specific vector register in the nested i1/i2/i3/i4 loop shown in Section 8.3.5.3.2:

- ONCE is for initialization before the loop.
- I234_ZERO is for initialization inside the i1 loop, before entering the i2 loop.
- I34_ZERO is for initialization inside the i2 loop, before entering the i3 loop.
- I4_ZERO is for initialization inside the i3 loop, before entering the i4 loop.
- ALWS is for initialization inside the i4 loop, running every iteration.

The VINIT feature is used to initialize vector registers to certain initial values. For example, FIR filtering is implemented by iterating over filter taps with MADD instructions, and VINIT is used to initialize the accumulator to zero at the proper iteration, i4 for 1-D filter, and i3 for 2-D filter. Constants are brought into the computation loop with VINIT as well. For example, to scale an entire input array by a constant factor.

Note that VINIT broadcasts one value to all SIMD lanes. Initializing a vector register to multiple values is accomplished with a vector load (VLD) with zero address increment.

Sometimes it is useful to refer to loop variables in the computation. For example, to collect the maximal five values in an array and recording their indices, software can first do this in SIMD fashion to get to 5 sets of values and indices, then merge among SIMD lanes. Indices are readily copied from the loop variable i4, instead of spending cycles to compute with ALU operations.

VINIT provides this capability by extending the parameter count to cover loop variables:

- P64 = i0, the repeat outer loop enabled via VCTRL <scalar_reg>, RPT_END
- P65 = i1
- P66 = i2
- P67 = i3
- P68 = i4

When using these parameter counts, only the unsigned halfword type is supported.

8.3.5.5.3 Address Generator (agen)

In image, video, vision computation, there are often many operations performed on arrays of the same dimension; which can use the same addressing pattern (with different base addresses) in the memory read/write. The address generator resource is included to take advantage of this. One agen can be shared among multiple loads and/or multiple stores.

The VAGEN instruction has this assembly format:

VAGEN *agen*, [*pinc4* **,** *pinc3* **,** *pinc2* **,** *pinc1*]

Agen field is a label (A0..7) to identify one of 8 agen resources.

VAGEN is defined in strict linear order A0, A1, A2, till the last agen used in the command. The agens defined can be used (in loads and stores) in any order. It is allowed to share agen between load(s) and store(s).

The address increments represent:
- pinc4: inside the i4 loop, default
- pinc3: inside the i3 loop, when i4 matches the ending loop count
- pinc2: inside the i2 loop, when where i3 and i4 both match the ending loop counts
- pinc1: inside the i1 loop, when i2, i3, and i4 all match the ending loop counts

Figure 8-62 shows an example of 2 × 3 × 2 × 4 data object, addressed by the loop variables i1, i2, i3 and i4, with i1 being the outer-most loop. The address increments to access the object are shown in Figure 8-62.

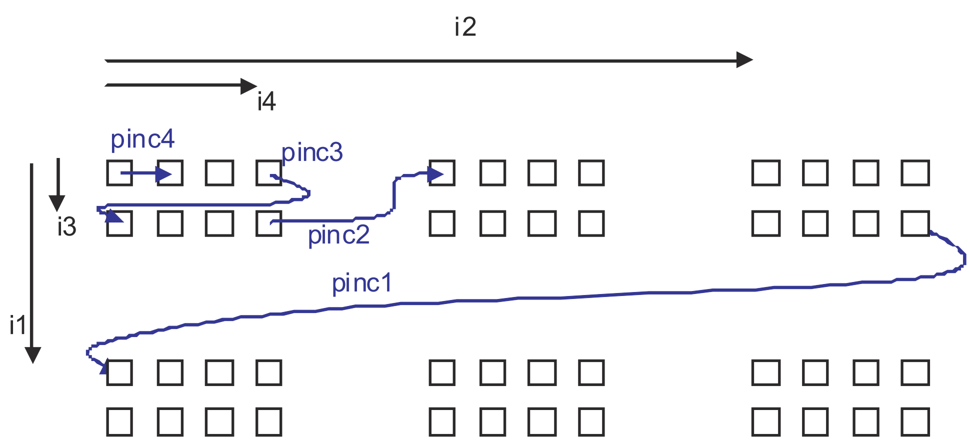


Figure 8-62. Addressing a Four-Dimensional Data Object

In the VAGEN instruction, pointers to the parameter file are encoded, not the address increments themselves. Most data objects do not require 4-dimension addressing. For example, coefficient data for 2D FIR filter is normally stored consecutively, and advance by 1 data point most of the times, and wraps back at the end of i3 loop.Then pinc4 = pinc3 is specified to one parameter register (initialized to 2 for 16-bit coefficients), and pinc2 = pinc1 to another parameter register for the wrap-back. Thus, this scheme reduces command plus parameter data.

All address increments are in bytes. This allows misaligned halfword (16-bit) and word (32-bit) accesses that are some times needed in the processing.

Each address increment is encoded with one 16-bit parameter register, treated as signed value. The above example shows address increment moving forward, but in many cases, for example coefficients for filtering, negative increments are used.

Note that the address increments together specify addressing pattern of a single pointer, and the data distribution controls how consecutive memory data from that pointer, up to 32N-bit, are distributed among the SIMD dimension of the vector register entry.

In case multiple VAGEN instructions are issued for the same agen resource (for example, A0), the last one takes effect, and the preceding ones with the same agen resource is ignored.

8.3.5.5.4 Vector Load

The VLD instruction specifies the data distribution, data type, base address, agen, and the vector register to load the data to. Format of the VLD instruction in assembly is:

```
VLD type _ distribution base [agen], vreg
```

The *base* field points to a pair of parameter registers (even/odd pair only), for example base = P8 to refer to the P8:P9 pair. The vector data memory space is 20-bit, with the lower 16-bit encoded in the even parameter register (P8 in this case), the upper 4-bit in the odd parameter register (P9 in this case).

The destination vector register specified in VLD must be an even register. VLD_DINTRLV loads into an even/odd pair of registers. VLD with any other distribution option loads into one register.There are up to eight VLD instructions in a loop to potentially load up all 16 registers. No register can be the destination of more than one VLD instruction in the same loop.

As EVE is an N-way SIMD machine, loading of N consecutive data points from local memory is supported, with each data point being a byte, short, or long (32-bit) word, either signed or unsigned.

In addition to loading N data points, EVE supports a few other distribution options. EVE supports the following:

- NPT: N data points, one to each way of SIMD
- 1PT: broadcasting one data point to all
- CIRC2: circuiting between two data points, useful for Bayer image processing
- DS2: downsample by 2; every other data point (**not available for word-size input**)
- US2: upsample by 2; repeat each input data point twice
- DINTRLV: deinterleave inputs; load 2N data points into two registers, deinterleave data items so that even items go to the first register, odd items go to the second register. Supported only for byte or halfword per data point, **not word data type**. Two registers must be an even/odd pair, but only the first register (even) is encoded in assembly and in instruction binary.
- CUST_P8: custom distribution as provided by {P8..} in the parameter file
- CUST_P16: custom distribution as provided by {P16..} in the parameter file
- CUST_P24: custom distribution as provided by {P24..} in the parameter file
- EXP: load with expansion using predicate register V0
- NBITS: N bits, one bit to each way of SIMD, regardless of the data type (for example, when N = 16, a halfword is read). Signed/unsigned data type affects how this bit is to be sign-extended. Unsigned means 0 -> 0, 1 -> 1, and signed means 0 -> 0, 1 -> -1. This is useful for expanding load and other cases where boolean arrays are packed.

For CUST_Pi options, number of bits required to specify the address offset for each way of SIMD depends on N, the number of ways of SIMD. Arbitrary distribution of N data items (each a byte, halfword, or word) is supported. For N = {2 | 4 | 8 | 16}, 4 bits is used to encode each destination. For N = 32, 5 bits is used; and each 16-bit parameter register supplies three fields (leaving 1 bit unused).

- N = 2, 2 × 4 = 8 bits, one parameter register required
- N = 4, 4× 4 = 16 bits, one parameter register required
- N = 8, 8 × 4 = 32 bits, two parameter registers required
- N = 16, 16 × 4 = 64 bits, four parameter registers required
- N = 32, ceiling(32/3) = 11 parameter registers required

The DINTRLV option is supported so software can fully utilize the two function units for simple operation sequences, by processing 2N data points in parallel.

VLD instruction normally loads into one destination vector register. VLD_DINTRLV is the exception in that it loads into a pair of vector registers. Only even registers, V0, V2, ..., V14, can be referenced in the destination register field. In case of VLD_DINTRLV, the referenced register and the next register shall be the destination, for example V0 and V1.

The following data types are supported:

- B: signed byte
- BU: unsigned byte
- H: signed halfword (16-bit)
- HU: unsigned halfword
- W: signed word (32-bit)
- WU: unsigned word

Address offsets for each way of SIMD, as the function of distribution option and data type are shown in Table 8-353 and Figure 8-63, Figure 8-64, and Figure 8-65.

Table 8-353. VLD Data Distribution Options

Distribution(1)(2)	vreg[r][0] gets	vreg[r][1] gets	vreg[r][2] gets	vreg[r][3] gets	vreg[r][4] gets	vreg[r][5] gets	vreg[r][6] gets	vreg[r][7] gets
NPT	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]	data[7]
1PT	data[0]	data[0]	data[0]	data[0]	data[0]	data[0]	data[0]	data[0]
CIRC2	data[0]	data[1]	data[0]	data[1]	data[0]	data[1]	data[0]	data[1]
DS2	data[0]	data[2]	data[4]	data[6]	data[8]	data[10]	data[12]	data[14]
US2	data[0]	data[0]	data[1]	data[1]	data[2]	data[2]	data[3]	data[3]
DINTRLV	vreg[r][0] = data[0], vreg[r+1][0] = data[1]	vreg[r][1] = data[2], vreg[r+1][1] = data[3]	vreg[r][2] = data[4], vreg[r+1][2] = data[5]	vreg[r][3] = data[6], vreg[r+1][3] = data[7]	vreg[r][4] = data[8], vreg[r+1][4] = data[9]	vreg[r][5] = data[10], vreg[r+1][5] = data[11]	vreg[r][6] = data[12], vreg[r+1][6] = data[13]	vreg[r][7] = data[14], vreg[r+1][7] = data[15]
CUST_Pi(3)	data[pf[0]]	data[pf[1]]	data[pf[2]]	data[pf[3]]	data[pf[4]]	data[pf[5]]	data[pf[6]]	data[pf[7]]

(1) Table “data” is type-cast to the appropriate signed/unsigned byte/halfword/word so that correct offsets are applied.
(2) This is a table of what data element gets loaded into each SIMD lane of a vector register, in the order lane 0 to lane 7. This does not reflect a debugger memory window hex display.
(3) pf[0], pf[1], ... pf[7] refer to bit fields in the indicated parameter registers. For example, for the 8-way SIMD EVE, it takes two parameter registers for the distribution information, so for CUST_P8 use: distrib_info = (P9 << 16) | P8, and assign (distrib_info >> (4 × i)) & 0xF to pf[i].

Loads are automatically predicated. Loads are always performed at the very first iteration, and then subsequently upon any change in the associated agen address pointer.

See Section 8.3.5.5.10 for address alignment requirement for various data type and distribution options.

LD_EXP is load with expansion, and is for reading a compacted (collated) array and expanding the elements to the original locations. It is the opposite of ST_COLLAT (collating store). Per-item address increment is implied by the load data type (1/2/4 bytes depending on byte/halfword/word type). The agen field is don’t care. Per iteration, load pointer (VCOP_LD_PTR_i/VCOP_ST_PTR_j) is increment by number of non-zeroes in predicate register V2 times the data size. Agen is not used, and so the agen field is not required in the assembly code:

VLD *type* **_ EXP** *base* **,** *vreg*

For example with this instruction, where V2 = {0, 0, 1, 0, 1, 1, 0, 0}, load_ptr = 0x100:

```
LDBU_EXP Pbase[A0], V1
```

There shall be, after this instruction:

```
V1 = {0, 0, mem[0x100], 0, mem[0x101], mem[0x102], 0, 0}, and load_ptr = 0x103
```

Load with expansion is restricted to be used in table lookup VLOOP. This is because the dependency (where the expanding load depends on predicated register data from another load) is more similar to table lookup, and thus it’s feasible to achieve N data points per clock cycle inside table lookup pipeline. As in normal table lookup, no operation instructions are allowed in table lookup VLOOP.

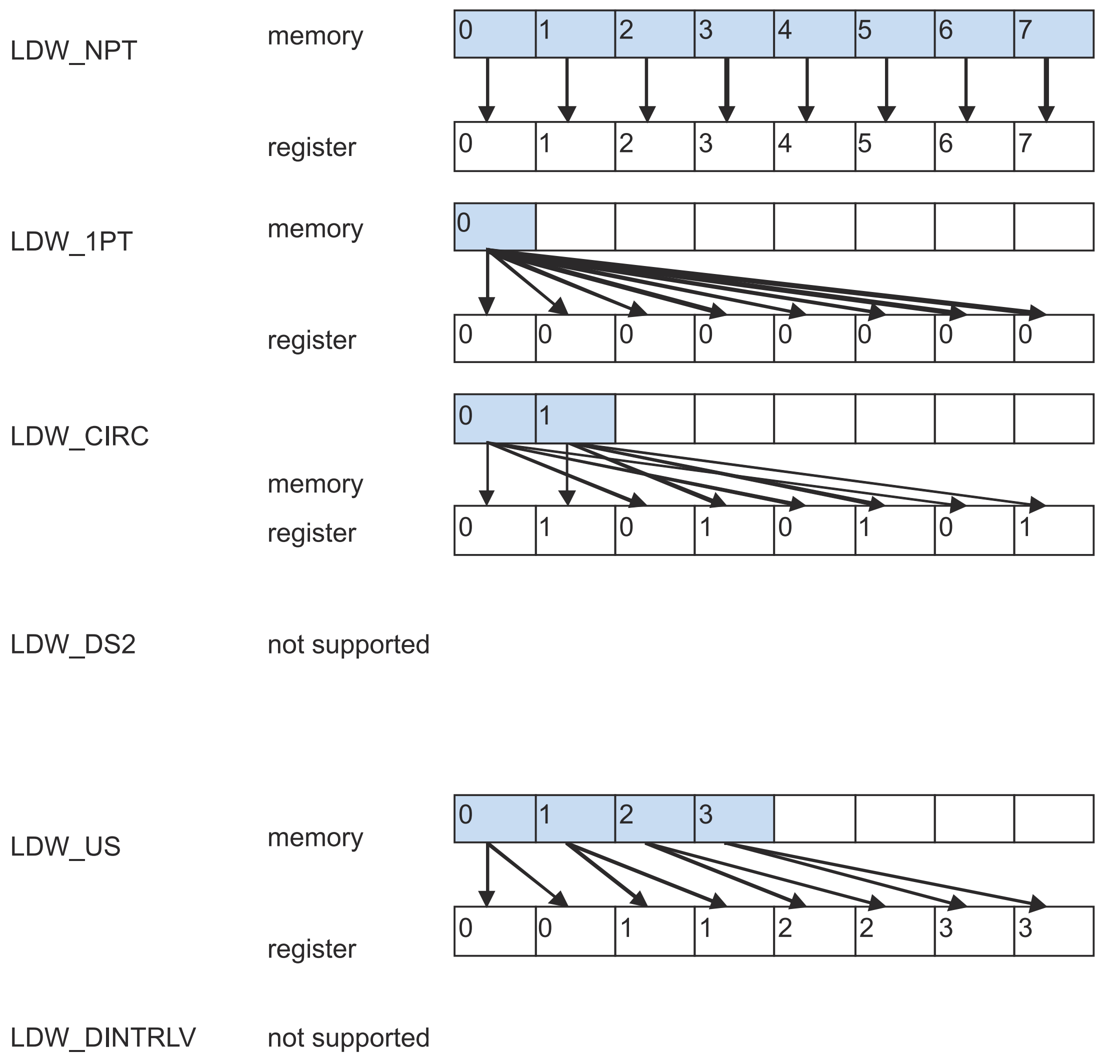


Figure 8-63. Load Word Distribution Options

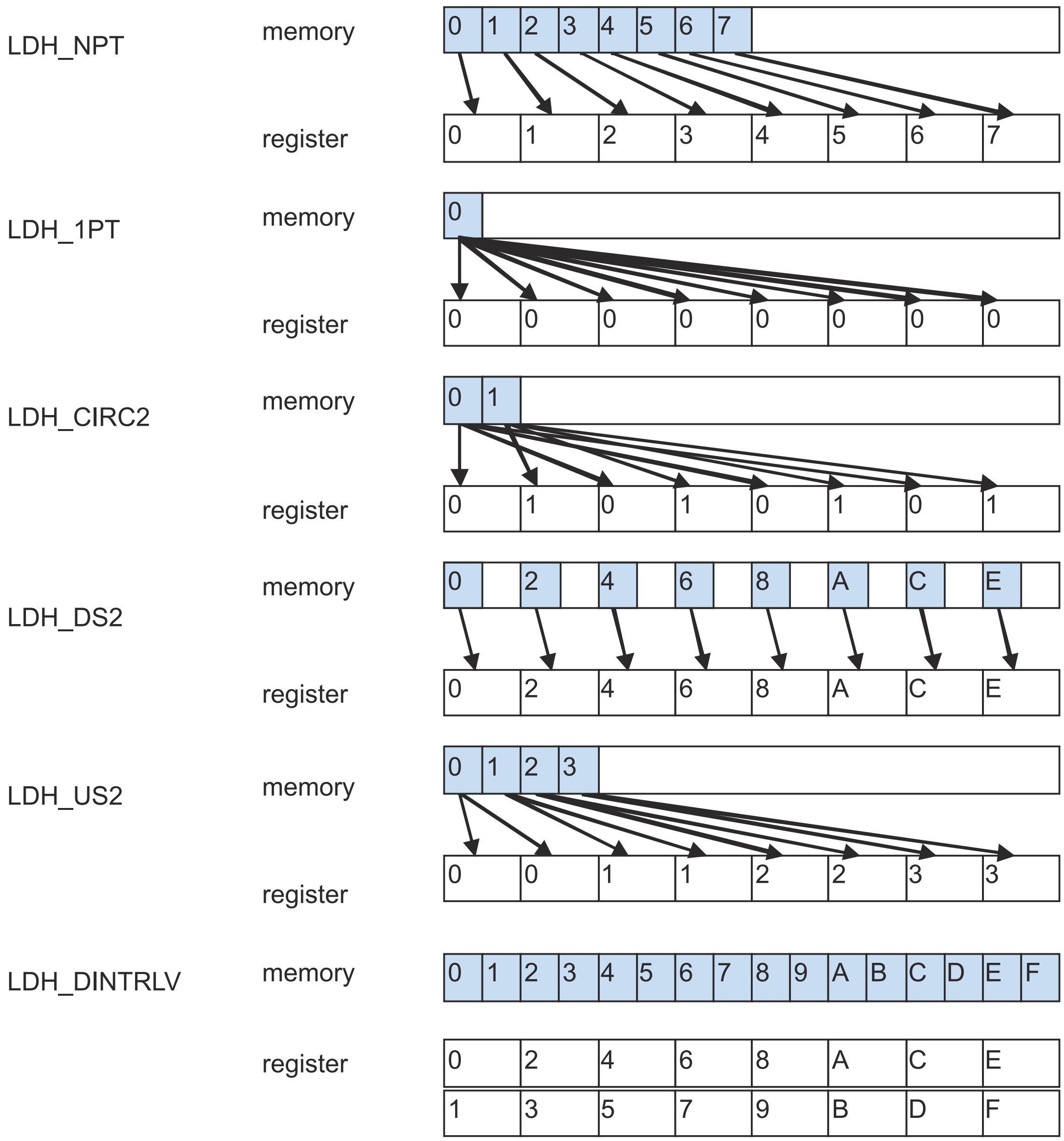


Figure 8-64. Load halfword Distribution Options

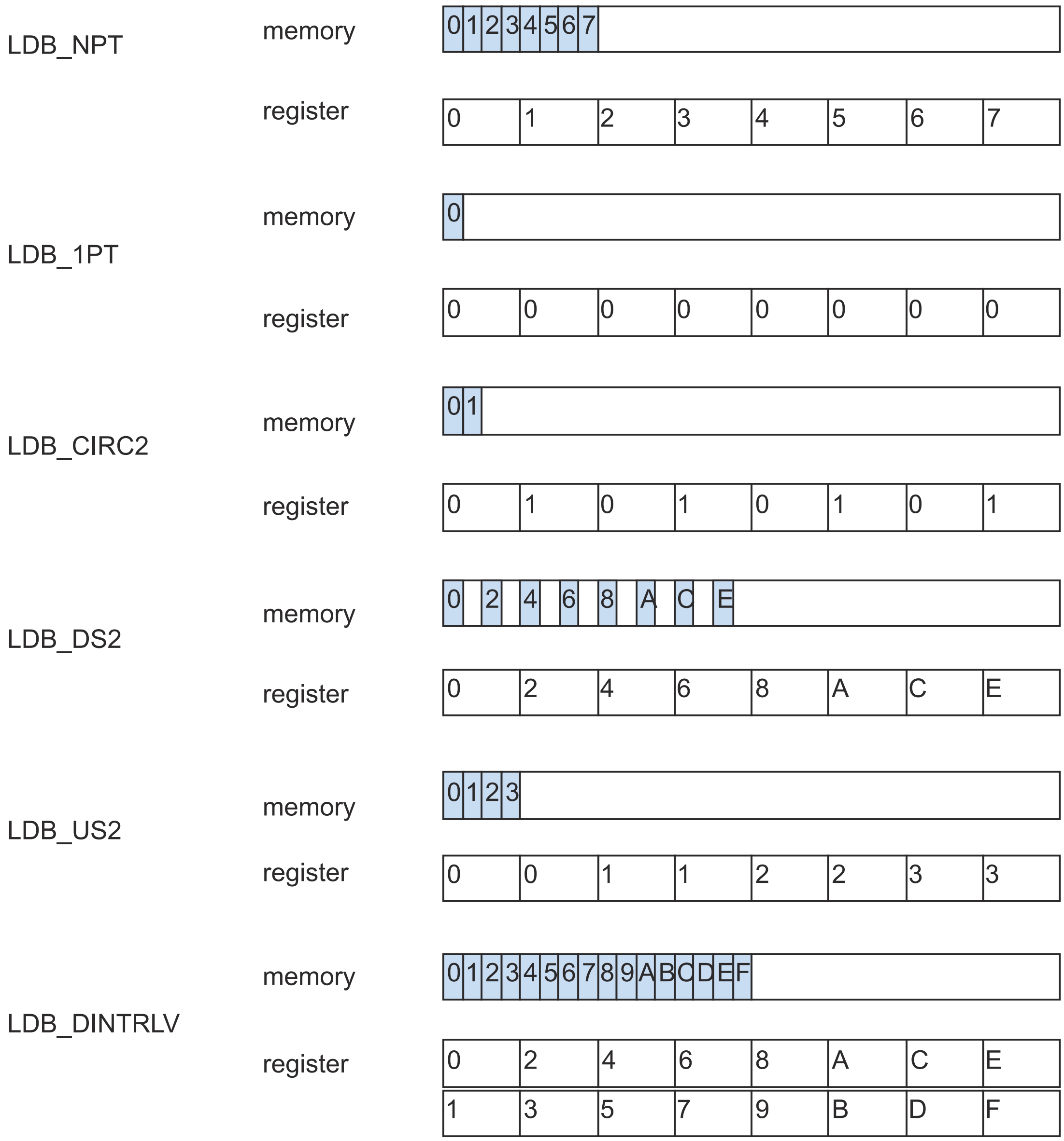


Figure 8-65. Load Byte Distribution Options

8.3.5.5.5 Vector Arithmetic/Logic Operations

Various arithmetic and logic operations are available in the compute command, which is indicated by starting the command with:

VLOOP COMP, *cmd_len* **,** *param_len*

Vector arithmetic/logic instructions have the following assembly formats, depending on whether it is 1-input-1-output, 2-input-1-output, 2-input-2-output, or 3-input-1-output, whether accumulator clearing is enabled, and whether rounding is enabled.

V<op_1i1o> src1, dst

V<op_2i1o> src1, src2, dst

V<op_2i1o> src1, src2, dst, RND: rnd_param

V<op_2i2o> src1/dst1, src2/dst2

V<op_2i2o> src1, src2/dst1, dst2

V<op_3i1o> src1, src2, src3, dst

V<op_3i1o> src1, src2, src3, dst, RND: rnd_param

All operations are 40-bit, except:
- Input to multiply (MPY, MADD, MSUB) is 17-bit data.
- BINLOG, BITC, BITDI, BITI, BITR are 32-bit instructions.
- BITPK, BITUNPK, SORT2, MIN, MAX are 33-bit instructions (to allow processing of up to 32-bit signed data)
- BITTR, bit transpose, bit width is limited to ways of SIMD, thus 8-bit data for 8-way SIMD.

The operations in Table 8-354 are supported. See Section 8.3.5.9 for details on each operation.

Table 8-354. VCOP Arithmetic/Logic Operations

Operation	#In- Out	#Bits	#Del	Syntax[(1)]	Note
VNOP		40		VNOP	
VADD	2-1	40		VADD src1, src2, dst	src1 + src2
VSUB	2-1	40		VSUB src1, src2, dst	src1 – src2
VABSDIF	2-1	40		VABSDIF src1, src2, dst	|src1 – src2|
VMPY	2-1	17/33	1	VMPY src1, src2, dst, RND:rnd_param	src1 * src2
VAND	2-1	40		VAND src1, src2, dst	src1 & src2
VOR	2-1	40		VOR src1, src2, dst	src1 | src2
VXOR	2-1	40		VXOR src1, src2, dst	src2 ^ src2
VMIN	2-1	33		VMIN src1, src2, dst	min(src1, src2)
VMAX	2-1	33		VMAX src1, src2, dst	max(src1, src2)
VANDN	2-1	40		VANDN src1, src2, dst	src1 & (~src2)
VSHF	2-1	40/6		VSHF src1, src2, dst	src1 << src2, or src1 >> (-src2)
VRND	2-1	40/5	1	VRND src1, src2, dst	(src1 + (1 << (src2-1)) >> src2
VCMPEQ	2-1	40		VCMPEQ src1, src2, dst	(src1 == src2) ? 1 : 0
VCMPGT	2-1	40		VCMPGT src1, src2, dst	(src1 > src2) ? 1 : 0
VCMPGE	2-1	40		VCMPGE src1, src2, dst	(src1 >= src2) ? 1 : 0
VBINLOG	1-1	32	1	VBINLOG src1, dst	approximate binary log
VBITC	1-1	32	1	VBITC src1, dst	count one bits
VNOT	1-1	40		VNOT src1, dst	~src1

Table 8-354. VCOP Arithmetic/Logic Operations (continued)

Operation	#In- Out	#Bits	#Del	Syntax[1]	Note
VMADD	3-1	17/40	1 / 2	VMADD src1, src2, src3, dst, RND: rnd_param	src3 + src1 * src2
VMSUB	3-1	17/40	1 / 2	VMSUB src1, src2, src3, dst, RND: rnd_param	src3 – src1 * src2
VADD3	3-1	40	1	VADD3 src1, src2, src3, dst	src1 + src2 + src3
VSAD	3-1	40	1	VSAD src1, src2, src3, dst	src3 + abs(src1 – src2)
VSEL	3-1	40		VSEL src1, src2, src3, dst	src1 ? src2 : src3
VAND3	3-1	40	1	VAND3 src1, src2, src3, dst	src1 & src2 & src3
VOR3	3-1	40	1	VOR3 src1, src2, src3, dst	src1 | src2 | src3
VSHFOR	3-1	40/6	1	VSHFOR src1, src2, src3, dst	src3 | (src1 << src2) or src3 | (src1 >> -src2)
VSORT2	2-2	33		VSORT2 src1/dst1, src2/dst2	dst1 = min(src1, src2) dst2 = max(src1, src2)
VBITPK	2-1	33	1	VBITPK src1, src2, dst	compare, bit-pack, broadcast
VBITUNPK	2-1	40	1	VBITUNPK src1, src2, dst	bit unpack
VEXITNZ		40		VEXITNZ level, src1	exit loop at end of iteration when (src1 != 0)
VCMOV		40		VCMOV cond, src1, dst	conditional move
VBITR	1-1	32	1	VBITR src1, dst	bit reverse
VBITI	2-1	32	1	VBITI src1, src2, dst	bit interleave
VBITDI	1-2	32		VBITI src1, dst1, dst2	bit deinterleave
VABS	1-1	40		VABS src1, dst	abs(src1)
VADDH	2-1	40		VADDH src1, src2, dst	src1+ signext(src2[39:32])
VLMBD	3-1	40	1	VLMBD src1, src2, dst	left-most-bit-detect
VBITTR	1-1	NSIMD	1	VBITTR src1, dst	bit transpose
VSIGN	2-1	40		VSIGN src1, src2, dst	apply sign of src1 on src2
VADDSUB	2-2	40		VADDSUB src1/dst1, src2/dst2	dst1 = src1 + src2 dst2 = src1 – src2
VINTRLV	2-2	40		VINTRLV src1/dst1, src2/dst2	interleave
VDINTRLV	2-2	40		VDINTRLV src1/dst1, src2/dst2	deinterleave
VMINSETF	2-2	33		VMINSETF src1, src2/dst1, dst2	minimum and set flag
VMAXSETF	2-2	33		VMAXSETFsrc1, src2/dst1, dst2	maximum and set flag
VINTRLV2	2-2	40		VINTRLV2 src1/dst1, src2/dst2	interleave with 2-element frequency
VDINTRLV2	2-2	40		VDINTRLV2 src1/dst1, src2/dst2	deinterleave with 2-element frequency
VINTRLV4	2-2	40		VINTRLV2 src1/dst1, src2/dst2	interleave with 4-element frequency
VSHF16	1-2	33		VSHF16 src1, dst1, dst2	shift up 16 bits into 2 registers
VADIF3	3-1	40	1	VADIF3 src1, src2, src3, dst	add difference, dst = src1 – src2 + src3
VSWAP	3-2	40		VSWAP cond, src1/dst1, src2/dst2	conditional swap

(1) The parameters src1, src2, src3, dst, dst2 refer to the vector register entry for source or destination.

Operations with two destinations must use two different destination registers, otherwise the outcome is undefined.

The accumulating register (being a source as well as the destination of an operation) must be src1 for 2-input-1-output operations, and must be src3 for 3-input-1-output operations.

The rounding parameter is needed for VMPY, VMADD and VMSUB instructions, and specifies an index to the parameter file, see Section 8.3.5.9.

EVE hardware executes up to 2 operations in parallel per clock cycle. Assembly program (by programmer or by compiler) contains the parallel bar notation to indicate if an instruction is to be executed by itself, or is to be executed in parallel with another instruction.

VMADD, VMSUB instruction have two delay slots for the multiplication input, and one delay slot for the addition/ subtraction input.

VMPY, VRND, VBINLOG, VBITC, VADD3, VSAD, VAND3, VOR3, VSHFOR, VBITPK, VBITUNPK, VBITR, VBITI, VLMBD, VBITTR, VADIF3 instructions have one delay slot.

All other operations do not have delay slots.

The hardware detects and treats write/read dependency between any two sequential instructions, but not inside parallel executed instruction pairs. When necessary for correctness, hardware inserts idle cycles automatically. Still to achieve good performance, software must try to schedule the operations to avoid automatic idle cycles.

There is a forwarding path *within each* functional unit and *between* the two functional units to forward from destination to source 3 (accumulator input) of 3-input-1-output operations.

Register forwarding, dependency checking and automatic idle cycle insertion work across iterations as well, from end of one iteration to the beginning of the next iteration. For example, the FIR filtering kernel executes in one cycle per iteration with the destination-to-source3 dependency across iterations.

Table 8-355 shows delay slots and automatically inserted idle cycles.

Table 8-355. Example of Operation Delay Slots

Time	VMPY	VMADD	VSUB	VMSUB	VAND	VOR	VADD
0	V0*V1						
1	V0*V1 => V7	V2*V3	V2-V6 => V6				
2		V2*V3 =>p		V4*V5	V3 & V6 => V6		
3		V7 + p => V7		V4*V5 => p		wait for V7	
4				V7 - p => V7		wait for V7	
5						V1|V8 => V9	V6 + V7 => V6

Zero-delay slot operation executes in one cycle. VMPY, an one-delay slot opeartion takes two cycles to execute. VMADD and VMSUB, two-delay slot operations take 3 cycles for execution, but need its additional input only on the third cycle. The dependency between the two instructions (VMADD-VMSUB) on the additional operand does not introduce idle cycles, but VMSUB-VADD dependency adds 2 idle cycles to execution per iteration.

8.3.5.5.6 Vector Store

The VST instruction specifies the data distribution, data type, base address (index to parameter file), agen, the vector register to get the store from, loop level when stores are to occur, and rounding/saturation configuration. Format of VST is:

[*pred* **] VST** *type* **_** *distribution* **_** *wr_loop vreg* **,** *base* **[** *agen* **], RND_SAT:** *rnd_sat_param*

Storing of up to N data points from vector register to local memory is supported, with each data point being a byte, short, or long (32-bit) word, either signed or unsigned. Signed/unsigned information is used in performing saturation.

There are up to eight VST instructions in one loop, and each VST can be from the full vector register file, V0..V15. It is allowed to overlap the source registers, i.e., having more than one VST instructions storing out the same vector register. Also, in the extreme case of having all eight VST being interleaving stores, all 16 vector registers can be written out.

There is a further constraint on source registers for vector store. Only V0, V1, V2, and V3 can be stored without being the destination of any operation.

VCOP supports the following store patterns:

- NPT: For 8-way SIMD, 8 data points
- 1PT: write only the first SIMD unit
- DS2: downsample by 2; every-other SIMD units are written
- INTRLV: interleave outputs; store 2N data points from two registers interleaved. Two registers must be an {i, i+1} pair (i = 0..14), but only the first register (i) is encoded in assembly and in instruction binary. Byte and halfword types are supported, but not word type. It is NOT allowed to predicate interleaving stores. (If one must predicate interleaving store, use VINTRLV operation to interleave two sets of N data points, then use two predicated VST<type>_NPT.)
- OFFST_NP1: write out all N data points, in (N + 1) words = (4N + 4) bytes address increments (for transposition). Note that although address offset is word-aligned, storing to partial word, in case of byte and halfword types, are supported. In case of byte or halfword type, the 2 LSBs of address determines the byte or halfword within each word that VCOP writes data into.
- COLLAT: collating store, use predicate register to indicate store/not-store, up to N data items stored per cycle, agen is don't care, and hardware increments store address by the size implied by the data type. The agen field is not required in the assembly code:

[*pred* **] VST** *type* **_ COLLAT _** *wr_loop vreg* **,** *base* **, RND_SAT:** *rnd_sat_param*

- SDDA: sequential data-driven addressing, taking M cycles to store M data points (predication can lead to less than N stores), store Vreg[i] to address base + agen + V0[i] Note that agen can be used to vary the pointer, in addition to V0. Only unsigned address offset in V0 is supported.
- PDDA: parallel data-driven addressing. Use this mode when address offsets in V0 is such that all lanes enabled for store go to separate memory banks. Hardware executes the store of up to N data points in one cycle. In case there is bank conflict, outcome is indefinite, and an error interrupt is generated.
- SKIP: store N data points into every other elements in memory. Allowed for byte and halfword types, not for word type.

The following data types are supported:

- B: signed byte
- BU: unsigned byte
- H: signed halfword (16-bit)
- HU: unsigned halfword
- W: signed word (32-bit)
- WU: unsigned word

Table 8-356. EVE ST Data Distribution Options for NWAY = 8

Distribution[(1)] [(2)]	vreg[r][0] goes to	vreg[r][1] goes to	vreg[r][2] goes to	vreg[r][3] goes to	vreg[r][4] goes to	vreg[r][5] goes to	vreg[r][6] goes to	vreg[r][7] goes to
NPT	dptr[0]	dptr[1]	dptr[2]	dptr[3]	dptr[4]	dptr[5]	dptr[6]	dptr[7]
1PT	dptr[0]	n/a	n/a	n/a	n/a	n/a	n/a	n/a
DS2	dptr[0]	n/a	dptr[1]	n/a	dptr[2]	n/a	dptr[3]	n/a
INTRLV	dptr[0] = vreg[r][0], dptr[1] = vreg[r+1][0]	dptr[2] = vreg[r][1], dptr[3] = vreg[r+1][1]	dptr[4] = vreg[r][2], dptr[5] = vreg[r+1][2]	dptr[6] = vreg[r][3], dptr[7] = vreg[r+1][3]	dptr[8] = vreg[r][4], dptr [9] = vreg[r+1][4]	dptr[10] = vreg[r][5], dptr[11] = vreg[r+1][5]	dptr[12] = vreg[r][6], dptr[13] = vreg[r+1][6]	dptr[14] = vreg[r][7], dptr[15] = vreg[r+1][7]
OFFST_NP1	dptr[0]	dptr[9]	dptr[18]	dptr[27]	dptr[36]	dptr[45]	dptr[54]	dptr[63]
COLLAT	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++
SDDA/PDDA	dptr[V0[0]]	dptr[V0[1]]	dptr[V0[2]]	dptr[V0[3]]	dptr[V0[4]]	dptr[V0[5]]	dptr[V0[6]]	dptr[V0[7]]
SKIP	dptr[0]	dptr[2]	dptr[4]	dptr[6]	dptr[8]	dptr[10]	dptr[12]	dptr[14]

(1) Table "dptr" is type-cast to the appropriate signed/unsigned byte/halfword/word so that correct offsets are applied. dptr = base + agen.
(2) Custom distribution is not supported for stores.

Stores are predicated to perform at the indicated loop level, with wr_loop selectable from:
- ALWS: write out on every i4 iteration
- LAST_I4: write out on the last i4 iteration, when i4 = lpend4
- LAST_I34: write out on the last i3 iteration, when i3 = lpend3 and i4 = lpend4
- LAST_I234: write out on the last i2 iteration, when i2 = lpend2, i3 = lpend3, and i4 = lpend4

Rounding and saturation arithmetic is available in the store pipeline to provide rounding and saturation without taking additional cycles.

Rounding and saturation parameters are provided in the parameter file to allow easy adaptation of EVE functions to different rounding and saturation configurations. The rnd_sat_param field in the instruction points to a parameter (instruction encoding limits this field to 5 bits, thus restricting it to P0..P31) that contains the following fields shown in Figure 8-66.

Figure 8-66. VST Rounding and Saturation Parameters

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sat_mode: 0 = NO_SAT 1 = SYMM 2 = ASYMM 3 = 4PARAM 4 = SYMM32 5 = ASYMM32			sat_bound_param: index to param that specifies saturation bound						rnd_mode: 0 = no rounding 1 = round 2 = truncate		rnd_shift: number of bits to round/shift down				

The following saturation modes are supported:
- NO_SAT: no saturation performed
- SYMM: signed symmetrical saturation [-bound, bound] (for unsigned store, [0, bound])
- ASYMM: signed asymmetrical saturation [-bound-1, bound] (for unsigned store, [0, bound]), useful for fixed bit width. For example, when bound = 1023, saturate to [-1024, 1023]
- 4PARAM: use 4 parameter registers to specify sat_high_cmp, sat_high_set, sat_low_cmp, sat_low_set.
- SYMM32: same as SYMM, except using a pair of parameters to specify a 32-bit bound
- ASYMM32: same as ASYMM, except using a pair of parameters to specify a 32-bit bound

Signed/unsigned designation is on the VST type, and signed/unsigned designation affects whether the bound parameter(s) are interpreted as signed or unsigned. The comparison is carried out between signed 40-bit register and such interpreted bounds.

When no rounding and saturation feature is needed, specifying RND_SAT: P0 is working, since constant 0 in these fields represents doing nothing.

Conditional store is specified by the pred field. Omission means no predication. Pred = V1, V2, V3 indicates that register V1, V2, V3 is used to predicate the store of each way of SIMD; store is executed when the register is not zero.

With the COLLAT distribution option, all SIMD ways are examined for conditional store. Truth in the predication register leads to the data item being stored and incrementing of the data pointer by size of the store type (1/2/4 bytes). False in the predication register leads to doing nothing. Thus, stored data items are compacted without leaving any gaps for the blocked data items.

With other distribution options, conditional store behaves differently. Agen is incremented as in unconditional stores. Truth in the predication register leads to the data item being stored, and false in the predication register leads to the data item being skipped. Thus, it is possible for the stored data items to contain gaps among them from the blocked data items.

The store stage hardware has separate resource for each VCOP data memory region (IBUFL, IBUFH, WBUF). Sequential data-driven store is executed in X cycles, X being the number of data items enabled in predication (or when predication is not used, N). The other store modes are executed in a single cycle per store (note that collating store executes in one cycle as well). All memory regions are carried out in parallel, but there is also potential memory contention with the load stage. There is write buffering in the store stage to reduce load/store contention to some degree. See Section 8.3.5.6 for load/store buffering details.

8.3.5.5.7 Table Lookup Operation

Table lookup operation is indicated by starting the vector command with

VLOOP TLU, *CL#* **:** *cmd_len* **,** *PL#* **:** *param_len*

The table lookup operation uses a subset of the available resources and instructions:
- VAGEN (address generator): fixed A0 for data, A1 for table pointer, A2 for output.
- VLD (load): one for loading data into V2.
- VTLD (table load): for performing the lookup, using V2 as indices to load table entry into V0.
- VST (store): for storing outcome in V0.
- Full set of 4 levels for looping.

The table pointer is allowed to move via agen, to allow loop-variable-dependent manipulation of table base. For example, software might process three color components (RGB or YUV), each having a different table, in the same command, and use the outer loop variables to index among the color components and the corresponding tables.

Data is loaded with normal load into V2, for example:

```
        VLDBU_1PT       data_base[A0], V2
```

The data load using VLD has access to a subset of load distribution options: {NPT, 1PT, DS2, US2}.
- 1PT works for single table configuration
- NPT works for all other cases of normal table lookup
- DS2 and US2 provides additional flexibility to downsample/upsample data

A special table load, VTLD, is used for the lookup:

VTLD *type* **_** *m* **TBL_** *n* **PT** *tbl_base*[*tbl_agen*][**V2**]**, V0, RND_SAT:** *rnd_sat*

Normally lookups only fetch one item per lookup. Multiple items are useful for bilinear and bicubic interpolations. The _*m*TBL field in VTLD specifies the number of parallel tables and the _*n*PT field in VTLD specifies the number of data items per lookup table.

There are constraints on num_par_tbl, num_data_per_lu, and the table data size:
- num_par_tbl × num_data_per_lu <= N (ways of SIMD)
- num_data_per_lu = {1 | 2 | 4} regardless of N (ways of SIMD)

Constraints for 8-way SIMD architecture are shown in Table 8-357.

Table 8-357. Lookup Constraints for 8-Way SIMD

Table type	Num items per lookup, num_data_per_lu	Number of parallel tables, num_par_tbl			
		1	2	4	8
Byte	1	√	√	√	√
	2	√	√	√	
	4	√	√		
	8	√			
Half word	1	√	√	√	√
	2	√	√	√	
	4	√	√		
	8	√			
Word	1	√	√	√	√
	2	√	√	√	
	4	√	√		
	8	√			

Results are stored back to memory with a normal data store as in compute loop with simplification:

VST*type_distribution* **_ALWS V0,** *base*[*agen*]**, RND_SAT:** *rnd_sat_param*

This VST has access to a subset of store distribution optons: {NPT | SKIP}.

Number of data points that are stored per i4 iteration is num_data_per_lu × num_par_tbl from the table load, and this is an implicit predication.

For example, when there is table load VTLDHU_1TBL_2PT, the store VSTHU_NPT is not storing N points, but just 2 point.

In forming the indices for lookup, data are read in, rounded and saturated. Rounding and saturation parameters are specified in VTLD's RND_SAT: rnd_sat field. This field is 5-bit (like VST), so only P0..P31 is allowed. The same set of round/saturate parameters as the normal store, VST, are used. VTLD's RND_SAT is for round/saturate the data to form table lookup index. VST also contains a RND_SAT field, which is used to round/ saturate looked up table entries before storing back to data memory.

Various other table lookup parameters are extracted from the instructions:
- Input data type is from VLD*type*, and is limited to unsigned types.
- Table entry type is from VTLD*type*, and can be signed or unsigned although the signed/ unsigned information is not used.
- Output type is from VST*type*, can be signed or unsigned, and can be different from the table entry type. Signed versus unsigned information is used for saturation.
- Input data distribution is from VLD V2, with SIMD units 0..num_par_tbl-1 used for the lookup.
- Output distribution option is either NPT, storing consecutively (which may not be N points), or SKIP, storing to every other memory location.

- Table entries are read into the vector register SIMD lanes by iterating on num_par_tbl in the outer loop, and num_data_per_lu in the inner loop. For example, with num_par_tbl = 2 and num_data_per_lu = 4, with VSTBU_NPT we have:

```
        V0[0] = tbl_base[data0],      V0[1] = tbl_base[data0 + 1],
        V0[2] = tbl_base[data0+2],    V0[3] = tbl_base[data0 + 3],
        V0[4] = tbl_base[data1],      V0[5] = tbl_base[data1 + 1],
        V0[6] = tbl_base[data1+2],    V0[7] = tbl_base[data1 + 3]
```

Table organization in memory depends on table type and num_par_tbl. The N physical banks of 32-bit memory is partitioned into num_par_tbl logical banks, and each parallel table is organized inside its own bank.

For example, when NWAY = 8, you can have 1, 2, 4, or 8 parallel tables. Data organization for these and byte, halfword, and word type entries is shown in Figure 8-67.

A few examples of the table lookup operation follow. The NPT distribution for load and store works for normal lookup scenarios.

```
; single table, byte data, short table/output
   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A1, ...
   VAGEN            A2, ...
   VLDBU_1PT        data_base[A0], V2                         ; load data to V0
   VTLDHU_1TBL_1PT  table_base[A1][V2], V0, RND_SAT: rnd_sat  ; look up
   VSTH_NPT_ALWS    V0, outp_base[A2], RND_SAT: rnd_sat_st    ; store outcome
```

```
; 4 parallel tables, 2 items per table, short data, byte table/output
   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A1, ...
   VAGEN            A2, ...
   VLDHU_NPT        data_base[A0], V2                         ; load data to V0
   VTLDBU_4TBL_2PT  table_base[A1][V2], V0, RND_SAT: rnd_sat  ; look up
   VSTB_NPT_ALWS    V0, outp_base[A2], RND_SAT: rnd_sat_st    ; store outcome
```

Load with expansion can only be used in a table lookup loop. An example follows.

```
; load with expansion, short input data, unsigned byte flags, short expanded output
   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A2, ...
   VLDBU_NPT        flag_base[A0], V2                         ; load flags to V2
   VLDH_EXP         input_base, V0                            ; load with expansion
   VSTH_NPT_ALWS    V0, outp_base[A2], RND_SAT: rnd_sat_st    ; store outcome
```

For example, in a particular iteration, suppose there are flags V2 = {0, 0, 1, 0, 1, 1, 0, 0}, and the expanding load pointer = 0x100 before the expanding load. After the expanding load:

```
        V0 = {0, 0, mem[0x100], 0, mem[0x102], mem[0x104], 0, 0}
```

The pointer is advanced to 0x106, incremented by 3 (number of nonzero flags) times the size of data type (2 bytes for halfword).

Optionally, predicated store can be used while writing the expanded outcome array, leaving flag==0 data points unaltered.

```
   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A2, ...
   VLDBU_NPT        flag_base[A0], V2                       ; load flags to V2
   VLDH_EXP         input_base, V0                          ; load with expansion
[V2] VSTH_NPT_ALWS  V0, outp_base[A2], RND_SAT: rnd_sat_st  ; store where V2 != 0
```

Performance of load with expansion is NWAY expanded data points per cycle when there is no memory contention, by pipelining the flag load, data load, and output store.

Byte-size single table

a[0], a[1], ... a[31]
a[32], a[33], ... a[63]
...

Byte-size 2 parallel tables

a[0], a[1], ... a[15]	b[0], b[1], ... b[15]
a[16], a[17], ... a[31]	b[16], b[17], ... b[31]
...	...

Byte-size 4 parallel tables

a[0], a[1], ... a[7]	b[0], a[1], ... b[7]	c[0], c[1], ... c[7]	d[0], d[1], ... d[7]
a[8], a[9], ... a[15]	b[8], b[9], ... b[15]	c[8], c[9], ... c[15]	d[8], d[9], ... d[15]
...	...	...	...

Byte-size 8 parallel tables

a[0]... a[3]	b[0]... b[3]	c[0]... c[3]	d[0]... d[3]	e[0]... e[3]	f[0]... f[3]	g[0]... g[3]	h[0]... h[3]
a[4]... a[7]	b[4]... b[7]	c[4]... c[7]	d[4]... d[7]	e[4]... e[7]	f[4]... f[7]	g[4]... g[7]	h[4]... h[7]
...	...	...	...	...	...	...	...

Half-word-size single table

a[0], a[1], ... a[15]
a[16], a[17], ... a[31]
...

Half-word-size 2 parallel tables

a[0], a[1], ... a[7]	b[0], b[1], ... b[7]
a[8], a[9], ... a[15]	b[8], b[9], ... b[15]
...	...

Half-word-size 4 parallel tables

a[0], a[1], ... a[3]	b[0], b[1], ... b[3]	c[0], c[1], ... c[3]	d[0], d[1], ... d[3]
a[4], a[5], ... a[7]	b[4], b[5], ... b[7]	c[4], c[5], ... c[7]	d[4], d[5], ... d[7]

Half-word-size 8 parallel tables

a[0]... a[1]	b[0]... b[1]	c[0]... c[1]	d[0]... d[1]	e[0]... e[1]	f[0]... f[1]	g[0]... g[1]	h[0]... h[1]
a[2]... a[3]	b[2]... b[3]	c[2]... c[3]	d[2]... d[3]	e[2]... e[3]	f[2]... f[3]	g[2]... g[3]	h[2]... h[3]
...	...	...	...	...	...	...	...

Word-size single table

a[0], a[1], ... a[7]
a[8], a[9], ... a[15]
...

Word-size 2 parallel tables

a[0], a[1], ... a[3]	b[0], b[1], ... b[3]
a[4], a[5], ... a[7]	b[4], b[5], ... b[7]
...	...

Word-size 4 parallel tables

a[0]... a[1]	b[0]... b[1]	c[0]... c[1]	d[0]... d[1]
a[2]... a[3]	b[2]... b[3]	c[2]... c[3]	d[2]... d[3]
...	...	...	...

Word-size 8 parallel tables

a[0]	b[0]	c[0]	d[0]	e[0]	f[0]	g[0]	h[0]
a[1]	b[1]	c[1]	d[1]	e[1]	f[1]	g[1]	h[1]
...	...	...	...	...	...	...	...

Figure 8-67. Lookup Table Organization for Various Entry Size and Parallel Tables (NWAY = 8)

8.3.5.5.8 Histogram Operation

Histogram functionality includes normal histogram, in which the addressed bin entry is incremented by 1, and weighted histogram, in which the addressed bin entry is incremented by an element in the weight array input.

Histogram operation is indicated by starting the vector command with:

VLOOP HIST, *CL#* **:** *cmd_len* **,** *PL#* **:** *param_len*

The histogram operation uses a subset of the available resources and instructions:
- VAGEN (address generator): fixed A0 for data, optional A1 for histogram input/output pointer, A2 for weights
- VINIT (register initialization) for normal histogram, providing a fixed increment value (normally 1)
- VLD (load): one load for data, and in the case of weighted histogram, one additional load for weights
- VHLD (histogram load): load histogram bins
- VADD (add) to increment histogram bins
- VHST (histogram store): store histogram bins back to memory
- Full set of 4 levels for looping

For normal histogram, the increment value is initialized via a VINIT instruction on V4:

```
        VINITHU_ONCE   P1, V4
```

Data is loaded with normal load into V2, for example,

```
        VLDBU_NPT      data_base[A0], V2
```

Distribution options of VLD are {1PT, NPT, DS2, US2}. 1PT works for single histogram configuration, NPT works for 2, 4, or 8 parallel histograms.

Optional weights (for weighted histogram) are loaded with another normal load into V4, for example,

```
        VLDBU_NPT      weight_base[A1], V4
```

A special histogram load, VHLD, is used to read in the bins:

VHLD *type* **_** *m* **HIST** *hist_base*[*hist_agen*][**V2**], **V0, RND_SAT:** *rnd_sat*

The bins are then incremented with normal VADD.

```
        VADD  V0, V4, V0
```

Results are stored back to the histogram memory with a special histogram store, VHST:

VHST *type* **_** *m* **HIST V0,** *hist_base*[*hist_agen*][**V2**]

Base and agen are supposed to be consistent between VHLD and VHST; otherwise, the outcome is undetermined.

Both signed and unsigned types are supported in data load, weight load, histogram load/store, to support unsigned histogram as well as weighted histogram with positive and negative weights.

Parameters for histogram are extracted from the instructions:
- Input data type is from VLD V2
- Input data distribution is from VLD V2, with SIMD units 0..num_par_hist-1 used for indexing histogram. NPT, 1PT, DS2, US2 distribution options are supported.
- Optional weight type is from VLD V4.

- Weight distribution behaves like data distribution, i.e., SIMD units 0..num_par_hist-1 used for incrementing parallel histogram bins
- Histogram data type is from VHLD and VHST, and the two instructions must be consistent in the data type, otherwise the outcome is indeterminate.
- rnd_sat in VHLD specifies round/truncate mode and how many bits to round off data before indexing the histogram, and also saturation mode and bound. This field is 5-bit (like VST and VTLD), so only P0..P31 is allowed.

The histogram command basically takes the data array, and processes num_par_hist data points at a time by assuming that many sets of parallel histogram bin storage. Each data value is rounded, saturated to generate a bin index. The address bin storage elements (which can be byte, halfword, or word) is then incremented by 1 or the weight, saturated to min/max values of that type, signed/unsigned byte/halfword/word:

- For unsigned byte, bin data is saturated to [0, 255].
- For signed byte, bin data is saturated to [-128, 127].
- For unsigned halfword, bin data is saturated to [0, 65535].
- For signed halfword, bin data is saturated to [-32768, 32767].
- For unsigned word, bin data is saturated to [0, 0xFFFF FFFF].
- For signed word, bin data is saturated to [-0x8000 0000, 0x7FFF FFFF].

There are restrictions on the weight data type (as indicated on the VLD into V4):

- Its signed/unsigned designation must agree with the histogram bin (as indicated in VHLD, VHST).
- Its size is limited to byte and halfword, and must be no bigger size then the histogram bin. Thus, for byte sized bins, only byte-sized weights are possible. For halfword or word sized bins, byte or halfword sized weights are possible.

C representation of histogram processing for a single histogram is as follows.

```
        for (i=0; i<num_data; i++)
        {
            data = data_base[i];
            bin = saturate(round(data));
            hist[bin]++;
        }
```

Weighted histogram has the following C representation:

```
        for (i=0; i<num_data; i++)
        {
            data = data_base[i];
            bin = saturate(round(data));
            hist[bin] += weight[i];
        }
```

Histogram organization in memory depends on histogram type and num_par_hist. The N physical banks of 32-bit memory is partitioned into num_par_hist logical banks, and each parallel histogram is organized inside its own bank, as with parallel lookup tables (see Figure 8-67).

The histogram command does not pre-initialize the histogram bin storage. The histogram command also does not sum up the parallel sets of storage into a single set; this is performed as a post-processing step.

Histogram operation takes 2M/P cycles plus some per-command overhead to process M data items with P parallel histograms (P = 1, 2, 4, or 8). As P parallel histograms takes P times the storage of a single histogram, it is a tradeoff of memory size and performance.

An example of histogram operation is:

```
; single histogram, byte data, short histogram, round down 2 bits, 33 bins
; rounding mode, shift, saturation bound coded in P7 using scalar instructions
   VLOOP             HIST, CL#: cmd_len, PL#: param_len
   VINITHU_ONCE      P1, V4                             ; initialize V4 = 1
   VLDBU_NPT         data_base[A0], V2                  ; load data to V2
```

```
VHLDHU_1HIST     hist_base[A1][V2], V0, RND_SAT: P7 ; load hist entry into V0
VADD             V0, V4, V0                         ; increment entry
VHSTH_1HIST      V0, hist_base[A1][V2]              ; store histogram entry
```

An example of weighted histogram operation is:

```
; single weighted histogram, byte data, byte weight, short histogram,
; round down 2 bits, 33 bins
; rounding mode, shift, saturation bound coded in P7 using scalar instructions
   VLOOP           HIST, CL#: cmd_len, PL#: param_len
   VLDBU_NPT       data_base[A0], V2                  ; load data to V2
   VLDBU_NPT       weight_base[A2], V4                ; load weights to V4
   VHLDHU_1HIST    hist_base[A1][V2], V0, RND_SAT: P7 ; load hist entry into V0
   VADD            V0, V4, V0                         ; increment entry by weights
   VHSTH_1HIST     V0, hist_base[A1][V2]              ; store histogram entry
```

Table lookup and histogram loops are limited to use designated vector register and address generator resources as shown in Table 8-358.

Table 8-358. Table Lookup and Histogram Hardware Resources

Loop type	Input/outputstream	Address generator	Vector register
table lookup	data	A0	V2
	table	A1	V0
	output	A2	V0
load with expansion	flag	A0	V2
	data	n/a	V0
	output	A2	V0
histogram	data	A0	V2
	histogram	A1	V0
	weight	A2	V4

8.3.5.5.9 Circular Buffer Addressing Support

In certain applications, it is desirable to maintain circular buffers for input data and intermediate results. Thus, circular buffer addressing is supported in VCOP.

- Support circular buffer in VLD and VST instructions in compute loops, table lookup loops, and histogram loops. No circular buffer in VTLD (table load) and VHLD/VHST (histogram bin load/store).
- Support is in pointer increment arithmetic, so no load or store in any iteration should straddle the circular buffer boundary.
- Support only 2's power buffer size from 1Kbytes, aligned to the size. For example, 1KB circular buffer can only start/end at 1KB alignment of the memory map.
- Support is in all memory regions.

Circular buffer addressing option is encoded in the circ_buf field taking up bits 23..20 of base address:

- 0: no circular buffer addressing
- 1: circular buffer size 1K bytes
- 2: circular buffer size 2K bytes
- 3: circular buffer size 4K bytes
- 4: circular buffer size 8K bytes
- 5: circular buffer size 16K bytes
- 6: circular buffer size 32K bytes
- 7~15: reserved

The base address does not need to be aligned to the circular buffer size, in the abive example's case, 0x400. To avoid any single load/store straddle the circular buffer boundary, the base address and the pointer increments must all be aligned to the access size of each iteration.

For example, when loading N data points of halfword data, where N = 8. Access size = 16, so the base address and any pointer increment must be aligned to multiples of 16.

Hardware adjusts the memory address for circular buffering by this process:

```
circ_buf = (base >> 20) & 0xF;          // extract circ_buf
mask = (0x200 << circ_buf) - 1;         // mask = size of buffer - 1
cbuf = (base & 0xFFFFF) & ~mask;        // find base addr of buffer
addr = cbuf + (base + agen) & mask;     // add offset from base of buffer
```

For example, base = 0x10420, agen = 0xFF0.

1. Extract circ_buf field by cir_buf = (base >> 20) & 0xF = 1, indicating 1KB circular buffer size.
2. Set mask = (0x200 << 1) – 1 = 0x3FF, forming the bit mask that defines the circular addressing bits versus linear, non-circular bits.
3. Get rid of the circ_buf field by ANDing base with 0xFFFFF, then zero out the circular addressing bits by ANDing with the complement of mask, to get cbuf = 0x400, so the circular buffer is 0x400 ~ 0x7FF.
4. Next, add the address generator value to the base address, and AND with mask to find offset from the cbuf base. (base + agen) & mask = (0x10420 + 0xFF0) & 0x3FF = 0x11410 & 0x3FF = 0x10.
5. Finally, add the offset 0x10 to the circular buffer base 0x400 to get the final address, 0x410.

The agen is allowed to go over the circular buffer many times over to wrap back inside, and that agen can even go backward to wrap from the beginning to the end of the buffer.

8.3.5.5.10 Load/Store Address Alignment Constraints

Load/store pointer alignment constraints are summarized in Table 8-359, for data load/store as well as table and histogram load/store, for various data types.

Table 8-359. Load and Store Address Alignment Constraints

Type of load/store and distribution	Byte type	halfword type	Word type
VLD_NPT	any	any	word-aligned
VLD_1PT	any	any	any
VLD_CIRC2	any	any	any
VLD_DS2	any	halfword-aligned	not supported
VLD_US2	any	any	any
VLD_DINTRLV	any	word-aligned	not supported
VLD_CUST_Pi	any	any	word-aligned
VLD_EXP	any	halfword-aligned	word-aligned
VST_NPT	any	halfword-aligned	word-aligned
VST_1PT	any	halfword-aligned	word-aligned
VST_DS2	any	halfword-aligned	word-aligned
VST_INTRLV	any	word-aligned	not supported
VST_OFFST_NP1	any	halfword-aligned	word-aligned
VST_COLLAT	any	halfword-aligned	word-aligned
VST_DDA	any	halfword-aligned	word-aligned
VST_SKIP	any	halfword-aligned	not supported
VTLD_1TBL_1PT	any	halfword-aligned	word-aligned
VTLD_mTBL_nPT (other than 1TBL_1PT)	32-byte-aligned	32-byte-aligned	32-byte-aligned
VHLD_mHIST	32-byte-aligned	32-byte-aligned	32-byte-aligned
VHST_mHIST	32-byte-aligned	32-byte-aligned	32-byte-aligned

The rationale is:

- For data load, in general any alignment is supported. Exception: When software loads 32 bytes, the pointer needs to be word-aligned.
- For data store, in general any alignment is supported for byte, halfword alignment for halfword, and word-alignment for word. Exception: When 32 or 64 bytes is stored, the value needs to be word-aligned. In case of LDH_DS2, the pointer must be 16-bit aligned.
- For table load with one table, 1-point lookup, the table base pointer can be of any alignment for byte, halfword alignment for halfword, and word-alignment for word.
- For table load with other configurations, only a 32-byte aligned table base is supported.
- Table store is really data store with 1, 2, 4, or 8 data points. The same alignment is supported as a data store, i.e., any alignment for byte, halfword alignment for halfword, and word-alignment for word.
- For histogram load/store, the histogram base pointer must be 32-byte aligned.

8.3.5.6 Load/Store Buffer and Scheduling

There is load data buffer associated with each VLD instruction in the loop to reduce memory reads and resulting contention and slow-down.

- Size of the buffer is 32 bytes, and is filled with 8 32-bit words on word alignment.
- When the requested data is not contained in the buffer (completely or partially), VCOP reads another 8 32-bit words from the first requested word.
- There is no sharing of data among separate load data buffers.

There is one common store data buffer, basically a shadow copy of all vector registers, for all the VST instructions in the loop to skew memory write timing to reduce memory contentions and resulting slow-down.

- The store data buffer is 16 × 8 × 40-bit in size.
- Vector registers are copied to the store data buffer at the end of i4 iteration, when there is any store to be performed for that iteration. In case stores are not all completed then, load and execution stages are stalled, until all stores are completed.
- Write data from multiple iterations of the same store instruction are not combined. For example, STB_NPT_ALWS causes a memory store of 8 bytes in each iteration.
- Write data from multiple stores of the loop, same iteration, are not combined.

Memory loads and stores are arbitrated for each memory port. There are three memory ports:

- WBUF: Dedicated to WBUF
- IBUFL: For IBULA and IBUFLB
- IBUFH: For IBUFHA and IBUFHB

Load store priorities are: force store > load > store. Force store is issued when the load and operation stages are stalled to free up the store stage so the hardware can advance to the next iteration. Other than force store, load has priority over normal store.

Example 8-9. 3-Tap Horizontal Filtering, Byte Type

```
      VLDB_DINTRLV  Pdata[a0], V0                 ; data, starts from 0x10000 (IBUFLA)
      VLDB_1PT      Pcoef[a1], V2                 ; coef, starts from 0x0 (WBUF)
      VMADD         V0, V2, V4, V4
||    VMADD         V1, V2, V5, V5
      VSTB_INTRLV   V4, Pout[a2], RND_SAT: rnd_sat_param ; output, start from 0x10400 (IBUFLA)
```

8.3.5.6.1

Table 8-360. Load and Store Buffering Example, Byte-Type Horizontal Filter

Cycle	LD stage	OP stage	ST stage
0	iter 0 LD0 need 0x10000..0x1000F, **read IBUFLA** LD0_buf = {0x10000..0x1001F} LD2 need 0x0, **read WBUF** LD2_buf = {0x0..0x1F}		
1	iter 1 LD0 need 0x10001..0x10010, from LD0_buf LD2 need 0x1, from LD2_buf	iter 0 MADD || MADD	
2	iter 2 LD0 need 0x10002..0x10011, from LD0_buf LD2 need 0x2, from LD2_buf	iter 1 MADD || MADD	iter 0
3	iter 3 LD0 need 0x10010..0x1001F, from LD0_buf LD2 need 0x0, from LD2_buf	iter 2 MADD || MADD	iter 1
4	iter 4 LD0 need 0x10011..0x10020, **read IBUFLA** LD0_buf = {0x10010..0x1002F} LD2 need 0x1, from LD2_buf	iter 3 MADD || MADD	iter 2 ST0 store 0x10400..0x1040F queue the store

Table 8-360. Load and Store Buffering Example, Byte-Type Horizontal Filter (continued)

Cycle	LD stage	OP stage	ST stage
5	iter 5 LD0 need 0x10012..0x10021, from LD0_buf LD2 need 0x2, from LD2_buf	iter 4 MADD || MADD	iter 3 store IBUFLA
6	iter 6 LD0 need 0x10020..0x1002F, from LD0_buf LD2 need 0x0, from LD2_buf	iter 5 MADD || MADD	iter 4
7	iter 7 LD0 need 0x10021..0x10030, **read IBUFLA** LD0_buf = {0x10020..0x1003F} LD2 need 0x1, from LD2_buf	iter 6 MADD || MADD	iter 5 ST0 store 0x10410..0x1041F queue the store

Here LDB_DINTRLV is used to read 16 data points, to operate on 16 outputs in parallel. The hardware issues memory reads for iteration 0, 4, 7, etc, is able to supply data for subsequent hold data read in for subsequent LD; on average there is 1 read every 3 cycles. For coefficient, since only 3 bytes is used, just one read in iteration 0 is sufficient to supply data for the whole loop. Output is produced in iteration 2, 5, etc, and the store buffer delays the write until there is no read traction, avoiding read/write contention and slowdown.

Due to memory read/write timing, delayed slot in operation, and round/saturate feature in the store stage, the skew between load, operation, and store stages is more than what is shown in Table 8-360, but for calculating steady-state performance, simplified analysis like this is sufficient.

Example 8-10. Horizontal Filtering, Short Type

```
        VLDH_NPT      Pdata0[a0], V0                ; data, start from 0x10000 (IBUFLA)
        VLDH_NPT      Pdata1[a0], V2                ; data next row, start from 0x10080 (IBUFLA)
        VLDH_1PT      Pcoef[a1], V4                 ; coef, start from 0x0 (WBUF)
        VMADD         V0, V4, V6, V6
||      VMADD         V2, V4, V7, V7
        VSTH_NPT      V6, Pout0[a2], RND_SAT: rnd_sat_param ; start from 0x10400 (IBUFLA)
        VSTH_NPT      V7, Pout1[a2], RND_SAT: rnd_sat_param ; start from 0x10480 (IBUFLA)
```

8.3.5.6.2

Software cannot issue VLDH_DINTRLV on odd halfwords, so use two LDH_NPT.

Table 8-361. Load and Store Buffering Example, Short-Type Horizontal Filter

Cycle	LD stage	OP stage	ST stage
0	iter 0 LD0 need 0x10000..0x1000F, **read IBUFLA** LD0_buf = {0x10000..0x1001F} LD1 need 0x10080 .. 0x1008F, **stalled** LD2 need 0..1, **read WBUF** LD2_buf = {0x0..0x1F}		
1	iter 0 LD1 **read IBUFLA** LD1_buf = {0x10080..0x1009F}		
2	iter 1 LD0 need 0x10002..0x10011, from LD0_buf LD1 need 0x10082..0x10091, from LD1_buf LD2 need 2..3, from LD2_buf	iter 0 MADD || MADD	
3	iter 2 LD0 need 0x10004..0x10013, from LD0_buf LD1 need 0x10084..0x10093, from LD1_buf LD2 need 4..5, from LD2_buf	iter 1 MADD || MADD	iter 0
4	iter 3 LD0 need 0x10010..0x1001F, from LD0_buf LD1 need 0x10090 .. 0x1009F, from LD1_buf LD2 need 0..1, from LD2_buf	iter 2 MADD || MADD	iter 1

Table 8-361. Load and Store Buffering Example, Short-Type Horizontal Filter (continued)

Cycle	LD stage	OP stage	ST stage
5	iter 4 LD0 need 0x10012..0x10021, **read IBUFLA** LD0_buf = {0x10010..0x1002F} LD1 need 0x10092 .. 0x100A1, **stall** LD2 need 2..3, from LD2_buf	iter 3 MADD || MADD	iter 2 ST0 store 0x10400..0x1040F ST0 queued ST1 store 0x10480..0x1048F ST1 queued
6	iter 4 LD1 **read IBUFLA** LD1_buf = {0x10090..0x100AF}	stalled	stalled
7	iter 5 LD0 need 0x10014..0x10023, from LD0_buf LD1 need 0x10094..0x100A3, from LD1_buf LD2 need 4..5, from LD2_buf	iter 4 MADD || MADD	iter 3 ST0 **write IBUFLA**
8	iter 6 LD0 need 0x10020..0x1002F, from LD0_buf LD1 need 0x100A0..0x100AF, from LD1_buf LD2 need 0..1, from LD2_buf	iter 5 MADD || MADD	iter 4 ST1 **write IBUFLA**
9	iter 7 LD0 need 0x10022..0x10031, **read IBUFLA** LD0_buf = {0x20..x3F} LD1 need 0x100A2..0x100B1, **stall** LD2 need 2..3, from LD2_buf	iter 6	iter 5 ST0 store 0x10410..0x1041F ST0 queuedST1 store 0x10490..0x1049F ST1 queued
10	iter 7 LD1 **read IBUFLA** LD1_buf = {0x100A0..0x100BF}	stalled	stalled

In this loop, with 2 VLDH_NPT to sustain 16 multiply-accumulates per iteration, the load stage stalled due to memory contention in IBUFL from the 2 loads. The read buffer supplies data for 2 subsequent iterations in steady state, leaving 2 memory-read free cycles every 3 iterations. With output writing back to IBUFL, the store buffer effectively delays the memory writes to use these free memory slots. Each i4 loop of 3 iterations thus takes 4 cycles to complete.

8.3.5.7 VCOP Per-Loop Overhead

For each vector command specifying a loop, several overhead components are present. In general, developers are encouraged to run place into each loop as much processing as possible and run as many iterations as possible to reduce the percentage of time spent on overhead.

Overhead components are:

- Command decode time: it takes 2 cycles per instruction to decode, since scalar core is providing instructions at this rate. Command decode time can be hidden if ARP32 feeds these instructions while VCOP is executing a previous command.
- Parameter fetch time: it takes 9 + ceiling(num_param/16) cycles to fetch parameter from data memory (WBUF, IBUFL, or IBUFH).
- Execution pipeline ramp up/down time: it takes time to ramp up and down the long pipeline of VCOP. Minimal time for any loop is execution is about 17 cycles, due to the pipeline ramp up/down time.

Figure 8-68 shows a decode-parameter-fetch-execution time line.

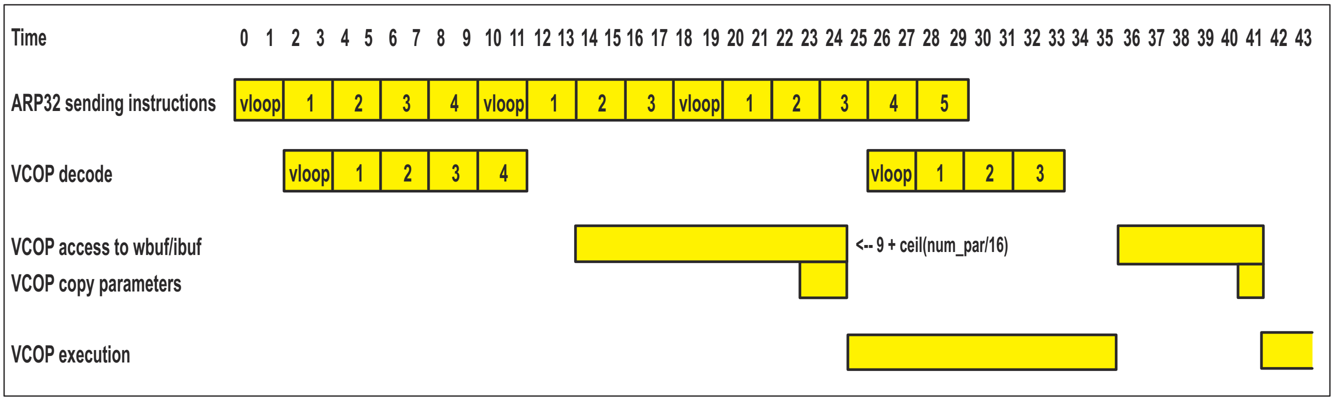


Figure 8-68. Example of Operation Delay Slots

- t = 0, ARP32 sends VLOOP, followed by VCOP instructions in the loop, instructions go into FIFO.
- t = 2, VCOP pulls instruction out of FIFO, recognizes VLOOP.
- t = 10 (depend on command length), VCOP pulls last instruction of loop from FIFO.
- t = 10, ARP32 sends next command, starting from VLOOP.
- t = 14, VCOP starts access to parameter memory, 16 parameters/cycle.
- t = 18, ARP32 keeps sending vector commands, until program stream switches to scalar code, or VCOP pre-decode instruction FIFO becomes full (capacity is 64 words).
- t = 23 (14 + 9), VCOP starts to copy parameters into decoded command buffer.
- t = 25 (after parameter copy), VCOP starts executing the loop.
- t = 26 (sync to 250-MHz clock), VCOP pulls instruction out of FIFO, recognizes VLOOP.
- t = 32 (depend on command length), VCOP pulls last instruction of loop from FIFO.
- t = 36 (after prev loop finishes executing), VCOP starts to access parameter memory.

This is for showing the dependency among activities; execution time here is not realistic; it takes minimally 17 cycles to execute any loop, due to the pipeline depth.

8.3.5.8 VCOP Error Handling

Upon detection of erroneous instruction code, parameter, load/store address, or loop in execution exceeding programmed max iteration count, VCOP signals the error by generating an interrupt and registering the error source in the MMR VCOP_ERROR. Refer to register manual for details.

Upon detection of error, VCOP:

1. Terminates current loop, leaving vloop_ptr, param_ptr, loop variables i0, i1, i2, i3, i4 in MMR.
2. Parses and clears vector instructions until encountering VWDONE.
3. Issues interrupt, assert vec_done = 1, deassert vec_rdy = 0 to block subsequent vector instructions.
4. Upon VCOP_ERROR status being cleared (by writing 1s to clear bits 10:0), VCOP becomes idle (assert VCOP_STATUS[2]VEC_RDY = 1) and ready to resume processing.
5. Alternatively, VCOP can be reset to resume processing.

Depending on the EVE-level memory switch setting, VCOP's access of the data memory space may be deemed invalid by the memory switch. When this happens, the memory switch signals VCOP to enter error handling state, as described above. VCOP does not send an error interrupt, as the memory switch is doing that (for invalid access from VCOP or from other masters).

8.3.5.9 Vector Operation Details

8.3.5.9.1 VABS

Assembly syntax:	**VABS** *src1* **,** *dst*
Operation:	Absolute value
Classification:	1-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = abs(src1);

8.3.5.9.2 VABSDIF

Assembly syntax:	**VABSDIF** *src1* **,** *src2* **,** *dst*
Operation:	absolute-difference
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = abs(src1 – src2);

8.3.5.9.3 VADD

Assembly syntax:	**VADD** *src1* **,** *src2* **,** *dst*
Operation:	addition
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 + src2;

8.3.5.9.4 VADDH

Assembly syntax:	**VADDH** *src1* **,** *src2* **,** *dst*
Operation:	Add high
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 + (src2 >> 32);

8.3.5.9.5 VADDSUB

Assembly syntax:	VADDSUB src1/dst1, src2/dst2
Operation:	Add-subtract
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst1 = src1 + src2; dst2 = src1 – src2;

8.3.5.9.6 VADD3

Assembly syntax:	**VADD3** *src1* **,** *src2* **,** *src3* **,** *dst* (dst same as src3)
Operation:	Add 3 items,
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst = src1 + src2 + src3;

8.3.5.9.7 VADIF3

Assembly syntax:	**VADIF3** *src1* **,** *src2* **,** *src3* **,** *dst* (dst same as src3)
Operation:	Add difference
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst = src1 – src2 + src3;

8.3.5.9.8 VAND

Assembly syntax:	**VAND** *src1* **,** *src2* **,** *dst*
Operation:	Bitwise and
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 & src2;

8.3.5.9.9 VANDN

Assembly syntax:	**VANDN** *src1* **,** *src2* **,** *dst*
Operation:	Bitwise and-not
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 & ~src2;

8.3.5.9.10 VAND3

Assembly syntax:	**VAND3** *src1* **,** *src2* **,** *src3* **,** *dst* (dst same as src3)
Operation:	Bitwise-and 3 items
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst3 = src1 & src2 & src3;

Using VAND3 accelerates a morphological erosion operation.

8.3.5.9.11 VBINLOG

Assembly syntax:	**VBINLOG** *src1* **,** *dst*
Operation:	Binary log
Classification:	1-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

VBINLOG computes an approximate binary log, by detecting the left-most 1 bit in the input, encoding the bit position in bits 31..28, and left-justifying the following input bits in bits 27..0. The detected position of bit 31 is encoded as 15 (bit position minus 16), bit 30 as 14, and so on, and bit 17 as 1. If the left-most 1 bit is not found in bits 31..17, 0 is coded.

The VBINLOG operation is a non-standard floating point representation. Say a 32-bit integer x = 2^exp * (1 + frac/(2^28)), where exp and frac are integers. If (exp >= 16), binlog(x) = (exp – 16) * (2^28) + frac, concatenating the 4-bit adjusted exponent, exp – 16, with the 28-bit fraction. Otherwise, the representation is subnormal, leaving 0 in the exponent and bits 16..0 of input left-justified in the fraction part of output, bits 27..0.

For example, the number 0x0700_0000 has leading 1 in bit 26, so the bit position is coded as 26 – 16 = 10 = 0xA. The subsequent bits, two 1s immediately following the leading 1, left-justified in 28 bits or 7 hex digits, is 0xC00_0000. Thus the VBINLOG outcome of the number is 0xAC00_0000.

For a subnormal example, the number 0x8000 < 2^16 (is half of that), so its VBINLOG outcome is 0x0400_0000.

input																		binlog output							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	...	31	30	29	28	27	26	...	0
1	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	1	1	b30	b29	...	b3
0	1	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	1	0	b29	b28	...	b2
0	0	1	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	0	1	b28	b27	...	b1
0	0	0	1	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	0	0	b27	b26	...	b0
0	0	0	0	1	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	1	1	b26	b25	...	0
0	0	0	0	0	1	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	1	0	b25	b24	...	0
0	0	0	0	0	0	1	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	0	1	b24	b23	...	0
0	0	0	0	0	0	0	1	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	0	0	b23	b22	...	0
0	0	0	0	0	0	0	0	1	b22	b21	b20	b19	b18	b17	b16	b15	...	0	1	1	1	b22	b21	...	0
0	0	0	0	0	0	0	0	0	1	b21	b20	b19	b18	b17	b16	b15	...	0	1	1	0	b21	b20	...	0
0	0	0	0	0	0	0	0	0	0	1	b20	b19	b18	b17	b16	b15	...	0	1	0	1	b20	b19	...	0
0	0	0	0	0	0	0	0	0	0	0	1	b19	b18	b17	b16	b15	...	0	1	0	0	b19	b18	...	0
0	0	0	0	0	0	0	0	0	0	0	0	1	b18	b17	b16	b15	...	0	0	1	1	b18	b17	...	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	b17	b16	b15	...	0	0	1	0	b17	b16	...	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	b16	b15	...	0	0	0	1	b16	b15	...	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b16	b15	...	0	0	0	0	b16	b15	...	0

Figure 8-69. Binlog Function

Input of VBINLOG is 32-bit, and so is the outcome, not 40-bit. VLMBD supports full 40-bit input range leading bit detection. VBINLOG is useful in compressing dynamic range and preserving precision. VBINLOG followed by table lookup can be used to produce the reciprocal of a number.

8.3.5.9.12 VBITC

Assembly syntax:	**VBIT** *src1* **,** *dst*
Operation:	Bit count
Classification:	1-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

Count number of “1” bits in src1[31:0].

8.3.5.9.13 VBITDI

Assembly syntax:	**VBITDI** *src1* **,** *dst1* **,** *dst2*
Operation:	Bit deinterleave
Classification:	1-input 2-output
Bit width:	32-bit
Delay slot:	0
C statement:	As follows

Bit deinterleave is basically the inverse of bit interleave (VBITI).

For example,

```
        src1 = 0x523(0000_0000_0000_0000_0000_1001_0010_0011)
```

Results in:

```
        dst1 = 0x25 (0000_0000_0010_0101),
        dst2 = 0x11(0000_0000_0001_0001)
```

8.3.5.9.14 VBITI

Assembly syntax:	**VBITI** *src1* **,** *src2* **,** *dst*
Operation:	Bit interleave
Classification:	2-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

Bit interleave 16 LSBs of src1 and src2 (each), and write to dst.

For example,

```
        src1 = 0x25 (0000_0000_0010_0101)
        src2 = 0x11(0000_0000_0001_0001)
```

Results in:

```
        dst = 0x523(0000_0000_0000_0000_0000_1001_0010_0011)
```

8.3.5.9.15 VBITPK

Assembly syntax:	**VBITPK** *src1* **,** *src2* **,** *dst*
Operation:	Bit-pack
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	one
C statement:	As follows

Pack boolean (src1[i] >= src2[i]) into bits in dst[0], and broadcast dst[0] to all SIMD lanes so that dst[j] = dst[0], 1 <= j <= NSIMD.

VBITPK is useful in accessing bit-packed data (binary images).

8.3.5.9.16 VBITR

Assembly syntax:	**VBITR** *src1* **,** *dst*
Operation:	Bit reverse
Classification:	1-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

Reverse bits in src1[31:0] and write to dst.

8.3.5.9.17 VBITTR

Assembly syntax:	**VBITTR** *src1* **,** *dst*
Operation:	Bit transpose
Classification:	1-input 1-output
Bit width:	NSIMD-bit
Delay slot:	one

VBITTR, bit transpose, also called corner turn, takes N × N bits in the source, from N LSBs of each of the N registers, and transpose them. Here is an example:

Input:

Lane	Value	Bit position 0 1 2 3
0	1	1 0 0 0
1	2	0 1 0 0
2	3	1 1 0 0
3	4	0 0 1 0

Output:

Lane	Value	Bit position 0 1 2 3
0	5	1 0 1 0
1	6	0 1 1 0
2	8	0 0 0 1
3	0	0 0 0 0

8.3.5.9.18 VBITUNPK

Assembly syntax:	**VBITUNPK** *src1* **,** *src2* **,** *dst*
Operation:	Bit-unpack
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	one
C statement:	As follows

Unpack bits in src1[0]. Set dst[i] = src2[i] if the unpacked bit to lane i is 1, otherwise dst[i] = 0.

8.3.5.9.19 VCMOV

Assembly syntax:	**VCMOV** *cond* **,** *src1*[**,** *src2*|*dst*]
Operation:	Conditional move
Classification:	special
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

cond = {I1234_ZERO, I234_ZERO, I34_ZERO, LAST_I4_ZERO, LAST_I34, LAST_I234, LAST_I1234}.

Move src1 to dst in the iterations designated by cond. This is parallel move within each SIMD lane; dst[i] = src1[i] when condition is true. Otherwise, dst[i] = src2[i]. By always writing to the dst register, dst register does not need to be {V0, V1, V2, V3} (see Section 8.3.5.5.6 for details on the store-without-being destination register constraint).

8.3.5.9.20 VCMPEQ

Assembly syntax:	**VCMPEQ** *src1* **,** *src2* **,** *dst*
Operation:	Compare equal
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 == src2) ? 1 : 0;

8.3.5.9.21 VCMPGE

Assembly syntax:	**VCMPGE** *src1* **,** *src2* **,** *dst*
Operation:	Compare greater-than-or-equal
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 >= src2) ? 1 : 0;

8.3.5.9.22 VCMPGT

Assembly syntax:	**VCMPGT** *src1* **,** *src2* **,** *dst*
Operation:	Compare greater-than
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 > src2) ? 1 : 0;

8.3.5.9.23 VDINTRLV

Assembly syntax:	**VDINTRLV** {*src1*|*dst1*}, {*src2*|*dst2*}
Operation:	Deinterleave
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VDINTRLV is basically the inverse of VINTRLV.

For 8-way SIMD, this is used:

```
        dst1[0] = src1[0];
        dst2[0] = src1[1];
        dst1[1] = src1[2];
        dst2[1] = src1[3];
        dst1[2] = src1[4];
        dst2[2] = src1[5];
        dst1[3] = src1[6];
        dst2[3] = src1[7];
        dst1[4] = src2[0];
        dst2[4] = src2[1];
        dst1[5] = src2[2];
        dst2[5] = src2[3];
        dst1[6] = src2[4];
        dst2[6] = src2[5];
        dst1[7] = src2[6];
        dst2[7] = src2[7];
```

8.3.5.9.24 VDINTRLV2

Assembly syntax:	**VDINTRLV2** {*src1*|*dst1*}, {*src2*|*dst2*}
Operation:	Deinterleave with 2-element frequency
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VDINTRLV2 is basically the inverse of VINTRLV2.

For 8-way SIMD, this is used:

```
dst1[0] = src1[0];
dst1[1] = src1[1];
dst2[0] = src1[2];
dst2[1] = src1[3];
dst1[2] = src1[4];
dst1[3] = src1[5];
dst2[2] = src1[6];
dst2[3] = src1[7];
dst1[4] = src2[0];
dst1[5] = src2[1];
dst2[4] = src2[2];
dst2[5] = src2[3];
dst1[6] = src2[4];
dst1[7] = src2[5];
dst2[6] = src2[6];
dst2[7] = src2[7];
```

8.3.5.9.25 VEXITNZ

Assembly syntax:	**VEXITNZ** *level* **,** *src1*
Operation:	Exit upon non-zero
Classification:	special
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VEXITNZ is for early termination of VLOOP or in the repeat-loop level. The designated src1 register's 0th way is examined every time the instruction executes, and if it is not zero, the hardware terminates either at VLOOP level or at VRPT level, as indicated by the level parameter.

Parameter src1 is examined at the time the instruction executes, and an exit/no-exit decision is made. However, any exit/no-exit action is not taken immediately. For VEXITNZ VLOOP, subsequent instructions in the same iteration, remaining operations and all the stores, are executed, then the loop terminates. For VEXITNZ VRPT; all subsequent instructions in the same loop are executed, including the rest of the iteration, and all remaining iterations until i1, i2, i3, i4 all run to respective end counts, then the repeated loop terminates.

There can only be one VEXITNZ in each vector command; no mixing of VEXITNZ VLOOP and VEXITNZ VRPT. When the vector command is not repeated (RPT_END = 0), VEXITNZ VRPT behaves like VEXITNZ VLOOP.

8.3.5.9.26 VINTRLV

Assembly syntax:	**VINTRLV** {*src1*|*dst1*}, {*src2*|*dst2*}
Operation:	Interleave
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VINTRLV is basically interleaving among SIMD lanes.

For 8-way SIMD, this is used:

```
        dst1[0] = src1[0];
        dst1[1] = src2[0];
        dst1[2] = src1[1];
        dst1[3] = src2[1];
        dst1[4] = src1[2];
        dst1[5] = src2[2];
        dst1[6] = src1[3];
        dst1[7] = src2[3];
        dst2[0] = src1[4];
        dst2[1] = src2[4];
        dst2[2] = src1[5];
        dst2[3] = src2[5];
        dst2[4] = src1[6];
        dst2[5] = src2[6];
        dst2[6] = src1[7];
        dst2[7] = src2[7];
```

8.3.5.9.27 VINTRLV2

Assembly syntax:	**VINTRLV2** {*src1*|*dst1*}, {*src2*|*dst2*}
Operation:	Interleave with 2-element frequency
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VINTRLV is basically interleaving among SIMD lanes.

For 8-way SIMD, this is used:

```
        dst1[0] = src1[0];
        dst1[1] = src1[1];
        dst1[2] = src2[0];
        dst1[3] = src2[1];
        dst1[4] = src1[2];
        dst1[5] = src1[3];
        dst1[6] = src2[2];
        dst1[7] = src2[3];
        dst2[0] = src1[4];
        dst2[1] = src1[5];
        dst2[2] = src2[4];
        dst2[3] = src2[5];
        dst2[4] = src1[6];
        dst2[5] = src1[7];
        dst2[6] = src2[6];
        dst2[7] = src2[7];
```

8.3.5.9.28 VINTRLV4

Assembly syntax:	**VINTRLV4** {*src1*|*dst1*}, {*src2*|*dst2*}
Operation:	Interleave with 4-element frequency

Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VINTRLV is basically interleaving among SIMD lanes.

For 8-way SIMD, this is used:

```
        dst1[0] = src1[0];
        dst1[1] = src1[1];
        dst1[2] = src1[2];
        dst1[3] = src1[3];
        dst1[4] = src2[0];
        dst1[5] = src2[1];
        dst1[6] = src2[2];
        dst1[7] = src2[3];
        dst2[0] = src1[4];
        dst2[1] = src1[5];
        dst2[2] = src1[6];
        dst2[3] = src1[7];
        dst2[4] = src2[4];
        dst2[5] = src2[5];
        dst2[6] = src2[6];
        dst2[7] = src2[7];
```

8.3.5.9.29 VLMBD

Assembly syntax:	**VLMBD** *src1* **,** *src2* **,** *dst*
Operation:	Left-most bit detect
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	As follows

Depending on src2 being zero or not, VLMBD searches for 0 or 1 in src1, starting from bit 39. Bit position where the first 0 or 1 occurs is returned, and -1 is returned if it’s not found.

8.3.5.9.30 VMADD

Assembly syntax:	**VMADD** *src1* **,** *src2* **,** *src3* **,** *dst* **, RND:** *rnd_param* (dst same as src3)
Operation:	Multiply-add
Classification:	3-input 1-output
Bit width:	17-bit src1/src2, 40-bit src3/dst
Delay slot:	2 delay from src1/src2, 1 delay from src3
C statement:	dst = src3 + round_or_shift(src1 * src2, rnd_param)

The RND field specifies a parameter, in which additional fields are read to support built-in round/shift feature in the operation:

Figure 8-70. VMPY, VMADD and VMSUB Rounding Parameters

Parameter Indexed by rnd_param

15 ... 7	6 ... 5	4 ... 0
Reserved	rnd_mode	rnd_shift

Table 8-362. Parameter Indexed by rnd_param Field Descriptions

Bits	Name	Description	
15-7	Reserved		
6-5	rnd_mode	Sets rounding mode:	
		0	No rounding
		1	Round
		2	Truncate
		3	Left-shift
4-0	rnd_shift	Number of bits to round/shift, limited to:	
		{8|15|16}	Round/truncate
		1	Left-shift

When rnd_mode = round or truncate, src1 * src2 is rounded down before the addition or subtraction. When rnd_mode = left-shift, src1 * src2 is shifted up before the addition or subtraction. Note the restriction on the number of bits to shift.

8.3.5.9.31 VMAX

Assembly syntax:	**VMAX** *src1* **,** *src2* **,** *dst*
Operation:	Maximum
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	no
C statement:	dst = (src1 > src2) ? src1 : src2;

8.3.5.9.32 VMAXSETF

Assembly syntax:	**VMAXSETF** *src1* **,** {*src2*|*dst1*}**,** *dst2*
Operation:	Maximum and set flag
Classification:	2-input 2-output
Bit width:	33-bit
Delay slot:	no
C statement	dst1 = max(src1, src2); dst2 = (src1>src2) ? 1:0

8.3.5.9.33 VMIN

Assembly syntax:	**VMIN** *src1* **,** *src2* **,** *dst*
Operation:	Minimum
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	no
C statement:	dst = (src1 < src2) ? src1 : src2;

8.3.5.9.34 VMINSETF

Assembly syntax:	**VMINSETF** *src1* **,** {*src2*|*dst1*}**,** *dst2*
Operation:	Minimum and set flag
Classification:	2-input 2-output
Bit width:	33-bit
Delay slot:	
C statement:	dst1 = min(src1, src2); dst2 = (src1 < src2) ? 1 : 0;

8.3.5.9.35 VMPY

Assembly syntax:	**VMPY** *src1* **,** *src2* **,** *dst* **, RND:** *rnd_param*
Operation:	multiply
Classification:	2-input 1-output
Bit width:	17-bit input, 40-bit output
Delay slot:	one
C statement:	dst = round_or_shift(src1 * src2, rnd_param);

See VMADD for the RND field.

8.3.5.9.36 VMSUB

Assembly syntax:	**VMSUB** *src1* **,** *src2* **,** *src3* **,** *dst* **, RND:** *rnd_param* (dst same as src3)
Operation:	Multiply-subtract
Classification:	3-input 1-output
Bit width:	17-bit src1/src2, 40-bit src3/dst
Delay slot:	2 delay from src1/src2, 1 delay from src3
C statement:	dst = src3 - round_or_shift(src1 * src2, rnd_param);

See VMADD for the RND field.

8.3.5.9.37 VNOP

Assembly syntax:	**VNOP**
Operation:	no operation
Classification:	n/a
Bit width:	n/a
Delay slot:	no
C statement:	n/a

NOP is used to pad the two operations per clock cycle operation pipeline, so the next operation is mapped to functional unit 1. This is used to align data dependency to within functional unit 0 or 1, to avoid idle cycles being inserted by hardware.

8.3.5.9.38 VNOT

Assembly syntax:	**VNOT** *src1* **,** *dst*
Operation:	Bitwise complement
Classification:	1-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = ~src1;

8.3.5.9.39 VOR

Assembly syntax:	**VOR** *src1* **,** *src2* **,** *dst*
Operation:	Bitwise or
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 | src2;

8.3.5.9.40 VOR3

Assembly syntax:	**VOR3** *src1* **,** *src2* **,** *src3* **,** *dst* (dst same as src3)
Operation:	Bitwise-or 3 items
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst3 = src1 | src2 | src3;

Using VOR3 accelerates the morphological dilation operation.

8.3.5.9.41 VRND

Assembly syntax:	**VRND** *src1* **,** *src2* **,** *dst*
Operation:	Round src1 by number of bits expressed in src2
Classification:	2-input 1-output
Bit width:	40-bit src1/dst, 5-bit src2, allowing rounding by 0 ~ 31 bits
Delay slot:	one
C statement:	rnd_add = (src2 > 0) ? (1 << (src2 – 1)) : 0; dst = (src1 + rnd_add) >> src2;

A half-unit is added before the right-shift. For example, if src2 = 8, dst = (src1 + 128) >> 8.

Only src2[4:0] are read, so for example src2 = 40 causes a rounding by 8 bits, since src2[4:0] = 0x08.

8.3.5.9.42 VSAD

Assembly syntax:	**VSAD** *src1* **,** *src2* **,** *src3* **,** *dst* (dst same as src3)
Operation:	Sum of absolute difference, dst = |src1 – src2| + src3
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst = src3 + abs(src1 – src2);

8.3.5.9.43 VSEL

Assembly syntax:	**VSEL** *src1* **,** *src2* **,** *src3* **,** *dst* (dst same as src3)
Operation:	Select
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 ? src2 : src3;

8.3.5.9.44 VSHF

Assembly syntax:	**VSHF** *src1* **,** *src2* **,** *dst*
Operation:	Arithmetic shift, shift left when src2 > 0, shift right by –src2 when src2 < 0
Classification:	2-input 1-output
Bit width:	40-bit src1/dst, 6-bit src2, allowing shifting by -32 ~ 31 bits
Delay slot:	no
C statement:	dst = (src2 >= 0) ? (src1 << src2) : (src1 >> –src2);

VSHF is an arithmetic shift, not logic shift, compared to conventional instruction sets that have both options. This is because VCOP performs sign extension in the load stage, and all signed/unsigned 8/16/32-bit operations are carried out with the 40-bit register file.

Only src2[5:0] are read, so for example src2 = 32 causes a right-shift by 32 bits, since src2[5:0] = 0x20.

8.3.5.9.45 VSHFOR

Assembly syntax:	**VSHFOR** *src1* **,** *src2* **,** *src3* **,** *dst* (dst same as src3)
Operation:	Shift-or
Classification:	3-input 1-output
Bit width:	40-bit src1/src3, 6-bit src2
Delay slot:	one
C statement:	dst3 = (src2 >= 0) ? (src3 | (src1 << src2)) : (src3 | (src1 >> –src2));

See VSHF for the supported range of src2.

8.3.5.9.46 VSHF16

Assembly syntax:	**VSHF16** *src1* **,** *dst2* **,** *dst2*
Operation:	Shift-up 16 bits
Classification:	1-input 2-output
Bit width:	33-bit src1
Delay slot:	no
C statement:	dst1 = (src1 & 0xffff) << 16;dst2 = src1 >> 16;

Shift up src1 by 16 bits, producing two destinations to hold the outcome, dst1 having the lower unsigned 32-bit, and dst2 having the signed upper portion, sign extended from src1[32] to full 40 bits in dst2.

VSHF16 is useful for 32 × 32 multiplication.

8.3.5.9.47 VSIGN

Assembly syntax:	**VSIGN** *src1* **,** *src2* **,** *dst*
Operation:	Sign
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 == 0) ? 0 :((src1 > 0) ? src2 : –src2);

8.3.5.9.48 VSORT2

Assembly syntax:	**VSORT2** *src1* **,** *src2* **,** *dst*
Operation:	Sort two items
Classification:	2-input 2-output
Bit width:	33-bit
Delay slot:	no
C statement:	dst1 = min(src1, src2); dst2 = max(src1, src2);

8.3.5.9.49 VSUB

Assembly syntax:	**VSUB** *src1* **,** *src2* **,** *dst*
Operation:	Subtraction
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 – src2;

8.3.5.9.50 VSWAP

Assembly syntax:	**VSWAP** *cond* **,** {*src1*|*dst1*}**,** {*src2*|*dst2*}
Operation:	Conditional swap
Classification:	3-input 2-output
Bit width:	1-bit cond, 40-bit src1 & src2
Delay slot:	no
C statement:	if (cond & 1) swap(src1, src2)

Only the LSB of the vector register indicated by cond is checked. A swap/no-swap decision is made in each SIMD lane using the LSB of cond register in the lane.

8.3.5.9.51 VXOR

Assembly syntax:	**VXOR** *src1* **,** *src2* **,** *dst*
Operation:	Bitwise exclusive-or
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 ^ src2;

8.3.6 Debug Support

VCOP supports single stepping, one innermost (i4) iteration per step. Due to the deeply pipelined architecture, instruction-by-instruction single stepping support is not feasible.

Normally VCOP executes load in one stage of the pipeline, and operations and stores in the next stage of the pipeline. In single step mode, only one iteration is executed at a time, so there is no pipelining among iterations. The hardware executes all the loads, then operations and stores, then halts until instructed to continue on the next iteration. The signle step operation is controlled via the VCOP_CTRL register. The operation is enabled by setting VCOP_CTRL[0]STEP_EN to 1. Setting VCOP_CTRL[1]STEP_GO to 1 the engine starts executing i4 iteration.

As outlined in the previous section, step_en, step_go, step_rdy register fields are provided for single stepping control. In addition, VCOP internal state is exported as read-only view as memory-map registers as well.

Interaction of debug driver, scalar core, and VCOP is as follows:

1. Debug driver enables debug mode by setting VCOP_CTRL[0]STEP_EN = 1 and wait for VCOP_STATUS[0]STEP_RDY = 1.
2. Scalar core sends the vector command to be single-stepped the normal fashion.
3. Vector core executes the vector command and stalls at the end of the first i4 iteration and assert sVCOP_STATUS[0]STEP_RDY.
4. When VCOP_STATUS[0]STEP_RDY =1, debug driver accesses buffer switch and switch WBUF, IBUFL/H memories to system bus.
5. Debug driver read out the memory contents to refresh on the debugger user interface.
6. Debug driver read out VCOP internal state to refresh on the debugger user interface.
7. Debug driver accesses buffer switch and switch memories to VCOP.
8. Debug driver triggers VCOP to take the next step by writing ‘1’ to step_go, and wait for VCOP_STATUS[0]STEP_RDY = 1.
9. Repeat 3 through 8, until completion of the vector command or user input to stop single-stepping.
10. The debug driver disables the debug mode by setting VCOP_CTRL[0]STEP_EN = 0.
11. VCOP and scalar core continue to operate.

8.3.7 VCOP Register Manual

8.3.7.1 VCOP Instance Summary

Table 8-363. VCOP Instance Summary

Module Name	Base Address L3_MAIN Interconnect	Size
EVE1_VCOP	0x4208 4000	4KiB
EVE2_VCOP	0x4218 4000	4KiB

8.3.7.2 VCOP Registers

8.3.7.2.1 VCOP Registers Mapping Summary

Table 8-364. VCOP Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_VCOP Base Address	EVE2_VCOP Base Address
VCOP_PID	R	32	0x0000 0000	0x4208 4000	0x4218 4000
VCOP_CTRL	RW	32	0x0000 0004	0x4208 4004	0x4218 4004
VCOP_STATUS	R	32	0x0000 0008	0x4208 4008	0x4218 4008
VCOP_MAX_ITERS	RW	32	0x0000 000C	0x4208 400C	0x4218 400C
VCOP_ERROR	RW	32	0x0000 0010	0x4208 4010	0x4218 4010
VCOP_VLOOP_PTR	R	32	0x0000 0020	0x4208 4020	0x4218 4020
VCOP_PARAM_PTR	R	32	0x0000 0024	0x4208 4024	0x4218 4024
VCOP_I0_I1	R	32	0x0000 0030	0x4208 4030	0x4218 4030
VCOP_I2_I3	R	32	0x0000 0034	0x4208 4034	0x4218 4034
VCOP_I4	R	32	0x0000 0038	0x4208 4038	0x4218 4038
VCOP_LD_PTR_i (1)	R	32	0x0000 0040 + (0x4*i)	0x4208 4040 + (0x4*i)	0x4218 4040 + (0x4*i)
VCOP_ST_PTR_j (2)	R	32	0x0000 0060 + (0x4*j)	0x4208 4060 + (0x4*j)	0x4218 4060 + (0x4*j)

(1) i = 0 to 7 for EVE1_VCOP
i = 0 to 7 for EVE2_VCOP
(2) j = 0 to 7 for EVE1_VCOP
j = 0 to 7 for EVE2_VCOP

8.3.7.2.2 VCOP Register Description

Table 8-365. VCOP_PID

Address Offset	0x0000 0000		
Physical Address	0x4208 4000 0x4218 4000	**Instance**	EVE1_VCOP EVE2_VCOP
Description			
Type	R		

31–0
PID

Bits	Field Name	Description	Type	Reset
31:0	PID	IP Revision	R	0x0

Table 8-366. Register Call Summary for Register VCOP_PID

VCOP CPU and Instruction Set

- VCOP Registers Mapping Summary: [1]

Table 8-367. VCOP_CTRL

Address Offset	0x0000 0004		
Physical Address	0x4208 4004 0x4218 4004	**Instance**	EVE1_VCOP EVE2_VCOP
Description	VCOP Control Register		
Type	RW		

31–2	1	0
RESERVED	STEP_GO	STEP_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	STEP_GO	Starts executing a single i4 iteration 0: NOP 1: START	RW	0x0
0	STEP_EN	Enable Single Step mode 0: Disable 1: Enable	RW	0x0

Table 8-368. Register Call Summary for Register VCOP_CTRL

VCOP CPU and Instruction Set

- Debug Support: [0] [1] [2] [3] [4]
- VCOP Registers Mapping Summary: [6]

Table 8-369. VCOP_STATUS

Address Offset	0x0000 0008		
Physical Address	0x4208 4008 0x4218 4008	**Instance**	EVE1_VCOP EVE2_VCOP
Description	VCOP status register.		
Type	R		

31–3	2	1	0
RESERVED	VEC_RDY	VEC_DONE	STEP_RDY

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved. Read returns 0s	R	0x0
2	VEC_RDY	Vector core ready to accept next vector instruction	R	0x0
1	VEC_DONE	Vector core has completed execution of submitted vector loops.	R	0x0
0	STEP_RDY	Ready for next step (single step) 0: Busy 1: Idle and ready for next step	R	0x0

Table 8-370. Register Call Summary for Register VCOP_STATUS

VCOP CPU and Instruction Set

- Wait for Vector Core Ready: [0]
- VCOP Error Handling: [1]
- Debug Support: [2] [3] [4] [5]
- VCOP Registers Mapping Summary: [7]

Table 8-371. VCOP_MAX_ITERS

Address Offset	0x0000 000C		
Physical Address	0x4208 400C 0x4218 400C	**Instance**	EVE1_VCOP EVE2_VCOP
Description			
Type	RW		

31–16	15–0
RESERVED	MAX_ITERS

Bits	Field Name	Description	Type	Reset
31:16	RESESERVED	Reserved. Read returns 0s	R	0x0
15:0	MAX_ITERS	Maximum iteration count. Send interrupt when a loop in execution exceeds the programmed max iteration count. This is to guard against VCOP hangs due to run-away program. 0: Disable (default) 1: Enable	RW	0x0

Table 8-372. Register Call Summary for Register VCOP_MAX_ITERS

VCOP CPU and Instruction Set

- VCOP Registers Mapping Summary: [1]

Table 8-373. VCOP_ERROR

Address Offset	0x0000 0010		
Physical Address	0x4208 4010 0x4218 4010	**Instance**	EVE1_VCOP EVE2_VCOP
Description	Error interrupt enalbe and status register. Writing 1 to the ERR_STi bits clears the interrupt status.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ERR_DIS7	ERR_DIS6	ERR_DIS5	ERR_DIS4	ERR_DIS3	ERR_DIS2	ERR_DIS1	ERR_DIS0	RESERVED								ERR_ST7	ERR_ST6	ERR_ST5	ERR_ST4	ERR_ST3	ERR_ST2	ERR_ST1	ERR_ST0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved. Read returns 0s	RW	0x0
23	ERR_DIS7	Error Interrupt disable. 0:Enable 1: Disable ST_PDDA bank conflict	RW	0x0
22	ERR_DIS6	Error Interrupt disable. 0:Enable 1: Disable ST WBUF out-of-bound	RW	0x0
21	ERR_DIS5	Error Interrupt disable. 0:Enable 1: Disable ST IBUF out-of-bound	RW	0x0
20	ERR_DIS4	Error Interrupt disable. 0:Enable 1: Disable LD WBUF out-of-bound	RW	0x0
19	ERR_DIS3	Error Interrupt disable. 0:Enable 1: Disable LD IBUF out-of-bound	RW	0x0
18	ERR_DIS2	Error Interrupt disable. 0: Enable 1: Disable Illegal parameter (pointer not 32-bit aligned, pointer out-of-bound, exceed max repeat count)	RW	0x0
17	ERR_DIS1	Error Interrupt disable. 0:Enable 1:Disable Illegal instruction, all other causes than inside-loop instructions detected outside loop	RW	0x0
16	ERR_DIS0	Error Interrupt disable. 0: Enable 1: Disable Illegal instruction; inside-loop instructions (eg, VADD) detected outside loop.When this occurs, the decode value is indeterminate, since VCOP expects valid PC on vec_paddr bus, and ARP32 only sends PC with valid VLOOP instruction.	RW	0x0
15:8	RESERVED	Reserved. Read returns 0s	RW	0x0
7	ERR_ST7	ST_PDDA bank conflict error status: 0: No error 1: Error	RW	0x0
6	ERR_ST6	ST WBUF out-of-bound error status: 0: No error 1: Error	RW	0x0
5	ERR_ST5	ST IBUF out-of-bound error status: 0: No error 1: Error	RW	0x0
4	ERR_ST4	LD WBUF out-of-bound error status: 0: No error 1: Error	RW	0x0

Bits	Field Name	Description	Type	Reset
3	ERR_ST3	LD IBUF out-of-bound error status: 0: No error 1: Error	RW	0x0
2	ERR_ST2	Illegal parameter error status. Effected when pointer is not 32-bit aligned, pointer is outof- bound, or exceed max repeat count. 0: No error 1: Error	RW	0x0
1	ERR_ST1	Illegal instruction error status. Effected by all other causes than inside-loop instructions detected outside loop. 0: No error 1: Error	RW	0x0
0	ERR_ST0	Illegal instruction error status. Effected by inside-loop instructions (eg, VADD) detected outside loop. When this occurs, VCOP_VLOOP_PTR_DEC value is indeterminate, since VCOP expects valid PC on vec_paddr bus, and ARP32 only sends PC with valid VLOOP instruction. 0: No error 1: Error	RW	0x0

Table 8-374. Register Call Summary for Register VCOP_ERROR

VCOP CPU and Instruction Set

- VCOP Error Handling: [0] [1]
- VCOP Registers Mapping Summary: [3]

Table 8-375. VCOP_VLOOP_PTR

Address Offset	0x0000 0020		
Physical Address	0x4208 4020 0x4218 4020	**Instance**	EVE1_VCOP EVE2_VCOP
Description	The VLOOP pointer		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLOOP_PTR																															

Bits	Field Name	Description	Type	Reset
31:0	VLOOP_PTR	VLOOP pointer.	R	0x0

Table 8-376. Register Call Summary for Register VCOP_VLOOP_PTR

VCOP CPU and Instruction Set

- Instruction Organization: [0]
- VCOP Registers Mapping Summary: [2]

Table 8-377. VCOP_PARAM_PTR

Address Offset	0x0000 0024		
Physical Address	0x4208 4024 0x4218 4024	**Instance**	EVE1_VCOP EVE2_VCOP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARAM_PTR																															

Bits	Field Name	Description	Type	Reset
31:0	PARAM_PTR	Points to the beginning of parameter block for the loop in execution.	R	0x0

Table 8-378. Register Call Summary for Register VCOP_PARAM_PTR

VCOP CPU and Instruction Set

- Parameter Pointer: [0] [1]
- VCOP Registers Mapping Summary: [3]

Table 8-379. VCOP_I0_I1

Address Offset	0x0000 0030		
Physical Address	0x4208 4030 0x4218 4030	**Instance**	EVE1_VCOP EVE2_VCOP
Description	I0, I1 loop variables register provides a snapshot of i0 and i1		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I1																I0															

Bits	Field Name	Description	Type	Reset
31:16	I1	Snapshot of I1 loop variable.	R	0x0
15:0	I0	Snapshot of I0 loop variable.	R	0x0

Table 8-380. Register Call Summary for Register VCOP_I0_I1

VCOP CPU and Instruction Set

- Nested for Loop Model: [0] [1]
- VCOP Registers Mapping Summary: [3]

Table 8-381. VCOP_I2_I3

Address Offset	0x0000 0034		
Physical Address	0x4208 4034 0x4218 4034	**Instance**	EVE1_VCOP EVE2_VCOP
Description	I2, I3 loop variables register provides a snapshot of i2 and i3		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I3																I2															

Bits	Field Name	Description	Type	Reset
31:16	I3	Snapshot of I2 loop variable.	R	0x0
15:0	I2	Snapshot of I3 loop variable.	R	0x0

Table 8-382. Register Call Summary for Register VCOP_I2_I3

VCOP CPU and Instruction Set

- Nested for Loop Model: [0] [1]
- VCOP Registers Mapping Summary: [3]

Table 8-383. VCOP_I4

Address Offset	0x0000 0038

Table 8-383. VCOP_I4 (continued)

Physical Address	0x4208 4038 0x4218 4038	**Instance**	EVE1_VCOP EVE2_VCOP
Description	I4 loop variables register provides a snapshot of i4		
Type	R		

31–16	15–0
RESERVED	I4

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved Read returns 0s	R	0x0
15:0	I4	Snapshot of I4 loop variable.	R	0x0

Table 8-384. Register Call Summary for Register VCOP_I4

VCOP CPU and Instruction Set

- Nested for Loop Model: [0]
- VCOP Registers Mapping Summary: [2]

Table 8-385. VCOP_LD_PTR_i

Address Offset	0x0000 0040 + (0x4*i)		
Physical Address	0x4208 4040 + (0x4*i) 0x4218 4040 + (0x4*i)	**Instance**	EVE1_VCOP EVE2_VCOP
Description	The LD pointer registers 0 to 7 or (VCOP_LD_PTR(0..7)) is a snapshot of the LD memory address. The LD unit is identified by the destination vector register V0..V7		
Type	R		

31–0
LD_PTRi

Bits	Field Name	Description	Type	Reset
31:0	LD_PTRi	LD pointer i (i = 0 to 7).	R	0x0

Table 8-386. Register Call Summary for Register VCOP_LD_PTR_i

VCOP CPU and Instruction Set

- Features: [0]
- Vector Load: [1]
- VCOP Registers Mapping Summary: [3]

Table 8-387. VCOP_ST_PTR_j

Address Offset	0x0000 0060 + (0x4*j)		
Physical Address	0x4208 4060 + (0x4*j) 0x4218 4060 + (0x4*j)	**Instance**	EVE1_VCOP EVE2_VCOP
Description	The ST pointer registers 0 to 7 (VCOP_VLOOP_ST_PTR(0..7)) is a snapshot of the ST memory address. The ST unit is identified by the order ST appears in program.		
Type	R		

31–0
ST_PTR0

Bits	Field Name	Description	Type	Reset
31:0	ST_PTR0	ST pointer j (j=0 to 7).	R	0x0

Table 8-388. Register Call Summary for Register VCOP_ST_PTR_j

VCOP CPU and Instruction Set
• Features: [0]
• Vector Load: [1]
• VCOP Registers Mapping Summary: [3]

Chapter 9

Video Input Port

This chapter describes the Video Input Port (VIP) module for the device.

Note

This chapter describes a module (subsystem) in the superset device. The available number of instances and supported set of features is device part number dependent. Refer to device Data Manual, for more information.

9.1 VIP Overview.......... 1883
9.2 VIP Environment.......... 1885
9.3 VIP Integration.......... 1891
9.4 VIP Functional Description.......... 1894
9.5 VIP Register Manual.......... 2031

9.1 VIP Overview

The VIP module provides video capture functions for the device. VIP incorporates a multi-channel raw video parser, various video processing blocks, and a flexible Video Port Direct Memory Access (VPDMA) engine to store incoming video in various formats. The device integrates three instantiations of the VIP module giving the ability of capturing up to ten video streams.

Figure 9-1 shows a block diagram with the VIP modules within the device.

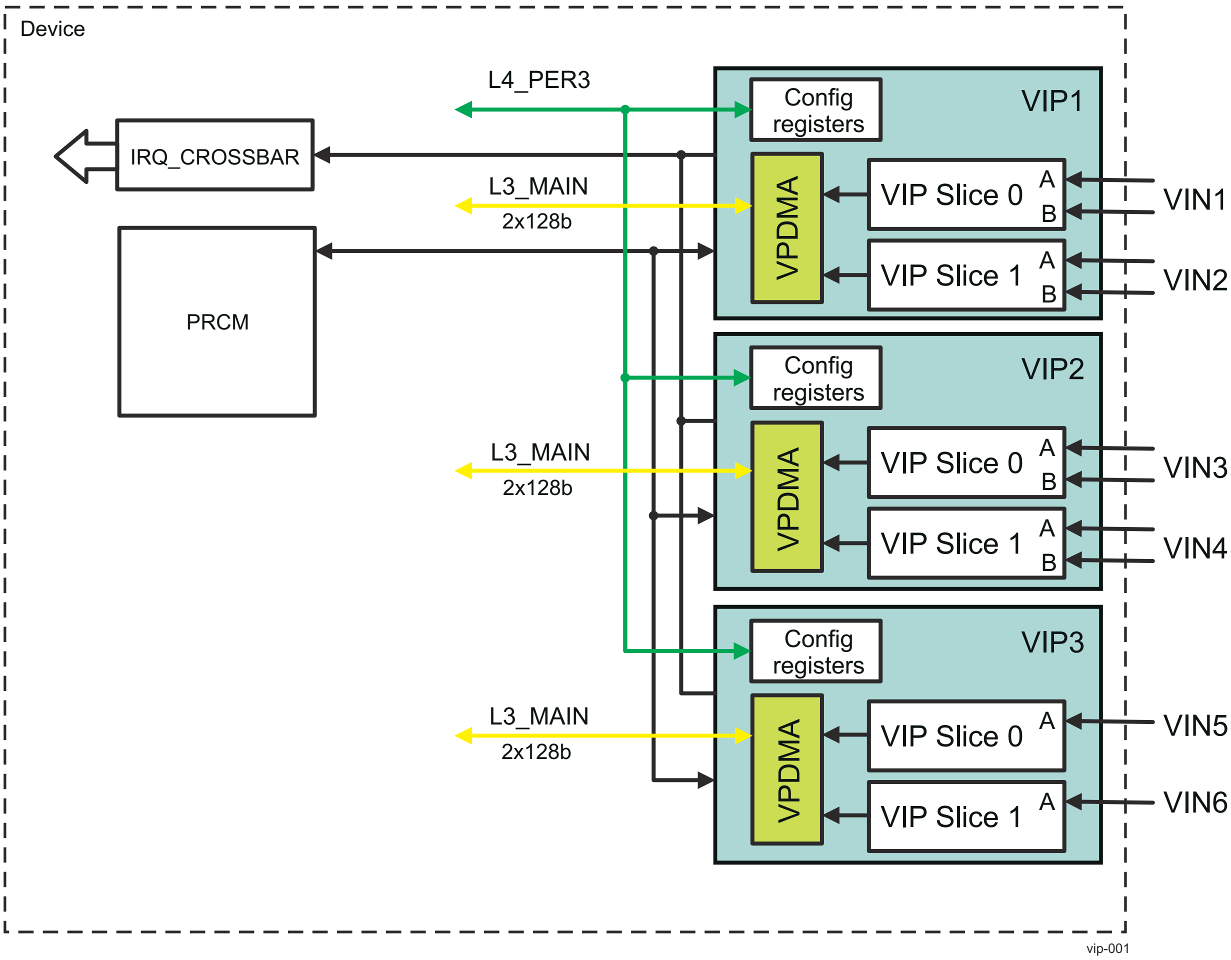


Figure 9-1. VIP Overview

A VIP module includes the following main features:

- Two independently configurable external video input capture slices (Slice 0 and Slice 1) each providing up to two video input ports, Port A and Port B:
  - Port A can be configured as a 24/16/8-bit port. Port A of VIP3 slices provides only 16-bit interface.
  - Port B is a fixed 8-bit port. Port B of VIP3 slices is not used.
  - At device level, the same device pads may be shared between Port A and Port B of each slice. For more information, see the multiplexing characteristics in device Data Manual.
- Each video Port A can be operated as a port with clock independent input channels (with interleaved or separated Y/C data input). Embedded sync and external sync modes are supported for all input configurations.
- Support for a single external asynchronous pixel clock, up to 165MHz per port.
- Pixel Clock Input Domain Port A supports up to one 24-bit input data bus, including BT.1120 style embedded sync for 16-bit and 24-bit data.
- Embedded Sync data interface mode supports single or multiplexed sources

- Discrete Sync data interface mode supports only single source input
- 24-bit data input plus discrete syncs can be configured to include:
  - 8-bit YUV422 (Y and U/V time interleaved)
  - 16-bit YUV422 (CbY and CrY time interleaved)
  - 24-bit YUV444
  - 16-bit RGB565
  - 24-bit RGB888
  - 12/16-bit RAW Capture
  - 24-bit RAW capture
- Discrete sync modes include:
  - VSYNC + HSYNC (FID determined by FID signal pin or HSYNC/VSYNC skew)
  - VSYNC + ACTVID + FID
  - VBLANK + ACTVID (ACTVID toggles in VBLANK) + FID
  - VBLANK + ACTVID (no ACTVID toggles in VBLANK) + FID
- Multichannel parser (embedded syncs only):
  - Embedded syncs only
  - Pixel (2x or 4x) or Line multiplexed modes supported
  - Performs demultiplexing and basic error checking
  - Supports maximum of 9 channels in Line Mux (8 normal + 1 split line)
- Ancillary data capture support:
  - For 16-bit or 24-bit input, ancillary data may be extracted from any single channel
  - For 8-bit time interleaved input, ancillary data can be chosen from the Luma channel, the Chroma channel, or both channels
  - Horizontal blanking interval data capture only supported when using discrete syncs (VSYNC + HSYNC or VSYNC + HBLANK)
  - Ancillary data extraction supported on multichannel capture as well as single source streams
- Format conversion and scaling:
  - Programmable color space conversion
  - YUV422 to YUV444 conversion
  - YUV444 to YUV422 conversion
  - YUV422 to YUV420 conversion
  - YUV444 Source: YUV444 to YUV444, YUV444 to RGB888, YUV444 to YUV422, YUV444 to YUV420
  - RGB888 Source: RGB888 to RGB888, RGB888 to YUV444, RGB888 to YUV422, RGB888 to YUV420
  - YUV422 Source: YUV422 to YUV422, YUV422 to YUV420, YUV422 to YUV444, YUV422 to RGB888
  - Supports RAW to RAW (no processing)
  - Scaling and format conversions do not work for multiplexed input
- Supports up to 2047 pixels wide input - when scaling is engaged
- Supports up to 3840 pixels wide input - when only chroma up/down sampling is engaged, without scaling
- Supports up to 4095 pixels wide input - without scaling and chroma up/down sampling
- The maximum supported input resolution is further limited by:
  - Pixel clock and feature-dependent constraints
  - For RGB24-bit format (RAW data), the maximum frame width is limited to 2730 pixels

A VPDMA module includes the following main features:

- VPDMA output buffer size restriction feature, which ensures that writes do not exceed allocated memory buffer size
- Support for Tiled (2D) and raster addressing without bandwidth penalty
- Dual clients per channel allows for capture of scaled and nonscaled versions of the data stream (non-multiplexed mode only)
- Start on new frame capability
- Interrupt every X number of frames
- Interrupt every X lines (synced to frame start)
- Dynamic MFLAG generation

9.2 VIP Environment

This section describes the VIP modules from an environment point of view (external connections). It describes the VIP connectivity options and lists all possible interfaces. Figure 9-2 is a block diagram of the VIP environment.

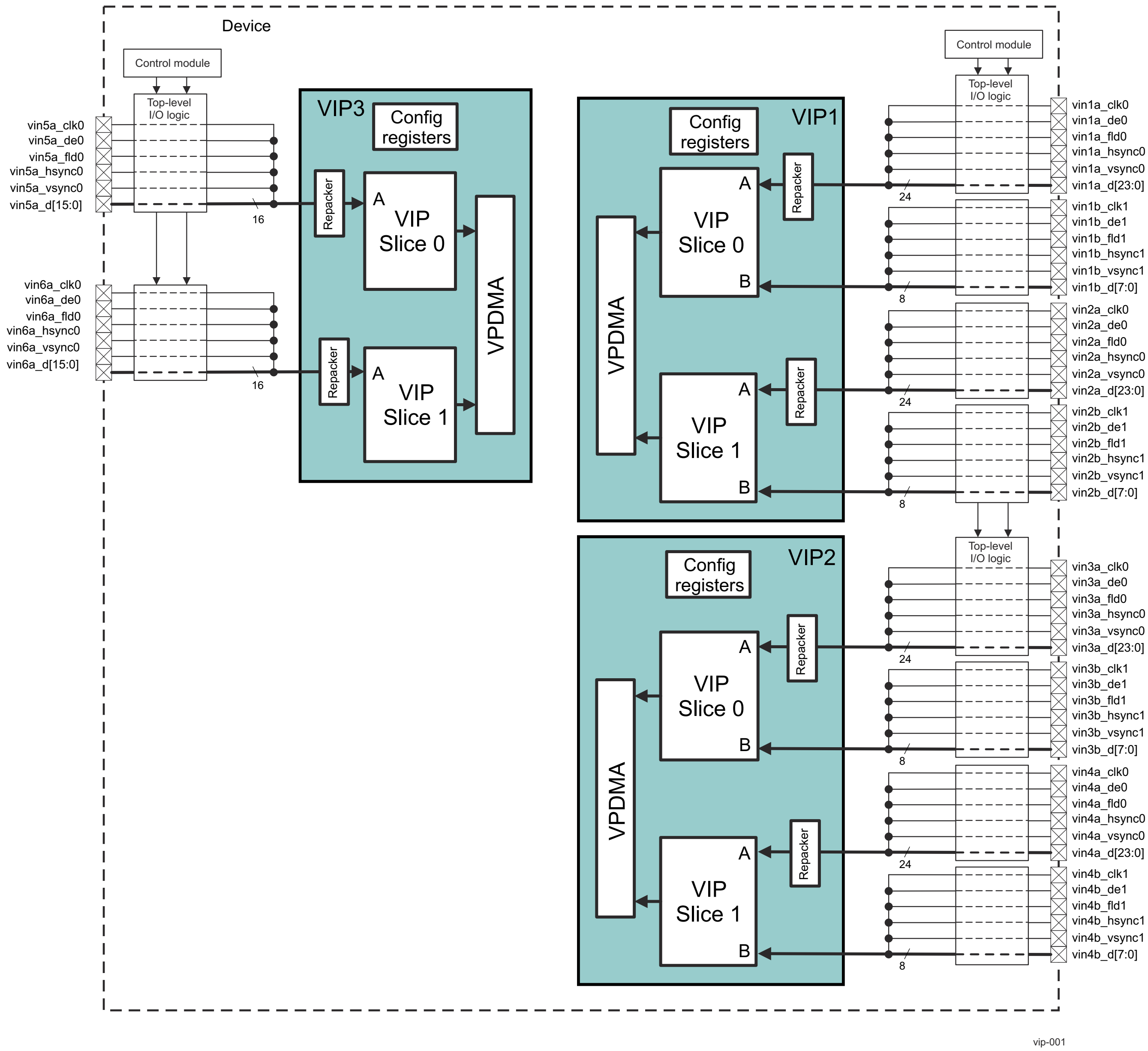


Figure 9-2. VIP Environment

Note

VIP3 does not include port B, and VIP3 Port A can be configured up to 16 bits, for each slice.

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to different pads of the device and is programmable in the device Control Module registers and/or dedicated module registers. For more information, see *Pad Configuration Registers* in *Control Module*.

Table 9-1. VIP1 Interface Signals

Sub-module Name	Signal Name	Type[(1)]	Description
Slice 0	vin1a_d[23:0]	I	Pixel data.
Port A	vin1a_clk0	I	Pixel clock.
	vin1a_vsync0	I	Vertical synchronization.
	vin1a_hsync0	I	Horizontal synchronization.
	vin1a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin1a_fld0	I	The FID signal indicates the field identifier for the video input field: • 0 means even. • 1 means odd.
Slice 0	vin1b_d[7:0]	I	Pixel data.
Port B	vin1b_clk1	I	Pixel clock.
	vin1b_vsync1	I	Vertical synchronization.
	vin1b_hsync1	I	Horizontal synchronization.
	vin1b_de1	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin1b_fld1	I	The FID signal indicates the field identifier for the video input field: • 0 means even. • 1 means odd.
Slice 1	vin2a_d[23:0]	I	Pixel data.
Port A	vin2a_clk0	I	Pixel clock.
	vin2a_vsync0	I	Vertical synchronization.
	vin2a_hsync0	I	Horizontal synchronization.
	vin2a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin2a_fld0	I	The FID signal indicates the field identifier for the video input field: • 0 means even. • 1 means odd.
Slice 1	vin2b_d[7:0]	I	Pixel data.
Port B	vin2b_clk1	I	Pixel clock.
	vin2b_vsync1	I	Vertical synchronization.
	vin2b_hsync1	I	Horizontal synchronization.

Table 9-1. VIP1 Interface Signals (continued)

Sub-module Name	Signal Name	Type(1)	Description
	vin2b_de1	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin2b_fld1	I	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.

(1) I = Input, O = Output, I/O = Input/Output

Table 9-2. VIP2 Interface Signals

Sub-module Name	Signal Name	Type(1)	Description
Slice 0	vin3a_d[23:0]	I	Pixel data.
Port A	vin3a_clk0	I	Pixel clock.
	vin3a_vsync0	I	Vertical synchronization.
	vin3a_hsync0	I	Horizontal synchronization.
	vin3a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin3a_fld0	I	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.
Slice 0	vin3b_d[7:0]	I	Pixel data.
Port B	vin3b_clk1	I	Pixel clock.
	vin3b_vsync1	I	Vertical synchronization.
	vin3b_hsync1	I	Horizontal synchronization.
	vin3b_de1	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin3b_fld1	I	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.
Slice 1	vin4a_d[23:0]	I	Pixel data
Port A	vin4a_clk0	I	Pixel clock
	vin4a_vsync0	I	Vertical synchronization.
	vin4a_hsync0	I	Horizontal synchronization.
	vin4a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.

Table 9-2. VIP2 Interface Signals (continued)

Sub-module Name	Signal Name	Type(1)	Description
	vin4a_fld0	I	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.
Slice 1	vin4b_d[7:0]	I	Pixel data.
Port B	vin4b_clk1	I	Pixel clock.
	vin4b_vsync1	I	Vertical synchronization.
	vin4b_hsync1	I	Horizontal synchronization.
	vin4b_de1	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin4b_fld1	I	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.

(1) I = Input, O = Output, I/O = Input/Output

Table 9-3. VIP3 Interface Signals

Sub-module Name	Signal Name	Type(1)	Description
Slice 0	vin5a_d[15:0]	I	Pixel data.
Port A	vin5a_clk0	I	Pixel clock.
	vin5a_vsync0	I	Vertical synchronization.
	vin5a_hsync0	I	Horizontal synchronization.
	vin5a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin5a_fld0	I	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.
Slice 1	vin6a_d[15:0]	I	Pixel data.
Port A	vin6a_clk0	I	Pixel clock.
	vin6a_vsync0	I	Vertical synchronization.
	vin6a_hsync0	I	Horizontal synchronization.
	vin6a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin6a_fld0	I	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.

(1) I = Input, O = Output, I/O = Input/Output

Note

At device level, the same device pads may be shared between Port A and Port B of each slice. For more information, see the multiplexing characteristics in device Data Manual.

Table 9-4 and Table 9-5 summarize the mapping of RGB and YUV color components to VIP input data signals, with corresponding settings of VIP_MAIN[1:0] DATA_INTERFACE_MODE register bit-field.

Table 9-4. VIP Port A Input Data Signals to RGB and YUV Color Components Mapping

VIP Port A Data Signals	24-bit RGB888 Input Mode	16-bit RGB565 Input Mode	24-bit YUV444 Input Mode	16-bit YUV422 Input Mode (1)	8-bit YUV422 Input Mode (2)
X = 1 to 6	DATA_INTERFACE_MODE = 00b	DATA_INTERFACE_MODE = 00b	DATA_INTERFACE_MODE = 00b	DATA_INTERFACE_MODE = 01b	DATA_INTERFACE_MODE = 10b
vinXa_d23	Red 7 (MS bit)	Red 4 (MS bit)	Y7 (MS bit)	-	-
vinXa_d22	Red 6	Red 3	Y6	-	-
vinXa_d21	Red 5	Red 2	Y5	-	-
vinXa_d20	Red 4	Red 1	Y4	-	-
vinXa_d19	Red 3	Red 0	Y3	-	-
vinXa_d18	Red 2	-	Y2	-	-
vinXa_d17	Red 1	-	Y1	-	-
vinXa_d16	Red 0	-	Y0	-	-
vinXa_d15	Green 7	Green 5	Cb7	Y7 (MS bit)	-
vinXa_d14	Green 6	Green 4	Cb6	Y6	-
vinXa_d13	Green 5	Green 3	Cb5	Y5	-
vinXa_d12	Green 4	Green 2	Cb4	Y4	-
vinXa_d11	Green 3	Green 1	Cb3	Y3	-
vinXa_d10	Green 2	Green 0	Cb2	Y2	-
vinXa_d9	Green 1	-	Cb1	Y1	-
vinXa_d8	Green 0	-	Cb0	Y0	-
vinXa_d7	Blue 7	Blue 4	Cr7	Cb7/Cr7/...	Cb7/Y7/Cr7/... (MS bit)
vinXa_d6	Blue 6	Blue 3	Cr6	Cb6/Cr6/...	Cb6/Y6/Cr6/...
vinXa_d5	Blue 5	Blue 2	Cr5	Cb5/Cr5/...	Cb5/Y5/Cr5/...
vinXa_d4	Blue 4	Blue 1	Cr4	Cb4/Cr4/...	Cb4/Y4/Cr4/...
vinXa_d3	Blue 3	Blue 0 (LS bit)	Cr3	Cb3/Cr3/...	Cb3/Y3/Cr3/...
vinXa_d2	Blue 2	-	Cr2	Cb2/Cr2/...	Cb2/Y2/Cr2/...
vinXa_d1	Blue 1	-	Cr1	Cb1/Cr1/...	Cb1/Y1/Cr1/...
vinXa_d0	Blue 0 (LS bit)	-	Cr0 (LS bit)	Cb0/Cr0/... (LS bit)	Cb0/Y0/Cr0/... (LS bit)

(1) Chroma is time division multiplexed (interleaved). For more details, see Section 9.4.5.6.2, *16b Interface Mode*.
(2) Luma and Chroma are time division multiplexed (interleaved). For more details, see Section 9.4.5.6.1, *8b Interface Mode*.

Note

16-bit RGB data can be captured also on the vinXa_d[15:0] input data bus of Port A. In this case, the 16-bit RGB data captured by the VIP_PARSER will be passed to VPDMA, as if it is a 16-bit YUV data. The VIP_MAIN[1:0] DATA_INTERFACE_MODE register bit-field must be configured for a 16-bit input mode. The VPDMA will then directly store this data in memory as a 16-bit data, provided that any 16-bit data type in the VPDMA outbound descriptor is set.

Table 9-5. VIP Port B Input Data Signals to YUV Color Components Mapping

VIP Port B Data Signals	8-bit YUV422 Input Mode (1)
Y = 1 to 4	DATA_INTERFACE_MODE = 10b
vinYb_d7	Cb7/Y7/Cr7/... (MS bit)
vinYb_d6	Cb6/Y6/Cr6/...

Table 9-5. VIP Port B Input Data Signals to YUV Color Components Mapping (continued)

VIP Port B Data Signals	8-bit YUV422 Input Mode (1)
vinYb_d5	Cb5/Y5/Cr5/...
vinYb_d4	Cb4/Y4/Cr4/...
vinYb_d3	Cb3/Y3/Cr3/...
vinYb_d2	Cb2/Y2/Cr2/...
vinYb_d1	Cb1/Y1/Cr1/...
vinYb_d0	Cb0/Y0/Cr0/... (LS bit)

(1) Luma and Chroma are time division multiplexed (interleaved). For more details, see Section 9.4.5.6.1, *8b Interface Mode*.

9.3 VIP Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests. Figure 9-3 summarizes the integration of the module in the device.

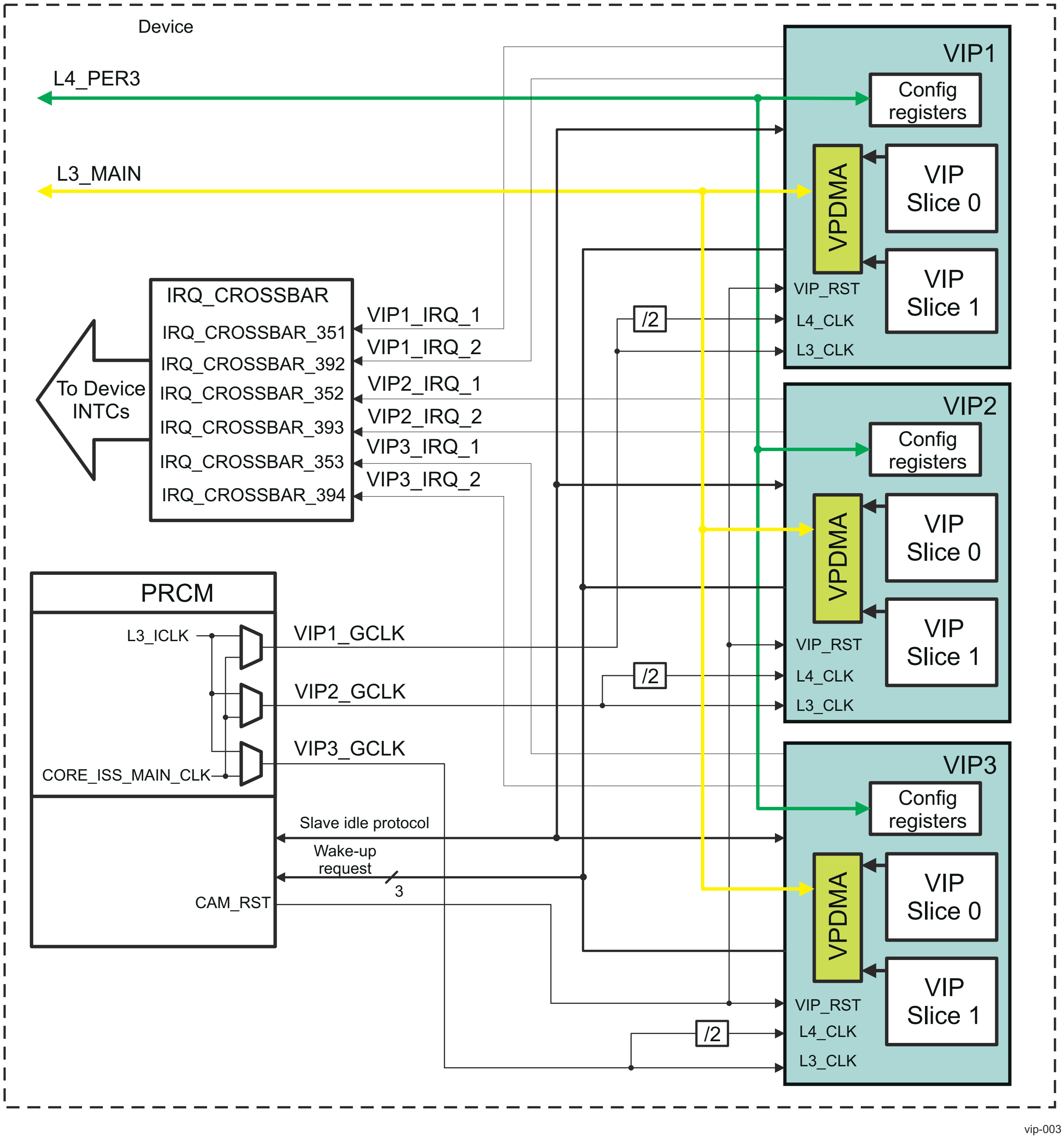


Figure 9-3. VIP Integration

Table 9-6 and Table 9-7 list the integration attributes and clock and resets, respectively.

Table 9-6. VIP Integration Attributes

Module Instance	Attributes	
	Power Domain	**Interconnect**
VIP1 VIP2 VIP3	PD_CAM	L4_PER3 for configuration L3_MAIN for data (through VPDMA module)

Table 9-7. VIP Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VIP1	L3_CLK PROC_CLK	VIP1_GCLK	PRCM	L3_CLK is the clock used to drive and receive data over the bus to L3_MAIN. The VIP subsystem uses this clock to fetch external data and transfer this data to internal processing PROC_CLK is the clock used to drive data processing within the VIP subsystem.
	L4_CLK	VIP1_GCLK/2	PRCM	L4_CLK is the interface clock for Memory Mapped Registers (MMR) configuration bus
VIP2	L3_CLK PROC_CLK	VIP2_GCLK	PRCM	L3_CLK is the clock used to drive and receive data over the bus to L3_MAIN. The VIP subsystem uses this clock to fetch external data and transfer this data to internal processing PROC_CLK is the clock used to drive data processing within the VIP subsystem.
	L4_CLK	VIP2_GCLK/2	PRCM	L4_CLK is the interface clock for Memory Mapped Registers (MMR) configuration bus
VIP3	L3_CLK PROC_CLK	VIP3_GCLK	PRCM	L3_CLK is the clock used to drive and receive data over the bus to L3_MAIN. The VIP subsystem uses this clock to fetch external data and transfer this data to internal processing PROC_CLK is the clock used to drive data processing within the VIP subsystem.
	L4_CLK	VIP3_GCLK/2	PRCM	L4_CLK is the interface clock for Memory Mapped Registers (MMR) configuration bus
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VIP1	VIP_RST	CAM_RST	PRCM	VIP1 Reset
VIP2	VIP_RST	CAM_RST	PRCM	VIP2 Reset
VIP3	VIP_RST	CAM_RST	PRCM	VIP3 Reset

Table 9-8. VIP Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
VIP1	VIP1_IRQ1 VIP1_IRQ2	IRQ_CROSSBAR_351 IRQ_CROSSBAR_392	N/A N/A	VIP1 interrupt requests. These IRQ source signals are not mapped by default to any device INTC.
VIP2	VIP2_IRQ1 VIP2_IRQ2	IRQ_CROSSBAR_352 IRQ_CROSSBAR_393	N/A N/A	VIP2 interrupt requests. These IRQ source signals are not mapped by default to any device INTC.

Table 9-8. VIP Hardware Requests (continued)

VIP3	VIP3_IRQ1	IRQ_CROSSBAR_353	N/A	VIP3 interrupt requests. These IRQ source signals are not mapped by default to any device INTC.
	VIP3_IRQ2	IRQ_CROSSBAR_394	N/A	

Note

The “**Default Mapping**” column in Table 9-8 *VIP Hardware Requests* shows the default mapping of module IRQ source signals. These module IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

9.4 VIP Functional Description

9.4.1 VIP Block Diagram

Figure 9-4 shows the internal structure of a single VIP module in the device.

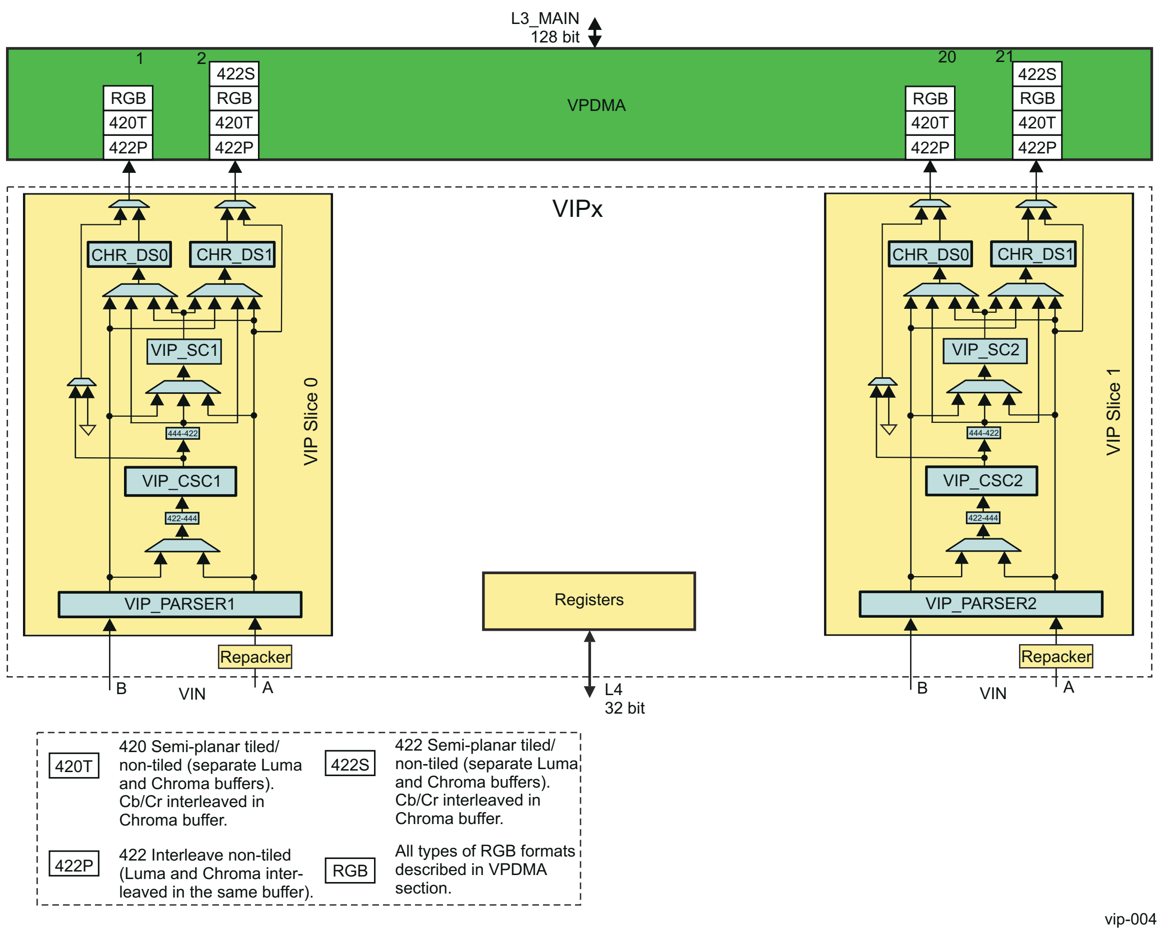


Figure 9-4. VIP Block Diagram

9.4.2 VIP Software Reset

Software reset in the VIP module can be done by setting the VIP_CLKC_RST[16] VIP1_DP_RST for Slice 0, VIP_CLKC_RST[17] VIP2_DP_RST for Slice 1, VIP_CLKC_RST[0] VPDMA_RST for VIP VPDMA to 0x1. By setting VIP_CLKC_RST[31] MAIN_RST reset is performed for all modules within the instance. Software must ensure that the software reset completes before performing operations within the VIP module.

9.4.3 VIP Power and Clocks Management

The VIP modules support the MStandby/Wait and IdleReq/SidleAck protocols as defined in *Power, Reset, and Clock Management.*

Power Management within the VIP module can be accomplished in several ways:

- L4 MConnect/SConnect can disable the internal L4 clock network
- L3 MConnect/Sconnect can disable the internal L3 clock network

These items are accomplished using the standard slave idle (for L4) and master standby (for L3) protocols. When these modules are instructed to disable clocks for the internal L3 or L4 (MMR) clock domains, the internal clock networks will be shut down. This shut down applies to the clock signals - L3_CLK and L4_CLK.

9.4.3.1 VIP Clocks

The VIP internal clock domains can only be shut down by writing the appropriate register bit within the Clock Enable register - VIP_CLKC_CLKEN[16] VIP1_DP_EN for slice0, VIP_CLKC_CLKEN[17] VIP2_DP_EN for slice1 and VIP_CLKC_CLKEN[0] VPDMA_EN for the VPDMA engine

9.4.3.2 VIP Idle Mode

The VIP supports no-idle mode, force-idle mode, and smart-idle mode. The mode can be selected byprogramming the appropriate value in the VIP_SYSCONFIG[3:2] IDLEMODE bit field.

Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.

- Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements.
- No-idle mode: local target never enters idle state.
- Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA related requests) wakeup events

9.4.3.3 VIP StandBy Mode

The VIP supports no-standby mode and force-standby mode. The mode is set in the VIP_SYSCONFIG[5:4] STANDBYMODE bit field.

Configuration of the local initiator state management mode:

- Force-standby mode: local initiator is unconditionally placed in standby state.
- No-standby mode: local initiator is unconditionally placed out of standby state.

9.4.4 VIP Slice

9.4.4.1 VIP Slice Processing Path Overview

Figure 9-5 shows in details the internal processing path and output signals to VPDMA for a single VIP Slice. External video source drives the input side of the VIP Slice. Port A[x:y] can be in YUV422 format (A[15:0] in the diagram) or RGB/YUV444 format (A[23:0] in the diagram), depending on the external video input source and configuration options within the VIP_PARSER. Port B[x:y] can be in YUV422 format (B[15:0] in the diagram). When the VIP_PARSER is configured to capture 24bit RGB/444 data, A[23:0] is used and the data path inside VIP must be configured correctly for it. Multiplexer selections and controls shown in Figure 9-5 are described in register VIP_CLKC_VIP0DPS for Slice 0, and register VIP_CLKC_VIP1DPS for Slice 1. The outputs of each VIP Slice drive the VPDMA module, which sends the resulting data to DDR memory.

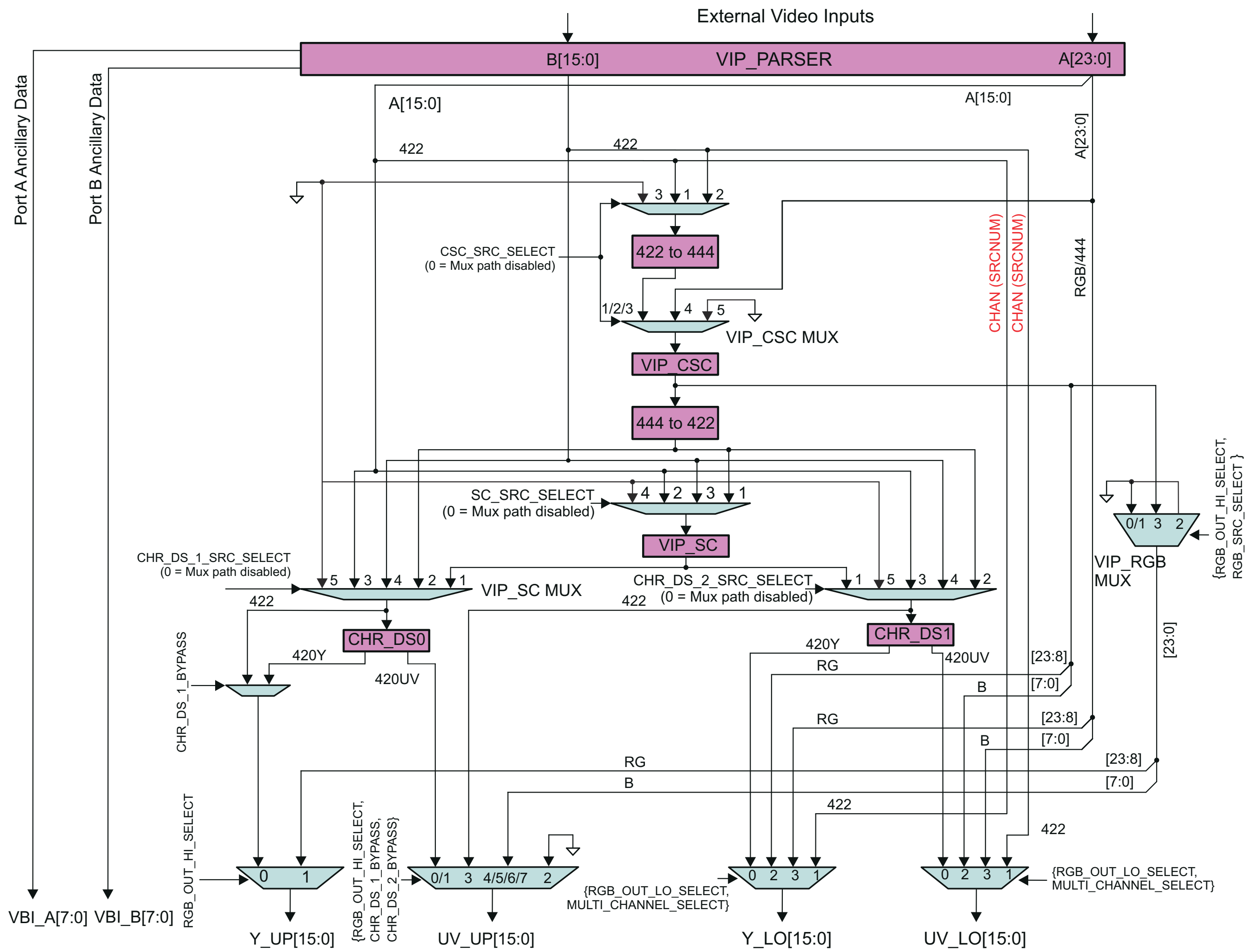


Figure 9-5. VIP Slice Processing Path Block Diagram

Note

Port A of VIP3 module supports up to 16-bit data, and Port B is not used (this applies to both Slice 0 and Slice 1 of VIP3).

Table 9-9 provides summary of the registers controlling the multiplexers within VIP slice processing path.

Table 9-9. VIP Slice Processing Path Control

Multiplexer Control	Register Bit-fields for Slice 0	Register Bit-fields for Slice 1	Description
CSC_SRC_SELECT	VIP_CLKC_VIP0DPS[2:0] VIP1_CSC_SRC_SELECT	VIP_CLKC_VIP1DPS[2:0] VIP2_CSC_SRC_SELECT	VIP CSC Source Selection MUX
SC_SRC_SELECT	VIP_CLKC_VIP0DPS [5:3] VIP1_SC_SRC_SELECT	VIP_CLKC_VIP1DPS [5:3] VIP2_SC_SRC_SELECT	VIP SC_M Source Selection MUX
CHR_DS_1_SRC_SELECT	VIP_CLKC_VIP0DPS[11:9] VIP1_CHR_DS_1_SRC_SELECT	VIP_CLKC_VIP1DPS[11:9] VIP2_CHR_DS_1_SRC_SELECT	VIP Chroma Downsampler 1 Source Selection MUX
CHR_DS_1_BYPASS	VIP_CLKC_VIP0DPS[16] VIP1_CHR_DS_1_BYPASS	VIP_CLKC_VIP1DPS[16] VIP2_CHR_DS_1_BYPASS	VIP Chroma Downsampler 1 Bypass MUX

Table 9-9. VIP Slice Processing Path Control (continued)

Multiplexer Control	Register Bit-fields for Slice 0	Register Bit-fields for Slice 1	Description
CHR_DS_2_SRC_SELECT	VIP_CLKC_VIP0DPS[14:12] VIP1_CHR_DS_2_SRC_SELECT	VIP_CLKC_VIP1DPS[14:12] VIP2_CHR_DS_2_SRC_SELECT	VIP Chroma Downsampler 2 Source Selection MUX
CHR_DS_2_BYPASS	VIP_CLKC_VIP0DPS[17] VIP1_CHR_DS_2_BYPASS	VIP_CLKC_VIP1DPS[17] VIP2_CHR_DS_2_BYPASS	VIP Chroma Downsampler 1 Bypass MUX
RGB_OUT_HI_SELECT	VIP_CLKC_VIP0DPS[8] VIP1_RGB_OUT_HI_SELECT	VIP_CLKC_VIP1DPS[8] VIP2_RGB_OUT_HI_SELECT	VIP HI RGB Output Selection MUX
RGB_OUT_LO_SELECT	VIP_CLKC_VIP0DPS[7] VIP1_RGB_OUT_LO_SELECT	VIP_CLKC_VIP1DPS[7] VIP2_RGB_OUT_LO_SELECT	VIP LO RGB Output Selection MUX
MULTI_CHANNEL_SELECT	VIP_CLKC_VIP0DPS[15] VIP1_MULTI_CHANNEL_SELECT	VIP_CLKC_VIP1DPS[15] VIP2_MULTI_CHANNEL_SELECT	VIP Multi Channel Selection MUX
RGB_SRC_SELECT	VIP_CLKC_VIP0DPS[6] VIP1_RGB_SRC_SELECT	VIP_CLKC_VIP1DPS[6] VIP2_RGB_SRC_SELECT	VIP RGB Output Path Selection MUX

9.4.4.2 VIP Slice Processing Path Multiplexers

9.4.4.2.1 VIP_CSC Multiplexers

The following registers are controling the VIP Color Space Converter (CSC) multiplexers: VIP_CLKC_VIP0DPS[2:0] VIP1_CSC_SRC_SELECT for Slice 0 and VIP_CLKC_VIP1DPS[2:0] VIP2_CSC_SRC_SELECT for Slice 1.

The VIP_CSC block (for each slice within the VIP subsystem) receives data from one the following sources:

- VIP_PARSER Port A Output (422)
- VIP_PARSER Port B Output (422)
- VIP_PARSER Port A Output (RGB)

The default state for this multiplexer is disabled, so there is no VIP_CSC input.

9.4.4.2.2 VIP_SC Multiplexer

The multiplexer for Slice 0 and Slice 1 is controleld by VIP_CLKC_VIP0DPS[5:3] VIP1_SC_SRC_SELECT and VIP_CLKC_VIP1DPS[5:3] VIP2_SC_SRC_SELECT, respectively.

The scaler module (VIP_SC) within the VIP subsystem (for a single slice) receives data from one of the following sources:

- VIP_CSC
- VIP_PARSER Port A Output
- VIP_PARSER Port B Output

The default state for this multiplexer is disabled, so there is no VIP_SC input.

9.4.4.2.3 Output to VPDMA Multiplexers

This section is under development and is included as a placeholder for future updates.

9.4.4.3 VIP Slice Processing Path Examples

The following sections provide VIP Slice data path examples for different types of input data, and describe the corresponding multiplexer configurations. Refer to Table 9-9, *VIP Slice Processing Path Control*, for mapping of the multiplexer controls to registers.

In the block diagrams of the following sections:

- Output A refers to the result of processing Input A data.
- Output B refers to the result of processing Input B data.

9.4.4.3.1 Input: A=RGB, B=YUV422; Output: A=RGB, B=RGB

Tested in single channel embedded and discrete mode.

Input: A=RGB, B=YUV422; Output: A=RGB, B=RGB

Multiplexers settings.

- VIPx_CSC_SRC_SELECT = 2
- VIPx_SC_SRC_SELECT = 0
- VIPx_CHR_DS_1_SRC_SELECT = 0
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 0
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 1
- VIPx_RGB_OUT_HI_SELECT = 1
- VIPx_RGB_OUT_LO_SELECT = 1
- VIPx_MULTI_CHANNEL_SELECT = 1

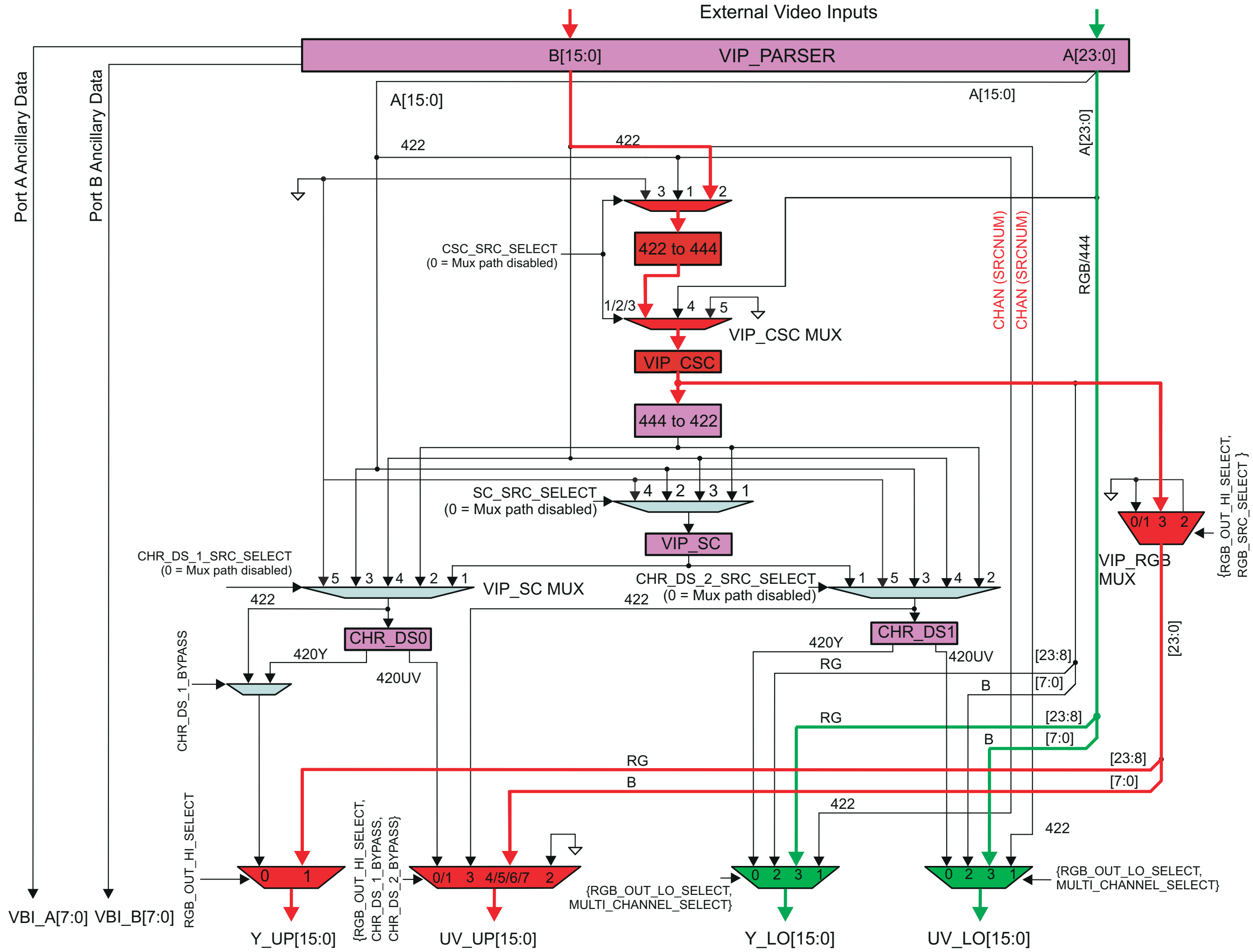


Figure 9-6. Input: A=RGB, B=YUV422; Output: A=RGB, B=RGB

9.4.4.3.2 Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=RGB

Tested in single channel embedded and discrete mode.

Multiplexers settings:

- VIPx_CSC_SRC_SELECT = 2
- VIPx_SC_SRC_SELECT = 2
- VIPx_CHR_DS_1_SRC_SELECT = 0
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 1
- VIPx_RGB_OUT_HI_SELECT = 1
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

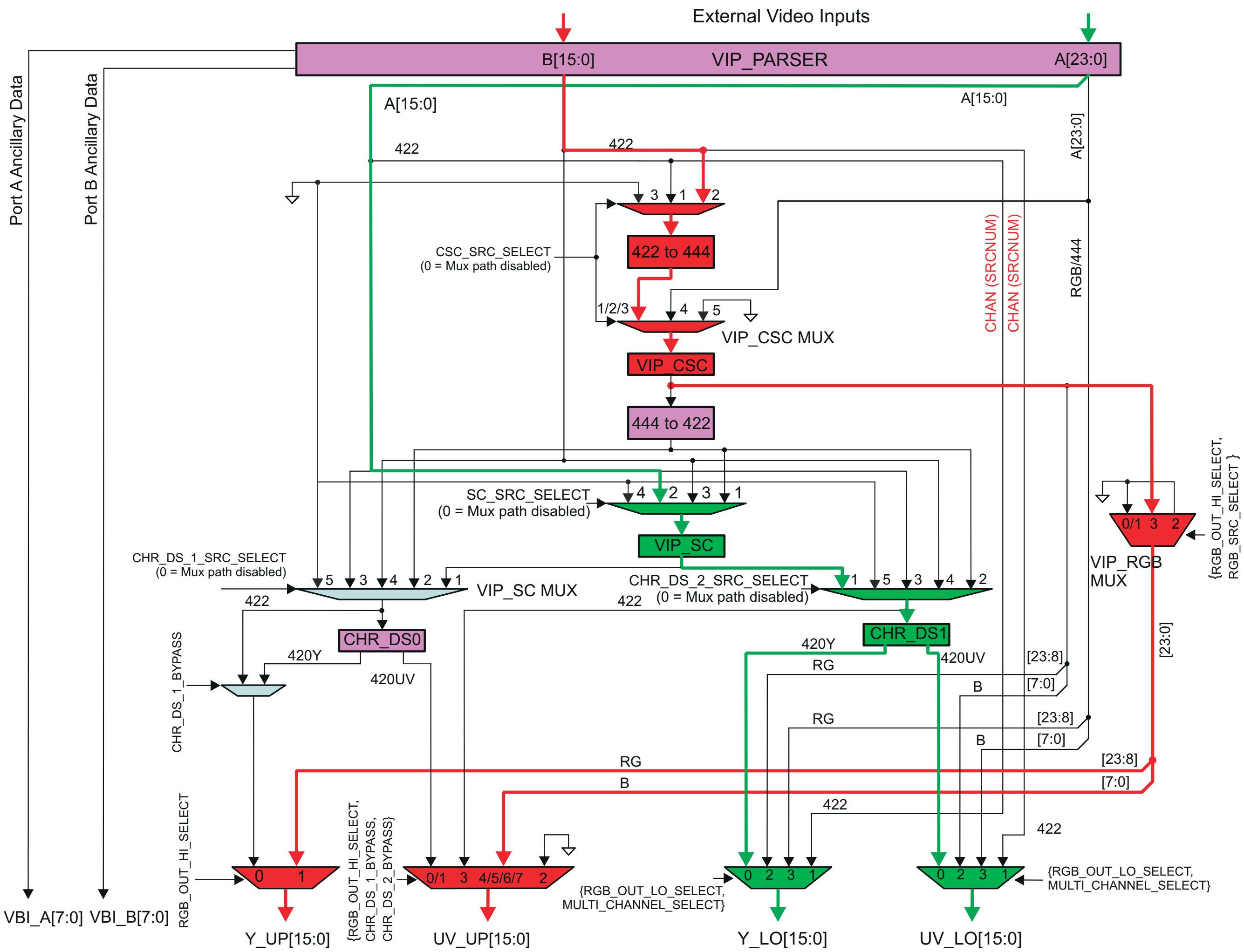


Figure 9-7. Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=RGB

9.4.4.3.3 Input: A=RGB, B=YUV422; Output: A=RGB, B=Scaled YUV420

Tested in single channel embedded and discrete mode and multi-channel mode.

Multiplexers settings:

- VIPx_CSC_SRC_SELECT = 4
- VIPx_SC_SRC_SELECT = 3
- VIPx_CHR_DS_1_SRC_SELECT = 1
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 1
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0
- VIPx_RGB_OUT_LO_SELECT = 1
- VIPx_MULTI_CHANNEL_SELECT = 1

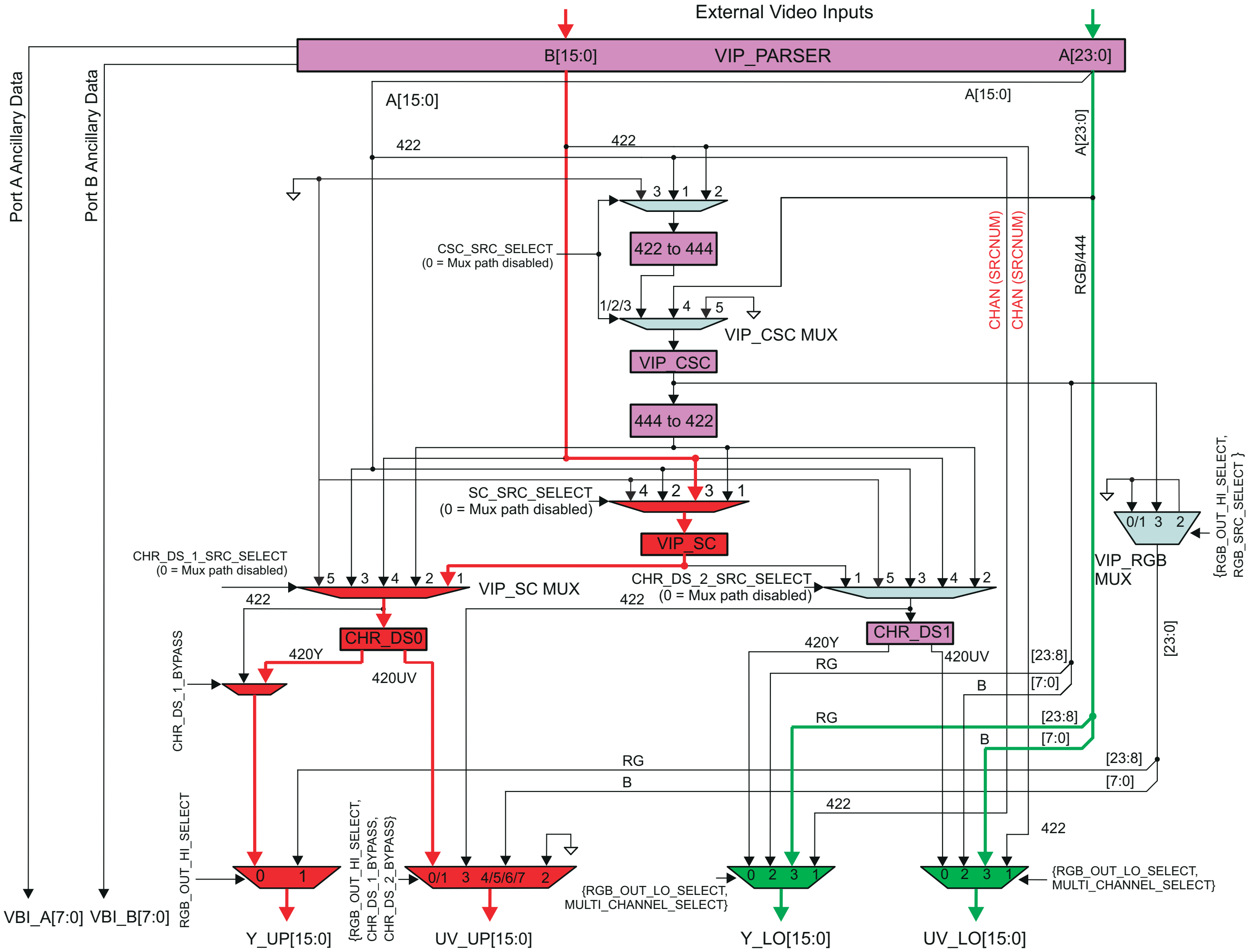


Figure 9-8. Input: A=RGB, B=YUV422; Output: A=RGB, B=Scaled YUV420

9.4.4.3.4 Input: A=YUV444, B=YUV422; Output: A=YUV422, A=Scaled YUV422, B=YUV422

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 4
- VIPx_SC_SRC_SELECT = 1
- VIPx_CHR_DS_1_SRC_SELECT = 2
- VIPx_CHR_DS_1_BYPASS = 1
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 1
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 1

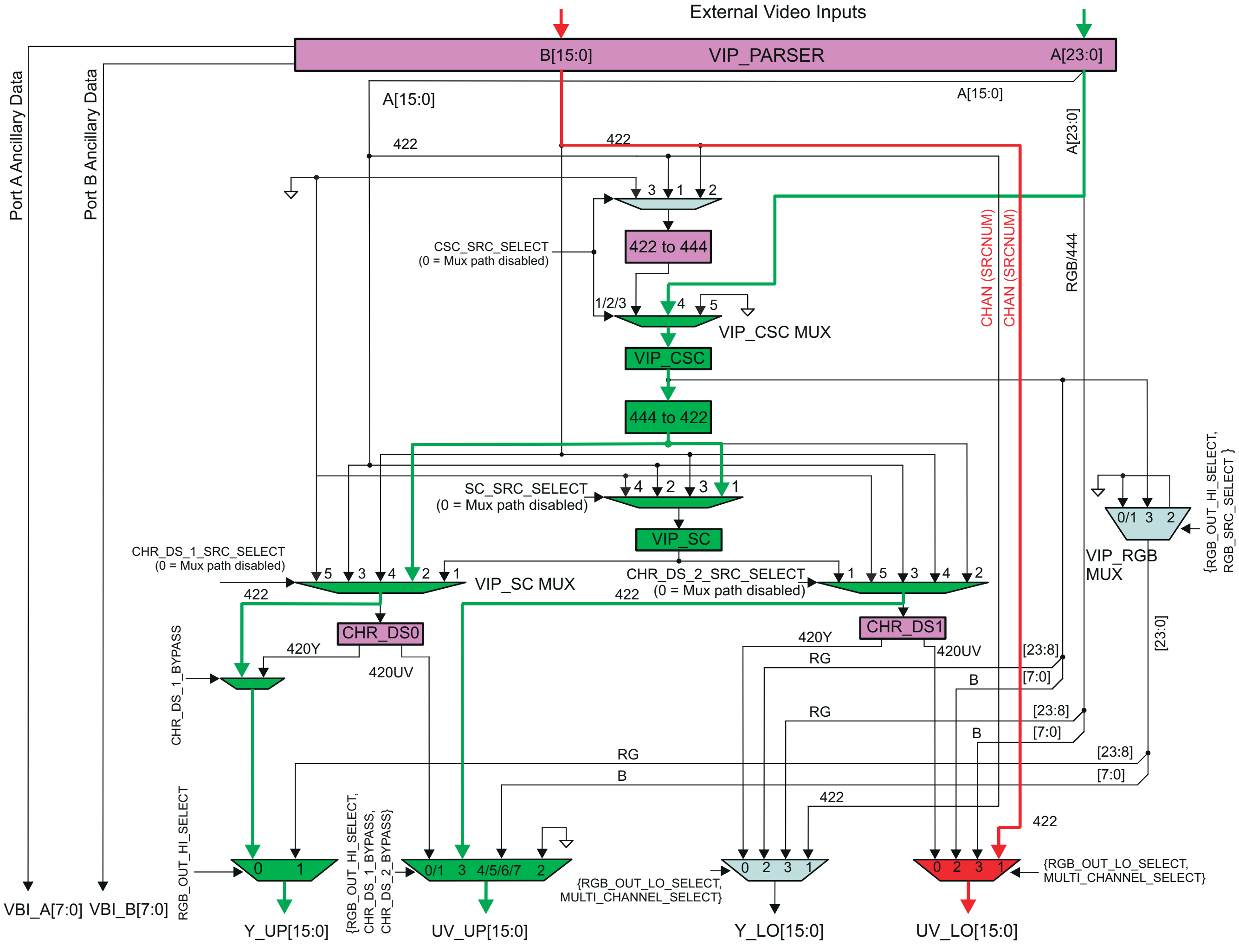


Figure 9-9. Input: A=YUV444, B=YUV422; Output: A=YUV422, A=Scaled YUV422, B=YUV422

9.4.4.3.5 Input: A=YUV444; Output: A=Scaled YUV420, A=YUV420

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 4
- VIPx_SC_SRC_SELECT = 1
- VIPx_CHR_DS_1_SRC_SELECT = 2
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

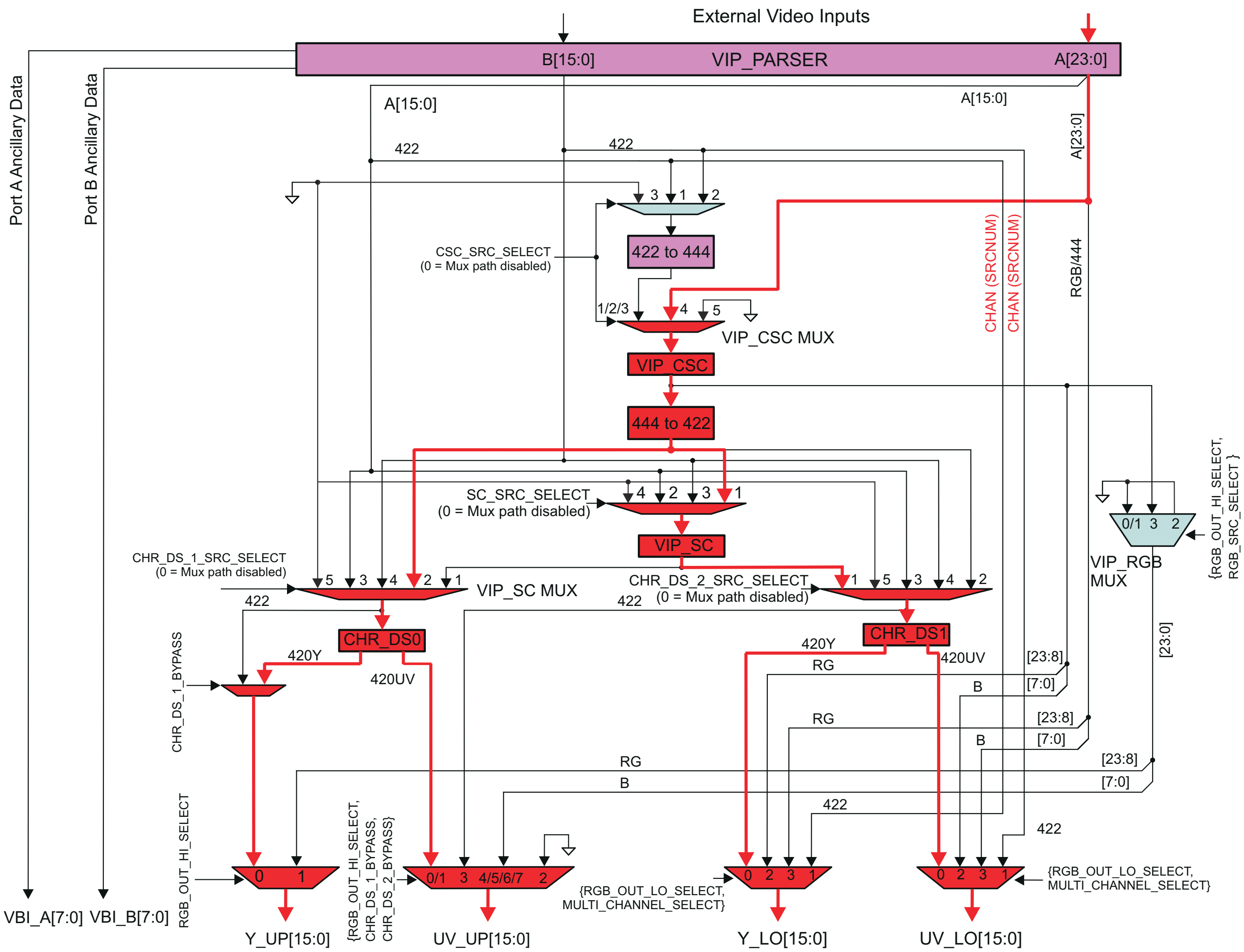


Figure 9-10. Input: A=YUV444; Output: A=Scaled YUV420, A=YUV420

9.4.4.3.6 Input: A=YUV444; Output: A=Scaled YUV420, A=YUV444

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 4
- VIPx_SC_SRC_SELECT = 1
- VIPx_CHR_DS_1_SRC_SELECT = 0
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 1
- VIPx_RGB_OUT_HI_SELECT = 1
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

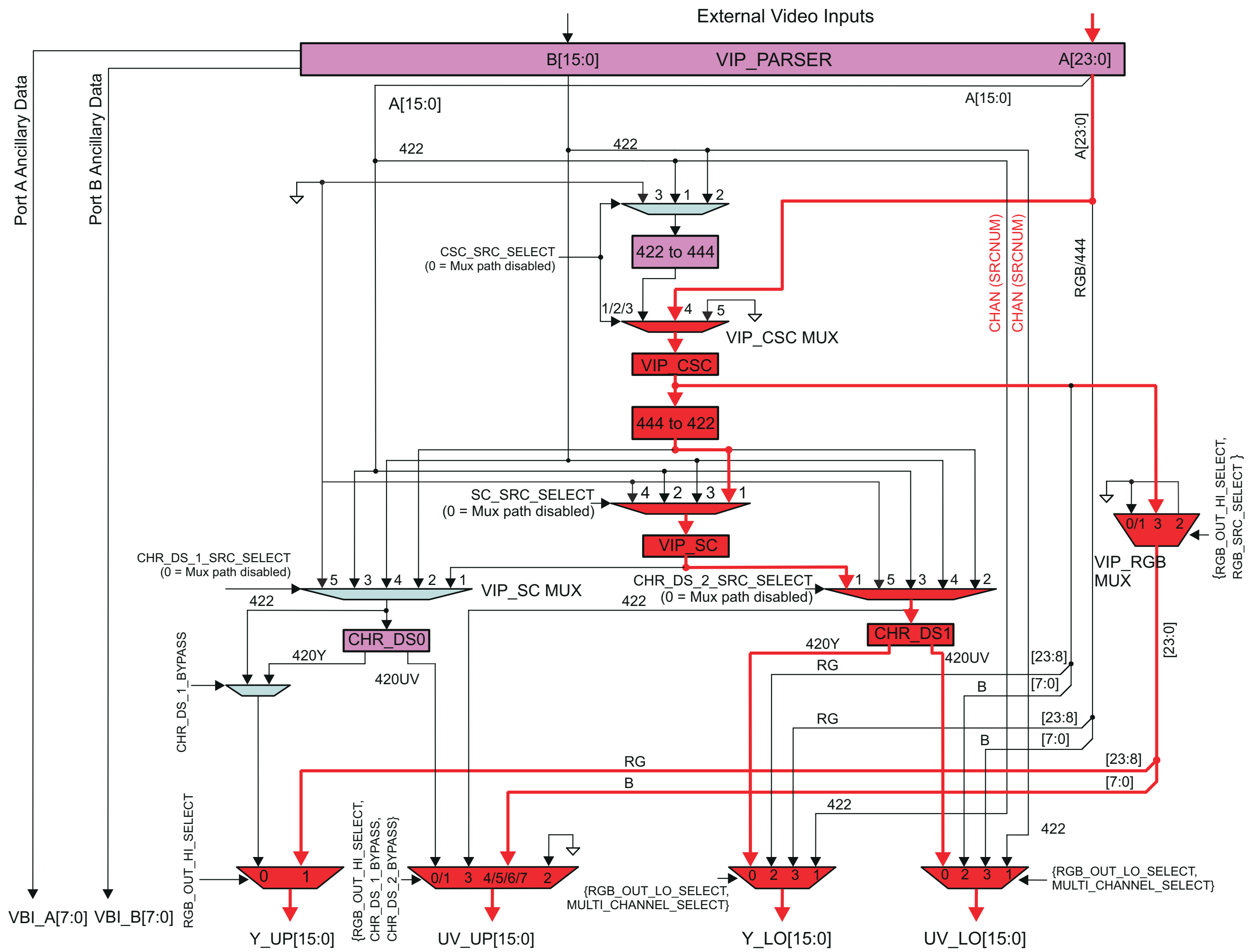


Figure 9-11. Input: A=YUV444; Output: A=Scaled YUV420, A=YUV444

9.4.4.3.7 Input: A=YUV422 8/16; Output: A=Scaled YUV420, A=YUV444

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 1
- VIPx_SC_SRC_SELECT = 1
- VIPx_CHR_DS_1_SRC_SELECT = 0
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 1
- VIPx_RGB_OUT_HI_SELECT = 1
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

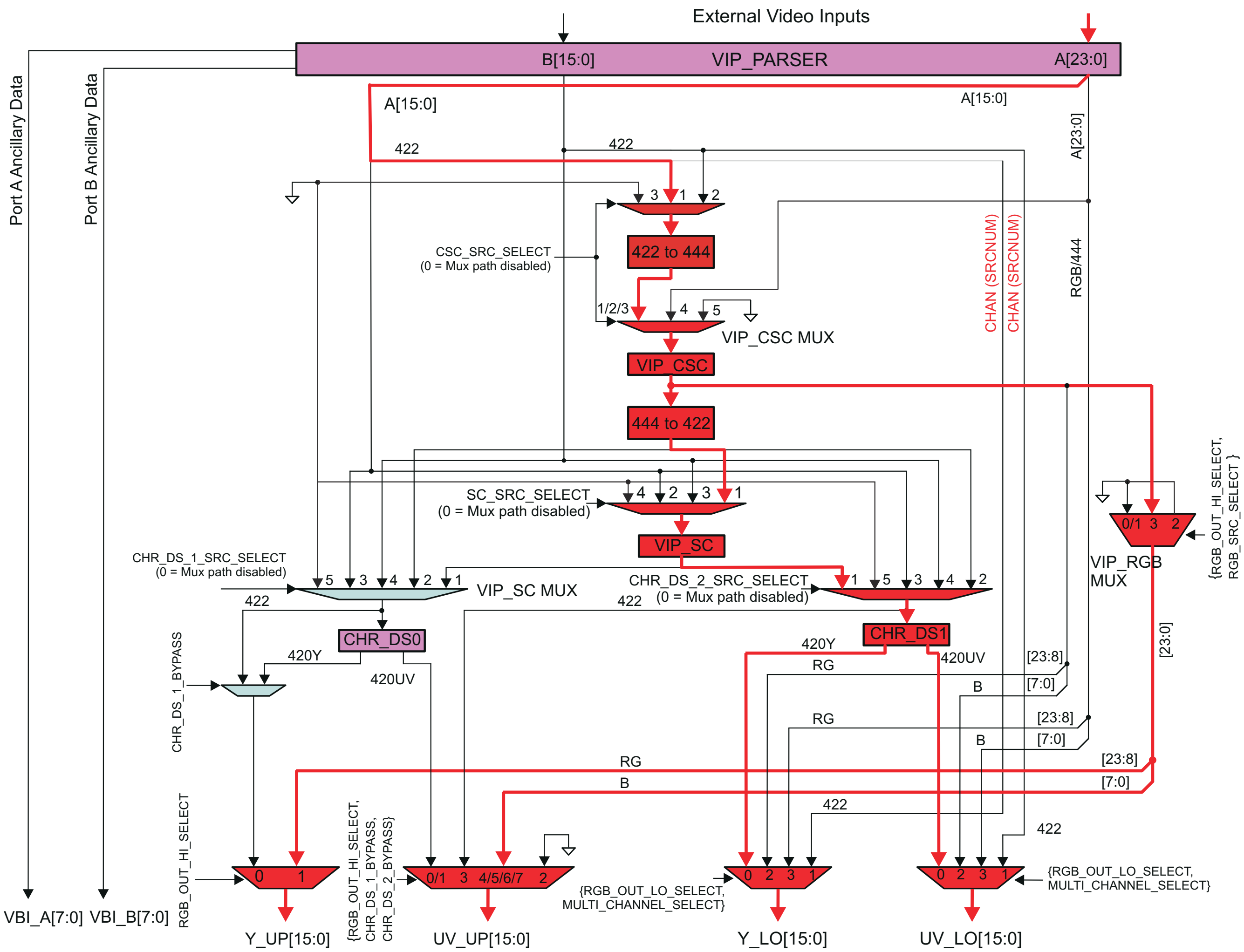


Figure 9-12. Input: A=YUV422 8/16; Output: A=Scaled YUV420, A=YUV444

9.4.4.3.8 Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=YUV420

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 0
- VIPx_SC_SRC_SELECT = 2
- VIPx_CHR_DS_1_SRC_SELECT = 4
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

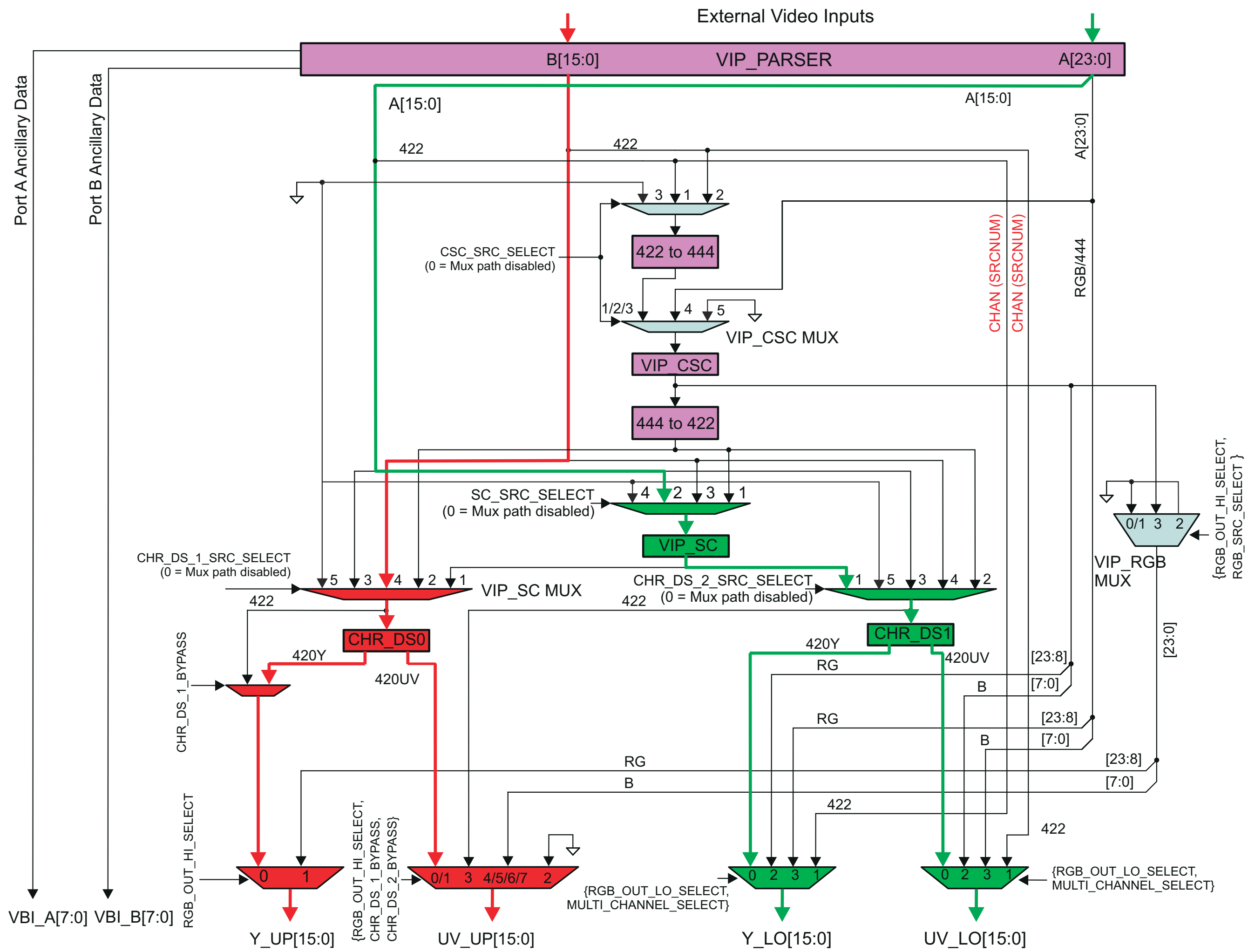


Figure 9-13. Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=YUV420

9.4.4.3.9 Input: A=YUV422 8/16, B=YUV422; Output: A=YUV420, B=YUV420

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 0
- VIPx_SC_SRC_SELECT = 0
- VIPx_CHR_DS_1_SRC_SELECT = 4
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 3
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

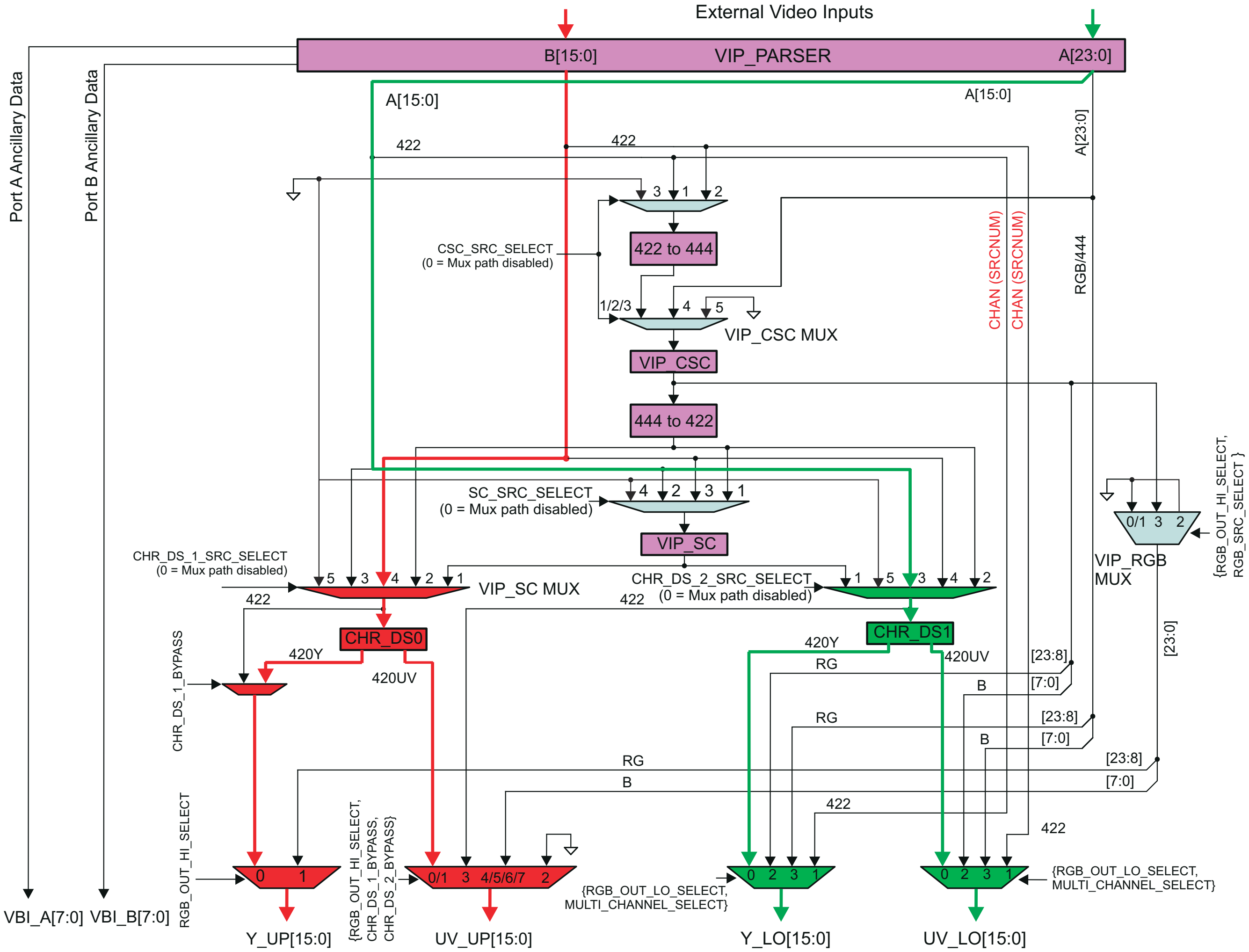


Figure 9-14. Input: A=YUV422 8/16, B=YUV422; Output: A=YUV420, B=YUV420

9.4.5 VIP Parser

The VIP Parser (VIP_PARSER) module is used to capture the external video data into the VIP module.

9.4.5.1 Features

Each VIP module contains two VIP_PARSER modules (one VIP_PARSER per slice).

For a single VIP_PARSER, the video capture functions include:

- Two Pixel Clock Input Domains are supported (Port A and Port B):
  - Each Pixel Clock Input Domain has separate clock and framing signals.
  - Each Pixel Clock Input Domain can support embedded (BT.656/1120 style in /24-bit, or BT.656 in 8-bit) or discrete (BT.601 style) sync.
  - Pixel Clock Input Domain Port B supports one 8-bit input data bus. Port A supports one 24-bit input data bus. At device level, the same device pads may be shared between Port A and Port B. For more information, see the multiplexing characteristics in device Data Manual.
- Embedded Sync data interface mode supports single or multiplexed sources;
- Discrete Sync data interface mode supports only single source input;
- The two Pixel Clock Input Domains can be individually configured in any combination of Embedded or Discrete Sync;
- Vertical Ancillary Data capture is supported for each input source;
- A maximum of 8 + 1 (8 normal line sources + 1 split-line source) multiplexed sources are supported for a single Pixel Clock Input Domain using TI Line Mux Mode;
- Multiplexed data can only appear in embedded sync mode;
- Where possible, blanking pixels that may contain embedded vertical ancillary data will be stored in a dedicated buffer per each video source;
- Optional selection of channel (Luma or Chroma, or both) from which Vertical Ancillary data is extracted for YUV422 source;
- For RGB source, Vertical Ancillary data can be found in one of the R, G, or B channels. The VIP_PARSER can select the channel from which Vertical Ancillary data is extracted;
- Ancillary Data can appear in the Horizontal Blanking as well as the Vertical Blanking. Typically, only Vertical Blanking Ancillary Data is captured. However, Horizontal Blanking Ancillary Data can be captured as well using HSYNC style discrete sync capture mode;
- Video up to WUXGA (1920 × 1200) can be supported using Port A in 16-bit or 24-bit mode.

9.4.5.2 Repacker

The Repacker module rearranges the input bit ordering of the 24-bit data bus on Port A of each VIP slice. This module allows external input data to be presented to the datapath such that various data packing formats can be achieved in memory. As shown in Figure 9-15, a Repacker exists for each 24-bit input port.

The Repacker module is a simple multiplexer that serves to move input bits to different locations on the its output bus to VIP Parser. Figure 9-15 shows the supported bytelane swapping modes corresponding to different VIP_XTRA_PORT_A[30:28] REPACK_SEL settings.

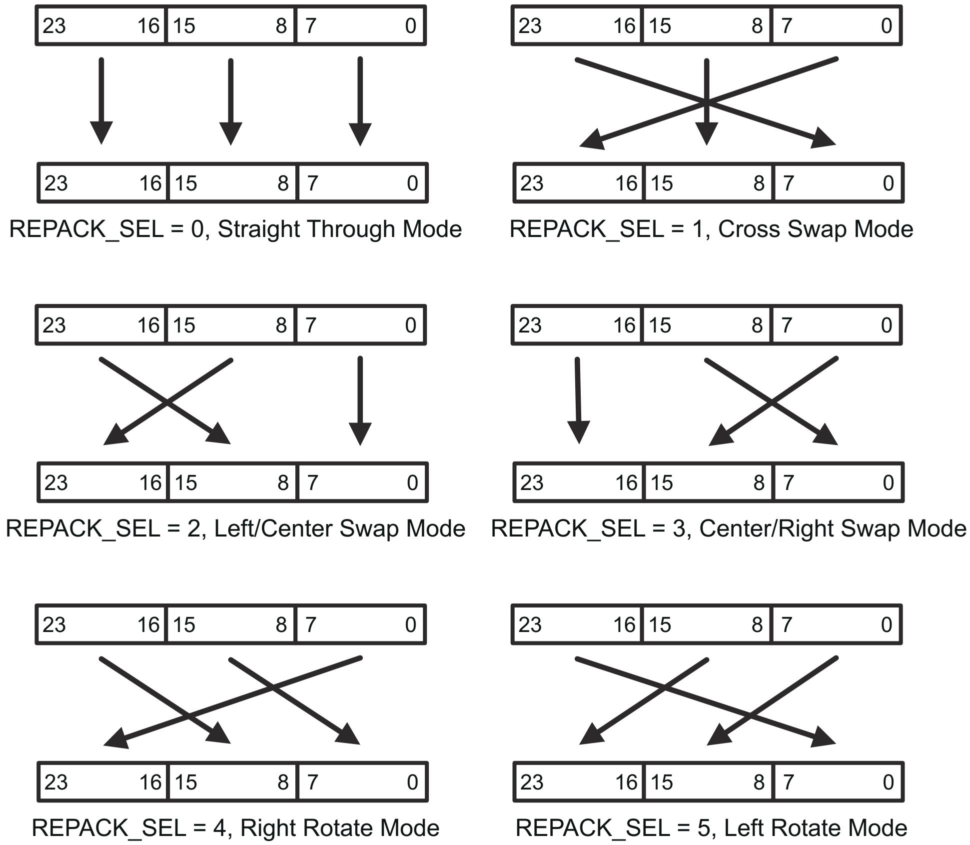


Figure 9-15. Bytelane Swapping Modes

16-bit RAW data entering the VIP subsystem is packed as a contiguous input bus from bits 15 to 0. This 16-bit RAW input must be remapped to the RGB565 format, so that it can be saved to DDR memory properly, because the VPDMA does not support the RAW16 input format natively. Instead, the RAW 16 format is first remapped as RGB565 data and then given to the VPDMA. Figure 9-16 describes the VIP_XTRA_PORT_A[30:28] REPACK_SEL = 6 option to remap a contiguous [15:0] RAW input data bus to a RGB565 compliant output bus. This RAW16 data will use RGB565 data types in the VPDMA Data Descriptors.

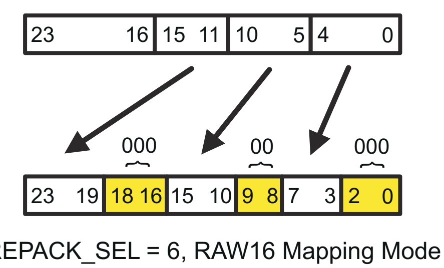
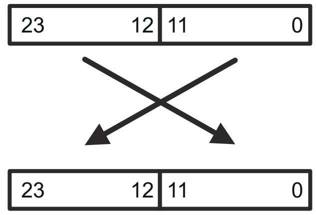


Figure 9-16. RAW16 to RGB565 Mapping

Figure 9-17 describes the mux configuration where 12 bit components are swapped. This mode may be useful when the input data is 12 bit per component YUV422 and is sent directly to memory.



REPACK_SEL = 7, RAW12 Swapping Mode

Figure 9-17. RAW12 Swap

Note

There is no repacker for the 8-bit input port (Port B of each VIP slice).

The RAW16 and RAW12 mapping modes do not work for embedded sync streams.

9.4.5.3 Analog Video

A digital interface stream is based on analog video. The waveform for a line of NTSC analog video is shown in Figure 9-18.

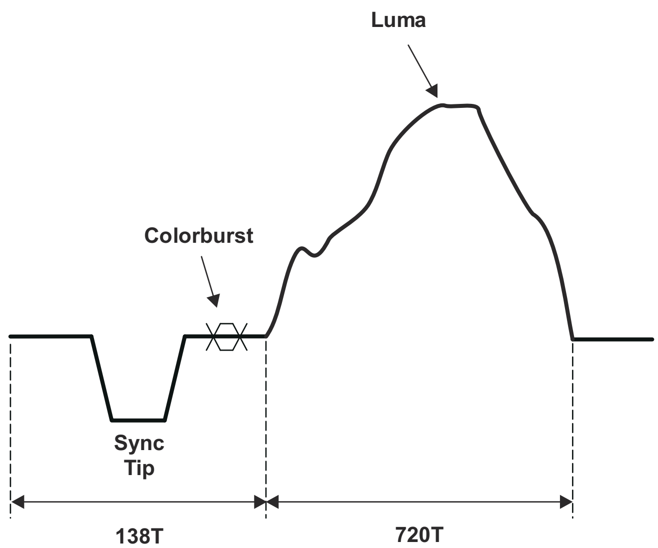


Figure 9-18. NTSC Analog Video Waveform for One Horizontal Line

T is a time constant. For NTSC, T=1/13.5 MHz = 74ns.

9.4.5.4 Digitized Video

Digitized video is based on scan lines in found in analog video. BT.601 uses various sync signals to specify when a new field and a new line starts. BT.656 and BT.1120 uses sync words embedded in the data stream to specify start of field and start of line.

An image can be digitized into regions shown in Figure 9-19.

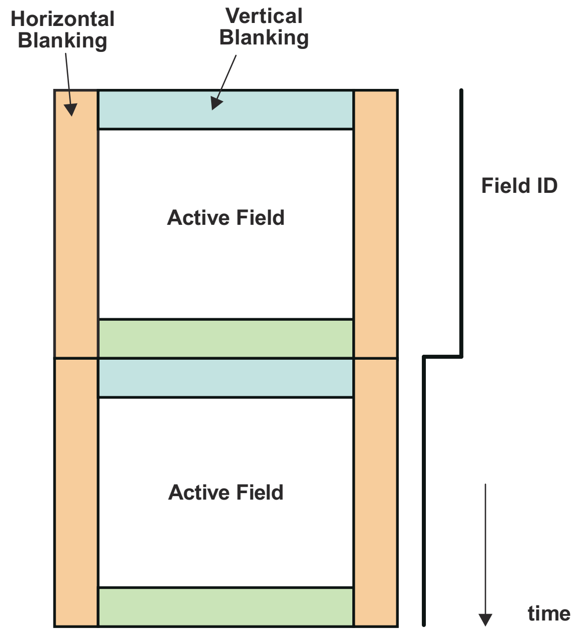


Figure 9-19. Digitized Video

With the capability to encode sync words inside the data stream, there is more flexibility for adding non-video related data, called Ancillary Data. Also, code words embedded in the digital stream can be used as a type of identifier for multiplexing several sources of video into one data stream.

Figure 9-20 shows End-of-Active-Video (EAV) and Start-of-Active-Video (SAV) code words added to a video transmission. The period between the EAV and SAV is equivalent to Horizontal Blanking. The period between the SAV to the next EAV is active video or vertical blanking.

In the BT.656 or BT.1120 embedded code word scheme, three bits of the EAV/SAV code word are important: F (field), H (horizontal blanking), and V (vertical blanking).

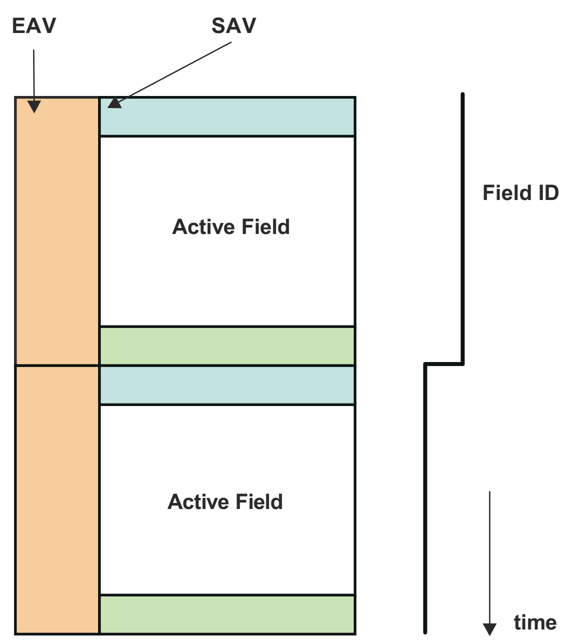


Figure 9-20. Code Word Embedded Video Format

Figure 9-21 shows the values of F, V, and H flags at different locations in the picture. The Field flag represents the state of the Field ID for the picture. For progressive frames, F is always ‘0.’ The V flag specifies vertical blanking areas. The H flag specifies horizontal blanking portions of the picture.

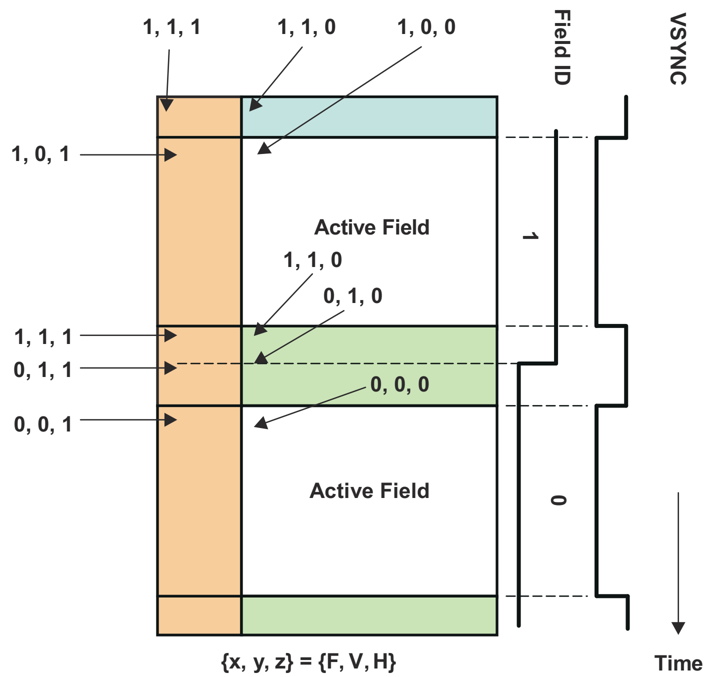


Figure 9-21. Digitized Video with F, V, and H Flags in EAV/SAV

9.4.5.5 Frame Buffers

The VIP/VPDMA support Frame Buffers in DDR memory for Active Video and Ancillary Data.

4:2:2 data is always saved in packed pixel buffers.

4:2:0 data is saved in Planar Luma buffers and Planar CbCr pair buffers.

A Luma Frame Buffer is a Planar storage area. Each line is the width in pixels (1Byte/pixel) of the output picture size format. The frame buffer contains the number of active video lines in the output picture size format.

A Chroma Pair Frame Buffer is Planar storage of CbCr pixel pairs, with each pixel being a byte. For 4:2:0 storage, N lines in the output picture active video results in N/2 lines of CbCr pairs being stored.

The Ancillary Data buffer is different than the Active Video Frame Buffers. The Ancillary Data buffer only stores Vertical Blanking Ancillary Data. The number of lines in the Ancillary Data buffer is the same as the number of Vertical Blanking lines. Typically, only one channel is extracted from the Vertical Blanking data, so the width of the Ancillary Data buffer is the same as the width of the Luma Buffer.

In 8-bit input mode, it is possible for both Luma and Chroma sites to be extracted for Vertical Ancillary data. Each color component is strobed on separate input clock cycles. In this case, the line width of the Ancillary data is twice the Luma line width of the picture. Both Luma and Chroma sites cannot be extracted for 16-bit input mode because both Luma and Chroma are sent on the same input clock cycle and the Ancillary port to the VPDMA VPI is only 8 bits wide.

Figure 9-22 shows how the planar data regions are stored in DDR memory. The vertical blanking data is stored in a set of Planar Buffers. Note that the bottom of the Vertical Ancillary Data from the previous field or frame is stored in the same buffer as the top of the Vertical Ancillary Data from the current field or frame.

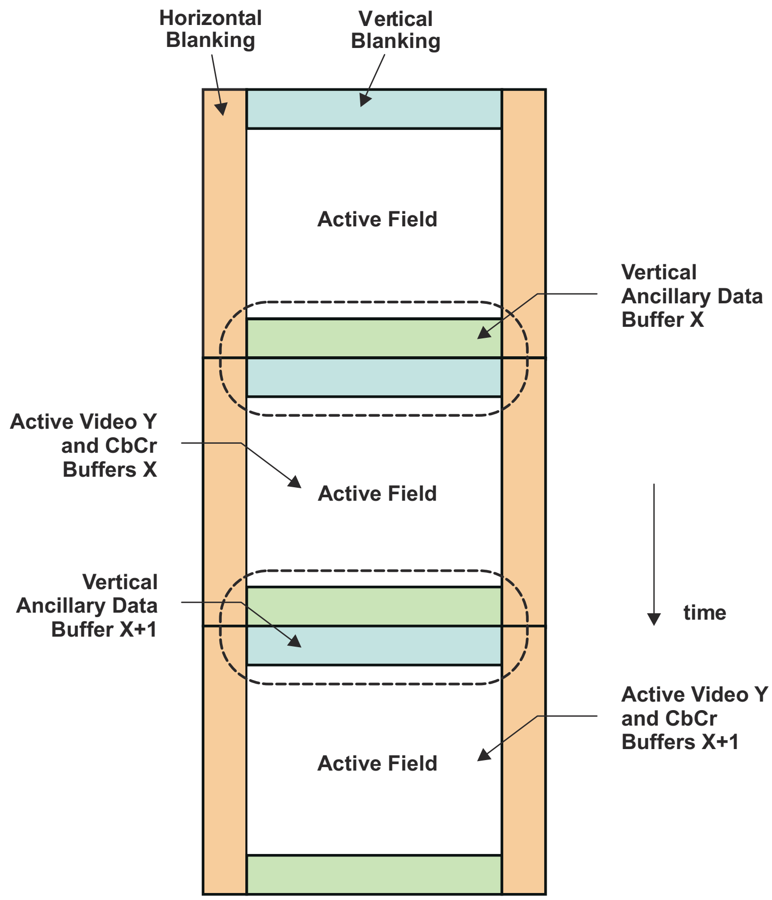


Figure 9-22. Planar Buffer Storage Description

The Luma representing Active Video is stored in a set of Planar Buffers. The CbCr Chroma Pairs are stored in a set of Planar Buffers.

9.4.5.6 Input Data Interface

This section describes how the data (luma and chroma data for YUV422 format capture and R,G, and B data for RGB888 format capture) is muxed for the various interface modes.

9.4.5.6.1 8b Interface Mode

In 8-bit data interface mode, the input pixels are multiplexed according to Figure 9-23. The Chroma Format is 4:2:2. Sites with Cb/Cr pixels are known as Chroma sites. Those sites with Y pixels are known as Luma sites.

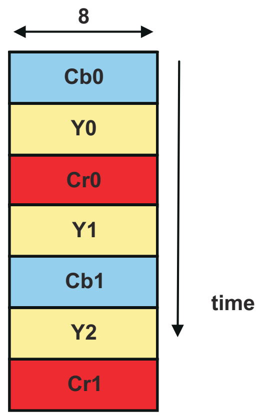


Figure 9-23. 8-bit Interface Discrete Sync Pixel Multiplexing

9.4.5.6.2 16b Interface Mode

In 16-bit interface mode, Luma is on 8 MSB bits of the data bus and Cb/Cr chroma pixels alternate on the other 8 bits of the data bus as shown in Figure 9-24.

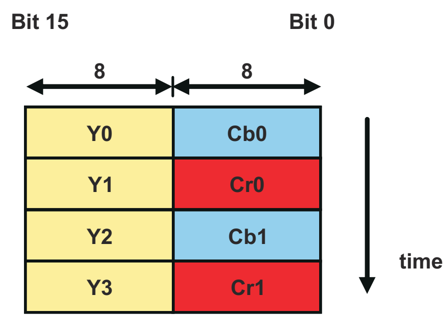


Figure 9-24. 16-bit Interface Discrete Sync Pixel Multiplexing

9.4.5.6.3 24b Interface Mode

RGB or YUV444 data is sent in 24b Interface Mode. The three components are packed into the data bus and sent to the VPDMA. The 24-bit Luma VPI client to the VPDMA carries all three components. This data is saved to the DDR in packed mode. That is, the three components are not separated by hardware. The 24-bit data bus is shown in Figure 9-25.

Ancillary data is saved in the Ancillary Data buffer. The VIP_PORT_A[5:4] CTRL_CHAN_SEL and VIP_PORT_B[5:4] CTRL_CHAN_SEL configuration register is used to select whether the ancillary data is from the R/Y, G/U, or B/V channels.

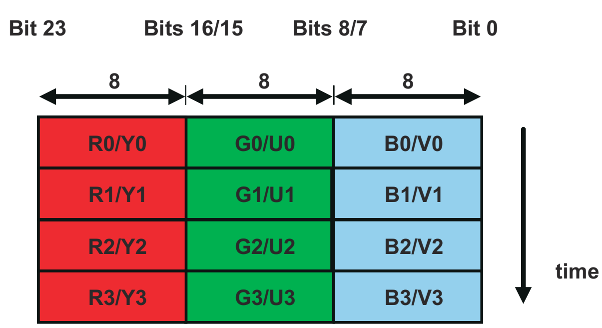


Figure 9-25. 24b Interface RGB Discrete Sync

In 24b interface mode YUV4:4:4 data is also supported. The flow for YUV4:4:4 data is the same as RGB data (Figure 9-25). Bits [23:16] of the input data bus are placed on bits [23:16] of the data bus going to VPDMA, bits [15:8] of the input bus are placed on bits [15:8] of the data bus going to VPDMA and finally bits [7:0] of the input data bus, are placed on bits [7:0] on the bus going to VPDMA.

9.4.5.6.4 Signal Relationships

A digital representation of video can be realized by using HSYNC and VSYNC signals to identify frame start and line start. Suppose HSYNC and VSYNC are active high, Figure 9-26 shows the general relationship of these signals.

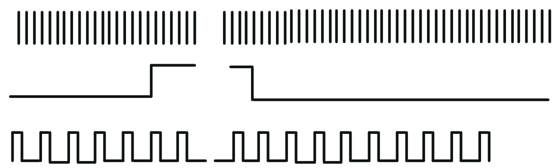


Figure 9-26. Discrete Sync Signals

Every PIXCLK cycle carries either an active pixel or a blanking pixel. VSYNC pulses between two fields (or frames, in the case of progressive video). HSYNC pulses to signify the beginning of every line. An ACTVID signal can be used as a data valid to specify active video.

Discrete Sync cannot be used with any multi-camera multiplexed stream inputs. In the device, if Port A is configured for 24-bit discrete sync, then Port B must be disabled since there are no more data input pins left over for Port B.

If Port A is not 24 bits, then the 8-bit Port B can be configured and enabled for either discrete or embedded sync.

9.4.5.6.5 General 5 Pin Interfaces

Discrete Sync signal handling varies among different sending devices. The information that must be conveyed includes the pixel data value, field ID, horizontal blanking, and vertical blanking. Many devices can be configured to adjust the timing of the signals relative to each other.

In this section, DATA will be depicted as 8 bits. However, discrete sync does optionally support 16-bit and 24-bit data input. Type 1 is named after a generic five pin interface between the sending and receiving devices.

In Figure 9-27, P0 represents the first pixel in the horizontal blanking interval following the last vertical blanking line of the previous field or frame. HSYNC specifies the horizontal blanking region and VSYNC specifies that the P0 pixel is in the vertical sync area. HSYNC can be a strobe that is active one or more cycles and can deassert before the actual end of horizontal blanking or HSYNC may be active for the full duration of horizontal blanking.

Likewise, VSYNC can be a strobe that is active one or more cycles and can deassert before the actual end of vertical blanking or VSYNC may be active for the full duration of vertical blanking.

FID can change at this pixel or it may change later. For interlaced source, though, the FID will be inverted for this pixel at the same time point in the next field. So, it does not really matter when FID is captured. Many sending devices allow the location of FID changes to be programmable.

In this diagram and all others in this document, the active polarities of the interface signals can be either high or low. For the sake of uniformity in this document, all polarities are drawn active high. Also, different vendors have different datasheet names for the interface signals.

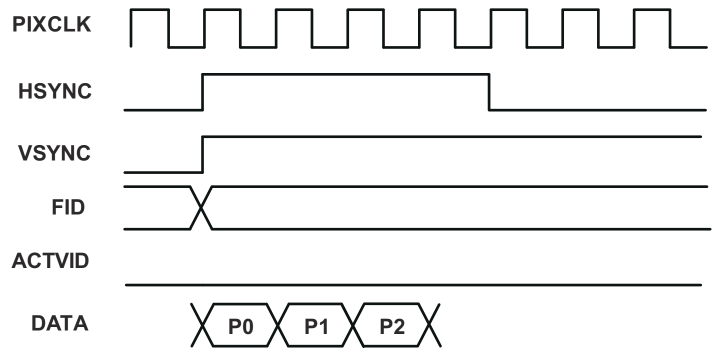


Figure 9-27. Type 1, First Horizontal Blanking Pixel

Figure 9-28 shows the P0 pixel being the first Chroma Channel data value in the Vertical Ancillary Data region. HSYNC is definitely de-asserted by now since P0 is no longer in horizontal blanking. ACTVID may or may not be active for Vertical Ancillary Data. Some devices consider these pixels to be Active (as in non-horizontal blanking). Other devices consider only video to be ACTIVE Video.

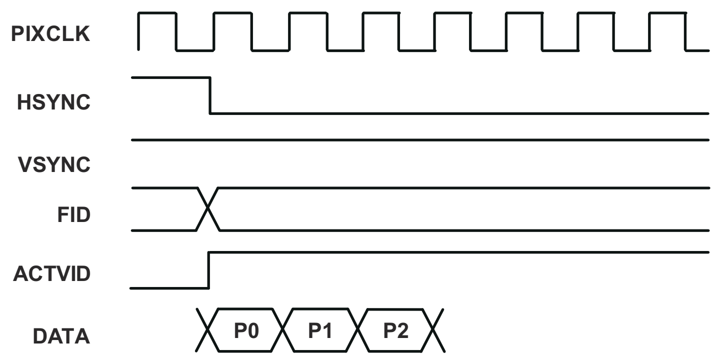


Figure 9-28. Type 1, First Vertical Ancillary Data Pixel

Following the vertical blanking region, the video portion of the field or frame starts. Figure 9-29 shows the horizontal blanking area in this video portion of the field or frame. P0 is the first pixel in the horizontal blanking. HSYNC is active for one or more pixel clocks. VSYNC is inactive in this video area. FID can change here. ACTVID is low since P0 is horizontal blanking.

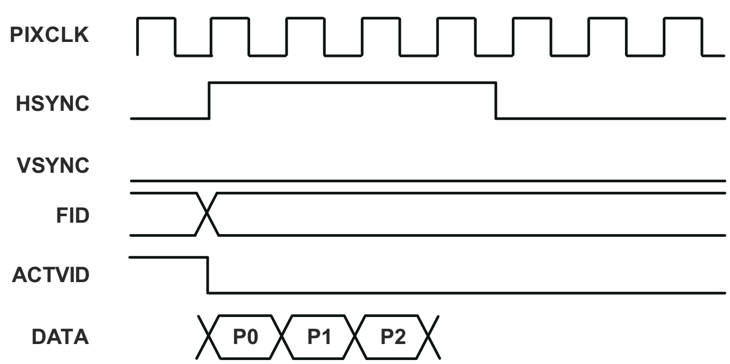


Figure 9-29. Type 1, Horizontal Blanking in Video Region

In Figure 9-30, P0 represents the first Chroma pixel in the Active video line. HSYNC is inactive, since P0 is in the active video region. Likewise, VSYNC is inactive. FID may or may not change here. ACTVID is high to signal capturing of video pixels.

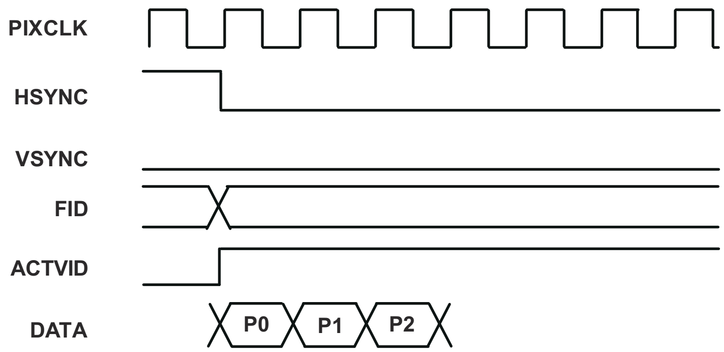


Figure 9-30. Type 1, First Video Pixel

9.4.5.6.6 Signal Subsets—4 Pin VSYNC, ACTVID, and FID

A sending device may use only a subset of the signals described in Section 9.4.5.6.4. The sending device just needs to convey important signals required to capture the field or frame. It can be shown that various selections of four pins can be used to satisfy all Type 1 conditions.

Three pins, VSYNC, ACTVID, and FID, plus a pixel clock can be used to support discrete sync. VSYNC would bump the capture buffer. An inactive to active level of ACTVID specifies a line of data to capture. FID determines the field ID polarity. The scenario in which the sending device wants the receiving end to capture Vertical Ancillary Data using 4-pin signaling is shown in Figure 9-31.

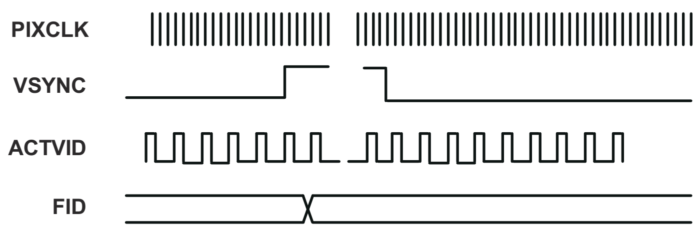


Figure 9-31. 4-Pin Reduced ACTVID Signaling with Vertical Ancillary Data

Figure 9-32 describes the case using the 4-pin interface in which the sending device does not send Vertical Ancillary Data.

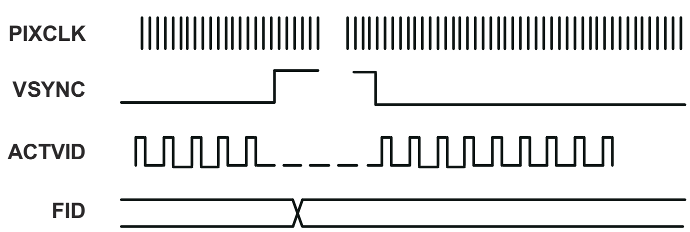


Figure 9-32. 4-Pin Reduced ACTVID Signaling with No Vertical Ancillary Data

9.4.5.6.7 Signal Subsets—4 Pin VSYNC, HSYNC, and FID

In this style of Discrete Sync, as shown in Figure 9-33, four pins are used including the Pixel Clock. HSYNC signals the beginning of the line. All data in the line is captured, including Horizontal Blanking Data. In fact, this signaling mode is the only one which allows Horizontal Blanking Data to be captured.

Of course, by capturing the horizontal blanking pixels in the frame buffers, there is no way to be certain exactly where the blanking ends and the active video starts. One would have to rely solely on video format specs to find the active video inside the frame buffer.

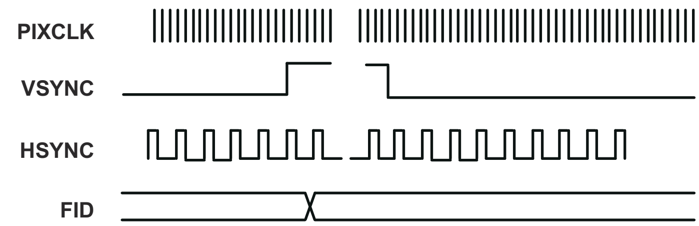


Figure 9-33. 4-Pin Reduced HSYNC Signaling with Vertical Ancillary Data

9.4.5.6.8 Vertical Sync

Vertical Sync is used to indicate lines that are in the vertical blanking interval. The VSYNC also separates fields or frames. To be spec compliant, VSYNC should be active for a few lines at the bottom of the picture. The exact number of vertical blanking lines at the bottom depends on the specification for the picture format (480i, 480p, 720p, 1080i, 1080p).

Likewise, the number of vertical blanking lines at the top of the picture depends on the video format specification corresponding to the incoming picture.

In the VIP_PARSER, lines associated with an Active VSYNC are stored in the vertical ancillary data buffers using the ANC VPI port to the VPDMA. Lines without an Active VSYNC are stored in the active video Luma and Chroma-pair buffers using the Y and UV VPI ports, respectively, to the VPDMA.

When using HSYNC signaling instead of ACTVID, the VSYNC signal may be derived from an analog source such as an NTSC/PAL decoder. In this case, VSYNC may not transition on the exact cycle as HSYNC. Thus, the VIP_PARSER supports a window region around HSYNC in which VSYNC transitions will be detected. A VSYNC transition occurring within the window is identified the same way as if VSYNC transitioned on the same cycle as HSYNC going active.

The window is defined by a pre-window, which is determined by the VIP_PORT_A[21:16] FID_SKEW_PRECOUNT and VIP_PORT_B[21:16] FID_SKEW_PRECOUNT registers. There is also a post-window that is defined by VIP_PORT_A[29:24] FID_SKEW_POSTCOUNT and VIP_PORT_B[29:24] FID_SKEW_POSTCOUNT for port B. Note that although the configuration registers are named FID_SKEW, they are also used for defining the VSYNC transition window. The window region definition is shown in Figure 9-34.

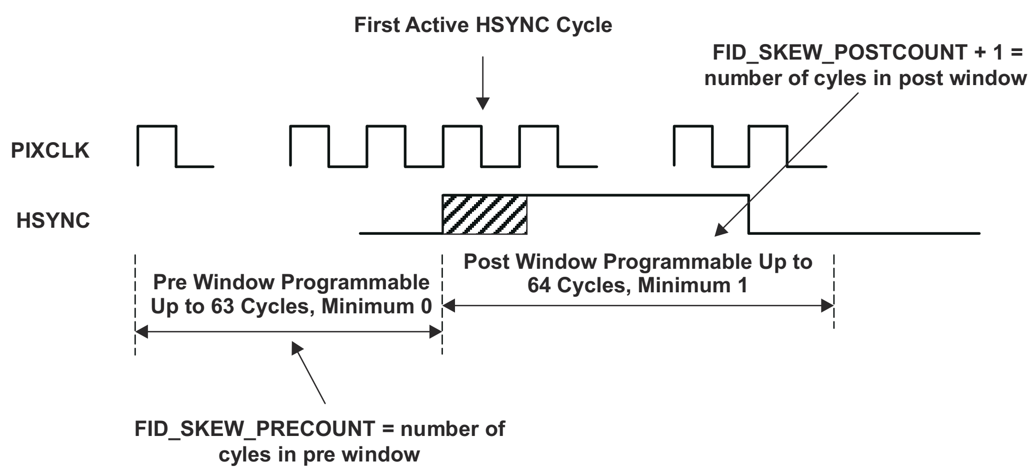


Figure 9-34. VSYNC Pre and Post Window

The results of VSYNC behavior in the transition window are shown in Figure 9-35. A low to high transition in the window is equivalent to VSYNC going low to high on the active HSYNC cycle.

Likewise, a high to low transition in the window is equivalent to VSYNC going high to low on the active HSYNC cycle.

Two transitions of VSYNC within the VSYNC window is not allow and is undefined behavior.

If VSYNC is high throughout the transition window, then the HSYNC line is in vertical blanking.

If VSYNC is low throughout the transition window, then the HSYNC line is in active video.

Note that VSYNC skew generally only applies to input signals that have been sampled from an analog source, as in a NTSC/PAL decoder. If the VSYNC is a VBLANK-type signal or if the sending device is another all-digital IC, then the VSYNC signal does not have a skew since VSYNC will be aligned to HSYNC. In this case, setting FID_SKEW_PRECOUNT = ‘0’ and FID_SKEW_POSTCOUNT = ‘0’ (within PORT_A and PORT_B registers) defines a minimum size window which will capture the value of VSYNC on the same cycle that HSYNC goes active.

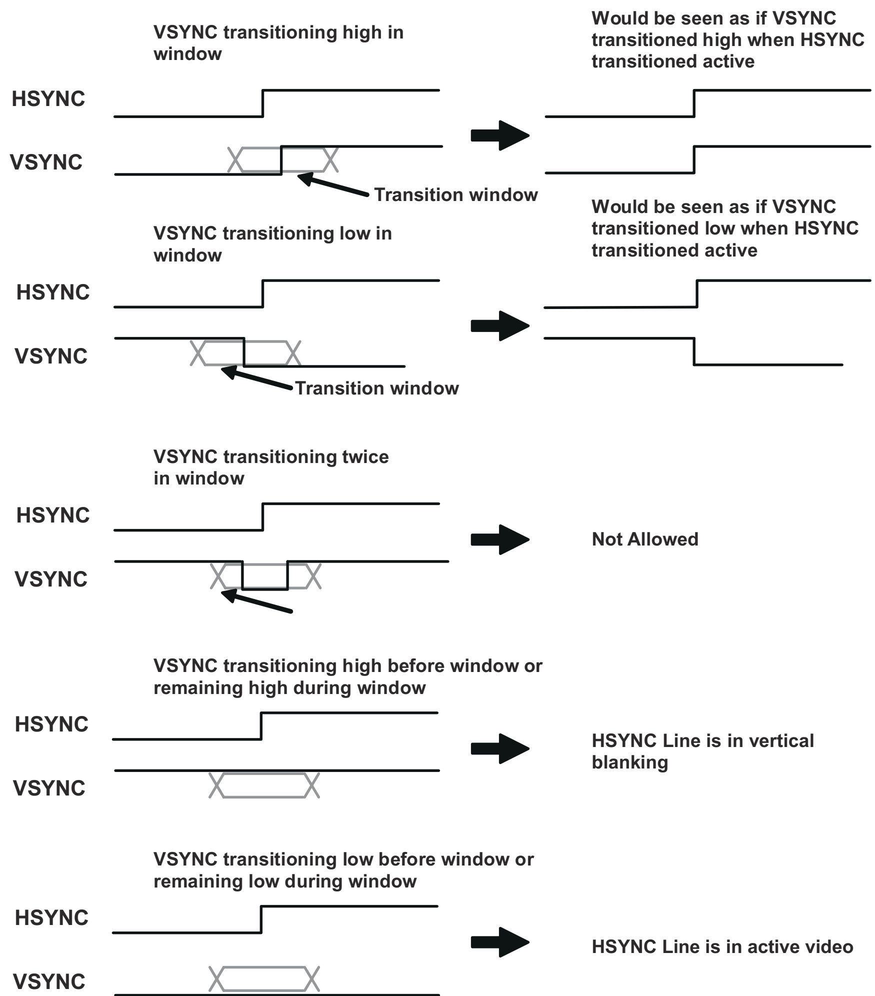


Figure 9-35. VSYNC Equivalence When Using Transition Window

9.4.5.6.9 Field ID Determination Using Dedicated Signal

For Progressive Source, FIELD ID is always ‘0.’

For Interlaced Source, FIELD ID needs to be extracted consistently.

In some cases, vertical sync is active on the first pixel of a line in the vertical blanking period and it stays active until the last line in the vertical blanking period.

However, the pixel where the FIELD ID signal transitions can be quite variable and depends on the external chip driving the VIP_PARSER. Many parts that generate digitized raw video have a programmable feature to specify when FIELD ID changes. FIELD ID is valid at the same point for every field. That is, if FIELD ID is read at one particular place in a field, the polarity of the signal will be reversed at the same location in the next field. So, FIELD ID can be corrected with a programmable polarity configuration bit FID_POLARITY (within VIP_PORT_A and VIP_PORT_B registers) that is XOR'ed with the captured value.

For discrete sync mode, FIELD ID will be registered on the first active pixel capture cycle of each line in both styles of HSYNC and ACTVID usage as specified in Figure 9-36 and Figure 9-37.

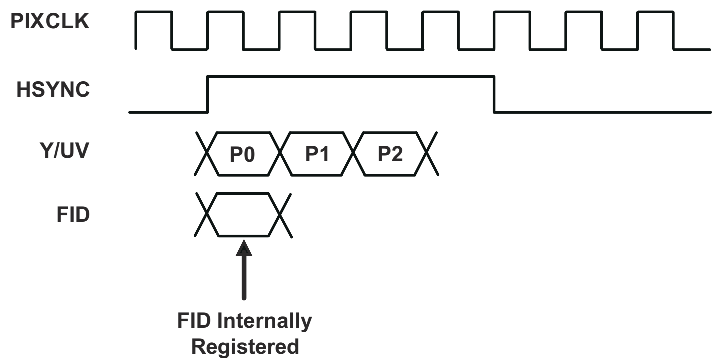


Figure 9-36. FID Registering When Using HSYNC

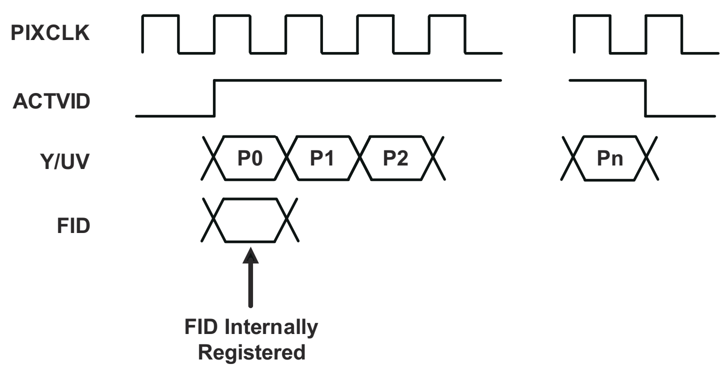


Figure 9-37. FID Registering When Using ACTVID

9.4.5.6.10 Field ID Determination Using VSYNC Skew

In order to save a device pin, there is a case where a skew may be inserted into VSYNC (with respect to HSYNC) when HSYNC is used as a start of line indicator as described in Figure 9-33. In this case, no FIELD ID signal is sent by the source chip. A description of Field ID determination by VSYNC skew is shown in Figure 9-38.

The active polarity of VSYNC falling within n pixel clock cycles of the first active cycle of HSYNC indicates the field id. If VSYNC is active before this time window, then the FIELD_ID = '1' for the next picture. If VSYNC becomes active within this window, then FIELD_ID = '0' for the next picture.

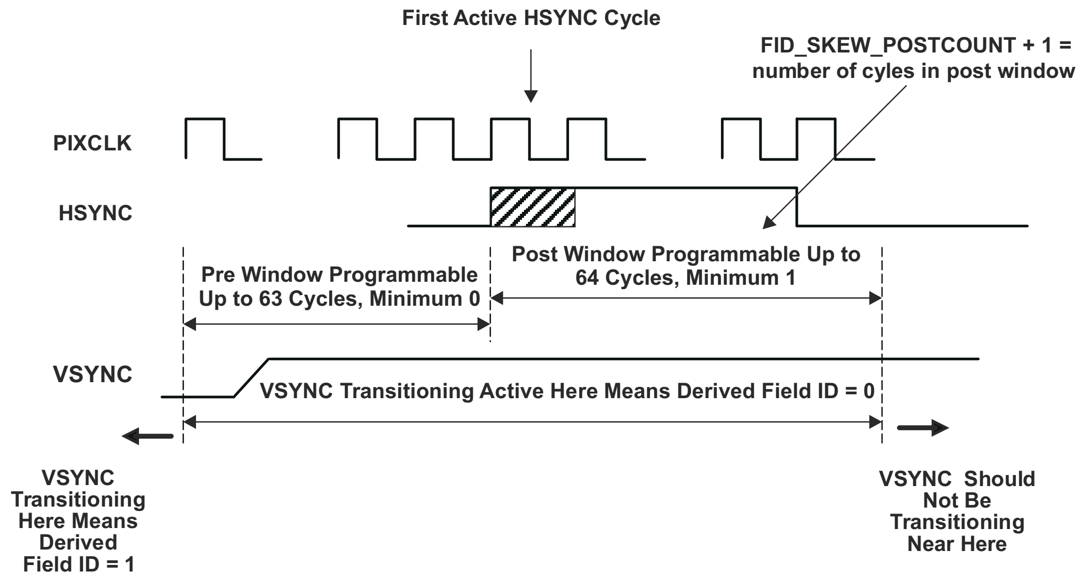


Figure 9-38. Field ID Determination By VSYNC Skew

When using FID determination by VSYNC skew, the value for VSYNC is also determined by transitions in the window as shown in Figure 9-35.

The VIP_PARSER supports a configuration FID_POLARITY bit within VIP_PORT_A and VIP_PORT_B registers. For FID determination by VSYNC Skew, the fid determination functions are described in Table 9-10.

Table 9-10. Polarity Table for FID Determination By VSYNC Skew

FID_POLARITY	Transition in Pre/Post Range	FID Determination
0	No	1
0	Yes	0
1	No	0
1	Yes	1

9.4.5.6.11 Rationale for FID Determination By VSYNC Skew

FID determination by VSYNC skew is a method for field ID determination derived from the analog NTSC and PAL interlaced specifications. Under this method, the sending device will not be providing a FID signal. NTSC has 525 total lines split between two fields. PAL has 625 total lines split between two fields. Each of these interlaced standards support an odd number of lines.

Let’s consider just the 525 active line NTSC signal. For the sake of consistency, let’s call Line 1 the first line of the 2-field pair and Line 525 the last line of the 2-field pair.

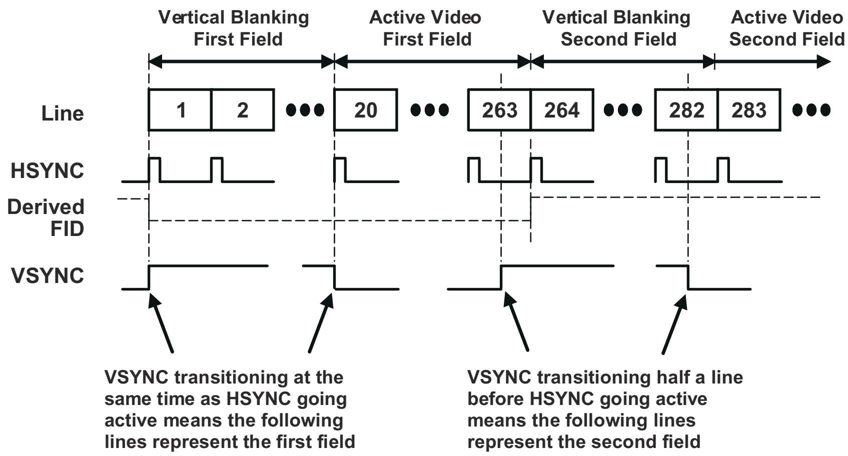


Figure 9-39. Example of 525-line FID Determination By VSYNC Skew

A waveform is shown in Figure 9-39. VSYNC is defined to go active at the same time as HSYNC for the first line of the first field in a two-field picture pair. For this first field, VSYNC will go inactive after Line 20.

For the second field, VSYNC will go active in the middle of Line 263 to signal that Line 264 is the start of a vertical blanking interval. When HSYNC for Line 264 arrives, coinciding with the vertical blanking interval for the beginning of the second field, VSYNC has already been active for half of Line 263. For the second field, VSYNC will go inactive midway through Line 282 to indicate that Line 283 is active video. When HSYNC for Line 283 appears, VSYNC has already been inactive for half a line.

By seeing whether VSYNC transitions at the beginning of a line or whether it transitions at the midway point of a line, one can determine whether the upcoming group of lines represents the first field or the second field. The derived FID is shown in dashed lines.

The analog NTSC specification defines the field ID changing part way into the vertical blanking. That is, the first few lines of vertical blanking belong to the previous field and the next several lines of vertical blanking belong to the upcoming field. The VIP_PARSER saves one channel of the entire vertical blanking interval between two active video fields into a single buffer. The hardware does not discriminate between whether the vertical blanking lines belong to the bottom of the previous field or those belonging to the start of the next field. This usage model is consistent with vertical ancillary data capture for embedded sync mode of operation.

Obviously, FID Determination by VSYNC skew cannot be used when framing does not use the VSYNC signal but rather relies on the ACTVID signal instead.

9.4.5.6.12 ACTVID Framing

Instead of an HSYNC signal, the VIP_PARSER can use ACTVID framing as described in Figure 9-32. Under ACTVID Framing, VSYNC is used to separate vertical blanking lines from active video lines.

FID determination by VSYNC Skew is not allowed for ACTVID framing because there is no HSYNC input signal in this mode. Also, the VSYNC transition window is not employed. VSYNC is captured at the first pixel of each ACTVID grouping of pixels. Lines are separated by ACTVID transitioning inactive.

9.4.5.6.13 Ancillary Data Storage in Descrete Sync Mode

Ancillary data appearing in horizontal blanking is called Horizontal Blanking Ancillary Data. Ancillary Data in vertical blanking is called Vertical Blanking Ancillary Data.

Horizontal Blanking Ancillary Data is not commonly used. For the ACTVID data valid mode described in Figure 9-31, there is no way to capture Horizontal Blanking Ancillary Data. Using the HSYNC mode in Figure 9-33, all blanking pixels are captured. However, the horizontal ancillary data is byte-by-byte distributed between the Luma and Chroma frame buffers. Chroma sited bytes are saved in the Chroma frame buffer and Luma sited bytes are saved in the Luma frtame buffer. Some CPU effort would be needed in order to extract ancillary data from the desired Luma or Chroma channel frame buffers. Also, the video on each line starts after the horizontal blanking period. This situation is shown in Figure 9-40.

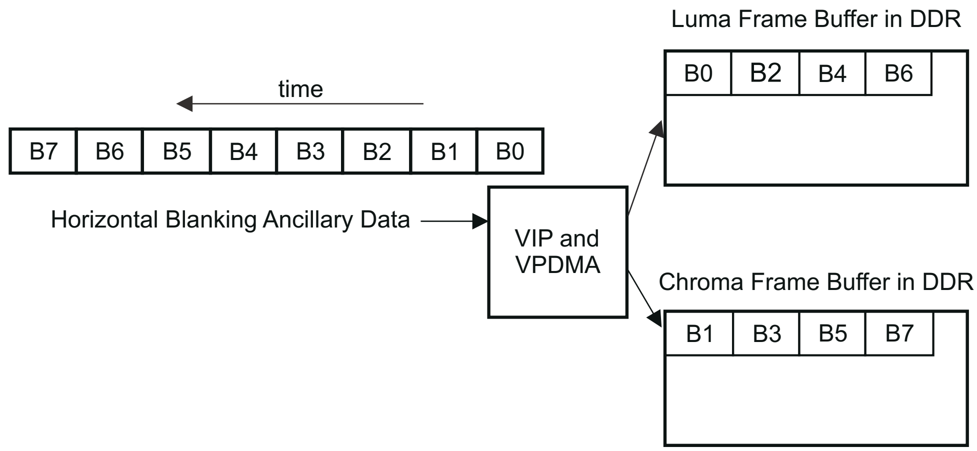


Figure 9-40. Horizontal Ancillary Data Packing When HSYNC Used as Sync Signal

With Interlaced source material, Vertical Blanking Ancillary data will be stored in a separate Vertical Ancillary Data Buffer as shown in Figure 9-41. The Channel from which vertical ancillary data is extracted is a configuration option. For an input image of x active pixels per line, each line of Vertical Blanking Ancillary Data will have x bytes. Unlike the horizontal case, the CPU parsing this Ancillary Data will see a contiguous section of Vertical Ancillary Data that is not intermixed with Video data.

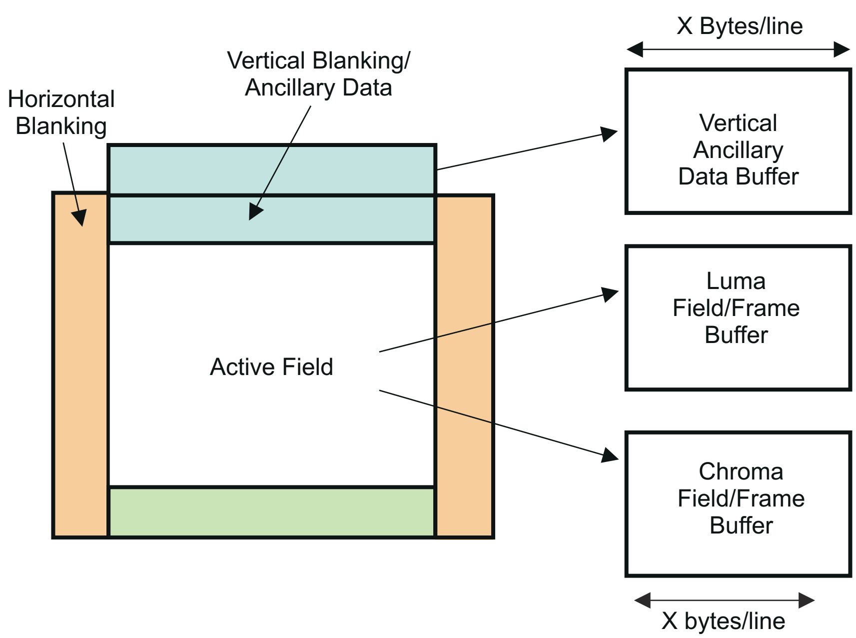


Figure 9-41. Interlaced Field Vertical Blanking Ancillary Data Storage

For Progressive source video, the FIELD ID does not change. So, the Vertical Ancillary Data Buffer will contain all the information beginning from the vertical blanking of the previous frame. This situation is shown in Figure 9-42.

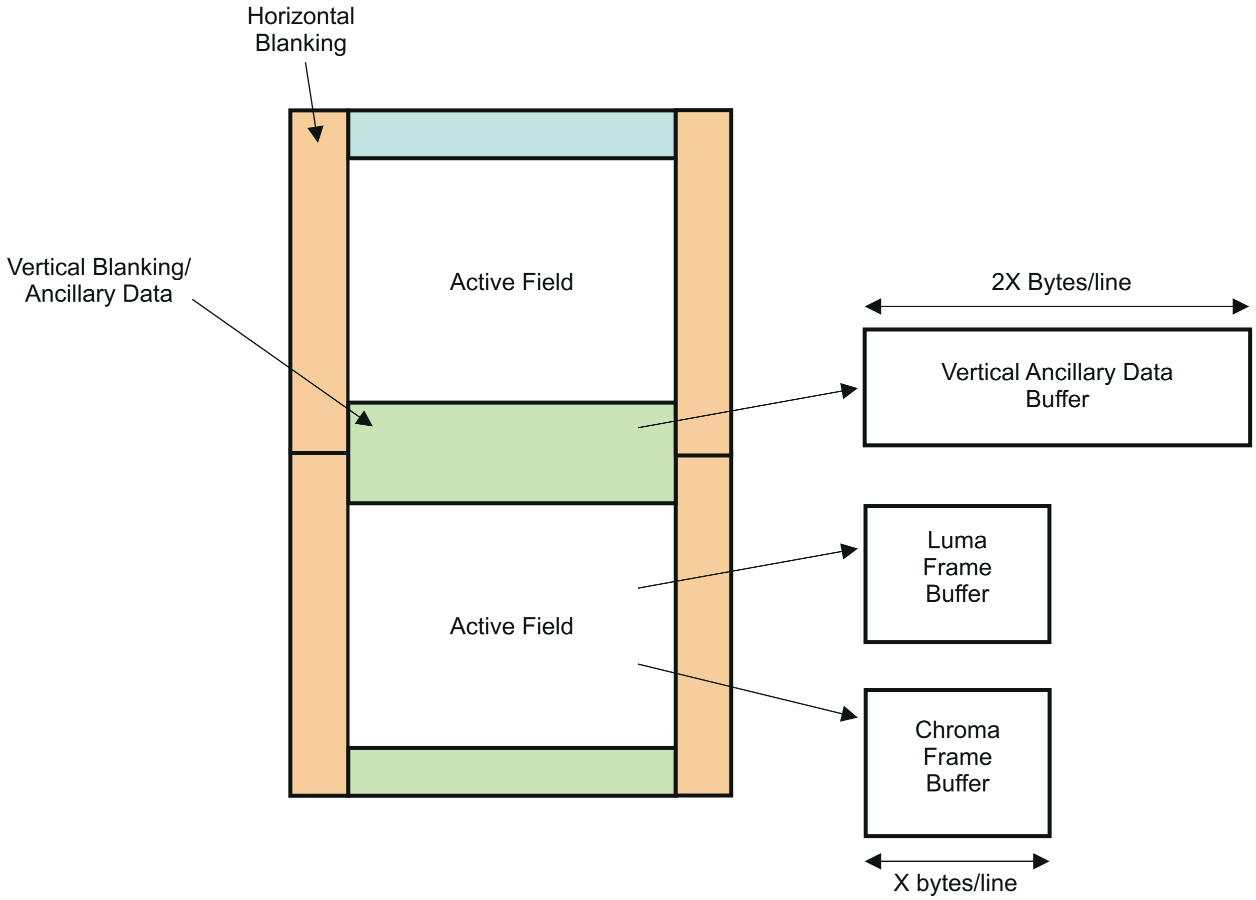


Figure 9-42. Progressive Frame Vertical Blanking Ancillary Data Storage

9.4.5.7 BT.656 Style Embedded Sync

9.4.5.7.1 Data Input

Like Discrete Sync Input, Embedded Sync mode takes data from the 24b input bus. Input data can be 8, 16, or 24 bits wide. A sample is retrieved each and every Pixel Clock cycle. There is no valid signal gating data entry. Figure 9-43 shows a valid data sample each Pixel Clock period.

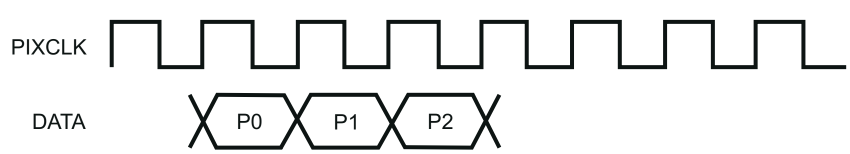


Figure 9-43. Embedded Sync Data Entry

9.4.5.7.2 Sync Words

In embedded sync mode, code words are inserted into the stream at pixel clock rates. For external devices that send out 10 bits (single pixel interface) or 20 bits (parallel Y-Cb/Cr interface) of data, only the 8 (single pixel interface) or 16 (parallel 8bY-8bCb/Cr interface) most significant bits of each pixel are used.

The key code words are Start of Active Video (SAV) and End of Active Video (EAV). Three flags are found in these code words: F (field), V (vertical sync), and H (horizontal sync). These flags signify the position in the frame corresponding to the data immediately following the codeword. The flags determine whether the code is EAV or SAV and where they lie in the picture. The first byte of the code word is 0xFF. The second and third bytes are 0x00. The bit ordering of the fourth byte is detailed in Table 9-11.

Table 9-11. Fourth Byte of EAV/SAV Code Word

7 (fixed)	6 (F)	5 (V)	4 (H)	3 (P3=V^H)	2 (P2=F^H)	1 (P1=F^V)	0 (P0=F^V^H)	Description
1	0	0	0	0	0	0	0	SAV, Field 0, Active Video
1	0	0	1	1	1	0	1	EAV. Field 0, Horizontal Blanking
1	0	1	0	1	0	1	1	SAV, Field 0, Vertical Blanking
1	0	1	1	0	1	1	0	EAV, Field 0, Horizontal Blanking in Vertical Blanking Region
1	1	0	0	0	1	1	1	SAV, Field 1, Active Video
1	1	0	1	1	0	1	0	EAV, Field 1, Horizontal Blanking
1	1	1	0	1	1	0	0	SAV, Field 1, Vertical Blanking
1	1	1	1	0	0	0	1	EAV, Field 1, Horizontal Blanking in Vertical Blanking Region

An example of the input ordering of the embedded code word, followed by active video, is shown in Figure 9-44. The input data mode is 8 bits for the example.

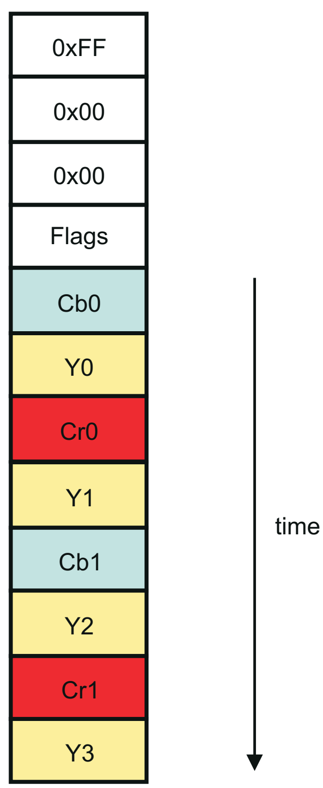


Figure 9-44. Code Word Format Example Followed by Video Data

9.4.5.7.3 Error Correction

The FVH flags are sent with four protection bits to support double error detection, single error correction. A non-correctable detected error is simply ignored. An option exists for the protection bits to correct a single bit error in the FVH flags. The correction table is shown in Table 9-12. n/c means that the error condition is detected, but it is non-correctable.

Table 9-12. Error Correction Matrix

	F, V, and H Flags							
P3, P2, P1, P0	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	000	000	000	n/c	000	n/c	n/c	111
0001	000	n/c	n/c	111	n/c	111	111	111
0010	000	n/c	n/c	011	n/c	101	n/c	n/c
0011	n/c	n/c	010	n/c	100	n/c	n/c	111
0100	000	n/c	n/c	011	n/c	n/c	110	n/c
0101	n/c	001	n/c	n/c	100	n/c	n/c	111
0110	n/c	011	011	011	100	n/c	n/c	011
0111	100	n/c	n/c	011	100	100	100	n/c
1000	000	n/c	n/c	n/c	n/c	101	110	n/c
1001	n/c	001	010	n/c	n/c	n/c	n/c	111
1010	n/c	101	010	n/c	101	101	n/c	101
1011	010	n/c	010	010	n/c	101	010	n/c
1100	n/c	001	110	n/c	110	n/c	110	110
1101	001	001	n/c	001	n/c	001	110	n/c
1110	n/c	n/c	n/c	011	n/c	101	110	n/c
1111	n/c	001	010	n/c	100	n/c	n/c	n/c

9.4.5.7.4 Embedded Sync Ancillary Data

With Embedded Sync streams, only Vertical Ancillary Data can be extracted. The Vertical Ancillary Data buffer is the same width as the corresponding Luma and Chroma buffers. The channel from which Vertical Ancillary Data is extracted is a configuration option.

Horizontal Ancillary data cannot be extracted using embedded sync mode.

The Vertical Ancillary Data is captured starting from the end of the previous active video. See Figure 9-45 for a more detailed description of embedded sync packing.

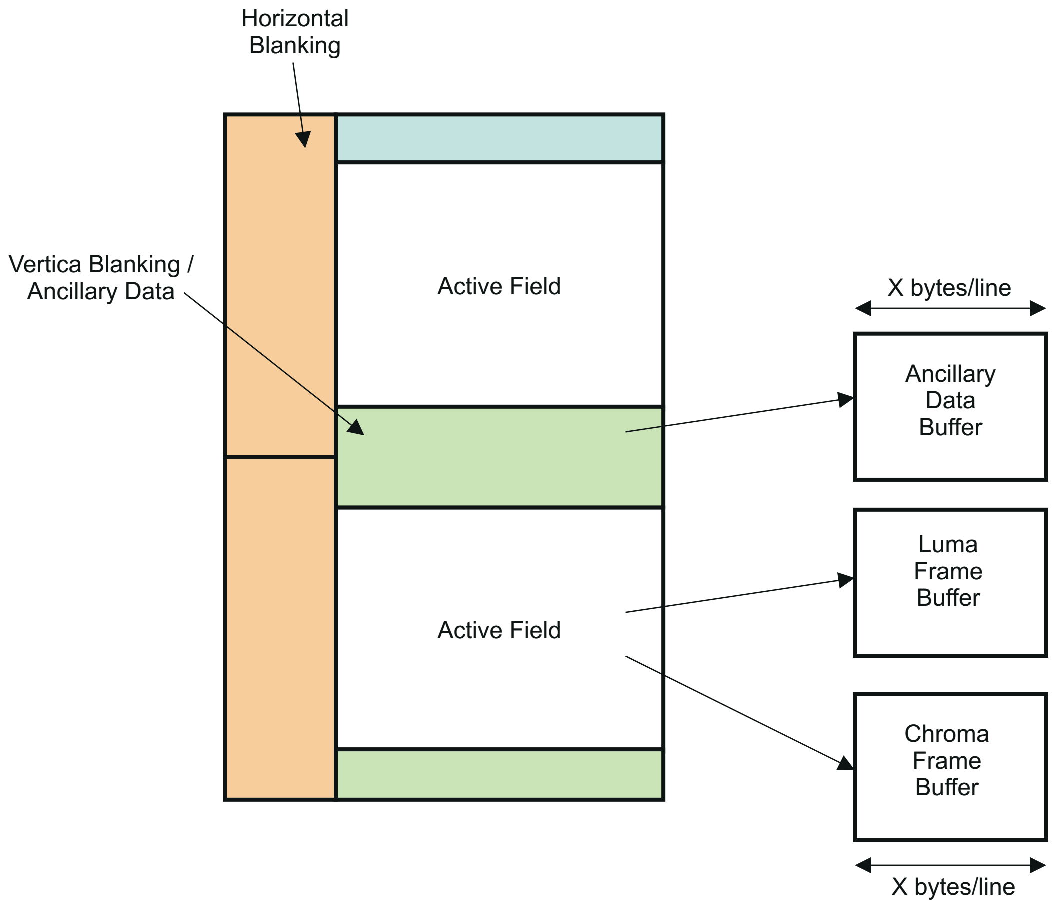


Figure 9-45. Embedded Sync Packing

9.4.5.7.5 Embedded Sync RGB 24-bit Data

YUV streams are separated into a planar Luma buffer, a planar Chroma Pair (CbCr) buffer, and a planar Vertical Ancillary Data buffer. RGB streams, on the other hand, are stored in a packed R-G-B format, as shown in Figure 9-46.

The BT.1120 standard defines a method of carrying RGB streams. After the SAV code, 8 bits of R, 8 bits of G, and 8 bits of B data are clocked in one cycle. A 24 bit data bus is required. The channel in which to search for the embedded sync codes and the FVH control code is determined by a configuration selection. Only vertical ancillary data from one channel (R, G, or B), which happens to be the channel where control codes are found, are captured. Vertical ancillary data in the other two channels are ignored.

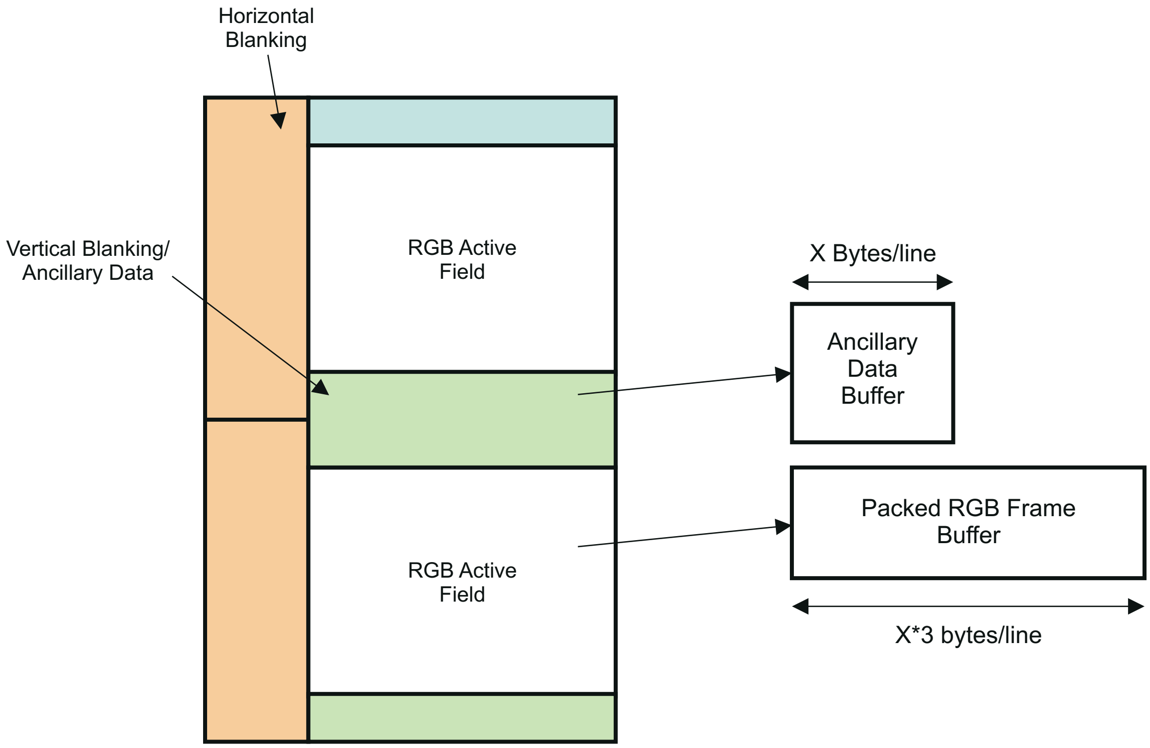


Figure 9-46. RGB Frame Storage

9.4.5.8 Source Multiplexing

9.4.5.8.1 Multiplexing Scenarios

Some applications require multiple camera sources to be used at the same time. For this type of device, one solution would be to support N-number of 8-bit or 16-bit data interfaces for each of N cameras. However, this solution does not efficiently minimize pin count. One set of 8-bit or 16-bit interfaces has the bandwidth to support more than one video source, depending on the resolution of the video. Table 9-13 is explanatory only and shows the number of sources that can be multiplexed in one VIP for 8-bit and 16-bit interface modes. Note that it does not reflect the capabilities of the VIP_PARSER. In addition, the interface pixel clock rates are shown. The VPDMA limits 16 camera sources to be saved to DDR memory per Pixel Clock Input Domain.

Table 9-13. Multiplexing Configurations and Pixel Clock Rates

	Maximum Channels in Single 16-bit Data Interface Mode	Maximum Channels in Dual 8-bit Data Interface Mode - Interleaved Channels per Single 8-bit Port. One 16-bit VIP can be configured to support two such 8-bit ports.	Interface Clock Rate (MHz)
HD Interlaced	2	1	148.5
D1 Interlaced	8	4	108.1
CIF Interlaced	n/a	n/a	n/a
HD Progressive	1	n/a	148.5
D1 Progressive	4	2	108.1
CIF Progressive (1)	32	16	162.2

(1) Blanking pixels are not used in the CIF clock rate calculations. Addition of blanking pixels would require a slightly higher clock rate.

Note

These Channel Density values reflect one VIP subsystem.

9.4.5.8.2 2-Way Multiplexing

For 2-Way Multiplexing, two embedded sync streams are interleaved a pixel at a time as shown in Figure 9-47.

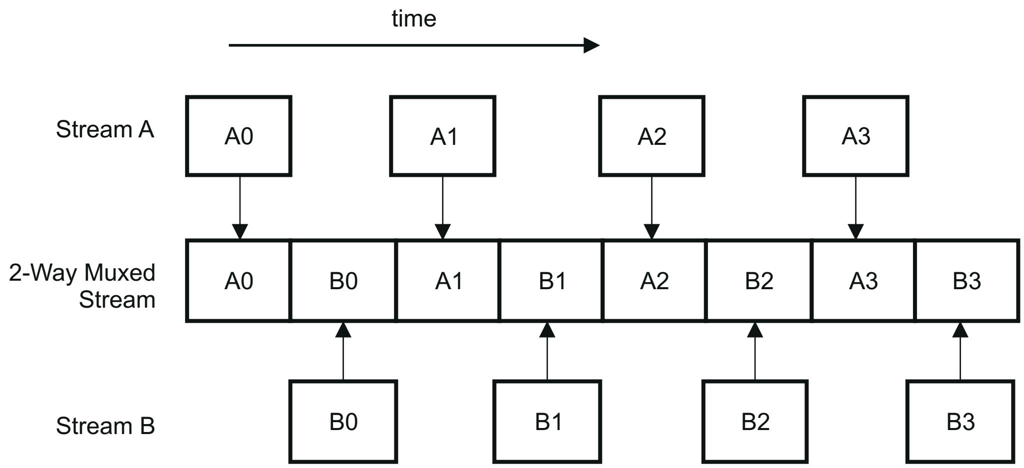


Figure 9-47. 2-Way Multiplexing

The sync codeword, FF-00-00-XY, is replicated in both source streams. In 2-Way Multiplexing, the sizes of both camera sources must be the same. Likewise, the Vertical Ancillary Data size for both sources must be identical. However, the two streams are not necessarily sending the same pixel site in adjacent clock cycles.

9.4.5.8.3 4-Way Multiplexing

For 4-Way Multiplexing, four embedded sync streams are multiplexed into one as seen in Figure 9-48.

Again, the sync codeword is in all four sources. Like 2-Way Multiplexing, the sizes of the four camera sources are the same and the sizes of the Vertical Ancillary Data regions are the same. The four streams are not necessarily sending the same pixel site in adjacent clock cycles.

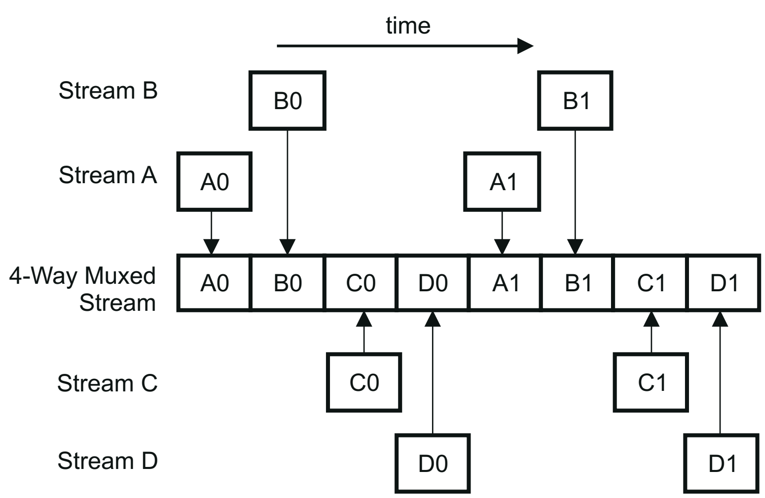


Figure 9-48. Example of 4-Way Multiplexing

9.4.5.8.4 Line Multiplexing

In Line Multiplexing, n-different sources are sent into the VIP a complete line at a time using a modified version of embedded sync. An example of Line Multiplexing for two sources is shown in Figure 9-49.

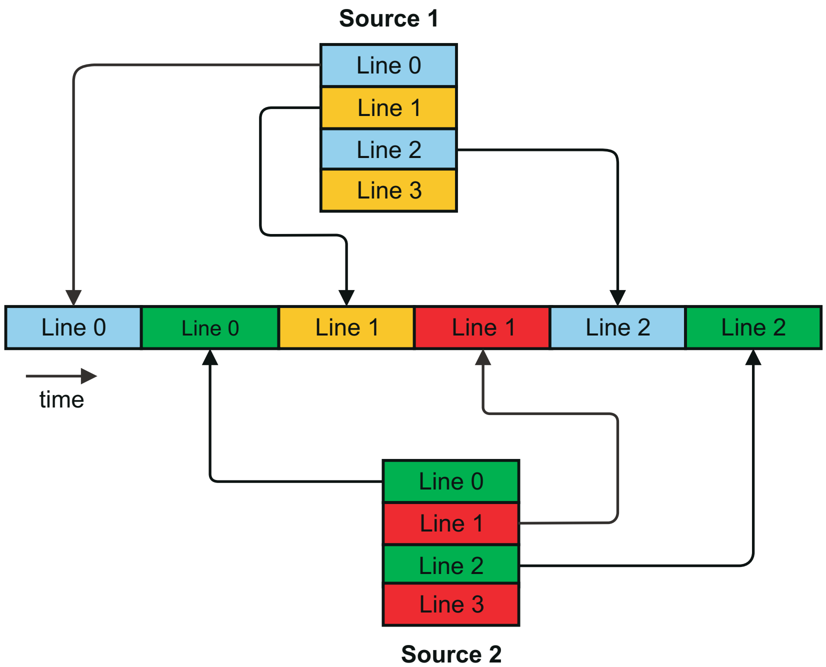


Figure 9-49. Example of Line Multiplexing

The width and height of each source in Line Multiplexed data can be different. For instance, one source can be PAL while another one can be NTSC. A line is comprised of YUV422 pixels in repeating patterns of CbYCrY.

9.4.5.8.5 Super Frame Concept in Line Multiplexing

Different camera sources are interleaved on a line-by-line basis. The beginning of each line carries a Metadata tag that provides key information about that line. This Metadata tag is preceded by a SAV codeword in which F=0, V=0, and H=0.

At the end of the line, an EAV code is inserted with an appropriate number of padding pixels following it. This EAV code has F=0, V=0, and H=1. The startcodes used for the Metadata wrapped lines and the dummy lines form the Super Frame. The VIP_PARSER module has logic to parse out the super frame, analyze the Metadata tags, and frame buffer the line contents appropriately.

9.4.5.8.6 8-bit Data Interface in Line Multiplexing

Figure 9-50 is an example of an 8-bit line multiplexing interface. Channel Data is the CbYCrY sequence representing a line. Preceding Channel Data is the four byte Meta Data tags. Note that the Meta Data bytes are replicated in both the Luma and the Chroma sites. The entire structure is bounded by a traditional SAV/EAV code in which the V flag is 0.

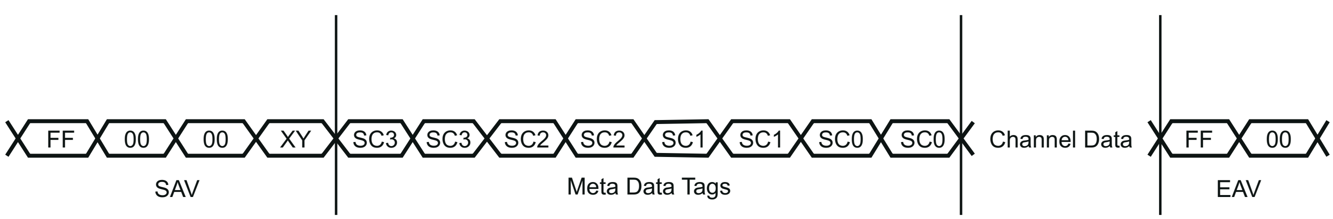


Figure 9-50. 8-bit Line Mux Interface

9.4.5.8.7 16-bit Data Interface in Line Multiplexing

Figure 9-51 describes the 16-bit line multiplexing interface. Channel Data is the active line. All the Y pixels are in the Luma Channel. The CbCr pixels are in the Chroma Channel. Each cycle, a 16-bit value representing one Luma sample and one Chroma sample enters the VIP_PARSER. The Meta Data tags are replicated in both channels. Likewise, the SAV/EAV startcodes are found in both channels. The V flags for the SAV/EAV startcodes are always 0.

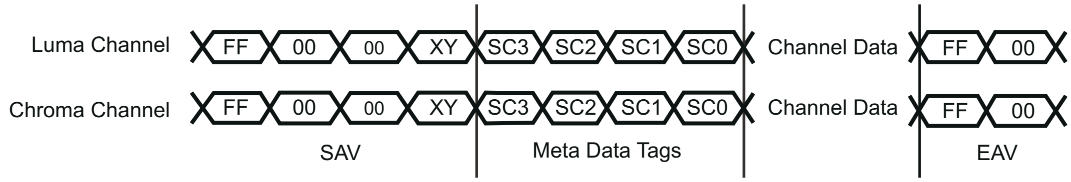


Figure 9-51. 16-bit Line Mux Interface

9.4.5.8.8 Split Lines in Line Multiplex Mode

Suppose an external device is sending two dissimilar sources in Line Multiplex mode. One narrower source has X pixels per line. The wider source has 2X pixels per line.

The Meta Data has provisions for the external device to split a line. The Beginning of Line (BOL) and End of Line (EOL) flags tag a split line as described in Table 9-14.

Table 9-14. Split Line Table

BOL	EOL	Function
0	0	Undefined
0	1	Line Segment is the second half of a line
1	0	Line Segment is the first half of a line
1	1	Line has not been segmented into two.

9.4.5.8.9 Meta Data

Table 9-15 shows the bitfields in the Meta Data start codes.

Table 9-15. Meta Data Layout

Byte	7	6	5	4	3	2	1	0
SC3	1	BOP	EOP	RSVD	CH_ID[3:0]			
SC3	1	BOP	EOP	RSVD	CH_ID[3:0]			
SC2	0	BOL	EOL	VDET	LINE_ID[10:7]			
SC1	~LINE_ID[6]	LINE_ID[6:0]						
SC0	PAD	F	V	H	P3	P2	P1	P0

BOP tags the line as a startline in a period. A period is defined as the contiguous lines in a vertical blanking period or the contiguous lines in a field or frame. For the vertical blanking period case, the vertical blanking at the bottom of the previous field or frame combined with the vertical blanking at the top of the current field or frame is combined to create one period.

EOP tags the line as an endline in a period. Figure 9-52 shows the definition of the two types of Periods as outlined by BOP and EOP.

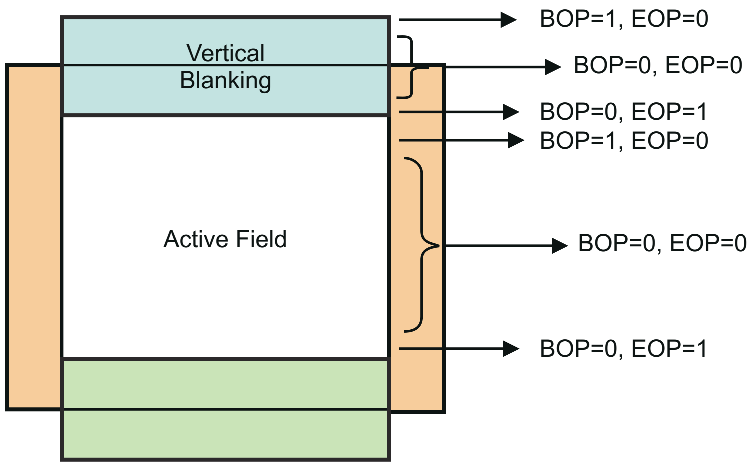


Figure 9-52. BOP/EOP Definition of a Period

For the Split Line at the top of a Period, the BOP bit is set for both halves of the split line. Likewise, for the Split Line at the bottom of a Period, the EOP bit is set for both halves of the split line.

CH_ID is the channel ID, which tags the camera source that generated the incoming line. A maximum of 16 camera sources are support per Pixel Input Clock Domain.

LINE_ID is the line number, starting from 0 and incrementing by one for each subsequent line from the same source.

PAD is a flag which tags the line as an artificially inserted padding line. When PAD is ‘0’, the line should be discarded.

F, V, H, P3, P2, P1, and P0 are the bits representing the normal XY code. F is the Field ID associated with line, V signals when the line is in the vertical blanking, H specifies that the line is in the Horizontal Blanking, and P3:P0 are the protection bits.

Since only active video and vertical ancillary data lines are encapsulated in the Meta Data, the H bit in the SC0 byte should never be ‘1’.

9.4.5.8.10 TI Line Mux Mode, Split Lines, and Channel ID Remapping

The VIP_PARSER supports a maximum of 8 different Channel IDs per Port. The Channel IDs must be in the range {0:7} (3 bits). In the source multiplex, only one source can be a split line source and have the same Channel ID as one of the non-split line sources.

This scenario involves an external NTSC decoder which supports 8 D1 cameras. The external NTSC decoder will downscale the 8 sources to SIF format. However, one camera source will be sent in the multiplex as both the downscaled version and the original D1 sized version. They will both have the same Channel ID. However, the D1 version will be sent as a split-line. The source multiplex will thus have 9 maximum streams.

The VIP_PARSER (for TI Line Mux Mode only), will left shift the Channel ID by one. Bit 0 is used as an indicator whether the Source is a split-line source or a normal non-split line source. Only one of the nine inputs can be a split line source.

Table 9-16. TI Line Mux Mode Channel ID Remapping

	Channel ID Sent to VPDMA	
Source Input Channel ID	**Non-split Line**	**Split Line**
0x0	0x0	0x1
0x1	0x2	0x3
0x2	0x4	0x5
0x3	0x6	0x7
0x4	0x8	0x9

Table 9-16. TI Line Mux Mode Channel ID Remapping (continued)

0x5	0xA	0xB
0x6	0xC	0xD
0x7	0xE	0xF

All subsequent references to the Camera Source, such as in a VPDMA return descriptor, will reference the remapped Channel ID.

The VIP_OUTPUT_PORT_A_SRC0_SIZE through VIP_OUTPUT_PORT_A_SRC15_SIZE registers for Port A, and VIP_OUTPUT_PORT_B_SRC0_SIZE through VIP_OUTPUT_PORT_B_SRC15_SIZE registers for Port B and the VIP_OUTPUT_PORT_A_SRC_FID and VIP_OUTPUT_PORT_B_SRC_FID status registers reflect the remapped Channel ID when the port is in TI Line Mux mode.

9.4.5.9 Channel ID Extraction for 2x/4x Multiplexed Source

9.4.5.9.1 Channel ID Extraction Overview

For 2-way and 4-way multiplexed source, the Channel ID is either embedded in the four protection bits inside the EAV/SAV code words or in the horizontal blanking pixel data. A configuration setting determines where the VIP_PARSER would search for the Channel ID.

9.4.5.9.2 Channel ID Embedded in Protection Bits for 2- and 4-Way Multiplexing

The four-bit channel ID is an identifier corresponding with the source number (camera) of the incoming video. As shown in Table 9-17, the Channel ID is placed in the code fourth byte of the EAV/SAV code words normally used for protection bits. With 4 bits, the maximum number of sources that can be defined in this range is 16. However, only Channel IDs in the range {0:7} are supported. Obviously, error correction cannot be performed on the FVH flags since the protection bits are no longer there.

Table 9-17. Channel ID Embedded in EAV/SAV

7 (fixed)	6 (F)	5 (V)	4 (H)	3 (ch_id[3])	2 (ch_id[2])	1 (ch_id[1])	0 (ch_id[0])	Description
1	0	0	0	Ch_id = {0:15}				SAV, Field 0, Active Video
1	0	0	1	Ch_id = {0:15}				EAV, Field 0, Horizontal Blanking
1	0	1	0	Ch_id = {0:15}				SAV, Field 0, Vertical Blanking
1	0	1	1	Ch_id = {0:15}				EAV, Field 0, Horizontal Blanking in Vertical Blanking Region
1	1	0	0	Ch_id = {0:15}				SAV, Field 1, Active Video
1	1	0	1	Ch_id = {0:15}				EAV, Field 1, Horizontal Blanking
1	1	1	0	Ch_id = {0:15}				SAV, Field 1, Vertical Blanking
1	1	1	1	Ch_id = {0:15}				EAV, Field 1, Horizontal Blanking in Vertical Blanking Region

9.4.5.9.3 Channel ID Embedded in Horizontal Blanking Pixel Data for 2- and 4-Way Multiplexing

In Horizontal Blanking and Vertical Blanking, non-ancillary data pixels should be Y=0x10 and Cb=Cr=0x80. When the Channel ID is embedded in the Horizontal Blanking for 2 and 4-way multiplexing, the lower nibbles of all Luma and Chroma pixels are replaced by the 4 bit Channel ID. This scenario is shown in Figure 9-53. The maximum number of values defined by this 4-bit range is 2^4 = 16. However, only Channel IDs in the range {0:7} are supported.

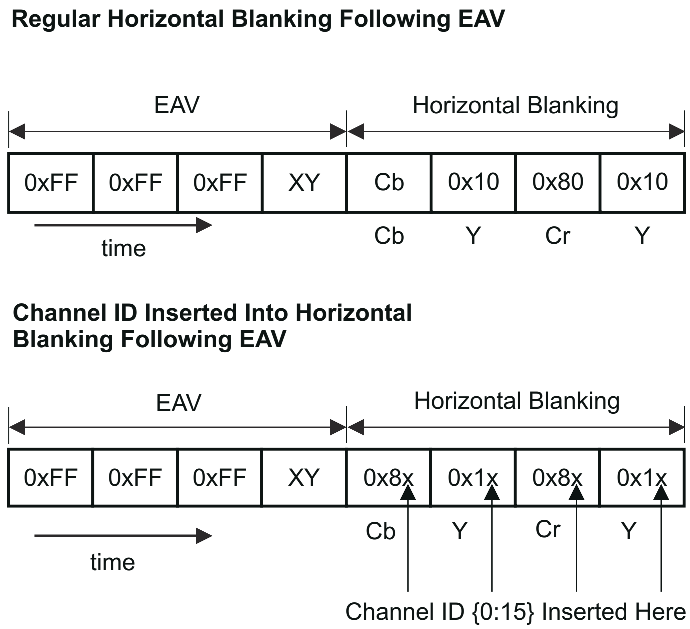


Figure 9-53. Channel ID Inserted Into Horizontal Blanking

9.4.5.10 Embedded Sync Mux Modes and Data Bus Widths

Legal combinations of Embedded Sync Mux Modes and Data Bus Widths are described in Table 9-18.

Table 9-18. Valid Embedded Sync Mux Mode and Data Bus Width Combinations

	1x Mux	2x Mux	4x Mux	Line Mux
8 Bit	v	v	v	v
16 Bit	v	n/a	n/a	v
24 Bit	v	n/a	n/a	n/a

9.4.5.11 Ancillary and Active Video Cropping

One Source Number for each Port can be cropped. Cropping is available for both Ancillary Data and Active Video.

For the Vertical Ancillary Data from Port A, cropping is enabled by setting the VIP_ANC_CROP_HORZ_PORT_A[15] ANC_BYPASS_N bit. The Source Number from Port A that gets cropped is defined by the VIP_ANC_CROP_HORZ_PORT_A[31:28] ANC_TARGET_SRCNUM register. VIP_ANC_CROP_HORZ_PORT_A[11:0] ANC_SKIP_NUMPIX, VIP_ANC_CROP_HORZ_PORT_A[27:16] ANC_USE_NUMPIX, VIP_ANC_CROP_VERT_PORT_A[11:0] ANC_SKIP_NUMLINES, and VIP_ANC_CROP_VERT_PORT_A[27:16] ANC_USE_NUMLINES define the region of the selected Source Number that is cropped and sent to DRAM. The Vertical Ancillary Data Cropping region is described in Figure 9-54.

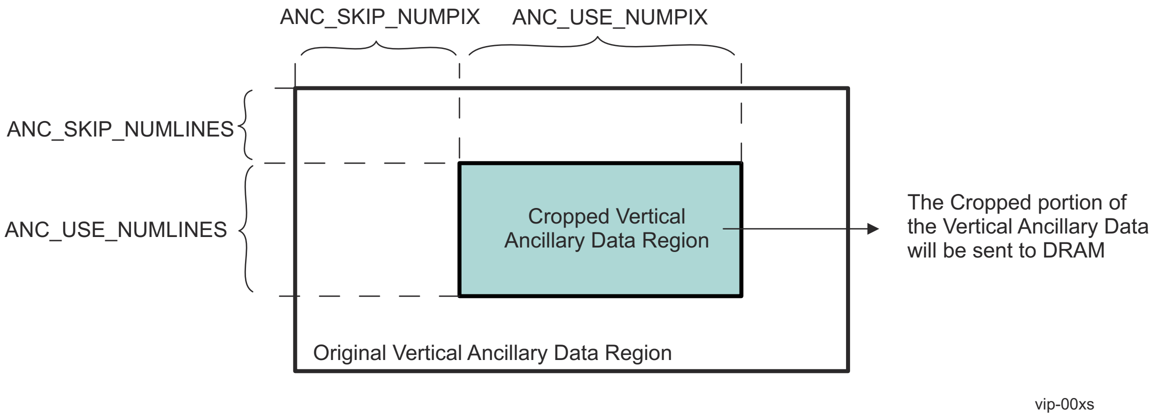


Figure 9-54. Vertical Ancillary Data Cropping

Note that for 8-bit input mode only, a setting exists that allow Vertical Ancillary data from both the Luma and Chroma channels to be captured. Both channels of Vertical Ancillary Data are captured when VIP_XTRA_PORT_A[14:13] ANC_CHAN_SEL_8B is set to “1x”. Thus, the number of data elements per line in this case is twice the equivalent number of Luma pixels per line. In other words, for this particular dual channel capture example, if there are 720 Luma pixels per line, then the total number of Vertical Ancillary Data Pixels in the source picture can be 2x720=1440 pixels.

For Active Video from Por tA, cropping is enabled by setting the VIP_CROP_HORZ_PORT_A[15] ACT_BYPASS_N bit. The Source Number from Port A that gets cropped is defined by the VIP_CROP_HORZ_PORT_A[31:28] ACT_TARGET_SRCNUM register.

VIP_CROP_HORZ_PORT_A[11:0] ACT_SKIP_NUMPIX, VIP_CROP_HORZ_PORT_A[27:16] ACT_USE_NUMPIX, VIP_CROP_VERT_PORT_A[11:0] ACT_SKIP_NUMLINES, and VIP_CROP_VERT_PORT_A[27:16] ACT_USE_NUMLINES define the region of the selected Source Number that is cropped and sent to DRAM. The Vertical Ancillary Data Cropping region is described in Figure 9-55

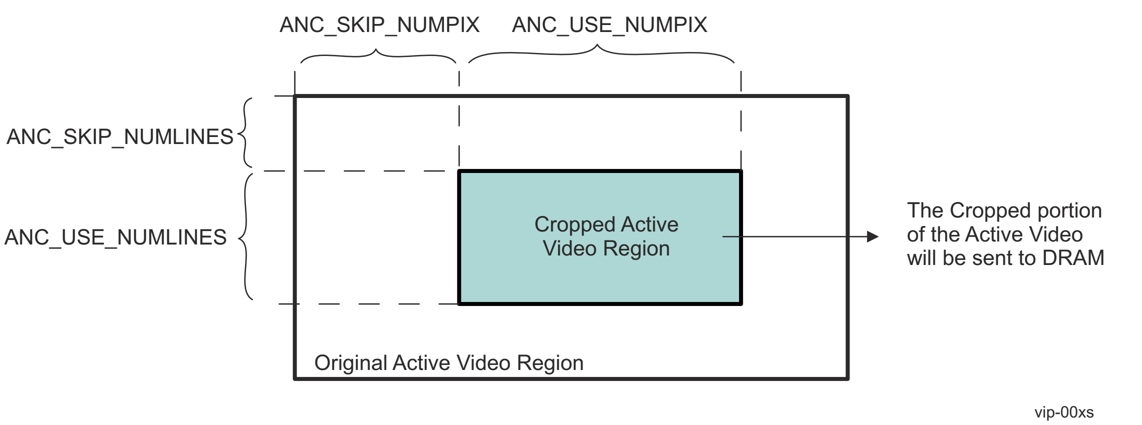


Figure 9-55. Active Video Cropping

Cropping for Port B works in a similar way to Port A. Since picture data is in 4:2:2 format, ANC_SKIP_NUMPIX(ACT_SKIP_NUMPIX) and ANC_USE_NUMPIX (ACT_USE_NUMPIX) must be evenly divisible by 2. If the output of VIP_PARSER is sent to a 4:2:2 to 4:2:0 converter, then ANC_USE_NUMLINES(ACT_USE_NUMLINES) must also be evenly divisible by 2.

Error cases in cropping occur when the crop window programmed is larger than the incoming video frame. Crop window errors normally result in the return of the crop region where the crop window overlays the incoming video frame. However, there is one problematic error cropping case, as illustrated in Figure 9-56.

The programmed crop window extends below the input picture and the last pixel of the input picture is not a part of the selected crop region. In this case, at the last pixel of the last line in the green crop output, there is no way to determine that this pixel is the last pixel of the cropped output.

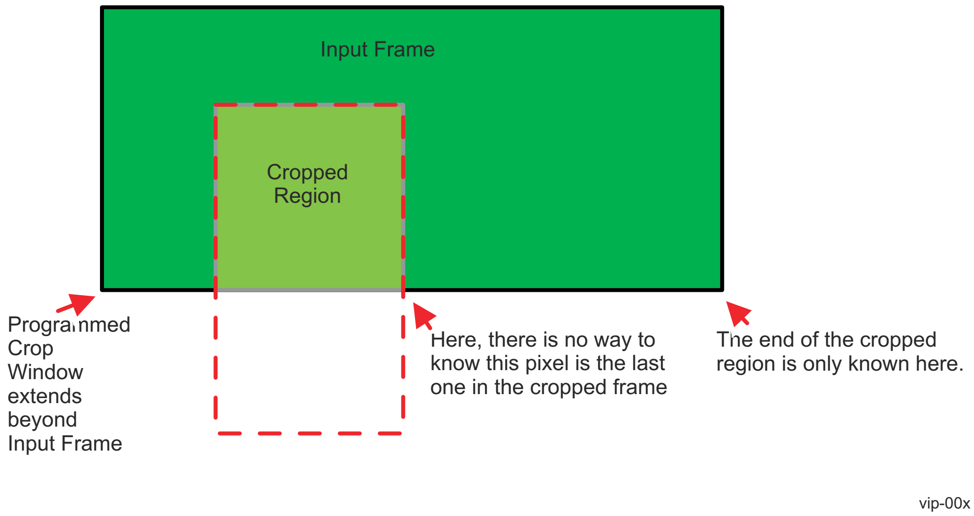


Figure 9-56. Problematic Error Cropping Case

In this case, the crop tool sends outs a single pixel with an endline and endframe when it reaches the last pixel of the input picture. If the green cropped region has n lines, the actual output from the crop tool will have n+1 lines and the width of the last line will be 1 pixel.

Figure 9-57 shows the endline (EL) and endframe (EF) signals corresponding to the last line of the cropped region. At the last line and last pixel of the cropped region, the crop tool will only output an endline. It cannot output an endframe at this point because the crop tool might get another line from the streaming input.

Later, on the same input line, the last pixel of the input frame appears. Here, the crop tool knows that the cropped region has ended. In this case, a single endline/endframe pixel is sent out to signal that the frame has ended.

Note

There is no interrupt to notify that application level that a crop error has occurred.

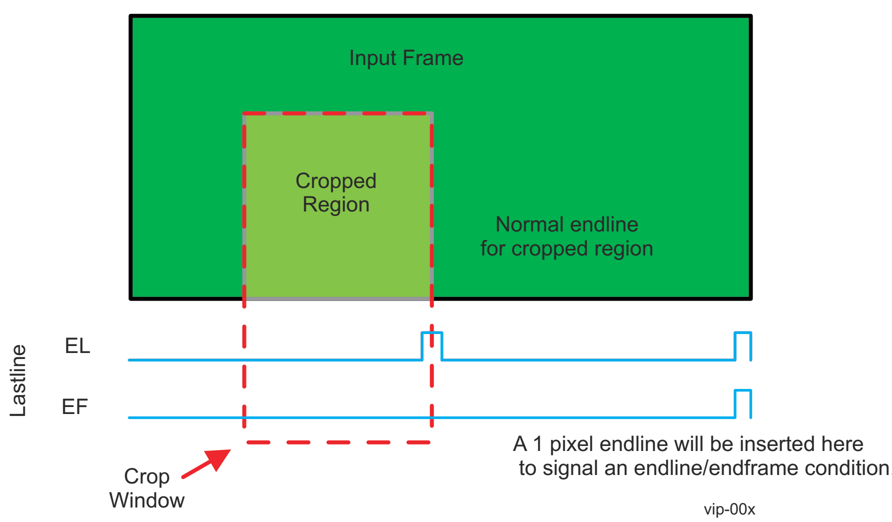


Figure 9-57. Endline/Endframe Behavior for Error Cropping Case

9.4.5.12 Interrupts

The VIP_PARSER module has 19 interrupts out of which one can be mapped to VIP top level.

When an interrupt occurs and is determined to be from the VIP_PARSER module, the VIP_PARSER level of masks, clears, and status registers must be checked and updated first.

Table 9-19 describes each of the interrupts events supported by the VIP_PARSER, together with associated Interrupt Mask (VIP_FIQ_MASK), Interrupt Clear (VIP_FIQ_CLEAR), and Interrupt Status (VIP_FIQ_STATUS) registers.

Table 9-19. VIP_PARSER Interrupt Events

Event Flag	Event Mask	Map to	Description
VIP_FIQ_STATUS[21] PORT_A_YUV_PROTOCOL_VIOLATION	VIP_FIQ_MASK[21] PORT_A_YUV_PROTOCOL_VIOLATION_MASK	PrtBDisableComplete	When a port is running and VIP_PORT_B[8] ENABLE bit is turned off, logic exists to ensure that a complete frame is sent out of the Ancillary and Active Video VPI ports. That is, a frame is not stopped in the middle. This interrupt is activated when all active frames have been sent out Port B following a disable.
VIP_FIQ_STATUS[20] PORT_A_ANC_PROTOCOL_VIOLATION	VIP_FIQ_MASK[20] PORT_A_ANC_PROTOCOL_VIOLATION_MASK	PrtADisableComplete	When a port is running and VIP_PORT_A[8] ENABLE bit is turned off, logic exists to ensure that a complete frame is sent out of the Ancillary and Active Video VPI ports. That is, a frame is not stopped in the middle. This interrupt is activated when all active frames have been sent out Port A following a disable.
VIP_FIQ_STATUS[19] PORT_B_YUV_PROTOCOL_VIOLATION	VIP_FIQ_MASK[19] PORT_B_YUV_PROTOCOL_VIOLATION_MASK	PrtBANCProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Ancillary VPI of Port B.
VIP_FIQ_STATUS[18] PORT_B_ANC_PROTOCOL_VIOLATION	VIP_FIQ_MASK[18] PORT_B_ANC_PROTOCOL_VIOLATION_MASK	PrtBYUVProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Active Video VPI of Port B.
VIP_FIQ_STATUS[17] PORT_A_CFG_DISABLE_COMPLETE	VIP_FIQ_MASK[17] PORT_A_CFG_DISABLE_COMPLETE_MASK	PrtAANCProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Ancillary VPI of Port A.
VIP_FIQ_STATUS[16] PORT_B_CFG_DISABLE_COMPLETE_CLR	VIP_FIQ_MASK[16] PORT_B_CFG_DISABLE_COMPLETE_MASK	PrtAYUVProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Active Video VPI of Port A.
VIP_FIQ_STATUS[15] PORT_B_SRC0_SIZE_STATUS	VIP_FIQ_MASK[15] PORT_B_SRC0_SIZE	PrtBSrc0Size	The output size for Srcnum=0 on Port B differs from the XTRA_PORT_B[11:0] SRC0_NUMLINES and [27:16] SRC0_NUMPIX register settings
VIP_FIQ_STATUS[14] PORT_A_SRC0_SIZE_STATUS	VIP_FIQ_MASK[14] PORT_A_SRC0_SIZE	PrtASrc0Size	The output size for Srcnum=0 on Port A differs from the XTRA_PORT_B[11:0] SRC0_NUMLINES and [27:16] SRC0_NUMPIX register settings
VIP_FIQ_STATUS[13] PORT_B_DISCONN_STATUS	VIP_FIQ_MASK[13] PORT_B_DISCONN	PrtBDisconn	Port B Link Disconnect for Srcnum 0
VIP_FIQ_STATUS[12] PORT_B_CONN_STATUS	VIP_FIQ_MASK[12] PORT_B_CONN	PrtBConn	Port B Link Connect for Srcnum 0
VIP_FIQ_STATUS[11] PORT_A_DISCONN_STATUS	VIP_FIQ_MASK[11] PORT_A_DISCONN	PrtADisConn	Port A Link Disconnect for Srcnum 0

Table 9-19. VIP_PARSER Interrupt Events (continued)

Event Flag	Event Mask	Map to	Description
VIP_FIQ_STATUS[10] PORT_A_CONN_STATUS	VIP_FIQ_MASK[10] PORT_A_CONN	PrtAConn	Port A Link Connect for Srcnum 0
VIP_FIQ_STATUS[9] OUTPUT_FIFO_PRTB_ANC_STATUS	VIP_FIQ_MASK[9] OUTPUT_FIFO_PRTB_ANC_OF	OpPrtBAnc	Overflow at Ancillary Data VPDMA interface for the Port B
VIP_FIQ_STATUS[7] OUTPUT_FIFO_PRTB_LUMA_STATUS	VIP_FIQ_MASK[7] OUTPUT_FIFO_PRTB_YUV_OF	OpPrtBYUV	Overflow at Luma VPDMA interface for Port B
VIP_FIQ_STATUS[6] OUTPUT_FIFO_PRTA_ANC_STATUS	VIP_FIQ_MASK[6] OUTPUT_FIFO_PRTA_ANC_OF	OpPrtAAnc	Overflow at Ancillary Data VPDMA interface for the Port A
VIP_FIQ_STATUS[4] OUTPUT_FIFO_PRTA_LUMA_STATUS	VIP_FIQ_MASK[4] OUTPUT_FIFO_PRTA_YUV_OF	OpPrtAYUV	Overflow at Luma VPDMA interface for Port A
VIP_FIQ_STATUS[3] ASYNC_FIFO_PRTB_STATUS	VIP_FIQ_MASK[3] ASYNC_FIFO_PRTB_OF	InPrtB	Overflow at Input Async FIFO for Port B
VIP_FIQ_STATUS[2] ASYNC_FIFO_PRTA_STATUS	VIP_FIQ_MASK[2] ASYNC_FIFO_PRTA_OF	InPrtA	Overflow at Input Async FIFO for Port A
VIP_FIQ_STATUS[1] PRTB_VDET_STATUS	VIP_FIQ_MASK[1] PRTB_VDET_MASK	PrtBVdet	Video Detect Interrupt for Port B
VIP_FIQ_STATUS[0] PRTA_VDET_STATUS	VIP_FIQ_MASK[0] PRTA_VDET_MASK	PrtAVdet	Video Detect Interrupt for Port A

A ‘1’ in the Status register associated with an Interrupt source shows that the interrupt source is pending. The Status register is read-only. To clear a bit in the Status register, the associated bit in the Clear register must be written with a ‘1.’

A ’1’ in the bit position of the Mask register associated with an Interrupt source ensures that the hardware interrupt will never be passed on to the VIP top level. A ‘0’ in the bit position of the Mask register associated with an Interrupt source will cause the interrupt controller to see a VIP_PARSER interrupt in the event the hardware in the parser triggers it.

A ‘1’ in the bit position of the Clear register associated with an Interrupt source clears the hardware interrupt status register until the next time the hardware triggers it. After a Clear, the CPU should set the bit back to a ‘0.’ Otherwise, the hardware would not be able to set any subsequent interrupts of the same type.

9.4.5.13 VDET Interrupt

For Line Multiplexing Embedded Sync mode only, the Meta Data Header includes a Video Detect (VDET) flag. The device sets this VDET flag whenever NTSC or PAL sync is found. Some other external devices using Line Multiplexing mode may not use VDET. However, when VDET changes, a VDET interrupt is issued (see Table 9-19, *Interrupts*, for more details on the interrupt). Each Pixel Clock Input Domain (Port A and Port B) has a separate VDET interrupt.

The VDET status register is comprised of 32 bits, each bit representing the value of the VDET flag found in the meta data of the Channel ID. Bit 0 is the VDET value from Channel ID 0, Bit 1is is VDET value from Channel ID 1, and so on. There is a separate status register for each Pixel Input Clock Domain (VIP_PORT_A_VDET_VEC and VIP_PORT_B_VDET_VEC registers).

In Line Mux mode, the meta-data field defining the srcnum is 5-bits wide. Only the last three bits of this field and the upper two bits are reserved. This bit should always be set to 1 in TI Line Mux mode.

9.4.5.14 Source Video Size

For each Pixel Input Clock Domain, status registers are available to log the last active video height and width found from 16 camera sources. There is no interrupt activated on the change in the source size in any of the input sources. These readonly registers only inform the application of the width and the height of the last active field or frame associated with each channel ID.

In 2x/4x pixel multiplexing, the four bit nibble carrying the Srcnum defines a maximum of 16 sources.

9.4.5.15 Clipping

In ITU-656/BT.1120 embedded sync streams, the values 0x00 and 0xFF are reserved for sync detection. These values are illegal in the rest of the stream. Only when the ITU-1364 standard came out to for digitally inserted vertical blanking data structures did the 0x00 and 0xFF codes get re-used in the packet synchronization structure.

The VIP_PARSER supports one configuration bit that, when enabled, changes all 0x00 to 0x01 and 0xFF to 0xFE in the vertical ancillary data. Another configuration bit, when enabled, changes all 0x00 to 0x01 and 0xFF to 0xFE in the active video portion of the input picture.

Generally, clipping is only desired for discrete sync input data captured from a NTSC/PAL decoder type of device which does not follow pixel range rules. For Discrete Sync input, the possibility to clip inputs to legal values exists. Clipping is desired if the picture sent to DRAM will be streamed out of the IC again usin an ITU-656/BT.1120 style output port. If the picture is processed inside the SOC before it is streamed out, then the processing procedure or the output streaming hardware needs to ensure that illegal values are not in the stream.

For Embedded Sync input, illegal values should not exist except for the ITU-1364 data sync sequence 00-FF-FF. Otherwise, the VIP Parser cannot determine good EAV/SAV. In the hardware, clipping is allowed for Embedded Sync streams even though it is not particularly a useful feature.

Note that if the clipping is enabled for ancillary data, the post processing software will never be able to find a data packet sync header, since the 00-FF-FF sequence will be changed to 01-FE-FE.

For 24-bit YUV, clipping is done on each 8 bit channel. If data[23:16]==0xFF, the clipped value will be 0xFE. If data[23:16]==0x00, the clipped value will be 0x01. Likewise, clipping is done for data bit ranges 15:8 and 7:0

From a software point of view, clipping should never be enabled for 24-bit RGB. RGB should use the full 8-bit quantization range for each color component. The hardware, however, will clip RGB in active video, if VIP_MAIN[5] CLIP_ACTIVE='1'. The clipping will be done on each 8-bit channel as described for 24-bit YUV.

9.4.5.16 Current and Last FID Value

The FID values for the current field or frame are reported in the Status Registers. When a new field or frame enters, the current FID values are saved into the previous FID status registers and the new FID value is loaded into the current FID register.

Following a reset, the previous and current FID status registers are set to '1.' The first two fields or frames are ignored. On the third input field or frame after a reset, the previous FID is loaded with the current FID ('1'), and the current FID is loaded with the actual FID. By the fourth field or frame after a reset, both the previous and current FID values should represent the values found in the input stream.

The FID values are reported for each camera source in both Pixel Input Clock Domains.

9.4.5.17 Disable Handling

A feature was defined for the case of single stream (either discrete sync or embedded sync) input handling where VIP_PORT_A[8] ENABLE for Port A and VIP_PORT_B[8] ENABLE for Port B is taken inactive. The single stream case was deemed more important than the multi-stream one since the output of the VIP_PARSER may be used to drive the Scaler module. The Scaler needs to work on a frame boundary (startframe to endframe) or it may lock up without a reset. The goal of the disable handling for the single stream case is to complete a field or frame of output data to the downstream module. Then, upon enabling of the port again, the system should start up properly without a need to reset the individual modules within the VIP instance.

In this scenario, suppose the VIP_PARSER has been processing a single input stream. Then, ENABLE is brought inactive. The VIP_PARSER will continue to output data downstream until it sends out an endframe pixel and the downstream module accepts the endframe pixel.

9.4.5.18 Picture Size Interrupt

Each VIP port can be set up to trigger an interrupt if the picture size varies from a pre-programmed expected picture size. This interrupt is supported only for the Active Video portion of the input video and not for the Vertical Ancillary portion. Also this interrupt is only support for source number 0 in multi channel capture.

The interrupts are named PrtASrc0Size and PrtBSrc0Size. They are described in Table 9-19, *VIP_PARSER Interrupt Events*

For Port A, the expected active video picture size values are programmed in VIP_XTRA_PORT_A[11:0] SRC0_NUMLINES and VIP_XTRA_PORT_A[27:16] SRC0_NUMPIX. For PortB, the expected active video picture size values are programmed in VIP_XTRA_PORT_B[11:0] SRC0_NUMLINES and VIP_XTRA_PORT_B[27:16] SRC0_NUMPIX.

Note

Picture Size Interrupt reflects the Active Video going into the DRAM. If cropping is enabled for Srcnum=0, the Picture Size is the post-cropped size.

9.4.5.19 Discrete Sync Signals

External ICs generally produce discrete sync interface signals seen in Figure 9-58.

VBLNKrepresents the vertical blanking interval. Generally, vertical blanking is at the top of a NTSC/PAL field. In certain standards, the last few lines of a field or frame are in vertical blanking in addition to the beginning few lines in the following field or frame.

VSYNC is the vertical sync indicator. VSYNC is active during a portion of the vertical blanking. For NTSC and PAL, since these standards define an odd number of lines for a field pair, VSYNC can be use in conjunction with HSYNC to determine FID polarity. Generally, VSYNC is defined to transition inactive to active sometime during vertical blanking. This signal then transitions to an inactive state before the end of vertical blanking. Certain standards define the line numbers where VSYNC transitions.

HBLNK is the horizontal blanking interval for each line. The horizontal blanking is the same number of pixels whether the line is in the active video region or in the vertical blanking region of the scan.

ACTVID is the region of a line that is active video. It is the inverse of the HBLNK signal. The number of pixels in the ACTVID region is the same for a line in vertical blanking as a line in active video. ACTVID(1) is a situation where the signal toggles in vertical blanking as well as active video. ACTVID(2) shows the signal toggling only in non-vertical blanking regions. Once ACTVID transitions active, it stays active for every PIXCLK until the end of the line.

HSYNC transitions from inactive to active for the first pixel of each line, which is a horizontal blanking pixel. HSYNC will transition to the inactive state before the end of the line. HSYNC is similar to HBLNK in that they both transition active on the same PIXCLK cycle. However, HBLNK transitions inactive at the end of the horizontal blanking period. HSYNC can transition inactive either before or after the horizontal blanking period.

Group 1 signals define the vertical separation between fields or frames. Group 2 signals define the separation between lines. One of the Group 1 signals can be tied to the VSYNC input. The ACTVID(1) and ACTVID(2) signals from Group 2 are tied to the ACTVID input. One of the other two Group 2 signals, HBLNK or HSYNC, can be tied to the HSYNC input.

VIP_PORT_A[15] USE_ACTVID_HSYNC_N for Port A and VIP_PORT_B[15] USE_ACTVID_HSYNC_N for Port B defines whether the line separation method uses the signal from the ACTVID or the HSYNC input of the VIP_PARSER module.

VIP_PORT_A[22] DISCRETE_BASIC_MODE for Port A VIP_PORT_B[22] DISCRETE_BASIC_MODE for Port B determines whether discrete sync works as described in Section 9.4.5.6 *Input Data Interface* or whether a "basic mode" input handler is invoked.

By choosing one signal from Group 1 and one signal from Group 2, there should be a way to capture the external data.

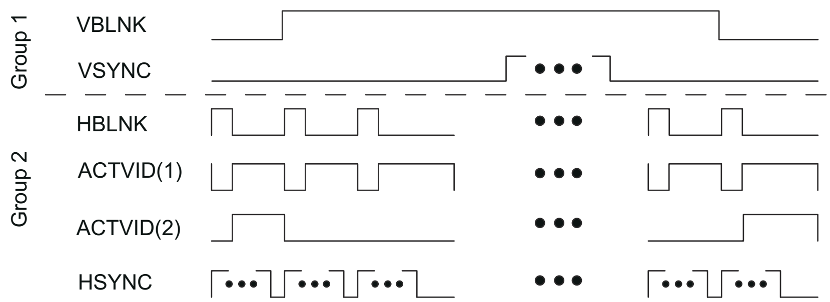


Figure 9-58. Generic External Sync Signals

Note

1. VIP_PARSER module defines three discrete sync control signals: ACTVID, HSYNC, and VSYNC.
2. In order to capture external data PIXCLK must never stop, for either horizontal or vertical blanking.

9.4.5.19.1 VBLNK and HBLNK

Figure 9-59 shows VBLNK from Group 1 and HBLNK from Group 2 being used. In this case, set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='0'. Vertical Ancillary lines will be sent to the Vertical Ancillary output at Active Video will be sent out the Active Video output.

If DISCRETE_BASIC_MODE='1' is chosen, all lines including vertical blanking ones will be sent to the Active Video buffer. Since a line is delineated by HBLNK and HBLNK toggles in vertical blanking as well as active video, every incoming pixel will be save to the Active Video buffer.

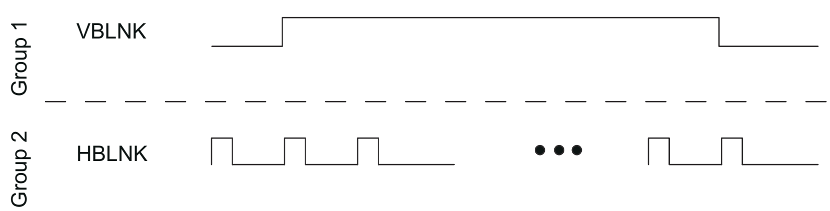


Figure 9-59. vblnk and hblnk

9.4.5.19.2 BLNK and ACTVID (1)

Figure 9-60 shows VBLNK from Group 1 and ACTVID(1) from Group 2 being used. ACTVID is toggling during Vertical Blanking. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='0'. Vertical Ancillary lines will be sent to the Vertical Ancillary output and Active Video will be sent out the Active Video output.

If DISCRETE_BASIC_MODE='1' is chosen, all lines will be sent to the Active Video output.

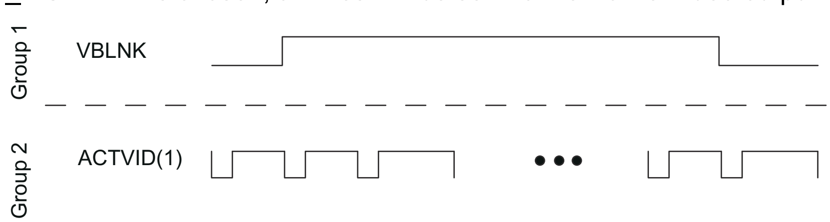


Figure 9-60. VBLNK and ACTID (1)

9.4.5.19.3 VBLNK and ACTVID(2)

Figure 9-61 shows VBLNK from Group 1 and ACTVID(2) from Group 2 being used. ACTVID is not toggling during Vertical Blanking. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='1'. Since there are no line sync/clocking signals in the Vertical Blanking period, only Active Video lines will be sent to the Active Video output.

If DISCRETE_BASIC_MODE='0' is set, then the hardware will lock up as there is no way for it to determine a frame boundary.

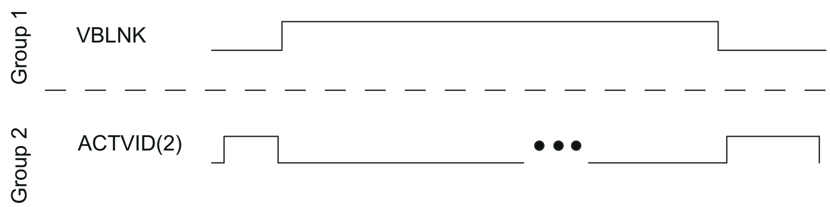


Figure 9-61. VBLNK and ACTVID(2)

9.4.5.19.4 VBLNK and HSYNC

Figure 9-62 shows VBLNK from Group 1 and HSYNC from Group 2 being used. In this scenario, set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODEN='0'. Vertical Ancillary lines will be sent to the Vertical Ancillary output at Active Video will be sent out the Active Video output.

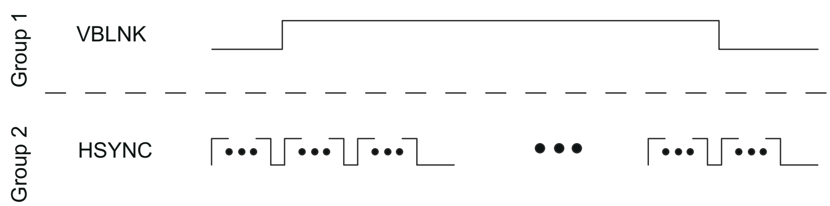


Figure 9-62. VBLNK and HSYNC

9.4.5.19.5 VSYNC and HBLNK

Figure 9-63 shows VSYNC from Group 1 and HBLINK from Group 2 being used. Set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='1'.

Also, no automatic parsing of vertical ancillary data will be performed so the Ancillary VPI port to the VPDMA should be disabled. All lines, including both vertical ancillary and active video, will appear in the Video DRAM buffer. Lines starting after an inactive to active transition on VSYNC will delineate a start of frame. Every data element strobed on the Pixel clock's active edge will be stored in the Active Video Buffer.

In Figure 9-63, it is likely the HBLINK will toggle when VSYNC is active. In this case, setting USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='0' mean that those lines appearing under the active VSYNC will be sent to the Ancillary Data Buffer. All other captured lines will be sent to the Active Video Buffer.

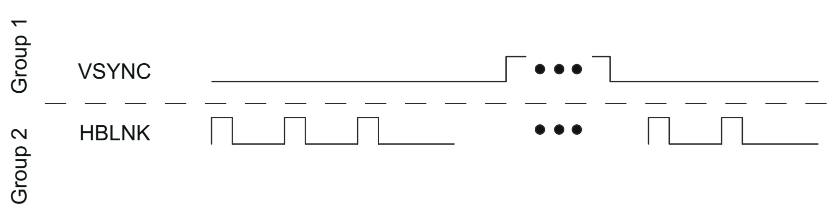


Figure 9-63. VSYNC and HBLNK

9.4.5.19.6 VSYNC and ACTIVID(1)

Figure 9-64 shows VSYNC from Group 1 and ACTIVID(1) from Group 2 being used. ACTIVID is toggling during the VBLNK interval. ACTIVID does not necessarily toggle during VSYNC. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='1'.

Also, no automatic parsing of vertical ancillary data will be performed so there is no activity on the Ancillary VPI port to the VPDMA. All lines, including both vertical ancillary and active video, will appear in the Video DRAM buffer. Lines starting after an inactive to active transition on VSYNC will delineate a start of frame. Lines are denoted by an inactive to active transition of ACTIVID. Only those pixels gated by an active ACTIVID will be saved.

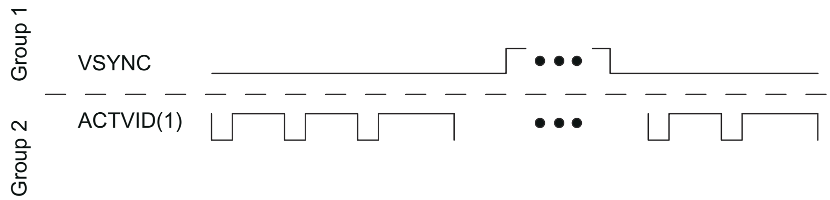


Figure 9-64. VSYNC and ACTIVID(1)

9.4.5.19.7 VSYNC and ACTIVID(2)

Figure 9-65 shows VSYNC from Group 1 and ACTIVID(2) from Group 2 being used. ACTIVID is not toggling during the entire VBLNK interval. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='1'.

Also, no automatic parsing of vertical ancillary data will be performed so the the Ancillary VPI port to the VPDMA should be turned off. All active video lines, since there are no vertical ancillary data lines, will appear in the Video DRAM buffer. Lines starting after an inactive to active transition on VSYNC will delineate a start of frame. Lines are denoted by an inactive to active transition of ACTIVID. Once a line starts, ACTIVID stays active for every pixel clock until the end of the line. Only those pixels gated by an active ACTIVID will be saved.

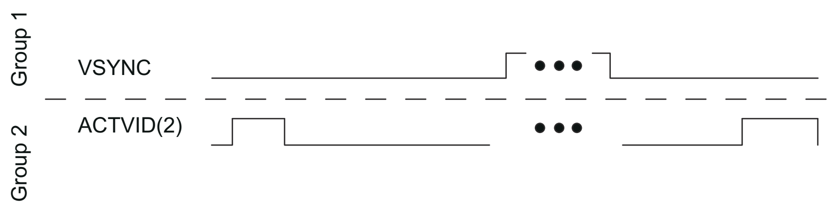


Figure 9-65. VSYNC and ACTIVID(2)

9.4.5.19.8 VSYNC and HSYNC

Figure 9-66 shows VSYNC from Group 1 and HSYNC from Group 2 being used. Set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='1'.

In the event that the machine is set to DISCRETE_BASIC_MODE='0', lockup will not occur as long as there is an HSYNC active transition during the time that VSYNC is active. This scenario is likely. However, if there is not at least one such transition on HSYNC, then the machine experiences a lock up. It cannot distinguish the end of one frame from the start of the next frame.

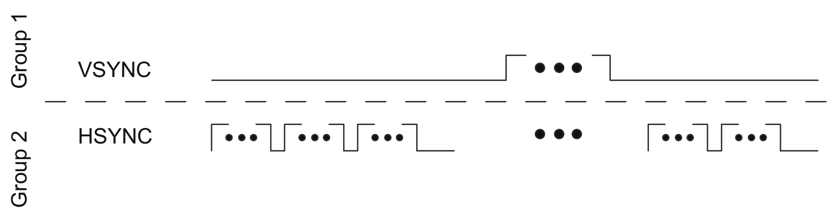


Figure 9-66. VSYNC and HSYNC

9.4.5.19.9 Line and Pixel Capture Examples

When DISCRETE_BASIC_MODE=‘0’, VBLNK is generally used. All the lines where the start of line is under an active VBLNK are sent to the Ancillary Data buffer. All the lines where the start of line is not under an active VBLNK are sent to the Active Video framebuffer. This situation is shown in Figure 9-67.

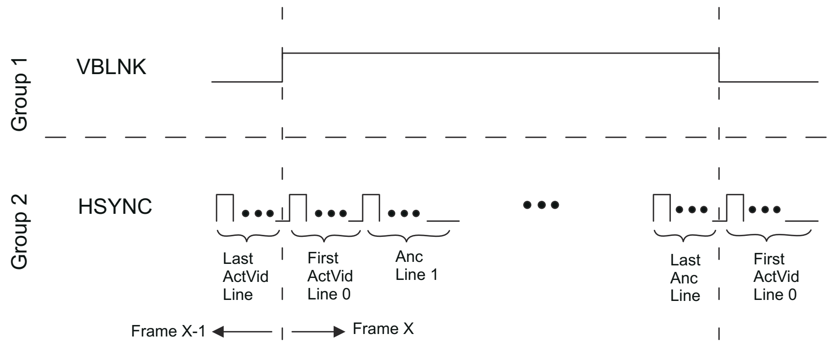


Figure 9-67. Ancillary and Active Video Line Determination

The start of line is the pixel represented by the inactive to active transition on HSYNC when USE_ACTVID_HSYNC_N = ‘1’. Figure 9-68 illustrates the delineation of a line when using USE_ACTVID_HSYNC_N = ‘1.’

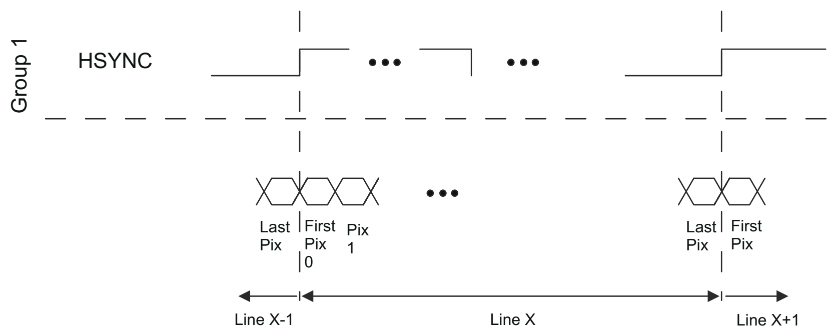


Figure 9-68. HSYNC Pixel Capture

The start of line is the pixel represented by the inactive to active transition on ACTVID when USE_ACTVID_HSYNC_N = ‘0.’ Note that ACTVID stays active for the entire duration of active video portion of the line. This scenario is shown in Figure 9-69

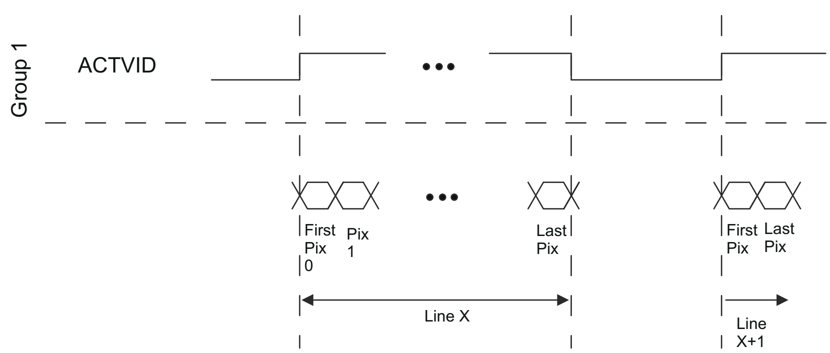


Figure 9-69. ACTVID Pixel Capture

In 8-bit mode, the 4:2:2 YUV input color component order is Cb, Y followed by Cr and Y. For 16-bit and 24-bit input modes, all the components are sent in the same cycle.

9.4.5.20 VIP Overflow Detection and Recovery

It is possible that an overflow can occur in the VIP_PARSER. Overflow detection is determined by reading the VIP_FIQ_STATUS register and checking for bits 8, 7, 5, 4, 3 and 2. If video is being captured, and any of these bits are set, it indicates that not all of the incoming video data was sent to DDR meory. VIP overflow can be caused by one of the following:

1. External pixel clock is faster than processing clock
2. DDR bandwidth is temporarily over-consumed
3. VIP scaler is being used inline with external video input, and is upscaling.
   - VIP scaler in this use case can only be used for downscaling
4. VIP scaler is being used inline with external video input, but has not been configured with scaler coefficients
   - VIP scaler will not accept video input if it is not first configured with scaler coefficients. This will cause overflow
5. VIP scaler is being used inline, but has not been enabled
6. External cables are connected or disconnected while the system is running, resulting in corrupted video streams going into the VIP
7. Bad external video cable, which causes corrupted video streams going into the VIP

Items 6 and 7 above are typically seen as noise events, where it is likely that multiple horizontal syncs per line and/or multiple vertical syncs per frame will be observed. These result in high peak throughput requirements, leading to DDR bandwidth being temporarily over-consumed, and thus VIP overflow.

The high level recovery method for VIP overflow on Port A is outlined in the steps below. Port B is similar.

1. Set VIP_XTRA6_PORT_A[31:16] YUV_SRCNUM_STOP_IMMEDIATELY = 0xFFFF_FFFF
2. Set VIP_XTRA6_PORT_A[15:0] ANC_SRCNUM_STOP_IMMEDIATELY = 0xFFFF_FFFF
3. Set VIP_PORT_A[8] ENABLE = 0
4. Set VIP_PORT_A[7] CLR_ASYNC_FIFO_RD and VIP_PORT_A[6] CLR_ASYNC_FIFO_WR to 1
5. Set VIP_PORT_A[23] SW_RESET to 1
6. Reset other VIP modules
   - For each module used downstream of VIP_PARSER, write 1 to the bit location of the VIP_CLKC_RST register which is connected to VIP_PARSER
7. Abort VPDMA channels
   - Write to list attribute to stop list 0
   - Write to list address register location of abort list
   - Write to list attribute register list 0 and size of abort list
8. Set VIP_PORT_A[23] SW_RESET to 0
9. Un-reset other VIP modules
   - For each module used downstream of VIP_PARSER, write 0 to the bit location of the VIP_CLKC_RST register which is connected to VIP_PARSER
10. (Delay)
11. SC coeff downloaded (if VIP_SCALER is being used)
12. (Delay)
13. Set VIP_XTRA6_PORT_A[31:16] YUV_SRCNUM_STOP_IMMEDIATELY = 0x0000_0000
14. Set VIP_XTRA6_PORT_A[15:0] ANC_SRCNUM_STOP_IMMEDIATELY = 0x0000_0000

15. Set VIP_PORT_A[8] ENABLE = 1

16. Set VIP_PORT_A[7] CLR_ASYNC_FIFO_RD and VIP_PORT_A[6] CLR_ASYNC_FIFO_WR to 0

9.4.6 VIP Color Space Converter (CSC)

The Color Space Converter (CSC) module is used to convert video data from one color space to another with nine programmable integer multipliers.

9.4.6.1 CSC Features

- All parameters are programmable
- Each parameter is configurable in signed 13-bits
- Support for Bypass Mode

9.4.6.2 CSC Functional Description

The conversion between the different color spaces requires addition and multiplication operations on color and intensity components. The mathematical expression of the conversion can be written as:

```
Y   = A0*R + B0*G + C0*B + D0
Cb  = A1*R + B1*G + C1*B + D1
Cr  = A2*R + B2*G + C2*B + D2
```

Color space coeficients are set through the following registers:

- For luma component:
  - VIP_CSC00[12:0] A0
  - VIP_CSC00[28:16] B0
  - VIP_CSC01[28:16] C0
  - VIP_CSC04[27:16] D0
- For Cb component:
  - VIP_CSC01[28:16] A1
  - VIP_CSC02[12:0] B1
  - VIP_CSC02[27:16] C1
  - VIP_CSC05[11:0] D1
- For Cr component :
  - VIP_CSC03[12:0] A2
  - VIP_CSC03[27:16] B2
  - VIP_CSC04[12:0] C2
  - VIP_CSC05[27:16] D2

Using YUV to RGB conversion as an example: YUV represents one color space and RGB represents another color space. The conversion can be written in the matrix format shown in Figure 9-70.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} A0 & B0 & C0 \\ A1 & B1 & C1 \\ A2 & B2 & C2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} D0 \\ D1 \\ D2 \end{bmatrix}$$

Figure 9-70. Matrix Format

Since HDTV and SDTV have different conversion requirements, both conversions of RGB-to-YCbCr and YCbCr-to-RGB are described. The details of derivations of these matrixes will be given in the following subsections.

9.4.6.2.1 HDTV Application

9.4.6.2.1.1 HDTV Application with Video Data Range

The two equations presented in this section are for the HDTV application. The chromaticity parameters are defined by ITU-R709 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4646 & -0.0468 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 9-71. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5396 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -197 \\ 82 \\ -232 \end{bmatrix} D$$

Figure 9-72. Conversion from YCbCr to RGB

9.4.6.2.1.2 HDTV Application with Graphics Data Range

The two equations presented in this section are for the HDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R709 standard.

The input data ranges for these equations are as follows.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.1826 & 0.6142 & 0.0620 \\ -0.1006 & -0.3385 & 0.4392 \\ 0.4392 & -0.3990 & -0.0402 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 9-73. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1644 & -0.0003 & 1.7927 \\ 1.1644 & -0.2132 & -0.5329 \\ 1.1642 & 2.1125 & -0.0001 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -248 \\ 77 \\ -289 \end{bmatrix} D$$

Figure 9-74. Conversion from YCbCr to RGB

9.4.6.2.1.3 Quantized Coefficients for Color Space Converter in HDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for HDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 9-20. Quantized Coefficients of HDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Coefficient Names**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VIP_CSC00[12:0] A0	0.2126	218	0x00DA	A0(13-bit)	1	1024	0x0400
B0(13-bit)	VIP_CSC00[28:16] B0	0.7152	732	0x02DC	B0(13-bit)	0	0	0x0000
C0(13-bit)	VIP_CSC01[28:16] C0	0.0722	74	0x004A	C0(13-bit)	1.5396	1577	0x0629
A1(13-bit)	VIP_CSC01[28:16] A1	-0.1172	-120	0x1F88	A1(13-bit)	1	1024	0x0400
B1(13-bit)	VIP_CSC02[12:0] B1	-0.3942	-404	0x1E6C	B1(13-bit)	-0.1831	-187	0x1F45
C1(13-bit)	VIP_CSC02[27:16] C1	0.5114	524	0x020C	C1(13-bit)	-0.4577	-469	0x1E2B
A2(13-bit)	VIP_CSC03[12:0] A2	0.5114	524	0x020C	A2(13-bit)	1	1024	0x0400
B2(13-bit)	VIP_CSC03[27:16] B2	-0.4646	-476	0x1E24	B2(13-bit)	1.8142	1858	0x0742
C2(13-bit)	VIP_CSC04[12:0] C2	-0.0468	-48	0x1FD0	C2(13-bit)	0	0	0x0000
D0(12-bit)	VIP_CSC04[27:16] D0	0	0	0x000	D0(12-bit)	-197	-788	0xCEC
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	82	328	0x148
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-232	-928	0xC60

Table 9-21. Quantized Coefficients of HDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Coefficient Names**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VIP_CSC00[12:0] A0	0.1826	187	0x00BB	A0(13-bit)	1.1644	1192	0x04A8
B0(13-bit)	VIP_CSC00[28:16] B0	0.6142	629	0x0275	B0(13-bit)	-0.0003	0	0x0000
C0(13-bit)	VIP_CSC01[28:16] C0	0.062	63	0x003F	C0(13-bit)	1.7927	1836	0x072C
A1(13-bit)	VIP_CSC01[28:16] A1	-0.1006	-103	0x1F99	A1(13-bit)	1.1644	1192	0x04A8
B1(13-bit)	VIP_CSC02[12:0] B1	-0.3385	-347	0x1EA5	B1(13-bit)	-0.2132	-218	0x1F26
C1(13-bit)	VIP_CSC02[27:16] C1	0.4392	450	0x01C2	C1(13-bit)	-0.5329	-546	0x1DDE
A2(13-bit)	VIP_CSC03[12:0] A2	0.4392	450	0x01C2	A2(13-bit)	1.1642	1192	0x04A8
B2(13-bit)	VIP_CSC03[27:16] B2	-0.399	-409	0x1E67	B2(13-bit)	2.1125	2163	0x0873
C2(13-bit)	VIP_CSC04[12:0] C2	-0.0402	-41	0x1FD7	C2(13-bit)	-0.0001	0	0x0000

Table 9-21. Quantized Coefficients of HDTV Application with Graphics Data Range (continued)

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Coefficient Names**	**Real Number Format**	**Quantized Format**	**Hex Format**
D0(12-bit)	VIP_CSC04[27:16] D0	16	64	0x040	D0(12-bit)	-248	-992	0xC20
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	77	308	0x134
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-289	-1156	0xB7C

9.4.6.2.2 SDTV Application

9.4.6.2.2.1 SDTV Application with Video Data Range

The two equations presented in this section are for the SDTV application. The chromaticity parameters are defined by ITU-R601 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.172 & -0.339 & 0.511 \\ 0.511 & -0.428 & -0.083 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 9-75. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & -0.0003 & 1.3717 \\ 1 & -0.3365 & -0.6984 \\ 1 & 1.7336 & -0.0016 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -176 \\ 132 \\ -222 \end{bmatrix} D$$

Figure 9-76. Conversion from YCbCr to RGB

9.4.6.2.2.2 SDTV Application with Graphics Data Range

The two equations presented in this section are for the SDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R601 standard.

The input data ranges for these equations are as following.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 9-77. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1641 & -0.0018 & 1.5958 \\ 1.1641 & -0.3914 & -0.8135 \\ 1.1641 & 2.0178 & -0.0012 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -223 \\ 136 \\ -277 \end{bmatrix} D$$

Figure 9-78. Conversion from YCbCr to RGB

9.4.6.2.2.3 Quantized Coefficients for Color Space Converter in SDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for SDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 9-22. Quantized Coefficients of SDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	Registers	Real Number Format	Quantized Format	Hex Format	Coefficient Names	Real Number Format	Quantized Format	Hex Format
A0(13-bit)	VIP_CSC00[12:0] A0	0.299	306	0x0132	A0(13-bit)	1	1024	0x0400
B0(13-bit)	VIP_CSC00[28:16] B0	0.587	601	0x0259	B0(13-bit)	-0.0003	0	0x0000
C0(13-bit)	VIP_CSC01[28:16] C0	0.114	117	0x0075	C0(13-bit)	1.3717	1405	0x057D
A1(13-bit)	VIP_CSC01[28:16] A1	-0.172	-176	0x1F50	A1(13-bit)	1	1024	0x0400
B1(13-bit)	VIP_CSC02[12:0] B1	-0.339	-347	0x1EA5	B1(13-bit)	-0.3365	-345	0x1EA7
C1(13-bit)	VIP_CSC02[27:16] C1	0.511	523	0x020B	C1(13-bit)	-0.6984	-715	0x1D35
A2(13-bit)	VIP_CSC03[12:0] A2	0.511	523	0x020B	A2(13-bit)	1	1024	0x0400
B2(13-bit)	VIP_CSC03[27:16] B2	-0.428	-438	0x1E4A	B2(13-bit)	1.7336	1775	0x06EF
C2(13-bit)	VIP_CSC04[12:0] C2	-0.083	-85	0x1FAB	C2(13-bit)	-0.0016	-2	0x1FFE
D0(12-bit)	VIP_CSC04[27:16] D0	0	0	0x000	D0(12-bit)	-176	-704	0xD40
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	132	528	0x210

Table 9-22. Quantized Coefficients of SDTV Application with Video Data Range (continued)

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Coefficient Names**	**Real Number Format**	**Quantized Format**	**Hex Format**
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-222	-888	0xC88

Table 9-23. Quantized Coefficients of SDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Coefficient Names**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VIP_CSC00[12:0] A0	0.257	263	0x0107	A0(13-bit)	1.1641	1192	0x04A8
B0(13-bit)	VIP_CSC00[28:16] B0	0.504	516	0x0204	B0(13-bit)	-0.0018	-2	0x1FFE
C0(13-bit)	VIP_CSC01[28:16] C0	0.098	100	0x0064	C0(13-bit)	1.5958	1634	0x0662
A1(13-bit)	VIP_CSC01[28:16] A1	-0.148	-152	0x1F68	A1(13-bit)	1.1641	1192	0x04A8
B1(13-bit)	VIP_CSC02[12:0] B1	-0.291	-298	0x1ED6	B1(13-bit)	-0.3914	-401	0x1E6F
C1(13-bit)	VIP_CSC02[27:16] C1	0.439	450	0x01C2	C1(13-bit)	-0.8135	-833	0x1CBF
A2(13-bit)	VIP_CSC03[12:0] A2	0.439	450	0x01C2	A2(13-bit)	1.1641	1192	0x04A8
B2(13-bit)	VIP_CSC03[27:16] B2	-0.368	-377	0x1E87	B2(13-bit)	2.0178	2066	0x0812
C2(13-bit)	VIP_CSC04[12:0] C2	-0.071	-73	0x1FB7	C2(13-bit)	-0.0012	-1	0x1FFF
D0(12-bit)	VIP_CSC04[27:16] D0	16	64	0x040	D0(12-bit)	-223	-892	0xC84
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	136	544	0x220
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-277	-1108	0xBAC

9.4.6.3 CSC Bypass Mode

CSC module can be bypassed by setting VIP_CSC05[28] BYPASS bit-field to 1.

9.4.7 VIP Scaler (SC)

This section describes the highly optimized video resizers, SC (scalers), in the VIP modules.

9.4.7.1 SC Features

- Independent vertical and horizontal up and down scaling
- Running average vertical down scaling for memory optimization
- Decimation and polyphase filtering for horizontal scaling
- Non-linear scaling for stretched/compressed left and right sides
- Input image trimmer for pan/scan support

- Pre-scaling peaking filter for enhanced sharpness
- Scale field as frame
- Interlacing of scaled output
- Full 1080p input and output support
- YCbCr422 input and output
- Maximum horizontal scaling ratio only limited by output line buffer (2047 pixels)
- Scaling filter Coefficient memory download via VPI (Video Port Interface) Control interface

9.4.7.2 SC Functional Description

Scaler takes in a 10-bit YCbCr 422 video frame from an upstream module, performs vertical/horizontal scaling and outputs a YCbCr422 scaled image to a next downstream module. All configurations are done via the MMR interface except for the scaler coefficient memory configuration that is done via the common VPI control interface bus by VPDMA. Figure 9-79 shows the high-level block diagram of the scaler module.

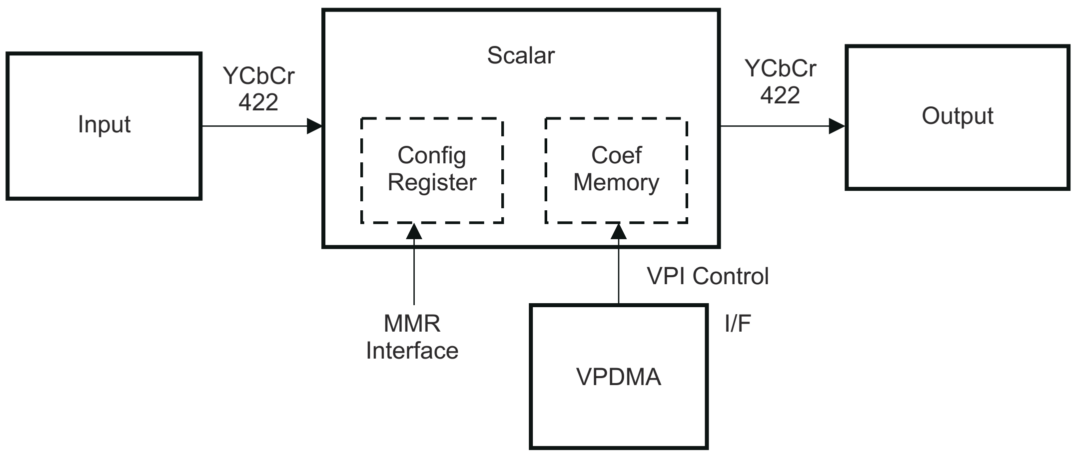


Figure 9-79. High Level Block Diagram

The SC is used in the video path and in all other video write-back data paths in the VIP module.

Scaling is performed in following three steps:

1. Trimming and Pre-peaking filtering
2. Vertical Scaling (Polyphase/Running Average Filter)
3. Horizontal polyphase scaling

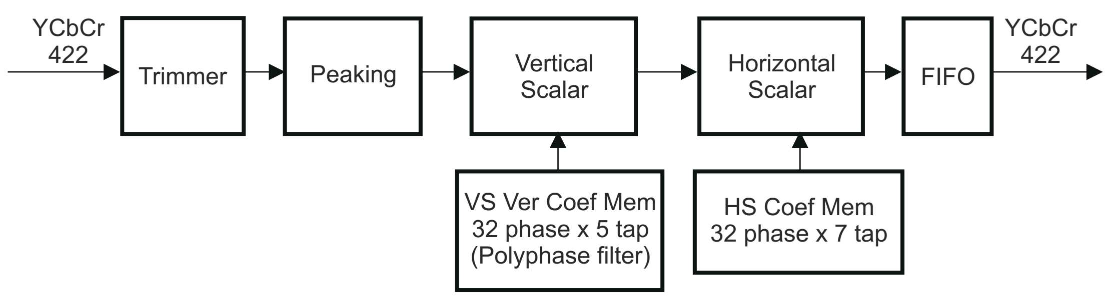


Figure 9-80. SC Block Diagram

9.4.7.2.1 Trimmer

The trimmer can be programmed to re-define a new source image within the input video frame sent from an upstream module before it is sent to the scaling sub-modules. This feature enables small area zoom out, source pan/scan, or removal of unwanted area in the video (such as black box / curtains / noisy line-21 video) without modifying the VPDMA parameters.

Horizontal and vertical offset is set through VIP_CFG_SC25[26:16] CFG_OFF_W and VIP_CFG_SC25[10:0] CFG_OFF_H registers.

Width and height are set through VIP_CFG_SC24[26:16] CFG_ORG_W and VIP_CFG_SC24[10:0] CFG_ORG_H registers.

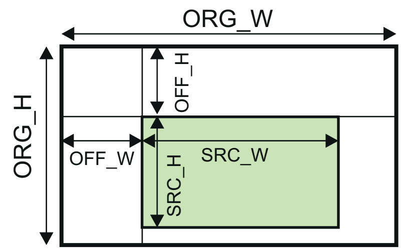


Figure 9-81. Input Image Trimming

Note

Width and height of the source image are global parameters and are set with VIP_CFG_SC5[22:12] CFG_SRC_W and VIP_CFG_SC5[10:0] CFG_SRC_H registers.

It is required that the input image frame (CFG_SRC_W x CFG_SRC_H) to be at least 32 × 32 (after trimming, if the trimming is enabled) to properly fill the input filter stage pipelines.

9.4.7.2.2

Note

Width and height of the source image are global parameters and are set with VIP_CFG_SC5[22:12] CFG_SRC_W and VIP_CFG_SC5[10:0] CFG_SRC_H registers.

It is required that the input image frame (CFG_SRC_W x CFG_SRC_H) to be at least 32 × 32 (after trimming, if the trimming is enabled) to properly fill the input filter stage pipelines.

9.4.7.2.3 Peaking

The peaking block increases the amplitude of high frequency luminance information in horizontal direction to increase the sharpness of a video image before it is scaled. As shown in Figure 9-82, the high-frequency luminance is increased using an 11-tap High-Pass filter with adjustable gain. The non-linear coring function removes low-level noise and the modified luminance is then added to the original luminance signal. The implementation details are shown in Figure 9-82.

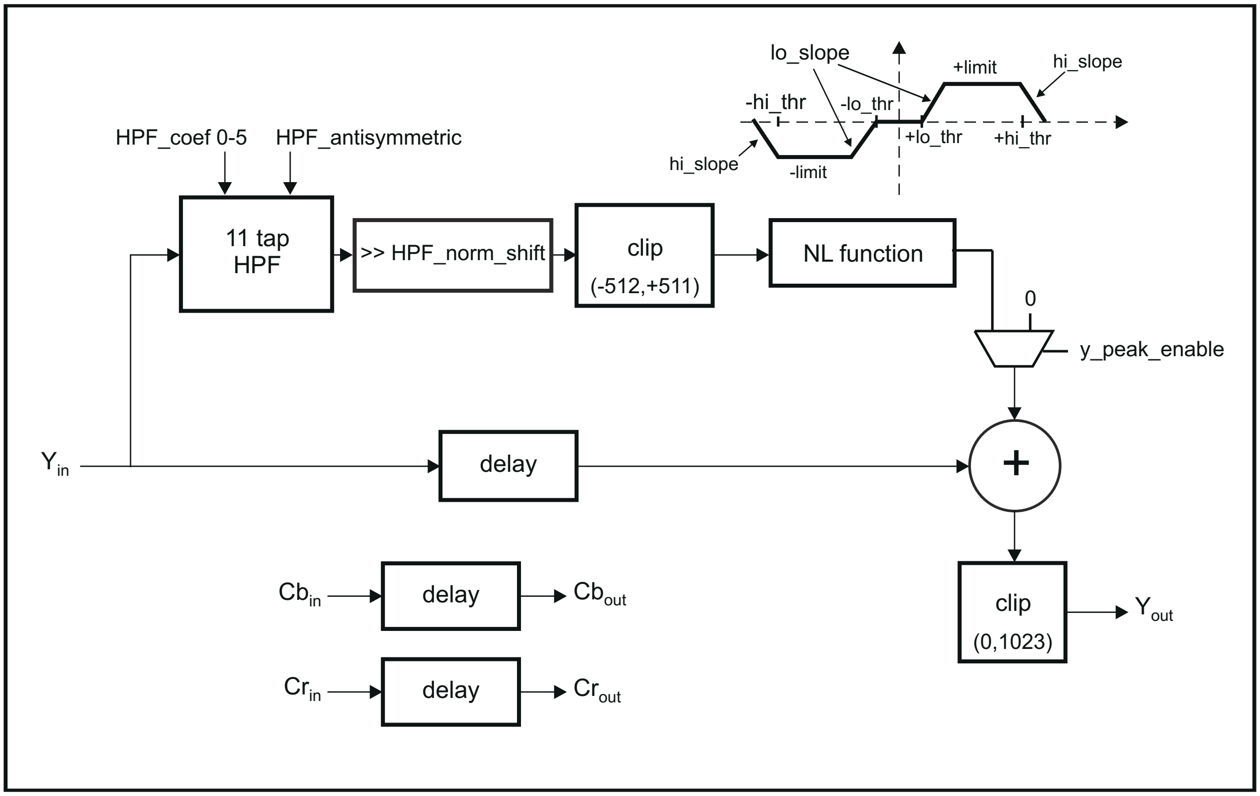


Figure 9-82. Filter Implementation and Parameter Description

Table 9-24. Parameter Description

Parameter	Description	Bits	Default
VIP_CFG_SC19[7:0] CFG_HPF_COEF0 to VIP_CFG_SC20[15:8] CFG_HPF_COEF5	FIR coefficients	8	[0 0 0-4 0 8]
VIP_CFG_SC20[18:16] CFG_HPF_NORM_SHIFT	Right shift	3	4
VIP_CFG_SC21[8:0] CFG_NL_LO_THR	Coring threshold	9	16
VIP_CFG_SC22[8:0] CFG_NL_HI_THR	High threshold	9	400
VIP_CFG_SC21[23:16] CFG_NL_LO_SLOPE	Lo slope= -CFG_NL_LO_SLOPE/16	8	16
VIP_CFG_SC22[18:16] CFG_NL_HI_SLOPE_SHIFT	Hi slope = $2^{(CFG\_NL\_HI\_SLOPE\_SHIFT-3)}$	3	4
VIP_CFG_SC20[28:20] CFG_NL_LIMIT	Clipping limit	9	200
VIP_CFG_SC0[14] CFG_Y_PK_EN	Control	1	0

Parameters for the Peaking filters are defined in VIP_CFG_SC19 through VIP_CFG_SC22 registers. The frequency responses of the peaking-filter with different sets of coefficients are shown in Figure 9-83. If the source of the input video is NTSC or PAL format, the peaking filter can be configured to reject the color subcarrier frequency.

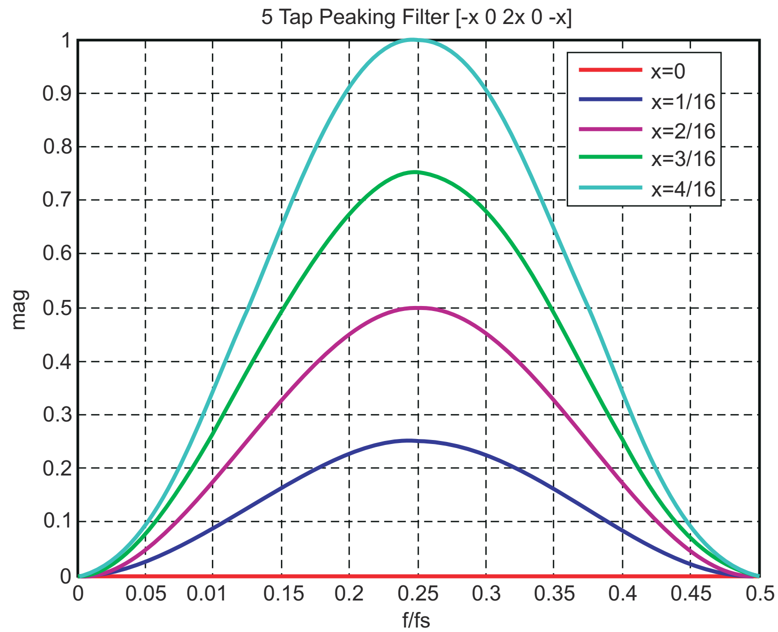


Figure 9-83. Peaking Filter at fs/4

9.4.7.2.4 Vertical Scaler

The vertical scaler has a polyphase (32-phase × 5-tap) filter and a running average filter as shown in Figure 9-84. While the polyphase filter can be used for any up-scaling and preferably downscaling to 3/16 scale factor, the running average filter is used only for downscaling to a ½ or less size. Selection between these two scalers is based on the user setting of VIP_CFG_SC0[4] CFG_USE_RAV parameter (CFG_USE_RAV= '0' for poliphase filter, and CFG_USE_RAV= '1' for running average filter), according to the user preference of the tradeoff between sharpness preserving and introduced artifacts.

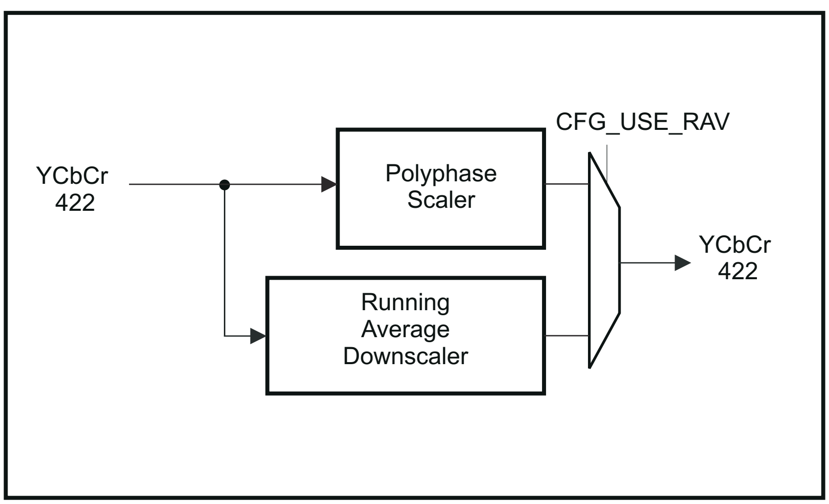


Figure 9-84. Vertical Scaler Block Diagram

9.4.7.2.4.1 Running Average Filter

When a poly-phase filter is used, usually it has to have many taps in order to achieve acceptable quality for very small downscaling ratio, which requires the use of many line buffers. In VIP, there is a weighted Running Average filter for downscaling when the scaling factor is small (for example, when the scaling ratio is less than 0.5). This highly optimized design requires only one line buffer for luma and one for chroma, which still achieving acceptable quality. The output of the running average filter is based on weighted average of pixels in the current and previous rows in vertical direction. Initializations of accumulators affect the weights.

9.4.7.2.4.2 Vertical Scaler Configuration Parameters

Table 9-25. Vertical Scaler Configuration Parameters

Parameter	Typical Value	Controls	Description
VIP_CFG_SC0[10] CFG_INTERLACE_I		Frame or Field	0 = progressive, 1 = interlace
VIP_CFG_SC0[0] CFG_INTERLACE_O			0 = progressive 1 = interlace
VIP_CFG_SC0[3] CFG_INVT_FID			Invert field ID input
VIP_CFG_SC0[4] CFG_USE_RAV		Scaler Mode	0 = use polyphase scaler 1 = use running average scaler
VIP_CFG_SC1[26:0] CFG_ROW_ACC_INC		Bilinear & Polyphase Scalers	For progressive in/progressive out: round(2^{16}*(srcH-1)/(tarH - 1)) For progressive_in/interlace_out: round(2^{16}*2*(srcH-1)/(2*tarH - 1)) For interlace_in/progressive_out: round(2^{16}*(2*srcH-1)/(2*(tarH - 1))) For interlace_in/interlace_out: round(2^{16}*(2*srcH - 1)/(2*tarH - 1)) For interlace in/out, srcH/tarH are number of field lines as specified in VIP_CFG_SC4/VIP_CFG_SC5 descriptions.
VIP_CFG_SC2[27:0] CFG_ROW_ACC_OFFSET	0		Initial row accumulator value for progressive frame and top field
VIP_CFG_SC3[27:0] CFG_ROW_ACC_OFFSET_B	0		Initial row accumulator value for bottom field
VIP_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR	4	Bilinear Scaler	Range for chroma soft switch based on pixel differences (max limit = 8)
VIP_CFG_SC13[21:12] CFG_CHROMA_INTP_THR	64		Threshold used in chroma soft switch based on pixel differences
VIP_CFG_SC13[9:0] CFG_SC_FACTOR_RAV		Running Average Scaler	Scale factor = round(1024 × tarH/srcH)
VIP_CFG_SC6[9:0] CFG_ROW_ACC_INIT_RAV			Initial row accumulator value for progressive frame and top field
VIP_CFG_SC6[19:10] CFG_ROW_ACC_INIT_RAV_B			Initial row accumulator value for bottom field

Note

Bi-linear scaler is not present in this device

9.4.7.2.5 Horizontal Scaler

The Horizontal scaler is implemented using a 32-phase × 7-tap polyphase filter preceded by two sets of 1/2x decimators. The general configuration of the Horizontal scaler is performed as follows:

- For up scaling, the input video is interpolated using the polyphase scaler.
- For downscaling, it is recommended that input video is decimated by 2 until the modified scale factor falls between 1/2 and 1. Then, a polyphase filter is configured with coefficients selected based on the mod_scale_factor calculated as shown below from one of nine different sets of coefficients: (8,9,10,11,12,13,14,15,16)/16.

```
if  (scale_factor>=1/2)  {
   mux=0;  mod_scale_factor=scale factor;
```

```
} else if  (scale_factor>=1/4) {
   mux=1; mod_scale_factor=2*scale_factor;
} else {
   mux=2; mod_scale_factor=4*scale_factor;
}
```

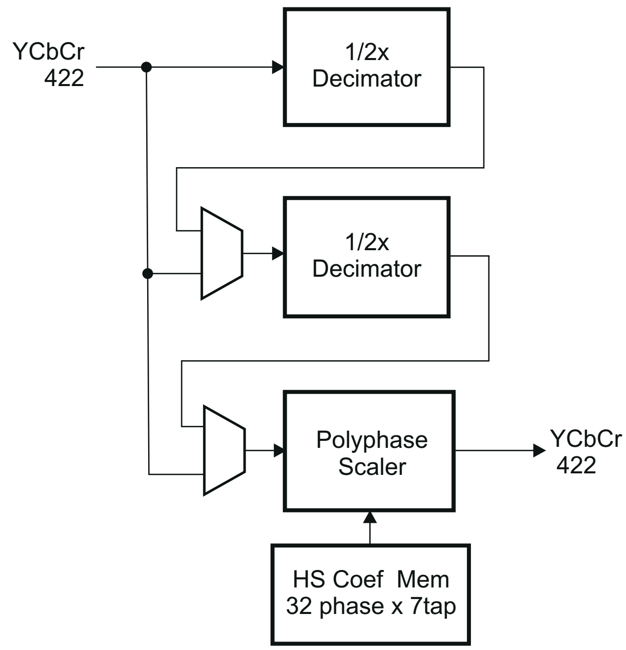


Figure 9-85. Horizontal Scaler Block Diagram

In auto mode (CFG_AUTO_HS == 1), scaler will operate as per above recommendation. In addition to this, for (CFG_AUTO_HS==1), polyphase filtering will be bypassed when (scale_factor == 1) or (scale_factor == ½) or (scale_factor == ¼). If CFG_AUTO_HS==0) is used, user must provide proper values for dcm_2x, dcm_4x, proper values for all inputs to polyphase filter in the registers as well as appropriate coefficient sets. There is no constraint on polyphase filtering in terms of scaling ratio. However, there are constraints on input and output image width for scaler. Frame dimensions are limited from 64x64 to 2047x2047.

9.4.7.2.5.1 Half Decimation Filter

The half-decimation filter is an 11-tap filter with following coefficients: (14, 0, -29, 0, 79, 128, 79, 0, -29, 0, 14). For processing left and right edge pixels, the first and last data are repeated to pre-fill and extend the filter data pipeline respectively. These coefficients are hard-coded into scaler design and user cannot modify these.

9.4.7.2.5.2 Polyphase Filter

The horizontal scaling is done by stepping across the target row and interpolating target pixels based on source pixels. Accumulator points to the source pixel that corresponds to the target pixel.

Figure 9-86 shows an up-scaling example.

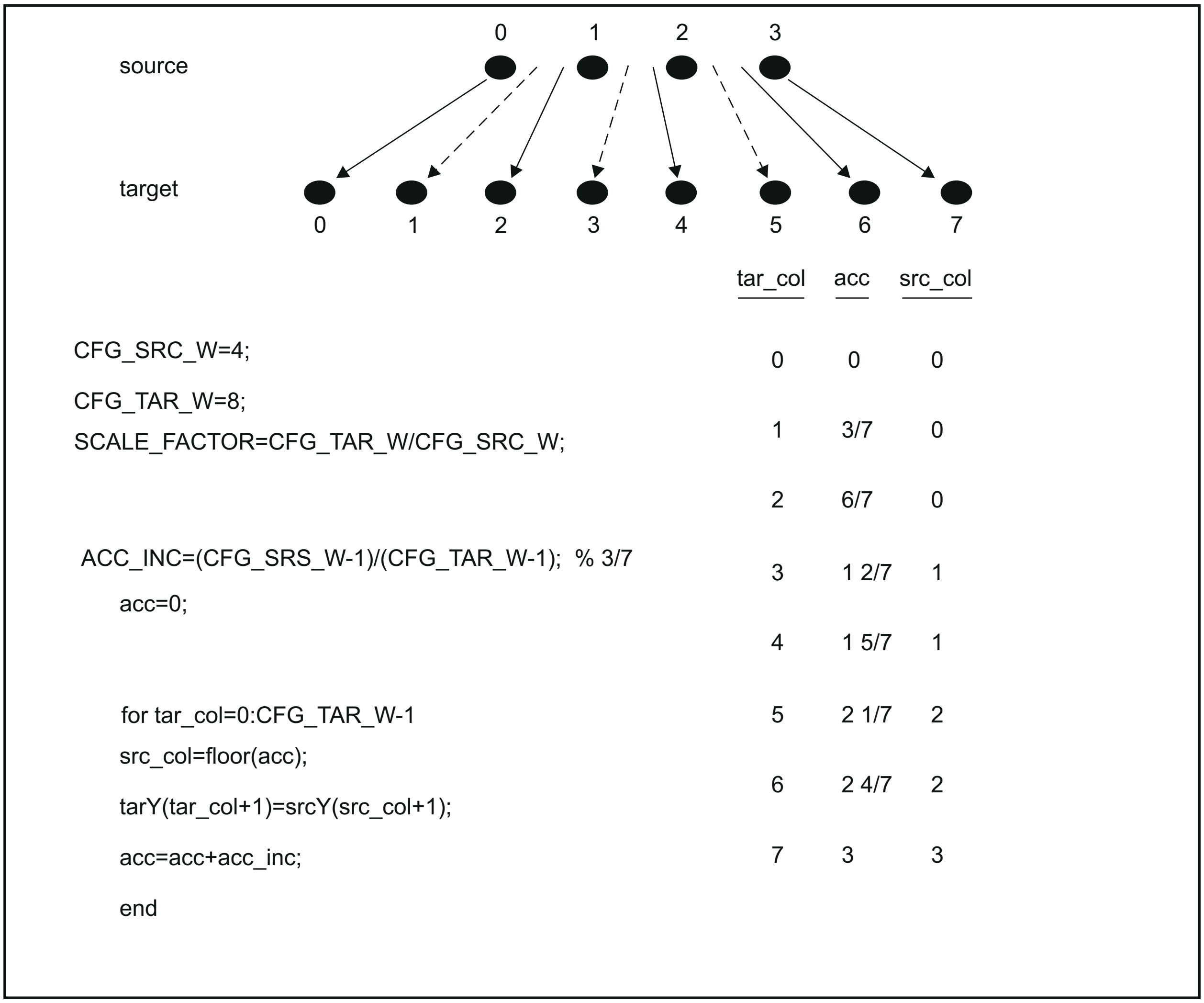


Figure 9-86. Polyphase Filtering Example

9.4.7.2.5.3 Nonlinear Horizontal Scaling

The horizontal scaler supports non-linear scaling to maintain the same aspect ratio for inner picture while stretching picture edges to fill the required resolution when capturing a 4 × 3 picture and fetching it as a 16 × 9 to memory. Non-linear scaling parameters are set with VIP_CFG_SC4[30:28] CFG_NLIN_ACC_INIT_U and VIP_CFG_SC4[26:24] CFG_LIN_ACC_INC_U registers. When setting up a non-linear scaling application, the inner picture is defined as the center square portion of the image with width and height equal to the height of the source image. Remaining side regions are then scaled non-linearly. Figure 9-87 shows a non-linear scaling case.

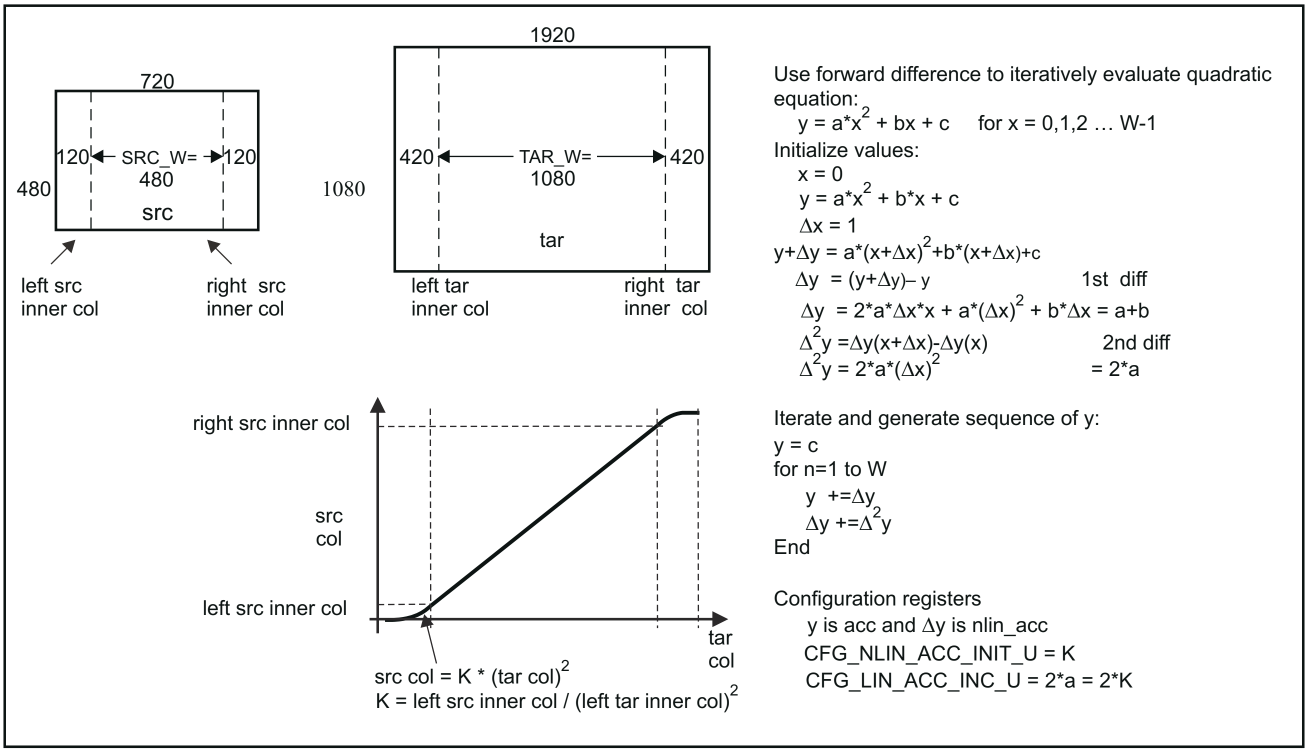


Figure 9-87. Non-linear Scaling Example

9.4.7.2.5.4 Horizontal Scaler Configuration Registers

Table 9-26. Register Group 1

<table>
<tr><th>Parameter</th><th>Controls</th><th colspan="4">Description</th></tr>
<tr><td>VIP_CFG_SC4[10:0]
CFG_TAR_H</td><td rowspan="2">Image Dimension</td><td colspan="4">Source Width</td></tr>
<tr><td>VIP_CFG_SC4[22:12]
CFG_TAR_W</td><td colspan="4">Target Width</td></tr>
<tr><td>VIP_CFG_SC0[1]
CFG_LINEAR</td><td rowspan="7">Scaler Mode</td><td colspan="4">If (linear == 1) SRC_Wi = SRC_W and TAR_Wi = TAR_W
Else SRC_W= SRC_H and TAR_W = TAR_H</td></tr>
<tr><td>VIP_CFG_SC0[2]
CFG_SC_BYPASS</td><td colspan="4">0 = enable scaler, 1 = bypass scaler</td></tr>
<tr><td rowspan="2">VIP_CFG_SC0[6]
CFG_AUTO_HS</td><td>CFG_AUTO_HS</td><td>CFG_DCM_2X</td><td>CFG_DCM_4X</td><td>Definition</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Polyphase scaling</td></tr>
<tr><td rowspan="2">VIP_CFG_SC0[7]
CFG_DCM_2X</td><td>0</td><td>0</td><td>1</td><td>Horizontal decimation by 4 and polyphase scaling</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Horizontal decimation by 2 and polyphase scaling</td></tr>
<tr><td>VIP_CFG_SC0[8]
CFG_DCM_4X</td><td>1</td><td>-</td><td>-</td><td>Automatic (selection of decimation filter is automatic)</td></tr>
</table>

Table 9-27. Register Group 2

<table>
<tr><th>Scale Factor</th><th>Decimation Usage</th><th>Control Register Bit</th></tr>
<tr><td>< 1/4</td><td>Decimation by 4</td><td rowspan="2">VIP_CFG_SC0[8] CFG_DCM_4X (set to 1 to enable decimation; disabled by default)</td></tr>
<tr><td>== 1/4</td><td>Decimation by 4</td></tr>
<tr><td>1/4 < and < 1/2</td><td>Decimation by 2</td><td rowspan="2">VIP_CFG_SC0[7] CFG_DCM_2X (set to 1 to enable decimation; disabled by default)</td></tr>
<tr><td>== 1/2</td><td>Decimation by 2</td></tr>
<tr><td>1/2 < and < 1</td><td>Bypassed</td><td rowspan="3">VIP_CFG_SC0[7] CFG_DCM_2X and CFG_DCM_4X (set to 0 to disable decimation; default value)</td></tr>
<tr><td>1</td><td>Bypassed</td></tr>
<tr><td>> 1</td><td>Bypassed</td></tr>
</table>

Table 9-28. Register Group 3

<table>
<tr><th>Parameter</th><th>Controls</th><th>Description</th></tr>
<tr><td>VIP_CFG_SC9[26:24]
CFG_LIN_ACC_INC</td><td rowspan="5">Polyphase Scaler</td><td>if upscaling then CFG_LIN_ACC_INC = round(2^{24}*(srcWi-1)/(tarWi-1))
elseif downscaling CFG_LIN_ACC_INC= round(2^{24}*(srcWi/n-1)/(tarWi-1))
where n=2 or 4</td></tr>
<tr><td>VIP_CFG_SC8[10:0]
CFG_NLIN_LEFT</td><td>if linear==1 CFG_NLIN_LEFT = 0
else CFG_NLIN_LEFT = (tarW - tarWi)/2</td></tr>
<tr><td>VIP_CFG_SC8[22:12]
CFG_NLIN_RIGHT</td><td>if linear==1 CFG_NLIN_RIGHT = tarW-1
else CFG_NLIN_RIGHT = Ltar + tarWi – 1</td></tr>
<tr><td>VIP_CFG_SC5[26:24]
CFG_NLIN_ACC_INC_U</td><td>if tarW/srcW >= 1 then
d = 0
if Ltar!=0 K = round[2^{24}*Lsrc/(Ltar*Ltar)] where Lsrc = (srcW-srcWi)/2
else K = 0
else
d = (tarW-1)/2
if Ltar!=0 K = round[2^{24} * Lsrc / (Ltar*(Ltar-2d))]
where Lsrc= (srcW-srcWi)/(2n) and n=1,2 or 4
else K = 0
CFG_LIN_ACC_INC = 2*K (negative for downscaling)</td></tr>
<tr><td>VIP_CFG_SC4[30:28]
CFG_NLIN_ACC_INIT_U</td><td>CFG_LIN_ACC_INC = K*(1-2*d)</td></tr>
</table>

Note

Source width and height variables for the polyphase filter are internally set with trimmer and decimation filter adjusted values.

9.4.7.2.6 Basic Configurations

Table 9-29 shows how the scaler should be configured based on the scale factor and the input/output mode.

Table 9-29. Scaler Configuration

Interlace		Mode(1)	VIP_CFG_SC5[10:0] CFG_SRC_H mod_srcH	VIP_CFG_SC4[10:0] CFG_TAR_H mod_tarH	Scale Factor
In	Out				
0	0	p->p	CFG_SRC_H	CFG_TAR_H	CFG_TAR_H/CFG_SRC_H
0	1	p->i	CFG_SRC_H	CFG_TAR_H/2	CFG_TAR_H/CFG_SRC_H
1	0	i->p	CFG_SRC_H/2	CFG_TAR_H	CFG_TAR_H/(CFG_SRC_H/2)
1	1	i->i	CFG_SRC_H/2	CFG_TAR_H/2	(CFG_TAR_H/2)/(CFG_SRC_H/2)

(1) p = progressive; i = interlaced

Table 9-30 shows how the vertical scaler should be configured based on the scale factor and the input/output mode.

Table 9-30. Vertical Scaler Configuration

Interlace		Mode(1)	VIP_CFG_SC9[26:24] CFG_ROW_ACC_INC/216	VIP_CFG_SC6[9:0] CFG_ROW_ACC_INIT_RAV/216	VIP_CFG_SC6[19:10] CFG_ROW_ACC_INIT_RAV_B/216
In	Out			Top	Bot
0	0	p->p	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	0
0	1	p->i	2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	(CFG_SRC_H-1)/(CFG_TAR_H-1)
1	0	i->p	1/2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	-0.5
1	1	i->i	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	[(CFG_SRC_H-1)/(CFG_TAR_H-1)-1]/2

(1) p = progressive; i = interlaced

9.4.7.2.7 Coefficient Memory

9.4.7.2.7.1 Overview

The scaler requires initialization of eight coefficient SRAMS prior to processing a video frame. These coefficients are stored in the external DDR memories and downloaded by the VPDMA. The VPDMA writes the coefficient data through the VPI Control Interface bus.

The eight coefficient SRAMs are:

- HS horizontal polyphase scaler, Luma and Chroma (7tap)
- VS vertical polyphase scaler, Luma and Chroma (5tap or 3tap)

9.4.7.2.7.2 Physical Coefficient SRAM Layout

Each of the six legacy coefficient SRAMs is 32 phases × 91 bits. A single coefficient value is 13 bits. Therefore, one word of the SRAM contains 7 coefficient values as shown in Figure 9-88, and 224 coefficient values are stored in each SRAM.

Phase 0	C6	C5	C4	C3	C2	C1	C0
Phase 31	C223	C222	C221	C220	C219	C218	C217

Figure 9-88. SRAM Layout for 7tap Coefficient

The two vertical polyphase SRAMs are 32 phases × 65 bits. A single coefficient is 13 bits. Therefore, one word of SRAM contains 5 coefficient values as shown in Figure 9-89.

Phase 0	C4	C3	C2	C1	C0
Phase 31	C221	C220	C219	C218	C217

Figure 9-89. SRAM Layout for 5tap Coefficient

9.4.7.2.7.3 Scaler Coefficients Packing on 128-bit VPI Control I/F

A coefficient value is 13 bits wide and takes a single half word. Thus, one VPI Control write carries one phase's worth of coefficients. The last half-word in a quad-word is not used for 7tap coefficients.

127 124 112 108 96 92 80 76 64 60 48 44 32 28 16 12 0

x	Unused	x	C6	x	C5	x	C4	x	C3	x	C2	x	C1	x	C0

Figure 9-90. VPI Control I/F Coef Data Format (7tap)

The Vertical Polyphase scaler coefficients have only five or three values per phase, so the last three (or five) entries are unused. The Vertical Polyphase coefficient packing is shown in Figure 9-91 and Figure 9-92.

127 124 112 108 96 92 80 76 64 60 48 44 32 28 16 12 0

x	Unused	x	Unused	x	Unused	x	C4	x	C3	x	C2	x	C1	x	C0

Figure 9-91. VPI Control I/F Coef Data Format (5tap)

127 124 112 108 96 92 80 76 64 60 48 44 32 28 16 12 0

x	Unused	x	Unused	x	Unused	x	Unused	x	Unused	x	C2	x	C1	x	C0

Figure 9-92. VPI Control I/F Coef Data Format (3tap)

9.4.7.2.7.4 VPI Control I/F Memory Map for Scaler Coefficients

The memory map of the VPI Control I/F for the Scaler coefficients is shown in Figure 9-93. All coefficients can be filled up by a single VPI Control write command. The update address feature of VPI Control I/F enables individual access to a certain location of memories.

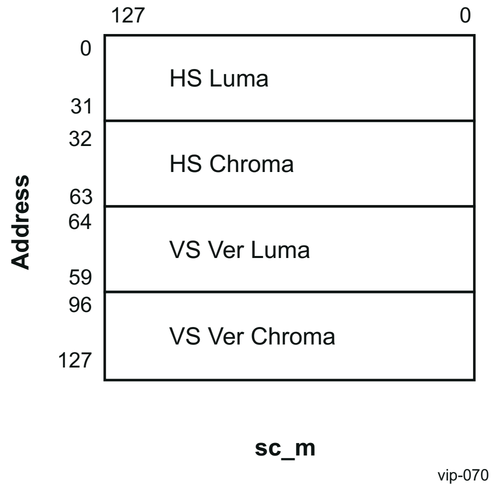


Figure 9-93. VPI Control I/F Memory Map (Write)

The module supports the VPI Control Read to read back the contents in the coefficient memories for debug purpose. Figure 9-94 shows the memory map of the VPI Control Read. As the VPI Control Read has only 32 bit data bus, it requires the word addressing while the write does the quad-word addressing.

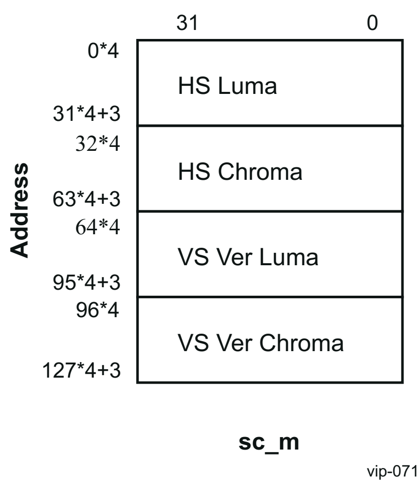


Figure 9-94. VPI Control I/F Memory Map (Read)

9.4.7.2.7.5 VPI Control Interface

VPDMA is used to configure the coefficient memories of scaler through VPI control interface. Since the coefficient memories are not shadowed (unlike the memory mapped registers), VPI control write access needs to be done only during the gap between video frame processing times. If a write request is made while the Scaler is active, the access will be held off by the hardware until the last data of the currently processed frame is sent out. Care must be given to the order of the DMA descriptors so that blocking of VPI control bus does not occur.

9.4.7.2.7.6 Coefficient Table Selection Guide

The scaler filter coefficient tables are pre-generated using a MATLAB® program for various scaling factor ranges. Table 9-31 provides a general selection guide table for coefficient data files.

The mentioned .dat files are available in Section 9.4.7.4.

Table 9-31. Coefficient Data Files

Scaler	Scale Factor	Coeff table
HS Polyphase Filter	All upscaling	Section 9.4.7.4.1.1 , *ppfcoef_scael_eq_1_32_phases_flip.dat*
	½ or ¼ down scaling	Section 9.4.7.4.1.1, ppfcoef_scale_eq_1_32_phases_flip.dat
	> 15/16	Section 9.4.7.4.1.9 , *ppfcoef_scale_eq_15div16_32_phases_flip.dat*
	> 14/16	Section 9.4.7.4.1.8 , *ppfcoef_scale_eq_14div16_32_phases_flip.dat*
	> 13/16	Section 9.4.7.4.1.7 , *ppfcoef_scale_eq_13div16_32_phases_flip.dat*
	> 12/16	Section 9.4.7.4.1.6 , *ppfcoef_scale_eq_12div16_32_phases_flip.dat*
	> 11/16	Section 9.4.7.4.1.5 , *ppfcoef_scale_eq_11div16_32_phases_flip.dat*
	> 10/16	Section 9.4.7.4.1.4 , *ppfcoef_scale_eq_10div16_32_phases_flip.dat*
	> 9/16	Section 9.4.7.4.1.3 , *ppfcoef_scale_eq_9div16_32_phases_flip.dat*
	> 8/16	Section 9.4.7.4.1.2 , *ppfcoef_scale_eq_8div16_32_phases_flip.dat* (1)
VS Polyphase Filter	Upscaling	Section 9.4.7.4.2.1 , *ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat*
	> 15/16	Section 9.4.7.4.2.6.8 , *ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat*
	> 14/16	Section 9.4.7.4.2.6.7 , *ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat*
	> 13/16	Section 9.4.7.4.2.6.6 , *ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat*
	> 12/16	Section 9.4.7.4.2.6.5 , *ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat*
	> 11/16	Section 9.4.7.4.2.6.4 , *ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat*
	> 10/16	Section 9.4.7.4.2.6.3 , *ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat*
	> 9/16	Section 9.4.7.4.2.6.2 , *ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat*
	> 8/16	Section 9.4.7.4.2.6.1 , *ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat*
	else	For down-scaling <8/16, RAV filter is recommended. In this case, coeffcients for vertical scaling need to be loaded.

(1) HS Scaler has two sets of ½ decimator to perform downscaling ratios below ½ and ¼.

9.4.7.3 SC Code

9.4.7.3.1 Generate Coefficient Memory Image

The following Perl script is used to generate a coefficient memory image for a given set of scaling factors.

```
#!/usr/local/bin/perl
#$dir="coef/";                      # directory which contains the coef files
#$cfg_file="sc_config1.cfg";        # configuration file name
#$spl_file="sc_config_supl.cfg";    # supplemental configuration file name
$cfg_file=$ARGV[0];                 # configuration file name
$spl_file=$ARGV[1];                 # supplemental configuration file name
$dir=$ARGV[2];                      # directory which contains the coef files
$coef_width=13;  # coef bit width
$coef_ntap=7;    # coef tap
$coef_nphase=32; # coef phase
$coef_norm=11;   # coef norm
#----------------------------------------------------------
# read config file to get srcH/tarH/interlace_i/interlace_o
```

```
#----------------------------------------------------------
open(INFILE, "<$cfg_file") or die "### ERROR: Cannot open $cfg_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcW/) {
    $srcW = $1;
  } elsif (m/([-0-9]+) +\/\/ +srcH/) {
    $srcH = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarW/) {
    $tarW = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarH/) {
    $tarH = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_in/) {
    $interlace_i = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_out/) {
    $interlace_o = $1;
  }
}
close(INFILE);
#----------------------------------------------------------
# read supplemental config file to get srcWi/tarWi from
#----------------------------------------------------------
open(INFILE,"<$spl_file") or die "### ERROR: Cannot open $spl_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcWi/) {
    $srcWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarWi/) {
    $tarWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +profile/) {
    $profile = $1; # 0:HIGH,1:MEDIUM,2:LOW
  }
}
close(INFILE);
#----------------------------------------------------------
# determine coef file based on the width/height
#----------------------------------------------------------
#VS
#$vsc_file0 = "mod_ppfcoef_scale_eq_1_32_phases_flip.dat";
$vsc_file0  = "ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat";
#VS VER
$mod_tarH = ($interlace_i == 0 && $interlace_o == 1)   $tarH<<1 : $tarH; if ($profile==2) {
  # LOW profile
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
} else {
    if ($mod_tarH >=($srcH>>1)) {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_3tap_flip.dat",$n);
    } else {
      $n = 0;
      $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
    }
  }
} else {
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat";
  } else {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_5tap_flip.dat",$n);
  }
}
# HS
if ($tarWi >= $srcWi) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} elsif ( ($tarWi == ($srcWi>>1)) || ($tarWi == ($srcWi>>2)) ) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} else {
if ($tarWi > ($srcWi>>1)) {
    $n = int(16.0*$tarWi/$srcWi);
  } elsif ($tarWi > ($srcWi>>2)) {
    $n = int(16.0*$tarWi/($srcWi>>1));
  } elsif ($tarWi >=($srcWi>>3)) {
    $n = int(16.0*$tarWi/($srcWi>>2));
  } else {
    $n = 0;
  }
  $hsc_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_flip.dat",$n);
```

```
}
#-----------------------------------------------------------
# write out the coef hex file
#-----------------------------------------------------------
&write_coef($hsc_file0);
&write_coef($hsc_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_file0);
&write_coef($vsc_file0);
sub write_coef {
  my ($filename) = @_;
  open(INFILE, "<$dir/$filename") or die "### ERROR: Cannot open $dir/$filename";
  $line=<INFILE>;
@val=split(' ',$line);
  $ntap=$val[0];
  $nphase=$val[1];
  $norm=$val[2];
for ($p=0;$p<$nphase;$p++) {
    $line=<INFILE>;@val=split(' ',$line);
    for($i=0;$i<$ntap;$i++) {
      if ($val[$i]<0) {
        $val[$i]+=(1<<$coef_width);
      }
    }
    undef(@coef);
    unshift(@coef, sprintf("%04x",$val[0]));
    unshift(@coef, sprintf("%04x",$val[1]));
    unshift(@coef, sprintf("%04x",$val[2]));
    unshift(@coef, sprintf("%04x",$val[3]));
    unshift(@coef, sprintf("%04x",$val[4]));
    unshift(@coef, sprintf("%04x",$val[5]));
    unshift(@coef, sprintf("%04x",$val[6]));
    unshift(@coef, sprintf("%04x",0));
    $coef=join("",@coef);
    print "$coef\n";
  }
  close(INFILE);
}
```

9.4.7.3.2 Scaler Configuration Calculation

The following C-code shows how configuration parameters are calculated:

```
// =======================================
// Required Input Parameter
// =======================================
//   srcW, srcH, tarW, tarH, srcWi, tarWi
//   input/output scan modes
//   Note:  srcH and tarH refer to number of lines in the frame even for interlace in/out
//          scaling.  Based on scaling scan mode input/output scan mode option,
//          heights are adjusted during internal calculations see mod_srcH and mod_tarH.
pixel_scale_factor=4;  // 10 bit pixel
hor_pixel_offset =0.0
  // ==========================================================================================
  // Peaking Filter Configuration
  // ==========================================================================================
  // -----------------------------------------------------------------
  // HPF Coef
  // -----------------------------------------------------------------
    y_peak_enable     =   0;
    peak_select=0; // 0=peak at fs/4  1=NTSC  2=PAL
    switch(peak_select) {
    case 0: {// peak at fs/4 and gain = 1
             HPF_coef0         =   0;
             HPF_coef1         =   0;
             HPF_coef2         =   0;
             HPF_coef3         =  -4;
             HPF_coef4         =   0;
             HPF_coef5         =   8; // mid tap
             HPF_norm_shift    =   4;
         break;
}
    case 1: {// NTSC: peak at 0.133*fs and gain=1
```

```
            HPF_coef0         =  -2;
            HPF_coef1         =  -8;
            HPF_coef2         =  -8;
            HPF_coef3         =  -2;
            HPF_coef4         =  12;
            HPF_coef5         =  16; // mid tap
            HPF_norm_shift    =   6;
            break;
    }
    case 2: {// PAL: peak at 0.163*fs and gain=1
            HPF_coef0         =   2;
            HPF_coef1         =  -4;
            HPF_coef2         = -11;
            HPF_coef3         =  -7;
            HPF_coef4         =   9;
            HPF_coef5         =  22; // mid tap
            HPF_norm_shift    =   6;
            break;
    }
    }
  // ----------------------------------------------------------------
  // NonLinear Coring Function typical values
  // ----------------------------------------------------------------
    NL_coring_thr     =  16;
    NL_limit          = 200;
    NL_lo_slope       =  16;
    NL_hi_thr         = 400;
    NL_hi_slope_shift =   4;
  // ============================================================================================
  // Edge Detection Configuration
  // ============================================================================================
    // edge detection
    confidence_default = 0; // 0  =use 5 tap polyphase filter for SC with ev_enable =0

    min_Gy_thr         =  64; // 64
    min_Gy_thr_range   =   3; // 3 power of 2
    gradient_thr       = 200; // 200
gradient_thr_range =   6; // 6 power of 2

    ev_thr = int(4.0*3.111+0.5); // edge vector soft switch threshold (3.2)
  // ============================================================================================
  // vertical scaler configuration
  // ============================================================================================
  // ----------------------------------------------------------------
  // vertical scaler typical parameters
  // ----------------------------------------------------------------
    invert_field_ID            =  0; // invert field ID input
    delta_ev_thr               =  1; // edge vector soft switch range
    ver_pixel_offset           =  0.0;
    uv_intp_thr                =  pixel_scale_factor*16;
    delta_y_thr                =  4; // luma soft switch range
    delta_uv_thr               =  4; // chroma soft switch range
  //
  //
  // ----------------------------------------------------------------
  // Vertical Scaler Mode Determination
  // ----------------------------------------------------------------
  //
  //  interlace
  //   in   out   mode mod_srcH mod_tarH         scale
  // ----- ------ ---- -------- --------   -----------------
  //   0      0   p->p  srcH     tarH            tarH/srcH
  //   0      1   p->i  srcH     tarH>>1         tarH/srcH
  //   1      0   i->p  srcH>>1  tarH            tarH/(srcH/2)
  //   1      1   i->i  srcH>>1  tarH>>1    (tarH/2)/(srcH/2)
  if (interlace_in)  mod_srcH=srcH>>1; // interlace
  else               mod_srcH=srcH;    // progressive
  if (interlace_out) mod_tarH=tarH>>1; // interlace
  else               mod_tarH=tarH;    // progressive
  // determine vertical scaler
  if ((interlace_in==0)&&(interlace_out==1)) {
    if (tarH>((1+srcH)>>1)) use_rav = 0;   // 1=use RAV scaler 0=use polyphase scaler
    else                    use_rav = 1;
  } else {
    if (mod_tarH>((1+mod_srcH)>>1)) use_rav = 0; // 1=use RAV scaler  0=use polyphase scaler
    else                            use_rav = 1;
```

```
}
  // ----------------------------------------------------------------
  // RAV or Polyphase parameters
  // ----------------------------------------------------------------
  if (use_rav) {  // downscale
    // ------------
    // --- RAV ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   0;
    if ((interlace_in==0)&&(interlace_out==1)) scale = double(tarH)/double(srcH);
    else                                        scale = double(mod_tarH)/double(mod_srcH);
    sc_factor_rav = int(1024.0*scale+0.5);
// Peter's method
    delta     = (1.0/scale-1.0)/2.0;
    int_part  = floor(delta);
    frac_part = delta-int_part;

    row_acc_init_rav   = int(1024*(scale+(1.0-scale)/
2.0)+0.5);                                              // top field
    row_acc_init_b_rav = int(1024*(scale+(1.0-2.0*frac_part)*(1.0-(1.0+2.0*int_part)*scale)/
2.0)+0.5); // bottom field
    row_acc_inc        = 0; // polyphase scaler
    row_acc_offset     = 0; // polyphase scaler
    row_acc_offset_b   = 0; // polyphase scaler
  } else { // upscale using polyphase scaler
    // ------------
    // --- PPF ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   1;
    sc_factor_rav      = 0;
    delta_rav          = 0;
    row_acc_init_rav   = 0;
    row_acc_init_b_rav = 0;
    // upscaler
    // interlace                                  row acc init value
        //  in   out mode      row acc inc      top          bottom
    // ---- ---- ---- -------------------- ---  -----------------------
    //   0    0  p->p     (srcH-1)/(tarH-1)  0             0
    //   0    1  p->i    2*(srcH-1)/(tarH-1)  0    (srcH-1)/(tarH-1)
    //   1    0  i->p 1/2*(srcH-1)/(tarH-1)  0             -0.5
    //   1    1  i->i     (srcH-1)/(tarH-1)  0    [(srcH-1)/(tarH-1)-1]/2
    row_acc_offset    = int(65536.0*ver_pixel_offset +0.5);  // progressive or top field
    if (interlace_in) {
if (interlace_out) {
          row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  (int(65536.0/2.0*(double(srcH-1)/(double(tarH-1))-
                            1.0)+0.5))+row_acc_offset;
      } else { // progressive out
          row_acc_inc      = int(65536.0*double(srcH-1)/(2.0*double(tarH-1))+0.5);
            if ((-0.5+row_acc_offset)<0.0) round_factor=-0.5;
            else                           round_factor= 0.5;
            row_acc_offset_b = int(65536.0*(-0.5)+round_factor)+row_acc_offset;
      }
    } else {   // progressive in
      if (interlace_out) {
        row_acc_inc      = int(65536.0*2.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5)+row_acc_offset;
      } else { // progressive out
          row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
           row_acc_offset_b = row_acc_offset;
      }
    }
  }
// ============================================================================================
  // Horizontal Scaler configuration
  // ============================================================================================
  // ----------------------------------------------------------------
  // horizontal scaler mode determination
  // ----------------------------------------------------------------
  auto_hs                = 1;
  dcm_2x                 = 0;
  dcm_4x                 = 0;
  hp_bypass              = 0;
  if (srcWi==srcW) linear = 1;
  else             linear = 0;
  // hor scaler parameters
```

```
  if (tarW>srcW) {               // upscale
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
} else if (tarW<=(srcW>>2)) {  // downscale by <=1/4
    mod_srcW  = srcW>>2;
    mod_srcWi = srcWi>>2;
  } else if (tarW<=(srcW>>1)) {  // downscale by <=1/2
    mod_srcW  = srcW>>1;
    mod_srcWi = srcWi>>1;
  } else {                       // downscale by <=1
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
  }
  // Not used any more:
  // hs_factor       = int(16.0*double(tarWi)/double(mod_srcWi)+0.5);  // hor scale factor (6.4)
  // -------------------------------------------------------------
  // Horizontal PolyPhase Settings  --
  // -------------------------------------------------------------
  lin_acc_inc     = int(16777216.0*double(mod_srcWi-1)/double(tarWi-1)+0.5);
  col_acc_offset  = int(16777216.0*hor_pixel_offset +0.5);
  nlin_left       = (tarW-tarWi)>>1;
  nlin_right      = nlin_left+tarWi-1;
  if (linear) {
    nlin_acc_inc    = 0;
    nlin_acc_init   = 0;
  } else {
     // ---------------------------------
     // Non-linear scaling configuration
     // ---------------------------------
    nlin_left_src   = (mod_srcW-mod_srcWi)>>1;

        if (tarWi>=srcWi) {    // upscale
      d               =  0.0;
      round_factor    =  0.5;
        } else {                 // downscale
      d               = (double(tarW)-1.0)/2.0;
      round_factor    = -0.5;
    }

      K               = 16777216.0*double(nlin_left_src)/(double(nlin_left)*double(nlin_left-2.0*d));
    nlin_acc_inc  = int(2.0*K+round_factor);
    nlin_acc_init = int(K*(1.0-2.0*d)+0.5);
  }
nlin_left_tar   = nlin_left;
  nlin_right_tar = nlin_right;
  // ==========================================================================================
  // Bypass Determination
  // ==========================================================================================
  // bypass
  if ((srcW==tarW)&&(srcWi==tarWi)&&(mod_srcH==mod_tarH)) sc_bypass = 1;
  else                                                     sc_bypass = 0;
  //
}
```

9.4.7.3.3 Typical Configuration Values

The following is the list of all scaler register fields that are set to constant values, representing typical settings:

VIP_CFG_SC0[3] CFG_INVT_FID = 0 (Field ID will be used without inversion)

VIP_CFG_SC0[5] CFG_ENABLE_EV = 1 (Field ID will be used without inversion)

VIP_CFG_SC0[6] CFG_AUTO_HS = 1 (The hardware will automatically decide, if current operation is up or down scaling. In down-scaling, it will also decide, if 2X or 4X decimation filter is needed)

VIP_CFG_SC0[7] CFG_DCM_2X = 0 (The 2X decimation filter is disabled)

VIP_CFG_SC0[8] CFG_DCM_4X = 0 (The 4X decimation filter is disabled)

VIP_CFG_SC0[11] CFG_ENABLE_SIN2_VER_INTP = 1 (Modified bilinear interpolation is used)

VIP_CFG_SC0[14] CFG_Y_PK_EN = 0 (Luma peaking is disabled)

VIP_CFG_SC0[15] CFG_TRIM= 1 (Trimming is enabled)

VIP_CFG_SC12[24:0] CFG_COL_ACC_OFFSET = 0 (No horizontal offset is involved)

VIP_CFG_SC13[21:12] CFG_CHROMA_INTP_THR = 64 (If the difference is less than this threshold, the interpolation of chroma should be done along edge direction. Otherwise, the interpolation of chroma should be done vertically)

VIP_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR = 4 (max limit=8)

VIP_CFG_SC18[24:16] CFG_CONF_DEFAULT = 0x100 (Defines confidence factor when edge detection is disabled (VIP_CFG_SC0[5] CFG_ENABLE_EV bit = 0))

VIP_CFG_SC19 = 0xFC000000

VIP_CFG_SC20 = 0x0C840800

VIP_CFG_SC21 = 0x00100010

VIP_CFG_SC22 = 0x00040190

9.4.7.4 SC Coefficient Data Files

9.4.7.4.1 HS Polyphase Filter Coefficients

9.4.7.4.1.1 ppfcoef_scale_eq_1_32_phases_flip.dat

```
    7  32  11
   31 -112  210 1790  210 -112   31
   28  -98  159 1787  264 -126   34
   25  -84  111 1779  320 -140   37
   22  -71   65 1767  379 -154   40
   19  -58   23 1750  439 -168   43
   16  -45  -17 1728  502 -181   45
   14  -33  -53 1701  565 -193   47
   11  -22  -86 1670  631 -205   49
    9  -11 -116 1635  696 -216   51
    7   -1 -142 1594  763 -225   52
    5    8 -166 1551  830 -233   53
    3   16 -186 1504  898 -240   53
    2   23 -204 1455  965 -245   52
    1   30 -218 1401 1031 -248   51
    0   35 -230 1345 1097 -249   50
   -1   40 -238 1286 1162 -248   47
   44 -244 1224 1224 -244   44    0
   47 -248 1162 1286 -238   40   -1
   50 -249 1097 1345 -230   35    0
   51 -248 1031 1401 -218   30    1
   52 -245  965 1455 -204   23    2
   53 -240  898 1504 -186   16    3
   53 -233  830 1551 -166    8    5
   52 -225  763 1594 -142   -1    7
   51 -216  696 1635 -116  -11    9
   49 -205  631 1670  -86  -22   11
   47 -193  565 1701  -53  -33   14
   45 -181  502 1728  -17  -45   16
   43 -168  439 1750   23  -58   19
   40 -154  379 1767   65  -71   22
   37 -140  320 1779  111  -84   25
   34 -126  264 1787  159  -98   28
```

9.4.7.4.1.2 ppfcoef_scale_eq_8div16_32_phases_flip.dat

```
7  32  11
  -28   61  542  898  542   61  -28
  -27   52  523  899  560   70  -29
  -26   44  505  898  578   79  -30
  -25   37  487  895  595   89  -30
  -24   30  468  892  613  100  -31
  -22   23  450  887  630  111  -31
  -21   17  432  883  647  122  -32
  -20   11  414  877  664  134  -32
  -19    6  396  871  680  146  -32
  -18    1  378  864  695  159  -31
```

```
   -16   -4  360  856  711  172  -31
   -15   -8  343  847  726  185  -30
   -14  -12  325  838  740  200  -29
   -13  -15  308  828  754  214  -28
   -12  -18  292  816  768  229  -27
   -10  -21  275  805  780  244  -25
   -23  258  789  789  258  -23    0
   -25  244  780  805  275  -21  -10
   -27  229  768  816  292  -18  -12
   -28  214  754  828  308  -15  -13
   -29  200  740  838  325  -12  -14
   -30  185  726  847  343   -8  -15
   -31  172  711  856  360   -4  -16
   -31  159  695  864  378    1  -18
   -32  146  680  871  396    6  -19
   -32  134  664  877  414   11  -20
   -32  122  647  883  432   17  -21
   -31  111  630  887  450   23  -22
   -31  100  613  892  468   30  -24
   -30   89  595  895  487   37  -25
   -30   79  578  898  505   44  -26
   -29   70  560  899  523   52  -27
```

9.4.7.4.1.3 ppfcoef_scale_eq_9div16_32_phases_flip.dat

```
7   32  11
   -33    8  547 1004  547    8  -33
   -31    0  525 1003  570   16  -35
   -29   -7  503 1001  592   25  -37
   -27  -13  481  998  614   34  -39
   -26  -19  459  995  636   44  -41
   -24  -25  437  990  658   55  -43
   -22  -29  414  983  679   67  -44
   -20  -34  393  976  700   79  -46
   -18  -38  371  968  721   91  -47
   -17  -41  350  959  742  104  -49
   -15  -44  330  948  761  118  -50
   -13  -46  309  936  780  133  -51
   -12  -48  289  924  799  148  -52
   -11  -50  270  911  817  163  -52
    -9  -51  250  897  833  180  -52
    -8  -52  232  882  850  196  -52
   -52  213  863  863  213  -52    0
   -52  196  850  882  232  -52   -8
   -52  180  833  897  250  -51   -9
   -52  163  817  911  270  -50  -11
   -52  148  799  924  289  -48  -12
   -51  133  780  936  309  -46  -13
   -50  118  761  948  330  -44  -15
   -49  104  742  959  350  -41  -17
   -47   91  721  968  371  -38  -18
   -46   79  700  976  393  -34  -20
   -44   67  679  983  414  -29  -22
   -43   55  658  990  437  -25  -24
   -41   44  636  995  459  -19  -26
   -39   34  614  998  481  -13  -27
   -37   25  592 1001  503   -7  -29
   -35   16  570 1003  525    0  -31
```

9.4.7.4.1.4 ppfcoef_scale_eq_10div16_32_phases_flip.dat

```
7   32  11
   -30  -46  542 1116  542  -46  -30
   -28  -52  515 1115  570  -39  -33
   -25  -57  488 1113  597  -32  -36
   -23  -62  462 1109  624  -24  -38
   -20  -65  435 1104  650  -15  -41
   -18  -69  409 1097  678   -5  -44
   -16  -71  383 1089  704    6  -47
   -14  -74  358 1081  730   17  -50
   -12  -75  333 1070  756   29  -53
   -11  -76  309 1058  782   42  -56
    -9  -77  285 1045  806   56  -58
```

```
   -8  -77  262 1030  831   71  -61
   -6  -77  239 1015  855   86  -64
   -5  -76  218  997  877  103  -66
   -4  -75  196  980  899  120  -68
   -3  -74  176  961  920  138  -70
  -72  156  940  940  156  -72    0
  -70  138  920  961  176  -74   -3
  -68  120  899  980  196  -75   -4
  -66  103  877  997  218  -76   -5
  -64   86  855 1015  239  -77   -6
  -61   71  831 1030  262  -77   -8
  -58   56  806 1045  285  -77   -9
  -56   42  782 1058  309  -76  -11
  -53   29  756 1070  333  -75  -12
  -50   17  730 1081  358  -74  -14
  -47    6  704 1089  383  -71  -16
  -44   -5  678 1097  409  -69  -18
  -41  -15  650 1104  435  -65  -20
  -38  -24  624 1109  462  -62  -23
  -36  -32  597 1113  488  -57  -25
  -33  -39  570 1115  515  -52  -28
```

9.4.7.4.1.5 ppfcoef_scale_eq_11div16_32_phases_flip.dat

```
7  32 11
  -19  -94  522 1230  522  -94  -19
  -17  -98  490 1230  555  -90  -22
  -14 -100  458 1227  587  -85  -25
  -12 -102  427 1223  620  -79  -29
  -10 -103  397 1217  652  -73  -32
   -8 -104  367 1209  685  -65  -36
   -6 -104  337 1199  717  -56  -39
   -4 -103  309 1187  749  -47  -43
   -3 -102  281 1174  781  -36  -47
   -1 -100  253 1159  812  -24  -51
    0  -98  227 1142  843  -11  -55
    1  -96  201 1124  874    3  -59
    1  -93  177 1105  903   18  -63
    2  -90  153 1084  932   34  -67
    2  -87  131 1062  961   51  -72
    3  -83  109 1038  987   69  -75
  -79   89 1014 1014   89  -79    0
  -75   69  987 1038  109  -83    3
  -72   51  961 1062  131  -87    2
  -67   34  932 1084  153  -90    2
  -63   18  903 1105  177  -93    1
  -59    3  874 1124  201  -96    1
  -55  -11  843 1142  227  -98    0
  -51  -24  812 1159  253 -100   -1
  -47  -36  781 1174  281 -102   -3
  -43  -47  749 1187  309 -103   -4
  -39  -56  717 1199  337 -104   -6
  -36  -65  685 1209  367 -104   -8
  -32  -73  652 1217  397 -103  -10
  -29  -79  620 1223  427 -102  -12
  -25  -85  587 1227  458 -100  -14
  -22  -90  555 1230  490  -98  -17
```

9.4.7.4.1.6 ppfcoef_scale_eq_12div16_32_phases_flip.dat

```
7  32 11
   -3 -132  486 1346  486 -132   -3
   -1 -132  449 1345  524 -131   -6
    1 -131  413 1342  562 -130   -9
    3 -130  378 1336  600 -127  -12
    4 -128  343 1328  639 -123  -15
    5 -125  309 1319  677 -119  -18
    6 -122  277 1306  716 -113  -22
    7 -118  245 1292  754 -106  -26
    8 -114  214 1276  793  -98  -31
    8 -109  185 1257  831  -89  -35
    9 -105  156 1237  869  -78  -40
    9 -100  130 1214  906  -66  -45
```

```
     9  -94  104 1190  942  -53  -50
     9  -89   79 1165  978  -38  -56
     8  -83   56 1138 1012  -22  -61
     8  -78   35 1108 1046   -4  -67
   -72   15 1081 1081   15  -72    0
   -67   -4 1046 1108   35  -78    8
   -61  -22 1012 1138   56  -83    8
   -56  -38  978 1165   79  -89    9
   -50  -53  942 1190  104  -94    9
   -45  -66  906 1214  130 -100    9
   -40  -78  869 1237  156 -105    9
   -35  -89  831 1257  185 -109    8
   -31  -98  793 1276  214 -114    8
   -26 -106  754 1292  245 -118    7
   -22 -113  716 1306  277 -122    6
   -18 -119  677 1319  309 -125    5
   -15 -123  639 1328  343 -128    4
   -12 -127  600 1336  378 -130    3
    -9 -130  562 1342  413 -131    1
    -6 -131  524 1345  449 -132   -1
```

9.4.7.4.1.7 ppfcoef_scale_eq_13div16_32_phases_flip.dat

```
7   32  11
    14 -154  435 1458  435 -154   14
    15 -150  393 1458  477 -157   12
    16 -146  353 1454  521 -160   10
    16 -141  314 1447  565 -161    8
    17 -135  276 1436  609 -161    6
    17 -129  239 1425  654 -161    3
    17 -123  204 1410  699 -159    0
    16 -116  170 1393  745 -156   -4
    16 -109  137 1373  790 -151   -8
    16 -102  107 1350  835 -146  -12
    15  -94   77 1325  879 -138  -16
    14  -87   50 1298  924 -130  -21
    13  -80   24 1269  968 -119  -27
    12  -72    0 1238 1010 -107  -33
    11  -65  -22 1204 1053  -94  -39
    10  -58  -43 1169 1093  -78  -45
   -52  -62 1138 1138  -62  -52    0
   -45  -78 1093 1169  -43  -58   10
   -39  -94 1053 1204  -22  -65   11
   -33 -107 1010 1238    0  -72   12
   -27 -119  968 1269   24  -80   13
   -21 -130  924 1298   50  -87   14
   -16 -138  879 1325   77  -94   15
   -12 -146  835 1350  107 -102   16
    -8 -151  790 1373  137 -109   16
    -4 -156  745 1393  170 -116   16
     0 -159  699 1410  204 -123   17
     3 -161  654 1425  239 -129   17
     6 -161  609 1436  276 -135   17
     8 -161  565 1447  314 -141   16
    10 -160  521 1454  353 -146   16
    12 -157  477 1458  393 -150   15
```

9.4.7.4.1.8 ppfcoef_scale_eq_14div16_32_phases_flip.dat

```
7   32  11
    27 -158  370 1570  370 -158   27
    27 -150  324 1568  417 -165   27
    26 -142  281 1563  465 -172   27
    25 -133  238 1555  515 -178   26
    24 -124  198 1543  565 -183   25
    23 -115  159 1527  616 -186   24
    22 -106  122 1510  667 -189   22
    21  -97   87 1489  719 -191   20
    19  -87   54 1464  772 -191   17
    18  -78   23 1437  824 -190   14
    16  -69   -6 1407  876 -187   11
    15  -60  -32 1373  927 -182    7
    13  -52  -57 1339  979 -176    2
```

```
    12  -44  -79 1300 1030 -168   -3
    11  -36  -99 1261 1079 -159   -9
     9  -28 -117 1218 1128 -147  -15
   -21 -134 1179 1179 -134  -21    0
   -15 -147 1128 1218 -117  -28    9
    -9 -159 1079 1261  -99  -36   11
    -3 -168 1030 1300  -79  -44   12
     2 -176  979 1339  -57  -52   13
     7 -182  927 1373  -32  -60   15
    11 -187  876 1407   -6  -69   16
    14 -190  824 1437   23  -78   18
    17 -191  772 1464   54  -87   19
    20 -191  719 1489   87  -97   21
    22 -189  667 1510  122 -106   22
    24 -186  616 1527  159 -115   23
    25 -183  565 1543  198 -124   24
    26 -178  515 1555  238 -133   25
    27 -172  465 1563  281 -142   26
    27 -165  417 1568  324 -150   27
```

9.4.7.4.1.9 ppfcoef_scale_eq_15div16_32_phases_flip.dat

```
7   32  11
    33 -143  294 1680  294 -143   33
    31 -132  246 1678  345 -155   35
    30 -121  199 1671  398 -165   36
    27 -109  154 1661  452 -175   38
    25  -97  112 1647  508 -185   38
    23  -86   72 1629  564 -193   39
    21  -75   35 1607  622 -201   39
    19  -64    0 1580  681 -207   39
    17  -53  -32 1551  740 -213   38
    15  -43  -61 1518  799 -217   37
    13  -33  -88 1481  859 -219   35
    11  -24 -113 1442  919 -220   33
     9  -15 -134 1399  978 -219   30
     8   -7 -153 1354 1036 -217   27
     6    0 -170 1307 1094 -212   23
     5    7 -184 1257 1150 -205   18
    13 -196 1207 1207 -196   13    0
    18 -205 1150 1257 -184    7    5
    23 -212 1094 1307 -170    0    6
    27 -217 1036 1354 -153   -7    8
    30 -219  978 1399 -134  -15    9
    33 -220  919 1442 -113  -24   11
    35 -219  859 1481  -88  -33   13
    37 -217  799 1518  -61  -43   15
    38 -213  740 1551  -32  -53   17
    39 -207  681 1580    0  -64   19
    39 -201  622 1607   35  -75   21
    39 -193  564 1629   72  -86   23
    38 -185  508 1647  112  -97   25
    38 -175  452 1661  154 -109   27
    36 -165  398 1671  199 -121   30
    35 -155  345 1678  246 -132   31
```

9.4.7.4.2 VS Polyphase Filter Coefficients

9.4.7.4.2.1 ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat

```
5   32  11
   -47  177 1788  177  -47
   -40  133 1785  225  -55
   -33   91 1778  276  -64
   -27   53 1765  330  -73
   -21   18 1747  386  -82
   -15  -13 1722  445  -91
   -11  -41 1693  507 -100
    -7  -66 1660  570 -109
    -3  -88 1622  635 -118
     0 -107 1579  703 -127
     2 -122 1532  771 -135
     4 -135 1482  839 -142
     5 -145 1428  909 -149
```

```
    6 -153 1371  978 -154
    7 -158 1310 1047 -158
    7 -161 1247 1116 -161
 -162 1186 1186 -162    0
 -161 1116 1247 -161    7
 -158 1047 1310 -158    7
 -154  978 1371 -153    6
 -149  909 1428 -145    5
 -142  839 1482 -135    4
 -135  771 1532 -122    2
 -127  703 1579 -107    0
 -118  635 1622  -88   -3
 -109  570 1660  -66   -7
 -100  507 1693  -41  -11
  -91  445 1722  -13  -15
  -82  386 1747   18  -21
  -73  330 1765   53  -27
  -64  276 1778   91  -33
  -55  225 1785  133  -40
```

9.4.7.4.2.2 ppfcoef_scale_eq_3_32_phases_flip.dat

```
5,   32,   11,
  130,  515,  758,  515,  130,
  121,  503,  757,  528,  139,
  113,  490,  756,  541,  148,
  105,  477,  755,  553,  158,
   97,  464,  753,  566,  168,
   90,  451,  751,  578,  178,
   83,  437,  749,  590,  189,
   76,  424,  746,  602,  200,
   69,  411,  743,  614,  211,
   63,  398,  739,  626,  222,
   57,  386,  734,  637,  234,
   52,  373,  729,  648,  246,
   46,  360,  725,  659,  258,
   41,  347,  719,  670,  271,
   37,  335,  713,  680,  283,
   32,  322,  707,  690,  297,
  314,  710,  710,  314,    0,
  297,  690,  707,  322,   32,
  283,  680,  713,  335,   37,
  271,  670,  719,  347,   41,
  258,  659,  725,  360,   46,
  246,  648,  729,  373,   52,
  234,  637,  734,  386,   57,
  222,  626,  739,  398,   63,
  211,  614,  743,  411,   69,
  200,  602,  746,  424,   76,
  189,  590,  749,  437,   83,
  178,  578,  751,  451,   90,
  168,  566,  753,  464,   97,
  158,  553,  755,  477,  105,
  148,  541,  756,  490,  113,
  139,  528,  757,  503,  121};
```

9.4.7.4.2.3 ppfcoef_scale_eq_4_32_phases_flip.dat

```
5,   32,   11,
  116,  515,  786,  515,  116,
  107,  502,  785,  530,  124,
   99,  488,  784,  544,  133,
   92,  473,  783,  557,  143,
   85,  459,  781,  571,  152,
   78,  445,  778,  585,  162,
   71,  431,  775,  598,  173,
   65,  417,  772,  611,  183,
   59,  403,  767,  624,  195,
   53,  389,  763,  637,  206,
   48,  375,  758,  649,  218,
   43,  362,  752,  661,  230,
   38,  348,  747,  673,  242,
   34,  334,  740,  685,  255,
```

```
   30,  321,  733,  696,  268,
   26,  308,  726,  707,  281,
  298,  726,  726,  298,    0,
  281,  707,  726,  308,   26,
  268,  696,  733,  321,   30,
  255,  685,  740,  334,   34,
  242,  673,  747,  348,   38,
  230,  661,  752,  362,   43,
  218,  649,  758,  375,   48,
  206,  637,  763,  389,   53,
  195,  624,  767,  403,   59,
  183,  611,  772,  417,   65,
  173,  598,  775,  431,   71,
  162,  585,  778,  445,   78,
  152,  571,  781,  459,   85,
  143,  557,  783,  473,   92,
  133,  544,  784,  488,   99,
  124,  530,  785,  502,  107};
```

9.4.7.4.2.4 ppfcoef_scale_eq_5_32_phases_flip.dat

```
5,   32,   11,
   98,  515,  822,  515,   98,
   90,  500,  821,  531,  106,
   83,  484,  820,  547,  114,
   75,  469,  819,  562,  123,
   69,  453,  816,  577,  133,
   63,  438,  813,  592,  142,
   57,  422,  809,  607,  153,
   51,  407,  805,  622,  163,
   46,  391,  801,  636,  174,
   41,  376,  795,  650,  186,
   37,  361,  789,  664,  197,
   32,  347,  782,  678,  209,
   28,  332,  775,  691,  222,
   25,  317,  767,  704,  235,
   22,  303,  759,  716,  248,
   18,  289,  750,  729,  262,
  278,  746,  746,  278,    0,
  262,  729,  750,  289,   18,
  248,  716,  759,  303,   22,
  235,  704,  767,  317,   25,
  222,  691,  775,  332,   28,
  209,  678,  782,  347,   32,
  197,  664,  789,  361,   37,
  186,  650,  795,  376,   41,
  174,  636,  801,  391,   46,
  163,  622,  805,  407,   51,
  153,  607,  809,  422,   57,
  142,  592,  813,  438,   63,
  133,  577,  816,  453,   69,
  123,  562,  819,  469,   75,
  114,  547,  820,  484,   83,
  106,  531,  821,  500,   90};
```

9.4.7.4.2.5 ppfcoef_scale_eq_6_32_phases_flip.dat

```
5,   32,   11,
   77,  513,  868,  513,   77,
   70,  496,  867,  531,   84,
   63,  479,  866,  548,   92,
   57,  461,  864,  566,  100,
   51,  444,  861,  583,  109,
   46,  427,  857,  600,  118,
   41,  409,  853,  617,  128,
   36,  393,  847,  633,  139,
   32,  376,  841,  650,  149,
   28,  359,  835,  666,  160,
   24,  343,  827,  682,  172,
   21,  327,  819,  697,  184,
   18,  311,  810,  712,  197,
   15,  296,  800,  727,  210,
   13,  281,  790,  741,  223,
```

```
   11,  266,  779,  755,  237,
  253,  771,  771,  253,    0,
  237,  755,  779,  266,   11,
  223,  741,  790,  281,   13,
  210,  727,  800,  296,   15,
  197,  712,  810,  311,   18,
  184,  697,  819,  327,   21,
  172,  682,  827,  343,   24,
  160,  666,  835,  359,   28,
  149,  650,  841,  376,   32,
  139,  633,  847,  393,   36,
  128,  617,  853,  409,   41,
  118,  600,  857,  427,   46,
  109,  583,  861,  444,   51,
  100,  566,  864,  461,   57,
   92,  548,  866,  479,   63,
   84,  531,  867,  496,   70};
```

9.4.7.4.2.6 ppfcoef_scale_eq_7_32_phases_flip.dat

```
5,   32,   11,
   53,  510,  922,  510,   53,
   47,  490,  922,  529,   60,
   41,  470,  921,  549,   67,
   36,  451,  918,  569,   74,
   32,  431,  915,  588,   82,
   27,  412,  910,  608,   91,
   23,  393,  905,  627,  100,
   20,  374,  898,  646,  110,
   17,  356,  890,  665,  120,
   14,  337,  882,  684,  131,
   11,  320,  873,  702,  142,
    9,  302,  863,  720,  154,
    7,  285,  852,  737,  167,
    6,  269,  840,  753,  180,
    4,  253,  827,  770,  194,
    3,  237,  815,  785,  208,
  223,  801,  801,  223,    0,
  208,  785,  815,  237,    3,
  194,  770,  827,  253,    4,
  180,  753,  840,  269,    6,
  167,  737,  852,  285,    7,
  154,  720,  863,  302,    9,
  142,  702,  873,  320,   11,
  131,  684,  882,  337,   14,
  120,  665,  890,  356,   17,
  110,  646,  898,  374,   20,
  100,  627,  905,  393,   23,
   91,  608,  910,  412,   27,
   82,  588,  915,  431,   32,
   74,  569,  918,  451,   36,
   67,  549,  921,  470,   41,
   60,  529,  922,  490,   47};
```

9.4.7.4.2.6.1 ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat

```
5   32  11
   28   502   988   502    28
   24   479   987   524    34
   19   457   985   547    40
   15   435   982   570    46
   12   413   978   592    53
    9   392   972   614    61
    6   371   965   637    69
    4   350   957   659    78
    2   330   948   680    88
    0   310   938   702    98
   -1   291   926   723   109
   -2   272   914   744   120
   -3   254   900   764   133
   -3   237   886   783   145
   -4   220   871   802   159
   -4   204   855   820   173
```

```
  188  836  836  188    0
  173  820  855  204   -4
  159  802  871  220   -4
  145  783  886  237   -3
  133  764  900  254   -3
  120  744  914  272   -2
  109  723  926  291   -1
   98  702  938  310    0
   88  680  948  330    2
   78  659  957  350    4
   69  637  965  371    6
   61  614  972  392    9
   53  592  978  413   12
   46  570  982  435   15
   40  547  985  457   19
   34  524  987  479   24
```

9.4.7.4.2.6.2 ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat

```
5   32  11
    3  489 1064  489    3
    0  464 1062  515    7
   -3  439 1060  540   12
   -5  414 1056  566   17
   -7  390 1050  592   23
   -9  366 1044  618   29
  -10  343 1035  644   36
  -11  320 1025  670   44
  -12  298 1014  695   53
  -12  277 1001  720   62
  -12  256  987  745   72
  -12  236  972  769   83
  -12  217  956  792   95
  -11  199  938  815  107
  -10  181  920  837  120
  -10  165  900  859  134
  148  876  876  148    0
  134  859  900  165  -10
  120  837  920  181  -10
  107  815  938  199  -11
   95  792  956  217  -12
   83  769  972  236  -12
   72  745  987  256  -12
   62  720 1001  277  -12
   53  695 1014  298  -12
   44  670 1025  320  -11
   36  644 1035  343  -10
   29  618 1044  366   -9
   23  592 1050  390   -7
   17  566 1056  414   -5
   12  540 1060  439   -3
    7  515 1062  464    0
```

9.4.7.4.2.6.3 ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat

```
5   32  11
  -20  470 1148  470  -20
  -22  442 1147  499  -18
  -23  413 1144  529  -15
  -24  386 1139  558  -11
  -24  359 1132  588   -7
  -24  333 1124  618   -3
  -24  308 1113  648    3
  -23  283 1101  678    9
  -23  260 1088  707   16
  -22  237 1072  737   24
  -21  215 1056  765   33
  -19  194 1037  793   43
  -18  174 1017  822   53
  -16  156  995  848   65
  -15  138  973  875   77
  -13  121  949  900   91
  105  919  919  105    0
```

```
    91   900   949  121   -13
    77   875   973  138   -15
    65   848   995  156   -16
    53   822  1017  174   -18
    43   793  1037  194   -19
    33   765  1056  215   -21
    24   737  1072  237   -22
    16   707  1088  260   -23
     9   678  1101  283   -23
     3   648  1113  308   -24
    -3   618  1124  333   -24
    -7   588  1132  359   -24
   -11   558  1139  386   -24
   -15   529  1144  413   -23
   -18   499  1147  442   -22
```

9.4.7.4.2.6.4 ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat

```
5   32  11
   -40   444  1240  444   -40
   -40   412  1240  476   -40
   -40   381  1236  510   -39
   -39   350  1231  544   -38
   -37   321  1223  577   -36
   -36   293  1212  612   -33
   -34   265  1200  646   -29
   -32   239  1185  681   -25
   -30   214  1169  715   -20
   -28   190  1150  750   -14
   -26   167  1130  783    -6
   -23   146  1107  816     2
   -21   126  1083  849    11
   -19   107  1057  882    21
   -17    90  1030  913    32
   -15    73  1002  943    45
    58   966   966   58     0
    45   943  1002   73   -15
    32   913  1030   90   -17
    21   882  1057  107   -19
    11   849  1083  126   -21
     2   816  1107  146   -23
    -6   783  1130  167   -26
   -14   750  1150  190   -28
   -20   715  1169  214   -30
   -25   681  1185  239   -32
   -29   646  1200  265   -34
   -33   612  1212  293   -36
   -36   577  1223  321   -37
   -38   544  1231  350   -39
   -39   510  1236  381   -40
   -40   476  1240  412   -40
```

9.4.7.4.2.6.5 ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat

```
5   32  11
   -56   409  1342  409   -56
   -54   373  1342  445   -58
   -51   339  1337  482   -59
   -49   306  1330  521   -60
   -46   274  1321  559   -60
   -43   244  1308  598   -59
   -40   215  1293  638   -58
   -36   187  1275  678   -56
   -33   161  1255  718   -53
   -30   137  1233  757   -49
   -27   114  1208  797   -44
   -24    93  1182  836   -39
   -21    73  1152  875   -31
   -18    55  1122  912   -23
   -16    38  1090  950   -14
   -14    23  1056  986    -3
     9  1015  1015    9     0
    -3   986  1056   23   -14
```

```
   -14  950 1090   38  -16
   -23  912 1122   55  -18
   -31  875 1152   73  -21
   -39  836 1182   93  -24
   -44  797 1208  114  -27
   -49  757 1233  137  -30
   -53  718 1255  161  -33
   -56  678 1275  187  -36
   -58  638 1293  215  -40
   -59  598 1308  244  -43
   -60  559 1321  274  -46
   -60  521 1330  306  -49
   -59  482 1337  339  -51
   -58  445 1342  373  -54
```

9.4.7.4.2.6.6 ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat

```
5  32  11
   -65  364 1450  364  -65
   -61  326 1448  404  -69
   -57  289 1443  445  -72
   -53  253 1435  488  -75
   -48  220 1423  531  -78
   -44  188 1408  576  -80
   -40  158 1390  621  -81
   -36  130 1370  666  -82
   -32  103 1346  713  -82
   -28   79 1320  758  -81
   -24   56 1290  805  -79
   -21   36 1259  850  -76
   -18   17 1224  896  -71
   -15    0 1188  940  -65
   -12  -15 1149  984  -58
   -10  -28 1109 1027  -50
   -40 1064 1064  -40    0
   -50 1027 1109  -28  -10
   -58  984 1149  -15  -12
   -65  940 1188    0  -15
   -71  896 1224   17  -18
   -76  850 1259   36  -21
   -79  805 1290   56  -24
   -81  758 1320   79  -28
   -82  713 1346  103  -32
   -82  666 1370  130  -36
   -81  621 1390  158  -40
   -80  576 1408  188  -44
   -78  531 1423  220  -48
   -75  488 1435  253  -53
   -72  445 1443  289  -57
   -69  404 1448  326  -61
```

9.4.7.4.2.6.7 ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat

```
5  32  11
   -67  310 1562  310  -67
   -61  269 1559  353  -72
   -55  230 1553  398  -78
   -50  193 1543  445  -83
   -44  158 1529  493  -88
   -39  125 1512  543  -93
   -34   94 1491  594  -97
   -30   66 1468  645 -101
   -25   41 1439  697 -104
   -22   17 1408  751 -106
   -18   -4 1373  804 -107
   -15  -23 1336  857 -107
   -12  -40 1296  910 -106
    -9  -55 1253  962 -103
    -7  -67 1208 1013  -99
    -5  -78 1161 1064  -94
   -86 1110 1110  -86    0
   -94 1064 1161  -78   -5
   -99 1013 1208  -67   -7
```

```
  -103  962 1253  -55   -9
  -106  910 1296  -40  -12
  -107  857 1336  -23  -15
  -107  804 1373   -4  -18
  -106  751 1408   17  -22
  -104  697 1439   41  -25
  -101  645 1468   66  -30
   -97  594 1491   94  -34
   -93  543 1512  125  -39
   -88  493 1529  158  -44
   -83  445 1543  193  -50
   -78  398 1553  230  -55
   -72  353 1559  269  -61
```

9.4.7.4.2.6.8 ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat

```
5  32 11
   -61  248 1674  248  -61
   -54  204 1673  293  -68
   -47  163 1665  342  -75
   -41  125 1654  392  -82
   -35   90 1638  445  -90
   -29   57 1618  499  -97
   -24   27 1593  556 -104
   -20    0 1565  613 -110
   -16  -24 1532  672 -116
   -12  -46 1495  732 -121
    -9  -65 1455  793 -126
    -6  -81 1411  854 -130
    -4  -95 1364  915 -132
    -2 -107 1315  975 -133
     0 -116 1262 1035 -133
     1 -123 1208 1094 -132
  -128 1152 1152 -128    0
  -132 1094 1208 -123    1
  -133 1035 1262 -116    0
  -133  975 1315 -107   -2
  -132  915 1364  -95   -4
  -130  854 1411  -81   -6
  -126  793 1455  -65   -9
  -121  732 1495  -46  -12
  -116  672 1532  -24  -16
  -110  613 1565    0  -20
  -104  556 1593   27  -24
   -97  499 1618   57  -29
   -90  445 1638   90  -35
   -82  392 1654  125  -41
   -75  342 1665  163  -47
   -68  293 1673  204  -54
```

9.4.7.4.3 VS (Bilinear Filter Coefficients)

9.4.7.4.3.1 ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat

This is not applicable for this device

```
7  32 11
   -11 -106  190 1902  190 -106  -11
    -9 -105  153 1897  230 -105  -13
    -8 -105  121 1887  274 -105  -16
    -6 -104   91 1869  320 -104  -18
    -5 -102   65 1843  370 -102  -21
    -4 -101   42 1812  424 -101  -24
    -3  -99   20 1776  480  -99  -27
    -2  -96    3 1730  539  -96  -30
    -1  -93  -12 1679  602  -93  -34
    -1  -90  -26 1627  665  -90  -37
     0  -87  -37 1568  732  -87  -41
     0  -84  -46 1506  801  -84  -45
     0  -80  -54 1439  871  -80  -48
     0  -76  -60 1371  941  -76  -52
     1  -72  -65 1299 1013  -72  -56
     1  -68  -69 1227 1085  -68  -60
   -64  -64 1152 1152  -64  -64    0
```

```
  -60  -68 1085 1227  -69  -68    1
  -56  -72 1013 1299  -65  -72    1
  -52  -76  941 1371  -60  -76    0
  -48  -80  871 1439  -54  -80    0
  -45  -84  801 1506  -46  -84    0
  -41  -87  732 1568  -37  -87    0
  -37  -90  665 1627  -26  -90   -1
  -34  -93  602 1679  -12  -93   -1
  -30  -96  539 1730    3  -96   -2
  -27  -99  480 1776   20  -99   -3
  -24 -101  424 1812   42 -101   -4
  -21 -102  370 1843   65 -102   -5
  -18 -104  320 1869   91 -104   -6
  -16 -105  274 1887  121 -105   -8
  -13 -105  230 1897  153 -105   -9
```

9.4.8 VIP Video Port Direct Memory Access (VPDMA)

9.4.8.1 VPDMA Introduction

The VPDMA primary function is to move data between external memory and internal processing modules that source or sink data. VPDMA is capable buffering this data and then delivering the data as demanded to the modules as programmed. The modules that source or sink data are referred to as clients. A channel is setup inside the VPDMA to connect a specific memory buffer to a specific client. The VPDMA centralizes the DMA control functions and buffering required to allow all the clients to minimize the effect of long latency times. The VPDMA also supports a descriptor based mode where lists of descriptors can be setup to configure all the channels as they become available.

Additionally, in a third-party configuration, the VPDMA is capable of performing DMA transfers as requested by the pulsing of an event strobe. The VPDMA is capable of generation of an address which may reach any location in the range for which it is configured. It is capable of moving this data either to or from a Shared Buffer and ultimately to or from a Client Buffer. For the two type of Client Buffers that are used to drive data into a subsystem (Streaming Buffer and Random Access Buffer) data is either pushed into the buffer or pulled out of the buffer dependent upon the direction of the data transfer. For third-party DMA operations the data is transferred into a Client buffer, and then transferred out of the Client Buffer to the transfer destination. These transfers are triggered by an Event pulse to each of the Client Buffers which are Routing Buffer types.

9.4.8.2 VPDMA Basic Definitions

9.4.8.2.1 Client

The modules that source or sink data are referred to as clients. The clients of the VPDMA are the physical between the processing modules (VIP) and external memory. A channel is the mechanism inside the VPDMA that connects a specific memory buffer or transfer to a specific client.

The start event can also be selected by a channel attribute or to be controlled by an internal frame signal controlled by the List Manager.

9.4.8.2.2 Channel

The VPDMA requires a channel to be setup for each group of transfers. All the channels are described through a Data Transfer Descriptor that has a common format. The client that the channel is mapped to interprets the information in the descriptor to perform the requested data transfer.

Each of the channels has a type of data that it can support based upon the client that it services. The VPDMA supports three types of channels:

- **YUV Channel** - Clients taking data YUV data
- **RGB Channel** - Clients taking RGB data
- **Miscellaneous Channel** - The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client.

9.4.8.2.3 List

A list is a group of descriptors that makes up a set of DMA transfers that need to be completed. The VPDMA supports one kind of list only:

- The **Regular List** is a single list that the VPDMA will execute each descriptor once and initiate an interrupt when the list has completed. A regular list can contain any kind of descriptor without limitation and be of any size.

The VPDMA Controller works on lists of descriptors. In this mode the processor writes the lists of descriptors in the order it wants them executed. It then writes the location of the list to the VIP_LIST_ADDR register, followed by writing the size (bit LIST_SIZE) and type (bit LIST_TYPE) of the list, and list number (bit LIST_NUM) to the VIP_LIST_ATTR register. The List Manager module then schedules a DMA transfer to pull in the portion of the list that it can store in internal VPDMA memory. The List Manager will sequentially process the active list of descriptors until either the descriptor requires the use of a client that is currently active, or if it is waiting for the next portion of the list to be transferred from a DMA request. If there is no active list to process the list manager will go into an IDLE mode waiting for any client that is blocking a list or a list DMA transfer to complete.

All the DMA transfers are controlled by List Manager module inside VPDMA. List Manager needs to be loaded with FIRMWARE, before any DMA transfer from memory, after the VPDMA reset. The first MMR write to VIP_LIST_ADDR register after VPDMA reset should be the address of the memory buffer(128-bit aligned, that is, last four bits of the buffer address should be zero) where the firmware is stored. List Manager then schedules a DMA transaction to fetch the firmware and sets the VIP_LIST_ATTR[19] RDY bit after the firmware loading is complete.

9.4.8.2.4 Data Formats Supported

Following list summarizes the data formats supported in the VPDMA. For more inforamtion see Section 9.4.8.10, *VPDMA Data Formats*.

- RGB Data Types:
  - RGB16-565
  - ARGB-1555
  - ARGB-4444
  - RGBA-5551
  - RGBA-4444
  - ARGB24-6666
  - RGB24-888
  - ARGB32-8888
  - RGBA24-6666
  - RGBA32-8888
  - BGR16-565
  - ABGR-1555
  - ABGR-4444
  - BGRA-5551
  - BGRA-4444
  - ABGR24-6666
  - BGR24-888
  - ABGR32-8888
  - BGRA24-6666
  - BGRA32-8888
- YUV Data Types:
  - Y 4:4:4
  - Y 4:2:2
  - Y 4:2:0
  - C 4:4:4
  - C 4:2:2
  - C 4:2:0

– CY 4:2:2
– YCbC 4:4:4
– YC 4:2:2

Note

VPDMA supports swapping formats (RGB/BGR and Cb/Cr)

9.4.8.3

Note

VPDMA supports swapping formats (RGB/BGR and Cb/Cr)

9.4.8.4 VPDMA Client Buffering and Functionality

Table 9-32 lists for each client:

- The channels used, amount of buffering allocated for it, and the shared buffer used for its memory
- The line sizes it handles for tiled and non-tiled memory spaces, as well as any additional features it supports

Table 9-32. VPDMA Client Buffering and Functionality

Client	Channel(s)	Tiled Memory Max Line Size	Non-Tiled Memory Max Line Size	Additional Features
vip1_lo_y	vip1_mult_porta_src0 vip1_mult_porta_src1 vip1_mult_porta_src2 vip1_mult_porta_src3 vip1_mult_porta_src4 vip1_mult_porta_src5 vip1_mult_porta_src6 vip1_mult_porta_src7 vip1_mult_porta_src8 vip1_mult_porta_src9 vip1_mult_porta_src10 vip1_mult_porta_src11 vip1_mult_porta_src12 vip1_mult_porta_src13 vip1_mult_porta_src14 vip1_mult_porta_src15 vip1_portb_luma vip1_portb_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip1_lo_uv	vip1_mult_portb_src0 vip1_mult_portb_src1 vip1_mult_portb_src2 vip1_mult_portb_src3 vip1_mult_portb_src4 vip1_mult_portb_src5 vip1_mult_portb_src6 vip1_mult_portb_src7 vip1_mult_portb_src8 vip1_mult_portb_src9 vip1_mult_portb_src10 vip1_mult_portb_src11 vip1_mult_portb_src12 vip1_mult_portb_src13 vip1_mult_portb_src14 vip1_mult_portb_src15 vip1_portb_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED
vip1_up_y	vip1_porta_luma vip1_porta_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip1_up_uv	vip1_porta_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED

Table 9-32. VPDMA Client Buffering and Functionality (continued)

vip2_lo_y	vip2_mult_porta_src0 vip2_mult_porta_src1 vip2_mult_porta_src2 vip2_mult_porta_src3 vip2_mult_porta_src4 vip2_mult_porta_src5 vip2_mult_porta_src6 vip2_mult_porta_src7 vip2_mult_porta_src8 vip2_mult_porta_src9 vip2_mult_porta_src10 vip2_mult_porta_src11 vip2_mult_porta_src12 vip2_mult_porta_src13 vip2_mult_porta_src14 vip2_mult_porta_src15 vip2_portb_luma vip2_portb_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip2_lo_uv	vip2_mult_portb_src0 vip2_mult_portb_src1 vip2_mult_portb_src2 vip2_mult_portb_src3 vip2_mult_portb_src4 vip2_mult_portb_src5 vip2_mult_portb_src6 vip2_mult_portb_src7 vip2_mult_portb_src8 vip2_mult_portb_src9 vip2_mult_portb_src10 vip2_mult_portb_src11 vip2_mult_portb_src12 vip2_mult_portb_src13 vip2_mult_portb_src14 vip2_mult_portb_src15 vip2_portb_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED
vip2_up_y	vip2_porta_luma vip2_porta_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip2_up_uv	vip2_porta_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED
vpi_ctl		Tiled Data Not Supported	4096	
vip1_anc_a	vip1_mult_anca_src0 vip1_mult_anca_src1 vip1_mult_anca_src2 vip1_mult_anca_src3 vip1_mult_anca_src4 vip1_mult_anca_src5 vip1_mult_anca_src6 vip1_mult_anca_src7 vip1_mult_anca_src8 vip1_mult_anca_src9 vip1_mult_anca_src10 vip1_mult_anca_src11 vip1_mult_anca_src12 vip1_mult_anca_src13 vip1_mult_anca_src14 vip1_mult_anca_src15	Tiled Data Not Supported	4096	
vip1_anc_b	vip1_mult_ancb_src0 vip1_mult_ancb_src1 vip1_mult_ancb_src2 vip1_mult_ancb_src3 vip1_mult_ancb_src4 vip1_mult_ancb_src5 vip1_mult_ancb_src6 vip1_mult_ancb_src7 vip1_mult_ancb_src8 vip1_mult_ancb_src9 vip1_mult_ancb_src10 vip1_mult_ancb_src11 vip1_mult_ancb_src12 vip1_mult_ancb_src13 vip1_mult_ancb_src14 vip1_mult_ancb_src15	Tiled Data Not Supported	4096	

Table 9-32. VPDMA Client Buffering and Functionality (continued)

vip2_anc_a	vip2_mult_anca_src0 vip2_mult_anca_src1 vip2_mult_anca_src2 vip2_mult_anca_src3 vip2_mult_anca_src4 vip2_mult_anca_src5 vip2_mult_anca_src6 vip2_mult_anca_src7 vip2_mult_anca_src8 vip2_mult_anca_src9 vip2_mult_anca_src10 vip2_mult_anca_src11 vip2_mult_anca_src12 vip2_mult_anca_src13 vip2_mult_anca_src14 vip2_mult_anca_src15	Tiled Data Not Supported	4096	

9.4.8.5 VPDMA Channels Assignment

Table 9-33 lists all of the channels in VPDMA and its base attributes. The Data Type column states what type of data YUV, RGB or OTHER the channel handles and in parentheses are the legal data type values that can be entered into a data transfer decriptor. The Client field states the name of the Client and in parentheses it states the reference number in Figure 9-4, *VIP Block Diagram*.

Table 9-33. VPDMA Channels Assignment

Channel	Description	Channel Number	Data Type	Client
vip1_mult_porta_src0	Video Input 1 Port A Channel 0	38	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src1	Video Input 1 Port A Channel 1	39	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src2	Video Input 1 Port A Channel 2	40	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src3	Video Input 1 Port A Channel 3	41	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src4	Video Input 1 Port A Channel 4	42	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src5	Video Input 1 Port A Channel 5	43	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src6	Video Input 1 Port A Channel 6	44	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src7	Video Input 1 Port A Channel 7	45	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src8	Video Input 1 Port A Channel 8	46	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src9	Video Input 1 Port A Channel 9	47	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src10	Video Input 1 Port A Channel 10	48	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src11	Video Input 1 Port A Channel 11	49	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src12	Video Input 1 Port A Channel 12	50	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src13	Video Input 1 Port A Channel 13	51	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src14	Video Input 1 Port A Channel 14	52	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src15	Video Input 1 Port A Channel 15	53	YUV (0x7)	vip1_lo_y (2)
vip1_mult_portb_src0	Video Input 1 Port B Channel 0	54	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src1	Video Input 1 Port B Channel 1	55	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src2	Video Input 1 Port B Channel 2	56	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src3	Video Input 1 Port B Channel 3	57	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src4	Video Input 1 Port B Channel 4	58	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src5	Video Input 1 Port B Channel 5	59	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src6	Video Input 1 Port B Channel 6	60	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src7	Video Input 1 Port B Channel 7	61	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src8	Video Input 1 Port B Channel 8	62	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src9	Video Input 1 Port B Channel 9	63	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src10	Video Input 1 Port B Channel 10	64	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src11	Video Input 1 Port B Channel 11	65	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src12	Video Input 1 Port B Channel 12	66	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src13	Video Input 1 Port B Channel 13	67	YUV (0x7)	vip1_lo_uv (1)

Table 9-33. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip1_mult_portb_src14	Video Input 1 Port B Channel 14	68	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src15	Video Input 1 Port B Channel 15	69	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_anca_src0	Video Input 1 Port A Ancillary Data Channel 0	70	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src1	Video Input 1 Port A Ancillary Data Channel 1	71	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src2	Video Input 1 Port A Ancillary Data Channel 2	72	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src3	Video Input 1 Port A Ancillary Data Channel 3	73	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src4	Video Input 1 Port A Ancillary Data Channel 4	74	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src5	Video Input 1 Port A Ancillary Data Channel 5	75	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src6	Video Input 1 Port A Ancillary Data Channel 6	76	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src7	Video Input 1 Port A Ancillary Data Channel 7	77	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src8	Video Input 1 Port A Ancillary Data Channel 8	78	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src9	Video Input 1 Port A Ancillary Data Channel 9	79	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src10	Video Input 1 Port A Ancillary Data Channel 10	80	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src11	Video Input 1 Port A Ancillary Data Channel 11	81	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src12	Video Input 1 Port A Ancillary Data Channel 12	82	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src13	Video Input 1 Port A Ancillary Data Channel 13	83	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src14	Video Input 1 Port A Ancillary Data Channel 14	84	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src15	Video Input 1 Port A Ancillary Data Channel 15	85	OTHER (8)	vip1_anc_a(2)
vip1_mult_ancb_src0	Video Input 1 Port B Ancillary Data Channel 0	86	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src1	Video Input 1 Port B Ancillary Data Channel 1	87	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src2	Video Input 1 Port B Ancillary Data Channel 2	88	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src3	Video Input 1 Port B Ancillary Data Channel 3	89	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src4	Video Input 1 Port B Ancillary Data Channel 4	90	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src5	Video Input 1 Port B Ancillary Data Channel 5	91	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src6	Video Input 1 Port B Ancillary Data Channel 6	92	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src7	Video Input 1 Port B Ancillary Data Channel 7	93	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src8	Video Input 1 Port B Ancillary Data Channel 8	94	OTHER (8)	vip1_anc_b(2)

Table 9-33. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip1_mult_ancb_src9	Video Input 1 Port B Ancillary Data Channel 9	95	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src10	Video Input 1 Port B Ancillary Data Channel 10	96	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src11	Video Input 1 Port B Ancillary Data Channel 11	97	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src12	Video Input 1 Port B Ancillary Data Channel 12	98	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src13	Video Input 1 Port B Ancillary Data Channel 13	99	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src14	Video Input 1 Port B Ancillary Data Channel 14	100	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src15	Video Input 1 Port B Ancillary Data Channel 15	101	OTHER (8)	vip1_anc_b(2)
vip1_porta_luma	Video Input 1 Port A 420 Data Luma	102	YUV (0x1, 0x2, 0x7)	vip1_up_y (1)
vip1_porta_chroma	Video Input 1 Port A 420 Data Chroma	103	YUV (0x5, 0x6, 0x7)	vip1_up_uv(1)
vip1_portb_luma	Video Input 1 Port B 420 Data Luma	104	YUV (0x1, 0x2, 0x7)	vip1_lo_y (2)
vip1_portb_chroma	Video Input 1 Port B 420 Data Chroma	105	YUV (0x5, 0x6, 0x7)	vip1_lo_uv (2)
vip1_porta_rgb	Video Input 1 Port A RGB Data	106	RGB (0x0 - 0x8)	vip1_up_y (1)
vip1_portb_rgb	Video Input 1 Port B RGB Data	107	RGB (0x0 - 0x8)	vip1_lo_y (2)
vip2_mult_porta_src0	Video Input 2 Port A Channel 0	108	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src1	Video Input 2 Port A Channel 1	109	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src2	Video Input 2 Port A Channel 2	110	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src3	Video Input 2 Port A Channel 3	111	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src4	Video Input 2 Port A Channel 4	112	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src5	Video Input 2 Port A Channel 5	113	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src6	Video Input 2 Port A Channel 6	114	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src7	Video Input 2 Port A Channel 7	115	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src8	Video Input 2 Port A Channel 8	116	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src9	Video Input 2 Port A Channel 9	117	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src10	Video Input 2 Port A Channel 10	118	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src11	Video Input 2 Port A Channel 11	119	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src12	Video Input 2 Port A Channel 12	120	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src13	Video Input 2 Port A Channel 13	121	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src14	Video Input 2 Port A Channel 14	122	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src15	Video Input 2 Port A Channel 15	123	YUV (0x7)	vip2_lo_y (21)
vip2_mult_portb_src0	Video Input 2 Port B Channel 0	124	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src1	Video Input 2 Port B Channel 1	125	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src2	Video Input 2 Port B Channel 2	126	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src3	Video Input 2 Port B Channel 3	127	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src4	Video Input 2 Port B Channel 4	128	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src5	Video Input 2 Port B Channel 5	129	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src6	Video Input 2 Port B Channel 6	130	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src7	Video Input 2 Port B Channel 7	131	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src8	Video Input 2 Port B Channel 8	132	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src9	Video Input 2 Port B Channel 9	133	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src10	Video Input 2 Port B Channel 10	134	YUV (0x7)	vip2_lo_uv (20)

Table 9-33. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip2_mult_portb_src11	Video Input 2 Port B Channel 11	135	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src12	Video Input 2 Port B Channel 12	136	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src13	Video Input 2 Port B Channel 13	137	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src14	Video Input 2 Port B Channel 14	138	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src15	Video Input 2 Port B Channel 15	139	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_anca_src0	Video Input 2 Port A Ancillary Data Channel 0	140	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src1	Video Input 2 Port A Ancillary Data Channel 1	141	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src2	Video Input 2 Port A Ancillary Data Channel 2	142	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src3	Video Input 2 Port A Ancillary Data Channel 3	143	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src4	Video Input 2 Port A Ancillary Data Channel 4	144	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src5	Video Input 2 Port A Ancillary Data Channel 5	145	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src6	Video Input 2 Port A Ancillary Data Channel 6	146	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src7	Video Input 2 Port A Ancillary Data Channel 7	147	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src8	Video Input 2 Port A Ancillary Data Channel 8	148	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src9	Video Input 2 Port A Ancillary Data Channel 9	149	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src10	Video Input 2 Port A Ancillary Data Channel 10	150	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src11	Video Input 2 Port A Ancillary Data Channel 11	151	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src12	Video Input 2 Port A Ancillary Data Channel 12	152	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src13	Video Input 2 Port A Ancillary Data Channel 13	153	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src14	Video Input 2 Port A Ancillary Data Channel 14	154	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src15	Video Input 2 Port A Ancillary Data Channel 15	155	OTHER (8)	vip2_anc_a(21)
vip2_mult_ancb_src0	Video Input 2 Port B Ancillary Data Channel 0	156	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src1	Video Input 2 Port B Ancillary Data Channel 1	157	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src2	Video Input 2 Port B Ancillary Data Channel 2	158	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src3	Video Input 2 Port B Ancillary Data Channel 3	159	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src4	Video Input 2 Port B Ancillary Data Channel 4	160	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src5	Video Input 2 Port B Ancillary Data Channel 5	161	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src6	Video Input 2 Port B Ancillary Data Channel 6	162	OTHER (8)	vip2_anc_b(21)

Table 9-33. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip2_mult_ancb_src7	Video Input 2 Port B Ancillary Data Channel 7	163	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src8	Video Input 2 Port B Ancillary Data Channel 8	164	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src9	Video Input 2 Port B Ancillary Data Channel 9	165	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src10	Video Input 2 Port B Ancillary Data Channel 10	166	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src11	Video Input 2 Port B Ancillary Data Channel 11	167	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src12	Video Input 2 Port B Ancillary Data Channel 12	168	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src13	Video Input 2 Port B Ancillary Data Channel 13	169	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src14	Video Input 2 Port B Ancillary Data Channel 14	170	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src15	Video Input 2 Port B Ancillary Data Channel 15	171	OTHER (8)	vip2_anc_b(21)
vip2_porta_luma	Video Input 2 Port A 420 Data Luma	172	YUV (0x1, 0x2, 0x7)	vip2_up_y (20)
vip2_porta_chroma	Video Input 2 Port A 420 Data Chroma	173	YUV (0x5, 0x6, 0x7)	vip2_up_uv(20)
vip2_portb_luma	Video Input 2 Port B 420 Data Luma	174	YUV (0x1, 0x2, 0x7)	vip2_lo_y (21)
vip2_portb_chroma	Video Input 2 Port B 420 Data Chroma	175	YUV (0x5, 0x6, 0x7)	vip2_lo_uv (21)
vip2_porta_rgb	Video Input 2 Port A RGB Data	176	RGB (0x0 - 0x8)	vip2_up_y(20)
vip2_portb_rgb	Video Input 2 Port B RGB Data	177	RGB (0x0 - 0x8)	vip2_lo_y (21)

9.4.8.6 VPDMA MFLAG Mechanism

The device L3_MAIN interconnect accepts MFLAG signals from certain initiators that can influence the internal L3_MAIN arbitration mechanisms. As a result, a higher priority is given to the data traffic initiated by these initiators. The VIP VPDMA can directly drive such MFLAG signals dynamically. The MFLAG generation for VIP VPDMA is enabled by default, and there is no register control over it.

The VPDMA arbitrates between multiple DMA sources within the VIP based on FIFO levels of DMA channels connected to VPDMA. Priority escalation mechanism implemented within VIP subsystem is based on overflow threshold and FIFO margin.

The following is a summary of priority and MFLAG levels provided by the VIP:

- High priority (MFLAG = 3) when FIFO margin is below 25%
- Medium priority (MFLAG = 1) when FIFO margin is between 25% and 50%
- Low priority (MFLAG = 0) when FIFO margin is above 50%

Additionally, the VIP subsystem also generates MReqPriority based upon a programmed descriptor configuration. The MReqPriority configuration influences the arbitration mechanism in the Memory Subsystem only and has no influence on the arbitration that takes place within L3_MAIN interconnect. For more information see Section 9.4.8.8.1.4, *Data Packet Descriptor Word 3*.

9.4.8.7 VPDMA Interrupts

The VPDMA has 4 interrupt group(s). Each group has an interrupt for all the client interrupts, an interrupt for every 32 channels, a interrupt for each list complete, an interrupt for each list notify and an interrupt for for all of the descriptor interrupts. Each of these groups can be individually masked so that only the interrupts specified will trigger the higher level interrupt.

Each interrupt source can be individually masked independently for each seperate interrupt group. A status register bit exists for each interrupt source for for each interrupt group, that is set whenever the interrupt event occurs even when if the interrupt is masked. The status register bit will remain set until cleared by software by writing a one to the status bit.

Table 9-34 shows all interrupt events from VPDMA that go to VIP top level. The interrupt events are mapped to two interrupt lines, INT0 and INT1, that go to VIP top level.

Table 9-34. VPDMA Interrupt Events

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_channel_group0	VIP_INT0_CHANNEL0_INT_STAT VIP_INT1_CHANNEL0_INT_STAT	VIP_INT0_CHANNEL0_INT_MASK VIP_INT1_CHANNEL0_INT_MASK	An unmasked channel interrupt for interrupt group 0 in channel register 0 or 1 has fired.
vpdma_int_channel_group1	VIP_INT0_CHANNEL1_INT_STAT VIP_INT1_CHANNEL1_INT_STAT	VIP_INT0_CHANNEL1_INT_MASK VIP_INT1_CHANNEL1_INT_MASK	An unmasked channel interrupt for interrupt group 1 in channel register 0 or 1 has fired.
vpdma_int_channel_group2	VIP_INT0_CHANNEL2_INT_STAT VIP_INT1_CHANNEL2_INT_STAT	VIP_INT0_CHANNEL2_INT_MASK VIP_INT1_CHANNEL2_INT_MASK	An unmasked channel interrupt for interrupt group 2 in channel register 0 or 1 has fired.
vpdma_int_channel_group3	VIP_INT0_CHANNEL3_INT_STAT VIP_INT1_CHANNEL3_INT_STAT	VIP_INT0_CHANNEL3_INT_MASK VIP_INT1_CHANNEL3_INT_MASK	An unmasked channel interrupt for interrupt group 3 in channel register 0 or 1 has fired.
vpdma_int_channel_group4	VIP_INT0_CHANNEL4_INT_STAT VIP_INT1_CHANNEL4_INT_STAT	VIP_INT0_CHANNEL4_INT_MASK VIP_INT1_CHANNEL4_INT_MASK	An unmasked channel interrupt for interrupt group 4 in channel register 0 or 1 has fired.
vpdma_int_channel_group5	VIP_INT0_CHANNEL5_INT_STAT VIP_INT1_CHANNEL5_INT_STAT	VIP_INT0_CHANNEL5_INT_MASK VIP_INT1_CHANNEL5_INT_MASK	An unmasked channel interrupt for interrupt group 5 in channel register 0 or 1 has fired.
vpdma_int_list0_complete			List 0 has completed
vpdma_int_list0_notify			The data transfer in list 0 with the Notify Field set in the descriptor has completed
vpdma_int_list1_complete			List 1 has completed
vpdma_int_list1_notify			The data transfer in list 1 with the Notify Field set in the descriptor has completed
vpdma_int_list2_complete			List 2 has completed
vpdma_int_list2_notify			The data transfer in list 2 with the Notify Field set in the descriptor has completed
vpdma_int_list3_complete			List 3 has completed
vpdma_int_list3_notify			The data transfer in list 3 with the Notify Field set in the descriptor has completed
vpdma_int_list4_complete	VIP_INT0_LIST0_INT_STAT VIP_INT1_LIST0_INT_STAT	VIP_INT0_LIST0_INT_MASK VIP_INT1_LIST0_INT_MASK	List 4 has completed

Table 9-34. VPDMA Interrupt Events (continued)

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_list4_notify			The data transfer in list 4 with the Notify Field set in the descriptor has completed
vpdma_int_list5_complete			List 5 has completed
vpdma_int_list5_notify			The data transfer in list 5 with the Notify Field set in the descriptor has completed
vpdma_int_list6_complete			List 6 has completed
vpdma_int_list6_notify			The data transfer in list 6 with the Notify Field set in the descriptor has completed
vpdma_int_list7_complete			List 7 has completed
vpdma_int_list7_notify			The data transfer in list 7 with the Notify Field set in the descriptor has completed
vpdma_int_client	VIP_INT0_CLIENT0_INT_STAT VIP_INT0_CLIENT1_INT_STAT VIP_INT1_CLIENT0_INT_STAT VIP_INT1_CLIENT1_INT_STAT	VIP_INT0_CLIENT0_INT_MASK VIP_INT0_CLIENT1_INT_MASK VIP_INT1_CLIENT0_INT_MASK VIP_INT1_CLIENT1_INT_MASK	Client Interrupt
vpdma_int_descriptor	VIP_INT0_LIST0_INT_STAT VIP_INT1_LIST0_INT_STAT	VIP_INT0_LIST0_INT_STAT VIP_INT1_LIST0_INT_STAT	Descriptor Interrupt

In Table 9-34 above, the “channel_group”, “client” and “descriptor” interrupts are actually a set of additional interrupts. When software receives an interrupt from a “channel_group,” “client,” or “descriptor” it must read the appropriate register within the VPDMA (refer to Table 9-35 to determine what the actual interrupt was).

Table 9-35. VIP Interrupt Sources

Interrupt	Interrupt Group	Description
channel_vip1_mult_anca_src0	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src1	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src10	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src11	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src12	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_anca_src13	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src14	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src15	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src2	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src3	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src4	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src5	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src6	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src7	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src8	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src9	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_ancb_src0	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_ancb_src1	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src10	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src11	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src12	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src13	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src14	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src15	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src2	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src3	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src4	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src5	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src6	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_ancb_src7	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src8	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src9	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_porta_src0	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src1	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src10	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src11	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src12	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src13	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src14	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src15	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src2	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_porta_src3	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src4	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src5	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src6	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src7	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src8	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src9	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_portb_src0	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src1	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src10	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src11	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src12	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_portb_src13	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src14	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src15	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src2	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src3	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src4	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src5	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src6	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src7	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src8	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src9	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_porta_chroma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_porta_luma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_rgb	channel_group3	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point.
channel_vip1_portb_chroma	channel_group3	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_portb_luma	channel_group3	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_portb_rgb	channel_group3	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip2_mult_anca_src0	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src1	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src10	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src11	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src12	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src13	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src14	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_anca_src15	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src2	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src3	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src4	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src5	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src6	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src7	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src8	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src9	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_ancb_src0	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src1	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src10	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_ancb_src11	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src12	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src13	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src14	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src15	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src2	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src3	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src4	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src5	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src6	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src7	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src8	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_ancb_src9	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_porta_src0	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src1	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src10	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src11	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src12	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src13	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src14	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src15	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src2	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src3	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src4	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_porta_src5	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src6	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src7	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src8	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src9	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_portb_src0	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src1	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src10	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src11	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src12	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src13	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src14	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_portb_src15	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src2	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src3	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src4	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src5	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src6	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src7	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src8	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src9	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_porta_chroma	channel_group5	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_porta_luma	channel_group5	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_porta_rgb	channel_group5	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_portb_chroma	channel_group5	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_portb_luma	channel_group5	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_portb_rgb	channel_group5	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
client_vip1_anc_a	client	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_anc_b	client	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_lo_uv	client	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_lo_y	client	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_up_uv	client	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_up_y	client	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_anc_a	client	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_anc_b	client	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_lo_uv	client	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
client_vip2_lo_y	client	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_up_uv	client	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_up_y	client	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vpi_ctl	client	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
control_descriptor_int0	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0.
control_descriptor_int1	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1.
control_descriptor_int10	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10.
control_descriptor_int11	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11.
control_descriptor_int12	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12.
control_descriptor_int13	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13.
control_descriptor_int14	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14.
control_descriptor_int15	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15.
control_descriptor_int2	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2.
control_descriptor_int3	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3.
control_descriptor_int4	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4.
control_descriptor_int5	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5.
control_descriptor_int6	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6.
control_descriptor_int7	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7.
control_descriptor_int8	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8.
control_descriptor_int9	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9.
list0_complete	list0_complete	List 0 has completed
list0_notify	list0_notify	The data transfer in list 0 with the Notify Field set in the descriptor has completed

Table 9-35. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
list1_complete	list1_complete	List 1 has completed
list1_notify	list1_notify	The data transfer in list 1 with the Notify Field set in the descriptor has completed
list2_complete	list2_complete	List 2 has completed
list2_notify	list2_notify	The data transfer in list 2 with the Notify Field set in the descriptor has completed
list3_complete	list3_complete	List 3 has completed
list3_notify	list3_notify	The data transfer in list 3 with the Notify Field set in the descriptor has completed
list4_complete	list4_complete	List 4 has completed
list4_notify	list4_notify	The data transfer in list 4 with the Notify Field set in the descriptor has completed
list5_complete	list5_complete	List 5 has completed
list5_notify	list5_notify	The data transfer in list 5 with the Notify Field set in the descriptor has completed
list6_complete	list6_complete	List 6 has completed
list6_notify	list6_notify	The data transfer in list 6 with the Notify Field set in the descriptor has completed
list7_complete	list7_complete	List 7 has completed
list7_notify	list7_notify	The data transfer in list 7 with the Notify Field set in the descriptor has completed

9.4.8.8 VPDMA Descriptors

The VPDMA needs to be programmed through descriptors (a pre-defined structure of eight or four 32-bit words depending on type of descriptors) other than VPDMA Memory Mapped Registers (MMR). Descriptors are of three types:

1. **Data Transfer Descriptors** - A memory structure used to describe a desired memory transaction to or from a client.
2. **Control Descriptors** - A memory structure used to perform a control operation inside the DMA controller
3. **Configuration Descriptors** - A memory structure used to described a setup that should be applied to an processing modules like MMR write, scalar coefficient write etc.

9.4.8.8.1 Data Transfer Descriptors

In order to set up data transfers from the VPDMA, a data transfer descriptor is added into a list. The fields used for an Inbound and Outbound descriptor vary slightly. An outbound transfer can have two different outbound transfers. The two transfers are the main data transfer and a write of an Inbound Descriptor. The created inbound descriptor is formatted so it can be read back in directly to the next channel specified in the original descriptor.

Word	31:24			23:16				15:8 / 7:0
Word 0	Data Type	Notify	Field	1D	Even Line Skip	RSV	Odd Line Skip	Line Stride
Word 1	Line Length							Transfer Height
Word 2	Start Address						RSV	RSV
Word 3	Packet Type	Mode	Dr	Channel		Reserved	Pri	Next Channel
Word 4	Frame Width							Frame Height
Word 5	Horizontal Start							Vertical Start
Word 6	Client Specific Attributes							
Word 7	Client Specific Attributes							

vip-072

Figure 9-95. Inbound Data Transfer Descriptor Format

Word	Fields (bit 31 → bit 0)								
Word 0	Data Type (31:26)	Notify (25)	Field (24)	1D (23)	Reserved	Even Line Skip	Reserved	Odd Line Skip	Line Stride
Word 1									
Word 2	Start Address								
Word 3	Packet Type	Mode	Dir	Channel	NoReject	Reserved	Pri	Next Channel	
Word 4	Descriptor Write Address	Reserved	write descriptor	Reserved					
Word 5	Reserved	Max Width	Reserved	Max Height					
Word 6	Reserved								
Word 7	Reserved								

Figure 9-96. Outbound Data Transfer Descriptor Format

The general data transfer Descriptor formats can be seen in the above figures. The descriptor consists of 8 × 32bit words. A Data Transfer Descriptor will be removed from the list when the resource specified by the Channel field is free. If the Channel is not free when the list reaches a data transfer descriptor then the list will stall until the current transfer on the channel has completed.

9.4.8.8.1.1 Data Packet Descriptor Word 0 (Data)

Table 9-36. Data Packet Descriptor Word 0 Field Descriptions

Bit	Field	Value	Description
31:26	Data Type		**Miscellaneous Channel**
			Sets the pixel size in bits plus 1
			RGB Channel
		0	RGB16-565
		1h	ARGB-1555
		2h	ARGB-4444
		3h	RGBA-5551
		4h	RGBA-4444
		5h	ARGB24-6666
		6h	RGB24-888
		7h	ARGB32-8888
		8h	RGBA24-6666
		9h	RGBA32-8888
		10h	BGR16-565
		11h	ABGR-1555
		12h	ABGR-4444
		13h	BGRA-5551
		14h	BGRA-4444
		15h	ABGR24-6666
		16h	BGR24-888
		17h	ABGR32-8888
		18h	BGRA24-6666
		19h	BGRA32-8888
			YUV Channel
		0	Y 4:4:4
		1	Y 4:2:2
		2	Y 4:2:0
		4	C 4:4:4
		5	C 4:2:2
		6	C 4:2:0
		7	CY 4:2:2
		8	YCbC 4:4:4
		14h	Cb 4:4:4
		15h	Cb 4:2:2
		16h	Cb 4:2:0
		17h	CbY 4:2:2
		27h	YC 4:2:2
		37h	YCb 4:2:2
25	Notify	0-1	Send List Notification Interrupt upon last transfer of this channel
24	Field	0-1	Field Value
23	1D	0	The transfer is one dimensional. The transfer and frame sizes are combined to make a single 32 bit transfer size. For writes this value is passed to the generated descriptor. This feature is not supported by all clients. Only clients that support the feature will recognize this bit. **Note:** 1D mode is not supported by VIP modules

Table 9-36. Data Packet Descriptor Word 0 Field Descriptions (continued)

Bit	Field	Value	Description
22:20	Even Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
19	Reserved		Reserved for future use
18:16	Odd Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
15:0	Line Stride	0-FFFFh	Address stride between lines in bytes

9.4.8.8.1.1.1 Data Type

Bits 31-26 indicate the type of data that is to be transferred. This value is used to compute the number of bytes per pixel so that transactions may be made for the appropriate amount of data. The types of data are dependent on the type of channel. The VPDMA descriptor data types RGB or YUV are defined associated with the VPDMA channel assignment. This helps the engine to distinguish between the overlapping values of the descriptor data types for RGB and YUV data.

- For the Miscellaneous channel, the Data Type selects the size in bits of the data. The range is from 0 to 63 to represent the sizes from 1 to 64 bits.
- For a YUV channel, the Data Type determines, if the data channel is interleaved or color space separate. If color spaced separate, it is still assumed that the two chroma pixels are interleaved.

CAUTION

VPDMA defines the component ordering for its RGB data types in the opposite direction of what commonly used image identifiers expect. To avoid color component swapping in the display and/or in the video/image data written out to the memory, the proper Data Type settings for both RGB and YUV data types must be made. The following paragraphs provide more details on how to set Data Type correctly, in order to match the data stored or expected in the memory.

Setting RGB Data Types

The commonly used RGB format identifiers require the color components to be stored in a little-endian style, where the left most component is the LSB component.

- For an ARGB data type, the A component is the LSB location, as shown in Table 9-37 and Table 9-38;
- For a BGRA data type, the B component would be in the LSB location;

Table 9-37. Common ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
A	R	G	B

Table 9-38. Common ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
B	G	R	A

VPDMA specifies its component ordering in the big-endian style, which requires the data to be stored in the reversed order. Example with ARGB data type is shown in Table 9-39 and Table 9-40. The VPDMA ordering for ARGB data type matches the common BGRA data format.

Table 9-39. VPDMA ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
B	G	R	A

Table 9-40. VPDMA ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
A	R	G	B

In order color components to be mapped correctly and to avoid swapping, the reversal must be taken into consideration when configuring the Data Type in the VPDMA transfer descriptor.

Table 9-41 shows the proper settings required for RGB data types for both storage schemes.

Table 9-41. VPDMA Descriptor RGB Data Type Mapping

Source/Destination Image		VPDMA Data Type Mapping Value	
RGB Component order	Common Image Format Names	**Column A** Source data stored in the VPDMA defined order	**Column B** Source data stored in the opposite of VPDMA defined order
RGB	RGB16-565	0x0	0x10
	ARGB-1555	0x1	0x13
	ARGB-4444	0x2	0x14
	RGBA-5551	0x3	0x11
	RGBA-4444	0x4	0x12
	ARGB24-6666	0x5	0x18
	RGB24-888	0x6	0x16
	ARGB32-8888	0x7	0x19
	RGBA24-6666	0x8	0x15
	RGBA32-8888	0x9	0x17
BGR	BGR16-565	0x10	0x0
	ABGR-1555	0x11	0x3
	ABGR-4444	0x12	0x4
	BGRA-5551	0x13	0x1
	BGRA-4444	0x14	0x2
	ABGR24-6666	0x15	0x8
	BGR24-888	0x16	0x6
	ABGR32-8888	0x17	0x7
	BGRA24-6666	0x18	0x5
	BGRA32-8888	0x19	0x9

In Table 9-41, if the application uses the same data type definition as the VPDMA (that is, RGB24 refers to the B in the LSB), the data types in Column A should be used. But, if the application expects the common data type component order for RGB data type names, the VPDMA data types in Column B should be used.

For example:

- To display an ARGB32-8888 source image data with A in the LSB, the data type in the descriptor should be set to 0x19. But, to display an ARGB32-888 source image data with B in the LSB, the data type in the descriptor should be set to 0x7.
- To capture and write out a RGB24-888 image in the memory with R in the LSB, the data type in the descriptor should be set to 0x16 (this case assumes the VIP VIN d[23:0] data input is mapped to RGB bus with B component in the LSB).

Setting YUV Data Types

There is no component order reversal for YUV data types. The VPDMA uses generic data type names to specify the memory storage format and the application simply needs to follow the VPDMA defined ordering.

Table 9-42 shows how common YUV data types map to the VPDMA YUV data types in order to clarify the YUV data type configuration.

Table 9-42. VPDMA Descriptor YUV Data Type Mapping

Source YUV Image Types			VPDMA Data Type Mapping (Value)		
Chroma Sub-sample	Common YUV Image Format Type Names	Memory Packed Order [MSB - LSB]	Luma/Chroma Interleaved Channel	Luma-only Channel	Chroma-only Channel
444	YUV	V U Y	YC 4:4:4 (0x8)		
	UVY	Y V U	Cb 4:4:4 (0x14)		
422	NV16 (YUV422SP_UV)	V U		Y 4:2:2 (0x1)	C 4:2:2 (0x5)
	NV16 (YUV422SP_VU)	U V		Y 4:2:2 (0x1)	Cb 4:2:2 (0x15)
	YUV2/YUYV/V422 (YUV422I_YUYV)	V Y U Y	YC 4:2:2 (0x7)		
	YUV422I_YVYU	U Y V Y	CbY 4:2:2 (0x17)		
	Y422/UYVY (YUV422I_UYVY)	Y V Y U	YC 4:2:2 (0x27)		
	YUV422I_VYUY	Y U Y V	YCb 4:2:2 (0x37)		
420	NV12 (YUV420SP_UV)	V U		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	C 4:2:0 (0x6) YC 4:2:2 (0x7) (see (1))
	NV21 (YUV420SP_VU)	U V		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	Cb 4:2:0 (0x16) YC 4:2:2 (0x7) (see (1))

(1) If 422 source data is used, unused component data fetched (either Luma or Chroma) will be discarded.

For further details on the data formats, refer to Section 9.4.8.10, *VPDMA Data Formats*.

9.4.8.8.1.1.2 Notify

The Notify bit is used in conjunction with the Notify interrupts and when the channel has completed the DMA transfer the Notify Interrupt for the list that contains the descriptor will fire. The last Notify bit set for a specific list will be used so only a single Notify should be set in a list.

9.4.8.8.1.1.3 Field

The Field bit is the field value of the data that will be passed down to the clients. If the data is interlaced, then this value should be cleared to 0.

9.4.8.8.1.1.4 Even Line Skip

The Even Line skip is used with Line Stride to generate the next line address on an even line. All frames start on line 0. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

9.4.8.8.1.1.5 Odd Line Skip

The Odd Line skip is used with Line Stride to generate the next line address on an odd line. All frames start on line 0, so this will apply starting with the second line. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

9.4.8.8.1.1.6 Line Stride

Bits 15:0 are the stride between lines in bytes at the external address. This value is added or subtracted based upon an adjustment using the current skip value. Operation of the external address pointer shall load the Source Address upon start of the transfer, then at the end of each line increment or decrement by the value computed using the Line Stride and Skip value for the line. The line stride must be aligned to an L3 data bus width. The lower bits of the stride will always be treated as zero to force the alignment.

9.4.8.8.1.2 Data Packet Descriptor Word 1

Table 9-43. Data Packet Descriptor Word 1 Field Description

Bits	Name	Description
31:16	Line Length	Line Length in Pixels
15:0	Transfer Height	Number of rows in transfer.

9.4.8.8.1.2.1 Line Length

Bits 31-16 are the line length in pixels of the current channel. This is ignored for the outbound transfer as the client will provide the line length with the end of line signal on the client interface. The maximum supported line length is currently 4096.

9.4.8.8.1.2.2 Transfer Height

Bits 15-0 are the number of lines to be transferred in the current channel. This is ignored for outbound transfers as the client will provide the line length with the end of frame signal on the client interface. The maximum supported transfer size is currently 2048.

9.4.8.8.1.3 Data Packet Descriptor Word 2

Table 9-44. Data Packet Descriptor Word 2 Field Descriptions

Bit	Field	Value	Description
31:0	Start Address		32-bit data source address [31:0]
			If Mode is TILED, then TILER specific ADDRESS Map is used:
			Bits 31-29:
		0	0-degree view
		1h	180-degree view + mirroring
		2h	0-degree view + mirroring
		3h	180-degree view
		4h	270-degree view + mirroring
		5h	270-degree view
		6h	90-degree view
		7h	90-degree view + mirroring
			Bits 28-27:
		0	8-bit container
		1h	16-bit container
		2h	32-bit container
		3h	Page Mode
			If Mode is NORMAL, then bits 31-26 are the upper bits of the address.

9.4.8.8.1.3.1 Start Address

This is the byte aligned address for the first data transfer. The address on the OCP bus will always be word aligned.

9.4.8.8.1.4 Data Packet Descriptor Word 3

Table 9-45. Data Packet Descriptor Word 3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xa
26	Mode	0= Normal, 1=TILED
25	Direction	Inbound = 0, Outbound = 1

Table 9-45. Data Packet Descriptor Word 3 Field Descriptions (continued)

Bits	Name	Description
24:16	Channel	Channel for which this descriptor describes
15	Reserved	Reserved for future use
11:9	Priority	Only Bit 9 and Bit 11 are used to set the priority. Bit 10 is ignored. Highest = 0, Lowest = 3 By default, hardware assigns priority = 3. This priority level is used in the arbitration between the masters (for DDR access). See Section 9.4.8.8.1.4.5, *Priority*, for more details.
8:0	Next Channel	Next Channel to execute on a line or the channel to use in the generated write descriptor.

9.4.8.8.1.4.1 Packet Type

Bits 31:27 are a unique code which indicates that this Descriptor is a VPDMA descriptor.

9.4.8.8.1.4.2 Mode

Bit 26 is used to indicate if the transfer is to regular memory space or to Tiled memory space. The VPDMA will use this to determine if it does standard raster based addressing or if it assumes that the data is stored in TILER format. If data is stored in TILER format then the buffer is turned into 2 or 4 line buffers depending on the container type. For shared clients that use the memory, such as the ancillary data and the VIP port, only one can be active, if the mode field is set. Only clients that support Tiling will properly pack the data for tiling on the output interface. This must only be set for channels going to clients that support the TILING feature in the client configuration.

9.4.8.8.1.4.3 Direction

Bit 25 is used to indicate the direction of transfer. This bit indicates that the data flow is from an external source to an internal buffer (inbound) or data transfers form an internal buffer to an external location (outbound).

9.4.8.8.1.4.4 Channel

Bits 24:16 are the Channels which is supported by the descriptor. This is the identification of the specific Channel which is controlled by the contents of the descriptor. The channel assignments can be found in the channel table.

9.4.8.8.1.4.5 Priority

Bits 11:9 are set to indicate priority of the transfer, these are directly mapped to the OCP reqinfo bits.

9.4.8.8.1.4.6 Next Channel

Bits 8:0 give the next channel to use to create a composite frame. The next channel must be to a free channel. The last channel of a row should point back to the initial channel which must be a channel tied directly to a client. The Descriptor for the Next Channel must be of the same type as the current descriptor.

9.4.8.8.1.5 Data Packet Descriptor Word 4

9.4.8.8.1.5.1 Inbound data

Table 9-46. Data Packet Descriptor Word 4 Inbound Data Field Descriptions

Bits	Name	Description
31:16	Frame Width	Width of the client frame.
15:0	Frame Height	Height of the client frame

9.4.8.8.1.5.1.1 Frame Width

Bits 31:16 indicate the width in pixels of the frame. This is the width of the entire frame and not just the width of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum width is 4096.

9.4.8.8.1.5.1.2 Frame Height

Bits 15:0 indicate the height in pixels of the entire frame. This is the height of the entire frame and not just the height of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum height is 2048.

9.4.8.8.1.5.2 Outbound data

Table 9-47. Data Packet Descriptor Word 4 Outbound Data Field Descriptions

Bits	Name	Description
31:5	Descriptor Write Address	The 32 byte aligned location to write an inbound descriptor
2	Write Descriptor	If set to 1, a descriptor will be generated when the client completes a frame.
1	Drop Data	If set to 1, the data will not be written out. Also, if this is set, the write descriptor bit must be set. This allows for descriptors to only be written out so that software can determine the size of the required buffer before data is written out.
0	Use Descriptor Register	If set to 1, the CURRENT_DESCRIPTOR register will be used for the write address location. If set to 0, the Descriptor Write Address field in this word will be used for the Descriptor Write Address.

9.4.8.8.1.5.2.1 Descriptor Write Address

Bits 31:5 set the 32 byte address to write the generated descriptor. This address is only used if the Write Descriptor bit is set to 1 and the Use Descriptor Register bit is set to 0. If this case is met when the channel is complete a descriptor that meets the inbound descriptor format will be written to the location specified by this field. This allows for software to have a specific channel write its descriptor to a specific address no matter what order the channel completes compared to other outbound channels.

9.4.8.8.1.5.2.2 Write Descriptor

Bit 2 determines if a descriptor should be written out when the client is completed. If this bit is set the descriptor will be written. The format of the descriptor will be of an Inbound Data Transfer descriptor with the LINE LENGTH and TRANSFER HEIGHT fields determined by the counters in the clients. This means that the direction bit will be the opposite of the inbound descriptor. The FRAME WIDTH and FRAME HEIGHT values will match the LINE LENGTH and TRANSFER HEIGHT fields. The CHANNEL and NEXT CHANNEL fields will match the NEXT CHANNEL field of the original descriptor. The FIELD bit will match the source field captured by the client. The NOTIFY bit will always be 0. All other fields will match the original outbound descriptor. If the Outbound descriptor MODE is set to TILED data then the descriptor address used will be in TILED data space and software must ensure that the address is in page mode for the address of the descriptor.

9.4.8.8.1.5.2.3 Drop Data

Bit 1 determines if the data should be written out or not. In some cases software might only want to write the descriptor out to determine the size of the buffer that will be required to store incoming data. By setting this bit,no data will be written out. If this bit is set then the Write Descriptor bit MUST be set. Bit 0 determines where the descriptor should be written.

9.4.8.8.1.6 Data Packet Descriptor Word 5

9.4.8.8.1.6.1 Outbound data

Width and Height are set in the following register bit-fields:

- For Max_Size1: VIP_MAX_SIZE1[31:16] MAX_WIDTH and VIP_MAX_SIZE1[15:0] MAX_HEIGHT registers
- For Max_Size2: VIP_MAX_SIZE2[31:16] MAX_WIDTH and VIP_MAX_SIZE2[15:0] MAX_HEIGHT registers
- For Max_Size3: VIP_MAX_SIZE3[31:16] MAX_WIDTH and VIP_MAX_SIZE3[15:0] MAX_HEIGHT registers

Table 9-48. Data Packet Descriptor Word 5 Outbound Data Field Descriptions

Bits	Name	Description
6:4	Max Width	The maximum allowable pixels per line. 0: Unlimited Line Size 1: Use Max_Size1 Max Width field 2: Use Max_Size2 Max Width field 3: Use Max_Size3 Max Width field 4: 352 pixels 5: 768 pixels 6: 1280 pixels 7: 1920 pixels Others: Reserved
2:0	Max Height	The maximum allowable lines per frame. 0: Unlimited Frame Size 1: Use Max_Size1 Max Height field 2: Use Max_Size2 Max Height field 3: Use Max_Size3 Max Height field 4: 288 lines 5: 576 lines 6: 720 lines 7: 1080 lines Others: Reserved

9.4.8.8.1.6.1.1 Max Width

Bits 6:4 are encoded to set the maximum transferred line size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed pixels the data will continue to be received from the client but will not be sent to external memory until an end of line is received from the client sending data. The outbound descriptor if created will still have the transmitted size of the last line of the frame which will match the max width. If the frame does not exceed the max width then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0. If Max Width is 0 and the line size is larger then 4096 pixels then the address counters will overflow and the data at the start of the line will be overwritten.

9.4.8.8.1.6.1.2 Max Height

Bits 2:0 are encoded to set the maximum transferred frame size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed lines the data will continue to be received from the client but will not be sent to external memory until an end of frame is received from the client sending data. The outbound descriptor if created will still have the transmitted number of lines received which would match the max height. If the frame does not exceed the max height then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0.

9.4.8.8.2 Configuration Descriptor

The Configuration Descriptor is used in a List to setup a client configuration. The configuration descriptor consists of a header and a payload portion. The payload can either be part of the list that the descriptor is contained in or it can be at a separate location. The VPDMA will pass the payload of the configuration descriptor down to the client over the VPI Control port interface or through the MMR configuration port depending on the destination field. A Configuration Descriptor to any single destination except Destination 0 must be on a single list. Configuration Descriptors to different destinations may be on different lists..

The Configuration Descriptor Header is 4 × 32 bit words. A configuration descriptor is not consumed until the destination specified has received the entire configuration payload. Therefore the list is expected to stall while the configuration takes place. This ensures that all descriptors after a configuration descriptor in the list will occur after the desired configuration.

9.4.8.8.2.1 Configuration Descriptor Header Word0

Table 9-49. Configuration Descriptor Header Word0 Field Descriptions

Bits	Name	Description
31:0	Address Offset of the Destination	This is the address offset location in the destination that the block of data in the payload should be written. This field is only used if the descriptor Class is a block type. Otherwise this field is reserved.

9.4.8.8.2.2 Configuration Descriptor Header Word1

Table 9-50. Configuration Descriptor Header Word1 Field Descriptions

Bits	Name	Description
15:0	Number of Data Words	Length of First Data Packet for Class 1(block).

9.4.8.8.2.2.1 Number of Data Words

Bits 15:0 indicate the length of the first data block if the class is 1 as specified in Configuration Descriptor Header Word3. If the class is not 1 then this field should be set to 0.

9.4.8.8.2.3 Configuration Descriptor Header Word2

Table 9-51. Configuration Descriptor Header Word2 Field Descriptions

Bits	Name	Description
31:0	Payload Location	Pointer to the data payload

9.4.8.8.2.3.1 Payload Location

Bits 31:0 contain the pointer to the data payload if the command packet is an indirect type. This value along with the Payload Length will be combined to issue a DMA transaction that must complete before the List Manager can finish processing this descriptor. The list will be made inactive pending this DMA completion. This address should be on a 16 byte boundary so the lower 4 bits should always be 0.

9.4.8.8.2.4 Configuration Descriptor Header Word3

Table 9-52. Configuration Descriptor Header Word3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xb
26	Direct	Direct Command = 1 Indirect Command = 0
25:24	Class	0 = Address, Data Set 1 = Block 2,3 Reserved for Future Use
23:16	Destination	Destination of the configuration payload
15:0	Payload Length	Length of Payload in Words.

9.4.8.8.2.4.1 Packet Type

Bits 31:27, this field indicates the type of descriptor and should be set 0xB for configuration descriptor.

9.4.8.8.2.4.2 Direct

Bit 26, this field indicates that the descriptor is a direct command or indirect command. A direct command means that the payload is contiguous with the descriptor and will be pulled in by the list manager descriptor control as it processes the list. A direct payload must end on or before a 256 byte boundary in the list. An indirect command is when the Address is located in Word2 is a pointer to the data payload. A DMA transaction will be scheduled to bring the payload into the List Manager to allow for the payload to be sent to the destination.

9.4.8.8.2.4.3 Class

Bits 25:24, this field indicates the type of payload is associated with command descriptor, and thus how to handle the payload.

A 0 indicates that the payload consists of blocks of data with each block having an address and length. The first word after a sub-block contains the next address and length in bytes. This allows for writing to multiple configuration regions in a client. The Configuration Descriptor word1 is the first address and length pair. All addresses used must be larger then the previous address for all destinations except for the MMR destination. If a sub block length is not evenly divisible into 16 then zeroes should be padded in the payload so that the Next Client Address and Length field start on a 16 byte boundary.

9.4.8.8.2.4.3.1 Address Data Block Format

Table 9-53. Address Data Block Format Field Descriptions

Bits	Name	Description
31-0		Next Client Address
31-0		Configuration for Next Client Address
31-0		Configuration for Next Client Address + 4
31-0		Configuration for Next Client Address + 8
31-0		Configuration for Next Client Address + 12
31-0		Configuration for Next Client Address + 16
31-0		Next Client Address 2
15-0		Sub Block Length

A 1 indicates the payload is simply block data. The data is contiguous and starts at the offset of the destination as specified in word1.

9.4.8.8.2.4.4 Destination

The Destination field is used to determine where the configuration payload should be sent. The values for the destination field can be seen in the table below.

Table 9-54. Destination Field Description

Destination Value	Actual Destination	Description
0	mmr_client	Write to MMR registers to setup other modules
7	VIP Slice 0	VIP Slice 0 Scaler Coefficient Tables
8	VIP Slice 1	VIP Slice 1 Scaler Coefficient Tables

9.4.8.8.2.4.5 Descriptor Length

Bits 15:0, this field indicates the size of the payload in words for the command. This is 128 bit words. The maximum number of words is 1023 words in a single payload

9.4.8.8.3 Control Descriptor

9.4.8.8.3.1 Generic Control Descriptor Format

A control descriptor is the mechanism that is used in a list to give commands to the List Manager. These descriptors are not considered consumed until the List Manager has done the instruction required by the descriptor. All control descriptors have a common Header located at Word 3 but the remaining words are based on the specific control descriptor.

9.4.8.8.3.2 Control Descriptor Header Description

Table 9-55. Control Descriptor Header Description

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xc
26:25	Reserved	Reserved
24:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	The type of control descriptor that should be run by the List Manager

9.4.8.8.3.2.1 Packet Type

This field indicates a VPDMA control descriptor.

9.4.8.8.3.2.2 Source

The source is combined with the control field to determine what the control descriptor should use as the source for its synchronization event.

9.4.8.8.3.2.3 Control

The Control field defines the specific function of the descriptor. Table 9-56 lists the different control descriptors.

9.4.8.8.3.3 Control Descriptor Types

Table 9-56. Control Descriptor Types Summary

Control Descriptor	Control Field Value	Description
Sync on Client	0	Wait for the client attached to the channel specified to reach a certain event before proceeding.
Sync on List	1h	Wait for another list(s) to reach its Sync on List Descriptor
Sync on External Event	2h	Wait for a Register write to the LIST_STAT_SYNC bit specified or for an external event.
Sync on Channel	4h	Wait for the channel specified to complete
Change Client Interrupt	5h	Change the interrupt event for a client interrupt but do not wait for the event to occur.
Send Interrupt	6h	Generate an interrupt event on one of the Control Descriptor Interrupts
Reload List	7h	Reload the list from the location and size specified.
Abort Channel	8h	Abort the transfer in the channel specified.

9.4.8.8.3.3.1 Sync on Client

A Sync on Client Control descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. For a client that supports multiple channels then only an event on the portion of the client that supports that client will cause the interrupt to be generation. After configuring the interrupt generation event the list will then stall until that event has occurred.

Table 9-57. Sync on Client Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.

Table 9-57. Sync on Client Field Descriptions (Word - 1) (continued)

Bits	Name	Description
15:0	LINE_COUNT	Specify the line where a line based event would trigger

Table 9-58. Sync on Client Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 0

9.4.8.8.3.3.2 Sync on List

The Sync on List Control descriptor is a mechanism to ensure that multiple lists have all reached a common point. The Sync on List descriptor uses the Source field as a mask for each list that should be synchronized which must include the list where the control descriptor resides. Once all lists specified in the source field have reached their Sync on List descriptor all lists will resume with the lowest list number resuming first. All Sync on List Control descriptors in each of the lists must have the same source field so that all lists will synchronize upon reaching that point.

If it is desired to synchronize list 0 and list 1 then the source field would be 0x3. If it is desired to synchronize list 1, list 3, and list 4 then the source field would be 0x1a. Word 0, Word 1 and Word 2 are reserved.

Table 9-59. Sync on List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 1h

9.4.8.8.3.3.3 Sync on External Event

A Sync on External Event descriptor ensures an external event has occurred before proceeding. The external event can be a write to a bit of the LIST_STAT_SYNC register or other external events that are determined at VPDMA elaboration to have occurred. This descriptor can be used to allow for external software to control progression of the list. The descriptor can also be used to select external signals that are brought into the VPDMA. The current implementation just synchronize on the LIST_STAT_SYNC bit for the list number that called the descriptor.

Table 9-60. Sync on External Event Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 2h

9.4.8.8.3.3.4 Sync on Channel

A Sync on Channel descriptor stalls the list until the channel specified is free. If the channel is already free then the descriptor will not cause the list to stall. Word 0, Word 1 and Word 2 are reserved.

Table 9-61. Sync on Channel Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 4h

9.4.8.8.3.3.5 Sync on LM Timer

A Sync on LM Timer descriptor sets a value from the current timer position to wait. The LM timer is a free running counter at the LM processing clock. The Timer Value in the descriptor is added to the value of the timer at the time the descriptor is received and the list will stall for this many cycles before it becomes active again.

9.4.8.8.3.3.6 Change Client Interrupt

A Change Channel Interrupt Source descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. The list will not stall on this descriptor. The format of the fields is identical to the Sync on Client Descriptor. Word 0 is reserved.

Table 9-62. Change Client Interrupt Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.
15:0	LINE_COUNT	Specify the line where a line based event would trigger.

Table 9-63. Change Client Interrupt Field Descriptions (Word - 2)

Bits	Name	Description
31:4	Reserved	Reserved
3:0	Event	Specify the event which should trigger the client interrupt.

Table 9-64. Change Client Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 5h

9.4.8.8.3.3.7 Send Interrupt

A Send Interrupt descriptor will cause the VPDMA to generate an interrupt on the list manager controlled interrupts as specified by the Source Field. The list will not stall on this descriptor. For example, if source is 0 then control_descriptor_int0 will fire. If source is 12, then control_descriptor_int12 will fire. For more information ot VPDMA interrupt events, see Section 9.4.8.7, *VPDMA Interrupts*.

Table 9-65. Send Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved

Table 9-65. Send Interrupt Field Descriptions (Word - 3) (continued)

Bits	Name	Description
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 6h

9.4.8.8.3.3.8 Reload List

A Reload List descriptor causes ending descriptors after this descriptor in the original list to be dropped and a new list at the location and of the size specified in the descriptor. This descriptor can be used to allow for linked lists in the VPDMA as the list will continue from this point and fetch the list specified and it does not have to be continuous with the current list.

Table 9-66. Reload List Field Descriptions (Word - 0)

Bits	Name	Description
31:0	LIST_ADDRESS	31 most-significant Bits of the memory address where the descriptors to be loaded are stored. Address must be 16 byte aligned.

Table 9-67. Reload List Field Descriptions (Word - 1)

Bits	Name	Description
31:16	Reserved	Reserved
15:0	LIST_SIZE	Size of the list to load

Table 9-68. Reload List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 7h

9.4.8.8.3.3.9 Abort Channel

An Abort Channel descriptor is used to clear a channel. This clears the channel from issuing any more requests. Any outstanding requests for that channel will complete as originally scheduled. All data inside the client will be flushed. For Tiled Clients such as the noise filter it is required to issue two consecutive abort channel descriptors to ensure the channel is aborted for both the current tile and next tile. Word 0, Word 1 and Word 2 are reserved.

Table 9-69. Abort Channel Field Descriptions (Word - 3)

Bit	Name	Description
31:27	Packet Type	Host Packet Descriptor type = 0xC
26:24	Reserved	Reserved
23:16	Source	VPDMA Channel Number whose transfers are to be aborted
15:4	Reserved	Reserved
3:0	Control	Control type = 9h

9.4.8.9 VPDMA Configuration

The following section describes the different ways of configuring VPDMA for data transfers.

9.4.8.9.1 Regular List

A regular list executes each descriptor in order until the end of the list is reached. When the end of the list is reached an interrupt is sent and the list can be reused by software. A regular list can contain any descriptor types. Software creates the list at some location in external memory. After completing writing the list software then writes the location of the list to the VIP_LIST_ADDR[31:0] VIP_LIST_ADDR register and then writes the VIP_LIST_ATTR register. If the NUMBER in the VIP_LIST_ATTR[26:24] LIST_NUM is not an active list then the List will be loaded and begin to execute the next time the List Manager gets to IDLE after processing previous loaded lists. If the NUMBER in the VIP_LIST_ATTR[26:24] LIST_NUM is busy then the VIP_LIST_ADDR and VIP_LIST_ATTR registers will be locked until the active list specified by NUMBER completes.

The different ports inside VPDMA requires different list setup, as explained in the following sections.

9.4.8.9.2 Video Input Ports

The Video Input Ports can be used in multiple ways. Depending on how the input ports are configured will determine how the lists to manage them need to be setup. The ways in which the ports can work are multiplexed data stream, single YUV color separate stream, dual YUV interleaved or single RGB stream. Each port also has an ancillary data port that can run either multiplexed or single data stream and shares the buffering with the main data stream. For all cases the descriptor must be loaded into the client before the vertical sync is received on the VIP port or the entire frame will be dropped. As with all write clients the VIP channels can be shadowed so the next frame/field descriptor can be loaded while the previous frame is running without stalling the list.

9.4.8.9.2.1 Multiplexed Data Streams

In the case of a multiplexed data stream input the channels that should be used are VIP(X)_MULT_PORT(Y)_SRC(Z). Where X is the specific VIP slice of the instance that wants to be used and port Y is the port A or port B that is receiving the data. Finally Z is the channel number. For Split line mux mode the LSB of the channel will determine, if the line is a split line or a complete line. This is required so that the data streams do not get mixed when a channel ends without completing a line. In this mode the data will always be sent out as 422 Interleaved data to the destination specified in the descriptor.

9.4.8.9.2.2 Single YUV Color Separate

If the port is configured to be used as sending out color separated YUV data then the channels that must be used are VIPX_PORTY_LUMA and VIPX_PORTY_CHROMA for the luma and chroma components respectively. The data type can be either 420 or 422 color separate in this case.

9.4.8.9.2.3 Dual YUV Interleaved

If the port is configured to be used to send out 422 interleaved data from a non-multiplexed source then the channels that must be used are VIPX_PORTY_LUMA or VIPX_PORTY_CHROMA depending on if the data stream is connected to the luma or chroma port. The data type must be 422 interleaved in this case.

9.4.8.10 VPDMA Data Formats

Each of the channels has a type of data that it can support based upon the client that it services. Also for each channel, the data type will assume that data is packed in memory a certain way and will be prevented to the client in the same manner to the client no matter what the format of the data in memory.

9.4.8.10.1 YUV Data Formats

The YUV formatted channels expect video data that is in YUV color space. The YUV data can be type is for YUV data and it can support both interleaved data where Luma and Chroma are in the same data buffer or it can support co-planar data where the Luma and Chroma are in separate data buffers. The YUV channel also can be given a position to give a different start position for the client instead of the default upper left hand corner of the frame. The storage format for YUV data depends on the data type field. The data type must be set to a data type that the channel can support. All the clients that data are provided too will either accept Luma Only, Chroma Only or Luma and Chroma in a parallel data bus.

9.4.8.10.1.1 Y 4:4:4 (Data Type 0)

The Y 4:4:4 data type is used for a co-planar 4:4:4 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and height set to the desired frame size expected by the receiving client.

Figure 9-97. Y 4:4:4 (Data Type 0)

9.4.8.10.1.2 Y 4:2:2 (Data Type 1)

The Y 4:2:2 data type is used for a co-planar 4:2:2 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the desired frame sent by the VPDMA to the receive client.

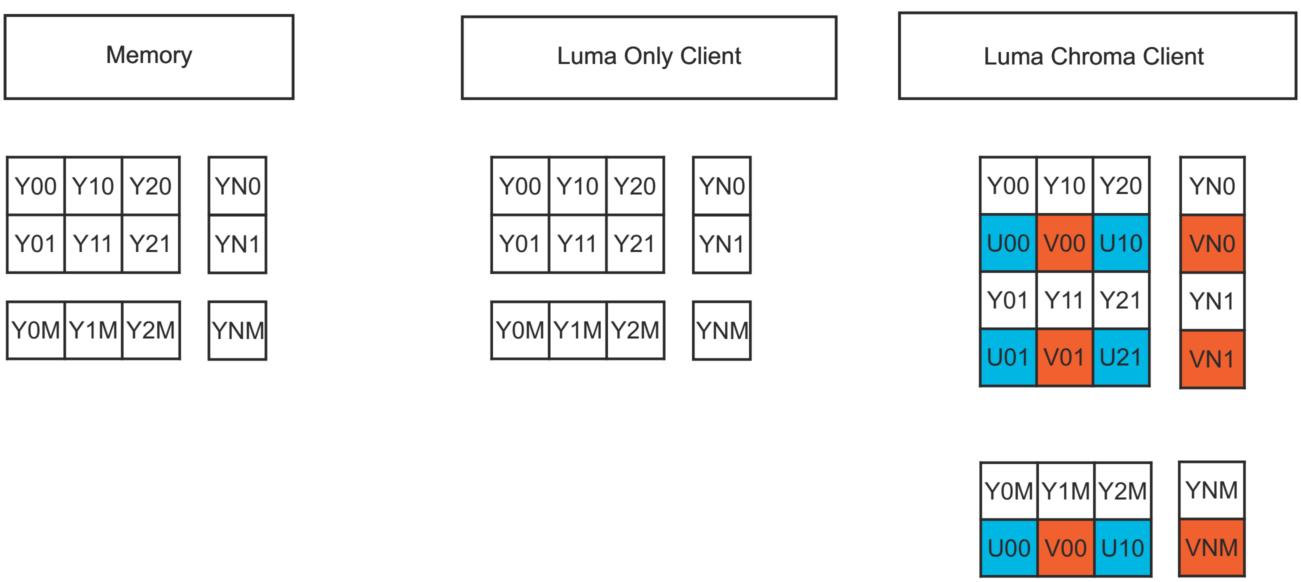


Figure 9-98. Y 4:2:2 (Data Type 1)

9.4.8.10.1.3 Y 4:2:0 (Data Type 2)

The Y 4:2:0 data type is used for a co-planar 4:2:0 data type. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the expected frame for the client.

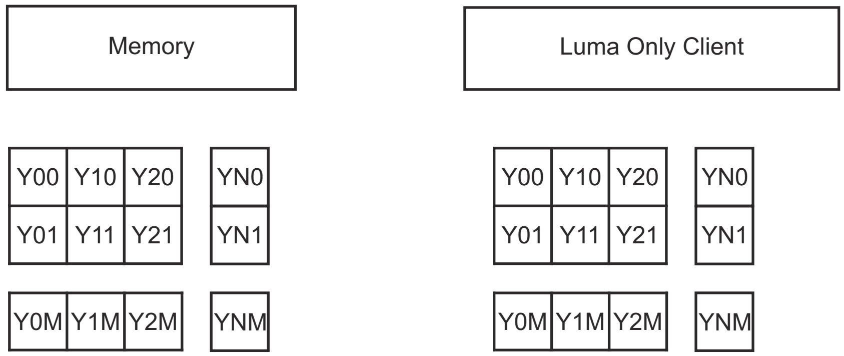


Figure 9-99. Y 4:2:0 (Data Type 2)

9.4.8.10.1.4 C 4:4:4 (Data Type 4)

The C 4:4:4 data type is used for a co-planar 4:4:4 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and height of the expected client frame.

Figure 9-100. C 4:4:4 (Data Type 4)

9.4.8.10.1.5 C 4:2:2 (Data Type 5)

The C 4:2:2 data type is used for co-planar 4:2:2 data.It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and the height of the expected client frame.

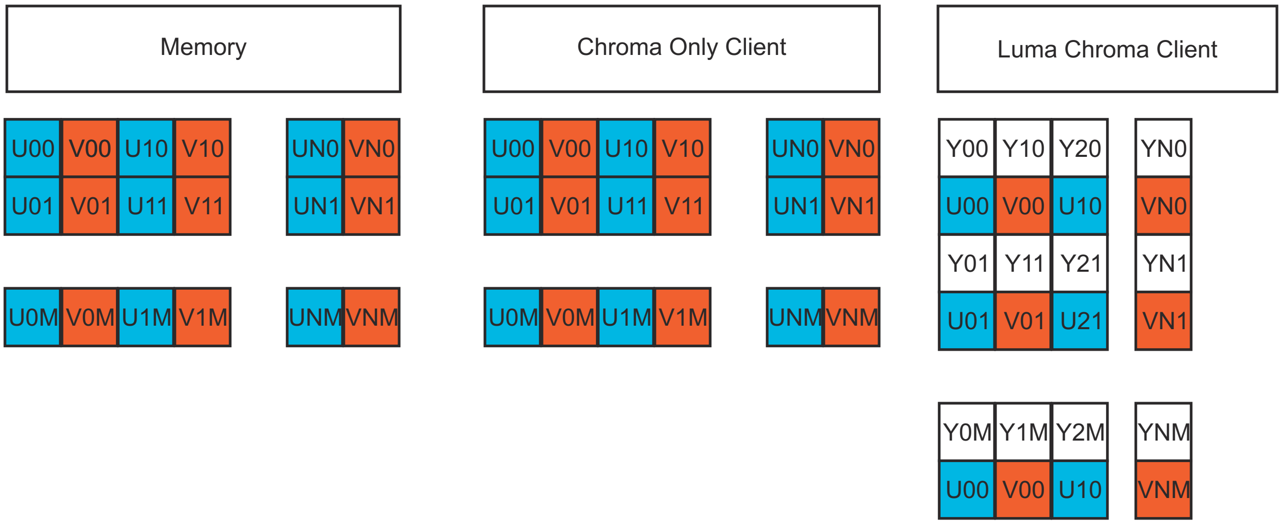


Figure 9-101. C 4:2:2 (Data Type 5)

9.4.8.10.1.6 C 4:2:0 (Data Type 6)

The C 4:2:0 data type is used for a co-planar 4:2:0 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in a 16 bit container. This data block should have the width and half the height of the expected clients frame.

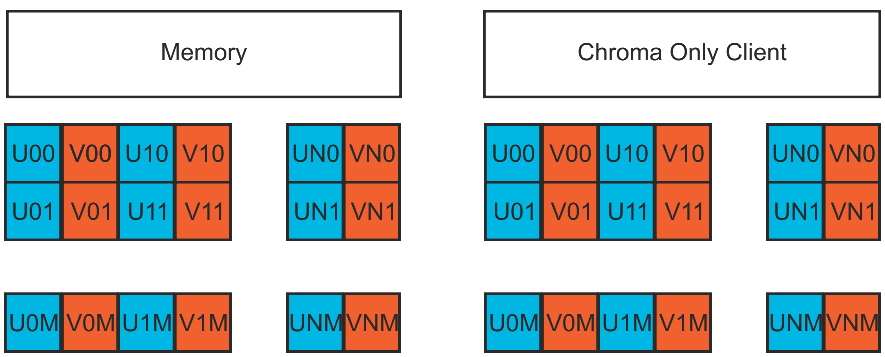


Figure 9-102. C 4:2:0 (Data Type 6)

9.4.8.10.1.7 YC 4:2:2 (Data Type 7)

The YC 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Y1 finally Cr in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

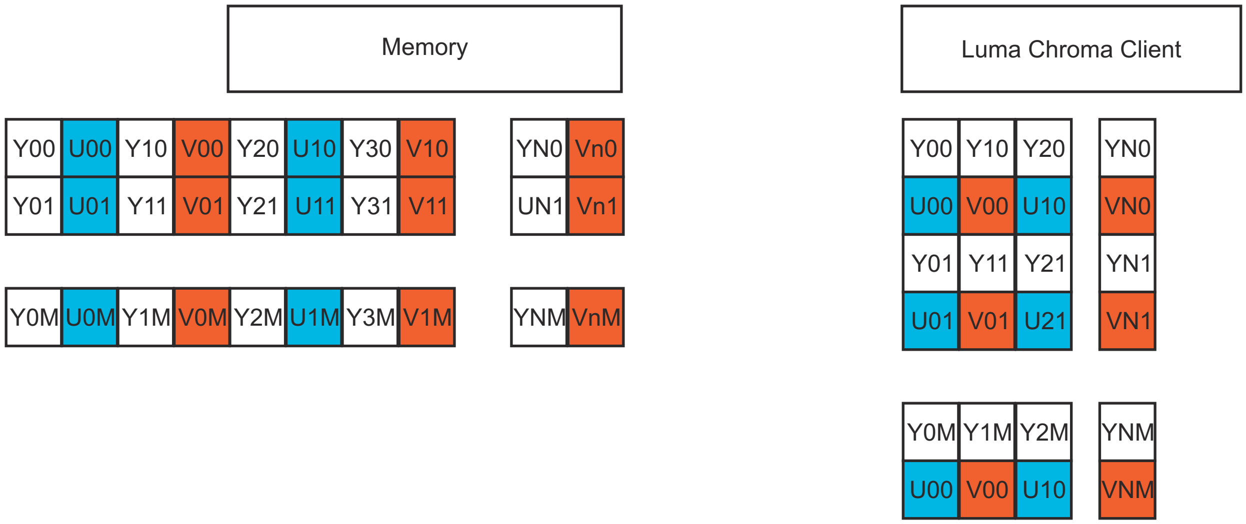


Figure 9-103. YC 4:2:2 (Data Type 7)

9.4.8.10.1.8 YC 4:4:4 (Data Type 8)

The YC 4:4:4 data type is used for interleaved 4:4:4 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Cr. The transfer counts each YCbCr triplet in a 24 bit word as a pixel. A 2D transfer of the data is NOT supported in the VPDMA. The data block width and height should be set to the number of pixels and lines that are expected by the client.

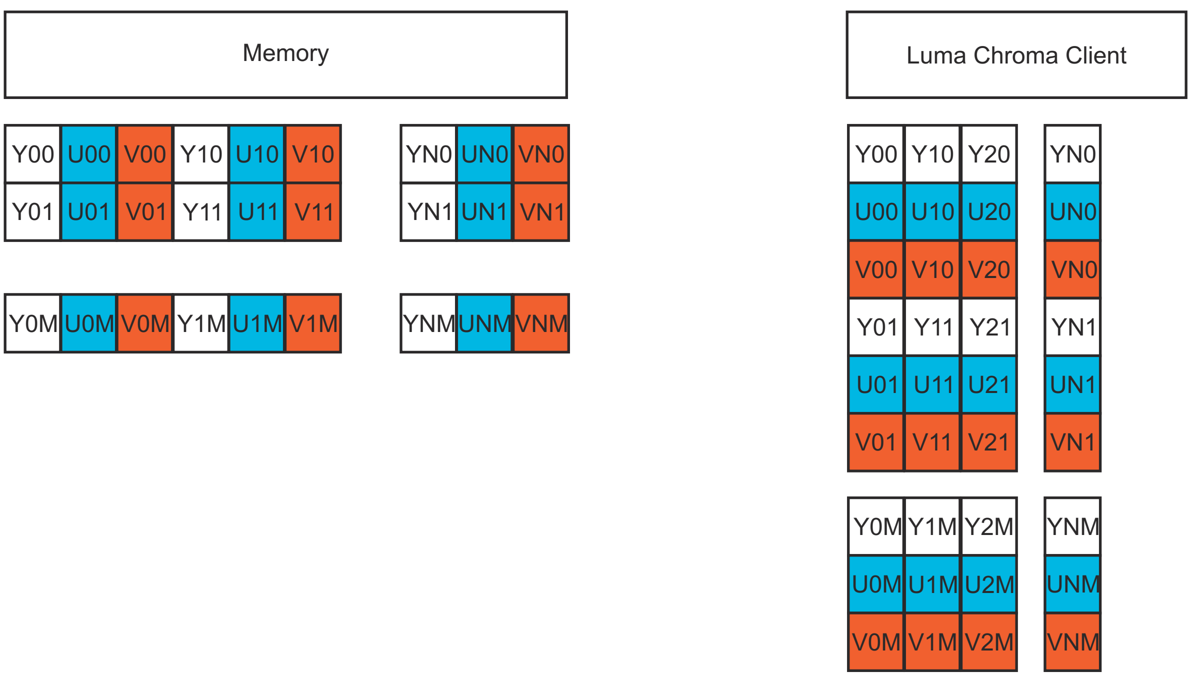


Figure 9-104. YC 4:4:4 (Data Type 8)

9.4.8.10.1.9 CY 4:2:2 (Data Type 23)

The CY 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Cb in the lowest byte followed by Y0 followed by Cr finally Y1 in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

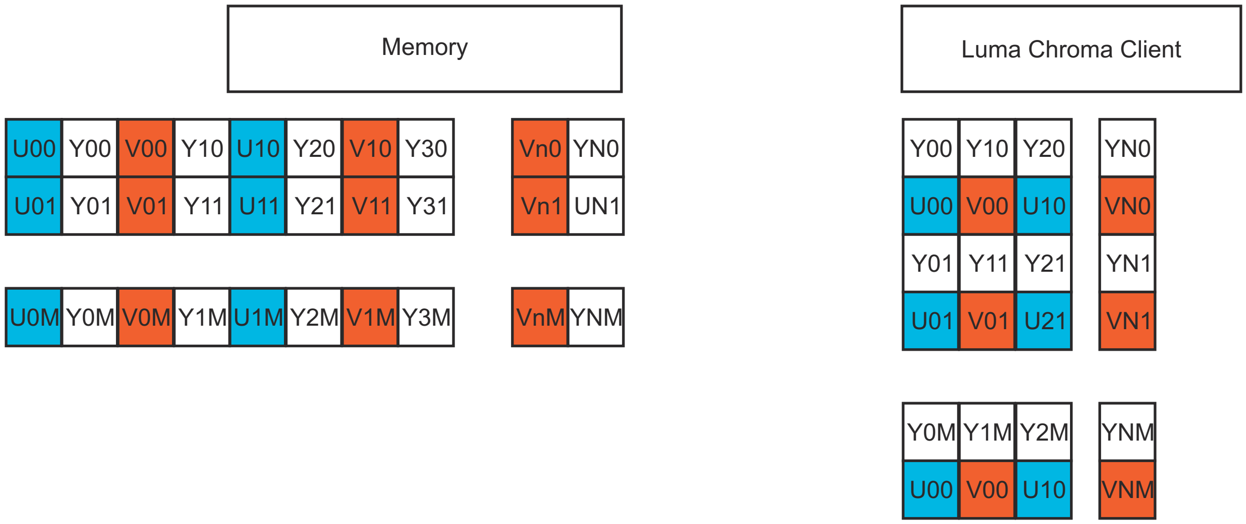


Figure 9-105. CY 4:2:2 (Data Type 23h)

9.4.8.10.2 RGB Data Formats

The RGB channel type is used to provide data for a client that expects to transmit RGB data. In all modes the client is always RGBA 8888 data. The lower bits, if not provided by the data stream, are a replication of the lower bit of the data. For outbound data the input is assumed to be RGB 888 and the Alpha value is taken from the Background Color register to make a full ARGB 8888. The lower bits are dropped from this data, if a data type specifies less then the full 8 bits per color. The client has individual data buses for each component so they have no order dependency in the data bus.

9.4.8.10.2.1 RGB16-565 (Data Type 0)

In RGB16-565 mode, each pixel is a single RGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the middle 6 bits for green data and the upper 5 bits for red data.

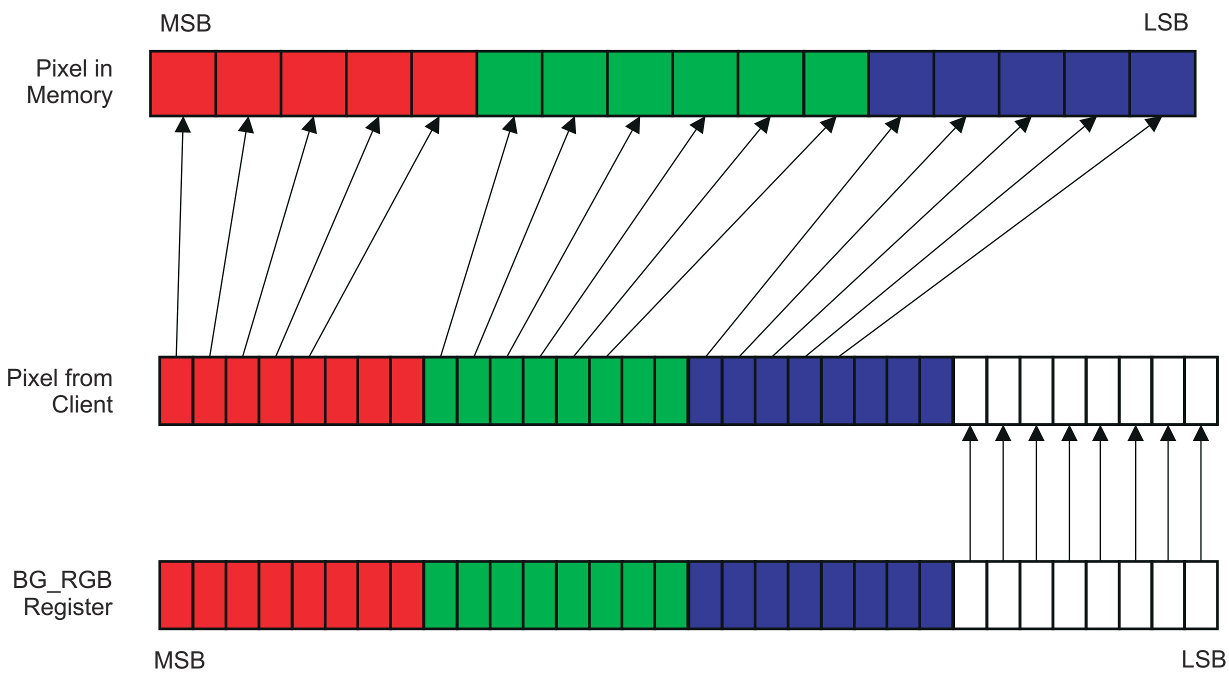


Figure 9-106. RGB16-565 (Data Type 0)

9.4.8.10.2.2 ARGB-1555 (Data Type 1)

In ARGB-1555 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data and the upper most bit for the blend value to use 0 or 0xFF.

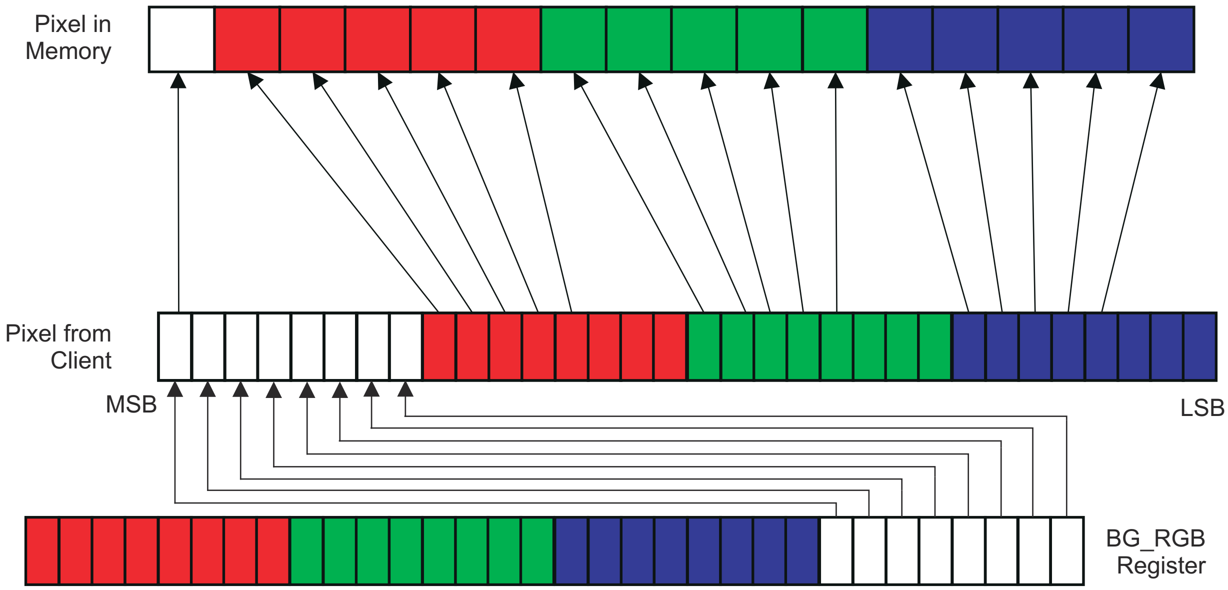


Figure 9-107. ARGB-1555 (Data Type 1)

9.4.8.10.2.3 ARGB-4444 (Data Type 2)

In ARGB16-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for blue data, the next 4 bits for green data the next 4 bits for red data and the upper most 4 bits for the blend value.

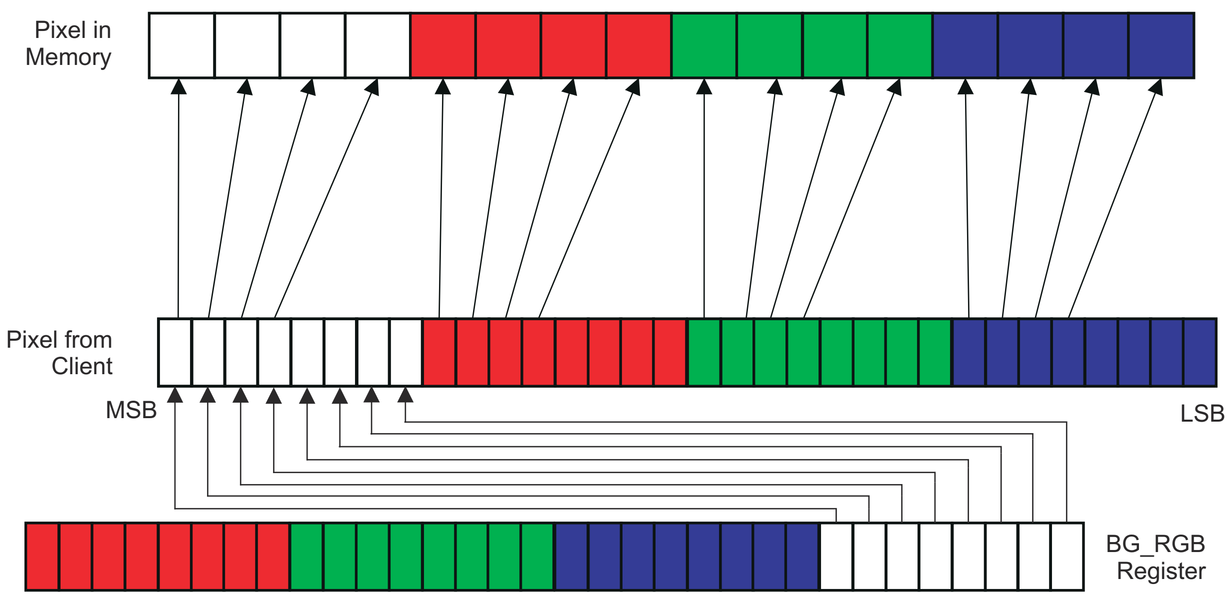


Figure 9-108. ARGB-4444 (Data Type 2)

9.4.8.10.2.4 RGBA-5551 (Data Type 3)

In RGBA-5551 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest most bit for the blend value to use 0 or 0xff the next 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data.

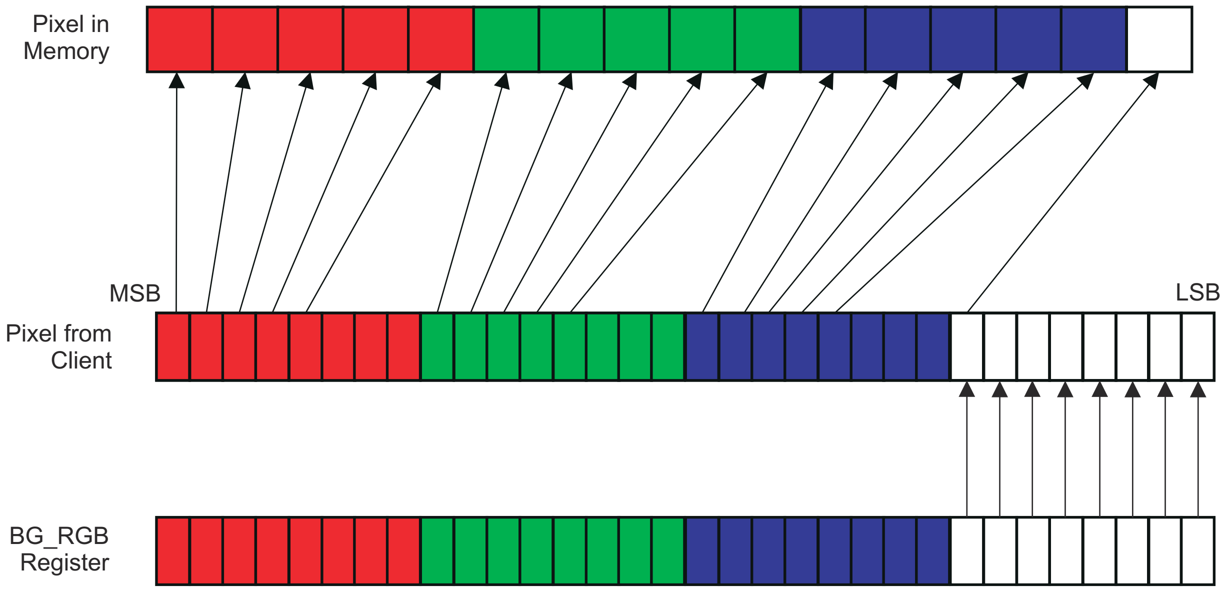


Figure 9-109. RGBA-5551 (Data Type 3)

9.4.8.10.2.5 RGBA-4444 (Data Type 4)

In RGBA-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for the blend value, the next 4 bits for blue data, the next 4 bits for green data and the upper most 4 bits for red data.

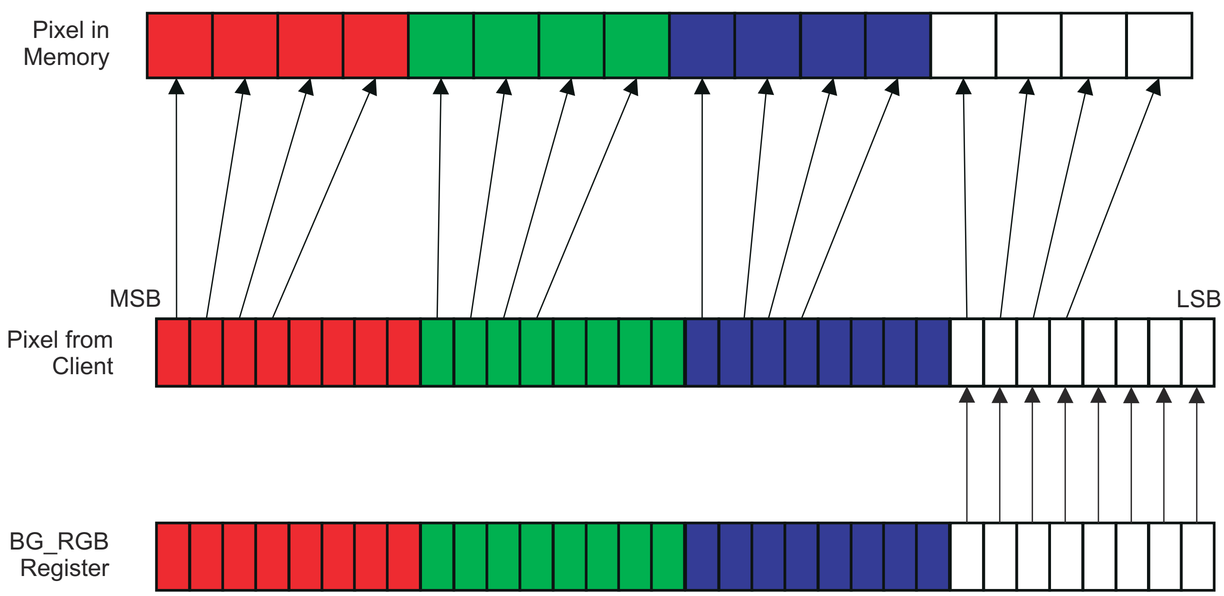


Figure 9-110. RGBA-4444 (Data Type 4)

9.4.8.10.2.6 ARGB24-6666 (Data Type 5)

In ARGB24-6666 mode, each pixel is a single ARGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper most 6 bits for red data.

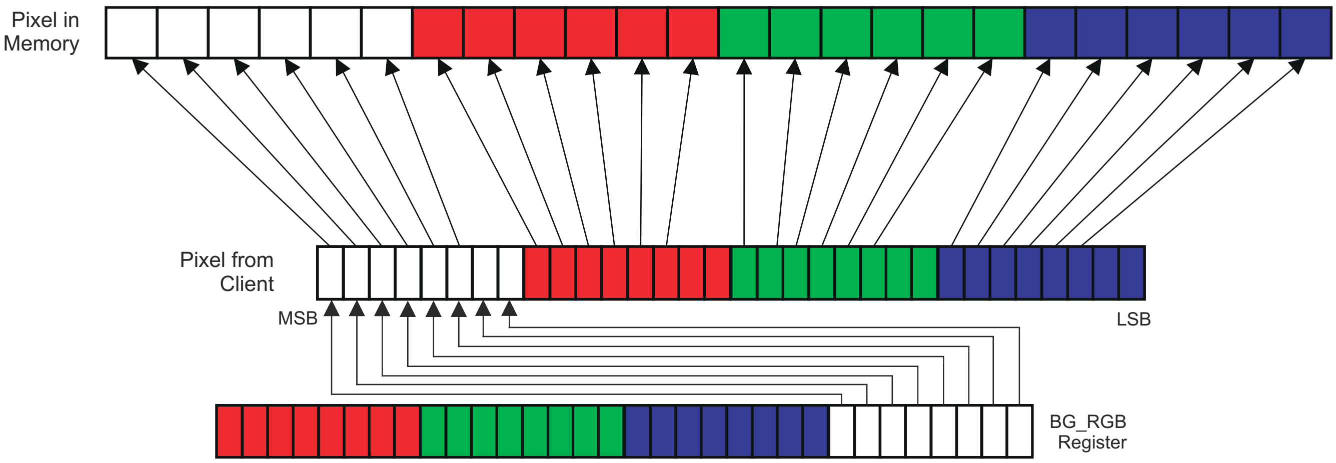


Figure 9-111. ARGB24-6666 (Data Type 5)

9.4.8.10.2.7 RGB24-888 (Data Type 6)

In RGB24-888 mode, each pixel is a single RGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data it uses the BG_RGB Blend value for the Blend value.

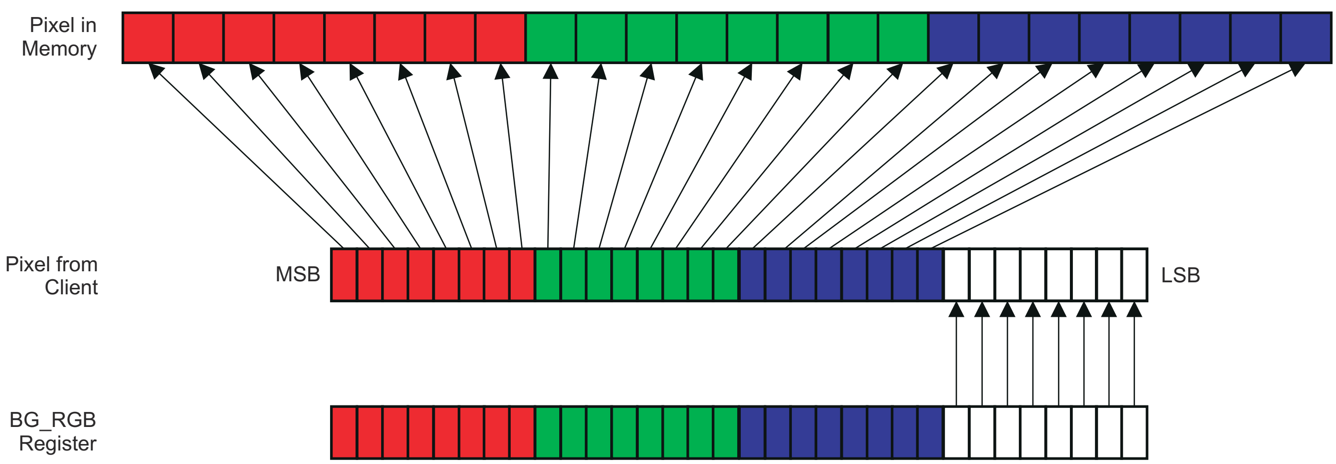


Figure 9-112. RGB24-888 (Data Type 6)

9.4.8.10.2.8 ARGB32-8888 (Data Type 7)

In ARGB32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bitsof data uses the lowest 8 bits for blue data, the next 8 bits for green data and the next 8 bits for red data it uses the upper most bits for the blend value.

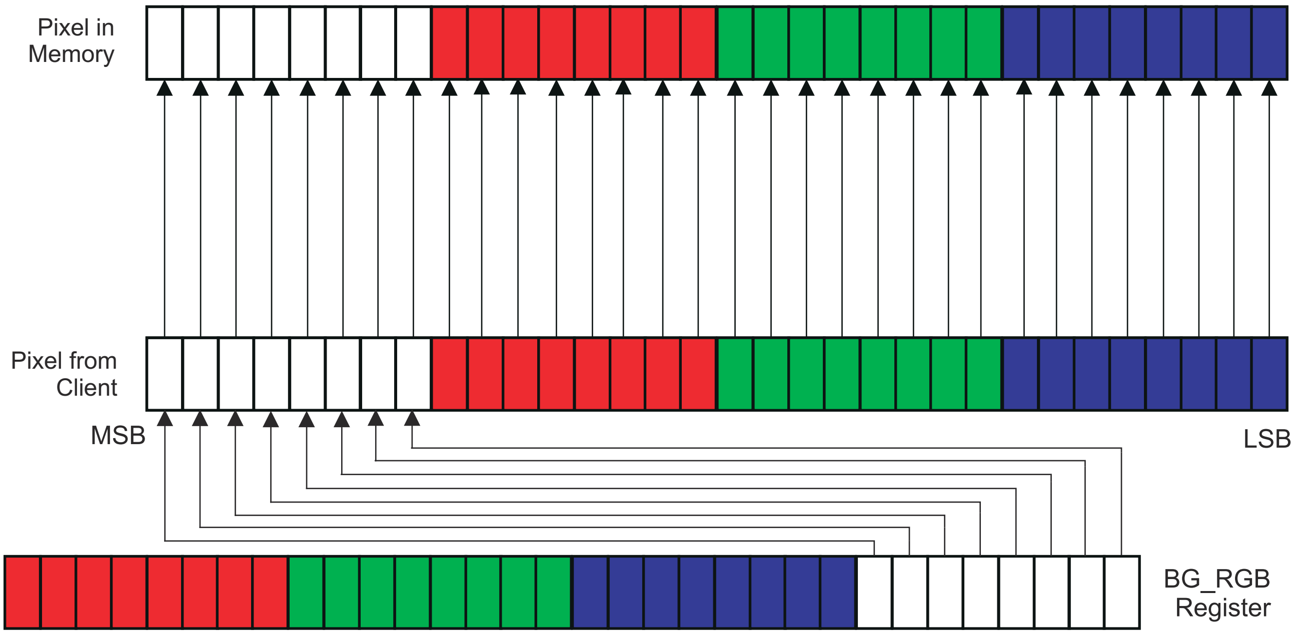


Figure 9-113. ARGB32-8888 (Data Type 7)

9.4.8.10.2.9 RGBA24-6666 (Data Type 8)

In RGBA24-6666 mode, each pixel is a single RGBA pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper 6 bits for red data.

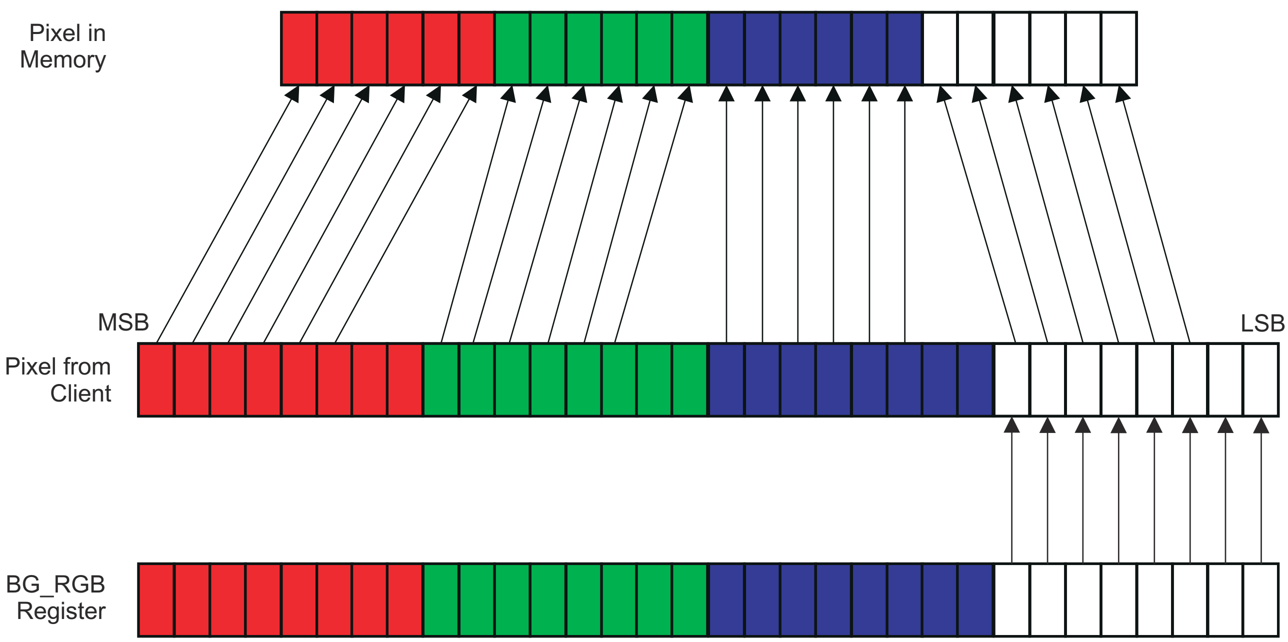


Figure 9-114. RGBA24-6666 (Data Type 8)

9.4.8.10.2.10 RGBA32-8888 (Data Type 9)

In RGBA32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bits of data uses the lowest 8 bits for the blend value the next 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data.

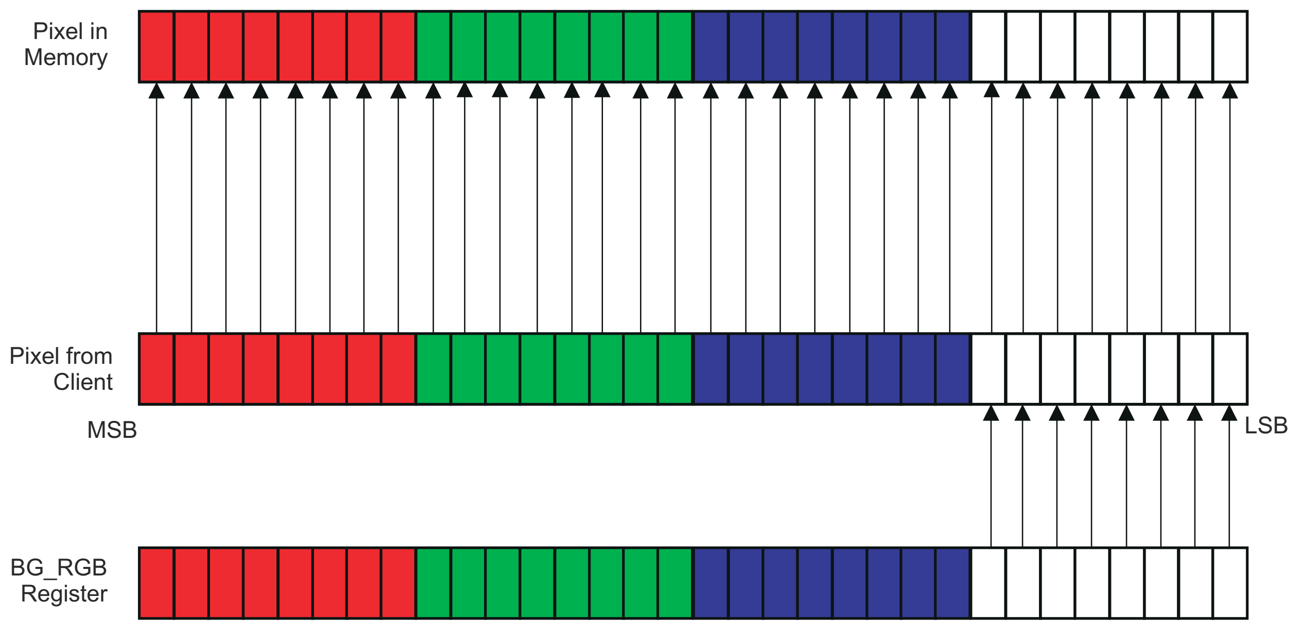


Figure 9-115. RGBA32-8888 (Data Type 9)

9.4.8.10.3 Miscellaneous Data Type

The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client. The size of the miscellaneous data type is always determined by the Data Type field of the descriptor which specifies the number of bits in the descriptor.

A memory structure used to describe a desired memory transaction to or from a client. The descriptor at a minimum gives an address location for the memory portion of the transfer, the channel to use for this transaction and the size of the transaction. The data descriptor can also contain attributes to be passed down to the client or be linked to another data descriptor to form a larger frame from many smaller frames.

9.5 VIP Register Manual

9.5.1 VIP Instance Summary

Table 9-70. VIP Instance Summary

Module Name	Module Base Address	Size
VIP1_top_level	0x4897 0000	276 Bytes
VIP1_Slice0_parser	0x4897 5500	216 Bytes
VIP1_Slice0_csc	0x4897 5700	24 Bytes
VIP1_Slice0_sc	0x4897 5800	128 Bytes
VIP1_Slice1_parser	0x4897 5A00	216 Bytes
VIP1_Slice1_csc	0x4897 5C00	24 Bytes
VIP1_Slice1_sc	0x4897 5D00	128 Bytes
VIP1_VPDMA	0x4897 D000	1016 Bytes
VIP2_top_level	0x4899 0000	276 Bytes
VIP2_Slice0_parser	0x4899 5500	216 Bytes
VIP2_Slice0_csc	0x4899 5700	24 Bytes
VIP2_Slice0_sc	0x4899 5800	128 Bytes
VIP2_Slice1_parser	0x4899 5A00	216 Bytes
VIP2_Slice1_csc	0x4899 5C00	24 Bytes
VIP2_Slice1_sc	0x4899 5D00	128 Bytes

Table 9-70. VIP Instance Summary (continued)

Module Name	Module Base Address	Size
VIP2_VPDMA	0x4899 D000	1016 Bytes
VIP3_top_level	0x489B 0000	276 Bytes
VIP3_Slice0_parser	0x489B 5500	216 Bytes
VIP3_Slice0_csc	0x489B 5700	24 Bytes
VIP3_Slice0_sc	0x489B 5800	128 Bytes
VIP3_Slice1_parser	0x489B 5A00	216 Bytes
VIP3_Slice1_csc	0x489B 5C00	24 Bytes
VIP3_Slice1_sc	0x489B 5D00	128 Bytes
VIP3_VPDMA	0x489B D000	1016 Bytes

9.5.2 VIP Top Level Registers

9.5.2.1 VIP Top Level Register Summary

Table 9-71. VIP Top Level Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_top_level Base Address	VIP2_top_level Base Address	VIP3_top_level Base Address
VIP_CLKC_PID	RW	32	0x0000 0000	0x4897 0000	0x4899 0000	0x489B 0000
VIP_SYSCONFIG	RW	32	0x0000 0010	0x4897 0010	0x4899 0010	0x489B 0010
VIP_INTC_INTR0_STATUS_RAW0	RW	32	0x0000 0020	0x4897 0020	0x4899 0020	0x489B 0020
VIP_INTC_INTR0_STATUS_RAW1	RW	32	0x0000 0024	0x4897 0024	0x4899 0024	0x489B 0024
VIP_INTC_INTR0_STATUS_ENA0	RW	32	0x0000 0028	0x4897 0028	0x4899 0028	0x489B 0028
VIP_INTC_INTR0_STATUS_ENA1	RW	32	0x0000 002C	0x4897 002C	0x4899 002C	0x489B 002C
VIP_INTC_INTR0_ENA_SET0	RW	32	0x0000 0030	0x4897 0030	0x4899 0030	0x489B 0030
VIP_INTC_INTR0_ENA_SET1	RW	32	0x0000 0034	0x4897 0034	0x4899 0034	0x489B 0034
VIP_INTC_INTR0_ENA_CLR0	RW	32	0x0000 0038	0x4897 0038	0x4899 0038	0x489B 0038
VIP_INTC_INTR0_ENA_CLR1	RW	32	0x0000 003C	0x4897 003C	0x4899 003C	0x489B 003C
VIP_INTC_INTR1_STATUS_RAW0	RW	32	0x0000 0040	0x4897 0040	0x4899 0040	0x489B 0040
VIP_INTC_INTR1_STATUS_RAW1	RW	32	0x0000 0044	0x4897 0044	0x4899 0044	0x489B 0044
VIP_INTC_INTR1_STATUS_ENA0	RW	32	0x0000 0048	0x4897 0048	0x4899 0048	0x489B 0048
VIP_INTC_INTR1_STATUS_ENA1	RW	32	0x0000 004C	0x4897 004C	0x4899 004C	0x489B 004C
VIP_INTC_INTR1_ENA_SET0	RW	32	0x0000 0050	0x4897 0050	0x4899 0050	0x489B 0050
VIP_INTC_INTR1_ENA_SET1	RW	32	0x0000 0054	0x4897 0054	0x4899 0054	0x489B 0054
VIP_INTC_INTR1_ENA_CLR0	RW	32	0x0000 0058	0x4897 0058	0x4899 0058	0x489B 0058
VIP_INTC_INTR1_ENA_CLR1	RW	32	0x0000 005C	0x4897 005C	0x4899 005C	0x489B 005C
VIP_INTC_EOI	RW	32	0x0000 00A0	0x4897 00A0	0x4899 00A0	0x489B 00A0
VIP_CLKC_CLKEN	RW	32	0x0000 0100	0x4897 0100	0x4899 0100	0x489B 0100
VIP_CLKC_RST	RW	32	0x0000 0104	0x4897 0104	0x4899 0104	0x489B 0104
VIP_CLKC_DPS	RW	32	0x0000 0108	0x4897 0108	0x4899 0108	0x489B 0108
VIP_CLKC_VIP0DPS	RW	32	0x0000 010C	0x4897 010C	0x4899 010C	0x489B 010C
VIP_CLKC_VIP1DPS	RW	32	0x0000 0110	0x4897 0110	0x4899 0110	0x489B 0110

9.5.2.2 VIP Top Level Register Description

Table 9-72. VIP_CLKC_PID

Address Offset	0x0000 0000		
Physical Address	0x4897 0000 0x4899 0000 0x489B 0000	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	This register follows the format described in PDR3.5		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		RESERVED		FUNC												RTL					MAJOR			CUSTOM		MINOR					

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	The scheme of the register used. This indicates the PDR3.5 Method	R	0x0
29:28	RESERVED		R	0x0
27:16	FUNC	The function of the module being used	R	0x0
15:11	RTL	RTL Release Version The PDR release number of this IP	R	0x0
10:8	MAJOR	ajor Release Number	R	0x0
7:6	CUSTOM	Custom IP	R	0x0
5:0	MINOR	inor Release Number	R	0x0

Table 9-73. Register Call Summary for Register VIP_CLKC_PID

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-74. VIP_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4897 0010 0x4899 0010 0x489B 0010	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	VIP_SYSCONFIG		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										STANDBYMODE		IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state 0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only 0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only 0x2: Same behavior as bit-field value of 0x1. 0x3: Reserved	RW	0x2

Bits	Field Name	Description	Type	Reset
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state 0x0 : Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only 0x1 : No-idle mode: local target never enters idle state. Backup mode, for debug only 0x2 : Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events 0x3 : Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module 'swakeup' output(s) is (are) implemented	RW	0x2
1:0	RESERVED		R	0x0

Table 9-75. Register Call Summary for Register VIP_SYSCONFIG

VIP Functional Description

- VIP Idle Mode: [0]
- VIP StandBy Mode: [1]

VIP Register Manual

- VIP Top Level Register Summary: [2]
- VIP Top Level Register Description: [5]

Table 9-76. VIP_INTC_INTR0_STATUS_RAW0

Address Offset	0x0000 0020		
Physical Address	0x4897 0020 0x4899 0020 0x489B 0020	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Status Raw/Set Register 0. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_RAW	VIP1_PARSER_INT_RAW	RESERVED			VPDMA_INT0_DESCRIPTOR_RAW	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA_INT0_LIST0_COMPLETE_RAW

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21	VIP2_PARSER_INT_RAW	VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_RAW	VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_RAW	VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VPDMA_INT0_LIST0_COMPLETE_RAW	VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 9-77. Register Call Summary for Register VIP_INTC_INTR0_STATUS_RAW0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-78. VIP_INTC_INTR0_STATUS_RAW1

Address Offset	0x0000 0024		
Physical Address	0x4897 0024 0x4899 0024 0x489B 0024	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Status Raw/Set Register 1. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1_CHR_DS_1_UV_ERR_INT_RAW	RESERVED														VPDMA_INT0_CLIENT_RAW	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_RAW	VPDMA_INT0_CHANNEL_GROUP4_RAW	VPDMA_INT0_CHANNEL_GROUP3_RAW	VPDMA_INT0_CHANNEL_GROUP2_RAW	VPDMA_INT0_CHANNEL_GROUP1_RAW	VPDMA_INT0_CHANNEL_GROUP0_RAW

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_RAW	VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
5	VPDMA_INT0_CHANNEL_GROUP5_RAW	VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_RAW	VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_RAW	VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_RAW	VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_RAW	VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_RAW	VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 9-79. Register Call Summary for Register VIP_INTC_INTR0_STATUS_RAW1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-80. VIP_INTC_INTR0_STATUS_ENA0

Address Offset	0x0000 0028		
Physical Address	0x4897 0028 0x4899 0028 0x489B 0028	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Status Enabled/Clear Register 0. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

Bits	Field
31:22	RESERVED
21	VIP2_PARSER_INT_ENA
20	VIP1_PARSER_INT_ENA
19:17	RESERVED
16	VPDMA_INT0_DESCRIPTOR_ENA
15	VPDMA_INT0_LIST7_NOTIFY_ENA
14	VPDMA_INT0_LIST7_COMPLETE_ENA
13	VPDMA_INT0_LIST6_NOTIFY_ENA
12	VPDMA_INT0_LIST6_COMPLETE_ENA
11	VPDMA_INT0_LIST5_NOTIFY_ENA
10	VPDMA_INT0_LIST5_COMPLETE_ENA
9	VPDMA_INT0_LIST4_NOTIFY_ENA
8	VPDMA_INT0_LIST4_COMPLETE_ENA
7	VPDMA_INT0_LIST3_NOTIFY_ENA
6	VPDMA_INT0_LIST3_COMPLETE_ENA
5	VPDMA_INT0_LIST2_NOTIFY_ENA
4	VPDMA_INT0_LIST2_COMPLETE_ENA
3	VPDMA_INT0_LIST1_NOTIFY_ENA
2	VPDMA_INT0_LIST1_COMPLETE_ENA
1	VPDMA_INT0_LIST0_NOTIFY_ENA
0	VPDMA_INT0_LIST0_COMPLETE_ENA

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA	VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
20	VIP1_PARSER_INT_ENA	VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA	VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA	VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA	VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA	VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA	VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA	VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA	VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA	VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA	VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA	VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA	VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA	VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA	VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA	VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA	VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA	VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA	VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-81. Register Call Summary for Register VIP_INTC_INTR0_STATUS_ENA0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-82. VIP_INTC_INTR0_STATUS_ENA1

Address Offset	0x0000 002C		
Physical Address	0x4897 002C 0x4899 002C 0x489B 002C	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Status Enabled/Clear Register 1. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>VIP2_CHR_DS_2_UV_ERR_INT_ENA</td><td>VIP2_CHR_DS_1_UV_ERR_INT_ENA</td><td>VIP1_CHR_DS_2_UV_ERR_INT_ENA</td><td>VIP1_CHR_DS_1_UV_ERR_INT_ENA</td><td colspan="14">RESERVED</td><td>VPDMA_INT0_CLIENT_ENA</td><td>RESERVED</td><td>VPDMA_INT0_CHANNEL_GROUP5_ENA</td><td>VPDMA_INT0_CHANNEL_GROUP4_ENA</td><td>VPDMA_INT0_CHANNEL_GROUP3_ENA</td><td>VPDMA_INT0_CHANNEL_GROUP2_ENA</td><td>VPDMA_INT0_CHANNEL_GROUP1_ENA</td><td>VPDMA_INT0_CHANNEL_GROUP0_ENA</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_ENA	VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA	VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA	VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
3	VPDMA_INT0_CHANNEL_GROUP3_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA	VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA	VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 9-83. Register Call Summary for Register VIP_INTC_INTR0_STATUS_ENA1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-84. VIP_INTC_INTR0_ENA_SET0

Address Offset	0x0000 0030		
Physical Address	0x4897 0030 0x4899 0030 0x489B 0030	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Enable/Set Register 0. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

Bits	Field
31-22	RESERVED
21	VIP2_PARSER_INT_ENA_SET
20	VIP1_PARSER_INT_ENA_SET
19-17	RESERVED
16	VPDMA_INT0_DESCRIPTOR_ENA_SET
15	VPDMA_INT0_LIST7_NOTIFY_ENA_SET
14	VPDMA_INT0_LIST7_COMPLETE_ENA_SET
13	VPDMA_INT0_LIST6_NOTIFY_ENA_SET
12	VPDMA_INT0_LIST6_COMPLETE_ENA_SET
11	VPDMA_INT0_LIST5_NOTIFY_ENA_SET
10	VPDMA_INT0_LIST5_COMPLETE_ENA_SET
9	VPDMA_INT0_LIST4_NOTIFY_ENA_SET
8	VPDMA_INT0_LIST4_COMPLETE_ENA_SET
7	VPDMA_INT0_LIST3_NOTIFY_ENA_SET
6	VPDMA_INT0_LIST3_COMPLETE_ENA_SET
5	VPDMA_INT0_LIST2_NOTIFY_ENA_SET
4	VPDMA_INT0_LIST2_COMPLETE_ENA_SET
3	VPDMA_INT0_LIST1_NOTIFY_ENA_SET
2	VPDMA_INT0_LIST1_COMPLETE_ENA_SET
1	VPDMA_INT0_LIST0_NOTIFY_ENA_SET
0	VPDMA_INT0_LIST0_COMPLETE_ENA_SET

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA_SET	VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_SET	VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
16	VPDMA_INT0_DESCRIPTOR_ENA_SET	VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_SET	VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_SET	VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_SET	VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA_SET	VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA_SET	VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_SET	VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA_SET	VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA_SET	VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA_SET	VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_SET	VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_SET	VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_SET	VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_SET	VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA_SET	VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
1	VPDMA_INT0_LIST0_NOTIFY_ENA_SET	VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_SET	VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-85. Register Call Summary for Register VIP_INTC_INTR0_ENA_SET0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-86. VIP_INTC_INTR0_ENA_SET1

Address Offset	0x0000 0034		
Physical Address	0x4897 0034 0x4899 0034 0x489B 0034	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Enable/Set Register 1. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	RESERVED														VPDMA_INT0_CLIENT_ENA_SET	VPDMA_INT0_CHANNEL_GROUP6_ENA_SET	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_ENA_SET	VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_CHANNEL_GROUP6_ENA_SET	VPDMA INT0 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET	VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-87. Register Call Summary for Register VIP_INTC_INTR0_ENA_SET1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-88. VIP_INTC_INTR0_ENA_CLR0

Address Offset	0x0000 0038		
Physical Address	0x4897 0038 0x4899 0038 0x489B 0038	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Enable/Clear Register 0. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIP2_PARSER_INT_ENA_CLR	VIP1_PARSER_INT_ENA_CLR	RESERVED	VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA_CLR	VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_CLR	VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
8	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR	VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-89. Register Call Summary for Register VIP_INTC_INTR0_ENA_CLR0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-90. VIP_INTC_INTR0_ENA_CLR1

Address Offset	0x0000 003C		
Physical Address	0x4897 003C 0x4899 003C 0x489B 003C	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC INTR0 Interrupt Enable/Clear Register 1. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	RESERVED	VPDMA_INT0_CLIENT_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP6_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 2 UV Error Enable/ Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 1 UV Error Enable/ Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 2 UV Error Enable/ Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 1 UV Error Enable/ Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_ENA_CLR	VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_CHANNEL_GROUP6_ENA_CLR	VPDMA INT0 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR	VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-91. Register Call Summary for Register VIP_INTC_INTR0_ENA_CLR1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-92. VIP_INTC_INTR1_STATUS_RAW0

Address Offset	0x0000 0040		
Physical Address	0x4897 0040 0x4899 0040 0x489B 0040	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Status Raw/Set Register 0. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31:22	21	20	19:17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	VIP2_PARSER_INT_RAW	VIP1_PARSER_INT_RAW	RESERVED	VPDMA_INT1_DESCRIPTOR_RAW	VPDMA_INT1_LIST7_NOTIFY_RAW	VPDMA_INT1_LIST7_COMPLETE_RAW	VPDMA_INT1_LIST6_NOTIFY_RAW	VPDMA_INT1_LIST6_COMPLETE_RAW	VPDMA_INT1_LIST5_NOTIFY_RAW	VPDMA_INT1_LIST5_COMPLETE_RAW	VPDMA_INT1_LIST4_NOTIFY_RAW	VPDMA_INT1_LIST4_COMPLETE_RAW	VPDMA_INT1_LIST3_NOTIFY_RAW	VPDMA_INT1_LIST3_COMPLETE_RAW	VPDMA_INT1_LIST2_NOTIFY_RAW	VPDMA_INT1_LIST2_COMPLETE_RAW	VPDMA_INT1_LIST1_NOTIFY_RAW	VPDMA_INT1_LIST1_COMPLETE_RAW	VPDMA_INT1_LIST0_NOTIFY_RAW	VPDMA_INT1_LIST0_COMPLETE_RAW

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_RAW	VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_RAW	VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_RAW	VPDMA INT1 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
15	VPDMA_INT1_LIST7_NOTIFY_RAW	VPDMA INT1 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
14	VPDMA_INT1_LIST7_COMPLETE_RAW	VPDMA INT1 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
13	VPDMA_INT1_LIST6_NOTIFY_RAW	VPDMA INT1 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
12	VPDMA_INT1_LIST6_COMPLETE_RAW	VPDMA INT1 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_RAW	VPDMA INT1 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_RAW	VPDMA INT1 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_RAW	VPDMA INT1 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_RAW	VPDMA INT1 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
7	VPDMA_INT1_LIST3_NOTIFY_RAW	VPDMA INT1 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_RAW	VPDMA INT1 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_RAW	VPDMA INT1 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_LIST2_COMPLETE_RAW	VPDMA INT1 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_RAW	VPDMA INT1 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_RAW	VPDMA INT1 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_RAW	VPDMA INT1 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_RAW	VPDMA INT1 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 9-93. Register Call Summary for Register VIP_INTC_INTR1_STATUS_RAW0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-94. VIP_INTC_INTR1_STATUS_RAW1

Address Offset	0x0000 0044

Table 9-94. VIP_INTC_INTR1_STATUS_RAW1 (continued)

Physical Address	0x4897 0044 0x4899 0044 0x489B 0044	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Status Raw/Set Register 1. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1_CHR_DS_1_UV_ERR_INT_RAW	RESERVED														VPDMA_INT1_CLIENT_RAW	RESERVED	VPDMA_INT1_CHANNEL_GROUP5_RAW	VPDMA_INT1_CHANNEL_GROUP4_RAW	VPDMA_INT1_CHANNEL_GROUP3_RAW	VPDMA_INT1_CHANNEL_GROUP2_RAW	VPDMA_INT1_CHANNEL_GROUP1_RAW	VPDMA_INT1_CHANNEL_GROUP0_RAW

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_RAW	VPDMA INT1 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT1_CHANNEL_GROUP5_RAW	VPDMA INT1 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_CHANNEL_GROUP4_RAW	VPDMA INT1 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_RAW	VPDMA INT1 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_RAW	VPDMA INT1 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
1	VPDMA_INT1_CHANNEL_GROUP1_RAW	VPDMA INT1 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_RAW	VPDMA INT1 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 9-95. Register Call Summary for Register VIP_INTC_INTR1_STATUS_RAW1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-96. VIP_INTC_INTR1_STATUS_ENA0

Address Offset	0x0000 0048		
Physical Address	0x4897 0048 0x4899 0048 0x489B 0048	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Status Enabled/Clear Register 0. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA	VIP1_PARSER_INT_ENA	RESERVED			VPDMA_INT1_DESCRIPTOR_ENA	VPDMA_INT1_LIST7_NOTIFY_ENA	VPDMA_INT1_LIST7_COMPLETE_ENA	VPDMA_INT1_LIST6_NOTIFY_ENA	VPDMA_INT1_LIST6_COMPLETE_ENA	VPDMA_INT1_LIST5_NOTIFY_ENA	VPDMA_INT1_LIST5_COMPLETE_ENA	VPDMA_INT1_LIST4_NOTIFY_ENA	VPDMA_INT1_LIST4_COMPLETE_ENA	VPDMA_INT1_LIST3_NOTIFY_ENA	VPDMA_INT1_LIST3_COMPLETE_ENA	VPDMA_INT1_LIST2_NOTIFY_ENA	VPDMA_INT1_LIST2_COMPLETE_ENA	VPDMA_INT1_LIST1_NOTIFY_ENA	VPDMA_INT1_LIST1_COMPLETE_ENA	VPDMA_INT1_LIST0_NOTIFY_ENA	VPDMA_INT1_LIST0_COMPLETE_ENA

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA	VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA	VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_ENA	VPDMA INT1 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT1_LIST7_NOTIFY_ENA	VPDMA INT1 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT1_LIST7_COMPLETE_ENA	VPDMA INT1 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
13	VPDMA_INT1_LIST6_NOTIFY_ENA	VPDMA INT1 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT1_LIST6_COMPLETE_ENA	VPDMA INT1 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_ENA	VPDMA INT1 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_ENA	VPDMA INT1 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_ENA	VPDMA INT1 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_ENA	VPDMA INT1 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT1_LIST3_NOTIFY_ENA	VPDMA INT1 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_ENA	VPDMA INT1 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_ENA	VPDMA INT1 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_LIST2_COMPLETE_ENA	VPDMA INT1 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_ENA	VPDMA INT1 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_ENA	VPDMA INT1 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_ENA	VPDMA INT1 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_ENA	VPDMA INT1 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-97. Register Call Summary for Register VIP_INTC_INTR1_STATUS_ENA0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-98. VIP_INTC_INTR1_STATUS_ENA1

Address Offset	0x0000 004C		
Physical Address	0x4897 004C 0x4899 004C 0x489B 004C	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Status Enabled/Clear Register 1. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIP2_CHR_DS_2_UV_ERR_INT_ENA	VIP2_CHR_DS_1_UV_ERR_INT_ENA	VIP1_CHR_DS_2_UV_ERR_INT_ENA	VIP1_CHR_DS_1_UV_ERR_INT_ENA	RESERVED	VPDMA_INT1_CLIENT_ENA	RESERVED	VPDMA_INT1_CHANNEL_GROUP5_ENA	VPDMA_INT1_CHANNEL_GROUP4_ENA	VPDMA_INT1_CHANNEL_GROUP3_ENA	VPDMA_INT1_CHANNEL_GROUP2_ENA	VPDMA_INT1_CHANNEL_GROUP1_ENA	VPDMA_INT1_CHANNEL_GROUP0_ENA

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_ENA	VPDMA INT1 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT1_CHANNEL_GROUP5_ENA	VPDMA INT1 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_CHANNEL_GROUP4_ENA	VPDMA INT1 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_ENA	VPDMA INT1 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_ENA	VPDMA INT1 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_CHANNEL_GROUP1_ENA	VPDMA INT1 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_ENA	VPDMA INT1 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 9-99. Register Call Summary for Register VIP_INTC_INTR1_STATUS_ENA1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-100. VIP_INTC_INTR1_ENA_SET0

Address Offset	0x0000 0050		
Physical Address	0x4897 0050 0x4899 0050 0x489B 0050	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Enable/Set Register 0. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA_SET	VIP1_PARSER_INT_ENA_SET	RESERVED			VPDMA_INT1_DESCRIPTOR_ENA_SET	VPDMA_INT1_LIST7_NOTIFY_ENA_SET	VPDMA_INT1_LIST7_COMPLETE_ENA_SET	VPDMA_INT1_LIST6_NOTIFY_ENA_SET	VPDMA_INT1_LIST6_COMPLETE_ENA_SET	VPDMA_INT1_LIST5_NOTIFY_ENA_SET	VPDMA_INT1_LIST5_COMPLETE_ENA_SET	VPDMA_INT1_LIST4_NOTIFY_ENA_SET	VPDMA_INT1_LIST4_COMPLETE_ENA_SET	VPDMA_INT1_LIST3_NOTIFY_ENA_SET	VPDMA_INT1_LIST3_COMPLETE_ENA_SET	VPDMA_INT1_LIST2_NOTIFY_ENA_SET	VPDMA_INT1_LIST2_COMPLETE_ENA_SET	VPDMA_INT1_LIST1_NOTIFY_ENA_SET	VPDMA_INT1_LIST1_COMPLETE_ENA_SET	VPDMA_INT1_LIST0_NOTIFY_ENA_SET	VPDMA_INT1_LIST0_COMPLETE_ENA_SET

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA_SET	VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_SET	VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_ENA_SET	VPDMA INT1 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT1_LIST7_NOTIFY_ENA_SET	VPDMA INT1 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT1_LIST7_COMPLETE_ENA_SET	VPDMA INT1 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT1_LIST6_NOTIFY_ENA_SET	VPDMA INT1 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
12	VPDMA_INT1_LIST6_COMPLETE_ENA_SET	VPDMA INT1 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_ENA_SET	VPDMA INT1 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_ENA_SET	VPDMA INT1 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_ENA_SET	VPDMA INT1 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_ENA_SET	VPDMA INT1 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT1_LIST3_NOTIFY_ENA_SET	VPDMA INT1 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_ENA_SET	VPDMA INT1 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_ENA_SET	VPDMA INT1 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_LIST2_COMPLETE_ENA_SET	VPDMA INT1 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_ENA_SET	VPDMA INT1 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_ENA_SET	VPDMA INT1 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_ENA_SET	VPDMA INT1 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_ENA_SET	VPDMA INT1 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-101. Register Call Summary for Register VIP_INTC_INTR1_ENA_SET0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-102. VIP_INTC_INTR1_ENA_SET1

Address Offset	0x0000 0054

Table 9-102. VIP_INTC_INTR1_ENA_SET1 (continued)

Physical Address	0x4897 0054 0x4899 0054 0x489B 0054	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Enable/Set Register 1. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

31:26	25	24	23	22	21:8	7	6	5	4	3	2	1	0
RESERVED	VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	RESERVED	VPDMA_INT1_CLIENT_ENA_SET	VPDMA_INT1_CHANNEL_GROUP6_ENA_SET	VPDMA_INT1_CHANNEL_GROUP5_ENA_SET	VPDMA_INT1_CHANNEL_GROUP4_ENA_SET	VPDMA_INT1_CHANNEL_GROUP3_ENA_SET	VPDMA_INT1_CHANNEL_GROUP2_ENA_SET	VPDMA_INT1_CHANNEL_GROUP1_ENA_SET	VPDMA_INT1_CHANNEL_GROUP0_ENA_SET

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_ENA_SET	VPDMA INT1 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_CHANNEL_GROUP6_ENA_SET	VPDMA INT1 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_CHANNEL_GROUP5_ENA_SET	VPDMA INT1 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
4	VPDMA_INT1_CHANNEL_GROUP4_ENA_SET	VPDMA INT1 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_ENA_SET	VPDMA INT1 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_ENA_SET	VPDMA INT1 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_CHANNEL_GROUP1_ENA_SET	VPDMA INT1 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_ENA_SET	VPDMA INT1 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-103. Register Call Summary for Register VIP_INTC_INTR1_ENA_SET1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-104. VIP_INTC_INTR1_ENA_CLR0

Address Offset	0x0000 0058		
Physical Address	0x4897 0058 0x4899 0058 0x489B 0058	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Enable/Clear Register 0. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

31-22	21	20	19-17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	VIP2_PARSER_INT_ENA_CLR	VIP1_PARSER_INT_ENA_CLR	RESERVED	VPDMA_INT1_DESCRIPTOR_ENA_CLR	VPDMA_INT1_LIST7_NOTIFY_ENA_CLR	VPDMA_INT1_LIST7_COMPLETE_ENA_CLR	VPDMA_INT1_LIST6_NOTIFY_ENA_CLR	VPDMA_INT1_LIST6_COMPLETE_ENA_CLR	VPDMA_INT1_LIST5_NOTIFY_ENA_CLR	VPDMA_INT1_LIST5_COMPLETE_ENA_CLR	VPDMA_INT1_LIST4_NOTIFY_ENA_CLR	VPDMA_INT1_LIST4_COMPLETE_ENA_CLR	VPDMA_INT1_LIST3_NOTIFY_ENA_CLR	VPDMA_INT1_LIST3_COMPLETE_ENA_CLR	VPDMA_INT1_LIST2_NOTIFY_ENA_CLR	VPDMA_INT1_LIST2_COMPLETE_ENA_CLR	VPDMA_INT1_LIST1_NOTIFY_ENA_CLR	VPDMA_INT1_LIST1_COMPLETE_ENA_CLR	VPDMA_INT1_LIST0_NOTIFY_ENA_CLR	VPDMA_INT1_LIST0_COMPLETE_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21	VIP2_PARSER_INT_ENA_CLR	VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_CLR	VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_ENA_CLR	VPDMA INT1 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT1_LIST7_NOTIFY_ENA_CLR	VPDMA INT1 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT1_LIST7_COMPLETE_ENA_CLR	VPDMA INT1 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT1_LIST6_NOTIFY_ENA_CLR	VPDMA INT1 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT1_LIST6_COMPLETE_ENA_CLR	VPDMA INT1 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_ENA_CLR	VPDMA INT1 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_ENA_CLR	VPDMA INT1 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_ENA_CLR	VPDMA INT1 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_ENA_CLR	VPDMA INT1 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT1_LIST3_NOTIFY_ENA_CLR	VPDMA INT1 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_ENA_CLR	VPDMA INT1 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_ENA_CLR	VPDMA INT1 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
4	VPDMA_INT1_LIST2_COMPLETE_ENA_CLR	VPDMA INT1 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_ENA_CLR	VPDMA INT1 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_ENA_CLR	VPDMA INT1 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_ENA_CLR	VPDMA INT1 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_ENA_CLR	VPDMA INT1 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-105. Register Call Summary for Register VIP_INTC_INTR1_ENA_CLR0

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-106. VIP_INTC_INTR1_ENA_CLR1

Address Offset	0x0000 005C		
Physical Address	0x4897 005C 0x4899 005C 0x489B 005C	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC intr1 Interrupt Enable/Clear Register 1. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

Bits	Field
31-26	RESERVED
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR
21-8	RESERVED
7	VPDMA_INT1_CLIENT_ENA_CLR
6	VPDMA_INT1_CHANNEL_GROUP6_ENA_CLR
5	VPDMA_INT1_CHANNEL_GROUP5_ENA_CLR
4	VPDMA_INT1_CHANNEL_GROUP4_ENA_CLR
3	VPDMA_INT1_CHANNEL_GROUP3_ENA_CLR
2	VPDMA_INT1_CHANNEL_GROUP2_ENA_CLR
1	VPDMA_INT1_CHANNEL_GROUP1_ENA_CLR
0	VPDMA_INT1_CHANNEL_GROUP0_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_ENA_CLR	VPDMA INT1 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_CHANNEL_GROUP6_ENA_CLR	VPDMA INT1 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_CHANNEL_GROUP5_ENA_CLR	VPDMA INT1 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_CHANNEL_GROUP4_ENA_CLR	VPDMA INT1 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_ENA_CLR	VPDMA INT1 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_ENA_CLR	VPDMA INT1 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_CHANNEL_GROUP1_ENA_CLR	VPDMA INT1 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_ENA_CLR	VPDMA INT1 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 9-107. Register Call Summary for Register VIP_INTC_INTR1_ENA_CLR1

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-108. VIP_INTC_EOI

Address Offset	0x0000 00A0

Table 9-108. VIP_INTC_EOI (continued)

Physical Address	0x4897 00A0 0x4899 00A0 0x489B 00A0	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	INTC EOI Register. This register contains the EOI vector register contents as defined by HL0.8		
Type	RW		

31–0
EOI_VECTOR

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the ipgenericirq for intr output. There are 4 interrupt outputs Write 0x0 : Write to intr0 IP Generic Write 0x1 : Write to intr1 IP Generic Write 0x2 : Write to intr2 IP Generic Write 0x3 : Write to intr3 IP Generic Any other write value is ignored.	RW	0x0

Table 9-109. Register Call Summary for Register VIP_INTC_EOI

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-110. VIP_CLKC_CLKEN

Address Offset	0x0000 0100		
Physical Address	0x4897 0100 0x4899 0100 0x489B 0100	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	CLKC Module Clock Enable Register. This register contains clock enables for the processing paths in the VIP module.		
Type	RW		

31–18	17	16	15–1	0
RESERVED	VIP2_DP_EN	VIP1_DP_EN	RESERVED	VPDMA_EN

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	VIP2_DP_EN	VIP Slice1 Data Path Clock Enable, 1 = Clock Enabled, 0 = Clock Disabled	RW	0x0
16	VIP1_DP_EN	VIP Slice0 Data Path Clock Enable, 1 = Clock Enabled, 0 = Clock Disabled	RW	0x0
15:1	RESERVED		R	0x0
0	VPDMA_EN	VPDMA Clock Enable, 1 = Clock Enabled, 0 = Clock Disabled	RW	0x0

Table 9-111. Register Call Summary for Register VIP_CLKC_CLKEN

VIP Functional Description

- VIP Clocks: [0] [1] [2]

VIP Register Manual

- VIP Top Level Register Summary: [3]

Table 9-112. VIP_CLKC_RST

Address Offset	0x0000 0104

Table 9-112. VIP_CLKC_RST (continued)

Physical Address	0x4897 0104 0x4899 0104 0x489B 0104	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	CLKC Module Reset Register. This register contains resets for the processing paths in the VIP module.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAIN_RST	RESERVED		S1_CHR_DS_1_RST	S0_CHR_DS_1_RST	S1_CHR_DS_0_RST	S0_CHR_DS_0_RST	RESERVED	S1_SC_RST	S0_SC_RST	S1_CSC_RST	S0_CSC_RST	S1_PARSER_RST	S0_PARSER_RST	VIP2_DP_RST	VIP1_DP_RST	RESERVED															VPDMA_RST

Bits	Field Name	Description	Type	Reset
31	MAIN_RST	Reset for all modules in VIP Main Data Path	RW	0x0
30:29	RESERVED	Reserved	R	0x0
28	S1_CHR_DS_1_RST	VIP Slice1 CHRDS1 reset	RW	0x0
27	S0_CHR_DS_1_RST	VIP Slice0 CHRDS1 reset	RW	0x0
26	S1_CHR_DS_0_RST	VIP Slice1 CHRDS0 reset	RW	0x0
25	S0_CHR_DS_0_RST	VIP Slice0 CHRDS0 reset	RW	0x0
24	RESERVED	Reserved	RW	0x0
23	S1_SC_RST	VIP Slice1 SC reset	RW	0x0
22	S0_SC_RST	VIP Slice0 SC reset	RW	0x0
21	S1_CSC_RST	VIP Slice1 CSC reset	RW	0x0
20	S0_CSC_RST	VIP Slice0 CSC reset	RW	0x0
19	S1_PARSER_RST	VIP Slice1 parser reset	RW	0x0
18	S0_PARSER_RST	VIP Slice0 parser reset	RW	0x0
17	VIP2_DP_RST	VIP Slice1 Data Path Reset	RW	0x0
16	VIP1_DP_RST	VIP Slice0 Data Path Reset	RW	0x0
15:1	RESERVED	Reserved	R	0x0000
0	VPDMA_RST	VPDMA Reset	RW	0x0

Table 9-113. Register Call Summary for Register VIP_CLKC_RST

VIP Functional Description

- VIP Software Reset: [0] [1] [2] [3]
- VIP Overflow Detection and Recovery: [4] [5]

VIP Register Manual

- VIP Top Level Register Summary: [6]

Table 9-114. VIP_CLKC_DPS

Address Offset	0x0000 0108		
Physical Address	0x4897 0108 0x4899 0108 0x489B 0108	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	CLKC Main Data Path Select Register. This register selects the various data paths within main portion (non-VIP) of the subsystem		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MAIN_RST	RESERVED	VIP2_DP_RST	VIP1_DP_RST	RESERVED	VPDMA_RST

Bits	Field Name	Description	Type	Reset
31	MAIN_RST	Reset for all modules in DSS Main Data Path	RW	0x0
30:18	RESERVED		R	0x0
17	VIP2_DP_RST	Video Input Port 2 Data Path Reset	RW	0x0
16	VIP1_DP_RST	Video Input Port 1 Data Path Reset	RW	0x0
15:1	RESERVED		R	0x0
0	VPDMA_RST	VPDMA Reset	RW	0x0

Table 9-115. Register Call Summary for Register VIP_CLKC_DPS

VIP Register Manual

- VIP Top Level Register Summary: [0]

Table 9-116. VIP_CLKC_VIP0DPS

Address Offset	0x0000 010C		
Physical Address	0x4897 010C 0x4899 010C 0x489B 010C	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	CLKC Video Input Port 1 Data Path Select Register. This register selects the various data paths within the Video Input Port portion of the subsystem		
Type	RW		

31 30 29 28	27	26	25 24 23 22 21 20 19 18	17	16	15	14 13 12	11 10 9	8	7	6	5 4 3	2 1 0
VIP1_DATAPATH_SELECT	VIP1_TESTPORT_A_SELECT	VIP1_TESTPORT_B_SELECT	RESERVED	VIP1_CHR_DS_2_BYPASS	VIP1_CHR_DS_1_BYPASS	VIP1_MULTI_CHANNEL_SELECT	VIP1_CHR_DS_2_SRC_SELECT	VIP1_CHR_DS_1_SRC_SELECT	VIP1_RGB_OUT_HI_SELECT	VIP1_RGB_OUT_LO_SELECT	VIP1_RGB_SRC_SELECT	VIP1_SC_SRC_SELECT	VIP1_CSC_SRC_SELECT

Bits	Field Name	Description	Type	Reset
31:28	VIP1_DATAPATH_SELECT	VIP1 Datapath Register Field Enable 0000 : All fields written 0001 : Only vip1_csc_src_select written 0010 : Only vip1_sc_src_select written 0011 : Only vip1_rgb_src_select written 0100 : Only vip1_rgb_out_lo_select written 0101 : Only vip1_rgb_out_hi_select written 0110 : Only vip1_chr_ds_1_src_select written 0111 : Only vip1_chr_ds_2_src_select written 1000 : Only vip1_multi_channel_select written 1001 : Only vip1_chr_ds_1_bypass written 1010 : Only vip1_chr_ds_2_bypass written 1011 : Reserved 1100 : Reserved 1101 : Reserved 1110 : Reserved 1111 : Reserved	RW	0x0
27	VIP1_TESTPORT_A_SELECT	0 : Normal mode 1: Test Mode	RW	0x0

Bits	Field Name	Description	Type	Reset
26	VIP1_TESTPORT_B_SELECT	0 : Normal mode 1: Test Mode	RW	0x0
25:18	RESERVED		R	0x0
17	VIP1_CHR_DS_2_BYPASS	Video Input Port 1 Chroma Downsampler 2 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0
16	VIP1_CHR_DS_1_BYPASS	Video Input Port 1 Chroma Downsampler 1 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0
15	VIP1_MULTI_CHANNEL_SELECT	Video Input Port 1 Multi Channel Select 0 : VIP_PARSER A and B channels operate in single channel mode 1 : VIP_PARSER A and B channels directly drive VPDMA (multi-channel case) Multi-Channel means that the A and B sources are from multiple channels and used in a multiplexed stream mode. The VIP Parser extracts the channel ID from each source and outputs this information to VPDMA, and thus to memory. When operating in a multiplexed stream mode, this bit must be set to 1 to enable the channel information to be passed to memory. If this is not set, the channel number (or source number) will be 0 for all streams. If vip1_rgb_out_select = 1, then VIP_PARSER A port is connected to VPDMA	RW	0x0
14:12	VIP1_CHR_DS_2_SRC_SELECT	Video Input Port 1 Chroma Downsampler 2 Source Select 0 : Path Disabled (no input to CHR_DS) 1 : Source from Scaler (SC_M) 2 : Source from Color Space Converter (CSC) 3 : Source from VIP_PARSER A port 4 : Source from VIP_PARSER B port 5 : Source from Transcode (422) 6 : Reserved 7 : Reserved	RW	0x0
11:9	VIP1_CHR_DS_1_SRC_SELECT	Video Input Port 1 Chroma Downsampler 1 Source Select 000 : Path Disabled (no input to CHR_DS) 001 : Source from Scaler (SC_M) 010 : Source from Color Space Converter (CSC) 011 : Source from VIP_PARSER A port 100 : Source from VIP_PARSER B port 101 : Source from Transcode (422) 110 : Reserved 111 : Reserved	RW	0x0
8	VIP1_RGB_OUT_HI_SELECT	Video Input Port 1 HI RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
7	VIP1_RGB_OUT_LO_SELECT	Video Input Port 1 LO RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
6	VIP1_RGB_SRC_SELECT	Video Input Port 1 RGB Output Path Select 0 : Source from Compositor RGB input 1 : Source from CSC	RW	0x0
5:3	VIP1_SC_SRC_SELECT	Video Input Port 1 SC_M Source Select 000 : Path Disabled 001 : Source from Color Space Converter (CSC) 010 : Source from VIP_PARSER A port 011 : Source from VIP_PARSER B port 100 : Source from Transcode (422) 101 : Reserved 110 : Reserved 111 : Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
2:0	VIP1_CSC_SRC_SELECT	Video Input Port 1 CSC Source Select 000 : Path Disabled 001 : Source from VIP_PARSER A (422) port 010 : Source from VIP_PARSER B port 011 : Source from Transcode (422) 100 : Source from VIP_PARSER A (RGB) port 101 : Source from Compositor (RGB) 110 : Reserved 111 : Reserved	RW	0x0

Table 9-117. Register Call Summary for Register VIP_CLKC_VIP0DPS

VIP Functional Description

- VIP Slice Processing Path Overview: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
- VIP Slice Processing Path Multiplexers: [11] [12]

VIP Register Manual

- VIP Top Level Register Summary: [13]

Table 9-118. VIP_CLKC_VIP1DPS

Address Offset	0x0000 0110		
Physical Address	0x4897 0110 0x4899 0110 0x489B 0110	**Instance**	VIP1_top_level VIP2_top_level VIP3_top_level
Description	CLKC Video Input Port 2 Data Path Select Register. This register selects the various data paths within the Video Input Port portion of the subsystem		
Type	RW		

31-28	27	26	25-18	17	16	15	14-12	11-9	8	7	6	5-3	2-0
VIP2_DATAPATH_SELECT	VIP2_TESTPORT_A_SELECT	VIP2_TESTPORT_B_SELECT	RESERVED	VIP2_CHR_DS_2_BYPASS	VIP2_CHR_DS_1_BYPASS	VIP2_MULTI_CHANNEL_SELECT	VIP2_CHR_DS_2_SRC_SELECT	VIP2_CHR_DS_1_SRC_SELECT	VIP2_RGB_OUT_HI_SELECT	VIP2_RGB_OUT_LO_SELECT	VIP2_RGB_SRC_SELECT	VIP2_SC_SRC_SELECT	VIP2_CSC_SRC_SELECT

Bits	Field Name	Description	Type	Reset
31:28	VIP2_DATAPATH_SELECT	VIP2 Datapath Register Field Enable 0000 : All fields written 0001 : Only vip2_csc_src_select written 0010 : Only vip2_sc_src_select written 0011 : Only vip2_rgb_src_select written 0100 : Only vip2_rgb_out_lo_select written 0101 : Only vip2_rgb_out_hi_select written 0110 : Only vip2_chr_ds_1_src_select written 0111 : Only vip2_chr_ds_2_src_select written 1000 : Only vip2_multi_channel_select written 1001 : Only vip2_chr_ds_1_bypass written 1010 : Only vip2_chr_ds_2_bypass written 1011 : Reserved 1100 : Reserved 1101 : Reserved 1110 : Reserved 1111 : Reserved	RW	0x0
27	VIP2_TESTPORT_A_SELECT	0 : Normal mode 1: Test Mode	RW	0x0
26	VIP2_TESTPORT_B_SELECT	0 : Normal mode 1: Test Mode	RW	0x0
25:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17	VIP2_CHR_DS_2_BYPASS	Video Input Port 2 Chroma Downsampler 2 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0
16	VIP2_CHR_DS_1_BYPASS	Video Input Port 2 Chroma Downsampler 1 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0
15	VIP2_MULTI_CHANNEL_SELECT	Video Input Port 2 Multi Channel Select 0 : VIP_PARSER A and B channels operate in single channel mode 1 : VIP_PARSER A and B channels directly drive VPDMA (multi-channel case) Multi-Channel means that the A and B sources are from multiple channels and used in a multiplexed stream mode. The VIP Parser extracts the channel ID from each source and outputs this information to VPDMA, and thus to memory. When operating in a multiplexed stream mode, this bit must be set to 1 to enable the channel information to be passed to memory. If this is not set, the channel number (or source number) will be 0 for all streams. If vip2_rgb_out_select = 1, then VIP_PARSER A port is connected to VPDMA	RW	0x0
14:12	VIP2_CHR_DS_2_SRC_SELECT	Video Input Port 2 Chroma Downsampler 2 Source Select 0 : Path Disabled (no input to CHR_DS) 1 : Source from Scaler (SC_M) 2 : Source from Color Space Converter (CSC) 3 : Source from VIP_PARSER A port 4 : Source from VIP_PARSER B port 5 : Source from Transcode (422) 6 : Reserved 7 : Reserved	RW	0x0
11:9	VIP2_CHR_DS_1_SRC_SELECT	Video Input Port 2 Chroma Downsampler 1 Source Select 000 : Path Disabled (no input to CHR_DS) 001 : Source from Scaler (SC_M) 010 : Source from Color Space Converter (CSC) 011 : Source from VIP_PARSER A port 100 : Source from VIP_PARSER B port 101 : Source from Transcode (422) 110 : Reserved 111 : Reserved	RW	0x0
8	VIP2_RGB_OUT_HI_SELECT	Video Input Port 2 HI RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
7	VIP2_RGB_OUT_LO_SELECT	Video Input Port 2 LO RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
6	VIP2_RGB_SRC_SELECT	Video Input Port 2 RGB Output Path Select 0 : Source from Compositor RGB input 1 : Source from CSC	RW	0x0
5:3	VIP2_SC_SRC_SELECT	Video Input Port 2 SC_M Source Select 000 : Path Disabled 001 : Source from Color Space Converter (CSC) 010 : Source from VIP_PARSER A port 011 : Source from VIP_PARSER B port 100 : Source from Transcode (422) 101 : Reserved 110 : Reserved 111 : Reserved	RW	0x0
2:0	VIP2_CSC_SRC_SELECT	Video Input Port 2 CSC Source Select 000 : Path Disabled 001 : Source from VIP_PARSER A (422) port 010 : Source from VIP_PARSER B port 011 : Source from Transcode (422) 100 : Source from VIP_PARSER A (RGB) port 101 : Source from Compositor (RGB) 110 : Reserved 111 : Reserved	RW	0x0

Table 9-119. Register Call Summary for Register VIP_CLKC_VIP1DPS

VIP Functional Description

- VIP Slice Processing Path Overview: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
- VIP Slice Processing Path Multiplexers: [11] [12]

Table 9-119. Register Call Summary for Register VIP_CLKC_VIP1DPS (continued)

VIP Register Manual

- VIP Top Level Register Summary: [13]

9.5.3 VIP Parser Registers

9.5.3.1 VIP Parser Register Summary

Table 9-120. VIP Parser Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_Slice0_parser Base Address	VIP1_Slice1_parser Base Address	VIP2_Slice0_parser Base Address
VIP_MAIN	RW	32	0x0000 0000	0x4897 5500	0x4897 5A00	0x4899 5500
VIP_PORT_A	RW	32	0x0000 0004	0x4897 5504	0x4897 5A04	0x4899 5504
VIP_XTRA_PORT_A	RW	32	0x0000 0008	0x4897 5508	0x4897 5A08	0x4899 5508
VIP_PORT_B	RW	32	0x0000 000C	0x4897 550C	0x4897 5A0C	0x4899 550C
VIP_XTRA_PORT_B	RW	32	0x0000 0010	0x4897 5510	0x4897 5A10	0x4899 5510
VIP_FIQ_MASK	RW	32	0x0000 0014	0x4897 5514	0x4897 5A14	0x4899 5514
VIP_FIQ_CLEAR	RW	32	0x0000 0018	0x4897 5518	0x4897 5A18	0x4899 5518
VIP_FIQ_STATUS	R	32	0x0000 001C	0x4897 551C	0x4897 5A1C	0x4899 551C
VIP_OUTPUT_PORT_A_SRC_FID	R	32	0x0000 0020	0x4897 5520	0x4897 5A20	0x4899 5520
VIP_OUTPUT_PORT_A_ENC_FID	R	32	0x0000 0024	0x4897 5524	0x4897 5A24	0x4899 5524
VIP_OUTPUT_PORT_B_SRC_FID	R	32	0x0000 0028	0x4897 5528	0x4897 5A28	0x4899 5528
VIP_OUTPUT_PORT_B_ENC_FID	R	32	0x0000 002C	0x4897 552C	0x4897 5A2C	0x4899 552C
VIP_OUTPUT_PORT_A_SRC0_SIZE	R	32	0x0000 0030	0x4897 5530	0x4897 5A30	0x4899 5530
VIP_OUTPUT_PORT_A_SRC1_SIZE	R	32	0x0000 0034	0x4897 5534	0x4897 5A34	0x4899 5534
VIP_OUTPUT_PORT_A_SRC2_SIZE	R	32	0x0000 0038	0x4897 5538	0x4897 5A38	0x4899 5538
VIP_OUTPUT_PORT_A_SRC3_SIZE	R	32	0x0000 003C	0x4897 553C	0x4897 5A3C	0x4899 553C
VIP_OUTPUT_PORT_A_SRC4_SIZE	R	32	0x0000 0040	0x4897 5540	0x4897 5A40	0x4899 5540
VIP_OUTPUT_PORT_A_SRC5_SIZE	R	32	0x0000 0044	0x4897 5544	0x4897 5A44	0x4899 5544
VIP_OUTPUT_PORT_A_SRC6_SIZE	R	32	0x0000 0048	0x4897 5548	0x4897 5A48	0x4899 5548
VIP_OUTPUT_PORT_A_SRC7_SIZE	R	32	0x0000 004C	0x4897 554C	0x4897 5A4C	0x4899 554C
VIP_OUTPUT_PORT_A_SRC8_SIZE	R	32	0x0000 0050	0x4897 5550	0x4897 5A50	0x4899 5550
VIP_OUTPUT_PORT_A_SRC9_SIZE	R	32	0x0000 0054	0x4897 5554	0x4897 5A54	0x4899 5554
VIP_OUTPUT_PORT_A_SRC10_SIZE	R	32	0x0000 0058	0x4897 5558	0x4897 5A58	0x4899 5558
VIP_OUTPUT_PORT_A_SRC11_SIZE	R	32	0x0000 005C	0x4897 555C	0x4897 5A5C	0x4899 555C
VIP_OUTPUT_PORT_A_SRC12_SIZE	R	32	0x0000 0060	0x4897 5560	0x4897 5A60	0x4899 5560
VIP_OUTPUT_PORT_A_SRC13_SIZE	R	32	0x0000 0064	0x4897 5564	0x4897 5A64	0x4899 5564
VIP_OUTPUT_PORT_A_SRC14_SIZE	R	32	0x0000 0068	0x4897 5568	0x4897 5A68	0x4899 5568
VIP_OUTPUT_PORT_A_SRC15_SIZE	R	32	0x0000 006C	0x4897 556C	0x4897 5A6C	0x4899 556C
VIP_OUTPUT_PORT_B_SRC0_SIZE	R	32	0x0000 0070	0x4897 5570	0x4897 5A70	0x4899 5570
VIP_OUTPUT_PORT_B_SRC1_SIZE	R	32	0x0000 0074	0x4897 5574	0x4897 5A74	0x4899 5574
VIP_OUTPUT_PORT_B_SRC2_SIZE	R	32	0x0000 0078	0x4897 5578	0x4897 5A78	0x4899 5578
VIP_OUTPUT_PORT_B_SRC3_SIZE	R	32	0x0000 007C	0x4897 557C	0x4897 5A7C	0x4899 557C
VIP_OUTPUT_PORT_B_SRC4_SIZE	R	32	0x0000 0080	0x4897 5580	0x4897 5A80	0x4899 5580
VIP_OUTPUT_PORT_B_SRC5_SIZE	R	32	0x0000 0084	0x4897 5584	0x4897 5A84	0x4899 5584
VIP_OUTPUT_PORT_B_SRC6_SIZE	R	32	0x0000 0088	0x4897 5588	0x4897 5A88	0x4899 5588
VIP_OUTPUT_PORT_B_SRC7_SIZE	R	32	0x0000 008C	0x4897 558C	0x4897 5A8C	0x4899 558C
VIP_OUTPUT_PORT_B_SRC8_SIZE	R	32	0x0000 0090	0x4897 5590	0x4897 5A90	0x4899 5590
VIP_OUTPUT_PORT_B_SRC9_SIZE	R	32	0x0000 0094	0x4897 5594	0x4897 5A94	0x4899 5594
VIP_OUTPUT_PORT_B_SRC10_SIZE	R	32	0x0000 0098	0x4897 5598	0x4897 5A98	0x4899 5598
VIP_OUTPUT_PORT_B_SRC11_SIZE	R	32	0x0000 009C	0x4897 559C	0x4897 5A9C	0x4899 559C
VIP_OUTPUT_PORT_B_SRC12_SIZE	R	32	0x0000 00A0	0x4897 55A0	0x4897 5AA0	0x4899 55A0
VIP_OUTPUT_PORT_B_SRC13_SIZE	R	32	0x0000 00A4	0x4897 55A4	0x4897 5AA4	0x4899 55A4
VIP_OUTPUT_PORT_B_SRC14_SIZE	R	32	0x0000 00A8	0x4897 55A8	0x4897 5AA8	0x4899 55A8

Table 9-120. VIP Parser Registers Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_Slice0_parser Base Address	VIP1_Slice1_parser Base Address	VIP2_Slice0_parser Base Address
VIP_OUTPUT_PORT_B_SRC15_SIZE	R	32	0x0000 00AC	0x4897 55AC	0x4897 5AAC	0x4899 55AC
VIP_PORT_A_VDET_VEC	R	32	0x0000 00B0	0x4897 55B0	0x4897 5AB0	0x4899 55B0
VIP_PORT_B_VDET_VEC	R	32	0x0000 00B4	0x4897 55B4	0x4897 5AB4	0x4899 55B4
VIP_ANC_CROP_HORZ_PORT_A	RW	32	0x0000 00B8	0x4897 55B8	0x4897 5AB8	0x4899 55B8
VIP_ANC_CROP_VERT_PORT_A	RW	32	0x0000 00BC	0x4897 55BC	0x4897 5ABC	0x4899 55BC
VIP_CROP_HORZ_PORT_A	RW	32	0x0000 00C0	0x4897 55C0	0x4897 5AC0	0x4899 55C0
VIP_CROP_VERT_PORT_A	RW	32	0x0000 00C4	0x4897 55C4	0x4897 5AC4	0x4899 55C4
VIP_ANC_VIP_CROP_HORZ_PORT_B	RW	32	0x0000 00C8	0x4897 55C8	0x4897 5AC8	0x4899 55C8
VIP_ANC_VIP_CROP_VERT_PORT_B	RW	32	0x0000 00CC	0x4897 55CC	0x4897 5ACC	0x4899 55CC
VIP_CROP_HORZ_PORT_B	RW	32	0x0000 00D0	0x4897 55D0	0x4897 5AD0	0x4899 55D0
VIP_CROP_VERT_PORT_B	RW	32	0x0000 00D4	0x4897 55D4	0x4897 5AD4	0x4899 55D4
VIP_XTRA6_PORT_A	RW	32	0x0000 00D8	0x4897 55D8	0x4897 5AD8	0x4899 55D8
VIP_XTRA7_PORT_B	RW	32	0x0000 00DC	0x4897 55DC	0x4897 5ADC	0x4899 55DC
VIP_XTRA8_PORT_A	RW	32	0x0000 00E0	0x4897 55E0	0x4897 5AE0	0x4899 55E0
VIP_XTRA9_PORT_B	RW	32	0x0000 00E4	0x4897 55E4	0x4897 5AE4	0x4899 55E4

Table 9-121. VIP Parser Registers Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	VIP2_Slice1_parser Base Address	VIP3_Slice0_parser Base Address	VIP3_Slice1_parser Base Address
VIP_MAIN	RW	32	0x0000 0000	0x4899 5A00	0x489B 5500	0x489B 5A00
VIP_PORT_A	RW	32	0x0000 0004	0x4899 5A04	0x489B 5504	0x489B 5A04
VIP_XTRA_PORT_A	RW	32	0x0000 0008	0x4899 5A08	0x489B 5508	0x489B 5A08
VIP_PORT_B	RW	32	0x0000 000C	0x4899 5A0C	0x489B 550C	0x489B 5A0C
VIP_XTRA_PORT_B	RW	32	0x0000 0010	0x4899 5A10	0x489B 5510	0x489B 5A10
VIP_FIQ_MASK	RW	32	0x0000 0014	0x4899 5A14	0x489B 5514	0x489B 5A14
VIP_FIQ_CLEAR	RW	32	0x0000 0018	0x4899 5A18	0x489B 5518	0x489B 5A18
VIP_FIQ_STATUS	R	32	0x0000 001C	0x4899 5A1C	0x489B 551C	0x489B 5A1C
VIP_OUTPUT_PORT_A_SRC_FID	R	32	0x0000 0020	0x4899 5A20	0x489B 5520	0x489B 5A20
VIP_OUTPUT_PORT_A_ENC_FID	R	32	0x0000 0024	0x4899 5A24	0x489B 5524	0x489B 5A24
VIP_OUTPUT_PORT_B_SRC_FID	R	32	0x0000 0028	0x4899 5A28	0x489B 5528	0x489B 5A28
VIP_OUTPUT_PORT_B_ENC_FID	R	32	0x0000 002C	0x4899 5A2C	0x489B 552C	0x489B 5A2C
VIP_OUTPUT_PORT_A_SRC0_SIZE	R	32	0x0000 0030	0x4899 5A30	0x489B 5530	0x489B 5A30
VIP_OUTPUT_PORT_A_SRC1_SIZE	R	32	0x0000 0034	0x4899 5A34	0x489B 5534	0x489B 5A34
VIP_OUTPUT_PORT_A_SRC2_SIZE	R	32	0x0000 0038	0x4899 5A38	0x489B 5538	0x489B 5A38
VIP_OUTPUT_PORT_A_SRC3_SIZE	R	32	0x0000 003C	0x4899 5A3C	0x489B 553C	0x489B 5A3C
VIP_OUTPUT_PORT_A_SRC4_SIZE	R	32	0x0000 0040	0x4899 5A40	0x489B 5540	0x489B 5A40
VIP_OUTPUT_PORT_A_SRC5_SIZE	R	32	0x0000 0044	0x4899 5A44	0x489B 5544	0x489B 5A44
VIP_OUTPUT_PORT_A_SRC6_SIZE	R	32	0x0000 0048	0x4899 5A48	0x489B 5548	0x489B 5A48
VIP_OUTPUT_PORT_A_SRC7_SIZE	R	32	0x0000 004C	0x4899 5A4C	0x489B 554C	0x489B 5A4C
VIP_OUTPUT_PORT_A_SRC8_SIZE	R	32	0x0000 0050	0x4899 5A50	0x489B 5550	0x489B 5A50
VIP_OUTPUT_PORT_A_SRC9_SIZE	R	32	0x0000 0054	0x4899 5A54	0x489B 5554	0x489B 5A54
VIP_OUTPUT_PORT_A_SRC10_SIZE	R	32	0x0000 0058	0x4899 5A58	0x489B 5558	0x489B 5A58
VIP_OUTPUT_PORT_A_SRC11_SIZE	R	32	0x0000 005C	0x4899 5A5C	0x489B 555C	0x489B 5A5C
VIP_OUTPUT_PORT_A_SRC12_SIZE	R	32	0x0000 0060	0x4899 5A60	0x489B 5560	0x489B 5A60
VIP_OUTPUT_PORT_A_SRC13_SIZE	R	32	0x0000 0064	0x4899 5A64	0x489B 5564	0x489B 5A64
VIP_OUTPUT_PORT_A_SRC14_SIZE	R	32	0x0000 0068	0x4899 5A68	0x489B 5568	0x489B 5A68
VIP_OUTPUT_PORT_A_SRC15_SIZE	R	32	0x0000 006C	0x4899 5A6C	0x489B 556C	0x489B 5A6C
VIP_OUTPUT_PORT_B_SRC0_SIZE	R	32	0x0000 0070	0x4899 5A70	0x489B 5570	0x489B 5A70
VIP_OUTPUT_PORT_B_SRC1_SIZE	R	32	0x0000 0074	0x4899 5A74	0x489B 5574	0x489B 5A74
VIP_OUTPUT_PORT_B_SRC2_SIZE	R	32	0x0000 0078	0x4899 5A78	0x489B 5578	0x489B 5A78
VIP_OUTPUT_PORT_B_SRC3_SIZE	R	32	0x0000 007C	0x4899 5A7C	0x489B 557C	0x489B 5A7C

Table 9-121. VIP Parser Registers Mapping Summary 2 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP2_Slice1_parser Base Address	VIP3_Slice0_parser Base Address	VIP3_Slice1_parser Base Address
VIP_OUTPUT_PORT_B_SRC4_SIZE	R	32	0x0000 0080	0x4899 5A80	0x489B 5580	0x489B 5A80
VIP_OUTPUT_PORT_B_SRC5_SIZE	R	32	0x0000 0084	0x4899 5A84	0x489B 5584	0x489B 5A84
VIP_OUTPUT_PORT_B_SRC6_SIZE	R	32	0x0000 0088	0x4899 5A88	0x489B 5588	0x489B 5A88
VIP_OUTPUT_PORT_B_SRC7_SIZE	R	32	0x0000 008C	0x4899 5A8C	0x489B 558C	0x489B 5A8C
VIP_OUTPUT_PORT_B_SRC8_SIZE	R	32	0x0000 0090	0x4899 5A90	0x489B 5590	0x489B 5A90
VIP_OUTPUT_PORT_B_SRC9_SIZE	R	32	0x0000 0094	0x4899 5A94	0x489B 5594	0x489B 5A94
VIP_OUTPUT_PORT_B_SRC10_SIZE	R	32	0x0000 0098	0x4899 5A98	0x489B 5598	0x489B 5A98
VIP_OUTPUT_PORT_B_SRC11_SIZE	R	32	0x0000 009C	0x4899 5A9C	0x489B 559C	0x489B 5A9C
VIP_OUTPUT_PORT_B_SRC12_SIZE	R	32	0x0000 00A0	0x4899 5AA0	0x489B 55A0	0x489B 5AA0
VIP_OUTPUT_PORT_B_SRC13_SIZE	R	32	0x0000 00A4	0x4899 5AA4	0x489B 55A4	0x489B 5AA4
VIP_OUTPUT_PORT_B_SRC14_SIZE	R	32	0x0000 00A8	0x4899 5AA8	0x489B 55A8	0x489B 5AA8
VIP_OUTPUT_PORT_B_SRC15_SIZE	R	32	0x0000 00AC	0x4899 5AAC	0x489B 55AC	0x489B 5AAC
VIP_PORT_A_VDET_VEC	R	32	0x0000 00B0	0x4899 5AB0	0x489B 55B0	0x489B 5AB0
VIP_PORT_B_VDET_VEC	R	32	0x0000 00B4	0x4899 5AB4	0x489B 55B4	0x489B 5AB4
VIP_ANC_CROP_HORZ_PORT_A	RW	32	0x0000 00B8	0x4899 5AB8	0x489B 55B8	0x489B 5AB8
VIP_ANC_CROP_VERT_PORT_A	RW	32	0x0000 00BC	0x4899 5ABC	0x489B 55BC	0x489B 5ABC
VIP_CROP_HORZ_PORT_A	RW	32	0x0000 00C0	0x4899 5AC0	0x489B 55C0	0x489B 5AC0
VIP_CROP_VERT_PORT_A	RW	32	0x0000 00C4	0x4899 5AC4	0x489B 55C4	0x489B 5AC4
VIP_ANC_VIP_CROP_HORZ_PORT_B	RW	32	0x0000 00C8	0x4899 5AC8	0x489B 55C8	0x489B 5AC8
VIP_ANC_VIP_CROP_VERT_PORT_B	RW	32	0x0000 00CC	0x4899 5ACC	0x489B 55CC	0x489B 5ACC
VIP_CROP_HORZ_PORT_B	RW	32	0x0000 00D0	0x4899 5AD0	0x489B 55D0	0x489B 5AD0
VIP_CROP_VERT_PORT_B	RW	32	0x0000 00D4	0x4899 5AD4	0x489B 55D4	0x489B 5AD4
VIP_XTRA6_PORT_A	RW	32	0x0000 00D8	0x4899 5AD8	0x489B 55D8	0x489B 5AD8
VIP_XTRA7_PORT_B	RW	32	0x0000 00DC	0x4899 5ADC	0x489B 55DC	0x489B 5ADC
VIP_XTRA8_PORT_A	RW	32	0x0000 00E0	0x4899 5AE0	0x489B 55E0	0x489B 5AE0
VIP_XTRA9_PORT_B	RW	32	0x0000 00E4	0x4899 5AE4	0x489B 55E4	0x489B 5AE4

9.5.3.2 VIP Parser Register Description

Table 9-122. VIP_MAIN

Address Offset	0x0000 0000		
Physical Address	0x4897 5500 0x4897 5A00 0x4899 5500 0x4899 5A00 0x489B 5500 0x489B 5A00	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Main Configuration for VIP Parser		
Type	RW		

31:6	5	4	3:2	1:0
RESERVED	CLIP_ACTIVE	CLIP_BLNK	RESERVED	DATA_INTERFACE_MODE

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	CLIP_ACTIVE	Discrete Sync Only; 0 = Do not clip active pixels; 1 = Clip Active Pixels as follows: 0xFF -> 0xFE, 0x00 -> 0x01	RW	0x0

Bits	Field Name	Description	Type	Reset
4	CLIP_BLNK	Discrete Sync Only; 0 = Do not clip Blanking Data; 1 = Clip Blanking Data as follows: 0xFF -> 0xFE, 0x00 -> 0x01	RW	0x0
3:2	RESERVED		R	0x0
1:0	DATA_INTERFACE_MODE	00 = 24b Port A data interface. 01 = 16b Port A data interface. 10 = 8b Port A data interfaces. 11 = Undefined. Port B is always an 8b data interface.	RW	0x0

Table 9-123. VIP_PORT_A

Address Offset	0x0000 0004		
Physical Address	0x4897 5504 0x4897 5A04 0x4899 5504 0x4899 5A04 0x489B 5504 0x489B 5A04	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Configuration for Input Port A		
Type	RW		

31	30	29 28 27 26 25 24	23	22	21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5 4	3 2 1 0
ANALYZER_FVH_ERR_CORRECTION_ENABLE	ANALYZER_2X4X_SRCNUM_POS	FID_SKEW_POSTCOUNT	SW_RESET	DISCRETE_BASIC_MODE	FID_SKEW_PRECOUNT	USE_ACTVID_HSYNC_N	FID_DETECT_MODE	ACTVID_POLARITY	VSYNC_POLARITY	HSYNC_POLARITY	PIXCLK_EDGE_POLARITY	FID_POLARITY	ENABLE	CLR_ASYNC_FIFO_RD	CLR_ASYNC_FIFO_WR	CTRL_CHAN_SEL	SYNC_TYPE

Bits	Field Name	Description	Type	Reset
31	ANALYZER_FVH_ERR_CORRECTION_ENABLE	Embedded Sync Only 0 = Ignore the protection bits in the XV (fvh) codeword header. This setting is typically desired. 1 = Use the protection bits in an attempt to do error correction for the fvh control bits.	RW	0x0
30	ANALYZER_2X4X_SRCNUM_POS	Embedded Sync Only 0 = For 2x/4x mux mode, srcnum is in the least significant nibble of the XV/fvh codeword (srcnum replaces the protection bits) 1 = For 2x/4x mux mode, srcnum is in the least significant nibble of a horizontal blanking pixel value	RW	0x0
29:24	FID_SKEW_POSTCOUNT	Discrete Sync Only post count value when using vsync skew in FID determination	RW	0x0
23	SW_RESET	0 = Normal 1 = Reset Port A logic. Must be set to ?0? again by the software for the module to function.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	DISCRETE_BASIC_MODE	This register is valid for Discrete Sync mode only. 0 = Normal Discrete Mode. Hsync Style Capture operates as follows: - Captures line starting from HSYNC inactive to active condition. - VSYNC determined by sync window. - FID can be determined by VSYNC skew or captured from pin at first pixel in first line. ACTVID style capture works as follows: - Captures line during contiguous ACTVID envelope. - VSYNC is captured at the first pixel in each line. - FID is captured on first pixel of ACTVID window. 1 = Basic Discrete Mode. When using hsync with Hsync Style Capture operates as follows: - The last line of active video ends on the pixel clock cycle where VSYNC transitions from inactive to active. - FID pin value is captured on this cycle and is used for the next field. - FID detection by VSYNC skew is not allowed. ACTVID style capture works as follows: - VSYNC is expected to transition from inactive to active between ACTVID window. - This VSYNC transition allows the next line in an ACTVID envelope to be sent to a new VPDMA buffer. - FID value is determined by the FID pin value on the cycle where VSYNC transitions from inactive to active. In basic discrete mode, there is no Vertical Ancillary Data. Therefore, VPDMA descriptors should not use Ancillary Data channels.	RW	0x0
21:16	FID_SKEW_PRECOUNT	Discrete Sync Only pre count value when using vsync skew in FID determination	RW	0x0
15	USE_ACTVID_HSYNC_N	Discrete Sync Only 0 = Use HSYNC style line capture 1 = Use ACTVID style line capture	RW	0x0
14	FID_DETECT_MODE	Discrete Sync Only 0 = Take FID from pin 1 = FID is determined by VSYNC skew	RW	0x0
13	ACTVID_POLARITY	Discrete Sync Only 0 = ACTVID is active low 1 = ACTVID is active high	RW	0x0
12	VSYNC_POLARITY	Discrete Sync Only 0 = VSYNC is active low 1 = VSYNC is active high	RW	0x0
11	HSYNC_POLARITY	Discrete Sync Only 0 = HSYNC is active low 1 = HSYNC is active high	RW	0x0
10	PIXCLK_EDGE_POLARITY	0 = Rising Edge is active PIXCLK edge 1 = Falling Edge is active PIXCLK edge	RW	0x0
9	FID_POLARITY	0 = Keep FID as found 1 = Invert Determined Value of FID	RW	0x0
8	ENABLE	0 = Disable Port 1 = Enable Port	RW	0x0
7	CLR_ASYNC_FIFO_RD	0 = Normal 1 = Clear Async FIFO Read Logic	RW	0x0
6	CLR_ASYNC_FIFO_WR	0 = Normal 1 = Clear Async FIFO Write Logic	RW	0x0
5:4	CTRL_CHAN_SEL	Embedded Sync Only In 8b mode.. there is only one channel on data[7:0]. In 16b mode.. there are two channels. The Luma Channel is on data[15:8]. The Chroma Channel is on data[7:0]. In 24b mode.. there are three channels. The R channel is on data[23:16].. the G channel is on [15:8]. and the B channel is on data[7:0]. 00 = Use data[7:0] to extract control codes. 01 = Use data[15:8] to extract control codes. 10 = Use data[23:16] to extract control codes. 11 = Undefined In 16b and 24b modes.. this register is also used to select the channel from which Ancillary Data is extracted. The Ancillary Data channel must be the same as the control code channel. For 8b mode.. the anc_chan_sel_8b register is used to select the Luma or Chroma channel from which Ancillary Data is taken.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	SYNC_TYPE	0000 = embedded sync single 4:2:2 YUV stream 0001 = embedded sync 2x multiplexed 4:2:2 YUV stream 0010 = embedded sync 4x multiplexed 4:2:2 YUV stream 0011 = embedded sync line multiplexed 4:2:2 YUV stream 0100 = discrete sync single 4:2:2 YUV stream 0101 = embedded sync single RGB stream or single 444 YUV stream 0110 = reserved 0111 = reserved 1000 = reserved 1001 = reserved 1010 = discrete sync single 24b RGB stream	RW	0x0

Table 9-124. VIP_XTRA_PORT_A

Address Offset	0x0000 0008		
Physical Address	0x4897 5508 0x4897 5A08 0x4899 5508 0x4899 5A08 0x489B 5508 0x489B 5A08	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	ore Configuration for Input Port A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	REPACK_SEL			SRC0_NUMPIX												RESERVED	ANC_CHAN_SEL_8B		RESERVED	SRC0_NUMLINES											

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	REPACK_SEL	000 = Straight Through 001 = Cross Swap 010 = Left Center Swap 011 = Center Right Swap 100 = Right Rotate 101 = Left Rotate 110 = RAW16 to RGB565 Mapping 111 = RAW12 Swap	RW	0x0
27:16	SRC0_NUMPIX	Number of expected pixels on Source Number 0. The Port_a_src0_size interrupt will trigger if a line is encountered that differs from this pixelcount.	RW	0x0
15	RESERVED		R	0x0
14:13	ANC_CHAN_SEL_8B	In 8b mode, Vertically Ancillary Data typically resides in the Luma sites. This bit allows vertical ancillary data to be extracted from the chroma sites instead . 00 = Extract 8b Mode Vertical Ancillary Data from Luma Sites 01 = Extract 8b Mode Vertical Ancillary Data from Chroma Sites 10, 11 = Extract every single sample of vertical ancillary data. The output line is twice as wide as the other modes. For 16b and 24b inputs, ctrl_chan_sel is used to select which channel is used as a source for vertical ancillary data.	RW	0x0
12	RESERVED		R	0x0
11:0	SRC0_NUMLINES	Number of expected lines on Source Number 0. The Port_a_src0_size interrupt will trigger if a field/frame is encountered that differs from this linecount.	RW	0x0

Table 9-125. VIP_PORT_B

Address Offset	0x0000 000C

Table 9-125. VIP_PORT_B (continued)

Physical Address	0x4897 550C 0x4897 5A0C 0x4899 550C 0x4899 5A0C 0x489B 550C 0x489B 5A0C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Configuration for Input Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANALYZER_FVH_ERR_CORRECTION_ENABLE	ANALYZER_2X4X_SRCNUM_POS	FID_SKEW_POSTCOUNT						SW_RESET	DISCRETE_BASIC_MODE	FID_SKEW_PRECOUNT						USE_ACTVID_HSYNC_N	FID_DETECT_MODE	ACTVID_POLARITY	VSYNC_POLARITY	HSYNC_POLARITY	PIXCLK_EDGE_POLARITY	FID_POLARITY	ENABLE	CLR_ASYNC_FIFO_RD	CLR_ASYNC_FIFO_WR	CTRL_CHAN_SEL		SYNC_TYPE			

Bits	Field Name	Description	Type	Reset
31	ANALYZER_FVH_ERR_CORRECTION_ENABLE	Embedded Sync Only 0 = Ignore the protection bits in the XV (fvh) codeword header. This setting is typically desired. 1 = Use the protection bits in an attempt to do error correction for the fvh control bits.	RW	0x0
30	ANALYZER_2X4X_SRCNUM_POS	Embedded Sync Only 0 = For 2x/4x mux mode, srcnum is in the least significant nibble of the XV/fvh codeword (srcnum replaces the protection bits) 1 = For 2x/4x mux mode, srcnum is in the least significant nibble of a horizontal blanking pixel value	RW	0x0
29:24	FID_SKEW_POSTCOUNT	Discrete Sync Only post count value when using vsync skew in FID determination	RW	0x0
23	SW_RESET	0 = Normal 1 = Reset Port B logic. Must be set to ?0? again by the software for the module to function.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	DISCRETE_BASIC_MODE	This register is valid for Discrete Sync mode only. 0 = Normal Discrete Mode. Hsync Style Capture operates as follows: - Captures line starting from HSYNC inactive to active condition. - VSYNC determined by sync window. - FID can be determined by VSYNC skew or captured from pin at first pixel in first line. ACTVID style capture works as follows: - Captures line during contiguous ACTVID envelope. - VSYNC is captured at the first pixel in each line. - FID is captured on first pixel of ACTVID window. 1 = Basic Discrete Mode. When using hsync with Hsync Style Capture operates as follows: - The last line of active video ends on the pixel clock cycle where VSYNC transitions from inactive to active. - FID pin value is captured on this cycle and is used for the next field. - FID detection by VSYNC skew is not allowed. ACTVID style capture works as follows: - VSYNC is expected to transition from inactive to active between ACTVID window. - This VSYNC transition allows the next line in an ACTVID envelope to be sent to a new VPDMA buffer. - FID value is determined by the FID pin value on the cycle where VSYNC transitions from inactive to active. In basic discrete mode, there is no Vertical Ancillary Data. Therefore, VPDMA descriptors should not use Ancillary Data channels.	RW	0x0
21:16	FID_SKEW_PRECOUNT	Discrete Sync Only pre count value when using vsync skew in FID determination	RW	0x0
15	USE_ACTVID_HSYNC_N	Discrete Sync Only 0 = Use HSYNC style line capture 1 = Use ACTVID style line capture	RW	0x0
14	FID_DETECT_MODE	Discrete Sync Only 0 = Take FID from pin 1 = FID is determined by VSYNC skew	RW	0x0
13	ACTVID_POLARITY	Discrete Sync Only 0 = ACTVID is active low 1 = ACTVID is active high	RW	0x0
12	VSYNC_POLARITY	Discrete Sync Only 0 = VSYNC is active low 1 = VSYNC is active high	RW	0x0
11	HSYNC_POLARITY	Discrete Sync Only 0 = HSYNC is active low 1 = HSYNC is active high	RW	0x0
10	PIXCLK_EDGE_POLARITY	0 = Rising Edge is active PIXCLK edge 1 = Falling Edge is active PIXCLK edge	RW	0x0
9	FID_POLARITY	0 = Keep FID as found 1 = Invert Determined Value of FID	RW	0x0
8	ENABLE	0 = Disable 1 = Enable	RW	0x0
7	CLR_ASYNC_FIFO_RD	0 = Normal 1 = Clear Async FIFO Read Logic	RW	0x0
6	CLR_ASYNC_FIFO_WR	0 = Normal 1 = Clear Async FIFO Write Logic	RW	0x0
5:4	CTRL_CHAN_SEL	PORT B supports on 8b mode. Always write 0 to this field. The anc_chan_sel_8b register is used to select the Luma or Chroma channel from which Ancillary Data is taken.	RW	0x0
3:0	SYNC_TYPE	0000 = embedded sync single YUV stream 0001 = embedded sync 2x multiplexed YUV stream 0010 = embedded sync 4x multiplexed YUV stream 0011 = embedded sync line multiplexed YUV stream 0100 = discrete sync single YUV stream 0101 = embedded sync single RGB stream 0110 = reserved 0111 = reserved 1000 = reserved 1001 = reserved 1010 = discrete sync single 24b RGB stream	RW	0x0

Table 9-126. VIP_XTRA_PORT_B

Address Offset	0x0000 0010

Table 9-126. VIP_XTRA_PORT_B (continued)

Physical Address	0x4897 5510 0x4897 5A10 0x4899 5510 0x4899 5A10 0x489B 5510 0x489B 5A10	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	ore Configuration for Input Port B		
Type	RW		

31:28	27:16	15	14:13	12	11:0
RESERVED	SRC0_NUMPIX	RESERVED	ANC_CHAN_SEL_8B	RESERVED	SRC0_NUMLINES

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	SRC0_NUMPIX	Number of expected pixels on Source Number 0. The Port_b_src0_size interrupt will trigger if a line is encountered that differs from this pixelcount.	RW	0x0
15	RESERVED		R	0x0
14:13	ANC_CHAN_SEL_8B	In 8b mode, Vertically Ancillary Data typically resides in the Luma sites. This bit allows vertical ancillary data to be extracted from the chroma sites instead . 00 = Extract 8b Mode Vertical Ancillary Data from Luma Sites 01 = Extract 8b Mode Vertical Ancillary Data from Chroma Sites 10, 11 = Extract every single sample of vertical ancillary data. The output line is twice as wide as the other modes. For 16b and 24b inputs, ctrl_chan_sel is used to select which channel is used as a source for vertical ancillary data.	RW	0x0
12	RESERVED		R	0x0
11:0	SRC0_NUMLINES	Number of expected lines on Source Number 0. The Port_b_src0_size interrupt will trigger if a field/frame is encountered that differs from this linecount.	RW	0x0

Table 9-127. VIP_FIQ_MASK

Address Offset	0x0000 0014		
Physical Address	0x4897 5514 0x4897 5A14 0x4899 5514 0x4899 5A14 0x489B 5514 0x489B 5A14	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	ask Bits for ARM FIQs		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PORT_B_CFG_DISABLE_COMPLETE_MASK	PORT_A_CFG_DISABLE_COMPLETE_MASK	PORT_B_ANC_PROTOCOL_VIOLATION_MASK	PORT_B_YUV_PROTOCOL_VIOLATION_MASK	PORT_A_ANC_PROTOCOL_VIOLATION_MASK	PORT_A_YUV_PROTOCOL_VIOLATION_MASK	PORT_B_SRC0_SIZE	PORT_A_SRC0_SIZE	PORT_B_DISCONN	PORT_B_CONN	PORT_A_DISCONN	PORT_A_CONN	OUTPUT_FIFO_PRTB_ANC_OF	RESERVED	OUTPUT_FIFO_PRTB_YUV_OF	OUTPUT_FIFO_PRTA_ANC_OF	RESERVED	OUTPUT_FIFO_PRTA_YUV_OF	ASYNC_FIFO_PRTB_OF	ASYNC_FIFO_PRTA_OF	PRTB_VDET_MASK	PRTA_VDET_MASK

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	PORT_B_CFG_DISABLE_COMPLETE_MASK	Port B Cfg Disable Complete Mask	RW	0x0
20	PORT_A_CFG_DISABLE_COMPLETE_MASK	Port A Cfg Disable Complete Mask	RW	0x0
19	PORT_B_ANC_PROTOCOL_VIOLATION_MASK	Port B ANC VPI Protocol Violation Mask	RW	0x0
18	PORT_B_YUV_PROTOCOL_VIOLATION_MASK	Port B YUV VPI Protocol Violation Mask	RW	0x0
17	PORT_A_ANC_PROTOCOL_VIOLATION_MASK	Port A ANC VPI Protocol Violation Mask	RW	0x0
16	PORT_A_YUV_PROTOCOL_VIOLATION_MASK	Port A YUV VPI Protocol Violation Mask	RW	0x0
15	PORT_B_SRC0_SIZE	Video size detected on Port B does not match size programmed in xtra_port_b register	RW	0x0
14	PORT_A_SRC0_SIZE	Video size detected on Port A does not match size programmed in xtra_port_a register	RW	0x0
13	PORT_B_DISCONN	Port B Link Disconnect Srcnum 0 Mask	RW	0x0
12	PORT_B_CONN	Port B Link Connect Srcnum 0 Mask	RW	0x0
11	PORT_A_DISCONN	Port A Link Disconnect Scrnum 0 Mask	RW	0x0
10	PORT_A_CONN	Port A Link Connect Srcnum 0 Mask	RW	0x0
9	OUTPUT_FIFO_PRTB_ANC_OF	Output FIFO Port B Ancillary Overflow Mask	RW	0x0
8	RESERVED		R	0x0
7	OUTPUT_FIFO_PRTB_YUV_OF	Output FIFO Port B Luma Overflow Mask	RW	0x0
6	OUTPUT_FIFO_PRTA_ANC_OF	Output FIFO Port A Ancillary Overflow Mask	RW	0x0
5	RESERVED		R	0x0
4	OUTPUT_FIFO_PRTA_YUV_OF	Output FIFO Port A Luma Overflow Mask	RW	0x0
3	ASYNC_FIFO_PRTB_OF	Port B Async FIFO Overflow FIQ Mask	RW	0x0
2	ASYNC_FIFO_PRTA_OF	Port A Async FIFO Overflow FIQ Mask	RW	0x0
1	PRTB_VDET_MASK	Port B Video Detect FIQ Mask	RW	0x0
0	PRTA_VDET_MASK	Port A Video Detect FIQ Mask	RW	0x0

Table 9-128. VIP_FIQ_CLEAR

Address Offset	0x0000 0018

Table 9-128. VIP_FIQ_CLEAR (continued)

Physical Address	0x4897 5518 0x4897 5A18 0x4899 5518 0x4899 5A18 0x489B 5518 0x489B 5A18	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Clears bits in the FIQ Status		
Type	RW		

31:22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PORT_A_YUV_PROTOCOL_VIOLATION_CLR	PORT_A_ANC_PROTOCOL_VIOLATION_CLR	PORT_B_YUV_PROTOCOL_VIOLATION_CLR	PORT_B_ANC_PROTOCOL_VIOLATION_CLR	PORT_A_CFG_DISABLE_COMPLETE_CLR	PORT_B_CFG_DISABLE_COMPLETE_CLR	PORT_B_SRC0_SIZE_CLR	PORT_A_SRC0_SIZE_CLR	PORT_B_DISCONN_CLR	PORT_B_CONN_CLR	PORT_A_DISCONN_CLR	PORT_A_CONN_CLR	OUTPUT_FIFO_PRTB_ANC_CLR	RESERVED	OUTPUT_FIFO_PRTB_YUV_CLR	OUTPUT_FIFO_PRTA_ANC_CLR	RESERVED	OUTPUT_FIFO_PRTA_YUV_CLR	ASYNC_FIFO_PRTB_CLR	ASYNC_FIFO_PRTA_CLR	PRTB_VDET_CLR	PRTA_VDET_CLR

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	PORT_A_YUV_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port B Cfg Disable Complete FIQ	RW	0x0
20	PORT_A_ANC_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port A Cfg Disable Complete FIQ	RW	0x0
19	PORT_B_YUV_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port B ANC VPI Protocol Violation FIQ	RW	0x0
18	PORT_B_ANC_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port B YUV VPI Protocol Violation FIQ	RW	0x0
17	PORT_A_CFG_DISABLE_COMPLETE_CLR	Write 1 followed by 0 to Clear Port A ANC VPI Protocol Violation FIQ	RW	0x0
16	PORT_B_CFG_DISABLE_COMPLETE_CLR	Write 1 followed by 0 to Clear Port A YUV VPI Protocol Violation FIQ	RW	0x0
15	PORT_B_SRC0_SIZE_CLR	Write 1 followed by 0 to Clear Port B Src0 Size FIQ	RW	0x0
14	PORT_A_SRC0_SIZE_CLR	Write 1 followed by 0 to Clear Port A Src0 Size FIQ	RW	0x0
13	PORT_B_DISCONN_CLR	Write 1 followed by 0 to Clear Port B Link Disconnect FIQ	RW	0x0
12	PORT_B_CONN_CLR	Write 1 followed by 0 to Clear Port B Link Connect FIQ	RW	0x0
11	PORT_A_DISCONN_CLR	Write 1 followed by 0 to Clear Port A Link Disconnect FIQ	RW	0x0
10	PORT_A_CONN_CLR	Write 1 followed by 0 to Clear Port A Link Connect FIQ	RW	0x0

Bits	Field Name	Description	Type	Reset
9	OUTPUT_FIFO_PRTB_ANC_CLR	Write 1 followed by 0 to Clear Output FIFO Port B Ancillary Overflow FIQ	RW	0x0
8	RESERVED		R	0x0
7	OUTPUT_FIFO_PRTB_YUV_CLR	Write 1 followed by 0 to Clear Output FIFO Port B Luma Overflow FIQ	RW	0x0
6	OUTPUT_FIFO_PRTA_ANC_CLR	Write 1 followed by 0 to Clear Output FIFO Port A Ancillary Overflow FIQ	RW	0x0
5	RESERVED		R	0x0
4	OUTPUT_FIFO_PRTA_YUV_CLR	Write 1 followed by 0 to Clear Output FIFO Port A Luma Overflow FIQ	RW	0x0
3	ASYNC_FIFO_PRTB_CLR	Write 1 followed by 0 to Clear Async FIFO Port B Overflow FIQ	RW	0x0
2	ASYNC_FIFO_PRTA_CLR	Write 1 followed by 0 to Clear Async FIFO Port A Overflow FIQ	RW	0x0
1	PRTB_VDET_CLR	Write 1 followed by 0 to Clear Video Detect FIQ for Port B	RW	0x0
0	PRTA_VDET_CLR	Write 1 followed by 0 to Clear Video Detect FIQ for Port A	RW	0x0

Table 9-129. VIP_FIQ_STATUS

Address Offset	0x0000 001C		
Physical Address	0x4897 551C 0x4897 5A1C 0x4899 551C 0x4899 5A1C 0x489B 551C 0x489B 5A1C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	FIQ Status values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										PORT_B_CFG_DISABLE_COMPLETE_CLR	PORT_A_CFG_DISABLE_COMPLETE	PORT_B_ANC_PROTOCOL_VIOLATION	PORT_B_YUV_PROTOCOL_VIOLATION	PORT_A_ANC_PROTOCOL_VIOLATION	PORT_A_YUV_PROTOCOL_VIOLATION	PORT_B_SRC0_SIZE_STATUS	PORT_A_SRC0_SIZE_STATUS	PORT_B_DISCONN_STATUS	PORT_B_CONN_STATUS	PORT_A_DISCONN_STATUS	PORT_A_CONN_STATUS	OUTPUT_FIFO_PRTB_ANC_STATUS	OUTPUT_FIFO_PRTB_CHROMA_STATUS	OUTPUT_FIFO_PRTB_LUMA_STATUS	OUTPUT_FIFO_PRTA_ANC_STATUS	OUTPUT_FIFO_PRTA_CHROMA_STATUS	OUTPUT_FIFO_PRTA_LUMA_STATUS	ASYNC_FIFO_PRTB_STATUS	ASYNC_FIFO_PRTA_STATUS	PRTB_VDET_STATUS	PRTA_VDET_STATUS

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	PORT_B_CFG_DISABLE_COMPLETE_CLR	Port B Cfg Disable Complete FIQ	R	0x0
20	PORT_A_CFG_DISABLE_COMPLETE	Port A Cfg Disable Complete FIQ	R	0x0
19	PORT_B_ANC_PROTOCOL_VIOLATION	Port B ANC VPI Protocol Violation FIQ	R	0x0
18	PORT_B_YUV_PROTOCOL_VIOLATION	Port B YUV VPI Protocol Violation FIQ	R	0x0

Bits	Field Name	Description	Type	Reset
17	PORT_A_ANC_PROTOCOL_VIOLATION	Port A ANC VPI Protocol Violation FIQ	R	0x0
16	PORT_A_YUV_PROTOCOL_VIOLATION	Port A YUV VPI Protocol Violation FIQ	R	0x0
15	PORT_B_SRC0_SIZE_STATUS	Port B Source 0 Size FIQ	R	0x0
14	PORT_A_SRC0_SIZE_STATUS	Port A Source 0 Size FIQ	R	0x0
13	PORT_B_DISCONN_STATUS	Port B Disconnect FIQ	R	0x0
12	PORT_B_CONN_STATUS	Port B Connect FIQ	R	0x0
11	PORT_A_DISCONN_STATUS	Port A Disconnect FIQ	R	0x0
10	PORT_A_CONN_STATUS	Port A Connect FIQ	R	0x0
9	OUTPUT_FIFO_PRTB_ANC_STATUS	Output FIFO Port B Ancillary Overflow Status	R	0x0
8	OUTPUT_FIFO_PRTB_CHROMA_STATUS	Output FIFO Port B Chroma Overflow Status	R	0x0
7	OUTPUT_FIFO_PRTB_LUMA_STATUS	Output FIFO Port B Luma Overflow Status	R	0x0
6	OUTPUT_FIFO_PRTA_ANC_STATUS	Output FIFO Port A Ancillary Overflow Status	R	0x0
5	OUTPUT_FIFO_PRTA_CHROMA_STATUS	Output FIFO Port A Chroma Overflow Status	R	0x0
4	OUTPUT_FIFO_PRTA_LUMA_STATUS	Output FIFO Port A Luma Overflow Status	R	0x0
3	ASYNC_FIFO_PRTB_STATUS	Async FIFO Port B Overflow Status	R	0x0
2	ASYNC_FIFO_PRTA_STATUS	Async FIFO Port A Overflow Status	R	0x0
1	PRTB_VDET_STATUS	VDET Status for Port B	R	0x0
0	PRTA_VDET_STATUS	VDET Status for Port A	R	0x0

Table 9-130. VIP_OUTPUT_PORT_A_SRC_FID

Address Offset	0x0000 0020		
Physical Address	0x4897 5520 0x4897 5A20 0x4899 5520 0x4899 5A20 0x489B 5520 0x489B 5A20	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Current and Previous Output Port A Source FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRTA_SRC15_CURR_SOURCE_FID	PRTA_SRC15_PREV_SOURCE_FID	PRTA_SRC14_CURR_SOURCE_FID	PRTA_SRC14_PREV_SOURCE_FID	PRTA_SRC13_CURR_SOURCE_FID	PRTA_SRC13_PREV_SOURCE_FID	PRTA_SRC12_CURR_SOURCE_FID	PRTA_SRC12_PREV_SOURCE_FID	PRTA_SRC11_CURR_SOURCE_FID	PRTA_SRC11_PREV_SOURCE_FID	PRTA_SRC10_CURR_SOURCE_FID	PRTA_SRC10_PREV_SOURCE_FID	PRTA_SRC9_CURR_SOURCE_FID	PRTA_SRC9_PREV_SOURCE_FID	PRTA_SRC8_CURR_SOURCE_FID	PRTA_SRC8_PREV_SOURCE_FID

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTA_SRC7_CURR_SOURCE_FID	PRTA_SRC7_PREV_SOURCE_FID	PRTA_SRC6_CURR_SOURCE_FID	PRTA_SRC6_PREV_SOURCE_FID	PRTA_SRC5_CURR_SOURCE_FID	PRTA_SRC5_PREV_SOURCE_FID	PRTA_SRC4_CURR_SOURCE_FID	PRTA_SRC4_PREV_SOURCE_FID	PRTA_SRC3_CURR_SOURCE_FID	PRTA_SRC3_PREV_SOURCE_FID	PRTA_SRC2_CURR_SOURCE_FID	PRTA_SRC2_PREV_SOURCE_FID	PRTA_SRC1_CURR_SOURCE_FID	PRTA_SRC1_PREV_SOURCE_FID	PRTA_SRC0_CURR_SOURCE_FID	PRTA_SRC0_PREV_SOURCE_FID

Bits	Field Name	Description	Type	Reset
31	PRTA_SRC15_CURR_SOURCE_FID	For Source ID 15 from Port A. Source Field ID for Current Field	R	0x0
30	PRTA_SRC15_PREV_SOURCE_FID	For Source ID 15 from Port A. Source Field ID for Previous Field	R	0x0

Bits	Field Name	Description	Type	Reset
29	PRTA_SRC14_CURR_SOURCE_FID	For Source ID 14 from Port A. Source Field ID for Current Field	R	0x0
28	PRTA_SRC14_PREV_SOURCE_FID	For Source ID 14 from Port A. Source Field ID for Previous Field	R	0x0
27	PRTA_SRC13_CURR_SOURCE_FID	For Source ID 13 from Port A. Source Field ID for Current Field	R	0x0
26	PRTA_SRC13_PREV_SOURCE_FID	For Source ID 13 from Port A. Source Field ID for Previous Field	R	0x0
25	PRTA_SRC12_CURR_SOURCE_FID	For Source ID 12 from Port A. Source Field ID for Current Field	R	0x0
24	PRTA_SRC12_PREV_SOURCE_FID	For Source ID 12 from Port A. Source Field ID for Previous Field	R	0x0
23	PRTA_SRC11_CURR_SOURCE_FID	For Source ID 11 from Port A. Source Field ID for Current Field	R	0x0
22	PRTA_SRC11_PREV_SOURCE_FID	For Source ID 11 from Port A. Source Field ID for Previous Field	R	0x0
21	PRTA_SRC10_CURR_SOURCE_FID	For Source ID 10 from Port A. Source Field ID for Current Field	R	0x0
20	PRTA_SRC10_PREV_SOURCE_FID	For Source ID 10 from Port A. Source Field ID for Previous Field	R	0x0
19	PRTA_SRC9_CURR_SOURCE_FID	For Source ID 9 from Port A. Source Field ID for Current Field	R	0x0
18	PRTA_SRC9_PREV_SOURCE_FID	For Source ID 9 from Port A. Source Field ID for Previous Field	R	0x0
17	PRTA_SRC8_CURR_SOURCE_FID	For Source ID 8 from Port A. Source Field ID for Current Field	R	0x0
16	PRTA_SRC8_PREV_SOURCE_FID	For Source ID 8 from Port A. Source Field ID for Previous Field	R	0x0
15	PRTA_SRC7_CURR_SOURCE_FID	For Source ID 7 from Port A. Source Field ID for Current Field	R	0x0
14	PRTA_SRC7_PREV_SOURCE_FID	For Source ID 7 from Port A. Source Field ID for Previous Field	R	0x0
13	PRTA_SRC6_CURR_SOURCE_FID	For Source ID 6 from Port A. Source Field ID for Current Field	R	0x0
12	PRTA_SRC6_PREV_SOURCE_FID	For Source ID 6 from Port A. Source Field ID for Previous Field	R	0x0
11	PRTA_SRC5_CURR_SOURCE_FID	For Source ID 5 from Port A. Source Field ID for Current Field	R	0x0
10	PRTA_SRC5_PREV_SOURCE_FID	For Source ID 5 from Port A. Source Field ID for Previous Field	R	0x0
9	PRTA_SRC4_CURR_SOURCE_FID	For Source ID 4 from Port A. Source Field ID for Current Field	R	0x0
8	PRTA_SRC4_PREV_SOURCE_FID	For Source ID 4 from Port A. Source Field ID for Previous Field	R	0x0
7	PRTA_SRC3_CURR_SOURCE_FID	For Source ID 3 from Port A. Source Field ID for Current Field	R	0x0
6	PRTA_SRC3_PREV_SOURCE_FID	For Source ID 3 from Port A. Source Field ID for Previous Field	R	0x0
5	PRTA_SRC2_CURR_SOURCE_FID	For Source ID 2 from Port A. Source Field ID for Current Field	R	0x0
4	PRTA_SRC2_PREV_SOURCE_FID	For Source ID 2 from Port A. Source Field ID for Previous Field	R	0x0
3	PRTA_SRC1_CURR_SOURCE_FID	For Source ID 1 from Port A. Source Field ID for Current Field	R	0x0

Bits	Field Name	Description	Type	Reset
2	PRTA_SRC1_PREV_SOURCE_FID	For Source ID 1 from Port A. Source Field ID for Previous Field	R	0x0
1	PRTA_SRC0_CURR_SOURCE_FID	For Source ID 0 from Port A. Source Field ID for Current Field	R	0x0
0	PRTA_SRC0_PREV_SOURCE_FID	For Source ID 0 from Port A. Source Field ID for Previous Field	R	0x0

Table 9-131. VIP_OUTPUT_PORT_A_ENC_FID

Address Offset	0x0000 0024		
Physical Address	0x4897 5524 0x4897 5A24 0x4899 5524 0x4899 5A24 0x489B 5524 0x489B 5A24	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Current and Previous Output Port A Encoder FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRTA_SRC15_CURR_ENC_FID	PRTA_SRC15_PREV_ENC_FID	PRTA_SRC14_CURR_ENC_FID	PRTA_SRC14_PREV_ENC_FID	PRTA_SRC13_CURR_ENC_FID	PRTA_SRC13_PREV_ENC_FID	PRTA_SRC12_CURR_ENC_FID	PRTA_SRC12_PREV_ENC_FID	PRTA_SRC11_CURR_ENC_FID	PRTA_SRC11_PREV_ENC_FID	PRTA_SRC10_CURR_ENC_FID	PRTA_SRC10_PREV_ENC_FID	PRTA_SRC9_CURR_ENC_FID	PRTA_SRC9_PREV_ENC_FID	PRTA_SRC8_CURR_ENC_FID	PRTA_SRC8_PREV_ENC_FID

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTA_SRC7_CURR_ENC_FID	PRTA_SRC7_PREV_ENC_FID	PRTA_SRC6_CURR_ENC_FID	PRTA_SRC6_PREV_ENC_FID	PRTA_SRC5_CURR_ENC_FID	PRTA_SRC5_PREV_ENC_FID	PRTA_SRC4_CURR_ENC_FID	PRTA_SRC4_PREV_ENC_FID	PRTA_SRC3_CURR_ENC_FID	PRTA_SRC3_PREV_ENC_FID	PRTA_SRC2_CURR_ENC_FID	PRTA_SRC2_PREV_ENC_FID	PRTA_SRC1_CURR_ENC_FID	PRTA_SRC1_PREV_ENC_FID	PRTA_SRC0_CURR_ENC_FID	PRTA_SRC0_PREV_ENC_FID

Bits	Field Name	Description	Type	Reset
31	PRTA_SRC15_CURR_ENC_FID	For Source ID 15 from Port A. Encoder Field ID for Current Field	R	0x0
30	PRTA_SRC15_PREV_ENC_FID	For Source ID 15 from Port A. Encoder Field ID for Previous Field	R	0x0
29	PRTA_SRC14_CURR_ENC_FID	For Source ID 14 from Port A. Encoder Field ID for Current Field	R	0x0
28	PRTA_SRC14_PREV_ENC_FID	For Source ID 14 from Port A. Encoder Field ID for Previous Field	R	0x0
27	PRTA_SRC13_CURR_ENC_FID	For Source ID 13 from Port A. Encoder Field ID for Current Field	R	0x0
26	PRTA_SRC13_PREV_ENC_FID	For Source ID 13 from Port A. Encoder Field ID for Previous Field	R	0x0
25	PRTA_SRC12_CURR_ENC_FID	For Source ID 12 from Port A. Encoder Field ID for Current Field	R	0x0
24	PRTA_SRC12_PREV_ENC_FID	For Source ID 12 from Port A. Encoder Field ID for Previous Field	R	0x0
23	PRTA_SRC11_CURR_ENC_FID	For Source ID 11 from Port A. Encoder Field ID for Current Field	R	0x0
22	PRTA_SRC11_PREV_ENC_FID	For Source ID 11 from Port A. Encoder Field ID for Previous Field	R	0x0
21	PRTA_SRC10_CURR_ENC_FID	For Source ID 10 from Port A. Encoder Field ID for Current Field	R	0x0

Bits	Field Name	Description	Type	Reset
20	PRTA_SRC10_PREV_ENC_FID	For Source ID 10 from Port A. Encoder Field ID for Previous Field	R	0x0
19	PRTA_SRC9_CURR_ENC_FID	For Source ID 9 from Port A. Encoder Field ID for Current Field	R	0x0
18	PRTA_SRC9_PREV_ENC_FID	For Source ID 9 from Port A. Encoder Field ID for Previous Field	R	0x0
17	PRTA_SRC8_CURR_ENC_FID	For Source ID 8 from Port A. Encoder Field ID for Current Field	R	0x0
16	PRTA_SRC8_PREV_ENC_FID	For Source ID 8 from Port A. Encoder Field ID for Previous Field	R	0x0
15	PRTA_SRC7_CURR_ENC_FID	For Source ID 7 from Port A. Encoder Field ID for Current Field	R	0x0
14	PRTA_SRC7_PREV_ENC_FID	For Source ID 7 from Port A. Encoder Field ID for Previous Field	R	0x0
13	PRTA_SRC6_CURR_ENC_FID	For Source ID 6 from Port A. Encoder Field ID for Current Field	R	0x0
12	PRTA_SRC6_PREV_ENC_FID	For Source ID 6 from Port A. Encoder Field ID for Previous Field	R	0x0
11	PRTA_SRC5_CURR_ENC_FID	For Source ID 5 from Port A. Encoder Field ID for Current Field	R	0x0
10	PRTA_SRC5_PREV_ENC_FID	For Source ID 5 from Port A. Encoder Field ID for Previous Field	R	0x0
9	PRTA_SRC4_CURR_ENC_FID	For Source ID 4 from Port A. Encoder Field ID for Current Field	R	0x0
8	PRTA_SRC4_PREV_ENC_FID	For Source ID 4 from Port A. Encoder Field ID for Previous Field	R	0x0
7	PRTA_SRC3_CURR_ENC_FID	For Source ID 3 from Port A. Encoder Field ID for Current Field	R	0x0
6	PRTA_SRC3_PREV_ENC_FID	For Source ID 3 from Port A. Encoder Field ID for Previous Field	R	0x0
5	PRTA_SRC2_CURR_ENC_FID	For Source ID 2 from Port A. Encoder Field ID for Current Field	R	0x0
4	PRTA_SRC2_PREV_ENC_FID	For Source ID 2 from Port A. Encoder Field ID for Previous Field	R	0x0
3	PRTA_SRC1_CURR_ENC_FID	For Source ID 1 from Port A. Encoder Field ID for Current Field	R	0x0
2	PRTA_SRC1_PREV_ENC_FID	For Source ID 1 from Port A. Encoder Field ID for Previous Field	R	0x0
1	PRTA_SRC0_CURR_ENC_FID	For Source ID 0 from Port A. Encoder Field ID for Current Field	R	0x0
0	PRTA_SRC0_PREV_ENC_FID	For Source ID 0 from Port A. Encoder Field ID for Previous Field	R	0x0

Table 9-132. VIP_OUTPUT_PORT_B_SRC_FID

Address Offset	0x0000 0028		
Physical Address	0x4897 5528 0x4897 5A28 0x4899 5528 0x4899 5A28 0x489B 5528 0x489B 5A28	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Current and Previous Output Port B Source FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PRTB_SRC15_CURR_SOURCE_FID	PRTB_SRC15_PREV_SOURCE_FID	PRTB_SRC14_CURR_SOURCE_FID	PRTB_SRC14_PREV_SOURCE_FID	PRTB_SRC13_CURR_SOURCE_FID	PRTB_SRC13_PREV_SOURCE_FID	PRTB_SRC12_CURR_SOURCE_FID	PRTB_SRC12_PREV_SOURCE_FID	PRTB_SRC11_CURR_SOURCE_FID	PRTB_SRC11_PREV_SOURCE_FID	PRTB_SRC10_CURR_SOURCE_FID	PRTB_SRC10_PREV_SOURCE_FID	PRTB_SRC9_CURR_SOURCE_FID	PRTB_SRC9_PREV_SOURCE_FID	PRTB_SRC8_CURR_SOURCE_FID	PRTB_SRC8_PREV_SOURCE_FID	PRTB_SRC7_CURR_SOURCE_FID	PRTB_SRC7_PREV_SOURCE_FID	PRTB_SRC6_CURR_SOURCE_FID	PRTB_SRC6_PREV_SOURCE_FID	PRTB_SRC5_CURR_SOURCE_FID	PRTB_SRC5_PREV_SOURCE_FID	PRTB_SRC4_CURR_SOURCE_FID	PRTB_SRC4_PREV_SOURCE_FID	PRTB_SRC3_CURR_SOURCE_FID	PRTB_SRC3_PREV_SOURCE_FID	PRTB_SRC2_CURR_SOURCE_FID	PRTB_SRC2_PREV_SOURCE_FID	PRTB_SRC1_CURR_SOURCE_FID	PRTB_SRC1_PREV_SOURCE_FID	PRTB_SRC0_CURR_SOURCE_FID	PRTB_SRC0_PREV_SOURCE_FID

Bits	Field Name	Description	Type	Reset
31	PRTB_SRC15_CURR_SOURCE_FID	For Source ID 15 from Port B. Source Field ID for Current Field	R	0x0
30	PRTB_SRC15_PREV_SOURCE_FID	For Source ID 15 from Port B. Source Field ID for Previous Field	R	0x0
29	PRTB_SRC14_CURR_SOURCE_FID	For Source ID 14 from Port B. Source Field ID for Current Field	R	0x0
28	PRTB_SRC14_PREV_SOURCE_FID	For Source ID 14 from Port B. Source Field ID for Previous Field	R	0x0
27	PRTB_SRC13_CURR_SOURCE_FID	For Source ID 13 from Port B. Source Field ID for Current Field	R	0x0
26	PRTB_SRC13_PREV_SOURCE_FID	For Source ID 13 from Port B. Source Field ID for Previous Field	R	0x0
25	PRTB_SRC12_CURR_SOURCE_FID	For Source ID 12 from Port B. Source Field ID for Current Field	R	0x0
24	PRTB_SRC12_PREV_SOURCE_FID	For Source ID 12 from Port B. Source Field ID for Previous Field	R	0x0
23	PRTB_SRC11_CURR_SOURCE_FID	For Source ID 11 from Port B. Source Field ID for Current Field	R	0x0
22	PRTB_SRC11_PREV_SOURCE_FID	For Source ID 11. from Port B Source Field ID for Previous Field	R	0x0
21	PRTB_SRC10_CURR_SOURCE_FID	For Source ID 10 from Port B. Source Field ID for Current Field	R	0x0
20	PRTB_SRC10_PREV_SOURCE_FID	For Source ID 10 from Port B. Source Field ID for Previous Field	R	0x0
19	PRTB_SRC9_CURR_SOURCE_FID	For Source ID 9 from Port B. Source Field ID for Current Field	R	0x0
18	PRTB_SRC9_PREV_SOURCE_FID	For Source ID 9 from Port B. Source Field ID for Previous Field	R	0x0
17	PRTB_SRC8_CURR_SOURCE_FID	For Source ID 8 from Port B. Source Field ID for Current Field	R	0x0
16	PRTB_SRC8_PREV_SOURCE_FID	For Source ID 8 from Port B. Source Field ID for Previous Field	R	0x0
15	PRTB_SRC7_CURR_SOURCE_FID	For Source ID 7 from Port B. Source Field ID for Current Field	R	0x0
14	PRTB_SRC7_PREV_SOURCE_FID	For Source ID 7 from Port B. Source Field ID for Previous Field	R	0x0
13	PRTB_SRC6_CURR_SOURCE_FID	For Source ID 6 from Port B. Source Field ID for Current Field	R	0x0

Bits	Field Name	Description	Type	Reset
12	PRTB_SRC6_PREV_SOURCE_FID	For Source ID 6 from Port B. Source Field ID for Previous Field	R	0x0
11	PRTB_SRC5_CURR_SOURCE_FID	For Source ID 5 from Port B. Source Field ID for Current Field	R	0x0
10	PRTB_SRC5_PREV_SOURCE_FID	For Source ID 5 from Port B. Source Field ID for Previous Field	R	0x0
9	PRTB_SRC4_CURR_SOURCE_FID	For Source ID 4 from Port B. Source Field ID for Current Field	R	0x0
8	PRTB_SRC4_PREV_SOURCE_FID	For Source ID 4 from Port B. Source Field ID for Previous Field	R	0x0
7	PRTB_SRC3_CURR_SOURCE_FID	For Source ID 3 from Port B. Source Field ID for Current Field	R	0x0
6	PRTB_SRC3_PREV_SOURCE_FID	For Source ID 3 from Port B. Source Field ID for Previous Field	R	0x0
5	PRTB_SRC2_CURR_SOURCE_FID	For Source ID 2 from Port B. Source Field ID for Current Field	R	0x0
4	PRTB_SRC2_PREV_SOURCE_FID	For Source ID 2 from Port B. Source Field ID for Previous Field	R	0x0
3	PRTB_SRC1_CURR_SOURCE_FID	For Source ID 1 from Port B. Source Field ID for Current Field	R	0x0
2	PRTB_SRC1_PREV_SOURCE_FID	For Source ID 1 from Port B. Source Field ID for Previous Field	R	0x0
1	PRTB_SRC0_CURR_SOURCE_FID	For Source ID 0 from Port B. Source Field ID for Current Field	R	0x0
0	PRTB_SRC0_PREV_SOURCE_FID	For Source ID 0 from Port B. Source Field ID for Previous Field	R	0x0

Table 9-133. VIP_OUTPUT_PORT_B_ENC_FID

Address Offset	0x0000 002C		
Physical Address	0x4897 552C 0x4897 5A2C 0x4899 552C 0x4899 5A2C 0x489B 552C 0x489B 5A2C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Current and Previous Output Port B Encoder FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRTB_SRC15_CURR_ENC_FID	PRTB_SRC15_PREV_ENC_FID	PRTB_SRC14_CURR_ENC_FID	PRTB_SRC14_PREV_ENC_FID	PRTB_SRC13_CURR_ENC_FID	PRTB_SRC13_PREV_ENC_FID	PRTB_SRC12_CURR_ENC_FID	PRTB_SRC12_PREV_ENC_FID	PRTB_SRC11_CURR_ENC_FID	PRTB_SRC11_PREV_ENC_FID	PRTB_SRC10_CURR_ENC_FID	PRTB_SRC10_PREV_ENC_FID	PRTB_SRC9_CURR_ENC_FID	PRTB_SRC9_PREV_ENC_FID	PRTB_SRC8_CURR_ENC_FID	PRTB_SRC8_PREV_ENC_FID

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTB_SRC7_CURR_ENC_FID	PRTB_SRC7_PREV_ENC_FID	PRTB_SRC6_CURR_ENC_FID	PRTB_SRC6_PREV_ENC_FID	PRTB_SRC5_CURR_ENC_FID	PRTB_SRC5_PREV_ENC_FID	PRTB_SRC4_CURR_ENC_FID	PRTB_SRC4_PREV_ENC_FID	PRTB_SRC3_CURR_ENC_FID	PRTB_SRC3_PREV_ENC_FID	PRTB_SRC2_CURR_ENC_FID	PRTB_SRC2_PREV_ENC_FID	PRTB_SRC1_CURR_ENC_FID	PRTB_SRC1_PREV_ENC_FID	PRTB_SRC0_CURR_ENC_FID	PRTB_SRC0_PREV_ENC_FID

Bits	Field Name	Description	Type	Reset
31	PRTB_SRC15_CURR_ENC_FID	For Source ID 15 from Port B. Encoder Field ID for Current Field	R	0x0

Bits	Field Name	Description	Type	Reset
30	PRTB_SRC15_PREV_ENC_FID	For Source ID 15 from Port B. Encoder Field ID for Previous Field	R	0x0
29	PRTB_SRC14_CURR_ENC_FID	For Source ID 14 from Port B. Encoder Field ID for Current Field	R	0x0
28	PRTB_SRC14_PREV_ENC_FID	For Source ID 14 from Port B. Encoder Field ID for Previous Field	R	0x0
27	PRTB_SRC13_CURR_ENC_FID	For Source ID 13 from Port B. Encoder Field ID for Current Field	R	0x0
26	PRTB_SRC13_PREV_ENC_FID	For Source ID 13 from Port B. Encoder Field ID for Previous Field	R	0x0
25	PRTB_SRC12_CURR_ENC_FID	For Source ID 12 from Port B. Encoder Field ID for Current Field	R	0x0
24	PRTB_SRC12_PREV_ENC_FID	For Source ID 12 from Port B. Encoder Field ID for Previous Field	R	0x0
23	PRTB_SRC11_CURR_ENC_FID	For Source ID 11 from Port B. Encoder Field ID for Current Field	R	0x0
22	PRTB_SRC11_PREV_ENC_FID	For Source ID 11 from Port B. Encoder Field ID for Previous Field	R	0x0
21	PRTB_SRC10_CURR_ENC_FID	For Source ID 10 from Port B. Encoder Field ID for Current Field	R	0x0
20	PRTB_SRC10_PREV_ENC_FID	For Source ID 10 from Port B. Encoder Field ID for Previous Field	R	0x0
19	PRTB_SRC9_CURR_ENC_FID	For Source ID 9 from Port B. Encoder Field ID for Current Field	R	0x0
18	PRTB_SRC9_PREV_ENC_FID	For Source ID 9 from Port B. Encoder Field ID for Previous Field	R	0x0
17	PRTB_SRC8_CURR_ENC_FID	For Source ID 8 from Port B. Encoder Field ID for Current Field	R	0x0
16	PRTB_SRC8_PREV_ENC_FID	For Source ID 8 from Port B. Encoder Field ID for Previous Field	R	0x0
15	PRTB_SRC7_CURR_ENC_FID	For Source ID 7 from Port B. Encoder Field ID for Current Field	R	0x0
14	PRTB_SRC7_PREV_ENC_FID	For Source ID 7 from Port B. Encoder Field ID for Previous Field	R	0x0
13	PRTB_SRC6_CURR_ENC_FID	For Source ID 6 from Port B. Encoder Field ID for Current Field	R	0x0
12	PRTB_SRC6_PREV_ENC_FID	For Source ID 6 from Port B. Encoder Field ID for Previous Field	R	0x0
11	PRTB_SRC5_CURR_ENC_FID	For Source ID 5 from Port B. Encoder Field ID for Current Field	R	0x0
10	PRTB_SRC5_PREV_ENC_FID	For Source ID 5 from Port B. Encoder Field ID for Previous Field	R	0x0
9	PRTB_SRC4_CURR_ENC_FID	For Source ID 4 from Port B. Encoder Field ID for Current Field	R	0x0
8	PRTB_SRC4_PREV_ENC_FID	For Source ID 4 from Port B. Encoder Field ID for Previous Field	R	0x0
7	PRTB_SRC3_CURR_ENC_FID	For Source ID 3 from Port B. Encoder Field ID for Current Field	R	0x0
6	PRTB_SRC3_PREV_ENC_FID	For Source ID 3 from Port B. Encoder Field ID for Previous Field	R	0x0
5	PRTB_SRC2_CURR_ENC_FID	For Source ID 2 from Port B. Encoder Field ID for Current Field	R	0x0
4	PRTB_SRC2_PREV_ENC_FID	For Source ID 2 from Port B. Encoder Field ID for Previous Field	R	0x0

Bits	Field Name	Description	Type	Reset
3	PRTB_SRC1_CURR_ENC_FID	For Source ID 1 from Port B. Encoder Field ID for Current Field	R	0x0
2	PRTB_SRC1_PREV_ENC_FID	For Source ID 1 from Port B. Encoder Field ID for Previous Field	R	0x0
1	PRTB_SRC0_CURR_ENC_FID	For Source ID 0 from Port B. Encoder Field ID for Current Field	R	0x0
0	PRTB_SRC0_PREV_ENC_FID	For Source ID 0 from Port B. Encoder Field ID for Previous Field	R	0x0

Table 9-134. VIP_OUTPUT_PORT_A_SRC0_SIZE

Address Offset	0x0000 0030		
Physical Address	0x4897 5530 0x4897 5A30 0x4899 5530 0x4899 5A30 0x489B 5530 0x489B 5A30	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 0		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC0_WIDTH	RESERVED	PRTA_SRC0_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC0_WIDTH	On Port A. Width of Source ID 0	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC0_HEIGHT	On Port A. Height of Source ID 0	R	0x0

Table 9-135. VIP_OUTPUT_PORT_A_SRC1_SIZE

Address Offset	0x0000 0034		
Physical Address	0x4897 5534 0x4897 5A34 0x4899 5534 0x4899 5A34 0x489B 5534 0x489B 5A34	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 1		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC1_WIDTH	RESERVED	PRTA_SRC1_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC1_WIDTH	On Port A. Width of Source ID 1	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC1_HEIGHT	On Port A. Height of Source ID 1	R	0x0

Table 9-136. VIP_OUTPUT_PORT_A_SRC2_SIZE

Address Offset	0x0000 0038

Table 9-136. VIP_OUTPUT_PORT_A_SRC2_SIZE (continued)

Physical Address	0x4897 5538 0x4897 5A38 0x4899 5538 0x4899 5A38 0x489B 5538 0x489B 5A38	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 2		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC2_WIDTH	RESERVED	PRTA_SRC2_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC2_WIDTH	On Port A. Width of Source ID 2	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC2_HEIGHT	On Port A. Height of Source ID 2	R	0x0

Table 9-137. VIP_OUTPUT_PORT_A_SRC3_SIZE

Address Offset	0x0000 003C		
Physical Address	0x4897 553C 0x4897 5A3C 0x4899 553C 0x4899 5A3C 0x489B 553C 0x489B 5A3C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 3		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC3_WIDTH	RESERVED	PRTA_SRC3_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC3_WIDTH	On Port A. Width of Source ID 3	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC3_HEIGHT	On Port A. Height of Source ID 3	R	0x0

Table 9-138. VIP_OUTPUT_PORT_A_SRC4_SIZE

Address Offset	0x0000 0040		
Physical Address	0x4897 5540 0x4897 5A40 0x4899 5540 0x4899 5A40 0x489B 5540 0x489B 5A40	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 4		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC4_WIDTH	RESERVED	PRTA_SRC4_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:16	PRTA_SRC4_WIDTH	On Port A. Width of Source ID 4	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC4_HEIGHT	On Port A. Height of Source ID 4	R	0x0

Table 9-139. VIP_OUTPUT_PORT_A_SRC5_SIZE

Address Offset	0x0000 0044		
Physical Address	0x4897 5544 0x4897 5A44 0x4899 5544 0x4899 5A44 0x489B 5544 0x489B 5A44	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 5		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC5_WIDTH	RESERVED	PRTA_SRC5_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC5_WIDTH	On Port A. Width of Source ID 5	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC5_HEIGHT	On Port A. Height of Source ID 5	R	0x0

Table 9-140. VIP_OUTPUT_PORT_A_SRC6_SIZE

Address Offset	0x0000 0048		
Physical Address	0x4897 5548 0x4897 5A48 0x4899 5548 0x4899 5A48 0x489B 5548 0x489B 5A48	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 6		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC6_WIDTH	RESERVED	PRTA_SRC6_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC6_WIDTH	On Port A. Width of Source ID 6	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC6_HEIGHT	On Port A. Height of Source ID 6	R	0x0

Table 9-141. VIP_OUTPUT_PORT_A_SRC7_SIZE

Address Offset	0x0000 004C		
Physical Address	0x4897 554C 0x4897 5A4C 0x4899 554C 0x4899 5A4C 0x489B 554C 0x489B 5A4C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 7		

Table 9-141. VIP_OUTPUT_PORT_A_SRC7_SIZE (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTA_SRC7_WIDTH											RESERVED					PRTA_SRC7_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC7_WIDTH	On Port A. Width of Source ID 7	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC7_HEIGHT	On Port A. Height of Source ID 7	R	0x0

Table 9-142. VIP_OUTPUT_PORT_A_SRC8_SIZE

Address Offset	0x0000 0050		
Physical Address	0x4897 5550 0x4897 5A50 0x4899 5550 0x4899 5A50 0x489B 5550 0x489B 5A50	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 8		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTA_SRC8_WIDTH											RESERVED					PRTA_SRC8_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC8_WIDTH	On Port A. Width of Source ID 8	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC8_HEIGHT	On Port A. Height of Source ID 8	R	0x0

Table 9-143. VIP_OUTPUT_PORT_A_SRC9_SIZE

Address Offset	0x0000 0054		
Physical Address	0x4897 5554 0x4897 5A54 0x4899 5554 0x4899 5A54 0x489B 5554 0x489B 5A54	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 9		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTA_SRC9_WIDTH											RESERVED					PRTA_SRC9_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC9_WIDTH	On Port A. Width of Source ID 9	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC9_HEIGHT	On Port A. Height of Source ID 9	R	0x0

Table 9-144. VIP_OUTPUT_PORT_A_SRC10_SIZE

Address Offset	0x0000 0058

Table 9-144. VIP_OUTPUT_PORT_A_SRC10_SIZE (continued)

Physical Address	0x4897 5558 0x4897 5A58 0x4899 5558 0x4899 5A58 0x489B 5558 0x489B 5A58	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 10		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC10_WIDTH	RESERVED	PRTA_SRC10_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC10_WIDTH	On Port A. Width of Source ID 10	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC10_HEIGHT	On Port A. Height of Source ID 10	R	0x0

Table 9-145. VIP_OUTPUT_PORT_A_SRC11_SIZE

Address Offset	0x0000 005C		
Physical Address	0x4897 555C 0x4897 5A5C 0x4899 555C 0x4899 5A5C 0x489B 555C 0x489B 5A5C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 11		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC11_WIDTH	RESERVED	PRTA_SRC11_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC11_WIDTH	On Port A. Width of Source ID 11	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC11_HEIGHT	On Port A. Height of Source ID 11	R	0x0

Table 9-146. VIP_OUTPUT_PORT_A_SRC12_SIZE

Address Offset	0x0000 0060		
Physical Address	0x4897 5560 0x4897 5A60 0x4899 5560 0x4899 5A60 0x489B 5560 0x489B 5A60	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 12		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTA_SRC12_WIDTH	RESERVED	PRTA_SRC12_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:16	PRTA_SRC12_WIDTH	On Port A. Width of Source ID 12	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC12_HEIGHT	On Port A. Height of Source ID 12	R	0x0

Table 9-147. VIP_OUTPUT_PORT_A_SRC13_SIZE

Address Offset	0x0000 0064		
Physical Address	0x4897 5564 0x4897 5A64 0x4899 5564 0x4899 5A64 0x489B 5564 0x489B 5A64	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 13		
Type	R		

31 30 29 28 27	26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8 7 6 5 4 3 2 1 0
RESERVED	PRTA_SRC13_WIDTH	RESERVED	PRTA_SRC13_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC13_WIDTH	On Port A. Width of Source ID 13	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC13_HEIGHT	On Port A. Height of Source ID 13	R	0x0

Table 9-148. VIP_OUTPUT_PORT_A_SRC14_SIZE

Address Offset	0x0000 0068		
Physical Address	0x4897 5568 0x4897 5A68 0x4899 5568 0x4899 5A68 0x489B 5568 0x489B 5A68	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 14		
Type	R		

31 30 29 28 27	26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8 7 6 5 4 3 2 1 0
RESERVED	PRTA_SRC14_WIDTH	RESERVED	PRTA_SRC14_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC14_WIDTH	On Port A. Width of Source ID 14	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC14_HEIGHT	On Port A. Height of Source ID 14	R	0x0

Table 9-149. VIP_OUTPUT_PORT_A_SRC15_SIZE

Address Offset	0x0000 006C		
Physical Address	0x4897 556C 0x4897 5A6C 0x4899 556C 0x4899 5A6C 0x489B 556C 0x489B 5A6C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 15		

Table 9-149. VIP_OUTPUT_PORT_A_SRC15_SIZE (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTA_SRC15_WIDTH											RESERVED					PRTA_SRC15_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC15_WIDTH	On Port A. Width of Source ID 15	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC15_HEIGHT	On Port A. Height of Source ID 15	R	0x0

Table 9-150. VIP_OUTPUT_PORT_B_SRC0_SIZE

Address Offset	0x0000 0070		
Physical Address	0x4897 5570 0x4897 5A70 0x4899 5570 0x4899 5A70 0x489B 5570 0x489B 5A70	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 0		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTB_SRC0_WIDTH											RESERVED					PRTB_SRC0_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC0_WIDTH	On Port B. Width of Source ID 0	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC0_HEIGHT	On Port B. Height of Source ID 0	R	0x0

Table 9-151. VIP_OUTPUT_PORT_B_SRC1_SIZE

Address Offset	0x0000 0074		
Physical Address	0x4897 5574 0x4897 5A74 0x4899 5574 0x4899 5A74 0x489B 5574 0x489B 5A74	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 1		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTB_SRC1_WIDTH											RESERVED					PRTB_SRC1_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC1_WIDTH	On Port B. Width of Source ID 1	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC1_HEIGHT	On Port B. Height of Source ID 1	R	0x0

Table 9-152. VIP_OUTPUT_PORT_B_SRC2_SIZE

Address Offset	0x0000 0078

Table 9-152. VIP_OUTPUT_PORT_B_SRC2_SIZE (continued)

Physical Address	0x4897 5578 0x4897 5A78 0x4899 5578 0x4899 5A78 0x489B 5578 0x489B 5A78	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 2		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTB_SRC2_WIDTH	RESERVED	PRTB_SRC2_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC2_WIDTH	On Port B. Width of Source ID 2	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC2_HEIGHT	On Port B. Height of Source ID 2	R	0x0

Table 9-153. VIP_OUTPUT_PORT_B_SRC3_SIZE

Address Offset	0x0000 007C		
Physical Address	0x4897 557C 0x4897 5A7C 0x4899 557C 0x4899 5A7C 0x489B 557C 0x489B 5A7C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 3		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTB_SRC3_WIDTH	RESERVED	PRTB_SRC3_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC3_WIDTH	On Port B. Width of Source ID 3	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC3_HEIGHT	On Port B. Height of Source ID 3	R	0x0

Table 9-154. VIP_OUTPUT_PORT_B_SRC4_SIZE

Address Offset	0x0000 0080		
Physical Address	0x4897 5580 0x4897 5A80 0x4899 5580 0x4899 5A80 0x489B 5580 0x489B 5A80	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 4		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTB_SRC4_WIDTH	RESERVED	PRTB_SRC4_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:16	PRTB_SRC4_WIDTH	On Port B. Width of Source ID 4	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC4_HEIGHT	On Port B. Height of Source ID 4	R	0x0

Table 9-155. VIP_OUTPUT_PORT_B_SRC5_SIZE

Address Offset	0x0000 0084		
Physical Address	0x4897 5584 0x4897 5A84 0x4899 5584 0x4899 5A84 0x489B 5584 0x489B 5A84	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 5		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTB_SRC5_WIDTH	RESERVED	PRTB_SRC5_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC5_WIDTH	On Port B. Width of Source ID 5	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC5_HEIGHT	On Port B. Height of Source ID 5	R	0x0

Table 9-156. VIP_OUTPUT_PORT_B_SRC6_SIZE

Address Offset	0x0000 0088		
Physical Address	0x4897 5588 0x4897 5A88 0x4899 5588 0x4899 5A88 0x489B 5588 0x489B 5A88	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 6		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTB_SRC6_WIDTH	RESERVED	PRTB_SRC6_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC6_WIDTH	On Port B. Width of Source ID 6	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC6_HEIGHT	On Port B. Height of Source ID 6	R	0x0

Table 9-157. VIP_OUTPUT_PORT_B_SRC7_SIZE

Address Offset	0x0000 008C		
Physical Address	0x4897 558C 0x4897 5A8C 0x4899 558C 0x4899 5A8C 0x489B 558C 0x489B 5A8C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 7		

Table 9-157. VIP_OUTPUT_PORT_B_SRC7_SIZE (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTB_SRC7_WIDTH											RESERVED					PRTB_SRC7_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC7_WIDTH	On Port B. Width of Source ID 7	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC7_HEIGHT	On Port B. Height of Source ID 7	R	0x0

Table 9-158. VIP_OUTPUT_PORT_B_SRC8_SIZE

Address Offset	0x0000 0090		
Physical Address	0x4897 5590 0x4897 5A90 0x4899 5590 0x4899 5A90 0x489B 5590 0x489B 5A90	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 8		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTB_SRC8_WIDTH											RESERVED					PRTB_SRC8_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC8_WIDTH	On Port B. Width of Source ID 8	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC8_HEIGHT	On Port B. Height of Source ID 8	R	0x0

Table 9-159. VIP_OUTPUT_PORT_B_SRC9_SIZE

Address Offset	0x0000 0094		
Physical Address	0x4897 5594 0x4897 5A94 0x4899 5594 0x4899 5A94 0x489B 5594 0x489B 5A94	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 9		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTB_SRC9_WIDTH											RESERVED					PRTB_SRC9_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC9_WIDTH	On Port B. Width of Source ID 9	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC9_HEIGHT	On Port B. Height of Source ID 9	R	0x0

Table 9-160. VIP_OUTPUT_PORT_B_SRC10_SIZE

Address Offset	0x0000 0098

Table 9-160. VIP_OUTPUT_PORT_B_SRC10_SIZE (continued)

Physical Address	0x4897 5598 0x4897 5A98 0x4899 5598 0x4899 5A98 0x489B 5598 0x489B 5A98	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 10		
Type	R		

31–27	26–16	15–11	10–0
RESERVED	PRTB_SRC10_WIDTH	RESERVED	PRTB_SRC10_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC10_WIDTH	On Port B. Width of Source ID 10	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC10_HEIGHT	On Port B. Height of Source ID 10	R	0x0

Table 9-161. VIP_OUTPUT_PORT_B_SRC11_SIZE

Address Offset	0x0000 009C		
Physical Address	0x4897 559C 0x4897 5A9C 0x4899 559C 0x4899 5A9C 0x489B 559C 0x489B 5A9C	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 11		
Type	R		

31–27	26–16	15–11	10–0
RESERVED	PRTB_SRC11_WIDTH	RESERVED	PRTB_SRC11_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC11_WIDTH	On Port B. Width of Source ID 11	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC11_HEIGHT	On Port B. Height of Source ID 11	R	0x0

Table 9-162. VIP_OUTPUT_PORT_B_SRC12_SIZE

Address Offset	0x0000 00A0		
Physical Address	0x4897 55A0 0x4897 5AA0 0x4899 55A0 0x4899 5AA0 0x489B 55A0 0x489B 5AA0	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 12		
Type	R		

31–27	26–16	15–11	10–0
RESERVED	PRTB_SRC12_WIDTH	RESERVED	PRTB_SRC12_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:16	PRTB_SRC12_WIDTH	On Port B. Width of Source ID 12	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC12_HEIGHT	On Port B. Height of Source ID 12	R	0x0

Table 9-163. VIP_OUTPUT_PORT_B_SRC13_SIZE

Address Offset	0x0000 00A4		
Physical Address	0x4897 55A4 0x4897 5AA4 0x4899 55A4 0x4899 5AA4 0x489B 55A4 0x489B 5AA4	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 13		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTB_SRC13_WIDTH	RESERVED	PRTB_SRC13_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC13_WIDTH	On Port B. Width of Source ID 13	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC13_HEIGHT	On Port B. Height of Source ID 13	R	0x0

Table 9-164. VIP_OUTPUT_PORT_B_SRC14_SIZE

Address Offset	0x0000 00A8		
Physical Address	0x4897 55A8 0x4897 5AA8 0x4899 55A8 0x4899 5AA8 0x489B 55A8 0x489B 5AA8	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 14		
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PRTB_SRC14_WIDTH	RESERVED	PRTB_SRC14_HEIGHT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC14_WIDTH	On Port B. Width of Source ID 14	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC14_HEIGHT	On Port B. Height of Source ID 14	R	0x0

Table 9-165. VIP_OUTPUT_PORT_B_SRC15_SIZE

Address Offset	0x0000 00AC		
Physical Address	0x4897 55AC 0x4897 5AAC 0x4899 55AC 0x4899 5AAC 0x489B 55AC 0x489B 5AAC	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Width and Height for Source 15		

Table 9-165. VIP_OUTPUT_PORT_B_SRC15_SIZE (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PRTB_SRC15_WIDTH											RESERVED					PRTB_SRC15_HEIGHT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC15_WIDTH	On Port B. Width of Source ID 15	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC15_HEIGHT	On Port B. Height of Source ID 15	R	0x0

Table 9-166. VIP_PORT_A_VDET_VEC

Address Offset	0x0000 00B0		
Physical Address	0x4897 55B0 0x4897 5AB0 0x4899 55B0 0x4899 5AB0 0x489B 55B0 0x489B 5AB0	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Each bit represents the VDET bit setting for Line Mux Mode		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTA_VDET_VEC																															

Bits	Field Name	Description	Type	Reset
31:0	PRTA_VDET_VEC	For Embedded Sync Only In Line Mux Mode. each bit represents the vdet value on Port A for the corresponding source id. This vector is meaningless for 1x/2x/4x mux modes.	R	0x0

Table 9-167. VIP_PORT_B_VDET_VEC

Address Offset	0x0000 00B4		
Physical Address	0x4897 55B4 0x4897 5AB4 0x4899 55B4 0x4899 5AB4 0x489B 55B4 0x489B 5AB4	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Each bit represents the VDET bit setting for Line Mux Mode		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTB_VDET_VEC																															

Bits	Field Name	Description	Type	Reset
31:0	PRTB_VDET_VEC	For Embedded Sync Only In Line Mux Mode. each bit represents the vdet value on Port B for the corresponding source id. This vector is meaningless for 1x/2x/4x mux modes.	R	0x0

Table 9-168. VIP_ANC_CROP_HORZ_PORT_A

Address Offset	0x0000 00B8

Table 9-168. VIP_ANC_CROP_HORZ_PORT_A (continued)

Physical Address	0x4897 55B8 0x4897 5AB8 0x4899 55B8 0x4899 5AB8 0x489B 55B8 0x489B 5AB8	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port A		
Type	RW		

31:28	27:16	15	14:12	11:0
ANC_TARGET_SRCNUM	ANC_USE_NUMPIX	ANC_BYPASS_N	RESERVED	ANC_SKIP_NUMPIX

Bits	Field Name	Description	Type	Reset
31:28	ANC_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Ancillary data).	RW	0x0
27:16	ANC_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ANC_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

Table 9-169. VIP_ANC_CROP_VERT_PORT_A

Address Offset	0x0000 00BC		
Physical Address	0x4897 55BC 0x4897 5ABC 0x4899 55BC 0x4899 5ABC 0x489B 55BC 0x489B 5ABC	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port A		
Type	RW		

31:28	27:16	15:12	11:0
RESERVED	ANC_USE_NUMLINES	RESERVED	ANC_SKIP_NUMLINES

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ANC_USE_NUMLINES	When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's ancillary data region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMLINES	The number of lines to crop from the top of the vertical ancillary data region.	RW	0x0

Table 9-170. VIP_CROP_HORZ_PORT_A

Address Offset	0x0000 00C0

Table 9-170. VIP_CROP_HORZ_PORT_A (continued)

Physical Address	0x4897 55C0 0x4897 5AC0 0x4899 55C0 0x4899 5AC0 0x489B 55C0 0x489B 5AC0	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Active Video Cropping Configuration for Input Port A		
Type	RW		

31:28	27:16	15	14:12	11:0
ACT_TARGET_SRCNUM	ACT_USE_NUMPIX	ACT_BYPASS_N	RESERVED	ACT_SKIP_NUMPIX

Bits	Field Name	Description	Type	Reset
31:28	ACT_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Active video).	RW	0x0
27:16	ACT_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ACT_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0
11:0	ACT_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

Table 9-171. VIP_CROP_VERT_PORT_A

Address Offset	0x0000 00C4		
Physical Address	0x4897 55C4 0x4897 5AC4 0x4899 55C4 0x4899 5AC4 0x489B 55C4 0x489B 5AC4	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Active Video Cropping Configuration for Input Port A		
Type	RW		

31:28	27:16	15:12	11:0
RESERVED	ACT_USE_NUMLINES	RESERVED	ACT_SKIP_NUMLINES

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ACT_USE_NUMLINES	When cropping.. the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's active video region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ACT_SKIP_NUMLINES	The number of lines to crop from the top of the vertical active video region.	RW	0x0

Table 9-172. VIP_ANC_VIP_CROP_HORZ_PORT_B

Address Offset	0x0000 00C8

Table 9-172. VIP_ANC_VIP_CROP_HORZ_PORT_B (continued)

Physical Address	0x4897 55C8 0x4897 5AC8 0x4899 55C8 0x4899 5AC8 0x489B 55C8 0x489B 5AC8	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port B		
Type	RW		

31:28	27:16	15	14:12	11:0
ANC_TARGET_SRCNUM	ANC_USE_NUMPIX	ANC_BYPASS_N	RESERVED	ANC_SKIP_NUMPIX

Bits	Field Name	Description	Type	Reset
31:28	ANC_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Ancillary data).	RW	0x0
27:16	ANC_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ANC_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

Table 9-173. VIP_ANC_VIP_CROP_VERT_PORT_B

Address Offset	0x0000 00CC		
Physical Address	0x4897 55CC 0x4897 5ACC 0x4899 55CC 0x4899 5ACC 0x489B 55CC 0x489B 5ACC	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port B		
Type	RW		

31:28	27:16	15:12	11:0
RESERVED	ANC_USE_NUMLINES	RESERVED	ANC_SKIP_NUMLINES

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ANC_USE_NUMLINES	When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnums active video region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMLINES	The number of lines to crop from the top of the vertical ancillary data region.	RW	0x0

Table 9-174. VIP_CROP_HORZ_PORT_B

Address Offset	0x0000 00D0

Table 9-174. VIP_CROP_HORZ_PORT_B (continued)

Physical Address	0x4897 55D0 0x4897 5AD0 0x4899 55D0 0x4899 5AD0 0x489B 55D0 0x489B 5AD0	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Active Video Cropping Configuration for Input Port B		
Type	RW		

31:28	27:16	15	14:12	11:0
ACT_TARGET_SRCNUM	ACT_USE_NUMPIX	ACT_BYPASS_N	RESERVED	ACT_SKIP_NUMPIX

Bits	Field Name	Description	Type	Reset
31:28	ACT_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Active video).	RW	0x0
27:16	ACT_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ACT_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0
11:0	ACT_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

Table 9-175. VIP_CROP_VERT_PORT_B

Address Offset	0x0000 00D4		
Physical Address	0x4897 55D4 0x4897 5AD4 0x4899 55D4 0x4899 5AD4 0x489B 55D4 0x489B 5AD4	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Active Video Cropping Configuration for Input Port B		
Type	RW		

31:28	27:16	15:12	11:0
RESERVED	ACT_USE_NUMLINES	RESERVED	ACT_SKIP_NUMLINES

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ACT_USE_NUMLINES	When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's active video region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ACT_SKIP_NUMLINES	The number of lines to crop from the top of the vertical active video region.	RW	0x0

Table 9-176. VIP_XTRA6_PORT_A

Address Offset	0x0000 00D8

Table 9-176. VIP_XTRA6_PORT_A (continued)

Physical Address	0x4897 55D8 0x4897 5AD8 0x4899 55D8 0x4899 5AD8 0x489B 55D8 0x489B 5AD8	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Cfg Disable Active Srcnum Vector Input for Port A		
Type	RW		

31–16	15–0
YUV_SRCNUM_STOP_IMMEDIATELY	ANC_SRCNUM_STOP_IMMEDIATELY

Bits	Field Name	Description	Type	Reset
31:16	YUV_SRCNUM_STOP_IMMEDIATELY	For the Active Video Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0
15:0	ANC_SRCNUM_STOP_IMMEDIATELY	For the Ancillary Data Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0

Table 9-177. VIP_XTRA7_PORT_B

Address Offset	0x0000 00DC		
Physical Address	0x4897 55DC 0x4897 5ADC 0x4899 55DC 0x4899 5ADC 0x489B 55DC 0x489B 5ADC	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Cfg Disable Active Srcnum Vector Input for Port B		
Type	RW		

31–16	15–0
YUV_SRCNUM_STOP_IMMEDIATELY	ANC_SRCNUM_STOP_IMMEDIATELY

Bits	Field Name	Description	Type	Reset
31:16	YUV_SRCNUM_STOP_IMMEDIATELY	For the Active Video Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0
15:0	ANC_SRCNUM_STOP_IMMEDIATELY	For the Ancillary Data Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0

Table 9-178. VIP_XTRA8_PORT_A

Address Offset	0x0000 00E0		
Physical Address	0x4897 55E0 0x4897 5AE0 0x4899 55E0 0x4899 5AE0 0x489B 55E0 0x489B 5AE0	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Reserved Register for Port A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	ust be 0x0 at all times.	RW	0x0

Table 9-179. VIP_XTRA9_PORT_B

Address Offset	0x0000 00E4		
Physical Address	0x4897 55E4 0x4897 5AE4 0x4899 55E4 0x4899 5AE4 0x489B 55E4 0x489B 5AE4	**Instance**	VIP1_Slice0_parser VIP1_Slice1_parser VIP2_Slice0_parser VIP2_Slice1_parser VIP3_Slice0_parser VIP3_Slice1_parser
Description	Reserved Register for Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	ust be 0x0 at all times.	RW	0x0

9.5.4 VIP CSC Registers

9.5.4.1 VIP CSC Register Summary

Table 9-180. VIP CSC Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_Slice0_csc Base Address	VIP1_Slice1_csc Base Address
VIP_CSC00	RW	32	0x0000 0000	0x4897 5700	0x4897 5C00
VIP_CSC01	RW	32	0x0000 0004	0x4897 5704	0x4897 5C04
VIP_CSC02	RW	32	0x0000 0008	0x4897 5708	0x4897 5C08
VIP_CSC03	RW	32	0x0000 000C	0x4897 570C	0x4897 5C0C
VIP_CSC04	RW	32	0x0000 0010	0x4897 5710	0x4897 5C10
VIP_CSC05	RW	32	0x0000 0014	0x4897 5714	0x4897 5C14

Table 9-181. VIP CSC Registers Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	VIP2_Slice0_csc Base Address	VIP2_Slice1_csc Base Address
VIP_CSC00	RW	32	0x0000 0000	0x4899 5700	0x4899 5C00
VIP_CSC01	RW	32	0x0000 0004	0x4899 5704	0x4899 5C04
VIP_CSC02	RW	32	0x0000 0008	0x4899 5708	0x4899 5C08
VIP_CSC03	RW	32	0x0000 000C	0x4899 570C	0x4899 5C0C
VIP_CSC04	RW	32	0x0000 0010	0x4899 5710	0x4899 5C10
VIP_CSC05	RW	32	0x0000 0014	0x4899 5714	0x4899 5C14

Table 9-182. VIP CSC Registers Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	VIP3_Slice0_csc Base Address	VIP3_Slice1_csc Base Address
VIP_CSC00	RW	32	0x0000 0000	0x489B 5700	0x489B 5C00
VIP_CSC01	RW	32	0x0000 0004	0x489B 5704	0x489B 5C04
VIP_CSC02	RW	32	0x0000 0008	0x489B 5708	0x489B 5C08
VIP_CSC03	RW	32	0x0000 000C	0x489B 570C	0x489B 5C0C
VIP_CSC04	RW	32	0x0000 0010	0x489B 5710	0x489B 5C10
VIP_CSC05	RW	32	0x0000 0014	0x489B 5714	0x489B 5C14

9.5.4.2 VIP CSC Register Description

Table 9-183. VIP_CSC00

Address Offset	0x0000 0000		
Physical Address	0x4897 5700 0x4897 5C00 0x4899 5700 0x4899 5C00 0x489B 5700 0x489B 5C00	**Instance**	VIP1_Slice0_csc VIP1_Slice1_csc VIP2_Slice0_csc VIP2_Slice1_csc VIP3_Slice0_csc VIP3_Slice1_csc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			B0													RESERVED			A0												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	B0	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0
15:13	RESERVED		RW	0x0
12:0	A0	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. Rules for converting a real number coefficient to a 12-bit hex number for this register: - If the real number is positive, then simply multiply it by 1024, and convert the integer part to hex format. For example, 0.673 X 1024 = 689.152, then 0x2B1 should fill in to this register - If the real number is negative, then multiply it by 1024, and convert it to 2's compliment format in 12-bit. For example, if a coefficient is -1.893, by *1024 to this number, it becomes -1938. The 2'S compliment format of -1938 is 0x186E (in 13-bit width). Then 0x186E should be the number assigned to this register.	RW	0x0

Table 9-184. Register Call Summary for Register VIP_CSC00

VIP Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VIP Register Manual

- VIP CSC Register Summary: [10] [12] [13]

Table 9-185. VIP_CSC01

Address Offset	0x0000 0004		
Physical Address	0x4897 5704 0x4897 5C04 0x4899 5704 0x4899 5C04 0x489B 5704 0x489B 5C04	**Instance**	VIP1_Slice0_csc VIP1_Slice1_csc VIP2_Slice0_csc VIP2_Slice1_csc VIP3_Slice0_csc VIP3_Slice1_csc
Description			
Type	RW		

31:29	28:16	15:13	12:0
RESERVED	A1	RESERVED	C0

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	A1	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0
15:13	RESERVED		RW	0x0
12:0	C0	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 9-186. Register Call Summary for Register VIP_CSC01

VIP Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

Table 9-186. Register Call Summary for Register VIP_CSC01 (continued)

VIP Register Manual

- VIP CSC Register Summary: [10] [12] [13]

Table 9-187. VIP_CSC02

Address Offset	0x0000 0008		
Physical Address	0x4897 5708 0x4897 5C08 0x4899 5708 0x4899 5C08 0x489B 5708 0x489B 5C08	**Instance**	VIP1_Slice0_csc VIP1_Slice1_csc VIP2_Slice0_csc VIP2_Slice1_csc VIP3_Slice0_csc VIP3_Slice1_csc
Description			
Type	RW		

31:29	28:16	15:13	12:0
RESERVED	C1	RESERVED	B1

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	C1	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0
15:13	RESERVED		RW	0x0
12:0	B1	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 9-188. Register Call Summary for Register VIP_CSC02

VIP Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VIP Register Manual

- VIP CSC Register Summary: [10] [12] [13]

Table 9-189. VIP_CSC03

Address Offset	0x0000 000C		
Physical Address	0x4897 570C 0x4897 5C0C 0x4899 570C 0x4899 5C0C 0x489B 570C 0x489B 5C0C	**Instance**	VIP1_Slice0_csc VIP1_Slice1_csc VIP2_Slice0_csc VIP2_Slice1_csc VIP3_Slice0_csc VIP3_Slice1_csc
Description			
Type	RW		

31:29	28:16	15:13	12:0
RESERVED	B2	RESERVED	A2

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	B2	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Bits	Field Name	Description	Type	Reset
15:13	RESERVED		RW	0x0
12:0	A2	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 9-190. Register Call Summary for Register VIP_CSC03

VIP Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VIP Register Manual

- VIP CSC Register Summary: [10] [12] [13]

Table 9-191. VIP_CSC04

Address Offset	0x0000 0010		
Physical Address	0x4897 5710 0x4897 5C10 0x4899 5710 0x4899 5C10 0x489B 5710 0x489B 5C10	**Instance**	VIP1_Slice0_csc VIP1_Slice1_csc VIP2_Slice0_csc VIP2_Slice1_csc VIP3_Slice0_csc VIP3_Slice1_csc
Description			
Type	RW		

31:28	27:16	15:13	12:0
RESERVED	D0	RESERVED	C2

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		RW	0x0
27:16	D0	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. For example, if this coefficient is 749, then 0x2ED (hex format) should be assigned to this register. Another example, if this coefficient is -1021, then 0xC03 should be assigned to this register.	RW	0x0
15:13	RESERVED		RW	0x0
12:0	C2	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 9-192. Register Call Summary for Register VIP_CSC04

VIP Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VIP Register Manual

- VIP CSC Register Summary: [10] [12] [13]

Table 9-193. VIP_CSC05

Address Offset	0x0000 0014		
Physical Address	0x4897 5714 0x4897 5C14 0x4899 5714 0x4899 5C14 0x489B 5714 0x489B 5C14	**Instance**	VIP1_Slice0_csc VIP1_Slice1_csc VIP2_Slice0_csc VIP2_Slice1_csc VIP3_Slice0_csc VIP3_Slice1_csc

Table 9-193. VIP_CSC05 (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			BYPASS	D2												RESERVED				D1											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28	BYPASS	Full CSC bypass mode	RW	0x0
27:16	D2	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0)	RW	0x0
15:12	RESERVED		RW	0x0
11:0	D1	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0)	RW	0x0

Table 9-194. Register Call Summary for Register VIP_CSC05

VIP Functional Description
- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
- CSC Bypass Mode: [10]

VIP Register Manual
- VIP CSC Register Summary: [11] [13] [14]

9.5.5 VIP SC registers

9.5.5.1 VIP SC Register Summary

Table 9-195. VIP SC Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_Slice0_sc Base Address	VIP1_Slice1_sc Base Address
VIP_CFG_SC0	RW	32	0x0000 0000	0x4897 5800	0x4897 5D00
VIP_CFG_SC1	RW	32	0x0000 0004	0x4897 5804	0x4897 5D04
VIP_CFG_SC2	RW	32	0x0000 0008	0x4897 5808	0x4897 5D08
VIP_CFG_SC3	RW	32	0x0000 000C	0x4897 580C	0x4897 5D0C
VIP_CFG_SC4	RW	32	0x0000 0010	0x4897 5810	0x4897 5D10
VIP_CFG_SC5	RW	32	0x0000 0014	0x4897 5814	0x4897 5D14
VIP_CFG_SC6	RW	32	0x0000 0018	0x4897 5818	0x4897 5D18
RESERVED	R	32	0x0000 001C	0x4897 581C	0x4897 5D1C
VIP_CFG_SC8	RW	32	0x0000 0020	0x4897 5820	0x4897 5D20
VIP_CFG_SC9	RW	32	0x0000 0024	0x4897 5824	0x4897 5D24
VIP_CFG_SC10	RW	32	0x0000 0028	0x4897 5828	0x4897 5D28
VIP_CFG_SC11	RW	32	0x0000 002C	0x4897 582C	0x4897 5D2C
VIP_CFG_SC12	RW	32	0x0000 0030	0x4897 5830	0x4897 5D30
VIP_CFG_SC13	RW	32	0x0000 0034	0x4897 5834	0x4897 5D34
RESERVED	R	32	0x0000 0038	0x4897 5838	0x4897 5D38
RESERVED	R	32	0x0000 003C	0x4897 583C	0x4897 5D3C
RESERVED	R	32	0x0000 0040	0x4897 5840	0x4897 5D40

Table 9-195. VIP SC Registers Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_Slice0_sc Base Address	VIP1_Slice1_sc Base Address
RESERVED	R	32	0x0000 0044	0x4897 5844	0x4897 5D44
VIP_CFG_SC18	RW	32	0x0000 0048	0x4897 5848	0x4897 5D48
VIP_CFG_SC19	RW	32	0x0000 004C	0x4897 584C	0x4897 5D4C
VIP_CFG_SC20	RW	32	0x0000 0050	0x4897 5850	0x4897 5D50
VIP_CFG_SC21	RW	32	0x0000 0054	0x4897 5854	0x4897 5D54
VIP_CFG_SC22	RW	32	0x0000 0058	0x4897 5858	0x4897 5D58
RESERVED	R	32	0x0000 005C	0x4897 585C	0x4897 5D5C
VIP_CFG_SC24	RW	32	0x0000 0060	0x4897 5860	0x4897 5D60
VIP_CFG_SC25	RW	32	0x0000 0064	0x4897 5864	0x4897 5D64
RESERVED	R	32	0x0000 0068	0x4897 5868	0x4897 5D68
RESERVED	R	32	0x0000 006C	0x4897 586C	0x4897 5D6C
RESERVED	R	32	0x0000 0070	0x4897 5870	0x4897 5D70
RESERVED	R	32	0x0000 0074	0x4897 5874	0x4897 5D74
RESERVED	R	32	0x0000 0078	0x4897 5878	0x4897 5D78
RESERVED	R	32	0x0000 007C	0x4897 587C	0x4897 5D7C

Table 9-196. VIP SC Registers Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	VIP2_Slice0_sc Base Address	VIP2_Slice1_sc Base Address
VIP_CFG_SC0	RW	32	0x0000 0000	0x4899 5800	0x4899 5D00
VIP_CFG_SC1	RW	32	0x0000 0004	0x4899 5804	0x4899 5D04
VIP_CFG_SC2	RW	32	0x0000 0008	0x4899 5808	0x4899 5D08
VIP_CFG_SC3	RW	32	0x0000 000C	0x4899 580C	0x4899 5D0C
VIP_CFG_SC4	RW	32	0x0000 0010	0x4899 5810	0x4899 5D10
VIP_CFG_SC5	RW	32	0x0000 0014	0x4899 5814	0x4899 5D14
VIP_CFG_SC6	RW	32	0x0000 0018	0x4899 5818	0x4899 5D18
RESERVED	R	32	0x0000 001C	0x4899 581C	0x4899 5D1C
VIP_CFG_SC8	RW	32	0x0000 0020	0x4899 5820	0x4899 5D20
VIP_CFG_SC9	RW	32	0x0000 0024	0x4899 5824	0x4899 5D24
VIP_CFG_SC10	RW	32	0x0000 0028	0x4899 5828	0x4899 5D28
VIP_CFG_SC11	RW	32	0x0000 002C	0x4899 582C	0x4899 5D2C
VIP_CFG_SC12	RW	32	0x0000 0030	0x4899 5830	0x4899 5D30
VIP_CFG_SC13	RW	32	0x0000 0034	0x4899 5834	0x4899 5D34
RESERVED	R	32	0x0000 0038	0x4899 5838	0x4899 5D38
RESERVED	R	32	0x0000 003C	0x4899 583C	0x4899 5D3C
RESERVED	R	32	0x0000 0040	0x4899 5840	0x4899 5D40
RESERVED	R	32	0x0000 0044	0x4899 5844	0x4899 5D44
VIP_CFG_SC18	RW	32	0x0000 0048	0x4899 5848	0x4899 5D48
VIP_CFG_SC19	RW	32	0x0000 004C	0x4899 584C	0x4899 5D4C
VIP_CFG_SC20	RW	32	0x0000 0050	0x4899 5850	0x4899 5D50
VIP_CFG_SC21	RW	32	0x0000 0054	0x4899 5854	0x4899 5D54
VIP_CFG_SC22	RW	32	0x0000 0058	0x4899 5858	0x4899 5D58
RESERVED	R	32	0x0000 005C	0x4899 585C	0x4899 5D5C
VIP_CFG_SC24	RW	32	0x0000 0060	0x4899 5860	0x4899 5D60
VIP_CFG_SC25	RW	32	0x0000 0064	0x4899 5864	0x4899 5D64
RESERVED	R	32	0x0000 0068	0x4899 5868	0x4899 5D68

Table 9-196. VIP SC Registers Mapping Summary 2 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP2_Slice0_sc Base Address	VIP2_Slice1_sc Base Address
RESERVED	R	32	0x0000 006C	0x4899 586C	0x4899 5D6C
RESERVED	R	32	0x0000 0070	0x4899 5870	0x4899 5D70
RESERVED	R	32	0x0000 0074	0x4899 5874	0x4899 5D74
RESERVED	R	32	0x0000 0078	0x4899 5878	0x4899 5D78
RESERVED	R	32	0x0000 007C	0x4899 587C	0x4899 5D7C

Table 9-197. VIP SC Registers Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	VIP3_Slice0_sc Base Address	VIP3_Slice1_sc Base Address
VIP_CFG_SC0	RW	32	0x0000 0000	0x489B 5800	0x489B 5D00
VIP_CFG_SC1	RW	32	0x0000 0004	0x489B 5804	0x489B 5D04
VIP_CFG_SC2	RW	32	0x0000 0008	0x489B 5808	0x489B 5D08
VIP_CFG_SC3	RW	32	0x0000 000C	0x489B 580C	0x489B 5D0C
VIP_CFG_SC4	RW	32	0x0000 0010	0x489B 5810	0x489B 5D10
VIP_CFG_SC5	RW	32	0x0000 0014	0x489B 5814	0x489B 5D14
VIP_CFG_SC6	RW	32	0x0000 0018	0x489B 5818	0x489B 5D18
RESERVED	R	32	0x0000 001C	0x489B 581C	0x489B 5D1C
VIP_CFG_SC8	RW	32	0x0000 0020	0x489B 5820	0x489B 5D20
VIP_CFG_SC9	RW	32	0x0000 0024	0x489B 5824	0x489B 5D24
VIP_CFG_SC10	RW	32	0x0000 0028	0x489B 5828	0x489B 5D28
VIP_CFG_SC11	RW	32	0x0000 002C	0x489B 582C	0x489B 5D2C
VIP_CFG_SC12	RW	32	0x0000 0030	0x489B 5830	0x489B 5D30
VIP_CFG_SC13	RW	32	0x0000 0034	0x489B 5834	0x489B 5D34
RESERVED	R	32	0x0000 0038	0x489B 5838	0x489B 5D38
RESERVED	R	32	0x0000 003C	0x489B 583C	0x489B 5D3C
RESERVED	R	32	0x0000 0040	0x489B 5840	0x489B 5D40
RESERVED	R	32	0x0000 0044	0x489B 5844	0x489B 5D44
VIP_CFG_SC18	RW	32	0x0000 0048	0x489B 5848	0x489B 5D48
VIP_CFG_SC19	RW	32	0x0000 004C	0x489B 584C	0x489B 5D4C
VIP_CFG_SC20	RW	32	0x0000 0050	0x489B 5850	0x489B 5D50
VIP_CFG_SC21	RW	32	0x0000 0054	0x489B 5854	0x489B 5D54
VIP_CFG_SC22	RW	32	0x0000 0058	0x489B 5858	0x489B 5D58
RESERVED	R	32	0x0000 005C	0x489B 585C	0x489B 5D5C
VIP_CFG_SC24	RW	32	0x0000 0060	0x489B 5860	0x489B 5D60
VIP_CFG_SC25	RW	32	0x0000 0064	0x489B 5864	0x489B 5D64
RESERVED	R	32	0x0000 0068	0x489B 5868	0x489B 5D68
RESERVED	R	32	0x0000 006C	0x489B 586C	0x489B 5D6C
RESERVED	R	32	0x0000 0070	0x489B 5870	0x489B 5D70
RESERVED	R	32	0x0000 0074	0x489B 5874	0x489B 5D74
RESERVED	R	32	0x0000 0078	0x489B 5878	0x489B 5D78
RESERVED	R	32	0x0000 007C	0x489B 587C	0x489B 5D7C

9.5.5.2 VIP SC Register Description

Table 9-198. VIP_CFG_SC0

Address Offset	0x0000 0000

Table 9-198. VIP_CFG_SC0 (continued)

Physical Address	0x4897 5800 0x4897 5D00 0x4899 5800 0x4899 5D00 0x489B 5800 0x489B 5D00	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															CFG_FID_SELFGEN	CFG_TRIM	CFG_Y_PK_EN	RESERVED			CFG_INTERLACE_I	CFG_HP_BYPASS	CFG_DCM_4X	CFG_DCM_2X	CFG_AUTO_HS	CFG_ENABLE_EV	CFG_USE_RAV	CFG_INVT_FID	CFG_SC_BYPASS	CFG_LINEAR	CFG_INTERLACE_O

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CFG_FID_SELFGEN	FID self generate enable. When input is progressive and this bit is set, the SC generates self-toggling (top/bottom) output FID when performing interlacing.	RW	0x0
15	CFG_TRIM	Trimming enable. When 1, the input image whose size is specified by orgW and orgH registers is trimmed to the size with srcW and srcH from the offset specified by offW and offH. 0: disable trimming 1: enable trimming	RW	0x0
14	CFG_Y_PK_EN	This parameter is used by peaking block. 0: disable luma peaking 1: enable luma peaking	RW	0x0
13:11	RESERVED		R	0x0
10	CFG_INTERLACE_I	This parameter is used by both horizontal and vertical scaling 0: the input video format is progressive 1: the input video format is interlace	RW	0x0
9	CFG_HP_BYPASS	This parameter is used by horizontal scaling. If cfg_auto_hs is 0, horizontal polyphase filter is always enabled. In this case, this register is DON?T CARE. If cfg_auto_hs is 1, 0 : The polyphase scaler is always used regardless of the scaling ratio. 1 : The polyphase scaler is bypassed only when (tar_w == src_w) or (tar_w == src_w/2) or (tar_w == src_w/4)	RW	0x0
8	CFG_DCM_4X	This parameter is used by horizontal scaling. 0: the 4X decimation filter is disabled 1: the 4X decimation filter is enabled Note: (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (horizontal scale ratio 0.25). (3) This register is DON?T CARE when cfg_auto_hs = 1.	RW	0x0
7	CFG_DCM_2X	This parameter is used by horizontal scaling. 0: the 2X decimation filter is disabled 1: the 2X decimation filter is enabled Note: (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (0.25 horizontal scale ratio 0.5). (3) This register is DON?T CARE when cfg_auto_hs = 1.	RW	0x0

Bits	Field Name	Description	Type	Reset
6	CFG_AUTO_HS	This parameter is used by horizontal scaling. 0: the cfg_dcm_2x and cfg_dcm_4x bits will enable appropriate decimation filters 1: HW will decide whether up-scaling or down-scaling is required based on horizontal scaling ratio (SR). SR 0.5 : horizontal polyphase filter is enabled, all decimation filters are disabled SR = 0.5 : dcm_2x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.5 SR 0.25 : dcm_2x and horizontal polyphase filter both are enabled SR = 0.25 : dcm_4x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.25 SR 0.125 : dcm_4x and horizontal polyphase filter are both enabled SR = 0.125 : Functionally supported, but not recommended in auto mode for image quality concerns	RW	0x0
5	CFG_ENABLE_EV	This parameter is used by the edge-detection block. 0: The output of edge-detection block will be force to ?0? 1: The calculation results of edge-detection block will be output normally	RW	0x0
4	CFG_USE_RAV	This parameter is used by vertical scaling. 0: Poly-phase filter will be used for the vertical scaling 1: Running average filter will be used for the vertical scaling (down scaling only)	RW	0x0
3	CFG_INVT_FID	This parameter is used by vertical scaling. 0: Progressive input 1: Interlaced input Must be set to 1 when CFG_INTERFACE_I = 1.	RW	0x0
2	CFG_SC_BYPASS	This parameter is a general purpose. 0: Scaling module will engaged 1: Scaling module will be bypassed	RW	0x0
1	CFG_LINEAR	This parameter is used by horizontal scaling. 0: Anamorphic scaling 1: Linear scaling	RW	0x0
0	CFG_INTERLACE_O	This parameter is used by vertical scaling. 0: The output format of SC is progressive 1: The output format of SC is interlace	RW	0x0

Table 9-199. Register Call Summary for Register VIP_CFG_SC0

VIP Functional Description

- SC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]
- SC Code: [14] [15] [16] [17] [18] [19] [20] [21] [22]

VIP Register Manual

- VIP SC Register Summary: [23] [25] [26]

Table 9-200. VIP_CFG_SC1

Address Offset	0x0000 0004		
Physical Address	0x4897 5804 0x4897 5D04 0x4899 5804 0x4899 5D04 0x489B 5804 0x489B 5D04	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_ROW_ACC_INC																										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:0	CFG_ROW_ACC_INC	This parameter is used by vertical scaling. It defines the increment of the row accumulator in vertical poly-phase filter. It can be calculated by following formula: row_acc_inc = round(2^16 *(src_h)/(tar_h)) In case of interlaced input, srcH is input field height In case of interlaced output, tarH is output field height.	RW	0x0

Table 9-201. Register Call Summary for Register VIP_CFG_SC1

VIP Functional Description

- SC Functional Description: [0]
- SC Code:

VIP Register Manual

- VIP SC Register Summary: [7] [9] [10]

Table 9-202. VIP_CFG_SC2

Address Offset	0x0000 0008		
Physical Address	0x4897 5808 0x4897 5D08 0x4899 5808 0x4899 5D08 0x489B 5808 0x489B 5D08	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CFG_ROW_ACC_OFFSET																											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this offset will be applied to a frame. In interlace mode: this offset will be applied to the top field.	RW	0x0

Table 9-203. Register Call Summary for Register VIP_CFG_SC2

VIP Functional Description

- SC Functional Description: [0]
- SC Code:

VIP Register Manual

- VIP SC Register Summary: [2] [4] [5]

Table 9-204. VIP_CFG_SC3

Address Offset	0x0000 000C		
Physical Address	0x4897 580C 0x4897 5D0C 0x4899 580C 0x4899 5D0C 0x489B 580C 0x489B 5D0C	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CFG_ROW_ACC_OFFSET_B

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET_B	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this parameter will not be used. In interlace mode: this offset will be applied to the bottom field.	RW	0x0

Table 9-205. Register Call Summary for Register VIP_CFG_SC3

VIP Functional Description

- SC Functional Description: [0]

VIP Register Manual

- VIP SC Register Summary: [1] [3] [4]

Table 9-206. VIP_CFG_SC4

Address Offset	0x0000 0010		
Physical Address	0x4897 5810 0x4897 5D10 0x4899 5810 0x4899 5D10 0x489B 5810 0x489B 5D10	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CFG_NLIN_ACC_INIT_U			RESERVED	CFG_LIN_ACC_INC_U			RESERVED	CFG_TAR_W											RESERVED	CFG_TAR_H										

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	CFG_NLIN_ACC_INIT_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_init' that is defined in CFG_SC10	RW	0x0
27	RESERVED		R	0x0
26:24	CFG_LIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'lin_acc_inc' that is defined in CFG_SC9	RW	0x0
23	RESERVED		R	0x0
22:12	CFG_TAR_W	This parameter is a general purpose. Scaled target picture width. unit is pixel. This parameter defines the final output picture size	RW	0x0
11	RESERVED		R	0x0
10:0	CFG_TAR_H	This parameter is a general purpose. Scaled target picture height.. unit is line... This parameter defines the final output picture size. For the interlace output.. it should be the number of lines per field.	RW	0x0

Table 9-207. Register Call Summary for Register VIP_CFG_SC4

VIP Functional Description

- SC Functional Description: [0] [1] [2] [3] [4] [5] [6]

Table 9-207. Register Call Summary for Register VIP_CFG_SC4 (continued)

VIP Register Manual
- VIP SC Register Summary: [7] [9] [10]

Table 9-208. VIP_CFG_SC5

Address Offset	0x0000 0014		
Physical Address	0x4897 5814 0x4897 5D14 0x4899 5814 0x4899 5D14 0x489B 5814 0x489B 5D14	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31:27	26:24	23	22:12	11	10:0
RESERVED	CFG_NLIN_ACC_INC_U	RESERVED	CFG_SRC_W	RESERVED	CFG_SRC_H

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	CFG_NLIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of ?nlin_acc_inc? that is defined in CFG_SC11	RW	0x0
23	RESERVED		R	0x0
22:12	CFG_SRC_W	This parameter is a general purpose. This parameter defines the width of the source image	RW	0x0
11	RESERVED		R	0x0
10:0	CFG_SRC_H	This parameter is a general purpose. This parameter defines the height of the source image. For the interlace input.. it should be the number of lines per field.	RW	0x0

Table 9-209. Register Call Summary for Register VIP_CFG_SC5

VIP Functional Description
- SC Functional Description: [0] [1] [2] [3] [4]

VIP Register Manual
- VIP SC Register Summary: [5] [7] [8]

Table 9-210. VIP_CFG_SC6

Address Offset	0x0000 0018		
Physical Address	0x4897 5818 0x4897 5D18 0x4899 5818 0x4899 5D18 0x489B 5818 0x489B 5D18	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31:20	19:10	9:0
RESERVED	CFG_ROW_ACC_INIT_RAV_B	CFG_ROW_ACC_INIT_RAV

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19:10	CFG_ROW_ACC_INIT_RAV_B	This parameter is used by vertical scaling. it is used only when the input is interlace format. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for bottom field of interlace format)	RW	0x0
9:0	CFG_ROW_ACC_INIT_RAV	This parameter is used by vertical scaling. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for progressive format or top field of interlace format)	RW	0x0

Table 9-211. Register Call Summary for Register VIP_CFG_SC6

VIP Functional Description

- SC Functional Description: [0] [1] [2] [3]

VIP Register Manual

- VIP SC Register Summary: [4] [6] [7]

Table 9-212. VIP_CFG_SC8

Address Offset	0x0000 0020		
Physical Address	0x4897 5820 0x4897 5D20 0x4899 5820 0x4899 5D20 0x489B 5820 0x489B 5D20	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31:23	22:12	11	10:0
RESERVED	CFG_NLIN_RIGHT	RESERVED	CFG_NLIN_LEFT

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:12	CFG_NLIN_RIGHT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on right-hand side. In other words. it defines the location of the last pixel where the linear scaling is ended. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0
11	RESERVED		R	0x0
10:0	CFG_NLIN_LEFT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on left-hand side. In other words. it defines the location of the last pixel in the left-sidenonlinear strip. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0

Table 9-213. Register Call Summary for Register VIP_CFG_SC8

VIP Functional Description

- SC Functional Description: [0] [1]

Table 9-213. Register Call Summary for Register VIP_CFG_SC8 (continued)

VIP Register Manual

- VIP SC Register Summary: [2] [4] [5]

Table 9-214. VIP_CFG_SC9

Address Offset	0x0000 0024		
Physical Address	0x4897 5824 0x4897 5D24 0x4899 5824 0x4899 5D24 0x489B 5824 0x489B 5D24	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31–0
CFG_LIN_ACC_INC

Bits	Field Name	Description	Type	Reset
31:0	CFG_LIN_ACC_INC	This parameter is used by horizontal scaling. It defines the increment of the linear accumulator. if SR 0.5 then lin_acc_inc = round(2^24*(srcWi -1) /(tarWi -1)) else if 0.25 SR ? 0.5 lin_acc_inc = round(2^24*(srcWi/2 -1) /(tarWi - 1)) else if SR ? 0.25 lin_acc_inc = round(2^24*(srcWi/4 -1) /(tarWi - 1)) where srcWi and tarWi are the inner source width and the inner target width respectively.	RW	0x0

Table 9-215. Register Call Summary for Register VIP_CFG_SC9

VIP Functional Description

- SC Functional Description: [0] [1]

VIP Register Manual

- VIP SC Register Summary: [2] [4] [5]

Table 9-216. VIP_CFG_SC10

Address Offset	0x0000 0028		
Physical Address	0x4897 5828 0x4897 5D28 0x4899 5828 0x4899 5D28 0x489B 5828 0x489B 5D28	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31–0
CFG_NLIN_ACC_INIT

Bits	Field Name	Description	Type	Reset
31:0	CFG_NLIN_ACC_INIT	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the initialization value of the nonlinear accumulator. nlin_acc_init = K*(1-2*d) Here the definitions of K and d are the same as in CFG_SC11	RW	0x0

Table 9-217. Register Call Summary for Register VIP_CFG_SC10

VIP Register Manual

- VIP SC Register Summary: [0] [2] [3]

Table 9-218. VIP_CFG_SC11

Address Offset	0x0000 002C		
Physical Address	0x4897 582C 0x4897 5D2C 0x4899 582C 0x4899 5D2C 0x489B 582C 0x489B 5D2C	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_NLIN_ACC_INC																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_NLIN_ACC_INC	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the increment of the nonlinear accumulator. if upscaling then d = 0 if Ltar !=0 then K =round[2^24*Lsrc/(Ltar*Ltar)] where Lsrc= (srcW-srcWi)/2 else K = 0 elseif downscaling d = (tarW-1)/2 if Ltar!=0 then K = round[2^24 * Lsrc / (Ltar*(Ltar-2d))] where Lsrc= (srcW-srcWi)/(2n) and n=1..2 or 4 else K = 0 nlin_acc_inc = 2*K (negative for downscaling)	RW	0x0

Table 9-219. Register Call Summary for Register VIP_CFG_SC11

VIP Register Manual

- VIP SC Register Summary: [0] [2] [3]

Table 9-220. VIP_CFG_SC12

Address Offset	0x0000 0030		
Physical Address	0x4897 5830 0x4897 5D30 0x4899 5830 0x4899 5D30 0x489B 5830 0x489B 5D30	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CFG_COL_ACC_OFFSET																								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	CFG_COL_ACC_OFFSET	This parameter is used in horizontal scaling. It defines the luma accumulator's offset. Normally this parameter can be set as 0 if no horizontal offset is involved. In some applications.. such as Pan and Scan.. a corresponding offset value should be set. The format is 1.24.	RW	0x0

Table 9-221. Register Call Summary for Register VIP_CFG_SC12

VIP Functional Description

- SC Code: [0]

VIP Register Manual

- VIP SC Register Summary: [1] [3] [4]

Table 9-222. VIP_CFG_SC13

Address Offset	0x0000 0034		
Physical Address	0x4897 5834 0x4897 5D34 0x4899 5834 0x4899 5D34 0x489B 5834 0x489B 5D34	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31:10	9:0
RESERVED	CFG_SC_FACTOR_RAV

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_SC_FACTOR_RAV	This parameter is used by vertical scaling. Vertical scaling factor: It is defined as following: 1024*tarH/srcH. It is used for downscaling by the running average filter	RW	0x0

Table 9-223. Register Call Summary for Register VIP_CFG_SC13

VIP Functional Description

- SC Functional Description: [0] [1] [2]
- SC Code: [3] [4]

VIP Register Manual

- VIP SC Register Summary: [5] [7] [8]

Table 9-224. VIP_CFG_SC18

Address Offset	0x0000 0048		
Physical Address	0x4897 5848 0x4897 5D48 0x4899 5848 0x4899 5D48 0x489B 5848 0x489B 5D48	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31:10	9:0
RESERVED	CFG_HS_FACTOR

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_HS_FACTOR	This parameter is used by horizontal scaling. Horizontal-scaling-factor = tarWi/srcWi. Numerical format: 6.4 (6 bit integer and 4 bit fraction)	RW	0x0

Table 9-225. Register Call Summary for Register VIP_CFG_SC18

VIP Functional Description
- SC Code: [0]

VIP Register Manual
- VIP SC Register Summary: [1] [3] [4]

Table 9-226. VIP_CFG_SC19

Address Offset	0x0000 004C		
Physical Address	0x4897 584C 0x4897 5D4C 0x4899 584C 0x4899 5D4C 0x489B 584C 0x489B 5D4C	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CFG_HPF_COEF3	CFG_HPF_COEF2	CFG_HPF_COEF1	CFG_HPF_COEF0

Bits	Field Name	Description	Type	Reset
31:24	CFG_HPF_COEF3	This parameter is used by the peaking block. Defines the coefficient 3 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
23:16	CFG_HPF_COEF2	This parameter is used by the peaking block. Defines the coefficient 2 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
15:8	CFG_HPF_COEF1	This parameter is used by the peaking block. Defines the coefficient 1 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF0	This parameter is used by the peaking block. Defines the coefficient 0 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Table 9-227. Register Call Summary for Register VIP_CFG_SC19

VIP Functional Description
- SC Functional Description: [0] [1]
- SC Code: [2]

VIP Register Manual
- VIP SC Register Summary: [3] [5] [6]

Table 9-228. VIP_CFG_SC20

Address Offset	0x0000 0050		
Physical Address	0x4897 5850 0x4897 5D50 0x4899 5850 0x4899 5D50 0x489B 5850 0x489B 5D50	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0

RESERVED	CFG_NL_LIMIT	RESERVED	CFG_HPF_NORM_SHIFT	CFG_HPF_COEF5	CFG_HPF_COEF4

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:20	CFG_NL_LIMIT	This parameter is used by the peaking block. The maximum of clipping.	RW	0x0
19	RESERVED		R	0x0
18:16	CFG_HPF_NORM_SHIFT	This parameter is used by the peaking block. Defines the decimal point of the hpf coefficient.	RW	0x0
15:8	CFG_HPF_COEF5	This parameter is used by the peaking block. Defines the coefficient 5 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF4	This parameter is used by the peaking block. Defines the coefficient 4 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Table 9-229. Register Call Summary for Register VIP_CFG_SC20

VIP Functional Description

- SC Functional Description: [0] [1] [2]
- SC Code: [3]

VIP Register Manual

- VIP SC Register Summary: [4] [6] [7]

Table 9-230. VIP_CFG_SC21

Address Offset	0x0000 0054		
Physical Address	0x4897 5854 0x4897 5D54 0x4899 5854 0x4899 5D54 0x489B 5854 0x489B 5D54	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_NL_LO_SLOPE								RESERVED							CFG_NL_LO_THR								

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	CFG_NL_LO_SLOPE	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The format is fixed point 4.4.	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_LO_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be 0	RW	0x0

Table 9-231. Register Call Summary for Register VIP_CFG_SC21

VIP Functional Description

- SC Functional Description: [0] [1]
- SC Code: [2]

Table 9-231. Register Call Summary for Register VIP_CFG_SC21 (continued)

VIP Register Manual

- VIP SC Register Summary: [3] [5] [6]

Table 9-232. VIP_CFG_SC22

Address Offset	0x0000 0058		
Physical Address	0x4897 5858 0x4897 5D58 0x4899 5858 0x4899 5D58 0x489B 5858 0x489B 5D58	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31:19	18:16	15:9	8:0
RESERVED	CFG_NL_HI_SLOPE_SHIFT	RESERVED	CFG_NL_HI_THR

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:16	CFG_NL_HI_SLOPE_SHIFT	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The gain is 2^(nl_hi_slope_shift-3).	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_HI_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be nl_hi_thr.	RW	0x0

Table 9-233. Register Call Summary for Register VIP_CFG_SC22

VIP Functional Description

- SC Functional Description: [0] [1] [2]
- SC Code: [3]

VIP Register Manual

- VIP SC Register Summary: [4] [6] [7]

Table 9-234. VIP_CFG_SC24

Address Offset	0x0000 0060		
Physical Address	0x4897 5860 0x4897 5D60 0x4899 5860 0x4899 5D60 0x489B 5860 0x489B 5D60	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	CFG_ORG_W	RESERVED	CFG_ORG_H

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_ORG_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:11	RESERVED		R	0x0
10:0	CFG_ORG_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 9-235. Register Call Summary for Register VIP_CFG_SC24

VIP Functional Description

- SC Functional Description: [0] [1]

VIP Register Manual

- VIP SC Register Summary: [2] [4] [5]

Table 9-236. VIP_CFG_SC25

Address Offset	0x0000 0064		
Physical Address	0x4897 5864 0x4897 5D64 0x4899 5864 0x4899 5D64 0x489B 5864 0x489B 5D64	**Instance**	VIP1_Slice0_sc VIP1_Slice1_sc VIP2_Slice0_sc VIP2_Slice1_sc VIP3_Slice0_sc VIP3_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_OFF_W											RESERVED					CFG_OFF_H										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_OFF_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0
15:11	RESERVED		R	0x0
10:0	CFG_OFF_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 9-237. Register Call Summary for Register VIP_CFG_SC25

VIP Functional Description

- SC Functional Description: [0] [1]

VIP Register Manual

- VIP SC Register Summary: [2] [4] [5]

9.5.6 VIP VPDMA Registers

Note

The functionality of the following sets of registers is not supported by VIP VPDMA in this family of devices:

- All VIP_INT2_* registers
- All VIP_INT3_* registers

The following channels are not used by VIP VPDMA in this family of devices. All register bit-fields corresponding to these channels should be kept at their reset value.

- HQ_*
- GRPX_*
- SCALER_OUT
- SCALER_LUMA
- SCALER_CHROMA
- NF_*
- TRANSCODE1_*
- TRANSCODE2_*
- AUX_IN
- PIP_FRAME
- POST_COMP_WR
- VBI_SD_VENC

The following clients are not used by VIP VPDMA in this family of devices. All register bit-fields corresponding to these clients should be kept at their reset value.

- DEI_HQ_*
- TRANS1_LUMA
- TRANS1_CHROMA
- TRANS2_LUMA
- TRANS2_CHROMA
- HDMI_WRBK
- VBI_SDVENC
- NF_420_UV_OUT
- NF_420_Y_OUT
- NF_420_UV_IN
- NF_420_Y_IN
- NF_422_IN
- GRPX1_ST
- GRPX2_ST
- GRPX3_ST
- GRPX1_DATA
- GRPX2_DATA
- GRPX3_DATA
- PIP_WRBK
- SC_IN_*
- SC_OUT
- COMP_WRBK

9.5.6.1 VIP VPDMA Register Summary

Table 9-238. VIP VPDMA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_VPDMA Base Address	VIP2_VPDMA Base Address	VIP3_VPDMA Base Address
VIP_PID	R	32	0x0000 0000	0x4897 D000	0x4899 D000	0x489B D000
VIP_LIST_ADDR	RW	32	0x0000 0004	0x4897 D004	0x4899 D004	0x489B D004

Table 9-238. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_VPDMA Base Address	VIP2_VPDMA Base Address	VIP3_VPDMA Base Address
VIP_LIST_ATTR	RW	32	0x0000 0008	0x4897 D008	0x4899 D008	0x489B D008
VIP_LIST_STAT_SYNC	RW	32	0x0000 000C	0x4897 D00C	0x4899 D00C	0x489B D00C
VIP_BG_RGB	RW	32	0x0000 0018	0x4897 D018	0x4899 D018	0x489B D018
VIP_BG_YUV	RW	32	0x0000 001C	0x4897 D01C	0x4899 D01C	0x489B D01C
VIP_VPDMA_SETUP	RW	32	0x0000 0030	0x4897 D030	0x4899 D030	0x489B D030
VIP_MAX_SIZE1	RW	32	0x0000 0034	0x4897 D034	0x4899 D034	0x489B D034
VIP_MAX_SIZE2	RW	32	0x0000 0038	0x4897 D038	0x4899 D038	0x489B D038
VIP_MAX_SIZE3	RW	32	0x0000 003C	0x4897 D03C	0x4899 D03C	0x489B D03C
VIP_INT0_CHANNEL0_INT_STAT	RW	32	0x0000 0040	0x4897 D040	0x4899 D040	0x489B D040
VIP_INT0_CHANNEL0_INT_MASK	RW	32	0x0000 0044	0x4897 D044	0x4899 D044	0x489B D044
VIP_INT0_CHANNEL1_INT_STAT	RW	32	0x0000 0048	0x4897 D048	0x4899 D048	0x489B D048
VIP_INT0_CHANNEL1_INT_MASK	RW	32	0x0000 004C	0x4897 D04C	0x4899 D04C	0x489B D04C
VIP_INT0_CHANNEL2_INT_STAT	RW	32	0x0000 0050	0x4897 D050	0x4899 D050	0x489B D050
VIP_INT0_CHANNEL2_INT_MASK	RW	32	0x0000 0054	0x4897 D054	0x4899 D054	0x489B D054
VIP_INT0_CHANNEL3_INT_STAT	RW	32	0x0000 0058	0x4897 D058	0x4899 D058	0x489B D058
VIP_INT0_CHANNEL3_INT_MASK	RW	32	0x0000 005C	0x4897 D05C	0x4899 D05C	0x489B D05C
VIP_INT0_CHANNEL4_INT_STAT	RW	32	0x0000 0060	0x4897 D060	0x4899 D060	0x489B D060
VIP_INT0_CHANNEL4_INT_MASK	RW	32	0x0000 0064	0x4897 D064	0x4899 D064	0x489B D064
VIP_INT0_CHANNEL5_INT_STAT	RW	32	0x0000 0068	0x4897 D068	0x4899 D068	0x489B D068
VIP_INT0_CHANNEL5_INT_MASK	RW	32	0x0000 006C	0x4897 D06C	0x4899 D06C	0x489B D06C
VIP_INT0_CLIENT0_INT_STAT	RW	32	0x0000 0078	0x4897 D078	0x4899 D078	0x489B D078
VIP_INT0_CLIENT0_INT_MASK	RW	32	0x0000 007C	0x4897 D07C	0x4899 D07C	0x489B D07C
VIP_INT0_CLIENT1_INT_STAT	RW	32	0x0000 0080	0x4897 D080	0x4899 D080	0x489B D080
VIP_INT0_CLIENT1_INT_MASK	RW	32	0x0000 0084	0x4897 D084	0x4899 D084	0x489B D084
VIP_INT0_LIST0_INT_STAT	RW	32	0x0000 0088	0x4897 D088	0x4899 D088	0x489B D088
VIP_INT0_LIST0_INT_MASK	RW	32	0x0000 008C	0x4897 D08C	0x4899 D08C	0x489B D08C
VIP_INT1_CHANNEL0_INT_STAT	RW	32	0x0000 0090	0x4897 D090	0x4899 D090	0x489B D090
VIP_INT1_CHANNEL0_INT_MASK	RW	32	0x0000 0094	0x4897 D094	0x4899 D094	0x489B D094
VIP_INT1_CHANNEL1_INT_STAT	RW	32	0x0000 0098	0x4897 D098	0x4899 D098	0x489B D098
VIP_INT1_CHANNEL1_INT_MASK	RW	32	0x0000 009C	0x4897 D09C	0x4899 D09C	0x489B D09C
VIP_INT1_CHANNEL2_INT_STAT	RW	32	0x0000 00A0	0x4897 D0A0	0x4899 D0A0	0x489B D0A0
VIP_INT1_CHANNEL2_INT_MASK	RW	32	0x0000 00A4	0x4897 D0A4	0x4899 D0A4	0x489B D0A4
VIP_INT1_CHANNEL3_INT_STAT	RW	32	0x0000 00A8	0x4897 D0A8	0x4899 D0A8	0x489B D0A8
VIP_INT1_CHANNEL3_INT_MASK	RW	32	0x0000 00AC	0x4897 D0AC	0x4899 D0AC	0x489B D0AC
VIP_INT1_CHANNEL4_INT_STAT	RW	32	0x0000 00B0	0x4897 D0B0	0x4899 D0B0	0x489B D0B0
VIP_INT1_CHANNEL4_INT_MASK	RW	32	0x0000 00B4	0x4897 D0B4	0x4899 D0B4	0x489B D0B4
VIP_INT1_CHANNEL5_INT_STAT	RW	32	0x0000 00B8	0x4897 D0B8	0x4899 D0B8	0x489B D0B8
VIP_INT1_CHANNEL5_INT_MASK	RW	32	0x0000 00BC	0x4897 D0BC	0x4899 D0BC	0x489B D0BC
VIP_INT1_CLIENT0_INT_STAT	RW	32	0x0000 00C8	0x4897 D0C8	0x4899 D0C8	0x489B D0C8
VIP_INT1_CLIENT0_INT_MASK	RW	32	0x0000 00CC	0x4897 D0CC	0x4899 D0CC	0x489B D0CC
VIP_INT1_CLIENT1_INT_STAT	RW	32	0x0000 00D0	0x4897 D0D0	0x4899 D0D0	0x489B D0D0
VIP_INT1_CLIENT1_INT_MASK	RW	32	0x0000 00D4	0x4897 D0D4	0x4899 D0D4	0x489B D0D4
VIP_INT1_LIST0_INT_STAT	RW	32	0x0000 00D8	0x4897 D0D8	0x4899 D0D8	0x489B D0D8
VIP_INT1_LIST0_INT_MASK	RW	32	0x0000 00DC	0x4897 D0DC	0x4899 D0DC	0x489B D0DC
VIP_INT2_CHANNEL0_INT_STAT	RW	32	0x0000 00E0	0x4897 D0E0	0x4899 D0E0	0x489B D0E0
VIP_INT2_CHANNEL0_INT_MASK	RW	32	0x0000 00E4	0x4897 D0E4	0x4899 D0E4	0x489B D0E4
VIP_INT2_CHANNEL1_INT_STAT	RW	32	0x0000 00E8	0x4897 D0E8	0x4899 D0E8	0x489B D0E8
VIP_INT2_CHANNEL1_INT_MASK	RW	32	0x0000 00EC	0x4897 D0EC	0x4899 D0EC	0x489B D0EC
VIP_INT2_CHANNEL2_INT_STAT	RW	32	0x0000 00F0	0x4897 D0F0	0x4899 D0F0	0x489B D0F0
VIP_INT2_CHANNEL2_INT_MASK	RW	32	0x0000 00F4	0x4897 D0F4	0x4899 D0F4	0x489B D0F4

Table 9-238. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_VPDMA Base Address	VIP2_VPDMA Base Address	VIP3_VPDMA Base Address
VIP_INT2_CHANNEL3_INT_STAT	RW	32	0x0000 00F8	0x4897 D0F8	0x4899 D0F8	0x489B D0F8
VIP_INT2_CHANNEL3_INT_MASK	RW	32	0x0000 00FC	0x4897 D0FC	0x4899 D0FC	0x489B D0FC
VIP_INT2_CHANNEL4_INT_STAT	RW	32	0x0000 0100	0x4897 D100	0x4899 D100	0x489B D100
VIP_INT2_CHANNEL4_INT_MASK	RW	32	0x0000 0104	0x4897 D104	0x4899 D104	0x489B D104
VIP_INT2_CHANNEL5_INT_STAT	RW	32	0x0000 0108	0x4897 D108	0x4899 D108	0x489B D108
VIP_INT2_CHANNEL5_INT_MASK	RW	32	0x0000 010C	0x4897 D10C	0x4899 D10C	0x489B D10C
VIP_INT2_CLIENT0_INT_STAT	RW	32	0x0000 0118	0x4897 D118	0x4899 D118	0x489B D118
VIP_INT2_CLIENT0_INT_MASK	RW	32	0x0000 011C	0x4897 D11C	0x4899 D11C	0x489B D11C
VIP_INT2_LIST0_INT_STAT	RW	32	0x0000 0128	0x4897 D128	0x4899 D128	0x489B D128
VIP_INT2_LIST0_INT_MASK	RW	32	0x0000 012C	0x4897 D12C	0x4899 D12C	0x489B D12C
VIP_INT3_CHANNEL0_INT_STAT	RW	32	0x0000 0130	0x4897 D130	0x4899 D130	0x489B D130
VIP_INT3_CHANNEL0_INT_MASK	RW	32	0x0000 0134	0x4897 D134	0x4899 D134	0x489B D134
VIP_INT3_CHANNEL1_INT_STAT	RW	32	0x0000 0138	0x4897 D138	0x4899 D138	0x489B D138
VIP_INT3_CHANNEL1_INT_MASK	RW	32	0x0000 013C	0x4897 D13C	0x4899 D13C	0x489B D13C
VIP_INT3_CHANNEL2_INT_STAT	RW	32	0x0000 0140	0x4897 D140	0x4899 D140	0x489B D140
VIP_INT3_CHANNEL2_INT_MASK	RW	32	0x0000 0144	0x4897 D144	0x4899 D144	0x489B D144
VIP_INT3_CHANNEL3_INT_STAT	RW	32	0x0000 0148	0x4897 D148	0x4899 D148	0x489B D148
VIP_INT3_CHANNEL3_INT_MASK	RW	32	0x0000 014C	0x4897 D14C	0x4899 D14C	0x489B D14C
VIP_INT3_CHANNEL4_INT_STAT	RW	32	0x0000 0150	0x4897 D150	0x4899 D150	0x489B D150
VIP_INT3_CHANNEL4_INT_MASK	RW	32	0x0000 0154	0x4897 D154	0x4899 D154	0x489B D154
VIP_INT3_CHANNEL5_INT_STAT	RW	32	0x0000 0158	0x4897 D158	0x4899 D158	0x489B D158
VIP_INT3_CHANNEL5_INT_MASK	RW	32	0x0000 015C	0x4897 D15C	0x4899 D15C	0x489B D15C
VIP_INT3_CLIENT0_INT_STAT	RW	32	0x0000 0168	0x4897 D168	0x4899 D168	0x489B D168
VIP_INT3_CLIENT0_INT_MASK	RW	32	0x0000 016C	0x4897 D16C	0x4899 D16C	0x489B D16C
VIP_INT3_LIST0_INT_STAT	RW	32	0x0000 0178	0x4897 D178	0x4899 D178	0x489B D178
VIP_INT3_LIST0_INT_MASK	RW	32	0x0000 017C	0x4897 D17C	0x4899 D17C	0x489B D17C
VIP_PERF_MON0	RW	32	0x0000 0200	0x4897 D200	0x4899 D200	0x489B D200
VIP_PERF_MON1	RW	32	0x0000 0204	0x4897 D204	0x4899 D204	0x489B D204
VIP_PERF_MON2	RW	32	0x0000 0208	0x4897 D208	0x4899 D208	0x489B D208
VIP_PERF_MON3	RW	32	0x0000 020C	0x4897 D20C	0x4899 D20C	0x489B D20C
VIP_PERF_MON4	RW	32	0x0000 0210	0x4897 D210	0x4899 D210	0x489B D210
VIP_PERF_MON5	RW	32	0x0000 0214	0x4897 D214	0x4899 D214	0x489B D214
VIP_PERF_MON6	RW	32	0x0000 0218	0x4897 D218	0x4899 D218	0x489B D218
VIP_PERF_MON7	RW	32	0x0000 021C	0x4897 D21C	0x4899 D21C	0x489B D21C
VIP_PERF_MON8	RW	32	0x0000 0220	0x4897 D220	0x4899 D220	0x489B D220
VIP_PERF_MON9	RW	32	0x0000 0224	0x4897 D224	0x4899 D224	0x489B D224
VIP_PERF_MON10	RW	32	0x0000 0228	0x4897 D228	0x4899 D228	0x489B D228
VIP_PERF_MON11	RW	32	0x0000 022C	0x4897 D22C	0x4899 D22C	0x489B D22C
VIP_PERF_MON12	RW	32	0x0000 0230	0x4897 D230	0x4899 D230	0x489B D230
VIP_PERF_MON13	RW	32	0x0000 0234	0x4897 D234	0x4899 D234	0x489B D234
VIP_PERF_MON14	RW	32	0x0000 0238	0x4897 D238	0x4899 D238	0x489B D238
VIP_PERF_MON15	RW	32	0x0000 023C	0x4897 D23C	0x4899 D23C	0x489B D23C
VIP_PERF_MON16	RW	32	0x0000 0240	0x4897 D240	0x4899 D240	0x489B D240
VIP_PERF_MON17	RW	32	0x0000 0244	0x4897 D244	0x4899 D244	0x489B D244
VIP_PERF_MON18	RW	32	0x0000 0248	0x4897 D248	0x4899 D248	0x489B D248
VIP_PERF_MON19	RW	32	0x0000 024C	0x4897 D24C	0x4899 D24C	0x489B D24C
VIP_PERF_MON20	RW	32	0x0000 0250	0x4897 D250	0x4899 D250	0x489B D250
VIP_PERF_MON21	RW	32	0x0000 0254	0x4897 D254	0x4899 D254	0x489B D254
VIP_PERF_MON22	RW	32	0x0000 0258	0x4897 D258	0x4899 D258	0x489B D258
VIP_PERF_MON23	RW	32	0x0000 025C	0x4897 D25C	0x4899 D25C	0x489B D25C

Table 9-238. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_VPDMA Base Address	VIP2_VPDMA Base Address	VIP3_VPDMA Base Address
VIP_PERF_MON24	RW	32	0x0000 0260	0x4897 D260	0x4899 D260	0x489B D260
VIP_PERF_MON25	RW	32	0x0000 0264	0x4897 D264	0x4899 D264	0x489B D264
VIP_PERF_MON26	RW	32	0x0000 0268	0x4897 D268	0x4899 D268	0x489B D268
VIP_PERF_MON27	RW	32	0x0000 026C	0x4897 D26C	0x4899 D26C	0x489B D26C
VIP_PERF_MON28	RW	32	0x0000 0270	0x4897 D270	0x4899 D270	0x489B D270
VIP_PERF_MON29	RW	32	0x0000 0274	0x4897 D274	0x4899 D274	0x489B D274
VIP_PERF_MON30	RW	32	0x0000 0278	0x4897 D278	0x4899 D278	0x489B D278
VIP_PERF_MON31	RW	32	0x0000 027C	0x4897 D27C	0x4899 D27C	0x489B D27C
VIP_PERF_MON32	RW	32	0x0000 0280	0x4897 D280	0x4899 D280	0x489B D280
VIP_PERF_MON33	RW	32	0x0000 0284	0x4897 D284	0x4899 D284	0x489B D284
VIP_PERF_MON34	RW	32	0x0000 0288	0x4897 D288	0x4899 D288	0x489B D288
VIP_PERF_MON35	RW	32	0x0000 028C	0x4897 D28C	0x4899 D28C	0x489B D28C
VIP_PERF_MON36	RW	32	0x0000 0290	0x4897 D290	0x4899 D290	0x489B D290
VIP_PERF_MON37	RW	32	0x0000 0294	0x4897 D294	0x4899 D294	0x489B D294
VIP_PERF_MON38	RW	32	0x0000 0298	0x4897 D298	0x4899 D298	0x489B D298
VIP_PERF_MON39	RW	32	0x0000 029C	0x4897 D29C	0x4899 D29C	0x489B D29C
VIP_PERF_MON40	RW	32	0x0000 02A0	0x4897 D2A0	0x4899 D2A0	0x489B D2A0
VIP_PERF_MON41	RW	32	0x0000 02A4	0x4897 D2A4	0x4899 D2A4	0x489B D2A4
VIP_PERF_MON42	RW	32	0x0000 02A8	0x4897 D2A8	0x4899 D2A8	0x489B D2A8
VIP_PERF_MON43	RW	32	0x0000 02AC	0x4897 D2AC	0x4899 D2AC	0x489B D2AC
VIP_PERF_MON44	RW	32	0x0000 02B0	0x4897 D2B0	0x4899 D2B0	0x489B D2B0
VIP_PERF_MON45	RW	32	0x0000 02B4	0x4897 D2B4	0x4899 D2B4	0x489B D2B4
VIP_PERF_MON46	RW	32	0x0000 02B8	0x4897 D2B8	0x4899 D2B8	0x489B D2B8
VIP_PERF_MON47	RW	32	0x0000 02BC	0x4897 D2BC	0x4899 D2BC	0x489B D2BC
VIP_PERF_MON48	RW	32	0x0000 02C0	0x4897 D2C0	0x4899 D2C0	0x489B D2C0
VIP_PERF_MON49	RW	32	0x0000 02C4	0x4897 D2C4	0x4899 D2C4	0x489B D2C4
VIP_PERF_MON50	RW	32	0x0000 02C8	0x4897 D2C8	0x4899 D2C8	0x489B D2C8
VIP_PERF_MON51	RW	32	0x0000 02CC	0x4897 D2CC	0x4899 D2CC	0x489B D2CC
VIP_PERF_MON52	RW	32	0x0000 02D0	0x4897 D2D0	0x4899 D2D0	0x489B D2D0
VIP_PERF_MON53	RW	32	0x0000 02D4	0x4897 D2D4	0x4899 D2D4	0x489B D2D4
VIP_PERF_MON54	RW	32	0x0000 02D8	0x4897 D2D8	0x4899 D2D8	0x489B D2D8
VIP_PERF_MON55	RW	32	0x0000 02DC	0x4897 D2DC	0x4899 D2DC	0x489B D2DC
VIP_PERF_MON56	RW	32	0x0000 02E0	0x4897 D2E0	0x4899 D2E0	0x489B D2E0
VIP_PERF_MON57	RW	32	0x0000 02E4	0x4897 D2E4	0x4899 D2E4	0x489B D2E4
VIP_PERF_MON58	RW	32	0x0000 02E8	0x4897 D2E8	0x4899 D2E8	0x489B D2E8
VIP_PERF_MON59	RW	32	0x0000 02EC	0x4897 D2EC	0x4899 D2EC	0x489B D2EC
VIP_PERF_MON60	RW	32	0x0000 02F0	0x4897 D2F0	0x4899 D2F0	0x489B D2F0
VIP_PERF_MON61	RW	32	0x0000 02F4	0x4897 D2F4	0x4899 D2F4	0x489B D2F4
VIP0_LO_Y_CSTAT	RW	32	0x0000 0388	0x4897 D388	0x4899 D388	0x489B D388
VIP0_LO_UV_CSTAT	RW	32	0x0000 038C	0x4897 D38C	0x4899 D38C	0x489B D38C
VIP0_UP_Y_CSTAT	RW	32	0x0000 0390	0x4897 D390	0x4899 D390	0x489B D390
VIP0_UP_UV_CSTAT	RW	32	0x0000 0394	0x4897 D394	0x4899 D394	0x489B D394
VIP1_LO_Y_CSTAT	RW	32	0x0000 0398	0x4897 D398	0x4899 D398	0x489B D398
VIP1_LO_UV_CSTAT	RW	32	0x0000 039C	0x4897 D39C	0x4899 D39C	0x489B D39C
VIP1_UP_Y_CSTAT	RW	32	0x0000 03A0	0x4897 D3A0	0x4899 D3A0	0x489B D3A0
VIP1_UP_UV_CSTAT	RW	32	0x0000 03A4	0x4897 D3A4	0x4899 D3A4	0x489B D3A4
VPI_CTL_CSTAT	RW	32	0x0000 03D0	0x4897 D3D0	0x4899 D3D0	0x489B D3D0
VIP0_ANC_A_CSTAT	RW	32	0x0000 03E8	0x4897 D3E8	0x4899 D3E8	0x489B D3E8
VIP0_ANC_B_CSTAT	RW	32	0x0000 03EC	0x4897 D3EC	0x4899 D3EC	0x489B D3EC
VIP1_ANC_A_CSTAT	RW	32	0x0000 03F0	0x4897 D3F0	0x4899 D3F0	0x489B D3F0

Table 9-238. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_VPDMA Base Address	VIP2_VPDMA Base Address	VIP3_VPDMA Base Address
VIP1_ANC_B_CSTAT	RW	32	0x0000 03F4	0x4897 D3F4	0x4899 D3F4	0x489B D3F4

9.5.6.2 VIP VPDMA Register Description

Table 9-239. VIP_PID

Address Offset	0x0000 0000		
Physical Address	0x4897 D000 0x4899 D000 0x489B D000	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	PID VIP VPDMA register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															

Bits	Field Name	Description	Type	Reset
31:0	PID	PID of VPDMA module	R	0x0

Table 9-240. Register Call Summary for Register VIP_PID

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-241. VIP_LIST_ADDR

Address Offset	0x0000 0004		
Physical Address	0x4897 D004 0x4899 D004 0x489B D004	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The location of a new list to begin processing.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Field Name	Description	Type	Reset
31:0	VIP_LIST_ADDR	Location of a new list of descriptors. This register must be written with the VPDMA Configuration Location after reset.	RW	0x0

Table 9-242. Register Call Summary for Register VIP_LIST_ADDR

VIP Functional Description

- VPDMA Introduction:
- VPDMA Basic Definitions: [1] [2]
- VPDMA Configuration: [3] [4] [5]

VIP Register Manual

- VIP VPDMA Register Summary: [6]
- VIP VPDMA Register Description: [9] [10] [11] [12]

Table 9-243. VIP_LIST_ATTR

Address Offset	0x0000 0008		
Physical Address	0x4897 D008 0x4899 D008 0x489B D008	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-243. VIP_LIST_ATTR (continued)

Description	The attributes of a new list. This register should always be written after VIP_LIST_ADDR.
Type	RW

31 30 29 28 27	26 25 24	23 22 21	20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	LIST_NUM	RESERVED	STOP	RDY	LIST_TYPE	LIST_SIZE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	LIST_NUM	The list number that should be assigned to the list located at VIP_LIST_ADDR. If the list is still active this will block all future list writes until the list is available.	RW	0x0
23:21	RESERVED		R	0x0
20	STOP	This bit is written with the LIST_NUMBER field to stop a self-modifying list. When this bit is written a one the list specified by the LIST_NUMBER is sent a stop signal and will finish the current frame of transfers and then free the list resources.	RW	0x0
19	RDY	This bit is low when a new list cannot be written to the VIP_LIST_ADDR register. The reasons this bit would be low are at initial startup if the LIST_MANAGER State Machine image has not completed loading. It also would be low if the last write to the VIP_LIST_ATTR attempted to start a list that is currently active. When this bit is low any writes to the list address register will cause access to not be accepted until this bit has set by the previous list having completed.	R	0x0
18:16	LIST_TYPE	The type of list that has been generated.\\n0: Normal List\\n1: Self-Modifying List\\n2: List Doorbell\\nOthers Reserved for future use	RW	0x0
15:0	LIST_SIZE	Number of 128 bit word in the new list of descriptors. Writes to this register will activate the list in the list stack of the list manager and begin transfer of the list into VPDMA. This size can not be 0.	RW	0x0

Table 9-244. Register Call Summary for Register VIP_LIST_ATTR

VIP Functional Description

- VPDMA Introduction:
- VPDMA Basic Definitions: [2] [3]
- VPDMA Configuration: [4] [5] [6] [7]

VIP Register Manual

- VIP VPDMA Register Summary: [8]
- VIP VPDMA Register Description: [11]

Table 9-245. VIP_LIST_STAT_SYNC

Address Offset	0x0000 000C		
Physical Address	0x4897 D00C 0x4899 D00C 0x489B D00C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register is used for processor to List Manager syncronization and status registers for the list.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LIST7_BUSY	LIST6_BUSY	LIST5_BUSY	LIST4_BUSY	LIST3_BUSY	LIST2_BUSY	LIST1_BUSY	LIST0_BUSY	RESERVED	SYNC_LISTS7	SYNC_LISTS6	SYNC_LISTS5	SYNC_LISTS4	SYNC_LISTS3	SYNC_LISTS2	SYNC_LISTS1	SYNC_LISTS0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	LIST7_BUSY	The list 7 is currently running. Any attempt to load a new list to list 7 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
22	LIST6_BUSY	The list 6 is currently running. Any attempt to load a new list to list 6 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
21	LIST5_BUSY	The list 5 is currently running. Any attempt to load a new list to list 5 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
20	LIST4_BUSY	The list 4 is currently running. Any attempt to load a new list to list 4 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
19	LIST3_BUSY	The list 3 is currently running. Any attempt to load a new list to list 3 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
18	LIST2_BUSY	The list 2 is currently running. Any attempt to load a new list to list 2 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
17	LIST1_BUSY	The list 1 is currently running. Any attempt to load a new list to list 1 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
16	LIST0_BUSY	The list 0 is currently running. Any attempt to load a new list to list 0 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
15:8	RESERVED	Reserved	R	0x0
7	SYNC_LISTS7	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 7 waiting on it.	RW	0x0
6	SYNC_LISTS6	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 6 waiting on it.	RW	0x0
5	SYNC_LISTS5	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 5 waiting on it.	RW	0x0
4	SYNC_LISTS4	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 4 waiting on it.	RW	0x0
3	SYNC_LISTS3	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 3 waiting on it.	RW	0x0
2	SYNC_LISTS2	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 2 waiting on it.	RW	0x0
1	SYNC_LISTS1	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 1 waiting on it.	RW	0x0
0	SYNC_LISTS0	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 0 waiting on it.	RW	0x0

Table 9-246. Register Call Summary for Register VIP_LIST_STAT_SYNC

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-247. VIP_BG_RGB

Address Offset	0x0000 0018		
Physical Address	0x4897 D018 0x4899 D018 0x489B D018	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The registers used to set the background color for RGB		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RED	GREEN	BLUE	BLEND

Bits	Field Name	Description	Type	Reset
31:24	RED	The red value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
23:16	GREEN	The green value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
15:8	BLUE	The blue value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	BLEND	The blend value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0

Table 9-248. Register Call Summary for Register VIP_BG_RGB

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-249. VIP_BG_YUV

Address Offset	0x0000 001C		
Physical Address	0x4897 D01C 0x4899 D01C 0x489B D01C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The registers used to set the background color for YUV		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	Y	CR	CB

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:16	Y	The Y value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0
15:8	CR	The Cr value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	CB	The Cb value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0

Table 9-250. Register Call Summary for Register VIP_BG_YUV

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-251. VIP_VPDMA_SETUP

Address Offset	0x0000 0030		
Physical Address	0x4897 D030 0x4899 D030 0x489B D030	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	Configures global parameters that are shared by all clients.		
Type	RW		

31–1	0
RESERVED	SEC_BASE_CH

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SEC_BASE_CH	Use Secondary Channels for Mosaic mode	RW	0x0

Table 9-252. Register Call Summary for Register VIP_VPDMA_SETUP

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-253. VIP_MAX_SIZE1

Address Offset	0x0000 0034		
Physical Address	0x4897 D034 0x4899 D034 0x489B D034	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 1 in write descriptor.		
Type	RW		

31–16	15–0
MAX_WIDTH	MAX_HEIGHT

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 1 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 1 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 9-254. Register Call Summary for Register VIP_MAX_SIZE1

VIP Functional Description

- VPDMA Descriptors: [0] [1]

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 9-255. VIP_MAX_SIZE2

Address Offset	0x0000 0038		
Physical Address	0x4897 D038 0x4899 D038 0x489B D038	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-255. VIP_MAX_SIZE2 (continued)

Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 2 in write descriptor.
Type	RW

31-16	15-0
MAX_WIDTH	MAX_HEIGHT

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 2 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 2 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 9-256. Register Call Summary for Register VIP_MAX_SIZE2

VIP Functional Description

- VPDMA Descriptors: [0] [1]

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 9-257. VIP_MAX_SIZE3

Address Offset	0x0000 003C		
Physical Address	0x4897 D03C 0x4899 D03C 0x489B D03C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 3 in write descriptor.		
Type	RW		

31-16	15-0
MAX_WIDTH	MAX_HEIGHT

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 3 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 3 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 9-258. Register Call Summary for Register VIP_MAX_SIZE3

VIP Functional Description

- VPDMA Descriptors: [0] [1]

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 9-259. VIP_INT0_CHANNEL0_INT_STAT

Address Offset	0x0000 0040		
Physical Address	0x4897 D040 0x4899 D040 0x489B D040	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-259. VIP_INT0_CHANNEL0_INT_STAT (continued)

Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX3	INT_STAT_GRPX2	INT_STAT_GRPX1	INT_STAT_SCALER_OUT	RESERVED								INT_STAT_SCALER_CHROMA	INT_STAT_SCALER_LUMA	INT_STAT_HQ_SCALER	RESERVED	INT_STAT_HQ_MV_OUT	RESERVED		INT_STAT_HQ_MV	RESERVED						INT_STAT_HQ_VID3_CHROMA	INT_STAT_HQ_VID3_LUMA	INT_STAT_HQ_VID2_CHROMA	INT_STAT_HQ_VID2_LUMA	INT_STAT_HQ_VID1_CHROMA	INT_STAT_HQ_VID1_LUMA

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX3	The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_GRPX2	The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_GRPX1	The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_SCALER_OUT	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_STAT_SCALER_CHROMA	The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_STAT_SCALER_LUMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_HQ_SCALER	The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_HQ_MV_OUT	The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_HQ_MV	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_HQ_VID3_CHROMA	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_HQ_VID3_LUMA	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_HQ_VID2_CHROMA	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_HQ_VID2_LUMA	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	INT_STAT_HQ_VID1_CHROMA	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_HQ_VID1_LUMA	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-260. Register Call Summary for Register VIP_INT0_CHANNEL0_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-261. VIP_INT0_CHANNEL0_INT_MASK

Address Offset	0x0000 0044		
Physical Address	0x4897 D044 0x4899 D044 0x489B D044	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27-20	19	18	17	16	15	14-13	12	11-6	5	4	3	2	1	0
INT_MASK_GRPX3	INT_MASK_GRPX2	INT_MASK_GRPX1	INT_MASK_SCALER_OUT	RESERVED	INT_MASK_SCALER_CHROMA	INT_MASK_SCALER_LUMA	INT_MASK_HQ_SCALER	RESERVED	INT_MASK_HQ_MV_OUT	RESERVED	INT_MASK_HQ_MV	RESERVED	INT_MASK_HQ_VID3_CHROMA	INT_MASK_HQ_VID3_LUMA	INT_MASK_HQ_VID2_CHROMA	INT_MASK_HQ_VID2_LUMA	INT_MASK_HQ_VID1_CHROMA	INT_MASK_HQ_VID1_LUMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX3	The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_GRPX2	The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_GRPX1	The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
28	INT_MASK_SCALER_OUT	The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_MASK_SCALER_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_SCALER_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_HQ_SCALER	The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_HQ_MV_OUT	The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_HQ_MV	The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_HQ_VID3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_HQ_VID3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_HQ_VID2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_HQ_VID2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_HQ_VID1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_HQ_VID1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-262. Register Call Summary for Register VIP_INT0_CHANNEL0_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-263. VIP_INT0_CHANNEL1_INT_STAT

Address Offset	0x0000 0048		
Physical Address	0x4897 D048 0x4899 D048 0x489B D048	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-263. VIP_INT0_CHANNEL1_INT_STAT (continued)

Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.
Type	RW

31	30	29	28	27	26	25	24
INT_STAT_VIP1_MULT_PORTB_SRC9	INT_STAT_VIP1_MULT_PORTB_SRC8	INT_STAT_VIP1_MULT_PORTB_SRC7	INT_STAT_VIP1_MULT_PORTB_SRC6	INT_STAT_VIP1_MULT_PORTB_SRC5	INT_STAT_VIP1_MULT_PORTB_SRC4	INT_STAT_VIP1_MULT_PORTB_SRC3	INT_STAT_VIP1_MULT_PORTB_SRC2

23	22	21	20	19	18	17	16
INT_STAT_VIP1_MULT_PORTB_SRC1	INT_STAT_VIP1_MULT_PORTB_SRC0	INT_STAT_VIP1_MULT_PORTA_SRC15	INT_STAT_VIP1_MULT_PORTA_SRC14	INT_STAT_VIP1_MULT_PORTA_SRC13	INT_STAT_VIP1_MULT_PORTA_SRC12	INT_STAT_VIP1_MULT_PORTA_SRC11	INT_STAT_VIP1_MULT_PORTA_SRC10

15	14	13	12	11	10	9	8
INT_STAT_VIP1_MULT_PORTA_SRC9	INT_STAT_VIP1_MULT_PORTA_SRC8	INT_STAT_VIP1_MULT_PORTA_SRC7	INT_STAT_VIP1_MULT_PORTA_SRC6	INT_STAT_VIP1_MULT_PORTA_SRC5	INT_STAT_VIP1_MULT_PORTA_SRC4	INT_STAT_VIP1_MULT_PORTA_SRC3	INT_STAT_VIP1_MULT_PORTA_SRC2

7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_PORTA_SRC1	INT_STAT_VIP1_MULT_PORTA_SRC0	RESERVED					

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP1_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP1_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_STAT_VIP1_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
10	INT_STAT_VIP1_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 9-264. Register Call Summary for Register VIP_INT0_CHANNEL1_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-265. VIP_INT0_CHANNEL1_INT_MASK

Address Offset	0x0000 004C		
Physical Address	0x4897 D04C 0x4899 D04C 0x489B D04C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_MASK_VIP1_MULT_PORTB_SRC9	INT_MASK_VIP1_MULT_PORTB_SRC8	INT_MASK_VIP1_MULT_PORTB_SRC7	INT_MASK_VIP1_MULT_PORTB_SRC6	INT_MASK_VIP1_MULT_PORTB_SRC5	INT_MASK_VIP1_MULT_PORTB_SRC4	INT_MASK_VIP1_MULT_PORTB_SRC3	INT_MASK_VIP1_MULT_PORTB_SRC2	INT_MASK_VIP1_MULT_PORTB_SRC1	INT_MASK_VIP1_MULT_PORTB_SRC0	INT_MASK_VIP1_MULT_PORTA_SRC15	INT_MASK_VIP1_MULT_PORTA_SRC14	INT_MASK_VIP1_MULT_PORTA_SRC13	INT_MASK_VIP1_MULT_PORTA_SRC12	INT_MASK_VIP1_MULT_PORTA_SRC11	INT_MASK_VIP1_MULT_PORTA_SRC10	INT_MASK_VIP1_MULT_PORTA_SRC9	INT_MASK_VIP1_MULT_PORTA_SRC8	INT_MASK_VIP1_MULT_PORTA_SRC7	INT_MASK_VIP1_MULT_PORTA_SRC6	INT_MASK_VIP1_MULT_PORTA_SRC5	INT_MASK_VIP1_MULT_PORTA_SRC4	INT_MASK_VIP1_MULT_PORTA_SRC3	INT_MASK_VIP1_MULT_PORTA_SRC2	INT_MASK_VIP1_MULT_PORTA_SRC1	INT_MASK_VIP1_MULT_PORTA_SRC0	RESERVED

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_PORTB_SRC9	The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_PORTB_SRC8	The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_PORTB_SRC7	The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_PORTB_SRC6	The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_PORTB_SRC5	The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_PORTB_SRC4	The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_PORTB_SRC3	The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_PORTB_SRC2	The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_PORTB_SRC1	The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_PORTB_SRC0	The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_PORTA_SRC15	The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_PORTA_SRC14	The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_PORTA_SRC13	The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_PORTA_SRC12	The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_MASK_VIP1_MULT_PORTA_SRC11	The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_PORTA_SRC10	The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_PORTA_SRC9	The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_PORTA_SRC8	The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_PORTA_SRC7	The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_PORTA_SRC6	The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_PORTA_SRC5	The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_PORTA_SRC4	The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_PORTA_SRC3	The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_PORTA_SRC2	The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_PORTA_SRC1	The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_PORTA_SRC0	The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 9-266. Register Call Summary for Register VIP_INT0_CHANNEL1_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-267. VIP_INT0_CHANNEL2_INT_STAT

Address Offset	0x0000 0050		
Physical Address	0x4897 D050 0x4899 D050 0x489B D050	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP1_MULT_ANCB_SRC9	INT_STAT_VIP1_MULT_ANCB_SRC8	INT_STAT_VIP1_MULT_ANCB_SRC7	INT_STAT_VIP1_MULT_ANCB_SRC6	INT_STAT_VIP1_MULT_ANCB_SRC5	INT_STAT_VIP1_MULT_ANCB_SRC4	INT_STAT_VIP1_MULT_ANCB_SRC3	INT_STAT_VIP1_MULT_ANCB_SRC2	INT_STAT_VIP1_MULT_ANCB_SRC1	INT_STAT_VIP1_MULT_ANCB_SRC0	INT_STAT_VIP1_MULT_ANCA_SRC15	INT_STAT_VIP1_MULT_ANCA_SRC14	INT_STAT_VIP1_MULT_ANCA_SRC13	INT_STAT_VIP1_MULT_ANCA_SRC12	INT_STAT_VIP1_MULT_ANCA_SRC11	INT_STAT_VIP1_MULT_ANCA_SRC10	INT_STAT_VIP1_MULT_ANCA_SRC9	INT_STAT_VIP1_MULT_ANCA_SRC8	INT_STAT_VIP1_MULT_ANCA_SRC7	INT_STAT_VIP1_MULT_ANCA_SRC6	INT_STAT_VIP1_MULT_ANCA_SRC5	INT_STAT_VIP1_MULT_ANCA_SRC4	INT_STAT_VIP1_MULT_ANCA_SRC3	INT_STAT_VIP1_MULT_ANCA_SRC2	INT_STAT_VIP1_MULT_ANCA_SRC1	INT_STAT_VIP1_MULT_ANCA_SRC0	INT_STAT_VIP1_MULT_PORTB_SRC15	INT_STAT_VIP1_MULT_PORTB_SRC14	INT_STAT_VIP1_MULT_PORTB_SRC13	INT_STAT_VIP1_MULT_PORTB_SRC12	INT_STAT_VIP1_MULT_PORTB_SRC11	INT_STAT_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP1_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP1_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP1_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_VIP1_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-268. Register Call Summary for Register VIP_INT0_CHANNEL2_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-269. VIP_INT0_CHANNEL2_INT_MASK

Address Offset	0x0000 0054		
Physical Address	0x4897 D054 0x4899 D054 0x489B D054	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-269. VIP_INT0_CHANNEL2_INT_MASK (continued)

Description The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_MASK_VIP1_MULT_ANCB_SRC9	INT_MASK_VIP1_MULT_ANCB_SRC8	INT_MASK_VIP1_MULT_ANCB_SRC7	INT_MASK_VIP1_MULT_ANCB_SRC6	INT_MASK_VIP1_MULT_ANCB_SRC5	INT_MASK_VIP1_MULT_ANCB_SRC4	INT_MASK_VIP1_MULT_ANCB_SRC3	INT_MASK_VIP1_MULT_ANCB_SRC2	INT_MASK_VIP1_MULT_ANCB_SRC1	INT_MASK_VIP1_MULT_ANCB_SRC0	INT_MASK_VIP1_MULT_ANCA_SRC15	INT_MASK_VIP1_MULT_ANCA_SRC14	INT_MASK_VIP1_MULT_ANCA_SRC13	INT_MASK_VIP1_MULT_ANCA_SRC12	INT_MASK_VIP1_MULT_ANCA_SRC11	INT_MASK_VIP1_MULT_ANCA_SRC10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_ANCA_SRC9	INT_MASK_VIP1_MULT_ANCA_SRC8	INT_MASK_VIP1_MULT_ANCA_SRC7	INT_MASK_VIP1_MULT_ANCA_SRC6	INT_MASK_VIP1_MULT_ANCA_SRC5	INT_MASK_VIP1_MULT_ANCA_SRC4	INT_MASK_VIP1_MULT_ANCA_SRC3	INT_MASK_VIP1_MULT_ANCA_SRC2	INT_MASK_VIP1_MULT_ANCA_SRC1	INT_MASK_VIP1_MULT_ANCA_SRC0	INT_MASK_VIP1_MULT_PORTB_SRC15	INT_MASK_VIP1_MULT_PORTB_SRC14	INT_MASK_VIP1_MULT_PORTB_SRC13	INT_MASK_VIP1_MULT_PORTB_SRC12	INT_MASK_VIP1_MULT_PORTB_SRC11	INT_MASK_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_ANCB_SRC9	The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_ANCB_SRC8	The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_ANCB_SRC7	The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_ANCB_SRC6	The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_ANCB_SRC5	The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_ANCB_SRC4	The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_ANCB_SRC3	The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_ANCB_SRC2	The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_ANCB_SRC1	The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_MASK_VIP1_MULT_ANCB_SRC0	The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_ANCA_SRC15	The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_ANCA_SRC14	The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_ANCA_SRC13	The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_ANCA_SRC12	The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_ANCA_SRC11	The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_ANCA_SRC10	The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_ANCA_SRC9	The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_ANCA_SRC8	The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_ANCA_SRC7	The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_ANCA_SRC6	The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_ANCA_SRC5	The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_ANCA_SRC4	The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_ANCA_SRC3	The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_ANCA_SRC2	The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_MASK_VIP1_MULT_ANCA_SRC1	The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_ANCA_SRC0	The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_PORTB_SRC15	The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_PORTB_SRC14	The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_PORTB_SRC13	The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_PORTB_SRC12	The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_PORTB_SRC11	The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_PORTB_SRC10	The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-270. Register Call Summary for Register VIP_INT0_CHANNEL2_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-271. VIP_INT0_CHANNEL3_INT_STAT

Address Offset	0x0000 0058		
Physical Address	0x4897 D058 0x4899 D058 0x489B D058	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP2_MULT_PORTB_SRC3	INT_STAT_VIP2_MULT_PORTB_SRC2	INT_STAT_VIP2_MULT_PORTB_SRC1	INT_STAT_VIP2_MULT_PORTB_SRC0	INT_STAT_VIP2_MULT_PORTA_SRC15	INT_STAT_VIP2_MULT_PORTA_SRC14	INT_STAT_VIP2_MULT_PORTA_SRC13	INT_STAT_VIP2_MULT_PORTA_SRC12	INT_STAT_VIP2_MULT_PORTA_SRC11	INT_STAT_VIP2_MULT_PORTA_SRC10	INT_STAT_VIP2_MULT_PORTA_SRC9	INT_STAT_VIP2_MULT_PORTA_SRC8	INT_STAT_VIP2_MULT_PORTA_SRC7	INT_STAT_VIP2_MULT_PORTA_SRC6	INT_STAT_VIP2_MULT_PORTA_SRC5	INT_STAT_VIP2_MULT_PORTA_SRC4	INT_STAT_VIP2_MULT_PORTA_SRC3	INT_STAT_VIP2_MULT_PORTA_SRC2	INT_STAT_VIP2_MULT_PORTA_SRC1	INT_STAT_VIP2_MULT_PORTA_SRC0	INT_STAT_VIP1_PORTB_RGB	INT_STAT_VIP1_PORTA_RGB	INT_STAT_VIP1_PORTB_CHROMA	INT_STAT_VIP1_PORTB_LUMA	INT_STAT_VIP1_PORTA_CHROMA	INT_STAT_VIP1_PORTA_LUMA	INT_STAT_VIP1_MULT_ANCB_SRC15	INT_STAT_VIP1_MULT_ANCB_SRC14	INT_STAT_VIP1_MULT_ANCB_SRC13	INT_STAT_VIP1_MULT_ANCB_SRC12	INT_STAT_VIP1_MULT_ANCB_SRC11	INT_STAT_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
25	INT_STAT_VIP2_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_VIP2_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_PORTB_RGB	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_PORTA_RGB	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_PORTB_CHROMA	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	INT_STAT_VIP1_PORTB_LUMA	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_PORTA_CHROMA	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_PORTA_LUMA	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_VIP1_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-272. Register Call Summary for Register VIP_INT0_CHANNEL3_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-273. VIP_INT0_CHANNEL3_INT_MASK

Address Offset	0x0000 005C		
Physical Address	0x4897 D05C 0x4899 D05C 0x489B D05C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_MASK_VIP2_MULT_PORTB_SRC3	INT_MASK_VIP2_MULT_PORTB_SRC2	INT_MASK_VIP2_MULT_PORTB_SRC1	INT_MASK_VIP2_MULT_PORTB_SRC0	INT_MASK_VIP2_MULT_PORTA_SRC15	INT_MASK_VIP2_MULT_PORTA_SRC14	INT_MASK_VIP2_MULT_PORTA_SRC13	INT_MASK_VIP2_MULT_PORTA_SRC12	INT_MASK_VIP2_MULT_PORTA_SRC11	INT_MASK_VIP2_MULT_PORTA_SRC10	INT_MASK_VIP2_MULT_PORTA_SRC9	INT_MASK_VIP2_MULT_PORTA_SRC8	INT_MASK_VIP2_MULT_PORTA_SRC7	INT_MASK_VIP2_MULT_PORTA_SRC6	INT_MASK_VIP2_MULT_PORTA_SRC5	INT_MASK_VIP2_MULT_PORTA_SRC4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_PORTA_SRC3	INT_MASK_VIP2_MULT_PORTA_SRC2	INT_MASK_VIP2_MULT_PORTA_SRC1	INT_MASK_VIP2_MULT_PORTA_SRC0	INT_MASK_VIP1_PORTB_RGB	INT_MASK_VIP1_PORTA_RGB	INT_MASK_VIP1_PORTB_CHROMA	INT_MASK_VIP1_PORTB_LUMA	INT_MASK_VIP1_PORTA_CHROMA	INT_MASK_VIP1_PORTA_LUMA	INT_MASK_VIP1_MULT_ANCB_SRC15	INT_MASK_VIP1_MULT_ANCB_SRC14	INT_MASK_VIP1_MULT_ANCB_SRC13	INT_MASK_VIP1_MULT_ANCB_SRC12	INT_MASK_VIP1_MULT_ANCB_SRC11	INT_MASK_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_PORTB_SRC3	The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_PORTB_SRC2	The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_PORTB_SRC1	The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_PORTB_SRC0	The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
27	INT_MASK_VIP2_MULT_PORTA_SRC15	The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_PORTA_SRC14	The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_PORTA_SRC13	The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_PORTA_SRC12	The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_PORTA_SRC11	The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_PORTA_SRC10	The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_PORTA_SRC9	The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_PORTA_SRC8	The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_PORTA_SRC7	The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_PORTA_SRC6	The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_PORTA_SRC5	The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_PORTA_SRC4	The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_PORTA_SRC3	The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_PORTA_SRC2	The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_PORTA_SRC1	The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_PORTA_SRC0	The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_PORTB_RGB	The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_PORTA_RGB	The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	INT_MASK_VIP1_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_ANCB_SRC15	The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_ANCB_SRC14	The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_ANCB_SRC13	The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_ANCB_SRC12	The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_ANCB_SRC11	The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_ANCB_SRC10	The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-274. Register Call Summary for Register VIP_INT0_CHANNEL3_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-275. VIP_INT0_CHANNEL4_INT_STAT

Address Offset	0x0000 0060		
Physical Address	0x4897 D060 0x4899 D060 0x489B D060	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP2_MULT_ANCB_SRC3	INT_STAT_VIP2_MULT_ANCB_SRC2	INT_STAT_VIP2_MULT_ANCB_SRC1	INT_STAT_VIP2_MULT_ANCB_SRC0	INT_STAT_VIP2_MULT_ANCA_SRC15	INT_STAT_VIP2_MULT_ANCA_SRC14	INT_STAT_VIP2_MULT_ANCA_SRC13	INT_STAT_VIP2_MULT_ANCA_SRC12	INT_STAT_VIP2_MULT_ANCA_SRC11	INT_STAT_VIP2_MULT_ANCA_SRC10	INT_STAT_VIP2_MULT_ANCA_SRC9	INT_STAT_VIP2_MULT_ANCA_SRC8	INT_STAT_VIP2_MULT_ANCA_SRC7	INT_STAT_VIP2_MULT_ANCA_SRC6	INT_STAT_VIP2_MULT_ANCA_SRC5	INT_STAT_VIP2_MULT_ANCA_SRC4	INT_STAT_VIP2_MULT_ANCA_SRC3	INT_STAT_VIP2_MULT_ANCA_SRC2	INT_STAT_VIP2_MULT_ANCA_SRC1	INT_STAT_VIP2_MULT_ANCA_SRC0	INT_STAT_VIP2_MULT_PORTB_SRC15	INT_STAT_VIP2_MULT_PORTB_SRC14	INT_STAT_VIP2_MULT_PORTB_SRC13	INT_STAT_VIP2_MULT_PORTB_SRC12	INT_STAT_VIP2_MULT_PORTB_SRC11	INT_STAT_VIP2_MULT_PORTB_SRC10	INT_STAT_VIP2_MULT_PORTB_SRC9	INT_STAT_VIP2_MULT_PORTB_SRC8	INT_STAT_VIP2_MULT_PORTB_SRC7	INT_STAT_VIP2_MULT_PORTB_SRC6	INT_STAT_VIP2_MULT_PORTB_SRC5	INT_STAT_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP2_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP2_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP2_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP2_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-276. Register Call Summary for Register VIP_INT0_CHANNEL4_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-277. VIP_INT0_CHANNEL4_INT_MASK

Address Offset	0x0000 0064		
Physical Address	0x4897 D064 0x4899 D064 0x489B D064	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_MASK_VIP2_MULT_ANCB_SRC3	INT_MASK_VIP2_MULT_ANCB_SRC2	INT_MASK_VIP2_MULT_ANCB_SRC1	INT_MASK_VIP2_MULT_ANCB_SRC0	INT_MASK_VIP2_MULT_ANCA_SRC15	INT_MASK_VIP2_MULT_ANCA_SRC14	INT_MASK_VIP2_MULT_ANCA_SRC13	INT_MASK_VIP2_MULT_ANCA_SRC12	INT_MASK_VIP2_MULT_ANCA_SRC11	INT_MASK_VIP2_MULT_ANCA_SRC10	INT_MASK_VIP2_MULT_ANCA_SRC9	INT_MASK_VIP2_MULT_ANCA_SRC8	INT_MASK_VIP2_MULT_ANCA_SRC7	INT_MASK_VIP2_MULT_ANCA_SRC6	INT_MASK_VIP2_MULT_ANCA_SRC5	INT_MASK_VIP2_MULT_ANCA_SRC4	INT_MASK_VIP2_MULT_ANCA_SRC3	INT_MASK_VIP2_MULT_ANCA_SRC2	INT_MASK_VIP2_MULT_ANCA_SRC1	INT_MASK_VIP2_MULT_ANCA_SRC0	INT_MASK_VIP2_MULT_PORTB_SRC15	INT_MASK_VIP2_MULT_PORTB_SRC14	INT_MASK_VIP2_MULT_PORTB_SRC13	INT_MASK_VIP2_MULT_PORTB_SRC12	INT_MASK_VIP2_MULT_PORTB_SRC11	INT_MASK_VIP2_MULT_PORTB_SRC10	INT_MASK_VIP2_MULT_PORTB_SRC9	INT_MASK_VIP2_MULT_PORTB_SRC8	INT_MASK_VIP2_MULT_PORTB_SRC7	INT_MASK_VIP2_MULT_PORTB_SRC6	INT_MASK_VIP2_MULT_PORTB_SRC5	INT_MASK_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_ANCB_SRC3	The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_ANCB_SRC2	The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_ANCB_SRC1	The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_ANCB_SRC0	The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_ANCA_SRC15	The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_ANCA_SRC14	The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_ANCA_SRC13	The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_ANCA_SRC12	The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_ANCA_SRC11	The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_ANCA_SRC10	The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
21	INT_MASK_VIP2_MULT_ANCA_SRC9	The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_ANCA_SRC8	The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_ANCA_SRC7	The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_ANCA_SRC6	The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_ANCA_SRC5	The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_ANCA_SRC4	The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_ANCA_SRC3	The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_ANCA_SRC2	The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_ANCA_SRC1	The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_ANCA_SRC0	The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_PORTB_SRC15	The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_PORTB_SRC14	The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_PORTB_SRC13	The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_PORTB_SRC12	The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_PORTB_SRC11	The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_PORTB_SRC10	The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_MASK_VIP2_MULT_PORTB_SRC9	The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_PORTB_SRC8	The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_PORTB_SRC7	The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_PORTB_SRC6	The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_PORTB_SRC5	The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_PORTB_SRC4	The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-278. Register Call Summary for Register VIP_INT0_CHANNEL4_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-279. VIP_INT0_CHANNEL5_INT_STAT

Address Offset	0x0000 0068		
Physical Address	0x4897 D068 0x4899 D068 0x489B D068	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

Bit	Field
31	INT_STAT_TRANSCODE2_CHROMA
30	INT_STAT_TRANSCODE2_LUMA
29	INT_STAT_TRANSCODE1_CHROMA
28	INT_STAT_TRANSCODE1_LUMA
27	INT_STAT_AUX_IN
26	INT_STAT_PIP_FRAME
25	INT_STAT_POST_COMP_WR
24	INT_STAT_VBI_SDVENC
23	RESERVED
22	INT_STAT_NF_LAST_CHROMA
21	INT_STAT_NF_LAST_LUMA
20	INT_STAT_NF_WRITE_CHROMA
19	INT_STAT_NF_WRITE_LUMA
18	INT_STAT_OTHER
17	INT_STAT_VIP2_PORTB_RGB
16	INT_STAT_VIP2_PORTA_RGB
15	INT_STAT_VIP2_PORTB_CHROMA
14	INT_STAT_VIP2_PORTB_LUMA
13	INT_STAT_VIP2_PORTA_CHROMA
12	INT_STAT_VIP2_PORTA_LUMA
11	INT_STAT_VIP2_MULT_ANCB_SRC15
10	INT_STAT_VIP2_MULT_ANCB_SRC14
9	INT_STAT_VIP2_MULT_ANCB_SRC13
8	INT_STAT_VIP2_MULT_ANCB_SRC12
7	INT_STAT_VIP2_MULT_ANCB_SRC11
6	INT_STAT_VIP2_MULT_ANCB_SRC10
5	INT_STAT_VIP2_MULT_ANCB_SRC9
4	INT_STAT_VIP2_MULT_ANCB_SRC8
3	INT_STAT_VIP2_MULT_ANCB_SRC7
2	INT_STAT_VIP2_MULT_ANCB_SRC6
1	INT_STAT_VIP2_MULT_ANCB_SRC5
0	INT_STAT_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_TRANSCODE2_CHROMA	The last write DMA transaction has completed for channel transcode2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_TRANSCODE2_LUMA	The last write DMA transaction has completed for channel transcode2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_TRANSCODE1_CHROMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_TRANSCODE1_LUMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_AUX_IN	The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_PIP_FRAME	The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_POST_COMP_WR	The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VBI_SD_VENC	The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_NF_LAST_CHROMA	The last write DMA transaction has completed for channel nf_last_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_NF_LAST_LUMA	The last write DMA transaction has completed for channel nf_last_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_NF_WRITE_CHROMA	The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_NF_WRITE_LUMA	The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_OTHER	This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_PORTB_RGB	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_PORTA_RGB	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_PORTB_CHROMA	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_PORTB_LUMA	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	INT_STAT_VIP2_PORTA_CHROMA	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_PORTA_LUMA	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_VIP2_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-280. Register Call Summary for Register VIP_INT0_CHANNEL5_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-281. VIP_INT0_CHANNEL5_INT_MASK

Address Offset	0x0000 006C		
Physical Address	0x4897 D06C 0x4899 D06C 0x489B D06C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-281. VIP_INT0_CHANNEL5_INT_MASK (continued)

Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_OTHER	INT_MASK_VIP2_PORTB_RGB	INT_MASK_VIP2_PORTA_RGB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_PORTB_CHROMA	INT_MASK_VIP2_PORTB_LUMA	INT_MASK_VIP2_PORTA_CHROMA	INT_MASK_VIP2_PORTA_LUMA	INT_MASK_VIP2_MULT_ANCB_SRC15	INT_MASK_VIP2_MULT_ANCB_SRC14	INT_MASK_VIP2_MULT_ANCB_SRC13	INT_MASK_VIP2_MULT_ANCB_SRC12	INT_MASK_VIP2_MULT_ANCB_SRC11	INT_MASK_VIP2_MULT_ANCB_SRC10	INT_MASK_VIP2_MULT_ANCB_SRC9	INT_MASK_VIP2_MULT_ANCB_SRC8	INT_MASK_VIP2_MULT_ANCB_SRC7	INT_MASK_VIP2_MULT_ANCB_SRC6	INT_MASK_VIP2_MULT_ANCB_SRC5	INT_MASK_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_OTHER	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_ANCB_SRC15	The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_ANCB_SRC14	The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_ANCB_SRC13	The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_ANCB_SRC12	The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_ANCB_SRC11	The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_ANCB_SRC10	The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_ANCB_SRC9	The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_ANCB_SRC8	The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_MASK_VIP2_MULT_ANCB_SRC7	The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_ANCB_SRC6	The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_ANCB_SRC5	The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_ANCB_SRC4	The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-282. Register Call Summary for Register VIP_INT0_CHANNEL5_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-283. VIP_INT0_CLIENT0_INT_STAT

Address Offset	0x0000 0078		
Physical Address	0x4897 D078 0x4899 D078 0x489B D078	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

Bits	Field
31	INT_STAT_GRPX1_DATA
30	INT_STAT_COMP_WRBK
29	INT_STAT_SC_OUT
28-21	RESERVED
20	INT_STAT_SC_IN_LUMA
19	INT_STAT_SC_IN_CHROMA
18	INT_STAT_PIP_WRBK
17	INT_STAT_DEI_SC_OUT
16	RESERVED
15	INT_STAT_DEI_HQ_MV_OUT
14-13	RESERVED
12	INT_STAT_DEI_HQ_MV_IN
11-6	RESERVED
5	INT_STAT_DEI_HQ_3_CHROMA
4	INT_STAT_DEI_HQ_3_LUMA
3	INT_STAT_DEI_HQ_2_CHROMA
2	INT_STAT_DEI_HQ_2_LUMA
1	INT_STAT_DEI_HQ_1_LUMA
0	INT_STAT_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX1_DATA	The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	INT_STAT_COMP_WRBK	The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_SC_OUT	The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_STAT_SC_IN_LUMA	The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_SC_IN_CHROMA	The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_PIP_WRBK	The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_DEI_SC_OUT	The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_DEI_HQ_MV_OUT	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_DEI_HQ_MV_IN	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_DEI_HQ_3_CHROMA	The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_DEI_HQ_3_LUMA	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_DEI_HQ_2_CHROMA	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_DEI_HQ_2_LUMA	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_DEI_HQ_1_LUMA	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_DEI_HQ_1_CHROMA	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-284. Register Call Summary for Register VIP_INT0_CLIENT0_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

Table 9-284. Register Call Summary for Register VIP_INT0_CLIENT0_INT_STAT (continued)

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-285. VIP_INT0_CLIENT0_INT_MASK

Address Offset	0x0000 007C		
Physical Address	0x4897 D07C 0x4899 D07C 0x489B D07C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX1_DATA	INT_MASK_COMP_WRBK	INT_MASK_SC_OUT	RESERVED								INT_MASK_SC_IN_LUMA	INT_MASK_SC_IN_CHROMA	INT_MASK_PIP_WRBK	INT_MASK_DEI_SC_OUT	RESERVED	INT_MASK_DEI_HQ_MV_OUT	RESERVED		INT_MASK_DEI_HQ_MV_IN	RESERVED						INT_MASK_DEI_HQ_3_CHROMA	INT_MASK_DEI_HQ_3_LUMA	INT_MASK_DEI_HQ_2_CHROMA	INT_MASK_DEI_HQ_2_LUMA	INT_MASK_DEI_HQ_1_LUMA	INT_MASK_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX1_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_COMP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_MASK_SC_IN_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_SC_IN_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_PIP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_DEI_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_DEI_HQ_MV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12	INT_MASK_DEI_HQ_MV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_DEI_HQ_3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_DEI_HQ_3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_DEI_HQ_2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_DEI_HQ_2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_DEI_HQ_1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_DEI_HQ_1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-286. Register Call Summary for Register VIP_INT0_CLIENT0_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-287. VIP_INT0_CLIENT1_INT_STAT

Address Offset	0x0000 0080		
Physical Address	0x4897 D080 0x4899 D080 0x489B D080	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	RESERVED	INT_STAT_VIP2_ANC_B	INT_STAT_VIP2_ANC_A	INT_STAT_VIP1_ANC_B	INT_STAT_VIP1_ANC_A	INT_STAT_TRANS2_LUMA	INT_STAT_TRANS2_CHROMA	INT_STAT_TRANS1_LUMA	INT_STAT_TRANS1_CHROMA	INT_STAT_HDMI_WRBK_OUT	INT_STAT_VPI_CTL	INT_STAT_VBI_SDVENC	RESERVED	INT_STAT_NF_420_UV_OUT	INT_STAT_NF_420_Y_OUT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_NF_420_UV_IN	INT_STAT_NF_420_Y_IN	INT_STAT_NF_422_IN	INT_STAT_GRPX3_ST	INT_STAT_GRPX2_ST	INT_STAT_GRPX1_ST	INT_STAT_VIP2_UP_UV	INT_STAT_VIP2_UP_Y	INT_STAT_VIP2_LO_UV	INT_STAT_VIP2_LO_Y	INT_STAT_VIP1_UP_UV	INT_STAT_VIP1_UP_Y	INT_STAT_VIP1_LO_UV	INT_STAT_VIP1_LO_Y	INT_STAT_GRPX3_DATA	INT_STAT_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
30	RESERVED	Reserved	R	0
29	INT_STAT_VIP2_ANC_B	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
28	INT_STAT_VIP2_ANC_A	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
27	INT_STAT_VIP1_ANC_B	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
26	INT_STAT_VIP1_ANC_A	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
25	INT_STAT_TRANS2_LUMA	The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
24	INT_STAT_TRANS2_CHROMA	The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
23	INT_STAT_TRANS1_LUMA	The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_TRANS1_CHROMA	The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
21	INT_STAT_HDMI_WRBK_OUT	The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
20	INT_STAT_VPI_CTL	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
19	INT_STAT_VBI_SDVENC	The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
18	RESERVED	Reserved	R	0
17	INT_STAT_NF_420_UV_OUT	The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
16	INT_STAT_NF_420_Y_OUT	The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
15	INT_STAT_NF_420_UV_IN	The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
14	INT_STAT_NF_420_Y_IN	The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
13	INT_STAT_NF_422_IN	The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
12	INT_STAT_GRPX3_ST	The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
11	INT_STAT_GRPX2_ST	The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
10	INT_STAT_GRPX1_ST	The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
9	INT_STAT_VIP2_UP_UV	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
8	INT_STAT_VIP2_UP_Y	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
7	INT_STAT_VIP2_LO_UV	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
6	INT_STAT_VIP2_LO_Y	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
5	INT_STAT_VIP1_UP_UV	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
4	INT_STAT_VIP1_UP_Y	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
3	INT_STAT_VIP1_LO_UV	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
2	INT_STAT_VIP1_LO_Y	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
1	INT_STAT_GRPX3_DATA	The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_GRPX2_DATA	The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Table 9-288. Register Call Summary for Register VIP_INT0_CLIENT1_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-289. VIP_INT0_CLIENT1_INT_MASK

Address Offset	0x0000 0084		
Physical Address	0x4897 D084 0x4899 D084 0x489B D084	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	RESERVED	INT_MASK_VIP2_ANC_B	INT_MASK_VIP2_ANC_A	INT_MASK_VIP1_ANC_B	INT_MASK_VIP1_ANC_A	INT_MASK_TRANS2_LUMA	INT_MASK_TRANS2_CHROMA	INT_MASK_TRANS1_LUMA	INT_MASK_TRANS1_CHROMA	INT_MASK_HDMI_WRBK_OUT	INT_MASK_VPI_CTL	INT_MASK_VBI_SDVENC	RESERVED	INT_MASK_NF_420_UV_OUT	INT_MASK_NF_420_Y_OUT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_NF_420_UV_IN	INT_MASK_NF_420_Y_IN	INT_MASK_NF_422_IN	INT_MASK_GRPX3_ST	INT_MASK_GRPX2_ST	INT_MASK_GRPX1_ST	INT_MASK_VIP2_UP_UV	INT_MASK_VIP2_UP_Y	INT_MASK_VIP2_LO_UV	INT_MASK_VIP2_LO_Y	INT_MASK_VIP1_UP_UV	INT_MASK_VIP1_UP_Y	INT_MASK_VIP1_LO_UV	INT_MASK_VIP1_LO_Y	INT_MASK_GRPX3_DATA	INT_MASK_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_MASK_VIP2_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
28	INT_MASK_VIP2_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
27	INT_MASK_VIP1_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
26	INT_MASK_VIP1_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
25	INT_MASK_TRANS2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
24	INT_MASK_TRANS2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
23	INT_MASK_TRANS1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
22	INT_MASK_TRANS1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
21	INT_MASK_HDMI_WRBK_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
20	INT_MASK_VPI_CTL	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
19	INT_MASK_VBI_SDVENC	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
18	RESERVED	Reserved	R	0
17	INT_MASK_NF_420_UV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
16	INT_MASK_NF_420_Y_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
15	INT_MASK_NF_420_UV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
14	INT_MASK_NF_420_Y_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
13	INT_MASK_NF_422_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
12	INT_MASK_GRPX3_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
11	INT_MASK_GRPX2_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
10	INT_MASK_GRPX1_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
9	INT_MASK_VIP2_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
8	INT_MASK_VIP2_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
7	INT_MASK_VIP2_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
6	INT_MASK_VIP2_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
5	INT_MASK_VIP1_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
4	INT_MASK_VIP1_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
3	INT_MASK_VIP1_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
2	INT_MASK_VIP1_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
1	INT_MASK_GRPX3_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
0	INT_MASK_GRPX2_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Table 9-290. Register Call Summary for Register VIP_INT0_CLIENT1_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-291. VIP_INT0_LIST0_INT_STAT

Address Offset	0x0000 0088		
Physical Address	0x4897 D088 0x4899 D088 0x489B D088	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_STAT_CONTROL_DESCRIPTOR_INT15	INT_STAT_CONTROL_DESCRIPTOR_INT14	INT_STAT_CONTROL_DESCRIPTOR_INT13	INT_STAT_CONTROL_DESCRIPTOR_INT12	INT_STAT_CONTROL_DESCRIPTOR_INT11	INT_STAT_CONTROL_DESCRIPTOR_INT10	INT_STAT_CONTROL_DESCRIPTOR_INT9	INT_STAT_CONTROL_DESCRIPTOR_INT8	INT_STAT_CONTROL_DESCRIPTOR_INT7	INT_STAT_CONTROL_DESCRIPTOR_INT6	INT_STAT_CONTROL_DESCRIPTOR_INT5	INT_STAT_CONTROL_DESCRIPTOR_INT4	INT_STAT_CONTROL_DESCRIPTOR_INT3	INT_STAT_CONTROL_DESCRIPTOR_INT2	INT_STAT_CONTROL_DESCRIPTOR_INT1	INT_STAT_CONTROL_DESCRIPTOR_INT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_LIST7_NOTIFY	INT_STAT_LIST7_COMPLETE	INT_STAT_LIST6_NOTIFY	INT_STAT_LIST6_COMPLETE	INT_STAT_LIST5_NOTIFY	INT_STAT_LIST5_COMPLETE	INT_STAT_LIST4_NOTIFY	INT_STAT_LIST4_COMPLETE	INT_STAT_LIST3_NOTIFY	INT_STAT_LIST3_COMPLETE	INT_STAT_LIST2_NOTIFY	INT_STAT_LIST2_COMPLETE	INT_STAT_LIST1_NOTIFY	INT_STAT_LIST1_COMPLETE	INT_STAT_LIST0_NOTIFY	INT_STAT_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_STAT_CONTROL_DESCRIPTOR_INT15	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_CONTROL_DESCRIPTOR_INT14	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_CONTROL_DESCRIPTOR_INT13	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_CONTROL_DESCRIPTOR_INT12	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_CONTROL_DESCRIPTOR_INT11	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_CONTROL_DESCRIPTOR_INT10	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_CONTROL_DESCRIPTOR_INT9	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_CONTROL_DESCRIPTOR_INT8	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_CONTROL_DESCRIPTOR_INT7	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_CONTROL_DESCRIPTOR_INT6	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_CONTROL_DESCRIPTOR_INT5	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_CONTROL_DESCRIPTOR_INT4	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_CONTROL_DESCRIPTOR_INT3	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_CONTROL_DESCRIPTOR_INT2	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_CONTROL_DESCRIPTOR_INT1	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_CONTROL_DESCRIPTOR_INT0	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_LIST7_NOTIFY	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_LIST7_COMPLETE	List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_LIST6_NOTIFY	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_LIST6_COMPLETE	List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_LIST5_NOTIFY	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_LIST5_COMPLETE	List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_LIST4_NOTIFY	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_LIST4_COMPLETE	List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_STAT_LIST3_NOTIFY	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_LIST3_COMPLETE	List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_LIST2_NOTIFY	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_LIST2_COMPLETE	List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_LIST1_NOTIFY	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_LIST1_COMPLETE	List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_LIST0_NOTIFY	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_LIST0_COMPLETE	List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-292. Register Call Summary for Register VIP_INT0_LIST0_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0] [1] [2]

VIP Register Manual

- VIP VPDMA Register Summary: [3]

Table 9-293. VIP_INT0_LIST0_INT_MASK

Address Offset	0x0000 008C		
Physical Address	0x4897 D08C 0x4899 D08C 0x489B D08C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_MASK_CONTROL_DESCRIPTOR_INT15	INT_MASK_CONTROL_DESCRIPTOR_INT14	INT_MASK_CONTROL_DESCRIPTOR_INT13	INT_MASK_CONTROL_DESCRIPTOR_INT12	INT_MASK_CONTROL_DESCRIPTOR_INT11	INT_MASK_CONTROL_DESCRIPTOR_INT10	INT_MASK_CONTROL_DESCRIPTOR_INT9	INT_MASK_CONTROL_DESCRIPTOR_INT8	INT_MASK_CONTROL_DESCRIPTOR_INT7	INT_MASK_CONTROL_DESCRIPTOR_INT6	INT_MASK_CONTROL_DESCRIPTOR_INT5	INT_MASK_CONTROL_DESCRIPTOR_INT4	INT_MASK_CONTROL_DESCRIPTOR_INT3	INT_MASK_CONTROL_DESCRIPTOR_INT2	INT_MASK_CONTROL_DESCRIPTOR_INT1	INT_MASK_CONTROL_DESCRIPTOR_INT0	INT_MASK_LIST7_NOTIFY	INT_MASK_LIST7_COMPLETE	INT_MASK_LIST6_NOTIFY	INT_MASK_LIST6_COMPLETE	INT_MASK_LIST5_NOTIFY	INT_MASK_LIST5_COMPLETE	INT_MASK_LIST4_NOTIFY	INT_MASK_LIST4_COMPLETE	INT_MASK_LIST3_NOTIFY	INT_MASK_LIST3_COMPLETE	INT_MASK_LIST2_NOTIFY	INT_MASK_LIST2_COMPLETE	INT_MASK_LIST1_NOTIFY	INT_MASK_LIST1_COMPLETE	INT_MASK_LIST0_NOTIFY	INT_MASK_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_MASK_CONTROL_DESCRIPTOR_INT15	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_CONTROL_DESCRIPTOR_INT14	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_CONTROL_DESCRIPTOR_INT13	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_CONTROL_DESCRIPTOR_INT12	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_CONTROL_DESCRIPTOR_INT11	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_CONTROL_DESCRIPTOR_INT10	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_CONTROL_DESCRIPTOR_INT9	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_CONTROL_DESCRIPTOR_INT8	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_CONTROL_DESCRIPTOR_INT7	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_CONTROL_DESCRIPTOR_INT6	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_CONTROL_DESCRIPTOR_INT5	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_CONTROL_DESCRIPTOR_INT4	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_CONTROL_DESCRIPTOR_INT3	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_MASK_CONTROL_DESCRIPTOR_INT2	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_CONTROL_DESCRIPTOR_INT1	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_CONTROL_DESCRIPTOR_INT0	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_LIST7_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_LIST7_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_LIST6_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_LIST6_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_LIST5_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_LIST5_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_LIST4_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_LIST4_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_LIST3_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_LIST3_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_LIST2_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_LIST2_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_LIST1_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_LIST1_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_LIST0_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_LIST0_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-294. Register Call Summary for Register VIP_INT0_LIST0_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-295. VIP_INT1_CHANNEL0_INT_STAT

Address Offset	0x0000 0090		
Physical Address	0x4897 D090 0x4899 D090 0x489B D090	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX3	INT_STAT_GRPX2	INT_STAT_GRPX1	INT_STAT_SCALER_OUT	RESERVED								INT_STAT_SCALER_CHROMA	INT_STAT_SCALER_LUMA	INT_STAT_HQ_SCALER	RESERVED	INT_STAT_HQ_MV_OUT	RESERVED		INT_STAT_HQ_MV	RESERVED						INT_STAT_HQ_VID3_CHROMA	INT_STAT_HQ_VID3_LUMA	INT_STAT_HQ_VID2_CHROMA	INT_STAT_HQ_VID2_LUMA	INT_STAT_HQ_VID1_CHROMA	INT_STAT_HQ_VID1_LUMA

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX3	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_GRPX2	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_GRPX1	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_SCALER_OUT	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
27:20	RESERVED	Reserved	R	0x00
19	INT_STAT_SCALER_CHROMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_SCALER_LUMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_HQ_SCALER	The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_HQ_MV_OUT	The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_HQ_MV	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_HQ_VID3_CHROMA	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_HQ_VID3_LUMA	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_HQ_VID2_CHROMA	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
2	INT_STAT_HQ_VID2_LUMA	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_HQ_VID1_CHROMA	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_HQ_VID1_LUMA	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-296. Register Call Summary for Register VIP_INT1_CHANNEL0_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-297. VIP_INT1_CHANNEL0_INT_MASK

Address Offset	0x0000 0094		
Physical Address	0x4897 D094 0x4899 D094 0x489B D094	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX3	INT_MASK_GRPX2	INT_MASK_GRPX1	INT_MASK_SCALER_OUT	RESERVED								INT_MASK_SCALER_CHROMA	INT_MASK_SCALER_LUMA	INT_MASK_HQ_SCALER	RESERVED	INT_MASK_HQ_MV_OUT	RESERVED		INT_MASK_HQ_MV	RESERVED						INT_MASK_HQ_VID3_CHROMA	INT_MASK_HQ_VID3_LUMA	INT_MASK_HQ_VID2_CHROMA	INT_MASK_HQ_VID2_LUMA	INT_MASK_HQ_VID1_CHROMA	INT_MASK_HQ_VID1_LUMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX3	The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	INT_MASK_GRPX2	The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_GRPX1	The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_SCALER_OUT	The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_MASK_SCALER_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_SCALER_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_HQ_SCALER	The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_HQ_MV_OUT	The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_HQ_MV	The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_HQ_VID3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_HQ_VID3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_HQ_VID2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_HQ_VID2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_HQ_VID1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_HQ_VID1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-298. Register Call Summary for Register VIP_INT1_CHANNEL0_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-299. VIP_INT1_CHANNEL1_INT_STAT

Address Offset	0x0000 0098		
Physical Address	0x4897 D098 0x4899 D098 0x489B D098	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5-0
INT_STAT_VIP1_MULT_PORTB_SRC9	INT_STAT_VIP1_MULT_PORTB_SRC8	INT_STAT_VIP1_MULT_PORTB_SRC7	INT_STAT_VIP1_MULT_PORTB_SRC6	INT_STAT_VIP1_MULT_PORTB_SRC5	INT_STAT_VIP1_MULT_PORTB_SRC4	INT_STAT_VIP1_MULT_PORTB_SRC3	INT_STAT_VIP1_MULT_PORTB_SRC2	INT_STAT_VIP1_MULT_PORTB_SRC1	INT_STAT_VIP1_MULT_PORTB_SRC0	INT_STAT_VIP1_MULT_PORTA_SRC15	INT_STAT_VIP1_MULT_PORTA_SRC14	INT_STAT_VIP1_MULT_PORTA_SRC13	INT_STAT_VIP1_MULT_PORTA_SRC12	INT_STAT_VIP1_MULT_PORTA_SRC11	INT_STAT_VIP1_MULT_PORTA_SRC10	INT_STAT_VIP1_MULT_PORTA_SRC9	INT_STAT_VIP1_MULT_PORTA_SRC8	INT_STAT_VIP1_MULT_PORTA_SRC7	INT_STAT_VIP1_MULT_PORTA_SRC6	INT_STAT_VIP1_MULT_PORTA_SRC5	INT_STAT_VIP1_MULT_PORTA_SRC4	INT_STAT_VIP1_MULT_PORTA_SRC3	INT_STAT_VIP1_MULT_PORTA_SRC2	INT_STAT_VIP1_MULT_PORTA_SRC1	INT_STAT_VIP1_MULT_PORTA_SRC0	RESERVED

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
27	INT_STAT_VIP1_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP1_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP1_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	INT_STAT_VIP1_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 9-300. Register Call Summary for Register VIP_INT1_CHANNEL1_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-301. VIP_INT1_CHANNEL1_INT_MASK

Address Offset	0x0000 009C		
Physical Address	0x4897 D09C 0x4899 D09C 0x489B D09C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24
INT_MASK_VIP1_MULT_PORTB_SRC9	INT_MASK_VIP1_MULT_PORTB_SRC8	INT_MASK_VIP1_MULT_PORTB_SRC7	INT_MASK_VIP1_MULT_PORTB_SRC6	INT_MASK_VIP1_MULT_PORTB_SRC5	INT_MASK_VIP1_MULT_PORTB_SRC4	INT_MASK_VIP1_MULT_PORTB_SRC3	INT_MASK_VIP1_MULT_PORTB_SRC2

23	22	21	20	19	18	17	16
INT_MASK_VIP1_MULT_PORTB_SRC1	INT_MASK_VIP1_MULT_PORTB_SRC0	INT_MASK_VIP1_MULT_PORTA_SRC15	INT_MASK_VIP1_MULT_PORTA_SRC14	INT_MASK_VIP1_MULT_PORTA_SRC13	INT_MASK_VIP1_MULT_PORTA_SRC12	INT_MASK_VIP1_MULT_PORTA_SRC11	INT_MASK_VIP1_MULT_PORTA_SRC10

15	14	13	12	11	10	9	8
INT_MASK_VIP1_MULT_PORTA_SRC9	INT_MASK_VIP1_MULT_PORTA_SRC8	INT_MASK_VIP1_MULT_PORTA_SRC7	INT_MASK_VIP1_MULT_PORTA_SRC6	INT_MASK_VIP1_MULT_PORTA_SRC5	INT_MASK_VIP1_MULT_PORTA_SRC4	INT_MASK_VIP1_MULT_PORTA_SRC3	INT_MASK_VIP1_MULT_PORTA_SRC2

7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_PORTA_SRC1	INT_MASK_VIP1_MULT_PORTA_SRC0	RESERVED					

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_PORTB_SRC9	The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_PORTB_SRC8	The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_PORTB_SRC7	The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_PORTB_SRC6	The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_PORTB_SRC5	The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_PORTB_SRC4	The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_PORTB_SRC3	The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_PORTB_SRC2	The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_PORTB_SRC1	The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_PORTB_SRC0	The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_PORTA_SRC15	The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_PORTA_SRC14	The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_PORTA_SRC13	The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_MASK_VIP1_MULT_PORTA_SRC12	The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_PORTA_SRC11	The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_PORTA_SRC10	The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_PORTA_SRC9	The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_PORTA_SRC8	The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_PORTA_SRC7	The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_PORTA_SRC6	The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_PORTA_SRC5	The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_PORTA_SRC4	The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_PORTA_SRC3	The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_PORTA_SRC2	The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_PORTA_SRC1	The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_PORTA_SRC0	The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 9-302. Register Call Summary for Register VIP_INT1_CHANNEL1_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-303. VIP_INT1_CHANNEL2_INT_STAT

Address Offset	0x0000 00A0		
Physical Address	0x4897 D0A0 0x4899 D0A0 0x489B D0A0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_STAT_VIP1_MULT_ANCB_SRC9	INT_STAT_VIP1_MULT_ANCB_SRC8	INT_STAT_VIP1_MULT_ANCB_SRC7	INT_STAT_VIP1_MULT_ANCB_SRC6	INT_STAT_VIP1_MULT_ANCB_SRC5	INT_STAT_VIP1_MULT_ANCB_SRC4	INT_STAT_VIP1_MULT_ANCB_SRC3	INT_STAT_VIP1_MULT_ANCB_SRC2	INT_STAT_VIP1_MULT_ANCB_SRC1	INT_STAT_VIP1_MULT_ANCB_SRC0	INT_STAT_VIP1_MULT_ANCA_SRC15	INT_STAT_VIP1_MULT_ANCA_SRC14	INT_STAT_VIP1_MULT_ANCA_SRC13	INT_STAT_VIP1_MULT_ANCA_SRC12	INT_STAT_VIP1_MULT_ANCA_SRC11	INT_STAT_VIP1_MULT_ANCA_SRC10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_ANCA_SRC9	INT_STAT_VIP1_MULT_ANCA_SRC8	INT_STAT_VIP1_MULT_ANCA_SRC7	INT_STAT_VIP1_MULT_ANCA_SRC6	INT_STAT_VIP1_MULT_ANCA_SRC5	INT_STAT_VIP1_MULT_ANCA_SRC4	INT_STAT_VIP1_MULT_ANCA_SRC3	INT_STAT_VIP1_MULT_ANCA_SRC2	INT_STAT_VIP1_MULT_ANCA_SRC1	INT_STAT_VIP1_MULT_ANCA_SRC0	INT_STAT_VIP1_MULT_PORTB_SRC15	INT_STAT_VIP1_MULT_PORTB_SRC14	INT_STAT_VIP1_MULT_PORTB_SRC13	INT_STAT_VIP1_MULT_PORTB_SRC12	INT_STAT_VIP1_MULT_PORTB_SRC11	INT_STAT_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP1_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP1_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP1_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_VIP1_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-304. Register Call Summary for Register VIP_INT1_CHANNEL2_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-305. VIP_INT1_CHANNEL2_INT_MASK

Address Offset	0x0000 00A4		
Physical Address	0x4897 D0A4 0x4899 D0A4 0x489B D0A4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-305. VIP_INT1_CHANNEL2_INT_MASK (continued)

Description The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.

Type RW

Bit	Field
31	INT_MASK_VIP1_MULT_ANCB_SRC9
30	INT_MASK_VIP1_MULT_ANCB_SRC8
29	INT_MASK_VIP1_MULT_ANCB_SRC7
28	INT_MASK_VIP1_MULT_ANCB_SRC6
27	INT_MASK_VIP1_MULT_ANCB_SRC5
26	INT_MASK_VIP1_MULT_ANCB_SRC4
25	INT_MASK_VIP1_MULT_ANCB_SRC3
24	INT_MASK_VIP1_MULT_ANCB_SRC2
23	INT_MASK_VIP1_MULT_ANCB_SRC1
22	INT_MASK_VIP1_MULT_ANCB_SRC0
21	INT_MASK_VIP1_MULT_ANCA_SRC15
20	INT_MASK_VIP1_MULT_ANCA_SRC14
19	INT_MASK_VIP1_MULT_ANCA_SRC13
18	INT_MASK_VIP1_MULT_ANCA_SRC12
17	INT_MASK_VIP1_MULT_ANCA_SRC11
16	INT_MASK_VIP1_MULT_ANCA_SRC10
15	INT_MASK_VIP1_MULT_ANCA_SRC9
14	INT_MASK_VIP1_MULT_ANCA_SRC8
13	INT_MASK_VIP1_MULT_ANCA_SRC7
12	INT_MASK_VIP1_MULT_ANCA_SRC6
11	INT_MASK_VIP1_MULT_ANCA_SRC5
10	INT_MASK_VIP1_MULT_ANCA_SRC4
9	INT_MASK_VIP1_MULT_ANCA_SRC3
8	INT_MASK_VIP1_MULT_ANCA_SRC2
7	INT_MASK_VIP1_MULT_ANCA_SRC1
6	INT_MASK_VIP1_MULT_ANCA_SRC0
5	INT_MASK_VIP1_MULT_PORTB_SRC15
4	INT_MASK_VIP1_MULT_PORTB_SRC14
3	INT_MASK_VIP1_MULT_PORTB_SRC13
2	INT_MASK_VIP1_MULT_PORTB_SRC12
1	INT_MASK_VIP1_MULT_PORTB_SRC11
0	INT_MASK_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_ANCB_SRC9	The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_ANCB_SRC8	The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_ANCB_SRC7	The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_ANCB_SRC6	The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_ANCB_SRC5	The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_ANCB_SRC4	The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_ANCB_SRC3	The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_ANCB_SRC2	The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_ANCB_SRC1	The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_MASK_VIP1_MULT_ANCB_SRC0	The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_ANCA_SRC15	The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_ANCA_SRC14	The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_ANCA_SRC13	The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_ANCA_SRC12	The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_ANCA_SRC11	The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_ANCA_SRC10	The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_ANCA_SRC9	The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_ANCA_SRC8	The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_ANCA_SRC7	The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_ANCA_SRC6	The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_ANCA_SRC5	The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_ANCA_SRC4	The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_ANCA_SRC3	The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_ANCA_SRC2	The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_MASK_VIP1_MULT_ANCA_SRC1	The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_ANCA_SRC0	The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_PORTB_SRC15	The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_PORTB_SRC14	The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_PORTB_SRC13	The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_PORTB_SRC12	The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_PORTB_SRC11	The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_PORTB_SRC10	The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-306. Register Call Summary for Register VIP_INT1_CHANNEL2_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-307. VIP_INT1_CHANNEL3_INT_STAT

Address Offset	0x0000 00A8		
Physical Address	0x4897 D0A8 0x4899 D0A8 0x489B D0A8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP2_MULT_PORTB_SRC3	INT_STAT_VIP2_MULT_PORTB_SRC2	INT_STAT_VIP2_MULT_PORTB_SRC1	INT_STAT_VIP2_MULT_PORTB_SRC0	INT_STAT_VIP2_MULT_PORTA_SRC15	INT_STAT_VIP2_MULT_PORTA_SRC14	INT_STAT_VIP2_MULT_PORTA_SRC13	INT_STAT_VIP2_MULT_PORTA_SRC12	INT_STAT_VIP2_MULT_PORTA_SRC11	INT_STAT_VIP2_MULT_PORTA_SRC10	INT_STAT_VIP2_MULT_PORTA_SRC9	INT_STAT_VIP2_MULT_PORTA_SRC8	INT_STAT_VIP2_MULT_PORTA_SRC7	INT_STAT_VIP2_MULT_PORTA_SRC6	INT_STAT_VIP2_MULT_PORTA_SRC5	INT_STAT_VIP2_MULT_PORTA_SRC4	INT_STAT_VIP2_MULT_PORTA_SRC3	INT_STAT_VIP2_MULT_PORTA_SRC2	INT_STAT_VIP2_MULT_PORTA_SRC1	INT_STAT_VIP2_MULT_PORTA_SRC0	INT_STAT_VIP1_PORTB_RGB	INT_STAT_VIP1_PORTA_RGB	INT_STAT_VIP1_PORTB_CHROMA	INT_STAT_VIP1_PORTB_LUMA	INT_STAT_VIP1_PORTA_CHROMA	INT_STAT_VIP1_PORTA_LUMA	INT_STAT_VIP1_MULT_ANCB_SRC15	INT_STAT_VIP1_MULT_ANCB_SRC14	INT_STAT_VIP1_MULT_ANCB_SRC13	INT_STAT_VIP1_MULT_ANCB_SRC12	INT_STAT_VIP1_MULT_ANCB_SRC11	INT_STAT_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
25	INT_STAT_VIP2_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_VIP2_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_PORTB_RGB	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_PORTA_RGB	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_PORTB_CHROMA	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	INT_STAT_VIP1_PORTB_LUMA	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_PORTA_CHROMA	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_PORTA_LUMA	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_VIP1_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-308. Register Call Summary for Register VIP_INT1_CHANNEL3_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-309. VIP_INT1_CHANNEL3_INT_MASK

Address Offset	0x0000 00AC		
Physical Address	0x4897 D0AC 0x4899 D0AC 0x489B D0AC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_MASK_VIP2_MULT_PORTB_SRC3	INT_MASK_VIP2_MULT_PORTB_SRC2	INT_MASK_VIP2_MULT_PORTB_SRC1	INT_MASK_VIP2_MULT_PORTB_SRC0	INT_MASK_VIP2_MULT_PORTA_SRC15	INT_MASK_VIP2_MULT_PORTA_SRC14	INT_MASK_VIP2_MULT_PORTA_SRC13	INT_MASK_VIP2_MULT_PORTA_SRC12	INT_MASK_VIP2_MULT_PORTA_SRC11	INT_MASK_VIP2_MULT_PORTA_SRC10	INT_MASK_VIP2_MULT_PORTA_SRC9	INT_MASK_VIP2_MULT_PORTA_SRC8	INT_MASK_VIP2_MULT_PORTA_SRC7	INT_MASK_VIP2_MULT_PORTA_SRC6	INT_MASK_VIP2_MULT_PORTA_SRC5	INT_MASK_VIP2_MULT_PORTA_SRC4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_PORTA_SRC3	INT_MASK_VIP2_MULT_PORTA_SRC2	INT_MASK_VIP2_MULT_PORTA_SRC1	INT_MASK_VIP2_MULT_PORTA_SRC0	INT_MASK_VIP1_PORTB_RGB	INT_MASK_VIP1_PORTA_RGB	INT_MASK_VIP1_PORTB_CHROMA	INT_MASK_VIP1_PORTB_LUMA	INT_MASK_VIP1_PORTA_CHROMA	INT_MASK_VIP1_PORTA_LUMA	INT_MASK_VIP1_MULT_ANCB_SRC15	INT_MASK_VIP1_MULT_ANCB_SRC14	INT_MASK_VIP1_MULT_ANCB_SRC13	INT_MASK_VIP1_MULT_ANCB_SRC12	INT_MASK_VIP1_MULT_ANCB_SRC11	INT_MASK_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_PORTB_SRC3	The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_PORTB_SRC2	The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_PORTB_SRC1	The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_PORTB_SRC0	The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
27	INT_MASK_VIP2_MULT_PORTA_SRC15	The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_PORTA_SRC14	The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_PORTA_SRC13	The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_PORTA_SRC12	The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_PORTA_SRC11	The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_PORTA_SRC10	The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_PORTA_SRC9	The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_PORTA_SRC8	The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_PORTA_SRC7	The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_PORTA_SRC6	The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_PORTA_SRC5	The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_PORTA_SRC4	The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_PORTA_SRC3	The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_PORTA_SRC2	The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_PORTA_SRC1	The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_PORTA_SRC0	The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_PORTB_RGB	The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_PORTA_RGB	The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	INT_MASK_VIP1_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_ANCB_SRC15	The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_ANCB_SRC14	The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_ANCB_SRC13	The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_ANCB_SRC12	The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_ANCB_SRC11	The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_ANCB_SRC10	The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-310. Register Call Summary for Register VIP_INT1_CHANNEL3_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-311. VIP_INT1_CHANNEL4_INT_STAT

Address Offset	0x0000 00B0		
Physical Address	0x4897 D0B0 0x4899 D0B0 0x489B D0B0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP2_MULT_ANCB_SRC3	INT_STAT_VIP2_MULT_ANCB_SRC2	INT_STAT_VIP2_MULT_ANCB_SRC1	INT_STAT_VIP2_MULT_ANCB_SRC0	INT_STAT_VIP2_MULT_ANCA_SRC15	INT_STAT_VIP2_MULT_ANCA_SRC14	INT_STAT_VIP2_MULT_ANCA_SRC13	INT_STAT_VIP2_MULT_ANCA_SRC12	INT_STAT_VIP2_MULT_ANCA_SRC11	INT_STAT_VIP2_MULT_ANCA_SRC10	INT_STAT_VIP2_MULT_ANCA_SRC9	INT_STAT_VIP2_MULT_ANCA_SRC8	INT_STAT_VIP2_MULT_ANCA_SRC7	INT_STAT_VIP2_MULT_ANCA_SRC6	INT_STAT_VIP2_MULT_ANCA_SRC5	INT_STAT_VIP2_MULT_ANCA_SRC4	INT_STAT_VIP2_MULT_ANCA_SRC3	INT_STAT_VIP2_MULT_ANCA_SRC2	INT_STAT_VIP2_MULT_ANCA_SRC1	INT_STAT_VIP2_MULT_ANCA_SRC0	INT_STAT_VIP2_MULT_PORTB_SRC15	INT_STAT_VIP2_MULT_PORTB_SRC14	INT_STAT_VIP2_MULT_PORTB_SRC13	INT_STAT_VIP2_MULT_PORTB_SRC12	INT_STAT_VIP2_MULT_PORTB_SRC11	INT_STAT_VIP2_MULT_PORTB_SRC10	INT_STAT_VIP2_MULT_PORTB_SRC9	INT_STAT_VIP2_MULT_PORTB_SRC8	INT_STAT_VIP2_MULT_PORTB_SRC7	INT_STAT_VIP2_MULT_PORTB_SRC6	INT_STAT_VIP2_MULT_PORTB_SRC5	INT_STAT_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP2_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP2_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP2_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP2_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-312. Register Call Summary for Register VIP_INT1_CHANNEL4_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-313. VIP_INT1_CHANNEL4_INT_MASK

Address Offset	0x0000 00B4		
Physical Address	0x4897 D0B4 0x4899 D0B4 0x489B D0B4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_MASK_VIP2_MULT_ANCB_SRC3	INT_MASK_VIP2_MULT_ANCB_SRC2	INT_MASK_VIP2_MULT_ANCB_SRC1	INT_MASK_VIP2_MULT_ANCB_SRC0	INT_MASK_VIP2_MULT_ANCA_SRC15	INT_MASK_VIP2_MULT_ANCA_SRC14	INT_MASK_VIP2_MULT_ANCA_SRC13	INT_MASK_VIP2_MULT_ANCA_SRC12	INT_MASK_VIP2_MULT_ANCA_SRC11	INT_MASK_VIP2_MULT_ANCA_SRC10	INT_MASK_VIP2_MULT_ANCA_SRC9	INT_MASK_VIP2_MULT_ANCA_SRC8	INT_MASK_VIP2_MULT_ANCA_SRC7	INT_MASK_VIP2_MULT_ANCA_SRC6	INT_MASK_VIP2_MULT_ANCA_SRC5	INT_MASK_VIP2_MULT_ANCA_SRC4	INT_MASK_VIP2_MULT_ANCA_SRC3	INT_MASK_VIP2_MULT_ANCA_SRC2	INT_MASK_VIP2_MULT_ANCA_SRC1	INT_MASK_VIP2_MULT_ANCA_SRC0	INT_MASK_VIP2_MULT_PORTB_SRC15	INT_MASK_VIP2_MULT_PORTB_SRC14	INT_MASK_VIP2_MULT_PORTB_SRC13	INT_MASK_VIP2_MULT_PORTB_SRC12	INT_MASK_VIP2_MULT_PORTB_SRC11	INT_MASK_VIP2_MULT_PORTB_SRC10	INT_MASK_VIP2_MULT_PORTB_SRC9	INT_MASK_VIP2_MULT_PORTB_SRC8	INT_MASK_VIP2_MULT_PORTB_SRC7	INT_MASK_VIP2_MULT_PORTB_SRC6	INT_MASK_VIP2_MULT_PORTB_SRC5	INT_MASK_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_ANCB_SRC3	The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_ANCB_SRC2	The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_ANCB_SRC1	The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_ANCB_SRC0	The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_ANCA_SRC15	The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_ANCA_SRC14	The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_ANCA_SRC13	The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_ANCA_SRC12	The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_ANCA_SRC11	The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_ANCA_SRC10	The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
21	INT_MASK_VIP2_MULT_ANCA_SRC9	The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_ANCA_SRC8	The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_ANCA_SRC7	The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_ANCA_SRC6	The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_ANCA_SRC5	The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_ANCA_SRC4	The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_ANCA_SRC3	The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_ANCA_SRC2	The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_ANCA_SRC1	The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_ANCA_SRC0	The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_PORTB_SRC15	The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_PORTB_SRC14	The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_PORTB_SRC13	The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_PORTB_SRC12	The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_PORTB_SRC11	The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_PORTB_SRC10	The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_MASK_VIP2_MULT_PORTB_SRC9	The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_PORTB_SRC8	The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_PORTB_SRC7	The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_PORTB_SRC6	The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_PORTB_SRC5	The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_PORTB_SRC4	The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-314. Register Call Summary for Register VIP_INT1_CHANNEL4_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-315. VIP_INT1_CHANNEL5_INT_STAT

Address Offset	0x0000 00B8		
Physical Address	0x4897 D0B8 0x4899 D0B8 0x489B D0B8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24
INT_STAT_TRANSCODE2_CHROMA	INT_STAT_TRANSCODE2_LUMA	INT_STAT_TRANSCODE1_CHROMA	INT_STAT_TRANSCODE1_LUMA	INT_STAT_AUX_IN	INT_STAT_PIP_FRAME	INT_STAT_POST_COMP_WR	INT_STAT_VBI_SD_VENC

23	22	21	20	19	18	17	16
RESERVED	INT_STAT_NF_LAST_CHROMA	INT_STAT_NF_LAST_LUMA	INT_STAT_NF_WRITE_CHROMA	INT_STAT_NF_WRITE_LUMA	INT_STAT_OTHER	INT_STAT_VIP2_PORTB_RGB	INT_STAT_VIP2_PORTA_RGB

15	14	13	12	11	10	9	8
INT_STAT_VIP2_PORTB_CHROMA	INT_STAT_VIP2_PORTB_LUMA	INT_STAT_VIP2_PORTA_CHROMA	INT_STAT_VIP2_PORTA_LUMA	INT_STAT_VIP2_MULT_ANCB_SRC15	INT_STAT_VIP2_MULT_ANCB_SRC14	INT_STAT_VIP2_MULT_ANCB_SRC13	INT_STAT_VIP2_MULT_ANCB_SRC12

7	6	5	4	3	2	1	0
INT_STAT_VIP2_MULT_ANCB_SRC11	INT_STAT_VIP2_MULT_ANCB_SRC10	INT_STAT_VIP2_MULT_ANCB_SRC9	INT_STAT_VIP2_MULT_ANCB_SRC8	INT_STAT_VIP2_MULT_ANCB_SRC7	INT_STAT_VIP2_MULT_ANCB_SRC6	INT_STAT_VIP2_MULT_ANCB_SRC5	INT_STAT_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_TRANSCODE2_CHROMA	The last write DMA transaction has completed for channel transcode2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_TRANSCODE2_LUMA	The last write DMA transaction has completed for channel transcode2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_TRANSCODE1_CHROMA	The last write DMA transaction has completed for channel transcode1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_TRANSCODE1_LUMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_AUX_IN	The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_PIP_FRAME	The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_POST_COMP_WR	The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VBI_SD_VENC	The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_NF_LAST_CHROMA	The last write DMA transaction has completed for channel nf_last_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_NF_LAST_LUMA	The last write DMA transaction has completed for channel nf_last_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_NF_WRITE_CHROMA	The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_NF_WRITE_LUMA	The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_OTHER	This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_PORTB_RGB	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_PORTA_RGB	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_PORTB_CHROMA	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_PORTB_LUMA	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	INT_STAT_VIP2_PORTA_CHROMA	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_PORTA_LUMA	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_VIP2_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-316. Register Call Summary for Register VIP_INT1_CHANNEL5_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-317. VIP_INT1_CHANNEL5_INT_MASK

Address Offset	0x0000 00BC		
Physical Address	0x4897 D0BC 0x4899 D0BC 0x489B D0BC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-317. VIP_INT1_CHANNEL5_INT_MASK (continued)

Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_OTHER	INT_MASK_VIP2_PORTB_RGB	INT_MASK_VIP2_PORTA_RGB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_PORTB_CHROMA	INT_MASK_VIP2_PORTB_LUMA	INT_MASK_VIP2_PORTA_CHROMA	INT_MASK_VIP2_PORTA_LUMA	INT_MASK_VIP2_MULT_ANCB_SRC15	INT_MASK_VIP2_MULT_ANCB_SRC14	INT_MASK_VIP2_MULT_ANCB_SRC13	INT_MASK_VIP2_MULT_ANCB_SRC12	INT_MASK_VIP2_MULT_ANCB_SRC11	INT_MASK_VIP2_MULT_ANCB_SRC10	INT_MASK_VIP2_MULT_ANCB_SRC9	INT_MASK_VIP2_MULT_ANCB_SRC8	INT_MASK_VIP2_MULT_ANCB_SRC7	INT_MASK_VIP2_MULT_ANCB_SRC6	INT_MASK_VIP2_MULT_ANCB_SRC5	INT_MASK_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_OTHER	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_ANCB_SRC15	The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_ANCB_SRC14	The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_ANCB_SRC13	The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_ANCB_SRC12	The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_ANCB_SRC11	The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_ANCB_SRC10	The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_ANCB_SRC9	The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_ANCB_SRC8	The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_MASK_VIP2_MULT_ANCB_SRC7	The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_ANCB_SRC6	The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_ANCB_SRC5	The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_ANCB_SRC4	The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-318. Register Call Summary for Register VIP_INT1_CHANNEL5_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-319. VIP_INT1_CLIENT0_INT_STAT

Address Offset	0x0000 00C8		
Physical Address	0x4897 D0C8 0x4899 D0C8 0x489B D0C8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_NF_READ	INT_MASK_VIP2_PORT_B_RGB	INT_MASK_VIP2_PORT_A_RGB	INT_STAT_DEI_HQ_MV_OUT	RESERVED		INT_STAT_DEI_HQ_MV_IN	RESERVED						INT_STAT_DEI_HQ_3_CHROMA	INT_STAT_DEI_HQ_3_LUMA	INT_STAT_DEI_HQ_2_CHROMA	INT_STAT_DEI_HQ_2_LUMA	INT_STAT_DEI_HQ_1_LUMA	INT_STAT_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_NF_READ	The interrupt for Noise Filter Input Data 422 Interleaved should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_STAT_DEI_HQ_MV_OUT	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_DEI_HQ_MV_IN	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_DEI_HQ_3_CHROMA	The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_DEI_HQ_3_LUMA	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_DEI_HQ_2_CHROMA	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_DEI_HQ_2_LUMA	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_DEI_HQ_1_LUMA	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_DEI_HQ_1_CHROMA	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-320. Register Call Summary for Register VIP_INT1_CLIENT0_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

Table 9-320. Register Call Summary for Register VIP_INT1_CLIENT0_INT_STAT (continued)

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-321. VIP_INT1_CLIENT0_INT_MASK

Address Offset	0x0000 00CC		
Physical Address	0x4897 D0CC 0x4899 D0CC 0x489B D0CC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX1_DATA	INT_MASK_COMP_WRBK	INT_MASK_SC_OUT	RESERVED								INT_MASK_SC_IN_LUMA	INT_MASK_SC_IN_CHROMA	INT_MASK_PIP_WRBK	INT_MASK_DEI_SC_OUT	RESERVED	INT_MASK_DEI_HQ_MV_OUT	RESERVED		INT_MASK_DEI_HQ_MV_IN	RESERVED						INT_MASK_DEI_HQ_3_CHROMA	INT_MASK_DEI_HQ_3_LUMA	INT_MASK_DEI_HQ_2_CHROMA	INT_MASK_DEI_HQ_2_LUMA	INT_MASK_DEI_HQ_1_LUMA	INT_MASK_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX1_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_COMP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_MASK_SC_IN_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_SC_IN_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_PIP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_DEI_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_DEI_HQ_MV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12	INT_MASK_DEI_HQ_MV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_DEI_HQ_3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_DEI_HQ_3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_DEI_HQ_2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_DEI_HQ_2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_DEI_HQ_1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_DEI_HQ_1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-322. Register Call Summary for Register VIP_INT1_CLIENT0_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-323. VIP_INT1_CLIENT1_INT_STAT

Address Offset	0x0000 00D0		
Physical Address	0x4897 D0D0 0x4899 D0D0 0x489B D0D0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	RESERVED	INT_STAT_VIP2_ANC_B	INT_STAT_VIP2_ANC_A	INT_STAT_VIP1_ANC_B	INT_STAT_VIP1_ANC_A	INT_STAT_TRANS2_LUMA	INT_STAT_TRANS2_CHROMA	INT_STAT_TRANS1_LUMA	INT_STAT_TRANS1_CHROMA	INT_STAT_HDMI_WRBK_OUT	INT_STAT_VPI_CTL	INT_STAT_VBI_SDVENC	RESERVED	INT_STAT_NF_420_UV_OUT	INT_STAT_NF_420_Y_OUT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_NF_420_UV_IN	INT_STAT_NF_420_Y_IN	INT_STAT_NF_422_IN	INT_STAT_GRPX3_ST	INT_STAT_GRPX2_ST	INT_STAT_GRPX1_ST	INT_STAT_VIP2_UP_UV	INT_STAT_VIP2_UP_Y	INT_STAT_VIP2_LO_UV	INT_STAT_VIP2_LO_Y	INT_STAT_VIP1_UP_UV	INT_STAT_VIP1_UP_Y	INT_STAT_VIP1_LO_UV	INT_STAT_VIP1_LO_Y	INT_STAT_GRPX3_DATA	INT_STAT_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
30	RESERVED	Reserved	R	0
29	INT_STAT_VIP2_ANC_B	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
28	INT_STAT_VIP2_ANC_A	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
27	INT_STAT_VIP1_ANC_B	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
26	INT_STAT_VIP1_ANC_A	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
25	INT_STAT_TRANS2_LUMA	The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
24	INT_STAT_TRANS2_CHROMA	The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
23	INT_STAT_TRANS1_LUMA	The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_TRANS1_CHROMA	The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
21	INT_STAT_HDMI_WRBK_OUT	The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
20	INT_STAT_VPI_CTL	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
19	INT_STAT_VBI_SDVENC	The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
18	RESERVED	Reserved	R	0
17	INT_STAT_NF_420_UV_OUT	The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
16	INT_STAT_NF_420_Y_OUT	The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
15	INT_STAT_NF_420_UV_IN	The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
14	INT_STAT_NF_420_Y_IN	The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
13	INT_STAT_NF_422_IN	The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
12	INT_STAT_GRPX3_ST	The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
11	INT_STAT_GRPX2_ST	The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
10	INT_STAT_GRPX1_ST	The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
9	INT_STAT_VIP2_UP_UV	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
8	INT_STAT_VIP2_UP_Y	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
7	INT_STAT_VIP2_LO_UV	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
6	INT_STAT_VIP2_LO_Y	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
5	INT_STAT_VIP1_UP_UV	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
4	INT_STAT_VIP1_UP_Y	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
3	INT_STAT_VIP1_LO_UV	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
2	INT_STAT_VIP1_LO_Y	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
1	INT_STAT_GRPX3_DATA	The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_GRPX2_DATA	The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Table 9-324. Register Call Summary for Register VIP_INT1_CLIENT1_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-325. VIP_INT1_CLIENT1_INT_MASK

Address Offset	0x0000 00D4		
Physical Address	0x4897 D0D4 0x4899 D0D4 0x489B D0D4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	RESERVED	INT_MASK_VIP2_ANC_B	INT_MASK_VIP2_ANC_A	INT_MASK_VIP1_ANC_B	INT_MASK_VIP1_ANC_A	INT_MASK_TRANS2_LUMA	INT_MASK_TRANS2_CHROMA	INT_MASK_TRANS1_LUMA	INT_MASK_TRANS1_CHROMA	INT_MASK_HDMI_WRBK_OUT	INT_MASK_VPI_CTL	INT_MASK_VBI_SDVENC	RESERVED	INT_MASK_NF_420_UV_OUT	INT_MASK_NF_420_Y_OUT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_NF_420_UV_IN	INT_MASK_NF_420_Y_IN	INT_MASK_NF_422_IN	INT_MASK_GRPX3_ST	INT_MASK_GRPX2_ST	INT_MASK_GRPX1_ST	INT_MASK_VIP2_UP_UV	INT_MASK_VIP2_UP_Y	INT_MASK_VIP2_LO_UV	INT_MASK_VIP2_LO_Y	INT_MASK_VIP1_UP_UV	INT_MASK_VIP1_UP_Y	INT_MASK_VIP1_LO_UV	INT_MASK_VIP1_LO_Y	INT_MASK_GRPX3_DATA	INT_MASK_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_MASK_VIP2_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
28	INT_MASK_VIP2_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
27	INT_MASK_VIP1_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
26	INT_MASK_VIP1_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
25	INT_MASK_TRANS2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
24	INT_MASK_TRANS2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
23	INT_MASK_TRANS1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
22	INT_MASK_TRANS1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
21	INT_MASK_HDMI_WRBK_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
20	INT_MASK_VPI_CTL	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
19	INT_MASK_VBI_SDVENC	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
18	RESERVED	Reserved	R	0
17	INT_MASK_NF_420_UV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
16	INT_MASK_NF_420_Y_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
15	INT_MASK_NF_420_UV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
14	INT_MASK_NF_420_Y_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
13	INT_MASK_NF_422_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
12	INT_MASK_GRPX3_ST	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
11	INT_MASK_GRPX2_ST	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
10	INT_MASK_GRPX1_ST	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
9	INT_MASK_VIP2_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
8	INT_MASK_VIP2_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
7	INT_MASK_VIP2_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
6	INT_MASK_VIP2_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
5	INT_MASK_VIP1_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
4	INT_MASK_VIP1_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
3	INT_MASK_VIP1_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
2	INT_MASK_VIP1_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
1	INT_MASK_GRPX3_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
0	INT_MASK_GRPX2_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Table 9-326. Register Call Summary for Register VIP_INT1_CLIENT1_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-327. VIP_INT1_LIST0_INT_STAT

Address Offset	0x0000 00D8		
Physical Address	0x4897 D0D8 0x4899 D0D8 0x489B D0D8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

Bit	Field
31	INT_STAT_CONTROL_DESCRIPTOR_INT15
30	INT_STAT_CONTROL_DESCRIPTOR_INT14
29	INT_STAT_CONTROL_DESCRIPTOR_INT13
28	INT_STAT_CONTROL_DESCRIPTOR_INT12
27	INT_STAT_CONTROL_DESCRIPTOR_INT11
26	INT_STAT_CONTROL_DESCRIPTOR_INT10
25	INT_STAT_CONTROL_DESCRIPTOR_INT9
24	INT_STAT_CONTROL_DESCRIPTOR_INT8
23	INT_STAT_CONTROL_DESCRIPTOR_INT7
22	INT_STAT_CONTROL_DESCRIPTOR_INT6
21	INT_STAT_CONTROL_DESCRIPTOR_INT5
20	INT_STAT_CONTROL_DESCRIPTOR_INT4
19	INT_STAT_CONTROL_DESCRIPTOR_INT3
18	INT_STAT_CONTROL_DESCRIPTOR_INT2
17	INT_STAT_CONTROL_DESCRIPTOR_INT1
16	INT_STAT_CONTROL_DESCRIPTOR_INT0
15	INT_STAT_LIST7_NOTIFY
14	INT_STAT_LIST7_COMPLETE
13	INT_STAT_LIST6_NOTIFY
12	INT_STAT_LIST6_COMPLETE
11	INT_STAT_LIST5_NOTIFY
10	INT_STAT_LIST5_COMPLETE
9	INT_STAT_LIST4_NOTIFY
8	INT_STAT_LIST4_COMPLETE
7	INT_STAT_LIST3_NOTIFY
6	INT_STAT_LIST3_COMPLETE
5	INT_STAT_LIST2_NOTIFY
4	INT_STAT_LIST2_COMPLETE
3	INT_STAT_LIST1_NOTIFY
2	INT_STAT_LIST1_COMPLETE
1	INT_STAT_LIST0_NOTIFY
0	INT_STAT_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_STAT_CONTROL_DESCRIPTOR_INT15	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_CONTROL_DESCRIPTOR_INT14	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_CONTROL_DESCRIPTOR_INT13	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_CONTROL_DESCRIPTOR_INT12	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_CONTROL_DESCRIPTOR_INT11	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_CONTROL_DESCRIPTOR_INT10	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_CONTROL_DESCRIPTOR_INT9	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_CONTROL_DESCRIPTOR_INT8	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_CONTROL_DESCRIPTOR_INT7	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_CONTROL_DESCRIPTOR_INT6	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_CONTROL_DESCRIPTOR_INT5	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_CONTROL_DESCRIPTOR_INT4	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_CONTROL_DESCRIPTOR_INT3	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_CONTROL_DESCRIPTOR_INT2	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_CONTROL_DESCRIPTOR_INT1	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_CONTROL_DESCRIPTOR_INT0	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_LIST7_NOTIFY	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_LIST7_COMPLETE	List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_LIST6_NOTIFY	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_LIST6_COMPLETE	List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_LIST5_NOTIFY	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_LIST5_COMPLETE	List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_LIST4_NOTIFY	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_LIST4_COMPLETE	List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_STAT_LIST3_NOTIFY	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_LIST3_COMPLETE	List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_LIST2_NOTIFY	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_LIST2_COMPLETE	List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_LIST1_NOTIFY	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_LIST1_COMPLETE	List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_LIST0_NOTIFY	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_LIST0_COMPLETE	List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 9-328. Register Call Summary for Register VIP_INT1_LIST0_INT_STAT

VIP Functional Description

- VPDMA Interrupts: [0] [1] [2]

VIP Register Manual

- VIP VPDMA Register Summary: [3]

Table 9-329. VIP_INT1_LIST0_INT_MASK

Address Offset	0x0000 00DC		
Physical Address	0x4897 D0DC 0x4899 D0DC 0x489B D0DC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_MASK_CONTROL_DESCRIPTOR_INT15	INT_MASK_CONTROL_DESCRIPTOR_INT14	INT_MASK_CONTROL_DESCRIPTOR_INT13	INT_MASK_CONTROL_DESCRIPTOR_INT12	INT_MASK_CONTROL_DESCRIPTOR_INT11	INT_MASK_CONTROL_DESCRIPTOR_INT10	INT_MASK_CONTROL_DESCRIPTOR_INT9	INT_MASK_CONTROL_DESCRIPTOR_INT8	INT_MASK_CONTROL_DESCRIPTOR_INT7	INT_MASK_CONTROL_DESCRIPTOR_INT6	INT_MASK_CONTROL_DESCRIPTOR_INT5	INT_MASK_CONTROL_DESCRIPTOR_INT4	INT_MASK_CONTROL_DESCRIPTOR_INT3	INT_MASK_CONTROL_DESCRIPTOR_INT2	INT_MASK_CONTROL_DESCRIPTOR_INT1	INT_MASK_CONTROL_DESCRIPTOR_INT0	INT_MASK_LIST7_NOTIFY	INT_MASK_LIST7_COMPLETE	INT_MASK_LIST6_NOTIFY	INT_MASK_LIST6_COMPLETE	INT_MASK_LIST5_NOTIFY	INT_MASK_LIST5_COMPLETE	INT_MASK_LIST4_NOTIFY	INT_MASK_LIST4_COMPLETE	INT_MASK_LIST3_NOTIFY	INT_MASK_LIST3_COMPLETE	INT_MASK_LIST2_NOTIFY	INT_MASK_LIST2_COMPLETE	INT_MASK_LIST1_NOTIFY	INT_MASK_LIST1_COMPLETE	INT_MASK_LIST0_NOTIFY	INT_MASK_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_MASK_CONTROL_DESCRIPTOR_INT15	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_CONTROL_DESCRIPTOR_INT14	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_CONTROL_DESCRIPTOR_INT13	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_CONTROL_DESCRIPTOR_INT12	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_CONTROL_DESCRIPTOR_INT11	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_CONTROL_DESCRIPTOR_INT10	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_CONTROL_DESCRIPTOR_INT9	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_CONTROL_DESCRIPTOR_INT8	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_CONTROL_DESCRIPTOR_INT7	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_CONTROL_DESCRIPTOR_INT6	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_CONTROL_DESCRIPTOR_INT5	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_CONTROL_DESCRIPTOR_INT4	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_CONTROL_DESCRIPTOR_INT3	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_MASK_CONTROL_DESCRIPTOR_INT2	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_CONTROL_DESCRIPTOR_INT1	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_CONTROL_DESCRIPTOR_INT0	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_LIST7_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_LIST7_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_LIST6_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_LIST6_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_LIST5_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_LIST5_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_LIST4_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_LIST4_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_LIST3_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_LIST3_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_LIST2_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_LIST2_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_LIST1_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_LIST1_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_LIST0_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_LIST0_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 9-330. Register Call Summary for Register VIP_INT1_LIST0_INT_MASK

VIP Functional Description

- VPDMA Interrupts: [0]

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-331. VIP_PERF_MON0

Address Offset	0x0000 0200		
Physical Address	0x4897 D200 0x4899 D200 0x489B D200	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: vip2_anc_b 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: vip2_anc_b 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-332. Register Call Summary for Register VIP_PERF_MON0

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-333. VIP_PERF_MON1

Address Offset	0x0000 0204		
Physical Address	0x4897 D204 0x4899 D204 0x489B D204	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-334. Register Call Summary for Register VIP_PERF_MON1

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-335. VIP_PERF_MON2

Address Offset	0x0000 0208		
Physical Address	0x4897 D208 0x4899 D208 0x489B D208	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-336. Register Call Summary for Register VIP_PERF_MON2

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-337. VIP_PERF_MON3

Address Offset	0x0000 020C		
Physical Address	0x4897 D20C 0x4899 D20C 0x489B D20C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-338. Register Call Summary for Register VIP_PERF_MON3

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-339. VIP_PERF_MON4

Address Offset	0x0000 0210		
Physical Address	0x4897 D210 0x4899 D210 0x489B D210	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-340. Register Call Summary for Register VIP_PERF_MON4

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-341. VIP_PERF_MON5

Address Offset	0x0000 0214		
Physical Address	0x4897 D214 0x4899 D214 0x489B D214	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-342. Register Call Summary for Register VIP_PERF_MON5

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-343. VIP_PERF_MON6

Address Offset	0x0000 0218		
Physical Address	0x4897 D218 0x4899 D218 0x489B D218	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-344. Register Call Summary for Register VIP_PERF_MON6

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-345. VIP_PERF_MON7

Address Offset	0x0000 021C		
Physical Address	0x4897 D21C 0x4899 D21C 0x489B D21C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-346. Register Call Summary for Register VIP_PERF_MON7

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-347. VIP_PERF_MON8

Address Offset	0x0000 0220		
Physical Address	0x4897 D220 0x4899 D220 0x489B D220	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-348. Register Call Summary for Register VIP_PERF_MON8

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-349. VIP_PERF_MON9

Address Offset	0x0000 0224		
Physical Address	0x4897 D224 0x4899 D224 0x489B D224	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-350. Register Call Summary for Register VIP_PERF_MON9

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-351. VIP_PERF_MON10

Address Offset	0x0000 0228		
Physical Address	0x4897 D228 0x4899 D228 0x489B D228	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-352. Register Call Summary for Register VIP_PERF_MON10

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-353. VIP_PERF_MON11

Address Offset	0x0000 022C		
Physical Address	0x4897 D22C 0x4899 D22C 0x489B D22C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-354. Register Call Summary for Register VIP_PERF_MON11

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-355. VIP_PERF_MON12

Address Offset	0x0000 0230		
Physical Address	0x4897 D230 0x4899 D230 0x489B D230	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-355. VIP_PERF_MON12 (continued)

Description	The register can be used to capture timing differences between events in the VPDMA\\n
Type	RW

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-356. Register Call Summary for Register VIP_PERF_MON12

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-357. VIP_PERF_MON13

Address Offset	0x0000 0234		
Physical Address	0x4897 D234 0x4899 D234 0x489B D234	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-358. Register Call Summary for Register VIP_PERF_MON13

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-359. VIP_PERF_MON14

Address Offset	0x0000 0238		
Physical Address	0x4897 D238 0x4899 D238 0x489B D238	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-360. Register Call Summary for Register VIP_PERF_MON14

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-361. VIP_PERF_MON15

Address Offset	0x0000 023C		
Physical Address	0x4897 D23C 0x4899 D23C 0x489B D23C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-362. Register Call Summary for Register VIP_PERF_MON15

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-363. VIP_PERF_MON16

Address Offset	0x0000 0240		
Physical Address	0x4897 D240 0x4899 D240 0x489B D240	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-364. Register Call Summary for Register VIP_PERF_MON16

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-365. VIP_PERF_MON17

Address Offset	0x0000 0244		
Physical Address	0x4897 D244 0x4899 D244 0x489B D244	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-366. Register Call Summary for Register VIP_PERF_MON17

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-367. VIP_PERF_MON18

Address Offset	0x0000 0248		
Physical Address	0x4897 D248 0x4899 D248 0x489B D248	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-368. Register Call Summary for Register VIP_PERF_MON18

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-369. VIP_PERF_MON19

Address Offset	0x0000 024C		
Physical Address	0x4897 D24C 0x4899 D24C 0x489B D24C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-370. Register Call Summary for Register VIP_PERF_MON19

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-371. VIP_PERF_MON20

Address Offset	0x0000 0250		
Physical Address	0x4897 D250 0x4899 D250 0x489B D250	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-372. Register Call Summary for Register VIP_PERF_MON20

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-373. VIP_PERF_MON21

Address Offset	0x0000 0254		
Physical Address	0x4897 D254 0x4899 D254 0x489B D254	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-374. Register Call Summary for Register VIP_PERF_MON21

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-375. VIP_PERF_MON22

Address Offset	0x0000 0258		
Physical Address	0x4897 D258 0x4899 D258 0x489B D258	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-376. Register Call Summary for Register VIP_PERF_MON22

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-377. VIP_PERF_MON23

Address Offset	0x0000 025C		
Physical Address	0x4897 D25C 0x4899 D25C 0x489B D25C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-378. Register Call Summary for Register VIP_PERF_MON23

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-379. VIP_PERF_MON24

Address Offset	0x0000 0260		
Physical Address	0x4897 D260 0x4899 D260 0x489B D260	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-380. Register Call Summary for Register VIP_PERF_MON24

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-381. VIP_PERF_MON25

Address Offset	0x0000 0264		
Physical Address	0x4897 D264 0x4899 D264 0x489B D264	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-382. Register Call Summary for Register VIP_PERF_MON25

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-383. VIP_PERF_MON26

Address Offset	0x0000 0268		
Physical Address	0x4897 D268 0x4899 D268 0x489B D268	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-384. Register Call Summary for Register VIP_PERF_MON26

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-385. VIP_PERF_MON27

Address Offset	0x0000 026C		
Physical Address	0x4897 D26C 0x4899 D26C 0x489B D26C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-385. VIP_PERF_MON27 (continued)

Description	The register can be used to capture timing differences between events in the VPDMA\\n
Type	RW

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-386. Register Call Summary for Register VIP_PERF_MON27

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-387. VIP_PERF_MON28

Address Offset	0x0000 0270		
Physical Address	0x4897 D270 0x4899 D270 0x489B D270	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-388. Register Call Summary for Register VIP_PERF_MON28

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-389. VIP_PERF_MON29

Address Offset	0x0000 0274		
Physical Address	0x4897 D274 0x4899 D274 0x489B D274	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-390. Register Call Summary for Register VIP_PERF_MON29

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-391. VIP_PERF_MON30

Address Offset	0x0000 0278		
Physical Address	0x4897 D278 0x4899 D278 0x489B D278	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-392. Register Call Summary for Register VIP_PERF_MON30

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-393. VIP_PERF_MON31

Address Offset	0x0000 027C		
Physical Address	0x4897 D27C 0x4899 D27C 0x489B D27C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-394. Register Call Summary for Register VIP_PERF_MON31

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-395. VIP_PERF_MON32

Address Offset	0x0000 0280		
Physical Address	0x4897 D280 0x4899 D280 0x489B D280	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3: vip1_lo_y	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3: vip1_lo_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-396. Register Call Summary for Register VIP_PERF_MON32

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-397. VIP_PERF_MON33

Address Offset	0x0000 0284		
Physical Address	0x4897 D284 0x4899 D284 0x489B D284	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: vip1_lo_y 3: vip1_lo_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: vip1_lo_y 3: vip1_lo_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-398. Register Call Summary for Register VIP_PERF_MON33

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-399. VIP_PERF_MON34

Address Offset	0x0000 0288		
Physical Address	0x4897 D288 0x4899 D288 0x489B D288	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_lo_y 1: 2: vip1_lo_uv 3: vip1_up_y	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_lo_y 1: 2: vip1_lo_uv 3: vip1_up_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-400. Register Call Summary for Register VIP_PERF_MON34

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-401. VIP_PERF_MON35

Address Offset	0x0000 028C		
Physical Address	0x4897 D28C 0x4899 D28C 0x489B D28C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_lo_uv 1: vip1_lo_y 2: vip1_up_y 3: vip1_up_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_lo_uv 1: vip1_lo_y 2: vip1_up_y 3: vip1_up_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-402. Register Call Summary for Register VIP_PERF_MON35

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-403. VIP_PERF_MON36

Address Offset	0x0000 0290		
Physical Address	0x4897 D290 0x4899 D290 0x489B D290	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_up_y 1: vip1_lo_uv 2: vip1_up_uv 3: vip2_lo_y	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_up_y 1: vip1_lo_uv 2: vip1_up_uv 3: vip2_lo_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-404. Register Call Summary for Register VIP_PERF_MON36

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-405. VIP_PERF_MON37

Address Offset	0x0000 0294		
Physical Address	0x4897 D294 0x4899 D294 0x489B D294	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_up_uv 1: vip1_up_y 2: vip2_lo_y 3: vip2_lo_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT		RW	0x0
23:22	RESERVED	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_up_uv 1: vip1_up_y 2: vip2_lo_y 3: vip2_lo_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-406. Register Call Summary for Register VIP_PERF_MON37

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-407. VIP_PERF_MON38

Address Offset	0x0000 0298		
Physical Address	0x4897 D298 0x4899 D298 0x489B D298	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_lo_y 1: vip1_up_uv 2: vip2_lo_uv 3: vip2_up_y	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_lo_y 1: vip1_up_uv 2: vip2_lo_uv 3: vip2_up_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-408. Register Call Summary for Register VIP_PERF_MON38

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-409. VIP_PERF_MON39

Address Offset	0x0000 029C		
Physical Address	0x4897 D29C 0x4899 D29C 0x489B D29C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		

Table 9-409. VIP_PERF_MON39 (continued)

Type	RW

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_lo_uv 1: vip2_lo_y 2: vip2_up_y 3: vip2_up_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_lo_uv 1: vip2_lo_y 2: vip2_up_y 3: vip2_up_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-410. Register Call Summary for Register VIP_PERF_MON39

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-411. VIP_PERF_MON40

Address Offset	0x0000 02A0		
Physical Address	0x4897 D2A0 0x4899 D2A0 0x489B D2A0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_up_y 1: vip2_lo_uv 2: vip2_up_uv 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_up_y 1: vip2_lo_uv 2: vip2_up_uv 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-412. Register Call Summary for Register VIP_PERF_MON40

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-413. VIP_PERF_MON41

Address Offset	0x0000 02A4		
Physical Address	0x4897 D2A4 0x4899 D2A4 0x489B D2A4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_up_uv 1: vip2_up_y 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_up_uv 1: vip2_up_y 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-414. Register Call Summary for Register VIP_PERF_MON41

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-415. VIP_PERF_MON42

Address Offset	0x0000 02A8		
Physical Address	0x4897 D2A8 0x4899 D2A8 0x489B D2A8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: vip2_up_uv 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: vip2_up_uv 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-416. Register Call Summary for Register VIP_PERF_MON42

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-417. VIP_PERF_MON43

Address Offset	0x0000 02AC		
Physical Address	0x4897 D2AC 0x4899 D2AC 0x489B D2AC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-417. VIP_PERF_MON43 (continued)

Description	The register can be used to capture timing differences between events in the VPDMA\\n
Type	RW

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-418. Register Call Summary for Register VIP_PERF_MON43

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-419. VIP_PERF_MON44

Address Offset	0x0000 02B0		
Physical Address	0x4897 D2B0 0x4899 D2B0 0x489B D2B0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-420. Register Call Summary for Register VIP_PERF_MON44

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-421. VIP_PERF_MON45

Address Offset	0x0000 02B4		
Physical Address	0x4897 D2B4 0x4899 D2B4 0x489B D2B4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-422. Register Call Summary for Register VIP_PERF_MON45

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-423. VIP_PERF_MON46

Address Offset	0x0000 02B8		
Physical Address	0x4897 D2B8 0x4899 D2B8 0x489B D2B8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-424. Register Call Summary for Register VIP_PERF_MON46

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-425. VIP_PERF_MON47

Address Offset	0x0000 02BC		
Physical Address	0x4897 D2BC 0x4899 D2BC 0x489B D2BC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-426. Register Call Summary for Register VIP_PERF_MON47

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-427. VIP_PERF_MON48

Address Offset	0x0000 02C0		
Physical Address	0x4897 D2C0 0x4899 D2C0 0x489B D2C0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-428. Register Call Summary for Register VIP_PERF_MON48

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-429. VIP_PERF_MON49

Address Offset	0x0000 02C4		
Physical Address	0x4897 D2C4 0x4899 D2C4 0x489B D2C4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-430. Register Call Summary for Register VIP_PERF_MON49

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-431. VIP_PERF_MON50

Address Offset	0x0000 02C8		
Physical Address	0x4897 D2C8 0x4899 D2C8 0x489B D2C8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3: vpi_ctl	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3: vpi_ctl	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-432. Register Call Summary for Register VIP_PERF_MON50

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-433. VIP_PERF_MON51

Address Offset	0x0000 02CC		
Physical Address	0x4897 D2CC 0x4899 D2CC 0x489B D2CC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: vpi_ctl 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: vpi_ctl 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-434. Register Call Summary for Register VIP_PERF_MON51

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-435. VIP_PERF_MON52

Address Offset	0x0000 02D0		
Physical Address	0x4897 D2D0 0x4899 D2D0 0x489B D2D0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vpi_ctl 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vpi_ctl 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-436. Register Call Summary for Register VIP_PERF_MON52

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-437. VIP_PERF_MON53

Address Offset	0x0000 02D4		
Physical Address	0x4897 D2D4 0x4899 D2D4 0x489B D2D4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: vpi_ctl 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: vpi_ctl 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-438. Register Call Summary for Register VIP_PERF_MON53

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-439. VIP_PERF_MON54

Address Offset	0x0000 02D8		
Physical Address	0x4897 D2D8 0x4899 D2D8 0x489B D2D8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-440. Register Call Summary for Register VIP_PERF_MON54

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-441. VIP_PERF_MON55

Address Offset	0x0000 02DC		
Physical Address	0x4897 D2DC 0x4899 D2DC 0x489B D2DC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-442. Register Call Summary for Register VIP_PERF_MON55

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-443. VIP_PERF_MON56

Address Offset	0x0000 02E0		
Physical Address	0x4897 D2E0 0x4899 D2E0 0x489B D2E0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT		RW	0x0
27	RESERVED	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3: vip1_anc_a	R	0x0
26:24	STOP_COUNT		RW	0x0
23:22	RESERVED	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	R	0x0
21:20	START_CLIENT		RW	0x0
19	RESERVED	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3: vip1_anc_a	R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-444. Register Call Summary for Register VIP_PERF_MON56

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-445. VIP_PERF_MON57

Address Offset	0x0000 02E4		
Physical Address	0x4897 D2E4 0x4899 D2E4 0x489B D2E4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: vip1_anc_a 3: vip1_anc_b	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: vip1_anc_a 3: vip1_anc_b	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-446. Register Call Summary for Register VIP_PERF_MON57

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-447. VIP_PERF_MON58

Address Offset	0x0000 02E8		
Physical Address	0x4897 D2E8 0x4899 D2E8 0x489B D2E8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-447. VIP_PERF_MON58 (continued)

Description	The register can be used to capture timing differences between events in the VPDMA\\n
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_anc_a 1: 2: vip1_anc_b 3: vip2_anc_a	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_anc_a 1: 2: vip1_anc_b 3: vip2_anc_a	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-448. Register Call Summary for Register VIP_PERF_MON58

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-449. VIP_PERF_MON59

Address Offset	0x0000 02EC		
Physical Address	0x4897 D2EC 0x4899 D2EC 0x489B D2EC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_anc_b 1: vip1_anc_a 2: vip2_anc_a 3: vip2_anc_b	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_anc_b 1: vip1_anc_a 2: vip2_anc_a 3: vip2_anc_b	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-450. Register Call Summary for Register VIP_PERF_MON59

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-451. VIP_PERF_MON60

Address Offset	0x0000 02F0		
Physical Address	0x4897 D2F0 0x4899 D2F0 0x489B D2F0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_anc_a 1: vip1_anc_b 2: vip2_anc_b 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_anc_a 1: vip1_anc_b 2: vip2_anc_b 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-452. Register Call Summary for Register VIP_PERF_MON60

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-453. VIP_PERF_MON61

Address Offset	0x0000 02F4		
Physical Address	0x4897 D2F4 0x4899 D2F4 0x489B D2F4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_anc_b 1: vip2_anc_a 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_anc_b 1: vip2_anc_a 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 9-454. Register Call Summary for Register VIP_PERF_MON61

VIP Register Manual

- VIP VPDMA Register Summary: [0]

Table 9-455. VIP0_LO_Y_CSTAT

Address Offset	0x0000 0388		
Physical Address	0x4897 D388 0x4899 D388 0x489B D388	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA

Table 9-455. VIP0_LO_Y_CSTAT (continued)

Description	The register holds status information and control for the client.\\n
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				RESERVED									

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-456. Register Call Summary for Register VIP0_LO_Y_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-457. VIP0_LO_UV_CSTAT

Address Offset	0x0000 038C		
Physical Address	0x4897 D38C 0x4899 D38C 0x489B D38C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\ \n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-458. Register Call Summary for Register VIP0_LO_UV_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-459. VIP0_UP_Y_CSTAT

Address Offset	0x0000 0390		
Physical Address	0x4897 D390 0x4899 D390 0x489B D390	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\ \n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-460. Register Call Summary for Register VIP0_UP_Y_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-461. VIP0_UP_UV_CSTAT

Address Offset	0x0000 0394		
Physical Address	0x4897 D394 0x4899 D394 0x489B D394	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				RESERVED									

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0

Bits	Field Name	Description	Type	Reset
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-462. Register Call Summary for Register VIP0_UP_UV_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-463. VIP1_LO_Y_CSTAT

Address Offset	0x0000 0398		
Physical Address	0x4897 D398 0x4899 D398 0x489B D398	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				RESERVED									

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0

Bits	Field Name	Description	Type	Reset
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-464. Register Call Summary for Register VIP1_LO_Y_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-465. VIP1_LO_UV_CSTAT

Address Offset	0x0000 039C		
Physical Address	0x4897 D39C 0x4899 D39C 0x489B D39C	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-466. Register Call Summary for Register VIP1_LO_UV_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-467. VIP1_UP_Y_CSTAT

Address Offset	0x0000 03A0		
Physical Address	0x4897 D3A0 0x4899 D3A0 0x489B D3A0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0

Bits	Field Name	Description	Type	Reset
9:0	RESERVED		R	0x0

Table 9-468. Register Call Summary for Register VIP1_UP_Y_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-469. VIP1_UP_UV_CSTAT

Address Offset	0x0000 03A4		
Physical Address	0x4897 D3A4 0x4899 D3A4 0x489B D3A4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-470. Register Call Summary for Register VIP1_UP_UV_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

Table 9-470. Register Call Summary for Register VIP1_UP_UV_CSTAT (continued)

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-471. VPI_CTL_CSTAT

Address Offset	0x0000 03D0		
Physical Address	0x4897 D3D0 0x4899 D3D0 0x489B D3D0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-472. Register Call Summary for Register VPI_CTL_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-473. VIP0_ANC_A_CSTAT

Address Offset	0x0000 03E8

Table 9-473. VIP0_ANC_A_CSTAT (continued)

Physical Address	0x4897 D3E8 0x4899 D3E8 0x489B D3E8	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-474. Register Call Summary for Register VIP0_ANC_A_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-475. VIP0_ANC_B_CSTAT

Address Offset	0x0000 03EC		
Physical Address	0x4897 D3EC 0x4899 D3EC 0x489B D3EC	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-476. Register Call Summary for Register VIP0_ANC_B_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-477. VIP1_ANC_A_CSTAT

Address Offset	0x0000 03F0		
Physical Address	0x4897 D3F0 0x4899 D3F0 0x489B D3F0	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				RESERVED									

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-478. Register Call Summary for Register VIP1_ANC_A_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Table 9-479. VIP1_ANC_B_CSTAT

Address Offset	0x0000 03F4		
Physical Address	0x4897 D3F4 0x4899 D3F4 0x489B D3F4	**Instance**	VIP1_VPDMA VIP2_VPDMA VIP3_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0

Bits	Field Name	Description	Type	Reset
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\ \n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 9-480. Register Call Summary for Register VIP1_ANC_B_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [1]

Chapter 10

Video Processing Engine

This chapter describes the video processing engine (VPE) for the device.

10.1 VPE Overview.......... 2305
10.2 VPE Integration.......... 2306
10.3 VPE Functional Description.......... 2308
10.4 VPE Register Manual.......... 2400

10.1 VPE Overview

VPE Features:

- Supports memory to memory operations only.
- VPE consist of a single memory to memory path which can perform the following operations:
  - Read of raster or tiled YUV420 coplanar, YUV422 coplanar or YUV422 interleaved video
  - Deinterlacing of the input video using a 4 field motion based algorithm
  - Scaling of the input video up to 1080p (1920x1080) resolution
  - Write of the resulting video in YUV420 coplanar (raster or tiled), YUV422 coplanar (raster or tiled), YUV422 interleaved (raster or tiled), YUV444 single plane (raster only) or RGB888 (raster only)
  - Deinterlacing up to two 1080i video sources.
  - The single data path performs operations in the following order
    - Chroma Upsampling from 420 to 422 (if needed)
    - Deinterlacing of 422 video from interlaced to progressive (if needed)
    - Scaling of 422 video after deinterlace
    - Conversion of 422 video to 420, 444 or RGB (if needed)
  - VC-1 Range Mapping and Range Reduction support on input video before Chroma Upsampling (if needed)
- Chroma Upsampling Features
  - 4 line Catmull-Rom based implementation
  - Programmable coefficients for interlaced or progressive conversion. Separate coefficients can be provided for top and bottom fields
- Deinterlacer Features
  - 8-bit, YCbCr 4:2:2
  - Motion-adaptive deinterlacing (MDT)
    - Motion detection is based on Luma only
    - 4-field data is used
    - Motion values adaptive to the frequency of luma texture
  - Edge-Directed Interpolation (EDI)
    - Edge detection using luma pixels in a 2x7 window
    - Seven edge vectors: -1.5, -1, -0.5, 0, 0.5, 1, 1.5
    - Edge-directed chroma interpolation
    - Soft-switch between edge directed interpolation and vertical interpolation depending on the confidence factor
  - Film Mode Detection (FMD)
    - 3-2 pull down detection
    - 2-2 pull down detection
    - Hysteresis controls how fast FMD can enter/exit film mode (software function)
    - Bad Edit Detection (BED)
  - Progessive Input
    - For Progressive Input, the module passes input to output. No internal processing is performed. This is essentially a bypass mode
  - Interlace Bypass
    - For Interlace Input, the module can pass the inputs data directly to the outputs in a bypass configuration. No internal processing is performed
- Scaler Features
  - Vertical and horizontal up/down scaling
    - Polyphase filter upscaling
  - Running average vertical down scaling for memory optimization
  - Decimation and polyphase filtering for horizontal scaling
  - Non-linear scaling for stretched/compressed left and right sides
  - Input image trimmer for pan/scan support
  - Pre-scaling peaking filter for enhanced sharpness

- Scale field as frame
- Interlacing of scaled output
- Full 1080p input and output support
- YCbCr422 input and output
- Minimum horizontal scaling ratio = 1/8x
- Maximum horizontal scaling ratio – limited by output line buffer (2014 pixels)
- Scaling filter Coefficient memory download

• Chroma Downsampler Features
  - Simple two-line averager capable of converting from YUV422 to YUV420 space

• 422 to 444 Features
  - Catmull-Rom based filter
  - 4 pixel, fixed coefficient

• Color Space Converter Features
  - Fully programmable 3x3 matrix multiplier with offset control

10.2 VPE Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests. Figure 10-1 summarizes the integration of the module in the device.

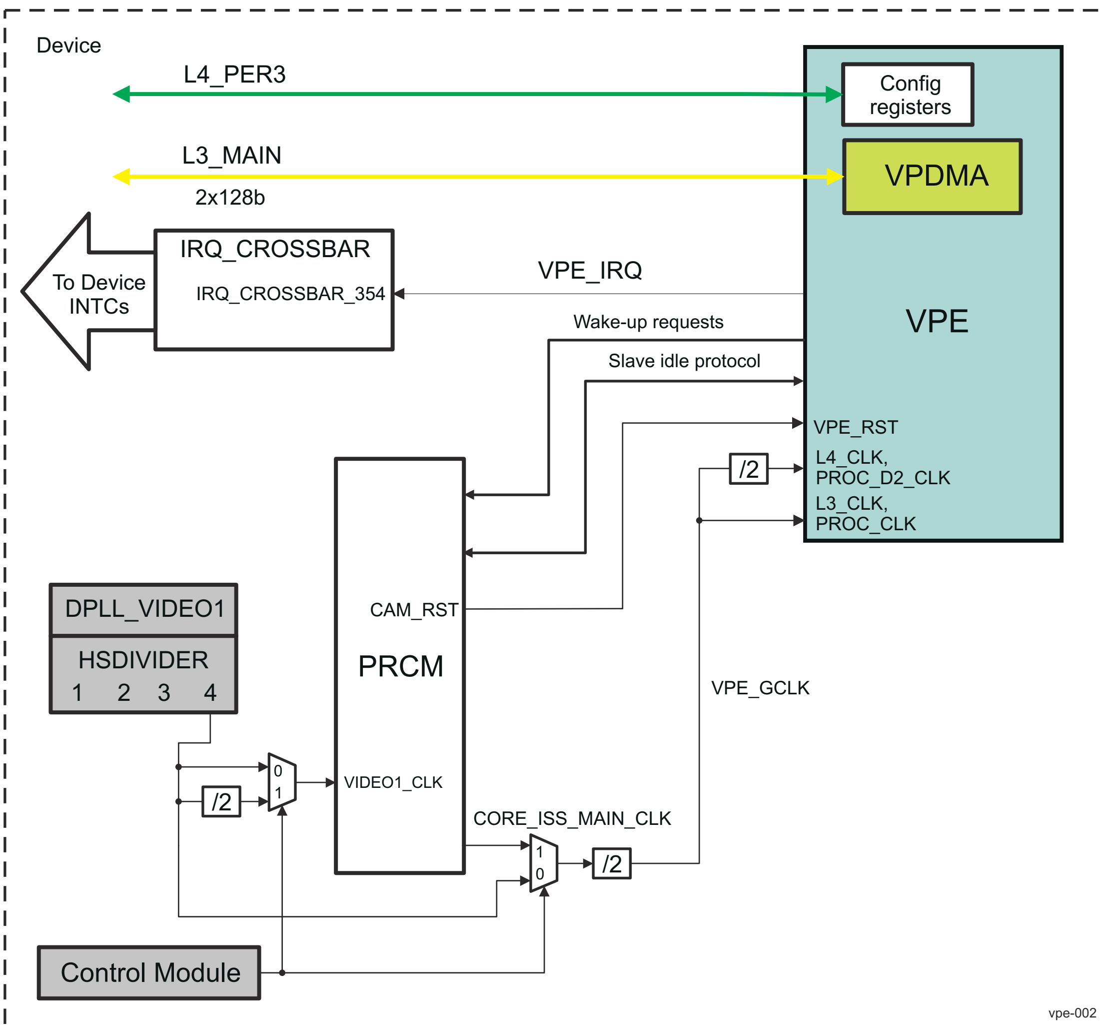


Figure 10-1. VPE Integration

Note

Alternative clocking to the VPE module is provided from DPLL_VIDEO1. For more information about configuring DPLL_VIDEO1, see Section 11.1.2.1, *Display Subsystem Clocks*. Source clock selection is done with CTRL_CORE_SMA_SW_1[8] VPE_CLK_DIV_BY_2_EN register from the Control Module. For more information, see *Control Module*.

Table 10-1 and Table 10-2 list the integration attributes and clock and resets, respectively.

Table 10-1. VPE Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
VPE	PD_VPE	L4_PER3 for configuration L3_MAIN for data (through VPDMA module)

Table 10-2. VPE Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
VPE	L3_CLK PROC_CLK	VPE_GCLK	PRCM	L3_CLK is the clock used to drive and receive data over the bus to L3_MAIN. The VPDMA uses this clock to send and receive external data and transfer this data to internal processing. PROC_CLK is the clock used to drive data processing within the VPE subsystem.
	L4_CLK PROC_D2_CLK	VPE_GCLK/2	PRCM	L4_CLK is the interface clock for Memory Mapped Registers (MMR) configuration bus PROC_D2_CLK is an additional clock used by the DEI within the DEI Subsystem. Inputs and outputs of the DEI operate on PROC_CLK, but internally, the data paths within the DEI operate on this clock.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VPE	VPE_RST	VPE_RST	PRCM	VPE Reset

Table 10-3. VPE Hardware Requests

Interrupt Requests				
Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
VPE	VPE_IRQ	IRQ_CROSSBAR_354	N/A	VPE interrupt requests. These IRQ source signals are not mapped by default to any device INTC.

Note

The "**Default Mapping**" column in Table 10-3 *VPE Hardware Requests* shows the default mapping of module IRQ source signals. These module IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

10.3 VPE Functional Description

10.3.1 VPE Block Diagram

Figure 10-2 shows the internal structure of the VPE module in the device.

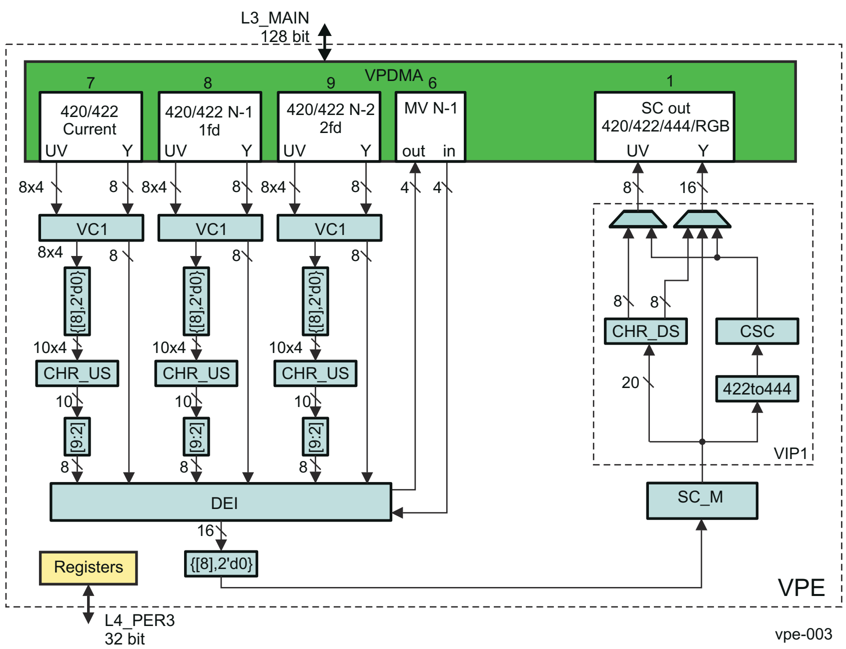


Figure 10-2. VPE Block Diagram

10.3.2 VPE VC1 Range Mapping/Range Reduction

VC1 range mapping and range reduction is implemented prior to the Chroma Upsampler (CHR_US) in the Primary Input Paths of the module.

Range Mapping is performed by setting VPE_CLKC_RANGE_MAP[6] RANGE_MAP_PRIM_ON bit to '1'. The output for every component is calculated by the follwing formulas:

Y[n] = (((Y[n] – 128) * (RANGE_MAPY_PRIM + 9) + 4) >> 3) + 128

Cb[n] = (((Cb[n] – 128) * (RANGE_MAPUV_PRIM + 9) + 4) >> 3) + 128

Cr[n] = (((Cr[n] – 128) * (RANGE_MAPUV_PRIM + 9) + 4) >> 3) + 128

In the above, VPE_CLKC_RANGE_MAP[2:0] RANGE_MAPY_PRIM and VPE_CLKC_RANGE_MAP[5:3] RANGE_MAPUV_PRIM are defined as in the register descriptions for each instantiation of this function.

Range Reduction is performed based by setting VPE_CLKC_RANGE_MAP[28] RANGE_REDUCTION_PRIM_ON bit to '1'. The output for the color components is calculated by the follwing formulas:

Y[n] =(Y[n] – 128) * 2 + 128

Cb[n] =(Cb[n] – 128) * 2 + 128

Cr[n] = (Cr[n] – 128) * 2 + 128

Note

The block performs Range Mapping first, and the output of Range Mapping drives Range Reduction. Although Range Mapping and Range Reduction are supposed to be mutually exclusive, the implementation allows both to be done simultaneous.

10.3.3 VPE Deinterlacer (DEI)

10.3.3.1 Functional Description

Figure 10-3 illustrates the block-diagram of motion-adaptive Deinterlacer. The general concept behind motion adaptive deinterlacing is that spatial filtering works very well for images with motion, while temporal filtering works very well for static images. So, the intuitive way is to combine them together. Motion detection is used to switch or fade between the use of spatial deinterlacing and temporal deinterlacing, as shown in the following formula:

$\hat{y}(j, i, n) = \alpha y_{spat}(j, i, n) + (1 - \alpha)y_{temp}(j, i, n)$

where $y_{spat}(j, i, n)$ is the spatial interpolation output, $y_{temp}(j, i, n)$ is temporal interpolation output, α is the motion detection output ranging from 0 to 1, $\hat{y}(j, i, n)$ is the final output from deinterlacer, and j, i, n are the vertical, horizontal, and temporal indexes, respectively. From the previous formula, the final output is controlled by the motion detector output, α. The higher the motion, the higher value of α, and the output favors spatial interpolation. If the motion is absent or very low, the temporal interpolation has higher weight.

Temporal interpolation can be disabled by programmable control registers. In that case we have:

$\hat{y}(j, i, n) = y_{spat}(j, i, n)$

Chroma interpolation is handled in the same manner as luma with regards to the above equations. There is a separate control for disabling temporal interpolation for chroma, such that luma can perform the full mixture, and chroma can only be spatial. If luma temporal interpolation is disabled, chroma temporal interpolation is also disabled.

Figure 10-3 provides a simple description of how a motion-adaptive deinterlacer operates. The actual interpolation used is edge directed interpolation. Edge directed interpolation is only performed spatially prior to the output mixing.

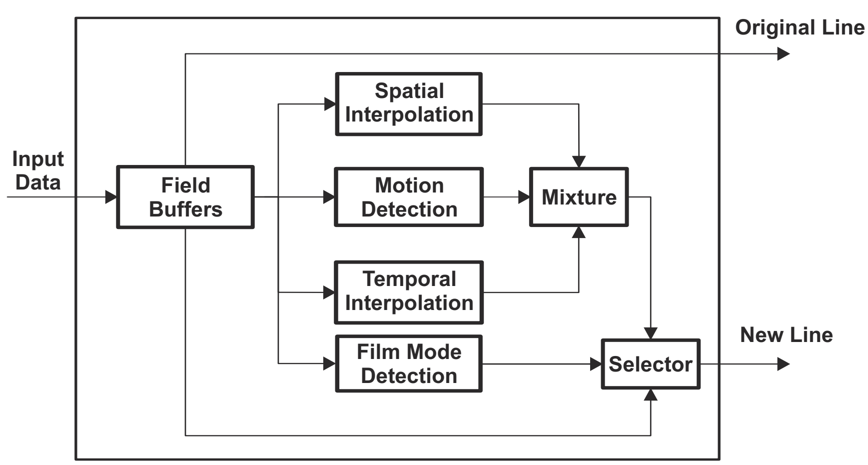


Figure 10-3. Block Diagram of Motion-Adaptive Deinterlacer

The Deinterlacer consists of:

- MMR Configuration Register Block
  - - Used to program configuration items for the Deinterlacer
- VPDMA Interface Block
  - Used to read source video/motion data from VPDMA
  - Used to write generated motion data to VPDMA
- Motion Detection (MDT) Block
  - Examines 3 fields of input video data (luma only) and calculates a 4 bit motion vector to drive the Edge Directed Interpolation Block
- Edge Directed Interpolation (EDI) Block
  - Performs the motion based edge directed interpolation on Luma and Chroma inputs to generate the missing line in the interlaced source
- Film Mode Detection (FMD) Block
  - Field Difference : Accumulated difference between the spatial interpolated frame and the adjacent input field of opposite field ID
  - Frame Difference : Accumulated difference between two fields with the same field ID
  - Combing Artifacts : Accumulated sum-of-difference between two adjacent fields
- Output Multiplexor (MUX) Block
- Line Buffer Block

10.3.3.2 Bypass Mode

The DEI can be operated in bypass mode and deinterlacer mode.

In the bypass mode, input luma and chroma are buffered and sent to the stage after DEI without processing. To bypass the DEI module registers VPE_DEI_REG0[31] PROGRESSIVE_BYPASS or VPE_DEI_REG0[29] INTERLACE_BYPASS must be set to '1' for progressive and interlaced source input respectivly.

10.3.3.3

10.3.3.3.1 VPDMA Interface

The VPDMA interface handles transactions between the internal core of the design and the VPI interface protocol, which is used for both external module input and motion vector output. All input data is pixel aligned, but has different request and ready signals. Motion Output is not pixel aligned with input.

The following are the VPDMA ports (from DEI to memory) available:

10.3.3.3.2 MDT

A block diagram representing the Motion Detection (MDT) Block is shown on Figure 10-4.

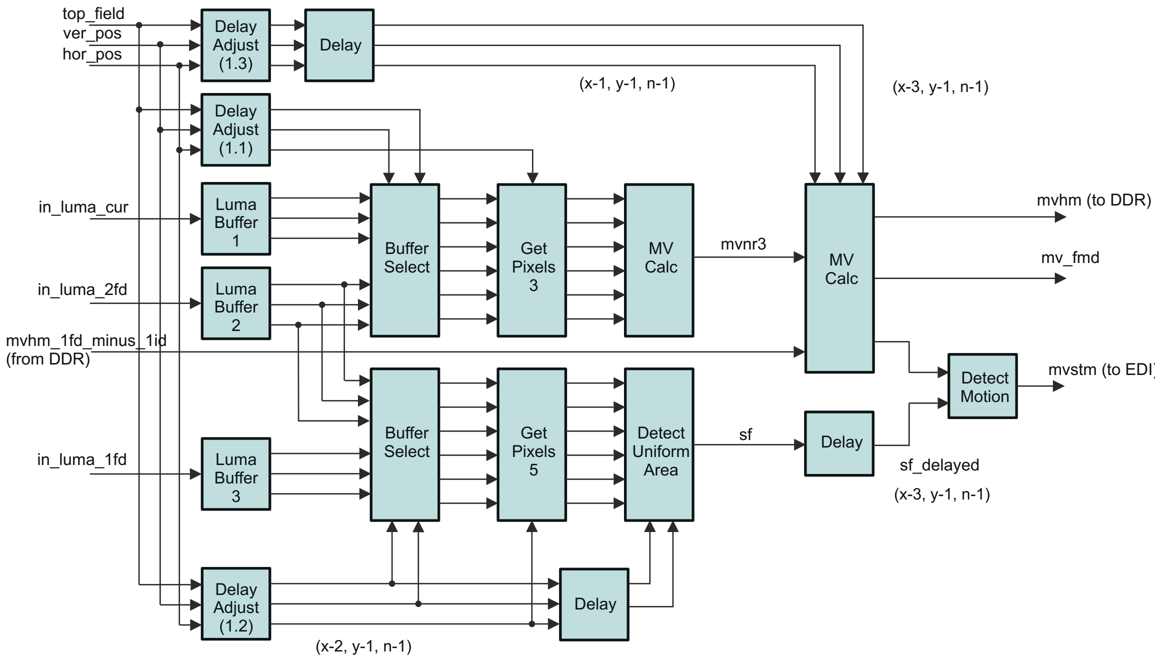


Figure 10-4. Motion Detection (MDT) Block Diagram

The Motion Detection block takes as input 3 adjacent fields of Luma data, motion data calculated from the previous 3 fields, and the current X, Y and field ID indicators and generates motion vector outputs which drive the Edge Directed Interpolation (EDI), Film Mode (FMD) and output motion values to DDR

There are two parallel paths - Motion Vector Calculation and Uniform Area Detection.

The Motion Vector Calculation path generates the FMD motion vector and the DDR output motion vector. The combination of the Motion Vector and Uniform Area Detection is used to calculate the EDI motion vector value.

The Motion Vector Calculation Path operates using a sliding 3 pixel wide by 2 pixel tall window over 2 adjacent fields of the same field ID polarity (2 top fields or 2 bottom fields). The first stage is to calculate the filtered motion value (MV Calc). The below diagram shows the dataflow for calculating the Filtered Motion Value from MV Calc:

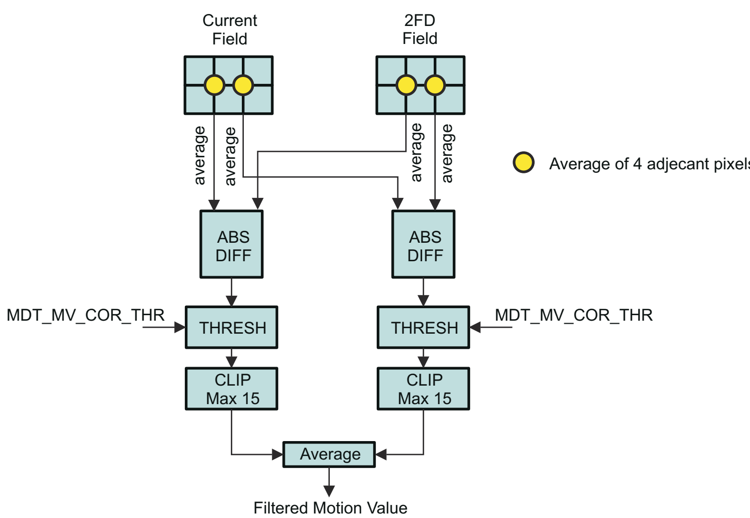


Figure 10-5. Motion Detection (MDT) MV Calc Data Path

The Filtered Motion Value is then operated on by the Max Filter operation, which performs a maximum operation between the last 3 filtered motion values and the last frame's motion value. The process is to calculate the maximum value of the last 3 filter motion values, and this value is written to DDR to be read back in on the next frame. The maximum value is taken between the previous frame's maximum value as read back from DDR, and the maximum value between the current last 3 motion values, and this is passed to the Detect Motion block which will drive the EDI module.

The Film Mode motion value output is just the output from the MV Calc data path. The following diagram shows the data path. Please note the line buffers, which are used to adjust for different polarity of fields (2 motion value fields are of opposite polarity, meaning a 1 line difference in one or the other field):

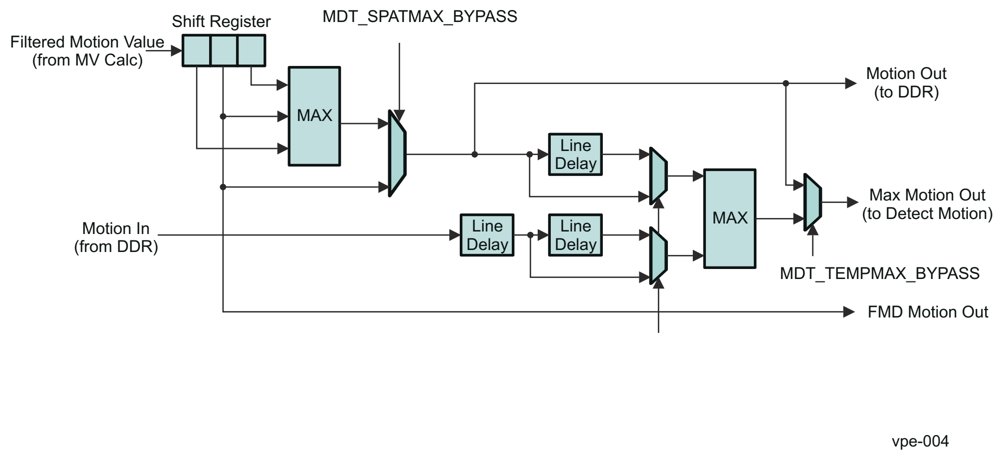


Figure 10-6. Motion Detection (MDT) Max Filter Data Path

The Uniform Area Detection Path operates using a sliding 5 pixel wide by 3 pixel tall window over 2 adjacent fields of opposite field ID polarity (1 field delay and 2 field delay) to calculate a spatial frequency component. The following diagram shows how this component is calculated within each pixel window:

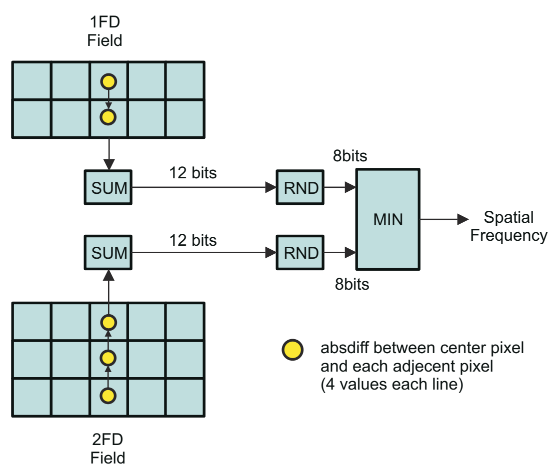


Figure 10-7. Motion Detection (MDT) Uniform Area Data Path

The spatial frequency output is used to scale the motion vector generated by the Maximum Filter operation based on thresholding against the MMR values VPE_DEI_REG2[7:0] MDT_SF_SC_THR1, VPE_DEI_REG2[15:8] MDT_SF_SC_THR2 and VPE_DEI_REG2[23:16] MDT_SF_SC_THR3. The following diagram (in the Detect Motion block) performs this thresholding to generate the final motion vector output to the EDI module:

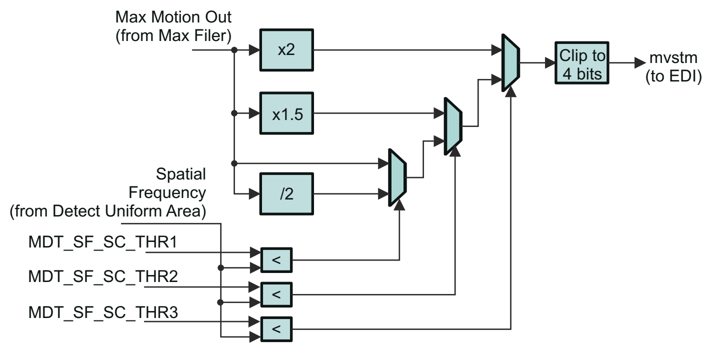


Figure 10-8. Motion Detection (MDT) Detect Motion Data Path

10.3.3.3.3 EDI

A block diagram representing the Edge Directed Interpolation (EDI) Block is shown on Figure 10-9

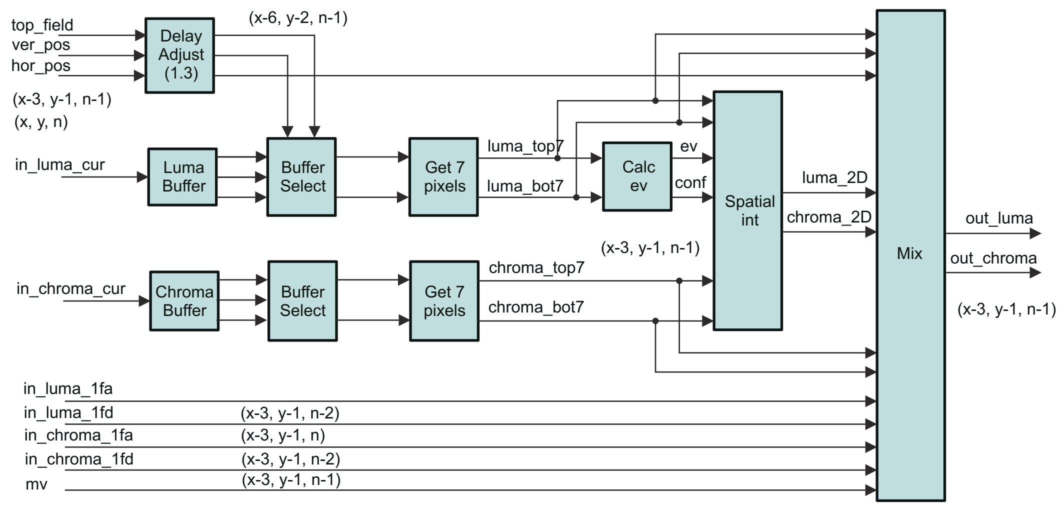


Figure 10-9. Edge Directed Interpolation (EDI) Block Diagram

The Edge Directed Interpolation module operates on a 7x2 window (7 pixels wide, 2 in height) in both Luma and Chroma. Luma data is used to calculate an edge vector and edge vector confidence. Correlation scaling factor is set with VPE_DEI_REG3[31:24] EDI_COR_SCALE_FACTOR register. A diagram showing this calculation follows (Calc ev):

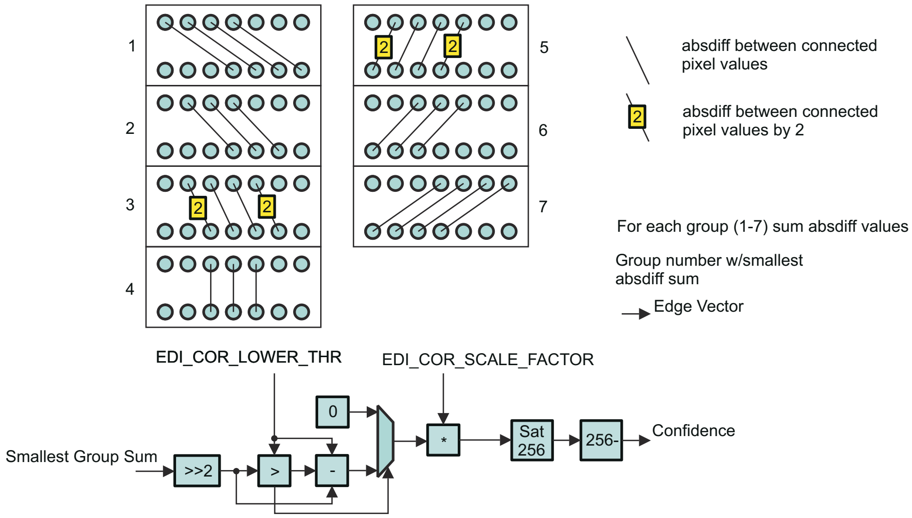


Figure 10-10. Edge Directed Interpolation Edge Vector Calculation

The edge vector and edge vector confidence are passed to the Spatial Interpolation module, which performs edge directed spatial interpolation. Following is a diagram showing how Luma is interpolated:

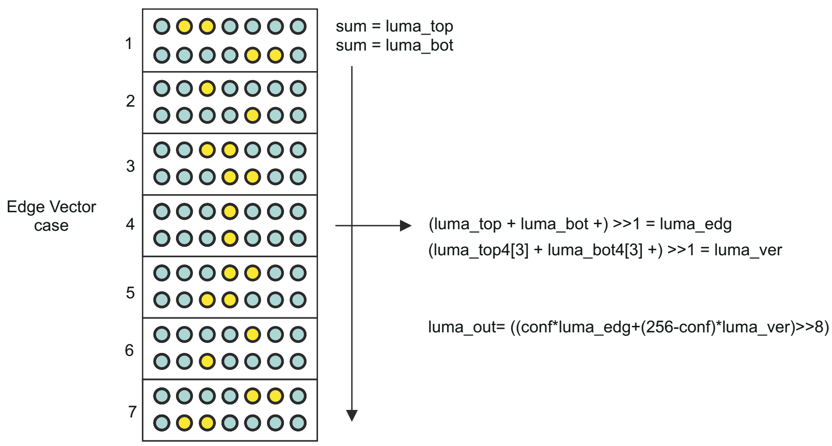


Figure 10-11. Edge Directed Interpolation Luma Interpolation Calculationn

The following diagram shows how Chroma is interpolated:

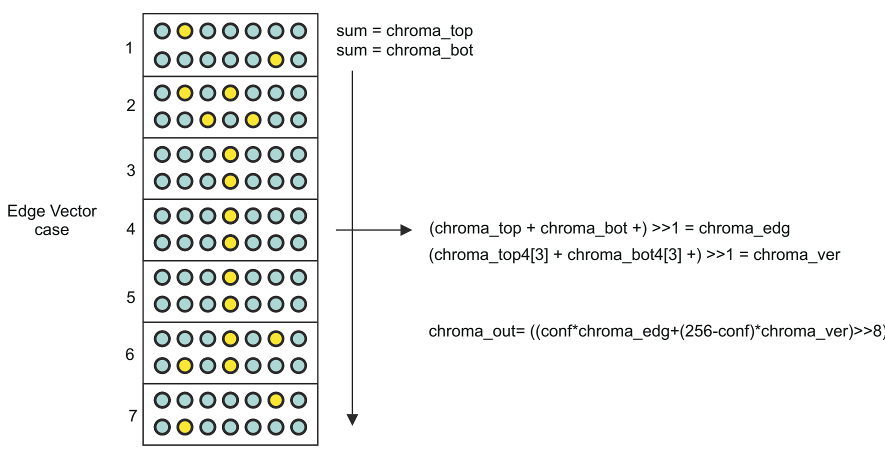


Figure 10-12. Edge Directed Interpolation Chroma Interpolation Calculation

Each of the cases shown above is on the 7x2 input pixel window for Luma and Chroma, based on the edge vector that was calculated. Each output equation (luma_out and chroma_out) is a combination of a straight vertical interpolation (luma/chroma_ver) and edge directed interpolation (luma/chroma_edge) using the Confidence factor output from the Edge Vector Calculation module. In the case of chroma interpolation, if the register VPE_DEI_REG3[1:0] EDI_INP_MODE = “11”, the chroma_out calculated for spatial interpolation is forced to the the vertical chroma output.

The Mix module does that actual mixing of spatial and temporal interpolation to produce the final result. 4 different interpolation modes are supported: Line double, Field double, 3D interpolation and 2D interpolation.

- Line double averages the top and bottom pixel to produce the interpolated pixel. Setting of MDT mode has no effect on output pictures.
- Field double averages the previous and next frame pixel to produce the interpolated pixel. In other words, if the current field is a top field, the interpolated bottom field picture is created by averaging pixels from bottom field pictures before and after the current field
- 2D interpolation uses the Edge Directed interpolation result only as the interpolated pixel
- 3D interpolation uses the EDI LUT table values with the MDT motion vector as an index to blend between the 2D result and the value from the previous field temporally.

Note

3D processing is enabled with VPE_DEI_REG3[2] EDI_ENABLE_3D register

The EDI Lut value selected based on the motion value will perform a blend between temporal and edge directed spatial interpolation using the equation:

$\hat{y}(j, i, n) = \alpha y_{spat}(j, i, n) + (1 - \alpha) y_{temp}(j, i, n)$

10.3.3.3.4 FMD

The following shows the block diagram of the Film Mode Detection (FMD) module:

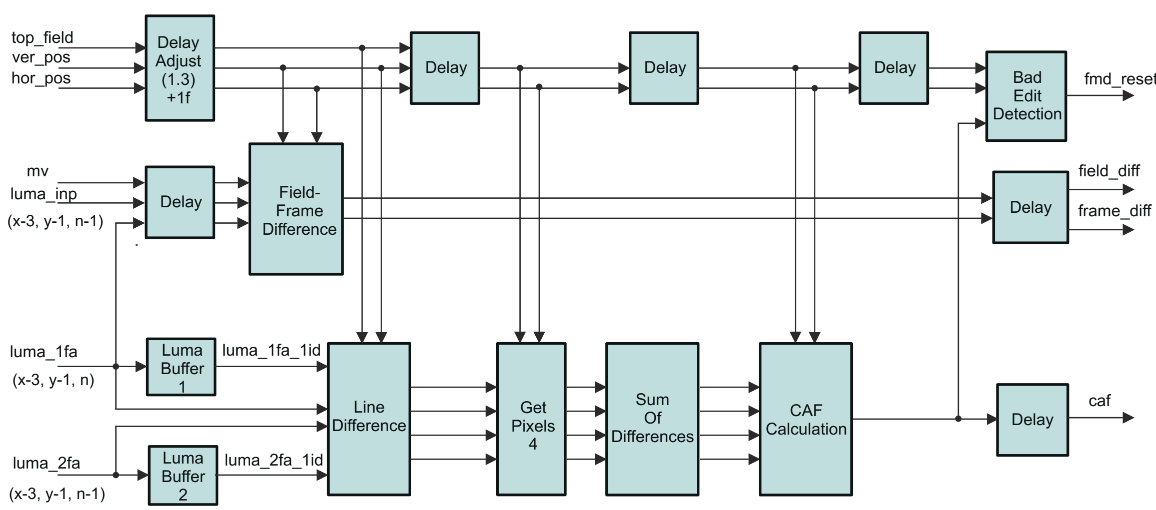


Figure 10-13. Film Mode Detection (FMD) Block Diagram

Film Mode Detection calculates four parameters which are used in conjuction with software to determine if the source of the video is film (3:2 or 2:2 sequences). This block is not part of the main data path, but calculates statistics based on the main data path for each frame generated, then interrupts the processor to read the information and then set to '1' the VPE_DEI_REG10[1] FMD_LOCK (lock to film mode) and VPE_DEI_REG10[2] FMD_JAM_DIR (direction of field jam) registers, to set the design to film mode operation.

All of the calculations are performed within a window defined by VPE_DEI_REG8[10:0] FMD_WINDOW_MINX, VPE_DEI_REG8[26:16] FMD_WINDOW_MAXX, VPE_DEI_REG9[10:0]

FMD_WINDOW_MINY, VPE_DEI_REG9[26:16] FMD_WINDOW_MAXY. All calculations are reset on each new input frame.

Field difference is calculated as the absolute difference (absdiff) between the current Luma input and the previous field Luma input. Value can be read on VPE_DEI_REG13[27:0] FMD_FIELD_DIFF register. Frame difference is calculated using the motion input from the MDT. Value can be read on VPE_DEI_REG14[19:0] FMD_FRAME_DIFF register. Figure 10-14 is showing how these are calculated is shown below:

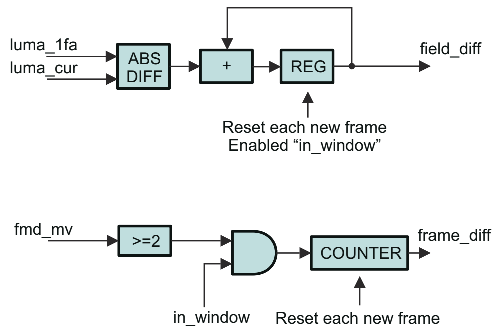


Figure 10-14. Film Mode Detection (FMD) Frame/Field Difference Calculation

---

Note

Value of FMD_WINDOW_MAXX must be less then WIDTH and value of FMD_WINDOW_MAXY must be less then ½ HEIGHT.

Combing Artifacts value is calculated as shown in Figure 10-15. Value can be read on VPE_DEI_REG12[20:0] FMD_CAF register.

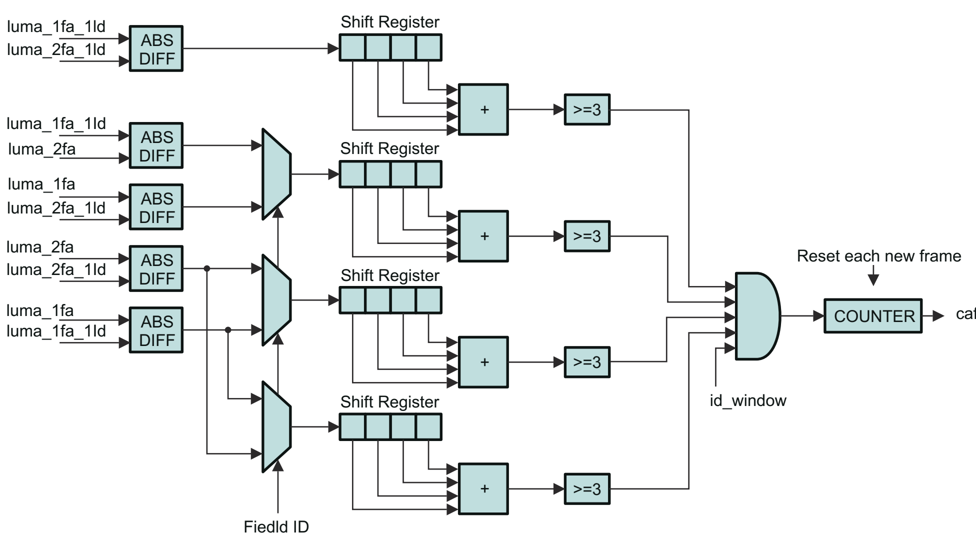


Figure 10-15. Film Mode Detection (FMD) Combing Artifacts Calculation

FMD_RESET is set to tell software when the Film Mode Detection should be unlocked back to normal operation. The logic controlling this is based on the caf output, and is set based on caf being above a threshold set by VPE_DEI_REG11[19:0] FMD_CAF_THR and where within the window the current x/y position is. If the position is over ¾ in height, and caf > FMD_CAF_THR OR if over ½ in height and caf > ¾*FMD_CAF_THR OR if over ¼ in height and caf > ¼*FMD_CAF_THR, then VPE_DEI_REG12[24] FMD_RESET must be set (it will clear at the start of the next field input).

The Film Mode Interrupt (fmd_int) will be generated at the end of the window defined for FMD operation and tells software to read the above parameters. When the interrupt fires, the above parameters are latched into the DEI Control Interface (MMR) so they can be read.

10.3.3.3.5 MUX

The MUX block generates the proper outputs based whether the design is in standard Deinterlacer modes (interpolated inputs from the EDI), Film Mode Deinterlacer mode (direct inputs) or progressive bypass mode. The following diagram shows this data path:

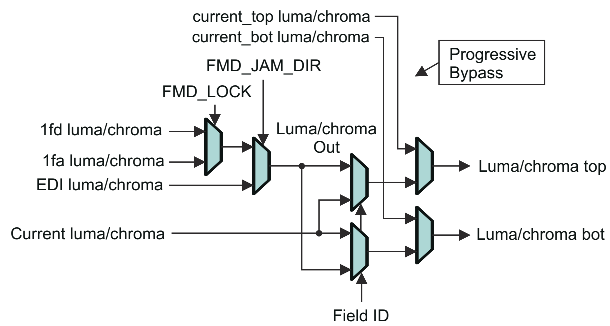


Figure 10-16. Film Mode Detection (FMD) Combing Artifacts Data Path

When in Film Mode (VPE_DEI_REG10[1] FMD_LOCK = '1'), the output of the DEI depends only on the input field sequences. Depending on the Jam Direction , either the previous or next field data is output. If not in Film Mode, the interpolated result from the EDI is output.

Because the DEI produces the missing line in an interlaced frame, what this part of the data path is selecting is the “interpolated” line. The other line output is the current line. Together, the interpolated line and the current line form the actual output of the main part of the DEI data path. Depending on if the current field is a top field or bottom field (as selected by Field ID), the output “top” of the MUX is either the current or interpolated lines, and the output “bot” is the opposite.

10.3.3.3.6 LINE BUFFER

The Line Buffer takes two lines of data (top line and bottom line) and serializes this into a single stream. The following diagram shows this data path:

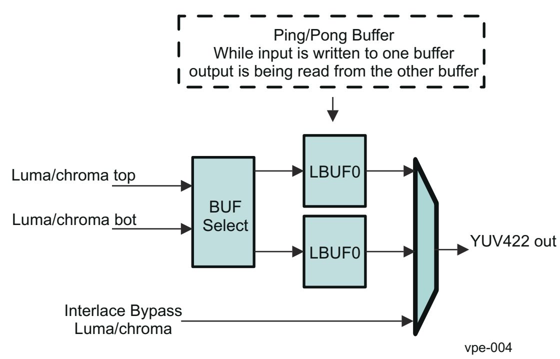


Figure 10-17. Output Data Path

The sequencing is such that LBUF0 is being written while LBUF1 is being read. When LBUF0 has been completely written, it will try to write to LBUF1 next. If on the output side LBUF1 has not been completely read, then the data path will stall waiting for the output to finish reading LBUF1 and switch to reading LBUF0 (and visa versa)

10.3.4 VPE Scaler (SC)

This section is used for driver development for the highly optimized video resizers, SC (scalers), in the VPE module.

10.3.4.1 SC Features

- Independent vertical and horizontal up and down scaling
- Running average vertical down scaling for memory optimization
- Decimation and polyphase filtering for horizontal scaling
- Non-linear scaling for stretched/compressed left and right sides
- Input image trimmer for pan/scan support
- Pre-scaling peaking filter for enhanced sharpness
- Scale field as frame
- Interlacing of scaled output
- Full 1080p input and output support
- YCbCr422 input and output
- Maximum horizontal scaling ratio only limited by output line buffer (2047 pixels)
- Scaling filter Coefficient memory download via VPI (Video Port Interface) Control interface

10.3.4.2 SC Functional Description

Scaler takes in a 10-bit YCbCr 422 video frame from an upstream module, performs vertical/horizontal scaling and outputs a YCbCr422 scaled image to a next downstream module. All configurations are done via the MMR interface except for the scaler coefficient memory configuration that is done via the common VPI control interface bus by VPDMA. Figure 10-18 shows the high-level block diagram of the scaler module.

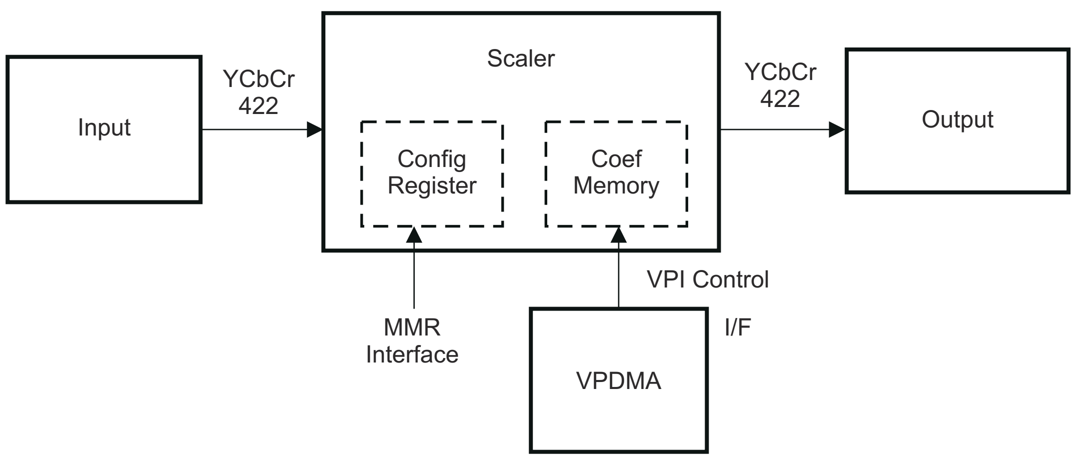


Figure 10-18. High Level Block Diagram

The SC is used in the video path and in all other video write-back data paths in the VPE module.

Scaling is performed in following three steps:

1. Trimming and Pre-peaking filtering
2. Vertical Scaling (Polyphase/Running Average Filter)
3. Horizontal polyphase scaling

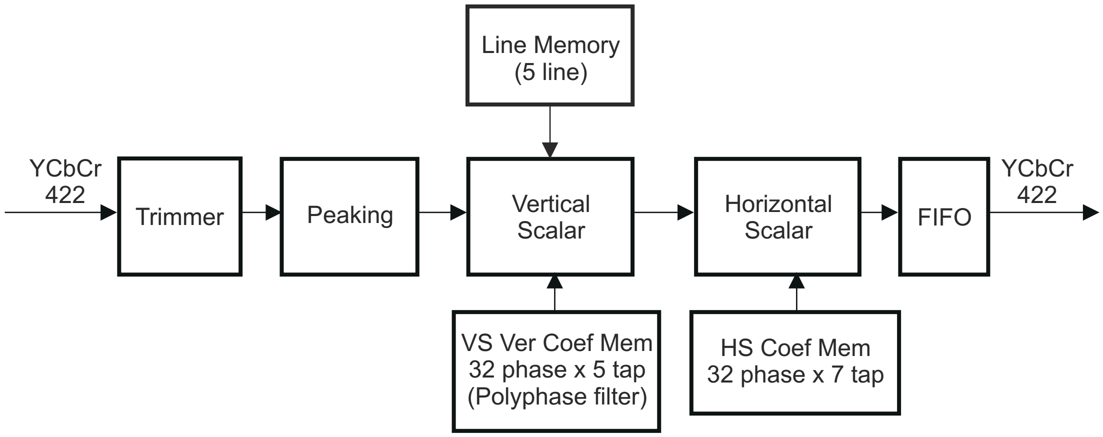


Figure 10-19. SC Block Diagram

10.3.4.2.1 Trimmer

The trimmer can be programmed to re-define a new source image within the input video frame sent from an upstream module before it is sent to the scaling sub-modules. This feature enables small area zoom out, source pan/scan, or removal of unwanted area in the video (such as black box / curtains / noisy line-21 video) without modifying the VPDMA parameters.

Horizontal and vertical offset is set through VPE_CFG_SC25[26:16] CFG_OFF_W and VPE_CFG_SC25[10:0] CFG_OFF_H registers.

Width and height are set through VPE_CFG_SC24[26:16] CFG_ORG_W and VPE_CFG_SC24[10:0] CFG_ORG_H registers.

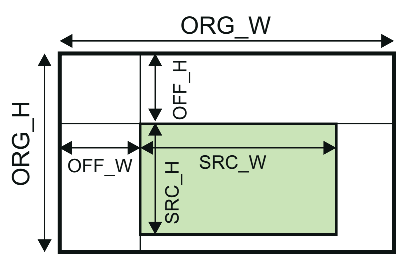


Figure 10-20. Input Image Trimming

10.3.4.2.2

Note

Width and height of the source image are global parameters and are set with VPE_CFG_SC5[22:12] CFG_SRC_W and VPE_CFG_SC5[10:0] CFG_SRC_H registers.

It is required that the input image frame (CFG_SRC_W x CFG_SRC_H) to be at least 32 × 32 (after trimming, if the trimming is enabled) to properly fill the input filter stage pipelines.

10.3.4.2.3 Peaking

The peaking block increases the amplitude of high frequency luminance information in horizontal direction to increase the sharpness of a video image before it is scaled. As shown in Figure 10-21, the high-frequency luminance is increased using an 11-tap High-Pass filter with adjustable gain. The non-linear coring function removes low-level noise and the modified luminance is then added to the original luminance signal. The implementation details are shown in Figure 10-21.

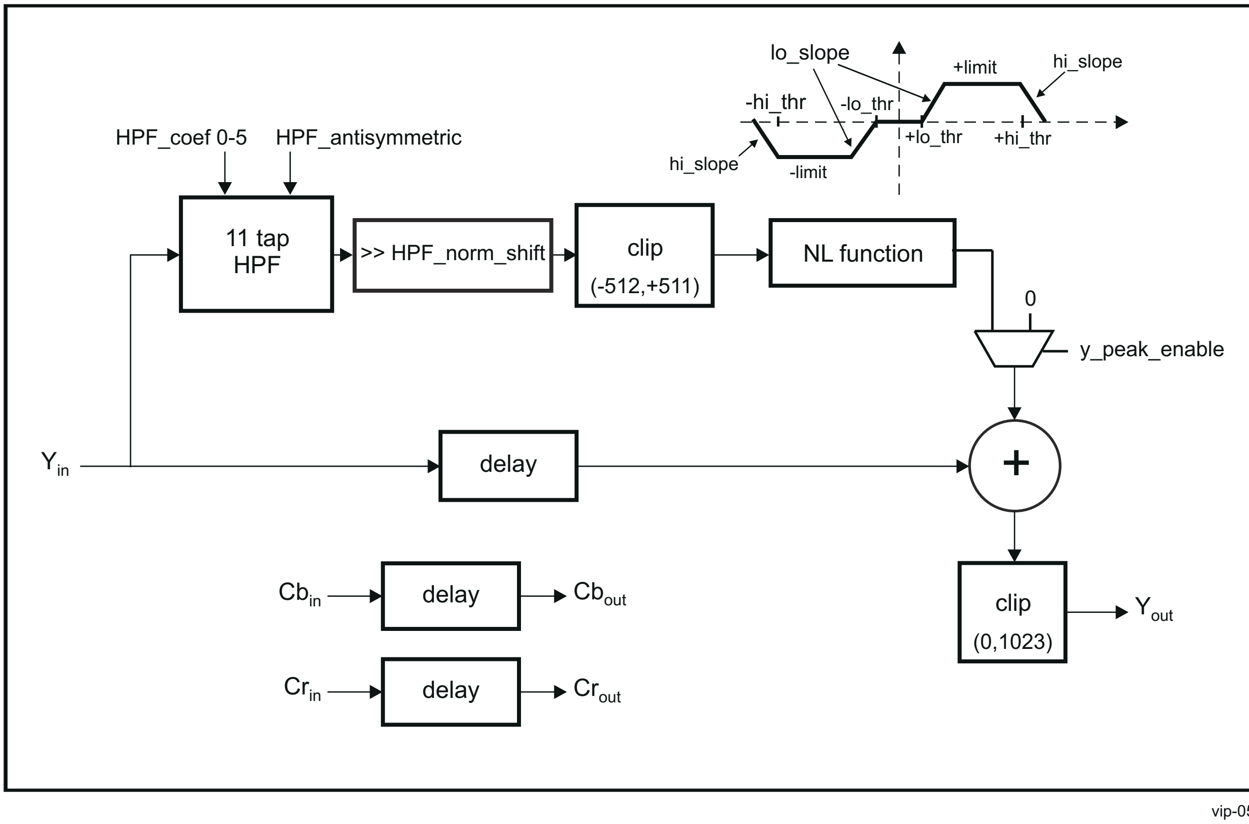


Figure 10-21. Filter Implementation and Parameter Description

Table 10-4. Parameter Description

Parameter	Description	Bits	Default
VPE_CFG_SC19[7:0] CFG_HPF_COEF0 to VPE_CFG_SC20[15:8] CFG_HPF_COEF5	FIR coefficients	8	[0 0 0-4 0 8]
VPE_CFG_SC20[18:16] CFG_HPF_NORM_SHIFT	Right shift	3	4
VPE_CFG_SC21[8:0] CFG_NL_LO_THR	Coring threshold	9	16
VPE_CFG_SC22[8:0] CFG_NL_HI_THR	High threshold	9	400
VPE_CFG_SC21[23:16] CFG_NL_LO_SLOPE	Lo slope= -CFG_NL_LO_SLOPE/16	8	16
VPE_CFG_SC22[18:16] CFG_NL_HI_SLOPE_SHIFT	Hi slope = $2^{(CFG\_NL\_HI\_SLOPE\_SHIFT-3)}$	3	4
VPE_CFG_SC20[28:20] CFG_NL_LIMIT	Clipping limit	9	200
VPE_CFG_SC0[14] CFG_Y_PK_EN	Control	1	0

Parameters for the Peaking filters are defined in VPE_CFG_SC19 through VPE_CFG_SC22 registers. The frequency responses of the peaking-filter with different sets of coefficients are shown in Figure 10-22. If the source of the input video is NTSC or PAL format, the peaking filter can be configured to reject the color subcarrier frequency.

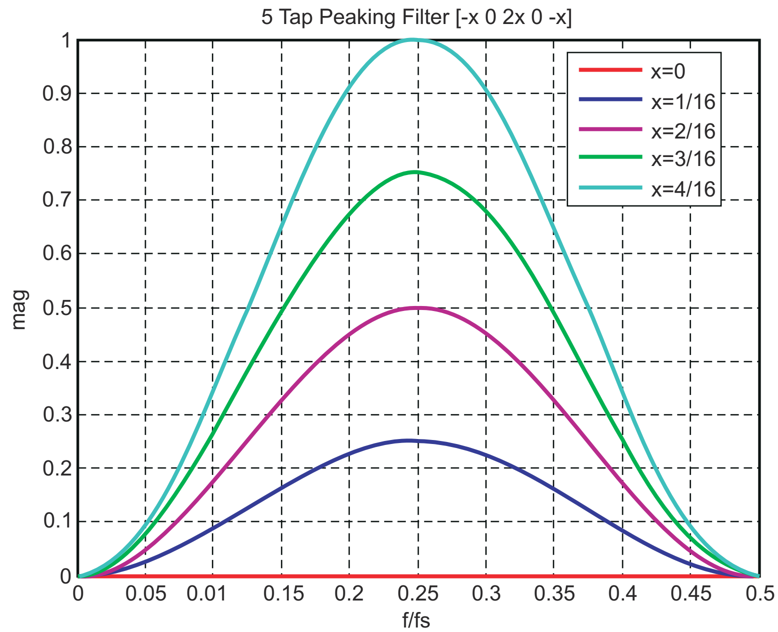


Figure 10-22. Peaking Filter at fs/4

10.3.4.2.4 Vertical Scaler

The vertical scaler has a polyphase (32-phase × 5-tap) filter and a running average filter as shown in Figure 10-23. While the polyphase filter can be used for any up-scaling and preferably downscaling to 3/16 scale factor, the running average filter is used only for downscaling to a ½ or less size. Selection between these two scalers is based on the user setting of VPE_CFG_SC0[4] CFG_USE_RAV parameter (CFG_USE_RAV= '0' for poliphase filter, and CFG_USE_RAV= '1' for running average filter), according to the user preference of the tradeoff between sharpness preserving and introduced artifacts.

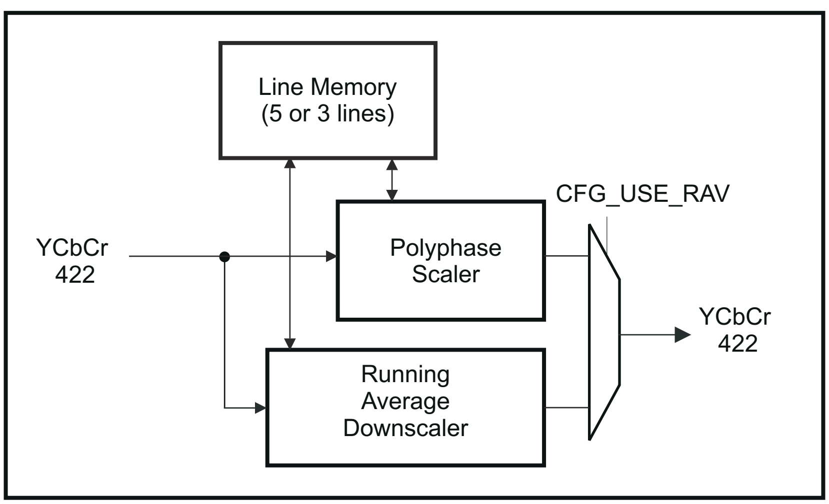


Figure 10-23. Vertical Scaler Block Diagram

10.3.4.2.4.1 Running Average Filter

When a poly-phase filter is used, usually it has to have many taps in order to achieve acceptable quality for very small downscaling ratio, which requires the use of many line buffers. In VPE, there is a weighted Running Average filter for downscaling when the scaling factor is small (for example, when the scaling ratio is less than 0.5). This highly optimized design requires only one line buffer for luma and one for chroma, which still achieving acceptable quality. The output of the running average filter is based on weighted average of pixels in the current and previous rows in vertical direction. Initializations of accumulators affect the weights.

10.3.4.2.4.2 Vertical Scaler Configuration Parameters

Table 10-5. Vertical Scaler Configuration Parameters

Parameter	Typical Value	Controls	Description
VPE_CFG_SC0[10] CFG_INTERLACE_I		Frame or Field	0 = progressive, 1 = interlace
VPE_CFG_SC0[0] CFG_INTERLACE_O			0 = progressive 1 = interlace
VPE_CFG_SC0[3] CFG_INVT_FID			Invert field ID input
VPE_CFG_SC0[4] CFG_USE_RAV		Scaler Mode	0 = use polyphase scaler 1 = use running average scaler
VPE_CFG_SC1[26:0] CFG_ROW_ACC_INC		Bilinear & Polyphase Scalers	For progressive in/progressive out: round(2^{16}*(srcH-1)/(tarH - 1)) For progressive_in/interlace_out: round(2^{16}*2*(srcH-1)/(2*tarH - 1)) For interlace_in/progressive_out: round(2^{16}*(2*srcH-1)/(2*(tarH - 1))) For interlace_in/interlace_out: round(2^{16}*(2*srcH - 1)/(2*tarH - 1)) For interlace in/out, srcH/tarH are number of field lines as specified in VPE_CFG_SC4/VPE_CFG_SC5 descriptions.
VPE_CFG_SC2[27:0] CFG_ROW_ACC_OFFSET	0		Initial row accumulator value for progressive frame and top field
VPE_CFG_SC3[27:0] CFG_ROW_ACC_OFFSET_B	0		Initial row accumulator value for bottom field
VPE_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR	4	Bilinear Scaler	Range for chroma soft switch based on pixel differences (max limit = 8)
VPE_CFG_SC13[21:12] CFG_CHROMA_INTP_THR	64		Threshold used in chroma soft switch based on pixel differences
VPE_CFG_SC13[9:0] CFG_SC_FACTOR_RAV		Running Average Scaler	Scale factor = round(1024 × tarH/srcH)
VPE_CFG_SC6[9:0] CFG_ROW_ACC_INIT_RAV			Initial row accumulator value for progressive frame and top field
VPE_CFG_SC6[19:10] CFG_ROW_ACC_INIT_RAV_B			Initial row accumulator value for bottom field

Note

Bi-linear scaler is not present in this device

10.3.4.2.5 Horizontal Scaler

The Horizontal scaler is implemented using a 32-phase × 7-tap polyphase filter preceded by two sets of 1/2x decimators. The general configuration of the Horizontal scaler is performed as follows:

- For up scaling, the input video is interpolated using the polyphase scaler.
- For downscaling, it is recommended that input video is decimated by 2 until the modified scale factor falls between 1/2 and 1. Then, a polyphase filter is configured with coefficients selected based on the mod_scale_factor calculated as shown below from one of nine different sets of coefficients: (8,9,10,11,12,13,14,15,16)/16.

```
if  (scale_factor>=1/2)  {
   mux=0;  mod_scale_factor=scale factor;
```

```
} else if  (scale_factor>=1/4) {
   mux=1; mod_scale_factor=2*scale_factor;
} else {
   mux=2; mod_scale_factor=4*scale_factor;
}
```

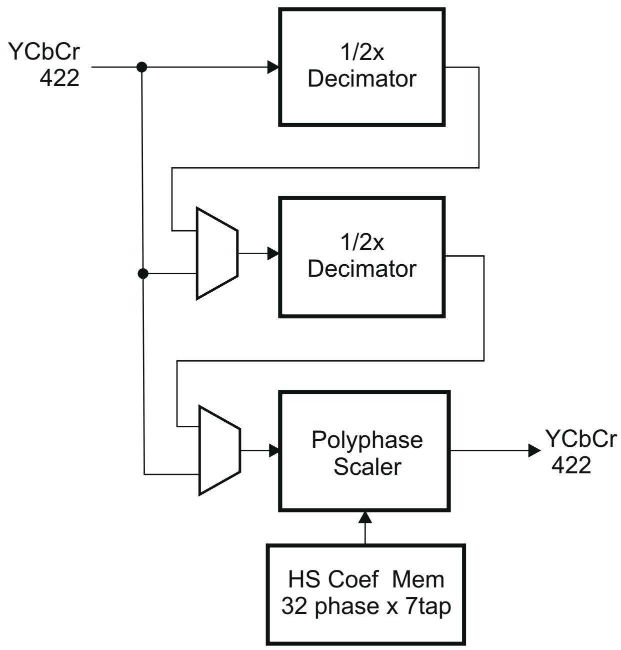


Figure 10-24. Horizontal Scaler Block Diagram

In auto mode (CFG_AUTO_HS = '1'), scaler will operate as per above recommendation. In addition to this, for (CFG_AUTO_HS = '1'), polyphase filtering will be bypassed when (scale_factor = '1') or (scale_factor = ½) or (scale_fator == ¼). If CFG_AUTO_HS = '0') is used, user must provide proper values for dcm_2x, dcm_4x, proper values for all inputs to polyphase filter in the registers as well as appropriate coefficient sets. There is no constraint on polyphase filtering in terms of scaling ratio. However, there are constraints on input and output image width for scaler. Frame dimensions are limited from 64x64 to 2047x2047.

10.3.4.2.5.1 Half Decimation Filter

The half-decimation filter is an 11-tap filter with following coefficients: (14, 0, -29, 0, 79, 128, 79, 0, -29, 0, 14). For processing left and right edge pixels, the first and last data are repeated to pre-fill and extend the filter data pipeline respectively. These coefficients are hard-coded into scaler design and user cannot modify these.

10.3.4.2.5.2 Polyphase Filter

The horizontal scaling is done by stepping across the target row and interpolating target pixels based on source pixels. Accumulator points to the source pixel that corresponds to the target pixel.

Figure 10-25 shows an up-scaling example.

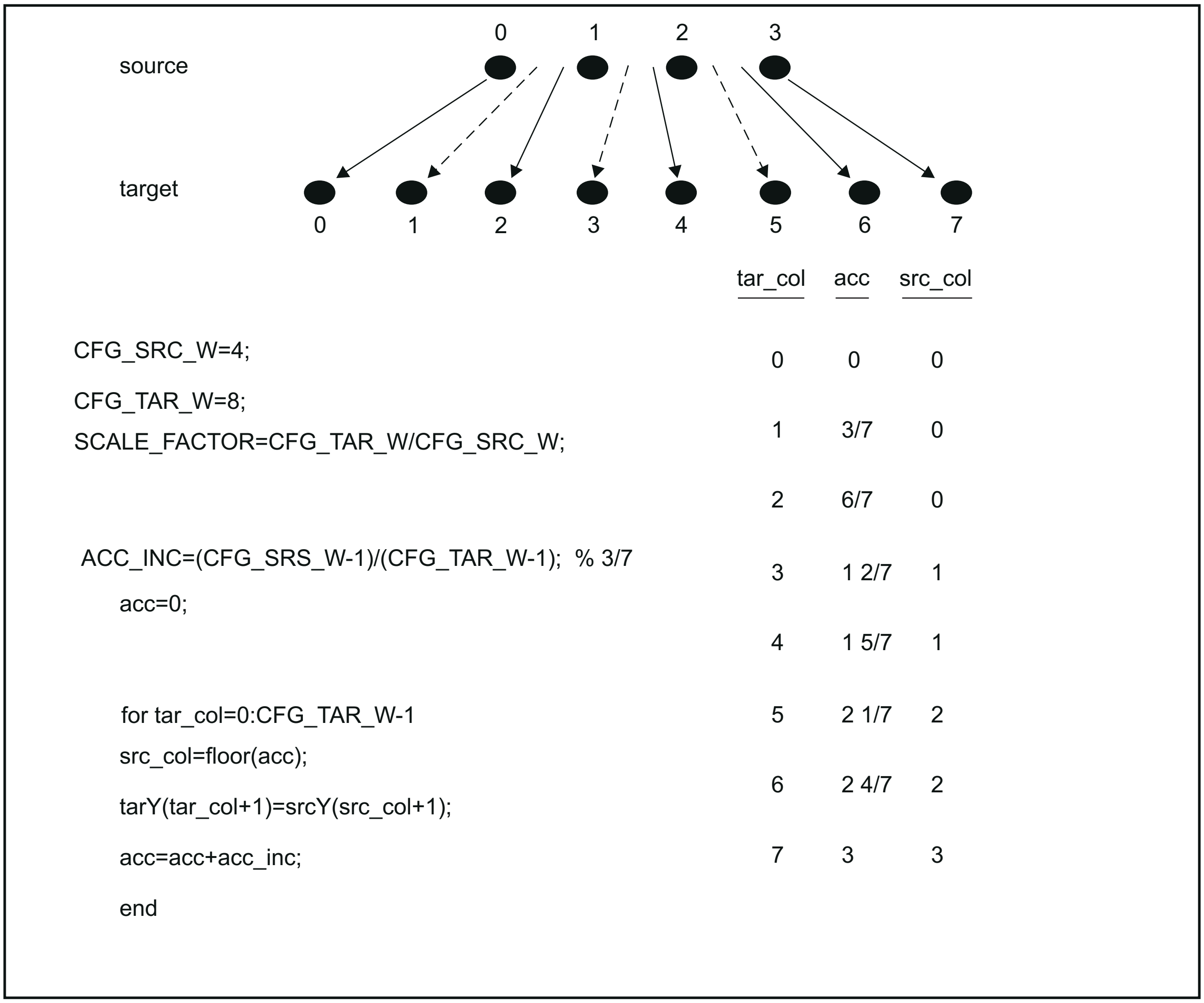


Figure 10-25. Polyphase Filtering Example

10.3.4.2.5.3 Nonlinear Horizontal Scaling

The horizontal scaler supports non-linear scaling to maintain the same aspect ratio for inner picture while stretching picture edges to fill the required resolution when capturing a 4 × 3 picture and fetching it as a 16 × 9 to memory. Non-linear scaling parameters are set with VPE_CFG_SC4[30:28] CFG_NLIN_ACC_INIT_U and VPE_CFG_SC4[26:24] CFG_LIN_ACC_INC_U registers. When setting up a non-linear scaling application, the inner picture is defined as the center square portion of the image with width and height equal to the height of the source image. Remaining side regions are then scaled non-linearly. Figure 10-26 shows a non-linear scaling case.

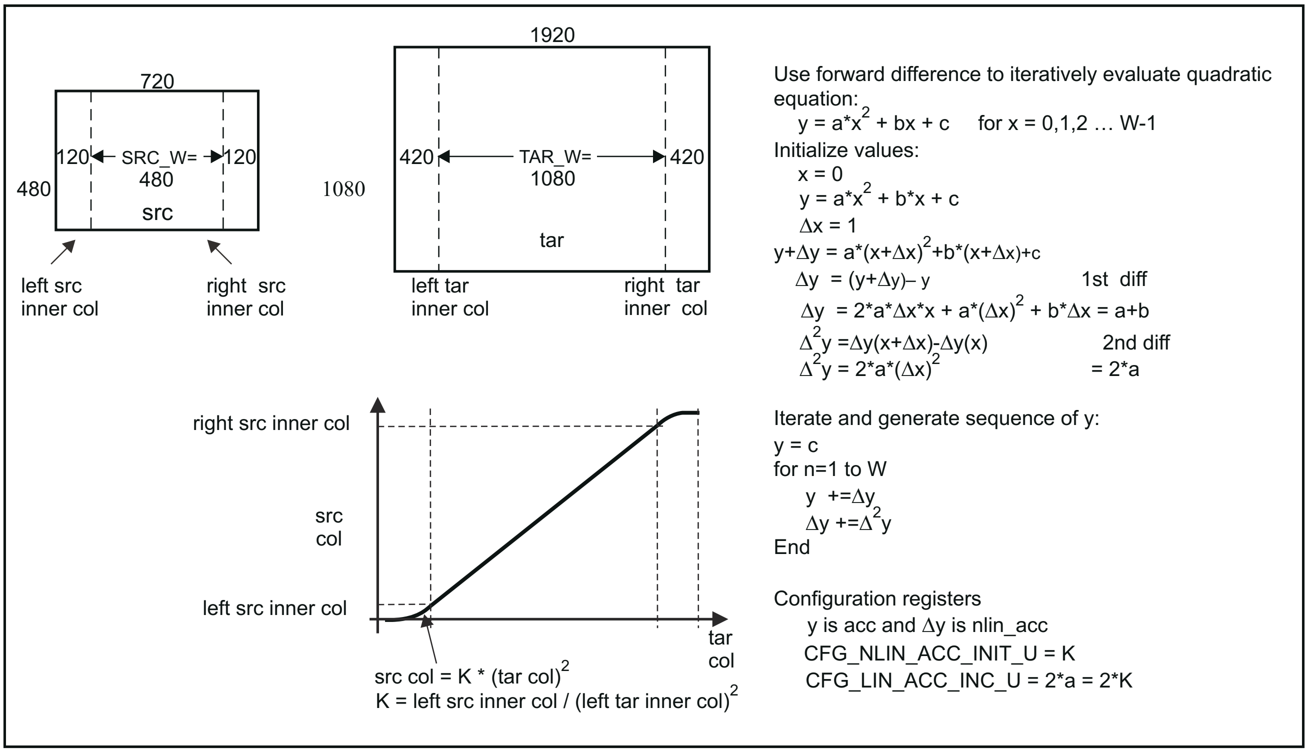


Figure 10-26. Non-linear Scaling Example

10.3.4.2.5.4 Horizontal Scaler Configuration Registers

Table 10-6. Register Group 1

<table>
<tr><th>Parameter</th><th>Controls</th><th colspan="4">Description</th></tr>
<tr><td>VPE_CFG_SC4[10:0]
CFG_TAR_H</td><td rowspan="2">Image Dimension</td><td colspan="4">Source Width</td></tr>
<tr><td>VPE_CFG_SC4[22:12]
CFG_TAR_W</td><td colspan="4">Target Width</td></tr>
<tr><td>VPE_CFG_SC0[1]
CFG_LINEAR</td><td rowspan="7">Scaler Mode</td><td colspan="4">If (linear == 1) SRC_Wi = SRC_W and TAR_Wi = TAR_W
Else SRC_W= SRC_H and TAR_W = TAR_H</td></tr>
<tr><td>VPE_CFG_SC0[2]
CFG_SC_BYPASS</td><td colspan="4">0 = enable scaler, 1 = bypass scaler</td></tr>
<tr><td rowspan="2">VPE_CFG_SC0[6]
CFG_AUTO_HS</td><td>CFG_AUTO_HS</td><td>CFG_DCM_2X</td><td>CFG_DCM_4X</td><td>Definition</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Polyphase scaling</td></tr>
<tr><td rowspan="2">VPE_CFG_SC0[7]
CFG_DCM_2X</td><td>0</td><td>0</td><td>1</td><td>Horizontal decimation by 4 and polyphase scaling</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Horizontal decimation by 2 and polyphase scaling</td></tr>
<tr><td>VPE_CFG_SC0[8]
CFG_DCM_4X</td><td>1</td><td>-</td><td>-</td><td>Automatic (selection of decimation filter is automatic)</td></tr>
</table>

Table 10-7. Register Group 2

<table>
<tr><th>Scale Factor</th><th>Decimation Usage</th><th>Control Register Bit</th></tr>
<tr><td>< 1/4</td><td>Decimation by 4</td><td rowspan="2">VPE_CFG_SC0[8] CFG_DCM_4X (set to 1 to enable decimation; disabled by default)</td></tr>
<tr><td>== 1/4</td><td>Decimation by 4</td></tr>
<tr><td>1/4 < and < 1/2</td><td>Decimation by 2</td><td rowspan="2">VPE_CFG_SC0[7] CFG_DCM_2X (set to 1 to enable decimation; disabled by default)</td></tr>
<tr><td>== 1/2</td><td>Decimation by 2</td></tr>
<tr><td>1/2 < and < 1</td><td>Bypassed</td><td rowspan="3">VPE_CFG_SC0[7] CFG_DCM_2X and CFG_DCM_4X (set to 0 to disable decimation; default value)</td></tr>
<tr><td>1</td><td>Bypassed</td></tr>
<tr><td>> 1</td><td>Bypassed</td></tr>
</table>

Table 10-8. Register Group 3

<table>
<tr><th>Parameter</th><th>Controls</th><th>Description</th></tr>
<tr><td>VPE_CFG_SC9[26:24]
CFG_LIN_ACC_INC</td><td rowspan="5">Polyphase Scaler</td><td>if upscaling then CFG_LIN_ACC_INC = round(2^{24}*(srcWi-1)/(tarWi-1))
elseif downscaling CFG_LIN_ACC_INC= round(2^{24}*(srcWi/n-1)/(tarWi-1))
where n=2 or 4</td></tr>
<tr><td>VPE_CFG_SC8[10:0]
CFG_NLIN_LEFT</td><td>if linear==1 CFG_NLIN_LEFT = 0
else CFG_NLIN_LEFT = (tarW - tarWi)/2</td></tr>
<tr><td>VPE_CFG_SC8[22:12]
CFG_NLIN_RIGHT</td><td>if linear==1 CFG_NLIN_RIGHT = tarW-1
else CFG_NLIN_RIGHT = Ltar + tarWi – 1</td></tr>
<tr><td>VPE_CFG_SC5[26:24]
CFG_NLIN_ACC_INC_U</td><td>if tarW/srcW >= 1 then
d = 0
if Ltar!=0 K = round[2^{24}*Lsrc/(Ltar*Ltar)] where Lsrc = (srcW-srcWi)/2
else K = 0
else
d = (tarW-1)/2
if Ltar!=0 K = round[2^{24} * Lsrc / (Ltar*(Ltar-2d))]
where Lsrc= (srcW-srcWi)/(2n) and n=1,2 or 4
else K = 0
CFG_LIN_ACC_INC = 2*K (negative for downscaling)</td></tr>
<tr><td>VPE_CFG_SC4[30:28]
CFG_NLIN_ACC_INIT_U</td><td>CFG_LIN_ACC_INC = K*(1-2*d)</td></tr>
</table>

Note

Source width and height variables for the polyphase filter are internally set with trimmer and decimation filter adjusted values.

10.3.4.2.6 Basic Configurations

Table 10-9 shows how the scaler should be configured based on the scale factor and the input/output mode.

Table 10-9. Scaler Configuration

Interlace		Mode[1]	VPE_CFG_SC5[10:0] CFG_SRC_H mod_srcH	VPE_CFG_SC4[10:0] CFG_TAR_H mod_tarH	Scale Factor
In	Out				
0	0	p->p	CFG_SRC_H	CFG_TAR_H	CFG_TAR_H/CFG_SRC_H
0	1	p->i	CFG_SRC_H	CFG_TAR_H/2	CFG_TAR_H/CFG_SRC_H
1	0	i->p	CFG_SRC_H/2	CFG_TAR_H	CFG_TAR_H/(CFG_SRC_H/2)
1	1	i->i	CFG_SRC_H/2	CFG_TAR_H/2	(CFG_TAR_H/2)/(CFG_SRC_H/2)

(1) p = progressive; i = interlaced

Table 10-10 shows how the vertical scaler should be configured based on the scale factor and the input/output mode.

Table 10-10. Vertical Scaler Configuration

Interlace		Mode[1]	VPE_CFG_SC9[26:24] CFG_ROW_ACC_INC/216	VPE_CFG_SC6[9:0] CFG_ROW_ACC_INIT_RAV/216	VPE_CFG_SC6[19:10] CFG_ROW_ACC_INIT_RAV_B/216
In	Out			Top	Bot
0	0	p->p	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	0
0	1	p->i	2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	(CFG_SRC_H-1)/(CFG_TAR_H-1)
1	0	i->p	1/2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	-0.5
1	1	i->i	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	[(CFG_SRC_H-1)/(CFG_TAR_H-1)-1]/2

(1) p = progressive; i = interlaced

10.3.4.2.7 Coefficient Memory

10.3.4.2.7.1 Overview

The scaler requires initialization of eight coefficient SRAMS prior to processing a video frame. These coefficients are stored in the external DDR memories and downloaded by the VPDMA. The VPDMA writes the coefficient data through the VPI Control Interface bus.

The eight coefficient SRAMs are:

- HS horizontal polyphase scaler, Luma and Chroma (7tap)
- VS vertical polyphase scaler, Luma and Chroma (5tap or 3tap)

10.3.4.2.7.2 Physical Coefficient SRAM Layout

Each of the six legacy coefficient SRAMs is 32 phases × 91 bits. A single coefficient value is 13 bits. Therefore, one word of the SRAM contains 7 coefficient values as shown in Figure 10-27, and 224 coefficient values are stored in each SRAM.

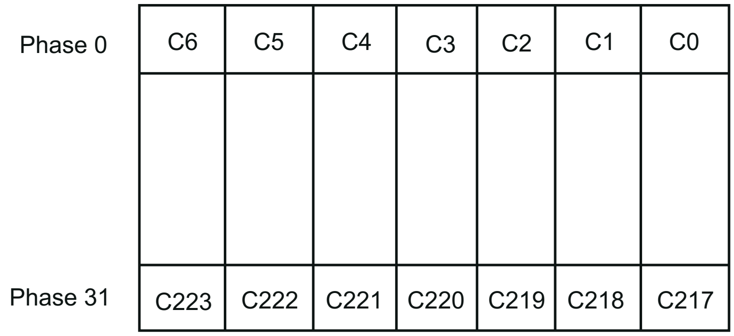


Figure 10-27. SRAM Layout for 7tap Coefficient

The two vertical polyphase SRAMs are 32 phases × 65 bits. A single coefficient is 13 bits. Therefore, one word of SRAM contains 5 coefficient values as shown in Figure 10-28.

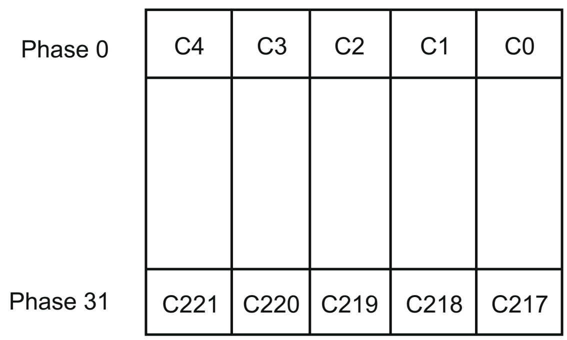


Figure 10-28. SRAM Layout for 5tap Coefficient

10.3.4.2.7.3 Scaler Coefficients Packing on 128-bit VPI Control I/F

A coefficient value is 13 bits wide and takes a single half word. Thus, one VPI Control write carries one phase's worth of coefficients. The last half-word in a quad-word is not used for 7tap coefficients.

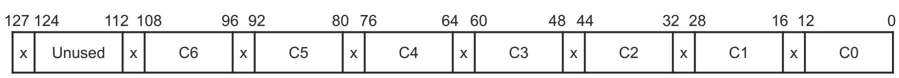


Figure 10-29. VPI Control I/F Coef Data Format (7tap)

The Vertical Polyphase scaler coefficients have only five or three values per phase, so the last three (or five) entries are unused. The Vertical Polyphase coefficient packing is shown in Figure 10-30 and Figure 10-31.

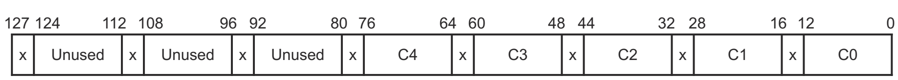


Figure 10-30. VPI Control I/F Coef Data Format (5tap)

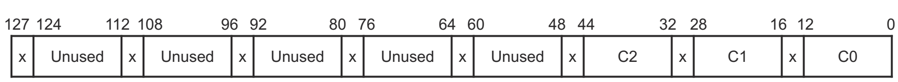


Figure 10-31. VPI Control I/F Coef Data Format (3tap)

10.3.4.2.7.4 VPI Control I/F Memory Map for Scaler Coefficients

The memory map of the VPI Control I/F for the Scaler coefficients is shown in Figure 10-32. All coefficients can be filled up by a single VPI Control write command. The update address feature of VPI Control I/F enables individual access to a certain location of memories.

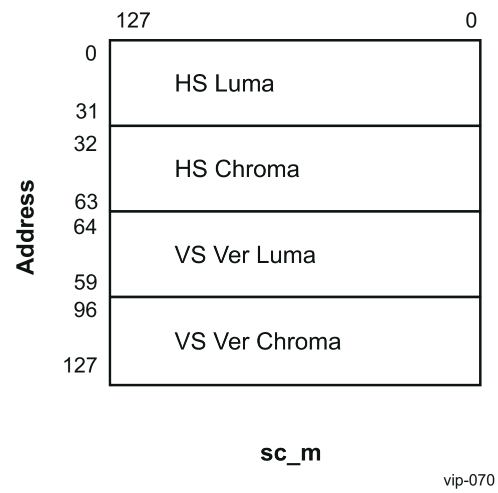


Figure 10-32. VPI Control I/F Memory Map (Write)

The module supports the VPI Control Read to read back the contents in the coefficient memories for debug purpose. Figure 10-33 shows the memory map of the VPI Control Read. As the VPI Control Read has only 32 bit data bus, it requires the word addressing while the write does the quad-word addressing.

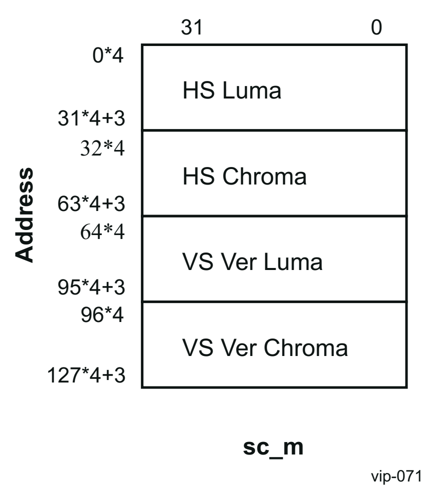


Figure 10-33. VPI Control I/F Memory Map (Read)

10.3.4.2.7.5 VPI Control Interface

VPDMA is used to configure the coefficient memories of scaler through VPI control interface. Since the coefficient memories are not shadowed (unlike the memory mapped registers), VPI control write access needs to be done only during the gap between video frame processing times. If a write request is made while the Scaler is active, the access will be held off by the hardware until the last data of the currently processed frame is sent out. Care must be given to the order of the DMA descriptors so that blocking of VPI control bus does not occur.

10.3.4.2.7.6 Coefficient Table Selection Guide

The scaler filter coefficient tables are pre-generated using a MATLAB® program for various scaling factor ranges. Table 10-11 provides a general selection guide table for coefficient data files.

The mentioned .dat files are available in Section 10.3.4.4.

Table 10-11. Coefficient Data Files

Scaler	Scale Factor	Coeff table
HS Polyphase Filter	All upscaling	Section 10.3.4.4.1.1 , *ppfcoef_scael_eq_1_32_phases_flip.dat*
	½ or ¼ down scaling	Section 10.3.4.4.1.1, ppfcoef_scale_eq_1_32_phases_flip.dat
	> 15/16	Section 10.3.4.4.1.9 , *ppfcoef_scale_eq_15div16_32_phases_flip.dat*
	> 14/16	Section 10.3.4.4.1.8 , *ppfcoef_scale_eq_14div16_32_phases_flip.dat*
	> 13/16	Section 10.3.4.4.1.7 , *ppfcoef_scale_eq_13div16_32_phases_flip.dat*
	> 12/16	Section 10.3.4.4.1.6 , *ppfcoef_scale_eq_12div16_32_phases_flip.dat*
	> 11/16	Section 10.3.4.4.1.5 , *ppfcoef_scale_eq_11div16_32_phases_flip.dat*
	> 10/16	Section 10.3.4.4.1.4 , *ppfcoef_scale_eq_10div16_32_phases_flip.dat*
	> 9/16	Section 10.3.4.4.1.3 , *ppfcoef_scale_eq_9div16_32_phases_flip.dat*
	> 8/16	Section 10.3.4.4.1.2 , *ppfcoef_scale_eq_8div16_32_phases_flip.dat* (1)
VS Polyphase Filter	Upscaling	Section 10.3.4.4.2.1 , *ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat*
	> 15/16	Section 10.3.4.4.2.6.8 , *ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat*
	> 14/16	Section 10.3.4.4.2.6.7 , *ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat*
	> 13/16	Section 10.3.4.4.2.6.6 , *ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat*
	> 12/16	Section 10.3.4.4.2.6.5 , *ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat*
	> 11/16	Section 10.3.4.4.2.6.4 , *ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat*
	> 10/16	Section 10.3.4.4.2.6.3 , *ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat*
	> 9/16	Section 10.3.4.4.2.6.2 , *ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat*
	> 8/16	Section 10.3.4.4.2.6.1 , *ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat*
	else	For down-scaling <8/16, RAV filter is recommended. In this case, coeffcients for vertical scaling need to be loaded.

(1) HS Scaler has two sets of ½ decimator to perform downscaling ratios below ½ and ¼.

10.3.4.3 SC Code

10.3.4.3.1 Generate Coefficient Memory Image

The following Perl script is used to generate a coefficient memory image for a given set of scaling factors.

```
#!/usr/local/bin/perl
#$dir="coef/";                     # directory which contains the coef files
#$cfg_file="sc_config1.cfg";       # configuration file name
#$spl_file="sc_config_supl.cfg";   # supplemental configuration file name
$cfg_file=$ARGV[0];                # configuration file name
$spl_file=$ARGV[1];                # supplemental configuration file name
$dir=$ARGV[2];                     # directory which contains the coef files
$coef_width=13;  # coef bit width
$coef_ntap=7;    # coef tap
$coef_nphase=32; # coef phase
$coef_norm=11;   # coef norm
#-----------------------------------------------------------
# read config file to get srcH/tarH/interlace_i/interlace_o
```

```
#---------------------------------------------------------
open(INFILE, "<$cfg_file") or die "### ERROR: Cannot open $cfg_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcW/) {
    $srcW = $1;
  } elsif (m/([-0-9]+) +\/\/ +srcH/) {
    $srcH = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarW/) {
    $tarW = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarH/) {
    $tarH = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_in/) {
    $interlace_i = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_out/) {
    $interlace_o = $1;
  }
}
close(INFILE);
#---------------------------------------------------------
# read supplemental config file to get srcWi/tarWi from
#---------------------------------------------------------
open(INFILE,"<$spl_file") or die "### ERROR: Cannot open $spl_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcWi/) {
    $srcWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarWi/) {
    $tarWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +profile/) {
    $profile = $1; # 0:HIGH,1:MEDIUM,2:LOW
  }
}
close(INFILE);
#---------------------------------------------------------
# determine coef file based on the width/height
#---------------------------------------------------------
#VS
#$vsc_file0 = "mod_ppfcoef_scale_eq_1_32_phases_flip.dat";
$vsc_file0  = "ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat";
#VS VER
$mod_tarH = ($interlace_i == 0 && $interlace_o == 1)   $tarH<<1 : $tarH; if ($profile==2) {
  # LOW profile
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
} else {
    if ($mod_tarH >=($srcH>>1)) {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_3tap_flip.dat",$n);
    } else {
      $n = 0;
      $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
    }
  }
} else {
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat";
  } else {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_5tap_flip.dat",$n);
  }
}
# HS
if ($tarWi >= $srcWi) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} elsif ( ($tarWi == ($srcWi>>1)) || ($tarWi == ($srcWi>>2)) ) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} else {
if ($tarWi > ($srcWi>>1)) {
    $n = int(16.0*$tarWi/$srcWi);
  } elsif ($tarWi > ($srcWi>>2)) {
    $n = int(16.0*$tarWi/($srcWi>>1));
  } elsif ($tarWi >=($srcWi>>3)) {
    $n = int(16.0*$tarWi/($srcWi>>2));
  } else {
    $n = 0;
  }
  $hsc_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_flip.dat",$n);
```

```
}
#--------------------------------------------------------------
# write out the coef hex file
#--------------------------------------------------------------
&write_coef($hsc_file0);
&write_coef($hsc_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_file0);
&write_coef($vsc_file0);
sub write_coef {
  my ($filename) = @_;
  open(INFILE, "<$dir/$filename") or die "### ERROR: Cannot open $dir/$filename";
  $line=<INFILE>;
@val=split(' ',$line);
  $ntap=$val[0];
  $nphase=$val[1];
  $norm=$val[2];
for ($p=0;$p<$nphase;$p++) {
    $line=<INFILE>;@val=split(' ',$line);
    for($i=0;$i<$ntap;$i++) {
      if ($val[$i]<0) {
        $val[$i]+=(1<<$coef_width);
      }
    }
    undef(@coef);
    unshift(@coef, sprintf("%04x",$val[0]));
    unshift(@coef, sprintf("%04x",$val[1]));
    unshift(@coef, sprintf("%04x",$val[2]));
    unshift(@coef, sprintf("%04x",$val[3]));
    unshift(@coef, sprintf("%04x",$val[4]));
    unshift(@coef, sprintf("%04x",$val[5]));
    unshift(@coef, sprintf("%04x",$val[6]));
    unshift(@coef, sprintf("%04x",0));
    $coef=join("",@coef);
    print "$coef\n";
  }
  close(INFILE);
}
```

10.3.4.3.2 Scaler Configuration Calculation

The following C-code shows how configuration parameters are calculated:

```
// =======================================
// Required Input Parameter
// =======================================
//   srcW, srcH, tarW, tarH, srcWi, tarWi
//   input/output scan modes
//   Note:  srcH and tarH refer to number of lines in the frame even for interlace in/out
//          scaling.  Based on scaling scan mode input/output scan mode option,
//          heights are adjusted during internal calculations see mod_srcH and mod_tarH.
pixel_scale_factor=4;  // 10 bit pixel
hor_pixel_offset =0.0
  // ==========================================================================================
  // Peaking Filter Configuration
  // ==========================================================================================
  // ----------------------------------------------------------------
  // HPF Coef
  // ----------------------------------------------------------------
    y_peak_enable     =   0;
    peak_select=0; // 0=peak at fs/4  1=NTSC  2=PAL
    switch(peak_select) {
    case 0: {// peak at fs/4 and gain = 1
             HPF_coef0         =   0;
             HPF_coef1         =   0;
             HPF_coef2         =   0;
             HPF_coef3         =  -4;
             HPF_coef4         =   0;
             HPF_coef5         =   8; // mid tap
             HPF_norm_shift    =   4;
         break;
}
    case 1: {// NTSC: peak at 0.133*fs and gain=1
```

```
            HPF_coef0         =  -2;
            HPF_coef1         =  -8;
            HPF_coef2         =  -8;
            HPF_coef3         =  -2;
            HPF_coef4         =  12;
            HPF_coef5         =  16; // mid tap
            HPF_norm_shift    =   6;
            break;
     }
     case 2: {// PAL: peak at 0.163*fs and gain=1
            HPF_coef0         =   2;
            HPF_coef1         =  -4;
            HPF_coef2         = -11;
            HPF_coef3         =  -7;
            HPF_coef4         =   9;
            HPF_coef5         =  22; // mid tap
            HPF_norm_shift    =   6;
            break;
     }
     }
   // ----------------------------------------------------------------
   // NonLinear Coring Function typical values
   // ----------------------------------------------------------------
     NL_coring_thr     =  16;
     NL_limit          = 200;
     NL_lo_slope       =  16;
     NL_hi_thr         = 400;
     NL_hi_slope_shift =   4;
   // =============================================================================================
   // Edge Detection Configuration
   // =============================================================================================
     // edge detection
     confidence_default = 0; // 0  =use 5 tap polyphase filter for SC with ev_enable =0

     min_Gy_thr          =  64; // 64
     min_Gy_thr_range    =   3; // 3 power of 2
     gradient_thr        = 200; // 200
gradient_thr_range =   6; // 6 power of 2

     ev_thr = int(4.0*3.111+0.5); // edge vector soft switch threshold (3.2)
   // =============================================================================================
   // vertical scaler configuration
   // =============================================================================================
   // ----------------------------------------------------------------
   // vertical scaler typical parameters
   // ----------------------------------------------------------------
     invert_field_ID           =  0; // invert field ID input
     delta_ev_thr              =  1; // edge vector soft switch range
     ver_pixel_offset          =  0.0;
     uv_intp_thr               =  pixel_scale_factor*16;
     delta_y_thr               =  4; // luma soft switch range
     delta_uv_thr              =  4; // chroma soft switch range
   //
   //
   // ----------------------------------------------------------------
   // Vertical Scaler Mode Determination
   // ----------------------------------------------------------------
   //
   //  interlace
   //   in    out   mode mod_srcH mod_tarH         scale
   // ----- ------ ---- -------- --------   -----------------
   //   0      0   p->p   srcH     tarH            tarH/srcH
   //   0      1   p->i   srcH     tarH>>1         tarH/srcH
   //   1      0   i->p   srcH>>1  tarH            tarH/(srcH/2)
   //   1      1   i->i   srcH>>1  tarH>>1    (tarH/2)/(srcH/2)
   if (interlace_in)  mod_srcH=srcH>>1; // interlace
   else               mod_srcH=srcH;    // progressive
   if (interlace_out) mod_tarH=tarH>>1; // interlace
   else               mod_tarH=tarH;    // progressive
   // determine vertical scaler
   if ((interlace_in==0)&&(interlace_out==1)) {
     if (tarH>((1+srcH)>>1)) use_rav = 0;   // 1=use RAV scaler 0=use polyphase scaler
     else                    use_rav = 1;
   } else {
     if (mod_tarH>((1+mod_srcH)>>1)) use_rav = 0; // 1=use RAV scaler  0=use polyphase scaler
     else                            use_rav = 1;
```

```
}
  // ----------------------------------------------------------------
  // RAV or Polyphase parameters
  // ----------------------------------------------------------------
  if (use_rav) {  // downscale
    // ------------
    // --- RAV ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   0;
    if ((interlace_in==0)&&(interlace_out==1)) scale = double(tarH)/double(srcH);
    else                                       scale = double(mod_tarH)/double(mod_srcH);
    sc_factor_rav = int(1024.0*scale+0.5);
// Peter's method
    delta     = (1.0/scale-1.0)/2.0;
    int_part  = floor(delta);
    frac_part = delta-int_part;

    row_acc_init_rav   = int(1024*(scale+(1.0-scale)/
2.0)+0.5);                                            // top field
    row_acc_init_b_rav = int(1024*(scale+(1.0-2.0*frac_part)*(1.0-(1.0+2.0*int_part)*scale)/
2.0)+0.5); // bottom field
    row_acc_inc        = 0; // polyphase scaler
    row_acc_offset     = 0; // polyphase scaler
    row_acc_offset_b   = 0; // polyphase scaler
  } else { // upscale using polyphase scaler
    // ------------
    // --- PPF ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   1;
    sc_factor_rav      = 0;
    delta_rav          = 0;
    row_acc_init_rav   = 0;
    row_acc_init_b_rav = 0;
    // upscaler
    // interlace                                  row acc init value
        //  in   out mode      row acc inc      top          bottom
    // ---- ---- ---- --------------------- ---  -----------------------
    //   0    0  p->p     (srcH-1)/(tarH-1)  0             0
    //   0    1  p->i   2*(srcH-1)/(tarH-1)  0    (srcH-1)/(tarH-1)
    //   1    0  i->p 1/2*(srcH-1)/(tarH-1)  0            -0.5
    //   1    1  i->i     (srcH-1)/(tarH-1)  0   [(srcH-1)/(tarH-1)-1]/2
    row_acc_offset    = int(65536.0*ver_pixel_offset +0.5);  // progressive or top field
    if (interlace_in) {
if (interlace_out) {
          row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  (int(65536.0/2.0*(double(srcH-1)/(double(tarH-1))-
                              1.0)+0.5))+row_acc_offset;
      } else { // progressive out
          row_acc_inc      = int(65536.0*double(srcH-1)/(2.0*double(tarH-1))+0.5);
            if ((-0.5+row_acc_offset)<0.0) round_factor=-0.5;
            else                           round_factor= 0.5;
            row_acc_offset_b = int(65536.0*(-0.5)+round_factor)+row_acc_offset;
      }
    } else {   // progressive in
      if (interlace_out) {
        row_acc_inc      = int(65536.0*2.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5)+row_acc_offset;
      } else { // progressive out
          row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
           row_acc_offset_b = row_acc_offset;
      }
    }
  }
// ===============================================================================================
  // Horizontal Scaler configuration
  // ===============================================================================================
  // ----------------------------------------------------------------
  // horizontal scaler mode determination
  // ----------------------------------------------------------------
  auto_hs                 = 1;
  dcm_2x                  = 0;
  dcm_4x                  = 0;
  hp_bypass               = 0;
  if (srcWi==srcW) linear = 1;
  else             linear = 0;
  // hor scaler parameters
```

```
  if (tarW>srcW) {               // upscale
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
} else if (tarW<=(srcW>>2)) {  // downscale by <=1/4
    mod_srcW  = srcW>>2;
    mod_srcWi = srcWi>>2;
  } else if (tarW<=(srcW>>1)) {  // downscale by <=1/2
    mod_srcW  = srcW>>1;
    mod_srcWi = srcWi>>1;
  } else {                       // downscale by <=1
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
  }
  // Not used any more:
  // hs_factor       = int(16.0*double(tarWi)/double(mod_srcWi)+0.5);  // hor scale factor (6.4)
  // -------------------------------------------------------------
  // Horizontal PolyPhase Settings  --
  // -------------------------------------------------------------
  lin_acc_inc     = int(16777216.0*double(mod_srcWi-1)/double(tarWi-1)+0.5);
  col_acc_offset  = int(16777216.0*hor_pixel_offset +0.5);
  nlin_left       = (tarW-tarWi)>>1;
  nlin_right      = nlin_left+tarWi-1;
  if (linear) {
    nlin_acc_inc    = 0;
    nlin_acc_init   = 0;
  } else {
     // ---------------------------------
     // Non-linear scaling configuration
     // ---------------------------------
    nlin_left_src   = (mod_srcW-mod_srcWi)>>1;

        if (tarWi>=srcWi) {    // upscale
      d              =  0.0;
      round_factor   =  0.5;
        } else {                 // downscale
      d              = (double(tarW)-1.0)/2.0;
      round_factor   = -0.5;
    }

      K              = 16777216.0*double(nlin_left_src)/(double(nlin_left)*double(nlin_left-2.0*d));
    nlin_acc_inc  = int(2.0*K+round_factor);
    nlin_acc_init = int(K*(1.0-2.0*d)+0.5);
  }
nlin_left_tar   = nlin_left;
  nlin_right_tar = nlin_right;
  // ==================================================================================
  // Bypass Determination
  // ==================================================================================
  // bypass
  if ((srcW==tarW)&&(srcWi==tarWi)&&(mod_srcH==mod_tarH)) sc_bypass = 1;
  else                                                      sc_bypass = 0;
  //
}
```

10.3.4.3.3 Typical Configuration Values

The following is the list of all scaler register fields that are set to constant values, representing typical settings:

VPE_CFG_SC0[3] CFG_INVT_FID = 0 (Field ID will be used without inversion)

VPE_CFG_SC0[5] CFG_ENABLE_EV = 1 (Field ID will be used without inversion)

VPE_CFG_SC0[6] CFG_AUTO_HS = 1 (The hardware will automatically decide, if current operation is up or down scaling. In down-scaling, it will also decide, if 2X or 4X decimation filter is needed)

VPE_CFG_SC0[7] CFG_DCM_2X = 0 (The 2X decimation filter is disabled)

VPE_CFG_SC0[8] CFG_DCM_4X = 0 (The 4X decimation filter is disabled)

VPE_CFG_SC0[11] CFG_ENABLE_SIN2_VER_INTP = 1 (Modified bilinear interpolation is used)

VPE_CFG_SC0[14] CFG_Y_PK_EN = 0 (Luma peaking is disabled)

VPE_CFG_SC0[15] CFG_TRIM= 1 (Trimming is enabled)

VPE_CFG_SC12[24:0] CFG_COL_ACC_OFFSET = 0 (No horizontal offset is involved)

VPE_CFG_SC13[21:12] CFG_CHROMA_INTP_THR = 64 (If the difference is less than this threshold, the interpolation of chroma should be done along edge direction. Otherwise, the interpolation of chroma should be done vertically)

VPE_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR = 4 (max limit=8)

VPE_CFG_SC18[24:16] CFG_CONF_DEFAULT = 0x100 (Defines confidence factor when edge detection is disabled (VPE_CFG_SC0[5] CFG_ENABLE_EV bit = 0))

VPE_CFG_SC19 = 0xFC000000

VPE_CFG_SC20 = 0x0C840800

VPE_CFG_SC21 = 0x00100010

VPE_CFG_SC22 = 0x00040190

10.3.4.4 SC Coefficient Data Files

10.3.4.4.1 HS Polyphase Filter Coefficients

10.3.4.4.1.1 ppfcoef_scale_eq_1_32_phases_flip.dat

```
  7  32  11
 31 -112  210 1790  210 -112   31
 28  -98  159 1787  264 -126   34
 25  -84  111 1779  320 -140   37
 22  -71   65 1767  379 -154   40
 19  -58   23 1750  439 -168   43
 16  -45  -17 1728  502 -181   45
 14  -33  -53 1701  565 -193   47
 11  -22  -86 1670  631 -205   49
  9  -11 -116 1635  696 -216   51
  7   -1 -142 1594  763 -225   52
  5    8 -166 1551  830 -233   53
  3   16 -186 1504  898 -240   53
  2   23 -204 1455  965 -245   52
  1   30 -218 1401 1031 -248   51
  0   35 -230 1345 1097 -249   50
 -1   40 -238 1286 1162 -248   47
 44 -244 1224 1224 -244   44    0
 47 -248 1162 1286 -238   40   -1
 50 -249 1097 1345 -230   35    0
 51 -248 1031 1401 -218   30    1
 52 -245  965 1455 -204   23    2
 53 -240  898 1504 -186   16    3
 53 -233  830 1551 -166    8    5
 52 -225  763 1594 -142   -1    7
 51 -216  696 1635 -116  -11    9
 49 -205  631 1670  -86  -22   11
 47 -193  565 1701  -53  -33   14
 45 -181  502 1728  -17  -45   16
 43 -168  439 1750   23  -58   19
 40 -154  379 1767   65  -71   22
 37 -140  320 1779  111  -84   25
 34 -126  264 1787  159  -98   28
```

10.3.4.4.1.2 ppfcoef_scale_eq_8div16_32_phases_flip.dat

```
7  32  11
    -28   61  542  898  542   61  -28
    -27   52  523  899  560   70  -29
    -26   44  505  898  578   79  -30
    -25   37  487  895  595   89  -30
    -24   30  468  892  613  100  -31
    -22   23  450  887  630  111  -31
    -21   17  432  883  647  122  -32
    -20   11  414  877  664  134  -32
    -19    6  396  871  680  146  -32
    -18    1  378  864  695  159  -31
```

```
   -16   -4  360  856  711  172  -31
   -15   -8  343  847  726  185  -30
   -14  -12  325  838  740  200  -29
   -13  -15  308  828  754  214  -28
   -12  -18  292  816  768  229  -27
   -10  -21  275  805  780  244  -25
   -23  258  789  789  258  -23    0
   -25  244  780  805  275  -21  -10
   -27  229  768  816  292  -18  -12
   -28  214  754  828  308  -15  -13
   -29  200  740  838  325  -12  -14
   -30  185  726  847  343   -8  -15
   -31  172  711  856  360   -4  -16
   -31  159  695  864  378    1  -18
   -32  146  680  871  396    6  -19
   -32  134  664  877  414   11  -20
   -32  122  647  883  432   17  -21
   -31  111  630  887  450   23  -22
   -31  100  613  892  468   30  -24
   -30   89  595  895  487   37  -25
   -30   79  578  898  505   44  -26
   -29   70  560  899  523   52  -27
```

10.3.4.4.1.3 ppfcoef_scale_eq_9div16_32_phases_flip.dat

```
7  32  11
   -33    8  547 1004  547    8  -33
   -31    0  525 1003  570   16  -35
   -29   -7  503 1001  592   25  -37
   -27  -13  481  998  614   34  -39
   -26  -19  459  995  636   44  -41
   -24  -25  437  990  658   55  -43
   -22  -29  414  983  679   67  -44
   -20  -34  393  976  700   79  -46
   -18  -38  371  968  721   91  -47
   -17  -41  350  959  742  104  -49
   -15  -44  330  948  761  118  -50
   -13  -46  309  936  780  133  -51
   -12  -48  289  924  799  148  -52
   -11  -50  270  911  817  163  -52
    -9  -51  250  897  833  180  -52
    -8  -52  232  882  850  196  -52
   -52  213  863  863  213  -52    0
   -52  196  850  882  232  -52   -8
   -52  180  833  897  250  -51   -9
   -52  163  817  911  270  -50  -11
   -52  148  799  924  289  -48  -12
   -51  133  780  936  309  -46  -13
   -50  118  761  948  330  -44  -15
   -49  104  742  959  350  -41  -17
   -47   91  721  968  371  -38  -18
   -46   79  700  976  393  -34  -20
   -44   67  679  983  414  -29  -22
   -43   55  658  990  437  -25  -24
   -41   44  636  995  459  -19  -26
   -39   34  614  998  481  -13  -27
   -37   25  592 1001  503   -7  -29
   -35   16  570 1003  525    0  -31
```

10.3.4.4.1.4 ppfcoef_scale_eq_10div16_32_phases_flip.dat

```
7  32  11
   -30  -46  542 1116  542  -46  -30
   -28  -52  515 1115  570  -39  -33
   -25  -57  488 1113  597  -32  -36
   -23  -62  462 1109  624  -24  -38
   -20  -65  435 1104  650  -15  -41
   -18  -69  409 1097  678   -5  -44
   -16  -71  383 1089  704    6  -47
   -14  -74  358 1081  730   17  -50
   -12  -75  333 1070  756   29  -53
   -11  -76  309 1058  782   42  -56
    -9  -77  285 1045  806   56  -58
```

```
     -8  -77  262 1030  831   71  -61
     -6  -77  239 1015  855   86  -64
     -5  -76  218  997  877  103  -66
     -4  -75  196  980  899  120  -68
     -3  -74  176  961  920  138  -70
    -72  156  940  940  156  -72    0
    -70  138  920  961  176  -74   -3
    -68  120  899  980  196  -75   -4
    -66  103  877  997  218  -76   -5
    -64   86  855 1015  239  -77   -6
    -61   71  831 1030  262  -77   -8
    -58   56  806 1045  285  -77   -9
    -56   42  782 1058  309  -76  -11
    -53   29  756 1070  333  -75  -12
    -50   17  730 1081  358  -74  -14
    -47    6  704 1089  383  -71  -16
    -44   -5  678 1097  409  -69  -18
    -41  -15  650 1104  435  -65  -20
    -38  -24  624 1109  462  -62  -23
    -36  -32  597 1113  488  -57  -25
    -33  -39  570 1115  515  -52  -28
```

10.3.4.4.1.5 ppfcoef_scale_eq_11div16_32_phases_flip.dat

```
7  32  11
    -19  -94  522 1230  522  -94  -19
    -17  -98  490 1230  555  -90  -22
    -14 -100  458 1227  587  -85  -25
    -12 -102  427 1223  620  -79  -29
    -10 -103  397 1217  652  -73  -32
     -8 -104  367 1209  685  -65  -36
     -6 -104  337 1199  717  -56  -39
     -4 -103  309 1187  749  -47  -43
     -3 -102  281 1174  781  -36  -47
     -1 -100  253 1159  812  -24  -51
      0  -98  227 1142  843  -11  -55
      1  -96  201 1124  874    3  -59
      1  -93  177 1105  903   18  -63
      2  -90  153 1084  932   34  -67
      2  -87  131 1062  961   51  -72
      3  -83  109 1038  987   69  -75
    -79   89 1014 1014   89  -79    0
    -75   69  987 1038  109  -83    3
    -72   51  961 1062  131  -87    2
    -67   34  932 1084  153  -90    2
    -63   18  903 1105  177  -93    1
    -59    3  874 1124  201  -96    1
    -55  -11  843 1142  227  -98    0
    -51  -24  812 1159  253 -100   -1
    -47  -36  781 1174  281 -102   -3
    -43  -47  749 1187  309 -103   -4
    -39  -56  717 1199  337 -104   -6
    -36  -65  685 1209  367 -104   -8
    -32  -73  652 1217  397 -103  -10
    -29  -79  620 1223  427 -102  -12
    -25  -85  587 1227  458 -100  -14
    -22  -90  555 1230  490  -98  -17
```

10.3.4.4.1.6 ppfcoef_scale_eq_12div16_32_phases_flip.dat

```
7  32  11
     -3 -132  486 1346  486 -132   -3
     -1 -132  449 1345  524 -131   -6
      1 -131  413 1342  562 -130   -9
      3 -130  378 1336  600 -127  -12
      4 -128  343 1328  639 -123  -15
      5 -125  309 1319  677 -119  -18
      6 -122  277 1306  716 -113  -22
      7 -118  245 1292  754 -106  -26
      8 -114  214 1276  793  -98  -31
      8 -109  185 1257  831  -89  -35
      9 -105  156 1237  869  -78  -40
      9 -100  130 1214  906  -66  -45
```

```
     9  -94  104 1190  942  -53  -50
     9  -89   79 1165  978  -38  -56
     8  -83   56 1138 1012  -22  -61
     8  -78   35 1108 1046   -4  -67
   -72   15 1081 1081   15  -72    0
   -67   -4 1046 1108   35  -78    8
   -61  -22 1012 1138   56  -83    8
   -56  -38  978 1165   79  -89    9
   -50  -53  942 1190  104  -94    9
   -45  -66  906 1214  130 -100    9
   -40  -78  869 1237  156 -105    9
   -35  -89  831 1257  185 -109    8
   -31  -98  793 1276  214 -114    8
   -26 -106  754 1292  245 -118    7
   -22 -113  716 1306  277 -122    6
   -18 -119  677 1319  309 -125    5
   -15 -123  639 1328  343 -128    4
   -12 -127  600 1336  378 -130    3
    -9 -130  562 1342  413 -131    1
    -6 -131  524 1345  449 -132   -1
```

10.3.4.4.1.7 ppfcoef_scale_eq_13div16_32_phases_flip.dat

```
7   32  11
    14 -154  435 1458  435 -154   14
    15 -150  393 1458  477 -157   12
    16 -146  353 1454  521 -160   10
    16 -141  314 1447  565 -161    8
    17 -135  276 1436  609 -161    6
    17 -129  239 1425  654 -161    3
    17 -123  204 1410  699 -159    0
    16 -116  170 1393  745 -156   -4
    16 -109  137 1373  790 -151   -8
    16 -102  107 1350  835 -146  -12
    15  -94   77 1325  879 -138  -16
    14  -87   50 1298  924 -130  -21
    13  -80   24 1269  968 -119  -27
    12  -72    0 1238 1010 -107  -33
    11  -65  -22 1204 1053  -94  -39
    10  -58  -43 1169 1093  -78  -45
   -52  -62 1138 1138  -62  -52    0
   -45  -78 1093 1169  -43  -58   10
   -39  -94 1053 1204  -22  -65   11
   -33 -107 1010 1238    0  -72   12
   -27 -119  968 1269   24  -80   13
   -21 -130  924 1298   50  -87   14
   -16 -138  879 1325   77  -94   15
   -12 -146  835 1350  107 -102   16
    -8 -151  790 1373  137 -109   16
    -4 -156  745 1393  170 -116   16
     0 -159  699 1410  204 -123   17
     3 -161  654 1425  239 -129   17
     6 -161  609 1436  276 -135   17
     8 -161  565 1447  314 -141   16
    10 -160  521 1454  353 -146   16
    12 -157  477 1458  393 -150   15
```

10.3.4.4.1.8 ppfcoef_scale_eq_14div16_32_phases_flip.dat

```
7   32  11
    27 -158  370 1570  370 -158   27
    27 -150  324 1568  417 -165   27
    26 -142  281 1563  465 -172   27
    25 -133  238 1555  515 -178   26
    24 -124  198 1543  565 -183   25
    23 -115  159 1527  616 -186   24
    22 -106  122 1510  667 -189   22
    21  -97   87 1489  719 -191   20
    19  -87   54 1464  772 -191   17
    18  -78   23 1437  824 -190   14
    16  -69   -6 1407  876 -187   11
    15  -60  -32 1373  927 -182    7
    13  -52  -57 1339  979 -176    2
```

```
   12  -44   -79 1300 1030 -168    -3
   11  -36   -99 1261 1079 -159    -9
    9  -28  -117 1218 1128 -147   -15
  -21 -134  1179 1179 -134  -21     0
  -15 -147  1128 1218 -117  -28     9
   -9 -159  1079 1261  -99  -36    11
   -3 -168  1030 1300  -79  -44    12
    2 -176   979 1339  -57  -52    13
    7 -182   927 1373  -32  -60    15
   11 -187   876 1407   -6  -69    16
   14 -190   824 1437   23  -78    18
   17 -191   772 1464   54  -87    19
   20 -191   719 1489   87  -97    21
   22 -189   667 1510  122 -106    22
   24 -186   616 1527  159 -115    23
   25 -183   565 1543  198 -124    24
   26 -178   515 1555  238 -133    25
   27 -172   465 1563  281 -142    26
   27 -165   417 1568  324 -150    27
```

10.3.4.4.1.9 ppfcoef_scale_eq_15div16_32_phases_flip.dat

```
7  32 11
   33 -143   294 1680  294 -143    33
   31 -132   246 1678  345 -155    35
   30 -121   199 1671  398 -165    36
   27 -109   154 1661  452 -175    38
   25  -97   112 1647  508 -185    38
   23  -86    72 1629  564 -193    39
   21  -75    35 1607  622 -201    39
   19  -64     0 1580  681 -207    39
   17  -53   -32 1551  740 -213    38
   15  -43   -61 1518  799 -217    37
   13  -33   -88 1481  859 -219    35
   11  -24  -113 1442  919 -220    33
    9  -15  -134 1399  978 -219    30
    8   -7  -153 1354 1036 -217    27
    6    0  -170 1307 1094 -212    23
    5    7  -184 1257 1150 -205    18
   13 -196  1207 1207 -196   13     0
   18 -205  1150 1257 -184    7     5
   23 -212  1094 1307 -170    0     6
   27 -217  1036 1354 -153   -7     8
   30 -219   978 1399 -134  -15     9
   33 -220   919 1442 -113  -24    11
   35 -219   859 1481  -88  -33    13
   37 -217   799 1518  -61  -43    15
   38 -213   740 1551  -32  -53    17
   39 -207   681 1580    0  -64    19
   39 -201   622 1607   35  -75    21
   39 -193   564 1629   72  -86    23
   38 -185   508 1647  112  -97    25
   38 -175   452 1661  154 -109    27
   36 -165   398 1671  199 -121    30
   35 -155   345 1678  246 -132    31
```

10.3.4.4.2 VS Polyphase Filter Coefficients

10.3.4.4.2.1 ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat

```
5  32 11
  -47  177 1788  177  -47
  -40  133 1785  225  -55
  -33   91 1778  276  -64
  -27   53 1765  330  -73
  -21   18 1747  386  -82
  -15  -13 1722  445  -91
  -11  -41 1693  507 -100
   -7  -66 1660  570 -109
   -3  -88 1622  635 -118
    0 -107 1579  703 -127
    2 -122 1532  771 -135
    4 -135 1482  839 -142
    5 -145 1428  909 -149
```

```
    6 -153 1371  978 -154
    7 -158 1310 1047 -158
    7 -161 1247 1116 -161
 -162 1186 1186 -162    0
 -161 1116 1247 -161    7
 -158 1047 1310 -158    7
 -154  978 1371 -153    6
 -149  909 1428 -145    5
 -142  839 1482 -135    4
 -135  771 1532 -122    2
 -127  703 1579 -107    0
 -118  635 1622  -88   -3
 -109  570 1660  -66   -7
 -100  507 1693  -41  -11
  -91  445 1722  -13  -15
  -82  386 1747   18  -21
  -73  330 1765   53  -27
  -64  276 1778   91  -33
  -55  225 1785  133  -40
```

10.3.4.4.2.2 ppfcoef_scale_eq_3_32_phases_flip.dat

```
5,    32,    11,
  130,  515,  758,  515,  130,
  121,  503,  757,  528,  139,
  113,  490,  756,  541,  148,
  105,  477,  755,  553,  158,
   97,  464,  753,  566,  168,
   90,  451,  751,  578,  178,
   83,  437,  749,  590,  189,
   76,  424,  746,  602,  200,
   69,  411,  743,  614,  211,
   63,  398,  739,  626,  222,
   57,  386,  734,  637,  234,
   52,  373,  729,  648,  246,
   46,  360,  725,  659,  258,
   41,  347,  719,  670,  271,
   37,  335,  713,  680,  283,
   32,  322,  707,  690,  297,
  314,  710,  710,  314,    0,
  297,  690,  707,  322,   32,
  283,  680,  713,  335,   37,
  271,  670,  719,  347,   41,
  258,  659,  725,  360,   46,
  246,  648,  729,  373,   52,
  234,  637,  734,  386,   57,
  222,  626,  739,  398,   63,
  211,  614,  743,  411,   69,
  200,  602,  746,  424,   76,
  189,  590,  749,  437,   83,
  178,  578,  751,  451,   90,
  168,  566,  753,  464,   97,
  158,  553,  755,  477,  105,
  148,  541,  756,  490,  113,
  139,  528,  757,  503,  121};
```

10.3.4.4.2.3 ppfcoef_scale_eq_4_32_phases_flip.dat

```
5,    32,    11,
  116,  515,  786,  515,  116,
  107,  502,  785,  530,  124,
   99,  488,  784,  544,  133,
   92,  473,  783,  557,  143,
   85,  459,  781,  571,  152,
   78,  445,  778,  585,  162,
   71,  431,  775,  598,  173,
   65,  417,  772,  611,  183,
   59,  403,  767,  624,  195,
   53,  389,  763,  637,  206,
   48,  375,  758,  649,  218,
   43,  362,  752,  661,  230,
   38,  348,  747,  673,  242,
   34,  334,  740,  685,  255,
```

```
    30,  321,  733,  696,  268,
    26,  308,  726,  707,  281,
   298,  726,  726,  298,    0,
   281,  707,  726,  308,   26,
   268,  696,  733,  321,   30,
   255,  685,  740,  334,   34,
   242,  673,  747,  348,   38,
   230,  661,  752,  362,   43,
   218,  649,  758,  375,   48,
   206,  637,  763,  389,   53,
   195,  624,  767,  403,   59,
   183,  611,  772,  417,   65,
   173,  598,  775,  431,   71,
   162,  585,  778,  445,   78,
   152,  571,  781,  459,   85,
   143,  557,  783,  473,   92,
   133,  544,  784,  488,   99,
   124,  530,  785,  502,  107};
```

10.3.4.4.2.4 ppfcoef_scale_eq_5_32_phases_flip.dat

```
5,   32,   11,
    98,  515,  822,  515,   98,
    90,  500,  821,  531,  106,
    83,  484,  820,  547,  114,
    75,  469,  819,  562,  123,
    69,  453,  816,  577,  133,
    63,  438,  813,  592,  142,
    57,  422,  809,  607,  153,
    51,  407,  805,  622,  163,
    46,  391,  801,  636,  174,
    41,  376,  795,  650,  186,
    37,  361,  789,  664,  197,
    32,  347,  782,  678,  209,
    28,  332,  775,  691,  222,
    25,  317,  767,  704,  235,
    22,  303,  759,  716,  248,
    18,  289,  750,  729,  262,
   278,  746,  746,  278,    0,
   262,  729,  750,  289,   18,
   248,  716,  759,  303,   22,
   235,  704,  767,  317,   25,
   222,  691,  775,  332,   28,
   209,  678,  782,  347,   32,
   197,  664,  789,  361,   37,
   186,  650,  795,  376,   41,
   174,  636,  801,  391,   46,
   163,  622,  805,  407,   51,
   153,  607,  809,  422,   57,
   142,  592,  813,  438,   63,
   133,  577,  816,  453,   69,
   123,  562,  819,  469,   75,
   114,  547,  820,  484,   83,
   106,  531,  821,  500,   90};
```

10.3.4.4.2.5 ppfcoef_scale_eq_6_32_phases_flip.dat

```
5,   32,   11,
    77,  513,  868,  513,   77,
    70,  496,  867,  531,   84,
    63,  479,  866,  548,   92,
    57,  461,  864,  566,  100,
    51,  444,  861,  583,  109,
    46,  427,  857,  600,  118,
    41,  409,  853,  617,  128,
    36,  393,  847,  633,  139,
    32,  376,  841,  650,  149,
    28,  359,  835,  666,  160,
    24,  343,  827,  682,  172,
    21,  327,  819,  697,  184,
    18,  311,  810,  712,  197,
    15,  296,  800,  727,  210,
    13,  281,  790,  741,  223,
```

```
   11,  266,  779,  755,  237,
  253,  771,  771,  253,    0,
  237,  755,  779,  266,   11,
  223,  741,  790,  281,   13,
  210,  727,  800,  296,   15,
  197,  712,  810,  311,   18,
  184,  697,  819,  327,   21,
  172,  682,  827,  343,   24,
  160,  666,  835,  359,   28,
  149,  650,  841,  376,   32,
  139,  633,  847,  393,   36,
  128,  617,  853,  409,   41,
  118,  600,  857,  427,   46,
  109,  583,  861,  444,   51,
  100,  566,  864,  461,   57,
   92,  548,  866,  479,   63,
   84,  531,  867,  496,   70};
```

10.3.4.4.2.6 ppfcoef_scale_eq_7_32_phases_flip.dat

```
5,   32,   11,
   53,  510,  922,  510,   53,
   47,  490,  922,  529,   60,
   41,  470,  921,  549,   67,
   36,  451,  918,  569,   74,
   32,  431,  915,  588,   82,
   27,  412,  910,  608,   91,
   23,  393,  905,  627,  100,
   20,  374,  898,  646,  110,
   17,  356,  890,  665,  120,
   14,  337,  882,  684,  131,
   11,  320,  873,  702,  142,
    9,  302,  863,  720,  154,
    7,  285,  852,  737,  167,
    6,  269,  840,  753,  180,
    4,  253,  827,  770,  194,
    3,  237,  815,  785,  208,
  223,  801,  801,  223,    0,
  208,  785,  815,  237,    3,
  194,  770,  827,  253,    4,
  180,  753,  840,  269,    6,
  167,  737,  852,  285,    7,
  154,  720,  863,  302,    9,
  142,  702,  873,  320,   11,
  131,  684,  882,  337,   14,
  120,  665,  890,  356,   17,
  110,  646,  898,  374,   20,
  100,  627,  905,  393,   23,
   91,  608,  910,  412,   27,
   82,  588,  915,  431,   32,
   74,  569,  918,  451,   36,
   67,  549,  921,  470,   41,
   60,  529,  922,  490,   47};
```

10.3.4.4.2.6.1 ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat

```
5   32  11
   28   502  988  502   28
   24   479  987  524   34
   19   457  985  547   40
   15   435  982  570   46
   12   413  978  592   53
    9   392  972  614   61
    6   371  965  637   69
    4   350  957  659   78
    2   330  948  680   88
    0   310  938  702   98
   -1   291  926  723  109
   -2   272  914  744  120
   -3   254  900  764  133
   -3   237  886  783  145
   -4   220  871  802  159
   -4   204  855  820  173
```

```
    188  836  836  188    0
    173  820  855  204   -4
    159  802  871  220   -4
    145  783  886  237   -3
    133  764  900  254   -3
    120  744  914  272   -2
    109  723  926  291   -1
     98  702  938  310    0
     88  680  948  330    2
     78  659  957  350    4
     69  637  965  371    6
     61  614  972  392    9
     53  592  978  413   12
     46  570  982  435   15
     40  547  985  457   19
     34  524  987  479   24
```

10.3.4.4.2.6.2 ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat

```
5   32  11
      3  489 1064  489    3
      0  464 1062  515    7
     -3  439 1060  540   12
     -5  414 1056  566   17
     -7  390 1050  592   23
     -9  366 1044  618   29
    -10  343 1035  644   36
    -11  320 1025  670   44
    -12  298 1014  695   53
    -12  277 1001  720   62
    -12  256  987  745   72
    -12  236  972  769   83
    -12  217  956  792   95
    -11  199  938  815  107
    -10  181  920  837  120
    -10  165  900  859  134
    148  876  876  148    0
    134  859  900  165  -10
    120  837  920  181  -10
    107  815  938  199  -11
     95  792  956  217  -12
     83  769  972  236  -12
     72  745  987  256  -12
     62  720 1001  277  -12
     53  695 1014  298  -12
     44  670 1025  320  -11
     36  644 1035  343  -10
     29  618 1044  366   -9
     23  592 1050  390   -7
     17  566 1056  414   -5
     12  540 1060  439   -3
      7  515 1062  464    0
```

10.3.4.4.2.6.3 ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat

```
5   32  11
    -20  470 1148  470  -20
    -22  442 1147  499  -18
    -23  413 1144  529  -15
    -24  386 1139  558  -11
    -24  359 1132  588   -7
    -24  333 1124  618   -3
    -24  308 1113  648    3
    -23  283 1101  678    9
    -23  260 1088  707   16
    -22  237 1072  737   24
    -21  215 1056  765   33
    -19  194 1037  793   43
    -18  174 1017  822   53
    -16  156  995  848   65
    -15  138  973  875   77
    -13  121  949  900   91
    105  919  919  105    0
```

```
    91   900   949  121   -13
    77   875   973  138   -15
    65   848   995  156   -16
    53   822  1017  174   -18
    43   793  1037  194   -19
    33   765  1056  215   -21
    24   737  1072  237   -22
    16   707  1088  260   -23
     9   678  1101  283   -23
     3   648  1113  308   -24
    -3   618  1124  333   -24
    -7   588  1132  359   -24
   -11   558  1139  386   -24
   -15   529  1144  413   -23
   -18   499  1147  442   -22
```

10.3.4.4.2.6.4 ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat

```
5   32  11
   -40   444  1240  444   -40
   -40   412  1240  476   -40
   -40   381  1236  510   -39
   -39   350  1231  544   -38
   -37   321  1223  577   -36
   -36   293  1212  612   -33
   -34   265  1200  646   -29
   -32   239  1185  681   -25
   -30   214  1169  715   -20
   -28   190  1150  750   -14
   -26   167  1130  783    -6
   -23   146  1107  816     2
   -21   126  1083  849    11
   -19   107  1057  882    21
   -17    90  1030  913    32
   -15    73  1002  943    45
    58   966   966   58     0
    45   943  1002   73   -15
    32   913  1030   90   -17
    21   882  1057  107   -19
    11   849  1083  126   -21
     2   816  1107  146   -23
    -6   783  1130  167   -26
   -14   750  1150  190   -28
   -20   715  1169  214   -30
   -25   681  1185  239   -32
   -29   646  1200  265   -34
   -33   612  1212  293   -36
   -36   577  1223  321   -37
   -38   544  1231  350   -39
   -39   510  1236  381   -40
   -40   476  1240  412   -40
```

10.3.4.4.2.6.5 ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat

```
5   32  11
   -56   409  1342  409   -56
   -54   373  1342  445   -58
   -51   339  1337  482   -59
   -49   306  1330  521   -60
   -46   274  1321  559   -60
   -43   244  1308  598   -59
   -40   215  1293  638   -58
   -36   187  1275  678   -56
   -33   161  1255  718   -53
   -30   137  1233  757   -49
   -27   114  1208  797   -44
   -24    93  1182  836   -39
   -21    73  1152  875   -31
   -18    55  1122  912   -23
   -16    38  1090  950   -14
   -14    23  1056  986    -3
     9  1015  1015    9     0
    -3   986  1056   23   -14
```

```
   -14  950 1090   38  -16
   -23  912 1122   55  -18
   -31  875 1152   73  -21
   -39  836 1182   93  -24
   -44  797 1208  114  -27
   -49  757 1233  137  -30
   -53  718 1255  161  -33
   -56  678 1275  187  -36
   -58  638 1293  215  -40
   -59  598 1308  244  -43
   -60  559 1321  274  -46
   -60  521 1330  306  -49
   -59  482 1337  339  -51
   -58  445 1342  373  -54
```

10.3.4.4.2.6.6 ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat

```
5  32  11
   -65  364 1450  364  -65
   -61  326 1448  404  -69
   -57  289 1443  445  -72
   -53  253 1435  488  -75
   -48  220 1423  531  -78
   -44  188 1408  576  -80
   -40  158 1390  621  -81
   -36  130 1370  666  -82
   -32  103 1346  713  -82
   -28   79 1320  758  -81
   -24   56 1290  805  -79
   -21   36 1259  850  -76
   -18   17 1224  896  -71
   -15    0 1188  940  -65
   -12  -15 1149  984  -58
   -10  -28 1109 1027  -50
   -40 1064 1064  -40    0
   -50 1027 1109  -28  -10
   -58  984 1149  -15  -12
   -65  940 1188    0  -15
   -71  896 1224   17  -18
   -76  850 1259   36  -21
   -79  805 1290   56  -24
   -81  758 1320   79  -28
   -82  713 1346  103  -32
   -82  666 1370  130  -36
   -81  621 1390  158  -40
   -80  576 1408  188  -44
   -78  531 1423  220  -48
   -75  488 1435  253  -53
   -72  445 1443  289  -57
   -69  404 1448  326  -61
```

10.3.4.4.2.6.7 ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat

```
5  32  11
   -67  310 1562  310  -67
   -61  269 1559  353  -72
   -55  230 1553  398  -78
   -50  193 1543  445  -83
   -44  158 1529  493  -88
   -39  125 1512  543  -93
   -34   94 1491  594  -97
   -30   66 1468  645 -101
   -25   41 1439  697 -104
   -22   17 1408  751 -106
   -18   -4 1373  804 -107
   -15  -23 1336  857 -107
   -12  -40 1296  910 -106
    -9  -55 1253  962 -103
    -7  -67 1208 1013  -99
    -5  -78 1161 1064  -94
   -86 1110 1110  -86    0
   -94 1064 1161  -78   -5
   -99 1013 1208  -67   -7
```

```
  -103   962 1253  -55   -9
  -106   910 1296  -40  -12
  -107   857 1336  -23  -15
  -107   804 1373   -4  -18
  -106   751 1408   17  -22
  -104   697 1439   41  -25
  -101   645 1468   66  -30
   -97   594 1491   94  -34
   -93   543 1512  125  -39
   -88   493 1529  158  -44
   -83   445 1543  193  -50
   -78   398 1553  230  -55
   -72   353 1559  269  -61
```

10.3.4.4.2.6.8 ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat

```
5  32 11
   -61  248 1674  248  -61
   -54  204 1673  293  -68
   -47  163 1665  342  -75
   -41  125 1654  392  -82
   -35   90 1638  445  -90
   -29   57 1618  499  -97
   -24   27 1593  556 -104
   -20    0 1565  613 -110
   -16  -24 1532  672 -116
   -12  -46 1495  732 -121
    -9  -65 1455  793 -126
    -6  -81 1411  854 -130
    -4  -95 1364  915 -132
    -2 -107 1315  975 -133
     0 -116 1262 1035 -133
     1 -123 1208 1094 -132
  -128 1152 1152 -128    0
  -132 1094 1208 -123    1
  -133 1035 1262 -116    0
  -133  975 1315 -107   -2
  -132  915 1364  -95   -4
  -130  854 1411  -81   -6
  -126  793 1455  -65   -9
  -121  732 1495  -46  -12
  -116  672 1532  -24  -16
  -110  613 1565    0  -20
  -104  556 1593   27  -24
   -97  499 1618   57  -29
   -90  445 1638   90  -35
   -82  392 1654  125  -41
   -75  342 1665  163  -47
   -68  293 1673  204  -54
```

10.3.4.4.2.6.9 ppcoef_scale_1x_ver_5tap.dat

```
5 32 11
0 0 2048 0 0
-40 133 1785 225 -55
-33 91 1778 276 -64
-27 53 1765 330 -73
-21 18 1747 386 -82
-15 -13 1722 445 -91
-11 -41 1693 507 -100
-7 -66 1660 570 -109
-3 -88 1622 635 -118
0 -107 1579 703 -127
2 -122 1532 771 -135
4 -135 1482 839 -142
5 -145 1428 909 -149
6 -153 1371 978 -154
7 -158 1310 1047 -158
7 -161 1247 1116 -161
162 1186 1186 -162 0
161 1116 1247 -161 7
158 1047 1310 -158 7
154 978 1371 -153 6
```

```
149 909 1428 -145 5
142 839 1482 -135 4
135 771 1532 -122 2
127 703 1579 -107 0
118 635 1622 -88 -3
109 570 1660 -66 -7
100 507 1693 -41 -11
-91 445 1722 -13 -15
-82 386 1747 18 -21
-73 330 1765 53 -27
-64 276 1778 91 -33
-55 225 1785 133 -40
```

10.3.4.4.3 VS (Bilinear Filter Coefficients)

10.3.4.4.3.1 ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat

This is not applicable for this device

```
7  32 11
  -11 -106  190 1902  190 -106  -11
   -9 -105  153 1897  230 -105  -13
   -8 -105  121 1887  274 -105  -16
   -6 -104   91 1869  320 -104  -18
   -5 -102   65 1843  370 -102  -21
   -4 -101   42 1812  424 -101  -24
   -3  -99   20 1776  480  -99  -27
   -2  -96    3 1730  539  -96  -30
   -1  -93  -12 1679  602  -93  -34
   -1  -90  -26 1627  665  -90  -37
    0  -87  -37 1568  732  -87  -41
    0  -84  -46 1506  801  -84  -45
    0  -80  -54 1439  871  -80  -48
    0  -76  -60 1371  941  -76  -52
    1  -72  -65 1299 1013  -72  -56
    1  -68  -69 1227 1085  -68  -60
  -64  -64 1152 1152  -64  -64    0
  -60  -68 1085 1227  -69  -68    1
  -56  -72 1013 1299  -65  -72    1
  -52  -76  941 1371  -60  -76    0
  -48  -80  871 1439  -54  -80    0
  -45  -84  801 1506  -46  -84    0
  -41  -87  732 1568  -37  -87    0
  -37  -90  665 1627  -26  -90   -1
  -34  -93  602 1679  -12  -93   -1
  -30  -96  539 1730    3  -96   -2
  -27  -99  480 1776   20  -99   -3
  -24 -101  424 1812   42 -101   -4
  -21 -102  370 1843   65 -102   -5
  -18 -104  320 1869   91 -104   -6
  -16 -105  274 1887  121 -105   -8
  -13 -105  230 1897  153 -105   -9
```

10.3.5 VPE Color Space Converter (CSC)

The color space converter (CSC) module is used to convert video data from one color space to another with nine programmable integer multipliers.

10.3.5.1 CSC Features

- All parameters are programmable
- Each parameter is configurable in signed 13-bits
- Support for Bypass Mode

10.3.5.2 CSC Functional Description

The conversion between the different color spaces requires addition and multiplication operations on color and intensity components. The mathematical expression of the conversion can be written as:

```
Y   = A0*R + B0*G + C0*B + D0
Cb  = A1*R + B1*G + C1*B + D1
Cr  = A2*R + B2*G + C2*B + D2
```

Color space coeficients are set through the following registers:

- For luma component:
  - VPE_CSC00[12:0] A0
  - VPE_CSC00[28:16] B0
  - VPE_CSC01[28:16] C0
  - VPE_CSC04[27:16] D0
- For Cb component:
  - VPE_CSC01[28:16] A1
  - VPE_CSC02[12:0] B1
  - VPE_CSC02[27:16] C1
  - VPE_CSC05[11:0] D1
- For Cr component :
  - VPE_CSC03[12:0] A2
  - VPE_CSC03[27:16] B2
  - VPE_CSC04[12:0] C2
  - VPE_CSC05[27:16] D2

10.3.5.3

Using YUV to RGB conversion as an example: YUV represents one color space and RGB represents another color space. The conversion can be written in the matrix format shown in Figure 10-34.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} A0 & B0 & C0 \\ A1 & B1 & C1 \\ A2 & B2 & C2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} D0 \\ D1 \\ D2 \end{bmatrix}$$

Figure 10-34. Matrix Format

Since HDTV and SDTV have different conversion requirements, both conversions of RGB-to-YCbCr and YCbCr-to-RGB are described. The details of derivations of these matrixes will be given in the following subsections.

10.3.5.3.1 HDTV Application

10.3.5.3.1.1 HDTV Application with Video Data Range

The two equations presented in this section are for the HDTV application. The chromaticity parameters are defined by ITU-R709 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]

- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4646 & -0.0468 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 10-35. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5396 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -197 \\ 82 \\ -232 \end{bmatrix} D$$

Figure 10-36. Conversion from YCbCr to RGB

10.3.5.3.1.2 HDTV Application with Graphics Data Range

The two equations presented in this section are for the HDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R709 standard.

The input data ranges for these equations are as follows.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.1826 & 0.6142 & 0.0620 \\ -0.1006 & -0.3385 & 0.4392 \\ 0.4392 & -0.3990 & -0.0402 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 10-37. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1644 & -0.0003 & 1.7927 \\ 1.1644 & -0.2132 & -0.5329 \\ 1.1642 & 2.1125 & -0.0001 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -248 \\ 77 \\ -289 \end{bmatrix} D$$

Figure 10-38. Conversion from YCbCr to RGB

10.3.5.3.1.3 Quantized Coefficients for Color Space Converter in HDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for HDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 10-12. Quantized Coefficients of HDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VPE_CSC00[12:0] A0	0.2126	218	0x00DA	1	1024	0x0400
B0(13-bit)	VPE_CSC00[28:16] B0	0.7152	732	0x02DC	0	0	0x0000
C0(13-bit)	VPE_CSC01[28:16] C0	0.0722	74	0x004A	1.5396	1577	0x0629
A1(13-bit)	VPE_CSC01[28:16] A1	-0.1172	-120	0x1F88	1	1024	0x0400
B1(13-bit)	VPE_CSC02[12:0] B1	-0.3942	-404	0x1E6C	-0.1831	-187	0x1F45
C1(13-bit)	VPE_CSC02[27:16] C1	0.5114	524	0x020C	-0.4577	-469	0x1E2B
A2(13-bit)	VPE_CSC03[12:0] A2	0.5114	524	0x020C	1	1024	0x0400
B2(13-bit)	VPE_CSC03[27:16] B2	-0.4646	-476	0x1E24	1.8142	1858	0x0742
C2(13-bit)	VPE_CSC04[12:0] C2	-0.0468	-48	0x1FD0	0	0	0x0000
D0(12-bit)	VPE_CSC04[27:16] D0	0	0	0x000	-197	-788	0xCEC
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	82	328	0x148
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-232	-928	0xC60

Table 10-13. Quantized Coefficients of HDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VPE_CSC00[12:0] A0	0.1826	187	0x00BB	1.1644	1192	0x04A8
B0(13-bit)	VPE_CSC00[28:16] B0	0.6142	629	0x0275	-0.0003	0	0x0000
C0(13-bit)	VPE_CSC01[28:16] C0	0.062	63	0x003F	1.7927	1836	0x072C
A1(13-bit)	VPE_CSC01[28:16] A1	-0.1006	-103	0x1F99	1.1644	1192	0x04A8
B1(13-bit)	VPE_CSC02[12:0] B1	-0.3385	-347	0x1EA5	-0.2132	-218	0x1F26
C1(13-bit)	VPE_CSC02[27:16] C1	0.4392	450	0x01C2	-0.5329	-546	0x1DDE
A2(13-bit)	VPE_CSC03[12:0] A2	0.4392	450	0x01C2	1.1642	1192	0x04A8
B2(13-bit)	VPE_CSC03[27:16] B2	-0.399	-409	0x1E67	2.1125	2163	0x0873
C2(13-bit)	VPE_CSC04[12:0] C2	-0.0402	-41	0x1FD7	-0.0001	0	0x0000
D0(12-bit)	VPE_CSC04[27:16] D0	16	64	0x040	-248	-992	0xC20
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	77	308	0x134

Table 10-13. Quantized Coefficients of HDTV Application with Graphics Data Range (continued)

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	Registers	Real Number Format	Quantized Format	Hex Format	Real Number Format	Quantized Format	Hex Format
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-289	-1156	0xB7C

10.3.5.3.2 SDTV Application

10.3.5.3.2.1 SDTV Application with Video Data Range

The two equations presented in this section are for the SDTV application. The chromaticity parameters are defined by ITU-R601 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.172 & -0.339 & 0.511 \\ 0.511 & -0.428 & -0.083 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 10-39. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & -0.0003 & 1.3717 \\ 1 & -0.3365 & -0.6984 \\ 1 & 1.7336 & -0.0016 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -176 \\ 132 \\ -222 \end{bmatrix} D$$

Figure 10-40. Conversion from YCbCr to RGB

10.3.5.3.2.2 SDTV Application with Graphics Data Range

The two equations presented in this section are for the SDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R601 standard.

The input data ranges for these equations are as following.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Figure 10-41. Conversion from RGB to YCbCr

Conversion from YCbCr to RGB:

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1641 & -0.0018 & 1.5958 \\ 1.1641 & -0.3914 & -0.8135 \\ 1.1641 & 2.0178 & -0.0012 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -223 \\ 136 \\ -277 \end{bmatrix} D$$

Figure 10-42. Conversion from YCbCr to RGB

10.3.5.3.2.3 Quantized Coefficients for Color Space Converter in SDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for SDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 10-14. Quantized Coefficients of SDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VPE_CSC00[12:0] A0	0.299	306	0x0132	1	1024	0x0400
B0(13-bit)	VPE_CSC00[28:16] B0	0.587	601	0x0259	-0.0003	0	0x0000
C0(13-bit)	VPE_CSC01[28:16] C0	0.114	117	0x0075	1.3717	1405	0x057D
A1(13-bit)	VPE_CSC01[28:16] A1	-0.172	-176	0x1F50	1	1024	0x0400
B1(13-bit)	VPE_CSC02[12:0] B1	-0.339	-347	0x1EA5	-0.3365	-345	0x1EA7
C1(13-bit)	VPE_CSC02[27:16] C1	0.511	523	0x020B	-0.6984	-715	0x1D35
A2(13-bit)	VPE_CSC03[12:0] A2	0.511	523	0x020B	1	1024	0x0400
B2(13-bit)	VPE_CSC03[27:16] B2	-0.428	-438	0x1E4A	1.7336	1775	0x06EF
C2(13-bit)	VPE_CSC04[12:0] C2	-0.083	-85	0x1FAB	-0.0016	-2	0x1FFE
D0(12-bit)	VPE_CSC04[27:16] D0	0	0	0x000	-176	-704	0xD40
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	132	528	0x210
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-222	-888	0xC88

Table 10-15. Quantized Coefficients of SDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VPE_CSC00[12:0] A0	0.257	263	0x0107	1.1641	1192	0x04A8
B0(13-bit)	VPE_CSC00[28:16] B0	0.504	516	0x0204	-0.0018	-2	0x1FFE
C0(13-bit)	VPE_CSC01[28:16] C0	0.098	100	0x0064	1.5958	1634	0x0662
A1(13-bit)	VPE_CSC01[28:16] A1	-0.148	-152	0x1F68	1.1641	1192	0x04A8
B1(13-bit)	VPE_CSC02[12:0] B1	-0.291	-298	0x1ED6	-0.3914	-401	0x1E6F
C1(13-bit)	VPE_CSC02[27:16] C1	0.439	450	0x01C2	-0.8135	-833	0x1CBF
A2(13-bit)	VPE_CSC03[12:0] A2	0.439	450	0x01C2	1.1641	1192	0x04A8
B2(13-bit)	VPE_CSC03[27:16] B2	-0.368	-377	0x1E87	2.0178	2066	0x0812
C2(13-bit)	VPE_CSC04[12:0] C2	-0.071	-73	0x1FB7	-0.0012	-1	0x1FFF
D0(12-bit)	VPE_CSC04[27:16] D0	16	64	0x040	-223	-892	0xC84
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	136	544	0x220
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-277	-1108	0xBAC

10.3.5.4 CSC Bypass Mode

CSC module can be bypassed by setting VPE_CSC05[28] BYPASS bit-field to 1.

10.3.6 VPE Chroma Up-Sampler (CHR_US)

The chroma up-sampler (CHR_US) module is used to convert from YCbCr 4:2:0 data format input to YCbCr 4:2:2 format output.

10.3.6.1 Features

- Supports both interlaced and progressive inputs
- 4-tap interpolation filtering
- Filter coefficients are all programmable
- Four sets of coefficients (each set has four coefficients) corresponding to anchor pixels and interpolated pixels of top field and bottom field
- For progressive inputs, the coefficients corresponding to top and bottom field must be identical
- Each coefficient is 14 bit (4.10 format)
- Default filter coefficients are based on Catmull-Rom algorithm
- Capable of removing the half pel vertical offset so all chroma samples are on-grid. This step ensures that the output does not suffer from any kind of rainbow effect due to chroma-upsampling.
- Provides a 10-bit interface in both directions: 10-bit input and 10-bit output
- Support bypass mode for 4:2:2 input

10.3.6.2 Functional Description

The YUV420 input to Chroma Upsampler module must be in the format shown in Figure 10-43, in which the chroma sample lies in the left column of a 2x2 pixel block with half pel vertical shift.

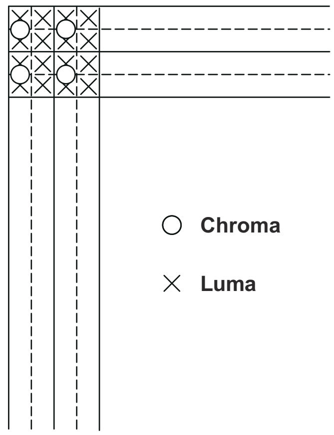


Figure 10-43. 4:2:0 YCrCb Color Space with Chroma Left-aligned

The upsampling is performed by an interpolation filter which uses Catmull-Rom algorithm. The Catmull-Rom Filter is based on four anchor pixels representing a four tap filter. The general 4-tap filter Catmull-Rom filter is defined in Figure 10-44.

$$[\,1 \quad x \quad x\text{^}2 \quad x\text{^}3\,] * \begin{bmatrix} 0 & 1 & 0 & 0 \\ -a & 0 & a & 0 \\ 2a & -3+a & 3-2a & -a \\ -a & 2-a & -(2-a) & a \end{bmatrix}$$

Figure 10-44. 4:2:0 YCrCb Color Space with Chroma Left-aligned

In the previous figure, x is the distance to the interpolated point between the two anchor points. In Figure 10-45, the example shows the desired interpolated point to be ¼ of the distance between Anchor0 and Anchor1. Thus, x = ¼.

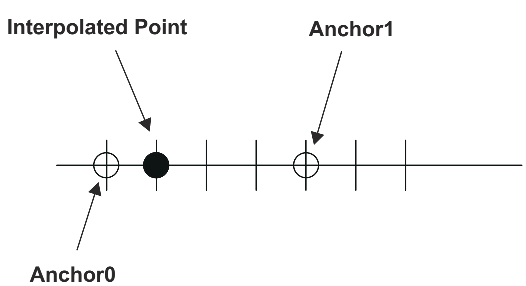


Figure 10-45. 4:2:0 YCrCb Color Space with Chroma Left-aligned

The variable 'a' determines the characteristics of the filter. a = ½ is generally used because the filter will produce an interpolated output that is an exact match to a linear input curve. In the literature, some people have noted that a=0.75 or a=1.0 may be more pleasing to the eye. In the implementation, the filter coefficients are programmable through MMR.

For the interpolated pixel, the variable x defines the positional offset relative to anchor pixel1. Figure 10-46 shows four anchor pixels with Anchor0 being near the top of the image and Anchor3 near the bottom. x is relative to Anchor1. Positive values of x goes down towards Anchor3. Negative x values imply a direction towards Anchor0. For example, if we want to interpolate a pixel midway between Anchor1 and Anchor2, x would be 2/4=1/2. There are four half pels between Anchor1 and Anchor2. Midway is two pels, so x=2/4.

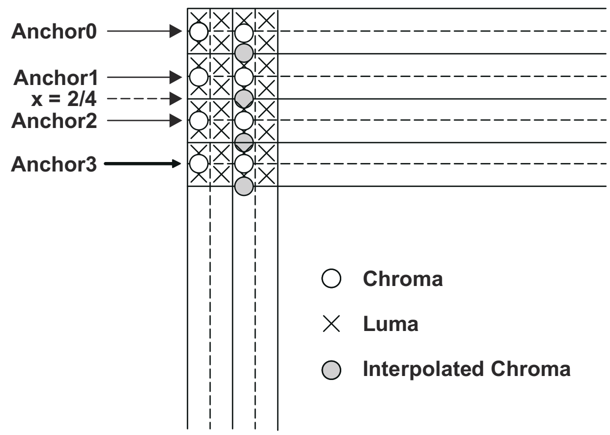


Figure 10-46. Anchor Pixels

In the implementation, we need to interpolate the anchor pixel to get it on-grid. Then, we also need to interpolate a completely new pixel for YPrPb 4:2:2.

For a progressive input, x = -1/4 for getting the anchor pixel on-grid. x = 1/4 for generating the new pixel.

Lines are scanned from the top of the picture to the bottom.

10.3.6.3 For Interlaced YUV420 Input Data

Figure 10-47 shows how 4:2:0 video is split into top and bottom fields in interlaced format. Chroma is attached to alternating Luma lines in the top and bottom fields such that color is equally spread out between the top field and the bottom field. This 4:2:0 interlaced chroma representation is shown in the following figure.

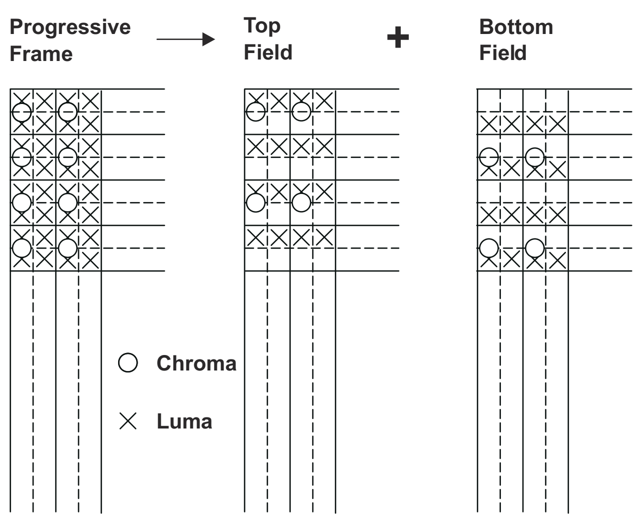


Figure 10-47. 4:2:0 Interlaced Scan

In each interlaced field, the chroma anchor points are separated by 4 pixel lines, or 8 half pels. The 4:2:2 chroma interpolated color space representation of interlaced pictures is shown in Figure 10-48.

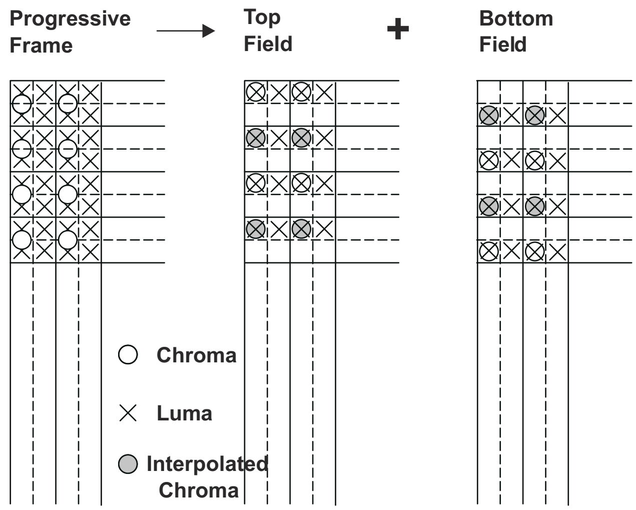


Figure 10-48. Ideal 4:2:2 Chroma Upsampling for Interlaced Scan

Following interpolation, the chroma samples lie on a 4:2:2 grid.

Anchor pixels for the top field have been sited up by half a pel (x = -1/8). The new interpolated pixel is sited 3 half pels (x=3/8) down. Samples from the top field are distinct from samples in the bottom field.

For the bottom field, anchor pixels are sited 3 haf pels up (x = -3/8). The new interpolated pixel is sited 1 half pel down (x=1/8).

The chroma upsampling filter accepts different coefficients for the top field and the bottom field. In the case of progressive input, the coefficients for the top field and bottom field must be the same.

It should be noted that a different implementation could have been chosen to use the same coefficients for the top and bottom fields. Instead of pushing pixels from the top to bottom of a picture, it can be shown that pushing the bottom field through the upsampling filter from the bottom to the top of the picture permits using the same values for x as in the top field case.

10.3.6.4 Edge Effects

Several methods with increasing levels of difficulty resulting in increasing quality can be employed to deal with chroma pixels near the edges. In this module, the edge pixels can be mathematically approximated using the same filter as the rest of the picture. Edge pixels are duplicated going into the filter.

10.3.6.5 Modes of Operation (VPDMA)

In both primary (PRI) and auxiliary (AUX) paths, the mode in which the VPDMA needs to be operated depends whether the chroma upsampler and de-interlacers are enabled or not. Table 1-12 shows the modes of operation.

Table 10-16. VPDMA Modes of Operation

Mode A	Mode B
Input data is 4:2:0	Input data is 4:2:2

These modes need to be set in the following register bit-fields of particular instances being used:

VPE_REG0[17:16] CFG_MODE

LINE_MODE bitfield in VPDMA registers of format RD_LB_CLIENT_CTL_STATUS need to configured as follows:

Mode A corresponds to MMR value 0

Mode B corresponds to MMR value 1

The following are the VPDMA client related bit-fields that need to be configured:

VPE_PRI_CHROMA_CSTAT[9:8] LINE_MODE

VPE_PRI_FLD1_CHROMA_CSTAT[9:8] LINE_MODE

VPE_PRI_FLD2_CHROMA_CSTAT[9:8] LINE_MODE

Note

1. For VPDMA luma clients, Mode B should always be used.
2. For VPDMA chroma clients, Mode A is used for 420 data and Mode B is used for 422 data.

10.3.6.6 Coefficient Configuration

The filter coefficients are left-aligned 14-bit binary values in signed Q4.10 format. The decimal point is between bits 9 and 10 using the convention of the least significant bit being at position zero. The most significant bit, 13, is the sign bit.

In the register map, the most significant nibble of the coefficient is the sign and the integer portion of the value. The next 10 bits represent the fractional portion of the coefficient value.

Chroma upsampling requires two sets of coefficients. Each coefficient set is comprised of four 14-bit Q4.10 values. One set is used for the top field of an interlaced picture, and the other set is used for the bottom field of an interlaced picture. For a progressive picture, both sets must be identical.

The coefficients and settings should be used for the follwoing video source types:

4:2:2 input (progressive or interlaced input)

VPDMA line mode = 1

VPE_REG0[17:16] CFG_MODE = 0x1 (mode B)

CHR_US coefficients are not used in this mode, so values are "don't care"

4:2:0 input (interlaced input):

VPDMA line mode = 0

VPE_REG0[17:16] CFG_MODE = 0x0 (mode A)

VPE_REG0[31:18] ANCHOR_FID0_C0 = 0x51

VPE_REG0[15:2] ANCHOR_FID0_C1 = 0x3d5

VPE_REG1[31:18] ANCHOR_FID0_C2 = 0x3fe3

VPE_REG1[15:2] ANCHOR_FID0_C3 = 0x3ff7

VPE_REG2[31:18] INTERP_FID0_C0 = 0x3fb5

VPE_REG2[15:2] INTERP_FID0_C1 = 0x2e9

VPE_REG3[31:18] INTERP_FID0_C2 = 0x18f

VPE_REG3[15:2] INTERP_FID0_C3 = 0x3fd3

VPE_REG4[31:18] ANCHOR_FID1_C0 = 0x16b

VPE_REG4[15:2] ANCHOR_FID1_C1 = 0x247

VPE_REG5[31:18] ANCHOR_FID1_C2 = 0xb1

VPE_REG5[15:2] ANCHOR_FID1_C3 = 0x3f9d

VPE_REG6[31:18] INTERP_FID1_C0 = 0x3fcf

VPE_REG6[15:2] INTERP_FID1_C1 = 0x3db

VPE_REG7[31:18] INTERP_FID1_C2 = 0x5d

VPE_REG7[15:2] INTERP_FID1_C3 = 0x3ff9

4:2:0 input (progressive input):

VPDMA line mode = 0

VPE_REG0[17:16] CFG_MODE = 0x0 (mode A)

VPE_REG0[31:18] ANCHOR_FID0_C0 = 0x00C8

VPE_REG0[15:2] ANCHOR_FID0_C1 = 0x0348

VPE_REG1[31:18] ANCHOR_FID0_C2 = 0x0018

VPE_REG1[15:2] ANCHOR_FID0_C3 = 0x3fd8

VPE_REG2[31:18] INTERP_FID0_C0 = 0x3fb8

VPE_REG2[15:2] INTERP_FID0_C1 = 0x0378

VPE_REG3[31:18] INTERP_FID0_C2 = 0x00e8

VPE_REG3[15:2] INTERP_FID0_C3 = 0x3fe8

VPE_REG4 to VPE_REG7 are not used so their values are "don't care".

10.3.7 VPE Chroma Down-Sampler (CHR_DS)

When the picture input is 4:2:2, the chroma must be downsampled to 4:2:0 before it is stored into DRAM for later compression by the imaging subsystem. The downsampling is performed by an averaging filter.

An array with chroma samples must be used- C_{IN}[] with range from 0 to N-1. (C_{IN}[0] is the topmost chroma sample and C_{IN}[N-1] is bottom chroma sample) .

The output array C_{OUT}(ranging from 0 to N/2 - 1) is calculated by the following formula:

$C_{OUT}[i] = CLIP[(C_{IN}[2i] + C_{IN}[2i+1]) / 2]CLIP[x]$ (x= 0 to 255).

This filter performs simple averaging of two input lines into one output line.

10.3.8 VPE YUV422 to YUV444 Conversion

As shown on Figure 10-2, prior the CSC the video format needs to be converted from YUV422 to YUV444 format. This conversion only applies to the Chroma samples in the color space, and is implemented using a 4-tap Catmull-Rom algorithm using the following equations:

$C_{OUT}[2*i[= C_{IN}[i]\ C_{OUT}[2*i+1] = CLIP\ (9/16*C_{IN}[i] + 9/16*C_{IN}[i+1] – 1/16*C_{IN}[i-1] – 1/16C_{IN}[i+2] + 1)$

Note

Edge effects are treated by repeating the first and last pixel per line.

10.3.9 VPE Video Port Direct Memory Access (VPDMA)

10.3.9.1 VPDMA Introduction

The VPDMA primary function is to move data between external memory and internal processing modules that source or sink data. VPDMA is capable buffering this data and then delivering the data as demanded to the modules as programmed. The modules that source or sink data are referred to as clients. A channel is setup inside the VPDMA to connect a specific memory buffer to a specific client. The VPDMA centralizes the DMA control functions and buffering required to allow all the clients to minimize the effect of long latency times. The VPDMA also supports a descriptor based mode where lists of descriptors can be setup to configure all the channels as they become available.

Additionally, in a third-party configuration, the VPDMA is capable of performing DMA transfers as requested by the pulsing of an event strobe. The VPDMA is capable of generation of an address which may reach any location in the range for which it is configured. It is capable of moving this data either to or from a Shared Buffer and ultimately to or from a Client Buffer. For the two type of Client Buffers that are used to drive data into a subsystem (Streaming Buffer and Random Access Buffer) data is either pushed into the buffer or pulled out of the buffer dependent upon the direction of the data transfer. For third-party DMA operations the data is transferred into a Client buffer, and then transferred out of the Client Buffer to the transfer destination. These transfers are triggered by an Event pulse to each of the Client Buffers which are Routing Buffer types.

10.3.9.2 VPDMA Basic Definitions

10.3.9.2.1 Client

The modules that source or sink data are referred to as clients. The clients of the VPDMA are the physical between the processing modules (VPE) and external memory. A channel is the mechanism inside the VPDMA that connects a specific memory buffer or transfer to a specific client.

10.3.9.2.2 Channel

The VPDMA requires a channel to be setup for each group of transfers. All the channels are described through a Data Transfer Descriptor that has a common format. The client that the channel is mapped to interprets the information in the descriptor to perform the requested data transfer.

Each of the channels has a type of data that it can support based upon the client that it services. The VPDMA supports four types of channels:

- **YUV Channel** - Clients taking data YUV data
- **RGB Channel** - Clients taking RGB data
- **Miscellaneous Channel** - The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client.
- **Free Channel** - Used in video compositions. The Free channel data type is always ignored as it uses the same data type of the descriptor that first calls the free channel.

10.3.9.2.3 List

A list is a group of descriptors that makes up a set of DMA transfers that need to be completed. The VPDMA supports two types of lists: a Regular List and a Self-Modifying List.

- The **Regular List** is a single list that the VPDMA will execute each descriptor once and initiate an interrupt when the list has completed. A regular list can contain any kind of descriptor without limitation and be of any size.

The VPDMA Controller works on lists of descriptors. In this mode the processor writes the lists of descriptors in the order it wants them executed. It then writes the location of the list to the VPE_LIST_ADDR register, followed by writing the size (bit LIST_SIZE) and type (bit LIST_TYPE) of the list, and list number (bit LIST_NUM) to the LIST_ATTR register. The List Manager module then schedules a DMA transfer to pull in the portion of the list that it can store in internal VPDMA memory. The List Manager will sequentially process the active list of descriptors until either the descriptor requires the use of a client that is currently active, or if it is waiting for the next portion of the list to be transferred from a DMA request. If there is no active list to process the list manager will go into an IDLE mode waiting for any client that is blocking a list or a list DMA transfer to complete.

All the DMA transfers are controlled by List Manager module inside VPDMA. List Manager needs to be loaded with FIRMWARE, before any DMA transfer from memory, after the VPDMA reset. The first MMR write to VPE_LIST_ADDR register after VPDMA reset should be the address of the memory buffer(128-bit aligned, that is, last four bits of the buffer address should be zero) where the firmware is stored. List Manager then schedules a DMA transaction to fetch the firmware and sets the list_attr.rdy bit after the firmware loading is complete.

10.3.9.2.4 Data Formats Supported

Following list summarizes the data formats supported in the VPDMA. For more inforamtion see Section 10.3.9.8, *VPDMA Data Formats*.

- RGB Data Types:
  - RGB16-565
  - ARGB-1555
  - ARGB-4444
  - RGBA-5551
  - RGBA-4444
  - ARGB24-6666
  - RGB24-888
  - ARGB32-8888
  - RGBA24-6666
  - RGBA32-8888
- YUV Data Types:
  - Y 4:4:4
  - Y 4:2:2
  - Y 4:2:0
  - C 4:4:4
  - C 4:2:2
  - C 4:2:0
  - CY 4:2:2
  - YCbC 4:4:4
  - YC 4:2:2

Note

VPDMA supports swapping formats (RGB/BGR and Cb/Cr)

10.3.9.3 VPDMA Client Buffering and Functionality

Table 10-17 lists for each client:

- The channels used, amount of buffering allocated for it, and the shared buffer used for its memory
- The line sizes it handles for tiled and non-tiled memory spaces, as well as any additional features it supports

Table 10-17. VPDMA Client Buffering and Functionality

Client	Channel(s)	Tiled Memory Max Line Size	Non-Tiled Memory Max Line Size	Additional Features
dei_hq_1_chroma	hq_vid1_chroma	1920 (color seperate) 960 (interleaved)	1920 (color seperate) 960 (interleaved)	Virtual Video Buffer with line buffer limitations TILED
dei_hq_1_luma	hq_vid1_luma	1920 (color seperate) 960 (interleaved)	4096	Virtual Video Buffer TILED
dei_hq_2_luma	hq_vid2_luma	1920 (color seperate) 960 (interleaved)	4096	Virtual Video Buffer TILED
dei_hq_2_chroma	hq_vid2_chroma	1920 (color seperate) 960 (interleaved)	1920 (color seperate) 960 (interleaved)	Virtual Video Buffer with line buffer limitations TILED
dei_hq_3_luma	hq_3_luma	1920 (color seperate) 960 (interleaved)	4096	Virtual Video Buffer TILED
dei_hq_3_chroma	hq_3_chroma	1920 (color seperate) 960 (interleaved)	1920 (color seperate) 960 (interleaved)	Virtual Video Buffer with line buffer limitations TILED
dei_hq_mv_in	hq_mv_in	Tiled Data Not Supported	4096	
dei_hq_mv_out	hq_mv_out	Tiled Data Not Supported	4096	
vip1_up_y	vip1_porta_luma vip1_porta_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED

Table 10-17. VPDMA Client Buffering and Functionality (continued)

vip1_up_uv	vip1_porta_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED

10.3.9.4 VPDMA Channels Assignment

Table 10-18 lists all of the channels in VPDMA and its base attributes. The Data Type column states what type of data YUV, RGB or OTHER the channel handles and in parentheses are the legal data type values that can be entered into a data transfer decriptor. The Client field states the name of the Client and in parentheses it states the reference number in Figure 10-2, *VPE Block Diagram*.

Table 10-18. VPDMA Channels Assignment

Channel	Description	Channel Number	Data Type	Client
hq_vid1_luma	High Quality DEI Video 420 Luma Data/ 422 Interleaved Data	0	YUV (0x1,0x2,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_1_luma (7)
hq_vid1_chroma	High Quality DEI Video 420 Chroma Data	1	YUV (0x5,0x6,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_1_chroma (7)
hq_vid2_luma	Low Cost DEI Field Minus 1420 Luma Data	2	YUV (0x1,0x2,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_2_luma (8)
hq_vid2_chroma	Low Cost DEI Field Minus 1 420 Chroma Data	3	YUV (0x5,0x6,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_2_chroma (8)
hq_vid3_luma	Low Cost DEI Field Minus 2 420 Luma Data	4	YUV (0x1,0x2,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_3_luma (9)
hq_vid3_chroma	Low Cost DEI Field Minus 2 420 Chroma Data	5	YUV (0x5,0x6,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_3_chroma (9)
hq_mv	Low Cost DEI Motion Vector	12	OTHER (4)	dei_hq_mv_in (6)
hq_mv_out	Low Cost DEI Motion Vector Write	15	OTHER (4)	dei_hq_mv_out (6)
vip1_porta_luma	Video Input 1 Port A 420 Data Luma	102	YUV (0x1, 0x2,0x7)	vip1_up_y (1)
vip1_porta_chroma	Video Input 1 Port A 420 Data Chroma	103	YUV (0x5,0x6,0x7)	vip1_up_uv (1)
vip1_porta_rgb	Video Input 1 Port A RGB Data	106	RGB (0x0-0x8)	vip1_up_y (1)

10.3.9.5 VPDMA Interrupts

The VPDMA has 4 interrupt group(s). Each group has an interrupt for all the client interrupts, an interrupt for every 32 channels, a interrupt for each list complete, an interrupt for each list notify and an interrupt for for all of the descriptor interrupts. Each of these groups can be individually masked so that only the interrupts specified will trigger the higher level interrupt.

Each interrupt source can be individually masked independently for each seperate interrupt group. A status register bit exists for each interrupt source for for each interrupt group, that is set whenever the interrupt event occurs even when if the interrupt is masked. The status register bit will remain set until cleared by software by writing a one to the status bit.

Table 10-19 shows all interrupt events from the VPDMA that go to VPE top level. The interrupt events are mapped to one interrupt line, INT0, that go to VPE top level.

Table 10-19. VPDMA Interrupt Events

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_channel_group0	VPE_INT0_CHANNEL0_INT_STAT	VPE_INT0_CHANNEL0_INT_MASK	An unmasked channel interrupt for interrupt group 0 in channel register 0 has fired.
vpdma_int_channel_group1	VPE_INT0_CHANNEL1_INT_STAT	VPE_INT0_CHANNEL1_INT_MASK	An unmasked channel interrupt for interrupt group 1 in channel register 0 has fired.
vpdma_int_channel_group2	VPE_INT0_CHANNEL2_INT_STAT	VPE_INT0_CHANNEL2_INT_MASK	An unmasked channel interrupt for interrupt group 2 in channel register 0 has fired.
vpdma_int_channel_group3	VPE_INT0_CHANNEL3_INT_STAT	VPE_INT0_CHANNEL3_INT_MASK	An unmasked channel interrupt for interrupt group 3 in channel register 0 has fired.
vpdma_int_channel_group4	VPE_INT0_CHANNEL4_INT_STAT	VPE_INT0_CHANNEL4_INT_MASK	An unmasked channel interrupt for interrupt group 4 in channel register 0 has fired.
vpdma_int_channel_group5	VPE_INT0_CHANNEL5_INT_STAT	VPE_INT0_CHANNEL5_INT_MASK	An unmasked channel interrupt for interrupt group 5 in channel register 0 has fired.
vpdma_int_list0_complete			List 0 has completed
vpdma_int_list0_notify			The data transfer in list 0 with the Notify Field set in the descriptor has completed
vpdma_int_list1_complete			List 1 has completed
vpdma_int_list1_notify			The data transfer in list 1 with the Notify Field set in the descriptor has completed
vpdma_int_list2_complete			List 2 has completed
vpdma_int_list2_notify			The data transfer in list 2 with the Notify Field set in the descriptor has completed
vpdma_int_list3_complete			List 3 has completed
vpdma_int_list3_notify			The data transfer in list 3 with the Notify Field set in the descriptor has completed
vpdma_int_list4_complete	VPE_INT0_LIST0_INT_STAT	VPE_INT0_LIST0_INT_MASK	List 4 has completed
vpdma_int_list4_notify			The data transfer in list 4 with the Notify Field set in the descriptor has completed
vpdma_int_list5_complete			List 5 has completed
vpdma_int_list5_notify			The data transfer in list 5 with the Notify Field set in the descriptor has completed
vpdma_int_list6_complete			List 6 has completed

Table 10-19. VPDMA Interrupt Events (continued)

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_list6_notify			The data transfer in list 6 with the Notify Field set in the descriptor has completed
vpdma_int_list7_complete			List 7 has completed
vpdma_int_list7_notify			The data transfer in list 7 with the Notify Field set in the descriptor has completed
vpdma_int_client	VPE_INT0_CLIENT0_INT_STAT VPE_INT0_CLIENT1_INT_STAT	VPE_INT0_CLIENT0_INT_MASK VPE_INT0_CLIENT1_INT_MASK	Client Interrupt
vpdma_int_descriptor	VPE_INT0_LIST0_INT_STAT	VPE_INT0_LIST0_INT_STAT	Descriptor Interrupt

In Table 10-19 above, the “channel_group”, “client” and “descriptor” interrupts are actually a set of additional interrupts. When software receives an interrupt from a “channel_group,” “client,” or “descriptor” it must read the appropriate register within the VPDMA (refer to Table 10-20 to determine what the actual interrupt was).

Table 10-20. VPE Interrupt Sources

Interrupt	Interrupt Group	Description
channel_hq_mv	channel_group0	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager.
channel_hq_mv_out	channel_group0	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager.
channel_hq_vid1_chroma	channel_group0	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid1_luma	channel_group0	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid2_chroma	channel_group0	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid2_luma	channel_group0	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid3_chroma	channel_group0	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid3_luma	channel_group0	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.

Table 10-20. VPE Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_porta_chroma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_luma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_rgb	channel_group3	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point.
client_dei_hq_1_chroma	client	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_1_luma	client	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_2_chroma	client	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_2_luma	client	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_3_chroma	client	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_3_luma	client	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_mv_in	client	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_mv_out	client	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_up_uv	client	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.

Table 10-20. VPE Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
client_vip1_up_y	client	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vpi_ctl	client	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
control_descriptor_int0	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0.
control_descriptor_int1	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1.
control_descriptor_int10	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10.
control_descriptor_int11	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11.
control_descriptor_int12	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12.
control_descriptor_int13	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13.
control_descriptor_int14	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14.
control_descriptor_int15	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15.
control_descriptor_int2	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2.
control_descriptor_int3	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3.
control_descriptor_int4	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4.
control_descriptor_int5	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5.
control_descriptor_int6	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6.
control_descriptor_int7	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7.
control_descriptor_int8	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8.
control_descriptor_int9	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9.
list0_complete	list0_complete	List 0 has completed and a new list can be loaded.
list0_notify	list0_notify	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel.
list1_complete	list1_complete	List 1 has completed and a new list can be loaded.
list1_notify	list1_notify	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel.
list2_complete	list2_complete	List 2 has completed and a new list can be loaded.
list2_notify	list2_notify	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel.
list3_complete	list3_complete	List 3 has completed and a new list can be loaded.
list3_notify	list3_notify	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel.

Table 10-20. VPE Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
list4_complete	list4_complete	List 4 has completed and a new list can be loaded.
list4_notify	list4_notify	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel.
list5_complete	list5_complete	List 5 has completed and a new list can be loaded.
list5_notify	list5_notify	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel.
list6_complete	list6_complete	List 6 has completed and a new list can be loaded.
list6_notify	list6_notify	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel.
list7_complete	list7_complete	List 7 has completed and a new list can be loaded.
list7_notify	list7_notify	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel.
other		Any channel that is not assigned to a specific client has completed.

10.3.9.6 VPDMA Descriptors

The VPDMA needs to be programmed through descriptors (a pre-defined structure of eight or four 32-bit words depending on type of descriptors) other than VPDMA Memory Mapped Registers (MMR). Descriptors are of three types:

1. **Data Transfer Descriptors** - A memory structure used to describe a desired memory transaction to or from a client.
2. **Control Descriptors** - A memory structure used to perform a control operation inside the DMA controller
3. **Configuration Descriptors** - A memory structure used to described a setup that should be applied to an processing modules like MMR write, scalar coefficient write etc.

10.3.9.6.1 Data Transfer Descriptors

In order to set up data transfers from the VPDMA, a data transfer descriptor is added into a list. The fields used for an Inbound and Outbound descriptor vary slightly. An outbound transfer can have two different outbound transfers. The two transfers are the main data transfer and a write of an Inbound Descriptor. The created inbound descriptor is formatted so it can be read back in directly to the next channel specified in the original descriptor.

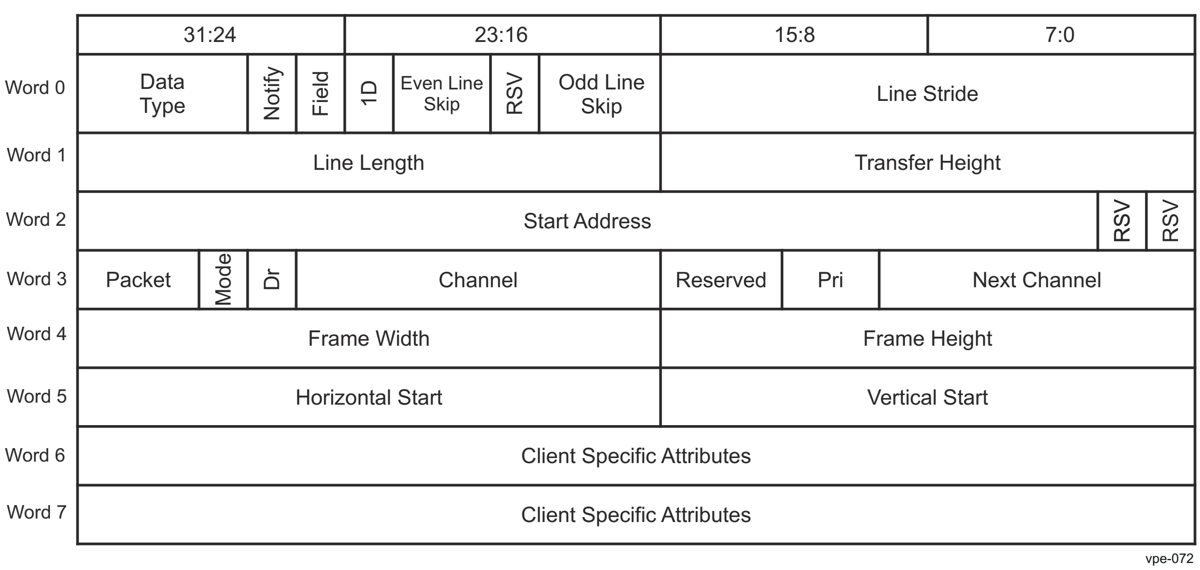


Figure 10-49. Inbound Data Transfer Descriptor Format

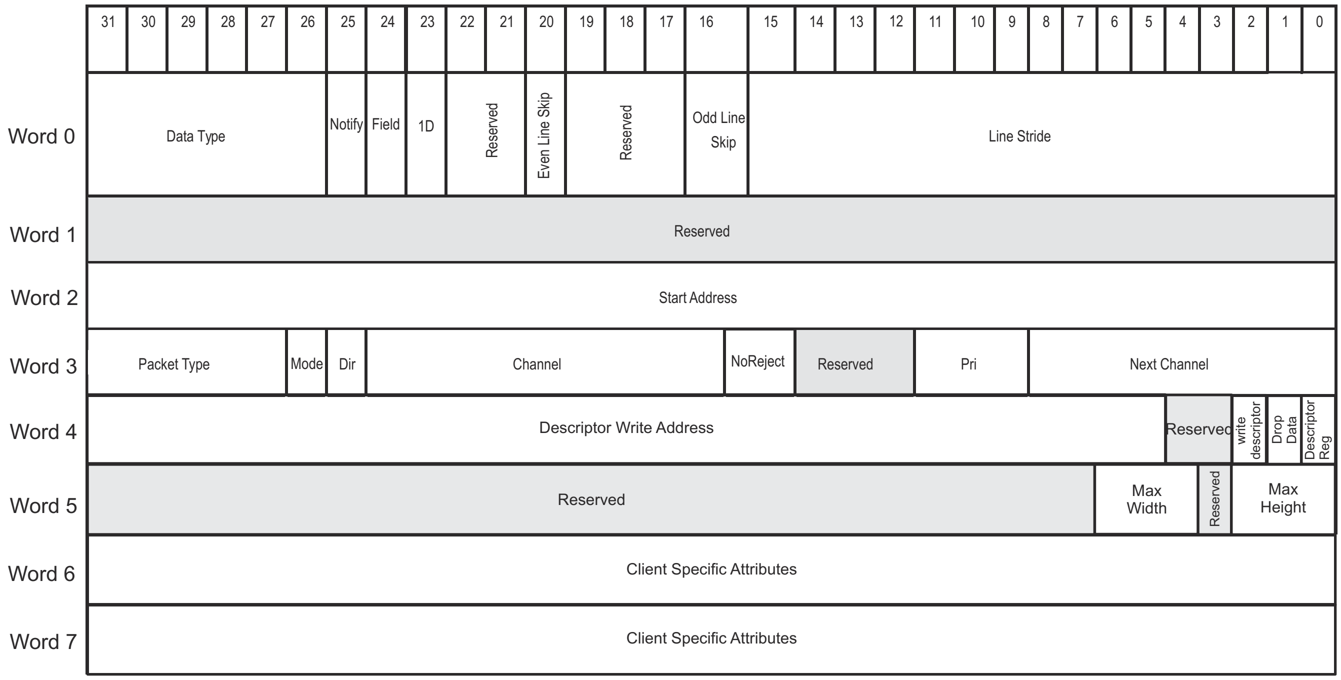


Figure 10-50. Outbound Data Transfer Descriptor Format

The general data transfer Descriptor formats can be seen in the above figures. The descriptor consists of 8 × 32bit words. A Data Transfer Descriptor will be removed from the list when the resource specified by the Channel field is free. If the Channel is not free when the list reaches a data transfer descriptor then the list will stall until the current transfer on the channel has completed.

10.3.9.6.1.1 Data Packet Descriptor Word 0 (Data)

Table 10-21. Data Packet Descriptor Word 0 Field Descriptions

Bit	Field	Value	Description
31:26	Data Type		**Miscellaneous Channel**
			Sets the pixel size in bits plus 1
			RGB Channel
		0	RGB16-565
		1h	ARGB-1555
		2h	ARGB-4444
		3h	RGBA-5551
		4h	RGBA-4444
		5h	ARGB24-6666
		6h	RGB24-888
		7h	ARGB32-8888
		8h	RGBA24-6666
		9h	RGBA32-8888
		10h	BGR16-565
		11h	ABGR-1555
		12h	ABGR-4444
		13h	BGRA-5551
		14h	BGRA-4444
		15h	ABGR24-6666
		16h	BGR24-888
		17h	ABGR32-8888
		18h	BGRA24-6666
		19h	BGRA32-8888
			YUV Channel
		0	Y 4:4:4
		1	Y 4:2:2
		2	Y 4:2:0
		4	C 4:4:4
		5	C 4:2:2
		6	C 4:2:0
		7	CY 4:2:2
		8	YCbC 4:4:4
		14h	Cb 4:4:4
		15h	Cb 4:2:2
		16h	Cb 4:2:0
		17h	CbY 4:2:2
		27h	YC 4:2:2
		37h	YCb 4:2:2
25	Notify	0-1	Send List Notification Interrupt upon last transfer of this channel
24	Field	0-1	Field Value
23	1D	0-1	The transfer is one dimensional. The transfer and frame sizes are combined to make a single 32 bit transfer size. For writes this value is passed to the generated descriptor. This feature is not supported by all clients. Only clients that support the feature will recognize this bit.

Table 10-21. Data Packet Descriptor Word 0 Field Descriptions (continued)

Bit	Field	Value	Description
22:20	Even Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
19	Reserved		Reserved for future use
18:16	Odd Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
15:0	Line Stride	0-FFFFh	Address stride between lines in bytes

10.3.9.6.1.1.1 Data Type

Bits 31-26 indicate the type of data that is to be transferred. This value is used to compute the number of bytes per pixel so that transactions may be made for the appropriate amount of data. The types of data are dependent on the type of channel. The VPDMA descriptor data types RGB or YUV are defined associated with the VPDMA channel assignment. This helps the engine to distinguish between the overlapping values of the descriptor data types for RGB and YUV data.

- For the Miscellaneous channel, the Data Type selects the size in bits of the data. The range is from 0 to 63 to represent the sizes from 1 to 64 bits.
- For a YUV channel, the Data Type determines, if the data channel is interleaved or color space separate. If color spaced separate, it is still assumed that the two chroma pixels are interleaved.

CAUTION

VPDMA defines the component ordering for its RGB data types in the opposite direction of what commonly used image identifiers expect. To avoid color component swapping in the display and/or in the video/image data written out to the memory, the proper Data Type settings for both RGB and YUV data types must be made. The following paragraphs provide more details on how to set Data Type correctly, in order to match the data stored or expected in the memory.

Setting RGB Data Types

The commonly used RGB format identifiers require the color components to be stored in a little-endian style, where the left most component is the LSB component.

- For an ARGB data type, the A component is the LSB location, as shown in Table 10-22 and Table 10-23;
- For a BGRA data type, the B component would be in the LSB location;

Table 10-22. Common ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
A	R	G	B

Table 10-23. Common ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
B	G	R	A

VPDMA specifies its component ordering in the big-endian style, which requires the data to be stored in the reversed order. Example with ARGB data type is shown in Table 10-24 and Table 10-25. The VPDMA ordering for ARGB data type matches the common BGRA data format.

Table 10-24. VPDMA ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
B	G	R	A

Table 10-25. VPDMA ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
A	R	G	B

In order color components to be mapped correctly and to avoid swapping, the reversal must be taken into consideration when configuring the Data Type in the VPDMA transfer descriptor.

Table 10-26 shows the proper settings required for RGB data types for both storage schemes.

Table 10-26. VPDMA Descriptor RGB Data Type Mapping

Destination Image		VPDMA Data Type Mapping Value	
RGB Component order	Common Image Format Names	**Column A** Data stored in the VPDMA defined order	**Column B** Data stored in the opposite of VPDMA defined order
RGB	RGB16-565	0x0	0x10
	ARGB-1555	0x1	0x13
	ARGB-4444	0x2	0x14
	RGBA-5551	0x3	0x11
	RGBA-4444	0x4	0x12
	ARGB24-6666	0x5	0x18
	RGB24-888	0x6	0x16
	ARGB32-8888	0x7	0x19
	RGBA24-6666	0x8	0x15
	RGBA32-8888	0x9	0x17
BGR	BGR16-565	0x10	0x0
	ABGR-1555	0x11	0x3
	ABGR-4444	0x12	0x4
	BGRA-5551	0x13	0x1
	BGRA-4444	0x14	0x2
	ABGR24-6666	0x15	0x8
	BGR24-888	0x16	0x6
	ABGR32-8888	0x17	0x7
	BGRA24-6666	0x18	0x5
	BGRA32-8888	0x19	0x9

In Table 10-26, if the application uses the same data type definition as the VPDMA (that is, RGB24 refers to the B in the LSB), the data types in Column A should be used. But, if the application expects the common data type component order for RGB data type names, the VPDMA data types in Column B should be used.

For example:

- To display an ARGB32-8888 source image data with A in the LSB, the data type in the descriptor should be set to 0x19. But, to display an ARGB32-888 source image data with B in the LSB, the data type in the descriptor should be set to 0x7.

Setting YUV Data Types

There is no component order reversal for YUV data types. The VPDMA uses generic data type names to specify the memory storage format and the application simply needs to follow the VPDMA defined ordering.

Table 10-27 shows how common YUV data types map to the VPDMA YUV data types in order to clarify the YUV data type configuration.

Table 10-27. VPDMA Descriptor YUV Data Type Mapping

Source YUV Image Types			VPDMA Data Type Mapping (Value)		
Chroma Sub-sample	Common YUV Image Format Type Names	Memory Packed Order [MSB - LSB]	Luma/Chroma Interleaved Channel	Luma-only Channel	Chroma-only Channel
444	YUV	V U Y	YC 4:4:4 (0x8)		
	UVY	Y V U	Cb 4:4:4 (0x14)		
422	NV16 (YUV422SP_UV)	V U		Y 4:2:2 (0x1)	C 4:2:2 (0x5)
	NV16 (YUV422SP_VU)	U V		Y 4:2:2 (0x1)	Cb 4:2:2 (0x15)
	YUV2/YUYV/V422 (YUV422I_YUYV)	V Y U Y	YC 4:2:2 (0x7)		
	YUV422I_YVYU	U Y V Y	CbY 4:2:2 (0x17)		
	Y422/UYVY (YUV422I_UYVY)	Y V Y U	YC 4:2:2 (0x27)		
	YUV422I_VYUY	Y U Y V	YCb 4:2:2 (0x37)		
420	NV12 (YUV420SP_UV)	V U		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	C 4:2:0 (0x6) YC 4:2:2 (0x7) (see (1))
	NV21 (YUV420SP_VU)	U V		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	Cb 4:2:0 (0x16) YC 4:2:2 (0x7) (see (1))

(1) If 422 source data is used, unused component data fetched (either Luma or Chroma) will be discarded.

For further details on the data formats, refer to Section 10.3.9.8, *VPDMA Data Formats*.

10.3.9.6.1.1.2 Notify

The Notify bit is used in conjunction with the Notify interrupts and when the channel has completed the DMA transfer the Notify Interrupt for the list that contains the descriptor will fire. The last Notify bit set for a specific list will be used so only a single Notify should be set in a list.

10.3.9.6.1.1.3 Field

The Field bit is the field value of the data that will be passed down to the clients. If the data is interlaced, then this value should be cleared to 0.

10.3.9.6.1.1.4 1D

This bit is set if a large one dimensional frame needs to be send to the client. In this case the stride is ignored and for the write the stride the generated descriptor will always be 0. If this bit is set then the transfer length and transfer height and frame width and frame height fields are combined to form one 32 bit field with the upper 8 bits reserved and the lower 24 bits being the size of the frame in pixels.

10.3.9.6.1.1.5 Even Line Skip

The Even Line skip is used with Line Stride to generate the next line address on an even line. All frames start on line 0. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

10.3.9.6.1.1.6 Odd Line Skip

The Odd Line skip is used with Line Stride to generate the next line address on an odd line. All frames start on line 0, so this will apply starting with the second line. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

10.3.9.6.1.1.7 Line Stride

Bits 15:0 are the stride between lines in bytes at the external address. This value is added or subtracted based upon an adjustment using the current skip value. Operation of the external address pointer shall load the Source Address upon start of the transfer, then at the end of each line increment or decrement by the value computed using the Line Stride and Skip value for the line. The line stride must be aligned to an L3 data bus width. The lower bits of the stride will always be treated as zero to force the alignment.

10.3.9.6.1.2 Data Packet Descriptor Word 1

Table 10-28. Data Packet Descriptor Word 1 Field Description

Bits	Name	Description
31:16	Line Length	Line Length in Pixels
15:0	Transfer Height	Number of rows in transfer.

10.3.9.6.1.2.1 Line Length

Bits 31-16 are the line length in pixels of the current channel. This is ignored for the outbound transfer as the client will provide the line length with the end of line signal on the client interface. The maximum supported line length is currently 4096.

10.3.9.6.1.2.2 Transfer Height

Bits 15-0 are the number of lines to be transferred in the current channel. This is ignored for outbound transfers as the client will provide the line length with the end of frame signal on the client interface. The maximum supported transfer size is currently 2048.

10.3.9.6.1.3 Data Packet Descriptor Word 2

Table 10-29. Data Packet Descriptor Word 2 Field Descriptions

Bit	Field	Value	Description
31:0	Start Address		32-bit data source address [31:0]
			If Mode is TILED, then TILER specific ADDRESS Map is used:
			Bits 31-29:
		0	0-degree view
		1h	180-degree view + mirroring
		2h	0-degree view + mirroring
		3h	180-degree view
		4h	270-degree view + mirroring
		5h	270-degree view
		6h	90-degree view
		7h	90-degree view + mirroring
			Bits 28-27:
		0	8-bit container
		1h	16-bit container
		2h	32-bit container
		3h	Page Mode
			If Mode is NORMAL, then bits 31-26 are the upper bits of the address.

10.3.9.6.1.3.1 Start Address

This is the byte aligned address for the first data transfer. The address on the OCP bus will always be word aligned.

10.3.9.6.1.4 Data Packet Descriptor Word 3

Table 10-30. Data Packet Descriptor Word 3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xa
26	Mode	0= Normal, 1=TILED
25	Direction	Inbound = 0, Outbound = 1
24:16	Channel	Channel for which this descriptor describes
15	Reserved	Reserved for future use
11:9	Priority	Only Bit 9 and Bit 11 are used to set the priority. Bit 10 is ignored. Highest = 0, Lowest = 3 By default, hardware assigns priority = 3. This priority level is used in the arbitration between the masters (for DDR access). See Section 10.3.9.6.1.4.5, *Priority*, for more details.
8:0	Next Channel	Next Channel to execute on a line or the channel to use in the generated write descriptor.

10.3.9.6.1.4.1 Packet Type

Bits 31:27 are a unique code which indicates that this Descriptor is a VPDMA descriptor.

10.3.9.6.1.4.2 Mode

Bit 26 is used to indicate if the transfer is to regular memory space or to Tiled memory space. The VPDMA will use this to determine if it does standard raster based addressing or if it assumes that the data is stored in TILER format. If data is stored in TILER format then the buffer is turned into 2 or 4 line buffers depending on the container type. For shared clients that use the memory, such as the ancillary data and the VIP port, only one can be active, if the mode field is set. Only clients that support Tiling will properly pack the data for tiling on the output interface. This must only be set for channels going to clients that support the TILING feature in the client configuration.

10.3.9.6.1.4.3 Direction

Bit 25 is used to indicate the direction of transfer. This bit indicates that the data flow is from an external source to an internal buffer (inbound) or data transfers form an internal buffer to an external location (outbound).

10.3.9.6.1.4.4 Channel

Bits 24:16 are the Channels which is supported by the descriptor. This is the identification of the specific Channel which is controlled by the contents of the descriptor. The channel assignments can be found in the channel table.

10.3.9.6.1.4.5 Priority

Bits 11:9 are set to indicate priority of the transfer, these are directly mapped to the OCP reqinfo bits.

10.3.9.6.1.4.6 Next Channel

Bits 8:0 give the next channel to use to create a composite frame. The next channel must be to a free channel. The last channel of a row should point back to the initial channel which must be a channel tied directly to a client. The Descriptor for the Next Channel must be of the same type as the current descriptor.

10.3.9.6.1.5 Data Packet Descriptor Word 4

10.3.9.6.1.5.1 Inbound data

Table 10-31. Data Packet Descriptor Word 4 Inbound Data Field Descriptions

Bits	Name	Description
31:16	Frame Width	Width of the client frame.

Table 10-31. Data Packet Descriptor Word 4 Inbound Data Field Descriptions (continued)

Bits	Name	Description
15:0	Frame Height	Height of the client frame

10.3.9.6.1.5.1.1 Frame Width

Bits 31:16 indicate the width in pixels of the frame. This is the width of the entire frame and not just the width of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum width is 4096.

10.3.9.6.1.5.1.2 Frame Height

Bits 15:0 indicate the height in pixels of the entire frame. This is the height of the entire frame and not just the height of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum height is 2048.

10.3.9.6.1.5.2 Outbound data

Table 10-32. Data Packet Descriptor Word 4 Outbound Data Field Descriptions

Bits	Name	Description
31:5	Descriptor Write Address	The 32 byte aligned location to write an inbound descriptor
2	Write Descriptor	If set to 1, a descriptor will be generated when the client completes a frame.
1	Drop Data	If set to 1, the data will not be written out. Also, if this is set, the write descriptor bit must be set. This allows for descriptors to only be written out so that software can determine the size of the required buffer before data is written out.
0	Use Descriptor Register	If set to 1, the CURRENT_DESCRIPTOR register will be used for the write address location. If set to 0, the Descriptor Write Address field in this word will be used for the Descriptor Write Address.

10.3.9.6.1.5.2.1 Descriptor Write Address

Bits 31:5 set the 32 byte address to write the generated descriptor. This address is only used if the Write Descriptor bit is set to 1 and the Use Descriptor Register bit is set to 0. If this case is met when the channel is complete a descriptor that meets the inbound descriptor format will be written to the location specified by this field. This allows for software to have a specific channel write its descriptor to a specific address no matter what order the channel completes compared to other outbound channels.

10.3.9.6.1.5.2.2 Write Descriptor

Bit 2 determines if a descriptor should be written out when the client is completed. If this bit is set the descriptor will be written. The format of the descriptor will be of an Inbound Data Transfer descriptor with the LINE LENGTH and TRANSFER HEIGHT fields determined by the counters in the clients. This means that the direction bit will be the opposite of the inbound descriptor. The FRAME WIDTH and FRAME HEIGHT values will match the LINE LENGTH and TRANSFER HEIGHT fields. The CHANNEL and NEXT CHANNEL fields will match the NEXT CHANNEL field of the original descriptor. The FIELD bit will match the source field captured by the client. The NOTIFY bit will always be 0. All other fields will match the original outbound descriptor. If the Outbound descriptor MODE is set to TILED data then the descriptor address used will be in TILED data space and software must ensure that the address is in page mode for the address of the descriptor.

10.3.9.6.1.5.2.3 Drop Data

Bit 1 determines if the data should be written out or not. In some cases software might only want to write the descriptor out to determine the size of the buffer that will be required to store incoming data. By setting this bit,no data will be written out. If this bit is set then the Write Descriptor bit MUST be set. Bit 0 determines where the descriptor should be written. If it is desired that all the descriptors are written in order to a set queue then this bit should be set and the CURRENT_DESCRIPTOR, DESCRIPTOR_TOP and DESCRIPTOR_BOTTOM registers are used to define the location of the descriptor queue that will be written.

10.3.9.6.1.5.2.4 Use Descriptor Register

Bit 0 determines where the descriptor should be written. If it is desired that all the descriptors are written in order to a set queue then this bit should be set and the CURRENT_DESCRIPTOR, DESCRIPTOR_TOP and DESCRIPTOR_BOTTOM registers are used to define the location of the descriptor queue that will be written.

10.3.9.6.1.6 Data Packet Descriptor Word 5

10.3.9.6.1.6.1 Outbound data

Table 10-33. Data Packet Descriptor Word 5 Outbound Data Field Descriptions

Bits	Name	Description
6:4	Max Width	The maximum allowable pixels per line. 0: Unlimited Line Size 1: Use VPE_MAX_SIZE1 Max Width field 2: Use VPE_MAX_SIZE2 Max Width field 3: Use VPE_MAX_SIZE3 Max Width field 4: 352 pixels 5: 768 pixels 6: 1280 pixels 7: 1920 pixels Others: Reserved
2:0	Max Height	The maximum allowable lines per frame. 0: Unlimited Frame Size 1: Use VPE_MAX_SIZE1 Max Height field 2: Use VPE_MAX_SIZE2 Max Height field 3: Use VPE_MAX_SIZE3 Max Height field 4: 288 lines 5: 576 lines 6: 720 lines 7: 1080 lines Others: Reserved

Note

Width and Height are set in the following register bit-fields:

- For VPE_MAX_SIZE1: VPE_MAX_SIZE1[31:16] MAX_WIDTH and VPE_MAX_SIZE1[15:0] MAX_HEIGHT
- For VPE_MAX_SIZE2: VPE_MAX_SIZE2[31:16] MAX_WIDTH and VPE_MAX_SIZE2[15:0] MAX_HEIGHT registers
- For VPE_MAX_SIZE3: VPE_MAX_SIZE3[31:16] MAX_WIDTH and VPE_MAX_SIZE3[15:0] MAX_HEIGHT

10.3.9.6.1.6.1.1 Max Width

Bits 6:4 are encoded to set the maximum transferred line size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed pixels the data will continue to be received from the client but will not be sent to external memory until an end of line is received from the client sending data. The outbound descriptor if created will still have the transmitted size of the last line of the frame which will match the max width. If the frame does not exceed the max width then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0. If Max Width is 0 and the line size is larger then 4096 pixels then the address counters will overflow and the data at the start of the line will be overwritten.

10.3.9.6.1.6.1.2 Max Height

Bits 2:0 are encoded to set the maximum transferred frame size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed lines the data will continue to be received from the client but will not be sent to external memory until an end of frame is received from the client sending data. The outbound descriptor if created will still have the transmitted number of lines received which would match the max height. If the frame does not exceed the max height then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0.

10.3.9.6.1.7 Data Packet Descriptor Word 6/7 (Data)

The words 4/5 give a 64 bit of configuration that can be passed specifically to the module that supports it. This is passed directly down to the module through a VPI Control port. Please see the section on the specific clients for the format of this data.

10.3.9.6.2 Configuration Descriptor

The Configuration Descriptor is used in a List to setup a client configuration. The configuration descriptor consists of a header and a payload portion. The payload can either be part of the list that the descriptor is contained in or it can be at a separate location. The VPDMA will pass the payload of the configuration descriptor down to the client over the VPI Control port interface or through the MMR configuration port depending on the destination field. A Configuration Descriptor to any single destination except Destination 0 must be on a single list. Configuration Descriptors to different destinations may be on different lists..

The Configuration Descriptor Header is 4 × 32 bit words. A configuration descriptor is not consumed until the destination specified has received the entire configuration payload. Therefore the list is expected to stall while the configuration takes place. This ensures that all descriptors after a configuration descriptor in the list will occur after the desired configuration.

10.3.9.6.2.1 Configuration Descriptor Header Word0

Table 10-34. Configuration Descriptor Header Word0 Field Descriptions

Bits	Name	Description
31:0	Address Offset of the Destination	This is the address offset location in the destination that the block of data in the payload should be written. This field is only used if the descriptor Class is a block type. Otherwise this field is reserved.

10.3.9.6.2.2 Configuration Descriptor Header Word1

Table 10-35. Configuration Descriptor Header Word1 Field Descriptions

Bits	Name	Description
15:0	Number of Data Words	Length of First Data Packet for Class 1(block).

10.3.9.6.2.2.1 Number of Data Words

Bits 15:0 indicate the length of the first data block if the class is 1 as specified in Configuration Descriptor Header Word3. If the class is not 1 then this field should be set to 0.

10.3.9.6.2.3 Configuration Descriptor Header Word2

Table 10-36. Configuration Descriptor Header Word2 Field Descriptions

Bits	Name	Description
31:0	Payload Location	Pointer to the data payload

10.3.9.6.2.3.1 Payload Location

Bits 31:0 contain the pointer to the data payload if the command packet is an indirect type. This value along with the Payload Length will be combined to issue a DMA transaction that must complete before the List Manager can finish processing this descriptor. The list will be made inactive pending this DMA completion. This address should be on a 16 byte boundary so the lower 4 bits should always be 0.

10.3.9.6.2.4 Configuration Descriptor Header Word3

Table 10-37. Configuration Descriptor Header Word3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xb
26	Direct	Direct Command = 1 Indirect Command = 0
25:24	Class	0 = Address, Data Set 1 = Block 2,3 Reserved for Future Use
23:16	Destination	Destination of the configuration payload
15:0	Payload Length	Length of Payload in Words.

10.3.9.6.2.4.1 Packet Type

Bits 31:27, this field indicates the type of descriptor and should be set 0xB for configuration descriptor.

10.3.9.6.2.4.2 Direct

Bit 26, this field indicates that the descriptor is a direct command or indirect command. A direct command means that the payload is contiguous with the descriptor and will be pulled in by the list manager descriptor control as it processes the list. A direct payload must end on or before a 256 byte boundary in the list. An indirect command is when the Address is located in Word2 is a pointer to the data payload. A DMA transaction will be scheduled to bring the payload into the List Manager to allow for the payload to be sent to the destination.

10.3.9.6.2.4.3 Class

Bits 25:24, this field indicates the type of payload is associated with command descriptor, and thus how to handle the payload.

A 0 indicates that the payload consists of blocks of data with each block having an address and length. The first word after a sub-block contains the next address and length in bytes. This allows for writing to multiple configuration regions in a client. The Configuration Descriptor word1 is the first address and length pair. All addresses used must be larger then the previous address for all destinations except for the MMR destination. If a sub block length is not evenly divisible into 16 then zeroes should be padded in the payload so that the Next Client Address and Length field start on a 16 byte boundary.

10.3.9.6.2.4.3.1 Address Data Block Format

Table 10-38. Address Data Block Format Field Descriptions

Bits	Name	Description
31-0		Next Client Address
31-0		Configuration for Next Client Address
31-0		Configuration for Next Client Address + 4
31-0		Configuration for Next Client Address + 8
31-0		Configuration for Next Client Address + 12
31-0		Configuration for Next Client Address + 16
31-0		Next Client Address 2
15-0		Sub Block Length

A 1 indicates the payload is simply block data. The data is contiguous and starts at the offset of the destination as specified in word1.

10.3.9.6.2.4.4 Destination

The Destination field is used to determine where the configuration payload should be sent. The values for the destination field can be seen in the table below.

Table 10-39. Destination Field Description

Destination Value	Actual Destination	Description
0	mmr_client	Write to MMR registers to setup other modules
4	VPE Scaler	VPE Scaler Coefficient Tables

10.3.9.6.2.4.5 Descriptor Length

Bits 15:0, this field indicates the size of the payload in words for the command. This is 128 bit words. The maximum number of words is 1023 words in a single payload

10.3.9.6.3 Control Descriptor

10.3.9.6.3.1 Generic Control Descriptor Format

A control descriptor is the mechanism that is used in a list to give commands to the List Manager. These descriptors are not considered consumed until the List Manager has done the instruction required by the descriptor. All control descriptors have a common Header located at Word 3 but the remaining words are based on the specific control descriptor.

10.3.9.6.3.2 Control Descriptor Header Description

Table 10-40. Control Descriptor Header Description

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xc
26:25	Reserved	Reserved
24:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	The type of control descriptor that should be run by the List Manager

10.3.9.6.3.2.1 Packet Type

This field indicates a VPDMA control descriptor.

10.3.9.6.3.2.2 Source

The source is combined with the control field to determine what the control descriptor should use as the source for its synchronization event.

10.3.9.6.3.2.3 Control

The Control field defines the specific function of the descriptor. Table 10-41 lists the different control descriptors.

10.3.9.6.3.3 Control Descriptor Types

Table 10-41. Control Descriptor Types Summary

Control Descriptor	Control Field Value	Description
Sync on Client	0	Wait for the client attached to the channel specified to reach a certain event before proceeding.
Sync on List	1h	Wait for another list(s) to reach its Sync on List Descriptor
Sync on External Event	2h	Wait for a Register write to the VPE_LIST_STAT_SYNC bit specified or for an external event.
Sync on Channel	4h	Wait for the channel specified to complete
Change Client Interrupt	5h	Change the interrupt event for a client interrupt but do not wait for the event to occur.

Table 10-41. Control Descriptor Types Summary (continued)

Control Descriptor	Control Field Value	Description
Send Interrupt	6h	Generate an interrupt event on one of the Control Descriptor Interrupts
Reload List	7h	Reload the list from the location and size specified.
Abort Channel	8h	Abort the transfer in the channel specified.

10.3.9.6.3.3.1 Sync on Client

A Sync on Client Control descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. For a client that supports multiple channels then only an event on the portion of the client that supports that client will cause the interrupt to be generation. After configuring the interrupt generation event the list will then stall until that event has occurred.

Table 10-42. Sync on Client Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.
15:0	LINE_COUNT	Specify the line where a line based event would trigger

Table 10-43. Sync on Client Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 0

10.3.9.6.3.3.2 Sync on List

The Sync on List Control descriptor is a mechanism to ensure that multiple lists have all reached a common point. The Sync on List descriptor uses the Source field as a mask for each list that should be synchronized which must include the list where the control descriptor resides. Once all lists specified in the source field have reached their Sync on List descriptor all lists will resume with the lowest list number resuming first. All Sync on List Control descriptors in each of the lists must have the same source field so that all lists will synchronize upon reaching that point.

If it is desired to synchronize list 0 and list 1 then the source field would be 0x3. If it is desired to synchronize list 1, list 3, and list 4 then the source field would be 0x1a. Word 0, Word 1 and Word 2 are reserved.

Table 10-44. Sync on List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 1h

10.3.9.6.3.3.3 Sync on External Event

A Sync on External Event descriptor ensures an external event has occurred before proceeding. The external event can be a write to a bit of the VPE_LIST_STAT_SYNC register or other external events that are determined at VPDMA elaboration to have occurred. This descriptor can be used to allow for external software to control progression of the list. The descriptor can also be used to select external signals that are brought into the VPDMA. The current implementation just synchronize on the VPE_LIST_STAT_SYNC bit for the list number that called the descriptor.

Table 10-45. Sync on External Event Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 2h

10.3.9.6.3.3.4 Sync on Channel

A Sync on Channel descriptor stalls the list until the channel specified is free. If the channel is already free then the descriptor will not cause the list to stall. Word 0, Word 1 and Word 2 are reserved.

Table 10-46. Sync on Channel Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 4h

10.3.9.6.3.3.5 Sync on LM Timer

A Sync on LM Timer descriptor sets a value from the current timer position to wait. The LM timer is a free running counter at the LM processing clock. The Timer Value in the descriptor is added to the value of the timer at the time the descriptor is received and the list will stall for this many cycles before it becomes active again.

10.3.9.6.3.3.6 Change Client Interrupt

A Change Channel Interrupt Source descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. The list will not stall on this descriptor. The format of the fields is identical to the Sync on Client Descriptor. Word 0 is reserved.

Table 10-47. Change Client Interrupt Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.
15:0	LINE_COUNT	Specify the line where a line based event would trigger.

Table 10-48. Change Client Interrupt Field Descriptions (Word - 2)

Bits	Name	Description
31:4	Reserved	Reserved
3:0	Event	Specify the event which should trigger the client interrupt.

Table 10-49. Change Client Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 5h

10.3.9.6.3.3.7 Send Interrupt

A Send Interrupt descriptor will cause the VPDMA to generate an interrupt on the list manager controlled interrupts as specified by the Source Field. The list will not stall on this descriptor. For example, if source is 0 then control_descriptor_int0 will fire. If source is 12, then control_descriptor_int12 will fire. For more information ot VPDMA interrupt events, see Section 10.3.9.5, *VPDMA Interrupts*.

Table 10-50. Send Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 6h

10.3.9.6.3.3.8 Reload List

A Reload List descriptor causes ending descriptors after this descriptor in the original list to be dropped and a new list at the location and of the size specified in the descriptor. This descriptor can be used to allow for linked lists in the VPDMA as the list will continue from this point and fetch the list specified and it does not have to be continuous with the current list.

Table 10-51. Reload List Field Descriptions (Word - 0)

Bits	Name	Description
31:0	LIST_ADDRESS	31 most-significant Bits of the memory address where the descriptors to be loaded are stored. Address must be 16 byte aligned.

Table 10-52. Reload List Field Descriptions (Word - 1)

Bits	Name	Description
31:16	Reserved	Reserved
15:0	LIST_SIZE	Size of the list to load

Table 10-53. Reload List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 7h

10.3.9.6.3.3.9 Abort Channel

An Abort Channel descriptor is used to clear a channel. This clears the channel from issuing any more requests. Any outstanding requests for that channel will complete as originally scheduled. All data inside the client will be flushed. For Tiled Clients such as the noise filter it is required to issue two consecutive abort channel descriptors to ensure the channel is aborted for both the current tile and next tile. Word 0, Word 1 and Word 2 are reserved.

Table 10-54. Abort Channel Field Descriptions (Word - 3)

Bit	Name	Description
31:27	Packet Type	Host Packet Descriptor type = 0xC
26:24	Reserved	Reserved
23:16	Source	VPDMA Channel Number whose transfers are to be aborted
15:4	Reserved	Reserved
3:0	Control	Control type = 9h

10.3.9.7 VPDMA Configuration

The following section describes the different ways of configuring VPDMA for data transfers.

10.3.9.7.1 Regular List

A regular list executes each descriptor in order until the end of the list is reached. When the end of the list is reached an interrupt is sent and the list can be reused by software. A regular list can contain any descriptor types. Software creates the list at some location in external memory. After completing writing the list software then writes the location of the list to the VPE_LIST_ADDR[31:0] VPE_LIST_ADDR register and then writes the LIST_ATTR register. If the NUMBER in the VPE_LIST_ATTR[26:24] LIST_NUM is not an active list then the List will be loaded and begin to execute the next time the List Manager gets to IDLE after processing previous loaded lists. If the NUMBER in the VPE_LIST_ATTR[26:24] LIST_NUM is busy then the VPE_LIST_ADDR[31:0] LIST_ADDR and VPE_LIST_ATTR registers will be locked until the active list specified by NUMBER completes.

The different ports inside VPDMA requires different list setup, as explained in the following sections.

10.3.9.7.2 Video Input Ports

The Video Input Ports can be used in multiple ways. Depending on how the input ports are configured will determine how the lists to manage them need to be setup. The ways in which the ports can work are multiplexed data stream, single YUV color separate stream, dual YUV interleaved or single RGB stream. Each port also has an ancillary data port that can run either multiplexed or single data stream and shares the buffering with the main data stream. For all cases the descriptor must be loaded into the client before the vertical sync is received on the VIP port or the entire frame will be dropped. As with all write clients the VIP channels can be shadowed so the next frame/field descriptor can be loaded while the previous frame is running without stalling the list.

10.3.9.7.2.1 Single YUV Color Separate

If the port is configured to be used as sending out color separated YUV data then the channels that must be used are VIP1_PORTA_LUMA and VIP1_PORTA_CHROMA for the luma and chroma components respectively. The data type can be either 420 or 422 color separate in this case.

10.3.9.7.2.2 Dual YUV Interleaved

If the port is configured to be used to send out 422 interleaved data from a non-multiplexed source then the channels that must be used are VIP1_PORTA_LUMA or VIP1_PORTA_CHROMA depending on if the data stream is connected to the luma or chroma port. The data type must be 422 interleaved in this case.

10.3.9.7.2.3 Single RGB Stream

If the port is configured to be used to send out RGB stream then the channel VIP1_PORTA_RGB must be used. The incoming data is then assumed to be RGB 888 data and this is combined with the Background Color Alpha register field to make ARGB8888 data that will then be sent out based on the data type. If the data type uses less then the full 8 bits the lower bits are dropped and just the upper bits are sent to get the correct data format.

10.3.9.8 VPDMA Data Formats

Each of the channels has a type of data that it can support based upon the client that it services. Also for each channel, the data type will assume that data is packed in memory a certain way and will be prevented to the client in the same manner to the client no matter what the format of the data in memory.

10.3.9.8.1 YUV Data Formats

The YUV formatted channels expect video data that is in YUV color space. The YUV data can be type is for YUV data and it can support both interleaved data where Luma and Chroma are in the same data buffer or it can support co-planar data where the Luma and Chroma are in separate data buffers. The YUV channel also can be given a position to give a different start position for the client instead of the default upper left hand corner of the frame. The storage format for YUV data depends on the data type field. The data type must be set to a data type that the channel can support. All the clients that data are provided too will either accept Luma Only, Chroma Only or Luma and Chroma in a parallel data bus.

10.3.9.8.1.1 Y 4:4:4 (Data Type 0)

The Y 4:4:4 data type is used for a co-planar 4:4:4 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and height set to the desired frame size expected by the receiving client.

Figure 10-51. Y 4:4:4 (Data Type 0)

10.3.9.8.1.2 Y 4:2:2 (Data Type 1)

The Y 4:2:2 data type is used for a co-planar 4:2:2 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the desired frame sent by the VPDMA to the receive client.

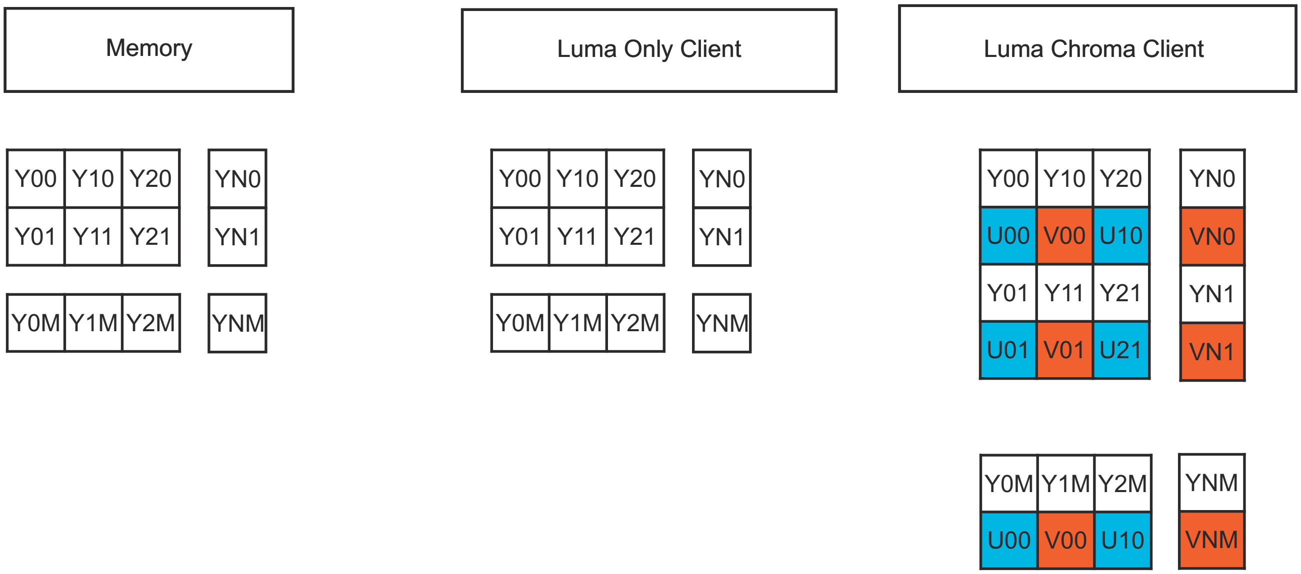


Figure 10-52. Y 4:2:2 (Data Type 1)

10.3.9.8.1.3 Y 4:2:0 (Data Type 2)

The Y 4:2:0 data type is used for a co-planar 4:2:0 data type. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the expected frame for the client.

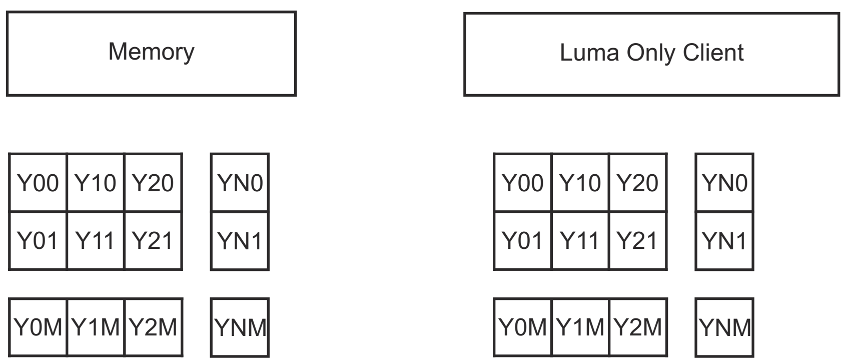


Figure 10-53. Y 4:2:0 (Data Type 2)

10.3.9.8.1.4 C 4:4:4 (Data Type 4)

The C 4:4:4 data type is used for a co-planar 4:4:4 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and height of the expected client frame.

Figure 10-54. C 4:4:4 (Data Type 4)

10.3.9.8.1.5 C 4:2:2 (Data Type 5)

The C 4:2:2 data type is used for co-planar 4:2:2 data.It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and the height of the expected client frame.

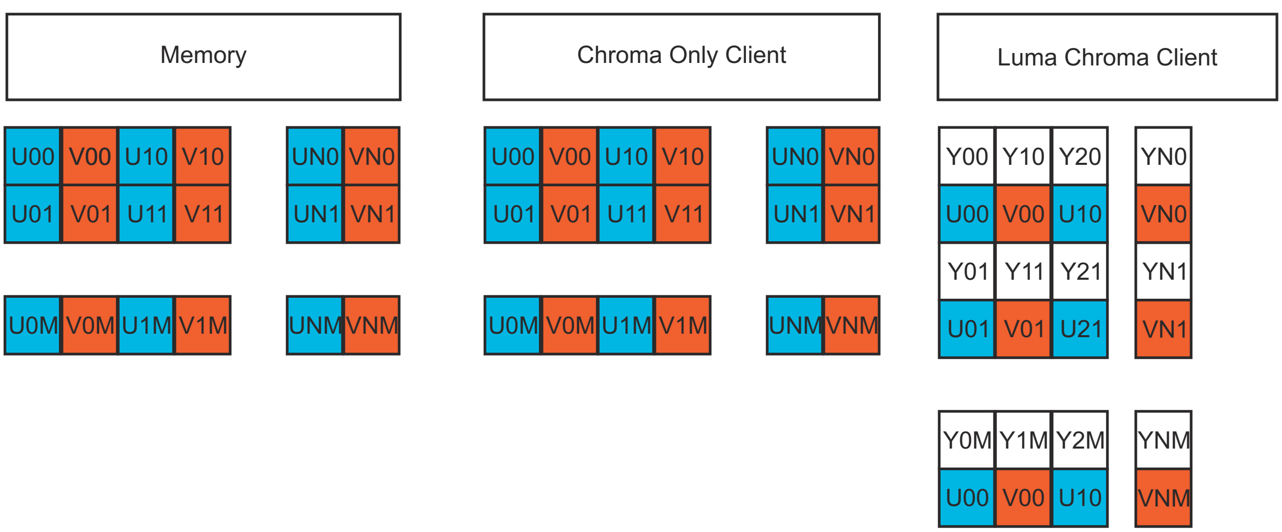


Figure 10-55. C 4:2:2 (Data Type 5)

10.3.9.8.1.6 C 4:2:0 (Data Type 6)

The C 4:2:0 data type is used for a co-planar 4:2:0 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in a 16 bit container. This data block should have the width and half the height of the expected clients frame.

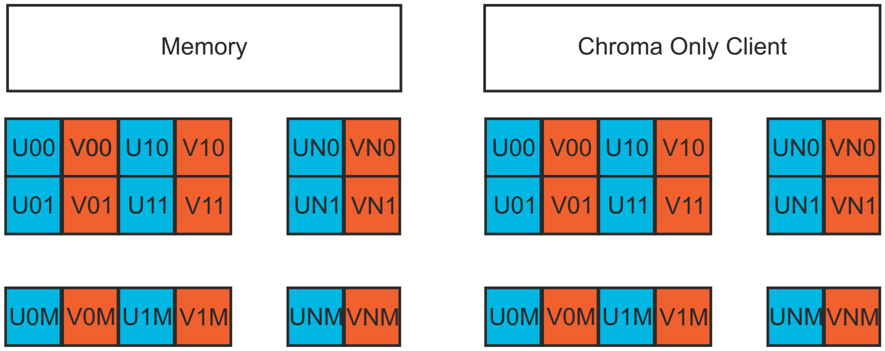


Figure 10-56. C 4:2:0 (Data Type 6)

10.3.9.8.1.7 YC 4:2:2 (Data Type 7)

The YC 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Y1 finally Cr in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

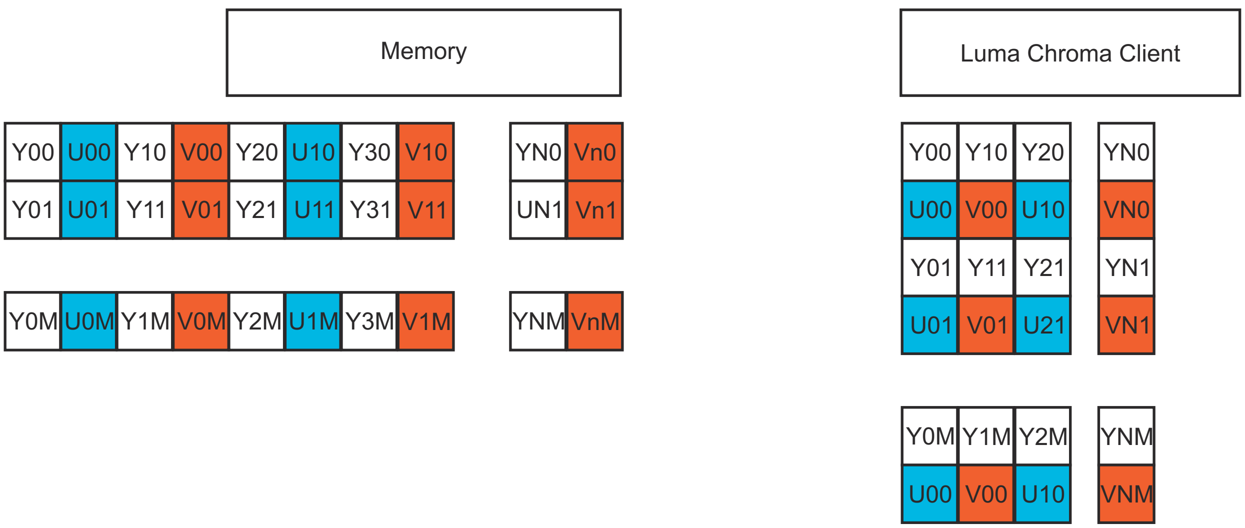


Figure 10-57. YC 4:2:2 (Data Type 7)

10.3.9.8.1.8 YC 4:4:4 (Data Type 8)

The YC 4:4:4 data type is used for interleaved 4:4:4 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Cr. The transfer counts each YCbCr triplet in a 24 bit word as a pixel. A 2D transfer of the data is NOT supported in the VPDMA. The data block width and height should be set to the number of pixels and lines that are expected by the client.

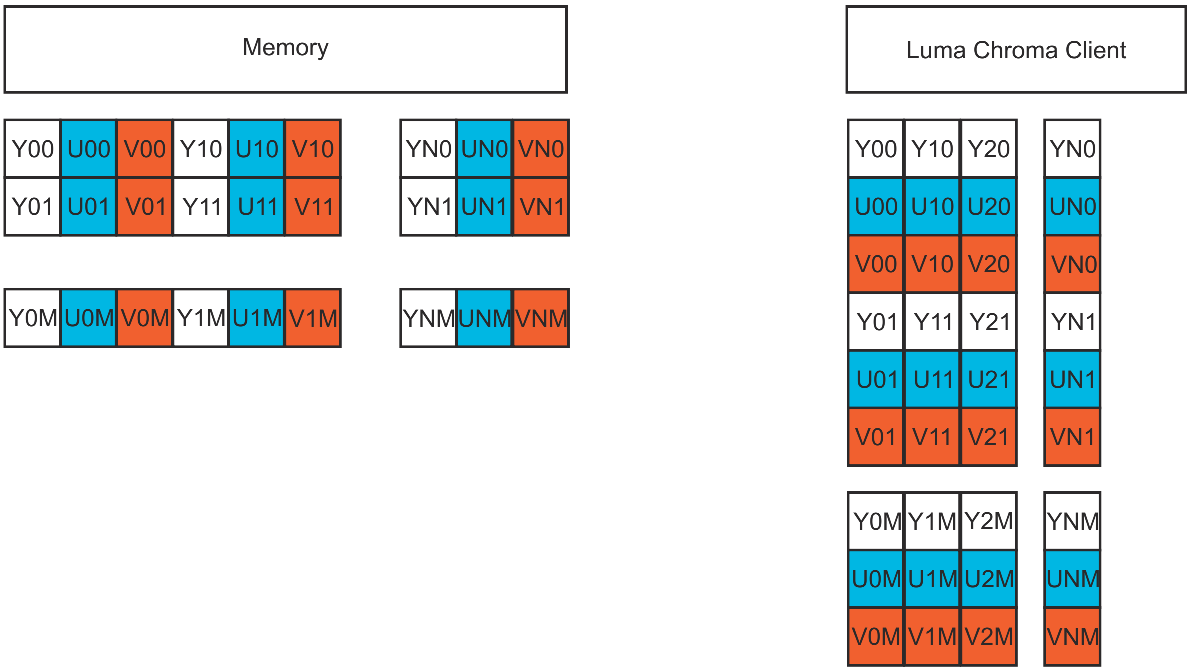


Figure 10-58. YC 4:4:4 (Data Type 8)

10.3.9.8.1.9 CY 4:2:2 (Data Type 23)

The CY 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Cb in the lowest byte followed by Y0 followed by Cr finally Y1 in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

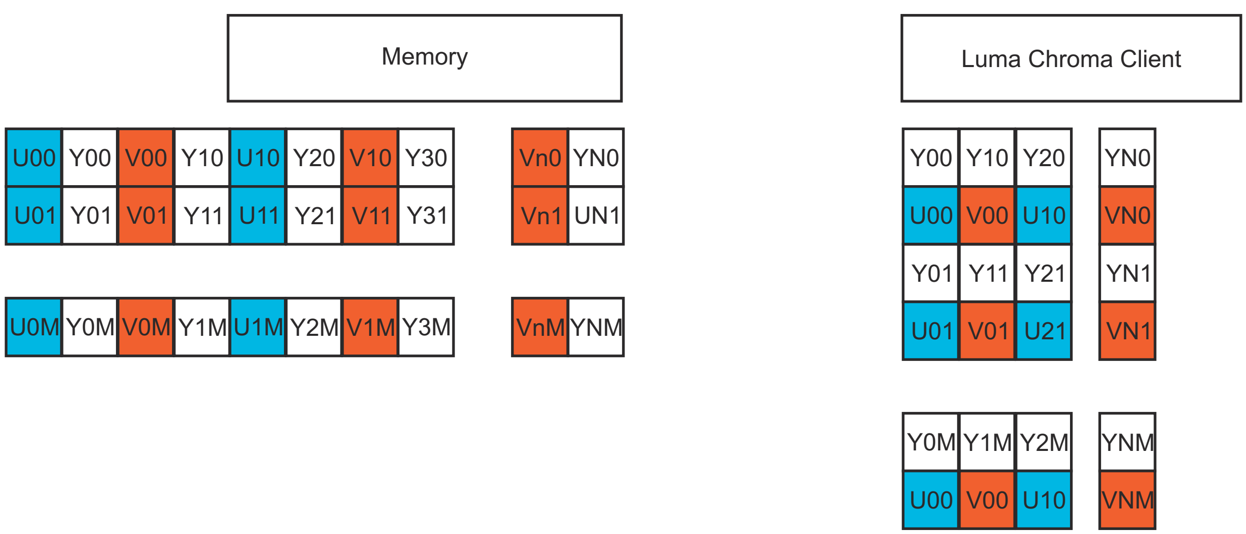


Figure 10-59. CY 4:2:2 (Data Type 23h)

10.3.9.8.2 RGB Data Formats

The RGB channel type is used to provide data for a client that expects to transmit RGB data. In all modes the client is always RGBA 8888 data. The lower bits, if not provided by the data stream, are a replication of the lower bit of the data. For outbound data the input is assumed to be RGB 888 and the Alpha value is taken from the Background Color register to make a full ARGB 8888. The lower bits are dropped from this data, if a data type specifies less then the full 8 bits per color. The client has individual data buses for each component so they have no order dependency in the data bus.

10.3.9.8.2.1 Input Data Formats

10.3.9.8.2.1.1 RGB16-565 (Data Type 0)

In RGB16-565 mode, each pixel is a single RGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the middle 6 bits for green data and the upper 5 bits for red data.

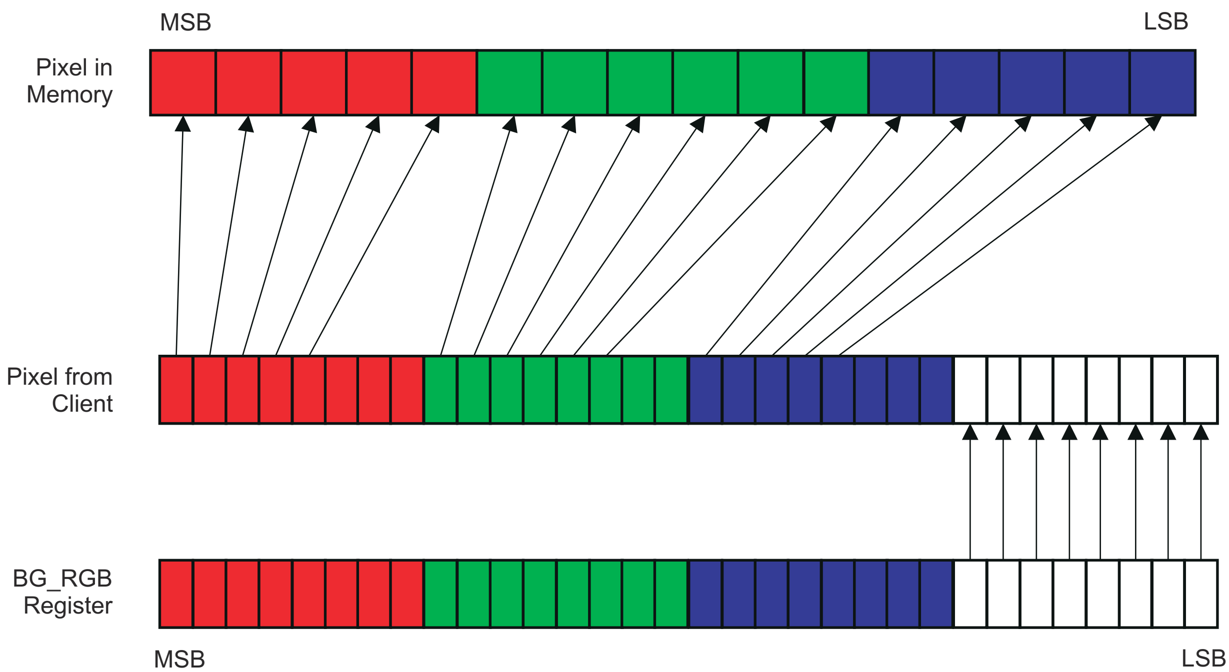


Figure 10-60. RGB16-565 (Data Type 0)

10.3.9.8.2.1.2 ARGB-1555 (Data Type 1)

In ARGB-1555 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data and the upper most bit for the blend value to use 0 or 0xFF.

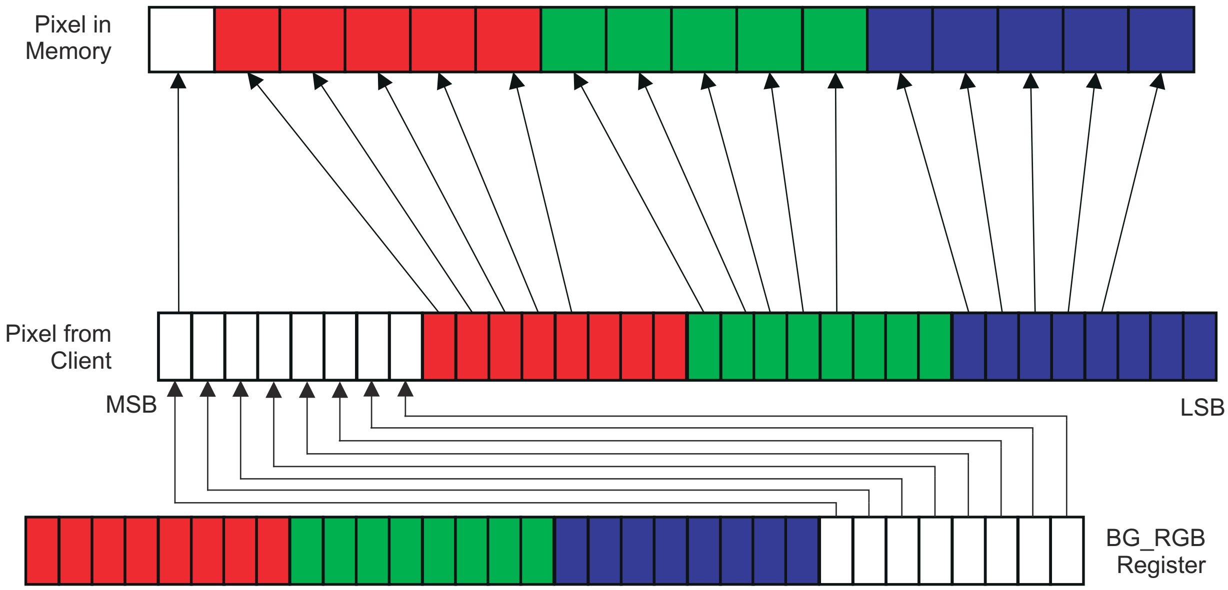


Figure 10-61. ARGB-1555 (Data Type 1)

10.3.9.8.2.1.3 ARGB-4444 (Data Type 2)

In ARGB16-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for blue data, the next 4 bits for green data the next 4 bits for red data and the upper most 4 bits for the blend value.

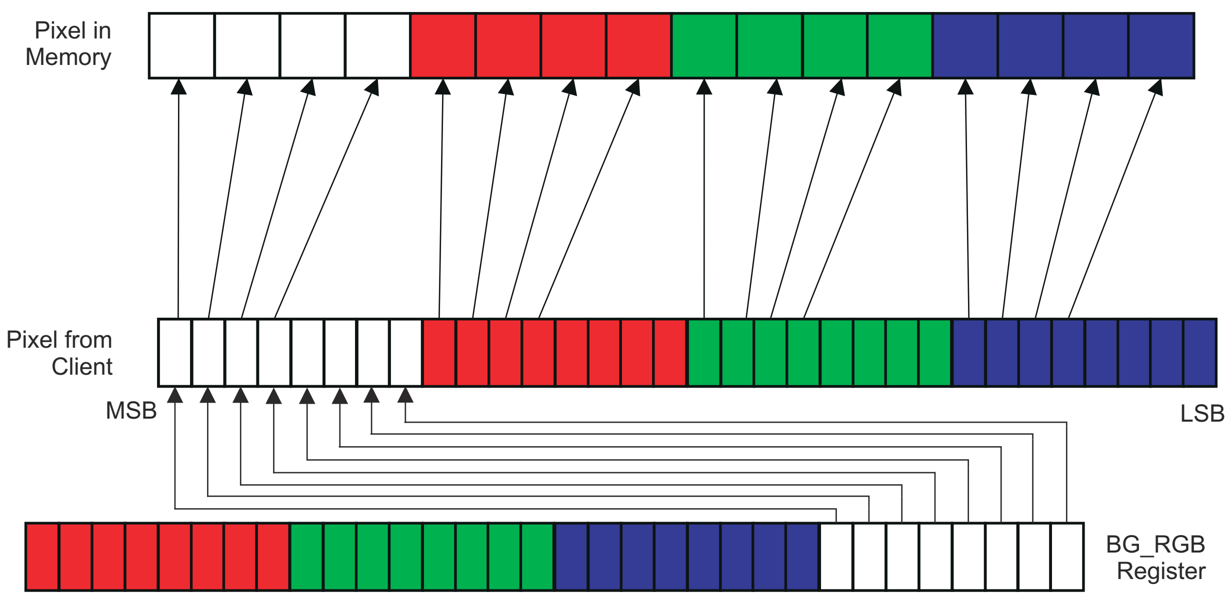


Figure 10-62. ARGB-4444 (Data Type 2)

10.3.9.8.2.1.4 RGBA-5551 (Data Type 3)

In RGBA-5551 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest most bit for the blend value to use 0 or 0xff the next 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data.

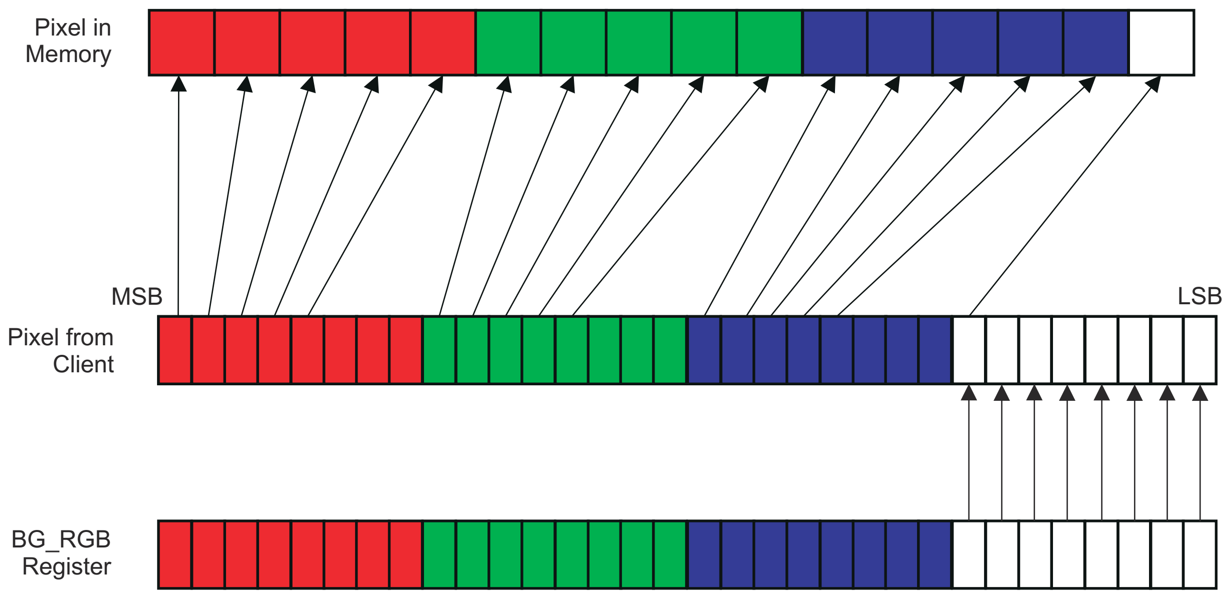


Figure 10-63. RGBA-5551 (Data Type 3)

10.3.9.8.2.1.5 RGBA-4444 (Data Type 4)

In RGBA-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for the blend value, the next 4 bits for blue data, the next 4 bits for green data and the upper most 4 bits for red data.

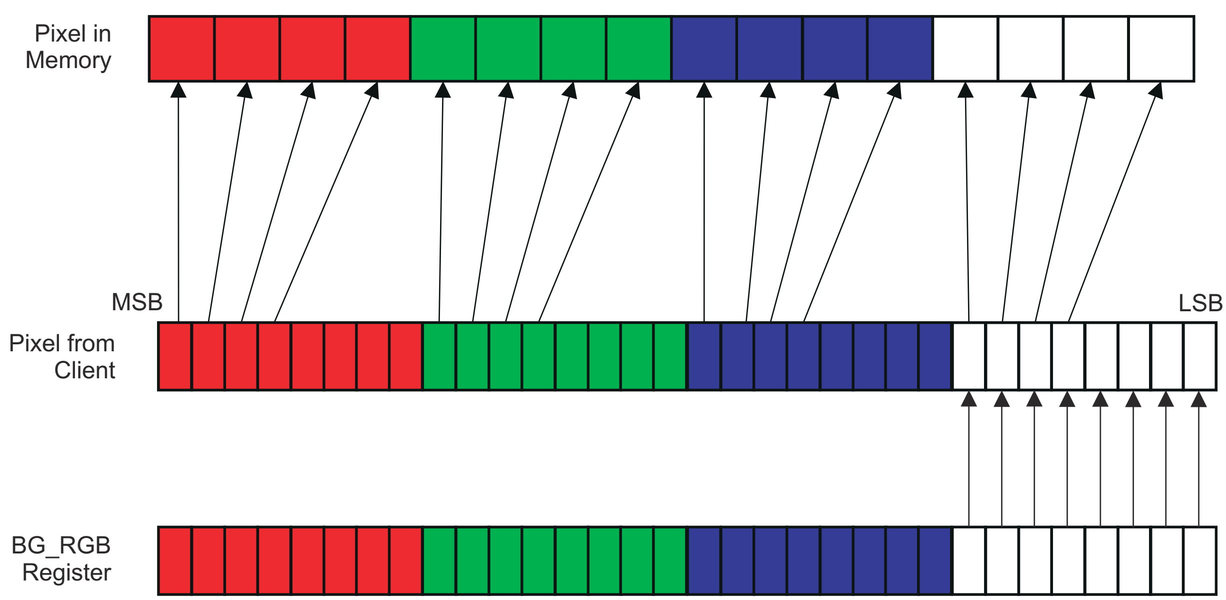


Figure 10-64. RGBA-4444 (Data Type 4)

10.3.9.8.2.1.6 ARGB24-6666 (Data Type 5)

In ARGB24-6666 mode, each pixel is a single ARGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper most 6 bits for red data.

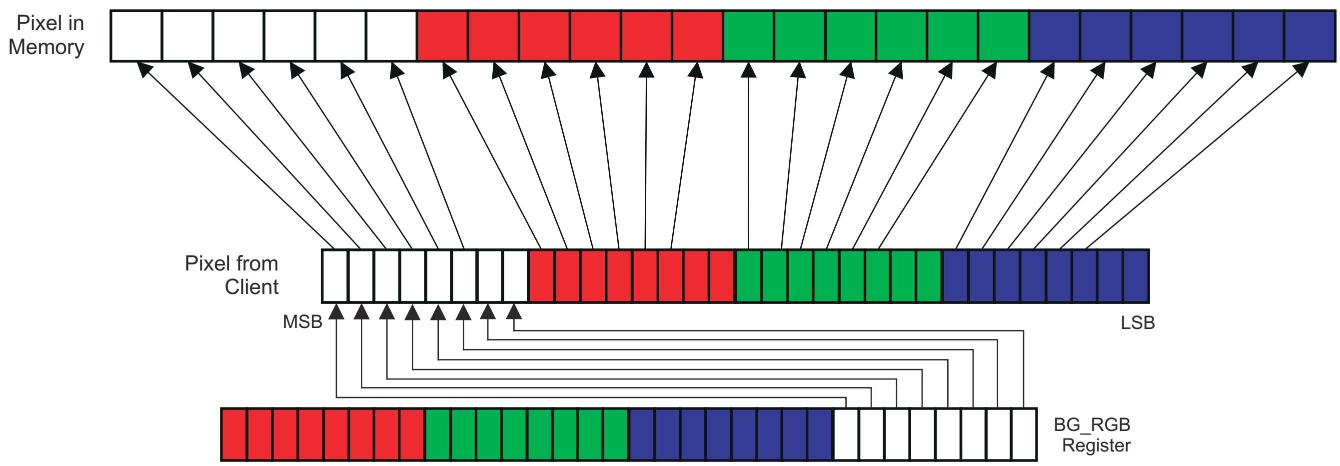


Figure 10-65. ARGB24-6666 (Data Type 5)

10.3.9.8.2.1.7 RGB24-888 (Data Type 6)

In RGB24-888 mode, each pixel is a single RGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data it uses the VPE_BG_RGB Blend value for the Blend value.

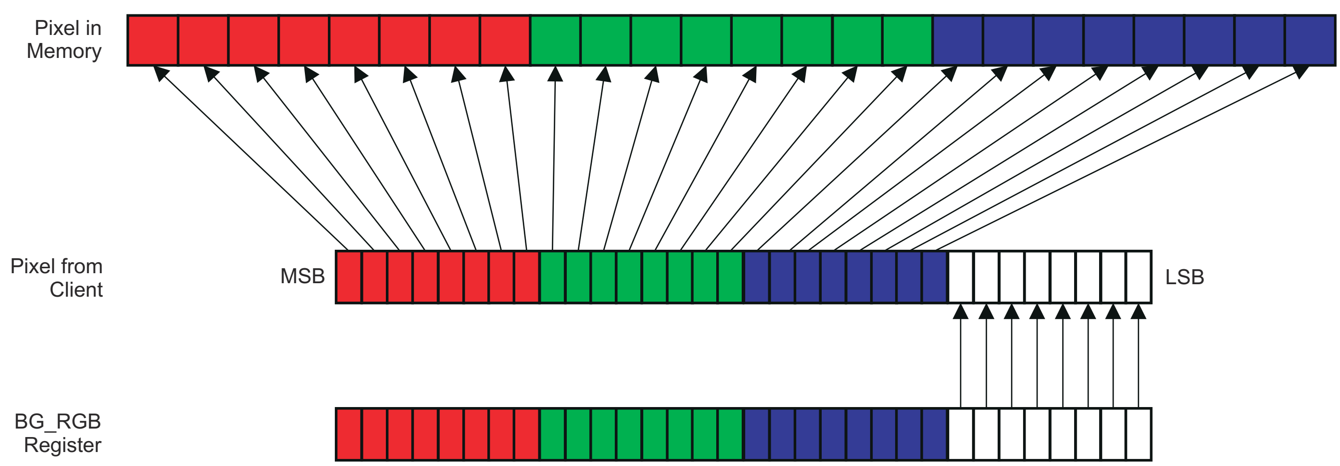


Figure 10-66. RGB24-888 (Data Type 6)

10.3.9.8.2.1.8 ARGB32-8888 (Data Type 7)

In ARGB32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bitsof data uses the lowest 8 bits for blue data, the next 8 bits for green data and the next 8 bits for red data it uses the upper most bits for the blend value.

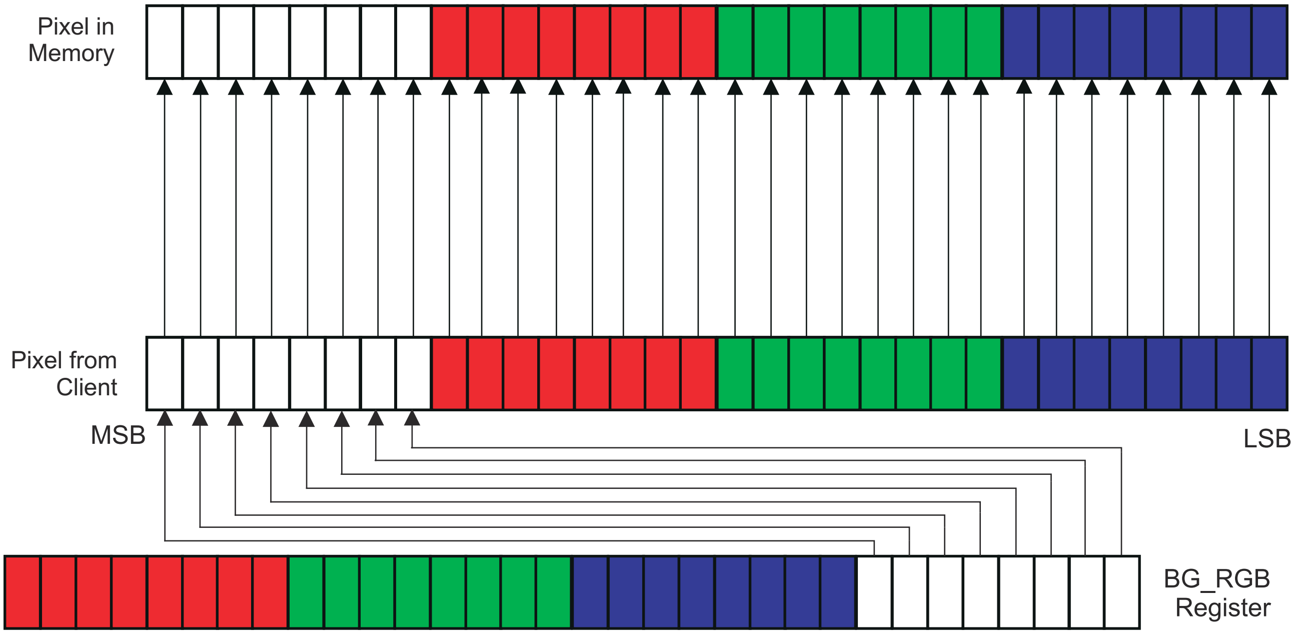


Figure 10-67. ARGB32-8888 (Data Type 7)

10.3.9.8.2.1.9 RGBA24-6666 (Data Type 8)

In RGBA24-6666 mode, each pixel is a single RGBA pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper 6 bits for red data.

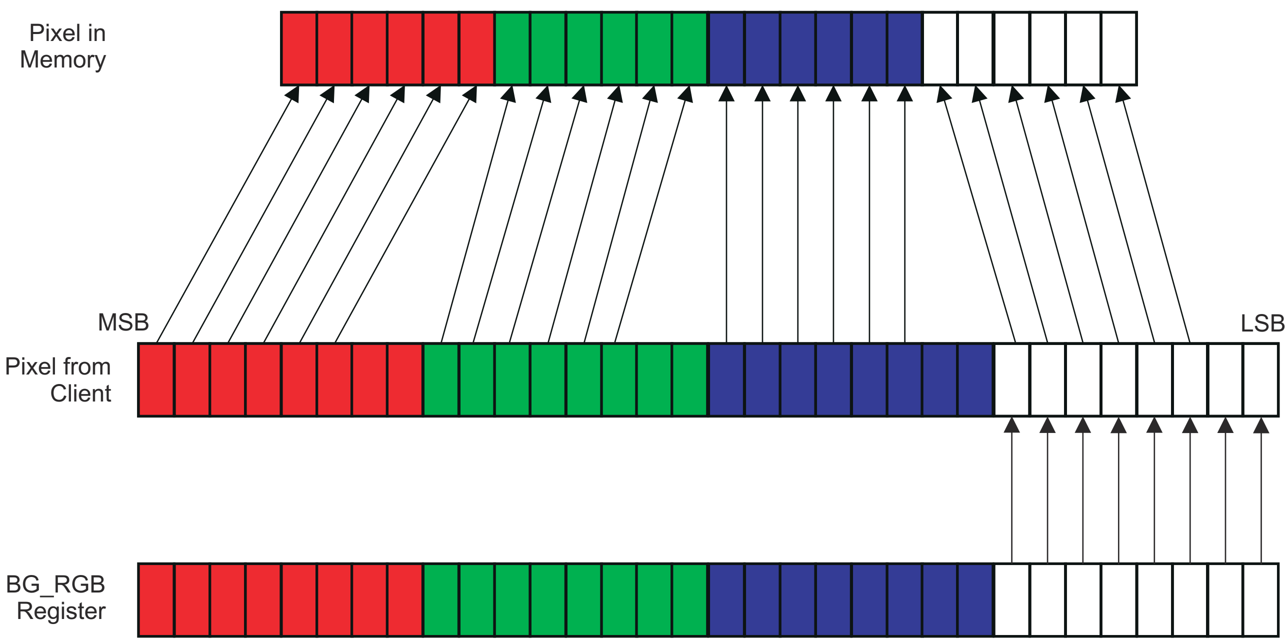


Figure 10-68. RGBA24-6666 (Data Type 8)

10.3.9.8.2.1.10 RGBA32-8888 (Data Type 9)

In RGBA32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bits of data uses the lowest 8 bits for the blend value the next 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data.

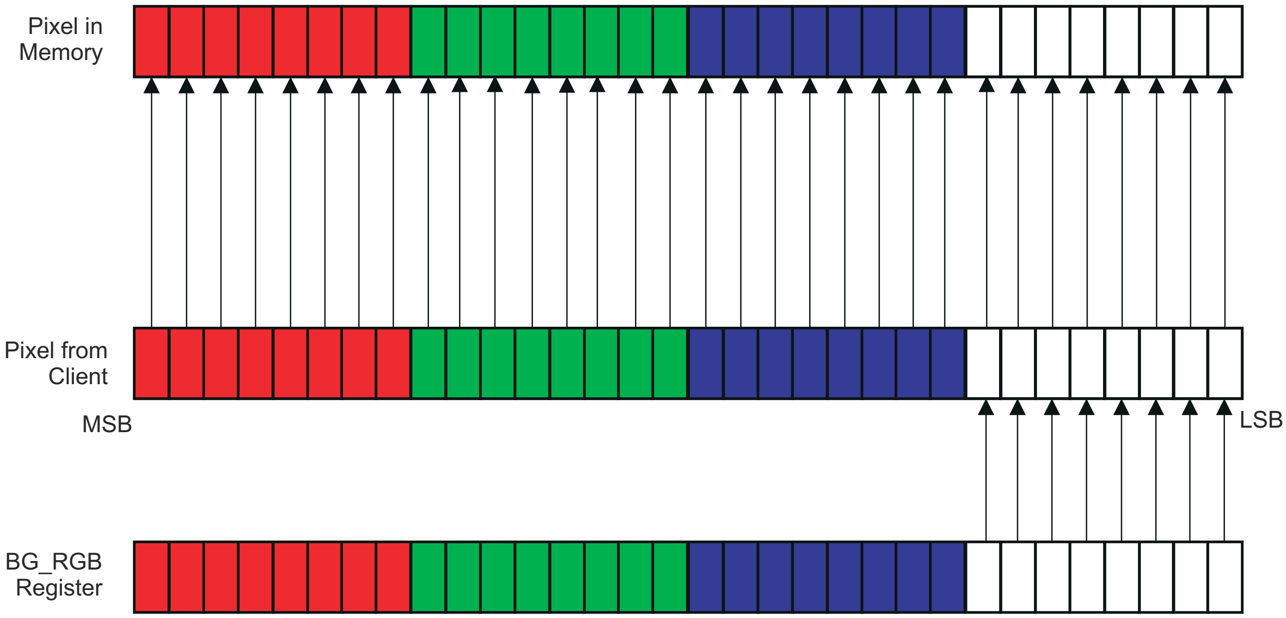


Figure 10-69. RGBA32-8888 (Data Type 9)

10.3.9.8.2.2 Output Data Formats

10.3.9.8.2.2.1 RGB16-565 (Data Type 0)

In RGB16-565 mode, each pixel is a single RGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the middle 6 bits for green data and the upper 5 bits for red data.

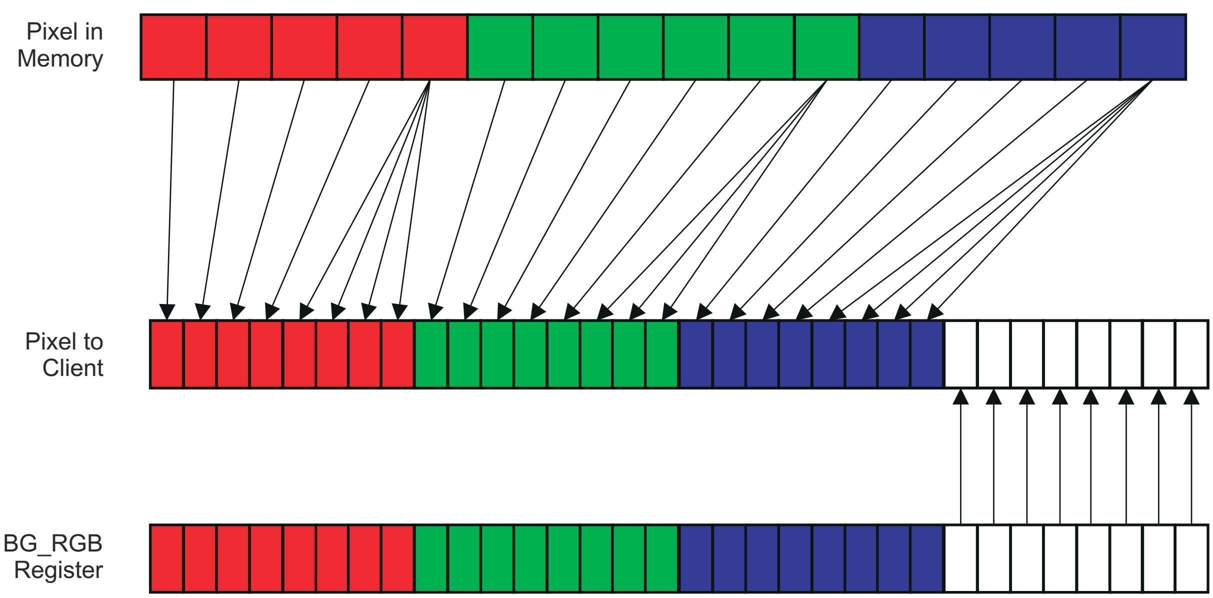


Figure 10-70. RGB16-565 (Data Type 0)

10.3.9.8.2.2.2 ARGB-1555 (Data Type 1)

In ARGB-1555 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data and the upper most bit for the blend value to use 0 or 0xFF.

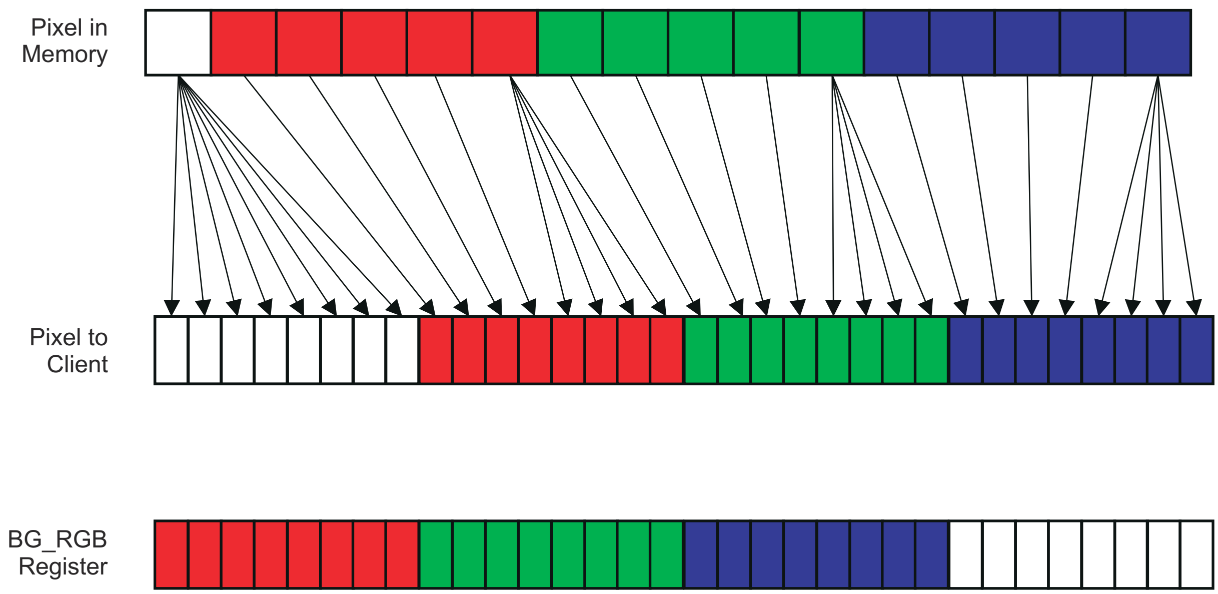


Figure 10-71. ARGB-1555 (Data Type 1)

10.3.9.8.2.2.3 ARGB-4444 (Data Type 2)

In ARGB16-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for blue data, the next 4 bits for green data the next 4 bits for red data and the upper most 4 bits for the blend value.

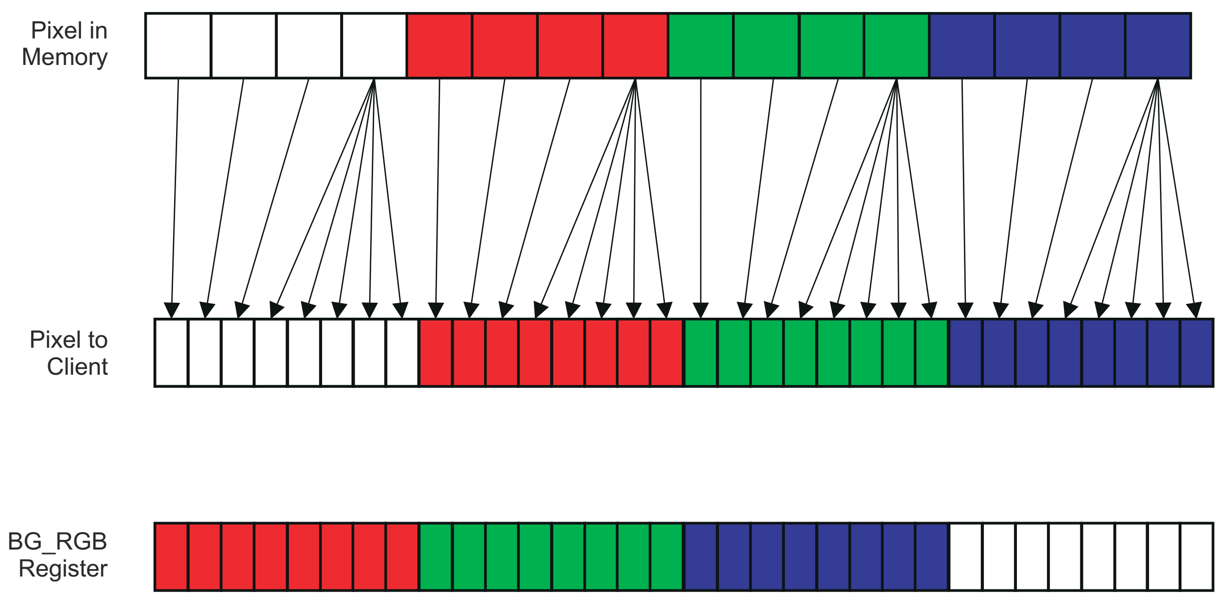


Figure 10-72. ARGB-4444 (Data Type 2)

10.3.9.8.2.2.4 RGBA-5551 (Data Type 3)

In RGBA-5551 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest most bit for the blend value to use 0 or 0xff the next 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data.

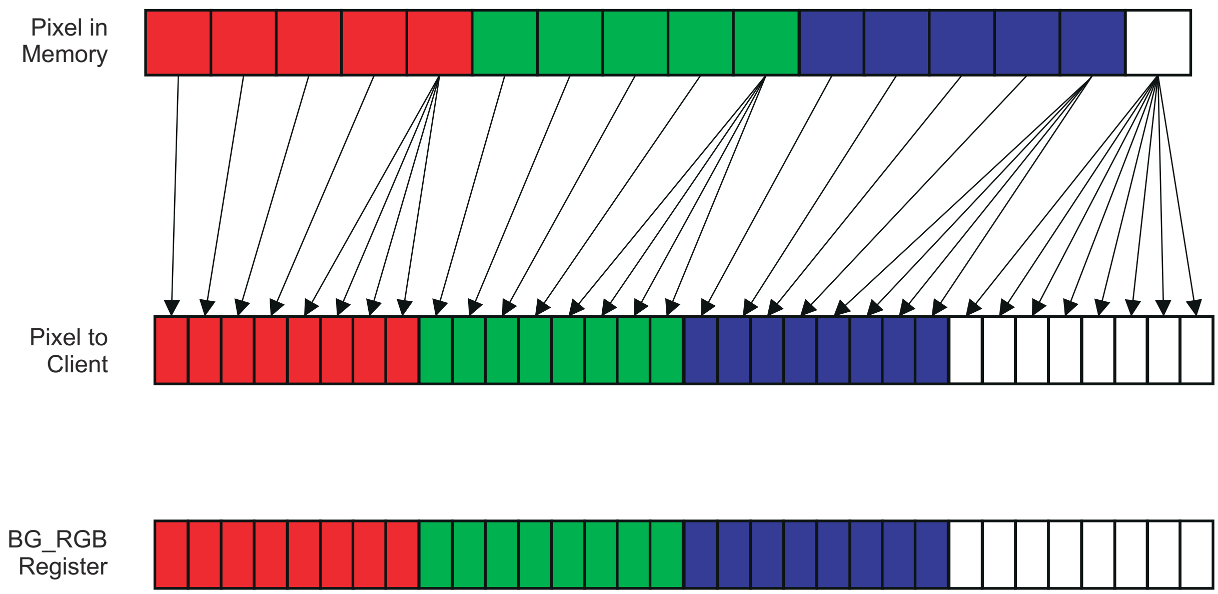


Figure 10-73. RGBA-5551 (Data Type 3)

10.3.9.8.2.2.5 RGBA-4444 (Data Type 4)

In RGBA-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for the blend value, the next 4 bits for blue data, the next 4 bits for green data and the upper most 4 bits for red data.

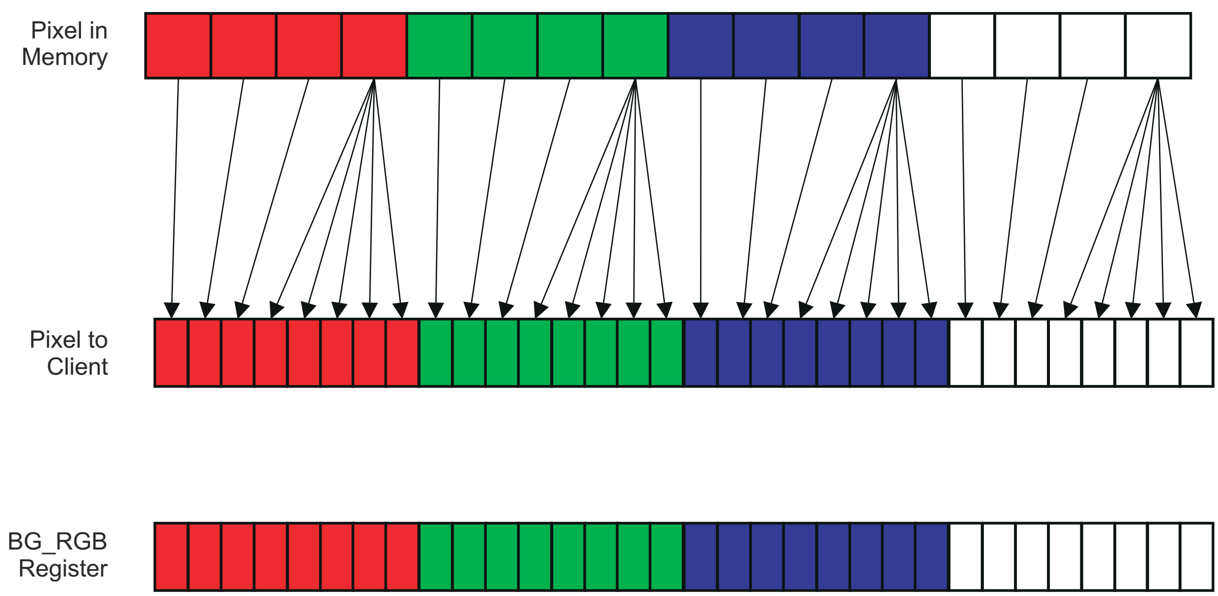


Figure 10-74. RGBA-4444 (Data Type 4)

10.3.9.8.2.2.6 ARGB24-6666 (Data Type 5)

In ARGB24-6666 mode, each pixel is a single ARGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper most 6 bits for red data.

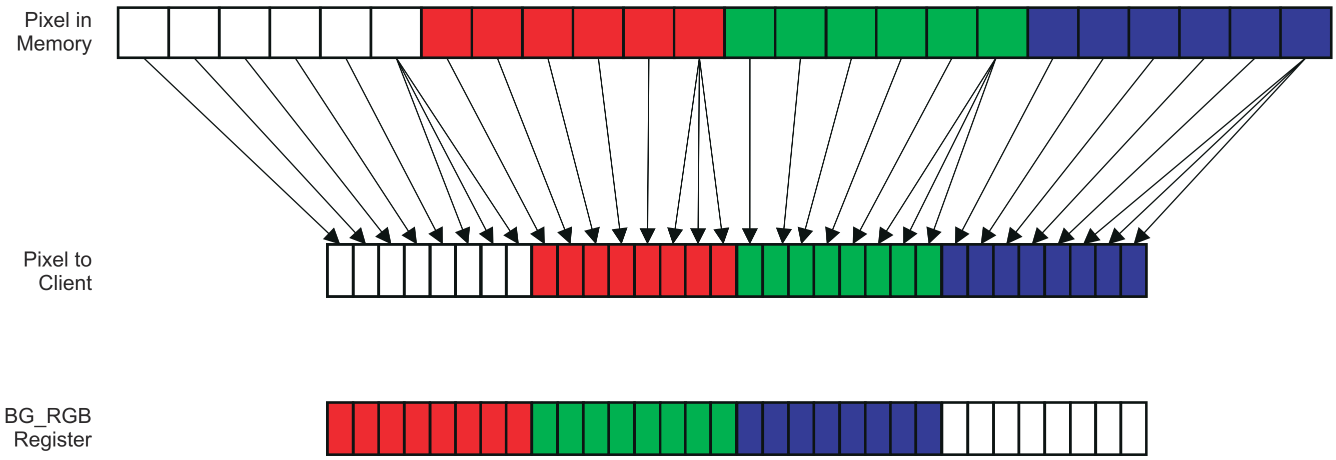


Figure 10-75. ARGB24-6666 (Data Type 5)

10.3.9.8.2.2.7 RGB24-888 (Data Type 6)

In RGB24-888 mode, each pixel is a single RGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data it uses the VPE_BG_RGB Blend value for the Blend value.

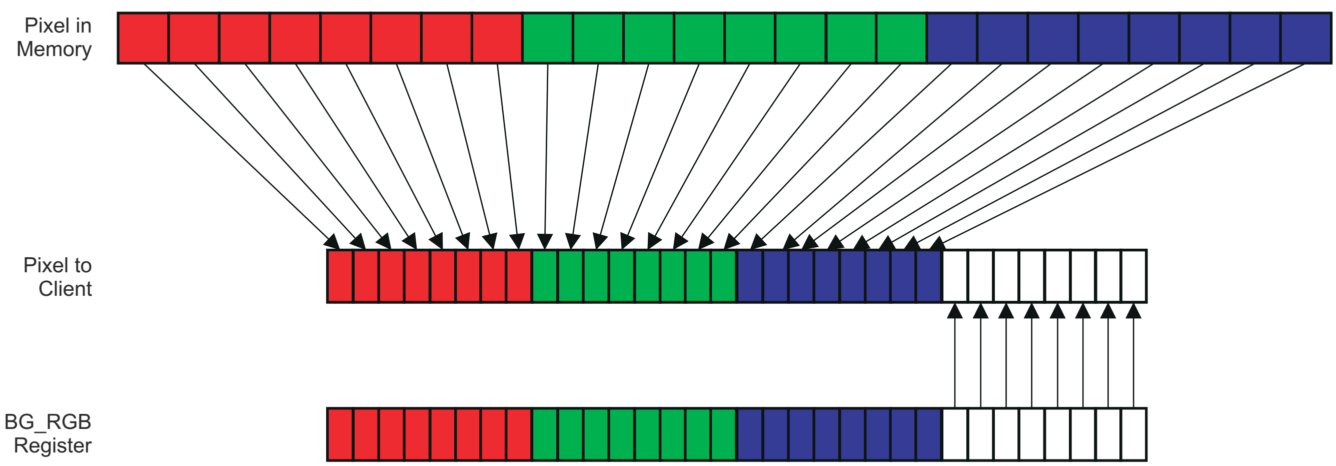


Figure 10-76. RGB24-888 (Data Type 6)

10.3.9.8.2.2.8 ARGB32-8888 (Data Type 7)

In ARGB32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bitsof data uses the lowest 8 bits for blue data, the next 8 bits for green data and the next 8 bits for red data it uses the upper most bits for the blend value.

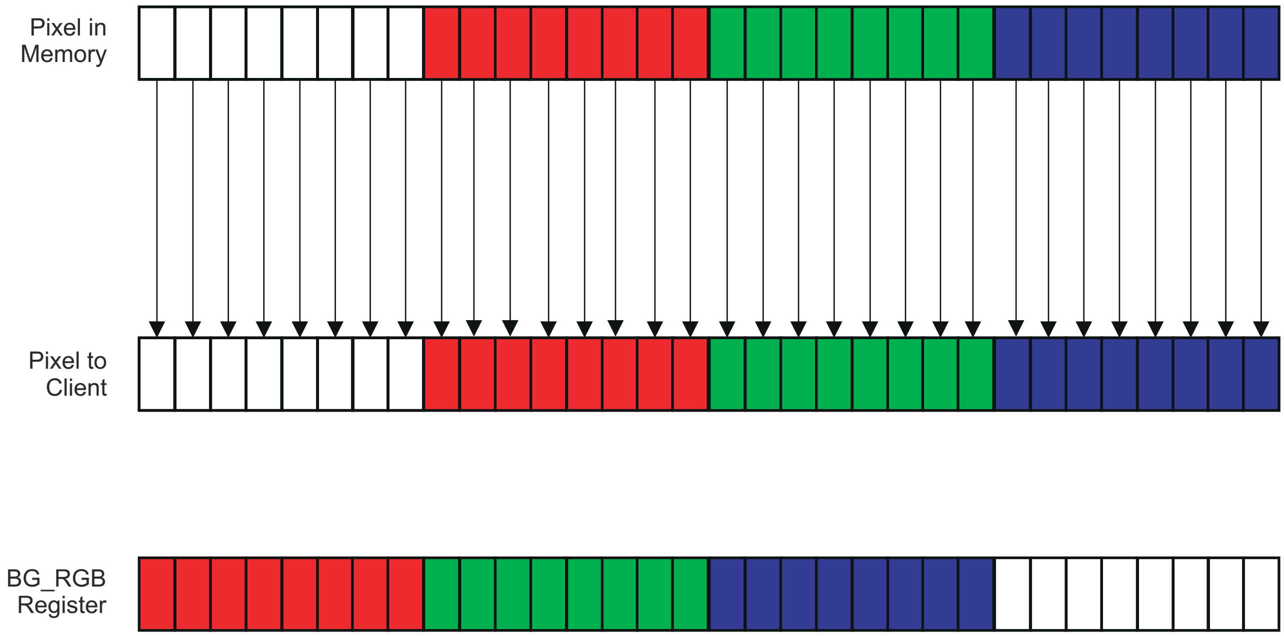


Figure 10-77. ARGB32-8888 (Data Type 7)

10.3.9.8.2.2.9 RGBA24-6666 (Data Type 8)

In RGBA24-6666 mode, each pixel is a single RGBA pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper 6 bits for red data.

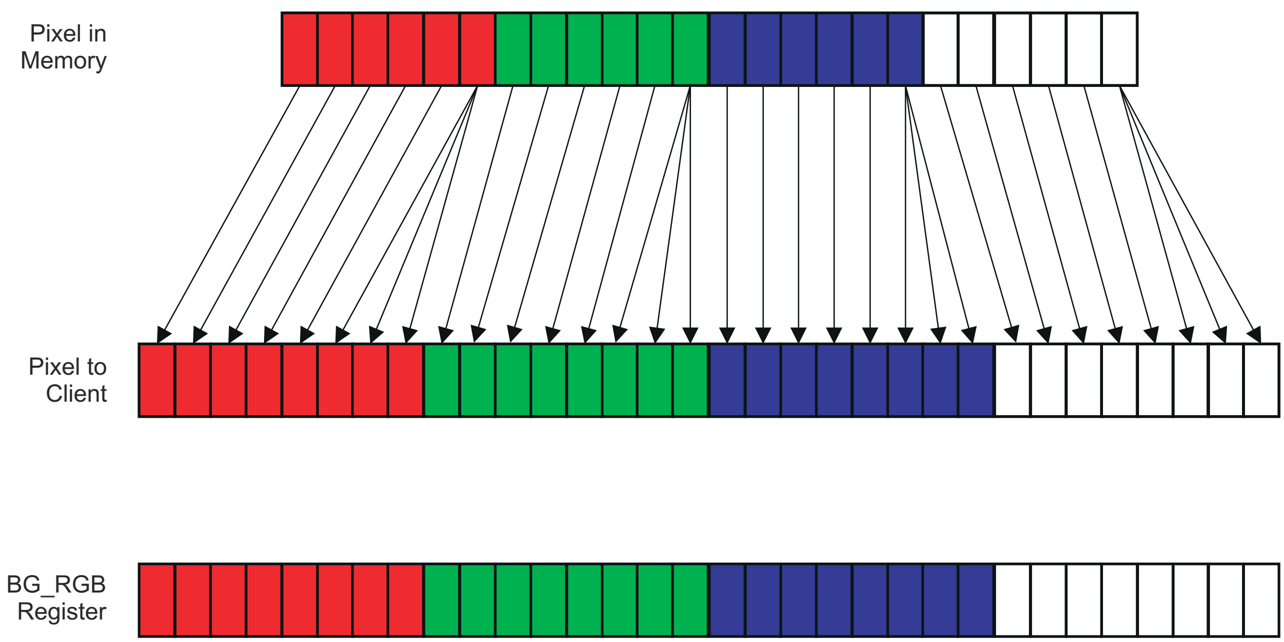


Figure 10-78. RGBA24-6666 (Data Type 8)

10.3.9.8.2.2.10 RGBA32-8888 (Data Type 9)

In RGBA32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bits of data uses the lowest 8 bits for the blend value the next 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data.

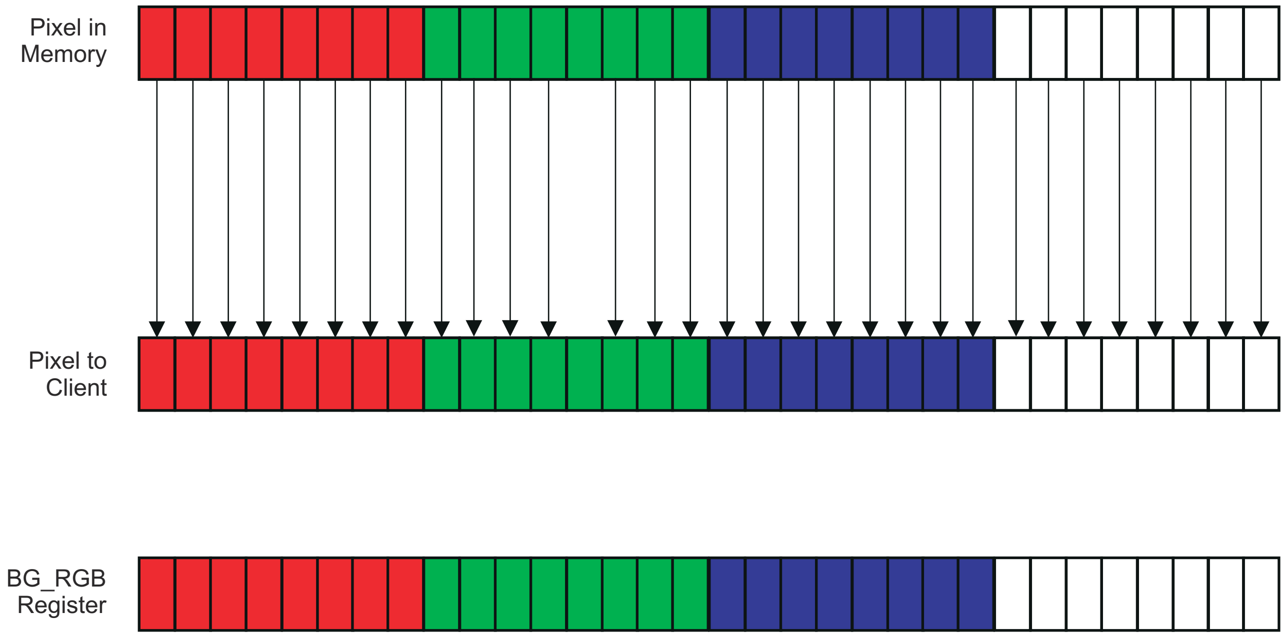


Figure 10-79. RGBA32-8888 (Data Type 9)

10.3.9.8.3 Miscellaneous Data Type

The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client. The size of the miscellaneous data type is always determined by the Data Type field of the descriptor which specifies the number of bits in the descriptor.

A memory structure used to describe a desired memory transaction to or from a client. The descriptor at a minimum gives an address location for the memory portion of the transfer, the channel to use for this transaction and the size of the transaction. The data descriptor can also contain attributes to be passed down to the client or be linked to another data descriptor to form a larger frame from many smaller frames.

10.3.10 VPE Software Reset

Software reset in the VPE module can be done by setting the VPE_CLKC_RST[1] PRIM_DP_RST and VPE_CLKC_RST[0] VPDMA_RST for VPE VPDMA to 0x1. Software must ensure that the software reset completes before performing operations within the VPE module.

10.3.11 VPE Power and Clocks Management

The VPE modules support the MStandby/Wait and IdleReq/SidleAck protocols as defined in *Power, Reset, and Clock Management.*

Power Management within the VPE module can be accomplished in several ways:

- L4 MConnect/SConnect can disable the internal L4 clock network
- L3 MConnect/Sconnect can disable the internal L3 clock network

These items are accomplished using the standard slave idle (for L4) and master standby (for L3) protocols. When these modules are instructed to disable clocks for the internal L3 or L4 (MMR) clock domains, the internal clock networks will be shut down. This shut down applies to the clock signals - L3_CLK and L4_CLK.

10.3.11.1 VPE Clocks

The VPE internal clock domains can only be shut down by writing the appropriate register bit within the Clock Enable register - VPE_CLKC_CLKEN[1] PRIM_DP_EN and VPE_CLKC_CLKEN[0] VPDMA_EN for the VPDMA engine

10.3.11.2 VPE Idle Mode

The VPE supports no-idle mode, force-idle mode, and smart-idle mode. The mode can be selected byprogramming the appropriate value in the VPE_SYSCONFIG[3:2] IDLEMODE bit field.

Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.

- Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements.
- No-idle mode: local target never enters idle state.
- Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA related requests) wakeup events

10.3.11.3 VPE StandBy Mode

The VPE supports no-standby mode, force-standby mode, and a single smart-standby mode. The mode is set in the VPE_SYSCONFIG[5:4] STANDBYMODE bit field.

Configuration of the local initiator state management mode:

- Force-standby mode: local initiator is unconditionally placed in standby state.
- No-standby mode: local initiator is unconditionally placed out of standby state.

10.4 VPE Register Manual

10.4.1 VPE Instance Summary

Table 10-55. VPE Instance Summary

Module Name	Module Base Address	Size
VPE_TOP_LEVEL	0x489D 0000	288 Bytes
VPE_CHR_US_INST_0	0x489D 0300	36 Bytes
VPE_CHR_US_INST_1	0x489D 0400	36 Bytes
VPE_CHR_US_INST_2	0x489D 0500	36 Bytes
VPE_DEI	0x489D 0600	60 Bytes
VPE_SC	0x489D 0700	128 Bytes
VPE_CSC	0x489D 5700	24 Bytes
VPE_VPDMA	0x489D D000	980 Bytes

10.4.2 VPE_CSC Registers

10.4.2.1 VPE_CSC Register Summary

Table 10-56. VPE_CSC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_CSC Base Address
VPE_CSC00	RW	32	0x0000 0000	0x489D 5700
VPE_CSC01	RW	32	0x0000 0004	0x489D 5704
VPE_CSC02	RW	32	0x0000 0008	0x489D 5708
VPE_CSC03	RW	32	0x0000 000C	0x489D 570C
VPE_CSC04	RW	32	0x0000 0010	0x489D 5710
VPE_CSC05	RW	32	0x0000 0014	0x489D 5714

10.4.2.2 VPE_CSC Register Description

Table 10-57. VPE_CSC00

Address Offset	0x0000 0000		
Physical Address	0x489D 5700	**Instance**	VPE_CSC
Description			

Table 10-57. VPE_CSC00 (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			B0													RESERVED			A0												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	B0	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0
12:0	A0	Its is represented as Q3.10 number. So the value ranges from -4 to +4. To convert a decimal number, multiply the number by 1024 and write it in the register in hex format. For example, to program 0.673, 0x2B1 should be written in the register. (int)(0.673 X 1024) = (int)689.152 = 689 = 0x2B1. If the real number is negative, then multiply it by 1024, and convert it to 2's compliment format in 12-bit. For example, if a coefficient is - 1.893, 0x186E needs to be written in the register. (int)(-1.893*1024)= -1938 = 0x186E (2'S compliment format of -1938 in 13-bit width)	RW	0x0

Table 10-58. Register Call Summary for Register VPE_CSC00

VPE Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual

- VPE_CSC Register Summary: [10]
- VPE_CSC Register Description: [11] [12] [13] [14] [15] [16] [17] [18]

Table 10-59. VPE_CSC01

Address Offset	0x0000 0004		
Physical Address	0x489D 5704	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			A1													RESERVED			C0												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	A1	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0
12:0	C0	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

Table 10-60. Register Call Summary for Register VPE_CSC01

VPE Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual

- VPE_CSC Register Summary: [10]

Table 10-61. VPE_CSC02

Address Offset	0x0000 0008		
Physical Address	0x489D 5708	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			C1													RESERVED			B1												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	C1	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0
12:0	B1	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

Table 10-62. Register Call Summary for Register VPE_CSC02

VPE Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual

- VPE_CSC Register Summary: [10]

Table 10-63. VPE_CSC03

Address Offset	0x0000 000C		
Physical Address	0x489D 570C	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			B2													RESERVED			A2												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	B2	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
12:0	A2	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

Table 10-64. Register Call Summary for Register VPE_CSC03

VPE Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual

- VPE_CSC Register Summary: [10]

Table 10-65. VPE_CSC04

Address Offset	0x0000 0010		
Physical Address	0x489D 5710	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				D0												RESERVED			C2												

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	D0	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. For example, if this coefficient is 749, then 0x2ED (hex format) should be assigned to this register. Another example, if this coefficient is -1021, then 0xC03 should be assigned to this register.	RW	0x0
15:13	RESERVED		R	0x0
12:0	C2	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

Table 10-66. Register Call Summary for Register VPE_CSC04

VPE Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual

- VPE_CSC Register Summary: [10]
- VPE_CSC Register Description: [11] [12]

Table 10-67. VPE_CSC05

Address Offset	0x0000 0014		
Physical Address	0x489D 5714	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	BYPASS	D2	RESERVED	D1

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	BYPASS	Full CSC bypass mode	RW	0x0
27:16	D2	Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in VPE_CSC04)	RW	0x0
15:12	RESERVED		R	0x0
11:0	D1	Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in VPE_CSC04)	RW	0x0

Table 10-68. Register Call Summary for Register VPE_CSC05

VPE Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
- CSC Bypass Mode: [10]

VPE Register Manual

- VPE_CSC Register Summary: [11]

10.4.3 VPE_SC Registers

10.4.3.1 VPE_SC Register Summary

Table 10-69. VPE_SC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_SC Base Address
VPE_CFG_SC0	RW	32	0x0000 0000	0x489D 0700
VPE_CFG_SC1	RW	32	0x0000 0004	0x489D 0704
VPE_CFG_SC2	RW	32	0x0000 0008	0x489D 0708
VPE_CFG_SC3	RW	32	0x0000 000C	0x489D 070C
VPE_CFG_SC4	RW	32	0x0000 0010	0x489D 0710
VPE_CFG_SC5	RW	32	0x0000 0014	0x489D 0714
VPE_CFG_SC6	RW	32	0x0000 0018	0x489D 0718
RESERVED	R	32	0x0000 001C	0x489D 071C
VPE_CFG_SC8	RW	32	0x0000 0020	0x489D 0720
VPE_CFG_SC9	RW	32	0x0000 0024	0x489D 0724
VPE_CFG_SC10	RW	32	0x0000 0028	0x489D 0728
VPE_CFG_SC11	RW	32	0x0000 002C	0x489D 072C
VPE_CFG_SC12	RW	32	0x0000 0030	0x489D 0730
VPE_CFG_SC13	RW	32	0x0000 0034	0x489D 0734
RESERVED	R	32	0x0000 0038	0x489D 0738
RESERVED	R	32	0x0000 003C	0x489D 073C
RESERVED	R	32	0x0000 0040	0x489D 0740
RESERVED	R	32	0x0000 0044	0x489D 0744
VPE_CFG_SC18	RW	32	0x0000 0048	0x489D 0748
VPE_CFG_SC19	RW	32	0x0000 004C	0x489D 074C
VPE_CFG_SC20	RW	32	0x0000 0050	0x489D 0750
VPE_CFG_SC21	RW	32	0x0000 0054	0x489D 0754

Table 10-69. VPE_SC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_SC Base Address
VPE_CFG_SC22	RW	32	0x0000 0058	0x489D 0758
RESERVED	R	32	0x0000 005C	0x489D 075C
VPE_CFG_SC24	RW	32	0x0000 0060	0x489D 0760
VPE_CFG_SC25	RW	32	0x0000 0064	0x489D 0764
RESERVED	R	32	0x0000 0068	0x489D 0768
RESERVED	R	32	0x0000 006C	0x489D 076C
RESERVED	R	32	0x0000 0070	0x489D 0770
RESERVED	R	32	0x0000 0074	0x489D 0774
RESERVED	R	32	0x0000 0078	0x489D 0778
RESERVED	R	32	0x0000 007C	0x489D 077C

10.4.3.2 VPE_SC Register Description

Table 10-70. VPE_CFG_SC0

Address Offset	0x0000 0000		
Physical Address	0x489D 0700	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															CFG_FID_SELFGEN	CFG_TRIM	CFG_Y_PK_EN	RESERVED		RESERVED	CFG_INTERLACE_I	CFG_HP_BYPASS	CFG_DCM_4X	CFG_DCM_2X	CFG_AUTO_HS	RESERVED	CFG_USE_RAV	CFG_INVT_FID	CFG_SC_BYPASS	CFG_LINEAR	CFG_INTERLACE_O

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CFG_FID_SELFGEN	FID self generate enable. When input is progressive and this bit is set, the SC generates self-toggling (top/bottom) output FID when performing interlacing.	RW	0x0
15	CFG_TRIM	Trimming enable. When 1, the input image whose size is specified by orgW and orgH registers is trimmed to the size with srcW and srcH from the offset specified by offW and offH. 0x0 : Disable trimming 0x1 : Enable trimming	RW	0x0
14	CFG_Y_PK_EN	This parameter is used by peaking block. 0: disable luma peaking 1: enable luma peaking	RW	0x0
13:12	RESERVED		R	0x0
11	RESERVED		R	0x0
10	CFG_INTERLACE_I	This parameter is used by horizontal and vertical scaling. 0x0 : The input video format is progressive 0x1 : The input video format is interlace	RW	0x0

Bits	Field Name	Description	Type	Reset
9	CFG_HP_BYPASS	This parameter is used by horizontal scaling. If cfg_auto_hs is 0, horizontal polyphase filter is always enabled. In this case, this register is DON'T CARE. If cfg_auto_hs is 1, then: 0x0: The polyphase scaler is always used regardless of the scaling ratio. 0x1: The polyphase scaler is bypassed only when (tar_w == src_w) or (tar_w == src_w/2) or (tar_w == src_w/4)	RW	0x0
8	CFG_DCM_4X	This parameter is used by horizontal scaling. 0: the 4X decimation filter is disabled 1: the 4X decimation filter is enabled **Note:** (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (horizontal scale ratio < 0.25). (3) This register is DON'T CARE when cfg_auto_hs = 1	RW	0x0
7	CFG_DCM_2X	This parameter is used by horizontal scaling. 0: the 2X decimation filter is disabled 1: the 2X decimation filter is enabled **Note:** (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (0.25 < horizontal scale ratio < 0.5). (3) This register is DON'T CARE when cfg_auto_hs = 1.	RW	0x0
6	CFG_AUTO_HS	This parameter is used by horizontal scaling. 0x0 : the cfg_dcm_2x and cfg_dcm_4x bits will enable appropriate decimation filters 0x1 : HW will decide whether up-scaling or down-scaling is required based on horizontal scaling ratio (SR). SR > 0.5 : horizontal polyphase filter is enabled, all decimation filters are disabled SR = 0.5 : dcm_2x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.5 > SR > 0.25 : dcm_2x and horizontal polyphase filter both are enabled SR = 0.25 : dcm_4x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.25 > SR > 0.125 : dcm_4x and horizontal polyphase filter are both enabled SR <= 0.125 : Functionally supported, but not recommended in auto mode for image quality concerns	RW	0x0
5	RESERVED		R	0x0
4	CFG_USE_RAV	This parameter is used by vertical scaling. 0x0 : Poly-phase filter will be used for the vertical scaling 0x1 : Running average filter will be used for the vertical scaling (down scaling only)	RW	0x0

Bits	Field Name	Description	Type	Reset
3	CFG_INVT_FID	This parameter is used by vertical scaling. 0x0 : Progressive input 0x1 : Interlaced input Must be set to 1 when CFG_INTERFACE_I = 1.	RW	0x0
2	CFG_SC_BYPASS	This parameter is general purpose. 0x0 : Scaling module will be engaged 0x1 : Scaling module will be bypassed	RW	0x0
1	CFG_LINEAR	This parameter is used by horizontal scaling. 0x0 : Anamorphic scaling 0x1 : Linear scaling	RW	0x0
0	CFG_INTERLACE_O	This parameter is used by vertical scaling. 0x0 : The output format of SC is progressive (default); 0x1 : The output format of SC is interlace	RW	0x0

Table 10-71. Register Call Summary for Register VPE_CFG_SC0

VPE Functional Description

- SC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]
- SC Code: [14] [15] [16] [17] [18] [19] [20] [21] [22]

VPE Register Manual

- VPE_SC Register Summary: [23]

Table 10-72. VPE_CFG_SC1

Address Offset	0x0000 0004		
Physical Address	0x489D 0704	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_ROW_ACC_INC																										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:0	CFG_ROW_ACC_INC	This parameter is used by vertical scaling. It defines the increment of the row accumulator in vertical poly-phase filter. It can be calculated by following formulas: For progressive in/progressive out row_acc_inc = round(2^16*(src_h-1)/(tar_h - 1)) For progressive_in/interlace_out row_acc_inc = round(2^16*2*(src_h-1)/(2*tar_h - 1)) For interlace_in/progressive_out row_acc_inc = round(2^16*(2*src_h-1)/(2*(tar_h - 1))) For interlace_in/interlace_out row_acc_inc = round(2^16*(2*src_h - 1)/(2*tar_h - 1)) In case of interlaced input, srcH is input field height (number of field lines), as specified in VPE_CFG_SC5. In case of interlaced output, tarH is output field height (number of field lines), as specified in VPE_CFG_SC4.	RW	0x0

Table 10-73. Register Call Summary for Register VPE_CFG_SC1

VPE Functional Description

- SC Functional Description: [0]

Table 10-73. Register Call Summary for Register VPE_CFG_SC1 (continued)

VPE Register Manual

- VPE_SC Register Summary: [1]

Table 10-74. VPE_CFG_SC2

Address Offset	0x0000 0008		
Physical Address	0x489D 0708	**Instance**	VPE_SC
Description			
Type	RW		

31:28	27:0
RESERVED	CFG_ROW_ACC_OFFSET

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this offset will be applied to a frame. In interlace mode: this offset will be applied to the top field.	RW	0x0

Table 10-75. Register Call Summary for Register VPE_CFG_SC2

VPE Functional Description

- SC Functional Description: [0]

VPE Register Manual

- VPE_SC Register Summary: [1]

Table 10-76. VPE_CFG_SC3

Address Offset	0x0000 000C		
Physical Address	0x489D 070C	**Instance**	VPE_SC
Description			
Type	RW		

31:28	27:0
RESERVED	CFG_ROW_ACC_OFFSET_B

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET_B	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this parameter will not be used. In interlace mode: this offset will be applied to the bottom field.	RW	0x0

Table 10-77. Register Call Summary for Register VPE_CFG_SC3

VPE Functional Description

- SC Functional Description: [0]

VPE Register Manual

- VPE_SC Register Summary: [1]

Table 10-78. VPE_CFG_SC4

Address Offset	0x0000 0010		
Physical Address	0x489D 0710	**Instance**	VPE_SC
Description			

Table 10-78. VPE_CFG_SC4 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CFG_NLIN_ACC_INIT_U			RESERVED	CFG_LIN_ACC_INC_U			RESERVED	CFG_TAR_W											RESERVED	CFG_TAR_H										

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	CFG_NLIN_ACC_INIT_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_init' that is defined in CFG_SC10	RW	0x0
27	RESERVED		RW	0x0
26:24	CFG_LIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'lin_acc_inc' that is defined in CFG_SC9	RW	0x0
23	RESERVED		RW	0x0
22:12	CFG_TAR_W	This parameter is a general purpose. Scaled target picture width. unit is pixel. This parameter defines the final output picture size	RW	0x0
11	RESERVED		RW	0x0
10:0	CFG_TAR_H	This parameter is a general purpose. Scaled target picture height (unit is line). This parameter defines the final output picture size. For the interlace output, it should be the number of lines per field.	RW	0x0

Table 10-79. Register Call Summary for Register VPE_CFG_SC4

VPE Functional Description

- SC Functional Description: [0] [1] [2] [3] [4] [5] [6]

VPE Register Manual

- VPE_SC Register Summary: [7]
- VPE_SC Register Description: [8]

Table 10-80. VPE_CFG_SC5

Address Offset	0x0000 0014		
Physical Address	0x489D 0714	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_NLIN_ACC_INC_U			RESERVED	CFG_SRC_W											RESERVED	CFG_SRC_H										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	CFG_NLIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_inc' that is defined in CFG_SC11	RW	0x0
23	RESERVED		RW	0x0
22:12	CFG_SRC_W	This parameter is a general purpose. This parameter defines the width of the source image	RW	0x0
11	RESERVED		RW	0x0

Bits	Field Name	Description	Type	Reset
10:0	CFG_SRC_H	This parameter is a general purpose. This parameter defines the height of the source image. For the interlace input, it should be the number of lines per field.	RW	0x0

Table 10-81. Register Call Summary for Register VPE_CFG_SC5

VPE Functional Description

- SC Functional Description: [0] [1] [2] [3] [4]

VPE Register Manual

- VPE_SC Register Summary: [5]
- VPE_SC Register Description: [6]

Table 10-82. VPE_CFG_SC6

Address Offset	0x0000 0018		
Physical Address	0x489D 0718	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CFG_ROW_ACC_INIT_RAV_B										CFG_ROW_ACC_INIT_RAV									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:10	CFG_ROW_ACC_INIT_RAV_B	This parameter is used by vertical scaling. it is used only when the input is interlace format. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for bottom field of interlace format)	RW	0x0
9:0	CFG_ROW_ACC_INIT_RAV	This parameter is used by vertical scaling. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for progressive format or top field of interlace format)	RW	0x0

Table 10-83. Register Call Summary for Register VPE_CFG_SC6

VPE Functional Description

- SC Functional Description: [0] [1] [2] [3]

VPE Register Manual

- VPE_SC Register Summary: [4]

Table 10-84. VPE_CFG_SC8

Address Offset	0x0000 0020		
Physical Address	0x489D 0720	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									CFG_NLIN_RIGHT											RESERVED	CFG_NLIN_LEFT										

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:12	CFG_NLIN_RIGHT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on right-hand side. In other words. it defines the location of the last pixel where the linear scaling is ended. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0
11	RESERVED		RW	0x0
10:0	CFG_NLIN_LEFT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on left-hand side. In other words. it defines the location of the last pixel in the left-sidenonlinear strip. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0

Table 10-85. Register Call Summary for Register VPE_CFG_SC8

VPE Functional Description

- SC Functional Description: [0] [1]

VPE Register Manual

- VPE_SC Register Summary: [2]

Table 10-86. VPE_CFG_SC9

Address Offset	0x0000 0024		
Physical Address	0x489D 0724	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_LIN_ACC_INC																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_LIN_ACC_INC	This parameter is used by horizontal scaling. It defines the increment of the linear accumulator. if SR > 0.5, then • lin_acc_inc = round(2^{24}*(srcWi -1) /(tarWi -1)) else if 0.25 < SR <= 0.5 • lin_acc_inc = round(2^24*(srcWi/2 -1) /(tarWi - 1)) else if SR <= 0.25 • lin_acc_inc = round(2^{24}*(srcWi/4 -1) /(tarWi - 1)) where srcWi and tarWi are the inner source width and the inner target width respectively.	RW	0x0

Table 10-87. Register Call Summary for Register VPE_CFG_SC9

VPE Functional Description

- SC Functional Description: [0] [1]

VPE Register Manual

- VPE_SC Register Summary: [2]

Table 10-88. VPE_CFG_SC10

Address Offset	0x0000 0028		
Physical Address	0x489D 0728	**Instance**	VPE_SC

Table 10-88. VPE_CFG_SC10 (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_NLIN_ACC_INIT																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_NLIN_ACC_INIT	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the initialization value of the nonlinear accumulator. nlin_acc_init = K*(1-2*d) Here the definitions of K and d are the same as in CFG_SC11	RW	0x0

Table 10-89. Register Call Summary for Register VPE_CFG_SC10

VPE Register Manual

- VPE_SC Register Summary: [0]

Table 10-90. VPE_CFG_SC11

Address Offset	0x0000 002C		
Physical Address	0x489D 072C	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_NLIN_ACC_INC																															

Bits	Field Name	Description	Type	Reset

31:0	CFG_NLIN_ACC_INC	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the increment of the nonlinear accumulator.
		if upscaling then
		d = 0
		if Ltar !=0 then
		K =round[2^{24}*Lsrc/(Ltar*Ltar)]
		where Lsrc= (srcW-srcWi)/2
		else K = 0
		elseif downscaling
		d = (tarW-1)/2
		if Ltar!=0 then
		K = round[2^{24} * Lsrc / (Ltar*(Ltar-2d))]
		where Lsrc= (srcW-srcWi)/(2n) and n=1..2 or 4
		else K = 0
		nlin_acc_inc = 2*K (negative for downscaling)
			RW	0x0

Table 10-91. Register Call Summary for Register VPE_CFG_SC11

VPE Register Manual

- VPE_SC Register Summary: [0]

Table 10-92. VPE_CFG_SC12

Address Offset	0x0000 0030		
Physical Address	0x489D 0730	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CFG_COL_ACC_OFFSET

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	CFG_COL_ACC_OFFSET	This parameter is used in horizontal scaling. It defines the luma accumulator's offset. Normally this parameter can be set as 0 if no horizontal offset is involved. In some applications, such as Pan and Scan. A corresponding offset value should be set. The format is 1.24.	RW	0x0

Table 10-93. Register Call Summary for Register VPE_CFG_SC12

VPE Functional Description

- SC Code: [0]

VPE Register Manual

- VPE_SC Register Summary: [1]

Table 10-94. VPE_CFG_SC13

Address Offset	0x0000 0034		
Physical Address	0x489D 0734	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CFG_SC_FACTOR_RAV									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_SC_FACTOR_RAV	This parameter is used by vertical scaling. Vertical scaling factor: It is defined as following: 1024*tarH/srcH. It is used for downscaling by the running average filter	RW	0x0

Table 10-95. Register Call Summary for Register VPE_CFG_SC13

VPE Functional Description

- SC Functional Description: [0] [1] [2]
- SC Code: [3] [4]

VPE Register Manual

- VPE_SC Register Summary: [5]

Table 10-96. VPE_CFG_SC18

Address Offset	0x0000 0048		
Physical Address	0x489D 0748	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CFG_HS_FACTOR									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_HS_FACTOR	This parameter is used by horizontal scaling. Horizontal-scaling-factor = tarWi/srcWi. Numerical format: 6.4 (6 bit integer and 4 bit fraction)	RW	0x0

Table 10-97. Register Call Summary for Register VPE_CFG_SC18

VPE Functional Description

- SC Code: [0]

VPE Register Manual

- VPE_SC Register Summary: [1]

Table 10-98. VPE_CFG_SC19

Address Offset	0x0000 004C		
Physical Address	0x489D 074C	**Instance**	VPE_SC
Description			
Type	RW		

31-24	23-16	15-8	7-0
CFG_HPF_COEF3	CFG_HPF_COEF2	CFG_HPF_COEF1	CFG_HPF_COEF0

Bits	Field Name	Description	Type	Reset
31:24	CFG_HPF_COEF3	This parameter is used by the peaking block. Defines the coefficient 3 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
23:16	CFG_HPF_COEF2	This parameter is used by the peaking block. Defines the coefficient 2 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
15:8	CFG_HPF_COEF1	This parameter is used by the peaking block. Defines the coefficient 1 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF0	This parameter is used by the peaking block. Defines the coefficient 0 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Table 10-99. Register Call Summary for Register VPE_CFG_SC19

VPE Functional Description

- SC Functional Description: [0] [1]
- SC Code: [2]

VPE Register Manual

- VPE_SC Register Summary: [3]

Table 10-100. VPE_CFG_SC20

Address Offset	0x0000 0050		
Physical Address	0x489D 0750	**Instance**	VPE_SC
Description			
Type	RW		

31-29	28-20	19	18-16	15-8	7-0
RESERVED	CFG_NL_LIMIT	RESERVED	CFG_HPF_NORM_SHIFT	CFG_HPF_COEF5	CFG_HPF_COEF4

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:20	CFG_NL_LIMIT	This parameter is used by the peaking block. The maximum of clipping.	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	CFG_HPF_NORM_SHIFT	This parameter is used by the peaking block. Defines the decimal point of the hpf coefficient.	RW	0x0
15:8	CFG_HPF_COEF5	This parameter is used by the peaking block. Defines the coefficient 5 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF4	This parameter is used by the peaking block. Defines the coefficient 4 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Table 10-101. Register Call Summary for Register VPE_CFG_SC20

VPE Functional Description

- SC Functional Description: [0] [1] [2]
- SC Code: [3]

VPE Register Manual

- VPE_SC Register Summary: [4]

Table 10-102. VPE_CFG_SC21

Address Offset	0x0000 0054		
Physical Address	0x489D 0754	**Instance**	VPE_SC
Description			
Type	RW		

31:24	23:16	15:9	8:0
RESERVED	CFG_NL_LO_SLOPE	RESERVED	CFG_NL_LO_THR

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	CFG_NL_LO_SLOPE	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The format is fixed point 4.4.	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_LO_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be 0	RW	0x0

Table 10-103. Register Call Summary for Register VPE_CFG_SC21

VPE Functional Description

- SC Functional Description: [0] [1]
- SC Code: [2]

VPE Register Manual

- VPE_SC Register Summary: [3]

Table 10-104. VPE_CFG_SC22

Address Offset	0x0000 0058		
Physical Address	0x489D 0758	**Instance**	VPE_SC
Description			
Type	RW		

31:19	18:16	15:9	8:0
RESERVED	CFG_NL_HI_SLOPE_SHIFT	RESERVED	CFG_NL_HI_THR

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:16	CFG_NL_HI_SLOPE_SHIFT	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The gain is $2^{(nl\_hi\_slope\_shift-3)}$.	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_HI_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be nl_hi_thr.	RW	0x0

Table 10-105. Register Call Summary for Register VPE_CFG_SC22

VPE Functional Description
- SC Functional Description: [0] [1] [2]
- SC Code: [3]

VPE Register Manual
- VPE_SC Register Summary: [4]

Table 10-106. VPE_CFG_SC24

Address Offset	0x0000 0060		
Physical Address	0x489D 0760	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_ORG_W											RESERVED					CFG_ORG_H										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_ORG_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0
15:11	RESERVED		R	0x0
10:0	CFG_ORG_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 10-107. Register Call Summary for Register VPE_CFG_SC24

VPE Functional Description
- SC Functional Description: [0] [1]

VPE Register Manual
- VPE_SC Register Summary: [2]

Table 10-108. VPE_CFG_SC25

Address Offset	0x0000 0064		
Physical Address	0x489D 0764	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_OFF_W											RESERVED					CFG_OFF_H										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_OFF_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:11	RESERVED		R	0x0
10:0	CFG_OFF_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 10-109. Register Call Summary for Register VPE_CFG_SC25

VPE Functional Description

- SC Functional Description: [0] [1]

VPE Register Manual

- VPE_SC Register Summary: [2]

10.4.4 VPE_CHR_US Registers

10.4.4.1 VPE_CHR_US Register Summary

Table 10-110. VPE_CHR_US Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_CHR_US_INST_0 Base Address	VPE_CHR_US_INST_1 Base Address	VPE_CHR_US_INST_2 Base Address
VPE_PID	RW	32	0x0000 0000	0x489D 0300	0x489D 0400	0x489D 0500
VPE_REG0	RW	32	0x0000 0004	0x489D 0304	0x489D 0404	0x489D 0504
VPE_REG1	RW	32	0x0000 0008	0x489D 0308	0x489D 0408	0x489D 0508
VPE_REG2	RW	32	0x0000 000C	0x489D 030C	0x489D 040C	0x489D 050C
VPE_REG3	RW	32	0x0000 0010	0x489D 0310	0x489D 0410	0x489D 0510
VPE_REG4	RW	32	0x0000 0014	0x489D 0314	0x489D 0414	0x489D 0514
VPE_REG5	RW	32	0x0000 0018	0x489D 0318	0x489D 0418	0x489D 0518
VPE_REG6	RW	32	0x0000 001C	0x489D 031C	0x489D 041C	0x489D 051C
VPE_REG7	RW	32	0x0000 0020	0x489D 0320	0x489D 0420	0x489D 0520

10.4.4.2 VPE_CHR_US Register Description

Table 10-111. VPE_PID

Address Offset	0x0000 0000		
Physical Address	0x489D 0300 0x489D 0400 0x489D 0500	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															

Bits	Field Name	Description	Type	Reset
31:0	PID		R	0x0

Table 10-112. Register Call Summary for Register VPE_PID

VPE Register Manual

- VPE_CHR_US Register Summary: [0]

Table 10-113. VPE_REG0

Address Offset	0x0000 0004		
Physical Address	0x489D 0304 0x489D 0404 0x489D 0504	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2

Table 10-113. VPE_REG0 (continued)

Description	
Type	RW

31–18	17–16	15–2	1–0
ANCHOR_FID0_C0	CFG_MODE	ANCHOR_FID0_C1	RESERVED

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID0_C0	C0 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
17:16	CFG_MODE	0x0 : Mode A 0x1 : Mode B	RW	0x0
15:2	ANCHOR_FID0_C1	C1 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-114. Register Call Summary for Register VPE_REG0

VPE Functional Description
- Modes of Operation (VPDMA): [0]
- Coefficient Configuration: [1] [2] [3] [4] [5] [6] [7]

VPE Register Manual
- VPE_CHR_US Register Summary: [8]

Table 10-115. VPE_REG1

Address Offset	0x0000 0008		
Physical Address	0x489D 0308 0x489D 0408 0x489D 0508	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31–18	17–16	15–2	1–0
ANCHOR_FID0_C2	RESERVED	ANCHOR_FID0_C3	RESERVED

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID0_C2	C2 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
17:16	RESERVED		RW	0x0
15:2	ANCHOR_FID0_C3	C3 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-116. Register Call Summary for Register VPE_REG1

VPE Functional Description
- Coefficient Configuration: [0] [1] [2] [3]

VPE Register Manual
- VPE_CHR_US Register Summary: [4]

Table 10-117. VPE_REG2

Address Offset	0x0000 000C		
Physical Address	0x489D 030C 0x489D 040C 0x489D 050C	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			

Table 10-117. VPE_REG2 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERP_FID0_C0														RESERVED		INTERP_FID0_C1														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID0_C0	C0 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
17:16	RESERVED		RW	0x0
15:2	INTERP_FID0_C1	C1 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-118. Register Call Summary for Register VPE_REG2

VPE Functional Description

- Coefficient Configuration: [0] [1] [2] [3]

VPE Register Manual

- VPE_CHR_US Register Summary: [4]

Table 10-119. VPE_REG3

Address Offset	0x0000 0010		
Physical Address	0x489D 0310 0x489D 0410 0x489D 0510	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERP_FID0_C2														RESERVED		INTERP_FID0_C3														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID0_C2	C2 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
17:16	RESERVED		RW	0x0
15:2	INTERP_FID0_C3	C3 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-120. Register Call Summary for Register VPE_REG3

VPE Functional Description

- Coefficient Configuration: [0] [1] [2] [3]

VPE Register Manual

- VPE_CHR_US Register Summary: [4]

Table 10-121. VPE_REG4

Address Offset	0x0000 0014		
Physical Address	0x489D 0314 0x489D 0414 0x489D 0514	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			

Table 10-121. VPE_REG4 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANCHOR_FID1_C0														RESERVED		ANCHOR_FID1_C1														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID1_C0	C0 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	ANCHOR_FID1_C1	C1 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-122. Register Call Summary for Register VPE_REG4

VPE Functional Description

- Coefficient Configuration: [0] [1] [2]

VPE Register Manual

- VPE_CHR_US Register Summary: [3]

Table 10-123. VPE_REG5

Address Offset	0x0000 0018		
Physical Address	0x489D 0318 0x489D 0418 0x489D 0518	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANCHOR_FID1_C2														RESERVED		ANCHOR_FID1_C3														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID1_C2	C2 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	ANCHOR_FID1_C3	C3 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-124. Register Call Summary for Register VPE_REG5

VPE Functional Description

- Coefficient Configuration: [0] [1]

VPE Register Manual

- VPE_CHR_US Register Summary: [2]

Table 10-125. VPE_REG6

Address Offset	0x0000 001C		
Physical Address	0x489D 031C 0x489D 041C 0x489D 051C	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INTERP_FID1_C0	RESERVED	INTERP_FID1_C1	RESERVED

Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID1_C0	C0 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	INTERP_FID1_C1	C1 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-126. Register Call Summary for Register VPE_REG6

VPE Functional Description

- Coefficient Configuration: [0] [1]

VPE Register Manual

- VPE_CHR_US Register Summary: [2]

Table 10-127. VPE_REG7

Address Offset	0x0000 0020		
Physical Address	0x489D 0320 0x489D 0420 0x489D 0520	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERP_FID1_C2														RESERVED		INTERP_FID1_C3														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID1_C2	C2 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	INTERP_FID1_C3	C3 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-128. Register Call Summary for Register VPE_REG7

VPE Functional Description

- Coefficient Configuration: [0] [1] [2]

VPE Register Manual

- VPE_CHR_US Register Summary: [3]

10.4.5 VPE_DEI Registers

10.4.5.1 VPE_DEI Register Summary

Table 10-129. VPE_DEI Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_DEI Base Address
VPE_DEI_REG0	RW	32	0x0000 0000	0x489D 0600
VPE_DEI_REG1	RW	32	0x0000 0004	0x489D 0604
VPE_DEI_REG2	RW	32	0x0000 0008	0x489D 0608
VPE_DEI_REG3	RW	32	0x0000 000C	0x489D 060C

Table 10-129. VPE_DEI Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_DEI Base Address
VPE_DEI_REG4	RW	32	0x0000 0010	0x489D 0610
VPE_DEI_REG5	RW	32	0x0000 0014	0x489D 0614
VPE_DEI_REG6	RW	32	0x0000 0018	0x489D 0618
VPE_DEI_REG7	RW	32	0x0000 001C	0x489D 061C
VPE_DEI_REG8	RW	32	0x0000 0020	0x489D 0620
VPE_DEI_REG9	RW	32	0x0000 0024	0x489D 0624
VPE_DEI_REG10	RW	32	0x0000 0028	0x489D 0628
VPE_DEI_REG11	RW	32	0x0000 002C	0x489D 062C
VPE_DEI_REG12	R	32	0x0000 0030	0x489D 0630
VPE_DEI_REG13	R	32	0x0000 0034	0x489D 0634
VPE_DEI_REG14	R	32	0x0000 0038	0x489D 0638

10.4.5.2 VPE_DEI Register Description

Table 10-130. VPE_DEI_REG0

Address Offset	0x0000 0000		
Physical Address	0x489D 0600	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28 27	26 … 16	15 … 11	10 … 0
PROGRESSIVE_BYPASS	FIELD_FLUSH	INTERLACE_BYPASS	RESERVED	HEIGHT	RESERVED	WIDTH

Bits	Field Name	Description	Type	Reset
31	PROGRESSIVE_BYPASS	Progressive Mode 0x0 : Normal Deinterlace Mode 0x1 : Progressive source	RW	0x0
30	FIELD_FLUSH	Field Flush Mode 0x0 : Normal Operation 0x1 : Flush Internal Pipe for Current output Frame	RW	0x0
29	INTERLACE_BYPASS	Interlace Bypass Mode 0x0 : Normal Deinterlace Mode 0x1 : Pass Interlace Content directly to output	RW	0x0
28:27	RESERVED	Always read as 0	R	0x0
26:16	HEIGHT	Frame Height	RW	0x0
15:11	RESERVED	Always read as 0	R	0x0
10:0	WIDTH	Frame Width	RW	0x0

Table 10-131. Register Call Summary for Register VPE_DEI_REG0

VPE Functional Description

- Bypass Mode: [0] [1]

Table 10-131. Register Call Summary for Register VPE_DEI_REG0 (continued)

VPE Register Manual

- VPE_DEI Register Summary: [2]

Table 10-132. VPE_DEI_REG1

Address Offset	0x0000 0004		
Physical Address	0x489D 0604	**Instance**	VPE_DEI
Description			
Type	RW		

31–2	1	0
RESERVED	MDT_SPATMAX_BYPASS	MDT_TEMPMAX_BYPASS

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	MDT_SPATMAX_BYPASS	Spatial Maximum Filtering Bypass for motion values used in EDI 0x0 : Enable 0x1 : Bypass	RW	0x0
0	MDT_TEMPMAX_BYPASS	Spatio-temporal Maximum Filtering Bypass for motion valued used in EDI 0x0 : Enable 0x1 : Bypass	RW	0x0

Table 10-133. Register Call Summary for Register VPE_DEI_REG1

VPE Register Manual

- VPE_DEI Register Summary: [0]

Table 10-134. VPE_DEI_REG2

Address Offset	0x0000 0008		
Physical Address	0x489D 0608	**Instance**	VPE_DEI
Description			
Type	RW		

31–28	27–24	23–16	15–8	7–0
MDT_MVSTMAX_COR_THR	MDT_MV_COR_THR	MDT_SF_SC_THR3	MDT_SF_SC_THR2	MDT_SF_SC_THR1

Bits	Field Name	Description	Type	Reset
31:28	MDT_MVSTMAX_COR_THR	This is used for increasing noise robustness. Increasing this threshold leads to more robustness to noise, but with the potential of introducing ghosting effect. Note that this threshold is used for motion values for EDI only, and it is in addition mdt_mv_cor_thr.	RW	0x0

Bits	Field Name	Description	Type	Reset
27:24	MDT_MV_COR_THR	This threshold is for the coring for motion value, mv. MDT will become more noise robust if this value increases. But the picture may be washed out if this value is set to high. This threshold can be interpreted as the noise threshold for calculating motion values for all blocks.	RW	0x0
23:16	MDT_SF_SC_THR3	Spatial frequency threshold 3	RW	0x0
15:8	MDT_SF_SC_THR2	Spatial frequency threshold 2	RW	0x0
7:0	MDT_SF_SC_THR1	Spatial frequency threshold It is used for adaptive scaling of motion values according to how busy the texture is. If the texture is flat, motion values need to be scaled up to reflect the sensitivity of motion values with respect to the detection error. Increasing the thresholds will make the motion value scaling more sensitive to the frequency of the texture. Note: 0 = mdt_sf_sc_thr1 = mdt_sf_sc_thr2 = mdt_sf_sc_thr3	RW	0x0

Table 10-135. Register Call Summary for Register VPE_DEI_REG2

VPE Functional Description

- Bypass Mode: [0] [1] [2]

VPE Register Manual

- VPE_DEI Register Summary: [3]

Table 10-136. VPE_DEI_REG3

Address Offset	0x0000 000C		
Physical Address	0x489D 060C	**Instance**	VPE_DEI
Description			
Type	RW		

31:24	23:16	15:8	7:4	3	2	1:0
EDI_COR_SCALE_FACTOR	EDI_DIR_COR_LOWER_THR	EDI_CHROMA3D_COR_THR	RESERVED	EDI_CHROMA3D_ENABLE	EDI_ENABLE_3D	EDI_INP_MODE

Bits	Field Name	Description	Type	Reset
31:24	EDI_COR_SCALE_FACTOR	Scaling factor for correlation along detected edge	RW	0x0
23:16	EDI_DIR_COR_LOWER_THR	Lower threshold used for correlation along detected edge	RW	0x0
15:8	EDI_CHROMA3D_COR_THR	Correlation threshold used in 3D processing for chroma. Because the motion values used for chroma 3D processing are based on luma only. Extra protection is needed. Temporal interpolation is only performed for chroma, when there is strong spatial or temporal correlation for the chroma pixel being processed. When the pixel difference is less than this threshold, it is assumed that there exists strong correlation between these two pixels. Thus, increasing this value leads to more chroma pixels being processed in 3D	RW	0x0
7:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3	EDI_CHROMA_3D_ENABLE	3D Chroma Enable 0x0 : Disable 3D processing for chroma 0x1 : Enable 3D processing (temporal interpolation)	RW	0x0
2	EDI_ENABLE_3D	3D Enable 0x0 : Disable 3D processing 0x1 : Enable 3D processing (temporal interpolation)	RW	0x0
1:0	EDI_INP_MODE	Interpolation mode. Note that mode 00 and 01 are used for debug purpose 0x0 : line average 0x1 : field average 0x2 : edge-directed interpolation for luma only 0x3 : edge-directed interpolation for both luma and chroma	RW	0x0

Table 10-137. Register Call Summary for Register VPE_DEI_REG3

VPE Functional Description

- Bypass Mode: [0] [1] [2]

VPE Register Manual

- VPE_DEI Register Summary: [3]

Table 10-138. VPE_DEI_REG4

Address Offset	0x0000 0010		
Physical Address	0x489D 0610	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT3					RESERVED			EDI_LUT2					RESERVED			EDI_LUT1					RESERVED			EDI_LUT0				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT3	EDI Lookup Table 3	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT2	EDI Lookup Table 2	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT1	EDI Lookup Table 1	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT0	EDI Lookup Table 0	RW	0x0

Table 10-139. Register Call Summary for Register VPE_DEI_REG4

VPE Register Manual

- VPE_DEI Register Summary: [0]

Table 10-140. VPE_DEI_REG5

Address Offset	0x0000 0014		
Physical Address	0x489D 0614	**Instance**	VPE_DEI
Description			

Table 10-140. VPE_DEI_REG5 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT7					RESERVED			EDI_LUT6					RESERVED			EDI_LUT5					RESERVED			EDI_LUT4				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT7	EDI Lookup Table 7	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT6	EDI Lookup Table 6	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT5	EDI Lookup Table 5	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT4	EDI Lookup Table 4	RW	0x0

Table 10-141. Register Call Summary for Register VPE_DEI_REG5

VPE Register Manual
- VPE_DEI Register Summary: [0]

Table 10-142. VPE_DEI_REG6

Address Offset	0x0000 0018		
Physical Address	0x489D 0618	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT11					RESERVED			EDI_LUT10					RESERVED			EDI_LUT9					RESERVED			EDI_LUT8				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT11	EDI Lookup Table 11	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT10	EDI Lookup Table 10	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT9	EDI Lookup Table 9	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT8	EDI Lookup Table 8	RW	0x0

Table 10-143. Register Call Summary for Register VPE_DEI_REG6

VPE Register Manual
- VPE_DEI Register Summary: [0]

Table 10-144. VPE_DEI_REG7

Address Offset	0x0000 001C		
Physical Address	0x489D 061C	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT15					RESERVED			EDI_LUT14					RESERVED			EDI_LUT13					RESERVED			EDI_LUT12				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT15	EDI Lookup Table 15	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT14	EDI Lookup Table 14	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT13	EDI Lookup Table 13	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT12	EDI Lookup Table 12	RW	0x0

Table 10-145. Register Call Summary for Register VPE_DEI_REG7

VPE Register Manual

- VPE_DEI Register Summary: [0]

Table 10-146. VPE_DEI_REG8

Address Offset	0x0000 0020		
Physical Address	0x489D 0620	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMD_WINDOW_ENABLE	RESERVED				FMD_WINDOW_MAXX											RESERVED					FMD_WINDOW_MINX										

Bits	Field Name	Description	Type	Reset
31	FMD_WINDOW_ENABLE	Enable FMD operation window	RW	0x0
30:27	RESERVED		R	0x0
26:16	FMD_WINDOW_MAXX	Right boundary of FMD operation window Must be less than width	RW	0x0
15:11	RESERVED		R	0x0
10:0	FMD_WINDOW_MINX	Left boundary of FMD operation window	RW	0x0

Table 10-147. Register Call Summary for Register VPE_DEI_REG8

VPE Functional Description

- Bypass Mode: [0] [1]

VPE Register Manual

- VPE_DEI Register Summary: [2]

Table 10-148. VPE_DEI_REG9

Address Offset	0x0000 0024		
Physical Address	0x489D 0624	**Instance**	VPE_DEI
Description			
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	FMD_WINDOW_MAXY	RESERVED	FMD_WINDOW_MINY

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	FMD_WINDOW_MAXY	Bottom boundary of FMD operation window Must be less than height/2	RW	0x0
15:11	RESERVED		R	0x0
10:0	FMD_WINDOW_MINY	Top boundary of FMD operation window	RW	0x0

Table 10-149. Register Call Summary for Register VPE_DEI_REG9

VPE Functional Description

- Bypass Mode: [0] [1]

VPE Register Manual

- VPE_DEI Register Summary: [2]

Table 10-150. VPE_DEI_REG10

Address Offset	0x0000 0028		
Physical Address	0x489D 0628	**Instance**	VPE_DEI
Description			
Type	RW		

31:24	23:16	15:4	3	2	1	0
FMD_CAF_LINE_THR	FMD_CAF_FIELD_THR	RESERVED	FMD_BED_ENABLE	FMD_JAM_DIR	FMD_LOCK	FMD_ENABLE

Bits	Field Name	Description	Type	Reset
31:24	FMD_CAF_LINE_THR	CAF threshold used for the pixels from two lines in one field This is the threshold used for combing artifacts detection. The difference of two consecutive lines from the same field (so there is one line in between if two fields are merged into one progressive frame) is compared with this threshold. Decreasing this threshold leads to be more conservative in detecting CAF. Both fmd_caf_field_thr and fmd_caf_line_thr are close the values that two pixels differed by this value is observable.	RW	0x0
23:16	FMD_CAF_FIELD_THR	CAF threshold used for the pixels from two fields This is the threshold used for combing artifacts detection. The difference of two consecutive lines (when merging two fields into one progressive frame) is used to compare with this threshold. Increasing this threshold leads to be more conservative in detecting CAF.	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3	FMD_BED_ENABLE	Film Mode Bad Edit Detection 0x0 : Disable 0x1 : Enable	RW	0x0
2	FMD_JAM_DIR	Film Mode Field Jamming Direction 0x0 : Current field jammed with previous field 0x1 : Current field jammed with next field	RW	0x0
1	FMD_LOCK	Film Mode Field Jamming Direction 0x0 : Current field jammed with previous field 0x1 : Current field jammed with next field	RW	0x0
0	FMD_ENABLE	Enable film mode processing 0x0 : Disable 0x1 : Enable	RW	0x0

Table 10-151. Register Call Summary for Register VPE_DEI_REG10

VPE Functional Description

- Bypass Mode: [0] [1] [2]

VPE Register Manual

- VPE_DEI Register Summary: [3]

Table 10-152. VPE_DEI_REG11

Address Offset	0x0000 002C		
Physical Address	0x489D 062C	**Instance**	VPE_DEI
Description			
Type	RW		

31:20	19:0
RESERVED	FMD_CAF_THR

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	FMD_CAF_THR	CAF threshold used for leaving film mode: If the combing artifacts is greater than this threshold, CAF is detected and thus the state machine will be forced to leave the film mode. If the user prefers to be more conservative in using film mode, decrease this threshold.	RW	0x0

Table 10-153. Register Call Summary for Register VPE_DEI_REG11

VPE Functional Description

- Bypass Mode: [0]

VPE Register Manual

- VPE_DEI Register Summary: [1]

Table 10-154. VPE_DEI_REG12

Address Offset	0x0000 0030		
Physical Address	0x489D 0630	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FMD_RESET	RESERVED	FMD_CAF

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	FMD_RESET	When “1”, the film mode detection module needs to be reset by the software. This bit needs to be checked at each occurrence of the film mode detection interrupt	R	0x0
23:21	RESERVED		R	0x0
20:0	FMD_CAF	Detected combing artifacts	R	0x0

Table 10-155. Register Call Summary for Register VPE_DEI_REG12

VPE Functional Description

- Bypass Mode: [0] [1]

VPE Register Manual

- VPE_DEI Register Summary: [2]

Table 10-156. VPE_DEI_REG13

Address Offset	0x0000 0034		
Physical Address	0x489D 0634	**Instance**	VPE_DEI
Description			
Type	R		

31:28	27:0
RESERVED	FMD_FIELD_DIFF

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	FMD_FIELD_DIFF	Field difference (difference between two neighboring fields, one top and one bottom)	R	0x0

Table 10-157. Register Call Summary for Register VPE_DEI_REG13

VPE Functional Description

- Bypass Mode: [0]

VPE Register Manual

- VPE_DEI Register Summary: [1]

Table 10-158. VPE_DEI_REG14

Address Offset	0x0000 0038		
Physical Address	0x489D 0638	**Instance**	VPE_DEI
Description			
Type	R		

31:20	19:0
RESERVED	FMD_FRAME_DIFF

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19:0	FMD_FRAME_DIFF	Frame difference (difference between two top or two bottom fields)	R	0x0

Table 10-159. Register Call Summary for Register VPE_DEI_REG14

VPE Functional Description

- Bypass Mode: [0]

VPE Register Manual

- VPE_DEI Register Summary: [1]

10.4.6 VPE_VPDMA Registers

Note

The functionality of the following sets of registers is not supported by VPE VPDMA in this family of devices:

- All VPE_INT1_* registers
- All VPE_INT2_* registers
- All VPE_INT3_* registers

The following channels are not used by VPE VPDMA in this family of devices. All register bit-fields corresponding to these channels should be kept at their reset value.

- VIP1_*, except for the following:
  - VIP1_PORTA_RGB
  - VIP1_PORTA_LUMA
  - VIP1_PORTA_CHROMA
- VIP2_*
- GRPX_*
- SCALER_OUT
- SCALER_LUMA
- SCALER_CHROMA
- NF_*
- TRANSCODE1_*
- TRANSCODE2_*
- AUX_IN
- PIP_FRAME
- POST_COMP_WR
- VBI_SD_VENC

Note

The following clients are not used by VPE VPDMA in this family of devices. All register bit-fields corresponding to these clients should be kept at their reset value.

- VIP1_*, except for the following:
  - VIP1_UP_UV
  - VIP1_UP_Y
- VIP2_*
- TRANS1_LUMA
- TRANS1_CHROMA
- TRANS2_LUMA
- TRANS2_CHROMA
- HDMI_WRBK
- VBI_SDVENC
- NF_420_UV_OUT
- NF_420_Y_OUT
- NF_420_UV_IN
- NF_420_Y_IN
- NF_422_IN
- GRPX1_ST
- GRPX2_ST
- GRPX3_ST
- GRPX1_DATA
- GRPX2_DATA
- GRPX3_DATA
- PIP_WRBK
- SC_IN_*
- SC_OUT
- COMP_WRBK

10.4.6.1 VPE_VPDMA Register Summary

Table 10-160. VPE_VPDMA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_VPDMA_PID	R	32	0x0000 0000	0x489D D000
VPE_LIST_ADDR	RW	32	0x0000 0004	0x489D D004
VPE_LIST_ATTR	RW	32	0x0000 0008	0x489D D008
VPE_LIST_STAT_SYNC	RW	32	0x0000 000C	0x489D D00C
VPE_BG_RGB	RW	32	0x0000 0018	0x489D D018
VPE_BG_YUV	RW	32	0x0000 001C	0x489D D01C
VPE_VPDMA_SETUP	RW	32	0x0000 0030	0x489D D030
VPE_MAX_SIZE1	RW	32	0x0000 0034	0x489D D034
VPE_MAX_SIZE2	RW	32	0x0000 0038	0x489D D038
VPE_MAX_SIZE3	RW	32	0x0000 003C	0x489D D03C
VPE_INT0_CHANNEL0_INT_STAT	RW	32	0x0000 0040	0x489D D040
VPE_INT0_CHANNEL0_INT_MASK	RW	32	0x0000 0044	0x489D D044
VPE_INT0_CHANNEL1_INT_STAT	RW	32	0x0000 0048	0x489D D048
VPE_INT0_CHANNEL1_INT_MASK	RW	32	0x0000 004C	0x489D D04C
VPE_INT0_CHANNEL2_INT_STAT	RW	32	0x0000 0050	0x489D D050
VPE_INT0_CHANNEL2_INT_MASK	RW	32	0x0000 0054	0x489D D054

Table 10-160. VPE_VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_INT0_CHANNEL3_INT_STAT	RW	32	0x0000 0058	0x489D D058
VPE_INT0_CHANNEL3_INT_MASK	RW	32	0x0000 005C	0x489D D05C
VPE_INT0_CHANNEL4_INT_STAT	RW	32	0x0000 0060	0x489D D060
VPE_INT0_CHANNEL4_INT_MASK	RW	32	0x0000 0064	0x489D D064
VPE_INT0_CHANNEL5_INT_STAT	RW	32	0x0000 0068	0x489D D068
VPE_INT0_CHANNEL5_INT_MASK	RW	32	0x0000 006C	0x489D D06C
VPE_INT0_CLIENT0_INT_STAT	RW	32	0x0000 0078	0x489D D078
VPE_INT0_CLIENT0_INT_MASK	RW	32	0x0000 007C	0x489D D07C
VPE_INT0_CLIENT1_INT_STAT	RW	32	0x0000 0080	0x489D D080
VPE_INT0_CLIENT1_INT_MASK	RW	32	0x0000 0084	0x489D D084
VPE_INT0_LIST0_INT_STAT	RW	32	0x0000 0088	0x489D D088
VPE_INT0_LIST0_INT_MASK	RW	32	0x0000 008C	0x489D D08C
VPE_INT1_CHANNEL0_INT_STAT	RW	32	0x0000 0090	0x489D D090
VPE_INT1_CHANNEL0_INT_MASK	RW	32	0x0000 0094	0x489D D094
VPE_INT1_CHANNEL1_INT_STAT	RW	32	0x0000 0098	0x489D D098
VPE_INT1_CHANNEL1_INT_MASK	RW	32	0x0000 009C	0x489D D09C
VPE_INT1_CHANNEL2_INT_STAT	RW	32	0x0000 00A0	0x489D D0A0
VPE_INT1_CHANNEL2_INT_MASK	RW	32	0x0000 00A4	0x489D D0A4
VPE_INT1_CHANNEL3_INT_STAT	RW	32	0x0000 00A8	0x489D D0A8
VPE_INT1_CHANNEL3_INT_MASK	RW	32	0x0000 00AC	0x489D D0AC
VPE_INT1_CHANNEL4_INT_STAT	RW	32	0x0000 00B0	0x489D D0B0
VPE_INT1_CHANNEL4_INT_MASK	RW	32	0x0000 00B4	0x489D D0B4
VPE_INT1_CHANNEL5_INT_STAT	RW	32	0x0000 00B8	0x489D D0B8
VPE_INT1_CHANNEL5_INT_MASK	RW	32	0x0000 00BC	0x489D D0BC
VPE_INT1_CLIENT0_INT_STAT	RW	32	0x0000 00C8	0x489D D0C8
VPE_INT1_CLIENT0_INT_MASK	RW	32	0x0000 00CC	0x489D D0CC
VPE_INT1_LIST0_INT_STAT	RW	32	0x0000 00D8	0x489D D0D8
VPE_INT1_LIST0_INT_MASK	RW	32	0x0000 00DC	0x489D D0DC
VPE_INT2_CHANNEL0_INT_STAT	RW	32	0x0000 00E0	0x489D D0E0
VPE_INT2_CHANNEL0_INT_MASK	RW	32	0x0000 00E4	0x489D D0E4
VPE_INT2_CHANNEL1_INT_STAT	RW	32	0x0000 00E8	0x489D D0E8
VPE_INT2_CHANNEL1_INT_MASK	RW	32	0x0000 00EC	0x489D D0EC
VPE_INT2_CHANNEL2_INT_STAT	RW	32	0x0000 00F0	0x489D D0F0
VPE_INT2_CHANNEL2_INT_MASK	RW	32	0x0000 00F4	0x489D D0F4
VPE_INT2_CHANNEL3_INT_STAT	RW	32	0x0000 00F8	0x489D D0F8
VPE_INT2_CHANNEL3_INT_MASK	RW	32	0x0000 00FC	0x489D D0FC
VPE_INT2_CHANNEL4_INT_STAT	RW	32	0x0000 0100	0x489D D100
VPE_INT2_CHANNEL4_INT_MASK	RW	32	0x0000 0104	0x489D D104
VPE_INT2_CHANNEL5_INT_STAT	RW	32	0x0000 0108	0x489D D108
VPE_INT2_CHANNEL5_INT_MASK	RW	32	0x0000 010C	0x489D D10C
VPE_INT2_CLIENT0_INT_STAT	RW	32	0x0000 0118	0x489D D118
VPE_INT2_CLIENT0_INT_MASK	RW	32	0x0000 011C	0x489D D11C
VPE_INT2_LIST0_INT_STAT	RW	32	0x0000 0128	0x489D D128
VPE_INT2_LIST0_INT_MASK	RW	32	0x0000 012C	0x489D D12C
VPE_INT3_CHANNEL0_INT_STAT	RW	32	0x0000 0130	0x489D D130

Table 10-160. VPE_VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_INT3_CHANNEL0_INT_MASK	RW	32	0x0000 0134	0x489D D134
VPE_INT3_CHANNEL1_INT_STAT	RW	32	0x0000 0138	0x489D D138
VPE_INT3_CHANNEL1_INT_MASK	RW	32	0x0000 013C	0x489D D13C
VPE_INT3_CHANNEL2_INT_STAT	RW	32	0x0000 0140	0x489D D140
VPE_INT3_CHANNEL2_INT_MASK	RW	32	0x0000 0144	0x489D D144
VPE_INT3_CHANNEL3_INT_STAT	RW	32	0x0000 0148	0x489D D148
VPE_INT3_CHANNEL3_INT_MASK	RW	32	0x0000 014C	0x489D D14C
VPE_INT3_CHANNEL4_INT_STAT	RW	32	0x0000 0150	0x489D D150
VPE_INT3_CHANNEL4_INT_MASK	RW	32	0x0000 0154	0x489D D154
VPE_INT3_CHANNEL5_INT_STAT	RW	32	0x0000 0158	0x489D D158
VPE_INT3_CHANNEL5_INT_MASK	RW	32	0x0000 015C	0x489D D15C
VPE_INT3_CLIENT0_INT_STAT	RW	32	0x0000 0168	0x489D D168
VPE_INT3_CLIENT0_INT_MASK	RW	32	0x0000 016C	0x489D D16C
VPE_INT3_LIST0_INT_STAT	RW	32	0x0000 0178	0x489D D178
VPE_INT3_LIST0_INT_MASK	RW	32	0x0000 017C	0x489D D17C
VPE_PERF_MON0	RW	32	0x0000 0200	0x489D D200
VPE_PERF_MON1	RW	32	0x0000 0204	0x489D D204
VPE_PERF_MON2	RW	32	0x0000 0208	0x489D D208
VPE_PERF_MON3	RW	32	0x0000 020C	0x489D D20C
VPE_PERF_MON4	RW	32	0x0000 0210	0x489D D210
VPE_PERF_MON5	RW	32	0x0000 0214	0x489D D214
VPE_PERF_MON6	RW	32	0x0000 0218	0x489D D218
VPE_PERF_MON7	RW	32	0x0000 021C	0x489D D21C
VPE_PERF_MON8	RW	32	0x0000 0220	0x489D D220
VPE_PERF_MON9	RW	32	0x0000 0224	0x489D D224
VPE_PERF_MON10	RW	32	0x0000 0228	0x489D D228
VPE_PERF_MON11	RW	32	0x0000 022C	0x489D D22C
VPE_PERF_MON12	RW	32	0x0000 0230	0x489D D230
VPE_PERF_MON13	RW	32	0x0000 0234	0x489D D234
VPE_PERF_MON14	RW	32	0x0000 0238	0x489D D238
VPE_PERF_MON15	RW	32	0x0000 023C	0x489D D23C
VPE_PERF_MON16	RW	32	0x0000 0240	0x489D D240
VPE_PERF_MON17	RW	32	0x0000 0244	0x489D D244
VPE_PERF_MON18	RW	32	0x0000 0248	0x489D D248
VPE_PERF_MON19	RW	32	0x0000 024C	0x489D D24C
VPE_PERF_MON20	RW	32	0x0000 0250	0x489D D250
VPE_PERF_MON21	RW	32	0x0000 0254	0x489D D254
VPE_PERF_MON22	RW	32	0x0000 0258	0x489D D258
VPE_PERF_MON23	RW	32	0x0000 025C	0x489D D25C
VPE_PERF_MON24	RW	32	0x0000 0260	0x489D D260
VPE_PERF_MON25	RW	32	0x0000 0264	0x489D D264
VPE_PERF_MON26	RW	32	0x0000 0268	0x489D D268
VPE_PERF_MON27	RW	32	0x0000 026C	0x489D D26C
VPE_PERF_MON28	RW	32	0x0000 0270	0x489D D270
VPE_PERF_MON29	RW	32	0x0000 0274	0x489D D274

Table 10-160. VPE_VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_PERF_MON30	RW	32	0x0000 0278	0x489D D278
VPE_PERF_MON31	RW	32	0x0000 027C	0x489D D27C
VPE_PERF_MON32	RW	32	0x0000 0280	0x489D D280
VPE_PERF_MON33	RW	32	0x0000 0284	0x489D D284
VPE_PERF_MON34	RW	32	0x0000 0288	0x489D D288
VPE_PERF_MON35	RW	32	0x0000 028C	0x489D D28C
VPE_PERF_MON36	RW	32	0x0000 0290	0x489D D290
VPE_PERF_MON37	RW	32	0x0000 0294	0x489D D294
VPE_PERF_MON38	RW	32	0x0000 0298	0x489D D298
VPE_PERF_MON39	RW	32	0x0000 029C	0x489D D29C
VPE_PERF_MON40	RW	32	0x0000 02A0	0x489D D2A0
VPE_PERF_MON41	RW	32	0x0000 02A4	0x489D D2A4
VPE_PERF_MON42	RW	32	0x0000 02A8	0x489D D2A8
VPE_PERF_MON43	RW	32	0x0000 02AC	0x489D D2AC
VPE_PERF_MON44	RW	32	0x0000 02B0	0x489D D2B0
VPE_PERF_MON45	RW	32	0x0000 02B4	0x489D D2B4
VPE_PERF_MON46	RW	32	0x0000 02B8	0x489D D2B8
VPE_PERF_MON47	RW	32	0x0000 02BC	0x489D D2BC
VPE_PERF_MON48	RW	32	0x0000 02C0	0x489D D2C0
VPE_PERF_MON49	RW	32	0x0000 02C4	0x489D D2C4
VPE_PERF_MON50	RW	32	0x0000 02C8	0x489D D2C8
VPE_PERF_MON51	RW	32	0x0000 02CC	0x489D D2CC
VPE_PERF_MON52	RW	32	0x0000 02D0	0x489D D2D0
VPE_PRI_CHROMA_CSTAT	RW	32	0x0000 0300	0x489D D300
VPE_PRI_LUMA_CSTAT	RW	32	0x0000 0304	0x489D D304
VPE_PRI_FLD1_LUMA_CSTAT	RW	32	0x0000 0308	0x489D D308
VPE_PRI_FLD1_CHROMA_CSTAT	RW	32	0x0000 030C	0x489D D30C
VPE_PRI_FLD2_LUMA_CSTAT	RW	32	0x0000 0310	0x489D D310
VPE_PRI_FLD2_CHROMA_CSTAT	RW	32	0x0000 0314	0x489D D314
VPE_PRI_MV0_CSTAT	RW	32	0x0000 0330	0x489D D330
VPE_PRI_MV_OUT_CSTAT	RW	32	0x0000 033C	0x489D D33C
VPE_VIP0_UP_Y_CSTAT	RW	32	0x0000 0390	0x489D D390
VPE_VIP0_UP_UV_CSTAT	RW	32	0x0000 0394	0x489D D394
VPE_VPI_CTL_CSTAT	RW	32	0x0000 03D0	0x489D D3D0

10.4.6.2 VPE_VPDMA Register Description

Table 10-161. VPE_VPDMA_PID

Address Offset	0x0000 0000		
Physical Address	0x489D D000	**Instance**	VPE_VPDMA
Description	This register follows the format described in PDR3.5		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SCHEME	FUNC	RTL	MAJOR	VPDMA_LOAD_COMPLETE	VPDMA_ACCESS_TYPE	MINOR

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	The scheme of the register used. Currently this is PDR 3.5 Scheme	R	0x0
29:16	FUNC	The funcition of the module being used. The value is for vpe0_vayu_vpdma.	R	0x0
15:11	RTL	RTL Release Version The PDR release number of this IP. After Bootup this value becomes the firmware Revision ID	R	0x0
10:8	MAJOR	Major Release Number	R	0x0
7	VPDMA_LOAD_COMPLETE	This bit will be 1 when the VPDMA state machines image and data image have successfuly been fetched and loaded.	R	0x0
6	VPDMA_ACCESS_TYPE	After bootup this bit states how DMA transaction are setup by lists or through register access. 0x0 : Lists 0x1 : Register Access	R	0x0
5:0	MINOR	Minor Release Number	R	0x0

Table 10-162. Register Call Summary for Register VPE_VPDMA_PID

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-163. VPE_LIST_ADDR

Address Offset	0x0000 0004		
Physical Address	0x489D D004	**Instance**	VPE_VPDMA
Description	The location of a new list to begin processing.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIST_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	LIST_ADDR	Location of a new list of descriptors. This register must be written with the VPDMA Configuration Location after reset.	RW	0x0

Table 10-164. Register Call Summary for Register VPE_LIST_ADDR

VPE Functional Description

- VPDMA Introduction:
- VPDMA Basic Definitions: [1] [2]
- VPDMA Configuration: [3] [4] [5]

Table 10-164. Register Call Summary for Register VPE_LIST_ADDR (continued)

VPE Register Manual

- VPE_VPDMA Register Summary: [6]
- VPE_VPDMA Register Description: [7]

Table 10-165. VPE_LIST_ATTR

Address Offset	0x0000 0008		
Physical Address	0x489D D008	**Instance**	VPE_VPDMA
Description	The attributes of a new list. This register should always be written after list_addr.		
Type	RW		

31–27	26–24	23–21	20	19	18–16	15–0
RESERVED	LIST_NUM	RESERVED	STOP	RDY	LIST_TYPE	LIST_SIZE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	LIST_NUM	The list number that should be assigned to the list located at LIST_ADDR. If the list is still active this will block all future list writes until the list is available.	RW	0x0
23:21	RESERVED		R	0x0
20	STOP	This bit is written with the LIST_NUMBER field to stop a self-modifying list. When this bit is written a one the list specified by the LIST_NUMBER is sent a stop signal and will finish the current frame of transfers and then free the list resources.	RW	0x0
19	RDY	This bit is low when a new list cannot be written to the VPE_LIST_ADDR register. The reasons this bit would be low are at initial startup if the LIST_MANAGER State Machine image has not completed loading. It also would be low if the last write to the LIST_ATTR attempted to start a list that is currently active. When this bit is low any writes to the list address register will cause access to not be accepted until this bit has set by the previous list having completed.	R	0x0
18:16	LIST_TYPE	The type of list that has been generated. 0x0 : Normal List 0x1 : Self-Modifying List 0x2 : List Doorbell Others Reserved for future use	RW	0x0
15:0	LIST_SIZE	Number of 128 bit word in the new list of descriptors. Writes to this register will activate the list in the list stack of the list manager and begin transfer of the list into VPDMA. This size can not be 0.	RW	0x0

Table 10-166. Register Call Summary for Register VPE_LIST_ATTR

VPE Functional Description

- VPDMA Configuration: [0] [1] [2]

VPE Register Manual

- VPE_VPDMA Register Summary: [3]

Table 10-167. VPE_LIST_STAT_SYNC

Address Offset	0x0000 000C		
Physical Address	0x489D D00C	**Instance**	VPE_VPDMA
Description	The register is used for processor to List Manager syncronization and status registers for the list.		

Table 10-167. VPE_LIST_STAT_SYNC (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LIST7_BUSY	LIST6_BUSY	LIST5_BUSY	LIST4_BUSY	LIST3_BUSY	LIST2_BUSY	LIST1_BUSY	LIST0_BUSY	RESERVED								SYNC_LISTS7	SYNC_LISTS6	SYNC_LISTS5	SYNC_LISTS4	SYNC_LISTS3	SYNC_LISTS2	SYNC_LISTS1	SYNC_LISTS0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	LIST7_BUSY	The list 7 is currently running. Any attempt to load a new list to list 7 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
22	LIST6_BUSY	The list 6 is currently running. Any attempt to load a new list to list 6 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
21	LIST5_BUSY	The list 5 is currently running. Any attempt to load a new list to list 5 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
20	LIST4_BUSY	The list 4 is currently running. Any attempt to load a new list to list 4 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
19	LIST3_BUSY	The list 3 is currently running. Any attempt to load a new list to list 3 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
18	LIST2_BUSY	The list 2 is currently running. Any attempt to load a new list to list 2 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
17	LIST1_BUSY	The list 1 is currently running. Any attempt to load a new list to list 1 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
16	LIST0_BUSY	The list 0 is currently running. Any attempt to load a new list to list 0 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
15:8	RESERVED		R	0x0
7	SYNC_LISTS7	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 7 waiting on it.	RW	0x0
6	SYNC_LISTS6	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 6 waiting on it.	RW	0x0
5	SYNC_LISTS5	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 5 waiting on it.	RW	0x0
4	SYNC_LISTS4	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 4 waiting on it.	RW	0x0
3	SYNC_LISTS3	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 3 waiting on it.	RW	0x0
2	SYNC_LISTS2	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 2 waiting on it.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	SYNC_LISTS1	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 1 waiting on it.	RW	0x0
0	SYNC_LISTS0	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 0 waiting on it.	RW	0x0

Table 10-168. Register Call Summary for Register VPE_LIST_STAT_SYNC

VPE Functional Description

- VPDMA Descriptors: [0] [1] [2]

VPE Register Manual

- VPE_VPDMA Register Summary: [3]

Table 10-169. VPE_BG_RGB

Address Offset	0x0000 0018		
Physical Address	0x489D D018	**Instance**	VPE_VPDMA
Description	The registers used to set the background color for RGB		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED								GREEN								BLUE								BLEND							

Bits	Field Name	Description	Type	Reset
31:24	RED	The red value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
23:16	GREEN	The green value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
15:8	BLUE	The blue value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	BLEND	The blend value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0

Table 10-170. Register Call Summary for Register VPE_BG_RGB

VPE Functional Description

- VPDMA Configuration:
- VPDMA Data Formats: [3] [4]

VPE Register Manual

- VPE_VPDMA Register Summary: [5]

Table 10-171. VPE_BG_YUV

Address Offset	0x0000 001C		
Physical Address	0x489D D01C	**Instance**	VPE_VPDMA
Description	The registers used to set the background color for YUV		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								Y								CR								CB							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	Y	The Y value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0

Bits	Field Name	Description	Type	Reset
15:8	CR	The Cr value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	CB	The Cb value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0

Table 10-172. Register Call Summary for Register VPE_BG_YUV

VPE Functional Description

- VPDMA Configuration:

VPE Register Manual

- VPE_VPDMA Register Summary: [5]

Table 10-173. VPE_VPDMA_SETUP

Address Offset	0x0000 0030		
Physical Address	0x489D D030	**Instance**	VPE_VPDMA
Description	Configures global parameters that are shared by all clients.		
Type	RW		

31:1	0
RESERVED	SEC_BASE_CH

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SEC_BASE_CH	Use Secondary Channels for Mosaic mode	RW	0x0

Table 10-174. Register Call Summary for Register VPE_VPDMA_SETUP

VPE Functional Description

- VPDMA Configuration:

VPE Register Manual

- VPE_VPDMA Register Summary: [2]

Table 10-175. VPE_MAX_SIZE1

Address Offset	0x0000 0034		
Physical Address	0x489D D034	**Instance**	VPE_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 1 in write descriptor.		
Type	RW		

31:16	15:0
MAX_WIDTH	MAX_HEIGHT

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 1 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 1 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 10-176. Register Call Summary for Register VPE_MAX_SIZE1

VPE Functional Description

- VPDMA Descriptors: [0] [1] [2] [3] [4]

VPE Register Manual

- VPE_VPDMA Register Summary: [5]

Table 10-177. VPE_MAX_SIZE2

Address Offset	0x0000 0038		
Physical Address	0x489D D038	**Instance**	VPE_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 2 in write descriptor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_WIDTH																MAX_HEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 2 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 2 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 10-178. Register Call Summary for Register VPE_MAX_SIZE2

VPE Functional Description

- VPDMA Descriptors: [0] [1] [2] [3] [4]

VPE Register Manual

- VPE_VPDMA Register Summary: [5]

Table 10-179. VPE_MAX_SIZE3

Address Offset	0x0000 003C		
Physical Address	0x489D D03C	**Instance**	VPE_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 3 in write descriptor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_WIDTH																MAX_HEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 3 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 3 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 10-180. Register Call Summary for Register VPE_MAX_SIZE3

VPE Functional Description

- VPDMA Descriptors: [0] [1] [2] [3] [4]

VPE Register Manual

- VPE_VPDMA Register Summary: [5]

Table 10-181. VPE_INT0_CHANNEL0_INT_STAT

Address Offset	0x0000 0040		
Physical Address	0x489D D040	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX3	INT_STAT_GRPX2	INT_STAT_GRPX1	INT_STAT_SCALER_OUT	RESERVED								INT_STAT_SCALER_CHROMA	INT_STAT_SCALER_LUMA	INT_STAT_HQ_SCALER	RESERVED	INT_STAT_HQ_MV_OUT	RESERVED		INT_STAT_HQ_MV	RESERVED						INT_STAT_HQ_VID3_CHROMA	INT_STAT_HQ_VID3_LUMA	INT_STAT_HQ_VID2_CHROMA	INT_STAT_HQ_VID2_LUMA	INT_STAT_HQ_VID1_CHROMA	INT_STAT_HQ_VID1_LUMA

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX3	The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_GRPX2	The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_GRPX1	The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_SCALER_OUT	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value	RW	0x0

Bits	Field Name	Description	Type	Reset
27:20	RESERVED	Reserved	R	0x00
19	INT_STAT_SCALER_CHROMA	The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_SCALER_LUMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_HQ_SCALER	The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_HQ_MV_OUT	The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_HQ_MV	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_HQ_VID3_CHROMA	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_HQ_VID3_LUMA	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_HQ_VID2_CHROMA	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
2	INT_STAT_HQ_VID2_LUMA	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_HQ_VID1_CHROMA	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_HQ_VID1_LUMA	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-182. Register Call Summary for Register VPE_INT0_CHANNEL0_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-183. VPE_INT0_CHANNEL0_INT_MASK

Address Offset	0x0000 0044		
Physical Address	0x489D D044	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX3	INT_MASK_GRPX2	INT_MASK_GRPX1	INT_MASK_SCALER_OUT	RESERVED								INT_MASK_SCALER_CHROMA	INT_MASK_SCALER_LUMA	INT_MASK_HQ_SCALER	RESERVED	INT_MASK_HQ_MV_OUT	RESERVED		INT_MASK_HQ_MV	RESERVED						INT_MASK_HQ_VID3_CHROMA	INT_MASK_HQ_VID3_LUMA	INT_MASK_HQ_VID2_CHROMA	INT_MASK_HQ_VID2_LUMA	INT_MASK_HQ_VID1_CHROMA	INT_MASK_HQ_VID1_LUMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX3	The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_GRPX2	The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
29	INT_MASK_GRPX1	The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_SCALER_OUT	The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_MASK_SCALER_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_SCALER_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_HQ_SCALER	The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_HQ_MV_OUT	The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_HQ_MV	The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_HQ_VID3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_HQ_VID3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_HQ_VID2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_HQ_VID2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_HQ_VID1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_HQ_VID1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-184. Register Call Summary for Register VPE_INT0_CHANNEL0_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-185. VPE_INT0_CHANNEL1_INT_STAT

Address Offset	0x0000 0048

Table 10-185. VPE_INT0_CHANNEL1_INT_STAT (continued)

Physical Address	0x489D D048	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_PORTB_SRC9	INT_STAT_VIP1_MULT_PORTB_SRC8	INT_STAT_VIP1_MULT_PORTB_SRC7	INT_STAT_VIP1_MULT_PORTB_SRC6	INT_STAT_VIP1_MULT_PORTB_SRC5	INT_STAT_VIP1_MULT_PORTB_SRC4	INT_STAT_VIP1_MULT_PORTB_SRC3	INT_STAT_VIP1_MULT_PORTB_SRC2	INT_STAT_VIP1_MULT_PORTB_SRC1	INT_STAT_VIP1_MULT_PORTB_SRC0	INT_STAT_VIP1_MULT_PORTA_SRC15	INT_STAT_VIP1_MULT_PORTA_SRC14	INT_STAT_VIP1_MULT_PORTA_SRC13	INT_STAT_VIP1_MULT_PORTA_SRC12	INT_STAT_VIP1_MULT_PORTA_SRC11	INT_STAT_VIP1_MULT_PORTA_SRC10	INT_STAT_VIP1_MULT_PORTA_SRC9	INT_STAT_VIP1_MULT_PORTA_SRC8	INT_STAT_VIP1_MULT_PORTA_SRC7	INT_STAT_VIP1_MULT_PORTA_SRC6	INT_STAT_VIP1_MULT_PORTA_SRC5	INT_STAT_VIP1_MULT_PORTA_SRC4	INT_STAT_VIP1_MULT_PORTA_SRC3	INT_STAT_VIP1_MULT_PORTA_SRC2	INT_STAT_VIP1_MULT_PORTA_SRC1	INT_STAT_VIP1_MULT_PORTA_SRC0	RESERVED					

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
27	INT_STAT_VIP1_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP1_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_STAT_VIP1_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP1_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	INT_STAT_VIP1_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
6	INT_STAT_VIP1_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 10-186. Register Call Summary for Register VPE_INT0_CHANNEL1_INT_STAT

VPE Functional Description • VPDMA Interrupts: [0]
VPE Register Manual • VPE_VPDMA Register Summary: [1]

Table 10-187. VPE_INT0_CHANNEL1_INT_MASK

Address Offset	0x0000 004C		
Physical Address	0x489D D04C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_PORTB_SRC9	INT_MASK_VIP1_MULT_PORTB_SRC8	INT_MASK_VIP1_MULT_PORTB_SRC7	INT_MASK_VIP1_MULT_PORTB_SRC6	INT_MASK_VIP1_MULT_PORTB_SRC5	INT_MASK_VIP1_MULT_PORTB_SRC4	INT_MASK_VIP1_MULT_PORTB_SRC3	INT_MASK_VIP1_MULT_PORTB_SRC2	INT_MASK_VIP1_MULT_PORTB_SRC1	INT_MASK_VIP1_MULT_PORTB_SRC0	INT_MASK_VIP1_MULT_PORTA_SRC15	INT_MASK_VIP1_MULT_PORTA_SRC14	INT_MASK_VIP1_MULT_PORTA_SRC13	INT_MASK_VIP1_MULT_PORTA_SRC12	INT_MASK_VIP1_MULT_PORTA_SRC11	INT_MASK_VIP1_MULT_PORTA_SRC10	INT_MASK_VIP1_MULT_PORTA_SRC9	INT_MASK_VIP1_MULT_PORTA_SRC8	INT_MASK_VIP1_MULT_PORTA_SRC7	INT_MASK_VIP1_MULT_PORTA_SRC6	INT_MASK_VIP1_MULT_PORTA_SRC5	INT_MASK_VIP1_MULT_PORTA_SRC4	INT_MASK_VIP1_MULT_PORTA_SRC3	INT_MASK_VIP1_MULT_PORTA_SRC2	INT_MASK_VIP1_MULT_PORTA_SRC1	INT_MASK_VIP1_MULT_PORTA_SRC0	RESERVED					

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_PORTB_SRC9	The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_PORTB_SRC8	The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_PORTB_SRC7	The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
28	INT_MASK_VIP1_MULT_PORTB_SRC6	The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_PORTB_SRC5	The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_PORTB_SRC4	The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_PORTB_SRC3	The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_PORTB_SRC2	The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_PORTB_SRC1	The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_PORTB_SRC0	The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_PORTA_SRC15	The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_PORTA_SRC14	The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_PORTA_SRC13	The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_PORTA_SRC12	The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_PORTA_SRC11	The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_PORTA_SRC10	The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_PORTA_SRC9	The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_PORTA_SRC8	The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	INT_MASK_VIP1_MULT_PORTA_SRC7	The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_PORTA_SRC6	The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_PORTA_SRC5	The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_PORTA_SRC4	The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_PORTA_SRC3	The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_PORTA_SRC2	The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_PORTA_SRC1	The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_PORTA_SRC0	The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 10-188. Register Call Summary for Register VPE_INT0_CHANNEL1_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-189. VPE_INT0_CHANNEL2_INT_STAT

Address Offset	0x0000 0050		
Physical Address	0x489D D050	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP1_MULT_ANCB_SRC9	INT_STAT_VIP1_MULT_ANCB_SRC8	INT_STAT_VIP1_MULT_ANCB_SRC7	INT_STAT_VIP1_MULT_ANCB_SRC6	INT_STAT_VIP1_MULT_ANCB_SRC5	INT_STAT_VIP1_MULT_ANCB_SRC4	INT_STAT_VIP1_MULT_ANCB_SRC3	INT_STAT_VIP1_MULT_ANCB_SRC2	INT_STAT_VIP1_MULT_ANCB_SRC1	INT_STAT_VIP1_MULT_ANCB_SRC0	INT_STAT_VIP1_MULT_ANCA_SRC15	INT_STAT_VIP1_MULT_ANCA_SRC14	INT_STAT_VIP1_MULT_ANCA_SRC13	INT_STAT_VIP1_MULT_ANCA_SRC12	INT_STAT_VIP1_MULT_ANCA_SRC11	INT_STAT_VIP1_MULT_ANCA_SRC10	INT_STAT_VIP1_MULT_ANCA_SRC9	INT_STAT_VIP1_MULT_ANCA_SRC8	INT_STAT_VIP1_MULT_ANCA_SRC7	INT_STAT_VIP1_MULT_ANCA_SRC6	INT_STAT_VIP1_MULT_ANCA_SRC5	INT_STAT_VIP1_MULT_ANCA_SRC4	INT_STAT_VIP1_MULT_ANCA_SRC3	INT_STAT_VIP1_MULT_ANCA_SRC2	INT_STAT_VIP1_MULT_ANCA_SRC1	INT_STAT_VIP1_MULT_ANCA_SRC0	INT_STAT_VIP1_MULT_PORTB_SRC15	INT_STAT_VIP1_MULT_PORTB_SRC14	INT_STAT_VIP1_MULT_PORTB_SRC13	INT_STAT_VIP1_MULT_PORTB_SRC12	INT_STAT_VIP1_MULT_PORTB_SRC11	INT_STAT_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP1_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP1_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP1_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_VIP1_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-190. Register Call Summary for Register VPE_INT0_CHANNEL2_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-191. VPE_INT0_CHANNEL2_INT_MASK

Address Offset	0x0000 0054		
Physical Address	0x489D D054	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		

Table 10-191. VPE_INT0_CHANNEL2_INT_MASK (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_ANCB_SRC9	INT_MASK_VIP1_MULT_ANCB_SRC8	INT_MASK_VIP1_MULT_ANCB_SRC7	INT_MASK_VIP1_MULT_ANCB_SRC6	INT_MASK_VIP1_MULT_ANCB_SRC5	INT_MASK_VIP1_MULT_ANCB_SRC4	INT_MASK_VIP1_MULT_ANCB_SRC3	INT_MASK_VIP1_MULT_ANCB_SRC2	INT_MASK_VIP1_MULT_ANCB_SRC1	INT_MASK_VIP1_MULT_ANCB_SRC0	INT_MASK_VIP1_MULT_ANCA_SRC15	INT_MASK_VIP1_MULT_ANCA_SRC14	INT_MASK_VIP1_MULT_ANCA_SRC13	INT_MASK_VIP1_MULT_ANCA_SRC12	INT_MASK_VIP1_MULT_ANCA_SRC11	INT_MASK_VIP1_MULT_ANCA_SRC10	INT_MASK_VIP1_MULT_ANCA_SRC9	INT_MASK_VIP1_MULT_ANCA_SRC8	INT_MASK_VIP1_MULT_ANCA_SRC7	INT_MASK_VIP1_MULT_ANCA_SRC6	INT_MASK_VIP1_MULT_ANCA_SRC5	INT_MASK_VIP1_MULT_ANCA_SRC4	INT_MASK_VIP1_MULT_ANCA_SRC3	INT_MASK_VIP1_MULT_ANCA_SRC2	INT_MASK_VIP1_MULT_ANCA_SRC1	INT_MASK_VIP1_MULT_ANCA_SRC0	INT_MASK_VIP1_MULT_PORTB_SRC15	INT_MASK_VIP1_MULT_PORTB_SRC14	INT_MASK_VIP1_MULT_PORTB_SRC13	INT_MASK_VIP1_MULT_PORTB_SRC12	INT_MASK_VIP1_MULT_PORTB_SRC11	INT_MASK_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_ANCB_SRC9	The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_ANCB_SRC8	The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_ANCB_SRC7	The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_ANCB_SRC6	The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_ANCB_SRC5	The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_ANCB_SRC4	The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_ANCB_SRC3	The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_ANCB_SRC2	The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_ANCB_SRC1	The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_MASK_VIP1_MULT_ANCB_SRC0	The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_ANCA_SRC15	The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_ANCA_SRC14	The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_ANCA_SRC13	The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_ANCA_SRC12	The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_ANCA_SRC11	The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_ANCA_SRC10	The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_ANCA_SRC9	The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_ANCA_SRC8	The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_ANCA_SRC7	The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_ANCA_SRC6	The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_ANCA_SRC5	The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_ANCA_SRC4	The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_ANCA_SRC3	The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_ANCA_SRC2	The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_MASK_VIP1_MULT_ANCA_SRC1	The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_ANCA_SRC0	The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_PORTB_SRC15	The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_PORTB_SRC14	The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_PORTB_SRC13	The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_PORTB_SRC12	The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_PORTB_SRC11	The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_PORTB_SRC10	The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-192. Register Call Summary for Register VPE_INT0_CHANNEL2_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-193. VPE_INT0_CHANNEL3_INT_STAT

Address Offset	0x0000 0058		
Physical Address	0x489D D058	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP2_MULT_PORTB_SRC3	INT_STAT_VIP2_MULT_PORTB_SRC2	INT_STAT_VIP2_MULT_PORTB_SRC1	INT_STAT_VIP2_MULT_PORTB_SRC0	INT_STAT_VIP2_MULT_PORTA_SRC15	INT_STAT_VIP2_MULT_PORTA_SRC14	INT_STAT_VIP2_MULT_PORTA_SRC13	INT_STAT_VIP2_MULT_PORTA_SRC12	INT_STAT_VIP2_MULT_PORTA_SRC11	INT_STAT_VIP2_MULT_PORTA_SRC10	INT_STAT_VIP2_MULT_PORTA_SRC9	INT_STAT_VIP2_MULT_PORTA_SRC8	INT_STAT_VIP2_MULT_PORTA_SRC7	INT_STAT_VIP2_MULT_PORTA_SRC6	INT_STAT_VIP2_MULT_PORTA_SRC5	INT_STAT_VIP2_MULT_PORTA_SRC4	INT_STAT_VIP2_MULT_PORTA_SRC3	INT_STAT_VIP2_MULT_PORTA_SRC2	INT_STAT_VIP2_MULT_PORTA_SRC1	INT_STAT_VIP2_MULT_PORTA_SRC0	INT_STAT_VIP1_PORTB_RGB	INT_STAT_VIP1_PORTA_RGB	INT_STAT_VIP1_PORTB_CHROMA	INT_STAT_VIP1_PORTB_LUMA	INT_STAT_VIP1_PORTA_CHROMA	INT_STAT_VIP1_PORTA_LUMA	INT_STAT_VIP1_MULT_ANCB_SRC15	INT_STAT_VIP1_MULT_ANCB_SRC14	INT_STAT_VIP1_MULT_ANCB_SRC13	INT_STAT_VIP1_MULT_ANCB_SRC12	INT_STAT_VIP1_MULT_ANCB_SRC11	INT_STAT_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP2_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP2_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP2_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_PORTB_RGB	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_PORTA_RGB	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_PORTB_CHROMA	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_PORTB_LUMA	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_PORTA_CHROMA	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_PORTA_LUMA	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_VIP1_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-194. Register Call Summary for Register VPE_INT0_CHANNEL3_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-195. VPE_INT0_CHANNEL3_INT_MASK

Address Offset	0x0000 005C		
Physical Address	0x489D D05C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		

Table 10-195. VPE_INT0_CHANNEL3_INT_MASK (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_PORTB_SRC3	INT_MASK_VIP2_MULT_PORTB_SRC2	INT_MASK_VIP2_MULT_PORTB_SRC1	INT_MASK_VIP2_MULT_PORTB_SRC0	INT_MASK_VIP2_MULT_PORTA_SRC15	INT_MASK_VIP2_MULT_PORTA_SRC14	INT_MASK_VIP2_MULT_PORTA_SRC13	INT_MASK_VIP2_MULT_PORTA_SRC12	INT_MASK_VIP2_MULT_PORTA_SRC11	INT_MASK_VIP2_MULT_PORTA_SRC10	INT_MASK_VIP2_MULT_PORTA_SRC9	INT_MASK_VIP2_MULT_PORTA_SRC8	INT_MASK_VIP2_MULT_PORTA_SRC7	INT_MASK_VIP2_MULT_PORTA_SRC6	INT_MASK_VIP2_MULT_PORTA_SRC5	INT_MASK_VIP2_MULT_PORTA_SRC4	INT_MASK_VIP2_MULT_PORTA_SRC3	INT_MASK_VIP2_MULT_PORTA_SRC2	INT_MASK_VIP2_MULT_PORTA_SRC1	INT_MASK_VIP2_MULT_PORTA_SRC0	INT_MASK_VIP1_PORTB_RGB	INT_MASK_VIP1_PORTA_RGB	INT_MASK_VIP1_PORTB_CHROMA	INT_MASK_VIP1_PORTB_LUMA	INT_MASK_VIP1_PORTA_CHROMA	INT_MASK_VIP1_PORTA_LUMA	INT_MASK_VIP1_MULT_ANCB_SRC15	INT_MASK_VIP1_MULT_ANCB_SRC14	INT_MASK_VIP1_MULT_ANCB_SRC13	INT_MASK_VIP1_MULT_ANCB_SRC12	INT_MASK_VIP1_MULT_ANCB_SRC11	INT_MASK_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_PORTB_SRC3	The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_PORTB_SRC2	The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_PORTB_SRC1	The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_PORTB_SRC0	The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_PORTA_SRC15	The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_PORTA_SRC14	The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_PORTA_SRC13	The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_PORTA_SRC12	The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_PORTA_SRC11	The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_MASK_VIP2_MULT_PORTA_SRC10	The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_PORTA_SRC9	The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_PORTA_SRC8	The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_PORTA_SRC7	The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_PORTA_SRC6	The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_PORTA_SRC5	The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_PORTA_SRC4	The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_PORTA_SRC3	The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_PORTA_SRC2	The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_PORTA_SRC1	The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_PORTA_SRC0	The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_PORTB_RGB	The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_PORTA_RGB	The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_MASK_VIP1_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_ANCB_SRC15	The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_ANCB_SRC14	The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_ANCB_SRC13	The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_ANCB_SRC12	The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_ANCB_SRC11	The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_ANCB_SRC10	The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-196. Register Call Summary for Register VPE_INT0_CHANNEL3_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-197. VPE_INT0_CHANNEL4_INT_STAT

Address Offset	0x0000 0060		
Physical Address	0x489D D060	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_VIP2_MULT_ANCB_SRC3	INT_STAT_VIP2_MULT_ANCB_SRC2	INT_STAT_VIP2_MULT_ANCB_SRC1	INT_STAT_VIP2_MULT_ANCB_SRC0	INT_STAT_VIP2_MULT_ANCA_SRC15	INT_STAT_VIP2_MULT_ANCA_SRC14	INT_STAT_VIP2_MULT_ANCA_SRC13	INT_STAT_VIP2_MULT_ANCA_SRC12	INT_STAT_VIP2_MULT_ANCA_SRC11	INT_STAT_VIP2_MULT_ANCA_SRC10	INT_STAT_VIP2_MULT_ANCA_SRC9	INT_STAT_VIP2_MULT_ANCA_SRC8	INT_STAT_VIP2_MULT_ANCA_SRC7	INT_STAT_VIP2_MULT_ANCA_SRC6	INT_STAT_VIP2_MULT_ANCA_SRC5	INT_STAT_VIP2_MULT_ANCA_SRC4	INT_STAT_VIP2_MULT_ANCA_SRC3	INT_STAT_VIP2_MULT_ANCA_SRC2	INT_STAT_VIP2_MULT_ANCA_SRC1	INT_STAT_VIP2_MULT_ANCA_SRC0	INT_STAT_VIP2_MULT_PORTB_SRC15	INT_STAT_VIP2_MULT_PORTB_SRC14	INT_STAT_VIP2_MULT_PORTB_SRC13	INT_STAT_VIP2_MULT_PORTB_SRC12	INT_STAT_VIP2_MULT_PORTB_SRC11	INT_STAT_VIP2_MULT_PORTB_SRC10	INT_STAT_VIP2_MULT_PORTB_SRC9	INT_STAT_VIP2_MULT_PORTB_SRC8	INT_STAT_VIP2_MULT_PORTB_SRC7	INT_STAT_VIP2_MULT_PORTB_SRC6	INT_STAT_VIP2_MULT_PORTB_SRC5	INT_STAT_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP2_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP2_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP2_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_VIP2_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-198. Register Call Summary for Register VPE_INT0_CHANNEL4_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-199. VPE_INT0_CHANNEL4_INT_MASK

Address Offset	0x0000 0064		
Physical Address	0x489D D064	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		

Table 10-199. VPE_INT0_CHANNEL4_INT_MASK (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_ANCB_SRC3	INT_MASK_VIP2_MULT_ANCB_SRC2	INT_MASK_VIP2_MULT_ANCB_SRC1	INT_MASK_VIP2_MULT_ANCB_SRC0	INT_MASK_VIP2_MULT_ANCA_SRC15	INT_MASK_VIP2_MULT_ANCA_SRC14	INT_MASK_VIP2_MULT_ANCA_SRC13	INT_MASK_VIP2_MULT_ANCA_SRC12	INT_MASK_VIP2_MULT_ANCA_SRC11	INT_MASK_VIP2_MULT_ANCA_SRC10	INT_MASK_VIP2_MULT_ANCA_SRC9	INT_MASK_VIP2_MULT_ANCA_SRC8	INT_MASK_VIP2_MULT_ANCA_SRC7	INT_MASK_VIP2_MULT_ANCA_SRC6	INT_MASK_VIP2_MULT_ANCA_SRC5	INT_MASK_VIP2_MULT_ANCA_SRC4	INT_MASK_VIP2_MULT_ANCA_SRC3	INT_MASK_VIP2_MULT_ANCA_SRC2	INT_MASK_VIP2_MULT_ANCA_SRC1	INT_MASK_VIP2_MULT_ANCA_SRC0	INT_MASK_VIP2_MULT_PORTB_SRC15	INT_MASK_VIP2_MULT_PORTB_SRC14	INT_MASK_VIP2_MULT_PORTB_SRC13	INT_MASK_VIP2_MULT_PORTB_SRC12	INT_MASK_VIP2_MULT_PORTB_SRC11	INT_MASK_VIP2_MULT_PORTB_SRC10	INT_MASK_VIP2_MULT_PORTB_SRC9	INT_MASK_VIP2_MULT_PORTB_SRC8	INT_MASK_VIP2_MULT_PORTB_SRC7	INT_MASK_VIP2_MULT_PORTB_SRC6	INT_MASK_VIP2_MULT_PORTB_SRC5	INT_MASK_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_ANCB_SRC3	The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_ANCB_SRC2	The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_ANCB_SRC1	The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_ANCB_SRC0	The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_ANCA_SRC15	The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_ANCA_SRC14	The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_ANCA_SRC13	The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_ANCA_SRC12	The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_ANCA_SRC11	The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_MASK_VIP2_MULT_ANCA_SRC10	The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_ANCA_SRC9	The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_ANCA_SRC8	The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_ANCA_SRC7	The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_ANCA_SRC6	The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_ANCA_SRC5	The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_ANCA_SRC4	The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_ANCA_SRC3	The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_ANCA_SRC2	The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_ANCA_SRC1	The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_ANCA_SRC0	The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_PORTB_SRC15	The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_PORTB_SRC14	The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_PORTB_SRC13	The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_PORTB_SRC12	The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_MASK_VIP2_MULT_PORTB_SRC11	The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_PORTB_SRC10	The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_PORTB_SRC9	The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_PORTB_SRC8	The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_PORTB_SRC7	The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_PORTB_SRC6	The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_PORTB_SRC5	The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_PORTB_SRC4	The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-200. Register Call Summary for Register VPE_INT0_CHANNEL4_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-201. VPE_INT0_CHANNEL5_INT_STAT

Address Offset	0x0000 0068		
Physical Address	0x489D D068	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_STAT_TRANSCODE2_CHROMA	INT_STAT_TRANSCODE2_LUMA	INT_STAT_TRANSCODE1_CHROMA	INT_STAT_TRANSCODE1_LUMA	INT_STAT_AUX_IN	INT_STAT_PIP_FRAME	INT_STAT_POST_COMP_WR	INT_STAT_VBI_SD_VENC	RESERVED	INT_STAT_NF_LAST_CHROMA	INT_STAT_NF_LAST_LUMA	INT_STAT_NF_WRITE_CHROMA	INT_STAT_NF_WRITE_LUMA	INT_STAT_OTHER	INT_STAT_VIP2_PORTB_RGB	INT_STAT_VIP2_PORTA_RGB	INT_STAT_VIP2_PORTB_CHROMA	INT_STAT_VIP2_PORTB_LUMA	INT_STAT_VIP2_PORTA_CHROMA	INT_STAT_VIP2_PORTA_LUMA	INT_STAT_VIP2_MULT_ANCB_SRC15	INT_STAT_VIP2_MULT_ANCB_SRC14	INT_STAT_VIP2_MULT_ANCB_SRC13	INT_STAT_VIP2_MULT_ANCB_SRC12	INT_STAT_VIP2_MULT_ANCB_SRC11	INT_STAT_VIP2_MULT_ANCB_SRC10	INT_STAT_VIP2_MULT_ANCB_SRC9	INT_STAT_VIP2_MULT_ANCB_SRC8	INT_STAT_VIP2_MULT_ANCB_SRC7	INT_STAT_VIP2_MULT_ANCB_SRC6	INT_STAT_VIP2_MULT_ANCB_SRC5	INT_STAT_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_TRANSCODE2_CHROMA	The last write DMA transaction has completed for channel transcode2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_TRANSCODE2_LUMA	The last write DMA transaction has completed for channel transcode2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_TRANSCODE1_CHROMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_TRANSCODE1_LUMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_AUX_IN	The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_PIP_FRAME	The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_POST_COMP_WR	The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VBI_SD_VENC	The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_STAT_NF_LAST_CHROMA	The last write DMA transaction has completed for channel nf_last_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_NF_LAST_LUMA	The last write DMA transaction has completed for channel nf_last_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_NF_WRITE_CHROMA	The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_NF_WRITE_LUMA	The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_OTHER	This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_VIP2_PORTB_RGB	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_PORTA_RGB	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_PORTB_CHROMA	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_PORTB_LUMA	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_PORTA_CHROMA	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_PORTA_LUMA	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
10	INT_STAT_VIP2_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_STAT_VIP2_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-202. Register Call Summary for Register VPE_INT0_CHANNEL5_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-203. VPE_INT0_CHANNEL5_INT_MASK

Address Offset	0x0000 006C		
Physical Address	0x489D D06C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_OTHER	INT_MASK_VIP2_PORTB_RGB	INT_MASK_VIP2_PORTA_RGB	INT_MASK_VIP2_PORTB_CHROMA	INT_MASK_VIP2_PORTB_LUMA	INT_MASK_VIP2_PORTA_CHROMA	INT_MASK_VIP2_PORTA_LUMA	INT_MASK_VIP2_MULT_ANCB_SRC15	INT_MASK_VIP2_MULT_ANCB_SRC14	INT_MASK_VIP2_MULT_ANCB_SRC13	INT_MASK_VIP2_MULT_ANCB_SRC12	INT_MASK_VIP2_MULT_ANCB_SRC11	INT_MASK_VIP2_MULT_ANCB_SRC10	INT_MASK_VIP2_MULT_ANCB_SRC9	INT_MASK_VIP2_MULT_ANCB_SRC8	INT_MASK_VIP2_MULT_ANCB_SRC7	INT_MASK_VIP2_MULT_ANCB_SRC6	INT_MASK_VIP2_MULT_ANCB_SRC5	INT_MASK_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_MASK_OTHER	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_ANCB_SRC15	The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_ANCB_SRC14	The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_ANCB_SRC13	The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_ANCB_SRC12	The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_ANCB_SRC11	The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_ANCB_SRC10	The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_ANCB_SRC9	The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_ANCB_SRC8	The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_ANCB_SRC7	The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
2	INT_MASK_VIP2_MULT_ANCB_SRC6	The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_ANCB_SRC5	The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_ANCB_SRC4	The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-204. Register Call Summary for Register VPE_INT0_CHANNEL5_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-205. VPE_INT0_CLIENT0_INT_STAT

Address Offset	0x0000 0078		
Physical Address	0x489D D078	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX1_DATA	INT_STAT_COMP_WRBK	INT_STAT_SC_OUT	RESERVED								INT_STAT_SC_IN_LUMA	INT_STAT_SC_IN_CHROMA	INT_STAT_PIP_WRBK	INT_STAT_DEI_SC_OUT	RESERVED	INT_STAT_DEI_HQ_MV_OUT	RESERVED		INT_STAT_DEI_HQ_MV_IN	RESERVED						INT_STAT_DEI_HQ_3_CHROMA	INT_STAT_DEI_HQ_3_LUMA	INT_STAT_DEI_HQ_2_CHROMA	INT_STAT_DEI_HQ_2_LUMA	INT_STAT_DEI_HQ_1_LUMA	INT_STAT_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX1_DATA	The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	INT_STAT_COMP_WRBK	The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_SC_OUT	The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_STAT_SC_IN_LUMA	The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_SC_IN_CHROMA	The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_PIP_WRBK	The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_DEI_SC_OUT	The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_DEI_HQ_MV_OUT	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_DEI_HQ_MV_IN	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_DEI_HQ_3_CHROMA	The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_DEI_HQ_3_LUMA	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_DEI_HQ_2_CHROMA	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_DEI_HQ_2_LUMA	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_DEI_HQ_1_LUMA	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_DEI_HQ_1_CHROMA	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-206. Register Call Summary for Register VPE_INT0_CLIENT0_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0]

Table 10-206. Register Call Summary for Register VPE_INT0_CLIENT0_INT_STAT (continued)

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-207. VPE_INT0_CLIENT0_INT_MASK

Address Offset	0x0000 007C		
Physical Address	0x489D D07C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX1_DATA	INT_MASK_COMP_WRBK	INT_MASK_SC_OUT	RESERVED								INT_MASK_SC_IN_LUMA	INT_MASK_SC_IN_CHROMA	INT_MASK_PIP_WRBK	INT_MASK_DEI_SC_OUT	RESERVED	INT_MASK_DEI_HQ_MV_OUT	RESERVED		INT_MASK_DEI_HQ_MV_IN	RESERVED						INT_MASK_DEI_HQ_3_CHROMA	INT_MASK_DEI_HQ_3_LUMA	INT_MASK_DEI_HQ_2_CHROMA	INT_MASK_DEI_HQ_2_LUMA	INT_MASK_DEI_HQ_1_LUMA	INT_MASK_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX1_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_COMP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_MASK_SC_IN_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_SC_IN_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_PIP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_DEI_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_DEI_HQ_MV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12	INT_MASK_DEI_HQ_MV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_DEI_HQ_3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_DEI_HQ_3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_DEI_HQ_2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_DEI_HQ_2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_DEI_HQ_1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_DEI_HQ_1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-208. Register Call Summary for Register VPE_INT0_CLIENT0_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-209. VPE_INT0_CLIENT1_INT_STAT

Address Offset	0x0000 0080		
Physical Address	0x489D D080	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24
RESERVED	RESERVED	INT_STAT_VIP2_ANC_B	INT_STAT_VIP2_ANC_A	INT_STAT_VIP1_ANC_B	INT_STAT_VIP1_ANC_A	INT_STAT_TRANS2_LUMA	INT_STAT_TRANS2_CHROMA

23	22	21	20	19	18	17	16
INT_STAT_TRANS1_LUMA	INT_STAT_TRANS1_CHROMA	INT_STAT_HDMI_WRBK_OUT	INT_STAT_VPI_CTL	INT_STAT_VBI_SDVENC	RESERVED	INT_STAT_NF_420_UV_OUT	INT_STAT_NF_420_Y_OUT

15	14	13	12	11	10	9	8
INT_STAT_NF_420_UV_IN	INT_STAT_NF_420_Y_IN	INT_STAT_NF_422_IN	INT_STAT_GRPX3_ST	INT_STAT_GRPX2_ST	INT_STAT_GRPX1_ST	INT_STAT_VIP2_UP_UV	INT_STAT_VIP2_UP_Y

7	6	5	4	3	2	1	0
INT_STAT_VIP2_LO_UV	INT_STAT_VIP2_LO_Y	INT_STAT_VIP1_UP_UV	INT_STAT_VIP1_UP_Y	INT_STAT_VIP1_LO_UV	INT_STAT_VIP1_LO_Y	INT_STAT_GRPX3_DATA	INT_STAT_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
29	INT_STAT_VIP2_ANC_B	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
28	INT_STAT_VIP2_ANC_A	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
27	INT_STAT_VIP1_ANC_B	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
26	INT_STAT_VIP1_ANC_A	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
25	INT_STAT_TRANS2_LUMA	The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
24	INT_STAT_TRANS2_CHROMA	The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
23	INT_STAT_TRANS1_LUMA	The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_TRANS1_CHROMA	The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
21	INT_STAT_HDMI_WRBK_OUT	The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
20	INT_STAT_VPI_CTL	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
19	INT_STAT_VBI_SDVENC	The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
18	RESERVED	Reserved	R	0
17	INT_STAT_NF_420_UV_OUT	The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
16	INT_STAT_NF_420_Y_OUT	The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
15	INT_STAT_NF_420_UV_IN	The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
14	INT_STAT_NF_420_Y_IN	The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
13	INT_STAT_NF_422_IN	The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
12	INT_STAT_GRPX3_ST	The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
11	INT_STAT_GRPX2_ST	The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
10	INT_STAT_GRPX1_ST	The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
9	INT_STAT_VIP2_UP_UV	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
8	INT_STAT_VIP2_UP_Y	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
7	INT_STAT_VIP2_LO_UV	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
6	INT_STAT_VIP2_LO_Y	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
5	INT_STAT_VIP1_UP_UV	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
4	INT_STAT_VIP1_UP_Y	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
3	INT_STAT_VIP1_LO_UV	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
2	INT_STAT_VIP1_LO_Y	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
1	INT_STAT_GRPX3_DATA	The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_GRPX2_DATA	The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Table 10-210. Register Call Summary for Register VPE_INT0_CLIENT1_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-211. VPE_INT0_CLIENT1_INT_MASK

Address Offset	0x0000 0084		
Physical Address	0x489D D084	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED	RESERVED	INT_MASK_VIP2_ANC_B	INT_MASK_VIP2_ANC_A	INT_MASK_VIP1_ANC_B	INT_MASK_VIP1_ANC_A	INT_MASK_TRANS2_LUMA	INT_MASK_TRANS2_CHROMA	INT_MASK_TRANS1_LUMA	INT_MASK_TRANS1_CHROMA	INT_MASK_HDMI_WRBK_OUT	INT_MASK_VPI_CTL	INT_MASK_VBI_SDVENC	RESERVED	INT_MASK_NF_420_UV_OUT	INT_MASK_NF_420_Y_OUT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_NF_420_UV_IN	INT_MASK_NF_420_Y_IN	INT_MASK_NF_422_IN	INT_MASK_GRPX3_ST	INT_MASK_GRPX2_ST	INT_MASK_GRPX1_ST	INT_MASK_VIP2_UP_UV	INT_MASK_VIP2_UP_Y	INT_MASK_VIP2_LO_UV	INT_MASK_VIP2_LO_Y	INT_MASK_VIP1_UP_UV	INT_MASK_VIP1_UP_Y	INT_MASK_VIP1_LO_UV	INT_MASK_VIP1_LO_Y	INT_MASK_GRPX3_DATA	INT_MASK_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_MASK_VIP2_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
28	INT_MASK_VIP2_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
27	INT_MASK_VIP1_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
26	INT_MASK_VIP1_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
25	INT_MASK_TRANS2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
24	INT_MASK_TRANS2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
23	INT_MASK_TRANS1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
22	INT_MASK_TRANS1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
21	INT_MASK_HDMI_WRBK_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
20	INT_MASK_VPI_CTL	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
19	INT_MASK_VBI_SDVENC	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
18	RESERVED	Reserved	R	0
17	INT_MASK_NF_420_UV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
16	INT_MASK_NF_420_Y_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
15	INT_MASK_NF_420_UV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
14	INT_MASK_NF_420_Y_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
13	INT_MASK_NF_422_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
12	INT_MASK_GRPX3_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
11	INT_MASK_GRPX2_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
10	INT_MASK_GRPX1_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
9	INT_MASK_VIP2_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
8	INT_MASK_VIP2_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
7	INT_MASK_VIP2_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
6	INT_MASK_VIP2_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
5	INT_MASK_VIP1_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
4	INT_MASK_VIP1_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
3	INT_MASK_VIP1_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
2	INT_MASK_VIP1_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
1	INT_MASK_GRPX3_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
0	INT_MASK_GRPX2_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Table 10-212. Register Call Summary for Register VPE_INT0_CLIENT1_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-213. VPE_INT0_LIST0_INT_STAT

Address Offset	0x0000 0088		
Physical Address	0x489D D088	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT_STAT_CONTROL_DESCRIPTOR_INT15	INT_STAT_CONTROL_DESCRIPTOR_INT14	INT_STAT_CONTROL_DESCRIPTOR_INT13	INT_STAT_CONTROL_DESCRIPTOR_INT12	INT_STAT_CONTROL_DESCRIPTOR_INT11	INT_STAT_CONTROL_DESCRIPTOR_INT10	INT_STAT_CONTROL_DESCRIPTOR_INT9	INT_STAT_CONTROL_DESCRIPTOR_INT8	INT_STAT_CONTROL_DESCRIPTOR_INT7	INT_STAT_CONTROL_DESCRIPTOR_INT6	INT_STAT_CONTROL_DESCRIPTOR_INT5	INT_STAT_CONTROL_DESCRIPTOR_INT4	INT_STAT_CONTROL_DESCRIPTOR_INT3	INT_STAT_CONTROL_DESCRIPTOR_INT2	INT_STAT_CONTROL_DESCRIPTOR_INT1	INT_STAT_CONTROL_DESCRIPTOR_INT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_LIST7_NOTIFY	INT_STAT_LIST7_COMPLETE	INT_STAT_LIST6_NOTIFY	INT_STAT_LIST6_COMPLETE	INT_STAT_LIST5_NOTIFY	INT_STAT_LIST5_COMPLETE	INT_STAT_LIST4_NOTIFY	INT_STAT_LIST4_COMPLETE	INT_STAT_LIST3_NOTIFY	INT_STAT_LIST3_COMPLETE	INT_STAT_LIST2_NOTIFY	INT_STAT_LIST2_COMPLETE	INT_STAT_LIST1_NOTIFY	INT_STAT_LIST1_COMPLETE	INT_STAT_LIST0_NOTIFY	INT_STAT_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_STAT_CONTROL_DESCRIPTOR_INT15	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	INT_STAT_CONTROL_DESCRIPTOR_INT14	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_CONTROL_DESCRIPTOR_INT13	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_CONTROL_DESCRIPTOR_INT12	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_CONTROL_DESCRIPTOR_INT11	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_CONTROL_DESCRIPTOR_INT10	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_CONTROL_DESCRIPTOR_INT9	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_CONTROL_DESCRIPTOR_INT8	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_CONTROL_DESCRIPTOR_INT7	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_CONTROL_DESCRIPTOR_INT6	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_CONTROL_DESCRIPTOR_INT5	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_CONTROL_DESCRIPTOR_INT4	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_CONTROL_DESCRIPTOR_INT3	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_STAT_CONTROL_DESCRIPTOR_INT2	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_CONTROL_DESCRIPTOR_INT1	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_CONTROL_DESCRIPTOR_INT0	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_LIST7_NOTIFY	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_LIST7_COMPLETE	List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_LIST6_NOTIFY	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_LIST6_COMPLETE	List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_LIST5_NOTIFY	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_LIST5_COMPLETE	List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_LIST4_NOTIFY	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_LIST4_COMPLETE	List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_LIST3_NOTIFY	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_LIST3_COMPLETE	List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	INT_STAT_LIST2_NOTIFY	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_LIST2_COMPLETE	List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_LIST1_NOTIFY	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_LIST1_COMPLETE	List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_LIST0_NOTIFY	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_LIST0_COMPLETE	List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-214. Register Call Summary for Register VPE_INT0_LIST0_INT_STAT

VPE Functional Description

- VPDMA Interrupts: [0] [1] [2]

VPE Register Manual

- VPE_VPDMA Register Summary: [3]

Table 10-215. VPE_INT0_LIST0_INT_MASK

Address Offset	0x0000 008C		
Physical Address	0x489D D08C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT_MASK_CONTROL_DESCRIPTOR_INT15	INT_MASK_CONTROL_DESCRIPTOR_INT14	INT_MASK_CONTROL_DESCRIPTOR_INT13	INT_MASK_CONTROL_DESCRIPTOR_INT12	INT_MASK_CONTROL_DESCRIPTOR_INT11	INT_MASK_CONTROL_DESCRIPTOR_INT10	INT_MASK_CONTROL_DESCRIPTOR_INT9	INT_MASK_CONTROL_DESCRIPTOR_INT8	INT_MASK_CONTROL_DESCRIPTOR_INT7	INT_MASK_CONTROL_DESCRIPTOR_INT6	INT_MASK_CONTROL_DESCRIPTOR_INT5	INT_MASK_CONTROL_DESCRIPTOR_INT4	INT_MASK_CONTROL_DESCRIPTOR_INT3	INT_MASK_CONTROL_DESCRIPTOR_INT2	INT_MASK_CONTROL_DESCRIPTOR_INT1	INT_MASK_CONTROL_DESCRIPTOR_INT0	INT_MASK_LIST7_NOTIFY	INT_MASK_LIST7_COMPLETE	INT_MASK_LIST6_NOTIFY	INT_MASK_LIST6_COMPLETE	INT_MASK_LIST5_NOTIFY	INT_MASK_LIST5_COMPLETE	INT_MASK_LIST4_NOTIFY	INT_MASK_LIST4_COMPLETE	INT_MASK_LIST3_NOTIFY	INT_MASK_LIST3_COMPLETE	INT_MASK_LIST2_NOTIFY	INT_MASK_LIST2_COMPLETE	INT_MASK_LIST1_NOTIFY	INT_MASK_LIST1_COMPLETE	INT_MASK_LIST0_NOTIFY	INT_MASK_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_MASK_CONTROL_DESCRIPTOR_INT15	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_CONTROL_DESCRIPTOR_INT14	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_CONTROL_DESCRIPTOR_INT13	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_CONTROL_DESCRIPTOR_INT12	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_CONTROL_DESCRIPTOR_INT11	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_CONTROL_DESCRIPTOR_INT10	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_CONTROL_DESCRIPTOR_INT9	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_CONTROL_DESCRIPTOR_INT8	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_CONTROL_DESCRIPTOR_INT7	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_CONTROL_DESCRIPTOR_INT6	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_CONTROL_DESCRIPTOR_INT5	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_CONTROL_DESCRIPTOR_INT4	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_CONTROL_DESCRIPTOR_INT3	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_MASK_CONTROL_DESCRIPTOR_INT2	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_CONTROL_DESCRIPTOR_INT1	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_CONTROL_DESCRIPTOR_INT0	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_LIST7_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_LIST7_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_LIST6_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_LIST6_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_LIST5_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_LIST5_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_LIST4_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_LIST4_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_LIST3_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_LIST3_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_LIST2_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_LIST2_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_LIST1_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_LIST1_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_LIST0_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_LIST0_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-216. Register Call Summary for Register VPE_INT0_LIST0_INT_MASK

VPE Functional Description

- VPDMA Interrupts: [0]

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-217. VPE_PERF_MON0

Address Offset	0x0000 0200		
Physical Address	0x489D D200	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_1_chroma 0x1 : vpi_ctl 0x2: dei_hq_1_luma 0x3: dei_hq_2_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_1_chroma 0x1 : vpi_ctl 0x2: dei_hq_1_luma 0x3: dei_hq_2_luma	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-218. Register Call Summary for Register VPE_PERF_MON0

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-219. VPE_PERF_MON1

Address Offset	0x0000 0204		
Physical Address	0x489D D204	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_1_luma 0x1 : dei_hq_1_chroma 0x2: dei_hq_2_luma 0x3: dei_hq_2_chroma	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_1_luma 0x1 : dei_hq_1_chroma 0x2: dei_hq_2_luma 0x3: dei_hq_2_chroma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-220. Register Call Summary for Register VPE_PERF_MON1

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-221. VPE_PERF_MON2

Address Offset	0x0000 0208		
Physical Address	0x489D D208	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-222. Register Call Summary for Register VPE_PERF_MON2

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-223. VPE_PERF_MON3

Address Offset	0x0000 020C

Table 10-223. VPE_PERF_MON3 (continued)

Physical Address	0x489D D20C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-224. Register Call Summary for Register VPE_PERF_MON3

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-225. VPE_PERF_MON4

Address Offset	0x0000 0210		
Physical Address	0x489D D210	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-226. Register Call Summary for Register VPE_PERF_MON4

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-227. VPE_PERF_MON5

Address Offset	0x0000 0214		
Physical Address	0x489D D214	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-228. Register Call Summary for Register VPE_PERF_MON5

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-229. VPE_PERF_MON6

Address Offset	0x0000 0218

Table 10-229. VPE_PERF_MON6 (continued)

Physical Address	0x489D D218	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : dei_hq_3_chroma 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_3_chroma 0x1 : dei_hq_3_chroma 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-230. Register Call Summary for Register VPE_PERF_MON6

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-231. VPE_PERF_MON7

Address Offset	0x0000 021C		
Physical Address	0x489D D21C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-232. Register Call Summary for Register VPE_PERF_MON7

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-233. VPE_PERF_MON8

Address Offset	0x0000 0220		
Physical Address	0x489D D220	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-234. Register Call Summary for Register VPE_PERF_MON8

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-235. VPE_PERF_MON9

Address Offset	0x0000 0224

Table 10-235. VPE_PERF_MON9 (continued)

Physical Address	0x489D D224	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-236. Register Call Summary for Register VPE_PERF_MON9

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-237. VPE_PERF_MON10

Address Offset	0x0000 0228		
Physical Address	0x489D D228	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: dei_hq_mv_in	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: dei_hq_mv_in	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-238. Register Call Summary for Register VPE_PERF_MON10

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-239. VPE_PERF_MON11

Address Offset	0x0000 022C		
Physical Address	0x489D D22C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_in 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_in 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-240. Register Call Summary for Register VPE_PERF_MON11

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-241. VPE_PERF_MON12

Address Offset	0x0000 0230

Table 10-241. VPE_PERF_MON12 (continued)

Physical Address	0x489D D230	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_mv_in 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_mv_in 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-242. Register Call Summary for Register VPE_PERF_MON12

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-243. VPE_PERF_MON13

Address Offset	0x0000 0234		
Physical Address	0x489D D234	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_in 0x2: 0x3:dei_hq_mv_out	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_in 0x2: 0x3:dei_hq_mv_out	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-244. Register Call Summary for Register VPE_PERF_MON13

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-245. VPE_PERF_MON14

Address Offset	0x0000 0238		
Physical Address	0x489D D238	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_out 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_out 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-246. Register Call Summary for Register VPE_PERF_MON14

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-247. VPE_PERF_MON15

Address Offset	0x0000 023C

Table 10-247. VPE_PERF_MON15 (continued)

Physical Address	0x489D D23C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_mv_out 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_mv_out 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-248. Register Call Summary for Register VPE_PERF_MON15

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-249. VPE_PERF_MON16

Address Offset	0x0000 0240		
Physical Address	0x489D D240	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_out 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_out 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-250. Register Call Summary for Register VPE_PERF_MON16

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-251. VPE_PERF_MON17

Address Offset	0x0000 0244		
Physical Address	0x489D D244	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-252. Register Call Summary for Register VPE_PERF_MON17

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-253. VPE_PERF_MON18

Address Offset	0x0000 0248

Table 10-253. VPE_PERF_MON18 (continued)

Physical Address	0x489D D248	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-254. Register Call Summary for Register VPE_PERF_MON18

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-255. VPE_PERF_MON19

Address Offset	0x0000 024C		
Physical Address	0x489D D24C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-256. Register Call Summary for Register VPE_PERF_MON19

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-257. VPE_PERF_MON20

Address Offset	0x0000 0250		
Physical Address	0x489D D250	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-258. Register Call Summary for Register VPE_PERF_MON20

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-259. VPE_PERF_MON21

Address Offset	0x0000 0254

Table 10-259. VPE_PERF_MON21 (continued)

Physical Address	0x489D D254	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-260. Register Call Summary for Register VPE_PERF_MON21

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-261. VPE_PERF_MON22

Address Offset	0x0000 0258		
Physical Address	0x489D D258	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-262. Register Call Summary for Register VPE_PERF_MON22

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-263. VPE_PERF_MON23

Address Offset	0x0000 025C		
Physical Address	0x489D D25C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-264. Register Call Summary for Register VPE_PERF_MON23

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-265. VPE_PERF_MON24

Address Offset	0x0000 0260

Table 10-265. VPE_PERF_MON24 (continued)

Physical Address	0x489D D260	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-266. Register Call Summary for Register VPE_PERF_MON24

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-267. VPE_PERF_MON25

Address Offset	0x0000 0264		
Physical Address	0x489D D264	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-268. Register Call Summary for Register VPE_PERF_MON25

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-269. VPE_PERF_MON26

Address Offset	0x0000 0268		
Physical Address	0x489D D268	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-270. Register Call Summary for Register VPE_PERF_MON26

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-271. VPE_PERF_MON27

Address Offset	0x0000 026C

Table 10-271. VPE_PERF_MON27 (continued)

Physical Address	0x489D D26C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-272. Register Call Summary for Register VPE_PERF_MON27

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-273. VPE_PERF_MON28

Address Offset	0x0000 0270		
Physical Address	0x489D D270	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-274. Register Call Summary for Register VPE_PERF_MON28

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-275. VPE_PERF_MON29

Address Offset	0x0000 0274		
Physical Address	0x489D D274	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-276. Register Call Summary for Register VPE_PERF_MON29

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-277. VPE_PERF_MON30

Address Offset	0x0000 0278

Table 10-277. VPE_PERF_MON30 (continued)

Physical Address	0x489D D278	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-278. Register Call Summary for Register VPE_PERF_MON30

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-279. VPE_PERF_MON31

Address Offset	0x0000 027C		
Physical Address	0x489D D27C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-280. Register Call Summary for Register VPE_PERF_MON31

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-281. VPE_PERF_MON32

Address Offset	0x0000 0280		
Physical Address	0x489D D280	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30:29	28	27:25	24:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-282. Register Call Summary for Register VPE_PERF_MON32

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-283. VPE_PERF_MON33

Address Offset	0x0000 0284

Table 10-283. VPE_PERF_MON33 (continued)

Physical Address	0x489D D284	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-284. Register Call Summary for Register VPE_PERF_MON33

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-285. VPE_PERF_MON34

Address Offset	0x0000 0288		
Physical Address	0x489D D288	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: vip1_up_y	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: vip1_up_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-286. Register Call Summary for Register VPE_PERF_MON34

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-287. VPE_PERF_MON35

Address Offset	0x0000 028C		
Physical Address	0x489D D28C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: vip1_up_y 0x3: vip1_up_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: vip1_up_y 0x3: vip1_up_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-288. Register Call Summary for Register VPE_PERF_MON35

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-289. VPE_PERF_MON36

Address Offset	0x0000 0290

Table 10-289. VPE_PERF_MON36 (continued)

Physical Address	0x489D D290	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-290. Register Call Summary for Register VPE_PERF_MON36

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-291. VPE_PERF_MON37

Address Offset	0x0000 0294		
Physical Address	0x489D D294	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-292. Register Call Summary for Register VPE_PERF_MON37

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-293. VPE_PERF_MON38

Address Offset	0x0000 0298		
Physical Address	0x489D D298	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-294. Register Call Summary for Register VPE_PERF_MON38

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-295. VPE_PERF_MON39

Address Offset	0x0000 029C

Table 10-295. VPE_PERF_MON39 (continued)

Physical Address	0x489D D29C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-296. Register Call Summary for Register VPE_PERF_MON39

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-297. VPE_PERF_MON40

Address Offset	0x0000 02A0		
Physical Address	0x489D D2A0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-298. Register Call Summary for Register VPE_PERF_MON40

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-299. VPE_PERF_MON41

Address Offset	0x0000 02A4		
Physical Address	0x489D D2A4	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-300. Register Call Summary for Register VPE_PERF_MON41

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-301. VPE_PERF_MON42

Address Offset	0x0000 02A8		
Physical Address	0x489D D2A8	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-302. Register Call Summary for Register VPE_PERF_MON42

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-303. VPE_PERF_MON43

Address Offset	0x0000 02AC		
Physical Address	0x489D D2AC	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-304. Register Call Summary for Register VPE_PERF_MON43

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-305. VPE_PERF_MON44

Address Offset	0x0000 02B0		
Physical Address	0x489D D2B0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-306. Register Call Summary for Register VPE_PERF_MON44

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-307. VPE_PERF_MON45

Address Offset	0x0000 02B4		
Physical Address	0x489D D2B4	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-308. Register Call Summary for Register VPE_PERF_MON45

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-309. VPE_PERF_MON46

Address Offset	0x0000 02B8		
Physical Address	0x489D D2B8	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-310. Register Call Summary for Register VPE_PERF_MON46

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-311. VPE_PERF_MON47

Address Offset	0x0000 02BC		
Physical Address	0x489D D2BC	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		

Table 10-311. VPE_PERF_MON47 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED	STOP_COUNT			RESERVED		START_CLIENT		RESERVED	START_COUNT			CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-312. Register Call Summary for Register VPE_PERF_MON47

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-313. VPE_PERF_MON48

Address Offset	0x0000 02C0		
Physical Address	0x489D D2C0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-314. Register Call Summary for Register VPE_PERF_MON48

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-315. VPE_PERF_MON49

Address Offset	0x0000 02C4		
Physical Address	0x489D D2C4	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-316. Register Call Summary for Register VPE_PERF_MON49

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-317. VPE_PERF_MON50

Address Offset	0x0000 02C8		
Physical Address	0x489D D2C8	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31:30	29:28	27	26:24	23:22	21:20	19	18:16	15:0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0:vpi_ctl 1:vpi_ctl 2:vpi_ctl 3:vpi_ctl	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0:vpi_ctl 1:vpi_ctl 2:vpi_ctl 3:vpi_ctl	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-318. Register Call Summary for Register VPE_PERF_MON50

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-319. VPE_PERF_MON51

Address Offset	0x0000 02CC		
Physical Address	0x489D D2CC	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-320. Register Call Summary for Register VPE_PERF_MON51

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-321. VPE_PERF_MON52

Address Offset	0x0000 02D0		
Physical Address	0x489D D2D0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAPTURE_MODE</td><td colspan="2">STOP_CLIENT</td><td>RESERVED</td><td colspan="3">STOP_COUNT</td><td colspan="2">RESERVED</td><td colspan="2">START_CLIENT</td><td>RESERVED</td><td colspan="3">START_COUNT</td><td colspan="16">CURR_COUNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-322. Register Call Summary for Register VPE_PERF_MON52

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-323. VPE_PRI_CHROMA_CSTAT

Address Offset	0x0000 0300

Table 10-323. VPE_PRI_CHROMA_CSTAT (continued)

Physical Address	0x489D D300	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:8	7:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	LINE_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:8	LINE_MODE	Selects the output mode of the line buffer. 0: repeat lines twice each output data line gets 2 times the number of frame lines. 1: each line once with Line Buffer Disabled, so no mirroring. Each line gets frame lines with identical data. 2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines. 3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.	RW	0x0
7:0	RESERVED		R	0x0

Table 10-324. Register Call Summary for Register VPE_PRI_CHROMA_CSTAT

VPE Functional Description

- Modes of Operation (VPDMA): [0]
- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [2]

Table 10-325. VPE_PRI_LUMA_CSTAT

Address Offset	0x0000 0304		
Physical Address	0x489D D304	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-326. Register Call Summary for Register VPE_PRI_LUMA_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

Table 10-326. Register Call Summary for Register VPE_PRI_LUMA_CSTAT (continued)

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-327. VPE_PRI_FLD1_LUMA_CSTAT

Address Offset	0x0000 0308		
Physical Address	0x489D D308	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-328. Register Call Summary for Register VPE_PRI_FLD1_LUMA_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-329. VPE_PRI_FLD1_CHROMA_CSTAT

Address Offset	0x0000 030C		
Physical Address	0x489D D30C	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:8	7:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	LINE_MODE	3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:8	LINE_MODE	Selects the output mode of the line buffer. 0: repeat lines twice each output data line gets 2 times the number of frame lines. 1: each line once with Line Buffer Disabled, so no mirroring. Each line gets frame lines with identical data. 2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines. 3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.	RW	0x0

Bits	Field Name	Description	Type	Reset
7:0	3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.		R	0x0

Table 10-330. Register Call Summary for Register VPE_PRI_FLD1_CHROMA_CSTAT

VPE Functional Description
- Modes of Operation (VPDMA): [0]
- VPDMA Basic Definitions:

VPE Register Manual
- VPE_VPDMA Register Summary: [2]

Table 10-331. VPE_PRI_FLD2_LUMA_CSTAT

Address Offset	0x0000 0310		
Physical Address	0x489D D310	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0

Bits	Field Name	Description	Type	Reset
9:0	RESERVED		R	0x0

Table 10-332. Register Call Summary for Register VPE_PRI_FLD2_LUMA_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-333. VPE_PRI_FLD2_CHROMA_CSTAT

Address Offset	0x0000 0314		
Physical Address	0x489D D314	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:8	7:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	LINE_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0

Bits	Field Name	Description	Type	Reset
9:8	LINE_MODE	Selects the output mode of the line buffer. 0: repeat lines twice each output data line gets 2 times the number of frame lines. 1: each line once with Line Buffer Disabled, so no mirroring. Each line gets frame lines with identical data. 2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines. 3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.	RW	0x0
7:0	RESERVED		R	0x0

Table 10-334. Register Call Summary for Register VPE_PRI_FLD2_CHROMA_CSTAT

VPE Functional Description

- Modes of Operation (VPDMA): [0]
- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [2]

Table 10-335. VPE_PRI_MV0_CSTAT

Address Offset	0x0000 0330		
Physical Address	0x489D D330	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-336. Register Call Summary for Register VPE_PRI_MV0_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-337. VPE_PRI_MV_OUT_CSTAT

Address Offset	0x0000 033C		
Physical Address	0x489D D33C	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-338. Register Call Summary for Register VPE_PRI_MV_OUT_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-339. VPE_VIP0_UP_Y_CSTAT

Address Offset	0x0000 0390		
Physical Address	0x489D D390	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-340. Register Call Summary for Register VPE_VIP0_UP_Y_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-341. VPE_VIP0_UP_UV_CSTAT

Address Offset	0x0000 0394		
Physical Address	0x489D D394	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31–24	23–16	15	14	13–10	9–0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-342. Register Call Summary for Register VPE_VIP0_UP_UV_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [1]

Table 10-343. VPE_VPI_CTL_CSTAT

Address Offset	0x0000 03D0		
Physical Address	0x489D D3D0	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31:24	23:16	15	14	13:10	9:0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-344. Register Call Summary for Register VPE_VPI_CTL_CSTAT

VPE Functional Description

- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [2]

10.4.7 VPE_TOP_LEVEL Registers

10.4.7.1 VPE_TOP_LEVEL Register Summary

Table 10-345. VPE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_TOP_LEVEL Base Address
VPE_CLKC_PID	RW	32	0x0000 0000	0x489D 0000
VPE_SYSCONFIG	RW	32	0x0000 0010	0x489D 0010
VPE_INTC_INTR0_STATUS_RAW0	RW	32	0x0000 0020	0x489D 0020
VPE_INTC_INTR0_STATUS_RAW1	RW	32	0x0000 0024	0x489D 0024
VPE_INTC_INTR0_STATUS_ENA0	RW	32	0x0000 0028	0x489D 0028
VPE_INTC_INTR0_STATUS_ENA1	RW	32	0x0000 002C	0x489D 002C
VPE_INTC_INTR0_ENA_SET0	RW	32	0x0000 0030	0x489D 0030
VPE_INTC_INTR0_ENA_SET1	RW	32	0x0000 0034	0x489D 0034
VPE_INTC_INTR0_ENA_CLR0	RW	32	0x0000 0038	0x489D 0038
VPE_INTC_INTR0_ENA_CLR1	RW	32	0x0000 003C	0x489D 003C
VPE_INTC_EOI	RW	32	0x0000 00A0	0x489D 00A0
VPE_CLKC_CLKEN	RW	32	0x0000 0100	0x489D 0100
VPE_CLKC_RST	RW	32	0x0000 0104	0x489D 0104
VPE_CLKC_DPS	RW	32	0x0000 010C	0x489D 010C
VPE_RANGE_MAP	RW	32	0x0000 011C	0x489D 011C

10.4.7.2 VPE_TOP_LEVEL Register Description

Table 10-346. VPE_CLKC_PID

Address Offset	0x0000 0000		
Physical Address	0x489D 0000	**Instance**	VPE_TOP_LEVEL
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															

Bits	Field Name	Description	Type	Reset
31:0	PID		R	0x0

Table 10-347. Register Call Summary for Register VPE_CLKC_PID

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-348. VPE_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x489D 0010	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31–6	5–4	3–2	1–0
RESERVED	STANDBYMODE	IDLEMODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state 0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only 0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only 0x2: Same behavior as bit-field value of 0x1. 0x3: Reserved	RW	0x2
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only 0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented	RW	0x2
1:0	RESERVED		R	0x0

Table 10-349. Register Call Summary for Register VPE_SYSCONFIG

VPE Functional Description

- VPE Idle Mode: [0]
- VPE StandBy Mode: [1]

Table 10-349. Register Call Summary for Register VPE_SYSCONFIG (continued)

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [2]

Table 10-350. VPE_INTC_INTR0_STATUS_RAW0

Address Offset	0x0000 0020		
Physical Address	0x489D 0020	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DEI_FMD_INT_RAW	RESERVED	VPDMA_INT0_DESCRIPTOR_RAW	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA_INT0_LIST0_COMPLETE_RAW

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_RAW	DEI Film Mode Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
17	RESERVED		RW	0x0
16	VPDMA_INT0_DESCRIPTOR_RAW	VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
8	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_RAW	VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 10-351. Register Call Summary for Register VPE_INTC_INTR0_STATUS_RAW0

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-352. VPE_INTC_INTR0_STATUS_RAW1

Address Offset	0x0000 0024		
Physical Address	0x489D 0024	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="9">RESERVED</td><td>VIP1_CHR_DS_1_UV_ERR_INT_RAW</td><td colspan="5">RESERVED</td><td>DEI_ERROR_INT_RAW</td><td colspan="8">RESERVED</td><td>VPDMA_INT0_CLIENT_RAW</td><td>RESERVED</td><td>VPDMA_INT0_CHANNEL_GROUP5_RAW</td><td>VPDMA_INT0_CHANNEL_GROUP4_RAW</td><td>VPDMA_INT0_CHANNEL_GROUP3_RAW</td><td>VPDMA_INT0_CHANNEL_GROUP2_RAW</td><td>VPDMA_INT0_CHANNEL_GROUP1_RAW</td><td>VPDMA_INT0_CHANNEL_GROUP0_RAW</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
21:17	RESERVED		RW	0x0
16	DEI_ERROR_INT_RAW	DEI Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
15:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_RAW	VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		RW	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_RAW	VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_RAW	VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_RAW	VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_RAW	VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_RAW	VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_RAW	VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 10-353. Register Call Summary for Register VPE_INTC_INTR0_STATUS_RAW1

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-354. VPE_INTC_INTR0_STATUS_ENA0

Address Offset	0x0000 0028		
Physical Address	0x489D 0028	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DEI_FMD_INT_ENA	RESERVED	VPDMA_INT0_DESCRIPTOR_ENA	VPDMA_INT0_LIST7_NOTIFY_ENA	VPDMA_INT0_LIST7_COMPLETE_ENA	VPDMA_INT0_LIST6_NOTIFY_ENA	VPDMA_INT0_LIST6_COMPLETE_ENA	VPDMA_INT0_LIST5_NOTIFY_ENA	VPDMA_INT0_LIST5_COMPLETE_ENA	VPDMA_INT0_LIST4_NOTIFY_ENA	VPDMA_INT0_LIST4_COMPLETE_ENA	VPDMA_INT0_LIST3_NOTIFY_ENA	VPDMA_INT0_LIST3_COMPLETE_ENA	VPDMA_INT0_LIST2_NOTIFY_ENA	VPDMA_INT0_LIST2_COMPLETE_ENA	VPDMA_INT0_LIST1_NOTIFY_ENA	VPDMA_INT0_LIST1_COMPLETE_ENA	VPDMA_INT0_LIST0_NOTIFY_ENA	VPDMA_INT0_LIST0_COMPLETE_ENA

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_ENA	DEI Film Mode Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA	VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA	VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA	VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA	VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA	VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA	VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA	VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA	VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA	VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
7	VPDMA_INT0_LIST3_NOTIFY_ENA	VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA	VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA	VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA	VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA	VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA	VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA	VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA	VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-355. Register Call Summary for Register VPE_INTC_INTR0_STATUS_ENA0

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-356. VPE_INTC_INTR0_STATUS_ENA1

Address Offset	0x0000 002C		
Physical Address	0x489D 002C	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIP1_CHR_DS_1_UV_ERR_INT_ENA	RESERVED	DEI_ERROR_INT_ENA	RESERVED	VPDMA_INT0_CLIENT_ENA	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_ENA	VPDMA_INT0_CHANNEL_GROUP4_ENA	VPDMA_INT0_CHANNEL_GROUP3_ENA	VPDMA_INT0_CHANNEL_GROUP2_ENA	VPDMA_INT0_CHANNEL_GROUP1_ENA	VPDMA_INT0_CHANNEL_GROUP0_ENA

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
21:17	RESERVED		RW	0x0
16	DEI_ERROR_INT_ENA	DEI Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
15:8	RESERVED		RW	0x0
7	VPDMA_INT0_CLIENT_ENA	VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		RW	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA	VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA	VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA	VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA	VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 10-357. Register Call Summary for Register VPE_INTC_INTR0_STATUS_ENA1

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-358. VPE_INTC_INTR0_ENA_SET0

Address Offset	0x0000 0030

Table 10-358. VPE_INTC_INTR0_ENA_SET0 (continued)

Physical Address	0x489D 0030	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DEI_FMD_INT_ENA_SET	RESERVED	VPDMA_INT0_DESCRIPTOR_ENA_SET	VPDMA_INT0_LIST7_NOTIFY_ENA_SET	VPDMA_INT0_LIST7_COMPLETE_ENA_SET	VPDMA_INT0_LIST6_NOTIFY_ENA_SET	VPDMA_INT0_LIST6_COMPLETE_ENA_SET	VPDMA_INT0_LIST5_NOTIFY_ENA_SET	VPDMA_INT0_LIST5_COMPLETE_ENA_SET	VPDMA_INT0_LIST4_NOTIFY_ENA_SET	VPDMA_INT0_LIST4_COMPLETE_ENA_SET	VPDMA_INT0_LIST3_NOTIFY_ENA_SET	VPDMA_INT0_LIST3_COMPLETE_ENA_SET	VPDMA_INT0_LIST2_NOTIFY_ENA_SET	VPDMA_INT0_LIST2_COMPLETE_ENA_SET	VPDMA_INT0_LIST1_NOTIFY_ENA_SET	VPDMA_INT0_LIST1_COMPLETE_ENA_SET	VPDMA_INT0_LIST0_NOTIFY_ENA_SET	VPDMA_INT0_LIST0_COMPLETE_ENA_SET

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_ENA_SET	DEI Film Mode Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA_SET	VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_SET	VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_SET	VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_SET	VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA_SET	VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA_SET	VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_SET	VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
9	VPDMA_INT0_LIST4_NOTIFY_ENA_SET	VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA_SET	VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA_SET	VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_SET	VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_SET	VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_SET	VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_SET	VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA_SET	VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA_SET	VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_SET	VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-359. Register Call Summary for Register VPE_INTC_INTR0_ENA_SET0

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-360. VPE_INTC_INTR0_ENA_SET1

Address Offset	0x0000 0034		
Physical Address	0x489D 0034	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	RESERVED	DEI_ERROR_INT_ENA_SET	RESERVED	VPDMA_INT0_CLIENT_ENA_SET	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
21:17	RESERVED		R	0x0
16	DEI_ERROR_INT_ENA_SET	DEI Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15:8	RESERVED		RW	0x0
7	VPDMA_INT0_CLIENT_ENA_SET	VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET	VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-361. Register Call Summary for Register VPE_INTC_INTR0_ENA_SET1

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-362. VPE_INTC_INTR0_ENA_CLR0

Address Offset	0x0000 0038		
Physical Address	0x489D 0038	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DEI_FMD_INT_ENA_CLR	RESERVED	VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_ENA_CLR	DEI Film Mode Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
11	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR	VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-363. Register Call Summary for Register VPE_INTC_INTR0_ENA_CLR0

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-364. VPE_INTC_INTR0_ENA_CLR1

Address Offset	0x0000 003C		
Physical Address	0x489D 003C	**Instance**	VPE_TOP_LEVEL
Description			

Table 10-364. VPE_INTC_INTR0_ENA_CLR1 (continued)

Type RW

31-23	22	21-17	16	15-8	7	6	5	4	3	2	1	0
RESERVED	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	RESERVED	DEI_ERROR_INT_ENA_CLR	RESERVED	VPDMA_INT0_CLIENT_ENA_CLR	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 1 UV Error Enable/ Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
21:17	RESERVED		R	0x0
16	DEI_ERROR_INT_ENA_CLR	DEI Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15:8	RESERVED		RW	0x0
7	VPDMA_INT0_CLIENT_ENA_CLR	VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR	VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-365. Register Call Summary for Register VPE_INTC_INTR0_ENA_CLR1

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-366. VPE_INTC_EOI

Address Offset	0x0000 00A0		
Physical Address	0x489D 00A0	**Instance**	VPE_TOP_LEVEL
Description	INTC EOI Register. This register contains the EOI vector register contents as defined by HL0.8		
Type	RW		

31–0
EOI_VECTOR

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the ipgenericirq for intr output. There are 4 interrupt outputs: 0x0 : Write to intr0 IP Generic 0x1 : Write to intr1 IP Generic 0x2 : Write to intr2 IP Generic 0x3 : Write to intr3 IP Generic Any other write value is ignored.	RW	0x0

Table 10-367. Register Call Summary for Register VPE_INTC_EOI

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-368. VPE_CLKC_CLKEN

Address Offset	0x0000 0100		
Physical Address	0x489D 0100	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31–2	1	0
RESERVED	PRIM_DP_EN	VPDMA_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	PRIM_DP_EN	Primary Video Data Path Clock Enable 0x0 : Clock Disabled 0x1 : Clock Enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VPDMA_EN	VPDMA Clock Enable 0x0 : Clock Disabled 0x1 : Clock Enabled	RW	0x0

Table 10-369. Register Call Summary for Register VPE_CLKC_CLKEN

VPE Functional Description

- VPE Clocks: [0] [1]

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [2]

Table 10-370. VPE_CLKC_RST

Address Offset	0x0000 0104		
Physical Address	0x489D 0104	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAIN_RST	RESERVED																													PRIM_DP_RST	VPDMA_RST

Bits	Field Name	Description	Type	Reset
31	MAIN_RST	Reset for entire data path in VPE0	RW	0x0
30:2	RESERVED		R	0x0
1	PRIM_DP_RST	Primary Video Data Path Reset	RW	0x0
0	VPDMA_RST	VPDMA Reset	RW	0x0

Table 10-371. Register Call Summary for Register VPE_CLKC_RST

VPE Functional Description

- VPE Software Reset: [0] [1]

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [2]

Table 10-372. VPE_CLKC_DPS

Address Offset	0x0000 010C		
Physical Address	0x489D 010C	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	COLOR_SEPARATE_422	CHR_DS_BYPASS	RESERVED	CHR_DS_SRC_SELECT	RGB_OUT_SELECT	RESERVED	CSC_SRC_SELECT	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	COLOR_SEPARATE_422	422 Color Separate Select 0x0 : 422 output will be 16 bit interleaved (YCbCr) 0x1 : 422 output will be 8 bit coplanar (Y, CbCr) This bit controls whether 422 output will be color separate or interleaved. This bit only applies IF chr_ds_bypass is 1 (means 422 output, not 420) and rgb_out_select is 0 (means 422 output, not RGB or 444). 420 is always coplanar, so this only applies if the output type is 422.	RW	0x0
17	RESERVED		R	0x0
16	CHR_DS_BYPASS	Chroma Downsampler Bypass 0x0 : Chroma Downsampler selected 0x1 : Chroma Downsampler Bypassed Chroma Downsampler Bypassed means the output format from VPE0 will be 422 data. Selected means the output format will be 420. This bit is only applicable if rgb_out_select is 0. It is a don't care if rgb_out_select is 1.	RW	0x0
15:12	RESERVED		R	0x0
11:9	CHR_DS_SRC_SELECT	Chroma Downsampler Source Select 000 : Path Disabled (no input to CHR_DS) 001 : Reserved (Path Disabled) 010 : Reserved (Path Disabled) 011 : Reserved (Path Disabled) 100 : Reserved (Path Disabled) 101 : Source from DEI Scaler (422) 110 : Reserved (Path Disabled) 111 : Reserved (Path Disabled)	RW	0x0
8	RGB_OUT_SELECT	RGB Output Select 0x0 : Output Type is 420/422 0x1 : Output Type is RGB/444	RW	0x0
7:3	RESERVED		R	0x0
2:0	CSC_SRC_SELECT	CSC Source Select: 000 : Path Disabled 001 : Reserved (Path Disabled) 010 : Reserved (Path Disabled) 011 : Source from DEI Scaler (422) 100 : Reserved (Path Disabled) 101 : Reserved (Path Disabled) 110 : Reserved (Path Disabled) 111 : Reserved (Path Disabled)	RW	0x0

Table 10-373. Register Call Summary for Register VPE_CLKC_DPS

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-374. VPE_RANGE_MAP

Address Offset	0x0000 011C		
Physical Address	0x489D 011C	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31:29	28	27:7	6	5:3	2:0
RESERVED	RANGE_REDUCTION_PRIM_ON	RESERVED	RANGE_MAP_PRIM_ON	RANGE_MAPUV_PRIM	RANGE_MAPY_PRIM

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	RANGE_REDUCTION_PRIM_ON	Range Reduction ON for Primary input	RW	0x0
27:7	RESERVED		R	0x0
6	RANGE_MAP_PRIM_ON	Range Mapping ON for Primary input	RW	0x0
5:3	RANGE_MAPUV_PRIM	Range Map UV for Primary input	RW	0x0
2:0	RANGE_MAPY_PRIM	Range Map Y for Primary input	RW	0x0

Table 10-375. Register Call Summary for Register VPE_RANGE_MAP

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Chapter 11

Display Subsystem

This chapter describes the display subsystem for the device.

Note

All the Display and HDMI features described in this chapter may not be supported in software. For example, 3D-frame packing is not supported. Refer to the software development kit (SDK) for which Display and HDMI features are supported.

11.1 Display Subsystem Overview.......... 2599
11.2 Display Controller.......... 2650
11.3 High-Definition Multimedia Interface.......... 2897

11.1 Display Subsystem Overview

The Display Subsystem (DSS) provides the logic to display a video frame from the system memory frame buffer on a liquid-crystal display (LCD) panel or TV set.

The display subsystem can display different pictures simultaneously by using three LCD outputs (LCD1, LCD2, and LCD3), in addition to a TV output.

All three LCD outputs are available on three parallel interfaces (DPI1, DPI2, and DPI3), providing support for MIPI DPI 2.0, or BT-656 or BT-1120.

The TV output is available on one of the following interfaces

- High-Definition Multimedia Interface (HDMI)
- DPI1 parallel interface

The modules integrated in the display subsystem are:

- Display controller (DISPC):
  - One direct memory access (DMA) engine
  - One graphics pipeline (GFX), three video (VID) pipelines, and one write-back (WB) pipeline
  - Three LCD outputs and one TV output, each with a dedicated overlay manager
- HDMI protocol engine:
  - HDMI 1.4a support up to 1080p@60Hz (including 3D frame-packing support)
  - 36-bit RGB color
  - HDCP 1.4 key protection
  - Deep color mode support (10-bit/12-bit for 148.5-MHz pixel clock)

The necessary video phase-locked loops (PLLs) with their corresponding control modules, and the physical layer (PHY) for HDMI are outside the display subsystem. The video PLLs allow independant display outputs at different frequencies. The supported PLLs and PHY are:

- DPLL_HDMI / HDMI_PHY
- DPLL_VIDEO1
- DPLL_VIDEO2

To ensure efficient bandwidth, the display subsystem integrates a connection between the device L3_MAIN interconnect and the DISPC to exchange data with synchronous dynamic random access memory (SDRAM) using the DISPC DMA engine. The same connection is also used for configuration.

Figure 11-1 is a high-level diagram of the display subsystem.

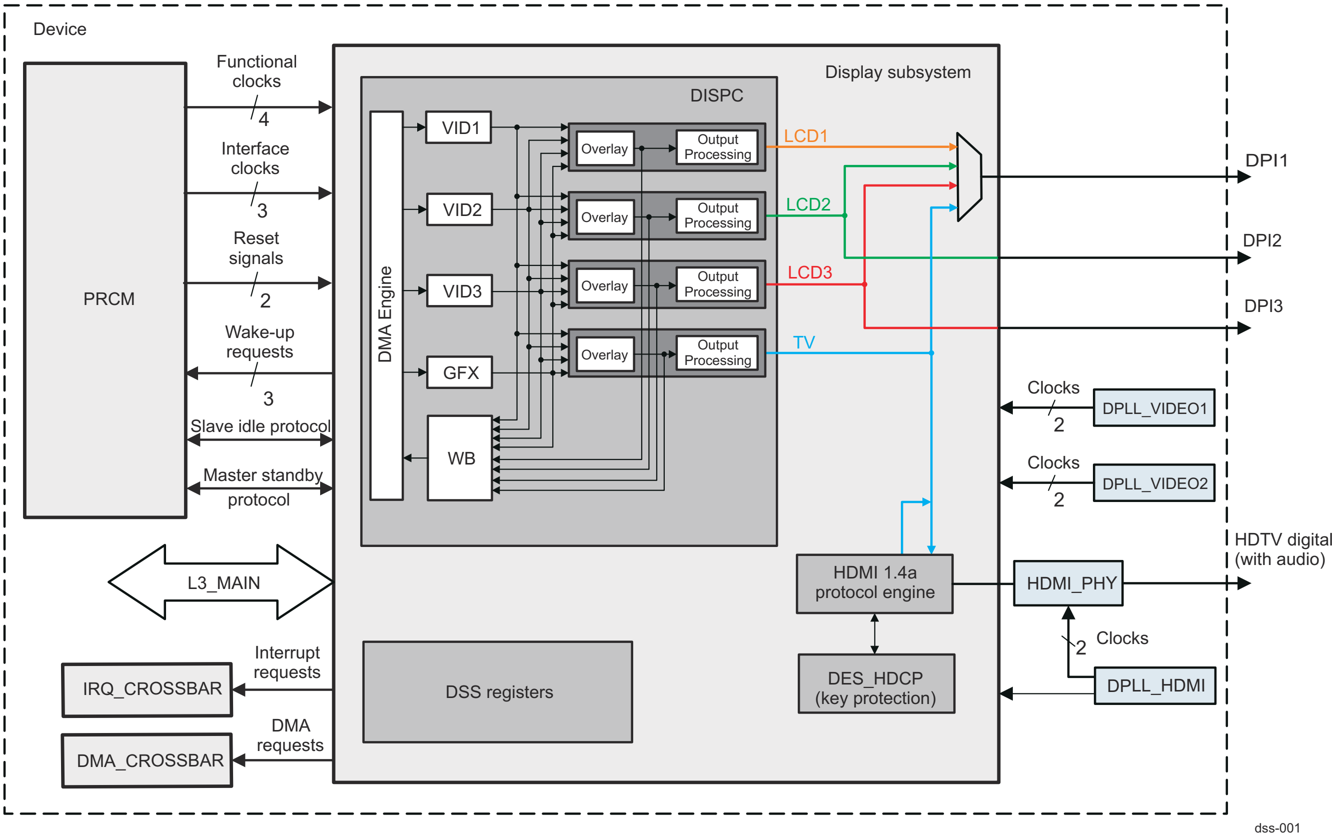


Figure 11-1. Display Subsystem Overview

11.1.1 Display Subsystem Environment

This section describes the various outputs handled by the display subsystem:

- LCD support
- TV display support

Figure 11-2 is a diagram of the display subsystem environment.

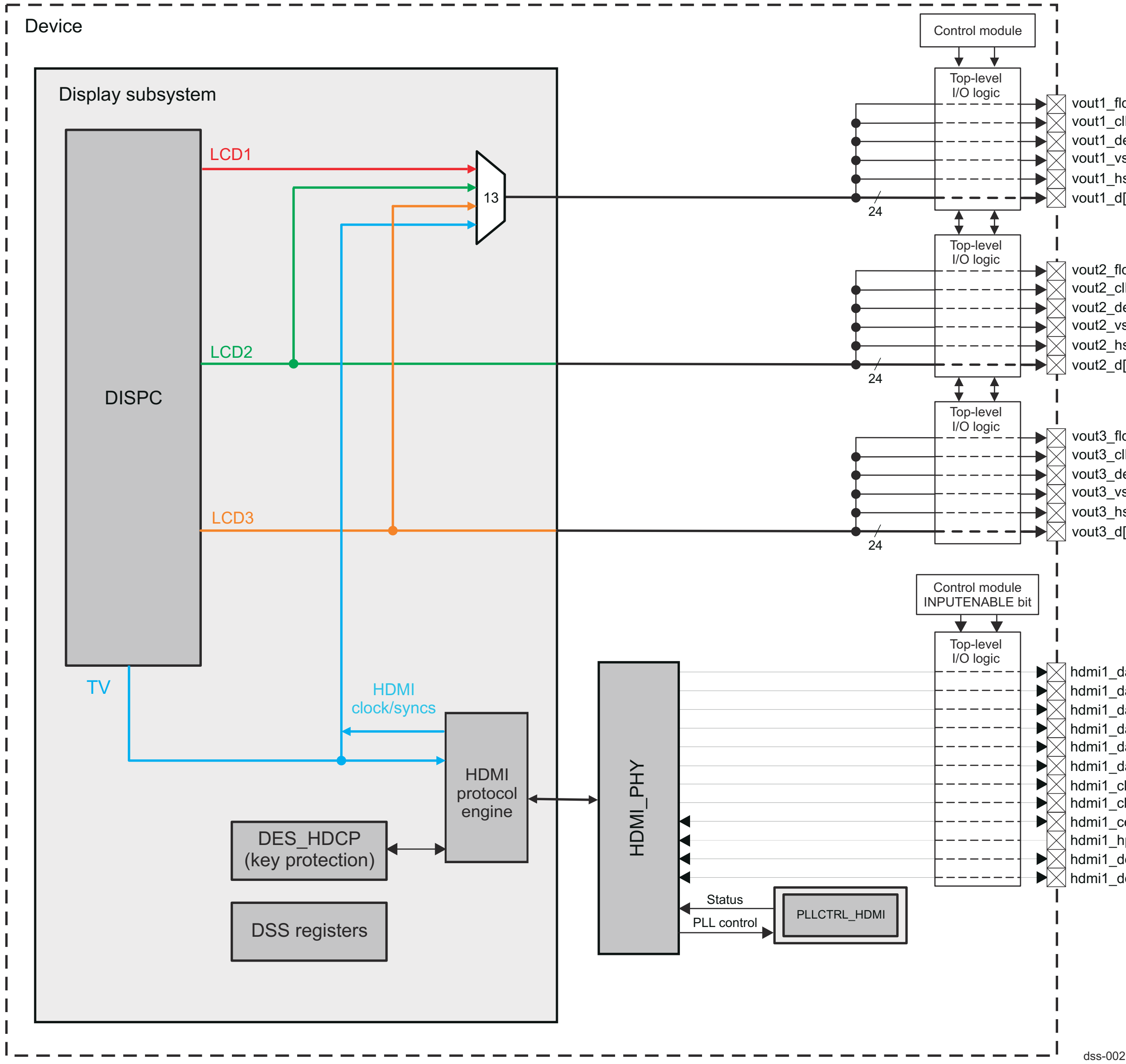


Figure 11-2. Display Subsystem Environment

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the control module registers and dedicated IP registers. For more information, see *Pad Configuration Registers* in *Control Module*.

11.1.1.1 Display Subsystem LCD Support

LCD panels can be connected to the display subsystem of the device using DPI1, DPI2, and/or DPI3 parallel interfaces.

11.1.1.1.1 Display Subsystem LCD with Parallel Interfaces

Using parallel output mode, the module on the display subsystem processing path is DISPC.

In synchronous parallel interface, the required data and control signals are provided directly to external MIPI DPI 2.0, or BT-656 or BT-1120 compatible parallel panels.

The DISPC is connected to the system memory through the device L3_MAIN interconnect and uses its own DMA engine (with embedded FIFO) to read data from the system memory. For more details, see Section 11.2.1, *Display Controller*.

All three DISPC LCD outputs are available for DSS DPI1 output, with selection done through the DSS_CTRL[17:16] PARALLEL_SEL bit field (multiplexer 13 in Figure 11-2). DSS DPI2 and DPI3 outputs are hardwired to DISPC LCD2 and LCD3 outputs.

Table 11-1 to Table 11-3 lists the outgoing display subsystem signals on the device boundary pads.

Table 11-1. Display Subsystem DPI1 Interface Signals Mapping

Signal Names at Device Pads (See Figure 11-2)	DSS_CTRL[17:16] PARALLEL_SEL = 1, 2, or 3 DISPC LCDx Channel Out (pixel data, clock, syncs) (where x = 1, 2, or 3)
vout1_fid	DISPC_LCDx_FID
vout1_clk	DISPC_LCDx_PCLK
vout1_de	DISPC_LCDx_DE
vout1_vsync	DISPC_LCDx_VSYNC
vout1_hsync	DISPC_LCDx_HSYNC
vout1_d[23:0]	DISPC_LCDx_DATA[23:0]

Note

By default, DSS_CTRL[17:16] PARALLEL_SEL = 0x0, and DISPC TV data and HDMI clk/sync signals are output on DPI1 interface pins. Software must program the PARALLEL_SEL bit-field to value other than 0x0, in order DISPC LCD data and clk/sync signals to be output on DPI1 interface pins.

Table 11-2. Display Subsystem DPI2 Interface Signals Mapping

Signal Names at Device Pads (See Figure 11-2)	DISPC LCD2 Channel Out (pixel data, clock, syncs)
vout2_fid	DISPC_LCD2_FID
vout2_clk	DISPC_LCD2_PCLK
vout2_de	DISPC_LCD2_DE
vout2_vsync	DISPC_LCD2_VSYNC
vout2_hsync	DISPC_LCD2_HSYNC
vout2_d[23:0]	DISPC_LCD2_DATA[23:0]

Table 11-3. Display Subsystem DPI3 Interface Signals Mapping

Signal Names at Device Pads (See Figure 11-2)	DISPC LCD3 Channel Out (pixel data, clock, syncs)
vout3_fid	DISPC_LCD3_FID
vout3_clk	DISPC_LCD3_PCLK
vout3_de	DISPC_LCD3_DE
vout3_vsync	DISPC_LCD3_VSYNC
vout3_hsync	DISPC_LCD3_HSYNC
vout3_d[23:0]	DISPC_LCD3_DATA[23:0]

For more details on LCD output pixel data formats for the parallel interface, see Section 11.2 *DISPC Environment*, in *Display Controller*.

11.1.1.2 Display Subsystem TV Display Support

When TV data flow is needed from the display subsystem, the DPI1 output or HDMI output can be used.

11.1.1.2.1 Display Subsystem TV With Parallel Interfaces

In synchronous parallel interface configuration, the required data is provided by the DISPC TV output, and control signals are provided by the HDMI module. Then they are merged and provided to the DPI1 output.

This configuration option is available and can be selected through the DSS_CTRL[17:16] PARALLEL_SEL bit field (multiplexer 13 in Figure 11-4).

Table 11-4 lists the outgoing display subsystem signals on the device boundary pads.

Table 11-4. Display Subsystem TV Parallel Interface Signals Mapping

Signal Names at Device Pads (See Figure 11-2)	DSS_CTRL[17:16] PARALLEL_SEL = 0 DISPC TV Channel Out (pixel data) HDMI (pixel clock, syncs)
vout1_fid	HDMI_M_FID
vout1_clk	DSS_HDMI_PCLK
vout1_de	HDMI_M_DE
vout1_vsync	HDMI_M_VS
vout1_hsync	HDMI_M_HS
vout1_d[23:0]	DISPC_TV_DATA[29:0]

For more details on TV output pixel data formats for the parallel interface, see Section 11.2.2 *DISPC Environment*, in *Section 11.2.1* *Display Controller*.

11.1.1.2.2 Display Subsystem TV With Serial Interfaces

In serial interface configuration, the HDMI module is used. The DISPC, HDMI, DES_HDCP, and the associated HDMI PHY module are used in the data path. The HDMI module converts the RGB video into standard high-definition digital video format. The HDMI module has a dedicated PLL, which can multiply the pixel clock by an appropriate factor. The data is transmitted on three differential data pairs and an additional clock pair.

For more details, see Section 11.3, in *High-Definition Multimedia Interface*.

11.1.2 Display Subsystem Integration

This section describes the integration of the display subsystem module in the device, including information about clocks, resets, and hardware requests.

Figure 11-3 shows the integration of the display subsystem in the device.

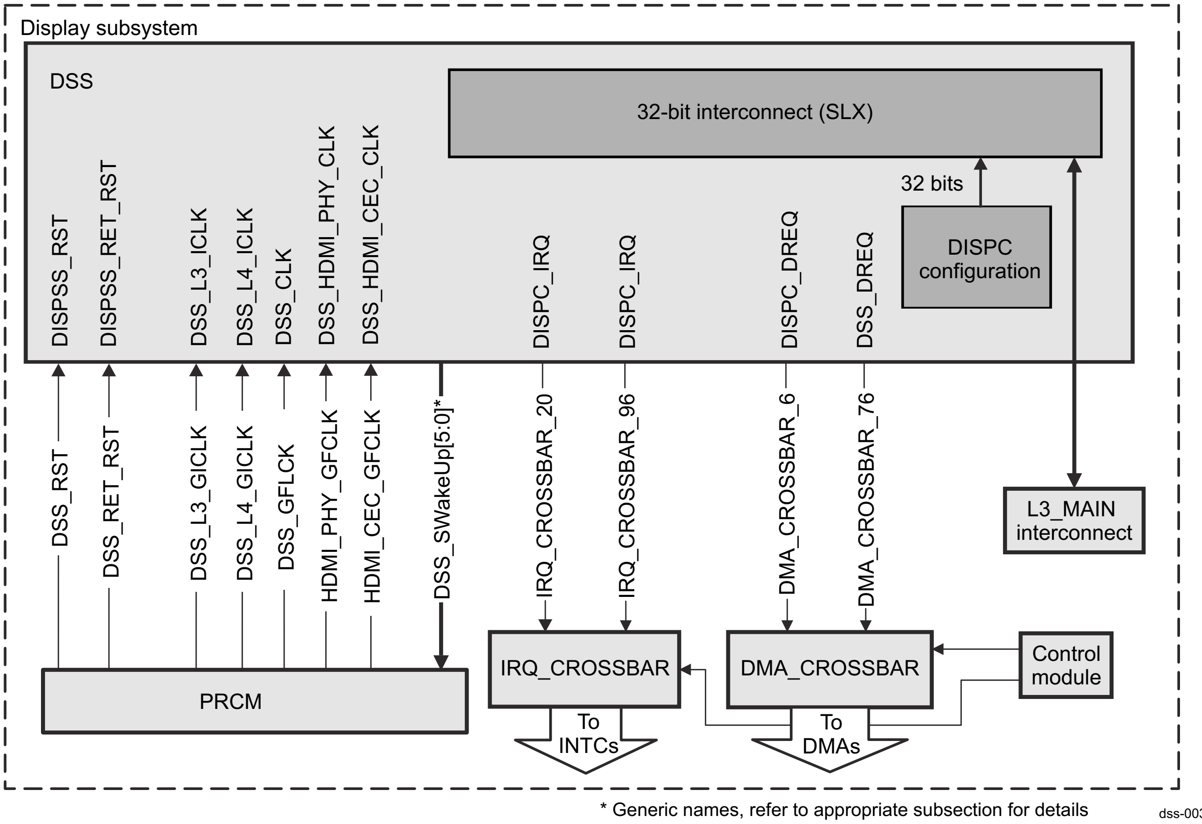


Figure 11-3. Display Subsystem Integration

Table 11-5. Display Subsystem Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DISPC	DISPC_IRQ	IRQ_CROSSBAR_20	MPU_IRQ_25	DISPC interrupt requests.
			DSP1_IRQ_58	
			DSP2_IRQ_58	
			IPU1_IRQ_26	
			IPU2_IRQ_26	
HDMI	HDMI_IRQ	IRQ_CROSSBAR_96	MPU_IRQ_101	HDMI interrupt requests.
			IPU1_IRQ_26	
			IPU2_IRQ_26	
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
DISPC	DISPC_DREQ	DMA_CROSSBAR_6	DMA_EDMA_DREQ_5	The line trigger signal to synchronize a memory-to-memory logical channel in the DMA is generated by the DISPC IP.
			DMA_SYSTEM_DREQ_5	
HDMI	DSS_DREQ	DMA_CROSSBAR_76	DMA_SYSTEM_DREQ_75	Display subsystem HDMI audio DMA request

Note

The "**Default Mapping**" column in Table 11-5 *Display Subsystem Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.
For more information about the device DMA_SYSTEM module, see *System DMA*.
For more information about the device EDMA module, see *Enhanced DMA*.

11.1.2.1 Display Subsystem Clocks

The power, reset, and clock management (PRCM) module provides clock signals to the display subsystem.

Figure 11-4 shows the details of the display subsystem clock tree.

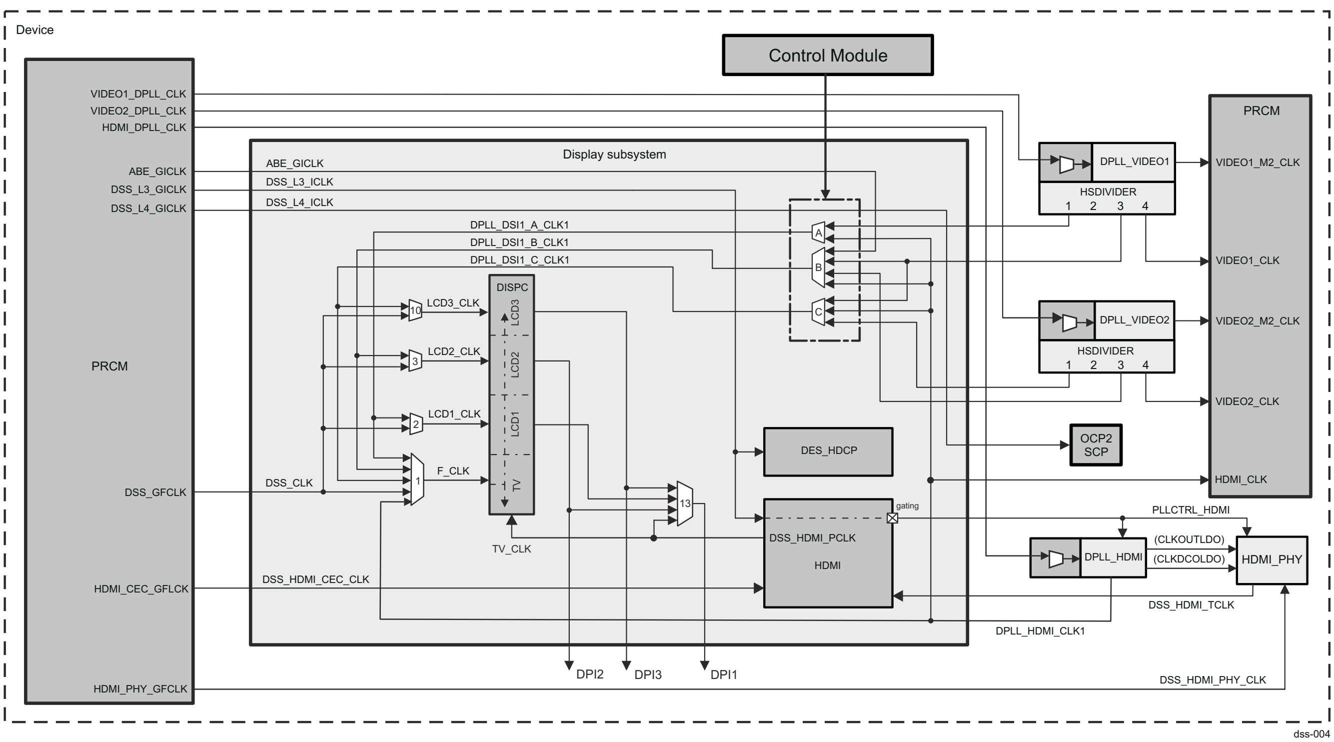


Figure 11-4. Display Subsystem Clock Tree

Table 11-6 lists the main DSS clocks and their sources.

Table 11-6. Display Subsystem Clocks

		Clocks		
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DSS	DPLL_VIDEO1 input	VIDEO1_DPLL_CLK	PRCM module	Functional clock
	DPLL_VIDEO2 input	VIDEO2_DPLL_CLK	PRCM module	Functional clock
	DPLL_HDMI input	HDMI_DPLL_CLK	PRCM module	Functional clock
	DSS_L3_ICLK	DSS_L3_GICLK	PRCM module	Interface clock
	DSS_L4_ICLK	DSS_L3_GICLK	PRCM module	Interface clock

Table 11-6. Display Subsystem Clocks (continued)

-	ABE_GICLK	PRCM, DPLL_ABE	Functional clock
DSS_CLK	DSS_GFCLK	PRCM module	Main display subsystem functional clock
HDMI_CEC_GFCLK	DSS_HDMI_CEC_CLK	PRCM module	HDMI core CEC engine clock
HDMI_PHY_GFCLK	DSS_HDMI_PHY_CLK	PRCM module	HDMI_PHY clock

- The clock source for the L3 interface clock is DSS_L3_ICLK. All clocks can be gated at PRCM level. See *Module-Level Clock Management*, in *Power, Reset and Clock Management*.
- The clock sources for the display subsystem modules are listed in Table 11-7.

Table 11-7. Display Subsystem Modules Clock Sources

Destination	Source Signal Name	Source	Multiplexer Number in Figure 11-4	DSS_CTRL Register Bit-field
DISPC functional clock (F_CLK)	DSS_CLK	PRCM	1	[9:7] F_CLK_SWITCH
	DPLL_DSI1_A_CLK1	DPLL_VIDEO1, DPLL_HDMI		
	DPLL_DSI1_B_CLK1	DPLL_VIDEO1, DPLL_VIDEO2, DPLL_HDMI, DPLL_ABE		
	DPLL_DSI1_C_CLK1	DPLL_VIDEO1, DPLL_VIDEO2, DPLL_HDMI		
	DPLL_HDMI_CLK1	DPLL_HDMI		
DISPC LCD1 functional clock (LCD1_CLK)	DSS_CLK	PRCM	2	[0] LCD1_CLK_SWITCH
	DPLL_DSI1_A_CLK1	DPLL_VIDEO1, DPLL_HDMI		
DISPC LCD2 functional clock (LCD2_CLK)	DSS_CLK	PRCM	3	[12] LCD2_CLK_SWITCH
	DPLL_DSI1_B_CLK1	DPLL_VIDEO1, DPLL_VIDEO2, DPLL_HDMI, DPLL_ABE		
DISPC LCD3 functional clock (LCD3_CLK)	DSS_CLK	PRCM	10	[19] LCD3_CLK_SWITCH
	DPLL_DSI1_C_CLK1	DPLL_VIDEO1, DPLL_VIDEO2, DPLL_HDMI		
DISPC TV functional clock	DPLL_HDMI_CLK1	DPLL_HDMI	N/A	N/A
DISPC TV pixel clock (TV_CLK)	DSS_HDMI_PCLK	HDMI	N/A	N/A
DISPC internal functional clock (DISPC_CLK after divider of F_CLK)	F_CLK	DSS	N/A	N/A
DPI1 functional/pixel clock	DSS_DISPC_LCD1_PCLK	DISPC	13	[17:16] PARALLEL_SEL
	DSS_DISPC_LCD2_PCLK	DISPC		
	DSS_DISPC_LCD3_PCLK	DISPC		
	DSS_HDMI_PCLK	HDMI		
DPI2 functional/pixel clock	DSS_DISPC_LCD2_PCLK	DISPC	N/A	N/A
DPI3 functional/pixel clock	DSS_DISPC_LCD3_PCLK	DISPC	N/A	N/A
HDMI timing clock (DSS_HDMI_TCLK)	N/A	HDMI_PHY	N/A	N/A

Note

- In Figure 11-4, clkout1, clkout3, and clkout4 of DPLL_VIDEO1 and DPLL_VIDEO2 correspond to M4, M6, and M7 outputs of the associated HS divider module (HSDIVIDER). For more information on the configuration of DPLL_VIDEO clocks, see Section 11.1.3.3, *DPLL_VIDEO Functional Description*.
- For synchronization purpose, the clkout4 (M7) output of each HSDIVIDER is used to source McASP module and peripheral timers. For more information, see *Timers,* and *Multichannel Audio Serial Ports*.
- Multiplexers A, B, and C in Figure 11-4 are controlled by [4:3] DSI1_A_CLK1_SELECTION, [6:5] DSI1_B_CLK1_SELECTION, and [8:7] DSI1_C_CLK1_SELECTION bit-fields, respectively, of DSS_PLL_CONTROL register in the device Control Module. For more information, see *Control Module*.

11.1.2.2 Display Subsystem Resets

The PRCM module provides two reset signals to the display subsystem.

Figure 11-5 shows the details of the reset tree for the display subsystem.

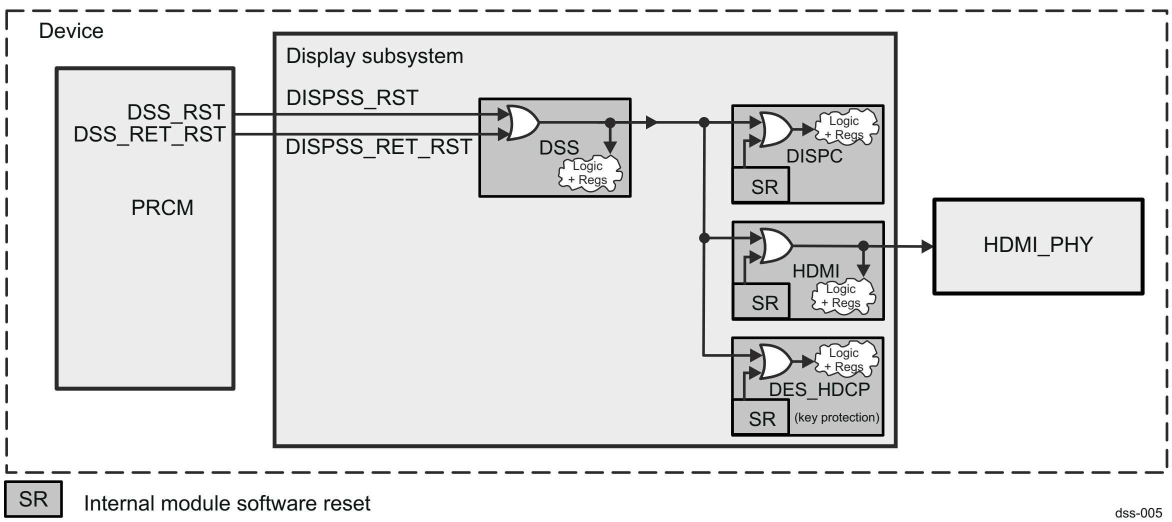


Figure 11-5. Display Subsystem Reset Scheme

Table 11-8 lists the resets for the display subsystem.

Table 11-8. Display Subsystem Resets

	Resets			
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DSS	DISPSS_RET_RST	DSS_RET_RST	PRCM module	Retention reset
	DISPSS_RST	DSS_RST	PRCM module	Nonretention reset

The display subsystem receives its DISPSS_RST reset signal (the reset signal of the display subsystem power domain) from the PRCM module. The DISPSS_RET_RST is used only for the DES_HDCP key protection module and HDMI.

11.1.2.3 Display Subsystem Power Management

The display subsystem modules are in the display subsystem power domain.

Table 11-9 lists the power domains in the display subsystem.

Table 11-9. Display Subsystem Power Domains

Module Instance	Attributes
	Power Domain
Display subsystem	PD_DSS
DISPC	For more details, see Section 11.2.1 *DISPC Overview*, in *Display Controller*.
HDMI	For more details, see Section 11.3.1 *HDMI Overview*, in *High-Definition Multimedia Interface*.

11.1.2.3.1 Display Subsystem Standby Mode

As part of the system-wide power-management scheme, the display subsystem supports the MStandby/MWait and SIdleReq/SIdleAck protocols:

- MStandby/MWait
  - DISPC
- SIdleReq/SIdleAck
  - DISPC
  - HDMI
  - DES_HDCP (key protection)
  - DSS interconnect

The PRCM module asserts the MWait and received MStandby directly from DISPC. When the display subsystem initiates a standby procedure, it also initiates a master standby/wait protocols with the PRCM module that lets the PRCM module cut the display subsystem clocks. For information about the conditions that allow the subsystem to exit standby mode, see Section 11.2.1 *DISPC Overview*, in *Display Controller*.

The PRCM also asserts the SIdleReq. Then, it is split at the display subsystem level and send to the appropriate modules. Consequently, all SIdleAck are merged into one and sent back to PRCM.

11.1.2.3.2

Figure 11-6 shows the generation of SIdleAck/MStandby in the display subsystem.

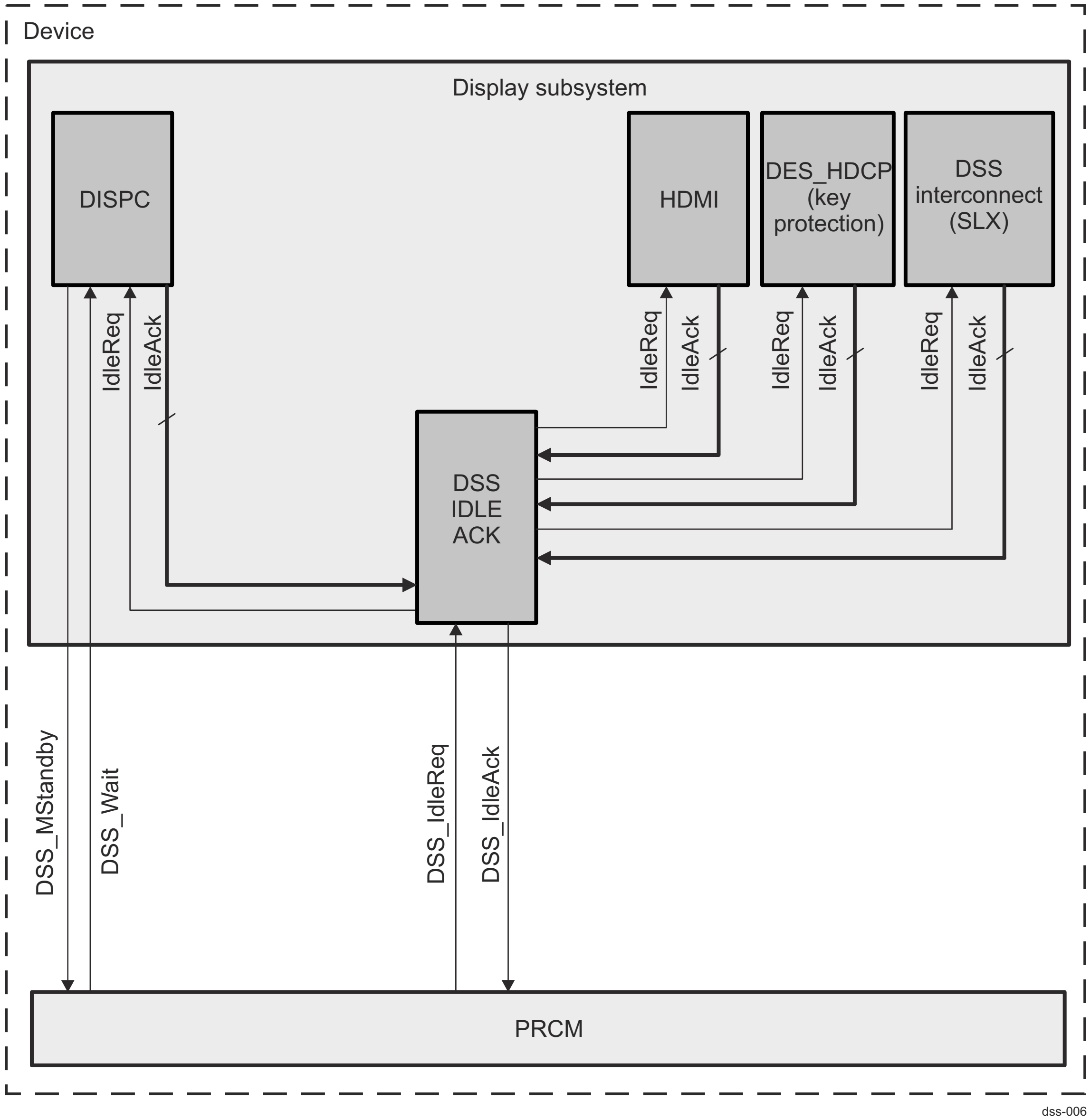


Figure 11-6. Display Subsystem SIdleAck/MStandby Generation

11.1.2.3.3 Display Subsystem Wake-Up Mode

The DISPC and HDMI modules support the wake-up protocol. DSS_HDMI_SWakeUp is associated with HMDI_IRQ, which is generated by the HDMI module. For the events that generate an Swakeup and the description and configuration of the registers, see the DISPC and HDMI TRM chapters.

Figure 11-7 shows wake-up generation in the display subsystem.

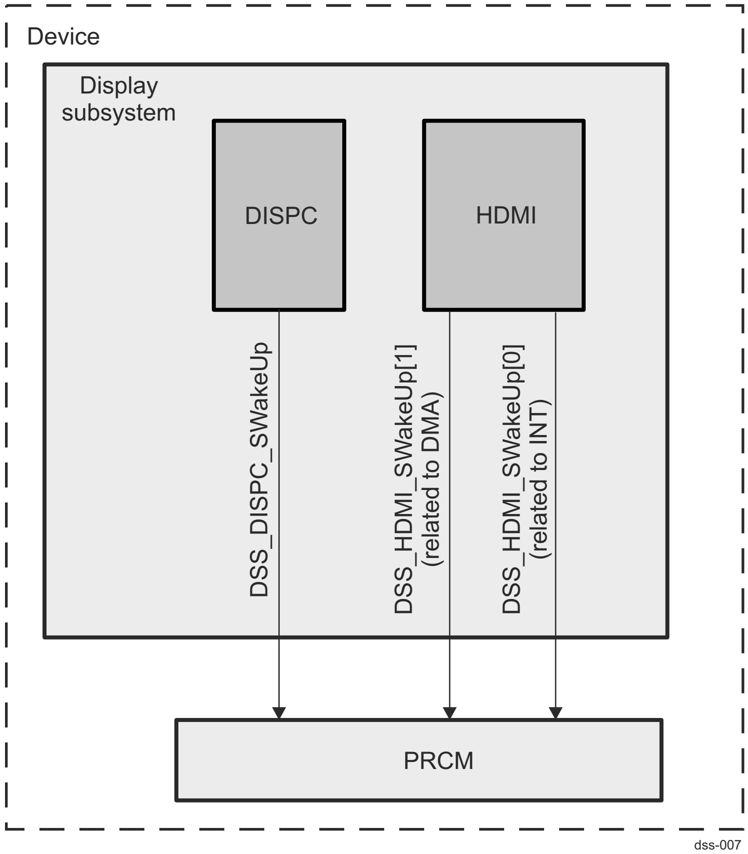


Figure 11-7. Display Subsystem Wake-Up Generation

11.1.3 Display Subsystem DPLL Controllers Functional Description

11.1.3.1 DPLL Controllers Overview

Three PLL controller modules are part of the display subsystem. They use the SCP and power-management port (PMP) as the primary interfaces. The SCP interface sets the configuration of the digital phase-locked loop (DPLL) and HSDIVIDER modules, primarily the various counter values. The PMP controls the power state of the DPLL and HSDIVIDER modules. Figure 11-8 is an overview of a single DPLL_VIDEO controller module in the display subsystem. For more information on DPLL_HDMI control, see Section 11.1.3.4, *DPLL_HDMI*.

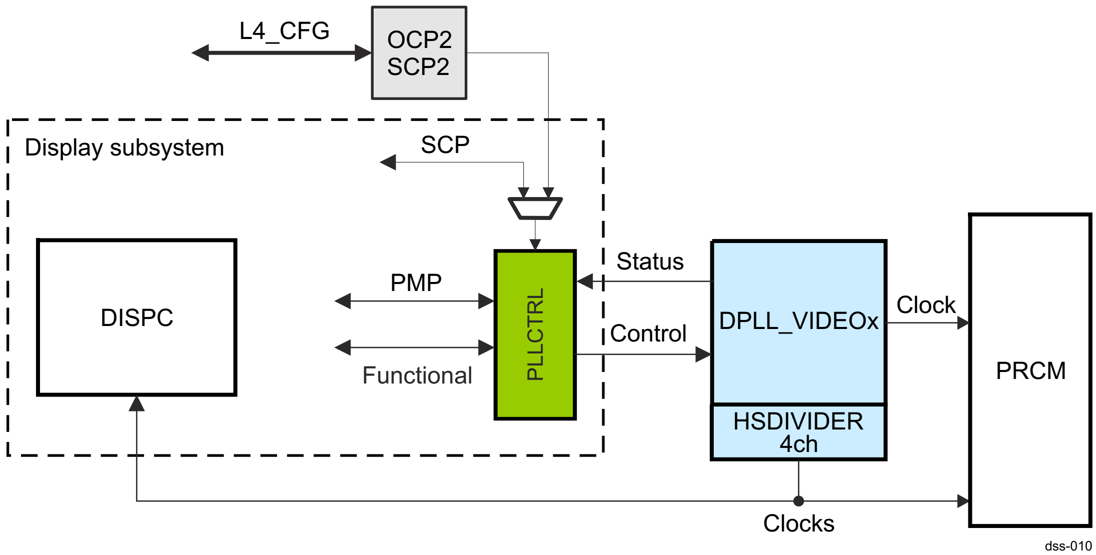


Figure 11-8. PLL Controller Overview

11.1.3.2 OCP2SCP2 Functional Description

In case DSS module is not used, DPLL_VIDEO1, DPLL_VIDEO2 and DPLL_HDMI can be still used to source other modules in the device. The OCP2SCP2 module, which serves as a L4_CFG interconnect adapter, lets user software configure the PLLCTRL for DPLL_VIDEO1, DPLL_VIDEO2 and DPLL_HDMI registers over the L4_CFG port. Figure 11-9 is an overview of the OCP2SCP2 bridge interface.

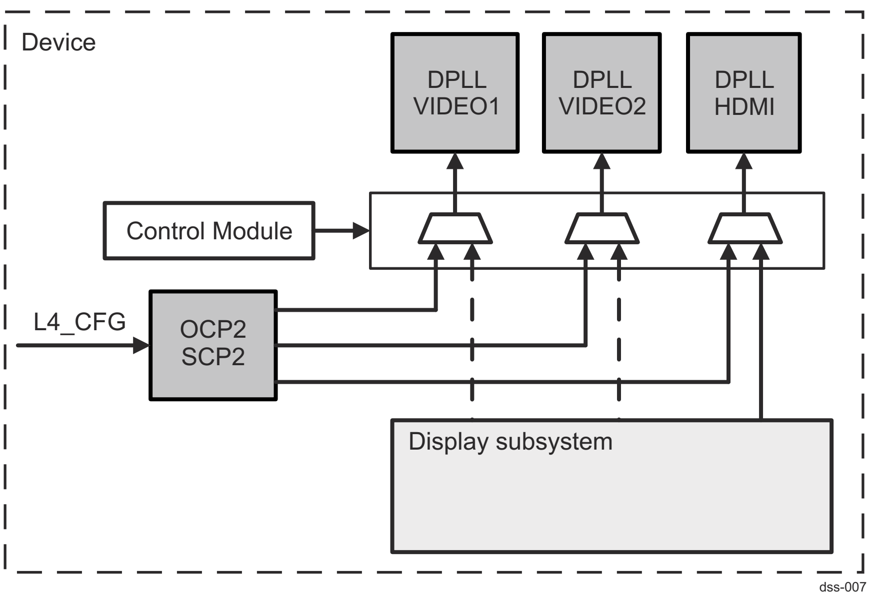


Figure 11-9. OCP2SCP2 Overview

Note

Selection of how DPLLs are controlled is done through registers DSS_PLL_CONTROL[0] PLL_VIDEO1_DSS_CONTROL_DISABLE, DSS_PLL_CONTROL[1] PLL_VIDEO2_DSS_CONTROL_DISABLE and DSS_PLL_CONTROL[2] PLL_HDMI_DSS_CONTROL_DISABLE of the device Control Module. For more information, see *Control Module*.

11.1.3.2.1 OCP2SCP2 Reset

11.1.3.2.1.1 Hardware Reset

The module receives an asynchronous hardware reset (L3INIT_RST) upon power-on reset (POR) at its active low PIRSTNA input. For more information on the hardware reset source, see *Reset Domains*, in *Power, Reset, and Clock Management.*

11.1.3.2.1.2 Software Reset

Setting the OCP2SCP_SYSCONFIG[1] SOFTRESET bit to 1 triggers a software reset on the OCP2SCP2 interconnect adapter. The OCP2SCP_SYSSTATUS[0] RESETDONE bit is used by software to monitor the status of reset completion. Hardware keeps this bit it at 0 while the reset is being executed. A RESETDONE bit transition from 0 to 1 indicates the OCP2SCP2 reset is complete.

11.1.3.2.2 OCP2SCP2 Power Management

The OCP2SCP2 features idle acknowledgement protocol with the PRCM.

11.1.3.2.2.1 Idle Mode

The smart-idle mode supported by OCP2SCP2 is not wakeup capable, meaning that software should explicitly take care to wake the OCP2SCP2, setting the OCP2SCP_SYSCONFIG[4:3] IDLEMODE bit field to 0x1 (no-idle), once it has previously gone to an idle mode. For more information, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management.*

Note

Software must get the OCP2SCP2 module out of IDLE state by setting the OCP2SCP_SYSCONFIG[4:3] IDLEMODE bit field to 0x1, once smart-ilde mode has been selected (OCP2SCP_SYSCONFIG[4:3] IDLEMODE = 0x2).

11.1.3.2.2.2 Clock Gating

OCP2SCP2 module has a local support for an automatic clock gating based on L4_CFG interconnect activity. This feature is enabled by setting the OCP2SCP2[0] AUTOIDLE bit to 0x1, otherwise the module interface clock is free running. For more information, see *Clock Domain-Level Clock Management*, in *Power, Reset, and Clock Management.*

11.1.3.2.3 OCP2SCP2 Timing Registers

The timing configuration register sets various parameters controlling the timing constraints of the OCP2SCP2 module. The division ratio between the L4_CFG interconnect clock - L3INIT_L4_GICLK and the serial configuration port output clock is set through the OCP2SCP_TIMING[9:7] DIVISIONRATIO bit field, with a valid range of 0x1 to 0x7. The OCP2SCP_TIMING[6:4] SYNC1 timing information is programmable in the range 0 to 7 clock cycles, and shows the acceptable delay between the enable and command availability on SCP. The value of OCP2SCP_TIMING[3:0] SYNC2 is also programmable in the range 1 to 15 clock cycles, measured from the moment the command is available on SCP until data is accessible.

11.1.3.3 DPLL_VIDEO Functional Description

11.1.3.3.1 DPLL_VIDEO Controller Architecture

VIDEO1 PLL and VIDEO2 PLL use type-A instances of the DPLL modules. For information regarding DPLL types, see *Power, Reset, and Clock Management*.

Figure 11-10 shows the internal reference diagram of a single VIDEO PLL.

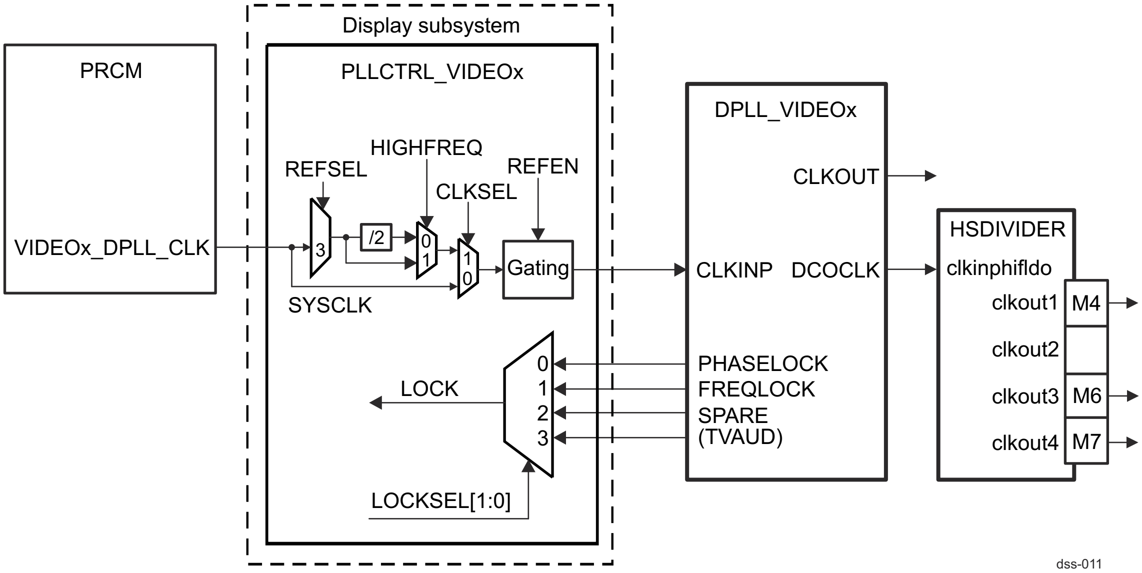


Figure 11-10. VIDEO PLL Reference Diagram

Reference clock control is enabled with PLL_CONFIGURATION2[13] PLL_REFEN bit.

Figure 11-11 is a simplified block diagram of the DPLL_VIDEO instance used for pixel clock generation.

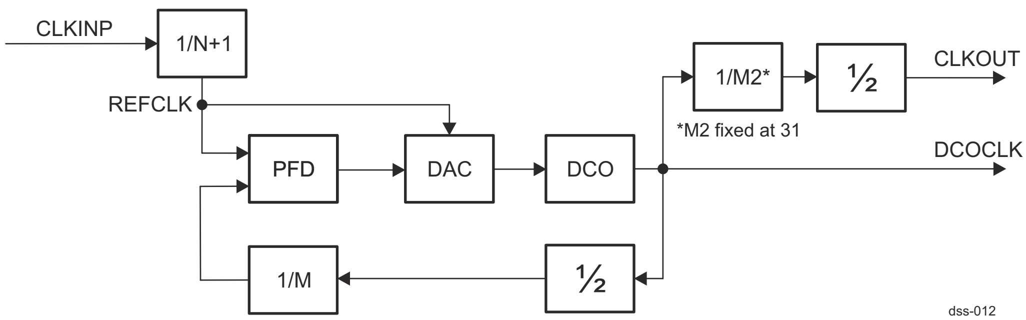


Figure 11-11. DPLL_VIDEO Functional Block Diagram

The input clock CLKINP goes to a pre-divider N + 1.The entire loop runs on the REFCLK clock after this pre-divider. The value of N + 1 is controlled through the PLL_CONFIGURATION1[8:1] PLL_REGN bit field. The CLKINP range is 0.032MHz to 52MHz. The REFCLK range is 0.15MHz to 52MHz.

The output clock DCOCLK is synthesized by a digitally controlled oscillator (the DCO block) that automatically detects the frequency range. The DCOCLK frequency can be given with DCOCLK = CLKINP × 2 × M / (N + 1). For that purpose the feedback multiplier M must be configured through the PLL_CONFIGURATION1[20:9] PLL_REGM bit field.

The CLKOUT frequency can be given with CLKOUT = DCOCLK / (M2 × 2) = DCOCLK / 62. The M2 divider value is hardcoded in HW at 31 (0x1F).

DPLL_VIDEO1 supports Spread Spectrum Clocking (SSC) using the configurations described in the device *Spread Spectrum Clocking Configuration Application Report* (SPRABY0). The SSC feature on DPLL_VIDEO2 is not supported.

11.1.3.3.2 DPLL_VIDEO Operations

The PLL configuration signals operate according to Table 11-10, which indicates the operation when the PLLs are not locked.

Table 11-10. VIDEO PLL Operation Modes When Not Locked

PLL Operation Mode	Stop Mode Low Power(1)	Stop Mode Fast Relock(1)	Idle Bypass
Mode Description	**Output clocks stopped Lowest power standby**	**Output clocks stopped Fastest start-up time**	**Selects when PLL and HSDIVIDER bypass clocks are used**
PLL_CONFIGURATION2[0] PLL_IDLE	0	0	1
PLL_CONFIGURATION2[6] PLL_LOWCURRSTBY	1	0	1

(1) This mode must be used for better performance.

When locked, the PLL output frequency (DCOCLK) is: Input frequency × 2 × M / (N + 1), where:

- M multiplier is programmed in the PLL_CONFIGURATION1[20:9] PLL_REGM bit field.
- N divider is programmed in the PLL_CONFIGURATION1[8:1] PLL_REGN bit field.

Note

When the PLL_REGM bit field is set to 1, the PLL enters a MN-Bypass mode. The DCOCLK clock output goes low and remains low until the PLL exits MN-Bypass mode (by changing the PLL_REGM bit field to a value other than 0 or 1).

11.1.3.3.3 DPLL_VIDEO Error Handling

The PLL lock and recalibration signals can be monitored to detect the loss of lock or the requirement to recalibrate (caused by a large temperature change since the last lock request):

- The PLL_STATUS[1] PLL_LOCK status bit gives the VIDEOx PLL lock state.
- The PLL_STATUS[2] PLL_RECAL status bit informs whether the PLL must be uncalibrated.

11.1.3.3.4 DPLL_VIDEO Software Reset

The VIDEO PLL control module does not have its own software reset. It is reset by the global DSS_RST signal at PRCM module level. See Section 11.1.2.2, *Display Subsystem Resets*. Nevertheless, software users can monitor the reset statuse of the VIDEO PLL control module by reading the PLL_STATUS[0] PLLCTRL_RESET_DONE status bits.

11.1.3.3.5 DPLL_VIDEO Power Management

PLLCTRL manages only the LDO power of the DPLL and HSDIVIDER; this is done by overriding the SYSRESET signals. All other power-management signals are integrated with the display subsystem power management.

The PMP is used to manage the LDO power of the VIDEO digital phase-locked loop (DPLL) through the PLLCTRL module.

Figure 11-12 shows the power states, which can be controlled through DSI_CLK_CTRL[31:30] PLL_PWR_CMD bit field.

The PLLCTRL power status can be read through the DSI_CLK_CTRL [29:28] PLL_PWR_STATUS bit field.

- OFF power state: PLLCTRL is powered down. Internal clock is off.
- ONHSCLK: Power command received
- ONDIV: Clock dividers set.
- ONALL: PLLCTRL is in active state.

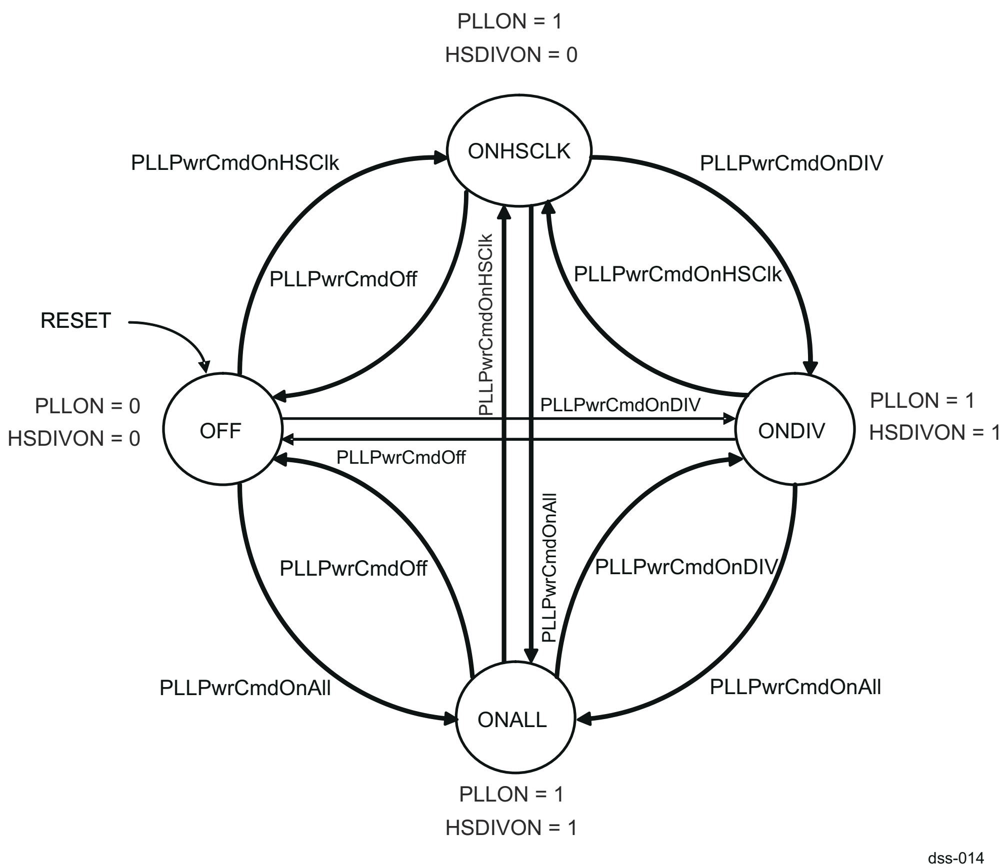


Figure 11-12. VIDEO PLL Power State Diagram

11.1.3.3.6 DPLL_VIDEO HSDIVIDER Loading Operation

In manual mode (PLL_CONTROL[0] PLL_AUTOMODE = 0), it is possible to update the configuration values of HSDIVIDER without starting the DPLL locking sequence. Once all the configuration values have been programmed into the registers, the PLL_GO[1] HSDIVLOAD bit must be set. The TENABLEDIV output is driven high for six SCP clock periods while the TINITZ and TENABLE signals remain unchanged. The HSDIVLOAD bit is cleared at the end of the sequence.

The SCPBUSY signal is high during the sequence until the HSDIVLOAD bit is cleared, thus indicating that there is pending activity in the SCPClk domain. SCPClk must be kept running while this signal is asserted.

The HSDIVIDER sequence and the GO sequence cannot be performed at the same time. If one of the two sequences is running and the trigger bit of the other sequence is set, PLLCTRL finishes the first sequence and then immediately starts the other sequence.

11.1.3.3.7 DPLL_VIDEO Clock Sequence

The VIDEO PLL generates DCOCLK clock to a dedicated HSDIVIDER, which outputs three clocks (clkout1, clkout3, clkout4). If these three clocks are not used, the HSDIVIDER functions are not required.

In addition, a CLKOUT clock is also generated. It is output directly, and does not go through HSDIVIDER.

The following must be considered for Figure 11-13:

- REGM factor is programmed by the PLL_CONFIGURATION1[20:9] PLL_REGM bit field. It is used to tune DCOCLK clock.
- REGN factor is programmed by the PLL_CONFIGURATION1[8:1] PLL_REGN bit field. It is used to tune DCOCLK clock.

- M4REG factor is programed by the PLL_CONFIGURATION1[25:21] M4_CLOCK_DIV bit field, and applies to clkout1 of the DPLL_VIDEO HSDIVIDER.
- M6REG factor is programmed by the PLL_CONFIGURATION3[4:0] M6_CLOCK_DIV bit field, and applies to clkout3 of the DPLL_VIDEO HSDIVIDER.
- M7REG factor is programmed by the PLL_CONFIGURATION3[9:5] M7_CLOCK_DIV bit field, and applies to clkout4 of the DPLL_VIDEO HSDIVIDER.
- M2 divider is hardcoded in HW at 31 (0x1F), and applies to CLKOUT.

Figure 11-13 shows the programming sequence.

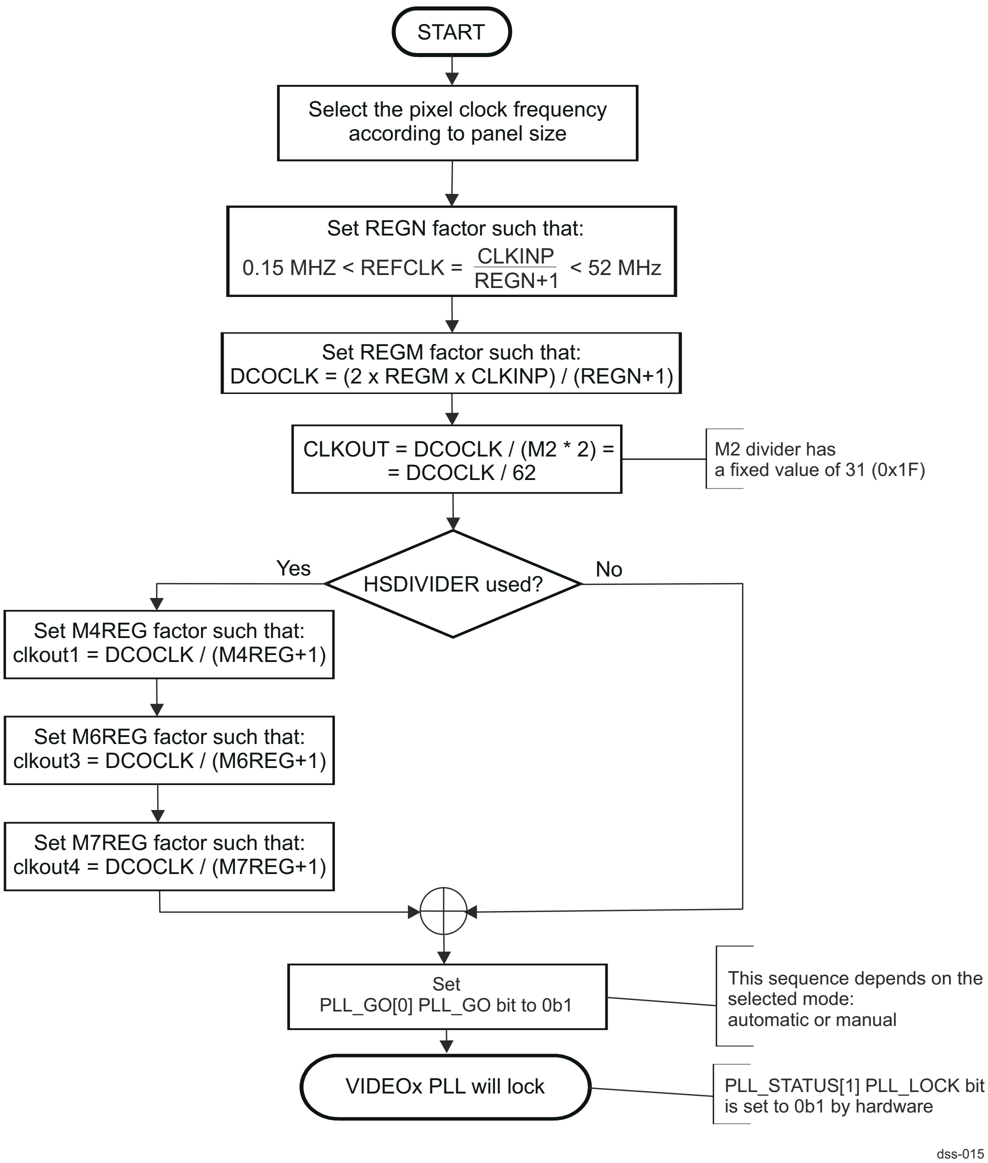


Figure 11-13. VIDEO PLL Programming Sequence

Note

- Most of the VIDEO PLL programming values are available for software flexibility, but it is not recommended to update the values in normal use. For the recommended VIDEO PLL values, see Section 11.1.3.3.9, *VIDEO PLL Recommended Values*.

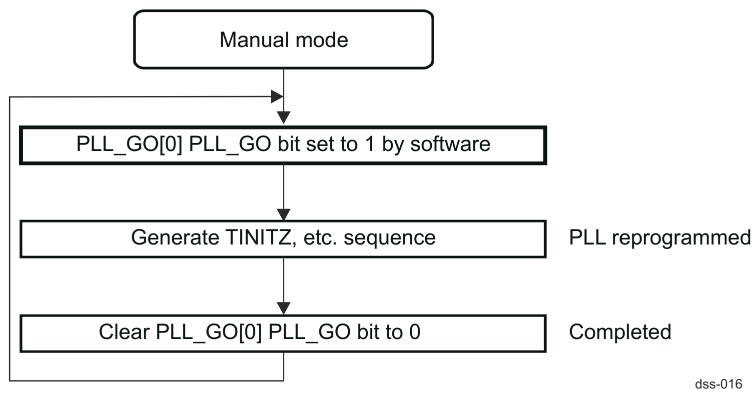
11.1.3.3.8 DPLL_VIDEO Go Sequence

In manual mode (the PLL_CONTROL[0] PLL_AUTOMODE bit is set to 0), the DPLL requires a sequence on TINITZ, TENABLE, and TENABLEDIV to update the configuration values and start the locking sequence.

When all the configuration values are programmed into the registers, the GO bit must be set. The appropriate sequence is then sent on the TINITZ, TENABLE, and TENABLEDIV pins, respecting the timing requirements of the DPLL. The PLL_GO[0] PLL_GO bit is cleared to 0 at the end of the sequence.

The TENABLEDIV signal is shared with the HSDIVIDER module so that it is programmed at the same time. In this mode, software must deassert CLKINEN by setting the PLL_CONFIGURATION2[14] PHY_CLKINEN bit to 0 and assert HSDIVBYPASS correctly by setting the PLL_CONFIGURATION2[20] HSDIVBYPASS bit to 1 to prevent uncontrolled frequencies affecting the display subsystem or the other modules using DPLL_VIDEO during PLL locking. In manual mode, the shadow register is updated anyway so that valid values are present when later selecting automatic mode.

Figure 11-14 shows the VIDEOx PLL Go flow chart in manual mode (the PLL_CONTROL[0] PLL_AUTOMODE bit is set to 0).

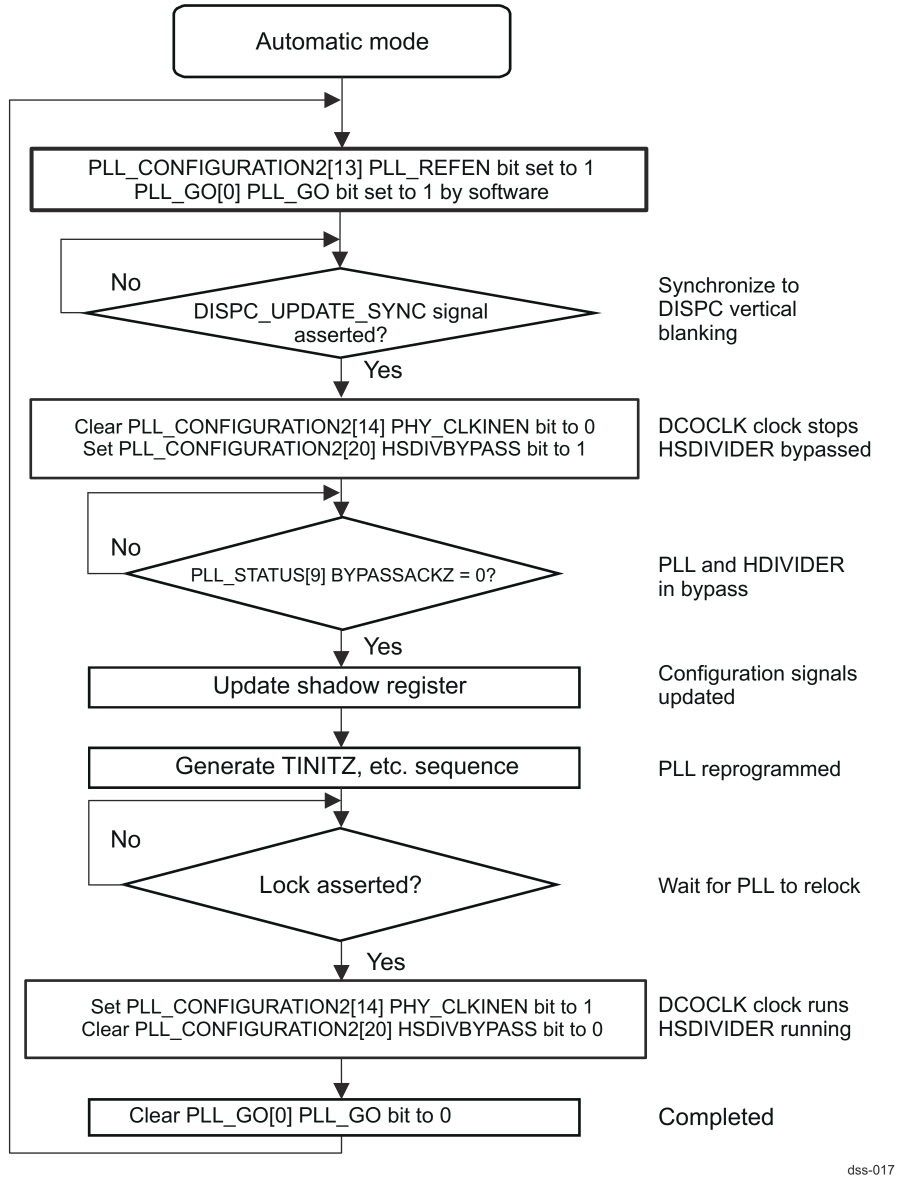


All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

Figure 11-14. VIDEO PLL Go Sequence (Manual Mode)

In automatic mode (the PLL_CONTROL[0] PLL_AUTOMODE bit is set to 1), the TINITZ, TENABLE, and TENABLEDIV sequence and the update of the PLL configuration from the PLL_CONFIGURATION2 register are deferred until the time of the front porch time signal sent by the DISPC module. This is intended to simplify the software to implement a configuration change (such as a frequency change to support a different link bandwidth). In this mode, CLKINEN, HSDIVBYPASS, and REFEN are automatically controlled and the register value is overridden.

Figure 11-15 shows the VIDEO PLL Go flow chart in automatic mode (the PLL_CONTROL[0] PLL_AUTOMODE bit is set to 1).



All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

Figure 11-15. VIDEO PLL Go Sequence (Automatic Mode)

11.1.3.3.9 DPLL_VIDEO Recommended Values

Table 11-11 shows the recommended values for PLL configuration.

Table 11-11. DPLL_VIDEO Recommended Programming Values

Field Name	Value	Description
PLL_CONTROL[4] HSDIV_SYSRESET	0	Allow power FSM to control
PLL_CONTROL[3] PLL_SYSRESET	0	Allow power FSM to control
PLL_GO[0] PLL_GO	1–>0	Write 1 when PLL is to be (re)locked with new parameters. This bit is cleared by hardware when the PLL request completes.
PLL_CONFIGURATION2[20] HSDIVBYPASS	0	PLL controls HSDIVIDER bypass
PLL_CONFIGURATION2[16] M4_CLOCK_EN or PLL_CONFIGURATION2[23] M6_CLOCK_EN or PLL_CONFIGURATION2[25] M7_CLOCK_EN	1	If PLL/HSDIVIDER is used as the clock source

Table 11-11. DPLL_VIDEO Recommended Programming Values (continued)

Field Name	Value	Description
PLL_CONFIGURATION2[14] PHY_CLKINEN	1	Enable DCOCLK clock
PLL_CONFIGURATION2[13] PLL_REFEN	1	Enable PLL reference
PLL_CONFIGURATION2[10:9] PLL_LOCKSEL	0x0	Phase lock criteria to lock the PLL
PLL_CONFIGURATION2[8] PLL_DRIFTGUARDEN	0x0	The RECAL status/interrupt must be used to decide when to perform a PLL uncalibration. No automatic uncalibration is performed.
PLL_CONFIGURATION2[6] PLL_LOWCURRSTDBY	0/1	Set to 0 for fast PLL unlock, but higher standby current. Set to 1 for leakage level standby current, but longer unlock time.
PLL_CONFIGURATION2[5] PLL_PLLLPMODE	0	Normal operation. For smaller display sizes, it may be possible to set to 1.
PLL_CONFIGURATION2[0] PLL_IDLE	0	PLL active

11.1.3.4 DPLL_HDMI Functional Description

11.1.3.4.1 DPLL_HDMI and PLLCTRL_HDMI Overview

The DPLL_HDMI control module (PLLCTRL_HDMI) uses the SCP and PMP as the primary interfaces to the HDMI module. The SCP interface is used to set the configuration of the digital phase-locked loop (DPLL) module, primarily the various counter values. The PMP is used to control the power state of the DPLL_HDMI module.

Figure 11-16 shows an overview of the PLLCTRL_HDMI and DPLL_HDMI modules in the display subsystem.

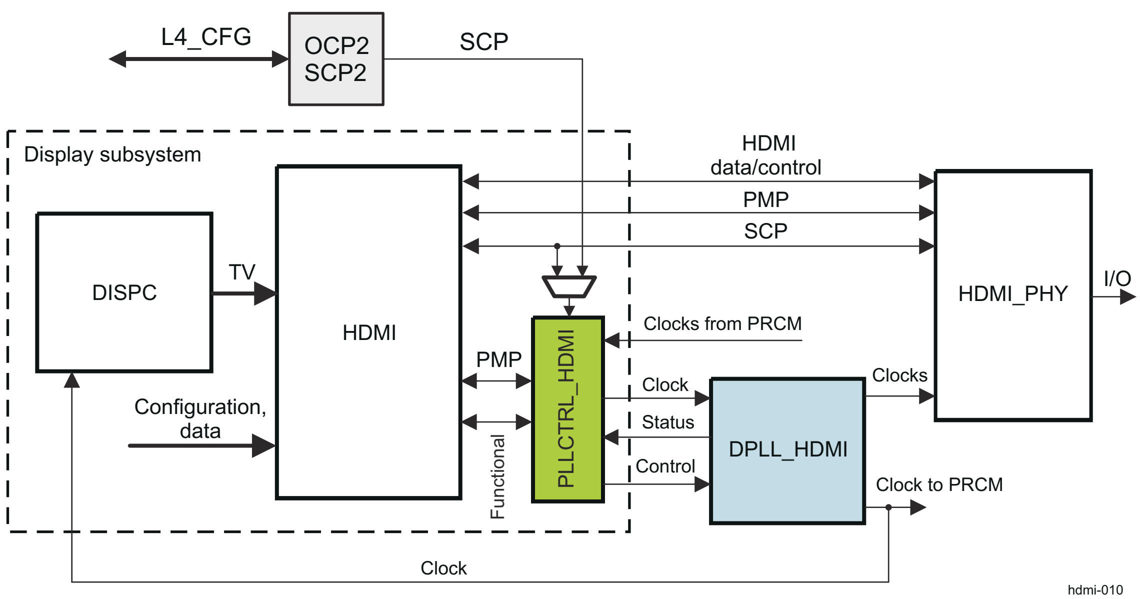


Figure 11-16. DPLL_HDMI and PLLCTRL_HDMI Overview

Note

The PLLCTRL_HDMI module does not have an interface to L3_MAIN interconnect. The programmable features are managed by registers mapped into the HDMI_WP module.

11.1.3.4.2 DPLL_HDMI and PLLCTRL_HDMI Architecture

The DPLL_HDMI uses instance of the DPLL module of type B. For information regarding DPLL types, see *PRCM.*

Figure 11-17 shows the internal reference diagram of the DPLL_HDMI and PLLCTRL_HDMI interconnection in the device.

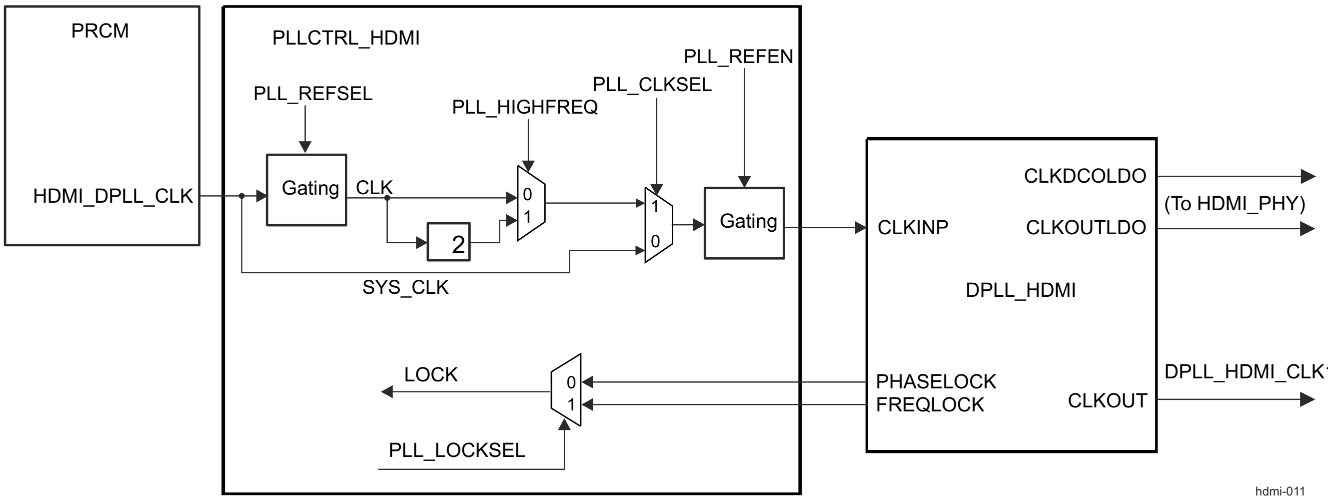


Figure 11-17. DPLL_HDMI and PLLCTRL_HDMI Reference Diagram

Figure 11-18 shows a simplified block diagram of the DPLL instance used for the DPLL_HDMI.

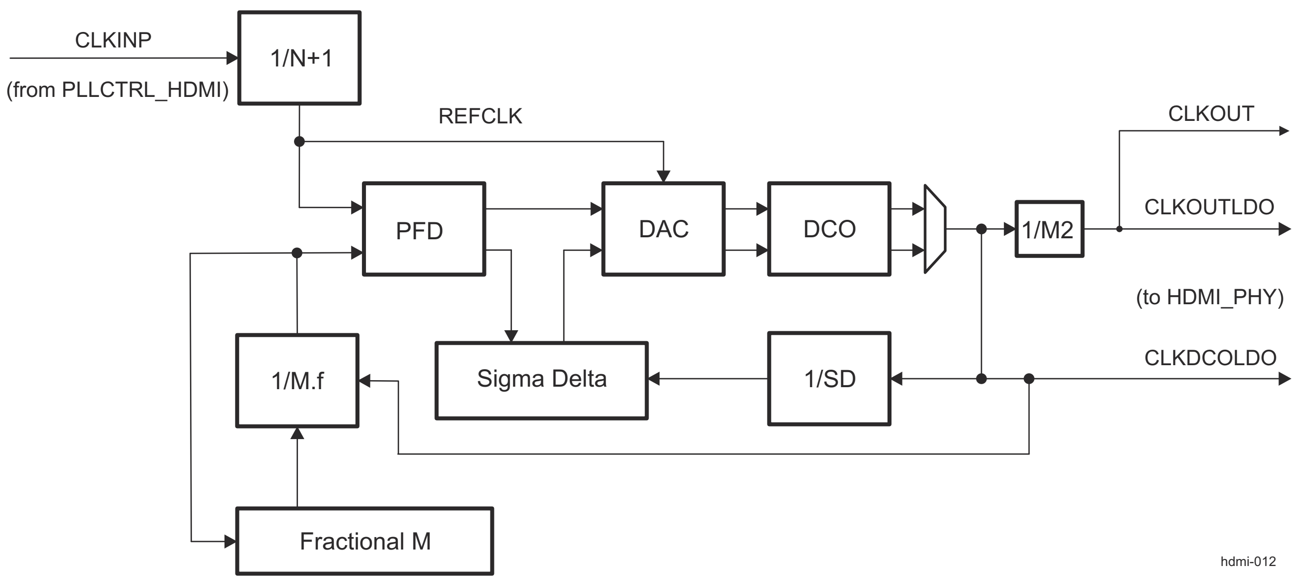


Figure 11-18. DPLL_HDMI Functional Block Diagram

The DPLL input clock CLKINP goes to a predivider N + 1. The entire loop runs on the REFCLK clock after this predivider. The value of N + 1 is controlled through the PLLCTRL_HDMI_CONFIGURATION1[8:1] PLL_REGN bit field. The CLKINP range is 0.62 to 60MHz. The REFCLK range is 0.62 to 2.5MHz.

The output clock CLKDCOLDO is synthesized by a digitally controlled oscillator (DCO block). The CLKDCOLDO frequency can be given with CLKDCOLDO = CLKINP × M/(N + 1). For this purpose, the feedback multiplier M must be configured through the PLLCTRL_HDMI_CONFIGURATION1[20:9] PLL_REGM bit field. The frequency range of the DCO must be selected via the PLLCTRL_HDMI_CONFIGURATION2[3:1] PLL_SELFREQDCO register bit-field depending on the CLKDCOLDO frequency.

The DPLL module supports fractional synthesis. That is, the frequency multiplication factor M can be programmed as fractional. This is done through the use of a sigma-delta feedback divider (M). A fractional value

(REGM.f) of 18 bits is supported, thereby enabling control for better accuracy. Programming the fractional value is done by setting the PLLCTRL_HDMI_CONFIGURATION4[17:0] PLL_REGM_F bit field. Fractional synthesis is not supported for M > 4093. For integer only division the PLL_REGM_F bit-field must be set to 0x00000.

The programming of the sigma-delta feedback divider is mandatory to get the optimal jitter performance. The value is determined by PLL_SD = ceiling ((M/(N + 1)) × CLKINP/250), and can be programmed into the PLLCTRL_HDMI_CONFIGURATION3[17:10] PLL_SD register bit-field.

11.1.3.4.3 DPLL_HDMI Operations

Note

The DPLL_HDMI must be configured before any transfer on the HDMI link. The HDMI module is fully operational only when the DPLL_HDMI runs and provides the TMDS clock.

The DPLL_HDMI configuration signals operate according to Table 11-12, which indicates the operation when the DPLL is not locked.

Table 11-12. DPLL_HDMI Operation Modes When Not Locked

DPLL_HDMI Operation Mode	Idle Bypass
Mode Description	Selects when DPLL_HDMI Bypass Clocks Are Used
PLLCTRL_HDMI_CONFIGURATION2[0] PLL_IDLE	1

When locked, the DPLL_HDMI output frequency is:

- For CLKOUT output: [(M+M.f)/(N+1)] * CLKINP * [1/M2]
- For CLKOUTLDO output: [(M+M.f)/(N+1)] * CLKINP * [1/M2]
- For CLKDCOLDO output: [(M+M.f)/(N+1)] * CLKINP

CLKINP is the input frequency in MHz. The divider and multiplier values in the above formulas can be set through the following registers:

- M frequency multiplier is programmed in the PLLCTRL_HDMI_CONFIGURATION1[20:9] PLL_REGM bit field.
- Fractional M (if fractional M divider is required by the use case) is programmed in the PLLCTRL_HDMI_CONFIGURATION4[17:0] PLL_REGM_F bit field
- N divider is programmed in the PLLCTRL_HDMI_CONFIGURATION1[8:1] PLL_REGN bit field.
- M2 divider is programmed in the PLLCTRL_HDMI_CONFIGURATION4[24:18] PLL_REGM2 bit field.
- SD divider is programmed in the PLLCTRL_HDMI_CONFIGURATION3[17:10] PLL_SD bit field.

Note

When the PLL_REGM bit field is set to 1, the DPLL enters a MN-Bypass mode. This is a low-power mode, where the DPLL gates its internal clocks and powers down the analog blocks. The CLKDCOLDO and CLKOUTLDO clock outputs go low and remain like that until the DPLL exits MN-Bypass mode (by changing the PLL_REGM bit-field to a value other than 0 or 1).

Note

The DPLL_HDMI must be configured before any data transfer to HDMI_PHY.

11.1.3.4.4 DPLL_HDMI Register Access

The configuration registers are accessed through the HDMI module register spaces using the SCP interface. This includes all the configuration signals and returning status signals.

CAUTION

All writes must be 32-bit operations because the SCP interface always transfers 32 bits; 16- or 8-bit operations can lead to unpredictable errors.

11.1.3.4.5 DPLL_HDMI Error Handling

The DPLL_HDMI lock and recalibration signals can be monitored to detect the loss of lock or the requirement to recalibrate (caused by a large temperature change since the last lock request):

- The PLLCTRL_HDMI_STATUS[1] PLL_LOCK bit gives the DPLL_HDMI lock state.
- The PLLCTRL_HDMI_STATUS[2] PLL_RECAL bit informs whether the PLL must be recalibrated.

These signals can also generate interrupts at the HDMI module level.

The PLL reference loss status signal can also be monitored:

- The PLLCTRL_HDMI_STATUS[3] PLL_LOSSREF bit informs whether the DPLL_HDMI has lost the reference clock.

11.1.3.4.6 DPLL_HDMI Software Reset

The PLLCTRL_HDMI module does not have its own software reset. It is reset by the HDMI module. Nevertheless, the reset status of the PLLCTRL_HDMI module can be monitored by reading the PLLCTRL_HDMI_STATUS[0] PLLCTRL_RESET_DONE bit.

11.1.3.4.7 DPLL_HDMI Power Management

The PLLCTRL_HDMI manages only the LDO power of the DPLL_HDMI. This is done by overriding the SYSRESET signal. All other power-management signals are integrated with the display subsystem power management.

The PMP is used to manage the LDO power of the HDMI_DPLL through the PLLCTRL_HDMI module.

Figure 11-19 shows the power states, which can be controlled through HDMI_WP_PWR_CTRL[3:2] PLL_PWR_CMD bit field.

The PLLCTRL_HDMI power status can be read through the HDMI_WP_PWR_CTRL[1:0] PLL_PWR_STATUS bit field.

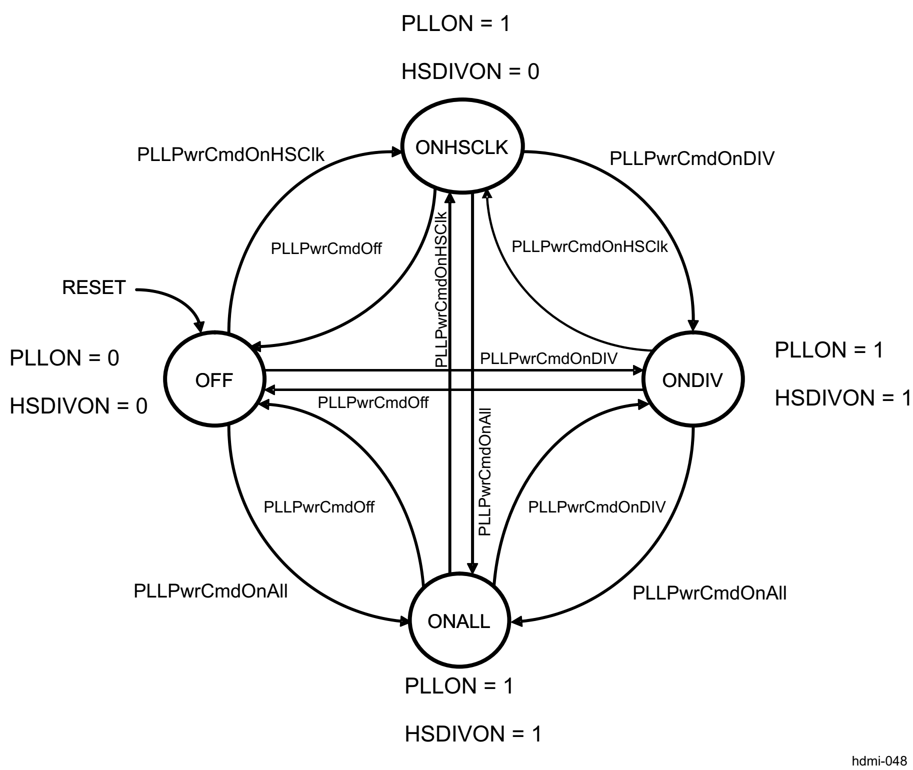


Figure 11-19. PLLCTRL_HDMI Power State Diagram

- OFF power state: PLLCTRL_HDMI is powered down. Internal clock is off.
- ONHSCLK: Power command received
- ONDIV: Clock dividers set.
- ONALL: PLLCTRL_HDMI is in active state.

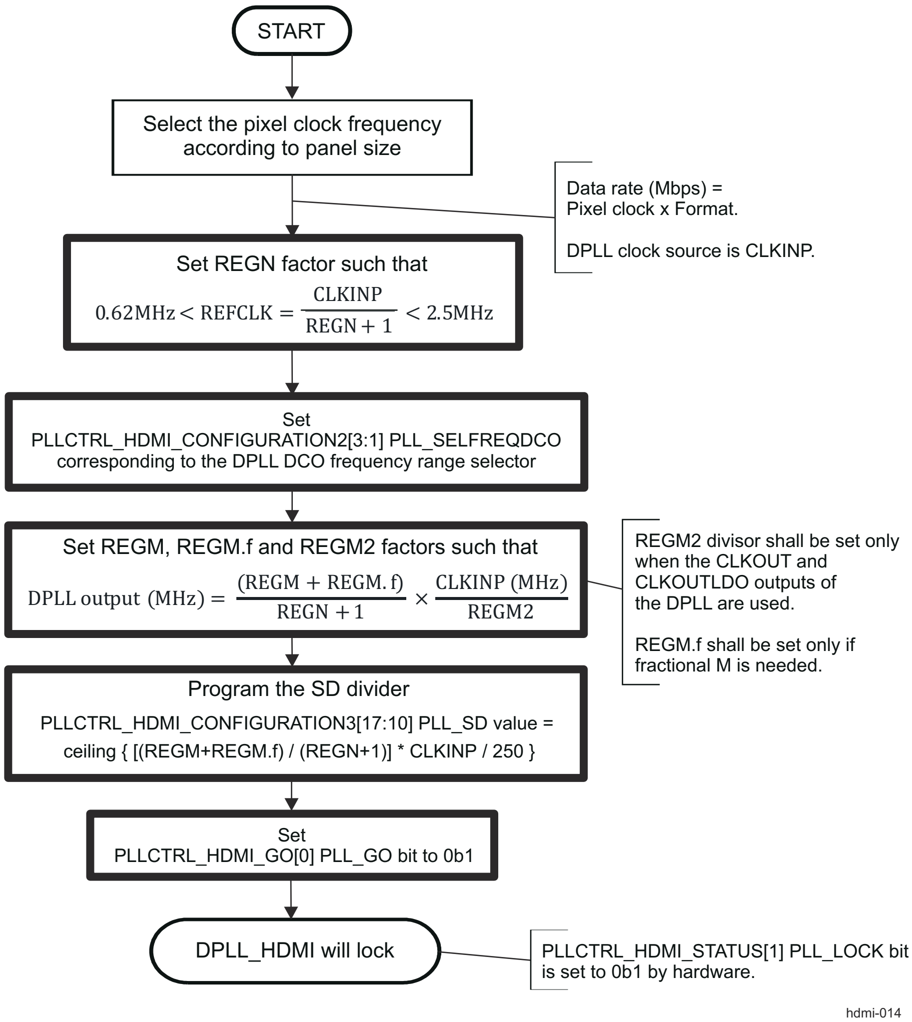
11.1.3.4.8 DPLL_HDMI Lock Sequence

The DPLL_HDMI generates the clocks to the HDMI_PHY module and the DPLL_HDMI_CLK1 clock used by the DISPC and various system modules.

The DPLL_HDMI factors must be calculated based on the required input and output frequencies, keeping the DPLL_HDMI internal reference frequency in the appropriate range:

- REGM factor is programmed in the PLLCTRL_HDMI_CONFIGURATION1[20:9] PLL_REGM bit field.
- REGM.f factor (if fractional M divider is required by the use case) is programmed in the PLLCTRL_HDMI_CONFIGURATION4[17:0] PLL_REGM_F bit field.
- REGN factor is programmed in the PLLCTRL_HDMI_CONFIGURATION1[8:1] PLL_REGN bit field.
- REGM2 factor is programmed in the PLLCTRL_HDMI_CONFIGURATION4[24:18] PLL_REGM2 bit field.
- SD divider is programmed in the PLLCTRL_HDMI_CONFIGURATION3[17:10] PLL_SD bit field.

Figure 11-20 shows the programming sequence.



All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

Figure 11-20. DPLL_HDMI Programming Sequence

Table 11-13 summarizes the registers for the DPLL_HDMI programming sequence.

Table 11-13. DPLL_HDMI Register Call Summary for HDMI PLL Programming Sequence

Register Name	Register Name	Register Name	Register Name	Register Name
PLLCTRL_HDMI_CONFIGURATION2	PLLCTRL_HDMI_GO	PLLCTRL_HDMI_STATUS	PLLCTRL_HDMI_CONFIGURATION3	PLLCTRL_HDMI_CONFIGURATION4

Note

The following should be considered for the programming sequence in Figure 11-20:

- The equation for the DPLL output applies to both CLKDCOLDO and CLKOUTLDO outputs of the DPLL_HDMI. The REGM2 divisor value must be considered only in case the CLKOUTLDO frequency needs to be calculated. Programming of REGM.f is needed only if fractional M divider is required by the use case.
- The DPLL_HDMI_CLK1 clock is sourced from the CLKDCOLDO output of the DPLL_HDMI:
  - The DPLL_HDMI_CLK1 frequency must be a multiple of the PCLK frequency (for proper settings of the PCD and LCD factors in the DISPC).
  - The DPLL_HDMI_CLK1 frequency must be lower than 186 MHz.
- The PLLCTRL_HDMI_CONFIGURATION2[3:1] PLL_SELFREQDCO register bit field should be programmed depending on the value of CLKDCOLDO = CLKINP × M/(N + 1). See Section 11.1.3.4.3, *DPLL_HDMI Operations*.
- Programming of other DPLL_HDMI parameters is available for software flexibility, but it is not recommended to update their values in normal use. For the recommended DPLL_HDMI values, see Section 11.1.3.4.10, *DPLL_HDMI Recommended Values*.

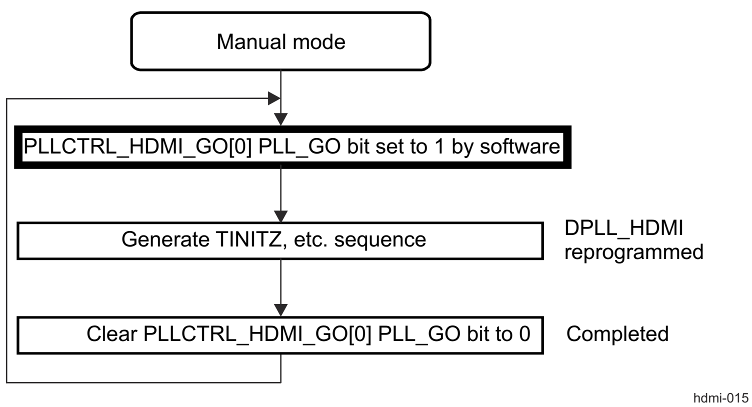
11.1.3.4.9 DPLL_HDMI Go Sequence

The DPLL_HDMI controller works in manual mode. In this mode the DPLL_HDMI requires a sequence on TINITZ, TENABLE, and TENABLEDIV to update the configuration values and start the locking sequence.

When all the configuration values are programmed into the registers, the GO bit must be set. The appropriate sequence is then sent on the TINITZ, TENABLE, and TENABLEDIV pins, respecting the timing requirements of the DPLL_HDMI. The PLLCTRL_HDMI_GO[0] PLL_GO bit is cleared to 0 at the end of the sequence.

In this mode, software must deassert CLKINEN by resetting the PLLCTRL_HDMI_CONFIGURATION2[14] PHY_CLKINEN bit to 0 to prevent uncontrolled frequencies affecting HDMI_PHY and the display subsystem during DPLL_HDMI locking.

Figure 11-21 shows the PLLCTRL HDMI GO flow chart in manual mode.



All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

Figure 11-21. DPLL_HDMI GO Sequence (Manual Mode)

11.1.3.4.10 DPLL_HDMI Recommended Values

Table 11-14 lists the main DPLL_HDMI and PLLCTRL_HDMI recommended values.

Table 11-14. DPLL_HDMI Recommended Programming Values

Field Name	Value	Description
PLLCTRL_HDMI_GO[0] PLL_GO	1 - 0	Write 1 when DPLL_HDMI is to be (re)locked with new parameters. This bit is cleared to 0 by hardware when the DPLL_HDMI request completes.

Table 11-14. DPLL_HDMI Recommended Programming Values (continued)

Field Name	Value	Description
PLLCTRL_HDMI_CONFIGURATION1[20:9] PLL_REGM	See (1)	DPLL_HDMI feedback clock divider.
PLLCTRL_HDMI_CONFIGURATION4[17:0] PLL_REGM_F	See (1)	Fractional part of the DPLL_HDMI feedback clock divider. Must be kept at 0, if integer divider only will be used.
PLLCTRL_HDMI_CONFIGURATION1[8:1] PLL_REGN	See (1)	DPLL_HDMI reference clock divider.
PLLCTRL_HDMI_CONFIGURATION2[15] BYPASSEN	0	When this bit is 0, the CLKOUT clock depends on the DPLL_HDMI configuration (DPLL locked). For small displays, it may be possible to use the functional clock, in which case this bit must be set to 1. When 1, the internal DPLL_HDMI configuration path is bypassed, and CLKINP is sent on the CLKOUT output.
PLLCTRL_HDMI_CONFIGURATION2[14] PHY_CLKINEN	1	Enable DPLL_HDMI clock output to HDMI_PHY.
PLLCTRL_HDMI_CONFIGURATION2[13] PLL_REFEN	1	Enable the DPLL_HDMI reference clock.
PLLCTRL_HDMI_CONFIGURATION2[10:9] PLL_LOCKSEL	0x0	Phase-lock criteria to lock the DPLL_HDMI.
PLLCTRL_HDMI_CONFIGURATION2[8] PLL_DRIFTGUARDEN	0	The RECAL status/interrupt must be used to decide when to perform a DPLL_HDMI uncalibration. No automatic uncalibration is performed.
PLLCTRL_HDMI_CONFIGURATION2[3:1] PLL_SELFREQDCO	See (1)	Must be programmed based on the DPLL_HDMI lock frequency.
PLLCTRL_HDMI_CONFIGURATION2[0] PLL_IDLE	0	DPLL_HDMI active.
PLLCTRL_HDMI_CONFIGURATION3[17:10] PLL_SD	See (1)	Sigma-delta divider = ceiling { [PLL_REGM / (PLL_REGN+1)] * CLKINP(MHz) / 250 }

(1) The value of the bit field must be set according to the desired clock frequency.

11.1.4 Display Subsystem Programming Guide

This section provides recommended steps for initializing the DSS module.

Table 11-15. DSS Initialization Sequence

Step	Register / Bit Field / Chapter	Value
Enable DSS clock domain		
Configure DPLL_PER and associated HS12 divider, according to the desired DSS_GFCLK frequency	Refer to *Power, Reset, and Clock Management*, section *DPLL_PER Description*	-
Enable DES_HDCP clock	Set CTRL_CORE_CONTROL_IO_2[0] DSS_DESHDCP_CLKEN register bit; Refer to *Control Module Register Manual*	0x1
Set SW wake-up transition for DSS clock domain	Set CM_DSS_CLKSTCTRL[1:0] CLKTRCTRL register bitfield; Refer to *PRCM Register Manual*	0x2
Enable (ungate) DSS_GFCLK clock	Set CM_DSS_DSS_CLKCTRL[8] OPTFCLKEN_DSSCLK register bit; Refer to *PRCM Register Manual*	0x1
Enable DSS clocks	Set CM_DSS_DSS_CLKCTRL[1:0] MODULEMODE register bitfield; Refer to *PRCM Register Manual*	0x2
Wait for DSS idle status (functional or in idle mode)	Read CM_DSS_DSS_CLKCTRL[17:16] IDLEST register bit-field; Refer to *PRCM Register Manual*	0x0 or 0x2
Make DSS DPLLs accessible via DSS address space		
Make VIDEO1, VIDEO2 and HDMI DPLLs accessible via DSS	Clear the following bits in CTRL_CORE_DSS_PLL_CONTROL register: [2] PLL_HDMI_DSS_CONTROL_DISABLE bit; [1] PLL_VIDEO2_DSS_CONTROL_DISABLE bit; [0] PLL_VIDEO1_DSS_CONTROL_DISABLE bit; Refer to *Control Module Register Manual*	0x0
Enable DSS internal SCP interface		
Enable DSS internal SCP interface for VIDEO1 and VIDEO2 DPLL register configuration	Set DSI_CLK_CTRL[14] CIO_CLK_ICG register bit	0x1
Configure DSS internal SCP interface for HDMI DPLL register configuration	Set HDMI_WP_CLK[10:8] SCP_PWR_DIV register bit-field	0x-
Configure DSS DPLLs		
Enable (ungate) VIDEO1, VIDEO2 and HDMI DPLLs source clocks	Set the following bits in CM_DSS_DSS_CLKCTRL register: [13] OPTFCLKEN_VIDEO2_CLK bit; [12] OPTFCLKEN_VIDEO1_CLK bit; [10] OPTFCLKEN_HDMI_CLK bit; Refer to *PRCM Register Manual*	0x1
Proceed with VIDEO1, VIDEO2 and HDMI DPLLs configuration	Refer to *Display Subsystem DPLL Controllers Functional Description*	-

11.1.5 Display Subsystem Register Manual

11.1.5.1 Display Subsystem Instance Summary

Table 11-16. Display Subsystem Instance Summary

Module Name	L3_MAIN Base Address	L4_CFG Base Address	Size
DSS	0x5800 0000	N/A	1 KiB
OCP2SCP2	N/A	0x4A0A 0000	16 KiB
DSI1_A	0x5800 4000	N/A	32 bit
DSI1_C	0x5800 9000	N/A	32 bit
DPLL_VIDEO1	0x5800 4300	0x4A0A 4000	32 bytes
DPLL_VIDEO2	0x5800 9300	0x4A0A 5000	32 bytes
HDMI_WP	0x5804 0000	N/A	256K bytes
DPLL_HDMI	0x5804 0200	0x4A0A 6000	1 KiB

11.1.5.2 Display Subsystem Registers

11.1.5.2.1 Display Subsystem Registers Mapping Summary

Table 11-17 summarizes the display subsystem register mapping.

Table 11-17. DSS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DSS_REVISION	R	32	0x0000 0000	0x5800 0000
RESERVED	R	32	0x0000 0010	0x5800 0010
DSS_SYSSTATUS	R	32	0x0000 0014	0x5800 0014
DSS_CTRL	RW	32	0x0000 0040	0x5800 0040
DSS_STATUS	R	32	0x0000 005C	0x5800 005C

11.1.5.2.2 Display Subsystem Register Description

Table 11-18. DSS_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5800 0000	**Instance**	DSS_MAIN_L3
Description	This register contains the DSS revision number.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See (1)

(1) TI internal data

Table 11-19. DSS_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x5800 0014	**Instance**	DSS_MAIN_L3
Description	This register provides status information about the module.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

<table>
<tr><td colspan="31">RESERVED</td><td>RESETDONE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is ongoing. Read 0x1: Reset complete	R	0x1

Table 11-20. DSS_CTRL

Address Offset	0x0000 0040		
Physical Address	0x5800 0040	**Instance**	DSS_MAIN_L3
Description	This register contains the DSS control bits.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">RESERVED</td><td>LCD3_CLK_SWITCH</td><td>RESERVED</td><td colspan="2">PARALLEL_SEL</td><td colspan="3">RESERVED</td><td>LCD2_CLK_SWITCH</td><td colspan="2">RESERVED</td><td colspan="3">F_CLK_SWITCH</td><td colspan="6">RESERVED</td><td>LCD1_CLK_SWITCH</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19	LCD3_CLK_SWITCH	DSS_CLK/DPLL_DSI1_C_CLK1 clock switch (multiplexer 10) Selects the clock source for the DISPC LCD3_CLK clock 0x0: DSS_CLK selected (from PRCM) 0x1: DPLL_DSI1_C_CLK1 selected	RW	0x0
18	RESERVED	Reserved	R	0x0
17:16	PARALLEL_SEL	Selection between LCD1, LCD2, LCD3 and TV channel out on the parallel output (multiplexer 13) 0x0: Select HDMI channel output. 0x1: Select LCD1 channel output. 0x3: Select LCD3 channel output. 0x2: Select LCD2 channel output.	RW	0x0
15:13	RESERVED	Reserved	R	0x0
12	LCD2_CLK_SWITCH	DSS_CLK clock switch (multiplexer 3) Selects the clock source for the DISPC LCD2_CLK clock 0x0: DSS_CLK selected (from PRCM) 0x1: DPLL_DSI1_B_CLK1 selected	RW	0x0
11:10	RESERVED	Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
9:7	F_CLK_SWITCH	Selects the clock source for the DISPC functional clock F_CLK 0x0: DSS_CLK selected (from PRCM) 0x1: DPLL_DSI1_A_CLK1 0x2: DPLL_DSI1_B_CLK1 0x3: DPLL_HDMI_CLK1 selected (from DPLL_HDMI) 0x4: DPLL_DSI1_C_CLK1	RW	0x0
6:1	RESERVED	Reserved	R	0x00
0	LCD1_CLK_SWITCH	DSS_CLK/DPLL_DSI1_A_CLK1 clock switch (multiplexer 2) Selects the clock source for the DISPC LCD1_CLK clock 0x0: DSS_CLK selected (from PRCM) 0x1: DPLL_DSI1_A_CLK1 selected (from VIDEO1 PLL)	RW	0x0

Table 11-21. DSS_STATUS

Address Offset	0x0000 005C		
Physical Address	0x5800 005C	**Instance**	DSS_MAIN_L3
Description	This register contains the DSS status.		
Type	R		

31:26	25:24	23:20	19:15	14:13	12:11	10:2	1:0
RESERVED	LCD3_CLK_STATUS	RESERVED	F_CLK_STATUS	RESERVED	LCD2_CLK_STATUS	RESERVED	LCD1_CLK_STATUS

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x00
25:24	LCD3_CLK_STATUS	LCD3_CLK clock selection status (multiplexer 10) indicates which clock is used by the glitch free mux selecting the source of LCD3_CLK. It is required to have the current clock and the new selected clock being running in order to be able to switch. Both clocks are used at the same time while the switch is on going. Read 0x2: DPLL_DSI1_C_CLK1 is used by DISPC as LCD3_CLK clock Read 0x1: DSS_CLK is used as LCD3_CLK Read 0x0: LCD3_CLK clock switch is on-going	R	0x1
23:20	RESERVED	Reserved	R	0x4

Bits	Field Name	Description	Type	Reset
19:15	F_CLK_STATUS	F_CLK clock selection status (multiplexer 1) indicates which clock is used by the glitch free mux selecting the source of F_CLK. It is required to have the current clock and the new selected clock being running in order to be able to switch. Both clocks are used at the same time while the switch is on going. Read 0x4: DPLL_DSI1_B_CLK1 is used by DISPC as F_CLK clock Read 0x2: DPLL_DSI1_A_CLK1 is used by DISPC as F_CLK clock Read 0x0: DSS_CLK clock switch is on-going Read 0x1: DSS_CLK is used by DISPC as F_CLK clock Read 0x8: DPLL_HDMI_CLK1 is used by DISPC as F_CLK clock Read 0x10: DPLL_DSI1_C_CLK1 is used by DISPC as F_CLK clock	R	0x01
14:13	RESERVED	Reserved	R	0x0
12:11	LCD2_CLK_STATUS	LCD2_CLK clock selection status (multiplexer 3) indicates which clock is used by the glitch free mux selecting the source of LCD2_CLK. It is required to have the current clock and the new selected clock being running in order to be able to switch. Both clocks are used at the same time while the switch is on going. Read 0x2: DPLL_DSI1_B_CLK1 is used by DISPC as LCD2_CLK clock Read 0x1: DSS_CLK is used as LCD2_CLK Read 0x0: LCD2_CLK clock switch is on-going	R	0x1
10:2	RESERVED	Reserved	R	0xA0
1:0	LCD1_CLK_STATUS	LCD1_CLK clock selection status (multiplexer 2) indicates which clock is used by the glitch free mux selecting the source of LCD1_CLK. It is required to have the current clock and the new selected clock being running in order to be able to switch. Both clocks are used at the same time while the switch is on going. Read 0x2: DPLL_DSI1_A_CLK1 is used by DISPC as LCD1_CLK clock Read 0x1: DSS_CLK is used as LCD1_CLK Read 0x0: LCD1_CLK clock switch is on-going	R	0x1

11.1.5.3 OCP2SCP2 registers

11.1.5.3.1 OCP2SCP2 Register Summary

Table 11-22. OCP2SCP2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCP2SCP2 L4_CFG Physical Address
OCP2SCP_REVISION	R	32	0x0000 0000	0x4A0A 0000
OCP2SCP_SYSCONFIG	RW	32	0x0000 0010	0x4A0A 0010
OCP2SCP_SYSSTATUS	R	32	0x0000 0014	0x4A0A 0014
OCP2SCP_TIMING	RW	32	0x0000 0018	0x4A0A 0018

11.1.5.3.2 OCP2SCP Register Description

Table 11-23. OCP2SCP_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A0A 0000	**Instance**	OCP2SCP2_CFG_L4
Description	IP Revision Identifier (X.Y.R)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See [(1)]

Table 11-24. OCP2SCP_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A0A 0010	**Instance**	OCP2SCP2_CFG_L4
Description	SYSTEM CONFIGURATION REGISTER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IDLEMODE		RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x000 0000
4:3	IDLEMODE	00 Force Idle. An idle request is acknowledged unconditionally. 01 No Idle. An idle request is never acknowledged. 10 Smart Idle. The acknowledgement to an idle request is given based on the internal activity (see 4.1.2). 11 Smart Idle Wakeup. 0x0: An idle request is acknowledged unconditionally. 0x1: An idle request is never acknowledged. Read 0x3: Reserved combination 0x2: The acknowledgement to an idle request is given based on the internal activity (see 4.1.2).	RW	0x2
2	RESERVED	Reserved.	R	0
1	SOFTRESET	Software Reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal Mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	OCP clock gating control. 0x0: Internal Interface OCP clock is free-running 0x1: Automatic internal OCP clock gating, based on the OCP interface activity	RW	1

Table 11-25. OCP2SCP_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x4A0A 0014	**Instance**	OCP2SCP2_CFG_L4
Description	System Status register.		

Table 11-25. OCP2SCP_SYSSTATUS (continued)

Type	R

31–1	0
RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Read 0x1: Reset completed Read 0x0: Internal Reset is on-going	R	1

Table 11-26. OCP2SCP_TIMING

Address Offset	0x0000 0018		
Physical Address	0x4A0A 0018	**Instance**	OCP2SCP2_CFG_L4
Description	Timing constraints for the OCP2SCP module.		
Type	RW		

31–10	9–7	6–4	3–0
RESERVED	DIVISIONRATIO	SYNC1	SYNC2

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved.	R	0x00 0000
9:7	DIVISIONRATIO(1)	Division Ration of the SCP clock in relation to OCP input clock.	RW	0x0
6:4	SYNC1	Number of SCPclock cycles defining SYNC1	RW	0x0
3:0	SYNC2	Number of SCPclock cycles defining SYNC2	RW	0x1

(1) When value “000” is programmed for the SCP clock division ratio, and the transaction to be made is a valid transaction on the SCP interface, the DIVISIONRATIO value is set internally to 0x7 (to avoid a block on the OCP interface).

11.1.5.4 DPLL_VIDEO Registers

11.1.5.4.1 DPLL_VIDEO Register Summary

Table 11-27. DPLL_VIDEO1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DPLL_VIDEO1 L3_MAIN Physical Address	DPLL_VIDEO1 L4_CFG Physical Address
PLL_CONTROL	RW	32	0x0000 0000	0x5800 4300	0x4A0A 4000
PLL_STATUS	R	32	0x0000 0004	0x5800 4304	0x4A0A 4004
PLL_GO	RW	32	0x0000 0008	0x5800 4308	0x4A0A 4008
PLL_CONFIGURATION1	RW	32	0x0000 000C	0x5800 430C	0x4A0A 400C
PLL_CONFIGURATION2	RW	32	0x0000 0010	0x5800 4310	0x4A0A 4010
PLL_CONFIGURATION3	RW	32	0x0000 0014	0x5800 4314	0x4A0A 4014
PLL_SSC_CONFIGURATION1	RW	32	0x0000 0018	0x5800 4318	0x4A0A 4018
PLL_SSC_CONFIGURATION2	RW	32	0x0000 001C	0x5800 431C	0x4A0A 401C
PLL_CONFIGURATION4	RW	32	0x0000 0020	0x5800 4320	0x4A0A 4020

Table 11-28. DPLL_VIDEO2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DPLL_VIDEO2 L3_MAIN Physical Address	DPLL_VIDEO2 L4_CFG Physical Address
PLL_CONTROL	RW	32	0x0000 0000	0x5800 9300	0x4A0A 5000
PLL_STATUS	R	32	0x0000 0004	0x5800 9304	0x4A0A 5004

Table 11-28. DPLL_VIDEO2 Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DPLL_VIDEO2 L3_MAIN Physical Address	DPLL_VIDEO2 L4_CFG Physical Address
PLL_GO	RW	32	0x0000 0008	0x5800 9308	0x4A0A 5008
PLL_CONFIGURATION1	RW	32	0x0000 000C	0x5800 930C	0x4A0A 500C
PLL_CONFIGURATION2	RW	32	0x0000 0010	0x5800 9310	0x4A0A 5010
PLL_CONFIGURATION3	RW	32	0x0000 0014	0x5800 9314	0x4A0A 5014
PLL_SSC_CONFIGURATION1	RW	32	0x0000 0018	0x5800 9318	0x4A0A 5018
PLL_SSC_CONFIGURATION2	RW	32	0x0000 001C	0x5800 931C	0x4A0A 501C
PLL_CONFIGURATION4	RW	32	0x0000 0020	0x5800 9320	0x4A0A 5020

11.1.5.4.2 DPLL_VIDEO Register Description

Table 11-29. PLL_CONTROL

Address Offset	0x0000 0000		
Physical Address	0x5800 4300 0x4A0A 4000 0x5800 9300 0x4A0A 5000	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	This register controls the PLL reset/power and modes		
Type	RW		

31:5	4	3	2	1	0
RESERVED	HSDIV_SYSRESET	PLL_SYSRESET	PLL_HALTMODE	PLL_GATEMODE	PLL_AUTOMODE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads as zero.	R	0x0000000
4	HSDIV_SYSRESET	Force HSDIVIDER SYSRESETN. Reserved when DBGSSV is 1. 0x0: HSDIVIDER SYSRESET forced active 0x1: HSDIVIDER SYSRESET controlled by power FSM	RW	1
3	PLL_SYSRESET	Force DPLL SYSRESETN. Reserved when DBGSSV is 1. 0x0: PLL SYSRESET forced active 0x1: PLL SYSRESET controlled by power FSM	RW	1
2	PLL_HALTMODE	Allow PLL to be halted if no activity. Reserved when PLLCTRL_AUTO is 0. 0x0: PLL will not be halted 0x1: PLL will be halted based on activity	RW	0
1	PLL_GATEMODE	Allow PLL clock gating for power saving Reserved when PLLCTRL_AUTO is 0. 0x0: PHY clock on 0x1: Reserved	RW	0

Bits	Field Name	Description	Type	Reset
0	PLL_AUTOMODE	Automatic update mode. If this bit is set then the configuration updates will be synchronized to DISPCUpdateSync. If this bit is clear configuration updates will be done immediately. Reserved when PLLCTRL_AUTO is 0. 0x0: Manual mode 0x1: Automatic mode	RW	0

Table 11-30. PLL_STATUS

Address Offset	0x0000 0004		
Physical Address	0x5800 4304 0x4A0A 4004 0x5800 9304 0x4A0A 5004	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	This register contains the status information		
Type	R		

31:17	16	15	14:13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PLL_TICOPWDN	PLL_LDOPWDN	BYPASSACKZ	SSC_EN_ACK	M7_CLOCK_ACK	M6_CLOCK_ACK	BYPASSACKZ_MERGED	RESERVED	M4_CLOCK_ACK	PLL_BYPASS	PLL_HIGHJITTER	RESERVED	PLL_LOSSREF	PLL_RECAL	PLL_LOCK	PLLCTRL_RESET_DONE

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reads as zero.	R	0x0000
16	PLL_TICOPWDN	PLL TICOPWDN status. Read 0x1: Internal oscillator power down Read 0x0: Internal oscillator power up	R	0
15	PLL_LDOPWDN	PLL LDOPWDN status. Read 0x1: PLL's internal LDO is power down Read 0x0: PLL's internal LDO is power up	R	0
14:13	BYPASSACKZ	State of bypass mode on PHY and HSDIVIDER. The status is shown separately for each source. Read 0x1: PLL outputs are still being used by the PHY or HSDIVIDER. Read 0x0: PHY or HSDIVIDER has switched to using the bypass clocks.	R	0x0
12	SSC_EN_ACK	Spread Spectrum Clocking acknowledge Read 0x1: Spread Spectrum Clocking active Read 0x0: Spread Spectrum Clocking inactive Note: SSC feature is supported on DPLL_VIDEO1 only, using the configurations described in the device *Spread Spectrum Clocking Configuration Application Report* (SPRABY0). SSC feature on DPLL_VIDEO2 is not supported.	R	0

Bits	Field Name	Description	Type	Reset
11	M7_CLOCK_ACK	Acknowledge for enable of sub-system clock Verify the status before selecting this source in the sub-system clock mux Read 0x0: M7 clock inactive Read 0x1: M7 clock active	R	0x0
10	M6_CLOCK_ACK	Acknowledge for enable of sub-system clock Verify the status before selecting this source in the sub-system clock mux Read 0x1: M6 clock active Read 0x0: M6 clock inactive	R	0
9	BYPASSACKZ_MERGED	Merged state of bypass mode on PHY and HSDIVIDER Read 0x1: PLL outputs are still being used by the PHY or HSDIVIDER Read 0x0: PHY and HSDIVIDER have switched to using the bypass clocks.	R	0
8	RESERVED	Reads as zero.	R	0x0000
7	M4_CLOCK_ACK	Acknowledge for enable of sub-system clock Verify the status before selecting this source in the sub-system clock mux Read 0x1: M4 clock active Read 0x0: M4 clock inactive	R	0
6	PLL_BYPASS	PLL Bypass status Read 0x1: PLL bypass Read 0x0: PLL not bypassing	R	0
5	PLL_HIGHJITTER	PLL High Jitter status Read 0x1: PLL in high jitter condition: Phase error > 24% Read 0x0: PLL in normal jitter condition	R	0
4	RESERVED	Read returns zero.	R	0
3	PLL_LOSSREF	PLL Reference Loss status Read 0x1: Reference input inactive Read 0x0: Reference input active	R	0
2	PLL_RECAL	PLL re-calibration status If this bit is active, the PLL needs to be re-calibrated Read 0x1: Recalibration is required Read 0x0: Recalibration is not required	R	0
1	PLL_LOCK	PLL Lock status See the programming guide for the use of this bit Read 0x1: PLL is locked Read 0x0: PLL is not locked	R	0
0	PLLCTRL_RESET_DONE	PLLCTRL reset done status Read 0x1: Reset has completed Read 0x0: Reset is in progress	R	0

Table 11-31. PLL_GO

Address Offset	0x0000 0008		
Physical Address	0x5800 4308 0x4A0A 4008 0x5800 9308 0x4A0A 5008	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	This register contains the GO bit		

Table 11-31. PLL_GO (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														HSDIVLOAD	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved. Write only zero for future compatibility. Reads return zero.	R	0x0000 0000
1	HSDIVLOAD	In manual mode start HSDIVIDER update sequence.	RW	0x0
0	PLL_GO	Request (re-)locking sequence of the PLL. If the AutoMode bit is set, then this will be deferred until DISPCUpdate Sync goes active 0x0: No pending action 0x1: Request PLL (re-)locking/locking pending	RW	0x0

Table 11-32. PLL_CONFIGURATION1

Address Offset	0x0000 000C		
Physical Address	0x5800 430C 0x4A0A 400C 0x5800 930C 0x4A0A 500C	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	This register contains the latched PLL and HSDIVDER configuration bits		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED					M4_CLOCK_DIV					PLL_REGM												PLL_REGN								RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED	Read returns zero.	R	0
30:26	RESERVED	Reserved	R	0x0000
25:21	M4_CLOCK_DIV	Divider value for clock source M4REG Divider value = M4_CLOCK_DIV + 1	RW	0x00
20:9	PLL_REGM	M Divider for PLL. Valid values range is from 1 to 2047. Values 2048 and above are reserved and must not be used. When the PLL_REGM bit field is set to 1, the PLL enters a MN-Bypass mode. The DCOCLK clock output goes low and remains low until the PLL exits MN-Bypass mode (by changing the PLL_REGM bit field to a value other than 0 or 1).	RW	0x000
8:1	PLL_REGN	N Divider for PLL (Reference). Divider value = PLL_REGN+1. Valid values range is from 0 to 127. Values 128 and above are reserved and must not be used.	RW	0x00
0	RESERVED	Read returns zero.	R	0

Table 11-33. PLL_CONFIGURATION2

Address Offset	0x0000 0010

Table 11-33. PLL_CONFIGURATION2 (continued)

Physical Address	0x5800 4310 0x4A0A 4010 0x5800 9310 0x4A0A 5010	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	This register contains the unlatched PLL and HSDIVDER configuration bits These bits are "shadowed" when automatic mode is selected		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						M7_CLOCK_EN	RESERVED	M6_CLOCK_EN	REFSEL		HSDIVBYPASS	RESERVED	RESERVED	RESERVED	M4_CLOCK_EN	BYPASSEN	PHY_CLKINEN	PLL_REFEN	PLL_HIGHFREQ	PLL_CLKSEL	PLL_LOCKSEL		PLL_DRIFTGUARDEN	RESERVED	PLL_LOWCURRSTBY	PLL_PLLPMODE	RESERVED	RESERVED			PLL_IDLE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Read as zero.	R	0x00
25	M7_CLOCK_EN	Enable for M7 clock source 0x0: M7 clock divider is disabled 0x1: M7 clock divider is enabled	RW	0x0
24	RESERVED	Read returns zero.	R	0
23	M6_CLOCK_EN	Enable for M6 clock source 0x0: M6 clock divider is disabled 0x1: M6 clock divider is enabled	RW	0
22:21	REFSEL	Selects the reference clock with optional divide by 2 0x0: Reserved 0x1: Reserved 0x3: Select SYSCLK reference 0x2: Reserved	RW	0x0
20	HSDIVBYPASS	Forces HSDIVIDER to bypass mode 0x0: HSDIVIDER in normal operation. Bypass controlled by PLL. 0x1: HSDIVIDER forced to bypass mode.	RW	0
19	RESERVED	Read returns zero.	R	0
18	RESERVED	Read returns zero.	R	0
17	RESERVED	Read returns zero	R	0
16	M4_CLOCK_EN	Enable for M4 clock source 0x0: Sub-system clock divider is disabled 0x1: Sub-system clock divider is enabled	RW	0
15	BYPASSEN	Selects sub-system functional clock as PHY clock source 0x0: PLL controls the PHY clock source: PLL DCO if PLL is locked Sub-system functional clock if not locked 0x1: Force sub-system functional clock to be used as the PHY clock source	RW	0

Bits	Field Name	Description	Type	Reset
14	PHY_CLKINEN	PHY clock control 0x0: PHY clock is disabled 0x1: PHY clock is enabled	RW	0
13	PLL_REFEN	PLL reference clock control 0x0: PLL reference clock disabled 0x1: PLL reference clock enabled	RW	1
12	PLL_HIGHFREQ	Enables a division of pixel clock by 2 before input to the PLL Required for pixel clock frequencies above 32 MHz (21 MHZ if N = 0) 0x0: Pixel clock is not divided 0x1: Pixel clock is divided by 2	RW	0
11	PLL_CLKSEL	Reference clock selection 0x0: Selects SYSCLK as PLL reference clock 0x1: Selects Pixel Clock (PCLK) as PLL reference clock	RW	0
10:9	PLL_LOCKSEL	Selects the lock criteria for the PLL 0x0: Phase Lock 0x1: Frequency Lock 0x2: Spare	RW	0x0
8	PLL_DRIFTGUARDEN	PLL DRIFTGUARDEN 0x0: Only RECAL flag is asserted in case of temperature drift. The programmer should take appropriate action. 0x1: Temperature drift will initiate automatic recalibration. RECAL flag will be asserted while this is taking place.	RW	0
7	RESERVED		R	0
6	PLL_LOWCURRSTBY	PLL LOW CURRENT STANDBY 0x0: LOWCURRSTBY is not selected 0x1: LOWCURRSTBY is selected	RW	0
5	PLL_PLLLPMODE	Select the power / performance of the PLL 0x0: Full performance, minimized jitter 0x1: Reduced power, increased jitter	RW	0
4	RESERVED	Reads as zero.	R	0
3:1	RESERVED	Reserved	R	0
0	PLL_IDLE	PLL IDLE: 0x0: IDLE is not selected 0x1: IDLE is selected	RW	0

Table 11-34. PLL_CONFIGURATION3

Address Offset	0x0000 0014		
Physical Address	0x5800 4314 0x4A0A 4014 0x5800 9314 0x4A0A 5014	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	HSDIVIDER configuration bits for the M5 and M6 dividers		
Type	RW		

31–10	9–5	4–0
RESERVED	M7_CLOCK_DIV	M6_CLOCK_DIV

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0000
9:5	M7_CLOCK_DIV	Divider value for M7 divider. Divider value = M7_CLOCK_DIV + 1	RW	0x00
4:0	M6_CLOCK_DIV	Divider value for M6 divider. Divider value = M6_CLOCK_DIV + 1	RW	0x00

Table 11-35. PLL_SSC_CONFIGURATION1

Address Offset	0x0000 0018		
Physical Address	0x5800 4318 0x4A0A 4018 0x5800 9318 0x4A0A 5018	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	Configuration for PLL Spread Spectrum Clocking modulation. Note: SSC feature is supported on DPLL_VIDEO1 only, using the configurations described in the device *Spread Spectrum Clocking Configuration Application Report* (SPRABY0). SSC feature on DPLL_VIDEO2 is not supported.		
Type	RW		

31–3	2	1	0
RESERVED	DOWNSPREAD	RESERVED	EN_SSC

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	RESERVED	R	0x0000 0000
2	DOWNSPREAD	Forces the clock spreading only in the down spectrum. 0x0: Clock spreading not forced. 0x1: Spectrum spreading only in down direction.	RW	0
1	RESERVED	Reserved. Reads return 0.	R	0
0	EN_SSC	Spread Spectrum Clocking enable 0x0: Spread Spectrum Clocking disabled 0x1: Spread Spectrum Clocking enabled	RW	0

Table 11-36. PLL_SSC_CONFIGURATION2

Address Offset	0x0000 001C		
Physical Address	0x5800 431C 0x4A0A 401C 0x5800 931C 0x4A0A 501C	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	Configuration for PLL Spread Spectrum Clocking modulation. Note: SSC feature is supported on DPLL_VIDEO1 only, using the configurations described in the device *Spread Spectrum Clocking Configuration Application Report* (SPRABY0). SSC feature on DPLL_VIDEO2 is not supported.		
Type	RW		

31	30	29–20	19–0
RESERVED	DELTAM2	MODFREQDIVIDER	DELTAM

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reads as zero	R	0x0
30	DELTAM2	MSB of DeltaM control bus.	RW	0x0
29:20	MODFREQDIVIDER	Modulation Frequency Divider (ModFreqDivider) control for SSC. The ModFreqDivider is split into Mantissa and 2^Exponent(ModFreqDivider = ModFreqDividerMantissa * 2^ModFreqDividerExponent). - Bits [29:23] define the Mantissa. - Bits [22:20] define the Exponent.	RW	0x000
19:0	DELTAM	DeltaM control for SSC. Split into integer and fractional parts. - Bits [19:18] define the integer part. - Bits [17:0] define the fractional part.	RW	0x00000

Table 11-37. PLL_CONFIGURATION4

Address Offset	0x0000 0020		
Physical Address	0x5800 4320 0x4A0A 4020 0x5800 9320 0x4A0A 5020	**Instance**	DPLL_VIDEO1_MAIN_L3 DPLL_VIDEO1_CFG_L4 DPLL_VIDEO2_MAIN_L3 DPLL_VIDEO2_CFG_L4
Description	Allows setting the fractional M divider and M2 divider for PLL.		
Type	RW		

31:25	24:18	17:0
RESERVED	PLL_REGM2	PLL_REGM_F

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reads as zero	R	0x0
24:18	PLL_REGM2	M2 divider to configure PLL REGM2. **NOTE**: In this device, M2 divider is hardcoded in HW at 31 (0x1F).	RW	0x1
17:0	PLL_REGM_F	Fractional part of M divider. **NOTE**: The feature is not supported in this device.	RW	0x0

11.1.5.5 DPLL_HDMI Registers

11.1.5.5.1 DPLL_HDMI Registers Mapping Summary

Table 11-38. DPLL_HDMI Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DPLL_HDMI L3_MAIN Physical Address	DPLL_HDMI L4_CFG Physical Address
PLLCTRL_HDMI_CONTROL	RW	32	0x0000 0000	0x5804 0200	0x4A0A 6000
PLLCTRL_HDMI_STATUS	R	32	0x0000 0004	0x5804 0204	0x4A0A 6004
PLLCTRL_HDMI_GO	RW	32	0x0000 0008	0x5804 0208	0x4A0A 6008
PLLCTRL_HDMI_CONFIGURATION1	RW	32	0x0000 000C	0x5804 020C	0x4A0A 600C
PLLCTRL_HDMI_CONFIGURATION2	RW	32	0x0000 0010	0x5804 0210	0x4A0A 6010
PLLCTRL_HDMI_CONFIGURATION3	RW	32	0x0000 0014	0x5804 0214	0x4A0A 6014
PLLCTRL_HDMI_SSC_CONFIGURATION1 (1)	RW	32	0x0000 0018	0x5804 0218	0x4A0A 6018
PLLCTRL_HDMI_SSC_CONFIGURATION2 (1)	RW	32	0x0000 001C	0x5804 021C	0x4A0A 601C
PLLCTRL_HDMI_CONFIGURATION4	RW	32	0x0000 0020	0x5804 0220	0x4A0A 6020

(1) SSC feature is not supported.

11.1.5.5.2 DPLL_HDMI Register Description

Table 11-39. PLLCTRL_HDMI_CONTROL

Address Offset	0x0000 0000

Table 11-39. PLLCTRL_HDMI_CONTROL (continued)

Physical Address	0x5804 0200 0x4A0A 6000	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	This register controls the PLL reset/power and modes		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											HSDIV_SYSRESETN	PLL_SYSRESETN	RESERVED		

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads as zero.	R	0x0
4	HSDIV_SYSRESETN	Force HSDIVIDER SYSRESETN. 0x0: HSDIVIDER SYSRESET forced active 0x1: HSDIVIDER SYSRESET controlled by power FSM	RW	0x1
3	PLL_SYSRESETN	Force SYSRESETN. 0x0: DPLL_HDMI SYSRESET forced active 0x1: DPLL_HDMI SYSRESET controlled by power FSM	RW	0x1
2:0	RESERVED	Reserved	R	0x0

Table 11-40. PLLCTRL_HDMI_STATUS

Address Offset	0x0000 0004		
Physical Address	0x5804 0204 0x4A0A 6004	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	This register contains the status information		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			SSC_EN_ACK	RESERVED		BYPASSACKZ_MERGED	RESERVED		PLL_BYPASS	PLL_HIGHJITTER	RESERVED	PLL_LOSSREF	PLL_RECAL	PLL_LOCK	PLLCTRL_RESET_DONE

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reads as zero.	R	0x0
12	SSC_EN_ACK	Spread Spectrum Clocking acknowledge 0x0: Spread Spectrum Clocking inactive 0x1: Spread Spectrum Clocking active Note: SSC feature is not supported	R	0x0
11:10	RESERVED	Read returns zero.	R	0x0

Bits	Field Name	Description	Type	Reset
9	BYPASSACKZ_MERGED	Merged state of bypass mode on HDMI_PHY 0x0: HDMI_PHY has switched to using the bypass clocks. 0x1: DPLL_HDMI outputs are still being used by the HDMI_PHY	R	0x0
8:7	RESERVED	Read returns zero.	R	0x0
6	PLL_BYPASS	DPLL_HDMI Bypass status 0x0: DPLL_HDMI not bypassing 0x1: DPLL_HDMI bypass	R	0x0
5	PLL_HIGHJITTER	DPLL_HDMI High Jitter status 0x0: DPLL_HDMI in normal jitter condition 0x1: DPLL_HDMI in high jitter condition: Phase error 24%	R	0x0
4	RESERVED	Read returns zero.	R	0x0
3	PLL_LOSSREF	DPLL_HDMI Reference Loss status 0x0: Reference input active 0x1: Reference input inactive	R	0x0
2	PLL_RECAL	DPLL_HDMI re-calibration status If this bit is active, the DPLL_HDMI needs to be re-calibrated 0x0: Recalibration is not required 0x1: Recalibration is required	R	0x0
1	PLL_LOCK	DPLL_HDMI Lock status See the programming guide for the use of this bit 0x0: DPLL_HDMI is not locked 0x1: DPLL_HDMI is locked	R	0x0
0	PLLCTRL_RESET_DONE	DPLL_HDMI reset done status 0x0: Reset is in progress 0x1: Reset has completed	R	0x0

Table 11-41. PLLCTRL_HDMI_GO

Address Offset	0x0000 0008		
Physical Address	0x5804 0208 0x4A0A 6008	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	This register contains the GO bit		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved. Wirte only zero for future compatibility. Reads return zero.	R	0x0
0	PLL_GO	Request (re-)locking sequence of the DPLL_HDMI. If the AutoMode bit is set, then this will be deferred until DISPC Update Sync goes active 0x0: No pending action 0x1: Request DPLL_HDMI (re-)locking/locking pending	RW	0x0

Table 11-42. PLLCTRL_HDMI_CONFIGURATION1

Address Offset	0x0000 000C

Table 11-42. PLLCTRL_HDMI_CONFIGURATION1 (continued)

Physical Address	0x5804 020C 0x4A0A 600C	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	This register contains the latched PLL and HSDIVDER configuration bits		
Type	RW		

31:21	20:9	8:1	0
RESERVED	PLL_REGM	PLL_REGN	RESERVED

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved.	R	0x0
20:9	PLL_REGM	M Divider for DPLL_HDMI.	RW	0x0
8:1	PLL_REGN	N Divider for DPLL_HDMI (Reference). Divider value = PLL_REGN+1.	RW	0x0
0	RESERVED	Reserved.	R	0x0

Table 11-43. PLLCTRL_HDMI_CONFIGURATION2

Address Offset	0x0000 0010		
Physical Address	0x5804 0210 0x4A0A 6010	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	This register contains the unlatched PLL and HSDIVDER configuration bits These bits are 'shadowed' when automatic mode is selected		
Type	RW		

31:23	22:21	20	19:16	15	14	13	12	11	10:9	8	7	6	5	4	3:1	0
RESERVED	REFSEL	HSDIVBYPASS	RESERVED	BYPASSEN	PHY_CLKINEN	PLL_REFEN	PLL_HIGHFREQ	PLL_CLKSEL	PLL_LOCKSEL	PLL_DRIFTGUARDEN	RESERVED	PLL_LOWCURRSTBY	PLL_PLLPMODE	RESERVED	PLL_SELFREQDCO	PLL_IDLE

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved.	R	0x0
22:21	REFSEL	Selects the reference clock with optional divide by 2 0x0: Select PCLK reference 0x1: Select REF1 reference 0x3: Select SYSCLK reference 0x2: Select REF2 Reference	RW	0x0
20	HSDIVBYPASS	Forces HSDIVIDER to bypass mode 0x0: HSDIVIDER in normal operation. Bypass controlled by DPLL_HDMI 0x1: HSDIVIDER forced to bypass mode.	RW	0x0
19:16	RESERVED	Reserved.	R	0x0

Bits	Field Name	Description	Type	Reset
15	BYPASSEN	Selects sub-system functional clock as PHY clock source 0x0: DPLL_HDMI controls the PHY clock source: PLL DCO if DPLL_HDMI is locked Sub-system functional clock if not locked 0x1: Force sub-system functional clock to be used as the PHY clock source	RW	0x0
14	PHY_CLKINEN	PHY clock control 0x0: PHY clock is disabled 0x1: PHY clock is enabled	RW	0x0
13	PLL_REFEN	DPLL_HDMI reference clock control 0x0: DPLL_HDMI reference clock disabled 0x1: DPLL_HDMI reference clock enabled	RW	0x1
12	PLL_HIGHFREQ	Enables a division of pixel clock by 2 before input to the DPLL_HDMI Required for pixel clock frequencies above 32 MHz (21 MHZ if N = 0) 0x0: Pixel clock is not divided 0x1: Pixel clock is divided by 2	RW	0x0
11	PLL_CLKSEL	Reference clock selection 0x0: Selects SYSCLK as DPLL_HDMI reference clock 0x1: Selects Pixel Clock (PCLK) as DPLL_HDMI reference clock	RW	0x0
10:9	PLL_LOCKSEL	Selects the lock criteria for the DPLL_HDMI 0x0: Phase Lock 0x1: Frequency Lock 0x2: Spare	RW	0x0
8	PLL_DRIFTGUARDEN	DPLL_HDMI DRIFTGUARDEN 0x0: Only RECAL flag is asserted in case of temperature drift. The programmer should take appropriate action. 0x1: Temperature drift will initiate automatic recalibration. RECAL flag will be asserted while this is taking place.	RW	0x0
7	RESERVED	Reserved.	R	0x0
6	PLL_LOWCURRSTBY	DPLL_HDMI LOW CURRENT STANDBY 0x0: LOWCURRSTBY is not selected 0x1: LOWCURRSTBY is selected	RW	0x0
5	PLL_PLLLPMODE	Select the power / performance of the DPLL_HDMI 0x0: Full performance, minimised jitter 0x1: Reduced power, increased jitter	RW	0x0
4	RESERVED	Reads as zero.	R	0x0
3:1	PLL_SELFREQDCO	DCO frequency range selector for DPLL_HDMI 0x2: Set if DCO frequency is between 750MHz and 1500MHz 0x4: Set if DCO frequency is between 1250MHz and 2500MHz Others: Reserved	RW	0x4
0	PLL_IDLE	DPLL_HDMI IDLE: 0x0: IDLE is not selected 0x1: IDLE is selected	RW	0x0

Table 11-44. PLLCTRL_HDMI_CONFIGURATION3

Address Offset	0x0000 0014

Table 11-44. PLLCTRL_HDMI_CONFIGURATION3 (continued)

Physical Address	0x5804 0214 0x4A0A 6014	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	HSDIVIDER configuration bits for the M5 and M6 dividers		
Type	RW		

31–18	17–10	9–0
RESERVED	PLL_SD	RESERVED

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved.	R	0x0
17:10	PLL_SD	Sigma delta divider setting for DPLL_HDMI based on the DPLL_HDMI lock configuration.	RW	0x0
9:0	RESERVED	Reserved.	R	0x0

Table 11-45. PLLCTRL_HDMI_SSC_CONFIGURATION1

Address Offset	0x0000 0018		
Physical Address	0x5804 0218 0x4A0A 6018	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	Configuration for PLL Spread Spectrum Clocking modulation. Note: SSC feature is not supported.		
Type	RW		

31–3	2	1	0
RESERVED	DOWNSPREAD	RESERVED	EN_SSC

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved.	R	0x0
2	DOWNSPREAD	Forces the clock spreading only in the down spectrum. 0x0: Clock spreading not forced. 0x1: Spectrum spreading only in down direction.	RW	0x0
1	RESERVED	Reserved.	R	0x0
0	EN_SSC	Spread Spectrum Clocking enable 0x0: Spread Spectrum Clocking disabled 0x1: Spread Spectrum Clocking enabled	RW	0x0

Table 11-46. PLLCTRL_HDMI_SSC_CONFIGURATION2

Address Offset	0x0000 001C		
Physical Address	0x5804 021C 0x4A0A 601C	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	Note: SSC feature is not supported.		
Type	RW		

31	30	29–20	19–0
RESERVED	DELTAM2	MODFREQDIVIDER	DELTAM

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reads as zero	R	0x0
30	DELTAM2	MSB of DeltaM control bus.	RW	0x0
29:20	MODFREQDIVIDER	Modulation Frequency Divider (ModFreqDivider) control for dithering.The ModFreqDivider is split into Mantissa and $2^{Exponent}$ (ModFreqDivider = ModFreqDividerMantissa * $2^{ModFreqDividerExponent}$). Bits [29:23] define the Mantissa Bits [22:20] define the Exponent	RW	0x0
19:0	DELTAM	DeltaM control for dithering. Split into integer and fractional part. Bits [19:18] define the integer part Bits [17:0] define the fractional part	RW	0x0

Table 11-47. PLLCTRL_HDMI_CONFIGURATION4

Address Offset	0x0000 0020		
Physical Address	0x5804 0220 0x4A0A 6020	**Instance**	DPLL_HDMI_MAIN_L3 DPLL_HDMI_CFG_L4
Description	Allows setting the fractional M divider and M2 divider for PLL.		
Type	RW		

31:25	24:18	17:0
RESERVED	PLL_REGM2	PLL_REGM_F

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reads as zero.	R	0x0
24:18	PLL_REGM2	M2 divider to configure DPLL_HDMI M2 divider factor.	RW	0x1
17:0	PLL_REGM_F	Fractional part of M divider.	RW	0x0

11.1.5.6 HDMI_WP Registers

11.1.5.6.1 HDMI_WP Registers Mapping Summary

Table 11-48 summarizes the mapping of the HDMI WP registers.

Table 11-48. HDMI_WP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	HDMI_WP L3_MAIN Physical Address
RESERVED	R	32	0x0000 0000	0x5804 0000
RESERVED	R	32	0x0000 0010	0x5804 0010
RESERVED	R	32	0x0000 0020	0x5804 0020
RESERVED	R	32	0x0000 0024	0x5804 0024
RESERVED	R	32	0x0000 0028	0x5804 0028
RESERVED	R	32	0x0000 002C	0x5804 002C
RESERVED	R	32	0x0000 0030	0x5804 0030
RESERVED	R	32	0x0000 0034	0x5804 0034
HDMI_WP_PWR_CTRL	RW	32	0x0000 0040	0x5804 0040
RESERVED	R	32	0x0000 0044	0x5804 0044
RESERVED	R	32	0x0000 0050	0x5804 0050
RESERVED	R	32	0x0000 0060	0x5804 0060
RESERVED	R	32	0x0000 0068	0x5804 0068
RESERVED	R	32	0x0000 006C	0x5804 006C
HDMI_WP_CLK	RW	32	0x0000 0070	0x5804 0070

Table 11-48. HDMI_WP Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	HDMI_WP L3_MAIN Physical Address
RESERVED	R	32	0x0000 0080	0x5804 0080
RESERVED	R	32	0x0000 0084	0x5804 0084
RESERVED	R	32	0x0000 0088	0x5804 0088
RESERVED	R	32	0x0000 008C	0x5804 008C
RESERVED	R	32	0x0000 0080	0x5804 0080
RESERVED	R	32	0x0000 0090	0x5804 0090
RESERVED	R	32	0x0000 0094	0x5804 0094

11.1.5.6.2 HDMI_WP Register Description

Table 11-49. HDMI_WP_PWR_CTRL

Address Offset	0x0000 0040		
Physical Address	0x5804 0040	**Instance**	HDMI_WP_MAIN_L3
Description	Power control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED				PLL_PWR_CMD		PLL_PWR_STATUS	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved.	R	0x0
7:4	RESERVED		R	0x0
3:2	PLL_PWR_CMD	Command for power control of the HDMI PLL Control module 0x0: Command to change to OFF state (PLL_PWR_CMD_OFF signal) 0x1: Command to change to ON state for PLL (DCOCLK is power down) (PLL_PWR_CMD_ON_HS_CLK signal) 0x3: Command to change to ON state for PLL (no DCOCLKLDO/CLKDCOLDO clock output to the HDMI-PHY) (PLL_PWR_CMD_ON_DIV signal) 0x2: Command to change to ON state for PLL (PLL_PWR_CMD_ON_ALL signal)	RW	0x0
1:0	PLL_PWR_STATUS	Status of the power control of the HDMI PLL Control module 0x0: HDMI PLL Control module in OFF state 0x1: HDMI PLL Control module in ON state for PLL 0x3: HDMI PLL Control module in ON state for PLL (no clock output to the HDMI-PHY) 0x2: HDMI PLL Control module in ON state for PLL	R	0x0

Table 11-50. HDMI_WP_CLK

Address Offset	0x0000 0070		
Physical Address	0x5804 0070	**Instance**	HDMI_WP_MAIN_L3
Description	Configuration of clocks		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SCP_PWR_DIV	RESERVED

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x00 0000
10:8	SCP_PWR_DIV	Defines the divisor value to be used for the generation of the SCP_PWR clock (up to 66.5MHz) from the input interface clock (up to 266MHz). 0x0 means gated 0x1 means free-running The valid values are from 0 to 7. In case of interface access to register through SCP interface, if the SCP_PWR clock is gated, the HW automatically generates the clock by using a divisor of 7 and updates the bit-field with the value 7. It is then software responsibility to change the value at any time in order to improve SCP latency when accessing the registers in the HDMI_PHY and PLLCTRL_HDMI by reducing the value.	RW	0x0
7:0	RESERVED		R	0x0

11.1.5.7 DSI Registers

Note

This family of devices does not support DSI functionality, but the registers documented in this section are required for the configuration of DPLL_VIDEO1 (via DSI1_A registers) and DPLL_VIDEO2 (via DSI1_C registers).

11.1.5.7.1 DSI Register Summary

Table 11-51. DSI1_A Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSI1_A L3_MAIN Physical Address
DSI_CLK_CTRL	RW	32	0x0000 0054	0x5800 4054

Table 11-52. DSI1_C Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSI1_C L3_MAIN Physical Address
DSI_CLK_CTRL	RW	32	0x0000 0054	0x5800 9054

11.1.5.7.2 DSI Register Description

Table 11-53. DSI_CLK_CTRL

Address Offset	0x0000 0054		
Physical Address	0x5800 4054 0x5800 9054	**Instance**	DSI1_A_MAIN_L3 DSI1_C_MAIN_L3
Description	CLOCK CONTROL This register controls the CLOCK GENERATION. The register can be modified only when IF_EN is reset.		
Type	RW		

31	30	29–15	14	13–0
PLL_PWR_CMD	PLL_PWR_STATUS	RESERVED	CIO_CLK_ICG	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	PLL_PWR_CMD	Command for power control of the DSI PLL Control Module 0x0: Command to change to OFF state 0x1: Command to change to ON state for PLL only (HSDIVISER is OFF) 0x2: Command to change to ON state for both PLL and HSDIVISER 0x3: Command to change to ON state for both PLL and HSDIVISER (no clock output to the DSI PHY)	RW	0x0
29:28	PLL_PWR_STATUS	Status of the power control of the DSI PLL Control module Read 0x0: DSI PLL Control module in OFF state Read 0x1: DSI PLL Control module in ON state for PLL only (HSDIVISER is OFF) Read 0x2: DSI PLL Control module in ON state for both PLL and HSDIVISER Read 0x3: DSI PLL Control module in ON state for both PLL and HSDIVISER (no clock output to the DSI PHY)	R	0x0
27:15	RESERVED	Reserved	R	0x0000
14	CIO_CLK_ICG	Gates SCPClk clock provided to DSI-PHY and PLL-CTRL module. 0x0: Disabled. SCPClk is not generated. It remains at 0. 0x1: Enabled. SCPClk is generated (OCP_CLK/4)	RW	0
13:0	RESERVED	Reserved	R	0x0001

11.2 Display Controller

11.2.1 DISPC Overview

Figure 11-22 shows a block diagram of the display controller (DISPC) within the display subsystem.

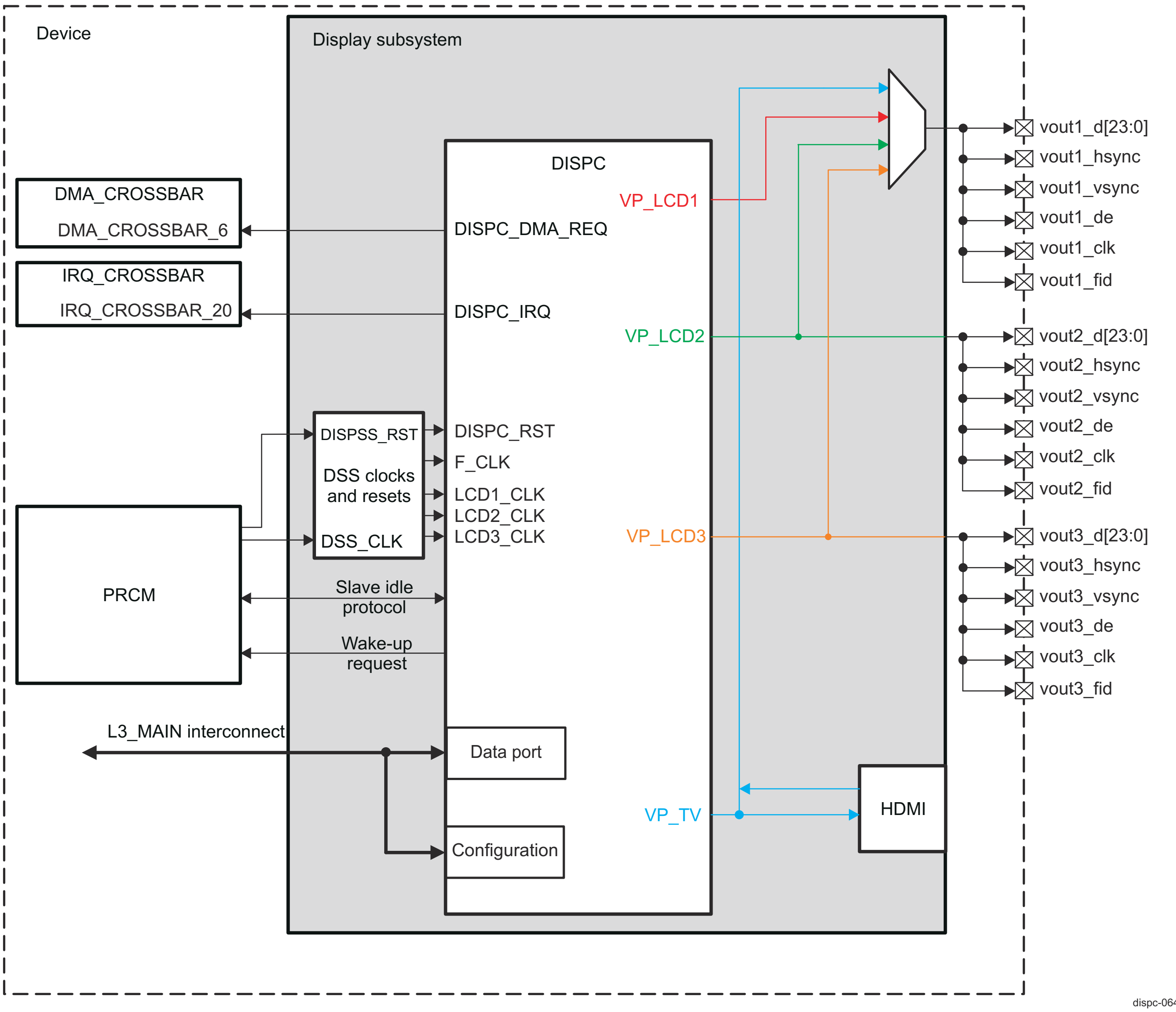


Figure 11-22. DISPC Overview

The DISPC includes the following main features:

- Five pipelines for processing:
  - One Graphics (GFX):
    - Pixel formats: ARGB16-4444, xRGB12-4444, RGBA16-4444, RGBx12-4444, RGB16-565, ARGB16-1555, xRGB16-1555, ARGB32-8888, RGBA32-8888, xRGB32-8888, RGBx32-8888, xRGB24-8888, RGBx24-8888, BGRA32-8888, RGB24-888 (packed), where x means that the corresponding bits in the container are not used
    - Premultiplied ARGB and RGBA formats
    - Selection of the color depth expansion from ARGB16-4444, RGBA16-4444, and ARGB16-1555 to ARGB32-8888, and from xRGB12-4444, RGBx12-4444, and xRGB16-1555 to xRGB32-8888 (replication of the most significant bits [MSBs] or adding 0s)
    - Support for antiflicker on RGB pixel formats using 3-tap filter
  - Three Video pipelines (VID1, VID2, and VID3):
    - Pixel formats: ARGB16-4444, xRGB12-4444, RGBx12-4444, RGBA12-3333, RGBA16-4444, RGB16-565, ARGB16-1555, xRGB16-1555, ARGB32-8888, RGBA32-8888, xRGB32-8888,

        RGBx32-8888, RGB24-888, BGRA32-8888, YUV4:2:2-UYVY, YUV4:2:2-YUV2, YUV4:2:0-NV12, and YUV4:2:0-NV21 (where x means that the corresponding bits in the container are not used)
    - Premultiplied ARGB and RGBA formats
    - Selection of the color depth expansion from ARGB16-4444, RGBA16-4444, and ARGB16-1555 to ARGB32-8888, and from xRGB12-4444, RGBx12-4444, and xRGB16-1555 to xRGB32-8888 (replication of the MSBs or adding 0s)
    - Programmable poly-phase filter:
      – Independent horizontal and vertical resampling: Upsampling (up to x8) and downsampling (down to 1/4)
      – Maximum input width of 1920 pixels
      – No limitation on the input height
      – Supported input formats are ARGB32-8888, YUV4:2:2-UYVY, YUV4:2:2-YUV2, YUV4:2:0-NV12, and YUV4:2:0-NV21
      – Alpha blending factor is rescaled like the R, G, and B color components.
    - Programmable color space conversion from YUV4:2:2 (YUV4:4:4, YUV4:2:0 after Chroma upsampling through the scaler) into ARGB32-8888. Images in YUV4:2:2 format with 90- or 270-degree rotation are preprocessed to YUV4:4:4 before the scaler, by duplicating the missing Chroma.
    - Programmable VC-1 range mapping
  – One write-back (WB) pipeline: Allows the use of the hardware processing available inside the DISPC, such as color space conversion, rescaling, and compositing to perform memory-to-memory transfer with data processing or capturing a displayed frame
    - Programmable color space conversion RGB24 into YUV4:4:4 or to YUV4:2:2-UYVY, YUV4:2:2-YUV2, YUV4:2:0-NV12, or YUV4:2:0-NV21 using programmable poly-phase filter
    - Programmable color space conversion RGB24 into YUV4:2:2-UYVY, YUV4:2:2-YUV2, YUV4:2:0-NV12, or YUV4:2:0-NV21
    - Selection of the color depth reduction from RGB24 to RGB16
    - Programmable poly-phase filter:
      – Independent horizontal and vertical resampling: Upsampling (up to x8) and downsampling (down to 1/4)
      – Maximum input width of 1920 pixels
      – No limitation on the input height
      – Supported input formats are ARGB32-8888, YUV4:2:2-UYVY, YUV4:2:2-YUV2, YUV4:2:0-NV12, and YUV4:2:0-NV21
      – Alpha blending factor is rescaled like the R, G, and B color components.
    - Selection of the source of the data:
      – Overlay output:
        - Primary LCD output
        - Secondary LCD output
        - Third LCD output
        - TV output
      – Pipelines:
        - Graphic
        - Video 1
        - Video 2
        - Video 3
- Three LCD outputs: primary (LCD1), secondary (LCD2), and tertiary (LCD3):
  – Input pixel format: ARGB32-8888
  – Output pixel format: RGB24-888 and YUV4:2:2 (YUV4:2:2 only available when BT mode output is enabled)
  – Overlay of graphic and video for one to four pipelines
  – Source and destination transparency color key
  – Global and pixel alpha blending (up to 8-bit blending factor)
  – Z-order programmable (full flexibility)
  – Displays supported:

    - Active matrix color: 12-, 16-, 18-, and 24-bit panel interface support (replicated or dithered encoded pixel values)
  - Independent programmable timing generators for LCD1, LCD2, and LCD3 to support:

Using DPI1 interface	SXGA VESA timing @ 60 fps, 1080i/720p @ 60 fps CEA-861-D, UXGA @ 60 fps
Using DPI2 interface	SXGA VESA timing @ 60 fps, 1080i/720p @ 60 fps CEA-861-D, UXGA @ 60 fps
Using DPI3 interface	SXGA VESA timing @ 60 fps, 1080i/720p @ 60 fps CEA-861-D, UXGA @ 60 fps

  - Configurable LCD output mode: progressive or interlace mode
  - Multiple-cycle output format on 8-, 9-, 12-, and 16-bit interface time division multiplexing (TDM)
- One TV output:
  - Input pixel format: ARGB40-10.10.10.10
  - Output pixel format: ARGB40-10.10.10.10
  - Overlay of graphic and video for one to four pipelines
  - Source and destination transparency color key
  - Global and pixel alpha blending (up to 10-bit blending factor)
  - Z-order programmable (full flexibility)
  - Slave mode support (no master mode support) with synchronization signals provided by HDMI TX:
    - HSYNC (horizontal synchronization signal)
    - VSYNC (vertical synchronization signal)
    - RE (data request signal)
    - FID (field ID: even and odd field information)
  - RGB30-10.10.10 data bus output for connection to HDMI TX and extended to 36 by duplication of the MSBs
  - HD-1080p, HD-1080i, HD-720p, SD-480p, SD-576p, SD-576i, and SD-480i using HDMI
  - HDMI deep color mode support, 30-bit data output to HDMI encoder
  - Pixel duplication capability (from one pixel clock cycle up to eight cycles)
- Panel support with MIPI DPI protocol:
  - 12-, 16-, 18-, and 24-bit active matrix panel interface support (replicated or dithered encoded pixel values)
- Common:
  - Rotation 0, 90, 180, and 270 degrees using DMM-TILER
  - Synchronized buffer update
  - Hardware cursor (using the graphics pipeline or one of the video pipelines)
  - Independent gamma curve support on LCDs outputs and TV output
  - Multiple-buffer support
  - Mirroring/flip-flop support (using DMM-TILER)
  - Programmable color phase rotation (CPR)
  - Alpha blending support:
    - Embedded pixel factor (ARGB and RGBA)
    - Global alpha
- DMA (internal to the DISPC):
  - Support for accessing tiled structure through the TILER inside the dynamic memory management (DMM)
  - Support for accessing nontiled structure through the TILER or directly
  - Support for rotation, flip-flop, and mirroring through the TILER inside the DMM
  - Support for memory fragmentation through the TILER inside the DMM
  - Integrated shared buffers between DMA engine and pipelines
  - Programmable buffer thresholds
  - Bandwidth limiter on write request (insertion on idle cycles between requests)
- Advanced:
  - Mode outputting data on display only from the DMA buffer (self-refresh using the DMA FIFO)
  - DMA buffer hand-check in stall mode

  - Arbitration between high and low priority (GFX, VID1, VID2, VID3, and WB pipelines)
- Power modes:
  - Low-power saving modes
  - Support on-the-fly dynamic voltage and frequency scaling (DVFS)
  - Merge capability of the DMA buffers to support greater OFF period on the L3_MAIN interconnect
    - All buffers associated to a single pipeline
    - Reallocation of the buffers of the nonactive pipelines to the active pipelines

11.2.2 DISPC Environment

The DISPC provides the required control signals to interface directly to an external parallel panel for the MIPI DPI protocol.

Figure 11-23 shows the LCD support parallel interface.

Note

Parallel interface is available through the LCD1, LCD2, LCD3, and TV outputs of the DISPC.

The LCD data and control signals are multiplexed with the TV output data, and control signals are provided by the HDMI module.

The selection can be done at the top level of the display subsystem. For further details and signal mapping, see Section 11.1.1, *Display Subsystem Environment*.

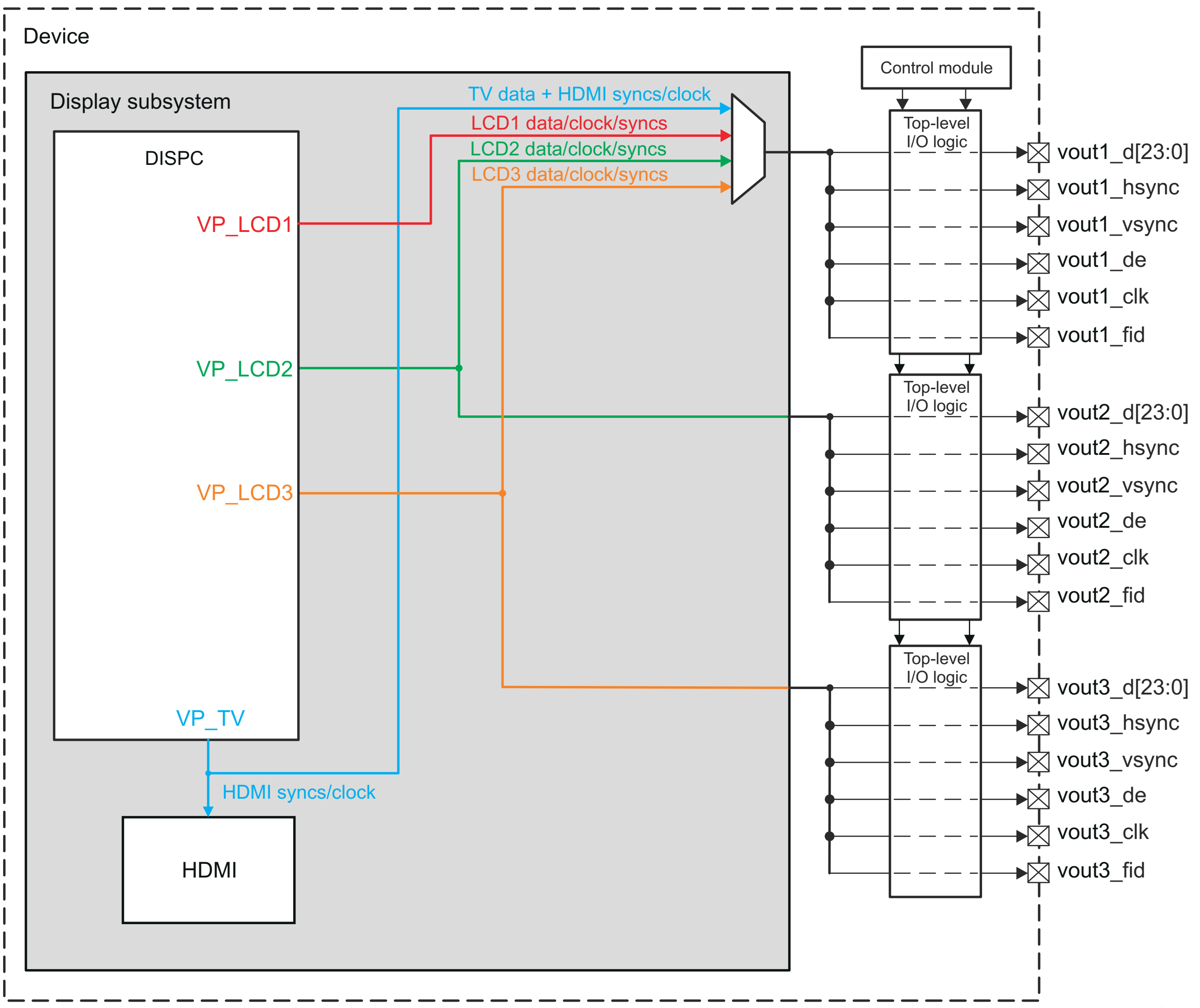


Figure 11-23. DISPC LCD Support Parallel Interface

Note

The path from a module pin to device pad (or pads) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the control module registers and dedicated IP registers. For more information, see *Pad Configuration Registers*, in *Control Module*.

Table 11-54 describes the interface signals to/from the LCD panel in bypass mode.

Table 11-54. DISPC Parallel Interface Signals

Signal Name(2)	Type(1)	Description
voutX_d[23:0]	O	Pixel data
voutX_clk	O	Pixel clock
voutX_vsync	O	Vertical synchronization. The LCD frame clock (vsync) toggles after all the lines in a frame are transmitted to the LCD panel and a programmable number of line clock cycles has elapsed at the beginning and end of each frame.
voutX_hsync	O	Horizontal synchronization. The LCD line clock (hsync) toggles after all pixels in a line are transmitted to the LCD panel and a programmable number of pixel clock wait-states has elapsed at the beginning and end of each line.
voutX_de	O	In active matrix technology, the DE signal acts as an output-enable signal to indicate when data must be latched using the pixel clock.
voutX_fid	O	The FID signal indicates the field identifier for the LCD output field: • 0 means even. • 1 means odd.

(1) I = Input, O = Output, I/O = Input/Output
(2) X = 1, 2, or 3, indicates the corresponding parallel video output (VOUT1, VOUT2, or VOUT3)

11.2.2.1 DISPC LCD Output and Data Format for the Parallel Interface

This section describes the pixel data bus and shows timing diagrams of transactions and synchronizations.

Figure 11-24 through Figure 11-27 show the pixel data bus, depending on the use of 12-, 16-, 18-, or 24-pixel data output pins.

In the Active matrix display type, one pixel per pixel clock is displayed.

Active matrix displays bypass the STN dithering logic block and the output FIFO. Each line represents one pixel.

Figure 11-24 through Figure 11-27 show 12-, 16-, 18-, and 24-bit active matrix displays, respectively.

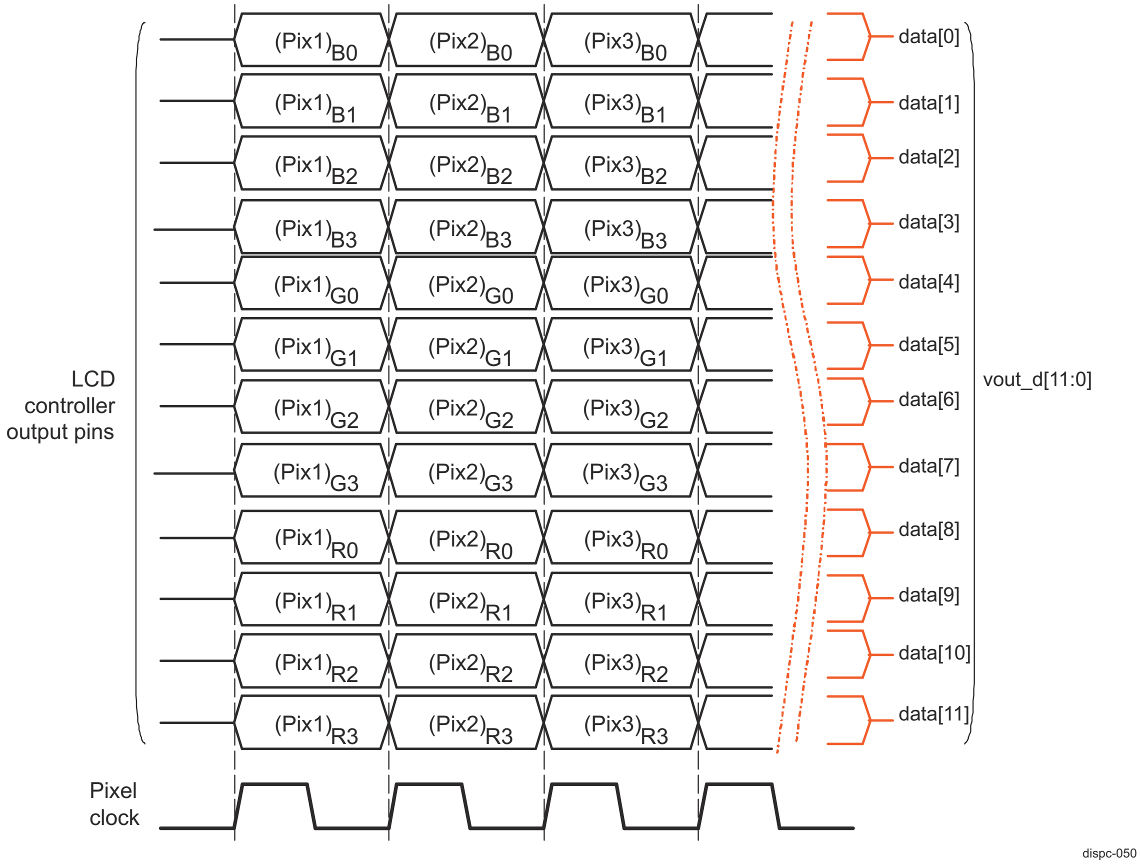


Figure 11-24. DISPC LCD Pixel Data Color12 Active Matrix

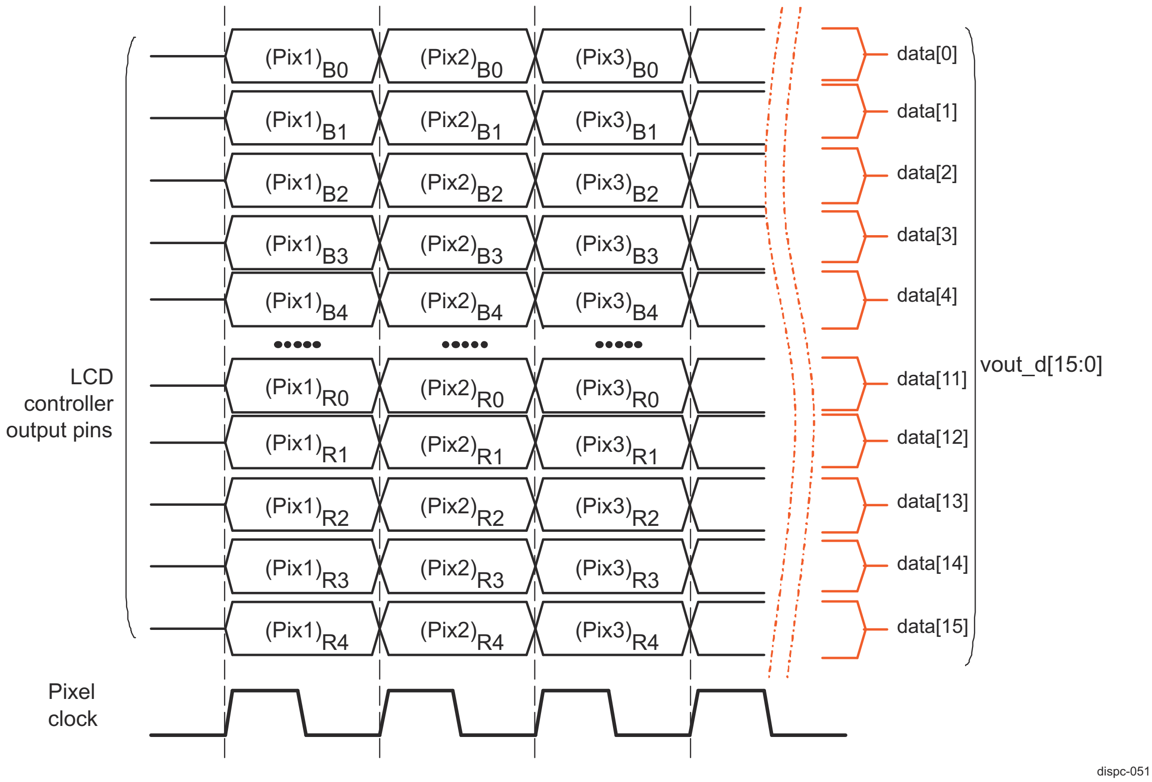


Figure 11-25. DISPC LCD Pixel Data Color16 Active Matrix

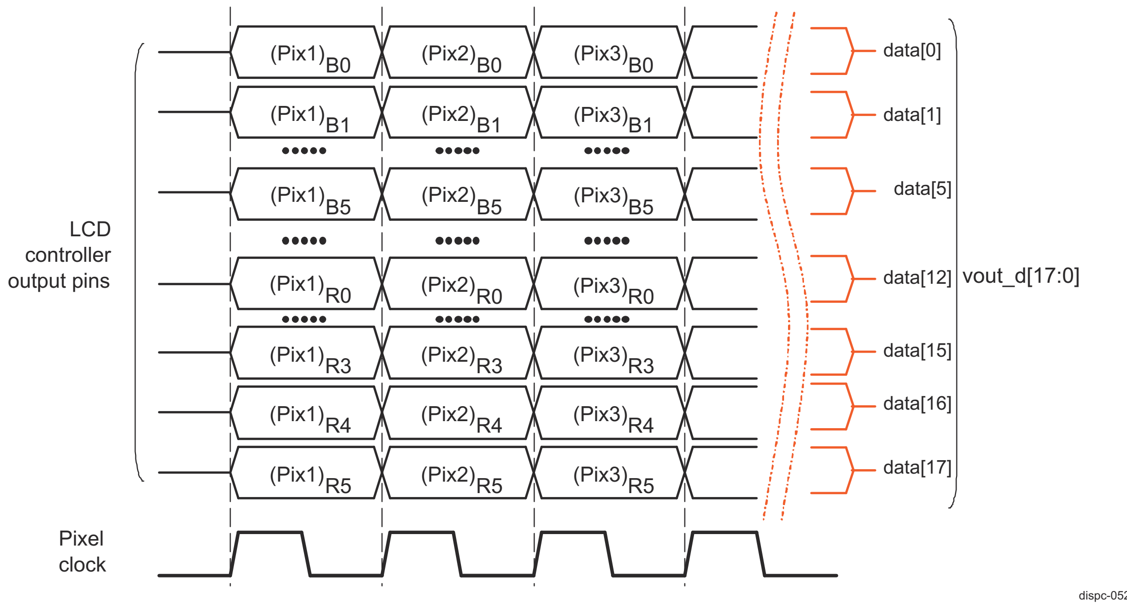


Figure 11-26. DISPC LCD Pixel Data Color18 Active Matrix

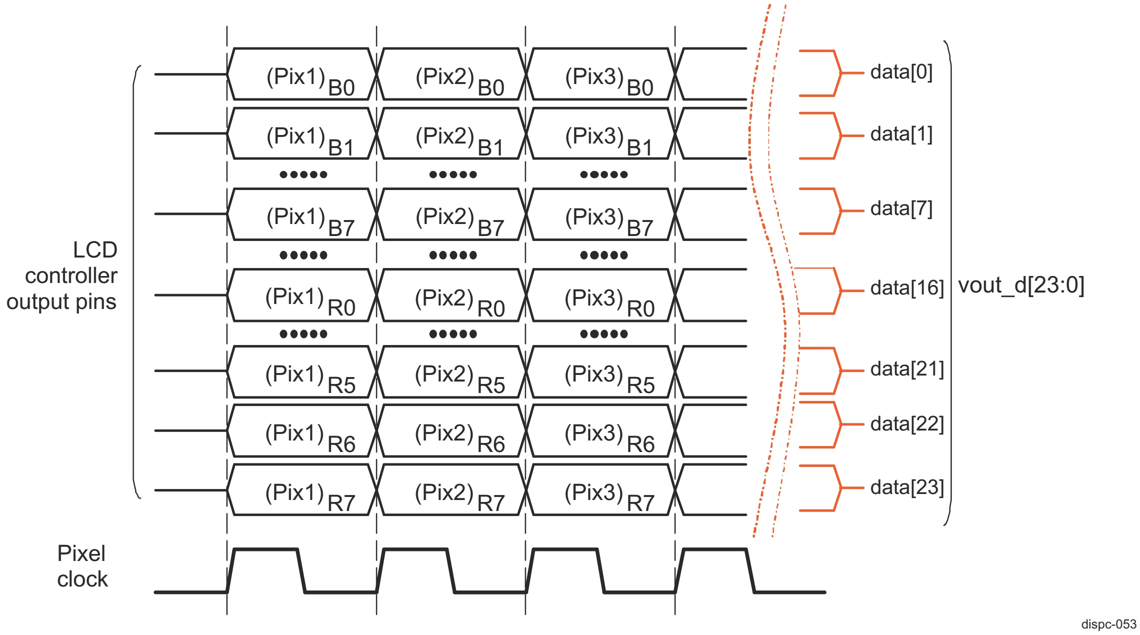


Figure 11-27. DISPC LCD Pixel Data Color24 Active Matrix

Table 11-55 summarizes the mapping of RGB color components to DSS output data signals, with corresponding settings of DISPC_CONTROLo[9:8] TFTDATALINES register bit-field.

Table 11-55. DSS Output Data Signals to RGB Color Components Mapping

Color Assignment	12-bit Output Mode	16-bit Output Mode	18-bit Output Mode	24-bit Output Mode
	TFTDATALINES=00b	TFTDATALINES=01b	TFTDATALINES=10b	TFTDATALINES=11b
Red 7 (MS bit)	-	-	-	voutX_d23[1]
Red 6	-	-	-	voutX_d22
Red 5	-	-	voutX_d17	voutX_d21
Red 4	-	voutX_d15	voutX_d16	voutX_d20
Red 3	voutX_d11	voutX_d14	voutX_d15	voutX_d19
Red 2	voutX_d10	voutX_d13	voutX_d14	voutX_d18
Red 1	voutX_d9	voutX_d12	voutX_d13	voutX_d17
Red 0	voutX_d8	voutX_d11	voutX_d12	voutX_d16
Green 7	-	-	-	voutX_d15
Green 6	-	-	-	voutX_d14
Green 5	-	voutX_d10	voutX_d11	voutX_d13
Green 4	-	voutX_d9	voutX_d10	voutX_d12
Green 3	voutX_d7	voutX_d8	voutX_d9	voutX_d11
Green 2	voutX_d6	voutX_d7	voutX_d8	voutX_d10
Green 1	voutX_d5	voutX_d6	voutX_d7	voutX_d9
Green 0	voutX_d4	voutX_d5	voutX_d6	voutX_d8
Blue 7	-	-	-	voutX_d7
Blue 6	-	-	-	voutX_d6
Blue 5	-	-	voutX_d5	voutX_d5
Blue 4	-	voutX_d4	voutX_d4	voutX_d4
Blue 3	voutX_d3	voutX_d3	voutX_d3	voutX_d3
Blue 2	voutX_d2	voutX_d2	voutX_d2	voutX_d2
Blue 1	voutX_d1	voutX_d1	voutX_d1	voutX_d1

Table 11-55. DSS Output Data Signals to RGB Color Components Mapping (continued)

Color Assignment	12-bit Output Mode	16-bit Output Mode	18-bit Output Mode	24-bit Output Mode
Blue 0 (LS bit)	voutX_d0	voutX_d0	voutX_d0	voutX_d0

(1) X = 1, 2, or 3, indicates the corresponding VOUT output (VOUT1, VOUT2, or VOUT3)

11.2.2.2 DISPC Transaction Timing Diagrams

Figure 11-28 through Figure 11-31 show timing diagrams of synchronization signals and pixel clocks for active matrix panels. The DISPC directly drives these signals, which are related to the programmable fields listed in Table 11-56.

Table 11-56. DISPC Programmable Fields in Bypass Mode

Name	Register	Description
PPL	DISPC_SIZE_LCDo[11:0] PPL value + 1	Pixels per line
LPP	DISPC_SIZE_LCDo[27:16] LPP value + 1	Lines per panel
HBP	DISPC_TIMING_Ho[31:20] HBP value + 1	Horizontal back porch
HFP	DISPC_TIMING_Ho[19:8] HFP value + 1	Horizontal front porch
HSW	DISPC_TIMING_Ho[7:0] HSW value + 1	Horizontal synchronization pulse width
VBP	DISPC_TIMING_Vo[31:20] VBP value	Vertical back porch
VFP	DISPC_TIMING_Vo[19:8] VFP value	Vertical front porch
VSW	DISPC_TIMING_Vo[7:0] VSW value + 1	Vertical synchronization pulse width
ONOFF(1)	DISPC_POL_FREQo[17] ONOFF	DISPC_HSYNC and DISPC_VSYNC pixel clock control
RF(2)	DISPC_POL_FREQo[16] RF	DISPC_HSYNC and DISPC_VSYNC pixel clock edge control
IEO	DISPC_POL_FREQo[15] IEO	Invert DISPC_ACBIAS
IPC(3)	DISPC_POL_FREQo[14] IPC	Invert DISPC_PCLK
IHS	DISPC_POL_FREQo[13] IHS	Invert DISPC_HSYNC
IVS	DISPC_POL_FREQo[12] IVS	Invert DISPC_VSYNC

(1) DISPC_POL_FREQo[17] ONOFF and Control Module register CTRL_CORE_SMA_SW_1[] DSS_CHx_ON_OFF must match (x = 0,1,2).
(2) DISPC_POL_FREQo[16] RF and Control Module register CTRL_CORE_SMA_SW_1[] DSS_CHx_RF must match (x = 0,1,2).
(3) DISPC_POL_FREQo[14] IPC and Control Module register CTRL_CORE_SMA_SW_1[] DSS_CHx_IPC must match (x = 0,1,2).

- Active matrix timing configuration 1:
  - DISPC_POL_FREQo[17] ONOFF = 0 and CTRL_CORE_SMA_SW_1[] DSS_CHx_ON_OFF = 0
  - DISPC_POL_FREQo[16] RF = 0 and CTRL_CORE_SMA_SW_1[] DSS_CHx_RF = 0

    The HSYNC and VSYNC signals are driven on the opposite edge of PCLK from the pixel data.
  - DISPC_POL_FREQo[15] IEO = 0

    The DE signal is active high.
  - DISPC_POL_FREQo[14] IPC = 0 and CTRL_CORE_SMA_SW_1[] DSS_CHx_IPC = 0

    The pixel data are driven on the rising edge of PCLK.
  - DISPC_POL_FREQo[13] IHS = 0

    The HSYNC signal is active high.
  - DISPC_POL_FREQo[12] IVS = 0

    The VSYNC signal is active high.

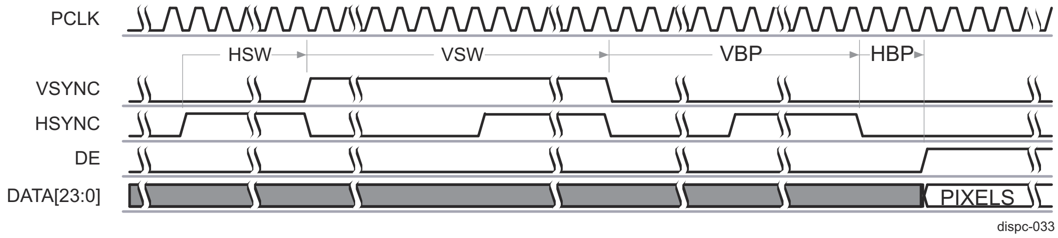


Figure 11-28. DISPC Active Matrix Timing Diagram of Configuration 1 (Start of Frame)

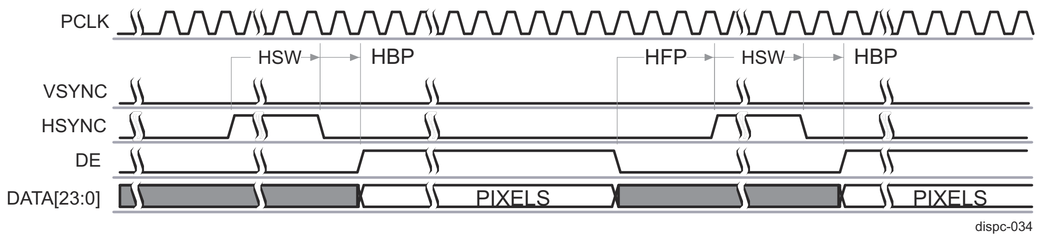


Figure 11-29. DISPC Active Matrix Timing Diagram of Configuration 1 (Between Lines)

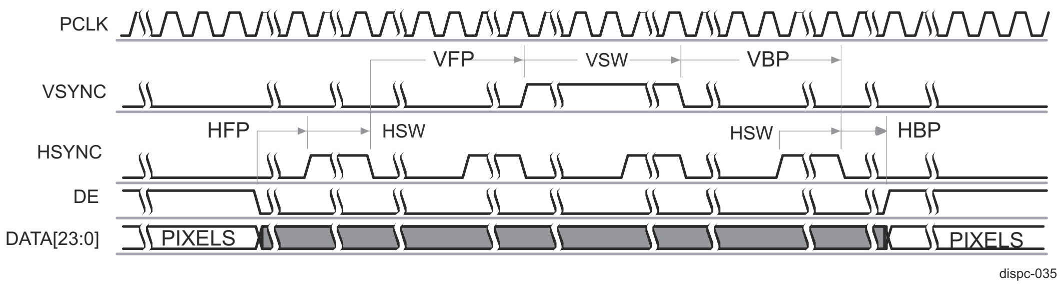


Figure 11-30. DISPC Active Matrix Timing Diagram of Configuration 1 (Between Frames)

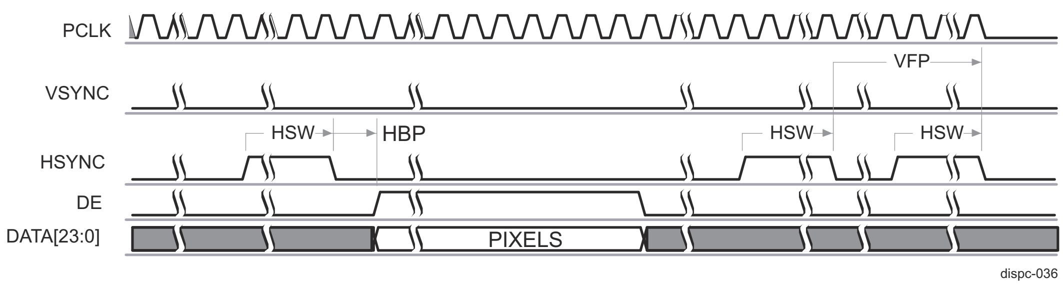


Figure 11-31. DISPC Active Matrix Timing Diagram of Configuration 1 (End of Frame)

- Active matrix timing configuration 2:
  - DISPC_POL_FREQo[17] ONOFF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CHx_ON_OFF = 1
  - DISPC_POL_FREQo[16] RF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CHx_RF = 1

    The HSYNC and VSYNC signals are driven on the rising edge of PCLK.
  - DISPC_POL_FREQo[15] IEO = 1

    The DE signal is active low.
  - DISPC_POL_FREQo[14] IPC = 1 and CTRL_CORE_SMA_SW_1[]DSS_CHx_IPC = 1

    The pixel data is driven on the falling edge of PCLK.
  - DISPC_POL_FREQo[13] IHS = 1

    The HSYNC signal is active low.
  - DISPC_POL_FREQo[12] IVS = 1

The VSYNC signal is active low.

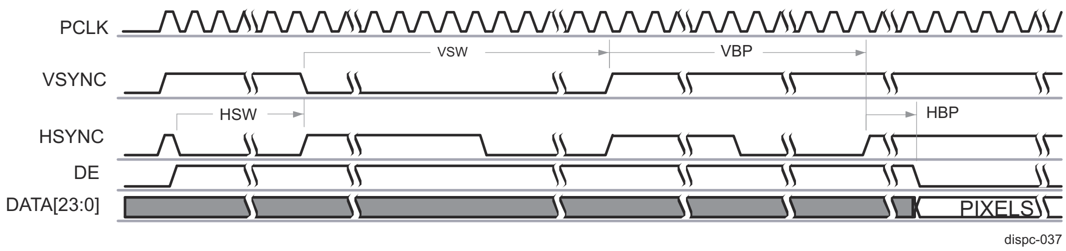


Figure 11-32. DISPC Active Matrix Timing Diagram of Configuration 2 (Start of Frame)

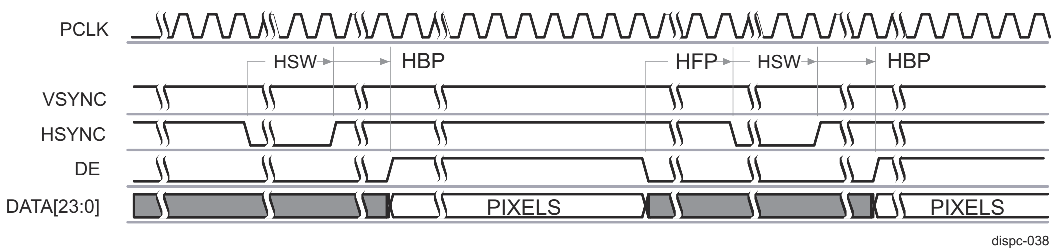


Figure 11-33. DISPC Active Matrix Timing Diagram of Configuration 2 (Between Lines)

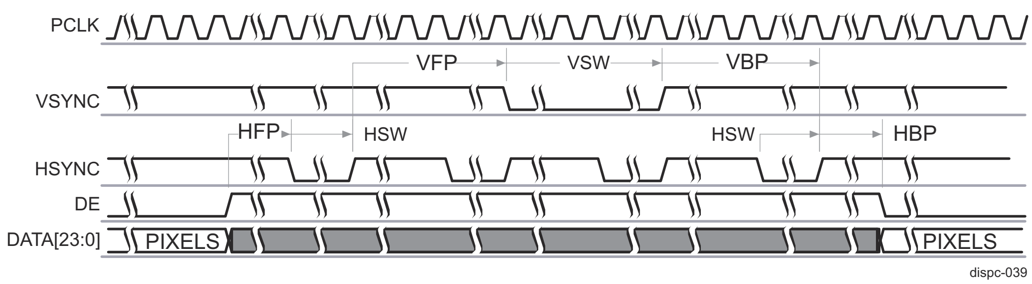


Figure 11-34. DISPC Active Matrix Timing Diagram of Configuration 2 (Between Frames)

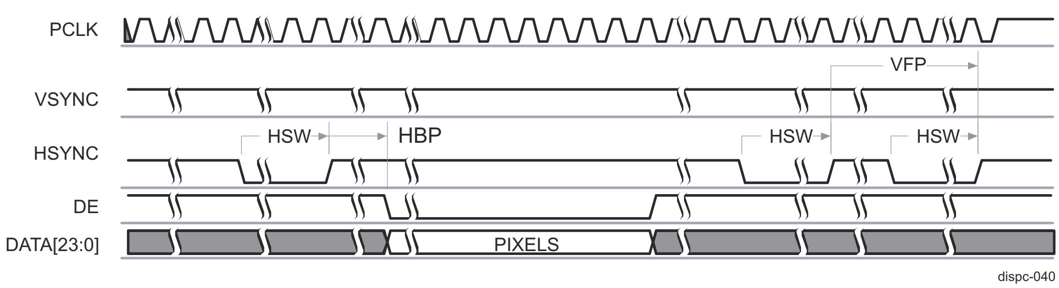


Figure 11-35. DISPC Active Matrix Timing Diagram of Configuration 2 (End of Frame)

- Active matrix timing configuration 3:
  - DISPC_POL_FREQo[17] ONOFF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CHx_ON_OFF = 1
  - DISPC_POL_FREQo[16] RF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CHx_RF = 1

    The HSYNC and VSYNC signals are driven on the rising edge of PCLK.
  - DISPC_POL_FREQo[15] IEO = 0

    The DE signal is active high.
  - DISPC_POL_FREQo[14] IPC = 0 and CTRL_CORE_SMA_SW_1[]DSS_CHx_IPC = 0

    The pixel data are driven on the rising edge of PCLK.
  - DISPC_POL_FREQo[13] IHS = 0

The HSYNC signal is active high.

– DISPC_POL_FREQo[12] IVS = 0

The VSYNC signal is active high.

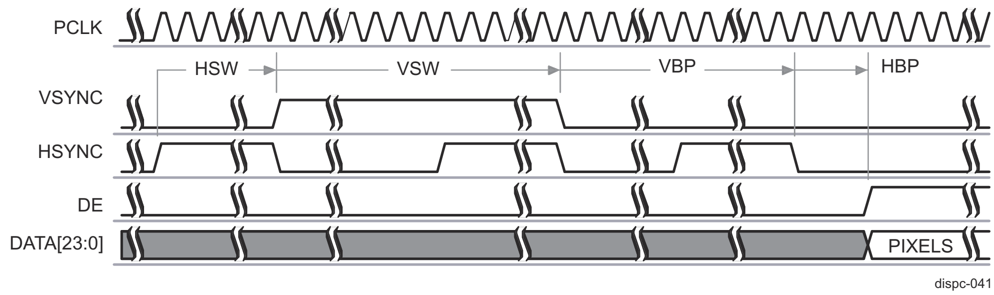


Figure 11-36. DISPC Active Matrix Timing Diagram of Configuration 3 (Start of Frame)

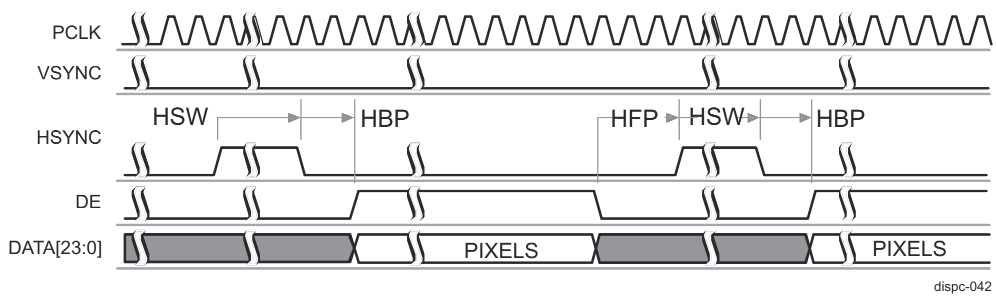


Figure 11-37. DISPC Active Matrix Timing Diagram of Configuration 3 (Between Lines)

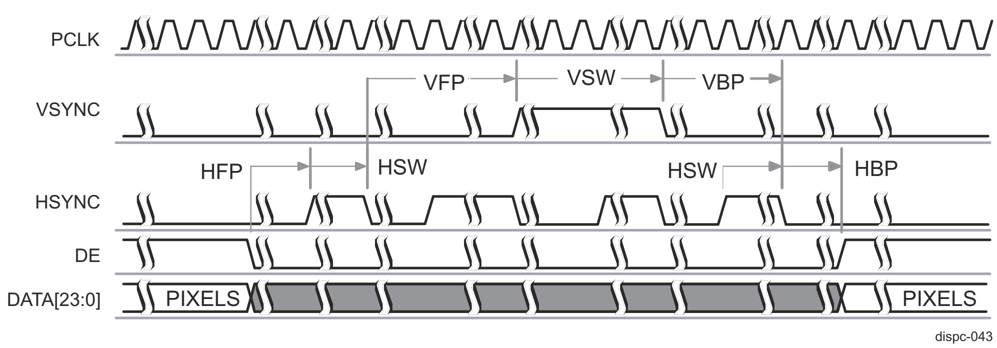


Figure 11-38. DISPC Active Matrix Timing Diagram of Configuration 3 (Between Frames)

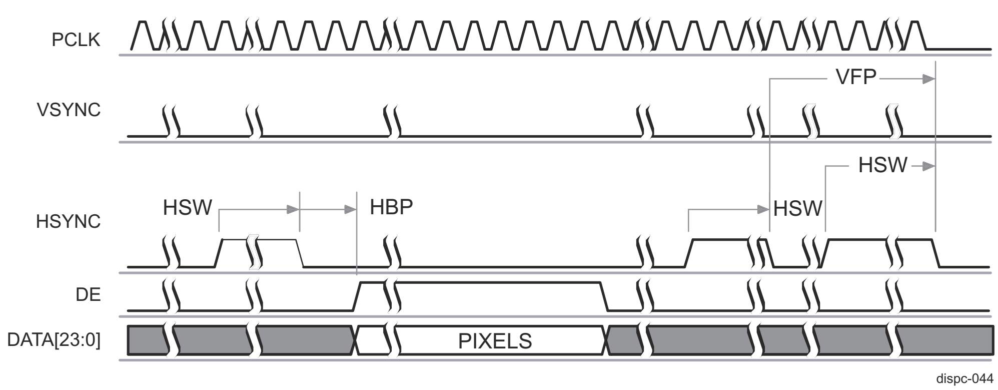


Figure 11-39. DISPC Active Matrix Timing Diagram of Configuration 3 (End of Frame)

11.2.2.3 DISPC TV Output and Data Format for the Parallel Interface

This section describes the TV output pixel data bus for the parallel interface.

The TV pixel data interface is a 30-bit RGB interface. Only the MSB part of each color component is connected to the display subsystem boundary: R[9:2], G[9:2], B[9:2]. The output of the data is synchronized to the data request signal (HDMI_M_DE) from the HDMI encoder.

Figure 11-40 shows the format of the TV output pixel data.

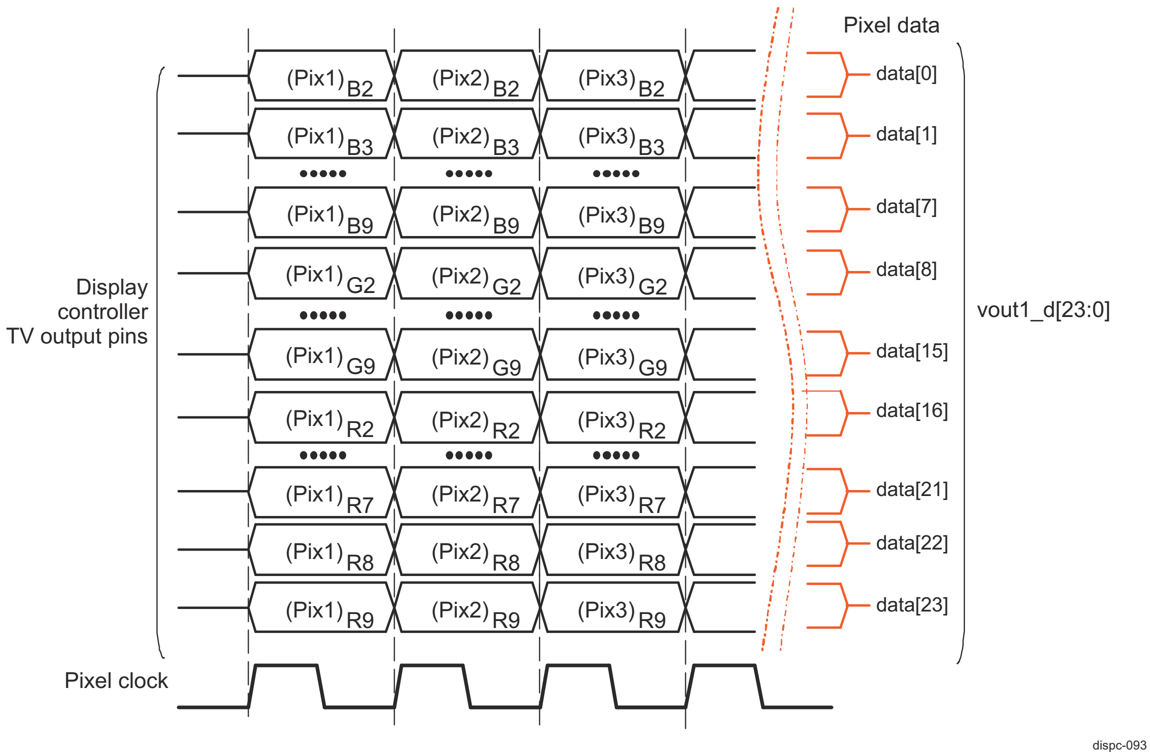


Figure 11-40. DISPC TV Output Pixel Data

11.2.3 DISPC Integration

This section describes the DISPC integration in the device (see Figure 11-41). For complete details about clocks and resets, see Section 11.1, *Display Subsystem Overview*.

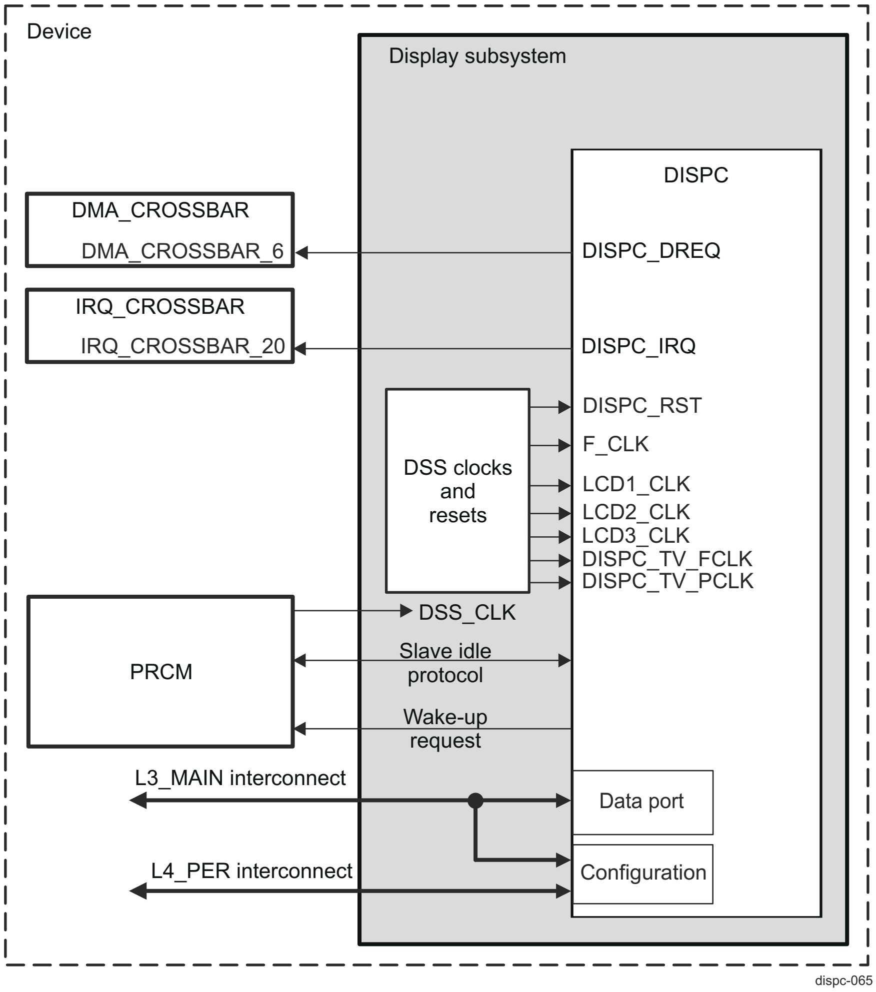


Figure 11-41. DISPC Integration

Table 11-57 and Table 11-58 list the integration attributes and clock and resets, respectively.

Table 11-57. DISPC Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
DISPC	PD_DSS	L3_MAIN for data transfer and configuration

Table 11-58. DISPC Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DISPC	F_CLK	DSS_CLK DPLL_DSI1_A_CLK1 DPLL_DSI1_B_CLK1 DPLL_DSI1_C_CLK1 DPLL_HDMI_CLK1	PRCM DPLL_VIDEO1 DPLL_VIDEO2 DPLL_HDMI	Functional clock for the DISPC logic. For the multiplexing description, see Section 11.1.2.1, *Display Subsystem Clocks*.
	LCD1_CLK	DSS_CLK DPLL_DSI1_A_CLK1	DPLL_VIDEO1 DPLL_HDMI	Clock used to generate the divided pixel clock for the primary LCD interface. For the multiplexing description, see Section 11.1.2.1, *Display Subsystem Clocks*.
	LCD2_CLK	DSS_CLK DPLL_DSI1_B_CLK1	PRCM DPLL_VIDEO1 DPLL_VIDEO2 DPLL_HDMI	Clock used to generate the divided pixel clock for the secondary LCD interface. For the multiplexing description, see Section 11.1.2.1, *Display Subsystem Clocks*.
	LCD3_CLK	DSS_CLK DPLL_DSI1_C_CLK1	DPLL_VIDEO1 DPLL_VIDEO2 DPLL_HDMI	Clock used to generate the divided pixel clock for the third LCD interface. For the multiplexing description, see Section 11.1.2.1, *Display Subsystem Clocks*.
	DISPC_TV_FCLK	F_CLK DPLL_HDMI_CLK1	DSS DPLL_HDMI	TV functional clock (this is either the F_CLK or a clock driven from the DPLL_HDMI). For the multiplexing description, see Section 11.1.2.1, *Display Subsystem Clocks*.
	DISPC_TV_CLK	DSS_HDMI_PCLK	HDMI	Pixel clock provided by the HDMI module. For the multiplexing description, see Section 11.1.2.1, *Display Subsystem Clocks*.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DISPC	DISPC_RST	DSS_RST	PRCM	Hardware reset is coming from the PRCM module or through a software reset performed at the DSS level. For the tree reset description, see Section 11.1.2, *Display Subsystem Integration*. **NOTE:** The DSS_RST signal is provided to the entire display subsystem. When inside the display subsystem boundaries, it is named DISPSS_RST, which on its end is provided to the DISPC and is named DISPC_RST.

Table 11-59. DISPC Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DISPC	DISPC_IRQ	IRQ_CROSSBAR_20	MPU_IRQ_25 DSP1_IRQ_58 DSP2_IRQ_58 IPU1_IRQ_26 IPU2_IRQ_26	DISPC interrupt requests.
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**

Table 11-59. DISPC Hardware Requests (continued)

DISPC	DISPC_DREQ	DMA_CROSSBAR_6	DMA_EDMA_DREQ_5	The line trigger signal to synchronize a memory-to-memory logical channel in the device DMA is generated by the DISPC IP.
			DMA_SYSTEM_DREQ_5	

Note

The “**Default Mapping**” column in Table 11-59 *DISPC Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.

For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

For more information about the device DMA_SYSTEM module, see *System DMA*.

For more information about the device EDMA module, see *Enhanced DMA*.

11.2.4 DISPC Functional Description

The DISPC can read and display the encoded pixel data stored in memory (see Figure 11-42).

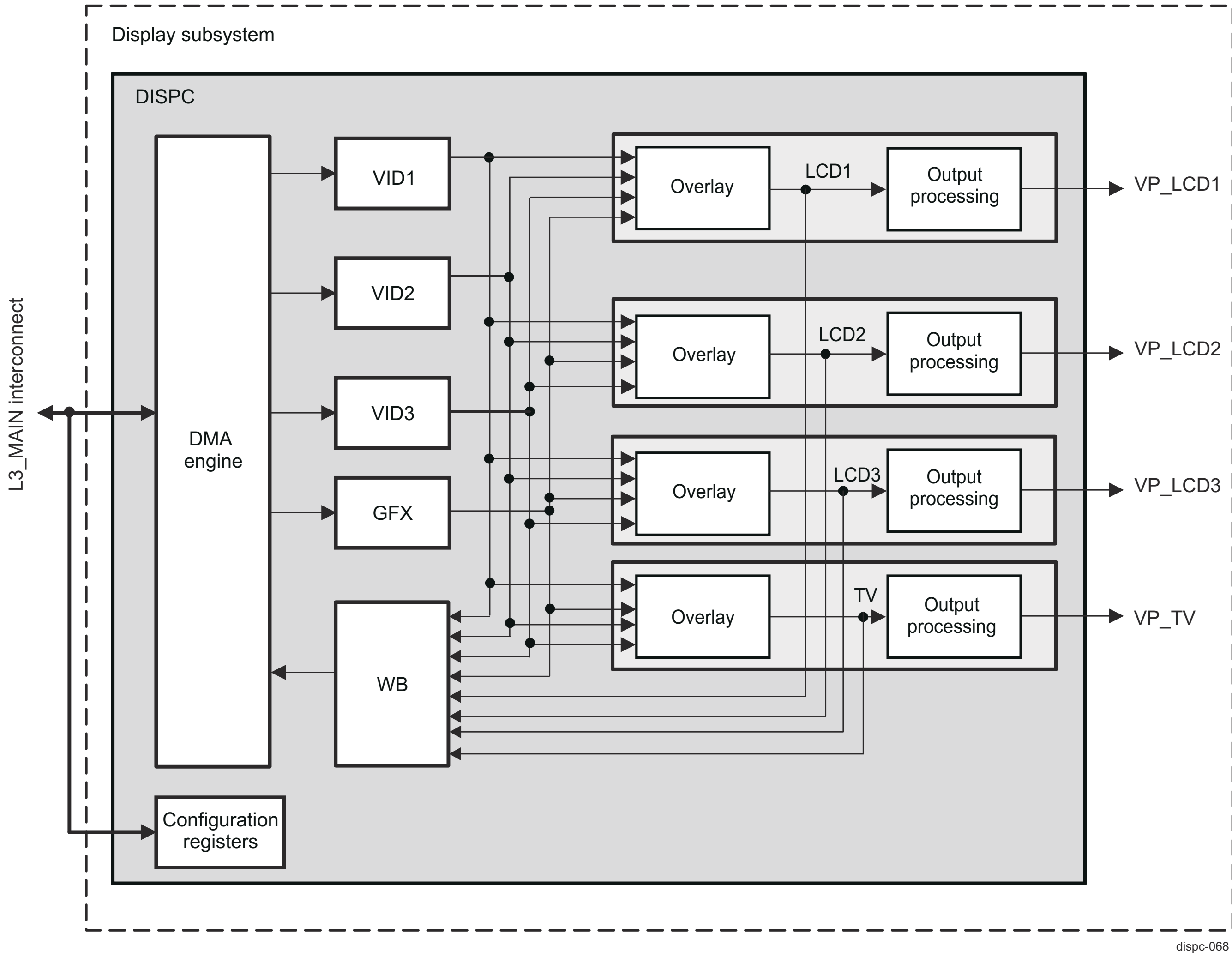


Figure 11-42. DISPC Architecture Overview

The DISPC can read and display the encoded pixel data stored in memory and write the output of one of the overlays or one of the pipelines into system memory.

Several processes can be configured to manage the graphics pipeline (replication, antiflicker) and video pipelines (VC-1, color space conversion, scaling, overlay, transparency, and so forth).

The data coming out of a pipeline is sent to one of the four outputs, depending on the user configuration. An overlay manager manages inputs of multiple pipelines. User timing configurations for LCD and TV are available.

The DISPC allows the capturing of one output of the pipeline or overlay manager to redirect it into the WB pipeline. It allows the use of the hardware processing available inside the DISPC, such as color space conversion, rescaling, and compositing, and so forth to perform memory-to-memory transfer with data processing.

11.2.4.1 DISPC Clock Configuration

The PCLK frequency for each LCD output is derived from a dedicated input clock: LCD1_CLK, LCD2_CLK, or LCD3_CLK for the three LCD outputs, respectively. Each input clock is divided by the values of the DISPC_DIVISORo[23:16] LCD bit field and then the DISPC_DIVISORo[7:0] PCD bit field independently for each LCD pixel clock (see Figure 11-43). DSS_DISPC_LCD1_PCLK, DSS_DISPC_LCD2_PCLK, and DSS_DISPC_LCD3_PCLK are independent:

- DSS_DISPC_LCD1_PCLK = (LCD1_CLK/LCD1) / PCD1

- DSS_DISPC_LCD2_PCLK = (LCD2_CLK/LCD2) / PCD2
- DSS_DISPC_LCD3_PCLK = (LCD3_CLK/LCD3) / PCD3

The functional clock of the DISPC is derived from F_CLK by an independent divisor. The dividing value is set in the DISPC_DIVISORo[23:16] LCD bit field.

For backward compatibility, the divisor value LCD can be set to the value of LCD1. To enable this functionality, the DISPC_DIVISOR[0] ENABLE bit must be set to 1.

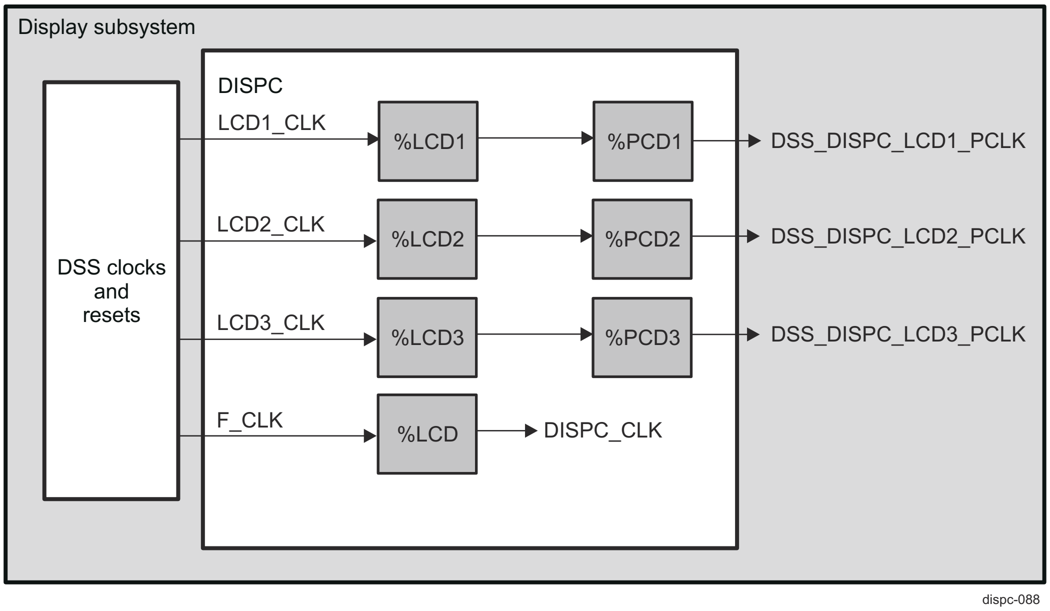


Figure 11-43. DISPC Clock Tree Overview

11.2.4.2 DISPC Software Reset

To perform a software reset on the DISPC, set the DISPC_SYSCONFIG[1] SOFTRESET bit to 0x1. The DISPC_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 0x1. When the software reset completes, the DISPC_SYSCONFIG[1] SOFTRESET bit is automatically reset. Software must ensure that the software reset completes before performing DISPC operations.

11.2.4.3 DISPC Power Management

The DISPC supports the MStandby/Wait, IdleReq/SidleAck, and wake-up protocols as defined in *Power, Reset, and Clock Management*.

11.2.4.3.1 DISPC Idle Mode

The DISPC supports no-idle mode, force-idle mode, and smart-idle mode. The mode can be selected by programming the appropriate value in the DISPC_SYSCONFIG[4:3] SIDLEMODE bit field.

Condition of assertion of the SIdleAck signal:

- In no-idle mode: SIdleAck is never asserted.
- In force-idle mode: SIdleAck is asserted unconditionally with a 1-configuration port interface clock cycle delay with respect to an IdleReq assertion.

Note

The proper use of force-idle mode assumes that no interrupt needs to be generated.

- In smart-idle mode: SIdleAck is asserted when at least the following conditions are satisfied:
  - No interrupt is pending.
  - The DISPC no longer uses the interface clock for the slave port.

Once SIdleAck is asserted, the DISPC interface lock used by the slave port can be shut down at any time.

The conditions of deassertion of the SIdleAck signal are:

- In force-idle mode: SIdleAck is deasserted with a 1-configuration port interface clock cycle delay with respect to an IdleReq deassertion.
- In smart-idle mode: SIdleAck is deasserted with a 1-configuration port interface clock cycle delay with respect to an IdleReq deassertion.

Once SIdleAck is released, the DISPC is fully operational and a DMA request can be processed normally.

11.2.4.3.2 DISPC StandBy Mode

The DISPC supports no-standby mode, force-standby mode, and a single smart-standby mode. The mode is set in the DISPC_SYSCONFIG[13:12] MIDLEMODE bit field. The functional clock is always active if the module is enabled. The L3_MAIN clock can be shut down at any time independently of the status of MStandby.

The conditions of assertion of the MStandby signal are:

- In no-standby mode: MStandby is never asserted.
- In force-standby mode: MStandby is asserted when the module is disabled.
- In smart-standby: In the case of one of the following conditions:
  - The GFX pipeline is disabled or enabled and the data fetch is complete for the GFX window, or the GFX pipeline is enabled but the data fetch did not complete and data in the DMA buffer is greater than the high threshold value.
  - The VID1 pipeline is disabled or enabled and the data fetch is complete for the VID1 window, or the VID1 pipeline is enabled but the data fetch did not complete and data in the DMA buffer is greater than the high threshold value.
  - The VID2 pipeline is disabled or enabled and the data fetch is complete for the VID2 window, or the VID2 pipeline is enabled but the data fetch did not complete and data in the DMA buffer is greater than the high threshold value.
  - The VID3 pipeline is disabled or enabled and the data fetch is complete for the VID3 window, or the VID3 pipeline is enabled but the data fetch did not complete and data in the DMA buffer is greater than the high threshold value.
  - The WB pipeline is disabled or enabled and the data store to memory is complete for the WB picture, or the WB pipeline is enabled but the data storage did not complete and data in the DMA buffer is lower than the low threshold value.

The MStandby signal asserts whenever all five events have occurred or the DISPC is disabled. While MStandby is asserted, the DISPC does not generate any transaction on the L3_MAIN master port.

The conditions of deassertion of the MStandby signal are:

- In force-standby mode: MStandby is deasserted only when the DISPC is enabled.
- In smart-standby mode: In the case of one of the following conditions:
  - The GFX pipeline is enabled but the data fetch did not complete for the GFX window, and the data in the DMA buffer is less than the low threshold value.
  - The VID1 pipeline is enabled but the data fetch did not complete for the VID1 window, and the data in the DMA buffer is less than the low threshold value.
  - The VID2 pipeline is enabled but the data fetch did not complete for the VID2 window, and the data in the DMA buffer is less than the low threshold value.
  - The VID3 pipeline is enabled but the data fetch did not complete for the VID3 window, and the data in the DMA buffer is less than the low threshold value.
  - The WB pipeline is enabled but the data store did not complete for the WB picture, and the data in the DMA buffer more than the high threshold value.

Detection of the deassertion conditions assumes that the interface clocks are active.

11.2.4.3.3 DISPC Wakeup

The DISPC supports wake-up signaling. The mode can be selected by programming the appropriate value in the DISPC_SYSCONFIG[2] ENAWAKEUP bit. Because the SWakeUp signal is asynchronous, it does not require the interface clock.

The conditions of assertion of the SWakeUp signal are:

- The GFX pipeline is enabled but the data fetch did not complete for the GFX window, and the data in the DMA buffer is less than the low threshold value.
- The VID1 pipeline is enabled but the data fetch did not complete for the VID1 window, and the data in the DMA buffer is less than the low threshold value.
- The VID2 pipeline is enabled but the data fetch did not complete for the VID2 window, and the data in the DMA buffer is less than the low threshold value.
- The VID3 pipeline is enabled but the data fetch did not complete for the VID3 window, and the data in the DMA buffer is less than the low threshold value.
- The WB pipeline is enabled and the data in the DMA buffer is more than the high threshold value.
- The last pixel displayed into the LCD1 panel if it is not the last frame
- The last pixel displayed into the LCD2 panel if it is not the last frame
- The last pixel displayed into the LCD3 panel if it is not the last frame
- The last pixel displayed into the TV panel if it is not the last frame

The SWakeUp signal is asserted whenever any one of these nine events occurs and IdleAck is asserted.

When one of the active pipelines reaches the low threshold and must refill its DMA buffer for the current frame, all other pipelines refill their own DMA buffer even if their low threshold has not been reached. The DISPC_CONFIG1[17] BUFFERFILLING bit is used to increase the probability that the time increases, where there is no access to the L3_MAIN interconnect.

The condition of deassertion of the SWakeUp signal is:

- Immediately after deassertion of IdleReq

11.2.4.4 DISPC Interrupt Requests

The interrupt line, DISPC_DREQ, indicates when one or more events are detected by the hardware. Each event is independently maskable by setting the DISPC_IRQENABLE register.

To check when a particular interrupt event occurs and to reset a particular event, the DISPC_IRQSTATUS register must be accessed. This register regroups the status of internal events in the module that generate an interrupt (read 0: no interrupt occurred; read 1: interrupt occurred; write 1: status bit reset). For more information about checking and clearing interrupt events, see Section 11.1.5, *Display Subsystem Register Manual*.

Table 11-60 describes the interrupts generated for the DISPC.

Table 11-60. DISPC Interrupts

Interrupt Name	Description
FLIPIMMEDIATEDONE_IRQ	Flip immediate done: Occurs when the DMA engine has acknowledged the immediate BA change, and software can write the new BA0.
FRAMEDONE1_IRQ	Frame done for LCD1 output: Active frame related to the LCD1 is complete and LCD1 output of the DISPC is disabled.
FRAMEDONE2_IRQ	Framedone for LCD2 output: Active frame related to the LCD2 is complete and LCD2 output of the DISPC is disabled.
FRAMEDONE3_IRQ	Frame done for LCD3 output: Active frame related to the LCD3 is complete and LCD3 output of the DISPC is disabled.
FRAMEDONETV_IRQ	Frame done for TV output: Active frame related to the TV output is complete and TV output of the DISPC is disabled.

Table 11-60. DISPC Interrupts (continued)

Interrupt Name	Description
FRAMEDONEWB_IRQ	Frame done for WB output: Active frame related to the WB is complete. First, it is used when the WB channel is connected to one of the pipelines to determine when the memory-to-memory transfer through DISPC is completed. Second, it is used when the WB channel is connected to one of the overlay output in nonreal-time mode to determine when the memory-to-memory transfer with overlay processing is completed.
VSYNC1_IRQ	VSYNC for primary LCD output: VSYNC interrupt for the primary LCD has occurred at the end of the frame.
VSYNC2_IRQ	VSYNC for secondary LCD output: VSYNC interrupt for the secondary LCD has occurred at the end of the frame.
VSYNC3_IRQ	VSYNC for third LCD output: VSYNC interrupt for the third LCD has occurred at the end of the frame.
EVSYNC_EVEN_IRQ	VSYNC for even field: EVSYNC_EVEN interrupt has occurred at the end of the frame (EVSYNC received and the field polarity is even) (HDMI)
EVSYNC_ODD_IRQ	VSYNC for odd field: EVSYNC_ODD interrupt has occurred at the end of the frame (EVSYNC received and the field polarity is odd) (HDMI)
ACBIASCOUNTSTATUS1_IRQ	AC bias count status for LCD1 output: AC bias transition counter has decremented to 0.
ACBIASCOUNTSTATUS2_IRQ	AC bias count status for LCD2 output: AC bias transition counter has decremented to 0.
ACBIASCOUNTSTATUS3_IRQ	AC bias count status for LCD3 output: AC bias transition counter has decremented to 0.
PROGRAMMEDLINENUMBER_IRQ	Programmed line number: The primary LCD has reached the user programmed line number.
VID1ENDWINDOW_IRQ	End of the VID1 window: The DMA engine has fetched all the data from memory for the VID1 for the current frame.
VID2ENDWINDOW_IRQ	End of the VID2 window: The DMA engine has fetched all the data from memory for the VID2 for the current frame.
VID3ENDWINDOW_IRQ	End of the VID3 window: The DMA engine has fetched all the data from memory for the VID3 for the current frame.
GFXENDWINDOW_IRQ	End of the graphics window: The DMA engine has fetched all the data from memory for the graphics for the current frame.
VID1BUFFERUNDERFLOW_IRQ	VID1 DMA buffer underflow: The input VID1 DMA buffer goes underflow.
VID2BUFFERUNDERFLOW_IRQ	VID2 DMA buffer underflow: The input VID2 DMA buffer goes underflow.
VID3BUFFERUNDERFLOW_IRQ	VID3 DMA buffer underflow: The input VID3 DMA buffer goes underflow.
WBBUFFEROVERFLOW_IRQ	WB DMA buffer overflow: The output WB DMA buffer goes overflow. It cannot occur when WB channel is used in memory-to-memory transfer mode but only in capture mode. In capture mode the timings are defined by the timer associated with the output. In memory-to-memory mode, there is timing constraint.
GFXBUFFERUNDERFLOW_IRQ	GFX DMA buffer underflow: The input graphics DMA buffer goes underflow.
PALETTEGAMMALOADING_IRQ	Gamma table loading: Gamma table in the graphics pipeline has been loaded using the DISPC DMA engine.
WBUNCOMPLETEERROR_IRQ	The write-back buffer has been flushed before being fully drained. In WB capture mode, if the new frame starts before the WB DMA buffers are fully drained (onto external memory), then the contents of the WB DMA buffers are lost (implying last few pixels/lines are corrupted in the captured frame in memory). This interrupt is an indication of that, and will trigger every frame.
OCPERROR_IRQ	OCP error: L3_MAIN interconnect has sent SResp = ERR

Table 11-60. DISPC Interrupts (continued)

Interrupt Name	Description
SYNCLOST1_IRQ	Synchronization lost on LCD1 output: Occurs when VSYNC width/front or back porches are not wide enough to load the pipelines with data (LCD output).
SYNCLOST2_IRQ	Synchronization lost on LCD2 output: Occurs when VSYNC width/front or back porches are not wide enough to load the pipelines with data (LCD output).
SYNCLOST3_IRQ	Synchronization lost on LCD3 output: Occurs when VSYNC width/front or back porches are not wide enough to load the pipelines with data (LCD output).
SYNCLOSTTV_IRQ	Synchronization lost on TV output: Occurs when porches are not wide enough to load the pipelines with data (TV output connected to the HDMI).
WAKEUP_IRQ	Wakeup: Occurs when the SWakeUp signal is asserted.

11.2.4.5 DISPC DMA Requests

The DMA synchronization line, DISPC_DREQ, is connected to the device DMA controllers (see Table 11-59, *DMA Requests*). This DMA request is not a classical one, but rather a synchronization signal between the DISPC and device DMA. The device DMA is informed that a programmable number of lines are output to the LCD and that a system memory can be updated. This request is related to the interrupt event PROGRAMMEDLINENUMBER_IRQ described in Table 11-60. This allows the device DMA channel to be synchronized with the internal DMA controller in the display subsystem.

In other words, it allows synchronizing a memory-to-memory frame buffer update based on the scan line of the frame buffer in system memory (SDRAM or SRAM) by the DISPC. The DISPC_DREQ request is generated at a programmable line number defined in the DISPC_LINE_NUMBER[11:0] LINENUMBER bit field. This process allows an application to use a single frame buffer and to update it after a certain number of lines are read by the DISPC.

11.2.4.6 DISPC DMA Engine

The DMA engine:

- Supplies data (encoded pixel data and gamma curve) from memories to the GFX, VID1, VID2, and VID3 pipelines through the interconnect based on the configuration of the DISPC and pipeline setting.
- Stores encoded pixel data from GFX/VID pipelines or overlays to memories through the WB pipeline and interconnect based on the configuration of the DISPC and WB pipeline setting.

Each pipeline has a dedicated buffer and a channel with independent settings. Table 11-61 lists the default size and allocation of the DMA buffer. Each DMA buffer is divided into two spaces, top and bottom. Depending on the application, a DMA buffer space can be associated to a pipeline or merged with other spaces. The total number of spaces for each pipeline is from 0 (pipeline inactive) to the number of pipelines × 2 (in that case, all the DMA buffers are associated to a single pipeline). The sum of the number of spaces allocated for each pipeline must not be greater than the maximum number of available spaces. The correct number of spaces must be allocated to ensure no underflow. The spaces allocated to each pipeline must be greater than or equal to the minimum number of spaces required to support the throughput and system latency. The space assignations are done in the DISPC_GLOBAL_BUFFER register.

Table 11-61. DISPC DMA Buffer Size

Pipelines	DMA Buffer Size
GFX	2 lines × 640 × 128 bits
VID1	2 lines × 1024 × 128 bits
VID2	2 lines × 1024 × 128 bits
VID3	2 lines × 1024 × 128 bits
WB	2 lines × 1024 × 128 bits

Figure 11-44 is an overview of the DMA engine.

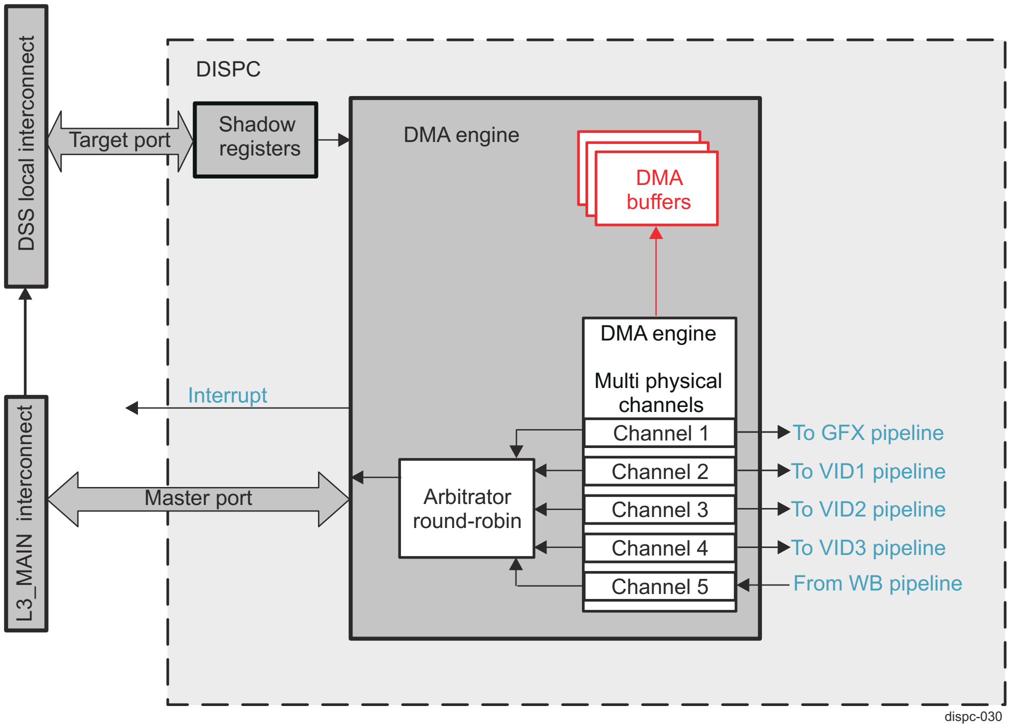


Figure 11-44. DISPC DMA Engine Overview

11.2.4.6.1 DISPC Addressing and Bursts

For each line to be fetched/stored, the DMA engine:

- Calculates the pixel address
- Aligns the address
- Defines the burst structure:
  - Type of burst (1D- or 2D-burst structure)
  - Length of the burst

The DMA engine generates scan addresses to read and write data to and from system memory. The base address defines the start address of the first pixel, and then the address is incremented based on the number of pixels per line, offset between two consecutive lines and number of lines. The byte address of each pixel in the frame buffer in the system memory is determined by:

Pixel address = Base address + x × bpp + (y × ((width × bpp) + increment))

where:

- Base address corresponds to the base address (for YUV–NV12 or YUV4:2:0-NV21 format) defined by:
  - DISPC_GFX_BA_0[31:0] BA bit field
  - DISPC_GFX_BA_1[31:0] BA bit field
  - DISPC_VIDp_BA_0[31:0] BA bit field
  - DISPC_VIDp_BA_1[31:0] BA bit field
  - DISPC_VIDp_BA_UV_0[31:0] BA bit field
  - DISPC_VIDp_BA_UV_1[31:0] BA bit field
- bpp corresponds to the number of bits per pixel defined by the DISPC_GFX_ATTRIBUTES[4:1] FORMAT bit field or the DISPC_VIDp_ATTRIBUTES[4:1] FORMAT bit field.
- width corresponds to the number of pixels per line defined by the DISPC_GFX_SIZE[10:0] SIZEX + 1 bit field or DISPC_VIDp_SIZE[10:0] SIZEX + 1 bit field.

- increment corresponds to the number of bytes to skip between two contiguous lines defined by the DISPC_GFX_ROW_INC[31:0] ROWINC – 1 bit field or the DISPC_VIDp_ROW_INC[31:0] ROWINC – 1 bit field.
- x corresponds to the pixel position on the x-axis.
- y corresponds to the pixel position on the y-axis.

Note

For YUV format, the pixel values are defined in two buffers (Y and UV). The base address of the Y buffers is defined in the DISPC_VIDp_BA_j[31:0] BA bit field. The base address of the UV buffers is defined in the DISPC_VIDp_BA_UV_j[31:0] BA bit field.

Table 11-62 summarizes the register settings for a simple access of a picture in the system memory.

Table 11-62. DISPC Register Settings for Accessing Image in Internal Memory

Registers	Value
DISPC_GFX_BA_j/DISPC_VIDp_BA_j/DISPC_WB_BA_j	PBA, the physical base address of image in the memory
DISPC_VIDp_BA_UV_j/DISPC_WB_BA_UV_j	PBA, the physical base address of UV buffers image in the memory
DISPC_GFX_PIXEL_INC/DISPC_VIDp_PIXEL_INC/DISPC_WB_PIXEL_INC	1 or other in pixel incremental value
DISPC_GFX_ROW_INC/DISPC_VIDp_ROW_INC/DISPC_WB_ROW_INC (1)	1 or other in row incremental value

(1) The DISPC_WB_ROW_INC register can be used only in 2D mode (using the Tiler). In order to use the DISPC_WB_ROW_INC register, the DISPC_WB_ATTRIBUTES[8] BURSTTYPE bit must be set to 1.

An interconnect request (128 bits) corresponds to one or several pixels, depending on the bits per pixel. Therefore, the DMA engine determines the appropriate burst sequence to optimize the fetching/storing of each new line. The DMA engine must prevent a single burst from crossing two lines. The DMA engine supports bursts of 2 × 128 bits, 4 × 128 bits, and 8 × 128 bits. The default burst size at reset time is 8 × 128 bits. The maximum burst size can be configured for each pipeline by setting the DISPC_GFX_ATTRIBUTES[7:6] BURSTSIZE or DISPC_VIDp_ATTRIBUTES[15:14] BURSTSIZE bit field. Because the burst size must be aligned to the burst boundary, in case of misalignment, the DMA engine may issue one request and/or smaller burst requests. Two types of burst are present which can be configured from the DISPC_GFX_ATTRIBUTES[29] BURSTTYPE or DISPC_VIDp_ATTRIBUTES[29] BURSTTYPE bit. Also, a force function is present configured from the DISPC_GFX_ATTRIBUTES[16] FORCE1DTILEDMODE or DISPC_VIDp_ATTRIBUTES[20] FORCE1DTILEDMODE or DISPC_WB_ATTRIBUTES[20] FORCE1DTILEDMODE bit for the following burst types:

- 1-D burst is used if the fetch/storage is linear in memory through the TILER. There is no rotation of the frame buffer. The frame buffer is not tiled.
- 2-D burst is used if the frame buffer is tiled.

Even if the DISPC_VIDp_ROW_INC register does not equal 1, the user can select 2-D burst. 2-D burst is used when the DISPC is configured to read one field of a frame by accessing only the even and odd lines.

Note

The burst size is initialized once at configuration and can be changed when the DISPC is disabled.

11.2.4.6.2 DISPC Immediate Base Address Flip Mechanism

The Flip Immediate mechanism is used to change of the fly the content of the frame buffer which is currently being displayed. The mechanism allows multiple changes (flips) of the base address (BA) during a frame. The mechanism is available for all pipelines (VID1, VID2, VID3, and GFX). Changes to the BA can be applied during the same line, or during different lines.

The following considerations must be considered:

- Data fetching from a new immediate BA is aligned with the data fetch mechanism of the DISPC DMA engine itself, and not with HSYNC or any other DISPC timing signal. After new VSYNC frame pulse, new BA is taken by hardware, as soon as the first set of lines is sent to internal pipeline of the DMA engine.
  - If multiple new BAs are written during the same line (before DMA engine acknowledges the first BA change), then programming steps 1 and 2 must be applied for each new BA (see the programming sequence in this section). The DMA takes only the last BA provided within the current line scan.
  - If multiple new BAs are written during different lines within the current frame, then programming steps 1 and 2 must be applied for each new BA (see the programming sequence in this section). The DMA takes the new BA each time it is updated.
- Immediate BA change cannot be achieved synchronously for two or more pipelines.
- Immediate BA change is possible only for BA0; it is not possible for BA1. Interlaced mode using BA0 and BA1 is not supported.
- Immediate BA change is supported only in RGB color space.

The Flip Immediate mechanism programming sequence follows:

1. Software writes the new immediate BA to the DISPC_GFX_BA_j register (where j=0) for the GFX pipeline, or to the DISPC_VID1_BA_j / DISPC_VID2_BA_j / DISPC_VID3_BA_j register (where j=0) for the required VID pipeline.
2. Software sets to 0x1 the corresponding EN bit for the required pipeline in the DISPC_BA0_FLIPIMMEDIATE_EN register.
3. The DISPC DMA engine, after completing its current line (or set of lines) fetch, takes the new immediate BA.
4. Hardware resets the EN bit in DISPC_BA0_FLIPIMMEDIATE_EN register and asserts a flip immediate IRQ. The IRQ can be enabled through the DISPC_IRQENABLE[31] FLIPIMMEDIATEDONE_EN bit. The status of the event is available in the DISPC_IRQSTATUS[31] FLIPIMMEDIATEDONE_IRQ bit.

11.2.4.6.3 DISPC DMA Buffers

11.2.4.6.3.1 DISPC READ DMA Buffers (GFX and VID Pipelines)

When the vertical front porch (VFP) period starts after the last horizontal front porch (HFP) of the last line or the external VSYNC is received, the DMA buffers are flushed according to the selected output associated with the pipeline. The DMA engine restarts fetching data from the memory through the L3_MAIN interconnect. Enabling or disabling the DISPC flushes the DMA buffers (except the WB DMA buffers).

Programmable high and low thresholds, independent for each DMA buffer, are used by the DMA engine to start and stop requesting data to the L3_MAIN interconnect.

- When low threshold (set in the DISPC_GFX_BUF_THRESHOLD[15:0] BUFLOWTHRESHOLD or DISPC_VIDp_BUF_THRESHOLD[15:0] BUFLOWTHRESHOLD bit field) is reached, the DMA engine starts a request on the L3_MAIN interconnect to fill the DMA buffer.
- When high threshold (set in the DISPC_GFX_BUF_THRESHOLD[31:16] BUFHIGHTHRESHOLD or DISPC_VIDp_BUF_THRESHOLD[31:16] BUFHIGHTHRESHOLD bit field) is reached, the DMA engine stops requesting encoded pixels.

To avoid underflow at the beginning of a frame and have sufficient encoded pixel data to start some processing, a preloading of the DMA buffer is configurable between a fixed value of bytes or the high threshold value. The preload ensures a minimum number of pixels present in the buffer. When the preload value is reached, the associated channel must start pulling pixels out of the DMA buffer. To enable the preload based on the value entered in the DISPC_GFX_PRELOAD[11:0] PRELOAD or DISPC_VIDp_PRELOAD[11:0] PRELOAD bit field, the DISPC_GFX_ATTRIBUTES[11] BUFPRELOAD bit, or the DISPC_VIDp_ATTRIBUTES[19] BUFPRELOAD bit must be set to 0x0.

Note

When self-refresh mode is selected, meaning the data in the DMA buffers are used for multiple frames, and at the end of each frame, the DMA buffers are not flushed.

11.2.4.6.3.2 DISPC WRITE DMA Buffer (WB Pipeline)

Two modes are supported by the WB channel, selectable through the DISPC_WB_ATTRIBUTES[19] WRITEBACKMODE bit:

- Capture mode, WRITEBACKMODE bit set to 0: One of the overlay outputs going to LCD or TV outputs is captured and at the same time the data are sent on the output. The WB timings are controlled by the LCD or TV timings. The output of a VID/GFX pipeline is not sent to the write back, when the WB pipeline is connected to an overlay used in capture mode.
- Memory-to-memory mode, WRITEBACKMODE bit set to 1: One of the overlay outputs or one of the pipelines is captured to perform a memory-to-memory transfer, with some processing by the DISPC (rescaling, overlaying, color space conversion, etc.).

In capture mode: The WB DMA buffers are flushed when the VFP period starts after the HFP following the last line, or when the external VSYNC is received, depending on which output (LCD/TV) the WB pipeline is capturing data, if the programmed value in DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNT bit field is set to 0. If the programmed value in DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNT bit field is set to N (1:255), the write buffers DMA are flushed N lines later. The DMA engine starts storing data to memory through the L3_MAIN-based interconnect as soon as enough data is available for the programmed burst size. When enabling/disabling the DISPC, the DMA buffers are flushed. The programmable thresholds low and high are used by the DMA engine to start and stop sending data to the L3_MAIN interconnect.

Note

If the DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNT bit field is set to 0, the WB is reinitialized at the end of the last line of a frame at the beginning of the VFP signal. To let the WB complete the data write to the external memory, the highest possible value compatible with the vertical blanking period must be set.

Note

In WB capture mode, if a new frame starts before the WB DMA buffers contents are fully written onto external memory, then the contents of the WB DMA buffers are lost (implying last few pixels/lines are corrupted in the captured frame in memory). The DISPC_IRQSTATUS[26] WBUNCOMPLETEERROR_IRQ interrupt bit indicates this situation and triggers every frame. The WBUNCOMPLETEERROR interrupt can be enabled through the DISPC_IRQENABLE[26] WBUNCOMPLETEERROR_EN register bit.

Software can avoid this by delaying the flush of WB DMA buffers through proper programming of the DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNT bit field.

In memory-to-memory mode: The WB pipeline is not synchronized to any internal or external timing generator. The WB pipeline stores the output of one of the overlay outputs or one of the pipelines. When enabling or disabling the DISPC, the DMA buffers are flushed. The programmable thresholds low and high are used by the DMA engine to start and stop sending data to the L3_MAIN interconnect.

Programmable high and low thresholds are used by the DMA engine to start and stop sending data to the L3_MAIN interconnect.

- When high threshold (set in the DISPC_WB_BUF_THRESHOLD[31:16] BUFHIGHTHRESHOLD bit field) is reached, the DMA engine starts sending data on the L3_MAIN interconnect to empty buffer.
- When low threshold (set in the DISPC_WB_BUF_THRESHOLD[15:0] BUFLOWTHRESHOLD bit field) is reached, the DMA engine stops sending encoded pixels.

At the end of the frame, to completely drain the DMA buffer, some smaller bursts (even single requests) must be issued. To limit the number of interconnect requests from the DISPC, a number of IDLE cycles between requests can be inserted. IDLE cycles can be inserted only when WB is used in memory-to-memory mode. It is ignored when WB is in capture mode.

The number of IDLE cycles between requests can be activated and determined by:

- Setting the DISPC_WB_ATTRIBUTES[27] IDLESIZE bit to 0x0 (default value) and entering the number of idles between requests in the DISPC_WB_ATTRIBUTES[31:28] IDLENUMBER bit field. Idle numbers vary from 0 to 15.
- Setting the DISPC_WB_ATTRIBUTES[27] IDLESIZE bit to 0x1, which considers the size of the burst (the DISPC_WB_ATTRIBUTES[15:14] BURSTSIZE bit field) to determine the number of IDLE cycles.
  - If BURSTSIZE = 0x0, then the number of IDLE cycles equals IDLENUMBER (0 to 15).
  - If BURSTSIZE = 0x1, then the number of IDLE cycles equals IDLENUMBER × 4 (0 to 60).
  - If BURSTSIZE = 0x2, then the number of IDLE cycles equals IDLENUMBER × 8 (0 to 120).

11.2.4.6.4 DISPC MFLAG Mechanism and Arbitration

The MFLAG mechanism allows a dynamic increase of the priority of DISPC real-time traffic, when required, based on the fullness of the DISPC DMA read and write buffers.

The mechanism is implemented for all DMA buffers (GFX, VID1, VID2, VID3 and WB). Only high-priority pipelines can contribute to the MFLAG mechanism.

Programmable buffer thresholds (hysteresis) for each pipeline are used to indicate when the local MFLAG signal for each pipeline is generated. All local MFLAG signals are ORed to generate a single DSS MFLAG out band signal, which is provided to the L3 interconnect for granting OCP requests. The out band DSS MFLAG signal is asynchronous to any ongoing OCP transaction.

The threshold for each pipeline corresponds to the fullness of the associated DMA buffer, and is defined by two threshold parameters:

- HT_MFLAG: High threshold.
  - For read access from the GFX, VID1, VID2, and VID3 pipelines, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes low (deasserted).
  - For write access from the WB pipeline, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes high (asserted).
  - This threshold can be programmed in the following bit fields:
    - For the GFX pipeline in the DISPC_GFX_MFLAG_THRESHOLD[31:16] HT_MFLAG bit field
    - For the VID1 pipeline in the DISPC_VID1_MFLAG_THRESHOLD[31:16] HT_MFLAG bit field
    - For the VID2 pipeline in the DISPC_VID2_MFLAG_THRESHOLD[31:16] HT_MFLAG bit field
    - For the VID3 pipeline in the DISPC_VID3_MFLAG_THRESHOLD[31:16] HT_MFLAG bit field
    - For the WB pipeline in the DISPC_WB_MFLAG_THRESHOLD[31:16] HT_MFLAG bit field
- LT_MFLAG: Low threshold.
  - For read access from the GFX, VID1, VID2, and VID3 pipelines, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes high (asserted).
  - For write access from the WB pipeline, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes low (deasserted).
  - This threshold can be programmed in the following bit fields:
    - For the GFX pipeline in the DISPC_GFX_MFLAG_THRESHOLD[15:0] LT_MFLAG bit field
    - For the VID1 pipeline in the DISPC_VID1_MFLAG_THRESHOLD[15:0] LT_MFLAG bit field
    - For the VID2 pipeline in the DISPC_VID2_MFLAG_THRESHOLD[15:0] LT_MFLAG bit field
    - For the VID3 pipeline in the DISPC_WB_MFLAG_THRESHOLD[15:0] LT_MFLAG bit field
    - For the WB pipeline in the DISPC_WB_MFLAG_THRESHOLD [15:0] LT_MFLAG bit field

By default, the MFLAG mechanism is disabled (DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field = 0x0), and the DSS MFLAG out band signal is low (deasserted). The arbitration scheme for the DISPC pipelines is the same as described in Section 11.2.4.6.7, *DISPC Arbitration*. That is, round-robin either between high-priority pipelines, or between normal-priority pipelines (if all pipelines are of normal priority).

When the DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field is set to 0x2, the MFLAG mechanism is enabled, and the DSS MFLAG out band signal is dynamically set to 0 or 1, depending on DMA buffers fullness and programmed threshold levels, as explained previously in this section. In this case, the arbitration scheme for DISPC pipelines is round-robin between those high-priority pipelines, which have asserted local MFLAG signals. If there are no high-priority pipelines with asserted local MFLAG signals, then the arbitration scheme is the same as described in Section 11.2.4.6.7, *DISPC Arbitration*.

The DISPC_GLOBAL_MFLAG_ATTRIBUTE[2] MFLAG_START bit defines additional rules for the MFLAG mechanism:

- If the MFLAG_START bit is set to 0x0 (default value), then in the beginning of the frame when the DMA buffer is empty, the local MFLAG signal of each pipeline is kept at 0 until PRELOAD is reached (for more information on preloading, see Section 11.2.4.6.3.1, *DISPC READ DMA Buffers (GFX and VID Pipelines)*. Then, based on the setting of the DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field, the MFLAG signal is generated and internal logic is arbitrating between pipeline requests.
- If the MFLAG_START bit is set to 0x1, then even in the beginning of the frame when the DMA buffer is empty, the DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field defines the generation of the MFLAG signal for each pipeline.

11.2.4.6.5 DISPC Predecimation

The predecimation process consists of downscaling an image by fetching only the necessary pixels in the memory. Vertical and horizontal predecimation are possible:

- Vertical predecimation: The picture stored in memory can be predecimated vertically by skipping lines. Burst mode is used to fetch the data when skipping lines. Only the lines that will be used by the DISPC are fetched from memory; the other lines are skipped. The DMA engine sends requests only for the useful lines using 1-D or 2-D burst, depending on the setting. The base address indicates the first valid pixel to fetch from memory. The number of lines to skip is set in the DISPC_GFX_ROW_INC[31:0] ROWINC or DISPC_VIDp_ROW_INC[31:0] ROWINC bit field.

  Note

  When 2-D burst mode is used, the access to data in memory is performed through the TILER module of DMM (for more details, see Section 15.1.2, *TILER Overview*, in Section 15.1, *Memory Subsystem*), and as a result a maximum of one line can be skipped.

- Horizontal predecimation: When fetching data from memory, it is possible to skip 1 of 2 pixel data containers, up to 1 of 16 pixel data containers, by setting the DISPC_GFX_PIXEL_INC[7:0] PIXELINC or DISPC_VIDp_PIXEL_INC[7:0] PIXELINC bit field to the number of pixel data containers to skip (n), multiplied by the size of a pixel data container (in bytes), + 1. See the following note for more details.

Note

The restriction to horizontal predecimation is that there is at least one useful pixel per 128-bit request. In that case, the DMA engine uses burst mode instead of singles to optimize the requests in terms of latency and SDRAM efficiency.

For RGB and YUV4:2:0 data formats, each pixel data container in memory holds 1 pixel. Thus, when configuring the PIXELINC bit field, the value of n equals the number of pixels to skip:

- For RGB format, one pixel data container = 32 bits = 1 pixel
- For YUV format:
  - One Y pixel data container = 8 bits = 1 pixel
  - One UV pixel data container = 16 bits = 1 pixel

For YUV4:2:2 format, each 32-bit pixel data container holds the Luma components for 2 pixels, and the Chrominance component of 1 pixel (see Figure 11-45). Therefore, for the valid values of the PIXELINC bit field in the case of the following YUV4:2:2 format, caution must be taken because n equals the number of pixel data containers to skip, not the number of pixels:

- For n = 1, PIXELINC = 5
- For n = 2, PIXELINC = 9
- For n = 3, PIXELINC = 13
- For n = 4, PIXELINC = 17, etc.

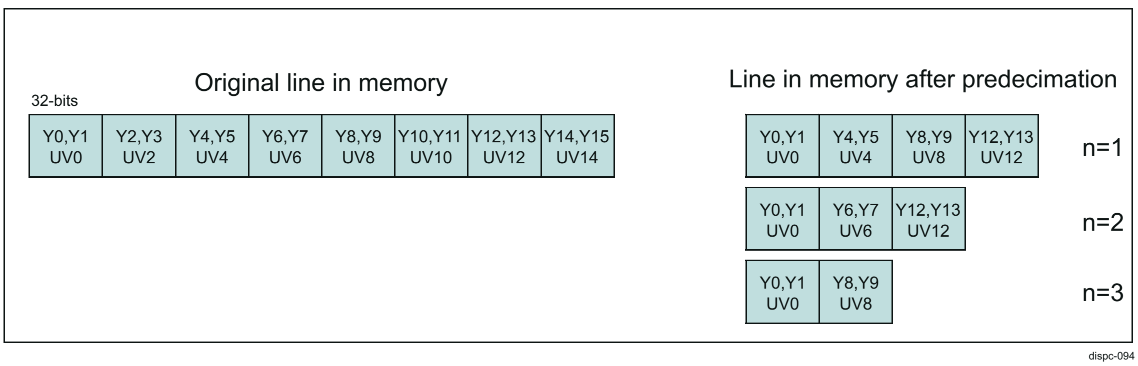


Figure 11-45. YUV4:2:2 Predecimation

11.2.4.6.6 DISPC Progressive-to-Interlaced Format Conversion

The DMA engine can be used to perform YUV4:2:0 NV12 and YUV4:2:0 NV21 progressive-to-interlaced data conversion with 0-degree orientation. Such conversion performed in the DSS consists of separating each progressive frame into two fields containing odd and even lines.This section provides generic approach details.

Two possible configurations are available, depending on the setting of the DISPC_VIDp_ATTRIBUTES[22] DOUBLESTRIDE bit, which defines the stride of each pixel value buffer for the YUV format. The following must be considered for both configurations:

- The DISPC_VIDn_BA_j[31:29] BA and DISPC_VIDn_BA_UV_j[31:29] BA bit fields, and the DISPC_VIDn_ATTRIBUTES[13:12] ROTATION bit field must be set to 0x0 to indicate 0-degree orientation.
- The DISPC_VIDn_BA_j[31:29] BA and DISPC_VIDn_BA_UV_j[31:29] BA bit fields must be set to 0x3, and the DISPC_VIDn_ATTRIBUTES[13:12] ROTATION bit field must be set to 0x2 to indicate 180-degree orientation.

Configuration 1 – YUV4:2:0 progressive to interlaced conversion

The DISPC_VIDp_ATTRIBUTES[22] DOUBLESTRIDE bit is set to 0x1. The CbCr container is twice the size of the Y container. All Luma and Chroma lines for even and odd fields are fetched from memory. The scaler unit of the respective pipeline can be used to downscale by 2 (through filtering) the fetched data to create the interlaced output. For more information about the scaler configuration, see Section 11.2.4.10.4, *Scaler Unit*.

Configuration 2 – YUV4:2:0 progressive to YUV4:2:2 interlaced conversion

The DISPC_VIDp_ATTRIBUTES[22] DOUBLESTRIDE bit is set to 0x0. The CbCr container is the same size as the Y container. The DISPC_VIDn_ROW_INC register for the respective pipeline must be configured so that only the Y data is vertically predecimated by 2 (for more information, see Section 11.2.4.6.5, *Predecimation*). The CbCr data must not be predecimated. As a result, only the even Luma lines for the even field and the odd Luma lines for the odd field are fetched from memory. To create the interlaced output, all the Chroma lines are fetched from memory.

11.2.4.6.7 DISPC Arbitration

The requests (reads or writes) sent to the L3_MAIN interconnect are pipelined and arbitrated in a round-robin scheme. The default arbitration scheme must be modified by setting the priority attribute of each pipeline as defined in the following bits:

- DISPC_GFX_ATTRIBUTES[14] ARBITRATION
- DISPC_VIDp_ATTRIBUTES[14] ARBITRATION
- DISPC_WB_ATTRIBUTES[14] ARBITRATION

By default, all pipelines have the same priority (normal), which means all pipeline requests are treated in a round-robin order manner. If one or more pipelines require a higher number of requests going to the L3_MAIN interconnect, its priority can be moved up to high priority. In this case, the high-priority pipeline is granted access before any pipeline in normal priority. If more than one active pipeline is in high priority, the behavior is the same as all active pipelines in normal priority. Normal active pipelines are not treated until all high active pipelines are finished. The ARBITRATION bit cannot be modified during the entire frame.

This functionality balances the bandwidth of the pipeline depending on its constraint. It can be used to give higher priority to the pipelines with real-time constraints versus non-real-time pipelines. For example, pipelines associated with the LCD output in stall mode must have lower priority than pipelines associated with TV output.

11.2.4.6.8 DISPC DMA Power Modes

11.2.4.6.8.1 DISPC DMA Low-Power Mode

Each DMA buffer is divided into two spaces. Each space can be associated with the pipeline or merged with other DMA buffers. The total number of DMA buffers for each pipeline is from 0 (pipeline inactive) to the number of pipelines × 2 (in that case all the DMA buffers are associated with a single pipeline). The sum of the number of DMA buffers allocated for each pipeline must not be greater than the maximum available. The correct number of DMA buffers must be allocated to ensure no underflow. The number of DMA buffers allocated to each pipeline must be greater than or equal to the minimum required to support the throughput and the system latency.

Note

When the number of buffers is changed, the thresholds must be reprogrammed to reflect the new configuration of the DMA buffer.

11.2.4.6.8.2 DISPC DMA Ultralow-Power Mode

In low-power mode, the L3_MAIN interconnect is used to fill up the DMA buffers to store all the data required to display a full frame. The L3_MAIN interconnect is not used to fetch new pixels for the following frames. The data in the DMA buffer are reused to display on the screen.

The setting of the mode is independent for each pipeline. One pipeline may have all the frame pixels in its DMA buffer and the other pipelines may have to refill their respective DMA buffers along the display scan because the frame buffer is too big to be stored in the DMA buffer.

The DMA buffers can be merged to optimize the L3_MAIN interconnect off time. Merging the DMA buffers into a single buffer can be used at the same time to improve ultralow-power mode (see Section 11.2.4.6.8.1, *Low-Power Mode*).

During the time in which the frames are fetched in the internal DMA buffer, MStandby must be asserted if the DISPC_SYSCONFIG[13:12] MIDLEMODE bit field is set to 0x2 (smart-standby mode).

Two ultralow-power modes can be entered manually or automatically:

- Self-refresh mode: Starting self-refresh mode is done manually by setting the DISPC_GFX_ATTRIBUTES[15] SELFREFRESH or DISPC_VIDp_ATTRIBUTES[15] SELFREFRESH bit to 0x1 after capturing a frame in the DMA buffers. Self-refresh mode is stopped by setting the SELFREFRESH bit to 0x0.
- Automatic self-refresh mode: By setting the DISPC_GFX_ATTRIBUTES[17] SELFREFRESHAUTO or DISPC_VIDp_ATTRIBUTES[17] SELFREFRESHAUTO bit to 0x1, the transition from off to on self-refresh mode is done by hardware after capturing the first frame. The hardware reflects the status of the self-refresh mode by setting the SELFREFRESH bit to 0x1, which means that the data are read inside the DMA buffer without accessing the interconnect and system memory during the frame. Setting the SELFREFRESH bit to 0x0 updates the DMA buffer.

Note

The WB pipeline does not support ultralow-power mode.

11.2.4.7 DISPC Rotation and Mirroring

The DISPC provides flexible mechanisms for efficient implementation of rotation using the DISPC, its DMA engine, and the rotation engine of the TILER. The rotation is handled only through the TILER, which supplies the encoded pixels to the DISPC.

Note

No rotation or mirroring is supported when accessing the SDRAM directly.

The TILER supports:

- Rotation: 0-, 90-, 180-, and 270-degree views
- Mirroring
- Any combination of rotation and mirroring (for example, Rot-90 + mirroring)

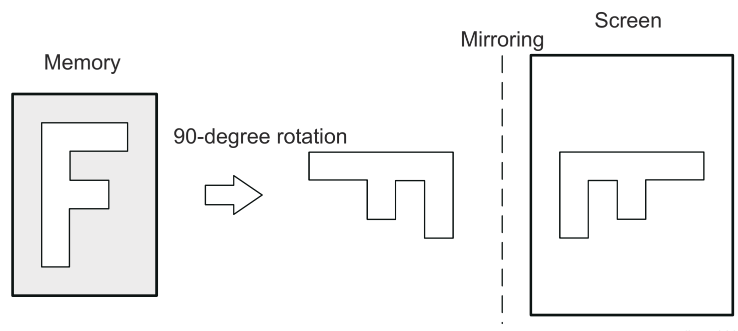


Figure 11-46. DISPC 90-Degree Rotation With Mirroring

When accessing YUV4:2:2 data, each 32-bit value loaded into the DMA buffer can represent either:

- Two consecutive pixels on the same line (for instance, 0- to 180-degree rotation)
- Two pixels adjacent vertically (for instance, 90- to 270-degree rotation)

The reading from the DMA buffer supports the extraction of the two pixels, regardless of the rotation. When the pixels are not consecutive on the same line (90- to 270-degree rotation), the chrominance sample of the first pixel of each 32-bit value is duplicated for the second pixel in the same 32-bit value.

The rotation flag DISPC_VIDn_ATTRIBUTES[13:12] ROTATION and DISPC_VIDn_ATTRIBUTES[4:1] FORMAT bit fields define the processing to extract the pixels from YUV4:2:2 32-bit values. Software must ensure that the settings of the ROTATION and FORMAT bit fields are coherent with the rotation and mirroring defined through the address format in the TILER-specific address map. For more information, see Section 15.2, *Dynamic Memory Manager*, in*Section 15.1, Memory Subsystem*.

Note

For YUV4:2:0 NV12, 2D tiled memory access cases, the DISPC_VIDn_ATTRIBUTES[13:12] ROTATION bit field should be set properly to match with the intended rotation of the use case.

Table 11-63 describes the register settings of the DISPC when accessing, rotating, and mirroring an image using the TILER. A physical base address (PBA) for each rotation is determined and set as the buffer address (BA). The row incremental is determined and set in ROW_INC. The value of the pixel increment, PIXEL_INC, is set to 0x1 (contiguous pixels).

Table 11-63. DISPC Register Settings for Rotation Using TILER

Rotation	Registers	Rotation With and Without Mirroring
0 degree	DISPC_GFX_BA_j/DISPC_VIDp_BA_j/DISPC_WB_BA_j	PBA0
	DISPC_GFX_PIXEL_INC/DISPC_VIDp_PIXEL_INC/DISPC_WB_PIXEL_INC	1 to 16
	DISPC_GFX_ROW_INC/DISPC_VIDp_ROW_INC/DISPC_WB_ROW_INC	ROW0
90 degrees	DISPC_GFX_BA_j/DISPC_VIDp_BA_j/DISPC_WB_BA_j	PBA90
	DISPC_GFX_PIXEL_INC/ DISPC_VIDp_PIXEL_INC/DISPC_WB_PIXEL_INC	1 to 16
	DISPC_GFX_ROW_INC/ DISPC_VIDp_ROW_INC/DISPC_WB_ROW_INC	ROW90
180 degrees	DISPC_GFX_BA_j/DISPC_VIDp_BA_j/DISPC_WB_BA_j	PBA180
	DISPC_GFX_PIXEL_INC/DISPC_VIDp_PIXEL_INC/DISPC_WB_PIXEL_INC	1 to 16
	DISPC_GFX_ROW_INC/ DISPC_VIDp_ROW_INC/DISPC_WB_ROW_INC	ROW180
270 degrees	DISPC_GFX_BA_j/DISPC_VIDp_BA_j/DISPC_WB_BA_j	PBA270
	DISPC_GFX_PIXEL_INC/DISPC_VIDp_PIXEL_INC/DISPC_WB_PIXEL_INC	1 to 16
	DISPC_GFX_ROW_INC/ DISPC_VIDp_ROW_INC/DISPC_WB_ROW_INC	ROW270

Note

For YUV format, in addition to the DISPC_VIDp_BA_j register used for the Y component, the DISPC_VIDp_BA_UV_j register must be set to define the base address of the UV frame buffer in memory.

The DISPC_WB_ROW_INC register can be used only in 2D mode (using the Tiler). In order to use the DISPC_WB_ROW_INC register, the DISPC_WB_ATTRIBUTES[8] BURSTTYPE bit must be set to 1.

The PBA rotation is determined by:

- PBA0 = PBA | (mode << 27) | (orientation << 29) | (1<<32)
- PBA90 = PBA | (mode << 27) | (orientation << 29) | (1<<32)
- PBA180 = PBA | (mode << 27) | (orientation << 29) | (1<<32)
- PBA270 = PBA | (mode << 27) | (orientation << 29) | (1<<32)

Where PBA is the physical base address of the image in the memory.

The ROW rotation is determined by:

- If 8 bits per pixel:
  – ROW0 = 16384: Width of the video picture in memory (in bytes) + 1
  – ROW90 = 8192: Width of the video picture in memory (in bytes) + 1
  – ROW180 = 16384: Width of the video picture in memory (in bytes) + 1
  – ROW270 = 8192: Width of the video picture in memory (in bytes) + 1
- If 16 bits per pixel:
  – ROW0 = 32768: Width of the video picture in memory (in bytes) + 1
  – ROW90 = 8192: Width of the video picture in memory (in bytes) + 1
  – ROW180 = 32768: Width of the video picture in memory (in bytes) + 1
  – ROW270 = 8192: Width of the video picture in memory (in bytes) + 1
- If 32 bits per pixel:
  – ROW0 = 32768: Width of the video picture in memory (in bytes) + 1
  – ROW90 = 16384: Width of the video picture in memory (in bytes) + 1
  – ROW180 = 32768: Width of the video picture in memory (in bytes) + 1
  – ROW270 = 16384: Width of the video picture in memory (in bytes) + 1

Table 11-64 and Table 11-65 list the DISPC rotation mode and rotation orientation definitions, respectively.

Table 11-64. DISPC Rotation Mode Definition

	8-Bit Tiled	16-Bit Tiled	32-Bit Tiled	Page Tiled
Mode	0	1	2	3

Table 11-65. DISPC Rotation Orientation Definition

Type of Orientation	Value
0-degree view	0x0
180-degree view with mirroring	0x1
0-degree view with mirroring	0x2
180-degree view	0x3
270-degree view with mirroring	0x4
270-degree view	0x5
90-degree view	0x6
90-degree view with mirroring	0x7

Note

For YUV4:2:0 progressive pixel format, because the value of the DISPC_VIDp_ROW_INC register is defined for the Y buffer, the DISPC_VIDp_ATTRIBUTES[22] DOUBLESTRIDE bit must be set to 1 when rotating the picture by 0 and 180 degrees, and must be reset to 0 when rotating the picture by 90 and 270 degrees.

For YUV4:2:0 interlaced pixel format, because the value of the DISPC_VIDp_ROW_INC register is defined for the Y buffer, the DISPC_VIDp_ATTRIBUTES[22] DOUBLESTRIDE bit must be set to 1 when rotating the picture by 0 and 180 degrees. Rotations of 90 and 270 degrees are not supported with this pixel format.

11.2.4.8 DISPC Memory Format

The graphic and video pipelines support various types of memory formats. Table 11-66 lists all supported formats for each pipeline.

Table 11-66. DISPC Memory Formats Supported

Formats	GFX	VID1	VID2	VID3	WB
xRGB12-4444	x	x	x	x	x
RGBx12-4444	x	x	x	x	x

Table 11-66. DISPC Memory Formats Supported (continued)

Formats	GFX	VID1	VID2	VID3	WB
ARGB16-4444	x	x	x	x	x
RGBA16-4444	x	x	x	x	x
RGB16-565	x	x	x	x	x
xRGB16-1555	x	x	x	x	x
ARGB16-1555	x	x	x	x	x
xRGB24-8888	x	x	x	x	x
RGBx24-8888	x	x	x	x	
RGB24-888	x	x	x	x	x
ARGB32-8888	x	x	x	x	x
RGBA32-8888	x	x	x	x	x
BGRA32-8888	x	x	x	x	x
UYUV4:2:2		x	x	x	x
YUV2 4:2:2		x	x	x	x
YUV4:2:0 – NV12		x	x	x	x
YUV4:2:0 – NV21		x	x	x	x

- xRGB12-4444 bpp data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused				R1				G1				B1				Unused				R0				G0				B0			

dss-T046

- RGBx12-4444 bpp data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1				G1				B1				Unused				R0				G0				B0				Unused			

dss-T047

- ARGB16-4444 bpp data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A1				R1				G1				B1				A0				R0				B0				G0			

dss-T048

- RGBA16-4444 bpp data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1				G1				B1				A1				R0				G0				B0				A0			

dss-T050

- xRGB16-1555 bpp data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	R1					G1					B1					U	R0					B0					G0				

dss-T062

- ARGB16-1555 bpp data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A1	R1					G1					B1					A0	R0					B0					G0				

dss-T051

- RGB16-565 bpp data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1					G1						B1					R0					G0						B0				

dss-T052

- xRGB24-8888 bpp data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Unused	R	G	B

dss-T053

- RGBx24-8888 bpp data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
R	G	B	Unused

dss-T054

- RGB24-888 bpp packed data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
B1	R0	G0	B0	+0x0
G2	B2	R1	G1	+0x4
R3	G3	B3	R2	+0x8

dss-T055

- ARGB32-8888 bpp data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
A	R	G	B

dss-T056

- RGBA32-8888 bpp data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
R	G	B	A

dss-T057

- BGRA32-8888 bpp data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
B	G	R	A

dss-T058

- UYVY4:2:2 data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Y1	Cr0	Y0	Cb0	+0x0
Y3	Cr2	Y2	Cb2	+0x4
Y5	Cr4	Y4	Cb4	+0x8
Y7	Cr6	Y6	Cb6	+0xC

dss-T059

- YUV2 4:2:2 data memory organization

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Cr0	Y1	Cb0	Y0	+0x0
Cr2	Y3	Cb2	Y2	+0x4
Cr4	Y5	Cb4	Y2	+0x8
Cr6	Y7	Cb6	Y6	+0xC

dss-T060

- YUV4:2:0-NV12 data memory organization (same for YUV4:2:0-NV12 with only UV in reverse order)

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Y3	Y2	Y1	Y0	+0x0
Y7	Y6	Y5	Y4	+0x4

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Cr1	Cb1	Cr0	Cb0	+0x0 + Offset
Cr3	Cb3	Cr2	Cb2	+0x4 + Offset
Cr5	Cb5	Cr4	Cb4	+0x8 + Offset
Cr7	Cb7	Cr6	Cb6	+0xC + Offset

dss-T061

11.2.4.9 DISPC Graphics Pipeline

The graphics pipeline input port is connected to the GFX FIFO, and to the four overlay managers or WB pipeline at its output. The pixel output is directed to an LCD, TV, or WB path by setting the DISPC_GFX_ATTRIBUTES[8] CHANNELOUT bit and the DISPC_GFX_ATTRIBUTES[31:30] CHANNELOUT2 bit field. Table 11-82 lists the bit field settings to orient a pipeline to an LCD, TV, or WB output. The default value directs the GFX pipeline to LCD1. The GFX pipeline can be enabled by setting the DISPC_GFX_ATTRIBUTES[0] ENABLE bit to 0x1.

Note

It is not possible to change the direction of the GFX pipeline on the fly. If the graphics pipeline must be connected to an overlay manager different from the one to which it is currently connected, then the following steps must be performed:

1. Disable the GFX pipeline by setting the DISPC_GFX_ATTRIBUTES[0] ENABLE bit to 0x0.
2. Direct the GFX pipeline to the new overlay manager by modifying the DISPC_GFX_ATTRIBUTES[8] CHANNELOUT and [31:30] CHANNELOUT2 bits.
3. Disable the DISPC output which corresponds with the overlay manager to which the GFX pipeline will be connected next. This is done by setting the in the following bits to 0x0 for the listed outputs:
   - The DISPC_CONTROL1[0] LCDENABLE bit for LCD1 output
   - The DISPC_CONTROL2[0] LCDENABLE bit for LCD2 output
   - The DISPC_CONTROL3[0] LCDENABLE bit for LCD3 output
   - The DISPC_CONTROL1[1] TVENABLE bit for TV output
4. Enable the GFX pipeline by setting the DISPC_GFX_ATTRIBUTES[0] ENABLE bit to 0x1.
5. Enable the DISPC output that corresponds with the overlay manager to which the GFX pipeline will be connected. This is done by setting the following bits to 0x1 for the listed outputs:
   - The DISPC_CONTROL1[0] LCDENABLE bit for LCD1 output
   - The DISPC_CONTROL2[0] LCDENABLE bit for LCD2 output
   - The DISPC_CONTROL3[0] LCDENABLE bit for LCD3 output
   - The DISPC_CONTROL1[1] TVENABLE bit for TV output

The pipeline consists of a programmable replication logic and an antiflicker filter. The replication logic is used to convert the RGB pixel formats into an ARGB40-based format. The antiflicker filter processes the graphics data in RGB format to remove some of the vertical flicker.

Note

The GFX pipeline does not include a scaler.

Table 11-66 lists the input formats supported by the graphics pipeline.

Figure 11-47 shows the graphics pipeline.

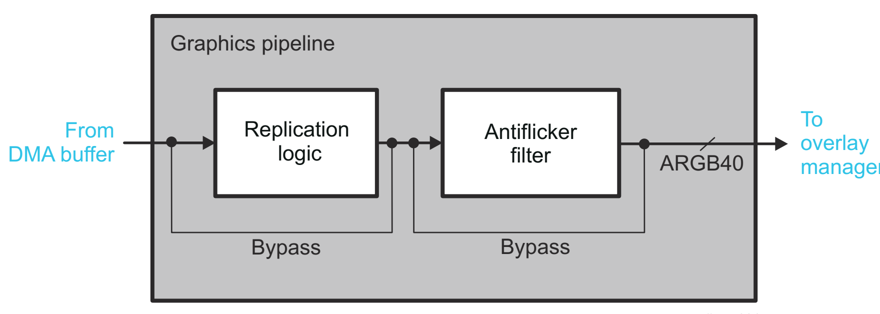


Figure 11-47. DISPC Graphics Pipeline

11.2.4.9.1 DISPC Replication Logic

The replication logic increases the color depth of the graphics-encoded pixels (from true color RGB 12, and 16 to 40 bpp).

- When the replication logic is enabled by setting the DISPC_GFX_ATTRIBUTES[5] REPLICATIONENABLE bit to 0x1, the MSBs are copied to the missing LSB. Table 11-67 describes the remapping of the RGB pixels into ARGB 40-bit values.

Table 11-67. DISPC Replication Enabled: RGB Pixel Formats Remapping Into ARGB40-10.10.10.10

	A[9:0]	R[9:0]	G[9:0]	B[9:0]
Format	**MSB LSB**	**MSB LSB**	**MSB LSB**	**MSB LSB**
xRGB12-4444	1111111111	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]
RGBx12-4444	1111111111	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]
RGB16-565	1111111111	R[4:0]R[4:0]	G[5:0]G[5:2]	B[4:0]B[4:0]
xRGB16-1555	1111111111	R[4:0]R[4:0]	G[4:0]G[4:0]	B[4:0]B[4:0]
ARGB16-1555	AAAAAAAAAA	R[4:0]R[4:0]	G[4:0]G[4:0]	B[4:0]B[4:0]
ARGB16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]
RGBA16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]

- When the replication logic is disabled by setting the DISPC_GFX_ATTRIBUTES[5] REPLICATIONENABLE bit to 0x0, the encoded pixel values are shifted to the MSB boundary of the 24-bit format. The missing bit values are filled with 0s. Table 11-68 describes the remapping of the RGB pixels into ARGB 40-bit values.

Table 11-68. DISPC Replication Disabled: RGB Pixel Formats Remapping Into ARGB40-10.10.10.10

	A[7:0]	R[7:0]	G[7:0]	B[7:0]
Format	**MSB LSB**	**MSB LSB**	**MSB LSB**	**MSB LSB**
xRGB12-4444	1111111111	R[3:0]000000	G[3:0]000000	B[3:0]000000
RGBx12-4444	1111111111	R[3:0]000000	G[3:0]000000	B[3:0]000000
RGB16-565	1111111111	R[4:0]00000	G[5:0]0000	B[4:0]00000
xRGB16-1555	1111111111	R[4:0]00000	G[4:0]00000	B[4:0]00000
ARGB16-1555	AAAAAAAAAA	R[4:0]00000	G[4:0]00000	B[4:0]00000
ARGB16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]000000	G[3:0]000000	B[3:0]000000
RGBA16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]000000	G[3:0]000000	B[3:0]000000

11.2.4.9.2 DISPC Antiflicker Filter

The antiflicker filter processes the graphics data to remove some of the vertical flicker. It is based on a 3-tap FIR filter with fixed coefficients. For each pixel to be output from the graphics pipeline, the pixel above and below the current line must be read from the DMA graphics FIFO. Therefore, three lines of pixels must be stored in the DMA graphics FIFO.

The antiflickering equations for A, R, G, and B components are:

$A_{out}(x,y) = 0.25 \times A_{in}(x,y - 1) + 0.5 \times A_{in}(x,y) + 0.25 \times A_{in}(x,y + 1)$

$R_{out}(x,y) = 0.25 \times R_{in}(x,y - 1) + 0.5 \times R_{in}(x,y) + 0.25 \times R_{in}(x,y + 1)$

$G_{out}(x,y) = 0.25 \times G_{in}(x,y - 1) + 0.5 \times G_{in}(x,y) + 0.25 \times G_{in}(x,y + 1)$

$B_{out}(x,y) = 0.25 \times B_{in}(x,y - 1) + 0.5 \times B_{in}(x,y) + 0.25 \times B_{in}(x,y + 1)$

For the first line of processing, because there is no pixel above, the value (x,y) is duplicated.

$Out(x,y) = 0.25 \times In(x,y) + 0.5 \times In(x,y) + 0.25 \times In(x,y + 1)$

For the last line of processing, because there is no pixel below, the value (x,y) is duplicated.

$Out(x,y) = 0.25 \times In(x,y-1) + 0.5 \times In(x,y) + 0.25 \times In(x,y)$

Note

Antiflicker filtering is supported only in RGB formats.

Antiflickering is not supported for pictures with fewer than two lines. In this case, the user must disable the antiflickering processing.

By default, the antiflicker filtering is disabled. It can be enabled by setting the DISPC_GFX_ATTRIBUTES[24] ANTIFLICKER bit to 0x1.

11.2.4.10 DISPC Video Pipelines

Three identical video pipelines are available, VID1, VID2, and VID3. Each video pipeline is connected to its video FIFO controller for the input port and to the four overlay managers, LCD1, LCD2, LCD3, and TV or WB pipeline. The pixel output is directed to the LCD, TV, or WB path by setting the DISPC_VIDp_ATTRIBUTES[16] CHANNELOUT bit and the DISPC_VIDp_ATTRIBUTES[31:30] CHANNELOUT2 bit field. Table 11-82 summarizes the bit field settings to orient a pipeline to LCD, TV, or WB output. The default value directs all video pipelines to LCD1.

Note

It is not possible to change the direction of the video pipelines on the fly. If a video pipeline needs to be connected to a different overlay manager than what it is currently connected, then the following steps need to be performed:

1. Disable the VIDp pipeline by setting the DISPC_VIDp_ATTRIBUTES[0] ENABLE bit to 0x0.
2. Direct the VIDp pipeline to the new overlay manager by modifying the [16] CHANNELOUT bit and the [31:30] CHANNELOUT2 bit in the DISPC_VIDp_ATTRIBUTES register.
3. Disable the DISPC output that corresponds with the overlay manager to which the VIDp pipeline will be connected next. This is done by setting the following bits to 0x0 for the listed outputs:
   - The DISPC_CONTROL1[0] LCDENABLE bit for LCD1 output
   - The DISPC_CONTROL2[0] LCDENABLE bit for LCD2 output
   - The DISPC_CONTROL3[0] LCDENABLE bit for LCD3 output
   - The DISPC_CONTROL1[1] TVENABLE bit for TV output
4. Enable the VIDp pipeline by writing 0x1 to the DISPC_VIDp_ATTRIBUTES [0] ENABLE bit.
5. Enable the DISPC output that corresponds with the overlay manager to which the VIDp pipeline will be connected next. This is done by setting the following bits to 0x1 for the listed outputs:
   - The DISPC_CONTROL1[0] LCDENABLE bit for LCD1
   - The DISPC_CONTROL2[0] LCDENABLE bit for LCD2 output
   - The DISPC_CONTROL3[0] LCDENABLE bit for LCD3 output
   - The DISPC_CONTROL1[1] TVENABLE bit for TV output

A video pipeline consists of a scaler unit, color space conversion (CSC) unit, VC-1 range mapping unit, and some programmable replication logic. The order of the video pipeline unit can be configured in two manners (see Figure 11-48 and Figure 11-49):

- Configuration 1 (YUVCHROMARESAMPLING = 0): VC-1 range mapping unit followed by a CSC unit and then a scaler unit. The configuration is used to support RGB, ARGB, and RGBA formats and YUV4:2:2 in backward mode for both data types. Each block can be independently bypassed.
- Configuration 2 (YUVCHROMARESAMPLING = 1): VC-1 range mapping unit followed by a scaler unit and then a CSC unit. The configuration is used to support RGB, ARGB, and RGBA formats and YUV4:2:2, YUV420-NV12, and YUV420-NV21 formats, taking advantage of the scaler to resample the chrominance using five taps horizontally and three or five taps vertically. Each block can be independently bypassed.

The DISPC_VIDn_ATTRIBUTES2[8] YUVCHROMARESAMPLING bit controls the order of the scaler unit in the video pipeline:

- When the YUVCHROMARESAMPLING bit is set to 0x0, the video pipeline is in configuration 1, and the scaler comes after the CSC unit. In case of YUV input data in 4:2:2 format, the chrominance resampling (4:2:2 to 4:4:4 format) is done by averaging the chrominance adjacent samples for only 0 degrees (zero rotation), because other rotation (90/180/270 degrees) is not supported in this mode. For more information about the supported chrominance resampling methods, see Section 11.2.4.10.3.1, *Chrominance Resampling*.
- When the YUVCHROMARESAMPLING bit is set to 0x1, the video pipeline is in configuration 2, and the scaler comes before the CSC unit.
  - In case of YUV4:2:2 input data with 90-/270-degree rotation, the data is preprocessed by duplicating the missing chrominance samples.
  - In case of YUV4:2:2 input data with 0-/180-degree rotation, the Chroma upsampling is performed in the scaler unit.

Table 11-66 lists the input formats supported by the video pipelines.

The video pipeline is enabled by setting the DISPC_VIDp_ATTRIBUTES[0] ENABLE bit to 0x1.

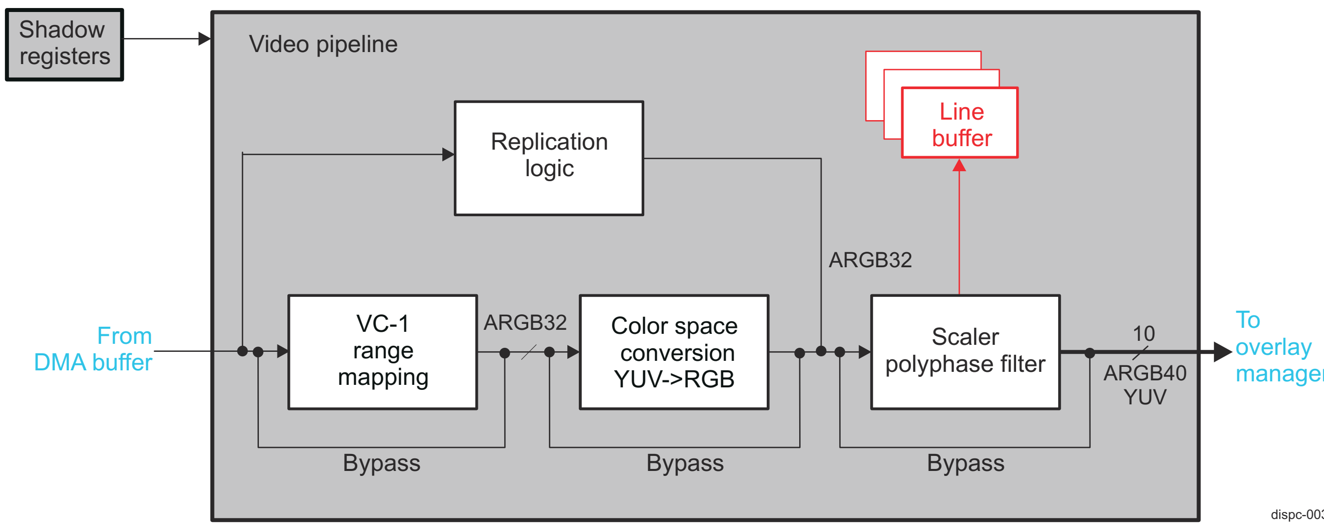


Figure 11-48. DISPC Configuration 1: Video Pipeline

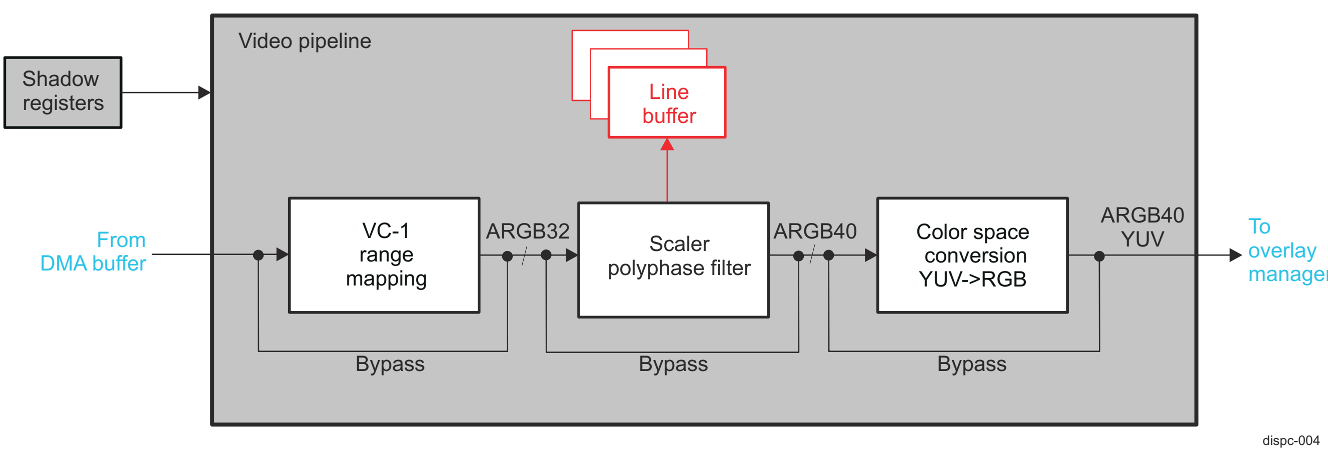


Figure 11-49. DISPC Configuration 2: Video Pipeline

11.2.4.10.1 DISPC Replication Logic

The replication logic increases the color depth of the video-encoded pixels (from true color RGB 12, and 16 to 32 bpp) available only in configuration 1.

The expansion from 8- to 10-bit color component is done by the following units:

- The CSC unit, when the scaler is after the CSC module and when the vertical scaler is disabled
- The vertical scaler, when the scaler is after the CSC unit and enabled
- The replication bit module, when the CSC unit and scaler are disabled
- When the replication logic is enabled by setting the DISPC_VIDp_ATTRIBUTES[10] REPLICATIONENABLE bit to 0x1, the MSBs are copied to the missing LSBs. Table 11-69 describes the remapping of the RGB pixels into ARGB 32-bit values.

Table 11-69. DISPC Replication Enabled: RGB Pixel Formats Remapping Into ARGB32-8888

	A[7:0]	R[7:0]	G[7:0]	B[7:0]
Formats	**MSB LSB**	**MSB LSB**	**MSB LSB**	**MSB LSB**
xRGB12-4444	11111111	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]
RGBx12-4444	11111111	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]
RGB16-565	11111111	R[4:0]R[4:2]	G[5:0]G[5:4]	B[4:0]B[4:2]
xRGB16-1555	11111111	R[4:0]R[4:2]	G[4:0]G[4:2]	B[4:0]B[4:2]
ARGB16-1555	AAAAAAAA	R[4:0]R[4:2]	G[4:0]G[4:2]	B[4:0]B[4:2]
ARGB16-4444	A[3:0]A[3:0]	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]
RGBA16-4444	A[3:0]A[3:0]	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]

- When the replication logic is disabled by setting the DISPC_VIDp_ATTRIBUTES[10] REPLICATIONENABLE bit to 0x0, the encoded pixel values are shifted to the MSB boundary of the 24-bit format. The missing bit values are filled with 0s. Table 11-70 describes the remapping of the RGB pixels into ARGB 32-bit values.

Table 11-70. DISPC Replication Disabled: RGB Pixel Formats Remapping Into ARGB32-8888

	A[7:0]	R[7:0]	G[7:0]	B[7:0]
Formats	**MSB LSB**	**MSB LSB**	**MSB LSB**	**MSB LSB**
xRGB12-4444	11111111	R[3:0]0000	G[3:0]0000	B[3:0]0000
RGBx12-4444	11111111	R[3:0]0000	G[3:0]0000	B[3:0]0000
RGB16-565	11111111	R[4:0]000	G[5:0]00	B[4:0]000
xRGB16-1555	11111111	R[4:0]000	G[4:0]000	B[4:0]000
ARGB16-1555	AAAAAAAA	R[4:0]000	G[4:0]000	B[4:0]000
ARGB16-4444	A[3:0]A[3:0]	R[3:0]0000	G[3:0]0000	B[3:0]0000
RGBA16-4444	A[3:0]A[3:0]	R[3:0]0000	G[3:0]0000	B[3:0]0000

11.2.4.10.2 DISPC VC-1 Range Mapping Unit

The VC-1 range mapping unit is used when the video frame picture is decoded using a VC-1 codec by the video accelerator. It remaps the Y, Cb, and Cr components. The unit is used primarily for YUV4:2:0-NV12 and YUV4:2:0-NV21 pixel formats but also can be applied to YUV4:2:2 pixel formats (YUV2 and UYVY).

The VC-1 range mapping unit is enabled by setting the DISPC_VIDp_ATTRIBUTES2[0] VC1ENABLE bit to 0x1. The DISPC_VIDp_ATTRIBUTES2[3:1] VC1_RANGE_Y and DISPC_VIDp_ATTRIBUTES2[6:4] VC1_RANGE_CBCR bit fields are two 3-bit values programmed by the user and are independent for each video pipeline. The module is governed by the equations:

$$Y_{out} = \mathrm{CLIP}((((\,Y_{int} - 128) \times (\mathrm{VC1\_RANGE\_Y} + 9) + 4) / 8) + 128)$$

$$C_b = \mathrm{CLIP}((((\,C_b - 128) \times (\mathrm{VC1\_RANGE\_CBCR} + 9) + 4) / 8) + 128)$$

$$C_r = \mathrm{CLIP}((((\,C_r - 128) \times (\mathrm{VC1\_RANGE\_CBCR} + 9) + 4) / 8) + 128)$$

Note

The input and output pixel values are unsigned (Y, Cr, and Cb).

The function CLIP () clips to 0 or 255 when minimum or maximum, respectively, are reached; otherwise, the resulting output remains identical.

11.2.4.10.3 DISPC CSC Unit YUV to RGB

The CSC unit converts the video-encoded pixel values from YUV4:4:4 format into RGB24 or RGB30 format. The output format depends on the video pipeline configuration selected:

- Configuration 1: RGB24 output format, with 8-bit value per component: red, green, and blue
- Configuration 2: RGB30 output format, with 10-bit value per component: red, green, and blue

In case of YUV4:2:0 or YUV4:2:2 formats, a chrominance resampling to YUV4:4:4 is required before converting the YUV into RGB values (see Section 11.2.4.10.3.1, *Chrominance Resampling*). YUV4:2:2 or YUV4:2:0 to YUV4:4:4 chrominance resampling is a preprocessing to the color space conversion.

Figure 11-50 through Figure 11-53 show the 3 × 3 11-bit coefficients used to convert from YUV4:4:4 into RGB24. The coefficients are set according to the standard used to encode the pixel data in YUV color space. Table 11-71 summarizes the coefficients with their respective bit fields.

Table 11-71. DISPC Color Space Conversion YUV to RGB Bit Field Setting

Coefficients	Bit Field Registers
R_Y	DISPC_VIDp_CONV_COEF0[10:0] RY
R_{Cr}	DISPC_VIDp_CONV_COEF0[26:16] RCR
R_{Cb}	DISPC_VIDp_CONV_COEF1[10:0] RCB
G_Y	DISPC_VIDp_CONV_COEF1[26:16] GY
G_{Cr}	DISPC_VIDp_CONV_COEF2[10:0] GCR
G_{Cb}	DISPC_VIDp_CONV_COEF2[26:16] GCB
B_Y	DISPC_VIDp_CONV_COEF3[10:0] BY
B_{Cr}	DISPC_VIDp_CONV_COEF3[26:16] BCR
B_{Cb}	DISPC_VIDp_CONV_COEF4[10:0] BCB

- For configuration 1 with an RGB24 output:

  If the active range for the luminance samples (Y) is [235:16] and [240:16] for the chrominance samples (Cb and Cr), the range selection is done by setting the DISPC_VIDp_ATTRIBUTES[11] FULLRANGE bit to 0x0. The values of R, G, and B output components are clipped to the range [255:0].

Note

The scaling and CSC clipping is set by the same bit, DISPC_VIDp_ATTRIBUTES[11] FULLRANGE.

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{cr} & R_{cb} \\ G_Y & G_{cr} & G_{cb} \\ B_Y & B_{cr} & B_{cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} - 16 \\ Cr_{IN} - 128 \\ Cb_{IN} - 128 \end{bmatrix}$$

dispc-005

Figure 11-50. DISPC YCbCr to RGB Registers (FULLRANGE = 0), 8-Bit Outputs

If the active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [255:0], the range selection is done by setting the DISPC_VIDp_ATTRIBUTES[11] FULLRANGE bit to 0x1. The values of R, G, and B output components are clipped to the range [255:0].

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{cr} & R_{cb} \\ G_Y & G_{cr} & G_{cb} \\ B_Y & B_{cr} & B_{cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} \\ Cr_{IN} - 128 \\ Cb_{IN} - 128 \end{bmatrix}$$

dispc-006

Figure 11-51. DISPC YCbCr to RGB Registers (FULLRANGE = 1), 8-Bit Outputs

- For configuration 2 with an RGB30 output:

  If the active range for the luminance samples (Y) is [940:64] and [960:64] for the chrominance samples (Cb and Cr), the range selection is done by setting the DISPC_VIDp_ATTRIBUTES[11] FULLRANGE bit to 0x0. The values of R, G, and B output components are clipped to the range [1023:0].

Note

The scaling and CSC clipping is set by the same bit, DISPC_VIDp_ATTRIBUTES[11] FULLRANGE.

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{cr} & R_{cb} \\ G_Y & G_{cr} & G_{cb} \\ B_Y & B_{cr} & B_{cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} - 64 \\ Cr_{IN} - 512 \\ Cb_{IN} - 512 \end{bmatrix}$$

dispc-074

Figure 11-52. DISPC YCbCr to RGB Registers (FULLRANGE = 0), 10-Bit Outputs

If the active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [1023:0], the range selection is done by setting the DISPC_VIDp_ATTRIBUTES[11] FULLRANGE bit to 0x1. The values of R, G, and B output components are clipped to the range [1023:0].

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{cr} & R_{cb} \\ G_Y & G_{cr} & G_{cb} \\ B_Y & B_{cr} & B_{cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} \\ Cr_{IN} - 512 \\ Cb_{IN} - 512 \end{bmatrix}$$

dispc-075

Figure 11-53. DISPC YCbCr to RGB Registers (FULLRANGE = 1), 10-Bit Outputs

11.2.4.10.3.1 DISPC Chrominance Resampling

Two methods are supported to resample chrominance:

- Averaging of the chrominance is done by software, followed by hardware conversion when the video pipeline is in configuration 1.
- Filtering of the chrominance using the scaler unit (chrominance resampling and rescaling can be combined to support native rescaling of YUV format) when the video pipeline is in configuration 2.

To convert the YUV4:2:2 encoded pixel values into YUV4:4:4 format, the averaging of the chrominance technique can be used as shown in Figure 11-54. The missing chrominance samples (Cb and Cr) are interpolated using the average values of the two closest values on the same line (,) or are repeated from the second pixel in the same 32-bit container (see Figure 11-55). For the last pixel, the chrominance samples are duplicated using the values from the previous pixel; otherwise, the chrominance samples are averaged using the two adjacent values. Figure 11-56 shows the flow of the pixel.

$$Cb_n(YCbCr\ 444) = \frac{Cb_{n-1}(YCbCr\ 422) + Cb_{n+1}(YCbCr\ 422)}{2}\ (\text{n odd})$$

$$Cr_n(YCbCr\ 444) = \frac{Cr_{n-1}(YCbCr\ 422) + Cr_{n+1}(YCbCr\ 422)}{2}\ (\text{n odd})$$

dispc-010

Figure 11-54. DISPC Averaging of the Chrominance Formula

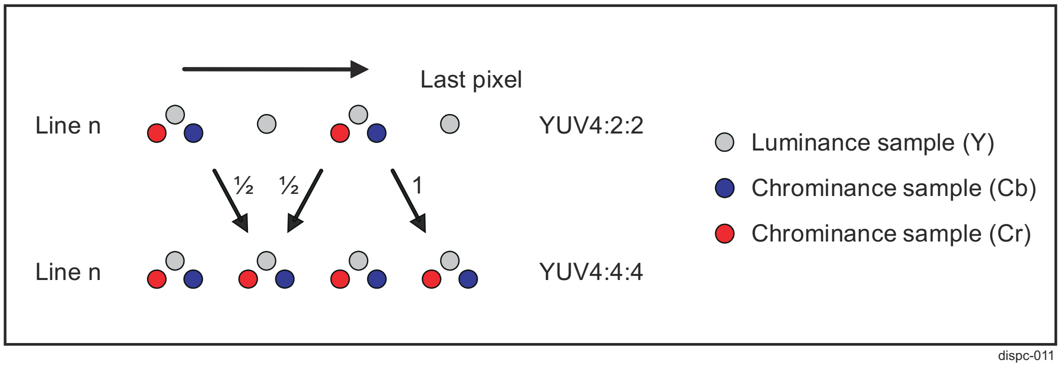


Figure 11-55. DISPC Averaging of the Chrominance Representation

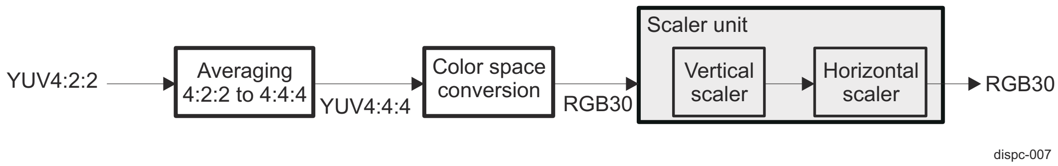


Figure 11-56. DISPC YUV4:2:2 to RGB30 Using Averaging of the Chrominance

The scaler unit can be used to resample the chrominance of YUV4:2:0 and YUV4:2:2, as shown in Figure 11-57 and Figure 11-58, respectively. The settings of the scaler unit to perform chrominance resampling are described in Section 11.2.4.10.4, *DISPC Scaler Unit*.

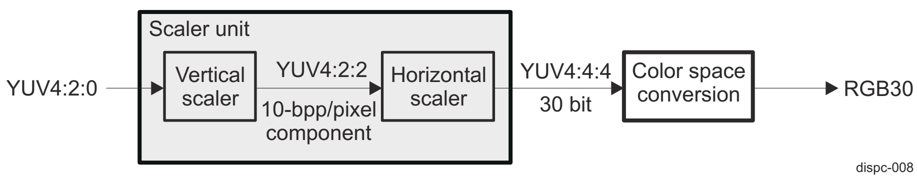


Figure 11-57. DISPC YUV4:2:0 to RGB30 Using Scaler Unit for Resampling Chrominance

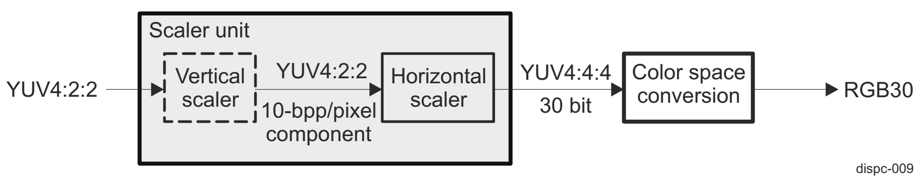


Figure 11-58. DISPC YUV4:2:2 to RGB30 Using Scaler Unit for Resampling Chrominance

Note

If rotation must be supported, YUV4:2:2 and YUV4:2:0 (0-/180-degree rotation) chrominance resampling is done as shown in Figure 11-58 and Figure 11-57, respectively.
For YUV4:2:2 (90-/270-degree rotation) data are preprocessed to present YUV4:4:4 on the scaler input (duplication of the missing chroma), as shown in Figure 11-56.

11.2.4.10.4 DISPC Scaler Unit

All video formats are supported, including formats with alpha blending. Alpha blending is scaled with the same parameters as RGB color components. For the YUV formats, Y and Cb/Cr are processed independently. The filter is based on a finite impulse response (FIR) filter. The filtering can be used for different processing:

- Upsampling of the picture
- Downsampling of the picture

- Antiflicker reduction
- Spatial de-interlacing using bob algorithm
- Chrominance resampling in case of YUV data formats

Note

The user must ensure that the resizing frame displays in the LCD/screen boundaries.

Figure 11-59 shows an example of video upsampling.

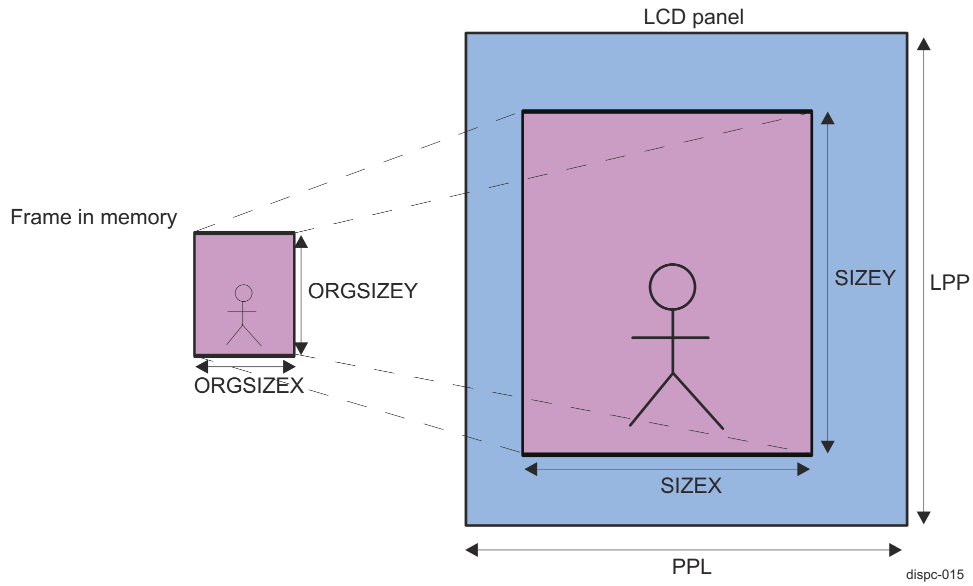


Figure 11-59. DISPC Video Upsampling

The upsampling and downsampling filter is a polyphase filter with five taps and eight phases for the horizontal filter, and a programmable number of taps (three or five) and eight phases for vertical filter. The input buffer has five input memory lines. The following limitations must be considered:

- The upsampling ratio is up to x8.
- The downsampling ratio using 3-tap configuration is down to x0.5 for RGB format.
- The downsampling ratio using 5-tap configuration is down to x0.25 for RGB format.
- If the input format is changed from YUV4:2:2 to YUV4:2:0 (WB pipeline), the downsampling ratio is further reduced:
  - Using 5-tap configuration, the ratio is down to x0.5 for RGB format.
  - Using 3-tap configuration, no downscaling is available.

For vertical upsampling and downsampling in a 3-tap configuration, the equations are:

For RGB formats

$$A_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi)*A_{in}(n+i)\right)>>5$$

$$R_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi)*R_{in}(n+i)\right)>>5$$

$$G_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi)*G_{in}(n+i)\right)>>5$$

$$B_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi)*B_{in}(n+i)\right)>>5$$

For YUV formats

$$Y_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi_y)*Y_{in}(n+i)\right)>>5$$

$$Cr_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vci}(\Phi_c)*Cb_{in}(n+i)\right)>>5$$

$$Cb_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vci}(\Phi_c)*Cr_{in}(n+i)\right)>>5$$

dispc-013

(1)

For vertical upsampling and downsampling in a 5-tap configuration, the equations are:

For RGB formats

$$A_{out}(n)=(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*A_{in}(n+i))>>5$$
$$R_{out}(n)=(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*R_{in}(n+i))>>5$$
$$G_{out}(n)=(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*G_{in}(n+i))>>5$$
$$B_{out}(n)=(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*B_{in}(n+i))>>5$$

For YUV formats

$$Y_{out}(n)=(\sum_{i=-2}^{i=2} C_{vi}(\Phi_y)*Y_{in}(n+i))>>5$$
$$Cb_{out}(n)=(\sum_{i=-2}^{i=2} C_{vci}(\Phi_c)*Cb_{in}(n+i))>>5$$
$$Cr_{out}(n)=(\sum_{i=-2}^{i=2} C_{vci}(\Phi_c)*Cr_{in}(n+i))>>5$$

dispc-012

(2)

For horizontal upsampling and downsampling in a 5-tap configuration, the equations are:

For RGB formats

$$A_{out}(n)=(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*A_{in}(n+i))>>7$$
$$R_{out}(n)=(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*R_{in}(n+i))>>7$$
$$G_{out}(n)=(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*G_{in}(n+i))>>7$$
$$B_{out}(n)=(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*B_{in}(n+i))>>7$$

For YUV formats

$$Y_{out}(n)=(\sum_{i=-2}^{i=2} C_{hi}(\Phi_y)*Y_{in}(n+i))>>7$$
$$Cb_{out}(n)=(\sum_{i=-2}^{i=2} C_{hci}(\Phi_c)*Cb_{in}(n+i))>>7$$
$$Cr_{out}(n)=(\sum_{i=-2}^{i=2} C_{hci}(\Phi_c)*Cr_{in}(n+i))>>7$$

dispc-014

(3)

Note

The pixel (n + 1) is the previous pixel with respect to pixel (n). The line (n + 1) is the previous line with respect to line (n).

The coefficients Ci() depend on the phase between input and output pixels.

Note

The coefficients are different for Y and Cr, Cb filtering because the calculations are independent due to the chrominance resampling for YUV4:2:2 and YUV4:2:0.

First, the vertical filter is applied to the encoded input pixel data, and then the horizontal filter is applied on the resulting pixel values to generate the output pixel values. The vertical input of the filter consists of five lines of 2048 × 32 bits for both 3-tap and 5-tap configurations (see Table 11-72).

Table 11-72. DISPC Line Buffer Width for Scaler Unit

Vertical Taps	Maximum Input Width (Pixels)
3, 5	2048 × 32 bits

At the beginning of frame scaling processing, the first line is duplicated to fill the first two lines in 3-tap configuration and the first three lines in 5-tap configuration.

At the end of frame scaling processing, the last line is duplicated if the scaling logic requires loading more lines and the last line has been reached.

The programmable coefficients of the polyphase filters are signed 8-bit values (except for the central coefficient, which is unsigned). The video scalers have an 8-bit input and a 10-bit output. The vertical scaling changes the 8-bit input into a 10-bit clipped output and the horizontal scaling takes the 10-bit input.

Figure 11-60 and Figure 11-61 show the scaler macro-architecture for the component A, R, G, B, and Y. Figure 11-62 and Figure 11-63 show the scaler macro-architecture for component Cr and Cb.

The scaling output can be clipped to an output range of [1023:0] or [960:64] by configuring the DISPC_VIDp_ATTRIBUTES[11] FULLRANGE bit.

Note

The scaling and CSC clipping is set by the same bit, DISPC_VIDp_ATTRIBUTES[11] FULLRANGE.

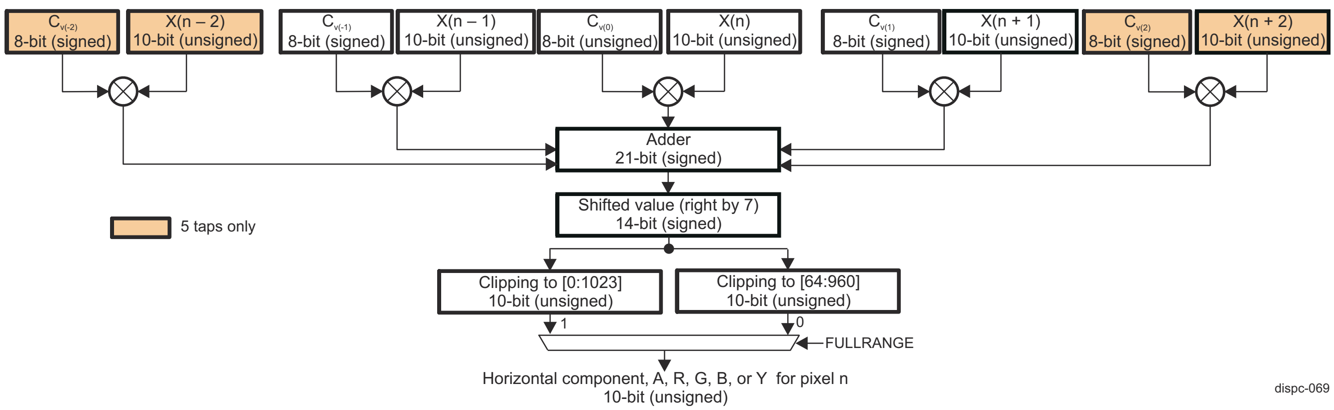


Figure 11-60. DISPC Macro-Architecture of the Horizontal Scaling for A, R, G, B, and Y Components (5-tap Restriction)

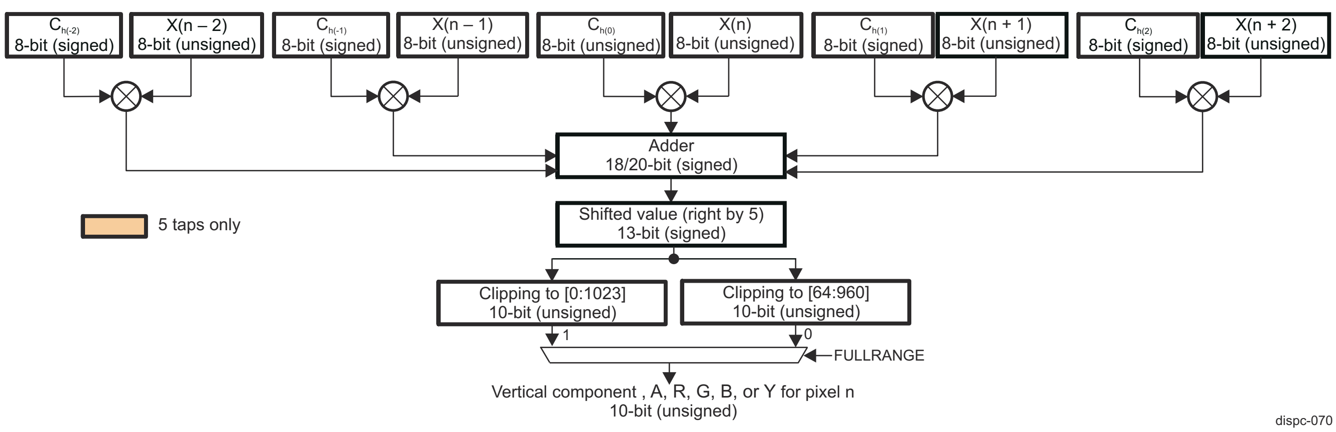


Figure 11-61. DISPC Macro-Architecture of the Vertical Scaling for A, R, G, B, and Y Components (5 and 3 taps)

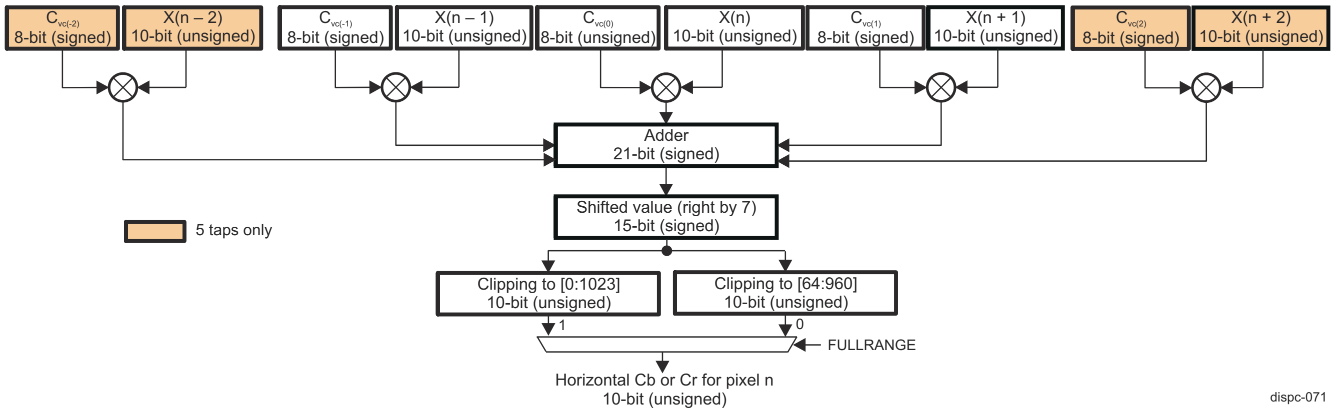


Figure 11-62. DISPC Macro-Architecture of the Horizontal Scaling for Cr and Cb Components (5-tap Restriction)

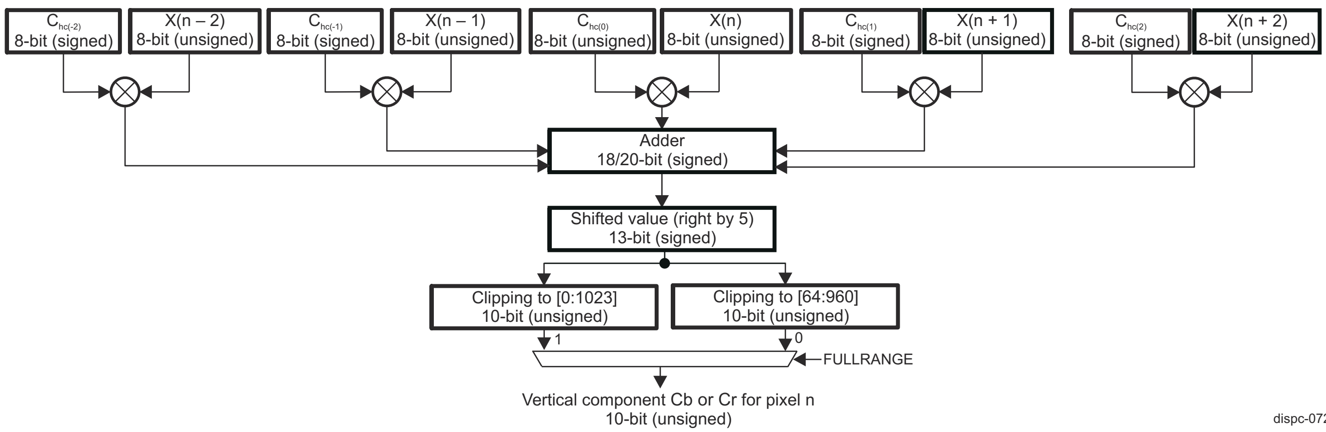


Figure 11-63. DISPC Macro-Architecture of the Vertical Scaling for Cr and Cb Components (5 and 3 taps)

Table 11-73 and Table 11-74 list the bit fields in the function to set for each coefficient.

Table 11-73. DISPC Register Bit Field Associated to Coefficient for ARGB and Y Configuration in VIDp Scaler

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cv(–2)		FIRVC22	DISPC_VIDp_FIR_COEF_V_i
	Cv(–1)	FIRVC2	FIRVC2	DISPC_VIDp_FIR_COEF_HV_i
	Cv(0)	FIRVC1	FIRVC1	DISPC_VIDp_FIR_COEF_HV_i
	Cv(1)	FIRVC0	FIRVC0	DISPC_VIDp_FIR_COEF_HV_i
	Cv(2)		FIRVC00	DISPC_VIDp_FIR_COEF_V_i

Table 11-73. DISPC Register Bit Field Associated to Coefficient for ARGB and Y Configuration in VIDp Scaler (continued)

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Horizontal	Ch(–2)	N/A	FIRHC4	DISPC_VIDp_FIR_COEF_HV_i
	Ch(–1)		FIRHC3	DISPC_VIDp_FIR_COEF_H_i
	Ch(0)		FIRHC2	DISPC_VIDp_FIR_COEF_H_i
	Ch(1)		FIRHC1	DISPC_VIDp_FIR_COEF_H_i
	Ch(2)		FIRHC0	DISPC_VIDp_FIR_COEF_H_i

Table 11-74. DISPC Register Bit Field Associated to Coefficient for Cb and Cr Configuration in VIDp Scaler

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cvc(–2)		FIRVC22	DISPC_VIDp_FIR_COEF_V2_i
	Cvc(–1)	FIRVC2	FIRVC2	DISPC_VIDp_FIR_COEF_HV2_i
	Cvc(0)	FIRVC1	FIRVC1	DISPC_VIDp_FIR_COEF_HV2_i
	Cvc(1)	FIRVC0	FIRVC0	DISPC_VIDp_FIR_COEF_HV2_i
	Cvc(2)		FIRVC00	DISPC_VIDp_FIR_COEF_V2_i
Horizontal	Chc(–2)	N/A	FIRHC4	DISPC_VIDp_FIR_COEF_HV2_i
	Chc(–1)		FIRHC3	DISPC_VIDp_FIR_COEF_H2_i
	Chc(0)		FIRHC2	DISPC_VIDp_FIR_COEF_H2_i
	Chc(1)		FIRHC1	DISPC_VIDp_FIR_COEF_H2_i
	Chc(2)		FIRHC0	DISPC_VIDp_FIR_COEF_H2_i

The VID scaler unit vertical or/and horizontal sampling is defined by setting/resetting the DISPC_VIDp_ATTRIBUTES[6:5] RESIZEENABLE bit field.

A set of configurations must be valid before enabling the video upsampling and downsampling block.

The following fields define the configuration of the video upsampling downsampling block for VIDp:

- Vertical upsampling and downsampling increments the value of the DISPC_VIDp_FIR[28:16] FIRVINC bit field. The unsigned integer value range is [4096:1]. Software calculates the value using the following equation:

$$FIRVINC = 1024 * \left(\frac{SIZEY}{MEMSIZEY}\right) \quad (4)$$

dispc-066

Note

- If the value of the DISPC_VIDp_FIR[28:16] FIRVINC bit field is greater than 4096, it is clipped to 4096.
- If the DISPC_VIDp_SIZE[27:16] SIZEY bit field equals 0x1, SIZEY is replaced by 0x2 in the previous equation.
- The values of the DISPC_VIDp_PICTURE_SIZE[27:16] MEMSIZEY and DISPC_VIDp_SIZE[27:16] SIZEY bit fields must be programmed with the value desired minus 1.

- Horizontal upsampling and downsampling increments the value of the DISPC_VIDp_FIR[12:0] FIRHINC bit field. The unsigned integer value range is [4096:1]. Software calculates the value using the following equation:

$$FIRHINC = 1024 * \left(\frac{SIZEX}{MEMSIZEX}\right) \quad (5)$$

dispc-067

Note

- If the value of the DISPC_VIDp_FIR[12:0] FIRHINC bit field is greater than 4096, it is clipped to 4096.
- If the DISPC_VIDp_SIZE[10:0] SIZEX bit field equals 1, SIZEX is replaced by 2 in the previous equation.
- The values of the DISPC_VIDp_PICTURE_SIZE[10:0] MEMSIZEX and DISPC_VIDp_SIZE[10:0] SIZEX bit fields must be programmed with the value desired minus 1.

- Vertical up/downsampling accumulator value DISPC_VIDp_ACCU_j[26:16] VERTICALACCU bit field: The signed integer value range is [–1024:1023]. The accumulator value indicates on which phase the vertical filtering starts. The register DISPC_VIDp_ACCU_0 is used for progressive output and for interlace output; the DISPC_VIDp_ACCU_0 and DISPC_VIDp_ACCU_1 registers are used. Similarly, DISPC_VIDp_ACCU2_0 and DISPC_VIDp_ACCU2_1 are used in progressive or interlace output to set the accumulator value of the Cb and Cr components when scaling YUV format.
- Vertical upsampling and downsampling line buffer configuration DISPC_VIDp_ATTRIBUTES[21] VERTICALTAPS bit: The default value at reset time is 0x0 (3-tap configuration is used). If the bit field is reset, the 3-tap configuration is used.
- Horizontal upsampling and downsampling accumulator value DISPC_VIDp_ACCU_j[10:0] HORIZONTALACCU bit field: The signed integer value range is [–1024:1023]. The accumulator value indicates on which phase the horizontal filtering starts. The register DISPC_VIDp_ACCU_0 is used for progressive output and for interlace output; the DISPC_VIDp_ACCU_0 and DISPC_VIDp_ACCU_1 registers are used. Similarly, DISPC_VIDp_ACCU2_0 and DISPC_VIDp_ACCU2_1 are used in progressive or interlace output to set the accumulator value of the Cb and Cr components when scaling YUV format.

Table 11-75 lists the DISPC vertical and horizonatal accumulator values and phases.

Table 11-75. DISPC Vertical and Horizontal Accumulator Phase

Accumulator Value	Phases f
0	0
128 or –896	1
256 or –768	2
384 or –640	3
512 or –512	4
640 or –384	5
768 or –256	6
896 or –128	7

- Vertical upsampling and downsampling coefficients:

– The 3-tap vertical upsampling and downsampling coefficients are defined in the DISPC_VIDp_FIR_COEF_HV_i registers. There are 8 registers for the 8 phases with 3 coefficients for each, or a total of 24 programmable coefficients for the vertical upsampling and downsampling block. Each register contains two 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one).
– The 5-tap vertical upsampling and downsampling coefficients: Two extra-tap vertical upsampling and downsampling coefficients are defined in the DISPC_VIDp_FIR_COEF_V_i registers. There are 8 registers for the 8 phases with 2 coefficients for each of them, so a total of 16 programmable coefficients for the vertical upsampling downsampling block are used in addition to the 3-tap registers previously defined.

Four YUV vertical upsampling and downsampling coefficients are set in DISPC_VIDp_FIR_COEF_HV2_i and DISPC_VIDp_FIR_COEF_V2_i registers. Table 11-73 and Table 11-74 summarize all coefficients and their respective registers.

- Horizontal upsampling and downsampling coefficients:
  – The DISPC_VIDp_FIR_COEF_H_i and DISPC_VIDp_FIR_COEF_HV_i registers define the 5-tap horizontal up/downsampling coefficients. Each DISPC_VIDp_FIR_COEF_H_i register contains three 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one). Each DISPC_VIDp_FIR_COEF_HV_i register contains one 8-bit signed coefficient. A total of 40 programmable coefficients for the horizontal upsampling and downsampling block are used.

Four YUV horizontal upsampling and downsampling coefficients are set in the DISPC_VIDp_FIR_COEF_HV2_i and DISPC_VIDp_FIR_COEF_H2_i registers. Table 11-73 and Table 11-74 summarize all coefficients and their respective registers.

11.2.4.10.4.1 DISPC Scaling Algorithms

Figure 11-64 and Figure 11-65 show details of the vertical and horizontal upsampling and downsampling finite state-machines (FSMs), respectively.

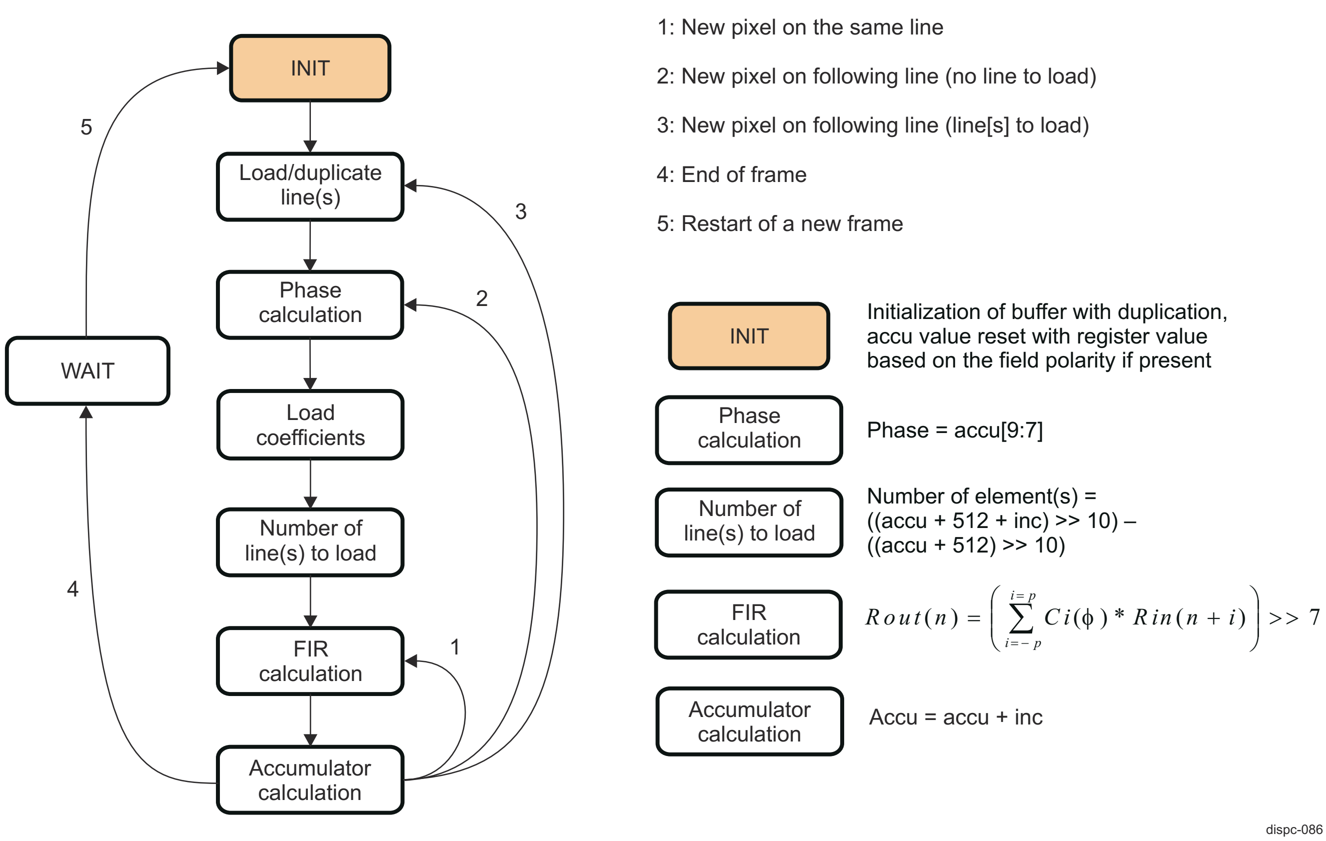


Figure 11-64. DISPC Vertical Upsampling and Downsampling Algorithm

Figure 11-65 shows the horizontal up/downsampling FSM.

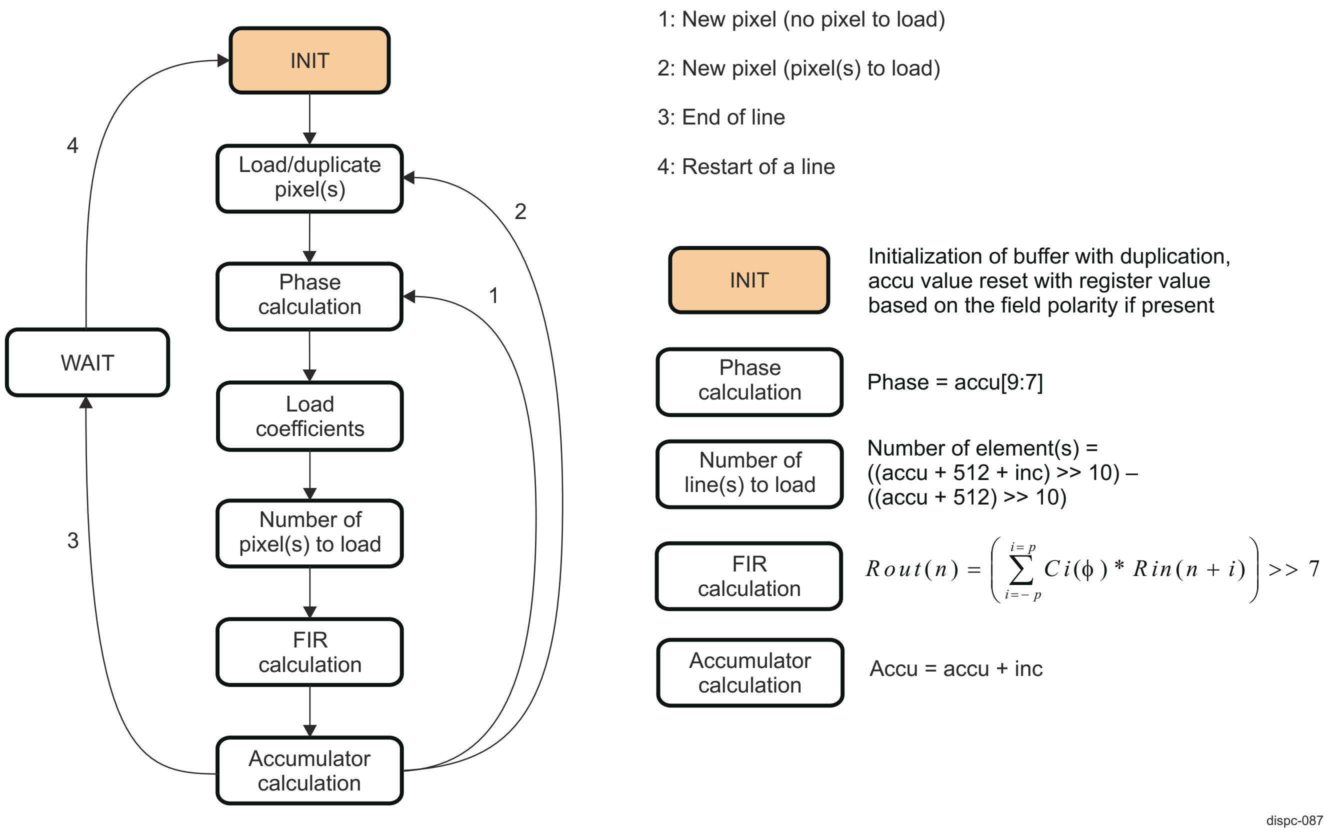


Figure 11-65. DISPC Horizontal Up/Downsampling Algorithm

11.2.4.10.4.2 DISPC Scaling limitations

Table 11-76 lists the minimum ratio between the pixel clock frequency (DSS_DISPC_LCDn_PCLK) and the functional clock (F_CLK) in the input pixel format when using the scaler unit. DSS_DISPC_LCDn_PCLK and F_CLK are asynchronous. For each LCD output, a dedicated LCD clock is programmable with the LCD and PCD divisor values in DISPC_DIVISORo[23:16][7:0].

Note

The F_CLK is derived from DSS_FCLK through the LCD divisor.

Note

The downscaling ratio is not an integer, it is the ratio F_CLK / DSS_DISPC_LCDn_PCLK, meaning if the ratio is 2.7, then the downscaling ratio is 2.7 and not 2.

Table 11-76. DISPC Pixel Clock Frequency Limitations (Any Pixel Format) – Active Matrix Display

F_CLK/DSS_DISPC_LCDn_PCLK Minimum Ratio	Horizontal Resampling				
	Off	Up	1:1–1:2	1:2–1:3	1:3–1:4
	2 or 1(1)	2 or 1(1))	2	3	4

(1) The minimum ratio can be 1 if the data are output on the rising edge of the PCLK (DISPC_POL_FREQo.IPC = 0 and CTRL_CORE_SMA_SW_1[]DSS_CHx_IPC); otherwise, the minimum ratio must be 2.

11.2.4.11 DISPC Write-Back Pipeline

The write-back pipeline is used to store in the system memory the capture of the overlay output or the output of one of the pipelines. The WB pipeline consists of a CSC unit, a scaler unit, and an RGB truncation logic. Because the overlay works on ARGB32-8888 format and the video accelerator works on YUV format, the color space conversion from RGB to YUV is used to directly output to memory the format that can be encoded with no extra processing.

The write-back pipeline is connected to all the pipeline outputs (GFX, VD1, VID2, and VID3 pipelines) and to the output of the three overlay managers (LCD1, LCD2, LCD3, and TV). The input is selected by setting the DISPC_WB_ATTRIBUTES[18:16] CHANNELIN bit field, and the capture frame rate is set in the DISPC_WB_ATTRIBUTES[26:24] CAPTUREMODE bit field.

Because the output format of the TV overlay manager is ARGB40, the graphics pipeline output is ARGB40, the video pipeline outputs are YUV4:2:2, YUV4:2:0, or ARGB40, and the write-back input is ARGB32, the write-back input does not consider the 2 LSBs of each ARGB component.

The write-back pipeline is enabled by setting the DISPC_WB_ATTRIBUTES[0] ENABLE bit to 0x1.

Figure 11-66 shows the graphics pipeline.

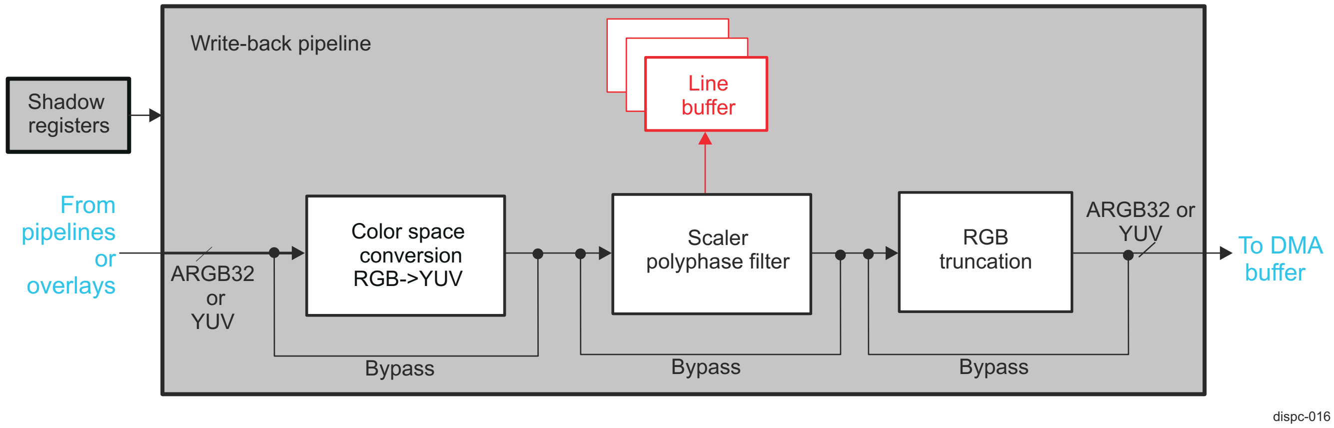


Figure 11-66. DISPC Write-Back Pipeline

11.2.4.11.1 DISPC Write-Back CSC Unit RGB to YUV

The RGB-to-YUV CSC unit converts the encoded pixel values from RGB24 into YUV4:4:4 format. For YUV4:2:0 or YUV4:2:2 formats, a chrominance sub-sampling is required after converting the RGB into YUV values. Because of the subsampling, the following limitations must be considered:

- When converting RGB into YUV4:2:0 NV12 format:
  - Maximum horizontal downscale = x0.5
  - Maximum vertical downscale = x0.5
- When converting RGB into YUV4:2:2 format:
  - Maximum horizontal downscale = x0.5
  - Maximum vertical downscale = x0.25

Figure 11-67 and Figure 11-68 show the 3 × 3 11-bit coefficients used to convert from RGB24 into YUV4:4:4. The user sets the coefficients according to the standard used to encode the pixel data in YUV color space. Table 11-77 lists the coefficients with their respective bit fields.

Table 11-77. DISPC CSC RGB to YUV Bit Field Setting

Coefficients	Bit Fields
Y_R	DISPC_WB_CONV_COEF0[10:0] YR
Y_G	DISPC_WB_CONV_COEF0[26:16] YG
Y_B	DISPC_WB_CONV_COEF1[10:0] YB

Table 11-77. DISPC CSC RGB to YUV Bit Field Setting (continued)

Coefficients	Bit Fields
Cr_R	DISPC_WB_CONV_COEF1[26:16] CRR
Cr_G	DISPC_WB_CONV_COEF2[10:0] CRG
Cr_B	DISPC_WB_CONV_COEF2[26:16] CRB
Cb_R	DISPC_WB_CONV_COEF3[10:0] CBR
Cb_G	DISPC_WB_CONV_COEF3[26:16] CBG
Cb_B	DISPC_WB_CONV_COEF4[10:0] CBB

If the active range for the luminance samples (Y) is [16:235] and [16:240] for the chrominance samples (Cb and Cr), the values of Y, Cb, and Cr output components are clipped to the range [0:255]. The range selection is done by setting the DISPC_WB_ATTRIBUTES[11] FULLRANGE bit to 0x0.

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} Y_R & Y_G & Y_B \\ Cb_R & Cb_G & Cb_B \\ Cr_R & Cr_G & Cr_B \end{bmatrix} * \begin{bmatrix} R_{IN} \\ G_{IN} \\ B_{IN} \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

dispc-017

Figure 11-67. DISPC RGB to YCbCr (FULLRANGE = 0)

If the active range for the luminance samples (Y) and or the chrominance samples (Cb and Cr)is [0:255], the values of Y, Cb, and Cr output components are clipped to the range [0:255]. The range selection is done by setting the DISPC_WB_ATTRIBUTES[11] FULLRANGE bit to 0x1.

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} Y_R & Y_G & Y_B \\ Cb_R & Cb_G & Cb_B \\ Cr_R & Cr_G & Cr_B \end{bmatrix} * \begin{bmatrix} R_{IN} \\ G_{IN} \\ B_{IN} \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

dispc-018

Figure 11-68. DISPC RGB to YCbCr (FULLRANGE = 1)

11.2.4.11.2 DISPC Write-Back Scaler Unit

The functional aspect of the write-back pipeline scaler unit is identical to the video pipeline scaler unit (see Section 11.2.4.10.4, *DISPC Scaler Unit*), except in the output width when scaling ARGB components. The resulting output format is ARGB32 instead of ARGB40.In addition, the scaling limitations described in Section 11.2.4.10.4.2 are relevant only to the video pipelines scaler units. In write-back memory-to-memory mode there are no limitations on the F_CLK/DSS_DISPC_LCDn_PCLK ratio for horizontal resampling.

The programmable coefficients of the polyphase filters are signed 8-bit values (except for the central coefficient, which is unsigned). The write-back scaler component has an 8-bit input and an 8-bit output.

Figure 11-69 and Figure 11-70 show the scaler macro-architecture for the component A, R, G, B, and Y. Figure 11-71 and Figure 11-72 show the scaler macro-architecture for component Cr and Cb.

The scaling output can be clipped to an output range of [0:255] or [16:240] by configuring the DISPC_WB_ATTRIBUTES[11] FULLRANGE bit.

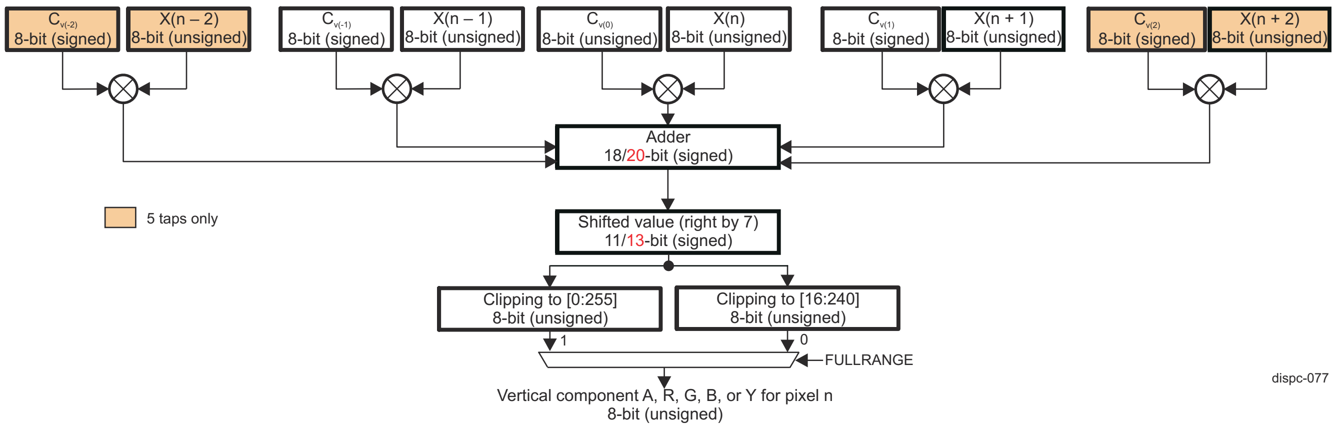


Figure 11-69. DISPC Macro-Architecture of the Vertical Scaling for A, R, G, B, and Y Components

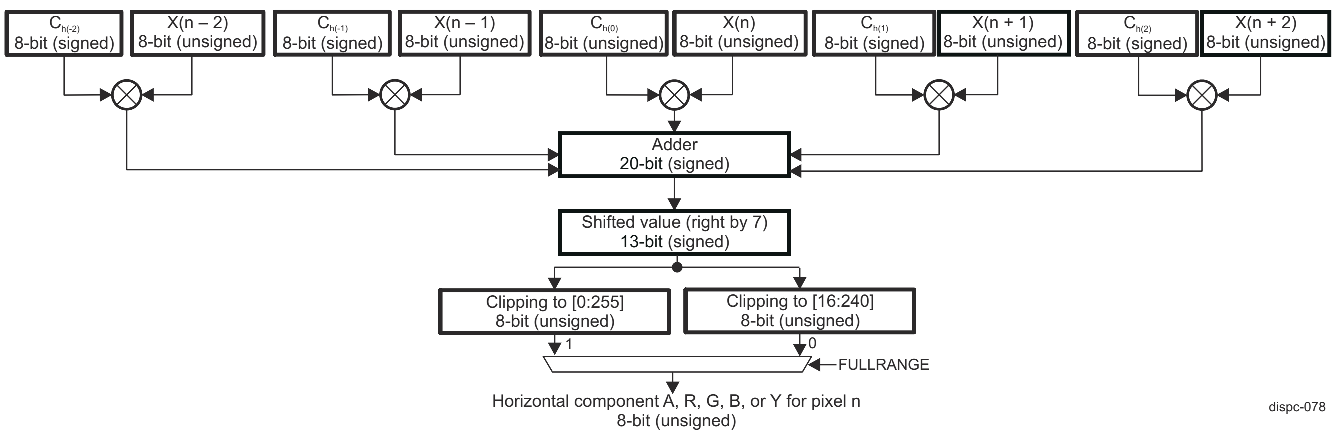


Figure 11-70. DISPC Macro-Architecture of the Horizontal Scaling for A, R, G, B, and Y Components

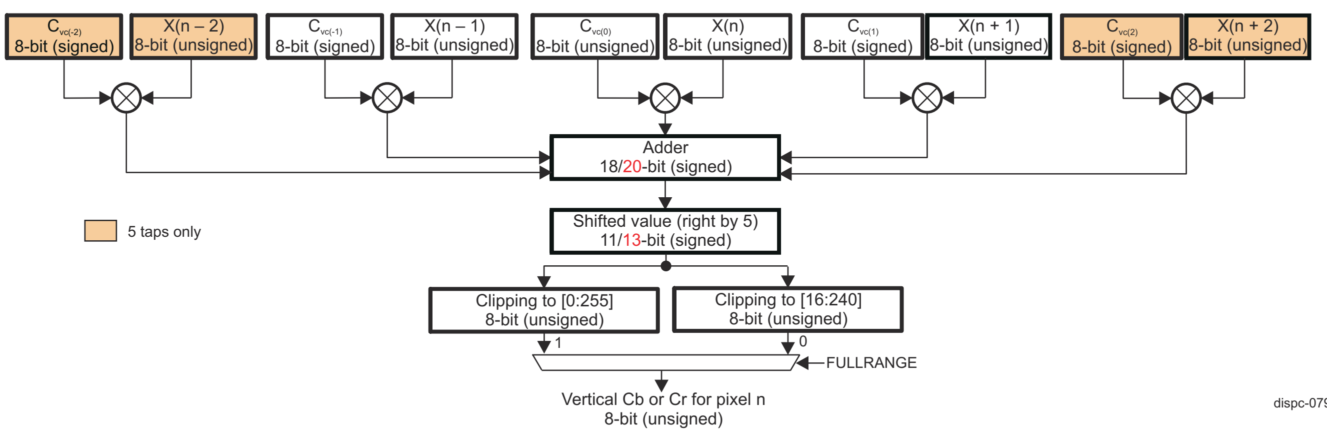


Figure 11-71. DISPC Macro-Architecture of the Vertical Scaling for Cr and Cb Components

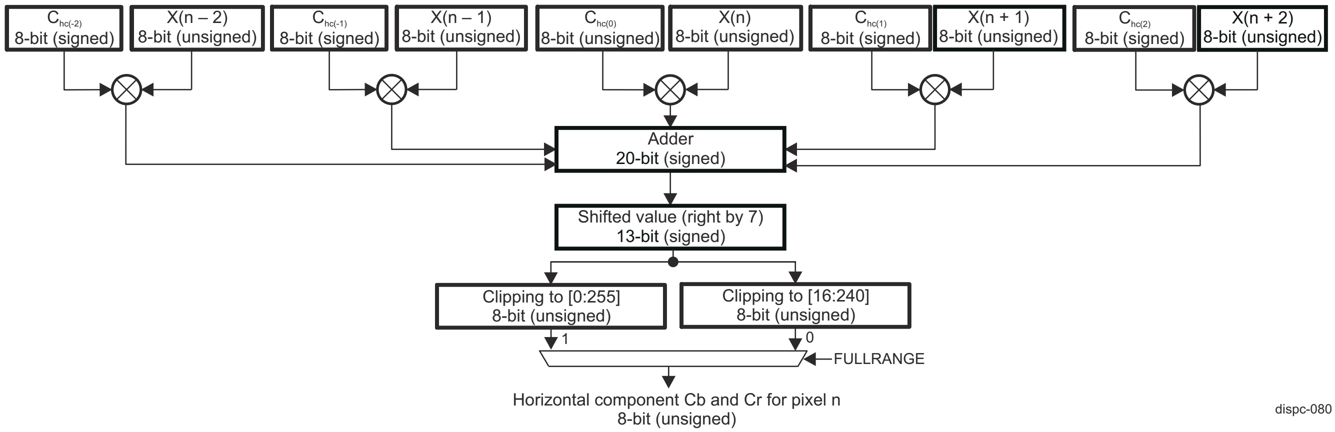


Figure 11-72. DISPC Macro-Architecture of the Horizontal Scaling for Cr and Cb Components

Table 11-78 and Table 11-79 list all the bit fields in the function to set each coefficient.

Table 11-78. DISPC Register Bit Field Associated With Coefficient for ARGB and Y Configuration in WB Scaler

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cv(–2)		FIRVC22	DISPC_WB_FIR_COEF_V_i
	Cv(–1)	FIRVC2	FIRVC2	DISPC_WB_FIR_COEF_HV_i
	Cv(0)	FIRVC1	FIRVC1	DISPC_WB_FIR_COEF_HV_i
	Cv(1)	FIRVC0	FIRVC0	DISPC_WB_FIR_COEF_HV_i
	Cv(2)		FIRVC00	DISPC_WB_FIR_COEF_V_i
Horizontal	Ch(–2)	N/A	FIRHC4	DISPC_WB_FIR_COEF_HV_i
	Ch(–1)		FIRHC3	DISPC_WB_FIR_COEF_H_i
	Ch(0)		FIRHC2	DISPC_WB_FIR_COEF_H_i
	Ch(1)		FIRHC1	DISPC_WB_FIR_COEF_H_i
	Ch(2)		FIRHC0	DISPC_WB_FIR_COEF_H_i

Table 11-79. DISPC Register Bit Field Associated With Coefficient for Cb and Cr Configuration in WB Scaler

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cvc(–2)		FIRVC22	DISPC_WB_FIR_COEF_V2_i
	Cvc(–1)	FIRVC2	FIRVC2	DISPC_WB_FIR_COEF_HV2_i
	Cvc(0)	FIRVC1	FIRVC1	DISPC_WB_FIR_COEF_HV2_i
	Cvc(1)	FIRVC0	FIRVC0	DISPC_WB_FIR_COEF_HV2_i
	Cvc(2)		FIRVC00	DISPC_WB_FIR_COEF_V2_i
Horizontal	Chc(–2)	N/A	FIRHC4	DISPC_WB_FIR_COEF_HV2_i
	Chc(–1)		FIRHC3	DISPC_WB_FIR_COEF_H2_i
	Chc(0)		FIRHC2	DISPC_WB_FIR_COEF_H2_i
	Chc(1)		FIRHC1	DISPC_WB_FIR_COEF_H2_i
	Chc(2)		FIRHC0	DISPC_WB_FIR_COEF_H2_i

The WB scaler unit vertical or/and horizontal sampling is defined by setting/resetting the DISPC_WB_ATTRIBUTES[6:5] RESIZEENABLE bit field.

A set of configuration must be valid before enabling the video up/downsampling block.

The following fields define the configuration of the video up/downsampling block for WB:

- Vertical up/downsampling increments the value of the DISPC_WB_FIR[28:16] FIRVINC bit field. The unsigned integer value range is [1:4096]. Software calculates the value using the following equation:

$$FIRVINC = 1024 * \left(\frac{SIZEY}{MEMSIZEY}\right) \quad (6)$$

dispc-066

Note

– If the value of the DISPC_WB_FIR[28:16] FIRVINC bit field is greater than 4096, it is clipped to 4096.
– If the DISPC_WB_SIZE[27:16] SIZEY bit field equals 0x1, SIZEY is replaced by 0x2 in the previous equation.
– The values of the DISPC_WB_PICTURE_SIZE[27:16] MEMSIZEY and DISPC_WB_SIZE[27:16] SIZEY bit fields must be programmed with the value desired minus 1.

- Horizontal up/downsampling increments the value of the DISPC_WB_FIR[12:0] FIRHINC bit field: The unsigned integer value range is [1:4096]. Software calculates the value using the following equation:

$$FIRHINC = 1024 * \left(\frac{SIZEX}{MEMSIZEX}\right) \quad (7)$$

dispc-067

Note

– If the value of the DISPC_WB_FIR[12:0] FIRHINC bit field is greater than 4096, it is clipped to 4096.
– If the DISPC_WB_SIZE[10:0] SIZEX bit field equals 1, the DISPC_WB_SIZE[10:0] SIZEX bit field is replaced by 2 in the previous equation.
– The value of the DISPC_WB_PICTURE_SIZE[10:0] MEMSIZEX and DISPC_WB_SIZE[10:0] SIZEX bit fields must be programmed with the value desired minus 1.

- Vertical up/downsampling accumulator value DISPC_WB_ACCU_j[26:16] VERTICALACCU bit field: The signed integer value range is [–1024:1023]. The accumulator value indicates on which phase the vertical filtering starts. The register DISPC_WB_ACCU_0 is used for progressive output and for interlace output the DISPC_WB_ACCU_0 and DISPC_WB_ACCU_1 registers are used. Similarly, DISPC_WB_ACCU2_0 and DISPC_WB_ACCU2_1 are used in progressive or interlace output to set the accumulator value of the Cb and Cr components when scaling YUV format.
- Vertical up/downsampling line buffer configuration DISPC_WB_ATTRIBUTES[21] VERTICALTAPS bit: The default value at reset time is 0x0 (3-tap configuration is used). If the bit field is reset, the 3-tap configuration is used.
- Horizontal up/downsampling accumulator value DISPC_WB_ACCU_j[10:0] HORIZONTALACCU bit field: The signed integer value range is [–1024:1023]. The accumulator value indicates on which phase the horizontal filtering starts. The register DISPC_WB_ACCU_0 is used for progressive output and for interlace output the DISPC_WB_ACCU_0 and DISPC_WB_ACCU_1 registers are used. Similarly, DISPC_WB_ACCU2_0 and DISPC_WB_ACCU2_1 are used in progressive or interlace output to set the accumulator value of Cb and Cr components when scaling YUV format.

Table 11-80 lists the DISPC vertical and horizontal accumulator values and phases.

Table 11-80. DISPC Vertical/Horizontal Accumulator Phase

Accumulator Value	Phases f
0	0
128 or –896	1
256 or –768	2
384 or –640	3
512 or –512	4
640 or –384	5
768 or –256	6
896 or –128	7

- Vertical up/downsampling coefficients:
  – The 3-tap vertical up/downsampling coefficients are defined in the DISPC_WB_FIR_COEF_HV_i registers. There are 8 registers for the 8 phases with 3 coefficients for each, or a total of 24 programmable coefficients for the vertical up/downsampling block. Each register contains two 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one).
  – The 5-tap vertical up/downsampling coefficients: Two extra-tap vertical up/downsampling coefficients are defined in the DISPC_WB_FIR_COEF_V_i registers. There are 8 registers for the 8 phases with 2 coefficients for each of them, so a total of 16 programmable coefficients for the vertical up/downsampling block are used in addition to the 3-tap registers previously defined.

  Four YUV vertical up/downsampling coefficients are set in the DISPC_WB_FIR_COEF_HV2_i and DISPC_WB_FIR_COEF_V2_i registers. Table 11-78 and Table 11-79 summarize all coefficients and their respective registers.
- Horizontal up/downsampling coefficients:
  – The DISPC_WB_FIR_COEF_H_i and DISPC_WB_FIR_COEF_HV_i registers define the 5-tap horizontal up/downsampling coefficients. Each DISPC_WB_FIR_COEF_H_i register contains three 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one). Each DISPC_WB_FIR_COEF_HV_i

register contains one 8-bit signed coefficient. A total of 40 programmable coefficients for the horizontal up/downsampling block are used.

Four YUV horizontal up/downsampling coefficients are set in the DISPC_WB_FIR_COEF_HV2_i and DISPC_WB_FIR_COEF_H2_i registers. Table 11-78 and Table 11-79 summarize all coefficients and their respective registers.

11.2.4.11.3 DISPC Write-Back RGB Truncation Logic

Truncation logic is used to convert a pixel from ARGB 32-bit format into a lower color depth: 12- or 16-bit format based. Setting the DISPC_WB_ATTRIBUTES[10] TRUNCATIONENABLE bit to 0x1 enables the truncation to the pixel format defined by the DISPC_WB_ATTRIBUTES[4:1] FORMAT bit field. The truncation is done by removing the necessary LSB of each component to match the output format. Table 11-81 describes the truncation done on each component of the pixel.

Table 11-81. DISPC Truncation Logic

	A[7:0]	R[7:0]	G[7:0]	B[7:0]
Output Formats	**MSB LSB**	**MSB LSB**	**MSB LSB**	**MSB LSB**
xRGB12-4444	Ignored	R[7:4]	G[7:4]	B[7:4]
RGBx12-4444	Ignored	R[7:4]	G[7:4]	B[7:4]
RGB16-565	Ignored	R[7:3]	G[7:2]	B[7:3]
xRGB16-1555	Ignored	R[7:3]	G[7:3]	B[7:3]
ARGB16-4444	A[7:4]	R[7:4]	G[7:4]	B[7:4]
RGBA16-4444	A[7:4]	R[7:4]	G[7:4]	B[7:4]
ARGB16-1555	A[7]	R[7:3]	G[7:3]	B[7:3]

Note

If there is no alpha field in the pixel format description, 0s or 1s must fill the container. 0s must be used for transparent and 1s for opaque. For example, in xRGB12 pixel format, the upper 4 bits are set to 0s because the RGB value is only 12 bits inside a 16-bit container.

11.2.4.12 DISPC Hardware Cursor

The video layer or graphics layer can be used to display the hardware cursor. The encoded pixel data for the cursor image are in one of the following formats if the source transparency color key is used:

- xRGB12-4444
- ARGB16-4444
- RGBx12-4444
- RGBA16-4444
- ARGB16-1555
- RGB16-565
- RGB24-888
- xRGB24-88888
- RGBA32-8888
- BGRA32-8888
- ARGB32-8888

Otherwise, any pixel format can be used for the cursor image, considering the limitation in terms of pixels supported by the pipeline used to display the cursor image. To display a nonrectangle cursor, the transparency color key or alpha blending (pixel alpha blending) can be used (see Section 11.2.4.13.1.3, *DISPC Transparency Color Keys*). Global alpha blending can be used in addition to the transparency source color key to create a fading effect when the cursor (with a nonrectangle shape) appears and disappears on the screen. The image of the cursor can be stored entirely inside the DMA buffer to use the self-refresh mode (see Section 11.2.4.6.8.2, *DISPC DMA Ultralow-Power Mode*). In that case, the image is loaded once and then displayed without accessing the system memory for loading the image for each frame. This saves bandwidth on interconnect and memory.

11.2.4.13 DISPC LCD Outputs

The LCD1, LCD2, LCD3 output paths consist of several processing blocks (see Figure 11-73):

- Overlay manager
- Gamma correction unit
- Color phase rotation (CPR) block, also used for RGB-to-YUV color space conversion (CSC)
- Active matrix dithering with TDM
- BT.656
- BT.1120
- Synchronization buffer
- Timing generator

The display subsystem supports active matrix technologie (monochrome and color modes):

- Active matrix displays: the configuration of colors depends on the color depth:
  - 24 bpp supports 16,777,216 colors.
  - 18 bpp supports 262,144 colors.
  - 16 bpp supports 65,536 colors.
  - 12 bpp supports 4096 colors.

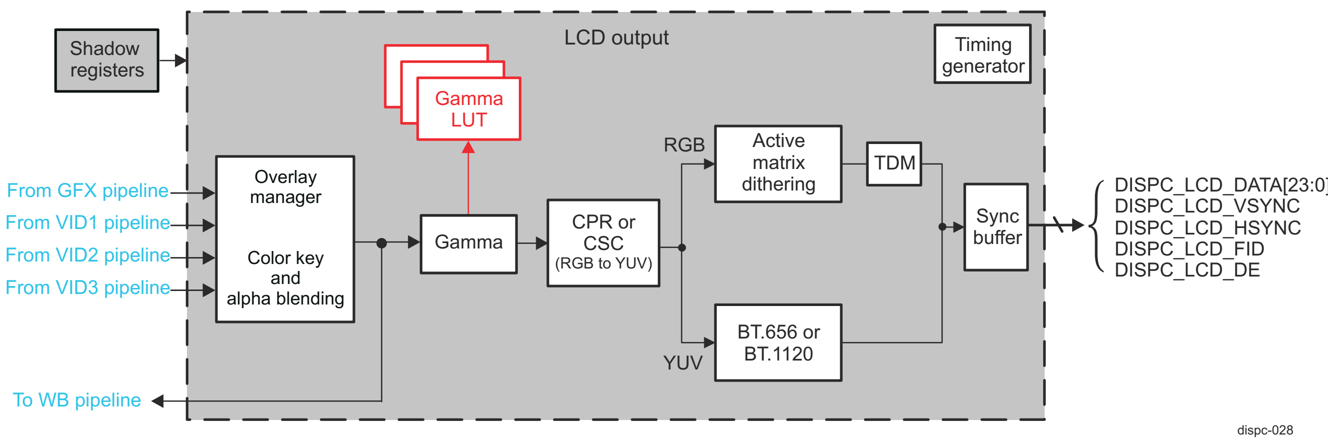


Figure 11-73. DISPC LCD Output Architecture

Note

In BT.656 mode only bits 9 through 0 are used from DISPC_LCD_DATA[23:0] data bus.

11.2.4.13.1 DISPC Overlay Manager

The overlay mechanism consists of displaying more than one layer (GFX, VID1, VID2, and VID3 layers) using:

- A priority rule based on a Z-order: Application can set the ordering layer of the frames.
- Transparency color keys: Destination and source transparency color keys can be set.
- Alpha blending values: Using the A component of a pixel or a blending set by the user for a layer, a level of transparency can be determined.

Each pipeline (GFX, VID1, VID2, and VID3) is assigned to a single overlay and, as a consequence, to a single display controller output, LCD1, LCD2, LCD3, TV, or WB pipeline. An overlay manager can be connected to all four pipelines outputs simultaneously. The pipeline output is directed using the DISPC_GFX_ATTRIBUTES[8] CHANNELOUT bit and the DISPC_VIDp_ATTRIBUTES[31:30] CHANNELOUT2 bit field. Table 11-82 summarizes the bit field settings to direct a pipeline to an LCD/TV or WB output. The default value directs all pipelines to LCD1.

Table 11-82. DISPC Pipeline Connection to LCD, TV, or WB Output

	DISPC_GFX_ATTRIBUTES/DISPC_VIDp_ATTRIBUTES	
Overlay Manager/Output	**CHANNELOUT Bit**	**CHANNELOUT2 Bit Field**
LCD1	0x0	0x0
LCD2	0x0	0x1
LCD3	0x0	0x2
WB	0x0	0x3
TV	0x1	0x0 (See the following note)

Note

When CHANNELOUT = 0x1, the settings CHANNELOUT2 = 0x0, 0x1, 0x2, and 0x3 are reserved.

The output of each LCD overlay manager is connected to CPR block through the gamma table unit in the case of gamma correction.

Note

When the pixel format is ARGB or RGBA, the color key match logic uses only the RGB value defined by ARGB or RGBA. The alpha blending factor is ignored.

11.2.4.13.1.1 DISPC Priority Rule

The overlay manager is configured using the Z-order parameter. The Z-order value defined for each pipeline indicates the visibility order to the window on the screen. If the Z-order value of window A is lower than the Z-order to layer B, layer A is displayed below layer B. The transparency color keys and the alpha blending factors are then used to blend the layers together (see Section 11.2.4.13.1.2, *DISPC ALPHA Blender*, and Section 11.2.4.13.1.3, *DISPC Transparency Color Keys*). The Z-order is enabled by setting the DISPC_GFX_ATTRIBUTES[25] ZORDERENABLE bit or the DISPC_VIDp_ATTRIBUTES[25] ZORDERENABLE bit to 0x1 and by defining the Z-order in the DISPC_GFX_ATTRIBUTES[27:26] and DISPC_VIDp_ATTRIBUTES[27:26] ZORDER bit fields. Table 11-83 summarizes the register settings to enable and set the Z-order of a pipeline. Table 11-83 shows the default Z-order values when LCDALPHABLENDERENABLE and ZORDERENABLE are disabled.

Table 11-83. DISPC Z-Order Register Settings and Default Configuration

Pipeline	LCDALPHA BLENDERENABLE[(1)]	ZORDERENABLE	ZORDER	Resulting Z-Order Number
GFX	0	0	Don't care	0
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	3
VID1	0	0	Don't care	1
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	0
VID2	0	0	Don't care	2
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	1

Table 11-83. DISPC Z-Order Register Settings and Default Configuration (continued)

Pipeline	LCDALPHA BLENDERENABLE[(1)]	ZORDERENABLE	ZORDER	Resulting Z-Order Number
VID3	0	0	Don't care	3
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	2

(1) Applies only to LCD1

Figure 11-74 shows the architecture of the priority rule.

Note

- If ZORDERENABLE = 1, each Z-order must be different for each active pipeline. It is not possible to use the same value for more than one pipeline.
- Two modes are maintained for backward compatibility with legacy devices:
  - LCDALPHABLENDERENABLE = 0 and ZORDERENABLE = 0 equivalent to the normal mode overlay settings
  - LCDALPHABLENDERENABLE = 1 equivalent to the alpha mode overlay settings

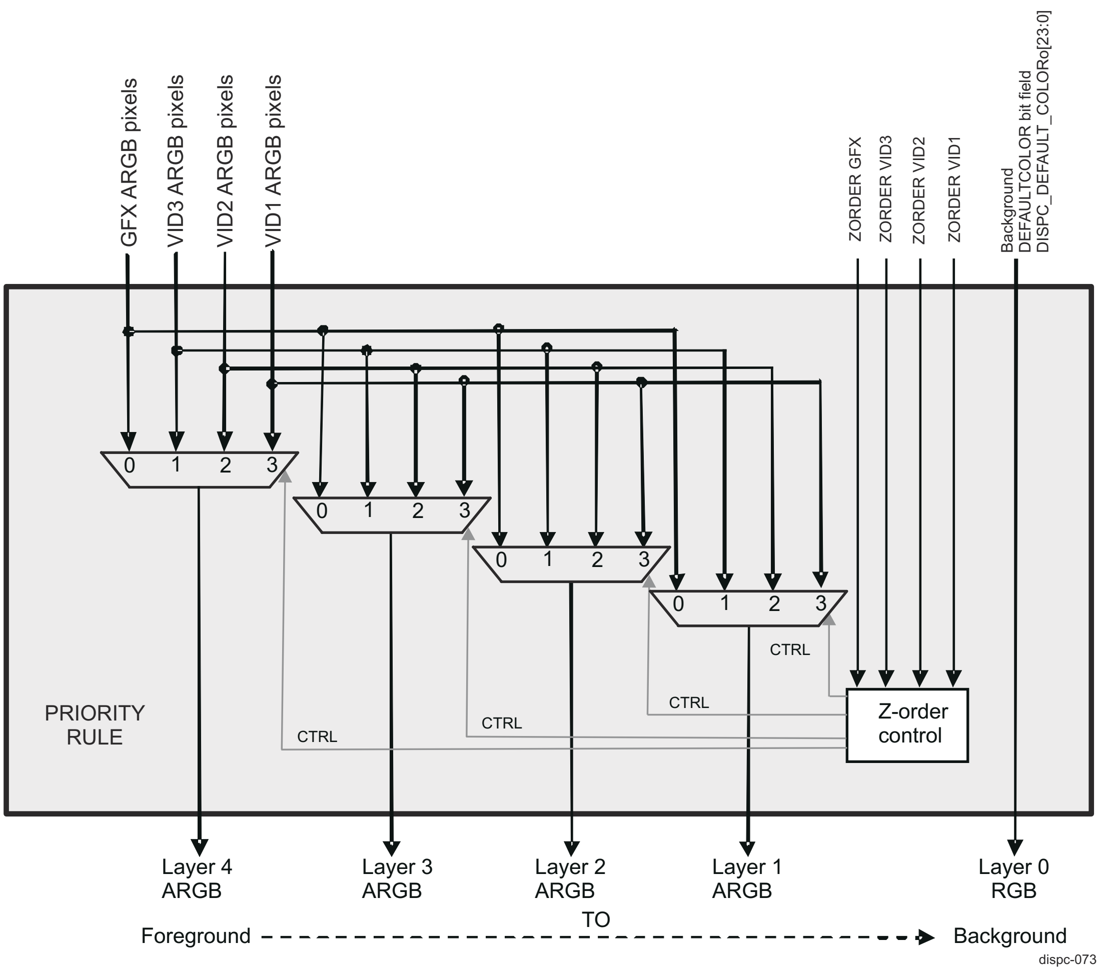


Figure 11-74. DISPC Priority Rule Architecture

The height and width of each enabled layer (pipeline) must be defined in the SIZEY and SIZEX bit fields DISPC_GFX_SIZE[27:16][10:0]/ DISPC_VIDp_SIZE[27:16][10:0], and its x and y positions defined in the POSX and POSY bit fields DISPC_GFX_POSITION[26:16][10:0]/DISPC_VIDp_POSITION[26:16][10:0]. If there are no graphics or video-encoded pixels at a specific position, the programmable, solid background color appears. The solid background color is set in the DISPC_DEFAULT_COLORo[23:0] DEFAULTCOLOR bit field. Figure 11-75 is an example of priority rule.

The Z-order reordering block must always map the pipelines to the blender logic in the same order—from background to foreground.

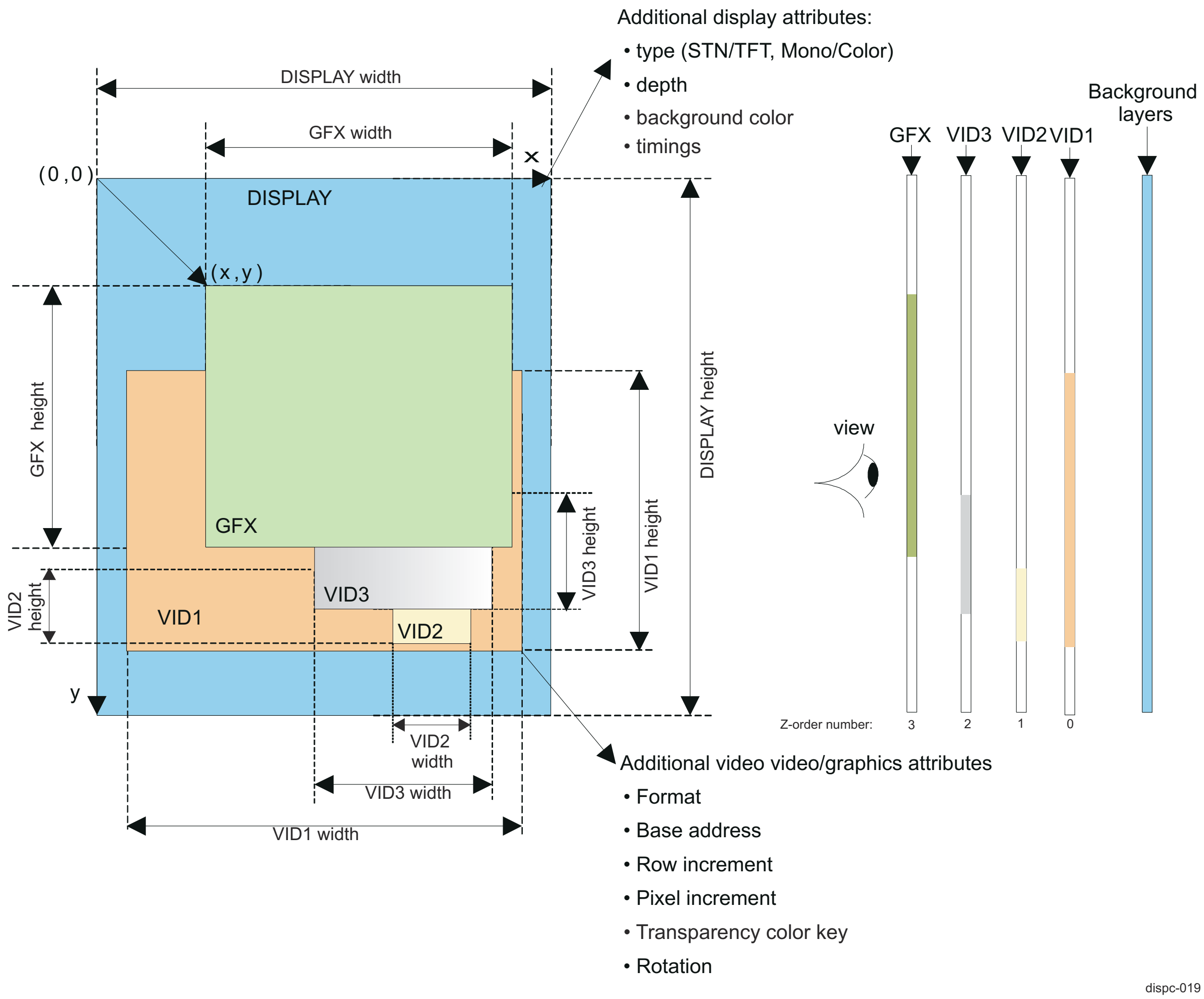


Figure 11-75. DISPC Example of Priority Rule: From Lower to Higher VID1, VID2, VID3, GFX

The DISPC is capable of outputting simultaniously two graphic/video windows in frame packing mode. Framepacking mode is enabled from DISPC_GFX_ATTRIBUTES[10] FRAMEPACKINGMODE / DISPC_VIDp_ATTRIBUTES[8] FRAMEPACKINGMODE. The position of the second graphic/video windows is defined in DISPC_GFX_POSITION2 / DISPC_VIDp_POSITION2 registers. The size is again defined in SIZEY and SIZEX bit fields of DISPC_GFX_SIZE[27:16][10:0]/ DISPC_VIDp_SIZE[27:16][10:0] registers.

11.2.4.13.1.2 DISPC Alpha Blender

Figure 11-76 shows the alpha blending processing in detail.

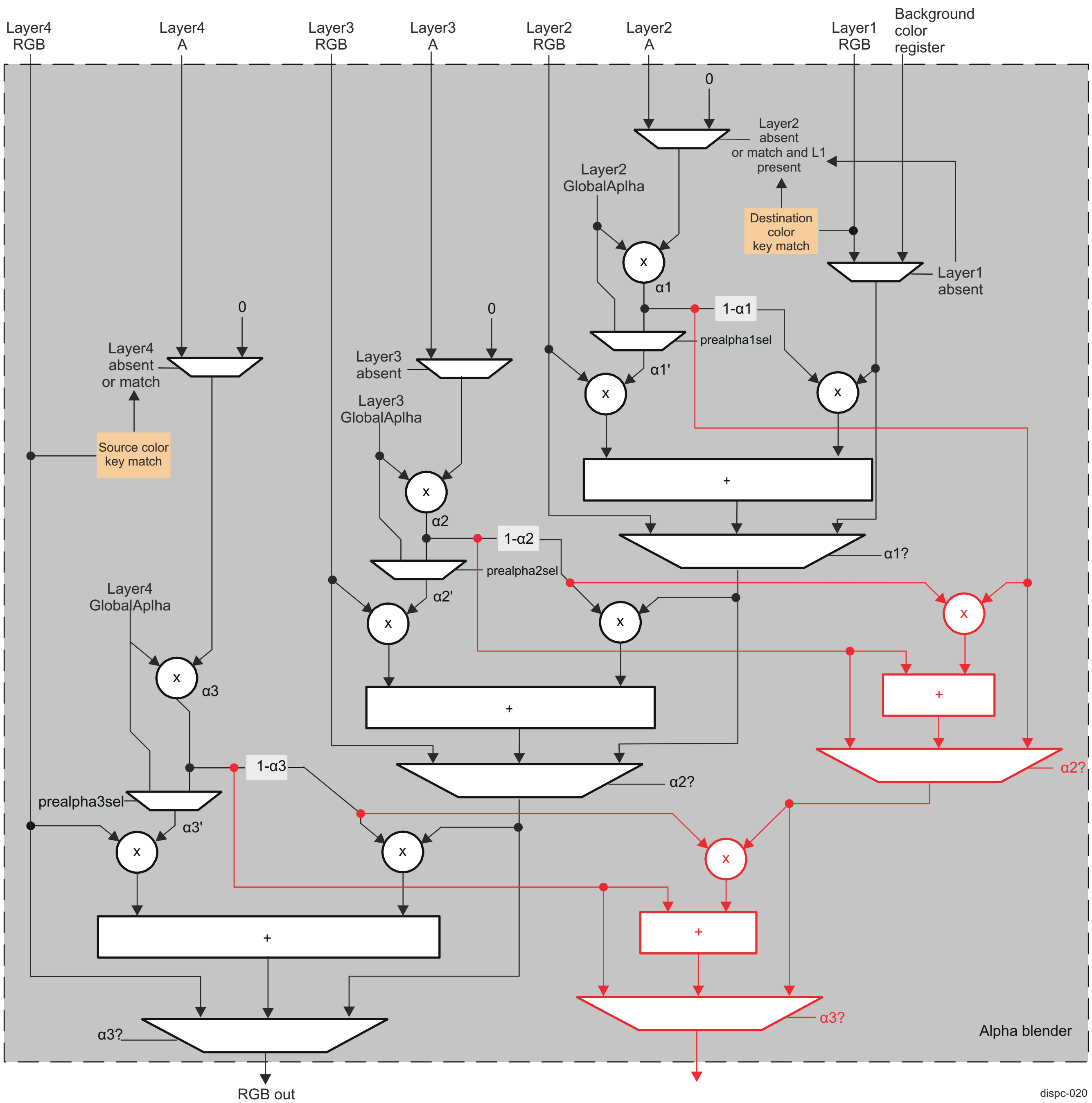


Figure 11-76. DISPC Alpha Blending Architecture With Premultiplied Alpha Support

CAUTION

The Z order of the Source Transparency Layer must have the value 3 (0x3: Z-order 3: layer above all the other layers).

Note

1-alpha operator corresponds to the basic 1s-complement operation.

The alpha blending value is defined by:

- The component value A when using an ARGB or RGBA pixel format.
  - For ARGB-1555, the alpha blending is defined using a 1-bit value. It is converted into an 8-bit value by duplicating the 1-bit value (see Table 11-84).
  - For ARGB-4444, the alpha blending is defined using a 4-bit value. It is converted into an 8-bit value by duplicating the 4-bit value (see Table 11-84).
  - If the pixel format contains no alpha blending value, the pixel alpha value is considered to be 0xFF and if alpha is equal to 0xFF, there is no multiplication.
  - For YUV formats, there is no alpha blending factor associated with each pixel value. Only the global alpha blending factor associated with the window displaying the YUV format is used.
- The global alpha blending value set in the individual bit fields of the DISPC_GLOBAL_ALPHA register for LCD:
  - VID3GLOBALALPHA
  - VID2GLOBALALPHA
  - VID1GLOBALALPHA
  - GFXGLOBALALPHA
- A total alpha blending value can be used when a combination of the pixel alpha blending value A and a global alpha blending is present. The resulting alpha value is determined as: Alpha = (Pixel Alpha × Global Alpha) / 256.

Table 11-84 lists the percentage of alpha blending in the function of the alpha blending value on 8 bits.

Table 11-84. DISPC Alpha Blending – ARGB

Alpha Blending 1-Bit Value	Alpha Blending 4-Bit Value	Alpha Blending 8-Bit Value (Converted Value or Resulting Alpha)	Percent Blending
0x0	0x0	0x00	100 (transparent)
N/A	0x1	0x11	93.33
N/A	0x2	0x22	86.6
N/A	...	...	...
N/A	0xE	0xEE	6.6
0x1	0xF	0xFF	0 (opaque)

Premultiplied Alpha

The image ARGB may have its RGB component already premultiplied with the alpha (ARGB) where:

- R = A*R
- G = A*G
- B = A*B

In that case, the processing is as follows:

- Color component of premultiplied layers are multiplied with the Global Alpha, if Global Alpha is not equal to 0.
- Color component of the composed underlying layers are multiplied with (1-A) × Global Alpha.

The additional premultiplied alpha option is associated with the pipelines GFX, VID1, VID2, and VID3. The option is accessible through the [28] PREMULTIPLYALPHA bit for the respective pipeline register:

- DISPC_GFX_ATTRIBUTES
- DISPC_VID1_ATTRIBUTES
- DISPC_VID2_ATTRIBUTES
- DISPC_VID3_ATTRIBUTES

The following settings are available:

- PREMULTIPLYALPHA bit = 0: Source is not premultiplied with alpha. Full blending is done in the DISPC.

- PREMULTIPLYALPHA bit = 1: Source is premultiplied with alpha. Partial blending is done.

Note

The *prealphasel* controls in Figure 11-76, correspond to the PREMULTIPLYALPHA of the pipelines mapped on the respective layers.

The logic marked in red in Figure 11-76 corresponds to the alpha value, computed when the write-back channel copies back to memory the premultiplied color component: A (destination) = A (source) + (1-A (source) × A (destination))

When the DISPC_WB_ATTRIBUTES[7] ALPHAENABLE bit is cleared, or when the overlay channel is not selected for WB, the computation of the A(destination) is disabled. The default value for DISPC_WB_ATTRIBUTES[7] ALPHAENABLE bit is 0x0. When the WB is configured to copy back one of the output channels (output of overlay), the following configurations are available:

- The ALPHAENABLE bit is set to 0x1: The WB pipe copies back to memory the premultiplied alpha calculated through the overlay.
- The ALPHAENABLE bit is set to 0x0: The alpha value is not written back.

Note

The DISPC_WB_ATTRIBUTES[7] ALPHAENABLE bit is effective only when one of the output channels is written back, otherwise it is ignored.

11.2.4.13.1.3 DISPC Transparency Color Keys

The two transparency color keys are the source transparency color key and the destination transparency color key. The transparency color key can be used only with RGB formats (ARGB, RGB, RGBA, xRGB, and RGBx). in this case the A information is ignored for the comparison between the pixel value and the color key value. It is possible to use YUV formats with some care because the comparison is between the input pixel value of the overlay manager from pipeline (GFX or one of the VID pipelines depending on the Z-order) and the color key value. The YUV data is converted to RGB format. If the original format is YUV, the user must consider the color space conversion processing to define the RGB color key value used for the comparison.

The transparency color key is enabled by setting the following bits to 0x1:

- DISPC_CONFIG1[10] TCKLCDENABLE (for LCD1)
- DISPC_CONFIG2[10] TCKLCDENABLE (for LCD2)
- DISPC_CONFIG3[10] TCKLCDENABLE (for LCD3)
- DISPC_CONFIG1[12] TCKTVENABLE (for TV)

The transparency color key is determined in the following bit fields.

- DISPC_TRANS_COLOR0[23:0] TRANSCOLORKEY (for LCD1)
- DISPC_TRANS_COLOR2[23:0] TRANSCOLORKEY (for LCD2)
- DISPC_TRANS_COLOR3[23:0] TRANSCOLORKEY (for LCD3)
- DISPC_TRANS_COLOR1[23:0] TRANSCOLORKEY (for TV)

Note

The video source transparency color key and graphics destination transparency color key cannot be active at the same time.

- Video source transparency color key

  The value of the video source transparency color key defines the encoded pixel data considered as a transparent pixel. The encoded pixel values with the source color key value are not visible, and the encoded pixel values of the under layers or solid background color are visible (the pixel alpha blending value of layer 4 is forced to 0x00, fully transparent).

The scaler can be enabled as a preprocessor in the VID pipeline, but it is necessary to consider the pixel scaling preprocessing in order to define the color key value to be used after the rescaling for the comparison between the input pixel value to the overlay manager and the color key value.

The source transparency color key mode is selected by setting the following bits to 0x1:
- DISPC_CONFIG1[11] TCKLCDSELECTION (for LCD1)
- DISPC_CONFIG2[11] TCKLCDSELECTION (for LCD2)
- DISPC_CONFIG3[11] TCKLCDSELECTION (for LCD3)
- DISPC_CONFIG1[13] TCKTVENABLE (for TV)

Figure 11-77 shows an example of source color key. The pixels with the transparency color key are not displayed; instead, pixels of the resulting layer underneath are shown.

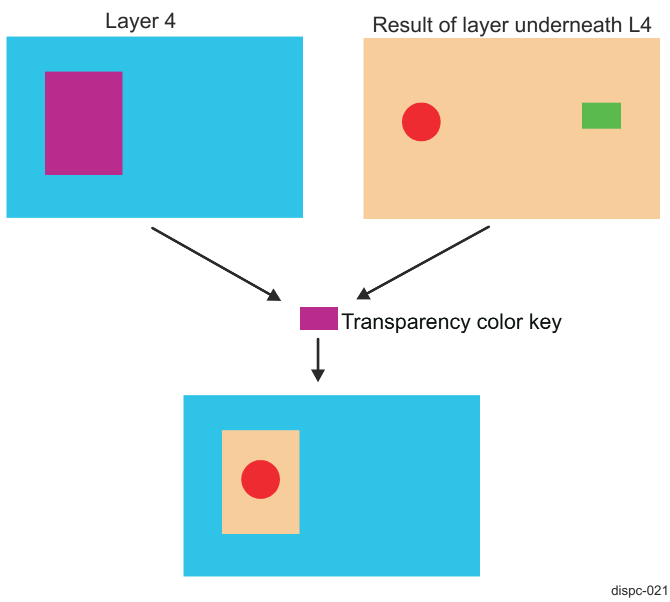


Figure 11-77. DISPC Source Transparency Color Key Example

- Graphics destination transparency color key value

The graphics destination transparency color key value defines the encoded pixels in layer 1, which are not displayed. Other layer 1 pixels (nonequal to destination transparency color key) are displayed over layer 2. The encoded pixel values with the destination color key value are pixels not visible on the screen because pixels at the same position in layer 2 are visible; otherwise, encoded pixels are visible above layer 2 . The destination transparency color key applies only if layer 1 overlaps layer 2 (see the Z-order section for details on layer position depending on the Z-order parameter in Section 11.2.4.13.1.1, *Priority Rule*); otherwise, the destination transparency color key is ignored.

The scaler can be enabled as a preprocessor in the VID pipeline. It is necessary, however, to consider the pixel scaling preprocessing in order to define the color key value to be used after rescaling for the comparison between the input pixel value to the overlay manager and the color key value.

The destination transparency color key mode is selected by setting the following bits to 0x0:
- DISPC_CONFIG1[11] TCKLCDSELECTION (for LCD1)
- DISPC_CONFIG2[11] TCKLCDSELECTION (for LCD2)
- DISPC_CONFIG3[11] TCKLCDSELECTION (for LCD3)
- DISPC_CONFIG1[13] TCKTVSELECTION (for TV)

Figure 11-78 shows an example of the destination color key. The pixels, equal to the transparency color key, are not displayed and are replaced by layer 2 pixels. All other layer 1 pixels, different from the transparency color key, are displayed over layer 2.

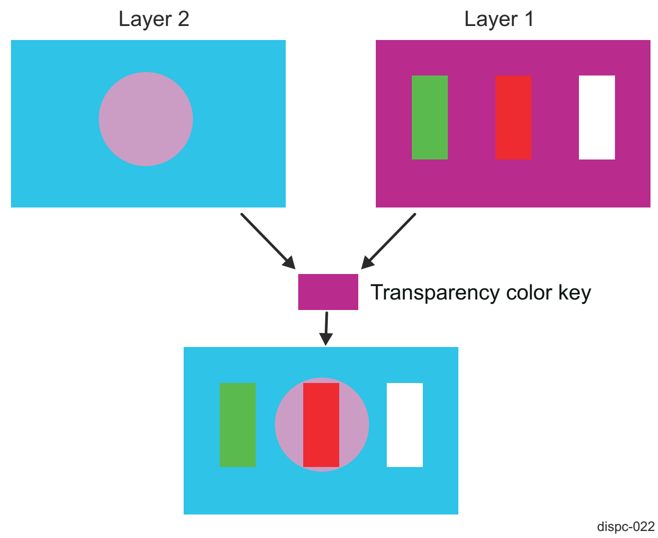


Figure 11-78. DISPC Destination Transparency Color Key Example

11.2.4.13.1.4 DISPC Overlay Optimization

The overlay optimization consists in fetching only the required pixels (that is, pixels that contribute to the final picture to be displayed [LCD1, LCD2, LCD3, or TV]). The decision to fetch the pixel from memory is based on the information available in the registers and on the following rules:

- The layer is enabled.
- The global alpha blending factor for the layer is different than 0x00.
- The current layer is behind an nonopaque layer (global alpha blending factor is different than 0xFF for the layer in the preceding).

The result of the overlay optimization is a reduction of the bandwidth by fetching only the mandatory pixels. The overlay mechanism is independent for each overlay: LCD1, LCD2, LCD3, and TV. Because each layer (GFX, VID1, VID2, and VID3) can be associated to only one overlay at a time, it is possible to optimize the fetch of the pixels for each layer based on the overlay information. The overlay optimization must be run on the DMA engine time window and not on the display time window. The pixels are fetched by the DMA engine before the display processing.

The overlay optimization is enabled by setting the following bits to 0x1:

- DISPC_CONTROL1[12] OVERLAYOPTIMIZATION (for LCD1)
- DISPC_CONTROL2[12] OVERLAYOPTIMIZATION (for LCD2)
- DISPC_CONTROL3[12] OVERLAYOPTIMIZATION (for LCD3)
- DISPC_CONTROL2[13] TVOVERLAYOPTIMIZATION (for TV)

Note

- The pixel alpha blending factor in case of ARGB and RGBA formats cannot be used to take advantage of the pixel fully transparent (alpha blending factor equals 0x00).
- By default, the overlay optimization is disabled (OVERLAYOPTIMIZATION bits = 0x0 for all DISPC outputs), and all the data for all the enabled pipelines are fetched from memory regardless of the overlay or alpha blending configuration.

11.2.4.13.2 DISPC Gamma Correction Unit

Each LCD output supports a look up table to perform gamma correction on the display output.

Figure 11-79 shows the internal architecture of the gamma table for LCD1 output.

The gamma table is split into three memories of 256 × 8-bit entries and indexed by the R/G/B component data.

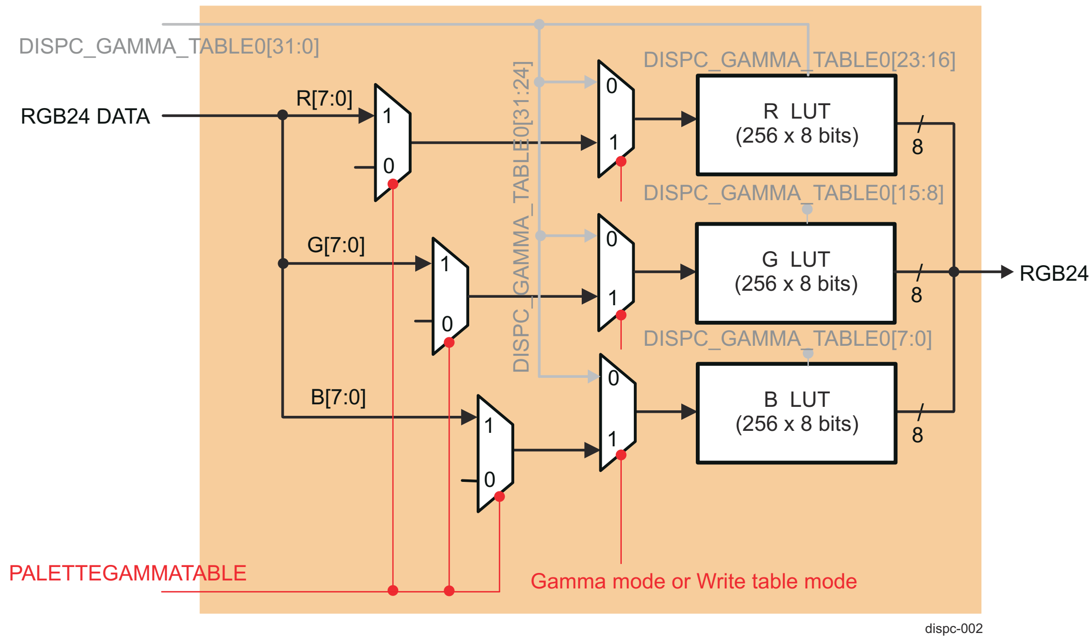


Figure 11-79. DISPC LCD1 Gamma Correction Architecture

When performing gamma correction, the selected encoded pixel values based on the color keys by the overlay manager from the video or graphics paths are sent to the gamma curve table. Each component of encoded pixel value is used as a pointer to index 1 out of 256 × 24-bit gamma curve entries in the table. Each 8-bit component is replaced with the 8-bit table value corresponding to R, G, or B component. The table is loaded by software. It is possible to load only part of the table. For each access to the table, the 24-bit value is associated with index in the table by concatenating the 24-bit value (LSB of 32-bit access) and the 8-bit index value (MSB of the 32-bit access).

The sequence to load the gamma table is:

1. SW writes (only writes are supported) 32-bit gamma correction values using single access, or burst access in streaming mode, into DISPC_GAMMA_TABLE0 register (for LCD1 output). The LSB 24 bits [23:0] are used for the value, and the MSB 8 bits [31:24] are used for the index into the table.
2. Loop to Step 1, if there is a new access to the gamma table register. The SW can access other registers between two accesses to the gamma table register.

SW needs to ensure that there is no visible effect when modifying the table, since it is not under HW control.

For LCD1 output the usage of the gamma table is activated by setting DISPC_CONFIG1[3] PALETTEGAMMATABLE register bit to 0x1.

For LCD2 and LCD3 outputs, the associated gamma tables are enabled by setting DISPC_CONFIG1[9] GAMATABLEENABLE register bit to 0x1.

- For LCD2 output the gamma values are loaded into DISPC_GAMMA_TABLE1 register
- For LCD3 output the gamma values are loaded into DISPC_GAMMA_TABLE3 register

Figure 11-80 describes the format of one of the gamma curve values in the memory.

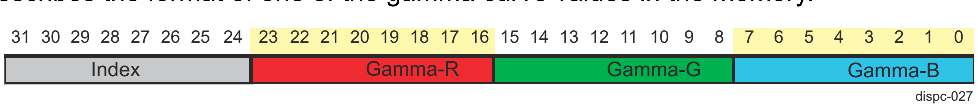


Figure 11-80. DISPC Data Memory Organization for Gamma Mode in LCD Output

11.2.4.13.3 DISPC Color Phase Rotation Unit

The CPR unit can be used to correct the LCD output colorimetry in case of nonpure white backlight.

The CPR is enabled by setting the DISPC_CONFIGo[15] CPR bit to 0x1. The coefficients are programmed in the following registers:

- Red 10-bit signed coefficients in DISPC_CPRo_COEF_R
- Green 10-bit signed coefficients in DISPC_CPRo_COEF_G
- Blue 10-bit signed coefficients in DISPC_CPRo_COEF_B

The CPR can be selected for active matrix panel. The logic is integrated after the LCD overlay manager and the gamma correction, and before the spatial/temporal dithering. The CPR can be selected to correct the nonpure white backlight of the LCD module by using a programmable matrix to convert the 24-bit RGB pixel value into a new 24-bit RGB pixel value. The matrix is programmed through a set of nine 10-bit signed coefficients. The output of the calculation is clipped to [0:255]. The CPR is processed by the equation shown in Figure 11-81. Table 11-85 lists all coefficients with their respective bit field registers for settings.

$$\begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} Rr & Rg & Rb \\ Gr & Gg & Gb \\ Br & Bg & Bb \end{bmatrix} * \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix}$$

dispc-023

Figure 11-81. DISPC CPR Matrix

Table 11-85. DISPC CPR or RGB to YUV Coefficients With Associated Bit Fields

Registers	Bit Field	Color Phase Rotation	RGB to YUV
DISPC_CPRo_COEF_R	RR	Rr	Yr
	RG	Rg	Yg
	RB	Rb	Yb
DISPC_CPRo_COEF_G	GR	Gr	Cbr
	GG	Gg	Cbg
	GB	Gb	Cbb
DISPC_CPRo_COEF_B	BR	Br	Crr
	BG	Bg	Crg
	BB	Bb	Crb

Figure 11-82 shows the CPR macro-architecture.

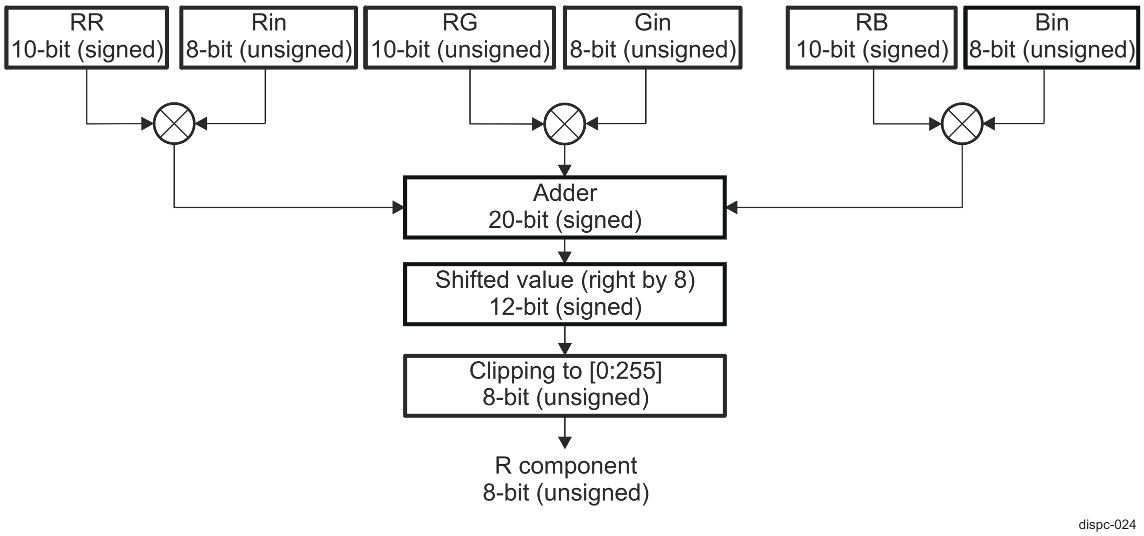


Figure 11-82. DISPC CPR Macro-Architecture

Note

CPR is kept as 8 bit output. For the optional 10 bit BT.656-4 format, zero is simply added as 2 LSBs.

11.2.4.13.4 DISPC Color Space Conversion

The Color Phase Rotation block can be used to perform color space conversion. Table 11-85 lists the associated coefficients with their respective register bit-fields. The Color Space Conversion logic uses the CPR registers for the coefficients. The Color Space Conversion processing is enabled by setting the DISPC_CONFIGo[24] COLORCONVENABLE register bit to 0x1.

The color space conversion on the LCD output can be selected in order to convert from 24-bit RGB to YUV444. The matrix is programmed through a set of nine 10-bit signed coefficients. The result is clipped to [16:235] for the luminance and [16:240] for the chrominance, when full-range equals zero (DISPC_CONFIGo[25] FULLRANGE = 0).

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} YR & YG & YB \\ CrR & CrG & CrB \\ CbR & CbG & CbB \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

dispc-098

Figure 11-83. DISPC RGB to YUV Registers (FullRange=0)

If the programmed active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [0:255], the values Y, Cb, and Cr are clipped to the range [0:255]. The following equation gives the 11-bit coefficients of the RGB to YUV color space conversion, for full-range (DISPC_CONFIGo[25] FULLRANGE = 1)

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} YR & YG & YB \\ CrR & CrG & CrB \\ CbR & CbG & CbB \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

dispc-098

Figure 11-84. DISPC RGB to YUV Registers (FullRange=1)

In order to provide YUV422 data to the BT-656 or BT-1120 blocks, the YUV444 format is converted to YUV422 by sub-sampling the chrominance values. The hardware does this by averaging two-by-two the chrominance samples. The conversion from YUV444 to YUV422 is performed when the color space conversion is enabled. Thus, the output of the CPR unit is always YUV422 after color space conversion.

11.2.4.13.5 DISPC BT.656 and BT.1120 Modes

The LCD outputs (LCD1, LCD2 and LCD3) can be configured in BT.656 or BT.1120 mode. The following standards are not supported in BT.656 mode:

- BT.470 (Support for conventional Analog TV Systems)
- BT.803 (The avoidance of interference generated by digital television studio equipment)
- BT.1364 (Format of ancillary data signals carried in digital component studio interfaces)

Unsupported formats when BT.1120 mode is used:

- BT.1364 (Support for Ancillary Data during blanking period)

BT.656/BT.1120 modes use embedded EAV/SAV syncs.

> **CAUTION**
>
> DISPC supports maximum 256 bytes of horizontal blanking when the LCD outputs are configured in BT modes (bitfield HSW of DISPC_TIMING_H1/2/3 registers is 8 bits wide). This is not compliant with BT.656 and BT.1120 standards, both of which require higher blanking periods. If more than 256 bytes of horizontal blanking are required by the application, then the RGB mode must be used for the LCD outputs.

Figure 11-85 shows signal mapping on DISPC_LCD_DATA[23:0] data bus. Bits 9 to 0 are used in BT.656 mode (10-bit).

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused														BT.656									

Figure 11-85. DISPC Signal Mapping in BT.656 Mode

> **Note**
>
> For compatibility with existing 8-bit interfaces, the two LSB are ignored, and only bits [9:2] are used.

Figure 11-86 shows signal mapping on DISPC_LCD_DATA[23:0] data bus for BT.1120 mode. Bits [19:10] (CbCr) and [9:0] (Y) are used in 20-bit mode. Bits [19:12] (CbCr) and [9:2] (Y) are used in 16-bit mode (YCbCr 4:2:2).

				BT.1120 (CbCr, 16-bit mode)										BT.1120 (Y, 16-bit mode)									
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused				BT.1120 (CbCr, 20-bit mode)										BT.1120 (Y, 20-bit mode)									

dispc-049x

Figure 11-86. DISPC Signal Mapping in BT.1120 Mode

Note

The LCD outputs support both interlace and progressive content in BT.656/BT.1120 modes.

In progressive mode the maximum resolution will be limited, as it requires two clock cycles to send out one pixel.

11.2.4.13.5.1 Blanking

During the transmission of the video signal, the portion of the stream in-between active video data segments is known as the horizontal blanking interval.

Strictly speaking this entire region is the blanking interval, but this interval also includes the EAV and SAV codes. The remaining bytes of information in a digital blanking interval are filled with values corresponding to the blanking levels of the Cb, Y and Cr signals respectively, and in accordance with the standard multiplex sequence for the stream (CbYCrY..). The blanking levels are as follows:

- Cb = 80h
- Y = 10h
- Cr = 80h.

The sequence in the BLANKING region of the data stream is therefore: *80h, 10h, 80h, 10h.....80h, 10h*

For more details on setting the blanking timing values for BT.1120 and/or BT.656 mode, see Section 11.2.4.13.8, *DISPC Timing Generator and Panel Settings*.

11.2.4.13.5.2 EAV and SAV

The End of Active Video(EAV) and Start of Active Video (SAV) parts of the stream are timing codes. Their function can be summarized as follows:

- EAV – marks the end of the active video data within the current line and therefore also the start of the subsequent line.
- SAV – heralds the start of the active video data within the current line.

These codes are embedded within the BT.656 video data stream, thereby eliminating the need for additional timing signals (HSYNC, VSYNC) to be included as part of the interface.

Both EAV and SAV codes are comprised of a sequence of four bytes (FFh – 00h – 00h - XY). The first three bytes in the sequence constitute a fixed preamble. The fourth byte, contains information about the field being transmitted (Field 1 or Field 2 in an interlaced video signal), the state of field blanking (Vertical) and the state of line blanking (Horizontal). The bit assignment for this byte of the code is shown in Figure 11-87, with the function of each bit described inTable 11-86.

MSB							LSB
1	F	V	H	P3	P2	P1	P0

Figure 11-87. Bit-assignment for the fourth byte of EAV/SAV codes

Table 11-86. Bit function

Bit	Symbol	Function
7	1	Always set to '1'.
6	F	Field bit. 0 – Field 1 1 – Field 2

Table 11-86. Bit function (continued)

Bit	Symbol	Function
5	V	Vertical Blanking Status bit. This bit goes High during a vertical field blanking interval, otherwise it remains Low.
4	H	Horizontal Blanking Status bit. 0 – byte is part of SAV code (i.e. stream is entering an active video data region for the current line) 1 – byte is part of EAV code (i.e. stream has entered a horizontal blanking interval - start of a new line)
3	P3	Protection bit 3
2	P2	Protection bit 2
1	P1	Protection bit 1
0	P0	Protection bit 0

The protection bits allow for detection and correction of 1-bit errors and the detection of 2-bit errors. The status of P3, P2, P1 and P0 depend on the states of bits F, V and H. This dependency is shown in Table 11-87.

Table 11-87. Status of protection bits as F, V and H vary

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Enabling BT.656 or BT.1120 format is done by setting:

- DISPC_CONFIG1[20] BT656ENABLE or DISPC_CONFIG1[21]BT1120ENABLE registers for LCD1 output
- DISPC_CONFIG2[20] BT656ENABLE or DISPC_CONFIG2[21]BT1120ENABLE registers for LCD2 output
- DISPC_CONFIG3[20] BT656ENABLE or DISPC_CONFIG3[21]BT1120ENABLE registers for LCD3 output

Note

It is not posssible to enable BT.656 and BT.1120 mode simultaneously on the same LCD output

11.2.4.13.6 DISPC Active Matrix

Depending on the color depth, the active matrix output:

- 24 bpp supports 16,777,216 colors.
- 18 bpp supports 262,144 colors.
- 16 bpp supports 65,536 colors.
- 12 bpp supports 4096 colors.

When in active matrix path configuration, after setting the DISPC_CONTROLo[3] STNTFT bit to 0x1, two submodules must be configured:

- Spatial/temporal dithering
- Multiple cycle output format (TDM)

11.2.4.13.6.1 DISPC Spatial/Temporal Dithering

When the active matrix path is used, the spatial/temporal dithering logic can be selected to enhance the quality of the active matrix outputs. The encoded pixel values are used by spatial/temporal dithering logic to display the data in a lower color depth on the LCD panel. The dithering logic is integrated after the CPR and before the TDM. The spatial/temporal dithering algorithm is based on the (x,y) pixel position and frame rate control. The dithering logic can process the pixels over one frame, two frames, or four frames. The number of frames is selected by setting the DISPC_CONTROLo[31:30] SPATIALTEMPORALDITHERINGFRAMES bit field. In the case of a single frame, only spatial processing is applied, and in multiple frames, spatial and temporal processing are applied to the pixels. The spatial/temporal dithering logic is enabled by setting the DISPC_CONTROLo[7] STDITHERENABLE bit to 0x1.

Note

When the spatial/temporal dithering logic is disabled (see also Table 11-88):

- If the interface data bus is smaller than the pixel format size, then the MSBs of the pixel color components are output on the interface data bus.
- If the interface data bus is larger than the pixel format size, then by programming the pixel components replication active/inactive, the MSB is replicated to the LSB of the interface data bus (when replication is enabled), or the LSB is filled with 0s (when replication is disabled). The pixel components replication is performed in the associated pipeline, as described in Section 11.2.4.9.1 and Section 11.2.4.10.1, *DISPC Replication Logic*.

Table 11-88. DISPC LCD Data Output (Spatial/Temporal Dithering Disabled)

Pixel Format	RGB12(2)				RGB16(2)				RGB24(2)			
Display Panel Interface(1)	12-bit	16-bit	18-bit	24-bit	12-bit	16-bit	18-bit	24-bit	12-bit	16-bit	18-bit	24-bit
voutX_d23				R3				R4				R7
voutX_d22				R2				R3				R6
voutX_d21				R1				R2				R5
voutX_d20				R0				R1				R4
voutX_d19				R3/0				R0				R3
voutX_d18				R2/0				R4/0				R2
voutX_d17			R3	R1/0			R4	R3/0			R7	R1
voutX_d16			R2	R0/0			R3	R2/0			R6	R0
voutX_d15		R3	R1	G3		R4	R2	G5		R7	R5	G7
voutX_d14		R2	R0	G2		R3	R1	G4		R6	R4	G6
voutX_d13		R1	R3/0	G1		R2	R0	G3		R5	R3	G5
voutX_d12		R0	R2/0	G0		R1	R4/0	G2		R4	R2	G4
voutX_d11	R3	R3/0	G3	G3/0	R4	R0	G5	G1	R7	R3	G7	G3
voutX_d10	R2	G3	G2	G2/0	R3	G5	G4	G0	R6	G7	G6	G2
voutX_d9	R1	G2	G1	G1/0	R2	G4	G3	G5/0	R5	G6	G5	G1
voutX_d8	R0	G1	G0	G0/0	R1	G3	G2	G4/0	R4	G5	G4	G0
voutX_d7	G3	G0	G3/0	B3	G5	G2	G1	B4	G7	G4	G3	B7
voutX_d6	G2	G3/0	G2/0	B2	G4	G1	G0	B3	G6	G3	G2	B6
voutX_d5	G1	G2/0	B3	B1	G3	G0	B4	B2	G5	G2	B7	B5
voutX_d4	G0	B3	B2	B0	G2	B4	B3	B1	G4	B7	B6	B4
voutX_d3	B3	B2	B1	B3/0	B4	B3	B2	B0	B7	B6	B5	B3
voutX_d2	B2	B1	B0	B2/0	B3	B2	B1	B4/0	B6	B5	B4	B2
voutX_d1	B1	B0	B3/0	B1/0	B2	B1	B0	B3/0	B5	B4	B3	B1
voutX_d0	B0	B3/0	B2/0	B0/0	B1	B0	B4/0	B2/0	B4	B3	B2	B0

(1) X = 1, 2, or 3 indicates the corresponding parallel video output (VOUT1, VOUT2, or VOUT3)
(2) Rx/0, Gx/0, and Bx/0 indicate, if the MSB is replicated to the LSB, or the LSB is filled with 0s.

11.2.4.13.6.2 DISPC Multiple Cycle Output Format (TDM)

When the TDM is enabled (DISPC_CONTROLo[20] TDMENABLE = 1), after the active matrix display processing, the pixels are formatted on one or multiple cycles (a maximum of three cycles). The number of bits for each cycle is set in the DISPC_DATAo_CYCLE1 register for the first cycle, the DISPC_DATAo_CYCLE2 register for the second cycle, and the DISPC_DATAo_CYCLE3 register for the third cycle. The interface data bus width can be 8, 9, 12, or 16 bits. The configuration of the data bus is done in the DISPC_CONTROLo[22:21] TDMPARALLELMODE bit field.

When the TDM is disabled (DISPC_CONTROLo[20] TDMENABLE = 0), the DISPC outputs the pixels using the conventional formats: active matrix display monochrome/color. In this case, the configuration of the data bus width is done through the DISPC_CONTROLo[9:8] TFTDATALINES bit field.

Figure 11-88 through Figure 11-91 show various examples of TDM settings in the function of pixel data formats and the interface data bus width.

	24-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[7]	G0[7]	B0[7]
Data[6]	R0[6]	G0[6]	B0[6]
Data[5]	R0[5]	G0[5]	B0[5]
Data[4]	R0[4]	G0[4]	B0[4]
Data[3]	R0[3]	G0[3]	B0[3]
Data[2]	R0[2]	G0[2]	B0[2]
Data[1]	R0[1]	G0[1]	B0[1]
Data[0]	R0[0]	G0[0]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x2
DISPC_DATAo_CYCLE1 = 0x00000008
DISPC_DATAo_CYCLE2 = 0x00000008
DISPC_DATAo_CYCLE3 = 0x00000008

	18-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[5]	G0[3]	x
Data[6]	R0[4]	G0[2]	x
Data[5]	R0[3]	G0[1]	x
Data[4]	R0[2]	G0[0]	x
Data[3]	R0[1]	B0[5]	x
Data[2]	R0[0]	B0[4]	x
Data[1]	G0[5]	B0[3]	B0[1]
Data[0]	G0[4]	B0[2]	B0[0]

DISPC_CONTROLo.TDMCycleFormat = 0x2
DISPC_DATAo_CYCLE1 = 0x00000008
DISPC_DATAo_CYCLE2 = 0x00000008
DISPC_DATAo_CYCLE3 = 0x00000002

	16-bpp	
	1st cycle	2nd cycle
Data[7]	R0[4]	G0[2]
Data[6]	R0[3]	G0[1]
Data[5]	R0[2]	G0[0]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x00000008
DISPC_DATAo_CYCLE2 = 0x00000008

	12-bpp	
	1st cycle	2nd cycle
Data[7]	R0[3]	x
Data[6]	R0[2]	x
Data[5]	R0[1]	x
Data[4]	R0[0]	x
Data[3]	G0[3]	B0[3]
Data[2]	G0[2]	B0[2]
Data[1]	G0[1]	B0[1]
Data[0]	G0[0]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x00000008
DISPC_DATAo_CYCLE2 = 0x00000004

dispc-057

Figure 11-88. DISPC 8-Bit Interface Settings

	24-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[8]	R0[7]	G0[6]	x
Data[7]	R0[6]	G0[5]	x
Data[6]	R0[5]	G0[4]	x
Data[5]	R0[4]	G0[3]	B0[5]
Data[4]	R0[3]	G0[2]	B0[4]
Data[3]	R0[2]	G0[1]	B0[3]
Data[2]	R0[1]	G0[0]	B0[2]
Data[1]	R0[0]	B0[7]	B0[1]
Data[0]	G0[7]	B0[6]	B0[0]

DISPC_CONTROLo.TDMCycleFormat = 0x2
DISPC_DATAo_CYCLE1 = 0x00000009
DISPC_DATAo_CYCLE2 = 0x00000009
DISPC_DATAo_CYCLE3 = 0x00000006

	18-bpp	
	1st cycle	2nd cycle
Data[8]	R0[5]	G0[2]
Data[7]	R0[4]	G0[1]
Data[6]	R0[3]	G0[0]
Data[5]	R0[2]	B0[5]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x00000009
DISPC_DATAo_CYCLE2 = 0x00000009

	16-bpp	
	1st cycle	2nd cycle
Data[8]	R0[4]	x
Data[7]	R0[3]	x
Data[6]	R0[2]	G0[1]
Data[5]	R0[1]	G0[0]
Data[4]	R0[0]	B0[4]
Data[3]	G0[5]	B0[3]
Data[2]	G0[4]	B0[2]
Data[1]	G0[3]	B0[1]
Data[0]	G0[2]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x00000009
DISPC_DATAo_CYCLE2 = 0x00000007

	12-bpp	
	1st cycle	2nd cycle
Data[8]	R0[3]	x
Data[7]	R0[2]	x
Data[6]	R0[1]	x
Data[5]	R0[0]	x
Data[4]	G0[3]	x
Data[3]	G0[2]	x
Data[2]	G0[1]	B0[2]
Data[1]	G0[0]	B0[1]
Data[0]	B0[3]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x00000009
DISPC_DATAo_CYCLE2 = 0x00000003

dispc-058

Figure 11-89. DISPC 9-Bit Interface Settings

	24-bpp	
	1st cycle	2nd cycle
Data[11]	R0[7]	G0[3]
Data[10]	R0[6]	G0[2]
Data[9]	R0[5]	G0[1]
Data[8]	R0[4]	G0[0]
Data[7]	R0[3]	B0[7]
Data[6]	R0[2]	B0[6]
Data[5]	R0[1]	B0[5]
Data[4]	R0[0]	B0[4]
Data[3]	G0[7]	B0[3]
Data[2]	G0[6]	B0[2]
Data[1]	G0[5]	B0[1]
Data[0]	G0[4]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x0000000C
DISPC_DATAo_CYCLE2 = 0x0000000C

	18-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[11]	R0[5]	B0[5]	G1[5]
Data[10]	R0[4]	B0[4]	G1[4]
Data[9]	R0[3]	B0[3]	G1[3]
Data[8]	R0[2]	B0[2]	G1[2]
Data[7]	R0[1]	B0[1]	G1[1]
Data[6]	R0[0]	B0[0]	G1[0]
Data[5]	G0[5]	R1[5]	B1[5]
Data[4]	G0[4]	R1[4]	B1[4]
Data[3]	G0[3]	R1[3]	B1[3]
Data[2]	G0[2]	R1[2]	B1[2]
Data[1]	G0[1]	R1[1]	B1[1]
Data[0]	G0[0]	R1[0]	B1[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x3
DISPC_DATAo_CYCLE1 = 0x0000000C
DISPC_DATAo_CYCLE2 = 0x00060606
DISPC_DATAo_CYCLE3 = 0x000C0000

	16-bpp	
	1st cycle	2nd cycle
Data[11]	R0[4]	x
Data[10]	R0[3]	x
Data[9]	R0[2]	x
Data[8]	R0[1]	x
Data[7]	R0[0]	x
Data[6]	G0[5]	x
Data[5]	G0[4]	x
Data[4]	G0[3]	x
Data[3]	G0[2]	B0[3]
Data[2]	G0[1]	B0[2]
Data[1]	G0[0]	B0[1]
Data[0]	B0[4]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x0000000C
DISPC_DATAo_CYCLE2 = 0x00000004

	12-bpp
	1st cycle
Data[11]	R0[3]
Data[10]	R0[2]
Data[9]	R0[1]
Data[8]	R0[0]
Data[7]	G0[3]
Data[6]	G0[2]
Data[5]	G0[1]
Data[4]	G0[0]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x0
DISPC_DATAo_CYCLE1 = 0x0000000C

dispc-059

Figure 11-90. DISPC 12-Bit Interface Settings

	24-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[15]	R0[7]	B0[7]	G1[7]
Data[14]	R0[6]	B0[6]	G1[6]
Data[13]	R0[5]	B0[5]	G1[5]
Data[12]	R0[4]	B0[4]	G1[4]
Data[11]	R0[3]	B0[3]	G1[3]
Data[10]	R0[2]	B0[2]	G1[2]
Data[9]	R0[1]	B0[1]	G1[1]
Data[8]	R0[0]	B0[0]	G1[0]
Data[7]	G0[7]	R1[7]	B1[7]
Data[6]	G0[6]	R1[6]	B1[6]
Data[5]	G0[5]	R1[5]	B1[5]
Data[4]	G0[4]	R1[4]	B1[4]
Data[3]	G0[3]	R1[3]	B1[3]
Data[2]	G0[2]	R1[2]	B1[2]
Data[1]	G0[1]	R1[1]	B1[1]
Data[0]	G0[0]	R1[0]	B1[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x3
DISPC_DATAo_CYCLE1 = 0x00000010
DISPC_DATAo_CYCLE2 = 0x00080808
DISPC_DATAo_CYCLE3 = 0x00100000

	18-bpp	
	1st cycle	2nd cycle
Data[15]	R0[5]	x
Data[14]	R0[4]	x
Data[13]	R0[3]	x
Data[12]	R0[2]	x
Data[11]	R0[1]	x
Data[10]	R0[0]	x
Data[9]	G0[5]	x
Data[8]	G0[4]	x
Data[7]	G0[3]	X
Data[6]	G0[2]	x
Data[5]	G0[1]	x
Data[4]	G0[0]	x
Data[3]	B0[5]	x
Data[2]	B0[4]	x
Data[1]	B0[3]	B0[1]
Data[0]	B0[2]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x1
DISPC_DATAo_CYCLE1 = 0x00000010
DISPC_DATAo_CYCLE2 = 0x00000002

	16-bpp
	1st cycle
Data[15]	R0[4]
Data[14]	R0[3]
Data[13]	R0[2]
Data[12]	R0[1]
Data[11]	R0[0]
Data[10]	G0[5]
Data[9]	G0[4]
Data[8]	G0[3]
Data[7]	G0[2]
Data[6]	G0[1]
Data[5]	G0[0]
Data[4]	B0[4]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x0
DISPC_DATAo_CYCLE1 = 0x00000010

	12-bpp
	1st cycle
Data[15]	x
Data[14]	x
Data[13]	x
Data[12]	x
Data[11]	R0[3]
Data[10]	R0[2]
Data[9]	R0[1]
Data[8]	R0[0]
Data[7]	G0[3]
Data[6]	G0[2]
Data[5]	G0[1]
Data[4]	G0[0]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

DISPC_CONTROLo.TDMCYCLEFORMAT = 0x0
DISPC_DATAo_CYCLE1 = 0x0000000C

dispc-060

Figure 11-91. DISPC 16-Bit Interface Settings

11.2.4.13.7 DISPC Synchronized Buffer Update

A synchronization mismatch between the frame buffer and the display refreshes, named tearing effect, can lead to images that appear to be stretched on the screen. To avoid it, a synchronization mechanism is used between the DISPC and the process that updates the buffer. An interrupt is generated when the display reaches a predefined line number. The PROGRAMMEDLINENUMBER_IRQ interrupt (DISPC_IRQSTATUS and DISPC_IRQENABLE) is a level signal and stays active during the programmed line of the display.

11.2.4.13.8 DISPC Timing Generator and Panel Settings

When Active Matrix Display is used the following registers must be set:

The size of and panel is defined by:

- Number of lines, DISPC_SIZE_LCDo[27:16] LPP bit field, with a value from 1 to 4096
- Number of pixels per line, DISPC_SIZE_LCDo[11:0]PPL bit field, with a value from 1 to 4096

Standard HSYNC/VSYNC timing generation are programmable for each LCD outputs independently:

- Horizontal front porch is set in the DISPC_TIMING_Ho[19:8] HFP bit field.
- Horizontal back porch is set in the DISPC_TIMING_Ho[31:20] HBP bit field.
- Horizontal synchronization pulse width is set in the DISPC_TIMING_Ho[7:0] HSW bit field.
- Vertical front porch is set in the DISPC_TIMING_Vo[19:8] VFP bit field.
- Vertical back porch is set in the DISPC_TIMING_Vo[31:20] VBP bit field.
- Vertical synchronization pulse width is set in the DISPC_TIMING_Vo[7:0] VSW bit field.

When the output is in BT.1120 or BT.656 mode, the following timing constant are mapped onto the DISPC_TIMING_Ho and DISPC_TIMING_Vo registers:

- Progressive mode:
  - Horizontal blanking is set in the DISPC_TIMING_Ho[7:0] HSW bit field
  - Vertical frame blanking No 1 is set in the DISPC_TIMING_Vo[19:8] VFPbit field.
  - Vertical frame blanking No 2 is set in the DISPC_TIMING_Vo[31:20] VBPbit field.
  - Number of lines is set in the DISPC_SIZE_LCD1[27:16] LPP bit field
  - Number of pixels per line is set in the DISPC_SIZE_LCD1[11:0] PPL bit field.
- Interlaced mode:
  - Horizontal blanking is set in the DISPC_TIMING_Ho[7:0] HSW bit field.
  - Vertical field blanking No 1 for Even Field is set in the DISPC_TIMING_Ho[19:8] HFP bit field.
  - Vertical field blanking No 2 for Even Field is set in the DISPC_TIMING_Ho[31:20] HBP bit field.
  - Vertical field blanking No 1 for Odd Field is set in the DISPC_TIMING_Vo[19:8] VFP bit field.
  - Vertical field blanking No 2 for Odd Field is set in the DISPC_TIMING_Vo[31:20] VBP bit field.
  - Number of lines per field (even) is set in the DISPC_SIZE_LCD1[27:16] LPP bit field.
  - Delta number of odd field compared to even field (in a single line) is set in the DISPC_SIZE_LCD1[15:14] DELTA bit field.
  - Number of pixels per line is set in the DISPC_SIZE_LCD1[11:0] PPL bit field.

Horizontal/vertical synchronization and ACBIAS signals polarity are programmable by setting the DISPC_POL_FREQo[12] IVS, DISPC_POL_FREQo[13] IHS, and DISPC_POL_FREQo[15] IEO bits. These signals can be gated by setting the DISPC_CONFIGo[7] VSYNCGATED and DISPC_CONFIGo[6] HSYNCGATED bits.

The latch of data can be driven on the rising or falling edge of the pixel clock by setting the IPC bit of DISPC_POL_FREQo[14] IPC and CTRL_CORE_SMA_SW_1[] DSS_CHx_IPC (the values must mach) registers. The drive of the SYNC and VSYNC signals in the function of the pixel clock is done by setting the DISPC_POL_FREQo[16] RF and CTRL_CORE_SMA_SW_1[] DSS_CHx_RF bit.

Table 11-89 describes the programming rules for LCD timing.

Table 11-89. DISPC Programming Rules

	No Downsampling	Downsampling H or V	Downsampling H + V
(HBP + HSW + HFP) × PCD	>8	>10	>20

Figure 11-92 shows the timing values description in the case of an active matrix display.

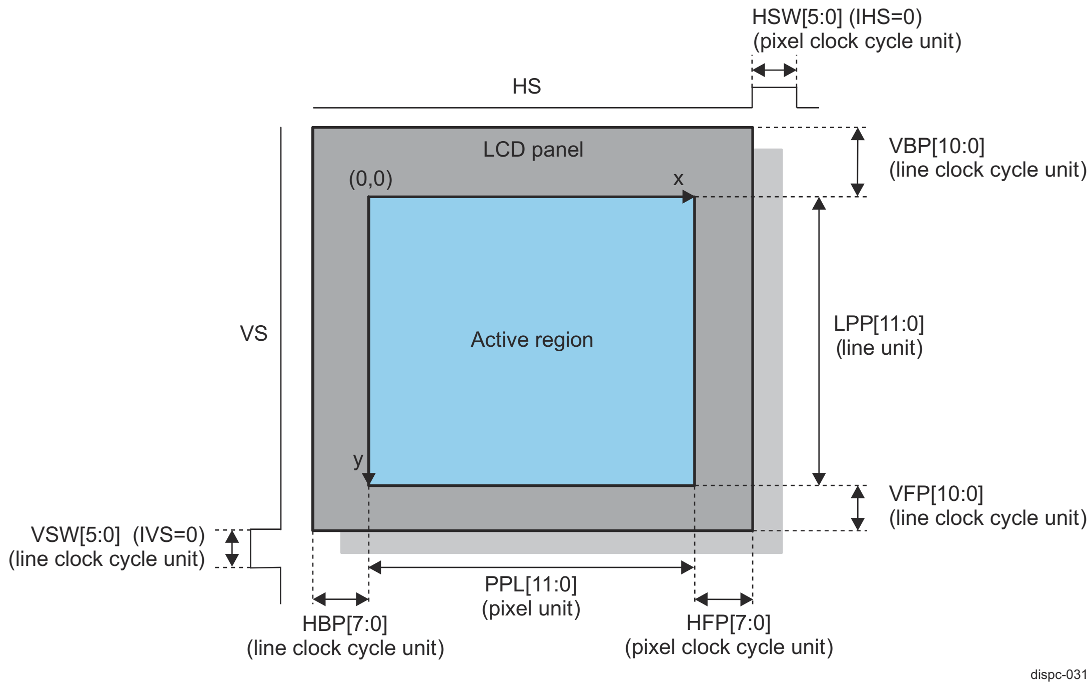


Figure 11-92. DISPC Timing Values (Active Matrix Display)

The 8-bit pixel clock divider (the DISPC_DIVISORo[7:0] PCD bit field) selects the pixel clock frequency. This bit field generates a range of pixel clock frequencies from LC/2 to LC/256, where LC is the logic clock from the divided functional clock of the DISPC by the DISPC_DIVISOR[23:16] LCD bit field.

The pixel clock is defined by the following equation:

Pixel Clock = (FunctionalClock/LCD[7:0])/PCD[7:0]

The pixel clock can be gated by setting the DISPC_CONFIGo[5] PIXELCLOCKGATED bit to 0x1.

The LCD output can be configured in progressive output or interlace output. The selection is done by writing into the DISPC_CONFIGo[22] OUTPUTMODEENABLE bit. The reset value is 0x0, which means progressive mode. When progressive mode is selected, the FID signal associated to the LCD output is driven low (INACTIVE state). The selection can be changed only if the corresponding LCD output is disabled. The configuration is independent for each LCD output.

When in interlaced mode, the DISPC_CONFIGo[23] FIDFIRST bit indicates which field is output first:

- 0x0: Even field first (FID = 0)
- 0x1: Odd field first (FID = 1)

11.2.4.14 DISPC TV Output

The TV output paths consist of several processing blocks (see Figure 11-93):

- Overlay manager
- Gamma correction unit
- Synchronization buffer

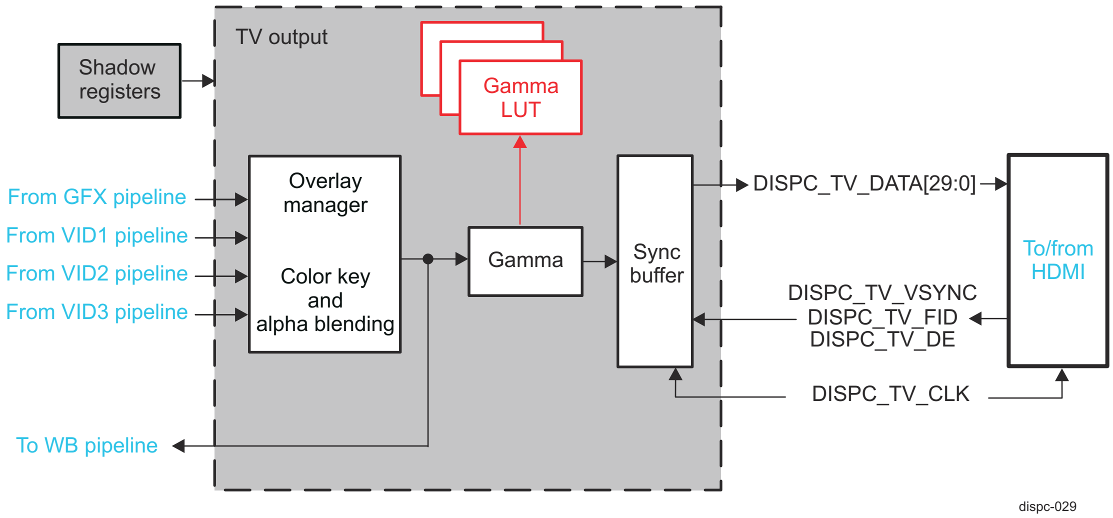


Figure 11-93. DISPC TV Output Architecture

The TV output is connected to the HDMI. In analog output or HDMI, the DISPC TV output receives an external clock, DISPC_TV_CLK, and based on the VSYNC generated by the HDMI, hold time, vertical offset, and horizontal offset, outputs the pixels synchronously. The size of the field/frame to output defines the number of pixels to output on each line and the number of lines for each field/frame.

11.2.4.14.1 DISPC Overlay Manager

The overlay mechanism is identical in LCD1, LCD2, and LCD3 (see Section 11.2.4.13.1, *Overlay Manager*).

11.2.4.14.2 DISPC Gamma Correction Unit

The gamma correction unit works as described in Section 11.2.4.13.2, *Gamma Correction Unit*. The only difference is in the input pixel format RGB30. Each component of encoded pixel value is used as a pointer to index 1 out of 1024 × 30-bit gamma curve entries in the table. Each 10-bit component is replaced with the 10-bit table value corresponding to R, G, or B component. The table is loaded by software, through the DISPC_GAMMA_TABLE2 register, and enabled via DISPC_CONFIG1[9] GAMATABLEENABLE register bit. It is possible to load only part of the table. For each write access to the table, the 30-bit gamma value is associated with bit [31] INDEX to indicate that a new table is defined. Setting bit [31] INDEX to 0x1 for the first time, resets the internal index counter. All the following accesses are considered to be for incremented index addressing in the table (bit [31] INDEX is set to 0 for each subsequent access). Figure 11-94 shows the format of one of the gamma curve values in the memory.

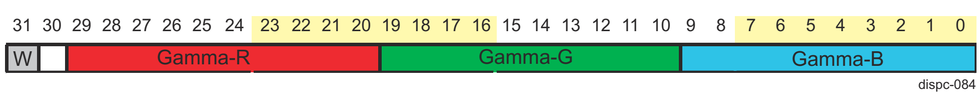


Figure 11-94. DISPC Data Memory Organization for Gamma Mode in TV Output

11.2.4.14.3 DISPC Synchronized Buffer Update

The synchronized buffer update is identical in LCD1, LCD2, and LCD3 (see Section 11.2.4.13.7, *Synchronized Buffer Update*).

11.2.4.14.4 DISPC Timing and TV Format Settings

Figure 11-95 shows the TV formats supported.

Figure 11-95. DISPC Example Timing TV Formats

The size of a TV output format is defined by:

- Number of lines, DISPC_SIZE_TV[27:16] LPP bit field, with a value from 1 to 4096
- Number of pixels per line, DISPC_SIZE_TV[11:0] PPL bit field, with a value from 1 to 4096
- Delta size between odd/even field, DISPC_SIZE_TV[15:14] DELTA_LPP bit field. This bit field controls only the output channel and not the size of the data field fetched from the frame buffer in memory.

The hold time of the pixels on the data bus is determined in clock cycles by the DISPC_CONTROL1[19:17] HT bit field. The default value at reset time is 0x0 (one cycle).

- When connected to the HDMI encoder, Table 11-90 indicates the DISPC_SIZE_TV (PPL and LPP) value for each HD standard.

Table 11-90. DISPC PPL and LLP Value for HD Standard

Standards		Active Pixels/Line	Active Lines	Digital Clock	DISPC_SIZE_TV
HDTV	720p	1280	720	74.25/74.125 MHz (60/59.99..frames/s)	0x02CF 04FF
	1080i	1920	540	74.25/74.125 MHz (60/59.99..frames/s)	0x021B 07FF
	1080p	1920	1080	148.5/148.25 MHz (60/59.99..frames/s)	0x0437 07FF
	720p 3D (frame packing)	1280	1470	148.5/148.35/148.5 MHz (60/59.94/50 frames/s)	0x05BD 04FF
	1080p 3D (frame packing)	1920	2205	148.5/148.35 MHz (24/23.98 frames/s)	0x089C 07FF
	1080p 3D (side-by-side half)	1920	1080	148.5 MHz (60 frames/s)	0x0437 07FF

Note

When configuring the DISPC for outputting any supported 3D frame-packing format, the following generic details must be considered:

- As defined by the *HDMI v1.4a Specification*, Section 8.2.3.2, 3D frame-packing is a video format structure composed of two stereoscopic pictures: left and right.
- The stereoscopic pictures can be processed through the three video pipelines (VID1, VID2, or VID3). For more information about the configuration of video pipelines, see Section 11.2.4.10, *Video Pipelines*.
- The 3D frame is generated by setting the TV overlay manager to combine the outputs of the selected video pipelines that hold the pictures. One video pipeline (VID1 or VID3) must carry the top field of the 3D frame (left stereoscopic picture), and the other video (VID2) pipeline must always carry the bottom field of the frame (right stereoscopic picture). The top and bottom fields are separated by an active space area. For more information, see the *HDMI v1.4a Specification*, Section 8.2.3.2.
- The pipeline carrying each field (top and bottom) of the 3D frame must have its height and width parameters defined in the [27:16] SIZEY and [10:0] SIZEX bit fields of the DISPC_VIDp_SIZE register, and its Y and X positions defined in the [26:16] POSY and [10:0] POSX bit fields of the DISPC_VIDp_POSITION register. The active space area of the 3D frame can be encoded by setting the solid background color for the TV output (the DISPC_DEFAULT_COLOR1[23:0] DEFAULTCOLOR bit field). For more information about the overlay mechanism, see Section 11.2.4.14.1, *DISPC Overlay Manager*.

11.2.4.15 DISPC Frame Width Considerations

DISPC provides three sets of configuration registers to control the maximum frame size in different stages of its internal pipelines (see Figure 11-96, *DISPC Frame Width Control*):

- DISPC_VID#_PICTURE_SIZE registers: MEMSIZEX and MEMSIZEY bit-fields define the number of pixels and lines to fetch from memory (via DISPC DMA engine).
- DISPC_GFX/VID#_SIZE registers: SIZEX and SIZEY bit-fields define the size of the output frame from each processing pipeline to LCD/TV overlay manager for blending.
- DISPC_SIZE_TV/LCD# registers: LPP and PPL bit-fields define the final display panel output resolution.

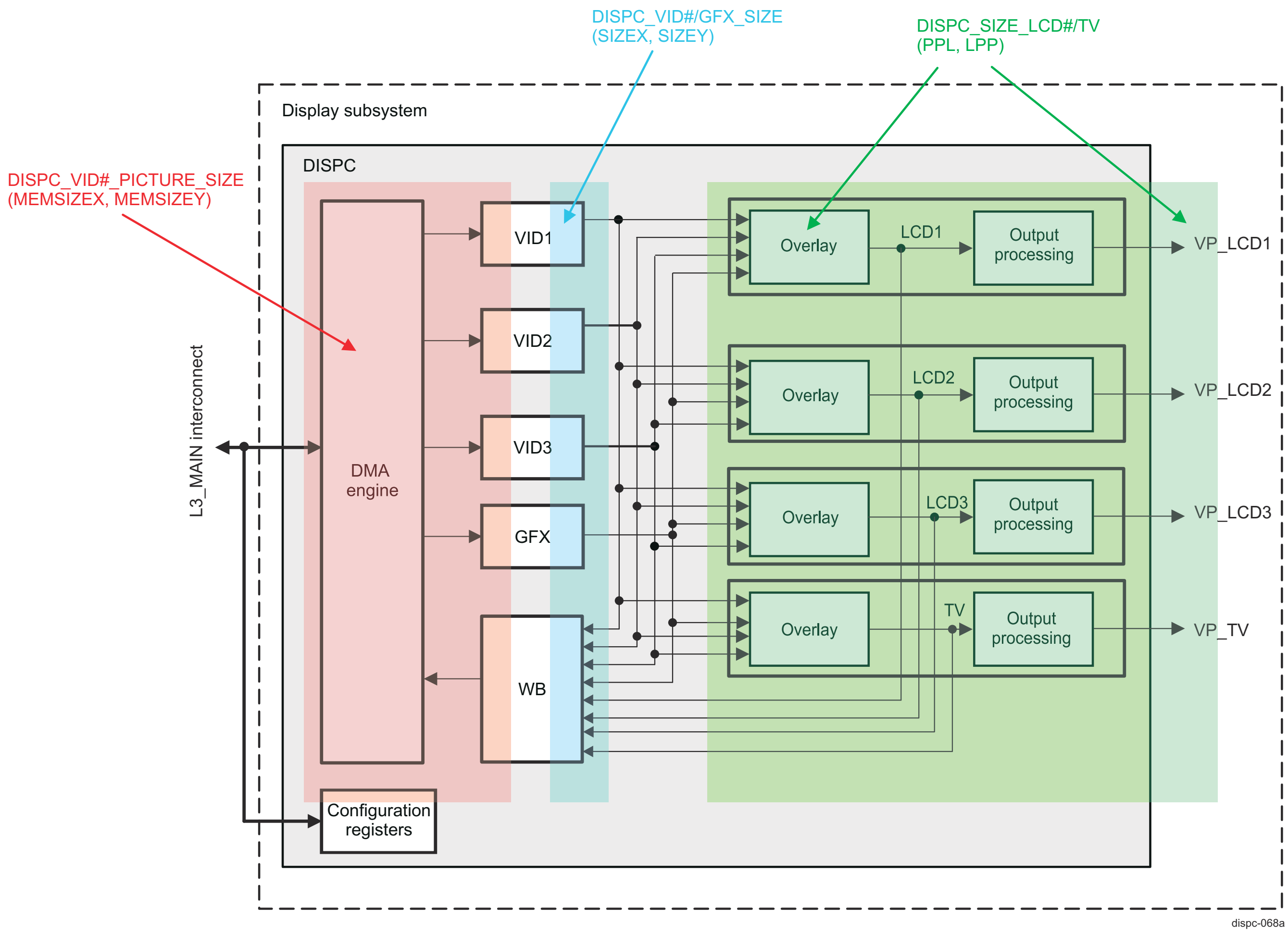


Figure 11-96. DISPC Frame Width Control

The MEMSIZEX and SIZEX bit-fields are 11-bit, meaning the maximum horizontal frame size allowed is 2048 pixels. This parameter determines the maximum source frame (including the input to the scalers) that can be displayed.

The MEMSIZEY and SIZEY bit-fields are 12-bit, meaning the maximum vertical size allowed is 4096 lines.

The LPP and PPL bit-fields are also 12-bit, which means that starting at the LCD/TV overlay manager and including the output to the display panel, the frame size can be up to 4096x4096 as long as the pixel clock constraints are met.

Implication: The output to display panel can be as large as 4096 x 4096 (4K x 4K), although the actual size of the output frame would be smaller due to the maximum pixel clock limitation. However, the maximum source frame that can be read from memory by a single GFX/VID pipeline can be 2048 x 4096 (2K x 4K) only. Therefore, to display a source frame with width > 2048 pixels, two layers must be used and corresponding pipeline outputs must be merged in the overlay manager before being sent out to the display panel.

Refer to device Data Manual for details on the maximum supported pixel clock (VOUTx_CLK) or HDMI supported frame rates.

11.2.4.16 DISPC Extended 3D Support

The DISPC supports several formats to support 3D displays. The following three sections provide further details.

11.2.4.16.1 DISPC Extended 3D Support - Line Alternative Format

Line alternative format is defined by the HDMI Specification 1.4a. In general the functionality is:

- Lines from the left and the right frames are interleaved alternatively on the screen to produce a 3D image.
- Required also to support interleaving column-wise to support all possible screen orientations.

The functionality is done in the overlay manager:

- The layers are grouped into two categories based on the z-order: Odd z-order may blend together and will interleave with even z-order pipes, which may blend together.
- Two types of interleaves are possible, line-based interleaving and pixel-based interleaving to support landscape and portrait screen orientations.

The configuration is done from the DISPC_CONFIG1[29:28] TVINTERLEAVE for the TV output, DISPC_CONFIG1[27:26] PLCDINTERLEAVE for the primary LCD, DISPC_CONFIG2[27:26] SLCDINTERLEAVE for the secondary LCD, and DISPC_CONFIG3[27:26] TLCDINTERLEAVE for the third LCD. The following values apply for the these bitfields:

- 0x0: If zero then no interleaving happens in the overlay manager.
- 0x1: Checkerboard pattern:
  - All even pixels on even lines have a contribution from even z-order pipes.
  - All odd pixels on even lines have a contribution from odd z-order pipes.
  - All even pixels on odd lines have a contribution from odd z-order pipes.
  - All odd pixels on odd lines have a contribution from even z-order pipes.
- 0x2: All even lines (all pixels) have a contribution from even z-order pipes. All odd lines (all pixels) have a contribution from the odd z-order pipes.
- 0x3: All even pixels (for all lines) have a contribution from even z-order pipes. All odd pixels (for all lines) have a contribution from the odd z-order pipes.

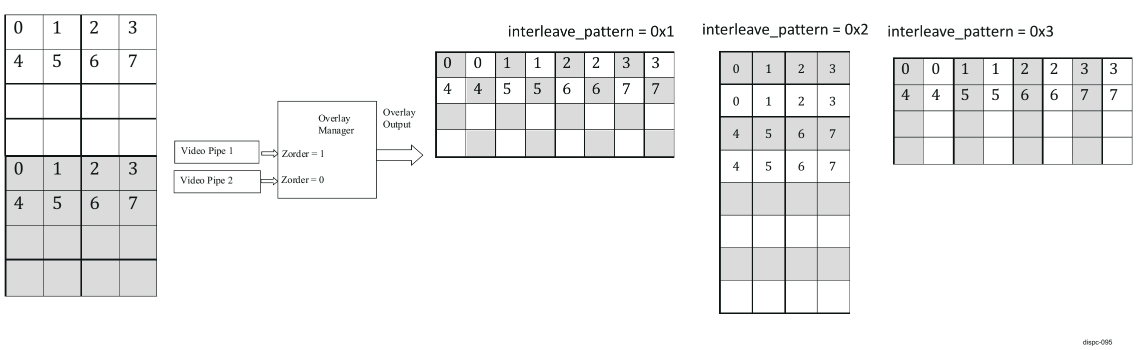


Figure 11-97. DISPC Illustration of 3D Interleaving

11.2.4.16.2

Moreover, from Figure 11-97 above, if interleave_pattern is programmed as 0x2, in this case the entire screen composited by the overlay manager (PPL x LPP) gets sub-divided into odd and even lines. It is possible for a 3D window created using the two pipes to have a non-zero position (that is, POSX and POSY not equal to zero).

In such cases depending on the oddness or evenness of the POSY setting, the first line in the window may come from Video Pipe 1 or Video Pipe 2.

For instance, if the POSX = 3, POSY = 3 (set in DISPC_GFX_POSITION, DISPC_VID1_POSITION, DISPC_VID2_POSITION, DISPC_VID3_POSITION registers), then the final composited screen with PPL = 8, LPP = 12 may look like below:

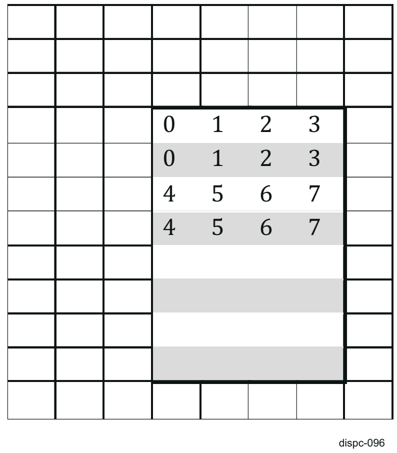


Figure 11-98. DISPC Illustration of a Non-zero Position of 3D Window

---

Note

In this case the first line comes from the odd z-order pipeline, that is Video Pipe 1.

The following limitations apply to the 3D Line alternative format:

- Software must ensure that the left/right frames are of same size.
- Software must ensure that in case of line interleaving the SIZEY is not greater than LPP/2.
- Software must ensure that in case of pixel interleaving the SIZEX is not greater than PPL/2.
- Software must program POSX and POSY for both left and right frame to be the same.
- Color-keying is not available for 3D formats.
- Since each overlay manager can support 4k x 4k frames, max resolution of each left/right frame will be restricted to 2k x 2k.

11.2.4.16.3 DISPC Extended 3D Support - Frame Packing Format Format

In this mode the pipe fetches two windows (of SIZEX x SIZEY) during a complete panel frame (PPL x LPP). The mode is enabled from DISPC_GFX_ATTRIBUTES[10] FRAMEPACKINGMODE and DISPC_VID1_ATTRIBUTES[8], DISPC_VID2_ATTRIBUTES[8], DISPC_VID3_ATTRIBUTES[8] FRAMEPACKINGMODE

The position for the first window is set in DISPC_GFX_POSITION, DISPC_VID1_POSITION, DISPC_VID2_POSITION, DISPC_VID3_POSITION, while the position for the second window is set in DISPC_GFX_POSITION2, DISPC_VID1_POSITION2, DISPC_VID2_POSITION2, DISPC_VID3_POSITION2

When it is detected that a pipe has finished sending the programmed number of lines to the overlay or Write-back pipe, a reset is generated for the pipe as well as the DMA channel associated with the pipe. At the same instant, the base address configuration for the DMA channel is changed from the BA0 to the BA1.

Due to the reset the pipe and DMA starts fetching the data from BA1. At the end of the processing of BA1 data, the DMA channel and the pipe become idle and clock gated.

The overlay manager composes the final frame to be sent to the panel and uses the (POSX, POSY) coordinates for the first window and (POSX2, POSY2) for the second window.

The following limitations apply to the 3D Frame Packing format:

- Only frame packing for progressive mode is supported. Frame packing of interlaced formats is not supported.
- End-of-window interrupts will be generated twice per outgoing-frame.
- Flip Immediate will not work as expected.
- It is software responsibility to calculate and program the (POSX2, POSY2) as per the active space required by the panel.

11.2.4.16.4 DISPC Extended 3D Support - DLP 3D Format

The DLP 3D format sub-samples the left and right frames across the diagonal before interleaving the frames in a checker-board fashion.

The sub-sampling required on the left and the right frame is complementary. This sampling pattern is also defined in the HDMI 3D Specification 1.4a as two Quincunx matrices. The format is configured from DISPC_GFX_ATTRIBUTES[20:18] SUBSAMPLINGPATTERN, DISPC_VID1_ATTRIBUTES2[11:9], DISPC_VID2_ATTRIBUTES2[11:9], DISPC_VID3_ATTRIBUTES2[11:9] SUBSAMPLINGPATTERN with the following possible configuration values:

- 0x0: All pixels from the pipes are used. No sub-sampling.
- 0x1: Quincunx Matrix Odd-position sub-sampling is used.
- 0x2: Quincunx Matrix Even-position sub-sampling is used.
- 0x3 to 0x7: Reserved

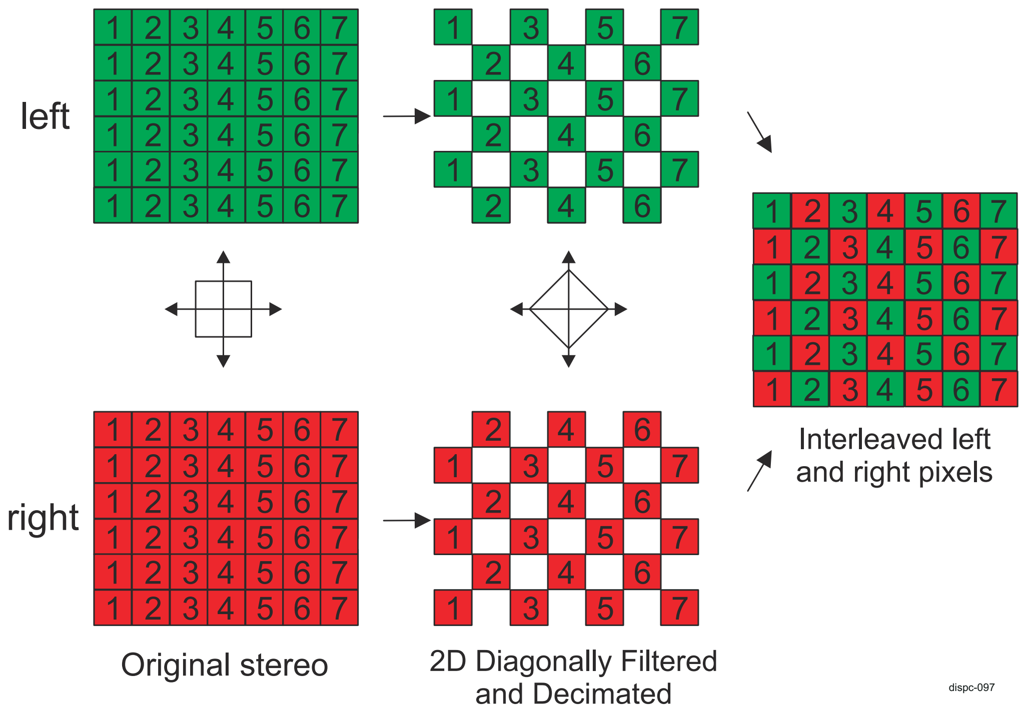


Figure 11-99. DISPC Illustration of DLP 3D Format

The following limitations apply to the DLP 3D Format :

- Even z-order pipes can have only sub-sampling = 0x1.
- Odd z-order pipes can only have sub-sampling = 0x2.

11.2.4.17 DISPC Shadow Registers

Some DISPC registers are termed *shadow registers*. A shadow register change has no direct effect on the configuration of the DISPC. The registers are shadow registers let software change the values of the registers at any time. When all the registers for a given configuration are into the registers, software must set 1 bit only to validate the configuration. When hardware reaches the end of the current frame and sees that the bit field has been set by software, the new configuration is now the configuration used by the hardware.

Note

As a general rule, all shadow registers are updated with the value of their shadows when:

- The interface is enabled.
- The GO bit field of the pipeline, with which the register is associated, is set and the output sync is active.

The bits enabling the hardware to use the new configuration are:

- DISPC_CONTROL1[5] GOLCD bit for all the registers associated to the LCD1 output, and for all registers of WB and DMA, if the LCD1 channel is captured. The update of the registers of the WB and DMA is further delayed by the DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNTER bit field or done when the next frame is captured (the DISPC_WB_ATTRIBUTES[26:24] CAPTUREMODE bit field must be different from 0). The update of the registers occurs at the VFP start period.
- DISPC_CONTROL1[6] GOTV bit for all the registers associated to the TV output, and for all registers of WB and DMA, if the TV channel is captured. The update of the registers of the WB and DMA is further delayed by the DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNTER bit field or done when the next frame is captured (the DISPC_WB_ATTRIBUTES[26:24] CAPTUREMODE bit field must be different from 0). The update of the registers occurs at the external EVSYNC.
- DISPC_CONTROL2[5] GOLCD bit for all the registers associated to the LCD2 output, and for all registers of WB and DMA, if the LCD2 channel is captured. The update of the registers of the WB and DMA is further delayed by the DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNTER bit field or done when the next frame is captured (the DISPC_WB_ATTRIBUTES[26:24] CAPTUREMODE bit field must be different from 0). The update of the registers occurs at the VFP start period.
- DISPC_CONTROL3[5] GOLCD bit for all the registers associated to the LCD3 output, and for all registers of WB and DMA, if the LCD3 channel is captured. The update of the registers of the WB and DMA is further delayed by the DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNTER bit field or done when the next frame is captured (the DISPC_WB_ATTRIBUTES[26:24] CAPTUREMODE bit field must be different from 0). The update of the registers occurs at the VFP start period.
- DISPC_WB_ATTRIBUTES[0] ENABLE bit for all the registers associated to the WB, if the transfer memory-to-memory is not associated with a channel out.
- The DISPC_CONTROL2[6] GOWB bit and the [5] GOLCD and [6] GOTV bits in the DISPC_CONTROL1/ DISPC_CONTROL2 registers, combined with the synchronization event of the channel output selected for write back. This applies to all registers associated with the selected channel out and further delayed by the setting in the DISPC_WB_ATTRIBUTES2[7:0] WBDELAYCOUNT bit field, for all registers of the write back and DMA. The GOWB bit is required to be set only in WB capture mode; it is not required when WB memory-to-memory mode is used.

Note

Before setting the GOLCD, GOTV, or GOWB bits, the user must ensure that the bits are cleared. The hardware resets the bits when the update completes.

Table 11-91 lists the shadow registers. Registers that do not have a mark in any column are not shadowed.

Table 11-91. DISPC Shadow Registers

Shadow Register Name	Updated on VFP Start Period (LCD1 Pipeline)	Updated on VFP Start Period (LCD2 Pipeline)	Updated on VFP Start Period (LCD3 Pipeline)	Updated on External VSYNC (TV Pipeline)	Updated on END of Frame (WB Pipeline)
DISPC_REVISION					
DISPC_SYSCONFIG					
DISPC_SYSSTATUS					
DISPC_IRQSTATUS					

Table 11-91. DISPC Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD1 Pipeline)	Updated on VFP Start Period (LCD2 Pipeline)	Updated on VFP Start Period (LCD3 Pipeline)	Updated on External VSYNC (TV Pipeline)	Updated on END of Frame (WB Pipeline)
DISPC_IRQENABLE					
DISPC_CONTROL1	x			x	
DISPC_CONFIG1	x	x	x	x	x
DISPC_DEFAULT_COLOR0	x				
DISPC_DEFAULT_COLOR1				x	
DISPC_TRANS_COLOR0	x				
DISPC_TRANS_COLOR1				x	
DISPC_LINE_STATUS					
DISPC_LINE_NUMBER	x				
DISPC_TIMING_H1	x				
DISPC_TIMING_V1	x				
DISPC_POL_FREQ1	x				
DISPC_DIVISOR1	x				
DISPC_GLOBAL_ALPHA	x	x	x	x	x
DISPC_SIZE_TV				x	
DISPC_SIZE_LCD1	x				
DISPC_GFX_BA_j (2)	x	x	x	x	x
DISPC_GFX_POSITION	x	x	x	x	x
DISPC_GFX_SIZE	x	x	x	x	x
DISPC_GFX_ATTRIBUTES	x	x	x	x	x
DISPC_GFX_BUF_THRESHOLD	x	x	x	x	x
DISPC_GFX_BUF_SIZE_STATUS					
DISPC_GFX_ROW_INC	x	x	x	x	x
DISPC_GFX_PIXEL_INC	x	x	x	x	x
DISPC_GFX_TABLE_BA	x	x	x	x	x
DISPC_VID1_BA_j (2)	x	x	x	x	x
DISPC_VID1_POSITION	x	x	x	x	x
DISPC_VID1_SIZE	x	x	x	x	x
DISPC_VID1_ATTRIBUTES	x	x	x	x	x
DISPC_VID1_BUF_THRESHOLD	x	x	x	x	x
DISPC_VID1_BUF_SIZE_STATUS					
DISPC_VID1_ROW_INC	x	x	x	x	x
DISPC_VID1_PIXEL_INC	x	x	x	x	x
DISPC_VID1_FIR	x	x	x	x	x
DISPC_VID1_PICTURE_SIZE	x	x	x	x	x
DISPC_VID1_ACCU_j (2)	x	x	x	x	x
DISPC_VID1_FIR_COEF_H_i (1)	x	x	x	x	x
DISPC_VID1_FIR_COEF_HV_i (1)	x	x	x	x	x
DISPC_VID1_CONV_COEF0	x	x	x	x	x

Table 11-91. DISPC Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD1 Pipeline)	Updated on VFP Start Period (LCD2 Pipeline)	Updated on VFP Start Period (LCD3 Pipeline)	Updated on External VSYNC (TV Pipeline)	Updated on END of Frame (WB Pipeline)
DISPC_VID1_CONV_COEF1	x	x	x	x	x
DISPC_VID1_CONV_COEF2	x	x	x	x	x
DISPC_VID1_CONV_COEF3	x	x	x	x	x
DISPC_VID1_CONV_COEF4	x	x	x	x	x
DISPC_VID2_BA_j (2)	x	x	x	x	x
DISPC_VID2_POSITION	x	x	x	x	x
DISPC_VID2_SIZE	x	x	x	x	x
DISPC_VID2_ATTRIBUTES	x	x	x	x	x
DISPC_VID2_BUF_THRESHOLD	x	x	x	x	x
DISPC_VID2_BUF_SIZE_STATUS					
DISPC_VID2_ROW_INC	x	x	x	x	x
DISPC_VID2_PIXEL_INC	x	x	x	x	x
DISPC_VID2_FIR	x	x	x	x	x
DISPC_VID2_PICTURE_SIZE	x	x	x	x	x
DISPC_VID2_ACCU_j (2)	x	x	x	x	x
DISPC_VID2_FIR_COEF_H_i (1)	x	x	x	x	x
DISPC_VID2_FIR_COEF_HV_i (1)	x	x	x	x	x
DISPC_VID2_CONV_COEF0	x	x	x	x	x
DISPC_VID2_CONV_COEF1	x	x	x	x	x
DISPC_VID2_CONV_COEF2	x	x	x	x	x
DISPC_VID2_CONV_COEF3	x	x	x	x	x
DISPC_VID2_CONV_COEF4	x	x	x	x	x
DISPC_DATA1_CYCLE1	x				
DISPC_DATA1_CYCLE2	x				
DISPC_DATA1_CYCLE3	x				
DISPC_VID1_FIR_COEF_V_i (1)	x	x	x	x	x
DISPC_VID2_FIR_COEF_V_i (1)	x	x	x	x	x
DISPC_CPR1_COEF_R	x				
DISPC_CPR1_COEF_G	x				
DISPC_CPR1_COEF_B	x				
DISPC_GFX_PRELOAD	x	x	x	x	x
DISPC_VID1_PRELOAD	x	x	x	x	x
DISPC_VID2_PRELOAD	x	x	x	x	x
DISPC_CONTROL2		x		x	x
DISPC_VID3_ACCU_j (1)	x	x	x	x	x
DISPC_VID3_BA_j (1)	x	x	x	x	x
DISPC_VID3_FIR_COEF_H_i (1)	x	x	x	x	x
DISPC_VID3_FIR_COEF_HV_i (1)	x	x	x	x	x

Table 11-91. DISPC Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD1 Pipeline)	Updated on VFP Start Period (LCD2 Pipeline)	Updated on VFP Start Period (LCD3 Pipeline)	Updated on External VSYNC (TV Pipeline)	Updated on END of Frame (WB Pipeline)
DISPC_VID3_FIR_COEF_V_i (1)	x	x	x	x	x
DISPC_VID3_ATTRIBUTES	x	x	x	x	x
DISPC_VID3_CONV_COEF0	x	x	x	x	x
DISPC_VID3_CONV_COEF1	x	x	x	x	x
DISPC_VID3_CONV_COEF2	x	x	x	x	x
DISPC_VID3_CONV_COEF3	x	x	x	x	x
DISPC_VID3_CONV_COEF4	x	x	x	x	x
DISPC_VID3_BUF_SIZE_STATUS					
DISPC_VID3_BUF_THRESHOLD	x	x	x	x	x
DISPC_VID3_FIR	x	x	x	x	x
DISPC_VID3_PICTURE_SIZE	x	x	x	x	x
DISPC_VID3_PIXEL_INC	x	x	x	x	x
DISPC_VID3_POSITION	x	x	x	x	x
DISPC_VID3_PRELOAD	x	x	x	x	x
DISPC_VID3_ROW_INC	x	x	x	x	x
DISPC_VID3_SIZE	x	x	x	x	x
DISPC_DEFAULT_COLOR2		x			
DISPC_TRANS_COLOR2		x			
DISPC_CPR2_COEF_B		x			
DISPC_CPR2_COEF_G		x			
DISPC_CPR2_COEF_R		x			
DISPC_DATA2_CYCLE1		x			
DISPC_DATA2_CYCLE2		x			
DISPC_DATA2_CYCLE3		x			
DISPC_SIZE_LCD2		x			
DISPC_TIMING_H2		x			
DISPC_TIMING_V2		x			
DISPC_POL_FREQ2		x			
DISPC_DIVISOR2		x			
DISPC_WB_ACCU_j (2)	x	x	x	x	x
DISPC_WB_BA_j (2)	x	x	x	x	x
DISPC_WB_FIR_COEF_H_i (1)	x	x	x	x	x
DISPC_WB_FIR_COEF_HV_i (1)	x	x	x	x	x
DISPC_WB_FIR_COEF_V_i (1)	x	x	x	x	x
DISPC_WB_ATTRIBUTES	x	x	x	x	x
DISPC_WB_CONV_COEF0	x	x	x	x	x
DISPC_WB_CONV_COEF1	x	x	x	x	x
DISPC_WB_CONV_COEF2	x	x	x	x	x
DISPC_WB_CONV_COEF3	x	x	x	x	x

Table 11-91. DISPC Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD1 Pipeline)	Updated on VFP Start Period (LCD2 Pipeline)	Updated on VFP Start Period (LCD3 Pipeline)	Updated on External VSYNC (TV Pipeline)	Updated on END of Frame (WB Pipeline)
DISPC_WB_CONV_COEF4	x	x	x	x	x
DISPC_WB_BUF_SIZE_STATUS	x	x	x	x	x
DISPC_WB_BUF_THRESHOLD	x	x	x	x	x
DISPC_WB_FIR	x	x	x	x	x
DISPC_WB_PICTURE_SIZE	x	x	x	x	x
DISPC_WB_PIXEL_INC	x	x	x	x	x
DISPC_WB_ROW_INC	x	x	x	x	x
DISPC_WB_SIZE	x	x	x	x	x
DISPC_VID1_BA_UV_j (2)	x	x	x	x	x
DISPC_VID2_BA_UV_j (2)	x	x	x	x	x
DISPC_VID3_BA_UV_j (2)	x	x	x	x	x
DISPC_WB_BA_UV_j (2)	x	x	x	x	x
DISPC_CONFIG2	x	x	x	x	x
DISPC_VID1_ATTRIBUTES2	x	x	x	x	x
DISPC_VID2_ATTRIBUTES2	x	x	x	x	x
DISPC_VID3_ATTRIBUTES2	x	x	x	x	x
DISPC_GAMMA_TABLE0					
DISPC_GAMMA_TABLE1					
DISPC_GAMMA_TABLE2					
DISPC_VID1_FIR2	x	x	x	x	x
DISPC_VID1_ACCU2_j (2)	x	x	x	x	x
DISPC_VID1_FIR_COEF_H2_i (1)	x	x	x	x	x
DISPC_VID1_FIR_COEF_HV2_i (1)	x	x	x	x	x
DISPC_VID1_FIR_COEF_V2_i (1)	x	x	x	x	x
DISPC_VID2_FIR2	x	x	x	x	x
DISPC_VID2_ACCU2_j (2)	x	x	x	x	x
DISPC_VID2_FIR_COEF_H2_i (1)	x	x	x	x	x
DISPC_VID2_FIR_COEF_HV2_i (1)	x	x	x	x	x
DISPC_VID2_FIR_COEF_V2_i (1)	x	x	x	x	x
DISPC_VID3_FIR2	x	x	x	x	x
DISPC_VID3_ACCU2_j (2)	x	x	x	x	x
DISPC_VID3_FIR_COEF_H2_i (1)	x	x	x	x	x
DISPC_VID3_FIR_COEF_HV2_i (1)	x	x	x	x	x
DISPC_VID3_FIR_COEF_V2_i (1)	x	x	x	x	x
DISPC_WB_FIR2	x	x	x	x	x
DISPC_WB_ACCU2_j (2)	x	x	x	x	x

Table 11-91. DISPC Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD1 Pipeline)	Updated on VFP Start Period (LCD2 Pipeline)	Updated on VFP Start Period (LCD3 Pipeline)	Updated on External VSYNC (TV Pipeline)	Updated on END of Frame (WB Pipeline)
DISPC_WB_FIR_COEF_H2_i (1)	x	x	x	x	x
DISPC_WB_FIR_COEF_HV2_i (1)	x	x	x	x	x
DISPC_WB_FIR_COEF_V2_i (1)	x	x	x	x	x
DISPC_GLOBAL_BUFFER					
DISPC_DIVISOR					
DISPC_WB_ATTRIBUTES2	x	x	x	x	x
DISPC_DEFAULT_COLOR3			x		
DISPC_TRANS_COLOR3			x		
DISPC_CPR3_COEF_B			x		
DISPC_CPR3_COEF_G			x		
DISPC_CPR3_COEF_R			x		
DISPC_DATA3_CYCLE1			x		
DISPC_DATA3_CYCLE2			x		
DISPC_DATA3_CYCLE3			x		
DISPC_SIZE_LCD3			x		
DISPC_DIVISOR3			x		
DISPC_POL_FREQ3			x		
DISPC_TIMING_H3			x		
DISPC_TIMING_V3			x		
DISPC_CONTROL3			x		
DISPC_CONFIG3			x		
DISPC_GAMMA_TABLE3			x		
DISPC_BA0_FLIPIMMEDIATE_EN					
DISPC_GLOBAL_MFLAG_ATTRIBUTE					
DISPC_GFX_MFLAG_THRESHOLD	x	x	x	x	x
DISPC_VID1_MFLAG_THRESHOLD	x	x	x	x	x
DISPC_VID2_MFLAG_THRESHOLD	x	x	x	x	x
DISPC_VID3_MFLAG_THRESHOLD	x	x	x	x	x
DISPC_WB_MFLAG_THRESHOLD	x	x	x	x	x
DISPC_GFX_POSITION2	x	x	x	x	x
DISPC_VID1_POSITION2	x	x	x	x	x
DISPC_VID2_POSITION2	x	x	x	x	x
DISPC_VID3_POSITION2	x	x	x	x	x

(1) i = 0 to 7
(2) j = 0 to 1

11.2.5 DISPC Programming Guide

11.2.5.1 DISPC Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

11.2.5.1.1 DISPC Global Initialization

11.2.5.1.1.1 DISPC Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the DISPC module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DISPC. For more information, see *DISPC Integration*, and *DISPC Environment*.

Table 11-92. DISPC Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing and configuration must be set in the control module. See *Pad Configuration Registers*, in *Control Module*.
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see *Interrupt Controllers*.
Interconnect	For more information about interconnect configuration, see *L3 Interconnect.*
DMA_CROSSBAR	DMA_CROSSBAR configuration must be done to allow module DREQs to be mapped to certain device DMA line. For more information see *DMA_CROSSBAR Module Functional Description*, in *Control Module.*
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *, IRQ_CROSSBAR Module Functional Description,* in *Control Module.*

11.2.5.1.2 DISPC Operational Modes Configuration

Table 11-93 lists the steps to configure the operational modes in the DISPC.

Table 11-93. DISPC Configuration

Step	Register/Bit Field/Programming Model
For a GFX pipeline configuration	
Configure the GFX DMA channel.	See Table 11-94.
Configure the GFX pipeline.	See Table 11-97.
Configure the LCD or TV output.	For LCD output, see Table 11-114.
	For TV output, see Table 11-123.
For a Video pipeline configuration	
Configure the video DMA channel.	See Table 11-95.
Configure the video pipeline.	See Table 11-101.
Configure the LCD or TV output.	For LCD output, see Table 11-114.
	For TV output, see Table 11-123.
For a WB pipeline configuration	
Configure the WB DMA channel.	See Table 11-96.
Configure the WB pipeline.	For video pipelines, see Table 11-109.

11.2.5.1.2.1 DISPC DMA Configuration

11.2.5.1.2.1.1 DISPC Main Sequence – DISPC DMA Channel Configuration

This procedure describes the parameters of the GFX, video, and WB DMA channel parameters (see Table 11-94 through Table 11-96, respectively).

Table 11-94. DISPC Configure the GFX DMA Channel

Step	Register/Bit Field/Programming Model	Value
Set the base address for RGB pixel format according to memory access type, rotation, mirroring (see Section 11.2.4.6, *DISPC DMA Engine*).	DISPC_GFX_BA_j[31:0] BA	0x—
Set the rotation flag.	DISPC_GFX_ATTRIBUTES[13:12] ROTATION	0x–
Set the number of bytes to increment at the end of the row.	DISPC_GFX_ROW_INC[31:0] ROWINC	0x–
Set the number of bytes to increment between two pixels.	DISPC_GFX_PIXEL_INC[7:0] PIXELINC	0x–
Determine the FIFO preload mode.	DISPC_GFX_ATTRIBUTES[11] BUFPRELOAD	0x–
Set the preload value.	DISPC_GFX_PRELOAD[11:0] PRELOAD	0x–
Determine the burst type.	DISPC_GFX_ATTRIBUTES[29] BURSTTYPE	0x–
Set the burst size.	DISPC_GFX_ATTRIBUTES[7:6] BURSTSIZE	0x–
Set the high level of DMA FIFO threshold.	DISPC_GFX_BUF_THRESHOLD[31:16] BUFHIGHTRESHOLD	0x–
Set the low level of DMA FIFO threshold.	DISPC_GFX_BUF_THRESHOLD[15:0] BUFLOWTRESHOLD	0x–
Enable self-refresh.	DISPC_GFX_ATTRIBUTES[24] SELFREFRESH	0x–
Select priority over the other pipeline.	DISPC_GFX_ATTRIBUTES[23] ARBITRATION	0x–

(1) Applicable only for YUV pixel format

Table 11-95. DISPC Configure the Video DMA Channel

Step	Register/Bit Field/Programming Model	Value
Set the base address for RGB pixel format or Y component format according to memory access type, rotation, mirroring (seeSection 11.2.4.6, *DISPC DMA Engine*).	DISPC_VIDp_BA_j[31:0] BA	0x—
Set the base address for Cb and Cr components according to memory access type, rotation, mirroring (seeSection 11.2.4.6, *DISPC DMA Engine*)[(1)].	DISPC_VIDp_BA_UV_j[31:0] BA	0x—
Set the rotation flag.	DISPC_VIDp_ATTRIBUTES[13:12] ROTATION	0x–
Set the number of bytes to increment at the end of the row.	DISPC_VIDp_ROW_INC[31:0] ROWINC	0x–
Set the number of bytes to increment between two pixels.	DISPC_VIDp_PIXEL_INC[7:0] PIXELINC	0x–
Set the X original image size.	DISPC_VIDp_PICTURE_SIZE[10:0] MEMSIZEX	0x–
Set the Y original image size.	DISPC_VIDp_PICTURE_SIZE[27:16] MEMSIZEY	0x–
Determine the FIFO preload mode.	DISPC_VIDp_ATTRIBUTES[19] BUFPRELOAD	0x–
Set the preload value.	DISPC_VIDp_PRELOAD[11:0] PRELOAD	0x–
Determine the burst type.	DISPC_VIDp_ATTRIBUTES[29] BURSTTYPE	0x–
Set the burst size.	DISPC_VIDp_ATTRIBUTES[15:14] BURSTSIZE	0x–
Set the high level of DMA FIFO threshold.	DISPC_VIDp_BUF_THRESHOLD[31:16] BUFHIGHTRESHOLD	0x–
Set the low level of DMA FIFO threshold.	DISPC_VIDp_BUF_THRESHOLD[15:0] BUFLOWTRESHOLD	0x–
Enable self-refresh.	DISPC_VIDp_ATTRIBUTES[24] SELFREFRESH	0x–
Select priority over the other pipeline.	DISPC_VIDp_ATTRIBUTES[23] ARBITRATION	0x–

(1) Applicable only for YUV pixel format

Table 11-96. DISPC Configure the WB DMA Channel

Step	Register/Bit Field/Programming Model	Value
Set the base address for RGB pixel format or Y component format according to memory access type, rotation, mirroring (see Section 11.2.4.6, *DISPC DMA Engine*).	DISPC_WB_BA_j[31:0] BA	0x—
Set the base address for Cb and Cr components according to memory access type, rotation, mirroring (see Section 11.2.4.6, *DISPC DMA Engine*) [(1)].	DISPC_WB_BA_UV_j[31:0] BA	0x—
Set the stride of CbCr component [(1)].	DISPC_WB_ATTRIBUTES[22] DOUBLESTRIDE	0x–
Set the rotation flag.	DISPC_WB_ATTRIBUTES[13:12] ROTATION	0x–
Set the number of bytes to increment at the end of the row.	DISPC_WB_ROW_INC[31:0] ROWINC [(2)]	0x–
Set the number of bytes to increment between two pixels.	DISPC_WB_PIXEL_INC[7:0] PIXELINC	0x–
Set the X final image size in system memory.	DISPC_WB_PICTURE_SIZE[10:0] ORGSIZEX	0x–
Set the Y final image size in system memory.	DISPC_WB_PICTURE_SIZE[27:16] ORGSIZEY	0x–
Set the burst size.	DISPC_WB_ATTRIBUTES[15:14] BURSTSIZE	0x–
Set the high level of DMA FIFO threshold.	DISPC_WB_BUF_THRESHOLD[31:16] BUFHIGHTRESHOLD	0x–
Set the low level of DMA FIFO threshold.	DISPC_WB_BUF_THRESHOLD[15:0] BUFLOWTRESHOLD	0x–
Select priority over the other pipeline.	DISPC_WB_ATTRIBUTES[23] ARBITRATION	0x–

(1) Applicable only for YUV pixel format.
(2) The DISPC_WB_ROW_INC register can be used only in 2D mode (using the Tiler). In order to use the DISPC_WB_ROW_INC register, the DISPC_WB_ATTRIBUTES[8] BURSTTYPE bit must be set to 1.

11.2.5.1.2.2 DISPC GFX Pipeline Configuration

11.2.5.1.2.2.1 DISPC Main Sequence – Configure the GFX Pipeline

This procedure details the steps for a GFX pipeline configuration (see Table 11-97).

Table 11-97. DISPC Configure the GFX Pipeline

Step	Register/Bit Field/Programming Model	Value
Configure the GFX window.	See Table 11-98.	
Configure the GFX pipeline processing.	See Table 11-99.	
Configure the GFX pipeline layer output.	See Table 11-100.	
Validate the GFX configuration according to outputs associated to the pipeline.	DISPC_CONTROL1[5] GOLCD	
	DISPC_CONTROL2[5] GOLCD	
	DISPC_CONTROL2[6] GOWB	
	DISPC_CONTROL1[6] GOTV	
Enable the GFX pipeline.	DISPC_GFX_ATTRIBUTES[0] ENABLE	0x1

11.2.5.1.2.2.2 DISPC Subsequence – Configure the GFX Window

This subsequence describes the parameters of the image to be displayed on the LCD panel (see Table 11-98).

Table 11-98. DISPC Configure the GFX Window

Step	Register/Bit Field/Programming Model	Value
Select the format of image.	DISPC_GFX_ATTRIBUTES[4:1] FORMAT	0x–
Set the X size of image to be displayed onto LCD panel.	DISPC_GFX_SIZE[10:0] SIZEX	0x–
Set the Y size of image to be displayed onto LCD panel.	DISPC_GFX_SIZE[27:16] SIZEY	0x–
Set the X position of image in respect to LCD panel.	DISPC_GFX_POSITION[10:0] POSX	0x–
Set the Y position of image in respect to LCD panel.	DISPC_GFX_POSITION[26:16] POSY	0x–

11.2.5.1.2.2.3 DISPC Subsequence – Configure the GFX Pipeline Processing

This subsequence describes the steps to configure the GFX pipeline processing in the DISPC (seeTable 11-99).

Table 11-99. DISPC Configure the GFX Pipeline Processing

Step	Register/Bit Field/Programming Model	Value
Enable replication logic.	DISPC_GFX_ATTRIBUTES[5] REPLICATIONENABLE	0x1
Enable antiflicker filter.	DISPC_GFX_ATTRIBUTES[24] ANTIFLICKER	0x1

Figure 11-100 shows the configuration of the DISPC graphics pipeline processing.

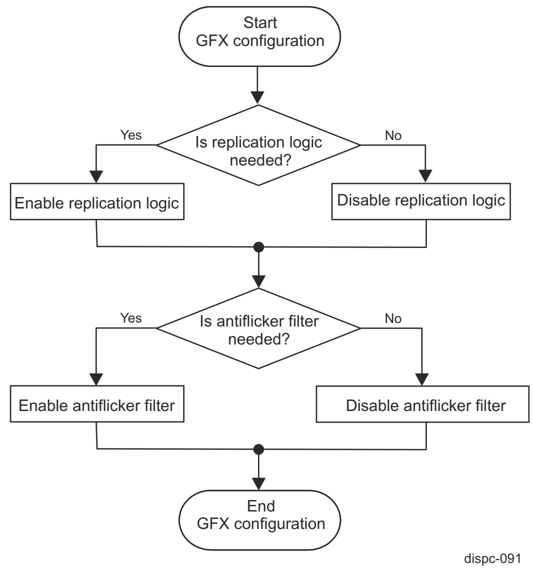


Figure 11-100. DISPC GFX Pipeline Processing Configuration

11.2.5.1.2.2.4 DISPC Subsequence – Configure the GFX Pipeline Layer Output

This subsequence describes the video layer settings available at the pipeline level necessary when using the overlay manager (see Table 11-100).

Table 11-100. DISPC Configure the GFX Pipeline Layer Output

Step	Register/Bit Field/Programming Model	Value
Set the LCD/TV output.	DISPC_GFX_ATTRIBUTES[8] CHANNELOUT	0x–
If DISPC_GFX_ATTRIBUTES[8] = 0, set the LCD output.	DISPC_GFX_ATTRIBUTES[31:30] CHANNELOUT2	0x–
Set the Z-order priority of the layer for overlay manager.	DISPC_GFX_ATTRIBUTES[27:26] ZORDER	0x–
Enable the video pipeline Z-order.	DISPC_GFX_ATTRIBUTES[25] ZORDERENABLE	0x–
Set the Global Alpha value for the alpha blender unit.	DISPC_GLOBAL_ALPHA[7:0] GFXGLOBALALPHA	0x–

11.2.5.1.2.3 DISPC Video Pipeline Configuration

11.2.5.1.2.3.1 DISPC Main Sequence – Configure the Video Pipeline

This sequence describes the steps to configure the video pipeline (see Table 11-101).

Table 11-101. DISPC Configure the Video Pipeline

Step	Register/Bit Field/Programming Model	Value
Configure the video window.	See Table 11-102.	
Configure the video pipeline processing.	See Table 11-103.	
Configure the video pipeline layer output.	See Table 11-108.	
Validate the video configuration according to outputs associated to the pipeline.	DISPC_CONTROL1[5] GOLCD	
	DISPC_CONTROL2[5] GOLCD	
	DISPC_CONTROL2[6] GOWB	
	DISPC_CONTROL1[6] GOTV	
Enable video pipeline.	DISPC_VIDp_ATTRIBUTES[0] ENABLE	0x1

11.2.5.1.2.3.2 DISPC Subsequence – Configure the Video Window

This subsequence describes the parameters of the image to be displayed on the LCD panel (see Table 11-102).

Table 11-102. DISPC Configure the Video Window

Step	Register/Bit Field/Programming Model	Value
Select the format of image.	DISPC_VIDp_ATTRIBUTES[4:1] FORMAT	0x–
Set the X size of image to be displayed onto LCD panel.	DISPC_VIDp_SIZE[10:0] SIZEX	0x–
Set the Y size of image to be displayed onto LCD panel.	DISPC_VIDp_SIZE[27:16] SIZEY	0x–
Set the X position of image in respect to LCD panel.	DISPC_VIDp_POSITION[10:0] POSX	0x–
Set the Y position of image in respect to LCD panel.	DISPC_VIDp_POSITION[26:16] POSY	0x–

11.2.5.1.2.3.3 DISPC Subsequence – Configure the Video Pipeline Processing

This subsequence describes the steps to configure video pipeline processing (see Table 11-103 and Figure 11-101).

Table 11-103. DISPC Configure the Video Pipeline Processing

Step	Register/Bit Field/Programming Model	Value
Select the video pipeline configuration 1 or 2.	DISPC_VIDp_ATTRIBUTES2[8] YUVCHROMARESAMPLING	0x–
Enable/disable replication logic[(1)].	DISPC_VIDp_ATTRIBUTES[10] REPLICATIONENABLE	0x1
Configure the VC-1 range mapping[(1)].	See Section 11.2.5.1.2.3.4.	
Configure the video color space conversion[(1)].	See Section 11.2.5.1.2.3.5.	
Configure the video scaler unit[(1)].	See Section 11.2.5.1.2.3.6.	

(1) This module configuration can be optional depending on:
- The video mode configuration selected. See Figure 11-101.
- The video format and application needs

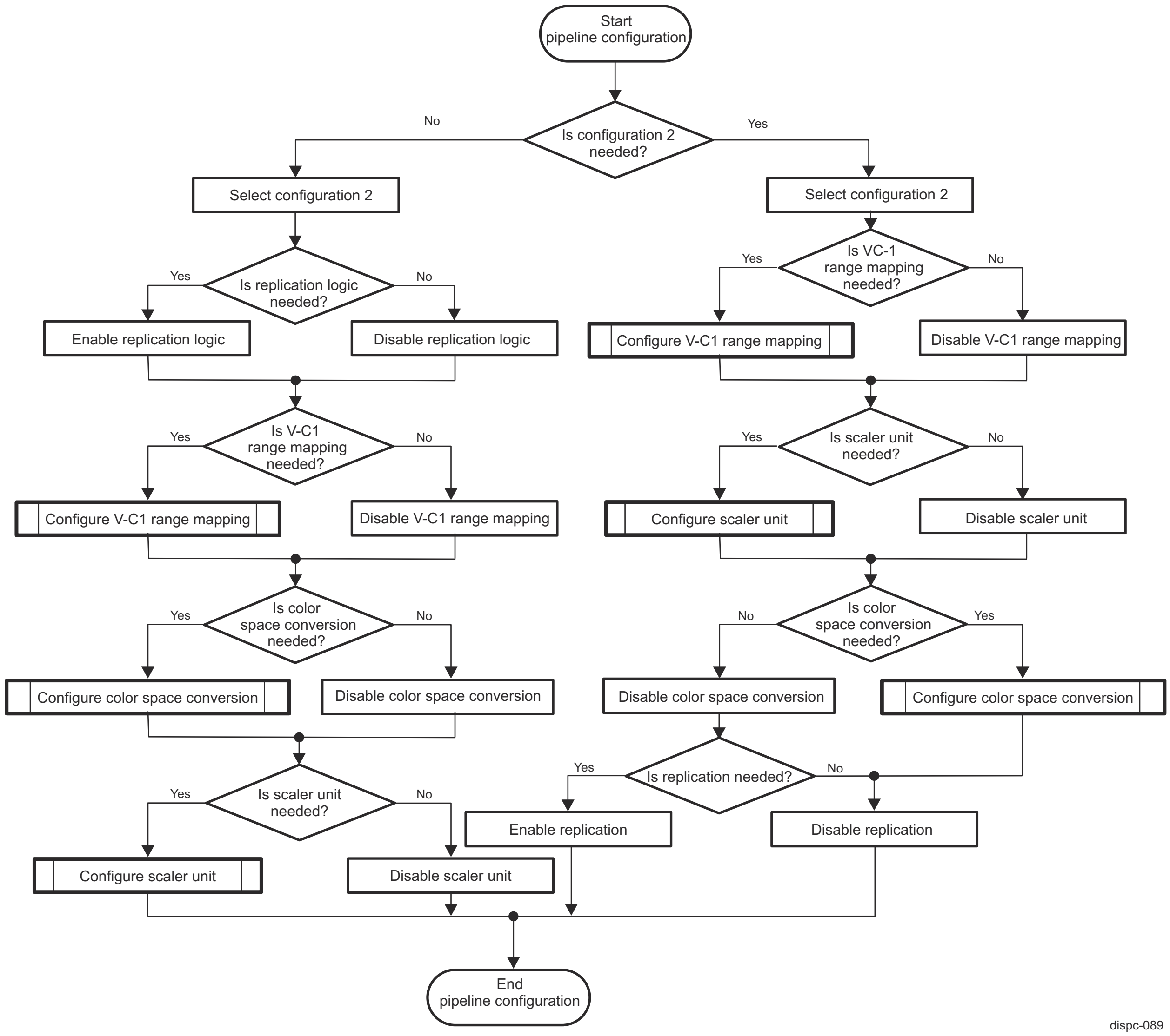


Figure 11-101. DISPC Video Pipeline Processing Configuration

11.2.5.1.2.3.4 DISPC Subsequence – Configure the VC-1 Range Mapping

This subsequence describes the steps to configure the VC-1 range mapping (see Table 11-104).

Table 11-104. DISPC Configure the VC-1 Range Mapping

Step	Register/Bit Field/Programming Model	Value
Set the Y component VC-1 range mapping.	DISPC_VIDp_ATTRIBUTES2[3:1] VC1_RANGE_Y	0x–
Set the Cb and Cr component VC-1 range mapping.	DISPC_VIDp_ATTRIBUTES2[6:4] VC1_RANGE_CBCR	0x–
Enable VC-1 range mapping.	DISPC_VIDp_ATTRIBUTES2[0] VC1ENABLE	0x1

11.2.5.1.2.3.5 DISPC Subsequence – Configure the Video Color Space Conversion

This subsequence describes the steps to configure the video color space conversion (see Table 11-105).

Table 11-105. DISPC Configure the Video Color Space Conversion

Step	Register/Bit Field/Programming Model	Value
Select the range of the color space conversion.	DISPC_VIDp_ATTRIBUTES[11] FULLRANGE	0x–
Set the RCr and RY coefficients.	DISPC_VIDp_CONV_COEF0[26:16][10:0] RCR, RY	0x–

Table 11-105. DISPC Configure the Video Color Space Conversion (continued)

Step	Register/Bit Field/Programming Model	Value
Set the GY and RCb coefficients.	DISPC_VIDp_CONV_COEF1[26:16][10:0] GY, RCB	0x–
Set the GCb and GCr coefficients.	DISPC_VIDp_CONV_COEF2[26:16][10:0] GCB, GCR	0x–
Set the BCr and BY coefficients.	DISPC_VIDp_CONV_COEF3[26:16][10:0] BCR, BY	0x–
Set the BCb coefficient.	DISPC_VIDp_CONV_COEF4[10:0] BCB	0x–
Enable color space conversion.	DISPC_VIDp_ATTRIBUTES[9] COLORCONVENABLE	0x1

11.2.5.1.2.3.6 DISPC Subsequence – Configure the Video Scaler Unit

This subsequence configures the video scaler unit. Table 11-106 is applicable for RGB pixel format. Table 11-106 and Table 11-107 are applicable for YUV pixel format.

Table 11-106. DISPC Configure the Video Scaler Unit for RGB Pixel Formats or Y Component

Step	Register/Bit Field/Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio.	DISPC_VIDp_FIR[12:0] FIRHINC	0x–
Set the horizontal FIR coefficients.	DISPC_VIDp_FIR_COEF_H_i[31:24][23:16][15:8] [7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0	0x–
	DISPC_VIDp_FIR_COEF_HV_i[7:0] FIRHC4	0x–
Set the horizontal accumulators value.	DISPC_VIDp_ACCU_j[10:0] HORIZONTALACCU	0x–
Configure vertical scaling		
Select number of vertical taps.	DISPC_VIDp_ATTRIBUTES[21] VERTICALTAPS	0x–
Set the vertical resizing ratio.	DISPC_VIDp_FIR[28:16] FIRVINC	0x–
Set the vertical FIR coefficients for RGB pixel format or Y component.	DISPC_VIDp_FIR_COEF_HV_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0	0x–
	Only for 5-taps vertical: DISPC_VIDp_FIR_COEF_V_i[15:8][7:0] FIRVC22, FIRVC00	0x–
Set the vertical accumulators value for RGB pixel format or Y component.	DISPC_VIDp_ACCU_j[26:16] VERTICALACCU	0x–
Enable horizontal and vertical scaler unit.	DISPC_VIDp_ATTRIBUTES[6:5] RESIZEENABLE	0x–

Table 11-107. DISPC Configure the Video Scaler Unit for Cb and Cr Components

Step	Register/Bit Field/Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio for Cb and Cr components.	DISPC_VIDp_FIR2[12:0] FIRHINC	0x–
Set the horizontal FIR coefficients for Cb and Cr components.	DISPC_VIDp_FIR_COEF_H2_i[31:24][23:16][15:8] [7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0	0x–
	DISPC_VIDp_FIR_COEF_HV2_i[7:0] FIRHC4	0x–
Set the horizontal accumulators value for Cb and Cr components.	DISPC_VIDp_ACCU2_j[10:0] HORIZONTALACCU	0x–
Configure vertical scaling		
Set the vertical resizing ratio for Cb and Cr components.	DISPC_VIDp_FIR2[28:16] FIRVINC	0x–
Set the vertical FIR coefficients for Cb and Cr components.	DISPC_VIDp_FIR_COEF_HV2_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0	0x–
	Only for 5-taps vertical: DISPC_VIDp_FIR_COEF_V2_i[15:8][7:0] FIRVC22, FIRVC00	0x–
Set the vertical accumulators value for Cb and Cr components.	DISPC_VIDp_ACCU2_j[26:16] VERTICALACCU	0x–

Figure 11-102 shows the programming flow of the DISPC scaler unit.

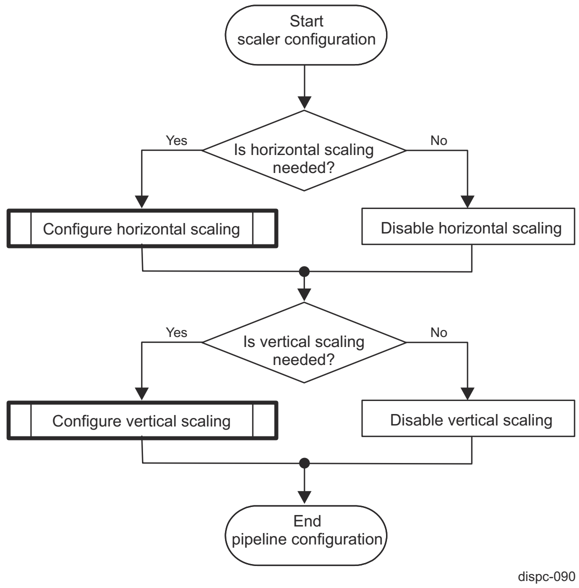


Figure 11-102. DISPC Scaler Unit Programming Flow

11.2.5.1.2.3.7 DISPC Subsequence – Configure the Video Pipeline Layer Output

This subsequence describes the video layer settings available at the pipeline level necessary when using the overlay manager (see Table 11-108).

Table 11-108. DISPC Configure the Video Pipeline Layer Output

Step	Register/Bit Field/Programming Model	Value
Set the LCD/TV output.	DISPC_VIDp_ATTRIBUTES[16] CHANNELOUT	0x–
If DISPC_VIDp_ATTRIBUTES[16] = 0, set the LCD output.	DISPC_VIDp_ATTRIBUTES[31:30] CHANNELOUT2	0x–
Set the Z-order priority of the layer for overlay manager.	DISPC_VIDp_ATTRIBUTES[27:26] ZORDER	0x–
Enable the video pipeline Z-order.	DISPC_VIDp_ATTRIBUTES[25] ZORDERENABLE	0x–
Set the Global Alpha value for the alpha blender unit.	DISPC_GLOBAL_ALPHA[31:24][23:16][15:8] VID3GLOBALALPHA, VID2GLOBALALPHA, VID1GLOBALALPHA	0x–

11.2.5.1.2.4 DISPC WB Pipeline Configuration

11.2.5.1.2.4.1 DISPC Main Sequence – Configure the WB Pipeline

This procedure describes the steps to configure the WB pipeline (see Table 11-109).

Table 11-109. DISPC Configure the WB Pipeline

Step	Register/Bit Field/Programming Model	Value
Configure the capture window.	See Table 11-110.	
Configure the WB scaler unit.	See Table 11-111.	
Enable truncation logic to match pixel size defined in the format of image DISPC_WB_ATTRIBUTES[4:1] FORMAT.	DISPC_WB_ATTRIBUTES[10] TRUNCATIONENABLE	0x–
If DISPC_WB_ATTRIBUTES[10] !=, configure the WB color space conversion unit.	See Table 11-113.	
Validate the configuration according to the registers modification.	DISPC_CONTROL2[6] GOWB	

Table 11-109. DISPC Configure the WB Pipeline (continued)

Step	Register/Bit Field/Programming Model	Value
Enable WB pipeline.	DISPC_WB_ATTRIBUTES[0] ENABLE	0x1

11.2.5.1.2.4.2 DISPC Subsequence – Configure the Capture Window

This subsequence describes the parameters of the image to be captured in the system memory (see Table 11-110).

Table 11-110. DISPC Configure the Capture Window

Step	Register/Bit Field/Programming Model	Value
Set the input source.	DISPC_WB_ATTRIBUTES[18:16] CHANNELIN	0x–
Select the format of image.	DISPC_WB_ATTRIBUTES[4:1] FORMAT	0x–
Set the X size of image to be captured.	DISPC_WB_SIZE[10:0] SIZEX	0x–
Set the Y size of image to be captured.	DISPC_WB_SIZE[10:0] SIZEY	0x–

11.2.5.1.2.4.3 DISPC Subsequence – Configure the WB Scaler Unit

This subsequence configures the scaler unit. Table 11-111 is applicable for RGB pixel format. Table 11-111 and Table 11-112 are applicable for YUV pixel format.

Table 11-111. DISPC Configure the WB Scaler Unit for RGB Pixel Formats or Y Component

Step	Register/Bit Field/Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio.	DISPC_WB_FIR[12:0] FIRHINC	0x–
Set the horizontal FIR coefficients.	DISPC_WB_FIR_COEF_H_i[31:24][23:16][15:8][7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0	0x–
	DISPC_WB_FIR_COEF_HV_i[7:0] FIRHC4	0x–
Set the horizontal accumulators value.	DISPC_WB_ACCU_j[10:0] HORIZONTALACCU	0x–
Configure vertical scaling		
Select number of vertical taps.	DISPC_WB_ATTRIBUTES[21] VERTICALTAPS	0x–
Set the vertical resizing ratio.	DISPC_WB_FIR[28:16] FIRVINC	0x–
Set the vertical FIR coefficients for RGB pixel format or Y component.	DISPC_WB_FIR_COEF_HV_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0	0x–
	Only for 5-taps vertical: DISPC_WB_FIR_COEF_V_i[15:8][7:0] FIRVC22, FIRVC00	0x–
Set the vertical accumulators value for RGB pixel format or Y component.	DISPC_WB_ACCU_j[26:16] VERTICALACCU	0x–
Enable horizontal and vertical scaler unit.	DISPC_WB_ATTRIBUTES[6:5] RESIZEENABLE	0x–

Table 11-112. DISPC Configure the WB Scaler Unit for Cb and Cr Components

Step	Register/Bit Field/Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio for Cb and Cr components.	DISPC_WB_FIR2[12:0] FIRHINC	0x–
Set the horizontal FIR coefficients for Cb and Cr components.	DISPC_WB_FIR_COEF_H2_i[31:24][23:16][15:8][7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0	0x–
	DISPC_WB_FIR_COEF_HV2_i[7:0] FIRHC4	0x–
Set the horizontal accumulators value for Cb and Cr components.	DISPC_WB_ACCU2_j[10:0] HORIZONTALACCU	0x–
Configure vertical scaling		
Set the vertical resizing ratio for Cb and Cr components.	DISPC_WB_FIR2[28:16] FIRVINC	0x–

Table 11-112. DISPC Configure the WB Scaler Unit for Cb and Cr Components (continued)

Step	Register/Bit Field/Programming Model	Value
Set the vertical FIR coefficients for Cb and Cr components.	DISPC_WB_FIR_COEF_HV2_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0	0x–
	Only for 5-taps vertical: DISPC_WB_FIR_COEF_V2_i[15:8][7:0] FIRVC22, FIRVC00	0x–
Set the vertical accumulators value for Cb and Cr components.	DISPC_WB_ACCU2_j[26:16] VERTICALACCU	0x–

11.2.5.1.2.4.4 DISPC Subsequence – Configure the WB Color Space Conversion Unit

This subsequence describes the steps to configure the WB color space conversion unit (see Table 11-113).

Table 11-113. DISPC Configure the WB Color Space Conversion Unit

Step	Register/Bit Field/Programming Model	Value
Select the range of the color space conversion.	DISPC_WB_ATTRIBUTES[11] FULLRANGE	0x–
Set the RCr and RY coefficients.	DISPC_WB_CONV_COEF0[26:16][10:0] YG, YR	0x–
Set the GY and RCb coefficients	DISPC_WB_CONV_COEF1[26:16][10:0] CRR, YB	0x–
Set the GCb and GCr coefficients.	DISPC_WB_CONV_COEF2[26:16][10:0] CRB, CRG	0x–
Set the BCr and BY coefficients.	DISPC_WB_CONV_COEF3[26:16][10:0] CBG, CBR	0x–
Set the BCb coefficient.	DISPC_WB_CONV_COEF4[10:0] CBB	0x–
Enable color space conversion.	DISPC_WB_ATTRIBUTES[9] COLORCONVENABLE	0x1

11.2.5.1.2.5 DISPC LCD Output Configuration

11.2.5.1.2.5.1 DISPC Main Sequence – Configure the LCD Output

This procedure details the LCD output configuration (see Table 11-114).

Table 11-114. DISPC Configure the LCD Output

Step	Register/Bit Field/Programming Model
Configure the LCD overlay manager.	See Table 11-115.
Configure the gamma table for gamma correction.	See Section 11.2.5.1.2.5.3.
Configure the CPR.	See Table 11-119.
Configure the LCD panel timings and parameters.	See Table 11-120.
Configure BT.656 or BT.1120 mode.	See Table 11-121
Validate the LCD output configuration according.	DISPC_CONTROL1[5] GOLCD
	DISPC_CONTROL2[5] GOLCD
Enable LCD output.	DISPC_CONTROL1[0] LCDENABLE
	DISPC_CONTROL2[0] LCDENABLE

11.2.5.1.2.5.2 DISPC Subsequence – Configure the Overlay Manager

This subsequence describes the overlay manager settings and transparency color key configuration (see Table 11-115).

Table 11-115. DISPC Configure the LCD Overlay Manager

Step	Register/Bit Field/Programming Model	Value
Set the LCD panel background color.	DISPC_DEFAULT_COLORo[23:0] DEFAULTCOLOR	0x–
Enable/disable overlay optimization.	DISPC_CONTROLo[12] OVERLAYOPTIMIZATION	0x–
Enable the alpha blender.	DISPC_CONFIG1[18] LCDALPHABLENDERENABLE	0x1
Configure the transparency color key		
Set source or destination transparency color key mode.	DISPC_CONFIGo[11] TCKLCDSELECTION	0x–
Set the transparency color value.	DISPC_TRANS_COLORo[23:0] TRANSCOLORKEY	0x–

Table 11-115. DISPC Configure the LCD Overlay Manager (continued)

Step	Register/Bit Field/Programming Model	Value
Enable transparency color key mode.	DISPC_CONFIGo[10] TCKLCDENABLE	0x–

11.2.5.1.2.5.3 DISPC Subsequence – Configure the Gamma Table for Gamma Correction

This subsequence describes the settings for configuring the gamma correction for LCD1, LCD2, and LCD3 (see Table 11-116 through Table 11-118, respectively).

Note

Software must ensure there is no visible effect when modifying the table because it is not under hardware control. The synchronization done using the DMA engine inside the DISPC to load the table when it is not used to display the picture on the screen is not present for this mode.

Table 11-116. DISPC Configure the Gamma Table for LCD1

Step	Register/Bit Field/Programming Model	Value
Initialize all entries for the gamma table by setting the table index and the RGB values associated to this index.	DISPC_GAMMA_TABLE0[31:24][23:16][15:8][7:0], INDEX, VALUE_R, VALUE_G, VALUE_B	0x–
Enable the gamma table.	DISPC_CONFIG1[3] PALETTEGAMMATABLE	0x1
Select the load mode of the gamma look-up table.	DISPC_CONFIG1[2:1] LOADMODE	0x–

Table 11-117. DISPC Configure the Gamma Table for LCD2

Step	Register/Bit Field/Programming Model	Value
Enable Gamma table for LCD2 and TV.	DISPC_CONFIG1[9] GAMATABLEENABLE	0x1
Initialize all entries for the gamma table by setting the table index and the RGB values associated to this index.	DISPC_GAMMA_TABLE2[31:24][23:16][15:8][7:0], INDEX, VALUE_R, VALUE_G, VALUE_B	0x–

Table 11-118. DISPC Configure the Gamma Table for LCD3

Step	Register/Bit Field/Programming Model	Value
Enable Gamma table for LCD3 and TV.	DISPC_CONFIG3[9] GAMATABLEENABLE	0x1
Initialize all entries for the gamma table by setting the table index and the RGB values associated to this index.	DISPC_GAMMA_TABLE3[31:24][23:16][15:8][7:0], INDEX, VALUE_R, VALUE_G, VALUE_B	0x–

11.2.5.1.2.5.4 DISPC Subsequence – Configure the Color Phase Rotation

Table 11-119 describes the settings for the CPR unit (see Table 11-119).

Table 11-119. DISPC Configure the Color Phase Rotation

Step	Register/Bit Field/Programming Model	Value
For Color Phase Rotation Processing		
Set the Red coefficients.	DISPC_CPRo_COEF_R [31:22][20:11][9:0] RR, RG, RB	0x–
Set the Green coefficients	DISPC_CPRo_COEF_G [31:22][20:11][9:0] GR, GG, GB	0x–
Set the Blue coefficients.	DISPC_CPRo_COEF_B [31:22][20:11][9:0] BR, BG, BB	0x–
Disable the CPR unit in RGB to YUV conversion.	DISPC_CONFIGo[24] COLORCONVENABLE	0x0
Enable the CPR unit.	DISPC_CONFIGo[15] CPR	0x1
For Color Space Conversion Processing		
Set the Y coefficients.	DISPC_CPRo_COEF_R [31:22][20:11][9:0] RR, RG, RB	0x–
Set the Cb coefficients.	DISPC_CPRo_COEF_G [31:22][20:11][9:0] GR, GG, GB	0x–
Set the Cr coefficients.	DISPC_CPRo_COEF_B [31:22][20:11][9:0] BR, BG, BB	0x–
Set the CSC range.	DISPC_CONFIGo[25] FULLRANGE	0x–

Table 11-119. DISPC Configure the Color Phase Rotation (continued)

Step	Register/Bit Field/Programming Model	Value
Enable the CPR unit in RGB to YUV conversion.	DISPC_CONFIGo[24] COLORCONVENABLE	0x1
Disable the CPR unit.	DISPC_CONFIGo[15] CPR	0x0

11.2.5.1.2.5.5 DISPC Subsequence – Configure the LCD Panel Timings and Parameters

This subsequence describes the setting for horizontal and vertical synchronization and signal polarity (see Table 11-120).

Table 11-120. DISPC Configure the LCD Panel Timings and Parameters

Step	Register/Bit Field/Programming Model	Value
Configure spatial/temporal dithering		
Select spatial/temporal number of frames.	DISPC_CONTROLo[31:30] SPATIALTEMPORALDITHERINGFRAMES	0x–
Enable spatial/temporal dithering.	DISPC_CONTROLo[7] STDITHERENABLE	0x–
Configure AC-bias		
Configure the VSYNC, HSYNC and AC bias polarity.	DISPC_POL_FREQo	0x–
Configure the gating of AC bias polarity.	DISPC_CONFIGo[8] ACBIASGATED	0x–
Configure the AC bias polarity.	DISPC_POL_FREQo[15] IEO	0x–
Set the AC bias frequency.	DISPC_POL_FREQo[7:0] ACB	0x–
Set the number of AC bias transitions per interrupt.	DISPC_POL_FREQo[11:8] ACBI	0x–
Configure the pixel clock		
Set the DISPC logic clock divisor.	DISPC_DIVISORo[23:16] LCD	0x–
Set the pixel clock divisor.	DISPC_DIVISORo[7:0] PCD	0x–
Configure the gating of pixel clock.	DISPC_CONFIGo[5] PIXELCLOCKGATED	0x–
Configure the data		
Set the pixel clock edge to drive data output.	DISPC_POL_FREQo[14] IPC and CTRL_CORE_SMA_SW_1[]DSS_CHx_IPC(1)	0x–
Configure the gating of data output.	DISPC_CONFIGo[4] PIXELDATAGATED	0x–
Set the data output mode.	DISPC_CONFIGo[22] OUTPUTMODEENABLE	0x–
Configure the panel parameters		
Set the vertical TV size.	DISPC_SIZE_LCDo[27:16] LPP	0x–
Set the horizontal TV size.	DISPC_SIZE_LCDo[11:0] PPL	0x–
Set the panel type.	DISPC_CONTROLo[3] STNTFT	0x–
Configure the refresh rate and horizontal and vertical parameters		
Set the vertical synchronization timing.	DISPC_TIMING_Vo[31:20][19:8][7:0], VBP, VFP, VSW	0x–
Configure the VSYNC polarity.	DISPC_POL_FREQo[12] IVS	0x–
Configure the gating of VSYNC.	DISPC_CONFIGo[7] VSYNCGATED	0x–
Set the horizontal synchronization timing.	DISPC_TIMING_Ho[31:20][19:8][7:0], HBP, HFP, HSW	0x–
Configure the HSYNC polarity.	DISPC_POL_FREQo[13] IHS	0x–
Configure the gating of HSYNC.	DISPC_CONFIGo[6] HSYNCGATED	0x–
Set the opposition of HSYNC and VSYNC driving.	DISPC_POL_FREQo[17] ONOFF and CTRL_CORE_SMA_SW_1[]DSS_CHx_ON_OFF(1)	0x–
If DISPC_POL_FREQo[17] = 1, set the pixel clock edge to drive HSYNC and VSYNC.	DISPC_POL_FREQo[16] RF and CTRL_CORE_SMA_SW_1[]DSS_CHx_RF(1)	0x–
Set the alignment of HSYNC and VSYNC.	DISPC_POL_FREQo[18] ALIGN	0x–

(1) The values of the DISPC and Control Module registers must mach

11.2.5.1.2.5.6 DISPC Subsequence – Configure BT.656 or BT.1120 Mode

CAUTION

DISPC supports maximum 256 bytes of horizontal blanking when the LCD outputs are configured in BT modes (bitfield HSW of DISPC_TIMING_H1/2/3 registers is 8 bits wide). This is not compliant with BT.656 and BT.1120 standards, both of which require higher blanking periods. If more than 256 bytes of horizontal blanking are required by the application, then the RGB mode must be used for the LCD outputs.

Table 11-121 lists the steps to configure BT.656 or BT1120.

Table 11-121. DISPC Configure BT.656 or BT.1120 Mode

Step	Register/Bit Field/Programming Model	Value
Enable BT.656 mode [(1)]	DISPC_CONFIGo[20] BT656ENABLE	0x1
Enable BT.1120 mode [(1)]	DISPC_CONFIGo[21] BT1120ENABLE	0x1
Configure the blanking values	See Table 11-122	-
Configure the color phase rotation block	See Table 11-119	-
Disable STALL mode	DISPC_CONTROLo[11] STALLMODE	0x0
Disable TDM	DISPC_CONTROLo[20] TDMENABLE	0x0

(1) It is not possible to enable BT.656 and BT.1120 modes simultaneously on one LCD output.

Table 11-122 lists the blankng values for BT.656 and BT.1120.

Table 11-122. DISPC Configure BT.656 or BT.1120 Blanking Values

Step	Register/Bit Field/Programming Model	Value
For Progressive Mode		
Select progressive mode	DISPC_CONFIGo[22] OUTPUTMODEENABLE	0x0
Set Horizontal blanking	DISPC_TIMING_Ho[7:0] HSW	0x-
Set Vertical frame blanking No 1	DISPC_TIMING_Vo[19:8] VFP	0x-
Set Vertical frame blanking No 2	DISPC_TIMING_Vo[31:20] VBP	0x-
Set Number of lines	DISPC_SIZE_LCD1[27:16] LPP	0x-
Set Number of pixels per line	DISPC_SIZE_LCD1[11:0] PPL	0x-
For Interlaced Mode		
Select interlaced mode	DISPC_CONFIGo[22] OUTPUTMODEENABLE	0x1
Set Horizontal blanking	DISPC_TIMING_Ho[7:0] HSW	0x-
Set Vertical field blanking No 1 for Even Field	DISPC_TIMING_Ho[19:8] HFP	0x-
Set Vertical field blanking No 2 for Even Field	DISPC_TIMING_Ho[31:20] HBP	0x-
Set Vertical field blanking No 1 for Odd Field	DISPC_TIMING_Vo[19:8] VFP	0x-
Set Vertical field blanking No 2 for Odd Field	DISPC_TIMING_Vo[31:20] VBP	0x-
Set Number of lines per field (even)	DISPC_SIZE_LCD1[27:16] LPP	0x-
Set Delta number of odd field compared to even field (in a single line)	DISPC_SIZE_LCD1[15:14] DELTA	0x-
Set Number of pixels per line	DISPC_SIZE_LCD1[11:0] PPL	0x-

11.2.5.1.2.6 DISPC TV Output Configuration

11.2.5.1.2.6.1 DISPC Main Sequence – Configure the TV Output

This procedure describes the TV output configuration (see Table 11-123).

Table 11-123. DISPC Configure the TV Output

Step	Register/Bit Field/Programming Model
Configure the TV overlay manager.	See Table 11-124.
Configure the gamma table for gamma correction.	See Table 11-125.
Configure the TV panel timings and parameters.	See Table 11-126.
Validate the TV output configuration accordingly.	DISPC_CONTROL1[6] GOTV
Enable the TV output.	DISPC_CONTROL1[1] TVENABLE

11.2.5.1.2.6.1.1 DISPC Subsequence – Configure the TV Overlay Manager

This subsequence describes the overlay manager settings and transparency color key configuration (see Table 11-124).

Table 11-124. DISPC Configure the TV Overlay Manager

Step	Register/Bit Field/Programming Model	Value
Set the TV panel background color.	DISPC_DEFAULT_COLOR1[23:0] DEFAULTCOLOR	0x–
Enable/disable overlay optimization.	DISPC_CONTROL2[13] TVOVERLAYOPTIMIZATION	0x–
Enable the alpha blender.	DISPC_CONFIG1[19] TVALPHABLENDERENABLE	0x1
Configure the transparency color key		
Set source or destination transparency color key mode.	DISPC_CONFIG1[13] TCKTVSELECTION	0x–
Set the transparency color value.	DISPC_TRANS_COLOR1[23:0] TRANSCOLORKEY	0x–
Enable transparency color key mode.	DISPC_CONFIG1[10] TCKTVENABLE	0x–

11.2.5.1.2.6.1.2 DISPC Subsequence – Configure the Gamma Table for Gamma Correction

This subsequence describes the steps to configure the gamma table for gamma correction (see Table 11-125).

Note

Software must ensure there is no visible effect when modifying the table because it is not under hardware control. The synchronization done using the DMA engine inside the DISPC to load the table when it is not used for displaying the picture on the screen is not present for this mode.

Table 11-125. DISPC Configure the Gamma Table for TV Output

Step	Register/Bit Field/Programming Model	Value
Enable Gamma table for LCD2 and TV.	DISPC_CONFIG1[9] GAMATABLEENABLE	0x1
Initialize all entries for the gamma table by setting the table index and the RGB values. For more information, see Section 11.2.4.14.2, *Gamma Correction Unit*.	DISPC_GAMMA_TABLE2[31][29:20][19:10][9:0] INDEX, VALUE_R, VALUE_G, VALUE_B	0x–

11.2.5.1.2.6.1.3 DISPC Subsequence – Configure the TV Panel Timings and Parameters

This subsequence describes the settings for horizontal and vertical synchronization and signal polarity (see Table 11-126).

Table 11-126. DISPC Configure the TV Panel Timings and Parameters

Step	Register/Bit Field/Programming Model	Value
Set the hold time for TV data outputs.	DISPC_CONFIG1[19:17]	0x–
Set the vertical TV size.	DISPC_SIZE_TV[27:16] LPP	See Table 11-90 for HD standards.
Set the horizontal TV size.	DISPC_SIZE_TV[11:0] PPL	

11.2.6 DISPC Register Manual

11.2.6.1 DISPC Instance Summary

Table 11-127. DISPC Instance Summary

Module Name	L3_MAIN Base Address	Size
DISPC	0x5800 1000	4 KiB

11.2.6.2 DISPC Logical Register Mapping

Table 11-128. DISPC_VIDp_BA_j Logical Register Mapping

Hardware Register	Description
DISPC_VID1_BA_j	Base address of video pipeline 1
DISPC_VID2_BA_j	Base address of video pipeline 2
DISPC_VID3_BA_j	Base address of video pipeline 3

Table 11-129. DISPC_VIDp_BA_UV_j Logical Register Mapping

Hardware Register	Description
DISPC_VID1_BA_UV_j	Base address of UV components for video pipeline 1
DISPC_VID2_BA_UV_j	Base address of UV components for video pipeline 2
DISPC_VID3_BA_UV_j	Base address of UV components for video pipeline 3

Table 11-130. DISPC_VIDp_POSITION Logical Register Mapping

Hardware Register	Description
DISPC_VID1_POSITION	Position of the video window 1
DISPC_VID2_POSITION	Position of the video window 2
DISPC_VID3_POSITION	Position of the video window 3

Table 11-131. DISPC_VIDp_SIZE Logical Register Mapping

Hardware Register	Description
DISPC_VID1_SIZE	Size of the video window 1
DISPC_VID2_SIZE	Size of the video window 2
DISPC_VID3_SIZE	Size of the video window 3

Table 11-132. DISPC_VIDp_ATTRIBUTES Logical Register Mapping

Hardware Register	Description
DISPC_VID1_ATTRIBUTES	Configuration of the video pipeline 1
DISPC_VID2_ATTRIBUTES	Configuration of the video pipeline 2
DISPC_VID3_ATTRIBUTES	Configuration of the video pipeline 3

Table 11-133. DISPC_VIDp_ATTRIBUTES2 Logical Register Mapping

Hardware Register	Description
DISPC_VID1_ATTRIBUTES2	Configuration of the video pipeline 1
DISPC_VID2_ATTRIBUTES2	Configuration of the video pipeline 2
DISPC_VID3_ATTRIBUTES2	Configuration of the video pipeline 3

Table 11-134. DISPC_VIDp_BUF_THRESHOLD Logical Register Mapping

Hardware Register	Description
DISPC_VID1_BUF_THRESHOLD	Configuration of the buffer for the video pipeline 1
DISPC_VID2_BUF_THRESHOLD	Configuration of the buffer for the video pipeline 2
DISPC_VID3_BUF_THRESHOLD	Configuration of the buffer for the video pipeline 3

Table 11-135. DISPC_VIDp_ROW_INC Logical Register Mapping

Hardware Register	Description
DISPC_VID1_ROW_INC	Configuration of the row increment for the video pipeline 1
DISPC_VID2_ROW_INC	Configuration of the row increment for the video pipeline 2
DISPC_VID3_ROW_INC	Configuration of the row increment for the video pipeline 3

Table 11-136. DISPC_VIDp_PIXEL_INC Logical Register Mapping

Hardware Register	Description
DISPC_VID1_PIXEL_INC	Configuration of the pixel increment for the video pipeline 1
DISPC_VID2_PIXEL_INC	Configuration of the pixel increment for the video pipeline 2
DISPC_VID3_PIXEL_INC	Configuration of the pixel increment for the video pipeline 3

Table 11-137. DISPC_VIDp_FIR Logical Register Mapping

Hardware Register	Description
DISPC_VID1_FIR	Configuration of the scaler for the video pipeline 1
DISPC_VID2_FIR	Configuration of the scaler for the video pipeline 2
DISPC_VID3_FIR	Configuration of the scaler for the video pipeline 3

Table 11-138. DISPC_VIDp_PICTURE_SIZE Logical Register Mapping

Hardware Register	Description
DISPC_VID1_PICTURE_SIZE	Size of the video window 1 before processing
DISPC_VID2_PICTURE_SIZE	Size of the video window 2 before processing
DISPC_VID3_PICTURE_SIZE	Size of the video window 3 before processing

Table 11-139. DISPC_VIDp_ACCU_j Logical Register Mapping

Hardware Register	Description
DISPC_VID1_ACCU_j	Configuration of the accumulator for the video pipeline 1
DISPC_VID2_ACCU_j	Configuration of the accumulator for the video pipeline 2
DISPC_VID2_ACCU_j	Configuration of the accumulator for the video pipeline 3

Table 11-140. DISPC_VIDp_FIR_COEF_H_i Logical Register Mapping

Hardware Register	Description
DISPC_VID1_FIR_COEF_H_i	Configuration of the horizontal scaling coefficients for the video pipeline 1
DISPC_VID2_FIR_COEF_H_i	Configuration of the horizontal scaling coefficients for the video pipeline 2
DISPC_VID3_FIR_COEF_H_i	Configuration of the horizontal scaling coefficients or the video pipeline 3

Table 11-141. DISPC_VIDp_FIR_COEF_HV_i Logical Register Mapping

Hardware Register	Description
DISPC_VID1_FIR_COEF_HV_i	Configuration of the horizontal scaling coefficients for the video pipeline 1
DISPC_VID2_FIR_COEF_HV_i	Configuration of the horizontal scaling coefficients for the video pipeline 2
DISPC_VID3_FIR_COEF_HV_i	Configuration of the horizontal scaling coefficients for the video pipeline 3

Table 11-142. DISPC_VIDp_FIR_COEF_V_i Logical Register Mapping

Hardware Register	Description
DISPC_VID1_FIR_COEF_V_i	Configuration of the vertical scaling coefficients for the video pipeline 1

Table 11-142. DISPC_VIDp_FIR_COEF_V_i Logical Register Mapping (continued)

Hardware Register	Description
DISPC_VID2_FIR_COEF_V_i	Configuration of the vertical scaling coefficients for the video pipeline 2
DISPC_VID3_FIR_COEF_V_i	Configuration of the vertical scaling coefficients for the video pipeline 3

Table 11-143. DISPC_VIDp_FIR_COEF_H2_i Logical Register Mapping

Hardware Register	Description
DISPC_VID1_FIR_COEF_H2_i	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 1
DISPC_VID2_FIR_COEF_H2_i	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 2
DISPC_VID3_FIR_COEF_H2_i	Configuration of the horizontal Cb and Cr scaling coefficients or the video pipeline 3

Table 11-144. DISPC_VIDp_FIR_COEF_HV2_i Logical Register Mapping

Hardware Register	Description
DISPC_VID1_FIR_COEF_HV2_i	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 1
DISPC_VID2_FIR_COEF_HV2_i	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 2
DISPC_VID3_FIR_COEF_HV2_i	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 3

Table 11-145. DISPC_VIDp_FIR_COEF_V2_i Logical Register Mapping

Hardware Register	Description
DISPC_VID1_FIR_COEF_V2_i	Configuration of the vertical Cb and Cr scaling coefficients for the video pipeline 1
DISPC_VID2_FIR_COEF_V2_i	Configuration of the vertical Cb and Cr scaling coefficients for the video pipeline 2
DISPC_VID3_FIR_COEF_V2_i	Configuration of the vertical Cb and Cr scaling coefficients for the video pipeline 3

Table 11-146. DISPC_VIDp_CONV_COEF0 Logical Register Mapping

Hardware Register	Description
DISPC_VID1_CONV_COEF0	Configuration of the Color Space Conversion coefficients for the video pipeline 1
DISPC_VID2_CONV_COEF0	Configuration of the Color Space Conversion coefficient for the video pipeline 2
DISPC_VID3_CONV_COEF0	Configuration of the Color Space Conversion coefficient for the video pipeline 3

Table 11-147. DISPC_VIDp_CONV_COEF1 Logical Register Mapping

Hardware Register	Description
DISPC_VID1_CONV_COEF1	Configuration of the Color Space Conversion coefficients for the video pipeline 1
DISPC_VID2_CONV_COEF1	Configuration of the Color Space Conversion coefficient for the video pipeline 2
DISPC_VID3_CONV_COEF1	Configuration of the Color Space Conversion coefficient for the video pipeline 3

Table 11-148. DISPC_VIDp_CONV_COEF2 Logical Register Mapping

Hardware Register	Description
DISPC_VID1_CONV_COEF2	Configuration of the Color Space Conversion coefficients for the video pipeline 1

Table 11-148. DISPC_VIDp_CONV_COEF2 Logical Register Mapping (continued)

Hardware Register	Description
DISPC_VID2_CONV_COEF2	Configuration of the Color Space Conversion coefficient for the video pipeline 2
DISPC_VID3_CONV_COEF2	Configuration of the Color Space Conversion coefficient for the video pipeline 3

Table 11-149. DISPC_VIDp_CONV_COEF3 Logical Register Mapping

Hardware Register	Description
DISPC_VID1_CONV_COEF3	Configuration of the Color Space Conversion coefficients for the video pipeline 1
DISPC_VID2_CONV_COEF3	Configuration of the Color Space Conversion coefficient for the video pipeline 2
DISPC_VID3_CONV_COEF3	Configuration of the Color Space Conversion coefficient for the video pipeline 3

Table 11-150. DISPC_VIDp_CONV_COEF4 Logical Register Mapping

Hardware Register	Description
DISPC_VID1_CONV_COEF4	Configuration of the Color Space Conversion coefficients for the video pipeline 1
DISPC_VID2_CONV_COEF4	Configuration of the Color Space Conversion coefficient for the video pipeline 2
DISPC_VID3_CONV_COEF4	Configuration of the Color Space Conversion coefficient for the video pipeline 3

Table 11-151. DISPC_CONTROLo Logical Register Mapping

Hardware Register	Description
DISPC_CONTROL1	Configuration control of the LCD1 and TV.
DISPC_CONTROL2	Configuration control of the LCD2.
DISPC_CONTROL3	Configuration control of the LCD3.

Table 11-152. DISPC_CONFIGo Logical Register Mapping

Hardware Register	Description
DISPC_CONFIG1	Configuration of the LCD1 and TV.
DISPC_CONFIG2	Configuration of the LCD2.
DISPC_CONFIG3	Configuration of the LCD3.

Table 11-153. DISPC_DEFAULT_COLORo Logical Register Mapping

Hardware Register	Description
DISPC_DEFAULT_COLOR0	Configuration of the background color for LCD1.
DISPC_DEFAULT_COLOR1	Configuration of the background color for TV.
DISPC_DEFAULT_COLOR2	Configuration of the background color for LCD2.
DISPC_DEFAULT_COLOR3	Configuration of the background color for LCD3.

Table 11-154. DISPC_TRANS_COLORo Logical Register Mapping

Hardware Register	Description
DISPC_TRANS_COLOR0	Configuration of the transparency color key for LCD1.
DISPC_TRANS_COLOR1	Configuration of the transparency color key for TV.
DISPC_TRANS_COLOR2	Configuration of the transparency color key for LCD2.
DISPC_DEFAULT_COLOR3	Configuration of the background color for LCD3.

Table 11-155. DISPC_GAMMA_TABLEo Logical Register Mapping

Hardware Register	Description
DISPC_GAMMA_TABLE0	Configuration of the gamma table for LCD1.
DISPC_GAMMA_TABLE1	Configuration of the gamma table for LCD2.
DISPC_GAMMA_TABLE2	Configuration of the gamma table for TV output.
DISPC_GAMMA_TABLE3	Configuration of the gamma table for LCD3.

Table 11-156. DISPC_TIMING_Ho Logical Register Mapping

Hardware Register	Description
DISPC_TIMING_H1	Configuration of the horizontal timing for LCD1.
DISPC_TIMING_H2	Configuration of the horizontal timing for LCD2.
DISPC_TIMING_H3	Configuration of the horizontal timing for LCD3.

Table 11-157. DISPC_TIMING_Vo Logical Register Mapping

Hardware Register	Description
DISPC_TIMING_V1	Configuration of the vertical timing for LCD1.
DISPC_TIMING_V2	Configuration of the vertical timing for LCD2.
DISPC_TIMING_V3	Configuration of the vertical timing for LCD3.

Table 11-158. DISPC_POL_FREQo Logical Register Mapping

Hardware Register	Description
DISPC_POL_FREQ1	Configuration of the output signals for LCD1.
DISPC_POL_FREQ2	Configuration of the output signals for LCD2.
DISPC_POL_FREQ3	Configuration of the output signals for LCD3.

Table 11-159. DISPC_DIVISORo Logical Register Mapping

Hardware Register	Description
DISPC_DIVISOR1	Configuration of the divisors for LCD1.
DISPC_DIVISOR2	Configuration of the divisors for LCD2.
DISPC_DIVISOR3	Configuration of the divisors for LCD3.

Table 11-160. DISPC_SIZE_LCDo Logical Register Mapping

Hardware Register	Description
DISPC_SIZE_LCD1	Configuration of the divisors for LCD1.
DISPC_SIZE_LCD2	Configuration of the divisors for LCD2.
DISPC_SIZE_LCD3	Configuration of the divisors for LCD3.

Table 11-161. DISPC_SIZE Logical Register Mapping

Hardware Register	Description
DISPC_SIZE_LCD1	Configuration of the LCD size on LCD1.
DISPC_SIZE_LCD2	Configuration of the LCD size on LCD2.
DISPC_SIZE_LCD3	Configuration of the LCD size on LCD3.

Table 11-162. DISPC_DATAo_CYCLE1 Logical Register Mapping

Hardware Register	Description
DISPC_DATA1_CYCLE1	Configuration of the output data format for first cycle on LCD1.
DISPC_DATA2_CYCLE1	Configuration of the output data format for first cycle on LCD2.
DISPC_DATA3_CYCLE1	Configuration of the output data format for first cycle on LCD3.

Table 11-163. DISPC_DATAo_CYCLE2 Logical Register Mapping

Hardware Register	Description
DISPC_DATA1_CYCLE2	Configuration of the output data format for second cycle on LCD1.
DISPC_DATA2_CYCLE2	Configuration of the output data format for second cycle on LCD2.
DISPC_DATA3_CYCLE2	Configuration of the output data format forsecond cycle on LCD3.

Table 11-164. DISPC_DATAo_CYCLE3 Logical Register Mapping

Hardware Register	Description
DISPC_DATA1_CYCLE3	Configuration of the output data format for third cycle on LCD1.
DISPC_DATA2_CYCLE3	Configuration of the output data format for third cycle on LCD2.
DISPC_DATA3_CYCLE3	Configuration of the output data format for third cycle on LCD3.

Table 11-165. DISPC_CPRo_COEF_R Logical Register Mapping

Hardware Register	Description
DISPC_CPR1_COEF_R	Configuration of the CPR matrix coefficients for the red component on LCD1.
DISPC_CPR2_COEF_R	Configuration of the CPR matrix coefficients for the red component on LCD2.
DISPC_CPR3_COEF_R	Configuration of the CPR matrix coefficients for the red component on LCD3

Table 11-166. DISPC_CPRo_COEF_G Logical Register Mapping

Hardware Register	Description
DISPC_CPR1_COEF_G	Configuration of the CPR matrix coefficients for the green component on LCD1.
DISPC_CPR2_COEF_G	Configuration of the CPR matrix coefficients for the green component on LCD2.
DISPC_CPR3_COEF_G	Configuration of the CPR matrix coefficients for the green component on LCD3.

Table 11-167. DISPC_CPRo_COEF_B Logical Register Mapping

Hardware Register	Description
DISPC_CPR1_COEF_B	Configuration of the CPR matrix coefficients for the blue component on LCD1.
DISPC_CPR2_COEF_B	Configuration of the CPR matrix coefficients for the blue component on LCD2.
DISPC_CPR3_COEF_B	Configuration of the CPR matrix coefficients for the blue component on LCD3.

11.2.6.3 DISPC Registers

11.2.6.3.1 DISPC Register Summary

Table 11-168. DISPC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DISPC_REVISION	R	32	0x0000 0000	0x5800 1000
DISPC_SYSCONFIG	RW	32	0x0000 0010	0x5800 1010
DISPC_SYSSTATUS	R	32	0x0000 0014	0x5800 1014
DISPC_IRQSTATUS	RW	32	0x0000 0018	0x5800 1018
DISPC_IRQENABLE	RW	32	0x0000 001C	0x5800 101C
DISPC_CONTROL1	RW	32	0x0000 0040	0x5800 1040
DISPC_CONFIG1	RW	32	0x0000 0044	0x5800 1044
RESERVED	R	32	0x0000 0048	0x5800 1048

Table 11-168. DISPC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DISPC_DEFAULT_COLOR0	RW	32	0x0000 004C	0x5800 104C
DISPC_DEFAULT_COLOR1	RW	32	0x0000 0050	0x5800 1050
DISPC_TRANS_COLOR0	RW	32	0x0000 0054	0x5800 1054
DISPC_TRANS_COLOR1	RW	32	0x0000 0058	0x5800 1058
DISPC_LINE_STATUS	R	32	0x0000 005C	0x5800 105C
DISPC_LINE_NUMBER	RW	32	0x0000 0060	0x5800 1060
DISPC_TIMING_H1	RW	32	0x0000 0064	0x5800 1064
DISPC_TIMING_V1	RW	32	0x0000 0068	0x5800 1068
DISPC_POL_FREQ1	RW	32	0x0000 006C	0x5800 106C
DISPC_DIVISOR1	RW	32	0x0000 0070	0x5800 1070
DISPC_GLOBAL_ALPHA	RW	32	0x0000 0074	0x5800 1074
DISPC_SIZE_TV	RW	32	0x0000 0078	0x5800 1078
DISPC_SIZE_LCD1	RW	32	0x0000 007C	0x5800 107C
DISPC_GFX_BA_j [(1)]	RW	32	0x0000 0080 + (0x4 * j)	0x5800 1080 + (0x4 * j)
DISPC_GFX_POSITION	RW	32	0x0000 0088	0x5800 1088
DISPC_GFX_SIZE	RW	32	0x0000 008C	0x5800 108C
DISPC_GFX_ATTRIBUTES	RW	32	0x0000 00A0	0x5800 10A0
DISPC_GFX_BUF_THRESHOLD	RW	32	0x0000 00A4	0x5800 10A4
DISPC_GFX_BUF_SIZE_STATUS	R	32	0x0000 00A8	0x5800 10A8
DISPC_GFX_ROW_INC	RW	32	0x0000 00AC	0x5800 10AC
DISPC_GFX_PIXEL_INC	RW	32	0x0000 00B0	0x5800 10B0
RESERVED	R	32	0x0000 00B4	0x5800 10B4
DISPC_GFX_TABLE_BA	RW	32	0x0000 00B8	0x5800 10B8
DISPC_VID1_BA_j [(1)]	RW	32	0x0000 00BC + (0x4 * j)	0x5800 10BC + (0x4 * j)
DISPC_VID1_POSITION	RW	32	0x0000 00C4	0x5800 10C4
DISPC_VID1_SIZE	RW	32	0x0000 00C8	0x5800 10C8
DISPC_VID1_ATTRIBUTES	RW	32	0x0000 00CC	0x5800 10CC
DISPC_VID1_BUF_THRESHOLD	RW	32	0x0000 00D0	0x5800 10D0
DISPC_VID1_BUF_SIZE_STATUS	R	32	0x0000 00D4	0x5800 10D4
DISPC_VID1_ROW_INC	RW	32	0x0000 00D8	0x5800 10D8
DISPC_VID1_PIXEL_INC	RW	32	0x0000 00DC	0x5800 10DC
DISPC_VID1_FIR	RW	32	0x0000 00E0	0x5800 10E0
DISPC_VID1_PICTURE_SIZE	RW	32	0x0000 00E4	0x5800 10E4
DISPC_VID1_ACCU_j [(1)]	RW	32	0x0000 00E8 + (0x4 * j)	0x5800 10E8 + (0x4 * j)
DISPC_VID1_FIR_COEF_H_i [(2)]	RW	32	0x0000 00F0 + (0x8 * i)	0x5800 10F0 + (0x8 * i)
DISPC_VID1_FIR_COEF_HV_i [(2)]	RW	32	0x0000 00F4 + (0x8 * i)	0x5800 10F4 + (0x8 * i)
DISPC_VID1_CONV_COEF0	RW	32	0x0000 0130	0x5800 1130
DISPC_VID1_CONV_COEF1	RW	32	0x0000 0134	0x5800 1134
DISPC_VID1_CONV_COEF2	RW	32	0x0000 0138	0x5800 1138
DISPC_VID1_CONV_COEF3	RW	32	0x0000 013C	0x5800 113C
DISPC_VID1_CONV_COEF4	RW	32	0x0000 0140	0x5800 1140
DISPC_VID2_BA_j [(1)]	RW	32	0x0000 014C + (0x4 * j)	0x5800 114C + (0x4 * j)
DISPC_VID2_POSITION	RW	32	0x0000 0154	0x5800 1154
DISPC_VID2_SIZE	RW	32	0x0000 0158	0x5800 1158
DISPC_VID2_ATTRIBUTES	RW	32	0x0000 015C	0x5800 115C

Table 11-168. DISPC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DISPC_VID2_BUF_THRESHOLD	RW	32	0x0000 0160	0x5800 1160
DISPC_VID2_BUF_SIZE_STATUS	R	32	0x0000 0164	0x5800 1164
DISPC_VID2_ROW_INC	RW	32	0x0000 0168	0x5800 1168
DISPC_VID2_PIXEL_INC	RW	32	0x0000 016C	0x5800 116C
DISPC_VID2_FIR	RW	32	0x0000 0170	0x5800 1170
DISPC_VID2_PICTURE_SIZE	RW	32	0x0000 0174	0x5800 1174
DISPC_VID2_ACCU_j (1)	RW	32	0x0000 0178 + (0x4 * j)	0x5800 1178 + (0x4 * j)
DISPC_VID2_FIR_COEF_H_i (2)	RW	32	0x0000 0180 + (0x8 * i)	0x5800 1180 + (0x8 * i)
DISPC_VID2_FIR_COEF_HV_i (2)	RW	32	0x0000 0184 + (0x8 * i)	0x5800 1184 + (0x8 * i)
DISPC_VID2_CONV_COEF0	RW	32	0x0000 01C0	0x5800 11C0
DISPC_VID2_CONV_COEF1	RW	32	0x0000 01C4	0x5800 11C4
DISPC_VID2_CONV_COEF2	RW	32	0x0000 01C8	0x5800 11C8
DISPC_VID2_CONV_COEF3	RW	32	0x0000 01CC	0x5800 11CC
DISPC_VID2_CONV_COEF4	RW	32	0x0000 01D0	0x5800 11D0
DISPC_DATA1_CYCLE1	RW	32	0x0000 01D4	0x5800 11D4
DISPC_DATA1_CYCLE2	RW	32	0x0000 01D8	0x5800 11D8
DISPC_DATA1_CYCLE3	RW	32	0x0000 01DC	0x5800 11DC
DISPC_VID1_FIR_COEF_V_i (2)	RW	32	0x0000 01E0 + (0x4 * i)	0x5800 11E0 + (0x4 * i)
DISPC_VID2_FIR_COEF_V_i (2)	RW	32	0x0000 0200 + (0x4 * i)	0x5800 1200 + (0x4 * i)
DISPC_CPR1_COEF_R	RW	32	0x0000 0220	0x5800 1220
DISPC_CPR1_COEF_G	RW	32	0x0000 0224	0x5800 1224
DISPC_CPR1_COEF_B	RW	32	0x0000 0228	0x5800 1228
DISPC_GFX_PRELOAD	RW	32	0x0000 022C	0x5800 122C
DISPC_VID1_PRELOAD	RW	32	0x0000 0230	0x5800 1230
DISPC_VID2_PRELOAD	RW	32	0x0000 0234	0x5800 1234
DISPC_CONTROL2	RW	32	0x0000 0238	0x5800 1238
DISPC_GFX_POSITION2	RW	32	0x0000 0240	0x5800 1240
DISPC_VID1_POSITION2	RW	32	0x0000 0244	0x5800 1244
DISPC_VID2_POSITION2	RW	32	0x0000 0248	0x5800 1248
DISPC_VID3_POSITION2	RW	32	0x0000 024C	0x5800 124C
DISPC_VID3_ACCU_j (1)	RW	32	0x0000 0300 + (0x4 * j)	0x5800 1300 + (0x4 * j)
DISPC_VID3_BA_j (1)	RW	32	0x0000 0308 + (0x4 * j)	0x5800 1308 + (0x4 * j)
DISPC_VID3_FIR_COEF_H_i (2)	RW	32	0x0000 0310 + (0x8 * i)	0x5800 1310 + (0x8 * i)
DISPC_VID3_FIR_COEF_HV_i (2)	RW	32	0x0000 0314 + (0x8 * i)	0x5800 1314 + (0x8 * i)
DISPC_VID3_FIR_COEF_V_i (2)	RW	32	0x0000 0350 + (0x4 * i)	0x5800 1350 + (0x4 * i)
DISPC_VID3_ATTRIBUTES	RW	32	0x0000 0370	0x5800 1370
DISPC_VID3_CONV_COEF0	RW	32	0x0000 0374	0x5800 1374
DISPC_VID3_CONV_COEF1	RW	32	0x0000 0378	0x5800 1378
DISPC_VID3_CONV_COEF2	RW	32	0x0000 037C	0x5800 137C
DISPC_VID3_CONV_COEF3	RW	32	0x0000 0380	0x5800 1380
DISPC_VID3_CONV_COEF4	RW	32	0x0000 0384	0x5800 1384
DISPC_VID3_BUF_SIZE_STATUS	R	32	0x0000 0388	0x5800 1388
DISPC_VID3_BUF_THRESHOLD	RW	32	0x0000 038C	0x5800 138C
DISPC_VID3_FIR	RW	32	0x0000 0390	0x5800 1390
DISPC_VID3_PICTURE_SIZE	RW	32	0x0000 0394	0x5800 1394

Table 11-168. DISPC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DISPC_VID3_PIXEL_INC	RW	32	0x0000 0398	0x5800 1398
DISPC_VID3_POSITION	RW	32	0x0000 039C	0x5800 139C
DISPC_VID3_PRELOAD	RW	32	0x0000 03A0	0x5800 13A0
DISPC_VID3_ROW_INC	RW	32	0x0000 03A4	0x5800 13A4
DISPC_VID3_SIZE	RW	32	0x0000 03A8	0x5800 13A8
DISPC_DEFAULT_COLOR2	RW	32	0x0000 03AC	0x5800 13AC
DISPC_TRANS_COLOR2	RW	32	0x0000 03B0	0x5800 13B0
DISPC_CPR2_COEF_B	RW	32	0x0000 03B4	0x5800 13B4
DISPC_CPR2_COEF_G	RW	32	0x0000 03B8	0x5800 13B8
DISPC_CPR2_COEF_R	RW	32	0x0000 03BC	0x5800 13BC
DISPC_DATA2_CYCLE1	RW	32	0x0000 03C0	0x5800 13C0
DISPC_DATA2_CYCLE2	RW	32	0x0000 03C4	0x5800 13C4
DISPC_DATA2_CYCLE3	RW	32	0x0000 03C8	0x5800 13C8
DISPC_SIZE_LCD2	RW	32	0x0000 03CC	0x5800 13CC
DISPC_TIMING_H2	RW	32	0x0000 0400	0x5800 1400
DISPC_TIMING_V2	RW	32	0x0000 0404	0x5800 1404
DISPC_POL_FREQ2	RW	32	0x0000 0408	0x5800 1408
DISPC_DIVISOR2	RW	32	0x0000 040C	0x5800 140C
DISPC_WB_ACCU_j (1)	RW	32	0x0000 0500 + (0x4 * j)	0x5800 1500 + (0x4 * j)
DISPC_WB_BA_j (1)	RW	32	0x0000 0508 + (0x4 * j)	0x5800 1508 + (0x4 * j)
DISPC_WB_FIR_COEF_H_i (2)	RW	32	0x0000 0510 + (0x8 * i)	0x5800 1510 + (0x8 * i)
DISPC_WB_FIR_COEF_HV_i (2)	RW	32	0x0000 0514 + (0x8 * i)	0x5800 1514 + (0x8 * i)
DISPC_WB_FIR_COEF_V_i (2)	RW	32	0x0000 0550 + (0x4 * i)	0x5800 1550 + (0x4 * i)
DISPC_WB_ATTRIBUTES	RW	32	0x0000 0570	0x5800 1570
DISPC_WB_CONV_COEF0	RW	32	0x0000 0574	0x5800 1574
DISPC_WB_CONV_COEF1	RW	32	0x0000 0578	0x5800 1578
DISPC_WB_CONV_COEF2	RW	32	0x0000 057C	0x5800 157C
DISPC_WB_CONV_COEF3	RW	32	0x0000 0580	0x5800 1580
DISPC_WB_CONV_COEF4	RW	32	0x0000 0584	0x5800 1584
DISPC_WB_BUF_SIZE_STATUS	R	32	0x0000 0588	0x5800 1588
DISPC_WB_BUF_THRESHOLD	RW	32	0x0000 058C	0x5800 158C
DISPC_WB_FIR	RW	32	0x0000 0590	0x5800 1590
DISPC_WB_PICTURE_SIZE	RW	32	0x0000 0594	0x5800 1594
DISPC_WB_PIXEL_INC	RW	32	0x0000 0598	0x5800 1598
DISPC_WB_ROW_INC	RW	32	0x0000 05A4	0x5800 15A4
DISPC_WB_SIZE	RW	32	0x0000 05A8	0x5800 15A8
DISPC_VID1_BA_UV_j (1)	RW	32	0x0000 0600 + (0x4 * j)	0x5800 1600 + (0x4 * j)
DISPC_VID2_BA_UV_j (1)	RW	32	0x0000 0608 + (0x4 * j)	0x5800 1608 + (0x4 * j)
DISPC_VID3_BA_UV_j (1)	RW	32	0x0000 0610 + (0x4 * j)	0x5800 1610 + (0x4 * j)
DISPC_WB_BA_UV_j (1)	RW	32	0x0000 0618 + (0x4 * j)	0x5800 1618 + (0x4 * j)
DISPC_CONFIG2	RW	32	0x0000 0620	0x5800 1620
DISPC_VID1_ATTRIBUTES2	RW	32	0x0000 0624	0x5800 1624
DISPC_VID2_ATTRIBUTES2	RW	32	0x0000 0628	0x5800 1628
DISPC_VID3_ATTRIBUTES2	RW	32	0x0000 062C	0x5800 162C
DISPC_GAMMA_TABLE0	W	32	0x0000 0630	0x5800 1630

Table 11-168. DISPC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DISPC_GAMMA_TABLE1	W	32	0x0000 0634	0x5800 1634
DISPC_GAMMA_TABLE2	W	32	0x0000 0638	0x5800 1638
DISPC_VID1_FIR2	RW	32	0x0000 063C	0x5800 163C
DISPC_VID1_ACCU2_j [(1)]	RW	32	0x0000 0640 + (0x4 * j)	0x5800 1640 + (0x4 * j)
DISPC_VID1_FIR_COEF_H2_i [(2)]	RW	32	0x0000 0648 + (0x8 * i)	0x5800 1648 + (0x8 * i)
DISPC_VID1_FIR_COEF_HV2_i [(2)]	RW	32	0x0000 064C + (0x8 * i)	0x5800 164C + (0x8 * i)
DISPC_VID1_FIR_COEF_V2_i [(2)]	RW	32	0x0000 0688 + (0x4 * i)	0x5800 1688 + (0x4 * i)
DISPC_VID2_FIR2	RW	32	0x0000 06A8	0x5800 16A8
DISPC_VID2_ACCU2_j [(1)]	RW	32	0x0000 06AC + (0x4 * j)	0x5800 16AC + (0x4 * j)
DISPC_VID2_FIR_COEF_H2_i [(2)]	RW	32	0x0000 06B4 + (0x8 * i)	0x5800 16B4 + (0x8 * i)
DISPC_VID2_FIR_COEF_HV2_i [(2)]	RW	32	0x0000 06B8 + (0x8 * i)	0x5800 16B8 + (0x8 * i)
DISPC_VID2_FIR_COEF_V2_i [(2)]	RW	32	0x0000 06F4 + (0x4 * i)	0x5800 16F4 + (0x4 * i)
DISPC_VID3_FIR2	RW	32	0x0000 0724	0x5800 1724
DISPC_VID3_ACCU2_j [(1)]	RW	32	0x0000 0728 + (0x4 * j)	0x5800 1728 + (0x4 * j)
DISPC_VID3_FIR_COEF_H2_i [(2)]	RW	32	0x0000 0730 + (0x8 * i)	0x5800 1730 + (0x8 * i)
DISPC_VID3_FIR_COEF_HV2_i [(2)]	RW	32	0x0000 0734 + (0x8 * i)	0x5800 1734 + (0x8 * i)
DISPC_VID3_FIR_COEF_V2_i [(2)]	RW	32	0x0000 0770 + (0x4 * i)	0x5800 1770 + (0x4 * i)
DISPC_WB_FIR2	RW	32	0x0000 0790	0x5800 1790
DISPC_WB_ACCU2_j [(1)]	RW	32	0x0000 0794 + (0x4 * j)	0x5800 1794 + (0x4 * j)
DISPC_WB_FIR_COEF_H2_i [(2)]	RW	32	0x0000 07A0 + (0x8 * i)	0x5800 17A0 + (0x8 * i)
DISPC_WB_FIR_COEF_HV2_i [(2)]	RW	32	0x0000 07A4 + (0x8 * i)	0x5800 17A4 + (0x8 * i)
DISPC_WB_FIR_COEF_V2_i [(2)]	RW	32	0x0000 07E0 + (0x4 * i)	0x5800 17E0 + (0x4 * i)
DISPC_GLOBAL_BUFFER	RW	32	0x0000 0800	0x5800 1800
DISPC_DIVISOR	RW	32	0x0000 0804	0x5800 1804
DISPC_WB_ATTRIBUTES2	RW	32	0x0000 0810	0x5800 1810
DISPC_DEFAULT_COLOR3	RW	32	0x0000 0814	0x5800 1814
DISPC_TRANS_COLOR3	RW	32	0x0000 0818	0x5800 1818
DISPC_CPR3_COEF_B	RW	32	0x0000 081C	0x5800 181C
DISPC_CPR3_COEF_G	RW	32	0x0000 0820	0x5800 1820
DISPC_CPR3_COEF_R	RW	32	0x0000 0824	0x5800 1824
DISPC_DATA3_CYCLE1	RW	32	0x0000 0828	0x5800 1828
DISPC_DATA3_CYCLE2	RW	32	0x0000 082C	0x5800 182C
DISPC_DATA3_CYCLE3	RW	32	0x0000 0830	0x5800 1830
DISPC_SIZE_LCD3	RW	32	0x0000 0834	0x5800 1834
DISPC_DIVISOR3	RW	32	0x0000 0838	0x5800 1838
DISPC_POL_FREQ3	RW	32	0x0000 083C	0x5800 183C
DISPC_TIMING_H3	RW	32	0x0000 0840	0x5800 1840
DISPC_TIMING_V3	RW	32	0x0000 0844	0x5800 1844
DISPC_CONTROL3	RW	32	0x0000 0848	0x5800 1848
DISPC_CONFIG3	RW	32	0x0000 084C	0x5800 184C
DISPC_GAMMA_TABLE3	W	32	0x0000 0850	0x5800 1850
DISPC_BA0_FLIPIMMEDIATE_EN	RW	32	0x0000 0854	0x5800 1854
DISABLE_MSTANDBY_ENHANCEMENT	RW	32	0x0000 0858	0x5800 1858
DISPC_GLOBAL_MFLAG_ATTRIBUTE	RW	32	0x0000 085C	0x5800 185C
DISPC_GFX_MFLAG_THRESHOLD	RW	32	0x0000 0860	0x5800 1860

Table 11-168. DISPC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DISPC_VID1_MFLAG_THRESHOLD	RW	32	0x0000 0864	0x5800 1864
DISPC_VID2_MFLAG_THRESHOLD	RW	32	0x0000 0868	0x5800 1868
DISPC_VID3_MFLAG_THRESHOLD	RW	32	0x0000 086C	0x5800 186C
DISPC_WB_MFLAG_THRESHOLD	RW	32	0x0000 0870	0x5800 1870

(1) j = 0 to 1
(2) i = 0 to 7

11.2.6.3.2 DISPC Register Description

Table 11-169. DISPC_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5800 1000	**Instance**	DISPC
Description	IP Revision		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See(1).

(1) TI internal data

Table 11-170. DISPC_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5800 1010	**Instance**	DISPC
Description	This register allows to control various parameters of the OCP interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		MIDLEMODE		RESERVED		CLOCKACTIVITY		RESERVED		WARMRESET	SIDLEMODE		ENWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Write 0s for future compatibility. Reads returns 0.	R	0x00000
13:12	MIDLEMODE	Master interface power management, standby/wait control 0x0: Force-standby. MStandby is only asserted when the module is disabled. MStandby is only asserted when the module is disabled. 0x1: No-Standby: MStandby is never asserted. 0x2: Smart-Standby. MStandby is asserted based on the internal activity of the module. 0x3: Reserved	RW	0x0
11:10	RESERVED	Write 0s for future compatibility. Reads returns 0	R	0x0
9:8	CLOCKACTIVITY	Clocks activity during wake up mode period 0x0: OCP and functional clocks can be switched off. 0x1: Functional clocks can be switched off and OCP clocks are maintained during wake up period. 0x2: OCP clocks can be switched off and Functional clocks are maintained during wake up period. 0x3: OCP and functional clocks are maintained during wake-up period.	RW	0x0
7:6	RESERVED	Write 0s for future compatibility. Reads returns 0	R	0x0

Bits	Field Name	Description	Type	Reset
5	WARMRESET	Warm reset. Set this bit to 1 triggers a module warm reset. The bit is automatically reset by the hardware. During reads, it always returns 0. The warm reset keep the configuration registers unchanged. 0x0: Normal mode 0x1: The warm reset is set.	RW	0
4:3	SIDLEMODE	Slave interface power management, Idle req/ack control 0x0: Force-idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged. 0x2: Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module. 0x3: Reserved	RW	0x0
2	ENWAKEUP	WakeUp feature control 0x0: Wakeup is disabled. 0x1: Wakeup is enabled.	RW	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	Internal interface clock gating strategy 0x0: Interface clock is free-running. 0x1: Automatic interface L3_MAIN gating strategy is applied, based on the OCP interface activity. Automatic functional clock gating is also applied to the functional clock based on the module activity (for instance DISPC_<pipe>_ATTRIBUTES.ENABLE).	RW	1

Table 11-171. DISPC_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x5800 1014	**Instance**	DISPC
Description	This register provides status information about the module, excluding the interrupt status information.		
Type	R		

31–1	0
RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is on-going. Read 0x1: Reset completed	R	1

Table 11-172. DISPC_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	0x5800 1018	**Instance**	DISPC
Description	This register regroups all the status of the module internal events that generate an interrupt. Write 1 to a given bit resets this bit		

Table 11-172. DISPC_IRQSTATUS (continued)

Type	RW

31	30	29	28	27	26	25	24
FLIPIMMEDIATEDONE_IRQ	FRAMEDONE3_IRQ	ACBIASCOUNTSTATUS3_IRQ	VSYNC3_IRQ	SYNCLOST3_IRQ	WBINCOMPLETEERROR_IRQ	WBBUFFEROVERFLOW_IRQ	FRAMEDONETV_IRQ

23	22	21	20	19	18	17	16
FRAMEDONEWB_IRQ	FRAMEDONE2_IRQ	ACBIASCOUNTSTATUS2_IRQ	VID3BUFFERUNDERFLOW_IRQ	VID3ENDWINDOW_IRQ	VSYNC2_IRQ	SYNCLOST2_IRQ	WAKEUP_IRQ

15	14	13	12	11	10	9	8
SYNCLOSTTV_IRQ	SYNCLOST1_IRQ	VID2ENDWINDOW_IRQ	VID2BUFFERUNDERFLOW_IRQ	VID1ENDWINDOW_IRQ	VID1BUFFERUNDERFLOW_IRQ	OCPERROR_IRQ	PALETTEGAMMALOADING_IRQ

7	6	5	4	3	2	1	0
GFXENDWINDOW_IRQ	GFXBUFFERUNDERFLOW_IRQ	PROGRAMMEDLINENUMBER_IRQ	ACBIASCOUNTSTATUS1_IRQ	EVSYNC_ODD_IRQ	EVSYNC_EVEN_IRQ	VSYNC1_IRQ	FRAMEDONE1_IRQ

Bits	Field Name	Description	Type	Reset
31	FLIPIMMEDIATEDONE_IRQ	Flip Immediate Done. The DMA engine has acknowledged the immediate BA change, and software can write the new BA0. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
30	FRAMEDONE3_IRQ	Frame done for the third LCD. The third LCD output has been disabled by user. All the data have been sent. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
29	ACBIASCOUNT STATUS3_IRQ	AC bias count status for the third LCD 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
28	VSYNC3_IRQ	Vertical synchronization for the third LCD 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
27	SYNCLOST3_IRQ	Synchronization lost on the third LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the third LCD output. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
26	WBUNCOMPLETE ERROR_IRQ	Write-back DMA buffer is flushed before it is completely drained. In WB capture mode, if the new frame starts before the WB DMA buffers are fully drained (onto external memory), then the contents of the WB DMA buffers are lost (implying last few pixels/lines are corrupted in the captured frame in memory). This interrupt is an indication of that case and will trigger every frame 0x0: READS: Event is false. WRITES: Status bit unchanged 0x1: READS: Event is true (Pending) WRITES: Status bit is reset	RW W1toClr	0
25	WBBUFFER OVERFLOW_IRQ	Write-back DMA buffer overflow. The DMA buffer is full. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
24	FRAME DONETV_IRQ	Frame done for the TV. The TV output has been disabled by user. All the data have been sent. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
23	FRAME DONEWB_IRQ	Frame done for the write-back channel. The write-back channel has output the frame. All the data of the frame have been sent to the memory. There is no pending data inside the DMA engine for the write-back channel to be transferred to memory. It is available only when the write-back pipeline transfers back to memory the output of one of the pipelines. In case of overlay capture, the interrupt is not generated and the user shall use the FrameDone for the corresponding captured output. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
22	FRAME DONE2_IRQ	Frame done for the secondary LCD. The secondary LCD output has been disabled by user. All the data have been sent. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
21	ACBIASCOUNT STATUS2_IRQ	AC bias count status for the secondary LCD 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
20	VID3BUFFER UNDERFLOW_IRQ	Video 3 DMA buffer underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
19	VID3END WINDOW_IRQ	The end of the video 3 window has been reached. It is detected by the overlay manager when the full video 3 has been displayed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
18	VSYNC2_IRQ	Vertical synchronization for the secondary LCD 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
17	SYNC LOST2_IRQ	Synchronization lost on the secondary LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the secondary LCD output. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
16	WAKEUP_IRQ	Wakeup 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
15	SYNCLOST TV_IRQ	Synchronization lost on the TV output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the TV output. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
14	SYNC LOST1_IRQ	Synchronizationl ost on the primary LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the primary LCD output. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
13	VID2END WINDOW_IRQ	The end of the video 2 Window has been reached. It is detected by the overlay manager when the full video 2 has been displayed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
12	VID2BUFFER UNDERFLOW_IRQ	Video 2 DMA buffer underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
11	VID1END WINDOW_IRQ	The end of the video 1 Window has been reached. It is detected by the overlay manager when the full video 1 has been displayed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	VID1BUFFER UNDERFLOW_IRQ	Video 1 DMA buffer underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
9	OCPERROR_IRQ	OCP error. L3_MAIN Interconnect has sent SResp=ERR. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
8	PALETTEGAMMA LOADING_IRQ	Palette Gamma loading status. The palette used as Color Look Up Table (CLUT) for the graphics BITMAP formats (1-, 2-, 4-, or 4-bpp) or as gamma table for the overlay output for the primary LCD output has been loaded successfully. **NOTE: CLUT and BITMAP formats are not supported in this family of devices.** 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
7	GFXEND WINDOW_IRQ	The end of the graphics wndow has been reached. It is detected by the overlay manager when the full graphics has been displayed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
6	GFXBUFFER UNDERFLOW_IRQ	Graphics DMA buffer underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
5	PROGRAMMED LINENUMBER_IRQ	Programmed line number. It indicates that the scan of the primary LCD has reached the programmed user line number. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
4	ACBIASCOUNT STATUS1_IRQ	AC bias count status for the primary LCD 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
3	EVSYNC_ ODD_IRQ	VSYNC for odd field from the TV encoder (HDMI) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
2	EVSYNC_ EVEN_IRQ	VSYNC for even field from the TV encoder (HDMI) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
1	VSYNC1_IRQ	Vertical synchronization for the primary LCD. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	FRAME DONE1_IRQ	Frame done for the primary LCD. The primary LCD output has been disabled by user. All the data have been sent. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Table 11-173. DISPC_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	0x5800 101C	**Instance**	DISPC
Description	This register allows to mask/unmask the module internal sources of interrupt, on an event-by-event basis		
Type	RW		

31	30	29	28	27	26	25	24
FLIPIMMEDIATEDONE_EN	FRAMEDONE3_EN	ACBIASCOUNTSTATUS3_EN	VSYNC3_EN	SYNCLOST3_EN	WBUNCOMPLETEERROR_EN	WBBUFFEROVERFLOW_EN	FRAMEDONETV_EN

23	22	21	20	19	18	17	16
FRAMEDONEWB_EN	FRAMEDONE2_EN	ACBIASCOUNTSTATUS2_EN	VID3BUFFERUNDERFLOW_EN	VID3ENDWINDOW_EN	VSYNC2_EN	SYNCLOST2_EN	WAKEUP_EN

15	14	13	12	11	10	9	8
SYNCLOSTTV_EN	SYNCLOST1_EN	VID2ENDWINDOW_EN	VID2BUFFERUNDERFLOW_EN	ENDVID1WINDOW_EN	VID1BUFFERUNDERFLOW_EN	OCPERROR_EN	PALETTEGAMMA_EN

7	6	5	4	3	2	1	0
GFXENDWINDOW_EN	GFXBUFFERUNDERFLOW_EN	PROGRAMMEDLINENUMBER_EN	ACBIASCOUNTSTATUS1_EN	EVSYNC_ODD_EN	EVSYNC_EVEN_EN	VSYNC1_EN	FRAMEDONE_EN

Bits	Field Name	Description	Type	Reset
31	FLIPIMMEDIATEDONE_EN	Flip Immediate Done. The DMA engine has acknowledged the immediate BA change, and software can write the new BA0. 0x0: FrameDone for the primary LCD output is masked 0x1: FrameDone for the primary LCD output generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
30	FRAME DONE3_EN	Frame done for the third LCD. The third LCD output has been disabled by user. All the data have been sent. 0x0: Frame Done for the secondary LCD is masked. 0x1: Frame Done for the secondary LCD generates an interrupt when it occurs.	RW	0
29	ACBIASCOUNT STATUS3_EN	AC Bias count status for the third LCD 0x0: ACBiasCountStatus for the secondary LCD output is masked 0x1: ACBiasCountStatus for the secondary LCD output generates an interrupt when it occurs	RW	0
28	VSYNC3_EN	Vertical synchronization for the third LCD 0x0: VSYNC for the secondary LCD output is masked. 0x1: VSYNC for the secondary LCD output generates an interrupt when it occurs.	RW	0
27	SYNC LOST3_EN	Synchronization lost on the third LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the third LCD output. 0x0: Synchronization Lost on the secondary LCD output is masked. 0x1: Synchronization Lost on the secondary LCD output generates an interrupt when it occurs.	RW	0
26	WBUNCOMPLETE ERROR_EN	The write back buffer has been flushed before been fully drained. Enable. 0x0: Interrupt is masked. 0x1: Interrupt is enabled.	RW	0
25	WBBUFFER OVERFLOW_EN	Write-back DMA buffer overflow. The DMA buffer is full. 0x0: WBBufferOverflow is masked. 0x1: WBBufferOverflow generates an interrupt when it occurs.	RW	0
24	FRAME DONETV_EN	Frame done for the TV. The TV output has been disabled by user. All the data have been sent. 0x0: Frame Done for the TV output is masked. 0x1: Frame Done for the TV output generates an interrupt when it occurs.	RW	0
23	FRAME DONEWB_EN	Frame done for the write-back channel. The write-back channel has output the frame. All the data have been sent for the frame have been sent to the memory. There is no pending data inside the DMA engine for the write-back channel to be transferred to memory. 0x0: Frame done for the write-back is masked. 0x1: Frame done for the write-back generates an interrupt when it occurs.	RW	0
22	FRAME DONE2_EN	Frame done for the secondary LCD. The secondary LCD output has been disabled by user. All the data have been sent. 0x0: Frame done for the secondary LCD is masked. 0x1: Frame done for the secondary LCD generates an interrupt when it occurs.	RW	0
21	ACBIASCOUNT STATUS2_EN	AC Bias count status for the secondary LCD 0x0: ACBiasCountStatus for the secondary LCD output is masked. 0x1: ACBiasCountStatus for the secondary LCD output generates an interrupt when it occurs.	RW	0

Bits	Field Name	Description	Type	Reset
20	VID3BUFFER UNDERFLOW_EN	Video 3 DMA Buffer Underflow. The DMA buffer is not necessary empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: Vid3BufferUnderflow is masked. 0x1: Vid3BufferUnderflow generates an interrupt when it occurs.	RW	0
19	VID3END WINDOW_EN	The end of the video 3 window has been reached. It is detected by the overlay manager when the full video 3 has been displayed. 0x0: Vid3EndWindow is masked. 0x1: Vid3EndWindow generates an interrupt when it occurs.	RW	0
18	VSYNC2_EN	Vertical synchronization for the secondary LCD 0x0: VSYNC for the secondary LCD output is masked. 0x1: VSYNC for the secondary LCD output generates an interrupt when it occurs.	RW	0
17	SYNC LOST2_EN	Synchronization lost on the secondary LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the secondary LCD output. 0x0: Synchronization Lost on the secondary LCD output is masked. 0x1: Synchronization Lost on the secondary LCD output generates an interrupt when it occurs.	RW	0
16	WAKEUP_EN	Wake up mask 0x0: WakeUp is masked. 0x1: WakeUp generates an interrupt when it occurs.	RW	0
15	SYNC LOSTTV_EN	Synchronization lost on the TV output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the TV output. 0x0: Synchronization Lost on the TV output is masked. 0x1: Synchronization Lost on the TV output generates an interrupt when it occurs.	RW	0
14	SYNC LOST1_EN	Synchronization lost for the primary LCD 0x0: SyncLost for the primary LCD output is masked. 0x1: SyncLost for the primary LCD output generates an interrupt when it occurs.	RW	0
13	VID2END WINDOW_EN	The end of the video 2 Window has been reached. It is detected by the overlay manager when the full video 2 has been displayed. 0x0: Vid2EndWindow is masked. 0x1: Vid2EndWindow generates an interrupt when it occurs.	RW	0
12	VID2BUFFER UNDERFLOW_EN	Video 2 DMA buffer underflow. The DMA buffer is not necessary empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: Vid2BufferUnderflow is masked. 0x1: Vid2BufferUnderflow generates an interrupt when it occurs.	RW	0

Bits	Field Name	Description	Type	Reset
11	ENDVID1 WINDOW_EN	The end of the video 1 window has been reached. It is detected by the overlay manager when the full video 1 has been displayed. 0x0: EndVid1Window is masked. 0x1: EndVid1Window generates an interrupt when it occurs.	RW	0
10	VID1BUFFER UNDERFLOW_EN	Video 1 DMA buffer underflow. The DMA buffer is not necessary empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: Vid1bufferunderflow is masked. 0x1: Vid1bufferunderflow generates an interrupt when it occurs.	RW	0
9	OCPERROR_EN	OCP Error. L3_MAIN Interconnect has sent SResp=ERR. 0x0: OCPError is masked. 0x1: OCPError generates an interrupt when it occurs.	RW	0
8	PALETTE GAMMA_EN	Palette gamma loading mask. The palette used as Color Look Up Table (CLUT) for the graphics BITMAP formats (1-, 2-, 4-, or 4-bpp) or as gamma table for the overlay output for the primary LCD output has been loaded successfully. **NOTE: CLUT and BITMAP formats are not supported in this family of devices.** 0x0: PaletteGamma is masked. 0x1: PaletteGamma generates an interrupt when it occurs.	RW	0
7	GFXEND WINDOW_EN	The end of the graphics Window has been reached. It is detected by the overlay manager when the full graphics has been displayed. 0x0: GfxEndWindow is masked. 0x1: GfxEndWindow generates an interrupt when it occurs.	RW	0
6	GFXBUFFER UNDERFLOW_EN	Graphics DMA Buffer Underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: GfxBufferUnderflow is masked. 0x1: GfxBufferUnderflow generates an interrupt when it occurs.	RW	0
5	PROGRAMMED LINENUMBER_EN	Programmed Line Number. It indicates that the scan of the primary LCD has reached the programmed user line number. 0x0: ProgrammedLineNumber is masked. 0x1: ProgrammedLineNumber generates an interrupt when it occurs.	RW	0
4	ACBIASCOUNT STATUS1_EN	AC Bias count status for the primary LCD 0x0: ACBiascountstatus for the primary LCD output is masked. 0x1: ACBiascountstatus for the primary LCD output generates an interrupt when it occurs.	RW	0
3	EVSYNC_ODD_EN	VSYNC for odd field from the TV encoder (HDMI) 0x0: EVSYNC_ODD for the TV output is masked. 0x1: EVSYNC_ODD for the TV output generates an interrupt when it occurs.	RW	0

Bits	Field Name	Description	Type	Reset
2	EVSYNC_EVEN_EN	VSYNC for even field from the TV encoder (HDMI) 0x0: EVSYNC_EVEN for the TV output is masked. 0x1: EVSYNC_EVEN for the TV output generates an interrupt when it occurs.	RW	0
1	VSYNC1_EN	Vertical synchronization for the primary LCD. 0x0: VSYNC for the primary LCD output is masked. 0x1: VSYNC for the primary LCD output generates an interrupt when it occurs.	RW	0
0	FRAMEDONE_EN	Frame done for the primary LCD. The primary LCD output has been disabled by user. All the data have been sent. 0x0: Frame Done for the primary LCD output is masked. 0x1: FrameDone for the primary LCD output generates an interrupt when it occurs.	RW	0

Table 11-174. DISPC_CONTROL1

Address Offset	0x0000 0040		
Physical Address	0x5800 1040	**Instance**	DISPC
Description	The control register configures the Display Controller module for the primary LCD and TV outputs.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPATIALTEMPORALDITHERINGFRAMES		LCDENABLEPOL	LCDENABLESIGNAL	PCKFREEENABLE	TDMUNUSEDBITS		TDMCYCLEFORMAT		TDMPARALLELMODE		TDMENABLE	HT			GPOUT1	GPOUT0	GPIN1	GPIN0	OVERLAYOPTIMIZATION	STALLMODE	RESERVED	TFTDATALINES		STDITHERENABLE	GOTV	GOLCD	M8B	STNTFT	MONOCOLOR	TVENABLE	LCDENABLE

Bits	Field Name	Description	Type	Reset
31:30	SPATIALTEMPORAL DITHERINGFRAMES	Spatial/temporal dithering number of frames for the primary LCD output wr: VFP start period of primary LCD 0x0: Spatial only 0x1: Spatial and temporal over 2 frames 0x2: Spatial and temporal over 4 frames 0x3: Reserved	RW	0x0
29	LCDENABLEPOL	Write 0s for future compatibility. Reads return 0.	R	0
28	LCDENABLESIGNAL	Write 0s for future compatibility. Reads return 0.	R	0
27	PCKFREEENABLE	Write 0s for future compatibility. Reads return 0.	R	0
26:25	TDMUNUSEDBITS	State of unused bits (TDM mode only) for the primary LCD output. wr: VFP start period of primary LCD 0x0: Low level (0) 0x1: High level (1) 0x2: Unchanged from previous state 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
24:23	TDMCYCLEFORMAT	Cycle format (TDM mode only) for the primary LCD output WR: VFP start period of primary LCD 0x0: 1 cycle for 1 pixel 0x1: 2 cycles for 1 pixel 0x2: 3 cycles for 1 pixel 0x3: 3 cycles for 2 pixels	RW	0x0
22:21	TDMPARALLELMODE	Output interface width (TDM mode only) for the primary LCD output WR: VFP start period of primary LCD 0x0: 8-bit parallel output interface selected 0x1: 9-bit parallel output interface selected 0x2: 12-bit parallel output interface selected 0x3: 16-bit parallel output interface selected	RW	0x0
20	TDMENABLE	Enable the multiple cycle format for the primary LCD output. WR: VFP start period of primary LCD 0x0: TDM disabled 0x1: TDM enabled	RW	0
19:17	HT	Hold time for TV output WR: EVSYNC Encoded value (from 1 to 8) to specify the number of external digital clock periods to hold the data (programmed value = value minus 1)	RW	0x0
16	GPOUT1	General purpose output signal I WR: immediate 0x0: The GPout1 is reset. 0x1: The GPout1 is set.	RW	0
15	GPOUT0	General Purpose Output Signal WR:immediate 0x0: The GPout0 is reset. 0x1: The GPout0 is set.	RW	0
14	GPIN1	General purpose input signal WR: immediately Read 0x0: The GPin1 has been reset. Read 0x1: The GPin1 has been set.	R	0
13	GPIN0	General purpose input signal WR: immediately Read 0x0: The GPin0 has been reset. Read 0x1: The GPin0 has been set.	R	0
12	OVERLAYOPTI MIZATION	Overlay optimization for the primary LCD output WR: VFP start period of the primary LCD 0x0: All the data for all the enabled pipelines are fetched from memory regardless of the overlay/alpha blending configuration. 0x1: The data not used by the overlay manager because of overlap between layers with no alpha blending between them must not be fetched from memory in order to optimize the bandwidth.	RW	0

Bits	Field Name	Description	Type	Reset
11	STALLMODE	STALL mode for the primary LCD output wr: VFP start period of primary LCD 0x0: Normal mode selected 0x1: STALL mode selected. The Display Controller sends the data without considering the VSYNC/HSYNC. The LCD output is disabled at the end of the transfer of the frame. To generate a new frame, the software must re-enable the LCD output.	RW	0
10	RESERVED	Reserved	R	0
9:8	TFTDATALINES	Number of lines of the primary LCD interface WR: VFP start period of primary LCD 0x0: 12-bit output aligned on the LSB of the pixel data interface 0x1: 16-bit output aligned on the LSB of the pixel data interface 0x2: 18-bit output aligned on the LSB of the pixel data interface 0x3: 24-bit output aligned on the LSB of the pixel data interface	RW	0x0
7	STDITHERENABLE	Spatial temporal dithering enable for the primary LCD output WR: VFP start period of primary LCD 0x0: Spatial/temporal dithering logic disabled 0x1: Spatial/temporal dithering logic enabled	RW	0
6	GOTV	GO command for the TV output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the TV output. WR: immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) associated with the TV output using the user values. The hardware resets the bit when the update is completed. 0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the TV output and the hardware can update the internal registers at the external VSYNC.	RW	0
5	GOLCD	GO command for the primary LCD output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the primary LCD output. WR: immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the LCD output using the user values. The hardware resets the bit when the update is completed. 0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the LCD output and the hardware can update the internal registers at the VFP start period	RW	0
4	M8B	Mono 8-bit mode of the primary LCD wr: VFP start period of primary LCD output 0x0: Reserved 0x1: Reserved	RW	0

Bits	Field Name	Description	Type	Reset
3	STNTFT	LCD Display type of the primary LCD WR: VFP start period of primary LCD output 0x0: STN display operation enabled. STN dither logic is enabled. 0x1: Active or TFT display operation enabled. STN Dither logic and output FIFO bypassed.	RW	0
2	MONOCOLOR	Monochrome/color selection for the primary LCD WR: VFP start period of primary LCD output 0x0: Color operation enabled (STN mode only) 0x1: Monochrome operation enabled (STN mode only)	RW	0
1	TVENABLE	Enable the TV output wr: immediate effect only occurs at the end of the current frame. 0x0: TV output disabled (at the end of the current field if interlace output when the bit is reset) 0x1: TV output enabled	RW	0
0	LCDENABLE	Enable the primary LCD outputs wr: immediate Effect only occurs at the end of the current frame 0x0: LCD output disabled (at the end of the frame when the bit is reset) 0x1: LCD output enabled	RW	0

Table 11-175. DISPC_CONFIG1

Address Offset	0x0000 0044		
Physical Address	0x5800 1044	**Instance**	DISPC
Description	The control register configures the Display Controller module for the primary LCD output and TV output. Shadow register, updated on VFP start period of primary LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		TVINTERLEAVE		PLCDINTERLEAVE		FULLRANGE	COLORCONVENABLE	FIDFIRST	OUTPUTMODEENABLE	BT1120ENABLE	BT656ENABLE	TVALPHABLENDERENABLE	LCDALPHABLENDERENABLE	BUFFERFILLING	BUFFERHANDCHECK	CPR	BUFFERMERGE	TCKTVSELECTION	TCKTVENABLE	TCKLCDSELECTION	TCKLCDENABLE	GAMMATABLEENABLE	ACBIASGATED	VSYNCGATED	HSYNCGATED	PIXELCLOCKGATED	PIXELDATAGATED	PALETTEGAMMATABLE	LOADMODE		PIXELGATED

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
29:28	TVINTERLEAVE	TV Interleave Pattern	RW	0x0
27:26	PLCDINTERLEAVE	pLCD Interleave Pattern	RW	0x0
25	FULLRANGE	Color Space Conversion full range setting. wr: VFP start of primary LCD 0x0: Limited range selected. 0x1: Full range selected.	RW	0

Bits	Field Name	Description	Type	Reset
24	COLORCONV ENABLE	Enable the color space conversion. It shall be reset when CPR bit field is set to 0x1. wr: VFP start of primary LCD 0x0: Disable Color Space Conversion RGB to YUV 0x1: Enable Color Space Conversion RGB to YUV	RW	0
23	FIDFIRST	Selects the first field to output in case of interlace mode. In case of progressive mode, the value is not used. wr: VFP start of primary LCD 0x0: First field is even. 0x1: Odd field is first.	RW	0
22	OUTPUTMODE ENABLE	Selects between progressive and interlace mode for the primary LCD output. wr: VFP start of primary LCD 0x0: Progressive mode selected. 0x1: Interlace mode selected.	RW	0
21	BT1120ENABLE	Selects BT.1120 format on the primary LCD output. It is not posssible to enable BT.656 and BT.1120 at the same time one the same LCD output. wr: VFP start of primary LCD 0x0: BT.1120 is disabled 0x1: BT.1120 is enabled.	RW	0
20	BT656ENABLE	Selects BT.656 format on the primary LCD output. It is not posssible to enable BT.656 and BT.1120 at the same time one the same LCD output. wr: VFP start of primary LCD 0x0: BT.656 is disabled. 0x1: BT.656 is enabled.	RW	0
19	TVALPHABLENDER ENABLE	Selects the alpha blender overlay manager for the TV output instead of the color key alpha blender (LCD output). The bit field is deprecated. It is present for software backward compatibility only. When it is enabled, the Z-order defined in each ATTRIBUTES registers for only the pipelines associated pipeline connected to the TV output are invalid and replaced by the following: graphics z-order = 3, video3 z-order = 2, video2 z-order =1 and video1 z-order=0 If it disabled, the z-order and z-order enable bit fields defined in each ATTRIBUTES register are used. wr: EVSYNC start of primary LCD 0x0: Alpha blender is disabled. 0x1: The alpha blender is enabled.	RW	0
18	LCDALPHABLENDER ENABLE	Selects the alpha blender overlay manager for the primary LCD output instead of the color key alpha blender (LCD output). The bit field is deprecated. It is present for software backward compatibility only. When it is enabled, the Z-order defined in each ATTRIBUTES registers for only the pipelines associated with the primary LCD output are invalid and replaced by the following: graphics z-order = 3, video3 z-order = 2, video2 z-order =1 and video1 z-order=0 If it disabled, the z-order and z-order enable bit fields defined in each ATTRIBUTES register are used. wr: VFP start of primary LCD 0x0: Alpha blender is disabled. The color key alpha blending is used. 0x1: The alpha blender is enabled.	RW	0

Bits	Field Name	Description	Type	Reset
17	BUFFERFILLING	Controls if the DMA buffers are refilled only when the LOW threshold is reached or if all DMA buffers are refilled when at least one of them reaches the LOW threshold. wr: immediate 0x0: Each DMA buffer is refilled when it reaches LOW threshold. 0x1: All DMA buffers are refilled up to high threshold when at least one of them reaches the LOW threshold. (only active DMA buffers shall be considered and when reaching the end of the frame the DMA buffer goes to empty condition so no need to fill it again).	RW	0
16	RESERVED	Write 0s for future compatibility. Reads return 0.	RW	0
15	CPR	Color phase rotation control (primary LCD output). It shall be reset when ColorConvEnable bit field is set to 1 wr: VFP start period of primary LCD output 0x0: Color Phase Rotation Disabled 0x1: Color Phase Rotation Enabled	RW	0
14	BUFFERMERGE	Buffer merge control wr: EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory or VFP When enabled, the DISPC_GLOBAL_BUFFER register is ignored. This bit must be set to zero when the write back channel is used. When DISPC_CONTROL2.GOWB is used BUFFERMERGE MUST be zero. When DISPC_CONTROL2.GOWB is used BUFFERMERGE MUST be zero. WR: immediate 0x0: DMA buffer merge disabled Each DMA buffer is dedicated to one pipeline. 0x1: DMA buffer merge enabled All the DMA buffers are merged into a single one to be used by the single active pipeline.	RW	0
13	TCKTV SELECTION	Transparency color key selection (TV output) wr: EVSYNC 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0
12	TCKTVENABLE	Transparency color key enabled (TV output) WR: EVSYNC 0x0: Disable the transparency color key for the TV output 0x1: Enable the transparency color key for the TV output	RW	0
11	TCKLCD SELECTION	Transparency color key selection (primary LCD output) wr: VFP start period of primary LCD output 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0
10	TCKLCDENABLE	Transparency color key enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: Disable the transparency color key for the LCD 0x1: Enable the transparency color key for the LCD	RW	0

Bits	Field Name	Description	Type	Reset
9	GAMATABLE ENABLE	For backward compatibility, an enable bit has been added on the 2 additional gamma tables (secondary display and TV). Gamma table of LCD1 is always enabled. 0x0: Gamma table LCD2 and TV are bypassed 0x1: Gamma table LCD2 and TV are enabled	RW	0
8	ACBIASGATED	ACBias Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: AcBias gated disabled 0x1: AcBias gated enabled	RW	0
7	VSYNCGATED	VSYNC Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: VSYNC gated disabled 0x1: VSYNC gated enabled	RW	0
6	HSYNCGATED	HSYNC Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: HSYNC gated disabled 0x1: HSYNC gated enabled	RW	0
5	PIXELCLOCK GATED	Pixel Clock Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: Pixel clock gated disabled 0x1: Pixel clock gated enabled	RW	0
4	PIXELDATAGATED	Pixel data gated enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: Pixel data gated disabled 0x1: Pixel data gated enabled	RW	0
3	PALETTEGAMMA TABLE	Palette/gamma table selection wr: VFP start period of primary LCD output or VFP start period of secondary LCD output or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the graphics pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory. In case of the table is used as gamma table, it is used for the primary LCD output only. **NOTE: CLUT and BITMAP formats are not supported in this family of devices.** 0x0: LUT used as palette (only if graphics format is BITMAP1, 2, 4, and 8) 0x1: LUT used as gamma table (only if graphics format is NOT BITMAP1, 2, 4, and 8 or no graphics window present)	RW	0

Bits	Field Name	Description	Type	Reset
2:1	LOADMODE	Loading mode for the palette/gamma table wr: VFP start period of primary LCD output or VFP start period of secondary LCD output or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory 0x0: Palette/Gamma Table and data are loaded every frame 0x1: Palette/Gamma Table to be loaded. The user sets the bit when the palette/gamma table has to be loaded. Hardware resets the bit to 0x2 when table has been loaded. (DISPC_GFX_ATTRIBUTES.ENABLE has to be set to 1). 0x2: Frame data only loaded every frame 0x3: Palette/Gamma Table and frame data loaded on first frame then switch to 0x2 (Hardware).	RW	0x0
0	PIXELGATED	Pixel gated enable (only for TFT) (primary LCD output) wr: VFP start period of primary LCD output 0x0: Pixel clock always toggles (only in TFT mode) 0x1: Pixel clock only toggles when there is valid data to display. (only in TFT mode)	RW	0

Table 11-176. DISPC_DEFAULT_COLOR0

Address Offset	0x0000 004C		
Physical Address	0x5800 104C	**Instance**	DISPC
Description	The control register allows to configure the default solid background color for the primary LCD. Shadow register, updated on VFP start period of the primary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEFAULTCOLOR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	DEFAULTCOLOR	24-bit RGB color value to specify the default solid color to display when there is no data from the overlays.	RW	0x000000

Table 11-177. DISPC_DEFAULT_COLOR1

Address Offset	0x0000 0050		
Physical Address	0x5800 1050	**Instance**	DISPC
Description	The control register allows to configure the default solid background color for the TV output. Shadow register, updated on EVSYNC		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEFAULTCOLOR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	DEFAULTCOLOR	24-bit RGB color value to specify the default solid color to display when there is no data from the overlays.	RW	0x000000

Table 11-178. DISPC_TRANS_COLOR0

Address Offset	0x0000 0054		
Physical Address	0x5800 1054	**Instance**	DISPC
Description	The register sets the transparency color value for the video/graphics overlays for the primary LCD output. Shadow register, updated on VFP start period of the primary LCD		
Type	RW		

31–24	23–0
RESERVED	TRANSCOLORKEY

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	TRANSCOLORKEY	Transparency color key value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24 **NOTE: CLUT and BITMAP formats are not supported in this family of devices.**	RW	0x000000

Table 11-179. DISPC_TRANS_COLOR1

Address Offset	0x0000 0058		
Physical Address	0x5800 1058	**Instance**	DISPC
Description	The register sets the transparency color value for the video/graphics overlays for the TV output. Shadow register, updated on EVSYNC		
Type	RW		

31–24	23–0
RESERVED	TRANSCOLORKEY

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	TRANSCOLORKEY	Transparency color key value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24 **NOTE: CLUT and BITMAP formats are not supported in this family of devices.**	RW	0x000000

Table 11-180. DISPC_LINE_STATUS

Address Offset	0x0000 005C		
Physical Address	0x5800 105C	**Instance**	DISPC
Description	The control register indicates the current primary LCD panel display line number.		
Type	R		

31–12	11–0
RESERVED	LINENUMBER

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	LINENUMBER	Current LCD panel line number Current display line number. The first active line has the value 0. During blanking lines the line number is not incremented.	R	0x000

Table 11-181. DISPC_LINE_NUMBER

Address Offset	0x0000 0060		
Physical Address	0x5800 1060	**Instance**	DISPC
Description	The control register indicates the primary LCD panel display line number for the interrupt and the DMA request. Shadow register, updated on VFP start period of primary LCD.		
Type	RW		

31–12	11–0
RESERVED	LINENUMBER

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	LINENUMBER	LCD panel line number programming LCD line number defines the line on which the programmable interrupt is generated and the DMA request occurs.	RW	0x000

Table 11-182. DISPC_TIMING_H1

Address Offset	0x0000 0064		
Physical Address	0x5800 1064	**Instance**	DISPC
Description	The register configures the timing logic for the HSYNC signal. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31–20	19–8	7–0
HBP	HFP	HSW

Bits	Field Name	Description	Type	Reset
31:20	HBP	Horizontal Back Porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (program to value minus 1). When in BT mode and interlaced, this field corresponds to the vertical field blanking No 2 for Even Field.	RW	0x000
19:8	HFP	Horizontal front porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted (program to value minus 1). When in BT mode and interlaced, this field corresponds to the vertical field blanking No 1 for Even Field.	RW	0x000
7:0	HSW	Horizontal synchronization pulse width. Encoded value (from 1 to 256) to specify the number of pixel clock periods to pulse the line clock at the end of each line (program to value minus 1). When in BT mode, this field corresponds to the horizontal blanking	RW	0x00

Table 11-183. DISPC_TIMING_V1

Address Offset	0x0000 0068		
Physical Address	0x5800 1068	**Instance**	DISPC
Description	The register configures the timing logic for the VSYNC signal. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31–20	19–8	7–0
VBP	VFP	VSW

Bits	Field Name	Description	Type	Reset
31:20	VBP	Vertical back porch. Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the beginning of a frame.	RW	0x000
19:8	VFP	Vertical front porch. Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the end of each frame.	RW	0x000
7:0	VSW	Vertical synchronization pulse width. In active mode, encoded value (from 1 to 256) to specify the number of line clock periods (program to value minus 1) to pulse the frame clock (VSYNC) pin at the end of each frame after the end of frame wait (VFP) period elapses. Frame clock uses as VSYNC signal in active mode.	RW	0x00

Table 11-184. DISPC_POL_FREQ1

Address Offset	0x0000 006C		
Physical Address	0x5800 106C	**Instance**	DISPC
Description	The register configures the signal configuration. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="13">RESERVED</td><td>ALIGN</td><td>ONOFF</td><td>RF</td><td>IEO</td><td>IPC</td><td>IHS</td><td>IVS</td><td colspan="4">ACBI</td><td colspan="8">ACB</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
18	ALIGN	Defines the alignment between HSYNC and VSYNC assertion. 0x0: VSYNC and HSYNC are not aligned 0x1: VSYNC and HSYNC assertions are aligned.	RW	0
17	ONOFF	HSYNC/VSYNC Pixel clock Control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit 16 **Note:** Control module register CTRL_CORE_SMA_SW_1[22]DSS_CH0_ON_OFF must be set to match	RW	0
16	RF	Program HSYNC/VSYNC Rise or Fall 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit 17 set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit 17 set to 1) **Note:** Control module register CTRL_CORE_SMA_SW_1[16]DSS_CH0_RF must be set to match	RW	0
15	IEO	Invert output enable 0x0: Ac-bias is active high (active display mode) 0x1: Ac-bias is active low (active display mode)	RW	0

Bits	Field Name	Description	Type	Reset
14	IPC	Invert pixel clock 0x0: Data is driven on the LCD data lines on the rising-edge of the pixel clock 0x1: Data is driven on the LCD data lines on the falling-edge of the pixel clock **Note:** Control module register CTRL_CORE_SMA_SW_1[19]DSS_CH0_IPC must be set to match	RW	0
13	IHS	Invert HSYNC 0x0: Line clock pin is active high and inactive low 0x1: Line clock pin is active low and inactive high	RW	0
12	IVS	Invert VSYNC 0x0: Frame clock pin is active high and inactive low 0x1: Frame clock pin is active low and inactive high	RW	0
11:8	ACBI	AC Bias pin transitions per interrupt Value (from 0 to 15) used to specify the number of AC Bias pin transitions	RW	0x0
7:0	ACB	AC Bias pin frequency value (from 0 to 255) used to specify the number of line clocks to count before transitioning the AC Bias pin. This pin is used to periodically invert the polarity of the power supply to prevent DC charge build-up within the display.	RW	0x00

Table 11-185. DISPC_DIVISOR1

Address Offset	0x0000 0070		
Physical Address	0x5800 1070	**Instance**	DISPC
Description	The register configures the divisors. It is used for the primary LCD output Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LCD								RESERVED								PCD							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:16	LCD	Display controller logic clock divisor value (from 1 to 255) to specify the intermediate pixel clock frequency based on the LCD1_CLK. The value 0 is invalid.	RW	0x04
15:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
7:0	PCD	Pixel clock divisor value (from 1 to 255) to specify the frequency of the pixel clock based on the LCD1_CLK divided by DISPC_DIVISOR1.LCD value. The values 0 is invalid.	RW	0x01

Table 11-186. DISPC_GLOBAL_ALPHA

Address Offset	0x0000 0074		
Physical Address	0x5800 1074	**Instance**	DISPC
Description	The register defines the global alpha value for the graphics and three video pipelines. Shadow register, updated on VFP start period of primary LCD or VFP start period of the third LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory for each bit field depending on the association of the each pipeline with the primary LCD, secondary LCD or TV output.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VID3GLOBALALPHA								VID2GLOBALALPHA								VID1GLOBALALPHA								GFXGLOBALALPHA							

Bits	Field Name	Description	Type	Reset
31:24	VID3GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF
23:16	VID2GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF
15:8	VID1GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF
7:0	GFXGLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF

Table 11-187. DISPC_SIZE_TV

Address Offset	0x0000 0078		
Physical Address	0x5800 1078	**Instance**	DISPC
Description	The register configures the size of the TV output field (interlace), frame (progressive) (horizontal and vertical). Shadow register, updated on EVSYNC. A delta value is used to indicate if the odd field has same vertical size as the even field or +/- one line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				LPP												DELTA_LPP		RESERVED		PPL											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
27:16	LPP	Lines per panel encoded value (from 1 to 4096) to specify the number of lines per panel.	RW	0x000
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: Same size 0x1: Odd size = Even size +1 0x2: Odd size = Even Size –1	RW	0x0
13:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:0	PPL	Pixels per line encoded value (from 1 to 4096) to specify the number of pixels contains within each line on the display.	RW	0x000

Table 11-188. DISPC_SIZE_LCD1

Address Offset	0x0000 007C		
Physical Address	0x5800 107C	**Instance**	DISPC
Description	The register configures the panel size (horizontal and vertical). Shadow register, updated on VFP start period of primary LCD. A delta value is used to indicate if the odd field has same vertical size as the even field or +/- one line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				LPP												DELTA_LPP		RESERVED		PPL											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
27:16	LPP	Lines per panel encoded value (from 1 to 4096) to specify the number of lines per panel (program to value minus 1).	RW	0x000
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: Same size 0x1: Odd size = Even size +1 0x2: Odd size = Even Size -1	RW	0x0
13:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:0	PPL	Pixels per line encoded value (from 1 to 4096) to specify the number of pixels contains within each line on the display (program to value minus 1). In STALL mode, any value is valid. In non STALL mode, only values multiple of 8 pixels are valid.	RW	0x000

Table 11-189. DISPC_GFX_BA_j

Address Offset	0x0000 0080 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1080 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the graphics buffer displayed in the graphics window (0 and 1 :for ping-pong mechanism with external trigger, based on the field polarity, 0 only used when graphics pipeline on the LCD output and 0 and 1 when on the TV output). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Graphics base address Base address of the graphics buffer (aligned on pixel size boundary) (in case 1-, 2-, and 4-bpp, byte alignment is required, in case of RGB24 packed format, 4-pixel alignment is required) When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-190. DISPC_GFX_POSITION

Address Offset	0x0000 0088		
Physical Address	0x5800 1088	**Instance**	DISPC
Description	The register configures the position of the graphics window. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	POSY	RESERVED	POSX

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	POSY	Y position of the graphics window. Encoded value (from 0 to 2047) to specify the Y position of the graphics window on the screen. The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	POSX	X position of the graphics window. Encoded value (from 0 to 2047) to specify the X position of the graphics window on the screen. The first pixel on the left of the screen has the X-position 0.	RW	0x000

Table 11-191. DISPC_GFX_SIZE

Address Offset	0x0000 008C		
Physical Address	0x5800 108C	**Instance**	DISPC
Description	The register configures the size of the graphics window. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SIZEY												RESERVED					SIZEX										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
27:16	SIZEY	Number of lines of the graphics window. Encoded value (from 1 to 4096) to specify the number of lines of the graphics window (program to value minus 1).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZEX	Number of pixels of the graphics window. Encoded value (from 1 to 2048) to specify the number of pixels per line of the graphics window (program to value minus 1).	RW	0x000

Table 11-192. DISPC_GFX_ATTRIBUTES

Address Offset	0x0000 00A0		
Physical Address	0x5800 10A0	**Instance**	DISPC
Description	The register configures the graphics attributes. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CHANNELOUT2	BURSTTYPE	PREMULTIPLYALPHA	ZORDER	ZORDERENABLE	ANTIFLICKER	RESERVED	SUBSAMPLINGPATTERN	SELFREFRESHAUTO	FORCE1DTILEDMODE	SELFREFRESH	ARBITRATION	ROTATION	BUFPRELOAD	FRAMEPACKINGMODE	NIBBLEMODE	CHANNELOUT	BURSTSIZE	REPLICATIONENABLE	FORMAT	ENABLE

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (should set to zero) wr: immediate 0x0: Primary LCD output selected. 0x1: Secondary LCD output selected. 0x2: Third LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. (It does not apply to the palette loading OCP requests using INCR burst only) 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPLYALPHA	The field configures the DISPC GFX to process incoming data as premultiplied alpha data or non premultiplied alpha data. Default setting is non premultiplied alpha data. 0x0: Non premultiplyalpha data color component 0x1: Premultiplyalpha data color component	RW	0
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0. 0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is only used when the Z-order is enabled. 0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	ANTIFLICKER	Antiflicker filtering using a 3-tap filter with hardcoded coefficients (1/4, 1/2, 1/4) 0x0: Antiflicker disabled. 0x1: Antiflicker enabled.	RW	0
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0

Bits	Field Name	Description	Type	Reset
20:18	SUBSAMPLINGPATTERN	Subsampling pattern setting.	RW	0x0
17	SELFREFRESHAUTO	Automatic self-refresh mode 0x0: The transition from Selfrefresh disabled to enabled is controlled by software 0x1: The transition from Selfrefresh disabled to enabled is controlled only by hardware	RW	0
16	FORCE1DTILEDMODE	Force TILED regions access to 1D or 2D. 0x0: 2D accesses for tiled regions 0x1: 1D accesses for tiled regions	RW	0x0
15	SELFREFRESH	Enables the self refresh of the graphics window from its own DMA buffer. This bit should be set only after having set the GO bit of the channel and read back a zero in its field. 0x0: The graphics pipeline accesses the interconnect to fetch data from the system memory. 0x1: The graphics pipeline does not need anymore to fetch data from memory. Only the graphics DMA buffer is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0
14	ARBITRATION	Determines the priority of the graphics pipeline. When the graphics pipeline is one of the high priority pipelines. The arbitration wheel gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The graphics pipeline is one of the normal priority pipeline. 0x1: The graphics pipeline is one of the high priority pipeline.	RW	0
13:12	ROTATION	Graphics rotation flag 0x0: No rotation 0x1: Rotation by 90 degrees 0x3: Rotation by 270 degrees 0x2: Rotation by 180 degrees	RW	0x0
11	BUFPRELOAD	Graphics preload value 0x0: Hardware prefetches pixels up to the preload value defined in the preload register 0x1: Hardware prefetches pixels up to high threshold value	RW	0
10	FRAMEPACKINGMODE	Frame packing mode control. 0x0: Frame Packing mode is disabled 0x1: Frame Packing mode is enabled	RW	0x0
9	NIBBLEMODE	Graphics nibble mode (only for 1-, 2- and 4 bpp) **NOTE: BITMAP formats and associated Nibble Mode are not supported in this family of devices.** 0x0: Nibble mode is disabled 0x1: Nibble mode is enabled	RW	0
8	CHANNELOUT	Graphics Channel Out configuration: LCD, WB or TV. wr: immediate 0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back). 0x1: TV output selected	RW	0

Bits	Field Name	Description	Type	Reset
7:6	BURSTSIZE	Graphics DMA burst size 0x0: 2 × 128-bit bursts 0x1: 4 × 128-bit bursts 0x3: Reserved 0x2: 8 × 128-bit bursts	RW	0x2
5	REPLICATIONENABLE	Graphics replication enabled: RGB . ARGB, and RGBA formats are converted into ARGB32-8888 using replication of the MSBs or 0s 0x0: Disable graphics replication logic. The conversion to ARGB32-8888 is done by adding 0s for the LSBs 0x1: Enable graphics replication logic. The conversion to ARGB32-8888 is done by duplicating the MSBs for the LSBs	RW	1
4:1	FORMAT	Graphics format. It defines the pixel format when fetching the graphics picture into memory. 0x6: RGB16-565 0xA: RGBx12-4444 0x7: ARGB16-1555 0xD: RGBA32-8888 0x8: xRGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: RGBA12-4444 0x4: xRGB12-4444 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x3: BGRA32-8888 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	Graphics enable 0x0: Graphics disabled (graphics pipeline inactive and graphics window not present) 0x1: Graphics enabled (graphics pipeline active and graphics window present on the screen)	RW	0

Table 11-193. DISPC_GFX_BUF_THRESHOLD

Address Offset	0x0000 00A4		
Physical Address	0x5800 10A4	**Instance**	DISPC
Description	The register configures the graphics buffer. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer high threshold number of 128 bits defining the threshold value	RW	0x04FF

Bits	Field Name	Description	Type	Reset
15:0	BUFLOWTHRESHOLD	DMA buffer low threshold number of 128 bits defining the threshold value. The value put in this register must always be greater than zero.	RW	0x04F8

Table 11-194. DISPC_GFX_BUF_SIZE_STATUS

Address Offset	0x0000 00A8		
Physical Address	0x5800 10A8	**Instance**	DISPC
Description	The register defines the Graphics buffer size		
Type	R		

31:16	15:0
RESERVED	BUFSIZE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer size in number of 128 bits	R	0x0500

Table 11-195. DISPC_GFX_ROW_INC

Address Offset	0x0000 00AC		
Physical Address	0x5800 10AC	**Instance**	DISPC
Description	The register configures the number of bytes to increment at the end of the row. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:0
ROWINC

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded unsigned value to specify the number of bytes to increment at the end of the row in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels. The value 1-(n+1)*bpp means decrement of n pixels.	RW	0x0000 0001

Table 11-196. DISPC_GFX_PIXEL_INC

Address Offset	0x0000 00B0		
Physical Address	0x5800 10B0	**Instance**	DISPC
Description	The register configures the number of bytes to increment between two pixels. For more information, see *Predecimation*. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:8	7:0
RESERVED	PIXELINC

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels.	RW	0x01

Table 11-197. DISPC_GFX_TABLE_BA

Address Offset	0x0000 00B8		
Physical Address	0x5800 10B8	**Instance**	DISPC
Description	The register configures the base address of the palette buffer or the gamma table buffer. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory. **NOTE: CLUT and BITMAP formats, and associated palette buffer, are not supported in this family of devices.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABLEBA																															

Bits	Field Name	Description	Type	Reset
31:0	TABLEBA	Base address of the palette/gamma table buffer (24-bit entries in 32-bit containers, aligned on 32-bit boundary). **NOTE: CLUT and BITMAP formats, and associated palette buffer, are not supported in this family of devices.**	RW	0x0000 0000

Table 11-198. DISPC_VID1_BA_j

Address Offset	0x0000 00BC + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 10BC + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the video buffer for the video window 1 (DISPC_VID1_BA_0 and DISPC_VID1_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID1_BA_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address Base address of the video buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2:0, byte alignment is supported)). It case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-199. DISPC_VID1_POSITION

Address Offset	0x0000 00C4		
Physical Address	0x5800 10C4	**Instance**	DISPC
Description	The register configures the position of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					POSY											RESERVED					POSX										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	POSY	Y position of the video window 1 Encoded value (from 0 to 2047) to specify the Y position of the video window 1 .The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	POSX	X position of the video window 1 Encoded value (from 0 to 2047) to specify the X position of the video window 1. The first pixel on the left of the display screen has the X-position 0.	RW	0x000

Table 11-200. DISPC_VID1_SIZE

Address Offset	0x0000 00C8		
Physical Address	0x5800 10C8	**Instance**	DISPC
Description	The register configures the size of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD, orEVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SIZEY												RESERVED					SIZEX										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
27:16	SIZEY	Number of lines of the video 1 Encoded value (from 1 to 4096) to specify the number of lines of the video window 1. Program to value minus 1.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZEX	Number of pixels of the video window 1 Encoded value (from 1 to 2048) to specify the number of pixels of the video window 1. Program to value minus 1.	RW	0x000

Table 11-201. DISPC_VID1_ATTRIBUTES

Address Offset	0x0000 00CC		
Physical Address	0x5800 10CC	**Instance**	DISPC
Description	The register configures the attributes of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:30	29	28	27:26	25	24	23	22	21	20	19	18	17	16	15:14	13:12	11	10	9	8	7	6:5	4:1	0
CHANNELOUT2	BURSTTYPE	PREMULTIPHYALPHA	ZORDER	ZORDERENABLE	SELFREFRESH	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	FORCE1DTILEDMODE	BUFPRELOAD	RESERVED	SELFREFRESHAUTO	CHANNELOUT	BURSTSIZE	ROTATION	FULLRANGE	REPLICATIONENABLE	COLORCONVENABLE	FRAMEPACKINGMODE	HRESIZECONF	RESIZEENABLE	FORMAT	ENABLE

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (should be set to zero) wr: immediate 0x0: Primary LCD output selected. 0x1: Secondary LCD output selected. 0x2: Third LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPHYALPHA	The field configures the DISPC VID1 to process incoming data as premultiplied alpha data or non premultiplied alpha data. Default setting is non premultiplied alpha data. 0x0: Non premultiplyalpha data color component 0x1: Premultiplyalpha data color component	RW	0

Bits	Field Name	Description	Type	Reset
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0. 0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is only used when the Z-order is enabled. 0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	SELFREFRESH	Enables the self refresh of the video window from its own DMA buffer only. 0x0: The video pipeline accesses the interconnect to fetch data from the system memory. 0x1: The video pipeline does not need anymore to fetch data from memory. Only the DMA buffer associated with the video1 is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0
23	ARBITRATION	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The video pipeline is one of the normal priority pipeline. 0x1: The video pipeline is one of the high priority pipeline.	RW	0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV4:2:0. 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0
21	VERTICALTAPS	Video vertical resize tap number. The vertical polyphase filter can be configured in 3-tap or 5-tap configuration. According to the number of taps, the maximum input picture width is double while using 3-tap compared to 5-tap. 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. The associated bit fields for the 2 other taps coefficients do not need to be initialized. 0x1: 5 taps are used for the vertical filtering logic.	RW	0
20	FORCE1DTILEDMODE	Force TILED regions access to 1D or 2D. 0x0: 2D accesses for tiled regions 0x1: 1D accesses for tiled regions	RW	0

Bits	Field Name	Description	Type	Reset
19	BUFPRELOAD	Video Preload Value 0x0: Hardware prefetches pixels up to the preload value defined in the preload register 0x1: Hardware prefetches pixels up to high threshold value	RW	0
18	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
17	SELFREFRESHAUTO	Automatic self-refresh mode 0x0: The transition from SELFREFRESH disabled to enabled is controlled by SW. 0x1: The transition from SELFREFRESH disabled to enabled is controlled only by hardware.	RW	0
16	CHANNELOUT	Video channel out configuration: LCD, WB or TV. wr: immediate 0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back). 0x1: TV output selected	RW	0
15:14	BURSTSIZE	Video DMA burst size 0x0: 2x128bit bursts 0x1: 4x128bit bursts 0x3: Reserved 0x2: 8x128bit bursts	RW	0x2
13:12	ROTATION	Video rotation flag 0x0: No rotation 0x1: Rotation by 90 degrees 0x3: Rotation by 270 degrees 0x2: Rotation by 180 degrees	RW	0x0
11	FULLRANGE	Color space conversion full range setting. 0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	REPLICATIONENABLE	Replication enable 0x0: Disable Video replication logic 0x1: Enable Video replication logic	RW	1
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion YUV to RGB 0x1: Enable Color Space Conversion YUV to RGB	RW	0
8	FRAMEPACKINGMODE	Frame packing mode control. 0x0: Frame Packing mode is disabled 0x1: Frame Packing mode is enabled	RW	0
7	HRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0

Bits	Field Name	Description	Type	Reset
6:5	RESIZEENABLE	Video Resize Enable 0x0: Disable both horizontal and vertical resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0
4:1	FORMAT	Video Format. It defines the pixel format when fetching the video 1 picture into memory. 0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: xRGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0x3: BGRA32-8888 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	Video Enable 0x0: Video disabled (video pipeline inactive and window not present) 0x1: Video enabled (video pipeline active and window present on the screen)	RW	0

Table 11-202. DISPC_VID1_BUF_THRESHOLD

Address Offset	0x0000 00D0		
Physical Address	0x5800 10D0	**Instance**	DISPC
Description	The register configures the video buffer associated with the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	Video DMA buffer high threshold number of 128 bits defining the threshold value	RW	0x07FF

Bits	Field Name	Description	Type	Reset
15:0	BUFLOWTHRESHOLD	DMA buffer low threshold number of 128 bits defining the threshold value	RW	0x07F8

Table 11-203. DISPC_VID1_BUF_SIZE_STATUS

Address Offset	0x0000 00D4		
Physical Address	0x5800 10D4	**Instance**	DISPC
Description	The register defines the Video buffer size for the video pipeline 1.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	Video 1 DMA buffer size in number of 128-bits	R	0x0800

Table 11-204. DISPC_VID1_ROW_INC

Address Offset	0x0000 00D8		
Physical Address	0x5800 10D8	**Instance**	DISPC
Description	The register configures the number of bytes to increment at the end of the row for the buffer associated with the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from $2^{31}1$ to 2^{31}) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1 + n * bpp means increment of n pixels. The value 1 (n + 1)* bpp means decrement of n pixels.	RW	0x0000 0001

Table 11-205. DISPC_VID1_PIXEL_INC

Address Offset	0x0000 00DC		
Physical Address	0x5800 10DC	**Instance**	DISPC
Description	The register configures the number of bytes to increment between two pixels for the buffer associated with the video window 2. For more information, see *Predecimation*.The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD orEVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PIXELINC							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1 + n * bpp means increment of n pixels. For YUV4:2:0, maximum supported value is 128.	RW	0x01

Table 11-206. DISPC_VID1_FIR

Address Offset	0x0000 00E0		
Physical Address	0x5800 10E0	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 1. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:29	28:16	15:13	12:0
RESERVED	FIRVINC	RESERVED	FIRHINC

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-207. DISPC_VID1_PICTURE_SIZE

Address Offset	0x0000 00E4		
Physical Address	0x5800 10E4	**Instance**	DISPC
Description	The register configures the size of the video picture associated with the video layer 1 before up/down-scaling. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD, EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:28	27:16	15:11	10:0
RESERVED	MEMSIZEY	RESERVED	MEMSIZEX

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:16	MEMSIZEY	Number of lines of the video picture. Encoded value (from 1 to 4096) to specify the number of lines of the video picture in memory (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded to 2^{11}.	RW	0x000

Bits	Field Name	Description	Type	Reset
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	MEMSIZEX	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus 1). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit. (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded to 2^{11}.	RW	0x000

Table 11-208. DISPC_VID1_ACCU_j

Address Offset	0x0000 00E8 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 10E8 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 1 (DISPC_VID1_ACCU_0 and DISPC_VID1_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					VERTICALACCU											RESERVED					HORIZONTALACCU										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accumulator value encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accumulator value encoded value (from –1024 to 1023).	RW	0x000

Table 11-209. DISPC_VID1_FIR_COEF_H_i

Address Offset	0x0000 00F0 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 10F0 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VPF start pertiod of the third LCD, EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00

Bits	Field Name	Description	Type	Reset
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-210. DISPC_VID1_FIR_COEF_HV_i

Address Offset	0x0000 00F4 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 10F4 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:24	23:16	15:8	7:0
FIRVC2	FIRVC1	FIRVC0	FIRHC4

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-211. DISPC_VID1_CONV_COEF0

Address Offset	0x0000 0130		
Physical Address	0x5800 1130	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	RCR	RESERVED	RY

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	RCR	RCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RY	RY coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-212. DISPC_VID1_CONV_COEF1

Address Offset	0x0000 0134		
Physical Address	0x5800 1134	**Instance**	DISPC

Table 11-212. DISPC_VID1_CONV_COEF1 (continued)

Description	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory
Type	RW

31:27	26:16	15:11	10:0
RESERVED	GY	RESERVED	RCB

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GY	GY coefficient encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RCB	RCb coefficient encoded signed value (from -1024 to 1023).	RW	0x000

Table 11-213. DISPC_VID1_CONV_COEF2

Address Offset	0x0000 0138		
Physical Address	0x5800 1138	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	GCB	RESERVED	GCR

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GCB	GCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	GCR	GCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-214. DISPC_VID1_CONV_COEF3

Address Offset	0x0000 013C		
Physical Address	0x5800 113C	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	BCR	RESERVED	BY

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	BCR	BCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	BY	BY coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-215. DISPC_VID1_CONV_COEF4

Address Offset	0x0000 0140		
Physical Address	0x5800 1140	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:11	10:0
RESERVED	BCB

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	BCB	BCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-216. DISPC_VID2_BA_j

Address Offset	0x0000 014C + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 114C + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the video buffer for the video window 2 (DISPC_VID2_BA_0 and DISPC_VID2_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID2_BA_0 is used)). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:0
BA

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address Base address of the video buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2:0, byte alignment is supported)). In case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-217. DISPC_VID2_POSITION

Address Offset	0x0000 0154		
Physical Address	0x5800 1154	**Instance**	DISPC
Description	The register configures the position of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					POSY											RESERVED					POSX										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	POSY	Y position of the video window 2 encoded value (from 0 to 2047) to specify the Y position of the video window 2 .The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	POSX	X position of the video window 2 encoded value (from 0 to 2047) to specify the X position of the video window 2. The first pixel on the left of the display screen has the X-position 0.	RW	0x000

Table 11-218. DISPC_VID2_SIZE

Address Offset	0x0000 0158		
Physical Address	0x5800 1158	**Instance**	DISPC
Description	The register configures the size of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SIZEY												RESERVED					SIZEX										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
27:16	SIZEY	Number of lines of the video 2 encoded value (from 1 to 4096) to specify the number of lines of the video window 2. Program to value minus 1.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZEX	Number of pixels of the video window 2 encoded value (from 1 to 2048) to specify the number of pixels of the video window 2. Program to value minus 1.	RW	0x000

Table 11-219. DISPC_VID2_ATTRIBUTES

Address Offset	0x0000 015C		
Physical Address	0x5800 115C	**Instance**	DISPC
Description	The register configures the attributes of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANNELOUT2		BURSTTYPE	PREMULTIPLYALPHA	ZORDER		ZORDERENABLE	SELFREFRESH	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	FORCE1DTILEDMODE	BUFPRELOAD	RESERVED	SELFREFRESHAUTO	CHANNELOUT	BURSTSIZE		ROTATION		FULLRANGE	REPLICATIONENABLE	COLORCONVENABLE	FRAMEPACKINGMODE	HRESIZECONF	RESIZEENABLE		FORMAT				ENABLE

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (must be set to zero) wr: immediate 0x0: Primary LCD output selected. 0x1: Secondary LCD output selected. 0x2: Third LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPLYALPHA	The field configures the DISPC VID2 to process incoming data as premultiplied alpha data or non premultiplied alpha data. Default setting is non premultiplied alpha data. 0x0: Non premultiplyalpha data color component 0x1: Premultiplyalpha data color component	RW	0

Bits	Field Name	Description	Type	Reset
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0. 0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is only used when the Z-order is enabled. 0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	SELFREFRESH	Enables the self refresh of the video window from its own DMA buffer only. 0x0: The video pipeline accesses the interconnect to fetch data from the system memory. 0x1: The video pipeline does not need anymore to fetch data from memory. Only the DMA buffer associated with the video2 is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0
23	ARBITRATION	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The video pipeline is one of the normal priority pipeline. 0x1: The video pipeline is one of the high priority pipeline.	RW	0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV4:2:0. 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0
21	VERTICALTAPS	Video Vertical Resize Tap Number 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. The associated bit fields for the 2 other taps coefficients do not need to be initialized. 0x1: 5 taps are used for the vertical filtering logic.	RW	0
20	FORCE1DTILEDMODE	Force TILED regions access to 1D or 2D. 0x0: 2D accesses for tiled regions 0x1: 1D accesses for tiled regions	RW	0
19	BUFPRELOAD	Video Preload Value 0x0: Hardware prefetches pixels up to the preload value defined in the preload register 0x1: Hardware prefetches pixels up to high threshold value	RW	0

Bits	Field Name	Description	Type	Reset
18	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
17	SELFREFRESHAUTO	Automatic self-refresh mode 0x0: The transition from SELFREFRESH disabled to enabled is controlled by SW. 0x1: The transition from SELFREFRESH disabled to enabled is controlled only by hardware.	RW	0
16	CHANNELOUT	Video Channel Out configuration: LCD, WB or TV. wr: immediate 0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back). 0x1: TV output selected	RW	0
15:14	BURSTSIZE	Video DMA burst size 0x0: 2 × 128-bit bursts 0x1: 4 × 128-bit bursts 0x3: Reserved 0x2: 8 × 128-bit bursts	RW	0x2
13:12	ROTATION	Video Rotation Flag 0x0: No rotation 0x1: Rotation by 90 degrees 0x3: Rotation by 270 degrees 0x2: Rotation by 180 degrees	RW	0x0
11	FULLRANGE	Color space conversion full range setting. 0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	REPLICATIONENABLE	Replication Enable 0x0: Disable Video replication logic 0x1: Enable Video replication logic	RW	1
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV. 0x0: Disable color space conversion YUV to RGB 0x1: Enable color space conversion YUV to RGB	RW	0
8	FRAMEPACKINGMODE	Frame packing mode control. 0x0: Frame Packing mode is disabled 0x1: Frame Packing mode is enabled	RW	0
7	HRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0
6:5	RESIZEENABLE	Video Resize Enable 0x0: Disable both horizontal and vertical resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0

Bits	Field Name	Description	Type	Reset
4:1	FORMAT	Video Format. It defines the pixel format when fetching the video 2 picture into memory. 0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: xRGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0x3: BGRA32-8888 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	VidEnable 0x0: Video disabled (video pipeline inactive and window not present) 0x1: Video enabled (video pipeline active and window present on the screen)	RW	0

Table 11-220. DISPC_VID2_BUF_THRESHOLD

Address Offset	0x0000 0160		
Physical Address	0x5800 1160	**Instance**	DISPC
Description	The register configures the DMA buffer associated with the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:16	15:0
BUFHIGHTHRESHOLD	BUFLOWTHRESHOLD

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer high threshold number of 128 bits defining the threshold value	RW	0x07FF
15:0	BUFLOWTHRESHOLD	DMA buffer low threshold number of 128 bits defining the threshold value	RW	0x07F8

Table 11-221. DISPC_VID2_BUF_SIZE_STATUS

Address Offset	0x0000 0164		
Physical Address	0x5800 1164	**Instance**	DISPC
Description	The register defines the DMA buffer size for the video pipeline 2.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer size in number of 128 bits	R	0x0800

Table 11-222. DISPC_VID2_ROW_INC

Address Offset	0x0000 0168		
Physical Address	0x5800 1168	**Instance**	DISPC
Description	The register configures the number of bytes to increment at the end of the row for the buffer associated with the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from $2^{31}1$ to 2^{31}) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1 + n * bpp means increment of n pixels. The value 1 (n + 1) * bpp means decrement of n pixels.	RW	0x0000 0001

Table 11-223. DISPC_VID2_PIXEL_INC

Address Offset	0x0000 016C		
Physical Address	0x5800 116C	**Instance**	DISPC
Description	The register configures the number of bytes to increment between two pixels for the buffer associated with the video window 2. For more information, see *Predecimation*.The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PIXELINC							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between2 pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1 + n * bpp means increment of n pixels. For YUV4:2:0, maximum supported value is 128.	RW	0x01

Table 11-224. DISPC_VID2_FIR

Address Offset	0x0000 0170		
Physical Address	0x5800 1170	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 2. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:29	28:16	15:13	12:0
RESERVED	FIRVINC	RESERVED	FIRHINC

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-225. DISPC_VID2_PICTURE_SIZE

Address Offset	0x0000 0174		
Physical Address	0x5800 1174	**Instance**	DISPC
Description	The register configures the size of the video picture associated with the video layer 2 before up/down-scaling. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:28	27:16	15:11	10:0
RESERVED	MEMSIZEY	RESERVED	MEMSIZEX

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
27:16	MEMSIZEY	Number of lines of the video picture Encoded value (from 1 to 4096) to specify the number of lines of the video picture in memory (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the maximum size of the unpredecimated image size in memory is still bounded 2^{11}.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
10:0	MEMSIZEX	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus 1). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit. (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded 2^{11}.	RW	0x000

Table 11-226. DISPC_VID2_ACCU_j

Address Offset	0x0000 0178 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1178 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 2 (DISPC_VID2_ACCU_0 and DISPC_VID2_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	VERTICALACCU	RESERVED	HORIZONTALACCU

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accumulator value encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accumulator value encoded value (from –1024 to 1023).	RW	0x000

Table 11-227. DISPC_VID2_FIR_COEF_H_i

Address Offset	0x0000 0180 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1180 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:24	23:16	15:8	7:0
FIRHC3	FIRHC2	FIRHC1	FIRHC0

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00

Bits	Field Name	Description	Type	Reset
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-228. DISPC_VID2_FIR_COEF_HV_i

Address Offset	0x0000 0184 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1184 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–24	23–16	15–8	7–0
FIRVC2	FIRVC1	FIRVC0	FIRHC4

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-229. DISPC_VID2_CONV_COEF0

Address Offset	0x0000 01C0		
Physical Address	0x5800 11C0	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–27	26–16	15–11	10–0
RESERVED	RCR	RESERVED	RY

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	RCR	RCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RY	RY coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-230. DISPC_VID2_CONV_COEF1

Address Offset	0x0000 01C4		
Physical Address	0x5800 11C4	**Instance**	DISPC

Table 11-230. DISPC_VID2_CONV_COEF1 (continued)

Description	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory
Type	RW

31–27	26–16	15–11	10–0
RESERVED	GY	RESERVED	RCB

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GY	GY coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RCB	RCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-231. DISPC_VID2_CONV_COEF2

Address Offset	0x0000 01C8		
Physical Address	0x5800 11C8	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–27	26–16	15–11	10–0
RESERVED	GCB	RESERVED	GCR

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GCB	GCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	GCR	GCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-232. DISPC_VID2_CONV_COEF3

Address Offset	0x0000 01CC		
Physical Address	0x5800 11CC	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–27	26–16	15–11	10–0
RESERVED	BCR	RESERVED	BY

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	BCR	BCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	BY	BY coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-233. DISPC_VID2_CONV_COEF4

Address Offset	0x0000 01D0		
Physical Address	0x5800 11D0	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:11	10:0
RESERVED	BCB

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	BCB	BCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-234. DISPC_DATA1_CYCLE1

Address Offset	0x0000 01D4		
Physical Address	0x5800 11D4	**Instance**	DISPC
Description	The control register configures the output data format for 1st cycle. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31:28	27:24	23:21	20:16	15:12	11:8	7:5	4:0
RESERVED	BITALIGNMENTPIXEL2	RESERVED	NBBITSPIXEL2	RESERVED	BITALIGNMENTPIXEL1	RESERVED	NBBITSPIXEL1

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment. Alignment of the bits from pixel 2 on the output interface.	RW	0x0
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment. Alignment of the bits from pixel 1 on the output interface.	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-235. DISPC_DATA1_CYCLE2

Address Offset	0x0000 01D8

Table 11-235. DISPC_DATA1_CYCLE2 (continued)

Physical Address	0x5800 11D8	**Instance**	DISPC
Description	The control register configures the output data format for 2nd cycle. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BITALIGNMENTPIXEL2				RESERVED			NBBITSPIXEL2					RESERVED				BITALIGNMENTPIXEL1				RESERVED			NBBITSPIXEL1				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment. Alignment of the bits from pixel 2 on the output interface.	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment. Alignment of the bits from pixel 1 on the output interface.	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-236. DISPC_DATA1_CYCLE3

Address Offset	0x0000 01DC		
Physical Address	0x5800 11DC	**Instance**	DISPC
Description	The control register configures the output data format for 3rd cycle. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BITALIGNMENTPIXEL2				RESERVED			NBBITSPIXEL2					RESERVED				BITALIGNMENTPIXEL1				RESERVED			NBBITSPIXEL1				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment. Alignment of the bits from pixel 2 on the output interface.	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment. Alignment of the bits from pixel 1 on the output interface.	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-237. DISPC_VID1_FIR_COEF_V_i

Address Offset	0x0000 01E0 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 11E0 + (0x4 * i)	**Instance**	DISPC

Table 11-237. DISPC_VID1_FIR_COEF_V_i (continued)

Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory
Type	RW

31–16	15–8	7–0
RESERVED	FIRVC22	FIRVC00

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-238. DISPC_VID2_FIR_COEF_V_i

Address Offset	0x0000 0200 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1200 + (0x4 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–16	15–8	7–0
RESERVED	FIRVC22	FIRVC00

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-239. DISPC_CPR1_COEF_R

Address Offset	0x0000 0220		
Physical Address	0x5800 1220	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the Red component. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31–22	21	20–11	10	9–0
RR	RESERVED	RG	RESERVED	RB

Bits	Field Name	Description	Type	Reset
31:22	RR	RR coefficient encoded signed value (from –512 to 511)	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	RG	RG coefficient encoded signed value (from –512 to 511)	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	RB	RB coefficient encoded signed value (from –512 to 511)	RW	0x000

Table 11-240. DISPC_CPR1_COEF_G

Address Offset	0x0000 0224		
Physical Address	0x5800 1224	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the Green component. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31:22	21	20:11	10	9:0
GR	RESERVED	GG	RESERVED	GB

Bits	Field Name	Description	Type	Reset
31:22	GR	GR coefficient encoded signed value (from –512 to 511)	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	GG	GG coefficient encoded signed value (from –512 to 511)	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	GB	GB coefficient encoded signed value (from –512 to 511)	RW	0x000

Table 11-241. DISPC_CPR1_COEF_B

Address Offset	0x0000 0228		
Physical Address	0x5800 1228	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the Blue component. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
Type	RW		

31:22	21	20:11	10	9:0
BR	RESERVED	BG	RESERVED	BB

Bits	Field Name	Description	Type	Reset
31:22	BR	BR coefficient encoded signed value (from –512 to 511)	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	BG	BG coefficient encoded signed value (from –512 to 511	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	BB	BB coefficient encoded signed value (from –512 to 511)	RW	0x000

Table 11-242. DISPC_GFX_PRELOAD

Address Offset	0x0000 022C		
Physical Address	0x5800 122C	**Instance**	DISPC

Table 11-242. DISPC_GFX_PRELOAD (continued)

Description	The register configures the graphics DMA buffer Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory
Type	RW

31–12	11–0
RESERVED	PRELOAD

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value number of 128-bit words defining the preload value.	RW	0x100

Table 11-243. DISPC_VID1_PRELOAD

Address Offset	0x0000 0230		
Physical Address	0x5800 1230	**Instance**	DISPC
Description	The register configures the DMA buffer of the video 1 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–12	11–0
RESERVED	PRELOAD

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value number of 128-bit words defining the preload value.	RW	0x100

Table 11-244. DISPC_VID2_PRELOAD

Address Offset	0x0000 0234		
Physical Address	0x5800 1234	**Instance**	DISPC
Description	The register configures the DMA buffer of the video 2 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–12	11–0
RESERVED	PRELOAD

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

Table 11-245. DISPC_CONTROL2

Address Offset	0x0000 0238		
Physical Address	0x5800 1238	**Instance**	DISPC

Table 11-245. DISPC_CONTROL2 (continued)

Description	The control register configures the Display Controller module for the secondary LCD output. Shadow registers are updated during the VFP start period of the secondary LCD, EVSYNC, or when DISPC_CONTROL2.GOWB is set to 1 by software and the current WB frame is complete (that is, has no more data in the write-back pipeline).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPATIALTEMPORALDITHERINGFRAMES		RESERVED			TDMUNUSEDBITS		TDMCYCLEFORMAT		TDMPARALLELMODE		TDMENABLE	RESERVED						TVOVERLAYOPTIMIZATION	OVERLAYOPTIMIZATION	STALLMODE	RESERVED	TFTDATALINES		STDITHERENABLE	GOWB	GOLCD	M8B	STNTFT	MONOCOLOR	RESERVED	LCDENABLE

Bits	Field Name	Description	Type	Reset
31:30	SPATIALTEMPORAL DITHERINGFRAMES	Spatial/temporal dithering number of frames for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: Spatial only 0x1: Spatial and temporal over 2 frames 0x2: Spatial and temporal over 4 frames 0x3: Reserved	RW	0x0
29:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
26:25	TDMUNUSED BITS	State of unused bits (TDM mode only) for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: Low level (0) 0x1: High level (1) 0x2: Unchanged from previous state 0x3: Reserved	RW	0x0
24:23	TDMCYCLE FORMAT	Cycle format (TDM mode only) for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: 1 cycle for 1 pixel 0x1: 2 cycles for 1 pixel 0x2: 3 cycles for 1 pixel 0x3: 3 cycles for 2 pixels	RW	0x0
22:21	TDMPARALLEL MODE	Output Interface width (TDM mode only) for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: 8-bit parallel output interface selected 0x1: 9-bit parallel output interface selected 0x2: 12-bit parallel output interface selected 0x3: 16-bit parallel output interface selected	RW	0x0
20	TDMENABLE	Enable the multiple cycle format for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: TDM disabled 0x1: TDM enabled	RW	0
19:14	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
13	TVOVERLAY OPTIMIZATION	Overlay optimization for the TV output wr: VFP or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory 0x0: All the data for all the enabled pipelines are fetched from memory regardless of the overlay/alpha blending configuration. 0x1: The data not used by the overlay manager because of overlap between layers with no alpha blending between them shall not be fetched from memory in order to optimize the bandwidth.	RW	0
12	OVERLAY OPTIMIZATION	Overlay optimization for the secondary LCD output wr: VFP or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory 0x0: All the data for all the enabled pipelines are fetched from memory regardless of the overlay/alpha blending configuration. 0x1: The data not used by the overlay manager because of overlap between layers with no alpha blending between them shall not be fetched from memory in order to optimize the bandwidth.	RW	0
11	STALLMODE	STALL mode for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: Normal mode selected 0x1: STALL mode selected. The Display Controller sends the data without considering the VSYNC/HSYNC. The LCD output is disabled at the end of the transfer of the frame. The S/W has to re-enable the LCD output in order to generate a new frame.	RW	0
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:8	TFTDATALINES	Number of lines of the secondary LCD interface wr: VFP start period of secondary LCD output 0x0: 12-bit output aligned on the LSB of the pixel data interface 0x1: 16-bit output aligned on the LSB of the pixel data interface 0x2: 18-bit output aligned on the LSB of the pixel data interface 0x3: 24-bit output aligned on the LSB of the pixel data interface	RW	0x0
7	STDITHER ENABLE	Spatial temporal dithering enable for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: Spatial/Temporal dithering logic disabled 0x1: Spatial/Temporal dithering logic enabled	RW	0

Bits	Field Name	Description	Type	Reset
6	GOWB	GO command for the write-back output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the write-back output to the memory. wr:immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the write-back pipeline using the user values. The hardware resets the bit when the update is completed. 0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the write-back pipeline and the hardware can update the internal registers immediately	RW	0
5	GOLCD	GO command for the secondary LCD output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the secondary LCD output. wr:immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the LCD output using the user values. The hardware resets the bit when the update is completed. 0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the LCD output and the hardware can update the internal registers at the VFP start period	RW	0
4	M8B	Mono 8-bit mode of the secondary LCD wr: VFP start period of secondary LCD output 0x0: Reserved 0x1: Reserved	RW	0
3	STNTFT	LCD Display type of the secondary LCD wr: VFP start period of secondary LCD output 0x0: Reserved 0x1: Active or TFT display operation enabled. STN Dither logic and output FIFO bypassed.	RW	0
2	MONOCOLOR	Monochrome/Color selection for the secondary LCD wr: VFP start period of secondary LCD output 0x0: Reserved 0x1: Reserved	RW	0
1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
0	LCDENABLE	Enable the secondary LCD output wr:immediate 0x0: LCD output disabled (at the end of the frame when the bit is reset) 0x1: LCD output enabled	RW	0

Table 11-246. DISPC_GFX_POSITION2

Address Offset	0x0000 0240		
Physical Address	0x5800 1240	**Instance**	DISPC
Description	The register configures the position of the 2nd graphics window in FramePacking mode. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2[6] GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory.		
Type	RW		

31-27	26-16	15-11	10-0
RESERVED	POSY	RESERVED	POSX

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
26:16	POSY	Y position of the 2nd graphics window. Encoded value (from 0 to 2047) to specify the Y position of the graphics window on the screen. The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
10:0	POSX	X position of the 2nd graphics window. Encoded value (from 0 to 2047) to specify the X position of the graphics window on the screen. The first pixel on the left of the screen has the X-position 0.	RW	0x000

Table 11-247. DISPC_VID1_POSITION2

Address Offset	0x0000 0244		
Physical Address	0x5800 1244	**Instance**	DISPC
Description	The register configures the position of the 2nd video window #1 in FramePacking mode. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2[6] GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	POSY	RESERVED	POSX

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
26:16	POSY	Y position of the 2nd video window #1 Encoded value (from 0 to 2047) to specify the Y position of the video window #1 .The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
10:0	POSX	X position of the 2nd video window #1 Encoded value (from 0 to 2047) to specify the X position of the video window #1. The first pixel on the left of the display screen has the X-position 0.	RW	0x000

Table 11-248. DISPC_VID2_POSITION2

Address Offset	0x0000 0248		
Physical Address	0x5800 1248	**Instance**	DISPC
Description	The register configures the position of the 2nd video window #2 in FramePacking mode. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2[6] GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	POSY	RESERVED	POSX

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
26:16	POSY	Y position of the 2nd video window #2 Encoded value (from 0 to 2047) to specify the Y position of the video window #2 .The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
10:0	POSX	X position of the 2nd video window #2 Encoded value (from 0 to 2047) to specify the X position of the video window #2. The first pixel on the left of the display screen has the X-position 0.	RW	0x000

Table 11-249. DISPC_VID3_POSITION2

Address Offset	0x0000 024C		
Physical Address	0x5800 124C	**Instance**	DISPC
Description	The register configures the position of the 2nd video window #3 in FramePacking mode. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2[6] GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	POSY	RESERVED	POSX

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
26:16	POSY	Y position of the 2nd video window #2 Encoded value (from 0 to 2047) to specify the Y position of the video window #2 .The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
10:0	POSX	X position of the 2nd video window #2 Encoded value (from 0 to 2047) to specify the X position of the video window #2. The first pixel on the left of the display screen has the X-position 0.	RW	0x000

Table 11-250. DISPC_VID3_ACCU_j

Address Offset	0x0000 0300 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1300 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 3 (DISPC_VID3_ACCU_0 and DISPC_VID3_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	VERTICALACCU	RESERVED	HORIZONTALACCU

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from –1024 to 1023).	RW	0x000

Table 11-251. DISPC_VID3_BA_j

Address Offset	0x0000 0308 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1308 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the video buffer for the video window 3 (DISPC_VID3_BA_0 and DISPC_VID3_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID3_BA_0 is used)). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address Base address of the video buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2:0, byte alignment is supported)). It case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-252. DISPC_VID3_FIR_COEF_H_i

Address Offset	0x0000 0310 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1310 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00

Bits	Field Name	Description	Type	Reset
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-253. DISPC_VID3_FIR_COEF_HV_i

Address Offset	0x0000 0314 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1314 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:24	23:16	15:8	7:0
FIRVC2	FIRVC1	FIRVC0	FIRHC4

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-254. DISPC_VID3_FIR_COEF_V_i

Address Offset	0x0000 0350 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1350 + (0x4 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:16	15:8	7:0
RESERVED	FIRVC22	FIRVC00

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-255. DISPC_VID3_ATTRIBUTES

Address Offset	0x0000 0370		
Physical Address	0x5800 1370	**Instance**	DISPC
Description	The register configures the attributes of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:30	29	28	27:26	25	24	23	22	21	20	19	18	17	16	15:14	13:12	11	10	9	8	7	6:5	4:1	0
CHANNELOUT2	BURSTTYPE	PREMULTIPLYALPHA	ZORDER	ZORDERENABLE	SELFREFRESH	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	FORCE1DTILEDMODE	BUFPRELOAD	RESERVED	SELFREFRESHAUTO	CHANNELOUT	BURSTSIZE	ROTATION	FULLRANGE	REPLICATIONENABLE	COLORCONVENABLE	FRAMEPACKINGMODE	HRESIZECONF	RESIZEENABLE	FORMAT	ENABLE

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (should be set to zero) wr: immediate 0x0: Primary LCD output selected. 0x1: Secondary LCD output selected. 0x2: Third LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPLYALPHA	The field configures the DISPC VID3 to process incoming data as premultiplied alpha data or non premultiplied alpha data. Default setting is non premultiplied alpha data. 0x0: Non premultiplyalpha data color component 0x1: Premultiplyalpha data color component	RW	0
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0. 0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0

Bits	Field Name	Description	Type	Reset
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is used only when the Z-order is enabled. 0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	SELFREFRESH	Enables the self refresh of the video window from its own DMA buffer only. 0x0: The video pipeline accesses the interconnect to fetch data from the system memory. 0x1: The video pipeline does not need anymore to fetch data from memory. Only the DMA buffer associated with the video3 is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0
23	ARBITRATION	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The video pipeline is one of the normal priority pipeline. 0x1: The video pipeline is one of the high priority pipeline.	RW	0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV4:2:0. 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0
21	VERTICALTAPS	Video vertical resize tap number 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. The associated bit fields for the 2 other taps coefficients do not need to be initialized. 0x1: 5 taps are used for the vertical filtering logic.	RW	0
20	FORCE1DTILEDMODE	Force TILED regions access to 1D or 2D. 0x0: 2D accesses for tiled regions 0x1: 1D accesses for tiled regions	RW	0
19	BUFPRELOAD	Video Preload Value 0x0: Hardware prefetches pixels up to the preload value defined in the preload register 0x1: Hardware prefetches pixels up to high threshold value	RW	0
18	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
17	SELFREFRESHAUTO	Automatic self-refresh mode 0x0: The transition from SELFREFRESH disabled to enabled is controlled by SW. 0x1: The transition from SELFREFRESH disabled to enabled is controlled only by hardware.	RW	0
16	CHANNELOUT	Video channel out configuration: LCD, WB or TV. wr: immediate 0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back). 0x1: TV output selected	RW	0

Bits	Field Name	Description	Type	Reset
15:14	BURSTSIZE	Video DMA burst size 0x0: 2 × 128-bit bursts 0x1: 4 × 128-bit bursts 0x3: Reserved 0x2: 8 × 128-bit bursts	RW	0x2
13:12	ROTATION	Video rotation flag 0x0: No rotation 0x1: Rotation by 90 degrees 0x3: Rotation by 270 degrees 0x2: Rotation by 180 degrees	RW	0x0
11	FULLRANGE	Color Space Conversion full range setting. 0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	REPLICATIONENABLE	Replication enable 0x0: Disable Video replication logic 0x1: Enable Video replication logic	RW	1
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion YUV to RGB 0x1: Enable Color Space Conversion YUV to RGB	RW	0
8	FRAMEPACKINGMODE	Frame packing mode control. 0x0: Frame Packing mode is disabled 0x1: Frame Packing mode is enabled	RW	0
7	HRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0
6:5	RESIZEENABLE	Video resize enable 0x0: Disable both horizontal and vertical resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0

Bits	Field Name	Description	Type	Reset
4:1	FORMAT	Video format. It defines the pixel format when fetching the video 3 picture into memory. 0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: RGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0x3: BGRA32-8888 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	Video Enable 0x0: Video disabled (video pipeline inactive and window not present) 0x1: Video enabled (video pipeline active and window present on the screen)	RW	0

Table 11-256. DISPC_VID3_CONV_COEF0

Address Offset	0x0000 0374		
Physical Address	0x5800 1374	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	RCR	RESERVED	RY

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	RCR	RCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RY	RY coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-257. DISPC_VID3_CONV_COEF1

Address Offset	0x0000 0378		
Physical Address	0x5800 1378	**Instance**	DISPC

Table 11-257. DISPC_VID3_CONV_COEF1 (continued)

Description	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					GY											RESERVED					RCB										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GY	GY coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RCB	RCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-258. DISPC_VID3_CONV_COEF2

Address Offset	0x0000 037C		
Physical Address	0x5800 137C	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					GCB											RESERVED					GCR										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GCB	GCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	GCR	GCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-259. DISPC_VID3_CONV_COEF3

Address Offset	0x0000 0380		
Physical Address	0x5800 1380	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					BCR											RESERVED					BY										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	BCR	BCr coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	BY	BY coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-260. DISPC_VID3_CONV_COEF4

Address Offset	0x0000 0384		
Physical Address	0x5800 1384	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–11	10–0
RESERVED	BCB

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	BCB	BCb coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-261. DISPC_VID3_BUF_SIZE_STATUS

Address Offset	0x0000 0388		
Physical Address	0x5800 1388	**Instance**	DISPC
Description	The register defines the DMA buffer size for the video pipeline 3.		
Type	R		

31–16	15–0
RESERVED	BUFSIZE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer Size in number of 128 bits.	R	0x0800

Table 11-262. DISPC_VID3_BUF_THRESHOLD

Address Offset	0x0000 038C		
Physical Address	0x5800 138C	**Instance**	DISPC
Description	The register configures the DMA buffer associated with the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–16	15–0
BUFHIGHTHRESHOLD	BUFLOWTHRESHOLD

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer high threshold number of 128 bits defining the threshold value	RW	0x07FF
15:0	BUFLOWTHRESHOLD	DMA buffer low threshold number of 128 bits defining the threshold value	RW	0x07F8

Table 11-263. DISPC_VID3_FIR

Address Offset	0x0000 0390		
Physical Address	0x5800 1390	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 3. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31 30 29	28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13	12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	FIRVINC	RESERVED	FIRHINC

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-264. DISPC_VID3_PICTURE_SIZE

Address Offset	0x0000 0394		
Physical Address	0x5800 1394	**Instance**	DISPC
Description	The register configures the size of the video picture associated with the video layer 3 before up/down-scaling. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31 30 29 28	27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8 7 6 5 4 3 2 1 0
RESERVED	MEMSIZEY	RESERVED	MEMSIZEX

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:16	MEMSIZEY	Number of lines of the video picture Encoded value (from 1 to 4096) to specify the number of lines of the video picture in memory (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded 2^{11}.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
10:0	MEMSIZEX	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus 1). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit. (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded 2^{11}.	RW	0x000

Table 11-265. DISPC_VID3_PIXEL_INC

Address Offset	0x0000 0398		
Physical Address	0x5800 1398	**Instance**	DISPC
Description	The register configures the number of bytes to increment between two pixels for the buffer associated with the video window 3. For more information, see *Predecimation*.The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31-8	7-0
RESERVED	PIXELINC

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1 + n * bpp means increment of n pixels. For YUV4:2:0, maximum supported value is 128.	RW	0x01

Table 11-266. DISPC_VID3_POSITION

Address Offset	0x0000 039C		
Physical Address	0x5800 139C	**Instance**	DISPC
Description	The register configures the position of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31-27	26-16	15-11	10-0
RESERVED	POSY	RESERVED	POSX

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	POSY	Y position of the video window 2 Encoded value (from 0 to 2047) to specify the Y position of the video window 2 .The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
10:0	POSX	X position of the video window 2 Encoded value (from 0 to 2047) to specify the X position of the video window 2. The first pixel on the left of the display screen has the X-position 0.	RW	0x000

Table 11-267. DISPC_VID3_PRELOAD

Address Offset	0x0000 03A0		
Physical Address	0x5800 13A0	**Instance**	DISPC
Description	The register configures the DMA buffer of the video 3 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–12	11–0
RESERVED	PRELOAD

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

Table 11-268. DISPC_VID3_ROW_INC

Address Offset	0x0000 03A4		
Physical Address	0x5800 13A4	**Instance**	DISPC
Description	The register configures the number of bytes to increment at the end of the row for the buffer associated with the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31–0
ROWINC

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from 2^{31}1 to 2^{31}) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1 + n * bpp means increment of n pixels. The value 1 (n + 1) * bpp means decrement of n pixels.	RW	0x0000 0001

Table 11-269. DISPC_VID3_SIZE

Address Offset	0x0000 03A8		
Physical Address	0x5800 13A8	**Instance**	DISPC
Description	The register configures the size of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SIZEY												RESERVED					SIZEX										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:16	SIZEY	Number of lines of the video 3 Encoded value (from 1 to 4096) to specify the number of lines of the video window 3. Program to value minus 1.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
10:0	SIZEX	Number of pixels of the video window 3 Encoded value (from 1 to 2048) to specify the number of pixels of the video window 3. Program to value minus 1.	RW	0x000

Table 11-270. DISPC_DEFAULT_COLOR2

Address Offset	0x0000 03AC		
Physical Address	0x5800 13AC	**Instance**	DISPC
Description	The control register allows to configure the default solid background color for the secondary LCD Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEFAULTCOLOR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	DEFAULTCOLOR	24-bit RGB color value to specify the default solid color to display when there is no data from the overlays.	RW	0x000000

Table 11-271. DISPC_TRANS_COLOR2

Address Offset	0x0000 03B0		
Physical Address	0x5800 13B0	**Instance**	DISPC
Description	The register sets the transparency color value for the video/graphics overlays for the secondary LCD output. Shadow register, updated on VFP start period of the secondary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TRANSCOLORKEY																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	TRANSCOLORKEY	Transparency Color Key Value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24 **NOTE: CLUT and BITMAP formats are not supported in this family of devices.**	RW	0x000000

Table 11-272. DISPC_CPR2_COEF_B

Address Offset	0x0000 03B4		
Physical Address	0x5800 13B4	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the Blue component. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		

Table 11-272. DISPC_CPR2_COEF_B (continued)

Type	RW

31–22	21	20–11	10	9–0
BR	RESERVED	BG	RESERVED	BB

Bits	Field Name	Description	Type	Reset
31:22	BR	BR coefficient encoded signed value (from –512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	BG	BG coefficient encoded signed value (from –512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	BB	BB coefficient encoded signed value (from –512 to 511).	RW	0x000

Table 11-273. DISPC_CPR2_COEF_G

Address Offset	0x0000 03B8		
Physical Address	0x5800 13B8	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the Green component. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31–22	21	20–11	10	9–0
GR	RESERVED	GG	RESERVED	GB

Bits	Field Name	Description	Type	Reset
31:22	GR	GR coefficient encoded signed value (from –512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	GG	GG coefficient encoded signed value (from –512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	GB	GB coefficient encoded signed value (from –512 to 511).	RW	0x000

Table 11-274. DISPC_CPR2_COEF_R

Address Offset	0x0000 03BC		
Physical Address	0x5800 13BC	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the Red component. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31–22	21	20–11	10	9–0
RR	RESERVED	RG	RESERVED	RB

Bits	Field Name	Description	Type	Reset
31:22	RR	RR coefficient encoded signed value (from –512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	RG	RG coefficient encoded signed value (from –512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0

Bits	Field Name	Description	Type	Reset
9:0	RB	RB coefficient encoded signed value (from –512 to 511).	RW	0x000

Table 11-275. DISPC_DATA2_CYCLE1

Address Offset	0x0000 03C0		
Physical Address	0x5800 13C0	**Instance**	DISPC
Description	The control register configures the output data format for 1st cycle. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31:28	27:24	23:21	20:16	15:12	11:8	7:5	4:0
RESERVED	BITALIGNMENTPIXEL2	RESERVED	NBBITSPIXEL2	RESERVED	BITALIGNMENTPIXEL1	RESERVED	NBBITSPIXEL1

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment. Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment. Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-276. DISPC_DATA2_CYCLE2

Address Offset	0x0000 03C4		
Physical Address	0x5800 13C4	**Instance**	DISPC
Description	The control register configures the output data format for 2nd cycle. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31:28	27:24	23:21	20:16	15:12	11:8	7:5	4:0
RESERVED	BITALIGNMENTPIXEL2	RESERVED	NBBITSPIXEL2	RESERVED	BITALIGNMENTPIXEL1	RESERVED	NBBITSPIXEL1

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment. Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment. Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-277. DISPC_DATA2_CYCLE3

Address Offset	0x0000 03C8		
Physical Address	0x5800 13C8	**Instance**	DISPC
Description	The control register configures the output data format for 3rd cycle. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31:28	27:24	23:21	20:16	15:12	11:8	7:5	4:0
RESERVED	BITALIGNMENTPIXEL2	RESERVED	NBBITSPIXEL2	RESERVED	BITALIGNMENTPIXEL1	RESERVED	NBBITSPIXEL1

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment. Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment. Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-278. DISPC_SIZE_LCD2

Address Offset	0x0000 03CC		
Physical Address	0x5800 13CC	**Instance**	DISPC
Description	The register configures the panel size (horizontal and vertical). It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD. A delta value is used to indicate if the odd field has same vertical size as the even field or +/- one line.		
Type	RW		

31:28	27:16	15:14	13:12	11:0
RESERVED	LPP	DELTA_LPP	RESERVED	PPL

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
27:16	LPP	Lines per panel encoded value (from 1 to 4096) to specify the number of lines per panel (program to value minus 1).	RW	0x000
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: same size 0x1: odd size = even size +1 0x2: Odd size = even size –1	RW	0x0
13:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:0	PPL	Pixels per line encoded value (from 1 to 4096) to specify the number of pixels contains within each line on the display (program to value minus 1). In STALL mode, any value is valid. In non STALL mode, only values multiple of 8 pixels are valid.	RW	0x000

Table 11-279. DISPC_TIMING_H2

Address Offset	0x0000 0400		
Physical Address	0x5800 1400	**Instance**	DISPC
Description	The register configures the timing logic for the HSYNC signal. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBP												HFP												HSW							

Bits	Field Name	Description	Type	Reset
31:20	HBP	Horizontal back porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (program to value minus 1).	RW	0x000
19:8	HFP	Horizontal front porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted (program to value minus 1).	RW	0x000
7:0	HSW	Horizontal synchronization pulse width. Encoded value (from 1 to 256) to specify the number of pixel clock periods to pulse the line clock at the end of each line (program to value minus 1).	RW	0x00

Table 11-280. DISPC_TIMING_V2

Address Offset	0x0000 0404		
Physical Address	0x5800 1404	**Instance**	DISPC
Description	The register configures the timing logic for the VSYNC signal. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBP												VFP												VSW							

Bits	Field Name	Description	Type	Reset
31:20	VBP	Vertical back porch. Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the beginning of a frame before the first set of pixels is output to the display.	RW	0x000
19:8	VFP	Vertical front porch. Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the end of each frame.	RW	0x000
7:0	VSW	Vertical synchronization pulse width. In active mode, encoded value (from 1 to 256) to specify the number of line clock periods (program to value minus 1) to pulse the frame clock (VSYNC) pin at the end of each frame after the end of frame wait (VFP) period elapses. Frame clock uses as VSYNC signal in active mode.	RW	0x00

Table 11-281. DISPC_POL_FREQ2

Address Offset	0x0000 0408		
Physical Address	0x5800 1408	**Instance**	DISPC
Description	The register configures the signal configuration. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ALIGN	ONOFF	RF	IEO	IPC	IHS	IVS	ACBI	ACB

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
18	ALIGN	Defines the alignment between HSYNC and VSYNC assertion. 0x0: VSYNC and HSYNC are not aligned. 0x1: VSYNC and HSYNC assertions are aligned.	RW	0
17	ONOFF	HSYNC/VSYNC Pixel clock Control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data. 0x1: HSYNC and VSYNC are driven according to bit 16. **Note:** Control module register CTRL_CORE_SMA_SW_1[23]DSS_CH1_ON_OFF must be set to match	RW	0
16	RF	Program HSYNC/VSYNC Rise or Fall 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit 17 set to 1). 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit 17 set to 1). **Note:** Control module register CTRL_CORE_SMA_SW_1[17]DSS_CH1_RF must be set to match	RW	0
15	IEO	Invert output enable 0x0: Ac-bias is active high (active display mode). 0x1: Ac-bias is active low (active display mode).	RW	0
14	IPC	Invert pixel clock 0x0: Data is driven on the LCD data lines on the rising-edge of the pixel clock. 0x1: Data is driven on the LCD data lines on the falling-edge of the pixel clock. **Note:** Control module register CTRL_CORE_SMA_SW_1[20]DSS_CH1_IPC must be set to match	RW	0
13	IHS	Invert HSYNC 0x0: Line clock pin is active high and inactive low. 0x1: Line clock pin is active low and inactive high.	RW	0
12	IVS	Invert VSYNC 0x0: Frame clock pin is active high and inactive low. 0x1: Frame clock pin is active low and inactive high.	RW	0
11:8	ACBI	AC Bias Pin transitions per interrupt Value (from 0 to 15) used to specify the number of AC Bias pin transitions	RW	0x0
7:0	ACB	AC Bias Pin Frequency Value (from 0 to 255) used to specify the number of line clocks to count before transitioning the AC Bias pin. This pin is used to periodically invert the polarity of the power supply to prevent DC charge build-up within the display.	RW	0x00

Table 11-282. DISPC_DIVISOR2

Address Offset	0x0000 040C		
Physical Address	0x5800 140C	**Instance**	DISPC

Table 11-282. DISPC_DIVISOR2 (continued)

Description	The register configures the divisors. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LCD								RESERVED								PCD							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:16	LCD	Display controller logic clock divisor value (from 1 to 255) to specify the intermediate pixel clock frequency based on the LCD2_CLK. The value 0 is invalid.	RW	0x04
15:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
7:0	PCD	Pixel clock divisor value (from 1 to 255) to specify the frequency of the pixel clock based on the LCD2_CLK divided by DISPC_DIVISOR2.LCD value. The value 0 is invalid.	RW	0x01

Table 11-283. DISPC_WB_ACCU_j

Address Offset	0x0000 0500 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1500 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the write back pipeline (DISPC_WB_ACCU_0 and DISPC_WB_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for ARGB and Y setting. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					VERTICALACCU											RESERVED					HORIZONTALACCU										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accumulator value Encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accumulator value encoded value (from –1024 to 1023).	RW	0x000

Table 11-284. DISPC_WB_BA_j

Address Offset	0x0000 0508 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1508 + (0x4 * j)	**Instance**	DISPC

Table 11-284. DISPC_WB_BA_j (continued)

Description	The register configures the base address of the WB buffer (DISPC_WB_BA_0 and DISPC_WB_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_WB_BA_0 is used). Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Write-back base address Base address of the WB buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2;0, byte alignment is supported)). It case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-285. DISPC_WB_FIR_COEF_H_i

Address Offset	0x0000 0510 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1510 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00

Bits	Field Name	Description	Type	Reset
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-286. DISPC_WB_FIR_COEF_HV_i

Address Offset	0x0000 0514 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1514 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
FIRVC2	FIRVC1	FIRVC0	FIRHC4

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-287. DISPC_WB_FIR_COEF_V_i

Address Offset	0x0000 0550 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1550 + (0x4 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	FIRVC22	FIRVC00

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000

Bits	Field Name	Description	Type	Reset
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-288. DISPC_WB_ATTRIBUTES

Address Offset	0x0000 0570		
Physical Address	0x5800 1570	**Instance**	DISPC
Description	The register configures the attributes of the viwrite back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31-28	27	26-24	23	22	21	20	19	18-16	15-14	13-12	11	10	9	8	7	6-5	4-1	0
IDLENUMBER	IDLESIZE	CAPTUREMODE	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	FORCE1DTILEDMODE	WRITEBACKMODE	CHANNELIN	BURSTSIZE	RESERVED	FULLRANGE	TRUNCATIONENABLE	COLORCONVENABLE	BURSTTYPE	ALPHAENABLE	RESIZEENABLE	FORMAT	ENABLE

Bits	Field Name	Description	Type	Reset
31:28	IDLENUMBER	Determines the number of idles between requests on the L3_MAIN interconnect. It is only used when the write-back pipeline does data transfer from memory to memory. When the output of an overlay is stored in memory through the write-back pipeline in capture mode, the bit field IDLENUMBER is ignored since a timing generator is used to time the transfer. The number of IDLE cycles is IDLENUMBER (from 0 to 15) if IDLESIZE = 0. The number of IDLE cycles is IDLENUMBERx8 (from 0 to 120) if IDLESIZE = 1 and BURSTSIZE = 2. The number of IDLE cycles is IDLENUMBERx4 (from 0 to 60) if IDLESIZE = 1 and BURSTSIZE = 1. The number of IDLE cycles is IDLENUMBERx2 (from 0 to 30) if IDLESIZE = 1 and BURSTSIZE = 0.	RW	0x0
27	IDLESIZE	Determines if the IDLENUMBER corresponds to a number of bursts or singles. 0x0: The number of idles between requests is defined by IDLENUMBER as number of cycles. 0x1: The number of idles between requests is defined by IDLENUMBER multiplied by burst size as number of cycles.	RW	0

Bits	Field Name	Description	Type	Reset
26:24	CAPTUREMODE	Defines the frame rate capture. 0x6: Only one out of six frames is captured. The first one is captured then the second one is skipped and so on. 0x1: Only one frame is captured. 0x7: Only one out of seven frames is captured. The first one is captured then the second one is skipped and so on. 0x0: All frames are captures until the write-back channel is disabled or there is no more data generated by the overlay or the pipeline attached to the write-back channel. 0x2: Only one out of two frames is captured. The first one is captured, and then the second one is skipped, and so on. 0x4: Only one out of four frames is captured. The first one is captured, and then the second one is skipped, and so on. 0x5: Only one out of five frames is captured. The first one is captured, and then the second one is skipped, and so on. 0x3: Only one out of three frames is captured. The first one is captured, and then the second one is skipped, and so on.	RW	0x0
23	ARBITRATION	Determines the priority of the write-back pipeline. The write-back pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The write-back pipeline is one of the normal priority pipeline. 0x1: The write-back pipeline is one of the high priority pipeline.	RW	0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV4:2:0. 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0
21	VERTICALTAPS	Video Vertical Resize Tap Number 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. 0x1: 5 taps are used for the vertical filtering logic.	RW	0
20	FORCE1DTILEDMODE	Force TILED regions access to 1D or 2D. 0x0: 2D accesses for tiled regions 0x1: 1D accesses for tiled regions	RW	0x0
19	WRITEBACKMODE	When connected to the overlay output of a channel the write back can operate as a simple transfer from memory to memory (composition engine) or as a capture channel. 0x0: Capture mode (default mode) 0x1: Memory-to-memory mode	RW	0x0

Bits	Field Name	Description	Type	Reset
18:16	CHANNELIN	Video Channel In configuration WR: immediate 0x6: Video3 pipeline output 0x1: Secondary LCD output 0x0: Primary LCD overlay output 0x2: TV overlay output 0x4: Video1 pipeline output 0x5: Video2 pipeline output 0x3: Graphics pipeline output 0x7: Third LCD output	RW	0x0
15:14	BURSTSIZE	Write-back DMA Burst Size 0x0: 2 × 128-bit bursts 0x1: 4 × 128-bit bursts 0x3: Reserved 0x2: 8 × 128-bit bursts	RW	0x2
13:12	RESERVED	Reserved	RW	0x0
11	FULLRANGE	Color Space Conversion full range setting. 0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	TRUNCATIONENABLE	It applies only when the input format to the write-back pipeline from the overlay or directly from one of the pipelines is ARGB32. If the format is one of the YUV supported formats, the bit field is ignored. 0x0: Disable truncation logic 0x1: Enable truncation logic from ARGB32 to the pixel format defined in the field FORMAT.	RW	0
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion RGB to YUV 0x1: Enable Color Space Conversion RGB to YUV	RW	0
8	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
7	ALPHAENABLE	Premultiplied alpha enable Read 0x1: Enabled Read 0x0: Disabled. This bit also disable the logic present in the associated channel out that compute the alpha component sent to the WB pipe. When the WB is configured to copy back one of the output channels (output of overlay), the following configurations are available: 0x1: The WB pipe copies back to memory the premultiplied alpha calculated through the overlay. 0x0: The alpha value is not written back.	RW	0

Bits	Field Name	Description	Type	Reset
6:5	RESIZEENABLE	Resize Enable 0x0: Disable the resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0
4:1	FORMAT	Write-back format. It defines the pixel format when storing the write-back picture into memory. 0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: xRGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0x3: BGRA32-8888 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	Write-back enable. wr: immediate 0x0: Write-back disabled 0x1: Write-back enabled	RW	0

Table 11-289. DISPC_WB_CONV_COEF0

Address Offset	0x0000 0574		
Physical Address	0x5800 1574	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the write back pipeline (YUV4:4:4 to RGB24). Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					YG											RESERVED					YR										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
26:16	YG	YG coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	YR	YR coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-290. DISPC_WB_CONV_COEF1

Address Offset	0x0000 0578		
Physical Address	0x5800 1578	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	CRR	RESERVED	YB

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	CRR	CrR coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	YB	YB coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-291. DISPC_WB_CONV_COEF2

Address Offset	0x0000 057C		
Physical Address	0x5800 157C	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	CRB	RESERVED	CRG

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	CRB	CrB coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Bits	Field Name	Description	Type	Reset
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	CRG	CrG coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-292. DISPC_WB_CONV_COEF3

Address Offset	0x0000 0580		
Physical Address	0x5800 1580	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	CBG	RESERVED	CBR

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	CBG	CbG coefficient encoded signed value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	CBR	CbR coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-293. DISPC_WB_CONV_COEF4

Address Offset	0x0000 0584		
Physical Address	0x5800 1584	**Instance**	DISPC
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31:11	10:0
RESERVED	CBB

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	CBB	CbB coefficient encoded signed value (from –1024 to 1023).	RW	0x000

Table 11-294. DISPC_WB_BUF_SIZE_STATUS

Address Offset	0x0000 0588		
Physical Address	0x5800 1588	**Instance**	DISPC
Description	The register defines the DMA buffer size for the write back pipeline.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer Size in number of 128 bits	R	0x0800

Table 11-295. DISPC_WB_BUF_THRESHOLD

Address Offset	0x0000 058C		
Physical Address	0x5800 158C	**Instance**	DISPC
Description	The register configures the DMA buffer associated with the write-back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer high threshold number of 128 bits defining the threshold value	RW	0x07FF
15:0	BUFLOWTHRESHOLD	DMA buffer low threshold number of 128 bits defining the threshold value	RW	0x07F8

Table 11-296. DISPC_WB_FIR

Address Offset	0x0000 0590		
Physical Address	0x5800 1590	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the write back pipeline. It is used for ARGB and Y setting. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			FIRVINC													RESERVED			FIRHINC												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-297. DISPC_WB_PICTURE_SIZE

Address Offset	0x0000 0594		
Physical Address	0x5800 1594	**Instance**	DISPC
Description	The register configures the size of the write-back picture associated with the write back pipeline after up/down-scaling. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				MEMSIZEY												RESERVED					MEMSIZEX										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
27:16	MEMSIZEY	Number of lines of the wb picture in memory. Encoded value (from 1 to 4096) to specify the number of lines of the picture in memory (program to value minus 1).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	MEMSIZEX	Number of pixels of the wb picture in memory. Encoded value (from 1 to 2048) to specify the number of pixels of the picture in memory (program to value minus 1).	RW	0x000

Table 11-298. DISPC_WB_PIXEL_INC

Address Offset	0x0000 0598		
Physical Address	0x5800 1598	**Instance**	DISPC
Description	The register configures the number of bytes to increment between two pixels for the buffer associated with the write back pipeline. The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PIXELINC

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Values other than 1 are invalid	RW	0x01

Table 11-299. DISPC_WB_ROW_INC

Address Offset	0x0000 05A4		
Physical Address	0x5800 15A4	**Instance**	DISPC
Description	The register configures the number of bytes to increment at the end of the row for the buffer associated with the vwrite back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used. Note: The DISPC_WB_ROW_INC register can be used only in 2D mode (using the Tiler). In order to use the DISPC_WB_ROW_INC register, the DISPC_WB_ATTRIBUTES[8] BURSTTYPE bit must be set to 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from $2^{31}1$ to 2^{31}) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1 + n *bpp means increment of n pixels. The value 1 (n + 1) * bpp means decrement of n pixels.	RW	0x0000 0001

Table 11-300. DISPC_WB_SIZE

Address Offset	0x0000 05A8		
Physical Address	0x5800 15A8	**Instance**	DISPC
Description	The register configures the size of the output of overlay connected to the write-back pipeline when the overlay output is only used by the write-back pipeline. When the overlay is output on the primary LCD or secondary LCD or TV outputs, the size of the frame is defined in the DISPC_SIZE_LCD1, DISPC_SIZE_LCD2, and DISPC_SIZE_TV respectively. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SIZEY												RESERVED					SIZEX										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
27:16	SIZEY	Number of lines of the Write-back picture Encoded value (from 1 to 4096) to specify the number of lines of the write-back picture from overlay or pipeline. Program to value minus 1.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZEX	Number of pixels of the Write-back picture Encoded value (from 1 to 2048) to specify the number of pixels of the write-back picture from overlay or pipeline. Program to value minus 1.	RW	0x000

Table 11-301. DISPC_VID1_BA_UV_j

Address Offset	0x0000 0600 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1600 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the UV buffer for the video window 1. (DISPC_VID1_BA_UV_0 and DISPC_VID1_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID1_BA_UV_0 is used)). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address aligned on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-302. DISPC_VID2_BA_UV_j

Address Offset	0x0000 0608 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1608 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the UV buffer for the video window 2. (DISPC_VID2_BA_UV_0 and DISPC_VID2_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID2_BA_UV_0 is used)). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address aligned on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-303. DISPC_VID3_BA_UV_j

Address Offset	0x0000 0610 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1610 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the UV buffer for the video window 3. (DISPC_VID3_BA_UV_0 and DISPC_VID3_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID3_BA_UV_0 is used)). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address alinged on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-304. DISPC_WB_BA_UV_j

Address Offset	0x0000 0618 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1618 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the base address of the UV buffer for the write-back pipeline. (DISPC_WB_BA_UV_0 and DISPC_WB_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_WB_BA_UV_0 is used)). Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		

Table 11-304. DISPC_WB_BA_UV_j (continued)

Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BA</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address alinged on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

Table 11-305. DISPC_CONFIG2

Address Offset	0x0000 0620		
Physical Address	0x5800 1620	**Instance**	DISPC
Description	The control register configures the Display Controller module for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD or VFP start period of the third LCD or EVSYNC		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">SLCDINTERLEAVE</td><td>FULLRANGE</td><td>COLORCONVENABLE</td><td>FIDFIRST</td><td>OUTPUTMODEENABLE</td><td>BT1120ENABLE</td><td>BT656ENABLE</td><td colspan="3">RESERVED</td><td>BUFFERHANDCHECK</td><td>CPR</td><td colspan="3">RESERVED</td><td>TCKLCDSELECTION</td><td>TCKLCDENABLE</td><td>RESERVED</td><td>ACBIASGATED</td><td>VSYNCGATED</td><td>HSYNCGATED</td><td>PIXELCLOCKGATED</td><td>PIXELDATAGATED</td><td colspan="3">RESERVED</td><td>PIXELGATED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:26	SLCDINTERLEAVE	sLCD Interleave Pattern	RW	0x0
25	FULLRANGE	Color space conversion full range setting. 0x0: Limited range selected. 0x1: Full range selected.	RW	0
24	COLORCONV ENABLE	Enable the color space conversion. It shall be reset when CPR bit field is set to 0x1. 0x0: Disable color space conversion RGB to YUV 0x1: Enable color space conversion RGB to YUV	RW	0
23	FIDFIRST	Selects the first field to output in case of interlace mode. In case of progressive mode, the value is not used. 0x0: First field is even. 0x1: Odd field is first.	RW	0

Bits	Field Name	Description	Type	Reset
22	OUTPUTMODE ENABLE	Selects between progressive and interlace mode for the secondary LCD output. 0x0: Progressive mode selected. 0x1: Interlace mode selected.	RW	0
21	BT1120ENABLE	Selects BT.1120 format on the primary LCD output. It is not posssible to enable BT.656 and BT.1120 at the same time one the same LCD output. wr: VFP start of primary LCD 0x0: BT.1120 is disabled 0x1: BT.1120 is enabled.	RW	0
20	BT656ENABLE	Selects BT.656 format on the primary LCD output. It is not posssible to enable BT.656 and BT.1120 at the same time one the same LCD output. wr: VFP start of primary LCD 0x0: BT.656 is disabled. 0x1: BT.656 is enabled.	RW	0
19:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
15	CPR	Color Phase Rotation Control secondary LCD output). It shall be reset when ColorConvEnable bit field is set to 1. wr: VFP start period of secondary LCD output 0x0: Color phase rotation disabled 0x1: Color phase rotation enabled	RW	0
14:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11	TCKLCD SELECTION	Transparency color key selection (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0
10	TCKLCDENABLE	Transparency color key enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Disable the transparency color key for the LCD 0x1: Enable the transparency color key for the LCD	RW	0
9	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
8	ACBIASGATED	ACBias gated enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: AcBias gated disabled 0x1: AcBias gated enabled	RW	0
7	VSYNCGATED	VSYNC gated enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: VSYNC gated disabled 0x1: VSYNC gated enabled	RW	0
6	HSYNCGATED	HSYNC gated enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: HSYNC gated disabled 0x1: HSYNC gated enabled	RW	0
5	PIXELCLOCK GATED	Pixel clock gated enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Pixel clock gated disabled 0x1: Pixel clock gated enabled	RW	0

Bits	Field Name	Description	Type	Reset
4	PIXELDATA GATED	Pixel data gated enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Pixel data gated disabled 0x1: Pixel data gated enabled	RW	0
3:1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
0	PIXELGATED	Pixel gated enable (only for active matrix) (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Pixel clock always toggles (only in active matrix mode). 0x1: Pixel clock only toggles when there is valid data to display (only in active matrix mode).	RW	0

Table 11-306. DISPC_VID1_ATTRIBUTES2

Address Offset	0x0000 0624		
Physical Address	0x5800 1624	**Instance**	DISPC
Description	The register configures the attributes of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:12	11:9	8	7	6:4	3:1	0
RESERVED	SUBSAMPLINGPATTERN	YUVCHROMARESAMPLING	RESERVED	VC1_RANGE_CBCR	VC1_RANGE_Y	VC1ENABLE

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:9	SUBSAMPLINGPATTERN	Subsampling pattern setting.	RW	0x0
8	YUVCHROMARE SAMPLING	The YUV chrominance can be resampled using averaging of the adjacent chrominance samples, without using the polyphase filter for 4:2:2 input or can be calculated using the polyphase filter for 4:2:2/4:2:0. The polyphase filter is mandatory for the 4:2:0 format. This bit controls the order in which the processing is done on the video pipe. 0x0: When input is 4:2:2, the missing chrominance samples are calculated by averaging the adjacent samples if DISPC_VID1_ATTRIBUTES. ROTATION=0 only. Other rotation configurations are not supported. 0x1: For 4:2:2 (or 4:2:0), the missing chrominance samples are calculated by filtering the adjacent samples (5-tap polyphase filter). See Figure 11-49, *Configuration 2: Video Pipeline*. All rotation configurations are supported.	RW	0
7	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0

Bits	Field Name	Description	Type	Reset
6:4	VC1_RANGE_ CBCR	Defines the VC-1 range value for the CbCr component from 0 to 7.	RW	0x0
3:1	VC1_RANGE_Y	Defines the VC-1 range value for the Y component from 0 to 7.	RW	0x0
0	VC1ENABLE	Enable/disable the VC-1 range mapping processing. The bit field is ignored if the format is not one of the supported YUV formats. 0x0: VC-1 range mapping disabled 0x1: VC-1 range mapping enabled	RW	0

Table 11-307. DISPC_VID2_ATTRIBUTES2

Address Offset	0x0000 0628		
Physical Address	0x5800 1628	**Instance**	DISPC
Description	The register configures the attributes of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:12	11:9	8	7	6:4	3:1	0
RESERVED	SUBSAMPLINGPATTERN	YUVCHROMARESAMPLING	RESERVED	VC1_RANGE_CBCR	VC1_RANGE_Y	VC1ENABLE

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:9	SUBSAMPLINGPATTERN	Subsampling pattern setting.	RW	0x0
8	YUVCHROMARE SAMPLING	The YUV chrominance can be resampled using averaging of the adjacent chrominance samples, without using the polyphase filter for 4:2:2 input or can be calculated using the polyphase filter for 4:2:2/4:2:0. The polyphase filter is mandatory for the 4:2:0 format. This bit controls the order in which the processing is done on the video pipe. 0x0: When input is in 4:2:2, the missing chrominance samples are calculated by averaging the adjacent samples if DISPC_VID1_ATTRIBUTES. ROTATION=0 only. Other rotation configurations are not supported. 0x1: For 4:2:2 (or 4:2:0), the missing chrominance samples are calculated by filtering the adjacent samples (5-tap polyphase filter). See Figure 11-49, *Configuration 2: Video Pipeline*. All rotation configurations are supported.	RW	0
7	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
6:4	VC1_RANGE_ CBCR	Defines the VC-1 range value for the CbCr component from 0 to 7.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:1	VC1_RANGE_Y	Defines the VC-1 range value for the Y component from 0 to 7.	RW	0x0
0	VC1ENABLE	Enable/disable the VC-1 range mapping processing. The bit field is ignored if the format is not one of the supported YUV formats. 0x0: VC-1 range mapping disabled 0x1: VC-1 range mapping enabled	RW	0

Table 11-308. DISPC_VID3_ATTRIBUTES2

Address Offset	0x0000 062C		
Physical Address	0x5800 162C	**Instance**	DISPC
Description	The register configures the attributes of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:12	11:9	8	7	6:4	3:1	0
RESERVED	SUBSAMPLINGPATTERN	YUVCHROMARESAMPLING	RESERVED	VC1_RANGE_CBCR	VC1_RANGE_Y	VC1ENABLE

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:9	SUBSAMPLINGPATTERN	Subsampling pattern setting.	RW	0x0
8	YUVCHROMARE SAMPLING	The YUV chrominance can be resampled using averaging of the adjacent chrominance samples, without using the polyphase filter for 4:2:2 input or can be calculated using the polyphase filter for 4:2:2/4:2:0. The polyphase filter is mandatory for the 4:2:0 format. This bit controls the order in which the processing is done on the video pipe. 0x0: When input is in 4:2:2, the missing chrominance samples are calculated by averaging the adjacent samples if DISPC_VID1_ATTRIBUTES. ROTATION=0 only. Other rotation configurations are not supported. 0x1: For 4:2:2 (or 4:2:0), the missing chrominance samples are calculated by filtering the adjacent samples (5-tap polyphase filter). See Figure 11-49, *Configuration 2: Video Pipeline*. All rotation configurations are supported.	RW	0
7	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
6:4	VC1_RANGE_ CBCR	Defines the VC-1 range value for the CbCr component from 0 to 7.	RW	0x0
3:1	VC1_RANGE_Y	Defines the VC-1 range value for the Y component from 0 to 7.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VC1ENABLE	Enable/disable the VC-1 range mapping processing. The bit field is ignored if the format is not one of the supported YUV formats. 0x0: VC-1 range mapping disabled 0x1: VC-1 range mapping enabled	RW	0

Table 11-309. DISPC_GAMMA_TABLE0

Address Offset	0x0000 0630		
Physical Address	0x5800 1630	**Instance**	DISPC
Description	The register configures the look up table used as color look up table for BITMAP formats (1-, 2-, 4, and 8-bpp) on the graphics pipeline or as gamma table on the primary LCD output. **NOTE: CLUT and BITMAP formats are not supported in this family of devices.**		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDEX								VALUE_R								VALUE_G								VALUE_B							

Bits	Field Name	Description	Type	Reset
31:24	INDEX	Defines the location in the table where the bit field VALUE is stored.	W	0x00
23:16	VALUE_R	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX	W	0x00
15:8	VALUE_G	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX	W	0x00
7:0	VALUE_B	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX	W	0x00

Table 11-310. DISPC_GAMMA_TABLE1

Address Offset	0x0000 0634		
Physical Address	0x5800 1634	**Instance**	DISPC
Description	The register configures the gamma table on the secondary LCD output.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDEX								VALUE_R								VALUE_G								VALUE_B							

Bits	Field Name	Description	Type	Reset
31:24	INDEX	Defines the location in the table where the bit field VALUE is stored.	W	0x00
23:16	VALUE_R	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX	W	0x00
15:8	VALUE_G	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX	W	0x00
7:0	VALUE_B	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX	W	0x00

Table 11-311. DISPC_GAMMA_TABLE2

Address Offset	0x0000 0638		
Physical Address	0x5800 1638	**Instance**	DISPC
Description	The register configures the gamma table on the TV output.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IN DE X	RE SE RV ED	VALUE_R	VALUE_G	VALUE_B

Bits	Field Name	Description	Type	Reset
31	INDEX	Setting this bit to 1 resets the internal index counter to zero. Each subsequent access to the register (with the INDEX bit kept at 0) increments the address for the next storage location into the table memory.	W	0
30	RESERVED		W	0
29:20	VALUE_R	10-bit color component value to store in the table	W	0x000
19:10	VALUE_G	10-bit color component value to store in the table	W	0x000
9:0	VALUE_B	10-bit color component value to store in the table	W	0x000

Table 11-312. DISPC_VID1_FIR2

Address Offset	0x0000 063C		
Physical Address	0x5800 163C	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 1. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31 30 29	28 ... 16	15 14 13	12 ... 0
RESERVED	FIRVINC	RESERVED	FIRHINC

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-313. DISPC_VID1_ACCU2_j

Address Offset	0x0000 0640 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1640 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 1 (DISPC_VID1_ACCU2_0 and DISPC_VID1_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		

Table 11-313. DISPC_VID1_ACCU2_j (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					VERTICALACCU											RESERVED					HORIZONTALACCU										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from –1024 to 1023).	RW	0x000

Table 11-314. DISPC_VID1_FIR_COEF_H2_i

Address Offset	0x0000 0648 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1648 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-315. DISPC_VID1_FIR_COEF_HV2_i

Address Offset	0x0000 064C + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 164C + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		

Table 11-315. DISPC_VID1_FIR_COEF_HV2_i (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-316. DISPC_VID1_FIR_COEF_V2_i

Address Offset	0x0000 0688 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1688 + (0x4 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FIRVC22								FIRVC00							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-317. DISPC_VID2_FIR2

Address Offset	0x0000 06A8		
Physical Address	0x5800 16A8	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 2. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FIRVINC	RESERVED	FIRHINC

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-318. DISPC_VID2_ACCU2_j

Address Offset	0x0000 06AC + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 16AC + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 2 (DISPC_VID2_ACCU2_0 and DISPC_VID2_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					VERTICALACCU											RESERVED					HORIZONTALACCU										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from –1024 to 1023).	RW	0x000

Table 11-319. DISPC_VID2_FIR_COEF_H2_i

Address Offset	0x0000 06B4 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 16B4 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FIRHC3	FIRHC2	FIRHC1	FIRHC0

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-320. DISPC_VID2_FIR_COEF_HV2_i

Address Offset	0x0000 06B8 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 16B8 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-321. DISPC_VID2_FIR_COEF_V2_i

Address Offset	0x0000 06F4 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 16F4 + (0x4 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FIRVC22	FIRVC00

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-322. DISPC_VID3_FIR2

Address Offset	0x0000 0724		
Physical Address	0x5800 1724	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 3. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:29	28:16	15:13	12:0
RESERVED	FIRVINC	RESERVED	FIRHINC

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-323. DISPC_VID3_ACCU2_j

Address Offset	0x0000 0728 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1728 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 3 (DISPC_VID3_ACCU2_0 and DISPC_VID3_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	VERTICALACCU	RESERVED	HORIZONTALACCU

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from –1024 to 1023).	RW	0x000

Table 11-324. DISPC_VID3_FIR_COEF_H2_i

Address Offset	0x0000 0730 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1730 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-325. DISPC_VID3_FIR_COEF_HV2_i

Address Offset	0x0000 0734 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1734 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-326. DISPC_VID3_FIR_COEF_V2_i

Address Offset	0x0000 0770 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 1770 + (0x4 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or VFP start period of the third LCD or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FIRVC22								FIRVC00							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-327. DISPC_WB_FIR2

Address Offset	0x0000 0790		
Physical Address	0x5800 1790	**Instance**	DISPC
Description	The register configures the resize factors for horizontal and vertical up/downsampling of the write-back pipeline. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FIRVINC	RESERVED	FIRHINC

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Table 11-328. DISPC_WB_ACCU2_j

Address Offset	0x0000 0794 + (0x4 * j)	**Index**	j = 0 to 1
Physical Address	0x5800 1794 + (0x4 * j)	**Instance**	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the write back pipeline (DISPC_WB_ACCU2_0 and DISPC_WB_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	VERTICALACCU	RESERVED	HORIZONTALACCU

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from –1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from –1024 to 1023).	RW	0x000

Table 11-329. DISPC_WB_FIR_COEF_H2_i

Address Offset	0x0000 07A0 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 17A0 + (0x8 * i)	**Instance**	DISPC

Table 11-329. DISPC_WB_FIR_COEF_H2_i (continued)

Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-330. DISPC_WB_FIR_COEF_HV2_i

Address Offset	0x0000 07A4 + (0x8 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 17A4 + (0x8 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00

Bits	Field Name	Description	Type	Reset
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

Table 11-331. DISPC_WB_FIR_COEF_V2_i

Address Offset	0x0000 07E0 + (0x4 * i)	**Index**	i = 0 to 7
Physical Address	0x5800 17E0 + (0x4 * i)	**Instance**	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31–16	15–8	7–0
RESERVED	FIRVC22	FIRVC00

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

Table 11-332. DISPC_GLOBAL_BUFFER

Address Offset	0x0000 0800		
Physical Address	0x5800 1800	**Instance**	DISPC
Description	The register configures the DMA buffers allocations to the pipeline (graphics, video1, video2, video3 and write-back). Both TOP and BOTTOM must be allocated to the same pipeline.		
Type	RW		

31–30	29–24	23–18	17–12	11–6	5–0
RESERVED	WB_BUFFER	VID3_BUFFER	VID2_BUFFER	VID1_BUFFER	GFX_BUFFER

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Bits	Field Name	Description	Type	Reset
29:24	WB_BUFFER	Write-back DMA buffer allocation to one of the pipelines. By default to write-back pipeline. 0x24: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x9: DMA buffer allocated to the video1 pipeline. 0x12: DMA buffer allocated to the vdieo2 pipeline. 0x1B: DMA buffer allocated to the vdieo3 pipeline.	RW	0x24
23:18	VID3_BUFFER	Video3 DMA buffer allocation to one of the pipelines. By default to video3 pipeline. 0x24: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x9: DMA buffer allocated to the video1 pipeline. 0x12: DMA buffer allocated to the vdieo2 pipeline. 0x1B: DMA buffer allocated to the vdieo3 pipeline.	RW	0x1B
17:12	VID2_BUFFER	Video2 DMA buffer allocation to one of the pipelines. By default to video2 pipeline. 0x24: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x9: DMA buffer allocated to the video1 pipeline. 0x12: DMA buffer allocated to the vdieo2 pipeline. 0x1B: DMA buffer allocated to the vdieo3 pipeline.	RW	0x12
11:6	VID1_BUFFER	Video1 DMA buffer allocation to one of the pipelines. By default to video 1 pipeline. 0x24: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x9: DMA buffer allocated to the video1 pipeline. 0x12: DMA buffer allocated to the vdieo2 pipeline. 0x1B: DMA buffer allocated to the vdieo3 pipeline.	RW	0x09
5:0	GFX_BUFFER	Graphics DMA buffer allocation to one of the pipelines. By default to graphics pipeline. 0x24: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x9: DMA buffer allocated to the video1 pipeline. 0x12: DMA buffer allocated to the vdieo2 pipeline. 0x1B: DMA buffer allocated to the vdieo3 pipeline.	RW	0x00

Table 11-333. DISPC_DIVISOR

Address Offset	0x0000 0804		
Physical Address	0x5800 1804	**Instance**	DISPC
Description	The register configures the divisor value for generating the core functional clock. There is a backward compatibility mode enabled by default in order to use DISPC_DIVISOR1.LCD value instead of DISPC_DIVISOR.LCD bit field for generating the core functional clock.		
Type	RW		

31:24	23:16	15:1	0
RESERVED	LCD	RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
23:16	LCD	Display Controller Logic Clock Divisor Value (from 1 to 255) to specify the frequency of the Display Controller logic clock based on the function clock. The value 0 is invalid.	RW	0x4
15:1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
0	ENABLE	When the bit field is set to 1, the bit field LCD is used to generated the core functional clock from the input clock. When the bit field is set to 0, the value DISPC_DIVISOR1.LCD is used instead. 0x0: DISPC_DIVISOR1.LCD bit field is used 0x1: DISPC_DIVISOR.LCD bit field is used	RW	0

Table 11-334. DISPC_WB_ATTRIBUTES2

Address Offset	0x0000 0810		
Physical Address	0x5800 1810	**Instance**	DISPC
Description	The register set the counter to control the delay to flush the WB pipe after the end of the frame in capture mode. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								WBDELAYCOUNT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	WBDELAYCOUNT	Delays the WB pipe flush after the end of the frame. Delay = n (number of lines), where n = 0:255. If n = 0, the WB is re-initialized just at the end of the last line of a frame at the beginning of the VFP signal. If n = 1:255, the write buffers DMA are flushed n lines later.	RW	0x00

Table 11-335. DISPC_DEFAULT_COLOR3

Address Offset	0x0000 0814		
Physical Address	0x5800 1814	**Instance**	DISPC
Description	The control register allows to configure the default solid background color for the third LCD. Shadow register, updated on VFP start period of third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEFAULTCOLOR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	DEFAULTCOLOR	24-bit RGB color value to specify the default solid color to display when there is no data from the overlays	RW	0x00 0000

Table 11-336. DISPC_TRANS_COLOR3

Address Offset	0x0000 0818		
Physical Address	0x5800 1818	**Instance**	DISPC
Description	The register sets the transparency color value for the video/graphics overlays for the third LCD output. Shadow register, updated on VFP start period of the third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TRANSCOLORKEY																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	TRANSCOLORKEY	Transparency color key value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24 **NOTE: CLUT and BITMAP formats are not supported in this family of devices.**	RW	0x00 0000

Table 11-337. DISPC_CPR3_COEF_B

Address Offset	0x0000 081C		
Physical Address	0x5800 181C	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the blue component. It is used for the secondary LCD output. Shadow register, updated on VFP start period of third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR										RESERVED	BG										RESERVED	BB									

Bits	Field Name	Description	Type	Reset
31:22	BR	BR coefficient Encoded signed value (from –512 to 511)	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	BG	BG coefficient Encoded signed value (from –512 to 511)	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	BB	BB coefficient Encoded signed value (from –512 to 511)	RW	0x000

Table 11-338. DISPC_CPR3_COEF_G

Address Offset	0x0000 0820		
Physical Address	0x5800 1820	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the green component. It is used for the secondary LCD output. Shadow register, updated on VFP start period of third LCD		

Table 11-338. DISPC_CPR3_COEF_G (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GR										RESERVED	GG										RESERVED	GB									

Bits	Field Name	Description	Type	Reset
31:22	GR	GRcoefficient Encoded signed value (from –512 to 511)	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	GG	GG coefficient Encoded signed value (from –512 to 511)	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	GB	GB coefficient Encoded signed value (from –512 to 511)	RW	0x000

Table 11-339. DISPC_CPR3_COEF_R

Address Offset	0x0000 0824		
Physical Address	0x5800 1824	**Instance**	DISPC
Description	The register configures the color phase rotation matrix coefficients for the red component. Shadow register, updated on VFP start period of third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RR										RESERVED	RG										RESERVED	RB									

Bits	Field Name	Description	Type	Reset
31:22	RR	RR coefficient Encoded signed value (from –512 to 511)	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	RG	RG coefficient Encoded signed value (from –512 to 511)	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	RB	RB coefficient Encoded signed value (from –512 to 511)	RW	0x000

Table 11-340. DISPC_DATA3_CYCLE1

Address Offset	0x0000 0828		
Physical Address	0x5800 1828	**Instance**	DISPC
Description	The control register configures the output data format for the first cycle. Shadow register, updated on VFP start period of third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	BITALIGNMENTPIXEL2	RESERVED	NBBITSPIXEL2	RESERVED	BITALIGNMENTPIXEL1	RESERVED	NBBITSPIXEL1

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-341. DISPC_DATA3_CYCLE2

Address Offset	0x0000 082C		
Physical Address	0x5800 182C	**Instance**	DISPC
Description	The control register configures the output data format for the second cycle. Shadow register, updated on VFP start period of third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BITALIGNMENTPIXEL2				RESERVED			NBBITSPIXEL2					RESERVED				BITALIGNMENTPIXEL1				RESERVED			NBBITSPIXEL1				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-342. DISPC_DATA3_CYCLE3

Address Offset	0x0000 0830

Table 11-342. DISPC_DATA3_CYCLE3 (continued)

Physical Address	0x5800 1830	**Instance**	DISPC
Description	The control register configures the output data format for the third cycle. Shadow register, updated on VFP start period of third LCD		
Type	RW		

31:28	27:24	23:21	20:16	15:12	11:8	7:5	4:0
RESERVED	BITALIGNMENTPIXEL2	RESERVED	NBBITSPIXEL2	RESERVED	BITALIGNMENTPIXEL1	RESERVED	NBBITSPIXEL1

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

Table 11-343. DISPC_SIZE_LCD3

Address Offset	0x0000 0834		
Physical Address	0x5800 1834	**Instance**	DISPC
Description	The register configures the panel size (horizontal and vertical). It is used for the third LCD output. Shadow register, updated on VFP start period of the third LCD. A delta value is used to indicate if the odd field is the same vertical size as the even field or ± one line.		
Type	RW		

31:28	27:16	15:14	13:12	11:0
RESERVED	LPP	DELTA_LPP	RESERVED	PPL

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	LPP	Lines per panel Encoded value (from 1 to 4096) to specify the number of lines per panel (program to value minus 1).	RW	0x000
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: Same size 0x1: Odd size = even size +1 0x2: Odd size = even size –1	RW	0x0
13:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11:0	PPL	Pixels per line Encoded value (from 1 to 4096) to specify the number of pixels contained within each line on the display (program to value minus 1). In STALL mode, any value is valid. In non-STALL mode, only values of multiples of 8 pixels are valiid.	RW	0x000

Table 11-344. DISPC_DIVISOR3

Address Offset	0x0000 0838		
Physical Address	0x5800 1838	**Instance**	DISPC
Description	The register configures the divisors. It is used for the third LCD output. Shadow register, updated on VFP start period of the third LCD		
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	LCD	RESERVED	PCD

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:16	LCD	Display controller logic clock divisor Value (from 1 to 255) to specify the intermediate pixel clock frequency based on LCD2_CLK. The value 0 is invalid.	RW	0x04
15:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
7:0	PCD	Pixel clock divisor Value (from 1 to 255) to specify the frequency of the pixel clock based on LCD2_CLK divided by the value of DISPC_DIVISOR2.LCD. The value 0 is invalid.	RW	0x01

Table 11-345. DISPC_POL_FREQ3

Address Offset	0x0000 083C		
Physical Address	0x5800 183C	**Instance**	DISPC
Description	The register configures the signal configuration. It is used for the third LCD output. Shadow register, updated on VFP start period of the third LCD		
Type	RW		

31–19	18	17	16	15	14	13	12	11–8	7–0
RESERVED	ALIGN	ONOFF	RF	IEO	IPC	IHS	IVS	ACBI	ACB

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
18	ALIGN	Defines the alignment betwwen HSYNC and VSYNC assertion 0x0: VSYNC and HSYNC are not aligned. 0x1: VSYNC and HSYNC assertions are aligned.	RW	0

Bits	Field Name	Description	Type	Reset
17	ONOFF	HSYNC/VSYNC pixel clock control on/off 0x0: HSYNC and VSYNC are driven on opposite edges of the pixel clock than pixel data. 0x1: HSYNC and VSYNC are driven according to bit 16. **Note:** Control module register CTRL_CORE_SMA_SW_1[24] DSS_CH2_ON_OFF must be set to match	RW	0
16	RF	Program HSYNC/VSYNC rise or fall 0x0: HSYNC and VSYNC are driven on the falling edge of the pixel clock (if bit 17 is set to 1). 0x1: HSYNC and VSYNC are driven onthe rising edge of the pixel clock (if bit 17 is set to 1). **Note:** Control module register CTRL_CORE_SMA_SW_1[18] DSS_CH2_RF must be set to match	RW	0
15	IEO	Invert output enable 0x0: Ac-bias is active high (active display mode). 0x1: Ac-bias is active low (active display mode).	RW	0
14	IPC	Invert pixel clock 0x0: Data is driven on the LCD data lines on the rising edge of the pixel clock. 0x1: Data is driven on the LCD data lines on the falling edge of the pixel clock. **Note:** Control module register CTRL_CORE_SMA_SW_1[21] DSS_CH2_IPC must be set to match	RW	0
13	IHS	Invert HSYNC 0x0: Line clock pin is active high and inactive low. 0x1: Line clock pin is active low and inactive high.	RW	0
12	IVS	Invert VSYNC 0x0: Frame clock pin is active high and inactive low. 0x1: Frame clock pin is active low and inactive high.	RW	0
11:8	ACBI	AC bias pin transitions per interrupt Value (from 0 to 15) used to specify the number of AC bias pin transitions	RW	0x0
7:0	ACB	AC bias pin frequency Value (from 0 to 255) used to specify the number of line clocks to count before transitioning the AC bias pin. This pin is used to periodically invert the polarity of the power supply to prevent DC charge buildup within the display.	RW	0x00

Table 11-346. DISPC_TIMING_H3

Address Offset	0x0000 0840		
Physical Address	0x5800 1840	**Instance**	DISPC
Description	The register configures the timing logic for the HSYNC signal. It is used for the third LCD output. Shadow register, updated on VFP start period of the third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBP												HFP												HSW							

Bits	Field Name	Description	Type	Reset
31:20	HBP	Horizontal back porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (program to value minus 1).	RW	0x000
19:8	HFP	Horizontal front porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the end of a line transmission before the line clock is asserted (program to value minus 1).	RW	0x000
7:0	HSW	Horizontal synchronization pulse width. Encoded value (from 1 to 256) to specify the number of pixel clock periods to pulse the line clock at the end of each line (program to value minus 1).	RW	0x00

Table 11-347. DISPC_TIMING_V3

Address Offset	0x0000 0844		
Physical Address	0x5800 1844	**Instance**	DISPC
Description	The register configures the timing logic for the VSYNC signal. It is used for the third LCD output. Shadow register, updated on VFP start period of the third LCD		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBP												VFP												VSW							

Bits	Field Name	Description	Type	Reset
31:20	VBP	Vertical back porch. Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the beginning of a frame before the first set of pixels is output to the display	RW	0x000
19:8	VFP	Vertical front porch. Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the end of each frame	RW	0x000
7:0	VSW	Vertical synchronization pulse width. In active mode, encoded value (from 1 to 256) to specify the number of line clock periods (program to value minus 1) to pulse the frame clock (VSYNC) pin at the end of each frame after the end of frame wait (VFP) period elapses. Frame clock uses as VSYNC signal in active mode.	RW	0x00

Table 11-348. DISPC_CONTROL3

Address Offset	0x0000 0848		
Physical Address	0x5800 1848	**Instance**	DISPC
Description	The control register configures the display controller module for the third LCD output.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPATIALTEMPORALDITHERINGFRAMES	RESERVED	TDMUNUSEDBITS	TDMCYCLEFORMAT	TDMPARALLELMODE	TDMENABLE	RESERVED	RESERVED	OVERLAYOPTIMIZATION	STALLMODE	RESERVED	TFTDATALINES	STDITHERENABLE	RESERVED	GOLCD	M8B	STNTFT	MONOCOLOR	RESERVED	LCDENABLE

Bits	Field Name	Description	Type	Reset
31:30	SPATIALTEMPORALDITHERINGFRAMES	Spatial/temporal dithering number of frames for the third LCD output wr: VFP start period of the third LCD output 0x0: Spatial only 0x1: Spatial and temporal over two frames 0x3: Reserved 0x2: Spatial and temporal over four frames	RW	0x0
29:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
26:25	TDMUNUSEDBITS	State of unused bits (TDM mode only) for the third LCD output wr: VFP start period of the third LCD output 0x0: Low level (0) 0x1: High level (1) 0x3: Reserved 0x2: Unchanged from previous state	RW	0x0
24:23	TDMCYCLEFORMAT	Cycle format (TDM mode only) for the third LCD output wr: VFP start period of third LCD output 0x0: One cycle for 1 pixel 0x1: Two cycles for 1 pixel 0x3:Three cycles for 2 pixels 0x2: Three cycles for 1 pixel	RW	0x0
22:21	TDMPARALLELMODE	Output interface width (TDM mode only) for the third LCD output wr: VFP start period of the third LCD output 0x0: 8-bit parallel output interface selected 0x1: 9-bit parallel output interface selected 0x3: 16-bit parallel output interface selected 0x2: 12-bit parallel output interface selected	RW	0x0
20	TDMENABLE	Enable the multiple cycle format for the third LCD output wr: VFP start period of third LCD output 0x0: TDM disabled 0x1: TDM enabled	RW	0
19:14	RESERVED	Write 0s for future compatibility. Reads return 0.	RW	0x00
13	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
12	OVERLAYOPTIMIZATION	Overlay optimization for the third LCD output wr: VFP or EVSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, third LCD, TV output, or write-back to the memory. 0x0: All the data for all the enabled pipelines are fetched from memory regardless of the overlay/alpha blending configuration. 0x1: The data not used by the overlay manager because of overlap between layers with no alpha blending between them must not be fetched from memory to optimize the bandwidth.	RW	0
11	STALLMODE	STALL mode for the third LCD output wr: VFP start period of the third LCD output 0x0: Normal mode selected 0x1: STALL mode selected. The display controller sends the data without considering the VSYNC/HSYNC. The LCD output is disabled at the end of the transfer of the frame. Software must reenable the LCD output to generate a new frame.	RW	0
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:8	TFTDATALINES	Number of lines of the third LCD interface wr: VFP start period of the third LCD output 0x0: 12-bit output aligned on the LSB of the pixel data interface 0x1: 16-bit output aligned on the LSB of the pixel data interface 0x3: 24-bit output aligned on the LSB of the pixel data interface 0x2: 18-bit output aligned on the LSB of the pixel data interface	RW	0x0
7	STDITHERENABLE	Spatial temporal dithering enable for the third LCD output wr: VFP start period of the third LCD output 0x0: Spatial/temporal dithering logic disabled 0x1: Spatial/temporal dithering logic enabled	RW	0
6	RESERVED	Reserved	R	0
5	GOLCD	GO command for the third LCD output. It is used to synchronized the pipelines (graphics and/or video) associated with the third LCD output. wr: Immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the LCD output using the user values. The hardware resets the bit when the update is complete. 0x1: The user has finished programming the shadow registers of the pipeline(s) associated with the LCD output, and the hardware can update the internal registers at the VFP start period.	RW	0
4	M8B	Mono 8-bit mode of the third LCD wr: VFP start period of the third LCD output 0x0: Reserved 0x1: Reserved	RW	0

Bits	Field Name	Description	Type	Reset
3	STNTFT	LCD Display type of the third LCD wr: VFP start period of the third LCD output 0x0: Reserved 0x1: Active matrix display operation enabled	RW	0
2	MONOCOLOR	Monochrome/color selection for the third LCD wr: VFP start period of the third LCD output 0x0: Reserved 0x1: Reserved	RW	0
1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
0	LCDENABLE	Enable the third LCD output wr: Immediate 0x0: LCD output disabled (at the end of the frame when the bit is reset) 0x1: LCD output enabled	RW	0

Table 11-349. DISPC_CONFIG3

Address Offset	0x0000 084C		
Physical Address	0x5800 184C	**Instance**	DISPC
Description	The control register configures the display controller module for the third LCD output. Shadow register, updated on VFP start period of the third LCD or EVSYNC		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">TLCDINTERLEAVE</td><td>FULLRANGE</td><td>COLORCONVENABLE</td><td>FIDFIRST</td><td>OUTPUTMODEENABLE</td><td>BT1120ENABLE</td><td>BT656ENABLE</td><td colspan="4">RESERVED</td><td>CPR</td><td colspan="3">RESERVED</td><td>TCKLCDSELECTION</td><td>TCKLCDENABLE</td><td>RESERVED</td><td>ACBIASGATED</td><td>VSYNCGATED</td><td>HSYNCGATED</td><td>PIXELCLOCKGATED</td><td>PIXELDATAGATED</td><td colspan="3">RESERVED</td><td>PIXELGATED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	R	0x0
27:26	TLCDINTERLEAVE	tLCD interleave Pattern	RW	0x0
25	FULLRANGE	Color space conversion full range setting 0x0: Limited range selected 0x1: Full range selected	RW	0
24	COLORCONVENABLE	Enable the color space conversion. It must be reset when the CPR bit field is set to 0x1. 0x0: Disable color space conversion RGB to YUV. 0x1: Enable color space conversion RGB to YUV.	RW	0
23	FIDFIRST	Selects the first field to output in case of interlace mode. In case of progressive mode, the value is not used. 0x0: First field is even. 0x1: Odd field is first.	RW	0
22	OUTPUTMODEENABLE	Selects between progressive and interlace mode for the third LCD output 0x0: Progressive mode selected 0x1: Interlace mode selected	RW	0

Bits	Field Name	Description	Type	Reset
21	BT1120ENABLE	Selects BT.1120 format on the third LCD output. It is not posssible to enable BT.656 and BT.1120 at the same time on the same LCD output. 0x0: BT.1120 is disabled. 0x1: BT.1120 is enabled.	RW	0
20	BT656ENABLE	Selects BT.656 format on the third LCD output. It is not posssible to enable BT.656 and BT.1120 at the same time on the same LCD output. 0x0: BT.656 is disabled. 0x1: BT.656 is enabled.	RW	0
19:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
15	CPR	Color phase rotation control (third LCD output). It must be reset when the ColorConvEnable bit field is set to 1. wr: VFP start period of the third LCD output 0x0: Color phase rotation disabled 0x1: Color phase rotation enabled	RW	0
14:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11	TCKLCDSELECTION	Transparency color key selection (third LCD output) wr: VFP start period of the third LCD output 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0
10	TCKLCDENABLE	Transparency color key enabled (third LCD output) wr: VFP start period of the third LCD output 0x0: Disable the transparency color key for the LCD 0x1: Enable the transparency color key for the LCD	RW	0
9	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
8	ACBIASGATED	ACBias gated enabled (third LCD output) wr: VFP start period of the third LCD output 0x0: AcBias gated disabled 0x1: AcBias gated enabled	RW	0
7	VSYNCGATED	VSYNC gated enabled (third LCD output) wr: VFP start period of the third LCD output 0x0: VSYNC gated disabled 0x1: VSYNC gated enabled	RW	0
6	HSYNCGATED	HSYNC gated enabled (third LCD output) wr: VFP start period of the third LCD output 0x0: HSYNC gated disabled 0x1: HSYNC gated enabled	RW	0
5	PIXELCLOCKGATED	Pixel clock gated enabled (third LCD output) wr: VFP start period of the third LCD output 0x0: Pixel clock gated disabled 0x1: Pixel clock gated enabled	RW	0
4	PIXELDATAGATED	Pixel data gated enabled (third LCD output) wr: VFP start period of the third LCD output 0x0: Pixel data gated disabled 0x1: Pixel data gated enabled	RW	0
3:1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0

Bits	Field Name	Description	Type	Reset
0	PIXELGATED	Pixel gated enable (only for TFT) (third LCD output) wr: VFP start period of the third LCD output 0x0: Pixel clock always toggles (only in TFT mode). 0x1: Pixel clock toggles only when there is valid data to display (only in TFT mode).	RW	0

Table 11-350. DISPC_GAMMA_TABLE3

Address Offset	0x0000 0850		
Physical Address	0x5800 1850	**Instance**	DISPC
Description	The register configures the gamma table on the third LCD output.		
Type	W		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
INDEX	VALUE_R	VALUE_G	VALUE_B

Bits	Field Name	Description	Type	Reset
31:24	INDEX	Defines the location in the table where the VALUE bit field is stored.	W	0x00
23:16	VALUE_R	8-bit value used to define the value to store at the location in the table defined by the INDEX bit field	W	0x00
15:8	VALUE_G	8-bit value used to define the value to store at the location in the table defined by the INDEX bit field	W	0x00
7:0	VALUE_B	8-bit value used to define the value to store at the location in the table defined by the INDEX bit field	W	0x00

Table 11-351. DISPC_BA0_FLIPIMMEDIATE_EN

Address Offset	0x0000 0854		
Physical Address	0x5800 1854	**Instance**	DISPC
Description	This register enables the flip immediate.		
Type	RW		

31 … 4	3	2	1	0
RESERVED	VID3	VID2	VID1	GFX

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved.	R	0x000 0000
3	VID3	Enable flip immediate for video3 pipeline	RW	0x0
2	VID2	Enable flip immediate for video2 pipeline	RW	0x0
1	VID1	Enable flip immediate for video1 pipeline	RW	0x0
0	GFX	Enable flip immediate for gfx pipeline	RW	0x0

Table 11-352. DISABLE_MSTANDBY_ENHANCEMENT

Address Offset	0x0000 0858		
Physical Address	0x5800 1858	**Instance**	DISPC
Description	This register disables the DISPC DMA Mstandby behavior enhancement.		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0

RESERVED	DISABLE_MSTANDBY_ENHANCEMENT

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved.	R	0x0
0	DISABLE_MSTANDBY_ENHANCEMENT	0: DISPC DMA Mstandby behavior enhancement is enabled. 1: Disable DISPC DMA Mstandby behavior enhancement. This is the recommended setting.	RW	0x0

Table 11-353. DISPC_GLOBAL_MFLAG_ATTRIBUTE

Address Offset	0x0000 085C		
Physical Address	0x5800 185C	**Instance**	DISPC
Description	Global MFLAG atrribute control register.		
Type	RW		

31:3	2	1:0
RESERVED	MFLAG_START	MFLAG_CTRL

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved.	R	0x0000 0000
2	MFLAG_START	MFLAG Start 0x0: When the DMA buffer is empty at the beginning of the frame, MFLAG signal of each pipeline is kept at 0 until PRELOAD is reached, then based on MFLAG_CTRL bitfield MFLAG is generated and internal logic is arbitrating between pipeline requests 0x1: Even at the beginning of the frame when the DMA buffer is empty, MFLAG_CTRL bitfield is used to determine how MFLAG signal for each pipeline shall be driven.	RW	0x0
1:0	MFLAG_CTRL	MFLAG control 0x0: MFLAG mechanism is disabled: MFLAG out of band signal is set to 0 0x1: MFLAG mechanism is enabled: MFLAG out of band signal is always set to 1 (force mode for debug) 0x2: MFLAG mechanism is enabled and MFLAG out of band signal is dynamically set and reset depending on MFLAG rules.	RW	0x0

Table 11-354. DISPC_GFX_MFLAG_THRESHOLD

Address Offset	0x0000 0860		
Physical Address	0x5800 1860	**Instance**	DISPC
Description	MFLAG thresholds for graphics pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or external VSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory.		
Type	RW		

31:16	15:0
HT_MFLAG	LT_MFLAG

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG	High Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches HT_MFLAG level, MFLAG is reset to 0	RW	0x0000
15:0	LT_MFLAG	Low Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches LT_MFLAG level, MFLAG is set to 1	RW	0x0000

Table 11-355. DISPC_VID1_MFLAG_THRESHOLD

Address Offset	0x0000 0864		
Physical Address	0x5800 1864	**Instance**	DISPC
Description	MFLAG thresholds for video1 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or external VSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory.		
Type	RW		

31:16	15:0
HT_MFLAG	LT_MFLAG

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG	High Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches HT_MFLAG level, MFLAG is reset to 0	RW	0x0000
15:0	LT_MFLAG	Low Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches LT_MFLAG level, MFLAG is set to 1	RW	0x0000

Table 11-356. DISPC_VID2_MFLAG_THRESHOLD

Address Offset	0x0000 0868		
Physical Address	0x5800 1868	**Instance**	DISPC
Description	MFLAG thresholds for video2 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or external VSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory.		
Type	RW		

31:16	15:0
HT_MFLAG	LT_MFLAG

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG	High Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches HT_MFLAG level, MFLAG is reset to 0	RW	0x0000
15:0	LT_MFLAG	Low Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches LT_MFLAG level, MFLAG is set to 1	RW	0x0000

Table 11-357. DISPC_VID3_MFLAG_THRESHOLD

Address Offset	0x0000 086C		
Physical Address	0x5800 186C	**Instance**	DISPC
Description	MFLAG thresholds for video3 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or external VSYNC or when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT_MFLAG																LT_MFLAG															

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG	High Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches HT_MFLAG level, MFLAG is reset to 0	RW	0x0000
15:0	LT_MFLAG	Low Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches LT_MFLAG level, MFLAG is set to 1	RW	0x0000

Table 11-358. DISPC_WB_MFLAG_THRESHOLD

Address Offset	0x0000 0870		
Physical Address	0x5800 1870	**Instance**	DISPC
Description	MFLAG thresholds for write-back pipeline. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline), when the WB pipeline is directly connected to one of the pipelines (graphics or video), combined with the synchronization event of the channel overlay output selected as an input to the WB pipeline (that is, VFP start period of primary LCD, or VFP start period of secondary LCD, or VFP start period of the third LCD, or EVSYNC), for all registers associated with the selected channel out and further delayed by the DISPC_WB_ATTRIBUTES2.WBDELAYCOUNT bit-field, for all registers of the Write back and DMA. In WB capture mode, both DISPC_CONTROL2.GOWB and DISPC_CONTROL#.GOLCD/TV corresponding to the selected output channel shall be set. It is not required to set the GOWB bit when WB memory-to-memory mode is used.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT_MFLAG																LT_MFLAG															

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG	High Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches HT_MFLAG level, MFLAG is reset to 0	RW	0x0000
15:0	LT_MFLAG	Low Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches LT_MFLAG level, MFLAG is set to 1	RW	0x0000

11.3 High-Definition Multimedia Interface

This section introduces the high-definition multimedia interface (HDMI) module and describes its main functions and connections in the device.

Note

All the HDMI features described in this chapter may not be supported in software. For example, 3D-frame packing is not supported. Refer to the software development kit (SDK) for which HDMI features are supported.

11.3.1 HDMI Overview

Figure 11-103 shows an overview of the HDMI module.

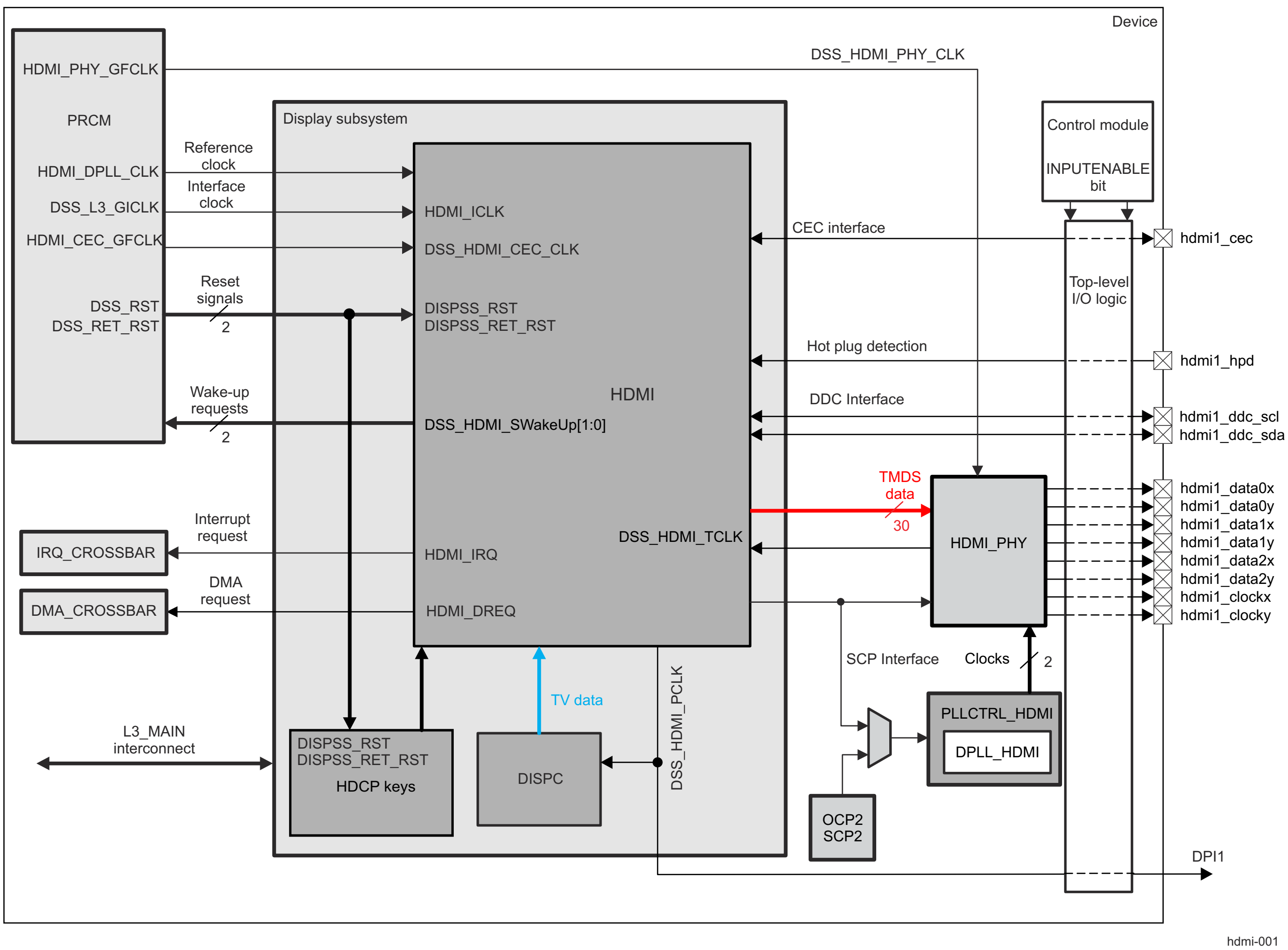


Figure 11-103. HDMI Overview

11.3.1.1 HDMI Main Features

The HDMI module provides the following main features:

- HDMI 1.4a, HDCP 1.4, and DVI 1.0 compliant
  - Includes support for the 3D stereoscopic frame-packing formats of the HDMI v1.4a standard (1080p24Hz, 720p50Hz, 720p60Hz + side-by-side half structure: 1080p60Hz)
- EIA/CEA-861-D video format support (See Table 11-359 for more information.)
- VESA® display monitor timing (DMT) video format support(See Table 11-360 for more information.)

- Support for deep-color mode:
  - 10-bit color component deep-color modes up to 1080p @ 60 Hz
  - 12-bit color component deep-color modes up to 720p/1080i @ 60 Hz
  - 16-bit/component color depth is not supported
- Support for x.v.Color (x.v.Color data from memory outputted by HDMI. No processing, pass-through mode only)
- Supports up to 148.5-MHz pixel clock with deep-color modes (10-bit or 12-bit)
- Supports up to 186 MHz pixel clock with no deep-color mode (8-bit)
- Video formats: 24-bit RGB input
  - 24/30/36-bit RGB/YCbCr 4:4:4 (deep color) output
  - 16/20/24-bit YCbCr 4:2:2 output
- Uncompressed multichannel (up to eight channels) audio (L-PCM) support.
- Master inter-integrated circuit (I^2C™) interface for display data channel (DDC) connection
- Consumer electronic control (CEC) interface
- Integrated transition minimized differential signaling (TMDS®) and TMDS error reduction coding (TERC4) encoders for data island support
- Integrated TMDS physical layer (PHY) (three TMDS differential data lanes + TMDS differential clock lane)
  - Up to 1.86 Gbps per lane at (1080p @ 60 Hz at 10 bits/component)

11.3.1.2 HDMI Video Formats and Timings

11.3.1.2.1 HDMI CEA-861-D Video Formats and Timings

Table 11-359 presents the video timings supported by the HDMI module, according to the CEA-861-D standard:

Table 11-359. HDMI Video Timings (CEA-861-D)

Refresh Rate	Resolution
24 Hz (Low field rate)	1920 x 1080p
50 Hz	1920 x 1080p
	1920 x 1080i
	2880 x 576p
	2880 x 288p
	1440 x 576p
	1440 x 288p
	1280x720p
	720 x 576p
59.94 Hz	1920 x 1080p
	1920 x 1080i
	1280 x 720p
	2880 x 480p
	2880 x 240p
	1440 x 480p
	1440 x 480i
	1440 x 240p
	720 x 480p
	640 x 480p

Table 11-359. HDMI Video Timings (CEA-861-D) (continued)

Refresh Rate	Resolution
60 Hz	1920 x 1080p 1920 x 1080i 2880 x 480p 2880 x 240p 1280 x 720p 1440 x 480p 1440 x 480i 1440 x 240p 1280 x 720p 720 x 576p 720 x 480p 640 x 480p
200 Hz	720 x 576p
239 Hz	720 x 480p
240 Hz	720 x 480p

11.3.1.2.2 VESA DMT Video Formats and Timings

Table 11-360 lists the video timings supported by the HDMI module and based on the VESA DMT standard:

Table 11-360. HDMI Video timings (VESA DMT)

Refresh Rate	Resolution
60 Hz	640 x 480p 800 x 600p 848 x 480p 1024 x 768p 1280 x 768p
	1280 x 800p
	1280 x 960p
	1280 x 1024p
	1360 x 768p
	1366 x 768p
	1400 x 1050p
	1440 x 900p
	1680 x 1200p
	1680 x 1050p
	1920 x 1080p
60 Hz (reduced blanking)	1280 x 768p
	1280 x 800p
	1400 x 1050p
	1440 x 900p
	1680 x 1050p
	1920 x 1200p

Note

Custom resolutions may work, but are not supported by the device. Using non-standard values can lead to noncompliance with the HDMI interface standard.

Chapter 12
3D Graphics Accelerator

This chapter is a basic overview and describes the integration of the 3D graphics processing unit (GPU) subsystem in the device.

Note

The GPU subsystem is an instantiation by Texas Instruments of the PowerVR® SGX544 core from Imagination Technologies Limited.

This document contains materials that are © Imagination Technologies Limited.

USSE and PowerVR are trademarks or registered trademarks of Imagination Technologies Limited.

12.1 GPU Overview.......... 2902
12.2 GPU Integration.......... 2905
12.3 GPU Functional Description.......... 2907
12.4 GPU Register Manual.......... 2910

12.1 GPU Overview

The 3D graphics processing unit (GPU) accelerates 2-dimensional (2D) and 3-dimensional (3D) graphics and compute applications. It is based on the PowerVR® SGX544-MP2 core from Imagination Technologies Limited. The SGX544-MP2 core is a multicore (dual-core) evolution of the PowerVR SGX544 GPU.

SGX is a new generation of programmable PowerVR graphics and video IP cores. The PowerVR SGX is a scalable architecture which efficiently processes a number of differing multimedia data types concurrently :

- Pixel Data
- Vertex Data
- General Purpose Processing

The dual core GPU splits geometry and pixel rendering among the cores to improve performance proportional to the number of cores.

Figure 12-1 shows the device GPU subsystem.

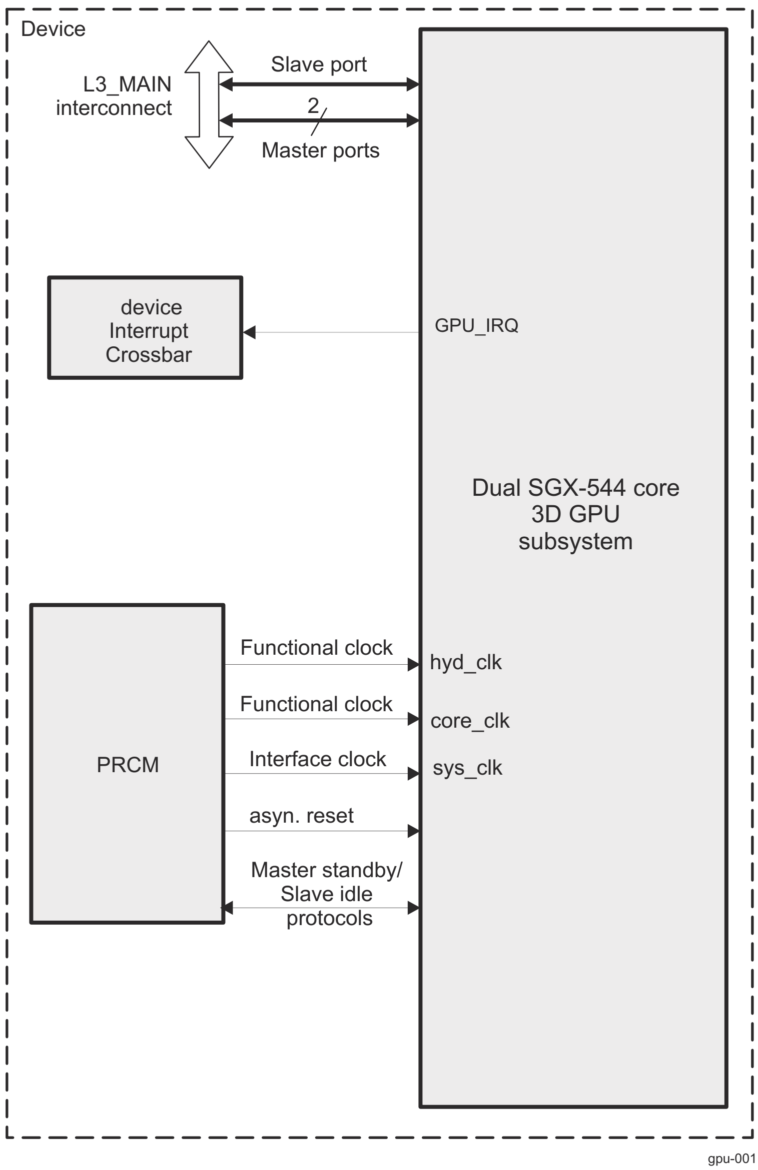


Figure 12-1. GPU Overview

12.1.1 GPU Features Overview

- API support for industry standards:
  - OpenCL™-EP 1.1
  - Direct3D® Feature Level 9.3
- Multicore GPU architecture:
  - 2 × SGX544 cores
  - Shared system level cache of 128 KiB (64 KiB per SGX-544 core)
- Tile-based deferred rendering architecture:
  - Reduces external bandwidth to SDRAM
- Universal Scalable Shader Engine (USSE™):
  - Multithreaded engine incorporating vertex and pixel shader functionality
  - Automatic load balancing of vertex and pixel processing tasks
- Present and texture load accelerator (PTLA):
  - Enables to move, rotate, twiddle, and scale texture surfaces
  - Supports RGB, ARGB, YUV4:2:2, and YUV4:2:0 surface formats
  - Supports bilinear upscale
  - Supports source color key
- Fully virtualized memory addressing for operating system (OS) in a unified memory architecture:
  - Memory management unit (MMU)
  - Up to 4-GiB virtual address space

The 3D-GPU subsystem generates a single (aggregate) interrupt connected to the device Interrupt Crossbar. This allows for this interrupt to be programmatically mapped to multiple device host interrupt controllers (see Section 12.2).

12.1.2 Graphics Feature Overview

- Texture support:
  - Cube map
  - Projected textures
  - Non-square textures
- Texture formats:
  - RGBA 8888, 565, 1555, and 1565
  - Monochromatic 8, 16, 16f, 32f, and 32int
  - Dual channel, 8:8, 16:16, and 16f:16f
  - Compressed textures:
    - PVRTC-i 2 bpp
    - PVRTC-i 4 bpp
    - PVRTC-ii 2 bpp
    - PVRTC-ii 4 bpp
    - ETC1
    - DXT 1-5 and BC 4-5
  - Programmable support for YUV formats:
    - Programmable matrix in hardware, coefficients on 12 bits
    - YUV4:2:2, YUV4:2:0, two planes (NV12 or NV21); YUV4:2:0, three planes
- Resolution support:
  - Frame buffer maximum = 4096 × 4096
  - Texture maximum size = 4096 × 4096
- Texture filtering:
  - Bilinear, trilinear
  - Independent minimum and mag control

- Anti-aliasing:
  - 4× multisampling
  - Programmable sample positions

Note

TI provides the DXT1-5 and BC4-5 texture compression technology for use only with a Microsoft Windows operating system. A separate license is required for the use of this technology (also referred to as S3 texture compression technology) with any other operating system.

12.2 GPU Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 12-2 shows the GPU integration.

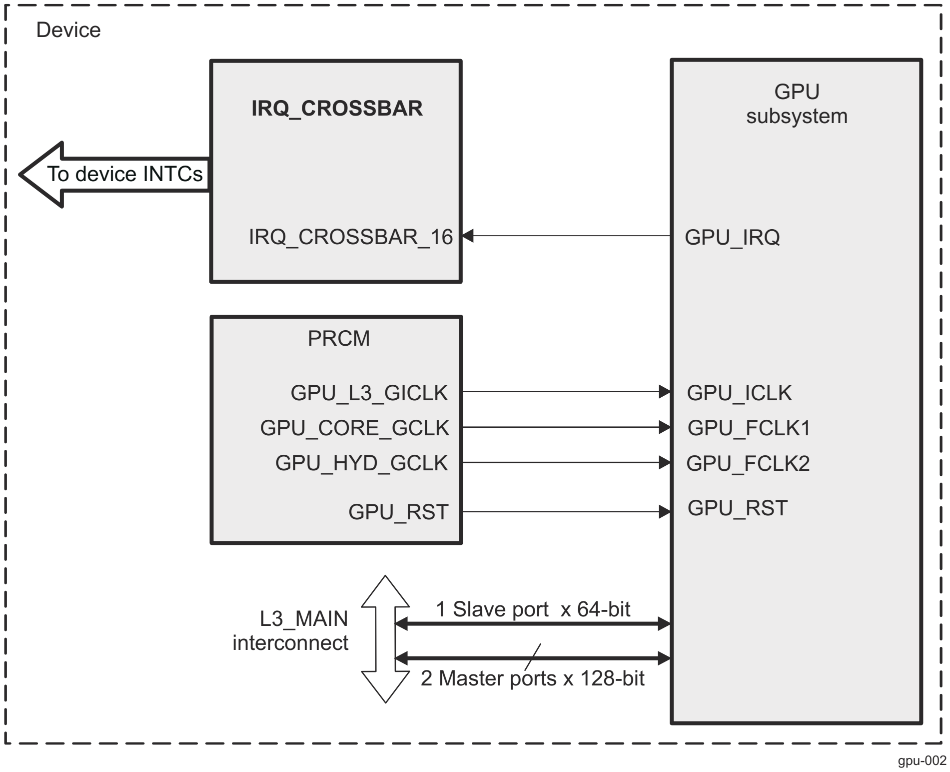


Figure 12-2. GPU Integration

The GPU subsystem is connected to the level 3 (L3_MAIN) interconnect by two 128-bit master and a 64-bit slave interfaces.

Table 12-1 through Table 12-3 summarize the integration of the module in the device.

Table 12-1. GPU Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
GPU	PD_GPU	L3_MAIN

Table 12-2. GPU Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GPU	GPU_ICLK	GPU_L3_GICLK	PRCM	GPU interface clock
GPU	GPU_FCLK1	GPU_CORE_GCLK	PRCM	GPU functional clock of the internal graphic cores
GPU	GPU_FCLK2	GPU_HYD_GCLK	PRCM	GPU functional clock of the internal L2-cache controller and memories
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GPU	GPU_RST	GPU_RST	PRCM	GPU non-retention reset signal

Table 12-3. GPU Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
GPU	GPU_IRQ	IRQ_CROSSBAR_16	MPU_IRQ_21 DSP1_IRQ_47 DSP2_IRQ_47	GPU interrupt request mapped to the device Interrupt Crossbar

Note

The "**Default Mapping**" column in Table 12-3, *GPU Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.
For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

Note

No DMA and no wake-up requests are generated by the GPU subsystem.

12.3 GPU Functional Description

12.3.1 GPU Block Diagram

The GPU subsystem is based on the PowerVR SGX544-MP2 core. Multicore GPU can split geometry and pixel rendering among the cores to improve performance proportional to the number of cores. The graphics software engineer programming the device GPU does not have to deal with the multiple cores, this is done by the graphics drivers. The multiple cores are completely hidden behind the standard high-level graphics APIs that are supported by the graphics core. The GPU architecture comprises the following elements:

- SGX544 cores
- PTLA
- Cross bar
- System-level cache (SLC)

Figure 12-3 shows the GPU top-level block diagram.

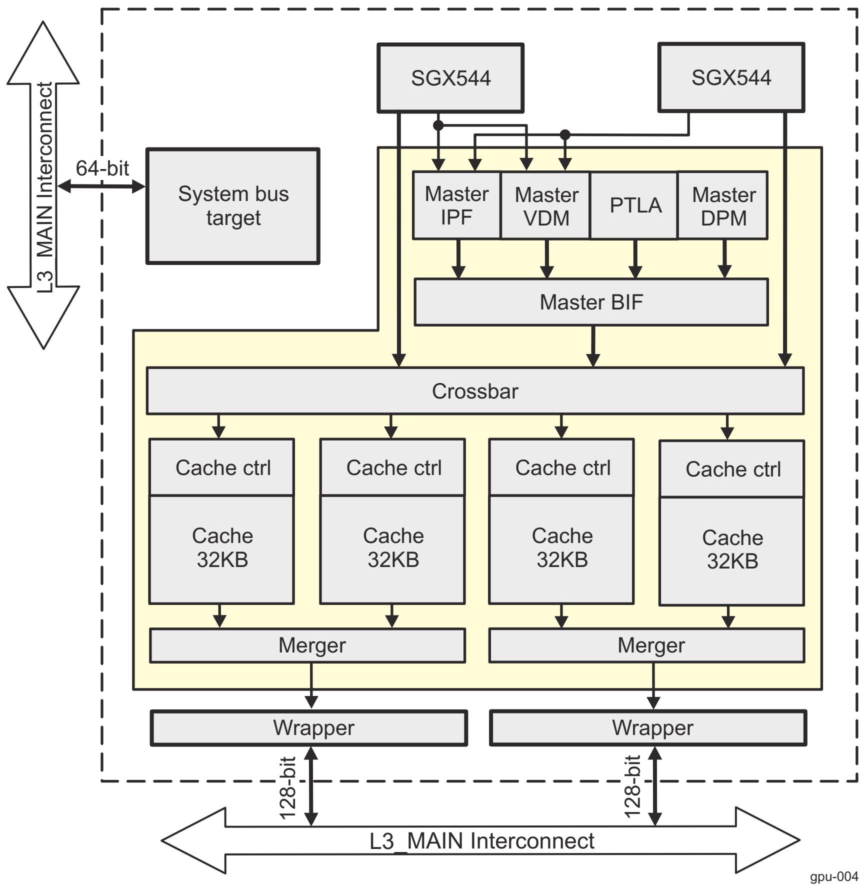


Figure 12-3. GPU Block Diagram

The SGX544-MP2 has 2× SGX544 cores. Graphics rendering is automatically load-balanced between the 2× SGX544 cores. It is possible to disable one or two of the cores if required. The glue logic is used to enable the multicore architecture. The crossbar enables any SGX544 core to access the SLC. The SLC is a 128-KiB unified multibanked cache with four banks of 32 KiB. The cache line size is 64 bytes. The SGX544 core accesses are interleaved in the different banks.

12.3.2 GPU Clock Configuration

The GPU subsystem operates from three clocks: an interface clock (GPU_ICLK) and two functional clocks (GPU_FCLK1 and GPU_FCLK2). The power, reset, and clock management (PRCM) module generates and distributes the clocks inside the device.

- The GPU_ICLK manages the data transfer on the L3_MAIN master and slave ports.

  The GPU_ICLK frequency is selected based on the L3_MAIN interconnect clock frequency of the whole device. For more information about the interface clock, see *CD_GPU Clock Domain*, in the *Power, Reset, and Clock Management*.

  When no longer required by the GPU subsystem, GPU_ICLK can be disabled by software at the PRCM level. For more information, see *PD_GPU Description*, in the *Power, Reset, and Clock Management*.

  Note

  GPU_ICLK is cut only if the GPU is ready to go into IDLE state.

- GPU_FCLK1 and GPU_FCLK2 are the functional clocks and are used inside the GPU subsystem to generate clock signals to multiple GPU clock domains. The GPU_FCLK1 input supplies clock to the internal graphics cores and the GPU_FCLK2 input supplies clock to the shared-cache memories and controllers.

  Using the clock source selection and the digital phase-locked loop (DPLL) settings, GPU_FCLK1 and GPU_FCLK2 frequencies can be adjusted.

  The GPU_FCLK1 and GPU_FCLK2 clocks are provided by the peripheral DPLL and the core DPLL, as described in *CD_GPU Clock Domain* in the *Power, Reset, and Clock Management*. Selection is made at the PRCM level.

  When no longer needed by the GPU subsystem, GPU_FCLK1 and GPU_FCLK2 can be cut by software at the PRCM level if the module is ready to enter IDLE state. For more information, see *PD_GPU Description* in the *Power, Reset, and Clock Management*.

12.3.3 GPU Software Reset

The GPU subsystem has its own reset domain. Global reset of the GPU is performed by activating the GPU_RST signal in the GPU_RST domain.

Note

The APIs delivered with the GPU provide a software reset functionally equivalent to a hardware reset.

12.3.4 GPU Power Management

The GPU subsystem has its own power domain (GPU power domain - PD_GPU).

The GPU handles automatic clock gating performed on the multiple internal module clock domains.

In addition, software-controlled clock gating is managed inside the GPU and handled by the related API delivered with the module.

For additional information about the GPU power domain, see Section 3.7.10, *PD_GPU Description* in the Chapter 3, *Power, Reset, and Clock Management*.

12.3.5 GPU Thermal Management

There is a dedicated thermal sensor to monitor operating temperature of the GPU subsystem in the chip. The GPU temperature processing logic is located within the device CTRL_MODULE_CORE and is capable to generate a thermal alert interrupt which can be mapped to all (host) interrupt controllers within the device, via the device IRQ_CROSSBAR. The thermal logic can also generate a thermal shut-down warm reset event to the device PRCM. For further details on the GPU thermal management operation features and register settings, refer to the Section 18.4.6.2, *Thermal Management Related Registers* of the Chapter 18, *Control Module*.

12.3.6 GPU Interrupt Requests

The GPU subsystem can generate one interrupt signal - GPU_IRQ mapped to the IRQ_CROSSBAR_16 input of the device interrupt crossbar.

For more details on programmable configuration of the GPU_IRQ mapping to the different device host interrupt controllers, refer to the Chapter 18, *Control Module* and the Chapter 17, *Interrupt Controllers*.

12.4 GPU Register Manual

CAUTION

All GPU registers are limited to 32-bit data accesses; 8- and 16-bit accesses are not allowed because they can corrupt register content.

12.4.1 GPU Instance Summary

Table 12-4. GPU Instance Summary

Module Name	Base Address	Size
GPU_WRAPPER	0x5600 FE00	512 bytes

The GPU domain's base address is at 0x5600 0000. GPU address space is 32MiB wide.

12.4.2 GPU Registers

12.4.2.1 GPU_WRAPPER Register Summary

Table 12-5. GPU_WRAPPER Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
REVISION	R	32	0x0000 0000	0x5600 FE00
HWINFO	R	32	0x0000 0004	0x5600 FE04
SYSCONFIG	RW	32	0x0000 0010	0x5600 FE10
IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x5600 FE24
IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x5600 FE28
IRQSTATUS_RAW_2	RW	32	0x0000 002C	0x5600 FE2C
IRQSTATUS_0	RW	32	0x0000 0030	0x5600 FE30
IRQSTATUS_1	RW	32	0x0000 0034	0x5600 FE34
IRQSTATUS_2	RW	32	0x0000 0038	0x5600 FE38
IRQENABLE_SET_0	RW	32	0x0000 003C	0x5600 FE3C
IRQENABLE_SET_1	RW	32	0x0000 0040	0x5600 FE40
IRQENABLE_SET_2	RW	32	0x0000 0044	0x5600 FE44
IRQENABLE_CLR_0	RW	32	0x0000 0048	0x5600 FE48
IRQENABLE_CLR_1	RW	32	0x0000 004C	0x5600 FE4C
IRQENABLE_CLR_2	RW	32	0x0000 0050	0x5600 FE50
PAGE_CONFIG	RW	32	0x0000 0100	0x5600 FF00
INTERRUPT_EVENT	RW	32	0x0000 0104	0x5600 FF04
DEBUG_CONFIG	RW	32	0x0000 0108	0x5600 FF08
DEBUG_STATUS_0	R	32	0x0000 010C	0x5600 FF0C
DEBUG_STATUS_1	R	32	0x0000 0110	0x5600 FF10

12.4.2.2 GPU_WRAPPER Register Description

Table 12-6. REVISION

Address Offset	0x0000 0000		
Physical Address	0x5600 FE00	**Instance**	GPU_WRAPPER
Description	Revision register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISIONID																															

Bits	Field Name	Description	Type	Reset
31:0	REVISIONID	Revision value	R	See [1]

(1) TI internal data

Table 12-7. Register Call Summary for Register REVISION

GPU Register Manual

- GPU Register Summary: [0]

Table 12-8. HWINFO

Address Offset	0x0000 0004		
Physical Address	0x5600 FE04	**Instance**	GPU_WRAPPER
Description	Hardware implementation information		
Type	R		

31–3	2	1–0
RESERVED	MEM_BUS_WIDTH	SYS_BUS_WIDTH

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	MEM_BUS_WIDTH	Memory bus width Read 0x0: 64 bits Read 0x1: 128 bits	R	1
1:0	SYS_BUS_WIDTH	System bus width Read 0x0: 32 bits Read 0x1: 64 bits Read 0x2: 128 bits Read 0x3: Reserved	R	0x1

Table 12-9. Register Call Summary for Register HWINFO

GPU Register Manual

- GPU Register Summary: [0]

Table 12-10. SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5600 FE10	**Instance**	GPU_WRAPPER
Description	System configuration register		
Type	RW		

31–6	5–4	3–2	1–0
RESERVED	STANDBY_MODE	IDLE_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000

Bits	Field Name	Description	Type	Reset
5:4	STANDBY_MODE	Clock standby mode: 0x0: Force-standby 0x1: No-standby 0x2: Smart-standby 0x3: Reserved	RW	0x2
3:2	IDLE_MODE	Clock idle mode: 0x0: Force-standby 0x1: No-standby 0x2: Smart-standby 0x3: Reserved	RW	0x2
1:0	RESERVED		R	0x0

Table 12-11. Register Call Summary for Register SYSCONFIG

GPU Register Manual

- GPU Register Summary: [0]

Table 12-12. IRQSTATUS_RAW_0

Address Offset	0x0000 0024		
Physical Address	0x5600 FE24	**Instance**	GPU_WRAPPER
Description	Raw IRQ 0 status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															INIT_MINTERRUPT_RAW

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	INIT_MINTERRUPT_RAW	Interrupt 0 raw event: Write 0x0: No action Write 0x1: Set event (used for debug) Read 0x0: No event pending Read 0x1: Event pending	RW	0

Table 12-13. Register Call Summary for Register IRQSTATUS_RAW_0

GPU Register Manual

- GPU Register Summary: [0]

Table 12-14. IRQSTATUS_RAW_1

Address Offset	0x0000 0028		
Physical Address	0x5600 FE28	**Instance**	GPU_WRAPPER
Description	Raw IRQ 1 status. Slave port interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	TARGET_SINTERRUPT_RAW

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_RAW	Interrupt 1 raw event: Write 0x0: No action Write 0x1: Set event (used for debug) Read 0x0: No event pending Read 0x1: Event pending	RW	0

Table 12-15. Register Call Summary for Register IRQSTATUS_RAW_1

GPU Register Manual

- GPU Register Summary: [0]

Table 12-16. IRQSTATUS_RAW_2

Address Offset	0x0000 002C		
Physical Address	0x5600 FE2C	**Instance**	GPU_WRAPPER
Description	Raw IRQ 2 status. Core interrupt.		
Type	RW		

31–1	0
RESERVED	THALIA_IRQ_RAW

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	THALIA_IRQ_RAW	Interrupt 0 raw event: Write 0x0: No action Write 0x1: Set event (used for debug) Read 0x0: No event pending Read 0x1: Event pending	RW	0

Table 12-17. Register Call Summary for Register IRQSTATUS_RAW_2

GPU Register Manual

- GPU Register Summary: [0]

Table 12-18. IRQSTATUS_0

Address Offset	0x0000 0030		
Physical Address	0x5600 FE30	**Instance**	GPU_WRAPPER
Description	Interrupt 0 status event. Master port interrupt.		

Table 12-18. IRQSTATUS_0 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															INIT_MINTERRUPT_STATUS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	INIT_MINTERRUPT_STATUS	Interrupt 0 raw event: Write 0x0: No action Write 0x1: Clear event Read 0x0: No event pending Read 0x1: Event pending and interrupt enabled	RW	0

Table 12-19. Register Call Summary for Register IRQSTATUS_0

GPU Register Manual

- GPU Register Summary: [0]

Table 12-20. IRQSTATUS_1

Address Offset	0x0000 0034		
Physical Address	0x5600 FE34	**Instance**	GPU_WRAPPER
Description	Interrupt 1 - slave port status event		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TARGET_SINTERRUPT_STATUS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_STATUS	Interrupt 0 raw event: Write 0x0: No action Write 0x1: Clear event Read 0x0: No event pending Read 0x1: Event pending and interrupt enabled	RW	0

Table 12-21. Register Call Summary for Register IRQSTATUS_1

GPU Register Manual

- GPU Register Summary: [0]

Table 12-22. IRQSTATUS_2

Address Offset	0x0000 0038		
Physical Address	0x5600 FE38	**Instance**	GPU_WRAPPER
Description	Interrupt 2 - Core status event		
Type	RW		

31–1	0
RESERVED	THALIA_IRQ_STATUS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	THALIA_IRQ_STATUS	Interrupt 0 raw event: Write 0x0: No action Write 0x1: Clear event Read 0x0: No event pending Read 0x1: Event pending and interrupt enabled	RW	0

Table 12-23. Register Call Summary for Register IRQSTATUS_2

GPU Register Manual

- GPU Register Summary: [0]

Table 12-24. IRQENABLE_SET_0

Address Offset	0x0000 003C		
Physical Address	0x5600 FE3C	**Instance**	GPU_WRAPPER
Description	Enable Interrupt 0 - Master port.		
Type	RW		

31–1	0
RESERVED	INIT_MINTERRUPT_ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000

Bits	Field Name	Description	Type	Reset
0	INIT_MINTERRUPT_ENABLE	To enable interrupt: Write 0x0: No action Write 0x1: Enable interrupt Read 0x0: Interrupt is disabled Read 0x1: Interrupt is enabled	RW	0

Table 12-25. Register Call Summary for Register IRQENABLE_SET_0

GPU Register Manual

- GPU Register Summary: [0]

Table 12-26. IRQENABLE_SET_1

Address Offset	0x0000 0040		
Physical Address	0x5600 FE40	**Instance**	GPU_WRAPPER
Description	Enable Interrupt 1. Core interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TARGET_SINTERRUPT_ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_ENABLE	To enable interrupt: Write 0x0: No action Write 0x1: Enable interrupt Read 0x0: Interrupt is disabled Read 0x1: Interrupt is enabled	RW	0

Table 12-27. Register Call Summary for Register IRQENABLE_SET_1

GPU Register Manual

- GPU Register Summary: [0]

Table 12-28. IRQENABLE_SET_2

Address Offset	0x0000 0044		
Physical Address	0x5600 FE44	**Instance**	GPU_WRAPPER
Description	Enable Interrupt 2. Core interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	THALIA_IRQ_ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	THALIA_IRQ_ENABLE	To enable interrupt: Write 0x0: No action Write 0x1: Enable interrupt Read 0x0: Interrupt is disabled Read 0x1: Interrupt is enabled	RW	0

Table 12-29. Register Call Summary for Register IRQENABLE_SET_2

GPU Register Manual

- GPU Register Summary: [0]

Table 12-30. IRQENABLE_CLR_0

Address Offset	0x0000 0048		
Physical Address	0x5600 FE48	**Instance**	GPU_WRAPPER
Description	Disable Interrupt 0 - Master port.		
Type	RW		

31–1	0
RESERVED	INIT_MINTERRUPT_DISABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	INIT_MINTERRUPT_DISABLE	To disable interrupt: Write 0x0: No action Write 0x1: Disable interrupt Read 0x0: Interrupt is disabled Read 0x1: Interrupt is enabled	RW	0

Table 12-31. Register Call Summary for Register IRQENABLE_CLR_0

GPU Register Manual

- GPU Register Summary: [0]

Table 12-32. IRQENABLE_CLR_1

Address Offset	0x0000 004C		
Physical Address	0x5600 FE4C	**Instance**	GPU_WRAPPER
Description	Disable Interrupt 2 - Core interrupt.		

Table 12-32. IRQENABLE_CLR_1 (continued)

Type	RW

31–1	0
RESERVED	TARGET_SINTERRUPT_DISABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_DISABLE	To disable interrupt: Write 0x0: No action Write 0x1: Disable interrupt Read 0x0: Interrupt is disabled Read 0x1: Interrupt is enabled	RW	0

Table 12-33. Register Call Summary for Register IRQENABLE_CLR_1

GPU Register Manual

- GPU Register Summary: [0]

Table 12-34. IRQENABLE_CLR_2

Address Offset	0x0000 0050		
Physical Address	0x5600 FE50	**Instance**	GPU_WRAPPER
Description	Disable Interrupt 2 - Core interrupt.		
Type	RW		

31–1	0
RESERVED	THALIA_IRQ_DISABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	THALIA_IRQ_DISABLE	To disable interrupt: Write 0x0: No action Write 0x1: Disable interrupt Read 0x0: Interrupt is disabled Read 0x1: Interrupt is enabled	RW	0

Table 12-35. Register Call Summary for Register IRQENABLE_CLR_2

GPU Register Manual

- GPU Register Summary: [0]

Table 12-36. PAGE_CONFIG

Address Offset	0x0000 0100		
Physical Address	0x5600 FF00	**Instance**	GPU_WRAPPER
Description	Configure memory pages.		
Type	RW		

31	30:5	4:3	2	1:0
THALIA_INT_BYPASS	RESERVED	OCP_PAGE_SIZE	MEM_PAGE_CHECK_EN	MEM_PAGE_SIZE

Bits	Field Name	Description	Type	Reset
31	THALIA_INT_BYPASS	Bypass OCP IPG interrupt logic 0x0: Do not bypass 0x1 Bypass core interrupt to I/O pin; that is, disregard the interrupt enable setting in the IPG register.	RW	0
30:5	RESERVED		R	0x000 0000
4:3	OCP_PAGE_SIZE	Defines the page size on OCP memory interface: 0x0: 4 KiB 0x1: 2 KiB 0x2: 1 KiB 0x3: 512B	RW	0x2
2	MEM_PAGE_CHECK_EN	To enable page boundary checking: 0x0: Disabled 0x1: Enabled	RW	1
1:0	MEM_PAGE_SIZE	Defines the page size on internal memory interface: 0x0: 4 KiB 0x1: 2 KiB 0x2: 1 KiB 0x3: 512B	RW	0x0

Table 12-37. Register Call Summary for Register PAGE_CONFIG

GPU Register Manual

- GPU Register Summary: [0]

Table 12-38. INTERRUPT_EVENT

Address Offset	0x0000 0104		
Physical Address	0x5600 FF04	**Instance**	GPU_WRAPPER
Description	Interrupt events		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	TARGET_INVALID_OCP_CMD	TARGET_CMD_FIFO_FULL	TARGET_RESP_FIFO_FULL	RESERVED	INIT_MEM_REQ_FIFO_OVERRUN_1	INIT_READ_TAG_FIFO_OVERRUN_1	INIT_PAGE_CROSS_ERROR_1	INIT_RESP_ERROR_1	INIT_RESP_UNUSED_TAG_1	INIT_RESP_UNEXPECTED_1	RESERVED	INIT_MEM_REQ_FIFO_OVERRUN_0	INIT_READ_TAG_FIFO_OVERRUN_0	INIT_PAGE_CROSS_ERROR_0	INIT_RESP_ERROR_0	INIT_RESP_UNUSED_TAG_0	INIT_RESP_UNEXPECTED_0

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	TARGET_INVALID_OCP_CMD	Invalid command from OCP: Write 0x0: Clear the event Write 0x1: Set the event and interrupt if enabled (debug only) Read 0x0: No event pending Read 0x1: Event pending	RW	0
17	TARGET_CMD_FIFO_FULL	Command FIFO full: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
16	TARGET_RESP_FIFO_FULL	Response FIFO full: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
15:14	RESERVED		R	0x0
13	INIT_MEM_REQ_FIFO_OVERRUN_1	Memory request FIFO overrun: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
12	INIT_READ_TAG_FIFO_OVERRUN_1	Read tag FIFO overrun: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
11	INIT_PAGE_CROSS_ERROR_1	Memory page had been crossed during a burst: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
10	INIT_RESP_ERROR_1	Receiving error response: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0

Bits	Field Name	Description	Type	Reset
9	INIT_RESP_UNUSED_TAG_1	Receiving response on an unused OCP TAG: Write 0x0: Clear the event Write 0x1: Set the event and interrupt if enabled (debug only) Read 0x0: No event pending Read 0x1: Event pending	RW	0
8	INIT_RESP_UNEXPECTED_1	Receiving response when not expected: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
7:6	RESERVED		R	0x0
5	INIT_MEM_REQ_FIFO_OVERRUN_0	Memory request FIFO overrun; Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
4	INIT_READ_TAG_FIFO_OVERRUN_0	Read tag FIFO overrun: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
3	INIT_PAGE_CROSS_ERROR_0	Memory page had been crossed during a burst. Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
2	INIT_RESP_ERROR_0	Receiving error response: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
1	INIT_RESP_UNUSED_TAG_0	Receiving response on an unused OCP TAG: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0
0	INIT_RESP_UNEXPECTED_0	Receiving response when not expected: Write 0x0: Clear the event. Write 0x1: Set the event and interrupt if enabled (debug only). Read 0x0: No event pending Read 0x1: Event pending	RW	0

Table 12-39. Register Call Summary for Register INTERRUPT_EVENT

GPU Register Manual

- GPU Register Summary: [0]

Table 12-40. DEBUG_CONFIG

Address Offset	0x0000 0108		
Physical Address	0x5600 FF08	**Instance**	GPU_WRAPPER
Description	Configuration of debug modes		

Table 12-40. DEBUG_CONFIG (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										SELECT_INT_IDLE	FORCE_PASS_DATA	FORCE_INIT_IDLE		FORCE_TARGET_IDLE	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	SELECT_INT_IDLE	To select which idle the disconnect protocol should act on: 0x0: Whole SGX idle 0x1: OCP initiator idle	RW	0
4	FORCE_PASS_DATA	Forces the initiator to pass data independent of disconnect protocol: 0x0: Do not force, normal operation 0x1: Never fence request to OCP	RW	0
3:2	FORCE_INIT_IDLE	Forces initiator idle: 0x0, 0x3: Do not force, normal operation 0x1: Always idle 0x2: Never idle	RW	0x0
1:0	FORCE_TARGET_IDLE	Forces target idle: 0x0, 0x3: Do not force, normal operation 0x1: Always idle 0x2: Never idle	RW	0x0

Table 12-41. Register Call Summary for Register DEBUG_CONFIG

GPU Register Manual

- GPU Register Summary: [0]

Table 12-42. DEBUG_STATUS_0

Address Offset	0x0000 010C		
Physical Address	0x5600 FF0C	**Instance**	GPU_WRAPPER
Description	Port0 debug status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD_DEBUG_STATE	CMD_RESP_DEBUG_STATE	TARGET_IDLE	RESP_FIFO_FULL	CMD_FIFO_FULL	RESP_ERROR	WHICH_TARGET_REGISTER					TARGET_CMD_OUT			INIT_MSTANDBY	INIT_MWAIT	INIT_MDISCREQ	INIT_MDISCACK		INIT_SCONNECT_2	INIT_SCONNECT_1	INIT_SCONNECT_0	INIT_MCONNECT		TARGET_SIDLEACK		TARGET_SDISCACK		TARGET_SIDLEREQ	TARGET_SCONNECT	TARGET_MCONNECT	

Bits	Field Name	Description	Type	Reset
31	CMD_DEBUG_STATE	Target command state-machine: 0x0: IDLE 0x1: Accept command	R	0
30	CMD_RESP_DEBUG_STATE	Target response state-machine: 0x0: Send accept 0x1: Wait accept	R	0
29	TARGET_IDLE	Target idle	R	0
28	RESP_FIFO_FULL	Target response FIFO full	R	0
27	CMD_FIFO_FULL	Target command FIFO full	R	0
26	RESP_ERROR	Respond to OCP with error, which could be caused by either address misalignment or invalid byte enable.	R	0
25:21	WHICH_TARGET_REGISTER	Indicates which OCP target registers to read	R	0x00
20:18	TARGET_CMD_OUT	Command received from OCP: 0x0: CMD_WRSYS 0x1: CMD_RDSYS 0x2: CMD_WR_ERROR 0x3: CMD_RD_ERROR 0x4: CMD_CHK_WRADDR_PAGE (not used) 0x5: CMD_CHK_RDADDR_PAGE (not used) 0x6: CMD_TARGET_REG_WRITE 0x7: CMD_TARGET_REG_READ	R	0x0
17	INIT_MSTANDBY	Status of init_MStandby signal	R	0
16	INIT_MWAIT	Status of init_MWait signal	R	0
15	INIT_MDISCREQ	Request to disconnect from OCP interface	R	0
14:13	INIT_MDISCACK	Disconnect status of the OCP interface: 0x0: FUNCT 0x1: TRANS 0x2: Reserved 0x3: IDLE	R	0x0
12	INIT_SCONNECT_2	Defines whether to wait in M_WAIT state for MConnect FSM: 0x0: Skip M_WAIT state 0x1: Wait in M_WAIT state	R	0
11	INIT_SCONNECT_1	Defines the busy-ness state of the slave: 0x0: Slave is drained 0x1: Slave is loaded	R	0
10	INIT_SCONNECT_0	Disconnect from slave: 0x0: Disconnect request from slave 0x1: Connect request from slave	R	0
9:8	INIT_MCONNECT	Initiator MConnect state: 0x0: M_OFF 0x1: M_WAIT 0x2: M_DISC 0x3: M_CON	R	0x0
7:6	TARGET_SIDLEACK	Acknowledge the SIdleAck state-machine: 0x0: FUNCT 0x1: SLEEP TRANS 0x2: Reserved 0x3: IDLE	R	0x0
5:4	TARGET_SDISCACK	Acknowledge the SDiscAck state-machine: 0x0: FUNCT 0x1: TRANS 0x2: Reserved 0x3: IDLE	R	0x0
3	TARGET_SIDLEREQ	Request the target to go idle: 0 Do not go idle, or go active 1 Go idle	R	0

Bits	Field Name	Description	Type	Reset
2	TARGET_SCONNECT	Target SConnect bit 0 state: 0x0: Disconnect interface 0x1: Connect OCP interface	R	0
1:0	TARGET_MCONNECT	Target MConnect state: 0x0: M_OFF 0x1: M_WAIT 0x2: M_DISC 0x3: M_CON	R	0x0

Table 12-43. Register Call Summary for Register DEBUG_STATUS_0

GPU Register Manual

- GPU Register Summary: [0]

Table 12-44. DEBUG_STATUS_1

Address Offset	0x0000 0110		
Physical Address	0x5600 FF10	**Instance**	GPU_WRAPPER
Description	Port1 debug status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD_DEBUG_STATE	CMD_RESP_DEBUG_STATE	TARGET_IDLE	RESP_FIFO_FULL	CMD_FIFO_FULL	RESP_ERROR	WHICH_TARGET_REGISTER					TARGET_CMD_OUT			INIT_MSTANDBY	INIT_MWAIT	INIT_MDISCREQ	INIT_MDISCACK		INIT_SCONNECT_2	INIT_SCONNECT_1	INIT_SCONNECT_0	INIT_MCONNECT		TARGET_SIDLEACK		TARGET_SDISCACK		TARGET_SIDLEREQ	TARGET_SCONNECT	TARGET_MCONNECT	

Bits	Field Name	Description	Type	Reset
31	CMD_DEBUG_STATE	Target command state-machine: 0x0: IDLE 0x1: Accept command	R	0
30	CMD_RESP_DEBUG_STATE	Target response state-machine: 0x0: Send accept 0x1: Wait accept	R	0
29	TARGET_IDLE	Target idle	R	0
28	RESP_FIFO_FULL	Target response FIFO full	R	0
27	CMD_FIFO_FULL	Target command FIFO full	R	0
26	RESP_ERROR	Respond to OCP with error, which could be caused by either address misalignment or invalid byte enable.	R	0
25:21	WHICH_TARGET_REGISTER	Indicates which OCP target registers to read	R	0x00
20:18	TARGET_CMD_OUT	Command received from OCP: 0x0: CMD_WRSYS 0x1: CMD_RDSYS 0x2: CMD_WR_ERROR 0x3: CMD_RD_ERROR 0x4: CMD_CHK_WRADDR_PAGE (not used) 0x5: CMD_CHK_RDADDR_PAGE (not used) 0x6: CMD_TARGET_REG_WRITE 0x7: CMD_TARGET_REG_READ	R	0x0
17	INIT_MSTANDBY	Status of init_MStandby signal	R	0
16	INIT_MWAIT	Status of init_MWait signal	R	0

Bits	Field Name	Description	Type	Reset
15	INIT_MDISCREQ	Request to disconnect from OCP interface	R	0
14:13	INIT_MDISCACK	Disconnect status of the OCP interface: 0x0: FUNCT 0x1: SLEEP TRANS 0x2: Reserved 0x3: IDLE	R	0x0
12	INIT_SCONNECT_2	Defines whether to wait in M_WAIT state for MConnect FSM: 0x0: Skip M_WAIT state. 0x1: Wait in M_WAIT state.	R	0
11	INIT_SCONNECT_1	Defines the busy-ness state of the slave: 0x0: Slave is drained. 0x1: Slave is loaded.	R	0
10	INIT_SCONNECT_0	Disconnect from slave: 0x0: Disconnect request from slave 0x1: Connect request from slave	R	0
9:8	INIT_MCONNECT	Initiator MConnect state: 0x0: M_OFF 0x1: M_WAIT 0x2: M_DISC 0x3: M_CON	R	0x0
7:6	TARGET_SIDLEACK	Acknowledge the SIdleAck state-machine: 0x0: FUNCT 0x1: SLEEP TRANS 0x2: Reserved 0x3: IDLE	R	0x0
5:4	TARGET_SDISCACK	Acknowledge the SDiscAck state-machine: 0x0: FUNCT 0x1: TRANS 0x2: Reserved 0x3: IDLE	R	0x0
3	TARGET_SIDLEREQ	Request the target to go idle: 0x0: Do not go idle, or go active 0x1: Go idle	R	0
2	TARGET_SCONNECT	Target SConnect bit 0 state: 0x0: Disconnect interface 0x1: Connect OCP interface	R	0
1:0	TARGET_MCONNECT	Target MConnect state: 0x0: M_OFF 0x1: M_WAIT 0x2: M_DISC 0x3: M_CON	R	0x0

Table 12-45. Register Call Summary for Register DEBUG_STATUS_1

GPU Register Manual

- GPU Register Summary: [0]

Chapter 13

2D Graphics Accelerator

This chapter describes the advanced bit blitter 2-dimensional (2D) graphics accelerator (BB2D) for the device.

Note

The BB2D subsystem is an instantiation by Texas Instruments of the GC320™ core from Vivante Corp.

This document contains materials that are © Vivante Corp.

GC320 is a trademark or registered trademark of Vivante Corp.

13.1 BB2D Overview 2927
13.2 BB2D Integration 2929
13.3 BB2D Functional Description 2931
13.4 BB2D Register Manual 2933

13.1 BB2D Overview

The 2D graphics accelerator subsystem accelerates 2D graphics applications. The 2D graphics accelerator subsystem is based on the GC320 2D GPU core from Vivante Corporation. The hardware acceleration is brought to numerous 2D applications, including on-screen display and touch screen user interfaces, graphical user interfaces (GUIs) and menu displays, flash animation, and gaming.

Figure 13-1 shows the BB2D subsystem.

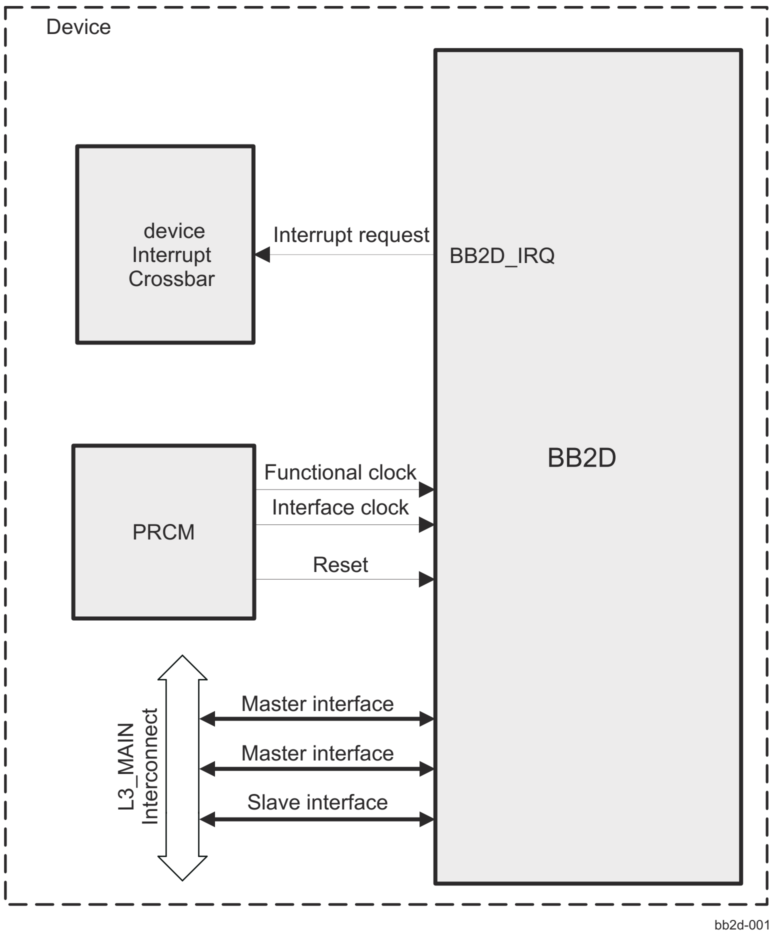


Figure 13-1. BB2D Overview

13.1.1 BB2D Key Features Overview

- API support:
  - OpenWF™, DirectFB
  - GDI/DirectDraw™
  - Flash
- **BB2D architecture:**
  - BitBlt and StretchBlt
  - DirectFB hardware acceleration
  - ROP2, ROP3, ROP4 full alpha blending and transparency
  - Clipping rectangle support
  - Alpha blending includes Java® 2 Porter-Duff compositing rules
  - 90-, 180-, 270-degree rotation on every primitive
  - YUV-to-RGB color space conversion
  - Programmable display format conversion with 14 source and 7 destination formats

  - High-quality 9-tap, 32-phase filter for image and video scaling at 1080p
  - Monochrome expansion for text rendering
  - 32 K × 32 K coordinate system
- **Hardware acceleration for DirectFB**:
  - High-speed video scaler
  - ROP2/3/4
  - Rectangle filling and drawing
  - Line drawing
  - Simple blitting
  - Stretch blitting
  - Blending with alpha channel (per-pixel alpha)
  - Blending with alpha factor (alpha modulation)
  - Nine source and destination blending functions
  - Porter-Duff rules support
  - Premultiplied alpha support
  - Colorized blitting (color modulation)
  - Source color keying
  - Destination color keying

The device BB2D generates a single (aggregate) interrupt request connected to the device Interrupt Crossbar. This allows for this interrupt to be programmatically mapped to multiple device host interrupt controllers (see Section 13.2).

13.2 BB2D Integration

This section describes subsystem integration in the device, including information about clocks, resets, and hardware requests.

Figure 13-2 shows BB2D integration.

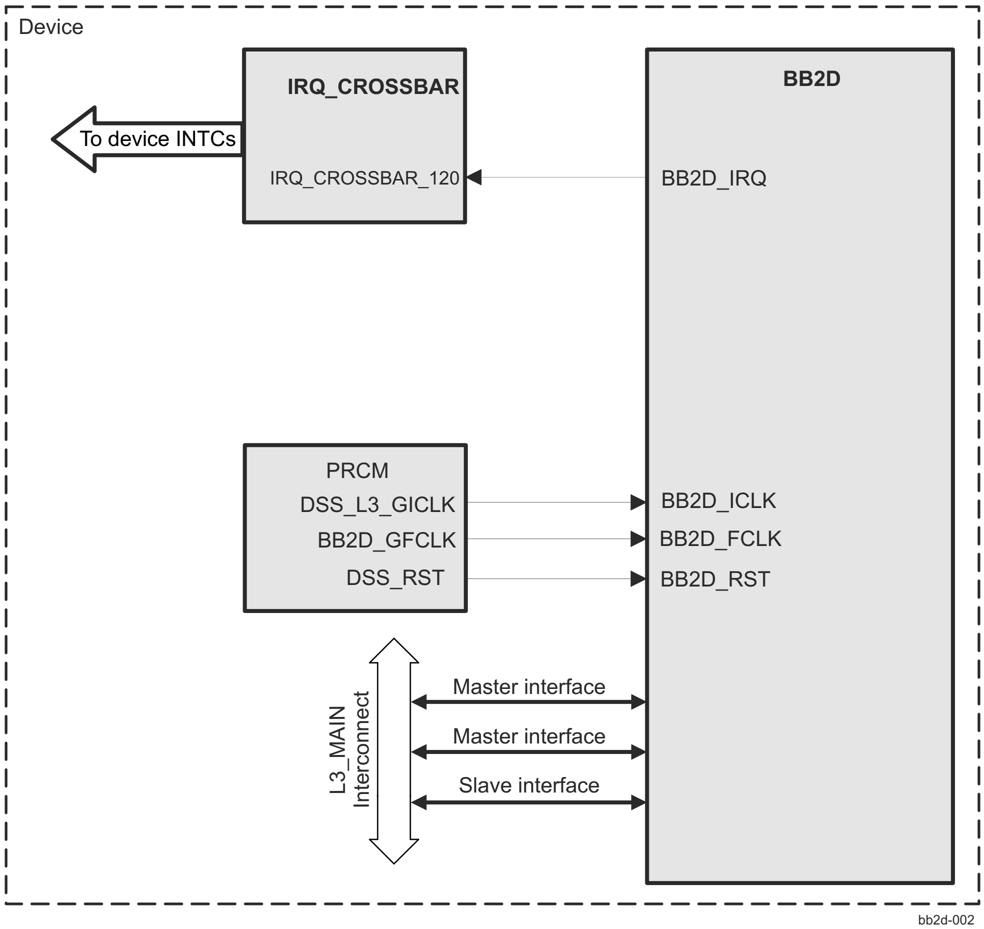


Figure 13-2. BB2D Integration

The BB2D subsystem is connected to the level 3 (L3_MAIN) interconnect by two 128-bit master interfaces and one 32-bit slave interface.

Table 13-1 through Table 13-3 summarize the integration of the subsystem in the device.

Table 13-1. BB2D Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
BB2D	PD_DSS	L3_MAIN

Table 13-2. BB2D Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
BB2D	BB2D_ICLK	DSS_L3_GICLK	PRCM	BB2D interfaces clock
	BB2D_FCLK	BB2D_GFCLK	PRCM	BB2D functional clock.
Resets				

Table 13-2. BB2D Clocks and Resets (continued)

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
BB2D	BB2D_RST	DSS_RST	PRCM	BB2D non-retention reset signal

Table 13-3. BB2D Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
BB2D	BB2D_IRQ	IRQ_CROSSBAR_120	MPU_IRQ_125 IPU1_IRQ_65 IPU2_IRQ_65	BB2D interrupt request to the device interrupt crossbar

Note

The "**Default Mapping**" column in Table 13-3, *BB2D Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

No DMA and no wake-up requests are generated by the BB2D module.

13.3 BB2D Functional Description

13.3.1 BB2D Block Diagram

The BB2D subsystem is based on the following main blocks:

- Host interface: Allows the BB2D core to communicate with external memory and the MPU through the L3_MAIN interconnect. In this block, data crosses clock domain boundaries.
- Memory controller: Internal memory unit that is the block-to-host interface for memory requests
- Graphics pipeline front-end: Inserts high-level primitives and commands into the graphics pipeline.
- 2D drawing and scaling engine: Draws 2D graphics primitives and rasterizes 2D images.
- Pixel engine: manipulates and filters pixels in rendered images. BB2D has four pixel pipes.

Figure 13-3 shows the BB2D top-level block diagram.

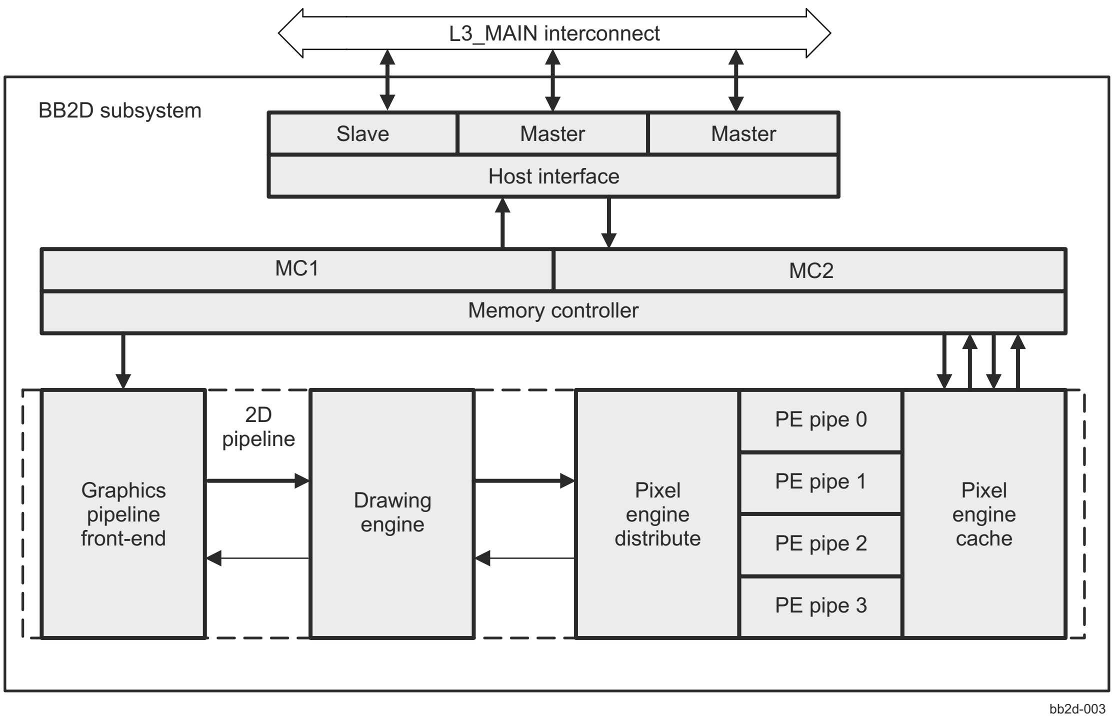


Figure 13-3. BB2D Block Diagram

13.3.2 BB2D Clock Configuration

The BB2D subsystem operates from two clocks: an interface clock (BB2D_ICLK) and functional clock (BB2D_FCLK). The power, reset, and clock management (PRCM) module generates and distributes the clocks inside the device.

- The BB2D_ICLK interface clock manages the data transfer on the L3_MAIN master and slave ports.

  The BB2D_ICLK frequency is selected based on the whole device L3_MAIN interconnect clock frequency. For more information on the interface clock, see *CD_DSS Clock Domain* , in the chapter, *Power, Reset and Clock Management* .

  When BB2D_ICLK is no longer required by the BB2D subsystem, it can be disabled by software at the PRCM level.

  Note

  BB2D_ICLK is cut only if the BB2D is ready to go into IDLE state.

- The BB2D_FCLK functional clock is used inside the BB2D subsystem to generate clock signals to multiple BB2D clock domains. The BB2D automatically gates clocks to domains, that are not currently in use.

Using the clock source selection and the DPLL settings, the frequency of BB2D_FCLK can be adjusted.

When BB2D_FCLK is no longer needed by the BB2D subsystem, it can be cut by software at the PRCM level, if the module is ready to enter IDLE state.

13.3.3 BB2D Software Reset

The BB2D subsystem is part of the DSS reset domain. A global reset of the BB2D is performed by activating the BB2D_RST signal in the DSS_RST domain.

Note

The APIs delivered with the BB2D provide a software reset functionally equivalent to a hardware reset.

13.3.4 BB2D Power Management

The BB2D subsystem is part of the DSS power domain (PD_DSS). For additional information about PD_DSS, see *PD_DSS Description* in the *Power, Reset, and Clock Management*.

Note

The BB2D handles automatic clock gating performed on the multiple internal clock domains.

When 2D operations are complete, software may set the GCGPOUT0[0] GCHOLD bit to 1 to enter a low-power state. Setting GCHOLD to 1 moves the BB2D operational state into IDLE. Once in IDLE state, the system standby hardware signal (mstandby) is asserted.

13.4 BB2D Register Manual

CAUTION

All BB2D registers are limited to 32-bit data accesses; 8- and 16-bit accesses are not allowed because they can corrupt register content.

13.4.1 BB2D Instance Summary

Table 13-4. BB2D Instance Summary

Module Name	Base Address	Size
BB2D	0x5900 0000	2 KiB

13.4.2 BB2D Registers

13.4.2.1 BB2D Register Summary

Table 13-5. BB2D Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
AQHICLOCKCONTROL	RW	32	0x0000 0000	0x5900 0000
AQHIIDLE	R	32	0x0000 0004	0x5900 0004
AQAXICONFIG	RW	32	0x0000 0008	0x5900 0008
AQAXISTATUS	R	32	0x0000 000C	0x5900 000C
AQINTRACKNOWLEDGE	R	32	0x0000 0010	0x5900 0010
AQINTRENBL	RW	32	0x0000 0014	0x5900 0014
AQIDENT	R	32	0x0000 0018	0x5900 0018
GCFEATURES	R	32	0x0000 001C	0x5900 001C
GCCHIPID	R	32	0x0000 0020	0x5900 0020
GCCHIPREV	R	32	0x0000 0024	0x5900 0024
GCCHIPDATE	R	32	0x0000 0028	0x5900 0028
GCCHIPTIME	R	32	0x0000 002C	0x5900 002C
GCCHIPCUSTOMER	R	32	0x0000 0030	0x5900 0030
GCMINORFEATURES0	R	32	0x0000 0034	0x5900 0034
GCRESETMEMCOUNTERS	W	32	0x0000 003C	0x5900 003C
GCTOTALREADS	R	32	0x0000 0040	0x5900 0040
GCTOTALWRITES	R	32	0x0000 0044	0x5900 0044
GCCHIPSPECS	R	32	0x0000 0048	0x5900 0048
GCTOTALWRITEBURSTS	R	32	0x0000 004C	0x5900 004C
GCTOTALWRITEREQS	R	32	0x0000 0050	0x5900 0050
GCTOTALWRITELASTS	R	32	0x0000 0054	0x5900 0054
GCTOTALREADBURSTS	R	32	0x0000 0058	0x5900 0058
GCTOTALREADREQS	R	32	0x0000 005C	0x5900 005C
GCTOTALREADLASTS	R	32	0x0000 0060	0x5900 0060
GCGPOUT0	RW	32	0x0000 0064	0x5900 0064
RESERVED	RW	32	0x0000 0068	0x5900 0068
RESERVED	RW	32	0x0000 006C	0x5900 006C
GCAXICONTROL	RW	32	0x0000 0070	0x5900 0070
GCMINORFEATURES1	R	32	0x0000 0074	0x5900 0074
GCTOTALCYCLES	RW	32	0x0000 0078	0x5900 0078
GCTOTALIDLECYCLES	RW	32	0x0000 007C	0x5900 007C
GCCHIPSPECS2	R	32	0x0000 0080	0x5900 0080

Table 13-5. BB2D Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
GCMINORFEATURES2	R	32	0x0000 0084	0x5900 0084
GCMODULEPOWERCONTROLS	RW	32	0x0000 0100	0x5900 0100
GCMODULEPOWERMODULECONTROL	RW	32	0x0000 0104	0x5900 0104
GCMODULEPOWERMODULESTATUS	R	32	0x0000 0108	0x5900 0108
GCREGMMUSTATUS	R	32	0x0000 0188	0x5900 0188
GCREGMMUCONTROL	W	32	0x0000 018C	0x5900 018C
GCREGMMUEXCEPTION0	RW	32	0x0000 0190	0x5900 0190
GCREGMMUEXCEPTION1	RW	32	0x0000 0194	0x5900 0194
GCREGMMUEXCEPTION2	RW	32	0x0000 0198	0x5900 0198
GCREGMMUEXCEPTION3	RW	32	0x0000 019C	0x5900 019C
AQMEMORYDEBUG	RW	32	0x0000 0414	0x5900 0414
AQREGISTERTIMINGCONTROL	RW	32	0x0000 042C	0x5900 042C
RESERVED	R	32	0x0000 0430	0x5900 0430
GCDISPLAYPRIORITY	RW	32	0x0000 0434	0x5900 0434
GCDBGCYCLECOUNTER	RW	32	0x0000 0438	0x5900 0438
GCOUTSTANDINGREADS0	R	32	0x0000 043C	0x5900 043C
GCOUTSTANDINGREADS1	R	32	0x0000 0440	0x5900 0440
GCOUTSTANDINGWRITES	R	32	0x0000 0444	0x5900 0444
GCDEBUGSIGNALSRA	R	32	0x0000 0448	0x5900 0448
GCDEBUGSIGNALSTX	R	32	0x0000 044C	0x5900 044C
GCDEBUGSIGNALSFE	R	32	0x0000 0450	0x5900 0450
GCDEBUGSIGNALSPE	R	32	0x0000 0454	0x5900 0454
GCDEBUGSIGNALSDE	R	32	0x0000 0458	0x5900 0458
GCDEBUGSIGNALSSH	R	32	0x0000 045C	0x5900 045C
GCDEBUGSIGNALSPA	R	32	0x0000 0460	0x5900 0460
GCDEBUGSIGNALSSE	R	32	0x0000 0464	0x5900 0464
GCDEBUGSIGNALSMC	R	32	0x0000 0468	0x5900 0468
GCDEBUGSIGNALSHI	R	32	0x0000 046C	0x5900 046C
GCDEBUGCONTROL0	RW	32	0x0000 0470	0x5900 0470
GCDEBUGCONTROL1	RW	32	0x0000 0474	0x5900 0474
GCDEBUGCONTROL2	RW	32	0x0000 0478	0x5900 0478
GCDEBUGCONTROL3	RW	32	0x0000 047C	0x5900 047C
GCBUSCONTROL	RW	32	0x0000 0480	0x5900 0480
GCREGENDIANNESS0	RW	32	0x0000 0484	0x5900 0484
GCREGENDIANNESS1	RW	32	0x0000 0488	0x5900 0488
GCREGENDIANNESS2	RW	32	0x0000 048C	0x5900 048C
GCREGDRAWPRIMITIVESTARTTIMESTAMP	R	32	0x0000 0490	0x5900 0490
GCREGDRAWPRIMITIVEENDTIMESTAMP	R	32	0x0000 0494	0x5900 0494
GCREGCONTROL0	RW	32	0x0000 0558	0x5900 0558
AQCMDBUFFERADDR	W	32	0x0000 0654	0x5900 0654
AQCMDBUFFERCTRL	W	32	0x0000 0658	0x5900 0658
AQFESTATUS	R	32	0x0000 065C	0x5900 065C
RESERVED	R	32	0x0000 0660	0x5900 0660
AQFEDEBUGCURCMDADR	R	32	0x0000 0664	0x5900 0664
RESERVED	R	32	0x0000 0668	0x5900 0668

Table 13-5. BB2D Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
RESERVED	R	32	0x0000 066C	0x5900 066C

13.4.2.2 BB2D Register Description

Table 13-6. AQHICLOCKCONTROL

Address Offset	0x0000 0000		
Physical Address	0x5900 0000	**Instance**	BB2D
Description	Clock control register.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19	18	17	16	15 14 13	12	11	10	9	8 7 6 5 4 3 2	1	0
RESERVED	MULTI_PIPE_USE_SINGLE_AXI	MULTI_PIPE_REG_SELECT	ISOLATE_GPU	IDLE_VG	IDLE2_D	IDLE3_D	RESERVED	SOFT_RESET	DISABLE_DEBUG_REGISTERS	DISABLE_RAM_CLOCK_GATING	FSCALE_CMD_LOAD	FSCALE_VAL	CLK2D_DIS	CLK3D_DIS

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	MULTI_PIPE_USE_SINGLE_AXI	Force all the transactions to go to one AXI	RW	0x0
23:20	MULTI_PIPE_REG_SELECT	Determines which HI/MC to use while reading registers	RW	0x0
19	ISOLATE_GPU	Isolate GPU bit	RW	0
18	IDLE_VG	VG pipe is idle	R	1
17	IDLE2_D	2D pipe is idle	R	1
16	IDLE3_D	3D pipe is idle	R	1
15:13	RESERVED		R	0x0
12	SOFT_RESET	Soft resets the subsystem	RW	0
11	DISABLE_DEBUG_REGISTERS	Disable debug registers. If this bit is 1, debug registers are clock gated	RW	0
10	DISABLE_RAM_CLOCK_GATING	Disables clock gating for RAMs	RW	0
9	FSCALE_CMD_LOAD		RW	0
8:2	FSCALE_VAL		RW	0x40
1	CLK2D_DIS	Disable 2D clock	RW	0
0	CLK3D_DIS	Disable 3D clock	RW	0

Table 13-7. AQHIIDLE

Address Offset	0x0000 0004		
Physical Address	0x5900 0004	**Instance**	BB2D
Description	Idle status register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

AXI_LP	RESERVED	IDLE_TS	IDLE_FP	IDLE_IM	IDLE_VG	IDLE_TX	IDLE_RA	IDLE_SE	IDLE_PA	IDLE_SH	IDLE_PE	IDLE_DE	IDLE_FE

Bits	Field Name	Description	Type	Reset
31	AXI_LP	AXI is in low power mode	R	0
30:12	RESERVED	Unused bits reserved for future expansion	R	0x7 FFFF
11	IDLE_TS	TS is idle	R	1
10	IDLE_FP	FP is idle	R	1
9	IDLE_IM	IM is idle	R	1
8	IDLE_VG	VG is idle	R	1
7	IDLE_TX	TX is idle	R	1
6	IDLE_RA	RA is idle	R	1
5	IDLE_SE	SE is idle	R	1
4	IDLE_PA	PA is idle	R	1
3	IDLE_SH	SH is idle	R	1
2	IDLE_PE	PE is idle	R	1
1	IDLE_DE	DE is idle	R	1
0	IDLE_FE	FE is idle	R	1

Table 13-8. AQAXICONFIG

Address Offset	0x0000 0008		
Physical Address	0x5900 0008	**Instance**	BB2D
Description	AXI config		
Type	RW		

31–16	15–12	11–8	7–4	3–0
RESERVED	ARCACHE	AWCACHE	ARID	AWID

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	ARCACHE		RW	0x0
11:8	AWCACHE		RW	0x0
7:4	ARID		RW	0x0
3:0	AWID		RW	0x0

Table 13-9. AQAXISTATUS

Address Offset	0x0000 000C		
Physical Address	0x5900 000C	**Instance**	BB2D
Description	AXI status		
Type	R		

31–10	9	8	7–4	3–0
RESERVED	DET_RD_ERR	DET_WR_ERR	RD_ERR_ID	WR_ERR_ID

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x00 0000
9	DET_RD_ERR		R	0
8	DET_WR_ERR		R	0
7:4	RD_ERR_ID		R	0x0
3:0	WR_ERR_ID		R	0x0

Table 13-10. AQINTRACKNOWLEDGE

Address Offset	0x0000 0010		
Physical Address	0x5900 0010	**Instance**	BB2D
Description	Interrupt acknowledge register. Each bit represents a corresponding event being triggered. Reading from this register clears the outstanding interrupt.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTR_VEC																															

Bits	Field Name	Description	Type	Reset
31:0	INTR_VEC		R	0x0000 0000

Table 13-11. AQINTRENBL

Address Offset	0x0000 0014		
Physical Address	0x5900 0014	**Instance**	BB2D
Description	Interrupt enable register. Each bit enables a corresponding event.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTR_ENBL_VEC																															

Bits	Field Name	Description	Type	Reset
31:0	INTR_ENBL_VEC		RW	0x0000 0000

Table 13-12. AQIDENT

Address Offset	0x0000 0018		
Physical Address	0x5900 0018	**Instance**	BB2D
Description	Identification register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAMILY								PRODUCT								REVISION				TECHNOLOGY				CUSTOMER							

Bits	Field Name	Description	Type	Reset
31:24	FAMILY	Family value 0x1: GC500 0x2: GC520 0x3: GC530 0x4: GC400 0x5: GC450 0x8: GC600 0x9: GC700 0xA: GC350 0xB: GC380 0xC: GC800 0x10: GC1000 0x14: GC2000	R	0x14

Bits	Field Name	Description	Type	Reset
23:16	PRODUCT	Product value	R	0x01
15:12	REVISION	Revision value	R	0x0
11:8	TECHNOLOGY	Technology value	R	0x0
7:0	CUSTOMER	Customer value	R	0x00

Table 13-13. GCFEATURES

Address Offset	0x0000 001C		
Physical Address	0x5900 001C	**Instance**	BB2D
Description	Shows which features are enabled in current subsystem implementation. 0 : NONE 1 : AVAILABLE		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FE20_BIT_INDEX	RS_YUV_TARGET	BYTE_WRITE_3D	FE20	VGTS	PIPE_VG	MEM32_BIT_SUPPORT	YUY2_RENDER_TARGET	HALF_TX_CACHE	HALF_PE_CACHE	YUY2_AVERAGING	NO_SCALER	BYTE_WRITE_2D	BUFFER_INTERLEAVING	NO422_TEXTURE	NO_EZ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_AREA	MODULE_CG	YUV420_TILER	HIGH_DYNAMIC_RANGE	FAST_SCALER	ETC1_TEXTURE_COMPRESSION	PIPE_2D	DC	MSAA	YUV420_FILTER	ZCOMPRESSION	DEBUG_MODE	DXT_TEXTURE_COMPRESSION	PIPE_3D	SPECIAL_ANTI_ALIASING	FAST_CLEAR

Bits	Field Name	Description	Type	Reset
31	FE20_BIT_INDEX	Supports 20 bit index.	R	1
30	RS_YUV_TARGET	Supports resolveing into YUV target.	R	1
29	BYTE_WRITE_3D	3D PE has byte write capability.	R	1
28	FE20	FE 2.0 is present.	R	0
27	VGTS	VG tesselator is present.	R	0
26	PIPE_VG	VG pipe is present.	R	0
25	MEM32_BIT_SUPPORT	32 bit memory address support.	R	0
24	YUY2_RENDER_TARGET	YUY2 support in PE and YUY2 to RGB conversion in resolve.	R	0
23	HALF_TX_CACHE	TX cache is half.	R	0
22	HALF_PE_CACHE	PE cache is half.	R	0
21	YUY2_AVERAGING	YUY2 averaging support in resolve.	R	1
20	NO_SCALER	No 2D scaler.	R	0
19	BYTE_WRITE_2D	Supports byte write in 2D.	R	1
18	BUFFER_INTERLEAVING	Supports interleaving depth and color buffers.	R	1
17	NO422_TEXTURE	No 422 texture input format.	R	0
16	NO_EZ	No early-Z.	R	0
15	MIN_AREA	Configured to have minimum area.	R	1
14	MODULE_CG	Second level clock gating is available.	R	1
13	YUV420_TILER	YUV 4:2:0 tiler is available.	R	1
12	HIGH_DYNAMIC_RANGE	Shows if there is HDR support.	R	1
11	FAST_SCALER	Shows if there is HD scaler.	R	1

Bits	Field Name	Description	Type	Reset
10	ETC1_TEXTURE_COMPRESSION	ETC1 texture compression.	R	1
9	PIPE_2D	Shows if there is 2D engine.	R	1
8	DC	Shows if there is a display controller.	R	0
7	MSAA	MSAA support.	R	1
6	YUV420_FILTER	YUV 4:2:0 support in filter blit.	R	1
5	ZCOMPRESSION	Depth and color compression.	R	0
4	DEBUG_MODE	Debug registers.	R	0
3	DXT_TEXTURE_COMPRESSION	DXT texture compression.	R	1
2	PIPE_3D	3D pipe.	R	0
1	SPECIAL_ANTI_ALIASING	Full-screen anti-aliasing.	R	1
0	FAST_CLEAR	Fast clear.	R	0

Table 13-14. GCCHIPID

Address Offset	0x0000 0020		
Physical Address	0x5900 0020	**Instance**	BB2D
Description	Shows the ID for the subsystem in BCD.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															

Bits	Field Name	Description	Type	Reset
31:0	ID	Subsystem ID in BCD	R	0x0000 0320

Table 13-15. GCCHIPREV

Address Offset	0x0000 0024		
Physical Address	0x5900 0024	**Instance**	BB2D
Description	Shows the revision for the subsystem in BCD.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															

Bits	Field Name	Description	Type	Reset
31:0	REV	Revision in BCD	R	0x0000 5301

Table 13-16. GCCHIPDATE

Address Offset	0x0000 0028		
Physical Address	0x5900 0028	**Instance**	BB2D
Description	Shows the release date for the subsystem.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATE																															

Bits	Field Name	Description	Type	Reset
31:0	DATE	Release date	R	0x2011 1103

Table 13-17. GCCHIPTIME

Address Offset	0x0000 002C		
Physical Address	0x5900 002C	**Instance**	BB2D
Description	Shows the release time for the subsystem.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															

Bits	Field Name	Description	Type	Reset
31:0	TIME	Release time	R	0x0014 0300

Table 13-18. GCCHIPCUSTOMER

Address Offset	0x0000 0030		
Physical Address	0x5900 0030	**Instance**	BB2D
Description	Shows the customer and group for the subsystem.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMPANY																GROUP															

Bits	Field Name	Description	Type	Reset
31:16	COMPANY	Company	R	0x0000
15:0	GROUP	Group	R	0x0000

Table 13-19. GCMINORFEATURES0

Address Offset	0x0000 0034		
Physical Address	0x5900 0034	**Instance**	BB2D
Description	Shows which minor features are enabled in the subsystem. 0 : NONE 1 : AVAILABLE		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENHANCE_VR	CORRECT_STENCIL	A8_TARGET_SUPPORT	NEW_TEXTURE	HIERARCHICAL_Z	BYPASS_IN_MSAA	VAA	BUG_FIXES0	SHADER_MSAA_SIDEBAND	MC_20	DEFAULT_REG0	EXTRA_SHADER_INSTRUCTIONS1	SHADER_GETS_W	VG_21	VG_FILTER	EXTRA_SHADER_INSTRUCTIONS0	COMPRESSION_FIFO_FIXED	TS_EXTENDED_COMMANDS	VG_20	SUPER_TILED_32X32	SEPARATE_TILE_STATUS_WHEN_INTERLEAVED	TILE_STATUS_2BITS	RENDER_8K	CORRECT_AUTO_DISABLE	PE20_2D	FAST_CLEAR_FLUSH	SPECIAL_MSAA_LOD	CORRECT_TEXTURE_CONVERTER	TEXTURE8_K	ENDIANNESS_CONFIG	DUAL_RETURN_BUS	FLIP_Y

Bits	Field Name	Description	Type	Reset
31	ENHANCE_VR	Enhance VR and add a mode to walk 16 pixels in 16-bit mode in vertical pass to improve cache hit rate when rotating 90/270.	R	1
30	CORRECT_STENCIL	Correct stencil behavior in depth only.	R	1
29	A8_TARGET_SUPPORT	2D engine supports A8 target.	R	0
28	NEW_TEXTURE	New texture unit is available.	R	0
27	HIERARCHICAL_Z	Hierarchiccal Z is supported.	R	1
26	BYPASS_IN_MSAA	Shader supports bypass mode when MSAA is enabled.	R	0
25	VAA	VAA is available or not.	R	0
24	BUG_FIXES0		R	1
23	SHADER_MSAA_SIDEBAND	Put the MSAA data into sideband fifo.	R	0
22	MC_20	New style MC with separate paths for color and depth.	R	0
21	DEFAULT_REG0	Unavailable registers will return 0.	R	1
20	EXTRA_SHADER_INSTRUCTIONS1	Sqrt, sin, cos instructions are available.	R	1
19	SHADER_GETS_W	W is sent to SH from RA.	R	1
18	VG_21	Minor updates to VG pipe (Event generation from VG, TS, PE). Tiled image support.	R	0
17	VG_FILTER	VG filter is available.	R	0
16	EXTRA_SHADER_INSTRUCTIONS0	Floor, ceil, and sign instructions are available.	R	1
15	COMPRESSION_FIFO_FIXED	If this bit is not set, the FIFO counter should be set to 50. Else, the default should remain.	R	1
14	TS_EXTENDED_COMMANDS	New commands added to the tessellator.	R	0
13	VG_20	Major updates to VG pipe (TS buffer tiling. State masking.).	R	0
12	SUPER_TILED_32X32	32 × 32 super tile is available.	R	1
11	SEPARATE_TILE_STATUS_WHEN_INTERLEAVED	Use 2 separate tile status buffers in interleaved mode.	R	1
10	TILE_STATUS_2BITS	2 bits are used instead of 4 bits for tile status.	R	1
9	RENDER_8K	Supports 8K render target.	R	1
8	CORRECT_AUTO_DISABLE	Reserved.	R	0
7	PE20_2D	2D PE 2.0 is present.	R	1
6	FAST_CLEAR_FLUSH	Proper flush is done in fast clear cache.	R	1
5	SPECIAL_MSAA_LOD	Special LOD calculation when MSAA is on.	R	1
4	CORRECT_TEXTURE_CONVERTER	Driver hack is not needed.	R	1
3	TEXTURE8_K	Supports 8K × 8K textures.	R	1
2	ENDIANNESS_CONFIG	Configurable endianness support.	R	1
1	DUAL_RETURN_BUS	Dual Return Bus from HI to clients.	R	1
0	FLIP_Y	Y flipping capability is added to resolve.	R	1

Table 13-20. GCRESETMEMCOUNTERS

Address Offset	0x0000 003C		
Physical Address	0x5900 003C	**Instance**	BB2D
Description	Writing 1 will reset the counters and stop counting. Write 0 to start counting again.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		W	0x649C CF7F
0	RESET	1: reset the counters and stop counting 0: start counting	W	1

Table 13-21. GCTOTALREADS

Address Offset	0x0000 0040		
Physical Address	0x5900 0040	**Instance**	BB2D
Description	Total reads in terms of 64 bits.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total reads in terms of 64 bits	R	0x0000 0000

Table 13-22. GCTOTALWRITES

Address Offset	0x0000 0044		
Physical Address	0x5900 0044	**Instance**	BB2D
Description	Total writes in terms of 64 bits.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total writes in terms of 64 bits	R	0x0000 0000

Table 13-23. GCCHIPSPECS

Address Offset	0x0000 0048		
Physical Address	0x5900 0048	**Instance**	BB2D
Description	Specs for the subsystem.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERTEX_OUTPUT_BUFFER_SIZE				NUM_PIXEL_PIPES			NUM_SHADER_CORES					RESERVED			VERTEX_CACHE_SIZE					THREAD_COUNT				TEMP_REGISTERS				STREAMS			

Bits	Field Name	Description	Type	Reset
31:28	VERTEX_OUTPUT_BUFFER_SIZE		R	0x0
27:25	NUM_PIXEL_PIPES		R	0x0
24:20	NUM_SHADER_CORES		R	0x00
19:17	RESERVED		R	0x0
16:12	VERTEX_CACHE_SIZE		R	0x00
11:8	THREAD_COUNT		R	0x0
7:4	TEMP_REGISTERS		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	STREAMS		R	0x0

Table 13-24. GCTOTALWRITEBURSTS

Address Offset	0x0000 004C		
Physical Address	0x5900 004C	**Instance**	BB2D
Description	Total write data count in terms of 64 bits.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total write data count in terms of 64 bits	R	0x0000 0000

Table 13-25. GCTOTALWRITEREQS

Address Offset	0x0000 0050		
Physical Address	0x5900 0050	**Instance**	BB2D
Description	Total write request count.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total write request count	R	0x0000 0000

Table 13-26. GCTOTALWRITELASTS

Address Offset	0x0000 0054		
Physical Address	0x5900 0054	**Instance**	BB2D
Description	Total WLAST count. This is used to match with GCTOTALWRITEREQS		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total WLAST count	R	0x0000 0000

Table 13-27. GCTOTALREADBURSTS

Address Offset	0x0000 0058		
Physical Address	0x5900 0058	**Instance**	BB2D
Description	Total read data count in terms of 64 bits.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total read data count in terms of 64 bits	R	0x0000 0000

Table 13-28. GCTOTALREADREQS

Address Offset	0x0000 005C		
Physical Address	0x5900 005C	**Instance**	BB2D
Description	Total read request count.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total read request count	R	0x0000 0000

Table 13-29. GCTOTALREADLASTS

Address Offset	0x0000 0060		
Physical Address	0x5900 0060	**Instance**	BB2D
Description	Total RLAST count. This is used to match with GCTOTALREADREQS		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Total RLAST count	R	0x0000 0000

Table 13-30. GCGPOUT0

Address Offset	0x0000 0064		
Physical Address	0x5900 0064	**Instance**	BB2D
Description	General purpose output register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															GCHOLD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	RW	0x0000 0000
0	GCHOLD	1 : Low power mode	RW	0

Table 13-31. GCAXICONTROL

Address Offset	0x0000 0070		
Physical Address	0x5900 0070	**Instance**	BB2D
Description	Special handling on AXI Bus		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:1	0
RESERVED	WR_FULL_BURST_MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	WR_FULL_BURST_MODE	0: NO_BURST_RESET_VALUE 1: BURST_RESET_VALUE	RW	0

Table 13-32. GCMINORFEATURES1

Address Offset	0x0000 0074		
Physical Address	0x5900 0074	**Instance**	BB2D
Description	Shows which features are enabled in the subsystem. 0 : NONE 1 : AVAILABLE		
Type	R		

31	30	29	28	27	26	25	24
FC_FLUSH_STALL	BUG_FIXES6	WIDE_LINE	MMU	OK_TO_GATE_AXI_CLOCK	RESOLVE_OFFSET	NEGATIVE_LOG_FIX	CORRECT_OVERFLOW_VG

23	22	21	20	19	18	17	16
HALTI0	LINEAR_TEXTURE_SUPPORT	NON_POWER_OF_TWO	TEXTURE_HORIZONTAL_ALIGNMENT_SELECT	NEW_FLOATING_POINT_ARITHMETIC	NEW_2D	BUG_FIXES5	DITHER_AND_FILTER_PLUS_ALPHA_2D

15	14	13	12	11	10	9	8
CORRECT_MIN_MAX_DEPTH	EXTENDED_PIXEL_FORMAT	TWO_STENCIL_REFERENCE	PIXEL_DITHER	HALF_FLOAT_PIPE	L2_WINDOWING	BUG_FIXES4	AUTO_RESTART_TS

7	6	5	4	3	2	1	0
CORRECT_AUTO_DISABLE	BUG_FIXES3	TEXTURE_STRIDE	BUG_FIXES2	BUG_FIXES1	VG_DOUBLE_BUFFER	V2_COMPRESSION	RSUV_SWIZZLE

Bits	Field Name	Description	Type	Reset
31	FC_FLUSH_STALL		R	1
30	BUG_FIXES6		R	1
29	WIDE_LINE		R	1
28	MMU		R	1
27	OK_TO_GATE_AXI_CLOCK		R	1
26	RESOLVE_OFFSET		R	1
25	NEGATIVE_LOG_FIX		R	1
24	CORRECT_OVERFLOW_VG		R	0
23	HALTI0		R	0
22	LINEAR_TEXTURE_SUPPORT		R	0

Bits	Field Name	Description	Type	Reset
21	NON_POWER_OF_TWO		R	0
20	TEXTURE_HORIZONTAL_ALIGNMENT_SELECT		R	1
19	NEW_FLOATING_POINT_ARITHMETIC		R	1
18	NEW_2D		R	1
17	BUG_FIXES5		R	1
16	DITHER_AND_FILTER_PLUS_ALPHA_2D	Dither and filter+alpha available.	R	1
15	CORRECT_MIN_MAX_DEPTH	EEZ and HZ are correct.	R	1
14	EXTENDED_PIXEL_FORMAT		R	0
13	TWO_STENCIL_REFERENCE		R	1
12	PIXEL_DITHER		R	1
11	HALF_FLOAT_PIPE		R	0
10	L2_WINDOWING		R	0
9	BUG_FIXES4		R	1
8	AUTO_RESTART_TS		R	0
7	CORRECT_AUTO_DISABLE		R	1
6	BUG_FIXES3		R	1
5	TEXTURE_STRIDE	Texture has stride and memory addressing.	R	0
4	BUG_FIXES2		R	1
3	BUG_FIXES1		R	1
2	VG_DOUBLE_BUFFER	Double buffering support for VG (second TS-->VG semaphore is present).	R	0
1	V2_COMPRESSION	V2 compression.	R	1
0	RSUV_SWIZZLE	Resolve UV swizzle.	R	1

Table 13-33. GCTOTALCYCLES

Address Offset	0x0000 0078		
Physical Address	0x5900 0078	**Instance**	BB2D
Description	Total cycles. This register is a free running counter. It can be reset by writing 0 to it.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLES																															

Bits	Field Name	Description	Type	Reset
31:0	CYCLES	Total cycles	RW	0x0000 1DE2

Table 13-34. GCTOTALIDLECYCLES

Address Offset	0x0000 007C		
Physical Address	0x5900 007C	**Instance**	BB2D
Description	Total cycles where the GPU is idle. It is reset when GCTOTALCYCLES register is written to. It looks at all the blocks but FE when determining the subsystem is idle.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLES																															

Bits	Field Name	Description	Type	Reset
31:0	CYCLES	Total cycles where the GPU is idle	RW	0x0000 1E08

Table 13-35. GCCHIPSPECS2

Address Offset	0x0000 0080		
Physical Address	0x5900 0080	**Instance**	BB2D
Description	Specs for the subsystem		
Type	R		

31–16	15–8	7–0
NUMBER_OF_CONSTANTS	INSTRUCTION_COUNT	BUFFER_SIZE

Bits	Field Name	Description	Type	Reset
31:16	NUMBER_OF_CONSTANTS		R	0x0000
15:8	INSTRUCTION_COUNT		R	0x00
7:0	BUFFER_SIZE		R	0x00

Table 13-36. GCMINORFEATURES2

Address Offset	0x0000 0084		
Physical Address	0x5900 0084	**Instance**	BB2D
Description	Shows which features are enabled in the subsystem 0 : NONE 1 : AVAILABLE		
Type	R		

Bit	Field
31–29	RESERVED
28	NO_INDEX_PATTERN
27	RESERVED
26	NOT_USED
25	MIXED_STREAMS
24	INTERLEAVER
23	FLUSH_FIXED_2D
22	YUV_CONVERSION
21	MULTI_SOURCE_BLT
20	YUV_STANDARD
19	TILE_FILLER
18	THREAD_WALKER_IN_PS
17	ONE_PASS_2D_FILTER
16	FULL_DIRECT_FB
15	TX_FILTER
14	DYNAMIC_FREQUENCY_SCALING
13	TX_YUV_ASSEMBLER
12	RGB888
11	HALTI1
10	S1S8
9	END_EVENT
8	PE_SWIZZLE
7	CORRECT_AUTO_DISABLE_COUNT_WIDTH
6	COMPOSITION
5	RECT_PRIMITIVE
4	LINEAR_PE
3	SUPER_TILED_TEXTURE
2	SEAMLESS_CUBE_MAP
1	LOGIC_OP
0	LINE_LOOP

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	NO_INDEX_PATTERN		R	1
27	RESERVED		R	1
26	NOT_USED		R	0
25	MIXED_STREAMS		R	1
24	INTERLEAVER		R	0
23	FLUSH_FIXED_2D		R	1
22	YUV_CONVERSION		R	1

Bits	Field Name	Description	Type	Reset
21	MULTI_SOURCE_BLT		R	1
20	YUV_STANDARD		R	1
19	TILE_FILLER		R	1
18	THREAD_WALKER_IN_PS		R	1
17	ONE_PASS_2D_FILTER		R	1
16	FULL_DIRECT_FB		R	1
15	TX_FILTER		R	0
14	DYNAMIC_FREQUENCY_SCALING		R	1
13	TX_YUV_ASSEMBLER		R	0
12	RGB888		R	0
11	HALTI1		R	0
10	S1S8		R	0
9	END_EVENT		R	0
8	PE_SWIZZLE		R	0
7	CORRECT_AUTO_DISABLE_COUNT_WIDTH		R	1
6	COMPOSITION		R	0
5	RECT_PRIMITIVE		R	0
4	LINEAR_PE		R	0
3	SUPER_TILED_TEXTURE		R	0
2	SEAMLESS_CUBE_MAP		R	0
1	LOGIC_OP		R	0
0	LINE_LOOP		R	0

Table 13-37. GCMODULEPOWERCONTROLS

Address Offset	0x0000 0100		
Physical Address	0x5900 0100	**Instance**	BB2D
Description	Control register for module level power controls.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">TURN_OFF_COUNTER</td><td colspan="8">RESERVED</td><td colspan="4">TURN_ON_COUNTER</td><td>RESERVED</td><td>DISABLE_STARVE_MODULE_CLOCK_GATING</td><td>DISABLE_STALL_MODULE_CLOCK_GATING</td><td>ENABLE_MODULE_CLOCK_GATING</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	TURN_OFF_COUNTER	Counter value for clock gating the module if the module is idle for this amount of clock cycles	RW	0x0014
15:8	RESERVED		R	0x00
7:4	TURN_ON_COUNTER	Number of clock cycles to wait after turning on the clock	RW	0x2
3	RESERVED		R	0
2	DISABLE_STARVE_MODULE_CLOCK_GATING	Disables module level clock gating for starve/idle condition	RW	0
1	DISABLE_STALL_MODULE_CLOCK_GATING	Disables module level clock gating for stall condition	RW	0
0	ENABLE_MODULE_CLOCK_GATING	Enables module level clock gating	RW	0

Table 13-38. GCMODULEPOWERMODULECONTROL

Address Offset	0x0000 0104		
Physical Address	0x5900 0104	**Instance**	BB2D
Description	Module level control registers.		
Type	RW		

31:8	7	6	5	4	3	2	1	0
RESERVED	DISABLE_MODULE_CLOCK_GATING_TX	DISABLE_MODULE_CLOCK_GATING_RA	DISABLE_MODULE_CLOCK_GATING_SE	DISABLE_MODULE_CLOCK_GATING_PA	DISABLE_MODULE_CLOCK_GATING_SH	DISABLE_MODULE_CLOCK_GATING_PE	DISABLE_MODULE_CLOCK_GATING_DE	DISABLE_MODULE_CLOCK_GATING_FE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x00 0000
7	DISABLE_MODULE_CLOCK_GATING_TX	Disables module level clock gating for starve/idle condition	RW	0
6	DISABLE_MODULE_CLOCK_GATING_RA	Disables module level clock gating for stall condition	RW	0
5	DISABLE_MODULE_CLOCK_GATING_SE	Enables module level clock gating	RW	0
4	DISABLE_MODULE_CLOCK_GATING_PA	Counter value for clock gating the module if the module is idle for this amount of clock cycles	RW	0
3	DISABLE_MODULE_CLOCK_GATING_SH	Number of clock cycles to wait after turning on the clock	RW	0
2	DISABLE_MODULE_CLOCK_GATING_PE	Disables module level clock gating for starve/idle condition	RW	0
1	DISABLE_MODULE_CLOCK_GATING_DE	Disables module level clock gating for stall condition	RW	0
0	DISABLE_MODULE_CLOCK_GATING_FE	Enables module level clock gating	RW	0

Table 13-39. GCMODULEPOWERMODULESTATUS

Address Offset	0x0000 0108		
Physical Address	0x5900 0108	**Instance**	BB2D
Description	Module level control status		
Type	R		

31:8	7	6	5	4	3	2	1	0
RESERVED	MODULE_CLOCK_GATED_TX	MODULE_CLOCK_GATED_RA	MODULE_CLOCK_GATED_SE	MODULE_CLOCK_GATED_PA	MODULE_CLOCK_GATED_SH	MODULE_CLOCK_GATED_PE	MODULE_CLOCK_GATED_DE	MODULE_CLOCK_GATED_FE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x00 0000
7	MODULE_CLOCK_GATED_TX	Module level clock gating is ON for TX	R	0
6	MODULE_CLOCK_GATED_RA	Module level clock gating is ON for RA	R	0
5	MODULE_CLOCK_GATED_SE	Module level clock gating is ON for SE	R	0
4	MODULE_CLOCK_GATED_PA	Module level clock gating is ON for PA	R	0
3	MODULE_CLOCK_GATED_SH	Module level clock gating is ON for SH	R	0
2	MODULE_CLOCK_GATED_PE	Module level clock gating is ON for PE	R	0
1	MODULE_CLOCK_GATED_DE	Module level clock gating is ON for DE	R	0
0	MODULE_CLOCK_GATED_FE	Module level clock gating is ON for FE	R	0

Table 13-40. GCREGMMUSTATUS

Address Offset	0x0000 0188		
Physical Address	0x5900 0188	**Instance**	BB2D
Description	Status register that holds which MMU generated an exception		
Type	R		

31:14	13:12	11:10	9:8	7:6	5:4	3:2	1:0
RESERVED	EXCEPTION3	RESERVED	EXCEPTION2	RESERVED	EXCEPTION1	RESERVED	EXCEPTION0

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	NOT USED	R	0x0 0000
13:12	EXCEPTION3	MMU 3 caused an exception and the GCREGMMUEXCEPTION3 register holds the offending address. 0x1: SLAVE_NOT_PRESENT 0x2: PAGE_NOT_PRESENT 0x3: WRITE_VIOLATION	R	0x0
11:10	RESERVED	NOT USED	R	0x0

Bits	Field Name	Description	Type	Reset
9:8	EXCEPTION2	MMU 2 caused an exception and the GCREGMMUEXCEPTION2 register holds the offending address. 0x1: SLAVE_NOT_PRESENT 0x2: PAGE_NOT_PRESENT 0x3: WRITE_VIOLATION	R	0x0
7:6	RESERVED	NOT USED	R	0x0
5:4	EXCEPTION1	MMU 1 caused an exception and the GCREGMMUEXCEPTION1 register holds the offending address. 0x1: SLAVE_NOT_PRESENT 0x2: PAGE_NOT_PRESENT 0x3: WRITE_VIOLATION	R	0x0
3:2	RESERVED	NOT USED	R	0x0
1:0	EXCEPTION0	MMU 0 caused an exception and the GCREGMMUEXCEPTION0 holds the offending address: 0x1: SLAVE_NOT_PRESENT 0x2: PAGE_NOT_PRESENT 0x3: WRITE_VIOLATION	R	0x0

Table 13-41. GCREGMMUCONTROL

Address Offset	0x0000 018C		
Physical Address	0x5900 018C	**Instance**	BB2D
Description	Control register that enables the MMU (one time shot).		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	NOT USED	W	0x0000 0000
0	ENABLE	Enable the MMU. For security reasons, once the MMU is enabled it cannot be disabled until reset.	W	0

Table 13-42. GCREGMMUEXCEPTION0

Address Offset	0x0000 0190		
Physical Address	0x5900 0190	**Instance**	BB2D
Description	Holds the original address that generated an exception		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	ADDRESS	The original address that generated an exception	RW	0x0000 0000

Table 13-43. GCREGMMUEXCEPTION1

Address Offset	0x0000 0194		
Physical Address	0x5900 0194	**Instance**	BB2D
Description	Holds the original address that generated an exception		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:0
ADDRESS

Bits	Field Name	Description	Type	Reset
31:0	ADDRESS	The original address that generated an exception	RW	0x0000 0000

Table 13-44. GCREGMMUEXCEPTION2

Address Offset	0x0000 0198		
Physical Address	0x5900 0198	**Instance**	BB2D
Description	Holds the original address that generated an exception		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ADDRESS

Bits	Field Name	Description	Type	Reset
31:0	ADDRESS	The original address that generated an exception	RW	0x0000 0000

Table 13-45. GCREGMMUEXCEPTION3

Address Offset	0x0000 019C		
Physical Address	0x5900 019C	**Instance**	BB2D
Description	Holds the original address that generated an exception		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ADDRESS

Bits	Field Name	Description	Type	Reset
31:0	ADDRESS	The original address that generated an exception	RW	0x0000 0000

Table 13-46. AQMEMORYDEBUG

Address Offset	0x0000 0414		
Physical Address	0x5900 0414	**Instance**	BB2D
Description			
Type	RW		

31	30	29-24	23	22	21-20	19	18	17	16-15	14	13-8	7-0
RESERVED	DONT_STALL_WRITES_TO_SAME_ADDRESS	ZCOMP_LIMIT	DISABLE_WRITE_DATA_SPEEDUP	DISABLE_STALL_READS	RESERVED	LIMIT_CONTROL	RESERVED	INTERLEAVE_BUFFER_LOW_LATENCY_MODE	RESERVED	DISABLE_MINI_MMU_CACHE	RESERVED	MAX_OUTSTANDING_READS

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	DONT_STALL_WRITES_TO_SAME_ADDRESS		RW	0
29:24	ZCOMP_LIMIT		RW	0x3C
23	DISABLE_WRITE_DATA_SPEEDUP		RW	0
22	DISABLE_STALL_READS		RW	0
21:20	RESERVED	Reserved	RW	0
19	LIMIT_CONTROL	Limit control 0: REQUESTS 1: DATA	RW	0
18	RESERVED	Reserved	R	0
17	INTERLEAVE_BUFFER_LOW_LATENCY_MODE		RW	0
16:15	RESERVED	Reserved	R	0x0
14	DISABLE_MINI_MMU_CACHE		RW	0
13:8	RESERVED	Reserved	R	0x00
7:0	MAX_OUTSTANDING_READS	Limits the total number of outstanding read requests.	RW	0x00

Table 13-47. AQREGISTERTIMINGCONTROL

Address Offset	0x0000 042C		
Physical Address	0x5900 042C	**Instance**	BB2D
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="9">RESERVED</td><td>LIGHT_SLEEP</td><td>DEEP_SLEEP</td><td>POWER_DOWN</td><td colspan="2">FAST_WTC</td><td colspan="2">FAST_RTC</td><td colspan="8">FOR_RF2P</td><td colspan="8">FOR_RF1P</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	RW	0x000
22	LIGHT_SLEEP	Light sleep	RW	0
21	DEEP_SLEEP	Deep sleep	RW	0
20	POWER_DOWN	Powerdown memory	RW	0
19:18	FAST_WTC	WTC for fast RAMs	RW	0x0
17:16	FAST_RTC	RTC for fast RAMs	RW	0x3
15:8	FOR_RF2P		RW	0x00
7:0	FOR_RF1P		RW	0x00

Table 13-48. GCDISPLAYPRIORITY

Address Offset	0x0000 0434		
Physical Address	0x5900 0434	**Instance**	BB2D
Description	Controls the priority of the display controller requests. This works like a PWM. One register gives the period, and the other gives the ON time. When PWM is ON, display requests are accepted if both display and the other request is valid. If it is OFF, the other request will be accepted. If only one request is valid, it takes the bus regardless of the PWM bit.		

Table 13-48. GCDISPLAYPRIORITY (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HIGH								PERIOD							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	HIGH	'Duty cycle'	RW	0x01
7:0	PERIOD	Period	RW	0x02

Table 13-49. GCDBGCYCLECOUNTER

Address Offset	0x0000 0438		
Physical Address	0x5900 0438	**Instance**	BB2D
Description	Increments every cycle.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Increments every cycle	RW	0x0000 1C5E

Table 13-50. GCOUTSTANDINGREADS0

Address Offset	0x0000 043C		
Physical Address	0x5900 043C	**Instance**	BB2D
Description	Number of outstanding reads per client in multiples of 8 bytes.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMU								FE								PEZ								PEC							

Bits	Field Name	Description	Type	Reset
31:24	MMU	Number of outstanding MMU reads in multiples of 8 bytes	R	0x00
23:16	FE	Number of outstanding FE reads in multiples of 8 bytes	R	0x00
15:8	PEZ	Number of outstanding PEZ reads in multiples of 8 bytes	R	0x00
7:0	PEC	Number of outstanding PEC reads in multiples of 8 bytes	R	0x00

Table 13-51. GCOUTSTANDINGREADS1

Address Offset	0x0000 0440		
Physical Address	0x5900 0440	**Instance**	BB2D
Description	Number of outstanding reads per client in multiples of 8 bytes.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOTAL								FC								TX								RA							

Bits	Field Name	Description	Type	Reset
31:24	TOTAL	This field keeps the value of total read requests or total requested data (in 64 bits) depending on the value of AQMEMORYDEBUG[19] LIMIT_CONTROL register field.	R	0x00
23:16	FC	Number of outstanding FC reads in multiples of 8 bytes	R	0x00

Bits	Field Name	Description	Type	Reset
15:8	TX	Number of outstanding TX reads in multiples of 8 bytes	R	0x00
7:0	RA	Number of outstanding RA reads in multiples of 8 bytes	R	0x00

Table 13-52. GCOUTSTANDINGWRITES

Address Offset	0x0000 0444		
Physical Address	0x5900 0444	**Instance**	BB2D
Description	Number of outstanding writes per client.		
Type	R		

31:24	23:16	15:8	7:0
TOTAL	FC	PEZ	PEC

Bits	Field Name	Description	Type	Reset
31:24	TOTAL	This field keeps the value of total write requests or total requested data (in 64 bits) depending on the value of AQMEMORYDEBUG[19] LIMIT_CONTROL register field.	R	0x00
23:16	FC	Number of outstanding FC writes in multiples of 8 bytes	R	0x00
15:8	PEZ	Number of outstanding PEZ writes in multiples of 8 bytes	R	0x00
7:0	PEC	Number of outstanding PEC writes in multiples of 8 bytes	R	0x00

Table 13-53. GCDEBUGSIGNALSRA

Address Offset	0x0000 0448		
Physical Address	0x5900 0448	**Instance**	BB2D
Description	32 bit debug signal from RA.		
Type	R		

31:0
SIGNAL

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL1[19:16] RA: 0x0: Valid pixel count. 0x1: Total quad count (after EEZ). 0x2: Valid quad count (after EZ and EEZ). 0x3: Total primitive count. 0x4: Various signals from input stage. See GC320 spec for details. 0x5: Various signals from input stage. See GC320 spec for details. 0x6: Various signals from render pipe. See GC320 spec for details. 0x7: Various signals from render cache. See GC320 spec for details. 0x8: Various signals from raster engine. See GC320 spec for details. 0x9: Cache miss count (in the pipeline). 0xA: Cache miss count (in the prefetcher). 0xB: EEZ culled quads. 0xF: Signature = 0x12344321.	R	0x0000 0000

Table 13-54. GCDEBUGSIGNALSTX

Address Offset	0x0000 044C		
Physical Address	0x5900 044C	**Instance**	BB2D
Description	32 bit debug signal from TX.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL1[27:24] TX : 0x0: Total bilinear texture requests. 0x1: Total trilinear texture requests. 0x2: Total discarded texture requests. 0x3: Total texture requests. 0x4: Various signals from input stage. See GC320 spec for details. 0x5: Memory read count. 0x6: Memory read count in 8B. 0x7: Cache miss count (in the pipeline). 0x8: Total hitting texels (in pre-fetcher). 0x9: Total missing texels (in pre-fetcher). 0xF: Signature = 0x12211221.	R	0x0000 0000

Table 13-55. GCDEBUGSIGNALSFE

Address Offset	0x0000 0450		
Physical Address	0x5900 0450	**Instance**	BB2D
Description	32 bit debug signal from FE.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL		R	0x0000 0000

Table 13-56. GCDEBUGSIGNALSPE

Address Offset	0x0000 0454		
Physical Address	0x5900 0454	**Instance**	BB2D
Description	32 bit debug signal from PE.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL0[19:16] PE: 0x0: pixel count killed by color pipe 0x1: pixel count killed by depth pipe 0x2: pixel count drawn by color pipe 0x3: pixel count drawn by depth pipe 0x4: debug signals for 3d_io, 2d_filter, 2d_fsm 0x5: debug signals for cache2d_cntrl 0x6: debug signals for cache2d_tag_alloc 0x7: debug signals for cache3d_c_cntrl, cache3d_c_tag_alloc 0x8: debug signals for cache3d_z_cntrl, cache3d_z_tag_alloc 0x9: debug signals for pref_2d, pref_3d 0xA : debug signals for cmd_state 0xB: 2d pixel count drawn by 2d pipe 0xF: Signature = 0xBABEF00D.	R	0x0000 0000

Table 13-57. GCDEBUGSIGNALSDE

Address Offset	0x0000 0458

Table 13-57. GCDEBUGSIGNALSDE (continued)

Physical Address	0x5900 0458	**Instance**	BB2D
Description	32 bit debug signal from DE.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL		R	0x0000 0000

Table 13-58. GCDEBUGSIGNALSSH

Address Offset	0x0000 045C		
Physical Address	0x5900 045C	**Instance**	BB2D
Description	32 bit debug signal from SH.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL0[27:24] SH. Please refer to GC320 spec for bit position information for all the signals 0x0 : interface signals for debug 0x1 : Instruction Sequencing and vertex input state machine 0x2 : vertex input/output buffer full/empty. Context PC. Physical page valid 0x3 : vertex/pixel, output attribute counts. Some interface signals 0x4 : Shader cycle count, for determining the shader clock 0x5 : Current pixel XY value 0x6 : Last pixels XY value 0x7 : Total pixel instructions executed 0x8 : Total pixels shaded 0x9 : Total vertex instructions executed 0xA : Total vertices shaded 0xB : Total vertex branch instructions 0xC : Total vertex texture instructions 0xD : Total pixel branch instructions 0xE : Total pixel texture instructions 0xF : Reserved signature 0xDEADBEEF	R	0x0000 0000

Table 13-59. GCDEBUGSIGNALSPA

Address Offset	0x0000 0460		
Physical Address	0x5900 0460	**Instance**	BB2D
Description	32 bit debug signal from PA.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL1[3:0] PA: 0x0: Various signals from input stage. See GC320 spec for details. 0x1: Various signals from input stage. See GC320 spec for details. 0x2: Various signals from input stage. See GC320 spec for details. 0x3: total vertex count 0x4: input primitive count 0x5: output primitive count 0x6: depth clipped primitive count 0x7: trivial rejected primitive count 0x8: culled primitive count 0xF: Signature = 0x0000AAAA	R	0x0000 0000

Table 13-60. GCDEBUGSIGNALSSE

Address Offset	0x0000 0464		
Physical Address	0x5900 0464	**Instance**	BB2D
Description	32 bit debug signal from SE.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL1[11:8] SE: 0x0: culled triangles count. 0x1: culled lines count. 0x2: [31:18] goto signals, [17:8] main state machine state, [7:0] output state machine state. See GC320 spec for details. 0x3: [31:22] unused, [21] early_isTriangle, [20] isTriangle, [19] increment_pc_e0, [18:14] jump_to_signals. See GC320 spec for details. [13:12] max_x_p_e2, [11:10] mid_x_p_e2, [9:8] min_x_p_e2, [7:6] max_y_p_e2, [5:4] mid_y_p_e2. See GC320 spec for details. [3:2] min_y_p_e2, [1:0] min_z_p_e2. See GC320 spec for details. 0x4: area_e2. See GC320 spec for details. 0x5: x0_e2. See GC320 spec for details. 0x6: x1_e2. See GC320 spec for details. 0x7: x2_e2. See GC320 spec for details. 0x8: y0_e2. See GC320 spec for details. 0x9: y1_e2. See GC320 spec for details. 0xA: y2_e2. See GC320 spec for details. 0xB: init_y_e2. See GC320 spec for details. 0xC: init_y_e2. See GC320 spec for details. 0xD: y2_e2. See GC320 spec for details. 0xE: y2_e2. See GC320 spec for details. 0xF: Signature = 0x5E5E5E5E.	R	0x0000 0000

Table 13-61. GCDEBUGSIGNALSMC

Address Offset	0x0000 0468		
Physical Address	0x5900 0468	**Instance**	BB2D
Description	32 bit debug signal from MC.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL2[3:0] MC: 0x0: Various signals from FC block. See GC320 spec for details. 0x1: Total read req in terms of 8B from pipeline. 0x2: Total read req in terms of 8B sent out from the subsystem. 0x3: Total write req in terms of 8B from pipeline. 0xF: Signature = 0x12345678.	R	0x0000 0000

Table 13-62. GCDEBUGSIGNALSHI

Address Offset	0x0000 046C		
Physical Address	0x5900 046C	**Instance**	BB2D
Description	32 bit debug signal from HI.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNAL																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNAL	Signals according to GCDEBUGCONTROL2[11:8] HI: 0x0: Number of cycles AXI read request is stalled. 0x1: Number of cycles AXI write request is stalled. 0x2: Number of cycles AXI write data is stalled. 0xF: Signature = 0xAAAAAAAA	R	0x0000 0000

Table 13-63. GCDEBUGCONTROL0

Address Offset	0x0000 0470		
Physical Address	0x5900 0470	**Instance**	BB2D
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SH				RESERVED				PE				RESERVED				DE				RESERVED				FE			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	SH	Selects which set of 32 bit data to get from SH. Resets the counters if set to 0xF	RW	0x0
23:20	RESERVED		R	0x0
19:16	PE	Selects which set of 32 bit data to get from PE. Resets the counters if set to 0xF	RW	0x0
15:12	RESERVED		R	0x0
11:8	DE	Selects which set of 32 bit data to get from DE. Resets the counters if set to 0xF	RW	0x0
7:4	RESERVED		R	0x0
3:0	FE	Selects which set of 32 bit data to get from FE. Resets the counters if set to 0xF	RW	0x0

Table 13-64. GCDEBUGCONTROL1

Address Offset	0x0000 0474		
Physical Address	0x5900 0474	**Instance**	BB2D
Description			
Type	RW		

31–28	27–24	23–20	19–16	15–12	11–8	7–4	3–0
RESERVED	TX	RESERVED	RA	RESERVED	SE	RESERVED	PA

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	TX	Selects which set of 32 bit data to get from TX. Resets the counters if set to 0xF	RW	0x0
23:20	RESERVED		R	0x0
19:16	RA	Selects which set of 32 bit data to get from RA. Resets the counters if set to 0xF	RW	0x0
15:12	RESERVED		R	0x0
11:8	SE	Selects which set of 32 bit data to get from SE. Resets the counters if set to 0xF	RW	0x0
7:4	RESERVED		R	0x0
3:0	PA	Selects which set of 32 bit data to get from PA. Resets the counters if set to 0xF	RW	0x0

Table 13-65. GCDEBUGCONTROL2

Address Offset	0x0000 0478		
Physical Address	0x5900 0478	**Instance**	BB2D
Description			
Type	RW		

31–12	11–8	7–4	3–0
RESERVED	HI	RESERVED	MC

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0 0000
11:8	HI	Selects which set of 32 bit data to get from HI. Resets the counters if set to 0xF	RW	0x0
7:4	RESERVED		R	0x0
3:0	MC	Selects which set of 32 bit data to get from MC. Resets the counters if set to 0xF	RW	0x0

Table 13-66. GCDEBUGCONTROL3

Address Offset	0x0000 047C		
Physical Address	0x5900 047C	**Instance**	BB2D
Description			
Type	RW		

31–12	11–8	7–4	3–0
RESERVED	PROBE1	RESERVED	PROBE0

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0 0000

Bits	Field Name	Description	Type	Reset
11:8	PROBE1	Selects which module's output will be put in the MSB 32 bits of 64 bit debug signal. 0x0: FE 0x1: DE 0x2: PE 0x3: SH 0x4: PA 0x5: SE 0x6: RA 0x7: TX 0x8: MC	RW	0x0
7:4	RESERVED		R	0x0
3:0	PROBE0	Selects which module's output will be put in the LSB 32 bits of 64 bit debug signal. 0x0: FE 0x1: DE 0x2: PE 0x3: SH 0x4: PA 0x5: SE 0x6: RA 0x7: TX 0x8: MC	RW	0x0

Table 13-67. GCBUSCONTROL

Address Offset	0x0000 0480		
Physical Address	0x5900 0480	**Instance**	BB2D
Description	Shows which features are enabled in the subsystem. 0 : NONE 1 : AVAILABLE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							FCC	TX	FC	MMU	RESERVED	FE	RESERVED	PEZ	PEC

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x00 0000
8	FCC	Select the return bus for FCC	RW	0
7	TX	Select the return bus for TX	RW	1
6	FC	Select the return bus for FC-Depth	RW	0
5	MMU	Select the return bus for MMU	RW	1
4	RESERVED		R	0
3	FE	Select the return bus for FE	RW	1
2	RESERVED		R	0
1	PEZ	Select the return bus for PEZ	RW	0
0	PEC	Select the return bus for PEC	RW	0

Table 13-68. GCREGENDIANNESS0

Address Offset	0x0000 0484		
Physical Address	0x5900 0484	**Instance**	BB2D
Description			
Type	RW		

31–0
WORD_SWAP

Bits	Field Name	Description	Type	Reset
31:0	WORD_SWAP	Flip the words of 32 bit data. 0x12345678 becomes 0x56781234	RW	0x0000 0000

Table 13-69. GCREGENDIANNESS1

Address Offset	0x0000 0488		
Physical Address	0x5900 0488	**Instance**	BB2D
Description			
Type	RW		

31–0
BYTE_SWAP

Bits	Field Name	Description	Type	Reset
31:0	BYTE_SWAP	Flip the bytes of 16 bit data. 0x12345678 becomes 0x34127856	RW	0x0000 0000

Table 13-70. GCREGENDIANNESS2

Address Offset	0x0000 048C		
Physical Address	0x5900 048C	**Instance**	BB2D
Description			
Type	RW		

31–0
BIT_SWAP

Bits	Field Name	Description	Type	Reset
31:0	BIT_SWAP	Flip the bits of 8 bit data. 0x12345678 becomes 0x84C2A6E1	RW	0x0000 0000

Table 13-71. GCREGDRAWPRIMITIVESTARTTIMESTAMP

Address Offset	0x0000 0490		
Physical Address	0x5900 0490	**Instance**	BB2D
Description			
Type	R		

31–0
START_TIME

Bits	Field Name	Description	Type	Reset
31:0	START_TIME	32-bit timestamp when PE received draw_primitive_start command	R	0x0000 0000

Table 13-72. GCREGDRAWPRIMITIVEENDTIMESTAMP

Address Offset	0x0000 0494		
Physical Address	0x5900 0494	**Instance**	BB2D
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

END_TIME

Bits	Field Name	Description	Type	Reset
31:0	END_TIME	32-bit timestamp when PE received draw_primitive_end command	R	0x0000 0000

Table 13-73. GCREGCONTROL0

Address Offset	0x0000 0558		
Physical Address	0x5900 0558	**Instance**	BB2D
Description	Composition trigger.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISC1						OUTSTANDING_READS_PER_CHANNEL										MISC0												ENABLE_UNALIGNED_WRITE_MERGE	ENABLE_WRITE_MERGE	ENABLE_UNALIGNED_MERGE	ENABLE_READ_MERGE

Bits	Field Name	Description	Type	Reset
31:26	MISC1		RW	0x00
25:16	OUTSTANDING_READS_PER_CHANNEL		RW	0x080
15:4	MISC0		RW	0x000
3	ENABLE_UNALIGNED_WRITE_MERGE		RW	0
2	ENABLE_WRITE_MERGE		RW	1
1	ENABLE_UNALIGNED_MERGE		RW	0
0	ENABLE_READ_MERGE		RW	1

Table 13-74. AQCMDBUFFERADDR

Address Offset	0x0000 0654		
Physical Address	0x5900 0654	**Instance**	BB2D
Description	Base address for the command buffer. The address must be 64-bit aligned and it is always physical. To use addresses above 0x8000_0000, program AQMemoryFE with the appropriate offset. Also, this register cannot be read. To check the value of the current fetch address use AQFEDEBUGCURCMDADR.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	ADDRESS																														

Bits	Field Name	Description	Type	Reset
31	TYPE	0: SYSTEM 1: VIRTUAL_SYSTEM	W	0

Bits	Field Name	Description	Type	Reset
30:0	ADDRESS	ADDRESS	W	0x0000 0000

Table 13-75. AQCMDBUFFERCTRL

Address Offset	0x0000 0658		
Physical Address	0x5900 0658	**Instance**	BB2D
Description	Command buffer control		
Type	W		

31:22	21:20	19:17	16	15:0
RESERVED	ENDIAN_CONTROL	RESERVED	ENABLE	PREFETCH

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		W	0x000
21:20	ENDIAN_CONTROL	Endian control 0: NO_SWAP 1: SWAP_WORD 2: SWAP_DWORD	W	0x0
19:17	RESERVED		W	0x0
16	ENABLE	Command buffer 0: DISABLE 1: ENABLE	W	0
15:0	PREFETCH	Number of 64-bit words to fetch from the command buffer.	W	0x0000

Table 13-76. AQFESTATUS

Address Offset	0x0000 065C		
Physical Address	0x5900 065C	**Instance**	BB2D
Description	FE status		
Type	R		

31:1	0
RESERVED	COMMAND_DATA

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	COMMAND_DATA	Status of the command parser. 0: Idle 1: Busy	R	0

Table 13-77. AQFEDEBUGCURCMDADR

Address Offset	0x0000 0664		
Physical Address	0x5900 0664	**Instance**	BB2D
Description	This is the command decoder address. The address is always physical so the MSB should always be 0.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUR_CMD_ADR																													RESERVED		

Bits	Field Name	Description	Type	Reset
31:3	CUR_CMD_ADR		R	0x0000 0000
2:0	RESERVED		R	0x0

Chapter 14

Interconnect

This chapter describes the device interconnect.

Note

The level 3 (L3) interconnect is an instantiation of the NoC interconnect from Arteris, Inc. Arteris is a registered trademark of Arteris, Inc.

This document contains materials that are ©Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris Reference manuals, or contact Arteris, Inc.

NoC is an abbreviation for Network On Chip.

Note

The level 4 (L4) interconnects are instantiations of the Sonics3220™ interconnect from Sonics, Inc.

This document contains materials that are ©2003-2010 Sonics, Inc., and that constitute proprietary information of Sonics, Inc.

SonicsLX and Sonics3220, are trademarks or registered trademarks of Sonics, Inc. All such materials and trademarks are used under license from Sonics, Inc. For additional information, see the SonicsLX or Sonics3220 Reference manuals, or contact Sonics, Inc.

SLX is an abbreviation for SonicsLX®.

14.1 Interconnect Overview....................2967
14.2 L3_MAIN Interconnect....................2970
14.3 L4 Interconnects....................3126

14.1 Interconnect Overview

14.1.1 Terminology

The following terminology is critical to understanding the interconnect:

- Initiator: Module able to initiate read and write requests to the chip interconnect (typically: processors, DMA, etc.).
- Target: Unlike an initiator, a target module cannot generate read/write requests to the chip interconnect, but it can respond to these requests. However, it may generate interrupts or a DMA request to the system (typically: peripherals, memory controllers).

Note

A module can have several separate ports; therefore, a module can be an initiator and a target.

- Agent: Each connection of one module to one interconnect is done using an agent, which is an adaptation (sometimes configurable) between the module and the interconnect. A target module is connected by a target agent (TA), and an initiator module is connected by an initiator agent (IA).
- Interconnect: The decoding, routing, and arbitration logic that enables the connection between multiple initiator modules and multiple target modules connected on it. Quality of service (QoS) is guaranteed.
- Register target (RT): Special TA used to access the interconnect internal configuration registers
- Data-flow signal: Any signal that is part of a clearly identified transfer or data flow (typically: command, address, byte enables, etc.). Signal behaviour is defined by the protocol semantics.
- Sideband signal: Any signal whose behaviour is not associated to a precise transaction or data flow.
- Out-of-band error: Any signal whose behaviour is associated to a device error-reporting scheme, as opposed to in-band errors.

Note

Interrupt requests and DMA requests are not routed by the interconnect in the device.

- Firewall: A programmable feature integrated in a target agent or L4 interconnect to prevent unauthorized access to or from a module. A firewall can be configured using three criteria:
  - Initiator requesting access
  - Address space access
  - Type of access
- ConnID: Any transaction in the system interconnect is tagged by an in-band qualifier ConnID, which uniquely identifies the initiator at a given interconnect point. A ConnID is transmitted in band with the request and is used for firewall and error-logging mechanism.
- Firewall comparison mechanism: A comparison made in the firewall between access in-band qualifiers and access permissions that are programmed in the firewall configuration registers. If the comparison is successful, access is allowed; otherwise, access is denied.
- MCmd qualifier: Command bus that indicates the type of transfer requested. Table 14-1 lists the commands encoded. For information specific to L3 Interconnect error logging, see Table 14-20 *L3 Firewall Error-Logging Registers*

Table 14-1. MCmd Qualifier Description

MCmd[2:0]	Transaction Type
0 0 0	Idle
0 0 1	Write
0 1 0	Read
0 1 1	ReadEx
1 0 0	Read link
1 0 1	Write nonposted

Table 14-1. MCmd Qualifier Description (continued)

MCmd[2:0]	Transaction Type
1 1 0	Write conditional
1 1 1	Write broadcast

- MReqInfo qualifier: Four MReqInfo qualifiers describe the access during the use of the firewall comparison mechanism, as described in Table 14-2.

Table 14-2. MReqInfo Qualifier Description

Qualifiers	Description
MReqType	0: Data access
	1: Opcode fetch
MReqSupervisor	0: User mode
	1: Supervisor mode
MReqDebug	0: Functional access
	1: Debug access

- Register that configures the combination of the MReqInfo, allowing access permission to the target module (TM) based on the MReqInfo in-band qualifier values.
- SError: Target that indicates an error condition to the initiator.
- SResp qualifier: Response from the target to the initiator concerning the transaction, as described in Table 14-3.

Table 14-3. SResp Qualifier Description

SResp[1:0]	Description
0 0	No response
0 1	Data valid/accept
1 0	Not used
1 1	Error

- MTagID: Interconnect qualifier generated by the L3_MAIN masters which purpose is to identify whether reordering is allowed or not relative to other transactions. Strong ordering is ensured by using same MTagID values between transactions. Reordering is allowed by using different MTagID values between transactions. The MTagID values may or may not be changed by the interconnect, but the intended reordering restriction must match what came from the master. In other words, the interconnect allocates dynamically MTagID values in such a way that the intended reordering restrictions from each master are honored.

14.1.2 Architecture Overview

The device memory hierarchy includes four levels:

- L1 is internal to the CPUs. It concerns data exchange with the internal Level1 cache memory subsystem, and it is the closest memory to the microprocessor unit (MPU) core and the IVA core.
- L2 is included in the IPU subsystem and the MPU subsystem.
- The chip-level interconnect consists of one L3 interconnect and five L4 interconnects. It enables communication among the modules and subsystems in the device.

Figure 14-1 shows an overview of the L3 and L4 interconnect architecture.

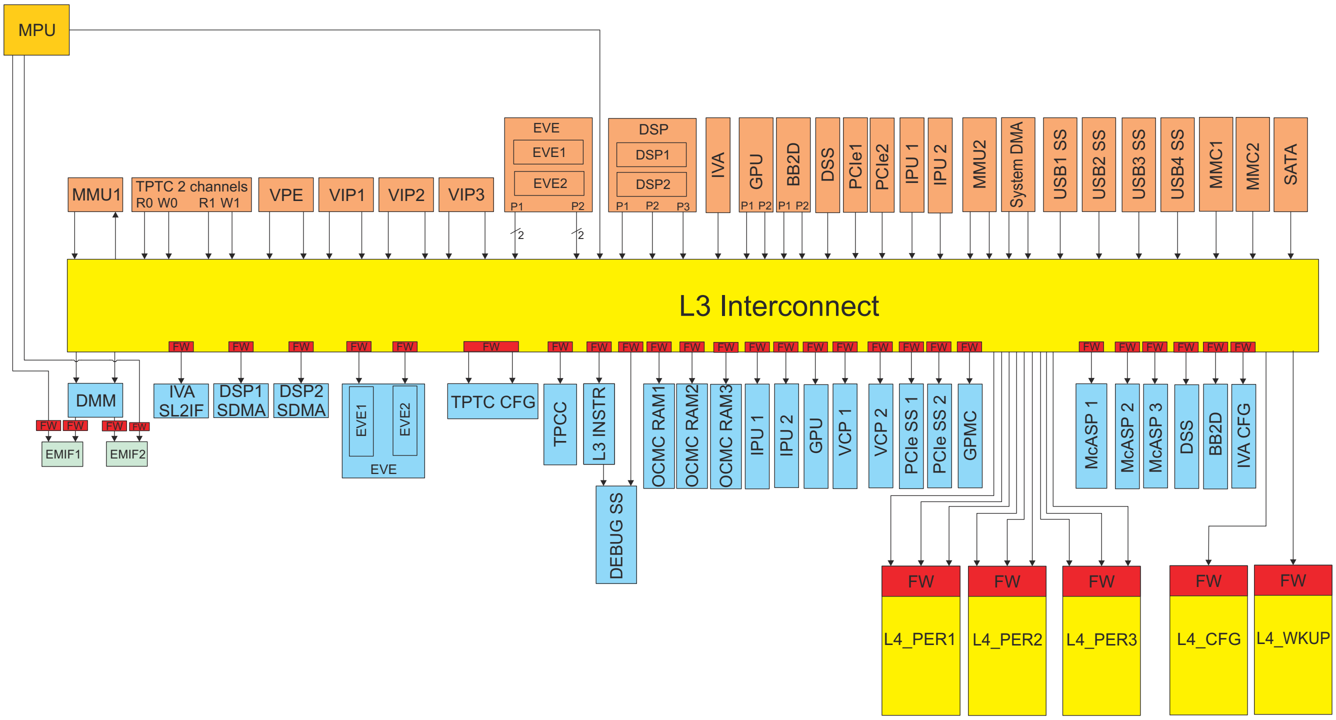


Figure 14-1. Interconnect Overview

- L3 handles many types of data transfers, especially exchanges with system-on-chip/external memories. L3 transfers data with a maximum width of 128 bits from the initiator to the target. The L3 interconnect is a little-endian platform
- The L4 is composed of the following:
  – L4_CFG: Includes the majority of the firewall configuration interface for level 3 (L3) system modules and peripheral interconnect;
  – L4_PER: Includes the main peripherals that require system direct memory access (sDMA) access. L4_PER has three instances L4_PER1, L4_PER2 and L4_PER3. Each of these three instances has three ports connecting it to L3_MAIN interconnect:
    - L4_PER1_P1, L4_PER1_P2, L4_PER1_P3
    - L4_PER2_P1, L4_PER2_P2, L4_PER2_P3
    - L4_PER3_P1, L4_PER3_P2, L4_PER3_P3

    Through L3_MAIN interconnect, different initiators can access each of these L4_PERx_Pi ports. For information regarding which initiator accesses which L4_PERx_Pi port, see Figure 14-3.
  – L4_WKUP: Includes peripherals attached to the WKUP power domain.

Modules are connected to the interconnect through an IA for the initiator module and a TA for target modules. Each module/subsystem connection is statically configured to tune the access, depending on the characteristics of the module.

To unauthorize a module or L4 interconnect access, some TAs include configurable firewalls (FWs). A firewall restricts or filters the accesses allowed to an initiator according to different access criteria. The firewalls can usually be configured by software.

The L3 and L4 interconnect default settings are fully functional; they enable all possible functional data paths and a minimal default protection setting.

14.2 L3_MAIN Interconnect

This section describes the L3_MAIN interconnect and its components. With the exception of register points, each component includes functions for the request and response networks.

14.2.1 L3_MAIN Interconnect Overview

The L3_MAIN interconnect links cores in a flexible topology that couples low power with high performance. Innovative physical structures and advanced protocols ensure bandwidth and latency to individual IP cores, providing dedicated connections between IP cores and logical connections over a shared interconnect.

The main features of the L3_MAIN interconnects are:

- NIUs: Master NIUs for the IAs and slave NIUs for the TAs
- A partially depleted cross-bar exchange network
- A special internal slave NIU for accessing L3_MAIN interconnect configuration registers
- QoS management for real-time hardware operators, while maintaining optimal memory latency for CPU access to memory resources
- True little-endian platform
- Transaction errors tracking and logging registers
- All signaling support for chip-level power-management infrastructure
- One interrupt line signaling transaction error
- One interrupt line for reporting statistical events on the L3_MAIN interconnect

Figure 14-2 shows an overview of the L3 interconnect and the peripherals attached to it.

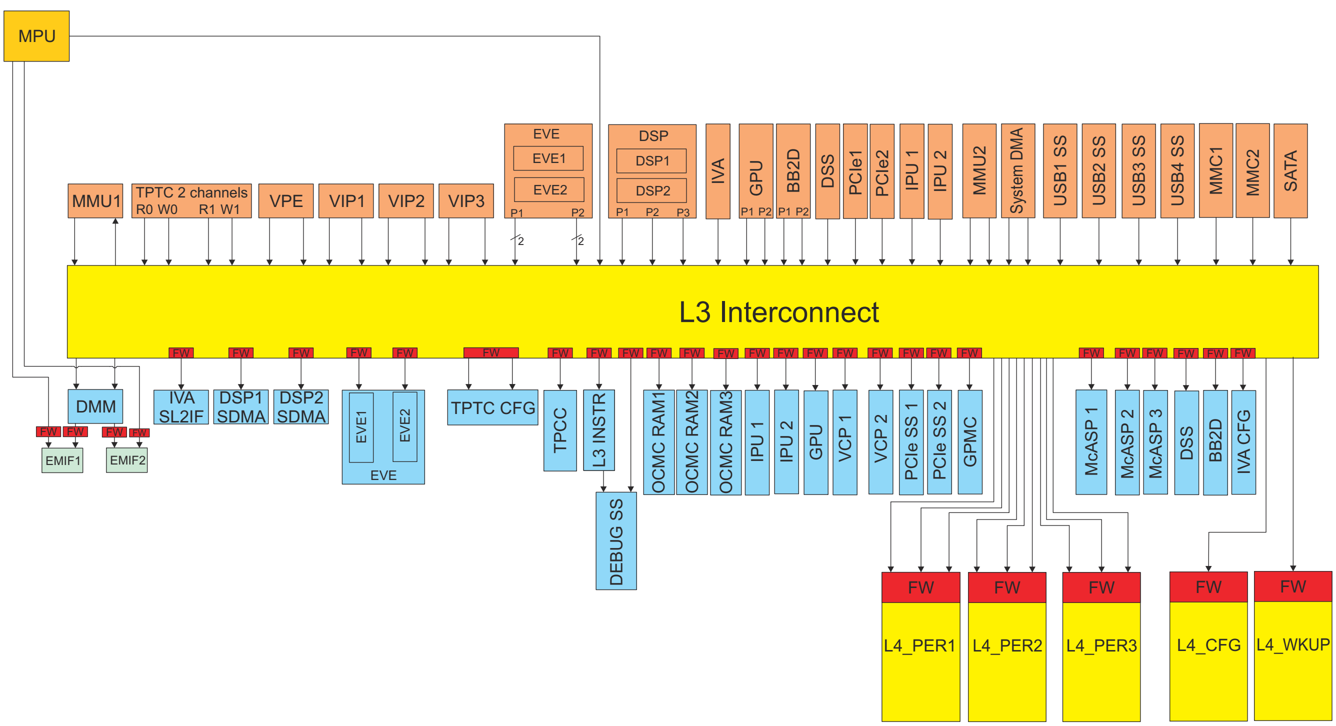


Figure 14-2. L3_MAIN Interconnect Overview

14.2.2 L3_MAIN Interconnect Integration

Table 14-4 through Table 14-6 summarize the integration of the module in the device.

Table 14-4. L3_MAIN Integration Attributes

Module Instance	Attributes
	Power Domain

Table 14-4. L3_MAIN Integration Attributes (continued)

L3_MAIN	PD_COREAON

Table 14-5. L3_MAIN Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
L3_MAIN	L3_CLK1	L3MAIN1_L3_GICLK	PRCM	Functional and interface clock
	L3_CLK2	L3INSTR_L3_GICLK	PRCM	Functional and interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
L3_MAIN	L3_CORE_RET_RST	CORE_PWRON_RET_RST	PRCM	Reset of L3_MAIN interconnect registers
	L3_CORE_RST	CORE_RST	PRCM	Reset of L3_MAIN interconnect

Table 14-6. L3_MAIN Hardware Requests

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
Interrupt Requests				
L3_MAIN	L3_MAIN_IRQ_DBG_ERR	IRQ_CROSSBAR_4	MPU_IRQ_9	Interrupt indicating debug error occurrence.
			DSP1_IRQ_35	
			DSP2_IRQ_35	
	L3_MAIN_IRQ_APP_ERR	IRQ_CROSSBAR_5	MPU_IRQ_10(1)	Interrupt indicating application error occurrence.
			DSP1_IRQ_36	
			DSP2_IRQ_36	
			EVE1_IRQ_4	
			EVE2_IRQ_4	
			IPU1_IRQ_46	
			IPU2_IRQ_46	
	L3_MAIN_IRQ_STAT_ALARM	IRQ_CROSSBAR_11	MPU_IRQ_16	Statistic collector alarm interrupt
			DSP1_IRQ_42	
			DSP2_IRQ_42	

(1) The L3_MAIN_IRQ_APP_ERR interrupt is directly mapped to the MPU_IRQ_10 line bypassing IRQ_CROSSBAR_5. In all other cases IRQ_CROSSBAR_5 is used to map L3_MAIN_IRQ_APP_ERR to the corresponding INTC.

Note

The “**Default Mapping**” column in Table 14-6 *L3_MAIN Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.
For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

14.2.3 L3_MAIN Interconnect Functional Description

14.2.3.1 Module Use in L3_MAIN Interconnect

The L3_MAIN interconnect network components have ConnID values for each master NIU or slave NIU. The ID uniquely identifies the master NIU or the slave NIU for an interconnect transfer. The interconnect uses ConnIDs for a number of purposes, including the following:

- Slave NIUs for error logging
- Power disconnect slave NIU for error logging
- Eight FLAGMUXes to mask interrupts
- STATCOLL for configuring and monitoring: The STATCOLL components compute the traffic statistics within a user-defined window and periodically report to the user through the debug interface.
- Bandwidth regulator for configuration

14.2.3.2 Module Distribution

Master NIUs and slave NIUs provide the interface to connect the different modules to their associated interconnect.

Table 14-7 and Table 14-8 list all the modules and subsystems with their associated agents. The agents are listed for each L3_MAIN interconnect domain.

14.2.3.2.1 L3_MAIN Interconnect Agents

Any initiator or target core is connected to the L3_MAIN interconnect through an NIU. NIUs act as entry points to the L3_MAIN interconnect, and also include various programming registers. Table 14-7 lists the supported master NIU ports.

Table 14-7. Master NIUs

Master NIU	Description
MPU_INIT	MPU initiator port. One 64b OCP initiator port, and two 128b ports connected directly to the EMIF, bypassing L3
DSP1_INIT	DSP Initiator port. One 128b MDMA interconnect initiator port (used also for cache requests) One EDMA initiator port per instance
DSP2_INIT	DSP Initiator port. One 128b MDMA interconnect initiator port (used also for cache requests) One EDMA initiator port per instance
EVE1_INIT	Embedded Vision Engine 1 (EVE1) initiator port. Two 128b interconnect initiator ports (P1/P2)
EVE2_INIT	Embedded Vision Engine 1 (EVE2) initiator port. Two 128b interconnect initiator ports (P1/P2)
IVA_INIT	Image and video accelerator (IVA) initiator port. One 128b initiator port
GPU_P1_INIT	GPU 128b initator port 1
GPU_P2_INIT	GPU 128b initiator port 2
BB2D_P1_INIT	2D Graphics accelerator 128b initiator port 1
BB2D_P2_INIT	BB2D 128b initiator port 2
DSS_INIT	Display SubSystem initiator port. One 128b initiator port
VIP1_INIT	VIP initiator ports. Two 128b initiator ports
VIP2_INIT	VIP initiator ports. Two 128b initiator ports
VIP3_INIT	VIP initiator ports. Two 128b initiator ports
VPE_P1_INIT	VPE 128b initiator port 1
VPE_P2_INIT	VPE 128b initiator port 2
PCIE1_INIT	PCIe 64b initiator port
PCIE2_INIT	PCIe 64b initiator port

Table 14-7. Master NIUs (continued)

Master NIU	Description
TPTC1_INIT	EDMA_TC1 initiator port. Two 128b initiator ports(one read port and one write port)
TPTC2_INI	EDMA_TC2 initiator port. Two 128b initiator ports(one read port and one write port)
MMU1_INIT	MMU initiator port.One 128b initiator port
MMU2_INIT	MMU initiator port. One 128b initiator port
IPU1_INIT	IPU initiator port. One 64b initiator port
IPU2_INIT	IPU initiator port. One 64b initiator port
DMA_SYSTEM_INIT	Two 64b initiator ports per instance (one read port and one write port)
GMAC_SW_INIT	GMAC_SW 32b initiator port
MMC1_INIT	MMC initiator port. One 32b initiator port
MMC2_INIT	MMC2 32b initiator port
SATA_INIT	SATA 32b initiator port
MLBSS_INIT	MLB 32b initiator port
USB1_INIT	USB3.0 initiator port. One 64b OTG initiator port
USB2_INIT	USB2.0 initiator port. One 64b OTG initiator port
USB3_INIT	USB2.0 initiator port. One 64b OTG initiator port
USB4_INIT	USB2.0 initiator port. One 64b OTG initiator port
IEEE1500_INIT	IEEE1500 32b initiator port
DEBUGSS_INIT	Debug subsystem 32b initiator port

Table 14-8 lists the supported slave NIU ports.

Table 14-8. Slave NIUs

Slave NIU	Description
DSP_TARG	DSP 128b target port per instance
DMM_P1_TARG	Dynamic memory management128b target port 1
DMM_P2_TARG	Dynamic memory management 128b target port 2
IVA_CONFIG_TARG	Video accelerator subsystem 32b configuration target port
IVA_SL2IF_TARG	Video accelerator subsystem 128b SL target port
L4_CFG_TARG	L4 CFG 32b target port
L4_WKUP_TARG	L4 WKUP 32b target port
TPTC_P1_TARG	TPTC 32b target port 1
TPTC_P2_TARG	TPTC 32b target port 2
TPCC_TARG	Third party channel controller (TPCC) 32b target port
OCMC_RAM1_TARG	On-chip memory controller 128b target port 1
OCMC_RAM2_TARG	On-chip memory controller 128b target port 2
OCMC_RAM3_TARG	On-chip memory controller 128b target port 3
PCIe1/2_TARG	PCIe1/2 64b target port
GPU_TARG	3D graphics accelerator 64b target port
IPU_P1_TARG	DUAL Cortex M4 subsystem 64b target port 1
IPU_P2_TARG	DUAL Cortex M4 subsystem 64b target port2
VCP1_TARG	VCP 64b target port 1
VCP2_TARG	VCP 64b target port 2
GPMC_TARG	General-purpose memory controller target port

Table 14-8. Slave NIUs (continued)

Slave NIU	Description
L4_PER/1/2/3_TARG	L4 interconnect peripherals 32b initiator port
MCASP1/2/3_TARG	McASP 32b target ports
DSS_TARG	Display subsystem 64b target port
BB2D_TARG	2D graphics accelerator (BB2D) 32b target port
QSPI_TARG	QSPI 32b target port
EVE1_TARG	Embedded vision engine (EVE) 128b target port
EVE2_TARG	EVE2 128b target port
MMU1_TARG	Memory management unit (MMU) 128b target port 1
MMU2_TARG	MMU 128b target port 2
L3_INSTR_TARG	L3 instrumentation 32b target port
DEBUGSS_TARG	Debug subsystem 32b target port

14.2.3.2.2 L3_MAIN Connectivity Matrix

The L3 interconnect is divided into two clock domains L3_CLK1 and L3_CLK2. CLK1 domain is further splitted into two sub groups:

- L3_CLK1_1: Low-power domain
- L3_CLK1_2: Peripherals and multimedia
- L3_CLK2: Instrumentation (debug)

The two clock elements (CLK1 and CLK2) are implemented in a different clock domain.

14.2.3.2.2.1 Clock Domain Mapping of the L3_MAIN Interconnect Modules

Each clock domain (CLK1 and CLK2) has it own host, flag mux, slave NIUs, and bandwidth regulators. Table 14-9 lists the relationships between these domains and these elements.

Table 14-9. L3_MAIN Clock Domains and Elements

Clock Domain	Elements
	HOST_CLK1_1
	L4_WKUP
	VCP1
	VCP2
	QSPI
	L4_PER3_P1
	L4_PER3_P2
	L4_PER3_P3
	L4_PER1_P1
	L4_PER1_P2
	L4_PER1_P3
	L4_PER2_P1
	L4_PER2_P2
	L4_PER2_P3
	McASP1
	McASP2
	McASP3
	GPMC
	L4_CFG
	IPU1

Table 14-9. L3_MAIN Clock Domains and Elements (continued)

Clock Domain	Elements
L3_CLK1_1	IPU2
	DSP1 SDMA
	DSP2 SDMA
	EVE1
	EVE2
	DSS
	IVA SL2IF
	DMM_P1
	DMM_P2
	OCMC_RAM2
	OCMC_RAM3
	IVA CFG
	BB2D
	MMU1
	OCMC_RAM1
	TPCC
	TPTC1 CFG
	TPTC2 CFG
	MMU2
	PCIe 1
	PCIe 2
	GPU
L3_CLK1_2	HOST_CLK1_2
	FLAGMUX_CLK1_MERGE
	FLAGMUX_CLK1_1
	FLAGMUX_CLK1_2
	FLAGMUX_CLK1_TIMEOUT1
	FLAGMUX_CLK1_TIMEOUT2
	DSP1_EDMA_BW_REGULATOR
	DSP2_EDMA_BW_REGULATOR
	EVE1_TC0_BW_REGULATOR
	EVE2_TC0_BW_REGULATOR
	EVE1_TC1_BW_REGULATOR
	EVE2_TC1_BW_REGULATOR
	VPE_P1_BW_LIMITER
	VPE_P2_BW_LIMITER
	IVA_BW_REGULATOR
	BB2D_P1_BW_REGULATOR
	BB2D_P2_BW_REGULATOR
	EDMA_TC1_RD_BW_LIMITER
	EDMA_TC1_WR_BW_LIMITER
	EDMA_TC2_RD_BW_LIMITER
	EDMA_TC2_WR_BW_LIMITER
	MMU1_BW_LIMITER
	PCIESS1_BW_REGULATOR

Table 14-9. L3_MAIN Clock Domains and Elements (continued)

Clock Domain	Elements
	PCIESS2_BW_REGULATOR
	GPU_P1_BW_REGULATOR
	GPU_P2_BW_REGULATOR
	MMU2_BW_REGULATOR
L3_CLK2	HOST_CLK2_1
	DEBUGSS_CT_TBR_TARG
	L3_INSTR
	STATCOLL0
	STATCOLL1
	STATCOLL2
	STATCOLL3
	STATCOLL4
	STATCOLL5
	STATCOLL6
	STATCOLL7
	STATCOLL8
	STATCOLL9
	FLAGMUX_CLK2_1
	FLAGMUX_CLK2_TIMEOUT
	FLAGMUX_STATCOLLS

14.2.3.2.2.2

Figure 14-3 shows the functional paths between the L3_MAIN master NIUs and the L3_MAIN and L3 slave NIU agents. The functional paths in L3_MAIN are indicated by the following:

- A cell contains the letter C when a functional path exists.
- A cell is empty when a functional path does not exist.

TARGETS

INITIATORS	DMM P1	DMM P2	OCMC RAM1	OCMC RAM2	OCMC RAM3	GPMC	DSP1 SDMA	DSP2 SDMA	EVE1	EVE2	IPU1	IPU2	IVA1 SL2IF	MMU1	MMU2	QSPI	GPU	BB2D	TC1_EDMA	TC2_EDMA	TPCC_EDMA	IVA_CONFIG	DSS	PCIE SS1	PCIE SS2	VCP1	VCP2	MCASP1	MCASP2	MCASP3	L4_PER1_P1	L4_PER1_P2	L4_PER1_P3	L4_PER2_P1	L4_PER2_P2	L4_PER2_P3	L4_PER3_P1	L4_PER3_P2	L4_PER3_P3	L4_WKUP	L4_CFG	L3_MAIN_SN	L3_INSTR	DEBUGSS_CT_TBR
MPU	C		C	C	C	C	C	C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	C	C
DSP1 MDMA	C		C	C	C	C		C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	C	C
DSP1 DMA		C	C	C	C	C		C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C				C			C			C	C	C	C	C
DSP1 CFG									C	C									C	C	C										C													
DSP2 MDMA		C	C	C	C	C	C		C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	C	C
DSP2 DMA	C		C	C	C	C	C		C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C		C			C			C		C	C	C	C	C
DSP2 CFG									C	C									C	C	C																							
EVE1 P1	C		C	C	C	C	C	C		C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C		
EVE1 P2		C	C	C	C	C	C	C		C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C		
EVE2 P1		C	C	C	C	C	C	C	C		C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C		
EVE2 P2	C		C	C	C	C	C	C	C		C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C		
IPU1	C		C	C	C	C	C	C	C	C		C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	C	C
IPU2		C	C	C	C	C	C	C	C	C	C		C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	C	C
VIP1 P1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
VIP1 P2		C	C	C	C	C	C	C	C	C	C	C	C			C																												
VIP2 P1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
VIP2 P2		C	C	C	C	C	C	C	C	C	C	C	C			C																												
VIP3 P1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
VIP3 P2		C	C	C	C	C	C	C	C	C	C	C	C			C																												
VPE P1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
VPE P2		C	C	C	C	C	C	C	C	C	C	C	C			C																												
DSS	C		C	C	C	C	C	C	C	C	C	C	C			C																												
IVA		C	C	C	C	C	C	C	C	C	C	C				C																												
GPU P1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
GPU P2		C	C	C	C	C	C	C	C	C	C	C	C			C																												
BB2D P1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
BB2D P2		C	C	C	C	C	C	C	C	C	C	C	C			C																												
DMA_SYSTEM_RD	C		C	C	C	C	C	C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C		C
DMA_SYSTEM_WR	C		C	C	C	C	C	C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	
EDMA_TC0_RD	C		C	C	C	C	C	C	C	C	C	C	C	C		C	C	C		C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C		C
EDMA_TC0_WR	C		C	C	C	C	C	C	C	C	C	C	C	C		C	C	C		C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	
EDMA_TC1_RD		C	C	C	C	C	C	C	C	C	C	C	C	C		C	C	C	C		C	C	C	C	C	C	C	C	C	C		C			C			C		C	C	C		C
EDMA_TC1_WR		C	C	C	C	C	C	C	C	C	C	C	C	C		C	C	C	C		C	C	C	C	C	C	C	C	C	C		C			C			C		C	C	C	C	
MMU1	C		C	C	C	C	C	C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	
MLB		C	C	C	C	C	C	C	C	C	C	C	C			C																												
PCIE_SS1	C		C	C	C	C	C	C	C	C	C	C	C		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C			C
PCIE_SS2		C	C	C	C	C	C	C	C	C	C	C	C		C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C			C
MMU2		C	C	C	C	C	C	C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C			C
GMAC_SW		C	C	C	C	C	C	C	C	C	C	C	C			C																												
SATA	C		C	C	C	C	C	C	C	C	C	C	C			C																												
MMC1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
MMC2		C	C	C	C	C	C	C	C	C	C	C	C			C																												
USB1	C		C	C	C	C	C	C	C	C	C	C	C			C																												
USB2		C	C	C	C	C	C	C	C	C	C	C	C			C																												C
USB3		C	C	C	C	C	C	C	C	C	C	C	C			C																												
USB4	C		C	C	C	C	C	C	C	C	C	C	C			C																												C
CS_DAP	C		C	C	C	C	C	C	C	C	C	C	C			C	C	C	C	C	C	C	C	C	C	C	C	C	C	C			C			C			C	C	C	C	C	C

Figure 14-3. Connectivity Matrix

14.2.3.2.3 Master NIU Identification

A master NIU ID (ConnID) is assigned to every module in the device. The ID uniquely identifies the master NIU for an interconnect transfer (see Table 14-10). The interconnect uses ConnID values for a number of purposes, including:

- Master source identification for the protection mechanism
- Response route generation

- Firewall error logging
- L3_MAIN interconnect error logging

Table 14-10. ConnID Values

8-bit ConnID (hex)	4-bit ConnID (hex)	Master NIU
0	0	MPU
10	1	CS_DAP
20	2	DSP1 MDMA
24	2	DSP1 CFG
28	2	DSP1 DMA
2C	2	DSP2 DMA
30	3	DSP2 CFG
34	3	DSP2 MDMA
3A	3	IVA
42	4	EVE1 P1
46	4	EVE2 P1
60	6	IPU1
64	6	IPU2
68	6	DMA_SYSTEM RD
6A	6	DMA_SYSTEM WR
70	7	EDMA_TC1_WR
72	7	EDMA_TC1_RD
74	7	EDMA_TC2_WR
76	7	EDMA_TC2_RD
80	8	DSS
84	8	MLB
86	8	MMU1
88	8	PCIE1
8C	8	PCIE2
8E	8	MMU2
90	9	VIP1 P1
92	9	VIP1 P2
94	9	VIP2 P1
96	9	VIP2 P2
98	9	VIP3 P1
9A	9	VIP3 P2
9C	9	VPE P1
9E	9	VPE P2
A0	A	MMC1
A2	A	GPU P1
A4	A	MMC2
A6	A	GPU P2
A8	A	BB2D P1
AA	A	BB2D P2
AC	A	GMAC_SW
B0	B	USB1 (USB3.0_SS)
B4	B	USB2 (USB2_SS)
B8	B	USB3 (USB2_ULPI_SS1)

Table 14-10. ConnID Values (continued)

8-bit ConnID (hex)	4-bit ConnID (hex)	Master NIU
BC	B	USB4 (USB2_ULPI_SS2)
CC	C	SATA
D2	D	EVE1 P2
D6	D	EVE2 P2

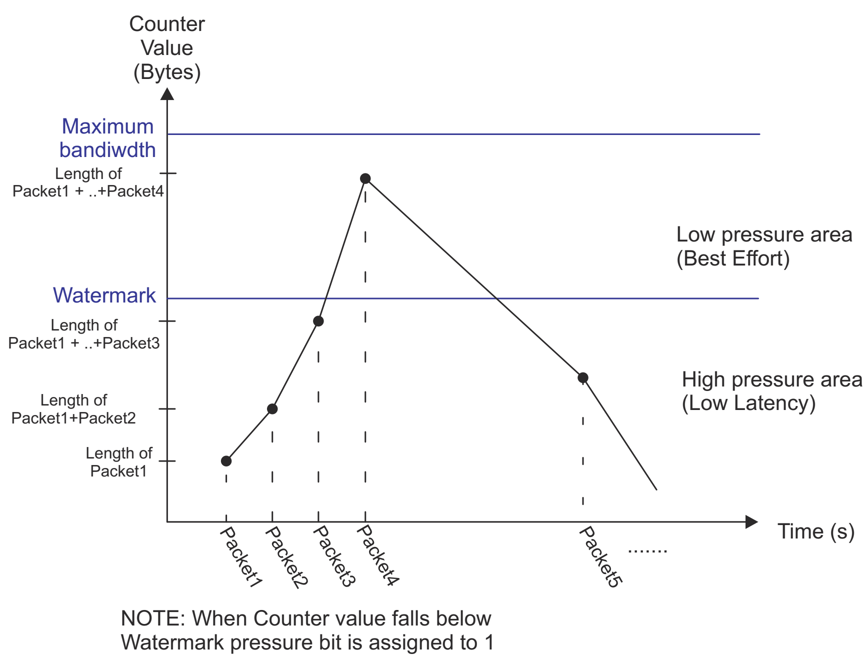
The 8-bit ConnID values are used by error decoding to distinguish the different initiators, see Section 14.2.3.8.5, *Example for Decoding Standard/Custom Errors Logged in L3_MAIN*. They are also used by the EMIF controller. The 4-bit ConnID values are used by the firewalls to allow or not an access to a slave NIU, see Table 14-18.

14.2.3.3 Bandwidth Regulators

The bandwidth regulators prevent master NIUs from consuming too much bandwidth of a link, or a slave NIU that is shared between several data flows: packets are then transported at a slower rate. The value of a bandwidth can be programmed in the bandwidth regulator. When the bandwidth is below the programmed value, the pressure bit is set to 1, giving priority to this master. When the bandwidth is above the programmed value, the pressure bit is set to 0 and the concerned master has the same weight as others.

A counter is used to store the sum of data lengths (in bytes) of each packet passing through the bandwidth regulator, and a value equal to the expected bandwidth is subtracted from the counter at each clock cycle. The value of the counter is compared to a programmable threshold (called Watermark), and this comparison determines whether the packet is processed with high pressure for minimum latency or low pressure for best effort processing.

The bandwidth regulator monitors the traffic using open connections between the initiators and the targets. If there is insufficient bandwidth allocated to the connection, the bandwidth regulator can increase the pressure on connections. Generally, the connection is a dataflow between master and slave NIUs. In some cases, the bandwidth regulator is attached to the master and monitors single dataflow to a target (single connection).



When Counter Value (bytes) falls below Watermark, the pressure bit is assigned to 1

Figure 14-4. Bandwidth Regulator Pressure Settings

The bandwidth regulator effective resolution is set to 8.3125 MBps

The following is an example of bandwidth regulator settings:

Suppose the bandwidth regulator is set to run at 200 MHz and the application requires an expected bandwidth of 165.888 MBps (±5 MBps), computed through a moving window of 5 µs (1000 cycles). To attribute high pressure on all packet requests, the watermark could be set to the maximum bandwidth needed in the 5-µs window.

Considering the example, the settings of the bandwidth regulator are:

- L3_BW_REGULATOR_WATERMARK[11:0] WATERMARK = moving window × expected bandwidth = 829.44 bytes = 0x33D
- L3_BW_REGULATOR_BANDWIDTH[15:0] BANDWIDTH = ceil(165.888/8.3125) = ceil(19.956) = 20d = 0x14

The bandwidth registers regulate the packet flow by applying flow control on the RX port, thus ensuring that the traffic does not exceed the allocated bandwidth. The next packet is sent only when an internal timer expires. The registers in this group are:

- L3_BW_REGULATOR_WATERMARK: Gives the amount of data allowed to exceed the average bandwidth during a short time period
- L3_BW_REGULATOR_PRESS: Describes the pressure applied to outgoing packets
- L3_BW_REGULATOR_CLEARHISTORY: Resets the traffic counter when set to 1. This register is used after an update in the L3_BW_REGULATOR_BANDWIDTH and L3_BW_REGULATOR_WATERMARK registers (see Section 14.2.5.1.7, *L3 BW Regulator Register Summary and Description*).

Bandwidth regulators are mainly used to give priority to the following masters: DSP1 MDMA, DSP2 MDMA, EVE1 and EVE2 (both ports per instance), IVA, MMU2, PCIe, DSP1 EDMA, DSP2 EDMA, GMAC SW, BB2D.

Priority to: DSP1_CFG, DSP2_CFG, DSS, GPU_P1, GPU_P2, IPU1, IPU2, MLB, MMC1, MMC2, MMU 2 initiator ports, MPU initiator port, PCIe1 and PCIe2 initiator ports, TPTC1 and TPTC2 (RD and WR initiator ports), USB3 initiator port, VIP1_P1/P2, VIP2_P1/P2, VIP3_P1/P2 initiator ports is given by setting their internal MFlag signal.

14.2.3.4 Bandwidth Limiters

The bandwidth limiter is added to control the bandwidth of the EDMA_TC1_RD, EDMA_TC1_WR, EDMA_TC2_RD, EDMA_TC2_WR, VPE, BB2D and GPU module. This prevents a large number of RD requests being processed together, thus avoiding a large number of RD responses.

The bandwidth limiter regulates the packet flow in the L3_MAIN interconnect by applying flow control when a user-defined bandwidth limit is reached. The next packet is served only after an internal timer expires, thus ensuring that traffic does not exceed the allocated bandwidth. Bandwidth limiter can be used with a watermark mechanism that allows traffic to temporarily exceeds the peak bandwidth.

The registers in this group are:

- L3_BW_LIMITER_WATERMARK_0: Gives the amount of data allowed to exceed the average bandwidth during a short time period. To set the actual watermark to n bytes, the register must be set to n + 1.
- L3_BW_LIMITER_CLEARHISTORY: Resets the traffic counter when set to 1.
- L3_BW_LIMITER_BANDWIDTH_FRACTIONAL and L3_BW_L_BANDWIDTH_INTEGER: These two registers are used to set the average payload bandwidth.

Example of setting the Bandwidth Limiters:

L3_BW_LIMITER_BANDWIDTH_FRACTIONAL must be set to BandwidthConf [4 down to 0] and L3_BW_LIMITER_BANDWIDTH_INTEGER contains the remaining BandiwdthConf bits, shifted to the right. BandwidthConf = AverageBandwidth / EffectiveResolution, where EffectiveResolution parameter is set to 8.3125 MHz at design time.

AverageBandwidth is a parameter representing the value to which the payload bandwidth must be limited in average; this parameter depends on the use case and is the reason that makes the content of the registers variable. In addition the maximum packet length is 8 cells, that is 32 bytes.

Assuming that the use case requires that in average the payload bandwidth is limited to 200 MB/s, then registers must be set to the following values:

BandwidthConf = 200 MBps / 8.3125 MHz = 38 (0x26)

L3_BW_LIMITER_BANDWIDTH_FRACTIONAL = LSBs [4 down to 0] of BandwidthConf = 0x6.

L3_BW_LIMITER_BANDWIDTH_INTEGER = the remaining bits of BandwidthConf shifted to the right = 0x1.

By setting L3_BW_LIMITER_WATERMARK to 65 (L3_BW_LIMITER_WATERMARK_0 = 0x41), the bandwidth limiter is able to send three packets consecutively during peaks, therefore exceeding the peak traffic with two packets (64 bytes)

14.2.3.5 Flag Muxing

The flag mux generator collects information such as errors and interrupts from slave NIUs and the interconnect firewall. The result signals are then sent to the MPU interrupt controller (INTC) without interfering with the interconnect traffic. Using the L3_FLAGMUX_MASK registers can prevent the flag mux from seeing certain events.

The unit has a standard COREREG register for identification of the attached core type. The L3_FLAGMUX_STDHOSTHDR_VERSIONREG register identifies the characteristics of the attached core. Use unit-specific registers (MASK bit 0 or bit 1 of the flag inputs, and L3_FLAGMUX_REGERR bit 0 or bit 1) to read the input errors. Each register is dedicated to reporting the bit corresponding to the register number; for example, L3_FLAGMUX_REGERR0 reports on bit 0, and L3_FLAGMUX_REGERR1 reports on bit 1. Any given L3_FLAGMUX_REGERR register reports the same bit for all flag source inputs (see Table 14-170).

There are three flag muxs (CLK1_FLAGMUX_CLK1_1, CLK1_FLAGMUX_CLK1_2 and CLK2_FLAGMUX_CLK2_1) collecting information from targats in each clock domain in L3_MAIN interconnect (CLK1_1, CLK1_2 and CLK2_1). Both CLK1_FLAGMUX_CLK1_1 and CLK1_FLAGMUX_CLK1_2 flag muxes are located in the CLK1_2 clock domain (with base address 0x4480 0000).

Also there is a separate mux (L3_FMAGMUX_CLK1MERGE) that merges the inputs from CLK1_1 and CLK1_2 flag muxes. Its functionality is similar to the functionality of the other L3_FLAGMUXES:

- L3_FLAGMUX_CLK1MERGE_COREREG and L3_FLAGMUX_STDHOSTHDR_VERSIONREG registers are used to identify the attached core and its characteristics.
- L3_FLAGMUX_CLK1MERGE_MASK0 is used to mask (enable) application error sources
- L3_FLAGMUX_CLK1MERGE_REGERR0 is used to check which application error sources are active
- L3_FLAGMUX_CLK1MERGE_MASK1 is used to mask debug error sources
- L3_FLAGMUX_CLK1MERGE_REGERR1 is used to check which debug error sources are active

Following is a block diagram of the flag mux organization in L3_MAIN interconnect:

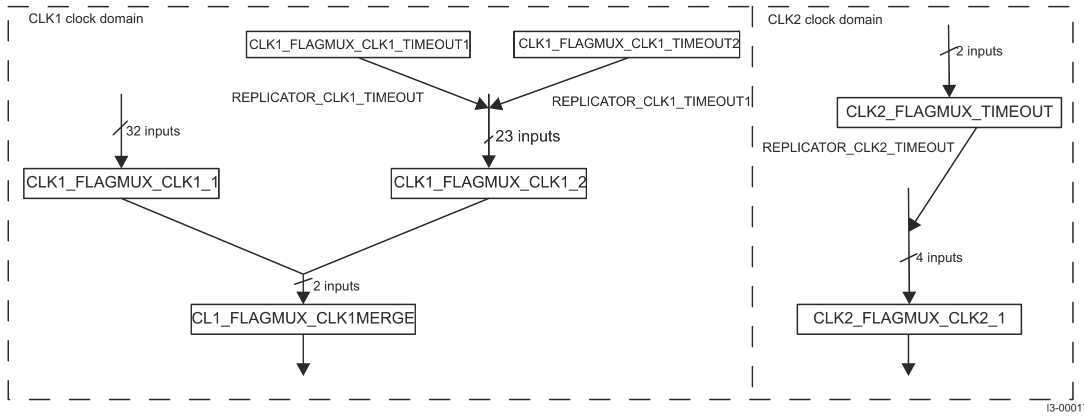


Figure 14-5. Flag Mux Structure

14.2.3.5.1 Flag Mux Time-out

If a target does not respond after a fixed number of clock cycles, an error time-out flag is generated, in case it is enabled. The value of the flag is always 3 if a time-out error is present.

To enable time-out for each target, set the corresponding bit in L3_FLAGMUX_TIMEOUT_MASK0 register to 1. See Table 14-11.

Table 14-11. L3 Time-out Flag Mapping

	Flag Mux Input	Source	
	2	DSP1 SDMA	
	3	DSP2 SDMA	
	4	DSS	
	5	EVE1	
	6	EVE2	
	9	GPU	
	10	BB2D	
	11	IPU1	
	12	IPU2	
	13	IVA CONFIG	
	14	MMU1	
	15	DMM P1	
	16	DMM P2	
CLK1_1 Time-out FlagMux (TIMEOUT1)	17	IVA SL2IF	
	18	MMU2	
	19	L4_PER1 P1	
	20	L4_PER1 P2	
	21	L4_PER1 P3	
	22	L4_PER2 P2	
	23	L4_PER2 P1	
	24	L4_PER2 P3	
	25	L4_PER3 P2	
	26	L4_PER3 P1	
	27	L4_PER3 P3	
	28	L4_CFG	
	29	L4_WKUP	
		1	PCIE1
2		PCIE2	
6		QSPI	
8		TPCC	
9		TPTC1	
10		TPTC2	
12		OCMC_RAM3	
13	CLK1_2 Time-out FlagMux (TIMEOUT2)	McASP1	
14		McASP2	
15		McASP3	
16		OCMC_RAM1	
17		OCMC_RAM2	
18		GPMC	
19		VCP1	
20		VCP2	
CLK2_1 Time-out FlagMux		0	L3_INSTR
	1	DEBUGSS_CT_TBR	

For example, to enable all targets in CLK1_1, write 0x3FL3_FLAGMUX_TIMEOUT1_MASK0; to enable all targets in CLK1_2, write 0x1F FFFFL3_FLAGMUX_TIMEOUT2_MASK0. To enable only OCMC_RAM1 (target in CLK1_2) write 0x1L3_FLAGMUX_TIMEOUT2_MASK0. If an error time-out occurs, read the registers L3_FLAGMUX_TIMEOUT1_REGERR0 and L3_FLAGMUX_TIMEOUT2_REGERR0 to determine the source of error concerning Table 14-11. CLK2 time-outs are reported through CLK1_FLAGMUX_CLK2_1 status registers (L3_FLAGMUX_REGERR0, L3_FLAGMUX_REGERR1) and masked through L3_FLAGMUX_MASK0 and L3_FLAGMUX_MASK1 registers.

Similarly, to enable a target in CLK2_1, write the appropriate value (as described in Table 14-11) in L3_FLAGMUX_TIMEOUT_MASK0.

If an error time-out occurs in CLK2_1, read the L3_FLAGMUX_TIMEOUT_REGERR0 register to determine the source of error concerning Table 14-11.

14.2.3.6 Statistic Collectors Group

Statistic collectors are internal masters that share the same master address as the master NIUs. These components compute the traffic statistics within a defined window and periodically report through the DEBUG interface. The key features of the statistic collector are:

- Nonintrusive monitoring
- Programmable filters and counters
- Collects results at a programmable time interval

Event detectors are programmed through the L3_STCOL_REQEVT and L3_STCOL_RSPEVT configuration registers for request and response ports, respectively. The following events can be identified:

- Word transfer
- WAIT cycles
- Flow control
- Payload transfers
- Latency measurements

Performance monitoring is enabled through the L3_STCOL_EN register. The L3_STCOL_SOFTEN register enables software to monitor the performance. Event muxes are programmed through the L3_STCOL_EVTMUX_SEL0 configuration register, which determines which port will be monitored by a filter configured by the filter registers (see Section 14.2.5.1.9).

Filters are programmed through the L3_STCOL_FILTER_i_GLOBALEN configuration register, along with additional selection criteria programmed through the mask/match registers (see Table 14-235). A filter can be configured to accept or reject:

- Read operations
- Write operations
- Errors
- Addresses

Filter operation is programmed through the L3_STCOL_OP registers (see Table 14-235).

There are ten statistic collectors used to monitor the traffic on DRAM (EMIF1, EMIF2, MA_MPU_P1 and MA_MPU_P2), MPU, MMU, TPTC, VIP, VPE, EVE Subsystem, DSP MDMA/EDMA, IVA, GPU, BB2D, DSS, IPU, OCMC RAM, USB, PCIe Subsystem, DSP CFG, MMC, SATA, VCP, GPMC, and McASP ports. For more detailed descriptions of statistic collectors, see Section 33.10.7.1, *L3 Target Load Monitoring*, and Section 33.10.7.2, *L3 Master Latency Monitoring*.

14.2.3.7 L3_MAIN Protection and Firewalls

Device protection relies on L3 firewalls and their configuration.

14.2.3.7.1 L3_MAIN Firewall Reset

The values of L3_MAIN firewall registers on reset are tied in hardware or exported from the control module registers.

Values exported from the control module are intended to give defined rights to the firewalls at reset and thus ensure the content after going out of reset.

The L3_MAIN firewall registers are located in the CORE AON power domain and thus no retention capability is needed. The control module registers are reset by a cold reset only, whereas the L3_MAIN firewall registers are reset by clearing the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit. When the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit comes back automatically to 1, the exported values are loaded.

CAUTION

Before reprogramming the firewall registers and/or before using the FW_LOAD_REQ mechanism, the request must be asserted by configuring the REGUPDATE_CONTROL[0] BUSY_REQ bit.

To load the exported values at run time:

1. Set the REGUPDATE_CONTROL[0] BUSY_REQ bit to 0x1 to ensure that no transaction can reach the slave NIU (suspend).
2. Clear the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit by writing 0x1 to it.
3. Wait until the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit is reset to 0x1 by hardware.
4. Set the REGUPDATE_CONTROL[0] BUSY_REQ bit to 0x0 to allow transactions to reach the slave NIU (resume).

To reprogram the firewall registers at run time:

1. Write 0x1 to the REGUPDATE_CONTROL register.
2. Update the firewall registers.
3. Write 0x0 to the REGUPDATE_CONTROL register.

Note

While reprogramming the firewall registers at run time it must be taken into account that the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit is written as '0' because a value of '1' reloads the firewall default values.

Note

At reset, exported values from the control module can modify hardware reset values.

14.2.3.7.1.1 L3_MAIN Firewall – Exported Reset Values

Table 14-12 and Table 14-13 list the exported reset values and mapping, respectively.

Table 14-12. L3_MAIN Firewall Exported Reset Values

MRM_PERMISSION_REGION_LOW_j **[15:12]**	MRM_PERMISSION_REGION_LOW_j **[11:0]**
0x0	0xFFF
0xF	0xFFF
0x0	0xFFF
0x2	0xFFF

Table 14-13. L3_MAIN Firewall Exported Values Mapping

CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF_LOCK_1 and CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF_DBG_1 Bits	Slave NIU Firewall
[0]	GPMC
[3]	L3 RAM1
[4]	DSS
[6]	GPU
[7]	IVAHD SL2IF
[8]	IVAHD CONFIG
[11]	EMIF and MA_MPU_NTTP
[12]	DEBUGSS
[13]	CT_TBR
[16]	EVE1
[17]	EVE2
[20]	PCIESS1
[21]	PCIESS2
[22]	IPU1
[23]	IPU2
[24]	VCP1
[25]	VCP2
[26]	McASP1
[27]	McASP2
[28]	McASP3
[31]	BB2D

Table 14-14. L3_MAIN Firewall Exported Values Mapping

CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF_LOCK_2 and CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF_DBG_2 Bits	Slave NIU Firewall
[0]	DSP1
[1]	DSP2
[2]	L3 RAM2
[3]	L3 RAM3
[8]	QSPI
[9]	EDMA TC
[10]	EDMA TPCC

For more information, see Chapter 18, *Control Module*.

14.2.3.7.2 Power Management

As part of the system-wide power-management scheme, the L3_MAIN interconnect goes into IDLE state after receiving a request from the power, reset, and clock management (PRCM) module after all commands are serviced. This function is handled by hardware.

To reduce power consumption, the L3_MAIN interconnect automatically performs internal clock autogating. This is managed by hardware; no software configurations or settings are required.

L3_MAIN supports a partial retention scheme. Retention is performed on the follwoing registers:

- Statistic collectors
- Bandwidth regulators
- Firewalls

This process prevents reconfiguration after a clock domain switches off.

14.2.3.7.3 L3_MAIN Firewall Functionality

The access to the slave NIUs is granted only to master NIUs according to in-band attributes sent in each transaction crossing the L3_MAIN interconnect, such as:

- MCMD: Specifies the type of access (read or write) required by the master NIU
- ConnID: Used to determine the permission of the master NIU
- MReqInfo: Transaction attribute adding information about the access type

Table 14-15 lists the MReqInfo values.

Table 14-15. MReqInfo Values

Qualifier	Access Definition	Access Description
MReqType	00: Processor data access 01: Processor instruction access 10: DMA access 11: Other	Indicates whether the request is for processor instruction fetch, processor data access or DMA access
MReqDebug	0: Functional 1: Debug	When set, indicates that the request has been issued by a master NIU in DEBUG state
MReqSupervisor	0: User 1: Privilege	When set, indicates that the request is qualified with the supervisor attribute. It can be provided by a processor running in supervisor mode or by a module that inherited this attribute from the processor (DMA channel with a supervisor attribute).

The firewall comparison mechanism enables access to a protected slave NIU only when a correct combination of three MReqInfo in-band parameters is transmitted.

MReqInfo is a combination of a fixed 3-bit pattern that corresponds to a combination of the parameters MReqDebug, MReqType, and MReqSupervisor. See Table 14-16.

Table 14-16. L3_MAIN ReqInfo Mapping

ReqInfo Name		MReqDebug	MReqType	MReqSupervisor
Master NIUs	MPU INIT	x	x	x
	MMU1 INIT	x		x
	TPTC1_RD INIT			x
	TPTC1_WR INIT			x
	TPTC2_RD INIT			x
	TPTC2_WR INIT			x
	VPE_P1 INIT			
	VPE_P2 INIT			
	VIP1_P1 INIT			
	VIP1_P2 INIT			
	VIP2_P1 INIT			
	VIP2_P2 INIT			
	VIP3_P1 INIT			
	VIP3_P2 INIT			
	EVE1_P1 INIT	x	x	
	EVE1_P2 INIT	x	x	
	EVE2_P1 INIT	x	x	
	EVE2_P2 INIT	x	x	
	DSP1 EDMA INIT	x	x	x
	DSP1 MDMA INIT	x	x	x
	DSP2 EDMA INIT	x	x	x
	DSP2 MDMA INIT	x	x	x
	IVA INIT			
	GPU_P1 INIT			
	GPU_P2 INIT			
	BB2D_P1 INIT			
	BB2D_P2 INIT			
	DSS INIT			
	MMU2 INIT	x		x
	IPU1 INIT	x	x	x
	IPU2 INIT	x	x	x
	DMA_SYSTEM_RD INIT		x	x
	DMA_SYSTEM_WR INIT		x	x
	USB1 INIT (USB3_SS)			
	USB2 INIT (USB2_SS)			
	USB3 INIT (USB2_ULPI_SS1)			
	USB4 INIT (USB2_ULPI_SS2)			
	PCIe_SS1 INIT			
	PCIe_SS2 INIT			
	DSP1_CFG INIT	x	x	x
	DSP2_CFG INIT	x	x	x
	GMAC SW INIT			
	MMC1 INIT			
	MMC2 INIT			

Table 14-16. L3_MAIN ReqInfo Mapping (continued)

ReqInfo Name		MReqDebug	MReqType	MReqSupervisor
	SATA INIT			
	MLB INIT			
	DAP INIT	x		x
Slave NIUs	DMM_P1 TARG	x	x	x
	DMM_P2 TARG	x	x	x
	DSP1 SDMA TARG	x	x	x
	DSP2 SDMA TARG	x	x	x
	EVE1 TARG	x		
	EVE2 TARG	x		
	L4_CFG TARG	x		x
	L4_WKUP TARG	x		x
	TPTC1_CFG TARG	x		x
	TPTC2_CFG TARG	x		x
	TPCC TARG	x	x	x
	L3_INSTR TARG	x		
	DEBUGSS TARG	x		
	OCMC_RAM1 TARG			
	OCMC_RAM2 TARG			
	OCMC_RAM3 TARG			
	GPU TARG			
	IPU1 TARG			
	IPU2 TARG			
	VCP1 TARG			
	VCP2 TARG			
	PCIESS1 TARG			
	PCIESS2 TARG			
	GPMC TARG			
	L4_PER1_P1 TARG	x		x
	L4_PER1_P2 TARG	x		x
	L4_PER1_P3 TARG	x		x
	L4_PER2_P1 TARG	x		x
	L4_PER2_P2 TARG	x		x
	L4_PER2_P3 TARG	x		x
	L4_PER3_P1 TARG	x		x
	L4_PER3_P2 TARG	x		x
	L4_PER3_P3 TARG	x		x
	QSPI TARG			
	McASP1 TARG			
	McASP2 TARG			
	McASP3 TARG			
	DSS TARG			x
	BB2D TARG			
	IVA_CFG TARG	x		
	MMU1 TARG	x		x

Table 14-16. L3_MAIN ReqInfo Mapping (continued)

ReqInfo Name		MReqDebug	MReqType	MReqSupervisor
	MMU2 TARG	x		x

14.2.3.7.3.1 Protection Regions

Each slave NIU address space is subdivided into protection regions (maximum of 10). The regions are configurable with a size of 4-KiB granularity. The firewalls can also be multiport while using the description of the same regions for dual access memories or to support interleaving mechanisms on several memories.

Table 14-17 lists the number of protected regions and ports for each slave NIU.

Table 14-17. Slave NIU Firewall and Region Configuration

Domain	Slave NIU	Firewall	Number of Regions	Number of Ports
CLK1	DSP1_SDMA	DSP1_SDMA_FW	1	1
	DSP2_SDMA	DSP2_SDMA_FW	1	1
	DSS	DSS_FW	8	1
	EVE1	EVE1_FW	1	1
	EVE2	EVE2_FW	1	1
	GPMC	GPMC_FW	8	1
	GPU	GPU_FW	1	1
	BB2D	BB2D_FW	1	1
	IPU1	IPU1_FW	4	1
	IPU2	IPU2_FW	4	1
	IVA_CFG	IVA_CFG_FW	1	1
	IVA_SL2IF	IVA_SL2IF_FW	4	1
	McASP1	MCASP1_FW	1	1
	McASP2	MCASP2_FW	1	1
	McASP3	MCASP3_FW	1	1
	OCMC_RAM1	OCMC1_RAM_FW	16	1
	OCMC_RAM2	OCMC2_RAM_FW	16	1
	OCMC_RAM3	OCMC3_RAM_FW	16	1
	EMIF1, EMIF2	EMIF_OCP_FW	8	2
	EMIF1, EMIF2	MA_MPU_NTTP_FW	8	2
	PCIESS1	PCIESS1_FW	8	1
	PCIESS2	PCIESS2_FW	8	1
	QSPI	QSPI_FW	1	1
	TPCC	TPCC_FW	1	1
	TPTC	TPTC_FW	2	2
	VCP1	VCP1_FW	1	1
	VCP2	VCP2_FW	1	1
CLK2	L3_INSTR	L3_INSTR_FW	2	1
	DEBUGSS_CT_TBR	DEBUGSS_CT_TBR_FW	1	1

Two types of regions are distinguished in a slave NIU firewall:

- Default region: Available in all slave NIUs. The default region covers the entire slave NIU address range. Other firewall-configured regions must reset or overlay the default region, because it always has the lowest priority.

- Normal region: The number of normal regions varies in a slave NIU; they have identical capabilities (see Table 14-17).

Each region has the following characteristics:

- Start address: Physical slave NIU start address
- End address: Physical slave NIU end address
- Specific access rights (see Section 14.2.3.7.3.3, *Protection Mechanism per Region Examples*)
- Priority level from 0 (lowest) to 10 (highest)

Depending on its priority level, a region can override the settings of another region; the access rights of the region with the highest priority apply. All regions have a fixed (not configurable) priority level that corresponds to their number: Region 1 has priority level 1, region 2 has priority level 2, and so on.

Figure 14-6 shows the priority level with associated regions. This priority level scheme allows multiplying the flexibility and capability of the firewall. Figure 14-6 shows a 7-region firewall setting that creates 16 regions (twice the number of regions created than originally available).

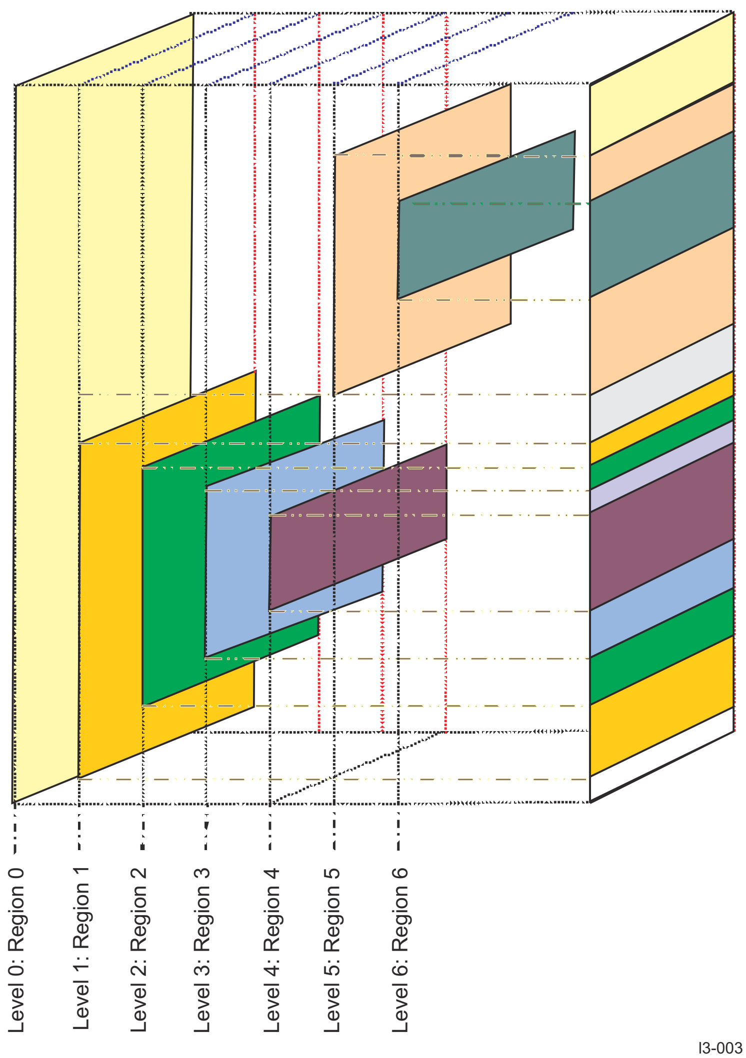


Figure 14-6. L3 Interconnect Region Overlay and Priority Level Overview

The address range covered by the regions is defined in the START_REGION_i and END_REGION_i registers. The boundary checks are done on a minimum size of 4-KiB pages; thus, bits [11:0] of those 32-bit registers are not checked.

The address space size of the slave NIUs ([bits [31:12]) depends on the size of the slave NIU to protect (that is, if a memory is only 48KiB, then the size is defined through bits [16:12] of the slave NIU start and end address registers of the firewall region (START_REGION_i[16:12] and END_REGION_i[16:12]).

On multiport firewalls (EMIF_OCP_FW), the checking of each REGION {0, n} can activate on one or several ports at the same time. However, if interleaving is not desired on some parts of the protected memory space, it is possible to apply or not apply region checking on selected ports using the region enable/disable on port instance capability of the END_REGION_i register.

Most slave NIUs support only one input port (port 0) except:

- The MA_MPU firewall (MA_MPU_NTTP_FW) which has two ports:
  – Port 0 for access from the MPU to EMIF1
  – Port 1 for access from the MPU to EMIF2
- The EMIF firewall which has two ports:
  – Port 0 for access from the DMM to EMIF1
  – Port 1 for access from the DMM to EMIF2
- The EDMA_TC firewall which has two ports:
  – Port 0 logs errors in the ERROR_LOG_0 register for the traffic to EDMA_TPTC0 module generated by the MPU, DSP1/2, EVE1/2, IPU1/2, DMA_SYSTEM, MLB, PCIESS1/2, MMU1/2 and DAP modules.
  – Port 1 logs errors in the ERROR_LOG_1 register for the traffic to EDMA_TPTC1 module generated by the MPU, DSP1/2, EVE1/2, IPU1/2, DMA_SYSTEM, MLB, PCIESS1/2, MMU1/2 and DAP modules.

A region can be applied or not to each port independently. To enable and disable the regions:

- For port 0: Set/clear the END_REGION_i[0] END_REGION_i_ENABLE_CORE0 bit (for all L3_MAIN firewalls).
- For port 1: Set/clear the END_REGION_i[1] END_REGION_i_ENABLE_CORE1 bit (for MA_MPU, EMIF and EDMA_TC firewalls).

The EMIF firewall (EMIF_OCP_FW) uses 16 GiB address space and protects both the SDRAM and the EMIF configuration registers. The EMIF firewall allows defining access restrictions per region. The region can apply to:

- Both EMIF1 and EMIF2 controllers (interleaving mode)
- Only the EMIF1 controller
- Only the EMIF2 controller

This per-region capability is defined by configuring the END_REGION_i[1:0] bit field for:

- Both EMIF1 and EMIF2 (interleaving mode): END_REGION_i[0] END_REGION_i_ENABLE_CORE_0 = 0x1 and END_REGION_i[1]END_REGION_i_ENABLE_CORE_1 = 0x1
- Only EMIF1: END_REGION_i[0] END_REGION_i_ENABLE_CORE_0 = 0x1
- Only EMIF2: END_REGION_i[1] END_REGION_i_ENABLE_CORE_1 = 0x1

The protection regions for EMIF firewall are configurable with a size of 4-KiB granularity (the same applies to the MA_MPU_NTTP_FW). The address range covered by the regions is defined in the START_REGION_i and END_REGION_i registers. Since EMIF firewall sees 16 GiB address space but these two registers have only 32 bits, the START_REGION_i/END_REGION_i[31] bit is mapped to bit [33] of the EMIF address space, that is, bits [31:10] of START_REGION_i/END_REGION_i are mapped to EMIF Address [33:12]. START_REGION_i/ END_REGION_i [9:0] are not checked for the minimum page size of 4 KiB.

The start address in the EMIF address space for accessing the EMIF configuration registers is 0x3_0000_0000. If protection of these registers is needed, that address must be right shifted by 2 bits and then the result (0xC000_0000) must be written to the START_REGION_i register. The REGUPDATE_CONTROL[19:16] FW_ADDR_SPACE_MSB bit field indicates how many bits are shifted in the firewall address space. In case of EMIF_OCP_FW and MA_MPU_NTTP_FW, 2 bits are shifted. If target address space is ≤ 4 GiB then REGUPDATE_CONTROL[19:16] FW_ADDR_SPACE_MSB is 0x0 and the START_REGION_i/END_REGION_i [31:12] bits match target physical address [31:12]. Bits [11:0] are ignored by the firewall. These bits are not involved in the address check. To match address spaces greater than 4 GiB but keep the minimum page size of

4 KiB some of the START_REGION_i/END_REGION_i [11:0] bits must also be configured. In the EMIF case bits [11:10] are also used.

For example, the EMIF1 configuration registers are accessible through system base address 0x4C00_0000. But the EMIF_OCP_FW sees them at start address 0x3_0000_0000. As previously mentioned that address must be shifted before being programmed to the START_REGION_i register. In this case the programmed value should be 0xC000_0000. This has to be taken into account.

14.2.3.7.3.2 L3_MAIN Firewall Registers Overview

Table 14-18 and Table 14-19 list the L3_MAIN firewall permission-setting registers.

Table 14-18. L3_MAIN Firewall Read/Write Permission-Setting Register

Register Name	Bits	Field Name	4-bit ConnID Value (hex) (see Table 14-10)	Field Modifiability
MRM_PERMISSION_REGION_HIGH_j	31	W	ConnID = F write permission	RW
	30	R	ConnID = F read permission	RW
	29	W	ConnID = E write permission	RW
	28	R	ConnID = E read permission	RW
	27	W	ConnID = D write permission	RW
	26	R	ConnID = D read permission	RW
	25	W	ConnID = C write permission	RW
	24	R	ConnID = C read permission	RW
	23	W	ConnID = B write permission	RW
	22	R	ConnID = B read permission	RW
	21	W	ConnID = A write permission	RW
	20	R	ConnID = A read permission	RW
	19	W	ConnID = 9 write permission	RW
	18	R	ConnID = 9 read permission	RW
	17	W	ConnID = 8 write permission	RW
	16	R	ConnID = 8 read permission	RW
	15	W	ConnID = 7 write permission	RW
	14	R	ConnID = 7 read permission	RW
	13	W	ConnID = 6 write permission	RW
	12	R	ConnID = 6 read permission	RW
	11	W	ConnID = 5 write permission	RW
	10	R	ConnID = 5 read permission	RW
	9	W	ConnID = 4 write permission	RW
	8	R	ConnID = 4 read permission	RW
	7	W	ConnID = 3 write permission	RW
	6	R	ConnID = 3 read permission	RW
	5	W	ConnID = 2 write permission	RW
	4	R	ConnID = 2 read permission	RW
	3	W	ConnID = 1 write permission	RW
	2	R	ConnID = 1 read permission	RW
	1	W	ConnID = 0 write permission	RW
	0	R	ConnID = 0 read permission	RW

Table 14-19. L3_MAIN Firewall Permission-Setting Register

Register name	Type of Permission	Bits	Field Name	Description	Field Modifiability
MRM_PERMISSION_REGION_LOW_j	Reserved	31:16		Reserved	
	Debug	15	DEBUG	PUBLIC PRIVILEGE DOMAIN DEBUG ALLOWED	RW
		14	DEBUG	PUBLIC USER DOMAIN DEBUG ALLOWED	RW
	Reserved	13:12		Reserved	
	Access	11	READ	PUBLIC PRIVILEGE READ ACCESS ALLOWED	RW
		10	WRITE	PUBLIC PRIVILEGE WRITE ACCESS ALLOWED	RW
		9	EXE	PUBLIC PRIVILEGE EXE ACCESS ALLOWED	RW
		8	READ	PUBLIC USER READ ACCESS ALLOWED	RW
		7	WRITE	PUBLIC USER WRITE ACCESS ALLOWED	RW
		6	EXE	PUBLIC USER ECXE ACCESS ALLOWED	RW

14.2.3.7.3.3 Protection Mechanism per Region Examples

The access permission of each region is configurable and defined through the MRM_PERMISSION_REGION_HIGH_j and MRM_PERMISSION_REGION_LOW_j registers (see Section 14.2.3.7.3.2, *L3_MAIN Firewall Registers Overview*).

Master NIU permissions:

1. To give read access to the master NIU with 4-bit ConnID = n(1), set the MRM_PERMISSION_REGION_HIGH_j[n × 2] R bit.
2. To give write access to the master NIU with 4-bit ConnID = n(1), set the MRM_PERMISSION_REGION_HIGH_j[n × 2 + 1] W bit.

   (1) - n should be first converted from hex to decimal value

Debug permissions:

1. To give privilege debug access, set the MRM_PERMISSION_REGION_LOW_j[15] DEBUG bit.
2. To give user debug access, set the MRM_PERMISSION_REGION_LOW_j[14] DEBUG bit.

User, read, write, and executable permissions:

1. To give privileged read access, set the MRM_PERMISSION_REGION_LOW_j[11] READ bit.
2. To give privileged write access, set the MRM_PERMISSION_REGION_LOW_j[10] WRITE bit.
3. To give privileged executable access, set the MRM_PERMISSION_REGION_LOW_j[9] EXE bit.
4. To give user read access, set the MRM_PERMISSION_REGION_LOW_j[8] READ bit.
5. To give user write access, set the MRM_PERMISSION_REGION_LOW_j[7] WRITE bit.
6. To give user executable access, set the MRM_PERMISSION_REGION_LOW_j[6] EXE bit.

Example: To provide debug write privilege access to the master NIU with 4-bit ConnID = 0x7, set the following bits:

- MRM_PERMISSION_REGION_HIGH_j[15] W
- MRM_PERMISSION_REGION_LOW_j[15] DEBUG

14.2.3.7.3.4 L3_MAIN Firewall Error Logging

If a protection violation error is detected, the following signals are generated:

- An in-band error (SRESP = ERROR) is generated to the master NIU of the access.
- An out-band error is sent to the control module
- An interrupt is generated to the Cortex-A15 INTC.

The L3_MAIN interconnect does not differentiate errors generated by firewalls from all other supported types of errors.

An in-band error is generated by modules each time an access is not allowed. When an in-band error is sent back into the transaction it is seen as an external prefetch or data abort by the initiator, depending on whether the transaction was an instruction fetch or a data access.

Information about in-band errors is logged into two registers:

- ERROR_LOG_k: Logs the information about the start/end address of the hit region and the qualifiers of the transaction
- LOGICAL_PHYSICAL_ADDRESS_ERRLOG_k[31:12]: Logs the address of the failed access

Note

When a multiport firewall is implemented, these registers are duplicated for each port.

Table 14-20 lists the L3_MAIN firewall error-logging registers.

Table 14-20. L3 Firewall Error-Logging Registers

Register Name	Register Field Name	Field Modifiability	Parameter Comments
ERROR_LOG_k (When multiport firewall is implemented the error log register is duplicated for each port.)	RESERVED[31:24]	Read only	Reads return 0s.
	BLK_BURST_VIOLATION[23]	Read/write	Read 0x1: 2D burst not allowed or exceeds allowed size. Write to clear the ERROR_LOG and LOGICAL_PHYSICAL_ADDRESS_ERRLOG registers.
	RESERVED[22]	Read only	Read return 0s.
	REGION_START_ERRLOG[21:17]	Read/write	Read: Wrong access hit this region number. Write to clear the ERROR_LOG and LOGICAL_PHYSICAL_ADDRESS_ERRLOG registers.
	REGION_END_ERRLOG[16:12]	Read/write	Read: Wrong access hit this region number. Write to clear the ERROR_LOG and LOGICAL_PHYSICAL_ADDRESS_ERRLOG registers.
	REQINFO_ERRLOG[11:0]	Read/write	Mapping of the error according to the reqinfo vector: ConnID [3:0] MCMD [0] [3] MReqDebug [1] MReqSupervisor [0] MReqType

L3 firewall errors can be cleared by writing to the ERROR_LOG_k register in the firewall that recorded the error. Clearing the ERROR_LOG_k register deasserts the corresponding error if it exists.

The L3 firewall register ERROR_LOG_k must be cleared before clearing the SEC_ERR_STATUS_FUNC_1/2 and SEC_ERR_STATUS_DEBUG_1/2 registers in the control module.

When a protection violation occurs, an interrupt is sent to the IRQ_CROSSBAR. An in-band error is sent back, and an error is logged in the SEC_ERR_STATUS_FUNC_1/2 and SEC_ERR_STATUS_DEBUG_1/2 registers, depending on the functional mode:

- In application mode:
  - SEC_ERR_STATUS_FUNC_1[1] – L3 RAM protection violation
  - SEC_ERR_STATUS_FUNC_1[2] – GPMC protection violation

  - SEC_ERR_STATUS_FUNC_1[3] – EMIF protection violation
  - SEC_ERR_STATUS_FUNC_1[4] – IVA protection violation
  - SEC_ERR_STATUS_FUNC_1[5] – IPU protection violation
  - SEC_ERR_STATUS_FUNC_1[6] – IVA SL2 protection violation
  - SEC_ERR_STATUS_FUNC_1[8] – SYSTEM DMA protection violation
  - SEC_ERR_STATUS_FUNC_1[10] – DSPDMA slave NIU protection violation
  - SEC_ERR_STATUS_FUNC_1[13] – GPU protection violation
  - SEC_ERR_STATUS_FUNC_1[14] – DSS protection violation
  - SEC_ERR_STATUS_FUNC_1[16] – L4_PER1 protection violation
  - SEC_ERR_STATUS_FUNC_1[17] – L4_CFG protection violation
  - SEC_ERR_STATUS_FUNC_1[18] – DEBUG subsystem protection violation
  - SEC_ERR_STATUS_FUNC_1[22] – L4_WKUP protection violation
  - SEC_ERR_STATUS_FUNC_1[23] – BB2D protection violation
  - SEC_ERR_STATUS_FUNC_1[26] – CT_TBR protection violation
  - SEC_ERR_STATUS_FUNC_1[28] – EVE1 protection violation
  - SEC_ERR_STATUS_FUNC_1[29] – EVE2 protection violation
  - SEC_ERR_STATUS_FUNC_2[0] – DSP1 protection violation
  - SEC_ERR_STATUS_FUNC_2[1] – DSP2 protection violation
  - SEC_ERR_STATUS_FUNC_2[2] – OCMC_RAM2 protection violation
  - SEC_ERR_STATUS_FUNC_2[3] – OCMC_RAM3 protection violation
  - SEC_ERR_STATUS_FUNC_2[4] – L4_PER2 protection violation
  - SEC_ERR_STATUS_FUNC_2[5] – L4_PER3 protection violation
  - SEC_ERR_STATUS_FUNC_2[6] – IPU2 protection violation
  - SEC_ERR_STATUS_FUNC_2[7] – PCIESS1 protection violation
  - SEC_ERR_STATUS_FUNC_2[8] – PCIESS2 protection violation
  - SEC_ERR_STATUS_FUNC_2[11] – MCASP1 protection violation
  - SEC_ERR_STATUS_FUNC_2[12] – MCASP2 protection violation
  - SEC_ERR_STATUS_FUNC_2[13] – MCASP3 protection violation
  - SEC_ERR_STATUS_FUNC_2[16] – EDMA_TPTC1 protection violation
  - SEC_ERR_STATUS_FUNC_2[17] – EDMA_TPCC protection violation
  - SEC_ERR_STATUS_FUNC_2[22] – QSPI protection violation
  - SEC_ERR_STATUS_FUNC_2[23] – EDMA_TPTC2 protection violation
- In debug mode:
  - SEC_ERR_STATUS_DEBUG_1[1] – L3 RAM protection violation
  - SEC_ERR_STATUS_DEBUG_1[2] – GPMC protection violation
  - SEC_ERR_STATUS_DEBUG_1[3] – EMIF protection violation
  - SEC_ERR_STATUS_DEBUG_1[4] – IVA protection violation
  - SEC_ERR_STATUS_DEBUG_1[5] – IPU protection violation
  - SEC_ERR_STATUS_DEBUG_1[6] – IVA SL2 protection violation
  - SEC_ERR_STATUS_DEBUG_1[8] – SYSTEM DMA protection violation
  - SEC_ERR_STATUS_DEBUG_1[10] – DSPDMA slave NIU protection violation
  - SEC_ERR_STATUS_DEBUG_1[13] – GPU protection violation
  - SEC_ERR_STATUS_DEBUG_1[14] – DSS protection violation
  - SEC_ERR_STATUS_DEBUG_1[16] – L4_PER1 protection violation
  - SEC_ERR_STATUS_DEBUG_1[17] – L4_CFG protection violation
  - SEC_ERR_STATUS_DEBUG_1[18] – DEBUG subsystem protection violation
  - SEC_ERR_STATUS_DEBUG_1[22] – L4_WKUP protection violation
  - SEC_ERR_STATUS_DEBUG_1[23] – BB2D protection violation
  - SEC_ERR_STATUS_DEBUG_1[26] – CT_TBR protection violation
  - SEC_ERR_STATUS_DEBUG_1[28] – EVE1 protection violation
  - SEC_ERR_STATUS_DEBUG_1[29] – EVE2 protection violation
  - SEC_ERR_STATUS_DEBUG_2[0] – DSP1 protection violation
  - SEC_ERR_STATUS_DEBUG_2[1] – DSP2 protection violation
  - SEC_ERR_STATUS_DEBUG_2[2] – OCMC_RAM2 protection violation

- SEC_ERR_STATUS_DEBUG_2[3] – OCMC_RAM3 protection violation
- SEC_ERR_STATUS_DEBUG_2[4] – L4_PER2 protection violation
- SEC_ERR_STATUS_DEBUG_2[5] – L4_PER3 protection violation
- SEC_ERR_STATUS_DEBUG_2[6] – IPU2 protection violation
- SEC_ERR_STATUS_DEBUG_2[7] – PCIESS1 protection violation
- SEC_ERR_STATUS_DEBUG_2[8] – PCIESS2 protection violation
- SEC_ERR_STATUS_DEBUG_2[11] – McASP1 protection violation
- SEC_ERR_STATUS_DEBUG_2[12] – McASP2 protection violation
- SEC_ERR_STATUS_DEBUG_2[13] – McASP3 protection violation
- SEC_ERR_STATUS_DEBUG_2[16] – EDMA_TPTC1 protection violation
- SEC_ERR_STATUS_DEBUG_2[17] – EDMA_TPCC protection violation
- SEC_ERR_STATUS_DEBUG_2[22] – QSPI protection violation
- SEC_ERR_STATUS_DEBUG_2[23] – EDMA_TPTC2 protection violation

For more information, see Chapter 18, *Control Module*.

14.2.3.7.3.5 L3_MAIN Firewall Default Configuration

Table 14-21 summarizes the configuration of the L3_MAIN firewalls.

Table 14-21. L3_MAIN Firewalls Default Configurations

Device/Region: 0						
Permission Type	**Reset Value**	**Reset Value**	**Reset Type**	**Run Time**	**Firewall Register (where j = 0)**	**Control Module Register**
ACCESS_ PERMISSION	All	0xFFF	Exported	Configurable	MRM_PERMISSION_REGION_LOW_j[11:0]	CONTROL_CORE_L3_HW_FW_ EXPORTED_VALUES_CONF_LOC K_1[k] CONTROL_CORE_L3_HW_FW_ EXPORTED_VALUES_CONF_LOC K_2[j]
DEBUG_ PERMISSION	All	0xF	Exported	Configurable	MRM_PERMISSION_REGION_LOW_j[15:12]	CONTROL_CORE_L3_HW_FW_ EXPORTED_VALUES_CONF_LOC K_1[k] CONTROL_CORE_L3_HW_FW_ EXPORTED_VALUES_CONF_LOC K_2[j]
INITIATOR_ PERMISSION	All	0xFFF FFFFF	Tied	Configurable	MRM_PERMISSION_REGION_HIGH_j[31:0]	N/A

Note

For the values of k and j see Table 14-22, Table 14-13 and Table 14-14.

Table 14-22. Control Module Register – Factorization

Variable Value	Module Name	Regions
k		
[0]	GPMC	8
[3]	OCMC RAM1	16
[4]	DSS	8
[6]	GPU	1
[7]	IVAHD SL2IF	4
[8]	IVAHD CONFIG	1
[11]	EMIF	8
[12]	DEBUGSS	1
[13]	CT_TBR	1
[16]	EVE1	1
[17]	EVE2	1
[20]	PCIESS1	8

Table 14-22. Control Module Register – Factorization (continued)

Variable Value	Module Name	Regions
[21]	PCIESS2	8
[22]	IPU1	4
[23]	IPU2	4
[24]	VCP1	1
[25]	VCP2	1
[26]	McASP1	1
[27]	McASP2	1
[28]	McASP3	1
[31]	BB2D	1
j		
0	DSP1	1
1	DSP2	1
2	OCMC RAM2	16
3	OCMC RAM3	16
8	QSPI	1
9	EDMA TC	1
10	EDMA TPCC	1

14.2.3.8 L3_MAIN Interconnect Error Handling

Error logging is enabled in the L3_MAIN interconnect. The three major types of errors are:

- Slave NIU errors
- Firewall errors (see Section 14.2.3.7.3.4, *L3_MAIN Firewall Error Logging*)
- Flag mux errors

14.2.3.8.1 Global Error-Routing Scheme

Figure 14-7 shows the L3_MAIN global error-routing scheme.

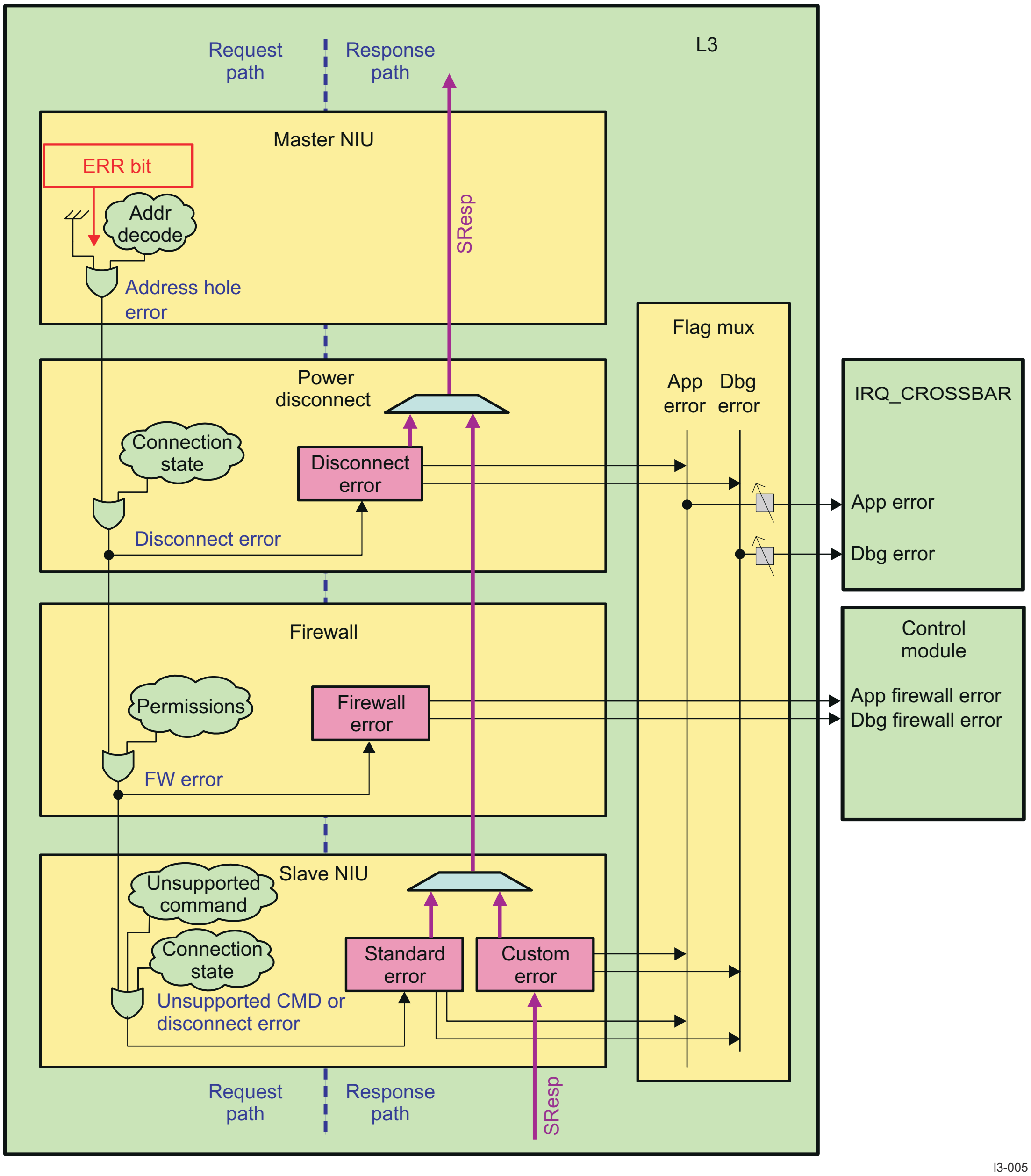


Figure 14-7. L3_MAIN Global Error-Routing Scheme

14.2.3.8.2 Slave NIU Error Logging

Error logging is implemented only at slave NIUs. Because the interconnect does not support master NIU error logging, an erroneous packet must be created and sent to one of the slave NIUs. The slave NIU that receives an erroneous packet is predictable but can change per master (see Table 14-23).

Table 14-23. L3_MAIN Connectivity and Holes Error Routing

Master	Connectivity and Holes Errors Logged Into Slave NIUs
All initiators except DSP1_CFG and DSP2_CFG	GPMC_TARG
DSP1_CFG	EVE1_TARG
DSP2_CFG	

The slave NIU can be configured to report standard errors (errors generated within the interconnect):

- Firewall error: Protection violation; this error indicates that a request was rejected by a firewall and is reported to the control module. For more information, see Section 14.2.3.7.3.4, *L3_MAIN Firewall Error Logging.*
- Address hole: This error reports an unknown address for a request. The address map is local to each master NIU; therefore, an address hole error is reported each time a master NIU requests an access to a slave NIU to which it is not logically connected, even if this address exists in the global L3_MAIN address map. This error is detected only once per burst.
- Unsupported commands: This error reports that the master NIU sent a command that cannot be processed, because the slave NIU cannot accept it and no conversion to another command is possible. This error is detected only once per burst.
- Report custom errors: Basically, when the slave answer is SResp = ERR
- Report severity level, for standard error and custom errors:
  - None: Error logging for this type of error is disabled.
  - Error: Error is logged for this type of error.
  - Fault: Error is logged and interrupt is generated for this type of error.
- Generate interrupt on 2 bits depending on the MReqDebug qualifier:
  - Application error - FAULT[0]
  - Debug error - FAULT[1]

  By default, all slave NIUs are configured with standard and custom error levels set to FAULT. The errors are reported on the two flag muxes (see Figure 14-7), depending on the access type, application or debug. For more information, see Section 14.2.3.8.3, *Flag Mux Error Logging*.

  The slave NIU power-disconnect component also has error logging enabled, because in this case the slave NIU is in a clock domain that is switched off and therefore cannot catch the error. By nature, this component can generate only standard errors. By default, it is configured with the error level set to FAULT.

  Wake up on demand: If an error packet reaches a slave NIU that is set with MDiscBehave = 1 (wake up on demand), then the active signal is asserted and L3 processes the error generation when the slave is awake. Although this is inefficient, it simplifies NIU implementation and should not be a problem because errors are supposed to occur only during software debug.

14.2.3.8.3 Flag Mux Error Logging

All fault signals are sent to a flag mux component. There are four important FLAGMUX registers:

- L3_FLAGMUX_MASK0: Masks application error sources
- L3_FLAGMUX_MASK1: Masks debug error sources
- L3_FLAGMUX_REGERR0: Checks which application error sources are active
- L3_FLAGMUX_REGERR1: Checks which debug error sources are active

The two L3_FLAGMUX_MASK registers mask bit 0 or bit 1 of the flag inputs, and the L3_FLAGMUX_REGERR registers read input errors. Each register is dedicated to reporting the bit corresponding to the register number.

Table 14-24 describes the mapping of the flags to the corresponding sources.

Table 14-24. Interconnect Flag Mapping

	Flag Mux Input	Source	
CLK1_1 Flag Mux	1	DMM_P1	
	2	DSP2 SDMA	
	3	EVE2	
	4	DMM_P2	
	6	DSP1 SDMA	
	7	EVE1	
	10	DSS	
	11	GPMC	
	12	PCIE SS1	
	13	IVA_CFG	
	14	IVA_SL2IF	
	15	L4_CFG	
	16	L4_WKUP	
	17	PCIE SS2	
	19	GPU	
	20	IPU1	
	21	IPU2	
	22	TPCC_EDMA	
	23	TC1_EDMA	
	24	TC2_EDMA	
	25	VCP1	
	26	L4_PER2_P3	
	27	L4_PER3_P3	
	28	MMU1	
	31	VCP2	
	CLK1_2 Flag Mux	0	HOST_CLK1_1
1		HOST_CLK1_2	
2		REPLICATOR_CLK1_TIMEOUT	
4		BB2D	
5		REPLICATOR_CLK1_TIMEOUT1	
6		L4_PER1_P3	
7		L4_PER1_P1	
8		L4_PER1_P2	
9		L4_PER2_P1	
10		L4_PER2_P2	
11		L4_PER3_P1	
12		L4_PER3_P2	
13		McASP1	
14		McASP2	
15		McASP3	
16		MMU2	
17		OCMC_RAM1	
18		OCMC_RAM2	
19		OCMC_RAM3	

Table 14-24. Interconnect Flag Mapping (continued)

	Flag Mux Input	Source
	21	QSPI
	23	CLK1_TARG_PWR_DISC_CLK2
CLK2_1 Flag Mux	0	L3_INSTR
	1	DEBUGSS_CT_TBR
	2	HOST_CLK2_1
	3	REPLICATOR_CLK2_TIMEOUT

Note

Missing inputs in Table 14-24 are reserved. Writing in them has no effect.

14.2.3.8.4 Severity Level of Standard and Custom Errors

The slave NIU registers are important for error logging.

- The L3_TARG_STDERRLOG_SVRTSTDLVL register shows the severity level for standard errors. According to the severity level, error logging is disabled, enabled with level ERROR, or enabled with level ERROR and flag FAULT.
- The L3_TARG_STDERRLOG_SVRTCUSTOMLVL register shows the severity level for custom errors.
- The L3_TARG_STDERRLOG_MAIN register (the main register for error-logging management) shows the validity of the logged information, standard or custom.
- The L3_TARG_STDERRLOG_HDR register stores packets in case of a standard error.
- The L3_TARG_STDERRLOG_MSTADDR register returns the MSTADDR field of the logged packet.
- The L3_TARG_STDERRLOG_SLVADDR register returns the SLVADDR field of the stored packet.
- The L3_TARG_STDERRLOG_INFO register saves the information field of the logged packet.

14.2.3.8.5 Example for Decoding Standard/Custom Errors Logged in L3_MAIN

Following is a procedure that must be followed for identifying Standard/Custom errors logged in L3_MAIN:

- Read the *_STDERRLOG_MAIN registers (L3_HOST_STDERRLOG_MAIN & L3_TARG_STDERRLOG_MAIN):
  – bit[0]STDERRLOG_MAIN_ERRLOGVLD = 0 -> NO ERROR
  – bit[0]STDERRLOG_MAIN_ERRLOGVLD = 1 -> ERROR
  – Error type is visible through:
    - bit[1]STDERRLOG_MAIN_ERRTYPE = 0 -> Standard Error (FW error, Address Hole, Unsupported command, OCP disconnect)
    - bit[1]STDERRLOG_MAIN_ERRTYPE = 0 -> Custom Error
- Standard Error:
  – Read L3_TARG_STDERRLOG_HDR[3:0] STDERRLOG_HDR_OPCODE
  – Read L3_TARG_STDERRLOG_MSTADDR[7:0] STDERRLOG_MSTADDR - > 8-bit NTTP master address used to distinguish the different initiators (see Table 14-10). That master address is also referred to as ConnID or MConnID.
  – Read L3_TARG_STDERRLOG_SLVADDR[4:0] STDERRLOG_SLVADDR -> NTTP Slave Address of the logged packet
  – Read L3_TARG_STDERRLOG_INFO[7:0]STDERRLOG_INFO -> see Section 14.1.1, *Terminology* for details.

The procedure for Custom error is the same, but the *_STDERRLOG_CUSTOMINFO_* registers should be read.

14.2.4 L3_MAIN Interconnect Programming Guide

14.2.4.1 L3 _MAIN Interconnect Low-Level Programming Models

This section describes the low-level hardware programming sequences for configuring and using the L3_MAIN interconnect module.

14.2.4.1.1 Global Initialization

14.2.4.1.1.1 Global Initialization of Surrounding Modules

This section identifies the requirements for initializing the surrounding modules when the L3_MAIN interconnect module is to be used for the first time after a device reset. The initialization of surrounding modules is based on the integration and environment of the L3_MAIN interconnect. For more information, see Section 14.2.2, *L3_MAIN Interconnect Integration* .

Table 14-25. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	For information about the configuration of the PRCM module, see Chapter 3, *Power, Reset, and Clock Management*.
Control module	For information about the configuration of the control module, see Chapter 18, *Control Module*.
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 17, *Interrupt Controllers*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.

14.2.4.2 Operational Modes Configuration

14.2.4.2.1 L3_MAIN Interconnect Error Analysis Mode

14.2.4.2.1.1 Main Sequence: L3_MAIN Interconnect Error Analysis Mode

The information required to analyze an error source is logged in several registers. The number of registers to access depends on the error source.

Figure 14-8 shows the software sequence required in most cases.

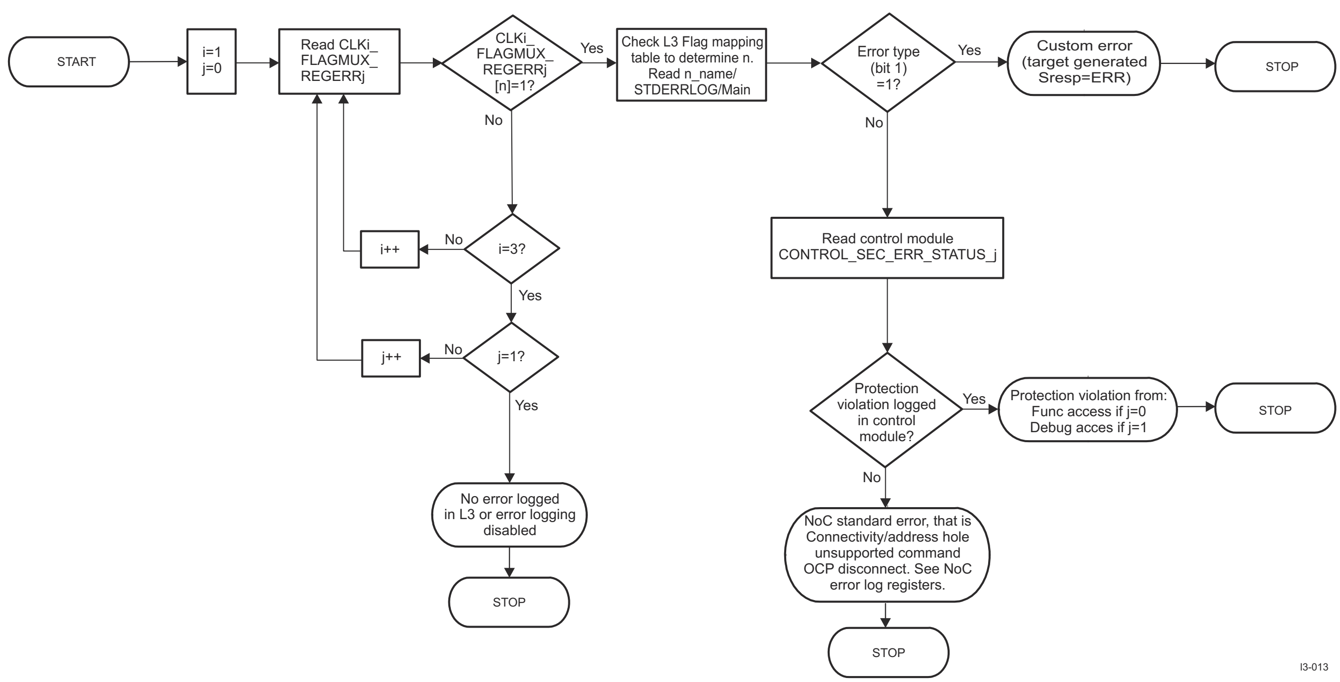


Figure 14-8. Typical Error Analysis Sequence

Note

In case flag is set due to timeout flagmux, user can check the status in CONTROL_SEC_ERR_STATUS for that Timeout Flagmux.

Table 14-26 lists the subprocess call summary for error analysis mode in the main sequence.

Table 14-26. Subprocess Call Summary for Main Sequence – Error Analysis Mode

Subprocess	Cross-Reference
L3 interconnect error analysis	See Section 14.2.4.2.1, *L3_MAIN Interconnect Error Analysis Mode.*
L3_MAIN interconnect protection violation error identification	See Section 14.2.4.2.1.1.2, *Subsequence: L3_MAIN Interconnect Protection Violation Error Identification.*
L3_MAIN interconnect unsupported command/address hole error identification	See Section 14.2.4.2.1.1.3, *Subsequence: L3_MAIN Interconnect Standard Error Identification.*
L3_MAIN interconnect reset FLAGMUX and module	See Section 14.2.4.2.1.1.4, *Subsequence: L3_MAIN Interconnect FLAGMUX Configuration.*

14.2.4.2.1.1.1 Subsequence: L3_MAIN Custom Error Identification

The procedure listed in Table 14-27 describes custom error identification.

Table 14-27. Custom Error Identification

Step	Register/Bit Field/Programming Model	Value
IF: Is custom error detected?	L3_TARG_STDERRLOG_MAIN[1] STDERRLOG_MAIN_ERRTYPE	=0x1
Read information field of the response packet.	L3_TARG_STDERRLOG_CUSTOMINFO_INFO[7:0] STDERRLOG_CUSTOMINFO_INFO	xxx
Read the address of the initiator that caused error.	L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR[10:0] STDERRLOG_CUSTOMINFO_MSTADDR	xxx
Read the type of operation (read/write).	L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE[1:0] STDERRLOG_CUSTOMINFO_OPCODE	xxx
ENDIF		

14.2.4.2.1.1.2 Subsequence: L3_MAIN Interconnect Protection Violation Error Identification

The procedure listed in Table 14-28 describes protection violation error identification and where it is logged in the control module registers. Two types of errors are logged: application errors and debug errors.

Table 14-28. L3_MAIN Protection Violation Error Identification

Step	Register/Bit Field/Programming Model	Value
Read the burst violation.	ERROR_LOG_k[23] 2D_BURST_VIOLATION	xxx
Read the initiator ID.	ERROR_LOG_k[3:0] ConnID	xxx
Read the command that caused the error.	ERROR_LOG_k[7] MCMD	xxx
Read the address of the request that caused the error.	ERROR_LOG_k[31:12] SLVOFS_LOGICAL	xxx
IF: Is it an application error?	L3_FLAGMUX_REGERR0 [31:0] REGERROR0	0x0
Read the status bits to see which module firewall has worked.	CONTROL.SEC_ERR_STATUS_FUNC_1[20:0]	xxx
Clear the status bits.	CONTROL.SEC_ERR_STATUS_FUNC_1[20:0]	xxx
Clear the status bit.	L3_TARG_STDERRLOG_MAIN [31] STDERRLOG_MAIN_CLRLOG	0x0
ELSE IF	L3_FLAGMUX_REGERR1[7:0] REGERROR1	= 0x1
Read the status bits to see the module.	CONTROL.SEC_ERR_STATUS_DEBUG_1[20:0]	xxx
Clear the status bits.	CONTROL.SEC_ERR_STATUS_DEBUG_1[20:0]	xxx
Clear the status bit.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG	0x0
ENDIF		

Table 14-28. L3_MAIN Protection Violation Error Identification (continued)

Step	Register/Bit Field/Programming Model	Value
Clear the burst violation.	ERROR_LOG_k[23] 2D_BURST_VIOLATION	0x0
Clear the error status.	ERROR_LOG_k[21:17] REGION_START_ERRLOG	0x00

14.2.4.2.1.1.3 Subsequence: L3_MAIN Interconnect Standard Error Identification

The procedure listed in Table 14-29 describes the identification of standard errors inside the L3_MAIN interconnect. The standard errors are: unsupported command, address hole, and disconnect.

Table 14-29. L3_MAIN Standard Error Identification

Step	Register/Bit Field/Programming Model	Value
IF: Is an error detected?	L3_TARG_STDERRLOG_MAIN[18] STDERRLOG_MAIN_ERRCNT	= 0x1
Read the corresponding flag.	L3_FLAGMUX_REGERR0[31:0] REGERROR0	xxx
Read the corresponding flag.	L3_FLAGMUX_REGERR1[31:0] REGERROR1	xxx
Localize the slave NIU that generated the error.	See Table 14-24.	
ELSE		
Clear the error log.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG	0x0
Clear the severity error status.	L3_TARG_STDERRLOG_SVRTSTDLVL[1:0] STDERRLOG_SVRTSTDLVL_0	0x2
ENDIF		

14.2.4.2.1.1.4 Subsequence: L3_MAIN Interconnect FLAGMUX Configuration

The procedures listed in Table 14-30 and Table 14-31 give information about the configuration of FLAGMUX masks.

Table 14-30. FLAGMUX Configuration

Step	Register/Bit Field/Programming Model	Value
Set the FLAGMUX masks to mask an event.	L3_FLAGMUX_MASK031:0] MASK0 L3_FLAGMUX_MASK1[31:0] MASK1	xxx
Read the REGERR bits to see if an error is recorded.	L3_FLAGMUX_REGERR0[31:0] REGERR0 L3_FLAGMUX_REGERR1[31:0] REGERR1	xxx
Clear the slave NIU error log and the FLAGMUX error.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_SVRTSTDLVL_0	0x1

Table 14-31. FLAGMUX Time-out Configuration (for CLK1_1 and CLK1_2)

Step	Register/Bit Field/Programming Model	Value
Enable time-out for target.	L3_FLAGMUX_TIMEOUTx_MASK0[24:0] MASK0 (1)	xxx
Read the REGERR bits to see if an error is recorded.	L3_FLAGMUX_TIMEOUTx_REGERR0[24:0] REGERR0 (1)	xxx

(1) x = 1 for CLK1_FLAGMUX_CLK1_1
x=2 for CLK1_FLAGMUX_CLK1_2
Bit fields are [24:0] for CLK1_FLAGMUX_CLK1_1, [20:0] for CLK1_FLAGMUX_CLK1_2

Table 14-32. FLAGMUX Time-out Configuration (for CLK2_1)

Step	Register/Bit Field/Programming Model	Value
Enable time-out for target.	L3_FLAGMUX_TIMEOUT_MASK0[1:0] MASK0	xxx
Read the REGERR bits to see if an error is recorded.	L3_FLAGMUX_TIMEOUT_REGERR0[1:0] REGERR0	xxx

14.2.5 L3_MAIN Interconnect Register Manual

14.2.5.1 L3_MAIN Register Group Summary

The registers in the L3 interconnect are divided into eight groups:

- Firewall registers (see Section 14.2.5.1.1)
- HOST registers (see Section 14.2.5.1.2)
- TARG registers (see Section 14.2.5.1.3)
- FLAGMUX registers (see Section 14.2.5.1.4 thru Section 14.2.5.1.6)
- BW registers (see Section 14.2.5.1.7 and Section 14.2.5.1.8.1)
- STATCOLL registers (see Section 14.2.5.1.9)

14.2.5.1.1 L3_MAIN Firewall Registers Summary and Description

Table 14-33. L3_MAIN Firewall Instance Summary

Module Name	Base Address	Size
DEBUGSS_CT_TBR_FW	0x4A22 4000	4KiB
DSP1_SDMA_FW	0x4A17 1000	4KiB
DSP2_SDMA_FW	0x4A17 3000	4KiB
DSS_FW	0x4A21 C000	4KiB
EVE1_FW	0x4A15 1000	4KiB
EVE2_FW	0x4A15 3000	4KiB
GPMC_FW	0x4A21 0000	4KiB
GPU_FW	0x4A21 4000	4KiB
BB2D_FW	0x4A21 A000	4KiB
IPU1_FW	0x4A15 B000	4KiB
IPU2_FW	0x4A21 8000	4KiB
IVA_CONFIG_FW	0x4A22 0000	4KiB
IVA_SL2IF_FW	0x4A21 E000	4KiB
L3_INSTR_FW	0x4A22 6000	4KiB
MCASP1_FW	0x4A16 7000	4KiB
MCASP2_FW	0x4A16 9000	4KiB
MCASP3_FW	0x4A16 B000	4KiB
OCMC_RAM1_FW	0x4A21 2000	4KiB
OCMC_RAM2_FW	0x4A20 E000	4KiB
OCMC_RAM3_FW	0x4A22 A000	4KiB
EMIF_OCP_FW	0x4A20 C000	4KiB
MA_MPU_NTTP_FW	0x4A20 A000	4KiB
PCIE1_FW	0x4A16 5000	4KiB
PCIE2_FW	0x4A15 9000	4KiB
QSPI_FW	0x4A17 9000	4KiB
EDMA_TPCC_FW	0x4A16 1000	4KiB
TPTC_FW	0x4A16 3000	4KiB
VCP1_FW	0x4A15 D000	4KiB
VCP2_FW	0x4A15 F000	4KiB

14.2.5.1.1.1 L3_MAIN Firewall Registers Summary

Table 14-34. L3_MAIN Firewall Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	DEBUGSS_CT_TBR_FW L3_MAIN Physical Address	DSP1_SDMA_FW L3_MAIN Physical Address	DSP2_SDMA_FW L3_MAIN Physical Address
ERROR_LOG_k [(2)]	RW	32	0x000+(0x10*k)	0x4A22 4000 + (0x10*k)	0x4A17 1000 + (0x10*k)	0x4A17 3000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k [(2)]	RO	32	0x004+(0x10*k)	0x4A22 4004 + (0x10*k)	0x4A17 1004 + (0x10*k)	0x4A17 3004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A22 4040	0x4A17 1040	0x4A17 3040
START_REGION_i	RW	32	0x080+(0x10*i)	-	-	-
END_REGION_i	RW	32	0x084+(0x10*i)	-	-	-
MRM_PERMISSION_REGION_HIGH_j [(1)]	RW	32	0x08C+(0x10*j)	0x4A22 408C + (0x10*j)	0x4A17 108C + (0x10*j)	0x4A17 308C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j [(1)]	RW	32	0x088+(0x10*j)	0x4A22 4088 + (0x10*j)	0x4A17 1088 + (0x10*j)	0x4A17 3088 + (0x10*j)

(1) j = 0 for DEBUGSS_CT_TBR_FW
j = 0 for DSP1_SDMA_FW
j = 0 for DSP2_SDMA_FW
(2) k = 0 for DEBUGSS_CT_TBR_FW
k = 0 for DSP1_SDMA_FW
k = 0 for DSP2_SDMA_FW

Table 14-35. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSS_FW L3_MAIN Physical Address	EVE1_FW L3_MAIN Physical Address	EVE2_FW L3_MAIN Physical Address	GPMC_FW L3_MAIN Physical Address
ERROR_LOG_k [(1)]	RW	32	0x000+(0x10*k)	0x4A21 C000 + (0x10*k)	0x4A15 1000 + (0x10*k)	0x4A15 3000 + (0x10*k)	0x4A21 0000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k [(1)]	RO	32	0x004+(0x10*k)	0x4A21 C004 + (0x10*k)	0x4A15 1004 + (0x10*k)	0x4A15 3004 + (0x10*k)	0x4A21 0004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A21 C040	0x4A15 1040	0x4A15 3040	0x4A21 0040
START_REGION_i [(2)]	RW	32	0x080+(0x10*i)	0x4A21 C080 + (0x10*i)	-	-	0x4A21 0080 + (0x10*i)
END_REGION_i [(2)]	RW	32	0x084+(0x10*i)	0x4A21 C084 + (0x10*i)	-	-	0x4A21 0084 + (0x10*i)
MRM_PERMISSION_REGION_HIGH_j [(3)]	RW	32	0x08C+(0x10*j)	0x4A21 C08C + (0x10*j)	0x4A15 108C + (0x10*j)	0x4A15 308C+ (0x10*j)	0x4A21 008C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j [(3)]	RW	32	0x088+(0x10*j)	0x4A21 C088 + (0x10*j)	0x4A15 1088 + (0x10*j)	0x4A15 3088+ (0x10*j)	0x4A21 0088 + (0x10*j)

(1) k = 0 for DSS_FW
k = 0 for EVE1_FW
k = 0 for EVE2_FW
k = 0 for GPMC_FW
(2) i = 1 to 7 for DSS_FW
i = 1 to 7 for GPMC_FW
(3) j = 0 to 7 for DSS_FW
j = 0 for EVE1_FW
j = 0 for EVE2_FW
j = 0 to 7 for GPMC_FW

Table 14-36. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPU_FW L3_MAIN Physical Address	BB2D_FW L3_MAIN Physical Address	IPU1_FW L3_MAIN Physical Address	IPU2_FW L3_MAIN Physical Address
ERROR_LOG_k [(2)]	RW	32	0x000+(0x10*k)	0x4A21 4000 + (0x10*k)	0x4A21 A000 + (0x10*k)	0x4A15 B000 + (0x10*k)	0x4A21 8000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k [(2)]	RO	32	0x004+(0x10*k)	0x4A21 4004 + (0x10*k)	0x4A21 A004 + (0x10*k)	0x4A15 B004 + (0x10*k)	0x4A21 8004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A21 4040	0x4A21 A040	0x4A15 B040	0x4A21 8040
START_REGION_i	RW	32	0x080+(0x10*i)	-	-	0x4A15 B080 + (0x10*i)	0x4A21 8080 + (0x10*i)
END_REGION_i	RW	32	0x084+(0x10*i)	-	-	0x4A15 B084 + (0x10*i)	0x4A21 8084 + (0x10*i)
MRM_PERMISSION_REGION_HIGH_j [(1)]	RW	32	0x08C+(0x10*j)	0x4A21 408C + (0x10*j)	0x4A21 A08C + (0x10*j)	0x4A15 B08C + (0x10*j)	0x4A21 808C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j [(1)]	RW	32	0x088+(0x10*j)	0x4A21 4088 + (0x10*j)	0x4A21 A088 + (0x10*j)	0x4A15 B088 + (0x10*j)	0x4A21 8088 + (0x10*j)

(1) j = 0 for GPU_FW
j = 0 for BB2D_FW
j = 0 to 3 for IPU1_FW
j = 0 to 3 for IPU2_FW
(2) k = 0 for GPU_FW
k = 0 for BB2D_FW
k = 0 for IPU1_FW
k = 0 for IPU2_FW

Table 14-37. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IVA_CONFIG_FW L3_MAIN Physical Address	IVA_SL2IF_FW L3_MAIN Physical Address	L3_INSTR_FW L3_MAIN Physical Address
ERROR_LOG_k [(3)]	RW	32	0x000+(0x10*k)	0x4A22 0000 + (0x10*k)	0x4A21 E000 + (0x10*k)	0x4A22 6000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k [(3)]	RO	32	0x004+(0x10*k)	0x4A22 0004 + (0x10*k)	0x4A21 E004 + (0x10*k)	0x4A22 6004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A22 0040	0x4A21 E040	0x4A22 6040
START_REGION_i [(1)]	RW	32	0x080+(0x10*i)	-	0x4A21 E080 + (0x10*i)	-
END_REGION_i [(1)]	RW	32	0x084+(0x10*i)	-	0x4A21 E084 + (0x10*i)	-
MRM_PERMISSION_REGION_HIGH_j [(2)]	RW	32	0x08C+(0x10*j)	0x4A22 008C + (0x10*j)	0x4A21 E08C + (0x10*j)	0x4A22 608C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j [(2)]	RW	32	0x088+(0x10*j)	0x4A22 0088 + (0x10*j)	0x4A21 E088 + (0x10*j)	0x4A22 6088 + (0x10*j)

(1) i = 1 to 3 for IVA_SL2IF_FW
(2) j = 0 for IVA_CONFIG_FW
j = 0 to 3 for IVA_SL2IF_FW
j= 0 for L3_INSTR_FW
(3) k = 0 for IVA_CONFIG_FW
k = 0 for IVA_SL2IF_FW
k = 0 for L3_INSTR_FW

Table 14-38. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_FW L3_MAIN Physical Address	MCASP2_FW L3_MAIN Physical Address	MCASP3_FW L3_MAIN Physical Address
ERROR_LOG_k [(1)]	RW	32	0x000+(0x10*k)	0x4A16 7000 + (0x10*k)	0x4A16 9000 + (0x10*k)	0x4A16 B000 + (0x10*k)

Table 14-38. L3_MAIN Firewall Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_FW L3_MAIN Physical Address	MCASP2_FW L3_MAIN Physical Address	MCASP3_FW L3_MAIN Physical Address
LOGICAL_ADDR_ERRLOG_k [(1)]	RO	32	0x004+(0x10*k)	0x4A16 7004 + (0x10*k)	0x4A16 9004 + (0x10*k)	0x4A16 B004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A16 7040	0x4A16 9040	0x4A16 B040
START_REGION_i	RW	32	0x080+(0x10*i)	-	-	-
END_REGION_i	RW	32	0x084+(0x10*i)	-	-	-
MRM_PERMISSION_REGION_HIGH_j [(2)]	RW	32	0x08C+(0x10*j)	0x4A16 708C + (0x10*j)	0x4A16 908C + (0x10*j)	0x4A16 B08C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j [(2)]	RW	32	0x088+(0x10*j)	0x4A16 7088 + (0x10*j)	0x4A16 9088 + (0x10*j)	0x4A16 B088 + (0x10*j)

(1) k = 0 for MCASP1_FW
k = 0 for MCASP2_FW
k = 0 for MCASP3_FW
(2) j = 0 for MCASP1_FW
j = 0 for MCASP2_FW
j = 0 for MCASP3_FW

Table 14-39. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM1_FW L3_MAIN Physical Address	OCMC_RAM2_FW L3_MAIN Physical Address	OCMC_RAM3_FW L3_MAIN Physical Address
ERROR_LOG_k [(1)]	RW	32	0x000+(0x10*k)	0x4A21 2000 + (0x10*k)	0x4A20 E000 + (0x10*k)	0x4A22 A000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k [(1)]	RO	32	0x004+(0x10*k)	0x4A21 2004 + (0x10*k)	0x4A20 E004 + (0x10*k)	0x4A22 A004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A21 2040	0x4A20 E040	0x4A22 A040
START_REGION_i [(2)]	RW	32	0x080+(0x10*i)	0x4A21 2080 + (0x10*i)	0x4A20 E080 + (0x10*i)	0x4A22 A080 + (0x10*i)
END_REGION_i [(2)]	RW	32	0x084+(0x10*i)	0x4A21 2084 + (0x10*i)	0x4A20 E084 + (0x10*i)	0x4A22 A084 + (0x10*i)
MRM_PERMISSION_REGION_HIGH_j [(3)]	RW	32	0x08C+(0x10*j)	0x4A21 208C + (0x10*j)	0x4A20 E08C + (0x10*j)	0x4A22 A08C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j [(3)]	RW	32	0x088+(0x10*j)	0x4A21 2088 + (0x10*j)	0x4A20 E088 + (0x10*j)	0x4A22 A088 + (0x10*j)

(1) k = 0 for OCMC_RAM1_FW
k = 0 for OCMC_RAM2_FW
k = 0 for OCMC_RAM3_FW
(2) i =1 to 15 for OCMC_RAM1_FW
i = 1 to 15 for OCMC_RAM2_FW
i = 1 to 15 for OCMC_RAM3_FW
(3) j = 0 to 15 for OCMC_RAM1_FW
j = 0 to 15 for OCMC_RAM2_FW
j = 0 to 15 for OCMC_RAM3_FW

Table 14-40. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMIF_OCP_FW L3_MAIN Physical Address	MA_MPU_NTTP_FW L3_MAIN Physical Address	PCIE1_FW L3_MAIN Physical Address
ERROR_LOG_k [(1)]	RW	32	0x000+(0x10*k)	0x4A20 C000 + (0x10*k)	0x4A20 A000 + (0x10*k)	0x4A16 5000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k [(1)]	RO	32	0x004+(0x10*k)	0x4A20 C004 + (0x10*k)	0x4A20 A004 + (0x10*k)	0x4A16 5004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A20 C040	0x4A20 A040	0x4A16 5040

Table 14-40. L3_MAIN Firewall Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF_OCP_FW L3_MAIN Physical Address	MA_MPU_NTTP_FW L3_MAIN Physical Address	PCIE1_FW L3_MAIN Physical Address
START_REGION_i (2)	RW	32	0x080+(0x10*i)	0x4A20 C080 + (0x10*i)	0x4A20 A080 + (0x10*i)	0x4A16 5080 + (0x10*i)
END_REGION_i (2)	RW	32	0x084+(0x10*i)	0x4A20 C084 + (0x10*i)	0x4A20 A084 + (0x10*i)	0x4A16 5084 + (0x10*i)
MRM_PERMISSION_REGION_HIGH_j (3)	RW	32	0x08C+(0x10*j)	0x4A20 C08C + (0x10*j)	0x4A20 A08C + (0x10*j)	0x4A16 508C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j (3)	RW	32	0x088+(0x10*j)	0x4A20 C088 + (0x10*j)	0x4A20 A088 + (0x10*j)	0x4A16 5088 + (0x10*j)

(1) k = 0 to 1 for EMIF_OCP_FW
k = 0 to 1 for MA_MPU_NTTP_FW
k = 0 for PCIE1_FW
(2) i= 1 to 7 for EMIF_OCP_FW
i = 1 to 7 for MA_MPU_NTTP_FW
i = 1 to 7 for PCIE1_FW
(3) j = 0 to 7 for EMIF_OCP_FW
j = 0 to 7 for MA_MPU_NTTP_FW
j = 0 to 7 for PCIE1_FW

Table 14-41. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIE2_FW L3_MAIN Physical Address	QSPI_FW L3_MAIN Physical Address	EDMA_TPCC_FW L3_MAIN Physical Address
ERROR_LOG_k (1)	RW	32	0x000+(0x10*k)	0x4A15 9000 + (0x10*k)	0x4A17 9000 + (0x10*k)	0x4A16 1000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k (1)	RO	32	0x004+(0x10*k)	0x4A15 9004 + (0x10*k)	0x4A17 9004 + (0x10*k)	0x4A16 1004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A15 9040	0x4A17 9040	0x4A16 1040
START_REGION_i (2)	RW	32	0x080+(0x10*i)	0x4A15 9080 + (0x10*i)	-	-
END_REGION_i (2)	RW	32	0x084+(0x10*i)	0x4A15 9084 + (0x10*i)	-	-
MRM_PERMISSION_REGION_HIGH_j (3)	RW	32	0x08C+(0x10*j)	0x4A15 908C + (0x10*j)	0x4A17 908C + (0x10*j)	0x4A16 108C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j (3)	RW	32	0x088+(0x10*j)	0x4A15 9088 + (0x10*j)	0x4A17 9088 + (0x10*j)	0x4A16 1088 + (0x10*j)

(1) k = 0 for PCIE2_FW
k = 0 for QSPI_FW
k = 0 for EDMA_TPCC_FW
(2) i = 1 to 7 for PCIE2_FW
(3) j = 0 to 7 for PCIE2_FW
j = 0 for QSPI_FW
j = 0 for EDMA_TPCC_FW

Table 14-42. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TPTC_FW L3_MAIN Physical Address	VCP1_FW L3_MAIN Physical Address	VCP2_FW L3_MAIN Physical Address
ERROR_LOG_k (1)	RW	32	0x000+(0x10*k)	0x4A16 3000+(0x10*k)	0x4A15 D000+(0x10*k)	0x4A15 F000+(0x10*k)
LOGICAL_ADDR_ERRLOG_k (1)	RO	32	0x004+(0x10*k)	0x4A16 3004+(0x10*k)	0x4A15 D004+(0x10*k)	0x4A15 F004+(0x10*k)
REGUPDATE_CONTROL	RW	32	0x040	0x4A16 3040	0x4A15 D040	0x4A15 F040
START_REGION_i (2)	RW	32	0x080+(0x10*i)	0x4A16 3080+(0x10*i)	-	-

Table 14-42. L3_MAIN Firewall Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	TPTC_FW L3_MAIN Physical Address	VCP1_FW L3_MAIN Physical Address	VCP2_FW L3_MAIN Physical Address
END_REGION_i (2)	RW	32	0x084+(0x10*i)	0x4A16 3084+(0x10*i)	-	-
MRM_PERMISSION_REGION_HIGH_j (2)	RW	32	0x08C+(0x10*j)	0x4A16 308C+(0x10*j)	0x4A15 D08C+(0x10*j)	0x4A15 F08C+(0x10*j)
MRM_PERMISSION_REGION_LOW_j (2)	RW	32	0x088+(0x10*j)	0x4A16 3088+(0x10*j)	0x4A15 D088+(0x10*j)	0x4A15 F088+(0x10*j)

(1) k = 0 to 1 for TPTC_FW
k = 0 for VCP1_FW
k = 0 for VCP2_FW
(2) i =1 for TPTC_FW

14.2.5.1.1.2 L3_MAIN Firewall Registers Description

Note

Hardware reset values can be modified by exported values from the control module at reset.

Table 14-43. ERROR_LOG_k

Address Offset	0x0000 0000+(0x10*k)	**Index**	See Table 14-34 to Table 14-42.
Physical Address		**Instance**	
Description	Error log register for port k		
Type	RW		

31:24	23	22	21:17	16:12	11:0
RESERVED	BLK_BURST_VIOLATION	RESERVED	REGION_START_ERRLOG	REGION_END_ERRLOG	REQINFO_ERRLOG

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0s.	R	0x00
23	BLK_BURST_VIOLATION	Read 0x1: 2D burst not allowed or exceeding allowed size Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0
22	RESERVED	Reads return 0s.	R	0
21:17	REGION_START_ERRLOG	Read: Wrong access hit this region number Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0x00
16:12	REGION_END_ERRLOG	Read: Wrong access hit this region number Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0x00
11:0	REQINFO_ERRLOG	Read: Error in reqinfo vector mapped as follows: [11: 8] ConnID [3:0] [7] MCMD [0] [6:4] Reserved [3] MReqDebug [2] Reserved [1] MReqSupervisor [0] MReqType Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0x000

Table 14-44. Register Call Summary for Register ERROR_LOG_k

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [0] [1] [2] [3] [4]
- L3_MAIN Interconnect Error Analysis Mode: [5] [6] [7] [8] [9] [10]
- L3_MAIN Firewall Registers Summary and Description: [11] [12] [13] [14] [15] [16] [17] [18] [20] [21] [22] [23] [24]

Table 14-45. LOGICAL_ADDR_ERRLOG_k

Address Offset	0x0000 0004+(0x10*k)	**Index**	See Table 14-34 to Table 14-42.
Physical Address		**Instance**	
Description	Logical Physical Address Error log register for port k		

Table 14-45. LOGICAL_ADDR_ERRLOG_k (continued)

Type	RO

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLVOFS_LOGICAL																															

Bits	Field Name	Description	Type	Reset
31[(1)]:0	SLVOFS_LOGICAL	Address generated by the Initiator before being translated	R	0x00000

(1) * = Size of the target

Table 14-46. Register Call Summary for Register LOGICAL_ADDR_ERRLOG_k

L3_MAIN Interconnect

- L3_MAIN Firewall Registers Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [9] [10] [11] [12] [13]

Table 14-47. REGUPDATE_CONTROL

Address Offset	0x0000 0040		
Physical Address		**Instance**	
Description	Register update control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												FW_ADDR_SPACE_MSB				RESERVED														FW_LOAD_REQ	BUSY_REQ

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads return 0s.	R	0x0000 0000
19:16	FW_ADDR_SPACE_MSB	Address space size	R	0x2
15:2	RESERVED	Reserved	R	0x0000 0000
1	FW_LOAD_REQ	Writing '1' to this bit causes the bit to self-clear and triggers the reload of L3 firewall default values. This bit will subsequently self-set when the reload procedure is complete. Writing '0' has no effect.	RW W1toClr	0x1
0	BUSY_REQ	Busy request 0x0: Allow transactions to reach the slave NIU (resume) 0x1: No transaction can reach the slave NIU (suspend)	RW	0x0

Table 14-48. Register Call Summary for Register REGUPDATE_CONTROL

L3_MAIN Interconnect

- L3_MAIN Firewall Reset: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
- L3_MAIN Firewall Functionality: [10] [11]
- L3_MAIN Firewall Registers Summary and Description: [12] [13] [14] [15] [16] [17] [18] [19] [21]

Table 14-49. START_REGION_i

Address Offset	0x0000 0080+(0x10*i)	**Index**	See Table 14-34 to Table 14-42.

Table 14-49. START_REGION_i (continued)

Physical Address	0x4A21 C080 + (0x10*i) 0x4A21 0080 + (0x10*i) 0x4A15 B080 + (0x10*i) 0x4A21 8080 + (0x10*i) 0x4A21 E080 + (0x10*i) 0x4A21 2080 + (0x10*i) 0x4A20 E080 + (0x10*i) 0x4A22 A080 + (0x10*i) 0x4A20 C080 + (0x10*i) 0x4A20 A080 + (0x10*i) 0x4A16 5080 + (0x10*i) 0x4A15 9080 + (0x10*i) 0x4A16 3080+(0x10*i)	**Instance**	DSS_FW GPMC_FW IPU1_FW IPU2_FW IVA_SL2IF_FW OCMC_RAM1_FW OCMC_RAM2_FW OCMC_RAM3_FW EMIF_OCP_FW MA_MPU_NTTP_FW PCIE1_FW PCIE2_FW TPTC_FW
Description	Start physical address of region i		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_REGION																						RESERVED									

Bits	Field Name	Description	Type	Reset
31:10	START_REGION	Physical target start address of firewall region i. The size of this bit field depends on target addressable space, the maximum is [31:10]. See Table 14-51. Each of the LSbits is assumed to be 0.The programmed address is included in the region i boundary.	RW	0x00000
9:0	RESERVED	Reads return 0s.	R	0x0000

Table 14-50. Register Call Summary for Register START_REGION_i

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
- L3_MAIN Firewall Registers Summary and Description: [11] [12] [13] [14] [15] [16] [17] [18] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33]

Table 14-51. Size of START_REGION_i[] START_REGION Bit Field

Firewall	Bit Field
DSS_FW	START_REGION_i[22:12] START_REGION
GPMC_FW	START_REGION_i[30:12] START_REGION
IPU1_FW	START_REGION_i[22:12] START_REGION
IPU2_FW	START_REGION_i[22:12] START_REGION
IVA_SL2IF_FW	START_REGION_i[17:12] START_REGION
OCMC_RAM1_FW	START_REGION_i[18:12] START_REGION
OCMC_RAM2_FW	START_REGION_i[19:12] START_REGION
OCMC_RAM3_FW	START_REGION_i[19:12] START_REGION
EMIF_OCP_FW	START_REGION_i[31:10] START_REGION
MA_MPU_NTTP_FW	START_REGION_i[31:10] START_REGION
PCIE1_FW	START_REGION_i[27:12] START_REGION
PCIE2_FW	START_REGION_i[27:12] START_REGION
TPTC_FW	START_REGION_i[19:12] START_REGION

Table 14-52. END_REGION_i

Address Offset	0x0000 0084+(0x10*i)	**Index**	See Table 14-34 to Table 14-42.

Table 14-52. END_REGION_i (continued)

Physical Address	0x4A21 C084 + (0x10*i) 0x4A21 0084 + (0x10*i) 0x4A15 B084 + (0x10*i) 0x4A21 8084 + (0x10*i) 0x4A21 E084 + (0x10*i) 0x4A21 2084 + (0x10*i) 0x4A20 E084 + (0x10*i) 0x4A22 A084 + (0x10*i) 0x4A20 C084 + (0x10*i) 0x4A20 A084 + (0x10*i) 0x4A16 5084 + (0x10*i) 0x4A15 9084 + (0x10*i) 0x4A16 3084+(0x10*i)	**Instance**	DSS_FW GPMC_FW IPU1_FW IPU2_FW IVA_SL2IF_FW OCMC_RAM1_FW OCMC_RAM2_FW OCMC_RAM3_FW EMIF_OCP_FW MA_MPU_NTTP_FW PCIE1_FW PCIE2_FW TPTC_FW
Description	End physical address of region i		
Type	RW		

31:10	9:2	1	0
END_REGION	RESERVED	END_REGION_i_ENABLE_CORE1	END_REGION_i_ENABLE_CORE0

Bits	Field Name	Description	Type	Reset
31:10	END_REGION	Physical target end address of firewall region i. The size of this bit field depends on target addressable space, the maximum is [31:10]. See Table 14-54. Each of the LSbits is assumed to be 1.The programmed address is included in the region i boundary.	RW	0x00000
9:2	RESERVED	Reads return 0s.	R	0x0000
1	END_REGION_i_ENABLE_CORE1	Enable this region for port 1(1).	RW	0x0
0	END_REGION_i_ENABLE_CORE0	Enable this region for port 0.	RW	0x0

(1) Only for multiport firewalls

Table 14-53. Register Call Summary for Register END_REGION_i

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21]
- L3_MAIN Firewall Registers Summary and Description: [23] [24] [25] [26] [27] [28] [29] [30] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45]

Table 14-54. Size of END_REGION_i[] END_REGION Bit Field

Firewall	Bit Field
DSS_FW	END_REGION_i[22:12] END_REGION
GPMC_FW	END_REGION_i[30:12] END_REGION
IPU1_FW	END_REGION_i[22:12] END_REGION
IPU2_FW	END_REGION_i[22:12] END_REGION
IVA_SL2IF_FW	END_REGION_i[17:12] END_REGION

Table 14-54. Size of END_REGION_i[] END_REGION Bit Field (continued)

Firewall	Bit Field
OCMC_RAM1_FW	END_REGION_i[18:12] END_REGION
OCMC_RAM2_FW	END_REGION_i[19:12] END_REGION
OCMC_RAM3_FW	END_REGION_i[19:12] END_REGION
EMIF_OCP_FW	END_REGION_i[31:10] END_REGION
MA_MPU_NTTP_FW	END_REGION_i[31:10] END_REGION
PCIE1_FW	END_REGION_i[27:12] END_REGION
PCIE2_FW	END_REGION_i[27:12] END_REGION
TPTC_FW	END_REGION_i[19:12] END_REGION

Table 14-55. MRM_PERMISSION_REGION_HIGH_j

Address Offset	0x0000 008C+(0x10*j)	**Index**	See Table 14-34 to Table 14-42.
Physical Address		**Instance**	
Description	Region j Permission High		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Master NIU ConnID = 115 write permission	RW	0x1
30	R15	Master NIU ConnID = 115 read permission	RW	0x1
29	W14	Master NIU ConnID = 14 write permission	RW	0x1
28	R14	Master NIU ConnID = 14 read permission	RW	0x1
27	W13	Master NIU ConnID = 13 write permission	RW	0x1
26	R13	Master NIU ConnID = 13 read permission	RW	0x1
25	W12	Master NIU ConnID = 12 write permission	RW	0x1
24	R12	Master NIU ConnID = 12 read permission	RW	0x1
23	W11	Master NIU ConnID = 11 write permission	RW	0x1
22	R11	Master NIU ConnID = 11 read permission	RW	0x1
21	W10	Master NIU ConnID = 10 write permission	RW	0x1
20	R10	Master NIU ConnID = 10 read permission	RW	0x1
19	W9	Master NIU ConnID = 9 write permission	RW	0x1
18	R9	Master NIU ConnID = 9 read permission	RW	0x1
17	W8	Master NIU ConnID = 8 write permission	RW	0x1
16	R8	Master NIU ConnID = 8 read permission	RW	0x1
15	W7	Master NIU ConnID = 7 write permission	RW	0x1
14	R7	Master NIU ConnID = 7 read permission	RW	0x1
13	W6	Master NIU ConnID = 6 write permission	RW	0x1
12	R6	Master NIU ConnID = 6 read permission	RW	0x1
11	W5	Master NIU ConnID = 5 write permission	RW	0x1
10	R5	Master NIU ConnID = 5 read permission	RW	0x1
9	W4	Master NIU ConnID = 4 write permission	RW	0x1
8	R4	Master NIU ConnID = 4 read permission	RW	0x1
7	W3	Master NIU ConnID = 3 write permission	RW	0x1
6	R3	Master NIU ConnID = 3 read permission	RW	0x1

Bits	Field Name	Description	Type	Reset
5	W2	Master NIU ConnID = 2 write permission	RW	0x1
4	R2	Master NIU ConnID = 2 read permission	RW	0x1
3	W1	Master NIU ConnID = 1 write permission	RW	0x1
2	R1	Master NIU ConnID = 1 read permission	RW	0x1
1	W0	Master NIU ConnID = 0 write permission	RW	0x1
0	R0	Master NIU ConnID = 0 read permission	RW	0x1

Table 14-56. Register Call Summary for Register MRM_PERMISSION_REGION_HIGH_j

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [0] [1] [2] [3] [4] [5]
- L3_MAIN Firewall Registers Summary and Description: [6] [7] [8] [9] [10] [11] [12] [13] [15]

Table 14-57. MRM_PERMISSION_REGION_LOW_j

Address Offset	0x0000 0088+(0x10*j)	**Index**	See Table 14-34 to Table 14-42.
Physical Address		**Instance**	
Description	Region j Permission Low		
Type	RW		

31:16	15	14	13:12	11	10	9	8	7	6	5:0
RESERVED	PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED	PUB_PRV_WRITE	PUB_PRV_READ	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	RESERVED	R	See Table 14-59.
15	PUB_PRV_DEBUG	Public Privilege Debug Allowed	RW	See Table 14-59.
14	PUB_USR_DEBUG	Public User Debug Allowed	RW	See Table 14-59.
13:12	RESERVED	RESERVED	R	See Table 14-59.
11	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	See Table 14-59.
10	PUB_PRV_READ	Public Privilege Read Allowed	RW	See Table 14-59.
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	See Table 14-59.
8	PUB_USR_READ	Public User Read Access Allowed	RW	See Table 14-59.
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	See Table 14-59.
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	See Table 14-59.
5:0	RESERVED	RESERVED	R	See Table 14-59.

Table 14-58. Register Call Summary for Register MRM_PERMISSION_REGION_LOW_j

L3_MAIN Interconnect

- L3_MAIN Firewall Reset: [0] [1]
- L3_MAIN Firewall Functionality: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]
- L3_MAIN Firewall Registers Summary and Description: [15] [16] [17] [18] [19] [20] [21] [22] [24]

Table 14-59. Reset Value for MRM_PERMISSION_REGION_LOW_j

Region	Reset Value
Region j = 0 (except EMIF firewall)	0xFFFF0000
Region j = 0 (for EMIF firewall)	0xFFFFFFFF
Region j 0 (for all firewalls)	0xFFFFFFFF

14.2.5.1.2 L3_MAIN Host Register Summary and Description

Table 14-60. HOST Instance Summary

Module Name	Base Address	Size
CLK1_HOST_CLK1_1	0x4400 0000	8MiB
CLK1_HOST_CLK1_2	0x4480 0000	8MiB
CLK2_HOST_CLK2_1	0x4500 0000	8MiB

14.2.5.1.2.1 L3_MAIN HOST Register Summary

Table 14-61. HOST Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_HOST_CLK1_1 L3_MAIN Physical Address	CLK1_HOST_CLK1_2 L3_MAIN Physical Address	CLK2_HOST_CLK2_1 L3_MAIN Physical Address
L3_HOST_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0000	0x4480 0000	0x4500 0000
L3_HOST_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0004	0x4480 0004	0x4500 0004
L3_HOST_STDHOSTHDR_MAINCTLREG	R	32	0x0000 0008	0x4400 0008	0x4480 0008	0x4500 0008
L3_HOST_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0040	0x4480 0040	0x4500 0040
L3_HOST_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0044	0x4480 0044	0x4500 0044
L3_HOST_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0048	0x4480 0048	0x4500 0048
L3_HOST_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 004C	0x4480 004C	0x4500 004C
L3_HOST_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0050	0x4480 0050	0x4500 0050
L3_HOST_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0054	0x4480 0054	0x4500 0054
L3_HOST_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0058	0x4480 0058	0x4500 0058
L3_HOST_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 005C	0x4480 005C	0x4500 005C
L3_HOST_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0060	0x4480 0060	0x4500 0060
L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0064	0x4400 0064	0x4480 0064	0x4500 0064
L3_HOST_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0068	0x4400 0068	0x4480 0068	0x4500 0068
L3_HOST_STDERRLOG_CUSTOMINFO_WR	R	32	0x0000 006C	0x4400 006C	0x4480 006C	0x4500 006C
L3_HOST_STDERRLOG_CUSTOMINFO_ADDR	R	32	0x0000 0070	0x4400 0070	0x4480 0070	0x4500 0070
L3_HOST_STDERRLOG_CUSTOMINFO_DECERR	R	32	0x0000 0074	0x4400 0074	0x4480 0074	0x4500 0074

14.2.5.1.2.2 L3_MAIN HOST Register Description

Table 14-62. L3_HOST_STDHOSTHDR_COREREG

Address Offset	See Table 14-61.		
Physical Address	0x4400 0000 0x4480 0000 0x4500 0000	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31:22	21:16	15:1	0
RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x1A.	R	0x1A
15:1	RESERVED		R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 14-63. Register Call Summary for Register L3_HOST_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-64. L3_HOST_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-61.		
Physical Address	0x4400 0004 0x4480 0004 0x4500 0004	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

STDHOSTHDR_VERSIONREG_REVISIONID	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 14-65. Register Call Summary for Register L3_HOST_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-66. L3_HOST_STDHOSTHDR_MAINCTLREG

Address Offset	See Table 14-61.		
Physical Address	0x4400 0008 0x4480 0008 0x4500 0008	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													STDHOSTHDR_MAINCTLREG_FLT	RESERVED	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	STDHOSTHDR_MAINCTLREG_FLT	Asserted when a Fault condition is detected: if the unit includes Error Logging, Fault is asserted when the Fault Control register field indicates a Fault, and de-asserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X.	R	0
1:0	RESERVED		R	0x0

Table 14-67. Register Call Summary for Register L3_HOST_STDHOSTHDR_MAINCTLREG

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-68. L3_HOST_STDERRLOG_SVRTSTDLVL

Address Offset	See Table 14-61.

Table 14-68. L3_HOST_STDERRLOG_SVRTSTDLVL (continued)

Physical Address	0x4400 0040 0x4480 0040 0x4500 0040	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	RW		

31–2	1–0
RESERVED	STDERRLOG_SVRTSTDLVL_0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	STDERRLOG_SVRTSTDLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

Table 14-69. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTSTDLVL

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-70. L3_HOST_STDERRLOG_SVRTCUSTOMLVL

Address Offset	See Table 14-61.		
Physical Address	0x4400 0044 0x4480 0044 0x4500 0044	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	RW		

31–2	1–0
RESERVED	STDERRLOG_SVRTCUSTOMLVL_0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	STDERRLOG_SVRTCUSTOMLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

Table 14-71. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTCUSTOMLVL

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-72. L3_HOST_STDERRLOG_MAIN

Address Offset	See Table 14-61.		
Physical Address	0x4400 0048 0x4480 0048 0x4500 0048	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	RW		

31	30:20	19	18	17:2	1	0
STDERRLOG_MAIN_CLRLOG	RESERVED	STDERRLOG_MAIN_FLTCNT	STDERRLOG_MAIN_ERRCNT	RESERVED	STDERRLOG_MAIN_ERRTYPE	STDERRLOG_MAIN_ERRLOGVLD

Bits	Field Name	Description	Type	Reset
31	STDERRLOG_MAIN_CLRLOG	Clears "Error Logging Valid" bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0.	RW	0
30:20	RESERVED		R	0x000
19	STDERRLOG_MAIN_FLTCNT	Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
18	STDERRLOG_MAIN_ERRCNT	Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
17:2	RESERVED		R	0x0000
1	STDERRLOG_MAIN_ERRTYPE	Indicates logging type. Type: Status. Reset value: X. Read 0x0: Logged Error format is standard (header and necker recorded). Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information.	R	0
0	STDERRLOG_MAIN_ERRLOGVLD	Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X.	R	0

Table 14-73. Register Call Summary for Register L3_HOST_STDERRLOG_MAIN

L3_MAIN Interconnect

- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [0]
- L3_MAIN Host Register Summary and Description: [1]

Table 14-74. L3_HOST_STDERRLOG_HDR

Address Offset	See Table 14-61.		
Physical Address	0x4400 004C 0x4480 004C 0x4500 004C	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			

Table 14-74. L3_HOST_STDERRLOG_HDR (continued)

Type R

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="10">STDERRLOG_HDR_LEN1</td><td colspan="2">RESERVED</td><td colspan="4">STDERRLOG_HDR_STOPOFSWRPSZ</td><td>STDERRLOG_HDR_ERR</td><td colspan="3">RESERVED</td><td colspan="2">STDERRLOG_HDR_PRESSURE</td><td colspan="2">RESERVED</td><td colspan="4">STDERRLOG_HDR_OPCODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x00
27:18	STDERRLOG_HDR_LEN1	This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X.	R	0x00
17:16	RESERVED		R	0x0
15:12	STDERRLOG_HDR_STOPOFSWRPSZ	StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X.	R	0x0
11	STDERRLOG_HDR_ERR	Err bit of the logged packet. Type: Status. Reset value: X.	R	0
10:8	RESERVED		R	0x0
7:6	STDERRLOG_HDR_PRESSURE	Pressure field of the logged packet. Type: Status. Reset value: X.	R	0
5:4	RESERVED		R	0x0
3:0	STDERRLOG_HDR_OPCODE	Opcode of the logged packet. Type: Status. Reset value: X. 0x0: Store without acknowledge, incrementing burst, non-atomic request 0x1: Store without acknowledge, wrapping burst, non-atomic request 0x2: Store with acknowledge, incrementing burst, non-atomic request 0x3: Store with acknowledge, wrapping burst, non-atomic request 0x4: Load, incrementing burst, non-atomic request 0x5: Load, wrapping burst, non-atomic request 0x6: Control packet 0x7: Flush 0x8: Store without acknowledge, incrementing burst, atomic request 0x9: Store without acknowledge, wrapping burst, atomic request 0xA: Store with acknowledge, incrementing burst, atomic request 0xB: Store with acknowledge, wrapping burst, atomic request 0xC: Load, incrementing burst, atomic request 0xD: Load, wrapping burst, atomic request 0xE: Reserved 0xF: Reserved	R	0x0

Table 14-75. Register Call Summary for Register L3_HOST_STDERRLOG_HDR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 14-76. L3_HOST_STDERRLOG_MSTADDR

Address Offset	See Table 14-61.		
Physical Address	0x4400 0050 0x4480 0050 0x4500 0050	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–8	7–0
RESERVED	STDERRLOG_MSTADDR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_MSTADDR	Master Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 14-77. Register Call Summary for Register L3_HOST_STDERRLOG_MSTADDR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 14-78. L3_HOST_STDERRLOG_SLVADDR

Address Offset	See Table 14-61.		
Physical Address	0x4400 0054 0x4480 0054 0x4500 0054	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–7	6–0
RESERVED	STDERRLOG_SLVADDR

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	STDERRLOG_SLVADDR	Slave Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 14-79. Register Call Summary for Register L3_HOST_STDERRLOG_SLVADDR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 14-80. L3_HOST_STDERRLOG_INFO

Address Offset	See Table 14-61.		
Physical Address	0x4400 0058 0x4480 0058 0x4500 0058	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STDERRLOG_INFO

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_INFO	Info field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 14-81. Register Call Summary for Register L3_HOST_STDERRLOG_INFO

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-82. L3_HOST_STDERRLOG_SLVOFSLSB

Address Offset	See Table 14-61.		
Physical Address	0x4400 005C 0x4480 005C 0x4500 005C	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–0
STDERRLOG_SLVOFSLSB

Bits	Field Name	Description	Type	Reset
31:0	STDERRLOG_SLVOFSLSB	LSB of the "slave offset" field, concatenated with "start offset" field of the logged packet. Type: Status. Reset value: X.	R	0x0000 0000

Table 14-83. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSLSB

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-84. L3_HOST_STDERRLOG_SLVOFSMSB

Address Offset	See Table 14-61.		
Physical Address	0x4400 0060 0x4480 0060 0x4500 0060	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–1	0
RESERVED	STDERRLOG_SLVOFSMSB

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000

Bits	Field Name	Description	Type	Reset
0	STDERRLOG_SLVOFSMSB	MSB of the "slave offset" field of the logged packet (according to NTTP packet format, this register field may exceed the actual "slave offset" size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X.	R	0

Table 14-85. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSMSB

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-86. L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR

Address Offset	See Table 14-61.		
Physical Address	0x4400 0064 0x4480 0064 0x4500 0064	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–8	7–0
RESERVED	STDERRLOG_CUSTOMINFO_MSTADDR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_CUSTOMINFO_MSTADDR	Type: Status. Reset value: X.	R	0x00

Table 14-87. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-88. L3_HOST_STDERRLOG_CUSTOMINFO_INFO

Address Offset	See Table 14-61.		
Physical Address	0x4400 0068 0x4480 0068 0x4500 0068	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–8	7–0
RESERVED	STDERRLOG_CUSTOMINFO_INFO

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_CUSTOMINFO_INFO	Type: Status. Reset value: X.	R	0x00

Table 14-89. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_INFO

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-90. L3_HOST_STDERRLOG_CUSTOMINFO_WR

Address Offset	See Table 14-61.

Table 14-90. L3_HOST_STDERRLOG_CUSTOMINFO_WR (continued)

Physical Address	0x4400 006C 0x4480 006C 0x4500 006C	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–1	0
RESERVED	STDERRLOG_CUSTOMINFO_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_CUSTOMINFO_WR	Type: Status. Reset value: X.	R	0

Table 14-91. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_WR

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-92. L3_HOST_STDERRLOG_CUSTOMINFO_ADDR

Address Offset	See Table 14-61.		
Physical Address	0x4400 0070 0x4480 0070 0x4500 0070	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31–21	20–0
RESERVED	STDERRLOG_CUSTOMINFO_ADDR

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20:0	STDERRLOG_CUSTOMINFO_ADDR	Type: Status. Reset value: X.	R	0x000000

Table 14-93. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_ADDR

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 14-94. L3_HOST_STDERRLOG_CUSTOMINFO_DECERR

Address Offset	See Table 14-61.		
Physical Address	0x4400 0074 0x4480 0074 0x4500 0074	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1

Table 14-94. L3_HOST_STDERRLOG_CUSTOMINFO_DECERR (continued)

Description	
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STDERRLOG_CUSTOMINFO_DECERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_CUSTOMINFO_DECERR	Type: Status. Reset value: X.	R	0

Table 14-95. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_DECERR

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

14.2.5.1.3 L3_MAIN TARG Register Summary and Description

Table 14-96. L3_MAIN TARG Instance Summary

Module Name	Base Address	Size
GPMC_TARG	0x4400 0100	4KiB
DMM_P1_TARG	0x4400 0200	4KiB
DSP1_SDMA_TARG	0x4400 0300	4KiB
L4_CFG_TARG	0x4400 0500	4KiB
DSP2_SDMA_TARG	0x4400 0600	4KiB
VCP1_TARG	0x4400 0700	4KiB
VCP2_TARG	0x4400 0800	4KiB
BB2D_TARG	0x4400 0900	4KiB
EVE1_TARG	0x4400 0A00	4KiB
EVE2_TARG	0x4400 0B00	4KiB
L4_PER3_P3_TARG	0x4400 0E00	4KiB
OCMC_RAM1_TARG	0x4400 0F00	4KiB
IPU1_TARG	0x4400 1000	4KiB
IPU2_TARG	0x4400 1100	4KiB
GPU_TARG	0x4400 1200	4KiB
DMM_P2_TARG	0x4400 1300	4KiB
IVA_CONFIG_TARG	0x4400 1600	4KiB
OCMC_RAM2_TARG	0x4400 1700	4KiB
IVA_SL2IF_TARG	0x4400 1800	4KiB

Table 14-96. L3_MAIN TARG Instance Summary (continued)

Module Name	Base Address	Size
OCMC_RAM3_TARG	0x4400 1900	4KiB
L4_PER1_P1_TARG	0x4400 1C00	4KiB
L4_WKUP_TARG	0x4400 1D00	4KiB
L4_PER1_P2_TARG	0x4400 1F00	4KiB
TPCC_TARG	0x4400 2000	4KiB
L4_PER1_P3_TARG	0x4400 2100	4KiB
MMU1_TARG	0x4400 2200	4KiB
L4_PER2_P1_TARG	0x4400 2300	4KiB
L4_PER2_P2_TARG	0x4400 2400	4KiB
L4_PER2_P3_TARG	0x4400 2500	4KiB
L4_PER3_P1_TARG	0x4400 2600	4KiB
L4_PER3_P2_TARG	0x4400 2700	4KiB
MMU2_TARG	0x4400 2800	4KiB
DSS_TARG	0x4400 2900	4KiB
TPTC2_TARG	0x4400 2B00	4KiB
TPTC1_TARG	0x4400 2E00	4KiB
MCASP1_TARG	0x4400 2F00	4KiB
MCASP2_TARG	0x4400 3000	4KiB
MCASP3_TARG	0x4400 3100	4KiB
PCIE1_TARG	0x4400 3700	4KiB
PCIE2_TARG	0x4400 3800	4KiB
QSPI_TARG	0x4400 3900	4KiB
L3_INSTR	0x4500 0100	4KiB
DEBUGSS_CT_TBR_TARG	0x4500 0300	4KiB

14.2.5.1.3.1 L3_MAIN TARG Register Summary

Table 14-97. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPMC_TARG L3_MAIN Physical Address	DMM_P1_TARG L3_MAIN Physical Address	DSP1_SDMA_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0100	0x4400 0200	0x4400 0300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0104	0x4400 0204	0x4400 0304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0108	0x4400 0208	0x4400 0308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0110	0x4400 0210	0x4400 0310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0140	0x4400 0240	0x4400 0340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0144	0x4400 0244	0x4400 0344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0148	0x4400 0248	0x4400 0348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 014C	0x4400 024C	0x4400 034C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0150	0x4400 0250	0x4400 0350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0154	0x4400 0254	0x4400 0354
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0158	0x4400 0258	0x4400 0358
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 015C	0x4400 025C	0x4400 035C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0160	0x4400 0260	0x4400 0360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0164	0x4400 0264	0x4400 0364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0168	0x4400 0268	0x4400 0368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 016C	0x4400 026C	0x4400 036C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0180	0x4400 0280	0x4400 0380

Table 14-98. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG_TARG L3_MAIN Physical Address	DSP2_SDMA_TARG L3_MAIN Physical Address	VCP1_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0500	0x4400 0600	0x4400 0700
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0504	0x4400 0604	0x4400 0704
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0508	0x4400 0608	0x4400 0708
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0510	0x4400 0610	0x4400 0710
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0540	0x4400 0640	0x4400 0740
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0544	0x4400 0644	0x4400 0744
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0548	0x4400 0648	0x4400 0748
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 054C	0x4400 064C	0x4400 074C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0550	0x4400 0650	0x4400 0750

Table 14-98. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG_TARG L3_MAIN Physical Address	DSP2_SDMA_TARG L3_MAIN Physical Address	VCP1_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0554	0x4400 0654	0x4400 0754
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0558	0x4400 0658	0x4400 0758
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 055C	0x4400 065C	0x4400 075C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0560	0x4400 0660	0x4400 0760
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0564	0x4400 0664	0x4400 0764
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0568	0x4400 0668	0x4400 0768
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 056C	0x4400 066C	0x4400 076C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0580	0x4400 0680	0x4400 0780

Table 14-99. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	VCP2_TARG L3_MAIN Physical Address	BB2D_TARG L3_MAIN Physical Address	EVE1_TARG L3_MAIN Physical Address	EVE2_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0800	0x4400 0900	0x4400 0A00	0x4400 0B00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0804	0x4400 0904	0x4400 0A04	0x4400 0B04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0808	0x4400 0908	0x4400 0A08	0x4400 0B08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0810	0x4400 0910	0x4400 0A10	0x4400 0B10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0840	0x4400 0940	0x4400 0A40	0x4400 0B40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0844	0x4400 0944	0x4400 0A44	0x4400 0B44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0848	0x4400 0948	0x4400 0A48	0x4400 0B48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 084C	0x4400 094C	0x4400 0A4C	0x4400 0B4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0850	0x4400 0950	0x4400 0A50	0x4400 0B50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0854	0x4400 0954	0x4400 0A54	0x4400 0B54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0858	0x4400 0958	0x4400 0A58	0x4400 0B58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 085C	0x4400 095C	0x4400 0A5C	0x4400 0B5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0860	0x4400 0960	0x4400 0A60	0x4400 0B60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0864	0x4400 0964	0x4400 0A64	0x4400 0B64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0868	0x4400 0968	0x4400 0A68	0x4400 0B68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 086C	0x4400 096C	0x4400 0A6C	0x4400 0B6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0880	0x4400 0980	0x4400 0A80	0x4400 0B80

Table 14-100. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER3_P3_TARG L3_MAIN Physical Address	OCMC_RAM1_TARG L3_MAIN Physical Address	IPU1_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0E00	0x4400 0F00	0x4400 1000
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0E04	0x4400 0F04	0x4400 1004
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0E08	0x4400 0F08	0x4400 1008
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0E10	0x4400 0F10	0x4400 1010
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0E40	0x4400 0F40	0x4400 1040
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0E44	0x4400 0F44	0x4400 1044
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0E48	0x4400 0F48	0x4400 1048
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 0E4C	0x4400 0F4C	0x4400 104C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0E50	0x4400 0F50	0x4400 1050
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0E54	0x4400 0F54	0x4400 1054
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0E58	0x4400 0F58	0x4400 1058
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 0E5C	0x4400 0F5C	0x4400 105C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0E60	0x4400 0F60	0x4400 1060
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0E64	0x4400 0F64	0x4400 1064
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0E68	0x4400 0F68	0x4400 1068
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 0E6C	0x4400 0F6C	0x4400 106C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0E80	0x4400 0F80	0x4400 1080

Table 14-101. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU2_TARG L3_MAIN Physical Address	GPU_TARG L3_MAIN Physical Address	DMM_P2_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 1100	0x4400 1200	0x4400 1300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 1104	0x4400 1204	0x4400 1304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 1108	0x4400 1208	0x4400 1308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 1110	0x4400 1210	0x4400 1310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 1140	0x4400 1240	0x4400 1340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 1144	0x4400 1244	0x4400 1344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 1148	0x4400 1248	0x4400 1348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 114C	0x4400 124C	0x4400 134C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 1150	0x4400 1250	0x4400 1350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 1154	0x4400 1254	0x4400 1354

Table 14-101. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IPU2_TARG L3_MAIN Physical Address	GPU_TARG L3_MAIN Physical Address	DMM_P2_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 1158	0x4400 1258	0x4400 1358
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 115C	0x4400 125C	0x4400 135C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 1160	0x4400 1260	0x4400 1360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 1164	0x4400 1264	0x4400 1364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 1168	0x4400 1268	0x4400 1368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 116C	0x4400 126C	0x4400 136C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 1180	0x4400 1280	0x4400 1380

Table 14-102. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	IVA_CONFIG_TARG L3_MAIN Physical Address	OCMC_RAM2_TARG L3_MAIN Physical Address	IVA_SL2IF_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 1600	0x4400 1700	0x4400 1800
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 1604	0x4400 1704	0x4400 1804
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 1608	0x4400 1708	0x4400 1808
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 1610	0x4400 1710	0x4400 1810
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 1640	0x4400 1740	0x4400 1840
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 1644	0x4400 1744	0x4400 1844
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 1648	0x4400 1748	0x4400 1848
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 164C	0x4400 174C	0x4400 184C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 1650	0x4400 1750	0x4400 1850
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 1654	0x4400 1754	0x4400 1854
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 1658	0x4400 1758	0x4400 1858
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 165C	0x4400 175C	0x4400 185C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 1660	0x4400 1760	0x4400 1860
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 1664	0x4400 1764	0x4400 1864
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 1668	0x4400 1768	0x4400 1868
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 166C	0x4400 176C	0x4400 186C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 1680	0x4400 1780	0x4400 1880

Table 14-103. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM3_TARG L3_MAIN Physical Address	L4_PER1_P1_TARG L3_MAIN Physical Address	L4_WKUP_TARG L3_MAIN Physical Address	L4_PER1_P2_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 1900	0x4400 1C00	0x4400 1D00	0x4400 1F00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 1904	0x4400 1C04	0x4400 1D04	0x4400 1F04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 1908	0x4400 1C08	0x4400 1D08	0x4400 1F08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 1910	0x4400 1C10	0x4400 1D10	0x4400 1F10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 1940	0x4400 1C40	0x4400 1D40	0x4400 1F40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 1944	0x4400 1C44	0x4400 1D44	0x4400 1F44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 1948	0x4400 1C48	0x4400 1D48	0x4400 1F48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 194C	0x4400 1C4C	0x4400 1D4C	0x4400 1F4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 1950	0x4400 1C50	0x4400 1D50	0x4400 1F50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 1954	0x4400 1C54	0x4400 1D54	0x4400 1F54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 1958	0x4400 1C58	0x4400 1D58	0x4400 1F58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 195C	0x4400 1C5C	0x4400 1D5C	0x4400 1F5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 1960	0x4400 1C60	0x4400 1D60	0x4400 1F60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 1964	0x4400 1C64	0x4400 1D64	0x4400 1F64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 1968	0x4400 1C68	0x4400 1D68	0x4400 1F68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 196C	0x4400 1C6C	0x4400 1D6C	0x4400 1F6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 1980	0x4400 1C80	0x4400 1D80	0x4400 1F80

Table 14-104. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	TPCC_TARG L3_MAIN Physical Address	L4_PER1_P3_TARG L3_MAIN Physical Address	MMU1_TARG L3_MAIN Physical Address	L4_PER2_P1_TARG L3_MAIN Physical Address	L4_PER2_P2_TARG L3_MAIN Physical Address	L4_PER2_P3_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 2000	0x4400 2100	0x4400 2200	0x4400 2300	0x4400 2400	0x4400 2500
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 2004	0x4400 2104	0x4400 2204	0x4400 2304	0x4400 2404	0x4400 2504
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 2008	0x4400 2108	0x4400 2208	0x4400 2308	0x4400 2408	0x4400 2508
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 2010	0x4400 2110	0x4400 2210	0x4400 2310	0x4400 2410	0x4400 2510

Table 14-104. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	TPCC_TARG L3_MAIN Physical Address	L4_PER1_P3_TARG L3_MAIN Physical Address	MMU1_TARG L3_MAIN Physical Address	L4_PER2_P1_TARG L3_MAIN Physical Address	L4_PER2_P2_TARG L3_MAIN Physical Address	L4_PER2_P3_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 2040	0x4400 2140	0x4400 2240	0x4400 2340	0x4400 2440	0x4400 2540
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 2044	0x4400 2144	0x4400 2244	0x4400 2344	0x4400 2444	0x4400 2544
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 2048	0x4400 2148	0x4400 2248	0x4400 2348	0x4400 2448	0x4400 2548
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 204C	0x4400 214C	0x4400 224C	0x4400 234C	0x4400 244C	0x4400 254C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 2050	0x4400 2150	0x4400 2250	0x4400 2350	0x4400 2450	0x4400 2550
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 2054	0x4400 2154	0x4400 2254	0x4400 2354	0x4400 2454	0x4400 2554
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 2058	0x4400 2158	0x4400 2258	0x4400 2358	0x4400 2458	0x4400 2558
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 205C	0x4400 215C	0x4400 225C	0x4400 235C	0x4400 245C	0x4400 255C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 2060	0x4400 2160	0x4400 2260	0x4400 2360	0x4400 2460	0x4400 2560
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 2064	0x4400 2164	0x4400 2264	0x4400 2364	0x4400 2464	0x4400 2564
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 2068	0x4400 2168	0x4400 2268	0x4400 2368	0x4400 2468	0x4400 2568
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 206C	0x4400 216C	0x4400 226C	0x4400 236C	0x4400 246C	0x4400 256C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 2080	0x4400 2180	0x4400 2280	0x4400 2380	0x4400 2480	0x4400 2580

Table 14-105. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER3_P1_TARG L3_MAIN Physical Address	L4_PER3_P2_TARG L3_MAIN Physical Address	MMU2_TARG L3_MAIN Physical Address	DSS_TARG L3_MAIN Physical Address	TPTC2_TARGL3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 2600	0x4400 2700	0x4400 2800	0x4400 2900	0x4400 2B00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 2604	0x4400 2704	0x4400 2804	0x4400 2904	0x4400 2B04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 2608	0x4400 2708	0x4400 2808	0x4400 2908	0x4400 2B08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 2610	0x4400 2710	0x4400 2810	0x4400 2910	0x4400 2B10

Table 14-105. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER3_P1_TARG L3_MAIN Physical Address	L4_PER3_P2_TARG L3_MAIN Physical Address	MMU2_TARG L3_MAIN Physical Address	DSS_TARG L3_MAIN Physical Address	TPTC2_TARGL3_MAIN Physical Address
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 2640	0x4400 2740	0x4400 2840	0x4400 2940	0x4400 2B40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 2644	0x4400 2744	0x4400 2844	0x4400 2944	0x4400 2B44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 2648	0x4400 2748	0x4400 2848	0x4400 2948	0x4400 2B48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 264C	0x4400 274C	0x4400 284C	0x4400 294C	0x4400 2B4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 2650	0x4400 2750	0x4400 2850	0x4400 2950	0x4400 2B50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 2654	0x4400 2754	0x4400 2854	0x4400 2954	0x4400 2B54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 2658	0x4400 2758	0x4400 2858	0x4400 2958	0x4400 2B58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 265C	0x4400 275C	0x4400 285C	0x4400 295C	0x4400 2B5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 2660	0x4400 2760	0x4400 2860	0x4400 2960	0x4400 2B60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 2664	0x4400 2764	0x4400 2864	0x4400 2964	0x4400 2B64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 2668	0x4400 2768	0x4400 2868	0x4400 2968	0x4400 2B68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 266C	0x4400 276C	0x4400 286C	0x4400 296C	0x4400 2B6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 2680	0x4400 2780	0x4400 2880	0x4400 2980	0x4400 2B80

Table 14-106. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	TPTC1_TARG L3_MAIN Physical Address	MCASP1_TARG L3_MAIN Physical Address	MCASP2_TARG L3_MAIN Physical Address	MCASP3_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 2E00	0x4400 2F00	0x4400 3000	0x4400 3100
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 2E04	0x4400 2F04	0x4400 3004	0x4400 3104
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 2E08	0x4400 2F08	0x4400 3008	0x4400 3108
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 2E10	0x4400 2F10	0x4400 3010	0x4400 3110
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 2E40	0x4400 2F40	0x4400 3040	0x4400 3140
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 2E44	0x4400 2F44	0x4400 3044	0x4400 3144
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 2E48	0x4400 2F48	0x4400 3048	0x4400 3148
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 2E4C	0x4400 2F4C	0x4400 304C	0x4400 314C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 2E50	0x4400 2F50	0x4400 3050	0x4400 3150

Table 14-106. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	TPTC1_TARG L3_MAIN Physical Address	MCASP1_TARG L3_MAIN Physical Address	MCASP2_TARG L3_MAIN Physical Address	MCASP3_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 2E54	0x4400 2F54	0x4400 3054	0x4400 3154
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 2E58	0x4400 2F58	0x4400 3058	0x4400 3158
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 2E5C	0x4400 2F5C	0x4400 305C	0x4400 315C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 2E60	0x4400 2F60	0x4400 3060	0x4400 3160
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 2E64	0x4400 2F64	0x4400 3064	0x4400 3164
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 2E68	0x4400 2F68	0x4400 3068	0x4400 3168
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 2E6C	0x4400 2F6C	0x4400 306C	0x4400 316C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 2E80	0x4400 2F80	0x4400 3080	0x4400 3180

Table 14-107. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIE1_TARG L3_MAIN Physical Address	PCIE2_TARG L3_MAIN Physical Address	QSPI_TARG L3_MAIN Physical Address	L3_INSTR_TARG L3_MAIN Physical Address	DEBUGSS_CT_TBR_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 3700	0x4400 3800	0x4400 3900	0x4500 0100	0x4500 0300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 3704	0x4400 3804	0x4400 3904	0x4500 0104	0x4500 0304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 3708	0x4400 3808	0x4400 3908	0x4500 0108	0x4500 0308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 3710	0x4400 3810	0x4400 3910	0x4500 0110	0x4500 0310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 3740	0x4400 3840	0x4400 3940	0x4500 0140	0x4500 0340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 3744	0x4400 3844	0x4400 3944	0x4500 0144	0x4500 0344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 3748	0x4400 3848	0x4400 3948	0x4500 0148	0x4500 0348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 374C	0x4400 384C	0x4400 394C	0x4500 014C	0x4500 034C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 3750	0x4400 3850	0x4400 3950	0x4500 0150	0x4500 0350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 3754	0x4400 3854	0x4400 3954	0x4500 0154	0x4500 0354
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 3758	0x4400 3858	0x4400 3958	0x4500 0158	0x4500 0358
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 375C	0x4400 385C	0x4400 395C	0x4500 015C	0x4500 035C

Table 14-107. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIE1_TARG L3_MAIN Physical Address	PCIE2_TARG L3_MAIN Physical Address	QSPI_TARG L3_MAIN Physical Address	L3_INSTR_TARG L3_MAIN Physical Address	DEBUGSS_CT_TBR_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 3760	0x4400 3860	0x4400 3960	0x4500 0160	0x4500 0360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 3764	0x4400 3864	0x4400 3964	0x4500 0164	0x4500 0364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 3768	0x4400 3868	0x4400 3968	0x4500 0168	0x4500 0368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 376C	0x4400 386C	0x4400 396C	0x4500 016C	0x4500 036C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 3780	0x4400 3880	0x4400 3980	0x4500 0180	0x4500 0380

14.2.5.1.3.2 L3_MAIN TARG Register Description

Table 14-108. L3_TARG_STDHOSTHDR_COREREG

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31:22	21:16	15:1	0
RESERVED	STDHOSTHDR_COREREG_ CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:16	STDHOSTHDR_COREREG_ CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x13.	R	0x13
15:1	RESERVED		R	0x0000
0	STDHOSTHDR_COREREG_ VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 14-109. Register Call Summary for Register L3_TARG_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [11]

Table 14-110. L3_TARG_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31:24	23:0
STDHOSTHDR_VERSIONREG_ REVISIONID	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 14-111. Register Call Summary for Register L3_TARG_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [11]

Table 14-112. L3_TARG_STDHOSTHDR_MAINCTLREG

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	RW		

31:4	3	2	1	0
RESERVED	STDHOSTSTHDR_MAINCTLREG_CM	STDHOSTSTHDR_MAINCTLREG_FLT	RESERVED	STDHOSTSTHDR_MAINCTLREG_EN

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3	STDHOSTHDR_MAINCTLREG_CM	Reserved for internal testing. Must be set to 0. Type: Control. Reset value: 0x0.	R	0
2	STDHOSTHDR_MAINCTLREG_FLT	Asserted when a Fault condition is detected: if the unit includes Error Logging, Flt is asserted when the FltCnt register field indicates a Fault, and deasserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X.	R	0
1	RESERVED	Reserved	R	0
0	STDHOSTHDR_MAINCTLREG_EN	Sets the global core enable. Note: A disabled master does not generate any NTTP requests, and a disabled slave replies with an error packet to any request it receives. Type: Control. Reset value: 0x1.	RW	1

Table 14-113. Register Call Summary for Register L3_TARG_STDHOSTHDR_MAINCTLREG

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [11]

Table 14-114. L3_TARG_STDHOSTHDR_NTTPADDR_0

Address Offset	See Table 14-97.

Table 14-114. L3_TARG_STDHOSTHDR_NTTPADDR_0 (continued)

Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31–7	6–0
RESERVED	STDHOSTHDR_NTTPADDR_0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	STDHOSTHDR_NTTPADDR_0	Shows the Rx port address. Type: Control. Reset value: 0x15.	R	See (1)

(1) GPMC- 0xC; DMM_P1 – 0x2; DSP1_SDMA - 0x4; L4_CFG - 0x14; DSP2_SDMA - 0x5; McASP1 - 0x1F; EVE1 - 0x7; EVE2 - 0x8; L4_PER3_P3 – 0x1; OCMC_RAM1 - 0x24; IPU1 - 0x10; IPU2 – 0x11; BB2D – 0xB; GPU- 0xD; DMM_P2 - 0x3; IVA_CONFIG - 0x12; OCMC_RAM2 - 0x25; IVA_SL2IF - 0x13; OCMC_RAM3 - 0x26; L4_PER1_P1 - 0x16; L4_WKUP - 0x1E; L4_PER1_P2 - 0x17; TPCC - 0x30; L4_PER1_P3 - 0x18; MMU1 - 0x22; L4_PER2_P1 - 0x18; L4_PER2_P2 - 0x42; L4_PER2_P3 - 0x1A; L4_PER3_P1 - 0x1B; L4_PER3_P2 - 0x1C; MMU2 - 0x23; DSS - 0x6; TPTC2 - 0x32; TPTC1 - 0x31; PCIE1 - 0x28; PCIE2 – 0x29; QSPI - 0x39; L3_INSTR- 0x19; DEBUGSS - 0x41;

Table 14-115. Register Call Summary for Register L3_TARG_STDHOSTHDR_NTTPADDR_0

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [11]

Table 14-116. L3_TARG_STDERRLOG_SVRTSTDLVL

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	RW		

31–2	1–0
RESERVED	STDERRLOG_SVRTSTDLVL_0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_SVRTSTDLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

Table 14-117. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTSTDLVL

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- L3_MAIN Interconnect Error Analysis Mode: [1]
- L3_MAIN TARG Register Summary and Description: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [13]

Table 14-118. L3_TARG_STDERRLOG_SVRTCUSTOMLVL

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107

Table 14-118. L3_TARG_STDERRLOG_SVRTCUSTOMLVL (continued)

Description	
Type	RW

31–2	1–0
RESERVED	STDERRLOG_SVRTCUSTOMLVL_0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_SVRTCUSTOMLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

Table 14-119. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTCUSTOMLVL

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- L3_MAIN TARG Register Summary and Description: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [12]

Table 14-120. L3_TARG_STDERRLOG_MAIN

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	RW		

31	30–20	19	18	17–2	1	0
STDERRLOG_MAIN_CLRLOG	RESERVED	STDERRLOG_MAIN_FLTCNT	STDERRLOG_MAIN_ERRCNT	RESERVED	STDERRLOG_MAIN_ERRTYPE	STDERRLOG_MAIN_ERRLOGVLD

Bits	Field Name	Description	Type	Reset
31	STDERRLOG_MAIN_CLRLOG	Clears "Error Logging Valid" bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0.	RW	0
30:20	RESERVED	Reserved	R	0x000
19	STDERRLOG_MAIN_FLTCNT	Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0

Bits	Field Name	Description	Type	Reset
18	STDERRLOG_MAIN_ERRCNT	Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
17:2	RESERVED	Reserved	R	0x0000
1	STDERRLOG_MAIN_ERRTYPE	Indicates logging type. Type: Status. Reset value: X. Read 0x0: Logged Error format is standard (header and necker recorded). Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information.	R	0
0	STDERRLOG_MAIN_ERRLOGVLD	Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X.	R	0

Table 14-121. Register Call Summary for Register L3_TARG_STDERRLOG_MAIN

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN Interconnect Error Analysis Mode: [2] [3] [4] [5] [6] [7]
- L3_MAIN TARG Register Summary and Description: [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [19]

Table 14-122. L3_TARG_STDERRLOG_HDR

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				STDERRLOG_HDR_LEN1										RESERVED		STDERRLOG_HDR_STOPOFSWRPSZ				STDERRLOG_HDR_ERR	RESERVED			STDERRLOG_HDR_PRESSURE		RESERVED		STDERRLOG_HDR_OPCODE			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	R	0x00
27:18	STDERRLOG_HDR_LEN1	This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X.	R	0x00
17:16	RESERVED	Reserved	R	0x0
15:12	STDERRLOG_HDR_STOPOFSWRPSZ	StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X.	R	0x0
11	STDERRLOG_HDR_ERR	Err bit of the logged packet. Type: Status. Reset value: X.	R	0
10:8	RESERVED	Reserved	R	0x0
7:6	STDERRLOG_HDR_PRESSURE	Pressure field of the logged packet. Type: Status. Reset value: X.	R	0
5:4	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
3:0	STDERRLOG_HDR_OPCODE	Opcode of the logged packet. Type: Status. Reset value: X. 0x0: Store without acknowledge, incrementing burst, non-atomic request 0x1: Store without acknowledge, wrapping burst, non-atomic request 0x2: Store with acknowledge, incrementing burst, non-atomic request 0x3: Store with acknowledge, wrapping burst, non-atomic request 0x4: Load, incrementing burst, non-atomic request 0x5: Load, wrapping burst, non-atomic request 0x6: Control packet 0x7: Flush 0x8: Store without acknowledge, incrementing burst, atomic request 0x9: Store without acknowledge, wrapping burst, atomic request 0xA: Store with acknowledge, incrementing burst, atomic request 0xB: Store with acknowledge, wrapping burst, atomic request 0xC: Load, incrementing burst, atomic request 0xD: Load, wrapping burst, atomic request 0xE: Reserved 0xF: Reserved	R	0x0

Table 14-123. Register Call Summary for Register L3_TARG_STDERRLOG_HDR

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [13]

Table 14-124. L3_TARG_STDERRLOG_MSTADDR

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_MSTADDR	Master Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 14-125. Register Call Summary for Register L3_TARG_STDERRLOG_MSTADDR

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [13]

Table 14-126. L3_TARG_STDERRLOG_SLVADDR

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31–7	6–0
RESERVED	STDERRLOG_SLVADDR

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	STDERRLOG_SLVADDR	Slave Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 14-127. Register Call Summary for Register L3_TARG_STDERRLOG_SLVADDR

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [13]

Table 14-128. L3_TARG_STDERRLOG_INFO

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31–8	7–0
RESERVED	STDERRLOG_INFO

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_INFO	Info field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 14-129. Register Call Summary for Register L3_TARG_STDERRLOG_INFO

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [13]

Table 14-130. L3_TARG_STDERRLOG_SLVOFSLSB

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_SLVOFSLSB																															

Bits	Field Name	Description	Type	Reset
31:0	STDERRLOG_SLVOFSLSB	LSB of the "slave offset" field, concatenated with "start offset" field of the logged packet. Type: Status. Reset value: X.	R	0x0000 0000

Table 14-131. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSLSB

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [11]

Table 14-132. L3_TARG_STDERRLOG_SLVOFSMSB

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STDERRLOG_SLVOFSMSB

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STDERRLOG_SLVOFSMSB	MSB of the "slave offset" field of the logged packet (according to NTTP packet format, this register field may exceed the actual "slave offset" size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X.	R	0

Table 14-133. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSMSB

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [11]

Table 14-134. L3_TARG_STDERRLOG_CUSTOMINFO_INFO

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_CUSTOMINFO_INFO							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000

Bits	Field Name	Description	Type	Reset
7:0	STDERRLOG_CUSTOMINFO_INFO	Info field of the response packet. Type: Status. Reset value: X.	R	0x00

Table 14-135. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_INFO

L3_MAIN Interconnect

- L3_MAIN Interconnect Error Analysis Mode: [0]
- L3_MAIN TARG Register Summary and Description: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [12]

Table 14-136. L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31–8	7–0
RESERVED	STDERRLOG_CUSTOMINFO_MSTADDR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_CUSTOMINFO_MSTADDR	MstAddr field of the response packet. Type: Status. Reset value: X.	R	0x00

Table 14-137. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR

L3_MAIN Interconnect

- L3_MAIN Interconnect Error Analysis Mode: [0]
- L3_MAIN TARG Register Summary and Description: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [12]

Table 14-138. L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	R		

31–2	1–0
RESERVED	STDERRLOG_CUSTOMINFO_OPCODE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_CUSTOMINFO_OPCODE	Opcode of the response packet. Type: Status. Reset value: X. 0x0: Logged request is *Store without acknowledge*. 0x1: Logged request is *Store with acknowledge*. 0x2: Logged request is *Load*. 0x3: Reserved	R	0x0

Table 14-139. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE

L3_MAIN Interconnect

- L3_MAIN Interconnect Error Analysis Mode: [0]
- L3_MAIN TARG Register Summary and Description: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [12]

Table 14-140. L3_TARG_ADDRSPACESIZELOG

Address Offset	See Table 14-97.		
Physical Address	See Table 14-97 to Table 14-107	**Instance**	See Table 14-97 to Table 14-107
Description			
Type	RW		

31–5	4–0
RESERVED	ADDRSPACESIZELOG

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:0	ADDRSPACESIZELOG	The address space size is equal to 2**AddrSpaceSizeLog * 4K in bytes. Type: Control. Reset value: 0x1F.	RW	0x1F

Table 14-141. Register Call Summary for Register L3_TARG_ADDRSPACESIZELOG

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [11]

14.2.5.1.4 L3_MAIN FLAGMUX Registers Summary and Description

Table 14-142. FLAGMUX Instance Summary

Module Name	Base Address	Size
CLK1_FLAGMUX_CLK1_1	0x4480 3500	4KiB
CLK1_FLAGMUX_CLK1_2	0x4480 3600	4KiB
CLK2_FLAGMUX_CLK2_1	0x4500 0200	4KiB

14.2.5.1.4.1 L3_MAIN FLAGMUX Registers Summary

Table 14-143. FLAGMUX Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_FLAGMUX_CLK1_1 L3_MAIN Physical Address	CLK1_FLAGMUX_CLK1_2 L3_MAIN Physical Address	CLK2_FLAGMUX_CLK2_1 L3_MAIN Physical Address
L3_FLAGMUX_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 3500	0x4480 3600	0x4500 0200
L3_FLAGMUX_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 3504	0x4480 3604	0x4500 0204
L3_FLAGMUX_MASK0	RW	32	0x0000 0008	0x4480 3508	0x4480 3608	0x4500 0208
L3_FLAGMUX_REGERR0	R	32	0x0000 000C	0x4480 350C	0x4480 360C	0x4500 020C
L3_FLAGMUX_MASK1	RW	32	0x0000 0010	0x4480 3510	0x4480 3610	0x4500 0210
L3_FLAGMUX_REGERR1	RW	32	0x0000 0014	0x4480 3514	0x4480 3614	0x4500 0214

14.2.5.1.4.2 L3_MAIN FLAGMUX Rebisters Description

Table 14-144. L3_FLAGMUX_STDHOSTHDR_COREREG

Address Offset	See Table 14-143		
Physical Address	0x4480 3500 0x4480 3600 0x4500 0200	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31–22	21–16	15–1	0
RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 14-145. Register Call Summary for Register L3_FLAGMUX_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX Registers Summary and Description: [0]

Table 14-146. L3_FLAGMUX_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-143		
Physical Address	0x4480 3504 0x4480 3604 0x4500 0204	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STDHOSTHDR_VERSIONREG_REVISIONID	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 14-147. Register Call Summary for Register L3_FLAGMUX_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- Flag Muxing: [0] [1]
- L3_MAIN FLAGMUX Registers Summary and Description: [2]

Table 14-148. L3_FLAGMUX_MASK0

Address Offset	See Table 14-143		
Physical Address	0x4480 3508 0x4480 3608 0x4500 0208	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK0																															

Bits	Field Name	Description	Type	Reset
31:0	MASK0(1)	Mask flag inputs 0 Type: Control.	RW	See (2)

(1) For CLK1_1 bit field is [31:0]; for CLK1_2 bit field is [22:0]; for CLK2 bit field is [3:0]
(2) Reset is 0xFFFFFFFF for CLK1_1; reset is 0x7FFFFF for CLK1_2; for CLK2 reset is 0xF

Table 14-149. Register Call Summary for Register L3_FLAGMUX_MASK0

L3_MAIN Interconnect

- Flag Mux Time-out: [0]
- Flag Mux Error Logging: [1]
- L3_MAIN Interconnect Error Analysis Mode: [2]
- L3_MAIN FLAGMUX Registers Summary and Description: [3]

Table 14-150. L3_FLAGMUX_REGERR0

Address Offset	See Table 14-143		
Physical Address	0x4480 350C 0x4480 360C 0x4500 020C	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REGERR0																															

Bits	Field Name	Description	Type	Reset
31:0	REGERR0	Flag inputs 0 Type: Status. Reset value: X.	R	0x00000

Table 14-151. Register Call Summary for Register L3_FLAGMUX_REGERR0

L3_MAIN Interconnect

- Flag Muxing: [0]
- Flag Mux Time-out: [1]
- Flag Mux Error Logging: [2]
- L3_MAIN Interconnect Error Analysis Mode: [3] [4] [5]
- L3_MAIN FLAGMUX Registers Summary and Description: [6]

Table 14-152. L3_FLAGMUX_MASK1

Address Offset	See Table 14-143		
Physical Address	0x4480 3510 0x4480 3610 0x4500 0210	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK1																															

Bits	Field Name	Description	Type	Reset
31:0	MASK1(1)	Mask flag inputs 1 Type: Control. Reset value: 0x7FFFFFF.	RW	See (2)

(1) For CLK1_1 bit field is [31:0]; for CLK1_2 bit field is [22:0]; for CLK2 bit field is [3:0]
(2) Reset is 0xFFFFFFFF for CLK1_1; reset is 0x7FFFFF for CLK1_2; for CLK2 reset is 0xF

Table 14-153. Register Call Summary for Register L3_FLAGMUX_MASK1

L3_MAIN Interconnect

- Flag Mux Time-out: [0]
- Flag Mux Error Logging: [1]
- L3_MAIN Interconnect Error Analysis Mode: [2]
- L3_MAIN FLAGMUX Registers Summary and Description: [3]

Table 14-154. L3_FLAGMUX_REGERR1

Address Offset	See Table 14-143		
Physical Address	0x4480 3514 0x4480 3614 0x4500 0214	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REGERR1																															

Bits	Field Name	Description	Type	Reset
31:0	REGERR1	Flag inputs 1 Type: Status. Reset value: X.	R	0x00000

Table 14-155. Register Call Summary for Register L3_FLAGMUX_REGERR1

L3_MAIN Interconnect

- Flag Muxing: [0]
- Flag Mux Time-out: [1]
- Flag Mux Error Logging: [2]
- L3_MAIN Interconnect Error Analysis Mode: [3] [4] [5]
- L3_MAIN FLAGMUX Registers Summary and Description: [6]

14.2.5.1.5 L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description

Table 14-156. Instance Summary

Module Name	Base Address	Size
CLK1_FLAGMUX_CLK1MERGE	0x4400 0000	4KiB

14.2.5.1.5.1 L3_MAIN FLAGMUX CLK1MERGE Registers Summary

Table 14-157. FLAGMUX CLK1MERGE Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_FLAGMUX_CLK1MERGE L3_MAIN Physical Address
L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG	R	32	0x0080 0400	0x4480 0400
L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG	R	32	0x0080 0404	0x4480 0404
L3_FLAGMUX_CLK1MERGE_MASK0	RW	32	0x0080 0408	0x4480 0408
L3_FLAGMUX_CLK1MERGE_REGERR0	R	32	0x0080 040C	0x4480 040C
L3_FLAGMUX_CLK1MERGE_MASK1	RW	32	0x0080 0410	0x4480 0410
L3_FLAGMUX_CLK1MERGE_REGERR1	R	32	0x0080 0414	0x4480 0414

14.2.5.1.5.2 L3_MAIN FLAGMUX CLK1MERGE Registers Description

Table 14-158. L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG

Address Offset	See Table 14-157.		
Physical Address	0x4480 0400	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31:22	21:16	15:1	0
RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED	Reserved	R	0x0000

Bits	Field Name	Description	Type	Reset
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 14-159. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [0]

Table 14-160. L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-157.		
Physical Address	0x4480 0404	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 14-161. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [0]

Table 14-162. L3_FLAGMUX_CLK1MERGE_MASK0

Address Offset	See Table 14-157.		
Physical Address	0x4480 0408	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														MASK0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	MASK0	Mask flag inputs 0 Type: Control. Reset value: 0x3	RW	0x3

Table 14-163. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_MASK0

L3_MAIN Interconnect

- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

Table 14-164. L3_FLAGMUX_CLK1MERGE_REGERR0

Address Offset	See Table 14-157.		
Physical Address	0x4480 040C	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31–2	1–0
RESERVED	REGERR0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	REGERR0	Flag inputs 0 Type: Control. Reset value: X	RW	0x0

Table 14-165. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_REGERR0

L3_MAIN Interconnect

- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

Table 14-166. L3_FLAGMUX_CLK1MERGE_MASK1

Address Offset	See Table 14-157.		
Physical Address	0x4480 0410	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	RW		

31–2	1–0
RESERVED	MASK1

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	MASK1	Mask flag inputs 0 Type: Control. Reset value: 0x3	RW	0x3

Table 14-167. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_MASK1

L3_MAIN Interconnect

- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

Table 14-168. L3_FLAGMUX_CLK1MERGE_REGERR1

Address Offset	See Table 14-157.		
Physical Address	0x4480 0414	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31–2	1–0
RESERVED	REGERR1

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	REGERR1	Flag inputs 0 Type: Control. Reset value: X	RW	0x0

Table 14-169. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_REGERR1

L3_MAIN Interconnect

- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

14.2.5.1.6 L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description

Table 14-170. FLAGMUX TIMEOUT Instance Summary

Module Name	Base Address	Size
CLK1_FLAGMUX_CLK1	0x4400 0000	4KiB
CLK2_FLAGMUX_CLK2 (1)	0x4500 0000	4KiB

(1) clk2_flagmux_clk2 refers to CLK2_1 clock domain

14.2.5.1.6.1 L3_MAIN FLAGMUX TIMEOUT Registers Summary

Table 14-171. FLAGMUX Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_FLAGMUX_CLK1 L3_MAIN Physical Address	CLK2_FLAGMUX_CLK2 L3_MAIN Physical Address
L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG	R	32	0x0000 0400	N/A	0x4500 0400
L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG	R	32	0x0000 0404	N/A	0x4500 0404
L3_FLAGMUX_TIMEOUT_MASK0	RW	32	0x0000 0408	N/A	0x4500 0408
L3_FLAGMUX_TIMEOUT_REGERR0	R	32	0x0000 040C	N/A	0x4500 040C
L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG	R	32	0x0080 5700	0x4480 5700	N/A
L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG	R	32	0x0080 5704	0x4480 5704	N/A
L3_FLAGMUX_TIMEOUT1_MASK0	RW	32	0x0080 5708	0x4480 5708	N/A
L3_FLAGMUX_TIMEOUT1_REGERR0	R	32	0x0080 570C	0x4480 570C	N/A
L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG	R	32	0x0080 5800	0x4480 5800	N/A
L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG	R	32	0x0080 5804	0x4480 5804	N/A
L3_FLAGMUX_TIMEOUT2_MASK0	RW	32	0x0080 5808	0x4480 5808	N/A
L3_FLAGMUX_TIMEOUT2_REGERR0	R	32	0x0080 580C	0x4480 580C	N/A

14.2.5.1.6.2 L3_MAIN FLAGMUX TIMEOUT Registers Description

Table 14-172. L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG

Address Offset	See Table 14-171.		
Physical Address	0x4500 0400	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0bxx xxxx xxxx
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED		R	0bxxx xxxx xxxx xxxx
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 14-173. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [0]

Table 14-174. L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-171.		
Physical Address	0x4500 0404	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	R		

31–24	23–0
STDHOSTHDR_VERSIONREG_REVISIONID	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x-- ----

Table 14-175. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [0]

Table 14-176. L3_FLAGMUX_TIMEOUT_MASK0

Address Offset	See Table 14-171.		
Physical Address	0x4500 0408	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														MASK0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx
1:0	MASK0	mask flag inputs 0 Type: Control. Reset value: 0x0.	RW	0x3

Table 14-177. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_MASK0

L3_MAIN Interconnect

- Flag Mux Time-out: [0] [1]
- L3_MAIN Interconnect Error Analysis Mode: [2]
- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [3]

Table 14-178. L3_FLAGMUX_TIMEOUT_REGERR0

Address Offset	See Table 14-171.		
Physical Address	0x4500 040C	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														REGERR0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx
1:0	REGERR0	flag inputs 0 Type: Status. Reset value: X.	R	0bx xxxx

Table 14-179. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_REGERR0

L3_MAIN Interconnect

- Flag Mux Time-out: [0]
- L3_MAIN Interconnect Error Analysis Mode: [1]
- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [2]

Table 14-180. L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG

Address Offset	See Table 14-171.		
Physical Address	0x4480 5700	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										STDHOSTHDR_COREREG_CORECODE						RESERVED															STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0bxx xxxx xxxx
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED		R	0bxxx xxxx xxxx xxxx
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 14-181. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [0]

Table 14-182. L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-171.		
Physical Address	0x4480 5704	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x-- ----

Table 14-183. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [0]

Table 14-184. L3_FLAGMUX_TIMEOUT1_MASK0

Address Offset	See Table 14-171.		
Physical Address	0x4480 5708	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		MASK0																													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx
29:0	MASK0	mask flag inputs 0 Type: Control. Reset value: 0x0.	RW	0x3FFFFFFF

Table 14-185. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_MASK0

L3_MAIN Interconnect

- Flag Mux Time-out: [0]
- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [1]

Table 14-186. L3_FLAGMUX_TIMEOUT1_REGERR0

Address Offset	See Table 14-171.		
Physical Address	0x4480 570C	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							REGERR0																								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx
24:0	REGERR0	flag inputs 0 Type: Status. Reset value: X.	R	0bx xxxx

Table 14-187. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_REGERR0

L3_MAIN Interconnect

- Flag Mux Time-out: [0]
- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [1]

Table 14-188. L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG

Address Offset	See Table 14-171.		
Physical Address	0x4480 5800	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0bxx xxxx xxxx
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED		R	0bxxx xxxx xxxx xxxx
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 14-189. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [0]

Table 14-190. L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-171.		
Physical Address	0x4480 5804	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
STDHOSTHDR_VERSIONREG_REVISIONID	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x-- ----

Table 14-191. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect
- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [0]

Table 14-192. L3_FLAGMUX_TIMEOUT2_MASK0

Address Offset	See Table 14-171.		
Physical Address	0x4480 5808	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	RW		

31:21	20:0
RESERVED	MASK0

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx
20:0	MASK0	mask flag inputs 0 Type: Control. Reset value: 0x0.	RW	0x1FFFFFFF

Table 14-193. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_MASK0

L3_MAIN Interconnect
- Flag Mux Time-out: [0] [1]
- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [2]

Table 14-194. L3_FLAGMUX_TIMEOUT2_REGERR0

Address Offset	See Table 14-171.		
Physical Address	0x4480 580C	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31:21	20:0
RESERVED	REGERR0

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx
20:0	REGERR0	flag inputs 0 Type: Status. Reset value: X.	R	0bx xxxx

Table 14-195. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_REGERR0

L3_MAIN Interconnect
- Flag Mux Time-out: [0]
- L3_MAIN FLAGMUX TIMEOUT Registers Summary and Description: [1]

14.2.5.1.7 L3_MAIN BW Regulator Register Summary and Description

Table 14-196. BW_REGULATOR Instance Summary

Module Name	Base Address	Size
CLK1_2_MMU2_BW_REGULATOR	0x4480 3B00	4KiB
CLK1_2_EVE1_TC0_BW_REGULATOR	0x4480 4200	4KiB
CLK1_2_EVE2_TC0_BW_REGULATOR	0x4480 4300	4KiB
CLK1_2_EVE1_TC1_BW_REGULATOR	0x4480 4600	4KiB
CLK1_2_EVE2_TC1_BW_REGULATOR	0x4480 4700	4KiB

Table 14-196. BW_REGULATOR Instance Summary (continued)

Module Name	Base Address	Size
CLK1_2_DSP2_EDMA_BW_REGULATOR	0x4480 4A00	4KiB
CLK1_2_DSP1_EDMA_BW_REGULATOR	0x4480 4B00	4KiB
CLK1_2_DSP1_MDMA_BW_REGULATOR	0x4480 4C00	4KiB
CLK1_2_DSP2_MDMA_BW_REGULATOR	0x4480 4D00	4KiB
CLK1_2_BB2D_P1_BW_REGULATOR	0x4480 4E00	4KiB
CLK1_2_IVA_BW_REGULATOR	0x4480 5000	4KiB
CLK1_2_BB2D_P2_BW_REGULATOR	0x4480 5100	4KiB
CLK1_2_GPU_P1_BW_REGULATOR	0x4480 5200	4KiB
CLK1_2_GPU_P2_BW_REGULATOR	0x4480 5300	4KiB
CLK1_2_PCIESS2_BW_REGULATOR	0x4480 5400	4KiB
CLK1_2_PCIESS1_BW_REGULATOR	0x4480 5500	4KiB
CLK1_2_GMAC_SW_BW_REGULATOR	0x4480 5600	4KiB

14.2.5.1.7.1 L3_MAIN BW_REGULATOR Register Summary

Table 14-197. BW_REGULATOR Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_MMU2_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 3B00
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 3B04
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 3B08
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 3B0C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 3B10
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 3B14

Table 14-198. BW_REGULATOR Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_EVE1_TC0_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4200
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4204
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 4208
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 420C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 4210
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4214

Table 14-199. BW_REGULATOR Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_EVE2_TC0_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_EVE1_TC1_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_EVE2_TC1_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4300	0x4480 4600	0x4480 4700
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4304	0x4480 4604	0x4480 4704
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 4308	0x4480 4608	0x4480 4708
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 430C	0x4480 460C	0x4480 470C

Table 14-199. BW_REGULATOR Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_EVE2_TC0_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_EVE1_TC1_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_EVE2_TC1_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 4310	0x4480 4610	0x4480 4710
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4314	0x4480 4614	0x4480 4714

Table 14-200. BW_REGULATOR Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_DSP2_EDMA_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_DSP1_EDMA_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_DSP1_MDMA_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4A00	0x4480 4B00	0x4480 4C00
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4A04	0x4480 4B04	0x4480 4C04
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 4A08	0x4480 4B08	0x4480 4C08
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 4A0C	0x4480 4B0C	0x4480 4C0C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 4A10	0x4480 4B10	0x4480 4C10
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4A14	0x4480 4B14	0x4480 4C14

Table 14-201. BW_REGULATOR Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_DSP2_MDMA_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_BB2D_P1_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_IVA_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4D00	0x4480 4E00	0x4480 5000
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4D04	0x4480 4E04	0x4480 5004
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 4D08	0x4480 4E08	0x4480 5008
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 4D0C	0x4480 4E0C	0x4480 500C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 4D10	0x4480 4E10	0x4480 5010
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4D14	0x4480 4E14	0x4480 5014

Table 14-202. BW_REGULATOR Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_BB2D_P2_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_GPU_P1_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_GPU_P2_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 5100	0x4480 5200	0x4480 5300
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 5104	0x4480 5204	0x4480 5304
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 5108	0x4480 5208	0x4480 5308
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 510C	0x4480 520C	0x4480 530C

Table 14-202. BW_REGULATOR Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_BB2D_P2_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_GPU_P1_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_GPU_P2_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 5110	0x4480 5210	0x4480 5310
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 5114	0x4480 5214	0x4480 5314

Table 14-203. BW_REGULATOR Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_2_PCIESS2_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_PCIESS1_BW_REGULATOR L3_MAIN Physical Address	CLK1_2_GMAC_SW_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 5400	0x4480 5500	0x4480 5600
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 5404	0x4480 5504	0x4480 5604
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 5408	0x4480 5508	0x4480 5608
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 540C	0x4480 550C	0x4480 560C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 5410	0x4480 5510	0x4480 5610
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 5414	0x4480 5514	0x4480 5614

14.2.5.1.7.2 L3_MAIN BW_REGULATOR Register Description

Table 14-204. L3_BW_REGULATOR_STDHOSTHDR_COREREG

Address Offset	See Table 14-197 to Table 14-203

Table 14-204. L3_BW_REGULATOR_STDHOSTHDR_COREREG (continued)

Physical Address	Instance
0x4480 3B00	CLK1_2_MMU2_BW_REGULATOR
0x4480 4200	CLK1_2_EVE1_TC0_BW_REGULATOR
0x4480 4300	CLK1_2_EVE2_TC0_BW_REGULATOR
0x4480 4600	CLK1_2_EVE1_TC1_BW_REGULATOR
0x4480 4700	CLK1_2_EVE2_TC1_BW_REGULATOR
0x4480 4A00	CLK1_2_DSP2_EDMA_BW_REGULATOR
0x4480 4B00	CLK1_2_DSP1_EDMA_BW_REGULATOR
0x4480 4C00	CLK1_2_DSP1_MDMA_BW_REGULATOR
0x4480 4D00	CLK1_2_DSP2_MDMA_BW_REGULATOR
0x4480 4E00	CLK1_2_BB2D_P1_BW_REGULATOR
0x4480 5000	CLK1_2_IVA_BW_REGULATOR
0x4480 5100	CLK1_2_BB2D_P2_BW_REGULATOR
0x4480 5200	CLK1_2_GPU_P1_BW_REGULATOR
0x4480 5300	CLK1_2_GPU_P2_BW_REGULATOR
0x4480 5400	CLK1_2_PCIESS2_BW_REGULATOR
0x4480 5500	CLK1_2_PCIESS1_BW_REGULATOR
0x4480 5600	CLK1_2_GMAC_SW_BW_REGULATOR

Description

Type R

31:22	21:16	15:1	0
RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x31.	R	0x31

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 14-205. Register Call Summary for Register L3_BW_REGULATOR_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN BW Regulator Register Summary and Description: [0] [1] [2] [4] [5] [6] [7]

Table 14-206. L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-197 to Table 14-203		
Physical Address	0x4480 3B04 0x4480 4204 0x4480 4304 0x4480 4604 0x4480 4704 0x4480 4A04 0x4480 4B04 0x4480 4C04 0x4480 4D04 0x4480 4E04 0x4480 5004 0x4480 5104 0x4480 5204 0x4480 5304 0x4480 5404 0x4480 5504 0x4480 5604	**Instance**	CLK1_2_MMU2_BW_REGULATOR CLK1_2_EVE1_TC0_BW_REGULATOR CLK1_2_EVE2_TC0_BW_REGULATOR CLK1_2_EVE1_TC1_BW_REGULATOR CLK1_2_EVE2_TC1_BW_REGULATOR CLK1_2_DSP2_EDMA_BW_REGULATOR CLK1_2_DSP1_EDMA_BW_REGULATOR CLK1_2_DSP1_MDMA_BW_REGULATOR CLK1_2_DSP2_MDMA_BW_REGULATOR CLK1_2_BB2D_P1_BW_REGULATOR CLK1_2_IVA_BW_REGULATOR CLK1_2_BB2D_P2_BW_REGULATOR CLK1_2_GPU_P1_BW_REGULATOR CLK1_2_GPU_P2_BW_REGULATOR CLK1_2_PCIESS2_BW_REGULATOR CLK1_2_PCIESS1_BW_REGULATOR CLK1_2_GMAC_SW_BW_REGULATOR
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 14-207. Register Call Summary for Register L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN BW Regulator Register Summary and Description: [0] [1] [2] [4] [5] [6] [7]

Table 14-208. L3_BW_REGULATOR_BANDWIDTH

Address Offset	See Table 14-197 to Table 14-203		
Physical Address	0x4480 3B08 0x4480 4208 0x4480 4308 0x4480 4608 0x4480 4708 0x4480 4A08 0x4480 4B08 0x4480 4C08 0x4480 4D08 0x4480 4E08 0x4480 5008 0x4480 5108 0x4480 5208 0x4480 5308 0x4480 5408 0x4480 5508 0x4480 5608	**Instance**	CLK1_2_MMU2_BW_REGULATOR CLK1_2_EVE1_TC0_BW_REGULATOR CLK1_2_EVE2_TC0_BW_REGULATOR CLK1_2_EVE1_TC1_BW_REGULATOR CLK1_2_EVE2_TC1_BW_REGULATOR CLK1_2_DSP2_EDMA_BW_REGULATOR CLK1_2_DSP1_EDMA_BW_REGULATOR CLK1_2_DSP1_MDMA_BW_REGULATOR CLK1_2_DSP2_MDMA_BW_REGULATOR CLK1_2_BB2D_P1_BW_REGULATOR CLK1_2_IVA_BW_REGULATOR CLK1_2_BB2D_P2_BW_REGULATOR CLK1_2_GPU_P1_BW_REGULATOR CLK1_2_GPU_P2_BW_REGULATOR CLK1_2_PCIESS2_BW_REGULATOR CLK1_2_PCIESS1_BW_REGULATOR CLK1_2_GMAC_SW_BW_REGULATOR
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BANDWIDTH															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	BANDWIDTH	Bandwidth, in bytes per second. Type: Control. Reset value: 0x0.	RW	0x0000

Table 14-209. Register Call Summary for Register L3_BW_REGULATOR_BANDWIDTH

L3_MAIN Interconnect

- Bandwidth Regulators: [0] [1]
- L3_MAIN BW Regulator Register Summary and Description: [2] [3] [4] [6] [7] [8] [9]

Table 14-210. L3_BW_REGULATOR_WATERMARK

Address Offset	See Table 14-197 to Table 14-203

Table 14-210. L3_BW_REGULATOR_WATERMARK (continued)

Physical Address	0x4480 3B0C 0x4480 420C 0x4480 430C 0x4480 460C 0x4480 470C 0x4480 4A0C 0x4480 4B0C 0x4480 4C0C 0x4480 4D0C 0x4480 4E0C 0x4480 500C 0x4480 510C 0x4480 520C 0x4480 530C 0x4480 540C 0x4480 550C 0x4480 560C	**Instance**	CLK1_2_MMU2_BW_REGULATOR CLK1_2_EVE1_TC0_BW_REGULATOR CLK1_2_EVE2_TC0_BW_REGULATOR CLK1_2_EVE1_TC1_BW_REGULATOR CLK1_2_EVE2_TC1_BW_REGULATOR CLK1_2_DSP2_EDMA_BW_REGULATOR CLK1_2_DSP1_EDMA_BW_REGULATOR CLK1_2_DSP1_MDMA_BW_REGULATOR CLK1_2_DSP2_MDMA_BW_REGULATOR CLK1_2_BB2D_P1_BW_REGULATOR CLK1_2_IVA_BW_REGULATOR CLK1_2_BB2D_P2_BW_REGULATOR CLK1_2_GPU_P1_BW_REGULATOR CLK1_2_GPU_P2_BW_REGULATOR CLK1_2_PCIESS2_BW_REGULATOR CLK1_2_PCIESS1_BW_REGULATOR CLK1_2_GMAC_SW_BW_REGULATOR
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				WATERMARK											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:0	WATERMARK	Peak permissible bandwidth, in bytes. Type: Control. Reset value: 0x1.	RW	0x001

Table 14-211. Register Call Summary for Register L3_BW_REGULATOR_WATERMARK

L3_MAIN Interconnect

- Bandwidth Regulators: [0] [1] [2]
- L3_MAIN BW Regulator Register Summary and Description: [3] [4] [5] [7] [8] [9] [10]

Table 14-212. L3_BW_REGULATOR_PRESS

Address Offset	See Table 14-197 to Table 14-203

Table 14-212. L3_BW_REGULATOR_PRESS (continued)

Physical Address	0x4480 3B10 0x4480 4210 0x4480 4310 0x4480 4610 0x4480 4710 0x4480 4A10 0x4480 4B10 0x4480 4C10 0x4480 4D10 0x4480 4E10 0x4480 5010 0x4480 5110 0x4480 5210 0x4480 5310 0x4480 5410 0x4480 5510 0x4480 5610	**Instance**	CLK1_2_MMU2_BW_REGULATOR CLK1_2_EVE1_TC0_BW_REGULATOR CLK1_2_EVE2_TC0_BW_REGULATOR CLK1_2_EVE1_TC1_BW_REGULATOR CLK1_2_EVE2_TC1_BW_REGULATOR CLK1_2_DSP2_EDMA_BW_REGULATOR CLK1_2_DSP1_EDMA_BW_REGULATOR CLK1_2_DSP1_MDMA_BW_REGULATOR CLK1_2_DSP2_MDMA_BW_REGULATOR CLK1_2_BB2D_P1_BW_REGULATOR CLK1_2_IVA_BW_REGULATOR CLK1_2_BB2D_P2_BW_REGULATOR CLK1_2_GPU_P1_BW_REGULATOR CLK1_2_GPU_P2_BW_REGULATOR CLK1_2_PCIESS2_BW_REGULATOR CLK1_2_PCIESS1_BW_REGULATOR CLK1_2_GMAC_SW_BW_REGULATOR
Description			
Type	R		

31–4	3–2	1–0
RESERVED	PRESS_LOW	PRESS_HIGH

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000 0000
3:2	PRESS_LOW	Pressure value inserted if the measured bandwidth is over the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x0.	R	0
1:0	PRESS_HIGH	Pressure value inserted if the measured bandwidth is under the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x1.	R	0x3

Table 14-213. Register Call Summary for Register L3_BW_REGULATOR_PRESS

L3_MAIN Interconnect

- Bandwidth Regulators: [0]
- L3_MAIN BW Regulator Register Summary and Description: [1] [2] [3] [5] [6] [7] [8]

Table 14-214. L3_BW_REGULATOR_CLEARHISTORY

Address Offset	See Table 14-197 to Table 14-203

Table 14-214. L3_BW_REGULATOR_CLEARHISTORY (continued)

Physical Address	0x4480 3B14 0x4480 4214 0x4480 4314 0x4480 4614 0x4480 4714 0x4480 4A14 0x4480 4B14 0x4480 4C14 0x4480 4D14 0x4480 4E14 0x4480 5014 0x4480 5114 0x4480 5214 0x4480 5314 0x4480 5414 0x4480 5514 0x4480 5614	**Instance**	CLK1_2_MMU2_BW_REGULATOR CLK1_2_EVE1_TC0_BW_REGULATOR CLK1_2_EVE2_TC0_BW_REGULATOR CLK1_2_EVE1_TC1_BW_REGULATOR CLK1_2_EVE2_TC1_BW_REGULATOR CLK1_2_DSP2_EDMA_BW_REGULATOR CLK1_2_DSP1_EDMA_BW_REGULATOR CLK1_2_DSP1_MDMA_BW_REGULATOR CLK1_2_DSP2_MDMA_BW_REGULATOR CLK1_2_BB2D_P1_BW_REGULATOR CLK1_2_IVA_BW_REGULATOR CLK1_2_BB2D_P2_BW_REGULATOR CLK1_2_GPU_P1_BW_REGULATOR CLK1_2_GPU_P2_BW_REGULATOR CLK1_2_PCIESS2_BW_REGULATOR CLK1_2_PCIESS1_BW_REGULATOR CLK1_2_GMAC_SW_BW_REGULATOR
Description			
Type	RW		

31–1	0
RESERVED	CLEARHISTORY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	CLEARHISTORY	Write a 1 clear the traffic counter Type: Give_AutoCleared. Reset value: 0x0.	RW	0

Table 14-215. Register Call Summary for Register L3_BW_REGULATOR_CLEARHISTORY

L3_MAIN Interconnect

- Bandwidth Regulators: [0]
- L3_MAIN BW Regulator Register Summary and Description: [1] [2] [3] [5] [6] [7] [8]

14.2.5.1.8 L3_MAIN Bandwidth Limiter Register Summary and Description

Table 14-216. BW_LIMITER Instance Summary

Module Name	Base Address	Size
CLK1_2_BB2D_P1_BW_LIMITER	0x4480 5900	256B

Table 14-216. BW_LIMITER Instance Summary (continued)

Module Name	Base Address	Size
CLK1_2_BB2D_P2_BW_LIMITER	0x4480 5A00	256B
CLK1_2_GPU_P1_BW_LIMITER	0x4480 5B00	256B
CLK1_2_GPU_P2_BW_LIMITER	0x4480 5C00	256B
CLK1_2_TPTC1_RD_BW_LIMITER	0x4480 3C00	4KiB
CLK1_2_TPTC2_RD_BW_LIMITER	0x4480 3D00	4KiB
CLK1_2_TPTC1_WR_BW_LIMITER	0x4480 3E00	4KiB
CLK1_2_TPTC2_WR_BW_LIMITER	0x4480 3F00	4KiB
CLK1_2_MMU1_BW_LIMITER	0x4480 3A00	4KiB
CLK1_2_VPE_P2_BW_LIMITER	0x4480 4000	4KiB
CLK1_2_VPE_P1_BW_LIMITER	0x4480 4100	4KiB

14.2.5.1.8.1 L3_MAIN BW Limiter Register Summary

Table 14-217. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_2_BB2D_P1_BW_LIMITER L3_MAIN Physical Address	CLK1_2_BB2D_P2_BW_LIMITER L3_MAIN Physical Address	CLK1_2_TPTC1_RD_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 5900	0x4480 5A00	0x4480 3C00
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 5904	0x4480 5A04	0x4480 3C04
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 5908	0x4480 5A08	0x4480 3C08
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 590C	0x4480 5A0C	0x4480 3C0C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 5910	0x4480 5A10	0x4480 3C10
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 5914	0x4480 5A14	0x4480 3C14

Table 14-218. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_2_GPU_P1_BW_LIMITER L3_MAIN Physical Address	CLK1_2_GPU_P2_BW_LIMITER L3_MAIN Physical Address	CLK1_2_TPTC2_RD_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 5B00	0x4480 5C00	0x4480 3D00
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 5B04	0x4480 5C04	0x4480 3D04
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 5B08	0x4480 5C08	0x4480 3D08
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 5B0C	0x4480 5C0C	0x4480 3D0C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 5B10	0x4480 5C10	0x4480 3D10
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 5B14	0x4480 5C14	0x4480 3D14

Table 14-219. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_2_TPTC1_WR_BW_LIMITER L3_MAIN Physical Address	CLK1_2_TPTC2_WR_BW_LIMITER L3_MAIN Physical Address	CLK1_2_MMU1_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 3E00	0x4480 3F00	0x4480 3A00
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 3E04	0x4480 3F04	0x4480 3A04
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 3E08	0x4480 3F08	0x4480 3A08
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 3E0C	0x4480 3F0C	0x4480 3A0C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 3E10	0x4480 3F10	0x4480 3A10
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 3E14	0x4480 3F14	0x4480 3A14

Table 14-220. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_2_VPE_P2_BW_LIMITER L3_MAIN Physical Address	CLK1_2_VPE_P1_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4000	0x4480 4100
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4004	0x4480 4104
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 4008	0x4480 4108
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 400C	0x4480 410C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 4010	0x4480 4110
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4014	0x4480 4114

14.2.5.1.8.2 L3_MAIN BW Limiter Register Description

Table 14-221. L3_BW_LIMITER_STDHOSTHDR_COREREG

Address Offset	See Table 14-217		
Physical Address	0x4480 5900 0x4480 5A00 0x4480 3C00 0x4480 5B00 0x4480 5C00 0x4480 3D00 0x4480 3E00 0x4480 3F00 0x4480 3A00 0x4480 4000 0x4480 4100	**Instance**	CLK1_2_BB2D_P1_BW_LIMITER CLK1_2_BB2D_P2_BW_LIMITER CLK1_2_TPTC1_RD_BW_LIMITER CLK1_2_GPU_P1_BW_LIMITER CLK1_2_GPU_P2_BW_LIMITER CLK1_2_TPTC2_RD_BW_LIMITER CLK1_2_TPTC1_WR_BW_LIMITER CLK1_2_TPTC2_WR_BW_LIMITER CLK1_2_MMU1_BW_LIMITER CLK1_2_VPE_P2_BW_LIMITER CLK1_2_VPE_P1_BW_LIMITER
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x2C.	R	0x2C
15:1	RESERVED		R	0x0
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 14-222. Register Call Summary for Register L3_BW_LIMITER_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN Bandwidth Limiter Register Summary and Description: [0] [1] [2] [3]

Table 14-223. L3_BW_LIMITER_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-217		
Physical Address	0x4480 5904 0x4480 5A04 0x4480 3C04 0x4480 5B04 0x4480 5C04 0x4480 3D04 0x4480 3E04 0x4480 3F04 0x4480 3A04 0x4480 4004 0x4480 4104	**Instance**	CLK1_2_BB2D_P1_BW_LIMITER CLK1_2_BB2D_P2_BW_LIMITER CLK1_2_TPTC1_RD_BW_LIMITER CLK1_2_GPU_P1_BW_LIMITER CLK1_2_GPU_P2_BW_LIMITER CLK1_2_TPTC2_RD_BW_LIMITER CLK1_2_TPTC1_WR_BW_LIMITER CLK1_2_TPTC2_WR_BW_LIMITER CLK1_2_MMU1_BW_LIMITER CLK1_2_VPE_P2_BW_LIMITER CLK1_2_VPE_P1_BW_LIMITER
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STDHOSTHDR_VERSIONREG_REVISIONID	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x0

Table 14-224. Register Call Summary for Register L3_BW_LIMITER_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Bandwidth Limiter Register Summary and Description: [0] [1] [2] [3]

Table 14-225. L3_BW_LIMITER_BANDWIDTH_FRACTIONAL

Address Offset	See Table 14-217		
Physical Address	0x4480 5908 0x4480 5A08 0x4480 3C08 0x4480 5B08 0x4480 5C08 0x4480 3D08 0x4480 3E08 0x4480 3F08 0x4480 3A08 0x4480 4008 0x4480 4108	**Instance**	CLK1_2_BB2D_P1_BW_LIMITER CLK1_2_BB2D_P2_BW_LIMITER CLK1_2_TPTC1_RD_BW_LIMITER CLK1_2_GPU_P1_BW_LIMITER CLK1_2_GPU_P2_BW_LIMITER CLK1_2_TPTC2_RD_BW_LIMITER CLK1_2_TPTC1_WR_BW_LIMITER CLK1_2_TPTC2_WR_BW_LIMITER CLK1_2_MMU1_BW_LIMITER CLK1_2_VPE_P2_BW_LIMITER CLK1_2_VPE_P1_BW_LIMITER
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											BANDWIDTH_FRACTIONAL				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	BANDWIDTH_FRACTIONAL	Fractional part of bandwitdh in terms of bytes per second Type: Control. Reset value: 0x0.	RW	0x0

Table 14-226. Register Call Summary for Register L3_BW_LIMITER_BANDWIDTH_FRACTIONAL

L3_MAIN Interconnect

- Bandwidth Limiters: [0] [1] [2]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [3] [4] [5] [6]

Table 14-227. L3_BW_LIMITER_BANDWIDTH_INTEGER

Address Offset	See Table 14-217

Table 14-227. L3_BW_LIMITER_BANDWIDTH_INTEGER (continued)

Physical Address	0x4480 590C 0x4480 5A0C 0x4480 3C0C 0x4480 5B0C 0x4480 5C0C 0x4480 3D0C 0x4480 3E0C 0x4480 3F0C 0x4480 3A0C 0x4480 400C 0x4480 410C	**Instance**	CLK1_2_BB2D_P1_BW_LIMITER CLK1_2_BB2D_P2_BW_LIMITER CLK1_2_TPTC1_RD_BW_LIMITER CLK1_2_GPU_P1_BW_LIMITER CLK1_2_GPU_P2_BW_LIMITER CLK1_2_TPTC2_RD_BW_LIMITER CLK1_2_TPTC1_WR_BW_LIMITER CLK1_2_TPTC2_WR_BW_LIMITER CLK1_2_MMU1_BW_LIMITER CLK1_2_VPE_P2_BW_LIMITER CLK1_2_VPE_P1_BW_LIMITER
Description			
Type	RW		

31–4	3–0
RESERVED	BANDWIDTH_INTEGER

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	BANDWIDTH_INTEGER	Integer part of bandwitdh in terms of bytes per second Type: Control. Reset value: 0x0.	RW	0x0

Table 14-228. Register Call Summary for Register L3_BW_LIMITER_BANDWIDTH_INTEGER

L3_MAIN Interconnect

- Bandwidth Limiters: [0] [1]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [2] [3] [4] [5]

Table 14-229. L3_BW_LIMITER_WATERMARK_0

Address Offset	See Table 14-217		
Physical Address	0x4480 5910 0x4480 5A10 0x4480 3C10 0x4480 5B10 0x4480 5C10 0x4480 3D10 0x4480 3E10 0x4480 3F10 0x4480 3A10 0x4480 4010 0x4480 4110	**Instance**	CLK1_2_BB2D_P1_BW_LIMITER CLK1_2_BB2D_P2_BW_LIMITER CLK1_2_TPTC1_RD_BW_LIMITER CLK1_2_GPU_P1_BW_LIMITER CLK1_2_GPU_P2_BW_LIMITER CLK1_2_TPTC2_RD_BW_LIMITER CLK1_2_TPTC1_WR_BW_LIMITER CLK1_2_TPTC2_WR_BW_LIMITER CLK1_2_MMU1_BW_LIMITER CLK1_2_VPE_P2_BW_LIMITER CLK1_2_VPE_P1_BW_LIMITER
Description			
Type	RW		

31–14	13–0
RESERVED	WATERMARK_0

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	WATERMARK_0	Peak bandwidth allowed Type: Control. Reset value: 0x3FF.	RW	0x3FFF

Table 14-230. Register Call Summary for Register L3_BW_LIMITER_WATERMARK_0

L3_MAIN Interconnect

- Bandwidth Limiters: [0] [1]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [2] [3] [4] [5]

Table 14-231. L3_BW_LIMITER_CLEARHISTORY

Address Offset	See Table 14-217		
Physical Address	0x4480 5914 0x4480 5A14 0x4480 3C14 0x4480 5B14 0x4480 5C14 0x4480 3D14 0x4480 3E14 0x4480 3F14 0x4480 3A14 0x4480 4014 0x4480 4114	**Instance**	CLK1_2_BB2D_P1_BW_LIMITER CLK1_2_BB2D_P2_BW_LIMITER CLK1_2_TPTC1_RD_BW_LIMITER CLK1_2_GPU_P1_BW_LIMITER CLK1_2_GPU_P2_BW_LIMITER CLK1_2_TPTC2_RD_BW_LIMITER CLK1_2_TPTC1_WR_BW_LIMITER CLK1_2_TPTC2_WR_BW_LIMITER CLK1_2_MMU1_BW_LIMITER CLK1_2_VPE_P2_BW_LIMITER CLK1_2_VPE_P1_BW_LIMITER
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLEARHISTORY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLEARHISTORY	Write a 1 clear the traffic counter Type: Give_AutoCleared. Reset value: 0x0.	RW	0

Table 14-232. Register Call Summary for Register L3_BW_LIMITER_CLEARHISTORY

L3_MAIN Interconnect

- Bandwidth Limiters: [0]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [1] [2] [3] [4]

14.2.5.1.9 L3_MAIN STATCOLL Register Summary and Description

Table 14-233. STATCOLL Instance Summary

Module Name	Base Address	Size
CLK3_FLAGMUX_STATCOLL	0x4500 0500	512 bytes
CLK2_STATCOLL0	0x4500 1000	512 bytes

Table 14-233. STATCOLL Instance Summary (continued)

Module Name	Base Address	Size
CLK2_STATCOLL1	0x4500 2000	512 bytes
CLK2_STATCOLL2	0x4500 3000	512 bytes
CLK2_STATCOLL3	0x4500 4000	512 bytes
CLK2_STATCOLL4	0x4500 5000	512 bytes
CLK2_STATCOLL5	0x4500 6000	512 bytes
CLK2_STATCOLL6	0x4500 7000	512 bytes
CLK2_STATCOLL7	0x4500 8000	512 bytes
CLK2_STATCOLL8	0x4500 9000	512 bytes
CLK2_STATCOLL9	0x4500 A000	512 bytes

14.2.5.1.9.1 L3_MAIN STATCOLL Register Summary

Table 14-234. STATCOLL Register Summary

Register Name	Type	Register Width (bits)	Address offset for FIAGMUX	CLK3_FLAGMUX_STATCOLL L3_MAIN Physical Address
L3_STCOL_STDHOSTHDR_COREREG	R	32	0x0100 0500	0x4500 0500
L3_STCOL_STDHOSTHDR_VERSIONREG	R	32	0x0100 0504	0x4500 0504
L3_STCOL_MASK0	RW	32	0x0100 0508	0x4500 0508
L3_STCOL_REGERR0	R	32	0x0100 050C	0x4500 050C

Table 14-235. STATCOLL Register Summary

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL0 L3_MAIN Physical Address	CLK2_STATCOLL1 L3_MAIN Physical Address	CLK2_STATCOLL2 L3_MAIN Physical Address	CLK2_STATCOLL3 L3_MAIN Physical Address	CLK2_STATCOLL4 L3_MAIN Physical Address
L3_STCOL_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4500 1 000	0x4500 2000	0x4500 3000	0x4500 4000	0x4500 5000
L3_STCOL_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4500 1004	0x4500 2004	0x4500 3004	0x4500 4004	0x4500 5004
L3_STCOL_EN	RW	32	0x0000 0008	0x4500 1008	0x4500 2008	0x4500 3008	0x4500 4008	0x4500 5008
L3_STCOL_SOFTEN	RW	32	0x0000 000C	0x4500 100C	0x4500 200C	0x4500 300C	0x4500 400C	0x4500 500C
L3_STCOL_IGNORESUSPEND	RW	32	0x0000 0010	0x4500 1010	0x4500 2010	0x4500 3010	0x4500 4010	0x4500 5010
L3_STCOL_TRIGEN	RW	32	0x0000 0014	0x4500 1014	0x4500 2014	0x4500 3014	0x4500 4014	0x4500 5014
L3_STCOL_REQEVT	RW	32	0x0000 0018	0x4500 1018	0x4500 2018	0x4500 3018	0x4500 4018	0x4500 5018
L3_STCOL_RSPEVT	RW	32	0x0000 001C	0x4500 101C	0x4500 201C	0x4500 301C	0x4500 401C	0x4500 501C
L3_STCOL_EVTMUX_SEL0	RW	32	0x0000 0020	0x45001020	0x4500 2020	0x4500 3020	0x4500 4020	0x4500 5020
L3_STCOL_EVTMUX_SEL1	RW	32	0x0000 0024	0x4500 1024	0x4500 2024	0x4500 3024	0x4500 4024	0x4500 5024
L3_STCOL_EVTMUX_SEL2	RW	32	0x0000 0028	0x4500 1028	0x4500 2028	0x4500 3028	0x4500 4028	0x4500 5028
L3_STCOL_EVTMUX_SEL3	RW	32	0x0000 002C	0x4500 102C	0x4500 202C	0x4500 302C	0x4500 402C	0x4500 502C
L3_STCOL_EVTMUX_SEL4	RW	32	0x0000 0030	0x4500 1030	0x4500 2030	N/A	0x4500 4030	N/A

Table 14-235. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL0 L3_MAIN Physical Address	CLK2_STATCOLL1 L3_MAIN Physical Address	CLK2_STATCOLL2 L3_MAIN Physical Address	CLK2_STATCOLL3 L3_MAIN Physical Address	CLK2_STATCOLL4 L3_MAIN Physical Address
L3_STCOL_EVTMUX_SEL5	RW	32	0x0000 0034	0x4500 1034	0x4500 2034	N/A	0x4500 4034	N/A
L3_STCOL_EVTMUX_SEL6	RW	32	0x0000 0038	0x4500 1038	N/A	N/A	0x4500 4038	N/A
L3_STCOL_EVTMUX_SEL7	RW	32	0x0000 003C	0x4500 103C	N/A	N/A	0x4500 403C	N/A
L3_STCOL_DUMP_IDENTIFIER	R	32	0x0000 0040	0x4500 1040	0x4500 2040	0x4500 3040	0x4500 4040	0x4500 5040
L3_STCOL_DUMP_COLLECTTIME	RW	32	0x0000 0044	0x4500 1044	0x4500 2044	0x4500 3044	0x4500 4044	0x4500 5044
L3_STCOL_DUMP_SLVADDR	R	32	0x0000 0048	0x4500 1048	0x4500 2048	0x4500 3048	0x4500 4048	0x4500 5048
L3_STCOL_DUMP_MSTADDR	R	32	0x0000 004C	0x4500 104C	0x4500 204C	0x4500 304C	0x4500 404C	0x4500 504C
L3_STCOL_DUMP_SLVOFS	RW	32	0x0000 0050	0x4500 1050	0x4500 2050	0x4500 3050	0x4500 4050	0x4500 5050
L3_STCOL_DUMP_MODE	RW	32	0x0000 0054	0x4500 1054	0x4500 2054	0x4500 3054	0x4500 4054	0x4500 5054
L3_STCOL_DUMP_SEND	RW	32	0x0000 0058	0x4500 1058	0x4500 2058	0x4500 3058	0x4500 4058	0x4500 5058
L3_STCOL_DUMP_DISABLE	RW	32	0x0000 005C	0x4500 105C	0x4500 205C	0x4500 305C	0x4500 405C	0x4500 505C
L3_STCOL_DUMP_ALARM_TRIG	RW	32	0x0000 0060	0x4500 1060	0x4500 2060	0x4500 3060	0x4500 4060	0x4500 5060
L3_STCOL_DUMP_ALARM_MINVAL	RW	32	0x0000 0064	0x4500 1064	0x4500 2064	0x4500 3064	0x4500 4064	0x4500 5064
L3_STCOL_DUMP_ALARM_MAXVAL	RW	32	0x0000 0068	0x4500 1068	0x4500 2068	0x4500 3068	0x4500 4068	0x4500 5068
L3_STCOL_DUMP_ALARM_MODE0	RW	32	0x0000 006C	0x4500 106C	0x4500 206C	0x4500 306C	0x4500 406C	0x4500 506C
L3_STCOL_DUMP_ALARM_MODE1	RW	32	0x0000 0070	0x4500 1070	0x4500 2070	0x4500 3070	0x4500 4070	0x4500 5070
L3_STCOL_DUMP_ALARM_MODE2	RW	32	0x0000 0074	0x4500 1074	0x4500 2074	0x4500 3074	0x4500 4074	0x4500 5074
L3_STCOL_DUMP_ALARM_MODE3	RW	32	0x0000 0078	0x4500 1078	0x4500 2078	0x4500 3078	0x4500 4078	0x4500 5078
L3_STCOL_DUMP_ALARM_MODE4	RW	32	0x0000 007C	0x4500 107C	0x4500 207C	N/A	0x4500 407C	N/A
L3_STCOL_DUMP_ALARM_MODE5	RW	32	0x0000 0080	0x4500 1080	0x4500 2080	N/A	0x4500 4080	N/A
L3_STCOL_DUMP_ALARM_MODE6	RW	32	0x0000 0084	0x4500 1084	N/A	N/A	0x4500 4084	N/A
L3_STCOL_DUMP_ALARM_MODE7	RW	32	0x0000 0088	0x4500 1088	N/A	N/A	0x4500 4088	N/A
L3_STCOL_DUMP_CNT0	R	32	0x0000 008C	0x4500 108C	0x4500 208C	0x4500 308C	0x4500 408C	0x4500 508C
L3_STCOL_DUMP_CNT1	R	32	0x0000 0090	0x4500 1090	0x4500 2090	0x4500 3090	0x4500 4090	0x4500 5090
L3_STCOL_DUMP_CNT2	R	32	0x0000 0094	0x4500 1094	0x4500 2094	0x4500 3094	0x4500 4094	0x4500 5094

Table 14-235. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL0 L3_MAIN Physical Address	CLK2_STATCOLL1 L3_MAIN Physical Address	CLK2_STATCOLL2 L3_MAIN Physical Address	CLK2_STATCOLL3 L3_MAIN Physical Address	CLK2_STATCOLL4 L3_MAIN Physical Address
L3_STCOL_DUMP_CNT3	R	32	0x0000 0098	0x4500 1098	0x4500 2098	0x4500 3098	0x4500 4098	0x4500 5098
L3_STCOL_DUMP_CNT4	R	32	0x0000 009C	0x4500 109C	0x4500 209C	N/A	0x4500 409C	N/A
L3_STCOL_DUMP_CNT5	R	32	0x0000 00A0	0x4500 10A0	0x4500 20A0	N/A	0x4500 40A0	N/A
L3_STCOL_DUMP_CNT6	R	32	0x0000 00A4	0x4500 10A4	N/A	N/A	0x4500 40A4	N/A
L3_STCOL_DUMP_CNT7	R	32	0x0000 00A8	0x4500 10A8	N/A	N/A	0x4500 40A8	N/A
L3_STCOL_FILTER_i_GLOBALEN (1)	RW	32	0x0000 00AC + (0x158*i)	0x4500 10AC + (0x158*i)	0x4500 20AC + (0x158*i)	0x4500 30AC + (0x158*i)	0x4500 40AC + (0x158*i)	0x4500 50AC + (0x158*i)
L3_STCOL_FILTER_i_ADDRMIN (1)	RW	32	0x0000 00B0 + (0x158*i)	0x4500 10B0 + (0x158*i)	0x4500 20B0 + (0x158*i)	0x4500 30B0 + (0x158*i)	0x4500 40B0 + (0x158*i)	0x4500 50B0 + (0x158*i)
L3_STCOL_FILTER_i_ADDRMAX (1)	RW	32	0x0000 00B4 + (0x158*i)	0x4500 10B4 + (0x158*i)	0x4500 20B4 + (0x158*i)	0x4500 30B4 + (0x158*i)	0x4500 40B4 + (0x158*i)	0x4500 50B4 + (0x158*i)
L3_STCOL_FILTER_i_ADDREN (1)	RW	32	0x0000 00B8 + (0x158*i)	0x4500 10B8 + (0x158*i)	0x4500 20B8 + (0x158*i)	0x4500 30B8 + (0x158*i)	0x4500 40B8 + (0x158*i)	0x4500 50B8 + (0x158*i)
L3_STCOL_FILTER_i_EN_k (1) (3)	RW	32	0x0000 00BC + (0x158*i) + (0x44*k)	0x4500 10BC + (0x158*i) + (0x44*k)	0x4500 20BC + (0x158*i) + (0x44*k)	0x4500 30BC + (0x158*i) + (0x44*k)	0x4500 40BC + (0x158*i) + (0x44*k)	0x4500 50BC + (0x158*i) + (0x44*k)
L3_STCOL_FILTER_i_MASK_m_RD (1) (2)	RW	32	0x0000 00C0 + (0x158*i) + (0x44*m)	0x4500 10C0 + (0x158*i) + (0x44*m)	0x4500 20C0 + (0x158*i) + (0x44*m)	0x4500 30C0 + (0x158*i) + (0x44*m)	0x4500 40C0 + (0x158*i) + (0x44*m)	0x4500 50C0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_WR (1) (2)	RW	32	0x0000 00C4 + (0x158*i) + (0x44*m)	0x4500 10C4 + (0x158*i) + (0x44*m)	0x4500 20C4 + (0x158*i) + (0x44*m)	0x4500 30C4 + (0x158*i) + (0x44*m)	0x4500 40C4 + (0x158*i) + (0x44*m)	0x4500 50C4 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_MSTADDR (1) (2)	RW	32	0x0000 00C8 + (0x158*i) + (0x44*m)	0x4500 10C8 + (0x158*i) + (0x44*m)	0x4500 20C8 + (0x158*i) + (0x44*m)	0x4500 30C8 + (0x158*i) + (0x44*m)	0x4500 40C8 + (0x158*i) + (0x44*m)	0x4500 50C8 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_SLVADDR (1) (2)	RW	32	0x0000 00CC + (0x158*i) + (0x44*m)	N/A	0x4500 20CC + (0x158*i) + (0x44*m)	0x4500 30CC + (0x158*i) + (0x44*m)	0x4500 40CC + (0x158*i) + (0x44*m)	0x4500 50CC + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_ERR (1) (2)	RW	32	0x0000 00D0 + (0x158*i) + (0x44*m)	0x4500 10D0 + (0x158*i) + (0x44*m)	0x4500 20D0 + (0x158*i) + (0x44*m)	0x4500 30D0 + (0x158*i) + (0x44*m)	0x4500 40D0 + (0x158*i) + (0x44*m)	0x4500 50D0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_USERINFO (1) (2)	RW	32	0x0000 00D4 + (0x158*i) + (0x44*m)	0x4500 10D4 + (0x158*i) + (0x44*m)	N/A	N/A	N/A	N/A
L3_STCOL_FILTER_i_MASK_m_REQUSERINFO (1) (2)	RW	32	0x0000 00D4 + (0x158*i) + (0x44*m)	N/A	0x4500 20D4 + (0x158*i) + (0x44*m)	0x4500 30D4 + (0x158*i) + (0x44*m)	0x4500 40D4 + (0x158*i) + (0x44*m)	0x4500 50D4 + (0x158*i) + (0x44*m)

Table 14-235. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL0 L3_MAIN Physical Address	CLK2_STATCOLL1 L3_MAIN Physical Address	CLK2_STATCOLL2 L3_MAIN Physical Address	CLK2_STATCOLL3 L3_MAIN Physical Address	CLK2_STATCOLL4 L3_MAIN Physical Address
L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO (1) (2)	RW	32	0x0000 00D8 + (0x158*i) + (0x44*m)	N/A	0x4500 20D8 + (0x158*i) + (0x44*m)	0x4500 30D8 + (0x158*i) + (0x44*m)	0x4500 40D8 + (0x158*i) + (0x44*m)	0x4500 50D8 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_RD (1) (2)	RW	32	0x0000 00E0 + (0x158*i) + (0x44*m)	0x4500 10E0 + (0x158*i) + (0x44*m)	0x4500 20E0 + (0x158*i) + (0x44*m)	0x4500 30E0 + (0x158*i) + (0x44*m)	0x4500 40E0 + (0x158*i) + (0x44*m)	0x4500 50E0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_WR (1) (2)	RW	32	0x0000 00E4 + (0x158*i) + (0x44*m)	0x4500 10E4 + (0x158*i) + (0x44*m)	0x4500 20E4 + (0x158*i) + (0x44*m)	0x4500 30E4 + (0x158*i) + (0x44*m)	0x4500 40E4 + (0x158*i) + (0x44*m)	0x4500 50E4 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_MSTADDR (1) (2)	RW	32	0x0000 00E8 + (0x158*i) + (0x44*m)	0x4500 10E8 + (0x158*i) + (0x44*m)	0x4500 20E8 + (0x158*i) + (0x44*m)	0x4500 30E8 + (0x158*i) + (0x44*m)	0x4500 40E8 + (0x158*i) + (0x44*m)	0x4500 50E8 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_SLVADDR (1) (2)	RW	32	0x0000 00EC + (0x158*i) + (0x44*m)	N/A	0x4500 20EC + (0x158*i) + (0x44*m)	0x4500 30EC + (0x158*i) + (0x44*m)	0x4500 40EC + (0x158*i) + (0x44*m)	0x4500 50EC + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_ERR (1) (2)	RW	32	0x0000 00F0 + (0x158*i) + (0x44*m)	0x4500 10F0 + (0x158*i) + (0x44*m)	0x4500 20F0 + (0x158*i) + (0x44*m)	0x4500 30F0 + (0x158*i) + (0x44*m)	0x4500 40F0 + (0x158*i) + (0x44*m)	0x4500 50F0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_USERINFO (1) (2)	RW	32	0x0000 00F4+ (0x158*i) + (0x44*m)	0x4500 10F4 + (0x158*i) + (0x44*m)	N/A	N/A	N/A	N/A
L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO (1) (2)	RW	32	0x0000 00F4 + (0x158*i) + (0x44*m)	N/A	0x4500 20F4 + (0x158*i) + (0x44*m)	0x4500 30F4 + (0x158*i) + (0x44*m)	0x4500 40F4 + (0x158*i) + (0x44*m)	0x4500 50F4 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO (1) (2)	RW	32	0x0000 00F8 + (0x158*i) + (0x44*m)	N/A	0x4500 20F8 + (0x158*i) + (0x44*m)	0x4500 30F8 + (0x158*i) + (0x44*m)	0x4500 40F8 + (0x158*i) + (0x44*m)	0x4500 50F8 + (0x158*i) + (0x44*m)
L3_STCOL_OP_i_THRESHOLD_MINVAL (1)	RW	32	0x0000 01F0 + (0x158*i)	0x4500 11F0 + (0x158*i)	0x4500 21F0 + (0x158*i)	0x4500 31F0 + (0x158*i)	0x4500 41F0 + (0x158*i)	0x4500 51F0 + (0x158*i)
L3_STCOL_OP_i_THRESHOLD_MAXVAL (1)	RW	32	0x0000 01F4 + (0x158*i)	0x4500 11F4 + (0x158*i)	0x4500 21F4 + (0x158*i)	0x4500 31F4 + (0x158*i)	0x4500 41F4 + (0x158*i)	0x4500 51F4 + (0x158*i)
L3_STCOL_OP_i_EVTINFOSEL (1)	RW	32	0x0000 01F8 + (0x158*i)	0x4500 11F8 + (0x158*i)	0x4500 21F8 + (0x158*i)	0x4500 31F8 + (0x158*i)	0x4500 41F8 + (0x158*i)	0x4500 51F8 + (0x158*i)
L3_STCOL_OP_i_SEL (1)	RW	32	0x0000 01FC + (0x158*i)	0x4500 11FC + (0x158*i)	0x4500 21FC + (0x158*i)	0x4500 31FC + (0x158*i)	0x4500 41FC + (0x158*i)	0x4500 51FC + (0x158*i)

(1) i = 0 to 7 for CLK2_STATCOLL0
i = 0 to 5 for CLK2_STATCOLL1
i = 0 to 3 for CLK2_STATCOLL2
i = 0 to 7 for CLK2_STATCOLL3
i = 0 to 3 for CLK2_STATCOLL4
(2) m = 0 to 1 for CLK2_STATCOLL0
m = 0 for CLK2_STATCOLL1

m = 0 for CLK2_STATCOLL2
m = 0 for CLK2_STATCOLL3
m = 0 for CLK2_STATCOLL4
(3) k = 0 to 1 for CLK2_STATCOLL0
k = 0 for CLK2_STATCOLL1
k = 0 for CLK2_STATCOLL2
k = 0 for CLK2_STATCOLL3
k = 0 for CLK2_STATCOLL4

Table 14-236. STATCOLL Register Summary

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL5 L3_MAIN Physical Address	CLK2_STATCOLL6 L3_MAIN Physical Address	CLK2_STATCOLL7 L3_MAIN Physical Address	CLK2_STATCOLL8 L3_MAIN Physical Address	CLK2_STATCOLL9 L3_MAIN Physical Address
L3_STCOL_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4500 6000	0x4500 7000	0x4500 8000	0x4500 9000	0x4500 A000
L3_STCOL_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4500 6004	0x4500 7004	0x4500 8004	0x4500 9004	0x4500 A004
L3_STCOL_EN	RW	32	0x0000 0008	0x4500 6008	0x4500 7008	0x4500 8008	0x4500 9008	0x4500 A008
L3_STCOL_SOFTEN	RW	32	0x0000 000C	0x4500 600C	0x4500 700C	0x4500 800C	0x4500 900C	0x4500 A00C
L3_STCOL_IGNORESUSPEND	RW	32	0x0000 0010	0x4500 6010	0x4500 7010	0x4500 8010	0x4500 9010	0x4500 A010
L3_STCOL_TRIGEN	RW	32	0x0000 0014	0x4500 6014	0x4500 7014	0x4500 8014	0x4500 9014	0x4500 A014
L3_STCOL_REQEVT	RW	32	0x0000 0018	0x4500 6018	0x4500 7018	0x4500 8018	0x4500 9018	0x4500 A018
L3_STCOL_RSPEVT	RW	32	0x0000 001C	0x4500 601C	0x4500 701C	0x4500 801C	0x4500 901C	0x4500 A01C
L3_STCOL_EVTMUX_SEL0	RW	32	0x0000 0020	0x4500 6020	0x4500 7020	0x4500 8020	0x4500 9020	0x4500 A020
L3_STCOL_EVTMUX_SEL1	RW	32	0x0000 0024	0x4500 6024	0x4500 7024	0x4500 8024	0x4500 9024	0x4500 A024
L3_STCOL_EVTMUX_SEL2	RW	32	0x0000 0028	0x4500 6028	0x4500 7028	0x4500 8028	0x4500 9028	0x4500 A028
L3_STCOL_EVTMUX_SEL3	RW	32	0x0000 002C	0x4500 602C	0x4500 702C	0x4500 802C	0x4500 902C	0x4500 A02C
L3_STCOL_EVTMUX_SEL4	RW	32	0x0000 0030	N/A	N/A	N/A	N/A	N/A
L3_STCOL_EVTMUX_SEL5	RW	32	0x0000 0034	N/A	N/A	N/A	N/A	N/A
L3_STCOL_EVTMUX_SEL6	RW	32	0x0000 0038	N/A	N/A	N/A	N/A	N/A
L3_STCOL_EVTMUX_SEL7	RW	32	0x0000 003C	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_IDENTIFIER	R	32	0x0000 0040	0x4500 6040	0x4500 7040	0x4500 8040	0x4500 9040	0x4500 A040
L3_STCOL_DUMP_COLLECTTIME	RW	32	0x0000 0044	0x4500 6044	0x4500 7044	0x4500 8044	0x4500 9044	0x4500 A044
L3_STCOL_DUMP_SLVADDR	R	32	0x0000 0048	0x4500 6048	0x4500 7048	0x4500 8048	0x4500 9048	0x4500 A048
L3_STCOL_DUMP_MSTADDR	R	32	0x0000 004C	0x4500 604C	0x4500 704C	0x4500 804C	0x4500 904C	0x4500 A04C
L3_STCOL_DUMP_SLVOFS	RW	32	0x0000 0050	0x4500 6050	0x4500 7050	0x4500 8050	0x4500 9050	0x4500 A050
L3_STCOL_DUMP_MODE	RW	32	0x0000 0054	0x4500 6054	0x4500 7054	0x4500 8054	0x4500 9054	0x4500 A054

Table 14-236. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL5 L3_MAIN Physical Address	CLK2_STATCOLL6 L3_MAIN Physical Address	CLK2_STATCOLL7 L3_MAIN Physical Address	CLK2_STATCOLL8 L3_MAIN Physical Address	CLK2_STATCOLL9 L3_MAIN Physical Address
L3_STCOL_DUMP_SEND	RW	32	0x0000 0058	0x4500 6058	0x4500 7058	0x4500 8058	0x4500 9058	0x4500 A058
L3_STCOL_DUMP_DISABLE	RW	32	0x0000 005C	0x4500 605C	0x4500 705C	0x4500 805C	0x4500 905C	0x4500 A05C
L3_STCOL_DUMP_ALARM_TRIG	RW	32	0x0000 0060	0x4500 6060	0x4500 7060	0x4500 8060	0x4500 9060	0x4500 A060
L3_STCOL_DUMP_ALARM_MINVAL	RW	32	0x0000 0064	0x4500 6064	0x4500 7064	0x4500 8064	0x4500 9064	0x4500 A064
L3_STCOL_DUMP_ALARM_MAXVAL	RW	32	0x0000 0068	0x4500 6068	0x4500 7068	0x4500 8068	0x4500 9068	0x4500 A068
L3_STCOL_DUMP_ALARM_MODE0	RW	32	0x0000 006C	0x4500 606C	0x4500 706C	0x4500 806C	0x4500 906C	0x4500 A06C
L3_STCOL_DUMP_ALARM_MODE1	RW	32	0x0000 0070	0x4500 6070	0x4500 7070	0x4500 8070	0x4500 9070	0x4500 A070
L3_STCOL_DUMP_ALARM_MODE2	RW	32	0x0000 0074	0x4500 6074	0x4500 7074	0x4500 8074	0x4500 9074	0x4500 A074
L3_STCOL_DUMP_ALARM_MODE3	RW	32	0x0000 0078	0x4500 6078	0x4500 7078	0x4500 8078	0x4500 9078	0x4500 A078
L3_STCOL_DUMP_ALARM_MODE4	RW	32	0x0000 007C	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_ALARM_MODE5	RW	32	0x0000 0080	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_ALARM_MODE6	RW	32	0x0000 0084	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_ALARM_MODE7	RW	32	0x0000 0088	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_CNT0	R	32	0x0000 008C	0x4500 608C	0x4500 708C	0x4500 808C	0x4500 908C	0x4500 A08C
L3_STCOL_DUMP_CNT1	R	32	0x0000 0090	0x4500 6090	0x4500 7090	0x4500 8090	0x4500 9090	0x4500 A090
L3_STCOL_DUMP_CNT2	R	32	0x0000 0094	0x4500 6094	0x4500 7094	0x4500 8094	0x4500 9094	0x4500 A094
L3_STCOL_DUMP_CNT3	R	32	0x0000 0098	0x4500 6098	0x4500 7098	0x4500 8098	0x4500 9098	0x4500 A098
L3_STCOL_DUMP_CNT4	R	32	0x0000 009C	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_CNT5	R	32	0x0000 00A0	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_CNT6	R	32	0x0000 00A4	N/A	N/A	N/A	N/A	N/A
L3_STCOL_DUMP_CNT7	R	32	0x0000 00A8	N/A	N/A	N/A	N/A	N/A
L3_STCOL_FILTER_i_GLOBALEN [1]	RW	32	0x0000 00AC + (0x158*i)	0x4500 60AC + (0x158*i)	0x4500 70AC + (0x158*i)	0x4500 80AC + (0x158*i)	0x4500 90AC + (0x158*i)	0x4500 A0AC + (0x158*i)
L3_STCOL_FILTER_i_ADDRMIN [1]	RW	32	0x0000 00B0 + (0x158*i)	0x4500 60B0 + (0x158*i)	0x4500 70B0 + (0x158*i)	0x4500 80B0 + (0x158*i)	0x4500 90B0 + (0x158*i)	0x4500 A0B0 + (0x158*i)
L3_STCOL_FILTER_i_ADDRMAX [1]	RW	32	0x0000 00B4 + (0x158*i)	0x4500 60B4 + (0x158*i)	0x4500 70B4 + (0x158*i)	0x4500 80B4 + (0x158*i)	0x4500 90B4 + (0x158*i)	0x4500 A0B4 + (0x158*i)

Table 14-236. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL5 L3_MAIN Physical Address	CLK2_STATCOLL6 L3_MAIN Physical Address	CLK2_STATCOLL7 L3_MAIN Physical Address	CLK2_STATCOLL8 L3_MAIN Physical Address	CLK2_STATCOLL9 L3_MAIN Physical Address
L3_STCOL_FILTER_i_ADDREN [(1)]	RW	32	0x0000 00B8 + (0x158*i)	0x4500 60B8 + (0x158*i)	0x4500 70B8 + (0x158*i)	0x4500 80B8 + (0x158*i)	0x4500 90B8 + (0x158*i)	0x4500 A0B8 + (0x158*i)
L3_STCOL_FILTER_i_EN_k [(1)] [(3)]	RW	32	0x0000 00BC + (0x158*i) + (0x44*k)	0x4500 60BC + (0x158*i) + (0x44*k)	0x4500 70BC + (0x158*i) + (0x44*k)	0x4500 80BC + (0x158*i) + (0x44*k)	0x4500 90BC + (0x158*i) + (0x44*k)	0x4500 A0BC + (0x158*i) + (0x44*k)
L3_STCOL_FILTER_i_MASK_m_RD [(1)] [(2)]	RW	32	0x0000 00C0 + (0x158*i) + (0x44*m)	0x4500 60C0 + (0x158*i) + (0x44*m)	0x4500 70C0 + (0x158*i) + (0x44*m)	0x4500 80C0 + (0x158*i) + (0x44*m)	0x4500 90C0 + (0x158*i) + (0x44*m)	0x4500 A0C0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_WR [(1)] [(2)]	RW	32	0x0000 00C4 + (0x158*i) + (0x44*m)	0x4500 60C4 + (0x158*i) + (0x44*m)	0x4500 70C4 + (0x158*i) + (0x44*m)	0x4500 80C4 + (0x158*i) + (0x44*m)	0x4500 90C4 + (0x158*i) + (0x44*m)	0x4500 A0C4 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_MSTADDR [(1)] [(2)]	RW	32	0x0000 00C8 + (0x158*i) + (0x44*m)	0x4500 60C8 + (0x158*i) + (0x44*m)	0x4500 70C8 + (0x158*i) + (0x44*m)	0x4500 80C8 + (0x158*i) + (0x44*m)	0x4500 90C8 + (0x158*i) + (0x44*m)	0x4500 A0C8 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_SLVADDR [(1)] [(2)]	RW	32	0x0000 00CC + (0x158*i) + (0x44*m)	0x4500 60CC + (0x158*i) + (0x44*m)	0x4500 70CC + (0x158*i) + (0x44*m)	0x4500 80CC + (0x158*i) + (0x44*m)	0x4500 90CC + (0x158*i) + (0x44*m)	0x4500 A0CC + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_ERR [(1)] [(2)]	RW	32	0x0000 00D0 + (0x158*i) + (0x44*m)	0x4500 60D0 + (0x158*i) + (0x44*m)	0x4500 70D0 + (0x158*i) + (0x44*m)	0x4500 80D0 + (0x158*i) + (0x44*m)	0x4500 90D0 + (0x158*i) + (0x44*m)	0x4500 A0D0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_USERINFO [(1)] [(2)]	RW	32	0x0000 00D4 + (0x158*i) + (0x44*m)	N/A	N/A	N/A	N/A	N/A
L3_STCOL_FILTER_i_MASK_m_REQUSERINFO [(1)] [(2)]	RW	32	0x0000 00D4 + (0x158*i) + (0x44*m)	0x4500 60D4 + (0x158*i) + (0x44*m)	0x4500 70D4 + (0x158*i) + (0x44*m)	0x4500 80D4 + (0x158*i) + (0x44*m)	0x4500 90D4 + (0x158*i) + (0x44*m)	0x4500 A0D4 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO [(1)] [(2)]	RW	32	0x0000 00D8 + (0x158*i) + (0x44*m)	0x4500 60D8 + (0x158*i) + (0x44*m)	0x4500 70D8 + (0x158*i) + (0x44*m)	0x4500 80D8 + (0x158*i) + (0x44*m)	0x4500 90D8 + (0x158*i) + (0x44*m)	0x4500 A0D8 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_RD [(1)] [(2)]	RW	32	0x0000 00E0 + (0x158*i) + (0x44*m)	0x4500 60E0 + (0x158*i) + (0x44*m)	0x4500 70E0 + (0x158*i) + (0x44*m)	0x4500 80E0 + (0x158*i) + (0x44*m)	0x4500 90E0 + (0x158*i) + (0x44*m)	0x4500 A0E0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_WR [(1)] [(2)]	RW	32	0x0000 00E4 + (0x158*i) + (0x44*m)	0x4500 60E4 + (0x158*i) + (0x44*m)	0x4500 70E4 + (0x158*i) + (0x44*m)	0x4500 80E4 + (0x158*i) + (0x44*m)	0x4500 90E4 + (0x158*i) + (0x44*m)	0x4500 A0E4 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_MSTADDR [(1)] [(2)]	RW	32	0x0000 00E8 + (0x158*i) + (0x44*m)	0x4500 60E8 + (0x158*i) + (0x44*m)	0x4500 70E8 + (0x158*i) + (0x44*m)	0x4500 80E8 + (0x158*i) + (0x44*m)	0x4500 90E8 + (0x158*i) + (0x44*m)	0x4500 A0E8 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_SLVADDR [(1)] [(2)]	RW	32	0x0000 00EC + (0x158*i) + (0x44*m)	0x4500 60EC + (0x158*i) + (0x44*m)	0x4500 70EC + (0x158*i) + (0x44*m)	0x4500 80EC + (0x158*i) + (0x44*m)	0x4500 90EC + (0x158*i) + (0x44*m)	0x4500 A0EC + (0x158*i) + (0x44*m)

Table 14-236. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL5 L3_MAIN Physical Address	CLK2_STATCOLL6 L3_MAIN Physical Address	CLK2_STATCOLL7 L3_MAIN Physical Address	CLK2_STATCOLL8 L3_MAIN Physical Address	CLK2_STATCOLL9 L3_MAIN Physical Address
L3_STCOL_FILTER_i_MATCH_m_ERR (1) (2)	RW	32	0x0000 00F0 + (0x158*i) + (0x44*m)	0x4500 60F0 + (0x158*i) + (0x44*m)	0x4500 70F0 + (0x158*i) + (0x44*m)	0x4500 80F0 + (0x158*i) + (0x44*m)	0x4500 90F0 + (0x158*i) + (0x44*m)	0x4500 A0F0 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_USERINFO (1) (2)	RW	32	0x0000 00F4+ (0x158*i) + (0x44*m)	N/A	N/A	N/A	N/A	N/A
L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO (1) (2)	RW	32	0x0000 00F4 + (0x158*i) + (0x44*m)	0x4500 260F4 + (0x158*i) + (0x44*m)	0x4500 70F4 + (0x158*i) + (0x44*m)	0x4500 80F4 + (0x158*i) + (0x44*m)	0x4500 90F4 + (0x158*i) + (0x44*m)	0x4500 A0F4 + (0x158*i) + (0x44*m)
L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO (1) (2)	RW	32	0x0000 00F8 + (0x158*i) + (0x44*m)	0x4500 60F8 + (0x158*i) + (0x44*m)	0x4500 70F8 + (0x158*i) + (0x44*m)	0x4500 80F8 + (0x158*i) + (0x44*m)	0x4500 90F8 + (0x158*i) + (0x44*m)	0x4500 A0F8 + (0x158*i) + (0x44*m)
L3_STCOL_OP_i_THRESHOLD_MINVAL (1)	RW	32	0x0000 01F0 + (0x158*i)	0x4500 61F0 + (0x158*i)	0x4500 71F0 + (0x158*i)	0x4500 81F0 + (0x158*i)	0x4500 91F0 + (0x158*i)	0x4500 A1F0 + (0x158*i)
L3_STCOL_OP_i_THRESHOLD_MAXVAL (1)	RW	32	0x0000 01F4 + (0x158*i)	0x4500 61F4 + (0x158*i)	0x4500 71F4 + (0x158*i)	0x4500 81F4 + (0x158*i)	0x4500 91F4 + (0x158*i)	0x4500 A1F4 + (0x158*i)
L3_STCOL_OP_i_EVTINFOSEL (1)	RW	32	0x0000 01F8 + (0x158*i)	0x4500 61F8 + (0x158*i)	0x4500 71F8 + (0x158*i)	0x4500 81F8 + (0x158*i)	0x4500 91F8 + (0x158*i)	0x4500 A1F8 + (0x158*i)
L3_STCOL_OP_i_SEL (1)	RW	32	0x0000 01FC + (0x158*i)	0x4500 61FC + (0x158*i)	0x4500 71FC + (0x158*i)	0x4500 81FC + (0x158*i)	0x4500 91FC + (0x158*i)	0x4500 A1FC + (0x158*i)

(1) i = 0 to 3 for CLK2_STATCOLL5
i = 0 to 3 for CLK2_STATCOLL6
i = 0 to 3 for CLK2_STATCOLL7
i = 0 to 3 for CLK2_STATCOLL8
i = 0 to 3 for CLK2_STATCOLL9
(2) m = 0 for CLK2_STATCOLL5
m = 0 for CLK2_STATCOLL6
m = 0 for CLK2_STATCOLL7
m = 0 for CLK2_STATCOLL8
m = 0 for CLK2_STATCOLL9
(3) k = 0 for CLK2_STATCOLL5
k = 0 for CLK2_STATCOLL6
k = 0 for CLK2_STATCOLL7
k = 0 for CLK2_STATCOLL8
k = 0 for CLK2_STATCOLL9

14.2.5.1.9.2 L3_MAIN STATCOLL Register Description

Table 14-237. L3_STCOL_STDHOSTHDR_COREREG

Address Offset	See Table 14-235.

Table 14-237. L3_STCOL_STDHOSTHDR_COREREG (continued)

Physical Address	0x4500 0500 0x4500 1 000 0x4500 2000 0x4500 3000 0x4500 4000 0x4500 5000 0x4500 6000 0x4500 7000 0x4500 8000 0x4500 9000 0x4500 A000	**Instance**	CLK3_FLAGMUX_STATCOLL CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31:22	21:16	15:1	0
RESERVED	STDHOSTHDR_COREREG_CORECODE	RESERVED	STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x3A. (When the instance is CLK3_FLAGMUX_STATCOLL reset value is 0x37)	R	0x3A
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R1	1

Table 14-238. Register Call Summary for Register L3_STCOL_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1] [2]

Table 14-239. L3_STCOL_STDHOSTHDR_VERSIONREG

Address Offset	See Table 14-235.

Table 14-239. L3_STCOL_STDHOSTHDR_VERSIONREG (continued)

Physical Address	0x4500 0504 0x4500 1004 0x4500 2004 0x4500 3004 0x4500 4004 0x4500 5004 0x4500 6004 0x4500 7004 0x4500 8004 0x4500 9004 0x4500 A004	**Instance**	CLK3_FLAGMUX_STATCOLL CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x1.	R	0x1
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 14-240. Register Call Summary for Register L3_STCOL_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1] [2]

Table 14-241. L3_STCOL_MASK0

Address Offset	See Table 14-234.		
Physical Address	0x4500 0508	**Instance**	CLK3_FLAGMUX_STATCOLL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						MASK0									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000 0000
9:0	MASK0	mask flag inputs 0 Type: Control. Reset value: 0x7.	RW	0x3ff

Table 14-242. Register Call Summary for Register L3_STCOL_MASK0

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 14-243. L3_STCOL_REGERR0

Address Offset	See Table 14-234.		
Physical Address	0x4500 050C	**Instance**	CLK3_FLAGMUX_STATCOLL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	REGERR0

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000 0000
9:0	REGERR0	flag inputs 0 Type: Status. Reset value: X.	R	0x0000 0000

Table 14-244. Register Call Summary for Register L3_STCOL_REGERR0

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 14-245. L3_STCOL_EN

Address Offset	See Table 14-235.		
Physical Address	0x4500 1008 0x4500 2008 0x4500 3008 0x4500 4008 0x4500 5008 0x4500 6008 0x4500 7008 0x4500 8008 0x4500 9008 0x4500 A008	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	EN	Enable performance monitoring, this will also shut down the clock if En = 0 Type: Control. Reset value: 0x0.	RW	0

Table 14-246. Register Call Summary for Register L3_STCOL_EN

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1] [2]

Table 14-247. L3_STCOL_SOFTEN

Address Offset	See Table 14-235.		
Physical Address	0x4500 100C 0x4500 200C 0x4500 300C 0x4500 400C 0x4500 500C 0x4500 600C 0x4500 700C 0x4500 800C 0x4500 900C 0x4500 A00C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SOFTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	SOFTEN	Software enable for performance monitoring Type: Control. Reset value: 0x0.	RW	0

Table 14-248. Register Call Summary for Register L3_STCOL_SOFTEN

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1] [2]

Table 14-249. L3_STCOL_IGNORESUSPEND

Address Offset	See Table 14-235.		
Physical Address	0x4500 1010 0x4500 2010 0x4500 3010 0x4500 4010 0x4500 5010 0x4500 6010 0x4500 7010 0x4500 8010 0x4500 9010 0x4500 A010	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															IGNORESUSPEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	IGNORESUSPEND	Ignore suspend if set to one for suspend mechanism Type: Control. Reset value: 0x0.	RW	0

Table 14-250. Register Call Summary for Register L3_STCOL_IGNORESUSPEND

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-251. L3_STCOL_TRIGEN

Address Offset	See Table 14-235.

Table 14-251. L3_STCOL_TRIGEN (continued)

Physical Address	0x4500 1014 0x4500 2014 0x4500 3014 0x4500 4014 0x4500 5014 0x4500 6014 0x4500 7014 0x4500 8014 0x4500 9014 0x4500 A014	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TRIGEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	TRIGEN	TrigEn when set, it enable the external trigger start and stop Type: Control. Reset value: 0x0.	RW	0

Table 14-252. Register Call Summary for Register L3_STCOL_TRIGEN

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-253. L3_STCOL_REQEVT

Address Offset	See Table 14-235.		
Physical Address	0x4500 1018 0x4500 2018 0x4500 3018 0x4500 4018 0x4500 5018 0x4500 6018 0x4500 7018 0x4500 8018 0x4500 9018 0x4500 A018	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												REQEVT			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000

Bits	Field Name	Description	Type	Reset
3:0	REQEVT	Req event select Type: Control. Reset value: 0x0. 0x0: Collect is disabled default value 0x1: Collect all event: hit always (cycle) 0x2: Collect transfers: actually used cycle for transferring aN NTTP word 0x3: Collect wait cycle: transfer has been delayed by source 0x4: Collect busy: transfer has been delayed by destination 0x5: Collect packet: new packet start 0x6: Collect data: data cycle transfer, write for requests, read for responses 0x7: Collect idles: transfer is not initiated by source 0x8: Collect latency: hit when actually detecting debug bit on response links	RW	0x0

Table 14-254. Register Call Summary for Register L3_STCOL_REQEVT

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1] [2]

Table 14-255. L3_STCOL_RSPEVT

Address Offset	See Table 14-235.		
Physical Address	0x4500 101C 0x4500 201C 0x4500 301C 0x4500 401C 0x4500 501C 0x4500 601C 0x4500 701C 0x4500 801C 0x4500 901C 0x4500 A01C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–4	3–0
RESERVED	RSPEVT

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000

Bits	Field Name	Description	Type	Reset
3:0	RSPEVT	Rsp event select Type: Control. Reset value: 0x0. 0x0: Collect is disabled default value 0x1: Collect all event: hit always (cycle) 0x2: Collect transfers: actually used cycle for transferring a NTTP word 0x3: Collect wait cycle: transfer has been delayed by source 0x4: Collect busy: transfer has been delayed by destination 0x5: Collect packet: new packet start 0x6: Collect data: data cycle transfer, write for requests, read for responses 0x7: Collect idles: transfer is not initiated by source 0x8: Collect latency: hit when actually detecting debug bit on response links	RW	0x0

Table 14-256. Register Call Summary for Register L3_STCOL_RSPEVT

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1] [2]

Table 14-257. L3_STCOL_EVTMUX_SEL0

Address Offset	See Table 14-235.		
Physical Address	0x45001020 0x4500 2020 0x4500 3020 0x4500 4020 0x4500 5020 0x4500 6020 0x4500 7020 0x4500 8020 0x4500 9020 0x4500 A020	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–3	2–0
RESERVED	EVTMUX_SEL0

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL0	The select of the mux 0 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-258. Register Call Summary for Register L3_STCOL_EVTMUX_SEL0

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1] [2]

Table 14-259. L3_STCOL_EVTMUX_SEL1

Address Offset	See Table 14-235.

Table 14-259. L3_STCOL_EVTMUX_SEL1 (continued)

Physical Address	0x4500 1024 0x4500 2024 0x4500 3024 0x4500 4024 0x4500 5024 0x4500 6024 0x4500 7024 0x4500 8024 0x4500 9024 0x4500 A024	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–3	2–0
RESERVED	EVTMUX_SEL1

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL1	The select of the mux 1 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-260. Register Call Summary for Register L3_STCOL_EVTMUX_SEL1

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-261. L3_STCOL_EVTMUX_SEL2

Address Offset	See Table 14-235.		
Physical Address	0x4500 1028 0x4500 2028 0x4500 3028 0x4500 4028 0x4500 5028 0x4500 6028 0x4500 7028 0x4500 8028 0x4500 9028 0x4500 A028	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–3	2–0
RESERVED	EVTMUX_SEL2

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL2	The select of the mux 2 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-262. Register Call Summary for Register L3_STCOL_EVTMUX_SEL2

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-263. L3_STCOL_EVTMUX_SEL3

Address Offset	See Table 14-235.

Table 14-263. L3_STCOL_EVTMUX_SEL3 (continued)

Physical Address	0x4500 102C 0x4500 202C 0x4500 302C 0x4500 402C 0x4500 502C 0x4500 602C 0x4500 702C 0x4500 802C 0x4500 902C 0x4500 A02C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													EVTMUX_SEL3		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL3	The select of the mux 3 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-264. Register Call Summary for Register L3_STCOL_EVTMUX_SEL3

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-265. L3_STCOL_EVTMUX_SEL4

Address Offset	See Table 14-235.		
Physical Address	0x4500 1030 0x4500 2030 0x4500 4030	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													EVTMUX_SEL4		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL4	The select of the mux 4 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-266. Register Call Summary for Register L3_STCOL_EVTMUX_SEL4

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-267. L3_STCOL_EVTMUX_SEL5

Address Offset	See Table 14-235.		
Physical Address	0x4500 1034 0x4500 2034 0x4500 4034	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	EVTMUX_SEL5

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL5	The select of the mux 5 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-268. Register Call Summary for Register L3_STCOL_EVTMUX_SEL5

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-269. L3_STCOL_EVTMUX_SEL6

Address Offset	See Table 14-235.		
Physical Address	0x4500 1038 0x4500 4038	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													EVTMUX_SEL6		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL6	The select of the mux 6 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-270. Register Call Summary for Register L3_STCOL_EVTMUX_SEL6

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-271. L3_STCOL_EVTMUX_SEL7

Address Offset	See Table 14-235.		
Physical Address	0x4500 103C 0x4500 403C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													EVTMUX_SEL7		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL7	The select of the mux 7 Type: Control. Reset value: 0x0.	RW	0x0

Table 14-272. Register Call Summary for Register L3_STCOL_EVTMUX_SEL7

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-273. L3_STCOL_DUMP_IDENTIFIER

Address Offset	See Table 14-235.

Table 14-273. L3_STCOL_DUMP_IDENTIFIER (continued)

Physical Address	0x4500 1040 0x4500 2040 0x4500 3040 0x4500 4040 0x4500 5040 0x4500 6040 0x4500 7040 0x4500 8040 0x4500 9040 0x4500 A040	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												DUMP_IDENTIFIER			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000 0000
3:0	DUMP_IDENTIFIER	Probe identifier Type: Control. Reset value: 0x0.	R	0x0

Table 14-274. Register Call Summary for Register L3_STCOL_DUMP_IDENTIFIER

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-275. L3_STCOL_DUMP_COLLECTTIME

Address Offset	See Table 14-235.		
Physical Address	0x4500 1044 0x4500 2044 0x4500 3044 0x4500 4044 0x4500 5044 0x4500 6044 0x4500 7044 0x4500 8044 0x4500 9044 0x4500 A044	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_COLLECTTIME																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_COLLECTTIME	Number of cycle to wait between two statistics frame Type: Control. Reset value: 0x0.	RW	0x0000

Table 14-276. Register Call Summary for Register L3_STCOL_DUMP_COLLECTTIME

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-277. L3_STCOL_DUMP_SLVADDR

Address Offset	See Table 14-235.

Table 14-277. L3_STCOL_DUMP_SLVADDR (continued)

Physical Address	0x4500 1048 0x4500 2048 0x4500 3048 0x4500 4048 0x4500 5048 0x4500 6048 0x4500 7048 0x4500 8048 0x4500 9048 0x4500 A048	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									DUMP_SLVADDR						

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	DUMP_SLVADDR	Dump slave address Type: Control. Reset value: 0x19.	R	0x19

Table 14-278. Register Call Summary for Register L3_STCOL_DUMP_SLVADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-279. L3_STCOL_DUMP_MSTADDR

Address Offset	See Table 14-235.		
Physical Address	0x4500 104C 0x4500 204C 0x4500 304C 0x4500 404C 0x4500 504C 0x4500 604C 0x4500 704C 0x4500 804C 0x4500 904C 0x4500 A04C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DUMP_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	DUMP_MSTADDR	Dump master address Type: Control. Reset value: 0xE0.	R	0x380

Table 14-280. Register Call Summary for Register L3_STCOL_DUMP_MSTADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-281. L3_STCOL_DUMP_SLVOFS

Address Offset	See Table 14-235.

Table 14-281. L3_STCOL_DUMP_SLVOFS (continued)

Physical Address	0x4500 1050 0x4500 2050 0x4500 3050 0x4500 4050 0x4500 5050 0x4500 6050 0x4500 7050 0x4500 8050 0x4500 9050 0x4500 A050	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_SLVOFS																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_SLVOFS	Dump slave offset Type: Control. Reset value: 0x800.	RW	0x0000 0800

Table 14-282. Register Call Summary for Register L3_STCOL_DUMP_SLVOFS

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-283. L3_STCOL_DUMP_MODE

Address Offset	See Table 14-235.		
Physical Address	0x4500 1054 0x4500 2054 0x4500 3054 0x4500 4054 0x4500 5054 0x4500 6054 0x4500 7054 0x4500 8054 0x4500 9054 0x4500 A054	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_MODE_CONDITIONAL	DUMP_MANUAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000

Bits	Field Name	Description	Type	Reset
1	DUMP_MODE_CONDITIONAL	Define the stat conditional dump, if one a dump will be generated when alarm is trigged Type: Control. Reset value: 0x0.	RW	0
0	DUMP_MODE_MANUAL	Define the dump mode: if != 0 the dump is controlled by the Send register. Type: Control. Reset value: 0x0.	RW	0

Table 14-284. Register Call Summary for Register L3_STCOL_DUMP_MODE

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-285. L3_STCOL_DUMP_SEND

Address Offset	See Table 14-235.		
Physical Address	0x4500 1058 0x4500 2058 0x4500 3058 0x4500 4058 0x4500 5058 0x4500 6058 0x4500 7058 0x4500 8058 0x4500 9058 0x4500 A058	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DUMP_SEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	DUMP_SEND	In manual mode, is used to send the dump content and initialize the counters. Type: Give_AutoCleared. Reset value: 0x0. • Dumping can be performed only if monitoring is enabled • For “one shot metrics dump” the DUMP_SEND command has to be issued before disabling monitoring	RW	0

Table 14-286. Register Call Summary for Register L3_STCOL_DUMP_SEND

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-287. L3_STCOL_DUMP_DISABLE

Address Offset	See Table 14-235.

Table 14-287. L3_STCOL_DUMP_DISABLE (continued)

Physical Address	0x4500 105C 0x4500 205C 0x4500 305C 0x4500 405C 0x4500 505C 0x4500 605C 0x4500 705C 0x4500 805C 0x4500 905C 0x4500 A05C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–1	0
RESERVED	DUMP_DISABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DUMP_DISABLE	If 1, the dump frame will be disabled, but counters still active.This is typically used when counters monitoring is enabled Type: Control. Reset value: 0x0.	RW	0

Table 14-288. Register Call Summary for Register L3_STCOL_DUMP_DISABLE

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-289. L3_STCOL_DUMP_ALARM_TRIG

Address Offset	See Table 14-235.		
Physical Address	0x4500 1060 0x4500 2060 0x4500 3060 0x4500 4060 0x4500 5060 0x4500 6060 0x4500 7060 0x4500 8060 0x4500 9060 0x4500 A060	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–1	0
RESERVED	DUMP_ALARM_TRIG

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
0	DUMP_ALARM_TRIG	In Alarm Mode, is used to reset Alarm Type: Take. Reset value: 0x0.	RW	0

Table 14-290. Register Call Summary for Register L3_STCOL_DUMP_ALARM_TRIG

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-291. L3_STCOL_DUMP_ALARM_MINVAL

Address Offset	See Table 14-235.		
Physical Address	0x4500 1064	**Instance**	CLK2_STATCOLL0
	0x4500 2064		CLK2_STATCOLL1
	0x4500 3064		CLK2_STATCOLL2
	0x4500 4064		CLK2_STATCOLL3
	0x4500 5064		CLK2_STATCOLL4
	0x4500 6064		CLK2_STATCOLL5
	0x4500 7064		CLK2_STATCOLL6
	0x4500 8064		CLK2_STATCOLL7
	0x4500 9064		CLK2_STATCOLL8
	0x4500 A064		CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_ALARM_MINVAL																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_ALARM_MINVAL	In Alarm Mode, used to trig an alert if any of counter value is less than AlarmMinVal Type: Control. Reset value: 0x0.	RW	0x0000 0000

Table 14-292. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MINVAL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-293. L3_STCOL_DUMP_ALARM_MAXVAL

Address Offset	See Table 14-235.		
Physical Address	0x4500 1068	**Instance**	CLK2_STATCOLL0
	0x4500 2068		CLK2_STATCOLL1
	0x4500 3068		CLK2_STATCOLL2
	0x4500 4068		CLK2_STATCOLL3
	0x4500 5068		CLK2_STATCOLL4
	0x4500 6068		CLK2_STATCOLL5
	0x4500 7068		CLK2_STATCOLL6
	0x4500 8068		CLK2_STATCOLL7
	0x4500 9068		CLK2_STATCOLL8
	0x4500 A068		CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_ALARM_MAXVAL																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_ALARM_MAXVAL	In Alarm Mode, used to trig an alert if any of counter value is larger or equal to AlarmMaxVal Type: Control. Reset value: 0x0.	RW	0x0000 0000

Table 14-294. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MAXVAL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-295. L3_STCOL_DUMP_ALARM_MODE0

Address Offset	See Table 14-235.		
Physical Address	0x4500 106C 0x4500 206C 0x4500 306C 0x4500 406C 0x4500 506C 0x4500 606C 0x4500 706C 0x4500 806C 0x4500 906C 0x4500 A06C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_ALARM_MODE0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE0	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 14-296. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE0

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-297. L3_STCOL_DUMP_ALARM_MODE1

Address Offset	See Table 14-235.		
Physical Address	0x4500 1070 0x4500 2070 0x4500 3070 0x4500 4070 0x4500 5070 0x4500 6070 0x4500 7070 0x4500 8070 0x4500 9070 0x4500 A070	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			

Table 14-297. L3_STCOL_DUMP_ALARM_MODE1 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_ALARM_MODE1	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE1	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 14-298. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE1

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-299. L3_STCOL_DUMP_ALARM_MODE2

Address Offset	See Table 14-235.		
Physical Address	0x4500 1074 0x4500 2074 0x4500 3074 0x4500 4074 0x4500 5074 0x4500 6074 0x4500 7074 0x4500 8074 0x4500 9074 0x4500 A074	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_ALARM_MODE2	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE2	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 14-300. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE2

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-301. L3_STCOL_DUMP_ALARM_MODE3

Address Offset	See Table 14-235.		
Physical Address	0x4500 1078 0x4500 2078 0x4500 3078 0x4500 4078 0x4500 5078 0x4500 6078 0x4500 7078 0x4500 8078 0x4500 9078 0x4500 A078	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_ALARM_MODE3	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE3	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 14-302. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE3

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-303. L3_STCOL_DUMP_ALARM_MODE4

Address Offset	See Table 14-235.		
Physical Address	0x4500 107C 0x4500 207C 0x4500 407C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_ALARM_MODE4	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000

Bits	Field Name	Description	Type	Reset
1:0	DUMP_ALARM_MODE4	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 14-304. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE4

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-305. L3_STCOL_DUMP_ALARM_MODE5

Address Offset	See Table 14-235.		
Physical Address	0x4500 1080 0x4500 2080 0x4500 4080	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL3
Description			
Type	RW		

31–2	1–0
RESERVED	DUMP_ALARM_MODE5

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE5	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 14-306. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE5

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-307. L3_STCOL_DUMP_ALARM_MODE6

Address Offset	See Table 14-235.		
Physical Address	0x4500 1084 0x4500 4084	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL3
Description			
Type	RW		

31–2	1–0
RESERVED	DUMP_ALARM_MODE6

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE6	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 14-308. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE6

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-309. L3_STCOL_DUMP_ALARM_MODE7

Address Offset	See Table 14-235.		
Physical Address	0x4500 1088 0x4500 4088	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL3
Description			
Type	RW		

31–2	1–0
RESERVED	DUMP_ALARM_MODE7

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE7	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: 0x1: 0x3: 0x2:	RW	0x0

Table 14-310. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE7

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-311. L3_STCOL_DUMP_CNT0

Address Offset	See Table 14-235.		
Physical Address	0x4500 108C 0x4500 208C 0x4500 308C 0x4500 408C 0x4500 508C 0x4500 608C 0x4500 708C 0x4500 808C 0x4500 908C 0x4500 A08C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT0																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT0	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 14-312. Register Call Summary for Register L3_STCOL_DUMP_CNT0

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-313. L3_STCOL_DUMP_CNT1

Address Offset	See Table 14-235.		
Physical Address	0x4500 1090 0x4500 2090 0x4500 3090 0x4500 4090 0x4500 5090 0x4500 6090 0x4500 7090 0x4500 8090 0x4500 9090 0x4500 A090	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT1																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT1	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 14-314. Register Call Summary for Register L3_STCOL_DUMP_CNT1

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-315. L3_STCOL_DUMP_CNT2

Address Offset	See Table 14-235.		
Physical Address	0x4500 1094 0x4500 2094 0x4500 3094 0x4500 4094 0x4500 5094 0x4500 6094 0x4500 7094 0x4500 8094 0x4500 9094 0x4500 A094	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT2																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT2	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 14-316. Register Call Summary for Register L3_STCOL_DUMP_CNT2

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-317. L3_STCOL_DUMP_CNT3

Address Offset	See Table 14-235.		
Physical Address	0x4500 1098 0x4500 2098 0x4500 3098 0x4500 4098 0x4500 5098 0x4500 6098 0x4500 7098 0x4500 8098 0x4500 9098 0x4500 A098	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT3																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT3	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 14-318. Register Call Summary for Register L3_STCOL_DUMP_CNT3

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-319. L3_STCOL_DUMP_CNT4

Address Offset	See Table 14-235.		
Physical Address	0x4500 109C 0x4500 209C 0x4500 409C	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL3
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT4																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT4	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 14-320. Register Call Summary for Register L3_STCOL_DUMP_CNT4

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-321. L3_STCOL_DUMP_CNT5

Address Offset	See Table 14-235.		
Physical Address	0x4500 10A0 0x4500 20A0 0x4500 40A0	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL3
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT5																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT5	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 14-322. Register Call Summary for Register L3_STCOL_DUMP_CNT5

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-323. L3_STCOL_DUMP_CNT6

Address Offset	See Table 14-235.		
Physical Address	0x4500 10A4 0x4500 40A4	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL3
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT6																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT6	Dump counter value Type: Status. Reset value: X.	R	0x---- ----

Table 14-324. Register Call Summary for Register L3_STCOL_DUMP_CNT6

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-325. L3_STCOL_DUMP_CNT7

Address Offset	See Table 14-235.		
Physical Address	0x4500 10A8 0x4500 40A8	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL3
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT7																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT7	Dump counter value Type: Status. Reset value: X.	R	0x---- ----

Table 14-326. Register Call Summary for Register L3_STCOL_DUMP_CNT7

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-327. L3_STCOL_FILTER_i_GLOBALEN

Address Offset	See Table 14-235.

Table 14-327. L3_STCOL_FILTER_i_GLOBALEN (continued)

Physical Address	0x4500 10AC + (0x158*i) 0x4500 20AC + (0x158*i) 0x4500 30AC + (0x158*i) 0x4500 40AC + (0x158*i) 0x4500 50AC + (0x158*i) 0x4500 60AC + (0x158*i) 0x4500 70AC + (0x158*i) 0x4500 80AC + (0x158*i) 0x4500 90AC + (0x158*i) 0x4500 A0AC + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–1	0
RESERVED	FILTER_i_GLOBALEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_GLOBALEN	Filter global enable Type: Control. Reset value: 0x0.	RW	0

Table 14-328. Register Call Summary for Register L3_STCOL_FILTER_i_GLOBALEN

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1] [2]

Table 14-329. L3_STCOL_FILTER_i_ADDRMIN

Address Offset	See Table 14-235.		
Physical Address	0x4500 10B0 + (0x158*i) 0x4500 20B0 + (0x158*i) 0x4500 30B0 + (0x158*i) 0x4500 40B0 + (0x158*i) 0x4500 50B0 + (0x158*i) 0x4500 60B0 + (0x158*i) 0x4500 70B0 + (0x158*i) 0x4500 80B0 + (0x158*i) 0x4500 90B0 + (0x158*i) 0x4500 A0B0 + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–23	22–0
RESERVED	FILTER0_ADDRMIN

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0000 0000
22:0	FILTER0_ADDRMIN	Min addr range Type: Control. Reset value: 0x0.	RW	0x00 0000

Table 14-330. Register Call Summary for Register L3_STCOL_FILTER_i_ADDRMIN

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-331. L3_STCOL_FILTER_i_ADDRMAX

Address Offset	See Table 14-235.		
Physical Address	0x4500 10B4 + (0x158*i) 0x4500 20B4 + (0x158*i) 0x4500 30B4 + (0x158*i) 0x4500 40B4 + (0x158*i) 0x4500 50B4 + (0x158*i) 0x4500 60B4 + (0x158*i) 0x4500 70B4 + (0x158*i) 0x4500 80B4 + (0x158*i) 0x4500 90B4 + (0x158*i) 0x4500 A0B4 + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FILTER0_ADDRMAX																						

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0000 0000
22:0	FILTER0_ADDRMAX	Max addr range Type: Control. Reset value: 0x0.	RW	0x00 0000

Table 14-332. Register Call Summary for Register L3_STCOL_FILTER_i_ADDRMAX

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-333. L3_STCOL_FILTER_i_ADDREN

Address Offset	See Table 14-235.		
Physical Address	0x4500 10B8 + (0x158*i) 0x4500 20B8 + (0x158*i) 0x4500 30B8 + (0x158*i) 0x4500 40B8 + (0x158*i) 0x4500 50B8 + (0x158*i) 0x4500 60B8 + (0x158*i) 0x4500 70B8 + (0x158*i) 0x4500 80B8 + (0x158*i) 0x4500 90B8 + (0x158*i) 0x4500 A0B8 + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER0_ADDREN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx

Bits	Field Name	Description	Type	Reset
0	FILTER0_ADDREN	max filtering enable Type: Control. Reset value: 0x0.	RW	0

Table 14-334. Register Call Summary for Register L3_STCOL_FILTER_i_ADDREN

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-335. L3_STCOL_FILTER_i_EN_k

Address Offset	See Table 14-235.		
Physical Address	0x4500 10BC + (0x158*i) + (0x44*k) 0x4500 20BC + (0x158*i) + (0x44*k) 0x4500 30BC + (0x158*i) + (0x44*k) 0x4500 40BC + (0x158*i) + (0x44*k) 0x4500 50BC + (0x158*i) + (0x44*k) 0x4500 60BC + (0x158*i) + (0x44*k) 0x4500 70BC + (0x158*i) + (0x44*k) 0x4500 80BC + (0x158*i) + (0x44*k) 0x4500 90BC + (0x158*i) + (0x44*k) 0x4500 A0BC + (0x158*i) + (0x44*k)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_EN0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_EN0	Enable filter stage 0 Type: Control. Reset value: 0x0.	RW	0

Table 14-336. Register Call Summary for Register L3_STCOL_FILTER_i_EN_k

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-337. L3_STCOL_FILTER_i_MASK_m_MSTADDR

Address Offset	See Table 14-235.

Table 14-337. L3_STCOL_FILTER_i_MASK_m_MSTADDR (continued)

Physical Address	0x4500 10C8 + (0x158*i) + (0x44*m) 0x4500 20C8 + (0x158*i) + (0x44*m) 0x4500 30C8 + (0x158*i) + (0x44*m) 0x4500 40C8 + (0x158*i) + (0x44*m) 0x4500 50C8 + (0x158*i) + (0x44*m) 0x4500 60C8 + (0x158*i) + (0x44*m) 0x4500 70C8 + (0x158*i) + (0x44*m) 0x4500 80C8 + (0x158*i) + (0x44*m) 0x4500 90C8 + (0x158*i) + (0x44*m) 0x4500 A0C8 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FILTER_i_MASK_m_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	FILTER_i_MASK_m_MSTADDR	Mask/Match of MstAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 14-338. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_MSTADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-339. L3_STCOL_FILTER_i_MASK_m_RD

Address Offset	See Table 14-235.		
Physical Address	0x4500 10C0 + (0x158*i) + (0x44*m) 0x4500 20C0 + (0x158*i) + (0x44*m) 0x4500 30C0 + (0x158*i) + (0x44*m) 0x4500 40C0 + (0x158*i) + (0x44*m) 0x4500 50C0 + (0x158*i) + (0x44*m) 0x4500 60C0 + (0x158*i) + (0x44*m) 0x4500 70C0 + (0x158*i) + (0x44*m) 0x4500 80C0 + (0x158*i) + (0x44*m) 0x4500 90C0 + (0x158*i) + (0x44*m) 0x4500 A0C0 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FILTER_i_MASK_m_RD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK_m_RD	Mask/Match of Rd Type: Control. Reset value: 0x0.	RW	0

Table 14-340. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_RD

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-341. L3_STCOL_FILTER_i_MASK_m_WR

Address Offset	See Table 14-235.		
Physical Address	0x4500 10C4 + (0x158*i) + (0x44*m) 0x4500 20C4 + (0x158*i) + (0x44*m) 0x4500 30C4 + (0x158*i) + (0x44*m) 0x4500 40C4 + (0x158*i) + (0x44*m) 0x4500 50C4 + (0x158*i) + (0x44*m) 0x4500 60C4 + (0x158*i) + (0x44*m) 0x4500 70C4 + (0x158*i) + (0x44*m) 0x4500 80C4 + (0x158*i) + (0x44*m) 0x4500 90C4 + (0x158*i) + (0x44*m) 0x4500 A0C4 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MASK_m_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK_m_WR	Mask/Match of Wr Type: Control. Reset value: 0x0.	RW	0

Table 14-342. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_WR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-343. L3_STCOL_FILTER_i_MASK_m_ERR

Address Offset	See Table 14-235.		
Physical Address	0x4500 10D0 + (0x158*i) + (0x44*m) 0x4500 20D0 + (0x158*i) + (0x44*m) 0x4500 30D0 + (0x158*i) + (0x44*m) 0x4500 40D0 + (0x158*i) + (0x44*m) 0x4500 50D0 + (0x158*i) + (0x44*m) 0x4500 60D0 + (0x158*i) + (0x44*m) 0x4500 70D0 + (0x158*i) + (0x44*m) 0x4500 80D0 + (0x158*i) + (0x44*m) 0x4500 90D0 + (0x158*i) + (0x44*m) 0x4500 A0D0 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–1	0
RESERVED	FILTER_i_MASK_m_ERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK_m_ERR	Mask/Match of Err Type: Control. Reset value: 0x0.	RW	0

Table 14-344. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_ERR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-345. L3_STCOL_FILTER_i_MASK_m_USERINFO

Address Offset	See Table 14-235.		
Physical Address	0x4500 10D4 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FILTER_i_MASK_m_USERINFO

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	FILTER_i_MASK_m_USERINFO	Mask/Match of UserInfo Type: Control. Reset value: 0x0.	RW	0x00000

Table 14-346. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_USERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-347. L3_STCOL_FILTER_i_MASK_m_SLVADDR

Address Offset	See Table 14-235.		
Physical Address	0x4500 20CC + (0x158*i) + (0x44*m) 0x4500 30CC + (0x158*i) + (0x44*m) 0x4500 40CC + (0x158*i) + (0x44*m) 0x4500 50CC + (0x158*i) + (0x44*m) 0x4500 60CC + (0x158*i) + (0x44*m) 0x4500 70CC + (0x158*i) + (0x44*m) 0x4500 80CC + (0x158*i) + (0x44*m) 0x4500 90CC + (0x158*i) + (0x44*m) 0x4500 A0CC + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									FILTER_i_MASK_m_SLVADDR						

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000 0000
6:0	FILTER_i_MASK_m_SLVADDR	Mask/Match of SlvAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 14-348. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_SLVADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-349. L3_STCOL_FILTER_i_MASK_m_REQUSERINFO

Address Offset	See Table 14-235.

Table 14-349. L3_STCOL_FILTER_i_MASK_m_REQUSERINFO (continued)

Physical Address	0x4500 20D4 + (0x158*i) + (0x44*m) 0x4500 30D4 + (0x158*i) + (0x44*m) 0x4500 40D4 + (0x158*i) + (0x44*m) 0x4500 50D4 + (0x158*i) + (0x44*m) 0x4500 60D4 + (0x158*i) + (0x44*m) 0x4500 70D4 + (0x158*i) + (0x44*m) 0x4500 80D4 + (0x158*i) + (0x44*m) 0x4500 90D4 + (0x158*i) + (0x44*m) 0x4500 A0D4 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FILTER0_MASK0_REQUSERINFO																											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0000 0000
27:0	FILTER_i_MASK_m_REQUSERINFO	Mask/Match of ReqUserInfo Type: Control. Reset value: 0x0.	RW	0x00

Table 14-350. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_REQUSERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-351. L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO

Address Offset	See Table 14-235.		
Physical Address	0x4500 20D8 + (0x158*i) + (0x44*m) 0x4500 30D8 + (0x158*i) + (0x44*m) 0x4500 40D8 + (0x158*i) + (0x44*m) 0x4500 50D8 + (0x158*i) + (0x44*m) 0x4500 60D8 + (0x158*i) + (0x44*m) 0x4500 70D8 + (0x158*i) + (0x44*m) 0x4500 80D8 + (0x158*i) + (0x44*m) 0x4500 90D8 + (0x158*i) + (0x44*m) 0x4500 A0D8 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FILTER_i_MASK_m_RSPUSERINFO

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	FILTER_i_MASK_m_RSPUSERINFO	Mask/Match of RspUserInfo Type: Control. Reset value: 0x0.	RW	0x0

Table 14-352. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-353. L3_STCOL_FILTER_i_MATCH_m_MSTADDR

Address Offset	See Table 14-235.		
Physical Address	0x4500 10E8 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0
	0x4500 20E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL1
	0x4500 30E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL2
	0x4500 40E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL3
	0x4500 50E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL4
	0x4500 60E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL5
	0x4500 70E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL6
	0x4500 80E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL7
	0x4500 90E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL8
	0x4500 A0E8 + (0x158*i) + (0x44*m)		CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FILTER_i_MATCH_m_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	FILTER_i_MATCH_m_MSTADDR	Mask/Match of MstAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 14-354. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_MSTADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-355. L3_STCOL_FILTER_i_MATCH_m_SLVADDR

Address Offset	See Table 14-235.

Table 14-355. L3_STCOL_FILTER_i_MATCH_m_SLVADDR (continued)

Physical Address	0x4500 20EC + (0x158*i) + (0x44*m) 0x4500 30EC + (0x158*i) + (0x44*m) 0x4500 40EC + (0x158*i) + (0x44*m) 0x4500 50EC + (0x158*i) + (0x44*m) 0x4500 60EC + (0x158*i) + (0x44*m) 0x4500 70EC + (0x158*i) + (0x44*m) 0x4500 80EC + (0x158*i) + (0x44*m) 0x4500 90EC + (0x158*i) + (0x44*m) 0x4500 A0EC + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											FILTER_i_MATCH_m_SLVADDR				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000 0000
4:0	FILTER0_MATCH0_SLVADDR	Mask/Match of SlvAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 14-356. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_SLVADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-357. L3_STCOL_FILTER_i_MATCH_m_RD

Address Offset	See Table 14-235.		
Physical Address	0x4500 10E0 + (0x158*i) + (0x44*m) 0x4500 20E0 + (0x158*i) + (0x44*m) 0x4500 30E0 + (0x158*i) + (0x44*m) 0x4500 40E0 + (0x158*i) + (0x44*m) 0x4500 50E0 + (0x158*i) + (0x44*m) 0x4500 60E0 + (0x158*i) + (0x44*m) 0x4500 70E0 + (0x158*i) + (0x44*m) 0x4500 80E0 + (0x158*i) + (0x44*m) 0x4500 90E0 + (0x158*i) + (0x44*m) 0x4500 A0E0 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FILTER_i_MATCH_m_RD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH_m_RD	Mask/Match of Rd Type: Control. Reset value: 0x0.	RW	0

Table 14-358. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_RD

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-359. L3_STCOL_FILTER_i_MATCH_m_WR

Address Offset	See Table 14-235.		
Physical Address	0x4500 10E4 + (0x158*i) + (0x44*m) 0x4500 20E4 + (0x158*i) + (0x44*m) 0x4500 30E4 + (0x158*i) + (0x44*m) 0x4500 40E4 + (0x158*i) + (0x44*m) 0x4500 50E4 + (0x158*i) + (0x44*m) 0x4500 60E4 + (0x158*i) + (0x44*m) 0x4500 70E4 + (0x158*i) + (0x44*m) 0x4500 80E4 + (0x158*i) + (0x44*m) 0x4500 90E4 + (0x158*i) + (0x44*m) 0x4500 A0E4 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MATCH_m_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000

Bits	Field Name	Description	Type	Reset
0	FILTER_i_MATCH_m_WR	Mask/Match of Wr Type: Control. Reset value: 0x0.	RW	0

Table 14-360. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_WR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-361. L3_STCOL_FILTER_i_MATCH_m_ERR

Address Offset	See Table 14-235.		
Physical Address	0x4500 10F0 + (0x158*i) + (0x44*m) 0x4500 20F0 + (0x158*i) + (0x44*m) 0x4500 30F0 + (0x158*i) + (0x44*m) 0x4500 40F0 + (0x158*i) + (0x44*m) 0x4500 50F0 + (0x158*i) + (0x44*m) 0x4500 60F0 + (0x158*i) + (0x44*m) 0x4500 70F0 + (0x158*i) + (0x44*m) 0x4500 80F0 + (0x158*i) + (0x44*m) 0x4500 90F0 + (0x158*i) + (0x44*m) 0x4500 A0F0 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–1	0
RESERVED	FILTER_i_MATCH_m_ERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH_m_ERR	Mask/Match of Err Type: Control. Reset value: 0x0.	RW	0

Table 14-362. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_ERR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-363. L3_STCOL_FILTER_i_MATCH_m_USERINFO

Address Offset	See Table 14-235.		
Physical Address	0x4500 10F4 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL0
Description			

Table 14-363. L3_STCOL_FILTER_i_MATCH_m_USERINFO (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														FILTER_i_MATCH_m_USERINFO																	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	FILTER_i_MATCH_m_USERINFO	Mask/Match of UserInfo Type: Control. Reset value: 0x0.	RW	0x00000

Table 14-364. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_USERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-365. L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO

Address Offset	See Table 14-235.		
Physical Address	0x4500 20F4 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL1
	0x4500 30F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL2
	0x4500 40F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL3
	0x4500 50F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL4
	0x4500 260F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL5
	0x4500 70F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL6
	0x4500 80F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL7
	0x4500 90F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL8
	0x4500 A0F4 + (0x158*i) + (0x44*m)		CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FILTER0_MASK0_REQUSERINFO																											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0000 0000
27:0	FILTER_i_MASK_m_REQUSERINFO	Mask/Match of ReqUserInfo Type: Control. Reset value: 0x0.	RW	0x00

Table 14-366. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-367. L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO

Address Offset	See Table 14-235.

Table 14-367. L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO (continued)

Physical Address	0x4500 20F8 + (0x158*i) + (0x44*m) 0x4500 30F8 + (0x158*i) + (0x44*m) 0x4500 40F8 + (0x158*i) + (0x44*m) 0x4500 50F8 + (0x158*i) + (0x44*m) 0x4500 60F8 + (0x158*i) + (0x44*m) 0x4500 70F8 + (0x158*i) + (0x44*m) 0x4500 80F8 + (0x158*i) + (0x44*m) 0x4500 90F8 + (0x158*i) + (0x44*m) 0x4500 A0F8 + (0x158*i) + (0x44*m)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31:3	2:0
RESERVED	FILTER_i_MASK_m_RSPUSERINFO

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	FILTER_i_MASK_m_RSPUSERINFO	Mask/Match of RspUserInfo Type: Control. Reset value: 0x0.	RW	0x0

Table 14-368. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-369. L3_STCOL_OP_i_THRESHOLD_MINVAL

Address Offset	See Table 14-235.		
Physical Address	0x4500 11F0 + (0x158*i) 0x4500 21F0 + (0x158*i) 0x4500 31F0 + (0x158*i) 0x4500 41F0 + (0x158*i) 0x4500 51F0 + (0x158*i) 0x4500 61F0 + (0x158*i) 0x4500 71F0 + (0x158*i) 0x4500 81F0 + (0x158*i) 0x4500 91F0 + (0x158*i) 0x4500 A1F0 + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31:13	12:0
RESERVED	OP_i_THRESHOLD_MINVAL

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	OP_i_THRESHOLD_MINVAL	Min value Type: Control. Reset value: 0x0.	RW	0x000

Table 14-370. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MINVAL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-371. L3_STCOL_OP_i_THRESHOLD_MAXVAL

Address Offset	See Table 14-235.		
Physical Address	0x4500 11F4 + (0x158*i) 0x4500 21F4 + (0x158*i) 0x4500 31F4 + (0x158*i) 0x4500 41F4 + (0x158*i) 0x4500 51F4 + (0x158*i) 0x4500 61F4 + (0x158*i) 0x4500 71F4 + (0x158*i) 0x4500 81F4 + (0x158*i) 0x4500 91F4 + (0x158*i) 0x4500 A1F4 + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–13	12–0
RESERVED	OP_i_THRESHOLD_MAXVAL

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	OP_i_THRESHOLD_MAXVAL	Max value Type: Control. Reset value: 0x0.	RW	0x000

Table 14-372. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MAXVAL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-373. L3_STCOL_OP_i_EVTINFOSEL

Address Offset	See Table 14-235.		
Physical Address	0x4500 11F8 + (0x158*i) 0x4500 21F8 + (0x158*i) 0x4500 31F8 + (0x158*i) 0x4500 41F8 + (0x158*i) 0x4500 51F8 + (0x158*i) 0x4500 61F8 + (0x158*i) 0x4500 71F8 + (0x158*i) 0x4500 81F8 + (0x158*i) 0x4500 91F8 + (0x158*i) 0x4500 A1F8 + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31–1	0
RESERVED	OP_i_EVTINFOSEL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000

Bits	Field Name	Description	Type	Reset
1:0	OP_i_EVTINFOSEL	Select event info data to add to counter (len/press or latency) Type: Control. Reset value: 0x0. 0x0: Select len from event info list 0x1: Select pressure if available from event info list 0x2: Select latency if available from event info list	RW	0x0

Table 14-374. Register Call Summary for Register L3_STCOL_OP_i_EVTINFOSEL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

Table 14-375. L3_STCOL_OP_i_SEL

Address Offset	See Table 14-235.		
Physical Address	0x4500 11FC + (0x158*i) 0x4500 21FC + (0x158*i) 0x4500 31FC + (0x158*i) 0x4500 41FC + (0x158*i) 0x4500 51FC + (0x158*i) 0x4500 61FC + (0x158*i) 0x4500 71FC + (0x158*i) 0x4500 81FC + (0x158*i) 0x4500 91FC + (0x158*i) 0x4500 A1FC + (0x158*i)	**Instance**	CLK2_STATCOLL0 CLK2_STATCOLL1 CLK2_STATCOLL2 CLK2_STATCOLL3 CLK2_STATCOLL4 CLK2_STATCOLL5 CLK2_STATCOLL6 CLK2_STATCOLL7 CLK2_STATCOLL8 CLK2_STATCOLL9
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												OP_i_SEL			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:0	OP_i_SEL	Select logical operation Type: Control. Reset value: 0x0. 0x0: Increment counter on each mask/match filter hit 0x1: Increment counter on each min/max level hit 0x2: Add to counter the selected event info value (len/ press or latency) 0x3: increment counter when all filter event hits (And(Fi)) 0x4: Increment counter if any of filter event hits (Or(Fi)) 0x5: Add to counter the number of current request event that hit 0x6: Add to counter the number of current response event that hit 0x7: Add to counter the number of all event that hit 0x8:Increment counter on each selected external event hit	RW	0x0

Table 14-376. Register Call Summary for Register L3_STCOL_OP_i_SEL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0] [1]

14.3 L4 Interconnects

This section details the device L4 interconnects.

14.3.1 L4 Interconnect Overview

The device uses three separate L4 interconnect structures to connect peripheral modules. All L4s handle transfers with peripherals but are located in distinct power domains.

Figure 14-9 is an overview of the L4 interconnects and the peripherals attached to them.

The L4 interconnect is composed of the following interconnects:

- L4_CFG: Includes the majority of the configuration interface for L3_MAIN system modules and peripheral interconnect
- L4 PER: Includes the main peripherals that require sDMA access. L4_PER is further divided into three sub-interconnects: L4_PER1, L4_PER2 and L4_PER3. As shown in Figure 14-9, each of these L4_PERx sub-interconnects is connected to L3_MAIN initiators via three ports:
  - L4_PER1_P1, L4_PER1_P2, L4_PER1_P3
  - L4_PER2_P1, L4_PER2_P2, L4_PER2_P3
  - L4_PER3_P1, L4_PER3_P2, L4_PER3_P3

  L3_MAIN initiators access the L4_PERx peripherals through these ports. All peripherals attached to L4_PER1 are visible from all three L4_PER1 ports; this is also correct for L4_PER2 and L4_PER3. For information on which initiator can access which L4_PERx port, see Figure 14-3.
- L4_WKUP: Includes peripherals attached to the WKUP power domain

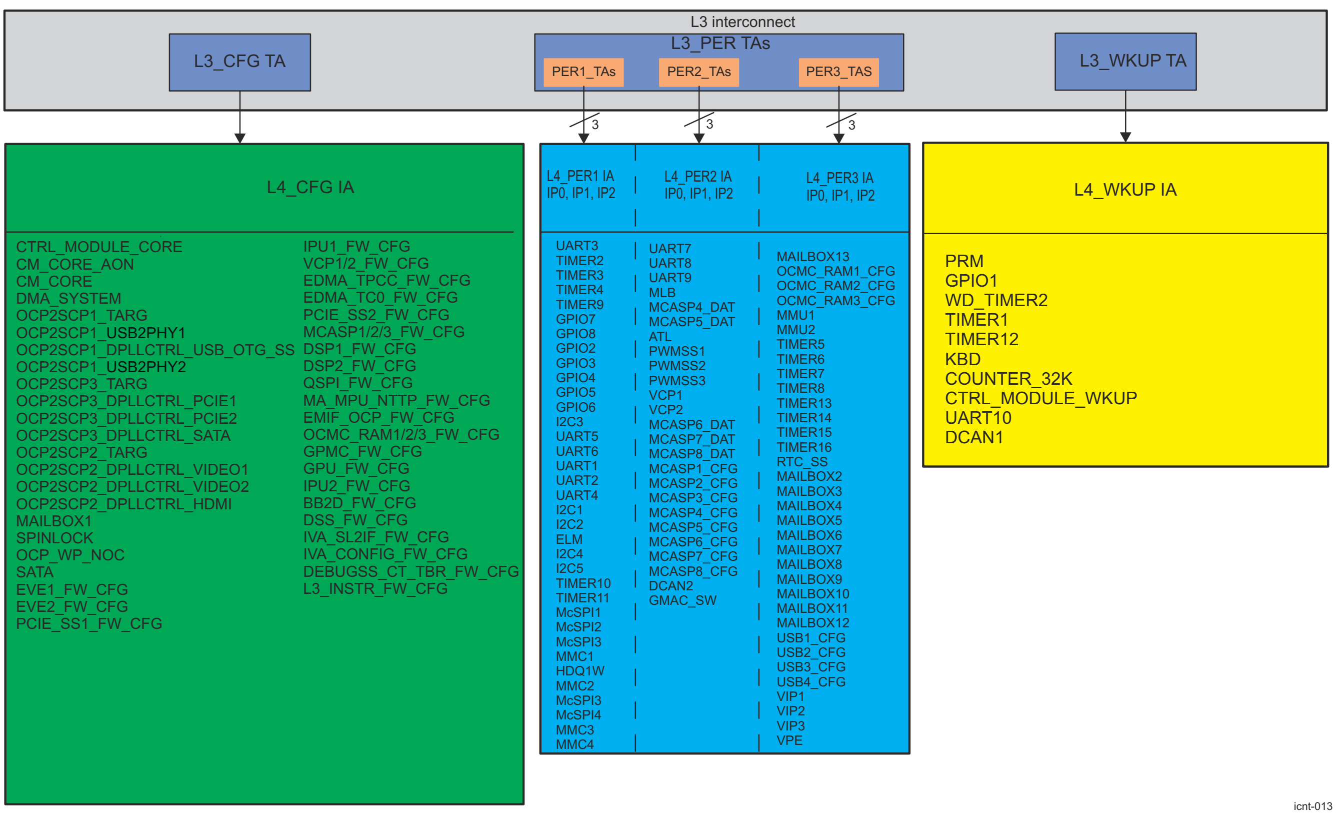


Figure 14-9. L4 Interconnect Overview

The main features of the L4 interconnects are:

- Eleven ports from L3_MAIN interconnect onto 5 parallel L4 interconnects
- From one to three 32-bit initiator ports for each L4 interconnect instance (11 in total)

- 8-, 16-, or 32-bit data, single, or burst transactions
- Little-endian platform
- Non-blocking architecture with fair arbitration between masters
- Target interfaces: Fully synchronous or divided synchronous clock frequencies
- L4_CFG and L4_PER1, L4_PER2, L4_PER3 frequency equals half of L3 frequency
- Protection logic that provides user-configurable access control to targets by each initiator

14.3.2 L4 Interconnect Integration

Table 14-377 and Table 14-378 summarize the integration of the module in the device.

Table 14-377. Integration Attributes

Module Instance	Attributes
	Power Domain
L4_PER1, L4_PER2, L4_PER3	PD_COREAON
L4_CFG	PD_COREAON
L4_WKUP	PD_WKUPAON

Table 14-378. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
L4_PER1	L4_PER1_CLK	L4PER_L3_GICLK	PRCM module	Functional and interface clock
L4_PER2	L4_PER2_CLK	L4PER_L3_GICLK	PRCM module	Functional and interface clock
L4_PER3	L4_PER3_CLK	L4PER_L3_GICLK	PRCM module	Functional and interface clock
L4_CFG	L4_CFG_CLK	L4CFG_L3_GICLK	PRCM module	Functional and interface clock
L4_WKUP	L4_WKUP_CLK	WKUPAON_GICLK	PRCM module	Functional and interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
L4_PER1, L4_PER2, L4_PER3	L4_PER_RST	L4_PER_RST	PRCM module	Reset of L4_PERx interconnect
	L4_PER_RET_RST	L4_PER_PWRON_RET_RST	PRCM module	Reset of L4_PERx interconnect retention registers. For information about retention reset, see Section 3.5 in Chapter 3, *Power, Reset, and Clock Management*.
L4_CFG	L4_CFG_RST	CORE_RST	PRCM module	Reset of L4_CFG interconnect.
	L4_CFG_RET_RST	CORE_PWRON_RET_RST	PRCM module	Reset of L4_CFG interconnect retention registers. For information about retention reset, see Section 3.5 in Chapter 3, *Power, Reset, and Clock Management*.
L4_WKUP	L4_WKUP_RST	WKUPAON_RST	PRCM module	Reset of L4_WKUP interconnect
	L4_WKUP_RET_RST	L4_WKUP_RET_RST	PRCM module	Reset of L4_WKUP interconnect. For information about retention reset, see Section 3.5 in Chapter 3, *Power, Reset, and Clock Management*.

14.3.3 L4 Interconnect Functional Description

14.3.3.1 Module Distribution

IAs and TAs provide the interface to connect the different modules to their associated interconnect.

Table 14-379 through Table 14-388 list all the modules and subsystems with their associated agents. The agents are listed for each L4 interconnect domain.

14.3.3.1.1 L4_PER1 Interconnect Agents

The L4_PER1 interconnect handles transfers only to peripherals in the PER power domain. Table 14-379 lists the L4_PER1 TAs.

Table 14-379. L4_PER1 TAs

Module Target Name	Description
UART3_TARG	Universal Asynchronous Receiver/Transmitter target port
TIMER2_TARG	General Purpose Timer 2 target port
TIMER3_TARG	TIMER3 target port
TIMER4_TARG	TIMER4 target port
TIMER9_TARG	TIMER9 target port
GPIO7_TARG	General-Purpose Interface 7 target port
GPIO8_TARG	GPIO8 target port
GPIO2_TARG	GPIO2 target port
GPIO3_TARG	GPIO3 target port
GPIO4_TARG	GPIO4 target port
GPIO5_TARG	GPIO5 target port
GPIO6_TARG	GPIO6 target port
I2C3_TARG	Inter-Integrated Circuit 3 target port
UART5_TARG	UART5 target port
UART6_TARG	UART6 target port
UART1_TARG	UART1 target port
UART2_TARG	UART2 target port
UART4_TARG	UART4 target port
I2C1_TARG	I2C1 target port
I2C2_TARG	I2C2 target port
ELM_TARG	Error Location Module target port
I2C4_TARG	I2C4 target port
I2C5_TARG	I2C5 target port
TIMER10_TARG	TIMER10 target port
TIMER11_TARG	TIMER11 target port
MCSPI1_TARG	Multi-channel Serial Peripheral Interface 1 target port
MCSPI2_TARG	McSPI2 target port
MMC1_TARG	MMC1 target port
HDQ1W_TARG	HDQ1W target port
MMC2_TARG	MMC2 target port
MMC3_TARG	MMC3 target port
MCSPI3_TARG	McSPI3 target port
MCSPI4_TARG	McSPI4 target port
MMC4_TARG	MMC4 target port

Four ports communicate between the L3_MAIN interconnect and the L4_PER1 interconnect to allow the L3_MAIN initiators to access the L4_PER1 targets. Table 14-380 lists the L4_PER1 initiator TAs.

For the list of initiators authorized to access the L4_PER1, peripherals, see Section 14.2.3.2.2, *Connectivity Matrix*.

Table 14-380. L4_PER1 IAs

Module Iniator Name	Description
L3_MAIN_IP0_INIT	L3 sDMA RD interconnect port
L3_MAIN_IP1_INIT	L3 sDMA WR interconnect port
L3_MAIN_IP2_INIT	L3 MPU susbystem interconnect port

14.3.3.1.2 L4_PER2 Interconnect Agents

The L4_PER2 interconnect handles transfers only to peripherals in the PER power domain. Table 14-381 lists the L4_PER2 TAs.

Table 14-381. L4_PER2 TAs

Module Target Name	Description
UART7_TARG	UART7 target port
UART8_TARG	UART8 target port
UART9_TARG	UART9 target port
MLB_TARG	Media Local Bus target port
MCASP4_DAT_TARG	Multi-Channel Audio Serial Port - DAT target port
MCASP5_DAT_TARG	MCASP5_DAT target port
ATL_TARG	Audio Tracking Logic target port
PWMSS1_TARG	Pulse-Width Modulation Subsystem1 target port
PWMSS2_TARG	PWMSS2 target port
PWMSS3_TARG	PWMSS3 target port
VCP1_CFG_TARG	Viterby Coder/Decoder 1 target port
VCP2_CFG_TARG	Viterby Coder/Decoder 2 target port
MCASP6_DAT_TARG	MCASP6_DAT target port
MCASP7_DAT_TARG	MCASP7_DAT target port
MCSAP8_DAT_TARG	MCASP8_DAT target port
MCASP1_CFG_TARG	MCASP1_CFG target port
MCASP2_CFG_TARG	MCASP2_CFG target port
MCASP3_CFG_TARG	MCASP3_CFG target port
MCASP4_CFG_TARG	MCASP4_CFG target port
MCASP5_CFG_TARG	MCASP5_CFG target port
MCASP6_CFG_TARG	MCASP6_CFG target port
MCASP7_CFG_TARG	MCASP7_CFG target port
MCASP8_CFG_TARG	MCASP8_CFG target port
DCAN2_TARG	DCAN2 target port
GMAC_SW_TARG	Ethernet Controller target port

Three ports communicate between the L3_MAIN interconnect and the L4_PER2 interconnect to allow the L3_MAIN initiators to access the L4_PER2 targets. Table 14-382 lists the L4_PER2 initiator TAs.

For the list of initiators authorized to access the L4_PER2 peripherals, see Section 14.2.3.2.2, *Connectivity Matrix*.

Table 14-382. L4_PER2 IAs

Module Iniator Name	Description
L3_MAIN_IP0_INIT	L3 sDMA RD interconnect port
L3_MAIN_IP1_INIT	L3 sDMA WR interconnect port
L3_MAIN_IP2_INIT	L3 MPU susbystem interconnect port

14.3.3.1.3 L4_PER3 Interconnect Agents

The L4_PER3 interconnect handles transfers only to peripherals in the PER power domain. Table 14-383 lists the L4_PER3 TAs.

Table 14-383. L4_PER3 TAs

Module Target Name	Description
TIMER5_TARG	TIMER5 target port
TIMER6_TARG	TIMER6 target port
TIMER7_TARG	TIMER7 target port
TIMER8_TARG	TIMER8 target port
TIMER13_TARG	TIMER13 target port
TIMER14_TARG	TIMER14 target port
TIMER15_TARG	TIMER15 target port
TIMER16_TARG	TIMER16 target port
VIP1_TARG	Video Input Parser(VIP) 1 target port
VIP2_TARG	VIP2 target port
VIP3_TARG	VIP3 target port
VPE_TARG	VPE target port
RTC_SS_TARG	Real-Time Clock (RTC) target port
MBX2_TARG	Mailbox 2 target port
MBX3_TARG	Mailbox 3 target port
MBX4_TARG	Mailbox 4 target port
MBX5_TARG	Mailbox 5 target port
MBX6_TARG	Mailbox 6 target port
MBX7_TARG	Mailbox 7 target port
MBX8_TARG	Mailbox 8 target port
MBX12_TARG	Mailbox 12 target port
MBX9_TARG	Mailbox 9 target port
MBX10_TARG	Mailbox 10 target port
MBX11_TARG	Mailbox 11 target port
USB4_CFG_TARG	USB4 configuration port
SATA_TARG	SATA target port
USB2_CFG_TARG	USB2 configuration port
OCMC_RAM1_TARG	On-Chip Memory Controller RAM1 target port
USB1_CFG_TARG	USB1 configuration target port
USB3_CFG_TARG	USB3 configuration target port
OCMC_RAM2_TARG	OCMC_RAM2 target port
OCMC_RAM3_TARG	OCMC_RAM3 target port
MMU1_TARG	Memory Management Unit 1 target port
MMU2_TARG	MMU2 target port
MBX13_TARG	Mailbox 13 target port

Three ports communicate between the L3_MAIN interconnect and the L4_PER3 interconnect to allow the L3_MAIN initiators to access the L4_PER3 targets. Table 14-384 lists the L4_PER3 initiator TAs.

For the list of initiators authorized to access the L4_PER3 peripherals, see Section 14.2.3.2.2, *Connectivity Matrix*.

Table 14-384. L4_PER3 IAs

Module Iniator Name	Description
L3_MAIN_IP0_INIT	L3 sDMA RD interconnect port
L3_MAIN_IP1_INIT	L3 sDMA WR interconnect port
L3_MAIN_IP2_INIT	L3 MPU susbystem interconnect port

14.3.3.1.4 L4_CFG Interconnect Agents

The L4_CFG interconnect handles only transfers to peripherals in the CORE power domain. Table 14-385 lists the TAs.

Table 14-385. L4_CFG TAs

Module Target Name	Description
CTRL_MODULE_CORE_TARG	Control Module core target
CM_CORE_AON_TARG	CORE_AON module
CM_CORE_TARG	CM_CORE module
DMA_SYSTEM_TARG	System DMA
SCP1_TARG	SCP1 module
SCP2_TARG	SCP2 module
MAILBOX_TARG	Mailbox module
SPINLOCK_TARG	Spinlock module
OCP_WP_NOC_TARG	OCP watchpoint module
SATA_TARG	SATA controller module
EVE1_FW_CFG_TARG	Embedded Vision Engine firewall
EVE2_FW_CFG_TARG	EVE2 firewall
IPU1_FW_CFG_TARG	Image Processing Unit (IPU) 1 firewall
IPU2_FW_CFG_TARG	IPU 2 firewall
VCP1_FW_CFG_TARG	Viterby Coder/Decoder 1 firewall
VCP2_FW_CFG_TARG	VCP2 module firewall
TPCC_FW_CFG_TARG	EDMA Channel Controller firewall
TPTC_FW_CFG_TARG	EDMA Transfer Controller firewall
PCIESS1_FW_CFG_TARG	PCIE1 firewall
MCASP1_FW_CFG_TARG	MCASP1 firewall
SCP3_TARG	SCP3 module
OCP2SCP1_USB2PHY1	OCP2SCP1_USB2PHY1 core target port
OCP2SCP1_DPLLCTRL_USB_OTG_SS_TARG	USB OTG Subsystem DPLLCTRL target port
OCP2SCP3_USB2PHY2	USB2PHY2 target port
OCP2SCP3_DPLLCTRL_PCIE1_TARG	PCIE1 DPLL CTRL target port
OCP2SCP3_DPLLCTRL_PCIE2_TARG	PCIE2 DPLL CTRL target port
OCP2SCP3_DPLLCTRL_SATA_TARG	SATA DPLL CTRL target port
OCP2SCP2_DPLLCTRL_VIDEO1_TARG	VIDEO1 DPLL CTRL target port
OCP2SCP2_DPLLCTRL_VIDEO2_TARG	VIDEO2 DPLL CTRL target port
OCP2SCP2_DPLLCTRL_HDMI_TARG	HDMI DPLL CTRL target port
DSP1_SDMA_FW_CFG_TARG	DSP1 firewall
DSP2_SDMA_FW_CFG_TARG	DSP2 firewall

Table 14-385. L4_CFG TAs (continued)

Module Target Name	Description
MA_MPU_NTTP_FW_CFG_TARG	MA_MPU firewall
EMIF_OCP_FW_CFG_TARG	EMIF firewall
OCMC_RAM2_FW_CFG_TARG	OCMC_RAM2 firewall
GPMC_FW_CFG_TARG	GPMC firewall
OCMC_RAM1_FW_CFG_TARG	OCMC_RAM1 firewall
GPU_FW_CFG_TARG	GPU firewall
OCMC_RAM3_FW_CFG_TARG	OCMC_RAM3 firewall
DSS_FW_CFG_TARG	DSS firewall
IVA_SL2IF_FW_CFG_TARG	IVA SL2IF firewall
IVA_CONFIG_FW_CFG_TARG	IVA Config firewall
DEBUGSS_CT_TBR_FW_CFG_TARG	Debug subsystem firewall
L3_INSTR_FW_CFG_TARG	L3 Instrumentation firewall
MCASP2_FW_CFG_TARG	MCASP2 firewall
QSPI_FW_CFG_TARG	QSPI firewall
MCASP3_FW_CFG_TARG	MCASP3 firewall
PCIESS2_FW_CFG_TARG	PCIE2 firewall

A unique port, L3_MAIN_INIT, communicates between the L3 interconnect and the L4_CFG interconnect to allow the L3 initiators to access the L4_CFG targets (see Table 14-386).

For the list of initiators authorized to access the L4_CFG peripherals, see Section 14.2.3.2.2, *Connectivity Matrix*.

Table 14-386. L4_CFG IAs

Module Iniator Name	Description
L3_MAIN_IP0	L3 interconnect port

14.3.3.1.5 L4_WKUP Interconnect Agents

The L4-WKUP interconnect handles transfers only to peripherals in the WKUP power domain. Table 14-387 lists the TAs. Table 14-388 lists the L4 WKUP IAs.

Table 14-387. L4_WKUP TAs

Module Target Name	Description
PRM_TARG	PRM module
GPIO1_TARG	GPIO1 module
WD_TIMER2_TARG	Watchdog Timer 2
TIMER1_TARG	Timer 1
TIMER12_TARG	Timer 12
KBD_TARG	Keyboard controller
COUNTER_32K_TARG	Counter 32k timer
CTRL_MODULE_WKUP_TARG	Control Module Wakeup module
UART10_TARG	UART10 module
DCAN1_TARG	CAN1 module

Table 14-388. L4_WKUP IAs

Module Iniator Name	Description
L3_MAIN_IP0	L3 interconnect port

14.3.3.2 Power Management

As part of the system-wide power-management scheme, the L4 interconnects go into IDLE state after receiving a request from the PRCM module after all commands are serviced. This function is handled by hardware. For more information, see Chapter 3, *Power, Reset, and Clock Management*.

To reduce power consumption, each L4 interconnect automatically performs internal clock autogating. This is managed by hardware; no software configurations or settings are required.

L4_CFG, L4_PER1, L4_PER2, L4_PER3, and L4_WKUP are located in the always-on power domain and no retention is needed for these L4 interconnects.

14.3.3.3 L4 Firewalls

Figure 14-10 is an internal view of the L4 interconnects in the overall interconnect. This architecture, with one initiator subsystem centralizing all initiator master requests and distributing them to all target modules (peripherals), enables the L4 interconnect firewall functions to be centralized at the L4 initiator subsystem level. The L4 firewall filters the accesses based on the configurable protection groups defined in the L4 address protection (AP) registers. Each module or TA is assigned to a protection group. The configuration is also defined in the L4 AP and is programmable on a module-per-module basis.

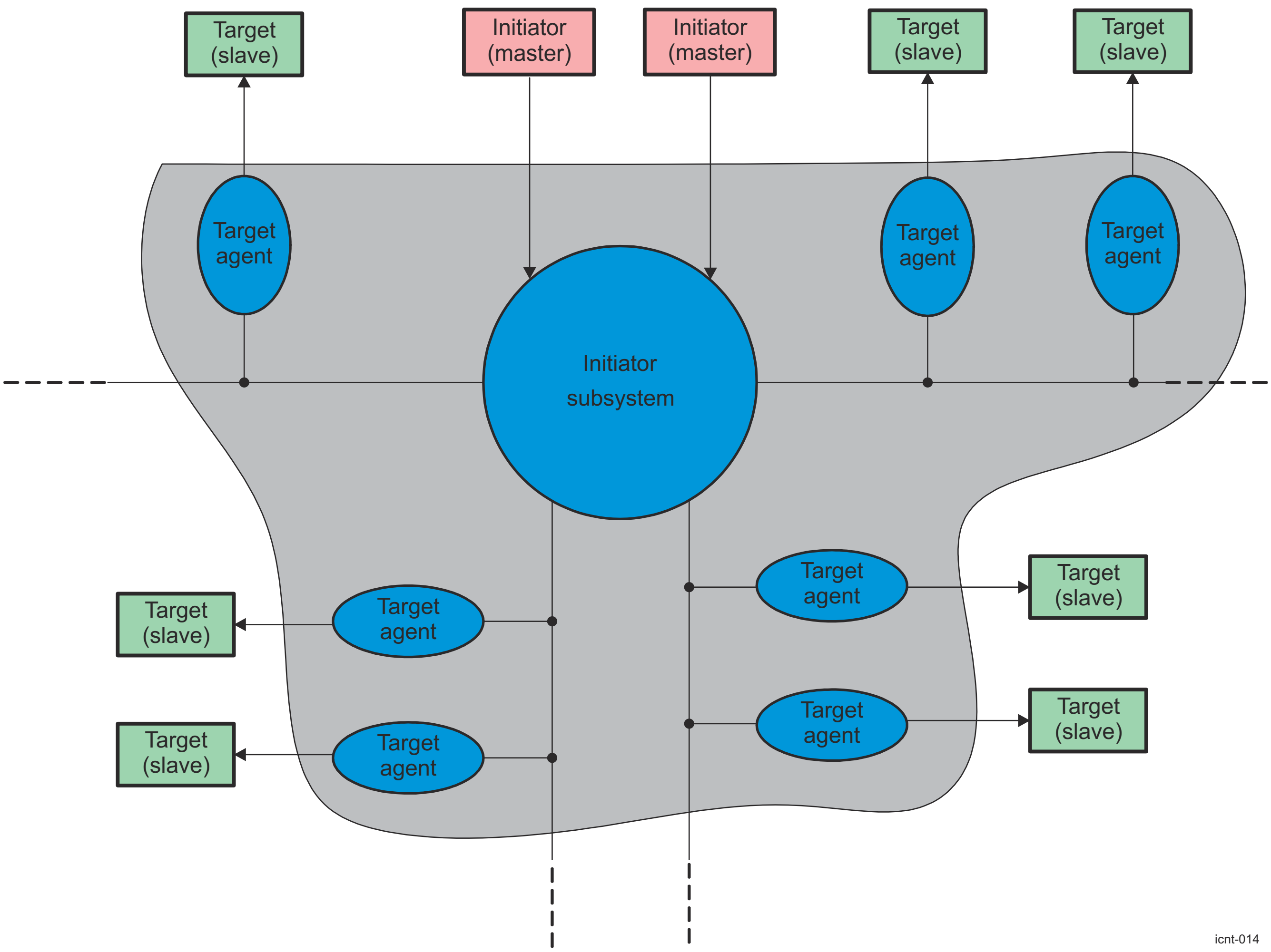


Figure 14-10. L4 Initiator-Target Connectivity

Note

As Figure 14-10 shows, targets are attached to branches. Branches do not impact the function of the L4 interconnect but are present to simplify timing closure and reduce active power consumption.

Because of the large address spaces and the number of peripherals connected to an L4 interconnect, two parameters are used to set up access permission:

- Programmable groups for initiators:
  – Eight protection groups for the L4 interconnect
- Segments divided into regions
  – 85 regions for the L4_PER1 interconnect
  – 63 regions for the L4_PER2 interconnect
  – 97 regions for the L4_PER3 interconnect
  – 129 regions for the L4_CFG interconnect
  – 44 regions for the L4_WKUP interconnect

Protection group members are TAs with the same protection settings. A region is programmed to allow access to a unique selectable protection group. For better protection, different regions are grouped into protection group regions and associated with a protection group member.

14.3.3.3.1 Protection Group

A protection group is defined by its initiators (or members) and MReqInfo is allowed. Two registers define these two settings:

- The 64-bit CONNID_BIT_VECTOR field L4_AP_PROT_GROUP_MEMBERS_k_L and L4_AP_PROT_GROUP_MEMBERS_k_H registers define which initiator belongs to a group. A protection group is accessible by an initiator when the bit position corresponding to its ConnID is set to 1 in the CONNID_BIT_VECTOR field. Table 14-389 lists all the ConnIDs available at the L4 levels.
- The ENABLE field L4_AP_PROT_GROUP_ROLES_k_L register lists all possible MReqInfo combinations associated with the L4_AP_PROT_GROUP_MEMBERS register. Setting a Req bit in this register determines the initiators type of access. For more information, see Section 14.2.3.7.3, *L3_MAIN Firewall Functionality*. Two MReqInfos are used in L4 interconnects: MReqDebug and MReqSupervisor.

Note

Permissions are identical for read and write accesses in L4 interconnect.

k indicates the protection group number.

L indicates the region number.

Table 14-389. L4 ConnID Definition

ConnID	Initiator
0	Cortex-A15 MPU subsystem
1	Debug subsystem
2	DSP1 subsystem (CFG, EDMA, MDMA), DSP2 (EDMA)
3	IVAHD, DSP2(CFG, MDMA)
4	EVE1, EVE2
5	Reserved
6	IPU1/2; SYSTEM_DMA
7	EDMA
8	DSS, MLB, MMU1, MMU2, PCIE1 and PCIE2
9	VIP1, VIP2, VIP3, VPE

Table 14-389. L4 ConnID Definition (continued)

ConnID	Initiator
A	MMC1, MMC2, GPU, BB2D, GMAC
B	USB1, USB2, USB3, USB4
C	SATA
D	Reserved
E	Reserved
F	Resered

Figure 14-11 is an example of CONNID_BIT_VECTOR.

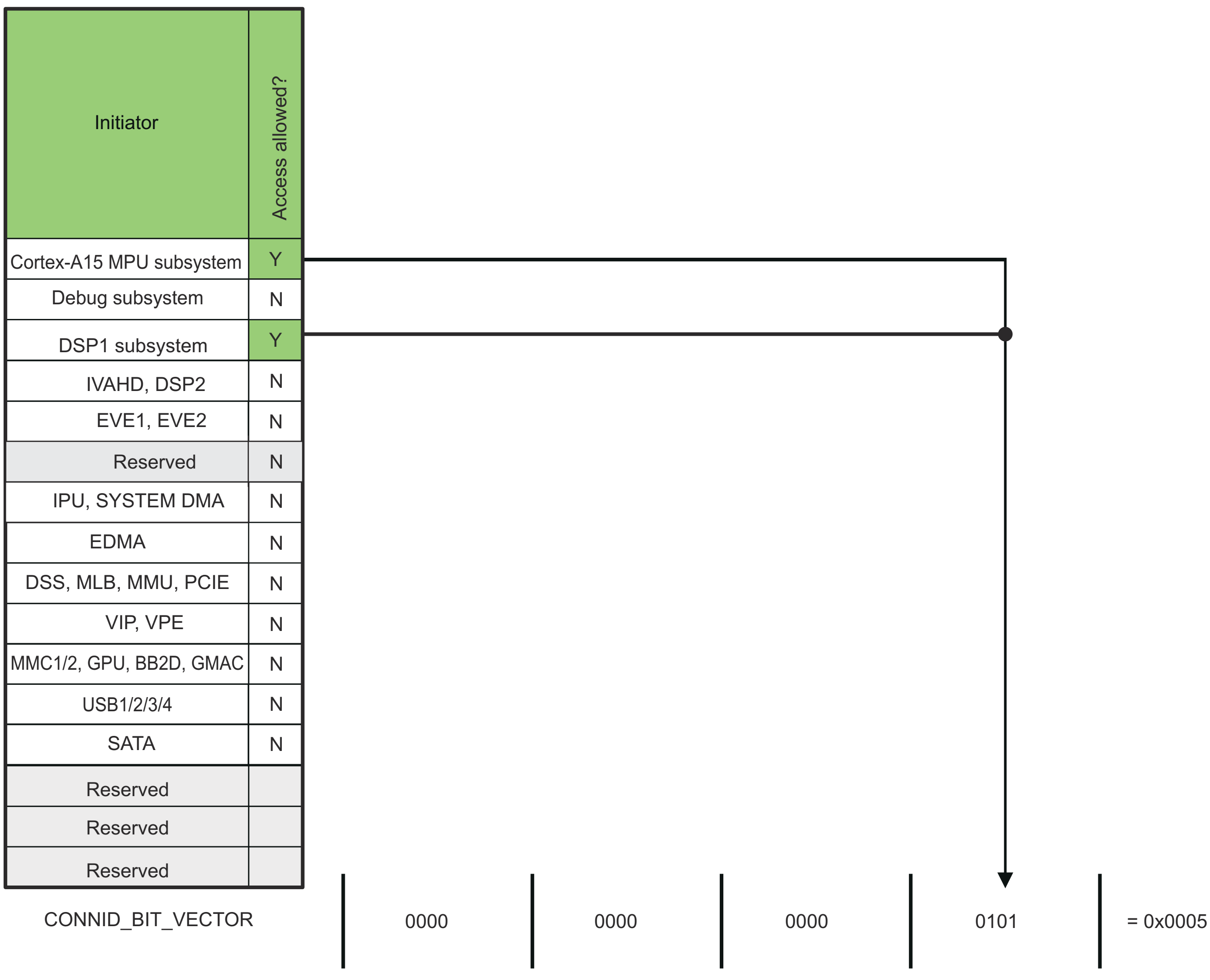


Figure 14-11. Example of CONNID_BIT_VECTOR L4_AP_PROT_GROUP_MEMBERS_k

Setting bits 0 and 2 in the PROT_GROUP_MEMBERS_L register defines a group initiator that can access targets in protection group 1, and includes the following:

- Cortex-A15 MPU subsystem
- DSP subsystem

Protection group 1 can be applied to multiple protection regions with no limitation. Each protection region that is configured with protection group 1 enables permission access only to the two initiators.

The L4_AP_REGION_i_H PROT_GROUP_ID field determines the region to which the protection group member is attached.

The values of some CONNID_BIT_VECTOR and ENABLE fields are exported by the system control module (SCM) at reset or are user writable (see Figure 14-11 for more information).

Table 14-390 and Table 14-393 list the default configuration of the various groups for each L4 interconnect. For each group, some modules or regions are associated with it using default initiator members.

Table 14-390. L4_PER1 Firewall Default Configuration

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 0 AP registers	L4_PER1_AP registers	L4_AP_PROT_GROUP_ROLES_0_L	No	EXPORTED
		L4_AP_PROT_GROUP_ROLES_0_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_L	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_H	No	EXPORTED
Group 1	Reserved	L4_AP_PROT_GROUP_ROLES_1_L	No	EXPORTED
		L4_AP_PROT_GROUP_ROLES_1_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_1_L	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_1_H	No	EXPORTED
Group 2–7	PER1 peripherals	L4_AP_PROT_GROUP_ROLES_k_L L4_AP_PROT_GROUP_ROLES_k_H where k = 2 to 7	Yes	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_k_L L4_AP_PROT_GROUP_MEMBERS_k_H where k = 2 to 7	Yes	0xFFFF (all)

Table 14-391. L4_PER2 Firewall Default Configuration

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 0 AP registers	L4_PER2_AP registers	L4_AP_PROT_GROUP_ROLES_0_L	No	EXPORTED
		L4_AP_PROT_GROUP_ROLES_0_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_L	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_H	No	EXPORTED

Table 14-391. L4_PER2 Firewall Default Configuration (continued)

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 1 - 7 L4_PER2 peripherals	PER2 peripherals	L4_AP_PROT_GROUP_ROLES_k_L L4_AP_PROT_GROUP_ROLES_k_H where k = 1 to 7	Yes	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_k_L L4_AP_PROT_GROUP_MEMBERS_k_H where k = 1 to 7	Yes	0xFFFF (all)

Table 14-392. L4_PER3 Firewall Default Configuration

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 0 AP registers	L4_PER_AP registers	L4_AP_PROT_GROUP_ROLES_0_L	No	EXPORTED
		L4_AP_PROT_GROUP_ROLES_0_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_L	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_H	No	EXPORTED
Group 1 - 7 L4_PER3 peripherals	PER3 peripherals	L4_AP_PROT_GROUP_ROLES_k_L L4_AP_PROT_GROUP_ROLES_k_H where k = 1 to 7	Yes	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_k_L L4_AP_PROT_GROUP_MEMBERS_k_H where k = 1 to 7	Yes	0xFFFF (all)

Table 14-393. L4_CFG Firewall Default Configuration

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 0 AP registers	AP registers	L4_AP_PROT_GROUP_ROLES_0_L L4_AP_PROT_GROUP_ROLES_0_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_L L4_AP_PROT_GROUP_MEMBERS_0_H	No	EXPORTED
Group 1	L3 firewall registers	L4_AP_PROT_GROUP_ROLES_1_L L4_AP_PROT_GROUP_ROLES_1_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_1_L L4_AP_PROT_GROUP_MEMBERS_1_H	No	EXPORTED

Table 14-393. L4_CFG Firewall Default Configuration (continued)

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 5 Free	CPFROM	L4_AP_PROT_GROUP_ROLES_5_L L4_AP_PROT_GROUP_ROLES_5_H	Yes	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_5_L L4_AP_PROT_GROUP_MEMBERS_5_H	Yes	EXPORTED
Group 2, 3, 4 Free	No modules attached	L4_AP_PROT_GROUP_ROLES_k_L L4_AP_PROT_GROUP_ROLES_k_L where k = 2, 3, 4 and 6	Yes	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_k_L L4_AP_PROT_GROUP_MEMBERS_k_H where k = 2, 3 and 4	Yes	0xFFFF (all)
Group 6, 7 Other modules	Other L4_CFG modules	L4_AP_PROT_GROUP_ROLES_k_L L4_AP_PROT_GROUP_ROLES_k_H where k = 6 and 7	Yes	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_k_L L4_AP_PROT_GROUP_MEMBERS_k_H where k = 6 and 7	Yes	0xFFFF (all)

Table 14-394. L4_WKUP Firewall Default Configuration

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 0 AP registers	AP registers	L4_AP_PROT_GROUP_ROLES_0_L L4_AP_PROT_GROUP_ROLES_0_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_0_L L4_AP_PROT_GROUP_MEMBERS_0_H	No	EXPORTED
Group 1	Reserved	L4_AP_PROT_GROUP_ROLES_1_L L4_AP_PROT_GROUP_ROLES_1_H	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_1_L L4_AP_PROT_GROUP_MEMBERS_1_H	No	EXPORTED
Group 2	No Modules Attached	L4_AP_PROT_GROUP_ROLES_2_L L4_AP_PROT_GROUP_ROLES_2_H	Yes	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_2_L L4_AP_PROT_GROUP_MEMBERS_2_H	Yes	0xFFFF (all)

Table 14-394. L4_WKUP Firewall Default Configuration (continued)

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 3 Free	Reserved	L4_AP_PROT_GROUP_ROLES_3_L L4_AP_PROT_GROUP_ROLES_3_L	No	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_3_L L4_AP_PROT_GROUP_MEMBERS_3_H	No	0xFFFF (all)
Group 4 Free	Reserved	L4_AP_PROT_GROUP_ROLES_4_L L4_AP_PROT_GROUP_ROLES_4_L	No	EXPORTED
		L4_AP_PROT_GROUP_MEMBERS_4_L L4_AP_PROT_GROUP_MEMBERS_4_H	No	EXPORTED
Group 5-7 Other modules	Other L4_WKUP modules	L4_AP_PROT_GROUP_ROLES_k_L L4_AP_PROT_GROUP_ROLES_k_H where k = 5, 6, 7	Yes	0xFFFF FFFF FFFF FFFF (all)
		L4_AP_PROT_GROUP_MEMBERS_k_L L4_AP_PROT_GROUP_MEMBERS_k_H where k = 5, 6, 7	Yes	0xFFFF (all)

Note

For EXPORTED default values see CONTROL_SEC_ LOAD_FW_EXPORTED_VALUE register in Chapter 18, *Control Module*

14.3.3.3.2 Segments and Regions

The protection mechanism for L4 interconnects is based on a hierarchical segmentation, as shown in Figure 14-12. By default, some regions are attached to specific protection group members. This specificity lets users set up the permission access to certain types of modules requiring the same access protection without managing region allocation.

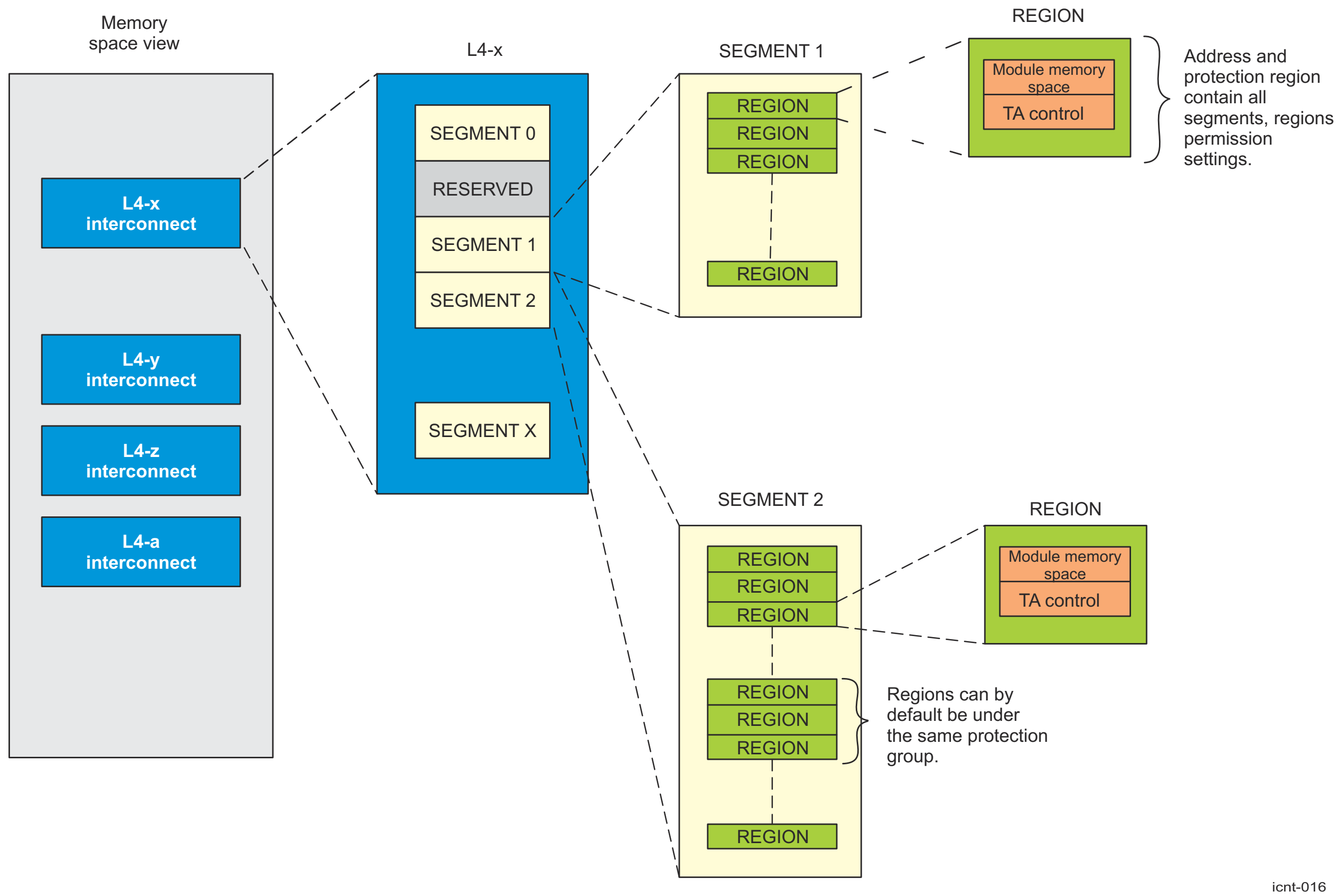


Figure 14-12. L4 Segmentation

All interconnect address spaces are covered by regions. Table 14-395 through Table 14-399 list the module mapping with their addresses, region numbers, and default protection group allocated to them.

Note

Module refers to the configuration registers of the module.

TA (Target Agent) refers to the interconnect configuration registers of the TA associated with the module.

Table 14-395. Region Allocations for L4_PER1 Interconnect

Module	Region	Description
L4_PER1_CONFIG	0	Address Protection
	1	L3_MAIN_IP0 initiator port
	2	Link Agent
UART3_TARG	3	Module
	4	TA
TIMER2_TARG	5	Module
	6	TA
TIMER3_TARG	7	Module
	8	TA

Table 14-395. Region Allocations for L4_PER1 Interconnect (continued)

Module	Region	Description
TIMER4_TARG	9	Module
	10	TA
TIMER9_TARG	11	Module
	12	TA
GPIO2_TARG	13	Module
	14	TA
GPIO3_TARG	15	Module
	16	TA
GPIO4_TARG	17	Module
	18	TA
GPIO5_TARG	19	Module
	20	TA
GPIO6_TARG	21	Module
	22	TA
I2C3	23	Module
UART1_TARG	24	Module
	25	TA
UART2_TARG	26	Module
	27	TA
UART4_TARG	28	Module
	29	TA
I2C1_TARG	30	Module
	31	TA
I2C2_TARG	32	Module
	33	TA
I2C3_TARG	34	TA
GPIO8_TARG	35	Module
	36	TA
HDQ1W_TARG	37	Module
	38	TA
ELM_TARG	39	Module
	40	TA
TIMER10_TARG	41	Module
	42	TA
TIMER11_TARG	43	Module
	44	TA
GPIO7_TARG	45	Module
	46	TA
MCSPI1_TARG	47	Module
	48	TA
MCSPI2_TARG	49	Module
	50	TA
MMC1_TARG	51	Module
	52	TA

Table 14-395. Region Allocations for L4_PER1 Interconnect (continued)

Module	Region	Description
UART6_TARG	53	Module
	54	TA
MMC3_TARG	61	Module
	62	TA
UART5_TARG	63	Module
	64	TA
MMC2_TARG	65	Module
	66	TA
MCSPI3_TARG	67	Module
	68	TA
MCSPI4_TARG	69	Module
	70	TA
MMC4_TARG	71	Module
	72	TA
L4_PER1 CONFIG	77	L3_MAIN_IP1
	78	L3_MAIN_IP2
I2C4_TARG	81	Module
	82	TA
I2C5_TARG	83	Module
	84	TA

Table 14-396. Region Allocations for L4_PER2 Interconnect

Module	Region	Description
L4_PER2_CONFIG	0	Address Protection
	1	L3_MAIN_IP0 initiator port
	2	Link Agent
GMAC_TARG	3	Module
L4_PER2_CONFIG	4	L3_MAIN_IP1 initiator port
	5	L3_MAIN_IP2 initiator port
GMAC_TARG	6	TA
MLB_TARG	7	Module
	8	TA
MCASP1_CFG_TARG	9	Module
	10	TA
MCASP2_CFG_TARG	11	Module
	12	TA
MCASP3_CFG_TARG	13	Module
	14	TA
MCASP4_CFG_TARG	15	Module
	16	TA
MCASP4_DAT_TARG	17	Module
	18	TA
MCASP5_CFG_TARG	19	Module
	20	TA

Table 14-396. Region Allocations for L4_PER2 Interconnect (continued)

Module	Region	Description
MCASP5_DAT_TARG	21	Module
	22	TA
PWM1_TARG	25	Module
	26	TA
PWM2_TARG	27	Module
	28	TA
PWM3_TARG	29	Module
	30	TA
DCAN2_TARG	31	Module
	32	TA
MCASP6_CFG_TARG	35	Module
	36	TA
MCASP7_DAT_TARG	37	Module
	38	TA
MCASP7_CFG_TARG	39	Module
	40	TA
MCASP8_DAT_TARG	41	Module
	42	TA
MCASP8_CFG_TARG	43	Module
	44	TA
MCASP6_DAT_TARG	45	Module
	46	TA
UART7_TARG	47	Module
	48	TA
UART8_TARG	49	Module
	50	TA
UART9_TARG	51	Module
	52	TA
VCP1_CFG_TARG	53	Module
	54	TA
VCP2_CFG_TARG	55	Module
	56	TA

Table 14-397. Region Allocations for L4_PER3 Interconnect

Module	Region	Description
L4_PER3_CONFIG	0	Address Protection
	1	Link Agent
	2	L3_MAIN_IP0 initiator port
	3	L3_MAIN_IP1 initiator port
	4	L3_MAIN_IP2 initiator port
TIMER5_TARG	5	Module
	6	TA
TIMER6_TARG	7	Module
	8	TA

Table 14-397. Region Allocations for L4_PER3 Interconnect (continued)

Module	Region	Description
TIMER7_TARG	9	Module
	10	TA
TIMER8_TARG	11	Module
	12	TA
TIMER13_TARG	13	Module
	14	TA
TIMER14_TARG	15	Module
	16	TA
TIMER15_TARG	17	Module
	18	TA
TIMER16_TARG	19	Module
	20	TA
VIP1_TARG	21	Module
	22	TA
VIP2_TARG	23	Module
	24	TA
VIP3_TARG	25	Module
	26	TA
VPE_TARG	27	Module
	28	TA
RTC_TARG	29	Module
	30	TA
MBX2_TARG	33	Module
	34	TA
MBX3_TARG	35	Module
	36	TA
MBX4_TARG	37	Module
	38	TA
MBX5_TARG	39	Module
	40	TA
MBX6_TARG	41	Module
	42	TA
MBX7_TARG	43	Module
	44	TA
MBX8_TARG	45	Module
	46	TA
MBX12_TARG	67	Module
	68	TA
MBX9_TARG	69	Module
	70	TA
MBX10_TARG	71	Module
	72	TA
MBX11_TARG	73	Module
	74	TA

Table 14-397. Region Allocations for L4_PER3 Interconnect (continued)

Module	Region	Description
USB4_CFG_TARG	75	Module
	76	TA
USB2_CFG_TARG	79	Module
	80	TA
OCMC_RAM1_TARG	81	Module
	82	TA
USB1_CFG_TARG	83	Module
	84	TA
USB3_CFG_TARG	85	Module
	86	TA
OCMC_RAM3_TARG	87	Module
	88	TA
MMU1_TARG	91	Module
	92	TA
MMU2_TARG	93	Module
	94	TA
MBX13_TARG	95	Module
	96	TA

Table 14-398. Region Allocations for L4_CFG Interconnect

Module	Region	Description
L4_CFG Configuration	0	AP
	1	LA
	2	IP0
CTRL_MODULE_CORE_TARG	3	Module
	4	TA
CM_CORE_AON_TARG	5	Module
	6	TA
CM_CORE_TARG	7	Module
	8	TA
DMA_SYSTEM_TARG	9	Module
	10	TA
SCP1_TARG	13	Module
	14	TA
SCP2_TARG	15	Module
	16	TA
MAILBOX_TARG	23	Module
	24	TA
SPINLOCK_TARG	25	Module
	26	TA
OCP_WP_NOC_TARG	27	Module
	28	TA
SATA_TARG	31	Module
	32	TA

Table 14-398. Region Allocations for L4_CFG Interconnect (continued)

Module	Region	Description
EVE1_FW_CFG_TARG	33	Module
	34	TA
EVE2_FW_CFG_TARG	35	Module
	36	TA
IPU1_FW_CFG_TARG	41	Module
	42	TA
IPU2_FW_CFG_TARG	43	Module
	44	TA
VCP1_FW_CFG_TARG	45	Module
	46	TA
VCP2_FW_CFG_TARG	47	Module
	48	TA
TPCC_FW_CFG_TARG	49	Module
	50	TA
TPTC_FW_CFG_TARG	51	Module
	52	TA
PCIESS1_FW_CFG_TARG	53	Module
	54	TA
MCASP1_FW_CFG_TARG	55	Module
	56	TA
SCP3_TARG	59	Module
	60	TA
DSP1_SDMA_FW_CFG_TARG	61	Module
	62	TA
DSP2_SDMA_FW_CFG_TARG	63	Module
	64	TA
MA_MPU_NTTP_FW_CFG_TARG	79	Module
	80	TA
EMIF_OCP_FW_CFG_TARG	81	Module
	82	TA
OCMC_RAM2_FW_CFG_TARG	83	Module
	84	TA
GPMC_FW_CFG_TARG	85	Module
	86	TA
OCMC_RAM1_FW_CFG_TARG	87	Module
	88	TA
GPU_FW_CFG_TARG	89	Module
	90	TA
OCMC_RAM3_FW_CFG	91	Module
	92	TA
DSS_FW_CFG_TARG	93	Module
	94	TA
IVA_SL2IF_FW_CFG_TARG	95	Module
	96	TA

Table 14-398. Region Allocations for L4_CFG Interconnect (continued)

Module	Region	Description
IVA_CONFIG_FW_CFG_TARG	97	Module
	98	TA
DEBUGSS_CT_TBR_FW_CFG_TARG	99	Module
	100	TA
L3_INSTR_FW_CFG_TARG	101	Module
	102	TA
MCASP2_FW_CFG_TARG	103	Module
	104	TA
QSPI_FW_CFG_TARG	105	Module
	106	TA
MCASP3_FW_CFG_TARG	107	Module
	108	TA
PCIESS2_FW_CFG_TARG	125	Module
	126	TA

Table 14-399. Region Allocations for L4_WKUP Interconnect

Module	Region	Description
L4 WKUP	0	AP
	1	IP0
	2	LA
PRM_TARG	3	Module
	4	TA
GPIO1_TARG	5	Module
	6	TA
WD_TIMER2_TARG	7	Module
	8	TA
TIMER1_TARG	9	Module
	10	TA
KBD_TARG	11	Module
	12	TA
COUNTER_32K	15	Module
	16	TA
CTRL_MODULE_WKUP	17	Module
	18	TA
TIMER12	19	Module
	20	TA
UART10_TARG	28	Module
	29	TA
DCAN1_TARG	30	Module
	31	TA

14.3.3.3.3 L4 Firewall Address and Protection Register Settings

Table 14-400 lists the settings of the AP registers relative to an L4 interconnect firewall. These values are computed based on the physical implementation of each L4 interconnect.

Table 14-400. L4 Firewall Register Description Overview

Register Type	Register Name	Bits	Field	Description
Segment	L4_AP_SEGMENT_i_L [(2)]	31:0	BASE	Segment base address
	L4_AP_SEGMENT_i_H [(2)]	5:0	SIZE	Segment size equals to 2^{SIZE}
Protection groups	L4_AP_PROT_GROUP_MEMBERS_k_L [(3)]	15:0	CONNID_BIT_VECTOR	For L4ConnID, see Table 14-389.
	L4_AP_PROT_GROUP_ROLES_k_L [(3)] L4_AP_PROT_GROUP_ROLES_k_H	31:0	ENABLE	See for REQ_INFO description. [(1)]
Region setting	L4_AP_REGION_l_L [(4)]	20:0	BASE	Defines the base address of region with respect to its segment base address
	L4_AP_REGION_l_H [(4)]	31:28	MADDRSPACE	Target interconnect MAddrSpace
		26:24	SEGMENT_ID	Segment ID number of the region
		22:20	PROT_GROUP_ID	Protection group member attached to the region
		18:17	BYTE_DATA_WIDTH_EXP	Determines the number of bytes in an access
		14:8	PHY_TARGET_ID	Physical target ID
		6:1	SIZE	Size of the region equals to 2^{SIZE}
		0	ENABLE	Enables the region protection

(1) For L4 interconnects, only MReqDebug and MReqSupervisor are available.
(2) i = 0 to 1 for PER1_AP
i = 0 for PER2_AP
i = 0 for PER3_AP
i = 0 to 2 for CFG_AP
i = 0 to 3 for WKUP_AP
(3) k = 0 to 7 for PER1_AP
k = 0 to 7 for PER2_AP
k = 0 to 7 for PER3_AP
k = 0 to 7 for CFG_AP
k = 0 to 7 for WKUP_AP
(4) l = 0 to 84 for PER1_AP
l = 0 to 56 for PER2_AP
l = 0 to 96 for PER3_AP
l = 0 to 128 for CFG_AP
l = 0 to 43 for WKUP_AP

14.3.3.4 L4 Error Detection and Reporting

14.3.3.4.1 IA and TA Error Detection and Logging

The L4 interconnect provides mechanisms for handling internally detected errors or errors reported by modules attached to the L4 target ports.

Note

L4_IA denotes the IA for all L4 interconnects: L4_PER1, L4_PER2, L4_PER3, L4_CFG, and L4_WKUP.

L4_TA denotes the TA for all L4 interconnects: L4_PER1, L4_PER2, L4_PER3, L4_CFG, and L4_WKUP.

The L4 interconnects handle four types of errors:

- No target core found or address hole, detected and logged at IA
- Unsupported command, detected and logged at IA
- Protection violation, detected and logged at IA (see Section 14.3.3.3, *L4 Firewalls*)
- Target does not service a request before a time-out expires. The error is detected and logged at TA (see Section 14.3.3.4.2, *Time-Out*).

Table 14-401 lists the value of the L4_IA_ERROR_LOG_L[25:24] CODE bit field stored when an error occurs.

Table 14-401. L4 CODE Bit Field Definition

CODE (bits 1:0)	Error Type	REQ_INFO	Secondary	ConnID	CMD
0	No error				
1	Unsupported command	x	x	x	x
2	Address hole	x	x	x	x
3	Protection violation	x		x	x

- No target core found/address hole: This error indicates that a request was addressed to a hole in the L4 address map. When this error occurs, an in-band error response is returned to the L3 level. The error is also logged into the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. Additionally, an address hole error code is logged to the L4_IA_ERROR_LOG_L[25:24] CODE bit field.
- Unsupported command: This error indicates that the command type of the request is not supported by the accessed target register. The error is logged into the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. An unsupported command error code is written to the L4_IA_ERROR_LOG_L[25:24] CODE bit field for the initiator interface.
- Protection violation: This error indicates that a request is not issued from an allowed initiator member or is issued with the inappropriate ReqInfo qualifiers associated with the target region. This error is reported using an in-band error and is written to the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. A protection violation error code is saved into the L4_IA_ERROR_LOG_L[25:24] CODE bit field for the same initiator interface. A protection violation is also logged in the L4_IA_AGENT_STATUS_L[31] PROT_ERROR_SECONDARY or [30] PROT_ERROR_PRIMARY bit when in debug or applicative mode, respectively.

  The L4_IA_ERROR_LOG_L[30] SECONDARY bit indicates whether the error occurred in application or debug.

  The L4_IA_ERROR_LOG_H[15:0] REQ_INFO bit field returns the type of access (REQ_INFO qualifier) that caused the error.

  The L4_IA_ERROR_LOG_L[13:8] CONNID bit field returns the ID of the initiator that caused the error.

  The L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR register logs the address for error conditions.

14.3.3.4.2 Time-Out

A time-out mechanism can be enabled at the interconnect level and on a per-target basis. If the mechanism is enabled for a TA and interconnect and commands are not accepted or responses are not returned within the expected delay, the L4 interconnect generates an error event.

Note

The time-out mechanism is not available on the L4_WKUP interconnect, but L4_WKUP time-outs are detected in CFG_TA_L4WKUP of the L4_CFG interconnect.

The error is logged in the L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT bit. The affected TA enters an error state that causes it to send an error response to any new request targeted at it. To recover from this state, system software must reset the TA. The time-out is counted starting from the moment a command is presented to the target, regardless of the target response to this command.

The L4 interconnect implements a centralized time-base circuit that broadcasts a set of four periodic pulse signals to all connected TAs. The time-base circuit offers four possible sets of four time-base signals. The time-base signals are selected by programming the L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE bit field.

The selected time-base signals are available at any TA. Each TA can be programmed to refer to one of these four time-base signals, using the L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT bit field. These four signals are referred to as 1X time-base, 4X time-base, 16X time-base, and 64X time-base.

Table 14-402 lists all values in number of L4 clock cycles.

Table 14-402. L4 Time-out Link and TA Programming

<table>
<tr><th></th><th colspan="5">L4_TA_AGENT_CONTROL_L
[10:8] REQ_TIMEOUT</th></tr>
<tr><th>L4_LA_NETWORK_CONTROL_L[10:8]
TIMEOUT_BASE</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td>0</td><td colspan="5">All L4 time-out features are disabled.</td></tr>
<tr><td>1</td><td rowspan="4">Locally disabled</td><td>64</td><td>256</td><td>1024</td><td>4096</td></tr>
<tr><td>2</td><td>256</td><td>1024</td><td>4096</td><td>16,384</td></tr>
<tr><td>3</td><td>1024</td><td>4096</td><td>16,384</td><td>65,536</td></tr>
<tr><td>4</td><td>4096</td><td>16,384</td><td>65,536</td><td>262,144</td></tr>
</table>

The default reset value is 0x2 for REQ_TIMEOUT and 0x4 for TIMEOUT_BASE, implying 16,384 clock cycles.

A time-out condition is detected when the command acceptance or the response is not received after a delay of from one to three time-base periods.

Example:

- L4 frequency = 65-MHz, 15.3-μs period
- TIMEOUT_BASE = 4 in the L4_LA_NETWORK_CONTROL_L register
- REQ_TIMEOUT = 2 in the L4_TA_AGENT_CONTROL_L for TA A
- REQ_TIMEOUT = 4 in the L4_TA_AGENT_CONTROL_L for TA B

At agent A, the time-base unit is 16,384 cycles. A time-out is issued when a request to the attached module is not accepted, or no response is sent after a delay of 252 μs to 756 μs.

At agent B, the time-base unit is 262,144 cycles. A time-out is issued when a request to the attached module is not accepted or no response is sent after a delay of 4 ms to 12 ms.

When a time-out condition is detected, the TA logs the error in the L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT bit, and it also reports the error to the IA, which forwards it to the L3 interconnects.

After the time-out is detected and logged, the behavior of the attached module is ignored. A new request targeting the module arriving at the timed-out TA receives an error response. If the request is addressed to the agent internal registers, it is processed normally.

To recover from a time-out error, software is assumed to first reset the faulty module and then the TA using the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit.

14.3.3.4.3 Error Reporting

Figure 14-13 shows the error-reporting scheme used in the L4 interconnects. All L4 in-band errors are reported to their respective L3 TA, where errors are converted in an out-of band error signal (the L3 applicative and debug composite flags) going to the L3 INTCs.

Two levels of mask are present to report the error at INTC level:

- At the applicative and debug composite flag, to enable interrupt reporting, the following bits must be set:
  – L4_CFG in L3_FLAGMUX_MASK0 and L3_FLAGMUX_STATUS1
  – L4_PER in L3_FLAGMUX_MASK0 and L3_FLAGMUX_STATUS1
- At the L3 TA level, see Section 14.2.1, *L3 Interconnect*.

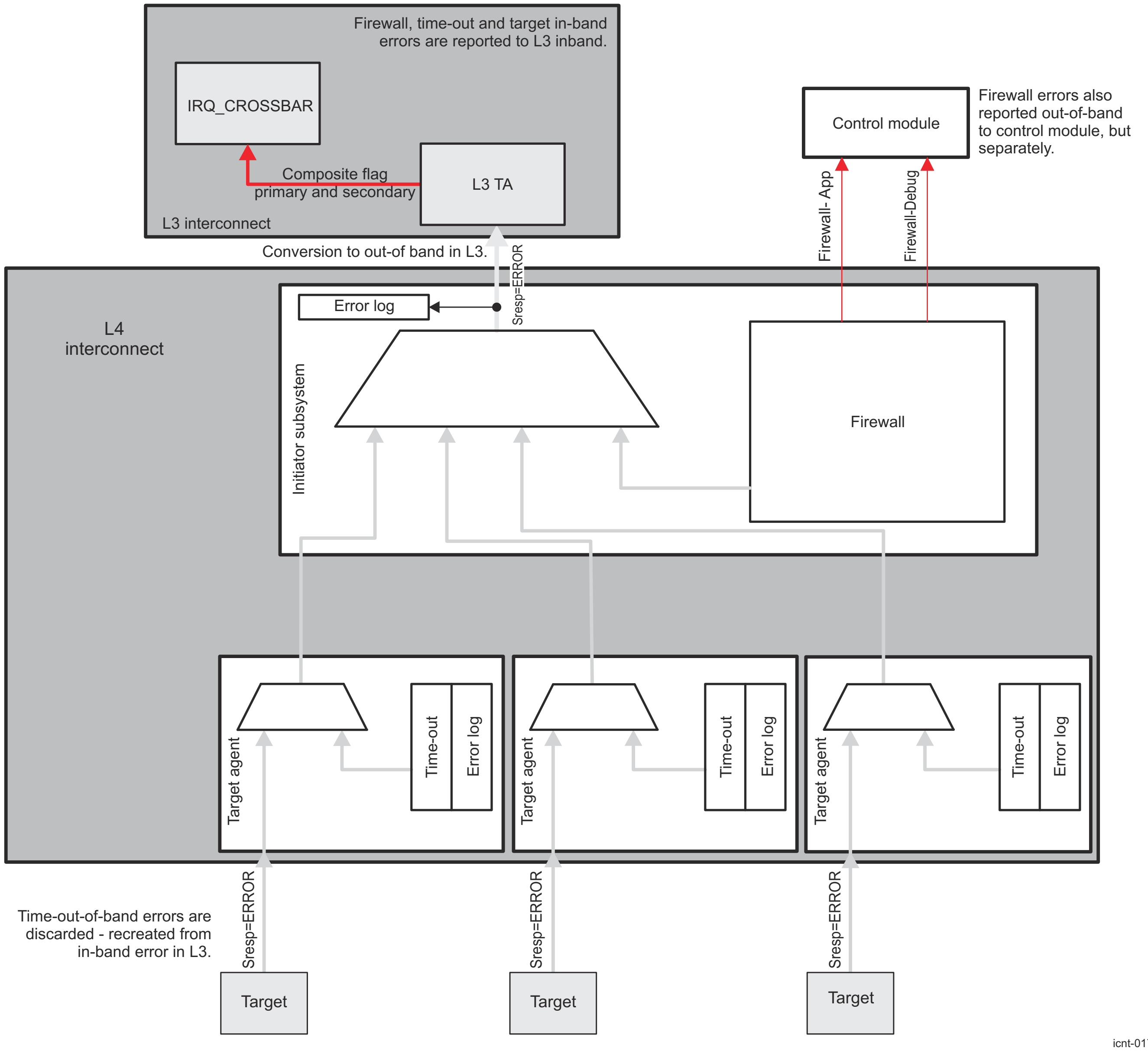


Figure 14-13. L4 Error Reporting

14.3.3.4.4 Error Recovery

Setting the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit to 1 initiates the software reset period. Software reset must be asserted for at least 16 cycles of the target module interface clock, which can be a divided clock with respect to the L4 clock.

During the software reset period:

- Requests sent to the target module receive error responses. Therefore, if the faulty request is part of a DMA transfer, it is necessary to stop the DMA to prevent unwanted errors.
- Requests sent to the TA register block are processed as usual.
- The L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT status bit is cleared.

Setting the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit to 0 terminates the software reset period.

Reset the attached module to complete the recovery.

14.3.3.4.5 Firewall Error Logging in the Control Module

When a protection violation occurs, an interrupt is send to the INTCs (if enabled). An in-band error is sent back to L3 IA and an out-of-band error can also be logged in the CONTROL.CONTROL_SEC_ERR_STATUS register in the SCM. These out-of-band errors are enabled and disabled at the L4 IA level by setting the L4_IA_AGENT_CONTROL_L[31][30] PROT_ERROR_SECONDARY_REP or PROT_ERROR_PRIMARY_REP bit to 1 for debug and application mode, respectively.

At the control module level, two logging registers are used, depending on the mode:

- In application mode or primary error reporting:
  - SEC_ERR_STATUS_FUNC_1[16] = L4_PER1 protection violation
  - SEC_ERR_STATUS_FUNC_2[4] = L4_PER2 protection violation
  - SEC_ERR_STATUS_FUNC_2[5] = L4_PER3 protection violation
  - SEC_ERR_STATUS_FUNC_1[17] = L4_CFG protection violation
  - SEC_ERR_STATUS_FUNC_1[22] = L4_WKUP protection violation
- In debug mode or secondary error reporting:
  - SEC_ERR_STATUS_DEBUG_1[16] = L4_PER1 protection violation
  - SEC_ERR_STATUS_DEBUG_2[4] = L4_PER2 protection violation
  - SEC_ERR_STATUS_DEBUG_2[5] = L4_PER3 protection violation
  - SEC_ERR_STATUS_DEBUG_1[17] = L4_CFG protection violation
  - SEC_ERR_STATUS_DEBUG_1[22] = L4_WKUP protection violation

Figure 14-14 shows the global protection error reporting to the control module.

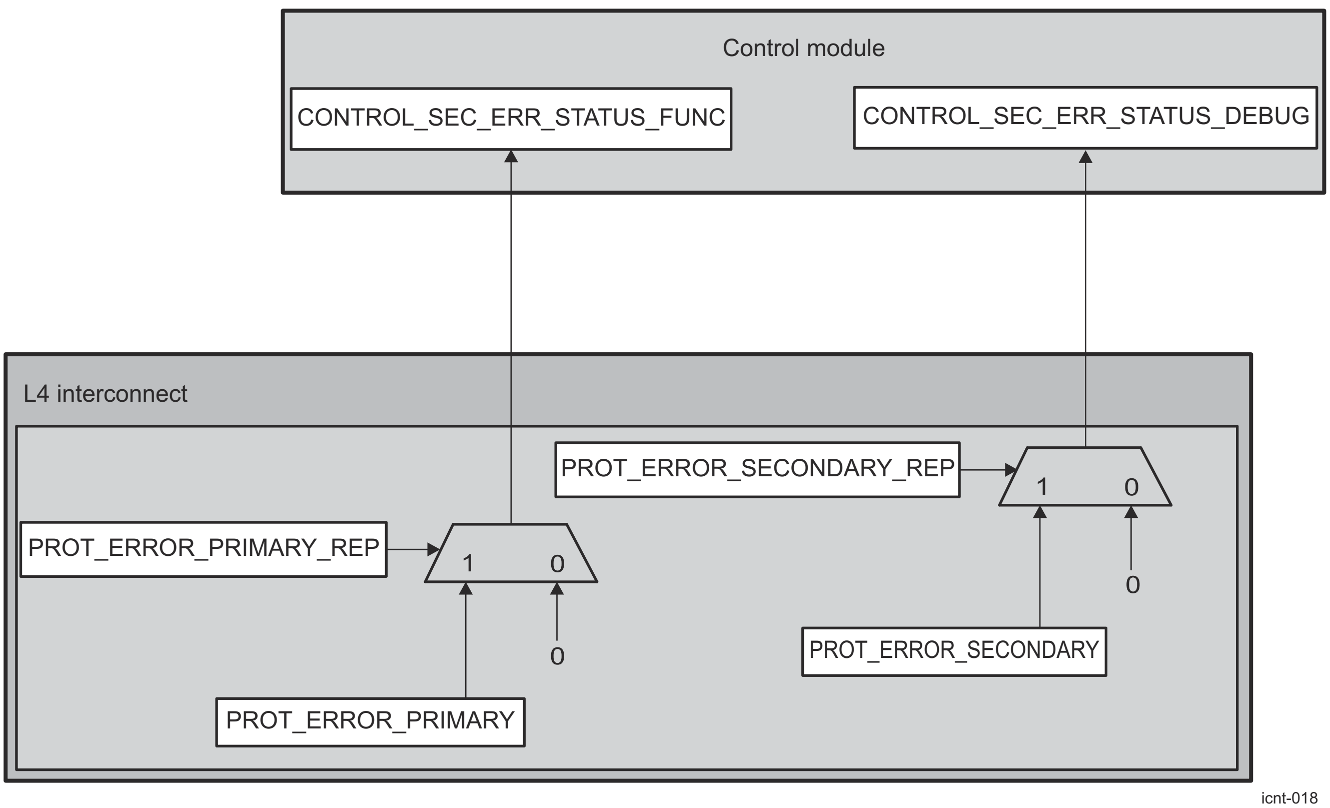


Figure 14-14. Protection Violation Out-of-Band Error Reporting

14.3.4 L4 Interconnect Programming Guide

14.3.4.1 L4 Interconnect Low-level Programming Models

This section describes the low-level hardware programming sequences for configuring and using the L4 interconnect module.

14.3.4.1.1 Global Initialization

14.3.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the L4 interconnect module is used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the L4 interconnect. For more information, see Section 14.3.2, *L4 Interconnect Integration*.

Table 14-403 lists the surrounding modules.

Table 14-403. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	For more information about the configuration of the module, see Section 3.6, in *Power, Reset, and Clock Management*.
Control module	For more information about the configuration of the module, see Chapter 18, *Control Module*.
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 17, *Interrupt Controllers*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.
L3 interconnect	For more information about the interconnect configuration, see Section 14.2.1, *L3 Interconnect*.

14.3.4.1.2 Operational Modes Configuration

14.3.4.1.2.1 L4 Interconnect Error Analysis Mode

14.3.4.1.2.1.1 Main Sequence: L4 Interconnect Error Analysis Mode

The information required to analyze an error source is logged in several registers. The number of registers to access depends on the error source.

Figure 14-15 shows the software sequence required in most cases.

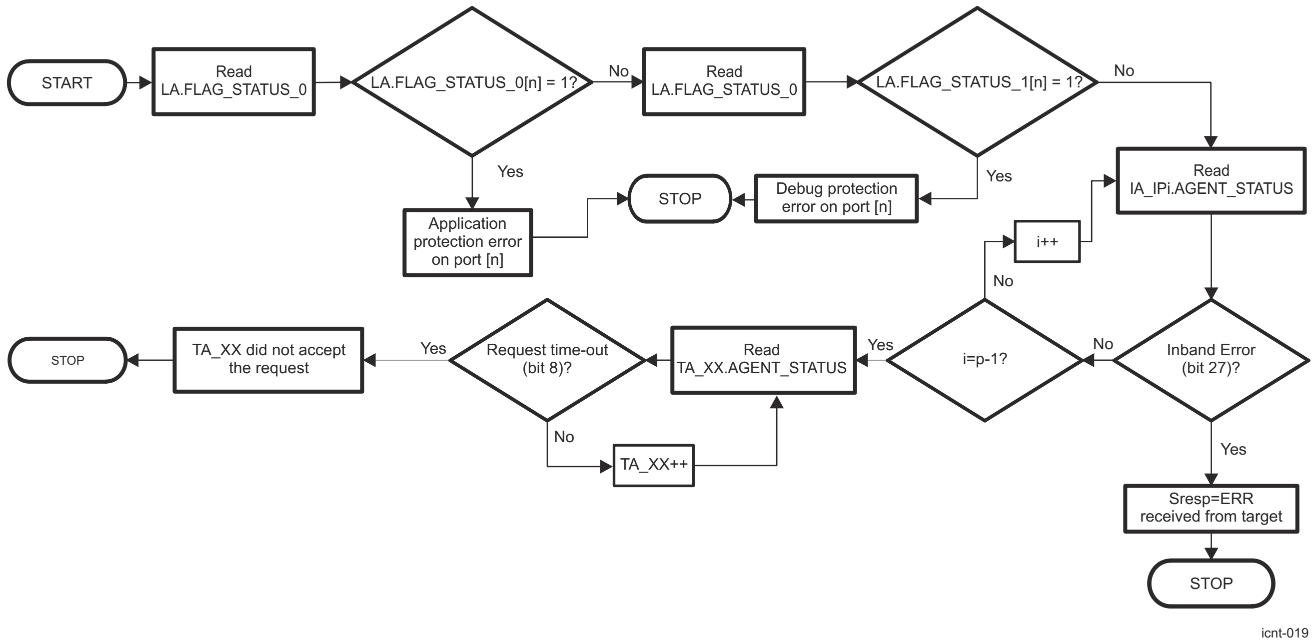


Figure 14-15. Typical Error Analysis Sequence

Note

L4 interconnects don't log any address or other specific information for a custom error returned from any target IP. They rather pass an error response up to the master supposed to analyze it.

In case of posted writes, the master access completes before it actually completed at the end slave level, this way no error response is sent back to the master, making it impossible to have a direct way of uderstanding the origin of L4 error during posted writes. However, even though no address is logged, an error flag is generated and needs to be processed.

14.3.4.1.2.1.2 Subsequence: L4 Interconnect Protection Violation Error Identification

This procedure describes the protection violation error identification (see Table 14-404).

Table 14-404. Protection Violation Error Identification

Step	Register/Bit Field/Programming Model	Value
Read multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	x
Read initiator ID.	L4_IA_ERROR_LOG_L[13:8] CONNID	xxxxx
Read command that cause the error.	L4_IA_ERROR_LOG_L[2:0] CMD	xxx
Read address of request that caused the error.	L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR	xxxxxxxxx
IF: Is it a primary error?	L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY	=0x1
Read status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_FUNC	xx
Write 1 to clear status bits.	CONTROL.SEC_ERR_STATUS_FUNC_1 [16][17][22] CONTROL.SEC_ERR_STATUS_FUNC_2 [4][5]	xxxx
Write 1 to clear IA status bit.	L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY	0x1
ELSE		
Read status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_DEBUG	xx
Write 1 to clear status bits.	CONTROL.SEC_ERR_STATUS_DEBUG_1 [16][17] [22] CONTROL.SEC_ERR_STATUS_DEBUG_2 [4][5]	xxx

Table 14-404. Protection Violation Error Identification (continued)

Step	Register/Bit Field/Programming Model	Value
Write 1 to clear IA status bit	L4_IA_AGENT_STATUS_L[31] PROT_ERROR_SECONDARY	0x1
ENDIF		
Write 1 to clear multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x1
Write 1 to clear in-band error status.	L4_IA_AGENT_STATUS_L[24] INBAND_ERROR	0x1

14.3.4.1.2.1.3 Subsequence: L4 Interconnect Unsupported Command/Address Hole Error Identification

This procedure describes the identification of unsupported command/address hole error (see Table 14-405).

Table 14-405. Unsupported Command/Address Hole Error Identification

Step	Register/Bit Field/Programming Model	Value
Read multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	x
Read initiator ID.	L4_IA_ERROR_LOG_L[11:8] CONNID	xxxx
Read command that caused the error.	L4_IA_ERROR_LOG_L[2:0] CMD	xxx
Read address of request that caused the error.	L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR	xxxxxxxx
Read secondary status.	L4_IA_ERROR_LOG_L[30] SECONDARY	x
Write 1 to clear secondary status.	L4_IA_ERROR_LOG_L[30] SECONDARY	0x1
Write 1 to clear multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x1
Write 1 to clear inband error status.	L4_IA_AGENT_STATUS_L[24] INBAND_ERROR	0x1

14.3.4.1.2.1.4 Subsequence: L4 Interconnect Reset TA and Module

This procedure describes the reset TA and module (see Table 14-406).

Table 14-406. Reset TA and Module

Step	Register/Bit Field/Programming Model	Value
Reset TA.	L4_TA_AGENT_CONTROL_L[0] OCP_RESET	0x1
Wait until target module clock = 16 cycles.		
Write 0 to clear TA time-out status.	L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT	0x0
Write 0 to clear TA reset.	L4_TA_AGENT_CONTROL_L[0] OCP_RESET	0x0
Reset the attached module.[(1)]		

(1) For more information, see the respective module chapter.

14.3.4.1.2.2 L4 Interconnect Time-Out Configuration Mode

14.3.4.1.2.2.1 Main Sequence: L4 Interconnect Time-Out Configuration Mode

This procedure describes the time-out configuration sequence (see Table 14-407).

Table 14-407. Time-Out Configuration

Step	Register/Bit Field/Programming Model	Value
Disable time-out.	L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE	0x0
Clear TA time-out error status.[(1)]	L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT	0x1
Set time-out at TA level.[(1)]	L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT	xxx
Set time-out base.	L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE	xxx

(1) Required for each TA.

14.3.4.1.2.3 L4 Interconnect Firewall Configuration Mode

14.3.4.1.2.3.1 Main Sequence: L4 Interconnect Firewall Configuration Mode

This procedure describes the firewall configuration sequence (see Table 14-408).

Table 14-408. Firewall Configuration

Step	Register/Bit Field/Programming Model	Value
Define the members of protection group k.(1)	L4_AP_PROT_GROUP_MEMBERS_k_L[15:0] CONNID_BIT_VECTOR	xxx
Define the access type of a protection group k.(1)	L4_AP_PROT_GROUP_ROLES_k_L[15:0] ENABLE	xx
Set region affiliation to protection group.(2)	L4_AP_REGION_l_L[22:20] PROT_GROUP_ID	xxx

(1) Required for each protection group.
(2) Required for each region.

14.3.5 L4 Interconnects Register Manual

Table 14-409 through Table 14-413 list all L4 register blocks for IA, TA, AP, and LA. Each module instance is shown with the module register mapping and bit and bit field definitions.

14.3.5.1 L4 Interconnects Instance Summary

Table 14-409. L4_PER1 Instance Summary

Module Name	L3_MAIN Base Address	Size
PER1_AP	0x4800 0000	2KB
PER1_LA	0x4800 0800	512B
PER1_IA_IP0	0x4800 1000	128B
PER1_IA_IP1	0x4800 1400	128B
PER1_IA_IP2	0x4800 1800	128B
UART3_TARG	0x4802 1000	64B
TIMER2_TARG	0x4803 3000	64B
TIMER3_TARG	0x4803 5000	64B
TIMER4_TARG	0x4803 7000	64B
TIMER9_TARG	0x4803 F000	64B
GPIO7_TARG	0x4805 2000	64B
GPIO8_TARG	0x4805 4000	64B
GPIO2_TARG	0x4805 6000	64B
GPIO3_TARG	0x4805 8000	64B
GPIO4_TARG	0x4805 A000	64B
GPIO5_TARG	0x4805 C000	64B
GPIO6_TARG	0x4805 E000	64B
I2C3_TARG	0x4806 1000	64B
UART5_TARG	0x4806 7000	64B
UART6_TARG	0x4806 9000	64B
UART1_TARG	0x4806 B000	64B
UART2_TARG	0x4806 D000	64B
UART4_TARG	0x4806 F000	64B
I2C1_TARG	0x4807 1000	64B
I2C2_TARG	0x4807 3000	64B
ELM_TARG	0x4807 9000	64B
I2C4_TARG	0x4807 B000	64B
I2C5_TARG	0x4807 D000	64B
TIMER10_TARG	0x4808 7000	64B
TIMER11_TARG	0x4808 9000	64B
MCSPI1_TARG	0x4809 9000	64B
MCSPI2_TARG	0x4809 B000	64B
MMC1_TARG	0x4809 D000	64B
MMC3_TARG	0x480A E000	64B
HDQ1W_TARG	0x480B 3000	64B
MMC2_TARG	0x480B 5000	64B
MCSPI3_TARG	0x480B 9000	64B
MCSPI4_TARG	0x480B B000	64B
MMC4_TARG	0x480D 2000	64B

Table 14-410. L4_PER2 Instance Summary

Module Name	L3_MAIN Base Address	Size
PER2_AP	0x4840 0000	2KBytes
PER2_LA	0x4840 0800	512Bytes
PER2_IA_IP0	0x4840 1000	128Bytes
PER2_IA_IP1	0x4840 1400	128Bytes
PER2_IA_IP2	0x4840 1800	128Bytes
UART7_TARG	0x4842 1000	64Bytes
UART8_TARG	0x4842 3000	64Bytes
UART9_TARG	0x4842 5000	64Bytes
MLB_TARG	0x4842 D000	64Bytes
MCASP4_DAT_TARG	0x4843 7000	64Bytes
MCASP5_DAT_TARG	0x4843 B000	64Bytes
ATL_TARG	0x4843 D000	64Bytes
PWM1_TARG	0x4843 F000	64Bytes
PWM2_TARG	0x4844 1000	64Bytes
PWM3_TARG	0x4844 3000	64Bytes
VCP1_CFG_TARG	0x4844 7000	64Bytes
VCP2_CFG_TARG	0x4844 9000	64Bytes
MCASP6_DAT_TARG	0x4844 D000	64Bytes
MCASP7_DAT_TARG	0x4845 1000	64Bytes
MCASP8_DAT_TARG	0x4845 5000	64Bytes
MCASP1_CFG_TARG	0x4846 2000	64Bytes
MCASP2_CFG_TARG	0x4846 6000	64Bytes
MCASP3_CFG_TARG	0x4846 A000	64Bytes
MCASP4_CFG_TARG	0x4846 E000	64Bytes
MCASP5_CFG_TARG	0x4847 2000	64Bytes
MCASP6_CFG_TARG	0x4847 6000	64Bytes
MCASP7_CFG_TARG	0x4847 A000	64Bytes
MCASP8_CFG_TARG	0x4847 E000	64Bytes
DCAN2_TARG	0x4848 2000	64Bytes
GMAC_TARG	0x4848 8000	64Bytes

Table 14-411. L4_PER3 Instance Summary

Module Name	L3_MAIN Base Address	Size
PER3_AP	0x4880 0000	2KBytes
PER3_LA	0x4880 0800	512Bytes
PER3_IA_IP0	0x4880 1000	128Bytes
PER3_IA_IP1	0x4880 1400	128Bytes
PER3_IA_IP2	0x4880 1800	128Bytes
MBX13_TARG	0x4880 3000	64Bytes
OCMC_RAM1_TARG	0x4880 5000	64Bytes
OCMC_RAM2_TARG	0x4880 B000	64Bytes
OCMC_RAM3_TARG	0x4881 1000	64Bytes
MMU1_TARG	0x4881 D000	64Bytes
MMU2_TARG	0x4881 F000	64Bytes
TIMER5_TARG	0x4882 1000	64Bytes
TIMER6_TARG	0x4882 3000	64Bytes
TIMER7_TARG	0x4882 5000	64Bytes

Table 14-411. L4_PER3 Instance Summary (continued)

Module Name	L3_MAIN Base Address	Size
TIMER8_TARG	0x4882 7000	64Bytes
TIMER13_TARG	0x4882 9000	64Bytes
TIMER14_TARG	0x4882 B000	64Bytes
TIMER15_TARG	0x4882 D000	64Bytes
TIMER16_TARG	0x4882 F000	64Bytes
RTC_TARG	0x4883 9000	64Bytes
MBX2_TARG	0x4883 B000	64Bytes
MBX3_TARG	0x4883 D000	64Bytes
MBX4_TARG	0x4883 F000	64Bytes
MBX5_TARG	0x4884 1000	64Bytes
MBX6_TARG	0x4884 3000	64Bytes
MBX7_TARG	0x4884 5000	64Bytes
MBX8_TARG	0x4884 7000	64Bytes
MBX9_TARG	0x4885 F000	64Bytes
MBX10_TARG	0x4886 1000	64Bytes
MBX11_TARG	0x4886 3000	64Bytes
MBX12_TARG	0x4886 5000	64Bytes
USB1_CFG_TARG	0x488A 0000	64Bytes
USB2_CFG_TARG	0x488E 0000	64Bytes
USB3_CFG_TARG	0x4892 0000	64Bytes
USB4_CFG_TARG	0x4896 0000	64Bytes
VIP1_TARG	0x4898 0000	64Bytes
VIP2_TARG	0x489A 0000	64Bytes
VIP3_TARG	0x489C 0000	64Bytes
VPE_TARG	0x489E 0000	64Bytes

Table 14-412. L4_CFG Instance Summary

Module Name	L3_MAIN Base Address	Size
CFG_AP	0x4A00 0000	2KBytes
CFG_LA	0x4A00 0800	64Bytes
CFG_IA_IP0	0x4A00 1000	128Bytes
CTRL_MODULE_CORE_TARG	0x4A00 4000	64Bytes
CM_CORE_AON_TARG	0x4A00 6000	64Bytes
CM_CORE_TARG	0x4A00 A000	64Bytes
DMA_SYSTEM_TARG	0x4A05 7000	64Bytes
SCP1_TARG	0x4A08 8000	64Bytes
SCP3_TARG	0x4A09 8000	64Bytes
SCP2_TARG	0x4A0A 8000	64Bytes
MAILBOX_TARG	0x4A0F 5000	64Bytes
SPINLOCK_TARG	0x4A0F 7000	64Bytes
OCP_WP_NOC_TARG	0x4A10 3000	64Bytes
EVE1_FW_CFG_TARG	0x4A15 2000	64Bytes
EVE2_FW_CFG_TARG	0x4A15 4000	64Bytes
PCIESS2_FW_CFG_TARG	0x4A15 A000	64Bytes
IPU1_FW_CFG_TARG	0x4A15 C000	64Bytes
VCP1_FW_CFG_TARG	0x4A15 E000	64Bytes
VCP2_FW_CFG_TARG	0x4A16 0000	64Bytes

Table 14-412. L4_CFG Instance Summary (continued)

Module Name	L3_MAIN Base Address	Size
TPCC_FW_CFG_TARG	0x4A16 2000	64Bytes
TPTC_FW_CFG_TARG	0x4A16 4000	64Bytes
PCIESS1_FW_CFG_TARG	0x4A16 6000	64Bytes
MCASP1_FW_CFG_TARG	0x4A16 8000	64Bytes
MCASP2_FW_CFG_TARG	0x4A16 A000	64Bytes
MCASP3_FW_CFG_TARG	0x4A16 C000	64Bytes
DSP1_SDMA_FW_CFG_TARG	0x4A17 2000	64Bytes
DSP2_SDMA_FW_CFG_TARG	0x4A17 4000	64Bytes
QSPI_FW_CFG_TARG	0x4A17 A000	64Bytes
MA_MPU_NTTP_FW_CFG_TARG	0x4A20 B000	64Bytes
EMIF_OCP_FW_CFG_TARG	0x4A20 D000	64Bytes
OCMC_RAM2_FW_CFG_TARG	0x4A20 F000	64Bytes
GPMC_FW_CFG_TARG	0x4A21 1000	64Bytes
OCMC_RAM1_FW_CFG_TARG	0x4A21 3000	64Bytes
GPU_FW_CFG_TARG	0x4A21 5000	64Bytes
IPU2_FW_CFG_TARG	0x4A21 9000	64Bytes
DSS_FW_CFG_TARG	0x4A21 D000	64Bytes
IVA_SL2IF_FW_CFG_TARG	0x4A21 F000	64Bytes
IVA_CONFIG_FW_CFG_TARG	0x4A22 1000	64Bytes
DEBUGSS_CT_TBR_FW_CFG_TARG	0x4A22 5000	64Bytes
L3_INSTR_FW_CFG_TARG	0x4A22 7000	64Bytes
OCMC_RAM3_FW_CFG_TARG	0x4A22 B000	64Bytes

Table 14-413. L4_WKUP Instance Summary

Module Name	L3_MAIN Base Address	Size
WKUP_AP	0x4AE0 0000	2KBytes
WKUP_LA	0x4AE0 0800	64Bytes
WKUP_IA_IP0	0x4AE0 1000	128Bytes
COUNTER_32K_TARG	0x4AE0 5000	64Bytes
PRM_TARG	0x4AE0 8000	64Bytes
CTRL_MODULE_WKUP_TARG	0x4AE0 D000	64Bytes
GPIO1_TARG	0x4AE1 1000	64Bytes
WD_TIMER2_TARG	0x4AE1 5000	64Bytes
TIMER1_TARG	0x4AE1 9000	64Bytes
KBD_TARG	0x4AE1 D000	64Bytes
TIMER12_TARG	0x4AE2 1000	64Bytes
UART10_TARG	0x4AE2 C000	64Bytes
DCAN1_TARG	0x4AE3 E000	64Bytes

14.3.5.2 L4 Initiator Agent (L4 IA)

14.3.5.2.1 L4 Initiator Agent (L4 IA) Register Summary

Table 14-414 summarizes the L4 IA register mapping.

Table 14-414. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PER1_IA_IP0 L3_MAIN Physical Address	PER1_IA_IP1 L3_MAIN Physical Address	PER1_IA_IP2 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4800 1000	0x4800 1400	0x4800 1800

Table 14-414. IA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PER1_IA_IP0 L3_MAIN Physical Address	PER1_IA_IP1 L3_MAIN Physical Address	PER1_IA_IP2 L3_MAIN Physical Address
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4800 1004	0x4800 1404	0x4800 1804
L4_IA_CORE_L	R	32	0x0000 0018	0x4800 1018	0x4800 1418	0x4800 1818
L4_IA_CORE_H	R	32	0x0000 001C	0x4800 101C	0x4800 141C	0x4800 181C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4800 1020	0x4800 1420	0x4800 1820
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4800 1024	0x4800 1424	0x4800 1824
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4800 1028	0x4800 1428	0x4800 1828
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4800 102C	0x4800 142C	0x4800 182C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4800 1058	0x4800 1458	0x4800 1858
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4800 105C	0x4800 145C	0x4800 185C
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4800 1060	0x4800 1460	0x4800 1860
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4800 1064	0x4800 1464	0x4800 1864

Table 14-415. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PER2_IA_IP0 L3_MAIN Physical Address	PER2_IA_IP1 L3_MAIN Physical Address	PER2_IA_IP2 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4840 1000	0x4840 1400	0x4840 1800
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4840 1004	0x4840 1404	0x4840 1804
L4_IA_CORE_L	R	32	0x0000 0018	0x4840 1018	0x4840 1418	0x4840 1818
L4_IA_CORE_H	R	32	0x0000 001C	0x4840 101C	0x4840 141C	0x4840 181C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4840 1020	0x4840 1420	0x4840 1820
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4840 1024	0x4840 1424	0x4840 1824
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4840 1028	0x4840 1428	0x4840 1828
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4840 102C	0x4840 142C	0x4840 182C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4840 1058	0x4840 1458	0x4840 1858
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4840 105C	0x4840 145C	0x4840 185C
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4840 1060	0x4840 1460	0x4840 1860
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4840 1064	0x4840 1464	0x4840 1864

Table 14-416. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PER3_IA_IP0 L3_MAIN Physical Address	PER3_IA_IP1 L3_MAIN Physical Address	PER3_IA_IP2 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4880 1000	0x4880 1400	0x4880 1800
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4880 1004	0x4880 1404	0x4880 1804
L4_IA_CORE_L	R	32	0x0000 0018	0x4880 1018	0x4880 1418	0x4880 1818
L4_IA_CORE_H	R	32	0x0000 001C	0x4880 101C	0x4880 141C	0x4880 181C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4880 1020	0x4880 1420	0x4880 1820
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4880 1024	0x4880 1424	0x4880 1824
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4880 1028	0x4880 1428	0x4880 1828
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4880 102C	0x4880 142C	0x4880 182C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4880 1058	0x4880 1458	0x4880 1858
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4880 105C	0x4880 145C	0x4880 185C

Table 14-416. IA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PER3_IA_IP0 L3_MAIN Physical Address	PER3_IA_IP1 L3_MAIN Physical Address	PER3_IA_IP2 L3_MAIN Physical Address
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4880 1060	0x4880 1460	0x4880 1860
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4880 1064	0x4880 1464	0x4880 1864

Table 14-417. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CFG_IA_IP0 L3_MAIN Physical Address	WKUP_IA_IP0 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4A00 1000	0x4AE0 1000
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4A00 1004	0x4AE0 1004
L4_IA_CORE_L	R	32	0x0000 0018	0x4A00 1018	0x4AE0 1018
L4_IA_CORE_H	R	32	0x0000 001C	0x4A00 101C	0x4AE0 101C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A00 1020	0x4AE0 1020
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A00 1024	0x4AE0 1024
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4A00 1028	0x4AE0 1028
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A00 102C	0x4AE0 102C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4A00 1058	0x4AE0 1058
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4A00 105C	0x4AE0 105C
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4A00 1060	0x4AE0 1060
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4A00 1064	0x4AE0 1064

14.3.5.2.2 L4 Initiator Agent (L4 IA) Register Description

Table 14-418 through Table 14-440 describe the L4 IA registers.

Table 14-418. L4_IA_COMPONENT_L

Address Offset	0x0000 0000		
Physical Address	0x4800 1000 0x4800 1400 0x4800 1800 0x4840 1000 0x4840 1400 0x4840 1800 0x4880 1000 0x4880 1400 0x4880 1800 0x4A00 1000 0x4AE0 1000	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code	R	See [(1)].
15:0	REV	Component revision code	R	See[(1)].

(1) TI Internal Data

Table 14-419. Register Call Summary for Register L4_IA_COMPONENT_L

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0] [1] [2] [3]

Table 14-420. L4_IA_COMPONENT_H

Address Offset	0x0000 0004		
Physical Address	0x4800 1004 0x4800 1404 0x4800 1804 0x4840 1004 0x4840 1404 0x4840 1804 0x4880 1004 0x4880 1404 0x4880 1804 0x4A00 1004 0x4AE0 1004	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000

Table 14-421. Register Call Summary for Register L4_IA_COMPONENT_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0] [1] [2] [3]

Table 14-422. L4_IA_CORE_L

Address Offset	0x0000 0018		
Physical Address	0x4800 1018 0x4800 1418 0x4800 1818 0x4840 1018 0x4840 1418 0x4840 1818 0x4880 1018 0x4880 1418 0x4880 1818 0x4A00 1018 0x4AE0 1018	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Provide information about the core initiator		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CODE																CORE_REV															

Bits	Field Name	Description	Type	Reset
31:16	CORE_CODE	Interconnect core code	R	See [(1)].
15:0	CORE_REV	Component revision code code	R	See[(1)].

(1) TI Internal Data

Table 14-423. Register Call Summary for Register L4_IA_CORE_L

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0] [1] [2] [3]

Table 14-424. L4_IA_CORE_H

Address Offset	0x0000 001C		
Physical Address	0x4800 101C 0x4800 141C 0x4800 181C 0x4840 101C 0x4840 141C 0x4840 181C 0x4880 101C 0x4880 141C 0x4880 181C 0x4A00 101C 0x4AE0 101C	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Provide information about the core initiator		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VENDOR_CODE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	VENDOR_CODE	Vendor revision core code	R	See (1).

(1) TI Internal Data

Table 14-425. Register Call Summary for Register L4_IA_CORE_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0] [1] [2] [3]

Table 14-426. L4_IA_AGENT_CONTROL_L

Address Offset	0x0000 0020		
Physical Address	0x4800 1020 0x4800 1420 0x4800 1820 0x4840 1020 0x4840 1420 0x4840 1820 0x4880 1020 0x4880 1420 0x4880 1820 0x4A00 1020 0x4AE0 1020	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Core control for an initiator OCP interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PROT_ERROR_SECONDARY_REP	PROT_ERROR_PRIMARY_REP	RESERVED	INBAND_ERROR_REP	RESERVED	MERROR_REP	RESERVED

Bits	Field Name	Description	Type	Reset
31	PROT_ERROR_SECONDARY_REP	Out-of-band reporting of protection mechanism secondary errors	RW	1
30	PROT_ERROR_PRIMARY_REP	Out-of-band reporting of protection mechanism primary errors	RW	1
29:28	RESERVED	Read returns 0.	R	0x0
27	INBAND_ERROR_REP	Setting this field to 1 reports on in-band errors using the INBAND_ERROR log bit of IA.AGENT_STATUS register. The error reporting mechanism is enabled when the INBAND_ERROR_REP bit field is set to 1.	RW	1
26:25	RESERVED	Read returns 0.	R	0x0
24	MERROR_REP	OCP MError reporting control. The out-of-band OCP MError reporting mechanism is enabled when the MERROR_REP bit field is set to 1.	R	0
23:0	RESERVED		R	0x0

Table 14-427. Register Call Summary for Register L4_IA_AGENT_CONTROL_L

L4 Interconnects

- Firewall Error Logging in the Control Module: [0]
- L4 Initiator Agent (L4 IA) Register Summary: [1] [2] [3] [4]

Table 14-428. L4_IA_AGENT_CONTROL_H

Address Offset	0x0000 0024		
Physical Address	0x4800 1024 0x4800 1424 0x4800 1824 0x4840 1024 0x4840 1424 0x4840 1824 0x4880 1024 0x4880 1424 0x4880 1824 0x4A00 1024 0x4AE0 1024	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Enable error reporting on an initiator interface.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

Table 14-429. Register Call Summary for Register L4_IA_AGENT_CONTROL_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0] [1] [2] [3]

Table 14-430. L4_IA_AGENT_STATUS_L

Address Offset	0x0000 0028		
Physical Address	0x4800 1028	**Instance**	PER1_IA_IP0
	0x4800 1428		PER1_IA_IP1
	0x4800 1828		PER1_IA_IP2
	0x4840 1028		PER2_IA_IP0
	0x4840 1428		PER2_IA_IP1
	0x4840 1828		PER2_IA_IP2
	0x4880 1028		PER3_IA_IP0
	0x4880 1428		PER3_IA_IP1
	0x4880 1828		PER3_IA_IP2
	0x4A00 1028		CFG_IA_IP0
	0x4AE0 1028		WKUP_IA_IP0
Description	Stores status information for an initiator. The INBAND_ERROR and MERROR fields are read/write and are implemented as log bits.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROT_ERROR_SECONDARY	PROT_ERROR_PRIMARY	RESERVED		INBAND_ERROR	RESERVED		MERROR	RESERVED																							

Bits	Field Name	Description	Type	Reset
31	PROT_ERROR_SECONDARY	0x0: Secondary Protection error not present.0x1: Secondary Protection error present	RW W1toClr	0
30	PROT_ERROR_PRIMARY	0x0: Primary Protection error not present.0x1: Primary Protection error present	RW W1toClr	0
29:28	RESERVED	Read returns 0.	R	0x0
27	INBAND_ERROR	0x0 No In-Band error present.0x1 In-Band error present.	RW W1toClr	0
26:25	RESERVED	Read returns 0.	R	0x0
24	MERROR	Value of the OCP MError signal	R	0
23:0	RESERVED	Read returns 0	R	0X0

Table 14-431. Register Call Summary for Register L4_IA_AGENT_STATUS_L

L4 Interconnects

- IA and TA Error Detection and Logging: [0] [1] [2] [3]
- Operational Modes Configuration: [4] [5] [6] [7] [8]
- L4 Initiator Agent (L4 IA) Register Summary: [9] [10] [11] [12]

Table 14-432. L4_IA_AGENT_STATUS_H

Address Offset	0x0000 002C

Table 14-432. L4_IA_AGENT_STATUS_H (continued)

Physical Address	0x4800 102C 0x4800 142C 0x4800 182C 0x4840 102C 0x4840 142C 0x4840 182C 0x4880 102C 0x4880 142C 0x4880 182C 0x4A00 102C 0x4AE0 102C	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Stores status information for an initiator.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

Table 14-433. Register Call Summary for Register L4_IA_AGENT_STATUS_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0] [1] [2] [3]

Table 14-434. L4_IA_ERROR_LOG_L

Address Offset	0x0000 0058		
Physical Address	0x4800 1058 0x4800 1458 0x4800 1858 0x4840 1058 0x4840 1458 0x4840 1858 0x4880 1058 0x4880 1458 0x4880 1858 0x4A00 1058 0x4AE0 1058	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Log information about error conditions. The CODE field logs any protection violation or address hole errors detected by the initiator subsystem while decoding a request.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MULTI	SECONDARY	RESERVED				CODE		RESERVED										CONNID						RESERVED					CMD		

Bits	Field Name	Description	Type	Reset
31	MULTI	Multiple errors detected	RW W1toClr	0
30	SECONDARY	Indicates whether protection violation was a primary or secondary error	RW W1toClr	0
29:26	RESERVED	Read returns 0.	R	0x0

Bits	Field Name	Description	Type	Reset
25:24	CODE	The error code of an initiator request. 0x00: No errors 0x01: Reserved 0x10: Address hole 0x11: Protection violation	RW W1toClr	0x0
23:14	RESERVED	Read returns 0.	R	0x000
13:8	CONNID	ConnID of request causing the error, refer to Table 14-389	R	0x00
7:3	RESERVED	Read returns 0.	R	0x00
2:0	CMD	Command that caused error	R	0x0

Table 14-435. Register Call Summary for Register L4_IA_ERROR_LOG_L

L4 Interconnects
- IA and TA Error Detection and Logging: [0] [1] [2] [3] [4] [5]
- Operational Modes Configuration: [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]
- L4 Initiator Agent (L4 IA) Register Summary: [16] [17] [18] [19]

Table 14-436. L4_IA_ERROR_LOG_H

Address Offset	0x0000 005C		
Physical Address	0x4800 105C 0x4800 145C 0x4800 185C 0x4840 105C 0x4840 145C 0x4840 185C 0x4880 105C 0x4880 145C 0x4880 185C 0x4A00 105C 0x4AE0 105C	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Log information about error conditions.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REQ_INFO															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Read returns 0.	R	0x0000
15:0	REQ_INFO	MReqInfo bits of request that caused the error REQ_INFO[0] = supervisor, REQ_INFO[1] = Debug	R	0x0000

Table 14-437. Register Call Summary for Register L4_IA_ERROR_LOG_H

L4 Interconnects
- IA and TA Error Detection and Logging: [0]
- L4 Initiator Agent (L4 IA) Register Summary: [1] [2] [3] [4]

Table 14-438. L4_IA_ERROR_LOG_ADDR_L

Address Offset	0x0000 0060

Table 14-438. L4_IA_ERROR_LOG_ADDR_L (continued)

Physical Address	0x4800 1060 0x4800 1460 0x4800 1860 0x4840 1060 0x4840 1460 0x4840 1860 0x4880 1060 0x4880 1460 0x4880 1860 0x4A00 1060 0x4AE0 1060	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Extended error log (address information)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Address of request that caused the error. N is the number MAddr bits.	R	0x0000 0000

Table 14-439. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_L

L4 Interconnects

- IA and TA Error Detection and Logging: [0]
- Operational Modes Configuration: [1] [2]
- L4 Initiator Agent (L4 IA) Register Summary: [3] [4] [5] [6]

Table 14-440. L4_IA_ERROR_LOG_ADDR_H

Address Offset	0x0000 0064		
Physical Address	0x4800 1064 0x4800 1464 0x4800 1864 0x4840 1064 0x4840 1464 0x4840 1864 0x4880 1064 0x4880 1464 0x4880 1864 0x4A00 1064 0x4AE0 1064	**Instance**	PER1_IA_IP0 PER1_IA_IP1 PER1_IA_IP2 PER2_IA_IP0 PER2_IA_IP1 PER2_IA_IP2 PER3_IA_IP0 PER3_IA_IP1 PER3_IA_IP2 CFG_IA_IP0 WKUP_IA_IP0
Description	Extended error log (address information)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

Table 14-441. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0] [1] [2] [3]

14.3.5.3 L4 Target Agent (L4 TA)

14.3.5.3.1 L4 Target Agent (L4 TA) Register Summary

Table 14-442 through Table 14-483 summarizes the L4 TA mapping of the CFG_TA, PER_TA, and WKUP_TA registers.

Table 14-442. CFG_TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE_TARG L3_MAIN Physical Address	CM_CORE_AON_TARG L3_MAIN Physical Address	CM_CORE_TARGL3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A00 4000	0x4A00 6000	0x4A00 A000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A00 4004	0x4A00 6004	0x4A00 A004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A00 4018	0x4A00 6018	0x4A00 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A00 401C	0x4A00 601C	0x4A00 A01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A00 4020	0x4A00 6020	0x4A00 A020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A00 4024	0x4A00 6024	0x4A00 A024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A00 4028	0x4A00 6028	0x4A00 A028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A00 402C	0x4A00 602C	0x4A00 A02C

Table 14-443. CFG_TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	DMA_SYSTEM_TARG L3_MAIN Physical Address	SCP1_TARG L3_MAIN Physical Address	SCP3_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A05 7000	0x4A08 8000	0x4A09 8000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A05 7004	0x4A08 8004	0x4A09 8004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A05 7018	0x4A08 8018	0x4A09 8018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A05 701C	0x4A08 801C	0x4A09 801C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A05 7020	0x4A08 8020	0x4A09 8020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A05 7024	0x4A08 8024	0x4A09 8024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A05 7028	0x4A08 8028	0x4A09 8028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A05 702C	0x4A08 802C	0x4A09 802C

Table 14-444. CFG_TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	SCP2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A0A 8000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A0A 8004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A0A 8018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A0A 801C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A0A 8020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A0A 8024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A0A 8028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A0A 802C

Table 14-445. CFG_TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX_TARG L3_MAIN Physical Address	SPINLOCK_TARG L3_MAIN Physical Address	OCP_WP_NOC_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A0F 5000	0x4A0F 7000	0x4A10 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A0F 5004	0x4A0F 7004	0x4A10 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A0F 5018	0x4A0F 7018	0x4A10 3018

Table 14-445. CFG_TA Register Mapping Summary 4 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX_TARG L3_MAIN Physical Address	SPINLOCK_TARG L3_MAIN Physical Address	OCP_WP_NOC_TARG L3_MAIN Physical Address
L4_TA_CORE_H	R	32	0x0000 001C	0x4A0F 501C	0x4A0F 701C	0x4A10 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A0F 5020	0x4A0F 7020	0x4A10 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A0F 5024	0x4A0F 7024	0x4A10 3024
L4_TA_AGENT_STATUS_H	R	32	0x0000 0028	0x4A0F 5028	0x4A0F 7028	0x4A10 3028
L4_TA_AGENT_STATUS_L	R	32	0x0000 002C	0x4A0F 502C	0x4A0F 702C	0x4A10 302C

Table 14-446. CFG_TA Register Mapping Summary 5

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_FW_CFG_TARG L3_MAIN Physical Address	EVE2_FW_CFG_TARG L3_MAIN Physical Address	PCIESS2_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A15 2000	0x4A15 4000	0x4A15 A000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A15 2004	0x4A15 4004	0x4A15 A004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A15 2018	0x4A15 4018	0x4A15 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A15 201C	0x4A15 401C	0x4A15 A01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A15 2020	0x4A15 4020	0x4A15 A020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A15 2024	0x4A15 4024	0x4A15 A024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A15 2028	0x4A15 4028	0x4A15 A028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A15 202C	0x4A15 402C	0x4A15 A02C

Table 14-447. CFG_TA Register Mapping Summary 6

Register Name	Type	Register Width (Bits)	Address Offset	IPU1_FW_CFG_TARG L3_MAIN Physical Address	VCP1_FW_CFG_TARG L3_MAIN Physical Address	VCP2_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A15 C000	0x4A15 E000	0x4A16 0000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A15 C004	0x4A15 E004	0x4A16 0004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A15 C018	0x4A15 E018	0x4A16 0018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A15 C01C	0x4A15 E01C	0x4A16 001C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A15 C020	0x4A15 E020	0x4A16 0020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A15 C024	0x4A15 E024	0x4A16 0024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A15 C028	0x4A15 E028	0x4A16 0028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A15 C02C	0x4A15 E02C	0x4A16 002C

Table 14-448. CFG_TA Register Mapping Summary 7

Register Name	Type	Register Width (Bits)	Address Offset	TPCC_FW_CFG_TARG L3_MAIN Physical Address	TPTC_FW_CFG_TARG L3_MAIN Physical Address	PCIESS1_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A16 2000	0x4A16 4000	0x4A16 6000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A16 2004	0x4A16 4004	0x4A16 6004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A16 2018	0x4A16 4018	0x4A16 6018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A16 201C	0x4A16 401C	0x4A16 601C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A16 2020	0x4A16 4020	0x4A16 6020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A16 2024	0x4A16 4024	0x4A16 6024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A16 2028	0x4A16 4028	0x4A16 6028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A16 202C	0x4A16 402C	0x4A16 602C

Table 14-449. CFG_TA Register Mapping Summary 8

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_FW_CFG_TARG L3_MAIN Physical Address	MCASP2_FW_CFG_TARG L3_MAIN Physical Address	MCASP3_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A16 8000	0x4A16 A000	0x4A16 C000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A16 8004	0x4A16 A004	0x4A16 C004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A16 8018	0x4A16 A018	0x4A16 C018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A16 801C	0x4A16 A01C	0x4A16 C01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A16 8020	0x4A16 A020	0x4A16 C020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A16 8024	0x4A16 A024	0x4A16 C024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A16 8028	0x4A16 A028	0x4A16 C028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A16 802C	0x4A16 A02C	0x4A16 C02C

Table 14-450. CFG_TA Register Mapping Summary 9

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_SDMA_FW_CFG_TARG L3_MAIN Physical Address	DSP2_SDMA_FW_CFG_TARG L3_MAIN Physical Address	QSPI_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A17 2000	0x4A17 4000	0x4A17 A000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A17 2004	0x4A17 4004	0x4A17 A004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A17 2018	0x4A17 4018	0x4A17 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A17 201C	0x4A17 401C	0x4A17 A01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A17 2020	0x4A17 4020	0x4A17 A020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A17 2024	0x4A17 4024	0x4A17 A024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A17 2028	0x4A17 4028	0x4A17 A028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A17 202C	0x4A17 402C	0x4A17 A02C

Table 14-451. CFG_TA Register Mapping Summary 10

Register Name	Type	Register Width (Bits)	Address Offset	MA_MPU_NTTP_FW_CFG_TARG L3_MAIN Physical Address	EMIF_OCP_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A20 B000	0x4A20 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A20 B004	0x4A20 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A20 B018	0x4A20 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A20 B01C	0x4A20 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A20 B020	0x4A20 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A20 B024	0x4A20 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A20 B028	0x4A20 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A20 B02C	0x4A20 D02C

Table 14-452. CFG_TA Register Mapping Summary 11

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM2_FW_CFG_TARG L3_MAIN Physical Address	GPMC_FW_CFG_TARG L3_MAIN Physical Address	OCMC_RAM1_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A20 F000	0x4A21 1000	0x4A21 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A20 F004	0x4A21 1004	0x4A21 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A20 F018	0x4A21 1018	0x4A21 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A20 F01C	0x4A21 101C	0x4A21 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A20 F020	0x4A21 1020	0x4A21 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A20 F024	0x4A21 1024	0x4A21 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A20 F028	0x4A21 1028	0x4A21 3028

Table 14-452. CFG_TA Register Mapping Summary 11 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM2_FW_CFG_TARG L3_MAIN Physical Address	GPMC_FW_CFG_TARG L3_MAIN Physical Address	OCMC_RAM1_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A20 F02C	0x4A21 102C	0x4A21 302C

Table 14-453. CFG_TA Register Mapping Summary 12

Register Name	Type	Register Width (Bits)	Address Offset	GPU_FW_CFG_TARG L3_MAIN Physical Address	IPU2_FW_CFG_TARG L3_MAIN Physical Address	DSS_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A21 5000	0x4A21 9000	0x4A21 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A21 5004	0x4A21 9004	0x4A21 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A21 5018	0x4A21 9018	0x4A21 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A21 501C	0x4A21 901C	0x4A21 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A21 5020	0x4A21 9020	0x4A21 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A21 5024	0x4A21 9024	0x4A21 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A21 5028	0x4A21 9028	0x4A21 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A21 502C	0x4A21 902C	0x4A21 D02C

Table 14-454. CFG_TA Register Mapping Summary 13

Register Name	Type	Register Width (Bits)	Address Offset	IVA_SL2IF_FW_CFG_TARG L3_MAIN Physical Address	IVA_CONFIG_FW_CFG_TARG L3_MAIN Physical Address	DEBUGSS_CT_TBR_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A21 F000	0x4A22 1000	0x4A22 5000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A21 F004	0x4A22 1004	0x4A22 5004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A21 F018	0x4A22 1018	0x4A22 5018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A21 F01C	0x4A22 101C	0x4A22 501C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A21 F020	0x4A22 1020	0x4A22 5020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A21 F024	0x4A22 1024	0x4A22 5024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A21 F028	0x4A22 1028	0x4A22 5028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A21 F02C	0x4A22 102C	0x4A22 502C

Table 14-455. CFG_TA Register Mapping Summary 14

Register Name	Type	Register Width (Bits)	Address Offset	L3_INSTR_FW_CFG_TARG L3_MAIN Physical Address	OCMC_RAM3_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A22 7000	0x4A22 B000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A22 7004	0x4A22 B004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A22 7018	0x4A22 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A22 701C	0x4A22 B01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A22 7020	0x4A22 B020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A22 7024	0x4A22 B024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A22 7028	0x4A22 B028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A22 702C	0x4A22 B02C

Table 14-456. PER1_TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	UART3_TARG L3_MAIN Physical Address	TIMER2_TARG L3_MAIN Physical Address	TIMER3_TARG L3_MAIN Physical Address	TIMER4_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4802 1000	0x4803 3000	0x4803 5000	0x4803 7000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4802 1004	0x4803 3004	0x4803 5004	0x4803 7004

Table 14-456. PER1_TA Register Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	UART3_TARG L3_MAIN Physical Address	TIMER2_TARG L3_MAIN Physical Address	TIMER3_TARG L3_MAIN Physical Address	TIMER4_TARG L3_MAIN Physical Address
L4_TA_CORE_L	R	32	0x0000 0018	0x4802 1018	0x4803 3018	0x4803 5018	0x4803 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4802 101C	0x4803 301C	0x4803 501C	0x4803 701C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4802 1020	0x4803 3020	0x4803 5020	0x4803 7020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4802 1024	0x4803 3024	0x4803 5024	0x4803 7024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4802 1028	0x4803 3028	0x4803 5028	0x4803 7028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4802 102C	0x4803 302C	0x4803 502C	0x4803 702C

Table 14-457. PER1_TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	TIMER9_TARG L3_MAIN Physical Address	GPIO7_TARG L3_MAIN Physical Address	GPIO8_TARG L3_MAIN Physical Address	GPIO2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4803 F000	0x4805 2000	0x4805 4000	0x4805 6000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4803 F004	0x4805 2004	0x4805 4004	0x4805 6004
L4_TA_CORE_L	R	32	0x0000 0018	0x4803 F018	0x4805 2018	0x4805 4018	0x4805 6018
L4_TA_CORE_H	R	32	0x0000 001C	0x4803 F01C	0x4805 201C	0x4805 401C	0x4805 601C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4803 F020	0x4805 2020	0x4805 4020	0x4805 6020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4803 F024	0x4805 2024	0x4805 4024	0x4805 6024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4803 F028	0x4805 2028	0x4805 4028	0x4805 6028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4803 F02C	0x4805 202C	0x4805 402C	0x4805 602C

Table 14-458. PER1_TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	GPIO3_TARG L3_MAIN Physical Address	GPIO4_TARG L3_MAIN Physical Address	GPIO5_TARG L3_MAIN Physical Address	GPIO6_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4805 8000	0x4805 A000	0x4805 C000	0x4805 E000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4805 8004	0x4805 A004	0x4805 C004	0x4805 E004
L4_TA_CORE_L	R	32	0x0000 0018	0x4805 8018	0x4805 A018	0x4805 C018	0x4805 E018
L4_TA_CORE_H	R	32	0x0000 001C	0x4805 801C	0x4805 A01C	0x4805 C01C	0x4805 E01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4805 8020	0x4805 A020	0x4805 C020	0x4805 E020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4805 8024	0x4805 A024	0x4805 C024	0x4805 E024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4805 8028	0x4805 A028	0x4805 C028	0x4805 E028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4805 802C	0x4805 A02C	0x4805 C02C	0x4805 E02C

Table 14-459. PER1_TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	I2C3_TARG L3_MAIN Physical Address	UART5_TARG L3_MAIN Physical Address	UART6_TARG L3_MAIN Physical Address	UART1_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4806 1000	0x4806 7000	0x4806 9000	0x4806 B000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4806 1004	0x4806 7004	0x4806 9004	0x4806 B004
L4_TA_CORE_L	R	32	0x0000 0018	0x4806 1018	0x4806 7018	0x4806 9018	0x4806 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4806 101C	0x4806 701C	0x4806 901C	0x4806 B01C

Table 14-459. PER1_TA Register Mapping Summary 4 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	I2C3_TARG L3_MAIN Physical Address	UART5_TARG L3_MAIN Physical Address	UART6_TARG L3_MAIN Physical Address	UART1_TARG L3_MAIN Physical Address
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4806 1020	0x4806 7020	0x4806 9020	0x4806 B020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4806 1024	0x4806 7024	0x4806 9024	0x4806 B024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4806 1028	0x4806 7028	0x4806 9028	0x4806 B028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4806 102C	0x4806 702C	0x4806 902C	0x4806 B02C

Table 14-460. PER1_TA Register Mapping Summary 5

Register Name	Type	Register Width (Bits)	Address Offset	UART2_TARG L3_MAIN Physical Address	UART4_TARG L3_MAIN Physical Address	I2C1_TARG L3_MAIN Physical Address	I2C2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4806 D000	0x4806 F000	0x4807 1000	0x4807 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4806 D004	0x4806 F004	0x4807 1004	0x4807 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4806 D018	0x4806 F018	0x4807 1018	0x4807 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4806 D01C	0x4806 F01C	0x4807 101C	0x4807 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4806 D020	0x4806 F020	0x4807 1020	0x4807 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4806 D024	0x4806 F024	0x4807 1024	0x4807 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4806 D028	0x4806 F028	0x4807 1028	0x4807 3028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4806 D02C	0x4806 F02C	0x4807 102C	0x4807 302C

Table 14-461. PER1_TA Register Mapping Summary 6

Register Name	Type	Register Width (Bits)	Address Offset	ELM_TARG L3_MAIN Physical Address	I2C4_TARG L3_MAIN Physical Address	I2C5_TARG L3_MAIN Physical Address	TIMER10_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4807 9000	0x4807 B000	0x4807 D000	0x4808 7000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4807 9004	0x4807 B004	0x4807 D004	0x4808 7004
L4_TA_CORE_L	R	32	0x0000 0018	0x4807 9018	0x4807 B018	0x4807 D018	0x4808 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4807 901C	0x4807 B01C	0x4807 D01C	0x4808 701C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4807 9020	0x4807 B020	0x4807 D020	0x4808 7020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4807 9024	0x4807 B024	0x4807 D024	0x4808 7024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4807 9028	0x4807 B028	0x4807 D028	0x4808 7028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4807 902C	0x4807 B02C	0x4807 D02C	0x4808 702C

Table 14-462. PER1_TA Register Mapping Summary 7

Register Name	Type	Register Width (Bits)	Address Offset	TIMER11_TARG L3_MAIN Physical Address	MCSPI1_TARG L3_MAIN Physical Address	MCSPI2_TARG L3_MAIN Physical Address	MMC1_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4808 9000	0x4809 9000	0x4809 B000	0x4809 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4808 9004	0x4809 9004	0x4809 B004	0x4809 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4808 9018	0x4809 9018	0x4809 B018	0x4809 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4808 901C	0x4809 901C	0x4809 B01C	0x4809 D01C

Table 14-462. PER1_TA Register Mapping Summary 7 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	TIMER11_TARG L3_MAIN Physical Address	MCSPI1_TARG L3_MAIN Physical Address	MCSPI2_TARG L3_MAIN Physical Address	MMC1_TARG L3_MAIN Physical Address
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4808 9020	0x4809 9020	0x4809 B020	0x4809 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4808 9024	0x4809 9024	0x4809 B024	0x4809 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4808 9028	0x4809 9028	0x4809 B028	0x4809 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4808 902C	0x4809 902C	0x4809 B02C	0x4809 D02C

Table 14-463. PER1_TA Register Mapping Summary 8

Register Name	Type	Register Width (Bits)	Address Offset	MMC3_TARG L3_MAIN Physical Address	HDQ1W_TARG L3_MAIN Physical Address	MMC2_TARG L3_MAIN Physical Address	MCSPI3_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x480A E000	0x480B 3000	0x480B 5000	0x480B 9000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x480A E004	0x480B 3004	0x480B 5004	0x480B 9004
L4_TA_CORE_L	R	32	0x0000 0018	0x480A E018	0x480B 3018	0x480B 5018	0x480B 9018
L4_TA_CORE_H	R	32	0x0000 001C	0x480A E01C	0x480B 301C	0x480B 501C	0x480B 901C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x480A E020	0x480B 3020	0x480B 5020	0x480B 9020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x480A E024	0x480B 3024	0x480B 5024	0x480B 9024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x480A E028	0x480B 3028	0x480B 5028	0x480B 9028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x480A E02C	0x480B 302C	0x480B 502C	0x480B 902C

Table 14-464. PER1_TA Register Mapping Summary 9

Register Name	Type	Register Width (Bits)	Address Offset	MCSPI4_TARG L3_MAIN Physical Address	MMC4_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x480B B000	0x480D 2000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x480B B004	0x480D 2004
L4_TA_CORE_L	R	32	0x0000 0018	0x480B B018	0x480D 2018
L4_TA_CORE_H	R	32	0x0000 001C	0x480B B01C	0x480D 201C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x480B B020	0x480D 2020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x480B B024	0x480D 2024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x480B B028	0x480D 2028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x480B B02C	0x480D 202C

Table 14-465. PER2_TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	UART7_TARG L3_MAIN Physical Address	UART8_TARG L3_MAIN Physical Address	UART9_TARG L3_MAIN Physical Address	MLB_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4842 1000	0x4842 3000	0x4842 5000	0x4842 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4842 1004	0x4842 3004	0x4842 5004	0x4842 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4842 1018	0x4842 3018	0x4842 5018	0x4842 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4842 101C	0x4842 301C	0x4842 501C	0x4842 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4842 1020	0x4842 3020	0x4842 5020	0x4842 D020

Table 14-465. PER2_TA Register Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	UART7_TARG L3_MAIN Physical Address	UART8_TARG L3_MAIN Physical Address	UART9_TARG L3_MAIN Physical Address	MLB_TARG L3_MAIN Physical Address
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4842 1024	0x4842 3024	0x4842 5024	0x4842 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4842 1028	0x4842 3028	0x4842 5028	0x4842 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4842 102C	0x4842 302C	0x4842 502C	0x4842 D02C

Table 14-466. PER2_TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MCASP4_DAT_TARG L3_MAIN Physical Address	MCASP5_DAT_TARG L3_MAIN Physical Address	ATL_TARG L3_MAIN Physical Address	PWM1_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4843 7000	0x4843 B000	0x4843 D000	0x4843 F000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4843 7004	0x4843 B004	0x4843 D004	0x4843 F004
L4_TA_CORE_L	R	32	0x0000 0018	0x4843 7018	0x4843 B018	0x4843 D018	0x4843 F018
L4_TA_CORE_H	R	32	0x0000 001C	0x4843 701C	0x4843 B01C	0x4843 D01C	0x4843 F01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4843 7020	0x4843 B020	0x4843 D020	0x4843 F020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4843 7024	0x4843 B024	0x4843 D024	0x4843 F024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4843 7028	0x4843 B028	0x4843 D028	0x4843 F028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4843 702C	0x4843 B02C	0x4843 D02C	0x4843 F02C

Table 14-467. PER2_TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	PWM2_TARG L3_MAIN Physical Address	PWM3_TARG L3_MAIN Physical Address	VCP1_CFG_TARG L3_MAIN Physical Address	VCP2_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4844 1000	0x4844 3000	0x4844 7000	0x4844 9000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4844 1004	0x4844 3004	0x4844 7004	0x4844 9004
L4_TA_CORE_L	R	32	0x0000 0018	0x4844 1018	0x4844 3018	0x4844 7018	0x4844 9018
L4_TA_CORE_H	R	32	0x0000 001C	0x4844 101C	0x4844 301C	0x4844 701C	0x4844 901C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4844 1020	0x4844 3020	0x4844 7020	0x4844 9020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4844 1024	0x4844 3024	0x4844 7024	0x4844 9024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4844 1028	0x4844 3028	0x4844 7028	0x4844 9028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4844 102C	0x4844 302C	0x4844 702C	0x4844 902C

Table 14-468. PER2_TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	MCASP6_DAT_TARG L3_MAIN Physical Address	MCASP7_DAT_TARG L3_MAIN Physical Address	MCASP8_DAT_TARG L3_MAIN Physical Address	MCASP1_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4844 D000	0x4845 1000	0x4845 5000	0x4846 2000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4844 D004	0x4845 1004	0x4845 5004	0x4846 2004
L4_TA_CORE_L	R	32	0x0000 0018	0x4844 D018	0x4845 1018	0x4845 5018	0x4846 2018
L4_TA_CORE_H	R	32	0x0000 001C	0x4844 D01C	0x4845 101C	0x4845 501C	0x4846 201C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4844 D020	0x4845 1020	0x4845 5020	0x4846 2020

Table 14-468. PER2_TA Register Mapping Summary 4 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP6_DAT_TARG L3_MAIN Physical Address	MCASP7_DAT_TARG L3_MAIN Physical Address	MCASP8_DAT_TARG L3_MAIN Physical Address	MCASP1_CFG_TARG L3_MAIN Physical Address
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4844 D024	0x4845 1024	0x4845 5024	0x4846 2024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4844 D028	0x4845 1028	0x4845 5028	0x4846 2028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4844 D02C	0x4845 102C	0x4845 502C	0x4846 202C

Table 14-469. PER2_TA Register Mapping Summary 5

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_CFG_TARG L3_MAIN Physical Address	MCASP3_CFG_TARG L3_MAIN Physical Address	MCASP4_CFG_TARG L3_MAIN Physical Address	MCASP5_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4846 6000	0x4846 A000	0x4846 E000	0x4847 2000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4846 6004	0x4846 A004	0x4846 E004	0x4847 2004
L4_TA_CORE_L	R	32	0x0000 0018	0x4846 6018	0x4846 A018	0x4846 E018	0x4847 2018
L4_TA_CORE_H	R	32	0x0000 001C	0x4846 601C	0x4846 A01C	0x4846 E01C	0x4847 201C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4846 6020	0x4846 A020	0x4846 E020	0x4847 2020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4846 6024	0x4846 A024	0x4846 E024	0x4847 2024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4846 6028	0x4846 A028	0x4846 E028	0x4847 2028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4846 602C	0x4846 A02C	0x4846 E02C	0x4847 202C

Table 14-470. PER2_TA Register Mapping Summary 6

Register Name	Type	Register Width (Bits)	Address Offset	MCASP6_CFG_TARG L3_MAIN Physical Address	MCASP7_CFG_TARG L3_MAIN Physical Address	MCASP8_CFG_TARG L3_MAIN Physical Address	DCAN2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4847 6000	0x4847 A000	0x4847 E000	0x4848 2000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4847 6004	0x4847 A004	0x4847 E004	0x4848 2004
L4_TA_CORE_L	R	32	0x0000 0018	0x4847 6018	0x4847 A018	0x4847 E018	0x4848 2018
L4_TA_CORE_H	R	32	0x0000 001C	0x4847 601C	0x4847 A01C	0x4847 E01C	0x4848 201C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4847 6020	0x4847 A020	0x4847 E020	0x4848 2020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4847 6024	0x4847 A024	0x4847 E024	0x4848 2024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4847 6028	0x4847 A028	0x4847 E028	0x4848 2028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4847 602C	0x4847 A02C	0x4847 E02C	0x4848 202C

Table 14-471. PER2_TA Register Mapping Summary 7

Register Name	Type	Register Width (Bits)	Address Offset	GMAC_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4848 8000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4848 8004

Table 14-471. PER2_TA Register Mapping Summary 7 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	GMAC_TARG L3_MAIN Physical Address
L4_TA_CORE_L	R	32	0x0000 0018	0x4848 8018
L4_TA_CORE_H	R	32	0x0000 001C	0x4848 801C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4848 8020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4848 8024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4848 8028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4848 802C

Table 14-472. PER3_TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	MBX13_TARG L3_MAIN Physical Address	OCMC_RAM1_TARG L3_MAIN Physical Address	OCMC_RAM2_TARG L3_MAIN Physical Address	OCMC_RAM3_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4880 3000	0x4880 5000	0x4880 B000	0x4881 1000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4880 3004	0x4880 5004	0x4880 B004	0x4881 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4880 3018	0x4880 5018	0x4880 B018	0x4881 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4880 301C	0x4880 501C	0x4880 B01C	0x4881 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4880 3020	0x4880 5020	0x4880 B020	0x4881 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4880 3024	0x4880 5024	0x4880 B024	0x4881 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4880 3028	0x4880 5028	0x4880 B028	0x4881 3028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4880 302C	0x4880 502C	0x4880 B02C	0x4881 302C

Table 14-473. PER3_TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MMU1_TARG L3_MAIN Physical Address	MMU2_TARG L3_MAIN Physical Address	TIMER5_TARG L3_MAIN Physical Address	TIMER6_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4881 D000	0x4881 F000	0x4882 1000	0x4882 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4881 D004	0x4881 F004	0x4882 1004	0x4882 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4881 D018	0x4881 F018	0x4882 1018	0x4882 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4881 D01C	0x4881 F01C	0x4882 101C	0x4882 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4881 D020	0x4881 F020	0x4882 1020	0x4882 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4881 D024	0x4881 F024	0x4882 1024	0x4882 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4881 D028	0x4881 F028	0x4882 1028	0x4882 3028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4881 D02C	0x4881 F02C	0x4882 102C	0x4882 302C

Table 14-474. PER3_TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	TIMER7_TARG L3_MAIN Physical Address	TIMER8_TARG L3_MAIN Physical Address	TIMER13_TARG L3_MAIN Physical Address	TIMER14_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4882 5000	0x4882 7000	0x4882 9000	0x4882 B000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4882 5004	0x4882 7004	0x4882 9004	0x4882 B004
L4_TA_CORE_L	R	32	0x0000 0018	0x4882 5018	0x4882 7018	0x4882 9018	0x4882 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4882 501C	0x4882 701C	0x4882 901C	0x4882 B01C

Table 14-474. PER3_TA Register Mapping Summary 3 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	TIMER7_TARG L3_MAIN Physical Address	TIMER8_TARG L3_MAIN Physical Address	TIMER13_TARG L3_MAIN Physical Address	TIMER14_TARG L3_MAIN Physical Address
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4882 5020	0x4882 7020	0x4882 9020	0x4882 B020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4882 5024	0x4882 7024	0x4882 9024	0x4882 B024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4882 5028	0x4882 7028	0x4882 9028	0x4882 B028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4882 502C	0x4882 702C	0x4882 902C	0x4882 B02C

Table 14-475. PER3_TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	TIMER15_TARG L3_MAIN Physical Address	TIMER16_TARG L3_MAIN Physical Address	RTC_TARG L3_MAIN Physical Address	MBX2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4882 D000	0x4882 F000	0x4883 9000	0x4883 B000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4882 D004	0x4882 F004	0x4883 9004	0x4883 B004
L4_TA_CORE_L	R	32	0x0000 0018	0x4882 D018	0x4882 F018	0x4883 9018	0x4883 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4882 D01C	0x4882 F01C	0x4883 901C	0x4883 B01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4882 D020	0x4882 F020	0x4883 9020	0x4883 B020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4882 D024	0x4882 F024	0x4883 9024	0x4883 B024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4882 D028	0x4882 F028	0x4883 9028	0x4883 B028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4882 D02C	0x4882 F02C	0x4883 902C	0x4883 B02C

Table 14-476. PER3_TA Register Mapping Summary 5

Register Name	Type	Register Width (Bits)	Address Offset	MBX3_TARG L3_MAIN Physical Address	MBX4_TARG L3_MAIN Physical Address	MBX5_TARG L3_MAIN Physical Address	MBX6_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4883 D000	0x4883 F000	0x4884 1000	0x4884 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4883 D004	0x4883 F004	0x4884 1004	0x4884 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4883 D018	0x4883 F018	0x4884 1018	0x4884 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4883 D01C	0x4883 F01C	0x4884 101C	0x4884 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4883 D020	0x4883 F020	0x4884 1020	0x4884 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4883 D024	0x4883 F024	0x4884 1024	0x4884 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4883 D028	0x4883 F028	0x4884 1028	0x4884 3028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4883 D02C	0x4883 F02C	0x4884 102C	0x4884 302C

Table 14-477. PER3_TA Register Mapping Summary 6

Register Name	Type	Register Width (Bits)	Address Offset	MBX7_TARG L3_MAIN Physical Address	MBX8_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4884 5000	0x4884 7000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4884 5004	0x4884 7004
L4_TA_CORE_L	R	32	0x0000 0018	0x4884 5018	0x4884 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4884 501C	0x4884 701C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4884 5020	0x4884 7020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4884 5024	0x4884 7024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4884 5028	0x4884 7028

Table 14-477. PER3_TA Register Mapping Summary 6 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MBX7_TARG L3_MAIN Physical Address	MBX8_TARG L3_MAIN Physical Address
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4884 502C	0x4884 702C

Table 14-478. PER3_TA Register Mapping Summary 7

Register Name	Type	Register Width (Bits)	Address Offset	MBX9_TARG L3_MAIN Physical Address	MBX10_TARG L3_MAIN Physical Address	MBX11_TARG L3_MAIN Physical Address	MBX12_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4885 F000	0x4886 1000	0x4886 3000	0x4886 5000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4885 F004	0x4886 1004	0x4886 3004	0x4886 5004
L4_TA_CORE_L	R	32	0x0000 0018	0x4885 F018	0x4886 1018	0x4886 3018	0x4886 5018
L4_TA_CORE_H	R	32	0x0000 001C	0x4885 F01C	0x4886 101C	0x4886 301C	0x4886 501C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4885 F020	0x4886 1020	0x4886 3020	0x4886 5020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4885 F024	0x4886 1024	0x4886 3024	0x4886 5024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4885 F028	0x4886 1028	0x4886 3028	0x4886 5028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4885 F02C	0x4886 102C	0x4886 302C	0x4886 502C

Table 14-479. PER3_TA Register Mapping Summary 8

Register Name	Type	Register Width (Bits)	Address Offset	USB1_CFG_TARG L3_MAIN Physical Address	USB2_CFG_TARG L3_MAIN Physical Address	USB3_CFG_TARG L3_MAIN Physical Address	USB4_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x488A 0000	0x488E 0000	0x4892 0000	0x4896 0000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x488A 0004	0x488E 0004	0x4892 0004	0x4896 0004
L4_TA_CORE_L	R	32	0x0000 0018	0x488A 0018	0x488E 0018	0x4892 0018	0x4896 0018
L4_TA_CORE_H	R	32	0x0000 001C	0x488A 001C	0x488E 001C	0x4892 001C	0x4896 001C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x488A 0020	0x488E 0020	0x4892 0020	0x4896 0020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x488A 0024	0x488E 0024	0x4892 0024	0x4896 0024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x488A 0028	0x488E 0028	0x4892 0028	0x4896 0028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x488A 002C	0x488E 002C	0x4892 002C	0x4896 002C

Table 14-480. PER3_TA Register Mapping Summary 9

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_TARG L3_MAIN Physical Address	VIP2_TARG L3_MAIN Physical Address	VIP3_TARG L3_MAIN Physical Address	VPE_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4898 0000	0x489A 0000	0x489C 0000	0x489E 0000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4898 0004	0x489A 0004	0x489C 0004	0x489E 0004
L4_TA_CORE_L	R	32	0x0000 0018	0x4898 0018	0x489A 0018	0x489C 0018	0x489E 0018

Table 14-480. PER3_TA Register Mapping Summary 9 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP1_TARG L3_MAIN Physical Address	VIP2_TARG L3_MAIN Physical Address	VIP3_TARG L3_MAIN Physical Address	VPE_TARG L3_MAIN Physical Address
L4_TA_CORE_H	R	32	0x0000 001C	0x4898 001C	0x489A 001C	0x489C 001C	0x489E 001C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4898 0020	0x489A 0020	0x489C 0020	0x489E 0020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4898 0024	0x489A 0024	0x489C 0024	0x489E 0024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4898 0028	0x489A 0028	0x489C 0028	0x489E 0028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4898 002C	0x489A 002C	0x489C 002C	0x489E 002C

Table 14-481. WKUP_TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	COUNTER_32K_TARG L3_MAIN Physical Address	PRM_TARG L3_MAIN Physical Address	CTRL_MODULE_WKUP_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4AE0 5000	0x4AE0 8000	0x4AE0 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4AE0 5004	0x4AE0 8004	0x4AE0 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4AE0 5018	0x4AE0 8018	0x4AE0 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4AE0 501C	0x4AE0 801C	0x4AE0 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4AE0 5020	0x4AE0 8020	0x4AE0 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4AE0 5024	0x4AE0 8024	0x4AE0 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4AE0 5028	0x4AE0 8028	0x4AE0 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4AE0 502C	0x4AE0 802C	0x4AE0 D02C

Table 14-482. WKUP_TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	GPIO1_TARG L3_MAIN Physical Address	WD_TIMER2_TARG L3_MAIN Physical Address	TIMER1_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4AE1 1000	0x4AE1 5000	0x4AE1 9000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4AE1 1004	0x4AE1 5004	0x4AE1 9004
L4_TA_CORE_L	R	32	0x0000 0018	0x4AE1 1018	0x4AE1 5018	0x4AE1 9018
L4_TA_CORE_H	R	32	0x0000 001C	0x4AE1 101C	0x4AE1 501C	0x4AE1 901C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4AE1 1020	0x4AE1 5020	0x4AE1 9020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4AE1 1024	0x4AE1 5024	0x4AE1 9024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4AE1 1028	0x4AE1 5028	0x4AE1 9028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4AE1 102C	0x4AE1 502C	0x4AE1 902C

Table 14-483. WKUP_TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	KBD_TARG L3_MAIN Physical Address	TIMER12_TARG L3_MAIN Physical Address	UART10_TARG L3_MAIN Physical Address	DCAN1_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4AE1 D000	0x4AE2 1000	0x4AE2 C000	0x4AE3 E000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4AE1 D004	0x4AE2 1004	0x4AE2 C004	0x4AE3 E004
L4_TA_CORE_L	R	32	0x0000 0018	0x4AE1 D018	0x4AE2 1018	0x4AE2 C018	0x4AE3 E018
L4_TA_CORE_H	R	32	0x0000 001C	0x4AE1 D01C	0x4AE2 101C	0x4AE2 C01C	0x4AE3 E01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4AE1 D020	0x4AE2 1020	0x4AE2 C020	0x4AE3 E020

Table 14-483. WKUP_TA Register Mapping Summary 3 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	KBD_TARG L3_MAIN Physical Address	TIMER12_TARG L3_MAIN Physical Address	UART10_TARG L3_MAIN Physical Address	DCAN1_TARG L3_MAIN Physical Address
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4AE1 D024	0x4AE2 1024	0x4AE2 C024	0x4AE3 E024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4AE1 D028	0x4AE2 1028	0x4AE2 C028	0x4AE3 E028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4AE1 D02C	0x4AE2 102C	0x4AE2 C02C	0x4AE3 E02C

14.3.5.3.2 L4 Target Agent (L4 TA) Register Description

Table 14-484 through Table 14-498 describe the L4 TA registers.

Table 14-484. L4_TA_COMPONENT_H

Address Offset	0x0000 0000		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483
Description	Contains a component code and revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 000

Table 14-485. Register Call Summary for Register L4_TA_COMPONENT_H

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0] [1] [2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [42] [43] [44]

Table 14-486. L4_TA_COMPONENT_L

Address Offset	0x0000 0004		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483
Description	Contains a component code and revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code.	R	See (1).
15:0	REV	Component revision code.	R	See (1).

(1) TI Internal Data

Table 14-487. Register Call Summary for Register L4_TA_COMPONENT_L

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0] [1] [2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [42] [43] [44]

Table 14-488. L4_TA_CORE_L

Address Offset	0x0000 0018		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483

Table 14-488. L4_TA_CORE_L (continued)

Description	Contains a component code and revision.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CODE																CORE_REV															

Bits	Field Name	Description	Type	Reset
31:16	CORE_CODE	Interconnect core code	R	See (1).
15:0	CORE_REV	Component revision code code	R	See (1).

(1) TI Internal Data

Table 14-489. Register Call Summary for Register L4_TA_CORE_L

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0] [1] [2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [42] [43] [44]

Table 14-490. L4_TA_CORE_H

Address Offset	0x0000 001C		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483
Description	Contains a component code and revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VENDOR_CODE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	VENDOR_CODE	Vendor revision core code	R	See (1).

(1) TI Internal Data

Table 14-491. Register Call Summary for Register L4_TA_CORE_H

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0] [1] [2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [42] [43] [44]

Table 14-492. L4_TA_AGENT_CONTROL_L

Address Offset	0x0000 0020		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483
Description	Enable error reporting		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							SERROR_REP	RESERVED											REQ_TIMEOUT			RESERVED									OCP_RESET

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	SERROR_REP	Enable logging of error	R	0x0

Bits	Field Name	Description	Type	Reset
23:11	RESERVED	Read returns 0.	R	0x0
10:8	REQ_TIMEOUT	Time-out Bound. Values are: 0 - No time-out 1 - 1x base cycles. 2 - 4x base cycles. 3 - 16x base cycles. 4 - 64x base cycles.	RW	0x0
7:1	RESERVED	Read returns 0.	R	0x00
0	OCP_RESET	The OCP_RESET field controls the OCP reset signal to the attached core. Setting this bit clears any pending transfers and resets the OCP interface. The bit must be cleared to deassert the OCP reset signal. When the software reset feature is available on a target agent, the target agent OCP must also have a reset signal directed to the target core.	RW	0

Table 14-493. Register Call Summary for Register L4_TA_AGENT_CONTROL_L

L4 Interconnects

- Time-Out: [0] [1] [2] [3] [4]
- Error Recovery: [5] [6]
- Operational Modes Configuration: [7] [8] [9] [10]
- L4 Target Agent (L4 TA) Register Summary: [11] [12] [13] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [53] [54] [55]

Table 14-494. L4_TA_AGENT_CONTROL_H

Address Offset	0x0000 0024		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483
Description	Enable clock power management		
Type	R		

31–10	9	8	7–0
RESERVED	AUTO_WAKEUP_RESP_CODE	EXT_CLOCK	RESERVED

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Read returns 0.	R	0x000000
9	AUTO_WAKEUP_RESP_CODE		R	0
8	EXT_CLOCK	When set to 1, the ext_clk_off_i signal on a target agent indicates when the target agent should shut off.	R	0
7:0	RESERVED	Read returns 0.	R	0x00

Table 14-495. Register Call Summary for Register L4_TA_AGENT_CONTROL_H

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0] [1] [2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [42] [43] [44]

Table 14-496. L4_TA_AGENT_STATUS_L

Address Offset	0x0000 0028		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483
Description	Error reporting		
Type	R		

31:25	24	23:9	8	7:1	0
RESERVED	SERROR	RESERVED	REQ_TIMEOUT	RESERVED	OCP_RESET

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	SERROR	Value of OCP SError signal	R	0
23:9	RESERVED	Read returns 0.	R	0x0000
8	REQ_TIMEOUT	Time-out status: 0x0: No request time-out 0x1: A request time-out has occurred	R 1toCLR	0
7:1	RESERVED	Read returns 0.	R	0x00
0	OCP_RESET	L3 Reset	R	0

Table 14-497. Register Call Summary for Register L4_TA_AGENT_STATUS_L

L4 Interconnects

- Time-Out: [0] [1]
- Error Recovery: [2]
- Operational Modes Configuration: [3]
- L4 Target Agent (L4 TA) Register Summary: [4] [5] [6] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [46] [47] [48]

Table 14-498. L4_TA_AGENT_STATUS_H

Address Offset	0x0000 002C		
Physical Address	See Table 14-442 to Table 14-483	**Instance**	See Table 14-442 to Table 14-483
Description	Error reporting		
Type	R		

31:0
RESERVED

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 000

Table 14-499. Register Call Summary for Register L4_TA_AGENT_STATUS_H

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0] [1] [2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [42] [43] [44]

14.3.5.4 L4 Link Agent (L4 LA)

14.3.5.4.1 L4 Link Agent (L4 LA) Register Summary

Table 14-500 summarizes the L4 LA register mapping.

Table 14-500. LA Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PER1_LA L3_MAIN Physical Address	PER2_LA L3_MAIN Physical Address	PER3_LA L3_MAIN Physical Address
L4_LA_COMPONENT_L	R	32	0x0000 0000	0x4800 0800	0x4840 0800	0x4880 0800
L4_LA_COMPONENT_H	R	32	0x0000 0004	0x4800 0804	0x4840 0804	0x4880 0804
L4_LA_NETWORK_L	R	32	0x0000 0010	0x4800 0810	0x4840 0810	0x4880 0810
L4_LA_NETWORK_H	R	32	0x0000 0014	0x4800 0814	0x4840 0814	0x4880 0814
L4_LA_INITIATOR_INFO_L	R	32	0x0000 0018	0x4800 0818	0x4840 0818	0x4880 0818
L4_LA_INITIATOR_INFO_H	R	32	0x0000 001C	0x4800 081C	0x4840 081C	0x4880 081C
L4_LA_NETWORK_CONTROL_L	RW	32	0x0000 0020	0x4800 0820	0x4840 0820	0x4880 0820
L4_LA_NETWORK_CONTROL_H	RW	32	0x0000 0024	0x4800 0824	0x4840 0824	0x4880 0824
L4_LA_FLAG_MASK_j_L (1)	RW	32	0x0000 0100 + (0x20*j)	0x4800 0900 + (0x20*j)	0x4840 0900 + (0x20*j)	0x4880 0900 + (0x20*j)
L4_LA_FLAG_MASK_j_H (1)	RW	32	0x0000 0104 + (0x20*j)	0x4800 0904 + (0x20*j)	0x4840 0904 + (0x20*j)	0x4880 0904 + (0x20*j)
L4_LA_FLAG_STATUS_j_L (1)	R	32	0x0000 0110 + (0x20*j)	0x4800 0910 + (0x20*j)	0x4840 0910 + (0x20*j)	0x4880 0910 + (0x20*j)
L4_LA_FLAG_STATUS_j_H (1)	R	32	0x0000 0114 + (0x20*j)	0x4800 0914 + (0x20*j)	0x4840 0914 + (0x20*j)	0x4880 0914 + (0x20*j)

(1) j = 0 to 1 for PER1_LA
j = 0 to 1 for PER2_LA
j = 0 to 1 for PER3_LA

Table 14-501. LA Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CFG_LA L3_MAIN Physical Address	WKUP_LA L3_MAIN Physical Address
L4_LA_COMPONENT_L	R	32	0x0000 0000	0x4A00 0800	0x4AE0 0800
L4_LA_COMPONENT_H	R	32	0x0000 0004	0x4A00 0804	0x4AE0 0804
L4_LA_NETWORK_L	R	32	0x0000 0010	0x4A00 0810	0x4AE0 0810
L4_LA_NETWORK_H	R	32	0x0000 0014	0x4A00 0814	0x4AE0 0814
L4_LA_INITIATOR_INFO_L	R	32	0x0000 0018	0x4A00 0818	0x4AE0 0818
L4_LA_INITIATOR_INFO_H	R	32	0x0000 001C	0x4A00 081C	0x4AE0 081C
L4_LA_NETWORK_CONTROL_L	RW	32	0x0000 0020	0x4A00 0820	0x4AE0 0820
L4_LA_NETWORK_CONTROL_H	RW	32	0x0000 0024	0x4A00 0824	0x4AE0 0824

14.3.5.4.2 L4 Link Agent (L4 LA) Register Description

Table 14-502 through Table 14-527 describe the L4 LA registers.

Table 14-502. L4_LA_COMPONENT_L

Address Offset	0x0000 0000

Table 14-502. L4_LA_COMPONENT_L (continued)

Physical Address	0x4800 0800 0x4840 0800 0x4880 0800 0x4A00 0800 0x4AE0 0800	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Contain a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31–16	15–0
CODE	REV

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code.	R	See (1).
15:0	REV	Component revision code.	R	See (1).

(1) TI Internal Data

Table 14-503. Register Call Summary for Register L4_LA_COMPONENT_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0] [1]

Table 14-504. L4_LA_COMPONENT_H

Address Offset	0x0000 0004		
Physical Address	0x4800 0804 0x4840 0804 0x4880 0804 0x4A00 0804 0x4AE0 0804	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Contain a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31–0
RESERVED

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

Table 14-505. Register Call Summary for Register L4_LA_COMPONENT_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0] [1]

Table 14-506. L4_LA_NETWORK_L

Address Offset	0x0000 0010		
Physical Address	0x4800 0810 0x4840 0810 0x4880 0810 0x4A00 0810 0x4AE0 0810	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Identify the interconnect		
Type	R		

31–0
RESERVED

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 0000

Table 14-507. Register Call Summary for Register L4_LA_NETWORK_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0] [1]

Table 14-508. L4_LA_NETWORK_H

Address Offset	0x0000 0014		
Physical Address	0x4800 0814 0x4840 0814 0x4880 0814 0x4A00 0814 0x4AE0 0814	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Identify the interconnect		
Type	R		

31–0
ID

Bits	Field Name	Description	Type	Reset
31:0	ID	The ID field uniquely identifies this interconnect.	R	0x00000000

Table 14-509. Register Call Summary for Register L4_LA_NETWORK_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0] [1]

Table 14-510. L4_LA_INITIATOR_INFO_L

Address Offset	0x0000 0018		
Physical Address	0x4800 0818 0x4840 0818 0x4880 0818 0x4A00 0818 0x4AE0 0818	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Contain initiator subsystem information.		
Type	R		

31–28	27–24	23–16	15–4	3–0
RESERVED	PROT_GROUPS	NUMBER_REGIONS	RESERVED	SEGMENTS

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:24	PROT_GROUPS	Number of protection group of in the current L4 0x0: No protection group 0x1: 1 protection group 0x2: 2 protection groups .... 0x8: 8 protection groups 0x9 to 0xF: Reserved	R	see Table 14-512

Bits	Field Name	Description	Type	Reset
23:16	NUMBER_REGIONS	Number of regions in the current L4 0x0: Reserved 0x1: 1 region 0x2: 2 regions .... Max regions +1 to 0xFF: Reserved, maximum regions is listed in Table 14-512	R	see Table 14-512
15:4	RESERVED	Read returns 0.	R	0x000
3:0	SEGMENTS	Number of segments in the current L4 0x0: Reserved 0x1: 1 segment 0x2: 2 segments .... 0x8: 8 segments	R	see Table 14-512

Table 14-511. Register Call Summary for Register L4_LA_INITIATOR_INFO_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0] [1]

Table 14-512. Reset value for L4_LA_INITIATOR_INFO_L

Field Name	L4 PER1	L4 PER2	L4 PER3	L4 CFG	L4 WKUP
PROT_GROUPS	0x8	0x8	0x8	0x8	0x8
NUMBER_REGIONS	0x55	0x3F	0x61	0x81	0x2C
SEGMENTS	0x2	0x1	0x1	0x3	0x4

Table 14-513. L4_LA_INITIATOR_INFO_H

Address Offset	0x0000 001C		
Physical Address	0x4800 081C 0x4840 081C 0x4880 081C 0x4A00 081C 0x4AE0 081C	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Contain initiator subsystem information.		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="13">RESERVED</td><td colspan="3">THREADS</td><td>RESERVED</td><td colspan="3">CONNID_WIDTH</td><td>RESERVED</td><td colspan="3">BYTE_DATA_WIDTH_EXP</td><td colspan="2">RESERVED</td><td colspan="6">ADDR_WIDTH</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Read returns 0.	R	0x0000
18:16	THREADS	The THREADS field specifies the number of initiator threads connected to the interconnect. The field contains read-only configuration information for the initiator subsystem.	R	see Table 14-515
15	RESERVED	Read returns 0.	R	0
14:12	CONNID_WIDTH	The initiator subsystem ConnID width. The CONNID_WIDTH field contains read-only configuration information for the initiator subsystem.	R	see Table 14-515
11	RESERVED	Read returns 0.	R	0

Bits	Field Name	Description	Type	Reset
10:8	BYTE_DATA_WIDTH_EXP	This field specifies the initiator subsystem data width.The BYTE_DATA_WIDTH_EXP field contains read-only configuration information for the initiator subsystem. 0x1: 16-bit data width is specified 0x2: 32-bit data width is specified	R	see Table 14-515
7:6	RESERVED	Read returns 0.	R	0x0
5:0	ADDR_WIDTH	This field specifies the initiator subsystem address width. The ADDR_WIDTH field contains read-only configuration information for the initiator subsystem.	R	see Table 14-515

Table 14-514. Register Call Summary for Register L4_LA_INITIATOR_INFO_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0] [1]

Table 14-515. Reset value for L4_LA_INITIATOR_INFO_H

Field Name	L4 PER1	L4 PER2	L4 PER3	L4 CFG	L4 WKUP
THREADS	0x4	0x3	0x3	0x1	0x1
CONNID_WIDTH	0x4	0x4	0x5	0x4	0x4
BYTE_DATA_WIDTH_EXP	0x2	0x2	0x2	0x2	0x2
ADDR_WIDTH	0x18	0x18	0x18	0x18	0x15

Table 14-516. L4_LA_NETWORK_CONTROL_L

Address Offset	0x0000 0020		
Physical Address	0x4800 0820 0x4840 0820 0x4880 0820 0x4A00 0820 0x4AE0 0820	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Control interconnect minimum timeout values.		
Type	RW		

31:11	10:8	7:0
RESERVED	TIMEOUT_BASE	RESERVED

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Read returns 0.	R	0x000000
10:8	TIMEOUT_BASE	The TIMEOUT_BASE field indicates the time-out period (that is, base cycles) for the highest frequency time-base signal sent from the L4 initiator subsystem to all target agents that have time-out enabled. Values for the field are: 0 - Time-out disabled 1 - L4 interconnect clock cycles divided by 64 2 - L4 interconnect clock cycles divided by 256 3 - L4 interconnect clock cycles divided by 1024 4 - L4 interconnect clock cycles divided by 4096	RW	0x4
7:0	RESERVED	Read returns 0.	R	0x00

Table 14-517. Register Call Summary for Register L4_LA_NETWORK_CONTROL_L

L4 Interconnects
- Time-Out: [0] [1] [2]
- Operational Modes Configuration: [3] [4]
- L4 Link Agent (L4 LA) Register Summary: [5] [6]

Table 14-518. L4_LA_NETWORK_CONTROL_H

Address Offset	0x0000 0024		
Physical Address	0x4800 0824 0x4840 0824 0x4880 0824 0x4A00 0824 0x4AE0 0824	**Instance**	PER1_LA PER2_LA PER3_LA CFG_LA WKUP_LA
Description	Control interconnect global power control		
Type	RW		

31:25	24	23:21	20	19:9	8	7:0
RESERVED	CLOCK_GATE_DISABLE	RESERVED	THREAD0_PRI	RESERVED	EXT_CLOCK	RESERVED

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	CLOCK_GATE_DISABLE	When set to 1 this field disables all clock gating.	RW	0
23:21	RESERVED	Read returns 0.	R	0x0
20	THREAD0_PRI	Sets thread priority. If the field is set to 0, the default, all initiator threads are treated the same. Setting the THREAD0_PRI field to 1 assigns a higher arbitration priority to thread 0 of the first initiator OCP interface. To avoid starvation, arbitration is imposed by the initiator subsystem. When multiple requests from different initiator threads are dispatched to targets simultaneously, the oldest request is dispatched first. If thread 0 is assigned a higher priority, a request on thread 0 always wins arbitration. Assigning thread 0 of the first initiator OCP the highest priority on a request or response can result in the starvation of other threads.	R	1
19:9	RESERVED	Read returns 0.	R	0x000
8	EXT_CLOCK	Global external clock control. When set to 1, the ext_clk_off_i signal on the initiator subsystem instructs the entire L4 to shut off.	R	1
7:0	RESERVED	Read returns 0.	R	0x00

Table 14-519. Register Call Summary for Register L4_LA_NETWORK_CONTROL_H

L4 Interconnects
- L4 Link Agent (L4 LA) Register Summary: [0] [1]

Table 14-520. L4_LA_FLAG_MASK_j_L

Address Offset	0x0000 0100 + (0x20*j)

Table 14-520. L4_LA_FLAG_MASK_j_L (continued)

Physical Address	0x4800 0900 + (0x20*j) 0x4840 0900 + (0x20*j) 0x4880 0900 + (0x20*j)	**Instance**	PER1_LA PER2_LA PER3_LA
Description	Mask of composite sideband flag(0)		
Type	RW		

31–4	3–0
RESERVED	MASK

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Read returns 0	R	0x0000 0000
3:0	MASK	Number of input sideband signals	RW	0xF

Table 14-521. Register Call Summary for Register L4_LA_FLAG_MASK_j_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0]

Table 14-522. Reset Value for L4_LA_FLAG_MASK_j_L

Initiator	Bit Field (MASK)	Reset
L4_PER1	[3:0]	0xF
L4_PER2	[2:0]	0x7
L4_PER3	[2:0]	0x7

Table 14-523. L4_LA_FLAG_MASK_j_H

Address Offset	0x0000 0104 + (0x20*j)		
Physical Address	0x4800 0904 + (0x20*j) 0x4840 0904 + (0x20*j) 0x4880 0904 + (0x20*j)	**Instance**	PER1_LA PER2_LA PER3_LA
Description	Status of composite sideband flag(0)		
Type	R		

31–0
RESERVED

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 0000

Table 14-524. Register Call Summary for Register L4_LA_FLAG_MASK_j_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0]

Table 14-525. L4_LA_FLAG_STATUS_j_L

Address Offset	0x0000 0110 + (0x20*j)		
Physical Address	0x4800 0910 + (0x20*j) 0x4840 0910 + (0x20*j) 0x4880 0910 + (0x20*j)	**Instance**	PER1_LA PER2_LA PER3_LA
Description	Mask of composite sideband flag(1)		
Type	RW		

31–4	3–0
RESERVED	STATUS

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Read returns 0	R	0x0000 0000
3:0	STATUS	Status of input sideband signals	RW	0x0

Table 14-526. Register Call Summary for Register L4_LA_FLAG_STATUS_j_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0]

Table 14-527. L4_LA_FLAG_STATUS_j_H

Address Offset	0x0000 0114 + (0x20*j)		
Physical Address	0x4800 0914 + (0x20*j) 0x4840 0914 + (0x20*j) 0x4880 0914 + (0x20*j)	**Instance**	PER1_LA PER2_LA PER3_LA
Description	Status of composite sideband flag(1)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x000 0000 0000 0000

Table 14-528. Register Call Summary for Register L4_LA_FLAG_STATUS_j_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0]

14.3.5.5 L4 Address Protection (L4 AP)

14.3.5.5.1 L4 Address Protection (L4 AP) Register Summary

Table 14-529 and Table 14-530 summarizes the L4 AP register mapping.

Table 14-529. L4 AP Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	PER1_AP Physical Address	PER2_AP Physical Address	PER3_AP Physical Address
L4_AP_COMPONENT_L	R	32	0x0000 0000	0x4800 0000	0x4840 0000	0x4880 0000
L4_AP_COMPONENT_H	R	32	0x0000 0004	0x4800 0004	0x4840 0004	0x4880 0004
L4_AP_SEGMENT_i_L (1)	RW	32	0x0000 0100 + (0x08*i)	0x4800 0100 + (0x08*i)	0x4840 0100 + (0x08*i)	0x4880 0100 + (0x08*i)
L4_AP_SEGMENT_i_H (1)	RW	32	0x0000 0104 + (0x08*i)	0x4800 0104 + (0x08*i)	0x4840 0104 + (0x08*i)	0x4880 0104 + (0x08*i)
L4_AP_PROT_GROUP_MEMBERS_k_L (2)	R	32	0x0000 0200 + (0x08*k)	0x4800 0200 + (0x08*k)	0x4840 0200 + (0x08*k)	0x4880 0200 + (0x08*k)
L4_AP_PROT_GROUP_MEMBERS_k_H (2)	R	32	0x0000 0204 + (0x08*k)	0x4800 0204 + (0x08*k)	0x4840 0204 + (0x08*k)	0x4880 0204 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_L (2)	R	32	0x0000 0280 + (0x08*k)	0x4800 0280 + (0x08*k)	0x4840 0280 + (0x08*k)	0x4880 0280 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_H (2)	R	32	0x0000 0284 + (0x08*k)	0x4800 0284 + (0x08*k)	0x4840 0284 + (0x08*k)	0x4880 0284 + (0x08*k)
L4_AP_REGION_l_L (3)	RW	32	0x0000 0300 + (0x08*l)	0x4800 0300 + (0x08*l)	0x4840 0300 + (0x08*l)	0x4880 0300 + (0x08*l)

Table 14-529. L4 AP Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PER1_AP Physical Address	PER2_AP Physical Address	PER3_AP Physical Address
L4_AP_REGION_l_H (3)	RW	32	0x0000 0304 + (0x08*l)	0x4800 0304 + (0x08*l)	0x4840 0304 + (0x08*l)	0x4880 0304 + (0x08*l)

(1) i = 0 to 1 for PER1_AP
i = 0 for PER2_AP
i = 0 for PER3_AP
(2) k = 0 to 7 for PER1_AP
k = 0 to 7 for PER2_AP
k = 0 to 7 for PER3_AP
(3) l = 0 to 84 for PER1_AP
l = 0 to 56 for PER2_AP
l = 0 to 96 for PER3_AP

Table 14-530. L4 AP Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CFG_AP Physical Address	WKUP_AP Physical Address
L4_AP_COMPONENT_L	R	32	0x0000 0000	0x4A00 0000	0x4AE0 0000
L4_AP_COMPONENT_H	R	32	0x0000 0004	0x4A00 0004	0x4AE0 0004
L4_AP_SEGMENT_i_L (1)	RW	32	0x0000 0100 + (0x08*i)	0x4A00 0100 + (0x08*i)	0x4AE0 0100 + (0x08*i)
L4_AP_SEGMENT_i_H (1)	RW	32	0x0000 0104 + (0x08*i)	0x4A00 0104 + (0x08*i)	0x4AE0 0104 + (0x08*i)
L4_AP_PROT_GROUP_MEMBERS_k_L (2)	R	32	0x0000 0200 + (0x08*k)	0x4A00 0200 + (0x08*k)	0x4AE0 0200 + (0x08*k)
L4_AP_PROT_GROUP_MEMBERS_k_H (2)	R	32	0x0000 0204 + (0x08*k)	0x4A00 0204 + (0x08*k)	0x4AE0 0204 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_L (2)	R	32	0x0000 0280 + (0x08*k)	0x4A00 0280 + (0x08*k)	0x4AE0 0280 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_H (2)	R	32	0x0000 0284 + (0x08*k)	0x4A00 0284 + (0x08*k)	0x4AE0 0284 + (0x08*k)
L4_AP_REGION_l_L (3)	RW	32	0x0000 0300 + (0x08*l)	0x4A00 0300 + (0x08*l)	0x4AE0 0300 + (0x08*l)
L4_AP_REGION_l_H (3)	RW	32	0x0000 0304 + (0x08*l)	0x4A00 0304 + (0x08*l)	0x4AE0 0304 + (0x08*l)

14.3.5.5.2 L4 Address Protection (L4 AP) Register Description

Table 14-531 through Table 14-556 describe the L4 AP registers.

Table 14-531. L4_AP_COMPONENT_L

Address Offset	0x000		
Physical Address	0x4800 0 000 0x4840 0 000 0x4880 0 000 0x4A00 0 000 0x4AE0 0 000	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Contains a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code	R	See (1).

Bits	Field Name	Description	Type	Reset
15:0	REV	Component revision code	R [2]	See [1].

(1) TI Internal Data
(2) For L4_PER1 and L4_WKUP, when k = 2 to 7 the access type of CONNID_BIT_VECTOR is RW.
For L4_PER2, L4_PER3 and L4_CFG, when k = 1 to 7 the access type of CONNID_BIT_VECTOR is RW.

Table 14-532. Register Call Summary for Register L4_AP_COMPONENT_L

L4 Interconnects

- L4 Address Protection (L4 AP) Register Summary: [0] [1]

Table 14-533. L4_AP_COMPONENT_H

Address Offset	0x004		
Physical Address	0x4800 0004 0x4840 0004 0x4880 0004 0x4A00 0004 0x4AE0 0004	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Contains a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 0000

Table 14-534. Register Call Summary for Register L4_AP_COMPONENT_H

L4 Interconnects

- L4 Address Protection (L4 AP) Register Summary: [0] [1]

Table 14-535. L4_AP_SEGMENT_i_L

Address Offset	0x100 + (0x08*i)	**Index**	
Physical Address	0x4800 0100 + (0x08*i) 0x4840 0100 + (0x08*i) 0x4880 0100 + (0x08*i) 0x4A00 0100 + (0x08*i) 0x4AE0 0100 + (0x08*i)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define the base address of each segments		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE																															

Bits	Field Name	Description	Type	Reset
31:0	BASE	The base address of the segment (with 0s from bit 0 to bit SIZE-1).	R	see Table 14-539

Table 14-536. Register Call Summary for Register L4_AP_SEGMENT_i_L

L4 Interconnects

- L4 Firewall Address and Protection Register Settings: [0]
- L4 Address Protection (L4 AP) Register Summary: [1] [2]

Table 14-537. L4_AP_SEGMENT_i_H

Address Offset	0x104 + (0x08*i)	**Index**	

Table 14-537. L4_AP_SEGMENT_i_H (continued)

Physical Address	0x4800 0104 + (0x08*i) 0x4840 0104 + (0x08*i) 0x4880 0104 + (0x08*i) 0x4A00 0104 + (0x08*i) 0x4AE0 0104 + (0x08*i)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define the size of each segments		
Type	RW		

31–5	4–0
RESERVED	SIZE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Read returns 0.	R	0x0000000
4:0	SIZE	Segment size is a power of 2, where 2^{SIZE} is the byte size of a segment (all segment registers use the same size).	R	see Table 14-539

Table 14-538. Register Call Summary for Register L4_AP_SEGMENT_i_H

L4 Interconnects

- L4 Firewall Address and Protection Register Settings: [0]
- L4 Address Protection (L4 AP) Register Summary: [1] [2]

Table 14-539. Reset Value for L4_AP_SEGMENT_i

i	L4 PER1		L4 PER2		L4 PER3		L4 CFG		L4 WKUP	
	BASE	SIZE	BASE	SIZE	BASE	SIZE	BASE	SIZE	BASE	SIZE
0	0x0000 0000	0x15	0x0000 0000	0x16	0x0000 0000	0x15	0x0000 0000	0x14	0x0000 0000	0x10
1	0x0020 0000	0x15	-	-	-	-	0x0010 0000	0x14	0x0001 0000	0x10
2	-	-	-	-	-	-	0x0020 0000	0x14	0x0002 0000	0x10
3	-	-	-	-	-	-	-	-	0x0003 0000	0x10

Table 14-540. L4_AP_PROT_GROUP_MEMBERS_k_L

Address Offset	0x200 + (0x08*k)	**Index**	
Physical Address	0x4800 0200 + (0x08*k) 0x4840 0200 + (0x08*k) 0x4880 0200 + (0x08*k) 0x4A00 0200 + (0x08*k) 0x4AE0 0200 + (0x08*k)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define ConnID bit vectors for a protection group.		
Type	R		

31–16	15–0
RESERVED	CONNID_BIT_VECTOR

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x000000000000
15:0	CONNID_BIT_VECTOR	Specifies protection group members N is 2**W, where W is the connID width	R(1)	0xFFFF

(1) For L4_PER1 and L4_WKUP, when k = 2 to 7 the access type of CONNID_BIT_VECTOR is RW.
For L4_PER2, L4_PER3 and L4_CFG, when k = 1 to 7 the access type of CONNID_BIT_VECTOR is RW.

Table 14-541. Register Call Summary for Register L4_AP_PROT_GROUP_MEMBERS_k_L

L4 Interconnects
- Protection Group: [0] [1] [2] [3] [4] [5] [6]
- L4 Firewall Address and Protection Register Settings: [7]
- Operational Modes Configuration: [8]
- L4 Address Protection (L4 AP) Register Summary: [9] [10]

Table 14-542. L4_AP_PROT_GROUP_MEMBERS_k_H

Address Offset	0x204 + (0x08*k)	**Index**	
Physical Address	0x4800 0204 + (0x08*k) 0x4840 0204 + (0x08*k) 0x4880 0204 + (0x08*k) 0x4A00 0204 + (0x08*k) 0x4AE0 0204 + (0x08*k)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define ConnID bit vectors for a protection group.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0's	R	0x0000 0000

Table 14-543. Register Call Summary for Register L4_AP_PROT_GROUP_MEMBERS_k_H

L4 Interconnects
- Protection Group: [0] [1] [2] [3] [4] [5] [6]
- L4 Address Protection (L4 AP) Register Summary: [7] [8]

Table 14-544. L4_AP_PROT_GROUP_ROLES_k_L

Address Offset	0x200 + (0x08*k)	**Index**	
Physical Address	0x4800 0280 + (0x08*k) 0x4840 0280 + (0x08*k) 0x4880 0280 + (0x08*k) 0x4A00 0280 + (0x08*k) 0x4AE0 0280 + (0x08*k)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define MReqInfo bit vectors for a protection group.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enabled bits N is 2**W, where W (W is less than or equal to 6) is the number of MReqInfo bits configured for role checking	R	0xFFFF

Table 14-545. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_L

L4 Interconnects
- Protection Group: [0] [1] [2] [3] [4] [5] [6] [7]
- L4 Firewall Address and Protection Register Settings: [8]
- Operational Modes Configuration: [9]
- L4 Address Protection (L4 AP) Register Summary: [10] [11]

Table 14-546. Reset Value for L4_AP_PROT_GROUP_ROLES_k

	L4_PER		L4_CFG		L4_WKUP	
k	Type	Reset Value	Type	Reset Value	Type	Reset value
0	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)
1	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)
2	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)
3	RW	0xFFFF FFFF	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)
4	RW	0xFFFF FFFF	R	0xFFFF FFFF(1)	R	0xFFFF FFFF(1)
5	RW	0xFFFF FFFF	RW	0xFFFF FFFF	RW	0xFFFF FFFF
6	RW	0xFFFF FFFF	RW	0xFFFF FFFF	RW	0xFFFF FFFF
7	RW	0xFFFF FFFF	RW	0xFFFF FFFF	RW	0xFFFF FFFF

(1) Value exported from SCM and valid for GP device only (see *Chapter 18*, *Control Module*). The values of the other protection registers can be modified during run time.

Table 14-547. L4_AP_PROT_GROUP_ROLES_k_H

Address Offset	0x204 + (0x08*k)	**Index**	
Physical Address	0x4800 0284 + (0x08*k) 0x4840 0284 + (0x08*k) 0x4880 0284 + (0x08*k) 0x4A00 0284 + (0x08*k) 0x4AE0 0284 + (0x08*k)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define ConnID bit vectors for a protection group.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enabled bits N is 2**W, where W (W is less than or equal to 6) is the number of MReqInfo bits configured for role checking.	R	0xFFFF

Table 14-548. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_H

L4 Interconnects

- Protection Group: [0] [1] [2] [3] [4]
- L4 Firewall Address and Protection Register Settings: [5]
- L4 Address Protection (L4 AP) Register Summary: [6] [7]

Table 14-549. L4_AP_REGION_l_L

Address Offset	0x300 + (0x08*l)	**Index**	
Physical Address	0x4800 0300 + (0x08*l) 0x4840 0300 + (0x08*l) 0x4880 0300 + (0x08*l) 0x4A00 0300 + (0x08*l) 0x4AE0 0300 + (0x08*l)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define the base address of the region in respect to the segment it belongs to.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											BASE																				

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Read returns 0.	R	0x00

Bits	Field Name	Description	Type	Reset
20:0	BASE	Sets the base address of the region relative to its segment base.	R	See Table 14-553 to Table 14-557

Table 14-550. Register Call Summary for Register L4_AP_REGION_I_L

L4 Interconnects

- L4 Firewall Address and Protection Register Settings: [0]
- Operational Modes Configuration: [1]
- L4 Address Protection (L4 AP) Register Summary: [2] [3]

Table 14-551. L4_AP_REGION_I_H

Address Offset	0x304 + (0x08*I)	**Index**	
Physical Address	0x4800 0304 + (0x08*I) 0x4840 0304 + (0x08*I) 0x4880 0304 + (0x08*I) 0x4A00 0304 + (0x08*I) 0x4AE0 0304 + (0x08*I)	**Instance**	PER1_AP PER2_AP PER3_AP CFG_AP WKUP_AP
Description	Define the size, protection group and segment ID of the region		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MADDRSPACE				RESERVED	SEGMENT_ID			RESERVED	PROT_GROUP_ID			RESERVED	BYTE_DATA_WIDTH_EXP		RESERVED		PHY_TARGET_ID							RESERVED	SIZE						ENABLE

Bits	Field Name	Description	Type	Reset
31:28	MADDRSPACE	Target interconnect MAddrSpace	R	See Table 14-553 to Table 14-557
27	RESERVED	Read returns 0	R	0x0
26:24	SEGMENT_ID	Segment ID of the region	R	SSee Table 14-553 to Table 14-557
23	RESERVED	Read returns 0	R	0x0
22:20	PROT_GROUP_ID	Protection group ID	RW	See Table 14-553 to Table 14-557
19	RESERVED	Read returns 0	R	0x0
18:17	BYTE_DATA_WIDTH_EXP	Target data byte width	R	See Table 14-553 to Table 14-557
16:15	RESERVED	Read returns 0	R	0x0
14:8	PHY_TARGET_ID	Physical target ID	R	See Table 14-553 to Table 14-557
7	RESERVED	Read returns 0.	R	0x0
6:1	SIZE	Define the size of the region in bytes. 2^{SIZE} equals the region.	R	See Table 14-553 to Table 14-557
0	ENABLE	0x0: Disable the region, no access allows 0x1: Enable the region, with access as define in registers	R	0x1

Table 14-552. Register Call Summary for Register L4_AP_REGION_I_H

L4 Interconnects

- L4 Firewall Address and Protection Register Settings: [0]
- L4 Address Protection (L4 AP) Register Summary: [1] [2]

Table 14-553. L4_AP_REGION_l Reset Value for L4 PER1

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x00 0000
1	0x1	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1000
2	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x00 0800
3	0x0	0x0	0x7	0x2	0x04	0x0C	0x1	0x02 0000
4	0x0	0x0	0x7	0x2	0x05	0x0C	0x1	0x02 1000
5	0x0	0x0	0x7	0x2	0x3E	0x0C	0x1	0x03 2000
6	0x0	0x0	0x7	0x2	0x3F	0x0C	0x1	0x03 3000
7	0x0	0x0	0x7	0x2	0x46	0x0C	0x1	0x03 4000
8	0x0	0x0	0x7	0x2	0x47	0x0C	0x1	0x03 5000
9	0x0	0x0	0x7	0x2	0x4E	0x0C	0x1	0x03 6000
10	0x0	0x0	0x7	0x2	0x4F	0x0C	0x1	0x03 7000
11	0x0	0x0	0x7	0x2	0x56	0x0C	0x1	0x03 E000
12	0x0	0x0	0x7	0x2	0x57	0x0C	0x1	0x03 F000
13	0x0	0x0	0x7	0x2	0x0E	0x0C	0x1	0x05 5000
14	0x0	0x0	0x7	0x2	0x0F	0x0C	0x1	0x05 6000
15	0x0	0x0	0x7	0x2	0x06	0x0C	0x1	0x05 7000
16	0x0	0x0	0x7	0x2	0x07	0x0C	0x1	0x05 8000
17	0x0	0x0	0x7	0x2	0x16	0x0C	0x1	0x05 9000
18	0x0	0x0	0x7	0x2	0x17	0x0C	0x1	0x05 A000
19	0x0	0x0	0x7	0x2	0x1E	0x0C	0x1	0x05 B000
20	0x0	0x0	0x7	0x2	0x1F	0x0C	0x1	0x05 C000
21	0x0	0x0	0x7	0x2	0x26	0x0C	0x1	0x05 D000
22	0x0	0x0	0x7	0x2	0x27	0x0C	0x1	0x05 E000
23	0x0	0x0	0x7	0x2	0x32	0x0C	0x1	0x06 0000
24	0x0	0x0	0x7	0x2	0x24	0x0C	0x1	0x06 A000
25	0x0	0x0	0x7	0x2	0x25	0x0C	0x1	0x06 B000
26	0x0	0x0	0x7	0x2	0x2C	0x0C	0x1	0x06 C000
27	0x0	0x0	0x7	0x2	0x2D	0x0C	0x1	0x06 D000
28	0x1	0x0	0x7	0x2	0x0C	0x0C	0x1	0x06 E000
29	0x0	0x0	0x7	0x2	0x0D	0x0C	0x1	0x06 F000
30	0x0	0x0	0x7	0x2	0x22	0x0C	0x1	0x07 0000
31	0x0	0x0	0x7	0x2	0x23	0x0C	0x1	0x07 1000
32	0x0	0x0	0x7	0x2	0x2A	0x0C	0x1	0x07 2000
33	0x0	0x0	0x7	0x2	0x2B	0x0C	0x1	0x07 3000
34	0x0	0x0	0x7	0x2	0x33	0x0C	0x1	0x06 1000
35	0x0	0x0	0x7	0x2	0x36	0x0C	0x1	0x05 3000
36	0x0	0x0	0x7	0x2	0x37	0x0C	0x1	0x05 4000
37	0x0	0x0	0x7	0x2	0x52	0x0C	0x1	0x0B 2000
38	0x0	0x0	0x7	0x2	0x53	0x0C	0x1	0x0B 3000
39	0x0	0x0	0x7	0x2	0x0A	0x0C	0x1	0x07 8000
40	0x0	0x0	0x7	0x2	0x0B	0x0C	0x1	0x07 9000
41	0x0	0x0	0x7	0x2	0x5E	0x0C	0x1	0x08 6000
42	0x0	0x0	0x7	0x2	0x5F	0x0C	0x1	0x08 7000
43	0x0	0x0	0x7	0x2	0x66	0x0C	0x1	0x08 8000

Table 14-553. L4_AP_REGION_l Reset Value for L4 PER1 (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
44	0x0	0x0	0x7	0x2	0x67	0x0C	0x1	0x08 9000
45	0x0	0x0	0x7	0x2	0x2E	0x0C	0x1	0x05 1000
46	0x0	0x0	0x7	0x2	0x2F	0x0C	0x1	0x05 2000
47	0x0	0x0	0x7	0x2	0x08	0x0C	0x1	0x09 8000
48	0x0	0x0	0x7	0x2	0x09	0x0C	0x1	0x09 9000
49	0x0	0x0	0x7	0x2	0x10	0x0C	0x1	0x09 A000
50	0x0	0x0	0x7	0x2	0x11	0x0C	0x1	0x09 B000
51	0x0	0x0	0x7	0x2	0x38	0x0C	0x1	0x09 C000
52	0x0	0x0	0x7	0x2	0x39	0x0C	0x1	0x09 D000
53	0x0	0x0	0x7	0x2	0x1C	0x0C	0x1	0x06 8000
54	0x0	0x0	0x7	0x2	0x1D	0x0C	0x1	0x06 9000
55	0x0	0x0	0x1	0x2	0x12	0x0D	0x1	0x09 0000
56	0x0	0x0	0x7	0x2	0x13	0x0C	0x1	0x09 2000
57	0x0	0x0	0x1	0x2	0x42	0x0C	0x1	0x0A 4000
58	0x0	0x0	0x7	0x2	0x43	0x0C	0x1	0x0A 6000
59	0x0	0x0	0x1	0x2	0x1A	0x0E	0x1	0x0A 8000
60	0x0	0x0	0x7	0x2	0x1B	0x0C	0x1	0x0A C000
61	0x0	0x0	0x7	0x2	0x20	0x0C	0x1	0x0A D000
62	0x0	0x0	0x7	0x2	0x21	0x0C	0x1	0x0A E000
63	0x0	0x0	0x7	0x2	0x14	0x0C	0x1	0x06 6000
64	0x0	0x0	0x7	0x2	0x15	0x0C	0x1	0x06 7000
65	0x0	0x0	0x7	0x2	0x40	0x0C	0x1	0x0B 4000
66	0x0	0x0	0x7	0x2	0x41	0x0C	0x1	0x0B 5000
67	0x0	0x0	0x7	0x2	0x48	0x0C	0x1	0x0B 8000
68	0x0	0x0	0x7	0x2	0x49	0x0C	0x1	0x0B 9000
69	0x0	0x0	0x7	0x2	0x18	0x0C	0x1	0x0B A000
70	0x0	0x0	0x7	0x2	0x19	0x0C	0x1	0x0B B000
71	0x0	0x0	0x7	0x2	0x28	0x0C	0x1	0x0D 1000
72	0x0	0x0	0x7	0x2	0x29	0x0C	0x1	0x0D 2000
73	0x0	0x0	0x7	0x2	0x30	0x0C	0x1	0x0D 5000
74	0x0	0x0	0x7	0x2	0x31	0x0C	0x1	0x0D 6000
75	0x0	0x0	0x7	0x2	0x02	0x0C	0x1	0x0A 2000
76	0x0	0x0	0x7	0x2	0x03	0x0C	0x1	0x0A 3000
77	0x2	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1400
78	0x3	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1800
79	0x4	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1C00
80	0x1	0x0	0x1	0x2	0x42	0x0C	0x1	0x0A 5000
81	0x0	0x0	0x7	0x2	0x3A	0x0C	0x1	0x07 A000
82	0x0	0x0	0x7	0x2	0x3B	0x0C	0x1	0x07 B000
83	0x0	0x0	0x7	0x2	0x4A	0x0C	0x1	0x07 C000
84	0x0	0x0	0x7	0x2	0x4B	0x0C	0x1	0x07 D000

Table 14-554. L4_AP_REGION_l Reset Value for L4 PER2

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x00 0000
1	0x1	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1000
2	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x00 0800
3	0x0	0x0	0x7	0x2	0x0E	0x0C	0x1	0x08 4000
4	0x2	0x0	0x7	0x2	0x0A	0x0C	0x1	0x00 1400
5	0x3	0x0	0x7	0x2	0x0A	0x0C	0x1	0x00 1800
6	0x0	0x0	0x7	0x2	0x11	0x0C	0x1	0x08 8000
7	0x0	0x0	0x7	0x2	0x18	0x0C	0x1	0x02 C000
8	0x0	0x0	0x7	0x2	0x19	0x0C	0x1	0x02 D000
9	0x0	0x0	0x7	0x2	0x0E	0x0D	0x1	0x06 0000
10	0x0	0x0	0x7	0x2	0x0F	0x0C	0x1	0x06 2000
11	0x0	0x0	0x7	0x2	0x1E	0x0D	0x1	0x06 4000
12	0x0	0x0	0x7	0x2	0x1F	0x0C	0x1	0x06 6000
13	0x0	0x0	0x7	0x2	0x26	0x0C	0x1	0x06 8000
14	0x0	0x0	0x7	0x2	0x27	0x0C	0x1	0x06 A000
15	0x0	0x0	0x7	0x2	0x2E	0x0D	0x1	0x06 C000
16	0x0	0x0	0x7	0x2	0x2F	0x0C	0x1	0x6 E000
17	0x0	0x0	0x7	0x2	0x06	0x0C	0x1	0x03 6000
18	0x0	0x0	0x7	0x2	0x07	0x0C	0x1	0x03 8000
19	0x0	0x0	0x7	0x2	0x36	0x0D	0x1	0x07 0000
20	0x0	0x0	0x7	0x2	0x37	0x0C	0x1	0x07 2000
21	0x0	0x0	0x7	0x2	0x3E	0x0C	0x1	0x03 A000
22	0x0	0x0	0x7	0x2	0x3F	0x0C	0x1	0x03 B000
23	0x0	0x0	0x7	0x2	0x08	0x0C	0x1	0x03 C000
24	0x0	0x0	0x7	0x2	0x09	0x0C	0x1	0x03 D000
25	0x0	0x0	0x7	0x2	0x30	0x0C	0x1	0x03 E000
26	0x0	0x0	0x7	0x2	0x31	0x0C	0x1	0x03 F000
27	0x0	0x0	0x7	0x2	0x38	0x0C	0x1	0x04 0000
28	0x1	0x0	0x7	0x2	0x39	0x0C	0x1	0x04 1000
29	0x0	0x0	0x7	0x2	0x20	0x0C	0x1	0x04 2000
30	0x0	0x0	0x7	0x2	0x21	0x0C	0x1	0x04 3000
31	0x0	0x0	0x7	0x2	0x16	0x0D	0x1	0x08 0000
32	0x0	0x0	0x7	0x2	0x17	0x0C	0x1	0x08 2000
33	0x0	0x0	0x7	0x2	0x1A	0x0C	0x1	0x04 A000
34	0x0	0x0	0x7	0x2	0x1B	0x0C	0x1	0x04 B000
35	0x0	0x0	0x7	0x2	0x14	0x0D	0x1	0x07 4000
36	0x0	0x0	0x7	0x2	0x15	0x0C	0x1	0x07 6000
37	0x0	0x0	0x7	0x2	0x24	0x0C	0x1	0x05 0000
38	0x0	0x0	0x7	0x2	0x25	0x0C	0x1	0x05 1000
39	0x0	0x0	0x7	0x2	0x0C	0x0D	0x1	0x07 8000
40	0x0	0x0	0x7	0x2	0x0D	0x0C	0x1	0x07 A000
41	0x0	0x0	0x7	0x2	0x2C	0x0C	0x1	0x05 4000
42	0x0	0x0	0x7	0x2	0x2D	0x0C	0x1	0x05 5000
43	0x0	0x0	0x7	0x2	0x04	0x0D	0x1	0x07 C000

Table 14-554. L4_AP_REGION_l Reset Value for L4 PER2 (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
44	0x0	0x0	0x7	0x2	0x05	0x0C	0x1	0x07 E000
45	0x0	0x0	0x7	0x2	0x1C	0x0C	0x1	0x04 C000
46	0x0	0x0	0x7	0x2	0x1D	0x0C	0x1	0x04 D000
47	0x0	0x0	0x7	0x2	0x02	0x0C	0x1	0x02 0000
48	0x0	0x0	0x7	0x2	0x03	0x0C	0x1	0x02 1000
49	0x0	0x0	0x7	0x2	0x0A	0x0C	0x1	0x02 2000
50	0x0	0x0	0x7	0x2	0x0B	0x0C	0x1	0x02 3000
51	0x0	0x0	0x7	0x2	0x12	0x0C	0x1	0x02 4000
52	0x0	0x0	0x7	0x2	0x13	0x0C	0x1	0x02 5000
53	0x0	0x0	0x7	0x2	0x40	0x0C	0x1	0x04 6000
54	0x0	0x0	0x7	0x2	0x41	0x0C	0x1	0x04 7000
55	0x0	0x0	0x1	0x2	0x48	0x0C	0x1	0x04 8000
56	0x0	0x0	0x7	0x2	0x49	0x0C	0x1	0x04 9000
57	0x0	0x0	0x1	0x2	0x28	0x0D	0x1	0x05 8000
58	0x0	0x0	0x7	0x2	0x29	0x0C	0x1	0x05 A000
59	0x0	0x0	0x1	0x2	0x46	0x0C	0x1	0x05 B000
60	0x0	0x0	0x7	0x2	0x47	0x0C	0x1	0x05 C000
61	0x0	0x0	0x7	0x2	0x22	0x0C	0x1	0x05 D000
62	0x0	0x0	0x7	0x2	0x23	0x0C	0x1	0x05 E000

Table 14-555. L4_AP_REGION_l Reset Value for L4 PER3

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x00 0000
1	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x00 0800
2	0x1	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1000
3	0x2	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1400
4	0x3	0x0	0x7	0x2	0x0A	0x0C	0x1	0x00 1800
5	0x0	0x0	0x7	0x2	0x08	0x0C	0x1	0x02 0000
6	0x0	0x0	0x7	0x2	0x09	0x0C	0x1	0x02 1000
7	0x0	0x0	0x7	0x2	0x24	0x0C	0x1	0x02 2000
8	0x0	0x0	0x7	0x2	0x25	0x0C	0x1	0x02 3000
9	0x0	0x0	0x7	0x2	0x26	0x0C	0x1	0x02 4000
10	0x0	0x0	0x7	0x2	0x27	0x0C	0x1	0x02 5000
11	0x0	0x0	0x7	0x2	0x0C	0x0C	0x1	0x02 6000
12	0x0	0x0	0x7	0x2	0x0D	0x0C	0x1	0x02 7000
13	0x0	0x0	0x7	0x2	0x16	0x0C	0x1	0x02 8000
14	0x0	0x0	0x7	0x2	0x17	0x0C	0x1	0x02 9000
15	0x0	0x0	0x7	0x2	0x10	0x0C	0x1	0x02 A000
16	0x0	0x0	0x7	0x2	0x11	0x0C	0x1	0x02 B000
17	0x0	0x0	0x7	0x2	0x02	0x0C	0x1	0x02 C000
18	0x0	0x0	0x7	0x2	0x03	0x0C	0x1	0x02 D000
19	0x0	0x0	0x7	0x2	0x14	0x0C	0x1	0x02 E000
20	0x0	0x0	0x7	0x2	0x15	0x0C	0x1	0x02 F000
21	0x0	0x0	0x7	0x2	0x0A	0x10	0x1	0x17 0000

Table 14-555. L4_AP_REGION_l Reset Value for L4 PER3 (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
22	0x0	0x0	0x7	0x2	0x0B	0x0C	0x1	0x18 0000
23	0x0	0x0	0x7	0x2	0x2E	0x0C	0x1	0x19 0000
24	0x0	0x0	0x7	0x2	0x2F	0x0C	0x1	0x1A 0000
25	0x0	0x0	0x7	0x2	0x34	0x10	0x1	0x1B 0000
26	0x0	0x0	0x7	0x2	0x35	0x0C	0x1	0x1C 0000
27	0x0	0x0	0x7	0x2	0x30	0x0C	0x1	0x1D 0000
28	0x1	0x0	0x7	0x2	0x31	0x0C	0x1	0x1E 0000
29	0x0	0x0	0x7	0x2	0x12	0x0C	0x1	0x03 8000
30	0x0	0x0	0x7	0x2	0x13	0x0C	0x1	0x03 9000
31	0x0	0x0	0x7	0x2	0x32	0x0D	0x1	0x05 C000
32	0x0	0x0	0x7	0x2	0x33	0x0C	0x1	0x05 D000
33	0x0	0x0	0x7	0x2	0x3E	0x0C	0x1	0x03 A000
34	0x0	0x0	0x7	0x2	0x3F	0x0C	0x1	0x03 B000
35	0x0	0x0	0x7	0x2	0x3A	0x0D	0x1	0x03 C000
36	0x0	0x0	0x7	0x2	0x3B	0x0C	0x1	0x03 D000
37	0x0	0x0	0x7	0x2	0x46	0x0C	0x1	0x03 E000
38	0x0	0x0	0x7	0x2	0x47	0x0C	0x1	0x03 F000
39	0x0	0x0	0x7	0x2	0x64	0x0D	0x1	0x04 0000
40	0x0	0x0	0x7	0x2	0x65	0x0D	0x1	0x04 1000
41	0x0	0x0	0x7	0x2	0x4E	0x0C	0x1	0x04 2000
42	0x0	0x0	0x7	0x2	0x4F	0x0C	0x1	0x04 3000
43	0x0	0x0	0x7	0x2	0x42	0x0C	0x1	0x04 4000
44	0x0	0x0	0x7	0x2	0x43	0x0C	0x1	0x04 5000
45	0x0	0x0	0x7	0x2	0x48	0x0C	0x1	0x04 6000
46	0x0	0x0	0x7	0x2	0x49	0x0C	0x1	0x04 7000
47	0x0	0x0	0x7	0x2	0x46	0x0C	0x1	0x04 8000
48	0x0	0x0	0x7	0x2	0x37	0x0C	0x1	0x04 9000
49	0x0	0x0	0x7	0x2	0x38	0x0C	0x1	0x04 A000
50	0x0	0x0	0x7	0x2	0x39	0x0C	0x1	0x04 B000
51	0x0	0x0	0x7	0x2	0x44	0x0C	0x1	0x04 C000
52	0x0	0x0	0x7	0x2	0x45	0x0C	0x1	0x04 D000
53	0x0	0x0	0x7	0x2	0x4C	0x0C	0x1	0x04 E000
54	0x0	0x0	0x7	0x2	0x4D	0x0C	0x1	0x04 F000
55	0x0	0x0	0x7	0x2	0x40	0x0C	0x1	0x05 0000
56	0x0	0x0	0x7	0x2	0x41	0x0C	0x1	0x05 1000
57	0x0	0x0	0x1	0x2	0x54	0x0C	0x1	0x05 2000
58	0x0	0x0	0x7	0x2	0x55	0x0C	0x1	0x05 3000
59	0x0	0x0	0x1	0x2	0x1A	0x0C	0x1	0x05 4000
60	0x0	0x0	0x7	0x2	0x1B	0x0C	0x1	0x05 5000
61	0x0	0x0	0x7	0x2	0x22	0x0C	0x1	0x05 6000
62	0x0	0x0	0x7	0x2	0x23	0x0C	0x1	0x05 7000
63	0x0	0x0	0x7	0x2	0x2A	0x0C	0x1	0x05 8000
64	0x0	0x0	0x7	0x2	0x2B	0x0C	0x1	0x05 9000
65	0x0	0x0	0x7	0x2	0x5C	0x0C	0x1	0x05 A000

Table 14-555. L4_AP_REGION_l Reset Value for L4 PER3 (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
66	0x0	0x0	0x7	0x2	0x5D	0x0C	0x1	0x05 B000
67	0x0	0x0	0x7	0x2	0x52	0x0C	0x1	0x06 4000
68	0x0	0x0	0x7	0x2	0x53	0x0C	0x1	0x06 5000
69	0x0	0x0	0x7	0x2	0x6C	0x0C	0x1	0x05 E000
70	0x0	0x0	0x7	0x2	0x6D	0x0C	0x1	0x05 F000
71	0x0	0x0	0x7	0x2	0x4A	0x0C	0x1	0x06 0000
72	0x0	0x0	0x7	0x2	0x4B	0x0C	0x1	0x06 1000
73	0x0	0x0	0x7	0x2	0x74	0x0C	0x1	0x06 2000
74	0x0	0x0	0x7	0x2	0x75	0x0C	0x1	0x06 3000
75	0x0	0x0	0x7	0x2	0x3C	0x0C	0x1	0x14 0000
76	0x0	0x0	0x7	0x2	0x3D	0x0C	0x1	0x16 0000
77	0x0	0x0	0x7	0x2	0x1E	0x0C	0x1	0x01 6000
78	0x0	0x0	0x7	0x2	0x1F	0x0C	0x1	0x01 7000
79	0x0	0x0	0x7	0x2	0x06	0x11	0x1	0x0C 0000
80	0x0	0x0	0x7	0x2	0x07	0x0C	0x1	0x0E 0000
81	0x0	0x0	0x7	0x2	0x20	0x0C	0x1	0x00 4000
82	0x0	0x0	0x7	0x2	0x21	0x0C	0x1	0x00 5000
83	0x0	0x0	0x7	0x2	0x0E	0x11	0x1	0x08 0000
84	0x0	0x0	0x7	0x2	0x0F	0x0C	0x1	0x0A 0000
85	0x0	0x0	0x7	0x2	0x01	0x11	0x1	0x10 0000
86	0x0	0x0	0x7	0x2	0x05	0x0C	0x1	0x12 0000
87	0x0	0x0	0x7	0x2	0x28	0x0C	0x1	0x01 0000
88	0x0	0x0	0x7	0x2	0x29	0x0C	0x1	0x01 1000
89	0x0	0x0	0x7	0x2	0x18	0x0C	0x1	0x00 A000
90	0x0	0x0	0x7	0x2	0x19	0x0C	0x1	0x00 B000
91	0x0	0x0	0x7	0x2	0x1C	0x0C	0x1	0x01 C000
92	0x0	0x0	0x7	0x2	0x1D	0x0C	0x1	0x01 D000
93	0x0	0x0	0x7	0x2	0x2C	0x0C	0x1	0x01 E000
94	0x0	0x0	0x7	0x2	0x2D	0x0C	0x1	0x01 F000
95	0x0	0x0	0x7	0x2	0x7C	0x0C	0x1	0x00 2000
96	0x0	0x0	0x7	0x2	0x7D	0x0C	0x1	0x00 3000

Table 14-556. L4_AP_REGION_l Reset Value for L4 CFG

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x0 0000
1	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x0 0800
2	0x1	0x0	0x7	0x2	0x00	0x0C	0x1	0x0 1000
3	0x0	0x0	0x7	0x2	0x08	0x0D	0x1	0x0 2000
4	0x0	0x0	0x7	0x2	0x09	0x0C	0x1	0x0 4000
5	0x0	0x0	0x7	0x2	0x10	0x0C	0x1	0x0 5000
6	0x0	0x0	0x7	0x2	0x11	0x0C	0x1	0x0 6000
7	0x0	0x0	0x7	0x2	0x0E	0x0D	0x1	0x0 8000
8	0x0	0x0	0x7	0x2	0x0F	0x0C	0x1	0x0 A000
9	0x0	0x0	0x7	0x2	0x02	0x0C	0x1	0x5 6000

Table 14-556. L4_AP_REGION_l Reset Value for L4 CFG (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
10	0x0	0x0	0x7	0x2	0x03	0x0C	0x1	0x5 7000
11	0x0	0x0	0x7	0x2	0x1A	0x0D	0x1	0x5 E000
12	0x0	0x0	0x7	0x2	0x1B	0x0C	0x1	0x6 0000
13	0x0	0x0	0x7	0x2	0x20	0x0F	0x1	0x8 0000
14	0x0	0x0	0x7	0x2	0x21	0x0C	0x1	0x8 8000
15	0x0	0x0	0x7	0x2	0x40	0x0F	0x1	0xA 0000
16	0x0	0x0	0x7	0x2	0x41	0x0C	0x1	0xA 8000
17	0x0	0x0	0x7	0x2	0x72	0x0C	0x1	0xD 9000
18	0x0	0x0	0x7	0x2	0x73	0x0C	0x1	0xD A000
19	0x0	0x0	0x7	0x2	0x18	0x0C	0x1	0xD D000
20	0x0	0x0	0x7	0x2	0x19	0x0C	0x1	0xD E000
21	0x0	0x0	0x5	0x2	0x28	0x0C	0x1	0xE 0000
22	0x0	0x0	0x7	0x2	0x29	0x0C	0x1	0xE 1000
23	0x0	0x0	0x7	0x2	0x04	0x0C	0x1	0xF 4000
24	0x0	0x0	0x7	0x2	0x05	0x0C	0x1	0xF 5000
25	0x0	0x0	0x7	0x2	0x78	0x0C	0x1	0xF 6000
26	0x0	0x0	0x7	0x2	0x79	0x0C	0x1	0xF 7000
27	0x0	0x1	0x7	0x2	0x3C	0x0C	0x1	0x0 2000
28	0x0	0x1	0x7	0x2	0x3D	0x0C	0x1	0x0 3000
29	0x0	0x1	0x7	0x2	0x1E	0x0C	0x1	0x0 8000
30	0x0	0x1	0x7	0x2	0x1F	0x0C	0x1	0x0 9000
31	0x0	0x1	0x7	0x2	0x06	0x10	0x1	0x4 0000
32	0x0	0x1	0x7	0x2	0x07	0x0C	0x1	0x5 0000
33	0x0	0x1	0x1	0x2	0x50	0x0C	0x1	0x5 1000
34	0x0	0x1	0x7	0x2	0x51	0x0C	0x1	0x5 2000
35	0x0	0x1	0x1	0x2	0x54	0x0C	0x1	0x5 3000
36	0x0	0x1	0x7	0x2	0x55	0x0C	0x1	0x5 4000
37	0x0	0x1	0x1	0x2	0x46	0x0C	0x1	0x5 5000
38	0x0	0x1	0x7	0x2	0x47	0x0C	0x1	0x5 6000
39	0x0	0x1	0x1	0x2	0x58	0x0C	0x1	0x5 7000
40	0x0	0x1	0x7	0x2	0x59	0x0C	0x1	0x5 8000
41	0x0	0x1	0x1	0x2	0x60	0x0C	0x1	0x5 B000
42	0x0	0x1	0x7	0x2	0x61	0x0C	0x1	0x5 C000
43	0x0	0x2	0x1	0x2	0x12	0x0C	0x1	0x1 8000
44	0x0	0x2	0x7	0x2	0x13	0x0C	0x1	0x1 9000
45	0x0	0x1	0x1	0x2	0x3A	0x0C	0x1	0x5 D000
46	0x0	0x1	0x7	0x2	0x3B	0x0C	0x1	0x5 E000
47	0x0	0x1	0x1	0x2	0x56	0x0C	0X1	0x5 F000
48	0x0	0x1	0x7	0x2	0x57	0x0C	0x1	0x6 0000
49	0x0	0x1	0x1	0x2	0x32	0x0C	0x1	0x6 1000
50	0x0	0x1	0x7	0x2	0x33	0x0C	0x1	0x6 2000
51	0x0	0x1	0x1	0x2	0x5C	0x0C	0x1	0x6 3000
52	0x0	0x1	0x7	0x2	0x5D	0x0C	0x01	0x6 4000
53	0x0	0x1	0x1	0x2	0x4E	0x0C	0x01	0x6 5000

Table 14-556. L4_AP_REGION_l Reset Value for L4 CFG (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
54	0x0	0x1	0x7	0x2	0x4F	0x0C	0x01	0x6 6000
55	0x0	0x1	0x1	0x2	0x5E	0x0C	0X01	0x6 7000
56	0x0	0x1	0x7	0x2	0x5F	0x0C	0x01	0x6 8000
57	0x0	0x1	0x1	0x2	0x68	0x0C	0x01	0x6 D000
58	0x0	0x1	0x7	0x2	0x69	0x0C	0x01	0x6 E000
59	0x0	0x0	0x7	0x2	0x42	0x0F	0x01	0x9 0000
60	0x0	0x0	0x7	0x2	0x43	0x0C	0x01	0x9 8000
61	0x0	0x1	0x1	0x2	0x48	0x0C	0x01	0x7 1000
62	0x0	0x1	0x7	0x2	0x49	0x0C	0x01	0x7 2000
63	0x0	0x1	0x1	0x2	0x2A	0x0C	0x01	0x7 3000
64	0x0	0x1	0x7	0x2	0x2B	0x0C	0x01	07 4000
65	0x0	0x1	0x1	0x2	0x64	0x0C	0x01	0x7 5000
66	0x0	0x1	0x7	0x2	0x65	0x0C	0x01	0x7 6000
67	0x0	0x1	0x1	0x2	0x66	0x0C	0x01	0x7 7000
68	0x0	0x1	0x7	0x2	0x67	0x0C	0x01	0x7 8000
69	0x0	0x1	0x7	0x2	0x26	0x0C	0x01	0x8 1000
70	0x0	0x1	0x7	0x2	0x27	0x0C	0x01	0x8 2000
71	0x0	0x1	0x7	0x2	0x2E	0x0C	0x01	0x8 3000
72	0x0	0x1	0x7	0x2	0x2F	0x0C	0x01	0x8 4000
73	0x0	0x1	0x7	0x2	0x36	0x0C	0x01	0x8 5000
74	0x0	0x1	0x7	0x2	0x37	0x0C	0x01	0x8 6000
75	0x0	0x1	0x7	0x2	0x74	0x0C	0x01	0x8 7000
76	0x0	0x1	0x7	0x2	0x75	0x0C	0x01	0x8 8000
77	0x0	0x2	0x1	0x2	0x3E	0x0C	0x01	0x0 0000
78	0x0	0x2	0x7	0x2	0x3F	0x0C	0x01	0x0 1000
79	0x0	0x2	0x1	0x2	0x30	0x0C	0x01	0x0 A000
80	0x0	0x2	0x7	0x2	0x31	0x0C	0x01	0x0 B000
81	0x0	0x2	0x1	0x2	0x0C	0x0C	0x01	0x0 C000
82	0x0	0x2	0x7	0x2	0x0D	0x0C	0x01	0x0 D000
83	0x0	0x2	0x1	0x2	0x22	0x0C	0x01	0x0 E000
84	0x0	0x2	0x7	0x2	0x23	0x0C	0x01	0x0 F000
85	0x0	0x2	0x1	0x2	0x14	0x0C	0x01	0x1 0000
86	0x0	0x2	0x7	0x2	0x15	0x0C	0x01	0x1 1000
87	0x0	0x2	0x1	0x2	0x16	0x0C	0x01	0x1 2000
88	0x0	0x2	0x7	0x2	0x17	0x0C	0x01	0x1 3000
89	0x0	0x2	0x1	0x2	0x1C	0x0C	0x01	0x1 4000
90	0x0	0x2	0x7	0x2	0x1D	0x0C	0x01	0x1 5000
91	0x0	0x2	0x1	0x2	0x4C	0x0C	0x01	0x2 A000
92	0x0	0x2	0x7	0x2	0x4D	0x0C	0x01	0x2 B000
93	0x0	0x2	0x1	0x2	0x38	0x0C	0x01	0x1 C000
94	0x0	0x2	0x7	0x2	0x39	0x0C	0X01	0x1 D000
95	0x0	0x2	0x1	0x2	0x0A	0x0C	0x01	0x1 E000
96	0x0	0x2	0x7	0x2	0x0B	0x0C	0x01	0x1 F000
97	0x0	0x2	0x1	0x2	0x24	0x0C	0x01	0x2 0000

Table 14-556. L4_AP_REGION_I Reset Value for L4 CFG (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
98	0x0	0x2	0x7	0x2	0x25	0x0C	0x01	0x2 1000
99	0x0	0x2	0x1	0x2	0x44	0x0C	0x01	0x2 4000
100	0x0	0x2	0x7	0x2	0x45	0x0C	0x01	0x2 5000
101	0x0	0x2	0x1	0x2	0x2C	0x0C	0x01	0x2 6000
102	0x0	0x2	0x7	0x2	0x2D	0x0C	0x01	0x2 7000
103	0x0	0x1	0x1	0x2	0x4A	0x0C	0x01	0x6 9000
104	0x0	0x1	0x7	0x2	0x4B	0x0C	0x01	0x6 A000
105	0x0	0x1	0x1	0x2	0x34	0x0C	0x01	0x7 9000
106	0x0	0x1	0x7	0x2	0x35	0x0C	0x01	0x7 A000
107	0x0	0x1	0x1	0x2	0x52	0x0C	0x01	0x6 B000
108	0x0	0x1	0x7	0x2	0x53	0x0C	0x01	0x6 C000
109	0x0	0x2	0x1	0x2	0x6C	0x0C	0x01	0x2 C000
110	0x0	0x2	0x7	0x2	0x6D	0x0C	0x01	0x2 D000
111	0x0	0x2	0x1	0x2	0x6E	0x0C	0x01	0x2 E000
112	0x0	0x2	0x7	0x2	0x6F	0x0C	0x01	0x2 F000
113	0x0	0x2	0x1	0x2	0x70	0x0C	0x01	0x3 0000
114	0x0	0x2	0x7	0x2	0x71	0x0C	0x01	0x3 1000
115	0x0	0x2	0x1	0x2	0x5A	0x0C	0x01	0x3 2000
116	0x0	0x2	0x7	0x2	0x5B	0x0C	0x01	0x3 3000
117	0x1	0x2	0x1	0x2	0x76	0x0C	0x01	0x3 4000
118	0x0	0x2	0x7	0x2	0x77	0x0C	0x01	0x3 5000
119	0x0	0x2	0x1	0x2	0x62	0x0C	0x01	0x3 6000
120	0x0	0x2	0x7	0x2	0x63	0x0C	0x01	0x3 7000
121	0x0	0x1	0x7	0x2	0x7C	0x0C	0x01	0x7 B000
122	0x0	0x1	0x7	0x2	0x7D	0x0C	0x01	0x7 C000
123	0x0	0x1	0x7	0x2	0x7E	0x0C	0x01	0x7 D000
124	0x0	0x1	0x7	0x2	0x7F	0x0C	0x01	0x7 E000
125	0x0	0x1	0x1	0x2	0x6A	0x0C	0x01	0x5 9000
126	0x0	0x1	0x7	0x2	0x6B	0x0C	0x01	0x5 A000
127	0x0	0x2	0x1	0x2	0x7A	0x0C	0x01	0x1 A000
128	0x0	0x2	0x7	0x2	0x7B	0x0C	0x01	0x1 B000

Table 14-557. L4_AP_REGION_I Reset values for L4 WKUP

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x0	0x0B	0x01	0x0000
1	0x1	0x0	0x7	0x2	0x0	0x0C	0x01	0x1000
2	0x5	0x0	0x0	0x2	0x01	0x0B	0x01	0x0800
3	0x0	0x0	0x7	0x2	0x10	0x0D	0x01	0x6000
4	0x0	0x0	0x7	0x2	0x11	0x0C	0x01	0x8000
5	0x0	0x1	0x7	0x2	0x20	0x0C	0x01	0x0000
6	0x0	0x1	0x7	0x2	0x21	0x0C	0x01	0x1000
7	0x0	0x1	0x7	0x2	0x28	0x0C	0x01	0x4000
8	0x0	0x1	0x7	0x2	0x29	0x0C	0x01	0x5000
9	0x0	0x1	0x7	0x2	0x30	0x0C	0x01	0x8000

Table 14-557. L4_AP_REGION_l Reset values for L4 WKUP (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
10	0x0	0x1	0x7	0x2	0x31	0x0C	0x01	0x9000
11	0x0	0x1	0x7	0x2	0x38	0x0C	0x01	0xC000
12	0x0	0x1	0x7	0x2	0x39	0x0C	0x01	0xD000
13	0x0	0x2	0x7	0x2	0x48	0x0C	0x01	0x6000
14	0x0	0x2	0x7	0x2	0x49	0x0C	0x01	0xA000
15	0x0	0x0	0x7	0x2	0x40	0x0C	0x01	0x4000
16	0x0	0x0	0x7	0x2	0x41	0x0C	0x01	0x5000
17	0x0	0x0	0x7	0x2	0x50	0x0C	0x01	0xC000
18	0x0	0x0	0x7	0x2	0x51	0x0C	0x01	0xD000
19	0x0	0x2	0x1	0x2	0x08	0x0C	0x01	0x0000
20	0x0	0x2	0x7	0x2	0x09	0x0C	0x01	0x1000
21	0x0	0x2	0x1	0x2	0x18	0x0C	0x01	0x2000
22	0x0	0x2	0x7	0x2	0x19	0x0C	0x01	0x3000
23	0x1	0x2	0x3	0x2	0x48	0x0A	0x01	0x7000
24	0x2	0x2	0x1	0x2	0x48	0x0B	0x01	0x8000
25	0x5	0x2	0x4	0x2	0x48	0x08	0x01	0x9000
26	0x3	0x2	0x1	0x2	0x48	0x09	0x01	0x8800
27	0x4	0x2	0x1	0x2	0x48	0x08	0x01	0x8A00
28	0x0	0x2	0x7	0x2	0x02	0x0C	0x01	0xB000
29	0x0	0x2	0x7	0x2	0x03	0x0C	0x01	0xC000
30	0x0	0x3	0x7	0x2	0x04	0x0D	0x01	0xC000
31	0x0	0x3	0x7	0x2	0x05	0x0C	0x01	0xE000
32	0x0	0x2	0x7	0x2	0x58	0x0C	0x01	0xF000
33	0x0	0x3	0x7	0x2	0x59	0x0C	0x01	0x0000
34	0x0	0x3	0x7	0x2	0x60	0x0C	0x01	0x1000
35	0x0	0x3	0x7	0x2	0x61	0x0C	0x01	0x2000
36	0x0	0x3	0x7	0x2	0x0A	0x0C	0x01	0x3000
37	0x0	0x3	0x7	0x2	0x0B	0x0C	0x01	0x4000
38	0x0	0x3	0x7	0x2	0x0C	0x0C	0x01	0x5000
39	0x0	0x3	0x7	0x2	0x0D	0x0C	0x01	0x6000
40	0x0	0x3	0x7	0x2	0x68	0x0C	0x01	0x7000
41	0x0	0x3	0x7	0x2	0x69	0x0C	0x01	0x8000
42	0x0	0x3	0x7	0x2	0x70	0x0C	0x01	0x9000
43	0x0	0x3	0x7	0x2	0x71	0x0C	0x01	0xA000

Chapter 15

Memory Subsystem

This chapter introduces the memory subsystem of the device.

15.1 Memory Subsystem Overview 3212
15.2 Dynamic Memory Manager 3217
15.3 EMIF Controller 3303
15.4 General-Purpose Memory Controller 3446
15.5 Error Location Module 3582
15.6 On-Chip Memory (OCM) Subsystem 3605

15.1 Memory Subsystem Overview

Figure 15-1 shows a functional diagram of all memory subsystems in the device.

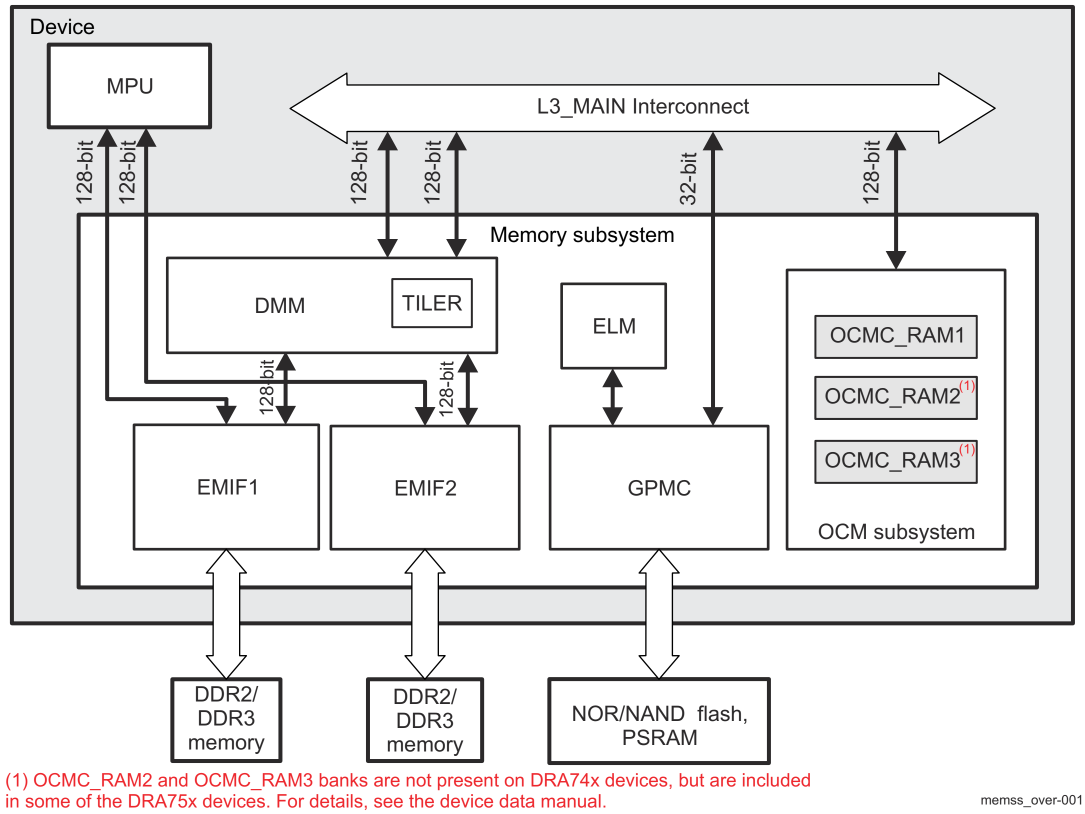


Figure 15-1. Memory Subsystem Functional Diagram

15.1.1 DMM Overview

The dynamic memory manager (DMM) module is typically located immediately in front of the SDRAM controllers (EMIFs), as shown in Figure 15-1, *Memory Subsystem Functional Diagram*.

In a broad sense, the DMM manages various aspects of memory accesses such as:

- Initiator-indexed priority generation
- Multizone SDRAM interleaving configuration
- Block object transfer optimization – tiling
- Centralized low-latency page translation – MMU-like feature

The dynamic qualifier for memory management highlights the software configurability, and hence the runtime nature, of the four aspects of memory management handled by the DMM.

From a functional perspective, the role of the DMM is to:

- Add initiator-based priority to any incoming requests
- Perform to/from tiling conversions of tiled requests
- Make on-the-fly basic transforms, such as quadrant rotations and mirroring
- Optionally, provide a low-latency page-based translation to handle memory fragmentation – MMU
- Distribute the traffic on all attached memory controllers according to the interleaving configuration

The main features of the DMM are:

- Programmable multizone DRAM mapping and interleaving configuration
- Programmable initiator-based request priority extension

- Multichannel memory transfer optimization
- Single SDRAM page request generation
- Low-latency interconnect port
- Page-grained address translation to manage memory fragmentation
- Automatic synchronized reloading of the address translation table

15.1.2 TILER Overview

The tiling and isometric lightweight engine for rotation (TILER) is a submodule of the DMM and is therefore described in Section 15.2, *Dynamic Memory Manager*.

The TILER is intended to improve bidimensional (tiled) block transfer efficiency. It is, therefore, a module aimed at:

- Primarily, efficient handling of 2-dimension (2D) data mapped in tiles, such as video or graphics macroblocks
- Optionally, managing the memory fragmentation and zero-copy physical frame-buffer swapping through a page-grained translation
- Making isometric (distance preserving) transforms, such as 90-, 180-, or 270-degree rotations, with a horizontal or vertical reflection

The lightweight qualifier of this engine highlights its limited support of isometric transforms, as:

- Rotation performs only basic quadrant rotations by some multiple of 90 degrees.
- Reflection is limited to horizontal and vertical flip.
- Translation is restricted to a 4-KiB page granularity.

Written differently, the functionality of the TILER is to map a 2D virtually addressed interconnect request into one or more physically addressed interconnect requests by:

- Transforming the virtual address, data, and byte-enable to match the requested 0-, 90-, 180-, or 270-degree orientation in a tiled 2D addressing space
- Optionally, translating the oriented tiled address by a page-specific vector to manage memory fragmentation and physical object aliasing

The main features of the TILER are:

- Efficiency improvement of 2D block access on SDRAMs
- Optimized interlaced access on tiled frames
- 2D virtual-to-physical address translation of SDRAM bidimensional objects to handle rotation
- Page-grained address translation to manage memory fragmentation and physical buffer aliasing
- Unlimited number of 2D tiled objects supported in any (0, 90, 180, or 270 degrees) orientation
- Full bandwidth use by minimizing the size of raster-based initiator buffers
- Optimization of multichannel memory transfers
- Interconnect request granularity balance (in X and Y directions)

15.1.3 EMIF Overview

The two external memory interface (EMIF) modules are typically located near the DMM module, as shown in Figure 15-1, *Memory Subsystem Functional Diagram*.

The EMIF module provides connectivity between DDR memory types and manages data bus read/write accesses between external memories and device subsystems which have master access to the L3_MAIN interconnect and DMA capability.

Each EMIF module has the following capabilities:

- Supports JEDEC standard-compliant DDR2-SDRAM and DDR3-SDRAM memory types
- 2-GiB SDRAM address range over one chip-select. This range is configurable through the dynamic memory manager (DMM) module
- Supports SDRAM devices with one, two, four or eight internal banks
- Supports SDRAM devices with single die (one chip select supported)
- Data bus widths:

  - 128-bit L3_MAIN (system) interconnect data bus width
  - 128-bit port for direct connection with MPU subsystem
  - 32-bit SDRAM data bus width
  - 16-bit SDRAM data bus width used in narrow mode
- Supported CAS latencies:
  - DDR3: 5, 6, 7, 8, 9, 10 and 11
  - DDR2: 2, 3, 4, 5, 6 and 7
- Supports 256-, 512-, 1024-, and 2048-word page sizes
- Supported burst length: 8
- Supports sequential burst type
- SDRAM auto initialization from reset or configuration change
- Supports self refresh and power-down modes for low power
- Partial array self-refresh mode for low power when DDR3 is used
- Output impedance (ZQ) calibration for DDR3
- Supports on-die termination (ODT) for DDR2 and DDR3
- Supports prioritized refresh
- Programmable SDRAM refresh rate and backlog counter
- Programmable SDRAM timing parameters
- Write and read leveling/calibration and data eye training for DDR3.
- ECC on the SDRAM data bus for EMIF1 only:
  - 7-bit ECC over 32-bit data
  - 6-bit ECC over 16-bit data when narrow mode is used
  - 1-bit error correction and 2-bit error detection
  - Programmable address ranges to define ECC protected region
  - ECC calculated and stored on all writes to ECC protected address region
  - ECC verified on all reads from ECC protected address region
  - Statistics for 1-bit ECC and 2-bit ECC errors
  - The total width of the ECC DDR data bus is 8 bits

The EMIF modules do not support:

- Burst chop for DDR3
- Interleave burst type
- Auto precharge because of better Bank Interleaving performance
- OCD calibration for DDR2
- CAS Read Latency of 2 and CAS Write Latency of 1 for DDR2
- DLL disabling from EMIF side
- SDRAM devices with more than one die, or topologies which require more than one chip select on a single EMIF channel

15.1.4 GPMC Overview

The General Purpose Memory Controller (GPMC) is an external memory controller of the device. Its data access engine provides a flexible programming model for communication with all standard memories.

The GPMC supports the following various access types:

- Asynchronous read/write access
- Asynchronous read page access (4, 8, and 16 Word16)
- Synchronous read/write access
- Synchronous read/write burst access without wrap capability (4, 8 and 16 Word16)
- Synchronous read/write burst access with wrap capability (4, 8 and 16 Word16)
- Address-data-multiplexed (AD) access
- Address-address-data (AAD) multiplexed access
- Little- and big-endian access

The GPMC can communicate with a wide range of external devices:

- External asynchronous or synchronous 8-bit wide memory or device (non burst device)
- External asynchronous or synchronous 16-bit wide memory or device
- External 16-bit non-multiplexed NOR flash device
- External 16-bit address and data multiplexed NOR Flash device
- External 8-bit and 16-bit NAND flash device
- External 16-bit pseudo-SRAM (pSRAM) device

The main features of the GPMC are:

- 8- or 16-bit-wide data path to external memory device
- Supports up to eight CS regions of programmable size and programmable base addresses in a total address space of 1 GiB
- Supports transactions controlled by a firewall
- On-the-fly error code detection using the Bose-ChaudhurI-Hocquenghem (BCH) (t = 4, 8, or 16) or Hamming code to improve the reliability of NAND with a minimum effect on software (NAND flash with 512-byte page size or greater)
- Fully pipelined operation for optimal memory bandwidth use
- The clock to the external memory is provided from GPMC functional clock divided by 1, 2, 3, or 4
- Supports programmable autoclock gating when no access is detected
- Independent and programmable control signal timing parameters for setup and hold time on a per-chip basis. Parameters are set according to the memory device timing parameters, with a timing granularity of one GPMC functional clock cycle.
- Flexible internal access time control (WAIT state) and flexible handshake mode using external WAIT pin monitoring
- Support bus keeping
- Support bus turnaround
- Prefetch and write posting engine associated with a device DMA controller to achieve full performance from the NAND device with minimum effect on NOR/SRAM concurrent access

Note

Page mode is available only in nonmultiplexed mode.

15.1.5 ELM Overview

In the case of NAND modules with no internal correction capability, sometimes referred to as bare NAND, the correction process can be delegated to the error location module (ELM) used in conjunction with the GPMC.

The ELM supports the following features:

- 4, 8, and 16 bits per 512-byte block error location based on BCH algorithm
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation when error location process completes:
  – When the full page has been processed in page mode
  – For each syndrome polynomial (checksum-like information) in continuous mode

15.1.6 OCM Overview

The on-chip memory (OCM) subsystem consists of the following OCM controllers (OCMC):

- Up to three OCMC connected to an on-chip RAM. Presence of RAM banks is device part number dependent. Refer to device Data Manual for more information.

An OCM Controller supports the following features:

- L3_MAIN data interface:
  – Used for maximum throughput performance
  – 128-bit data bus width
  – Burst supported

- L4 interface:
  - Used for access to configuration registers
  - 32-bit data bus width
  - Only single accesses supported
  - The L4 associated OCMC clock is two times lower than the L3 associated OCMC clock
- Error correction and detection:
  - Single error correction and dual error detection
  - 9-bit Hamming error correction code (ECC) calculated on 128-bit data word which is concatenated with memory address bits
  - Hamming distance of 4
  - Enable/Disable mode control through a dedicated register
  - Single bit error correction on a read transaction
  - Exclusion of repeated addresses from correctable error address trace history
  - ECC valid for all write transactions to an enabled region
  - Sub-128-bit writes supported via read modify write
- ECC Error Status Reporting:
  - Trace history buffer (FIFO) with depth of 4 for corrected error address
  - Trace history buffer with depth of 4 for non correctable error address and also including double error detection
  - Interrupt generation for correctable and uncorrectable detected errors
- ECC Diagnostics Configuration:
  - Counters for single error correction (SEC), double error detection (DED) and address error events (AEE)
  - Programmable threshold registers for exceptions associated with SEC, DED and AEE counters
  - Register control for enabling and disabling of diagnostics
  - Configuration registers and ECC status accessible through L4 interconnect
- Circular buffer for sliced based VIP frame transfers:
  - Up to 12 programmable circular buffers mapped with unique virtual frame addresses
  - On the fly (with no additional latency) address translation from virtual to OCMC circular buffer memory space
  - Virtual frame size up to 8 MiB and circular buffer size up to 1 MiB
  - Error handling and reporting of illegal CBUF addressing
  - Underflow and Overflow status reporting and error handling
  - Last access read/write address history
- Two Interrupt outputs configured independently to service either ECC or CBUF interrupt events

The OCM controllers do not have a memory protection logic and do not support endianism conversion.

15.2 Dynamic Memory Manager

15.2.1 DMM Overview

The function of the dynamic memory manager (DMM) is to:

- Add initiator-based priority to any incoming requests
- Perform to-and-from tiling and subtiling conversions of tiled requests
- Make on-the-fly basic transforms, such as quadrant rotations and mirroring
- Optionally provide a low-latency page-based translation to handle memory fragmentation – memory management unit (MMU)
- Distribute the traffic on all attached memory controllers according to the interleaving configuration

The DMM is introduced also in Section 15.1.1, *DMM Overview* of Section 15.1, *Memory Subsystem Overview*.

The functions of the TILER are:

- Primary handling efficiently 2-dimensional (2D) data mapped in tiles, such as video or graphics macroblocks
- Optionally managing the memory fragmentation and zero-copy physical frame buffers swapping through a page-grained translation
- Allowing optimized interlaced accesses on tiled frames
- Making (distance preserving) transforms, such as 90-, 180-, or 270-degree rotations, with a horizontal or vertical reflection
- Interleaving the memory accesses among the two memory controllers (also called EMIF).

The TILER is also introduced in Section 15.1.2, *TILER Overview* of Section 15.1, *Memory Subsystem Overview*.

Figure 15-2 is an overview of the DMM and TILER in the device.

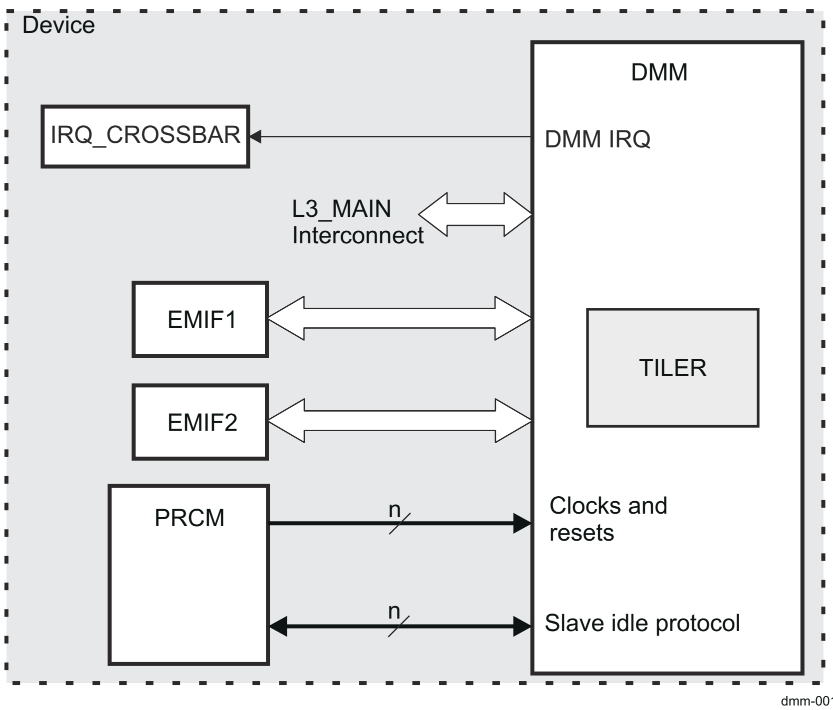


Figure 15-2. DMM and Tiler Overview

15.2.2 DMM Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

- Slave idle protocol (the DMM supports only smart-idle mode)

- No master standby protocol
- No wake-up request
- No system direct memory access (sDMA) requests
- One interrupt line for interrupt request (IRQ)
- One functional clock

Figure 15-3 shows the integration of DMM in the device.

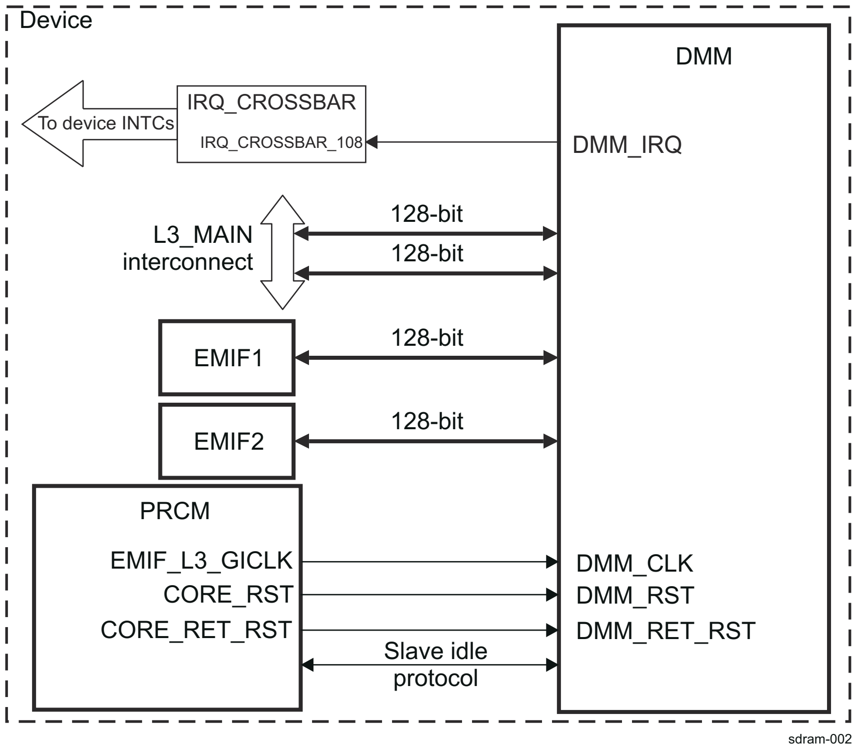


Figure 15-3. DMM Integration

Note

For more information about the slave idle protocol, see *Power, Reset, and Clock Management*.

Table 15-1 through Table 15-3 summarize the integration of DMM in the device.

Table 15-1. DMM Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
DMM	PD_COREAON	No	L3_MAIN

Table 15-2. DMM Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DMM	DMM_CLK	EMIF_L3_GICLK	PRCM	DMM interface and functional clock. For information about power, reset, and clock management (PRCM) module clock gating and management, see *Power, Reset, and Clock Management*.
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description

Table 15-2. DMM Clocks and Resets (continued)

DMM	DMM_RST	CORE_RST	PRCM	Functional reset. For information about PRCM reset sources and distribution, see *Power, Reset, and Clock Management*.
DMM	DMM_RET_RST	CORE_RET_RST	PRCM	Reset for the following registers: DMM_SYSCONFIG DMM_LISA_LOCK DMM_LISA_MAP_i DMM_TILER_OR0 DMM_TILER_OR1 DMM_PAT_VIEW0 DMM_PAT_VIEW1 DMM_PAT_VIEW_MAP_i DMM_PAT_VIEW_MAP_BASE DMM_PAT_DESCR_i DMM_PAT_AREA_i DMM_PAT_CTRL_i DMM_PAT_DATA_i DMM_PEG_PRIO_k DMM_PEG_PRIO_PAT For information about PRCM reset sources and distribution, see *Power, Reset, and Clock Management*.

Table 15-3. DMM Hardware Requests

Module Instance	Source Signal Name	IRQ_CROSSBAR	Destination Signal Name	Description
Interrupt Requests				
DMM	DMM_IRQ	IRQ_CROSSBAR_108	MPU_IRQ_113	DMM interrupt to IRQ_CROSSBAR
	DMM_IRQ	IRQ_CROSSBAR_108	IPU_IRQ_64	DMM interrupt to IRQ_CROSSBAR
No DMA Requests				

15.2.2.1 DMM Configuration

Table 15-4 lists all configured parameters for the DMM. These parameters are read-only.

Table 15-4. DMM TILER Container Geometry

Scope	Bit field	Configuration Value	Description
LISA	DMM_HWINFO[3:0] TILER_CNT	0x2	Two TILER instances in the DMM
	DMM_HWINFO[19:16] ROBIN_CNT	0x2	Two ROBIN instances in the DMM.
	DMM_LISA_HWINFO[4:0] SECTION_CNT	0x4	Four DMM sections
	DMM_LISA_HWINFO[11:8] SDRC_CNT	0x2	Two SDRAM controllers (EMIF) attached.
TILER	DMM_TILER_HWINFO[6:0] OR_CNT	0x10	16 orientation entries
PAT	DMM_PAT_GEOMETRY[4:0] PAGE_SZ	0x1	4-KiB page granularity
	DMM_PAT_GEOMETRY[19:16] CONT_WDTH	0x8	Container width of 256 pages
	DMM_PAT_GEOMETRY[26:24] CONT_HGHT	0x8	Container height of 256 pages
	DMM_PAT_GEOMETRY[13:8] ADDR_RANGE	0x10	2-GiB PAT output physical address range
	DMM_PAT_HWINFO[6:0] VIEW_CNT	0x10	16 view entries
	DMM_PAT_HWINFO[11:8] VIEW_MAP_CNT	0x4	Four view maps
	DMM_PAT_HWINFO[20:16] LUT_CNT	0x1	One PAT LUT

Table 15-4. DMM TILER Container Geometry (continued)

Scope	Bit field	Configuration Value	Description
	DMM_PAT_HWINFO[28:24] ENGINE_CNT	0x4	Four PAT refill engines
PEG	DMM_PEG_HWINFO[6:0] PRIO_CNT	0x40	64 priority entries

15.2.3 DMM Functional Description

15.2.3.1 DMM Block Diagram

shows the DMM macro architecture. The DMM consists of six blocks:

- Two TILERs, each with its own interconnect slave port for converting requests back and forth between the input virtual addressing mode and the output physical tiled addressing mode. The tiling conversions of requests, write data, and responses is performed entirely in the TILER blocks.
- Two reordering buffer and initiator nodes (ROBINs), each with its own interconnect master port to initiate requests to the SDRC and allow tiled data, tiled response, and split response reconstruction. The ROBIN block manages only the reordering buffer and performs data reordering due to the orientation.
- A physical address translator (PAT) for managing the memory fragmentation
- A priority extension generator (PEG) to generate priorities required by the SDRC; these priorities are not used in the DMM.
- A local interconnect and synchronization agent (LISA) to synchronize all DMM subsystems and provide access to their configuration registers

Figure 15-4 is a block diagram of the DMM.

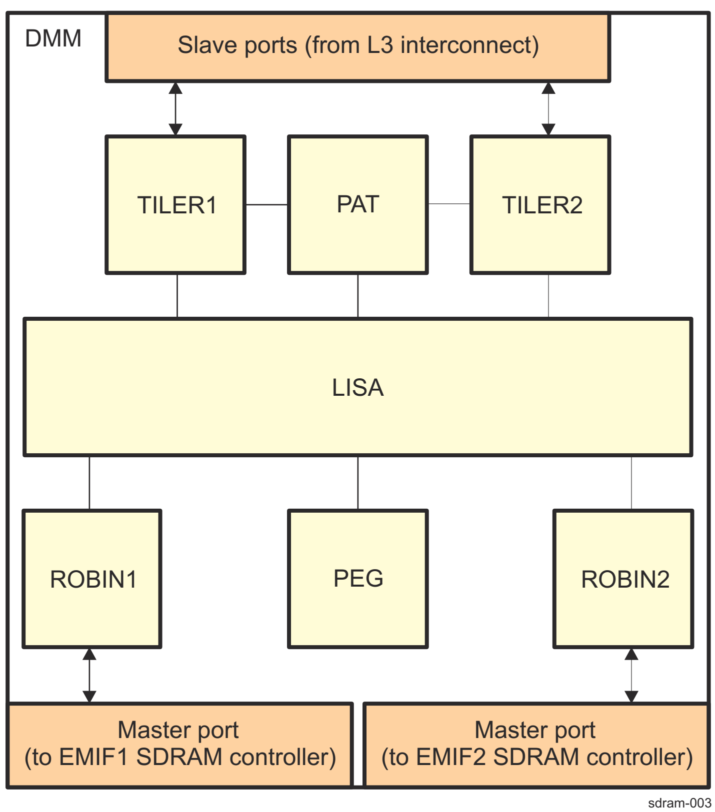


Figure 15-4. DMM Block Diagram

CAUTION

The interconnect must ensure that virtually addressed requests target only a TILER port.

15.2.3.2 DMM Clock Configuration

Table 15-5 describes the DMM clocks.

Table 15-5. DMM Clocks

Signal	I/O(1)	Description
DMM_CLK	I	Functional and interface clock

(1) I = Input; O = Output

The DMM is concerned with SDRC management and is in the MEMIF clock domain among the SDRCs. The DMM is a fully synchronous module, which uses the clock and clock-enable signals provided in the MEMIF clock domain to generate its interface and functional clocks.

To configure DMM_CLK control and settings, see Table 15-2.

15.2.3.3 DMM Power Management

DMM power is supplied by the CORE power domain, and DMM power management complies with system power-management guidelines.

Table 15-6 describes the power-management features available for the DMM module.

Table 15-6. DMM Local Power-Management Features

Feature	Registers	Description
Slave idle modes	DMM_SYSCONFIG[3:2] SIDLEMODE bit field	Only smart-idle wake-up mode is available.

Some of the DMM module registers are affected by the save-and-restore (SAR) mechanism. For more information about SAR, see *Power, Reset, and Clock Management*.

15.2.3.4 DMM Interrupt Requests

Errors in PAT area refill registers are reported through the DMM_PAT_STATUS_i[15:10] ERROR bit field (see Table 15-7).

Table 15-7. DMM Hardware Status Features

Feature	Type	Register	Description
PAT error flags	Read-only	DMM_PAT_STATUS_i[15:10] ERROR (where i = 0 to 3)	Unexpected update of the PAT area refill registers

Table 15-8 lists the event flags, and their masks, that can cause module interrupts.

Table 15-8. Events

Interrupt	Event Flag	Event Mask
ERR_LUT_MISS3	DMM_PAT_IRQSTATUS[31] ERR_LUT_MISS3 DMM_PAT_IRQSTATUS_RAW[31] ERR_LUT_MISS3	DMM_PAT_IRQENABLE_SET[31] ERR_LUT_MISS3 DMM_PAT_IRQENABLE_CLR[31] ERR_LUT_MISS3
ERR_UPD_DATA3	DMM_PAT_IRQSTATUS[30] ERR_UPD_DATA3 DMM_PAT_IRQSTATUS_RAW[30] ERR_UPD_DATA3	DMM_PAT_IRQENABLE_SET[30] ERR_UPD_DATA3 DMM_PAT_IRQENABLE_CLR[30] ERR_UPD_DATA3
ERR_UPD_CTRL3	DMM_PAT_IRQSTATUS[29] ERR_UPD_CTRL3 DMM_PAT_IRQSTATUS_RAW[29] ERR_UPD_CTRL3	DMM_PAT_IRQENABLE_SET[29] ERR_UPD_CTRL3 DMM_PAT_IRQENABLE_CLR[29] ERR_UPD_CTRL3

Table 15-8. Events (continued)

Interrupt	Event Flag	Event Mask
ERR_UPD_AREA3	DMM_PAT_IRQSTATUS[28] ERR_UPD_AREA3 DMM_PAT_IRQSTATUS_RAW[28] ERR_UPD_AREA3	DMM_PAT_IRQENABLE_SET[28] ERR_UPD_AREA3 DMM_PAT_IRQENABLE_CLR[28] ERR_UPD_AREA3
ERR_INV_DATA3	DMM_PAT_IRQSTATUS[27] ERR_INV_DATA3 DMM_PAT_IRQSTATUS_RAW[27] ERR_INV_DATA3	DMM_PAT_IRQENABLE_SET[27] ERR_INV_DATA3 DMM_PAT_IRQENABLE_CLR[27] ERR_INV_DATA3
ERR_INV_DSC3	DMM_PAT_IRQSTATUS[26] ERR_INV_DSC3 DMM_PAT_IRQSTATUS_RAW[26] ERR_INV_DSC3	DMM_PAT_IRQENABLE_SET[26] ERR_INV_DSC3 DMM_PAT_IRQENABLE_CLR[26] ERR_INV_DSC3
FILL_LST3	DMM_PAT_IRQSTATUS[25] FILL_LST3 DMM_PAT_IRQSTATUS_RAW[25] FILL_LST3	DMM_PAT_IRQENABLE_SET[25] FILL_LST3 DMM_PAT_IRQENABLE_CLR[25] FILL_LST3
FILL_DSC3	DMM_PAT_IRQSTATUS[24] FILL_DSC3 DMM_PAT_IRQSTATUS_RAW[24] FILL_DSC3	DMM_PAT_IRQENABLE_SET[24] FILL_DSC3 DMM_PAT_IRQENABLE_CLR[24] FILL_DSC3
ERR_LUT_MISS2	DMM_PAT_IRQSTATUS[23] ERR_LUT_MISS2 DMM_PAT_IRQSTATUS_RAW[23] ERR_LUT_MISS2	DMM_PAT_IRQENABLE_SET[23] ERR_LUT_MISS2 DMM_PAT_IRQENABLE_CLR[23] ERR_LUT_MISS2
ERR_UPD_DATA2	DMM_PAT_IRQSTATUS[22] ERR_UPD_DATA2 DMM_PAT_IRQSTATUS_RAW[22] ERR_UPD_DATA2	DMM_PAT_IRQENABLE_SET[22] ERR_UPD_DATA2 DMM_PAT_IRQENABLE_CLR[22] ERR_UPD_DATA2
ERR_UPD_CTRL2	DMM_PAT_IRQSTATUS[21] ERR_UPD_CTRL2 DMM_PAT_IRQSTATUS_RAW[21] ERR_UPD_CTRL2	DMM_PAT_IRQENABLE_SET[21] ERR_UPD_CTRL2 DMM_PAT_IRQENABLE_CLR[21] ERR_UPD_CTRL2
ERR_UPD_AREA2	DMM_PAT_IRQSTATUS[20] ERR_UPD_AREA2 DMM_PAT_IRQSTATUS_RAW[20] ERR_UPD_AREA2	DMM_PAT_IRQENABLE_SET[20] ERR_UPD_AREA2 DMM_PAT_IRQENABLE_CLR[20] ERR_UPD_AREA2
ERR_INV_DATA2	DMM_PAT_IRQSTATUS[19] ERR_INV_DATA2 DMM_PAT_IRQSTATUS_RAW[19] ERR_INV_DATA2	DMM_PAT_IRQENABLE_SET[19] ERR_INV_DATA2 DMM_PAT_IRQENABLE_CLR[19] ERR_INV_DATA2
ERR_INV_DSC2	DMM_PAT_IRQSTATUS[18] ERR_INV_DSC2 DMM_PAT_IRQSTATUS_RAW[18] ERR_INV_DSC2	DMM_PAT_IRQENABLE_SET[18] ERR_INV_DSC2 DMM_PAT_IRQENABLE_CLR[18] ERR_INV_DSC2
FILL_LST2	DMM_PAT_IRQSTATUS[17] FILL_LST2 DMM_PAT_IRQSTATUS_RAW[17] FILL_LST2	DMM_PAT_IRQENABLE_SET[17] FILL_LST2 DMM_PAT_IRQENABLE_CLR[17] FILL_LST2
FILL_DSC2	DMM_PAT_IRQSTATUS[16] FILL_DSC2 DMM_PAT_IRQSTATUS_RAW[16] FILL_DSC2	DMM_PAT_IRQENABLE_SET[16] FILL_DSC2 DMM_PAT_IRQENABLE_CLR[16] FILL_DSC2
ERR_LUT_MISS1	DMM_PAT_IRQSTATUS[15] ERR_LUT_MISS1 DMM_PAT_IRQSTATUS_RAW[15] ERR_LUT_MISS1	DMM_PAT_IRQENABLE_SET[15] ERR_LUT_MISS1 DMM_PAT_IRQENABLE_CLR[15] ERR_LUT_MISS1
ERR_UPD_DATA1	DMM_PAT_IRQSTATUS[14] ERR_UPD_DATA1 DMM_PAT_IRQSTATUS_RAW[14] ERR_UPD_DATA1	DMM_PAT_IRQENABLE_SET[14] ERR_UPD_DATA1 DMM_PAT_IRQENABLE_CLR[14] ERR_UPD_DATA1
ERR_UPD_CTRL1	DMM_PAT_IRQSTATUS[13] ERR_UPD_CTRL1 DMM_PAT_IRQSTATUS_RAW[13] ERR_UPD_CTRL1	DMM_PAT_IRQENABLE_SET[13] ERR_UPD_CTRL1 DMM_PAT_IRQENABLE_CLR[13] ERR_UPD_CTRL1

Table 15-8. Events (continued)

Interrupt	Event Flag	Event Mask
ERR_UPD_AREA1	DMM_PAT_IRQSTATUS[12] ERR_UPD_AREA1 DMM_PAT_IRQSTATUS_RAW[12] ERR_UPD_AREA1	DMM_PAT_IRQENABLE_SET[12] ERR_UPD_AREA1 DMM_PAT_IRQENABLE_CLR[12] ERR_UPD_AREA1
ERR_INV_DATA1	DMM_PAT_IRQSTATUS[11] ERR_INV_DATA1 DMM_PAT_IRQSTATUS_RAW[11] ERR_INV_DATA1	DMM_PAT_IRQENABLE_SET[11] ERR_INV_DATA1 DMM_PAT_IRQENABLE_CLR[11] ERR_INV_DATA1
ERR_INV_DSC1	DMM_PAT_IRQSTATUS[10] ERR_INV_DSC1 DMM_PAT_IRQSTATUS_RAW[10] ERR_INV_DSC1	DMM_PAT_IRQENABLE_SET[10] ERR_INV_DSC1 DMM_PAT_IRQENABLE_CLR[10] ERR_INV_DSC1
FILL_LST1	DMM_PAT_IRQSTATUS[9] FILL_LST1 DMM_PAT_IRQSTATUS_RAW[9] FILL_LST1	DMM_PAT_IRQENABLE_SET[9] FILL_LST1 DMM_PAT_IRQENABLE_CLR[9] FILL_LST1
FILL_DSC1	DMM_PAT_IRQSTATUS[8] FILL_DSC1 DMM_PAT_IRQSTATUS_RAW[8] FILL_DSC1	DMM_PAT_IRQENABLE_SET[8] FILL_DSC1 DMM_PAT_IRQENABLE_CLR[8] FILL_DSC1
ERR_LUT_MISS0	DMM_PAT_IRQSTATUS[7] ERR_LUT_MISS0 DMM_PAT_IRQSTATUS_RAW[7] ERR_LUT_MISS0	DMM_PAT_IRQENABLE_SET[7] ERR_LUT_MISS0 DMM_PAT_IRQENABLE_CLR[7] ERR_LUT_MISS0
ERR_UPD_DATA0	DMM_PAT_IRQSTATUS[6] ERR_UPD_DATA0 DMM_PAT_IRQSTATUS_RAW[6] ERR_UPD_DATA0	DMM_PAT_IRQENABLE_SET[6] ERR_UPD_DATA0 DMM_PAT_IRQENABLE_CLR[6] ERR_UPD_DATA0
ERR_UPD_CTRL0	DMM_PAT_IRQSTATUS[5] ERR_UPD_CTRL0 DMM_PAT_IRQSTATUS_RAW[5] ERR_UPD_CTRL0	DMM_PAT_IRQENABLE_SET[5] ERR_UPD_CTRL0 DMM_PAT_IRQENABLE_CLR[5] ERR_UPD_CTRL0
ERR_UPD_AREA0	DMM_PAT_IRQSTATUS[4] ERR_UPD_AREA0 DMM_PAT_IRQSTATUS_RAW[4] ERR_UPD_AREA0	DMM_PAT_IRQENABLE_SET[4] ERR_UPD_AREA0 DMM_PAT_IRQENABLE_CLR[4] ERR_UPD_AREA0
ERR_INV_DATA0	DMM_PAT_IRQSTATUS[3] ERR_INV_DATA0 DMM_PAT_IRQSTATUS_RAW[3] ERR_INV_DATA0	DMM_PAT_IRQENABLE_SET[3] ERR_INV_DATA0 DMM_PAT_IRQENABLE_CLR[3] ERR_INV_DATA0
ERR_INV_DSC0	DMM_PAT_IRQSTATUS[2] ERR_INV_DSC0 DMM_PAT_IRQSTATUS_RAW[2] ERR_INV_DSC0	DMM_PAT_IRQENABLE_SET[2] ERR_INV_DSC0 DMM_PAT_IRQENABLE_CLR[2] ERR_INV_DSC0
FILL_LST0	DMM_PAT_IRQSTATUS[1] FILL_LST0 DMM_PAT_IRQSTATUS_RAW[1] FILL_LST0	DMM_PAT_IRQENABLE_SET[1] FILL_LST0 DMM_PAT_IRQENABLE_CLR[1] FILL_LST0
FILL_DSC0	DMM_PAT_IRQSTATUS[0] FILL_DSC0 DMM_PAT_IRQSTATUS_RAW[0] FILL_DSC0	DMM_PAT_IRQENABLE_SET[0] FILL_DSC0 DMM_PAT_IRQENABLE_CLR[0] FILL_DSC0

15.2.3.5 DMM

15.2.3.5.1 DMM Concepts

The DMM introduces the concepts of dynamic mapping and DMM atomic units.

15.2.3.5.1.1 Dynamic Mapping

The DMM manages its internal memory space as an ordered set of up to four sections. Figure 15-5 shows the DMM sections and memory mapping.

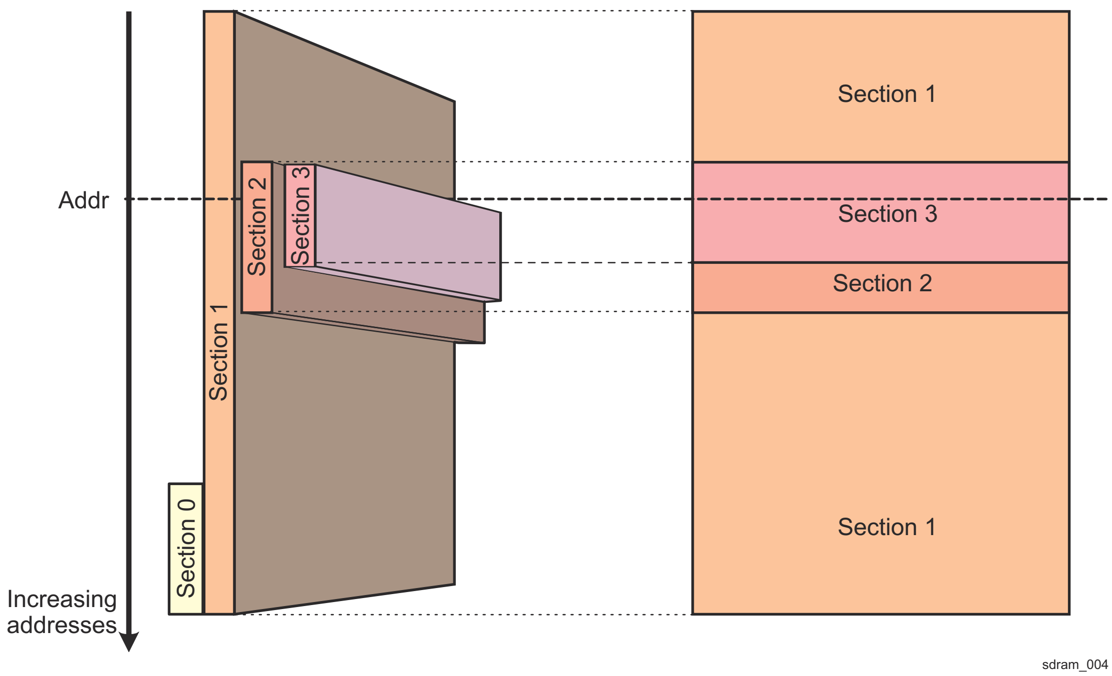


Figure 15-5. DMM Sections and Memory Mapping

In the DMM, a section is:

- A segment of 16MiB to 2GiB, which is power-of-two in size and aligned to that size in the system map
- An area with a constant interleaving scheme: constant interleaving granularity on a constant set of SDRC (EMIF) targets
- Given a priority equal to its index: the higher the index, the higher the priority

Note

Whenever a request hits more than one DMM section, it follows the interleaving scheme of the section having the highest index.

The interleaving configuration of the DMM, its section-based dynamic memory mapping, is shown on the right in Figure 15-5. In this example, a request at the system address Addr follows the interleaving scheme of section 3, although it hits sections 1, 2, and 3. Similarly, the DMM configuration given in this example prevents any request from using the interleaving scheme of section 0, because section 0 is fully masked by section 1, which has a higher priority.

Each of the four sections is configured through a DMM_LISA_MAP_i register, where i = 0 to 3.

Note

The DMM and EMIF registers (see *EMIF Controller*) are declared in two extra static DMM sections of the highest priority so that they cannot be masked by any standard programmable DMM section.

Note

The DMM atomic size is:

- 1KiB in noninterleaved sections
- Equal to the interleaving granularity (128, 256, or 512 bytes) in interleaved sections

15.2.3.5.1.2 Address Mapping

The address mapping inside the DMM is configurable through up to four sections. A DMM section description fits in a single register (DMM_LISA_MAP_i). Each section is defined based on:

- Its system address: The base address of the decoding range for the section
- Its size: The encoding is the number of bits used in the upper 8 bits of the incoming system address
- Its physical address: The base address of the memory range access in the external memory controller
- Its address space: The address space used on the external memory controller (for the DMM) when hitting this section. For details, see *Local Interface* in *EMIF Controller*.
- The target memory controller. A section may hit a single or pair of controllers.
- Its interleaving definition

The address decoding is priority-based. In case of overlapping sections, only the highest-order one is hit. Register memory spaces have priority over regular memory sections. In case of four sections, the priority order is therefore:

1. Registers
2. DMM_LISA_MAP_3
3. DMM_LISA_MAP_2
4. DMM_LISA_MAP_1
5. DMM_LISA_MAP_0

All register-related addresses are reserved and fixed in the overall address mapping:

- DMM registers: 0x4E00_0000 to 0x4FFF_FFFF
- EMIF1 registers: 0x4C00_0000 to 0x4CFF_FFFF
- EMIF2 registers: 0x4D00_0000 to 0x4DFF_FFFF

There is no overlapping between register sections.

Note

Section decoding happens after PAT address translation in the case of TILER. In non-bypass mode, the system address considered for TILER accesses is the virtual address computed based on the PAT translation tables.

The DMM_LISA_LOCK register is used to lock the configuration once set. If written to 1, the LOCK bit prevents further writes to all DMM_LISA_MAP_i registers. The LOCK bit cannot be written back to 0. A reset is required to reprogram the sections.

15.2.3.5.1.3 Address Translation

The PAT engine of the DMM is composed of a 32-k entry physical address translation vector table and one or two refill engines. The refill engine is a specialized DMA for refilling the content of the PAT table.

The address translation mechanism is available only when the incoming request hits a page mode or tiled mode container; that is, when the incoming address targets the TILER or its aliased view in the system addressing space. Otherwise, the PAT logic is bypassed so that the resulting physical address corresponds to the input address.

The PAT engine supports multiple address translation schemes, called views, which can be bound to one or more initiators through a view mapping mechanism.

15.2.3.5.1.3.1 PAT View Mappings

The PAT engine can have up to 16 groups of initiators that share a set of four PAT views. The connection from an initiator to a PAT view is made through the DMM_PAT_VIEW register. Given that each PAT view index is coded on 4 bits, the DMM_PAT_VIEW register is a 64-bit register split into two 32-bit registers (DMM_PAT_VIEW0 for the first eight PAT view indexes and DMM_PAT_VIEW1 for the last eight PAT view indexes.

The PAT view index that corresponds to the initiator having the value i as the 4 most-significant bits (MSBs) of its L3 ConnID uses the view referenced in entry i of the DMM_PAT_VIEW. For example, the initiator 0xC7 uses the thirteenth view index of the DMM_PAT_VIEW register, the fifth view index in the DMM_PAT_VIEW1 register.

The PAT view index of the initiator i is found in the Vi field. The Wi field is aimed at writing the corresponding Vi. When writing to the DMM_PAT_VIEW registers, the only Vi view indexes that are updated are those having their corresponding Wi bit, and byte enable, set. The Wi bits are always read as 0. For instance, to set the PAT view indexes V3 and V7 to 2 and 1, respectively, the DMM_PAT_VIEW0 register must be written with 5000 6000h.

15.2.3.5.1.3.2 PAT View Map Base Address

The PAT view map base address defines the base address of all PAT translated addresses.

Bit [31] of all PAT translated addresses is set to BASE_ADDR. For example, if the DMM_PAT_VIEW_MAP_BASE register is set to 8000 0000h, all PAT translated addresses will have bit [31] set to 1 so that the translated addresses range from 8000 0000h to FFFF FFFFh. All reserved bits of this DMM PAT base address register are read to 0.

15.2.3.5.1.3.3 PAT Views

A PAT view defines the kind of physical address translation to perform for each mode accesses (page, 8-bit, 16-bit and 32-bit). Each mode of each PAT view can be configured to use a container-grained translation or page-mode translation.

15.2.3.5.1.3.3.1 PAT Direct Access Translation

The container-grained translation is named the direct access. In this mode, the translation vector is given directly by the CONT_x bit field that corresponds to the accessed mode.

- A page mode access uses the vector contained in the DMM_PAT_VIEW_MAP_i[26:24] CONT_PAGE bit field concatenated with 1 (bit 27 of translated address).
- A 32-bit mode access uses the vector contained in the DMM_PAT_VIEW_MAP_i[18:16] CONT_32 bit field concatenated with 0 (bit 27 of translated address).
- A 16-bit mode access uses the vector contained in the DMM_PAT_VIEW_MAP_i[10:8] CONT_16 bit field concatenated with 0 (bit 27 of translated address).
- An 8-bit mode access uses the vector contained in the DMM_PAT_VIEW_MAP_i[2:0] CONT_8 bit field concatenated with 0 (bit 27 of translated address).

Figure 15-6, *PAT Direct Access Translation* describes PAT direct access translation .

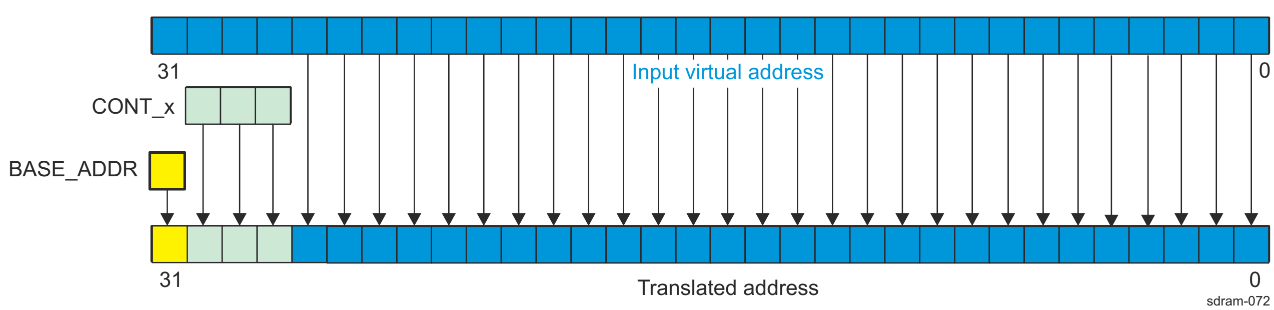


Figure 15-6. PAT Direct Access Translation

15.2.3.5.1.3.3.2 PAT Indirect Access Translation

The page-grained translation is named the indirect access. In this mode the translation vector is found in the internal 32-k entry physical address translation vector table at the index given by bits [26:12] of the input virtual address, and the DMM_PAT_VIEW_MAP_i CONT_x bit field references the internal physical address translation table to use. Because the DMM uses only one such table, in this mode the CONT_x bit field must be written as 0. See Figure 15-7.

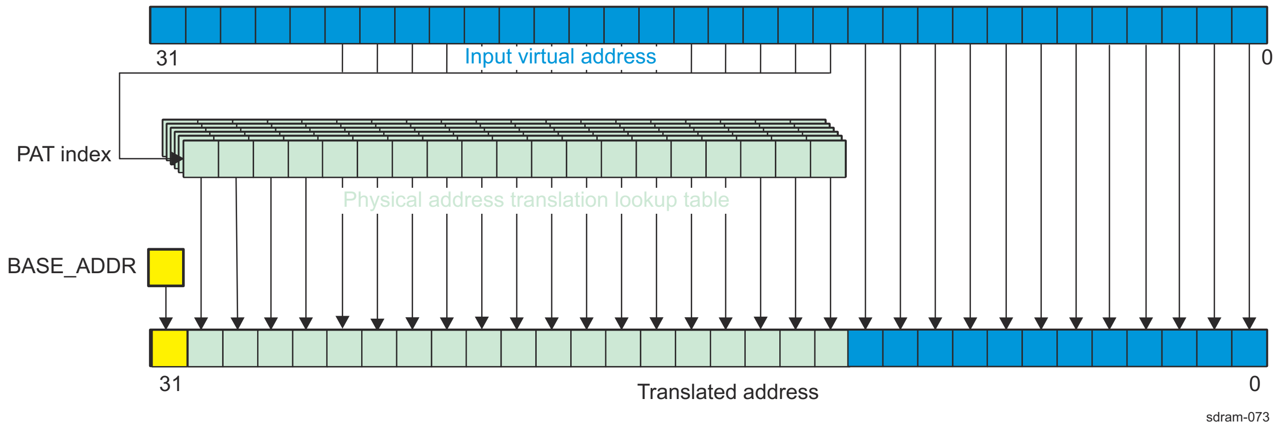


Figure 15-7. PAT Indirect Access Translation

Each entry of the PAT lookup table (LUT) is a 19-bit vector that replaces bits [30:12] of the input virtual address. The PAT index aimed at selecting the vector in the table consists of bits [26:12] of the input virtual address.

The mode associated to the transaction is used to define if the upper or lower half of the LUT is effectively used:

- If the mode is 8-, 16- or 32-bit, the lower part of the LUT (indexes 0x0000 to 0x7FFF; that is, lines 0 to 127) is used.
- If the mode is page, the upper part of the LUT (indexes 0x8000 to 0xFFFF; that is, ines 128 to 256) is used.

This means the PAT index used is the concatenation of 0 and address bits 12 to 26 in 8-, 16- or 32-bit mode, and the concatenation of 1 and address bits 12 to 26 in page mode.

Using different LUT indexes depending on the mode enables the user to define a larger tiled space, with the constraint that half of the space can be used only in 8-, 16- or 32-bit mode and the other half only in page mode. If the user wishes to preserve software compatibility with the case CONT_HGHT = 128, it is required to mirror the configuration for the lower half of the LUT on the upper half of the LUT. In that way, all views see the same address decoding through PAT.

15.2.3.5.1.3.3.3 PAT View Configuration

There are four different views in a DMM, each with its own DMM_PAT_VIEW_MAP_i register for defining the kind of address translation to perform.

The PAT view type of each mode is selected by the ACCESS_x bit field in the DMM_PAT_VIEW_MAP_i register, i being the index of the considered view. When this field is set to 1, the indirect access scheme is used; otherwise, the PAT performs the address translation in a direct way with the corresponding CONT_x vector.

When configuring a mode in a view to use the indirect access, the corresponding CONT_x bit field must be filled with 0.

15.2.3.5.1.3.3.4 PAT Address Translation LUT

The PAT LUT does not have the same geometry as the DMM container. The PAT LUT has 256 lines of 256 entries of 19 bits each. The geometry of the DMM container is 256 × 128 entries.

- The lower 128MiB are restricted for use with 8-, 16-, and 32-bit modes.
- The upper 128MiB are restricted for use with page mode.

See Figure 15-8.

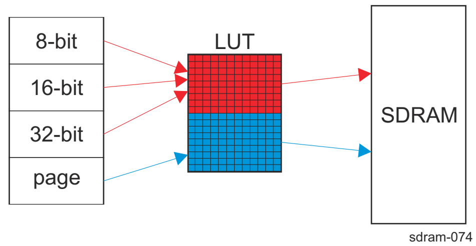


Figure 15-8. Physical Address Translation Table

15.2.3.5.1.3.3.5 Direct Access to the PAT Table Vectors

The PAT table is typically refilled with specialized DMA called refill engines. Some direct read and write access to the content of this table is granted when disabling the use of one or the other refill engine by writing 1 in the MODEi field of the DMM_PAT_CONFIG register that corresponds to the refill engine i to disable.

In this mode, often called the debug mode, the data read in or written to the DMM_PAT_DATA_i register corresponds to the vector in the PAT table, whose index is in the X0 and Y0 fields of the DMM_PAT_AREA_i register.

15.2.3.5.1.3.3.6 Automatic Refill Through the Refill Engines

See Section 15.2.4.1, *PAT Use Cases*.

15.2.3.5.2 DMM Transaction Flows

15.2.3.5.2.1 Nontiled Transaction Flow

Each nontiled interconnect transaction that reaches the DMM on a TILER port is subject to the same processing.

The TILER blocks consider separated request, data, and response paths. An overview of each path follows and more detailed information is in Section 15.2.3.5.3, *DMM Internal Macro-Architecture*.

On the request path, the flow consists of:

- Allocating a TILER response context for the timely generation of the appropriate responses
- Splitting 2D requests in a collection of 1D requests – TILER ports
- Splitting requests at DMM unit boundaries; the split granularity is provided by the LISA mapping registers
- Allocating an available buffer in the appropriate ROBIN, for both read and write requests
- In case of a write request, allocating and updating a TILER write context to direct the incoming write data into the relevant reordering buffer
- Generating the initiator-indexed priority extension by use of the PEG block

On the write data path, the flow consists of forwarding incoming data to the relevant reordering buffer in accordance with the corresponding TILER write context.

On the response path, responses are returned when:

- At least one response has entered each related buffer, in case of read requests.
- All related responses have returned from the SDRC, in case of write requests.
- No other previous pending response with the same tag exists.

15.2.3.5.2.2 Tiled Transaction Flow

Similarly, each tiled interconnect transaction that reaches the DMM is subject to the same processing.

The TILER blocks consider separated request, data, and response paths. An overview of each path follows, and more detailed information is in Section 15.2.3.5.3, *DMM Internal Macro-Architecture*.

On the request path, the processing phase consists of:

- Decoding the address to qualify whether the request targets the TILER or the memory directly
- Transforming TILER-specific requests to their natural representation; the address, width, and height are modified accordingly

On the request path, the generation phase consists of:

- Allocating a TILER response context for the timely generation of the appropriate responses
- Splitting malformed: The request stride differs from the container stride and from the double of this container stride – tiled 2D requests in a collection of 1D requests
- Splitting tiled requests at tile boundaries
- Performing the page-based address translation by use of the PAT block
- Allocating an available buffer in the appropriate ROBIN, for read and write requests
- In case of a write request, allocating and updating a TILER write context to direct the incoming write data into the relevant reordering buffer
- Generating the initiator-indexed priority extension by use of the PEG block

On the data path, any incoming data is transformed in accordance with the corresponding TILER write context and sent to the appropriate reordering buffer.

On the response path, responses are returned when:

- A minimal number of responses have entered the corresponding buffers, in case of read requests.
- All related responses have returned from the SDRCs, in case of write requests.
- No other previous pending response with the same tag exists.

15.2.3.5.3 DMM Internal Macro-Architecture

This section describes the DMM internal macro-architecture, specifically:

- Input request decoding
- PAT and synchronized translation table reloading
- Output request and response generation
- Memory mapping in the system addressing space, including interleaving
- Tag management
- Priority handling
- Contexts
- Maximum allowed burst size
- Reconstruction buffer dimensioning

15.2.3.5.3.1 LISA Description

The LISA is a full crossbar aimed at setting priorities, managing tags, and mapping memory.

The interconnect routes are:

- TILER requests on the ROBIN initiator nodes
- TILER write data on the ROBIN write buffers
- ROBIN read data to the relevant TILER initiators

The LISA block registers are DMM_LISA_MAP_i (where i = 0 to 3) and DMM_LISA_LOCK.

When two ROBINs are in use, the two ports are interleaved at a programmable boundary from 128 bytes or more (configurable with the DMM_LISA_MAP_i[19:18] SDRC_INTL bit field).

15.2.3.5.3.2 PAT Description

The PAT block maps physical pages to each TILER container page. The internal address translation memories used in the PAT block are designed with RFFs. It consists of:

- A memory-based LUT which has the same geometry as the container pages
- A refill engine for modifying entries in a given area of the internal LUT

A PAT descriptor is a singly-linked list node (see Figure 15-9) that contains:

- A description of the LUT area to reload
- A description of how to reload the defined LUT area
- A pointer to the location where the corresponding LUT entries are stored

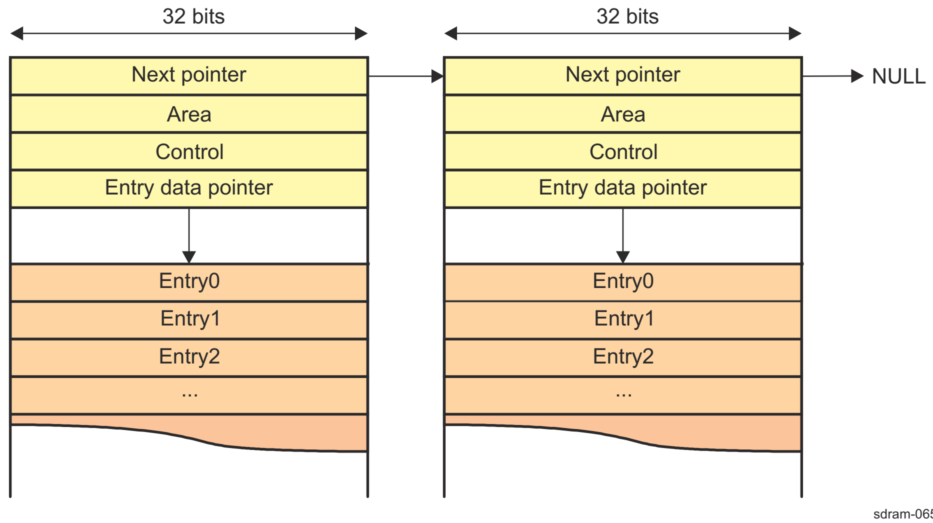


Figure 15-9. PAT Descriptors

PAT descriptors are chained and processed until a NULL pointer is encountered. The PAT bank allocation scheme allows the updating of four consecutive entries of a line in a single cycle regardless of the refilling orientation.

The PAT descriptor structure directly maps the following registers (where i = 0 to 3):

- DMM_PAT_DESCR_i
- DMM_PAT_AREA_i
- DMM_PAT_CTRL_i
- DMM_PAT_DATA_i

The PAT refill engine can be started either of the following:

- Filling manually all the necessary registers:
  – DMM_PAT_AREA_i with the (x0, y0) (x1, y1) area definition
  – DMM_PAT_DATA_i with the physical address of the corresponding area entry table
  – DMM_PAT_CTRL_i[6:4] DIRECTION bit field with the relevant (S, Y, X) direction of the area refill
  – DMM_PAT_CTRL_i[0] START bit with 1
- Writing the physical address of a memory-mapped PAT descriptor in DMM_PAT_DESCR_i register, which updates:
  – DMM_PAT_AREA_i with the area value of the descriptor
  – DMM_PAT_DATA_i with the data value of the descriptor
  – DMM_PAT_CTRL_i with the control value of the descriptor
  – DMM_PAT_DESCR_i with the next value of the descriptor

The data part of the PAT refill starts only when the DMM_PAT_CTRL_i[0] START bit is asserted.

The DMM_PAT_STATUS_i register can be used to determine whether the process has completed without errors.

15.2.3.5.3.3 PEG Description

The PEG is a dynamic software-programmable, initiator-indexed table of priorities. Its unique role is to bind a priority to an initiator on the fly. The mapping of each initiator to the table (split into eight registers) is based on its 6-MSB group ConnID (see *L3_MAIN Connectivity Matrix*, in *Interconnect*).

When an interconnect request enters the DMM, its priority is extracted from the PEG LUT.

The 64 priority entries are software-programmable with the DMM_PEG_PRIO_k register. A priority of 0 defines the highest priority and a priority of 7 defines the lowest priority. At reset, all priorities are set to 4.

These registers are each split into eight 4-bit fields, each field mapping an entry of the LUT with:

- The 3-bit priority coded on the 3 least-significant bits (LSBs): P field
- A $\overline{W}$ field-specific active-low local write enable bit, always read as 0, on the MSB. The role of the $\overline{W}$ bit is to allow the modification of a single entry without requiring a read-modify-write sequence. Because its $\overline{W}$ bits are always read as 0, writing back the modified register updates all priority fields of the register.

Table 15-9 lists the initiator ConnIDs that are mapped to PEG priority register fields (P/W).

Table 15-9. ConnIDs vs PEG Priority Register Fields

Registers	P0/W0	P1/W1	P2/W2	P3/W3	P4/W4	P5/W5	P6/W6	P7/W7
DMM_PEG_PRIO_0	0	1	2	3	4	5	6	7
DMM_PEG_PRIO_1	8	9	10(0xA)	11(0xB)	12(0xC)	13(0xD)	14(0xE)	15(0xF)
DMM_PEG_PRIO_2	16(0x10)	17(0x11)	18(0x12)	19(0x13)	20(0x14)	21(0x15)	22(0x16)	23(0x17)
DMM_PEG_PRIO_3	24(0x18)	25(0x19)	26(0x1A)	27(0x1B)	28(0x1C)	29(0x1D)	30(0x1E)	31(0x1F)
DMM_PEG_PRIO_4	32(0x20)	33(0x21)	34(0x22)	35(0x23)	36(0x24)	37(0x25)	38(0x26)	39(0x27)
DMM_PEG_PRIO_5	40(0x28)	41(0x29)	42(0x2A)	43(0x2B)	44(0x2C)	45(0x2D)	46(0x2E)	47(0x2F)
DMM_PEG_PRIO_6	48(0x30)	49(0x31)	50(0x32)	51(0x33)	52(0x34)	53(0x35)	54(0x36)	55(0x37)
DMM_PEG_PRIO_7	56(0x38)	57(0x39)	58(0x3A)	59(0x3B)	60(0x3C)	61(0x3D)	62(0x3E)	63(0x3F)

Although this priority information is generated before entering the LISA block, it is not used internally in the local interconnect arbitration but is forwarded to the EMIF as MReqInfo, where it indicates the command priority.

It is also possible to give a priority for the internal PAT engine through the DMM_PEG_PRIO_PAT register.

15.2.3.5.3.4 LISA Interconnect Arbitration

When Mflag[i][63:0] ≠ 0 is signaled at TILERi or both TILERs, the transactions of TILERi or both TILERs form an additional new group called the high priority group. Within this group, reads and writes are sorted and arbitrated the same way as in the normal priority group. Arbitration between transactions out of the high priority group and out of the normal priority group is based on a weighted round-robin algorithm. Weight is software selectable using the DMM_EMERGENCY[20:16] WEIGHT bit field. Weight selection is considered static (that is, behavior is undefined when software changes the weight, and DMM is not empty of transaction).

The counter that is implemented to support the weighted round-robin is enabled as long as emergency signaling is present. The counter increments by one each time a command (from the normal or the high priority group) is pushed to ROBIN. The counter wraps on DMM_EMERGENCY[20:16] WEIGHT value. One command from the normal priority group is serviced in any of the following scenarios:

- The counter is enabled and wraps.
- The counter is enabled, has not reached its wrap value, and no command is queued in the high priority group (a case where one device module is signaling emergency but its requests have not yet reached the TILERs).
- The counter is disabled.

Setting the DMM_EMERGENCY[0] ENABLE bit to 1 enables the emergency arbitration scheme.

Note

It is recommended to set the DMM_EMERGENCY[0] ENABLE bit to 1, and the DMM_EMERGENCY[20:16] WEIGHT bit to 0x8. This provides better and faster recovery behavior, and benefits system performance.

15.2.3.5.3.5 ROBIN Description

The ROBIN is a block that provides some working buffering for converting data and responses to-and-from between raster and tiled organizations, and a master port to connect to the EMIF.

The ROBIN block:

- Forwards requests
- Writes data and buffers responses

- Keeps write data ordering
- Performs intraword tiling and orientation transforms
- Handles tags

15.2.3.5.3.6 TILER Description

The main function of the TILER is request conversion caused by tiling .

The TILER block:

- Decodes the address to qualify whether the request targets the TILER or the memory directly
- Converts TILER-specific requests to their natural representation; address, width, and height are modified accordingly
- Allocates an internal response context for the timely generation of appropriate responses
- Splits tiled requests at tile boundaries and not-tiled requests at DMM atomic section boundaries
- Requests the page-based address translation
- Requests buffer allocation in the appropriate ROBIN
- In case of a write request, allocates and updates an internal write context to direct the incoming write data into the relevant reordering buffer

The interdependent tiling and isometric transform concepts introduced in the TILER are described in the following section.

15.2.3.6 TILER

15.2.3.6.1 TILER Concepts

This section describes the concepts behind the TILER transforms, through a top-down approach starting from the main object container.

15.2.3.6.1.1 TILER Rationale

This section is a synthesis of all TILER concepts, giving one rule per TILER structure level.

Figure 15-10 shows the TILER address space structure for tiled modes.

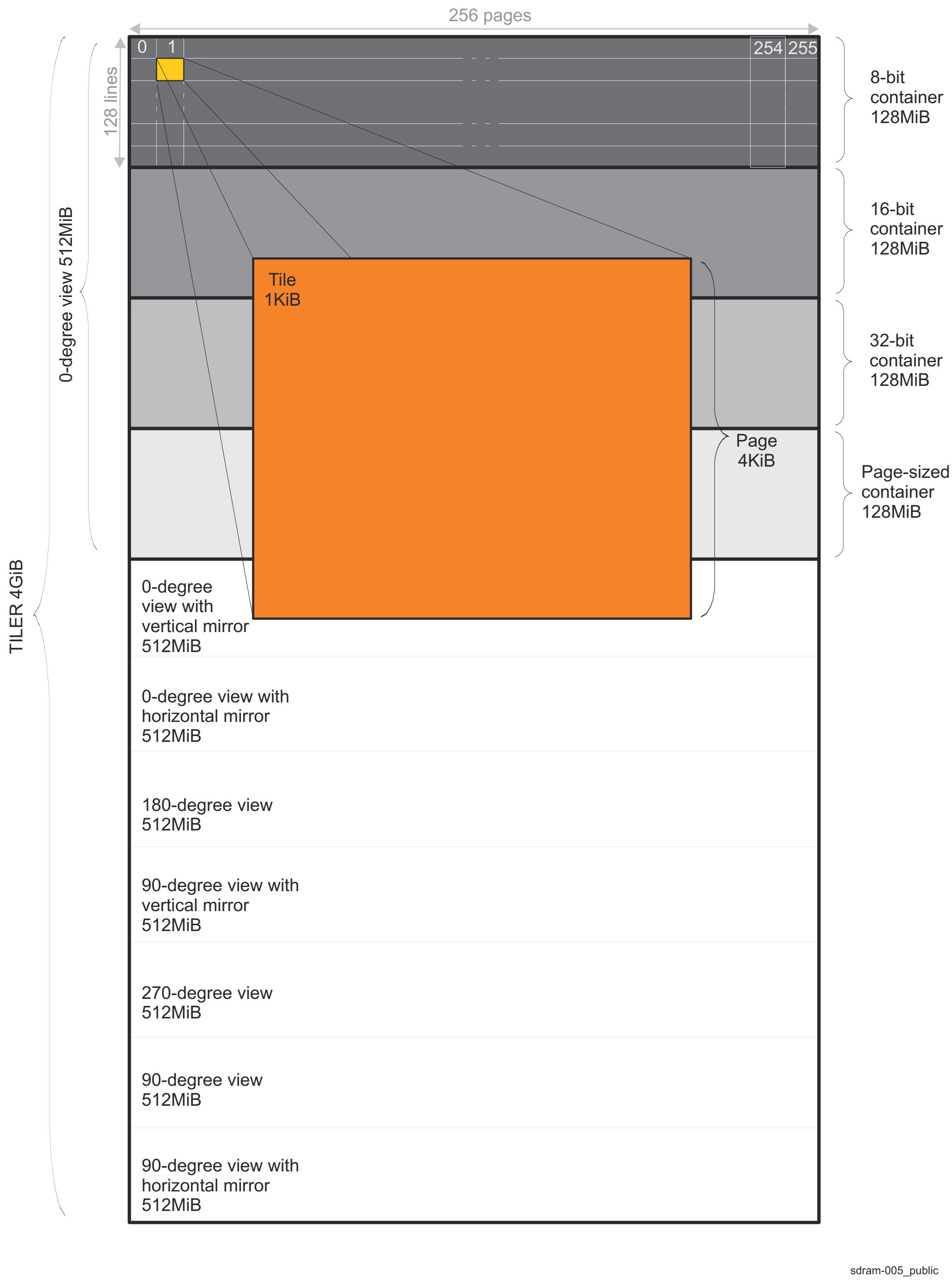


Figure 15-10. TILER Address Space Structure for Tiled Modes

15.2.3.6.1.1.1 The TILER is a 4-GiB Virtual Address Space Composed of Eight Views

There is one view for each of the eight possible ways of scanning a frame-buffer:

- From left to right and then from top to bottom

- From right to left and then from top to bottom
- From left to right and then from bottom to top
- From right to left and then from bottom to top
- From top to bottom and then from left to right
- From top to bottom and then from right to left
- From bottom to top and then from left to right
- From bottom to top and then from right to left

15.2.3.6.1.1.2 A View is a 512-MiB Virtual Address Space Composed of Four Containers

There is one container per element size to allow correct access patterns in any of the eight possible orientations. The container is the entity where all objects of a given element type are allocated.

The element is the entity of maximum size (8, 16, 32 bits, or page-sized), which is invariant in any orientation.

15.2.3.6.1.1.3 A Container is a 128-MiB Virtual Address Space

A container is composed of an array of 128 lines of 256 pages of 4kiB each.

The page defines the granularity of physical memory allocation through a PAT unit – MMU.

15.2.3.6.1.1.4 A Page is a 4-kiB Virtual Address Space

A page is composed of two lines. Each line consists of two tiles.

15.2.3.6.1.1.5 A Tile is a 1-kiB Address Space

The tile is designed to offer bidimensional data locality in a single SDRAM page. In this respect, it is sized to 1kiB; that is, to the size of the smallest SDRAM page.

15.2.3.6.1.1.6

15.2.3.6.1.1.7 A Subtile is a 128-Bit Address Space

A subtile is composed of:

- In 8-bit tiled mode: An array of four horizontal lines of four elements of 8 bits
- In 16-bit tiled mode: An array of two horizontal lines of four elements of 16 bits
- In 32-bit tiled mode: An array of two horizontal lines of two elements of 32 bits

The subtile structure is designed to increase the SDRAM access payload on macroblock requests; that is, to minimize the required line buffering of raster-based initiators to handle on-the-fly rotation, and to improve interlaced accesses to tiled frame-buffers.

15.2.3.6.1.2 TILER Modes

The TILER supports three major modes, bypass, page, and tiled. Each mode has a specific output request generation.

15.2.3.6.1.2.1 Bypass Mode

This mode is transparent from the TILER perspective. However, from the DMM perspective:

- 2D block bursts are broken down on a line basis in a set of incremental bursts.
- Incremental bursts, including those issued by a 2D block burst breakdown, are split at the DMM atomic unit of the section hit by the burst at:
  - The interleaving granularity of the section (128, 256, or 512 bytes) in interleaved sections
  - 1-kiB boundary in noninterleaved sections

15.2.3.6.1.2.2 Page Mode

This mode uses the DMM address translation mechanism for nontiled accesses. In this respect it is similar to bypass mode:

- 2D block bursts are broken down on a line basis in a set of incremental bursts.

- Incremental bursts, including those issued by a 2D block burst breakdown, are split at:
  - The interleaving granularity of the section (128, 256, or 512 bytes) in interleaved sections
  - 1-kiB boundary in noninterleaved sections

15.2.3.6.1.2.3 Tiled Mode

Tiled mode has two major breakdown algorithms:

- One for well-formed 2D block requests that conform to the orientation, mode, and stride listed in Table 15-10
- One for 1D incremental requests and ill-formed 2D block requests

Table 15-10. Well-Formed Tiled Mode 2D Block Requests

Orientation			Mode		Stride	Description
S	$\overline{Y}$	$\overline{X}$	M1	M0	(bytes)	
0	x	x	0	0	16,384	Plain access to an 8-bit progressive frame in 0 or 180 degrees
					32,768	Field access to an 8-bit interlaced frame 0 or 180 degrees
			0	1	32,768	Plain access to a 16-bit progressive frame in 0 or 180 degrees
					65,536	Field access to a 16-bit interlaced frame 0 or 180 degrees
			1	0	32,768	Plain access to a 32-bit progressive frame in 0 or 180 degrees
					65,536	Field access to a 32-bit interlaced frame 0 or 180 degrees
1	x	x	0	0	8192	Plain access to an 8-bit progressive frame in 90 or 270 degrees
					16,384	Field access to an 8-bit interlaced frame 90 or 270 degrees
			0	1	8192	Plain access to a 16-bit progressive frame in 90 or 270 degrees
					16,384	Field access to a 16-bit interlaced frame 90 or 270 degrees
			1	0	16,384	Plain access to a 32-bit progressive frame in 90 or 270 degrees
					32,768	Field access to a 32-bit interlaced frame 90 or 270 degrees

Similar to the bypass and page modes, ill-formed 2D block requests are broken down on a line basis in a set of incremental bursts. In tiled mode, however, these incremental virtual bursts do not translate to 1D physical burst requests.

15.2.3.6.1.3 Object Container Definition

The object container is the unique addressable entry point of the TILER. It is a 128-MiB virtual address space, where all objects of a same kind and orientation are allocated.

Four main types of containers are present in the TILER, each referred by a mode:

- 8-bit element mode, for efficiently accessing bidimensional arrays of 8-bit data
- 16-bit element mode, for efficiently accessing bidimensional arrays of 16-bit data
- 32-bit element mode, for efficiently accessing bidimensional arrays of 32-bit data
- Page mode, for efficient 1D accesses

Figure 15-11 shows the TILER object containers and views.

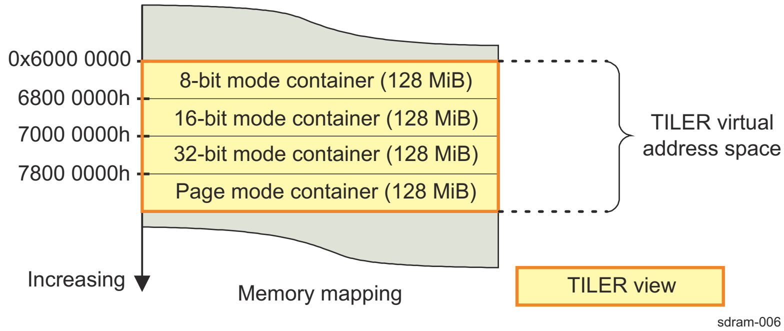


Figure 15-11. TILER Object Containers and Views

The 512-MiB virtual address space composed of four 128-MiB object containers of different modes is called a view. Because eight orientations are available per mode, the TILER actually manages 32 kinds of containers.

The physical memory footprint of a 512-MiB TILER view is directly subject to the nature of the PAT unit.

A unique PAT LUT is instantiated in the DMM. This table is shared by all TILER modes. Hence, each of the four modes cannot be given its own private page-grained PAT LUT. A maximum of 128MiB of objects among all TILER modes can be managed as shown in Figure 15-12.

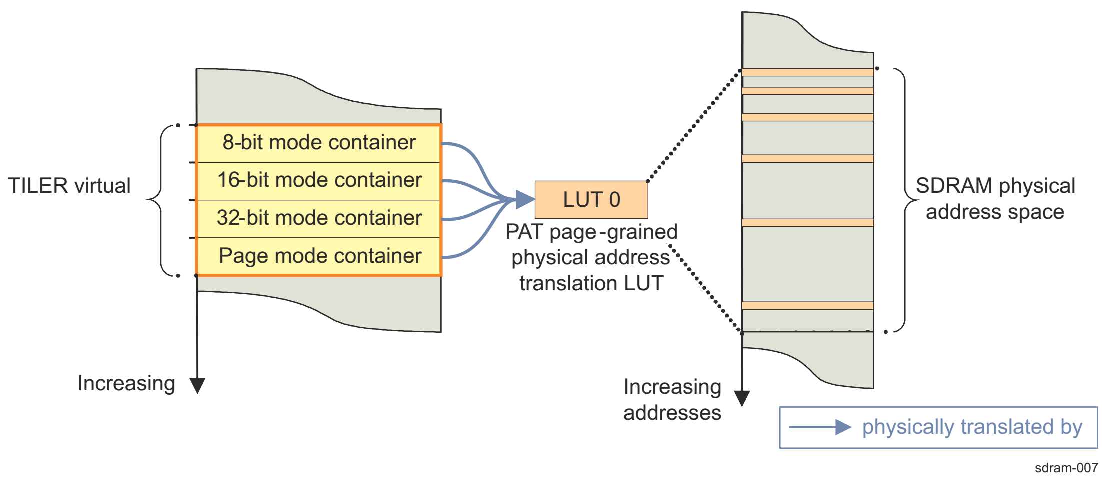


Figure 15-12. TILER Memory Footprint With PAT and Shared Physical Address Translation LUT

Although each of the four modes has its own separate virtual 128-MiB object container, these containers are all mapped to the same piecewise 128-MiB physical address space, and are then not physically separated. Consequently, a memory-related system constraint is that no more than 128-MiB of objects can be available simultaneously in a TILER view.

Note

Software must ensure that any object allocated in a mode does not physically overlap with any other object, even in another mode.

15.2.3.6.1.4 Page Definition

A TILER page defines the granularity of object allocation in virtual TILER containers.

Because the subpage structure is mode-specific, the page is the smallest granularity common to all modes, making it the granularity to consider in the TILER resource manager for object allocation.

15.2.3.6.1.4.1 Container Geometry With 4-kiB Pages

Because the size of the page is 4kiB, any 128-MiB object container is a set of 32,768 pages, organized in an array of 256 columns and 128 rows, as shown in Figure 15-13.

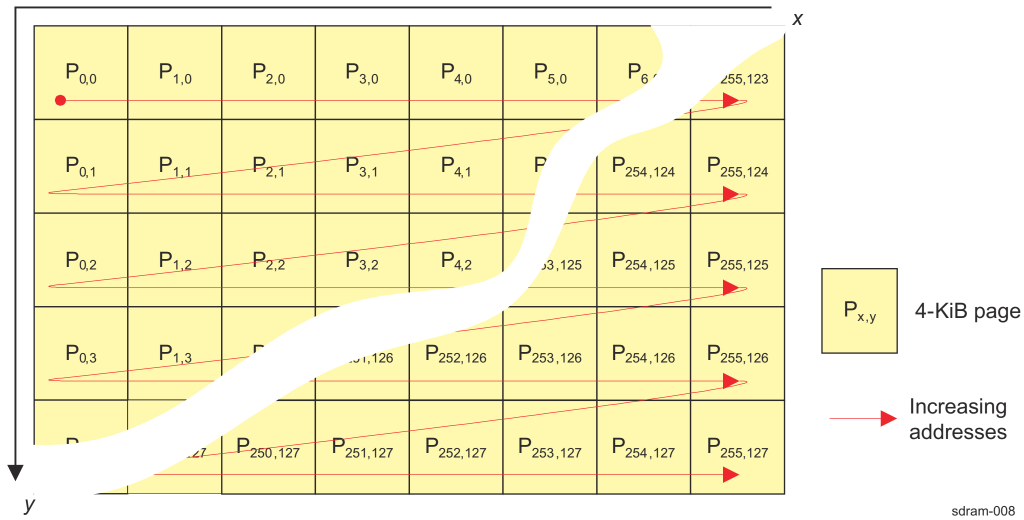


Figure 15-13. Object Container Geometry With 4-kiB Pages

This array of pages is mapped to the system address space, as shown in Figure 15-14.

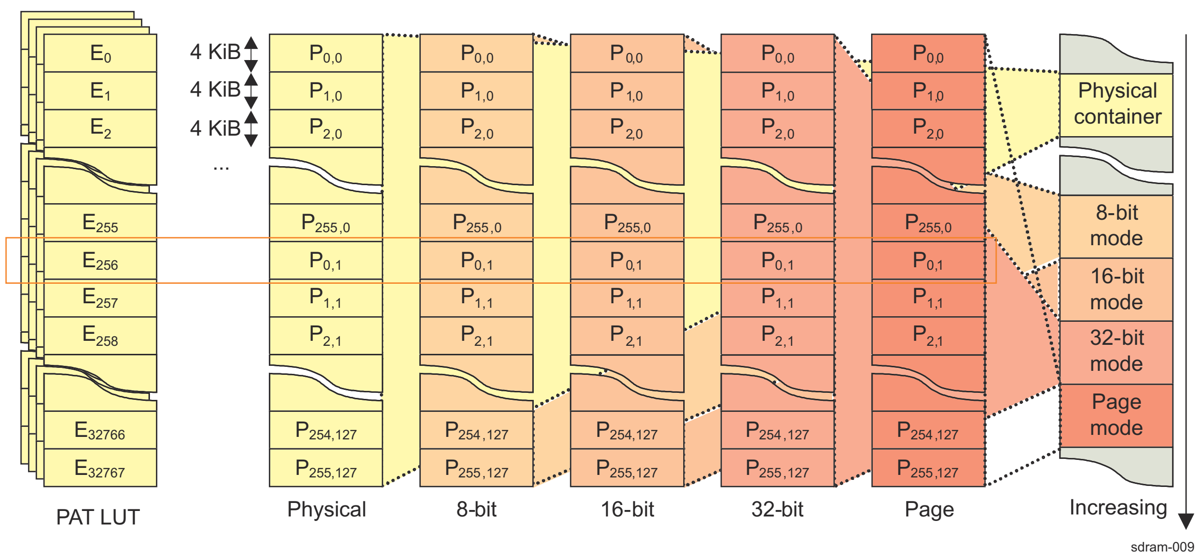


Figure 15-14. TILER Page Mapping When Using 4-kiB Pages

In any 128-MiB object container, the 4-kiB page Px,y at column x (where 0 =< x < 256) and row y (where 0 =< y < 128), is found at an offset of 4096.(x+256.y) bytes from the base address of the related object container.

Similarly, the page Px,y at column x (where 0 =< x < 256) and row y (where 0 =< y < 128), is translated by the LUT entry Ex+256.y found at the index x+256.y.

15.2.3.6.1.4.2 Container Geometry and Page Mapping Summary

The TILER has a page size of 4096 bytes. The page $P_{x,y}$:

- Has max_x = 256 and max_y = 128
- Is found at an offset of 4096.(x + max_x.y) bytes from the base address of the related object container
- Is translated by the entry at the index (x + max_x.y) of the LUT

15.2.3.6.1.5 Orientation

This section describes the eight on-the-fly orientation-related isometric transforms, which correspond to all available changes of orthonormal basis in the bidimensional space of the TILER container.

Figure 15-15 shows isometric transforms in the TILER container

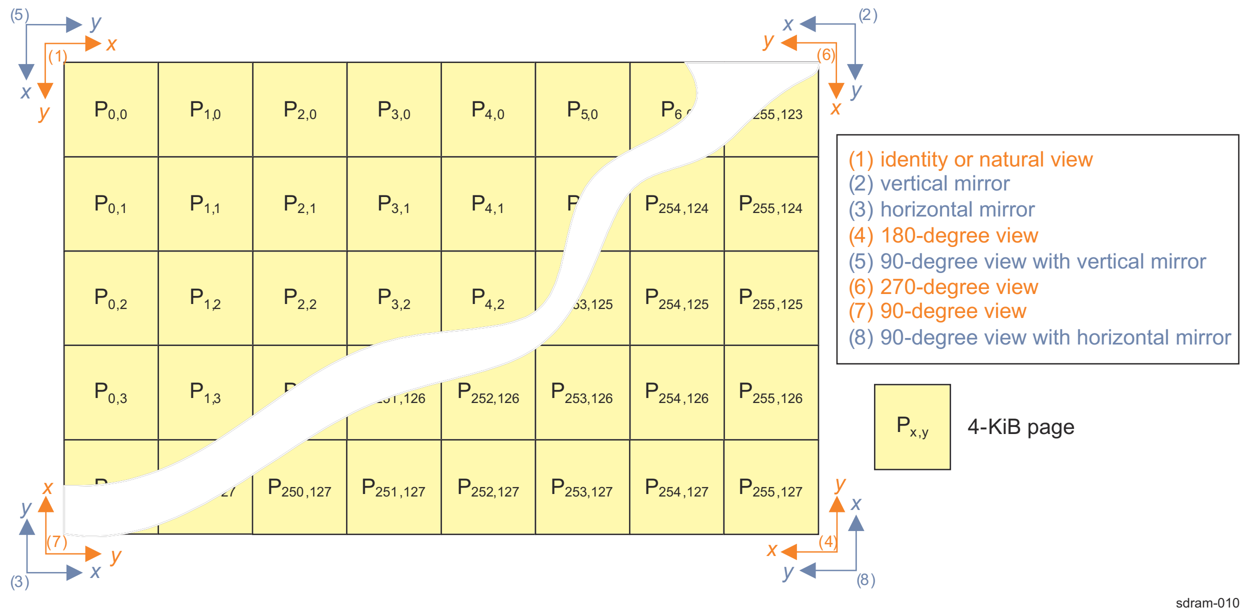


Figure 15-15. Isometric Transforms in the TILER Container

Mathematically speaking, all these transforms correspond to the composition of a 0-, 90-, 180-, or 270-degree rotation with an optional reflection. The nature of this orientation is based on the three following binary parameters:

- *X* to change the direction of the$_x$ axis of the TILER container
- *Y* to change the direction of the$_y$ axis of the TILER container
- *S* to swap the modified$_x$ and$_y$ axis

Hereafter in this document the term orientation refers to any composition of a "quadrant" rotation with an optional horizontal (flip-flop) or vertical mirroring.

15.2.3.6.1.6 Tile Definition

A tile is a subdivision of a page that is aimed at:

- Representing a 2D block to better balance accesses in both directions
- Ensuring that any tiled access that fits within a tile is made atomic in the SDRC and fits in a single SDRAM memory page
- Minimizing the number of SDRAM page openings per 2D block transfer

The tile is defined as a 1-kiB 2D block, and a 4-kiB page as an array of two lines of two tiles each.

When the considered page is in an interleaved DMM section, it is necessary that:

- The DMM memory interleaving size of tiled accesses is set to 1kiB (a tile size) so that any tiled request that fits within a tile, fits in a single SDRAM memory page.
- Any request that spans two or four tiles is distributed on a maximum number of SDRCs.

15.2.3.6.1.7 Subtiles

15.2.3.6.1.7.1 Subtiling Definition

To summarize, a subtile is a subtle refinement of a tile whose purpose is to:

- Lower the size (length × height) of 128-bit bursts in case of relatively small 2D block requests, such as video macroblocks
- Better balance the granularity in the two directions
- Better balance the accesses in the two (x and y) directions
- Minimize the size of the line buffer of the raster-based initiators to handle isometric transforms efficiently

A subtile is defined as a 128-bit 2D block, and a tile as an array of eight lines of eight subtiles, as shown in Figure 15-16.

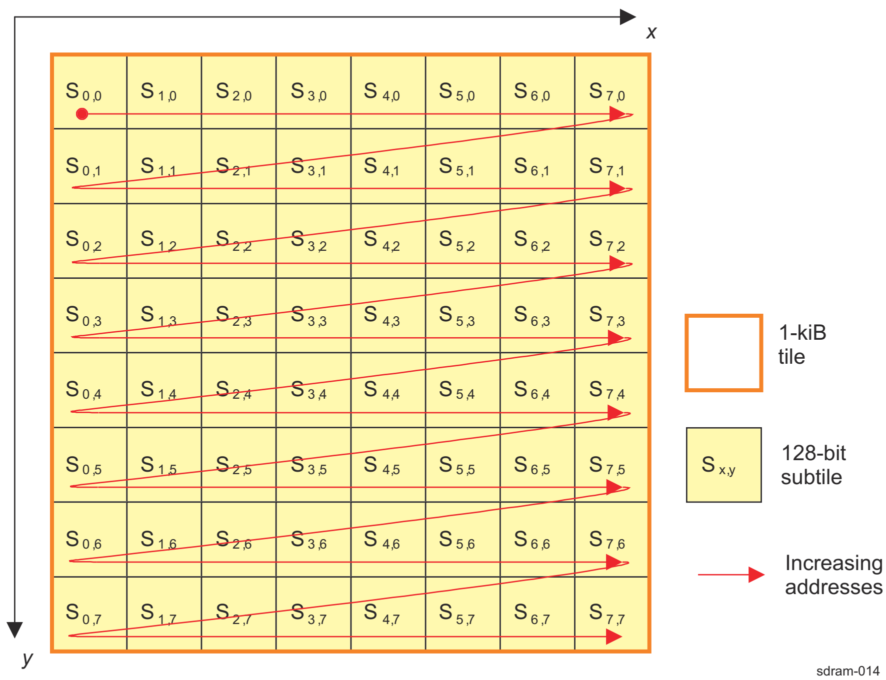


Figure 15-16. Tile Geometry

In any tile, the subtile Sx,y at column x (0 =< x < 8) and row y (0 =< y < 8) is found at an offset of 16.(x + 8.y) bytes from the base address of the related tile. Besides, this array organization of subtiles is common to each tiled mode. For instance, the subtile S0,1 is always located at an offset of 128 bytes (that is, 16.(0 + 8 x 1)) from the base address of the related tile.

Figure 15-17 shows the subtile mapping.

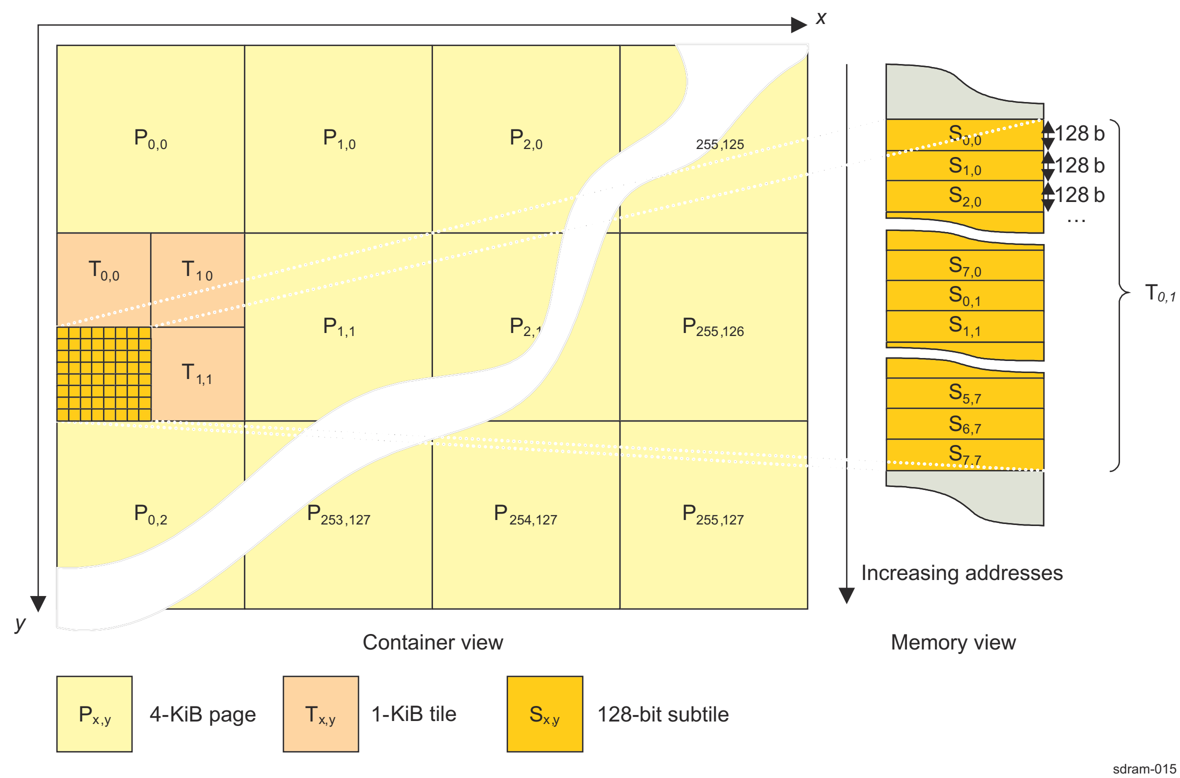


Figure 15-17. Subtile Mapping

15.2.3.6.1.8 TILER Virtual Addressing

The TILER can be virtually accessed in four different modes: 8-, 16-, 32-bit, and page modes. Each mode defines the element granularity to apply isometric transforms, as summarized in Table 15-11.

Table 15-11. Coding and Description of TILER Modes

Mode	Name	Granularity (Element Size)
0	8-bit tiled mode	8 bits
1	16-bit tiled mode	16 bits
2	32-bit tiled mode	32 bits
3	Page mode	4096 bytes

For instance, making a vertical mirror of a 16-byte horizontal line that contains the word 000102030405060708090A0B0C0D0E0Fh, leads to:

- 0F0E0D0C0B0A09080706050403020100h in 8-bit tiled mode
- 0E0F0C0D0A0B08090607040502030001h in 16-bit tiled mode
- 0C0D0E0F08090A0B0405060700010203h in 32-bit tiled mode
- 000102030405060708090A0B0C0D0E0Fh in page mode – unchanged because the element granularity is 4 KiB

Besides, because each of the eight orientations is available for any of the four modes, the TILER has 32 addressing possibilities (see Table 15-12).

Table 15-12. Coding and Description of TILER Orientations

S	$\overline{Y}$	$\overline{X}$	Description	Alternate description
0	0	0	0-degree view	Natural view
0	0	1	0-degree view with vertical mirror	180-degree view with horizontal mirror
0	1	0	0-degree view with horizontal mirror	180-degree view with vertical mirror

Table 15-12. Coding and Description of TILER Orientations (continued)

S	$\overline{Y}$	$\overline{X}$	Description	Alternate description
0	1	1	180-degree view	
1	0	0	90-degree view with vertical mirror	270-degree view with horizontal mirror
1	0	1	270-degree view	
1	1	0	90-degree view	
1	1	1	90-degree view with horizontal mirror	270-degree view with vertical mirror

15.2.3.6.1.8.1 Page Mode Virtual Addressing and Characteristics

When used in page mode, the 128-MiB TILER space is seen as an orientation-specific sequence of 32,768 pages of 4kiB each. The access sequence inside a page is left unchanged.

Therefore, in page mode, the TILER is accessed similarly to any 128-MiB memory, with a 27-bit byte-based address.

Note

From here forward, the address is noted as *a* (see Figure 15-18).

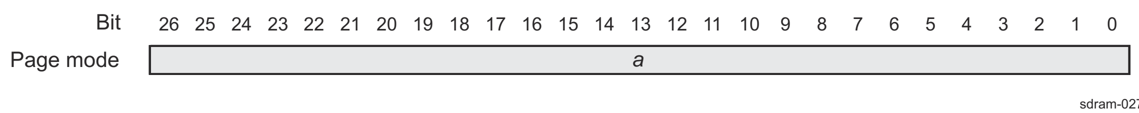


Figure 15-18. Page Mode Virtual Addressing

15.2.3.6.1.8.2 Tiled Mode Virtual Addressing and Characteristics

When used in tiled mode, the 128-MiB TILER space is seen as a giant frame-buffer, the container. The addressing and characteristics of this giant frame-buffer depend on:

- The tiled mode, which defines the considered atomic element size
- The orientation, which potentially swaps its x and y axis, and hence the container geometry

Table 15-13 summarizes the container characteristics in tiled mode.

Table 15-13. Tiled Mode Container Characteristics

Orientation			Element Size (Bits)	Width (Elements)	Height (Elements)	Stride (Bytes)	
S	$\overline{Y}$	$\overline{X}$				Progressive	Interlaced
0	x	x	8	16,384	8192	16,384	32,768
			16	16,384	4096	32,768	65,536
			32	8192	4096	32,768	65,536
1	x	x	8	8192	16,384	8192	16,384
			16	4096	16,384	8192	16,384
			32	4096	8192	16,384	32,768

As a result, the coordinate (x_0, y_0) of a pixel in an oriented view is translated in a virtual address, as shown in Figure 15-19 and Figure 15-20.

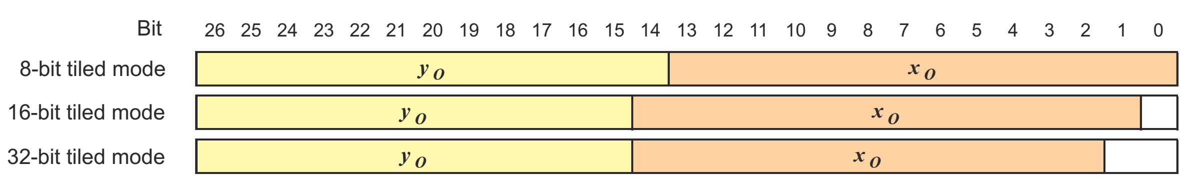


Figure 15-19. Tiled Mode Addressing in 0- or 180-Degree Orientation (S = 0)

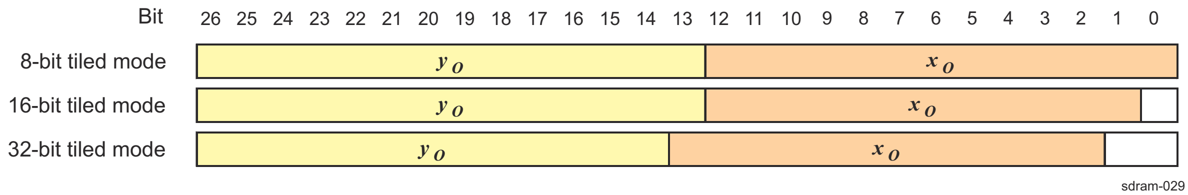


Figure 15-20. Tiled Mode Addressing in 90- or 270-Degree Orientation (S = 1)

15.2.3.6.1.8.3 Element Ordering in the TILER Container

This section describes how elements (8-, 16-, or 32-bit data or a 4-kiB page) are ordered in the container. In other words, this section describes how the path of incrementing virtual addresses is mapped in the container.

Regardless of the mode, and hence the element size, the sequence for ordering the elements in their related container is strictly similar and depends only on the related orientation. In other words:

- Mode is concerned with element granularity.
- Orientation is concerned with change of orthornormal basis for ordering the elements in the mode-specific container.

A corollary to the previous statements is that in a given mode the internal structure of an element is unchanged regardless of the orientation. In page mode for instance, the offset of a word inside a page is invariant by orientation; the content of a page is always accessed in the same manner.

In the following sections, the natural container orthonormal basis is referenced as:

$$(\vec{x}_N, \vec{y}_N)$$

sdram-030

and the oriented orthonormal basis is referenced as:

$$(\vec{x}_O, \vec{y}_O)$$

sdram-031

15.2.3.6.1.8.3.1 Natural View or 0-Degree View (Orientation 0)

This orientation defined by S = 0, $\overline{Y}$ = 0, and $\overline{X}$ = 0 means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = \vec{x}_N \\ \vec{y}_O = \vec{y}_N \end{cases}$$

sdram-032

In any TILER mode, the elements are ordered from left to right and then from top to bottom in their container, as shown in Figure 15-21 and Figure 15-22.

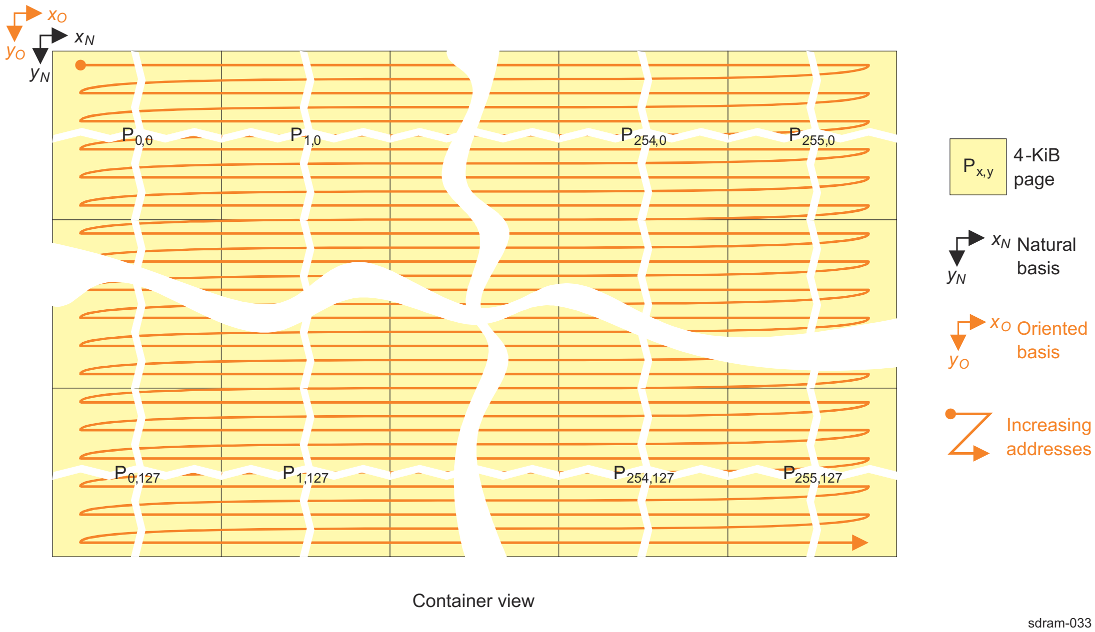


Figure 15-21. Tiled Mode Ordering of Elements in Natural View

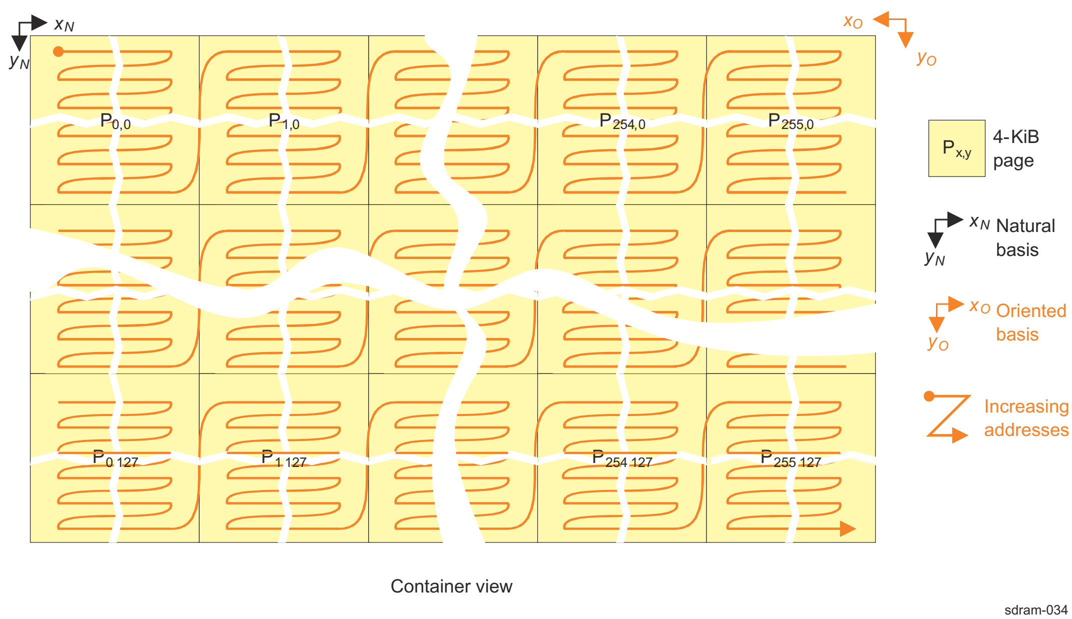


Figure 15-22. Page Mode Ordering of Elements in Natural View

15.2.3.6.1.8.3.2 0-Degree View With Vertical Mirror or 180-Degree View With Horizontal Mirror (Orientation 1)

This orientation defined by S = 0, $\overline{Y}$ = 0, and $\overline{X}$ = 1 and means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = -\vec{x}_N \\ \vec{y}_O = \vec{y}_N \end{cases}$$

sdram-035

In any TILER mode, the elements are then ordered from right to left and then from top to bottom in their container, as shown in Figure 15-23 and Figure 15-24.

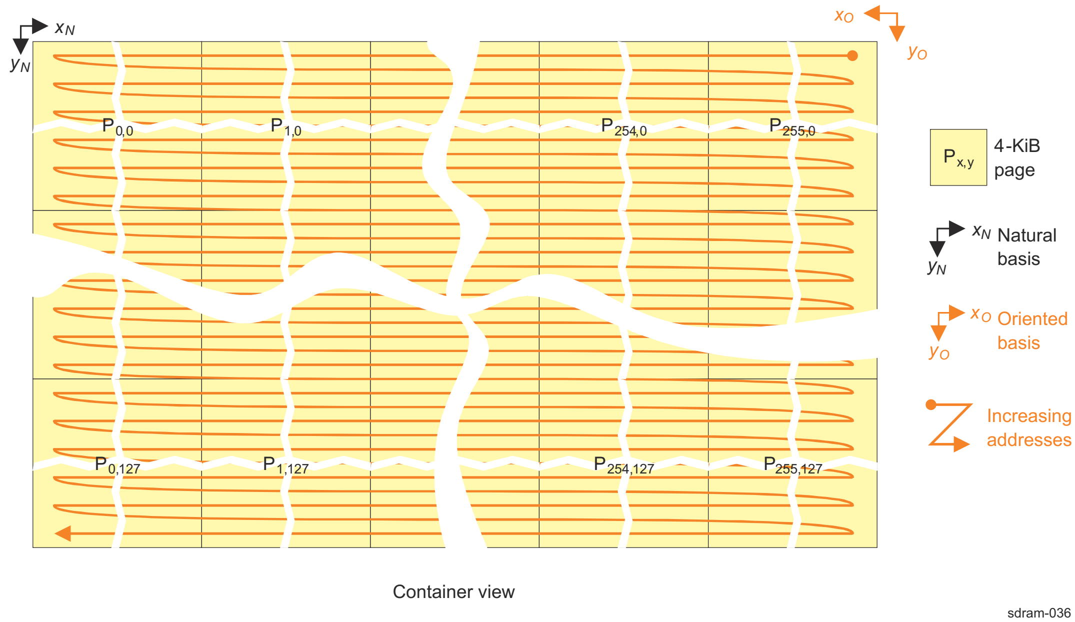


Figure 15-23. Tiled Mode Ordering of Elements in 0-Degree View With Vertical Mirror

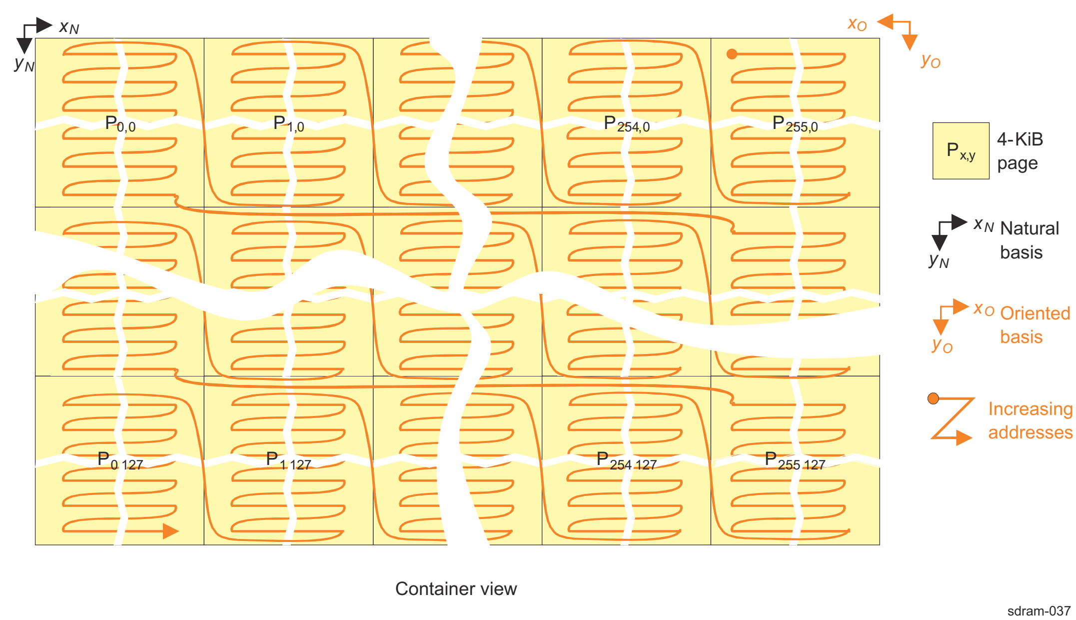


Figure 15-24. Page Mode Ordering of Elements in 0-Degree View With Vertical Mirror

15.2.3.6.1.8.3.3 0-Degree View With Horizontal Mirror or 180-Degree View With Vertical Mirror (Orientation 2)

This orientation defined by S = 0, $\overline{Y}$ = 1, and $\overline{X}$ = 0 and means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = \vec{x}_N \\ \vec{y}_O = -\vec{y}_N \end{cases}$$

sdram-038

In any TILER mode, the elements are ordered from left to right and then from bottom to top in their container, as shown in Figure 15-25 and Figure 15-26.

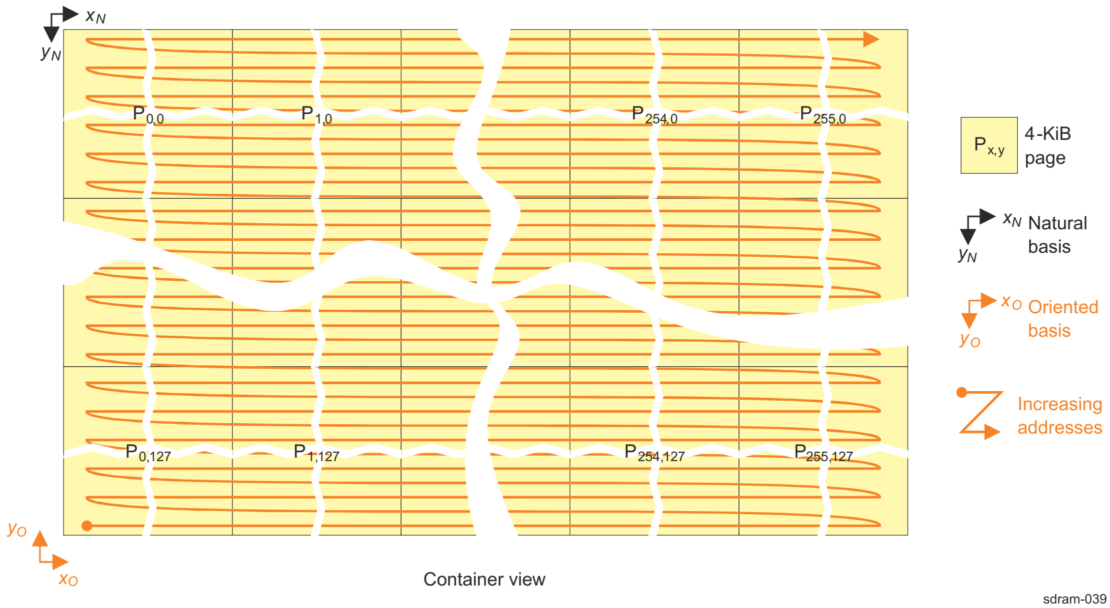


Figure 15-25. Tiled Mode Ordering of Elements in 0-Degree View With Horizontal Mirror

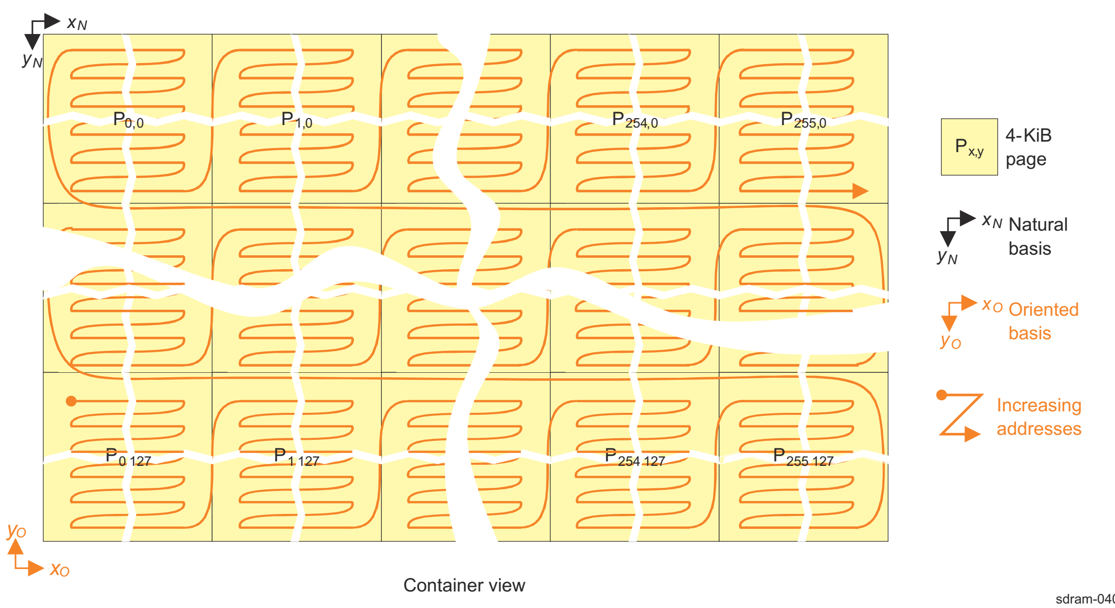


Figure 15-26. Page Mode Ordering of Elements in 0-Degree View With Horizontal Mirror

15.2.3.6.1.8.3.4 180-Degree View (Orientation 3)

This orientation defined by S = 0, $\overline{Y}$ = 1, and $\overline{X}$ = 1 and means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = -\vec{x}_N \\ \vec{y}_O = -\vec{y}_N \end{cases}$$

sdram-041

In any TILER mode, the elements are ordered from right to left and then from bottom to top in their container, as shown in Figure 15-27 and Figure 15-28.

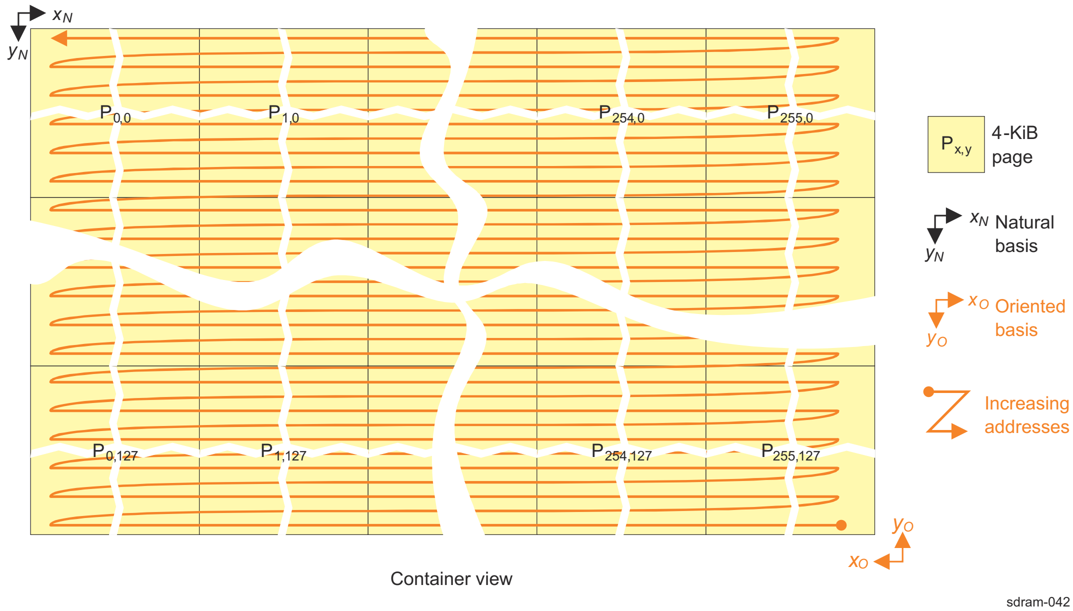


Figure 15-27. Tiled Mode Ordering of Elements in 180-Degree View

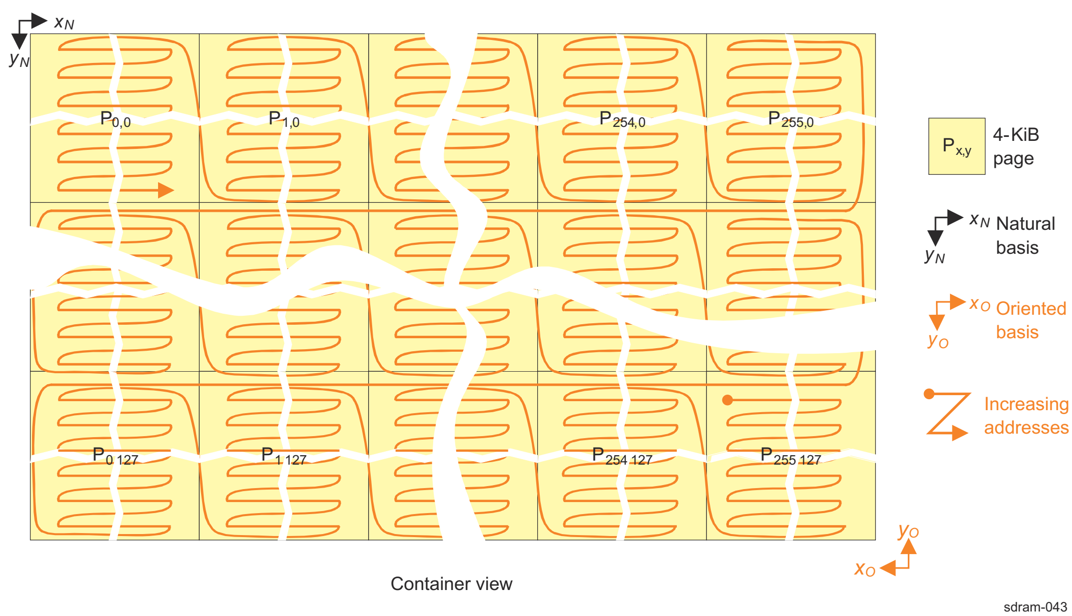


Figure 15-28. Page Mode Ordering of Elements in 180-Degree View

15.2.3.6.1.8.3.5 90-Degree View With Vertical Mirror or 270-Degree View With Horizontal Mirror (Orientation 4)

This orientation defined by S = 1, $\overline{Y}$ = 0, and $\overline{X}$ = 0 and means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = \vec{y}_N \\ \vec{y}_O = \vec{x}_N \end{cases}$$

sdram-044

In any TILER mode, the elements are ordered from top to bottom and then from left to right in their container, as shown in Figure 15-29 and Figure 15-30.

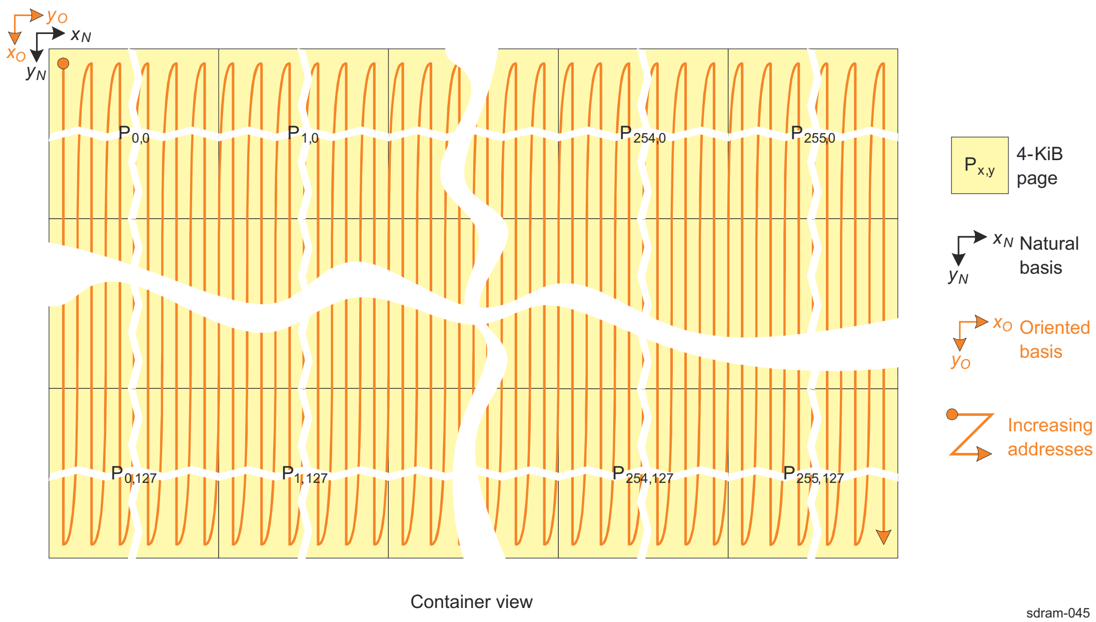


Figure 15-29. Tiled Mode Ordering of Elements in 90-Degree View With Vertical Mirror

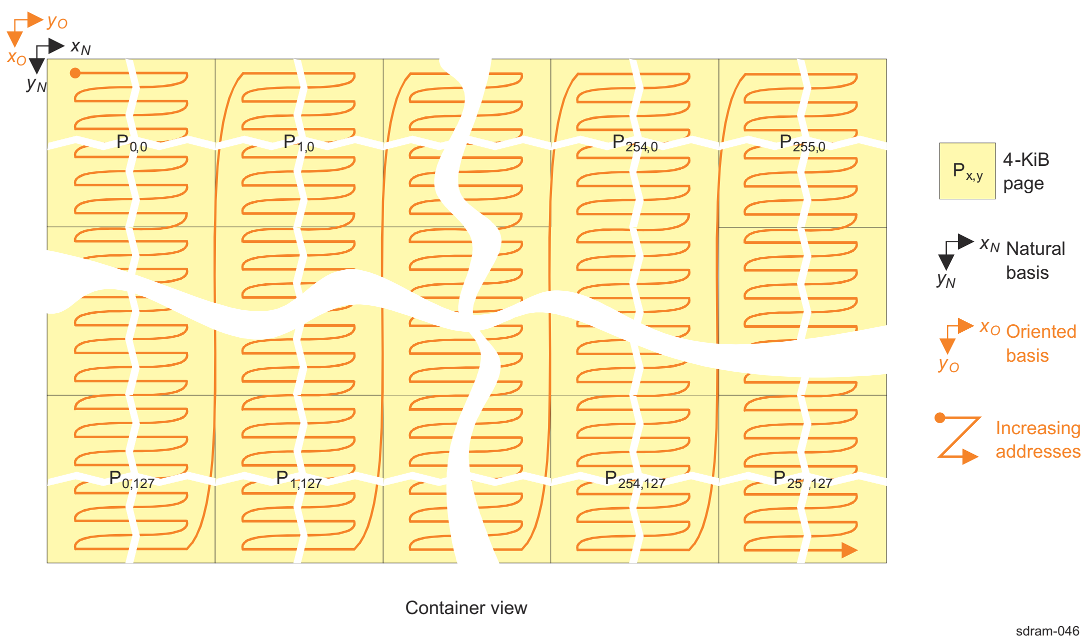


Figure 15-30. Page Mode Ordering of Elements in 90-Degree View With Vertical Mirror

15.2.3.6.1.8.3.6 270-Degree View (Orientation 5)

This orientation defined by S = 1, $\overline{Y}$ = 0, and $\overline{X}$ = 1 and means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = \vec{y}_N \\ \vec{y}_O = -\vec{x}_N \end{cases}$$

sdram-047

In any TILER mode, the elements are ordered from top to bottom and then from right to left in their container, as shown in Figure 15-31 and Figure 15-32.

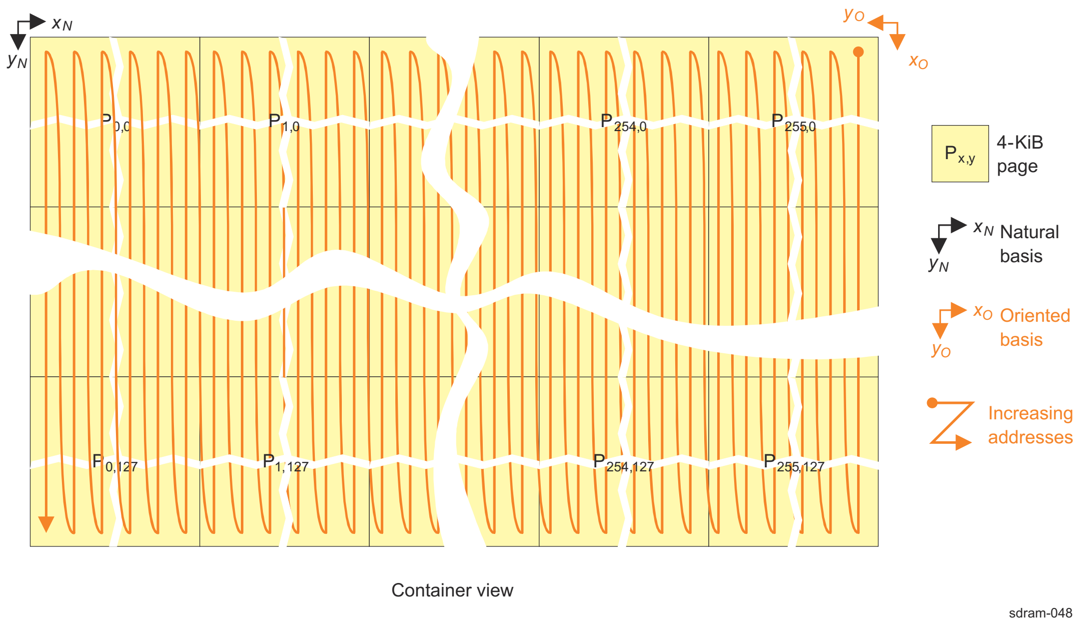


Figure 15-31. Tiled Mode Ordering of Elements in 270-Degree View

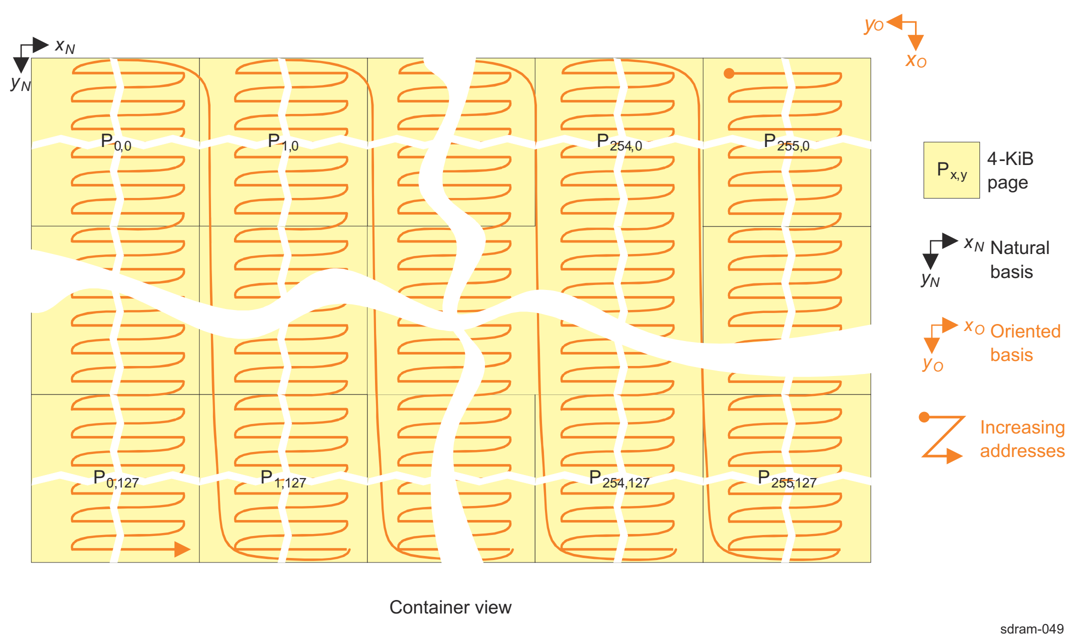


Figure 15-32. Page Mode Ordering of Elements in 270-Degree View

15.2.3.6.1.8.3.7 90-Degree View (Orientation 6)

This orientation defined by S = 1, $\overline{Y}$ = 1, and $\overline{X}$ = 0 and means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = -\vec{y}_N \\ \vec{y}_O = \vec{x}_N \end{cases}$$

sdram-050

In any TILER mode, the elements are ordered from bottom to top and then from left to right in their container, as shown in Figure 15-33 and Figure 15-34.

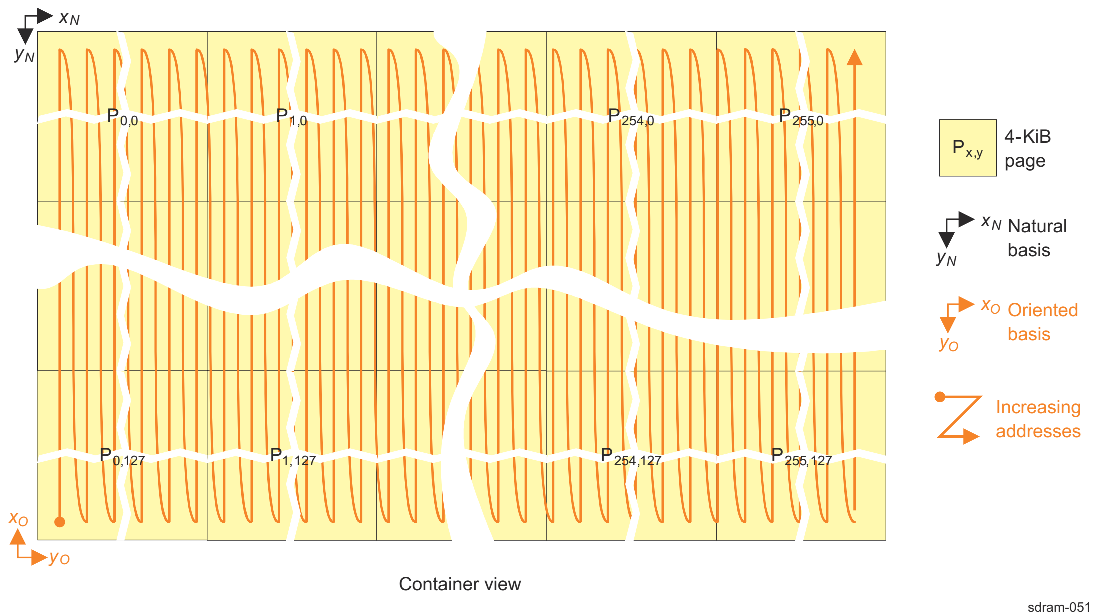


Figure 15-33. Tiled Mode Ordering of Elements in 90-Degree View

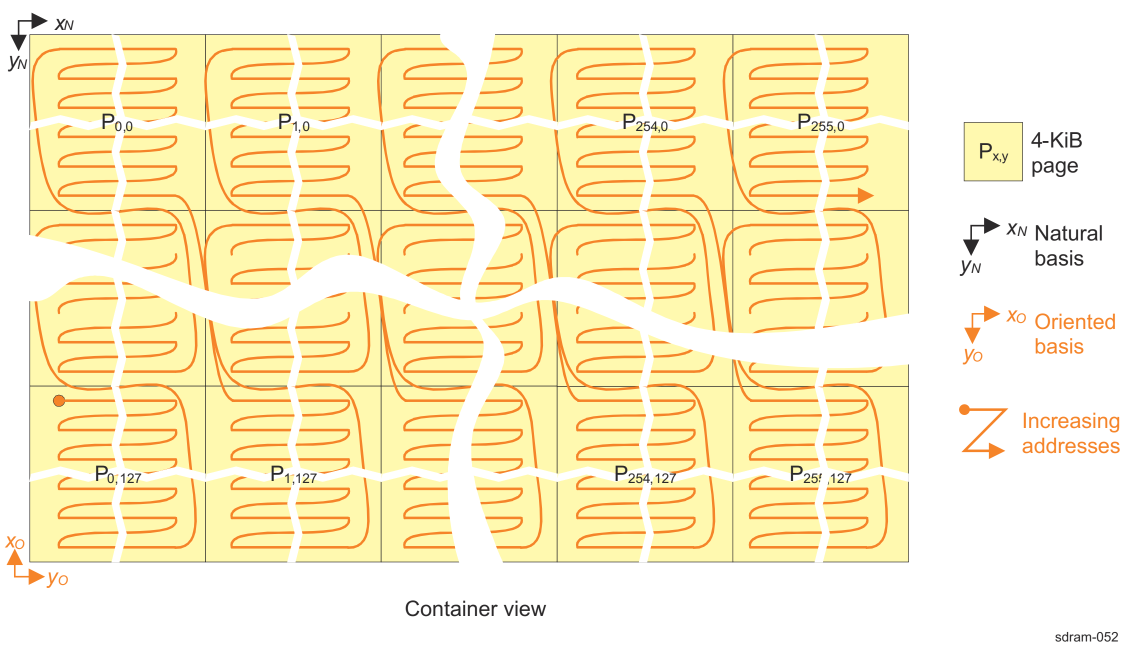


Figure 15-34. Page Mode Ordering of Elements in 90-Degree View

15.2.3.6.1.8.3.8 90-Degree View With Horizontal Mirror or 270-Degree View With Vertical Mirror (Orientation 7)

This orientation defined by S = 1, $\overline{Y}$ = 1, and $\overline{X}$ = 1 and means that the operated change of basis is:

$$\begin{cases} \vec{x}_O = -\vec{y}_N \\ \vec{y}_O = -\vec{x}_N \end{cases}$$

sdram-053

In any TILER mode, the elements are ordered from bottom to top and then from right to left in their container, as shown in Figure 15-35 and Figure 15-36.

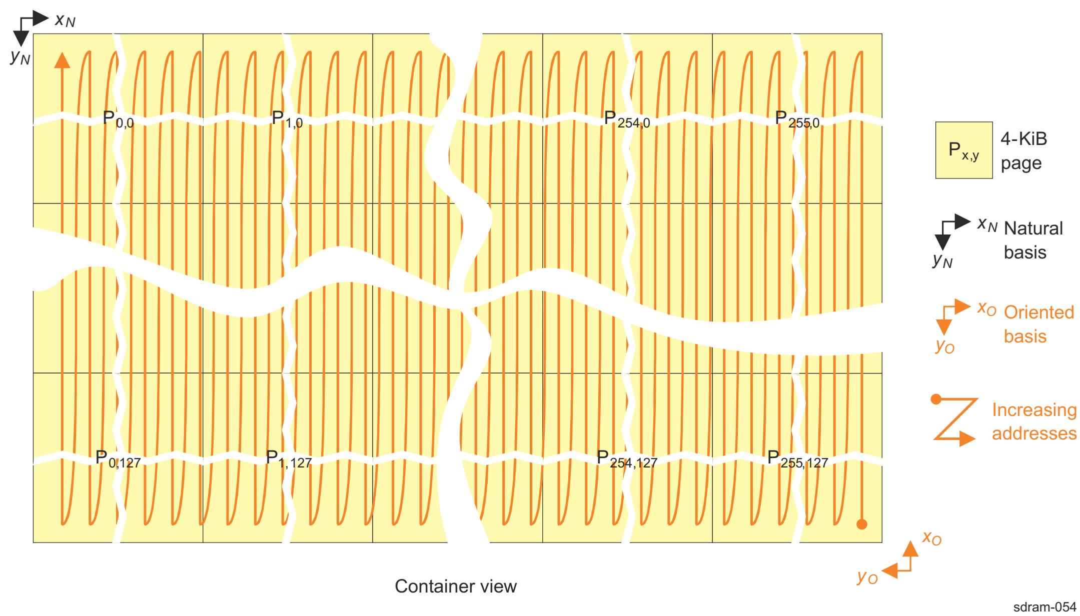


Figure 15-35. Tiled Mode Ordering of Elements in 90-Degree View With Horizontal Mirror

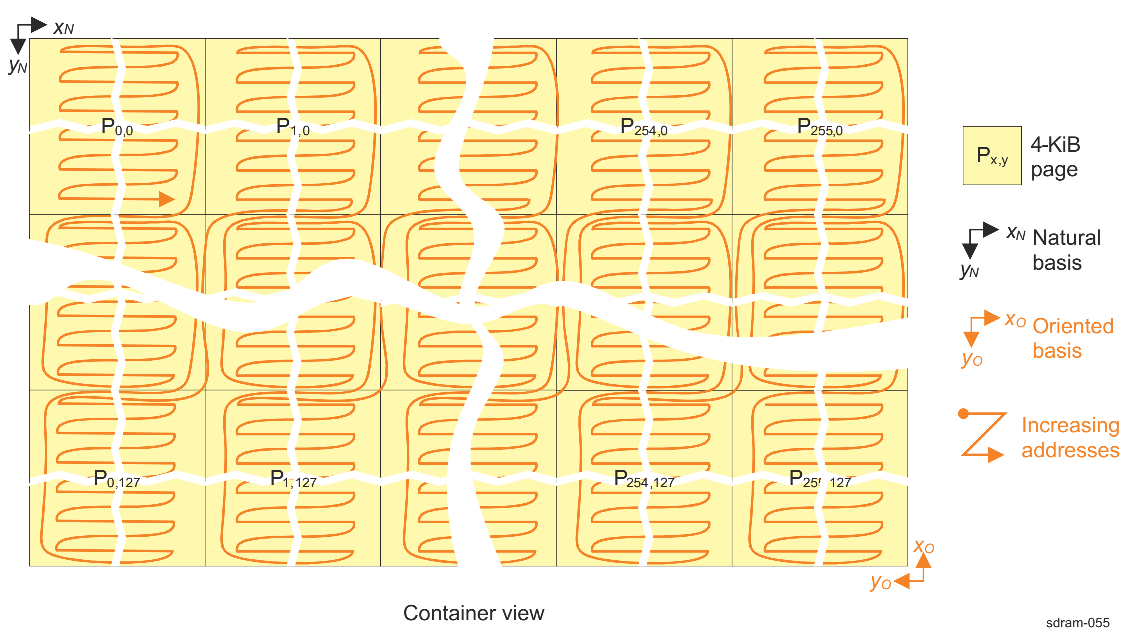


Figure 15-36. Page Mode Ordering of Elements in 90-Degree View With Horizontal Mirror

15.2.3.6.2 TILER Macro-Architecture

The TILER requires a 4-GiB addressing space to map its 128-MiB physical container in four modes and eight orientations. Because its addressing space alone fills the 4-GiB global system address map, the TILER addressing space cannot enter as-is into the system address map. Besides, putting in place a register-based mechanism per initiator to specify the orientation of the following accesses and then reducing the TILER addressing space to a single 512-MiB view is not an option; this is because most of the bandwidth-hungry initiators require simultaneous accesses to the TILER container in different views.

As a result, 32 bits are not enough to address all these requests, because the TILER port must convey not only virtually addressed requests to the "oriented" TILER containers but also physically addressed requests to the attached SDRCs. A thirty-third address bit is necessary to distinguish the two separated address maps, as shown in Figure 15-37.

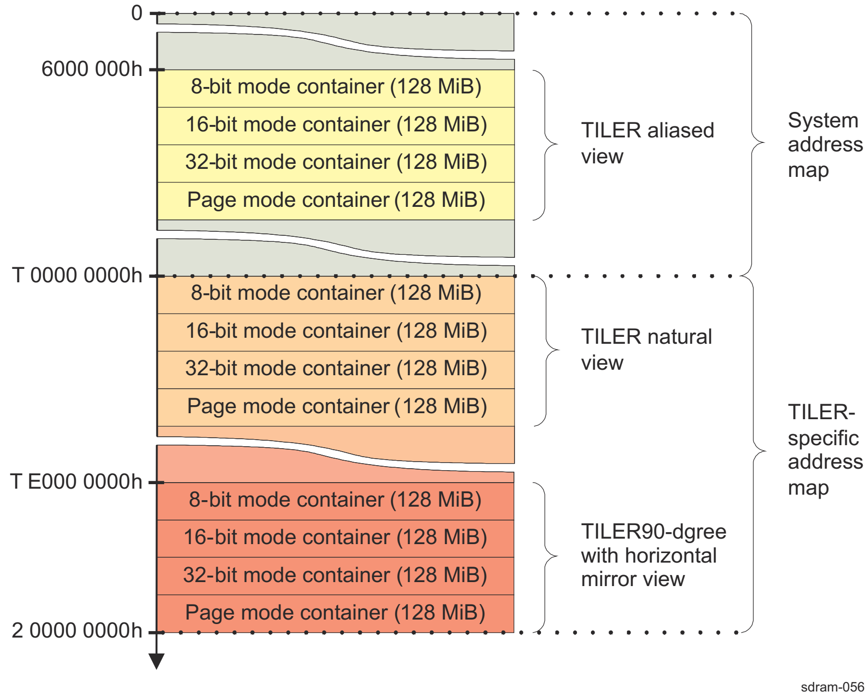


Figure 15-37. TILER Port Address Map

Still, having separated systems and TILER-specific address maps is not sufficient. In a system many existing and external IP blocks still rely on a 32-bit address and would then be limited to one or the other 4-GiB addressing space. To overcome this limitation, one 512-MiB view is aliased in the system address map as (see Table 15-14 and Figure 15-38).

Table 15-14. TILER Aliased View in the L3 Interconnect Mapping

Start Address (hex)	End Address (hex)	Size
0x6000_0000	0x7FFF_FFFF	512 MiB

From all incoming requests, TILER requests are filtered as having an address that fits one of the following:

- The address format in the TILER-specific address map given in Table 15-15
- The address format of the aliased view in the system address map given in Table 15-16

Other requests are forwarded directly to the SDRC in TILER bypass mode.

Table 15-15. Address Format in the TILER-Specific Address Map

32	31	30	29	28	27	26 ... 4	3 ... 0
T	Orientation			Mode		Virtual address	
1	S	$\bar{Y}$	$\bar{X}$	M1	M0	A26 ... A4	0

Table 15-16. Address Format of the TILER Aliased View in the System Address Map

32	31	30	29	28	27	26 ... 4	3 ... 0
T	TILER aliasing			Mode		Virtual address	
0	0	1	1	M1	M0	A26 ... A4	0

In these address formats:

- The thirty-third bit, noted T, is aimed at distinguishing the standard 4-GiB system address map from the 4-GiB TILER-specific address map.
- The orientation bits, noted S, $\overline{Y}$, and $\overline{X}$, define the request orientation, as specified in Table 15-12.
- The mode bits, noted M1 and M0, define the request mode, as specified in Table 15-11.
- The remaining 27 bits, noted A0 to A26, define the mode and orientation specific virtual address, as defined in Figure 15-18, Figure 15-19, and Figure 15-20.

The orientation of the aliased TILER view is extracted from an initiator-indexed LUT, as shown in Figure 15-38.

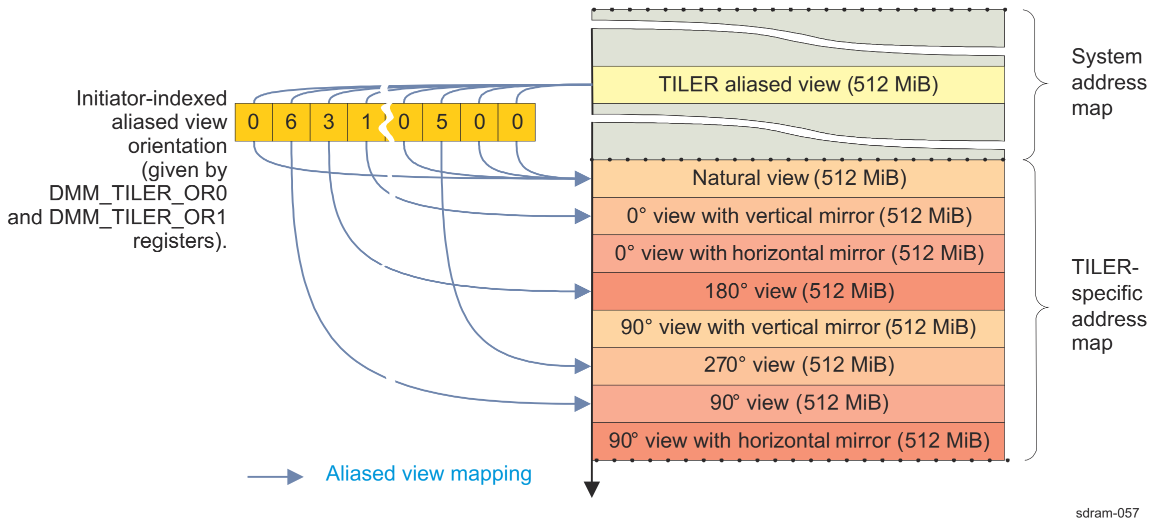


Figure 15-38. TILER Aliased View Orientation

As shown in Figure 15-38, an internal initiator-indexed LUT stores the current orientation of the aliased view for each initiator.

When an interconnect request hits the TILER aliased view in the system address map, the request initiator orientation is extracted from the LUT and the request address is translated according to this orientation: the T bit is set and bits [31:29] are replaced with the orientation.

Regarding the dimensioning of this LUT, given that initiators simultaneously accessing multiple views use the TILER-specific address map, and many initiators do not need to access any view or can be restricted to accessing only the natural view, only a limited number of initiators are likely to dynamically modify the orientation of its aliased TILER view.

The orientation LUT is limited to 16 entries. These 16 orientation entries are mapped on the two 32-bit DMM_TILER_OR0 and DMM_TILER_OR1 registers.

The first eight entries of the LUT are mapped in the DMM_TILER_OR0 register, and the last eight entries are mapped in the DMM_TILER_OR1 register. Therefore, given an index x, the orientation related to this index is given in the ORx field.

Each of these registers is split into eight 4-bit fields, each field mapping an entry of the LUT with:

- An S, $\overline{Y}$, $\overline{X}$ orientation code on the 3 LSBs
- An $\overline{W}$ field-specific active-low local write-enable bit, always read as 0, on the MSB.

The registers fields that correspond to initiators that do not need any dynamic configuration of their aliased view orientation must be specified as reserved fields, and only written with zeros.

The $\overline{W}$ bit allows the modification of a single entry without requiring a read-modify-write sequence. This approach is then more accommodating:

- In a system where multiple initiators can modify their own fields in the registers
- With initiators unable to make the read-modify-write sequence, such as DMA

Still, the $\overline{W}$ bit is active-low to keep the compatibility with the usual read-modify-write sequence. When reading an aliased view orientation register, because all its $\overline{W}$ bits are read as 0, if these bits are untouched by the modification—as they should be—writing back the modified register updates all orientation fields of the register.

15.2.3.6.3 TILER Guidelines for Initiators

15.2.3.6.3.1 Buffered Raster-Based Initiators

15.2.3.6.3.1.1 Buffer Size

The necessary minimum buffer size depends on the SDRAM prefetch size and on initiator support in terms of the following:

- Element size
- Orientation
- Maximum number of elements per line in all supported element sizes and orientations

Let N be the maximum number of elements per line for a given mode (element size) and orientation, P be the SDRAM memory prefetch size in bytes, and *max* be the function returning the maximum of the two parameters. The minimum necessary line buffer size to handle all resolutions in a given mode and orientation is listed in Table 15-17.

Table 15-17. Minimum Buffer Size to Efficiently Handle Lines of up to N Elements

	0- or 180- Degree Orientation (S = 0)	90- or 270-Degree Orientation (S = 1)
8-bit mode	4 × N bytes	*max* (4,P/4) × N bytes
16-bit mode	None	*max* (8,P/2) × N bytes
32-bit mode	None	*max* (8,P/2) × N bytes

This minimal buffer size can be reached only with an advanced FIFO management scheme. The standard ping-pong buffer requires twice this buffer size.

Note

Given their nature, field accesses to an interlaced frame-buffer require buffers twice as small as in the standard progressive case.

For instance, an initiator that must handle only a single progressive frame-buffer in any orientation of:

- Up to 1920 × 1080 in YUV4:2:0
- Up to 1600 × 1200 in 16-bit RGB565
- Up to 800 × 600 in 32-bit ARGB

requires a line buffer of at least 15,360 bytes because:

- 4 × 1920 + 4 × 2960 = 15,360 bytes required for a 3-plane YUV4:2:0 frame of 1920 × 1080
- 4 × 1920 + 8 × 960 = 15,360 bytes required for a 2-plane YUV4:2:0 frame of 1920 × 1080
- 8 × 1600 = 12,800 bytes required for a 16-bit RGB565 frame of 1600 × 1200
- 8 × 800 = 6400 bytes required for a 32-bit ARGB frame of 800 × 600

and a line buffer of at least 30,720 bytes when using a simple ping-pong buffer scheme.

15.2.3.6.3.1.2 Performance

Table 15-18 lists the ratio of effective data-to-transferred data at the external memory interface for the buffered raster-based initiators in each mode and orientation when using a 32-bit DDR2 or DDR3 memory.

Table 15-18. Memory Data Payload for Buffered Raster-Based Initiators on 32-Bit DDR2 or DDR3

	0- or 180-Degree Orientation (S = 0)		90- or 270-Degree Orientation (S = 1)	
	Progressive	**Interlaced**	**Progressive**	**Interlaced**
8-bit mode	100%	50%	100%	50%
16-bit mode	100%	50%	100%	50%
32-bit mode	100%	50%	100%	50%
Page mode	100%	N/A	100%	100%

When using 32-bit DDR2 or DDR3 SDRAMs, whenever all previous guidelines are fulfilled, there is no penalty for these initiators to access a tiled object in any orientation, except for accesses to an interlaced frame in any orientation where the memory bandwidth is twice the requested bandwidth.

When using DDR2 or DDR3 SDRAMs, all accesses, except interlaced accesses, are equally efficient and offer the full possible memory bandwidth.

Similarly, page mode does not introduce any performance hit.

15.2.4 DMM Use Cases and Tips

15.2.4.1 PAT Use Cases

Five ways to use PAT are:

- Simple manual area refill: Section 15.2.4.1.1
- Single auto-configured area refill: Section 15.2.4.1.2
- Chained auto-configured area refill: Section 15.2.4.1.3
- Synchronized auto-configured area refill: Section 15.2.4.1.4
- Cyclic synchronized auto-configured area refill: Section 15.2.4.1.5

Note

PAT refill area must be fully contained within either the LUT lower half or the LUT upper half. It must not span over both areas. That is, y0 MS

bit y0[7] must be equal to y1 MSbit y1[7] when defining the area descriptor.

15.2.4.1.1 Simple Manual Area Refill

The following must be performed to create a 16-byte aligned memory-mapped entry table containing all entries of the defined area (see Figure 15-39):

1. Write the DMM_PAT_AREA_i register with the relevant (x0, y0) (x1, y1) area definition.
2. Write the DMM_PAT_DATA_i register with the physical address of the created entry table.
3. Write the DMM_PAT_CTRL_i register with the requested refill direction and assert the DMM_PAT_CTRL_i[0] START bit with the requested refill direction D and with the LUT ID L if multiple LUTs are present in the system.
4. The refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
5. A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

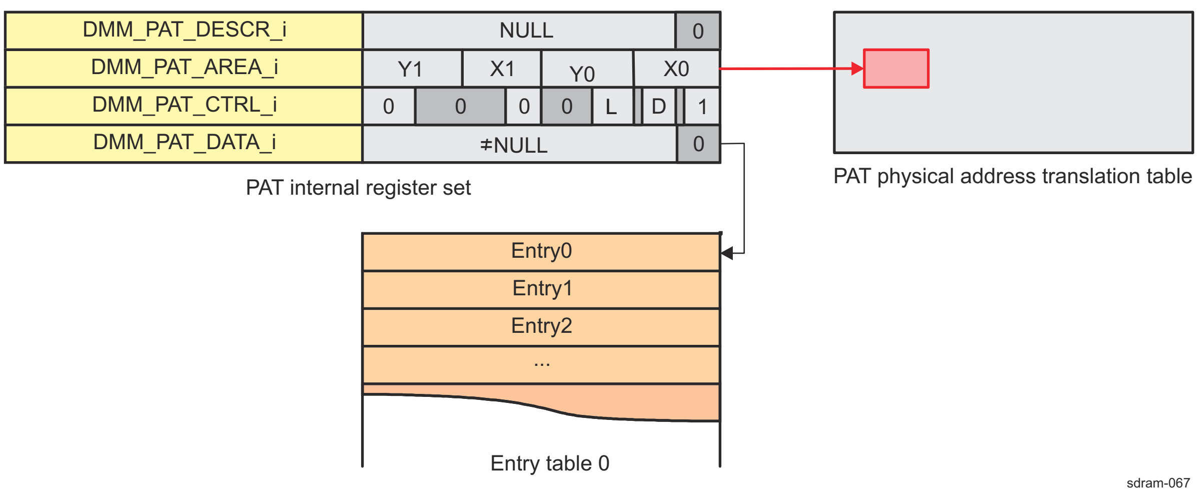


Figure 15-39. Simple Manual Area Refill Scheme

15.2.4.1.2 Single Auto-Configured Area Refill

The following must be performed to create a 16-byte aligned memory-mapped entry table containing all entries of the defined area (see Figure 15-40):

1. Create a 16-byte aligned memory-mapped descriptor structure where:
   - The next field is set to NULL.
   - The area field is set with the relevant (x0, y0) (x1, y1) area definition.

- The ctrl field is set with the requested direction D, with the requested LUT ID L if multiple LUTs are present in the system, and the START bit is asserted to start refilling as soon as this descriptor enters the PAT refill engine.
- The data field is set to the physical address of the created entry table.

2. Write the DMM_PAT_DESCR_i register with the physical address of the created descriptor.
3. The refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

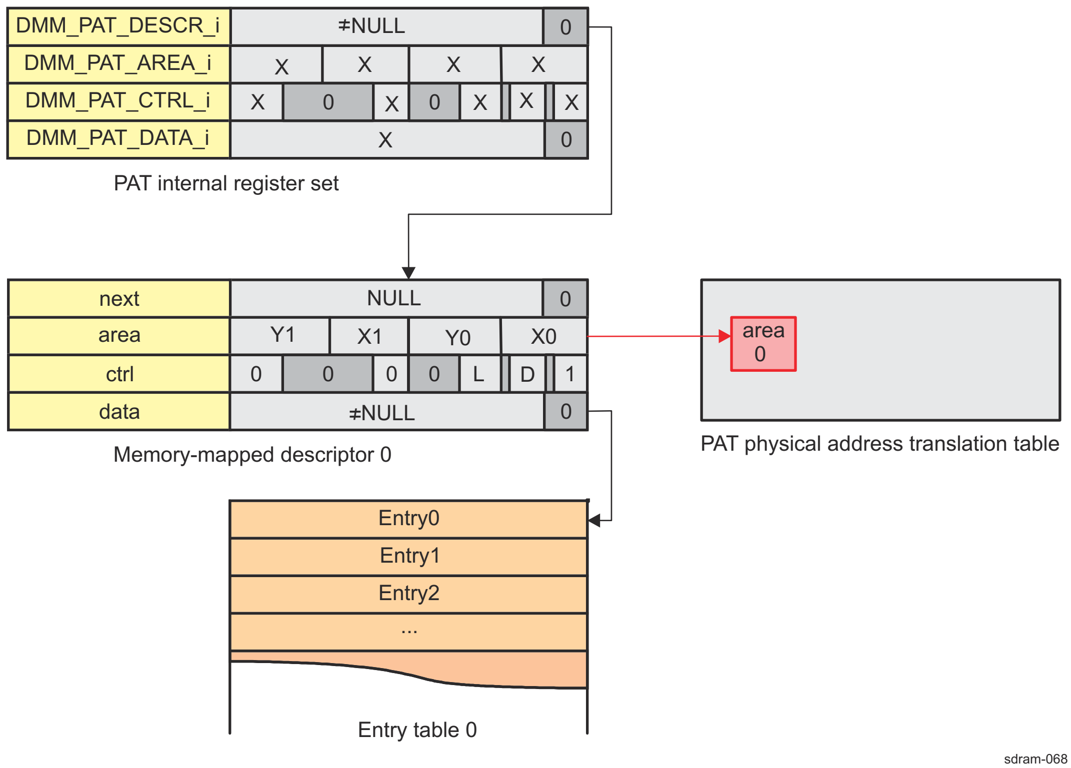


Figure 15-40. Single Auto-Configured Area Refill Scheme

15.2.4.1.3 Chained Auto-Configured Area Refill

The following must be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area (see Figure 15-41):

1. Create one 16-byte aligned memory-mapped descriptor structure per area where:
   - The next field is set to the physical address of the next descriptor or NULL for the last one.
   - The area field is set with the relevant (x0, y0) (x1, y1) area definition.
   - The ctrl field is set with the requested direction D, with the requested LUT ID L if multiple LUTs are present in the system, and the START bit is asserted to start refilling as soon as the previous area refill is done.
   - The data field is set to the physical address of the corresponding entry table.
2. Write the DMM_PAT_DESCR_i register with the physical address of the first created descriptor.
3. Each area refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. All area refills are done when the DMM_PAT_STATUS_i[0] READY bit is set.
5. A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

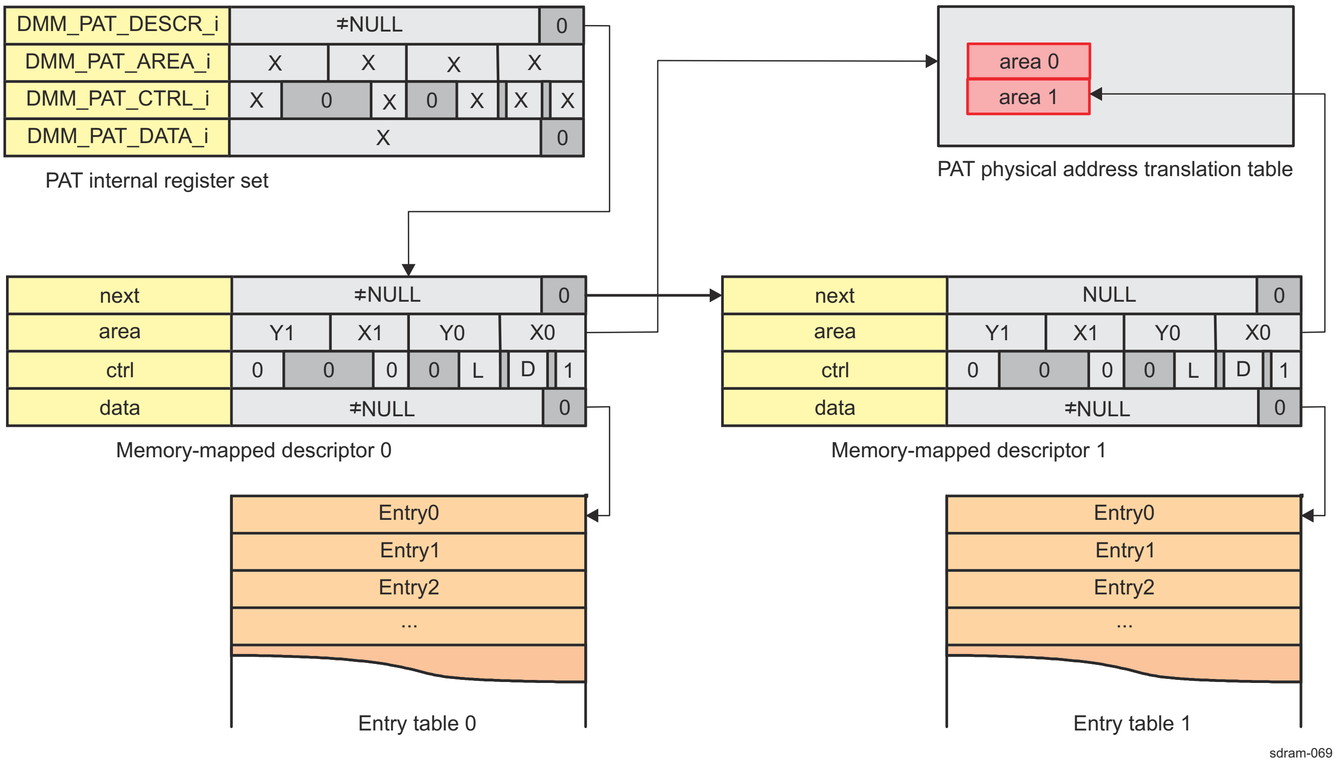


Figure 15-41. Chained Auto-Configured Area Refill Scheme

15.2.4.1.4 Synchronized Auto-Configured Area Refill

The following must be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area (see Figure 15-42):

1. Create one 16-byte aligned memory-mapped descriptor structure per area where:
   - The next field is set to the physical address of the next descriptor or NULL for the last one.
   - The area field is set with the relevant (x0, y0) (x1, y1) area definition.
   - The ctrl field is set with the synchronizing initiator identifier I, the SYNC bit is asserted, the requested direction D and the requested LUT ID L if multiple LUTs are present in the system, and the START bit is asserted to start refilling as soon as the previous area refill is done and initiator I has made one access in the previous area.
   - The data field is set to the physical address of the corresponding entry table.
2. Write the DMM_PAT_DESCR_i register with the physical address of the first created descriptor.
3. Each area refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. All area refills are done when the DMM_PAT_STATUS_i[0] READY bit is set.
5. A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

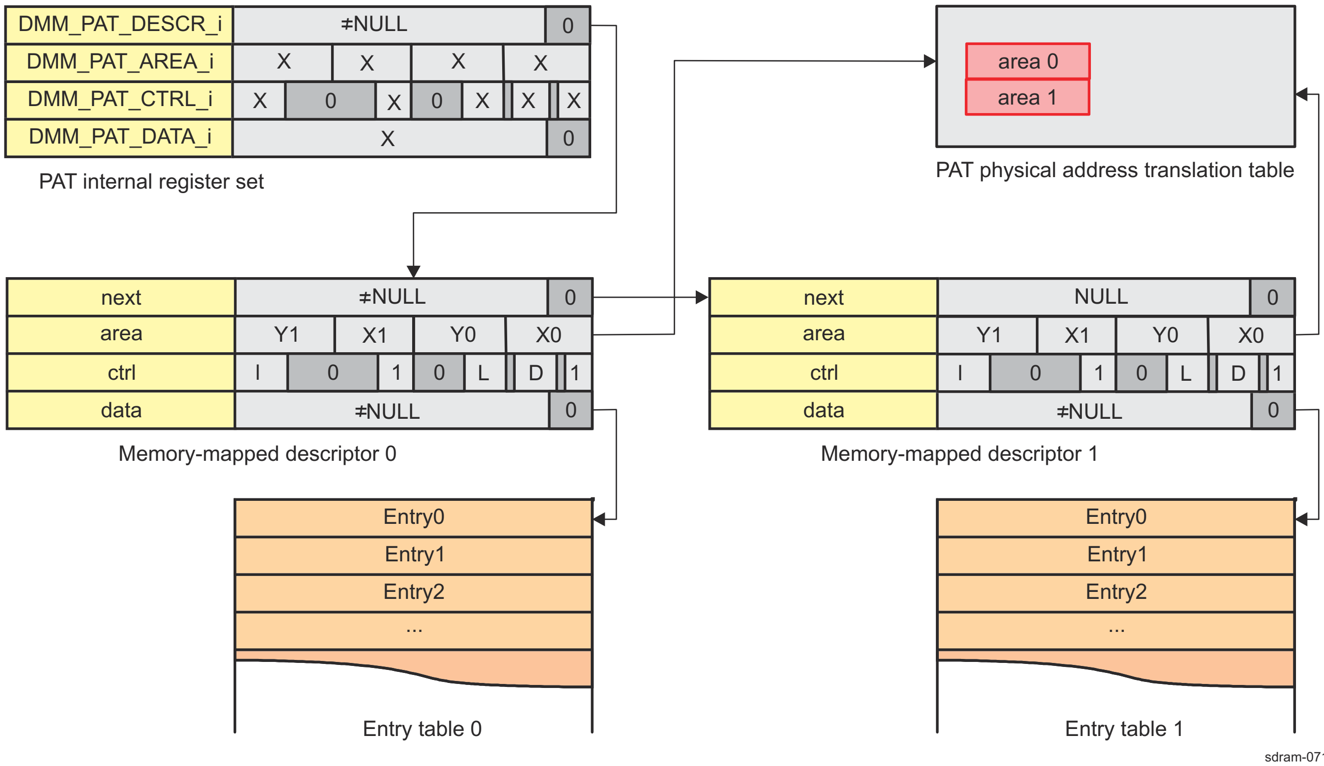


Figure 15-42. Synchronized Auto-Configured Area Refill Scheme

15.2.4.1.5 Cyclic Synchronized Auto-Configured Area Refill

The following must be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area (see Figure 15-43):

1. Create one 16-byte aligned memory-mapped descriptor structure per area where:
   - The next field is set to the physical address of the next descriptor in the circular list.
   - The area field is set with the relevant (x0, y0) (x1, y1) area definition.
   - The ctrl field is set with the synchronizing initiator identifier I, the SYNC bit asserted, the requested direction D and requested LUT ID L if multiple LUTs are present in the system, and the START bit is asserted to start refilling as soon as the previous area refill is done and initiator I has made one access in the previous area.
   - The data field is set to the physical address of the corresponding entry table.
2. Write the DMM_PAT_DESCR_i register with the physical address of the initial descriptor.
3. Each area refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. A new refill can be initiated by writing any value to the DMM_PAT_DESCR_i register to abort the current one.

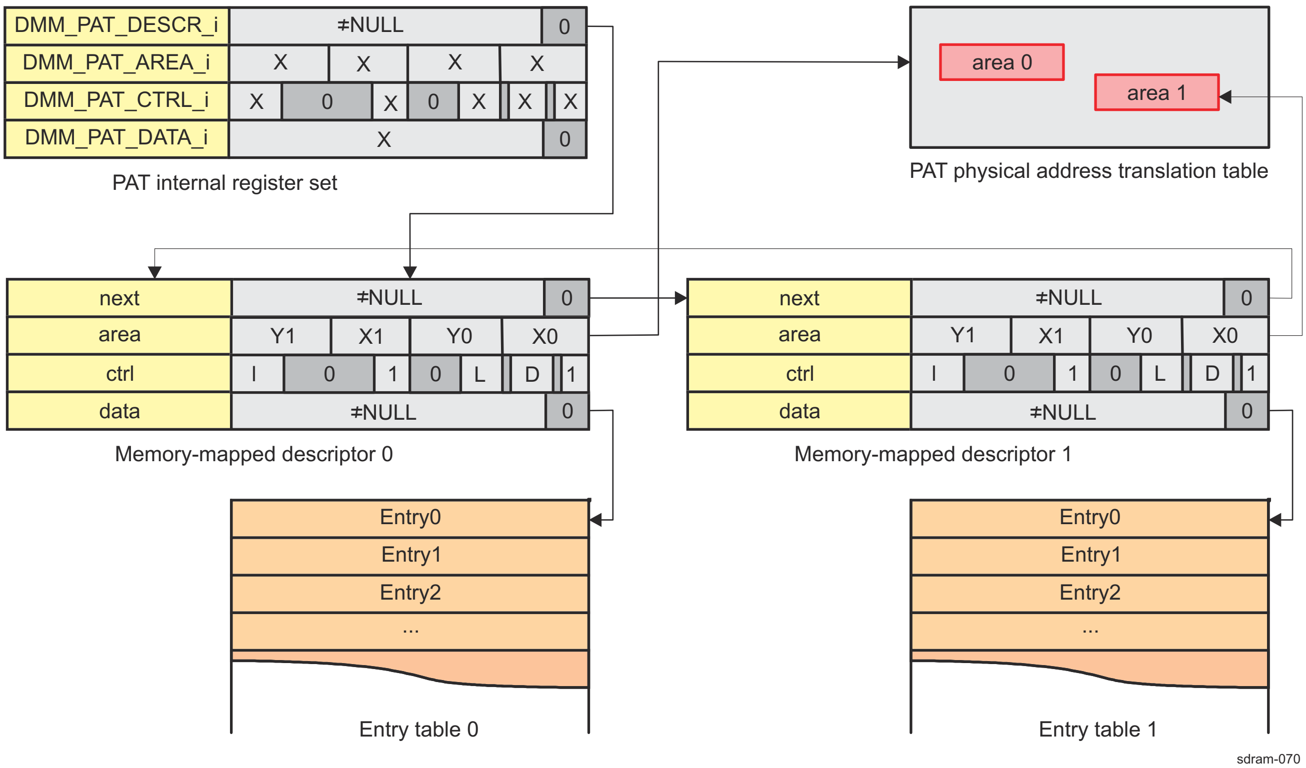


Figure 15-43. Cyclic Synchronized Auto-Configured Area Refill Scheme

---

Note

Never use circular lists of descriptors where all descriptors have the DMM_PAT_CTRL_i[0] START bit set and there is no synchronization. This leads to an endless continuous refill.

15.2.4.2 Addressing Management with LISA

15.2.4.2.1 Case 1: Use of One Memory Controller

In this example, assume there is 1 GiB of external memory evenly spread onto two address spaces. The address range for address space 0 must start at offset 0x2000_0000 (see Table 15-19).

Table 15-19. Address Definition

Address Range	Memory Controller	Memory Controller Address Space(1)	Memory Controller Address Range
0x8000_0000 to 0x9FFF_FFFF	EMIF1	0x0	0x0000_0000 to 0x1FFF_FFFF
0xA000_0000 to 0xBFFF_FFFF	EMIF1	0x0	0x2000_0000 to 0x3FFF_FFFF

(1) For memory controller address spaces, see *Local Interface* in *EMIF Controller*.

This configuration requires two nonoverlapping sections to be set. They can be defined in any order because there is no concern with priority in this case (see Table 15-20):

Table 15-20. Configuration

Bit Field	Section 0 (DMM_LISA_MAP_0)	Section 1 (DMM_LISA_MAP_1)
[31:24] SYS_ADDR	0x80	0xA0
[22:20] SYS_SIZE	0x5	0x5
[19:18] SDRC_INTL	0x0 (not applicable)	0x0 (not applicable)
[17:16] SDRC_ADDRSPC	0x0	0x0
[9:8] SDRC_MAP	0x1 (only EMIF1)	0x1 (only EMIF1)

Table 15-20. Configuration (continued)

Bit Field	Section 0 (DMM_LISA_MAP_0)	Section 1 (DMM_LISA_MAP_1)
[7:0] SDRC_ADDR	0x00	0x20

To check whether an address hits a section, use the 8 upper address bits of the address and mask them with the hit mask: $2^8–2^{SYS\_SIZE}$. If the result is equal to SYS_ADDR, the section is hit.

To define the physical address to be issued to the memory controller, use the 8 upper address bits of the system address, mask them with the address mask: $2^{SYS\_SIZE}–1$, and OR them with SDRC_ADDR. This gives the resulting 8 upper physical address bits. All lower address bits are forwarded unchanged.

Request to address 0x99AE_37F0:

- Upper address bits: 0x99
- Hit mask: $2^8–2^5$ = 0xE0
- Masked upper address bits: 0x80, that is, hits section 0
- Address mask: $2^5–1$ = 0x1F
- Masked upper address bits: 0x19
- OR with SDRC_ADDR: 0x19
- Physical address: 0x19AE_37F0

This request is forwarded to address 0x19AE_37F0, address space 1 of the memory controller.

Request to address 0xB7FF_0340:

- Upper address bits: 0xB7
- Hit mask: $2^8–2^5$ = 0xE0
- Masked upper address bits: 0xA0, that is, hits section 1
- Address mask: $2^5–1$ = 0x1F
- Masked upper address bits: 0x17
- OR with SDRC_ADDR: 0x37
- Physical address: 0x37FF_0340

This request is forwarded to address 0x37FF_0340, address space 0 of the memory controller.

15.2.4.2.2 Case 2: Use of Two Memory Controllers

In the case of 512 MiB interleaved at 128-byte boundaries and 256 MiB noninterleaved on the second memory controller, one address space per memory controller, we have the following results (see Table 15-21).

Table 15-21. Address Definition

Address Range	Memory Controller	Memory Controller Address Space(1)	Memory Controller Address Range
0x8000_0000 to 0x9FFF_FFFF	EMIF1 and EMIF2, interleaved at 128-byte boundaries	0x0 on both controllers	0x0000_0000 to 0x0FFF_FFFF on both controllers
0xA000_0000 to 0xAFFF_FFFF	EMIF2 only	0x0	0x1000_0000 to 0x1FFF_FFFF

(1) For memory controller address spaces, see *Local Interface* in *EMIF Controller*.

Two sections are used to map such a configuration (see Table 15-22).

Table 15-22. Configuration

Bit Field	Section 0 (DMM_LISA_MAP_0)	Section 1 (DMM_LISA_MAP_1)
[31:24] SYS_ADDR	0x80	0xA0
[22:20] SYS_SIZE	0x5	0x4
[19:18] SDRC_INTL	0x1	0x0 (not applicable)
[17:16] SDRC_ADDRSPC	0x0	0x0
[9:8] SDRC_MAP	0x3	0x2
[7:0] SDRC_ADDR	0x00	0x10

Detecting a section hit is the same for the interleaved case as for the single controller case: To check whether an address hits a section, use the 8 upper address bits of the address and mask them with the hit mask: $2^8–2^{SYS\_SIZE}$. If the result is equal to SYS_ADDR, the section is hit.

The physical address generation is modified for the interleaved case. In case of 256-byte interleaving, the first chunk of 256 bytes is mapped to the first controller, the second chunk to the second controller, and so on. This results in system address bit 8 to be decoded as the controller indicator, 0 for the first controller, and 1 for the second controller (system address bit 7 is used for interleaving at the 128-byte boundary, and bit 9 for 512 bytes). This bit is not included in the computed physical address, meaning the upper system address bits [31:9] are shifted to bits [30:8] when generating the physical address.

The rest of the address generation is handled the same as for the single controller case: to define the physical address to be issued to the memory controller, use the 8 upper address bits of the system address, mask them with the address mask: $2^{SYS\_SIZE}–1$, and OR them with SDRC_ADDR. This gives the resulting 8 upper physical address bits. All lower address bits are forwarded unchanged.

Request to address 0x99AE_37F0:

- Upper address bits: 0x99
- Hit mask: $2^8–2^5$ = 0xE0
- Masked upper address bits: 0x80, that is, hits section 0
- Address mask: $2^5–1$ = 0x1F
- Masked upper address bits: 0x19
- Full masked address: 0x19AE_37F0
- Bit 7 is 1, that is, targets the second memory controller
- Full masked shifted address (suppressing bit 7): 0x0CD7_1BF0 (see Section 15.2.4.2.2.1)
- OR upper physical address bits with SDRC_ADDR: 0x0CD7_1BF0
- Physical address: 0x0CD7_1BF0

This request is forwarded to address 0x0CD7_1BF0, address space 0, of the second memory controller.

15.2.4.2.2.1 Address Upper Bits Shifting

Table 15-23 describes shifting the address upper bits.

Table 15-23. Address Upper Bits Shifting

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Address Before Interleaving																															
0	0	0	1	1	0	0	1	1	0	1	0	1	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0
Address After Interleaving																															
0	0	0	0	1	1	0	0	1	1	0	1	0	1	1	1	0	0	0	1	1	0	1	1	1	1	1	1	0	0	0	0

15.2.5 DMM Basic Programming Model

The programming model section:

- Describes how objects can be addressed in all TILER modes and orientations
- Explains how the physical containers and PAT LUT can be shared between different modes
- Does not give an exhaustive description of the TILER and DMM registers, because these are described in Section 15.2.3.5, *DMM*, and Section 15.2.3.6, *TILER*.

15.2.5.1 Global Initialization

This section identifies the requirements for initializing the surrounding modules when the DMM is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DMM (see Table 15-24).

Table 15-24. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. See *Power, Reset, and Clock Management.*
Image processing unit (IPU)	IPU interrupt controller (INTC) configuration must be done to enable the interrupts from the DMM. See *Interrupt Controllers.*
Main processing unit (MPU)	MPU INTC configuration must be done to enable interrupts from the DMM. See *Interrupt Controllers.*
L3_MAIN interconnect	Data interface

15.2.5.2 DMM Module Global Initialization

Theprocedure in Table 15-25 initializes the DMM after a power-on reset (POR).

Table 15-25. DMM Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure the DMM for smart-idle power management mode.	DMM_SYSCONFIG[3:2] IDLE_MODE	0x2

15.2.5.3 DMM Operational Modes Configuration

15.2.5.3.1 Different Operational Modes

The TILER can be virtually accessed in four different modes: 8-, 16-, 32-bit, and page modes. Each mode defines the element granularity to apply isometric transforms (see Table 15-26).

Table 15-26. Coding and Description of TILER Modes

Mode	Name	Granularity (Element Size)
0	8-bit tiled mode	8 bits
1	16-bit tiled mode	16 bits
2	32-bit tiled mode	32 bits
3	Page mode	4096 bytes

15.2.5.3.2 Configuration Settings and LUT Refill

The procedure in Table 15-27 provides the configuration settings and LUT refill.

Table 15-27. Configuration Settings and LUT Refill

Step	Register/Bit Field/Programming Model	Value
PAT configuration refill Engine 0	DMM_PAT_CONFIG[0] MODE0	xxx
PAT configuration refill Engine 1	DMM_PAT_CONFIG[1] MODE1	xxx
Set DMM PAT view register value for each initiator.	DMM_PAT_VIEW0 Vx DMM_PAT_VIEW1 Vx	xxx

Table 15-27. Configuration Settings and LUT Refill (continued)

Step	Register/Bit Field/Programming Model	Value
Set DMM PAT write enable for the initiators.	DMM_PAT_VIEW0 Wx DMM_PAT_VIEW1 Wx	xxx
Choosing container type and access method	DMM_PAT_VIEW_MAP_i	xxx
Define the base address of all view mappings.	DMM_PAT_VIEW_MAP_BASE[31] BASE_ADDR	xxx
Set area definition for DMM physical address translator.	DMM_PAT_AREA_i	xxx
Set the physical address of the current table refill entry data or entry data when in manual mode.	DMM_PAT_DATA_i[31:4] ADDR	xxx
Define the direction of this PAT table refill (S Y X), different from 0x011.	DMM_PAT_CTRL_i[6:4] DIRECTION	xxx
Start a PAT table refill.	DMM_PAT_CTRL_i[0] START	0x1

15.2.5.3.3 Interleaving Settings

The procedure in Table 15-28 provides the interleaving settings.

Table 15-28. Interleaving Settings

Step	Register/Bit Field/Programming Model	Value
Set DMM system section address MSB.	DMM_LISA_MAP_i[31:24] SYS_ADDR	xxx
Set DMM system section size.	DMM_LISA_MAP_i[22:20] SYS_SIZE	xxx
Set SDRC interleaving mode.	DMM_LISA_MAP_i[19:18] SDRC_INTL	xxx
Set SDRC address space.	DMM_LISA_MAP_i[17:16] SDRC_ADDRSPC	xxx
Set SDRC mapping.	DMM_LISA_MAP_i[9:8] SDRC_MAP	xxx
Set SDRC address MSB.	DMM_LISA_MAP_i[7:0] SDRC_ADDR	xxx
Enable/disable DMM memory mapping lock.	DMM_LISA_LOCK[0] LOCK	xxx

15.2.5.3.4 Aliased Tiled View Orientation Settings and LUT Refill

The procedure in Table 15-29 provides the settings for aliased tiled view and LUT refill.

Table 15-29. Aliased Tiled View Orientation Settings and LUT Refill

Step	Register/Bit Field/Programming Model	Value
Set DMM TILER orientation for each initiator.	DMM_TILER_OR0 ORx DMM_TILER_OR1 ORx	xxx
Set DMM TILER write enable for the initiators.	DMM_TILER_OR0 Wx DMM_TILER_OR1 Wx	xxx
Define the base address of all view mappings.	DMM_PAT_VIEW_MAP_BASE[31] BASE_ADDR	xxx
DMM PAT initiator for synchronization	DMM_PAT_CTRL_i[31:28] INITIATOR	xxx
Set DMM PAT table reload synchronization.	DMM_PAT_CTRL_i[16] SYNC	xxx
Set DMM PAT LUT index (when more than one LUTs).	DMM_PAT_CTRL_i[9:8] LUT_ID	xxx
Define the direction of this PAT table refill (S Y X).	DMM_PAT_CTRL_i[6:4] DIRECTION	xxx
Start a PAT table refill.	DMM_PAT_CTRL_i[0] START	0x1

15.2.5.3.5 Priority Settings

The procedure in Table 15-30 provides the sequence to set priorities.

Table 15-30. Priority Settings

Step	Register/Bit Field/Programming Model	Value
Set priority for each initiator.	DMM_PEG_PRIO_k Px	xxx
Set write enable for P_PAT field.	DMM_PEG_PRIO_PAT[4] W_PAT	xxx
Set priority for PAT engine.	DMM_PEG_PRIO_PAT[2:0] W_PAT	xxx

15.2.5.3.6 Error Handling

The procedure in Table 15-31 provides the sequence for error handling.

Table 15-31. Error Handling

Step	Register/Bit Field/Programming Model	Value
Enable interrupt for selected type of error.	DMM_PAT_IRQENABLE_SET[15:9] DMM_PAT_IRQENABLE_SET[7:0]	xxx
When interrupt occurs		
Disable type of error to handle.	DMM_PAT_IRQENABLE_CLR[15:9] DMM_PAT_IRQENABLE_CLR[7:0]	xxx
Check error status.	DMM_PAT_IRQSTATUS[15:9] DMM_PAT_IRQSTATUS[7:0]	xxx

15.2.5.3.7 PAT Programming Model

The PAT maps the tiled data anywhere in the 4-GiB physical address range, with a PAGE granularity. (The TILER page is the granularity of physical memory allocation in the TILER container. Each page is 4kiB).

A PAT view defines the kind of PAT to perform for each page, 8-, 16-, and 32-bit mode access. Each mode in each PAT view can be programmed in two different modes: direct translation and indirect translation.

15.2.5.3.7.1 PAT in Direct Translation Mode

In this mode, the PAT performs a translation of the 128-MiB virtual container as a whole in the physical address space (that is, in the SDRAM). This mode is used only for debug or in case of a DMM without a PAT.

15.2.5.3.7.2 PAT in Indirect Translation Mode

This is the most commonly used mode. In this mode, the PAT performs a translation of each 4-KiB page individually. In this way the 128-MiB virtual address space can be scattered in the whole 2 GiB of the physical address space. This is achieved by using a 32,678-word LUT that converts each page index (32,768 possible values) into 19 address bits that represent this page address in the physical memory. The main characteristic of this mode is there is no constraint on the use of the physical memory except that it is uses a multiple of 4-KiB areas located at 4-KiB boundaries in the physical memory. In this mode, the translation vector is in the internal 32-k entry PAT vector table at the index given by bits [26:12] of the input virtual address, and CONT_x = 0.

Programming sequence:

1. Set the DMM_PAT_VIEW0 register (or DMM_PAT_VIEW1, depending on the L3 CONN_ID of the initiator that is to perform the tiled accesses) with the appropriate value. For example, if the L3 CONN_ID equals 0, then the DMM_PAT_VIEW0 register must be programmed with the value 0x0000_0001 (PAT view 1 selected for initiator 0).
2. Set DMM_PAT_VIEW_MAP_BASE[31] = 0x8000_0000 (must always be programmed at 1 to select the upper 2 GiB of the physical address space that corresponds to the external memory).
3. Set all 16-bit tiled accesses in indirect translation mode by setting at least the field ACCESS_16 to 1 (DMM_PAT_VIEW_MAP_1 = 0x0000_8000), program DMM_PAT_VIEW_MAP_1 at 0x8080_8080.

15.2.5.4 Addressing an Object in Tiled Mode

This section describes how a frame-buffer with a top-left pixel at (x0,y0) in the natural view container and a bottom right pixel at (x1,y1) in the natural view container is addressed in any tiled mode and orientation.

15.2.5.4.1 Frame-Buffer Addressing

Note

In this section, the (x,y) coordinates are given in pixel units.

Given an (S, $\overline{Y}$, $\overline{X}$) orientation, the (x,y) coordinates of the first pixel of the frame in the oriented view (where W = the width and H = the height in pixels of the container in the considered frame mode) is:

$(x,y) = (x_{or}, y_{or})$ when S, (y_{or}, x_{or}) otherwise
with:
$x_{or} = x_0$ when $\overline{X}$, $W\text{-}1\text{-}x_1$ otherwise
$y_{or} = y_0$ when $\overline{Y}$, $H\text{-}1\text{-}y_1$ otherwise

Given a (M_1, M_0) TILER mode where M_1 # 0 or M_0 # 0, the size in bytes of a pixel is 2^{2M1+M0}.

Therefore, given an (S, $\overline{Y}$, $\overline{X}$) orientation and a $n(M_1, M_0)$ TILER mode where M_1 # 0 or M_0 # 0, the byte offset of the base address of the considered oriented frame in its container is:
base_address($(x_0, y_0), (x_1, y_1)$, (S, $\overline{Y}$, $\overline{X}$), (M_1, M_0)) = $(y_{or}W + x_{or})\ P_5$ when S, $(x_{or}H + y_{or})\ P_5$ otherwise
with:
$x_{or} = x_0$ when $\overline{X}$, $W\text{-}1\text{-}x_1$ otherwise
$y_{or} = y_0$ when $\overline{Y}$, $H\text{-}1\text{-}y_1$ otherwise
$P_5 = 2^{2M1+M0}$

In its natural orientation, the TILER container consists of 8192 lines of 16,384 pixels of 8 bits, or 4096 lines of 16,384 pixels of 16 bits, or 4096 lines of 8192 pixels of 32 bits.

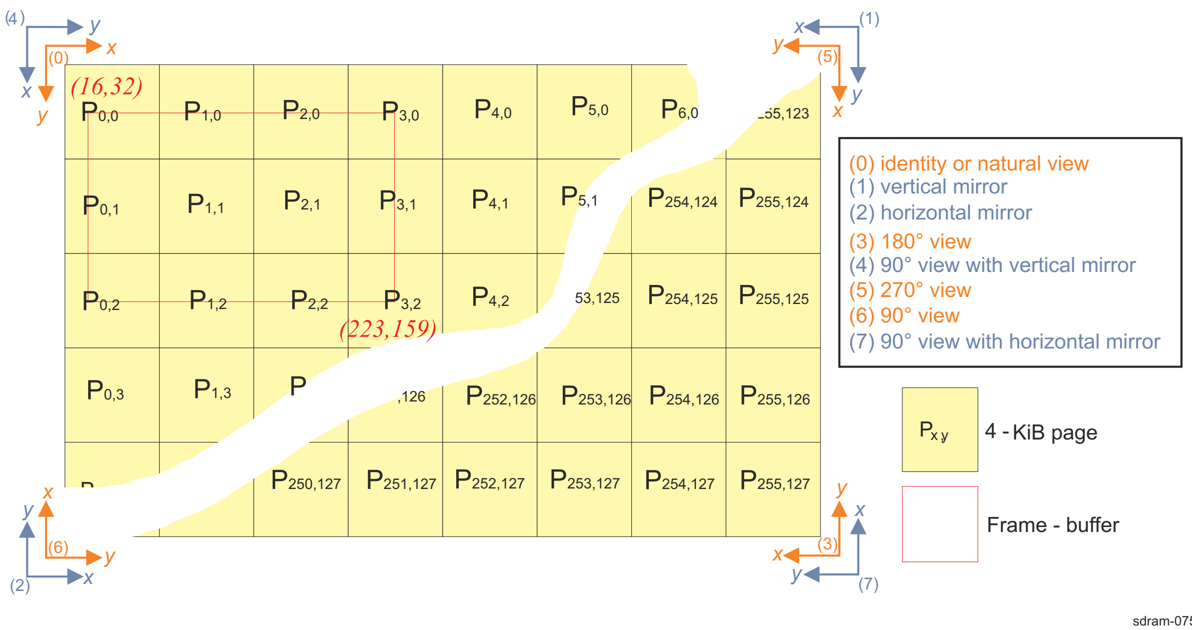


Figure 15-44. Example of 8-Bit Frame-Buffer Addressing in any Orientation

This frame-buffer address generation is independent from the page size. For instance, the 8-bit frame-buffer shown in Figure 15-44 that ranges from the top-left pixel at (16, 32) to the bottom-right pixel at (223, 159) in the natural orientation view, corresponds to the 29-bit view address offsets and full 33-bit TILER addresses given in Table 15-32.

Table 15-32. 29-Bit View Address Offset and 33-Bit Full TILER Address for an 8-Bit Frame-Buffer

S	$\overline{Y}$	$\overline{X}$	x_{or}	y_{or}	29-Bit Address Offset in View	Full 33-Bit TILER Address
0	0	0	16	32	00080010h	100080010h
0	0	1	16,160	32	00083F20h	120083F20h
0	1	0	16	8032	07D80010h	147D80010h
0	1	1	16,160	8032	07D83F20h	167D83F20h
1	0	0	16	32	00020020h	180020020h
1	0	1	16,160	32	07E40020h	1A7E40020h
1	1	0	16	8032	00021F60h	1C0021F60h
1	1	1	16,160	8032	07E41F60h	1E7E41F60h

In this example the TILER is addressed in 8-bit mode, which translates to addresses with all mode bits (bits 27 and 28) cleared in the 29-bit address offset in view and in the full 33-bit TILER address.

15.2.5.4.2 TILER Page Mapping

Let:

- c (c = 0, 1, or 2) be the TILER page configuration (0 for 4-kiB pages, 1 for 16-kiB pages, and 2 for 64-kiB pages)
- (x_0, y_0) be the top-left pixel and (x_1, y_1) be the bottom-right pixel of a frame in its natural orientation
- W be the width and H the height in pixels of the container in the considered frame mode

The top-left page coordinates (P_{x0}, P_{y0}) and bottom-right page coordinates (P_{x1}, P_{y1}) that correspond to the frame are given by: $P_{x0} = (64.x_0.2^{2-c}) / W$, $P_{y0} = (32.y_0.2^{2-c}) / H$, $P_{x1} = (64.x_1.2^{2-c}) / W$ and $P_{y1} = (32.y_1.2^{2-c}) / H$.

In its natural orientation, the TILER container consists of 8192 lines of 16,384 pixels of 8 bits, or 4096 lines of 16,384 pixels of 16 bits, or 4096 lines of 8192 pixels of 32 bits.

Note

The page area type has the same structure as the DMM_PAT_AREA_i registers. For instance, the 8-bit frame-buffer described in Figure 15-44 and ranging from the top-left pixel at (16, 32) to the bottom-right pixel at (223, 159) in the natural orientation view, is mapped from the top-left page at P0,0 to the bottom-right page P3,2.

15.2.5.5 Addressing an Object in Page Mode

In page mode, the orientation modifies only the sequence of accessed pages, not their content. In this respect the orientation must be seen as a way to optimize the one-dimensional (1D) object allocation in a TILER container by allowing a 1D object spanning multiple pages to map a set of adjacent pages in any direction. For instance, a 1.25-MiB object, mapped on 320 pages of 4kiB can be allocated in a TILER container in any of the eight orientations (see Section 15.2.3.6.1.8.3, *Element Ordering in the TILER Container*).

The initial page can be chosen freely among all pages. In this respect, the address offset of a page in a TILER container is expressed as:

Let:

- c (c = 0, 1 or 2) be the TILER page configuration (0 for 4-kiB pages, 1 for 16-kiB pages, and 2 for 64-kiB pages)
- (S, $\overline{Y}$, $\overline{X}$) be the orientation of the considered 1D object in page mode
- (P_x, P_y) be the coordinate of the initial page in the natural orientation view

The byte offset of the base address of the considered initial page in its oriented container is:
base_address((P_x, P_y), (S, $\overline{Y}$, $\overline{X}$)) = $4096.(64.y_{or}.2^{2-c} + x_{or}).2^{2.c}$ when S, $4096.(32.x_{or}.2^{2-c} + y_{or}).2^{2.c}$ otherwise with:
$x_{or} = P_x$ when $\overline{X}$, $64.2^{2-c}-1-P_x$ otherwise
$y_{or} = P_y$ when $\overline{Y}$, $32.2^{2-c}-1-P_y$ otherwise

For instance, a 1D object starting from the page P32,63 in the natural orientation view on a DMM using 4-kiB pages, corresponds to the 29-bit view address offsets and full 33-bit TILER addresses given in Table 15-33.

Table 15-33. 29-Bit View Address Offset and 33-Bit Full TILER Address for a 1D Object

S	$\overline{Y}$	$\overline{X}$	x_{or}	y_{or}	29-Bit Address Offset in View	Full 33-Bit TILER Address
0	0	0	32	63	1BF20000h	11BF20000h
0	0	1	223	63	1BFDF000h	13BFDF000h
0	1	0	32	64	1C020000h	15C020000h
0	1	1	223	64	1C0DF000h	17C0DF000h
1	0	0	32	63	1903F000h	19903F000h

Table 15-33. 29-Bit View Address Offset and 33-Bit Full TILER Address for a 1D Object (continued)

S	$\overline{Y}$	$\overline{X}$	x_{or}	y_{or}	29-Bit Address Offset in View	Full 33-Bit TILER Address
1	0	1	223	63	1EFBF000h	1BEFBF000h
1	1	0	32	64	19040000h	1D9040000h
1	1	1	223	64	1EFC0000h	1FEFC0000h

In this example the TILER is addressed in page mode, which translates to addresses with mode bits (bits 27 and 28) set in the 29-bit address offset in view and full 33-bit TILER address.

15.2.5.6 Sharing Containers Between Different Modes

When allocating objects in TILER containers, ensure that no two objects physically overlap.

In this respect, and to ease the object allocation and deallocation, it is strongly advised to share a TILER page only for different objects that do both of the following:

- Belong to the same mode
- Have the same lifetime (that is, are allocated and deallocated simultaneously)

Two objects of different modes can physically overlap if:

- One physical page is mapped twice in two different locations of the DMM LUT.
- The two objects share the DMM LUT and the intersection of their two page set is not empty.

  These two issues can be easily avoided by allocating a physical page only once in the DMM LUT.

  The second issue is a bit more difficult, especially if the allocation of objects in the various TILER containers must be dynamic. Managing the fragmentation within a flat memory model in a space-constrained system is not straightforward, and is the main reason for the existence of MMUs in most current processors, which adds yet another constraint to the problem (the y dimension, which makes it even more difficult).

Returning to the CPU-analogy, the memory fragmentation issue is mostly solved by using a virtual memory space larger than the actual physical memory space and a virtual-to-physical translation system. This allows contiguous virtual memory allocation when sufficient physical memory is available and the larger contiguous physical buffer is smaller than the requested memory allocation.

Similarly, the DMM answer to this object allocation in the TILER containers is two-fold:

- The DMM must have sufficient virtual address space in all modes to permit a static allocation of a pool of virtual buffers in all TILER modes for all TILER-aware initiators.
- Each TILER-aware initiator must dynamically manage its own pool of virtual buffers.

15.2.6 DMM Register Manual

This section provides information about the DMM registers. Table 15-34 describes the DMM instance.

15.2.6.1 DMM Instance Summary

Table 15-34. DMM Instance Summary

Module Name	Base Address	Size
DMM	0x4E00 0000	32 MiB

15.2.6.2 DMM Registers

15.2.6.2.1 DMM Register Summary

Table 15-35 provides a summary of the DMM registers.

Table 15-35. DMM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address DMM
DMM_REVISION	R	32	0x0000 0000	0x4E00 0000
DMM_HWINFO	R	32	0x0000 0004	0x4E00 0004
DMM_LISA_HWINFO	R	32	0x0000 0008	0x4E00 0008
DMM_SYSCONFIG	RW	32	0x0000 0010	0x4E00 0010
DMM_LISA_LOCK	RW	32	0x0000 001C	0x4E00 001C
DMM_EMERGENCY	RW	32	0x0000 0020	0x4E00 0020
DMM_LISA_MAP_i (1)	RW	32	0x0000 0040 + (0x4 * i)	0x4E00 0040 + (0x4 * i)
DMM_TILER_HWINFO	R	32	0x0000 0208	0x4E00 0208
DMM_TILER_OR0	RW	32	0x0000 0220	0x4E00 0220
DMM_TILER_OR1	RW	32	0x0000 0224	0x4E00 0224
DMM_PAT_HWINFO	R	32	0x0000 0408	0x4E00 0408
DMM_PAT_GEOMETRY	R	32	0x0000 040C	0x4E00 040C
DMM_PAT_CONFIG	RW	32	0x0000 0410	0x4E00 0410
DMM_PAT_VIEW0	RW	32	0x0000 0420	0x4E00 0420
DMM_PAT_VIEW1	RW	32	0x0000 0424	0x4E00 0424
DMM_PAT_VIEW_MAP_i (1)	RW	32	0x0000 0440 + (0x4 * i)	0x4E00 0440 + (0x4 * i)
DMM_PAT_VIEW_MAP_BASE	RW	32	0x0000 0460	0x4E00 0460
DMM_PAT_IRQ_EOI	RW	32	0x0000 0478	0x4E00 0478
DMM_PAT_IRQSTATUS_RAW	RW	32	0x0000 0480	0x4E00 0480
DMM_PAT_IRQSTATUS	RW	32	0x0000 0490	0x4E00 0490
DMM_PAT_IRQENABLE_SET	RW	32	0x0000 04A0	0x4E00 04A0
DMM_PAT_IRQENABLE_CLR	RW	32	0x0000 04B0	0x4E00 04B0
DMM_PAT_STATUS_i (1)	R	32	0x0000 04C0 + (0x4 * i)	0x4E00 04C0 + (0x4 * i)
DMM_PAT_DESCR_i (1)	RW	32	0x0000 0500 + (0x10 * i)	0x4E00 0500 + (0x10 * i)
DMM_PAT_AREA_i (1)	RW	32	0x0000 0504 + (0x10 * i)	0x4E00 0504 + (0x10 * i)
DMM_PAT_CTRL_i (1)	RW	32	0x0000 0508 + (0x10 * i)	0x4E00 0508 + (0x10 * i)
DMM_PAT_DATA_i (1)	RW	32	0x0000 050C + (0x10 * i)	0x4E00 050C + (0x10 * i)
DMM_PEG_HWINFO	R	32	0x0000 0608	0x4E00 0608
DMM_PEG_PRIO_k (2)	RW	32	0x0000 0620 + (0x4 * k)	0x4E00 0620 + (0x4 * k)
DMM_PEG_PRIO_PAT	RW	32	0x0000 0640	0x4E00 0640

(1) i = 0 to 3 for DMM
(2) k = 0 to 7

15.2.6.2.2 DMM Register Description

Table 15-36. DMM_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4E00 0000	**Instance**	DMM
Description	DMM revision number		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REVISION</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision number	R	0x-(1)

(1) TI internal data

Table 15-37. DMM_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x4E00 0004	**Instance**	DMM
Description	DMM hardware configuration		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">RESERVED</td><td colspan="4">ROBIN_CNT</td><td colspan="4">RESERVED</td><td colspan="4">ELLA_CNT</td><td colspan="4">RESERVED</td><td colspan="4">TILER_CNT</td></tr>
</table>

Table 15-38. DMM_LISA_HWINFO

Address Offset	0x0000 0008		
Physical Address	0x4E00 0008	**Instance**	DMM
Description	DMM hardware configuration for LISA		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="20">RESERVED</td><td colspan="4">SDRC_CNT</td><td colspan="3">RESERVED</td><td colspan="5">SECTION_CNT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:8	SDRC_CNT	Number of attached SDRAM controllers	R	0x2
7:5	RESERVED	Reserved	R	0x0
4:0	SECTION_CNT	Number of DMM sections	R	0x04

Table 15-39. DMM_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4E00 0010	**Instance**	DMM
Description	DMM clock management configuration		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="28">RESERVED</td><td colspan="2">IDLE_MODE</td><td colspan="2">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000

Bits	Field Name	Description	Type	Reset
3:2	IDLE_MODE	Configuration of the local target state management mode. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. **Backup mode, for debug only.** 0x1: No-idle mode: local target never enters idle state. **Backup mode, for debug only.** 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wake-up events. 0x3: Reserved	RW	0x2
1:0	RESERVED	Reserved	RW W0Only	0x0

Table 15-40. DMM_LISA_LOCK

Address Offset	0x0000 001C		
Physical Address	0x4E00 001C	**Instance**	DMM
Description	DMM memory mapping lock		
Type	RW		

31–1	0
RESERVED	LOCK

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Should be written as 0	R	0x0000 0000
0	LOCK	DMM lock map Write 0x0: No effect (clear on reset only) Read 0x0: DMM_LISA_MAP_i unlocked Read 0x1: DMM_LISA_MAP_i locked Write 0x1: Locking DMM_LISA_MAP_i registers	RW	0

Table 15-41. DMM_EMERGENCY

Address Offset	0x0000 0020		
Physical Address	0x4E00 0020	**Instance**	DMM
Description	DMM memory mapping register		
Type	RW		

31–21	20–16	15–1	0
RESERVED	WEIGHT	RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	R	0x000
20:16	WEIGHT	Weight for the LISA arbitration when any bit of the vector Mflag[63:0] is set. The recommendation is to set this field to 0x8 with ENABLE =1, after reset.	RW	0x04
15:1	Reserved	Reserved	R	0x0000

Bits	Field Name	Description	Type	Reset
0	ENABLE	0: Emergency feature is disabled. 1: Enable the emergency feature. LISA arbitration priority is higher for the initiator that set Mflag input of this initiator. The recommendation is to enable the feature (=1) after reset.	RW	0

Table 15-42. DMM_LISA_MAP_i

Address Offset	0x0000 0040 + (0x4 * i)	**Index**	i = 0 to 3
Physical Address	0x4E00 0040 + (0x4 * i)	**Instance**	DMM
Description	DMM memory mapping register		
Type	RW		

31:24	23	22:20	19:18	17:16	15:10	9:8	7:0
SYS_ADDR	RESERVED	SYS_SIZE	SDRC_INTL	SDRC_ADDRSPC	RESERVED	SDRC_MAP	SDRC_ADDR

Bits	Field Name	Description	Type	Reset
31:24	SYS_ADDR	DMM system section address MSB for view mapping i	RW	0x00
23	RESERVED	Reserved	RW W0Only	0
22:20	SYS_SIZE	DMM system section size for view mapping i 0x0: 16-MiB section 0x1: 32-MiB section 0x2: 64-MiB section 0x3: 128-MiB section 0x4: 256-MiB section 0x5: 512-MiB section 0x6: 1-GiB section 0x7: 2-GiB section	RW	0x0
19:18	SDRC_INTL	SDRAM controller interleaving mode 0x0: No interleaving 0x1: 128-byte interleaving 0x2: 256-byte interleaving 0x3: 512-byte interleaving The 128-/256-/512-byte interleaving applies only to nontiled regions. If accesses are made to tiled regions, interleaving is forced to 1kiB. SDRC_INTL is don't care if SDRC_MAP is not 0x3 (no interleaving).	RW	0x0
17:16	SDRC_ADDRSPC	SDRAM controller address space for view mapping i	RW	0x0
15:10	RESERVED	Reserved	RW W0Only	0x00
9:8	SDRC_MAP	SDRAM controller mapping for view mapping i 0x0: Unmapped 0x1: Mapped on EMIF1 only (not interleaved) 0x2: Mapped on EMIF2 only (not interleaved) 0x3: Mapped on EMIF1 and EMIF2 (interleaved) To enable interleaving, SDRC_MAP must be 0x3 and SDRC_INTL must be a nonzero value.	RW	0
7:0	SDRC_ADDR	SDRAM controller address MSB for view mapping i	RW	0x00

Table 15-43. DMM_TILER_HWINFO

Address Offset	0x0000 0208

Table 15-43. DMM_TILER_HWINFO (continued)

Physical Address	0x4E00 0208	**Instance**	DMM
Description	DMM hardware configuration for TILER		
Type	R		

31:7	6:0
RESERVED	OR_CNT

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x000 0000
6:0	OR_CNT	Number of TILER orientation entries Read 0x4: Four orientation entries Read 0x20: Thirty-two orientation entries Read 0x40: Sixty-four orientation entries Read 0x2: Two orientation entries Read 0x1: One orientation entry Read 0x8: Eight orientation entries Read 0x10: Sixteen orientation entries	R	0x10

Table 15-44. DMM_TILER_OR0

Address Offset	0x0000 02200	**Index**	0
Physical Address	0x4E00 0220	**Instance**	DMM
Description	DMM TILER orientation (initiators 0 to 7)		
Type	RW		

31	30:28	27	26:24	23	22:20	19	18:16	15	14:12	11	10:8	7	6:4	3	2:0
W7	OR7	W6	OR6	W5	OR5	W4	OR4	W3	OR3	W2	OR2	W1	OR1	W0	OR0

Bits	Field Name	Description	Type	Reset
31	W7	Write-enable for OR7 bit field Write 0x0: OR7 field is unchanged. Write 0x1: OR7 field is updated.	RW	0
30:28	OR7	Orientation for initiator 7	RW	0x0
27	W6	Write-enable for OR6 bit field Write 0x0: OR6 field is unchanged. Write 0x1: OR6 field is updated.	RW	0
26:24	OR6	Orientation for initiator 6	RW	0x0
23	W5	Write-enable for OR5 bit field Write 0x0: OR5 field is unchanged. Write 0x1: OR5 field is updated.	RW	0
22:20	OR5	Orientation for initiator 5	RW	0x0
19	W4	Write-enable for OR4 bit field Write 0x0: OR4 field is unchanged. Write 0x1: OR4 field is updated.	RW	0
18:16	OR4	Orientation for initiator 4	RW	0x0
15	W3	Write-enable for OR3 bit field Write 0x0: OR3 field is unchanged. Write 0x1: OR3 field is updated.	RW	0
14:12	OR3	Orientation for initiator 3	RW	0x0

Bits	Field Name	Description	Type	Reset
11	W2	Write-enable for OR2 bit field Write 0x0: OR2 field is unchanged. Write 0x1: OR2 field is updated.	RW	0
10:8	OR2	Orientation for initiator 2	RW	0x0
7	W1	Write-enable for OR1 bit field Write 0x0: OR1 field is unchanged. Write 0x1: OR1 field is updated.	RW	0
6:4	OR1	Orientation for initiator 1	RW	0x0
3	W0	Write-enable for OR0 bit field Write 0x0: OR0 field is unchanged. Write 0x1: OR0 field is updated.	RW	0
2:0	OR0	Orientation for initiator 0	RW	0x0

Table 15-45. DMM_TILER_OR1

Address Offset	0x0000 02204	**Index**	0
Physical Address	0x4E00 0224	**Instance**	DMM
Description	DMM TILER orientation (initiators 8 to 15)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	OR15			W14	OR14			W13	OR13			W12	OR12			W11	OR11			W10	OR10			W9	OR9			W8	OR8		

Bits	Field Name	Description	Type	Reset
31	W15	Write-enable for OR15 bit field Write 0x0: OR15 field is unchanged. Write 0x1: OR15 field is updated.	RW	0
30:28	OR15	Orientation for initiator 15	RW	0x0
27	W14	Write-enable for OR14 bit field Write 0x0: OR14 field is unchanged. Write 0x1: OR14 field is updated.	RW	0
26:24	OR14	Orientation for initiator 14	RW	0x0
23	W13	Write-enable for OR13 bit field Write 0x0: OR13 field is unchanged. Write 0x1: OR13 field is updated.	RW	0
22:20	OR13	Orientation for initiator 13	RW	0x0
19	W12	Write-enable for OR12 bit field Write 0x0: OR12 field is unchanged. Write 0x1: OR12 field is updated.	RW	0
18:16	OR12	Orientation for initiator 12	RW	0x0
15	W11	Write-enable for OR11 bit field Write 0x0: OR11 field is unchanged. Write 0x1: OR11 field is updated.	RW	0
14:12	OR11	Orientation for initiator 11	RW	0x0
11	W10	Write-enable for OR10 bit field Write 0x0: OR10 field is unchanged. Write 0x1: OR10 field is updated.	RW	0
10:8	OR10	Orientation for initiator 10	RW	0x0

Bits	Field Name	Description	Type	Reset
7	W9	Write-enable for OR9 bit field Write 0x0: OR9 field is unchanged. Write 0x1: OR9 field is updated.	RW	0
6:4	OR9	Orientation for initiator 9	RW	0x0
3	W8	Write-enable for OR8 bit field Write 0x0: OR8 field is unchanged. Write 0x1: OR8 field is updated.	RW	0
2:0	OR8	Orientation for initiator 8	RW	0x0

Table 15-46. DMM_PAT_HWINFO

Address Offset	0x0000 0408		
Physical Address	0x4E00 0408	**Instance**	DMM
Description	DMM hardware configuration for PAT		
Type	R		

31:29	28:24	23:21	20:16	15:12	11:8	7	6:0
RESERVED	ENGINE_CNT	RESERVED	LUT_CNT	RESERVED	VIEW_MAP_CNT	RESERVED	VIEW_CNT

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:24	ENGINE_CNT	Number of PAT refill engines	R	0x04
23:21	RESERVED	Reserved	R	0x0
20:16	LUT_CNT	Number of PAT LUT for page-grained physical address translation	R	0x01
15:12	RESERVED	Reserved	R	0x0
11:8	VIEW_MAP_CNT	Number of internal PAT view mappings. Read 0x1: One view map Read 0x2: Two view maps Read 0x4: Four view maps Read 0x8: Eight view maps	R	0x4
7	RESERVED	Reserved	R	0
6:0	VIEW_CNT	Number of PAT view entries Read 0x1: One view entry Read 0x2: Two view entries Read 0x4: Four view entries Read 0x8: Eight view entries Read 0x10: Sixteen view entries Read 0x20: Thirty-two view entries Read 0x40: Sixty-four view entries	R	0x10

Table 15-47. DMM_PAT_GEOMETRY

Address Offset	0x0000 040C		
Physical Address	0x4E00 040C	**Instance**	DMM
Description	PAT geometry-related settings		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CONT_HGHT	RESERVED	CONT_WDTH	RESERVED	ADDR_RANGE	RESERVED	PAGE_SZ

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26:24	CONT_HGHT	Container height in pages Read 0x1: Container height of 32 pages Read 0x2: Container height of 64 pages Read 0x4: Container height of 128 pages Read 0x8: Container height of 256 pages	R	0x8
23:20	RESERVED	Reserved	R	0x0
19:16	CONT_WDTH	Container width in pages Read 0x2: Container width of 64 pages Read 0x4: Container width of 128 pages Read 0x8: Container width of 256 pages	R	0x8
15:14	RESERVED	Reserved	R	0x0
13:8	ADDR_RANGE	PAT output physical address range Read 0x1: 128-MiB range Read 0x2: 256-MiB range Read 0x4: 512-MiB range Read 0x8: 1-GiB range Read 0x10: 2-GiB range Read 0x20: 4-GiB range	R	0x10
7:5	RESERVED	Reserved	R	0x0
4:0	PAGE_SZ	Page size in 4-kiB granularity Read 0x1: 4-kiB page Read 0x4: 16-kiB page Read 0x10: 64-kiB page	R	0x01

Table 15-48. DMM_PAT_CONFIG

Address Offset	0x0000 0410		
Physical Address	0x4E00 0410	**Instance**	DMM
Description	This is the PAT configuration register aimed at defining the major PAT configuration of each refill engine.		
Type	RW		

31:4	3	2	1	0
RESERVED	MODE3	MODE2	MODE1	MODE0

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3	MODE3	Mode of refill engine 3 0: Normal mode 1: Direct LUT access	RW	0
2	MODE2	Mode of refill engine 2 0: Normal mode 1: Direct LUT access	RW	0

Bits	Field Name	Description	Type	Reset
1	MODE1	Mode of refill engine 1 0: Normal mode 1: Direct LUT access	RW	0
0	MODE0	Mode of refill engine 0 0: Normal mode 1: Direct LUT access	RW	0

Table 15-49. DMM_PAT_VIEW0

Address Offset	0x0000 0420	**Index**	
Physical Address	0x4E00 0420	**Instance**	DMM
Description	DMM PAT View register (initiators 0 to 7)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W7	RESERVED	V7		W6	RESERVED	V6		W5	RESERVED	V5		W4	RESERVED	V4		W3	RESERVED	V3		W2	RESERVED	V2		W1	RESERVED	V1		W0	RESERVED	V0	

Bits	Field Name	Description	Type	Reset
31	W7	Write-enable for V7 bit field Write 0x0: V7 field is unchanged. Write 0x1: V7 field is updated.	RW	0
30	RESERVED	Reserved	RW W0Only	0
29:28	V7	PAT view for initiator 7	RW	0x0
27	W6	Write-enable for V6 bit field Write 0x0: V6 field is unchanged. Write 0x1: V6 field is updated.	RW	0
26	RESERVED	Reserved	RW W0Only	0
25:24	V6	PAT view for initiator 6	RW	0x0
23	W5	Write-enable for V5 bit field Write 0x0: V5 field is unchanged. Write 0x1: V5 field is updated.	RW	0
22	RESERVED	Reserved	RW W0Only	0
21:20	V5	PAT view for initiator 5	RW	0x0
19	W4	Write-enable for V4 bit field Write 0x0: V4 field is unchanged. Write 0x1: V4 field is updated.	RW	0
18	RESERVED	Reserved	RW W0Only	0
17:16	V4	PAT view for initiator 4	RW	0x0
15	W3	Write-enable for V3 bit field Write 0x0: V3 field is unchanged. Write 0x1: V3 field is updated.	RW	0
14	RESERVED	Reserved	RW W0Only	0
13:12	V3	PAT view for initiator 3	RW	0x0

Bits	Field Name	Description	Type	Reset
11	W2	Write-enable for V2 bit field Write 0x0: V2 field is unchanged. Write 0x1: V2 field is updated.	RW	0
10	RESERVED	Reserved	RW W0Only	0
9:8	V2	PAT view for initiator 2	RW	0x0
7	W1	Write-enable for V1 bit field Write 0x0: V1 field is unchanged. Write 0x1: V1 field is updated.	RW	0
6	RESERVED	Reserved	RW W0Only	0
5:4	V1	PAT view for initiator 1	RW	0x0
3	W0	Write-enable for V0 bit field Write 0x0: V0 field is unchanged. Write 0x1: V0 field is updated.	RW	0
2	RESERVED	Reserved	RW W0Only	0
1:0	V0	PAT view for initiator 0	RW	0x0

Table 15-50. DMM_PAT_VIEW1

Address Offset	0x0000 0424	**Index**	
Physical Address	0x4E00 0424	**Instance**	DMM
Description	DMM PAT view register (initiators 8 to 15)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	RESERVED	V15		W14	RESERVED	V14		W13	RESERVED	V13		W12	RESERVED	V12		W11	RESERVED	V11		W10	RESERVED	V10		W9	RESERVED	V9		W8	RESERVED	V8	

Bits	Field Name	Description	Type	Reset
31	W15	Write-enable for V15 bit field Write 0x0: V15 field is unchanged. Write 0x1: V15 field is updated.	RW	0
30	RESERVED	Reserved	RW W0Only	0
29:28	V15	PAT view for initiator 15	RW	0x0
27	W14	Write-enable for V14 bit field Write 0x0: V14 field is unchanged. Write 0x1: V14 field is updated.	RW	0
26	RESERVED	Reserved	RW W0Only	0
25:24	V14	PAT view for initiator 14	RW	0x0
23	W13	Write-enable for V13 bit field Write 0x0: V13 field is unchanged. Write 0x1: V13 field is updated.	RW	0
22	RESERVED	Reserved	RW W0Only	0
21:20	V13	PAT view for initiator 13	RW	0x0

Bits	Field Name	Description	Type	Reset
19	W12	Write-enable for V12 bit field Write 0x0: V12 field is unchanged. Write 0x1: V12 field is updated.	RW	0
18	RESERVED	Reserved	RW W0Only	0
17:16	V12	PAT view for initiator 12	RW	0x0
15	W11	Write-enable for V11 bit field Write 0x0: V11 field is unchanged. Write 0x1: V11 field is updated.	RW	0
14	RESERVED	Reserved	RW W0Only	0
13:12	V11	PAT view for initiator 11	RW	0x0
11	W10	Write-enable for V10 bit field Write 0x0: V10 field is unchanged. Write 0x1: V10 field is updated.	RW	0
10	RESERVED	Reserved	RW W0Only	0
9:8	V10	PAT view for initiator 10	RW	0x0
7	W9	Write-enable for V9 bit field Write 0x0: V9 field is unchanged. Write 0x1: V9 field is updated.	RW	0
6	RESERVED	Reserved	RW W0Only	0
5:4	V9	PAT view for initiator 9	RW	0x0
3	W8	Write-enable for V8 bit field Write 0x0: V8 field is unchanged. Write 0x1: V8 field is updated.	RW	0
2	RESERVED	Reserved	RW W0Only	0
1:0	V8	PAT view for initiator 8	RW	0x0

Table 15-51. DMM_PAT_VIEW_MAP_i

Address Offset	0x0000 0440 + (0x4 * i)	**Index**	i = 0 to 3
Physical Address	0x4E00 0440 + (0x4 * i)	**Instance**	DMM
Description	PAT view mapping register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_PAGE	RESERVED				CONT_PAGE			ACCESS_32	RESERVED				CONT_32			ACCESS_16	RESERVED				CONT_16			ACCESS_8	RESERVED				CONT_8		

Bits	Field Name	Description	Type	Reset
31	ACCESS_PAGE	Kind of access for this page mode container in view mapping i 0x0: Direct access, container base address given in CONT_PAGE 0x1: indirect access through the LUT indexed by CONT_PAGE	RW	0
30:27	RESERVED	Reserved	RW W0Only	0x0
26:24	CONT_PAGE	Container for page mode in view mapping i	RW	0x0
23	ACCESS_32	Kind of access for this 32-bit mode container in view mapping i 0x0: Direct access, container base address given in CONT_32 0x1: indirect access through the LUT indexed by CONT_32	RW	0
22:19	RESERVED	Reserved	RW W0Only	0x0
18:16	CONT_32	Container for 32-bit mode in view mapping i	RW	0x0
15	ACCESS_16	Kind of access for this 16-bit mode container in view mapping i 0x0: Direct access, container base address given in CONT_16 0x1: indirect access through the LUT indexed by CONT_16	RW	0
14:11	RESERVED	Reserved	RW W0Only	0x0
10:8	CONT_16	Container for 16-bit mode in view mapping i	RW	0x0
7	ACCESS_8	Kind of access for this 8-bit mode container in view mapping i 0x0: Direct access, container base address given in CONT_8 0x1: indirect access through the LUT indexed by CONT_8	RW	0
6:3	RESERVED	Reserved	RW W0Only	0x0
2:0	CONT_8	Container for 8-bit mode in view mapping i	RW	0x0

Table 15-52. DMM_PAT_VIEW_MAP_BASE

Address Offset	0x0000 0460		
Physical Address	0x4E00 0460	**Instance**	DMM
Description	Base address of all view mappings		
Type	RW		

31	30:0
BASE_ADDR	RESERVED

Bits	Field Name	Description	Type	Reset
31	BASE_ADDR	MSB of the PAT view mapping base address	RW	0

Bits	Field Name	Description	Type	Reset
30:0	RESERVED	Reserved	RW W0Only	0x0000 0000

Table 15-53. DMM_PAT_IRQ_EOI

Address Offset	0x0000 0478		
Physical Address	0x4E00 0478	**Instance**	DMM
Description	PAT end of interrupt		
Type	RW		

31–1	0
RESERVED	EOI

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI	End of PAT interrupt 0x0: Acknowledge the PAT interrupts	RW	0x0

Table 15-54. DMM_PAT_IRQSTATUS_RAW

Address Offset	0x0000 0480		
Physical Address	0x4E00 0480	**Instance**	DMM
Description	Per-event raw interrupt status vector. Raw status is set even if the related event is not enabled. Write 1 to set the (raw) status, mostly for debug. n = 0 for the first interrupt status raw register, n = 1 for the second interrupt status raw register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR_LUT_MISS3	ERR_UPD_DATA3	ERR_UPD_CTRL3	ERR_UPD_AREA3	ERR_INV_DATA3	ERR_INV_DSC3	FILL_LST3	FILL_DSC3	ERR_LUT_MISS2	ERR_UPD_DATA2	ERR_UPD_CTRL2	ERR_UPD_AREA2	ERR_INV_DATA2	ERR_INV_DSC2	FILL_LST2	FILL_DSC2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0

Bits	Field Name	Description	Type	Reset
31	ERR_LUT_MISS3	Access to a yet-to-be-refilled area event in area 4.n+3 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
30	ERR_UPD_DATA3	Data register update whilst refilling error event in area 4.n+3 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
29	ERR_UPD_CTRL3	Control register update whilst refilling error event in area 4.n+3 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
28	ERR_UPD_AREA3	Area register update whilst refilling error event in area 4.n+3 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
27	ERR_INV_DATA3	Invalid entry-table pointer error event in area 4.n+3 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event.	RW W1toSet	0
26	ERR_INV_DSC3	Invalid descriptor pointer error event in area 4.n+3 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
25	FILL_LST3	End of refill event for the last descriptor in area 4.n+3 Write 0x0: Keep area 3 refill done event. Write 0x1: Set area 3 refill done event. Read 0x1: Area 3 is refilled. Read 0x0: Area 3 is yet-to-be refilled.	RW W1toSet	0
24	FILL_DSC3	End of refill event for any descriptor in area 4.n+3 Write 0x0: Keep area 3 refill done event. Write 0x1: Set area 3 refill done event. Read 0x1: Area 3 is refilled. Read 0x0: Area 3 is yet-to-be refilled.	RW W1toSet	0
23	ERR_LUT_MISS2	Access to a yet-to-be-refilled area event in area 4.n+2 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
22	ERR_UPD_DATA2	Data register update whilst refilling error event in area 4.n+2 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
21	ERR_UPD_CTRL2	Control register update whilst refilling error event in area 4.n+2 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
20	ERR_UPD_AREA2	Area register update whilst refilling error event in area 4.n+2 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
19	ERR_INV_DATA2	Invalid entry-table pointer error event in area 4.n+2 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
18	ERR_INV_DSC2	Invalid descriptor pointer error event in area 4.n+2 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
17	FILL_LST2	End of refill event for the last descriptor in area 4.n+2 Write 0x0: Keep area 2 refill done event. Write 0x1: Set area 2 refill done event. Read 0x1: Area 2 is refilled. Read 0x0: Area 2 is yet-to-be refilled.	RW W1toSet	0
16	FILL_DSC2	End of refill event for any descriptor in area 4.n+2 Write 0x0: Keep area 2 refill done event. Write 0x1: Set area 2 refill done event. Read 0x1: Area 2 is refilled. Read 0x0: Area 2 is yet-to-be refilled.	RW W1toSet	0
15	ERR_LUT_MISS1	Access to a yet-to-be-refilled area event in area 4.n+1 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
14	ERR_UPD_DATA1	Data register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
13	ERR_UPD_CTRL1	Control register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
12	ERR_UPD_AREA1	Area register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
11	ERR_INV_DATA1	Invalid entry-table pointer error event in area 4.n+1 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
10	ERR_INV_DSC1	Invalid descriptor pointer error event in area 4.n+1 Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
9	FILL_LST1	End of refill event for the last descriptor in area 4.n+1 Write 0x0: Keep area 1 refill done event. Write 0x1: Set area 1 refill done event. Read 0x1: Area 1 is refilled. Read 0x0: Area 1 is yet-to-be refilled.	RW W1toSet	0
8	FILL_DSC1	End of refill event for any descriptor in area 4.n+1 Write 0x0: Keep area 1 refill done event. Write 0x1: Set area 1 refill done event. Read 0x1: Area 1 is refilled. Read 0x0: Area 1 is yet-to-be refilled.	RW W1toSet	0
7	ERR_LUT_MISS0	Access to a yet-to-be-refilled area event in area 4.n Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
6	ERR_UPD_DATA0	Data register update whilst refilling error event in area 4.n Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
5	ERR_UPD_CTRL0	Control register update whilst refilling error event in area 4.n Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
4	ERR_UPD_AREA0	Area register update whilst refilling error event in area 4.n Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
3	ERR_INV_DATA0	Invalid entry-table pointer error event in area 4.n Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
2	ERR_INV_DSC0	Invalid descriptor pointer error event in area 4.n Write 0x0: Keep current error event. Write 0x1: Set error event. Read 0x1: Error event happened. Read 0x0: No such error event	RW W1toSet	0
1	FILL_LST0	End of refill event for the last descriptor in area 4.n Write 0x0: Keep area 0 refill done event. Write 0x1: Set area 0 refill done event. Read 0x1: Area 0 is refilled. Read 0x0: Area 0 is yet-to-be refilled.	RW W1toSet	0
0	FILL_DSC0	End of refill event for any descriptor in area 4.n Write 0x0: Keep area 0 refill done event. Write 0x1: Set area 0 refill done event. Read 0x1: Area 0 is refilled. Read 0x0: Area 0 is yet-to-be refilled.	RW W1toSet	0

Table 15-55. DMM_PAT_IRQSTATUS

Address Offset	0x0000 0490		
Physical Address	0x4E00 0490	**Instance**	DMM
Description	Per-event "enabled" interrupt status vector. Enabled status is not set unless the event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). n = 0 for the first interrupt status register, n = 1 for the second interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR_LUT_MISS3	ERR_UPD_DATA3	ERR_UPD_CTRL3	ERR_UPD_AREA3	ERR_INV_DATA3	ERR_INV_DSC3	FILL_LST3	FILL_DSC3	ERR_LUT_MISS2	ERR_UPD_DATA2	ERR_UPD_CTRL2	ERR_UPD_AREA2	ERR_INV_DATA2	ERR_INV_DSC2	FILL_LST2	FILL_DSC2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0

Bits	Field Name	Description	Type	Reset
31	ERR_LUT_MISS3	Access to a yet-to-be-refilled area event in area 4.n+3 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
30	ERR_UPD_DATA3	Data register update whilst refilling error event in area 4.n+3 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
29	ERR_UPD_CTRL3	Control register update whilst refilling error event in area 4.n+3 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
28	ERR_UPD_AREA3	Area register update whilst refilling error event in area 4.n+3 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
27	ERR_INV_DATA3	Invalid entry-table pointer error event in area 4.n+3 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
26	ERR_INV_DSC3	Invalid descriptor pointer error event in area 4.n+3 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
25	FILL_LST3	End of refill event for the last descriptor in area 4.n+3 Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0
24	FILL_DSC3	End of refill event for any descriptor in area 4.n+3 Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0
23	ERR_LUT_MISS2	Access to a yet-to-be-refilled area event in area 4.n+2 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
22	ERR_UPD_DATA2	Data register update whilst refilling error event in area 2 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
21	ERR_UPD_CTRL2	Control register update whilst refilling error event in area 4.n+2 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
20	ERR_UPD_AREA2	Area register update whilst refilling error event in area 4.n+2 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
19	ERR_INV_DATA2	Invalid entry-table pointer error event in area 4.n+2 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
18	ERR_INV_DSC2	Invalid descriptor pointer error event in area 4.n+2 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
17	FILL_LST2	End of refill event for the last descriptor in area 4.n+2 Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0
16	FILL_DSC2	End of refill event for any descriptor in area 4.n+2 Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0
15	ERR_LUT_MISS1	Access to a yet-to-be-refilled area event in area 4.n+1 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
14	ERR_UPD_DATA1	Data register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
13	ERR_UPD_CTRL1	Control register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
12	ERR_UPD_AREA1	Area register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
11	ERR_INV_DATA1	Invalid entry-table pointer error event in area 4.n+1 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked	RW W1toClr	0
10	ERR_INV_DSC1	Invalid descriptor pointer error event in area 4.n+1 Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked	RW W1toClr	0
9	FILL_LST1	End of refill event for the last descriptor in area 4.n+1 Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0
8	FILL_DSC1	End of refill event for any descriptor in area 4.n+1 Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0
7	ERR_LUT_MISS0	Access to a yet-to-be-refilled area event in area 4.n Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
6	ERR_UPD_DATA0	Data register update whilst refilling error event in area 4.n Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
5	ERR_UPD_CTRL0	Control register update whilst refilling error event in area 4.n Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
4	ERR_UPD_AREA0	Area register update whilst refilling error event in area 4.n Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
3	ERR_INV_DATA0	Invalid entry-table pointer error event in area 4.n Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
2	ERR_INV_DSC0	Invalid descriptor pointer error event in area 4.n Write 0x0: Keep current maskable error event. Write 0x1: Clear this maskable error event. Read 0x1: Error event happened. Read 0x0: No such error event or this event is masked.	RW W1toClr	0
1	FILL_LST0	End of refill event for the last descriptor in area 4.n Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0
0	FILL_DSC0	End of refill event for any descriptor in area 4.n Write 0x0: Keep current area refill done maskable event. Write 0x1: Clear current area refill done maskable event. Read 0x1: Current area is refilled. Read 0x0: Current area is yet-to-be refilled or this event is masked.	RW W1toClr	0

Table 15-56. DMM_PAT_IRQENABLE_SET

Address Offset	0x0000 04A0		
Physical Address	0x4E00 04A0	**Instance**	DMM
Description	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register. n = 0 for the first interrupt enable set register, n = 1 for the second interrupt enable set register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ERR_LUT_MISS3	ERR_UPD_DATA3	ERR_UPD_CTRL3	ERR_UPD_AREA3	ERR_INV_DATA3	ERR_INV_DSC3	FILL_LST3	FILL_DSC3	ERR_LUT_MISS2	ERR_UPD_DATA2	ERR_UPD_CTRL2	ERR_UPD_AREA2	ERR_INV_DATA2	ERR_INV_DSC2	FILL_LST2	FILL_DSC2	ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0

Bits	Field Name	Description	Type	Reset
31	ERR_LUT_MISS3	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
30	ERR_UPD_DATA3	Unexpected data register update whilst refilling interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
29	ERR_UPD_CTRL3	Unexpected control register update whilst refilling interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
28	ERR_UPD_AREA3	Unexpected area register update whilst refilling interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
27	ERR_INV_DATA3	Invalid entry-table pointer interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
26	ERR_INV_DSC3	Invalid descriptor pointer interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
25	FILL_LST3	End of refill interrupt source mask for the last descriptior in area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
24	FILL_DSC3	End of refill interrupt source mask for any descriptior in area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
23	ERR_LUT_MISS2	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
22	ERR_UPD_DATA2	Unexpected data register update whilst refilling interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
21	ERR_UPD_CTRL2	Unexpected control register update whilst refilling interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
20	ERR_UPD_AREA2	Unexpected area register update whilst refilling interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
19	ERR_INV_DATA2	Invalid entry-table pointer interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
18	ERR_INV_DSC2	Invalid descriptor pointer interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
17	FILL_LST2	End of refill interrupt source mask for the last descriptior in area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
16	FILL_DSC2	End of refill interrupt source mask for any descriptior in area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
15	ERR_LUT_MISS1	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
14	ERR_UPD_DATA1	Unexpected data register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
13	ERR_UPD_CTRL1	Unexpected control register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
12	ERR_UPD_AREA1	Unexpected area register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
11	ERR_INV_DATA1	Invalid entry-table pointer interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
10	ERR_INV_DSC1	Invalid descriptor pointer interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
9	FILL_LST1	End of refill interrupt source mask for the last descriptior in area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
8	FILL_DSC1	End of refill interrupt source mask for any descriptior in area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
7	ERR_LUT_MISS0	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
6	ERR_UPD_DATA0	Unexpected data register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
5	ERR_UPD_CTRL0	Unexpected control register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
4	ERR_UPD_AREA0	Unexpected area register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
3	ERR_INV_DATA0	Invalid entry-table pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
2	ERR_INV_DSC0	Invalid descriptor pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0
1	FILL_LST0	End of refill interrupt source mask for the last descriptior in area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
0	FILL_DSC0	End of refill interrupt source mask for any descriptior in area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Enable (unmask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toSet	0

Table 15-57. DMM_PAT_IRQENABLE_CLR

Address Offset	0x0000 04B0
Physical Address	0x4E00 04B0 — **Instance** DMM
Description	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register. n = 0 for the first interrupt enable clear register, n = 1 for the second interrupt enable clear register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR_LUT_MISS3	ERR_UPD_DATA3	ERR_UPD_CTRL3	ERR_UPD_AREA3	ERR_INV_DATA3	ERR_INV_DSC3	FILL_LST3	FILL_DSC3	ERR_LUT_MISS2	ERR_UPD_DATA2	ERR_UPD_CTRL2	ERR_UPD_AREA2	ERR_INV_DATA2	ERR_INV_DSC2	FILL_LST2	FILL_DSC2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0

Bits	Field Name	Description	Type	Reset
31	ERR_LUT_MISS3	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
30	ERR_UPD_DATA3	Unexpected data register update whilst refilling interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
29	ERR_UPD_CTRL3	Unexpected control register update whilst refilling interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
28	ERR_UPD_AREA3	Unexpected area register update whilst refilling interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
27	ERR_INV_DATA3	Invalid entry-table pointer interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
26	ERR_INV_DSC3	Invalid descriptor pointer interrupt source mask for area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
25	FILL_LST3	End of refill interrupt source mask for the last descriptior in area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked).. Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
24	FILL_DSC3	End of refill interrupt source mask for any descriptior in area 4.n+3 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
23	ERR_LUT_MISS2	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
22	ERR_UPD_DATA2	Unexpected data register update whilst refilling interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
21	ERR_UPD_CTRL2	Unexpected control register update whilst refilling interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
20	ERR_UPD_AREA2	Unexpected area register update whilst refilling interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
19	ERR_INV_DATA2	Invalid entry-table pointer interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
18	ERR_INV_DSC2	Invalid descriptor pointer interrupt source mask for area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
17	FILL_LST2	End of refill interrupt source mask for the last descriptior in area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
16	FILL_DSC2	End of refill interrupt source mask for any descriptior in area 4.n+2 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
15	ERR_LUT_MISS1	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
14	ERR_UPD_DATA1	Unexpected data register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
13	ERR_UPD_CTRL1	Unexpected control register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
12	ERR_UPD_AREA1	Unexpected area register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
11	ERR_INV_DATA1	Invalid entry-table pointer interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
10	ERR_INV_DSC1	Invalid descriptor pointer interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
9	FILL_LST1	End of refill interrupt source mask for the last descriptior in area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
8	FILL_DSC1	End of refill interrupt source mask for any descriptior in area 4.n+1 Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
7	ERR_LUT_MISS0	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
6	ERR_UPD_DATA0	Unexpected data register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
5	ERR_UPD_CTRL0	Unexpected control register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
4	ERR_UPD_AREA0	Unexpected area register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
3	ERR_INV_DATA0	Invalid entry-table pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
2	ERR_INV_DSC0	Invalid descriptor pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
1	FILL_LST0	End of refill interrupt source mask for the last descriptior in area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0
0	FILL_DSC0	End of refill interrupt source mask for any descriptior in area 4.n Write 0x0: Keep current mask of this interrupt source. Write 0x1: Disable (mask) this interrupt source. Read 0x1: This interrupt source is enabled (unmasked). Read 0x0: This interrupt source is disabled (masked).	RW W1toClr	0

Table 15-58. DMM_PAT_STATUS_i

Address Offset	0x0000 04C0 + (0x4 * i)	**Index**	i = 0 to 3
Physical Address	0x4E00 04C0 + (0x4 * i)	**Instance**	DMM
Description	Status register for each refill engine n = 0 for the first engine status register, n = 1 for the second engine status register.		
Type	R		

31:25	24:16	15:10	9:8	7	6:5	4	3	2	1	0
RESERVED	CNT	ERROR	RESERVED	BYPASSED	RESERVED	LINKED	DONE	RUN	VALID	READY

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x00
24:16	CNT	Counter of remaining lines to reload for engine n	R	0x000

Bits	Field Name	Description	Type	Reset
15:10	ERROR	Error happened in engine n Read 0x0: No error Read 0x1: Invalid descriptor provided Read 0x2: Invalid data pointer provided Read 0x4: Unexpected area register update while refilling Read 0x8: Unexpected control register update while refilling Read 0x10: Unexpected data register update while refilling Read 0x20: Unexpected access to a yet-to-be-refilled location	R	0x00
9:8	RESERVED	Reserved	R	0x0
7	BYPASSED	Engine n is bypassed. Direct access to the LUT is provided.	R	0
6:5	RESERVED	Reserved	R	0x0
4	LINKED	Area reconfiguration link asserted for engine n	R	0
3	DONE	Area reloading finished for engine n	R	0
2	RUN	Area currently reloading for engine n	R	0
1	VALID	Valid area description for engine n	R	0
0	READY	Area registers ready for engine n	R	1

Table 15-59. DMM_PAT_DESCR_i

Address Offset	0x0000 0500 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x4E00 0500 + (0x10 * i)	**Instance**	DMM
Description	Physical address of the next table refill descriptor n = 0 for the descriptor register of the first engine, n = 1 for the descriptor register of the second engine. Writing to this register aborts the current ongoing area reload and resets the corresponding DMM_PAT_AREA_i, DMM_PAT_CTRL_i and DMM_PAT_DATA_i registers.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																												RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Physical address of the next table refill descriptor of engine n	RW WtoClr	0x0000000
3:0	RESERVED	Reserved	RW W0Only	0x0

Table 15-60. DMM_PAT_AREA_i

Address Offset	0x0000 0504 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x4E00 0504 + (0x10 * i)	**Instance**	DMM
Description	Area definition for DMM physical address translator n = 0 for the area register of the first engine, n = 1 for the area register of the second engine.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y1								X1								Y0								X0							

Bits	Field Name	Description	Type	Reset
31:24	Y1	Y-coordinate of the bottom-right corner of the PAT area for engine n	RW	0x00

Bits	Field Name	Description	Type	Reset
23:16	X1	X-coordinate of the bottom-right corner of the PAT area for engine n	RW	0x00
15:8	Y0	Y-coordinate of the top-left corner of the PAT area for engine n	RW	0x00
7:0	X0	X-coordinate of the top-left corner of the PAT area for engine n	RW	0x00

Table 15-61. DMM_PAT_CTRL_i

Address Offset	0x0000 0508 + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x4E00 0508 + (0x10 * i)	**Instance**	DMM
Description	DMM physical address translator control register n = 0 for the control register of the first engine, n = 1 for the control register of the second engine.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITIATOR				RESERVED											SYNC	RESERVED									DIRECTION			RESERVED			START

Bits	Field Name	Description	Type	Reset
31:28	INITIATOR	DMM PAT initiator for synchronization in engine n	RW	0x0
27:17	RESERVED	Reserved	RW W0Only	0x000
16	SYNC	DMM PAT table reload synchronization for engine n 0x0: Not synchronized 0x1: Synchronized	RW	0
15:7	RESERVED	Reserved	RW W0Only	0x000
6:4	DIRECTION	Direction of this PAT table refill for engine n	RW	0x0
3:1	RESERVED	Reserved	RW W0Only	0x0
0	START	Starting a PAT table refill with engine n	RW	0

Table 15-62. DMM_PAT_DATA_i

Address Offset	0x0000 050C + (0x10 * i)	**Index**	i = 0 to 3
Physical Address	0x4E00 050C + (0x10 * i)	**Instance**	DMM
Description	Physical address of the current table refill entry data n = 0 for the data register of the first engine, n = 1 for the data register of the second engine.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																												RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Physical address of the current table refill entry data or single actual entry data when in manual mode for engine n	RW	0x0000000
3:0	RESERVED	Reserved	RW W0Only	0x0

Table 15-63. DMM_PEG_HWINFO

Address Offset	0x0000 0608		
Physical Address	0x4E00 0608	**Instance**	DMM

Table 15-63. DMM_PEG_HWINFO (continued)

Description	DMM hardware configuration for PEG
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									PRIO_CNT						

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	PRIO_CNT	Number of PEG priority entries Read 0x1: One priority entry Read 0x2: Two priority entries Read 0x4: Four priority entries Read 0x8: Eight priority entries Read 0x10: Sixteen priority entries Read 0x20: Thirty-two priority entries Read 0x40: Sixty-four priority entries	R	0x40

Table 15-64. DMM_PEG_PRIO_k

Address Offset	0x0000 0620 + (0x4 * k)	**Index**	k = 0 to 7
Physical Address	0x4E00 0620 + (0x4 * k)	**Instance**	DMM
Description	DMM PEG Priority register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W7	P7			W6	P6			W5	P5			W4	P4			W3	P3			W2	P2			W1	P1			W0	P0		

Bits	Field Name	Description	Type	Reset
31	W7	Write-enable for P7 bit field Write 0x0: P7 field is unchanged. Write 0x1: P7 field is updated.	RW	0
30:28	P7	Priority for initiator ConnID = 8 × k + 7	RW	0x4
27	W6	Write-enable for P6 bit field Write 0x0: P6 field is unchanged. Write 0x1: P6 field is updated.	RW	0
26:24	P6	Priority for initiator ConnID = 8 × k + 6	RW	0x4
23	W5	Write-enable for P5 bit field Write 0x0: P5 field is unchanged. Write 0x1: P5 field is updated.	RW	0
22:20	P5	Priority for initiator ConnID = 8 × k + 5	RW	0x4
19	W4	Write-enable for P4 bit field Write 0x0: P4 field is unchanged. Write 0x1: P4 field is updated.	RW	0
18:16	P4	Priority for initiator ConnID = 8 × k + 4	RW	0x4
15	W3	Write-enable for P3 bit field Write 0x0: P3 field is unchanged. Write 0x1: P3 field is updated.	RW	0
14:12	P3	Priority for initiator ConnID = 8 × k + 3	RW	0x4

Bits	Field Name	Description	Type	Reset
11	W2	Write-enable for P2 bit field Write 0x0: P2 field is unchanged. Write 0x1: P2 field is updated.	RW	0
10:8	P2	Priority for initiator ConnID = 8 × k + 2	RW	0x4
7	W1	Write-enable for P1 bit field Write 0x0: P1 field is unchanged. Write 0x1: P1 field is updated.	RW	0
6:4	P1	Priority for initiator ConnID = 8 × k + 1	RW	0x4
3	W0	Write-enable for P0 bit field Write 0x0: P0 field is unchanged. Write 0x1: P0 field is updated.	RW	0
2:0	P0	Priority for initiator ConnID = 8 × k	RW	0x4

Table 15-65. DMM_PEG_PRIO_PAT

Address Offset	0x0000 0640		
Physical Address	0x4E00 0640	**Instance**	DMM
Description	DMM PEG priority register for the internal PAT engine.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												W_PAT	P_PAT		

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW W0Only	0x0000000
3	W_PAT	Write-enable for P_PAT bit field Write 0x0: P_PAT field is updated. Write 0x1: P_PAT field is unchanged.	RW	0
2:0	P_PAT	Priority for PAT engine	RW	0x4

15.3 EMIF Controller

This section describes mainly the features and functions of the two external memory interface (EMIF) controllers and also their associated PHYs.

15.3.1 EMIF Controller Overview

The EMIF controller provides connectivity between the device and DDR2 type or DDR3 type of memories and manages data bus read/write accesses between external memories and the device subsystems which have access to the L3_MAIN interconnect and DMA capability too.

The EMIF features are introduced in Section 15.1.3 *EMIF Overview* of Section 15.1 *Memory Subsystem Overview*.

The device includes two EMIF controllers. Each controller supports up to 2 GiB of SDRAM over one chip select. As a result the total physical available SDRAM space is 4 GiB. Figure 15-45 shows an overview of these EMIF controllers and also the connections to the other surrounding modules. As can be seen on Figure 15-45 the two EMIFs are not directly available on device pads. That is, they are not directly connected to the external SDRAMs. There are DDR PHYs and then DDR I/Os between each EMIF controller and external SDRAM. The EMIF controller, the DDR PHYs and the DDR I/Os work like a single unit to manage data exchanges to and from external memories. To achieve successful data transaction between an internal device initiator and external SDRAM all these three components must be used.

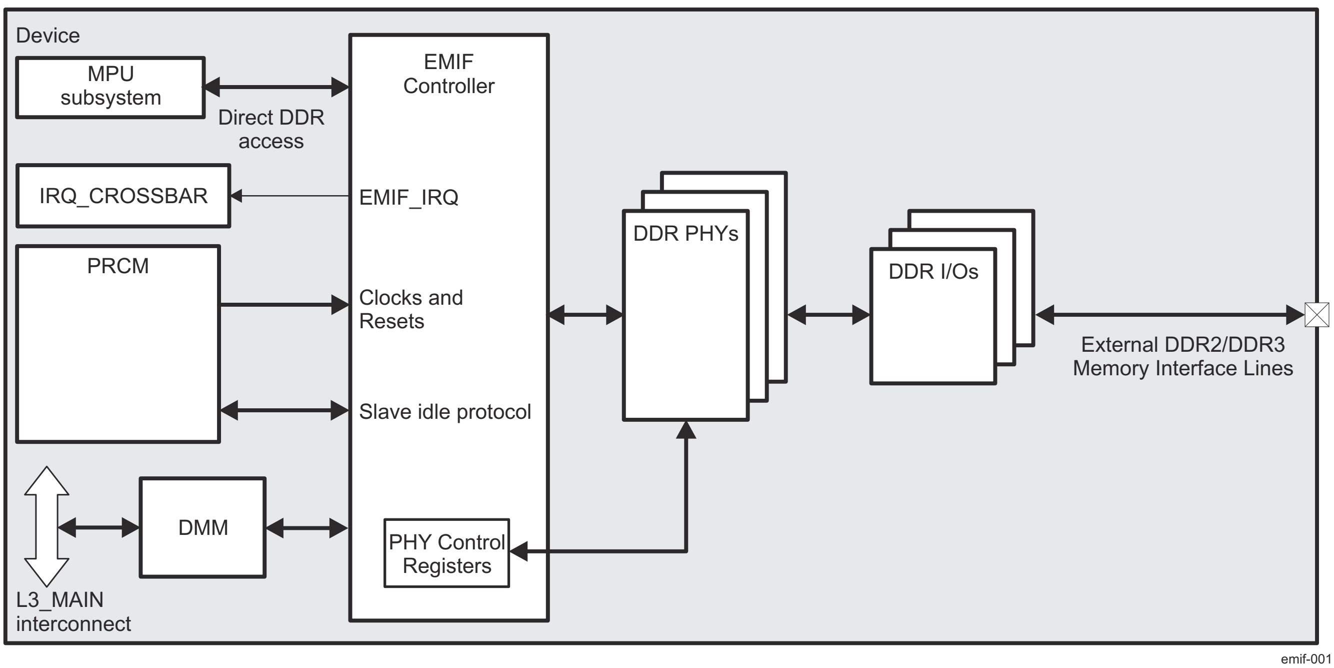


Figure 15-45. EMIF Controller Overview

15.3.2 EMIF Module Environment

This section describes the external connections of the two EMIF modules.

Figure 15-46 shows an example EMIF DDR2/DDR3 configuration without ECC memory connected.

Figure 15-47 shows an example EMIF DDR2/DDR3 configuration with ECC memory connected.

For simplification the DDR PHYs and DDR I/Os are not shown. Only the I/O signals and their corresponding EMIF pins are shown.

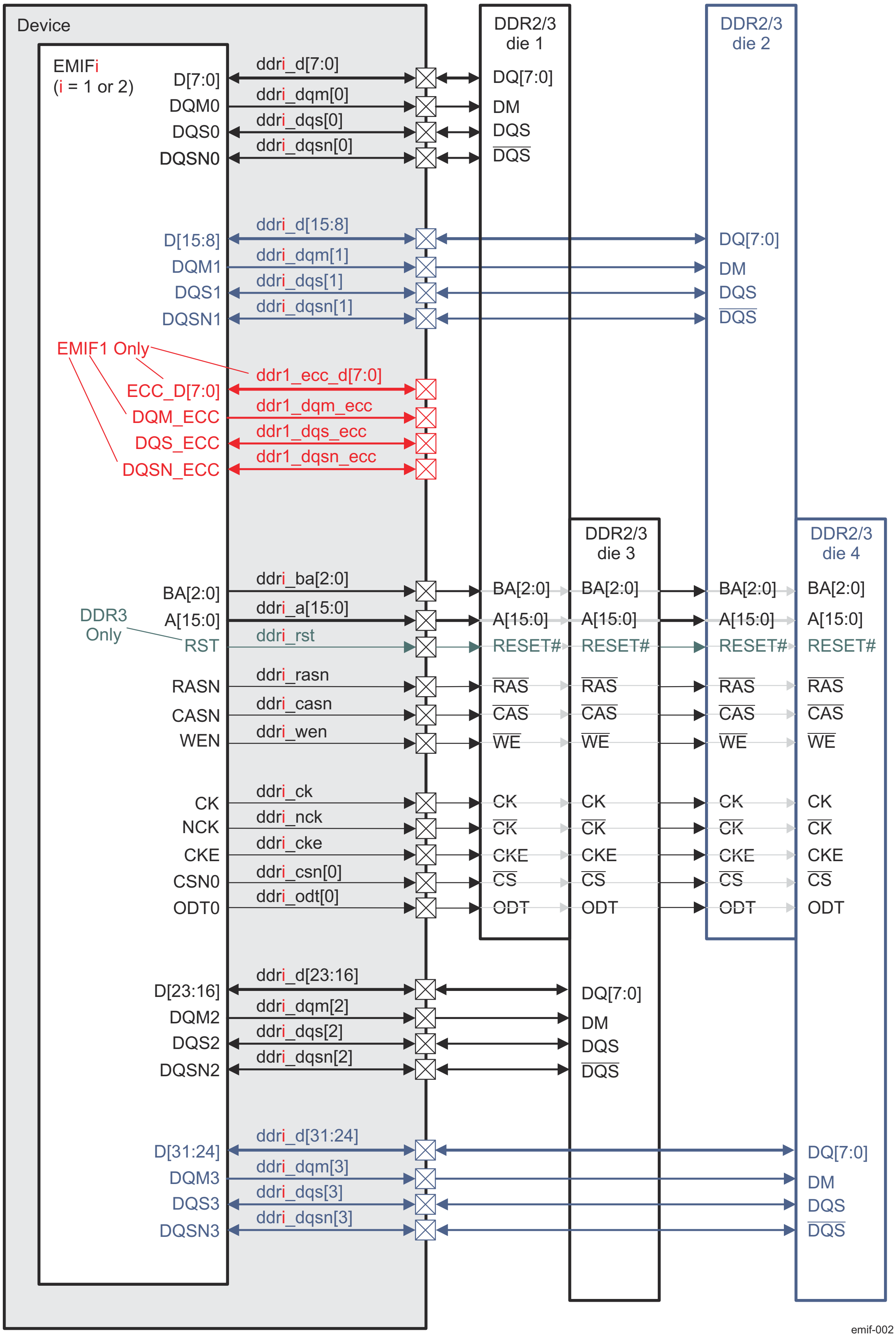


Figure 15-46. EMIF DDR2/DDR3 Configuration Without ECC

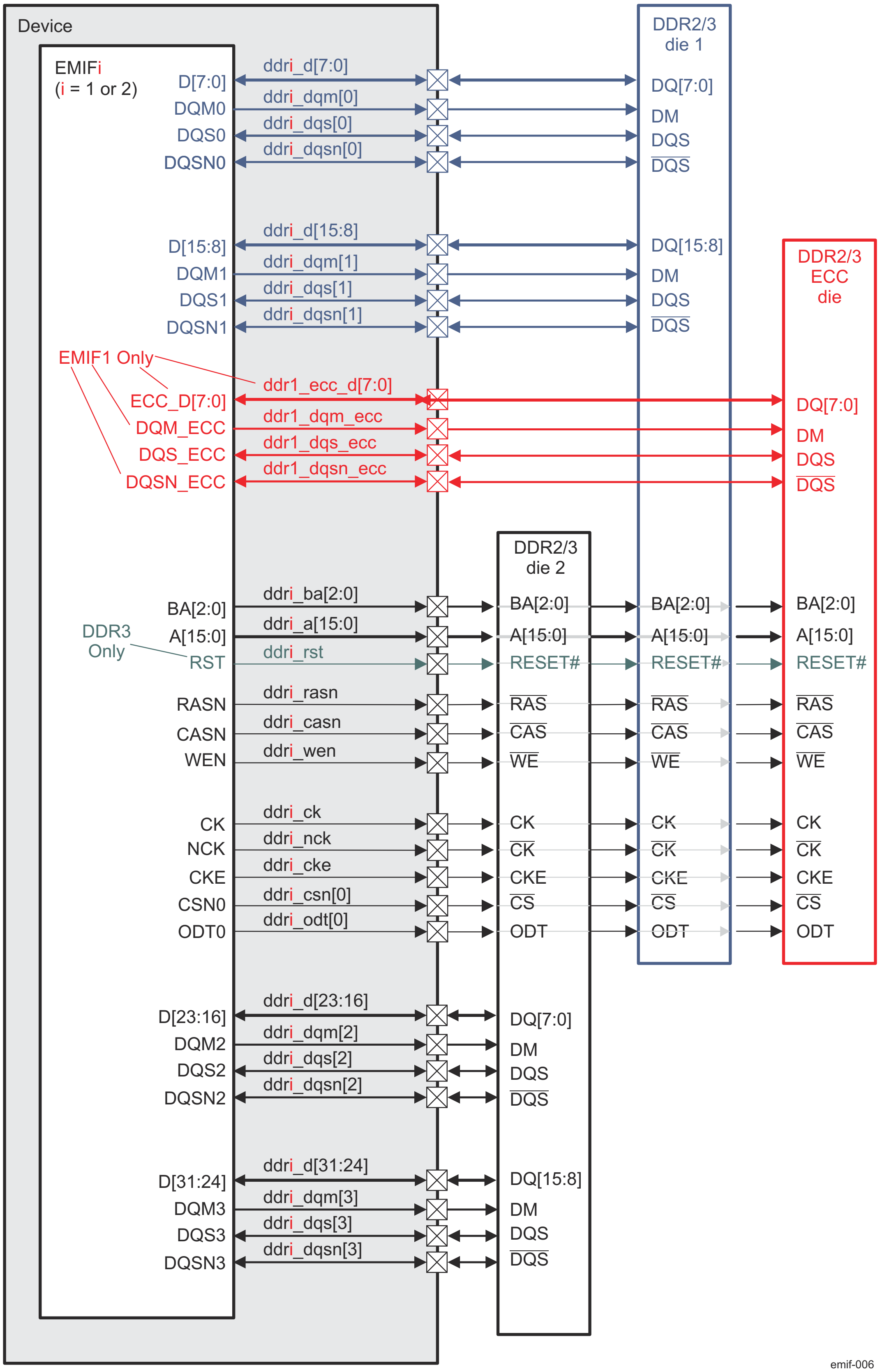


Figure 15-47. EMIF DDR2/DDR3 Configuration With ECC

Table 15-66 describes the EMIF module associated I/O signals used for connection to DDR2 and DDR3 memory types.

Table 15-66. EMIF Module I/O signals

EMIF Pin Name	Device DDR2 and DDR3 I/O Signal Names	I/O(1)	Description
		EMIF1 Data PHYs	
D[31:0]	ddr1_d[31:0]	I/O	Data bus
DQM[3:0]	ddr1_dqm[3:0]	O	Data mask
DQS[3:0]	ddr1_dqs[3:0]	I/O	Data strobe
DQSN[3:0]	ddr1_dqsn[3:0]	I/O	Data strobe invert
ECC_D	ddr1_ecc_d[7:0]	I/O	Data bus used for ECC (EMIF1 Only)
DQM_ECC	ddr1_dqm_ecc	O	Data mask used for ECC (EMIF1 Only)
DQS_ECC	ddr1_dqs_ecc	I/O	Data strobe used for ECC (EMIF1 Only)
DQSN_ECC	ddr1_dqsn_ecc	I/O	Data strobe invert used for ECC (EMIF1 Only)
		EMIF1 Command PHYs	
A[15:0]	ddr1_a[15:0]	O	Row/column address bus
BA[2:0]	ddr1_ba[2:0]	O	Bank select
CK	ddr1_ck	O	Differential clock
NCK	ddr1_nck	O	Differential clock
CSN[0]	ddr1_csn[0]	O	Active low rank select signal (chip select)
CKE	ddr1_cke	O	Clock enable
CASN	ddr1_casn	O	Command
RASN	ddr1_rasn	O	Command
WEN	ddr1_wen	O	Command
RST	ddr1_rst	O	Active low asynchronous reset (DDR3 only)
ODT[0]	ddr1_odt[0]	O	On-die termination enable signal
		EMIF2 Data PHYs	
D[31:0]	ddr2_d[31:0]	I/O	Data bus
DQM[3:0]	ddr2_dqm[3:0]	O	Data mask
DQS[3:0]	ddr2_dqs[3:0]	I/O	Data strobe
DQSN[3:0]	ddr2_dqsn[3:0]	I/O	Data strobe invert
		EMIF2 Command PHYs	
A[15:0]	ddr2_a[15:0]	O	Row/column address bus
BA[2:0]	ddr2_ba[2:0]	O	Bank select
CK	ddr2_ck	O	Differential clock
NCK	ddr2_nck	O	Differential clock
CSN[0]	ddr2_csn[0]	O	Active low rank select signal (chip select)
CKE	ddr2_cke	O	Clock enable
CASN	ddr2_casn	O	Command
RASN	ddr2_rasn	O	Command
WEN	ddr2_wen	O	Command
RST	ddr2_rst	O	Active low asynchronous reset (DDR3 only)
ODT[0]	ddr2_odt[0]	O	On-die termination enable signal

(1) I = Input; O = Output

The CKE memory pad is dynamically driven by the EMIF module according to the memory interface activity. It can also be forced to tri-state by a dedicated Control Module register. For more information, see Section 15.3.4.17 *Forcing CKE to tri-state*.

Note

The index numbers 1 and 2 which are part of the EMIF DDR2 and DDR3 associated signals listed in Table 15-66 *EMIF Associated I/O signals*, column "**Device DDR2 and DDR3 I/O Signal Names**" should not be confused with DDR1 and DDR2 types of memories. They are used to indicate whether the corresponding signal is associated with EMIF1 or with EMIF2 controller. For example, ddr1_csn[0] signal is the chip select associated with EMIF1 and ddr2_csn[0] is the chip select associated with EMIF2.

The DDR memory connected to the DDR ECC bus does NOT need to be the same part number as the DDR memories connected to the DDR data bus. However, some constraints do apply. When selecting a memory for the DDR ECC bus, the following restrictions must be adhered to.

Compared to the DDR memories on the data bus, the DDR ECC memory must:

- Match the same DDR type (DDR3, DDR2, etc.) and speed grade
- Have an equal number of internal banks
- Have an equal number of columns
- Have a greater or equal number of rows

In addition,

- Unused pins should be properly tied off as described in the routing guidelines of the device data manual.
- EMIF register settings should be configured to satisfy the larger minimum timing requirements and the smaller maximum timing requirements between the two different DDR memories to ensure that all DDR memories connected to the EMIF channel are running within their specified range.

Note

For a full list of supported DDR device types, frequencies, and topologies, refer to the routing guidelines of the device data manual.

15.3.3 EMIF Module Integration

This section describes the integration of the EMIF modules in the device and includes information about clocks, resets, and hardware requests.

Figure 15-48 shows the integration of the two EMIF modules in the device.

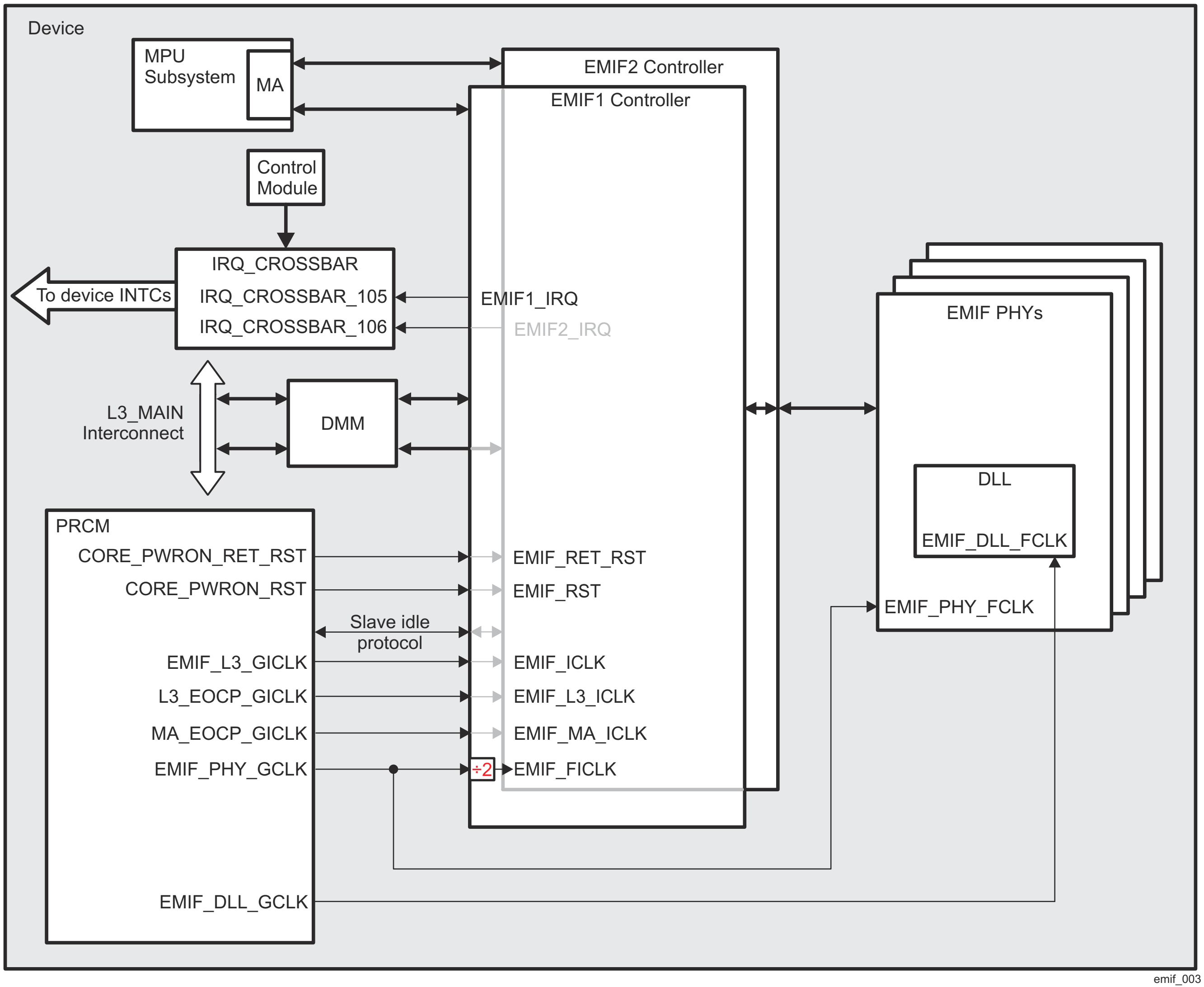


Figure 15-48. EMIF Modules Integration

Note

For more information about the slave idle protocol, see Section 3.1.1.1.3, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 15-67 through Table 15-69 summarize the integration of the EMIF modules in the device.

Table 15-67. EMIF Integration Attributes

Module Instance	Attributes		
	Power Domain	**Wake-Up Capability**	**Interconnect**

Table 15-67. EMIF Integration Attributes (continued)

EMIF1	PD_COREAON	No	EMIF1 Controller is accessible via L3_MAIN interconnect but not directly, and only through the DMM.
EMIF2	PD_COREAON	No	EMIF2 Controller is accessible via L3_MAIN interconnect but not directly, and only through the DMM.

Table 15-68. EMIF Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EMIF1/ EMIF2	EMIF_ICLK	EMIF_L3_GICLK	PRCM	Interface clock for EMIF1/EMIF2 Controller used for driving the L3 interface logic and for SDRAM Read Data FIFO.
	EMIF_L3_ICLK	L3_EOCP_GICLK	PRCM	Interface clock for EMIF1/ EMIF2 Controller which frequency is equal to EMIF_L3_GICLK interface clock. Used for command/write data pre-FIFO to Command/Write Data FIFO paths when MPU is idle.
	EMIF_MA_ICLK	MA_EOCP_GICLK	PRCM	Additional interface clock for EMIF1/ EMIF2 Controller which frequency is equal to MPU_GCLK/4. Used for command/write data pre-FIFO to Command/Write Data FIFO paths when the MPU is active.
	EMIF_PHY_FCLK	EMIF_PHY_GCLK	PRCM	Common functional clock for the EMIF1/EMIF2 associated PHYs. This clock is equal to the DDR2/ DDR3 clock rate.
	EMIF_FICLK	EMIF_PHY_GCLK/2	PRCM	Functional and interface clock for EMIF1/EMIF2 Controller. This clock runs at half the DDR2/ DDR3 clock rate.
	EMIF_DLL_FCLK	EMIF_DLL_GCLK	PRCM	Common functional clock for all DLLs associated with the EMIF1/ EMIF2 PHYs.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EMIF1/ EMIF2	EMIF_RET_RST	CORE_PWRON_RET_RST	PRCM	Power-on reset
	EMIF_RST	CORE_PWRON_RST	PRCM	Power-on reset

Note

The two clocks MA_EOCP_GICLK and L3_EOCP_GICLK are mutually exclusive. EMIF controllers are clocked by EMIF_MA_ICLK when MPU interface is active. When system interface is active, EMIF_L3_ICLK clock is used. This action is done automatically by the PRCM.

Table 15-69. EMIF Hardware Requests

Interrupt Requests				
Module Instance	**IRQ Source Name**	**IRQ_CROSSBAR Input**	**Default IRQ Source Mapping**	**Description**

Table 15-69. EMIF Hardware Requests (continued)

EMIF1	EMIF1_IRQ	IRQ_CROSSBAR_105	MPU_IRQ_110	EMIF1 interrupt request
EMIF2	EMIF2_IRQ	IRQ_CROSSBAR_106	MPU_IRQ_111	EMIF2 interrupt request

Note

The “**Default IRQ Source Mapping**” column in Table 15-69 *EMIF Hardware Requests* shows the default mapping of the IRQ sources listed in column “**IRQ Source Name**” to a certain interrupt line of one of the device interrupt controllers. These IRQ sources can also be mapped to other interrupt lines of each device interrupt controller through the IRQ_CROSSBAR module. For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*. For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

Note

For the description of the interrupt source, see Section 15.3.4.5, *Interrupt Requests*.

15.3.4 EMIF Functional Description

15.3.4.1 Block Diagram

The EMIF module provides an interface to DDR2 and DDR3 SDRAM memories.

Figure 15-49 shows the interconnection between the EMIF module and the other modules.

Digital locked loops (DLLs) are used to delay the input DQS signals during reads so that these strobe signals can be used to latch incoming data on the DQ pins, as required by the DDR standard.

Physical layers (PHYs) convert single-data rate (SDR) signals to DDR signals.

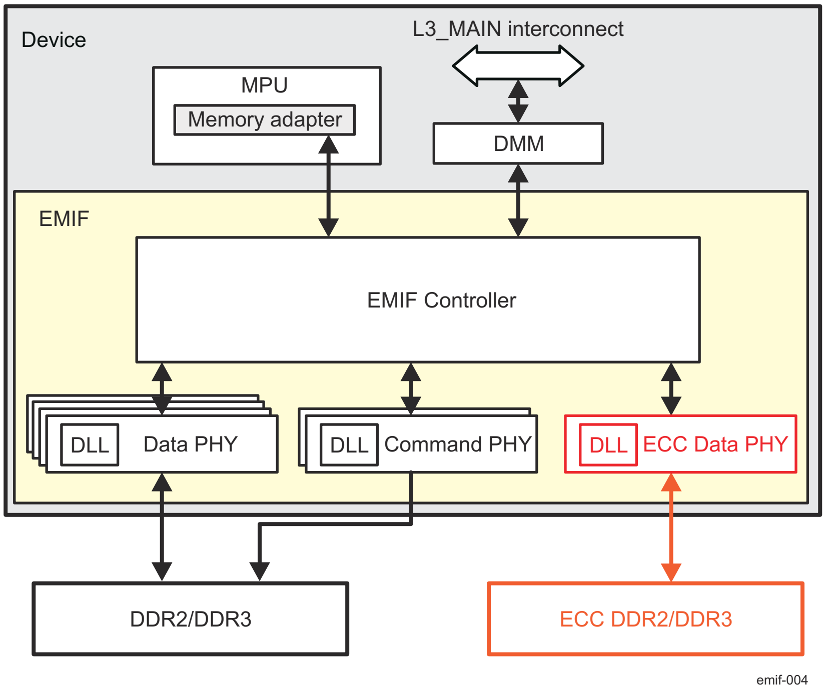


Figure 15-49. EMIF Block Diagram

15.3.4.1.1 Local Interface

The EMIF supports two local (on-chip) interfaces:

- System interface (from L3_MAIN interconnect)
- MPU interface (from MPU subsystem)

System interface is used to request all external memory device accesses, to access the EMIF registers, and to transfer all data to and from the EMIF controller. MPU interface is used to process memory accesses. Table 15-70 shows the MAddrSpace mapping.

Table 15-70. MAddrSpace Mapping

MAddrSpace(1)	Chip-Select	Description	Exclusions
0x0	CSN0	SDRAM(s)	
0x1	N/A	Reserved.	
0x2	N/A	Reserved. Any access to this area will generate an error on the L3 interface. This can be used by the software to track any unwanted access to the section defined in the DMM_LISA_MAP_i register.	Not visible thtough the MPU port
0x3	N/A	Internal registers	Not visible through the MPU port

(1) See DMM_LISA_MAP_i [17:16] SDRC_ADDRSPC register bitfield in Section 15.2, *Dynamic Memory Manager*.

15.3.4.1.2 FIFO Description

The EMIF module contains the following FIFOs:

- Command FIFO
- Write data FIFO
- Return command FIFO
- Two read data FIFOs

Figure 15-50 shows the overall architecture of the EMIF FIFOs.

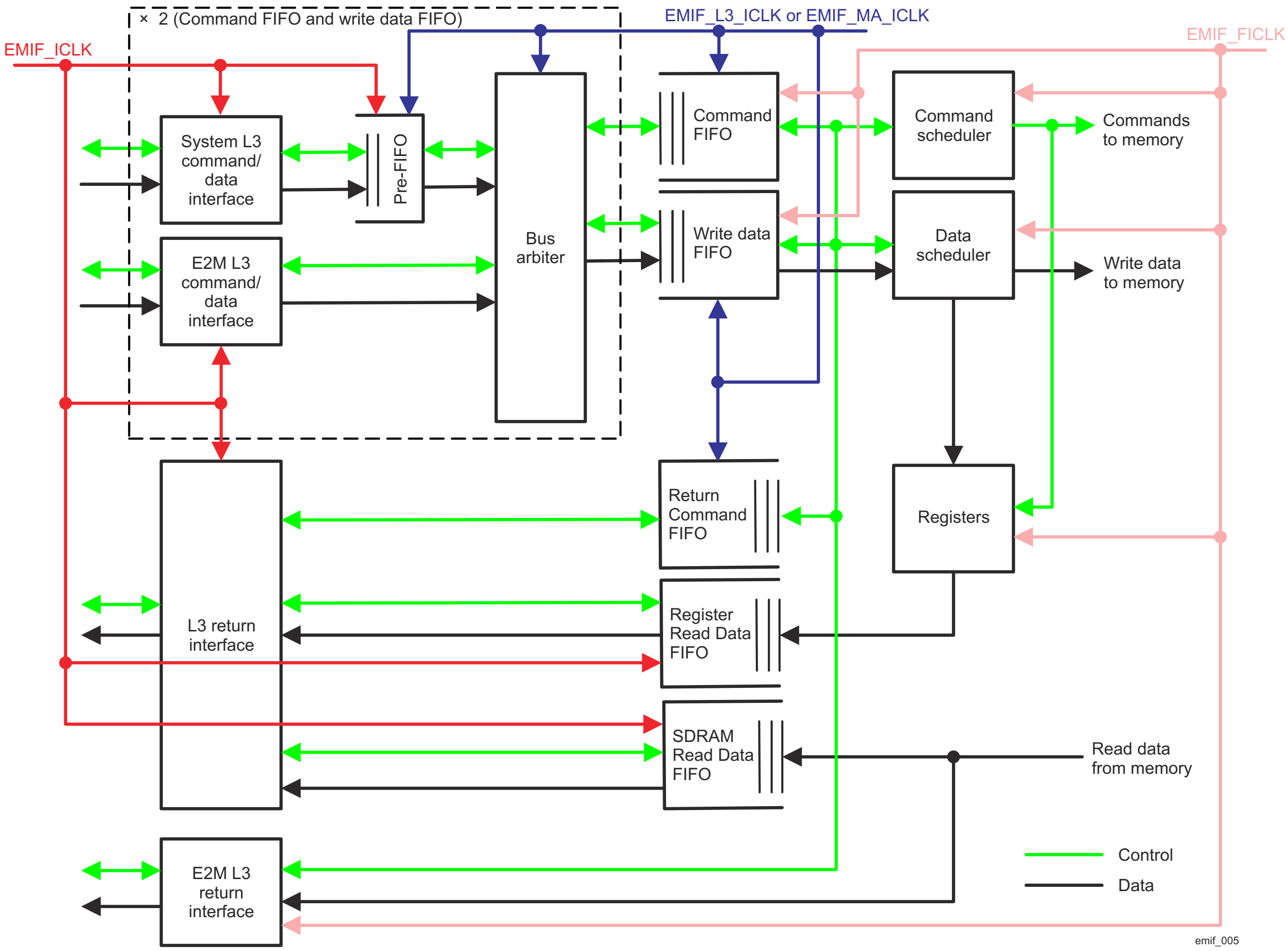


Figure 15-50. FIFO Block Diagram

Table 15-71 lists the allocation of the entries.

Table 15-71. FIFO Allocation

Parameter	System Local Interface Entries	MPU Local Interface Entries
Pre Command FIFO	6	4
Command FIFO	Programmable [(1)]	Programmable [(1)]
Pre Write FIFO	6	8
Write Data FIFO (256-bit)	Up to (19 × 256 bits) + 6	Up to 19 + 8
Return Command FIFO	22	24
SDRAM Read Data FIFO	22	24

Table 15-71. FIFO Allocation (continued)

Parameter	System Local Interface Entries	MPU Local Interface Entries
Register Read Data FIFO	2	0

(1) The total number of entries in the command FIFO is 10.

The command FIFO is shared between the two local interfaces, whereas there are two different FIFOs for every other type, one dedicated to each local interface.

The command FIFO stores all the commands coming in on the local command interface. The allocation of entries in the command FIFO is programmable per local interface using the following bit fields:

- EMIF_OCP_CONFIG[27:24] SYS_THRESH_MAX
- EMIF_OCP_CONFIG[23:20] MPU_THRESH_MAX

15.3.4.1.3 MPU Port Restrictions

The EMIF MPU port is defined only to process memory requests. All register accesses are processed through the system port of the EMIF. The EMIF MPU port does not support 2D or register requests required or provided by the system interface. The MPU port has a fixed ConnID equal to 0x0. The access burst length of the MPU port must not exceed 7.

To maintain coherency, the following rules must be followed:

- Any command arriving on MPU or system interface that matches an address in the command FIFO is executed after the command in the command FIFO
- The matching address is any address within a 2,000-address boundary
- On a 2D transfer, the starting address is the compared address. The computed addresses of the 2D transfer are not considered in address overlapping.
- Any command arriving within a 10-cycle window of another, from the different interfaces that do not match any address in the command FIFO, but may match command addresses arriving on a different interface, can be executed in any order.

15.3.4.1.4 Arbitration of Commands in the Command FIFO

The EMIF looks at all the commands stored in the command FIFO to schedule commands to the external memory. All commands with the same MTagID on a particular local interface complete in order. The EMIF does not ensure ordering between commands with different MTagIDs or between commands from two local interfaces.

However, the EMIF does maintain data coherency. Therefore, the EMIF blocks a command, regardless of priority or the local interface, if that command is to the same block address (2048 bytes) as an older command that is not complete. Thus, the EMIF may have one pending read or write for each MTagID. For information about MTagID, see Section 14.1.1, *Terminology* in Chapter 14, *Interconnect*. Among all pending reads, the EMIF selects all reads that have their corresponding SDRAM banks already open. Similarly, among all pending writes, the EMIF selects all writes that have their corresponding SDRAM banks already open. Accesses to memory mapped registers are treated as accesses that have open banks.

As a result of this reordering, the EMIF may now have several pending reads and writes that have their corresponding banks open. The EMIF then selects the highest priority read from pending reads, and the highest priority write from pending writes. If two or more commands have the highest priority, the EMIF selects the oldest command. As a result, the EMIF may now have the next read and a write command. If the return command FIFO and the read data FIFO have space and the external bus conflict is resolved, the EMIF performs the final read command before the final write command. If the return command FIFO has space but the read data FIFO is full, the EMIF performs the final write command before the final read command. Resolution of external bus conflict means all the SDRAM command-to-command counters are satisfied and the read-to-write or write-to-read turnaround time is met.

The EMIF does not support tag interleaving. In other words, for an local interface, the EMIF completes executing an local command before it switches to another command. The EMIF can, however, interleave execution between commands from two local interfaces.

The data coherency inside the EMIF is ensured only in a single level of local infrastructure. For example, if a write from a secondary local bus segment is blocked by a bridge element, the read from a tertiary bus can still beat the write to the EMIF. In such a case, to confirm that a write from master A has landed before a read from master B is performed, master A must wait for the write status from the EMIF before indicating to master B that the data is ready to be read. If master A does not use the local wait status, it must do the following:

1. Perform the required write.
2. Perform a dummy write to the EMIF_REVISION register.
3. Perform a dummy read to the EMIF_REVISION register.
4. Indicate to master B that the data is ready to be read after completion of read in Step 3. The completion of read in Step 3 ensures that the previous writes were done.

Apart from reads and writes, the EMIF must also open and close SDRAM banks and maintain the refresh counts for an SDRAM. The priority of SDRAM commands with respect to refresh levels are:

1. SDRAM refresh request when refresh-must level is reached (highest priority)
2. ZQ calibration
3. Leveling
4. local request for a read or write
5. local request for a write
6. SDRAM activate commands
7. SDRAM deactivate commands
8. SDRAM power-down request
9. SDRAM refresh request when refresh-may or release level is reached
10. SDRAM self-refresh request (lowest priority)

To avoid continuous blocking effect which can be caused by a continuous stream of high-priority commands which thus block the lower priority commands, the EMIF momentarily raises the priority of the oldest command over all other commands when the time for the oldest command configured through the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT bit field expires.

It should be taken into account that while performing the scheduling algorithm described, the EMIF may also encounter a condition in which continuous stream of SDRAM commands to a row in an open bank can block commands to another row in the same bank.

In addition to this scheduling, the highest priority condition is a reset command. If this condition occurs, the EMIF abandons what it is currently doing and begins its start-up sequence. In this case, commands and data stored in the FIFOs are lost. The EMIF also starts its start-up sequence whenever the EMIF_SDRAM_CONFIG register is written and the EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS bit is set to 0. In this case, commands and data stored in the FIFOs are not lost. The EMIF ensures that in-flight read or write transactions to the SDRAM are complete before starting the initialization sequence.

All the accesses to an SDRAM are pipelined to maximize use of the external bus. All of these are done while fulfilling the access timing requirements of an SDRAM.

15.3.4.2 Clock Management

15.3.4.2.1 EMIF_FICLK Overview

The EMIF can gate EMIF_FICLK. There is an internal mechanism that can stop EMIF_FICLK automatically. EMIF_FICLK is stopped only after the SDRAM is put into self-refresh mode and the power-idle protocol on the local bus completes. The EMIF_FICLK frequency can be changed only after putting the external SDRAM in self-refresh mode.

The EMIF waits for the DLL lock before performing any memory access.

EMIF_FICLK frequency is equal to half of the EMIF_PHY_FCLK frequency.

15.3.4.2.2 EMIF Dependency on MPU Clock Rate

The EMIF Write Data FIFO and Command FIFO clocks are derived from the MPU clock (specifically, EMIF_MA_ICLK = MPU_GCLK/4) any time the MPU is active. As such, the DDR Peak Write bandwidth scales in proportion to the MPU clock frequency. At lower MPU clock frequencies (< 1 GHz), the Write Data FIFO limits writes to less than peak DDR bandwidth (assuming DDR3-1066 operating conditions.)

The SDRAM Read Data FIFO is always clocked by the EMIF_ICLK, so there is no relationship for read bandwidth relative to the MPU operating frequency.

The Command FIFO bandwidth is also controlled by the EMIF_MA_ICLK. However, because the command bandwidth is much lower than the data bandwidth, there is no visible bandwidth scaling for the command interface (and as a result, the read interface can be fully used).

The total available bandwidth is not affected for most systems. The write bandwidth is limited on an internal path, not on the DDR pins. The DDR pin bandwidth is still available for reads. Because most systems have higher bandwidth requirements for read relative to write, the available DDR bandwidth is still usable.

15.3.4.3 Reset

The EMIF does not support a software reset.

The EMIF supports a global warm reset mode, during which the EMIF keeps the SDRAM content. Upon a request from the PRCM module indicating a need to enter global warm reset mode, the EMIF does the following:

1. During leveling operation, EMIF will immediately exit this mode and automatically perform a write to the MR1 register of DDR3 memory to disable the leveling at the memory side too.
2. EMIF completes the ongoing access, and then puts the SDRAM in self-refresh mode. If the EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS field is set to 1, the EMIF does not put SDRAM in self-refresh mode.
3. EMIF clears all its FIFO contents.
4. EMIF does not wait for all interrupts to be serviced.

To exit the global warm reset:

1. If the EMIF was in Self Refresh state, it will exit Self Refresh state.
2. If leveling was enabled at the time of a global warm reset, a PHY reset must occur to bring the PHY back into a known state, as it may have been left in a leveling state upon warm reset assertion. To guarantee that the SDRAM memory clocks are off when issuing PHY reset, software can use the EMIF_POWER_MANAGEMENT_CONTROL register to enter self refresh before asserting the PHY reset.

15.3.4.4 System Power Management

15.3.4.4.1 Power-Down Mode

The EMIF supports SDRAM power-down mode for low power. The EMIF automatically puts the SDRAM into power-down mode after it is idle for EMIF_POWER_MANAGEMENT_CONTROL[15:12] PD_TIM number of DDR clock cycles and the EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is set to 0x4. In power-down mode, the EMIF does not stop the clocks to the SDRAM. The EMIF maintains CKE pin low to maintain the power-down mode.

If refresh-must level is not reached before power-down entering, EMIF will not precharge all SDRAM banks before it issues the power-down command. As a result of this EMIF puts the SDRAM in active power-down mode. If refresh-must level is reached before power-down entering, EMIF will precharge all SDRAM banks and before it issues the power-down command, EMIF issues refreshes until refresh-release level is reached. As a result of this EMIF puts the SDRAM in precharge power-down mode.

When the SDRAM is in power-down mode, the EMIF services register accesses normally.

If the SDRAM is in power-down mode and one of the following occurs, the EMIF brings SDRAM out of power-down mode:

- EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is changed from 0x4 to some other value.
- An SDRAM access is requested.
- The refresh-must level is reached.

If refresh-must level brings the SDRAM out of power-down mode, EMIF puts it in power-down again when the refreshes are complete and keeps this state until the next SDRAM request.

To exit power-down, the EMIF:

1. Drives CKE high after t_cke + 1 cycles have elapsed since the power-down command was issued. The value of t_cke is taken from EMIF_SDRAM_TIMING_2[2:0] T_CKE bit field.
2. Waits for EMIF_SDRAM_TIMING_2[30:28] T_XP + 1 cycles
3. Enters its idle state and can issue any commands

15.3.4.4.2 Self-Refresh Mode

The EMIF supports SDRAM self-refresh mode for low power. The EMIF automatically puts the SDRAM into self-refresh mode after the EMIF is idle for EMIF_POWER_MANAGEMENT_CONTROL[7:4] SR_TIMING number of DDR clock cycles and the EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is set to 0x2. The EMIF will complete all pending refreshes before it puts the SDRAM into self-refresh. Therefore, after the expiration of SR_TIMING, the EMIF will start issuing refreshes to complete the refresh backlog, and then issue a self-refresh command to the SDRAM.

In self-refresh mode, the EMIF automatically stops the SDRAM clock. The EMIF drives CKE pin low to maintain self-refresh mode.

When the SDRAM is in self-refresh mode, the EMIF services register accesses normally.

If the SDRAM is in self-refresh mode and one of the following occurs, the EMIF brings SDRAM out of self-refresh mode:

- The EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is changed from 0x2 to some other value.
- SDRAM access is requested.
- EMIF_SDRAM_TIMING_2[2:0] T_CKE + 1 cycles have elapsed since the last self-refresh command.

To exit self-refresh, for DDR2, the EMIF:

1. Enables the SDRAM clock
2. Drives CKE pin high
3. Waits EMIF_SDRAM_TIMING_2[24:16] T_XSNR + 1 cycles
4. Starts a refresh cycle in the next cycle.
5. Enters its idle state and can issue any other command except write or read command. A write or a read command is issued only after EMIF_SDRAM_TIMING_2[15:6] T_XSRD + 1 cycles clock cycles have elapsed since pin CKE is driven high.

To exit self-refresh, for DDR3, the EMIF does the following:

1. Enables the SDRAM clock
2. Drives CKE pin high
3. Waits EMIF_SDRAM_TIMING_2[24:16] T_XSNR + 1 cycles
4. Starts a refresh cycle in the next cycle. EMIF also services all refreshes down to the refresh-release level.
5. Enters its idle state and can issue any other command except write or read command. A write or a read command is issued only after EMIF_SDRAM_TIMING_2[15:6] T_XSRD + 1 cycles clock cycles have elapsed since pin CKE is driven high.

To use partial array self-refresh, the EMIF_SDRAM_REFRESH_CONTROL[26:24] PASR bits must be appropriately programmed. The EMIF performs bank interleaving when EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0x0. Because the SDRAM is partially refreshed during partial array self-refresh, for software ease, it is recommended that the IBANK_POS bit field to be set to 0x1, 0x2, or 0x3 depending on the scheme used. If IBANK_POS is set to 0x0, software must move critical data into the banks that are going to be refreshed during partial array self-refresh.

15.3.4.5 Interrupt Requests

Each EMIF controller is able to generate one interrupt request which is connected to the IRQ_CROSSBAR module.Totally, there are two interrupt lines connected to two IRQ_CROSSBAR inputs. EMIF1 produces EMIF1_IRQ and EMIF2 produces EMIF2_IRQ. These interrupt lines can be asserted by one of the interrupt events listed in Table 15-72.

Each EMIF controller generates an interrupt on its interrupt line only if the interrupts are enabled by setting to 0x1 the corresponding bits in the EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET register. These interrupts can be disabled by setting to 0x1 the corresponding bits in the EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR register. After the interrupt has been serviced the corresponding status flag must be cleared by software. This is done by setting to 0x1 the corresponding bit in the EMIF_SYSTEM_OCP_INTERRUPT_STATUS register which also clears the corresponding bit in the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register. The status flags in the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register are set even if the corresponding interrupt is disabled unlike those in the EMIF_SYSTEM_OCP_INTERRUPT_STATUS register, which are set only if the corresponding interrupt is enabled. An interrupt is also generated by the EMIF controller, if certain bit in the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register is set to 0x1 and the corresponding interrupt is enabled through the EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET register. This feature is useful when user software debugging is performed. In addition, even if interrupts are not enabled, certain status bit in EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register can be cleared by setting to 0x1 the corresponding bit in the EMIF_SYSTEM_OCP_INTERRUPT_STATUS register.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[0] ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[0] ERR_SYS bits to 0x1, if access request for an unsupported command type, an unsupported addressing mode or an access request to an unsupported MAddrSpace is received. If such an error occurs, it is due to bad programming of the DMM. For more information about addressing, see Section 15.2, *Dynamic Memory Manager*.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[3] WR_ECC_ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[3] WR_ECC_ERR_SYS bits to 0x1, if a write access with byte count that is not multiple of the ECC quanta or with a non ECC quanta aligned address is performed within the address range protected by the ECC.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[4] TWOBIT_ECC_ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[4] TWOBIT_ECC_ERR_SYS bits to 0x1, if 2-bit ECC error for a read access performed within the address range protected by the ECC occurs.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[5] ONEBIT_ECC_ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[5] ONEBIT_ECC_ERR_SYS bits to 0x1, if the threshold for 1-bit ECC error is reached. For more information about the EMIF ECC feature, see Section 15.3.4.14, *Error Correction And Detection Feature*.

Table 15-72 lists the event flags and their corresponding event mask bits of the sources which can cause module interrupts.

Table 15-72. Events

Event Flag	Event Mask	Description
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[5] ONEBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[5] ONEBIT_ECC_ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[5] ONEBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[5] ONEBIT_ECC_ERR_SYS	Interrupt if one bit ECC error threshold is reached
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[4] TWOBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[4] TWOBIT_ECC_ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[4] TWOBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[4] TWOBIT_ECC_ERR_SYS	Interrupt for two bit error detection

Table 15-72. Events (continued)

Event Flag	Event Mask	Description
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[3] WR_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[3] WR_ECC_ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[3] WR_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[3] WR_ECC_ERR_SYS	Interrupt for memory access made to a non-quanta aligned location or done with byte count not multiple of the ECC quanta
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[0] ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[0] ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[0] EN_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[0] EN_ERR_SYS	Interrupt for command or address error

15.3.4.6 SDRAM Refresh Scheduling

The EMIF uses two counters to schedule the Refresh (REF) commands: a 13-bit decrementing refresh interval counter and a 4-bit refresh backlog counter. The interval counter is used to define the rate at which connected SDRAM devices are refreshed. It is loaded with the value of the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field at reset (only the 13 LSBs are taken). The interval counter decrements by 1 each cycle until it reaches 0x0, at which point it reloads from the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field and restarts decrementing. The counter also reloads and restarts decrementing whenever the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field is updated.

The refresh backlog counter records the number of the outstanding REF commands which the EMIF controller currently has. The backlog counter increments by 1 each time the interval counter reloads (unless it has reached its maximum value of 8). The backlog counter decrements by 1 each time the EMIF issues a REF command (unless it is already 0). For the range of values that the backlog counter can take, there are three levels of urgency with which the EMIF must perform refresh cycle in which it issues REF commands:

1. Refresh-may level is reached when the backlog count is greater than 0x0, which indicates that there is a refresh backlog and if the EMIF is not busy and there are no open SDRAM banks, the EMIF must perform refresh cycle.
2. Refresh-release level is reached when the backlog count is greater than 0x4, which indicates that the refresh backlog is getting bigger and if the EMIF is not busy it must perform refresh cycle even if there is an open SDRAM bank.
3. Refresh-must level is reached when the backlog count is greater than 0x7, which indicates that the refresh backlog is becoming excessive and the EMIF must perform refresh cycle before any new memory access request being serviced. The EMIF starts servicing new memory accesses after the refresh-release level is cleared.

The two counters do not operate when SDRAM is in self-refresh mode. They start tracking the missed refreshes (the outstanding REF commands) only after initialization is complete.

The time between two REF commands is set through the EMIF_SDRAM_TIMING_3[12:4] T_RFC bit field.

15.3.4.7 SDRAM Initialization

Note

To avoid error responses from the EMIF controller the SDRAM initialization must be performed per 16-Byte blocks if EMIF_SDRAM_CONFIG[15:14] NARROW_MODE = 0x0 or per 8-Byte blocks if EMIF_SDRAM_CONFIG[15:14] NARROW_MODE = 0x1.

15.3.4.7.1 DDR2 SDRAM Initialization

On coming out of reset, the EMIF controller begins the DDR2 initialization sequence after a write to any one of the following three registers, providing that the corresponding listed conditions are met.

1. EMIF_SDRAM_CONFIG
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_CONFIG)

2. EMIF_SDRAM_TIMING_1
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_TIMING_1) AND
   b. Condition 2: The write access modifies register bit field T_WR (bits 20:17)
3. EMIF_SDRAM_REFRESH_CONTROL
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 AND
   b. Condition 2: The write access modifies register bit field SRT (bit 29), ASR (bit 28), or PASR (bits 26:24).

For the first DDR2 initialization sequence, the EMIF controller performs the following actions:

1. Drives the CKE pin low.
2. After 16 SDRAM refresh rate intervals, issues a NOP command with CKE pin held high. The SDRAM refresh rate is as defined in the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field.
3. After 1 SDRAM refresh rate interval, issues Precharge all Banks command.
4. Issues EMRS command to the DDR2 EMR(2) register (bits BA[2:0] = 0x2) with bits A[15:0] = 0x0.
5. Issues EMRS command to the DDR2 EMR(3) register (bits BA[2:0] = 0x3) with bits A[15:0] = 0x0.
6. Issues EMRS command to the DDR2 EMR(1) register (bits BA[2:0] = 0x1) with bits A[15:0] set as in Table 15-73.

Table 15-73. Load Value For The EMR(1) Register During DDR2 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Output buffer enabled
A[11]	0x0	RDQS disable
A[10]	EMIF_SDRAM_CONFIG[23] DDR2_DDQS	Differential DQS enable value
A[9:7]	0x0	Exit OCD calibration mode
A[6], A[2]	EMIF_SDRAM_CONFIG[26:24] DDR_TERM	DDR2 termination resistor value. For DDR2 the EMIF_SDRAM_CONFIG[26] bit is not used.
A[5:3]	0x0	Additive latency = 0
A[1]	EMIF_SDRAM_CONFIG[19:18] SDRAM_DRIVE	SDRAM drive strength. For DDR2 the EMIF_SDRAM_CONFIG[19] bit is not used.
A[0]	EMIF_SDRAM_CONFIG[20] DDR_DISABLE_DLL = 0x0	Enable DLL

7. Issues MRS command to the DDR2 MR register (BA[2:0] = 0x0) with A[15:0] set as in Table 15-74.

Table 15-74. Load Value For The MR Register During DDR2 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Active power down exit time - fast exit
A[11:9]	EMIF_SDRAM_TIMING_1[20:17] T_WR	Write recovery for autoprecharge
A[8]	0x1	DLL reset
A[7]	0x0	Normal mode
A[6:4]	EMIF_SDRAM_CONFIG[13:10] CL	CAS latency value. For DDR2 the EMIF_SDRAM_CONFIG[13] bit is not used.
A[3]	0x0	Sequential burst type
A[2:0]	0x3	Burst length of 8

8. After 267 clock cycles, issues Precharge all Banks command.
9. After two Refresh commands, issues MRS command to the DDR2 MR register (BA[2:0] = 0x0) with A[15:0] set as in Table 15-75.

Table 15-75. Another Load Value For The MR Register During DDR2 SDRAM Initialization

Bits	Value	Description
A[15:9]	Equal to Step 7	
A[8]	0x0	No DLL reset
A[7:0]	Equal to Step 7	

10. Issues EMRS command to the DDR2 EMR(1) register (bits BA[2:0] = 0x1) with bits A[15:0] equal to Step 6.
11. The EMIF enters its IDLE state.

The EMIF updates the DDR Mode registers if the DDR2 initialization sequence is triggered again. However, the EMIF controller starts from Step 3.

The EMIF does not perform any transactions until the DDR2 initialization sequence is complete.

When the EMIF comes out of reset, the delay time in Step 2 resulting from the 16 refresh rate intervals + 8 cycles is approximately 16 × REFRESH_RATE / input frequency.

Note

The values of the bit fields in the EMIF_SDRAM_CONFIG register are loaded by the control module at reset. These values can be modified by the configuration header (CH) feature of the ROM code or by the initial boot image running from an external booting memory or internal RAM. They must not be modified during run time, because they reflect the used hardware SDRAM memory configurations.

15.3.4.7.2 DDR3 SDRAM Initialization

On coming out of reset, the EMIF controller begins the DDR3 initialization sequence after a write to any one of the following three registers, providing that the corresponding listed conditions are met.

1. EMIF_SDRAM_CONFIG
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_CONFIG)
2. EMIF_SDRAM_TIMING_1
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_TIMING_1) AND
   b. Condition 2: The write access modifies register bit field T_WR (bits 20:17)
3. EMIF_SDRAM_REFRESH_CONTROL
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 AND
   b. Condition 2: The write access modifies register bit field SRT (bit 29), ASR (bit 28), or PASR (bits 26:24).

For the first DDR3 initialization sequence, the EMIF controller performs the following actions:

1. After 7 SDRAM refresh rate intervals, de-asserts the RST pin.
2. After 16 SDRAM refresh rate intervals, issues a NOP command with CKE pin held high. The SDRAM refresh rate is as defined in the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field.
3. After 1 SDRAM refresh rate interval, issues MRS command to the DDR3 MR2 register (bits BA[2:0] = 0x2) with bits A[15:0] set as in Table 15-76

Table 15-76. Load Value For The MR2 Register During DDR3 SDRAM Initialization

Bits	Value	Description
A[15:11]	0x0	Reserved
A[10:9]	EMIF_SDRAM_CONFIG[22:21] DYN_ODT	Dynamic ODT value

Table 15-76. Load Value For The MR2 Register During DDR3 SDRAM Initialization (continued)

Bits	Value	Description
A[8]	0x0	Reserved
A[7]	EMIF_SDRAM_REFRESH_CONTROL[29] SRT	Self-refresh temperature range
A[6]	EMIF_SDRAM_REFRESH_CONTROL[28] ASR	Auto self-refresh enable
A[5]	0x0	Reserved
A[4:3]	EMIF_SDRAM_CONFIG[17:16] CWL	CAS write latency
A[2:0]	EMIF_SDRAM_REFRESH_CONTROL[26:24] PASR	Partial array self-refresh

4. After T_RFC + 1 DDR clock cycles, issues MRS command to the DDR3 MR3 register (bits BA[2:0] = 0x3) with A[15:0] = 0x0
5. After T_RFC + 1 DDR clock cycles, issues MRS command to the DDR3 MR1 register (BA[2:0] = 0x1) with A[15:0] set as in Table 15-77

Table 15-77. Load Value For The MR1 Register During DDR3 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Output buffer enabled
A[11]	0x0	TDQS disable
A[10]	0x0	Reserved
A[9], A[6], A[2]	EMIF_SDRAM_CONFIG[26:24] DDR_TERM	DDR3 termination resistor value
A[8]	0x0	Reserved
A[7]	0x0	Write leveling disabled
A[5], A[1]	EMIF_SDRAM_CONFIG[19:18] SDRAM_DRIVE	SDRAM drive strength
A[4:3]	0x0	Additive latency = 0
A[0]	EMIF_SDRAM_CONFIG[20] DDR_DISABLE_DLL = 0x0	Enable SDRAM DLL

6. After T_RFC + 1 DDR clock cycles, issues MRS command to the DDR3 MR0 register (BA[2:0] = 0x0) with A[15:0] set as in Table 15-78

Table 15-78. Load Value For The MR0 Register During DDR3 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Slow exit. The DDR3 SDRAM DLL is “OFF” after entering precharge power-down.
A[11:9]	EMIF_SDRAM_TIMING_1[20:17] T_WR	Write recovery for autoprecharge
A[8]	0x1	DLL reset
A[7]	0x0	Normal mode
A[6:4], A[2]	EMIF_SDRAM_CONFIG[13:10] CL	Value for CAS latency
A[3]	0x0	Nibble sequential read burst type

Table 15-78. Load Value For The MR0 Register During DDR3 SDRAM Initialization (continued)

Bits	Value	Description
A[1:0]	0x0	Burst length of 8

7. After T_RFC + 1 DDR clock cycles, issues a ZQCL command to start long ZQ calibration
8. Waits for t_{DLLK} and t_{ZQinit} to complete
9. Issues REF command
10. The EMIF enters its IDLE state.

The EMIF updates the DDR Mode registers if the DDR3 initialization sequence is triggered again. However, the EMIF controller first issues a precharge command and then starts from Step 3.

The EMIF does not perform any transactions until the DDR3 initialization sequence is complete.

When the EMIF comes out of reset, the delay time in Step 2 resulting from the 16 refresh rate intervals + 8 cycles is approximately 16 × REFRESH_RATE / input frequency.

15.3.4.8 DDR3 Read-Write Leveling

The EMIF supports one type of write/read leveling for DDR3 - full leveling.

The full leveling consists of three parts:

1. Write leveling
2. Read data eye training
3. Read DQS gate training

Write leveling

The goal of write leveling is to locate the delay between the rising edge of the write DQS signal and the rising edge of the SDRAM memory clock (CK). When this delay is identified, the system is able to accurately align the write DQS signal with the DDR3 memory clock. During Write leveling, the ODT function must be on and proper ODT values must be selected at the external memory side by setting Rtt_Nom (A9, A6, A2) bits in MR1 register of the external DDR3 memory. For more information about write leveling, see the *DDR3 SDRAM Standard*, section *Write leveling*.

Read data eye training

Through the read data eye training the delay between the rising edge of the read DQS signal and the rising and falling edges of the associated DQ data eye is determined. By identifying these delays, the midpoint between them can be calculated and thus the rising edge of the read DQS signal can be accurately centered within the DQ data eye.

Read DQS gate training

Read DQS Gate training is used for timing the internal read window during a read operation as opposed to the write leveling and read data eye training which are used for skew compensation of external signals. The goal of read DQS gate training is to locate the shortest delay that can be applied to each DQS gate such that it functions properly, then find the longest delay that can be applied to each DQS gate and keep its proper function again, and then align the midpoint of the DQS gate delay between these two.

15.3.4.8.1 Full Leveling

The EMIF does not perform full leveling after initialization upon reset. Full leveling must be triggered by software after the EMIF's registers are properly configured.

Full leveling is triggered by setting the EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START bit to 0x1. The leveling execution order is as follows:

1. Write leveling
2. Read DQS gate training
3. Read data eye training

After leveling procedure has finished the EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START bit clears itself automatically.

Note

SDRAM Refreshes must be disabled before triggering full leveling.

Note

The EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_21 registers have to be configured only in case of software leveling.

15.3.4.8.2 Software Leveling

In case of software leveling, the following registers must be configured:

- EMIF_EXT_PHY_CONTROL_2 to EMIF_EXT_PHY_CONTROL_6 containing the PHY_REG_FIFO_WE_SLAVE_RATIO value
- EMIF_EXT_PHY_CONTROL_26 to EMIF_EXT_PHY_CONTROL_30 containing the REG_PHY_GATELVL_INIT_RATIO value
- EMIF_EXT_PHY_CONTROL_25 containing the DQ offset value

The bit fields of the EMIF_EXT_PHY_CONTROL_2 to EMIF_EXT_PHY_CONTROL_6 registers must be loaded with same values. The bit fields of the EMIF_EXT_PHY_CONTROL_26 to EMIF_EXT_PHY_CONTROL_30 registers must also be loaded with same values.

The REG_PHY_DQ_OFFSET fields in register EMIF_EXT_PHY_CONTROL_25 must be loaded with 0x40. That value corresponds to quarter cycle shift between the DQS signals and the data to be written to the SDRAM.

To calculate the PHY_REG_FIFO_WE_SLAVE_RATIO value one of the following two formulas can be used:

- 0x80 + 2*[(Board Delay in ps)*0x100]/(Clock Period in ps). This formula is used when the EMIF_DDR_PHY_CONTROL_1[18] PHY_INVERT_CLKOUT bit is set to 0x1.
- 2*[(Board Delay in ps)*0x100]/(Clock Period in ps). This formula is used when the EMIF_DDR_PHY_CONTROL_1[18] PHY_INVERT_CLKOUT bit is set to 0x0.

The Board Delay in the previously described formulas is measured directly from the board. It depends on the trace length. When the calculated value for PHY_REG_FIFO_WE_SLAVE_RATIO is greater than 0x20, then a value of 0x27 must be used. That means, 0x27 must be loaded in registers EMIF_EXT_PHY_CONTROL_2 to EMIF_EXT_PHY_CONTROL_6.

To calculate the REG_PHY_GATELVL_INIT_RATIO value the following formula is used:

REG_PHY_GATELVL_INIT_RATIO = PHY_REG_FIFO_WE_SLAVE_RATIO – 0x20

When the calculated value for REG_PHY_GATELVL_INIT_RATIO is less than 0x0, then a value of 0x0 must be used. When the calculated value is greater than or equal to 0x0, then that value is used for registers EMIF_EXT_PHY_CONTROL_26 to EMIF_EXT_PHY_CONTROL_30.

In addition, when read DQS gate training is not performed the PHY_REG_FIFO_WE_SLAVE_RATIO value is used. When read DQS gate training is performed REG_PHY_GATELVL_INIT_RATIO is used and PHY_REG_FIFO_WE_SLAVE_RATIO is don't care.

Note

Software leveling is not recommended to be used. Hardware leveling must be used instead.

15.3.4.9 EMIF Access Cycles

By default, the EMIF keeps its SDRAM $\overline{CS}$ signal high. To direct a command to only one of the SDRAMs, EMIF asserts its $\overline{CS}$ signal to the SDRAM for the duration of the command.

The EMIF always performs burst accesses to the SDRAM. Multiple SDRAM bursts may need to service a single local burst request. Table 15-79 through Table 15-82 show a few examples how EMIF performs SDRAM accesses for a linear incrementing transaction type. T0, T1, etc. are clock cycles. R0 is read starting at column 0, R8 is read starting at column 8, and R16 is read starting at column 16. D0-1 is the data from column 0 and 1, D2-3 is the data from column 2 and 3, and so on.

Table 15-79. 64-Byte Linear Read Starting at Address 0x0 (DDR2 and DDR3)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
R0				R8							
				D0-1	D2-3	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15

Table 15-80. 64-Byte Linear Read Starting at Address 0x8 (DDR2)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
R2		R4		R8				R16							
				D2-3	Unused	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	Unused	Unused	Unused

Table 15-81. 64-Byte Linear Read Starting at Address 0x10 (DDR2 and DDR3)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R4		R8				R16							
				D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	Unused	Unused

Table 15-82. 64-Byte Linear Read Starting at Address 0x18 (DDR2 and DDR3)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R6		R8				R16							
				D6-7	Unused	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	D20-21	Unused

The EMIF uses the unused data phases in the preceding figures by issuing successive read commands if there are reads to open banks pending in the command FIFO.

The write data conversion from SDR to DDR is done outside the EMIF.

15.3.4.10 Turnaround Time

Table 15-83 lists the turnaround time that EMIF introduces on the data bus for various back-to-back accesses. The EMIF takes advantage of the CAS latencies and packs the commands as close as possible on the control bus to introduce the following turnaround time on the data bus.

Table 15-83. Turnaround Time

Current Access	Next Access	Turnaround Time (Number of DDR Clock Cycles)
SDRAM write	SDRAM read	EMIF_SDRAM_TIMING_1[2:0] T_WTR + 1 + CL
SDRAM read	SDRAM write	EMIF_SDRAM_TIMING_1[31:29] T_RTW + 1

15.3.4.11 PHY DLL Calibration

When running in normal locked mode, the PHY DLL gets a reference clock (EMIFi_DLL_FCLK) from the PRCM, which is used by the DLL master to lock to the right frequency and provide the control code for a full period phase shift to the slave. The slave uses this code as a control for its internal delay line to produce the required delay for the signal considered.

When working in locked mode, the delay lines only get an updated control value from the master DLLs when an explicit dll_calib command is issued by the EMIF controller. Failure to send such commands on a timely basis will result in inaccurate delay-line information if there is a significant voltage and temperature drift in the system. It is recommended to issue at least one command every 100 µs. EMIF automatically sends ctrl_update commands for:

- Refresh Exit
- Self Refresh Exit
- phy_ready asserted during initialization

The PHY also internally generates a control value update upon completion of a leveling operation. Control is also added when leveling is not used and there are gradual voltage changes during frequency change. The EMIF_DLL_CALIB_CTRL register can be programmed to generate a phy_dll_calib for a periodic interval based on EMIF_FICLK cycles, so allow continued memory access as voltage is changing. A safe window of no activity will be guaranteed for this periodic generation of phy_dll_calib. In addition, a one shot generator for phy_pll_calib has also been added that will generate a single phy_dll_calib by setting the EMIF_MISC_REG[0] DLL_CALIB_OS bit to 1.

15.3.4.12 SDRAM Address Mapping

From the system point of view, the external SDRAM is seen as one block of SDRAM. If two external 64-MiB devices are used, a 128-MiB memory block is observed. If two external 32-MiB devices are used, a 64-MiB block is observed.

Table 15-84 shows the SDRAM address space.

Table 15-84. SDRAM Addressing Space

Module Name	Base Address	Size
EMIF1 | EMIF2-CS0-SDRAM(1)	0x8000 0000	0 to 1GiB, programmable in DMM (see Section 15.2, *Dynamic Memory Manager*)

(1) The addressing space is interleaved on two SDRAM memory controllers (EMIF1, EMIF2), each activating itsCS0 line. These CSs can address 64, 128, 256, 512, 1024, or 2048MiB. Interleaving occurs at 128-byte granularity. EMIF1-CS0 base address is always 0x8000 0000 at reset, and occupies a 1-GiB address space at reset (interleaving disabled at reset).

When addressing SDRAM, if the EMIF_SDRAM_CONFIG[28:27] IBANK_POS bit field is set to 0, the EMIF uses the following two bit fields to determine the mapping from the source address to the SDRAM row, column and bank:

- EMIF_SDRAM_CONFIG[6:4] IBANK
- EMIF_SDRAM_CONFIG[2:0] PAGESIZE

If the EMIF_SDRAM_CONFIG[28:27] IBANK_POS bit field is set to 1, 2, or 3, the EMIF uses the following three bit fields to determine the mapping from the source address to the SDRAM row, column and bank:

- EMIF_SDRAM_CONFIG[6:4] IBANK
- EMIF_SDRAM_CONFIG[2:0] PAGESIZE
- EMIF_SDRAM_CONFIG[9:7] ROWSIZE

In all cases the EMIF considers its SDRAM address space to be a single logical block, regardless of the number of physical devices.

15.3.4.12.1 Address Mapping for IBANK_POS = 0 and EBANK_POS = 0

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0 and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 0, Table 15-85 lists which source address bits (MAddr) are mapped to the SDRAM row, column and bank bits for all combinations of IBANK and PAGESIZE.

Table 15-85. Local Address to SDRAM Address Mapping for IBANK_POS = 0 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
row address		bank address		column address	
ROWSIZE value	row width (bits)	IBANK value	bank[2:0] width (bits)	PAGESIZE value	col width (bits)
In this case the ROWSIZE bit field is not used	16	0	0	0	8
		1	1	1	9
		2	2	2	10
		3	3	3	11

Note

The ROWSIZE bit field is unused in case of IBANK_POS = 0 and EBANK_POS = 0.

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0, the effect of the address-mapping scheme is that as the source address increments across the SDRAM pages, EMIF moves to page with the same number as in the previous bank. This movement across the banks continues until the same page is accessed in all banks and then EMIF moves to the next page in the first bank. The EMIF uses this movement across internal banks while remaining on the same page to maximize the number of the open SDRAM banks within the overall SDRAM space.

Thus, the EMIF can keep a maximum of 8 banks open at a time, and can interleave among all of them.

15.3.4.12.2 Address Mapping for IBANK_POS = 1 and EBANK_POS = 0

Table 15-86 list the local address to SDRAM address mapping when IBANK_POS = 1 and EBANK_POS = 0.

Table 15-86. Local Address to SDRAM Address Mapping for IBANK_POS = 1 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2]		row address		bank address[1:0]		column address	
IBANK value	bank[2] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[1:0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8
1	0	1	10	1	1	1	9
2	0	2	11	2	2	2	10
3	1	3	12	3	2	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 1, the EMIF interleaves banks the same as for EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0 but the interleaving of banks is limited to four banks. Thus, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 4 of them.

15.3.4.12.3 Address Mapping for IBANK_POS = 2 and EBANK_POS = 0

Table 15-87 list the local address to SDRAM address mapping when IBANK_POS = 2 and EBANK_POS = 0.

Table 15-87. Local Address to SDRAM Address Mapping for IBANK_POS = 2 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2:1]		row address		bank address[0]		column address	
IBANK value	bank[2:1] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8

Table 15-87. Local Address to SDRAM Address Mapping for IBANK_POS = 2 and EBANK_POS = 0 (continued)

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2:1]		row address		bank address[0]		column address	
IBANK value	bank[2:1] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[0] width (bits)	PAGESIZE value	col width (bits)
1	0	1	10	1	1	1	9
2	1	2	11	2	1	2	10
3	2	3	12	3	1	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 2, the EMIF interleaves banks the same as for EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0 but the interleaving of banks is limited to two banks. Thus, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 2 of them.

15.3.4.12.4 Address Mapping for IBANK_POS = 3 and EBANK_POS = 0

Table 15-88 list the local address to SDRAM address mapping when IBANK_POS = 3 and EBANK_POS = 0.

Table 15-88. Local Address to SDRAM Address Mapping for IBANK_POS = 3 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
bank address		row address		column address	
IBANK value	bank width (bits)	ROWSIZE value	row width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	8
1	1	1	10	1	9
2	2	2	11	2	10
3	3	3	12	3	11
		4	13		
		5	14		
		6	15		
		7	16		

If EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 3, the bank interleaving is not possible for EMIF.

15.3.4.12.5 Address Mapping for IBANK_POS = 0 and EBANK_POS = 1

Table 15-89 list the local address to SDRAM address mapping when IBANK_POS = 0 and EBANK_POS = 1.

Table 15-89. Local Address to SDRAM Address Mapping for IBANK_POS = 0 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
row address		bank address		column address	
ROWSIZE value	row width (bits)	IBANK value	bank width (bits)	PAGESIZE value	col width (bits)
0	9	0	0	0	8
1	10	1	1	1	9
2	11	2	2	2	10
3	12	3	3	3	11
4	13				
5	14				
6	15				

Table 15-89. Local Address to SDRAM Address Mapping for IBANK_POS = 0 and EBANK_POS = 1 (continued)

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
row address		bank address		column address	
ROWSIZE value	row width (bits)	IBANK value	bank width (bits)	PAGESIZE value	col width (bits)
7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0, and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the EMIF can keep a maximum of 8 banks open at a time and can interleave among all of them.

15.3.4.12.6 Address Mapping for IBANK_POS = 1 and EBANK_POS = 1

Table 15-90 list the local address to SDRAM address mapping when IBANK_POS = 1 and EBANK_POS = 1.

Table 15-90. Local Address to SDRAM Address Mapping for IBANK_POS = 1 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2]		row address		bank address[1:0]		column address	
IBANK value	bank[2] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[1:0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8
1	0	1	10	1	1	1	9
2	0	2	11	2	2	2	10
3	1	3	12	3	2	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 1, and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 4 of them.

15.3.4.12.7 Address Mapping for IBANK_POS = 2 and EBANK_POS = 1

Table 15-91 list the local address to SDRAM address mapping when IBANK_POS = 2 and EBANK_POS = 1.

15.3.4.12.8

Table 15-91. Local Address to SDRAM Address Mapping for IBANK_POS = 2 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2:1]		row address		bank address[0]		column address	
IBANK value	bank[2:1] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8
1	0	1	10	1	1	1	9
2	1	2	11	2	1	2	10
3	2	3	12	3	1	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 2 and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 2 of them.

15.3.4.12.9 Address Mapping for IBANK_POS = 3 and EBANK_POS = 1

Table 15-92 list the local address to SDRAM address mapping when IBANK_POS = 3 and EBANK_POS = 1.

Table 15-92. Local Address to SDRAM Address Mapping for IBANK_POS = 3 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
bank address		row address		column address	
IBANK value	bank width (bits)	ROWSIZE value	row width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	8
1	1	1	10	1	9
2	2	2	11	2	10
3	3	3	12	3	11
		4	13		
		5	14		
		6	15		
		7	16		

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 3 and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the bank interleaving is not possible for EMIF.

15.3.4.13 DDR3 Output Impedance Calibration

The EMIF controller supports automatic output impedance (ZQ) calibration for DDR3 memories. This feature is supported by DDR3 and not supported by DDR2 memories. The ZQ calibration can be enabled by setting to 0x1 the EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[30] ZQ_CS0EN bit. The EMIF supports three types of ZQ calibration commands:

- ZQINIT: ZQ calibration command during initialization
- ZQCL: ZQ calibration long command
- ZQCS: ZQ calibration short command

The EMIF automatically issues ZQINIT command during DDR3 memory initialization. It also issues ZQCS command each time the EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[15:0] ZQ_REFINTERVAL bit field expires. In other words, the ZQ_REFINTERVAL defines the interval between two ZQCS commands. When ZQCS command is issued, the EMIF waits and blocks any other command for EMIF_SDRAM_TIMING_3[20:15] ZQ_ZQCS + 1 number of DDR clock cycles.

If the EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[28] ZQ_SFEXITEN bit field is set to 0x1, the EMIF issues ZQCL command every time it exits self-refresh, active power-down and precharge power-down modes. When ZQCL command is issued, the EMIF waits and blocks any other command for (EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[17:16] ZQ_ZQCL_MULT + 1) × (EMIF_SDRAM_TIMING_3[20:15] ZQ_ZQCS + 1) number of DDR clock cycles.

The ZQINIT is a non periodic command issued only once during DDR3 initialization as opposed to the ZQCL and ZQCS calibration commands which are issued by the EMIF periodically at regular intervals.

15.3.4.14 Error Correction And Detection Feature

For data integrity, the EMIF1 supports ECC on the data written or read from the SDRAM and is enabled by programming the EMIF_ECC_CTRL_REG register. ECC accesses are allowed for the both SYS and the MPU ports. 7-bit ECC is calculated over 32-bit data when in 32-bit DDR mode. 6-bit ECC is calculated over 16-bit data when in 16-bit DDR mode. The ECC is calculated for all accesses that are within the address ranges protected by ECC. The address ranges are specified in the EMIF_ECC_ADDRESS_RANGE_1 and EMIF_ECC_ADDRESS_RANGE_2 registers. Both registers have identical bits and functionality. This provides flexibility allowing two non-overlapping memory regions to be ECC protected.

The system must ensure that any burst access with starting address in the ECC protected region must not cross over to the un-protected region and vice-versa. The ECC is stored inside the SDRAM during writes. If a write access with byte count that is not a multiple of ECC quanta or with a non quanta aligned address is performed within the address range protected by ECC, the EMIF will send out a write ECC error interrupt. The EMIF will also report an error on the SYS and MPU response interface. In this case, the EMIF will perform the write to the SDRAM. However, the ECC value written to the SDRAM will be corrupted. The EMIF will also log the MConnID, MCmd, MBurstSeq, and MAddrSpace for the first error transaction in the EMIF_OCP_ERROR_LOG register .

The ECC quanta is either 32 bit or 16 bit based on the EMIF_SDRAM_CONFIG[15:14] NARROW_MODE bit field. For 32 bit mode (128 bits per EMIF clock cycle), an ECC quanta is 32 bits. For 16 bit narrow mode (64 bits per EMIF clock cycle), the ECC quanta is 16 bits.

Once ECC is enabled, the entire protected region must be initialized with data. These writes must be quanta-sized and quanta-aligned.

The ECC is read and verified during reads . For 1-bit ECC error in the data, the EMIF will correct the data and send it on the SYS or MPU return interface. The EMIF will log the starting address of the SDRAM burst in an internal 4 deep address FIFO. The internal FIFO will store the first four 1-bit ECC errors. The 1-Bit ECC Error Address Log register will display the address on top of the internal FIFO. The software must write a 0x1 to the EMIF_1B_ECC_ERR_ADDR_LOG register to pop the FIFO and display the next address stored. For subsequent reads in the ECC regions, the FIFO will be loaded with the address for the next 1-bit ECC error if it is not full. It must be noted that no address comparison will be performed, that is, if a single address has ECC errors back-to-back, that address will be logged twice.

The number of 1-bit ECC errors can be counted using the EMIF_1B_ECC_ERR_CNT register. The EMIF also supports programming a threshold and a window in the EMIF_1B_ECC_ERR_THRSH register. The window is programmed in number of refresh periods. When the programmed window value is 0x0, that is, window is disabled, and the internal error count meets the programmed threshold, the EMIF will generate a 1-bit ECC error interrupt. When the programmed window value is non-zero, that is, window is enabled, the EMIF will generate a 1-bit ECC error interrupt only if the internal error count meets the programmed threshold in that window. The internal error count is reset every time the window expires. The software can use this to gauge the degree of 1-bit ECC errors occurring in the system.

For diagnosis, the EMIF supports a 1-Bit ECC data error distribution register (EMIF_1B_ECC_ERR_DIST_1) that represent whether an error has occurred in a given data channel location. This is advantageous to detect whether the errors are random or systemic. The distribution registers will be overlay of all 1-bit ECC errors until the software clears the register. Therefore, multiple bits could be set as a result of multiple 1-bit ECC errors occurring over multiple read accesses.

For 2-bit ECC errors in the data, the EMIF will generate a 2-bit ECC error interrupt. For any bit errors in the address, the EMIF will generate an address error interrupt. It must be noted that the EMIF will neither correct the data for these uncorrectable errors. Along with generating the interrupts, the EMIF will also report an error on the SYS or MPU return interface. In this case the EMIF will send the resultant data from the ECC correction logic. The read data received from the memory may have further been corrupted by the ECC correction logic since it will have attempted to correct the read data but failed due to uncorrectable error.

For all uncorrectable ECC errors listed above, the EMIF will log the starting address of the SDRAM burst in the EMIF_2B_ECC_ERR_ADDR_LOG register. This register will show the address of the first uncorrectable error. After the software clears the register, it will be loaded with the address for the next uncorrectable error.

In the event that the EMIF detects a single bit ECC error, although the error is corrected on the returned data, the data in the SDRAM is not corrected. It is the responsibility of the system software to correct the ECC error at that location. To the extent possible, the system software should correct multiple bit errors with the caveat that the returned data was not corrected but corrupted by the ECC correction logic.

15.3.4.15 Class of Service

The class of service mechanism can be enabled by setting to 0x1 the EMIF_READ_WRITE_EXECUTION_THRESHOLD[31] MFLAG_OVERRIDE bit. In this case EMIF services the order of commands given by the class of service rules described in the following paragraphs. If MFLAG_OVERRIDE is set to 0x0 then class of service does not apply. When MFLAG_OVERRIDE is 0x0 and the MFLAG is high then the EMIF alternates between one command coming from the MPU interface and one from the System interface and so forth. The MFLAG value is ignored if MFLAG_OVERRIDE is 0x1.

The commands in the Command FIFO can be mapped to two classes of service namely 1 and 2. The mapping of commands to a particular class of service can be done based on the priority or the master ID. The mapping based on priority can be done by setting the appropriate values in the EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING register. The mapping based on master ID can be done by setting the appropriate values of master ID and the masks in the EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING and EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING registers. There are 3 master ID and mask values that can be set for each class of service. In conjunction with the masks, each class of service can have a maximum of 144 master IDs mapped to it. For example, a master ID value of 0xFF along with a mask value of 0x3 will map all master IDs from 0xF8 to 0xFF to that particular class of service.

Each class of service has an associated latency counter (EMIF_COS_CONFIG[23:16] COS_COUNT_1 and EMIF_COS_CONFIG[15:8] COS_COUNT_2). When the latency counter for a command expires, that is, reaches the value programmed for the class of service that the command belongs to, that command is executed next. If there is more than one command that has expired latency counter, the command with the highest priority is executed first. One exception to this rule is, if the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT value expires for the oldest command in the queue. That command is executed first irrespective of priority or class of service. This is done to prevent the continuous blocking effect.

The EMIF_COS_CONFIG[7:0] PR_OLD_COUNT value is used to identify when the oldest command in the command FIFO has timed out. At this point during the arbitration process, this oldest command is issued regardless of the priority of the other commands in the FIFO. This feature is disabled when writing 0x0 to the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT bit field. After issuing the oldest command, the other remaining commands in the FIFO are reordered by age. The next oldest command in the FIFO is given highest priority again and issued after the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT value expires. If a new value in the PR_OLD_COUNT bit field is written during counting, that is, before PR_OLD_COUNT expires, the counter keeps working but if this value is smaller the oldest command is issued sooner and if this value is larger the oldest command is issued later.

The master ID mapping allows the same master ID to be put in both class of service 1 and 2. Also, a transaction might belong to one class of service if viewed by master ID and might belong to another class of service if viewed by priority. In these cases, the command will belong to both class of service. The EMIF will try executing the command as soon as possible, when the smaller of the two counters (EMIF_COS_CONFIG[23:16] COS_COUNT_1 and EMIF_COS_CONFIG[15:8] COS_COUNT_2) expires.

15.3.4.16 Performance Counters

The EMIF_PERFORMANCE_COUNTER_1 and EMIF_PERFORMANCE_COUNTER_2 registers are used to monitor or calculate the EMIF Controller bandwidth and efficiency. These counters are able to count events such as accesses made to EMIF, Activate (ACT) commands sent to SDRAM, read and write accesses made to EMIF, and other events. Each counter counts independently of the other. In addition to the ability of events counting, the counters can also filter the events from a particular master or address space. The events counting and filter enabling are configured using the EMIF_PERFORMANCE_COUNTER_CONFIG register. The filter value used is configured through the EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT register. Each counter can be configured independently.

Table 15-93 lists all the events that can be counted and whether a filter can be applied to a particular event. A filter is applied to an event if the following bits are set to 0x1 for that event:

- For Performance Counter 1: EMIF_PERFORMANCE_COUNTER_CONFIG[15] CNTR1_MCONNID_EN and EMIF_PERFORMANCE_COUNTER_CONFIG[14] CNTR1_REGION_EN;
- For Performance Counter 2: EMIF_PERFORMANCE_COUNTER_CONFIG[31] CNTR2_MCONNID_EN and EMIF_PERFORMANCE_COUNTER_CONFIG[30] CNTR2_REGION_EN.

Table 15-93. Performance Counter Filter Configuration

CNTRn_CFG[1]	CNTRn_REGION_EN	CNTRn_MCONNID_EN	Description
0x0	0x0	0x0 or 0x1	Count the accesses made to EMIF
0x1	0x0	0x0 or 0x1	Count the Activate (ACT) commands sent to SDRAM
0x2	0x0 or 0x1	0x0 or 0x1	Count the read accesses made to EMIF
0x3	0x0 or 0x1	0x0 or 0x1	Count the write accesses made to EMIF
0x4	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Command FIFO is full
0x5	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Write Data FIFO is full
0x6	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Read Data FIFO is full
0x7	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Return Command FIFO is full
0x8	0x0 or 0x1	0x0 or 0x1	Count number of priority elevations
0x9	0x0	0x0	Count number of EMIF_FICLK clock cycles that a command was pending
0xA	0x0	0x0	Count number of EMIF_FICLK cycles used by the EMIF controller for reads and writes.
0xB - 0xF	0x0	0x0	Reserved for future use.

(1) n = 1 or 2

Note

When the MReqDebug qualifier is set to 0x1 for a particular local command, the performance counters are not incremented for that particular command if the CNTRn_CFG values are equal to 0x0, 0x1, 0x2, 0x3, or 0xA.

Note

The EMIF performance counters cannot distinguish between single access and burst access. In both cases they are incremented by 1. If the actual SDRAM bandwidth of an initiator has to be measured the EMIF performance counters may not be sufficient.

15.3.4.16.1 Performance Counters General Examples

- **General Example for Counting All Write Accesses made to EMIF**
  If the EMIF_PERFORMANCE_COUNTER_1 register is used to count all write accesses made to EMIF from master with connection ID equal to 0x86 (this is the system MMU) the following steps should be performed:
  – To enable the writes counting, the EMIF_PERFORMANCE_COUNTER_CONFIG[3:0] CNTR1_CFG bit field must be set to 0x3.
  – The EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT[15:8] MCONNID1 bit field must be set to 0x86.
  – To enable filtering, the EMIF_PERFORMANCE_COUNTER_CONFIG[15] CNTR1_MCONNID_EN bit must be set to 0x1.

  With this configuration EMIF_PERFORMANCE_COUNTER_1 counts every write made to the EMIF from master 0x86 to any address space. This does not include accesses from other masters and commands other than writes.
- **General Example for Counting Total Accesses made to EMIF**
  If the EMIF_PERFORMANCE_COUNTER_2 register is used to count total accesses made to EMIF regardless of the address space or master the following steps should be performed:

- To enable counting of all accesses to the SDRAM, the EMIF_PERFORMANCE_COUNTER_CONFIG[19:16] CNTR2_CFG bit field must be set to 0x0.
- To disable filtering, both the EMIF_PERFORMANCE_COUNTER_CONFIG[31] CNTR2_MCONNID_EN and EMIF_PERFORMANCE_COUNTER_CONFIG[30] CNTR2_REGION_EN bits must be set to 0x0.

  With this configuration EMIF_PERFORMANCE_COUNTER_2 counts every access made to the EMIF. This includes all accesses from all masters and to any address space.

- **General Example for Counting All Read Accesses made to EMIF**
  If the EMIF_PERFORMANCE_COUNTER_1 register is used to count all read accesses made to EMIF from master with connection ID equal to 0x86 (this is the system MMU) to address space 0x0 the following steps should be performed:
  - To enable the reads counting, the EMIF_PERFORMANCE_COUNTER_CONFIG[3:0] CNTR1_CFG bit field must be set to 0x2.
  - The EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT[15:8] MCONNID1 bit field must be set to 0x86
  - The EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT[1:0] REGION_SEL1 bit field must be set to 0x0.
  - To enable filtering, both the EMIF_PERFORMANCE_COUNTER_CONFIG[15] CNTR1_MCONNID_EN and the EMIF_PERFORMANCE_COUNTER_CONFIG[14] CNTR1_REGION_EN bits must be set to 0x1.

    With this configuration, EMIF_PERFORMANCE_COUNTER_1 counts every read made to the EMIF from master 0x86 to address space 0x0. This does not include accesses from other masters or to other address spaces and does not include commands other than reads.

15.3.4.17 Forcing CKE to tri-state

The CKE pad can be forced to tri-state when the corresponding bit in the CTRL_CORE_SMA_SW_0 register is set to 0x1. For the ddr1_cke pad this is the CTRL_CORE_SMA_SW_0[0] EMIF1_CKE_GATING_CTRL bit and for the ddr2_cke pad the CTRL_CORE_SMA_SW_0[1] EMIF2_CKE_GATING_CTRL bit. This functionality facilitates fast resume by allowing a strong external pull-down resistor to hold the CKE memory pad low thus keeping the SDRAM in self-refresh while the device is completely powered off. Figure 15-51 and the following example sequence provide more details how this functionality can be used:

- When device is running configure EMIF for self refresh. As a result it drives CKE low.
- Disable V_{TT} supply (can be controlled through a device GPIO pad).
- Ramp down all power rails to the device. When power is already off, EMIF doesn't drive CKE low. The external pull-down does this instead. Note that only V_{TT} must be off. All SDRAM supplies (including V_{REF}) must be on.
- After these three steps the device is completely powered off and the SDRAM is in self-refresh. The waking-up from this state is system specific.
- To resume from self-refresh the CKE pad must be forced to tri-state by writing 0x1 to one of the CKE_GATING_CTRL bits previously mentioned. Then EMIF must be configured for self-refresh so that its state becomes consistent with the actual SDRAM state.
- Write 0x0 to the corresponding CKE_GATING_CTRL bit to release CKE driver from tri-state. This allows EMIF to take control over CKE which is now driven low as EMIF has already been configured for self-refresh.
- Enable V_{TT} supply.
- Access the SDRAM to bring it out of self refresh and resume application.

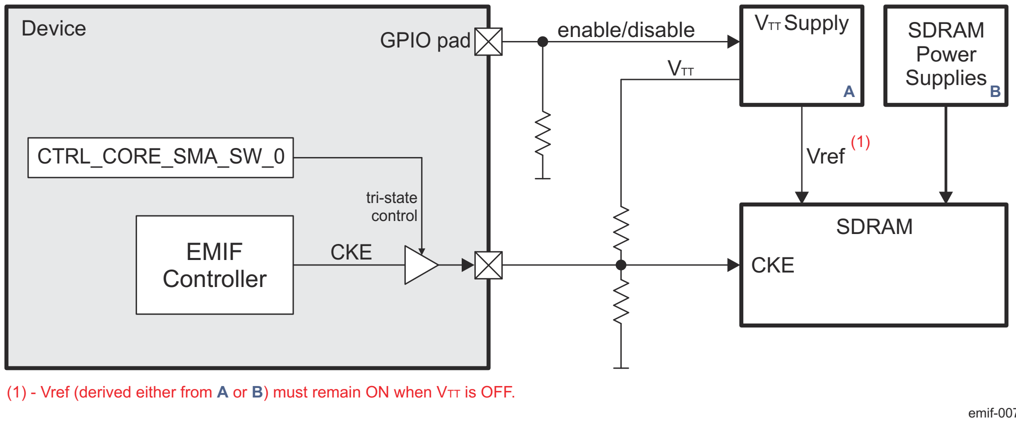


Figure 15-51. Example for Using the CKE Tri-state Functionality

---

Note

In case the device is powered off but the SDRAM is in self-refresh (as previously described) it must be taken into account that the SDRAM RESET# signal has to be controlled externally to preserve the SDRAM contents. This is needed as ddr1_rst/ddr2_rst signal is not controlled by the CTRL_CORE_SMA_SW_0 register and therefore cannot be used.

15.3.5 EMIF Programming Guide

15.3.5.1 EMIF Low-Level Programming Models

This section describers the low-level hardware programming sequences for the configuration and use of the EMIF module.

The following programming sequences are doubled, when the two EMIF modules (EMIF1 and EMIF2) are used.

15.3.5.1.1 Global Initialization

Table 15-94. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. See Chapter 3, *Power, Reset, and Clock Management*.
DMM	The EMIF is a slave to the DMM. The DMM must be enabled and configured. See Section 15.2, *Dynamic Memory Manager*.
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 17, *Interrupt Controllers*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18 , *Control Module*.

15.3.5.1.1.1 EMIF Configuration Sequence

Table 15-95 shows all steps needed to configure and use the EMIF.

Table 15-95. EMIF Configuration Sequence

Step	Register/ Bit Field	Value
Configure DPLL_DDR to the required frequency:		
· **IF** DPLL_DDR is locked:	CM_IDLEST_DPLL_DDR[0] ST_DPLL_CLK	0x1
Unlock DPLL_DDR	CM_CLKMODE_DPLL_DDR[2:0] DPLL_EN	0x6
· **ENDIF**		
· Configure the DPLL multiplier and divider factors	CM_CLKSEL_DPLL_DDR[18:8] DPLL_MULT and CM_CLKSEL_DPLL_DDR[6:0] DPLL_DIV	0x-
· Configure the M2 post-divider factor	CM_DIV_M2_DPLL_DDR[4:0] DIVHS	0x-
· Configure the H11 post-divider factor	CM_DIV_H11_DPLL_DDR[5:0] DIVHS	0x-
· Lock the DPLL_DDR	CM_CLKMODE_DPLL_DDR[2:0] DPLL_EN	0x7
Disable DLL Override	CM_DLL_CTRL[0] DLL_OVERRIDE	0x0
Configure the output impedance, slew rate and weak pull resistors of the DDR IO cells. For more information, see Section 18.4.6.10 *Software Controls for the DDR2/ DDR3 I/O Cells*.	For EMIF1:	0x-
	· CTRL_CORE_CONTROL_DDRCACH1_0	
	· CTRL_CORE_CONTROL_DDRCH1_0	
	· CTRL_CORE_CONTROL_DDRCH1_1	
	· CTRL_CORE_CONTROL_DDRCH1_2	
	For EMIF2:	
	· CTRL_CORE_CONTROL_DDRCACH2_0	
	· CTRL_CORE_CONTROL_DDRCH2_0	
	· CTRL_CORE_CONTROL_DDRCH2_1	

Table 15-95. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
If needed, configure the Vref-Generation Cells. For more information, see Section 18.4.6.11 *Reference Voltage for the Device DDR2/DDR3 Receivers*.	For EMIF1:	0x-
	· CTRL_CORE_CONTROL_DDRIO_0	
	For EMIF2:	
	· CTRL_CORE_CONTROL_DDRIO_1	
Set the number of DQ samples required for read leveling	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT[15:14] EMIF1_REG_PHY_NUM_OF_SAMPLES	0x3
	CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT[15:14] EMIF2_REG_PHY_NUM_OF_SAMPLES	0x3
Choose SDRAM read response on only one DQ bit during read leveling	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT[12] EMIF1_REG_PHY_ALL_DQ_MPR_RD_RESP	0x0
	CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT[12] EMIF2_REG_PHY_ALL_DQ_MPR_RD_RESP	0x0
Configure ODT for the device DDR IOs	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT [6:5] EMIF1_PHY_RD_LOCAL_ODT	0x1 for 60 Ohms
		0x2 for 80 Ohms
		0x3 for 120 Ohms
	CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT [6:5] EMIF2_PHY_RD_LOCAL_ODT	0x1 for 60 Ohms
		0x2 for 80 Ohms
		0x3 for 120 Ohms
IF ECC is used **OR IF** special [(6)] use-cases:		
Enable ECC	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT[16] EMIF1_EN_ECC	0x1
ENDIF		
Based on the memory type other bits from CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT/ CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT can also be configured.		
Read the EMIF_IODFT_TLGC register	EMIF_IODFT_TLGC	RD_VAL_1
Read the EMIF_DDR_PHY_CONTROL_2 register	EMIF_DDR_PHY_CONTROL_2	RD_VAL_2
IF External warm reset or global warm software reset has occurred:	PRM_RSTST[5]EXTERNAL_WARM_RST or	0x1
	PRM_RSTST[1] GLOBAL_WARM_SW_RST	0x1
Reset the DDR PHY	EMIF_IODFT_TLGC[10] RESET_PHY	0x1
ENDIF		
Program the necessary ratio values to the EMIF_EXT_PHY_CONTROL_1 register	EMIF_EXT_PHY_CONTROL_1[19:10] PHY_REG_CTRL_SLAVE_RATIO1	0x80 if PHY_INVERT_CLKOUT = 0x0 or 0x100 if PHY_INVERT_CLKOUT = 0x1
	EMIF_EXT_PHY_CONTROL_1[9:0] PHY_REG_CTRL_SLAVE_RATIO0	0x80 if PHY_INVERT_CLKOUT = 0x0 or 0x100 if PHY_INVERT_CLKOUT = 0x1
Program the shadow register of EMIF_EXT_PHY_CONTROL_1	EMIF_EXT_PHY_CONTROL_1_SHADOW	EMIF_EXT_PHY_CONTROL_1

Table 15-95. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
If software leveling will be used, program registers EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_21 and their corresponding shadow registers **NOTE: Software leveling is not recommended to be used. Hardware leveling must be used instead.**	EMIF_EXT_PHY_CONTROL_2/ EMIF_EXT_PHY_CONTROL_2_SHADOW through EMIF_EXT_PHY_CONTROL_21/ EMIF_EXT_PHY_CONTROL_21_SHADOW	0x-
Program the necessary delay values to the EMIF_EXT_PHY_CONTROL_22 register	EMIF_EXT_PHY_CONTROL_22 [24:16] PHY_REG_FIFO_WE_IN_DELAY (1) (2)	Recommended value is 0x0
	EMIF_EXT_PHY_CONTROL_22 [8:0] PHY_REG_CTRL_SLAVE_DELAY (1) (2)	Recommended value is 0x0
Program the shadow register of EMIF_EXT_PHY_CONTROL_22	EMIF_EXT_PHY_CONTROL_22_SHADOW (1) (2)	EMIF_EXT_PHY_CONTROL_22
Program the necessary delay values to the EMIF_EXT_PHY_CONTROL_23 register	EMIF_EXT_PHY_CONTROL_23[24:16] PHY_REG_WR_DQS_SLAVE_DELAY (1) (2)	Recommended value is 0x0
	EMIF_EXT_PHY_CONTROL_23[8:0] PHY_REG_RD_DQS_SLAVE_DELAY (1) (2)	Recommended value is 0x0
Program the shadow register of EMIF_EXT_PHY_CONTROL_23	EMIF_EXT_PHY_CONTROL_23_SHADOW (1) (2)	EMIF_EXT_PHY_CONTROL_23
Program the EMIF_EXT_PHY_CONTROL_24 register	EMIF_EXT_PHY_CONTROL_24[30:24] REG_PHY_DQ_OFFSET_HI (3)	0x-
	EMIF_EXT_PHY_CONTROL_24[16] REG_PHY_GATELVL_INIT_MODE	Recommended value is 0x1
	EMIF_EXT_PHY_CONTROL_24[12] REG_PHY_USE_RANK0_DELAYS (2)	0x1
	EMIF_EXT_PHY_CONTROL_24[8:0] REG_PHY_WR_DATA_SLAVE_DELAY (1) (2)	Recommended value is 0x0
Program the shadow register of EMIF_EXT_PHY_CONTROL_24	EMIF_EXT_PHY_CONTROL_24_SHADOW	EMIF_EXT_PHY_CONTROL_24
Program the necessary offset ratio values to the EMIF_EXT_PHY_CONTROL_25 register	EMIF_EXT_PHY_CONTROL_25[27:21] REG_PHY_DQ_OFFSET3 (3)	0x-
	EMIF_EXT_PHY_CONTROL_25[20:14] REG_PHY_DQ_OFFSET2 (3)	0x-
	EMIF_EXT_PHY_CONTROL_25[13:7] REG_PHY_DQ_OFFSET1 (3)	0x-
	EMIF_EXT_PHY_CONTROL_25[6:0] REG_PHY_DQ_OFFSET0 (3)	0x-
Program the shadow register of EMIF_EXT_PHY_CONTROL_25	EMIF_EXT_PHY_CONTROL_25_SHADOW	EMIF_EXT_PHY_CONTROL_25
If hardware leveling (read-write leveling) will be used, program with zeros registers EMIF_EXT_PHY_CONTROL_26 through EMIF_EXT_PHY_CONTROL_35 and their corresponding shadow registers	EMIF_EXT_PHY_CONTROL_26/ EMIF_EXT_PHY_CONTROL_26_SHADOW through EMIF_EXT_PHY_CONTROL_35/ EMIF_EXT_PHY_CONTROL_35_SHADOW	0x0
Program the EMIF_EXT_PHY_CONTROL_36 register	EMIF_EXT_PHY_CONTROL_36	0x-
Program the shadow register of EMIF_EXT_PHY_CONTROL_36	EMIF_EXT_PHY_CONTROL_36_SHADOW	EMIF_EXT_PHY_CONTROL_36
Define the SDRAM refresh rate in the shadow register of EMIF_SDRAM_REFRESH_CONTROL	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Define the SDRAM refresh rate	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
Disable SDRAM initialization and refreshes	EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS	0x1
Configure the timing parameters in EMIF_SDRAM_TIMING_1 (4)		0x-
Program the shadow register of EMIF_SDRAM_TIMING_1	EMIF_SDRAM_TIMING_1_SHADOW (4)	EMIF_SDRAM_TIMING_1
Configure the timing parameters in EMIF_SDRAM_TIMING_2 (4)		0x-

Table 15-95. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Program the shadow register of EMIF_SDRAM_TIMING_2	EMIF_SDRAM_TIMING_2_SHADOW (4)	EMIF_SDRAM_TIMING_2
Configure the timing parameters in EMIF_SDRAM_TIMING_3 (4)		0x-
Program the shadow register of EMIF_SDRAM_TIMING_3	EMIF_SDRAM_TIMING_3_SHADOW (4)	EMIF_SDRAM_TIMING_3
Program the EMIF_LPDDR2_NVM_TIMING (5) and EMIF_LPDDR2_NVM_TIMING_SHADOW (5) registers. **NOTE: These registers are not supported. They are kept only for code compatibility.**		0x0
Disable automatic power management	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x0
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Power Down mode when EMIF is idle	EMIF_POWER_MANAGEMENT_CONTROL[15:12] PD_TIM	0x-
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Self Refresh mode when EMIF is idle	EMIF_POWER_MANAGEMENT_CONTROL[7:4] SR_TIM	0x-
Program the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	EMIF_POWER_MANAGEMENT_CONTROL
Configure the System and MPU maximum number of commands in the command FIFO	EMIF_OCP_CONFIG[27:24] SYS_THRESH_MAX	0x-
	EMIF_OCP_CONFIG[23:20] MPU_THRESH_MAX	0x-
Program the EMIF_IODFT_TLGC register	EMIF_IODFT_TLGC	RD_VAL_1
Determine the required wait time after a phy_dll_calib is generated before another command can be sent	EMIF_DLL_CALIB_CTRL[19:16] ACK_WAIT	0x-
Determine the interval between phy_dll_calib generation	EMIF_DLL_CALIB_CTRL[8:0] DLL_CALIB_INTERVAL	0x-
Program the shadow register of EMIF_DLL_CALIB_CTRL	EMIF_DLL_CALIB_CTRL_SHADOW	EMIF_DLL_CALIB_CTRL
Define the interval (number of refresh periods) between ZQCS commands	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[15:0] ZQ_REFINTERVAL	0x-
Define the number of ZQCS intervals that build the ZQCL duration	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[17:16] ZQ_ZQCL_MULT	0x-
Enable issuing of ZQCL on Self- Refresh, Active Power-Down, and Precharge Power-Down exit	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[28] ZQ_SFEXITEN	0x1
Enable ZQ calibration for CS0	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[30] ZQ_CS0EN	0x1
Program the EMIF_TEMP_ALERT_CONFIG (5) register. **NOTE: This register is not supported on this device. It is kept only for code compatibility.**		0x-
Program the EMIF_READ_WRITE_LEVELING_RAMP_WINDOW register. **NOTE: Incremental leveling is not supported on this device.**		0x-
IF Memory type == DDR3 **AND** SDRAM content doesn't need to be maintained:		
Enable read-write leveling	EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVL_EN	0x1
ELSE:		
Disable read-write leveling	EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVL_EN	0x0
ENDIF		
Program bits [30:0] of EMIF_READ_WRITE_LEVELING_RAMP_CONTROL **NOTE: Incremental leveling is not supported on this device.**		0x-
Program the EMIF_READ_WRITE_LEVELING_CONTROL register		0x0

Table 15-95. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Define the read latency for the read data from SDRAM in number of DDR clock cycles	EMIF_DDR_PHY_CONTROL_1[4:0] READ_LATENCY	0x- (typical value >= (CL + 4))
Configure whether the MDLL lock is asserted based on single sample or average of 16 samples	EMIF_DDR_PHY_CONTROL_1[9] PHY_FAST_DLL_LOCK	0x-
Define the maximum number of delay line taps variation while maintaining the master DLL lock. The recommended value is 0x10	EMIF_DDR_PHY_CONTROL_1[17:10] PHY_DLL_LOCK_DIFF	0x10
Configure whether the clock to the SDRAM is inverted or not	EMIF_DDR_PHY_CONTROL_1[18] PHY_INVERT_CLKOUT	0x-
When leveling is used set PHY_DIS_CALIB_RST to 0x0	EMIF_DDR_PHY_CONTROL_1[19] PHY_DIS_CALIB_RST	0x0
Program the slave delay line delays to support 2× mode	EMIF_DDR_PHY_CONTROL_1[21] PHY_HALF_DELAYS	0x1
IF Memory type == DDR3:		
Unmask read data eye training, DQS gate training and write leveling training during full leveling	EMIF_DDR_PHY_CONTROL_1[27] RDLVL_MASK	0x0
	EMIF_DDR_PHY_CONTROL_1[26] RDLVLGATE_MASK	0x0
	EMIF_DDR_PHY_CONTROL_1[25] WRLVL_MASK	0x0
ELSE:		
Mask read data eye training, DQS gate training and write leveling training during full leveling	EMIF_DDR_PHY_CONTROL_1[27] RDLVL_MASK	0x1
	EMIF_DDR_PHY_CONTROL_1[26] RDLVLGATE_MASK	0x1
	EMIF_DDR_PHY_CONTROL_1[25] WRLVL_MASK	0x1
ENDIF		
Program the shadow register of EMIF_DDR_PHY_CONTROL_1	EMIF_DDR_PHY_CONTROL_1_SHADOW	EMIF_DDR_PHY_CONTROL_1
Program the EMIF_DDR_PHY_CONTROL_2 register	EMIF_DDR_PHY_CONTROL_2 (5)	RD_VAL_2
If needed, configure and enable the priority-to-class-of-service mapping for the commands in the command FIFO	EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING	0x-
If needed, configure and enable the master-ID-to-class-of-service mapping for the commands in the command FIFO	EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING	0x-
	EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING	0x-
Chose whether Mflag or Class of Service is used	EMIF_READ_WRITE_EXECUTION_THRESHOLD[31] MFLAG_OVERRIDE	0x-
Configure the write threshold after which EMIF switches to read commands	EMIF_READ_WRITE_EXECUTION_THRESHOLD[12:8] WR_THRSH	0x-
Configure the read threshold after which EMIF switches to write commands	EMIF_READ_WRITE_EXECUTION_THRESHOLD[4:0] RD_THRSH	0x-
Configure the priority rise counters	EMIF_COS_CONFIG	0x-
IF Memory type == DDR3:		
Configure the SDRAM refresh rate with a value of 31.25µs to get 500µs delay between RESET de-assertion to CKE assertion after power-up	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
ELSEIF Memory type == DDR2:		

Table 15-95. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Configure the SDRAM refresh rate with value used initially to get 200μs delay between POWER UP and PRECHARGE ALL command	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
ELSEIF Memory type == LPDDR2:		
Configure the SDRAM refresh rate with value used initially to get 200μs delay between POWER UP and RESET command to LPDDR2	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
ENDIF		
Assign the external bank address bits from lower or higher L3 address as shown in Section 15.3.4.12, *SDRAM Address Mapping*	EMIF_SDRAM_CONFIG_2[27] EBANK_POS	0x-
Select the SDRAM type	EMIF_SDRAM_CONFIG[31:29] SDRAM_TYPE	0x-
Assign internal bank address bits from L3 address as shown in Section 15.3.4.12, *SDRAM Address Mapping*	EMIF_SDRAM_CONFIG[28:27] IBANK_POS	0x-
Choose SDRAM data bus width	EMIF_SDRAM_CONFIG[15:14] NARROW_MODE	0x-
Define CAS latency when accessing connected SDRAM devices.	EMIF_SDRAM_CONFIG[13:10] CL	0x-
Define the number of row address bits of connected SDRAM device.	EMIF_SDRAM_CONFIG[9:7] ROWSIZE	0x-
Define the number of banks inside connected SDRAM device.	EMIF_SDRAM_CONFIG[6:4] IBANK	0x-
Define the internal page size of connected SDRAM device.	EMIF_SDRAM_CONFIG[2:0] PAGESIZE	0x-
Wait 1ms		
Configure the SDRAM refresh rate with value according to the actual memory refresh period requirements	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
IF Memory type == DDR3 **AND** Hardware leveling is used **AND** SDRAM content doesn't need to be maintained:		
IF ECC is used:		
Perform dummy ECC setup just to allow hardware leveling of ECC memories	EMIF_ECC_ADDRESS_RANGE_1	0x0
	EMIF_ECC_ADDRESS_RANGE_2	0x0
	EMIF_ECC_CTRL_REG	0xC000 0000
ENDIF		
Clear the phy_reg_fifo_we_in_misaligned_sticky status flag	EMIF_EXT_PHY_CONTROL_36[8] REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	0x1

Table 15-95. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Temporarily disable SDRAM refreshes	EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS	0x1
Trigger read-write leveling	EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START	0x1
Wait 300µs		
Wait till read-write leveling completes	EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START	0x0
Enable the temporarily disabled SDRAM refreshes	EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS	0x0
Check EMIF_STATUS for leveling timeouts	EMIF_STATUS[6] RDLVLGATETO	0x1
	EMIF_STATUS[5] RDLVLTO	0x1
	EMIF_STATUS[4] WRLVLTO	0x1
IF special [(6)] use-cases:		
Copy EMIF_PHY_STATUS_12 through EMIF_PHY_STATUS_16 to EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_6		
Copy EMIF_PHY_STATUS_7 through EMIF_PHY_STATUS_11 to EMIF_EXT_PHY_CONTROL_7 through EMIF_EXT_PHY_CONTROL_11		
Copy EMIF_PHY_STATUS_17 through EMIF_PHY_STATUS_26 to EMIF_EXT_PHY_CONTROL_12 through EMIF_EXT_PHY_CONTROL_21		
Mask read data eye training, DQS gate training and write leveling training to disable the use of registers EMIF_PHY_STATUS_7 through EMIF_PHY_STATUS_26 registers	EMIF_DDR_PHY_CONTROL_1[27:25]	0x7
Program the shadow register of EMIF_DDR_PHY_CONTROL_1	EMIF_DDR_PHY_CONTROL_1_SHADOW	EMIF_DDR_PHY_CONTROL_1
Disable read-write leveling	EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVLFULL_START	0x0
ENDIF		
Clear the ECC control register	EMIF_ECC_CTRL_REG	0x0
ENDIF		
Configure the LISA_MAP registers based on the EMIF1/EMIF2 usage and EMIF interleaving modes. For more information see Section 15.2.4.2, *Addressing Management with LISA*.	DMM_LISA_MAP_i (i = 0 to 3)	0x-
	MA_LISA_MAP_i (i = 0 to 3)	0x-

(1) These values are only used when CM_DLL_CTRL[0] DLL_OVERRIDE is set to 0x1. This is used only for debug purposes.
(2) The values of these fields do not depend on the board topology.
(3) These fields control the delay between the corresponding DQ byte lane and DQS/DQSN pair. Their values should correspond to delay of 1/4 clock cycle to center the DQ relative to the rising/falling edges of DQS/DQSN. Regardless of topology, it is recommended that all the DQ bits are skew matched to the corresponding DQS/DQSN pair.
(4) Values loaded in these registers depend on OPP.
(5) Writes to these registers do not have any impact. It's up to the user to skip these steps.
(6) In some special use-cases with features like suspend to RAM where EMIF may be reset but SDRAM content needs to be maintained correctly across such an event. In such cases, when EMIF is configured at first boot with hardware leveling enabled, the hardware leveling output from EMIF_PHY_STATUS_x registers is copied to the EMIF_EXT_PHY_CONTROL_x registers. When EMIF comes back from a reset state and needs to be reconfigured without losing SDRAM content, the EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVL_EN bit must be 0 and hardware leveling must not be triggered. In this case, the EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_21 registers take effect. Here it is expected that SDRAM is moved to self-refresh mode before EMIF enters reset state and SDRAM content is maintained correctly by appropriate circuitry external to the SoC.

15.3.5.1.2 Operational Modes Configuration

15.3.5.1.2.1 EMIF Output Impedance Calibration Mode

Table 15-96. EMIF Output Impedance Calibration Mode

Step	Register/ Bit Field / Programming Model	Value
Enable ZQ callibration for CS0.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[30] ZQ_CS0EN	0x1
Define the interval (number of refresh periods) between ZQCS commands.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[15:0] ZQ_REFINTERVAL	0x-
Define the number of ZQCS intervals that build the ZQCL duration.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[17:16] ZQ_ZQCL_MULT	0x-
Enable the issuing of ZQ-Long Command on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[28] ZQ_SFEXITEN	0x1

15.3.5.1.2.2 EMIF SDRAM Self-Refresh

Table 15-97. EMIF SDRAM Self-Refresh Entering

Step	Register/ Bit Field / Programming Model	Value
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Self Refresh mode, when EMIF is idle.	EMIF_POWER_MANAGEMENT_CONTROL[7:4] SR_TIM	0x-
Enable the Self-Refresh mode	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x2
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

Table 15-98. EMIF SDRAM Self-Refresh Exiting

Step	Register/ Bit Field / Programming Model	Value
Change LP_MODE bitfield from 0x2 to any value.	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x-
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

15.3.5.1.2.3 EMIF SDRAM Power-Down Mode

Table 15-99. EMIF SDRAM Power-Down Mode Entering

Step	Register/ Bit Field / Programming Model	Value
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Power Down mode, when EMIF is idle.	EMIF_POWER_MANAGEMENT_CONTROL[15:12] PD_TIM	0x-
Enable (enter) the Power-down mode	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x4
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

Table 15-100. EMIF SDRAM Power-Down Mode Exiting

Step	Register/ Bit Field / Programming Model	Value
Change LP_MODE bitfield from 0x4 to any value.	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x-
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

15.3.5.1.2.4 EMIF ECC Configuration

Table 15-101 lists the ECC configuration settings.

Table 15-101. EMIF ECC Configuration

Step	Register/ Bit Field / Programming Model	Value
Before configuring the EMIF ECC registers, the ECC must be enabled from the Control Module	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT [16] EMIF1_EN_ECC	0x1
Configure the address ranges which have to be ECC protected	EMIF_ECC_ADDRESS_RANGE_1 and EMIF_ECC_ADDRESS_RANGE_2 (1)	0x-
Enable the first ECC protected address range	EMIF_ECC_CTRL_REG[0] REG_ECC_ADDR_RGN_1_EN	0x1
Enable the second ECC protected address range	EMIF_ECC_CTRL_REG[1] REG_ECC_ADDR_RGN_2_EN(1)	0x1
Configure whether the EMIF accesses within the address ranges defined by EMIF_ECC_ADDRESS_RANGE_1 and EMIF_ECC_ADDRESS_RANGE_2 are ECC protected (=0x1) or the EMIF accesses outside these address ranges are ECC protected (=0x0).	EMIF_ECC_CTRL_REG[30] REG_ECC_ADDR_RGN_PROT	0x-
(Optional) Configure the thresholds to generate 1-bit error interrupt	EMIF_1B_ECC_ERR_THRSH	0x-
Enable ECC	EMIF_ECC_CTRL_REG[31] REG_ECC_EN	0x1
Initialize the ECC protected memory regions with quanta-sized and quanta-aligned data		
Clear the status flags and other status history	EMIF_1B_ECC_ERR_CNT	Read the register value and write it back to clear
	EMIF_1B_ECC_ERR_DIST_1	0xFFFF FFFF
	EMIF_2B_ECC_ERR_ADDR_LOG	0x1
	EMIF_SYSTEM_OCP_INTERRUPT_STATUS [5:3]	0x7

(1) It is not mandatory to enable both address ranges. This is provided for flexibility, allowing two non-overlapping ECC protected regions to be created.

Note

If ECC is used the following must be taken into account to avoid unexpected behavior and possible errors:

- The memory interleaving must be disabled through setting the DMM_LISA_MAP_i[19:18] SDRC_INTL bit field to 0x0.
- The xxx_RATIO8 and xxx_RATIO9 bit fields are associated with the ECC data PHY and in case of software leveling must be loaded with values same as in the xxx_RATIO0 through xxx_RATIO7 bit fields from registers EMIF_EXT_PHY_CONTROL_x.

Note

Software leveling is not recommended to be used. Hardware leveling must be used instead.

15.3.6 EMIF Register Manual

Table 15-102 lists the EMIF instance.

15.3.6.1 EMIF Instance Summary

Table 15-102. EMIF Instance Summary

Module Name	Base Address	Size
EMIF1	0x4C00 0000	16 MiB
EMIF2	0x4D00 0000	16 MiB

15.3.6.2 EMIF Registers

Many register values are programmed with full-speed DDR clock cycles, whereas EMIF always runs at half the speed of this clock. Because EMIF is only capable of decrementing these values at half speed (taking into account the minus 1 programming model) the following model applies for data issued on a lower bus:

- REG_VALUE = 0 → 2 DDR clocks
- REG_VALUE = 1 → 2 DDR clocks
- REG_VALUE = 2 → 4 DDR clocks
- REG_VALUE = 3 → 4 DDR clocks
- REG_VALUE = 4 → 6 DDR clocks
- REG_VALUE = 5 → 6 DDR clocks
- REG_VALUE = 6 → 8 DDR clocks
- REG_VALUE = 7 → 8 DDR clocks

Activate and deactivate commands use the upper bus. Register values associated with these operations follow the general rule:

- REG_VALUE = 0 → 1 DDR clocks
- REG_VALUE = 1 → 2 DDR clocks
- REG_VALUE = 2 → 2 DDR clocks
- REG_VALUE = 3 → 4 DDR clocks
- REG_VALUE = 4 → 4 DDR clocks
- REG_VALUE = 5 → 6 DDR clocks
- REG_VALUE = 6 → 6 DDR clocks

Note

Shadow registers are loaded on any frequency change or SidleReq/SidleAck transition at the point where the EMIF has put SDRAM into self-refresh mode.

Table 15-103 summarizes the EMIF register mapping.

15.3.6.2.1 EMIF Register Summary

Table 15-103. EMIF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMIF1 Physical Address	EMIF2 Physical Address
EMIF_REVISION	R	32	0x0000 0000	0x4C00 0000	0x4D00 0000
EMIF_STATUS	R	32	0x0000 0004	0x4C00 0004	0x4D00 0004
EMIF_SDRAM_CONFIG	RW	32	0x0000 0008	0x4C00 0008	0x4D00 0008
EMIF_SDRAM_CONFIG_2	RW	32	0x0000 000C	0x4C00 000C	0x4D00 000C
EMIF_SDRAM_REFRESH_CONTROL	RW	32	0x0000 0010	0x4C00 0010	0x4D00 0010
EMIF_SDRAM_REFRESH_CONTROL_SHADOW	RW	32	0x0000 0014	0x4C00 0014	0x4D00 0014
EMIF_SDRAM_TIMING_1	RW	32	0x0000 0018	0x4C00 0018	0x4D00 0018
EMIF_SDRAM_TIMING_1_SHADOW	RW	32	0x0000 001C	0x4C00 001C	0x4D00 001C

Table 15-103. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF1 Physical Address	EMIF2 Physical Address
EMIF_SDRAM_TIMING_2	RW	32	0x0000 0020	0x4C00 0020	0x4D00 0020
EMIF_SDRAM_TIMING_2_SHADOW	RW	32	0x0000 0024	0x4C00 0024	0x4D00 0024
EMIF_SDRAM_TIMING_3	RW	32	0x0000 0028	0x4C00 0028	0x4D00 0028
EMIF_SDRAM_TIMING_3_SHADOW	RW	32	0x0000 002C	0x4C00 002C	0x4D00 002C
EMIF_LPDDR2_NVM_TIMING	RW	32	0x0000 0030	0x4C00 0030	0x4D00 0030
EMIF_LPDDR2_NVM_TIMING_SHADOW	RW	32	0x0000 0034	0x4C00 0034	0x4D00 0034
EMIF_POWER_MANAGEMENT_CONTROL	RW	32	0x0000 0038	0x4C00 0038	0x4D00 0038
EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	RW	32	0x0000 003C	0x4C00 003C	0x4D00 003C
RESERVED	R	32	0x0000 0040	0x4C00 0040	0x4D00 0040
RESERVED	R	32	0x0000 0050	0x4C00 0050	0x4D00 0050
EMIF_OCP_CONFIG	RW	32	0x0000 0054	0x4C00 0054	0x4D00 0054
EMIF_OCP_CONFIG_VALUE_1	R	32	0x0000 0058	0x4C00 0058	0x4D00 0058
EMIF_OCP_CONFIG_VALUE_2	R	32	0x0000 005C	0x4C00 005C	0x4D00 005C
EMIF_IODFT_TLGC	RW	32	0x0000 0060	0x4C00 0060	0x4D00 0060
EMIF_PERFORMANCE_COUNTER_1	R	32	0x0000 0080	0x4C00 0080	0x4D00 0080
EMIF_PERFORMANCE_COUNTER_2	R	32	0x0000 0084	0x4C00 0084	0x4D00 0084
EMIF_PERFORMANCE_COUNTER_CONFIG	RW	32	0x0000 0088	0x4C00 0088	0x4D00 0088
EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT	RW	32	0x0000 008C	0x4C00 008C	0x4D00 008C
EMIF_PERFORMANCE_COUNTER_TIME	R	32	0x0000 0090	0x4C00 0090	0x4D00 0090
EMIF_MISC_REG	RW	32	0x0000 0094	0x4C00 0094	0x4D00 0094
EMIF_DLL_CALIB_CTRL	RW	32	0x0000 0098	0x4C00 0098	0x4D00 0098
EMIF_DLL_CALIB_CTRL_SHADOW	RW	32	0x0000 009C	0x4C00 009C	0x4D00 009C
EMIF_END_OF_INTERRUPT	RW	32	0x0000 00A0	0x4C00 00A0	0x4D00 00A0
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS	RW	32	0x0000 00A4	0x4C00 00A4	0x4D00 00A4
RESERVED	R	32	0x0000 00A8	0x4C00 00A8	0x4D00 00A8
EMIF_SYSTEM_OCP_INTERRUPT_STATUS	RW	32	0x0000 00AC	0x4C00 00AC	0x4D00 00AC
RESERVED	R	32	0x0000 00B0	0x4C00 00B0	0x4D00 00B0
EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET	RW	32	0x0000 00B4	0x4C00 00B4	0x4D00 00B4
RESERVED	R	32	0x0000 00B8	0x4C00 00B8	0x4D00 00B8
EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR	RW	32	0x0000 00BC	0x4C00 00BC	0x4D00 00BC
RESERVED	R	32	0x0000 00C0	0x4C00 00C0	0x4D00 00C0
EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG	RW	32	0x0000 00C8	0x4C00 00C8	0x4D00 00C8
EMIF_TEMP_ALERT_CONFIG	RW	32	0x0000 00CC	0x4C00 00CC	0x4D00 00CC
EMIF_OCP_ERROR_LOG	R	32	0x0000 00D0	0x4C00 00D0	0x4D00 00D0
EMIF_READ_WRITE_LEVELING_RAMP_WINDOW	RW	32	0x0000 00D4	0x4C00 00D4	0x4D00 00D4
EMIF_READ_WRITE_LEVELING_RAMP_CONTROL	RW	32	0x0000 00D8	0x4C00 00D8	0x4D00 00D8
EMIF_READ_WRITE_LEVELING_CONTROL	RW	32	0x0000 00DC	0x4C00 00DC	0x4D00 00DC
EMIF_DDR_PHY_CONTROL_1	RW	32	0x0000 00E4	0x4C00 00E4	0x4D00 00E4
EMIF_DDR_PHY_CONTROL_1_SHADOW	RW	32	0x0000 00E8	0x4C00 00E8	0x4D00 00E8
EMIF_DDR_PHY_CONTROL_2	RW	32	0x0000 00EC	0x4C00 00EC	0x4D00 00EC
EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING	RW	32	0x0000 0100	0x4C00 0100	0x4D00 0100
EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING	RW	32	0x0000 0104	0x4C00 0104	0x4D00 0104

Table 15-103. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF1 Physical Address	EMIF2 Physical Address
EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING	RW	32	0x0000 0108	0x4C00 0108	0x4D00 0108
EMIF_ECC_CTRL_REG	RW	32	0x0000 0110	0x4C00 0110	N/A
EMIF_ECC_ADDRESS_RANGE_1	RW	32	0x0000 0114	0x4C00 0114	N/A
EMIF_ECC_ADDRESS_RANGE_2	RW	32	0x0000 0118	0x4C00 0118	N/A
EMIF_READ_WRITE_EXECUTION_THRESHOLD	RW	32	0x0000 0120	0x4C00 0120	0x4D00 0120
EMIF_COS_CONFIG	RW	32	0x0000 0124	0x4C00 0124	0x4D00 0124
EMIF_1B_ECC_ERR_CNT	RW	32	0x0000 0130	0x4C00 0130	N/A
EMIF_1B_ECC_ERR_THRSH	RW	32	0x0000 0134	0x4C00 0134	N/A
EMIF_1B_ECC_ERR_DIST_1	RW	32	0x0000 0138	0x4C00 0138	N/A
EMIF_1B_ECC_ERR_ADDR_LOG	RW	32	0x0000 013C	0x4C00 013C	N/A
EMIF_2B_ECC_ERR_ADDR_LOG	RW	32	0x0000 0140	0x4C00 0140	N/A
EMIF_PHY_STATUS_1	R	32	0x0000 0144	0x4C00 0144	0x4D00 0144
EMIF_PHY_STATUS_2	R	32	0x0000 0148	0x4C00 0148	0x4D00 0148
EMIF_PHY_STATUS_3	R	32	0x0000 014C	0x4C00 014C	0x4D00 014C
EMIF_PHY_STATUS_4	R	32	0x0000 0150	0x4C00 0150	0x4D00 0150
EMIF_PHY_STATUS_5	R	32	0x0000 0154	0x4C00 0154	0x4D00 0154
EMIF_PHY_STATUS_6	R	32	0x0000 0158	0x4C00 0158	0x4D00 0158
EMIF_PHY_STATUS_7	R	32	0x0000 015C	0x4C00 015C	0x4D00 015C
EMIF_PHY_STATUS_8	R	32	0x0000 0160	0x4C00 0160	0x4D00 0160
EMIF_PHY_STATUS_9	R	32	0x0000 0164	0x4C00 0164	0x4D00 0164
EMIF_PHY_STATUS_10	R	32	0x0000 0168	0x4C00 0168	0x4D00 0168
EMIF_PHY_STATUS_11	R	32	0x0000 016C	0x4C00 016C	N/A
EMIF_PHY_STATUS_12	R	32	0x0000 0170	0x4C00 0170	0x4D00 0170
EMIF_PHY_STATUS_13	R	32	0x0000 0174	0x4C00 0174	0x4D00 0174
EMIF_PHY_STATUS_14	R	32	0x0000 0178	0x4C00 0178	0x4D00 0178
EMIF_PHY_STATUS_15	R	32	0x0000 017C	0x4C00 017C	0x4D00 017C
EMIF_PHY_STATUS_16	R	32	0x0000 0180	0x4C00 0180	N/A
EMIF_PHY_STATUS_17	R	32	0x0000 0184	0x4C00 0184	0x4D00 0184
EMIF_PHY_STATUS_18	R	32	0x0000 0188	0x4C00 0188	0x4D00 0188
EMIF_PHY_STATUS_19	R	32	0x0000 018C	0x4C00 018C	0x4D00 018C
EMIF_PHY_STATUS_20	R	32	0x0000 0190	0x4C00 0190	0x4D00 0190
EMIF_PHY_STATUS_21	R	32	0x0000 0194	0x4C00 0194	N/A
EMIF_PHY_STATUS_22	R	32	0x0000 0198	0x4C00 0198	0x4D00 0198
EMIF_PHY_STATUS_23	R	32	0x0000 019C	0x4C00 019C	0x4D00 019C
EMIF_PHY_STATUS_24	R	32	0x0000 01A0	0x4C00 01A0	0x4D00 01A0
EMIF_PHY_STATUS_25	R	32	0x0000 01A4	0x4C00 01A4	0x4D00 01A4
EMIF_PHY_STATUS_26	R	32	0x0000 01A8	0x4C00 01A8	N/A
EMIF_PHY_STATUS_27	R	32	0x0000 01AC	0x4C00 01AC	0x4D00 01AC
EMIF_PHY_STATUS_28	R	32	0x0000 01B0	0x4C00 01B0	0x4D00 01B0
EMIF_EXT_PHY_CONTROL_1	RW	32	0x0000 0200	0x4C00 0200	0x4D00 0200
EMIF_EXT_PHY_CONTROL_1_SHADOW	RW	32	0x0000 0204	0x4C00 0204	0x4D00 0204
EMIF_EXT_PHY_CONTROL_2	RW	32	0x0000 0208	0x4C00 0208	0x4D00 0208
EMIF_EXT_PHY_CONTROL_2_SHADOW	RW	32	0x0000 020C	0x4C00 020C	0x4D00 020C
EMIF_EXT_PHY_CONTROL_3	RW	32	0x0000 0210	0x4C00 0210	0x4D00 0210

Table 15-103. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF1 Physical Address	EMIF2 Physical Address
EMIF_EXT_PHY_CONTROL_3_SHADOW	RW	32	0x0000 0214	0x4C00 0214	0x4D00 0214
EMIF_EXT_PHY_CONTROL_4	RW	32	0x0000 0218	0x4C00 0218	0x4D00 0218
EMIF_EXT_PHY_CONTROL_4_SHADOW	RW	32	0x0000 021C	0x4C00 021C	0x4D00 021C
EMIF_EXT_PHY_CONTROL_5	RW	32	0x0000 0220	0x4C00 0220	0x4D00 0220
EMIF_EXT_PHY_CONTROL_5_SHADOW	RW	32	0x0000 0224	0x4C00 0224	0x4D00 0224
EMIF_EXT_PHY_CONTROL_6	RW	32	0x0000 0228	0x4C00 0228	N/A
EMIF_EXT_PHY_CONTROL_6_SHADOW	RW	32	0x0000 022C	0x4C00 022C	N/A
EMIF_EXT_PHY_CONTROL_7	RW	32	0x0000 0230	0x4C00 0230	0x4D00 0230
EMIF_EXT_PHY_CONTROL_7_SHADOW	RW	32	0x0000 0234	0x4C00 0234	0x4D00 0234
EMIF_EXT_PHY_CONTROL_8	RW	32	0x0000 0238	0x4C00 0238	0x4D00 0238
EMIF_EXT_PHY_CONTROL_8_SHADOW	RW	32	0x0000 023C	0x4C00 023C	0x4D00 023C
EMIF_EXT_PHY_CONTROL_9	RW	32	0x0000 0240	0x4C00 0240	0x4D00 0240
EMIF_EXT_PHY_CONTROL_9_SHADOW	RW	32	0x0000 0244	0x4C00 0244	0x4D00 0244
EMIF_EXT_PHY_CONTROL_10	RW	32	0x0000 0248	0x4C00 0248	0x4D00 0248
EMIF_EXT_PHY_CONTROL_10_SHADOW	RW	32	0x0000 024C	0x4C00 024C	0x4D00 024C
EMIF_EXT_PHY_CONTROL_11	RW	32	0x0000 0250	0x4C00 0250	N/A
EMIF_EXT_PHY_CONTROL_11_SHADOW	RW	32	0x0000 0254	0x4C00 0254	N/A
EMIF_EXT_PHY_CONTROL_12	RW	32	0x0000 0258	0x4C00 0258	0x4D00 0258
EMIF_EXT_PHY_CONTROL_12_SHADOW	RW	32	0x0000 025C	0x4C00 025C	0x4D00 025C
EMIF_EXT_PHY_CONTROL_13	RW	32	0x0000 0260	0x4C00 0260	0x4D00 0260
EMIF_EXT_PHY_CONTROL_13_SHADOW	RW	32	0x0000 0264	0x4C00 0264	0x4D00 0264
EMIF_EXT_PHY_CONTROL_14	RW	32	0x0000 0268	0x4C00 0268	0x4D00 0268
EMIF_EXT_PHY_CONTROL_14_SHADOW	RW	32	0x0000 026C	0x4C00 026C	0x4D00 026C
EMIF_EXT_PHY_CONTROL_15	RW	32	0x0000 0270	0x4C00 0270	0x4D00 0270
EMIF_EXT_PHY_CONTROL_15_SHADOW	RW	32	0x0000 0274	0x4C00 0274	0x4D00 0274
EMIF_EXT_PHY_CONTROL_16	RW	32	0x0000 0278	0x4C00 0278	N/A
EMIF_EXT_PHY_CONTROL_16_SHADOW	RW	32	0x0000 027C	0x4C00 027C	N/A
EMIF_EXT_PHY_CONTROL_17	RW	32	0x0000 0280	0x4C00 0280	0x4D00 0280
EMIF_EXT_PHY_CONTROL_17_SHADOW	RW	32	0x0000 0284	0x4C00 0284	0x4D00 0284
EMIF_EXT_PHY_CONTROL_18	RW	32	0x0000 0288	0x4C00 0288	0x4D00 0288
EMIF_EXT_PHY_CONTROL_18_SHADOW	RW	32	0x0000 028C	0x4C00 028C	0x4D00 028C
EMIF_EXT_PHY_CONTROL_19	RW	32	0x0000 0290	0x4C00 0290	0x4D00 0290
EMIF_EXT_PHY_CONTROL_19_SHADOW	RW	32	0x0000 0294	0x4C00 0294	0x4D00 0294
EMIF_EXT_PHY_CONTROL_20	RW	32	0x0000 0298	0x4C00 0298	0x4D00 0298
EMIF_EXT_PHY_CONTROL_20_SHADOW	RW	32	0x0000 029C	0x4C00 029C	0x4D00 029C
EMIF_EXT_PHY_CONTROL_21	RW	32	0x0000 02A0	0x4C00 02A0	N/A
EMIF_EXT_PHY_CONTROL_21_SHADOW	RW	32	0x0000 02A4	0x4C00 02A4	N/A
EMIF_EXT_PHY_CONTROL_22	RW	32	0x0000 02A8	0x4C00 02A8	0x4D00 02A8
EMIF_EXT_PHY_CONTROL_22_SHADOW	RW	32	0x0000 02AC	0x4C00 02AC	0x4D00 02AC
EMIF_EXT_PHY_CONTROL_23	RW	32	0x0000 02B0	0x4C00 02B0	0x4D00 02B0
EMIF_EXT_PHY_CONTROL_23_SHADOW	RW	32	0x0000 02B4	0x4C00 02B4	0x4D00 02B4
EMIF_EXT_PHY_CONTROL_24	RW	32	0x0000 02B8	0x4C00 02B8	0x4D00 02B8
EMIF_EXT_PHY_CONTROL_24_SHADOW	RW	32	0x0000 02BC	0x4C00 02BC	0x4D00 02BC
EMIF_EXT_PHY_CONTROL_25	RW	32	0x0000 02C0	0x4C00 02C0	0x4D00 02C0
EMIF_EXT_PHY_CONTROL_25_SHADOW	RW	32	0x0000 02C4	0x4C00 02C4	0x4D00 02C4

Table 15-103. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF1 Physical Address	EMIF2 Physical Address
EMIF_EXT_PHY_CONTROL_26	RW	32	0x0000 02C8	0x4C00 02C8	0x4D00 02C8
EMIF_EXT_PHY_CONTROL_26_SHADOW	RW	32	0x0000 02CC	0x4C00 02CC	0x4D00 02CC
EMIF_EXT_PHY_CONTROL_27	RW	32	0x0000 02D0	0x4C00 02D0	0x4D00 02D0
EMIF_EXT_PHY_CONTROL_27_SHADOW	RW	32	0x0000 02D4	0x4C00 02D4	0x4D00 02D4
EMIF_EXT_PHY_CONTROL_28	RW	32	0x0000 02D8	0x4C00 02D8	0x4D00 02D8
EMIF_EXT_PHY_CONTROL_28_SHADOW	RW	32	0x0000 02DC	0x4C00 02DC	0x4D00 02DC
EMIF_EXT_PHY_CONTROL_29	RW	32	0x0000 02E0	0x4C00 02E0	0x4D00 02E0
EMIF_EXT_PHY_CONTROL_29_SHADOW	RW	32	0x0000 02E4	0x4C00 02E4	0x4D00 02E4
EMIF_EXT_PHY_CONTROL_30	RW	32	0x0000 02E8	0x4C00 02E8	N/A
EMIF_EXT_PHY_CONTROL_30_SHADOW	RW	32	0x0000 02EC	0x4C00 02EC	N/A
EMIF_EXT_PHY_CONTROL_31	RW	32	0x0000 02F0	0x4C00 02F0	0x4D00 02F0
EMIF_EXT_PHY_CONTROL_31_SHADOW	RW	32	0x0000 02F4	0x4C00 02F4	0x4D00 02F4
EMIF_EXT_PHY_CONTROL_32	RW	32	0x0000 02F8	0x4C00 02F8	0x4D00 02F8
EMIF_EXT_PHY_CONTROL_32_SHADOW	RW	32	0x0000 02FC	0x4C00 02FC	0x4D00 02FC
EMIF_EXT_PHY_CONTROL_33	RW	32	0x0000 0300	0x4C00 0300	0x4D00 0300
EMIF_EXT_PHY_CONTROL_33_SHADOW	RW	32	0x0000 0304	0x4C00 0304	0x4D00 0304
EMIF_EXT_PHY_CONTROL_34	RW	32	0x0000 0308	0x4C00 0308	0x4D00 0308
EMIF_EXT_PHY_CONTROL_34_SHADOW	RW	32	0x0000 030C	0x4C00 030C	0x4D00 030C
EMIF_EXT_PHY_CONTROL_35	RW	32	0x0000 0310	0x4C00 0310	N/A
EMIF_EXT_PHY_CONTROL_35_SHADOW	RW	32	0x0000 0314	0x4C00 0314	N/A
EMIF_EXT_PHY_CONTROL_36	RW	32	0x0000 0318	0x4C00 0318	0x4D00 0318
EMIF_EXT_PHY_CONTROL_36_SHADOW	RW	32	0x0000 031C	0x4C00 031C	0x4D00 031C

15.3.6.2.2 EMIF Register Description

Table 15-104. EMIF_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4C00 0000 0x4D00 0000	**Instance**	EMIF1 EMIF2
Description	Revision number register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Module revision	R	0x- [(1)]

(1) TI internal data

Table 15-105. Register Call Summary for Register EMIF_REVISION

EMIF Controller

- Arbitration of Commands in the Command FIFO: [0] [1]
- EMIF Register Summary: [2]

Table 15-106. EMIF_STATUS

Address Offset	0x0000 0004		
Physical Address	0x4C00 0004 0x4D00 0004	**Instance**	EMIF1 EMIF2

Table 15-106. EMIF_STATUS (continued)

Description	SDRAM Status Register (STATUS)
Type	R

31	30	29	28:7	6	5	4	3	2	1:0
BE	DUAL_CLK_MODE	FAST_INIT	RESERVED	RDLVLGATETO	RDLVLTO	WRLVLTO	RESERVED	PHY_DLL_READY	RESERVED

Bits	Field Name	Description	Type	Reset
31	BE	Big endian mode select for 8 and 16-bit devices, set to 1 for big endian or 0 for little endian operation. In current implementation, only 32-bit devices are supported - this bit is don't care.	R	0
30	DUAL_CLK_MODE	Dual Clock mode. Defines whether the EMIFi_L3_ICLK and EMIF_FICLK clock are asynchronous. EMIFi_L3_ICLK and EMIF_FICLK clock are asynchronous, if set to 1.	R	0
29	FAST_INIT	Fast Init. Defines whether the EMIF fast initialization mode has been enabled. Fast initialization is enabled if set to 1.	R	0
28:7	RESERVED	Reserved	R	0x00 0000
6	RDLVLGATETO	Read DQS Gate Training Timeout. Value of 1 indicates read DQS gate training has timed out because read DQS gate training done was not received from the PHY.	R	0
5	RDLVLTO	Read Data Eye Training Timeout. Value of 1 indicates read data eye training has timed out because read data eye training done was not received from the PHY.	R	0
4	WRLVLTO	Write Leveling Timeout. Value of 1 indicates write leveling has timed out because write leveling done was not received from the PHY.	R	0
3	RESERVED		R	0
2	PHY_DLL_READY	DDR PHY Ready. The DDR PHY is ready for normal operation, if set to 1.	R	0
1:0	RESERVED	Reserved	R	0x0

Table 15-107. Register Call Summary for Register EMIF_STATUS

EMIF Controller

- Global Initialization: [0] [1] [2] [3]
- EMIF Register Summary: [4]

Table 15-108. EMIF_SDRAM_CONFIG

Address Offset	0x0000 0008		
Physical Address	0x4C00 0008 0x4D00 0008	**Instance**	EMIF1 EMIF2
Description	SDRAM Config Register. A write to this register will cause the EMIF to start the SDRAM initialization sequence. **CAUTION:** This register is loaded with values by control module at device reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SDRAM_TYPE	IBANK_POS	DDR_TERM	DDR2_DDQS	DYN_ODT	DDR_DISABLE_DLL	SDRAM_DRIVE	CWL	NARROW_MODE	CL	ROWSIZE	IBANK	RESERVED	PAGESIZE

Bits	Field Name	Description	Type	Reset
31:29	SDRAM_TYPE	SDRAM Type selection. This field is loaded from e-fuse. Set to 2 for DDR2 Set to 3 for DDR3 All other values are reserved.	RW	0x0
28:27	IBANK_POS	Internal bank position. See Section 15.3.4.12, *SDRAM Address Mapping*.	RW	0x0
26:24	DDR_TERM	DDR3 termination resistor value. Set to 0 to disable termination. For DDR3, set to 1 for RZQ/4, set to 2 for RZQ/2, set to 3 for RZQ/6, set to 4 for RZQ/12, and set to 5 for RZQ/8. All other values are reserved.	RW	0x0
23	DDR2_DDQS	DDR2 differential DDQS enable. NOT SUPPORTED. Set to 1 for compatibility.	RW	0
22:21	DYN_ODT	DDR3 Dynamic ODT. NOT SUPPORTED. Set to 0 to turn off dynamic ODT.	RW	0x0
20	DDR_DISABLE_DLL	Disable DLL select. Set to 1 to disable DLL inside SDRAM.	RW	0
19:18	SDRAM_DRIVE	SDRAM drive strength.For DDR3, set to 0 for RZQ/6 and set to 1 for RZQ/7. All other values are reserved.	RW	0x0
17:16	CWL	DDR3 CAS Write latency. Value of 0, 1, 2, and 3 (CAS write latency of 5, 6, 7, and 8) are supported. Use the lowest value supported for best performance. All other values are reserved.	RW	0x0
15:14	NARROW_MODE	SDRAM data bus width. Set to 0 for 32-bit data bus width. Set to 1 for 16-bit data bus width. All other values are reserved.	RW	0x0
13:10	CL	CAS Latency (referred to as read latency (RL) in some SDRAM specs). The value of this field defines the CAS latency to be used when accessing connected SDRAM devices. Values of 2, 3, 4 and 5 (CAS latency of 2, 3, 4 and 5) are supported for DDR2. Values of 2, 4, 6, 8, 10, 12 and 14 (CAS latency of 5, 6, 7, 8, 9, 10 and 11) are supported for DDR3. All other values are reserved.	RW	0x0

Bits	Field Name	Description	Type	Reset
9:7	ROWSIZE	Row Size. Defines the number of row address bits of connected SDRAM devices. Set to 0 for 9 row bits, Set to 1 for 10 row bits, Set to 2 for 11 row bits, Set to 3 for 12 row bits, Set to 4 for 13 row bits, Set to 5 for 14 row bits, Set to 6 for 15 row bits, Set to 7 for 16 row bits. This field is only used when EMIF_SDRAM_CONFIG[28:27] IBANK_POS field is set to 1, 2, or 3 or EBANK_POS field in EMIF_SDRAM_CONFIG_2 register is set to 1.	RW	0x0
6:4	IBANK	Internal Bank setup. Defines number of banks inside connected SDRAM devices. Set to 0 for 1 bank, Set to 1 for 2 banks, Set to 2 for 4 banks, Set to 3 for 8 banks. All other values are reserved.	RW	0x0
3	RESERVED		R	0
2:0	PAGESIZE	Page Size. Defines the internal page size of connected SDRAM devices. Set to 0 for 256-word page (8 column bits), Set to 1 for 512-word page (9 column bits), Set to 2 for 1024-word page (10 column bits), Set to 3 for 2048-word page (11 column bits). All other values are reserved.	RW	0x0

Table 15-109. Register Call Summary for Register EMIF_SDRAM_CONFIG

EMIF Controller

- Arbitration of Commands in the Command FIFO: [0]
- Self-Refresh Mode: [1]
- SDRAM Initialization: [2] [3]
- DDR2 SDRAM Initialization: [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]
- DDR3 SDRAM Initialization: [15] [16] [17] [18] [19] [20] [21] [22]
- SDRAM Address Mapping: [23] [24] [25] [26] [27] [28] [29]
- Address Mapping for IBANK_POS = 0 and EBANK_POS = 0: [30] [31]
- Address Mapping for IBANK_POS = 1 and EBANK_POS = 0: [32] [33]
- Address Mapping for IBANK_POS = 2 and EBANK_POS = 0: [34] [35]
- Address Mapping for IBANK_POS = 3 and EBANK_POS = 0: [36]
- Address Mapping for IBANK_POS = 0 and EBANK_POS = 1: [37]
- Address Mapping for IBANK_POS = 1 and EBANK_POS = 1: [38]
- Address Mapping for IBANK_POS = 2 and EBANK_POS = 1: [39]
- Address Mapping for IBANK_POS = 3 and EBANK_POS = 1: [40]
- Error Correction And Detection Feature: [41]
- Global Initialization: [42] [43] [44] [45] [46] [47] [48]
- EMIF Register Summary: [49]
- EMIF Register Description: [50]

Table 15-110. EMIF_SDRAM_CONFIG_2

Address Offset	0x0000 000C		
Physical Address	0x4C00 000C 0x4D00 000C	**Instance**	EMIF1 EMIF2
Description	SDRAM Config Register 2 **CAUTION:** This register is loaded with values by control module at device reset.		

Table 15-110. EMIF_SDRAM_CONFIG_2 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				EBANK_POS	RESERVED																										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	EBANK_POS	External bank position. Set to 0 to assign external bank address bits from lower OCP address. Set to 1 to assign external bank address bits from higher OCP address bits. See Section 15.3.4.12, *SDRAM Address Mapping*.	RW	0
26:0	RESERVED		R	0x00 0000

Table 15-111. Register Call Summary for Register EMIF_SDRAM_CONFIG_2

EMIF Controller

- Address Mapping for IBANK_POS = 0 and EBANK_POS = 0: [0]
- Address Mapping for IBANK_POS = 0 and EBANK_POS = 1: [1]
- Address Mapping for IBANK_POS = 1 and EBANK_POS = 1: [2]
- Address Mapping for IBANK_POS = 2 and EBANK_POS = 1: [3]
- Address Mapping for IBANK_POS = 3 and EBANK_POS = 1: [4]
- Global Initialization: [5]
- EMIF Register Summary: [6]
- EMIF Register Description: [7]

Table 15-112. EMIF_SDRAM_REFRESH_CONTROL

Address Offset	0x0000 0010		
Physical Address	0x4C00 0010 0x4D00 0010	**Instance**	EMIF1 EMIF2
Description	SDRAM Refresh Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITREF_DIS	RESERVED	SRT	ASR	RESERVED	PASR			RESERVED								REFRESH_RATE															

Bits	Field Name	Description	Type	Reset
31	INITREF_DIS	Initialization and Refresh disable. When set to 1, EMIF will disable SDRAM initialization and refreshes, but will carry out SDRAM write/read transactions.	RW	1
30	RESERVED		R	0
29	SRT	DDR3 Self Refresh temperature range. Set to 0 for normal operating temperature range and set to 1 for extended operating temperature range when the ASR field is set to 0. This bit must be set to 0 if the ASR field is set to 1. A write to this field will cause the EMIF to start the SDRAM initialization sequence.	RW	0

Bits	Field Name	Description	Type	Reset
28	ASR	DDR3 Auto Self Refresh enable. Set to 1 for auto Self Refresh enable. Set to 0 for manual Self Refresh reference indicated by the SRT field. A write to this field will cause the EMIF to start the SDRAM initialization sequence.	RW	0
27	RESERVED		R	0
26:24	PASR	Partial Array Self Refresh. These bits get loaded into the Extended Mode Register of DDR3 during initialization. For DDR3, set to 0 for full array, set to 1 or 5 for 1/2 array, set to 2 or 6 for 1/4 array, set to 3 or 7 for 1/8 array, and set to 4 for 3/4 array to be refreshed. All other values are reserved. A write to this field will cause the EMIF to start the SDRAM initialization sequence.	RW	0x0
23:16	RESERVED		R	0x00
15:0	REFRESH_RATE	Refresh Rate. Value in this field is used to define the rate at which connected SDRAM devices will be refreshed. SDRAM refresh rate = REFRESH_RATE / EMIF_PHY_FCLK. A 533-MHz DDR clock rate system that requires a 7.8 µs refresh rate would need 7.8 × 533 = 4157 or 0x103D value to be written. To avoid lock-up situations, the programmer must not program REFRESH_RATE < (6 × EMIF_SDRAM_TIMING_3[12:4] T_RFC). **Note** **NOTE:** The SDRAM refresh rate can be changed on-the-fly by writing to this field. When changing the SDRAM refresh rate all timing parameters that use the refresh rate value have to be recalculated. For example, tRASmax specified in the EMIF_SDRAM_TIMING_3[3:0] T_RAS_MAX field must be recalculated.	RW	0x0000

Table 15-113. Register Call Summary for Register EMIF_SDRAM_REFRESH_CONTROL

EMIF Controller

- Arbitration of Commands in the Command FIFO: [0]
- Reset: [1]
- Self-Refresh Mode: [2]
- SDRAM Refresh Scheduling: [3] [4] [5]
- DDR2 SDRAM Initialization: [6] [7] [8] [9] [10]
- DDR3 SDRAM Initialization: [11] [12] [13] [14] [15] [16] [17] [18]
- Global Initialization: [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31]
- EMIF Register Summary: [32]
- EMIF Register Description: [33] [34] [35] [36] [37] [38] [39]

Table 15-114. EMIF_SDRAM_REFRESH_CONTROL_SHADOW

Address Offset	0x0000 0014		
Physical Address	0x4C00 0014 0x4D00 0014	**Instance**	EMIF1 EMIF2
Description	SDRAM Refresh Control Shadow Register		

Table 15-114. EMIF_SDRAM_REFRESH_CONTROL_SHADOW (continued)

Type	RW

31:16	15:0
RESERVED	REFRESH_RATE_SHDW

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	REFRESH_RATE_SHDW	Shadow field for REFRESH_RATE. This field is loaded into EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE field when SIdleAck is asserted.	RW	0x0000

Table 15-115. Register Call Summary for Register EMIF_SDRAM_REFRESH_CONTROL_SHADOW

EMIF Controller

- Global Initialization: [0] [1] [2] [3] [4]
- EMIF Register Summary: [5]

Table 15-116. EMIF_SDRAM_TIMING_1

Address Offset	0x0000 0018		
Physical Address	0x4C00 0018 0x4D00 0018	**Instance**	EMIF1 EMIF2
Description	SDRAM Timing 1 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
Type	RW		

31:29	28:25	24:21	20:17	16:12	11:6	5:3	2:0
T_RTW	T_RP	T_RCD	T_WR	T_RAS	T_RC	T_RRD	T_WTR

Bits	Field Name	Description	Type	Reset
31:29	T_RTW	Minimum number of DDR clock cycles between Read to Write data phases, minus one.	RW	0x0
28:25	T_RP	Minimum number of DDR clock cycles from Precharge to Activate or Refresh, minus one.	RW	0x0
24:21	T_RCD	Minimum number of DDR clock cycles from Activate to Read or Write, minus one.	RW	0x0
20:17	T_WR	Minimum number of DDR clock cycles from last Write transfer to Precharge, minus one.	RW	0x0
16:12	T_RAS	Minimum number of DDR clock cycles from Activate to Precharge, minus one. T_RAS value needs to be bigger than or equal to T_RDC value.	RW	0x00
11:6	T_RC	Minimum number of DDR clock cycles from Activate to Activate, minus one.	RW	0x00
5:3	T_RRD	Minimum number of DDR clock cycles from Activate to Activate for a different bank, minus one. For an 8-bank, this field must be equal to ((tFAW / (4 × tCK)) - 1).	RW	0x0
2:0	T_WTR	Minimum number of DDR clock cycles from last Write to Read, minus one.	RW	0x0

Table 15-117. Register Call Summary for Register EMIF_SDRAM_TIMING_1

EMIF Controller

- DDR2 SDRAM Initialization: [0] [1] [2]
- DDR3 SDRAM Initialization: [3] [4] [5]
- Turnaround Time: [6] [7]
- Global Initialization: [8] [9] [10]
- EMIF Register Summary: [11]
- EMIF Register Description: [12] [13] [14] [15] [16] [17] [18] [19]

Table 15-118. EMIF_SDRAM_TIMING_1_SHADOW

Address Offset	0x0000 001C		
Physical Address	0x4C00 001C 0x4D00 001C	**Instance**	EMIF1 EMIF2
Description	SDRAM Timing 1 Shadow Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_RTW_SHDW			T_RP_SHDW				T_RCD_SHDW				T_WR_SHDW				T_RAS_SHDW					T_RC_SHDW						T_RRD_SHDW			T_WTR_SHDW		

Bits	Field Name	Description	Type	Reset
31:29	T_RTW_SHDW	Shadow field for T_RTW. This field is loaded into EMIF_SDRAM_TIMING_1[31:29] T_RTW field when SIdleAck is asserted.	RW	0x0
28:25	T_RP_SHDW	Shadow field for T_RP. This field is loaded into EMIF_SDRAM_TIMING_1[28:25] T_RP field when SIdleAck is asserted.	RW	0x0
24:21	T_RCD_SHDW	Shadow field for T_RCD. This field is loaded into EMIF_SDRAM_TIMING_1[24:21] T_RCD field when SIdleAck is asserted.	RW	0x0
20:17	T_WR_SHDW	Shadow field for T_WR. This field is loaded into EMIF_SDRAM_TIMING_1[20:17] T_WR field when SIdleAck is asserted.	RW	0x0
16:12	T_RAS_SHDW	Shadow field for T_RAS. This field is loaded into EMIF_SDRAM_TIMING_1[16:12] T_RAS field when SIdleAck is asserted.	RW	0x00
11:6	T_RC_SHDW	Shadow field for T_RC. This field is loaded into EMIF_SDRAM_TIMING_1[11:6] T_RC field when SIdleAck is asserted.	RW	0x00
5:3	T_RRD_SHDW	Shadow field for T_RRD. This field is loaded into EMIF_SDRAM_TIMING_1[5:3] T_RRD field when SIdleAck is asserted.	RW	0x0
2:0	T_WTR_SHDW	Shadow field for T_WTR. This field is loaded into EMIF_SDRAM_TIMING_1[2:0] T_WTR field when SIdleAck is asserted.	RW	0x0

Table 15-119. Register Call Summary for Register EMIF_SDRAM_TIMING_1_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-120. EMIF_SDRAM_TIMING_2

Address Offset	0x0000 0020		
Physical Address	0x4C00 0020 0x4D00 0020	**Instance**	EMIF1 EMIF2

Table 15-120. EMIF_SDRAM_TIMING_2 (continued)

Description	SDRAM Timing 2 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.
Type	RW

31	30:28	27:25	24:16	15:6	5:3	2:0
RESERVED	T_XP	RESERVED	T_XSNR	T_XSRD	T_RTP	T_CKE

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	T_XP	Minimum number of DDR clock cycles from power-down exit to any command other than a read command, minus one.	RW	0x0
27:25	RESERVED	Reserved	RW	0x0
24:16	T_XSNR	Minimum number of DDR clock cycles from Self-Refresh exit to any command other than a Read command, minus one.	RW	0x000
15:6	T_XSRD	Minimum number of DDR clock cycles from Self-Refresh exit to a Read command, minus one.	RW	0x000
5:3	T_RTP	Minimum number of DDR clock cycles for the last read command to a Precharge command, minus one.	RW	0x0
2:0	T_CKE	Minimum number of DDR clock cycles between CKE pin changes, minus one.	RW	0x0

Table 15-121. Register Call Summary for Register EMIF_SDRAM_TIMING_2

EMIF Controller

- Power-Down Mode: [0] [1]
- Self-Refresh Mode: [2] [3] [4] [5] [6]
- Global Initialization: [7] [8] [9]
- EMIF Register Summary: [10]
- EMIF Register Description: [11] [12] [13] [14] [15]

Table 15-122. EMIF_SDRAM_TIMING_2_SHADOW

Address Offset	0x0000 0024		
Physical Address	0x4C00 0024 0x4D00 0024	**Instance**	EMIF1 EMIF2
Description	SDRAM Timing 2 Shadow Register		
Type	RW		

31	30:28	27:25	24:16	15:6	5:3	2:0
RESERVED	T_XP_SHDW	RESERVED	T_XSNR_SHDW	T_XSRD_SHDW	T_RTP_SHDW	T_CKE_SHDW

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	T_XP_SHDW	Shadow field for T_XP. This field is loaded into EMIF_SDRAM_TIMING_2[30:28] T_XP field when SIdleAck is asserted.	RW	0x0
27:25	RESERVED	Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
24:16	T_XSNR_SHDW	Shadow field for T_XSNR. This field is loaded into EMIF_SDRAM_TIMING_2[24:16] T_XSNR field when SIdleAck is asserted.	RW	0x000
15:6	T_XSRD_SHDW	Shadow field for T_XSRD. This field is loaded into EMIF_SDRAM_TIMING_2[15:6] T_XSRD field when SIdleAck is asserted.	RW	0x000
5:3	T_RTP_SHDW	Shadow field for T_RTP. This field is loaded into EMIF_SDRAM_TIMING_2[5:3] T_RTP field when SIdleAck is asserted.	RW	0x0
2:0	T_CKE_SHDW	Shadow field for T_CKE. This field is loaded into EMIF_SDRAM_TIMING_2[2:0] T_CKE field when SIdleAck is asserted.	RW	0x0

Table 15-123. Register Call Summary for Register EMIF_SDRAM_TIMING_2_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-124. EMIF_SDRAM_TIMING_3

Address Offset	0x0000 0028		
Physical Address	0x4C00 0028 0x4D00 0028	**Instance**	EMIF1 EMIF2
Description	SDRAM Timing 3 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
Type	RW		

31:28	27:24	23:21	20:15	14:13	12:4	3:0
T_PDLL_UL	RESERVED	T_CKESR	ZQ_ZQCS	RESERVED	T_RFC	T_RAS_MAX

Bits	Field Name	Description	Type	Reset
31:28	T_PDLL_UL	Minimum number of DDR clock cycles for PHY DLL to unlock. A value of N will be equal to N x 128 clocks.	RW	0x0
27:24	RESERVED		R	0x0
23:21	T_CKESR	Minimum number of DDR clock cycles for which SDRAM must remain in Self Refresh, minus one.	RW	0x0
20:15	ZQ_ZQCS	Number of DDR clock cycles for a ZQCS command, minus one.	RW	0x00
14:13	RESERVED		R	0x0
12:4	T_RFC	Minimum number of DDR clock cycles from Refresh or Load Mode to Refresh or Activate, minus one.	RW	0x000
3:0	T_RAS_MAX	Maximum number of REFRESH_RATE intervals from Activate to Precharge command. This field must be equal to ((tRASmax / tREFI)-1) rounded down to the next lower integer. Value for T_RAS_MAX can be calculated as follows: If tRASmax = 120 us and tREFI = 15.7 us, then T_RAS_MAX = ((120/15.7)-1) = 6.64. Round down to the next lower integer. Therefore, the programmed value must be 6.	RW	0x0

Table 15-125. Register Call Summary for Register EMIF_SDRAM_TIMING_3

EMIF Controller

- SDRAM Refresh Scheduling: [0]
- DDR3 Output Impedance Calibration: [1] [2]
- Global Initialization: [3] [4] [5]
- EMIF Register Summary: [6]
- EMIF Register Description: [7] [8] [9] [10] [11] [12] [13] [14]

Table 15-126. EMIF_SDRAM_TIMING_3_SHADOW

Address Offset	0x0000 002C		
Physical Address	0x4C00 002C 0x4D00 002C	**Instance**	EMIF1 EMIF2
Description	SDRAM Timing 3 Shadow Register		
Type	RW		

31:28	27:24	23:21	20:15	14:13	12:4	3:0
T_PDLL_UL_SHDW	RESERVED	T_CKESR_SHDW	ZQ_ZQCS_SHDW	RESERVED	T_RFC_SHDW	T_RAS_MAX_SHDW

Bits	Field Name	Description	Type	Reset
31:28	T_PDLL_UL_SHDW	Shadow field for T_PDLL_UL. This field is loaded into T_PDLL_UL field in EMIF_SDRAM_TIMING_3 register when SIdleAck is asserted.	RW	0x0
27:24	RESERVED		R	0x0
23:21	T_CKESR_SHDW	Shadow field for T_CKESR. This field is loaded into T_CKESR field in EMIF_SDRAM_TIMING_3 register when SIdleAck is asserted.	RW	0x0
20:15	ZQ_ZQCS_SHDW	Shadow field for ZQ_ZQCS. This field is loaded into ZQ_ZQCS field in EMIF_SDRAM_TIMING_3 register when SIdleAck is asserted.	RW	0x00
14:13	RESERVED		R	0x0
12:4	T_RFC_SHDW	Shadow field for T_RFC. This field is loaded into EMIF_SDRAM_TIMING_3[12:4] T_RFC when SIdleAck is asserted.	RW	0x000
3:0	T_RAS_MAX_SHDW	Shadow field for T_RAS_MAX. This field is loaded into EMIF_SDRAM_TIMING_3[3:0] T_RAS_MAX field when SIdleAck is asserted.	RW	0x0

Table 15-127. Register Call Summary for Register EMIF_SDRAM_TIMING_3_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-128. EMIF_LPDDR2_NVM_TIMING

Address Offset	0x0000 0030		
Physical Address	0x4C00 0030 0x4D00 0030	**Instance**	EMIF1 EMIF2
Description	**NOTE: This register is not supported. It is kept only for code compatibility.**		
Type	RW		

31:0
RESERVED

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0

Table 15-129. Register Call Summary for Register EMIF_LPDDR2_NVM_TIMING

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-130. EMIF_LPDDR2_NVM_TIMING_SHADOW

Address Offset	0x0000 0034		
Physical Address	0x4C00 0034 0x4D00 0034	**Instance**	EMIF1 EMIF2
Description	**NOTE: This register is not supported. It is kept only for code compatibility.**		
Type	RW		

31–0
RESERVED

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0

Table 15-131. Register Call Summary for Register EMIF_LPDDR2_NVM_TIMING_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-132. EMIF_POWER_MANAGEMENT_CONTROL

Address Offset	0x0000 0038		
Physical Address	0x4C00 0038 0x4D00 0038	**Instance**	EMIF1 EMIF2
Description	Power Management Control Register. Updating the *_TIM fields must be followed by at least one access to SDRAM for the new value to take an effect.		
Type	RW		

31–16	15–12	11	10–8	7–4	3–0
RESERVED	PD_TIM	RESERVED	LP_MODE	SR_TIM	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PD_TIM	Power Management timer for Power-Down. The EMIF will put the external SDRAM in Power-Down mode after the EMIF is idle for these number of DDR clock cycles and if LP_MODE field is set to 4. Set to 0 to immediately enter Power-Down mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.	RW	0x0
11	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
10:8	LP_MODE	Automatic Power Management enable. 0x0: Disable automatic power management 0x1: Reserved 0x2: Self Refresh mode 0x3: Disable automatic power management 0x4: Power-Down mode All other values disable automatic power management.	RW	0x0
7:4	SR_TIM	Power Management timer for Self Refresh. The EMIF will put the external SDRAM in Self Refresh mode after the EMIF is idle for these number of DDR clock cycles and if LP_MODE field is set to 2. Set to 0 to immediately enter Self Refresh mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.	RW	0x0
3:0	RESERVED		RW	0x0

Table 15-133. Register Call Summary for Register EMIF_POWER_MANAGEMENT_CONTROL

EMIF Controller

- Reset: [0]
- Power-Down Mode: [1] [2] [3]
- Self-Refresh Mode: [4] [5] [6]
- Global Initialization: [7] [8] [9] [10] [11]
- Operational Modes Configuration: [12] [13] [14] [15] [16] [17] [18] [19] [20] [21]
- EMIF Register Summary: [22]
- EMIF Register Description: [23] [24]

Table 15-134. EMIF_POWER_MANAGEMENT_CONTROL_SHADOW

Address Offset	0x0000 003C		
Physical Address	0x4C00 003C 0x4D00 003C	**Instance**	EMIF1 EMIF2
Description	Power Management Control Shadow Register		
Type	RW		

31–16	15–12	11–8	7–4	3–0
RESERVED	PD_TIM_SHDW	RESERVED	SR_TIM_SHDW	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PD_TIM_SHDW	Shadow field for PD_TIM. This field is loaded into PD_TIM field in EMIF_POWER_MANAGEMENT_CONTROL register when SIdleAck is asserted.	RW	0x0
11:8	RESERVED		R	0x0
7:4	SR_TIM_SHDW	Shadow field for SR_TIM. This field is loaded into SR_TIM field in EMIF_POWER_MANAGEMENT_CONTROL register when SIdleAck is asserted.	RW	0x0
3:0	RESERVED		RW	0x0

Table 15-135. Register Call Summary for Register EMIF_POWER_MANAGEMENT_CONTROL_SHADOW

EMIF Controller

- Global Initialization: [0]
- Operational Modes Configuration: [1] [2] [3] [4]
- EMIF Register Summary: [5]

Table 15-136. EMIF_OCP_CONFIG

Address Offset	0x0000 0054		
Physical Address	0x4C00 0054 0x4D00 0054	**Instance**	EMIF1 EMIF2
Description	OCP Config Register		
Type	RW		

31:28	27:24	23:20	19:0
RESERVED	SYS_THRESH_MAX	MPU_THRESH_MAX	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	SYS_THRESH_MAX	System OCP Threshold Maximum. The number of commands the system interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1.	RW	0x7
23:20	MPU_THRESH_MAX	MPU Threshold Maximum. The number of commands the MPU interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1.	RW	0x7
19:0	RESERVED		R	0x70000

Table 15-137. Register Call Summary for Register EMIF_OCP_CONFIG

EMIF Controller

- FIFO Description: [0] [1]
- Global Initialization: [2] [3]
- EMIF Register Summary: [4]

Table 15-138. EMIF_OCP_CONFIG_VALUE_1

Address Offset	0x0000 0058		
Physical Address	0x4C00 0058 0x4D00 0058	**Instance**	EMIF1 EMIF2
Description	OCP Config Value 1 Register		
Type	R		

31:30	29:16	15:8	7:0
SYS_BUS_WIDTH	RESERVED	WR_FIFO_DEPTH	CMD_FIFO_DEPTH

Bits	Field Name	Description	Type	Reset
31:30	SYS_BUS_WIDTH	System OCP data bus width 0 = 32-bit wide, 1 = 64-bit wide, 2 = 128-bit wide, 3 = Reserved	R	0x2
29:16	RESERVED		R	0x1000
15:8	WR_FIFO_DEPTH	Write Data FIFO depth	R	0x19
7:0	CMD_FIFO_DEPTH	Command FIFO depth	R	0x0A

Table 15-139. Register Call Summary for Register EMIF_OCP_CONFIG_VALUE_1

EMIF Controller

- EMIF Register Summary: [0]

Table 15-140. EMIF_OCP_CONFIG_VALUE_2

Address Offset	0x0000 005C		
Physical Address	0x4C00 005C 0x4D00 005C	**Instance**	EMIF1 EMIF2
Description	OCP Config Value 2 Register		
Type	R		

31:24	23:16	15:8	7:0
RESERVED	RREG_FIFO_DEPTH	RSD_FIFO_DEPTH	RCMD_FIFO_DEPTH

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:16	RREG_FIFO_DEPTH	Register Read Data FIFO depth	R	0x04
15:8	RSD_FIFO_DEPTH	SDRAM Read Data FIFO depth	R	0x27
7:0	RCMD_FIFO_DEPTH	Read Command FIFO depth	R	0x27

Table 15-141. Register Call Summary for Register EMIF_OCP_CONFIG_VALUE_2

EMIF Controller

- EMIF Register Summary: [0]

Table 15-142. EMIF_IODFT_TLGC

Address Offset	0x0000 0060		
Physical Address	0x4C00 0060 0x4D00 0060Added registers pertaining to DDR configuration.	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31:16	15	14	13	12	11	10	9	8	7:6	5:4	3:1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESET_PHY	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved. This field must not be modified.	RW	0x0
15	RESERVED	Reserved	R	0x0
14	RESERVED	Reserved. This bit must not be modified.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	RESERVED	Reserved. This bit must not be modified.	RW	0x1
12	RESERVED	Reserved. This bit must not be modified.	RW	0x0
11	RESERVED	Reserved	R	0x0
10	RESET_PHY	Reset the DDR PHY. Writing 1 to this bit resets the DDR PHY. This bit will self clear to 0.	RW	0x0
9	RESERVED	Reserved	R	0x0
8	RESERVED	Reserved. This bit must not be modified.	RW	0x0
7:6	RESERVED	Reserved	R	0x0
5:4	RESERVED	Reserved. This field must not be modified.	RW	0x1
3:1	RESERVED	Reserved. This field must not be modified.	RW	0x0
0	RESERVED	Reserved. This bit must not be modified.	RW	0x1

Table 15-143. Register Call Summary for Register EMIF_IODFT_TLGC

EMIF Controller

- Global Initialization: [0] [1] [2] [3] [4]
- EMIF Register Summary: [5]

Table 15-144. EMIF_PERFORMANCE_COUNTER_1

Address Offset	0x0000 0080		
Physical Address	0x4C00 0080 0x4D00 0080	**Instance**	EMIF1 EMIF2
Description	Performance Counter 1 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER1																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER1	32-bit counter that can be configured as specified in the EMIF_PERFORMANCE_COUNTER_CONFIG register and EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT register.	R	0x0000 0000

Table 15-145. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_1

EMIF Controller

- Performance Counters: [0]
- Performance Counters General Examples: [1] [2] [3] [4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6] [7] [8] [9] [10]

Table 15-146. EMIF_PERFORMANCE_COUNTER_2

Address Offset	0x0000 0084		
Physical Address	0x4C00 0084 0x4D00 0084	**Instance**	EMIF1 EMIF2
Description	Performance Counter 2 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER2																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER2	32-bit counter that can be configured as specified in the EMIF_PERFORMANCE_COUNTER_CONFIG register and EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT register.	R	0x0000 0000

Table 15-147. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_2

EMIF Controller

- Performance Counters: [0]
- Performance Counters General Examples: [1] [2]
- EMIF Register Summary: [3]
- EMIF Register Description: [4] [5] [6] [7] [8]

Table 15-148. EMIF_PERFORMANCE_COUNTER_CONFIG

Address Offset	0x0000 0088		
Physical Address	0x4C00 0088 0x4D00 0088	**Instance**	EMIF1 EMIF2
Description	Performance Counter Config Register		
Type	RW		

31	30	29:20	19:16	15	14	13:4	3:0
CNTR2_MCONNID_EN	CNTR2_REGION_EN	RESERVED	CNTR2_CFG	CNTR1_MCONNID_EN	CNTR1_REGION_EN	RESERVED	CNTR1_CFG

Bits	Field Name	Description	Type	Reset
31	CNTR2_MCONNID_EN	MConnID filter enable for EMIF_PERFORMANCE_COUNTER_2 register.	RW	0
30	CNTR2_REGION_EN	Chip Select filter enable for EMIF_PERFORMANCE_COUNTER_2 register.	RW	0
29:20	RESERVED	Reserved for future use	R	0x000
19:16	CNTR2_CFG	Filter configuration for EMIF_PERFORMANCE_COUNTER_2. Refer to Table 15-93 for details.	RW	0x1
15	CNTR1_MCONNID_EN	MConnID filter enable for EMIF_PERFORMANCE_COUNTER_1 register.	RW	0
14	CNTR1_REGION_EN	Chip Select filter enable for EMIF_PERFORMANCE_COUNTER_1 register.	RW	0
13:4	RESERVED	Reserved for future use	R	0x000
3:0	CNTR1_CFG	Filter configuration for EMIF_PERFORMANCE_COUNTER_1. Refer to Table 15-93 for details.	RW	0x0

Table 15-149. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_CONFIG

EMIF Controller

- Performance Counters: [0] [1] [2] [3] [4]
- Performance Counters General Examples: [5] [6] [7] [8] [9] [10] [11] [12]
- EMIF Register Summary: [13]
- EMIF Register Description: [14] [15]

Table 15-150. EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT

Address Offset	0x0000 008C		
Physical Address	0x4C00 008C 0x4D00 008C	**Instance**	EMIF1 EMIF2
Description	Performance Counter Master Region Select Register The values programmed into the MCONNIDx fields are those in the *ConnID Values* table in Chapter 14, *Interconnect*.		
Type	RW		

31:24	23:18	17:16	15:8	7:2	1:0
MCONNID2	RESERVED	REGION_SEL2	MCONNID1	RESERVED	REGION_SEL1

Bits	Field Name	Description	Type	Reset
31:24	MCONNID2	MConnID for EMIF_PERFORMANCE_COUNTER_2 register.	RW	0x00
23:18	RESERVED	Reserved	R	0x00
17:16	REGION_SEL2	MAddrSpace for EMIF_PERFORMANCE_COUNTER_2 register.	RW	0x0
15:8	MCONNID1	MConnID for EMIF_PERFORMANCE_COUNTER_1 register.	RW	0x00
7:2	RESERVED	Reserved	R	0x00
1:0	REGION_SEL1	MAddrSpace for EMIF_PERFORMANCE_COUNTER_1 register.	RW	0x0

Table 15-151. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT

EMIF Controller

- Performance Counters: [0]
- Performance Counters General Examples: [1] [2] [3]
- EMIF Register Summary: [4]
- EMIF Register Description: [5] [6]

Table 15-152. EMIF_PERFORMANCE_COUNTER_TIME

Address Offset	0x0000 0090		
Physical Address	0x4C00 0090 0x4D00 0090	**Instance**	EMIF1 EMIF2
Description	Performance Counter Time Register. This is a free running counter.		
Type	R		

31:0
TOTAL_TIME

Bits	Field Name	Description	Type	Reset
31:0	TOTAL_TIME	32-bit counter that continuously counts number for EMIF_FICLK clock cycles elapsed after EMIF is brought out of reset.	R	0x0000 0000

Table 15-153. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_TIME

EMIF Controller

- EMIF Register Summary: [0]

Table 15-154. EMIF_MISC_REG

Address Offset	0x0000 0094		
Physical Address	0x4C00 0094 0x4D00 0094	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31–1	0
RESERVED	DLL_CALIB_OS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	DLL_CALIB_OS	Phy_dll_calib one shot : Setting bit to 1 generates a phy_pll_calib pulse. Bit is self cleared when pll_calib gets generated and ack_wait has been satisfied. Software can poll to confirm completion. Uses the EMIF_DLL_CALIB_CTRL[19:16] ACK_WAIT bit field for time to wait after firing off the phy_dll_calib.	RW	0x0

Table 15-155. Register Call Summary for Register EMIF_MISC_REG

EMIF Controller

- PHY DLL Calibration: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 15-156. EMIF_DLL_CALIB_CTRL

Address Offset	0x0000 0098		
Physical Address	0x4C00 0098 0x4D00 0098	**Instance**	EMIF1 EMIF2
Description	Control register to force idle window time to generate a phy_dll_calib that can be used for updating PHY DLLs during voltage ramps. NOTE: Should always be loaded via the shadow register.		
Type	RW		

31–20	19–16	15–9	8–0
RESERVED	ACK_WAIT	RESERVED	DLL_CALIB_INTERVAL

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000

Bits	Field Name	Description	Type	Reset
19:16	ACK_WAIT	The ack_wait determines the required wait time after a phy_dll_calib is generated before another command can be sent. Value program is in terms of EMIF_FICLK cycle count. **CAUTION:** 5 must be the minimum value ever programmed.	RW	0x9
15:9	RESERVED		R	0x00
8:0	DLL_CALIB_INTERVAL	This field determines the interval between phy_dll_calib generation. This value is multiplied by a precounter of 16 EMIF_FICLK cycles. Program this field one less the value you are targeting; program 1 to achieve interval of 2 (minimum interval supported). Programming zero turns off function. Note the final intervals between dll_calib generation is also a function of ACK_WAIT. Final periodic interval is calculated by: ((DLL_CALIB_INTERVAL + 1) × 16) + ACK_WAIT	RW	0x000

Table 15-157. Register Call Summary for Register EMIF_DLL_CALIB_CTRL

EMIF Controller

- PHY DLL Calibration: [0]
- Global Initialization: [1] [2] [3] [4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6] [7] [8]

Table 15-158. EMIF_DLL_CALIB_CTRL_SHADOW

Address Offset	0x0000 009C		
Physical Address	0x4C00 009C 0x4D00 009C	**Instance**	EMIF1 EMIF2
Description	Read Idle Control Shadow Register		
Type	RW		

31:20	19:16	15:9	8:0
RESERVED	ACK_WAIT_SHDW	RESERVED	DLL_CALIB_INTERVAL_SHDW

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:16	ACK_WAIT_SHDW	Shadow field for ACK_WAIT. This field is loaded into ACK_WAIT field in EMIF_DLL_CALIB_CTRL register when SIdleAck is asserted	RW	0x9
15:9	RESERVED		R	0x00
8:0	DLL_CALIB_INTERVAL_SHDW	Shadow field for DLL_CALIB_INTERVAL. This field is loaded into DLL_CALIB_INTERVAL field in the EMIF_DLL_CALIB_CTRL register when SIdleAck is asserted	RW	0x000

Table 15-159. Register Call Summary for Register EMIF_DLL_CALIB_CTRL_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-160. EMIF_END_OF_INTERRUPT

Address Offset	0x0000 00A0		
Physical Address	0x4C00 00A0 0x4D00 00A0	**Instance**	EMIF1 EMIF2

Table 15-160. EMIF_END_OF_INTERRUPT (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EOI

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI	Software End Of Interrupt (EOI) control. Write 0x0 for system OCP interrupt. This field always reads 0 (no EOI memory).	RW	0x0

Table 15-161. Register Call Summary for Register EMIF_END_OF_INTERRUPT

EMIF Controller

- EMIF Register Summary: [0]

Table 15-162. EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS

Address Offset	0x0000 00A4		
Physical Address	0x4C00 00A4 0x4D00 00A4	**Instance**	EMIF1 EMIF2
Description	System OCP Interrupt Raw Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED		ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Raw status of system ECC one bit error correction interrupt.	RW	0x0
4	TWOBIT_ECC_ERR_SYS	Raw status of system ECC two bit error detection interrupt.	RW	0x0
3	WR_ECC_ERR_SYS	Raw status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location .	RW	0x0
2:1	RESERVED		R	0x0
0	ERR_SYS	Raw status of system OCP interrupt for command or address error. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.	RW	0

Table 15-163. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS

EMIF Controller

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]
- EMIF Register Summary: [12]

Table 15-164. EMIF_SYSTEM_OCP_INTERRUPT_STATUS

Address Offset	0x0000 00AC		
Physical Address	0x4C00 00AC 0x4D00 00AC	**Instance**	EMIF1 EMIF2
Description	System OCP Interrupt Status Register		
Type	RW		

31:6	5	4	3	2:1	0
RESERVED	ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED	ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Enabled status of system ECC one bit error correction interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is even if not enabled). Writing a 0 has no effect	RW	0x0
4	TWOBIT_ECC_ERR_SYS	Enabled status of system ECC two bit error detection interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is even if not enabled). Writing a 0 has no effect.	RW	0x0
3	WR_ECC_ERR_SYS	Enabled status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location . Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is even if not enabled). Writing a 0 has no effect.	RW	0x0
2:1	RESERVED		R	0x0
0	ERR_SYS	Enabled status of system OCP interrupt interrupt for command or address error. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Writing a 0 has no effect.	RW	0

Table 15-165. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_STATUS

EMIF Controller

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
- Operational Modes Configuration: [11]
- EMIF Register Summary: [12]
- EMIF Register Description: [13]

Table 15-166. EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET

Address Offset	0x0000 00B4

Table 15-166. EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET (continued)

Physical Address	0x4C00 00B4 0x4D00 00B4	**Instance**	EMIF1 EMIF2
Description	System OCP Interrupt Enable Set Register		
Type	RW		

31–6	5	4	3	2–1	0
RESERVED	ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED	EN_ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Enabled status of sysem ECC one bit error correction interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0x0
4	TWOBIT_ECC_ERR_SYS	Enabled status of system ECC two bit error detection interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0x0
3	WR_ECC_ERR_SYS	Enabled status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location . Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0x0
2:1	RESERVED		R	0x0
0	EN_ERR_SYS	Enable set for system OCP interrupt for command or address error. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0

Table 15-167. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET

EMIF Controller

- Interrupt Requests: [0] [1] [2] [3] [4] [5]
- EMIF Register Summary: [6]

Table 15-168. EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR

Address Offset	0x0000 00BC		
Physical Address	0x4C00 00BC 0x4D00 00BC	**Instance**	EMIF1 EMIF2
Description	System OCP Interrupt Enable Clear Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED	EN_ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Enabled status of system ECC one bit error correction interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0x0
4	TWOBIT_ECC_ERR_SYS	Enabled status of system ECC two bit error detection interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0x0
3	WR_ECC_ERR_SYS	Enabled status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location . Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0x0
2:1	RESERVED		R	0x0
0	EN_ERR_SYS	Enable clear for system OCP interrupt for command or address error. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0

Table 15-169. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR

EMIF Controller

- Interrupt Requests: [0] [1] [2] [3] [4]
- EMIF Register Summary: [5]

Table 15-170. EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG

Address Offset	0x0000 00C8		
Physical Address	0x4C00 00C8 0x4D00 00C8	**Instance**	EMIF1 EMIF2
Description	SDRAM Output Impedance Calibration Config Register		
Type	RW		

31	30	29	28	27–20	19–18	17–16	15–0
RESERVED	ZQ_CS0EN	RESERVED	ZQ_SFEXITEN	RESERVED	ZQ_ZQINIT_MULT	ZQ_ZQCL_MULT	ZQ_REFINTERVAL

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ZQ_CS0EN	Writing a 1 enables ZQ calibration for CS0.	RW	0x0
29	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
28	ZQ_SFEXITEN	Writing a 1 enables the issuing of ZQCL on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.	RW	0x0
27:20	RESERVED		R	0x00
19:18	ZQ_ZQINIT_MULT	Indicates the number of ZQCL durations that make up a ZQINIT duration, minus one.	RW	0x0
17:16	ZQ_ZQCL_MULT	Indicates the number of ZQCS intervals that make up a ZQCL duration, minus one. ZQCS interval is defined by ZQ_ZQCS in EMIF_SDRAM_TIMING_3.	RW	0x0
15:0	ZQ_REFINTERVAL	Number of refresh periods between ZQCS commands. This field supports between one refresh period to 256 ms between ZQCS calibration commands. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register.	RW	0x0000

Table 15-171. Register Call Summary for Register EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG

EMIF Controller

- DDR3 Output Impedance Calibration: [0] [1] [2] [3]
- Global Initialization: [4] [5] [6] [7]
- Operational Modes Configuration: [8] [9] [10] [11]
- EMIF Register Summary: [12]

Table 15-172. EMIF_TEMP_ALERT_CONFIG

Address Offset	0x0000 00CC		
Physical Address	0x4C00 00CC 0x4D00 00CC	**Instance**	EMIF1 EMIF2
Description	Temperature Alert Configuration Register. **NOTE: This register is only applicable to LPDDR2 memories and cannot be used in this device.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TA_CS1EN	TA_CS0EN	RESERVED	TA_SFEXITEN	TA_DEVWDT		TA_DEVCNT		RESERVED		TA_REFINTERVAL																					

Bits	Field Name	Description	Type	Reset
31	TA_CS1EN	Writing 1 enables temperature alert polling for CS1.	RW	0x0
30	TA_CS0EN	Writing 1 enables temperature alert polling for CS0.	RW	0x0
29	RESERVED	Reserved	R	0x0
28	TA_SFEXITEN	Temperature Alert Poll on Self-Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing 1 enables the issuing of a temperature alert poll on Self-Refresh exit.	RW	0x0
27:26	TA_DEVWDT	This field indicates how wide a physical device is. It is used in conjunction with the TA_DEVCNT field to determine which byte lanes contain the temperature alert info. A value of 0: 8-bit wide, 1: 16-bit wide, 2: 32-bit wide. All others are reserved. If this field is set to 1 and the TA_DEVCNT field is set to 1 the byte mask for checking is 4'b0101.	RW	0x0

Bits	Field Name	Description	Type	Reset
25:24	TA_DEVCNT	This field indicates which external byte lanes contain a device for temperature monitoring. A value of 0: one device, 1: two devices, 2: four devices. All other reserved.	RW	0x0
23:22	RESERVED	Reserved	R	0x0
21:0	TA_REFINTERVAL	Number of refresh periods between temperature alert polls. This field supports between one refresh period to 10 seconds between temperature alert polls. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register.	RW	0x0

Table 15-173. Register Call Summary for Register EMIF_TEMP_ALERT_CONFIG

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-174. EMIF_OCP_ERROR_LOG

Address Offset	0x0000 00D0		
Physical Address	0x4C00 00D0 0x4D00 00D0	**Instance**	EMIF1 EMIF2
Description	OCP Error Log Register. This register is overwritten by any first error transaction once after the interrupt is serviced and cleared by writing 0x1 to the EMIF_SYSTEM_OCP_INTERRUPT_STATUS[0] ERR_SYS bit .		
Type	R		

31:16	15:14	13:11	10:8	7:0
RESERVED	MADDRSPACE	MBURSTSEQ	MCMD	MCONNID

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved for future use.	R	0x0000
15:14	MADDRSPACE	Address space of the first errored transaction. 0x0: SDRAM 0x1: reserved 0x2: reserved 0x3: internal registers	R	0x0
13:11	MBURSTSEQ	Addressing mode of the first errored transaction. (see Section 14.2.1, *L3_MAIN Interconnect* for more information)	R	0x0
10:8	MCMD	Command type of the first errored transaction. (see Section 14.2.1, *L3_MAIN Interconnect* for more information)	R	0x0
7:0	MCONNID	Connection ID of the first errored transaction.	R	0x00

Table 15-175. Register Call Summary for Register EMIF_OCP_ERROR_LOG

EMIF Controller

- Error Correction And Detection Feature: [0]
- EMIF Register Summary: [1]

Table 15-176. EMIF_READ_WRITE_LEVELING_RAMP_WINDOW

Address Offset	0x0000 00D4		
Physical Address	0x4C00 00D4 0x4D00 00D4	**Instance**	EMIF1 EMIF2

Table 15-176. EMIF_READ_WRITE_LEVELING_RAMP_WINDOW (continued)

Description	Read/write leveling ramp window register
Type	RW

31–13	12–0
RESERVED	RDWRLVLINC_RMP_WIN

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0 0000
12:0	RDWRLVLINC_RMP_WIN	Incremental leveling ramp window in number of refresh periods. The value programmed is minus one the required value. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register. **NOTE: Incremental leveling is not supported on this device.**	RW	0x0000

Table 15-177. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_RAMP_WINDOW

EMIF Controller

- Global Initialization: [3]
- EMIF Register Summary: [4]

Table 15-178. EMIF_READ_WRITE_LEVELING_RAMP_CONTROL

Address Offset	0x0000 00D8		
Physical Address	0x4C00 00D8 0x4D00 00D8	**Instance**	EMIF1 EMIF2
Description	Read/write leveling ramp control register		
Type	RW		

31	30–24	23–16	15–8	7–0
RDWRLVL_EN	RDWRLVLINC_RMP_PRE	RDLVLINC_RMP_INT	RDLVLGATEINC_RMP_INT	WRLVLINC_RMP_INT

Bits	Field Name	Description	Type	Reset
31	RDWRLVL_EN	Read-Write Leveling enable. Set 1 to enable leveling. Set 0 to disable leveling.	RW	0
30:24	RDWRLVLINC_RMP_PRE	Incremental leveling pre-scalar in number of refresh periods during ramp window. The value programmed is minus one the required value. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00
23:16	RDLVLINC_RMP_INT	Incremental read data eye training interval during ramp window. Number of RDWRLVLINC_RMP_PRE intervals between incremental read data eye training during ramp window. A value of 0 will disable incremental read data eye training.**NOTE: Incremental leveling is not supported on this device.**	RW	0x00
15:8	RDLVLGATEINC_RMP_INT	Incremental read DQS gate training interval during ramp window. Number of RDWRLVLINC_RMP_PRE intervals between incremental read DQS gate training during ramp window. A value of 0 will disable incremental read DQS gate training.**NOTE: Incremental leveling is not supported on this device.**	RW	0x00

Bits	Field Name	Description	Type	Reset
7:0	WRLVLINC_RMP_INT	Incremental write leveling interval during ramp window. Number of RDWRLVLINC_RMP_PRE intervals between incremental write leveling during ramp window. A value of 0 will disable incremental write leveling.**NOTE: Incremental leveling is not supported on this device.**	RW	0x00

Table 15-179. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_RAMP_CONTROL

EMIF Controller

- Global Initialization: [3] [4] [5] [6] [7]
- EMIF Register Summary: [8]

Table 15-180. EMIF_READ_WRITE_LEVELING_CONTROL

Address Offset	0x0000 00DC		
Physical Address	0x4C00 00DC 0x4D00 00DC	**Instance**	EMIF1 EMIF2
Description	Read/write leveling control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDWRLVLFULL_START	RDWRLVLINC_PRE							RDLVLINC_INT								RDLVLGATEINC_INT								WRLVLINC_INT							

Bits	Field Name	Description	Type	Reset
31	RDWRLVLFULL_START	Full leveling trigger. Writing a 1 to this field triggers full read and write leveling. This bit will self clear to 0.	RW	0
30:24	RDWRLVLINC_PRE	Incremental leveling pre-scalar in number of refresh periods. The value programmed is minus one the required value. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00
23:16	RDLVLINC_INT	Incremental read data eye training interval. Number of RDWRLVLINC_PRE intervals between incremental read data eye training. A value of 0 will disable incremental read data eye training. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00
15:8	RDLVLGATEINC_INT	Incremental read DQS gate training interval. Number of RDWRLVLINC_PRE intervals between incremental read DQS gate training. A value of 0 will disable incremental read DQS gate training. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00
7:0	WRLVLINC_INT	Incremental write leveling interval. Number of RDWRLVLINC_PRE intervals between incremental write leveling. A value of 0 will disable incremental write leveling. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00

Table 15-181. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_CONTROL

EMIF Controller

- Full Leveling: [0] [1]
- Global Initialization: [4] [5] [6]
- EMIF Register Summary: [7]

Table 15-182. EMIF_DDR_PHY_CONTROL_1

Address Offset	0x0000 00E4		
Physical Address	0x4C00 00E4 0x4D00 00E4	**Instance**	EMIF1 EMIF2
Description	PHY control register 1		
Type	RW		

31-28	27	26	25	24-22	21	20	19	18	17-10	9	8-5	4-0
RESERVED	RDLVL_MASK	RDLVLGATE_MASK	WRLVL_MASK	RESERVED	PHY_HALF_DELAYS	PHY_CLK_STALL_LEVEL	PHY_DIS_CALIB_RST	PHY_INVERT_CLKOUT	PHY_DLL_LOCK_DIFF	PHY_FAST_DLL_LOCK	RESERVED	READ_LATENCY

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	RW	0x0
27	RDLVL_MASK	Writing a 1 to this field will mask read data eye training during full leveling command, plus drives reg_phy_use_rd_data_eye_level control low to allow user to use programmed ratio values.	RW	0
26	RDLVLGATE_MASK	Writing a 1 to this field will mask dqs gate training during full leveling command, plus drives reg_phy_use_rd_dqs_level control low to allow user to use programmed ratio values.	RW	0
25	WRLVL_MASK	Writing a 1 to this field will mask write leveling training during full leveling command, plus drives reg_phy_use_wr_level control low to allow user to use programmed ratio values.	RW	0
24:22	RESERVED	Reserved	RW	0x0
21	PHY_HALF_DELAYS	Adjust slave delay line delays to support 2× mode 1: 2× mode (MDLL clock is half the rate of PHY) 0: 1× mode (MDLL clock rate is same as PHY)	RW	0
20	PHY_CLK_STALL_LEVEL	Enable variable idle value for delay lines. Enable during normal operations to avoid differential aging in the delay lines.	RW	0
19	PHY_DIS_CALIB_RST	Disable the dll_calib (internally generated) signal from resetting the Read Capture FIFO pointers and portions of data PHYs. Debug only. Note: dll_calib is generated by 1. EMIF_MISC_REG[0] DLL_CALIB_OS set to 1,or 2. by the PHY when it detects that the clock frequency variation has exceeded the bounds set by PHY_DLL_LOCK_DIFF or 3. periodically throughout the leveling process.	RW	0
18	PHY_INVERT_CLKOUT	Inverts the polarity of DRAM clock. 0: core clock is passed on to DRAM 1: inverted core clock is passed on to DRAM	RW	0

Bits	Field Name	Description	Type	Reset
17:10	PHY_DLL_LOCK_DIFF	The maximum number of delay line taps variation while maintaining the master DLL lock. When the PHY is in locked state and the variation on the clock exceeds the variation indicated by this field, the lock signal is de-asserted and a dll_calib signal is generated. To prevent the dll_calib signal from being asserted in the middle of traffic when the clock jitter exceeds the variation, this register needs to be set to a value which will ensure that the lock will not be lost. Recommended value is 16.	RW	0x02
9	PHY_FAST_DLL_LOCK	Controls master DLL to lock fast or average logic must be part of locking process. Set to 1 before OPP transition commences, and set back to 0 after OPP transition completes. 1: MDLL lock is asserted based on single sample 0: MDLL lock is asserted based on average of 16 samples.	RW	0
8:5	RESERVED	Reserved	RW	0x00
4:0	READ_LATENCY	This field defines the read latency for the read data from SDRAM in number of DDR clock cycles. This field is used by the EMIF as well as the PHY. READ_LATENCY = RL + reg_phy_rdc_we_to_re -1. EMIF uses above equation to calculate reg_phy_rdc_we_to_re and forward it to the PHY. For DDR3, the true RL is used, not the decoded value. See JEDEC spec.	RW	0x01E

Table 15-183. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_1

EMIF Controller

- Software Leveling: [0] [1]
- Global Initialization: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18]
- EMIF Register Summary: [19]

Table 15-184. EMIF_DDR_PHY_CONTROL_1_SHADOW

Address Offset	0x0000 00E8		
Physical Address	0x4C00 00E8 0x4D00 00E8	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RDLVL_MASK_SHDW	RDLVLGATE_MASK_SHDW	WRLVL_MASK_SHDW	RESERVED			PHY_HALF_DELAYS_SHDW	PHY_CLK_STALL_LEVEL_SHDW	PHY_DIS_CALIB_RST_SHDW	PHY_INVERT_CLKOUT_SHDW	PHY_DLL_LOCK_DIFF_SHDW								PHY_FAST_DLL_SHDW	RESERVED				READ_LATENCY_SHDW				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	RW	0x0
27	RDLVL_MASK_SHDW	Shadow field for RDLVL_MASK	RW	0
26	RDLVLGATE_MASK_SHDW	Shadow field for RDLVLGATE_MASK	RW	0
25	WRLVL_MASK_SHDW	Shadow field for WRLVL_MASK	RW	0

Bits	Field Name	Description	Type	Reset
24:22	RESERVED	Reserved	RW	0x0
21	PHY_HALF_DELAYS_SHDW	Shadow field for PHY_HALF_DELAYS	RW	0
20	PHY_CLK_STALL_LEVEL_SHDW	Shadow field for PHY_CLK_STALL_LEVEL	RW	0
19	PHY_DIS_CALIB_RST_SHDW	Shadow field for PHY_DIS_CALIB_RST	RW	0
18	PHY_INVERT_CLKOUT_SHDW	Shadow field for PHY_INVERT_CLKOUT	RW	0
17:10	PHY_DLL_LOCK_DIFF_SHDW	Shadow field for PHY_DLL_LOCK_DIFF	RW	0x00
9	PHY_FAST_DLL_SHDW	Shadow field for PHY_FAST_DLL	RW	0
8:5	RESERVED	Reserved	RW	0x00
4:0	READ_LATENCY_SHDW	Shadow field for READ_LATENCY	RW	0x000

Table 15-185. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_1_SHADOW

EMIF Controller

- Global Initialization: [0] [1]
- EMIF Register Summary: [2]

Table 15-186. EMIF_DDR_PHY_CONTROL_2

Address Offset	0x0000 00EC		
Physical Address	0x4C00 00EC 0x4D00 00EC	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0

Table 15-187. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_2

EMIF Controller

- Global Initialization: [0] [1] [2] [3]
- EMIF Register Summary: [4]

Table 15-188. EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING

Address offset	0x0000 0100		
Physical Address	0x4C00 0100 0x4D00 0100	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_COS_MAP_EN	RESERVED															PRI_7_COS		PRI_6_COS		PRI_5_COS		PRI_4_COS		PRI_3_COS		PRI_2_COS		PRI_1_COS		PRI_0_COS	

Bits	Field Name	Description	Type	Reset
31	PRI_COS_MAP_EN	Set 1 to enable priority to class of service mapping. Set 0 to disable mapping.	RW	0x0
30:16	RESERVED		R	0x0
15:14	PRI_7_COS	Class of service for commands with priority of 7. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
13:12	PRI_6_COS	Class of service for commands with priority of 6. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
11:10	PRI_5_COS	Class of service for commands with priority of 5. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
9:8	PRI_4_COS	Class of service for commands with priority of 4. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
7:6	PRI_3_COS	Class of service for commands with priority of 3. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
5:4	PRI_2_COS	Class of service for commands with priority of 2. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
3:2	PRI_1_COS	Class of service for commands with priority of 1. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
1:0	PRI_0_COS	Class of service for commands with priority of 0. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0

Table 15-189. Register Call Summary for Register EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING

EMIF Controller

- Class of Service: [0]
- Global Initialization: [1]
- EMIF Register Summary: [2]

Table 15-190. EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING

Address offset 0x0000 0104

Physical Address 0x4C00 0104 **Instance** EMIF1
0x4D00 0104 EMIF2

Description

Type RW

31	30:23	22:20	19:12	11:10	9:2	1:0
CONNID_COS_1_MAP_EN	CONNID_1_COS_1	MSK_1_COS_1	CONNID_2_COS_1	MSK_2_COS_1	CONNID_3_COS_1	MSK_3_COS_1

Bits	Field Name	Description	Type	Reset
31	CONNID_COS_1_MAP_EN	Set 1 to enable Connection ID to class of service 1 mapping. Set 0 to disable mapping.	RW	0x0

Bits	Field Name	Description	Type	Reset
30:23	CONNID_1_COS_1	Connection ID value 1 for class of service 1.	RW	0x0
22:20	MSK_1_COS_1	Mask for Connection ID value 1 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0. Value of 4 will mask Connection ID bits 3:0. Value of 5 will mask Connection ID bits 4:0. Value of 6 will mask Connection ID bits 5:0. Value of 7 will mask Connection ID bits 6:0.	RW	0x0
19:12	CONNID_2_COS_1	Connection ID value 2 for class of service 1.	RW	0x0
11:10	MSK_2_COS_1	Mask for Connection ID value 2 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0
9:2	CONNID_3_COS_1	Connection ID value 3 for class of service 1.	RW	0x0
1:0	MSK_3_COS_1	Mask for Connection ID value 3 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0

Table 15-191. Register Call Summary for Register EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING

EMIF Controller

- Class of Service: [0]
- Global Initialization: [1]
- EMIF Register Summary: [2]

Table 15-192. EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING

Address offset	0x0000 0108		
Physical Address	0x4C00 0108 0x4D00 0108	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONNID_COS_2_MAP_EN	CONNID_1_COS_2							MSK_1_COS_2			CONNID_2_COS_2								MSK_2_COS_2		CONNID_3_COS_2								MSK_3_COS_2	

Bits	Field Name	Description	Type	Reset
31	CONNID_COS_2_MAP_EN	Set 1 to enable Connection ID to class of service 2 mapping. Set 0 to disable mapping.	RW	0x0
30:23	CONNID_1_COS_2	Connection ID value 1 for class of service 2.	RW	0x0

Bits	Field Name	Description	Type	Reset
22:20	MSK_1_COS_2	Mask for Connection ID value 1 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0. Value of 4 will mask Connection ID bits 3:0. Value of 5 will mask Connection ID bits 4:0. Value of 6 will mask Connection ID bits 5:0. Value of 7 will mask Connection ID bits 6:0.	RW	0x0
19:12	CONNID_2_COS_2	Connection ID value 2 for class of service 2.	RW	0x0
11:10	MSK_2_COS_2	Mask for Connection ID value 2 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0
9:2	CONNID_3_COS_2	Connection ID value 3 for class of service 2.	RW	0x0
1:0	MSK_3_COS_2	Mask for Connection ID value 3 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0

Table 15-193. Register Call Summary for Register EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING

EMIF Controller

- Class of Service: [0]
- Global Initialization: [1]
- EMIF Register Summary: [2]

Table 15-194. EMIF_ECC_CTRL_REG

Address offset	0x4C00 0110		
Physical Address	0x4C00 0110	**Instance**	EMIF1
Description			
Type	RW		

31	30	29:2	1	0
REG_ECC_EN	REG_ECC_ADDR_RGN_PROT	RESERVED	REG_ECC_ADDR_RGN_2_EN	REG_ECC_ADDR_RGN_1_EN

Bits	Field Name	Description	Type	Reset
31	REG_ECC_EN	Set 1 to enable ECC. Set 0 to disable ECC.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	REG_ECC_ADDR_RGN_PROT	Setting this field to 1 and reg_ecc_en to a 1 will enable ECC calculation for accesses within the address ranges and disable ECC calculation for accesses outside the address ranges. Setting this field to 0 and reg_ecc_en to a 1 will disable ECC calculation for accesses within the address ranges and enable ECC calculation for accesses outside the address ranges. The address ranges can be specified using the ECC Address Range 1 and 2 registers.	RW	0x0
29:2	RESERVED		R	0x0
1	REG_ECC_ADDR_RGN_2_EN	Set 1 to enable ECC address range 2. Set 0 to disable ECC address range 2.	RW	0x0
0	REG_ECC_ADDR_RGN_1_EN	Set 1 to enable ECC address range 1. Set 0 to disable ECC address range 1.	RW	0x0

Table 15-195. Register Call Summary for Register EMIF_ECC_CTRL_REG

EMIF Controller

- Error Correction And Detection Feature: [0]
- Global Initialization: [5] [6]
- Operational Modes Configuration: [7] [8] [9] [10]
- EMIF Register Summary: [11]

Table 15-196. EMIF_ECC_ADDRESS_RANGE_1

Address offset	0x4C00 0114		
Physical Address	0x4C00 0114	**Instance**	EMIF1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_ECC_END_ADDR_1																REG_ECC_STRT_ADDR_1															

Bits	Field Name	Description	Type	Reset
31:16	REG_ECC_END_ADDR_1	End address[31:16] for ECC address range 1. If this bit field is set to 0x1000, this indicates that the SDRAM physical end address on which the ECC applies is 0x1000 FFFF. If this bit field is set to 0x0FFF the physical end address on which the ECC applies is 0x0FFF FFFF. This bit field controls only the 16 MSbs of the physical end address of the ECC protected range. The other 16 LSbs are always 0xFFFF.	RW	0x0
15:0	REG_ECC_STRT_ADDR_1	Start address[31:16] for ECC address range 1. If this bit field is set to 0x0000, this indicates that the SDRAM physical start address on which the ECC applies is 0x0000 0000. This bit field controls only the 16 MSbs of the physical start address of the ECC protected range. The other 16 LSbs are always 0x0000.	RW	0x0

Table 15-197. Register Call Summary for Register EMIF_ECC_ADDRESS_RANGE_1

EMIF Controller

- Error Correction And Detection Feature: [0]
- Global Initialization: [1]
- Operational Modes Configuration: [2] [3]
- EMIF Register Summary: [4]

Table 15-198. EMIF_ECC_ADDRESS_RANGE_2

Address offset	0x4C00 0118		
Physical Address	0x4C00 0118	**Instance**	EMIF1
Description			
Type	RW		

31–16	15–0
REG_ECC_END_ADDR_2	REG_ECC_STRT_ADDR_2

Bits	Field Name	Description	Type	Reset
31:16	REG_ECC_END_ADDR_2	End address[31:16] for ECC address range 2. If this bit field is set to 0x1000, this indicates that the SDRAM physical end address on which the ECC applies is 0x1000 FFFF. If this bit field is set to 0x0FFF the physical end address on which the ECC applies is 0x0FFF FFFF. This bit field controls only the 16 MSbs of the physical end address of the ECC protected range. The other 16 LSbs are always 0xFFFF.	RW	0x0
15:0	REG_ECC_STRT_ADDR_2	Start address[31:16] for ECC address range 2. If this bit field is set to 0x0000, this indicates that the SDRAM physical start address on which the ECC applies is 0x0000 0000. This bit field controls only the 16 MSbs of the physical start address of the ECC protected range. The other 16 LSbs are always 0x0000.	RW	0x0

Table 15-199. Register Call Summary for Register EMIF_ECC_ADDRESS_RANGE_2

EMIF Controller

- Error Correction And Detection Feature: [0]
- Global Initialization: [1]
- Operational Modes Configuration: [2] [3]
- EMIF Register Summary: [4]

Table 15-200. EMIF_READ_WRITE_EXECUTION_THRESHOLD

Address Offset	0x0000 0120		
Physical Address	0x4C00 0120 0x4D00 0120	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31	30–13	12–8	7–5	4–0
MFLAG_OVERRIDE	RESERVED	WR_THRSH	RESERVED	RD_THRSH

Bits	Field Name	Description	Type	Reset
31	MFLAG_OVERRIDE	Mflag override. 0x0: Use MFLAG 0x1: Use Priority Class of Service	RW	0
30:13	RESERVED	Reserved	R	0x0000

Bits	Field Name	Description	Type	Reset
12:8	WR_THRSH	Write Threshold. Number of SDRAM write bursts after which the EMIF arbitration will switch to executing read commands. The value programmed is always minus one the required number	RW	0x03
7:5	RESERVED	Reserved	R	0x0
4:0	RD_THRSH	Read threshold. Number of SDRAM read bursts after which the EMIF arbitration will switch to executing write commands. The value that is programmed is always minus one the required number	R	0x05

Table 15-201. Register Call Summary for Register EMIF_READ_WRITE_EXECUTION_THRESHOLD

EMIF Controller

- Class of Service: [0]
- Global Initialization: [1] [2] [3]
- EMIF Register Summary: [4]

Table 15-202. EMIF_COS_CONFIG

Address Offset	0x0000 0124		
Physical Address	0x4C00 0124 0x4D00 0124	**Instance**	EMIF1 EMIF2
Description	Priority Raise Counter Register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COS_COUNT_1								COS_COUNT_2								PR_OLD_COUNT							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	COS_COUNT_1	Priority Raise Counter for class of service 1. Number of EMIF_FICLK cycles after which the EMIF momentarily raises the priority of the class of service 1 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.	RW	0xFF
15:8	COS_COUNT_2	Priority Raise Counter for class of service 2. Number of EMIF_FICLK cycles after which the EMIF momentarily raises the priority of the class of service 2 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.	RW	0xFF
7:0	PR_OLD_COUNT	Priority Raise Old Counter. Number of EMIF_FICLK cycles after which the EMIF momentarily raises the priority of the oldest command in the Command FIFO. A value of N will be equal to N x 16 clocks.	RW	0xFF

Table 15-203. Register Call Summary for Register EMIF_COS_CONFIG

EMIF Controller

- Arbitration of Commands in the Command FIFO: [0]
- Class of Service: [1] [2] [3] [4] [5] [6] [7] [8]
- Global Initialization: [9]
- EMIF Register Summary: [10]

Table 15-204. EMIF_1B_ECC_ERR_CNT

Address offset	0x130		
Physical Address	0x4C00 0130	**Instance**	EMIF1
Description			

Table 15-204. EMIF_1B_ECC_ERR_CNT (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_1B_ECC_ERR_CNT																															

Bits	Field Name	Description	Type	Reset
31:0	REG_1B_ECC_ERR_CNT	32 bit counter that displays number of 1-bit ECC errors. Writing a value will decrement the count by that value. For example, if the count is 0x1234_ABF3, writing 0x1234_ABF3 to this register will clear it.	RW	0x0

Table 15-205. Register Call Summary for Register EMIF_1B_ECC_ERR_CNT

EMIF Controller

- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 15-206. EMIF_1B_ECC_ERR_THRSH

Address offset 0x134

Physical Address 0x4C00 0134 **Instance** EMIF1

Description

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_1B_ECC_ERR_THRSH								RESERVED								REG_1B_ECC_ERR_WIN															

Bits	Field Name	Description	Type	Reset
31:24	REG_1B_ECC_ERR_THRSH	1-bit ECC error threshold. The EMIF will generate an interrupt when the 1-bit ECC error count is greater than or equal to this threshold. A value of 0 will disable the generation of the interrupt.	RW	0x0
23:16	RESERVED		R	0x0
15:0	REG_1B_ECC_ERR_WIN	1-bit ECC error window in number of refresh periods. The EMIF will generate an interrupt when the 1-bit ECC error count is equal to or greater than the threshold within this window. A value of 0 will disable the window. Refresh period is defined by EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE. The software can set this bitfield to 0x0 to reset the internal counter.	RW	0x0

Table 15-207. Register Call Summary for Register EMIF_1B_ECC_ERR_THRSH

EMIF Controller

- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 15-208. EMIF_1B_ECC_ERR_DIST_1

Address offset 0x138

Physical Address 0x4C00 0138 **Instance** EMIF1

Description

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

REG_1B_ECC_ERR_DIST_1

Bits	Field Name	Description	Type	Reset
31:0	REG_1B_ECC_ERR_DIST_1	1-bit ECC error distribution over data bus bit 31:0. A value of 1 on a bit indicates 1-bit error on the corresponding bit on the data bus. Writing a 1 to any bit will clear that bit. Writing a 0 has no effect.	RW	0x0

Table 15-209. Register Call Summary for Register EMIF_1B_ECC_ERR_DIST_1

EMIF Controller

- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 15-210. EMIF_1B_ECC_ERR_ADDR_LOG

Address offset	0x13C		
Physical Address	0x4C00 013C	**Instance**	EMIF1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_1B_ECC_ERR_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	REG_1B_ECC_ERR_ADDR	1-bit ECC error address. Most significant bits of the starting address(es) related to the SDRAM reads that had a 1-bit ECC error. This field displays up to four addresses logged in the 4 deep address logging FIFO. Writing a 0x1 will pop one element of the FIFO. Writing a 0x2 will pop all elements of the FIFO. Writing any other alue will have no effect.	RW	0x0

Table 15-211. Register Call Summary for Register EMIF_1B_ECC_ERR_ADDR_LOG

EMIF Controller

- Error Correction And Detection Feature: [0]
- EMIF Register Summary: [1]

Table 15-212. EMIF_2B_ECC_ERR_ADDR_LOG

Address offset	0x140		
Physical Address	0x4C00 0140	**Instance**	EMIF1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_2B_ECC_ERR_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	REG_2B_ECC_ERR_ADDR	2-bit ECC error address. Most significant bits of the starting address of the first SDRAM burst that had the 2-bit ECC error. Writing a 1 will clear this field. Writing any other value has no effect.	RW	0x0

Table 15-213. Register Call Summary for Register EMIF_2B_ECC_ERR_ADDR_LOG

EMIF Controller
- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 15-214. EMIF_PHY_STATUS_1

Address Offset	0x0000 0144		
Physical Address	0x4C00 0144 0x4D00 0144	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:30	29:12	11:9	8:4	3:2	1:0
RESERVED	PHY_REG_PHY_CTRL_DLL_SLAVE_VALUE	RESERVED	PHY_REG_STATUS_DLL_LOCK	RESERVED	PHY_REG_PHY_CTRL_DLL_LOCK

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:12	PHY_REG_PHY_CTRL_DLL_SLAVE_VALUE	DLL Slave Value	R	0x0
11:9	RESERVED		R	0x0
8:4	PHY_REG_STATUS_DLL_LOCK	Lock Status for Data DLLs	R	0x0
3:2	RESERVED		R	0x0
1:0	PHY_REG_PHY_CTRL_DLL_LOCK	Lock Status for Command DLLs	R	0x0

Table 15-215. Register Call Summary for Register EMIF_PHY_STATUS_1

EMIF Controller
- EMIF Register Summary: [0]

Table 15-216. EMIF_PHY_STATUS_2

Address Offset	0x0000 0148		
Physical Address	0x4C00 0148 0x4D00 0148	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:0
PHY_REG_STATUS_DLL_SLAVE_VALUE_LO

Bits	Field Name	Description	Type	Reset
31:0	PHY_REG_STATUS_DLL_SLAVE_VALUE_LO	Bits 31:0 of Phy_reg_status_dll_slave_value	R	0x0

Table 15-217. Register Call Summary for Register EMIF_PHY_STATUS_2

EMIF Controller
- EMIF Register Summary: [0]

Table 15-218. EMIF_PHY_STATUS_3

Address Offset	0x0000 014C		
Physical Address	0x4C00 014C 0x4D00 014C	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31	30:16	15:13	12:0
RESERVED	PHY_REG_RDFIFO_RDPTR	RESERVED	PHY_REG_STATUS_DLL_SLAVE_VALUE_HI

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:16	PHY_REG_RDFIFO_RDPTR	Read FIFO Read Pointer	R	0x0
15:13	RESERVED		R	0x0
12:0	PHY_REG_STATUS_DLL_SLAVE_VALUE_HI	Bits 44:32 of Phy_reg_status_dll_slave_value	R	0x0

Table 15-219. Register Call Summary for Register EMIF_PHY_STATUS_3

EMIF Controller
- EMIF Register Summary: [0]

Table 15-220. EMIF_PHY_STATUS_4

Address Offset	0x0000 0150		
Physical Address	0x4C00 0150 0x4D00 0150	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31	30:16	15	14:0
RESERVED	PHY_REG_GATELVL_FSM	RESERVED	PHY_REG_RDFIFO_WRPTR

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:16	PHY_REG_GATELVL_FSM	Gate Leveling FSM	R	0x0
15	RESERVED		R	0x0
14:0	PHY_REG_RDFIFO_WRPTR	Read FIFO Write Pointer	R	0x0

Table 15-221. Register Call Summary for Register EMIF_PHY_STATUS_4

EMIF Controller
- EMIF Register Summary: [0]

Table 15-222. EMIF_PHY_STATUS_5

Address Offset	0x0000 0154		
Physical Address	0x4C00 0154 0x4D00 0154	**Instance**	EMIF1 EMIF2
Description			

Table 15-222. EMIF_PHY_STATUS_5 (continued)

Type	R

31:20	19:0
RESERVED	PHY_REG_RD_LEVEL_FSM

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	PHY_REG_RD_LEVEL_FSM	Read Leveling FSM	R	0x0

Table 15-223. Register Call Summary for Register EMIF_PHY_STATUS_5

EMIF Controller

- EMIF Register Summary: [0]

Table 15-224. EMIF_PHY_STATUS_6

Address Offset	0x0000 0158		
Physical Address	0x4C00 0158 0x4D00 0158	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:15	14:0
RESERVED	PHY_REG_WR_LEVEL_FSM

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:0	PHY_REG_WR_LEVEL_FSM	Write Leveling FSM	R	0x0

Table 15-225. Register Call Summary for Register EMIF_PHY_STATUS_6

EMIF Controller

- EMIF Register Summary: [0]

Table 15-226. EMIF_PHY_STATUS_7

Address Offset	0x0000 015C		
Physical Address	0x4C00 015C 0x4D00 015C	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RDLVL_DQS_RATIO1	RESERVED	PHY_REG_RDLVL_DQS_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO1	Read leveling DQS ratio1	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO0	Read leveling DQS ratio0	R	0x0

Table 15-227. Register Call Summary for Register EMIF_PHY_STATUS_7

EMIF Controller

- Global Initialization: [0] [1]
- EMIF Register Summary: [2]

Table 15-228. EMIF_PHY_STATUS_8

Address Offset	0x0000 0160		
Physical Address	0x4C00 0160 0x4D00 0160	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RDLVL_DQS_RATIO3										RESERVED						PHY_REG_RDLVL_DQS_RATIO2									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO3	Read leveling DQS ratio3	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO2	Read leveling DQS ratio2	R	0x0

Table 15-229. Register Call Summary for Register EMIF_PHY_STATUS_8

EMIF Controller

- EMIF Register Summary: [0]

Table 15-230. EMIF_PHY_STATUS_9

Address Offset	0x0000 0164		
Physical Address	0x4C00 0164 0x4D00 0164	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RDLVL_DQS_RATIO5										RESERVED						PHY_REG_RDLVL_DQS_RATIO4									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO5	Read Leveling DQS ratio5	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO4	Read Leveling DQS ratio4	R	0x0

Table 15-231. Register Call Summary for Register EMIF_PHY_STATUS_9

EMIF Controller

- EMIF Register Summary: [0]

Table 15-232. EMIF_PHY_STATUS_10

Address Offset	0x0000 0168		
Physical Address	0x4C00 0168 0x4D00 0168	**Instance**	EMIF1 EMIF2

Table 15-232. EMIF_PHY_STATUS_10 (continued)

Description	
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RDLVL_DQS_RATIO7										RESERVED						PHY_REG_RDLVL_DQS_RATIO6									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO7	Read leveling DQS ratio7	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO6	Read leveling DQS ratio6	R	0x0

Table 15-233. Register Call Summary for Register EMIF_PHY_STATUS_10

EMIF Controller

- EMIF Register Summary: [0]

Table 15-234. EMIF_PHY_STATUS_11

Address Offset	0x0000 016C		
Physical Address	0x4C00 016C	**Instance**	EMIF1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RDLVL_DQS_RATIO9										RESERVED						PHY_REG_RDLVL_DQS_RATIO8									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO9	Read leveling DQS ratio9	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO8	Read leveling DQS ratio8	R	0x0

Table 15-235. Register Call Summary for Register EMIF_PHY_STATUS_11

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-236. EMIF_PHY_STATUS_12

Address Offset	0x0000 0170		
Physical Address	0x4C00 0170 0x4D00 0170	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO1											RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO0										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO1	Read leveling FIFO Write Enable Ratio1	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO0	Read leveling FIFO Write Enable Ratio0	R	0x0

Table 15-237. Register Call Summary for Register EMIF_PHY_STATUS_12

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-238. EMIF_PHY_STATUS_13

Address Offset	0x0000 0174		
Physical Address	0x4C00 0174 0x4D00 0174	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_RDLVL_FIFOWEIN_RATIO3	RESERVED	PHY_REG_RDLVL_FIFOWEIN_RATIO2

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO3	Read leveling FIFO Write Enable Ratio3	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO2	Read leveling FIFO Write Enable Ratio2	R	0x0

Table 15-239. Register Call Summary for Register EMIF_PHY_STATUS_13

EMIF Controller

- EMIF Register Summary: [0]

Table 15-240. EMIF_PHY_STATUS_14

Address Offset	0x0000 0178		
Physical Address	0x4C00 0178 0x4D00 0178	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_RDLVL_FIFOWEIN_RATIO5	RESERVED	PHY_REG_RDLVL_FIFOWEIN_RATIO4

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO5	Read leveling FIFO Write Enable Ratio5	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO4	Read leveling FIFO Write Enable Ratio4	R	0x0

Table 15-241. Register Call Summary for Register EMIF_PHY_STATUS_14

EMIF Controller

- EMIF Register Summary: [0]

Table 15-242. EMIF_PHY_STATUS_15

Address Offset	0x0000 017C		
Physical Address	0x4C00 017C 0x4D00 017C	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO7											RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO6										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO7	Read leveling FIFO Wrie Enable Ratio7	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO6	Read leveling FIFO Wrie Enable Ratio6	R	0x0

Table 15-243. Register Call Summary for Register EMIF_PHY_STATUS_15

EMIF Controller

- EMIF Register Summary: [0]

Table 15-244. EMIF_PHY_STATUS_16

Address Offset	0x0000 0180		
Physical Address	0x4C00 0180	**Instance**	EMIF1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO9											RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO8										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO9	Read leveling FIFO Write Enable Ratio9	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO8	Read leveling FIFO Write Enable Ratio8	R	0x0

Table 15-245. Register Call Summary for Register EMIF_PHY_STATUS_16

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-246. EMIF_PHY_STATUS_17

Address Offset	0x0000 0184		
Physical Address	0x4C00 0184 0x4D00 0184	**Instance**	EMIF1 EMIF2

Table 15-246. EMIF_PHY_STATUS_17 (continued)

Description	
Type	R

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WRLVL_DQ_RATIO1	RESERVED	PHY_REG_WRLVL_DQ_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO1	Write leveling DQ ratio1	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO0	Write leveling DQ ratio0	R	0x0

Table 15-247. Register Call Summary for Register EMIF_PHY_STATUS_17

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-248. EMIF_PHY_STATUS_18

Address Offset	0x0000 0188		
Physical Address	0x4C00 0188 0x4D00 0188	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WRLVL_DQ_RATIO3	RESERVED	PHY_REG_WRLVL_DQ_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO3	Write leveling DQ ratio3	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO2	Write leveling DQ ratio2	R	0x0

Table 15-249. Register Call Summary for Register EMIF_PHY_STATUS_18

EMIF Controller

- EMIF Register Summary: [0]

Table 15-250. EMIF_PHY_STATUS_19

Address Offset	0x0000 018C		
Physical Address	0x4C00 018C 0x4D00 018C	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WRLVL_DQ_RATIO5	RESERVED	PHY_REG_WRLVL_DQ_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO5	Write leveling DQ ratio5	R	0x0

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO4	Write leveling DQ ratio4	R	0x0

Table 15-251. Register Call Summary for Register EMIF_PHY_STATUS_19

EMIF Controller

- EMIF Register Summary: [0]

Table 15-252. EMIF_PHY_STATUS_20

Address Offset	0x0000 0190		
Physical Address	0x4C00 0190 0x4D00 0190	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WRLVL_DQ_RATIO7	RESERVED	PHY_REG_WRLVL_DQ_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO7	Write leveling DQ ratio7	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO6	Write leveling DQ ratio6	R	0x0

Table 15-253. Register Call Summary for Register EMIF_PHY_STATUS_20

EMIF Controller

- EMIF Register Summary: [0]

Table 15-254. EMIF_PHY_STATUS_21

Address Offset	0x0000 0194		
Physical Address	0x4C00 0194	**Instance**	EMIF1
Description			
Type	R		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WRLVL_DQ_RATIO9	RESERVED	PHY_REG_WRLVL_DQ_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO9	Write leveling DQ ratio9	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO8	Write leveling DQ ratio8	R	0x0

Table 15-255. Register Call Summary for Register EMIF_PHY_STATUS_21

EMIF Controller

- EMIF Register Summary: [0]

Table 15-256. EMIF_PHY_STATUS_22

Address Offset	0x0000 0198		
Physical Address	0x4C00 0198 0x4D00 0198	**Instance**	EMIF1 EMIF2

Table 15-256. EMIF_PHY_STATUS_22 (continued)

Description	
Type	R

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WRLVL_DQS_RATIO1	RESERVED	PHY_REG_WRLVL_DQS_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO 1	Write leveling DQS ratio 1	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO 0	Write leveling DQS ratio 0	R	0x0

Table 15-257. Register Call Summary for Register EMIF_PHY_STATUS_22

EMIF Controller

- EMIF Register Summary: [0]

Table 15-258. EMIF_PHY_STATUS_23

Address Offset	0x0000 019C		
Physical Address	0x4C00 019C 0x4D00 019C	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WRLVL_DQS_RATIO3	RESERVED	PHY_REG_WRLVL_DQS_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO 3	Write leveling DQS ratio3	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO 2	Write leveling DQS ratio2	R	0x0

Table 15-259. Register Call Summary for Register EMIF_PHY_STATUS_23

EMIF Controller

- EMIF Register Summary: [0]

Table 15-260. EMIF_PHY_STATUS_24

Address Offset	0x0000 01A0		
Physical Address	0x4C00 01A0 0x4D00 01A0	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WRLVL_DQS_RATIO5	RESERVED	PHY_REG_WRLVL_DQS_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25:16	PHY_REG_WRLVL_DQS_RATIO5	Write leveling DQS ratio5	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO4	Write leveling DQS ratio4	R	0x0

Table 15-261. Register Call Summary for Register EMIF_PHY_STATUS_24

EMIF Controller

- EMIF Register Summary: [0]

Table 15-262. EMIF_PHY_STATUS_25

Address Offset	0x0000 01A4		
Physical Address	0x4C00 01A4 0x4D00 01A4	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WRLVL_DQS_RATIO7	RESERVED	PHY_REG_WRLVL_DQS_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO7	Write leveling DQS ratio7	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO6	Write leveling DQS ratio6	R	0x0

Table 15-263. Register Call Summary for Register EMIF_PHY_STATUS_25

EMIF Controller

- EMIF Register Summary: [0]

Table 15-264. EMIF_PHY_STATUS_26

Address Offset	0x0000 01A8		
Physical Address	0x4C00 01A8	**Instance**	EMIF1
Description			
Type	R		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WRLVL_DQS_RATIO9	RESERVED	PHY_REG_WRLVL_DQS_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO9	Write leveling DQS ratio9	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO8	Write leveling DQS ratio8	R	0x0

Table 15-265. Register Call Summary for Register EMIF_PHY_STATUS_26

EMIF Controller

- Global Initialization: [0] [1]
- EMIF Register Summary: [2]

Table 15-266. EMIF_PHY_STATUS_27

Address Offset	0x0000 01AC		
Physical Address	0x4C00 01AC 0x4D00 01AC	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:30	29:28	27:25	24:20	19:0
RESERVED	PHY_REG_PHY_CONTROL_MDLL_UNLOCK_STICKY	RESERVED	PHY_REG_STATUS_MDLL_UNLOCK_STICKY	PHY_REG_RDC_FIFO_RST_ERR_CNT

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PHY_REG_PHY_CONTROL_MDLL_UNLOCK_STICKY	Phy control MDLL unlock sticky	R	0x0
27:25	RESERVED		R	0x0
24:20	PHY_REG_STATUS_MDLL_UNLOCK_STICKY	Phy data MDLL unlock sticky	R	0x0
19:0	PHY_REG_RDC_FIFO_RST_ERR_CNT	RDC FIFO reset error count	R	0x0

Table 15-267. Register Call Summary for Register EMIF_PHY_STATUS_27

EMIF Controller

- EMIF Register Summary: [0]

Table 15-268. EMIF_PHY_STATUS_28

Address Offset	0x0000 01B0		
Physical Address	0x4C00 01B0 0x4D00 01B0	**Instance**	EMIF1 EMIF2
Description			
Type	R		

31:29	28:24	23:21	20:16	15:13	12:8	7:5	4:0
RESERVED	PHY_REG_GATELVL_INC_FAIL	RESERVED	PHY_REG_WRLVL_INC_FAIL	RESERVED	PHY_REG_RDLVL_INC_FAIL	RESERVED	PHY_REG_FIFO_WE_IN_MIASALIGNED_STICKY

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	PHY_REG_GATELVL_INC_FAIL	Gate leveling failure. **NOTE: Incremental leveling is not supported on this device.**	R	0x0
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20:16	PHY_REG_WRLVL_INC_FAIL	Write leveling failure. **NOTE: Incremental leveling is not supported on this device.**	R	0x0
15:13	RESERVED		R	0x0
12:8	PHY_REG_RDLVL_INC_FAIL	Read leveling failure. **NOTE: Incremental leveling is not supported on this device.**	R	0x0
7:5	RESERVED		R	0x0
4:0	PHY_REG_FIFO_WE_IN_MIASALIGNED_STICKY	FIFO write enable in misaligned sticky	R	0x0

Table 15-269. Register Call Summary for Register EMIF_PHY_STATUS_28

EMIF Controller

- EMIF Register Summary: [0]

Table 15-270. EMIF_EXT_PHY_CONTROL_1

Address Offset	0x0000 0200		
Physical Address	0x4C00 0200 0x4D00 0200	**Instance**	EMIF1 EMIF2
Description	Control DLL Slave Ratio Register		
Type	RW		

31:30	29:20	19:10	9:0
RESERVED	PHY_REG_CTRL_SLAVE_RATIO2	PHY_REG_CTRL_SLAVE_RATIO1	PHY_REG_CTRL_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	PHY_REG_CTRL_SLAVE_RATIO2	The user programmable ratio value for address/ command launch timing in PHY control macro 2. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **Required for LPDDR2 only. Please check the device data manual for supported DDR memory types.**	RW	0x40
19:10	PHY_REG_CTRL_SLAVE_RATIO1	The user programmable ratio value for address/ command launch timing in PHY control macro 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_CTRL_SLAVE_RATIO0	The user programmable ratio value for address/ command launch timing in PHY control macro 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80

Table 15-271. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_1

EMIF Controller

- Global Initialization: [0] [1] [2] [3] [4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6]

Table 15-272. EMIF_EXT_PHY_CONTROL_1_SHADOW

Address Offset	0x0000 0204		
Physical Address	0x4C00 0204 0x4D00 0204	**Instance**	EMIF1 EMIF2
Description	Control DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_1 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PHY_REG_CTRL_SLAVE_RATIO2										PHY_REG_CTRL_SLAVE_RATIO1										PHY_REG_CTRL_SLAVE_RATIO0									

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	PHY_REG_CTRL_SLAVE_RATIO2	The user programmable ratio value for address/ command launch timing in PHY control macro 2. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **Required for LPDDR2 only. Please check the device data manual for supported DDR memory types.**	RW	0x40

Bits	Field Name	Description	Type	Reset
19:10	PHY_REG_CTRL_SLAVE_RATIO1	The user programmable ratio value for address/ command launch timing in PHY control macro 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80
9:0	PHY_REG_CTRL_SLAVE_RATIO0	The user programmable ratio value for address/ command launch timing in PHY control macro 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80

Table 15-273. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_1_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-274. EMIF_EXT_PHY_CONTROL_2

Address Offset	0x0000 0208		
Physical Address	0x4C00 0208 0x4D00 0208	**Instance**	EMIF1 EMIF2
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO1	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO1	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO0	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-275. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_2

EMIF Controller

- Full Leveling: [0]
- Software Leveling: [1] [2] [3]
- Global Initialization: [4] [5] [6] [7]
- EMIF Register Summary: [8]
- EMIF Register Description: [9]

Table 15-276. EMIF_EXT_PHY_CONTROL_2_SHADOW

Address Offset	0x0000 020C		
Physical Address	0x4C00 020C 0x4D00 020C	**Instance**	EMIF1 EMIF2
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_2 on any frequency change.		
Type	RW		

31–27	26–16	15–11	10–0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO1	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO1	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO0	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-277. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_2_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-278. EMIF_EXT_PHY_CONTROL_3

Address Offset	0x0000 0210		
Physical Address	0x4C00 0210 0x4D00 0210	**Instance**	EMIF1 EMIF2
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO3											RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO2										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO3	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO2	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-279. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_3

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-280. EMIF_EXT_PHY_CONTROL_3_SHADOW

Address Offset	0x0000 0214		
Physical Address	0x4C00 0214 0x4D00 0214	**Instance**	EMIF1 EMIF2
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_3 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO3											RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO2										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO3	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO2	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-281. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_3_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-282. EMIF_EXT_PHY_CONTROL_4

Address Offset	0x0000 0218		
Physical Address	0x4C00 0218 0x4D00 0218	**Instance**	EMIF1 EMIF2
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO5	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO5	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO4	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-283. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_4

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-284. EMIF_EXT_PHY_CONTROL_4_SHADOW

Address Offset	0x0000 021C		
Physical Address	0x4C00 021C 0x4D00 021C	**Instance**	EMIF1 EMIF2
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_4 on any frequency change.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO5	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO5	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO4	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-285. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_4_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-286. EMIF_EXT_PHY_CONTROL_5

Address Offset	0x0000 0220		
Physical Address	0x4C00 0220 0x4D00 0220	**Instance**	EMIF1 EMIF2
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO7	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO7	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO6	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-287. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_5

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-288. EMIF_EXT_PHY_CONTROL_5_SHADOW

Address Offset	0x0000 0224		
Physical Address	0x4C00 0224 0x4D00 0224	**Instance**	EMIF1 EMIF2
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Ratio Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_5 on any frequency change.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO7	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO7	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO6	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-289. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_5_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-290. EMIF_EXT_PHY_CONTROL_6

Address Offset	0x0000 0228		
Physical Address	0x4C00 0228	**Instance**	EMIF1
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO9	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO8

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO9	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO8	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-291. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_6

EMIF Controller

- Software Leveling: [0] [1] [2]
- Global Initialization: [3]
- EMIF Register Summary: [4]
- EMIF Register Description: [5]

Table 15-292. EMIF_EXT_PHY_CONTROL_6_SHADOW

Address Offset	0x0000 022C

Table 15-292. EMIF_EXT_PHY_CONTROL_6_SHADOW (continued)

Physical Address	0x4C00 022C	**Instance**	EMIF1
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_6 on any frequency change.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO9	RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO8

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO9	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO8	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-293. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_6_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-294. EMIF_EXT_PHY_CONTROL_7

Address Offset	0x0000 0230		
Physical Address	0x4C00 0230 0x4D00 0230	**Instance**	EMIF1 EMIF2
Description	Data macro 0, read DQS DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO1	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO1	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO0	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-295. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_7

EMIF Controller
- Global Initialization: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 15-296. EMIF_EXT_PHY_CONTROL_7_SHADOW

Address Offset	0x0000 0234		
Physical Address	0x4C00 0234 0x4D00 0234	**Instance**	EMIF1 EMIF2
Description	Data macro 0, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_7 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO1	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO1	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO0	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-297. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_7_SHADOW

EMIF Controller
- EMIF Register Summary: [0]

Table 15-298. EMIF_EXT_PHY_CONTROL_8

Address Offset	0x0000 0238		
Physical Address	0x4C00 0238 0x4D00 0238	**Instance**	EMIF1 EMIF2
Description	Data macro 1, read DQS DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO3	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25:16	PHY_REG_RD_DQS_SLAVE_RATIO3	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO2	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-299. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_8

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-300. EMIF_EXT_PHY_CONTROL_8_SHADOW

Address Offset	0x0000 023C		
Physical Address	0x4C00 023C 0x4D00 023C	**Instance**	EMIF1 EMIF2
Description	Data macro 1, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_8 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO3	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO3	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO2	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-301. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_8_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-302. EMIF_EXT_PHY_CONTROL_9

Address Offset	0x0000 0240		
Physical Address	0x4C00 0240 0x4D00 0240	**Instance**	EMIF1 EMIF2

Table 15-302. EMIF_EXT_PHY_CONTROL_9 (continued)

Description	Data macro 2, read DQS DLL Slave Ratio Register
Type	RW

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO5	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO5	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO4	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-303. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_9

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-304. EMIF_EXT_PHY_CONTROL_9_SHADOW

Address Offset	0x0000 0244		
Physical Address	0x4C00 0244 0x4D00 0244	**Instance**	EMIF1 EMIF2
Description	Data macro 2, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_9 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO5	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO5	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO4	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-305. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_9_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-306. EMIF_EXT_PHY_CONTROL_10

Address Offset	0x0000 0248		
Physical Address	0x4C00 0248 0x4D00 0248	**Instance**	EMIF1 EMIF2
Description	Data macro 3, read DQS DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO7	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO7	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO6	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-307. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_10

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-308. EMIF_EXT_PHY_CONTROL_10_SHADOW

Address Offset	0x0000 024C		
Physical Address	0x4C00 024C 0x4D00 024C	**Instance**	EMIF1 EMIF2
Description	Data macro 3, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_10 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO7	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO7	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO6	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-309. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_10_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-310. EMIF_EXT_PHY_CONTROL_11

Address Offset	0x0000 0250		
Physical Address	0x4C00 0250	**Instance**	EMIF1
Description	ECC Data macro, read DQS DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO9	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO9	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO8	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-311. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_11

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 15-312. EMIF_EXT_PHY_CONTROL_11_SHADOW

Address Offset	0x0000 0254		
Physical Address	0x4C00 0254	**Instance**	EMIF1
Description	ECC Data macro, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_11 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO9	RESERVED	PHY_REG_RD_DQS_SLAVE_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO9	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO8	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-313. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_11_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-314. EMIF_EXT_PHY_CONTROL_12

Address Offset	0x0000 0258		
Physical Address	0x4C00 0258 0x4D00 0258	**Instance**	EMIF1 EMIF2
Description	Data macro 0, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO1	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO1	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO0	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-315. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_12

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 15-316. EMIF_EXT_PHY_CONTROL_12_SHADOW

Address Offset	0x0000 025C

Table 15-316. EMIF_EXT_PHY_CONTROL_12_SHADOW (continued)

Physical Address	0x4C00 025C 0x4D00 025C	**Instance**	EMIF1 EMIF2
Description	Data macro 0, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_12 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO1	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO1	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO0	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-317. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_12_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-318. EMIF_EXT_PHY_CONTROL_13

Address Offset	0x0000 0260		
Physical Address	0x4C00 0260 0x4D00 0260	**Instance**	EMIF1 EMIF2
Description	Data macro 1, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO3	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO3	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WR_DATA_SLAVE_RATIO2	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-319. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_13

EMIF Controller
- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-320. EMIF_EXT_PHY_CONTROL_13_SHADOW

Address Offset	0x0000 0264		
Physical Address	0x4C00 0264 0x4D00 0264	**Instance**	EMIF1 EMIF2
Description	Data macro 1, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_13 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO3	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO3	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO2	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-321. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_13_SHADOW

EMIF Controller
- EMIF Register Summary: [0]

Table 15-322. EMIF_EXT_PHY_CONTROL_14

Address Offset	0x0000 0268		
Physical Address	0x4C00 0268 0x4D00 0268	**Instance**	EMIF1 EMIF2
Description	Data macro 2, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO5	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO5	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO4	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-323. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_14

EMIF Controller
- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-324. EMIF_EXT_PHY_CONTROL_14_SHADOW

Address Offset	0x0000 026C		
Physical Address	0x4C00 026C 0x4D00 026C	**Instance**	EMIF1 EMIF2
Description	Data macro 2, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_14 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO5	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO5	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO4	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-325. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_14_SHADOW

EMIF Controller
- EMIF Register Summary: [0]

Table 15-326. EMIF_EXT_PHY_CONTROL_15

Address Offset	0x0000 0270

Table 15-326. EMIF_EXT_PHY_CONTROL_15 (continued)

Physical Address	0x4C00 0270 0x4D00 0270	**Instance**	EMIF1 EMIF2
Description	Data macro 3, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO7	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO7	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO6	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-327. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_15

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-328. EMIF_EXT_PHY_CONTROL_15_SHADOW

Address Offset	0x0000 0274		
Physical Address	0x4C00 0274 0x4D00 0274	**Instance**	EMIF1 EMIF2
Description	Data macro 3, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_15 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO7	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO7	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WR_DATA_SLAVE_RATIO6	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-329. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_15_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-330. EMIF_EXT_PHY_CONTROL_16

Address Offset	0x0000 0278		
Physical Address	0x4C00 0278	**Instance**	EMIF1
Description	ECC Data macro, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO9	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO9	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO8	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-331. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_16

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-332. EMIF_EXT_PHY_CONTROL_16_SHADOW

Address Offset	0x0000 027C		
Physical Address	0x4C00 027C	**Instance**	EMIF1
Description	ECC Data macro, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_16 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO9	RESERVED	PHY_REG_WR_DATA_SLAVE_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25:16	PHY_REG_WR_DATA_SLAVE_RATIO9	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO8	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 15-333. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_16_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-334. EMIF_EXT_PHY_CONTROL_17

Address Offset	0x0000 0280		
Physical Address	0x4C00 0280 0x4D00 0280	**Instance**	EMIF1 EMIF2
Description	Data macro 0, write DQS DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO1	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO1	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO0	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-335. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_17

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-336. EMIF_EXT_PHY_CONTROL_17_SHADOW

Address Offset	0x0000 0284		
Physical Address	0x4C00 0284 0x4D00 0284	**Instance**	EMIF1 EMIF2

Table 15-336. EMIF_EXT_PHY_CONTROL_17_SHADOW (continued)

Description	Data macro 0, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_17 on any frequency change.
Type	RW

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO1	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO1	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO0	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-337. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_17_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-338. EMIF_EXT_PHY_CONTROL_18

Address Offset	0x0000 0288		
Physical Address	0x4C00 0288 0x4D00 0288	**Instance**	EMIF1 EMIF2
Description	Data macro 1, write DQS DLL Slave Ratio Register		
Type	RW		

31–26	25–16	15–10	9–0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO3	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO3	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO2	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-339. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_18

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-340. EMIF_EXT_PHY_CONTROL_18_SHADOW

Address Offset	0x0000 028C		
Physical Address	0x4C00 028C 0x4D00 028C	**Instance**	EMIF1 EMIF2
Description	Data macro 1, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_18 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO3	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO3	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO2	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-341. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_18_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-342. EMIF_EXT_PHY_CONTROL_19

Address Offset	0x0000 0290		
Physical Address	0x4C00 0290 0x4D00 0290	**Instance**	EMIF1 EMIF2
Description	Data macro 2, write DQS DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO5	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO5	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO4	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-343. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_19

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-344. EMIF_EXT_PHY_CONTROL_19_SHADOW

Address Offset	0x0000 0294		
Physical Address	0x4C00 0294 0x4D00 0294	**Instance**	EMIF1 EMIF2
Description	Data macro 2, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_19 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO5	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO5	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO4	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-345. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_19_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-346. EMIF_EXT_PHY_CONTROL_20

Address Offset	0x0000 0298		
Physical Address	0x4C00 0298 0x4D00 0298	**Instance**	EMIF1 EMIF2
Description	Data macro 3, write DQS DLL Slave Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO7	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO7	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO6	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-347. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_20

EMIF Controller
- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-348. EMIF_EXT_PHY_CONTROL_20_SHADOW

Address Offset	0x0000 029C		
Physical Address	0x4C00 029C 0x4D00 029C	**Instance**	EMIF1 EMIF2
Description	Data macro 3, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_20 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO7	RESERVED	PHY_REG_WR_DQS_SLAVE_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO7	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO6	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-349. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_20_SHADOW

EMIF Controller
- EMIF Register Summary: [0]

Table 15-350. EMIF_EXT_PHY_CONTROL_21

Address Offset	0x0000 02A0		
Physical Address	0x4C00 02A0	**Instance**	EMIF1

Table 15-350. EMIF_EXT_PHY_CONTROL_21 (continued)

Description	ECC Data macro, write DQS DLL Slave Ratio Register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO9										RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO8									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO9	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO8	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-351. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_21

EMIF Controller

- Full Leveling: [0]
- Global Initialization: [1] [2] [3] [4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6]

Table 15-352. EMIF_EXT_PHY_CONTROL_21_SHADOW

Address Offset	0x0000 02A4		
Physical Address	0x4C00 02A4	**Instance**	EMIF1
Description	ECC Data macro, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_21 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO9										RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO8									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO9	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WR_DQS_SLAVE_RATIO8	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-353. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_21_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-354. EMIF_EXT_PHY_CONTROL_22

Address Offset	0x0000 02A8		
Physical Address	0x4C00 02A8 0x4D00 02A8	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED (31:25)	PHY_REG_FIFO_WE_IN_DELAY (24:16)	RESERVED (15:9)	PHY_REG_CTRL_SLAVE_DELAY (8:0)

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_FIFO_WE_IN_DELAY	The user programmable FIFO write enable delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_CTRL_SLAVE_DELAY	The user programmable command delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 15-355. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_22

EMIF Controller

- Global Initialization: [0] [1] [2] [3] [4]
- EMIF Register Summary: [5]

Table 15-356. EMIF_EXT_PHY_CONTROL_22_SHADOW

Address Offset	0x0000 02AC		
Physical Address	0x4C00 02AC 0x4D00 02AC	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED (31:25)	PHY_REG_FIFO_WE_IN_DELAY (24:16)	RESERVED (15:9)	PHY_REG_CTRL_SLAVE_DELAY (8:0)

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_FIFO_WE_IN_DELAY	The user programmable FIFO write enable delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_CTRL_SLAVE_DELAY	The user programmable command delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 15-357. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_22_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-358. EMIF_EXT_PHY_CONTROL_23

Address Offset	0x0000 02B0		
Physical Address	0x4C00 02B0 0x4D00 02B0	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	PHY_REG_WR_DQS_SLAVE_DELAY	RESERVED	PHY_REG_RD_DQS_SLAVE_DELAY

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_WR_DQS_SLAVE_DELAY	The user programmable write DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_RD_DQS_SLAVE_DELAY	The user programmable read DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 15-359. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_23

EMIF Controller

- Global Initialization: [0] [1] [2] [3] [4]
- EMIF Register Summary: [5]

Table 15-360. EMIF_EXT_PHY_CONTROL_23_SHADOW

Address Offset	0x0000 02B4		
Physical Address	0x4C00 02B4 0x4D00 02B4	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	PHY_REG_WR_DQS_SLAVE_DELAY	RESERVED	PHY_REG_RD_DQS_SLAVE_DELAY

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_WR_DQS_SLAVE_DELAY	The user programmable write DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_RD_DQS_SLAVE_DELAY	The user programmable read DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 15-361. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_23_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-362. EMIF_EXT_PHY_CONTROL_24

Address Offset	0x0000 02B8		
Physical Address	0x4C00 02B8 0x4D00 02B8	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31	30:24	23:17	16	15:13	12	11:9	8:0
RESERVED	REG_PHY_DQ_OFFSET_HI	RESERVED	REG_PHY_GATELVL_INIT_MODE	RESERVED	REG_PHY_USE_RANK0_DELAYS	RESERVED	REG_PHY_WR_DATA_SLAVE_DELAY

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:24	REG_PHY_DQ_OFFSET_HI	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY ECC data macro. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
23:17	RESERVED		R	0x0
16	REG_PHY_GATELVL_INIT_MODE	The user programmable init ratio selection mode. Recommended value is 0x1.	RW	0x1
15:13	RESERVED		R	0x0
12	REG_PHY_USE_RANK0_DELAYS	Delay selection. Chip select 0 delay line ratios are used for all chip selects when set to 1. Each chip select uses its own delays when set to 0.	RW	0x0
11:9	RESERVED		R	0x0
8:0	REG_PHY_WR_DATA_SLAVE_DELAY	The user programmable write DQ delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 15-363. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_24

EMIF Controller

- Global Initialization: [0] [1] [2] [3] [4] [5] [6]
- EMIF Register Summary: [7]

Table 15-364. EMIF_EXT_PHY_CONTROL_24_SHADOW

Address Offset	0x0000 02BC		
Physical Address	0x4C00 02BC 0x4D00 02BC	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	REG_PHY_DQ_OFFSET_HI	RESERVED	REG_PHY_GATELVL_INIT_MODE	RESERVED	REG_PHY_USE_RANK0_DELAYS	RESERVED	REG_PHY_WR_DATA_SLAVE_DELAY

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:24	REG_PHY_DQ_OFFSET_HI	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY ECC data macro. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
23:17	RESERVED		R	0x0
16	REG_PHY_GATELVL_INIT_MODE	The user programmable init ratio selection mode. Recommended value is 0x1.	RW	0x1
15:13	RESERVED		R	0x0
12	REG_PHY_USE_RANK0_DELAYS	Delay selection. Chip select 0 delay line ratios are used for all chip selects when set to 1. Each chip select uses its own delays when set to 0.	RW	0x0
11:9	RESERVED		R	0x0
8:0	REG_PHY_WR_DATA_SLAVE_DELAY	The user programmable write DQ delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 15-365. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_24_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-366. EMIF_EXT_PHY_CONTROL_25

Address Offset	0x0000 02C0		
Physical Address	0x4C00 02C0 0x4D00 02C0	**Instance**	EMIF1 EMIF2
Description	DQ DLL Slave Ratio Offset Register		
Type	RW		

31:28	27:21	20:14	13:7	6:0
RESERVED	REG_PHY_DQ_OFFSET3	REG_PHY_DQ_OFFSET2	REG_PHY_DQ_OFFSET1	REG_PHY_DQ_OFFSET0

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27:21	REG_PHY_DQ_OFFSET3	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 3. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
20:14	REG_PHY_DQ_OFFSET2	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 2. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
13:7	REG_PHY_DQ_OFFSET1	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 1. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
6:0	REG_PHY_DQ_OFFSET0	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 0. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0

Table 15-367. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_25

EMIF Controller

- Software Leveling: [0] [1]
- Global Initialization: [2] [3] [4] [5] [6] [7] [8]
- EMIF Register Summary: [9]
- EMIF Register Description: [10]

Table 15-368. EMIF_EXT_PHY_CONTROL_25_SHADOW

Address Offset	0x0000 02C4		
Physical Address	0x4C00 02C4 0x4D00 02C4	**Instance**	EMIF1 EMIF2
Description	DQ DLL Slave Ratio Offset Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_25 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				REG_PHY_DQ_OFFSET3							REG_PHY_DQ_OFFSET2							REG_PHY_DQ_OFFSET1							REG_PHY_DQ_OFFSET0						

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27:21	REG_PHY_DQ_OFFSET3	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 3. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
20:14	REG_PHY_DQ_OFFSET2	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 2. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
13:7	REG_PHY_DQ_OFFSET1	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 1. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
6:0	REG_PHY_DQ_OFFSET0	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 0. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0

Table 15-369. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_25_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-370. EMIF_EXT_PHY_CONTROL_26

Address Offset	0x0000 02C8		
Physical Address	0x4C00 02C8 0x4D00 02C8	**Instance**	EMIF1 EMIF2
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	REG_PHY_GATELVL_INIT_RATIO1	RESERVED	REG_PHY_GATELVL_INIT_RATIO0

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO1	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	REG_PHY_GATELVL_INIT_RATIO0	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-371. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_26

EMIF Controller

- Software Leveling: [0] [1] [2]
- Global Initialization: [3] [4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6]

Table 15-372. EMIF_EXT_PHY_CONTROL_26_SHADOW

Address Offset	0x0000 02CC		
Physical Address	0x4C00 02CC 0x4D00 02CC	**Instance**	EMIF1 EMIF2
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_26 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					REG_PHY_GATELVL_INIT_RATIO1											RESERVED					REG_PHY_GATELVL_INIT_RATIO0										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO1	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO0	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-373. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_26_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-374. EMIF_EXT_PHY_CONTROL_27

Address Offset	0x0000 02D0		
Physical Address	0x4C00 02D0 0x4D00 02D0	**Instance**	EMIF1 EMIF2
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		

Table 15-374. EMIF_EXT_PHY_CONTROL_27 (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					REG_PHY_GATELVL_INIT_RATIO3											RESERVED					REG_PHY_GATELVL_INIT_RATIO2										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO3	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO2	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-375. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_27

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-376. EMIF_EXT_PHY_CONTROL_27_SHADOW

Address Offset	0x0000 02D4		
Physical Address	0x4C00 02D4 0x4D00 02D4	**Instance**	EMIF1 EMIF2
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_27 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					REG_PHY_GATELVL_INIT_RATIO3											RESERVED					REG_PHY_GATELVL_INIT_RATIO2										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO3	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	REG_PHY_GATELVL_INIT_RATIO2	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-377. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_27_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-378. EMIF_EXT_PHY_CONTROL_28

Address Offset	0x0000 02D8		
Physical Address	0x4C00 02D8 0x4D00 02D8	**Instance**	EMIF1 EMIF2
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	REG_PHY_GATELVL_INIT_RATIO5	RESERVED	REG_PHY_GATELVL_INIT_RATIO4

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO5	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO4	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-379. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_28

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-380. EMIF_EXT_PHY_CONTROL_28_SHADOW

Address Offset	0x0000 02DC		
Physical Address	0x4C00 02DC 0x4D00 02DC	**Instance**	EMIF1 EMIF2
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_28 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	REG_PHY_GATELVL_INIT_RATIO5	RESERVED	REG_PHY_GATELVL_INIT_RATIO4

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO5	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO4	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-381. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_28_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-382. EMIF_EXT_PHY_CONTROL_29

Address Offset	0x0000 02E0		
Physical Address	0x4C00 02E0 0x4D00 02E0	**Instance**	EMIF1 EMIF2
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					REG_PHY_GATELVL_INIT_RATIO7											RESERVED					REG_PHY_GATELVL_INIT_RATIO6										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO7	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO6	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-383. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_29

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-384. EMIF_EXT_PHY_CONTROL_29_SHADOW

Address Offset	0x0000 02E4		
Physical Address	0x4C00 02E4 0x4D00 02E4	**Instance**	EMIF1 EMIF2
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_29 on any frequency change.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	REG_PHY_GATELVL_INIT_RATIO7	RESERVED	REG_PHY_GATELVL_INIT_RATIO6

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO7	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO6	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-385. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_29_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-386. EMIF_EXT_PHY_CONTROL_30

Address Offset	0x0000 02E8		
Physical Address	0x4C00 02E8	**Instance**	EMIF1
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	REG_PHY_GATELVL_INIT_RATIO9	RESERVED	REG_PHY_GATELVL_INIT_RATIO8

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:16	REG_PHY_GATELVL_INIT_RATIO9	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO8	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-387. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_30

EMIF Controller

- Software Leveling: [0] [1] [2]
- EMIF Register Summary: [3]
- EMIF Register Description: [4]

Table 15-388. EMIF_EXT_PHY_CONTROL_30_SHADOW

Address Offset	0x0000 02EC		
Physical Address	0x4C00 02EC	**Instance**	EMIF1
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_30 on any frequency change.		
Type	RW		

31:27	26:16	15:11	10:0
RESERVED	REG_PHY_GATELVL_INIT_RATIO9	RESERVED	REG_PHY_GATELVL_INIT_RATIO8

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO9	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO8	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 15-389. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_30_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-390. EMIF_EXT_PHY_CONTROL_31

Address Offset	0x0000 02F0		
Physical Address	0x4C00 02F0 0x4D00 02F0	**Instance**	EMIF1 EMIF2
Description	Data macro 0, write DQS DLL Slave Init Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO1	RESERVED	REG_PHY_WRLVL_INIT_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO1	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO0	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-391. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_31

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-392. EMIF_EXT_PHY_CONTROL_31_SHADOW

Address Offset	0x0000 02F4		
Physical Address	0x4C00 02F4 0x4D00 02F4	**Instance**	EMIF1 EMIF2
Description	Data macro 0, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_31 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO1	RESERVED	REG_PHY_WRLVL_INIT_RATIO0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO1	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	REG_PHY_WRLVL_INIT_RATIO0	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-393. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_31_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-394. EMIF_EXT_PHY_CONTROL_32

Address Offset	0x0000 02F8		
Physical Address	0x4C00 02F8 0x4D00 02F8	**Instance**	EMIF1 EMIF2
Description	Data macro 1, write DQS DLL Slave Init Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO3	RESERVED	REG_PHY_WRLVL_INIT_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO3	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO2	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-395. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_32

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-396. EMIF_EXT_PHY_CONTROL_32_SHADOW

Address Offset	0x0000 02FC		
Physical Address	0x4C00 02FC 0x4D00 02FC	**Instance**	EMIF1 EMIF2
Description	Data macro 1, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_32 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	REG_PHY_WRLVL_INIT_RATIO3	RESERVED	REG_PHY_WRLVL_INIT_RATIO2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO3	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO2	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-397. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_32_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-398. EMIF_EXT_PHY_CONTROL_33

Address Offset	0x0000 0300		
Physical Address	0x4C00 0300 0x4D00 0300	**Instance**	EMIF1 EMIF2
Description	Data macro 2, write DQS DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						REG_PHY_WRLVL_INIT_RATIO5										RESERVED						REG_PHY_WRLVL_INIT_RATIO4									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO5	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO4	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-399. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_33

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-400. EMIF_EXT_PHY_CONTROL_33_SHADOW

Address Offset	0x0000 0304		
Physical Address	0x4C00 0304 0x4D00 0304	**Instance**	EMIF1 EMIF2
Description	Data macro 2, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_33 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO5	RESERVED	REG_PHY_WRLVL_INIT_RATIO4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO5	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO4	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-401. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_33_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-402. EMIF_EXT_PHY_CONTROL_34

Address Offset	0x0000 0308		
Physical Address	0x4C00 0308 0x4D00 0308	**Instance**	EMIF1 EMIF2
Description	Data macro 3, write DQS DLL Slave Init Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO7	RESERVED	REG_PHY_WRLVL_INIT_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25:16	REG_PHY_WRLVL_INIT_RATIO 7	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO 6	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-403. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_34

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 15-404. EMIF_EXT_PHY_CONTROL_34_SHADOW

Address Offset	0x0000 030C		
Physical Address	0x4C00 030C 0x4D00 030C	**Instance**	EMIF1 EMIF2
Description	Data macro 3, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_34 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO7	RESERVED	REG_PHY_WRLVL_INIT_RATIO6

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO 7	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO 6	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-405. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_34_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 15-406. EMIF_EXT_PHY_CONTROL_35

Address Offset	0x0000 0310		
Physical Address	0x4C00 0310	**Instance**	EMIF1
Description	ECC Data macro, write DQS DLL Slave Init Ratio Register		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO9	RESERVED	REG_PHY_WRLVL_INIT_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO9	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO8	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-407. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_35

EMIF Controller

- Global Initialization: [0] [1]
- EMIF Register Summary: [2]
- EMIF Register Description: [3]

Table 15-408. EMIF_EXT_PHY_CONTROL_35_SHADOW

Address Offset	0x0000 0314		
Physical Address	0x4C00 0314	**Instance**	EMIF1
Description	ECC Data macro, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_35 on any frequency change.		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	REG_PHY_WRLVL_INIT_RATIO9	RESERVED	REG_PHY_WRLVL_INIT_RATIO8

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO9	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	REG_PHY_WRLVL_INIT_RATIO 8	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 15-409. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_35_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 15-410. EMIF_EXT_PHY_CONTROL_36

Address Offset	0x0000 0318		
Physical Address	0x4C00 0318 0x4D00 0318	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31:11	10	9	8	7:4	3:0
RESERVED	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	REG_PHY_MDLL_UNLOCK_CLR	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	REG_PHY_WRLVL_NUM_OF_DQ0	REG_PHY_GATELVL_NUM_OF_DQ0

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	Clear/reset the phy_reg_rdc_fifo_rst_err_cnt, phy_reg_rdfifo_wrptr and phy_reg_rdfifo_rdptr status flags. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
9	REG_PHY_MDLL_UNLOCK_CLR	Clears the phy_reg_status_mdll_unlock_sticky flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
8	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	Clears the phy_reg_fifo_we_in_misaligned_sticky status flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0

Bits	Field Name	Description	Type	Reset
7:4	REG_PHY_WRLVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the write leveling finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **Note** **NOTE:** In this case *dq0_in* represents all 8 DQ bits OR-ed together.	RW	0x7
3:0	REG_PHY_GATELVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the gate training finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **Note** **NOTE:** In this case *dq0_in* represents the corresponding DQS signal.	RW	0x7

Table 15-411. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_36

EMIF Controller

- Global Initialization: [0] [1] [2] [3] [4]
- EMIF Register Summary: [5]

Table 15-412. EMIF_EXT_PHY_CONTROL_36_SHADOW

Address Offset	0x0000 031C		
Physical Address	0x4C00 031C 0x4D00 031C	**Instance**	EMIF1 EMIF2
Description			
Type	RW		

31:11	10	9	8	7:4	3:0
RESERVED	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	REG_PHY_MDLL_UNLOCK_CLR	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	REG_PHY_WRLVL_NUM_OF_DQ0	REG_PHY_GATELVL_NUM_OF_DQ0

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	Clear/reset the phy_reg_rdc_fifo_rst_err_cnt, phy_reg_rdfifo_wrptr and phy_reg_rdfifo_rdptr status flags. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
9	REG_PHY_MDLL_UNLOCK_CLR	Clears the phy_reg_status_mdll_unlock_sticky flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	Clears the phy_reg_fifo_we_in_misaligned_sticky status flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
7:4	REG_PHY_WRLVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the write leveling finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **Note** **NOTE:** In this case *dq0_in* represents all 8 DQ bits OR-ed together.	RW	0x7
3:0	REG_PHY_GATELVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the gate training finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **Note** **NOTE:** In this case *dq0_in* represents the corresponding DQS signal.	RW	0x7

Table 15-413. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_36_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

15.4 General-Purpose Memory Controller

This section describes the features and functions of the device GPMC controller.

15.4.1 GPMC Overview

The general-purpose memory controller (GPMC) is a unified memory controller dedicated for interfacing with external memory devices like:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (available only in nonmultiplexed mode) burst NOR flash devices
- NAND flash
- Pseudo-SRAM devices

Figure 15-52 shows the GPMC module overview.

The GPMC features are introduced in Section 15.1.4, *GPMC Overview* of Section 15.1, *Memory Subsystem Overview*.

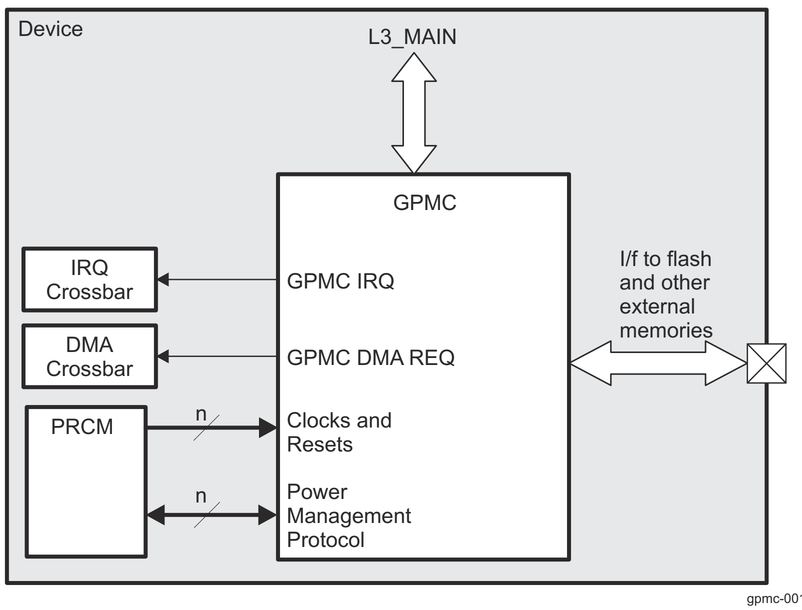


Figure 15-52. GPMC Overview

15.4.2 GPMC Environment

This section describes the GPMC application fields from an environment point of view (external connections). It describes GPMC connectivity options, and gives three possible interfaces.

15.4.2.1 GPMC Modes

This section shows three GPMC external connections options:

- Figure 15-53 shows a connection between the GPMC and a 16-bit synchronous address/data-multiplexed (or AAD-multiplexed, but this protocol uses fewer address pins) external memory device.
- Figure 15-54 shows a connection between the GPMC and a 16-bit synchronous nonmultiplexed external memory device.
- Figure 15-55 shows a connection between the GPMC and an 8-bit synchronous non-multiplexed external memory device.
- Figure 15-56 shows a connection between the GPMC and a 8-bit NAND device.

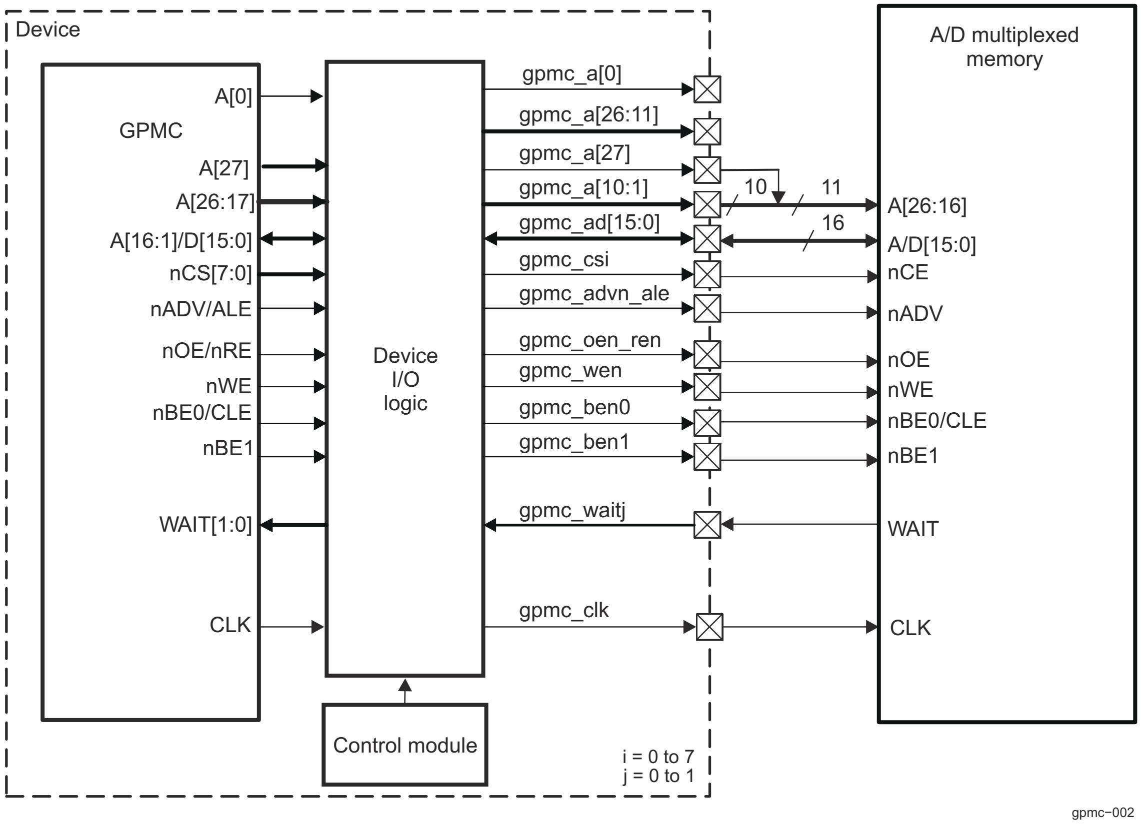


Figure 15-53. GPMC to 16-Bit Address/Data-Multiplexed Memory

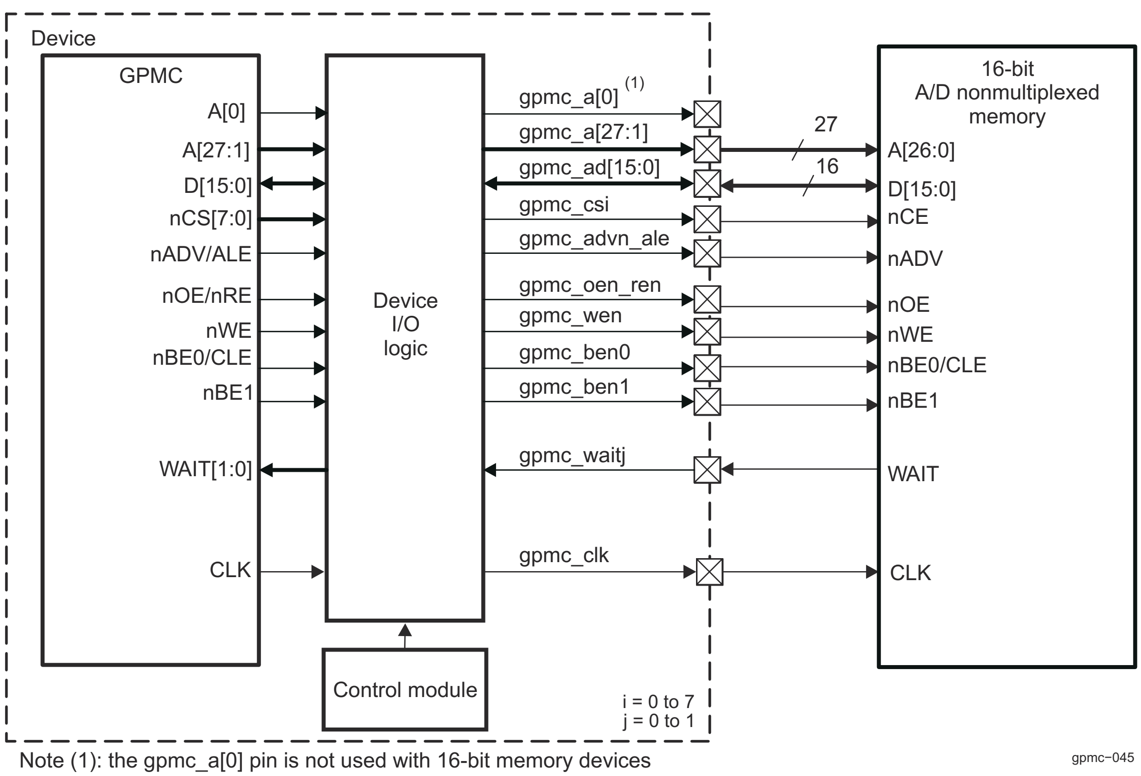


Figure 15-54. GPMC to 16-Bit Nonmultiplexed Memory

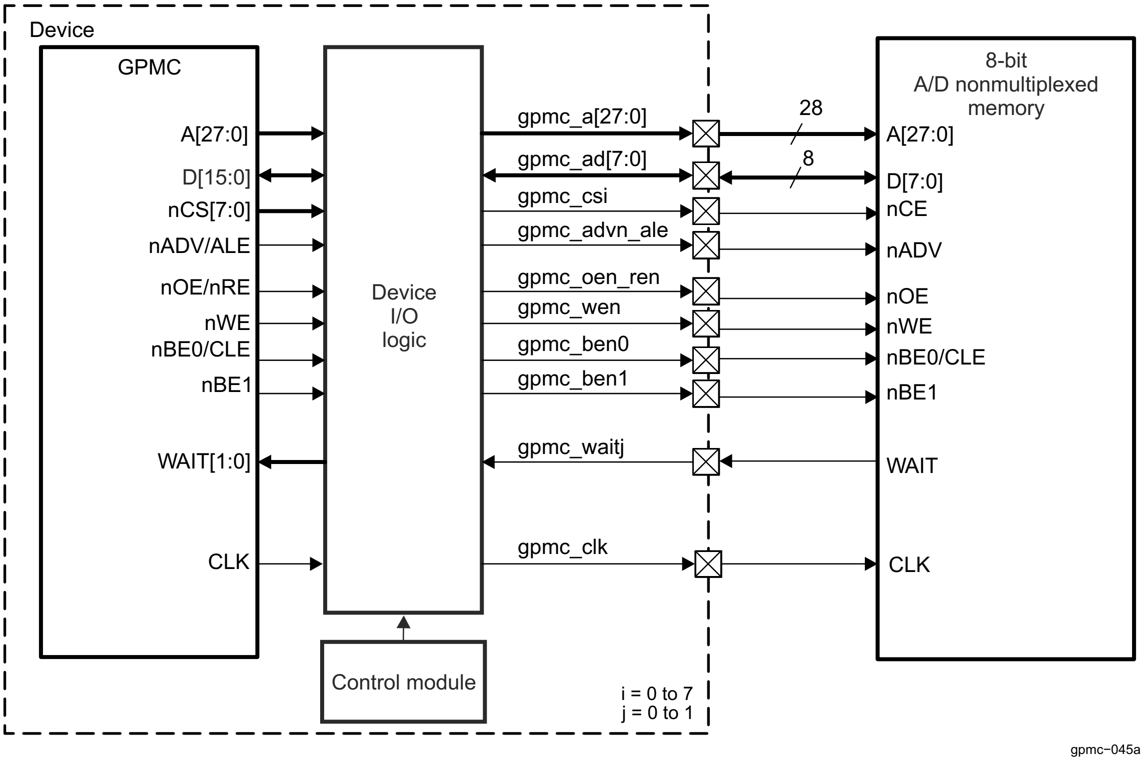


Figure 15-55. GPMC to 8-Bit Nonmultiplexed Memory

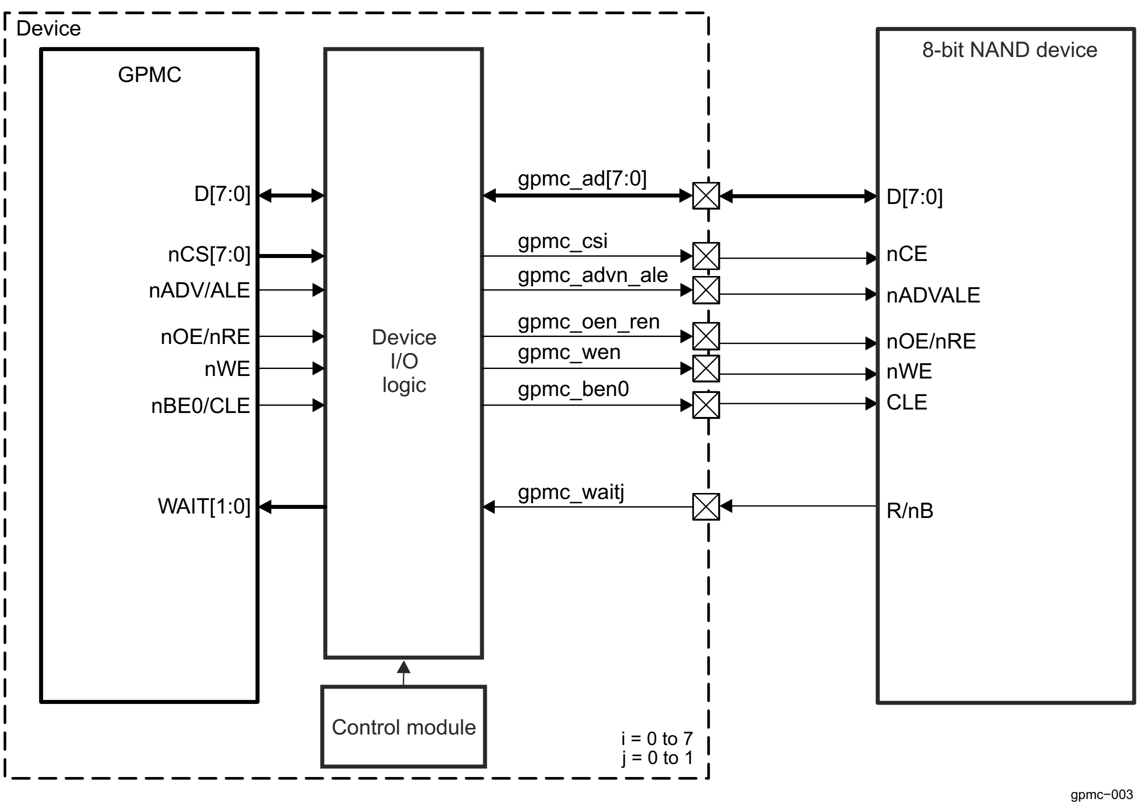


Figure 15-56. GPMC to 8-Bit NAND Device

15.4.2.2 GPMC Signals

Table 15-414 lists the GPMC subsystem input/output (I/O) pins.

Table 15-414. GPMC I/O Description

Pin Name	Device Signal	I/O(1)	Description
A[27:0]	gpmc_a[27:0]	O	28-bit output address bus
A[16:1]/D[15:0]	gpmc_ad[15:0]	I/O	Multiplexed address/data
nCS[7:0]	gpmc_cs[7:0]	O	Chip-selects (active low)
CLK	gpmc_clk	O	Clock generated for the external memory or device
nADV/ALE	gpmc_advn_ale	O	Address valid (active low). Also used as address latch enable (active high) for NAND protocol memories.
nOE/nRE	gpmc_oen_ren	O	Output enable (active low). Also used as read enable (active low) for NAND protocol memories.
nWE	gpmc_wen	O	Write enable (active low)
nBE0/CLE	gpmc_ben0	O	Lower-byte enable (active low). Also used as command latch enable for NAND protocol memories.
nBE1	gpmc_ben1	O	Upper-byte enable (active low).
WAIT[1:0]	gpmc_wait[1:0]	I	External wait signal for NOR and NAND protocol memories. The wait signals can be mapped on any of the chip-select.

(1) I = Input; O = Output

Note

For the gpmc_clk signal to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x register should be set to 0x1 because of retiming purposes.

Note

On SR2.0 devices the internal PU/PD resistors on pads gpmc_a[27:24, 22:19] can be permanently disabled. For more information, see *Permanent PU/PD disabling (SR 2.0 only)* in *Control Module*.

Table 15-415 shows the use of address and data GPMC controller pins based on the type of external device.

Table 15-415. GPMC Pin Multiplexing Options

GPMC Pin	Multiplexed Address Data 16-Bit Device	Nonmultiplexed Address Data 16-Bit Device (complete 28-bit address range)	Nonmultiplexed Address Data 8-Bit Device (complete 28-bit address range)	16-Bit NAND Device	8-Bit NAND Device
gpmc_a[27]	A27	A27	A27	Not used	Not used
gpmc_a[26]	Not used	A26	A26	Not used	Not used
gpmc_a[25]	Not used	A25	A25	Not used	Not used
gpmc_a[24]	Not used	A24	A24	Not used	Not used
gpmc_a[23]	Not used	A23	A23	Not used	Not used
gpmc_a[22]	Not used	A22	A22	Not used	Not used
gpmc_a[21]	Not used	A21	A21	Not used	Not used
gpmc_a[20]	Not used	A20	A20	Not used	Not used
gpmc_a[19]	Not used	A19	A19	Not used	Not used
gpmc_a[18]	Not used	A18	A18	Not used	Not used
gpmc_a[17]	Not used	A17	A17	Not used	Not used
gpmc_a[16]	Not used	A16	A16	Not used	Not used
gpmc_a[15]	Not used	A15	A15	Not used	Not used

Table 15-415. GPMC Pin Multiplexing Options (continued)

GPMC Pin	Multiplexed Address Data 16-Bit Device	Nonmultiplexed Address Data 16-Bit Device (complete 28-bit address range)	Nonmultiplexed Address Data 8-Bit Device (complete 28-bit address range)	16-Bit NAND Device	8-Bit NAND Device
gpmc_a[14]	Not used	A14	A14	Not used	Not used
gpmc_a[13]	Not used	A13	A13	Not used	Not used
gpmc_a[12]	Not used	A12	A12	Not used	Not used
gpmc_a[11]	Not used	A11	A11	Not used	Not used
gpmc_a[10]	A26	A10	A10	Not used	Not used
gpmc_a[9]	A25	A9	A9	Not used	Not used
gpmc_a[8]	A24	A8	A8	Not used	Not used
gpmc_a[7]	A23	A7	A7	Not used	Not used
gpmc_a[6]	A22	A6	A6	Not used	Not used
gpmc_a[5]	A21	A5	A5	Not used	Not used
gpmc_a[4]	A20	A4	A4	Not used	Not used
gpmc_a[3]	A19	A3	A3	Not used	Not used
gpmc_a[2]	A18	A2	A2	Not used	Not used
gpmc_a[1]	A17	A1	A1	Not used	Not used
gpmc_a[0][(1)]	A0 - Not used	Not used	A0	Not used	Not used
gpmc_ad[15]	A16/D15	D15	Not used	D15	Not used
gpmc_ad[14]	A15/D14	D14	Not used	D14	Not used
gpmc_ad[13]	A14/D13	D13	Not used	D13	Not used
gpmc_ad[12]	A13/D12	D12	Not used	D12	Not used
gpmc_ad[11]	A12/D11	D11	Not used	D11	Not used
gpmc_ad[10]	A11/D10	D10	Not used	D10	Not used
gpmc_ad[9]	A10/D9	D9	Not used	D9	Not used
gpmc_ad[8]	A9/D8	D8	Not used	D8	Not used
gpmc_ad[7]	A8/D7	D7	D7	D7	D7
gpmc_ad[6]	A7/D6	D6	D6	D6	D6
gpmc_ad[5]	A6/D5	D5	D5	D5	D5
gpmc_ad[4]	A5/D4	D4	D4	D4	D4
gpmc_ad[3]	A4/D3	D3	D3	D3	D3
gpmc_ad[2]	A3/D2	D2	D2	D2	D2
gpmc_ad[1]	A2/D1	D1	D1	D1	D1
gpmc_ad[0]	A1/D0	D0	D0	D0	D0

(1) Used to effectively address 8-bit (only) non-multiplexed memories

With all device types, the GPMC does not drive unnecessary address lines. They stay at their reset value of 0x00.

Address mapping supports address/data-multiplexed 16-bit-wide devices:

- The NOR flash memory controller still supports nonmultiplexed address and data memory devices.
- Multiplexing mode can be selected through the GPMC_CONFIG1_i[9:8] MUXADDDATA bit field (where i = 0 to 7).
- Asynchronous page mode is not supported for multiplexed address and data devices.

15.4.3 GPMC Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

- No master standby protocol
- Supports slave idle protocol with device PRCM
- Supports Auto Idle
- No wake-up request
- One direct memory access request mapped via the device DMA crossbar (DMA_CROSSBAR) to all device integrated DMA controllers
- One interrupt request mapped via the device interrupt crossbar (IRQ_CROSSBAR) to all device integrated interrupt controllers (MPU_INTC, etc.)
- One clock for functional and interface domains

Figure 15-57 shows GPMC integration.

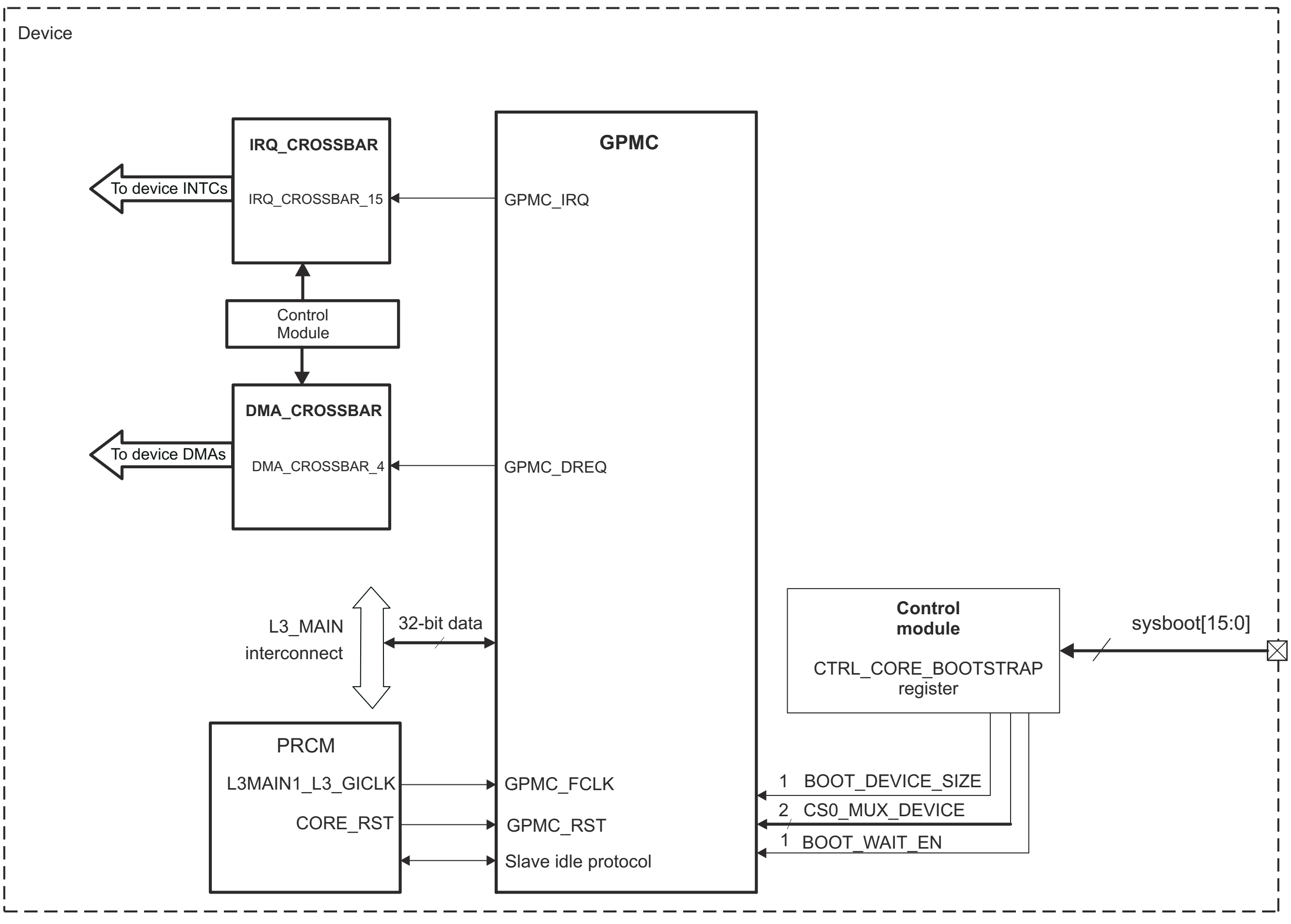


Figure 15-57. GPMC Integration

Table 15-416, Table 15-417 and Table 15-418 summarize the integration of the module in the device.

Table 15-416. GPMC Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
GPMC	PD_COREAON	No	L3_MAIN

Note

- For the description of the operation performance point (OPP) configuration, see *Module Level Clock Management*, in *Power, Reset, and Clock Management*.
- For information about frequencies associated with each OPP, see the device data manual.

Table 15-417. GPMC Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GPMC	GPMC_FCLK	L3MAIN1_L3_GICLK	PRCM	Functional clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GPMC	GPMC_RST	CORE_RET_RST	PRCM	GPMC reset

Note

For the clock description, see Section 15.4.4.2, *GPMC Clock Configuration*.

Table 15-418. GPMC Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
GPMC	GPMC_IRQ	IRQ_CROSSBAR_15	MPU_IRQ_20 DSP1_IRQ_46 DSP2_IRQ_46	GPMC interrupt
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
GPMC	GPMC_DREQ	DMA_CROSSBAR_4	DMA_EDMA_DREQ_3 DMA_SYSTEM_DREQ_3	DMA request from GPMC prefetch engine.

Note

The “**Default Mapping**” column in Table 15-418 *GPMC Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.
For more information about the device DMA_SYSTEM module, see Section 16.1, *System DMA*.
For more information about the device EDMA module, see Section 16.2, *Enhanced DMA*.

Note

For the description of the interrupt source, see Section 15.4.4.5, *GPMC Interrupt Requests*.

15.4.4 GPMC Functional Description

The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the eight configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device:

- Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices.
- The address and data bus can be multiplexed on the same external bus.
- Read and write access can be independently defined as asynchronous or synchronous.
- System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support).
- System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode.
- To simulate a programmable internal-wait-state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access.

Each control signal is controlled independently for each chip-select. The internal functional clock of the GPMC (GPMC_FCLK) is used as a time reference to specify the following:

- Read- and write-access duration
- Most GPMC external interface control-signal assertion and deassertion times
- Data-capture time during read access
- External wait-pin monitoring time
- Duration of idle time between accesses, when required

15.4.4.1 GPMC Block Diagram

Figure 15-58 shows the GPMC functional block diagram. The GPMC consists of six blocks:

- L3 interconnect port interface
- Address decoder, GPMC configuration, and chip-select configuration register file
- Access engine
- Prefetch and write-posting engine
- Error correction code engine (ECC)
- External device/memory port interface

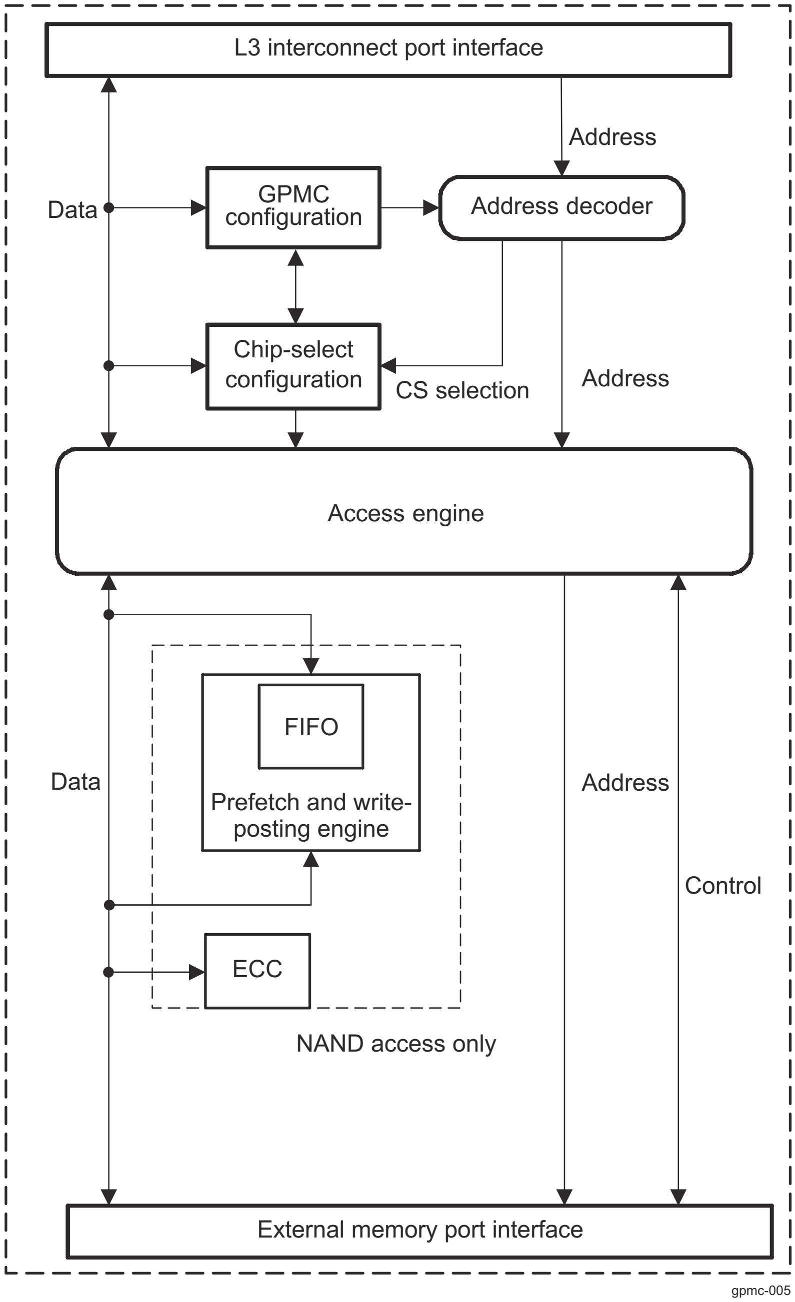


Figure 15-58. GPMC Block Diagram

The GPMC can access various external devices. The flexible programming model allows a wide range of attached device types and access schemes.

Based on the programmed configuration bit fields stored in the GPMC registers, the GPMC can generate the timing of all control signals depending on the attached device and access type.

Given the chip-select decoding and its associated configuration registers, the GPMC selects the appropriate control signal timing for the device type.

15.4.4.2 GPMC Clock Configuration

Table 15-419 describes the GPMC clocks.

Table 15-419. GPMC Clocks

Signal	I/O[1]	Description
GPMC_FCLK	I	Functional and interface clock
gpmc_clk	O	External clock provided to synchronous external memory devices

(1) I = Input; O = Output

The gpmc_clk is generated by the GPMC from the internal GPMC_FCLK clock. The source of the GPMC_FCLK is described in Table 15-417. The gpmc_clk is configured using the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field (where i = 0 to 7), as shown in Table 15-420.

Table 15-420. gpmc_clk Configuration

Source Clock	GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER	gpmc_clk Generated Clock Provided to External Memory Device
GPMC_FCLK	00	GPMC_FCLK
	01	GPMC_FCLK/2
	10	GPMC_FCLK/3
	11	GPMC_FCLK/4

15.4.4.3 GPMC Software Reset

The GPMC can be reset by software through the GPMC_SYSCONFIG[1] SOFTRESET bit. Setting the bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Hardware and software resets initialize all GPMC registers and the finite state-machine (FSM) immediately and unconditionally. The GPMC_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. Software must ensure that the software reset completes before performing GPMC operations.

15.4.4.4 GPMC Power Management

GPMC power is supplied by the CORE power domain, and GPMC power management complies with system power-management guidelines.

Table 15-421 describes the power-management features available for the GPMC module.

Note

- For information about source clock gating and sleep/wake-up transitions, see *Clock Management*, in *Power, Reset, and Clock Management*.
- For descriptions of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see *Clock Management Functional Description*, in *Power, Reset, and Clock Management*.

Table 15-421. GPMC Local Power-Management Features

Feature	Registers	Description
Clock autogating	GPMC_SYSCONFIG[0] AUTOIDLE	This bit allows a local power optimization inside the module, by gating the GPMC_FCLK clock upon the internal activity.
Slave idle modes	GPMC_SYSCONFIG[4:3] SIDLEMODE	Force-idle, no-idle and smart-idle wake-up modes are available.
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	N/A	Feature not available

15.4.4.5 GPMC Interrupt Requests

The GPMC generates one interrupt request (see Figure 15-57).

Table 15-422 lists the event flags, and their mask, that can cause module interrupts.

Table 15-422. GPMC Interrupt Events

Event Flag	Event Mask	Sensitivity	Description
GPMC_IRQSTATUS[9] WAIT1EDGE DETECTIONSTATUS	GPMC_IRQENABLE[9] WAIT1EDGE DETECTIONENABLE	Edge	Wait1 edge detection interrupt: Triggered if a rising or falling edge is detected on the gpmc_wait1 signal. The rising or falling edge detection of Wait1 is selected through the GPMC_CONFIG[9] WAIT1PINPOLARITY bit.
GPMC_IRQSTATUS[8] WAIT0EDGE DETECTIONSTATUS	GPMC_IRQENABLE[8] WAIT0EDGE DETECTIONENABLE	Edge	Wait0 edge detection interrupt: Triggered if a rising or falling edge is detected on the gpmc_wait0 signal. The rising or falling edge detection of Wait0 is selected through the GPMC_CONFIG[8] WAIT0PINPOLARITY bit.
GPMC_IRQSTATUS[1] TERMINAL COUNTSTATUS	GPMC_IRQENABLE[1] TERMINAL COUNTENABLE	Level	Terminal count event: Triggered on prefetch process completion; that is, when the number of currently remaining data to be requested reaches 0
GPMC_IRQSTATUS[0] FIFOEVENTSTATUS	GPMC_IRQENABLE[0] FIFOEVENTENABLE	Level	FIFO event interrupt: Indicates available FIFO levels for write-posting mode and prefetch mode. GPMC_PREFETCH_CONFIG1[2] DMAMODE must be set to 0.

15.4.4.6 L3 Interconnect Interface

The GPMC L3 interconnect interface is a pipelined interface including a 16 × 32-bit word write buffer.

Any system host can issue external access requests through the GPMC.

The device system can issue the following requests through this interface:

- One 8-, 16-, or 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

Only linear burst transactions are supported; interleaved burst transactions are not supported. Only power-of-two-length precise bursts 2 × 32, 4 × 32, 8 × 32, and 16 × 32, with the burst base address aligned on the total burst size, are supported (this limitation applies to incrementing bursts only).

This interface also provides one interrupt and one DMA request line for specific event control.

It is recommended to program the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field according to the page length of the effective attached device and to enable the GPMC_CONFIG1_i[31] WRAPBURST bit if the attached device supports wrapping burst.

It is possible, however, to emulate wrapping burst on a nonwrapping memory by providing relevant addresses within the page or by splitting transactions. Bursts larger than the memory page length are chopped into multiple burst transactions. Because of the alignment requirements, a page boundary is never crossed.

15.4.4.7 GPMC Address and Data Bus

The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Connection to address/data-nonmultiplexed memories or devices is supported with an address range up to 256 MB (that is, 28 address bits in 8-bit mode or 27 address bits in 16-bit mode).

Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus).

- For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus.
- 8-bit-wide NOR devices do not use GPMC I/O: gpmc_ad[15:8] for data (they are used for address if needed).

- 16-bit-wide NAND devices do not use GPMC I/O: gpmc_a[27:0].
- 8-bit-wide NAND devices do not use GPMC I/O: gpmc_a[27:0] and GPMC I/O: gpmc_ad[15:8].

15.4.4.7.1 GPMC I/O Configuration Setting

Note

In this section and the following sections, the i in GPMC_CONFIGx_i stands for the GPMC chip-select i, where i = 0 to 7.

To select a NAND device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b10
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b00

To select an address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b00
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b10

To select an address/address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b00
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b01

To select an address/data-nonmultiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b00
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b00

15.4.4.7.2 GPMC CS0 Default Configuration at Device Reset

To ensure a correct, fast external boot (see *Fast External Booting*, in *Initialization*) with a GPMC access on device reset, several pins are sampled:

- The "sysboot0" through "sysboot5" pins (device boundary) define the sequence of interfaces and devices to use for booting (i.e. SYSBOOT[5:0] vector). They are sampled by the control module at reset and used later by the device ROM code. For more information, see *Sysboot Configuration*, in *Initialization*.
- Additional pins are used to configure reset values in the GPMC_CONFIG1_i register (where i = 0) as explained in the following and in Table 15-423:
  - The *bootdevicesize* input pin (at the GPMC boundary) defines the size of the attached device on chip-select 0 (CS0) and is used to configure the GPMC_CONFIG1_i[13:12] DEVICESIZE bit field (where i = 0). The BOOT_DEVICE_SIZE signal is propagated from the device Control Module. Its value 0b0 (8-bit memories) or 0b1 (16-bit memories) can be externally determined upon booting by **user hardware** via the device external input signal - "sysboot13".
  - The *cs0muxdevice* input pin (at the GPMC boundary) selects whether or not the device attached to CS0 is an address/data-multiplexed device. The input pin is used to configure the GPMC_CONFIG1_i[9:8] MUXADDDATA bit field (where i = 0). The CS0_MUX_DEVICE[1:0] signal is propagated from the Control Module. Its value 0x0 **(non-muxed memory attached)** or 0x2 **(Addr-Data Mux memory attached)** can be externally determined upon booting by **user hardware** via combining the device external input signals - "sysboot12" and "sysboot11", i.e. SYSBOOT[12:11].
  - The *bootwaiten* input pin (at the GPMC boundary) enables the monitoring on CS0 of the wait pin at device reset release time for read accesses. The input pin is used to configure the GPMC_CONFIG1_i[22] WAITREADMONITORING bit (where i = 0). The BOOT_WAIT_EN signal is propagated from the Control Module. Its value 0x0 **(wait pin is not monitored for read accesses)** or 0x1 **(wait pin is monitored for read accesses)** can be externally determined upon booting by **user hardware** via the device external input signal - "sysboot10".

Note

If WAIT pin monitoring function is enabled upon booting (i.e. BOOT_WAIT_EN="1"), the default (power-on-reset) monitored input for CS0 is always the device gpmc_wait0 input.

Table 15-423. Boot Control Interface Input Signals Description

Signal Name	Width	Description
BOOT_DEVICE_SIZE	1	Size of the device attached on CS0 0b00: 8-bit 0b01: 16-bit 0b10: Reserved (not used) 0b11: Reserved (not used)
CS0_MUX_DEVICE	2	Multiplexing mode of the device on CS0 0b00: Nonmultiplexed device on CS0 0b01: AAD-multiplexed device on CS0 (address-address-data) 0b10: Address/data-multiplexed device on CS0 0b11: Reserved
BOOT_WAIT_EN	1	Wait monitoring on CS0 at device reset release time for read accesses 0: Wait pin is not monitored 1: Wait pin is monitored

CAUTION

Using the internal boot code, the entire CS0 configuration can be modified before the first CS0 access. For more information, see *Memory Booting*, and *Image Format*, in *Initialization*. This modification of internal boot code is necessary for two external devices:

- NAND device attached to CS0
- Nonmultiplexed memory device attached to CS0

At reset time, the device can boot from the internal ROM.

The reset values of the timing control parameters are defined to cope with direct boot on address and data-multiplexed NOR flash devices, nonmultiplexed NOR flash devices, or any asynchronous device with large timing margins, assuming a low GPMC_FCLK frequency (for example, 19.2 MHz) at boot time.

For a multiplexed access, the address 16 low-order bits are presented onto gpmc_ad[15:0], while the high-order bits are presented onto gpmc_a[26:16]. If the external chip interface to the memories uses a 16-bit data bus, the high-order address bits are sampled on the address bus.

The reset values of timing parameters used at boot time are:

- CSONTIME = 1
- CSRDOFFTIME = 16
- ADVONTIME = 4
- ADVRDOFFTIME = 5
- OEONTIME = 6
- OEOFFTIME = 16
- RDACCESSTIME = 15
- RDCYCLETIME = 17

For an AAD-multiplexed access, all address bits are passed onto the data bus using two nADV rising edges. The first rising edge latches the address most-significant bit (MSB) down to bit 17, while the second rising edge latches address bits 16 down to 1. This configuration is only used for 16-bit memories.

The reset values of these timing parameters used at boot time are:

- ADVAADMUXONTIME = 1
- ADVAADMUXRDOFFTIME = 2

- OEAADMUXONTIME = 1
- OEAADMUXOFFTIME = 3

15.4.4.8 Address Decoder and Chip-Select Configuration

Addresses are decoded accordingly with the address request of the chip-select and the content of the chip-select base address register file, which includes a set of global GPMC configuration registers and eight sets of chip-select configuration registers.

The GPMC configuration register file is memory-mapped and can be read or written with byte, 16-bit word, or 32-bit word accesses. The register file must be configured as a noncacheable, nonbufferable region to prevent any desynchronization between host execution (write request) and the completion of register configuration (write completed with register updated). Section 15.4.7, *GPMC Register Manual*, provides the GPMC register locations. For the map of GPMC memory locations, see *Memory Mapping*.

After the chip-select is configured, the access engine accesses the external device, drives the external interface control signals, and applies the interface protocol based on user-defined timing parameters and settings.

15.4.4.8.1 Chip-Select Base Address and Region Size

Any external memory or ASIC device attached to the GPMC external interface can be accessed by any device system host within the GPMC 512-MiB address space. For more information, see *Memory Mapping*.

Note

Even though GPMC supports total address space of 1GB, only 512MB are physically available in this device.

The GPMC 512-MiB address space can be divided into a maximum of eight chip-select regions with programmable base address and programmable chip-select size. The chip-select size is programmable from 16 MiB to 256 MiB (must be a power-of-two) and is defined by the mask field. Attached memory smaller than the programmed chip-select region size is accessed through the entire chip-select region (aliasing).

Each chip-select has a 6-bit base address encoding and 4-bit decoding mask, which must be programmed according to the following rules:

- The programmed chip-select region base address must be aligned on the chip-select region size address boundary and is limited to a power-of-two address value. During access decoding, the value of the register base address is used to compare the address with the address bit line mapping, as shown in Figure 15-59 (with A0 as the device system byte-address line). The base address is programmed through the GPMC_CONFIG7_i[5:0] BASEADDRESS bit field.
- The register mask is used to exclude some address lines from the decoding. A register mask bit field set to 0 suppresses the associated address line from the address comparison (incoming address bit line is don't care). The value of the register mask must be limited to the subsequent value, based on the desired chip-select region size. Any other value has an undefined result. When multiple chip-select regions with overlapping addresses are enabled concurrently, access to these chip-select regions is cancelled and a GPMC access error is posted. The mask field is programmed through the GPMC_CONFIG7_i[11:8] MASKADDRESS bit field.

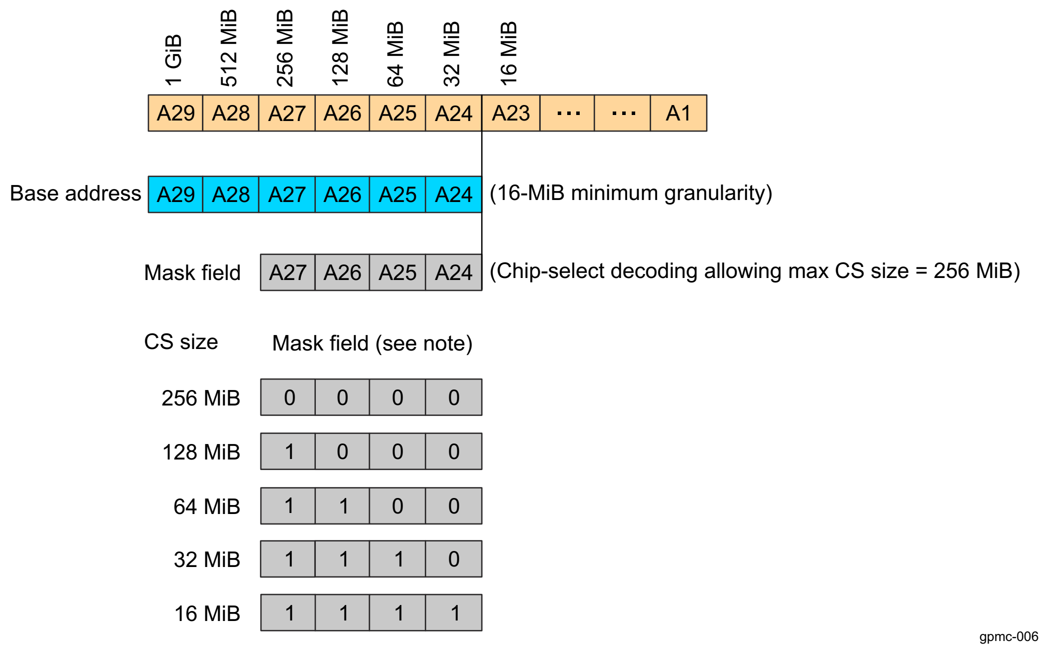


Figure 15-59. Chip-Select Address Mapping and Decoding Mask

Following is an example mapping that divides the 512MB memory reach into two 256MB regions:

- 1st 256MB region – 0x0000 0000 to 0x0FFF FFFF – program base address for that chip select equal to 0x0
- 2nd 256MB region – 0x1000 0000 to 0x1FFF FFFF – program base address for that chip select equal to 0x10

Chip-select configuration (base and mask address or any protocol and timing settings) must be performed while the associated chip-select is disabled through the GPMC_CONFIG7_i[6] CSVALID bit (where i stands for the GPMC chip-select value, 0 to 7). In addition, a chip-select configuration can be disabled only if there is no ongoing access to that chip-select. This requires monitoring the activity of the prefetch or write-posting engine if the engine is active on the chip-select. Also, the write buffer state must be monitored to wait for any posted write completion to the chip-select.

Any access attempted to a nonvalid GPMC address region (CSVALID disabled or address decoding outside a valid chip-select region) is not propagated to the external interface and a GPMC access error is posted. In case of overlapping chip-selects, an error is generated and no access occurs on either chip-select.

CS0 is the only chip-select region enabled after a power up or GPMC reset.

CAUTION

Although the GPMC interface can drive up to eight chip-selects, the frequency specified for this interface is for a specific load. If this load is exceeded, the maximum frequency cannot be reached. One solution is to implement a board with buffers to allow the slowest device to maintain the total load on the lines .

15.4.4.8.2 Access Protocol

15.4.4.8.2.1 Supported Devices

The access protocol of each chip-select can be independently specified through the GPMC_CONFIG1_i[11:10] DEVICETYPE parameter (where i = 0 to 7) for:

- Random-access synchronous or asynchronous memory, such as NOR flash and SRAM
- NAND flash asynchronous devices

Note

For more information about the NAND flash GPMC basic programming model and NAND support, see Section 15.4.4.12, *NAND Device Basic Programming Model*, and Section 15.4.4.12.1, *NAND Memory Device in Byte or Word16 Stream Mode*.

15.4.4.8.2.2 Access Size Adaptation and Device Width

Each chip-select can be independently configured through the GPMC_CONFIG1_i[13:12] DEVICESIZE bit field (where i = 0 to 7) to interface with a 16- or 8-bit-wide device. System requests with data width greater than the external device data bus width are split into successive accesses according to the external device data-bus width and little-endian data organization.

15.4.4.8.2.3 Address/Data-Multiplexing Interface

For random synchronous or asynchronous memory interfacing (DEVICETYPE = 0b00), an address- and data-multiplexing protocol can be selected through the GPMC_CONFIG1_i[9:8] MUXADDDATA bit field (where i = 0 to 7). The nADV signal must be used as the external device address latch control signal. For the associated chip-select configuration, nADV assertion and deassertion time and nOE assertion time must be set to the appropriate value to meet the address latch setup/hold time requirements of the external device. See Section 15.4.3, *GPMC Integration*.

Note

This address/data-multiplexing interface is not applicable to NAND device interfacing. NAND devices require a specific address, command, and data-multiplexing protocol. See Section 15.4.4.12, *NAND Device Basic Programming Model*.

15.4.4.8.3 External Signals

Note

The "DIR" (direction control) output signal is NOT pinned out on any of the device pads. It is an internal signal only representing a signal direction on the GPMC data bus.

15.4.4.8.3.1 Wait Pin Monitoring Control

GPMC access time can be dynamically controlled using an external gpmc_wait pin when the external device access time is not deterministic and cannot be defined and controlled using only the GPMC internal RDACCESSTIME, WRACCESSTIME, and PAGEBURSTACCESSTIME wait-state generator.

The GPMC features two input wait pins: gpmc_wait1, and gpmc_wait0. These pins allow control of external devices with different wait pin polarity. They also allow the overlap of wait pin assertion from different devices without affecting access to devices for which the wait pin is not asserted.

- The GPMC_CONFIG1_i[17:16] WAITPINSELECT bit field (where i = 0 to 7) selects which input gpmc_wait pin is used for the device attached to the corresponding chip-select.
- The polarity of the wait pin is defined through the WAITxPINPOLARITY bit of the GPMC_CONFIG register. A wait pin configured to be active low means that low level on the WAIT signal indicates that the data is not ready and that the data bus is invalid. When a wait pin is inactive, data is valid.

The GPMC access engine can be configured per chip-select to monitor or not the wait pin of the external memory device, based on the access type: read or write.

- The GPMC_CONFIG1_i[22] WAITREADMONITORING bit defines whether or not the wait pin must be monitored during read accesses.
- The GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit defines whether or not the wait pin must be monitored during write accesses.

The GPMC access engine can be configured to monitor the wait pin of the external memory device asynchronously or synchronously with the GPMC_CLK clock, depending on the access type: synchronous or asynchronous (the GPMC_CONFIG1_i[29] READTYPE and GPMC_CONFIG1_i[27] WRITETYPE bits).

15.4.4.8.3.1.1 Wait Monitoring During Asynchronous Read Access

When wait pin monitoring is enabled for read accesses (WAITREADMONITORING), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state.

During asynchronous read accesses with wait pin monitoring enabled, the wait pin must be at a valid level (asserted or deasserted) for at least two GPMC clock cycles before RDACCESSTIME completes, to ensure correct dynamic access-time control through wait pin monitoring. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

In this context, RDACCESSTIME is used as a wait invalid timing window and is set to such a value that the wait pin is at a valid state two GPMC clock cycles before RDACCESSTIME completes.

Similarly, during a multiple-access cycle (for example, asynchronous read page mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-deasserted state. Wait monitoring pipelining is also applicable to multiple accesses (access within a page).

- Wait monitored as active freezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, wait monitored as asserted extends the current access time in the page. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- Wait monitored as inactive unfreezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, wait monitored as inactive completes the current access time and starts the next access phase in the page. The data bus is considered valid, and data are captured during this clock cycle. In case of a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their related control timing value and according to the CYCLETIME counter status.

When a delay larger than two GPMC clocks must be observed between wait-pin deactivation time and data valid time (including the required GPMC and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data-capture time and the effective unlock of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME bit field (where i = 0 to 7).

Note

- The WAITMONITORINGTIME parameter does not delay the wait pin active or inactive detection, nor does it modify the two GPMC clocks pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and no GPMC_CLK clock is provided to the external device. Still, because GPMCFCLKDIVIDER is used as a divider for the GPMC clock, it must be programmed to define the correct WAITMONITORINGTIME delay.

Figure 15-60 shows wait behavior during an asynchronous single read access.

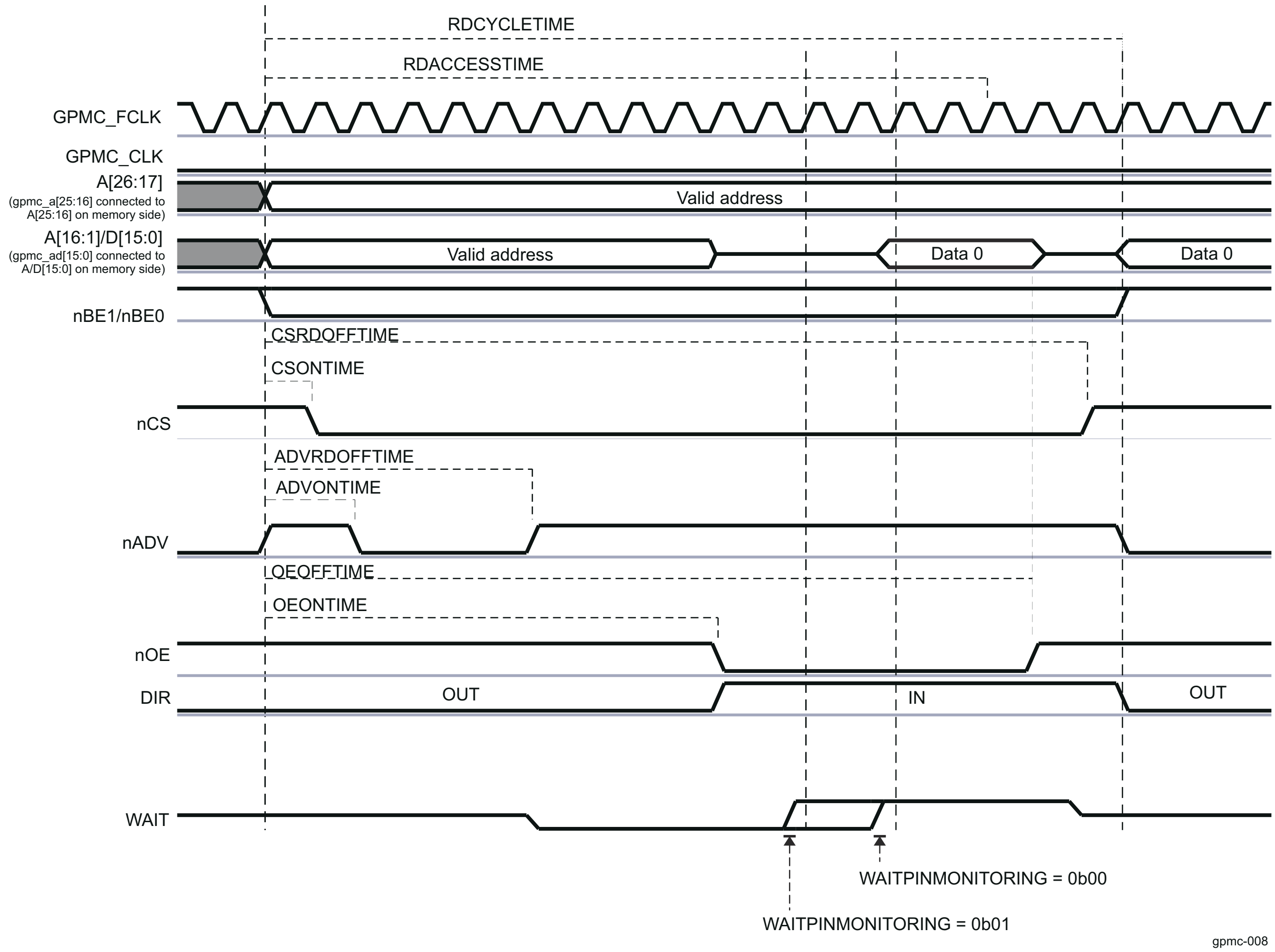


Figure 15-60. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1)

Note

The WAIT signal is active low. GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME = 0b00, or 0b01.

15.4.4.8.3.1.2 Wait Monitoring During Asynchronous Write Access

When wait pin monitoring is enabled for write accesses (GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit = 0x1), the wait invalid timing window is defined by the WRACCESSTIME field. WRACCESSTIME must be set so that the wait pin is at a valid state two GPMC clock cycles before WRACCESSTIME completes. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

- Wait monitored as active freezes the CYCLETIME counter. This informs the GPMC that the data bus is not captured by the external device. The control signals are kept in their current state. The data bus still drives the data.
- Wait monitored as inactive unfreezes the CYCLETIME counter. This informs that the data bus is correctly captured by the external device. All signals, including the data bus, are controlled according to their related control timing value and to the CYCLETIME counter status.

When a delay larger than two GPMC clock cycles must be observed between wait-pin deassertion time and the effective data write into the external device (including the required GPMC data setup time and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data write time into the external device and the effective unfreezing of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME bit field (where i = 0 to 7).

Note

- The WAITMONITORINGTIME parameter does not delay the wait pin assertion or deassertion detection, nor does it modify the two GPMC clock cycles pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and even though no clock is provided to the external device. Still, because the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field is used as a divider for the GPMC clock, it must be programmed to define the correct WAITMONITORINGTIME delay.

15.4.4.8.3.1.3 Wait Monitoring During Synchronous Read Access

During synchronous accesses with wait pin monitoring enabled, the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

The WAIT signal can be programmed to apply to the same clock cycle in which it is captured. Alternatively, it can be sampled one or two GPMC_CLK cycles ahead of the clock cycle to which it applies. This pipelining is applicable to the entire burst access and to all data phases in the burst access. This wait pipelining depth is programmed in the GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME bit field (where i = 0 to 7), and is expressed as a number of GPMC_CLK clock cycles.

In synchronous mode, when wait pin monitoring is enabled (the GPMC_CONFIG1_i[22] WAITREADMONITORING bit), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state detection.

Depending on the programmed value of WAITMONITORINGTIME, the wait pin must be at a valid level, either asserted or deasserted:

- In the same clock cycle the data is valid if WAITMONITORINGTIME = 0 (at RDACCESSTIME completion)
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK clock cycles before RDACCESSTIME completion if WAITMONITORINGTIME is not equal to 0

Similarly, during a multiple-access cycle (burst mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the WAIT-INACTIVE state. The wait pipelining-depth programming applies to the whole burst access.

- Wait monitored as active freezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in a lock state), wait monitored as active extends the current access time in the burst. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- Wait monitored as inactive unfreezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in lock state), wait monitored as inactive completes the current access time and starts the next access phase in the burst. The data bus is considered valid, and data are captured during this clock cycle. In a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their relative control timing value and the CYCLETIME counter status.

Figure 15-61 shows wait behavior during a synchronous read burst access.

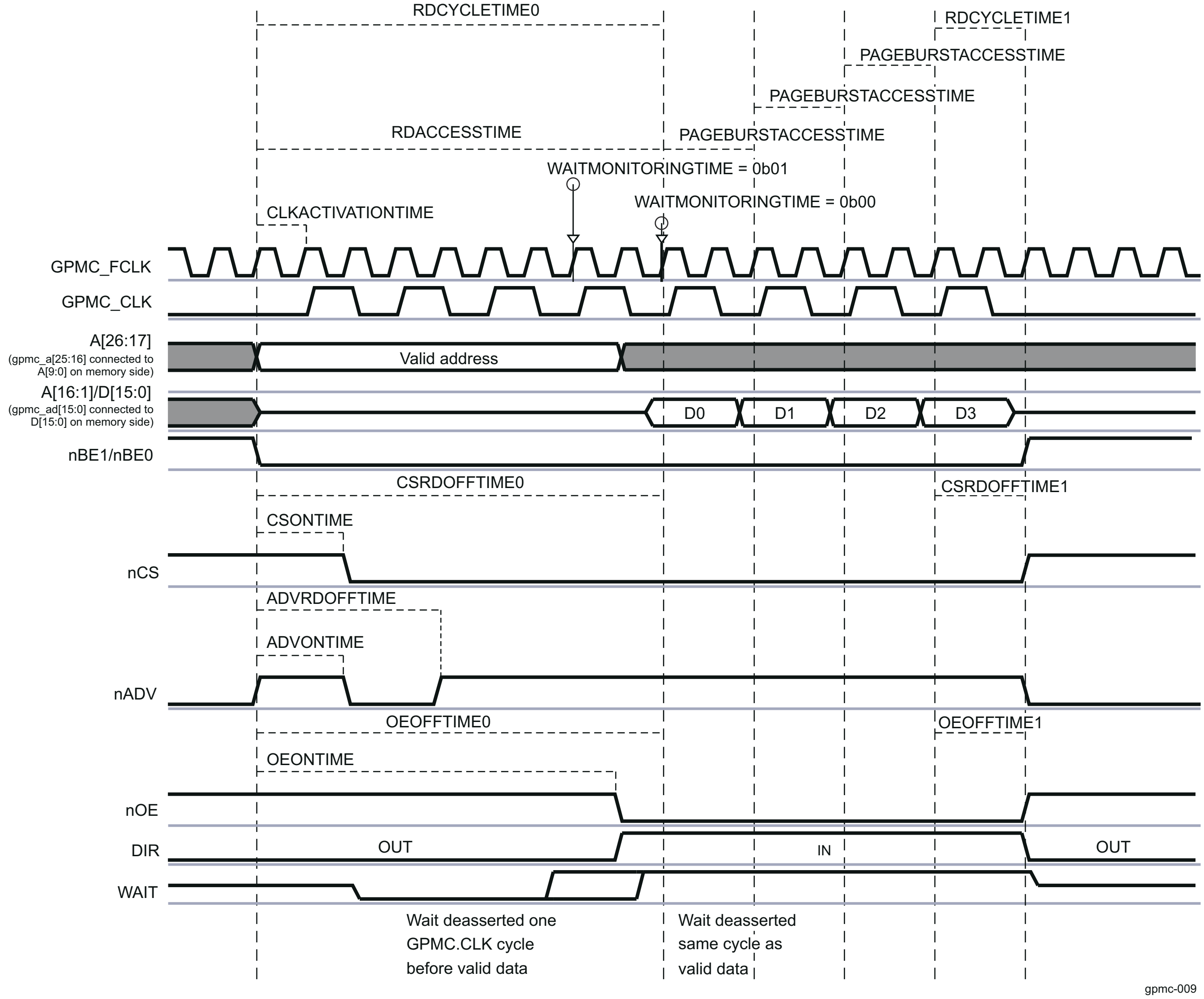


Figure 15-61. Wait Behavior During a Synchronous Read Burst Access

---

Note

The WAIT signal is active low. WAITMONITORINGTIME = 00, 01.

15.4.4.8.3.1.4 Wait Monitoring During Synchronous Write Access

During synchronous accesses with wait pin monitoring enabled (the WAITWRITEMONITORING bit), the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

If enabled, external wait pin monitoring can be used in combination with WRACCESSTIME to delay the GPMC_CLK capture edge of the effective memory device.

Wait-monitoring pipelining depth is similar to synchronous read access:

- At WRACCESSTIME completion if WAITMONITORINGTIME = 0
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK cycles before WRACCESSTIME completion if WAITMONITORINGTIME is not equal to 0

Wait-monitoring pipelining definition applies to whole burst accesses:

- Wait monitored as active freezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, wait monitored as active indicates that the data bus is not being captured by the external device. Control signals are kept in their current state. The data bus is kept in its current state.
- Wait monitored as inactive unfreezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, wait monitored as inactive indicates the effective data capture of the bus by the external device and starts the next access of the burst. In case of a single access or if this was the last access in a multiple access cycle, all signals, including the data bus, are controlled according to their related control timing value and the CYCLETIME counter status.

Note

Wait monitoring is supported for all configurations except GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME = 0x0 (where i = 0 to 7) for write bursts with a clock divider of 1 or 2 (the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field is equal to 0x0 or 0x1, respectively).

15.4.4.8.3.1.5 Wait With NAND Device

For information about the use of the wait pin for communication with a NAND flash external device, see Section 15.4.4.12.2, *NAND Device-Ready Pin*.

15.4.4.8.3.1.6 Idle Cycle Control Between Successive Accesses

15.4.4.8.3.1.6.1 Bus Turnaround (BUSTURNAROUND)

To prevent data-bus contention, an access that follows a read access to a slow memory/device must be delayed (in other words, control the nCS/nOE deassertion to data bus in high-impedance delay).

The bus turnaround is a time-out counter starting after nCS or nOE deassertion time, whichever occurs first, and delays the next access start-cycle time. The counter is programmed through the GPMC_CONFIG6_i[3:0] BUSTURNAROUND bit field (where i = 0 to 7).

After a read access to a chip-select with a nonzero BUSTURNAROUND, the next access is delayed until the BUSTURNAROUND delay completes, if the next access is one of the following:

- A write access to any chip-select (the same or different chip-select from which the data was read)
- A read access to a different chip-select than the chip-select from which the data was read access
- A read or write access to a chip-select associated with an address/data-multiplexed device

Note

Bus keeping starts after bus turnaround completion so that DIR changes from IN to OUT after bus turnaround. The bus does not have enough time to go into high-impedance even though it can be driven with the same value before bus turnaround timing.

BUSTURNAROUND delay runs in parallel with GPMC_CONFIG6_i[3:0] CYCLE2CYCLEDELAY bi field delays. BUSTURNAROUND is a timing parameter for the ending chip-select access, while CYCLE2CYCLEDELAY is a timing parameter for the following chip-select access. The effective minimum delay between successive accesses is driven by these delay timing parameters and by the access type of the following access (see Figure 15-62 through Figure 15-64).

Another way to prevent bus contention is to define an earlier nCS or nOE deassertion time for slow devices or to extend the value of RDCYCLETIME. Doing this prevents bus contention, but it also affects all accesses of this specific chip-select.

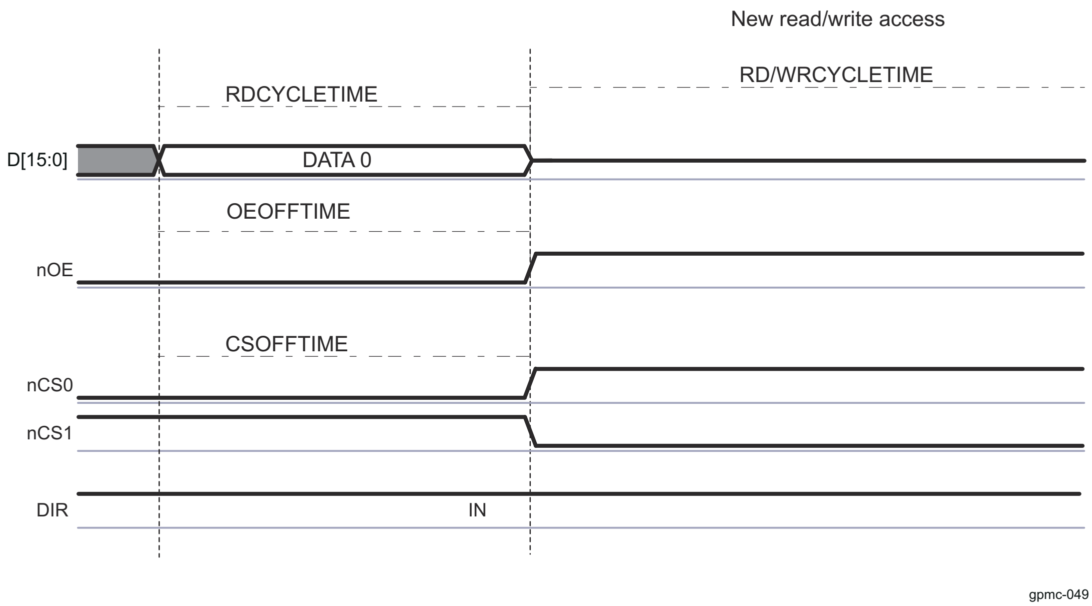


Figure 15-62. Read-to-Read for an Address-Data Multiplexed Device, on Different Chip-Select, Without Bus Turnaround (nCS Attached to a Fast Device)

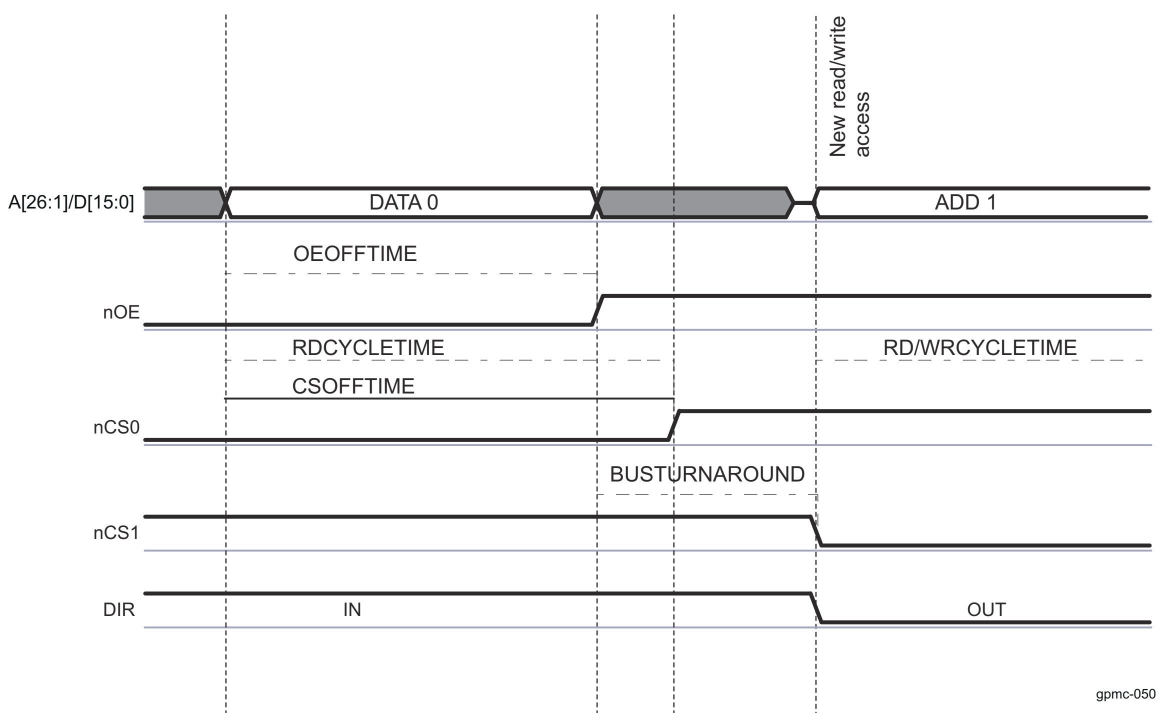


Figure 15-63. Read- to-Read/Write for an Address-Data Multiplexed Device, on Different Chip-Select, With Bus Turnaround

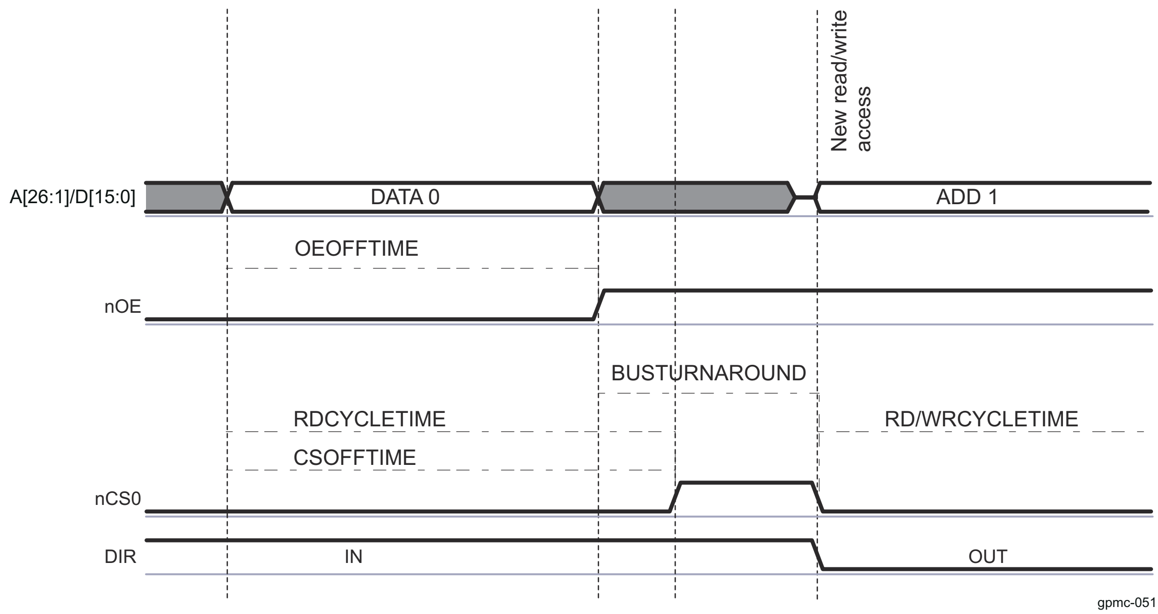


Figure 15-64. Read-to-Read/Write for a Address-Data or AAD-Multiplexed Device, on Same Chip-Select, With Bus Turnaround

15.4.4.8.3.1.6.2 Idle Cycles Between Accesses to Same Chip-Select (CYCLE2CYCLESAMECSEN, CYCLE2CYCLEDELAY)

Some devices require a minimum chip-select signal inactive time between accesses. The GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN bit (where i = 0 to 7) enables insertion of a minimum number of GPMC_FCLK cycles, defined by the GPMC_CONFIG6_i[11:8] CYCLE2CYCLEDELAY bit field, between successive accesses of any type (read or write) to the same chip-select.

If CYCLE2CYCLESAMECSEN is enabled, any subsequent access to the same chip-select is delayed until its CYCLE2CYCLEDELAY completes. The CYCLE2CYCLEDELAY counter starts when CSRDOFFTIME/ CSWROFFTIME completes.

The same applies to successive accesses occurring during 32-bit word or burst accesses split into successive single accesses when the single-access mode is used (GPMC_CONFIG1_i[30] READMULTIPLE = 0 or GPMC_CONFIG1_i[28] WRITEMULTIPLE = 0).

All control signals (CS, ADV#/ALE, BE0#/CLE, WE#, and GPMC.CLK) are kept inactive (ADV#/ALE, BE0#/CLE, and GPMC.CLK at low level; and CS, OE#/RE, and WE at high level) during the idle GPMC.FCLK cycles. This prevents back-to-back accesses to the same chip-select without idle cycles between accesses.

15.4.4.8.3.1.6.3 Idle Cycles Between Accesses to Different Chip-Select (CYCLE2CYCLEDIFFCSEN, CYCLE2CYCLEDELAY)

Because of the pipelined behavior of the system, successive accesses to different chip-selects can occur back-to-back with no idle cycles between accesses. Depending on the control signals (nCS, nADV/ALE, nBE0/ CLE, nOE/RE, nWE) assertion and deassertion timing parameters and on the device timing parameters, the assertion times of some control signals may overlap between the successive accesses to a different chip-select. Similarly, some control signals (WE, OE/RE) may not respect required transition times.

To work around overlapping and to observe the required control-signal transitions, a minimum of CYCLE2CYCLEDELAY inactive cycles is inserted between the access being initiated to this chip-select and the previous access ending for a different chip-select. This applies to any type of access (read or write).

If the GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN bit is enabled, the chip-select access is delayed until CYCLE2CYCLEDELAY cycles have expired since the end of a previous access to a different chip-select.

CYCLE2CYCLEDELAY count starts at CSRDOFFTIME/CSWROFFTIME completion. All control signals are kept inactive during the idle GPMC_FCLK cycles.

Note

CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDIFFCSEN must be set in the GPMC_CONFIG6_i registers to get idle cycles inserted between accesses on this chip-select and after accesses to a different chip-select, respectively.

The CYCLE2CYCLEDELAY delay runs in parallel with the BUSTURNAROUND delay. The BUSTURNAROUND is a timing parameter defined for the ending chip-select access, whereas CYCLE2CYCLEDELAY is a timing parameter defined for the starting chip-select access. The effective minimum delay between successive accesses is based on the larger delay timing parameter and on access type combination, because bus turnaround does not apply to all access types. For more information about bus turnaround, see Section 15.4.4.8.3.1.6.1, *Bus Turnaround (BUSTURNAROUND)*.

Table 15-424 describes the configuration required for idle cycle insertion.

Table 15-424. Idle Cycle Insertion Configuration

1st Access Type	BUSTURNAROUND Timing Parameter	Second Access Type	Chip-Select	Add/Data Multiplexed	CYCLE2CYCLE SAMECSEN Parameter	CYCLE2CYCLE DIFFCSEN Parameter	Idle Cycle Insertion Between the Two Accesses
R/W	= 0	R/W	Any	Any	0	x	No idle cycles are inserted if the two accesses are well pipelined.
R	> 0	R	Same	Nonmuxed	x	0	No idle cycles are inserted if the two accesses are well pipelined.
R	> 0	R	Different	Nonmuxed	0	0	BUSTURNAROUND cycles are inserted.
R	> 0	R/W	Any	Muxed	0	0	BUSTURNAROUND cycles are inserted.
R	> 0	W	Any	Any	0	0	BUSTURNAROUND cycles are inserted.
W	> 0	R/W	Any	Any	0	0	No idle cycles are inserted if the two accesses are well pipelined.
R/W	= 0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted.
R/W	= 0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted.
R/W	> 0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is max (BUSTURNAROUND, CYCLE2CYCLEDELAY).
R/W	> 0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is maximum (BUSTURNAROUND, CYCLE2CYCLEDELAY).

15.4.4.8.3.1.7 Slow Device Support (TIMEPARAGRANULARITY Parameter)

All access-timing parameters can be multiplied by 2 by setting the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit (where i stands for the GPMC chip-select value, 0 to 7). Increasing all access timing parameters allows support of slow devices.

15.4.4.8.3.2 Reset

No reset signal is sent to the external memory device by the GPMC. For more information about external-device reset, see *Power, Reset, and Clock Management*.

The PRCM module provides an input pin, global_rst_n, to the GPMC:

- The global_rst_n pin is activated during device warm reset and cold reset.
- The global_rst_n pin initializes the internal state-machine and the internal configuration registers.

15.4.4.8.3.3 Byte Enable (nBE1/nBE0)

Byte enable signals (nBE1/nBE0) are:

- Valid (asserted or nonasserted according to the incoming system request) from access start to access completion for asynchronous and synchronous single accesses
- Asserted low from access start to access completion for asynchronous and synchronous multiple read accesses
- Valid (asserted or nonasserted, according to the incoming system request) synchronously to each written data for synchronous multiple write accesses

15.4.4.8.4 Error Handling

When an error occurs in the GPMC, the error information is stored in the GPMC_ERR_TYPE register and the address of the illegal access is stored in the GPMC_ERR_ADDRESS register. The GPMC keeps only the first error abort information until the GPMC_ERR_TYPE register is reset. Subsequent accesses that cause errors are not logged until the error is cleared by hardware with the GPMC_ERR_TYPE[0] ERRORVALID bit.

- ERRORNOTSUPPADD occurs when an incoming system request address decoding does not match any valid chip-select region, or if two chip-select regions are defined as overlapped, or if a register file access is tried outside the valid address range of 1KiB.
- ERRORNOTSUPPMCMD occurs when an unsupported command request is decoded at the L3 interconnect interface.
- ERRORTIMEOUT: A time-out mechanism prevents the system from hanging. The start value of the 9-bit time-out counter is defined in the GPMC_TIMEOUT_CONTROL register and enabled with the GPMC_TIMEOUT_CONTROL[0] TIMEOUTENABLE bit. When enabled, the counter starts at start-cycle time until it reaches 0 and data is not responded to from memory, and then a time-out error occurs. When data are sent from memory, this counter is reset to its start value. With multiple accesses (asynchronous page mode or synchronous burst mode), the counter is reset to its start value for each data access within the burst.

The GPMC does not generate interrupts on these errors. True abort to the MPU or interrupt generation is handled at interconnect level.

15.4.4.9 Timing Setting

The GPMC offers maximum flexibility to support various access protocols. Most of the timing parameters of the protocol access used by the GPMC to communicate with attached memories or devices are programmable on a chip-select basis. Assertion and deassertion times of control signals are defined to match the attached memory or device timing specifications and to get maximum performance during accesses. For more information about GPMC_CLK and GPMC_FCLK see Section 15.4.4.9.6, *GPMC_CLK*.

Note

In the following sections, the start access time refers to the time at which the access begins.

15.4.4.9.1 Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME)

The GPMC_CONFIG5_i[4:0] RDCYCLETIME and GPMC_CONFIG5_i[12:8] WRCYCLETIME bit fields (where i = 0 to 7) define the address bus and byte-enable valid times for read and write accesses. To ensure a correct duty cycle of GPMC_CLK between accesses, RDCYCLETIME and WRCYCLETIME are expressed in GPMC_FCLK cycles and must be multiples of the GPMC_CLK cycle. The RDCYCLETIME and WRCYCLETIME bit fields can be set with a granularity of 1 or 2 through the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit.

When RDCYCLETIME or WRCYCLETIME completes, if they are not already deasserted, all control signals (nCS, nADV/ALE, nOE/RE, nWE, and BE0/CLE) are deasserted to their reset values, regardless of their deassertion time parameters.

An exception to this forced deassertion occurs when a pipelined request to the same chip-select or to a different chip-select is pending. In such a case, it is not necessary to deassert a control signal with deassertion time parameters equal to the cycle-time parameter. This exception to forced deassertion prevents any unnecessary glitches. This requirement also applies to BE signals, thus avoiding an unnecessary BE glitch transition when pipelining requests.

Note

All control signals (CS, ADV#/ALE, BE0#/CLE, WE#, and GPMC.CLK) are kept inactive (ADV#/ALE, BE0#/CLE, and GPMC.CLK at low level; and CS, OE#/RE, and WE at high level) during the idle GPMC.FCLK cycles.

If no inactive cycles are required between successive accesses to the same chip-select or a different chip-select (GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN = 0 or GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN = 0, where i = 0 to 7), and if assertion-time parameters associated with the pipelined access are equal to 0, asserted control signals (nCS, nADV/ALE, nBE0/CLE, nWE, and nOE/RE) are kept asserted. This applies to any read/write to read/write access combination.

If inactive cycles are inserted between successive accesses (that is, CYCLE2CYCLESAMECSEN = 1 or CYCLE2CYCLEDIFFCSEN = 1), the control signals are forced to their respective default reset values for the number of GPMC_FCLK cycles defined in CYCLE2CYCLEDELAY.

15.4.4.9.2 nCS: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY)

The GPMC_CONFIG2_i[3:0] CSONTIME bit field (where i = 0 to 7) defines the nCS signal-assertion time relative to the start access time. It is common for read and write accesses.

The GPMC_CONFIG2_i[12:8] CSRDOFFTIME (read access) and GPMC_CONFIG2_i[20:16] CSWROFFTIME (write access) bit fields define the nCS signal deassertion time relative to start access time.

The CSONTIME, CSRDOFFTIME, and CSWROFFTIME parameters apply to synchronous and asynchronous modes. CSONTIME can be used to control an address and byte-enable setup time before chip-select assertion. CSRDOFFTIME and CSWROFFTIME can be used to control an address and byte-enable hold time after chip-select deassertion.

nCS signal transitions, as controlled through CSONTIME, CSRDOFFTIME, and CSWROFFTIME, can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG2_i[7] CSEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on the nCS assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. CSEXTRADELAY is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but it can also be used for all GPMC configurations. If enabled, CSEXTRADELAY applies to all parameters that control nCS transitions.

The CSEXTRADELAY bit must be used carefully to avoid control signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than the nCS signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

15.4.4.9.3 nADV/ALE: Address Valid/Address Latch Enable Signal Control Assertion/ Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME)

The GPMC_CONFIG3_i[3:0] ADVONTIME field (where i = 0 to 7) defines the nADV/ALE signal-assertion time relative to start access time. It is common to read and write accesses.

The GPMC_CONFIG3_i[12:8] ADVRDOFFTIME (read access) and GPMC_CONFIG3_i[20:16] ADVWROFFTIME (write access) bit fields define the nADV/ALE signal-deassertion time relative to start access time.

ADVONTIME can be used to control an address and byte-enable valid setup time control before nADV/ALE assertion. ADVRDOFFTIME and ADVWROFFTIME can be used to control an address and byte-enable valid hold time control after nADV/ALE deassertion. ADVRDOFFTIME and ADVWROFFTIME apply to synchronous and asynchronous modes.

The nADV/ALE signal transitions as controlled through ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG3_i[7] ADVEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on nADV/ALE assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. The ADVEXTRADELAY configuration parameter is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, ADVEXTRADELAY applies to all parameters controlling nADV/ALE transitions.

ADVEXTRADELAY must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than nADV/ALE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME, GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME, and GPMC_CONFIG3_i[30:28] ADVAADMUXWROFFTIME parameters have the same functions as ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME, but apply to the first address phase in the AAD-multiplexed protocol. The user must ensure that ADVAADMUXxxOFFTIME is programmed to a value less than or equal to ADVxxOFFTIME. Functionality in AAD-multiplexed mode is undefined if the settings do not comply with this requirement. ADVAADMUXxxOFFTIME can be programmed to the same value as ADVONTIME if no high nADV pulse is needed between the two AAD-multiplexed address phases, which is the typical case in synchronous mode. In this configuration, nADV is kept low until it reaches the correct ADVxxOFFTIME.

For more information about the use of ADVONTIME, ADVRDOFFTIME, ADVWROFFTIME, and ADVAADMUXRDOFFTIME and ADVAADMUXWROFFTIME for command latch enable (CLE) and address latch enable (ALE) use for a NAND flash interface, see Section 15.4.4.12, *NAND Access Description*.

15.4.4.9.4 nOE/nRE: Output Enable/Read Enable Signal Control Assertion/Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME)

The GPMC_CONFIG4_i[3:0] OEONTIME bit field (where i = 0 to 7) defines the nOE/nRE signal assertion time relative to start access time. It applies only to read accesses.

The GPMC_CONFIG4_i[12:8] OEOFFTIME bit field defines the nOE/nRE signal deassertion time relative to start access time. It applies only to read accesses. nOE/nRE is not asserted during a write cycle.

The OEONTIME, OEOFFTIME, OEAADMUXONTIME, and OEAADMUXOFFTIME parameters apply to synchronous and asynchronous modes. OEONTIME can be used to control an address and byte enable valid setup time control before nOE/nRE assertion. OEOFFTIME can be used to control an address and byte-enable valid hold time control after nOE/nRE assertion.

The OEAADMUXONTIME and OEAADMUXOFFTIME parameters have the same functions as OEONTIME and OEOFFTIME, but apply to the first OE assertion in the AAD-multiplexed protocol for a read phase, or to the only OE assertion for a write phase. The user mut ensure that OEAADMUXOFFTIME is programmed to a value less than OEONTIME. Functionality in AAD-multiplexed mode is undefined if the settings do not comply with this requirement. OEAADMUXOFFTIME must never be equal to OEONTIME because the AAD-multiplexed protocol requires a second address phase with the nOE signal deasserted before nOE can be asserted again to define a read command.

The nOE/RE signal transitions as controlled through OEONTIME, OEOFFTIME, OEAADMUXONTIME, and OEAADMUXOFFTIME can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG4_i[7] OEEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on the nOE/RE assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. If enabled, OEEXTRADELAY applies to all parameters controlling nOE/nRE transitions.

OEEXTRADELAY must be used carefully, to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program RDCYCLETIME and WRCYCLETIME to be greater than the nOE/RE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

Note

When the GPMC generates a read access to an address-/data-multiplexed device, it drives the address bus until nOE assertion time.

15.4.4.9.5 nWE: Write Enable Signal Control Assertion/Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY)

The GPMC_CONFIG4_i[19:16] WEONTIME bit field (where i = 0 to 7) defines the nWE signal-assertion time relative to start access time. The GPMC_CONFIG4_i[28:24] WEOFFTIME bit field defines the nWE signal-deassertion time relative to start access time. These bit fields apply only to write accesses. nWE is not asserted during a read cycle.

WEONTIME can be used to control an address and byte-enable valid setup time control before nWE assertion. WEOFFTIME can be used to control an address and byte-enable valid hold time control after nWE assertion.

nWE signal transitions as controlled through WEONTIME, and WEOFFTIME can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG4_i[23] WEEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on nWE assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. If enabled, WEEXTRADELAY applies to all parameters controlling nWE transitions.

The WEEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the WRCYCLETIME bit field to be greater than the nWE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

15.4.4.9.6 GPMC_CLK

GPMC_CLK is the external clock provided to the attached synchronous memory or device.

- The GPMC_CLK clock frequency is the GPMC_FCLK functional clock frequency divided by 1, 2, 3, or 4, depending on the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field (where i = 0 to 7), with a guaranteed 50-percent duty cycle. For information about the duty cycle error, see the device data manual.
- The GPMC_CLK clock is activated only when the access in progress is defined as synchronous (read or write access).
- The GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME bit field (where i = 0 to 7) defines the number of GPMC_FCLK cycles from start access time to GPMC_CLK activation.
- The GPMC_CLK clock is stopped when cycle time completes and is asserted low between accesses.
- The GPMC_CLK clock is kept low when access is defined as asynchronous.

CAUTION

When the cycle time completes, the GPMC_CLK may be high because of the GPMCFCLKDIVIDER bit field. To ensure correct stoppage of the GPMC_CLK clock within the required 50-percent duty cycle, the user must extend the RDCYCLETIME or WRCYCLETIME value.

Note

To ensure a correct external clock cycle, the following rules must be applied:

- (RDCYCLETIME CLKACTIVATIONTIME) must be a multiple of (GPMCFCLKDIVIDER + 1).
- The PAGEBURSTACCESSTIME value must be a multiple of (GPMCFCLKDIVIDER + 1).

15.4.4.9.7 GPMC_CLK and Control Signals Setup and Hold

Control-signal transition (assertion and deassertion) setup and hold values with respect to the GPMC_CLK edge can be controlled in the following ways:

- For the GPMC_CLK signal, the GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME bit field (where i = 0 to 7) allows setup and hold control of control-signal assertion time.
- The use of a divided GPMC_CLK allows setup and hold control of the control-signal assertion and deassertion times.
- When GPMC_CLK runs at the GPMC_FCLK frequency so that GPMC_CLK edge and control-signal transitions refer to the same GPMC_FCLK edge, the control-signal transitions can be delayed by a half-GPMC_FCLK period to provide minimum setup and hold times. This half-GPMC_FCLK delay is enabled with the CSEXTRADELAY, ADVEXTRADELAY, OEEXTRADELAY, or WEEXTRADELAY parameter. This delay must be used carefully to prevent control-signal overlap between successive accesses to different chip-selects. This implies that the RDCYCLETIME and WRCYCLETIME are greater than the last control-signal deassertion time, including the extra half-GPMC_FCLK cycle.

15.4.4.9.8 Access Time (RDACCESSTIME / WRACCESSTIME)

The read/write access time durations can be programmed independently through the GPMC_CONFIG5_i[20:16] RDACCESSTIME and GPMC_CONFIG6_i[28:24] WRACCESSTIME bit fields (where i = 0 to 7). This allows nOE and GPMC data-capture timing parameters to be independent of nWE and memory device data capture timing parameters. The RDACCESSTIME and WRACCESSTIME bit fields can be set with a granularity of 1 or 2 through the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit.

15.4.4.9.8.1 Access Time on Read Access

In asynchronous read mode, for single and paged accesses, the GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field (where i = 0 to 7) defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge used for the first data capture. RDACCESSTIME must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached memory device.

In synchronous read mode, for single or burst accesses, RDACCESSTIME defines the number of GPMC_FCLK cycles from the start access time to the GPMC_FCLK rising edge corresponding to the GPMC_CLK rising edge used for the first data capture.

GPMC_CLK, which is sent to the memory device for synchronization with the GPMC controller, is internally retimed to correctly latch the returned data. The GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field must be greater than RDACCESSTIME to let the GPMC latch the last return data using the internally retimed GPMC_CLK.

The external WAIT signal can be used in conjunction with RDACCESSTIME to control the effective GPMC data-capture GPMC_FCLK edge on read access in asynchronous and synchronous modes. For more information about wait monitoring, see Section 15.4.4.8.3.1, *Wait Pin Monitoring Control*.

15.4.4.9.8.2 Access Time on Write Access

In asynchronous write mode, the GPMC_CONFIG6_i[28:24] WRACCESSTIME timing parameter is not used to define the effective write access time. Instead, it is used as a wait invalid timing window and must be set to a correct value so that the gpmc_wait pin is at a valid state two GPMC_CLK cycles before WRACCESSTIME completes. For more information about wait monitoring, see Section 15.4.4.8.3.1, *Wait Pin Monitoring Control*.

In synchronous write mode , for single or burst accesses, WRACCESSTIME defines the number of GPMC_FCLK cycles from the start access time to the GPMC_CLK rising edge used by the memory device for the first data capture.

The external WAIT signal can be used in conjunction with WRACCESSTIME to control the effective memory device data-capture GPMC_CLK edge for a synchronous write access. For more information about wait monitoring, see Section 15.4.4.8.3.1, *Wait Pin Monitoring Control*.

15.4.4.9.9 Page Burst Access Time (PAGEBURSTACCESSTIME)

The GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME bit field (where i = 0 to 7) can be set with a granularity of 1 or 2 through theGPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit.

15.4.4.9.9.1 Page Burst Access Time on Read Access

In asynchronous page read mode, the delay between successive word captures in a page is controlled through the PAGEBURSTACCESSTIME bit field. The PAGEBURSTACCESSTIME parameter must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached device.

In synchronous burst read mode, the delay between successive word captures in a burst is controlled through the PAGEBURSTACCESSTIME bit field.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective GPMC data-capture GPMC_FCLK edge on read access. For more information about wait monitoring, see Section 15.4.4.8.3.1, *Wait Pin Monitoring Control*.

15.4.4.9.9.2 Page Burst Access Time on Write Access

Asynchronous page write mode is not supported. PAGEBURSTACCESSTIME is irrelevant in this case.

In synchronous burst write mode, PAGEBURSTACCESSTIME controls the delay between successive memory device word captures in a burst.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective memory device data capture GPMC_CLK edge in synchronous write mode. For more information about wait monitoring, see Section 15.4.4.8.3.1, *Wait Pin Monitoring Control*.

15.4.4.9.10 Bus Keeping Support

At the end cycle time of a read access, if no other access is pending, the GPMC drives the bus with the last data read after RDCYCLETIME completes to prevent bus floating and reduce power consumption.

After a write access, if no other access is pending, the GPMC keeps driving the data bus after WRCYCLETIME completes with the same data to prevent bus floating and power consumption.

15.4.4.10 NOR Access Description

For each chip-select configuration, the read access can be specified as asynchronous or synchronous access through the GPMC_CONFIG1_i[29] READTYPE bit (where i = 0 to 7). For each chip-select configuration, the write access can be specified as synchronous or asynchronous access through the GPMC_CONFIG1_i[27] WRITETYPE bit where (i = 0 to 7).

Asynchronous and synchronous read and write access time and related control signals are controlled through timing parameters that refer to GPMC_FCLK. The primary difference of synchronous mode is the availability of a configurable clock interface (GPMC_CLK) to control the external device. Synchronous mode also affects data-capture and wait-pin monitoring schemes in read access.

For more information about asynchronous and synchronous access, see the descriptions of GPMC_CLK, RdAccessTime, WrAccessTime, and wait pin monitoring.

For more information about timing-parameter settings, see the sample timing diagrams in this chapter.

Note

The address bus and nBE[1:0] are fixed for the duration of a synchronous burst read access, but they are updated for each beat of an asynchronous page-read access.

15.4.4.10.1 Asynchronous Access Description

This section describes:

- Asynchronous single-read operation on an address/data multiplexed device
- Asynchronous single write operation on an address/data-multiplexed device
- Asynchronous single read operation on an AAD-multiplexed device
- Asynchronous single write operation on an AAD-multiplexed device
- Asynchronous multiple (page) read operation on a non-multiplexed device

In asynchronous operations GPMC_CLK is not provided outside the GPMC and is kept low.

15.4.4.10.1.1 Access on Address/Data Multiplexed Devices

15.4.4.10.1.1.1 Asynchronous Single-Read Operation on an Address/Data Multiplexed Device

Figure 15-65 shows an asynchronous single read operation on an address/data-multiplexed device.

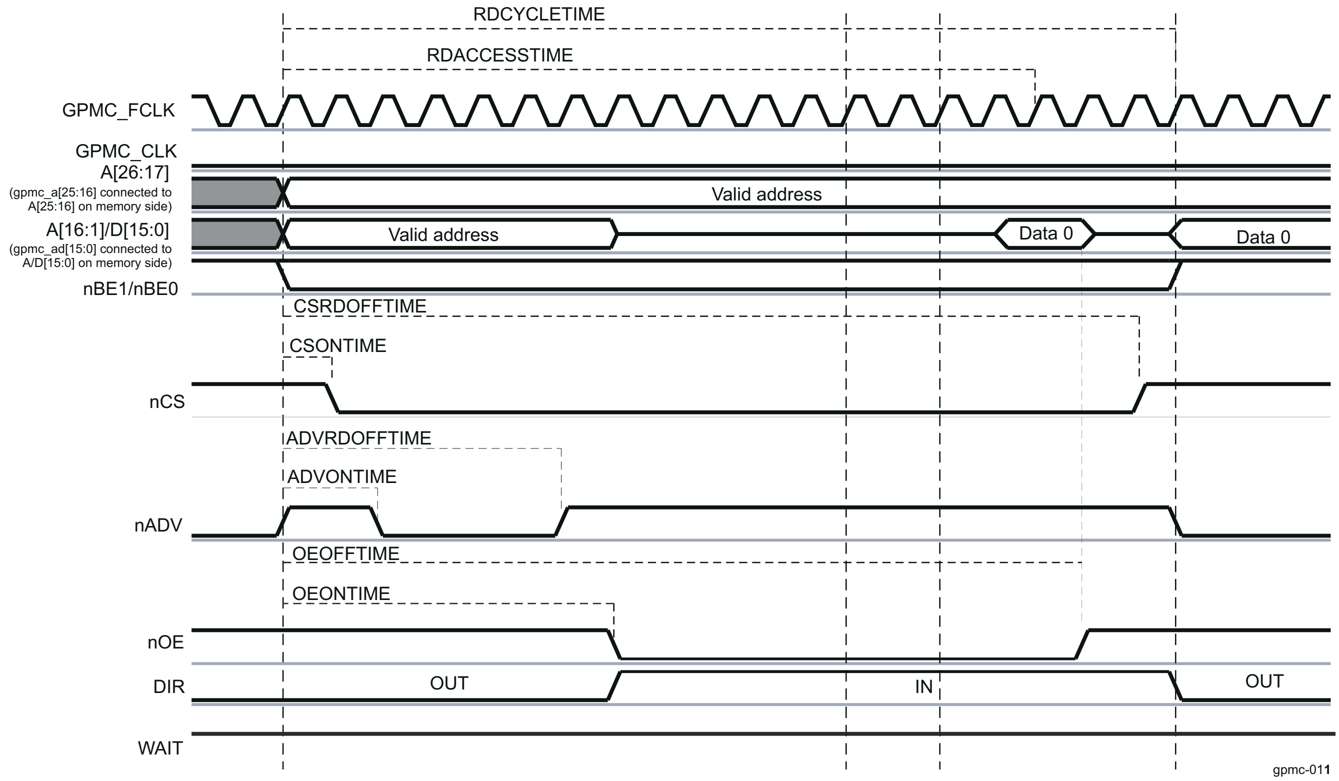


Figure 15-65. Asynchronous Single Read on an Address/Data-Multiplexed Device

For formulas to calculate timing parameters, see Section 15.4.5.6.1, *GPMC Timing Parameters Formulas*.

Table 15-456 lists the timing bit fields to set up to configure the GPMC in asynchronous single-read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For more information, see Section 15.4.4.8.2.3, *Address/Data-Multiplexing Interface*.

Address bits (A[16:1] from a GPMC perspective, A[15:0] from an external device perspective) are placed on the address/data bus, and the remaining address bits gpmc_a[27:16] are placed on the address bus. The address phase ends at nOE assertion, when the DIR signal goes from OUT to IN.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC_CONFIG2_i[3:0] CSONTIME bit field. It controls the address setup time to nCS assertion.
  - nCS deassertion time is controlled by the GPMC_CONFIG2_i[12:8] CSRDOFFTIME bit field. It controls the address hold time from nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output enable signal nOE:
  - nOE assertion indicates a read cycle.
  - nOE assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.
- Read data is latched when RDACCESSTIME completes. Access time is defined in the GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field.

- Direction signal DIR: DIR goes from OUT to IN at the same time that nOE is asserted.
- The end of the access is defined by the GPMC_CONFIG5_i[4:0] RDCYCLETIME parameter.

In the GPMC, when a 16-bit wide device is attached to the controller, a 32-bit word write access is split into two 16-bit word write accesses. For more information about GPMC access size and type adaptation, see Section 15.4.4.10.5, *System Burst Versus External Device Burst Support*.

Between two successive accesses, if an nCS pulse is needed:

- The GPMC_CONFIG6_i[11:8] CYCLE2CYCLEDELAY bit field can be programmed with the GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN bit enabled.
- The CSWROFFTIME and CSONTIME parameters also allow a chip-select pulse, but this affects all other types of access.

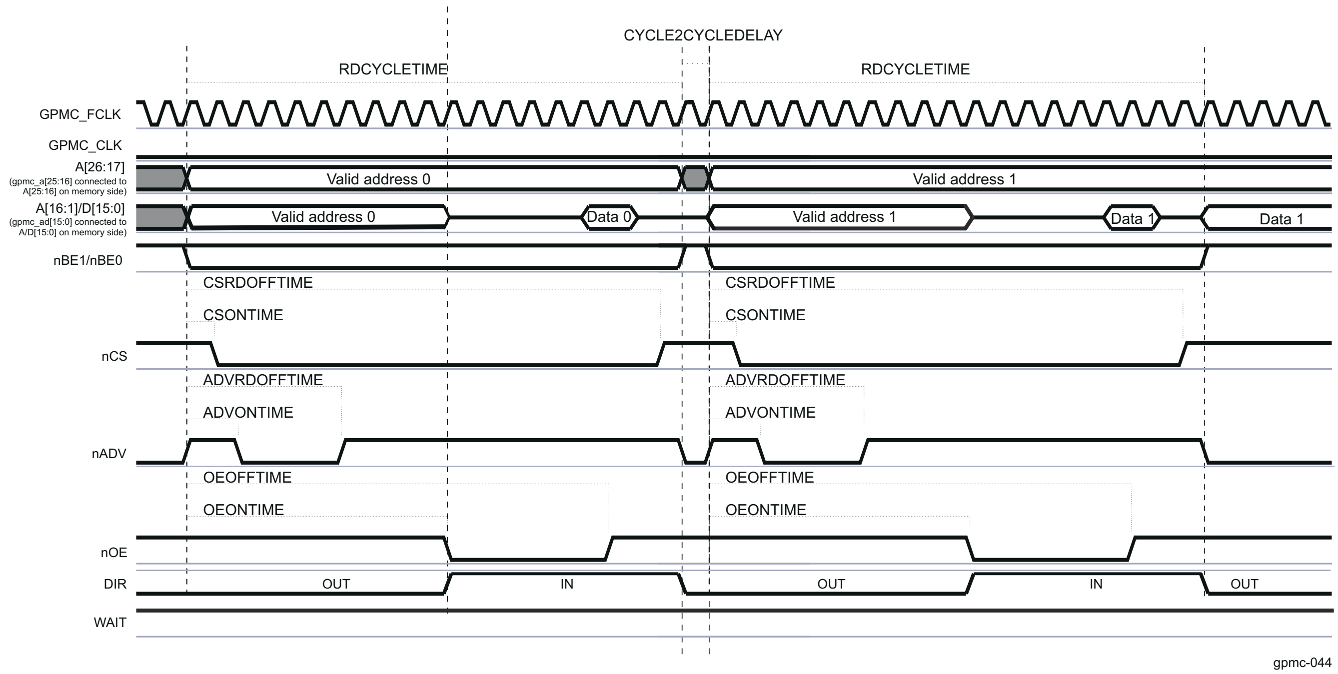


Figure 15-66. Two Asynchronous Single-Read Accesses on an Address/Data-Multiplexed Device (32-Bit Read Split Into 2 x 16-Bit Read)

15.4.4.10.1.1.2 Asynchronous Single-Write Operation on an Address/Data-Multiplexed Device

Figure 15-67 shows an asynchronous single-write operation on an address/data-multiplexed device.

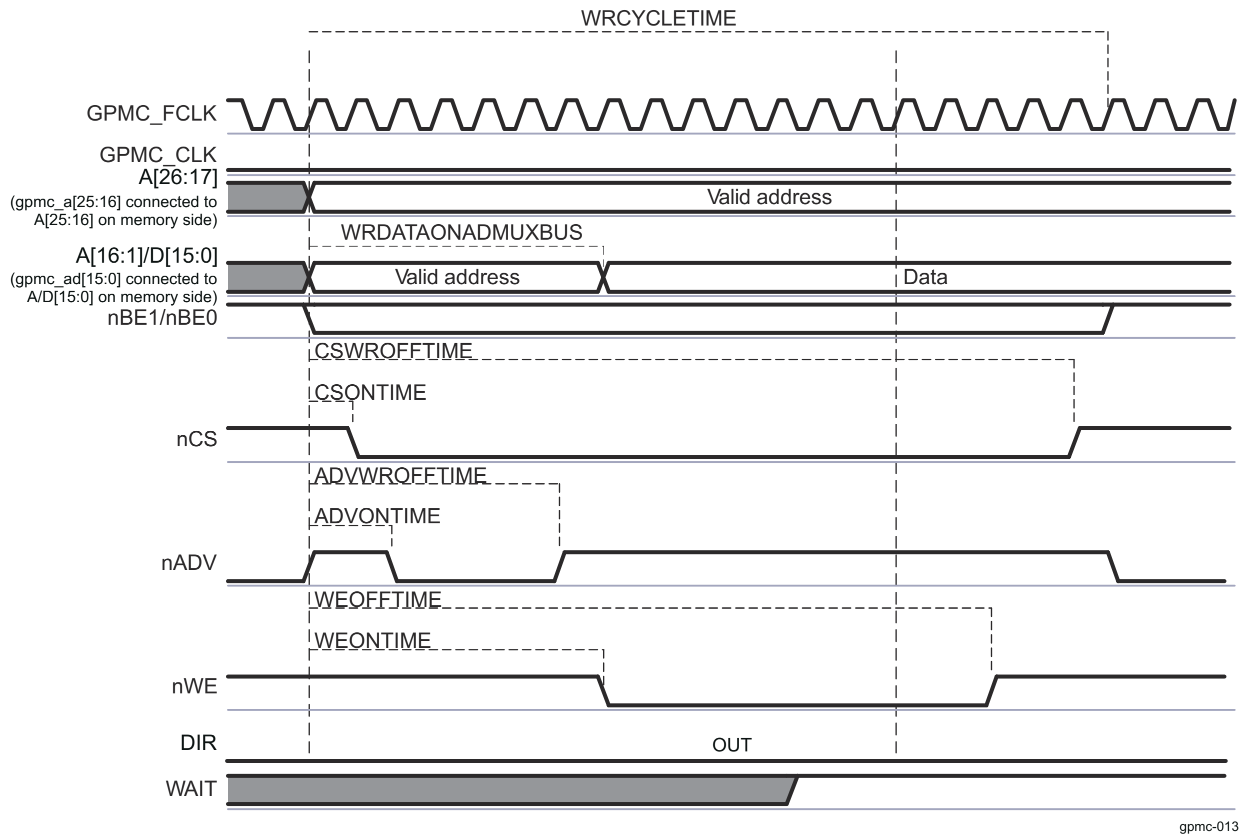


Figure 15-67. Asynchronous Single-Write on an Address/Data-Multiplexed Device

For formulas to calculate timing parameters, see Section 15.4.5.6.1, *GPMC Timing Parameters Formulas*.

Table 15-456 lists the timing bit fields to set up to configure the GPMC in asynchronous single-write mode.

When the GPMC generates a write access to an address/data-multiplexed device, it drives the address bus until nWE assertion time. For more information, see Section 15.4.4.8.2.3, *Address/Data-Multiplexing Interface*.

The nCS and nADV signals are controlled in the same way as for a asynchronous single-read operation on an address/data-multiplexed device.

- Write enable signal nWE:
  - nWE assertion indicates a write cycle.
  - nWE assertion time is controlled by the GPMC_CONFIG4_i[19:16] WEONTIME bit field.
  - nWE deassertion time is controlled by the GPMC_CONFIG4_i[28:24] WEOFFTIME bit field.
- Direction signal DIR: DIR signal is OUT during the entire access.
- The end of the access is defined by the GPMC_CONFIG5_i[12:8] WRCYCLETIME parameter.

Address bits A[16:1] (GPMC point of view) are placed on the address/data bus at the start of cycle time, and the remaining address bits A[26:17] are placed on the address bus.

Data is driven on the address/data bus at a GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS time.

Note

Write multiple access in asynchronous mode is not supported. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

After a write operation, if no other access (read or write) is pending, the data bus keeps its previous value. See Section 15.4.4.9.10, *Bus Keeping Support*.

15.4.4.10.1.1.3 Asynchronous Multiple (Page) Write Operation on an Address/Data-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for address/data-multiplexed devices.

If the GPMC_CONFIG1_i[28] WRITEMULTIPLE bit is enabled (0x1) with the GPMC_CONFIG1_i[27] WRITETYPE bit as asynchronous (0x0), the GPMC processes single asynchronous accesses.

For accesses on nonmultiplexed devices, see Section 15.4.4.10.3, *Asynchronous and Synchronous Accesses in Nonmultiplexed Mode*.

15.4.4.10.1.2 Access on Address/Address/Data-Multiplexed Devices

15.4.4.10.1.2.1 Asynchronous Single Read Operation on an AAD-Multiplexed Device

Figure 15-68 shows an asynchronous single-read operation on an AAD-multiplexed device.

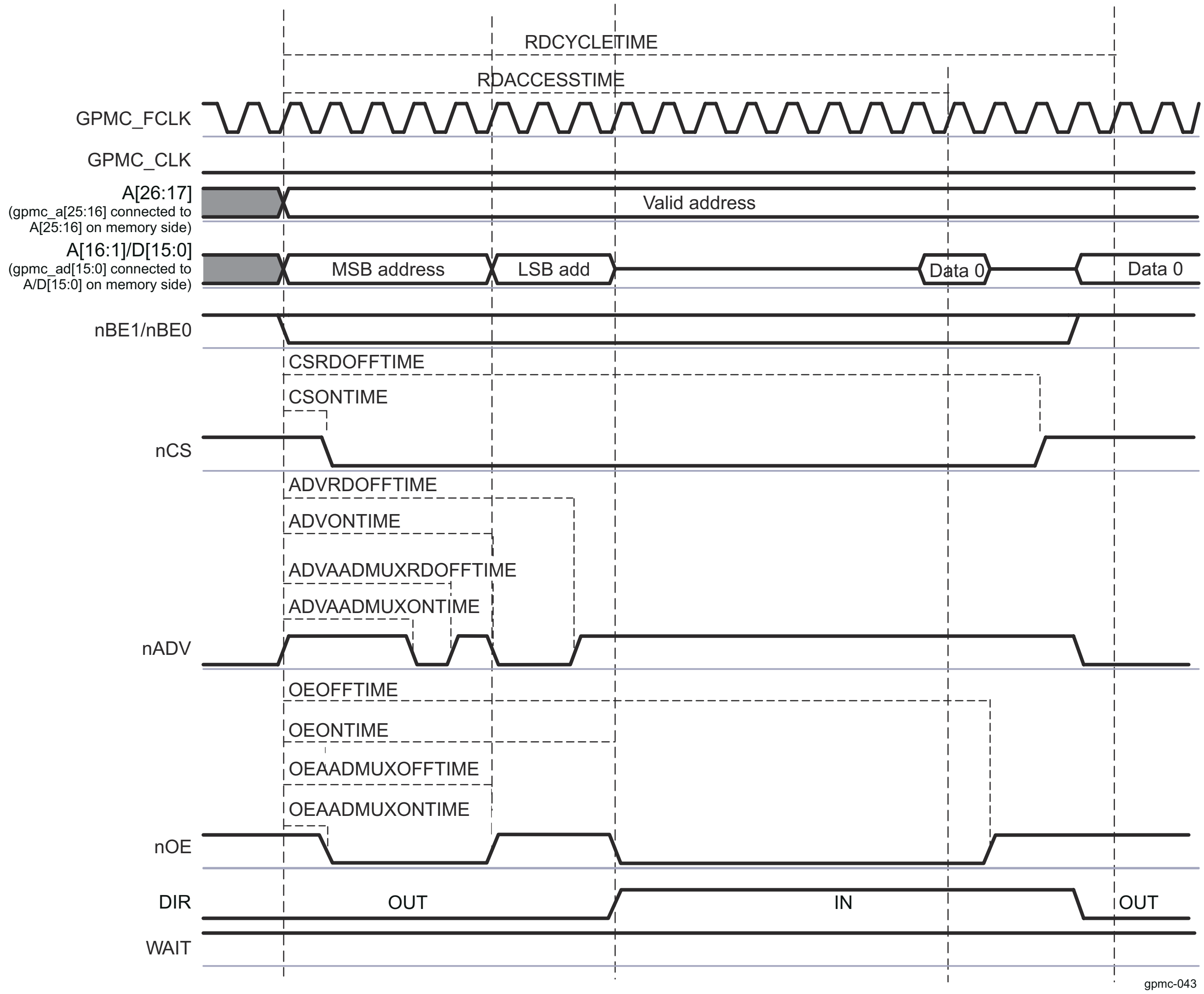


Figure 15-68. Asynchronous Single Read on an AAD-Multiplexed Device

For formulas to calculate timing parameters, see Section 15.4.5.6.1, *GPMC Timing Parameters Formulas*.

Table 15-456 lists the timing bit fields to set up to configure the GPMC in asynchronous single write mode.

When the GPMC generates a read access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The first address phase ends at the first nOE deassertion time. The second phase for LSB address is qualified with nOE driven high. The second address phase ends at the second nOE assertion time, when the DIR signal goes from OUT to IN.

The nCS and DIR signals are controlled in the same way as for an asynchronous single-read operation on an address/data-multiplexed device.

- Address valid signal nADV. nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.

  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output Enable signal nOE. nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE first deassertion time is controlled by the GPMC_CONFIG3_i[15:13] OEAADMUXOFFTIME bit field.
  - nOE second assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE second deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.

15.4.4.10.1.2.2 Asynchronous Single-Write Operation on an AAD-Multiplexed Device

Figure 15-69 shows an asynchronous single-write operation on an AAD-multiplexed device.

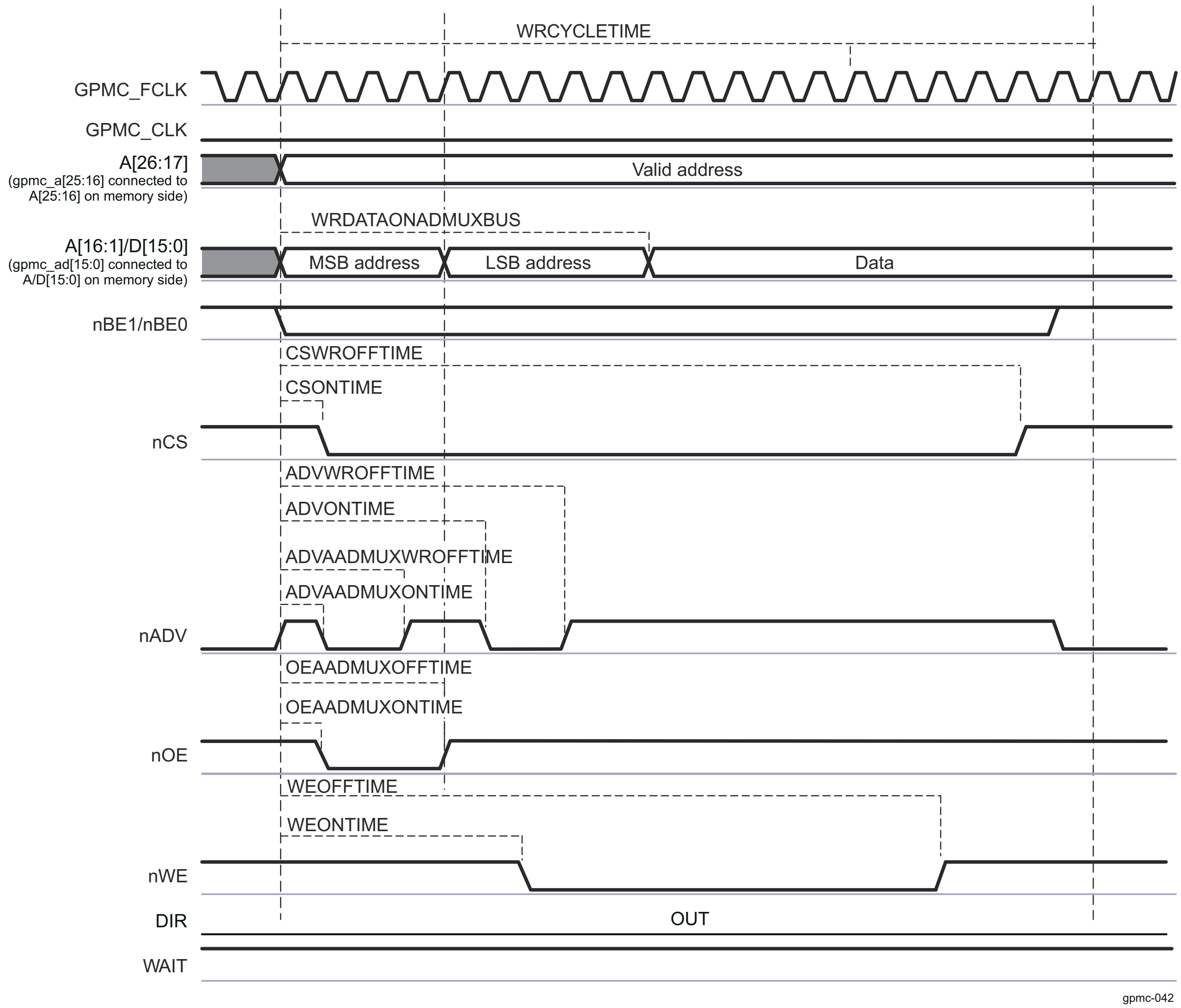


Figure 15-69. Asynchronous Single Write on an AAD-Multiplexed Device

For formulas to calculate timing parameters, see Section 15.4.5.6.1, *GPMC Timing Parameters Formulas*.

Table 15-456 lists the timing bit fields to set up to configure the GPMC in asynchronous single-write mode.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS, nWE, and DIR signals are controlled in the same way as for an asynchronous single-write operation on an address/data-multiplexed device. See Table 15-447.

- Address valid signal nADV is asserted and deasserted twice during a write transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.
  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[30:28] ADVAADMUXWROFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[20:16] ADVWROFFTIME bit field.
- Output enable signal nOE is asserted during the address phase of a write transaction:
  - nOE assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE deassertion time is controlled by the GPMC_CONFIG3_i[15:13] OEAADMUXOFFTIME bit field.

The address bits for the first address phase are driven onto the data bus until nOE deassertion. Data is driven onto the address/data bus at the clock edge defined by the GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS parameter.

15.4.4.10.1.2.3 Asynchronous Multiple (Page) Read Operation on an AAD-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for AAD-multiplexed devices.

If the GPMC_CONFIG1_i[28] WRITEMULTIPLE bit is enabled (0x1) with the GPMC_CONFIG1_i[27] WRITETYPE bit as asynchronous (0x0), the GPMC processes single asynchronous accesses.

For accesses on nonmultiplexed devices, see Section 15.4.4.10.3, *Asynchronous and Synchronous Accessed in Nonmultiplexed Mode*.

15.4.4.10.2 Synchronous Access Description

This section describes read and write synchronous accesses on address/data-multiplexed devices. All information in this section can be applied to any type of memory (nonmultiplexed, address and data-multiplexed, or AAD-multiplexed) with the difference limited to the address phase. For accesses on nonmultiplexed devices, see Section 15.4.4.10.3, *Asynchronous and Synchronous Accessed in Nonmultiplexed Mode*.

In synchronous operations:

- The GPMC_CLK clock is provided outside the GPMC when accessing the memory device.
- The GPMC_CLK clock is derived from the GPMC_FCLK clock using the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field. In the following section i stands for the chip-select number, i = 0 to 7.
- The GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME bit field specifies that the GPMC_CLK is provided outside the GPMC for 0 to 2 GPMC_FCLK cycles after start access time until RDCYCLETIME or WRCYCLETIME completes.

15.4.4.10.2.1 Synchronous Single Read

Figure 15-70 and Figure 15-71 show a synchronous single-read operation with GPMCFCLKDIVIDER equal to 0 and 1, respectively.

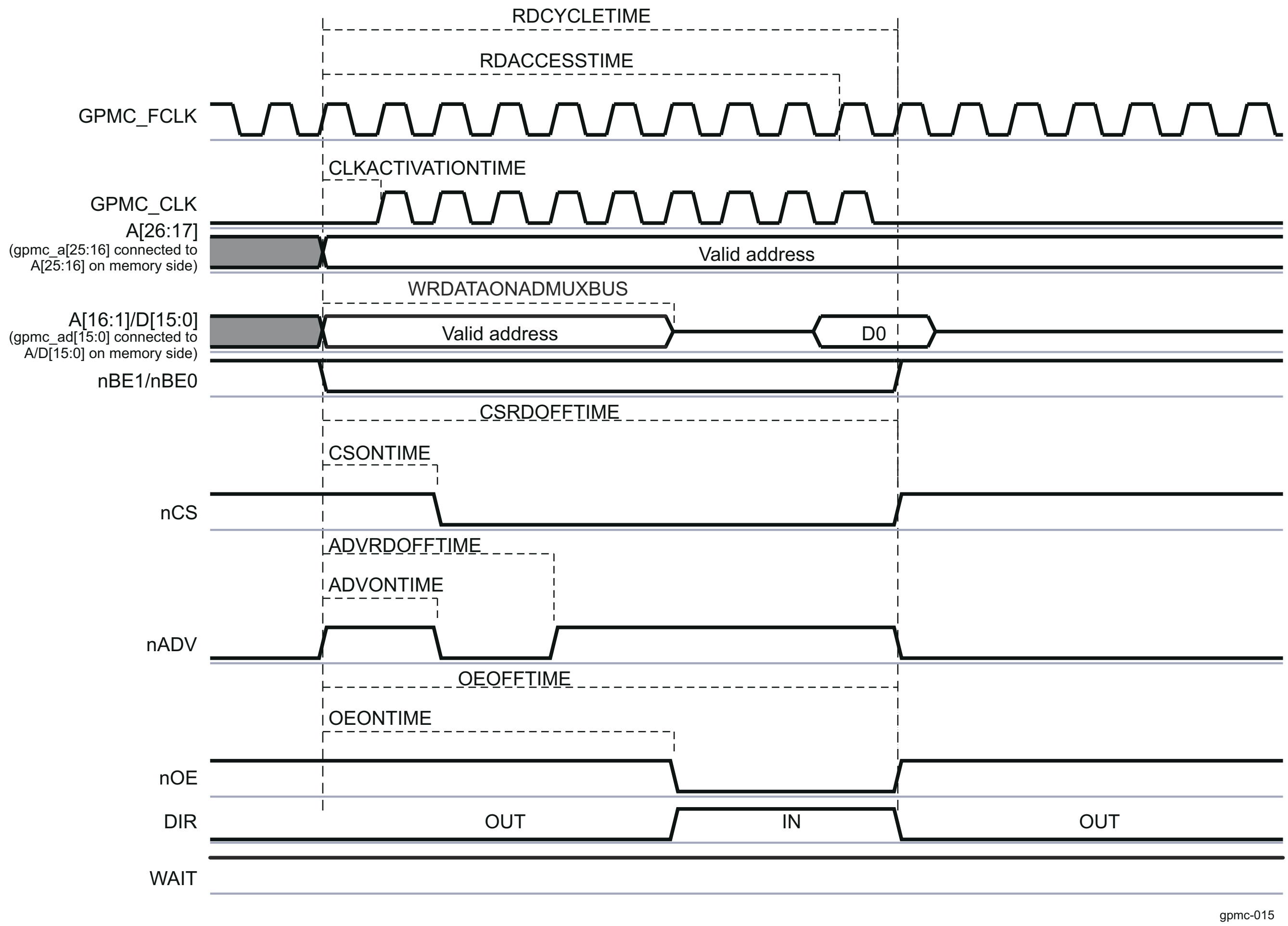


Figure 15-70. Synchronous Single Read (GPMCFCLKDIVIDER = 0)

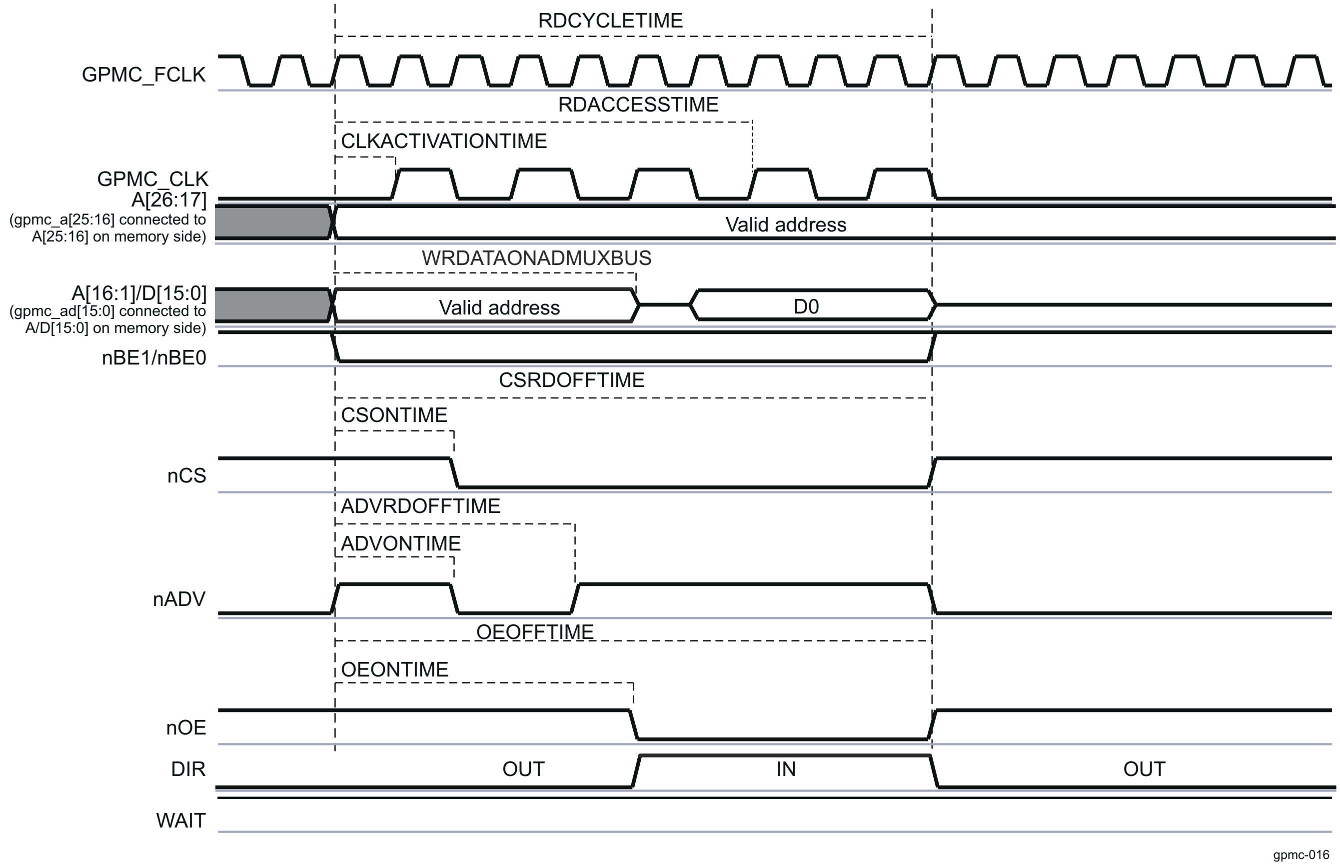


Figure 15-71. Synchronous Single Read (GPMCFCLKDIVIDER = 1)

For formulas to calculate timing parameters, see Section 15.4.5.6.1, *GPMC Timing Parameters Formulas*.

Table 15-456 lists the timing bit fields to set up to configure the GPMC in asynchronous single-read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For more information, see Section 15.4.4.8.2.3, *Address/Data-Multiplexing Interface*.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC_CONFIG2_i[3:0] CSONTIME bit field and ensures address setup time to nCS assertion.
  - nCS deassertion time is controlled by the GPMC_CONFIG2_i[12:8] CSRDOFFTIME bit field and ensures address hold time to nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output enable signal nOE:
  - nOE assertion indicates a read cycle.
  - nOE assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.
- Initial latency for the first read data is controlled by GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field or by monitoring the WAIT signal.
- Total access time (the GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field) corresponds to RDACCESSTIME plus the address hold time from nCS deassertion, plus time from RDACCESSTIME to CSRDOFFTIME.
- Direction signal DIR: DIR goes from OUT to IN at the same time as nOE assertion.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS and DIR signals are controlled in the same way as for a synchronous single-read operation on an address/data-multiplexed device.

- Address valid signal nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.
  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output Enable signal nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE first deassertion time is controlled by the GPMC_CONFIG3_i[15:13] OEAADMUXOFFTIME bit field.
  - nOE second assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE second deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 15.4.4.9.10, *Bus Keeping Support*.

15.4.4.10.2.2 Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability)

Figure 15-72 and Figure 15-73 show a synchronous multiple-read operation with GPMCFCLKDivider equal to 0 and 1, respectively.

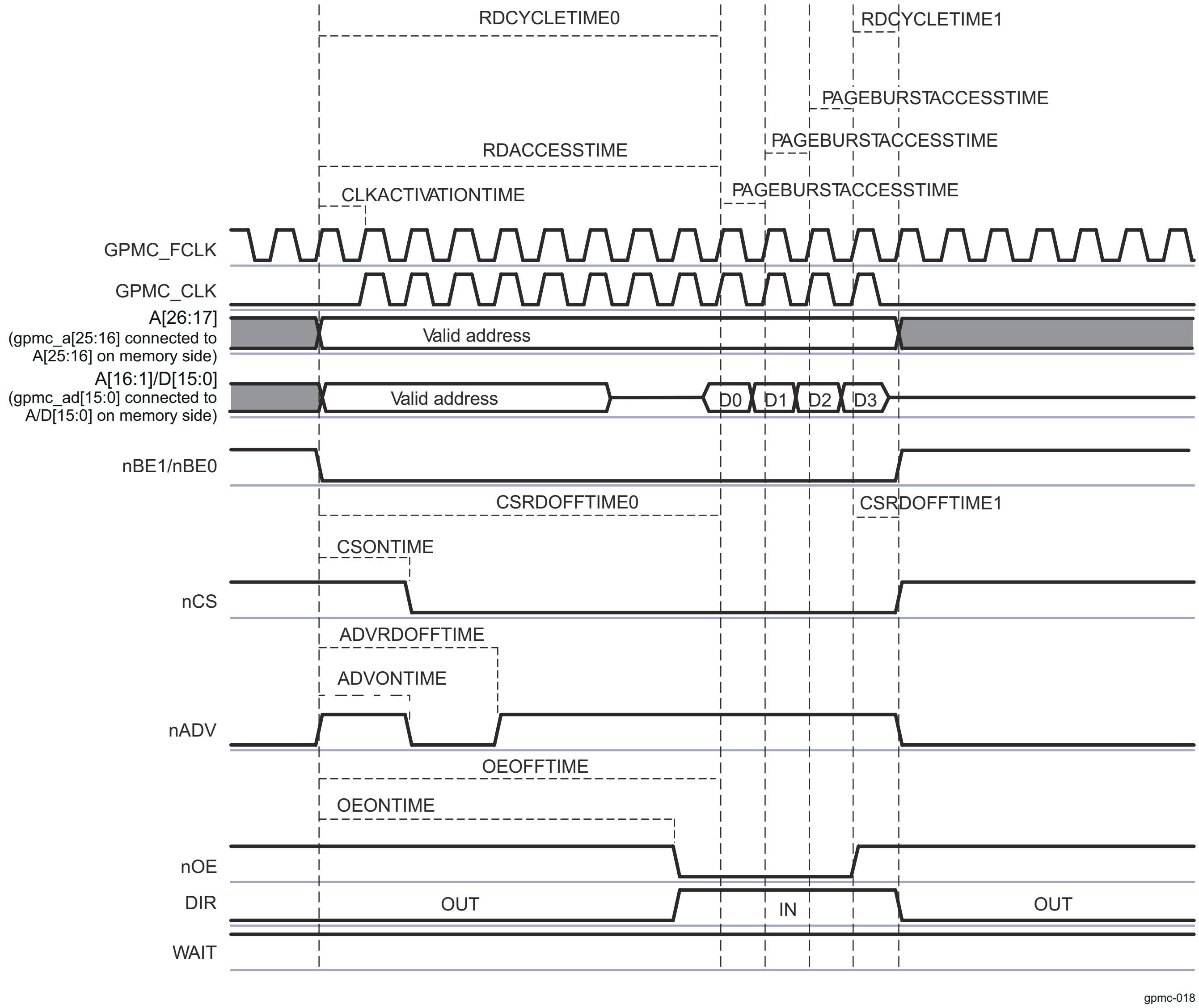


Figure 15-72. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0)

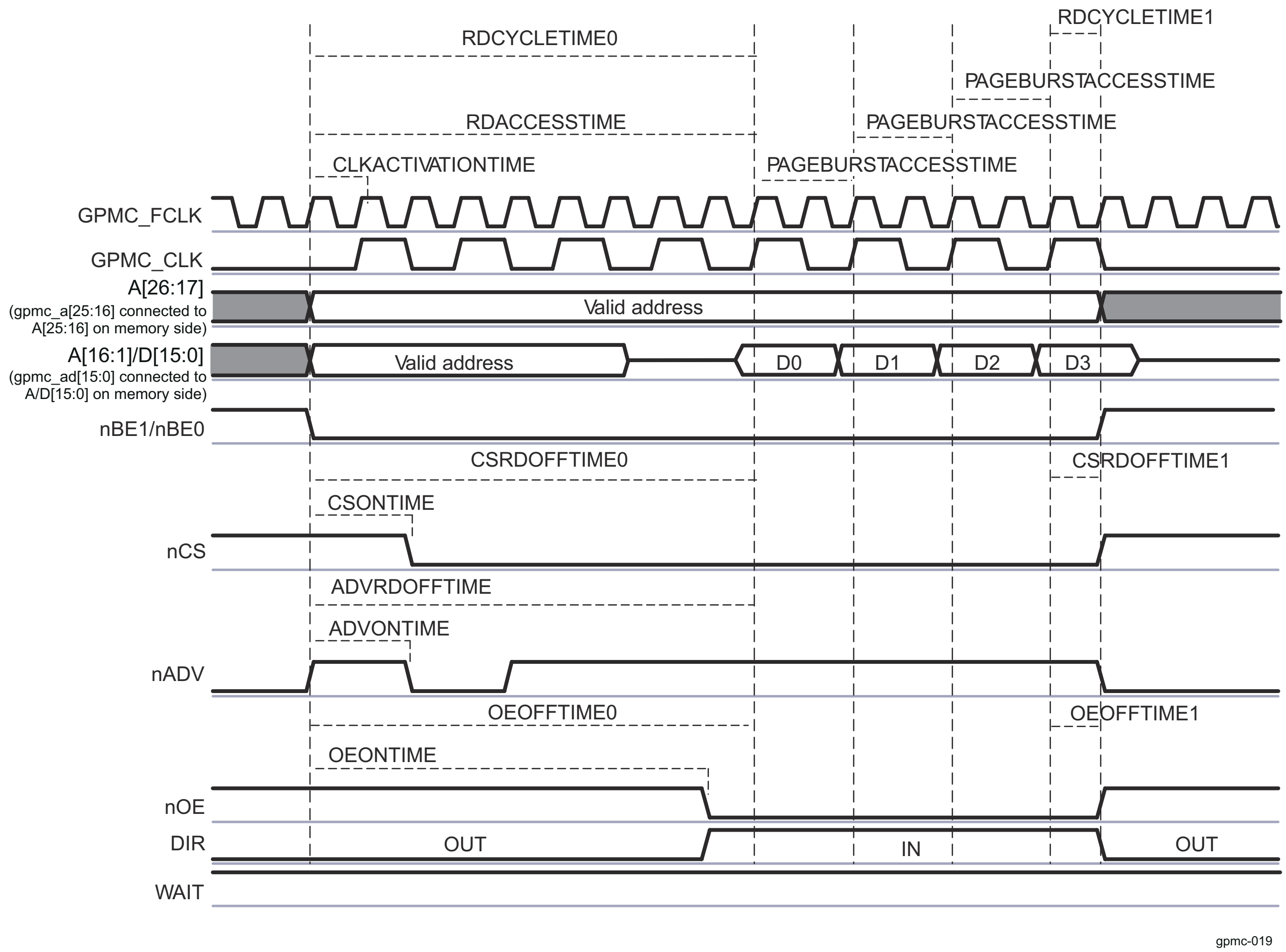


Figure 15-73. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1)

When the GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field completes, control-signal timings are frozen during the multiple data transactions, corresponding to the GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME bit field multiplied by the number of remaining data transactions.

The nCS, nADV, nOE, and DIR signals are controlled in the same way as for a synchronous single-read operation. See Table 15-442.

Initial latency for the first read data is controlled by RDACCESSTIME or by monitoring the WAIT signal. Successive read data are provided by the memory device every one or two GPMC_CLK cycles. The PAGEBURSTACCESSTIME parameter must be set accordingly with the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field and the memory-device internal configuration. Depending on the device page length, the GPMC checks the device page crossing during a new burst request and purposely inserts initial latency (of RDACCESSTIME) when required.

Total access time GPMC_CONFIG5_i[4:0] RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time from nCS deassertion. In Figure 15-73, the programmed value of RDCYCLETIME equals RDCYCLETIME0 + RDCYCLETIME1.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 15.4.4.9.10, *Bus Keeping Support*.

Burst wraparound is enabled through the GPMC_CONFIG1_i[31] WRAPBURST bit and allows a 4-, 8-, or 16-Word16 linear burst access to wrap within its burst-length boundary through the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field.

15.4.4.10.2.3 Synchronous Single Write

Burst write mode is used for synchronous single or burst accesses.

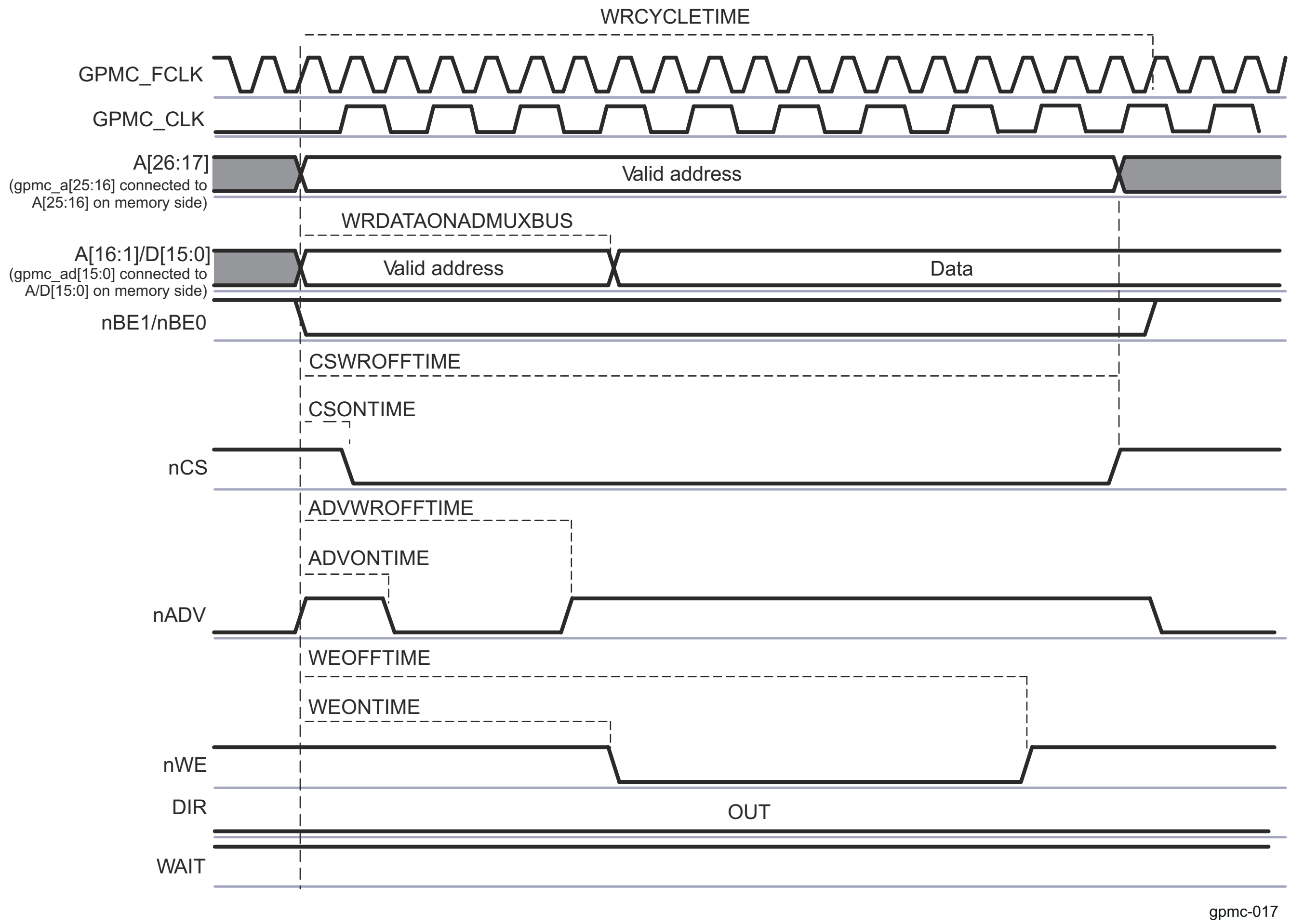


Figure 15-74. Synchronous Single Write on an Address/Data-Multiplexed Device

When the GPMC generates a write access to an address/data-multiplexed device, it drives the data bus (with address bits A[16:1]) until the GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS bit field time. The first data of the burst is driven on the address/data bus at WRDATAONADMUXBUS time.

15.4.4.10.2.4 Synchronous Multiple (Burst) Write

Synchronous burst write mode provides synchronous single or consecutive accesses.

Figure 15-75 shows a synchronous burst write access when the chip-select is configured in address/data-multiplexed mode.

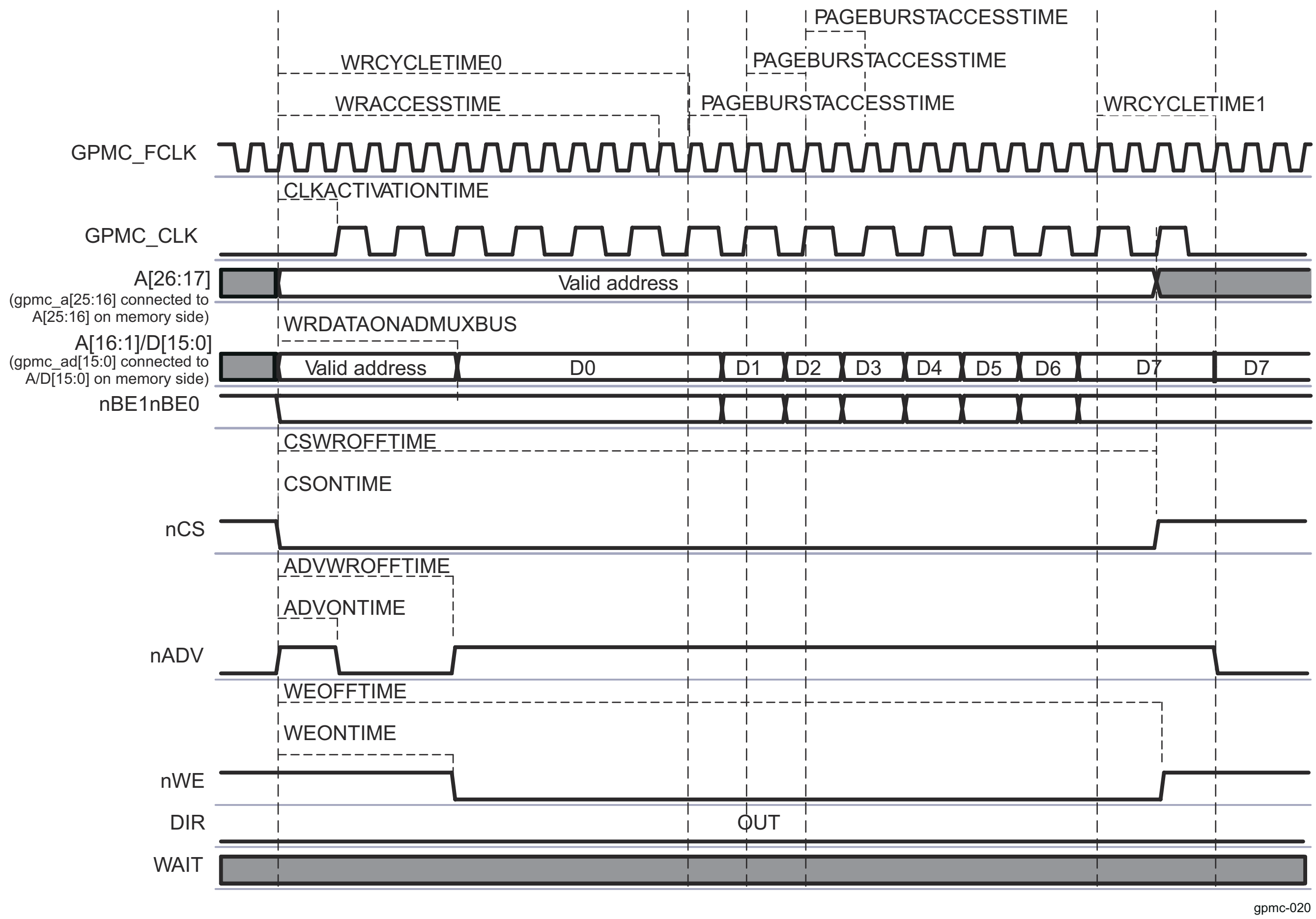


Figure 15-75. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode

Figure 15-76 shows the same synchronous burst write access when the chip-select is configured in address/ address/data-multiplexed (AAD-multiplexed) mode.

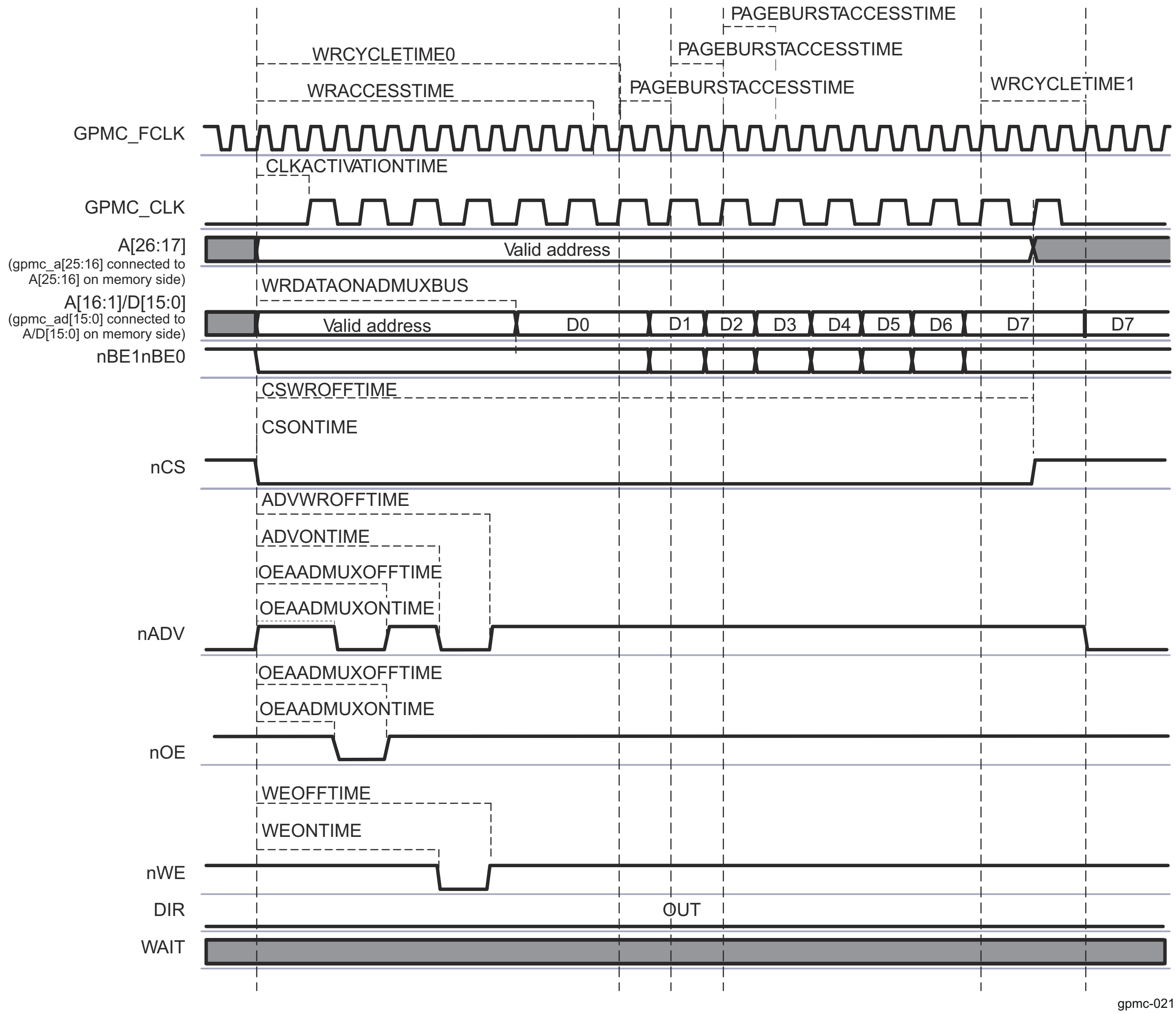


Figure 15-76. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode

The first data of the burst is driven on the A/D bus at the GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS bit field.

When WRACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to the GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME bit field multiplied by the number of remaining data transactions.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For more information, see Section 15.4.4.8.2.3, *Address/Data-Multiplexing Interface*.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC_CONFIG2_i[3:0] CSONTIME bit field (where i = 0 to 7) and ensures address setup time to nCS assertion.
  - nCS deassertion time controlled by the GPMC_CONFIG2_i[20:16] CSWROFFTIME bit field and ensures address hold time to nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.

  - nADV deassertion time is controlled by the GPMC_CONFIG3_i[20:16] ADVWROFFTIME bit field.
- Write enable signal nWE:
  - nWE assertion indicates a read cycle.
  - nWE assertion time is controlled by the GPMC_CONFIG4_i[19:16] WEONTIME bit field.
  - nWE deassertion time is controlled by the GPMC_CONFIG4_i[28:24] WEOFFTIME bit field.

Note

The nWE falling edge must not be used to control the time when the burst first data is driven in the address/data bus, because some new devices require the nWE signal to be low during the address phase.

- Direction signal DIR is OUT during the entire access.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS, and DIR signals are controlled as previously described..

- Address valid signal nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.
  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output Enable signal nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE first deassertion time is controlled by the GPMC_CONFIG4_i[15:13] OEAADMUXOFFTIME bit field.
  - nOE second assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE second deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.

First write data is driven by the GPMC at GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS, when in address/data-multiplexed configuration. The next write data of the burst is driven on the bus at WRACCESSTIME + 1 during GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME GPMC_FCLK cycles. The last data of the synchronous burst write is driven until GPMC_CONFIG5_i[12:8] WRCYCLETIME completes.

- WRACCESSTIME is defined in the GPMC_CONFIG6_i[28:24] bit field.
- The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMCFCLKDIVIDER and the memory-device internal configuration.

Total access time GPMC_CONFIG5_i[12:8] WRCYCLETIME corresponds to WRACCESSTIME plus the address hold time from nCS deassertion. In Figure 15-75, the programmed value of WRCYCLETIME equals WRCYCLETIME0 + WRCYCLETIME1. WRCYCLETIME0 and WRCYCLETIME1 delays are not actual parameters and are only a graphical representation of the full WRCYCLETIME value.

After a write operation, if no other access (read or write) is pending, the data bus keeps the previous value. See Section 15.4.4.9.10, *Bus Keeping Support*.

15.4.4.10.3 Asynchronous and Synchronous Accesses in Nonmultiplexed Mode

Page mode is available only in nonmultiplexed mode.

- Asynchronous single-read operation on a nonmultiplexed device
- Asynchronous single-write operation on a nonmultiplexed device
- Asynchronous multiple- (page mode) read operation on a nonmultiplexed device
- Synchronous operations on a nonmultiplexed device

15.4.4.10.3.1 Asynchronous Single-Read Operation on Nonmultiplexed Device

Figure 15-77 shows an asynchronous single-read operation on a nonmultiplexed device.

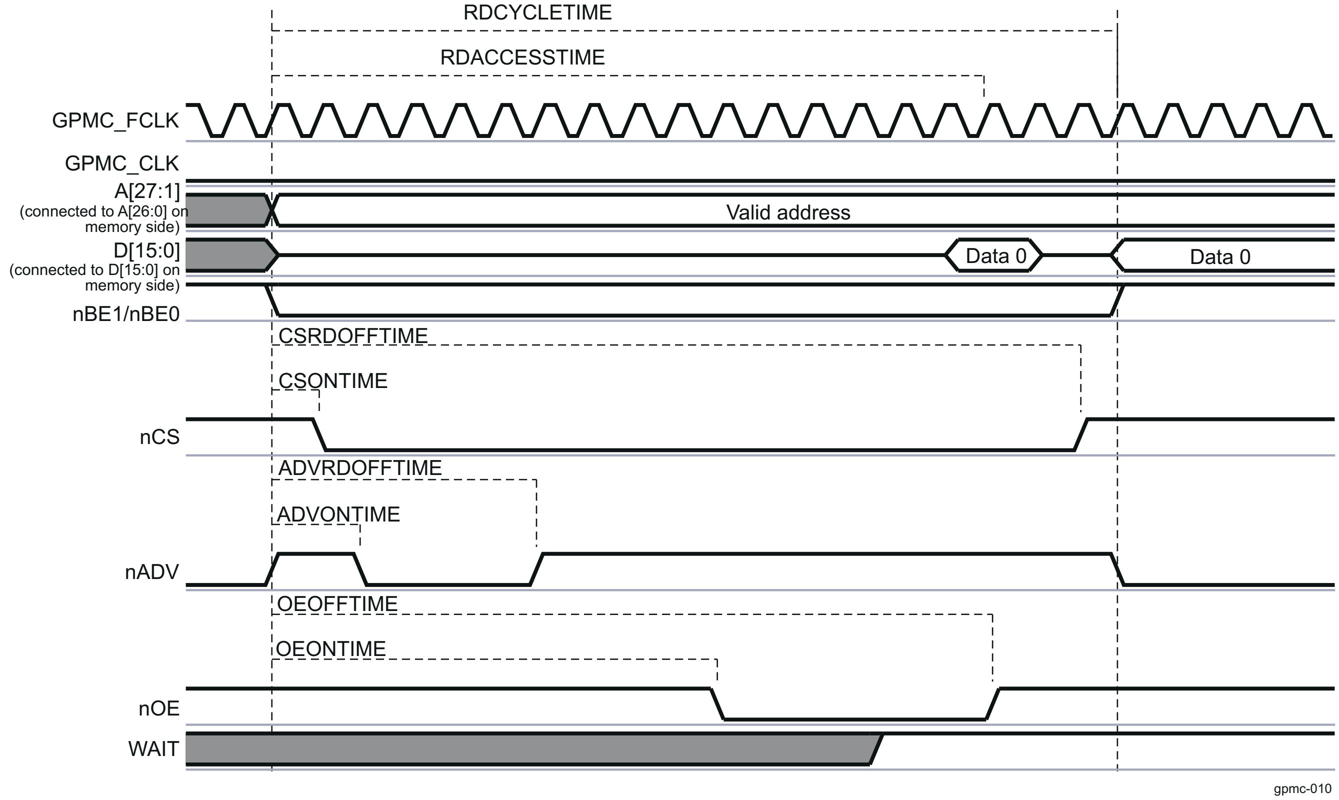


Figure 15-77. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device

The 27-bit address (For a 16-bit data memory device, hence GPMC A[0] is not necessary to be output) is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

Read data is latched at GPMC_CONFIG1_5[20:16] RDACCESSTIME completion time. The end of the access is defined by the GPMC_CONFIG1_5[4:0] RDCYCLETIME parameter.

The nCS, nADV, nOE, and DIR signals are controlled in the same way as address/data-multiplexed accesses (see Table 15-447).

15.4.4.10.3.2 Asynchronous Single-Write Operation on Nonmultiplexed Device

Figure 15-78 shows an asynchronous single-write operation on a nonmultiplexed device.

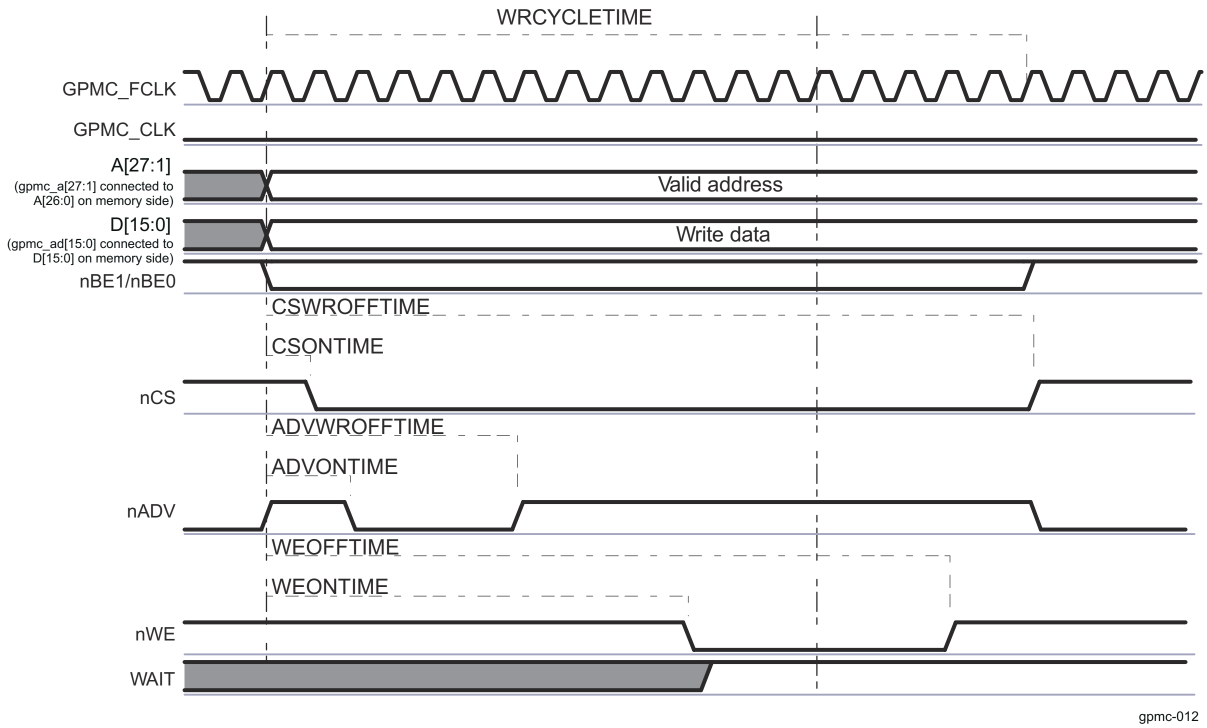


Figure 15-78. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device

The nCS, nADV, nWE, and DIR signals are controlled in the same way as address/data-multiplexed accesses (see Table 15-447).

15.4.4.10.3.3 Asynchronous Multiple (Page Mode) Read Operation on Nonmultiplexed Device

Figure 15-79 shows an asynchronous multiple-read operation on a nonmultiplexed device in which two word32 host read accesses to the GPMC are split into one multiple- (page mode of 4 word16) read access to the attached device.

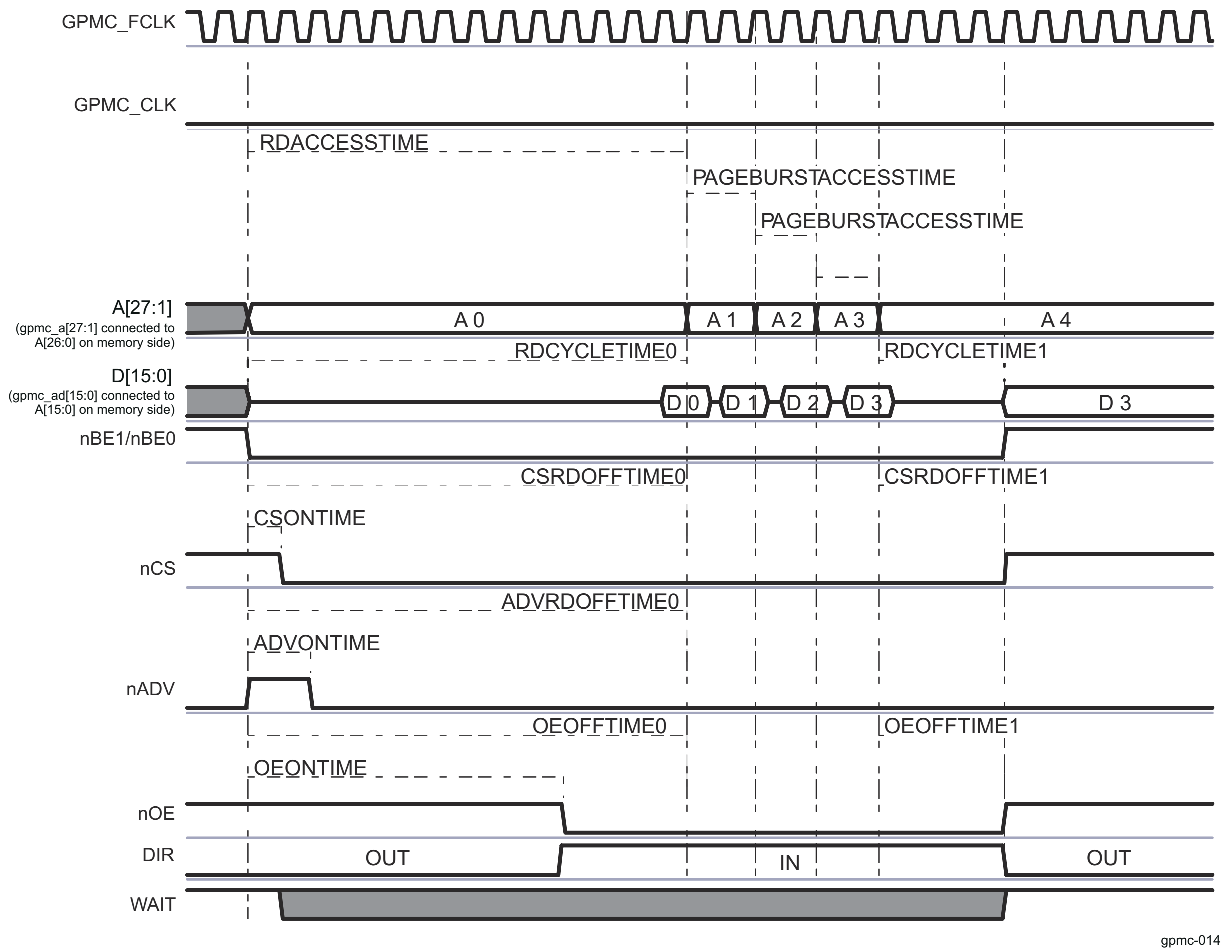


Figure 15-79. Asynchronous Multiple (Page Mode) Read

Note

The WAIT signal is active low.

The nCS, nADV, nOE, and DIR signals are controlled in the same way as address/data-multiplexed accesses (see Table 15-447).

When RDACCESSTIME completes, control signal timings are frozen during the multiple data transactions, corresponding to PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

Read data is latched at GPMC_CONFIG5_i[20:16] RDACCESSTIME completion time (where i = 0 to 7). The end of the access is defined by the GPMC_CONFIG5_i[4:0] RDCYCLETIME parameter.

During consecutive accesses, the GPMC increments the address after each data read completes.

Delay between successive read data in the page is controlled by the GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME parameter. Depending on the device page length, the GPMC can control device page crossing during a burst request and insert initial RDACCESSTIME latency. Page crossing is possible only with a new burst access, meaning a new initial access phase is initiated.

Total access time RDCYCLETIME corresponds to RDACCESSTIME, plus the address hold time, starting from the nCS deassertion.

- The read cycle time is defined in the GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field.

- In Figure 15-79, the programmed value of RDCYCLETIME equals RDCYCLETIME0 (before paged accesses) + RDCYCLETIME1 (after paged accesses).

15.4.4.10.3.4 Synchronous Operations on a Nonmultiplexed Device

All information for this section is equivalent to similar operations for address/data-multiplexed or AAD-multiplexed accesses. The only difference resides in the address phase. See Section 15.4.5.3, *GPMC Configuration in NOR Mode*.

15.4.4.10.4 Page and Burst Support

Each chip-select can be configured to process system single or burst requests into successive single accesses or asynchronous page/synchronous burst accesses, with appropriate access size adaptation.

Depending on the external device page or burst capability, read and write accesses can be independently configured through the GPMC. The GPMC_CONFIG1_i[30] READMULTIPLE and GPMC_CONFIG1_i[28] WRITMULTIPLE bits (where i = 0 to 7) are associated with the READTYPE and WRITETYPE parameters.

Note

- Asynchronous write page mode is not supported.
- 8-bit-wide device support is limited to nonburstable devices (READMULTIPLE and WRITEMULTIPLE are ignored).
- Not applicable to NAND device interfacing

15.4.4.10.5 System Burst vs External Device Burst Support

The device system can issue the following requests to the GPMC:

- Byte, 16-bit word, 32-bit word requests (byte-enable-controlled). This is always a single request from the interconnect point of view.
- Incrementing fixed-length bursts of two, four, and eight words
- Wrapped (critical word access first) fixed-length burst of two, four, or eight words

To process a system request with the optimal protocol, the READMULTIPLE (and READTYPE) and WRITEMULTIPLE (and WRITETYPE) parameters must be set according to the burstable capability (synchronous or asynchronous) of the attached device.

The GPMC access engine issues only fixed-length bursts. The maximum length that can be issued is defined per chip-select by the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field (where i = 0 to 7). When the value of ATTACHEDDEVICEPAGELENGTH is less than the length of the system burst request (including the appropriate access size adaptation according to the device width), the GPMC splits the system burst request into multiple bursts. Within the specified 4-, 8-, or 16-word value, the value of the ATTACHEDDEVICEPAGELENGTH bit field must correspond to the maximum length burst supported by the memory device configured in fixed-length burst mode (as opposed to continuous burst mode).

To get optimal performance from memory devices that natively support 16 Word16-length-wrapping burst capability (critical word access first), the ATTACHEDDEVICEPAGELENGTH parameter must be set to 16 words and the GPMC_CONFIG1_i[31] WRAPBURST bit (where i = 0 to 7) must be set to 1. Similarly DEVICEPAGELENGTH is set to 4 and 8 for memories supporting 4 and 8 Word16-length-wrapping burst, respectively.

When the memory device does not offer (or is not configured to offer) native 16 Word16-length-wrapping burst, the WRAPBURST parameter must be cleared, and the GPMC access engine emulates the wrapping burst by issuing the appropriate burst sequences according to the value of ATTACHEDDEVICEPAGELENGTH.

When the memory device does not support native-wrapping burst, there is usually no difference in behavior between a fixed-burst length mode and a continuous-burst mode configuration (except for a potential power increase from a memory-speculative data prefetch in a continuous burst read). However, even though continuous burst mode is compatible with GPMC behavior, because the GPMC access engine issues only fixed-length burst and does not benefit from continuous burst mode, it is best to configure the memory device in fixed-length burst mode.

The memory device maximum-length burst (configured in fixed-length burst wrap or nonwrap mode) usually corresponds to the memory device data buffer size. Memory devices with a minimum of 16 half-word buffers are the most appropriate (especially with wrap support), but memory devices with smaller buffer size (4 or 8) are also supported, assuming that the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field is set accordingly to 4 or 8 words.

The device system issues only requests with addresses or starting addresses for nonwrapping burst requests; that is, the request size boundary is aligned. In case of an eight-word-wrapping burst, the wrapping address always occurs on the eight-word boundary. As a consequence, all words requested must be available from the memory data buffer when the buffer size is equal to or greater than the value of ATTACHEDDEVICEPAGELENGTH. This usually means that data can be read from or written to the buffer at a constant rate (number of cycles between data) without wait-states between data accesses. If the memory does not behave this way (nonzero wait-state burstable memory), wait pin monitoring must be enabled to dynamically control data access completion within the burst.

Note

When the system burst request length is less than the value of ATTACHEDDEVICEPAGELENGTH, the GPMC proceeds with the required accesses.

15.4.4.11 pSRAM Access Specificities

pSRAM devices are SRAM-pin-compatible low-power memories that contain a self-refreshed DRAM memory array. The GPMC_CONFIG1_i[11:10] DEVICETYPE bit field (where i = 0 to 7) must be set to 0b00.

The pSRAM device uses the NOR protocol. It supports the following operations:

- Asynchronous single read
- Asynchronous page read
- Asynchronous single write
- Synchronous single read and write
- Synchronous burst read
- Synchronous burst write (not supported by NOR flash memory)

pSRAM devices must be powered up and initialized in a predefined manner according to the specifications of the attached device.

pSRAM devices can be programmed to use either mode: fixed or variable latency. pSRAM devices can automatically schedule autorefresh operations, which force the GPMC to use its WAIT signal capability when read or write operations occur during an internal self-refresh operation, or they can automatically include the autorefresh operation in the access time. These devices do not require additional WAIT signal capability or a minimum nCS high pulse width between consecutive accesses to ensure that the correct internal refresh operation is scheduled.

15.4.4.12 NAND Access Description

NAND (8-bit and 16-bit) memory devices using a standard NAND asynchronous address/data-multiplexing scheme can be supported on any chip-select with the appropriate asynchronous configuration settings.

As for any other type of memory compatible with the GPMC interface, accesses to a chip-select allocated to a NAND device can be interleaved with accesses to chip-selects allocated to other external devices. This interleaved capability limits the system to *chip enable don't care* NAND devices, because the chip-select allocated to the NAND device must be deasserted if accesses to other chip-selects are requested.

15.4.4.12.1 NAND Memory Device in Byte or 16-bit Word Stream Mode

NAND devices require correct command and address programming before data array read or write accesses. The GPMC does not include specific hardware to translate a random address system request into a NAND-specific multiphase access. In that sense, GPMC NAND support, as opposed to random memory-map device support, is data stream-oriented (byte or 16-bit word).

The GPMC NAND programming model relies on a software driver for address and command formatting with the correct data address pointer value according to the block and page structure. Because of NAND structure and protocol interface diversity, the GPMC does not support automatic command and address phase programming, and software drivers must access the NAND device ID to ensure that correct command and address formatting are used for the identified device.

NAND device data read and write accesses are achieved through an asynchronous read or write access. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Any chip-select region can be qualified as a NAND region to constrain the nADV/ALE signal as ALE (ALE active high, default state value at low) during address program access, and the nBE0/CLE signal as CLE (CLE active high, default state value at low) during command program access. GPMC address lines are not used (the previous value is not changed) during NAND access.

15.4.4.12.1.1 Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode

The GPMC_CONFIG7_i register (where i = 0 to 7) associated with a NAND device region interfaced in byte or word stream mode can be initialized with a minimum size of 16MiB, because any address location in the chip-select memory region can be used to access a NAND data array. The NAND flash protocol specifies an address sequence where address bits are passed through the data bus in a series of write accesses with the ALE pin asserted. After this address phase, all operations are streamed and the system requests address is irrelevant.

CAUTION

To allow correct command, address, and data-access controls, the GPMC_CONFIG1_i register associated with a NAND device region must be initialized in asynchronous read and write modes with the parameters listed in Table 15-425. Failure to comply with these settings corrupts the NAND interface protocol.

Table 15-425. Chip-Select Configuration for NAND Interfacing

Bit Field	Register	Value	Comments
WRAPBURST	GPMC_CONFIG1_i[31] (1)	0	No wrap
READMULTIPLE	GPMC_CONFIG1_i[30]	0	Single access
READTYPE	GPMC_CONFIG1_i[29]	0	Asynchronous mode
WRITEMULTIPLE	GPMC_CONFIG1_i[28]	0	Single access
WRITETYPE	GPMC_CONFIG1_i[27]	0	Asynchronous mode
CLKACTIVATIONTIME	GPMC_CONFIG1_i[26:25]	0b00	
ATTACHEDDEVICEPAGELENGTH	GPMC_CONFIG1_i[24:23]	Don't care	Single-access mode
WAITREADMONITORING	GPMC_CONFIG1_i[22]	0	Wait not monitored by GPMC access engine
WAITWRITEMONITORING	GPMC_CONFIG1_i[21]	0	Wait not monitored by GPMC access engine
WAITMONITORINGTIME	GPMC_CONFIG1_i[19:18]	Don't care	Wait not monitored by GPMC access engine
WAITPINSELECT	GPMC_CONFIG1_i[17:16]		Select which wait is monitored by edge detectors
DEVICESIZE	GPMC_CONFIG1_i[13:12]	0b00 or 0b01	8- or 16-bit interface

Table 15-425. Chip-Select Configuration for NAND Interfacing (continued)

Bit Field	Register	Value	Comments
DEVICETYPE	GPMC_CONFIG1_i[11:10]	0b10	NAND device in stream mode
MUXADDDATA	GPMC_CONFIG1_i[9:8]	0b00	Nonmultiplexed mode
TIMEPARAGRANULARITY	GPMC_CONFIG1_i[4]	0	Timing achieved with best GPMC clock granularity
GPMCFCLKDIVIDER	GPMC_CONFIG1_i[1:0]	Don't care	Asynchronous mode

(1) i = 0 to 7

The GPMC_CONFIG1_i to GPMC_CONFIG4_i registers (where i = 0 to 7) associated with a NAND device region must be initialized with the correct control-signal timing value according to the NAND device timing parameters.

15.4.4.12.1.2 NAND Device Command and Address Phase Control

NAND devices require multiple address programming phases. The MPU software driver must issue the correct number of command and address program accesses, according to the device command set and the device address-mapping scheme.

NAND device-command and address-phase programming is achieved through write requests to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i register locations (where i = 0 to 7) with the correct command and address values. These locations are mapped in the associated chip-select register region. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Command and address values are not latched during the access and cannot be read back at the register location.

- Only write accesses must be issued to these locations, but the GPMC does not discard any read access. Accessing a NAND device with nOE and CLE or ALE asserted (read access) can produce undefined results.
- Write accesses to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i register locations must be posted for faster operations (where i = 0 to 7). The GPMC_CONFIG[0] NANDFORCEPOSTEDWRITE bit enables write accesses to these locations as posted, even if they are defined as nonposted.

A write buffer is used to store write transaction information before the external device is accessed:

- Up to eight consecutive posted write accesses can be accepted and stored in the write buffer.
- For nonposted write, the pipeline is one deep.
- An GPMC_STATUS[0] EMPTYWRITEBUFFERSTATUS bit stores the empty status of the write buffer.

The GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i registers (where i = 0 to 7) are 32-bit word locations, which means any 32-or 16-bit word access is split into 4- or 2-byte accesses if an 8-bit-wide NAND device is attached. For multiple-command phase or multiple-address phase, the software driver can use 32- or 16-bit word access to these registers, but it must consider the splitting and little-endian ordering scheme. When only one byte command or address phase is required, only byte write access to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i registers can be used, and any of the four byte locations of the registers is valid.

The same applies to a GPMC_NAND_COMMAND_i and a GPMC_NAND_ADDRESS_i (where i = 0 to 7) 32-bit word write access to a 16-bit-wide NAND device (split into two 16-bit word accesses). In the case of a 16-bit word write access, the MSByte of the 16-bit word value must be set according to the NAND device requirement (usually 0). Either 16-bit word location or any one of the four byte locations of the registers is valid.

15.4.4.12.1.3 Command Latch Cycle

Writing data at the GPMC_NAND_COMMAND_i location (where i = 0 to 7) places the data as the NAND command value on the bus, using a regular asynchronous write access.

- nCE is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- CLE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.

- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE and nRE (nOE) are maintained inactive.

Figure 15-80 shows the NAND command latch cycle.

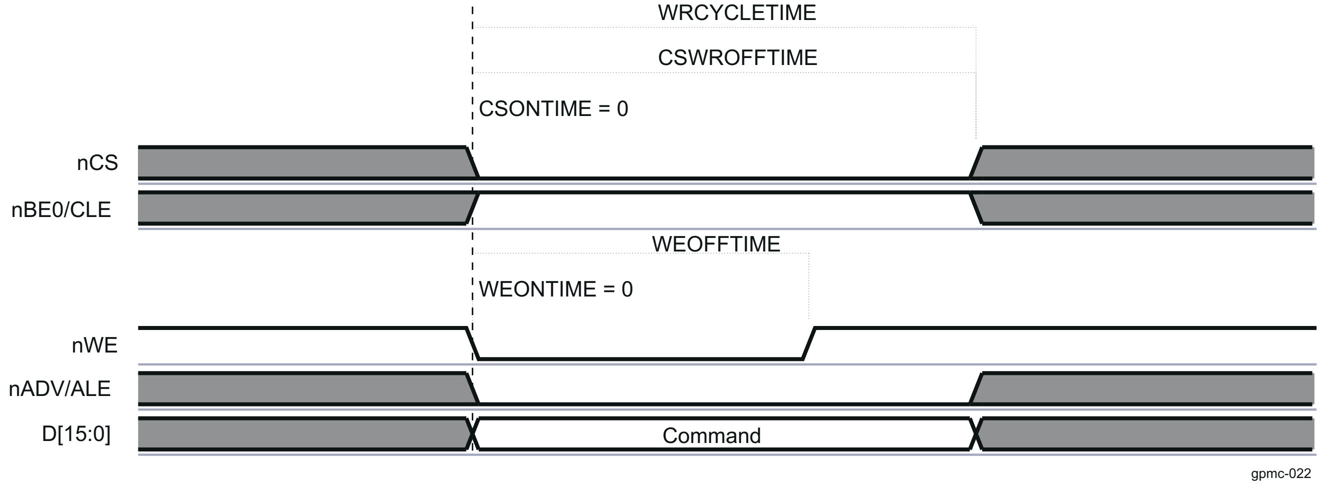


Figure 15-80. NAND Command Latch Cycle

Note

CLE is shared with the nBE0 output signal and has an inverted polarity from BE0. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, nBE0 (also nBE1) must not toggle, because it is shared with CLE.

NAND flash memories do not use byte-enable signals.

15.4.4.12.1.4 Address Latch Cycle

Writing data at the GPMC_NAND_ADDRESS_i location (where i = 0 to 7) places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

- nCS is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- ALE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- CLE and nRE (nOE) are maintained inactive.

Figure 15-81 shows the NAND address latch cycle.

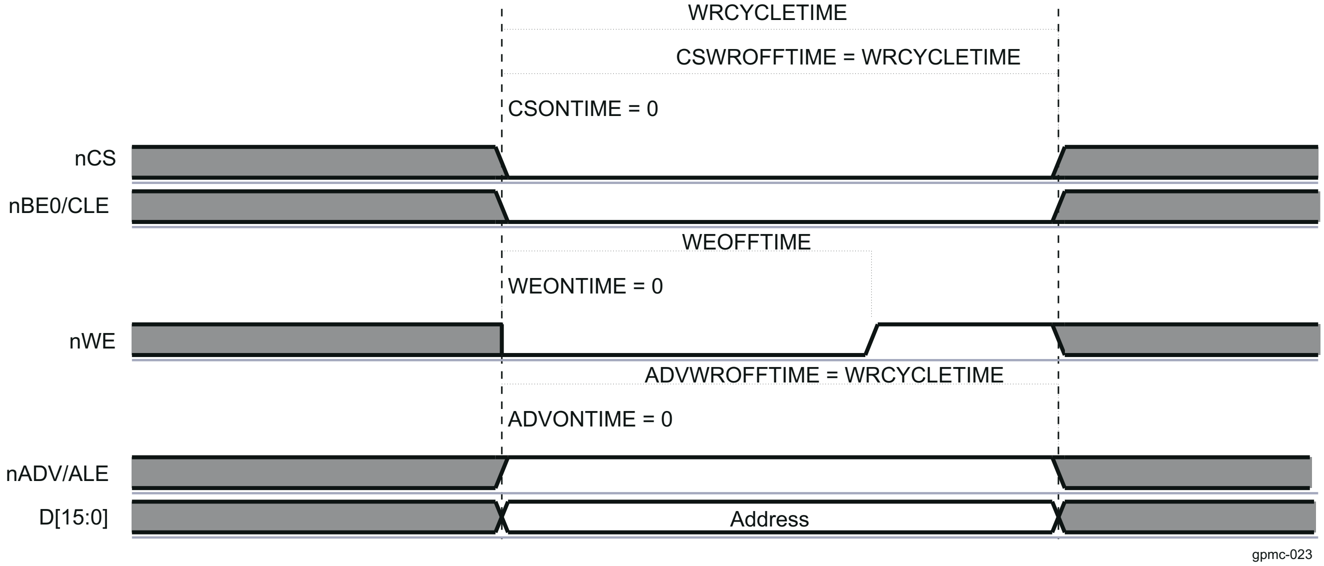


Figure 15-81. NAND Address Latch Cycle

Note

ALE is shared with the nADV output signal and has an inverted polarity from ADV. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, ALE is kept stable.

15.4.4.12.1.5 NAND Device Data Read and Write Phase Control in Stream Mode

NAND device data read and write accesses are achieved through a read or write request to the chip-select-associated memory region at any address location in the region or through a read or write request to the GPMC_NAND_DATA_i location (where i = 0 to 7) mapped in the chip-select-associated control register region. GPMC_NAND_DATA_i is not a true register, but an address location to enable nRE or nWE signal control. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.

- nCS is controlled by the CSONTIME and CSRDOFFTIME timing parameters.
- nRE is controlled by the OEONTIME and OEOFFTIME timing parameters.
- To take advantage of nRE high-to-data invalid minimum timing value, RDACCESSTIME can be set so that data are effectively captured after nRE deassertion. This allows optimization of NAND read access cycle time completion. For optimal timing parameter settings, see the NAND device and the device timing parameters.

ALE, CLE, and nWE are maintained inactive.

Figure 15-82 shows the NAND data read cycle.

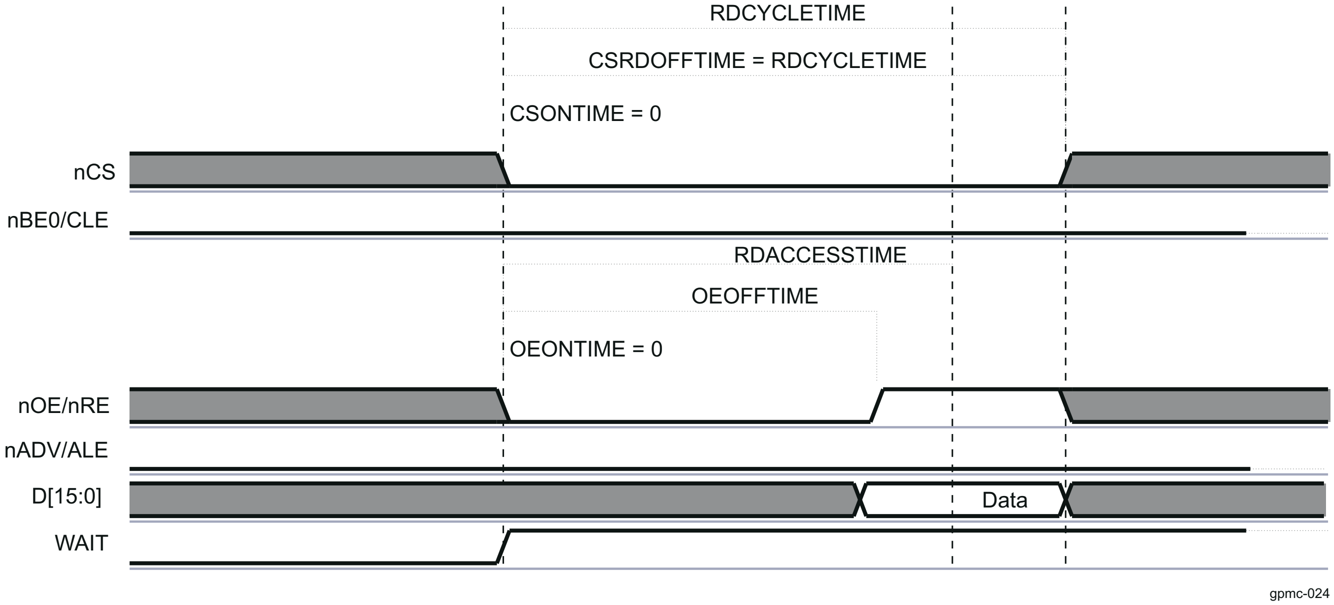


Figure 15-82. NAND Data Read Cycle

Writing data to the GPMC_NAND_DATA_i location or to any location in the associated chip-select memory region activates an asynchronous write access.

- nCS is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE, CLE, and nRE (nOE) are maintained inactive.

Figure 15-83 shows the NAND data write cycle.

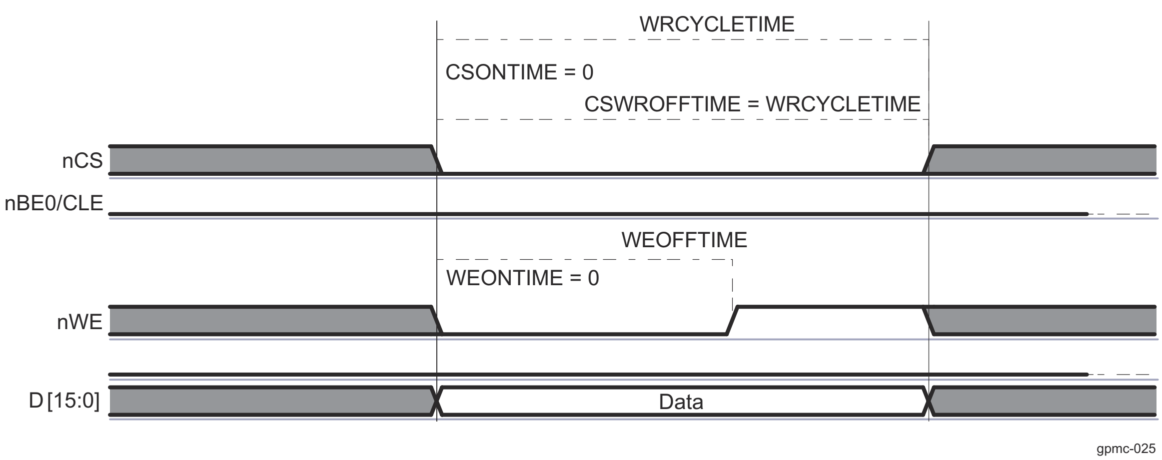


Figure 15-83. NAND Data Write Cycle

15.4.4.12.1.6 NAND Device General Chip-Select Timing Control Requirement

For most NAND devices, read data access time is dominated by nCS-to-data-valid timing and has faster nRE-to-data-valid timing. Successive accesses with nCS deassertions between accesses are affected by this timing constraint. Because accesses to a NAND device can be interleaved with other chip-select accesses, there is no certainty that nCS always stays low between two accesses to the same chip-select. Moreover, an nCS deassertion time between the same chip-select NAND accesses is likely to be required as follows: the nCS deassertion requires programming CYCLETIME and RDACCESSTIME according to the nCS-to-data-valid critical timing.

To get full performance from NAND read and write accesses, the prefetch engine can dynamically reduce the following on back-to-back NAND accesses (to the same memory) and suppress the minimum nCS high pulse width between accesses:

- RDCYCLETIME
- WRCYCLETIME
- RDACCESSTIME
- WRACCESSTIME
- CSRDOFFTIME
- CSWROFFTIME
- ADVRDOFFTIME
- ADVWROFFTIME
- OEOFFTIME
- WEOFFTIME

For more information about optimal prefetch engine access, see Section 15.4.4.12.4, *Prefetch and Write-Posting Engine*.

Some NAND devices require minimum write-to-read idle time, especially for device-status read accesses following status-read command programming (write access). If such write-to-read transactions are used, a minimum nCS high pulse width must be set. For this, CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDELAY must be set according to the appropriate timing requirement to prevent any timing violation.

NAND devices usually have an important nRE high-to-data bus in three-state mode. This requires a bus turnaround setting (BUSTURNAROUND = 1) so that the next access to a different chip-select is delayed until the BUSTURNAROUND delay completes. Back-to-back NAND read accesses to the same NAND flash are not affected by the programmed bus turnaround delay.

15.4.4.12.1.7 Read and Write Access Size Adaptation

15.4.4.12.1.7.1 8-Bit-Wide NAND Device

Host 16- and 32-bit word read and write access requests to a chip-select associated with an 8-bit-wide NAND device are split into successive read and write byte accesses to the NAND memory device. Byte access is ordered according to little-endian organization. A NAND 8-bit-wide device must be interfaced on the D0D7 interface bus lane. GPMC data accesses are justified on this bus lane when the cs is associated with an 8-bit-wide NAND device.

15.4.4.12.1.7.2 16-Bit-Wide NAND Device

Host 32-bit word read and write access requests to a chip-select associated with a 16-bit-wide NAND device are split into successive read and write 16-bit word accesses to the NAND memory device. 16-bit word access is ordered according to little-endian organization.

Host byte read and write access requests to a 16-bit-wide NAND device are completed as 16-bit accesses on the device itself, because there is no byte-addressing capability on 16-bit-wide NAND devices. This means that the NAND device address pointer is incremented on a 16-bit word basis and not on a byte basis. For a read access, only the requested byte is given back to the host, but the remaining byte is not stored or saved by the GPMC, and the next byte or 16-bit word read access gets the next 16-bit word NAND location. For a write access, the invalid byte part of the 16-bit word is driven to FF, and the next byte or 16-bit word write access programs the next 16-bit word NAND location.

Generally, byte access to a 16-bit-wide NAND device must be avoided, especially when ECC calculation is enabled. 8- or 16-bit ECC-based computations are corrupted by a byte read to a 16-bit-wide NAND device, because the nonrequested byte is considered invalid on a read access (not captured on the external data bus; FF is fed to the ECC engine) and is set to FF on a write access.

Host requests (read/write) issued in the chip-select memory region are translated in successive single or split accesses (read/write) to the attached device. Therefore, incrementing 32-bit burst requests are translated in multiple 32-bit sequential accesses following the access adaptation of the 32-bit to 8- or 16-bit device.

15.4.4.12.2 NAND Device-Ready Pin

The NAND memory device provides a ready pin to indicate data availability after a block/page opening and to indicate that data programming is complete. The ready pin can be connected to one of the wait GPMC input pins; data read accesses must not be tried when the ready pin is sampled inactive (device is not ready) even if the associated chip-select WAITREADMONITORING bit field is set. The duration of the NAND device busy state after the block/page opening is so long (up to 50 micro second) that accesses occurring when the ready pin is sampled inactive can stall GPMC access and eventually cause a system time-out.

Note

If a read access to a NAND flash is done using wait monitoring mode, the device is blocked during a page opening, and so is the GPMC. If the correct settings are used, other chip-selects can be used while the memory processes the page opening command.

To avoid a time-out caused by a block/page opening delay in NAND flash, disable the wait pin monitoring for read and write accesses (that is, set the GPMC_CONFIG1_i[21] WAITWRITEMONITORING and GPMC_CONFIG1_i[22] WAITREADMONITORING bits to 0, where i = 0 to 7), and use one of the following methods instead:

- Use software to poll the WAITxSTATUS bit (where x = 0 to 2) of the GPMC_STATUS .
- Configure an interrupt that is generated on the WAIT signal change (through the GPMC_IRQENABLE register bits[11:8]).

Even if the READWAITMONITORING bit is not set, the external memory nR/B pin status is captured in the programmed wait bit in the GPMC_STATUS register.

The READWAITMONITORING bit method must be used for other memories than NAND flash, if they require the use of a WAIT signal.

15.4.4.12.2.1 Ready Pin Monitored by Software Polling

The ready signal state can be monitored through the GPMC_STATUS WAITxSTATUS bit (where x = 0 to 2). Software must monitor the ready pin only when the signal is declared valid. Refer to the NAND device timing parameters to set the correct software temporization to monitor ready only after the invalid window is complete from the last read command written to the NAND device.

15.4.4.12.2.2 Ready Pin Monitored by Hardware Interrupt

Each gpmc_wait input pin can generate an interrupt when a wait-to-no-wait transition is detected. Depending on whether the GPMC_CONFIG WAITxPINPOLARITY bits (where x = 0 to 2) is active low or active high, the wait-to-no-wait transition is a low-to-high external WAIT signal transition or a high-to-low external WAIT signal transition, respectively.

The wait transition pin detector must be cleared before any transition detection. This is done by writing 1 to the WAITxEDGEDETECTIONSTATUS bit (where x = 0 to 2) of the GPMC_IRQSTATUS register according to the gpmc_wait pin used for the NAND device-ready signal monitoring. To detect a wait-to-no-wait transition, the transition detector requires a wait active time detection of a minimum of two GPMC_FCLK cycles. Software must incorporate precautions to clear the wait transition pin detector before wait (busy) time completes.

A wait-to-no-wait transition detection can issue a GPMC interrupt if the WAITxEDGEDETECTIONENABLE bit in the GPMC_IRQENABLE register is set and if the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register is set.

The WAITMONITORINGTIME bit field does not affect wait-to-no-wait transition time detection.

It is also possible to poll the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register according to the gpmc_wait pin used for NAND device ready signal monitoring.

15.4.4.12.3 ECC Calculator

The GPMC includes an error code correction (ECC) calculator circuitry that enables ECC calculation on the fly during data read or data program (that is, write) operations. The page size supported by the ECC calculator in one calculation/context is 512 bytes.

The user can choose from two different algorithms with different error correction capabilities through the GPMC_ECC_CONFIG[16] ECCALGORITHM bit:

1. Hamming code for 1-bit error code correction on 8- or 16-bit NAND flash organized with page size greater than 512 bytes
2. Bose-Chaudhurl-Hocquenghem (BCH) code for 4- to 16-bit error correction

The GPMC does not handle the error code correction directly. During writes, the GPMC computes parity bits. During reads, the GPMC provides enough information for the processor to correct errors without reading the data buffer all over again.

The Hamming code ECC is based on a 2-dimensional (2D) (row and column) bit parity accumulation. This parity accumulation is accomplished on the programmed number of bytes or 16-bit words read from the memory device, or is written to the memory device in stream mode.

Because the ECC engine includes only one accumulation context, it can be allocated to only one chip-select at a time through the GPMC_ECC_CONFIG[3:1] ECCCS bit field. Even if two chip-selects use different ECC algorithms, one the Hamming code and the other a BCH code, they must define separate ECC contexts because some of the ECC registers are common to all types of algorithms.

15.4.4.12.3.1 Hamming Code

All references to ECC in this subsection refer to the 1-bit error correction Hamming code.

The ECC is based on a 2D (row and column) bit parity accumulation known as the Hamming code. The parity accumulation is done for a programmed number of bytes or 16-bit word read from the memory device or written to the memory device in stream mode.

There is no automatic error detection or correction, and the software NAND driver must read the multiple ECC calculation results, compare them to the expected code value, and take the appropriate corrective actions according to the error handling strategy (ECC storage in spare byte, error correction on read, block invalidation).

The ECC engine includes a single accumulation context. It can be allocated to a single designated chip-select at a time, and parallel computations on different chip-selects are not possible. Because it is allocated to a single chip-select, the ECC computation is not affected by interleaved GPMC accesses to other chip-selects and devices. The ECC accumulation is sequentially processed in the order of data read from or written to the memory on the designated chip-select. The ECC engine does not differentiate read accesses from write accesses and does not differentiate data from command or status information. Software must ensure that only relevant data are passed to the NAND flash memory while the ECC computation engine is active.

The starting NAND page location must be programmed first, followed by an ECC accumulation context reset with an ECC enabling, if required. The NAND device accesses discussed in the following sections must be limited to data read or write until the specified number of ECC calculations is complete.

15.4.4.12.3.1.1 ECC Result Register and ECC Computation Accumulation Size

The GPMC includes up to nine ECC result registers (GPMC_ECCj_RESULT, where j = 1 to 9) to store ECC computation results when the specified number of bytes or 16-bit words has been computed.

The ECC result registers are used sequentially: one ECC result is stored in one ECC result register on the list, the next ECC result is stored in the next ECC result register on the list, and so forth, until the last ECC computation. The value of the GPMC_ECCj_RESULT register is valid only when the programmed number of bytes or 16-bit words has been accumulated, which means that the same number of bytes or 16-bit words has been read from or written to the NAND device in sequence.

The GPMC_ECC_CONTROL[3:0] ECCPOINTER bit field must be set to the correct value to select the ECC result register to be used first in the list for the incoming ECC computation process. The ECCPointer can be read to determine which ECC register is used in the next ECC result storage for the ongoing ECC computation. The value of the GPMC_ECCj_RESULT register (where j = 1 to 9) can be considered valid when ECCPOINTER equals j + 1. When the GPMC_ECCj_RESULT register (where j = 9) is updated, ECCPOINTER is frozen at 10, and ECC computing is stopped (ECCENABLE = 0).

The ECC accumulator must be reset before any ECC computation accumulation process. The GPMC_ECC_CONTROL[8] ECCCLEAR bit must be set to 1 (nonpersistent bit) to clear the accumulator and all ECC result registers.

For each ECC result (each GPMC_ECCj_RESULT register, where j = 1 to 9), the number of bytes or 16-bit words used for ECC computing accumulation can be selected from between two programmable values.

The ECCjRESULTSIZE bits (where j = 1 to 9) in the GPMC_ECC_SIZE_CONFIG register select which programmable size value (ECCSIZE0 or ECCSIZE1) must be used for this ECC result (stored in the GPMC_ECCj_RESULT register).

The ECCSIZE0 and ECCSIZE1 bit fields allow selection of the number of bytes or 16-bit words used for ECC computation accumulation. Any even values from 2 to 512 are allowed.

Flexibility in the number of ECCs computed and the number of bytes or 16-bit words used in the successive ECC computations enables different NAND page error-correction strategies. Usually based on 256 or 512 bytes and on 128 or 256 16-bit word, the number of ECC results required is a function of the NAND device page size. Specific ECC accumulation size can be used when computing the ECC on the NAND spare byte.

For example, with a 2-KiB data page, 8-bit-wide NAND device, eight ECCs accumulated on 256 bytes can be computed and added to one extra ECC computed on the 24 spare bytes area where the eight ECC results used for comparison and correction with the computed data page ECC are stored. The GPMC then provides nine GPMC_ECCj_RESULT registers (j = 1 to 9) to store the results. In this case, ECCSIZE0 is set to 256, and ECCSIZE1 is set to 24; the ECC[1:8] RESULTSIZE bits are set to 0, and the ECC9RESULTSIZE bit is set to 1.

15.4.4.12.3.1.2 ECC Enabling

The GPMC_ECC_CONFIG[3:1] ECCCS bit field selects the allocated chip-select. The GPMC_ECC_CONFIG[0] ECCENABLE bit enables ECC computation on the next detected read or write access to the selected chip-select.

The following fields must not be changed or cleared while an ECC computation is in progress:

- CCPOINTER
- ECCCLEAR
- ECCSIZE
- ECCjRESULTSIZE (where j = 1 to 9)
- ECC16B
- ECCCS

The ECC accumulator and ECC result register must not be changed or cleared while an ECC computation is in progress.

Table 15-426 describes the ECC enable settings.

Table 15-426. ECC Enable Settings

Bit Field	Register	Value	Comments
ECCCS	GPMC_ECC_CONFIG	0–3	Selects the chip-select where ECC is computed
ECC16B	GPMC_ECC_CONFIG	0/1	Selects column number for ECC calculation
ECCCLEAR	GPMC_ECC_CONTROL	0–7	Clears all ECC result registers
ECCPOINTER	GPMC_ECC_CONTROL	0–7	A write to this bit field selects the ECC result register where the first ECC computation is stored. Set to 1 by default.

Table 15-426. ECC Enable Settings (continued)

Bit Field	Register	Value	Comments
ECCSIZE1	GPMC_ECC_SIZE_CONFIG	0x00–0xFF	Defines ECCSIZE1
ECCSIZE0	GPMC_ECC_SIZE_CONFIG	0x00–0xFF	Defines ECCSIZE0
ECCjRESULTSIZE (j from 1 to 9)	GPMC_ECC_SIZE_CONFIG	0/1	Selects the size of ECCn result register
ECCENABLE	GPMC_ECC_CONFIG	1	Enables the ECC computation

15.4.4.12.3.1.3 ECC Computation

The ECC algorithm is a multiple parity bit accumulation computed on the odd and even bit streams extracted from the byte or Word16 streams. The parity accumulation is split into row and column accumulations, as shown in Figure 15-84 and Figure 15-85. The intermediate row and column parities are used to compute the upper level row and column parities. Only the final computation of each parity bit is used for ECC comparison and correction.

P1o = bit7 XOR bit5 XOR bit3 XOR bit1 on each byte of the data stream
P1e = bit6 XOR bit4 XOR bit2 XOR bit0 on each byte of the data stream

P2o = bit7 XOR bit6 XOR bit3 XOR bit2 on each byte of the data stream
P2e = bit5 XOR bit4 XOR bit1 XOR bit0 on each byte of the data stream

P4o = bit7 XOR bit6 XOR bit5 XOR bit4 on each byte of the data stream
P4e = bit3 XOR bit2 XOR bit1 XOR bit0 on each byte of the data stream

Each column parity bit is XORed with the previous accumulated value.

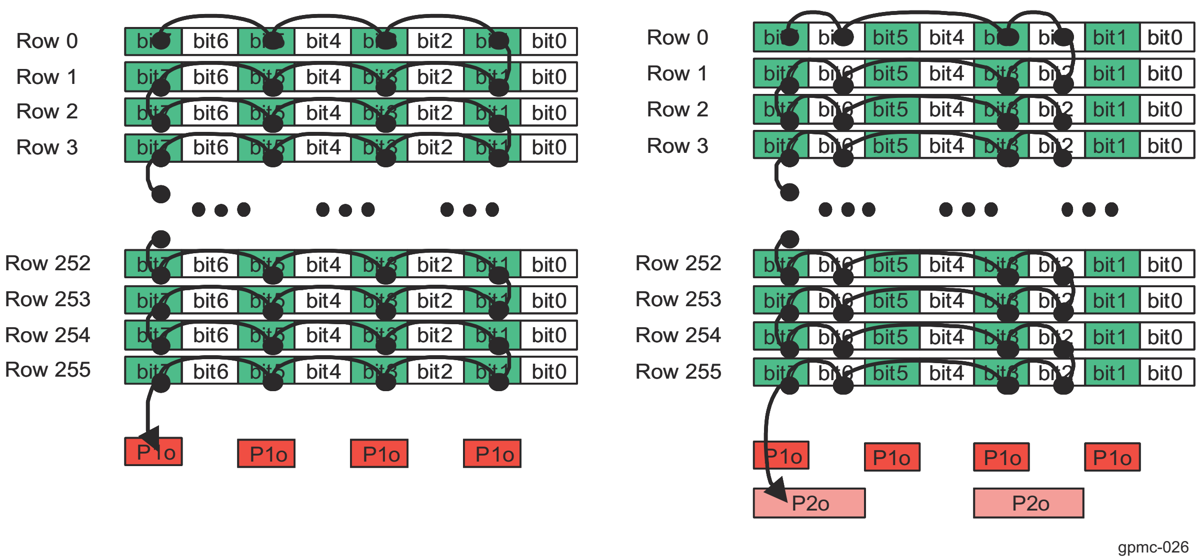


Figure 15-84. Hamming Code Accumulation Algorithm (1/2)

For line parities, the bits of each new data are XORed together, and line parity bits are computed as described below:

P8e = row0 XOR row2 XOR row4 XOR ... XOR row254
P8o = row1 XOR row3 XOR row5 XOR ... XOR row255

P16e = row0 XOR row1 XOR row4 XOR row5 XOR ... XOR row252 XOR row 253
P16o = row2 XOR row3 XOR row6 XOR row7 XOR ... XOR row254 XOR row 255

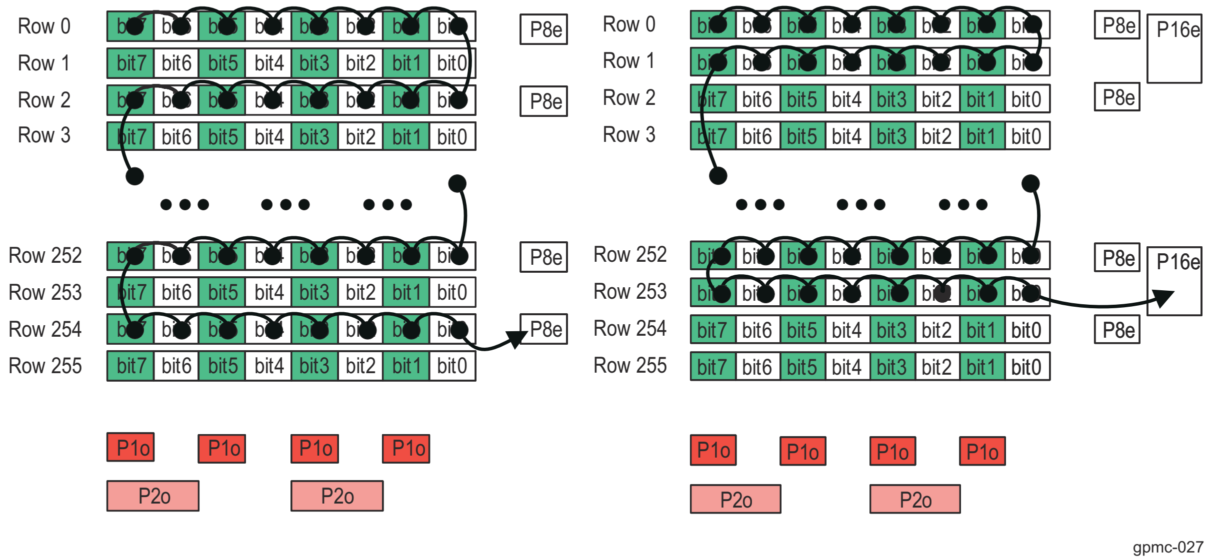


Figure 15-85. Hamming Code Accumulation Algorithm (2/2)

Unused parity bits in the result registers are set to 0.

Figure 15-86 shows ECC computation for a 256-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and sixteen row parity bits (P8o-P16o-P32o--P1024o for odd parities, and P8e-P16e-P32e--P1024e for even parities).

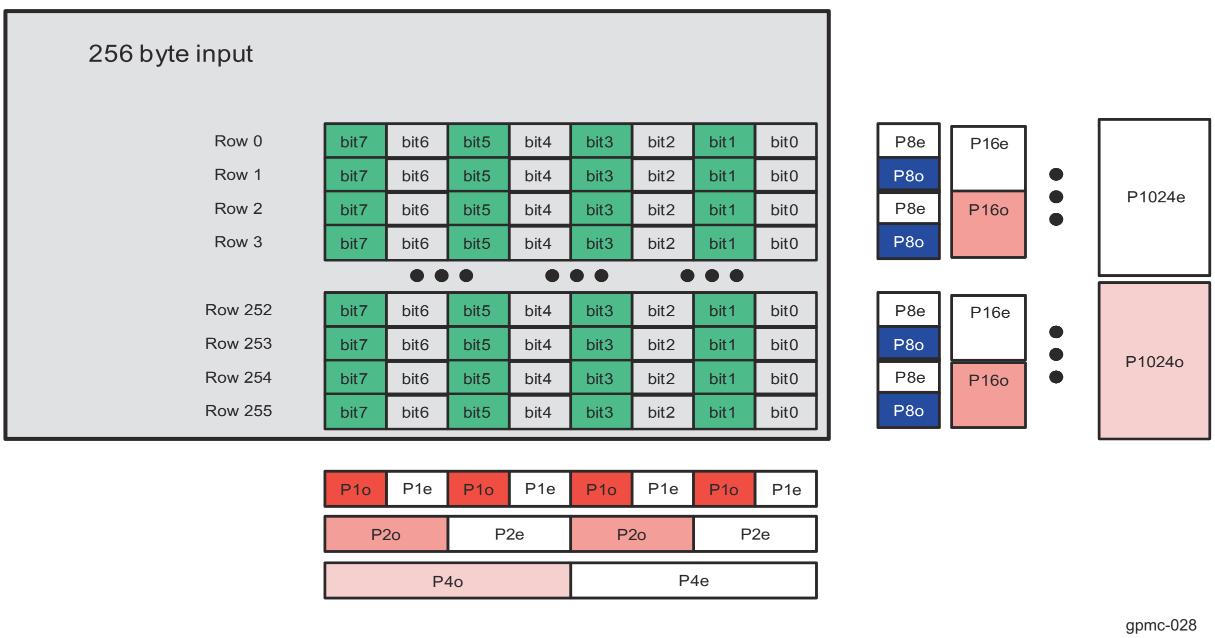


Figure 15-86. ECC Computation for a 256-Byte Data Stream (Read or Write)

Figure 15-87 shows ECC computation for a 512-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and eighteen row parity bits (P8o-P16o-P32o--P1024o- - P2048o for odd parities, and P8e-P16e-P32e--P1024e- P2048e for even parities).

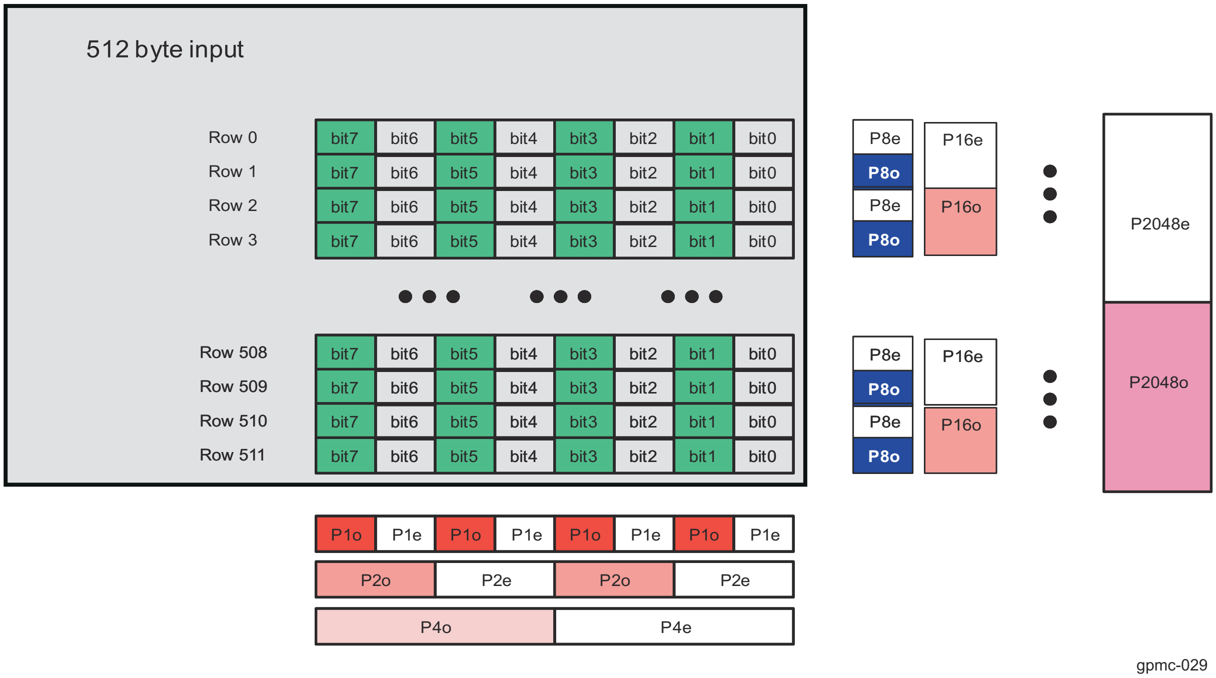


Figure 15-87. ECC Computation for a 512-Byte Data Stream (Read or Write)

For a 2-KiB page, four 512 bytes ECC calculations plus 1 for the spare area are required. Results are stored in the GPMC_ECCj_RESULT registers (where j = 1 to 9).

15.4.4.12.3.1.4 ECC Comparison and Correction

To detect an error, the computed ECC result must be XORed with the parity value stored in the spare area of the accessed page.

- If the result of this logical XOR is all 0s, no error is detected and the read data is correct.
- If every second bit in the parity result is a 1, 1 bit is corrupted and is located at bit address (P2048o, P1024o, P512o, P256o, P128o, P64o, P32o, P16o, P8o, P4o, P2o, P1o). Software must correct the corresponding bit.
- If only 1 bit in the parity result is 1, it is an ECC error and the read data is correct.

15.4.4.12.3.1.5 ECC Calculation Based on 8-Bit Word

The 8-bit-based ECC computation is used for 8-bit-wide NAND device interfacing.

The 8-bit-based ECC computation can be used for 16-bit-wide NAND device interfacing to get backward compatibility on the error-handling strategy used with 8-bit-wide NAND devices. In this case, the 16-bit-wide data read from or written to the NAND device is fragmented into 2 bytes. According to little-endian access, the LSB of the 16-bit-wide data is ordered first in the byte stream used for 8-bit-based ECC computation.

15.4.4.12.3.1.6 ECC Calculation Based on 16-Bit Word

ECC computation based on an 16-bit word is used for 16-bit-wide NAND device interfacing. This ECC computation is not supported when interfacing an 8-bit-wide NAND device, and the GPMC_ECC_CONFIG[7] ECC16B bit must be set to 0 when interfacing an 8-bit-wide NAND device.

The parity computation based on 16-bit words affects the row and column parity mapping. The main difference is that the odd and even parity bits P8o and P8e are computed on rows for an 8-bit-based ECC and on columns for a 16-bit based ECC. Figure 15-88 and Figure 15-89 show a 128 Word16 ECC computation scheme and a 256 16-bit word ECC computation scheme.

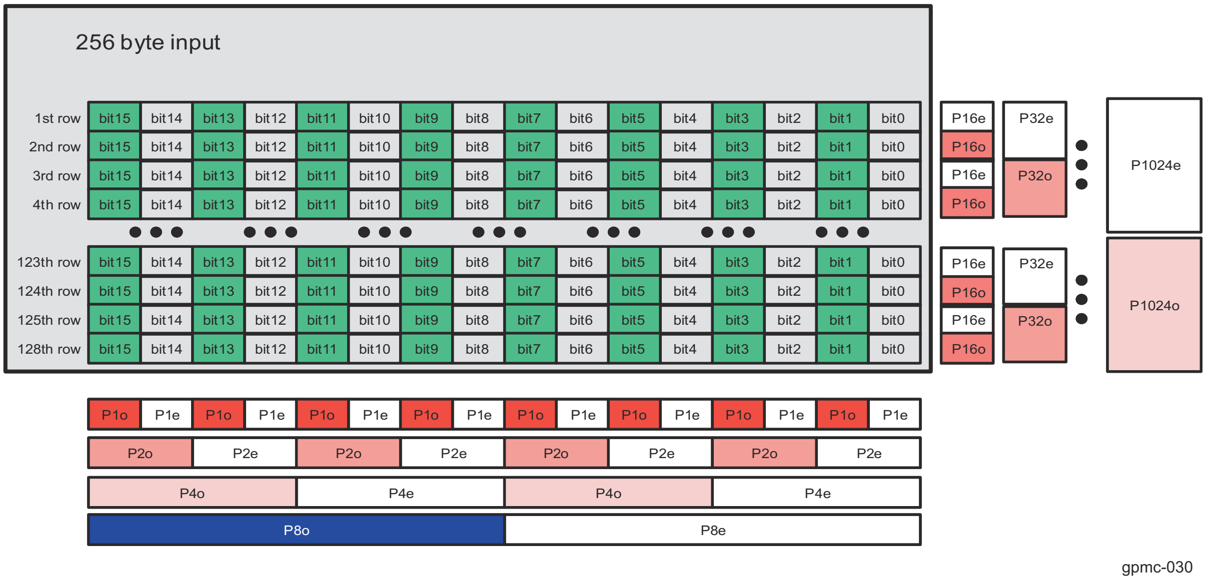


Figure 15-88. 128 Word16 ECC Computation

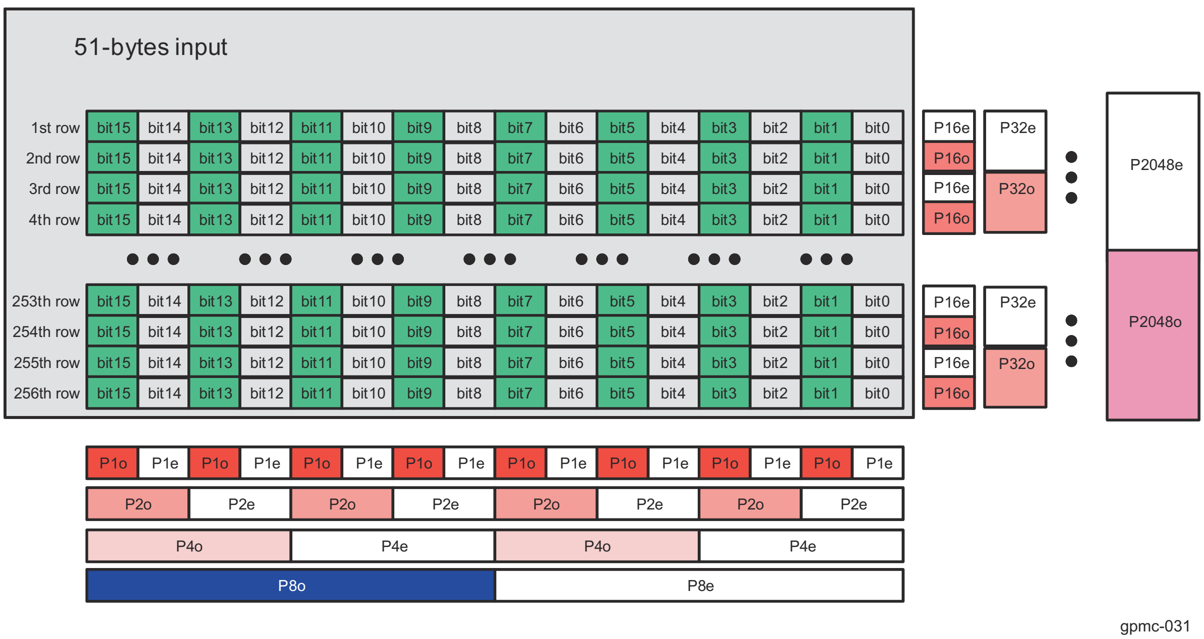


Figure 15-89. 256 Word16 ECC Computation

15.4.4.12.3.2 BCH Code

All references to ECC in this subsection refer to the 4- to 16-bit error correction BCH code.

15.4.4.12.3.2.1 Requirements

1. Read and write accesses to a NAND flash take place by whole pages, in a predetermined sequence: first the data byte page itself, and then some spare bytes, including the BCH ECC (and other information). The NAND device can cache a full page, including spares, for read and write accesses.
   - Typical page write sequence:
     – Sequential write to NAND cache of main data plus spare data, for a page. ECC is calculated on the fly. Calculated ECC can be inserted on the fly in the spares or replaced by dummy accesses.

  - When the calculated ECC is replaced by dummy accesses, it must be written to the cache in a second, separate phase. The ECC module is disabled during that time.
  - NAND writes its cache line (page) to the array.
- Typical page read sequence:
  - Sequential read of a page. ECC is calculated on the fly.
  - The status of the ECC module buffers determines the presence of errors.

2. Accesses to several memories can be interleaved by the GPMC, but only one of those memories at a time can be a NAND using the BCH engine; in other words, only one BCH calculation (for example, for a single page) can be ongoing at any time. The sequential nature of NAND accesses ensures that the data is always written or read out in the same order. BCH-relevant accesses are selected by the chip-selects of the GPMC.
3. Each page can hold up to 4KiB of data, spare bytes not included. This means up to 8 × 512-byte BCH messages. Because all the data is written or read out first, followed by the BCH ECC, the BCH engine must be able to hold eight 104-bit remainders or syndromes (or smaller, 52-bit ones) at the same time.

   The BCH module can store all remainders internally. After the page start, an internal counter is used to detect the 512-byte sector boundaries. On those boundaries, the current remainder is stored and the divider reset for the next calculation. At the end of the page, the BCH module contains all remainders.
4. NAND access cycles hold 8 or 16 bits of data each (1 or 2 bytes); Each NAND cycle takes at least four cycles of the GPMC internal clock. This means the NAND flash timing parameters must define a RDCYCLETIME and a WRCYCLETIME of at least four clock cycles after optimization when using the BCH calculator.
5. The spare area is assumed to be large enough to hold the BCH ECC; that is, to have a message of at least 13 bytes available per 512-byte sector of data. The zone of unused spare area by the ECC may or may not be protected by the same ECC scheme, by extending the BCH message beyond 512 bytes (the maximum codeword is 1023 bytes long, ECC included, which leaves much space to cover spare bytes).

15.4.4.12.3.2.2 Memory Mapping of BCH Codeword

BCH encoding considers a block of data to protect as a polynomial message M(x). In a standard case, 512 bytes of data (that is, 2^{12} bits = 4096 bits) are seen as a polynomial of degree $2^{12} – 1 = 4095$, with parameters ranging from M0 to M4095. For 512 bytes of data, 52 bits are required for 4-bit error correction, 104 bits are required for 8-bit error correction, and 207 bits are required for 16-bit error correction. The ECC is a remainder polynomial R(x) of degree 103 (or 51, depending on the selected mode). The complete codeword C(x) is the concatenation of M(x) and R(x), as described in Table 15-427.

Table 15-427. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits)

	Message M(x)			ECC R(x)		
Bit Number	M4095	...	M0	R103	...	R0

If the message is extended by the addition of spare bytes to be protected by the same ECC, the principle is still valid. For example, a 3-byte extension of the message gives a polynomial message M(x) of degree ((512 + 3) × 8) – 1 = 4119, for a total of 3 + 13 = 16 spare bytes of spare, all protected as part of the same codeword.

The message and the ECC bits are manipulated and mapped in the GPMC byte-oriented system. The ECC bits are stored in the following registers (where i = 0 to 7):

- GPMC_BCH_RESULT0_i
- GPMC_BCH_RESULT1_i
- GPMC_BCH_RESULT2_i
- GPMC_BCH_RESULT3_i

15.4.4.12.3.2.2.1 Memory Mapping of Data Message

The data message mapping must follow the following rules:

- Bit endianness within a byte is little-endian; that is, the bytes LSB is also the lowest-degree polynomial parameter: a byte b7-b0 (with b0 the LSB) represents a segment of polynomial $b7 * x^{(7+i)} + b6 * x^{(6+i)} + ... + b0 * x^{i}$

- The message is mapped in the NAND starting with the highest-order parameters, that is, in the lowest addresses of a NAND page.
- Byte endianness within the 16-bit words in the NAND is big-endian (that is, the same message mapped in 8- and 16-bit memories has the same content at the same byte address).

Note

The BCH module has no visibility over actual addresses. The most important point is the sequence of data words the BCH sees. However, the NAND page is always scanned incrementally in read and write accesses, which produces the mapping patterns described in the following.

Table 15-428 and Table 15-429 describe the mapping of the same 512-byte vector (typically, a BCH message) in the NAND memory space. The byte address is only an offset modulo 512 (0x200), because the same page may contain several contiguous 512-byte sectors (BCH blocks). The LSB and MSB are, respectively, the bits M0 and M(2^{12}–1) of the codeword mapping discussed previously. In both cases the data vectors are aligned; that is, their boundaries coincide with the RAM data word boundaries.

Table 15-428. Aligned Message Byte Mapping in 8-Bit NAND

Byte Offset	8-Bit Word
0x000	(MSB) Byte 511 (0x1FF)
0x001	Byte 510 (0x1FE)
...	...
0x1FF	Byte 0 (0x0) (LSB)

Table 15-429. Aligned Message Byte Mapping in 16-Bit NAND

Byte Offset	16-Bit Word MSB	16-Bit Word LSB
0x000	Byte 510 (0x1FE)	(MSB) Byte 511 (0x1FF)
0x002	Byte 508 (0x1FC)	Byte 509 (0x1FD)
...	...	...
0x1FE	Byte 0 (0x0)	(LSB) Byte 1 (0x1)

Table 15-430 through Table 15-435 list the mapping in memory of arbitrarily-sized messages, starting on access (byte or 16-bit word) boundaries for more clarity. Note that message may actually start and stop on arbitrary nibbles. A nibble is a 4-bit entity. The unused nibbles are not discarded, and they can still be used by the BCH module, but as part of the next message section (for example, on the ECC of another sector).

Table 15-430. Aligned Nibble Mapping of Message in 8-Bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Least Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
...	...	...
S/2 – 2	Nibble 3	Nibble 2
S/2 – 1	Nibble 1	Nibble 0 (LSB)

Table 15-431. Misaligned Nibble Mapping of Message in 8-Bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Least Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
...	...	...
(S + 1) / 2 – 2	Nibble 2	Nibble 1

Table 15-431. Misaligned Nibble Mapping of Message in 8-Bit NAND (continued)

Byte Offset	8-Bit Word	
(S + 1) / 2 – 1	Nibble 0 (LSB)	

Table 15-432. Aligned Nibble Mapping of Message in 16-Bit NAND

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
S/2 – 4	Nibble 5	Nibble 4	Nibble 7	Nibble 6
S/2 – 2	Nibble 1	Nibble 0 (LSB)	Nibble 3	Nibble 2

Table 15-433. Misaligned Nibble Mapping of Message in 16-Bit NAND (1 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
(S + 1) / 2 – 4	Nibble 4	Nibble 3	Nibble 6	Nibble 5
(S + 1) / 2 – 2	Nibble 0 (LSB)		Nibble 2	Nibble 1

Table 15-434. Misaligned Nibble Mapping of Message in 16-Bit NAND (2 Unused Nibbles)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
(S + 2) / 2 – 4	Nibble 3	Nibble 2	Nibble 5	Nibble 4
(S + 2) / 2 – 2			Nibble 1	Nibble 0 (LSB)

Table 15-435. Misaligned Nibble Mapping of Message in 16-Bit NAND (3 Unused Nibbles)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
(S + 3) / 2 – 4	Nibble 2	Nibble 1	Nibble 4	Nibble 3
(S + 3) / 2 – 2			Nibble 0 (LSB)	

Many other cases exist than those given in the previous tables; for example, where the message does not start on a word boundary.

15.4.4.12.3.2.2.2 Memory-Mapping of the ECC

The ECC (or remainder) is presented by the BCH module as a single 104-bit (or 52-bit), little-endian vector. Software must fetch those 13 bytes (or 6 bytes) from the module interface and then store them to the spare area (page write) in the NAND or to an intermediate buffer for comparison with the stored ECC (page read). There are no constraints on the ECC mapping inside the spare area: it is software-controlled.

It is advised, however, to maintain a coherence in the respective formats of the message or the ECC remainder once they have been read out of the NAND. The error correction algorithm works from the complete codeword (concatenated message and remainder) once an error is detected. The creation of this codeword must be made as straightforward as possible.

There are cases in which the same NAND access contains both data and the ECC protecting that data. This is the case when the data/ECC boundary (which can be on any nibble) does not coincide with an access boundary. The ECC is calculated on the fly following the write. In that case, the write must also contain part of the ECC because it is impossible to insert the ECC on the fly. Instead:

- During the initial page write (BCH encoding), the ECC is replaced by dummy bits. The BCH encoder is by definition turned OFF during the ECC section, so the BCH result is unmodified.
- During a second phase, the ECC is written to the correct location, next to the actual data.
- The completed line buffer is then written to the NAND array.

15.4.4.12.3.2.2.3 Wrapping Modes

For a given wrapping mode, the module automatically goes through a specific number of sections as data is being fed into the module. For each section, the BCH core can be enabled (in which case the data is fed to the BCH divider) or not (in which case the BCH simply counts to the end of the section). When enabled, the data is added to the ongoing calculation for a given sector number (for example, number 0).

Wrapping modes are described as follows. To better understand and see the real-life read and write sequences implemented with each mode, see Section 15.4.4.12.3.2.3, *Supported NAND Page Mappings and ECC Schemes*.

For each mode:

- A sequence describes the mode in pseudo language, with, for each section, the size and the buffer used for ECC processing (if ON). The programmable lengths are size, size0, and size1.
- A checksum condition is given. If the checksum condition is not respected for a given mode, the module behavior is unpredictable. S is the number of sectors in the page; size0 and size1 are the section sizes programmed for the mode, in nibbles.

Wrapping modes 8 through 11 insert a 1-nibble padding where the BCH processing is OFF. This is intended for t = 4 ECC, where ECC is 6 bytes long and the ECC area is expected to include (at least) 1 unused nibble to remain byte-aligned.

15.4.4.12.3.2.2.3.1 Manual Mode (0x0)

This mode is intended for short sequences, added manually to a given buffer through the software data port input. A complete page may be built out of several such sequences.

To process an arbitrary sequence of 4-bit nibbles, accesses to the software data port, containing the appropriate data, must be made. If the sequence end does not coincide with an access boundary (for example, to process 5 nibbles = 20 bits in 16-bit access mode) and those nibbles must be skipped, a number of unused nibbles must be programmed in GPMC_ECC_SIZE_CONFIG[29:22] ECCSIZE1 In the same example, 5 nibbles to process + 3 to discard = 8 nibbles = 2 × 16-bit accesses. Software must set:

- GPMC_ECC_SIZE_CONFIG[19:12] ECCSIZE0 = 0x5
- GPMC_ECC_SIZE_CONFIG[29:22] ECCSIZE1 = 0x3

Note

In the following figures, size and size0 are the same parameter.

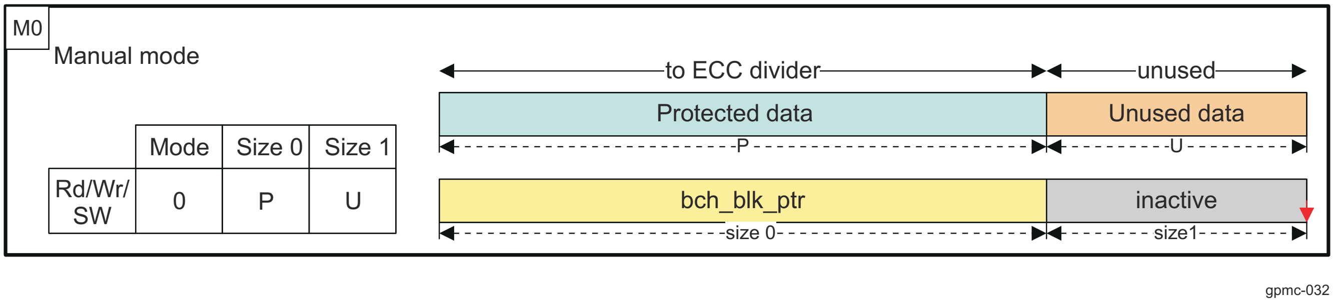


Figure 15-90. Manual Mode Sequence and Mapping

Section processing sequence:

- One time with buffer
  - size0 nibbles of data, processing ON
  - size1 nibbles of unused data, processing OFF

Checksum: size0 + size1 nibbles must fit in a whole number of accesses.

In the following sections, S is the number of sectors in the page.

15.4.4.12.3.2.2.3.2 Mode 0x1

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S – (size0 + size1)

15.4.4.12.3.2.2.3.3 Mode 0xA (10)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - 1 nibble pad spare, processing OFF
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S – (size0 + 1 + size1)

15.4.4.12.3.2.2.3.4 Mode 0x2

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S – (size0 + size1)

15.4.4.12.3.2.2.3.5 Mode 0x3

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – size1)

15.4.4.12.3.2.2.3.6 Mode 0x7

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = size0 + (S – size1)

15.4.4.12.3.2.2.3.7 Mode 0x8

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – (1 + size1))

15.4.4.12.3.2.2.3.8 Mode 0x4

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – size1)

15.4.4.12.3.2.2.3.9 Mode 0x9

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – (1 + size1))

15.4.4.12.3.2.2.3.10 Mode 0x5

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S – (size0 + size1)

15.4.4.12.3.2.2.3.11 Mode 0xB (11)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S – (size0 + 1 + size1)

15.4.4.12.3.2.2.3.12 Mode 0x6

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S – (size0 + size1)

15.4.4.12.3.2.3 Supported NAND Page Mappings and ECC Schemes

The following rules apply to the entire mapping description:

- Main data area (sectors) size is hardcoded to 512 bytes.
- Spare area size is programmable.
- All page sections (of main area data bytes, protected spare bytes, unprotected spare bytes, and ECC) are defined as explained in Section 15.4.4.12.3.2.2.1, *Memory Mapping of Data Message*.

Each of the following sections gives a NAND page mapping example (per-sector spare mappings, pooled spare mapping, per-sector spare mapping, with ECC separated at the end of the page).

In the mapping diagrams, sections that belong to the same BCH codeword have the same color (blue or green); unprotected sections are not covered (orange) by the BCH scheme.

Below each mapping diagram, a write (encoding) and read (decoding: syndrome generation) sequence is given, with the number of the active buffers at each point in time (yellow). In the inactive zones (grey), no computing is taking place but the data counter is still active.

In Figure 15-91 through Figure 15-93, the tables on the left summarize the mode, size0, and size1 parameters to program for, respectively, write and read processing of a page, with the given mapping, where:

- P is the size of spare byte section Protected by the ECC (in nibbles)
- U is the size of spare byte section Unprotected by the ECC (in nibbles)
- E is the size of the ECC (in nibbles)
- S is the number of Sectors per page (two in the current diagrams)

Each time the processing of a BCH block is complete (ECC calculation for write/encoding, syndrome generation for read/decoding, indicated by red arrows), the update pointer is pulsed. The processing for block 0 can be the first or the last to complete, depending on the NAND page mapping and operation (read or write). All examples show a page size of 1 KiB + spares; that is, S = 2 sectors of 512 bytes. The same principles can be extended to larger pages by adding more sectors.

The actual BCH codeword size is used during the error location work to restrict the search range: by definition, errors can happen only in the codeword that was actually written to the NAND, not in the mathematical codeword of $n = 2^{13} – 1 = 8191$ bits; that codeword (higher-order bits) is all-zero and implicit during computations.

The actual BCH codeword size depends on the mode, the programmed sizes, and the sector number (all sizes in nibbles):

- Spares mapped and protected per sector (below: see M1-M2-M3-M9-M10):
  - All sectors: (512) + P + E
- Spares pooled and protected by sector 0 (below: see M5-M6):
  - Sector 0 codeword: (512) + P + E
  - Other sectors: (512) + E
- Unprotected spares (below: see M4-M7-M8-M11-M12):
  - All codewords (512) + E

15.4.4.12.3.2.3.1 Per-Sector Spare Mappings

In these schemes, each 512-byte sector of the main area has its own dedicated section of the spare area. The spare area of each sector is composed of:

- ECC, which must be located after the data it protects
- Other data, which may or may not be protected by the ECC for its sector.

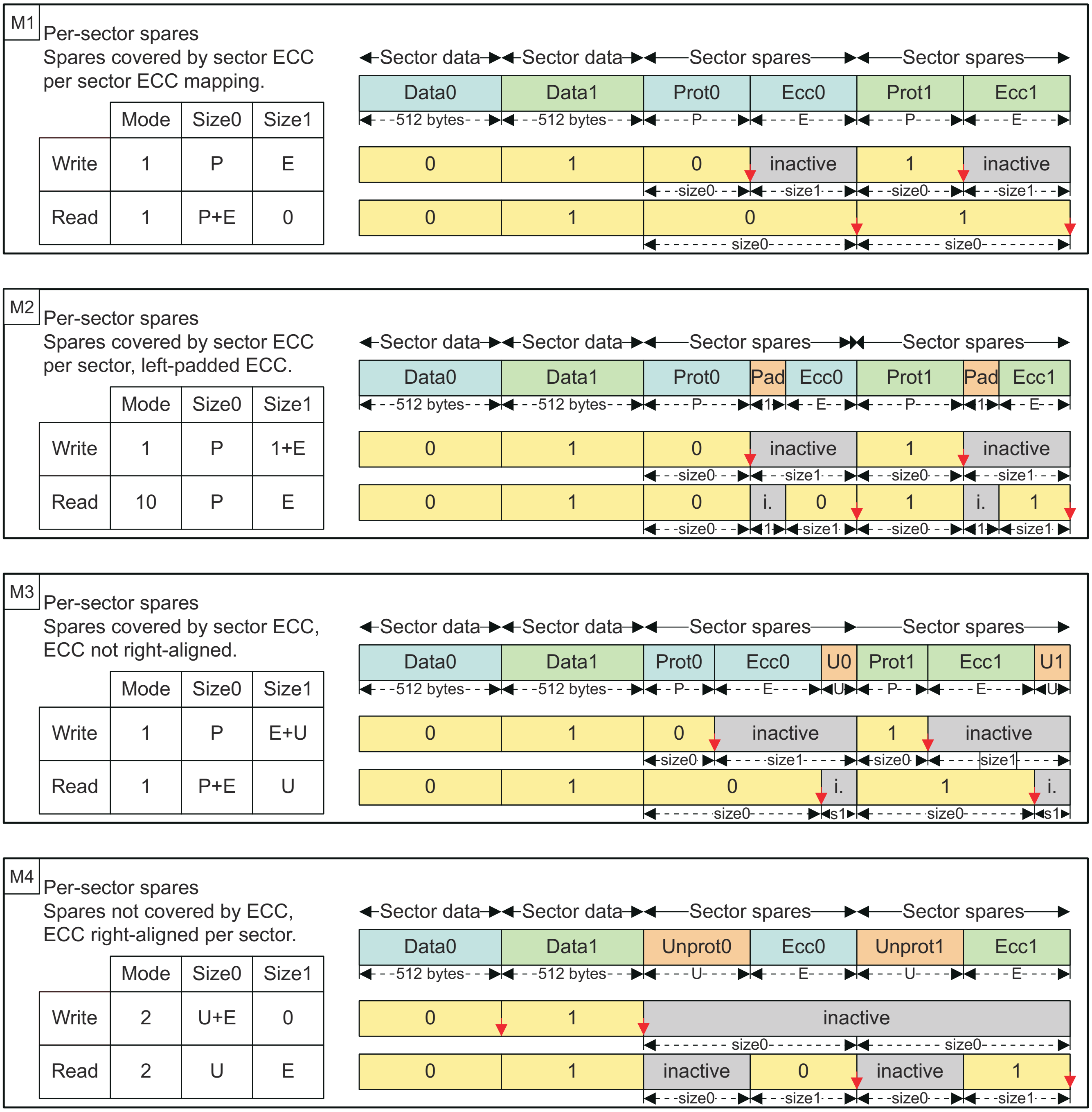


Figure 15-91. NAND Page Mapping and ECC: Per-Sector Schemes

15.4.4.12.3.2.3.2 Pooled Spare Mapping

In the following schemes, the spare area is pooled for the page.

- The ECC of each sector is aligned at the end of the spare area.
- The non-ECC spare data may or may not be covered by the ECC of sector 0.

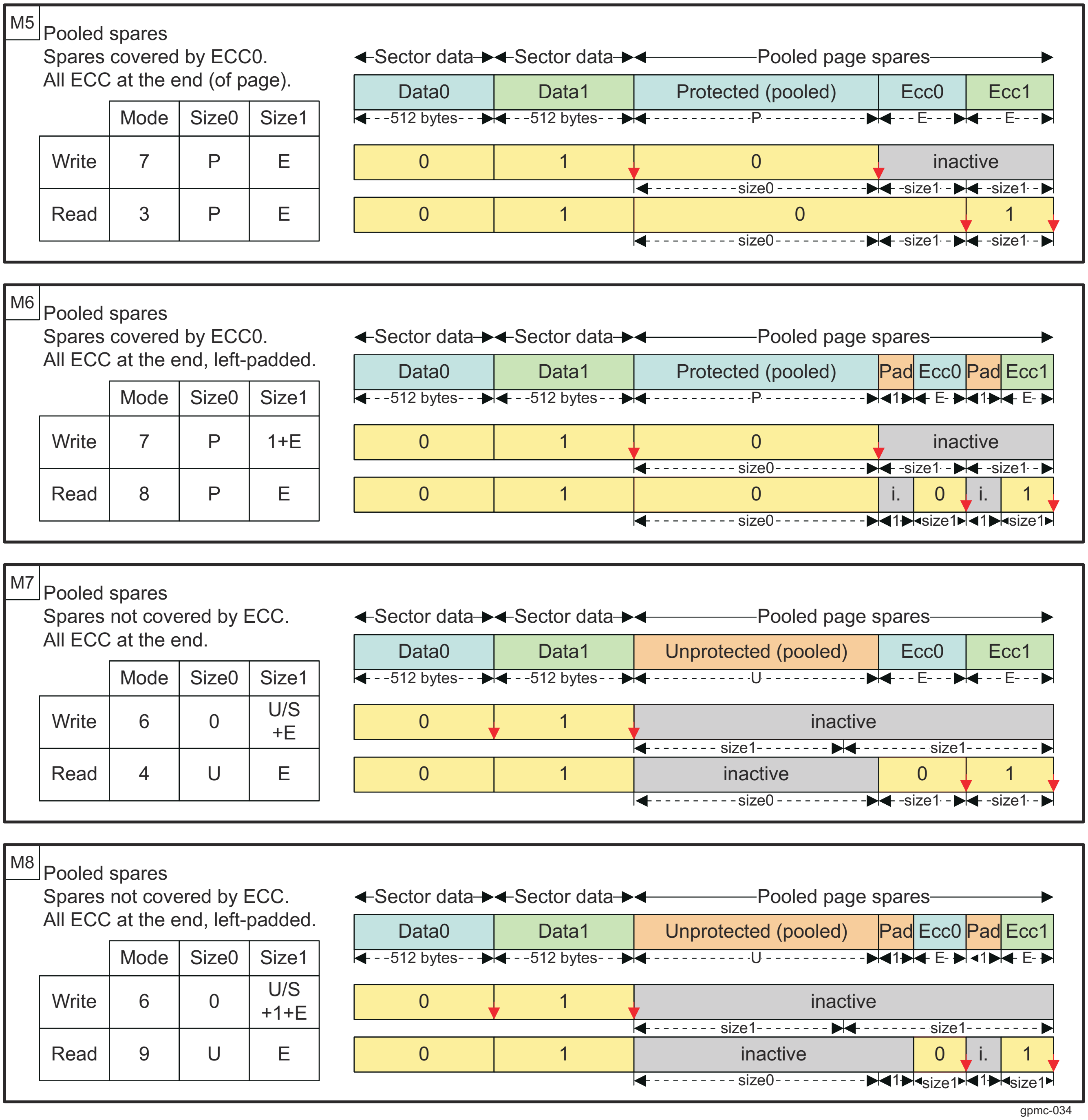


Figure 15-92. NAND Page Mapping and ECC: Pooled Spare Schemes

15.4.4.12.3.2.3.3 Per-Sector Spare Mapping, with ECC Separated at the End of the Page

In these schemes, each 512-byte sector of the main area is associated with two sections of the spare area.

- ECC section, all aligned at the end of the page
- Other data section, aligned before the ECCs, each of which may or may not be protected by the ECC for its sector.

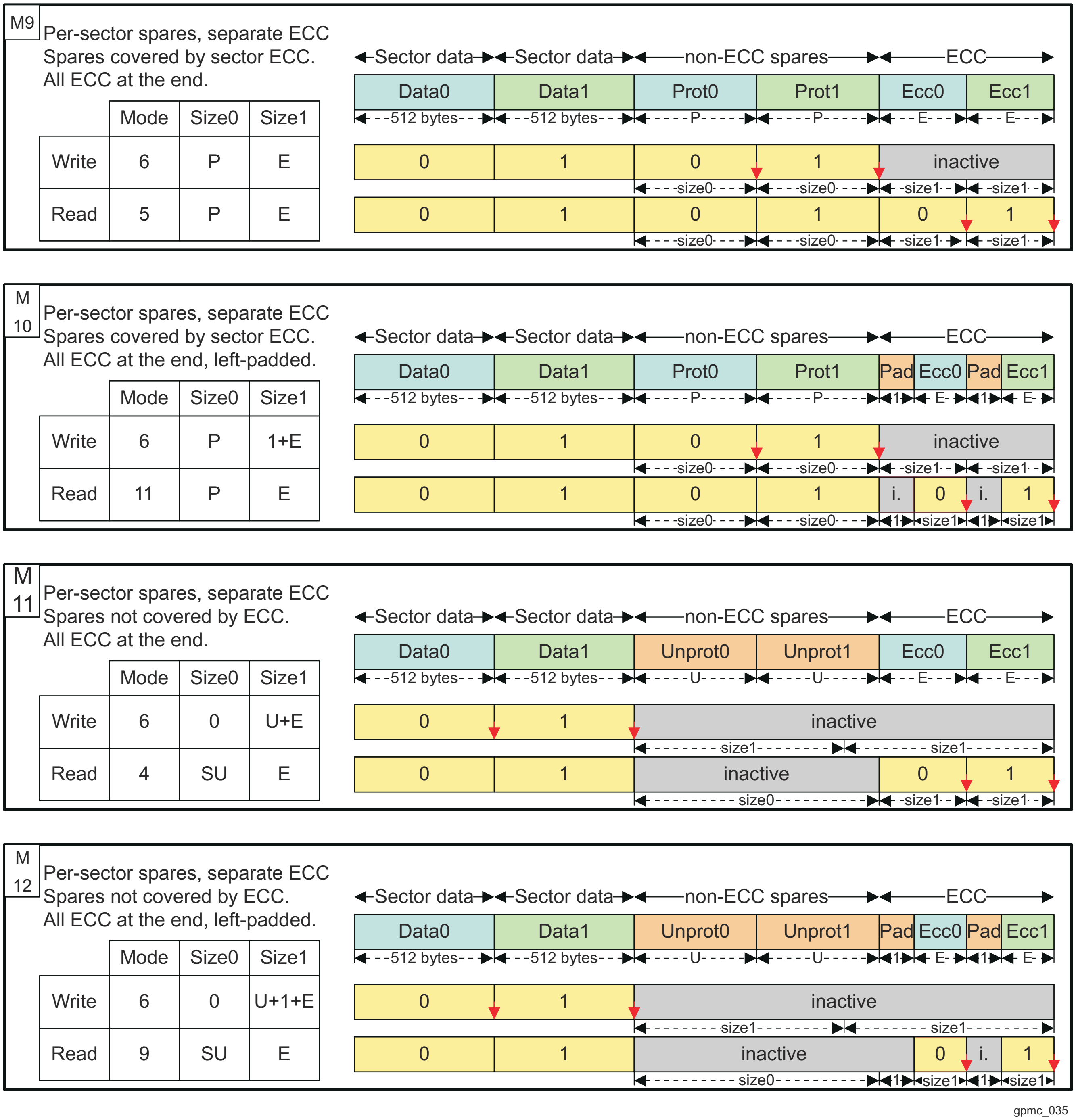


Figure 15-93. NAND Page Mapping and ECC: Per-Sector Schemes, With Separate ECC

15.4.4.12.4 Prefetch and Write-Posting Engine

NAND device data access cycles are usually much slower than the MPU system frequency; such NAND read or write accesses issued by the processor affect the overall system performance, especially considering long read or write sequences required for NAND page loading or programming. To minimize this effect on system performance, the GPMC includes a prefetch and write-posting engine, which can be used to read from or write to any chip-select location in a buffered manner.

The prefetch and write-posting engine is a simplified embedded-access requester that presents requests to the access engine on a user-defined chip-select target. The access engine interleaves these requests with any request coming from the L3 interface; as a default, the prefetch and write-posting engine has the lowest priority.

The prefetch and write-posting engine is dedicated to data-stream access (as opposed to random data access); thus, it is primarily dedicated to NAND support. The engine does not include an address generator; the request is limited to chip-select target identification. It includes a 64-byte FIFO associated with a DMA request synchronization line, for optimal DMA-based use.

The prefetch and write-posting engine uses an embedded 64-byte (32 16-bit word) FIFO to prefetch data from the NAND device in read mode (prefetch mode) or to store host data to be programmed into the NAND device in write mode (write-posting mode). The FIFO draining and filling (read and write) can be controlled by a device host processor through interrupt synchronization (an interrupt is triggered whenever a programmable threshold is reached) or by a device DMA module through DMA request synchronization, with a programmable request byte size in prefetch or posting mode.

The prefetch and write-posting engine includes a single memory pool. Therefore, only one mode, read or write, can be used at any given time. In other words, the prefetch and write-posting engine is a single-context engine that can be allocated to only one chip-select at a time for a read prefetch or a write-posting process.

The engine does not support atomic command and address phase programming and is limited to linear memory read or write access. As a consequence, it is limited to NAND data-stream access. The engine relies on the MPU NAND software driver to control block and page opening with the correct data address pointer initialization, before the engine can read from or write to the NAND memory device.

Once started, engine data read and write sequencing is based solely on FIFO location availability and until the total programmed number of bytes is read or written.

Any host-concurrent accesses to a different chip-select are correctly interleaved with ongoing engine accesses. The engine has the lowest priority access so that host accesses to a different chip-select do not suffer a large latency.

A round-robin arbitration scheme can be enabled to ensure minimum bandwidth to the prefetch and write-posting engine in the case of back-to-back direct memory requests to a different chip-select. If the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit is enabled, the arbitration grants the prefetch and write posting engine access to the GPMC bus for a number of requests programmed in the GPMC_PREFETCH_CONFIG1[19:16] PFPWWEIGHTEDPRIO bit field.

The prefetch/write-posting engine read or write request is routed to the access engine with the chip-select destination ID. After the required arbitration phase, the access engine processes the request as a single access with the data access size equal to the device size specified in the corresponding chip-select configuration.

Note

The destination chip-select configuration must be set to the NAND protocol-compatible configuration for which address lines are not used (the address bus is not changed from its current value). Selecting a different chip-select configuration can produce undefined behavior.

15.4.4.12.4.1 General Facts About the Engine Configuration

The engine can be configured only if the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is deasserted.

The engine must be correctly configured in prefetch or write-posting mode and must be linked to a NAND chip-select before it can be started. The chip-select is linked using the GPMC_PREFETCH_CONFIG1[26:24] ENGINECSSELECTOR bit field.

In prefetch and write-posting modes, the engine uses byte or 16-bit word access requests, respectively, for an 8- or 16-bit-wide NAND device attached to the linked chip-select. The FIFOTHRESHOLD and TRANSFERCOUNT bit fields must be programmed accordingly as a number of bytes.

When the GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE bit is set, the FIFO entry on the L3 interconnect port side is accessible at any address in the associated chip-select memory region. When the ENABLEENGINE bit is set, any host access to this chip-select is rerouted to the FIFO input. Directly accessing the NAND device linked to this chip-select from the host is still possible through the following registers (where i = 0 to 7):

- GPMC_NAND_COMMAND_i
- GPMC_NAND_ADDRESS_i
- GPMC_NAND_DATA_i

The FIFO entry on the L3 interconnect port can be accessed with byte, 16-bit word, or 32-bit word access size, according to little-endian format, even though the FIFO input is 32 bits wide.

The FIFO control is made easier through the use of interrupts or DMA requests associated with the FIFOTHRESHOLD bit field. The GPMC_PREFETCH_STATUS[30:24] FIFOPOINTER bit field monitors the number of available bytes to be read in prefetch mode or the number of free empty slots that can be written in write-posting mode. The GPMC_PREFETCH_STATUS[13:0] COUNTVALUE bit field monitors the number of remaining bytes to be read or written by the engine according to the value of the TRANSFERCOUNT bit field. The FIFOPOINTER and COUNTVALUE bit fields are always expressed as a number of bytes even if a 16-bit-wide NAND device is attached to the linked chip-select.

In prefetch mode, when the FIFOPOINTER equals 0 (that is, the FIFO is empty), a host read access receives the byte last read from the FIFO as its response. In case of 32-bit word or 16-bit word read accesses, the last byte read from the FIFO is copied the required number of times to fit the requested word size. In write-posting mode, when the FIFOPOINTER equals 0 (that is, the FIFO is full), a host write overwrites the last FIFO byte location. There is no underflow or overflow error reporting in the GPMC.

15.4.4.12.4.2 Prefetch Mode

The prefetch mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is cleared.

The MPU NAND software driver must issue the block and page opening (READ) command with the correct data address pointer initialization before the engine can be started to read from the NAND memory device. The engine is started by asserting the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit. The STARTENGINE bit automatically clears when the prefetch process completes.

If required, the ECC calculator engine must be initialized (that is, reset, configured, and enabled) before the prefetch engine is started so that the ECC is computed correctly on all data read by the prefetch engine.

When the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit is cleared, the prefetch engine starts requesting data as soon as the STARTENGINE bit is set. If using this configuration, the host must monitor the NAND device-ready pin so that it sets the STARTENGINE bit only when the NAND device is in a ready state (that is, data is valid for prefetching).

When the SYNCHROMODE bit is set, the prefetch engine starts requesting data when an active-to-inactive WAIT signal transition is detected. The transition detector must be cleared before any transition detection (see Section 15.4.4.12.2.2, *Ready Pin Monitored by Hardware Interrupt*). The GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR bit field selects which gpmc_wait pin edge detector triggers the prefetch engine in this synchronized mode.

If the STARTENGINE bit is set after the NAND address phase (page opening command), the engine is effectively started only after the actual NAND address phase completion. To prevent GPMC stall during this NAND address phase, set the STARTENGINE bit before NAND address phase completion when in synchronized mode. The prefetch engine starts when an active-to-inactive WAIT signal transition is detected. The STARTENGINE bit is automatically cleared on prefetch process completion.

The prefetch engine issues a read request to fill the FIFO with the amount of data specified by the GPMC_PREFETCH_CONFIG2[13:0] TRANSFERCOUNT bit field.

Table 15-436 describes the prefetch mode configuration.

Table 15-436. Prefetch Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	0	Prefetch engine can be configured only if STARTENGINE is set to 0.
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1[26:24]	0 to 3	Selects the chip-select associated with a NAND device where the prefetch engine is active.
ACCESSMODE	GPMC_PREFETCH_CONFIG1[0]	0	Selects prefetch mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1[14:8]		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG2[13:0]		Selects the number of bytes to be read or written by the engine to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1[3]	0/1	Selects when the engine starts the access to the chip-select
WAITPINSELECT	GPMC_PREFETCH_CONFIG1[17:16]	0 to 1	Selects wait pin edge detector (if GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE = 0x1)
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1[27]	0/1	See Section 15.4.4.12.4.6, *Optimizing NAND Access Using the Prefetch and Write-Posting Engine*.
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG1[30:28]		Number of clock cycle removed to timing parameters
ENABLEENGINE	GPMC_PREFETCH_CONFIG1[7]	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	1	Starts the prefetch engine

15.4.4.12.4.3 FIFO Control in Prefetch Mode

The FIFO can be drained directly by a device host processor or a DMA module channel.

In MPU draining mode, the FIFO status can be monitored through the GPMC_PREFETCH_STATUS[30:24] FIFOPOINTER bit field or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. The FIFOPOINTER indicates the current number of available data to be read; FIFOTHRESHOLDSTATUS set to 1 indicates that at least FIFOTHRESHOLD bytes are available from the FIFO.

An interrupt can be triggered by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. The FIFO interrupt event is logged, and the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must read all the available bytes , or at least enough bytes to get below the programmed FIFO threshold, and the FIFOEVENTSTATUS bit must be cleared to enable further interrupt events. The FIFOEVENTSTATUS bit must always be reset before asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt generation must be enabled after enabling the STARTENGINE bit.

Prefetch completion can be monitored through the GPMC_PREFETCH_STATUS[13:0] COUNTVALUE bit field. COUNTVALUE indicates the number of currently remaining data to be requested according to the TRANSFERCOUNT value. An interrupt can be triggered by the GPMC when the prefetch process is complete (that is, COUNTVALUE equals 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. At prefetch completion, the TERMINALCOUNT interrupt event is also logged, and the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit . The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

Note

The COUNTVALUE value is valid only when the prefetch engine is active (started), and an interrupt is only triggered when COUNTVALUE reaches 0, that is, when the prefetch engine automatically goes from an active to an inactive state.

The number of bytes to be prefetched (programmed in TRANSFERCOUNT) must be a multiple of the programmed FIFOTHRESHOLD to trigger the correct number of interrupts allowing a deterministic and transparent FIFO control. If this guideline is respected, the number of ISR accesses is always required and the FIFO is always empty after the last interrupt is triggered. In other cases, the TERMINALCOUNT interrupt must be used to read the remaining bytes in the FIFO (the number of remaining bytes being lower than the FIFOTHRESHOLD value).

In DMA draining mode, the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes are ready to be read from the FIFO. The DMA channel that owns this DMA request must be programmed so that the number of bytes programmed in FIFOTHRESHOLD is read from the FIFO during the DMA request process. The DMA request is kept active until this number of bytes has effectively been read from the FIFO, and no other DMA request can be issued until the ongoing active request is complete.

In prefetch mode, the TERMINALCOUNT event is also a source of DMA requests if the number of bytes to be prefetched is not a multiple of FIFOTHRESHOLD, the remaining bytes in the FIFO can be read by the DMA channel using the last DMA request. This assumes that the number of remaining bytes to be read is known and controlled through the DMA channel programming model.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive-to-active prefetch (the STARTENGINE bit is set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that the out-of-date active DMA request does not trigger spurious DMA transfers.

15.4.4.12.4.4 Write-Posting Mode

The write-posting mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is set.

The MPU NAND software driver must issue the correct address pointer initialization command (page program) before the engine can start writing data into the NAND memory device. The engine starts when the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is set to 1. The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion (adding ECC handling, if required).

If used, the ECC calculator engine must be started (configured, reset, and enabled) before the posting engine is started so that the ECC parities are calculated properly on all data written by the prefetch engine to the associated chip-select.

In write-posting mode, the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit must be cleared so that posting starts as soon as the STARTENGINE bit is set and the FIFO is not empty.

If the STARTENGINE bit is set after the NAND address phase (page program command), the STARTENGINE setting is effective only after the actual NAND command completion. To prevent GPMC stall during this NAND command phase, set the STARTENGINE bit field before the NAND address completion and ensure that the associated DMA channel is enabled after the NAND address phase.

The posting engine issues a write request when valid data are available from the FIFO and until the programmed GPMC_PREFETCH_CONFIG2[13:0] TRANSFERCOUNT accesses are complete.

The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion. The closing program command phase must be issued only when the full NAND page has been written into the NAND flash write buffer, including the spare area data and the ECC parities, if used.

Table 15-437 describes the write-posting configuration.

Table 15-437. Write-Posting Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	0	Write-posting engine can be configured only if STARTENGINE is set to 0.

Table 15-437. Write-Posting Mode Configuration (continued)

Bit Field	Register	Value	Comments
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1[26:24]	0 to 3	Selects the chip-select associated with a NAND device where the prefetch engine is active
ACCESSMODE	GPMC_PREFETCH_CONFIG1[0]	1	Selects write-posting mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1[14:8]		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG2[13:0]		Selects the number of bytes to be read or written by the engine from/to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1[3]	0	Engine starts the access to chip-select as soon as STARTENGINE is set.
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1[27]	0/1	See Section 15.4.4.12.4.6, *Optimizing NAND Access Using the Prefetch and Write-Posting Engine*.
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG1[30:28]		
ENABLEENGINE	GPMC_PREFETCH_CONFIG1[7]	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	1	Starts the prefetch engine

15.4.4.12.4.5 FIFO Control in Write-Posting Mode

The FIFO can be filled directly by a device host processor or a DMA module channel.

In MPU filling mode, the FIFO status can be monitored through the FIFOPOINTER or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. FIFOPOINTER indicates the current number of available free byte places in the FIFO, and the FIFOTHRESHOLDSTATUS bit, when set, indicates that at least FIFOTHRESHOLD free byte places are available in the FIFO.

An interrupt can be issued by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must write enough bytes to fill the FIFO or to get below the programmed threshold, and the FIFOEVENTSTATUS bit must be cleared to get further interrupt events. The FIFOEVENTSTATUS bit must always be cleared before asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt must be enabled after enabling the STARTENGINE bit.

The posting completion can be monitored through the GPMC_PREFETCH_STATUS[13:0] COUNTVALUE bit field. COUNTVALUE indicates the current number of remaining data to be written based on the value of the TRANSFERCOUNT bit field. An interrupt is issued by the GPMC when the write-posting process completes (that is, COUNTVALUE equal to 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit . The TERMINALCOUNTSTATUS bit must always be cleared before asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

Note

The value of the COUNTVALUE bit field is valid only if the write-posting engine is active and started, and an interrupt is issued only when COUNTVALUE reaches 0; that is, when the posting engine automatically goes from active to inactive.

In DMA filling mode, the DMAMode bit field in the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes-free places are available in the FIFO. The DMA channel that owns this DMA request must be programmed so that a number of bytes equal to the value programmed in the FIFOTHRESHOLD bit field are written into the FIFO during the DMA access. The DMA request remains active until the associated number of bytes has effectively been written into the FIFO, and no other DMA request can be issued until the ongoing active request completes.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive-to-active prefetch (STARTENGINE set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that an out-of-date active DMA request does not trigger spurious DMA transfers.

In write-posting mode, the DMA or MPU fills the FIFO with no consideration for the associated byte enables. Any byte stored in the FIFO is written into the memory device.

15.4.4.12.4.6 Optimizing NAND Access Using the Prefetch and Write-Posting Engine

Access time to a NAND memory device can be optimized for back-to-back accesses if the associated nCS signal is not deasserted between accesses. The GPMC access engine can track prefetch engine accesses to optimize the access timing parameter programmed for the allocated chip-select, if no accesses to other chip-selects (that is, interleaved accesses) occur. Similarly, the access engine also eliminates CYCLE2CYCLEDELAY even if CYCLE2CYCLESAMECSEN is set. This capability is limited to the prefetch and write-posting engine accesses, and MPU accesses to a NAND memory device (through the defined chip-select memory region or through the GPMC_NAND_DATA_i location, where i = 0 to 7) are never optimized.

The GPMC_PREFETCH_CONFIG1[27] ENABLEOPTIMIZEDACCESS bit must be set to enable optimized accesses. To optimize access time, the GPMC_PREFETCH_CONFIG1[30:28] CYCLEOPTIMIZATION bit field defines the number of GPMC_FCLK cycles to be suppressed from the following timing parameters:

- RDCYCLETIME
- WRCYCLETIME
- RDACCESSTIME
- WRACCESSTIME
- CSOFFTIME
- ADVOFFTIME
- OEOFFTIME
- WEOFFTIME

Figure 15-94 shows that in the case of back-to-back accesses to the NAND flash through the prefetch engine, CYCLE2CYCLESAMECSEN is forced to 0 when using optimized accesses. The first access uses the regular timing settings for this chip-select. All accesses after this one use settings reduced by x clock cycles, x being defined by the GPMC_PREFETCH_CONFIG1[30:28] CYCLEOPTIMIZATION bit field.

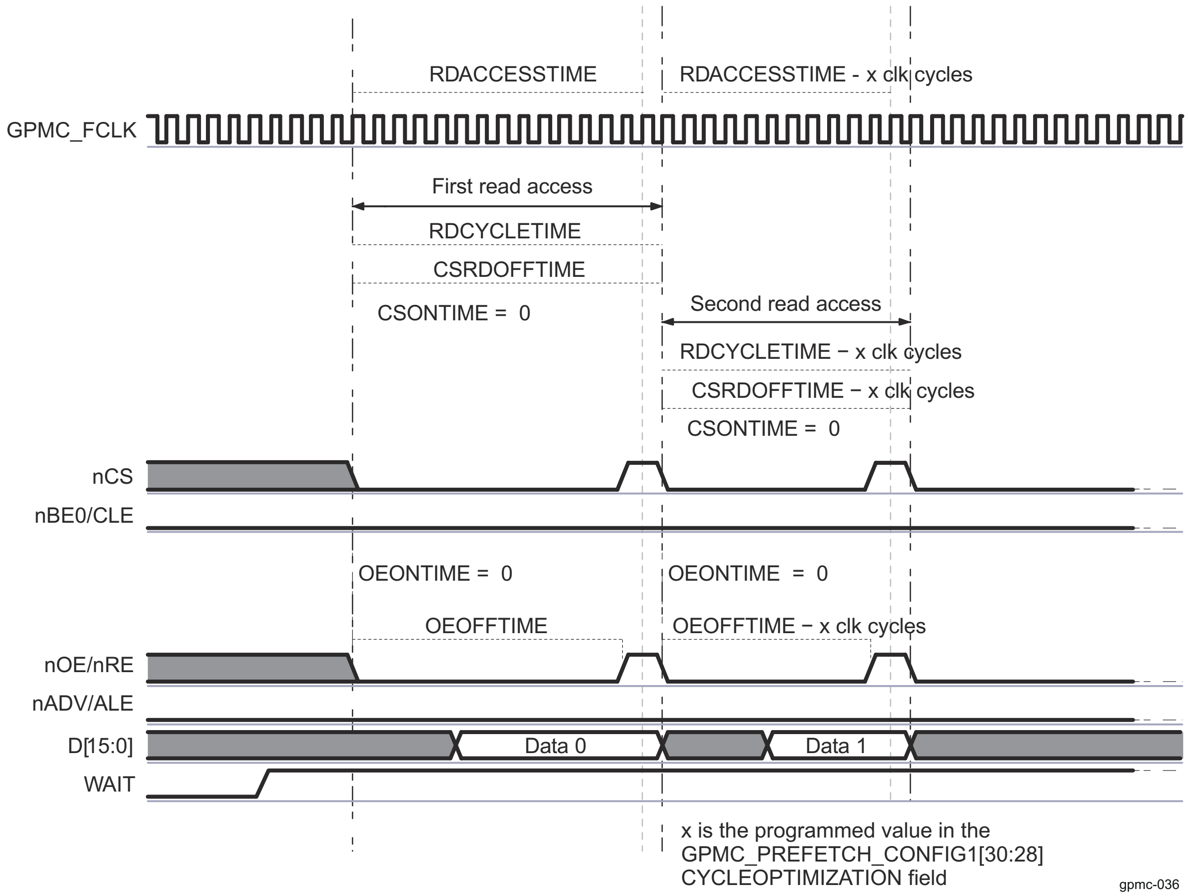


Figure 15-94. NAND Read Cycle Optimization Timing Description

15.4.4.12.4.7 Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects

Any on-going read or write access from the prefetch and write-posting engine is completed before an access to any other chip-select can be initiated. As a default, the arbiter uses a fixed-priority algorithm, and the prefetch and write-posting engine has the lowest priority. The maximum latency added to access starting time in this case equals the RDCYCLETIME or WRCYCLETIME (optimized or not) plus the requested BUSTURNAROUND delay for bus turnaround completion programmed for the chip-select to which the NAND device is connected.

Alternatively, a round-robin arbitration can be used to prioritize accesses to the external bus. This arbitration scheme is enabled by setting the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit. When a request to another chip-select is received while the prefetch and write-posting engine is active, priority is given to the new request. The request processed thereafter is the prefetch and write-posting engine request, even if another interconnect request is passed in the mean time. The engine keeps control of the bus for an additional number of requests programmed in the GPMC_PREFETCH_CONFIG1[19:16] PFPWWEIGHTEDPRIO bit field. Control is then passed to the direct interconnect request.

As an example, the round-robin arbitration scheme is selected with PFPWWEIGHTEDPRIO set to 0x2. Considering that the prefetch and write-posting engine and the interconnect interface are always requesting access to the external interface, the GPMC grants priority to the direct interconnect access for one request. The GPMC then grants priority to the engine for three requests, and finally back to the direct interconnect access, until the arbiter is reset when one of the two initiators stops initiating requests.

15.4.5 GPMC Basic Programming Model

15.4.5.1 GPMC High-Level Programming Model Overview

The goal of the basic high-level programming model is to introduce a top-down approach to users that need to configure the GPMC module.

Figure 15-95 and Table 15-438 through Table 15-440 show a programming model top-level diagram for the GPMC, and a description of each step. Each block of the diagram is described in one of the following sections through a set of registers to configure.

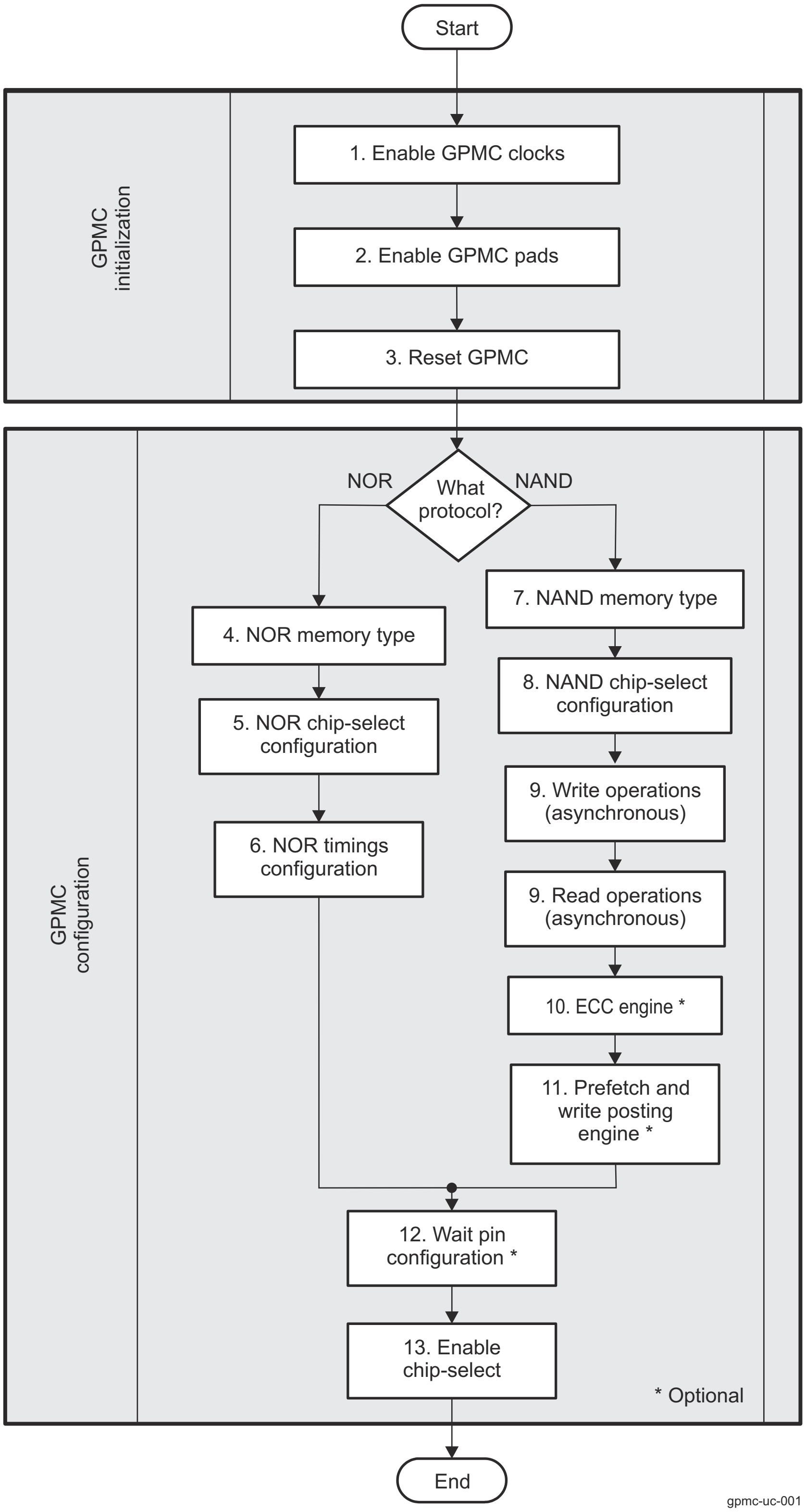


Figure 15-95. Programming Model Top-Level Diagram

Table 15-438. GPMC Initialization

Step	Description
Enable GPMC clocks.	Module interface and functional clocks must be enabled. See *Power, Reset, and Clock Management*.
Enable GPMC pads.	Module-specific pad multiplexing and configuration must be set in the control module. See *Pad Configuration Registers* in *Control Module*.
Reset GPMC.	See Table 15-441.

Table 15-439. GPMC Configuration in NOR Mode

Step	Description
NOR Memory Type	See Table 15-442.
NOR Chip-Select Configuration	See Table 15-443.
NOR Timings Configuration	See Table 15-444.
Wait Pin Configuration	See Table 15-452.
Enable Chip-Select	See Table 15-453.

Table 15-440. GPMC Configuration in NAND Mode

Step	Description
NAND Memory Type	See Table 15-447.
NAND Chip-Select Configuration	See Table 15-448.
Write Operations (Asynchronous)	See Table 15-449.
Read Operations (Asynchronous)	See Table 15-449.
ECC Engine	See Table 15-450.
Prefetch and Write-Posting Engine	See Table 15-451.
Wait Pin Configuration	See Table 15-452.
Enable Chip-Select	See Table 15-453.

15.4.5.2 GPMC Initialization

Table 15-469 describes the settings required to prepare the GPMC; that is enabling its clock and pads, and proceeding to a GPMC reset.

Table 15-441. Reset GPMC

Subprocess Name	Register/Bit Field	Value
Start a software reset.	GPMC_SYSCONFIG[1] SOFTRESET	0x1
Wait until	GPMC_SYSSTATUS[0] RESETDONE =	0x1

15.4.5.3 GPMC Configuration in NOR Mode

This section gives a generic configuration for parameters related to the NOR memory connected to the GPMC.

Table 15-442. NOR Memory Type

Subprocess Name	Register / Bit Field	Value
Set the NOR protocol.	GPMC_CONFIG1_i[11:10] DEVICETYPE	0x0
Set a device size.	GPMC_CONFIG1_i[13:12] DEVICESIZE	x
Select an address and data multiplexing protocol.	GPMC_CONFIG1_i[9] MUXADDDATA	x
Set the attached device page length.	GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH	x
Set the wrapping burst capabilities.	GPMC_CONFIG1_i[31] WRAPBURST	x
Select a timing signals latencies factor.	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Select an output clock frequency [(1)].	GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER	x
Choose an output clock activation time [(1)].	GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME	x
Set a single or multiple access for read operations [(1)].	GPMC_CONFIG1_i[30] READMULTIPLE	x

Table 15-442. NOR Memory Type (continued)

Subprocess Name	Register / Bit Field	Value
Set a synchronous or asynchronous mode for read operations.	GPMC_CONFIG1_i[29] READTYPE	x
Set a single or multiple access for write operations.	GPMC_CONFIG1_i[28] WRITEMULTIPLE	x
Set a synchronous or asynchronous mode for write operations.	GPMC_CONFIG1_i[27] WRITETYPE	x

(1) Applies only to synchronous configurations (or nonmultiplexed asynchronous for multiple access one)

Table 15-443. NOR Chip-Select Configuration

Subprocess Name	Register/Bit Field	Value
Select the chip-select base address.	GPMC_CONFIG7_i[5:0] BASEADDRESS	x
Select the chip-select mask address.	GPMC_CONFIG7_i[11:8] MASKADDRESS	x

Table 15-444. NOR Timings Configuration

Subprocess Name	Register/Bit Field	Value
Configure adequate timing parameters in various memory modes.	See Section 15.4.5.6, *GPMC Timing Parameters*	

Table 15-445. Wait Pin Configuration

Subprocess Name	Register/Bit Field	Value
Enable or disable wait pin monitoring for read operations.	GPMC_CONFIG1_i[22] WAITREADMONITORING	x
Enable or disable wait pin monitoring for write operations.	GPMC_CONFIG1_i[21] WAITWRITEMONITORING	x
Select a wait pin monitoring time.	GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME	x
Choose the input wait pin for the chip-select.	GPMC_CONFIG1_i[17:16] WAITPINSELECT	x

Table 15-446. Enable Chip-Select

Subprocess Name	Register/Bit Field	Value
When all parameters are configured, enable the chip-select.	GPMC_CONFIG7_i[6] CSVALID	x

15.4.5.4 GPMC Configuration in NAND Mode

This section gives a generic configuration for parameters related to the NAND memory connected to the GPMC.

Table 15-447. NAND Memory Type

Subprocess Name	Register/Bit Field	Value
Set the NAND protocol.	GPMC_CONFIG1_i[11:10] DEVICETYPE	0x2
Set a device size.	GPMC_CONFIG1_i[13:12] DEVICESIZE	x
Set the address and data multiplexing protocol to non-multiplexed attached device.	GPMC_CONFIG1_i[9] MUXADDDATA	0x0
Select a timing signals latencies factor.	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Set a synchronous or asynchronous mode and a single or multiple access for read and write operations.	See Section 15.4.5.5, *Set Memory Access*.	x

Table 15-448. NAND Chip-Select Configuration

Subprocess Name	Register/Bit Field	Value
Select the chip-select base address.	GPMC_CONFIG7_i[5:0] BASEADDRESS	x
Select the chip-select minimum granularity (16MiB).	GPMC_CONFIG7_i[11:8] MASKADDRESS	x

Table 15-449. Asynchronous Read and Write Operations

Subprocess Name	Register/Bit Field	Value
Configure adequate timing parameters in asynchronous modes	See Section 15.4.5.6, *GPMC Timing Parameters*.	

Table 15-450. ECC Engine

Subprocess Name	Register/Bit Field	Value
Select the ECC result register where the first ECC computation is stored (applies only to Hamming).	GPMC_ECC_CONTROL[3:0] ECCPOINTER	x (2)
Clear all ECC result registers.	GPMC_ECC_CONTROL[8] ECCCLEAR	Write 1 to clear.
Define ECCSIZE0 and ECCSIZE1.	GPMC_ECC_SIZE_CONFIG[19:12] ECCSIZE0 and [29:22] ECCSIZE1	x (1)
Select the size of each of the 9 result registers (size specified by ECCSIZE0 or ECCSIZE1).	GPMC_ECC_SIZE_CONFIG[j-1] ECCjRESULTSIZE where j = 1 to 9	x
Select the chip-select where ECC is computed.	GPMC_ECC_CONFIG[3:1] ECCCS	x
Select the Hamming code or BCH code ECC algorithm in use.	GPMC_ECC_CONFIG[16] ECCALGORITHM	x
Select word size for ECC calculation.	GPMC_ECC_CONFIG[7] ECC16B	x
If the BCH code is used, Set an error correction capability and Select a number of sectors to process.	GPMC_ECC_CONFIG[13:12] ECCBCHTSEL and GPMC_ECC_CONFIG[6:4] ECCTOPSECTOR	x
Enable the ECC computation.	GPMC_ECC_CONFIG[0] ECCENABLE	0x1

(1) Depends on the size of each sector in the NAND page
(2) This parameter depends on the numbers of sectors in a page.

Table 15-451. Prefetch and Write-Posting Engine

Subprocess Name	Register/Bit Field	Value
Disable the engine before configuration.	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0x0
Select the chip-select associated with a NAND device where the prefetch engine is active.	GPMC_PREFETCH_CONFIG1[26:24] ENGINECSSELECTOR	x
Select access direction through prefetch engine, read or write.	GPMC_PREFETCH_CONFIG1[0] ACCESSMODE	x
Select the threshold used to issue a DMA request.	GPMC_PREFETCH_CONFIG1[14:8] FIFOTHRESHOLD	x
Select DMA synchronized mode or software manual mode.	GPMC_PREFETCH_CONFIG1[2] DMAMODE	x
Select if the engine immediately starts accessing the memory upon STARTENGINE assertion or if hardware synchronization based on a WAIT signal is used.	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode.	GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR	x
Enter a number of clock cycles removed to timing parameters (for all back-to-back accesses to the NAND flash except the first one).	GPMC_PREFETCH_CONFIG1[30:28] CYCLEOPTIMIZATION	x
Enable the prefetch postwrite engine.	GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE	0x1
Select the number of bytes to be read or written by the engine to the selected chip-select.	GPMC_PREFETCH_CONFIG2[13:0] TRANSFERCOUNT	x
Start the prefetch engine.	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0x1

Table 15-452. Wait Pin Configuration

Subprocess Name	Register/Bit Field	Value
Selects when the engine starts the access to chip-select.	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode.	GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR	x

Table 15-453. Enable Chip-Select

Subprocess Name	Register/Bit Field	Value
When all parameters are configured, enable the chip-select.	GPMC_CONFIG7_i[6] CSVALID	x

15.4.5.5 Set Memory Access

This section describes the bit field to configure to set the GPMC in various memory modes. Table 15-454 and Table 15-455 provide check lists for mode parameters and access type parameters, respectively.

Table 15-454. Mode Parameters Check List

Register	Bit	Name	Asynchronous				Synchronous			
			Single Read Access	Single Write Access	Multiple Read (Page) Access	Multiple Write (Page) Access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access
GPMC_CONFIG1_i	30	READMULTIPLE	0x0	–	0x1 [(1)]	N/S	0x0	–	0x1	–
GPMC_CONFIG1_i	29	READTYPE	0x0	–	0x0 [(1)]	N/S	0x1	–	0x1	–
GPMC_CONFIG1_i	28	WRITEMULTIPLE	–	0x0	- [(1)]	N/S	–	0x0	–	0x1
GPMC_CONFIG1_i	27	WRITETYPE	–	0x0	- [(1)]	N/S	–	0x1	–	0x1

(1) Multiple read is not supported in address/data-multiplexed and AAD-multiplexed modes. Multiple read is supported in nonmultiplexed mode.

Table 15-455. Access Type Parameters Check List

Register	Bit	Name	Access Type		
			Nonmultiplexed	Address/ Data-Multiplexed	AAD-Multiplexed
GPMC_CONFIG1_i	9:8	MUXADDDATA	0x0	0x2	0x1

15.4.5.6 GPMC Timing Parameters

Figure 15-96 shows a programming model diagram for the NOR interfacing timing parameters.

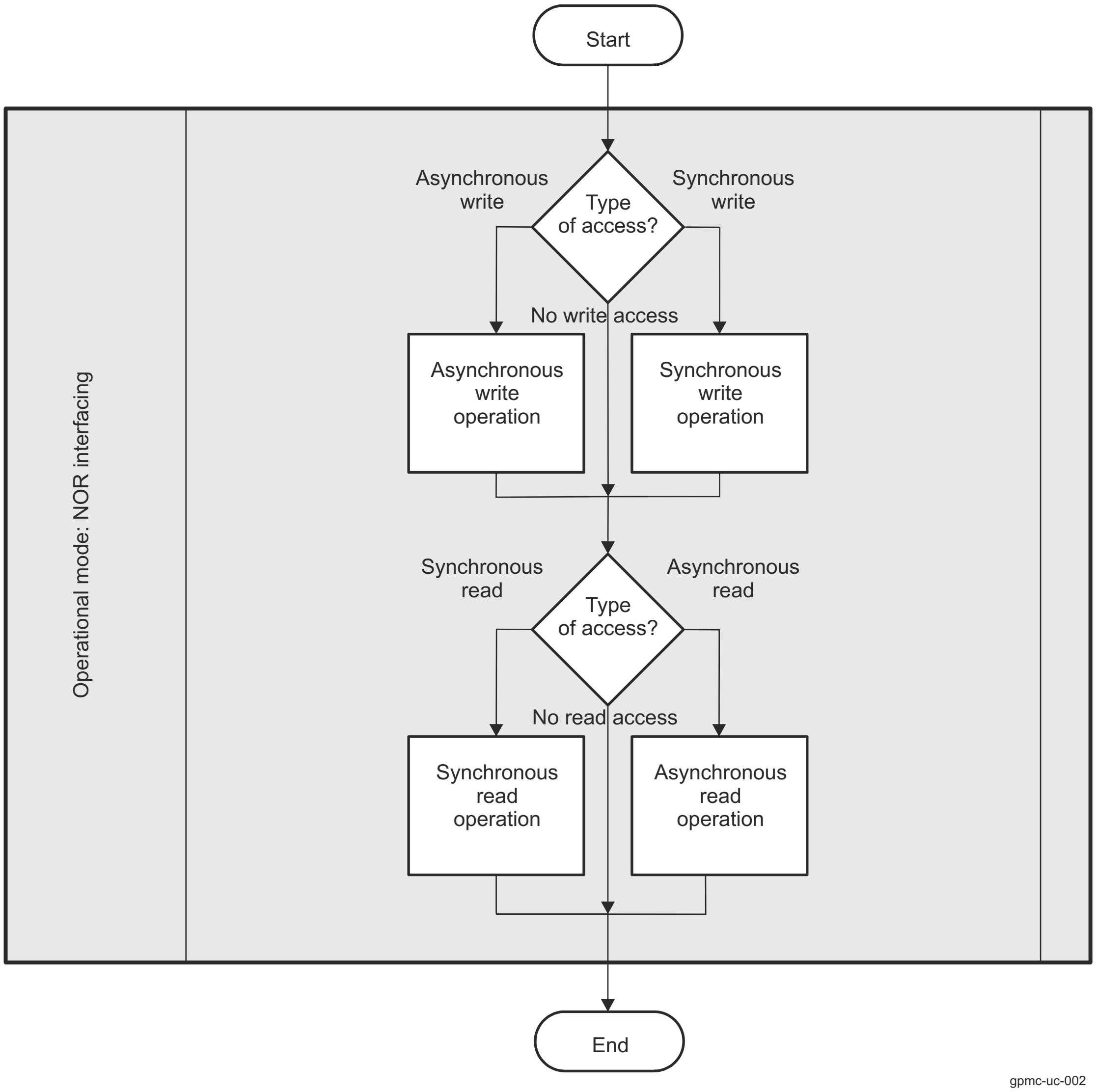


Figure 15-96. NOR Interfacing Timing Parameters Diagram

Table 15-456 lists the bit fields to configure adequate timing parameters in various memory modes.

Table 15-456. Timing Parameters

			Asynchronous			Synchronous				Access Type		
Register	Bit	Name	Single Read Access	Single Write Access	Multiple Read (Page) access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access	Non-multiplexed	Address/Data-Multiplexed	AAD Multiplexed
GPMC_CONFIG1_i	9	MUXADDDATA	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	29	READTYPE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	30	READMULTIPLE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	27	WRITETYPE		y			y		y	y	y	y
GPMC_CONFIG1_i	28	WRITEMULTIPLE		y			y		y	y	y	y
GPMC_CONFIG1_i	31	WRAPBURST						y	y	y	y	y

Table 15-456. Timing Parameters (continued)

			Asynchronous			Synchronous				Access Type		
GPMC_CONFIG1_i	26:25	CLKACTIVATIONTIME				y	y	y	y	y	y	y
GPMC_CONFIG1_i	19:18	WAITMONITORINGTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	4	TIMEPARAGRANULARITY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	20:16	CSWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG2_i	12:8	CSRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG2_i	7	CSEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	3:0	CSONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	30:28	ADVAADMUXWROFFTIME		y			y		y			y
GPMC_CONFIG3_i	30:29	ADVAADMUXRDOFFTIME	y		y	y		y				y
GPMC_CONFIG3_i	6:4	ADVAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG3_i	20:16	ADVWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG3_i	12:8	ADVRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG3_i	7	ADVEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	3:0	ADVONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG4_i	15:13	OEAADMUXOFFTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	6:4	OEAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	28:24	WEOFFTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	23	WEEXTRADELAY		y			y		y	y	y	y
GPMC_CONFIG4_i	19:16	WEONTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	12:8	OEOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG4_i	7	OEEXTRADELAY	y		y	y		y		y	y	y
GPMC_CONFIG4_i	3:0	OEONTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_i	27:24	PAGEBURSTACCESSTIME			y			y	y	y	y	y
GPMC_CONFIG5_i	20:16	RDACCESSTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_i	12:8	WRCYCLETIME		y			y		y	y	y	y
GPMC_CONFIG5_i	4:0	RDCYCLETIME	y		y	y		y		y	y	y
GPMC_CONFIG6_i	28:24	WRACCESSTIME		y			y		y	y	y	y
GPMC_CONFIG6_i	19:16	WRDATAONADMUXBUS		y			y		y		y	y
GPMC_CONFIG6_i	11:8	CYCLE2CYCLEDELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	7	CYCLE2CYCLESAMECSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	6	CYCLE2CYCLEDIFFCSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	3:0	BUSTURNAROUND	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG7_i	6	CSVALID	y	y	y	y	y	y	y	y	y	y

15.4.5.6.1 GPMC Timing Parameters Formulas

This section is intended to help the user calculate the GPMC timing bit field values. Formulas are not listed exhaustively.

The section describes:

- NAND flash interface timing parameters formulas
- Synchronous NOR flash timing parameters formulas
- Asynchronous NOR flash timing parameters formulas

For complete information, such as OPP and board effects on timings, see the device data manual.

15.4.5.6.1.1 NAND Flash Interface Timing Parameters Formulas

This section lists formulas to calculate NAND timing parameters. This is the case when GPMC_CONFIG1_i[11:10] DEVICETYPE = 0x2. Table 15-457 describes the NAND timing parameters.

Table 15-457. NAND Formulas Description

Configuration Parameter	Unit	Description
A	ns	Pulse duration – gpmc_wen valid time
B	ns	Delay time – gpmc_cs valid to gpmc_wen valid
C	ns	Delay time – gpmc_ben0/gpmc_advn_ale high to gpmc_wen valid
D	ns	Delay time – gpmc_ad[15:0] valid to gpmc_wen valid
E	ns	Delay time – gpmc_wen invalid to gpmc_ad[15:0] invalid
F	ns	Delay time – gpmc_wen invalid to gpmc_ben0/gpmc_advn_ale invalid
G	ns	Delay time – gpmc_wen invalid to gpmc_cs invalid
H	ns	Cycle time – Write cycle time
I	ns	Delay time – gpmc_cs valid to gpmc_oen_ren valid
J	ns	Setup time – gpmc_ad[15:0] valid to gpmc_oen_ren invalid
K	ns	Pulse duration – gpmc_oen_ren valid time
L	ns	Cycle time – Read cycle time
M	ns	Delay time – gpmc_oen_ren invalid to gpmc_cs invalid

The configuration parameters are calculated through the following formulas. For more information, see the device data manual.

A = (WEOffTime – WEOnTime) * (TimeParaGranularity + 1) * GPMC_FCLK period

B = ((WEOnTime – CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEExtraDelay – CSExtraDelay)) * GPMC_FCLK period

C = ((WEOnTime – ADVOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEExtraDelay - ADVExtraDelay)) * GPMC_FCLK period

D = (WEOnTime * (TimeParaGranularity + 1) + 0.5 * WEExtraDelay) * GPMC_FCLK period

E = (WrCycleTime – WEOffTime * (TimeParaGranularity + 1) – 0.5 * WEExtraDelay) * GPMC_FCLK period

F = (ADVWrOffTime – WEOffTime * (TimeParaGranularity + 1) + 0.5 * (ADVExtraDelay – WEExtraDelay) * GPMC_FCLK period

G = (CSWrOffTime – WEOffTime * (TimeParaGranularity + 1) + 0.5 * (CSExtraDelay – WEExtraDelay) * GPMC_FCLK period

H = WrCycleTime * (1 + TimeParaGranularity) * GPMC_FCLK period

I = ((OEOnTime – CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (OEExtraDelay – CSExtraDelay)) * GPMC_FCLK period

J = ((RdAccessTime – OEOffTime) * (TimeParaGranularity + 1) – 0.5 * OEExtraDelay)) * GPMC_FCLK period

K = (OEOffTime – OEOnTime) * (1 + TimeParaGranularity) * GPMC_FCLK period

L = RdCycleTime * (1 + TimeParaGranularity) * GPMC_FCLK period

M = (CSRdOffTime – OEOffTime * (TimeParaGranularity + 1) + 0.5 * (CSExtraDelay - OEExtraDelay) * GPMC_FCLK period

Figure 15-97 shows a simplified example of command latch cycle timing where formulas are associated with signal waves.

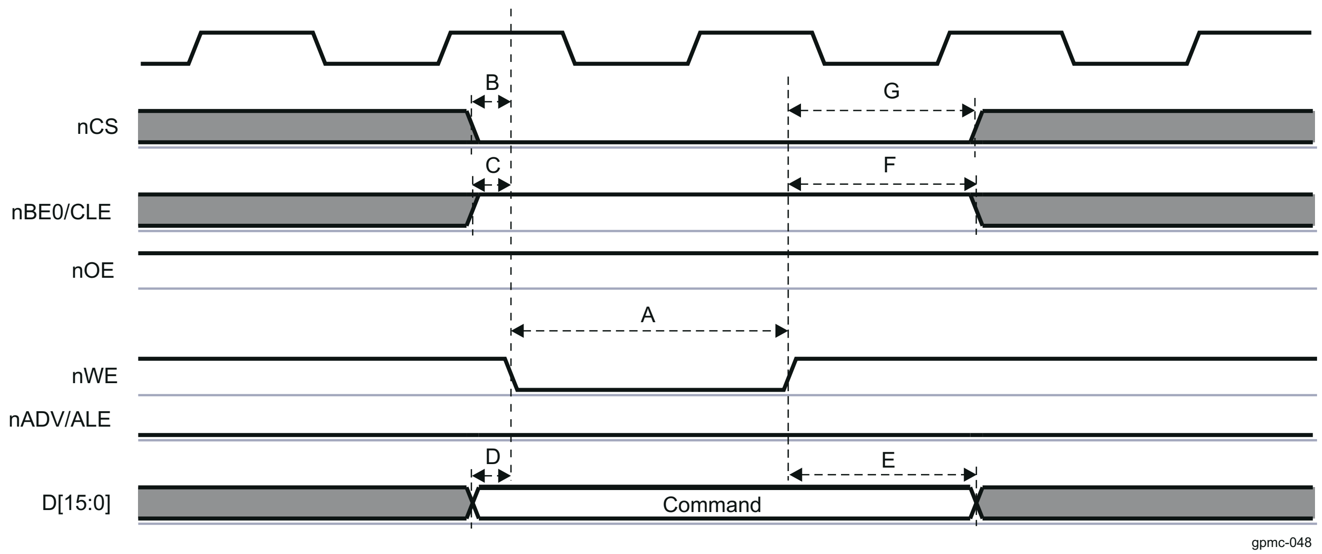


Figure 15-97. NAND Command Latch Cycle Timing Simplified Example

15.4.5.6.1.2 Synchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to calculate synchronous NOR timing parameters. This is the case when GPMC_CONFIG1_i[11:10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to synchronous mode. Table 15-458 describes the synchronous NOR formulas.

Table 15-458. Synchronous NOR Formulas Description

Configuration Parameter	Unit	Description
A	ns	Pulse duration – gpmc_cs low
B	ns	Delay time – address bus valid to gpmc_clk first edge
		Delay time – gpmc_ben0/gpmc_ben1 valid to gpmc_clk first edge
C	ns	Pulse duration – gpmc_ben0/gpmc_ben1 low
D	ns	Delay time – gpmc_clk rising edge to gpmc_ben0/gpmc_ben1 invalid
		Delay time – gpmc_clk rising edge to gpmc_advn_ale invalid
E	ns	Delay time – gpmc_clk rising edge to gpmc_cs invalid
		Delay time – gpmc_clk rising edge to gpmc_oen_ren invalid
F	ns	Delay time – gpmc_clk rising edge to gpmc_cs transition
G	ns	Delay time – gpmc_clk rising edge to gpmc_advn_ale transition
H	ns	Delay time – gpmc_clk rising edge to gpmc_oen_ren transition
I	ns	Delay time – gpmc_clk rising edge to gpmc_wen transition
J	ns	Delay time – gpmc_clk rising edge to gpmc_ad data bus transition
		Delay time – gpmc_clk rising edge to gpmc_ben0/gpmc_ben1 transition
K	ns	Pulse duration – gpmc_advn_ale low
L	ns	Delay time – gpmc_wait invalid to first data latching gpmc_clk edge

The configuration parameters are calculated through the following formulas. For more information, see the device data manual.

1. For single read accesses:
   A = (CSRDOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   C = RDCYCLETIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   D = (RDCYCLETIME – RDACCESSTIME) * GPMC_FCLK period
   E = (CSRDOFFTIME – RDACCESSTIME) * GPMC_FCLK period

2. For burst read accesses (where n is the page burst access number):
   A = (CSRDOFFTIME – CSONTIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   C = (RDCYCLETIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   D = (RDCYCLETIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period
   E = (CSRDOFFTIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period
3. For burst write accesses (where n is the page burst access number):
   A = (CSWROFFTIME – CSONTIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   C = (WRCYCLETIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   D = (WRCYCLETIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period
   E = (CSWROFFTIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period
4. For all accesses:
   For nCS falling edge (chip-select activated):
   - Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
   - Case where GPMCFCLKDIVIDER = 0x1:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSONTIME are odd) or (CLKACTIVATIONTIME and CSONTIME are even)
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise.
   - Case where GPMCFCLKDIVIDER = 0x2:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME) is a multiple of 3
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
     F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
   - Case where GPMCFCLKDIVIDER = 0x3:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME) is a multiple of 4
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
     F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
     F = (3 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

   For nCS rising edge (chip-select deactivated) in reading mode:
   - Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
   - Case where GPMCFCLKDIVIDER = 0x1:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSRDOFFTIME are odd) or (CLKACTIVATIONTIME and CSRDOFFTIME are even)
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise.
   - Case where GPMCFCLKDIVIDER = 0x2:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME) is a multiple of 3
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
     F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
   - Case where GPMCFCLKDIVIDER = 0x3:

F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
F = (3 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nCS rising edge (chip-select deactivated) in writing mode:

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSWROFFTIME are odd) or (CLKACTIVATIONTIME and CSWROFFTIME are even)
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  F = (3 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nADV falling edge (nADV activated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVONTIME are odd) or (CLKACTIVATIONTIME and ADVONTIME are even)
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME) is a multiple of 3
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME) is a multiple of 4
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4

G = (3 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nADV rising edge (nADV deactivated) in reading mode:

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVRDOFFTIME are odd) or (CLKACTIVATIONTIME and ADVRDOFFTIME are even)
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  G = (3 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nADV rising edge (nADV deactivated) in writing mode:

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVWROFFTIME are odd) or (CLKACTIVATIONTIME and ADVWROFFTIME are even)
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  G = (3 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nOE falling edge (nOE activated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:

H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and OEONTIME are odd) or (CLKACTIVATIONTIME and OEONTIME are even)
H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period otherwise.

- Case where GPMCFCLKDIVIDER = 0x2:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME) is a multiple of 3
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME) is a multiple of 4
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  H = (3 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nOE rising edge (nOE deactivated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and OEOFFTIME are odd) or (CLKACTIVATIONTIME and OEOFFTIME are even)
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  H = (3 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nWE falling edge (nWE activated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and WEONTIME are odd) or (CLKACTIVATIONTIME and WEONTIME are even)
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME) is a multiple of 3

I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME) is a multiple of 4
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  I = (3 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nWE rising edge (nWE deactivated):
- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and WEOFFTIME are odd) or (CLKACTIVATIONTIME and WEOFFTIME are even)
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  I = (3 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For gpmc_nadv low pulse duration:
- Read operation:
  K = (ADVRDOFFTIME – ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
- Write operation:
  K = (ADVWROFFTIME – ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

For gpmc_wait invalid to first data latching gpmc_clk edge:
- L = WAITMONITORINGTIME * (GPMCFCLKDIVIDER + 1) * GPMC_FCLK period + GPMC_CLK period

Figure 15-98 shows a simplified example of a synchronous NOR single read where formulas are associated with signal waves.

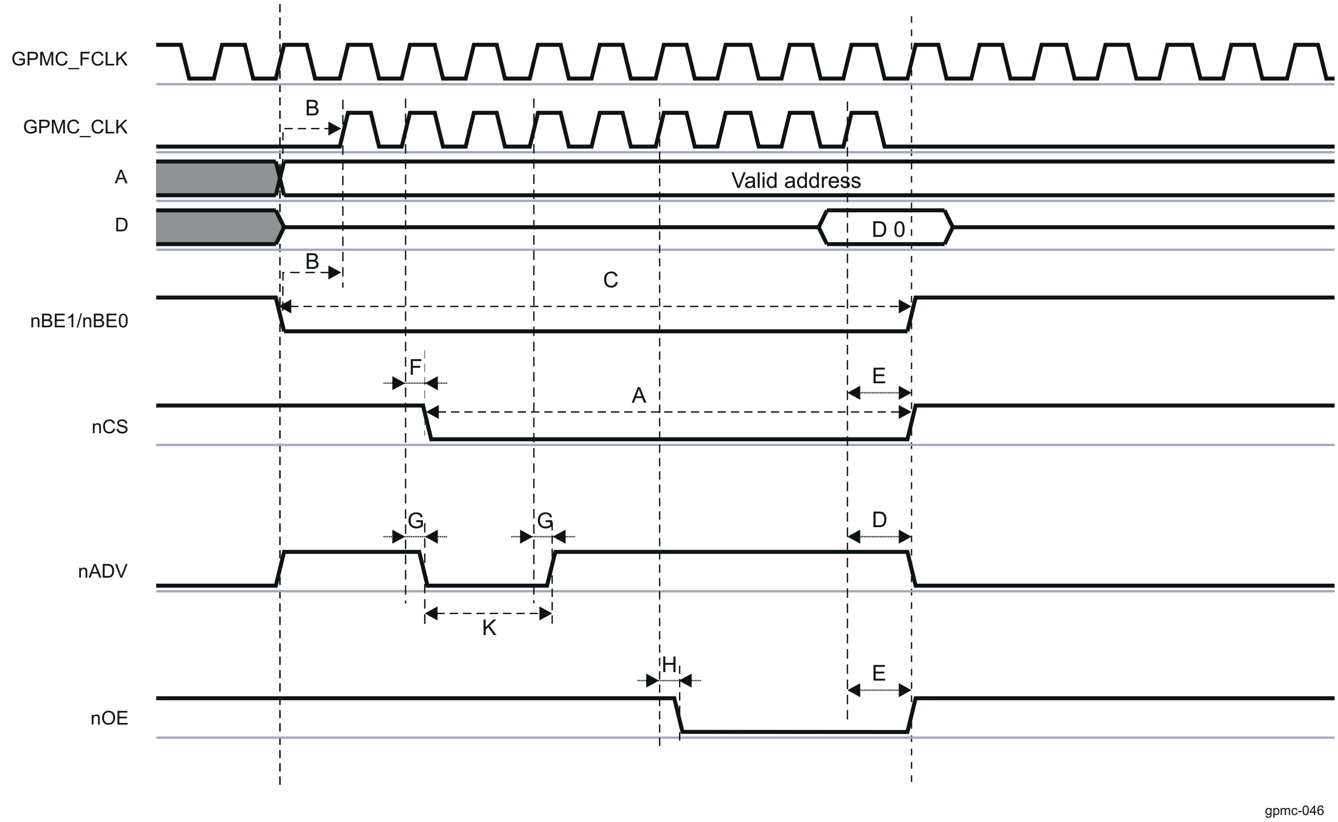


Figure 15-98. Synchronous NOR Single Read Simplified Example

15.4.5.6.1.3 Asynchronous NOR Flash Timing Parameters Formulas

This section lists all the formulas to calculate asynchronous NOR timing parameters. This is the case when GPMC_CONFIG1_i[11:10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to asynchronous mode. Table 15-459 describes the asynchronous NOR formulas.

Table 15-459. Asynchronous NOR Formulas Description

Configuration Parameter	Unit	Description
A	ns	Pulse duration – gpmc_cs low
B	ns	Delay time – gpmc_cs valid to gpmc_advn_ale invalid
C	ns	Delay time – gpmc_cs valid to gpmc_oen_ren invalid (single read)
D	ns	Pulse duration – address bus valid - 2nd, 3rd and 4th accesses
E	ns	Delay time – gpmc_cs valid to gpmc_wen valid
F	ns	Delay time – gpmc_cs valid to gpmc_wen invalid
G	ns	Address invalid duration between two successive R/W accesses
H	ns	Setup time – read data valid before gpmc_oen_ren high
I	ns	Delay time – gpmc_cs valid to gpmc_oen_ren invalid (burst read)
J	ns	Delay time – address bus valid to gpmc_cs valid
		Delay time – data bus valid to gpmc_cs valid
		Delay time – gpmc_ben0/gpmc_ben1 valid to gpmc_cs valid
K	ns	Delay time – gpmc_cs valid to gpmc_advn_ale valid
L	ns	Delay time – gpmc_cs valid to gpmc_oen_ren valid
M	ns	Delay time – gpmc_cs valid to first data latching edge
N	ns	Pulse duration – gpmc_ben0/gpmc_ben1 valid time

Table 15-459. Asynchronous NOR Formulas Description (continued)

Configuration Parameter	Unit	Description
O	ns	Delay time – gpmc_cs valid to gpmc_advn_ale valid

The configuration parameters are calculated through the following formulas. These formulas are not exhaustive. For more information, see the device data manual.

- gpmc_cs low pulse:
  For single read: A = (CSRDOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
  For burst read: A = (CSRDOFFTIME – CSONTIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
  For single write: A = (CSWROFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
  For burst write: A = (CSWROFFTIME – CSONTIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
- gpmc_cs valid to gpmc_advn_ale invalid delay:
  For reading: B = ((ADVRDOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
  For writing: B = ((ADVWROFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- C = ((OEOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- D = PAGEBURSTACCESSTIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
- E = ((WEONTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (WEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- F = ((WEOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (WEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- G = CYCLE2CYCLEDELAY * GPMC_FCLK period
- H = ((OEOFFTIME – RDACCESSTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * OEEXTRADELAY) * GPMC_FCLK period
- I = ((OEOFFTIME + (N – 1) * PAGEBURSTACCESSTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period, where N = page burst access number
- J = (CSONTIME * (TIMEPARAGRANULARITY + 1) + 0.5 * CSEXTRADELAY) * GPMC_FCLK period
- K = ((ADVONTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- L = ((OEONTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- M = ((RDACCESSTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) – 0.5 * CSEXTRADELAY) * GPMC_FCLK period
- gpmc_ben0/gpmc_ben1 pulse:
  For single read: N = RDCYCLETIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
  For burst read: N = (RDCYCLETIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
  For burst write: N = (WRCYCLETIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
- O = ((WRCYCLETIME + (N – 1) * PAGEBURSTACCESSTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period

Figure 15-99 shows a simplified example of an asynchronous NOR single write where formulas are associated with signal waves.

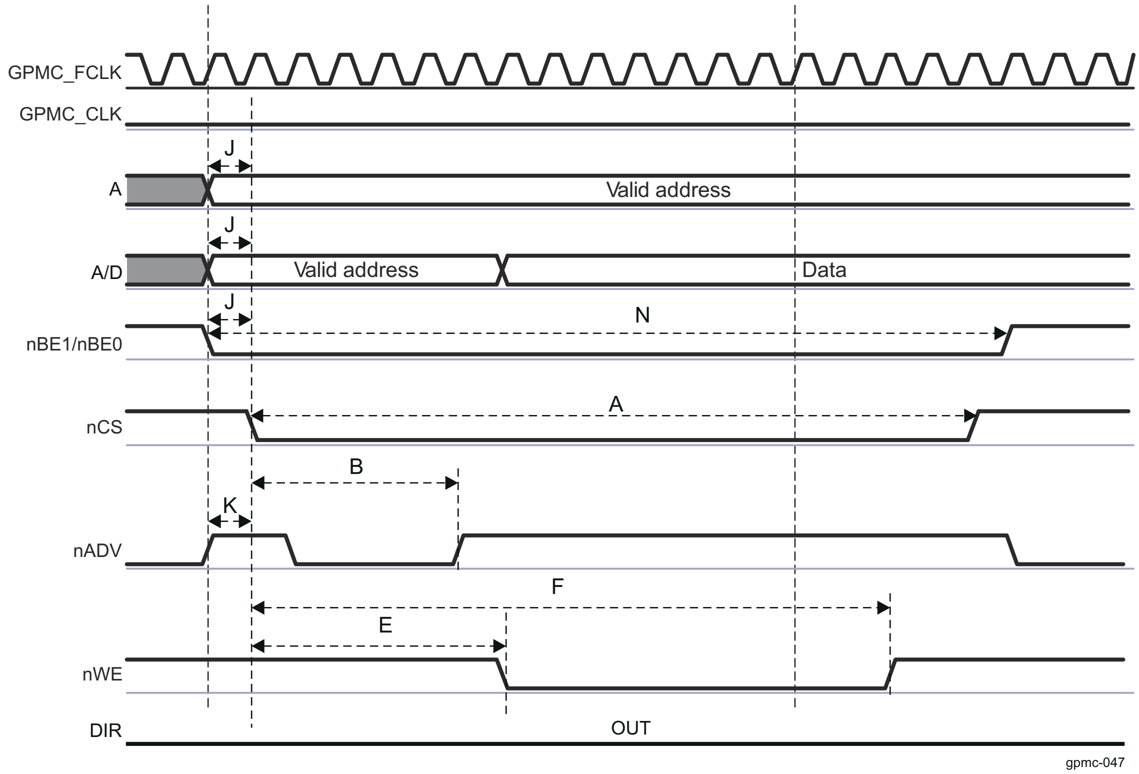


Figure 15-99. Asynchronous NOR Single Write Simplified Example

---

Note

Write multiple access is not supported in asynchronous mode. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

15.4.6 GPMC Use Cases and Tips

15.4.6.1 How to Set GPMC Timing Parameters for Typical Accesses

15.4.6.1.1 External Memory Attached to the GPMC Module

As discussed in the introduction to this chapter, the GPMC module supports the following external memory types:

- Asynchronous or synchronous, 8- or 16-bit-wide memory or device
- 16-bit address/data-multiplexed or not multiplexed NOR flash device
- 8- or 16-bit NAND flash device

The following examples describe how to calculate GPMC timing parameters by showing a typical parameter setup for the access to be performed.

The example is based on a 512-Mb multiplexed NOR flash memory with the following characteristics:

- Type: NOR flash (address/data-multiplexed mode)
- Size: 512M bits
- Data Bus: 16 bits wide
- Speed: 104-MHz clock frequency
- Read access time: 80 ns

15.4.6.1.2 Typical GPMC Setup

Table 15-460 lists some of the I/Os of the GPMC module.

Table 15-460. GPMC Signals

Signal Name	I/0	Description
GPMC_FCLK	Internal	Functional and interface clock. Acts as the time reference.
gpmc_clk	O	External clock provided to the external device for synchronous operations
gpmc_a[26:16]	O	Address
gpmc_ad[15:0]	I/O	Data-multiplexed with addresses A[16:1] on memory side
gpmc_csx	O	Chip-select (where x = 0, or 1)
gpmc_advn_ale	O	Address valid enable
gpmc_oen_ren	O	Output enable (read access only)
gpmc_wen	O	Write enable (write access only)
gpmc_wait[1:0]	I	Ready signal from memory device. Indicates when valid burst data is ready to be read

Figure 15-100 shows the typical connection between the GPMC module and an attached NOR Flash memory.

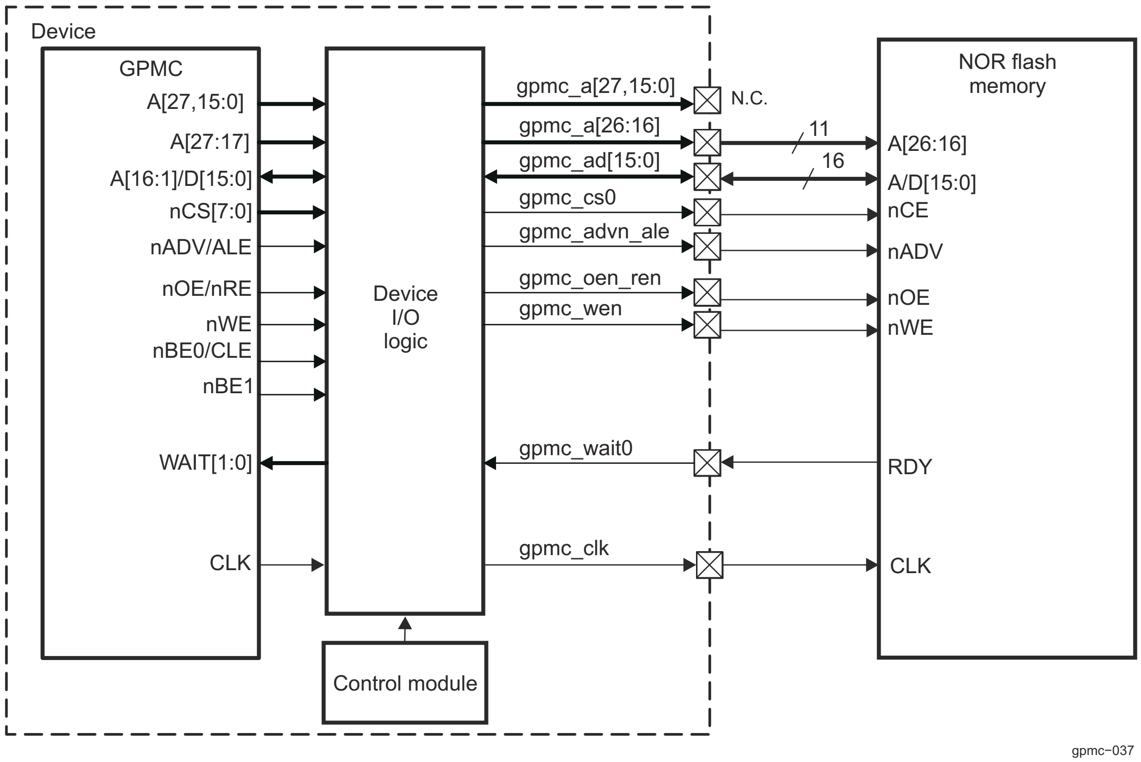


Figure 15-100. GPMC Connection to an External NOR Flash Memory

The following sections demonstrate how to calculate GPMC parameters for three access types:

- Synchronous burst read
- Asynchronous read
- Asynchronous single write

15.4.6.1.2.1 GPMC Configuration for Synchronous Burst Read Access

---

Note

The examples in Section 15.4.6.1.2.1 through Section 15.4.6.1.2.3 are based on a clock rate of 104 MHz. Refer to the device Data Manual for the maximum frequency appropriate for this device and to the memory datasheet for the maximum frequency for the particular memory device.

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 15-461 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 15-462 shows how to calculate timings for the GPMC using the memory parameters.

Figure 15-101 shows the synchronous burst read access.

Table 15-461. Useful Timing Parameters on the Memory Side

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCES	nCS setup time to clock	0
tACS	Address setup time to clock	3
tIACC	Synchronous access time	80
tBACC	Burst access time valid clock to output delay	5,2
tCEZ	Chip-select to High-Z	7

Table 15-461. Useful Timing Parameters on the Memory Side (continued)

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tOEZ	Output enable to High-Z	7
tAVC	nADV setup time	6
tAVD	nAVD pulse	6
tACH	Address hold time from clock	3

The following terms, which describe the timing interface between the controller and its attached device, are used to calculate the timing parameters on the GPMC side:

- Read access time (GPMC side): Time required to activate the clock + read access time requested on the memory side + data setup time required for optimal capture of a burst of data
- Data setup time (GPMC side): Ensures a good capture of a burst of data (as opposed to taking a burst of data out). One word of data is processed in one clock cycle (T = 9,615 ns). The read access time between two bursts of data is tBACC = 5.2 ns. Therefore, data setup time is a clock period – tBACC = 4,415 ns of data setup.
- Access completion (GPMC side): (Different from page burst access time) Time required between the last burst access and access completion: nCS/nOE hold time (nCS and nOE must be released at the end of an access. These signals are held to allow the access to complete).
- Read cycle time (GPMC side): Read access time + access completion
- Write cycle time for burst access: Not supported for NOR flash memory

Table 15-462. Calculating GPMC Timing Parameters

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
GPMC FCLK Divider	–	–	–	GPMCFCLKDIVIDER = 0x0
ClkActivation Time	min (tCES, tACS)	3	1	CLKACTIVATIONTIME = 0x1
RdAccessTime	roundmax (ClkActivationTime + tIACC + DataSetupTime)	94.03: (9,615 + 80 + 4,415)	10: roundmax (94.03 / 9,615)	RDACCESSTIME = 0x0A
PageBurst RdAccessTime	roundmax (tBACC)	roundmax (5.2)	1	PAGEBURSTACCESSTIME = 0x1
RdCycleTime	RdAccess time + max (tCEZ, tOEZ)	101.03: (94.03 + 7)	11	RDCYCLETIME = 0x0B
CsOnTime	tCES	0	0	CSONTIME = 0x0
CsReadOffTime	RdCycleTime	-	11	CSRDOFFTIME = 0x0B
AdvOnTime	tAVC (1)	0	0	ADVONTIME = 0x0
AdvRdOffTime	tAVD + tAVC (2)	12	2	ADVRDOFFTIME = 0x02
OeOnTime (3)	(ClkActivationTime + tACH) < OeOnTime (ClkActivationTime + tIACC)	–	3, for instance	OEONTIME = 0x3
OeOffTime	RdCycleTime	–	11	OEOFFTIME = 0x0B

(1) The external clock provided to the NOR flash is not yet available.
(2) AdvRdOffTime – AdvOnTime = tAVD; thus, AdvRdOffTime = tAVD + AdvOnTime = tAVD + tAVC.
(3) OeOnTime must ensure that addresses are available. It must not exceed the availability of the first burst of data.

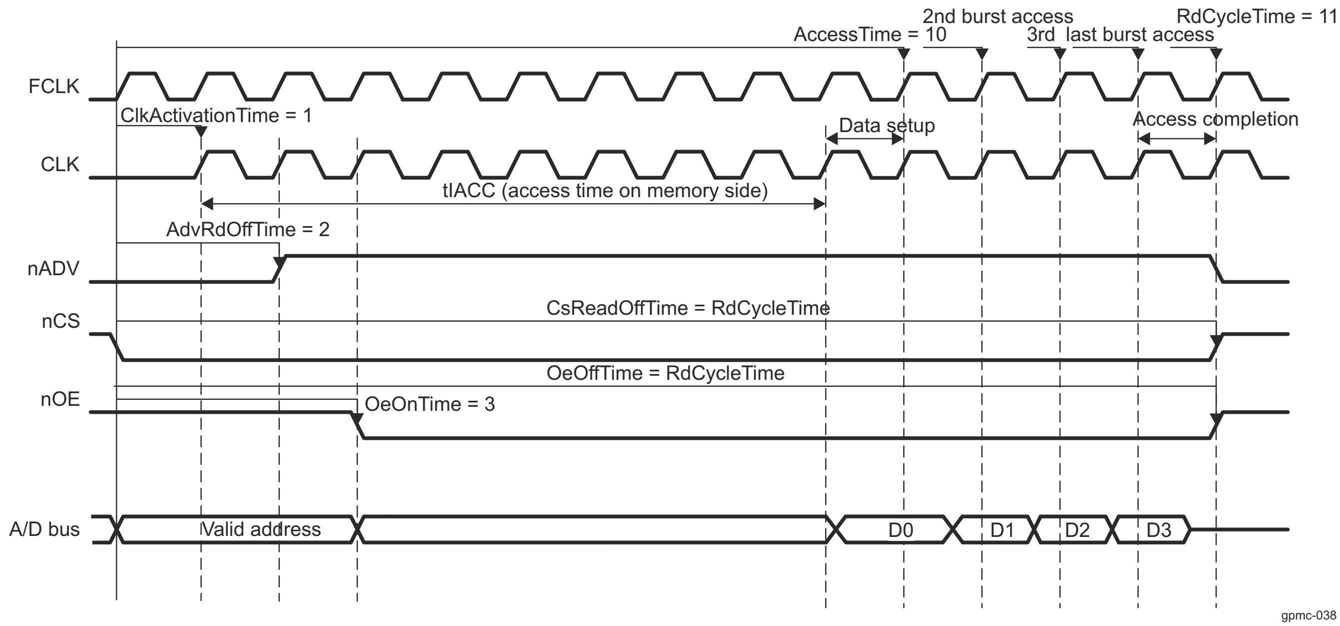


Figure 15-101. Synchronous Burst Read Access (Timing Parameters in Clock Cycles)

15.4.6.1.2.2 GPMC Configuration for Asynchronous Read Access

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 15-463 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 15-464 shows how to calculate timings for the GPMC using the memory parameters.

Figure 15-102 shows the asynchronous read access.

Table 15-463. AC Characteristics for Asynchronous Read Access

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCE	Read Access time from nCS low	80
tAAVDS	Address setup time to rising edge of nADV	3
tAVDP	nADV low time	6
tCAS	nCS setup time to nADV	0
tOE	Output enable to output valid	6
tOEZ	Output enable to High-Z	7

Use the following formula to calculate the RdCycleTime parameter for this typical access:

RdCycleTime = RdAccessTime + AccessCompletion = RdAccessTime + 1 clock cycle + tOEZ:

1. On the memory side, the external memory makes the data available to the output bus. This is the memory-side read access time defined in Table 15-463: the number of clock cycles between the address capture (nADV rising edge) and the data valid on the output bus.

   The GPMC requires some hold time to allow the data to be captured correctly and the access to be finished.
2. To read the data correctly, the GPMC must be configured to meet the data setup time requirement of the memory; the GPMC module captures the data on the next rising edge. This is access time on the GPMC side.
3. There must also be a data hold time for correctly reading the data (checking that there is no nOE/nCS deassertion while reading the data). This data hold time is one clock cycle (that is, RdAccessTime + 1).

4. To complete the access, nOE/nCS signals are driven to High-Z. RdAccessTime + 1 + tOEZ is the read cycle time.
5. Addresses can now be relatched and a new read cycle begun.

Table 15-464. GPMC Timing Parameters for Asynchronous Read Access

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
ClkActivationTime	n/a (asynchronous mode)			
RdAccessTime	round max (tCE)	80	10	RDACCESSTIME = 0x0A
PageBurstAccessTime	N/A (single access)			
RdCycleTime	RdAccessTime + 1cycle + tOEZ	96, 615	12	RDCYCLETIME = 0x0C
CsOnTime	tCAS	0	0	CSONTIME = 0x0
CsReadOffTime	RdAccessTime + 1 cycle	89, 615	11	CSRDOFFTIME = 0x0B
AdvOnTime	tAAVDS	3	1	ADVONTIME = 0X1
AdvRdOffTime	tAAVDS + tAVDP	9	1	ADVRDOFFTIME = 0x01
OeOnTime	OeOnTime >= AdvRdOffTime (multiplexed mode)	-	3, for instance	OEONTIME = 0x3
OeOffTime	RdAccessTime + 1cycle	89, 615	11	OEOFFTIME = 0x0B

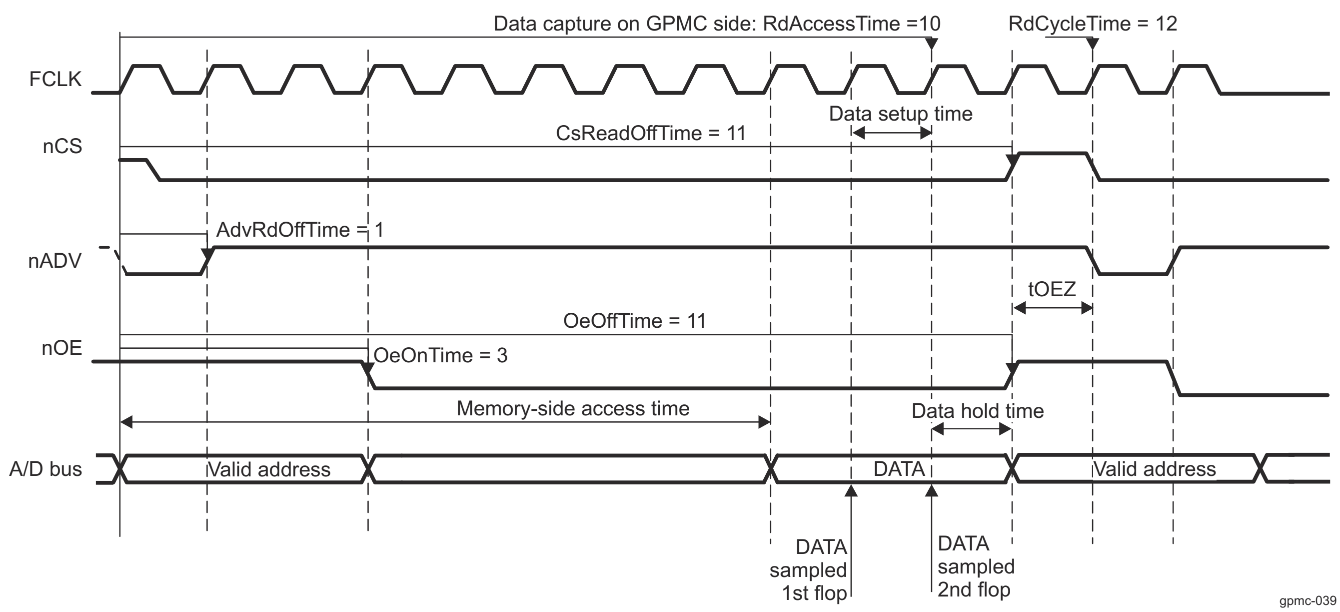


Figure 15-102. Asynchronous Single Read Access (Timing Parameters in Clock Cycles)

15.4.6.1.2.3 GPMC Configuration for Asynchronous Single Write Access

The clock runs at 104 MHz: (f = 104 MHz; T = 9, 615 ns).

Table 15-466 shows how to calculate timings for the GPMC using the memory parameters.

Table 15-465 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Figure 15-103 shows the synchronous burst write access.

Table 15-465. AC Characteristics for Asynchronous Single Write (Memory Side)

AC Characteristics on the Memory Side	Description	Duration (ns)
tWC	Write cycle time	60

Table 15-465. AC Characteristics for Asynchronous Single Write (Memory Side) (continued)

AC Characteristics on the Memory Side	Description	Duration (ns)
tAVDP	nADV low time	6
tWP	Write pulse width	25
tWPH	Write pulse width high	20
tCS	nCS setup time to nWE	3
tCAS	nCS setup time to nADV	0
tAVSC	nADV setup time	3

For asynchronous single write access, write cycle time is WrCycleTime = WeOffTime + AccessCompletion = WeOffTime + 1. For the AccesCompletion, the GPMC requires one cycle of data hold time (nCS deassertion). For more information, see the device data manual.

Table 15-466. GPMC Timing Parameters for Asynchronous Single Write

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Registers Configuration
ClkActivationTime	N/A (asynchronous mode)			
WdAccessTime	Applicable only to WAITMONITORING (the value is the same as for read access)			
PageBurstAccessTime	N/A (single access)			
WrCycleTime	WeOffTime + AccessCompletion	57, 615	6	WRCYCLETIME = 0x06
CsOnTime	tCAS	0	0	CSONTIME = 0x0
CsWrOffTime	WeOffTime + 1	57, 615	6	CSWROFFTIME = 0x06
AdvOnTime	tAVSC	3	1	ADVONTIME = 0x1
AdvWrOffTime	tAVSC + tAVDP	9	1	ADVWROFFTIME = 0x01
WeOnTime	tCS	3	1	WEONTIME = 0x1
WeOffTime	tCS + tWP + tWPH	48	5	WEOFFTIME = 0x05

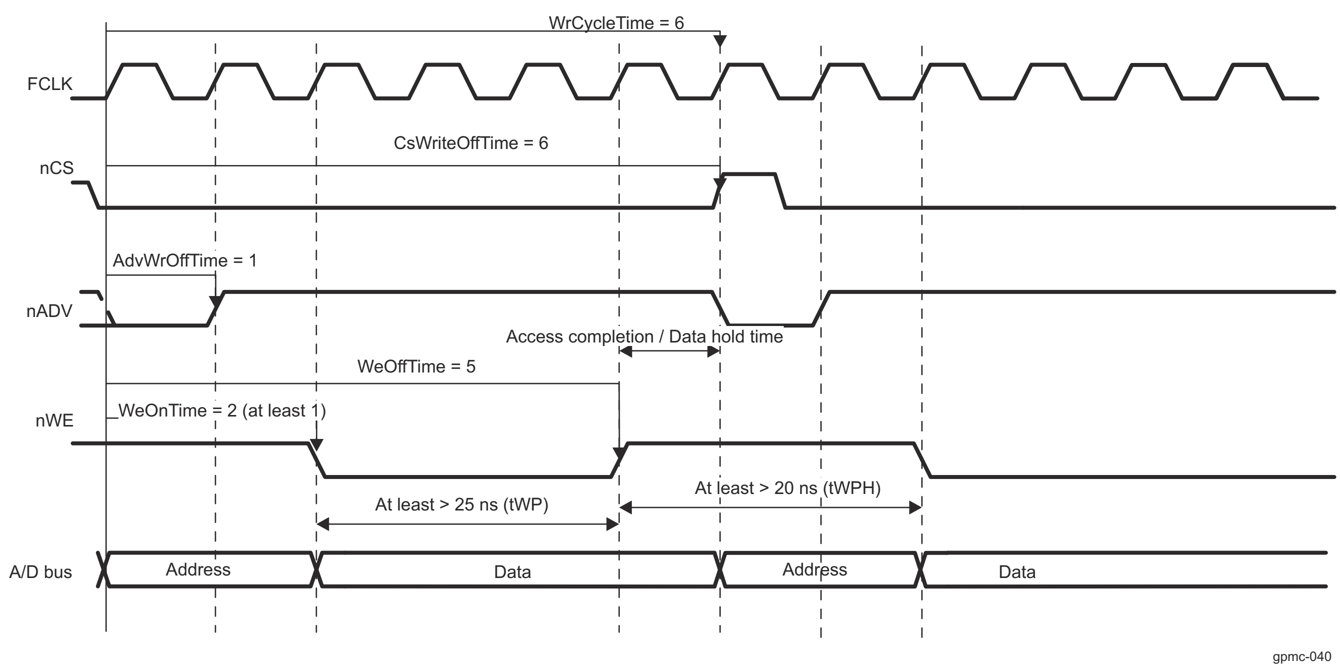


Figure 15-103. Asynchronous Single Write Access (Timing Parameters in Clock Cycles)

15.4.6.2 How to Choose a Suitable Memory to Use With the GPMC

This section is intended to help the user select a suitable memory device to interface with the GPMC controller.

15.4.6.2.1 Supported Memories or Devices

NAND flash and NOR flash architectures are the two flash technologies. The GPMC supports various types of external memory or devices, basically any one that supports NAND or NOR protocols:

- 8- and 16-bit-wide asynchronous or synchronous memory or device (only 8-bit: nonburst device)
- 16-bit address and data-multiplexed NOR flash devices (pSRAM, OneNAND™, etc.)
- 8- and 16-bit NAND flash devices

15.4.6.2.1.1 Memory Pin Multiplexing

This section describes the interfacing differences of the GPMC supported memories.

Table 15-467. Supported Memories Interfaces

<table>
<tr><th>Function</th><th>16-Bit Address/
Data-Multiplexed
pSRAM or NOR Flash [(1)]</th><th>OneNAND</th><th>16-Bit NAND</th><th>8-Bit NAND</th></tr>
<tr><td>gpmc_a[27]</td><td>A27</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[26]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[25]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[24]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[23]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[22]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[21]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[20]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[19]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[18]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[17]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[16]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[15]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[14]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[13]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[12]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[11]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[10]</td><td>A26</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[9]</td><td>A25</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[8]</td><td>A24</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[7]</td><td>A23</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[6]</td><td>A22</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[5]</td><td>A21</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[4]</td><td>A20</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[3]</td><td>A19</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[2]</td><td>A18</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[1]</td><td>A17</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_a[0]</td><td>A16</td><td></td><td></td><td></td></tr>
<tr><td>gpmc_ad[15]</td><td colspan="2">D15 or A16</td><td>IO15</td><td></td></tr>
<tr><td>gpmc_ad[14]</td><td colspan="2">D14 or A15</td><td>IO14</td><td></td></tr>
<tr><td>gpmc_ad[13]</td><td colspan="2">D13 or A14</td><td>IO13</td><td></td></tr>
<tr><td>gpmc_ad[12]</td><td colspan="2">D12 or A13</td><td>IO12</td><td></td></tr>
</table>

Table 15-467. Supported Memories Interfaces (continued)

Function	16-Bit Address/ Data-Multiplexed pSRAM or NOR Flash [(1)]	OneNAND	16-Bit NAND	8-Bit NAND
gpmc_ad[11]	D11 or A12		IO11	
gpmc_ad[10]	D10 or A11		IO10	
gpmc_ad[9]	D9 or A10		IO9	
gpmc_ad[8]	D8 or A9		IO8	
gpmc_ad[7]	D7 or A8		IO7	
gpmc_ad[6]	D6 or A7		IO6	
gpmc_ad[5]	D5 or A6		IO5	
gpmc_ad[4]	D4 or A5		IO4	
gpmc_ad[3]	D3 or A4		IO3	
gpmc_ad[2]	D2 or A3		IO2	
gpmc_ad[1]	D1 or A2		IO1	
gpmc_ad[0]	D0 or A1		IO0	
gpmc_clk	CLK			
gpmc_cs0	nCS0 (chip-select)		nCE0 (chip-enable)	
gpmc_cs1	nCS1		nCE1	
gpmc_cs2	nCS2		nCE2	
gpmc_cs3	nCS3		nCE3	
gpmc_cs4	nCS4		nCE4	
gpmc_cs5	nCS5		nCE5	
gpmc_cs6	nCS6		nCE6	
gpmc_cs7	nCS7		nCE7	
gpmc_advn_ale	nADV (address valid)		ALE (address latch enable)	
gpmc_oen_ren	nOE (output enable)		nRE (read enable)	
gpmc_wen	nWE (Write enable)		nWE (write enable)	
gpmc_ben0	nBE0 (byte enable)		CLE (command latch enable)	
gpmc_ben1	nBE1			
gpmc_wait0	WAIT0		R/nB0 (ready/busy)	
gpmc_wait1	WAIT1		R/nB1	

(1) Addresses seen from the device side. When interfacing to the external device, A1 is connected to the memory A0, A2 to the memory A1, and so on.

15.4.6.2.1.2 NAND Interface Protocol

NAND flash architecture, introduced in 1989, is a flash technology. NAND is a page-oriented memory device; that is, read and write accesses are done by pages. NAND achieves great density by sharing common areas of the storage transistor, which creates strings of serially connected transistors (in NOR devices, each transistor stands alone). Because of its high density NAND is best suited to devices that require high capacity data storage, such as pictures, music, and data files. NAND nonvolatility makes of it a good storage solution for many applications where mobility, low power, and speed are key factors. Low pin count and simple interface are other advantages of NAND.

Table 15-468 summarizes the NAND interface signals level applied to external device or memories.

Table 15-468. NAND Interface Bus Operations Summary

Bus Operation	CLE	ALE	nCE	nWE [(1)]	nRE [(1)]
Read (cmd input)	H	L	L	RE	H
Read (add input)	L	H	L	RE	H

Table 15-468. NAND Interface Bus Operations Summary (continued)

Bus Operation	CLE	ALE	nCE	nWE [(1)]	nRE [(1)]
Write (cmd input)	H	L	L	RE	H
Write (add input)	L	H	L	RE	H
Data input	L	L	L	RE	H
Data output	L	L	L	H	FE
Busy (during read)	x	x	H [(2)]	H [(2)]	H [(2)]
Busy (during program)	x	x	x	x	x
Busy (during erase)	x	x	x	x	x
Standby	x	x	H	x	x

(1) RE stands for rising edge; FE stands for falling edge
(2) Can be nCE high, or WE and nRE high.

15.4.6.2.1.3 NOR Interface Protocol

NOR flash architecture, introduced in 1988, is a flash technology. Unlike NAND, which is a sequential access device, NOR is directly addressable; that is, it is designed to be a random access device. NOR is best suited to devices used to store and run code or firmware, usually in small capacities. While NOR has fast read capabilities, it also has slow write and erase functions when compared to the NAND architecture.

Table 15-469 summarizes the level of the NOR interface signals applied to external devices or memories.

Table 15-469. NOR Interface Bus Operations Summary

Bus Operation	CLK	nADV	nCS	nOE	nWE	WAIT	DQ[15:0]
Read (asynchronous)	x	L	L	L	H	Asserted	Output
Read (synchronous)	Running	L	L	L	H	Driven	Output
Read (burst suspend)	Halted	x	L	H	H	Active	Output
Write	x	L	L	H	L	Asserted	Input
Output disable	x	x	L	H	H	Asserted	High-Z
Standby	x	x	H	x	x	High-Z	High-Z

15.4.6.2.1.4 Other Technologies

Other supported device types interact with the GPMC through the NOR interface protocol.

OneNAND is a high-density, low-power memory device. OneNAND is based on single- or multilevel-cell NAND core with SRAM and logic. It interfaces as a synchronous NOR flash and has synchronous write capability. It reads faster than conventional NAND and writes faster than conventional NOR flash. Hence, it is appropriate for mass storage and code storage.

pSRAM (pseudo-static random access memory) is a low-power memory device. pSRAM is based on the DRAM cell with internal refresh and address control features, and interfaces as a synchronous NOR flash. It also has synchronous write capability.

15.4.6.2.1.5 Supported Protocols

The GPMC supports the following interface protocols when communicating with external memory or external devices:

- Asynchronous read/write access
- Asynchronous read page access (4, 8, 16 Word16)
- Synchronous read/write access
- Synchronous read burst access without wrap capability (4, 8, 16 Word16)
- Synchronous read burst access with wrap capability (4, 8, 16 Word16)

15.4.6.2.2 GPMC Features and Settings

The features and settings of the GPMC are:

- Supported device type: Up to four NAND or NOR protocol external memories or devices
- Operating voltage: 1.8 V, 3.3 V
- Maximum operating frequency provided externally: See the device data manual for precise information.
- Maximum GPMC addressing capability: 512MiB divided into eight chip-selects
- Maximum supported memory size: 256MiB (must be a power-of-two)
- Minimum supported memory size: 16MiB (must be a power-of-two). Aliasing occurs when addressing smaller memories.
- Data path to external memory or device: 8, 16 bits wide
- Burst and page access: burst of 4, 8, 16 Word16
- Supports bus keeping
- Supports bus turnaround

15.4.7 GPMC Register Manual

This section provides information about the GPMC instance in this product. Table 15-471 provides a summary of the GPMC registers. The remaining parts of this section describe the registers within the module instance.

15.4.7.1 GPMC Register Summary

Table 15-470. GPMC Instance Summary

Module Name	Base Address	Size
GPMC	0x5000 0000	16 MiB

Table 15-471. GPMC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address GPMC
GPMC_REVISION	R	32	0x0000 0000	0x5000 0000
GPMC_SYSCONFIG	RW	32	0x0000 0010	0x5000 0010
GPMC_SYSSTATUS	R	32	0x0000 0014	0x5000 0014
GPMC_IRQSTATUS	RW	32	0x0000 0018	0x5000 0018
GPMC_IRQENABLE	RW	32	0x0000 001C	0x5000 001C
GPMC_TIMEOUT_CONTROL	RW	32	0x0000 0040	0x5000 0040
GPMC_ERR_ADDRESS	RW	32	0x0000 0044	0x5000 0044
GPMC_ERR_TYPE	RW	32	0x0000 0048	0x5000 0048
GPMC_CONFIG	RW	32	0x0000 0050	0x5000 0050
GPMC_STATUS	RW	32	0x0000 0054	0x5000 0054
GPMC_CONFIG1_i (1)	RW	32	0x0000 0060 + (0x0000 0030 * i)	0x5000 0060 + (0x0000 0030 * i)
GPMC_CONFIG2_i (1)	RW	32	0x0000 0064 + (0x0000 0030 * i)	0x5000 0064 + (0x0000 0030 * i)
GPMC_CONFIG3_i (1)	RW	32	0x0000 0068 + (0x0000 0030 * i)	0x5000 0068 + (0x0000 0030 * i)
GPMC_CONFIG4_i (1)	RW	32	0x0000 006C + (0x0000 0030 * i)	0x5000 006C + (0x0000 0030 * i)
GPMC_CONFIG5_i (1)	RW	32	0x0000 0070 + (0x0000 0030 * i)	0x5000 0070 + (0x0000 0030 * i)
GPMC_CONFIG6_i (1)	RW	32	0x0000 0074 + (0x0000 0030 * i)	0x5000 0074 + (0x0000 0030 * i)
GPMC_CONFIG7_i (1)	RW	32	0x0000 0078 + (0x0000 0030 * i)	0x5000 0078 + (0x0000 0030 * i)
GPMC_NAND_COMMAND_i (1)	W	32	0x0000 007C + (0x0000 0030 * i)	0x5000 007C + (0x0000 0030 * i)
GPMC_NAND_ADDRESS_i (1)	W	32	0x0000 0080 + (0x0000 0030 * i)	0x5000 0080 + (0x0000 0030 * i)
GPMC_NAND_DATA_i (1)	RW	32	0x0000 0084 + (0x0000 0030 * i)	0x5000 0084 + (0x0000 0030 * i)
GPMC_PREFETCH_CONFIG1	RW	32	0x0000 01E0	0x5000 01E0
GPMC_PREFETCH_CONFIG2	RW	32	0x0000 01E4	0x5000 01E4
GPMC_PREFETCH_CONTROL	RW	32	0x0000 01EC	0x5000 01EC
GPMC_PREFETCH_STATUS	RW	32	0x0000 01F0	0x5000 01F0
GPMC_ECC_CONFIG	RW	32	0x0000 01F4	0x5000 01F4
GPMC_ECC_CONTROL	RW	32	0x0000 01F8	0x5000 01F8
GPMC_ECC_SIZE_CONFIG	RW	32	0x0000 01FC	0x5000 01FC
GPMC_ECCj_RESULT (2)	RW	32	0x0000 0200 + (0x0000 0004 * j)	0x5000 0200 + (0x0000 0004 * j)
GPMC_BCH_RESULT0_i (1)	RW	32	0x0000 0240 + (0x0000 0010 * i)	0x5000 0240 + (0x0000 0010 * i)
GPMC_BCH_RESULT1_i (1)	RW	32	0x0000 0244 + (0x0000 0010 * i)	0x5000 0244 + (0x0000 0010 * i)
GPMC_BCH_RESULT2_i (1)	RW	32	0x0000 0248 + (0x0000 0010 * i)	0x5000 0248 + (0x0000 0010 * i)
GPMC_BCH_RESULT3_i (1)	RW	32	0x0000 024C + (0x0000 0010 * i)	0x5000 024C + (0x0000 0010 * i)
GPMC_BCH_RESULT4_i (1)	RW	32	0x0000 0300 + (0x0000 0010 * i)	0x5000 0300 + (0x0000 0010 * i)
GPMC_BCH_RESULT5_i (1)	RW	32	0x0000 0304 + (0x0000 0010 * i)	0x5000 0304 + (0x0000 0010 * i)
GPMC_BCH_RESULT6_i (1)	RW	32	0x0000 0308 + (0x0000 0010 * i)	0x5000 0308 + (0x0000 0010 * i)
GPMC_BCH_SWDATA	RW	32	0x0000 02D0	0x5000 02D0

(1) i = 0 to 7 for GPMC
(2) j = 0 to 8 for GPMC

15.4.7.2 GPMC Register Descriptions

Note

All GPMC registers are aligned to 32-bit address boundaries. All register file accesses, except to GPMC_NAND_DATA_i register, are little-endian. If the GPMC_NAND_DATA_i register location is accessed, the endianness is access-dependent.

In this section i corresponds to the chip-select number, where i = 0 to 7.

Table 15-472. GPMC_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5000 0000	**Instance**	GPMC
Description	This register contains the IP revision code.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 15-473. GPMC_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5000 0010	**Instance**	GPMC
Description	This register controls the various parameters of the interconnect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IDLEMODE		RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0000000
4:3	IDLEMODE	0x0: Force-idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged. 0x2: Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module. 0x3: Do not use.	RW	0x0
2	RESERVED	Write 0 for future compatibility Read returns 0.	RW	0x0
1	SOFTRESET	Software reset. Set this bit to 1 triggers a module reset. This bit is automatically reset by hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0x0
0	AUTOIDLE	Internal interface clock-gating strategy 0x0: Interface clock is free-running. 0x1: Automatic Interface clock gating strategy is applied, based on the interconnect activity.	RW	0x0

Table 15-474. GPMC_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x5000 0014	**Instance**	GPMC

Table 15-474. GPMC_SYSSTATUS (continued)

Description	This register provides status information about the module, excluding the interrupt status information.
Type	R

31-8	7-1	0
RESERVED	RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:1	RESERVED	Read returns 0 (reserved for interconnect-socket status information).	R	0x00
0	RESETDONE	Internal reset monitoring 0x0: Internal module reset is ongoing. 0x1: Reset is complete.	R	0x-

Table 15-475. GPMC_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	0x5000 0018	**Instance**	GPMC
Description	This interrupt status register regroups all the status of the module internal events that can generate an interrupt.		
Type	RW		

31-10	9	8	7-2	1	0
RESERVED	WAIT1EDGEDETECTIONSTATUS	WAIT0EDGEDETECTIONSTATUS	RESERVED	TERMINALCOUNTSTATUS	FIFOEVENTSTATUS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
9	WAIT1EDGEDETECTIONSTATUS	Status of the Wait1 Edge Detection interrupt Read 0x0: A transition on WAIT1 input pin has not been detected. Write 0x0: WAIT1EDGEDETECTIONSTATUS bit is unchanged. Read 0x1: A transition on WAIT1 input pin has been detected. Write 0x1: WAIT1EDGEDETECTIONSTATUS bit is reset.	RW	0x0
8	WAIT0EDGEDETECTIONSTATUS	Status of the Wait0 Edge Detection interrupt Read 0x0: A transition on WAIT0 input pin has not been detected. Write 0x0: WAIT0EDGEDETECTIONSTATUS bit is unchanged. Read 0x1: A transition on WAIT0 input pin has been detected. Write 0x1: WAIT0EDGEDETECTIONSTATUS bit is reset.	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00

Bits	Field Name	Description	Type	Reset
1	TERMINALCOUNTSTATUS	Status of the TerminalCountEvent interrupt Read 0x0: Indicates that CountValue is greater than 0 Write 0x0: TERMINALCOUNTSTATUS bit is unchanged. Read 0x1: Indicates that CountValue is equal to 0 Write 0x1: TERMINALCOUNTSTATUS bit is reset.	RW	0x0
0	FIFOEVENTSTATUS	Status of the FIFOEvent interrupt Read 0x0: Indicates that less than GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and less than FIFOTHRESHOLD bytes free places are available in write-posting mode Write 0x0: FIFOEVENTSTATUS bit is unchanged. Read 0x1: Indicates that at least GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and at least FIFOTHRESHOLD bytes free places are available in write-posting mode Write 0x1: FIFOEVENTSTATUS bit is reset.	RW	0x0

Table 15-476. GPMC_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	0x5000 001C	**Instance**	GPMC
Description	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td>WAIT1EDGEDETECTIONENABLE</td><td>WAIT0EDGEDETECTIONENABLE</td><td colspan="6">RESERVED</td><td>TERMINALCOUNTEVENTENABLE</td><td>FIFOEVENTENABLE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
9	WAIT1EDGEDETECTIONENABLE	Enables the Wait1 Edge Detection interrupt 0x0: Wait1EdgeDetection interrupt is masked. 0x1: Wait1EdgeDetection event generates an interrupt if occurs.	RW	0x0
8	WAIT0EDGEDETECTIONENABLE	Enables the Wait0 Edge Detection interrupt 0x0: Wait0EdgeDetection interrupt is masked. 0x1: Wait0EdgeDetection event generates an interrupt if occurs.	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
1	TERMINALCOUNTEVENTENABLE	Enables TerminalCountEvent interrupt issuing in prefetch or write-posting mode 0x0: TerminalCountEvent interrupt is masked. 0x1: TerminalCountEvent interrupt is not masked.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	FIFOEVENTENABLE	Enables the FIFOEvent interrupt 0x0: FIFOEvent interrupt is masked. 0x1: FIFOEvent interrupt is not masked.	RW	0x0

Table 15-477. GPMC_TIMEOUT_CONTROL

Address Offset	0x0000 0040		
Physical Address	0x5000 0040	**Instance**	GPMC
Description	The GPMC_TIMEOUT_CONTROL register allows the user to set the start value of the timeout counter.		
Type	RW		

31:13	12:4	3:1	0
RESERVED	TIMEOUTSTARTVALUE	RESERVED	TIMEOUTENABLE

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
12:4	TIMEOUTSTARTVALUE	Start value of the time-out counter 0x000: Zero GPMC_FCLK cycle 0x001: One GPMC_FCLK cycle ... 0x1FF: 511 GPMC_FCLK cycles	RW	0x1FF
3:1	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
0	TIMEOUTENABLE	Enable bit of the TimeOut feature 0x0: TimeOut feature is disabled. 0x1: TimeOut feature is enabled.	RW	0x0

Table 15-478. GPMC_ERR_ADDRESS

Address Offset	0x0000 0044		
Physical Address	0x5000 0044	**Instance**	GPMC
Description	The GPMC_ERR_ADDRESS register stores the address of the illegal access when an error occurs.		
Type	RW		

31	30:0
RESERVED	ILLEGALADD

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:0	ILLEGALADD	Address of illegal access A30: 0 for memory region, 1 for GPMC register region A29-A0: 1 GiB maximum	R	0x00000000

Table 15-479. GPMC_ERR_TYPE

Address Offset	0x0000 0048		
Physical Address	0x5000 0048	**Instance**	GPMC
Description	The GPMC_ERR_TYPE register stores the type of error when an error occurs.		
Type	RW		

31:11	10:8	7:5	4	3	2	1	0
RESERVED	ILLEGALMCMD	RESERVED	ERRORNOTSUPPADD	ERRORNOTSUPPMCMD	ERRORTIMEOUT	RESERVED	ERRORVALID

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
10:8	ILLEGALMCMD	System command of the transaction that caused the error	R	0x0
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4	ERRORNOTSUPPADD	Not supported address error 0x0: No error occurs. 0x1: The error is due to a nonsupported address.	R	0x0
3	ERRORNOTSUPPMCMD	Not supported command error 0x0: No error occurs. 0x1: The error is due to a nonsupported command	R	0x0
2	ERRORTIMEOUT	Time-out error 0x0: No error occurs. 0x1: The error is due to a timeout.	R	0x0
1	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
0	ERRORVALID	Error validity status - Must be explicitly cleared with a write 1 transaction 0x0: All error fields no longer valid 0x1: Error detected and logged in the other error fields	RW	0x0

Table 15-480. GPMC_CONFIG

Address Offset	0x0000 0050		
Physical Address	0x5000 0050	**Instance**	GPMC
Description	The configuration register allows global configuration of the GPMC.		
Type	RW		

31:10	9	8	7:2	1	0
RESERVED	WAIT1PINPOLARITY	WAIT0PINPOLARITY	RESERVED	RESERVED	NANDFORCEPOSTEDWRITE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000

Bits	Field Name	Description	Type	Reset
9	WAIT1PINPOLARITY	Selects the polarity of input pin WAIT1 0x0: WAIT1 active low 0x1: WAIT1 active high	RW	0x1
8	WAIT0PINPOLARITY	Selects the polarity of input pin WAIT0 0x0: WAIT0 active low 0x1: WAIT0 active high	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
1	RESERVED	Write 0 for future compatibility. Read returns 0.	RW	0x0
0	NANDFORCEPOSTEDWRITE	Enables the Force Posted Write feature to NAND Cmd/Add/Data location 0x0: Disables Force Posted Write 0x1: Enables Force Posted Write	RW	0x0

Table 15-481. GPMC_STATUS

Address Offset	0x0000 0054		
Physical Address	0x5000 0054	**Instance**	GPMC
Description	The status register provides global status bits of the GPMC.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td>WAIT1STATUS</td><td>WAIT0STATUS</td><td colspan="7">RESERVED</td><td>EMPTYWRITEBUFFERSTATUS</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
9	WAIT1STATUS	Is a copy of input pin WAIT1. (Reset value is WAIT1 input pin sampled at device reset.) 0x0: WAIT1 asserted (inactive state) 0x1: WAIT1 deasserted	R	0x-
8	WAIT0STATUS	Is a copy of input pin WAIT0. (Reset value is WAIT0 input pin sampled at device reset.) 0x0: WAIT0 asserted (inactive state) 0x1: WAIT0 deasserted	R	0x-
7:1	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x00
0	EMPTYWRITEBUFFERSTATUS	Stores the empty status of the write buffer 0x0: Write buffer is not empty. 0x1: Write buffer is empty.	R	0x1

Table 15-482. GPMC_CONFIG1_i

Address Offset	0x0000 0060 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0060 + (0x0000 0030 * i)	**Instance**	GPMC
Description	The configuration register 1 sets signal control parameters per chip-select.		

Table 15-482. GPMC_CONFIG1_i (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRAPBURST	READMULTIPLE	READTYPE	WRITEMULTIPLE	WRITETYPE	CLKACTIVATIONTIME		ATTACHEDDEVICEPAGELENGTH		WAITREADMONITORING	WAITWRITEMONITORING	RESERVED	WAITMONITORINGTIME		WAITPINSELECT		RESERVED		DEVICESIZE		DEVICETYPE		MUXADDDATA		RESERVED			TIMEPARAGRANULARITY	RESERVED		GPMCFCLKDIVIDER	

Bits	Field Name	Description	Type	Reset
31	WRAPBURST	Enables the wrapping burst capability. Must be set if the attached device is configured in wrapping burst 0x0: Synchronous wrapping burst not supported 0x1: Synchronous wrapping burst supported	RW	0x0
30	READMULTIPLE	Selects the read single or multiple access 0x0: Single access 0x1: Multiple access (burst if synchronous, page if asynchronous)	RW	0x0
29	READTYPE	Selects the read mode operation 0x0: Read asynchronous 0x1: Read synchronous	RW	0x0
28	WRITEMULTIPLE	Selects the write single or multiple access 0x0: Single access 0x1: Multiple access (burst if synchronous, considered as single if asynchronous)	RW	0x0
27	WRITETYPE	Selects the write mode operation 0x0: Write asynchronous 0x1: Write synchronous	RW	0x0
26:25	CLKACTIVATIONTIME	Output GPMC_CLK activation time 0x0: First rising edge of GPMC_CLK at start access time 0x1: First rising edge of GPMC_CLK one GPMC_FCLK cycle after start access time 0x2: First rising edge of GPMC_CLK two GPMC_FCLK cycles after start access time 0x3: Reserved	RW	0x0
24:23	ATTACHEDDEVICEPAGELENGTH	Specifies the attached device page (burst) length 0x0: 4 words 0x1: 8 words 0x2: 16 words 0x3: Reserved (1 word = interface size)	RW	0x0
22	WAITREADMONITORING	Selects the Wait monitoring configuration for Read accesses (Reset value is *bootwaiten* input pin sampled at device reset) 0x0: Wait pin is not monitored for read accesses. 0x1: Wait pin is monitored for read accesses.	RW	0x-

Bits	Field Name	Description	Type	Reset
21	WAITWRITEMONITORING	Selects the Wait monitoring configuration for Write accesses 0x0: Wait pin is not monitored for write accesses. 0x1: Wait pin is monitored for write accesses.	RW	0x0
20	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
19:18	WAITMONITORINGTIME	Selects input pin Wait monitoring time 0x0: Wait pin is monitored with valid data. 0x1: Wait pin is monitored one GPMC_CLK cycle before valid data. 0x2: Wait pin is monitored two GPMC_CLK cycle before valid data. 0x3: Reserved	RW	0x0
17:16	WAITPINSELECT	Selects the input wait pin for this chip-select (The reset value is HW fixed to 0x0 for CS0-CS7) 0x0: Wait input pin is WAIT0. 0x1: Wait input pin is WAIT1. 0x2, 0x3: Reserved	RW	0x0
15:14	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x0
13:12	DEVICESIZE	Selects the device size attached (Reset value is *bootdevicesize* input pin sampled at device reset for CS0 and 0x1 for CS1 to CS7) 0x0: 8 bit 0x1: 16 bit 0x2: Reserved 0x3: Reserved	RW	0x-
11:10	DEVICETYPE	Selects the attached device type 0x0: NOR flash-like, asynchronous and synchronous devices 0x1: Reserved 0x2: NAND flash-like devices, stream mode 0x3: Reserved	RW	0x0
9:8	MUXADDDATA	Enables the address and data multiplexed protocol (Reset value is *cs0muxdevice* input pin sampled at device reset for CS0 and 0 for CS1-CS7) 0x0: Nonmultiplexed attached device 0x1: AAD-multiplexed protocol device 0x2: Address and data multiplexed attached device 0x3: Reserved	RW	0x-
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4	TIMEPARAGRANULARITY	Signals timing latencies scalar factor (RD/WRCYCLETIME, RD/WRACCESSTIME, PAGEBURSTACCESSTIME, CSONTIME, CSRD/WROFFTIME, ADVONTIME, ADVRD/WROFFTIME, OEONTIME, OEOFFTIME, WEONTIME, WEOFFTIME, CYCLE2CYCLEDELAY, BUSTURNAROUND, TIMEOUTSTARTVALUE, WRDATAONADMUXBUS) 0x0: x1 latencies 0x1: x2 latencies	RW	0x0
3:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	GPMCFCLKDIVIDER	Divides the GPMC_FCLK clock 0x0: GPMC_CLK frequency = GPMC_FCLK frequency 0x1: GPMC_CLK frequency = GPMC_FCLK frequency / 2 0x2: GPMC_CLK frequency = GPMC_FCLK frequency / 3 0x3: GPMC_CLK frequency = GPMC_FCLK frequency /4	RW	0x0

Table 15-483. GPMC_CONFIG2_i

Address Offset	0x0000 0064 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0064 + (0x0000 0030 * i)	**Instance**	GPMC
Description	CS signal timing parameter configuration		
Type	RW		

31:21	20:16	15:13	12:8	7	6:4	3:0
RESERVED	CSWROFFTIME	RESERVED	CSRDOFFTIME	CSEXTRADELAY	RESERVED	CSONTIME

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000
20:16	CSWROFFTIME	CS i deassertion time from start cycle time for write accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
15:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
12:8	CSRDOFFTIME	CS i de-assertion time from start cycle time for read accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
7	CSEXTRADELAY	CS i Add extra half-GPMC_FCLK cycle 0x0: CS i Timing control signal is not delayed 0x1: CS i Timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0x0
6:4	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
3:0	CSONTIME	CS i assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1

Table 15-484. GPMC_CONFIG3_i

Address Offset	0x0000 0068 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0068 + (0x0000 0030 * i)	**Instance**	GPMC
Description	nADV signal timing parameter configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ADVAADMUXWROFFTIME	RESERVED	ADVAADMUXRDOFFTIME	RESERVED	ADVWROFFTIME	RESERVED	ADVRDOFFTIME	ADVEXTRADELAY	ADVAADMUXONTIME	ADVONTIME

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0
30:28	ADVAADMUXWROFFTIME	nADV deassertion for first address phase when using the AAD-multiplexed protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x2
27	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0
26:24	ADVAADMUXRDOFFTIME	nADV assertion for first address phase when using the AAD-multiplexed protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x2
23:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
20:16	ADVWROFFTIME	nADV deassertion time from start cycle time for write accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x06
15:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
12:8	ADVRDOFFTIME	nADV deassertion time from start cycle time for read accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x05
7	ADVEXTRADELAY	nADV add extra half-GPMC_FCLK cycle 0x0: nADV timing control signal is not delayed 0x1: nADV timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0
6:4	ADVAADMUXONTIME	nADV assertion for first address phase when using the AAD-multiplexed protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x1
3:0	ADVONTIME	nADV assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x4

Table 15-485. GPMC_CONFIG4_i

Address Offset	0x0000 006C + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 006C + (0x0000 0030 * i)	**Instance**	GPMC
Description	nWE and nOE signals timing parameter configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	WEOFFTIME	WEEXTRADELAY	RESERVED	WEONTIME	OEAADMUX OFFTIME	OEOFFTIME	OEEXTRADELAY	OEAADMUX ONTIME	OEONTIME

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
28:24	WEOFFTIME	nWE deassertion time from start cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
23	WEEXTRADELAY	nWE add extra half-GPMC_FCLK cycle 0x0: nWE timing control signal is not delayed 0x1: nWE timing control signal is delayed of half-GPMC_FCLK clock cycle	RW	0
22:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	WEONTIME	nWE assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x5
15:13	OEAADMUX OFFTIME	nOE deassertion time for the first address phase in an AAD-multiplexed access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x3
12:8	OEOFFTIME	nOE deassertion time from start cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
7	OEEXTRADELAY	nOE add extra half-GPMC_FCLK cycle 0x0: nOE timing control signal is not delayed 0x1: nOE timing control signal is delayed of half-GPMC_FCLK clock cycle	RW	0
6:4	OEAADMUX ONTIME	nOE assertion time for the first address phase in an AAD-mux access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x1
3:0	OEONTIME	nOE assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x6

Table 15-486. GPMC_CONFIG5_i

Address Offset	0x0000 0070 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0070 + (0x0000 0030 * i)	**Instance**	GPMC
Description	RdAccessTime and CycleTime timing parameters configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PAGEBURSTACCESSTIME	RESERVED	RDACCESSTIME	RESERVED	WRCYCLETIME	RESERVED	RDCYCLETIME

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
27:24	PAGEBURSTACCESSTIME	Delay between successive words in a multiple access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1
23:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
20:16	RDACCESSTIME	Delay between start cycle time and first data valid 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x0F
15:13	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x0
12:8	WRCYCLETIME	Total write cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x11
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4:0	RDCYCLETIME	Total read cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x11

Table 15-487. GPMC_CONFIG6_i

Address Offset	0x0000 0074 + (0x0000 0030 * i)	**Index**	i = 0 to7
Physical Address	0x5000 0074 + (0x0000 0030 * i)	**Instance**	GPMC
Description	WrAccessTime, WrDataOnADmuxBus, Cycle2Cycle and BusTurnAround parameters configuration		
Type	RW		

31	30:29	28:24	23:20	19:16	15:12	11:8	7	6	5:4	3:0
RESERVED	RESERVED	WRACCESSTIME	RESERVED	WRDATAONADMUXBUS	RESERVED	CYCLE2CYCLEDELAY	CYCLE2CYCLESAMECSEN	CYCLE2CYCLEDIFFCSEN	RESERVED	BUSTURNAROUND

Bits	Field Name	Description	Type	Reset
31	RESERVED	TI Internal use - Do not modify.	RW	1
30:29	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
28:24	WRACCESSTIME	Delay from start access time to the GPMC_FCLK rising edge corresponding the GPMC_CLK rising edge used by the attached memory for the first data capture 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x0F
23:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0

Bits	Field Name	Description	Type	Reset
19:16	WRDATAONADMUXBUS	Specifies on which GPMC_FCLK rising edge the first data of the write is driven in the add/data mux bus	RW	0x7
15:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
11:8	CYCLE2CYCLEDELAY	Chip-select high pulse delay between successive accesses 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x0
7	CYCLE2CYCLESAMECSEN	Add CYCLE2CYCLEDELAY between successive accesses to the same chip-select (any access type) 0x0: No delay between the two accesses 0x1: Add CYCLE2CYCLEDELAY	RW	0
6	CYCLE2CYCLEDIFFCSEN	Add CYCLE2CYCLEDELAY between successive accesses to a different chip-select (any access type) 0x0: No delay between the two accesses 0x1: Add CYCLE2CYCLEDELAY	RW	0x0
5:4	RESERVED	Write 0s for future compatibility. Reads return 0.	RW	0x0
3:0	BUSTURNAROUND	Bus turnaround latency between successive accesses to the same chip-select (read to write) or to a different chip-select (read to read and read to write) 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x0

Table 15-488. GPMC_CONFIG7_i

Address Offset	0x0000 0078 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0078 + (0x0000 0030 * i)	**Instance**	GPMC
Description	CS address mapping configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				MASKADDRESS				RESERVED	CSVALID	BASEADDRESS					

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
11:8	MASKADDRESS	CS mask address. 0x0000: Chip-select size of 256 MiB 0x1000: Chip-select size of 128 MiB 0x1100: Chip-select size of 64 MiB 0x1110: Chip-select size of 32 MiB 0x1111: Chip-select size of 16 MiB Other values must be avoided as they create holes in the chip-select address space.	RW	0xF
7	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
6	CSVALID	CS enable 0x0: CS disabled 0x1: CS enabled	RW	See (1)
5:0	BASEADDRESS	CSi base address where i = 0 to 7 (16-MiB minimum granularity) bits [5:0] corresponds to A29, A28, A27, A26, A25, and A24. See Figure 15-59	RW	0x00

(1) Reset value is 0x1 for CS0 and 0x0 for CS1 to CS7

Table 15-489. GPMC_NAND_COMMAND_i

Address Offset	0x0000 007C + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 007C + (0x0000 0030 * i)	**Instance**	GPMC
Description	This register is not a true register, only an address location.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_COMMAND																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_COMMAND	This register is not a true register, only an address location. Writing data at the GPMC_NAND_COMMAND_i location places the data as the NAND command value on the bus, using a regular asynchronous write access.	W	0x-

Table 15-490. GPMC_NAND_ADDRESS_i

Address Offset	0x0000 0080 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0080 + (0x0000 0030 * i)	**Instance**	GPMC
Description	This register is not a true register, only an address location.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_ADDRESS	This register is not a true register, only an address location. Writing data at the GPMC_NAND_ADDRESS_i location places the data as the NAND partial address value on the bus, using a regular asynchronous write access.	W	0x-

Table 15-491. GPMC_NAND_DATA_i

Address Offset	0x0000 0084 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0084 + (0x0000 0030 * i)	**Instance**	GPMC
Description	This register is not a true register,only an address location.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_DATA																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_DATA	This register is not a true register, only an address location. Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.	W	0x-

Table 15-492. GPMC_PREFETCH_CONFIG1

Address Offset	0x0000 01E0		
Physical Address	0x5000 01E0	**Instance**	GPMC
Description	Prefetch engine configuration 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CYCLEOPTIMIZATION	ENABLEOPTIMIZEDACCESS	ENGINECSSELECTOR	PFPWENROUNDROBIN	RESERVED	PFPWWEIGHTEDPRIO	RESERVED	FIFOTHRESHOLD	ENABLEENGINE	RESERVED	WAITPINSELECTOR	SYNCHROMODE	DMAMODE	RESERVED	ACCESSMODE

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:28	CYCLEOPTIMIZATION	Define the number of GPMC_FCLK cycles to be subtracted from RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, CSRDOFFTIME, CSWROFFTIME, ADVRDOFFTIME, ADVWROFFTIME, OEOFFTIME, WEOFFTIME 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x0
27	ENABLEOPTIMIZEDACCESS	Enables access cycle optimization 0x0: Access cycle optimization is disabled. 0x1: Access cycle optimization is enabled.	RW	0x0
26:24	ENGINECSSELECTOR	Selects the chip-select where Prefetch Postwrite engine is active 0x0: CS0 0x1: CS1 0x2: CS2 0x3: CS3 0x4: CS4 0x5: CS5 0x6: CS6 0x7: CS7	RW	0x0
23	PFPWENROUNDROBIN	Enables the PFPW RoundRobin arbitration 0x0: Prefetch Postwrite engine round robin arbitration is disabled. 0x1: Prefetch Postwrite engine round robin arbitration is enabled.	RW	0x0
22:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	PFPWWEIGHTEDPRIO	When an arbitration occurs between a DMA and a PFPW engine access, the DMA is always serviced. If the PFPWEnRoundRobin is enabled, 0x0: The next access is granted to the PFPW engine. 0x1: The next two accesses are granted to the PFPW engine. ..., 0xF: The next 16 accesses are granted to the PFPW engine.	RW	0x0
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
14:8	FIFOTHRESHOLD	Selects the maximum number of bytes read from the FIFO or written to the FIFO by the host on a DMA or interrupt request 0x00: 0 byte 0x01: 1 byte ... 0x40: 64 bytes	RW	0x40

Bits	Field Name	Description	Type	Reset
7	ENABLEENGINE	Enables the Prefetch Postwite engine 0x0: Prefetch Postwrite engine is disabled. 0x1: Prefetch Postwrite engine is enabled.	RW	0x0
6	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
5:4	WAITPINSELECTOR	Select which wait pin edge detector should start the engine in synchronized mode 0x0: Selects Wait0 EdgeDetection 0x1: Selects Wait1 EdgeDetection 0x2, 0x3: Reserved	RW	0x0
3	SYNCHROMODE	Selects when the engine starts the access to chip-select 0x0: Engine starts the access to chip-select as soon as STARTENGINE is set 0x1: Engine starts the access to chip-select as soon as STARTENGINE is set AND wait to nonwait edge detection on the selected wait pin	RW	0x0
2	DMAMODE	Selects interrupt synchronization or DMA request synchronization 0x0: Interrupt synchronization is enabled. Only interrupt line is activated on FIFO threshold crossing. 0x1: DMA request synchronization is enabled. A DMA request protocol is used.	RW	0x0
1	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
0	ACCESSMODE	Selects prefetch read or write-posting accesses 0x0: Prefetch read mode 0x1: Write-posting mode	RW	0x0

Table 15-493. GPMC_PREFETCH_CONFIG2

Address Offset	0x0000 01E4		
Physical Address	0x5000 01E4	**Instance**	GPMC
Description	Prefetch engine configuration 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		TRANSFERCOUNT													

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
13:0	TRANSFERCOUNT	Selects the number of bytes to be read or written by the engine to the selected chip-select 0x0000: 0 byte 0x0001: 1 byte ... 0x2000: 8 Kbytes	RW	0x0000

Table 15-494. GPMC_PREFETCH_CONTROL

Address Offset	0x0000 01EC		
Physical Address	0x5000 01EC	**Instance**	GPMC
Description	Prefetch engine control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STARTENGINE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000000
0	STARTENGINE	Resets the FIFO pointer and starts the engine Read 0x0: Engine is stopped. Write 0x0: Stops the engine Read 0x1: Engine is running. Write 0x1: Resets the FIFO pointer to 0x0 in prefetch mode and 0x40 in postwrite mode and starts the engine	RW	0x0

Table 15-495. GPMC_PREFETCH_STATUS

Address Offset	0x0000 01F0		
Physical Address	0x5000 01F0	**Instance**	GPMC
Description	Prefetch engine status		
Type	RW		

31	30:24	23:17	16	15:14	13:0
RESERVED	FIFOPOINTER	RESERVED	FIFOTHRESHOLDSTATUS	RESERVED	COUNTVALUE

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:24	FIFOPOINTER	Number of available bytes to be read or number of free empty byte places to be written 0x00: 0 byte available to be read or 0 free empty place to be written ... 0x40: 64 bytes available to be read or 64 empty places to be written	R	0x00
23:17	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
16	FIFOTHRESHOLDSTATUS	Set when FIFOPointer exceeds FIFOThreshold value 0x0: FIFOPointer smaller or equal to FIFOThreshold. Writing to this bit has no effect. 0x1: FIFOPointer greater than FIFOThreshold. Writing to this bit has no effect.	R	0x0
15:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
13:0	COUNTVALUE	Number of remaining bytes to be read or to be written by the engine according to the TransferCount value 0x0000: 0 byte remaining to be read or to be written 0x0001: 1 byte remaining to be read or to be written ... 0x2000: 8 KiB remaining to be read or to be written	R	0x0000

Table 15-496. GPMC_ECC_CONFIG

Address Offset	0x0000 01F4		
Physical Address	0x5000 01F4	**Instance**	GPMC
Description	ECC configuration		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">RESERVED</td><td>ECCALGORITHM</td><td colspan="2">RESERVED</td><td colspan="2">ECCBCHTSEL</td><td colspan="4">ECCWRAPMODE</td><td>ECC16B</td><td colspan="3">ECCTOPSECTOR</td><td colspan="3">ECCCS</td><td>ECCENABLE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0000
16	ECCALGORITHM	ECC algorithm used 0x0: Hamming code 0x1: BCH code	RW	0x0
15:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
13:12	ECCBCHTSEL	Error correction capability used for BCH 0x0: Up to 4 bits error correction (t = 4) 0x1: Up to 8 bits error correction (t = 8) 0x2: Up to 16 bits error correction (t = 16) 0x3: Reserved	RW	0x1
11:8	ECCWRAPMODE	Spare area organization definition for the BCH algorithm. See the BCH syndrome/parity calculator module functional specification for more details	RW	0x0
7	ECC16B	Selects an ECC calculated on 16 columns 0x0: ECC calculated on 8 columns 0x1: ECC calculated on 16 columns	RW	0x0
6:4	ECCTOPSECTOR	Number of sectors to process with the BCH algorithm 0x0: 1 sector (512-kB page) 0x1: 2 sectors ... 0x3: 4 sectors (2-kB page) ... 0x7: 8 sectors (4-kB page)	RW	0x3
3:1	ECCCS	Selects the CS where ECC is computed 0x0: CS0 0x1: CS1 0x2: CS2 0x3: CS3 Other: Reserved	RW	0x0
0	ECCENABLE	Enables the ECC feature 0x0: ECC disabled 0x1: ECC enabled	RW	0x0

Table 15-497. GPMC_ECC_CONTROL

Address Offset	0x0000 01F8		
Physical Address	0x5000 01F8	**Instance**	GPMC
Description	ECC control		

Table 15-497. GPMC_ECC_CONTROL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							ECCCLEAR	RESERVED				ECCPOINTER			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
8	ECCCLEAR	Clear all ECC result registers Reads return 0. Write 0x1 to this field clears all ECC result registers. Write 0x0 is ignored.	RW	0x0
7:4	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
3:0	ECCPOINTER	Selects ECC result register (Reads to this field give the dynamic position of the ECC pointer - Writes to this field select the ECC result register where the first ECC computation will be stored.); Other enums: writing other values disables the ECC engine (ECCENABLE bit of GPMC_ECC_CONFIG set to 0) 0x0: Writing 0x0 disables the ECC engine (ECCENABLE bit of GPMC_ECC_CONFIG set to 0) 0x1: ECC result register 1 selected 0x2: ECC result register 2 selected 0x3: ECC result register 3 selected 0x4: ECC result register 4 selected 0x5: ECC result register 5 selected 0x6: ECC result register 6 selected 0x7: ECC result register 7 selected 0x8: ECC result register 8 selected 0x9: ECC result register 9 selected	RW	0x0

Table 15-498. GPMC_ECC_SIZE_CONFIG

Address Offset	0x0000 01FC		
Physical Address	0x5000 01FC	**Instance**	GPMC
Description	ECC size		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		ECCSIZE1								RESERVED		ECCSIZE0								RESERVED			ECC9RESULTSIZE	ECC8RESULTSIZE	ECC7RESULTSIZE	ECC6RESULTSIZE	ECC5RESULTSIZE	ECC4RESULTSIZE	ECC3RESULTSIZE	ECC2RESULTSIZE	ECC1RESULTSIZE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0s for future compatibility. Read returns 3.	RW	0x3

Bits	Field Name	Description	Type	Reset
29:22	ECCSIZE1	Defines Hamming code ECC size 1 in bytes 0x00: 2 bytes 0x01: 4 bytes 0x02: 6 bytes 0x03: 8 bytes ... 0xFF: 512 bytes For BCH code ECC, the size 1 is programmed directly with the number of nibbles. For details, see *Wrapping Modes*.	RW	0xFF
21:20	RESERVED	Write 0s for future compatibility. Read returns 3.	RW	0x3
19:12	ECCSIZE0	Defines Hamming code ECC size 0 in bytes 0x00: 2 bytes 0x01: 4 bytes 0x02: 6 bytes 0x03: 8 bytes ... 0xFF: 512 bytes For BCH code ECC, the size 0 is programmed directly with the number of nibbles. For details, see *Wrapping Modes*.	RW	0xFF
11:9	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
8	ECC9RESULTSIZE	Selects ECC size for ECC 9 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
7	ECC8RESULTSIZE	Selects ECC size for ECC 8 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
6	ECC7RESULTSIZE	Selects ECC size for ECC 7 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
5	ECC6RESULTSIZE	Selects ECC size for ECC 6 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
4	ECC5RESULTSIZE	Selects ECC size for ECC 5 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
3	ECC4RESULTSIZE	Selects ECC size for ECC 4 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
2	ECC3RESULTSIZE	Selects ECC size for ECC 3 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
1	ECC2RESULTSIZE	Selects ECC size for ECC 2 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
0	ECC1RESULTSIZE	Selects ECC size for ECC 1 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0

Table 15-499. GPMC_ECCj_RESULT

Address Offset	0x0000 0200 + (0x0000 0004 * (j - 1))	**Index**	j = 1 to 9
Physical Address	0x5000 0200 + (0x0000 0004 * j)	**Instance**	GPMC

Table 15-499. GPMC_ECCj_RESULT (continued)

Description	ECC result register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				P2048O	P1024O	P512O	P256O	P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O	RESERVED				P2048E	P1024E	P512E	P256E	P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
27	P2048O	Odd row parity bit 2048, only used for ECC computed on 512 bytes	R	0x0
26	P1024O	Odd row parity bit 1024	R	0x0
25	P512O	Odd row parity bit 512	R	0x0
24	P256O	Odd row parity bit 256	R	0x0
23	P128O	Odd row parity bit 128	R	0x0
22	P64O	Odd row parity bit 64	R	0x0
21	P32O	Odd row parity bit 32	R	0x0
20	P16O	Odd row parity bit 16	R	0x0
19	P8O	Odd row parity bit 8	R	0x0
18	P4O	Odd Column Parity bit 4	R	0x0
17	P2O	Odd Column Parity bit 2	R	0x0
16	P1O	Odd Column Parity bit 1	R	0x0
15:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
11	P2048E	Even row parity bit 2048, only used for ECC computed on 512 bytes	R	0x0
10	P1024E	Even row parity bit 1024	R	0x0
9	P512E	Even row parity bit 512	R	0x0
8	P256E	Even row parity bit 256	R	0x0
7	P128E	Even row parity bit 128	R	0x0
6	P64E	Even row parity bit 64	R	0x0
5	P32E	Even row parity bit 32	R	0x0
4	P16E	Even row parity bit 16	R	0x0
3	P8E	Even row parity bit 8	R	0x0
2	P4E	Even column parity bit 4	R	0x0
1	P2E	Even column parity bit 2	R	0x0
0	P1E	Even column parity bit 1	R	0x0

Table 15-500. GPMC_BCH_RESULT0_i

Address Offset	0x0000 0240 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0240 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 0 to 31)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_0																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_0	BCH ECC result (bits 0 to 31)	RW	0x00000000

Table 15-501. GPMC_BCH_RESULT1_i

Address Offset	0x0000 0244 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0244 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 32 to 63)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_1																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_1	BCH ECC result (bits 32 to 63)	RW	0x00000000

Table 15-502. GPMC_BCH_RESULT2_i

Address Offset	0x0000 0248 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0248 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 64 to 95)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_2																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_2	BCH ECC result (bits 64 to 95)	RW	0x00000000

Table 15-503. GPMC_BCH_RESULT3_i

Address Offset	0x0000 024C + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 024C + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 96 to 127)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_3																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_3	BCH ECC result (bits 96 to 127)	RW	0x00000000

Table 15-504. GPMC_BCH_RESULT4_i

Address Offset	0x0000 0300 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0300 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 128 to 159)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_4																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_4	BCH ECC result (bits 128 to 159)	RW	0x00000000

Table 15-505. GPMC_BCH_RESULT5_i

Address Offset	0x0000 0304 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0304 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 160 to 191)		

Table 15-505. GPMC_BCH_RESULT5_i (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_5																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_5	BCH ECC result (bits 160 to 191)	RW	0x00000000

Table 15-506. GPMC_BCH_RESULT6_i

Address Offset	0x0000 0308 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0308 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 192 to 207)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCH_RESULT_6															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Read returns 0s.	R	0x0000
15:0	BCH_RESULT_6	BCH ECC result (bits 192 to 207)	RW	0x0000

Table 15-507. GPMC_BCH_SWDATA

Address Offset	0x0000 02D0		
Physical Address	0x5000 02D0	**Instance**	GPMC
Description	This register is used to directly pass data to the BCH ECC calculator without accessing the actual NAND flash interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCH_DATA															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Read returns 0s.	R	0x0000
15:0	BCH_DATA	Data to be included in the BCH calculation Only bits 0 to 7 are considered if the calculator is configured to use 8-bit data (GPMC_ECC_CONFIG[7] ECC16B = 0)	RW	0x0000

15.5 Error Location Module

15.5.1 Error Location Module Overview

When reading from NAND flash memories, some level of error-correction is required. In the case of NAND modules with no internal correction capability, sometimes referred to as *bare NANDs*, the correction process is delegated to the memory controller.

The general-purpose memory controller (GPMC) probes data read from an external NAND flash and uses this to compute checksum-like information, called syndrome polynomials, on a per-block basis. Each syndrome polynomial gives a status of the read operations for a full block, including 512 bytes of data, parity bits, and an optional spare-area data field, with a maximum block size of 1023 bytes. Computation is based on a Bose-Chaudhuri-Hocquenghem (BCH) algorithm. The error-location module (ELM) extracts error addresses from these syndrome polynomials.

Based on the syndrome polynomial value, the ELM can detect errors, compute the number of errors, and give the location of each error bit. The actual data is not required to complete the error-correction algorithm. Errors can be reported anywhere in the NAND flash block, including in the parity bits.

The maximum acceptable number of errors that can be corrected depends on a programmable configuration parameter. 4-, 8-, and 16-bit error-correction levels are supported. The ELM relies on a static and fixed definition of the generator polynomial for each error-correction level that corresponds to the generator polynomials defined in the GPMC (there are three fixed polynomial for the three correction error levels). A larger number of errors than the programmed error-correction level may be detected, but the ELM cannot correct them all. The offending block is then tagged as *uncorrectable* in the associated computation exit status register. If the computation is successful, that is, if the number of errors detected does not exceed the maximum value authorized for the chosen correction capability, the exit status register contains the information on the number of detected errors.

When the error-location process completes, an interrupt is triggered to inform the software that its status can be checked. The number of detected errors and their locations in the NAND block can be retrieved from the module through register accesses.

The ELM has the following features:

- 4, 8 and 16 bits per 512-byte block error-location based on BCH algorithms
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation on error-location process completion:
  - When the full page has been processed in page mode
  - For each syndrome polynomial in continuous mode

Figure 15-104 shows the ELM overview.

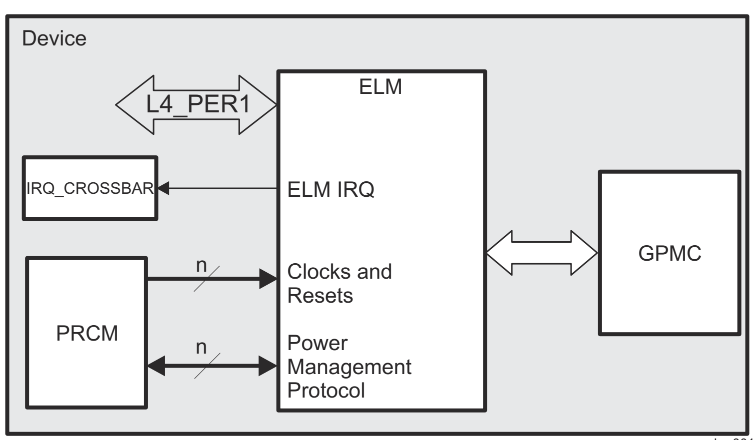


Figure 15-104. ELM Overview

15.5.2 ELM Integration

The ELM extracts error addresses from generated syndrome polynomials.

The ELM is used with the GPMC. Syndrome polynomials generated on-the-fly when reading a NAND flash page and stored in GPMC registers are passed to the ELM. The microprocessor unit (MPU) can then correct the data block by flipping the bits to which the ELM error-location outputs point.

Figure 15-105 shows the integration of the ELM subsystem in the device.

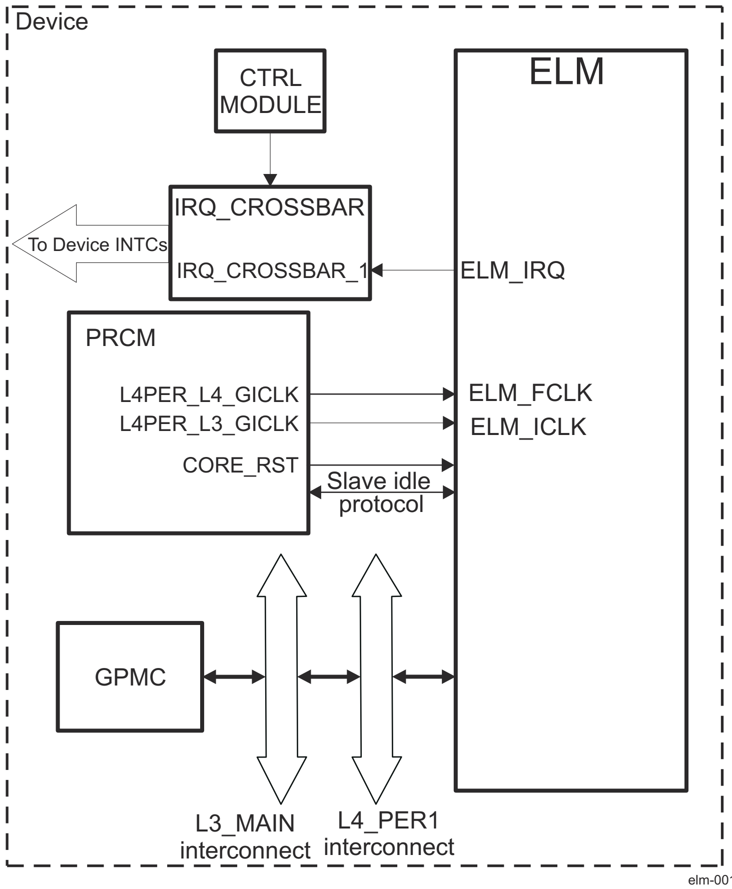


Figure 15-105. ELM Integration

Table 15-508 through Table 15-510 summarize the integration of the module in the device.

Table 15-508. ELM Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
ELM	PD_COREAON	No	L4_PER1

Table 15-509. ELM Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ELM	ELM_FCLK	L4PER_L4_GICLK	PRCM	Functional clock
	ELM_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ELM	ELM_RST	L4PER_RST	PRCM	Module hardware reset

Table 15-510. ELM Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR**	**Default Mapping**	**Description**
			MPU_IRQ_4	BCH error-location module interrupt
ELM	ELM_IRQ	IRQ_CROSSBAR_1	DSP1_IRQ_32	BCH error-location module interrupt
			DSP2_IRQ_32	BCH error-location module interrupt
			EVE1_IRQ_1	BCH error-location module interrupt
			EVE2_IRQ_1	BCH error-location module interrupt

Note

The “**Default Mapping**” column in Table 15-510 *ELM Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

15.5.3 ELM Functional Description

The ELM is designed around the error-location engine, which handles the computation based on the input syndrome polynomials.

The ELM maps the error-location engine to a standard interconnect interface by using a set of registers to control inputs and outputs.

15.5.3.1 ELM Software Reset

To perform a software reset, set the ELM_SYSCONFIG[1] SOFTRESET bit to 1. The ELM_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the ELM_SYSCONFIG[1] SOFTRESET bit is automatically reset.

15.5.3.2 ELM Power Management

Table 15-511 describes the power-management features available to the ELM.

Note

- For information about source clock gating and a description of the sleep/wake-up transitions, see the *CD_L4_PER1 Clock Domain*, in *Power, Reset, and Clock Management*.
- For a general description of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see *Clock Management Functional Description*, in *Power, Reset, and Clock Management*

Table 15-511. Local Power-Management Features

Feature	Registers	Description
Clock autogating	ELM_SYSCONFIG[0] AUTOGATING	This bit allows a local power optimization inside the module by gating the ELM_FCLK clock upon the interface activity.
Slave idle modes	ELM_SYSCONFIG[4:3] SIDLEMODE	Force-idle, no-idle, and smart-idle modes are available.
Clock activity	ELM_SYSCONFIG[8] CLOCKACTIVITY	The clock can be switched off or maintained during the wake-up period.
Master standby modes	N/A	
Global wake-up enable	N/A	
Wake-up sources enable	N/A	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the ELM CLOCKACTIVITY and ELM clock PRCM control bits. For a description of the ClockActivity feature, see *Power, Reset, and Clock Management*.

15.5.3.3 ELM Interrupt Requests

Table 15-512 lists the event flags, and their masks, that can cause module interrupts.

Table 15-512. Events

Event Flag	Event Mask	Description
ELM_IRQSTATUS[8] PAGE_VALID	ELM_IRQENABLE[8] PAGE_MASK	Page interrupt
ELM_IRQSTATUS[7] LOC_VALID_7	ELM_IRQENABLE[7] LOCATION_MASK_7	Error-location interrupt for syndrome polynomial 7
ELM_IRQSTATUS[6] LOC_VALID_6	ELM_IRQENABLE[6] LOCATION_MASK_6	Error-location interrupt for syndrome polynomial 6
ELM_IRQSTATUS[5] LOC_VALID_5	ELM_IRQENABLE[5] LOCATION_MASK_5	Error-location interrupt for syndrome polynomial 5
ELM_IRQSTATUS[4] LOC_VALID_4	ELM_IRQENABLE[4] LOCATION_MASK_4	Error-location interrupt for syndrome polynomial 4
ELM_IRQSTATUS[3] LOC_VALID_3	ELM_IRQENABLE[3] LOCATION_MASK_3	Error-location interrupt for syndrome polynomial 3
ELM_IRQSTATUS[2] LOC_VALID_2	ELM_IRQENABLE[2] LOCATION_MASK_2	Error-location interrupt for syndrome polynomial 2
ELM_IRQSTATUS[1] LOC_VALID_1	ELM_IRQENABLE[1] LOCATION_MASK_1	Error-location interrupt for syndrome polynomial 1
ELM_IRQSTATUS[0] LOC_VALID_0	ELM_IRQENABLE[0] LOCATION_MASK_0	Error-location interrupt for syndrome polynomial 0

15.5.3.4 Processing Initialization

ELM_LOCATION_CONFIG global setting parameters must be set before using the error-location engine. The ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL bit field defines the error-correction level used (4-,8-, or 16-bit error correction). The ELM_LOCATION_CONFIG[26:16] ECC_SIZE bit field defines the maximum buffer length beyond which the engine processing no longer looks for errors.

Software can choose to use the ELM in continuous mode or page mode. If all ELM_PAGE_CTRL[i] SECTOR_i bits (i is the syndrome polynomial number, where i = 0 to 7) are reset, continuous mode is used. In any other case, page mode is implicitly selected.

- Continuous mode: Each syndrome polynomial is processed independently. Results for a syndrome can be retrieved and acknowledged at any time, regardless of the status of the other seven processing contexts.

- Page mode: Syndrome polynomials are grouped into atomic entities: only one page can be processed at any given time, even if all eight contexts are not used for this page. Unused contexts are lost and cannot be affected to any other processing. The full page must be acknowledged and cleared before moving to the next page.

For completion interrupts to be generated correctly, all ELM_IRQENABLE[i] LOCATION_MASK_i bits (where i = 0 to 7) must be forced to 0 when in page mode, and set to 1 in continuous mode. Additionally, the ELM_IRQENABLE[8] PAGE_MASK bit must be set to 1 when in page mode.

Software initiates error-location processing by writing a syndrome polynomial into one of the eight possible register sets. Each of these register sets includes seven registers: ELM_SYNDROME_FRAGMENT_0_i to ELM_SYNDROME_FRAGMENT_6_i. The first six registers can be written in any order, but ELM_SYNDROME_FRAGMENT_6_i must be written last because it includes the validity bit, which instructs the ELM that this syndrome polynomial must be processed (the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit).

As soon as one validity bit is asserted (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x1, where i = 0 to 7), error-location processing can start for the corresponding syndrome polynomial. The associated ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i registers (where i = 0 to 7) are not reset. Software must not consider them until the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit is set.

15.5.3.5 Processing Sequence

While the error-location engine is busy processing one syndrome polynomial, further syndrome polynomials can be written. They are processed when the current processing completes.

The engine completes early when:

- No error is detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 1 and the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set to 0x0.
- Too many errors are detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 0 while the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set with the value output by the error-location engine. The reported number of errors is not ensured if ECC_CORRECTABLE is 0.

If the engine completes early, the associated error-location registers ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i (where i = 0 to 7) are not updated.

In all other cases, the engine goes through the entire error-location process. Each time an error location is found, it is logged in the associated ECC_ERROR_LOCATION bit field. The first error detected is logged in the ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION bit field; the second is logged in the ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION bit field, and so on.

Table 15-513 describes the ELM_LOCATION_STATUS_i value decoding.

Table 15-513. ELM_LOCATION_STATUS_i Value Decoding

ECC_CORRECTABLE Value	ECC_NB_ERRORS Value	Status	Number of Errors Detected	Action Required
1	0	OK	0	None
1	≠ 0	OK	ECC_NB_ERRORS	Correct the data buffer read based on the ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i results.
0	Any	Failed	Unknown	Software-dependent

15.5.3.6 Processing Completion

When the processing for a given syndrome polynomial completes, its ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit is reset. It must not be set again until the exit status registers, ELM_LOCATION_STATUS_i (where i = 0 to 7) for this processing are checked. Failure to comply with this rule leads to potential loss of the first polynomial process data output.

The error-location engine signals the process completion to the ELM. When this event is detected, the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit (where i = 0 to 7) is set. The processing exit status is available from the associated ELM_LOCATION_STATUS_i register, and error locations are stored in order in the ECC_ERROR_LOCATION bit fields. Software must read only valid error-location registers based on the number of errors detected and located.

Immediately after the error-location engine completes, a new syndrome polynomial can be processed, if any is available, as reported by the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit, depending on the configured error-correction level. If several syndrome polynomials are available, a round-robin arbitration is used to select one for processing.

In continuous mode (that is, all bits in ELM_PAGE_CTRL are reset), an interrupt is triggered whenever a ELM_IRQSTATUS[i] LOC_VALID_i bit is asserted. Software must read the ELM_IRQSTATUS register to determine which polynomial is processed and retrieve the exit status and error locations (ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i). When done, software must clear the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit by setting it to 1. Other status bits must be set to 0 so that other interrupts are not unintentionally cleared. When using this mode, the ELM_IRQSTATUS[8] PAGE_VALID interrupt is never triggered.

In page mode, the module does not trigger interrupts for the processing completion of each polynomial because the ELM_IRQENABLE[i] LOCATION_MASK_i bits are cleared. A page is defined using the ELM_PAGE_CTRL register. Each SECTOR_i bit set means the corresponding polynomial i is part of the page processing. A page is fully processed when all tagged polynomials have been processed, as logged in the ELM_IRQSTATUS[i] LOC_VALID_i bits. The module triggers an ELM_IRQSTATUS[8] PAGE_VALID interrupt whenever it detects that the full page has been processed. To make sure the next page can be correctly processed, all status bits in the ELM_IRQSTATUS register must be cleared by using a single atomic-write access.

Note

Do not modify page setting parameters in the ELM_PAGE_CTRL register unless the engine is idle, no polynomial input is valid, and all interrupts have been cleared.

Because no polynomial-level interrupt is triggered in page mode, polynomials cleared in the ELM_PAGE_CTRL[i] SECTOR_i bits (where i = 0 to 7) are processed as usual, but are essentially ignored. Software must manually poll the ELM_IRQSTATUS bits to check for their status.

15.5.4 ELM Basic Programming Model

15.5.4.1 ELM Low-Level Programming Model

15.5.4.1.1 Processing Initialization

Table 15-514. ELM Processing Initialization

Step	Register/Bit Field/Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management.	ELM_SYSCONFIG[4:3] SIDLEMODE	Set value.
Defines the error-correction level used	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	Set value.
Defines the maximum buffer length	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	Set value.
Sets the ELM in continuous mode or page mode	ELM_PAGE_CTRL	Set value.
If continuous mode is used	All ELM_PAGE_CTRL[i] SECTOR_i (where i = 0 to 7)	0x0
Enables interrupt for syndrome polynomial i	ELM_IRQENABLE[i] LOCATION_MASK_i	0x1
else (page mode is used)	One syndrome polynomial i is set ELM_PAGE_CTRL[i] SECTOR_i (where i = 0 to 7)	0x1
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	All ELM_IRQENABLE[i] LOCATION_MASK_i = 0x0 and ELM_IRQENABLE[8] PAGE_MASK = 0x1	Set value.
endif		Set value.

Table 15-514. ELM Processing Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Set the input syndrome polynomial i.	ELM_SYNDROME_FRAGMENT_0_i	Set value.
	ELM_SYNDROME_FRAGMENT_1_i	Set value.
	ELM_SYNDROME_FRAGMENT_5_i	Set value.
	ELM_SYNDROME_FRAGMENT_6_i	Set value.
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID	0x1

15.5.4.1.2 Read Results

The engine goes through the entire error-location process and results can be read. Table 15-515 and Table 15-516 describe the processing completion for continuous and page modes, respectively.

Table 15-515. ELM Processing Completion for Continuous Mode

Step	Register/Bit Field/Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Read for which i the error-location process is complete.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1
if the process fails (too many errors)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
It is software dependant.		
else (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers. Software must correct errors in the data buffer.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	
endif		
Clear the corresponding i interrupt.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1

A new syndrome polynomial can be processed after the end of processing (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x0) and after the exit status register check (ELM_LOCATION_STATUS_i).

Table 15-516. ELM Processing Completion for Page Mode

Step	Register/Bit Field/Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
Repeat the following actions the necessary number of times. That is, once for each valid defined block in the page.		
Read the process exit status.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	
if the process fails (too many errors)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
It is software dependent.		
else (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	

Table 15-516. ELM Processing Completion for Page Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	
endif		
End Repeat		
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

15.5.4.1.3

Next page can be correctly processed after a page is fully processed, when all tagged polynomials have been processed (ELM_IRQSTATUS[i] LOC_VALID_i = 0x1 for all syndrome polynomials i used in the page).

15.5.4.2 Use Case: ELM Used in Continuous Mode

In this example, the ELM is programmed for an 8-bit error-correction capability in continuous mode (see Table 15-517). After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, a nonzero polynomial syndrome is reported from the GPMC (polynomial syndrome 0 is used in the ELM):

- P = 0x0A16ABE115E44F767BFB0D0980

Table 15-517. Use Case: Continuous Mode

Step	Register/Bit Field/Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 8 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x1
Defines the maximum buffer length: 528 bytes (2 × 528 = 1056)	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in continuous mode	ELM_PAGE_CTRL	0
Enables interrupt for syndrome polynomial 0	ELM_IRQENABLE[0] LOCATION_MASK_0	0x1
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (where i = 0)	0xFB0D0980
	ELM_SYNDROME_FRAGMENT_1_i (where i = 0)	0xE44F767B
	ELM_SYNDROME_FRAGMENT_2_i (where i = 0)	0x16ABE115
	ELM_SYNDROME_FRAGMENT_3_i (where i = 0)	0x0000000A
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 0)	0x1
Wait until process is complete for syndrome polynomial 0: IRQ_ELM is generated or poll the status register.		
Read that error-location process is complete for syndrome polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1
Read the process exit status: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 0)	0x1
Read the number of errors: Four errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 0)	0x4
Read the error-location bit addresses for syndrome polynomial 0 of the first four registers: Errors are located in the data buffer at decimal addresses 431, 1062, 1909, 3452.	ELM_ERROR_LOCATION_0_i (where i = 0)	0x1AF
	ELM_ERROR_LOCATION_1_i (where i = 0)	0x426
	ELM_ERROR_LOCATION_2_i (where i = 0)	0x775
	ELM_ERROR_LOCATION_3_i (where i = 0)	0xD7C

Table 15-517. Use Case: Continuous Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Clear the corresponding interrupt for polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1

The NAND flash data in the sector are seen as a polynomial of degree 4223 (number of bits in a 528 byte buffer minus 1), with each data bit being a coefficient in the polynomial. When reading from a NAND flash using the GPMC module, computation of the polynomial syndrome assumes that the first NAND word read at address 0x0 contains the highest-order coefficient in the message. Furthermore, in the 16-bit NAND word, bits are ordered from bit 7 to bit 0, and then from bit 15 to bit 8. Based on this convention, an address table of the data buffer can be built. NAND memory addresses in Table 15-518 are given in decimal format.

Table 15-518. 16-Bit NAND Sector Buffer Address Map

NAND Memory Address	Message Bit Addresses in Memory Word															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	4215	4214	4213	4212	4211	4210	4209	4208	4223	4222	4221	4220	4219	4218	4217	4216
1	4175	4174	4173	4172	4171	4170	4169	4168	4183	4182	4181	4180	4179	4178	4177	4176
...																
47	3463	3462	3461	3460	3459	3458	3457	3456	3471	3470	3469	3468	3467	3466	3465	3464
48	3447	3446	3445	3444	3443	3442	3441	3440	3455	3454	3453	3452	3451	3450	3449	3448
49	3431	3430	3429	3428	3427	3426	3425	3424	3439	3438	3437	3436	3435	3434	3433	3432
50	3415	3414	3413	3412	3411	3410	3409	3408	3423	3422	3421	3420	3419	3418	3417	3416
...																
255	135	134	133	132	131	130	129	128	143	142	141	140	139	138	137	136
256	119	118	117	116	115	114	113	112	127	126	125	124	123	122	121	120
257	103	102	101	100	99	98	97	96	111	110	109	108	107	106	105	104
258	87	86	85	84	83	82	81	80	95	94	93	92	91	90	89	88
259	71	70	69	68	67	66	65	64	79	78	77	76	75	74	73	72
260	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
261	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40
262	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
263	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

The table can now be used to determine which bits in the buffer were incorrect and must be flipped. In this example, the first bit to be flipped is bit 4 from the 49th byte read from memory. It is up to the processor to correctly map this word to the copied buffer and flip this bit. The same process must be repeated for all detected errors.

15.5.4.3 Use Case: ELM Used in Page Mode

In this example, the ELM module is programmed for an 16-bit error-correction capability in page mode (see Table 15-519). After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, four non-zero polynomial syndromes are reported from the GPMC (polynomial syndrome 0, 1, 2, and 3 are used in the ELM):

- P0 = 0xE8B0 12ADDB5A318E05BE B0693DB28330B5CC A329AA05E0B718EF
- P1 = 0xBAD0 49A0D932C22E6669 0948DF08BE093336 79C6BA10E5F935EB
- P2 = 0x69D9 B86ABCD5EC3697FA A6498FEE54556EA0 1579EF7D60BA3189
- P3 = 0x0

Table 15-519. Use Case: Page Mode

Step	Register/Bit Field/Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1

Table 15-519. Use Case: Page Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 16 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x2
Defines the maximum buffer length: 528 bytes	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in page mode (four blocks in a page)	ELM_PAGE_CTRL[0] SECTOR_0	0x1
	ELM_PAGE_CTRL[1] SECTOR_1	0x1
	ELM_PAGE_CTRL[2] SECTOR_2	0x1
	ELM_PAGE_CTRL[3] SECTOR_3	0x1
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	ELM_IRQENABLE	0x100
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (where i = 0)	0xE0B718EF
	ELM_SYNDROME_FRAGMENT_1_i (where i = 0)	0xA329AA05
	ELM_SYNDROME_FRAGMENT_2_i (where i = 0)	0x8330B5CC
	ELM_SYNDROME_FRAGMENT_3_i (where i = 0)	0xB0693DB2
	ELM_SYNDROME_FRAGMENT_4_i (where i = 0)	0x318E05BE
	ELM_SYNDROME_FRAGMENT_5_i (where i = 0)	0x12ADDB5A
	ELM_SYNDROME_FRAGMENT_6_i (where i = 0)	0xE8B0
Set the input syndrome polynomial 1.	ELM_SYNDROME_FRAGMENT_0_i (where i = 1)	0xE5F935EB
	ELM_SYNDROME_FRAGMENT_1_i (where i = 1)	0x79C6BA10
	ELM_SYNDROME_FRAGMENT_2_i ((where i = 1)	0xBE093336
	ELM_SYNDROME_FRAGMENT_3_i (where i = 1)	0x0948DF08
	ELM_SYNDROME_FRAGMENT_4_i (where i = 1)	0xC22E6669
	ELM_SYNDROME_FRAGMENT_5_i (where i = 1)	0x49A0D932
	ELM_SYNDROME_FRAGMENT_6_i (where i = 1)	0xBAD0
Set the input syndrome polynomial 2.	ELM_SYNDROME_FRAGMENT_0_i (where i = 2)	0x60BA3189
	ELM_SYNDROME_FRAGMENT_1_i (where i = 2)	0x1579EF7D
	ELM_SYNDROME_FRAGMENT_2_i (where i = 2)	0x54556EA0
	ELM_SYNDROME_FRAGMENT_3_i (where i = 2)	0xA6498FEE
	ELM_SYNDROME_FRAGMENT_4_i (where i = 2)	0xEC3697FA
	ELM_SYNDROME_FRAGMENT_5_i (where i = 2)	0xB86ABCD5
	ELM_SYNDROME_FRAGMENT_6_i (where i = 2)	0x69D9
Set the input syndrome polynomial 3.	ELM_SYNDROME_FRAGMENT_0_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_1_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_2_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_3_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_4_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_5_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_6_i (where i = 3)	0x0
Initiates the computation process for syndrome polynomial 0	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 0)	0x1
Initiates the computation process for syndrome polynomial 1	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 1)	0x1
Initiates the computation process for syndrome polynomial 2	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 2)	0x1
Initiates the computation process for syndrome polynomial 3	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 3)	0x1

Table 15-519. Use Case: Page Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Wait until process is complete for syndrome polynomial 0, 1, 2, and 3: Wait until the ELM_IRQ interrupt is generated or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
Read the process exit status for syndrome polynomial 0: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 0)	0x1
Read the process exit status for syndrome polynomial 1: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 1)	0x1
Read the process exit status for syndrome polynomial 2: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 2)	0x1
Read the process exit status for syndrome polynomial 3: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 3)	0x1
Read the number of errors for syndrome polynomial 0: four errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 0)	0x4
Read the number of errors for syndrome polynomial 1: two errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 1)	0x2
Read the number of errors for syndrome polynomial 2: one error detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 2)	0x1
Read the number of errors for syndrome polynomial 3: no errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 3)	0x0
Read the error-location bit addresses for syndrome polynomial 0 of the first four registers:	ELM_ERROR_LOCATION_0_i (where i = 0)	0x1FE
	ELM_ERROR_LOCATION_1_i (where i = 0)	0x617
	ELM_ERROR_LOCATION_2_i (where i = 0)	0x650
	ELM_ERROR_LOCATION_3_i (where i = 0)	0xA83
Read the error-location bit addresses for syndrome polynomial 1 of the first two registers:	ELM_ERROR_LOCATION_0_i (where i = 1)	0x4
	ELM_ERROR_LOCATION_1_i (where i = 1)	0x1036
Read the errors location bit addresses for syndrome polynomial 2 of the first registers:	ELM_ERROR_LOCATION_0_i (where i = 1)	0x3E8
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

15.5.5 ELM Register Manual

15.5.5.1 ELM Instance Summary

Table 15-520summarizes the ELM instance.

Table 15-520. ELM Instance Summary

Module Name	Base Address	Size
ELM	0x4807 8000	4 KiB

15.5.5.2 ELM Registers

15.5.5.2.1 ELM Register Summary

Table 15-521 summarizes the ELM register mapping.

Table 15-521. ELM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address ELM
ELM_REVISION	R	32	0x0000 0000	0x4807 8000
ELM_SYSCONFIG	RW	32	0x0000 0010	0x4807 8010
ELM_SYSSTATUS	R	32	0x0000 0014	0x4807 8014
ELM_IRQSTATUS	RW	32	0x0000 0018	0x4807 8018
ELM_IRQENABLE	RW	32	0x0000 001C	0x4807 801C
ELM_LOCATION_CONFIG	RW	32	0x0000 0020	0x4807 8020
ELM_PAGE_CTRL	RW	32	0x0000 0080	0x4807 8080
ELM_SYNDROME_FRAGMENT_0_i (1)	RW	32	0x0000 0400 + (0x40 * i)	0x4807 8400 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_1_i (1)	RW	32	0x0000 0404 + (0x40 * i)	0x4807 8404 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_2_i (1)	RW	32	0x0000 0408 + (0x40 * i)	0x4807 8408 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_3_i (1)	RW	32	0x0000 040C + (0x40 * i)	0x4807 840C + (0x40 * i)
ELM_SYNDROME_FRAGMENT_4_i (1)	RW	32	0x0000 0410 + (0x40 * i)	0x4807 8410 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_5_i (1)	RW	32	0x0000 0414 + (0x40 * i)	0x4807 8414 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_6_i (1)	RW	32	0x0000 0418 + (0x40 * i)	0x4807 8418 + (0x40 * i)
ELM_LOCATION_STATUS_i (1)	R	32	0x0000 0800 + (0x100 * i)	0x4807 8800 + (0x100 * i)
ELM_ERROR_LOCATION_0_i (1)	R	32	0x0000 0880 + (0x100 * i)	0x4807 8880 + (0x100 * i)
ELM_ERROR_LOCATION_1_i (1)	R	32	0x0000 0884 + (0x100 * i)	0x4807 8884 + (0x100 * i)
ELM_ERROR_LOCATION_2_i (1)	R	32	0x0000 0888 + (0x100 * i)	0x4807 8888 + (0x100 * i)
ELM_ERROR_LOCATION_3_i (1)	R	32	0x0000 088C + (0x100 * i)	0x4807 888C + (0x100 * i)
ELM_ERROR_LOCATION_4_i (1)	R	32	0x0000 0890 + (0x100 * i)	0x4807 8890 + (0x100 * i)
ELM_ERROR_LOCATION_5_i (1)	R	32	0x0000 0894 + (0x100 * i)	0x4807 8894 + (0x100 * i)
ELM_ERROR_LOCATION_6_i (1)	R	32	0x0000 0898 + (0x100 * i)	0x4807 8898 + (0x100 * i)
ELM_ERROR_LOCATION_7_i (1)	R	32	0x0000 089C + (0x100 * i)	0x4807 889C + (0x100 * i)
ELM_ERROR_LOCATION_8_i (1)	R	32	0x0000 08A0 + (0x100 * i)	0x4807 88A0 + (0x100 * i)
ELM_ERROR_LOCATION_9_i (1)	R	32	0x0000 08A4 + (0x100 * i)	0x4807 88A4 + (0x100 * i)
ELM_ERROR_LOCATION_10_i (1)	R	32	0x0000 08A8 + (0x100 * i)	0x4807 88A8 + (0x100 * i)
ELM_ERROR_LOCATION_11_i (1)	R	32	0x0000 08AC + (0x100 * i)	0x4807 88AC + (0x100 * i)
ELM_ERROR_LOCATION_12_i (1)	R	32	0x0000 08B0 + (0x100 * i)	0x4807 88B0 + (0x100 * i)
ELM_ERROR_LOCATION_13_i (1)	R	32	0x0000 08B4 + (0x100 * i)	0x4807 88B4 + (0x100 * i)
ELM_ERROR_LOCATION_14_i (1)	R	32	0x0000 08B8 + (0x100 * i)	0x4807 88B8 + (0x100 * i)
ELM_ERROR_LOCATION_15_i (1)	R	32	0x0000 08BC + (0x100 * i)	0x4807 88BC + (0x100 * i)

(1) i = 0 to 7 for ELM

15.5.5.2.2 ELM Register Description

Table 15-522 through Table 15-552 describe the individual ELM registers.

Table 15-522. ELM_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4807 8000	**Instance**	ELM
Description	This register contains the IP revision code. (A write or reset of to this register has no effect.)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision (TI internal data)	R	0x-

Table 15-523. ELM_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4807 8010	**Instance**	ELM
Description	This register allows controlling various parameters of the OCP interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLOCKACTIVITYOCP	RESERVED			SIDLEMODE		RESERVED	SOFTRESET	AUTOGATING

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	CLOCKACTIVITYOCP	OCP clock activity when module is in IDLE mode (during wake-up mode period) 0x0: OCP clock can be switched off. 0x1: OCP clock is maintained during wake-up period.	RW	0
7:5	RESERVED	Reserved	R	0x0
4:3	SIDLEMODE	Slave interface power management (IDLE req/ack control) 0x0: Force-idle. IDLE request is acknowledged unconditionally and immediately (Default *Dumb* mode for safety) 0x1: No-idle. IDLE request is never acknowledged. 0x2: Smart-idle. The acknowledgment to an IDLE request is given based on the internal activity. 0x3: Reserved - do not use	RW	0x2
2	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
1	SOFTRESET	Module software reset This bit is automatically reset by hardware (during reads, it always returns 0). It has same effect as the OCP hardware reset. 0x0: Normal mode 0x1: Start soft reset sequence.	RW	0
0	AUTOGATING	Internal OCP clock gating strategy (no module visible effect other than saving power) 0x0: OCP clock is free-running. 0x1: Automatic internal OCP clock gating strategy is applied based on the OCP interface activity.	RW	1

Table 15-524. ELM_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x4807 8014	**Instance**	ELM
Description	Internal reset monitoring (OCP domain) Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Internal reset monitoring (OCP domain) Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1. Read 0x0: Reset is ongoing. Read 0x1: Reset is done (completed).	R	1

Table 15-525. ELM_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	0x4807 8018	**Instance**	ELM
Description	Interrupt status. This register doubles as a status register for the error-location processes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PAGE_VALID	LOC_VALID_7	LOC_VALID_6	LOC_VALID_5	LOC_VALID_4	LOC_VALID_3	LOC_VALID_2	LOC_VALID_1	LOC_VALID_0

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000

Bits	Field Name	Description	Type	Reset
8	PAGE_VALID	Error-location status for a full page, based on the mask definition Read 0x0: Error locations invalid for all polynomials enabled in the ECC_INTERRUPT_MASK register Read 0x1: All error locations valid Write 0x0: No effect Write 0x1: Clear interrupt	RW	0
7	LOC_VALID_7	Error-location status for syndrome polynomial 7 Read 0x0: No syndrome processed or process in progress Read 0x1: Error-location process completed Write 0x0: No effect Write 0x1: Clear interrupt	RW W1toClr	0
6	LOC_VALID_6	Error-location status for syndrome polynomial 6	RW W1toClr	0
5	LOC_VALID_5	Error-location status for syndrome polynomial 5	RW W1toClr	0
4	LOC_VALID_4	Error-location status for syndrome polynomial 4	RW W1toClr	0
3	LOC_VALID_3	Error-location status for syndrome polynomial 3	RW W1toClr	0
2	LOC_VALID_2	Error-location status for syndrome polynomial 2	RW W1toClr	0
1	LOC_VALID_1	Error-location status for syndrome polynomial 1	RW W1toClr	0
0	LOC_VALID_0	Error-location status for syndrome polynomial 0	RW W1toClr	0

Table 15-526. ELM_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	0x4807 801C	**Instance**	ELM
Description	Interrupt enable		
Type	RW		

31–9	8	7	6	5	4	3	2	1	0
RESERVED	PAGE_MASK	LOCATION_MASK_7	LOCATION_MASK_6	LOCATION_MASK_5	LOCATION_MASK_4	LOCATION_MASK_3	LOCATION_MASK_2	LOCATION_MASK_1	LOCATION_MASK_0

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	PAGE_MASK	Page interrupt mask bit 0: Disable interrupt 1: Enable interrupt	RW	0
7	LOCATION_MASK_7	Error-location interrupt mask bit for syndrome polynomial 7	RW	0
6	LOCATION_MASK_6	Error-location interrupt mask bit for syndrome polynomial 6	RW	0
5	LOCATION_MASK_5	Error-location interrupt mask bit for syndrome polynomial 5	RW	0
4	LOCATION_MASK_4	Error-location interrupt mask bit for syndrome polynomial 4	RW	0
3	LOCATION_MASK_3	Error-location interrupt mask bit for syndrome polynomial 3	RW	0
2	LOCATION_MASK_2	Error-location interrupt mask bit for syndrome polynomial 2	RW	0

Bits	Field Name	Description	Type	Reset
1	LOCATION_MASK_1	Error-location interrupt mask bit for syndrome polynomial 1	RW	0
0	LOCATION_MASK_0	Error-location interrupt mask bit for syndrome polynomial 0 0: Disable interrupt 1: Enable interrupt	RW	0

Table 15-527. ELM_LOCATION_CONFIG

Address Offset	0x0000 0020		
Physical Address	0x4807 8020	**Instance**	ELM
Description	ECC algorithm parameters		
Type	RW		

31:27	26:16	15:2	1:0
RESERVED	ECC_SIZE	RESERVED	ECC_BCH_LEVEL

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26:16	ECC_SIZE	Maximum size of the buffers for which the error-location engine is used, in number of nibbles (4-bit entities)	RW	0x000
15:2	RESERVED	Reserved	R	0x0000
1:0	ECC_BCH_LEVEL	Error correction level 0x0: 4 bits 0x1: 8 bits 0x2: 16 bits 0x3: Reserved	RW	0x0

Table 15-528. ELM_PAGE_CTRL

Address Offset	0x0000 0080		
Physical Address	0x4807 8080	**Instance**	ELM
Description	Page definition		
Type	RW		

31:8	7	6	5	4	3	2	1	0
RESERVED	SECTOR_7	SECTOR_6	SECTOR_5	SECTOR_4	SECTOR_3	SECTOR_2	SECTOR_1	SECTOR_0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7	SECTOR_7	Set to 1 if syndrome polynomial 7 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
6	SECTOR_6	Set to 1 if syndrome polynomial 6 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
5	SECTOR_5	Set to 1 if syndrome polynomial 5 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
4	SECTOR_4	Set to 1 if syndrome polynomial 4 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
3	SECTOR_3	Set to 1 if syndrome polynomial 3 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
2	SECTOR_2	Set to 1 if syndrome polynomial 2 is part of the page in page mode. Must be 0 in continuous mode.	RW	0

Bits	Field Name	Description	Type	Reset
1	SECTOR_1	Set to 1 if syndrome polynomial 1 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
0	SECTOR_0	Set to 1 if syndrome polynomial 0 is part of the page in page mode. Must be 0 in continuous mode.	RW	0

Table 15-529. ELM_SYNDROME_FRAGMENT_0_i

Address Offset	0x0000 0400 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8400 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 0 to 31.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_0																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_0	Syndrome bits 0 to 31	RW	0x0000 0000

Table 15-530. ELM_SYNDROME_FRAGMENT_1_i

Address Offset	0x0000 0404 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8404 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 32 to 63.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_1																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_1	Syndrome bits 32 to 63	RW	0x0000 0000

Table 15-531. ELM_SYNDROME_FRAGMENT_2_i

Address Offset	0x0000 0408 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8408 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 64 to 95.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_2																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_2	Syndrome bits 64 to 95	RW	0x0000 0000

Table 15-532. ELM_SYNDROME_FRAGMENT_3_i

Address Offset	0x0000 040C + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 840C + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 96 to 127		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_3																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_3	Syndrome bits 96 to 127	RW	0x0000 0000

Table 15-533. ELM_SYNDROME_FRAGMENT_4_i

Address Offset	0x0000 0410 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8410 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 128 to 159.		
Type	RW		

31–0
SYNDROME_4

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_4	Syndrome bits 128 to 159	RW	0x0000 0000

Table 15-534. ELM_SYNDROME_FRAGMENT_5_i

Address Offset	0x0000 0414 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8414 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 160 to 191.		
Type	RW		

31–0
SYNDROME_5

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_5	Syndrome bits 160 to 191	RW	0x0000 0000

Table 15-535. ELM_SYNDROME_FRAGMENT_6_i

Address Offset	0x0000 0418 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8418 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 192 to 207.		
Type	RW		

31–17	16	15–0
RESERVED	SYNDROME_VALID	SYNDROME_6

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000
16	SYNDROME_VALID	Syndrome valid bit 0x0: This syndrome polynomial must not be processed. 0x1: This syndrome polynomial must be processed.	RW	0
15:0	SYNDROME_6	Syndrome bits 192 to 207	RW	0x0000

Table 15-536. ELM_LOCATION_STATUS_i

Address Offset	0x0000 0800 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8800 + (0x100 * i)	**Instance**	ELM

Table 15-536. ELM_LOCATION_STATUS_i (continued)

Description	Exit status for the syndrome polynomial processing
Type	R

31–9	8	7–5	4–0
RESERVED	ECC_CORRECTABLE	RESERVED	ECC_NB_ERRORS

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	ECC_CORRECTABLE	Error-location process exit status 0x0: ECC error-location process failed. Number of errors and error locations are invalid. 0x1: All errors were successfully located. Number of errors and error locations are valid.	R	0
7:5	RESERVED	Reserved	R	0x0
4:0	ECC_NB_ERRORS	Number of errors detected and located	R	0x00

Table 15-537. ELM_ERROR_LOCATION_0_i

Address Offset	0x0000 0880 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8880 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-538. ELM_ERROR_LOCATION_1_i

Address Offset	0x0000 0884 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8884 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-539. ELM_ERROR_LOCATION_2_i

Address Offset	0x0000 0888 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8888 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		

Table 15-539. ELM_ERROR_LOCATION_2_i (continued)

Type	R

31:13	12:0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-540. ELM_ERROR_LOCATION_3_i

Address Offset	0x0000 088C + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 888C + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31:13	12:0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-541. ELM_ERROR_LOCATION_4_i

Address Offset	0x0000 0890 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8890 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31:13	12:0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-542. ELM_ERROR_LOCATION_5_i

Address Offset	0x0000 0894 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8894 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31:13	12:0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-543. ELM_ERROR_LOCATION_6_i

Address Offset	0x0000 0898 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8898 + (0x100 * i)	**Instance**	ELM

Table 15-543. ELM_ERROR_LOCATION_6_i (continued)

Description	Error-location register
Type	R

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-544. ELM_ERROR_LOCATION_7_i

Address Offset	0x0000 089C + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 889C + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-545. ELM_ERROR_LOCATION_8_i

Address Offset	0x0000 08A0 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88A0 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-546. ELM_ERROR_LOCATION_9_i

Address Offset	0x0000 08A4 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88A4 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-547. ELM_ERROR_LOCATION_10_i

Address Offset	0x0000 08A8 + (0x100 * i)	**Index**	i = 0 to 7

Table 15-547. ELM_ERROR_LOCATION_10_i (continued)

Physical Address	0x4807 88A8 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			ECC_ERROR_LOCATION												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-548. ELM_ERROR_LOCATION_11_i

Address Offset	0x0000 08AC + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88AC + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			ECC_ERROR_LOCATION												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-549. ELM_ERROR_LOCATION_12_i

Address Offset	0x0000 08B0 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88B0 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			ECC_ERROR_LOCATION												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-550. ELM_ERROR_LOCATION_13_i

Address Offset	0x0000 08B4 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88B4 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			ECC_ERROR_LOCATION												

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-551. ELM_ERROR_LOCATION_14_i

Address Offset	0x0000 08B8 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88B8 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 15-552. ELM_ERROR_LOCATION_15_i

Address Offset	0x0000 08BC + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88BC + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31–13	12–0
RESERVED	ECC_ERROR_LOCATION

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

15.6 On-Chip Memory (OCM) Subsystem

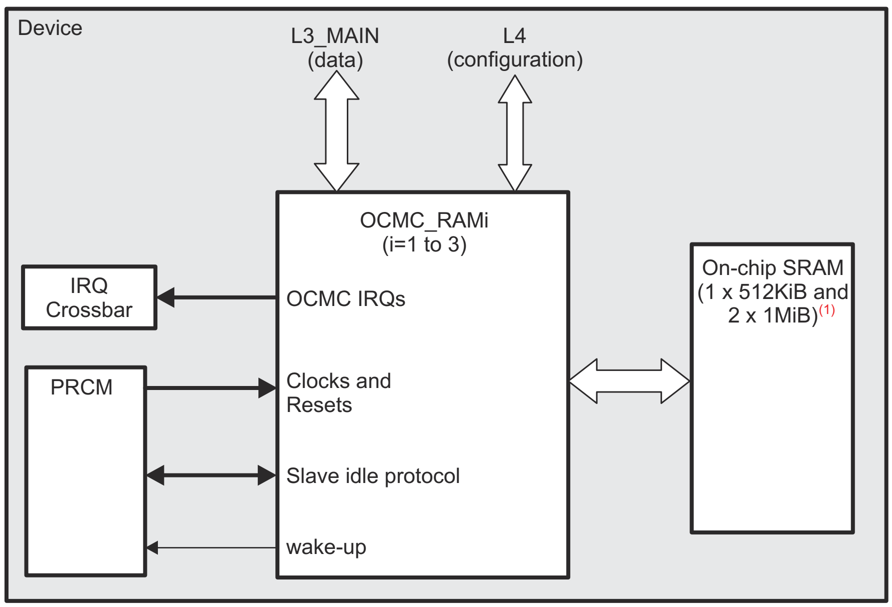
15.6.1 OCM Subsystem Overview

The OCM subsystem consists of three OCM Controllers (OCMC) that are associated with the on-chip RAM. The OCM controllers are also introduced in Section 15.1.6, *OCM Overview* of Section 15.1, *Memory Subsystem Overview*.

The RAM associated controllers are as follows:

- OCMC_RAM1 with 512 KiB of dedicated memory space.
- OCMC_RAM2 with 1024 KiB of dedicated memory space.
- OCMC_RAM3 with 1024 KiB of dedicated memory space.

Figure 15-106 shows the three OCMC_RAM controllers.



(1) Presence of RAM bank is device part number dependent. For more information, see the device Data Manual.

Figure 15-106. OCMC_RAMi (i = 1 to 3) Overview

15.6.2 OCM Subsystem Integration

Figure 15-107 shows the integration of the OCM Subsystem in the device.

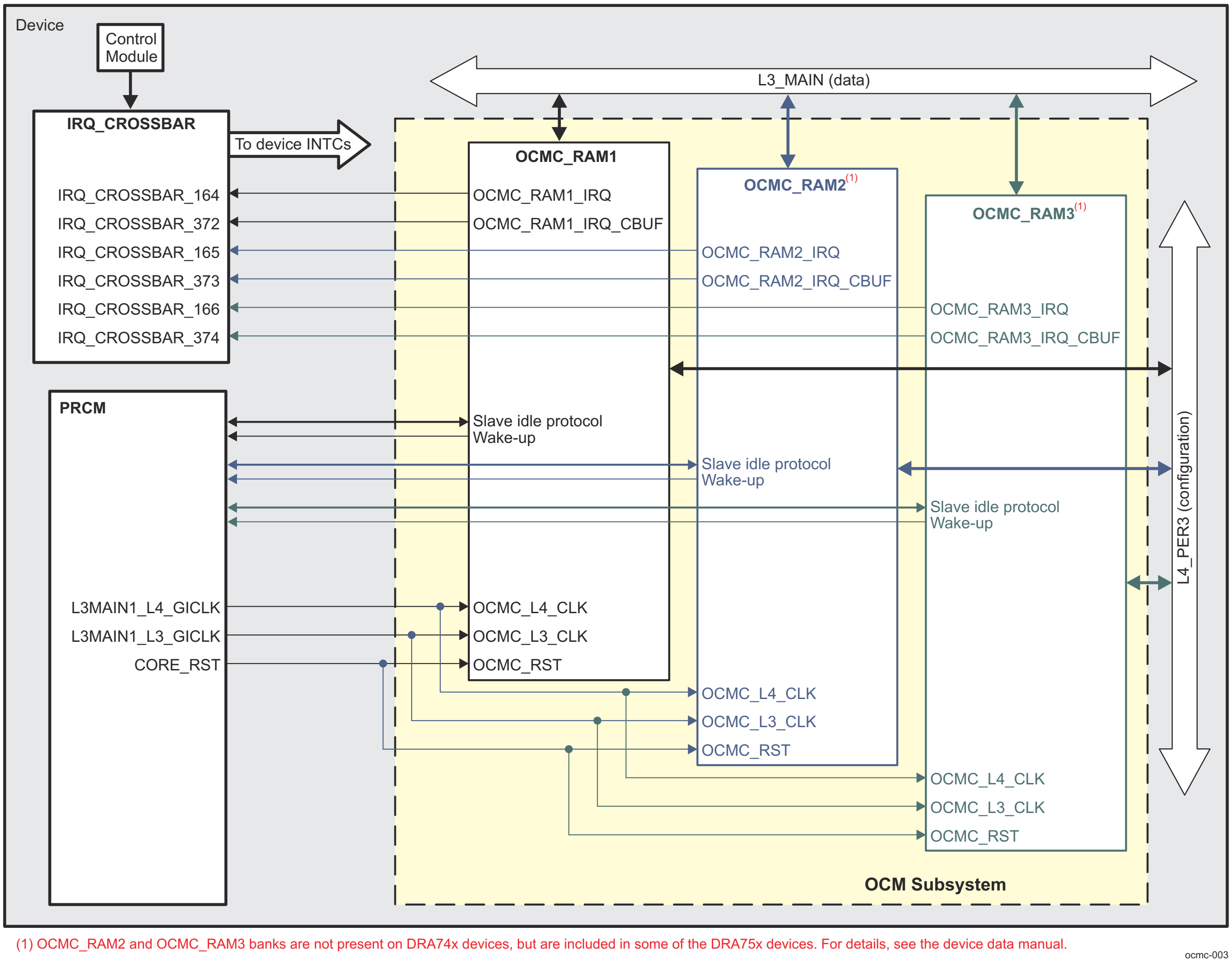


Figure 15-107. OCM Subsystem Integration

Table 15-553 through Table 15-555 summarize the integration of the OCM Subsystem in the device.

Table 15-553. OCM Subsystem Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
OCMC_RAM1	PD_COREAON	Yes	L3_MAIN
			L4_PER3
OCMC_RAM2	PD_COREAON	Yes	L3_MAIN
			L4_PER3
OCMC_RAM3	PD_COREAON	Yes	L3_MAIN
			L4_PER3

Table 15-554. OCM Subsystem Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description

Table 15-554. OCM Subsystem Clocks and Resets (continued)

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
OCMC_RAM1	OCMC_L3_CLK	L3MAIN1_L3_GICLK	PRCM	Clock used to drive and receive data over the L3 data bus. This is also the processing clock of the OCM Controller and the SRAM. With this clock all internal data transfers are clocked.
	OCMC_L4_CLK	L3MAIN1_L4_GICLK	PRCM	Clock used to drive and receive data over the L4_PER3 configuration bus. This clock should run at half the OCMC_L3_CLK clock rate.
OCMC_RAM2	OCMC_L3_CLK	L3MAIN1_L3_GICLK	PRCM	Clock used to drive and receive data over the L3 data bus. This is also the processing clock of the OCM Controller and the SRAM. With this clock all internal data transfers are clocked.
	OCMC_L4_CLK	L3MAIN1_L4_GICLK	PRCM	Clock used to drive and receive data over the L4_PER3 configuration bus. This clock should run at half the OCMC_L3_CLK clock rate.
OCMC_RAM3	OCMC_L3_CLK	L3MAIN1_L3_GICLK	PRCM	Clock used to drive and receive data over the L3 data bus. This is also the processing clock of the OCM Controller and the SRAM. With this clock all internal data transfers are clocked.
	OCMC_L4_CLK	L3MAIN1_L4_GICLK	PRCM	Clock used to drive and receive data over the L4_PER3 configuration bus. This clock should run at half the OCMC_L3_CLK clock rate.

Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
OCMC_RAM1	OCMC_RST	CORE_RST	PRCM	Reset signal for the OCM Subsystem
OCMC_RAM2	OCMC_RST	CORE_RST	PRCM	
OCMC_RAM3	OCMC_RST	CORE_RST	PRCM	

Table 15-555. OCM Subsystem Hardware Requests

Interrupt Requests

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
OCMC_RAM1	OCMC_RAM1_IRQ	IRQ_CROSSBAR_164	-	First OCMC_RAM1 interrupt request. This IRQ source signal is not mapped by default to any device INTC.
	OCMC_RAM1_IRQ_CBUF	IRQ_CROSSBAR_372	-	Second OCMC_RAM1 interrupt request. This IRQ source signal is not mapped by default to any device INTC.
OCMC_RAM2	OCMC_RAM2_IRQ	IRQ_CROSSBAR_165	-	First OCMC_RAM2 interrupt request. This IRQ source signal is not mapped by default to any device INTC.
	OCMC_RAM2_IRQ_CBUF	IRQ_CROSSBAR_373	-	Second OCMC_RAM2 interrupt request. This IRQ source signal is not mapped by default to any device INTC.

Table 15-555. OCM Subsystem Hardware Requests (continued)

OCMC_RAM3	OCMC_RAM3_IRQ	IRQ_CROSSBAR_166	-	First OCMC_RAM3 interrupt request. This IRQ source signal is not mapped by default to any device INTC.
	OCMC_RAM3_IRQ_CBUF	IRQ_CROSSBAR_374	-	Second OCMC_RAM3 interrupt request. This IRQ source signal is not mapped by default to any device INTC.

Note

The “**Default Mapping**” column in Table 15-555 *OCM Subsystem Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For description of the interrupt sources, see Section 15.6.3.4, *Interrupt Requests*.

15.6.3 OCM Subsystem Functional Desctiption

15.6.3.1 Block Diagram

Figure 15-108 shows the OCMC block diagram.

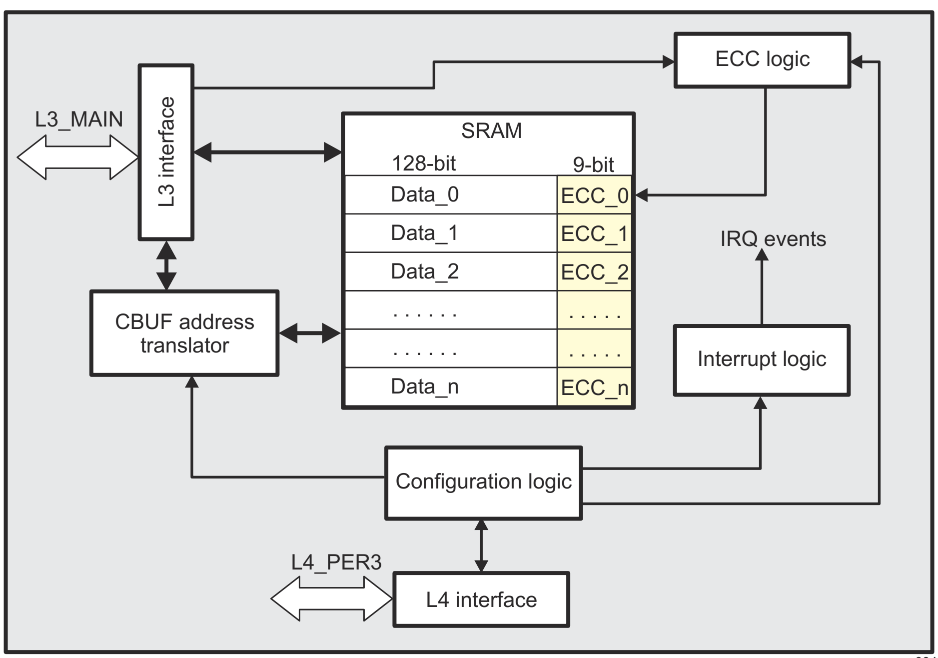


Figure 15-108. OCMC Block Diagram

The module is composed by the following main blocks:

- L3 interface used for data transactions
- L4 interface used for configuration

- CBUF address translator which converts the L3_MAIN VBUF addresses into SRAM addresses
- ECC logic to support single error correction and double error detection
- SRAM used for storring data and the corresponding for each 128-bit word ECC
- Interrupt logic used for generating interrupt requests
- Configuration logic in which reside the registers used for configuring the OCM controller operation modes

15.6.3.2 Resets

The OCMC_RST is the reset signal for the OCM controller which asynchronously resets the whole internal logic of the controller, including all configuration registers. It does not reset the SRAM. In addition, the OCMC_SYSCONFIG_RST[0] SW_RST bit provides a software way to reset the OCM controller. In this case all of its internal logic is reset except the configuration registers accessible through the L4_PER3.

15.6.3.3 Clock Management

The OCM controller complies with the PRCM slave idle protocol. The OCMC_L3_CLK and OCMC_L4_CLK clocks are gated based on the values loaded in the OCMC_SYSCONFIG_PM[3:2] IDLEMODE bit field.

15.6.3.4 Interrupt Requests

The OCM controller generates two interrupt requests, OCMC_RAMi_IRQ and OCMC_RAMi_IRQ_CBUF (i = 1, 2 and 3. i depends on the device part number).

The OCMC_RAMi_IRQ line is associated with the following registers:

- INTR0_STATUS_RAW_SET - interrupt raw status register
- INTR0_STATUS_ENABLED_CLEAR - interrupt status register
- INTR0_ENABLE_SET - interrupt enable register
- INTR0_ENABLE_CLEAR - interrupt disable register

The OCMC_RAMi_IRQ_CBUF line is associated with the following registers:

- INTR1_STATUS_RAW_SET - interrupt raw status register
- INTR1_STATUS_ENABLED_CLEAR - interrupt status register
- INTR1_ENABLE_SET - interrupt enable (event unmask) register
- INTR1_ENABLE_CLEAR - interrupt disable (event mask) register

Both the register groups previously described have identical bits. When for example, only one event occurs one or the two IRQ lines will be asserted depending on the masks applied to the INTR0_ENABLE_SET and INTR1_ENABLE_SET registers. When for example, a short frame detection event occurs and if both the INTR0_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND and INTR1_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND bits are set to 0x1, then both the OCMC_RAMi_IRQ and OCMC_RAMi_IRQ_CBUF lines are asserted. If only the INTR0_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND bit is set to 0x1, then only the OCMC_RAMi_IRQ line is asserted and the OCMC_RAMi_IRQ_CBUF line remains inactive (deasserted). Thus depending on the mask applied one IRQ line of the OCM controller can be associated only with CBUF events for example, and the other one IRQ line can be associated with ECC events. In other words, two unique interrupts can be provided.

Table 15-556 lists the interrupt events which can assert the two interrupt lines of the OCM controller.

The OCM controller asserts each of its interrupt lines only if the interrupts are enabled by setting to 0x1 the corresponding bits in the INTR0_ENABLE_SET/INTR1_ENABLE_SET register. These interrupts can be disabled by setting to 0x1 the corresponding bits in the INTR0_ENABLE_CLEAR/ INTR1_ENABLE_CLEAR register. After the interrupt has been serviced the corresponding status flag must be cleared by software. This is done by setting to 0x1 the corresponding bit in the INTR0_STATUS_ENABLED_CLEAR/INTR1_STATUS_ENABLED_CLEAR register which also clears the corresponding bit in the INTR0_STATUS_RAW_SET/INTR1_STATUS_RAW_SET register. The status flags in the INTR0_STATUS_RAW_SET/INTR1_STATUS_RAW_SET register are set even if the corresponding interrupt is disabled as opposed to those in the INTR0_STATUS_ENABLED_CLEAR/ INTR1_STATUS_ENABLED_CLEAR register, which are set only if the corresponding interrupt is enabled. An interrupt is also generated by the OCM controller, if certain bit in the INTR0_STATUS_RAW_SET/

INTR1_STATUS_RAW_SET register is set to 0x1 and the corresponding interrupt is enabled through the INTR0_ENABLE_SET/INTR1_ENABLE_SET register. This is useful when user software debugging is performed. Additionally, even if interrupts are not enabled, certain status bit in the INTR0_STATUS_RAW_SET/ INTR1_STATUS_RAW_SET register can be cleared by setting to 0x1 the corresponding bit in the INTR0_STATUS_ENABLED_CLEAR/INTR1_STATUS_ENABLED_CLEAR register.

For additional details regarding the CBUF related events, see Section 15.6.3.11, *CBUF Mode Error Handling*.

Table 15-556. OCM Subsystem Events

Event Flag	Event Mask	Description
INTR0_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND/ INTR1_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND and INTR0_STATUS_ENABLED_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND/ INTR1_STATUS_ENABLED_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND	INTR0_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND/ INTR1_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND and INTR0_ENABLE_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND/ INTR1_ENABLE_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND	This bit indicates a short frame detection. It is set if at least one of the bits in the STATUS_CBUF_SHORT_FRAME_DETECT register is set to 0x1.
INTR0_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND/ INTR1_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND	INTR0_ENABLE_SET[13] CBUF_UNDERFLOW_ERR_FOUND/ INTR1_ENABLE_SET[13] CBUF_UNDERFLOW_ERR_FOUND and INTR0_ENABLE_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND/ INTR1_ENABLE_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND	Indicates CBUF underflow detction. This bit is set if at least one of the bits in the STATUS_CBUF_UNDERFLOW register is set to 0x1.
INTR0_STATUS_RAW_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/ INTR1_STATUS_RAW_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND	INTR0_ENABLE_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/ INTR1_ENABLE_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND and INTR0_ENABLE_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/ INTR1_ENABLE_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND	Indicates Cbuf_Overflow_Wrap event
INTR0_STATUS_RAW_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND/ INTR1_STATUS_RAW_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND	INTR0_ENABLE_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND/ INTR1_ENABLE_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND and INTR0_ENABLE_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND/ INTR1_ENABLE_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND	Indicates Cbuf_Overflow_Mid event

Table 15-556. OCM Subsystem Events (continued)

Event Flag	Event Mask	Description
INTR0_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND/ INTR1_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND	INTR0_ENABLE_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND/ INTR1_ENABLE_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND and INTR0_ENABLE_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND/ INTR1_ENABLE_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND	This flag indicates that at least one CBUF read address is not incrementing in raster scan order(1).
INTR0_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND/ INTR1_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND	INTR0_ENABLE_SET[9] CBUF_VBUF_READ_START_ERR_FOUND/ INTR1_ENABLE_SET[9] CBUF_VBUF_READ_START_ERR_FOUND and INTR0_ENABLE_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND/ INTR1_ENABLE_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND	This flag indicates when at least one of the CBUF read accesses does not start at the VBUF start address(1).
INTR0_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND	INTR0_ENABLE_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND/ INTR1_ENABLE_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND and INTR0_ENABLE_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND/ INTR1_ENABLE_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND	This flag indicates when at least one of the CBUF read addresses is out of the CBUF address range(1).
INTR0_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/ INTR1_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND	INTR0_ENABLE_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/ INTR1_ENABLE_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND and INTR0_ENABLE_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/ INTR1_ENABLE_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND	This flag indicates that at least one CBUF write address is not incrementing in raster scan order(1).
INTR0_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND/ INTR1_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND	INTR0_ENABLE_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND/ INTR1_ENABLE_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND and INTR0_ENABLE_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND/ INTR1_ENABLE_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND	This flag indicates when at least one of the CBUF write accesses does not start at the VBUF start address(1).

Table 15-556. OCM Subsystem Events (continued)

Event Flag	Event Mask	Description
INTR0_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND	INTR0_ENABLE_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/ INTR1_ENABLE_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND and INTR0_ENABLE_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/ INTR1_ENABLE_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND	This flag indicates when at least one of the CBUF write addresses is out of the CBUF address range(1).
INTR0_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/ INTR1_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND	INTR0_ENABLE_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/ INTR1_ENABLE_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND and INTR0_ENABLE_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/ INTR1_ENABLE_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND	This flag indicates when a virtual address error is detected. This is a general interrupt event which is not associated with any specific CBUF(1).
INTR0_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[3] OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[3] OUT_OF_RANGE_ERR_FOUND	INTR0_ENABLE_SET[3] OUT_OF_RANGE_ERR_FOUND/ INTR1_ENABLE_SET[3] OUT_OF_RANGE_ERR_FOUND and INTR0_ENABLE_CLEAR[3] OUT_OF_RANGE_ERR_FOUND/ INTR1_ENABLE_CLEAR[3] OUT_OF_RANGE_ERR_FOUND	General interrupt for an access made with illegal VBUF address(1).
INTR0_STATUS_RAW_SET[2] ADDR_ERR_FOUND/ INTR1_STATUS_RAW_SET[2] ADDR_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[2] ADDR_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[2] ADDR_ERR_FOUND	INTR0_ENABLE_SET[2] ADDR_ERR_FOUND/ INTR1_ENABLE_SET[2] ADDR_ERR_FOUND and INTR0_ENABLE_CLEAR[2] ADDR_ERR_FOUND/ INTR1_ENABLE_CLEAR[2] ADDR_ERR_FOUND	This status flag is set when the value of the STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT bit field reaches the threshold counter value configured through the CFG_OCMC_ECC_ERROR[23:20] CFG_ADDR_ERR_CNT_MAX bit field. The status remains asserted until the counter is not cleared. This is done by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[2] CLEAR_ADDR_ERR_CNT bit. If the counter is not cleared another interrupt is re-issued.

Table 15-556. OCM Subsystem Events (continued)

Event Flag	Event Mask	Description
INTR0_STATUS_RAW_SET[1] DED_ERR_FOUND/ INTR1_STATUS_RAW_SET[1] DED_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[1] DED_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[1] DED_ERR_FOUND	INTR0_ENABLE_SET[1] DED_ERR_FOUND/ INTR1_ENABLE_SET[1] DED_ERR_FOUND and INTR0_ENABLE_CLEAR[1] DED_ERR_FOUND/ INTR1_ENABLE_CLEAR[1] DED_ERR_FOUND	This status flag is set when the value of the STATUS_ERROR_CNT[19:16] DED_ERROR_CNT bit field reaches the threshold counter value configured through the CFG_OCMC_ECC_ERROR[19:16] CFG_DED_CNT_MAX bit field. The status remains asserted until the counter is not cleared. This is done by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[1] CLEAR_DED_ERR_CNT bit. If the counter is not cleared another interrupt is re-issued.
INTR0_STATUS_RAW_SET[0] SEC_ERR_FOUND/ INTR1_STATUS_RAW_SET[0] SEC_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[0] SEC_ERR_FOUND/ INTR1_STATUS_ENABLED_CLEAR[0] SEC_ERR_FOUND	INTR0_ENABLE_SET[0] SEC_ERR_FOUND/ INTR1_ENABLE_SET[0] SEC_ERR_FOUND and INTR0_ENABLE_CLEAR[0] SEC_ERR_FOUND/ INTR1_ENABLE_CLEAR[0] SEC_ERR_FOUND	This status flag is set when the value of the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT bit field reaches the threshold counter value configured through the CFG_OCMC_ECC_ERROR[15:0] CFG_SEC_CNT_MAX bit field. The status remains asserted until the counter is not cleared. This is done by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[0] CLEAR_SEC_ERR_CNT bit. If the counter is not cleared another interrupt is re-issued.

(1) The LAST_ILLEGAL_OCMC_ADDR register stores the last illegal L3_MAIN address caused the assertion of this event.

15.6.3.5 OCM Subsystem Memory Regions

The SRAM associated with the OCMC_RAM1 is accessible through the L3_MAIN interconnect. The start address is 0x4030 0000 and the end address is 0x4037 FFFF. That is address space of 512KiB. The configuration registers of the OCMC_RAM1 are accessible through the L4_PER3 starting at address 0x4880 4000.

The SRAM associated with the OCMC_RAM2 is accessible through the L3_MAIN interconnect. The start address is 0x4040 0000 and the end address is 0x404F FFFF. That is address space of 1MiB. The configuration registers of the OCMC_RAM2 are accessible through the L4_PER3 starting at address 0x4880 A000.

The SRAM associated with the OCMC_RAM3 is accessible through the L3_MAIN interconnect. The start address is 0x4050 0000 and the end address is 0x405F FFFF. That is address space of 1MiB. The configuration registers of the OCMC_RAM3 are accessible through the L4_PER3 starting at address 0x4881 0000.

Note

OCMC_RAM2 and OCMC_RAM3 banks are not present on DRA74x devices, but are included in some of the DRA75x devices. For details, see the device data manual.

15.6.3.6 OCM Controller Modes Of Operation

The OCM controller supports four modes of operation. Each of these modes is selected through the CFG_OCMC_ECC[2:0] CFG_OCMC_MODE bit field. The four supported modes are the following:

- Non-ECC mode (Data Access) - Accesses to the SRAM are non-ECC-enabled and an ECC is not calculated.

- Non-ECC mode (Code Access) - The L3_MAIN address is mapped to the SRAM memory space where the ECC code is stored. This mode allows read and write access to the 9-bit ECC word associated with each 128-bit data word. This mode is used for test purposes.
- Full-ECC mode - Accesses to the SRAM are ECC-enabled and a 9-bit ECC is calcualted for each 128 bits of data.
- Block-ECC mode - A 9-bit ECC is calcualted only for a 128KiB block of the SRAM. Accesses outside the 128KiB ECC-enabled block are non-ECC data acceses. In other words, a 9-bit ECC will be calculated not for each 128-bit data word of the whole SRAM space, but for each 128-bit data word within the boundaries of this 128KiB block of the SRAM. The selection of a 128KiB ECC-enabled block is done using bits[19:0] of the CFG_OCMC_ECC_MEM_BLK register. Each bit specifies which 128KiB block of the SRAM is ECC-enabled. 0x1 is the active value for each bit. In addition, more than one 128KiB ECC-enabled block can be selected.

The default mode of operation for the OCM controller is the non-ECC data access mode.

15.6.3.7 ECC Associated FIFOs

There are three FIFOs used when ECC mode is enabled. Each FIFO is four level deep. The FIFOs are the following:

- Address FIFO for single errors also referred to as SEC FIFO
- Address FIFO for double errors also referred to as DED FIFO
- Address FIFO for address errors also referred to as ADDRERR FIFO

The SEC FIFO stores the SRAM addresses at which a single error is detected. The FIFO is able to store up to four unique addresses of the single errors occured. If there are more than four single errors associated with unique addresses, then only the first four addresses are stored in the FIFO. In case of occurrence of multiple correctable errors, the addresses are stored in the order the errors occurred. The SEC FIFO can be optionally configured to store all addresses of the single errors occured including also the repeated addresses. This is done by setting to 0x0 the CFG_OCMC_ECC_ERROR[24] CFG_DISCARD_DUP_ADDR bit. The SEC FIFO can be cleared by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[0] CLEAR_SEC_ERR_CNT bit or by reding FIFO's content one by one through the STATUS_SEC_ERROR_TRACE[17:0] ADDRESS_128BIT bit field which points to the SEC FIFO. In addition, the STATUS_SEC_ERROR_TRACE[18] VALID bit shows whether the FIFO is empty or not. A value of 0x1 means that the SEC FIFO is not empty and valid address can be read.

The DED FIFO stores the SRAM addresses when a double error is detected. The FIFO is able to store up to four unique addresses, but it can also be configured to store the repeated addresses by setting to 0x0 the CFG_OCMC_ECC_ERROR[24] CFG_DISCARD_DUP_ADDR bit. The STATUS_DED_ERROR_TRACE[17:0] ADDRESS_128BIT bit field points to the DED FIFO. The STATUS_DED_ERROR_TRACE[18] VALID bit shows whether the FIFO is empty or not. The DED FIFO can be cleared by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[1] CLEAR_DED_ERR_CNT bit or by reading its content one by one through the STATUS_DED_ERROR_TRACE[17:0] ADDRESS_128BIT.

The ADDRERR FIFO stores the SRAM addresses where the single error occured is an addres error. The STATUS_ADDR_TRANSLATION_ERROR_TRACE[17:0] ADDRESS_128BIT bit field points to the ADDRERR FIFO. The STATUS_ADDR_TRANSLATION_ERROR_TRACE[18] VALID bit shows whether the FIFO is empty or not. The ADDRERR FIFO can be cleared by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[2] CLEAR_ADDR_ERR_CNT bit or by reading its content one by one through the STATUS_ADDR_TRANSLATION_ERROR_TRACE[17:0] ADDRESS_128BIT pointer. The ADDRERR FIFO is also able to store up to four unique addresses and can also be configured to store repeated addresses by setting to 0x0 the CFG_OCMC_ECC_ERROR[24] CFG_DISCARD_DUP_ADDR bit.

15.6.3.8 ECC Counters And Corrected Bit Distribution Register

There are three counters which are used to count different types of errors occured when the ECC mode is enabled. The counters are the following:

- SEC Counter - STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT. This is the counter for the single errors occured. It keeps track of all single errors. That is, even for those with error addresses that have already been logged.

- DED Counter - STATUS_ERROR_CNT[19:16] DED_ERROR_CNT. This is the counter for the double error detections.
- ADDRERR Counter - STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT. This is the counter for the address errors found when a single error occurs. That is, when the single error is an addres error.

When a single error in the 128-bit data word occurs and this error is corrected, the OCM controller indicates for the corrected bit of this 128-bit word by setting to 0x1 a corresponding bit in the corrected bit distribution register. This is a 128-bit register composed by the following registers:

- STATUS_SEC_ERROR_DISTR_0[31:0] SEC_BIT_ERROR_FOUND - Bits [31:0] of the corrected 128-bit data word.
- STATUS_SEC_ERROR_DISTR_1[31:0] SEC_BIT_ERROR_FOUND - Bits [63:32] of the corrected 128-bit data word.
- STATUS_SEC_ERROR_DISTR_2[31:0] SEC_BIT_ERROR_FOUND - Bits [95:64] of the corrected 128-bit data word.
- STATUS_SEC_ERROR_DISTR_3[31:0] SEC_BIT_ERROR_FOUND - Bits [127:96] of the corrected 128-bit data word.

When a single error in the ECC itself occurs, the OCM controller indicates for this error by setting to 0x1 a corresponding bit in the STATUS_SEC_ERROR_DISTR_4[7:0] SEC_ECC_CODE_ERROR_FOUND bit field. The parity bit is not associated with this bit field.

15.6.3.9 ECC Support

The ECC mode is enabled through the CFG_OCMC_ECC[2:0] CFG_OCMC_MODE bit field. When enabled a 9-bit Hamming ECC is calculated and stored for each consecutive 128-bit block of the SRAM. The ECC is calculated based on a codeword constructed by concatenating the 128 bits of data with the address bits A21 through A4 of the L3_MAIN. The ECC generated is Hamming(155,146) code and has a Hamming distance of 4. The OCM controller uses this code to validate the content of the SRAM, to correct a single bit error that occurs within the 128-bit boundary or to determine if a non-correctable error has occurred within the 128-bit boundary.

During write transactions and when ECC is enabled, for every 128-bit data word, the ECC is calculated and stored in a 9-bit field of the SRAM associated only with the address where that 128-bit data word is written. If an initiator performs a write transaction which is less than 128 bits or it is non-128-bit aligned transaction, then the content of all 128 bits is read, the new sub-quanta of data is inserted and the ECC is calculated based on all 128 bits.

During read transactions and when ECC is enabled, the ECC is calcualated based on the memory address (bits A21 through A4) and the 128-bit data word read from the SRAM. This ECC is then compared to the ECC stored at the address when the data was written to the SRAM. If the two ECCs are matching then the data is transferred to the requesting initiator without further exceptions. If the two ECCs are not matching then a check is made to determine if the error is correctable or not. If the error is not correctable, that is a double error, then the address of the erroneous word is stored in the DED FIFO, the STATUS_ERROR_CNT[19:16] DED_ERROR_CNT counter is incremented by 1 and the double error flag is asserted. The double error flag is indicated by the INTR0_STATUS_RAW_SET[1] DED_ERR_FOUND/ INTR1_STATUS_RAW_SET[1] DED_ERR_FOUND bits, but its assertion depends on the threshold configured through the CFG_OCMC_ECC_ERROR[19:16] CFG_DED_CNT_MAX bit field. For more information see, Table 15-556, *OCM Subsystem Events*.

In case of a single error, the OCM controller first determines whether the erroneous bit is in the data, address or the ECC itself. In this case the following applies:

- If the single error is located in the data read, then the erroneous bit is corrected and the data is passed to the requesting initiator. The erroneous bit is corrected also in the SRAM location by the auto re-write feature of the OCM controller if this feature is enabled by setting to 0x1 the CFG_OCMC_ECC[3] CFG_ECC_SEC_AUTO_CORRECT bit field. In addition, the address of the corrected data word is pushed to the SEC FIFO and the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counter is incremented by 1. The single error flag is asserted too, but only if the value of the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counter reaches the threshold configured by the CFG_OCMC_ECC_ERROR[15:0] CFG_SEC_CNT_MAX bit field. The single error flag is indicated by the INTR0_STATUS_RAW_SET[0] SEC_ERR_FOUND/ INTR1_STATUS_RAW_SET[0] SEC_ERR_FOUND bits. For more information see, Table 15-556, *OCM Subsystem Events*.

- If the single error is located in the address portion of the quanta, then the data is passed to the requesting initiator unchanged and the STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT and STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counters are incremented by 1. The address associated with this error is stored in the ADDRERR FIFO and the address error flag is asserted. This flag is indicated by the INTR0_STATUS_RAW_SET[2] ADDR_ERR_FOUND/INTR1_STATUS_RAW_SET[2] ADDR_ERR_FOUND bits, but its assertion depends on the threshold configured through the CFG_OCMC_ECC_ERROR[23:20] CFG_ADDR_ERR_CNT_MAX bit field. For more information see, Table 15-556, *OCM Subsystem Events*.
- If the single error is located in the ECC itself, then the data is passed to the initiator unchanged but the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counter is incremented by 1 and the address associated with this error is stored in the SEC FIFO.

Table 15-557 summarizes the actions taken by the OCM controller for the different error types.

Table 15-557. OCMC Error Handling In Case Of Different Error Types

Error Type	Data Returned	SRAM Data/ECC Update	SEC Counter	SEC FIFO	DED Counter	DED FIFO	ADDRERR Counter	ADDRERR FIFO	Error Bit Distribution
Single data error	Corrected	Data corrected/ECC re-generated	Incremented by 1	Address written	-	-	-	-	The content of the corrected bit distribution register is updated
Single address error	Unchanged	ECC re-generated	Incremented by 1	-	-	-	Incremented by 1	Address written	-
Single ECC error	Unchanged	ECC re-generated	Incremented by 1	Address written	-	-	-	-	The content of the STATUS_SEC_ERROR_DISTR_4[7:0] SEC_ECC_CODE_ERROR_FOUND bit field is updated
Double error	Unchanged	-	-	-	Incremented by 1	Address written	-	-	-

If ECC has to be used, before performing SRAM read/write operations the following steps should be performed:

- Configuring the ECC registers
- Enabling the ECC
- Initializing the SRAM with data
- Clearing the status flags

Table 15-558 describes in detail the steps to be performed.

Table 15-558. OCMC ECC Configuration

Step	Associated Register/ Bit Field	Value
(Optional) Enable the error response on L3_MAIN for a non-correctable error	CFG_OCMC_ECC[4] CFG_ECC_ERR_SRESP_EN	0x1
(Optional) Enable data auto correction in case of a single error	CFG_OCMC_ECC[3] CFG_ECC_SEC_AUTO_CORRECT	0x1
Enable the memory blocks to be ECC protected, if ECC block mode is used	CFG_OCMC_ECC_MEM_BLK[19:0] CFG_ECC_ENABLED_128K_BLK	0x-
(Optional) Configure the address errors count needed for triggering an interrupt	CFG_OCMC_ECC_ERROR[23:20] CFG_ADDR_ERR_CNT_MAX	0x-

Table 15-558. OCMC ECC Configuration (continued)

Step	Associated Register/ Bit Field	Value
(Optional) Configure the DED count needed for triggering an interrupt	CFG_OCMC_ECC_ERROR[19:16] CFG_DED_CNT_MAX	0x-
(Optional) Configure the SEC count needed for triggering an interrupt	CFG_OCMC_ECC_ERROR[15:0] CFG_SEC_CNT_MAX	0x-
Enable ECC	CFG_OCMC_ECC[2:0] CFG_OCMC_MODE	0x2 for full ECC or 0x3 for block ECC mode
Initialize the memory with data		
Clear the error counters in the STATUS_ERROR_CNT register	CFG_OCMC_ECC_CLEAR_HIST[2:0]	0x7
Clear the SEC bit distribution history from the following registers:	CFG_OCMC_ECC_CLEAR_HIST[3] CLEAR_SEC_BIT_DISTR	0x1
• STATUS_SEC_ERROR_DISTR_0		
• STATUS_SEC_ERROR_DISTR_1		
• STATUS_SEC_ERROR_DISTR_2		
• STATUS_SEC_ERROR_DISTR_3		
• STATUS_SEC_ERROR_DISTR_4		

15.6.3.10 Circular Buffer (CBUF) Support

The OCM controller provides up to 12 programmable circular buffers that are mapped to virtual video frames to support sliced based on-the-fly video frame processing. Each circular buffer must be programmed with the following:

- Unique virtual frame start address - configured through CBUF_i_VBUF_START_ADDR[31:4] ADDR.
- Unique virtual frame end address - configured through CBUF_i_VBUF_END_ADDR[31:4] ADDR.
- SRAM start address - configured through CBUF_i_OCMC_START_ADDR[31:4] ADDR.
- SRAM size allocated for that circular buffer - configured through CBUF_i_OCMC_BUF_SIZE[19:4] BUF_SIZE.

The circular buffer mode is enabled when the CFG_OCMC_CBUF_EN[0] CBUF_MODE_EN bit is set to 0x1. In addition, to enable certain circular buffer the corresponding bit in the CFG_OCMC_CBUF_EN[27:16] bit field must be set to 0x1. For example, to enable CBUF_0 both the CFG_OCMC_CBUF_EN[0] CBUF_MODE_EN and CFG_OCMC_CBUF_EN[16] CBUF_EN_0 bits must be set to 0x1.

The data transfers are made to the virtual frame addresses. The OCM controller translates these virtual addresses to SRAM addresses allocated for the CBUF. The address translation is done on-the-fly without leading to any additional latency and therefore has no performance impact on the overall operation of the OCM controller.

The L3_MAIN address ranges associated with the virtual frame addresses are the following:

- For OCMC_RAM1 the start address is 0x4180 0000 and the end address is 0x41FF FFFF.
- For OCMC_RAM2 the start address is 0x4900 0000 and the end address is 0x497F FFFF.
- For OCMC_RAM3 the start address is 0x4980 0000 and the end address is 0x49FF FFFF.

Note

OCMC_RAM2 and OCMC_RAM3 banks are not present on DRA74x devices, but are included in some of the DRA75x devices. For details, see the device data manual.

When an initiator performs an access to these L3_MAIN addresses, these virtual frame addresses will be translated automatically by the OCM controller into SRAM addresses. Figure 15-109 shows the address mapping of the virtual frame addresses to CBUF addresses.

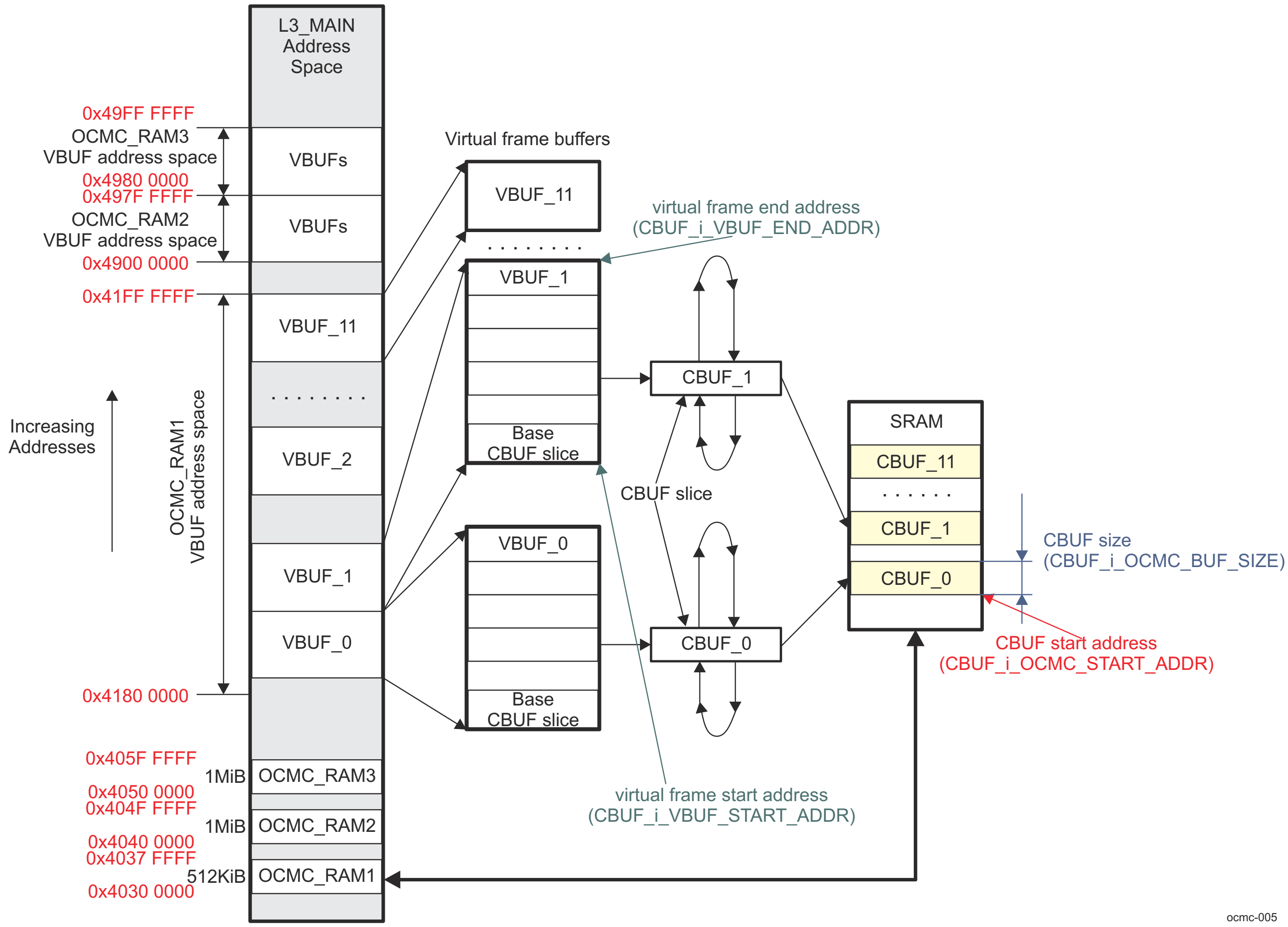


Figure 15-109. VBUF to CBUF Address Mapping

It must be taken into account that the start address of a virtual frame always begins at the start of the CBUF. Since the virtual frame buffer size is not necessarily to be a multiple of the CBUF size, the last VBUF write location may not be the last line of the CBUF. As the CBUF read and write pointers are independently managed the initiators that are writing to and reading from the CBUF should be coordinated in order to avoid overflow and underflow conditions.

15.6.3.11 CBUF Mode Error Handling

For each L3_MAIN write or read access associated with the virtual frame buffer, the OCM controller performs various address error checks to prevent illegal CBUF accesses from causing false overflow and underflow conditions. Once the L3_MAIN VBUF address is determined as valid, the OCM controller further checks for overflow and underflow conditions to properly handle the access to the CBUF during each one of these conditions and to notify for error condition an external host.

For more information regarding all CBUF events see, Table 15-556, *OCM Subsystem Events*.

15.6.3.11.1 VBUF Address Not Mapped to a CBUF Memory Space

The L3_MAIN VBUF address should be mapped to a valid CBUF region. If this address is mapped to none of the configured CBUF regions, then the OCM controller generates a virtual address error or out of range CBUF address error event. It is possible both virtual address error and out of range CBUF address error to be generated. The L3_MAIN VBUF address caused these two events can be read through the LAST_ILLEGAL_OCMC_ADDR[31:0] ADDR bit field.

Virtual address error is generated when the L3_MAIN VBUF address is lower than the virtual frame start address. This is indicated by asserting the INTR0_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/INTR1_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND bit.

The out of range CBUF address error is generated when the L3_MAIN VBUF address is greater than the virtual frame end address. This error is an indication of an unexpected change in the frame size or a long frame and should be handled immediately to prevent further memory corruption. The out of range CBUF address error can be caused when performing write or read access. The out of range CBUF write address error is indicated by the STATUS_CBUF_WR_OUT_OF_RANGE_ERR[11:0] CBUF_ERR bit field. The out of range CBUF read address error is indicated by the STATUS_CBUF_RD_OUT_OF_RANGE_ERROR[11:0] CBUF_ERR bit field. Each bit in these registers is associated only with one CBUF. Bit 0 corresponds to CBUF_0, bit 1 corresponds to CBUF_1 and so on. A value of 0x1 for each bit means that there is an error (CBUF address is out of CBUF range) for the corresponding CBUF. Writing 0x1 for each bit clears the status.

The INTR0_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_WR_OUT_OF_RANGE_ERR[11:0] CBUF_ERR bit field is set to 0x1.

The INTR0_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND/ INTR1_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_RD_OUT_OF_RANGE_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

15.6.3.11.2 VBUF Access Not Starting At The Base Address

After (hardware of software) reset or on a new frame start condition, a frame write or read access for any VBUF must start from its virtual frame start address (CBUF_i_VBUF_START_ADDR[31:4] ADDR). If this is not met the OCM controller ignors all accesses which don't start at address CBUF_i_VBUF_START_ADDR[31:4] ADDR and also generates an interrupt. The STATUS_CBUF_WR_VBUF_START_ERR[11:0] CBUF_ERR register contains the status bits of this interrupt which are associated with frame write access. The STATUS_CBUF_VBUF_RD_START_ERROR[11:0] CBUF_ERR register contains the status bits associated with frame read access. Each bit in these two registers is associated only with one CBUF. A value of 0x1 for each bit means that the read/write access does not start at the VBUF start address.

While this error condition exists, no updates to the CBUF channel status registers are made. An exception to this constraint is when the last read/write access is still within the base CBUF slice. The enforcing of a frame starting at VBUF starting address is not done in this case to allow random access within the same CBUF. However, this will be flagged and handled as a short frame occurrence.

The INTR0_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND/ INTR1_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_WR_VBUF_START_ERR[11:0] CBUF_ERR bit field is set to 0x1.

The INTR0_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND/INTR1_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_VBUF_RD_START_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

15.6.3.11.3 Illegal Address Change Between Two Same Type Accesses

The OCM controller requires both read and write accesses to a virtual frame buffer to be performed in the positive raster scan order, that is, successive write or read accesses should be to a higher address except on a new frame start for which the address jumps back down to the beginning of the virtual frame buffer. The OCM controller checks to make sure that a virtual address meets one of the following conditions:

- It is within the same CBUF slice as the last access of the same type (read or write)
- It is within the next CBUF slice of the last access of the same type (read or write)
- The virtual address is to the VBUF frame start address

Based on these constraints, the OCM controller generates an write/read address sequence error interrupt and invalidates the access, that is, writes are disabled and reads return unknown data, if the following two conditions are met:

- The current access address is not to a current or next CBUF slice space
- The current access address is not to the base of the virtual frame buffer (the first CBUF slice)

The random accesses within the current CBUF slice are permited to allow some variations in the line width of the captured video which often happens with weak video signal conditions.

To avoid illegal CBUF slice backward switching, the modules capturing the input video and sending it to the CBUF of the OCM controller should limit the maximum width of a line width to be less than the stride size (in pixels). This guarantees that a start of a new line does not point back to the previous CBUF slice.

If a write or read address sequence error event is detected on an access, the OCM controller will simply invalidate the accesses and will not update the CBUF status registers.

The bits in the STATUS_CBUF_WR_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field indicate write address sequence error event.

The bits in the STATUS_CBUF_RD_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field indicate read address sequence error event.

In both the registers each bit is associated only with one CBUF.

The INTR0_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/INTR1_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_WR_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

The INTR0_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND/ INTR1_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_RD_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

15.6.3.11.4 Illegal Frame SIze (Short Frame Detection)

If a frame write access for any VBUF begins from its virtual frame start address without a CBUF software reset and the last write address is not the virtual frame end address or is in the last CBUF slice, then the previous frame is considered to be short. This includes the case where a new frame write happens twice without a new frame read in between, which is an indication of an extremely short frame. When a short frame event is detected, the OCM controller generates a short frame detection interrupt and temporarily disables overflow checking on the write access, including also the current access, until a frame read happens in order to avoid false overflow indication on a suspended read side following the short frame detection.

If the last write address is in the last CBUF slice the short frame detection can be ignored if the CFG_OCMC_CBUF_ERR_HANDLER[1] SHORT_FRAME_PREV_EOF_SEL bit is set to 0x1.

The STATUS_CBUF_SHORT_FRAME_DETECT[11:0] CBUF_ERR bit field has status flags to indicate a short frame detection for each CBUF. When a short frame condition occurs the corresponding bit is set to 0x1.

The INTR0_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND/ INTR1_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_SHORT_FRAME_DETECT[11:0] CBUF_ERR bit field is set to 0x1.

In addition, the short frame detection enabling/disabling is controlled by the CFG_OCMC_CBUF_ERR_HANDLER[0] SHORT_FRAME_DETECT_CHECK_EN bit.

15.6.3.11.5 CBUF Overflow

For each valid CBUF write access, the write address is compared to the last read address for an overflow condition. If an overflow is detected, the corresponding CBUF overflow status bit in the STATUS_CBUF_OVERFLOW_MID or STATUS_CBUF_OVERFLOW_WRAP registers is set to 0x1 and an interrupt is generated. This interrupt condition exists until the corresponding overflow status bit is cleared.

There are two types of overflow conditions referred to as Cbuf_overflow_wrap and Cbuf_overflow_mid.

Cbuf_overflow_wrap occurs in case at least one of the following two:
- VBUF write address – VBUF frame start address < CBUF size
- VBUF write address – VBUF frame start address > VBUF last read address – Current VBUF start address

Cbuf_overflow_mid occurs in case at least one of the following two:
- VBUF write address – VBUF frame start address > CBUF size
- VBUF write address – VBUF last read address > CBUF size

If an overflow condition is detected on a CBUF write access, the OCM controller generates a CBUF overflow (Cbuf_overflow_wrap or Cbuf_overflow_mid) interrupt and enter an overflow error handling state. In this state, the OCM controller do the following in the default error handling configuration:
- When a Cbuf_Overflow_Wrap condition occurs (write buffer wraparound with read from previous frame still remaining), writes are disabled but reads are serviced normally. This mechanism preserves the back end of the previous frame and discards the new frame which is already showing signs of overflowing. Writes will be re-enabled automatically on write to the virtual frame start address which indicates the next frame after the overflowed frame has begun. The overflow (and underflow) check for this CBUF will be disabled until detection of write to/read from the virtual frame start address. The module which reads should not start reading the next frame (which is being discarded) or stop reading the frame if the read of this new frame has already started. This mechanism allows both write and read sides to re-synchronize on the next frame after one-frame drop.
- When a Cbuf_Overflow_Mid condition occurs, writes and reads are serviced normally allowing the read side to catch up to the write side. The overflow (and underflow) check will be disabled until first a write to the virtual frame start address and then a read from the virtual frame start address is detected. This mechanism allows the frame with a “momentary” overflow condition to be saved. The reader side has an option to stop reading and discarding the data by just resetting its CBUF read pointer at the next frame start, if the read side is too far back.

These steps are taken to minimize the frame loss due to an overflow condition. To support the previously described behavior but to allow software to override the default error handling behavior the following controls are available:
- CFG_OCMC_CBUF_ERR_HANDLER[7:6] OVERFLOW_CHECK_REENABLE_SEL
- CFG_OCMC_CBUF_ERR_HANDLER[5:4] OVERFLOW_WRITE_HANDLER_SEL

In addition, the overflow check can be enabled or disabled through the CFG_OCMC_CBUF_ERR_HANDLER[2] OVERFLOW_ERR_CHECK_EN bit.

15.6.3.11.6 CBUF Underflow

According to the CBUF operation a read request can only be made when there are enough number of video lines written to the CBUF. Therefore, there should not be any underflow conditions due to normal processing speed variations. An underflow condition occurs when the VBUF read address is greater than the last VBUF write address. To detect an abnormal failure in the read/write mechanism which could cause an underflow condition, each valid CBUF read access is checked for an underflow condition and if detected, an underflow interrupt is generated for the selected CBUF. The STATUS_CBUF_UNDERFLOW[11:0] CBUF_ERR bit field has status flags indicating when underflow occurs. This interrupt condition exists until the corresponding underflow status bit in the STATUS_CBUF_UNDERFLOW register is cleared.

If an underflow is detected on a CBUF read access, the OCM controller generates an underflow interrupt and continues to perform reads and writes normally, but the return data will not be correct while the underflow condition exists.

To avoid a false underflow detection at the end of a frame (due to external decoders sending short last line or due to interlaced input video having different number of lines in even/odd fields), the CFG_OCMC_CBUF_ERR_HANDLER[8] UNDERFLOW_LAST_CBUF_SLICE_DISABLE bit can be set to 0x1 which disables the underflow checking in the last CBUF slice.

The INTR0_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND/INTR1_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_UNDERFLOW[11:0] CBUF_ERR bit field is set to 0x1.

In addition, the underflow check can be enabled or disabled through the CFG_OCMC_CBUF_ERR_HANDLER[3] UNDERFLOW_ERR_CHECK_EN bit.

15.6.3.12 Status Reporting

The OCM controller keeps a history of last valid write and read addresses for each CBUF. The CBUF_k_LAST_WR_ADDR register stores the address for the last valid write and the CBUF_k_LAST_RD_ADDR register stores the address for the last valid read access.

The OCMC_ECC generates also a general VBUF mode error event for an access made with illegal VBUF address. The INTR0_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND bit indicates for this event.

15.6.4 OCM Subsystem Register Manual

15.6.4.1 OCM Subsystem Instance Summary

Table 15-559. OCM Subsystem Instance Summary

Module Name	Module Base Address	Size
OCMC_RAM1	0x4880 4000	4KiB
OCMC_RAM2	0x4880 A000	4KiB
OCMC_RAM3	0x4881 0000	4KiB

Note

OCMC_RAM2 and OCMC_RAM3 banks are not present on DRA74x devices, but are included in some of the DRA75x devices. For details, see the device data manual.

15.6.4.2 OCM Subsystem Registers

15.6.4.2.1 OCM Subsystem Register Summary

Table 15-560. OCM Subsystem Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM1 Base Address	OCMC_RAM2 Base Address	OCMC_RAM3 Base Address
OCMC_ECC_PID	R	32	0x0000 0000	0x4880 4000	0x4880 A000	0x4881 0000
OCMC_SYSCONFIG_PM	RW	32	0x0000 0004	0x4880 4004	0x4880 A004	0x4881 0004
OCMC_SYSCONFIG_RST	RW	32	0x0000 0008	0x4880 4008	0x4880 A008	0x4881 0008
OCMC_MEM_SIZE_READ	R	32	0x0000 000C	0x4880 400C	0x4880 A00C	0x4881 000C
INTR0_STATUS_RAW_SET	RW	32	0x0000 0040	0x4880 4040	0x4880 A040	0x4881 0040
INTR0_STATUS_ENABLED_CLEAR	RW	32	0x0000 0044	0x4880 4044	0x4880 A044	0x4881 0044
INTR0_ENABLE_SET	RW	32	0x0000 0048	0x4880 4048	0x4880 A048	0x4881 0048
INTR0_ENABLE_CLEAR	RW	32	0x0000 004C	0x4880 404C	0x4880 A04C	0x4881 004C
OCMC_INTR0_EOI	RW	32	0x0000 0050	0x4880 4050	0x4880 A050	0x4881 0050
INTR1_STATUS_RAW_SET	RW	32	0x0000 0060	0x4880 4060	0x4880 A060	0x4881 0060
INTR1_STATUS_ENABLED_CLEAR	RW	32	0x0000 0064	0x4880 4064	0x4880 A064	0x4881 0064
INTR1_ENABLE_SET	RW	32	0x0000 0068	0x4880 4068	0x4880 A068	0x4881 0068
INTR1_ENABLE_CLEAR	RW	32	0x0000 006C	0x4880 406C	0x4880 A06C	0x4881 006C
OCMC_INTR1_EOI	RW	32	0x0000 0070	0x4880 4070	0x4880 A070	0x4881 0070
CFG_OCMC_ECC	RW	32	0x0000 0080	0x4880 4080	0x4880 A080	0x4881 0080
CFG_OCMC_ECC_MEM_BLK	RW	32	0x0000 0084	0x4880 4084	0x4880 A084	0x4881 0084

Table 15-560. OCM Subsystem Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM1 Base Address	OCMC_RAM2 Base Address	OCMC_RAM3 Base Address
CFG_OCMC_ECC_ERROR	RW	32	0x0000 0088	0x4880 4088	0x4880 A088	0x4881 0088
CFG_OCMC_ECC_CLEAR_HIST	RW	32	0x0000 008C	0x4880 408C	0x4880 A08C	0x4881 008C
STATUS_ERROR_CNT	R	32	0x0000 0090	0x4880 4090	0x4880 A090	0x4881 0090
STATUS_SEC_ERROR_TRACE	R	32	0x0000 0094	0x4880 4094	0x4880 A094	0x4881 0094
STATUS_DED_ERROR_TRACE	R	32	0x0000 0098	0x4880 4098	0x4880 A098	0x4881 0098
STATUS_ADDR_TRANSLATION_ERROR_TRACE	R	32	0x0000 009C	0x4880 409C	0x4880 A09C	0x4881 009C
STATUS_SEC_ERROR_DISTR_0	R	32	0x0000 00A0	0x4880 40A0	0x4880 A0A0	0x4881 00A0
STATUS_SEC_ERROR_DISTR_1	R	32	0x0000 00A4	0x4880 40A4	0x4880 A0A4	0x4881 00A4
STATUS_SEC_ERROR_DISTR_2	R	32	0x0000 00A8	0x4880 40A8	0x4880 A0A8	0x4881 00A8
STATUS_SEC_ERROR_DISTR_3	R	32	0x0000 00AC	0x4880 40AC	0x4880 A0AC	0x4881 00AC
STATUS_SEC_ERROR_DISTR_4	R	32	0x0000 00B0	0x4880 40B0	0x4880 A0B0	0x4881 00B0
CFG_OCMC_CBUF_EN	RW	32	0x0000 0200	0x4880 4200	0x4880 A200	0x4881 0200
CFG_OCMC_CBUF_RESET	RW	32	0x0000 0204	0x4880 4204	0x4880 A204	0x4881 0204
CFG_OCMC_CBUF_ERR_HANDLER	RW	32	0x0000 0208	0x4880 4208	0x4880 A208	0x4881 0208
STATUS_CBUF_WR_OUT_OF_RANGE_ERR	RW	32	0x0000 020C	0x4880 420C	0x4880 A20C	0x4881 020C
STATUS_CBUF_WR_VBUF_START_ERR	RW	32	0x0000 0210	0x4880 4210	0x4880 A210	0x4881 0210
STATUS_CBUF_WR_ADDR_SEQ_ERROR	RW	32	0x0000 0214	0x4880 4214	0x4880 A214	0x4881 0214
STATUS_CBUF_RD_OUT_OF_RANGE_ERROR	RW	32	0x0000 0218	0x4880 4218	0x4880 A218	0x4881 0218
STATUS_CBUF_VBUF_RD_START_ERROR	RW	32	0x0000 021C	0x4880 421C	0x4880 A21C	0x4881 021C
STATUS_CBUF_RD_ADDR_SEQ_ERROR	RW	32	0x0000 0220	0x4880 4220	0x4880 A220	0x4881 0220
STATUS_CBUF_OVERFLOW_MID	RW	32	0x0000 0224	0x4880 4224	0x4880 A224	0x4881 0224
STATUS_CBUF_OVERFLOW_WRAP	RW	32	0x0000 0228	0x4880 4228	0x4880 A228	0x4881 0228
STATUS_CBUF_UNDERFLOW	RW	32	0x0000 022C	0x4880 422C	0x4880 A22C	0x4881 022C
STATUS_CBUF_SHORT_FRAME_DETECT	RW	32	0x0000 0230	0x4880 4230	0x4880 A230	0x4881 0230
CBUF_i_VBUF_START_ADDR (1)	RW	32	0x0000 0240 + (i*16)	0x4880 4240 + (i*16)	0x4880 A240 + (i*16)	0x4881 0240 + (i*16)
CBUF_i_VBUF_END_ADDR (1)	RW	32	0x0000 0244 + (i*16)	0x4880 4244 + (i*16)	0x4880 A244 + (i*16)	0x4881 0244 + (i*16)
CBUF_i_OCMC_START_ADDR (1)	RW	32	0x0000 0248 + (i*16)	0x4880 4248 + (i*16)	0x4880 A248 + (i*16)	0x4881 0248 + (i*16)
CBUF_i_OCMC_BUF_SIZE (1)	RW	32	0x0000 024C + (i*16)	0x4880 424C + (i*16)	0x4880 A24C + (i*16)	0x4881 024C + (i*16)
CBUF_k_LAST_WR_ADDR (2)	R	32	0x0000 0300 + (k*8)	0x4880 4300 + (k*8)	0x4880 A300 + (k*8)	0x4881 0300 + (k*8)
CBUF_k_LAST_RD_ADDR (2)	R	32	0x0000 0304 + (k*8)	0x4880 4304 + (k*8)	0x4880 A304 + (k*8)	0x4881 0304 + (k*8)
LAST_ILLEGAL_OCMC_ADDR	R	32	0x0000 0360	0x4880 4360	0x4880 A360	0x4881 0360

(1) i = 0 to 11
(2) k = 0 to 11

15.6.4.2.2 OCM Subsystem Register Description

Table 15-561. OCMC_ECC_PID

Address Offset	0x0000 0000

Table 15-561. OCMC_ECC_PID (continued)

Physical Address	0x4880 4000 0x4880 A000 0x4881 0000	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	TI internal data	R	0x-

Table 15-562. Register Call Summary for Register OCMC_ECC_PID

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [0]

Table 15-563. OCMC_SYSCONFIG_PM

Address Offset	0x0000 0004		
Physical Address	0x4880 4004 0x4880 A004 0x4881 0004	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only. 0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate IRQ-request-related wakeup events. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate IRQ-request-related wakeup events when in idle state.	RW	0x2
1:0	RESERVED		R	0x0

Table 15-564. Register Call Summary for Register OCMC_SYSCONFIG_PM

On-Chip Memory (OCM) Subsystem

- Clock Management: [0]
- OCM Subsystem Register Summary: [1]

Table 15-565. OCMC_SYSCONFIG_RST

Address Offset	0x0000 0008		
Physical Address	0x4880 4008 0x4880 A008 0x4881 0008	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:1	0
RESERVED	SW_RST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SW_RST	Software reset of the OCM controller configuration and history logic (does not reset L4 interface) 0x0: Normal operation (OCM controller is not reset) 0x1: Reset the OCM controller (except its registers). This bit must be set back to 0x0 to resume the normal operation of the OCM Controller.	RW	0x0

Table 15-566. Register Call Summary for Register OCMC_SYSCONFIG_RST

On-Chip Memory (OCM) Subsystem

- Resets: [0]
- OCM Subsystem Register Summary: [1]

Table 15-567. OCMC_MEM_SIZE_READ

Address Offset	0x0000 000C		
Physical Address	0x4880 400C 0x4880 A00C 0x4881 000C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	This register provides the status of the OCM Controller configuration.		
Type	R		

31:17	16:12	11:10	9	8	7:5	4:0
RESERVED	VBUF_ADDR_MSB	RESERVED	MEM_CBUF_ENABLE	MEM_ECC_ENABLE	RESERVED	MEM_SIZE_128K_CNT

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
16:12	VBUF_ADDR_MSB	This bit field returns the MSB bit of the valid VBUF address range. The default value of 23 means that the valid VBUF address range is from 0x8000 0000 to 0x80FF FFFF	R	0x-
11:10	RESERVED		R	0x0
9	MEM_CBUF_ENABLE	Indicates whether CBUF is supported or not. 0x0: CBUF not supported 0x1: CBUF supported	R	0x-
8	MEM_ECC_ENABLE	Indicates whether ECC is supported or not. 0x0: ECC not supported 0x1: ECC supported	R	0x-
7:5	RESERVED		R	0x0
4:0	MEM_SIZE_128K_CNT	This bit field indicates how many 128KiB memory blocks are present in the SRAM. Access beyond the memory size reported in the MEM_SIZE_128K_CNT bit field results in an address error interrupt. 0x1: One 128KiB memory block 0x2: Two 128KiB memory blocks ... 0x14: 20 memory blocks of 128KiB	R	0x-

Table 15-568. Register Call Summary for Register OCMC_MEM_SIZE_READ

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [0]
- OCM Subsystem Register Description: [3] [4]

Table 15-569. INTR0_STATUS_RAW_SET

Address Offset	0x0000 0040		
Physical Address	0x4880 4040 0x4880 A040 0x4881 0040	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	This register contains the raw interrupt status. Read indicates RAW interrupt status (0=inactive, 1=active). Writing 1 will SET the corresponding raw status bit (soft interrupt set). Writing 0 has no effect.		
Type	RW		

31:15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame.	RW	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0
0	SEC_ERR_FOUND		RW	0x0

Table 15-570. Register Call Summary for Register INTR0_STATUS_RAW_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]
- ECC Support: [20] [21] [22]
- VBUF Address Not Mapped to a CBUF Memory Space: [23] [24] [25]
- VBUF Access Not Starting At The Base Address: [26] [27]
- Illegal Address Change Between Two Same Type Accesses: [28] [29]
- Illegal Frame SIze (Short Frame Detection): [30]
- CBUF Underflow: [31]
- Status Reporting: [32]
- OCM Subsystem Register Summary: [33]

Table 15-571. INTR0_STATUS_ENABLED_CLEAR

Address Offset	0x0000 0044		
Physical Address	0x4880 4044 0x4880 A044 0x4881 0044	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	Read indicates ENABLED interrupt status (0=inactive, 1=active). Writing 1 will CLEAR the corresponding enabled status bit. Writing 0 has no effect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 15-572. Register Call Summary for Register INTR0_STATUS_ENABLED_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18]
- OCM Subsystem Register Summary: [19]

Table 15-573. INTR0_ENABLE_SET

Address Offset	0x0000 0048		
Physical Address	0x4880 4048 0x4880 A048 0x4881 0048	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will set the corresponding interrupt enable bit. Writing 0 has no effect. Interrupt_enable_set		
Type	RW		

31:15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0

Bits	Field Name	Description	Type	Reset
0	SEC_ERR_FOUND		RW	0x0

Table 15-574. Register Call Summary for Register INTR0_ENABLE_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20]
- OCM Subsystem Register Summary: [21]

Table 15-575. INTR0_ENABLE_CLEAR

Address Offset	0x0000 004C		
Physical Address	0x4880 404C 0x4880 A04C 0x4881 004C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will clear interrupt enabled. Writing 0 has no effect. Interrupt_enable_clear		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WRITE_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 15-576. Register Call Summary for Register INTR0_ENABLE_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16]
- OCM Subsystem Register Summary: [17]

Table 15-577. OCMC_INTR0_EOI

Address Offset	0x0000 0050		
Physical Address	0x4880 4050 0x4880 A050 0x4881 0050	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	This register contains the EOI vector.		
Type	RW		

31–1	0
RESERVED	EOI_VECTOR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI_VECTOR		RW	0x0

Table 15-578. Register Call Summary for Register OCMC_INTR0_EOI

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [0]

Table 15-579. INTR1_STATUS_RAW_SET

Address Offset	0x0000 0060		
Physical Address	0x4880 4060 0x4880 A060 0x4881 0060	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	This register contains the raw interrupt status. Read indicates RAW interrupt status (0=inactive, 1=active). Writing 1 will SET the corresponding raw status bit (soft interrupt set). Writing 0 has no effect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0
0	SEC_ERR_FOUND		RW	0x0

Table 15-580. Register Call Summary for Register INTR1_STATUS_RAW_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]
- ECC Support: [20] [21] [22]
- VBUF Address Not Mapped to a CBUF Memory Space: [23] [24] [25]
- VBUF Access Not Starting At The Base Address: [26] [27]
- Illegal Address Change Between Two Same Type Accesses: [28] [29]
- Illegal Frame SIze (Short Frame Detection): [30]
- CBUF Underflow: [31]
- Status Reporting: [32]
- OCM Subsystem Register Summary: [33]

Table 15-581. INTR1_STATUS_ENABLED_CLEAR

Address Offset	0x0000 0064		
Physical Address	0x4880 4064 0x4880 A064 0x4881 0064	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	Read indicates ENABLED interrupt status (0=inactive, 1=active). Writing 1 will CLEAR the corresponding enabled status bit. Writing 0 has no effect.		
Type	RW		

31:15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 15-582. Register Call Summary for Register INTR1_STATUS_ENABLED_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18]
- OCM Subsystem Register Summary: [19]

Table 15-583. INTR1_ENABLE_SET

Address Offset	0x0000 0068		
Physical Address	0x4880 4068 0x4880 A068 0x4881 0068	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will set the corresponding interrupt enable bit. Writing 0 has no effect. Interrupt_enable_set		
Type	RW		

31:15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW	0x0

Bits	Field Name	Description	Type	Reset
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0
0	SEC_ERR_FOUND		RW	0x0

Table 15-584. Register Call Summary for Register INTR1_ENABLE_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]
- OCM Subsystem Register Summary: [20]

Table 15-585. INTR1_ENABLE_CLEAR

Address Offset	0x0000 006C		
Physical Address	0x4880 406C 0x4880 A06C 0x4881 006C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will clear interrupt enabled. Writing 0 has no effect. Interrupt_enable_clear		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 15-586. Register Call Summary for Register INTR1_ENABLE_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16]
- OCM Subsystem Register Summary: [17]

Table 15-587. OCMC_INTR1_EOI

Address Offset	0x0000 0070		
Physical Address	0x4880 4070 0x4880 A070 0x4881 0070	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	This register contains the EOI vector.		
Type	RW		

31–1	0
RESERVED	EOI_VECTOR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI_VECTOR		RW	0x0

Table 15-588. Register Call Summary for Register OCMC_INTR1_EOI

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [0]

Table 15-589. CFG_OCMC_ECC

Address Offset	0x0000 0080		
Physical Address	0x4880 4080 0x4880 A080 0x4881 0080	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31–6	5	4	3	2–0
RESERVED	CFG_ECC_OPT_NON_ECC_READ	CFG_ECC_ERR_SRESP_EN	CFG_ECC_SEC_AUTO_CORRECT	CFG_OCMC_MODE

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	CFG_ECC_OPT_NON_ECC_READ	Optimize read latency for non-ECC read. Returns the data one cycle faster if the read access is from a non-ECC enabled space. 0x0: Disable 0x1: Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
4	CFG_ECC_ERR_SRESP_EN	ECC non-correctable error SRESP enable. Enables ERR return on L3 OCP SRESP when a non-correctable data (DED) or address error is detected. 0x0: Disable 0x1: Enable	RW	0x0
3	CFG_ECC_SEC_AUTO_CORRECT	SEC error auto correction mode. Enables the OCM Controller to automatically update the wrong data word with the corrected word. 0x0: Disable 0x1: Enable (If the OCM Controller is performing a read-modify operation for a sub-128b write to an ECC enabled memory, the error found during the read phase will be corrected always regardless of the value of this bit)	RW	0x0
2:0	CFG_OCMC_MODE	OCM Controller memory access modes. 0x0: Non-ECC mode (data access) 0x1: Non-ECC mode (code access) 0x2: Full ECC enabled mode 0x3: Block ECC enabled mode 0x4-0x7: Reserved (internally defaults to 0x0 mode)	RW	0x0

Table 15-590. Register Call Summary for Register CFG_OCMC_ECC

On-Chip Memory (OCM) Subsystem

- OCM Controller Modes Of Operation: [0]
- ECC Support: [1] [2] [3] [4] [5]
- OCM Subsystem Register Summary: [6]

Table 15-591. CFG_OCMC_ECC_MEM_BLK

Address Offset	0x0000 0084		
Physical Address	0x4880 4084 0x4880 A084 0x4881 0084	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CFG_ECC_ENABLED_128K_BLK																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	CFG_ECC_ENABLED_128K_BLK	ECC memory block enable bits. The active level of each bit is 0x1. Bit [0] -> Address offset range 0x0 to 0x1FFFF Bit [1] -> Address offset range 0x20000 to 0x3FFFF ... Bit [19] -> Address offset range 0x260000 to 0x27FFFF	RW	0x0

Table 15-592. Register Call Summary for Register CFG_OCMC_ECC_MEM_BLK

On-Chip Memory (OCM) Subsystem

- OCM Controller Modes Of Operation: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [2]

Table 15-593. CFG_OCMC_ECC_ERROR

Address Offset	0x0000 0088		
Physical Address	0x4880 4088 0x4880 A088 0x4881 0088	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:25	24	23:20	19:16	15:0
RESERVED	CFG_DISCARD_DUP_ADDR	CFG_ADDR_ERR_CNT_MAX	CFG_DED_CNT_MAX	CFG_SEC_CNT_MAX

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CFG_DISCARD_DUP_ADDR	Do not save duplicate error address. This bit applies to the SEC, DED and ADDRERR FIFOs. 0x0: Save the duplicated addresses 0x1: Save only the unique addresses	RW	0x0
23:20	CFG_ADDR_ERR_CNT_MAX	Number of ADDR errors to trigger an interrupt (The value configured must be > 0 to generate an interrupt).	RW	0x1
19:16	CFG_DED_CNT_MAX	Number of DED errors to trigger an interrupt (The value configured must be > 0 to generate an interrupt).	RW	0x1
15:0	CFG_SEC_CNT_MAX	Number of SEC error to trigger an interrupt (The value configured must be > 0 to generate an interrupt).	RW	0x1

Table 15-594. Register Call Summary for Register CFG_OCMC_ECC_ERROR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2]
- ECC Associated FIFOs: [3] [4] [5]
- ECC Support: [6] [7] [8] [9] [10] [11]
- OCM Subsystem Register Summary: [12]

Table 15-595. CFG_OCMC_ECC_CLEAR_HIST

Address Offset	0x0000 008C		
Physical Address	0x4880 408C 0x4880 A08C 0x4881 008C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												CLEAR_SEC_BIT_DISTR	CLEAR_ADDR_ERR_CNT	CLEAR_DED_ERR_CNT	CLEAR_SEC_ERR_CNT

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	CLEAR_SEC_BIT_DISTR	Clear stored single error correction (SEC) bit distribution history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Cleares the following registers: • STATUS_SEC_ERROR_DISTR_0 • STATUS_SEC_ERROR_DISTR_1 • STATUS_SEC_ERROR_DISTR_2 • STATUS_SEC_ERROR_DISTR_3 • STATUS_SEC_ERROR_DISTR_4	RW	0x0
2	CLEAR_ADDR_ERR_CNT	Clear stored address error history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Clears the STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT bit field and the ADDRERR FIFO	RW	0x0
1	CLEAR_DED_ERR_CNT	Clear stored double error detection (DED) history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Clears the STATUS_ERROR_CNT[19:16] DED_ERROR_CNT bit field and the DED FIFO	RW	0x0
0	CLEAR_SEC_ERR_CNT	Clear stored single error correction history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Clears the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT bit field and the SEC FIFO	RW	0x0

Table 15-596. Register Call Summary for Register CFG_OCMC_ECC_CLEAR_HIST

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2]
- ECC Associated FIFOs: [3] [4] [5]
- ECC Support: [6] [7]
- OCM Subsystem Register Summary: [8]
- OCM Subsystem Register Description: [11] [12] [13]

Table 15-597. STATUS_ERROR_CNT

Address Offset	0x0000 0090		
Physical Address	0x4880 4090 0x4880 A090 0x4881 0090	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	OCM Controller error status		
Type	R		

31–24	23–20	19–16	15–0
RESERVED	ADDR_ERROR_CNT	DED_ERROR_CNT	SEC_ERROR_CNT

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:20	ADDR_ERROR_CNT	Counter for the address errors found. This bit field is reset when 0x1 is written to the CFG_OCMC_ECC_CLEAR_HIST[2] CLEAR_ADDR_ERR_CNT bit.	R	0x0
19:16	DED_ERROR_CNT	Counter for the double error detections. This bit field is reset when 0x1 is written to the CFG_OCMC_ECC_CLEAR_HIST[1] CLEAR_DED_ERR_CNT bit.	R	0x0
15:0	SEC_ERROR_CNT	Counter for the single errors occured. This bit field is reset when 0x1 is written to the CFG_OCMC_ECC_CLEAR_HIST[0] CLEAR_SEC_ERR_CNT bit.	R	0x0

Table 15-598. Register Call Summary for Register STATUS_ERROR_CNT

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0] [1] [2]
- ECC Counters And Corrected Bit Distribution Register: [3] [4] [5]
- ECC Support: [6] [7] [8] [9] [10] [11] [12]
- OCM Subsystem Register Summary: [13]
- OCM Subsystem Register Description: [16] [17] [18]

Table 15-599. STATUS_SEC_ERROR_TRACE

Address Offset	0x0000 0094		
Physical Address	0x4880 4094 0x4880 A094 0x4881 0094	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	SEC error 128-bit memory address		
Type	R		

31–19	18	17–0
RESERVED	VALID	ADDRESS_128BIT

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	VALID	SEC FIFO valid addres indication. 0x0: The SEC FIFO is empty 0x1: There is a valid address in the SEC FIFO	R	0x0
17:0	ADDRESS_128BIT	SEC error 128-bit memory address (Read from the SEC error address trace fifo)	R	0x0

Table 15-600. Register Call Summary for Register STATUS_SEC_ERROR_TRACE

On-Chip Memory (OCM) Subsystem

- ECC Associated FIFOs: [0] [1]
- OCM Subsystem Register Summary: [2]

Table 15-601. STATUS_DED_ERROR_TRACE

Address Offset	0x0000 0098		
Physical Address	0x4880 4098 0x4880 A098 0x4881 0098	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	DED error 128-bit memory address		
Type	R		

31–19	18	17–0
RESERVED	VALID	ADDRESS_128BIT

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	VALID	DED FIFO valid addres indication. 0x0: The DED FIFO is empty 0x1: There is a valid address in the DED FIFO	R	0x0
17:0	ADDRESS_128BIT	DED error 128-bit memory address (Read from the DED error address trace fifo)	R	0x0

Table 15-602. Register Call Summary for Register STATUS_DED_ERROR_TRACE

On-Chip Memory (OCM) Subsystem
- ECC Associated FIFOs: [0] [1] [2]
- OCM Subsystem Register Summary: [3]

Table 15-603. STATUS_ADDR_TRANSLATION_ERROR_TRACE

Address Offset	0x0000 009C		
Physical Address	0x4880 409C 0x4880 A09C 0x4881 009C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	ADDR error 128-bit memory address		
Type	R		

31–19	18	17–0
RESERVED	VALID	ADDRESS_128BIT

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	VALID	ADDRERR FIFO valid addres indication. 0x0: The ADDRERR FIFO is empty 0x1: There is a valid address in the ADDRERR FIFO	R	0x0
17:0	ADDRESS_128BIT	ADDR error 128-bit memory address (Read from the ADDR error address trace fifo)	R	0x0

Table 15-604. Register Call Summary for Register STATUS_ADDR_TRANSLATION_ERROR_TRACE

On-Chip Memory (OCM) Subsystem
- ECC Associated FIFOs: [0] [1] [2]
- OCM Subsystem Register Summary: [3]

Table 15-605. STATUS_SEC_ERROR_DISTR_0

Address Offset	0x0000 00A0

Table 15-605. STATUS_SEC_ERROR_DISTR_0 (continued)

Physical Address	0x4880 40A0 0x4880 A0A0 0x4881 00A0	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	SEC data error bit distribution status [31:0]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 15-606. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_0

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [2]
- OCM Subsystem Register Description: [5]

Table 15-607. STATUS_SEC_ERROR_DISTR_1

Address Offset	0x0000 00A4		
Physical Address	0x4880 40A4 0x4880 A0A4 0x4881 00A4	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	SEC data error bit distribution status [63:32]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 15-608. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_1

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [2]
- OCM Subsystem Register Description: [5]

Table 15-609. STATUS_SEC_ERROR_DISTR_2

Address Offset	0x0000 00A8		
Physical Address	0x4880 40A8 0x4880 A0A8 0x4881 00A8	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	SEC data error bit distribution status [95:64]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 15-610. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_2

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [2]
- OCM Subsystem Register Description: [5]

Table 15-611. STATUS_SEC_ERROR_DISTR_3

Address Offset	0x0000 00AC		
Physical Address	0x4880 40AC 0x4880 A0AC 0x4881 00AC	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	SEC data error bit distribution status [127:96]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 15-612. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_3

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [2]
- OCM Subsystem Register Description: [5]

Table 15-613. STATUS_SEC_ERROR_DISTR_4

Address Offset	0x0000 00B0		
Physical Address	0x4880 40B0 0x4880 A0B0 0x4881 00B0	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	SEC ecc code error bit distribution status [7:0]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SEC_ECC_CODE_ERROR_FOUND							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	SEC_ECC_CODE_ERROR_FOUND	ECC Code (excluding the parity bit) error distribution [7:0]. For each bit: 0x0: SEC error not found 0x1: SEC error found In the corresponding bit location	R	0x0

Table 15-614. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_4

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1] [2]
- OCM Subsystem Register Summary: [3]
- OCM Subsystem Register Description: [6]

Table 15-615. CFG_OCMC_CBUF_EN

Address Offset	0x0000 0200		
Physical Address	0x4880 4200 0x4880 A200 0x4881 0200	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	CBUF mode enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBUF_EN_11	CBUF_EN_10	CBUF_EN_9	CBUF_EN_8	CBUF_EN_7	CBUF_EN_6	CBUF_EN_5	CBUF_EN_4	CBUF_EN_3	CBUF_EN_2	CBUF_EN_1	CBUF_EN_0	RESERVED													NEW_FRAME_SEL	CBUF_DEBUG_EN	CBUF_MODE_EN

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	CBUF_EN_11	CBUF 11 enable. 0x0: Disable 0x1: Enable	RW	0x0
26	CBUF_EN_10	CBUF 10 enable. 0x0: Disable 0x1: Enable	RW	0x0
25	CBUF_EN_9	CBUF 9 enable. 0x0: Disable 0x1: Enable	RW	0x0
24	CBUF_EN_8	CBUF 8 enable. 0x0: Disable 0x1: Enable	RW	0x0
23	CBUF_EN_7	CBUF 7 enable. 0x0: Disable 0x1: Enable	RW	0x0
22	CBUF_EN_6	CBUF 6 enable. 0x0: Disable 0x1: Enable	RW	0x0
21	CBUF_EN_5	CBUF 5 enable. 0x0: Disable 0x1: Enable	RW	0x0
20	CBUF_EN_4	CBUF 4 enable. 0x0: Disable 0x1: Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
19	CBUF_EN_3	CBUF 3 enable. 0x0: Disable 0x1: Enable	RW	0x0
18	CBUF_EN_2	CBUF 2 enable. 0x0: Disable 0x1: Enable	RW	0x0
17	CBUF_EN_1	CBUF 1 enable. 0x0: Disable 0x1: Enable	RW	0x0
16	CBUF_EN_0	CBUF 0 enable. 0x0: Disable 0x1: Enable	RW	0x0
15:3	RESERVED		R	0x0
2	NEW_FRAME_SEL	CBUF New Frame Event Definition Select. 0x0: New frame event flag is set when a VBUF access is made to the base address of the VBUF 0x1: New frame event flag is set when a VBUF access is made to the base CBUF slice address range of the VBUF	RW	0x0
1	CBUF_DEBUG_EN	CBUF Debug Enable Mode. 0x0: Default Normal mode. All CBUF accesses with MReqDebug=1 are rejected. 0x1: Debug mode. MReqDebug Interconnect qualifier is ignored.	RW	0x0
0	CBUF_MODE_EN	CBUF Mode Enable. 0x0: Disable all CBUF address translation 0x1: Enable CBUF address translation	RW	0x0

Table 15-616. Register Call Summary for Register CFG_OCMC_CBUF_EN

On-Chip Memory (OCM) Subsystem

- Circular Buffer (CBUF) Support: [0] [1] [2] [3]
- OCM Subsystem Register Summary: [4]

Table 15-617. CFG_OCMC_CBUF_RESET

Address Offset	0x0000 0204		
Physical Address	0x4880 4204 0x4880 A204 0x4881 0204	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description	Writing 1 to bit n will set a reset bit to clear the corresponding CBUF_n address translation logic. Sliding CBUF frame tracking will be cleared so that the CBUF now points to the base of the virtual frame buffer. Normally, a reset is not required since the CBUF logic will clear itself when a VBUF access is to the base of the virtual frame.		
Type	RW		

31:12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CBUF_RESET_11	CBUF_RESET_10	CBUF_RESET_9	CBUF_RESET_8	CBUF_RESET_7	CBUF_RESET_6	CBUF_RESET_5	CBUF_RESET_4	CBUF_RESET_3	CBUF_RESET_2	CBUF_RESET_1	CBUF_RESET_0

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	CBUF_RESET_11	cbuf_reset_11	RW W1toClr	0x0
10	CBUF_RESET_10	cbuf_reset_10	RW W1toClr	0x0
9	CBUF_RESET_9	cbuf_reset_9	RW W1toClr	0x0
8	CBUF_RESET_8	cbuf_reset_8	RW W1toClr	0x0
7	CBUF_RESET_7	cbuf_reset_7	RW W1toClr	0x0
6	CBUF_RESET_6	cbuf_reset_6	RW W1toClr	0x0
5	CBUF_RESET_5	cbuf_reset_5	RW W1toClr	0x0
4	CBUF_RESET_4	cbuf_reset_4	RW W1toClr	0x0
3	CBUF_RESET_3	cbuf_reset_3	RW W1toClr	0x0
2	CBUF_RESET_2	cbuf_reset_2	RW W1toClr	0x0
1	CBUF_RESET_1	cbuf_reset_1	RW W1toClr	0x0
0	CBUF_RESET_0	cbuf_reset_0	RW W1toClr	0x0

Table 15-618. Register Call Summary for Register CFG_OCMC_CBUF_RESET

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [0]

Table 15-619. CFG_OCMC_CBUF_ERR_HANDLER

Address Offset	0x0000 0208		
Physical Address	0x4880 4208 0x4880 A208 0x4881 0208	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	UNDERFLOW_LAST_CBUF_SLICE_DISABLE	OVERFLOW_CHECK_REENABLE_SEL	OVERFLOW_WRITE_HANDLER_SEL	UNDERFLOW_ERR_CHECK_EN	OVERFLOW_ERR_CHECK_EN	SHORT_FRAME_PREV_EOF_SEL	SHORT_FRAME_DETECT_CHECK_EN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	UNDERFLOW_LAST_CBUF_SLICE_DISABLE	0x0: Check underflow even when read is from the last CBUF slice 0x1: Disable underflow check when read is from the last CBUF slice	RW	0x0
7:6	OVERFLOW_CHECK_REENABLE_SEL	Overflow check re-enable selection. 0x0: Overflow check is disabled until next wtire to or read from virtual frame start address is detected 0x1: Overflow check is disabled until next write to virtual frame start address is detected 0x2: Overflow check is disabled until next read from virtual frame start address is detected 0x3: Overflow check is re-enabled immediately	RW	0x0
5:4	OVERFLOW_WRITE_HANDLER_SEL	Overflow write handler selection. 0x0: Writes disabled only on CBUF_overflow_wrap cases until next write to virtual frame start address is detected 0x1: Writes disabled on all overflow cases until next write to virtual frame start address is detected 0x2: Writes serviced with CBUF pointer updated even on overflow condition 0x3: Reserved	RW	0x0
3	UNDERFLOW_ERR_CHECK_EN	Underflow chek enable. 0x0: Underflow check enabled 0x1: Underflow check disabled	RW	0x0
2	OVERFLOW_ERR_CHECK_EN	Overflow chek enable. 0x0: Overflow check enabled 0x1: Overflow check disabled	RW	0x0
1	SHORT_FRAME_PREV_EOF_SEL	0x0: previous frame EOF history is set if the last write address is equal to the VBUF frame end address 0x1: previous frame EOF history is set if the last write address is in the Last CBUF slice	RW	0x0
0	SHORT_FRAME_DETECT_CHECK_EN	Short frame detection enable. 0x0: Detection enabled 0x1: Detection disabled	RW	0x0

Table 15-620. Register Call Summary for Register CFG_OCMC_CBUF_ERR_HANDLER

On-Chip Memory (OCM) Subsystem

- Illegal Frame SIze (Short Frame Detection): [0] [1]
- CBUF Overflow: [2] [3] [4]
- CBUF Underflow: [5] [6]
- OCM Subsystem Register Summary: [7]

Table 15-621. STATUS_CBUF_WR_OUT_OF_RANGE_ERR

Address Offset	0x0000 020C		
Physical Address	0x4880 420C 0x4880 A20C 0x4881 020C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	Indicates that the CBUF write address is out of the CBUF range. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-622. Register Call Summary for Register STATUS_CBUF_WR_OUT_OF_RANGE_ERR

On-Chip Memory (OCM) Subsystem

- VBUF Address Not Mapped to a CBUF Memory Space: [0] [1]
- OCM Subsystem Register Summary: [2]

Table 15-623. STATUS_CBUF_WR_VBUF_START_ERR

Address Offset	0x0000 0210		
Physical Address	0x4880 4210 0x4880 A210 0x4881 0210	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF write is not to the base address at vbuf access start. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-624. Register Call Summary for Register STATUS_CBUF_WR_VBUF_START_ERR

On-Chip Memory (OCM) Subsystem

- VBUF Access Not Starting At The Base Address: [0] [1]
- OCM Subsystem Register Summary: [2]

Table 15-625. STATUS_CBUF_WR_ADDR_SEQ_ERROR

Address Offset	0x0000 0214		
Physical Address	0x4880 4214 0x4880 A214 0x4881 0214	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF address is not incrementing in raster scan order. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-626. Register Call Summary for Register STATUS_CBUF_WR_ADDR_SEQ_ERROR

On-Chip Memory (OCM) Subsystem

- Illegal Address Change Between Two Same Type Accesses: [0] [1]
- OCM Subsystem Register Summary: [2]

Table 15-627. STATUS_CBUF_RD_OUT_OF_RANGE_ERROR

Address Offset	0x0000 0218		
Physical Address	0x4880 4218 0x4880 A218 0x4881 0218	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	Indicates that the CBUF read address is out of the CBUF range. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-628. Register Call Summary for Register STATUS_CBUF_RD_OUT_OF_RANGE_ERROR

On-Chip Memory (OCM) Subsystem

- VBUF Address Not Mapped to a CBUF Memory Space: [0] [1]
- OCM Subsystem Register Summary: [2]

Table 15-629. STATUS_CBUF_VBUF_RD_START_ERROR

Address Offset	0x0000 021C		
Physical Address	0x4880 421C 0x4880 A21C 0x4881 021C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF read is not from the base address at VBUF access start. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-630. Register Call Summary for Register STATUS_CBUF_VBUF_RD_START_ERROR

On-Chip Memory (OCM) Subsystem

- VBUF Access Not Starting At The Base Address: [0] [1]
- OCM Subsystem Register Summary: [2]

Table 15-631. STATUS_CBUF_RD_ADDR_SEQ_ERROR

Address Offset	0x0000 0220		
Physical Address	0x4880 4220 0x4880 A220 0x4881 0220	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF read address is not incrementing in raster scan order. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-632. Register Call Summary for Register STATUS_CBUF_RD_ADDR_SEQ_ERROR

On-Chip Memory (OCM) Subsystem

- Illegal Address Change Between Two Same Type Accesses: [0] [1]
- OCM Subsystem Register Summary: [2]

Table 15-633. STATUS_CBUF_OVERFLOW_MID

Address Offset	0x0000 0224		
Physical Address	0x4880 4224 0x4880 A224 0x4881 0224	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF overflow condition detected in the middle of a frame. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-634. Register Call Summary for Register STATUS_CBUF_OVERFLOW_MID

On-Chip Memory (OCM) Subsystem

- CBUF Overflow: [0]
- OCM Subsystem Register Summary: [1]

Table 15-635. STATUS_CBUF_OVERFLOW_WRAP

Address Offset	0x0000 0228		
Physical Address	0x4880 4228 0x4880 A228 0x4881 0228	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF overflow condition detected during buffer switching. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-636. Register Call Summary for Register STATUS_CBUF_OVERFLOW_WRAP

On-Chip Memory (OCM) Subsystem

- CBUF Overflow: [0]
- OCM Subsystem Register Summary: [1]

Table 15-637. STATUS_CBUF_UNDERFLOW

Address Offset	0x0000 022C		
Physical Address	0x4880 422C 0x4880 A22C 0x4881 022C	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF underflow condition detected. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-638. Register Call Summary for Register STATUS_CBUF_UNDERFLOW

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0]
- CBUF Underflow: [1] [2] [3]
- OCM Subsystem Register Summary: [4]

Table 15-639. STATUS_CBUF_SHORT_FRAME_DETECT

Address Offset	0x0000 0230		
Physical Address	0x4880 4230 0x4880 A230 0x4881 0230	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:12	11:0
RESERVED	CBUF_ERR

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF short frame detected. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 15-640. Register Call Summary for Register STATUS_CBUF_SHORT_FRAME_DETECT

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0]
- Illegal Frame SIze (Short Frame Detection): [1] [2]
- OCM Subsystem Register Summary: [3]

Table 15-641. CBUF_i_VBUF_START_ADDR

Address Offset	0x0000 0240 + (i*16); i = 0 to 11		
Physical Address	0x4880 4240 + (i*16) 0x4880 A240 + (i*16) 0x4881 0240 + (i*16)	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ADDR																				RESERVED			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	The virtual address range is determined by the OCMC_MEM_SIZE_READ[16:12] VBUF_ADDR_MSB bit field. For default value of 23, the valid VBUF address is 0x80xx_xxxx. Writing to this field above the MSB bit returned by VBUF_ADDR_MSB will be ignored and reading will return all zeroes except for bit[31] = 1.	R	0x80
23:4	ADDR	Virtual frame start address for this CBUF - bits [23:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 15-642. Register Call Summary for Register CBUF_i_VBUF_START_ADDR

On-Chip Memory (OCM) Subsystem

- Circular Buffer (CBUF) Support: [0]
- VBUF Access Not Starting At The Base Address: [1] [2]
- OCM Subsystem Register Summary: [3]

Table 15-643. CBUF_i_VBUF_END_ADDR

Address Offset	0x0000 0244 + (i*16); i = 0 to 11		
Physical Address	0x4880 4244 + (i*16) 0x4880 A244 + (i*16) 0x4881 0244 + (i*16)	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ADDR																				RESERVED			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	The virtual address range is determined by the OCMC_MEM_SIZE_READ[16:12] VBUF_ADDR_MSB bit field. For default value of 23, the valid VBUF address is 0x80xx_xxxx. Writing to this field above the MSB bit returned by VBUF_ADDR_MSB will be ignored and reading will return all zeroes except for bit[31] = 1.	R	0x80
23:4	ADDR	Virtual frame end address for this CBUF - bits [23:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 15-644. Register Call Summary for Register CBUF_i_VBUF_END_ADDR

On-Chip Memory (OCM) Subsystem

- Circular Buffer (CBUF) Support: [0]
- OCM Subsystem Register Summary: [1]

Table 15-645. CBUF_i_OCMC_START_ADDR

Address Offset	0x0000 0248 + (i*16); i = 0 to 11		
Physical Address	0x4880 4248 + (i*16) 0x4880 A248 + (i*16) 0x4881 0248 + (i*16)	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:22	21:4	3:0
RESERVED	ADDR	RESERVED

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:4	ADDR	SRAM start address for this CBUF - bits [21:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 15-646. Register Call Summary for Register CBUF_i_OCMC_START_ADDR

On-Chip Memory (OCM) Subsystem

- Circular Buffer (CBUF) Support: [0]
- OCM Subsystem Register Summary: [1]

Table 15-647. CBUF_i_OCMC_BUF_SIZE

Address Offset	0x0000 024C + (i*16); i = 0 to 11		
Physical Address	0x4880 424C + (i*16) 0x4880 A24C + (i*16) 0x4881 024C + (i*16)	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	RW		

31:20	19:4	3:0
RESERVED	BUF_SIZE	RESERVED

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:4	BUF_SIZE	SRAM size allocated for this CBUF - bits [19:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 15-648. Register Call Summary for Register CBUF_i_OCMC_BUF_SIZE

On-Chip Memory (OCM) Subsystem

- Circular Buffer (CBUF) Support: [0]
- OCM Subsystem Register Summary: [1]

Table 15-649. CBUF_k_LAST_WR_ADDR

Address Offset	0x0000 0300 + (k*8); k = 0 to 11		
Physical Address	0x4880 4300 + (k*8) 0x4880 A300 + (k*8) 0x4881 0300 + (k*8)	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			

Table 15-649. CBUF_k_LAST_WR_ADDR (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Last virtual write address accessing CBUF	R	0x0

Table 15-650. Register Call Summary for Register CBUF_k_LAST_WR_ADDR

On-Chip Memory (OCM) Subsystem

- Status Reporting: [0]
- OCM Subsystem Register Summary: [1]

Table 15-651. CBUF_k_LAST_RD_ADDR

Address Offset	0x0000 0304 + (k*8); k = 0 to 11		
Physical Address	0x4880 4304 + (k*8) 0x4880 A304 + (k*8) 0x4881 0304 + (k*8)	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Last virtual read address accessing CBUF	R	0x0

Table 15-652. Register Call Summary for Register CBUF_k_LAST_RD_ADDR

On-Chip Memory (OCM) Subsystem

- Status Reporting: [0]
- OCM Subsystem Register Summary: [1]

Table 15-653. LAST_ILLEGAL_OCMC_ADDR

Address Offset	0x0000 0360		
Physical Address	0x4880 4360 0x4880 A360 0x4881 0360	**Instance**	OCMC_RAM1 OCMC_RAM2 OCMC_RAM3
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Last Illegal OCMC Address. This register returns the OCMC L3_MAIN address of the last access that was invalidated due to an OUT_OF_RANGE_ERR_FOUND (non-VBUF address) error or any one of the CBUF related access errors (including any write access disabled during overflow error handling).	R	0x0

Table 15-654. Register Call Summary for Register LAST_ILLEGAL_OCMC_ADDR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0]
- VBUF Address Not Mapped to a CBUF Memory Space: [1]
- OCM Subsystem Register Summary: [2]

Chapter 16

DMA Controllers

This chapter describes the features and operation of the device System DMA (DMA_SYSTEM) and Enhanced DMA (EDMA) controllers.

16.1 System DMA....3659
16.2 Enhanced DMA....3732

16.1 System DMA

This chapter describes the system direct memory access (DMA_SYSTEM) module.

16.1.1 DMA_SYSTEM Module Overview

The system direct memory access (DMA_SYSTEM) module, also called DMA4, performs high-performance data transfers between memories and peripheral devices without microprocessor unit (MPU) or digital signal processor (DSP) support during transfer. A DMA transfer is programmed through a logical DMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

The DMA controller includes the following main features:

- Data transfer support in either direction between:
  - Memory and memory
  - Memory and peripheral device
- 32 logical DMA channels supporting:
  - Multiple concurrent transfers
  - Independent transfer profile for each channel
  - 8-bit, 16-bit, or 32-bit data element transfer size
  - Software-triggered or hardware-synchronized transfers
  - Flexible source and destination address generation
  - Burst read and write - max burst size is 16
  - Chained multiple-channel transfers
  - Endianism conversion
  - Draining
  - Linked-list support for descriptor types 1, 2, and 3
- First-come, first-serve DMA scheduling with fixed priority
- Up to 127 Hardware DMA requests
- DMA_CROSSBAR
- Constant fill
- Transparent copy
- Four programmable interrupt request output lines
- FIFO depth: 256 × 64-bit
- Data buffering
- FIFO budget allocation
- Power-management support
- Auto-idle power-saving support

Figure 16-1 shows an overview of the DMA_SYSTEM module.

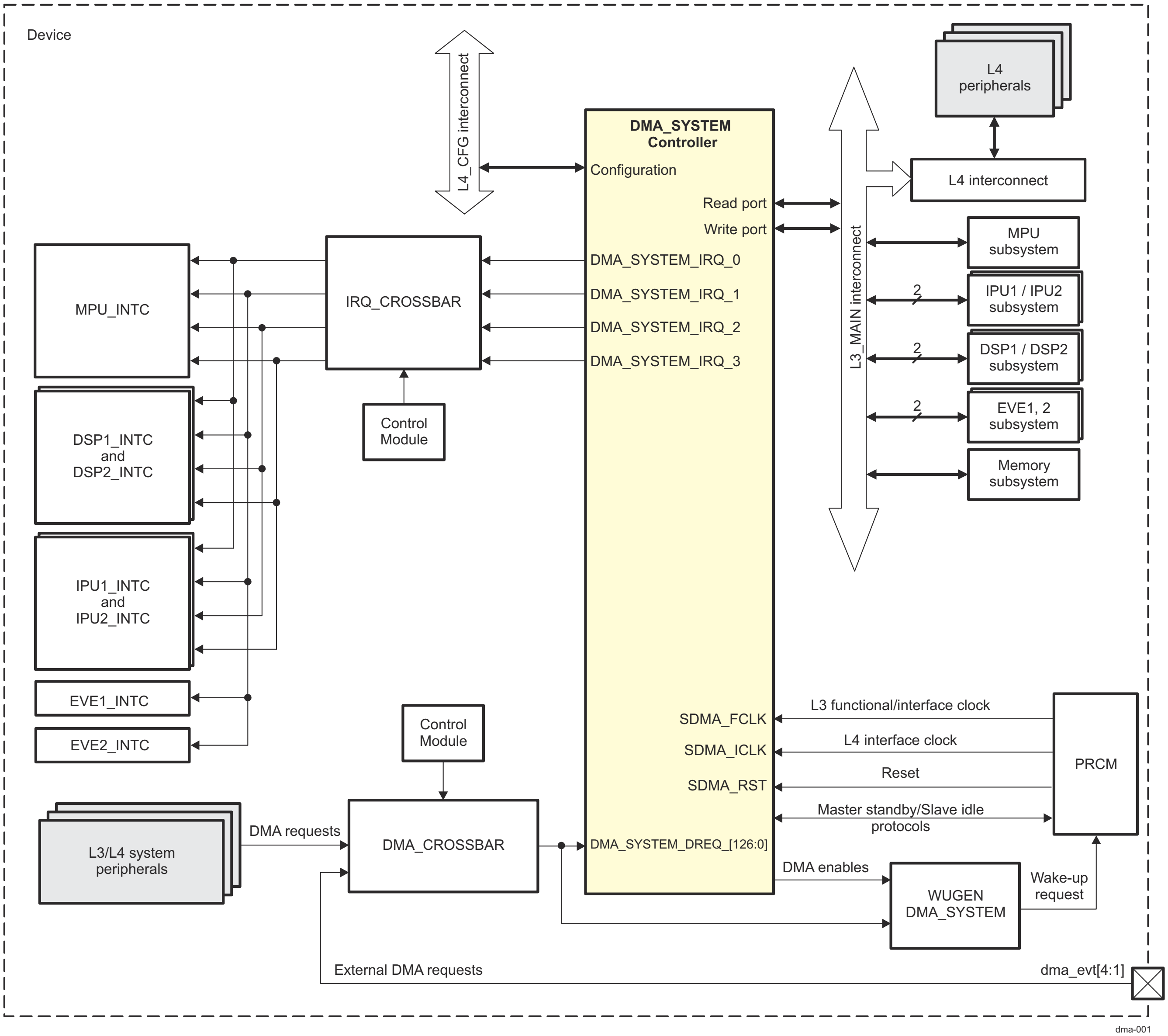


Figure 16-1. DMA_SYSTEM Overview

The DMA_SYSTEM module has three ports: one read, one write, and one configuration port, and provides multiple logical channel support. A dynamically allocated FIFO queue memory pool provides buffering between the read and write ports, which are multithreaded (two threads for the write port and four threads for the read port); this means that each transaction is flagged by a thread ID (0, 1, 2, or 3) in the request direction and in the response direction. This allows the read port to have four outstanding requests at a time. The write port has two threads budget available.

The MPU (or DSP) configures the DMA_SYSTEM through the L4_CFG interconnect.

16.1.2 DMA_SYSTEM Controller Environment

The DMA_SYSTEM controller supports external DMA requests through the dma_evt[4:1] pins (see Table 16-1). A logical channel can be configured to respond to an external synchronization request.

Table 16-1. External DMA_SYSTEM Request Signals

Pin Name	DMA_CROSSBAR Input	Signal Name	I/O[1]	Description	Module Reset Value
dma_evt1	DMA_CROSSBAR_2	EXT_SYS_DREQ_0	I	External DMA request 0 (system expansion)	Z
dma_evt2	DMA_CROSSBAR_3	EXT_SYS_DREQ_1	I	External DMA request 1 (system expansion)	Z
dma_evt3	DMA_CROSSBAR_167	EXT_SYS_DREQ_2	I	External DMA request 2 (system expansion)	Z
dma_evt4	DMA_CROSSBAR_168	EXT_SYS_DREQ_3	I	External DMA request 3 (system expansion)	Z

(1) I = Input, O = Output

Figure 16-2 shows an example of how to use the external hardware DMA request pins in the DMA_SYSTEM environment.

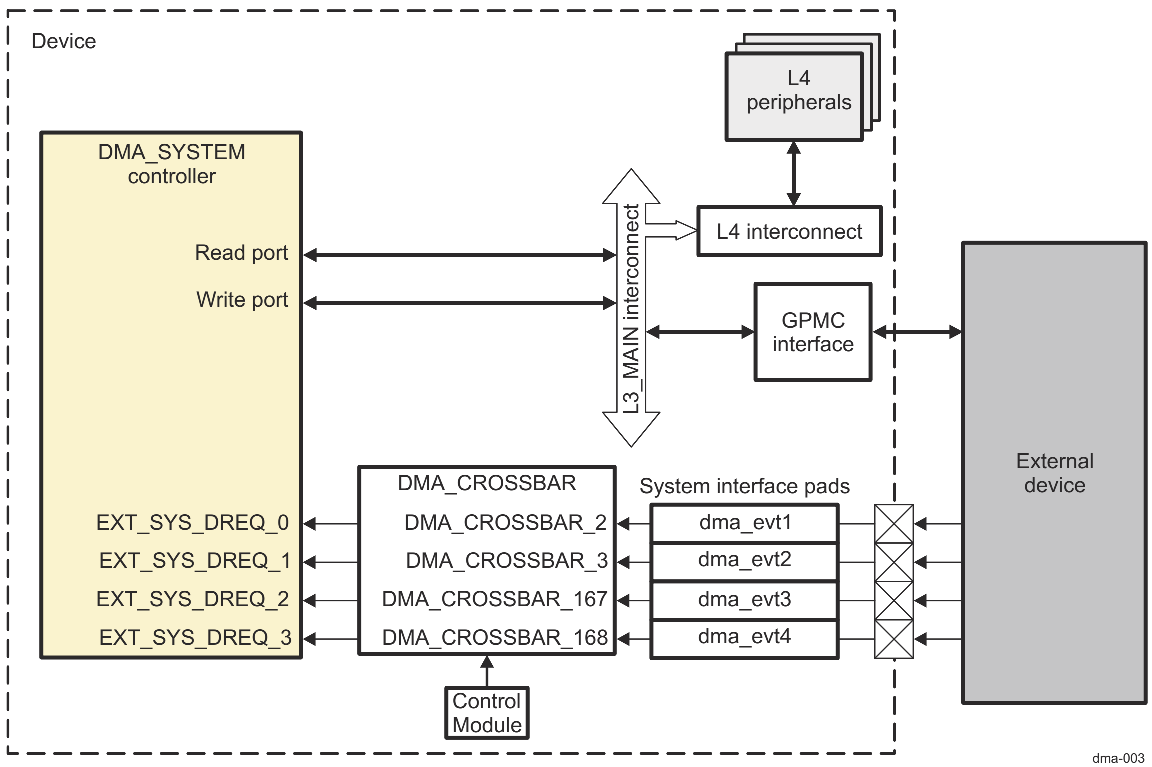


Figure 16-2. Example of External DMA Requests Use

An external device can use the external DMA request pins to start a logical channel transfer over the general-purpose memory controller (GPMC) interface. The transfer can be a memory-to-memory transfer in which the source memory is in the external device.

By default, the external DMA request signals are not available on external pins after a cold reset. For more information about multiplexing out the two signal lines to pins, see *Pad Configuration Registers* in the *Control Module*.

All 127 DMA request lines are transition sensitive.

For a transition-sensitive DMA request (see Figure 16-3), the line must be maintained low (asserted) until the first DMA access is complete, after which the line must be maintained high (deasserted) for greater than one clock cycle (DMA_L3_GICLK):

- When the deassertion time is less than one clock cycle, the DMA_SYSTEM may not detect the deassertion.
- When the channel is enabled one cycle after a DMA request is disabled, the channel detects the DMA request and starts the corresponding transfer.
- When the channel is enabled two cycles after the DMA request is disabled, the channel does not detect the DMA request.

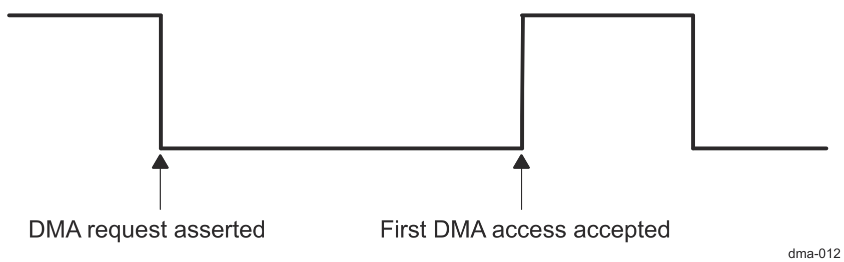
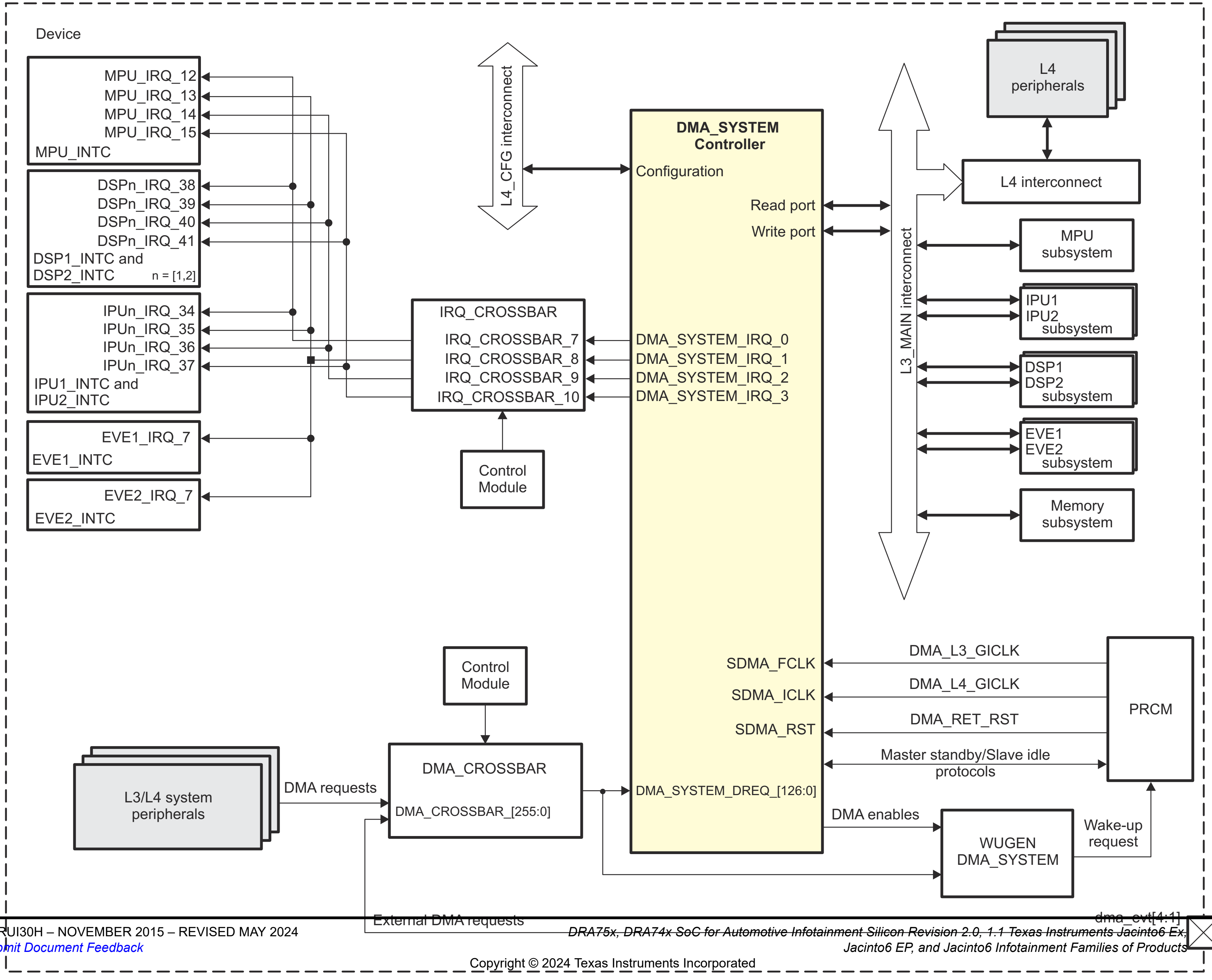


Figure 16-3. Transition-Sensitive DMA Request Scheme

16.1.3 DMA_SYSTEM Module Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 16-4 shows the DMA_SYSTEM controller integration.


Device
MPU_IRQ_12
MPU_IRQ_13
MPU_IRQ_14
MPU_IRQ_15
MPU_INTC
DSPn_IRQ_38
DSPn_IRQ_39
DSPn_IRQ_40
DSPn_IRQ_41
DSP1_INTC and
DSP2_INTC
n = [1,2]
IPUn_IRQ_34
IPUn_IRQ_35
IPUn_IRQ_36
IPUn_IRQ_37
IPU1_INTC and
IPU2_INTC
EVE1_IRQ_7
EVE1_INTC
EVE2_IRQ_7
EVE2_INTC
L4_CFG interconnect
IRQ_CROSSBAR
IRQ_CROSSBAR_7
IRQ_CROSSBAR_8
IRQ_CROSSBAR_9
IRQ_CROSSBAR_10
Control Module
DMA_SYSTEM Controller
Configuration
Read port
Write port
DMA_SYSTEM_IRQ_0
DMA_SYSTEM_IRQ_1
DMA_SYSTEM_IRQ_2
DMA_SYSTEM_IRQ_3
SDMA_FCLK
SDMA_ICLK
SDMA_RST
DMA_SYSTEM_DREQ_[126:0]
L3_MAIN interconnect
L4 peripherals
L4 interconnect
MPU subsystem
IPU1
IPU2
subsystem
DSP1
DSP2
subsystem
EVE1
EVE2
subsystem
Memory subsystem
DMA_L3_GICLK
DMA_L4_GICLK
DMA_RET_RST
Master standby/Slave idle protocols
PRCM
DMA enables
WUGEN DMA_SYSTEM
Wake-up request
Control Module
DMA_CROSSBAR
DMA_CROSSBAR_[255:0]
DMA requests
L3/L4 system peripherals
External DMA requests
dma_evt[4:1]
dma-002

Note

For more information about the system DMA wake-up generator (WUGEN_DMA_SYSTEM), the master standby/slave idle protocols, and the wake-up request, see *Clock Management*, in *Power, Reset, and Clock Management*.

Table 16-2 through Table 16-4 summarize the integration of the module in the device.

Table 16-2. DMA_SYSTEM Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
DMA_SYSTEM	PD_COREAON	L3_MAIN and L4_CFG

Table 16-3. DMA_SYSTEM Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DMA_SYSTEM	SDMA_FCLK	DMA_L3_GICLK	PRCM	Functional clock for all internal logic and for the two master read and write ports. For information about the power, reset, and clock management (PRCM) clock gating and management, see *Power, Reset, and Clock Management*.
	SDMA_ICLK	DMA_L4_GICLK		Interface clock. It supports the configuration port. For information about PRCM clock gating and management, see *Power, Reset, and Clock Management*.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DMA_SYSTEM	SDMA_RST	DMA_RET_RST	PRCM	Hardware retention reset. It initializes all internal logic of the DMA_SYSTEM module, all global registers, and some of the per-channel registers, implemented in flip-flops. However, all remaining per-channel registers are memory-based, and, therefore, are not reset (have undefined values). Thus, when programming a channel for the first time, all bits that have undefined reset values must be configured before enabling the channel. For information about PRCM reset sources and distribution, see *Power, Reset, and Clock Management*.

Table 16-4. DMA_SYSTEM Hardware Requests

Interrupt Requests					
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR INPUT**	**Default Mapping**	**Destination**	**Description**
DMA_SYSTEM	DMA_SYSTEM_IRQ_0	IRQ_CROSSBAR_7	MPU_IRQ_12	Cortex-A15 MPU INTC	DMA_SYSTEM interrupt request 0. For information about the MPU_INTC, see *Interrupt Requests to MPU_INTC*.
			IPU1_IRQ_34	IPU1 INTC	DMA_SYSTEM interrupt request 0. For information about the IPU1_INTC, see *Interrupt Requests to IPU1_Cx_INTC*.
			IPU2_IRQ_34	IPU2 INTC	DMA_SYSTEM interrupt request 0. For information about the IPU2_INTC, see *Interrupt Requests to IPU2_Cx_INTC*.
			DSP1_IRQ_38	DSP1 INTC	DMA_SYSTEM interrupt request 0. For information about the DSP1_INTC, see *Interrupt Requests to DSP1_INTC*.
			DSP2_IRQ_38	DSP2 INTC	DMA_SYSTEM interrupt request 0. For information about the DSP2_INTC, see *Interrupt Requests to DSP2_INTC*.
	DMA_SYSTEM_IRQ_1	IRQ_CROSSBAR_8	MPU_IRQ_13	Cortex-A15 MPU INTC	DMA_SYSTEM interrupt request 1
			IPU1_IRQ_35	IPU1 INTC	
			IPU2_IRQ_35	IPU2 INTC	
			DSP1_IRQ_39	DSP1 INTC	
			DSP2_IRQ_39	DSP2 INTC	
			EVE1_IRQ_7	EVE1 INTC	DMA_SYSTEM interrupt request 1. For information about the EVE1_INTC, see *Interrupt Requests to EVE1_INTC*.
			EVE2_IRQ_7	EVE2 INTC	DMA_SYSTEM interrupt request 1. For information about the EVE2_INTC, see *Interrupt Requests to EVE2_INTC*.
	DMA_SYSTEM_IRQ_2	IRQ_CROSSBAR_9	MPU_IRQ_14	Cortex-A15 MPU INTC	DMA_SYSTEM interrupt request 2
			IPU1_IRQ_36	IPU1 INTC	
			IPU2_IRQ_36	IPU2 INTC	
			DSP1_IRQ_40	DSP1 INTC	
			DSP2_IRQ_40	DSP2 INTC	
	DMA_SYSTEM_IRQ_3	IRQ_CROSSBAR_10	MPU_IRQ_15	Cortex-A15 MPU INTC	DMA_SYSTEM interrupt request 3
			IPU1_IRQ_37	IPU1 INTC	
			IPU2_IRQ_37	IPU2 INTC	
			DSP1_IRQ_41	DSP1 INTC	

Table 16-4. DMA_SYSTEM Hardware Requests (continued)

	DSP2_IRQ_41	DSP2 INTC

Note

The "**Default Mapping**" column in Table 16-4 *Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR or DMA_CROSSBAR modules.

For more information about the IRQ_CROSSBAR and DMA_CROSSBAR modules, see sections: *IRQ_CROSSBAR Module Functional Description* and *DMA_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, refer to *Interrupt Controllers* in the device TRM.

Note

For a description of the interrupt source, see Section 16.1.4.2, *DMA_SYSTEM Controller Interrupt Requests*.

16.1.3.1 DMA Requests to the DMA_SYSTEM Controller

Table 16-5 lists the default DMA sources for the DMA_SYSTEM controller. In addition, the DMA_SYSTEM inputs (DMA_SYSTEM_DREQ_[126:0]) can alternatively be sourced through the associated DMA_CROSSBAR from one of the 256 multiplexed device DMA sources listed in Table 16-6. The CTRL_CORE_DMA_SYSTEM_DREQ_y_z registers (where *y* and *z* are indexes of DMA_SYSTEM input lines) in the Control Module are used to select between the default DMA sources and the multiplexed DMA sources.

Table 16-5. DMA_SYSTEM Default Request Mapping

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_SYSTEM_DREQ_0	1	CTRL_CORE_DMA_SYSTEM_DREQ_0_1[7:0]	1	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_1	2	CTRL_CORE_DMA_SYSTEM_DREQ_0_1[23:16]	2	EXT_SYS_DREQ_0	External DMA request 0 (system expansion)
DMA_SYSTEM_DREQ_2	3	CTRL_CORE_DMA_SYSTEM_DREQ_2_3[7:0]	3	EXT_SYS_DREQ_1	External DMA request 1 (system expansion)
DMA_SYSTEM_DREQ_3	4	CTRL_CORE_DMA_SYSTEM_DREQ_2_3[23:16]	4	GPMC_DREQ	GPMC data transmit request from prefetch engine
DMA_SYSTEM_DREQ_4	5	CTRL_CORE_DMA_SYSTEM_DREQ_4_5[7:0]	5	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_5	6	CTRL_CORE_DMA_SYSTEM_DREQ_4_5[23:16]	6	DISPC_DREQ	Frame update request
DMA_SYSTEM_DREQ_6	7	CTRL_CORE_DMA_SYSTEM_DREQ_6_7[7:0]	7	CT_TBR_DREQ	DEBUG subsystem CT_TBR request
DMA_SYSTEM_DREQ_7	8	CTRL_CORE_DMA_SYSTEM_DREQ_6_7[23:16]	8	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_8	9	CTRL_CORE_DMA_SYSTEM_DREQ_8_9[7:0]	9	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_9	10	CTRL_CORE_DMA_SYSTEM_DREQ_8_9[23:16]	10	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_10	11	CTRL_CORE_DMA_SYSTEM_DREQ_10_11[7:0]	11	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_11	12	CTRL_CORE_DMA_SYSTEM_DREQ_10_11[23:16]	12	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_12	13	CTRL_CORE_DMA_SYSTEM_DREQ_12_13[7:0]	13	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_13	14	CTRL_CORE_DMA_SYSTEM_DREQ_12_13[23:16]	14	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_14	15	CTRL_CORE_DMA_SYSTEM_DREQ_14_15[7:0]	15	MCSPI3_DREQ_TX0	McSPI3 transmit request channel 0
DMA_SYSTEM_DREQ_15	16	CTRL_CORE_DMA_SYSTEM_DREQ_14_15[23:16]	16	MCSPI3_DREQ_RX0	McSPI3 receive request channel 0
DMA_SYSTEM_DREQ_16	17	CTRL_CORE_DMA_SYSTEM_DREQ_16_17[7:0]	17	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_17	18	CTRL_CORE_DMA_SYSTEM_DREQ_16_17[23:16]	18	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_18	19	CTRL_CORE_DMA_SYSTEM_DREQ_18_19[7:0]	19	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_19	20	CTRL_CORE_DMA_SYSTEM_DREQ_18_19[23:16]	20	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_20	21	CTRL_CORE_DMA_SYSTEM_DREQ_20_21[7:0]	21	Reserved	Reserved by default but can be remapped to a valid DMA source

Table 16-5. DMA_SYSTEM Default Request Mapping (continued)

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_SYSTEM_DREQ_21	22	CTRL_CORE_DMA_SYSTEM_DREQ_20_21[23:16]	22	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_22	23	CTRL_CORE_DMA_SYSTEM_DREQ_22_23[7:0]	23	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_23	24	CTRL_CORE_DMA_SYSTEM_DREQ_22_23[23:16]	24	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_22	23	CTRL_CORE_DMA_SYSTEM_DREQ_22_23[7:0]	23	MCSPI3_DREQ_TX1	McSPI module 3 - transmit request channel 1
DMA_SYSTEM_DREQ_23	24	CTRL_CORE_DMA_SYSTEM_DREQ_22_23[23:16]	24	MCSPI3_DREQ_RX1	McSPI module 3 - receive request channel 1
DMA_SYSTEM_DREQ_24	25	CTRL_CORE_DMA_SYSTEM_DREQ_24_25[7:0]	25	I2C3_DREQ_TX	I2C3 transmit request
DMA_SYSTEM_DREQ_25	26	CTRL_CORE_DMA_SYSTEM_DREQ_24_25[23:16]	26	I2C3_DREQ_RX	I2C3 receive request
DMA_SYSTEM_DREQ_26	27	CTRL_CORE_DMA_SYSTEM_DREQ_26_27[7:0]	27	I2C1_DREQ_TX	I2C1 transmit request
DMA_SYSTEM_DREQ_27	28	CTRL_CORE_DMA_SYSTEM_DREQ_26_27[23:16]	28	I2C1_DREQ_RX	I2C1 receive request
DMA_SYSTEM_DREQ_28	29	CTRL_CORE_DMA_SYSTEM_DREQ_28_29[7:0]	29	I2C2_DREQ_TX	I2C2 transmit request
DMA_SYSTEM_DREQ_29	30	CTRL_CORE_DMA_SYSTEM_DREQ_28_29[23:16]	30	I2C2_DREQ_RX	I2C2 receive request
DMA_SYSTEM_DREQ_30	31	CTRL_CORE_DMA_SYSTEM_DREQ_30_31[7:0]	31	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_31	32	CTRL_CORE_DMA_SYSTEM_DREQ_30_31[23:16]	32	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_32	33	CTRL_CORE_DMA_SYSTEM_DREQ_32_33[7:0]	33	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_33	34	CTRL_CORE_DMA_SYSTEM_DREQ_32_33[23:16]	34	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_34	35	CTRL_CORE_DMA_SYSTEM_DREQ_34_35[7:0]	35	MCSPI1_DREQ_TX0	McSPI1 transmit request channel 0
DMA_SYSTEM_DREQ_35	36	CTRL_CORE_DMA_SYSTEM_DREQ_34_35[23:16]	36	MCSPI1_DREQ_RX0	McSPI1 receive request channel 0
DMA_SYSTEM_DREQ_36	37	CTRL_CORE_DMA_SYSTEM_DREQ_36_37[7:0]	37	MCSPI1_DREQ_TX1	McSPI1 transmit request channel 1
DMA_SYSTEM_DREQ_37	38	CTRL_CORE_DMA_SYSTEM_DREQ_36_37[23:16]	38	MCSPI1_DREQ_RX1	McSPI1 receive request channel 1
DMA_SYSTEM_DREQ_38	39	CTRL_CORE_DMA_SYSTEM_DREQ_38_39[7:0]	39	MCSPI1_DREQ_TX2	McSPI1 transmit request channel 2
DMA_SYSTEM_DREQ_39	40	CTRL_CORE_DMA_SYSTEM_DREQ_38_39[23:16]	40	MCSPI1_DREQ_RX2	McSPI1 receive request channel 2
DMA_SYSTEM_DREQ_40	41	CTRL_CORE_DMA_SYSTEM_DREQ_40_41[7:0]	41	MCSPI1_DREQ_TX3	McSPI1 transmit request channel 3
DMA_SYSTEM_DREQ_41	42	CTRL_CORE_DMA_SYSTEM_DREQ_40_41[23:16]	42	MCSPI1_DREQ_RX3	McSPI1 receive request channel 3
DMA_SYSTEM_DREQ_42	43	CTRL_CORE_DMA_SYSTEM_DREQ_42_43[7:0]	43	MCSPI2_DREQ_TX0	McSPI2 transmit request channel 0

Table 16-5. DMA_SYSTEM Default Request Mapping (continued)

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_SYSTEM_DREQ_43	44	CTRL_CORE_DMA_SYSTEM_DREQ_42_43[23:16]	44	MCSPI2_DREQ_RX0	McSPI2 receive request channel 0
DMA_SYSTEM_DREQ_44	45	CTRL_CORE_DMA_SYSTEM_DREQ_44_45[7:0]	45	MCSPI2_DREQ_TX1	McSPI2 transmit request channel 1
DMA_SYSTEM_DREQ_45	46	CTRL_CORE_DMA_SYSTEM_DREQ_44_45[23:16]	46	MCSPI2_DREQ_RX1	McSPI2 receive request channel 1
DMA_SYSTEM_DREQ_46	47	CTRL_CORE_DMA_SYSTEM_DREQ_46_47[7:0]	47	MMC2_DREQ_TX	MMC2 transmit request
DMA_SYSTEM_DREQ_47	48	CTRL_CORE_DMA_SYSTEM_DREQ_46_47[23:16]	48	MMC2_DREQ_RX	MMC2 receive request
DMA_SYSTEM_DREQ_48	49	CTRL_CORE_DMA_SYSTEM_DREQ_48_49[7:0]	49	UART1_DREQ_TX	UART1 transmit request
DMA_SYSTEM_DREQ_49	50	CTRL_CORE_DMA_SYSTEM_DREQ_48_49[23:16]	50	UART1_DREQ_RX	UART1 receive request
DMA_SYSTEM_DREQ_50	51	CTRL_CORE_DMA_SYSTEM_DREQ_50_51[7:0]	51	UART2_DREQ_TX	UART2 transmit request
DMA_SYSTEM_DREQ_51	52	CTRL_CORE_DMA_SYSTEM_DREQ_50_51[23:16]	52	UART2_DREQ_RX	UART2 receive request
DMA_SYSTEM_DREQ_52	53	CTRL_CORE_DMA_SYSTEM_DREQ_52_53[7:0]	53	UART3_DREQ_TX	UART3 transmit request
DMA_SYSTEM_DREQ_53	54	CTRL_CORE_DMA_SYSTEM_DREQ_52_53[23:16]	54	UART3_DREQ_RX	UART3 receive request
DMA_SYSTEM_DREQ_54	55	CTRL_CORE_DMA_SYSTEM_DREQ_54_55[7:0]	55	UART4_DREQ_TX	UART4 transmit request
DMA_SYSTEM_DREQ_55	56	CTRL_CORE_DMA_SYSTEM_DREQ_54_55[23:16]	56	UART4_DREQ_RX	UART4 receive request
DMA_SYSTEM_DREQ_56	57	CTRL_CORE_DMA_SYSTEM_DREQ_56_57[7:0]	57	MMC4_DREQ_TX	MMC4 transmit request
DMA_SYSTEM_DREQ_57	58	CTRL_CORE_DMA_SYSTEM_DREQ_56_57[23:16]	58	MMC4_DREQ_RX	MMC4 receive request
DMA_SYSTEM_DREQ_58	59	CTRL_CORE_DMA_SYSTEM_DREQ_58_59[7:0]	59	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_59	60	CTRL_CORE_DMA_SYSTEM_DREQ_58_59[23:16]	60	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_60	61	CTRL_CORE_DMA_SYSTEM_DREQ_60_61[7:0]	61	MMC1_DREQ_TX	MMC1 transmit request
DMA_SYSTEM_DREQ_61	62	CTRL_CORE_DMA_SYSTEM_DREQ_60_61[23:16]	62	MMC1_DREQ_RX	MMC1 receive request
DMA_SYSTEM_DREQ_62	63	CTRL_CORE_DMA_SYSTEM_DREQ_62_63[7:0]	63	UART5_DREQ_TX	UART5 transmit request
DMA_SYSTEM_DREQ_63	64	CTRL_CORE_DMA_SYSTEM_DREQ_62_63[23:16]	64	UART5_DREQ_RX	UART5 receive request
DMA_SYSTEM_DREQ_64	65	CTRL_CORE_DMA_SYSTEM_DREQ_64_65[7:0]	65	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_65	66	CTRL_CORE_DMA_SYSTEM_DREQ_64_65[23:16]	66	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_66	67	CTRL_CORE_DMA_SYSTEM_DREQ_66_67[7:0]	67	Reserved	Reserved by default but can be remapped to a valid DMA source

Table 16-5. DMA_SYSTEM Default Request Mapping (continued)

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_SYSTEM_DREQ_67	68	CTRL_CORE_DMA_SYSTEM_DREQ_66_67[23:16]	68	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_68	69	CTRL_CORE_DMA_SYSTEM_DREQ_68_69[7:0]	69	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_69	70	CTRL_CORE_DMA_SYSTEM_DREQ_68_69[23:16]	70	MCSPI4_DREQ_TX0	McSPI4 transmit request channel 0
DMA_SYSTEM_DREQ_70	71	CTRL_CORE_DMA_SYSTEM_DREQ_70_71[7:0]	71	MCSPI4_DREQ_RX0	McSPI4 receive request channel 0
DMA_SYSTEM_DREQ_71	72	CTRL_CORE_DMA_SYSTEM_DREQ_70_71[23:16]	72	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_72	73	CTRL_CORE_DMA_SYSTEM_DREQ_72_73[7:0]	73	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_73	74	CTRL_CORE_DMA_SYSTEM_DREQ_72_73[23:16]	74	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_74	75	CTRL_CORE_DMA_SYSTEM_DREQ_74_75[7:0]	75	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_75	76	CTRL_CORE_DMA_SYSTEM_DREQ_74_75[23:16]	76	DSS_DREQ	Display subsystem HDMI audio request
DMA_SYSTEM_DREQ_76	77	CTRL_CORE_DMA_SYSTEM_DREQ_76_77[7:0]	77	MMC3_DREQ_TX	MMC3 transmit request
DMA_SYSTEM_DREQ_77	78	CTRL_CORE_DMA_SYSTEM_DREQ_76_77[23:16]	78	MMC3_DREQ_RX	MMC3 receive request
DMA_SYSTEM_DREQ_78	79	CTRL_CORE_DMA_SYSTEM_DREQ_78_79[7:0]	79	UART6_DREQ_TX	UART6 transmit request
DMA_SYSTEM_DREQ_79	80	CTRL_CORE_DMA_SYSTEM_DREQ_78_79[23:16]	80	UART6_DREQ_RX	UART6 receive request
DMA_SYSTEM_DREQ_80	81	CTRL_CORE_DMA_SYSTEM_DREQ_80_81[7:0]	81	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_81	82	CTRL_CORE_DMA_SYSTEM_DREQ_80_81[23:16]	82	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_82	83	CTRL_CORE_DMA_SYSTEM_DREQ_82_83[7:0]	83	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_83	84	CTRL_CORE_DMA_SYSTEM_DREQ_82_83[23:16]	84	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_84	85	CTRL_CORE_DMA_SYSTEM_DREQ_84_85[7:0]	85	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_85	86	CTRL_CORE_DMA_SYSTEM_DREQ_84_85[23:16]	86	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_86	87	CTRL_CORE_DMA_SYSTEM_DREQ_86_87[7:0]	87	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_87	88	CTRL_CORE_DMA_SYSTEM_DREQ_86_87[23:16]	88	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_88	89	CTRL_CORE_DMA_SYSTEM_DREQ_88_89[7:0]	89	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_89	90	CTRL_CORE_DMA_SYSTEM_DREQ_88_89[23:16]	90	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_90	91	CTRL_CORE_DMA_SYSTEM_DREQ_90_91[7:0]	91	Reserved	Reserved by default but can be remapped to a valid DMA source

Table 16-5. DMA_SYSTEM Default Request Mapping (continued)

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_SYSTEM_DREQ_91	92	CTRL_CORE_DMA_SYSTEM_DREQ_90_91[23:16]	92	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_92	93	CTRL_CORE_DMA_SYSTEM_DREQ_92_93[7:0]	93	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_93	94	CTRL_CORE_DMA_SYSTEM_DREQ_92_93[23:16]	94	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_94	95	CTRL_CORE_DMA_SYSTEM_DREQ_94_95[7:0]	95	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_95	96	CTRL_CORE_DMA_SYSTEM_DREQ_94_95[23:16]	96	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_96	97	CTRL_CORE_DMA_SYSTEM_DREQ_96_97[7:0]	97	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_97	98	CTRL_CORE_DMA_SYSTEM_DREQ_96_97[23:16]	98	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_98	99	CTRL_CORE_DMA_SYSTEM_DREQ_98_99[7:0]	99	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_99	100	CTRL_CORE_DMA_SYSTEM_DREQ_98_99[23:16]	100	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_100	101	CTRL_CORE_DMA_SYSTEM_DREQ_100_101[7:0]	101	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_101	102	CTRL_CORE_DMA_SYSTEM_DREQ_100_101[23:16]	102	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_102	103	CTRL_CORE_DMA_SYSTEM_DREQ_102_103[7:0]	103	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_103	104	CTRL_CORE_DMA_SYSTEM_DREQ_102_103[23:16]	104	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_104	105	CTRL_CORE_DMA_SYSTEM_DREQ_104_105[7:0]	105	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_105	106	CTRL_CORE_DMA_SYSTEM_DREQ_104_105[23:16]	106	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_106	107	CTRL_CORE_DMA_SYSTEM_DREQ_106_107[7:0]	107	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_107	108	CTRL_CORE_DMA_SYSTEM_DREQ_106_107[23:16]	108	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_108	109	CTRL_CORE_DMA_SYSTEM_DREQ_108_109[7:0]	109	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_109	110	CTRL_CORE_DMA_SYSTEM_DREQ_108_109[23:16]	110	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_110	111	CTRL_CORE_DMA_SYSTEM_DREQ_110_111[7:0]	111	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_111	112	CTRL_CORE_DMA_SYSTEM_DREQ_110_111[23:16]	112	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_112	113	CTRL_CORE_DMA_SYSTEM_DREQ_112_113[7:0]	113	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_113	114	CTRL_CORE_DMA_SYSTEM_DREQ_112_113[23:16]	114	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_114	115	CTRL_CORE_DMA_SYSTEM_DREQ_114_115[7:0]	115	Reserved	Reserved by default but can be remapped to a valid DMA source

Table 16-5. DMA_SYSTEM Default Request Mapping (continued)

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_SYSTEM_DREQ_115	116	CTRL_CORE_DMA_SYSTEM_DREQ_114_115[23:16]	116	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_116	117	CTRL_CORE_DMA_SYSTEM_DREQ_116_117[7:0]	117	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_117	118	CTRL_CORE_DMA_SYSTEM_DREQ_116_117[23:16]	118	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_118	119	CTRL_CORE_DMA_SYSTEM_DREQ_118_119[7:0]	119	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_119	120	CTRL_CORE_DMA_SYSTEM_DREQ_118_119[23:16]	120	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_120	121	CTRL_CORE_DMA_SYSTEM_DREQ_120_121[7:0]	121	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_121	122	CTRL_CORE_DMA_SYSTEM_DREQ_120_121[23:16]	122	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_122	123	CTRL_CORE_DMA_SYSTEM_DREQ_122_123[7:0]	123	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_123	124	CTRL_CORE_DMA_SYSTEM_DREQ_122_123[23:16]	124	I2C4_DREQ_TX	I2C4 transmit request
DMA_SYSTEM_DREQ_124	125	CTRL_CORE_DMA_SYSTEM_DREQ_124_125[7:0]	125	I2C4_DREQ_RX	I2C4 receive request
DMA_SYSTEM_DREQ_125	126	CTRL_CORE_DMA_SYSTEM_DREQ_124_125[23:16]	126	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_SYSTEM_DREQ_126	127	CTRL_CORE_DMA_SYSTEM_DREQ_126_127[7:0]	127	Reserved	Reserved by default but can be remapped to a valid DMA source

16.1.3.2 Mapping of DMA Requests to DMA_CROSSBAR Inputs

Note

For information about the DMA_CROSSBAR module, refer to *DMA_CROSSBAR Module Functional Description* in *Control Module*.

Table 16-6 shows the mapping of device DMA requests to DMA_CROSSBAR inputs.

Table 16-6. Connection of The Device DREQs to The DMA_CROSSBAR Inputs

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_0	Reserved	Reserved
DMA_CROSSBAR_1	Reserved	Reserved
DMA_CROSSBAR_2	EXT_SYS_DREQ_0	External DMA request 0 (system expansion) - coming from SOC IOs. Level sensitive only
DMA_CROSSBAR_3	EXT_SYS_DREQ_1	External DMA request 1 (system expansion) - coming from SOC IOs - level sentitive only
DMA_CROSSBAR_4	GPMC_DREQ	GPMC request from prefetch engine
DMA_CROSSBAR_5	Reserved	Reserved
DMA_CROSSBAR_6	DISPC_DREQ	The line trigger signal to synchronize a memory to memory logical channel in the DMA4 (system DMA) is generated by the Display Controller IP.
DMA_CROSSBAR_7	CT_TBR_DREQ	DMA request coming from CT_TBR in DEBUGSS (used to be External DMA request 2 - coming from SOC IOs)
DMA_CROSSBAR_8 to DMA_CROSSBAR_14	Reserved	Reserved
DMA_CROSSBAR_15	MCSPI3_DREQ_TX0	McSPI module 3 - transmit request channel 0
DMA_CROSSBAR_16	MCSPI3_DREQ_RX0	McSPI module 3 - receive request channel 0
DMA_CROSSBAR_17 to DMA_CROSSBAR_22	Reserved	Reserved
DMA_CROSSBAR_23	MCSPI3_DREQ_TX1	McSPI module 3 - transmit request channel 1
DMA_CROSSBAR_24	MCSPI3_DREQ_RX1	McSPI module 3 - receive request channel 1
DMA_CROSSBAR_25	I2C3_DREQ_TX	I2C module 3 - transmit request
DMA_CROSSBAR_26	I2C3_DREQ_RX	I2C module 3 - receive request
DMA_CROSSBAR_27	I2C1_DREQ_TX	I2C module 1 - transmit request
DMA_CROSSBAR_28	I2C1_DREQ_RX	I2C module 1 - receive request
DMA_CROSSBAR_29	I2C2_DREQ_TX	I2C module 2 - transmit request
DMA_CROSSBAR_30	I2C2_DREQ_RX	I2C module 2 - receive request
DMA_CROSSBAR_31 to DMA_CROSSBAR_34	Reserved	Reserved
DMA_CROSSBAR_35	MCSPI1_DREQ_TX0	McSPI module 1 - transmit request channel 0
DMA_CROSSBAR_36	MCSPI1_DREQ_RX0	McSPI module 1 - receive request channel 0
DMA_CROSSBAR_37	MCSPI1_DREQ_TX1	McSPI module 1 - transmit request channel 1
DMA_CROSSBAR_38	MCSPI1_DREQ_RX1	McSPI module 1 - receive request channel 1
DMA_CROSSBAR_39	MCSPI1_DREQ_TX2	McSPI module 1 - transmit request channel 2
DMA_CROSSBAR_40	MCSPI1_DREQ_RX2	McSPI module 1 - receive request channel 2
DMA_CROSSBAR_41	MCSPI1_DREQ_TX3	McSPI module 1 - transmit request channel 3
DMA_CROSSBAR_42	MCSPI1_DREQ_RX3	McSPI module 1 - receive request channel 3
DMA_CROSSBAR_43	MCSPI2_DREQ_TX0	McSPI module 2 - transmit request channel 0
DMA_CROSSBAR_44	MCSPI2_DREQ_RX0	McSPI module 2 - receive request channel 0
DMA_CROSSBAR_45	MCSPI2_DREQ_TX1	McSPI module 2 - transmit request channel 1

Table 16-6. Connection of The Device DREQs to The DMA_CROSSBAR Inputs (continued)

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_46	MCSPI2_DREQ_RX1	McSPI module 2 - receive request channel 1
DMA_CROSSBAR_47	MMC2_DREQ_TX	MMC/SD2 transmit request
DMA_CROSSBAR_48	MMC2_DREQ_RX	MMC/SD2 receive request
DMA_CROSSBAR_49	UART1_DREQ_TX	UART module 1 - transmit request
DMA_CROSSBAR_50	UART1_DREQ_RX	UART module 1 - receive request
DMA_CROSSBAR_51	UART2_DREQ_TX	UART module 2 - transmit request
DMA_CROSSBAR_52	UART2_DREQ_RX	UART module 2 - receive request
DMA_CROSSBAR_53	UART3_DREQ_TX	UART module 3 - transmit request (Also infrared - IRDA)
DMA_CROSSBAR_54	UART3_DREQ_RX	UART module 3 - receive request (Also infrared - IRDA)
DMA_CROSSBAR_55	UART4_DREQ_TX	UART module 4 – transmit request
DMA_CROSSBAR_56	UART4_DREQ_RX	UART module 4 – receive request
DMA_CROSSBAR_57	MMC4_DREQ_TX	MMC/SD4 transmit request
DMA_CROSSBAR_58	MMC4_DREQ_RX	MMC/SD4 receive request
DMA_CROSSBAR_59	Reserved	Reserved
DMA_CROSSBAR_60	Reserved	Reserved
DMA_CROSSBAR_61	MMC1_DREQ_TX	MMC/SD1 transmit request
DMA_CROSSBAR_62	MMC1_DREQ_RX	MMC/SD1 receive request
DMA_CROSSBAR_63	UART5_DREQ_TX	UART module 5 – transmit request (used to be External DMA request 3 - coming from SOC IOs)
DMA_CROSSBAR_64	UART5_DREQ_RX	UART module 5 – receive request (used to be USIM receive request)
DMA_CROSSBAR_65 to DMA_CROSSBAR_69	Reserved	Reserved
DMA_CROSSBAR_70	MCSPI4_DREQ_TX0	McSPI module 4 - transmit request channel 0
DMA_CROSSBAR_71	MCSPI4_DREQ_RX0	McSPI module 4 - receive request channel 0
DMA_CROSSBAR_72 to DMA_CROSSBAR_75	Reserved	Reserved
DMA_CROSSBAR_76	DSS_DREQ	Display subsystem HDMI Audio DMA request
DMA_CROSSBAR_77	MMC3_DREQ_TX	MMC/SD3 transmit request
DMA_CROSSBAR_78	MMC3_DREQ_RX	MMC/SD3 receive request
DMA_CROSSBAR_79	UART6_DREQ_TX	UART module 6 – transmit request (used to be USIM transmit request)
DMA_CROSSBAR_80	UART6_DREQ_RX	UART module 6 – receive request (used to be USIM receive request)
DMA_CROSSBAR_81 to DMA_CROSSBAR_123	Reserved	Reserved
DMA_CROSSBAR_124	I2C4_DREQ_TX	I2C module 4 - transmit request
DMA_CROSSBAR_125	I2C4_DREQ_RX	I2C module 4 - receive request
DMA_CROSSBAR_126	Reserved	Reserved
DMA_CROSSBAR_127	Reserved	Reserved
DMA_CROSSBAR_128	McASP1_DREQ_RX	McASP receive event
DMA_CROSSBAR_129	McASP1_DREQ_TX	McASP transmit event
DMA_CROSSBAR_130	McASP2_DREQ_RX	McASP receive event
DMA_CROSSBAR_131	McASP2_DREQ_TX	McASP transmit event
DMA_CROSSBAR_132	McASP3_DREQ_RX	McASP receive event
DMA_CROSSBAR_133	McASP3_DREQ_TX	McASP transmit event
DMA_CROSSBAR_134	McASP4_DREQ_RX	McASP receive event

Table 16-6. Connection of The Device DREQs to The DMA_CROSSBAR Inputs (continued)

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_135	McASP4_DREQ_TX	McASP transmit event
DMA_CROSSBAR_136	McASP5_DREQ_RX	McASP receive event
DMA_CROSSBAR_137	McASP5_DREQ_TX	McASP transmit event
DMA_CROSSBAR_138	McASP6_DREQ_RX	McASP receive event
DMA_CROSSBAR_139	McASP6_DREQ_TX	McASP transmit event
DMA_CROSSBAR_140	McASP7_DREQ_RX	McASP receive event
DMA_CROSSBAR_141	McASP7_DREQ_TX	McASP transmit event
DMA_CROSSBAR_142	McASP8_DREQ_RX	McASP receive event
DMA_CROSSBAR_143	McASP8_DREQ_TX	McASP receive event
DMA_CROSSBAR_144	UART7_DREQ_TX	UART module 7 - transmit request
DMA_CROSSBAR_145	UART7_DREQ_RX	UART module 7 - receive request
DMA_CROSSBAR_146	UART8_DREQ_TX	UART module 8 - transmit request
DMA_CROSSBAR_147	UART8_DREQ_RX	UART module 8 - receive request
DMA_CROSSBAR_148	UART9_DREQ_TX	UART module 9 - transmit request
DMA_CROSSBAR_149	UART9_DREQ_RX	UART module 9 - receive request
DMA_CROSSBAR_150	UART10_DREQ_TX	UART module 10 - transmit request
DMA_CROSSBAR_151	UART10_DREQ_RX	UART module 10 - receive request
DMA_CROSSBAR_152	I2C5_DREQ_TX	I2C module 5 - transmit request
DMA_CROSSBAR_153	I2C5_DREQ_RX	I2C module 5 - receive request
DMA_CROSSBAR_154	VCP1_DREQ_RX	VCP RX Event
DMA_CROSSBAR_155	VCP1_DREQ_TX	VCP TX Event
DMA_CROSSBAR_156	VCP2_DREQ_RX	VCP RX Event
DMA_CROSSBAR_157	VCP2_DREQ_TX	VCP TX Event
DMA_CROSSBAR_158	DCAN1_DREQ_IF1	DCAN IF1 Event
DMA_CROSSBAR_159	DCAN1_DREQ_IF2	DCAN IF2 Event
DMA_CROSSBAR_160	DCAN1_DREQ_IF3	DCAN IF3 Event
DMA_CROSSBAR_161	DCAN2_DREQ_IF1	DCAN IF1 Event
DMA_CROSSBAR_162	DCAN2_DREQ_IF2	DCAN IF2 Event
DMA_CROSSBAR_163	DCAN2_DREQ_IF3	DCAN IF3 Event
DMA_CROSSBAR_164 to DMA_CROSSBAR_166	Reserved	Reserved
DMA_CROSSBAR_167	EXT_SYS_DREQ_2	External DMA request 2 (system expansion) - coming from SOC IOs. Level sensitive only
DMA_CROSSBAR_168	EXT_SYS_DREQ_3	External DMA request 3 (system expansion) - coming from SOC IOs. Level sentitive only
DMA_CROSSBAR_169	MCSPI2_DREQ_TX2	McSPI module 2 - transmit request channel 2
DMA_CROSSBAR_170	MCSPI2_DREQ_RX2	McSPI module 2 - receive request channel 2
DMA_CROSSBAR_171	MCSPI2_DREQ_TX3	McSPI module 2 - transmit request channel 3
DMA_CROSSBAR_172	MCSPI2_DREQ_RX3	McSPI module 2 - receive request channel 3
DMA_CROSSBAR_173	MCSPI3_DREQ_TX2	McSPI module 3 - transmit request channel 2
DMA_CROSSBAR_174	MCSPI3_DREQ_RX2	McSPI module 3 - receive request channel 2
DMA_CROSSBAR_175	MCSPI3_DREQ_TX3	McSPI module 3 - transmit request channel 3
DMA_CROSSBAR_176	MCSPI3_DREQ_RX3	McSPI module 3 - receive request channel 3
DMA_CROSSBAR_177	MCSPI4_DREQ_TX1	McSPI module 4 - transmit request channel 1
DMA_CROSSBAR_178	MCSPI4_DREQ_RX1	McSPI module 4 - receive request channel 1
DMA_CROSSBAR_179	MCSPI4_DREQ_TX2	McSPI module 4 - transmit request channel 2

Table 16-6. Connection of The Device DREQs to The DMA_CROSSBAR Inputs (continued)

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_180	MCSPI4_DREQ_RX2	McSPI module 4 - receive request channel 2
DMA_CROSSBAR_181	MCSPI4_DREQ_TX3	McSPI module 4 - transmit request channel 3
DMA_CROSSBAR_182	MCSPI4_DREQ_RX3	McSPI module 4 - receive request channel 3
DMA_CROSSBAR_183 to DMA_CROSSBAR_186	Reserved	Reserved
DMA_CROSSBAR_187	GPIO1_DREQ_EVT	GPIO module 1 - event/interrupt 1
DMA_CROSSBAR_188	GPIO2_DREQ_EVT	GPIO module 2 - event/interrupt 1
DMA_CROSSBAR_189	GPIO3_DREQ_EVT	GPIO module 3 - event/interrupt 1
DMA_CROSSBAR_190	GPIO4_DREQ_EVT	GPIO module 4 - event/interrupt 1
DMA_CROSSBAR_191	GPIO5_DREQ_EVT	GPIO module 5 - event/interrupt 1
DMA_CROSSBAR_192	GPIO6_DREQ_EVT	GPIO module 6 - event/interrupt 1
DMA_CROSSBAR_193	GPIO7_DREQ_EVT	GPIO module 7 - event/interrupt 1
DMA_CROSSBAR_194	GPIO8_DREQ_EVT	GPIO module 8 - event/interrupt 1
DMA_CROSSBAR_195	PWMSS1_DREQ_ePWM0_EVT	eHRPWM0 event/interrupt
DMA_CROSSBAR_196	PWMSS2_DREQ_ePWM1_EVT	eHRPWM1 event/interrupt
DMA_CROSSBAR_197	PWMSS3_DREQ_ePWM2_EVT	eHRPWM2 event/interrupt
DMA_CROSSBAR_198	PWMSS1_DREQ_eQEP0_EVT	eQEP0 event/interrupt
DMA_CROSSBAR_199	PWMSS2_DREQ_eQEP1_EVT	eQEP1 event/interrupt
DMA_CROSSBAR_200	PWMSS3_DREQ_eQEP2_EVT	eQEP2 event/interrupt
DMA_CROSSBAR_201	PWMSS1_DREQ_eCAP0_EVT	eCAP0 event/interrupt
DMA_CROSSBAR_202	PWMSS2_DREQ_eCAP1_EVT	eCAP1 event/interrupt
DMA_CROSSBAR_203	PWMSS3_DREQ_eCAP2_EVT	eCAP2 event/interrupt
DMA_CROSSBAR_204 to DMA_CROSSBAR_255	Reserved	Reserved

16.1.4 DMA_SYSTEM Functional Description

The DMA_SYSTEM module provides high-performance data transfers between memories and peripheral devices with low processor use. A DMA transfer is programmed through a logical DMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

Figure 16-5 shows the DMA_SYSTEM controller top-level block diagram.

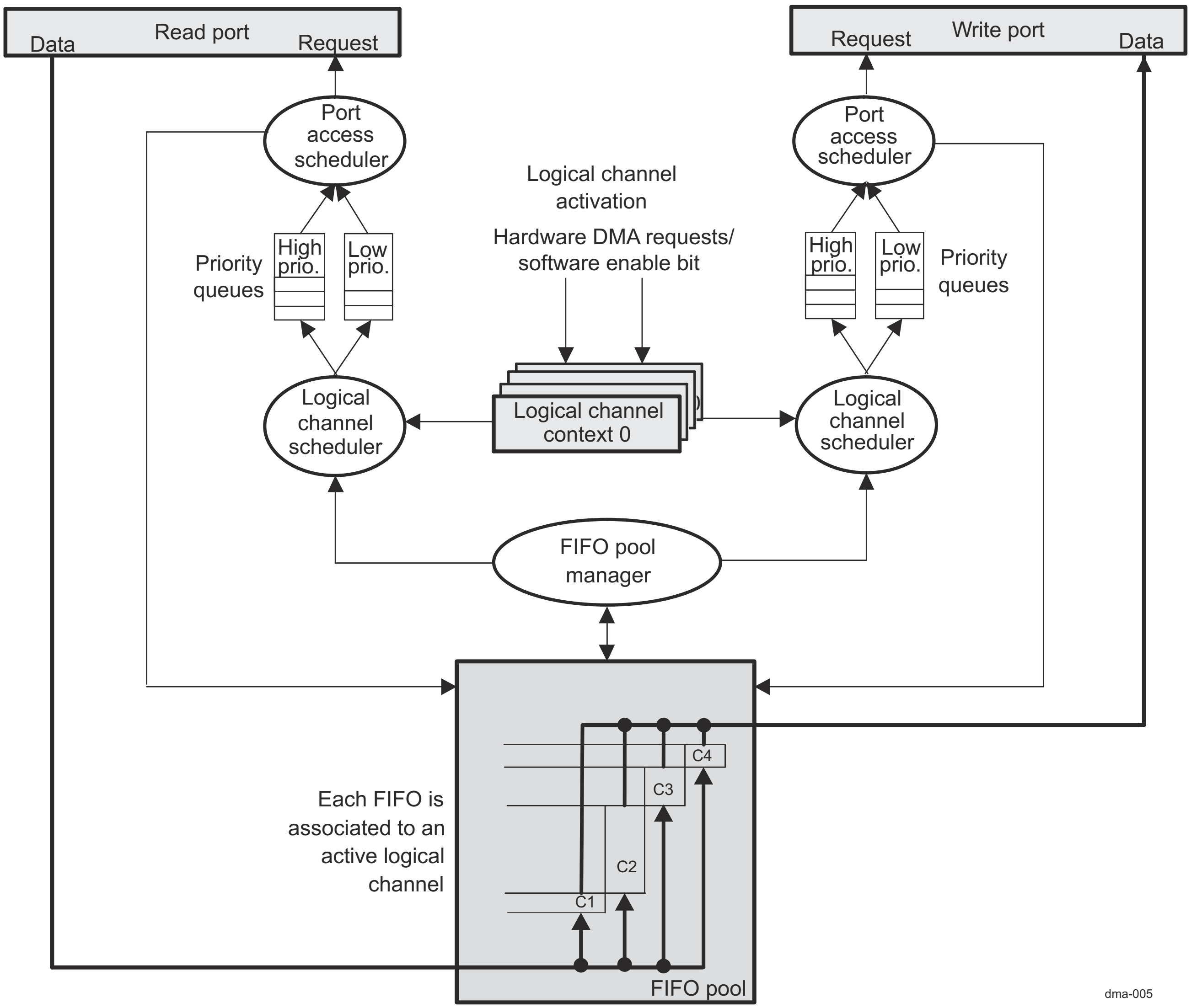


Figure 16-5. DMA_SYSTEM Controller Top-Level Block Diagram

16.1.4.1 DMA_SYSTEM Controller Power Management

Table 16-7 describes power-management features available for the DMA_SYSTEM controller.

Note

- For information about source clock gating and sleep/wake-up transitions, see secton *Clock Management*, in *Power, Reset, and Clock Management*.
- For a description of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see *Module Level Clock Management*, in *Power, Reset, and Clock Management*.

Table 16-7. Local Power-Management Features

Feature	Registers	Description
Clock auto gating	DMA4_OCP_SYSCONFIG[0] AUTOIDLE bit	This bit allows local power optimization inside the module by gating the SDMA_ICLK clock upon the interface activity.
Slave idle modes	DMA4_OCP_SYSCONFIG[4:3] SIDLEMODE bit field	Force-idle, no-idle, and smart-idle modes are available.
Clock activity	DMA4_OCP_SYSCONFIG[9:8] CLOCKACTIVITY bit field	For configuration details, see Table 16-8.
Master standby modes	DMA4_OCP_SYSCONFIG[13:12] MIDLEMODE bit field	Force-standby, no-standby, and smart-standby modes are available.
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	N/A	Feature not available

Table 16-8. Clock Activity Settings

SDMA_CLOCKACTIVITY Values	Clock State When Module is in IDLE State	
	SDMA_ICLK	SDMA_FCLK
00	Off	Off
10	Off	On
01	On	Off
11	On	On

CAUTION

Because the PRCM module cannot read CLOCKACTIVITY settings through hardware, software must ensure consistent programming between the SDMA_CLOCKACTIVITY and DMA_SYSTEM clock PRCM control bits. For a description of the ClockActivity feature, see *Module Level Clock Management*, in *Power, Reset, and Clock Management*.

16.1.4.2 DMA_SYSTEM Controller Interrupt Requests

DMA4 has four interrupt lines (Lj, where j = 0, 1, 2, 3). Each logical channel can request an interrupt over any line. The attachment of a channel interrupt event to one of these four external lines is programmable. Software determines whether it attaches a channel interrupt to a single IRQ line or to multiple IRQ lines.

There are two different registers per interrupt line:

- The DMA4_IRQSTATUS_Lj CH_31_0_Lj field shows the status of the different sources of interrupt. If the DMA4_IRQENABLE_Lj bit is 1, channel *i* is the source of interrupt in line *j*. In contrast to the DMA4_CSRi registers, the DMA4_IRQSTATUS_Lj registers are updated regardless of the corresponding bits in the DMA4_IRQENABLE_Lj registers.
- The DMA4_IRQENABLE_Lj CH_31_0_Lj_EN field masks/unmasks the channel interrupt. If the DMA4_IRQENABLE_Lj bit is set to 0, channel interrupt *i* of the line *j* is masked.

Each logical channel can generate a number of different interrupt events when enabled (that is, set to 1) in the DMA4_CICRi register. Each status bit is updated in the DMA4_CSRi register only when the corresponding enable bit is enabled in the DMA4_CICRi register.

To determine an interrupt source when an interrupt rises on an interrupt line L*j*:

- Identify the channel (LCH*i*) generating the interrupt.

  Read DMA4_IRQSTATUS_Lj.LCH*i* (LCH0 to LCH31). If LCH*i* = 1, channel *i* is the originator of the interrupt.
- Identify the interrupt event.

  Read the LCH*i* DMA4_CSRi. For example, if the drop event (the DMA4_CSRi[1] DROP bit) is 1, a request collision will occur.

The interrupt event status bit in the DMA4_CSRi register is immediately reset after it is written to 1.

The interrupt status bit in the DMA4_IRQSTATUS_Lj register is cleared after it is written to 1.

16.1.4.2.1 Interrupt Generation

The DMA_SYSTEM module has four interrupt request output lines, DMA_SYSTEM_IRQ_0 to DMA_SYSTEM_IRQ_3. One or more logical channels can be programmed to generate an interrupt request on any of these lines when any one of the maskable DMA events listed in Table 16-9 occurs.

Table 16-9. Logical DMA Channel Events

Event	Description
End of packet	A packet transfer completed.
End of block	A block transfer completed.
End of frame	A frame transfer completed.
End of super block	A super block transfer completed.
Half of frame	Half of the current frame transferred.
Start of last frame	The first element of the last frame transferred.
Transaction error	A transaction error is returned by the interconnect in either the read or write port.
Address error	An attempt was made to perform a DMA access to an address not aligned on an ES boundary. Condition to occur: if DMA4_CENi[23:0] CHANNEL_ELMNT_NBR = 0x000000 or DMA4_CFNi[15:0] CHANNEL_FRAME_NBR = 0x0000 or DMA4_CSDPi[1:0] DATA_TYPE = 0x3.
Supervisor transaction error	An error occurred, for example, when an unauthorized initiator (that is not a supervisor) tries to use a supervisor transfer.
Drain end	Drain is completed (DMA4_CCRi[10] WR_ACTIVE becomes 0).
Drop error	A drop event interrupt is generated when a DMA request is being serviced while a second one is asserted and a third one arrives before the second DMA request is serviced.

The logical DMA channels that generate an interrupt on a particular IRQ output are specified through the DMA4_IRQENABLE_Lj register (where *j* is the IRQ number: 0, 1, 2, or 3). The events that generate an interrupt for a particular channel can be configured through the channel DMA4_CICRi register.

When an interrupt is detected, the logical DMA channel generating the event can first be identified by reading the DMA4_IRQSTATUS_Lj register. The event causing the interrupt then can be identified by reading the interrupt status via the relevant DMA channel DMA4_CSRi register.

16.1.4.3 Logical Channel Transfer Overview

As Figure 16-5 shows, the DMA_SYSTEM module has one read port and one write port operating independently of one another. Buffering is provided between the read and write ports through a FIFO queue memory pool that is shared dynamically between the active logical channels.

- Logical channel synchronization

  A logical channel is described as hardware-synchronized when the DMA transfers are triggered by DMA requests from a hardware device. Alternatively, a logical channel is described as nonsynchronized when the DMA transfer is triggered by software.
- Logical channel activation

  A logical channel becomes active as follows:
  – For hardware-synchronized transfers, when the logical channel is enabled and the hardware DMA request line is asserted
  – For software-triggered (nonsynchronized) transfers, as soon as software enables the logical channel
- Logical channel transfer composition

  A DMA transfer is divided automatically into a number of transactions. Depending on the logical channel context configured, the transfer size, start address alignment, addressing mode, and configured maximum burst size, each transaction can be a single access or a burst of accesses.
- Logical channel scheduling

When several logical channels are active at the same time, schedulers manage the read and write ports. The scheduling of logical channel transfers is similar for both read and write ports. When a logical channel becomes active, it is added to the tail of a scheduling queue. If more than one logical channel becomes active at the same time, the one with the lower number is queued first. This mechanism provides a first-come, first-serve scheduling scheme between the concurrently active logical channels.

In addition, each read and write port has a high-priority queue and a low-priority queue. The priority bits (WRITE_PRIORITY and READ_ PRIORITY) in the logical channel DMA4_CCRi register determine whether a logical channel is queued as high or low priority. A software-configurable 8-bit priority counter gives weighting to the priority write queue. For every N (1 to 255) schedules from the priority write queue, one is scheduled from the regular write queue. A channel that is scheduled goes to the end of the queue after it completes its turn on the port. The relative weighting of the scheduling of the high-priority queue to the low priority queue is programmable from 1:1 to 1:256 through the DMA global channel register using the DMA4_GCR[23:16] ARBITRATION_RATE bit field.

Note

The DMA4_GCR[23:16] ARBITRATION_RATE bit field does not depend on the DMA4_GCR[13:12] HI_THREAD_RESERVED bit field. The ARBITRATION_RATE bit field depends on the DMA4_CCRi[26] WRITE_PRIORITY bit and the DMA4_CCRi[6] READ_PRIORITY bit.

- Read/write port access scheduling policy

  When either the read or write port becomes available, the port access scheduler selects the next logical channel for which to perform a DMA transaction from either the high- or low-priority queue.

  When the current DMA transaction (single or burst access) is complete and the full DMA transfer is not finished, the logical channel returns to the tail of the queue. Because the port access scheduling is on a per-transaction basis, a logical channel can be queued repeatedly this way several times during its block transfer.

The DMA_SYSTEM module can have up to four outstanding read transactions and two outstanding write transactions in the system interconnect; four read and two write thread IDs exist. For an arbitration cycle to occur, these two conditions must be met:

- At least one channel is requesting
- At least one free thread ID is available

On an arbitration cycle, the scheduler grants the highest priority channel that has an active request, allocates the thread ID, and tags this thread as busy. At a given time, a channel cannot be allocated for more than one thread ID.

Note

If more than one channel is active, each channel is given a thread ID for the current service only, not for the whole channel transfer.

When only one channel is enabled, only one thread is allocated for the channel. In such a situation the channel can have maximum of four outstanding commands (without getting the responses) without rescheduling the channel at the end of each transaction. Each command can be either single access (8-bit, 16-bit or 32-bit) or burst access (2 × M, 4 × M, 8 × M or 16 × M, where M can be 8, 16, or 32 bits).

When nonburst alignment is at the beginning of the transfer, the channel is rescheduled for each smaller access until burst-aligned. When the end of the transfer is not burst-aligned, the channel is rescheduled for each of the remaining smaller accesses.

For a logical channel transfer completion, when the last access is written to the destination, the logical channel becomes inactive. If enabled, an interrupt request is generated (see Section 16.1.4.2.1, *Interrupt Generation*).

16.1.4.4 FIFO Queue Memory Pool

A FIFO queue memory pool provides buffering between the read and write ports. The hardware allocates the space dynamically to a number of FIFO queues, and each queue is associated with an active logical channel.

To avoid a memory pool overflow, if there are fewer entries in the FIFO queue memory pool than are required for the maximum configured source burst size of the next logical channel to be scheduled, the logical channel is returned to the tail of the queue, and the port access scheduler continues to search the queue until it finds a logical channel that can be scheduled.

The maximum FIFO depth that can be allocated to each individual logical channel can be limited globally through the DMA4_GCR[7:0] MAX_CHANNEL_FIFO_DEPTH bit field. This value should be configured to allow a fair allocation of the memory pool between the active channels.

A logical channel is scheduled if it has not yet reached its allocation limit, even if the access to be performed will exceed this limit. This means that the effective number of entries used by a particular logical channel is limited to the configured maximum entries per channel + channel maximum configured burst size (in words) 1.

16.1.4.5 Addressing Modes

A DMA transfer block consists of a number of frames (FN). Each frame consists of a number of elements (EN), and each element can have a size of 8, 16, or 32 bits (ES), as follows:

transfer block size = FN × EN × ES

The FN, EN, and ES are common for the source and destination. However, the way in which the data is represented (addressing profile/mode) is independently programmable for the source and destination devices, using one of these four addressing modes:

- Constant: The address remains the same for consecutive element accesses.
- Post-increment: The address increases by the ES, even across consecutive frames.
- Single-index: The address increases by the ES plus the element index (EI) value minus 1 (even across consecutive frames).
- Double-index: The address increases by the ES plus the EI value minus 1 within a frame. When a full frame is transferred, the address increases by the ES plus the frame index (FI) value minus 1.

The ES, EI, and FI values are expressed in bytes. The EI and FI values can be positive or negative.

When calculating the EI and FI values, it is critical to note that, after an element is accessed, the logical channel address pointer equals the address of the last byte (highest address) of the accessed element. The correct value for the EI or FI must be such that, when added to the logical channel address pointer, it results in the address of the first byte (lowest address) of the next element to be accessed.

The EI and FI values must be configured so that the address of each element in the transfer is aligned on an ES boundary.

Consequently, the single-index addressing mode with EI = 1 or double-index addressing mode with EI = 1 and FI = 1 is equivalent to post-increment addressing.

Note

The source and destination start addresses must also be aligned on an ES boundary.

When the address of an element to be accessed is not aligned on an ES boundary, the transfer is stopped and a misaligned address error interrupt occurs, if enabled (see Section 16.1.4.2.1, *Interrupt Generation*).

The DMA4_CFNi register configures the FN in a block.

The DMA4_CENi register configures the EN.

The DMA4_CSDPi register configures the ES.

The DMA4_CSSAi and DMA4_CDSAi registers configure the source and destination start addresses.

The DMA4_CCRi register configures the source and destination addressing modes.

The DMA4_CSEIi, DMA4_CSFIi, DMA4_CDEIi, and DMA4_CDFIi registers configure the source EI, source FI, destination EI, and destination FI, respectively.

The addressing profiles are expressed as equations as follows:

Equation 1. Constant addressing:

$$A(n + 1) = A(n)$$

Note

Constant addressing mode with DMA4 to/from DDR memory is not supported on the device. To fill the DDR memory with a single value, the constant fill feature of the DMA4 must be used, instead of a constant-addressing mode transfer.

Equation 2. Post-increment addressing:

$$A(n + 1) = A(n) + ES$$

Equation 3. Single-indexed addressing:

$$A(n + 1) = A(n) + ES + (EI\ 1)$$

Equation 4. Double-indexed addressing:

When not at the end of a frame or transfer (that is, when the element counter ≠ 0):

$$A(n + 1) = A(n) + ES + (EI\ 1)$$

When at the end of a frame but not at the end of the transfer (that is, when the element counter = 0 and the frame counter ≠ 0):

$$A(n + 1) = A(n) + ES + (FI\ 1)$$

Calculate the element and frame index as follows:

Equation 5. Element index

$$EI = [(Stride\ EI\ 1) * ES] + 1$$

Equation 6. Frame index

$$FI = [(Stride\ FI\ 1) * ES] + 1$$

where:

A(n): Byte address of the element *n* within the transfer.

ES is in bytes, ES{1, 2, 4}.

EI is in bytes, specified in a configuration register, 32768 EI 32767.

Stride EI: The difference in the number of elements between the start of the current element *n* to the start of next element, *n+1*.

Element counter: A counter that is (re)initiated with the number of elements per frame or per transfer. Decreased by 1 for each element transferred. The initial value is configured in the register DMA channel element number, DMA4_CENi.

FI is in bytes, specified in a configuration register, 2147483648 FI 2147483647.

Stride FI: The difference in the number of elements between the start of the last element of the current frame and the beginning of the first element of the next frame.

Frame counter: A counter that is (re)initiated with the FN per transfer. Decreased by 1 for each frame transferred. The initial value is configured in the register DMA channel frame number, DMA4_CFNi.

Figure 16-6 shows how a stride EI and FI are defined. When handling complex configurations, using strides can make it easier to calculate EI and FI because you can calculate in elements instead of bytes. (This approach is used in the 90-degree clockwise image rotation example shown in Figure 16-10.) The double-index addressing example shown in Figure 16-6 uses ES = 4, EN = 2, EI = 5, FI = 5, and FN = 2.

Figure 16-6 through Figure 16-9 show examples of addressing mode configurations. Table 16-10 lists parameter values for the examples.

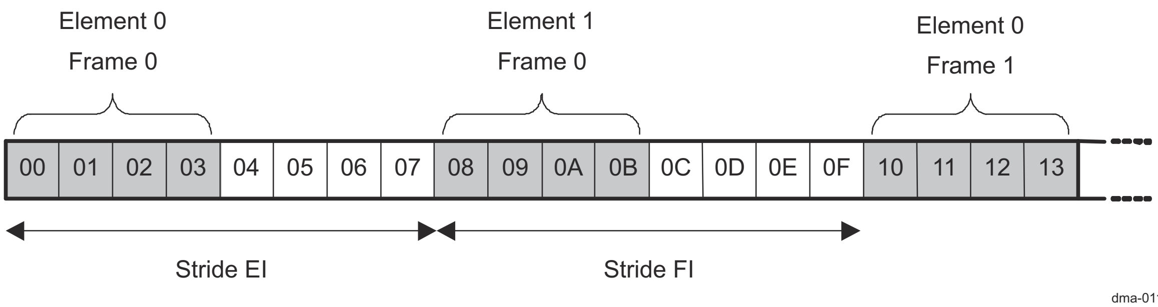


Figure 16-6. Example Showing Double-Index Addressing, Elements, Frames, and Strides

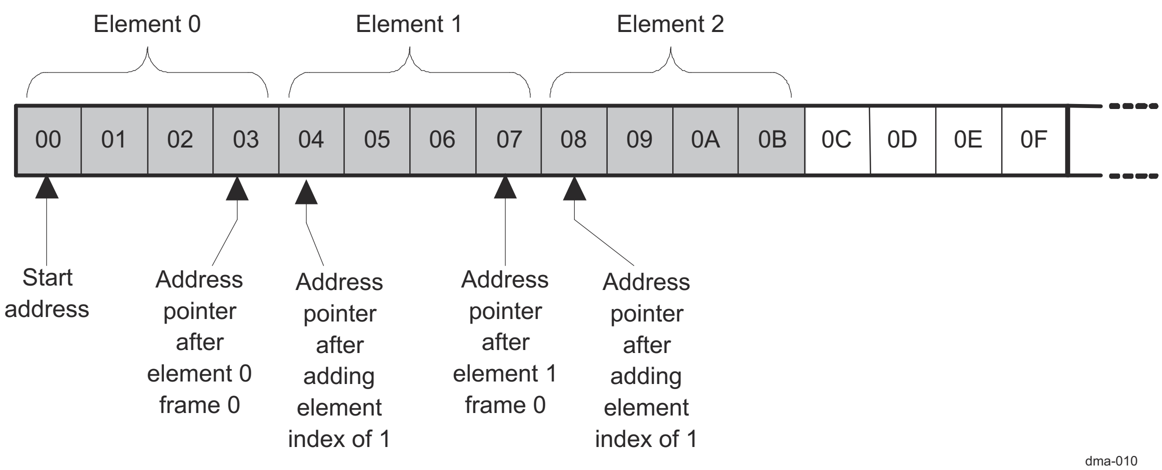


Figure 16-7. Addressing Mode Example (a)

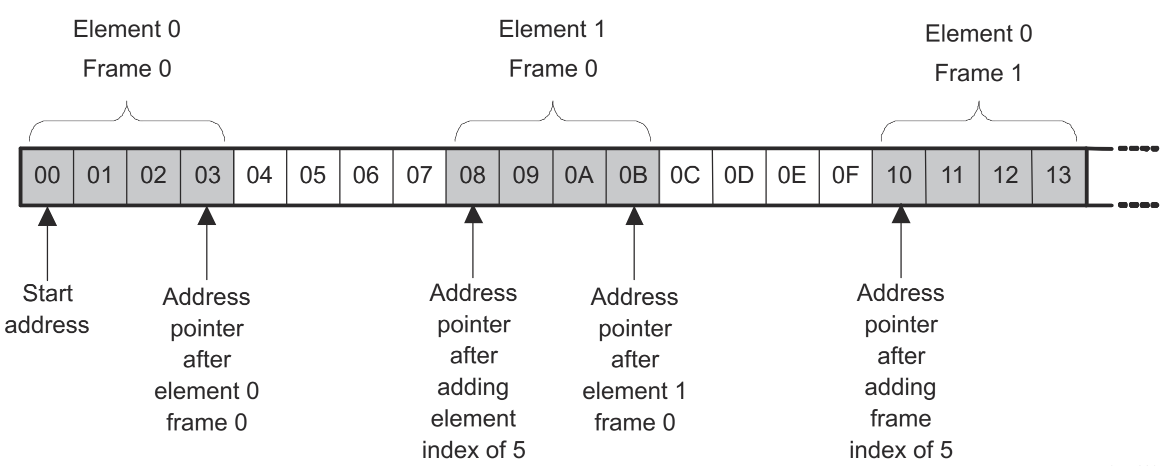


Figure 16-8. Addressing Mode Example (b)

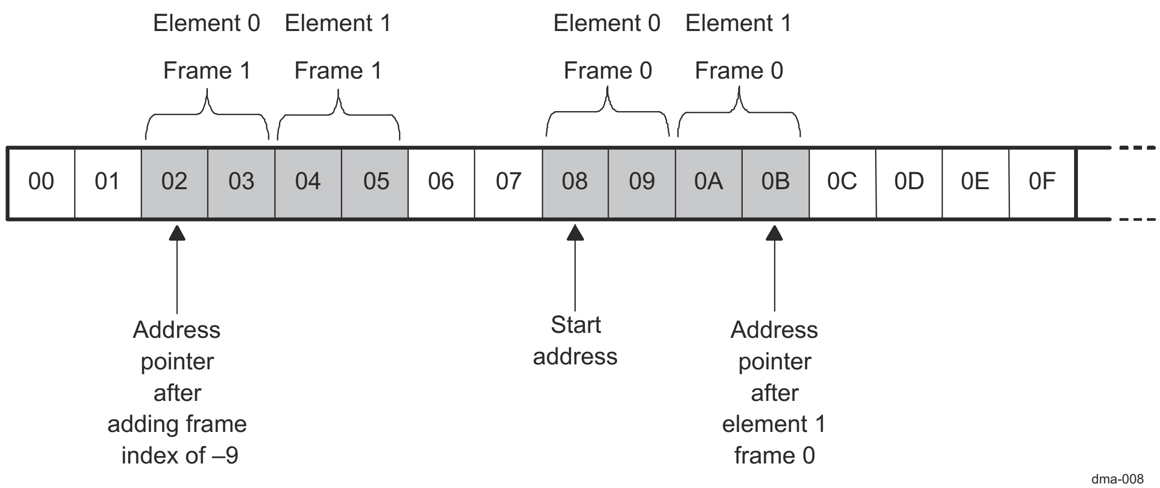


Figure 16-9. Addressing Mode Example (c)

Table 16-10. Parameter Values for Addressing Mode Examples (a), (b), and (c)

Parameter	Example (a)	Example (b)	Example (c)
Addressing mode	Single index (or post-increment)	Double index	Double index
Start address	0	0	8
ES	4 (32-bit)	4 (32-bit)	2 (16-bit)
EN	3	2	2
EI	1	5	1
FN	1	2	2
Frame index	N/A	5	-9

Double indexing can occur on source (read) or destination (write). Equations for rotation of xx degrees on destination are obtained by taking equations for rotation of (360 – xx) degrees on source, and swapping the width (x) and height (y) of the image in them. The opposite is also true. Table 16-11 lists the equations for 90-, 180-, and 270-degree rotations.

Table 16-11. Equations for Rotation

		90° Rotation	180° Rotation	270° Rotation
Double indexing on destination (write)	Base address	ES*(y–1)	ES*(x*y–1)	ES*y*(x–1)
	EI	ES*(y–1) + 1	1–2*ES	1–ES*(y + 1)
	FI	1 ES*[(x–1)*y + 2]	1–2*ES	1+ES*(x–1)*y
Double indexing on source (read)	Base address	ES*x*(y–1)	ES*(x*y–1)	ES*(x–1)
	EI	1–ES*(x + 1)	1–2*ES	ES*(x–1) + 1
	FI	1+ES*(y–1)*x	1–2*ES	1 ES*[(y–1)*x + 2]

Table 16-12 and Figure 16-10 show the configuration required to perform a 90-degree clockwise rotation of a 240 × 160 pixel, 32-bit image. The EI, frame size, and FI values are configured so that the image is rotated line-by-line starting at the left end of the top line.

Note

The FI value for the destination is negative so that the first pixel of each subsequent line of the source image is written to the correct location at the destination.

Equation 5 and Equation 6 calculate the destination, FI and EI. The example assumes that the image lines are stored at consecutive addresses in memory, meaning that both EI and FI on the source side are 1.

Rotations:

Section 16.1.5.7, *90-Degree Clockwise Image Rotation*, describes how to program an example of a 90-degree clockwise image rotation.

Observe that:

- One pixel = one element
- One line = one DMA frame
- Pixel size = element size = ES

Table 16-12. Example Parameter Values for a 90-Degree Clockwise Image Rotation

Parameter	Source Value	Destination Value
Bits per pixel	32	32
ES	4	4
Image width	SW	SH
Image height	SH	SW
Stride elements (stride EI)	1 element	SH
Stride frames (stride FI)	1 element	–[(SW-1)*SH + 1] = 38,241 elements
Start address	0x100000	0x200000 + (SH 1) x ES = 0x20027C
EN	SW	SW
EI	[(Stride EI 1) * ES] + 1 = 1	[(Stride EI 1) * ES] + 1 = 637
FN	SH	SH
FI	[(Stride FI 1) * ES] + 1= 1	[(Stride FI 1) * ES] + 1 = 152,967

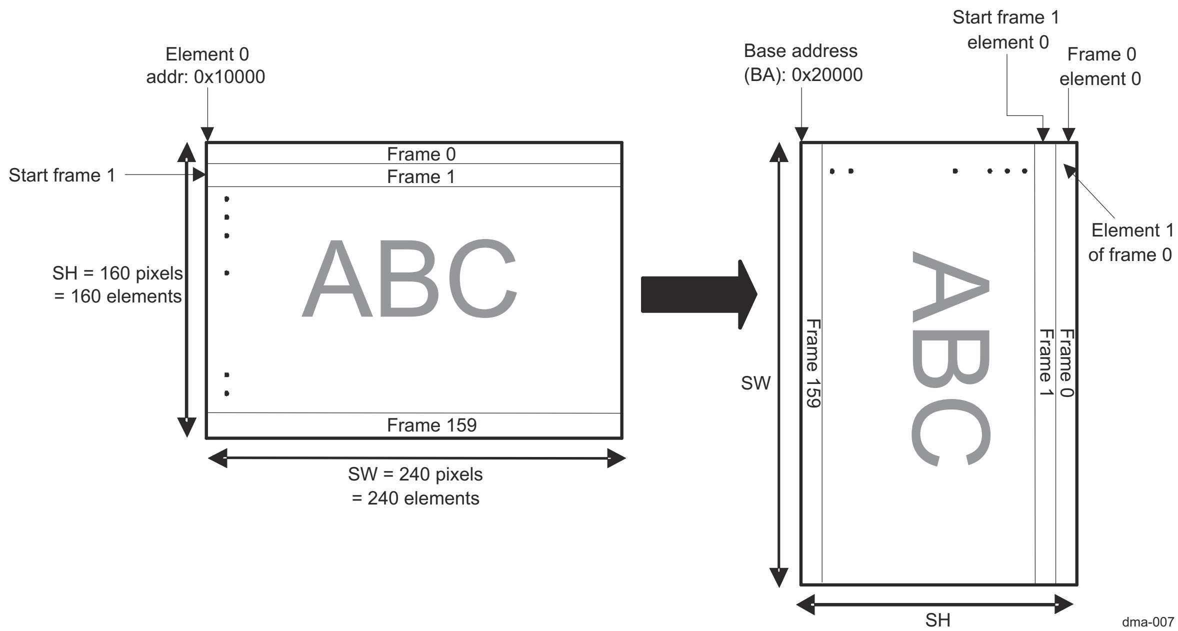


Figure 16-10. Example of a 90-Degree Clockwise Image Rotation

16.1.4.6 Packed Accesses

To pack data means to group data to match the bus size, thus optimizing a transfer. When the logical channel ES is less than the DMA module read/write port size, and the addressing profile supports it (post-increment mode or single- or double-index mode with EI = 1), the number of elements to transfer in each read/write port access can be maximized by specifying that the source or destination is packed through the channel DMA4_CSDPi register. Thus:

- For a read/write port size of 32 bits, the source or destination can be configured as packed for transfer ESs of 8 bits (four elements per access) and 16 bits (two elements per access).
- For a read/write port size of 64 bits, the source or destination can be configured as packed for transfer ESs of 8 bits (eight elements per access), 16 bits (four elements per access), and 32 bits (two elements per access).

Depending on the start address and transfer length, the first or last packed access can be only partially filled. This is indicated to the source or destination using the byte-enable signals.

16.1.4.7 Burst Transactions

Transfer performance can be improved so that the source or destination and addressing profile supports it. This can be achieved by configuring the logical channel to perform burst transactions consisting of multiple instead of single accesses. The channel can be programmed to use burst sizes equal to 16, 32, or 64 bytes through the DMA4_CSDPi register, with the read burst size programmable independently of the write burst size. Typically, the optimal burst size is 64 bytes (16 accesses for a 32-bit read/write port size or 8 accesses for a 64-bit read/write port size).

To obtain the maximum benefit from burst transactions, the source and destination start addresses must be aligned with the burst size. If this is not the case, the start of the transfer can consist of a number of smaller (single or burst) transactions until the first burst size boundary is reached.

Similarly, if the end of the transfer is not aligned on a burst size boundary, the final part of the transfer can consist of a number of smaller transactions.

Note

If post-incrementing is used, data must be packed to DMA data-port width, to use burst.

16.1.4.8 Endianism Conversion

The source and destination are each specified as little-endian or big-endian through the DMA4_CSDPi register for the particular logical channel. If the endianism of the source and destination differ, and if the logical channel ES is less than the DMA_SYSTEM module read/write port size, an endianism conversion is applied to the data before it is written to the destination.

When transferring data between a source and a destination with different endianism, it is important to specify an ES that equals the type of data being transferred to preserve the correct data image at the destination.

In the system, endianism conversion can be performed in more than one place. It is possible to instruct the source and/or destination to lock the endianism (that is, to not perform a conversion) through the logical DMA channel DMA4_CSDPi register.

Note

Because the device is little-endian by construction, the DMA_SYSTEM endianism registers must never be set to big-endian.

If DMA_SYSTEM is used to execute endian conversion by setting the source and destination to different endianism values, it is important to consider that the L3_MAIN interconnect also executes endian conversion if the DMA_SYSTEM and the source or destination have a different data bus width.

16.1.4.9 Transfer Synchronization

A logical channel can be programmed for software-triggered or hardware synchronized transfers.

16.1.4.9.1 Software Synchronization

A transfer is software-triggered when the logical channel is set up and started by software. To specify a software-triggered transfer, set the channel DMA DMA4_CCRi[4:0] and DMA4_CCRi[20:19] bit fields to 0. The transfer starts as soon as the DMA DMA4_CCRi[7] bit is set (when it enters the scheduling process).

16.1.4.9.2 Hardware Synchronization

A transfer is hardware-synchronized if the logical channel activation is driven by hardware requests from the source or destination target. A hardware-synchronized transfer is specified by configuring the DMA request line number in the channel DMA4_CCRi register to a value that corresponds to the DMA request line from the source or destination that generates the DMA requests. The DMA request numbers to be configured are specified in the DMA request mapping (see Table 16-9).

Specify the DMA request number in the DMA4_CCRi[4:0] SYNCHRO_CONTROL and DMA4_CCRi[20:19] SYNCHRO_CONTROL_UPPER bit fields. After the DMA4_CCRi[7] ENABLE bit is set, the logical channel becomes enabled but not activated (it does not enter the scheduling process), which means that channel registers are not updated until the first DMA request is received.

Note

The channel synchronization control registers are 1-based. For example, to enable the S_DMA_1 request, the DMA4_CCRi[4:0] SYNCHRO_CONTROL bit field must be set to 0x2 (DMA request number + 1).

Note

A DMA request line must not be shared between concurrently enabled DMA channels. However, a DMA request line can be shared among several chained logical channels.

For hardware synchronization, the amount of data to be transferred for each assertion of the DMA request line is configured through the frame synchronization (FS) and block synchronization (BS) bits in the logical channel DMA4_CCRi register and the DMA4_CCRi[5] FS and DMA4_CCRi[18] BS bits, respectively.

The amount of data can be any of the following:

- A single element transfer: A complete element defined by data type. For example, 8/16/32 bits are transferred in response to a DMA request.
- A full frame: A complete frame of several elements is transferred in response to a DMA request.
- A full block (a full channel transfer): A complete block of several frames is transferred in response to a DMA request.
- A full packet (a full channel transfer): A complete packet of several elements is transferred in response to a DMA request.

Packets allow the size of each part of the full DMA transfer to be configured independently of the organization of the data to be transferred (typically a number of elements). This can be useful when the source or destination has a buffer (such as a FIFO queue) with a size unrelated to the frame size of the transfer. The packet size then can be set to the size of the buffer.

Packet transfer must be used only where the source or destination is addressed in constant addressing mode, because FI registers are reused to specify the packet size.

To support the burst mode, the logical channel must also be configured in target-port packed access mode.

The packet size is configured based on the DMA4_CCRi[24] SEL_SRC_DST_SYNC bit through either the channel DMA4_CDFIi register (source synchronized) or the DMA4_CSFIi register (destination synchronized).

When the logical channel transfer block is not an exact multiple of the packet size, the final packet consists of the remaining elements in the transfer, using burst or single accesses to complete the block transfer.

The maximum transfer size, regardless of the packet size, is always as follows:

Block_size = Number_of_Frame_in_Block * Number_of_Element_in_Frame * Element_Size

- Synchronized at the source

The DMA module optimizes the transfer with respect to the number and size of burst transactions for the given source and destination addressing profiles and configured maximum burst sizes. When writing to the destination is slower than reading from the source, data is buffered in the channel FIFO queue. If the transfer is packet-synchronized at the source, the end-of-packet interrupt is disabled (see Section 16.1.4.13, *Reprogramming an Active Channel*).

For a source synchronized transfer, buffering can be enabled or disabled by setting the DMA4_CCRi[25] BUFFERING_DISABLE bit. For a packet source synchronization with buffering disabled and the packed/burst across the packet boundary, the last packed/burst write transaction is split in optimized smaller accesses to complete the packet transfer size. However, for a packet source synchronized transfer with buffering enabled and with the packed/burst across the packet boundary, the DMA module waits for the next DMA request(s) to read enough data to issue an atomic packed/burst write transaction (assuming that the address is packed/burst aligned).

Note

Buffering is not performed between frames, even if it is enabled. If the packed/burst is across the frame boundary, the last packed/burst write transaction is split in optimized smaller accesses to complete the frame transfer size.

- Synchronized at the destination

The performance of a hardware-synchronized transfer can be improved by using the prefetch mode, enabled through the channel DMA4_CCRi[23] PREFETCH bit. Data is prefetched on the read port side before the DMA request received and buffered in the FIFO queue. Up to a full transfer block can be prefetched, although this can be limited by the specified maximum channel FIFO queue depth (see Section 16.1.4.4, *FIFO Queue Memory Pool*).

Buffering disable is not allowed for a destination-synchronized transfer.

Note

Behavior is undefined when prefetch is enabled and a transfer is synchronized to the source.

Regardless of whether buffering is enabled, the last transaction in the frame or in the block is write nonposted (WNP) even if the write mode is specified as write last nonposted (WLNP; the DMA4_CSDPi[17:16] WRITE_MODE bit field = 0x2). However, in a packet synchronization mode, the last transaction of each packet in the transfer is WNP only if the buffering disable is on (even if the write mode is specified as WLNP).

Regardless of whether buffering is enabled, the packet interrupt is not generated in the packet source synchronized mode.

CAUTION

The DMA4_CCRi[25] BUFFERING_DISABLE bit must be filled with an allowed value, as specified in Table 16-13.

Table 16-13. Buffering Disable

	BUFFERING_DISABLE	
	(0: Buffering enable, 1: Buffering disable)	
Destination synchronized	0	Allowed
	1	Not allowed
Source synchronized	0	Allowed
	1	Allowed

- Synchronized transfer monitoring using CDAC (DMA4_CDACi):

Context is restored only when the channel becomes active on a DMA request (not at software enable). The channel is software-enabled first, and then a DMA request is asserted followed by the first context restore.

The CDAC register is writable; thus, the CDAC can be initialized to monitor the transfer and determine whether the transfer is started (for more information, see Section 16.1.5.4, *Synchronized Transfer Monitoring Using CDAC*).

Note

For 16-bit transactions, start reading from or writing to the LSByte first to enable the register update. This is not an issue for 32-bit read-write transactions.

16.1.4.10 Thread Budget Allocation

When several concurrent channels are latency critical and hardware synchronized, a specific latency cannot be ensured until the target is served. This situation occurs when the number of concurrent channels is greater than the number of available threads.

Note

Four threads are available on the read port, and two threads are available on the write port.

For a hardware-synchronized transfer (memory to peripheral), a minimum bandwidth for a latency-critical transfer must be ensured to avoid collisions between two hardware requests.

Because it is latency critical, the software user is responsible for the following:

- Programming the synchronized channel as a high-priority channel
- Reserving one or several threads for high-priority channels

The proposed implementation is as follows (see Section 16.1.5.5, *Concurrent Software and Hardware Synchronization*):

Prevent the regular channel queue from exceeding more than a programmable (3, 2, or 1) number of threads on the read port and no more than one thread on the write port. This number can be set in the global register DMA4_GCR[13:12] .

The thread reservation is programmable for maximum use of thread resources for concurrent, low-priority channel transfer. Programmability can also allow a partial throughput control by limiting in software the number of concurrent outstanding requests that break the pipelining.

Depending on the DMA4_GCR [13:12] value, the following threadID on the read/write ports are allocated for a high-priority channel:

Read port priority thread reservation:

- DMA4_GCR[13:12] = 0x0 => No ThreadID is reserved for high-priority channels.
- DMA4_GCR[13:12] = 0x1 => Read ThreadID 0 is reserved for high-priority channels.
- DMA4_GCR[13:12] = 0x2 => Read ThreadID 0 and Read ThreadID 1 are reserved for high-priority channels.
- DMA4_GCR[13:12] = 0x3 => Read ThreadID 0, Read ThreadID 1, and Read ThreadID 2 are reserved for high-priority channels.

Write port priority thread reservation:

- DMA4_GCR[13:12] = 0x0 => No ThreadID is reserved for high-priority channels
- DMA4_GCR[13:12] = 0x1 => Write ThreadID 0 is reserved for high-priority channels.
- DMA4_GCR[13:12] = 0x2 => Write ThreadID 0 is reserved for high-priority channels.
- DMA4_GCR[13:12] = 0x3 => Write ThreadID 0 is reserved for high-priority channels.

Regardless of whether the enabled channels are high priority, only the setting of the DMA4_GCR[13:12] value forces the thread reservation to these values. Set the appropriate value to avoid losing threads using only regular channels.

To have an independent read and write priority context, a per-channel bit (DMA4_CCRi[26]) is added for write priority, and the previous priority bit becomes read priority bit (DMA4_CCRi[6]).

Note

The device has one priority bit per logical channel, not one priority bit per port.

16.1.4.11 FIFO Budget Allocation

To avoid fully occupying the FIFO with a high-priority transfer while low-priority channels wait in the arbitration queue, two separate FIFO budgets are specified: one for high-priority channels and one for low-priority channels. This is defined in the DMA4_GCR register, allowing the user to share the FIFO budget between the low- and high-priority channels. The amount of the FIFO allocated by the low- and high-priority channels is fixed by the value set in the DMA4_GCR[15:14] HI_LO_FIFO_BUDGET field. The maximum channel FIFO depth is limited by the HI_LO_FIFO_BUDGET field as follows:

If the channel is low priority:

- When HI_LO_FIFO_BUDGET = 0x1, then low priority cannot exceed 75 percent of the total FIFO.
- When HI_LO_FIFO_BUDGET = 0x2, then low priority cannot exceed 25 percent of the total FIFO.
- When HI_LO_FIFO_BUDGET = 0x3, then low priority cannot exceed 50 percent of the total FIFO.

If channel is high priority

- When HI_LO_FIFO_BUDGET = 0x1, then high priority cannot exceed 25 percent of the total FIFO.
- When HI_LO_FIFO_BUDGET = 0x2, then high priority cannot exceed 75 percent of the total FIFO.
- When HI_LO_FIFO_BUDGET = 0x3, then high priority cannot exceed 50 percent of the total FIFO.

The user must perform the following equation:

- For a high-priority channel: (Per_Channel_Maximum FIFO Depth + 1) x Number of High Channel =< High Budget FIFO
- For a low-priority channel: (Per_Channel_Maximum FIFO Depth + 1) x Number of Low Channel =< Low Budget FIFO

Note

Ensure that *Number of High Channel* means *Number of Active High-Priority Channel* and that *Number of Low Channel* means *Number of Active Low-Priority Channel*.

16.1.4.12 Chained Logical Channel Transfers

Chaining multiple logical channels permits transfers consisting of multiple parts to be executed without repeated software intervention. This results in better performance than the alternative of software setting up and starting each transfer separately. Each part of a chained transfer can have the data addressed in a different manner that permits the programming of a variety of complex transfers. For example:

- Interlaced video data with one logical channel configured to transfer the even lines and another logical channel configured to transfer the odd lines
- Protocol headers with a separate DMA4 channel configured to transfer each field in the header

Channels can be chained through each channel DMA4_CLNK_CTRLi register. When the transfer for the first channel completes, the next channel in the chain is enabled. The number of channels in the chain that are configured for hardware-synchronized transfers is flexible (although typically it may be all, none, or simply the first one). The DMA request line number must be set to 0 to specify that any or all of the channels in a chain are software-triggered or nonsynchronized.

The last channel in a chain can be chained to the first channel to create a continuously looping chain. The continuously looping transfer can be stopped on the fly at a specific channel by disabling the

DMA4_CLNK_CTRLi[15] ENABLE_LNK bit. The looping transfer stops after the specified channel transfer is complete.

Note

A DMA request line must not be shared between concurrently enabled DMA channels. However, a DMA request line can be shared between several chained logical channels.

For more information about the programming model, see Section 16.1.5, *DMA_SYSTEM Basic Programming Model*.

16.1.4.13 Reprogramming an Active Channel

A currently active logical DMA channel can be disabled through the DMA4_CCRi[7] ENABLE bit. When an ongoing transaction is complete and the read-active and write-active bits in the DMA4_CCRi register (DMA4_CCRi[9] RD_ACTIVE and DMA4_CCRi[10] WR_ACTIVE) are reset, the channel can be reprogrammed for a new transfer.

16.1.4.14 Packet Synchronization

A packet transfer notion is related to the behavior of some peripherals, which have certain buffering capability and requires to transfer the buffer content once an element number threshold is reached (a hardware DMA request is generated). To associate a frame synchronization to each DMA request is possible, but this limits the maximum transfer size. Indeed the maximum transfer size is proportional to the FIFO depth of the peripheral:

maximum_transfer_size =peripheral_FIFO_depth x number_of_frame_in_block

The packet synchronization allows to dissociate the transfer size from the FIFO depth of the peripheral. Only Constant addressing mode is allowed on a read port or a write port if source target or destination target is packet synchronized respectively.

Example:

Consider a camera interface with a FIFO_depth of 128 words and a FIFO_element_number_threshold of 128, and a picture to transfer with a size 320 lines by 240 columns. If frame synchronization is associated with each DMA request then the maximum transfer size that can be performed is $128x2^{16}$ words. In this case, a frame is 128-word long, which does not fit the size of a line, and it is not possible to generate an interrupt at the end of line. However, by introducing the packet transfer notion, which is related to the peripheral FIFO behavior/structure, the maximum transfer size (maximum_transfer_size = 2^{24} x 2^{16} words) is independent of both peripheral_FIFO_depth and FIFO_element_number_threshold. This allows a long-enough transfer within one channel context and rotation operation on a large image format.

The main features of DMA Packet transfer are as follows:

- DMA Packet_Data_Size for each DMA Request: The Peripheral_element_number_threshold (the number of elements in a packet) shares the DMA4_CSFIi and DMA4_CDFIi configuration registers. If the peripheral is the source target, the addressing mode is constant, and the DMA4_CSFIi[15:0] bit field is used to specify the packet data size in the DMA4_CSFIi register. The user must set the DMA.SDMA4_CCRi[24] SEL_SRC_DST_SYNC bit to 1. If the peripheral is the destination target, the addressing mode is constant, the DMA4_CDFIi[15:0], is used to specify the packet data size (PKT_ELNT_NBR), and the bit field [31:16] is unused. To specify the packet data size in the DMA4_CDFIi register, the user must set the DMA4_CCRi[24] SEL_SRC_DST_SYNC bit to 0.

Note

The packet size can be a submultiple or non-submultiple of a frame size. If DMA Packet_Data_Size is aligned on the DMA channel block data size boundary, then DMA transfers the last data in the channel block boundary and stops at the block boundary for the last packet DMA Request. If the Packet_Data_size is not aligned on the block boundary, the remaining data smaller than a packet size are transferred using burst or single accesses to complete the block.

- DMA Packet_Data_Transfer does not affect DMA channel capabilities in term of packing and bursting.

  The packet synchronization mode is active when DMA4_CCRi[5] FS = DMA4_CCRi[18] BS = 1. Then:
  - If DMA4_CCRi[24] SEL_SRC_DST_SYNC = 0, the DMA4_CDFIi[15:0] bit field gives the number of elements in the packet and the DMA4_CDFIi[31:16] bit field is unused for the packet size.
  - If DMA4_CCRi[24] SEL_SRC_DST_SYNC = 1, the DMA4_CSFIi[15:0] bit field gives the number of elements in the packet and the DMA4_CSFIi[31:16] bit field is unused for the packet size.

Note

The maximum transfer size, regardless of the packet size, is always:

Block_size = Number_of_Frame_in_Block x Number_of_Element_in_Frame x Element_Size.

If DMA channel packet/burst access is across the packet boundary, the DMA hardware automatically splits this packing/burst access into multiple smaller accesses that are aligned on the packet boundary. Otherwise, the DMA transfers data as a usual packing/burst access.

16.1.4.15 Graphics Acceleration Support

The DMA_SYSTEM supports two graphic acceleration features: transparent copy and constant fill.

Only one of these features can be enabled at a time through the DMA4_CCRi register for the particular logical DMA channel.

The transparent copy feature enables specification of a particular color through the DMA4_COLORi register so that when it is recognized in the data from the source, it is not copied to the corresponding location in the destination but instead leaves the data in the corresponding location in the destination as it is.

Figure 16-11 shows the 2-D graphic transparent color block diagram.

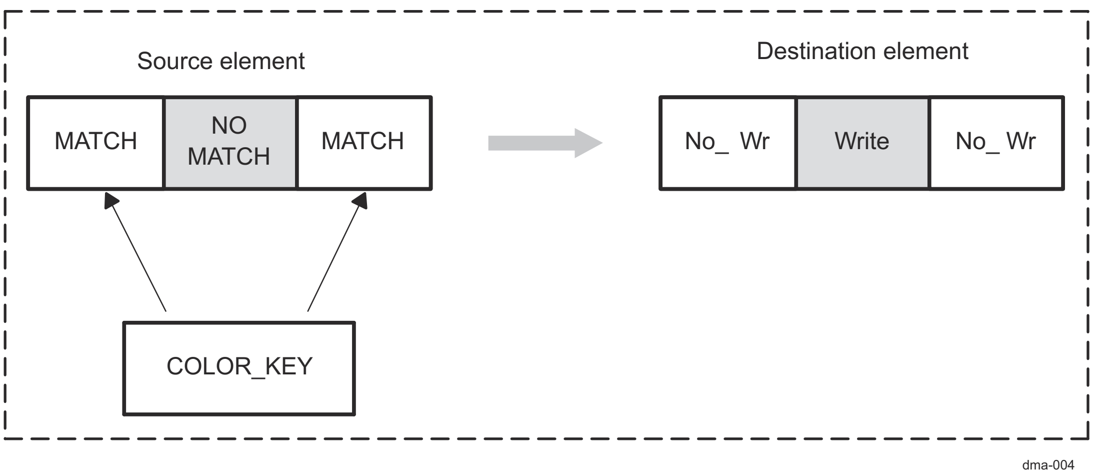


Figure 16-11. 2-D Graphic Transparent Color Block Diagram

The constant fill feature provides the ability to specify a particular color through the DMA4_COLORi register for every specified location in the destination. In this case, the transfer consists only of writing to the destination without reading from a source.

Both features support 8, 16, and 24 bpp, depending on what is specified as the DMA transfer ES through the DMA4_CSDPi register. An ES of 32 bits corresponds to 24 bpp. During a 32-bit (24 bpp) transfer, the 8 most-significant bits (MSBs) ([31:24]) are 0. Both features are compatible with packed and burst transactions.

16.1.4.16 Supervisor Modes

A logical DMA channel can be configured to operate in supervisor mode through the DMA4_CCRi[22] SUPERVISOR bit. This must be done using supervisor access. Once a channel is configured in supervisor mode, the channel configuration is protected from nonsupervisor accesses. All DMA transactions on a supervisor channel are supervisor transactions.

16.1.4.17 Posted and Nonposted Writes

A logical channel can be configured in its DMA4_CSDPi[17:16] bits to use one of three write access handshake modes for the destination:

- Nonposted write: Each write must complete before transfer can continue or complete.
- Posted write: Transfer continues without waiting for each write to complete (may improve performance with slow devices).
- Posted with final write nonposted: Transfer continues without waiting for each write to complete, but final write completes before transfer can complete.

16.1.4.18 Disabling a Channel During Transfer

When a channel is disabled during a transfer, the channel undergoes an abort, unless it is hardware-source-synchronized with buffering enabled (DMA4_CCRi[25] BUFFERING_DISABLE = 0). If this is the case, the FIFO is drained to prevent the loss of data. For more information about this feature, see Section 16.1.4.19, *FIFO Draining Mechanism*.

16.1.4.19 FIFO Draining Mechanism

When a source-synchronized channel is disabled during a transfer, the current hardware request (element/packet/frame/block) service completes and the channel DMA4_CCRi[9] RD_ACTIVE bit is set to 0, which means the channel is not active on the read port. The remaining data in the corresponding disabled channel FIFO is drained onto the write port and transferred to the programmed destination as in normal transfer.

At the end of the draining the DMA4_CCRi[10] WR_ACTIVE bit is set to 0 (channel is no longer active on the write port) and if the DMA4_CICRi[12] DRAIN_END_IE is set to 1, the DMA4_CSRi[12] DRAIN_END status bit is updated and an interrupt is generated.

Once a channel is disabled during a transfer, it must wait for the DMA4_CCRi[9] RD_ACTIVE and DMA4_CCRi[10] WR_ACTIVE bits to become 0 before being reenabled for a new transfer. The FIFO drain for a channel occurs only in the following cases:

- If the channel is a source synchronized channel and DMA4_CCRi[25] BUFFERING_DISABLE = 0
  and
- If the channel is not a solid fill channel
  and
- If the channel is not a transparent and copy channel
  and
- If the channel is a hardware, synchronized channel

Note

For a self-linked or chain-linked channel, the user must disable the DMA4_CLNK_CTRLi[15] ENABLE_LINK bit before disabling the channel.

In all other cases, the channel undergoes an abort.

16.1.4.20 Linked List

16.1.4.20.1 Overview

The DMA_SYSTEM supports the logical transfer-descriptor loader feature. A transfer descriptor represents a set of values that maps to a set of logical channel configuration registers.

A logical channel transfer descriptor can be loaded by DMA from memories, and then successive transfer descriptors can be autonomously loaded based on a linked-list scheme. This enables DMA4 scatter-gather transfers with minimum MPU support by removing successive channel configuration processing and associated interrupt handling overheads. It also optimizes DMA4 channel resources by enabling efficient transfer serialization on a single logical channel versus concurrent (multiple) logical channel use.

Different types of transfer descriptors are supported (full or partial logical channel configuration registers are set). This optimizes the memory size required for storing a long linked list, because parameter changes are limited to only a few logical channel configuration registers.

16.1.4.20.2 Link-List Transfer Profile

A linked-list transfer can be seen as a super-block transfer (where the block is composed of FN frames and each frame includes EN elements). The block size (FN x EN x ES) can be changed in the linked list by loading an updated transfer descriptor.

The end of the super block is signaled in the last descriptor associated with the last block. Generally, for a given link-list transfer, the logical channel is set at the beginning of the transfer and the logical channel configurations for the subsequent blocks are slightly changed. Thus, the descriptor can be limited to an update of only few parameters, such as FN or EN. This assumes that the content of unmodified registers is preserved when a new descriptor is loaded.

A transfer descriptor is composed of a set of channel configuration register values with the addition of the next-descriptor pointer register (DMA4_CNDPi) and a channel-descriptor parameter register (DMA4_CDPi). The next-descriptor pointer is the 32-bit address pointer from where the next transfer descriptor is to be loaded. The next-descriptor pointer is mapped depending on the descriptor type (1, 2, or 3).

16.1.4.20.3 Descriptors

A transfer descriptor is a set of values that maps to a set of logical channel configuration registers. The descriptor contains the parameters associated with a transfer profile (transfer size, source or destination addresses, etc). Four different types of transfer descriptors are supported to optimize the memory size required to store a long linked list and to minimize MPU use to create and maintain the descriptor list.

A transfer descriptor is a list of 32-bit values. A descriptor must be 32-bit aligned in memory. Only the 30 least-significant bits (LSBs) of the next-descriptor address pointer are updated from the descriptor, and the DMA4 forces the 2 LSBs to 0 on generation of the pointer address. The descriptor size is variable, depending on the descriptor type and the Nxt_Dv and Nxt_Sv bit fields.

Transfer descriptor bit mapping is the same as DMA4 logical-channel configuration register bit mapping, with the following exceptions:

- Src_Element_index and Dst_Element_index are concatenated in the same 32-bit location.
- DMA4_CICRi (interrupt event mask)
- CFN (frame number)
- Bit fields:
  – P: Corresponds to the PAUSE_LINK_LIST bit:
    - When set to 1 in the descriptor, the channel is suspended when the descriptor load completes.
    - The user must not set the PAUSE_LINK_LIST bit through the configuration port. Otherwise, behavior is undefined.
    - When set to 0 (through the configuration port) after pause, the linked-list channel resumes its transfer (descriptor load or data load).
  – B: Corresponds to the end-of-block enable bit (BLOCK_IE) of the DMA4_CICRi register; valid only for type 3. This value is don't care for descriptor types 1 and 2, where DMA4_CICRi is fully specified.
  – Nxt_Dv, Nxt_Sv: Mapped in the DMA4_CDPi register. They indicate one of the following possibilities:
    - Next descriptor contains an updated destination or source address.
    - Next descriptor does not update the source or destination address, but increments the last source or destination address (from the end of the last transfer).
    - The next source address and/or destination address are the last valid ones in the configuration memory. This means that the corresponding location in the configuration memory is not updated (assuming that they were initialized at least once in the past). This is also called wrapping addressing.

– Next_Descriptor_Type: Specifies the next descriptor type that corresponds to the NEXT_DESCRIPTOR_TYPE bit field in the DMA4_CDPi register

16.1.4.20.3.1 Type 1

A type 1 descriptor includes the overall channel configuration register value to be loaded (global registers are not part of the type 1 descriptor). This descriptor is used primarily when major changes are required:

- Channel read or write access profiles must be modified; for example, bursting and packing (included in the DMA4_CSDPi register)
- Attach a new DMA request to the same channel or change the priority or access privilege (included in the DMA4_CCRi register)
- Enable solid or transparent color fill (included in the DMA4_CCRi and DMA4_COLORi registers)
- Enable a channel link (included in the DMA4_CLNK_CTRLi register)

Table 16-14 shows a type 1 descriptor.

Table 16-14. Type 1

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Ptr+ 0x2C</td><td colspan="32">CCR</td></tr>
<tr><td>Ptr+ 0x28</td><td colspan="32">CLNK_CTRL</td></tr>
<tr><td>Ptr+ 0x24</td><td colspan="32">CSDP</td></tr>
<tr><td>Ptr+ 0x20</td><td colspan="32">COLOR</td></tr>
<tr><td>Ptr+ 0x1C</td><td colspan="32">Src_Frame_index/Src_Packet_size</td></tr>
<tr><td>Ptr+ 0x18</td><td colspan="32">Dst_Frame_index/Dst_Packet_size</td></tr>
<tr><td>Ptr+ 0x14</td><td colspan="16">Src_Element_index</td><td colspan="16">Dst_Element_index</td></tr>
<tr><td>Ptr+ 0x10</td><td colspan="16">CICR (interrupt events mask)</td><td colspan="16">CFN frame number</td></tr>
<tr><td>Ptr+ 0xC</td><td colspan="32">Destination_Start_Address</td></tr>
<tr><td>Ptr+ 0x8</td><td colspan="32">Source_Start_Address</td></tr>
<tr><td>Ptr+ 0x4</td><td colspan="3">N_type</td><td>B</td><td colspan="2">Dv</td><td colspan="2">Sv</td><td colspan="24">Element_number</td></tr>
<tr><td>Ptr</td><td colspan="30">Next_descriptor_address_pointer</td><td>Rsv</td><td>P</td></tr>
</table>

16.1.4.20.3.2 Type 2

A type 2 descriptor includes the overall logical channel transfer address register and transfer format register to be loaded. This descriptor enables 2D addressing linked-list transfer (for example, a multimedia application where 2D objects are moved in a link). Table 16-15 shows a type 2 descriptor with source and destination address updates. Table 16-16 shows a type 2 descriptor with one source or destination address update.

Table 16-15. Type 2 With Source and Destination Address Updates

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Ptr+ 0x1C</td><td colspan="32">Src_Frame_index/Src_Packet_size</td></tr>
<tr><td>Ptr+ 0x18</td><td colspan="32">Dst_Frame_index/Dst_Packet_size</td></tr>
</table>

Table 16-15. Type 2 With Source and Destination Address Updates (continued)

Ptr+ 0x14	Src_Element_index																Dst_Element_index															
Ptr+ 0x10	CICR (interrupt events Mask)																CFN frame number															
Ptr+ 0xC	Destination_Start_Address																															
Ptr+ 0x8	Source_Start_Address																															
Ptr+ 0x4	N_type			B	Dv		Sv		Element_number																							
Ptr	Next_descriptor_address_pointer																														Rsv	P

Table 16-16. Type 2 With Source or Destination Address Update

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ptr+ 0x18	Src_Frame_index/Src_Packet_size																															
Ptr+ 0x14	Dst_Frame_index/Dst_Packet_size																															
Ptr+ 0x10	Src_Element_index																Dst_Element_index															
Ptr+ 0xC	CICR (interrupt events Mask)																CFN frame number															
Ptr+ 0x8	Source_Start_Address or Destination_Start_Address																															
Ptr+ 0x4	N_type			B	Dv		Sv		Element_number																							
Ptr	Next_descriptor_address_pointer																														Rsv	P

16.1.4.20.3.3 Type 3

A type 3 descriptor is limited to a few logical channel transfer address registers and transfer format registers to be loaded. This descriptor enables simple 1D addressing link transfer (for example, scatter-gather or ping-pong memory movement using a linked list). Table 16-17 shows a type 3 descriptor with source and destination address updates. Table 16-18 shows a type 3 descriptor with one source or address destination update.

Table 16-17. Type 3 With Source and Destination Address Updates

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ptr+ 0xC	Destination_Start_Address																															
Ptr+ 0x8	Source_Start_Address																															
Ptr+ 0x4	N_type			B	Dv		Sv		Element_number																							
Ptr	Next_descriptor_address_pointer																														Rsv	P

Table 16-18. Type 3 With Source or Destination Address Update

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ptr+ 0x8	Source_Start_Address or Destination_Start_Address																															
Ptr+ 0x4	N_type			B	Dv		Sv		Element_number																							

Table 16-18. Type 3 With Source or Destination Address Update (continued)

Ptr	Next_descriptor_address_pointer	Rsv	P

16.1.4.20.4 Linked-List Control and Monitoring

16.1.4.20.4.1 Transfer Mode Setting

Four descriptor types are available in DMA4_CDPi[9:8] TRANSFER_MODE to distinguish the different transfer modes:

- DMA4_CDPi[9:8] TRANSFER_MODE = 00: The current channel is using normal mode.
- DMA4_CDPi[9:8] TRANSFER_MODE = 01: The current channel is using link-list channel mode for a type 1, 2, or 3 descriptor.

The reset value is normal mode (DMA4_CDPi[9:8] TRANSFER_MODE = 0).

16.1.4.20.4.2 Starting a Linked List

Like a nonlinked-list transfer, a link transfer starts under host control by enabling the associated logical channel (set the DMA4_CCRi[7] ENABLE bit to 1). The DMA4_CDPi[10] FAST bit sets the start mode of the link-list transfer:

In nonfast-start mode, the logical channel configuration is fully initialized so that the transfer can start without descriptor loading.

In fast-start mode, the descriptor pointer and other inputs are given. The channel starts by loading the descriptor and then starts the data transfer phase.

16.1.4.20.4.3 Monitoring a Linked-List Progression

In addition to the DMA4_CCENi (remaining elements) and DMA4_CCFNi (remaining frames) registers that are used to monitor the transfer progress, a per-channel register, DMA4_CCDNi (channel current active descriptor number), monitors which descriptor in the list is active. The user must initialize the DMA4_CCDNi register to 0 during the initial configuration. When the DMA4_CCDNi register is updated, the DMA4_CCFNi and the DMA4_CCENi registers are updated. The user must also initialize the DMA4_CCFNi and DMA4_CCENi registers to 0xFFFF and to 0xFFFFFF, respectively, to track the effective transfer start of synchronized transfer.

16.1.4.20.4.4 Interrupt During Linked-List Execution

Any logical channel source of interrupt can be triggered during a linked-list execution, if the interrupt source is enabled during the initial configuration in CICR. The DMA4_CICRi register can also be updated during the linked-list execution if descriptor types 1 and 2 are used.

The use of an interrupt event in a link execution can be difficult, because the link can progress in parallel with interrupt service routine (ISR) execution. This makes it difficult to synchronize them unless system assumptions are used. The most appropriate synchronization model is to get an interrupt-only on linked-list completion, when the last transfer block is complete. This prevents the interrupt from occurring during the link execution. An end-of-super-block interrupt event available in the DMA4_CICRi and DMA4_CSRi registers can be enabled at initial configuration or when using descriptor types 1 and 2. To prevent the use of descriptor type 1 or 2 to update BLOCK_IE (full DMA4_CICRi update), a dedicated BLOCK_IE bit field is also available in a type 3 descriptor.

16.1.4.20.4.5 Pause a Linked List

When the channel is suspended, it remains enabled.

The pause behaves differently, depending on the transfer mode:

- Normal transfer mode: If the user sets the DMA4_CDPi[7] PAUSE_LINK_LIST bit to 1, the channel completes the current read and write transactions and then suspends the channel. The channel can be resumed by setting the channel DMA4_CDPi[7] PAUSE_LINK_LIST bit to 0.
- Linked-list type 1, 2, or 3 mode: The user must not set the DMA4_CDPi[7] PAUSE_LINK_LIST bit through the configuration port; otherwise, transfer behavior is undefined.

A PAUSE_LINK_LIST bit (P) is set to 1 in the descriptor.
  – The channel is suspended after the descriptor load, translation, and configuration memory update are complete.
  – The linked list can be resumed by resetting the DMA4_CDPi[7] PAUSE_LINK_LIST bit (through the configuration port).

16.1.4.20.4.6 Stop a Linked List (Abort or Drain)

The channel can be stopped for a drain or an abort. These cases are exclusive.

16.1.4.20.4.6.1 Drain

- Drain conditions:

  A channel is a drain candidate if it is a hardware-source-synchronized transfer with DMA4_CCRi[25] BUFFERING_DISABLE = 0 and should not be doing any of the graphics operation (transparent copy or solid-color fill).
- Drain trigger:

  A drain candidate channel is drained if it is disabled (DMA4_CCRi[7] ENABLE = 0) or if it receives a transaction error on the read port.
- Drain behavior with a type 1, 2, or 3 descriptor. Drain trigger can occur in two situations:
  – During descriptor loading: Any ongoing current transaction is complete and the channel is aborted.
  – During data loading: The read is completed at the boundary of the request (element/frame/packet/block boundary), the FIFO is drained to the destination, and then a DRAIN_END interrupt can be asserted.

16.1.4.20.4.6.2 Abort

- Abort condition:

  A channel is an abort candidate if it is software-synchronized, hardware-destination-synchronized, solid color-fill, transparent-color fill, or hardware-source-synchronized with DMA4_CCRi[25] BUFFERING_DISABLE = 1.
- Abort trigger:

  A channel is an abort candidate if it is disabled (DMA4_CCRi[7] ENABLE = 0), if it receives a transaction error on the read or write port, or if there is a MISALIGNMENT_ERROR.
- Abort behavior with a type 1, 2, or 3 descriptor:

  If an abort trigger occurs, the channel aborts immediately after completion of current read/write transactions and then the FIFO is cleaned up.

  In type 1, 2, or 3, if an abort trigger or drain trigger occurs during the descriptor load phase, the channel aborts.

16.1.4.20.4.7 Status Bit Behavior

This section describes the behavior of the DMA4_CSRi[6] SYNC, DMA4_CCRi[9] RD_ACTIVE and DMA4_CCRi[10] WR_ACTIVE status bits:

- For a hardware-synchronized channel in linked-list mode, the DMA4_CSRi[6] SYNC bit becomes active (DMA4_CSRi[6] SYNC = 1) when the first data load transaction is scheduled and remains active until the last data load transaction in the block (not super block) is descheduled (DMA4_CSRi[6] SYNC = 0). The SYNC bit is not active during the descriptor load phase.
- The DMA4_CCRi[9] RD_ACTIVE bit is active during the data load phase and the descriptor load phase. It becomes active when the first read transaction is scheduled. It becomes inactive:
  – When (during the descriptor load phase) the last descriptor write request is descheduled
  – When (during the data load phase) the last read transaction in the block (not super block) is descheduled for software-synchronized transfer or destination-synchronized transfer with prefetch enabled
  – When (during the data load phase) the last read transaction in the request (element/frame/packet/block sync) is descheduled for hardware-source-synchronized transfer or hardware-destination-synchronized transfer without prefetch

- The DMA4_CCRi[10] WR_ACTIVE bit is active only during the data load phase. It becomes active when the first write transaction is scheduled and becomes inactive:
  - Until the last write transaction in the block (not super block) is descheduled and the FIFO is cleaned up for software-synchronized transfer
  - Until the last write transaction in the request (element/frame/packet/block sync) is descheduled and the FIFO is cleaned up for hardware-source-synchronized transfer (with DMA4_CCRi[25] BUFFERING_DISABLE = 0) or hardware-destination-synchronized transfer.

16.1.4.20.4.8 Linked-List Channel Linking

Channel linking for inter- and intra-super blocks is supported for type 1, 2, and 3 descriptors.

Assume that CHx and CHz are linked-list channels using generic descriptors. If CHx is composed of N descriptors and CHz is composed of M descriptors, then in nonfast mode:

CHx: CHx[Data1]-> CHx[DES1] -> . -> CHx[DESN]->CHx[DataN + 1]

CHz: CHz[Data1]-> CHz[DES1] -> . -> CHz[DESM]->CHz[DataM + 1]

It is possible to link CHx to CHz or CHx to itself after the completion of the CHx transfer (end of super block). To do this, the user must set the DMA4_CLNK_CTRLi[15] ENABLE_LNK bit to 1 and the DMA4_CLNK_CTRLi[4:0] NEXTLCH_ID bit to z (or to x for self linking) through the last descriptor using a type 1 descriptor. The sequence is:

CHx: CHx[Data1]-> CHx[DES1] -> . -> CHx[DESN]-CHx[DataN+1] -> CHz: CHz[Data1]-> CHz[DES1] -> . -> CHz[DESM]->CHz[DataM+1]

It is also possible to link CHx to CHz during the CHx transfer and before the end of super block. The user must set the DMA4_CLNK_CTRLi[15] ENABLE_LNK bit to 1 and the DMA4_CLNK_CTRLi[4:0] NEXTLCH_ID bit to z through descriptor p (CHx[DESp]) using a type 1 descriptor. The sequence is:

CHx: CHx[Data1]-> CHx[DES1] ->.-> CHx[DESp]->CHx[Data(p + 1)] -> CHz[Data1]-> CHz[DES1] -> .

The user must continue the linking until channels CHx and CHz complete their super-block transfers; otherwise, the channels remain enabled.

Note

In channel linking, the head of a chain can be in fast mode or nonfast mode. All channels that are not in the head of the chain can be in nonfast mode only. In self-linking, the channel cannot be in fast mode.

Note

If channel CHx links to CHz in the middle of the superblock transfer (remember link bit can be set through Type-1 descriptor load), CHx is disabled after the corresponding data load and enables the channel CHz.

16.1.5 DMA_SYSTEM Basic Programming Model

16.1.5.1 Setup Configuration

After a hardware reset, program all fields in the logical channel registers to default values for any channels used, because most fields are undefined following reset.

Before programming any DMA transfers, the priority arbitration rate and the maximum FIFO depth must be configured through the DMA4_GCR register, and any required interrupts must be enabled through the DMA4_IRQENABLE_Lj registers and the logical channel DMA4_CICRi registers.

Software clears the DMA4_CSRi register and the IRQSTATUS bit for the different interrupt lines before enabling the channel.

16.1.5.2 Software-Triggered (Nonsynchronized) Transfer

To program a software-triggered DMA transfer:

1. Configure the transfer parameters in the logical DMA channel registers:
   - DMA4_CSDPi:
     - Transfer ES (8, 16, or 32 bits) in the DMA DMA4_CSDPi[1:0] bit field.
     - Read and write port access types (single/burst), DMA DMA4_CSDPi[8:7] and DMA4_CSDPi[15:14] bit fields
     - Source and destination endianism, DMA DMA4_CSDPi[21] and DMA4_CSDPi[19] bits
     - Write mode (posted or nonposted) and DMA DMA4_CSDPi[17:16] bit field
     - Source or destination packed or nonpacked (if the ES is less than the read/write port size), DMA DMA4_CSDPi[6] and DMA4_CSDPi[13] bits
   - DMA4_CENi: EN
   - DMA4_CFNi: FN per transfer block
   - DMA4_CSSAi and DMA4_CDSAi: Source and destination start address (aligned with transfer ES)
   - DMA4_CCRi:
     - Read and write port addressing modes, DMA DMA4_CCRi[13:12] and DMA4_CCRi[15:14] bit field
     - Priority bit for both read and write ports, DMA DMA4_CCRi[6] and DMA4_CCRi[26] bits
     - DMA request number (set to 0 for a software-triggered transfer) and DMA register bit fields DMA4_CCRi[4:0] = 0 and DMA4_CCRi[20:19] = 0
   - DMA4_CSEIi, DMA4_CSFIi, DMA4_CDEIi, and DMA4_CDFIi: Source and destination element and frame indexes (depending on addressing mode)
2. Start the transfer through the enable bit in the channel DMA4_CCRi register and DMA DMA4_CCRi[7] bit

The following example performs a DMA transfer on channel 10 of a 240*160 picture from RAM to RAM (0x80C00000 to 0x80F00000):

```
UWORD32 RegVal = 0;
DMA4_t *DMA4;
DMA4 = (DMA4_t
      *)malloc(sizeof(DMA4_t));
/* Init. parameters
      */
DMA4->DataType = 0x2; //
      DMA4_CSDPi[1:0]
DMA4->ReadPortAccessType = 0; //
      DMA4_CSDPi[8:7]
DMA4->WritePortAccessType = 0; //
      DMA4_CSDPi[15:14]
DMA4->SourceEndiansim = 0; //
      DMA4_CSDPi[21]
DMA4->DestinationEndianism = 0; //
      DMA4_CSDPi[19]
DMA4->WriteMode = 0; //
      DMA4_CSDPi[17:16]
DMA4->SourcePacked = 0; //
      DMA4_CSDPi[6]
DMA4->DestinationPacked = 0; //
      DMA4_CSDPi[13]
```

```
DMA4->NumberOfElementPerFrame = 240; //
      DMA4_CENi
DMA4->NumberOfFramePerTransferBlock = 160; //
      DMA4_CFNi
DMA4->SourceStartAddress = 0x80C00000; //
      DMA4_CSSAi
DMA4->DestinationStartAddress = 0x80F00000; //
      DMA4_CDSAi
DMA4->SourceElementIndex = 1; //
      DMA4_CSEIi
DMA4->SourceFrameIndex = 1; //
      DMA4_CSFIi
DMA4->DestinationElementIndex = 1; //
      DMA4_CDEIi
DMA4->DestinationFrameIndex = 1; //
      DMA4_CDFIi
DMA4->ReadPortAccessMode = 1; //
      DMA4_CCRi[13:12]
DMA4->WritePortAccessMode = 1; //
      DMA4_CCRi[15:14]
DMA4->ReadPriority = 0; //
      DMA4_CCRi[6]
DMA4->WritePriority = 0; //
      DMA4_CCRi[23]
DMA4->ReadRequestNumber = 0; //
      DMA4_CCRi[4:0]
DMA4->WriteRequestNumber = 0; //
      DMA4_CCRi[20:19]
/* 1) Configure the transfer
      parameters in the logical DMA registers
      */
/*-------------------------------------------------------------------*/
/*
      a) Set the data type CSDP[1:0], the Read/Write Port access type
      CSDP[8:7]/[15:14], the Source/dest endianism CSDP[21]/CSDP[19], write
      mode CSDP[17:16], source/dest packed or non-packed
      CSDP[6]/CSDP[13]*/

// Read CSDP
RegVal =
      DMA4_CSDP_CH10;
// Build reg
RegVal = ((RegVal &~ 0x3)
      | DMA4->DataType );
RegVal = ((RegVal &~(0x3 << 7)) |
      (DMA4->ReadPortAccessType << 7));
RegVal = ((RegVal &~(0x3 << 14)) |
      (DMA4->WritePortAccessType << 14));
RegVal = ((RegVal &~(0x1 << 21)) |
      (DMA4->SourceEndiansim << 21));
RegVal = ((RegVal &~(0x1 << 19)) |
      (DMA4->DestinationEndianism << 19));
RegVal = ((RegVal &~(0x3 << 16)) |
      (DMA4->WriteMode << 16));
RegVal = ((RegVal &~(0x1 << 6)) |
      (DMA4->SourcePacked << 6));
RegVal = ((RegVal &~(0x1 << 13)) |
      (DMA4->DestinationPacked << 13));
// Write CSDP

DMA4_CSDP_CH10 = RegVal;
/* b) Set the number of
      element per frame CEN[23:0]*/
DMA4_CEN_CH10 =
      DMA4->NumberOfElementPerFrame;
/* c) Set the number of frame
      per block CFN[15:0]*/
DMA4_CFN_CH10 =
      DMA4->NumberOfFramePerTransferBlock;
/* d) Set the
      Source/dest start address index CSSA[31:0]/CDSA[31:0]*/

DMA4_CSSA_CH10 = DMA4->SourceStartAddress; // address start

DMA4_CDSA_CH10 = DMA4->DestinationStartAddress; // address dest
```

```
/* e) Set the Read Port addressing mode CCR[13:12], the
      Write Port addressing mode CCR[15:14], read/write priority
      CCR[6]/CCR[26], the current LCH CCR[20:19]=00 and CCR[4:0]=00000*/

// Read CCR
RegVal = DMA4_CCR_CH10;
//
      Build reg
RegVal = ((RegVal &~(0x3 << 12)) | (DMA4->ReadPortAccessMode      << 12));
RegVal = ((RegVal &~(0x3 << 14)) | (DMA4->WritePortAccessMode
      << 14));
RegVal = ((RegVal &~(0x1 << 6)) | (DMA4->ReadPriority << 6));

RegVal = ((RegVal &~(0x1 << 26)) | (DMA4->WritePriority << 26));

RegVal &= 0xFFCFFFE0 ;
// Write CCR
DMA4_CCR_CH10
      = RegVal;
/* f)- Set the source element index CSEI[15:0]*/

DMA4_CSEI_CH10 = DMA4->SourceElementIndex;
/* g)-
      Set the source frame index CSFI[15:0]*/
DMA4_CSFI_CH10 =
      DMA4->SourceFrameIndex ;
/* h)- Set the destination element
      index CDEI[15:0]*/
DMA4_CDEI_CH10 =
      DMA4->DestinationElementIndex;
/* i)- Set the destination
      frame index CDFI[31:0]*/
DMA4_CDFI_CH10 =
      DMA4->DestinationFrameIndex;
/* 2) Start the DMA transfer by
      Setting the enable bit CCR[7]=1 */

/*---------------------------------------------------------------*/

//write enable bit
DMA4_CCR_CH10 |= 1 << 7; /* start */
```

16.1.5.3 Hardware-Synchronized Transfer

To monitor a hardware synchronized DMA transfer, initialize the DMA4_CDACi register before the software enable.

To configure an LCh to synchronize by element, packet, frame, or block, the frame synchronization DMA4_CCRi[5] FS bit and the block synchronization DMA4_CCRi[18] BS bit must be programmed. For all the following synchronized transfers (element, packet, and frame or block-synchronized transfers), the user must first set the DMA4_CCRi[24] SEL_SRC_DST_SYNC bit to 1 when the source triggers on the DMA request and set it the DMA4_CCRi[24] SEL_SRC_DST_SYNC bit to 0 when the destination triggers on the DMA request.

Note

The user must take care when setting the DMA4_CCRi[23] PREFETCH bit it is in conjunction with DMA4_CCRi[24] SEL_SRC_DST_SYNC bit.

- To configure an LCh to transfer one element per DMA request:
  1. Set the number of DMA request associated with the current LCH in the DMA4_CCRi[20:19] SYNCHRO_CONTROL_UPPER and DMA4_CCRi[4:0] SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the DMA4_CSDPi[1:0] DATA_TYPE bit field.
  3. Set the Read Port access type (single or burst access) in the DMA4_CSDPi[8:7] SRC_BURST_EN bit field.
  4. Set the Write Port access type (single or burst access) in the DMA4_CSDPi[15:14] DST_BURST_EN bit field.
  5. Set the Read Port addressing mode in the DMA4_CCRi[13:12] SRC_AMODE bit field.

6. Set the Write Port addressing mode in the DMA4_CCRi[15:14] DST_AMODE bit field.
7. Set the Read start address in the DMA4_CSSAi[31:0] SRC_START_ADRS bit field.
8. Set the Write start address in the DMA4_CDSAi[31:0] DST_START_ADRS bit field.
9. Set both FS and BS to 0 in DMA4_CCRi[5] FS and DMA4_CCRi[18] BS.
10. Set to 1 the channel enable bit DMA4_CCRi[7] EN.

- To configure an LCh to transfer one frame per DMA request:
  1. Set the number of DMA request associated to the current LCH in the DMA4_CCRi[20:19] SYNCHRO_CONTROL_UPPER and DMA4_CCRi[4:0] SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the DMA4_CSDPi[1:0] DATA_TYPE bit field.
  3. Set the number of element per frame in the DMA4_CENi[23:0] CHANNEL_ELMNT_NBR bit field.
  4. Set the Read Port access type (single or burst access) in the DMA4_CSDPi[8:7] SRC_BURST_EN bit field.
  5. Set the Write Port access type (single or burst access) in the DMA4_CSDPi[15:14] DST_BURST_EN bit field.
  6. Set the Read Port addressing mode in the DMA4_CCRi[13:12] SRC_AMODE bit field.
  7. Set the Write Port addressing mode in the DMA4_CCRi[15:14] DST_AMODE bit field.
  8. Set the Read start address in the DMA4_CSSAi[31:0] SRC_START_ADRS bit field.
  9. Set the Write start address in the DMA4_CDSAi[31:0] DST_START_ADRS bit field.
  10. Set FS to 1 and BS to 0, respectively, in DMA4_CCRi[5] FS and DMA4_CCRi[18] BS.
  11. Set to 1 the channel enable bit DMA4_CCRi[7] EN.
- To configure an LCh to transfer one block per DMA request:
  1. Set the number of DMA request associated to the current LCH in the DMA4_CCRi[20:19] SYNCHRO_CONTROL_UPPER and DMA4_CCRi[4:0] SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the DMA4_CSDPi[1:0] DATA_TYPE bit field.
  3. Set the number of element per frame in the DMA4_CENi[23:0] CHANNEL_ELMNT_NBR bit field.
  4. Set in the DMA4_CFNi[15:0] CHANNEL_FRAME_NBR bit field the number of frame (transfers), to take place before the LCH gets disabled.
  5. Set the Read Port access type (single or burst access) in the DMA4_CSDPi[8:7] SRC_BURST_EN bit field.
  6. Set the Write Port access type (single or burst access) in the DMA4_CSDPi[15:14] DST_BURST_EN bit field.
  7. Set the Read Port addressing mode in the DMA4_CCRi[13:12] SRC_AMODE bit field.
  8. Set the Write Port addressing mode in the DMA4_CCRi[15:14] DST_AMODE bit field.
  9. Set the Read start address in the DMA4_CSSAi[31:0] SRC_START_ADRS bit field.
  10. Set the Write start address in the DMA4_CDSAi[31:0] DST_START_ADRS bit field.
  11. Set FS to 0 and BS to 1, respectively, in DMA4_CCRi[5] FS and DMA4_CCRi[18] BS.
  12. Set to 1 the channel enable bit DMA4_CCRi[7] EN.
- To configure an LCh to transfer one packet per DMA request:
  1. Set the number of DMA request associated to the current LCH in the DMA4_CCRi[20:19] SYNCHRO_CONTROL_UPPER and DMA4_CCRi[4:0] SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the DMA4_CSDPi[1:0] DATA_TYPE bit field.
  3. Set the number of elements per packet to transfer: If the packet requestor is in the source, set DMA4_CCRi [24] SEL_SRC_DST_SYNC to 1 and set the packet element number in the DMA4_CSFIi register and set the addressing mode of source to constant addressing in DMA4_CCRi[13:12] SRC_AMODE bit field; else, if the packet requestor is in the destination, set the DMA4_CCRi[24] SEL_SRC_DST_SYNC to 0 and set the packet element number in the DMA4_CDFIi register and set the addressing mode of destination to constant addressing in DMA4_CCRi[15:14] DST_AMODE bit field.
  4. Set the number of elements per frame in the DMA4_CENi[23:0] CHANNEL_ELMNT_NBR bit field.
  5. Set in the DMA4_CFNi[15:0] CHANNEL_FRAME_NBR bit field the number of frames (transfers), to take place before the LCH gets disabled.
  6. Set the element number in the packet in the DMA4_CSFIi[15:0] PKT_ELNT_NBR, if constant addressing or post-incremented addressing modes are used in the source side. However, the number of element in the packet is set in the DMA4_CDFIi[15:0] PKT_ELNT_NBR if constant addressing mode is used in the destination side.

7. Set the Read Port access type (single or burst access) in the DMA4_CSDPi[8:7] SRC_BURST_EN bit field.
8. Set the Write Port access type (single or burst access) in the DMA4_CSDPi[15:14] DST_BURST_EN bit field.
9. Set the Read Port addressing mode in the DMA4_CCRi[13:12] SRC_AMODE bit field.
10. Set the Write Port addressing mode in the DMA4_CCRi[15:14] DST_AMODE bit field.
11. Set the Read start address in the DMA4_CSSAi[31:0] SRC_START_ADRS bit field.
12. Set the Write start address in the DMA4_CDSAi[31:0] DST_START_ADRS bit field.
13. Set FS to 1 and BS to 1, respectively, in DMA4_CCRi[5] FS and DMA4_CCRi[18] BS.
14. Set to 1 the channel enable bit DMA4_CCRi[7] EN.

Note

It is possible to stop a transfer by disabling the channel by resetting the DMA4_CCRi[7] ENABLE bit.

16.1.5.4 Synchronized Transfer Monitoring Using CDAC

The DMA4_CDACi register is writable and non-initialized (value undefined). It can be initialized to monitor a transfer by applying the following programming model:

1. Write 0 in the DMA4_CDACi register.
2. Enable the channel.
3. If a time-out occurs, read DMA4_CDACi register.
4. If DMA4_CDACi != 0 (it is the value configured in DMA4_CDACi):

   This indicates that the corresponding transfer has started. The user can then rely on DMA4_CCENi and DMA4_CCFNi element and frame counters.

   Otherwise, if DMA4_CDACi = 0 (it is the value configured in the DMA4_CDACi):

   This indicates that the corresponding transfer did not start.

16.1.5.5 Concurrent Software and Hardware Synchronization

This section describes thread allocation only; it does not describe the entire transfer. Because synchronized transfers are latency critical, you must allocate a thread at least on the synchronized target side.

Even for multiple concurrent channels, thread reservation ensures that when a hardware DMA request arrives, the read/write scheduler finds available thread(s) to initiate a channel schedule and issue a read/write transaction.

Consider six concurrent channels:

- Channels 0, 1, 2, and 3 are dedicated to memory-memory transfer; they are software triggered and not synchronized.
- Channel 4 is dedicated to memory-peripheral transfer, hardware triggered, and synchronized on the write side.
- Channel 5 is dedicated to peripheral-memory transfer, hardware triggered, and synchronized on the read side.

To perform thread reservation:

1. Allow thread reservation for priority channel 4 and channel 5:

   Reserve one thread (Read ThreadID 0) on the read port and one thread (Write ThreadID 0) on the write port: set the DMA4_GCR[13:12] HI_THREAD_RESERVED bit field to 0x1.
2. Specify channel priority:

   Channel 4 is a write high priority channel: set DMA4_CCRi[26] WRITE_PRIORITY = 1.

   Channel 5 is a read high priority channel: set DMA4_CCRi[6] READ_PRIORITY = 1.

16.1.5.6 Chained Transfer

A chained DMA transfer can be programmed as follows:

1. Configure the transfer parameters for each logical DMA channel in the chain as in step 1 for either the synchronized or non-synchronized transfers described in Section 16.1.5.5, *Concurrent Software and Hardware Synchronization*.
2. For each channel in the chain, configure the DMA4_CLNK_CTRLi register as follows:
   - Next logical DMA channel number (for a looping chained transfer link last channel to first channel number), in the DMA4_CLNK_CTRLi[4:0] NEXTLCH_ID bit field.
   - Include the logical channel to the chain and enable link by setting the DMA4_CLNK_CTRLi[15] ENABLE_LNK bit.
   - For a non-looping chain, the last logical channel in the chain must have the DMA4_CLNK_CTRLi[15] ENABLE_LNK bit set to 0 to indicate the end of the chain.
3. Enable the transfer through the enable bit in the first logical channel DMA4_CCRi[7] ENABLE bit. All other channels in the chain must be disabled. Each channel is enabled automatically in turn when the previous logical channel transfer completes. A non-synchronized transfer starts immediately; a hardware-synchronized transfer starts when the DMA request line corresponding to the first DMA channel in the chain is asserted.

To stop a looping chained transfer, disable the DMA4_CLNK_CTRLi[15] ENABLE_LNK bit (by setting it to 0x0), of the final channel transfer.

In the RAM-to-RAM copy example, to copy in loop, it is possible to link channel 10 on itself. The following line can be added in the channel configuration:

```
/* g) Set link for loop */
DMA4_CLINK_CTRL_CH10 =
      0x0000800A;
```

16.1.5.7 90-Degree Clockwise Image Rotation

The 90-degree clockwise image rotation example described in Section 16.1.4.5, *Addressing Modes*, can be programmed as follows:

1. Configure the transfer parameters in the logical DMA channel registers:
   - DMA4_CSDPi:
     – Transfer ES = 32-bit (32 bpp), DMA4_CSDPi[1:0] DATA_TYPE bit field
     – Read and write port access types = maximum burst size supported by memory device, DMA4_CSDPi[8:7] SRC_BURST_EN and DMA4_CSDPi[15:14] DST_BURST_EN bit fields
     – Source and destination endianism, DMA4_CSDPi[21] SRC_ENDIAN and DMA4_CSDPi[19] DST_ENDIAN bits
     – Write mode = posted with last element nonposted, DMA4_CSDPi[17:16] WRITE_MODE bit field
     – Source and destination packed = Yes (although destination writes do not benefit because EI1), DMA4_CSDPi[6] SRC_PACKED and DMA4_CSDPi[13] DST_PACKED bits
   - DMA4_CENi: EN = 240
   - DMA4_CFNi: FN per transfer block = 160
   - DMA4_CSSAi: Source start address = 0x100000
   - DMA4_CDSAi: destination start address = 0x20013E
   - DMA4_CCRi:
     – Read and write port addressing modes = double-index addressing mode for both or post-increment addressing on source and double-index addressing on destination, DMA4_CCRi[13:12] SRC_AMODE and DMA4_CCRi[15:14] DST_AMODE bit fields
     – Low or high priority, DMA4_CCRi[6] READ_PRIORITY bit
     – DMA request number = 0 (for software-triggered transfer), DMA4_CCRi[4:0] SYNCHRO_CONTROL and DMA4_CCRi[20:19] SYNCHRO_CONTROL_UPPER bit fields
   - DMA4_CSEIi: Source EI = 1
   - DMA4_CSFIi: Source frame index = 1

- DMA4_CDEIi: destination EI = 637
- DMA4_CDFIi: destination frame index = 152967

2. Start the transfer through the enable bit in the channel DMA4_CCRi register.

The following parameters are used to perform this rotation from 0x80C00000 RAM address to 0x80F00000, with the same code used in Section 16.1.5.2, *Software-Triggered (Nonsynchronized) Transfer*:

```
/* Init. parameters */
DMA4->DataType = 0x2; //
      DMA4_CSDPi[1:0]
DMA4->ReadPortAccessType = 0x3; // DMA4_CSDPi[8:7]

DMA4->WritePortAccessType = 0x3; // DMA4_CSDPi[15:14]

DMA4->SourceEndiansim = 0; // DMA4_CSDPi[21]

DMA4->DestinationEndianism = 0; // DMA4_CSDPi[19]

DMA4->WriteMode = 0x2; // DMA4_CSDPi[17:16]
DMA4->SourcePacked
      = 0x1; // DMA4_CSDPi[6]
DMA4->DestinationPacked = 0x1; //
      DMA4_CSDPi[13]
DMA4->NumberOfElementPerFrame = 240; // DMA4_CENi

DMA4->NumberOfFramePerTransferBlock = 160; // DMA4_CFNi

DMA4->SourceStartAddress = 0x80C00000; // DMA4_CSSAi

DMA4->DestinationStartAddress = 0x80F00000; // DMA4_CDSAi

DMA4->SourceElementIndex = 1; // DMA4_CSEIi

DMA4->SourceFrameIndex = 1; // DMA4_CSFIi

DMA4->DestinationElementIndex = 637; // DMA4_CDEIi

DMA4->DestinationFrameIndex = -152967; // DMA4_CDFIi

DMA4->ReadPortAccessMode = 0x3; // DMA4_CCRi[13:12]

DMA4->WritePortAccessMode = 0x3; // DMA4_CCRi[15:14]

DMA4->ReadPriority = 0; // DMA4_CCRi[6]
DMA4->WritePriority =
      0; // DMA4_CCRi[23]
DMA4->ReadRequestNumber = 0; // DMA4_CCRi[4:0]

DMA4->WriteRequestNumber = 0; // DMA4_CCRi[20:19]
```

16.1.5.8 Graphic Operations

- Transparent copy:
  1. Set the DMA4_CCRi[17] TRANSPARENT_COPY_ENABLE bit to 1
  2. Set the DMA4_CCRi[16] CONST_FILL_ENABLE bit to 0
  3. Set the value of the key color in the DMA4_COLORi[15:0] COLOR_KEY bit field

  To perform this graphic operation, the following lines can be added to the example of Section 16.1.5.2, *Software-Triggered (Nonsynchronized) Transfer*.

  ```
  DMA4_CCR_CH10 &= ~(0x1 << 16);
  DMA4_CCR_CH10 |= 0x1 << 17;

  DMA4_COLOR_CH10 = 0x00000003;
  ```

- Solid Color fill:
  1. Set the DMA4_CCRi[16] CONST_FILL_ENABLE bit to 1
  2. Set the DMA4_CCRi[17] TRANSPARENT_COPY_ENABLE bit to 0
  3. Set the value of key the color in the DMA4_COLORi[15:0] SOLID_COLOR bit field

To perform this graphic operation, the following lines can be added to the example of Section 16.1.5.2, *Software-Triggered (Nonsynchronized) Transfer*.

```
DMA4_CCR_CH10 &= ~(0x1 << 17);
DMA4_CCR_CH10 |= 0x1 << 16;

DMA4_COLOR_CH10 = 0x00000003;
```

16.1.5.9 Linked-List Programming Guidelines

- With the exception of the DMA4_CCRi[7] ENABLE bit and the DMA4_CDPi[7] PAUSE_LINK_LIST bit during a linked-list transfer (descriptor load phase or data load phase), avoid programming any register through the configuration port.
- Before enabling any linked-list transfer, ensure that all global registers and all registers in the descriptor are initialized. Some static channel registers (registers that are not updated by the descriptor to be loaded) must also be initialized correctly:
  – For type 2, the following registers must be initialized with consistent values:
    - All global registers
    - DMA4_CCRi
    - DMA4_CSDPi
    - DMA4_CLNK_CTRLi
  – For type 3, the following registers must be initialized with consistent values:
    - All global registers
    - DMA4_CCRi
    - DMA4_CSDPi
    - DMA4_CLNK_CTRLi
    - DMA4_CICRi
    - DMA4_CFNi
- In case of a linked list with descriptor types 2 and 3, the content of the DMA4_CCRi register must not change during super-block life.
- The PAUSE_LINK_LIST bit must not be set in the initialization phase.

16.1.6 DMA_SYSTEM Register Manual

16.1.6.1 DMA_SYSTEM Instance Summary

Table 16-19. DMA_SYSTEM Instance Summary

Module Name	Base Address	Size
DMA_SYSTEM	0x4A05 6000	4 KiB

16.1.6.2 DMA_SYSTEM Registers

16.1.6.2.1 DMA_SYSTEM Register Summary

Index *i* represents the logical channel number (where i = 0 to 31). The offset address for some registers is calculated from the channel *c* number. For example, the DMA4_CCR10 (channel 10) register has an offset address of 10 × 0x60 = 0x3C0, and thus a physical address of 0x4A05 6080 + 0x3C0 = 0x4A05 6440.

Index *j* represents the interrupt line number (where j = 0 to 3) The offset address for some registers is calculated from the channel *c* number. For example, the DMA4_IRQSTATUS_L3 (line 3) register has an offset address of 3 × 0x4 = 0xC, and thus a physical address of 0x4A05 6008 + 0xC = 0x4A05 6014.

Table 16-20. DMA_SYSTEM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DMA_SYSTEM Base Address
DMA4_REVISION	R	32	0x0000 0000	0x4A05 6000
DMA4_IRQSTATUS_Lj (2)	RW	32	0x0000 0008 + (0x4 * j)	0x4A05 6008 + (0x4 * j)
DMA4_IRQENABLE_Lj (2)	RW	32	0x0000 0018 + (0x4 * j)	0x4A05 6018 + (0x4 * j)
DMA4_SYSSTATUS	R	32	0x0000 0028	0x4A05 6028
DMA4_OCP_SYSCONFIG	RW	32	0x0000 002C	0x4A05 602C
DMA4_CAPS_0	RW	32	0x0000 0064	0x4A05 6064
DMA4_CAPS_2	R	32	0x0000 006C	0x4A05 606C
DMA4_CAPS_3	R	32	0x0000 0070	0x4A05 6070
DMA4_CAPS_4	RW	32	0x0000 0074	0x4A05 6074
DMA4_GCR	RW	32	0x0000 0078	0x4A05 6078
DMA4_CCRi (1)	RW	32	0x0000 0080 + (0x60 * i)	0x4A05 6080 + (0x60 * i)
DMA4_CLNK_CTRLi (1)	RW	32	0x0000 0084 + (0x60 * i)	0x4A05 6084 + (0x60 * i)
DMA4_CICRi (1)	RW	32	0x0000 0088 + (0x60 * i)	0x4A05 6088 + (0x60 * i)
DMA4_CSRi (1)	RW	32	0x0000 008C + (0x60 * i)	0x4A05 608C + (0x60 * i)
DMA4_CSDPi (1)	RW	32	0x0000 0090 + (0x60 * i)	0x4A05 6090 + (0x60 * i)
DMA4_CENi (1)	RW	32	0x0000 0094 + (0x60 * i)	0x4A05 6094 + (0x60 * i)
DMA4_CFNi (1)	RW	32	0x0000 0098 + (0x60 * i)	0x4A05 6098 + (0x60 * i)
DMA4_CSSAi (1)	RW	32	0x0000 009C + (0x60 * i)	0x4A05 609C + (0x60 * i)
DMA4_CDSAi (1)	RW	32	0x0000 00A0 + (0x60 * i)	0x4A05 60A0 + (0x60 * i)
DMA4_CSEIi (1)	RW	32	0x0000 00A4 + (0x60 * i)	0x4A05 60A4 + (0x60 * i)
DMA4_CSFIi (1)	RW	32	0x0000 00A8 + (0x60 * i)	0x4A05 60A8 + (0x60 * i)
DMA4_CDEIi (1)	RW	32	0x0000 00AC + (0x60 * i)	0x4A05 60AC + (0x60 * i)
DMA4_CDFIi (1)	RW	32	0x0000 00B0 + (0x60 * i)	0x4A05 60B0 + (0x60 * i)
DMA4_CSACi (1)	R	32	0x0000 00B4 + (0x60 * i)	0x4A05 60B4 + (0x60 * i)
DMA4_CDACi (1)	RW	32	0x0000 00B8 + (0x60 * i)	0x4A05 60B8 + (0x60 * i)
DMA4_CCENi (1)	RW	32	0x0000 00BC + (0x60 * i)	0x4A05 60BC + (0x60 * i)
DMA4_CCFNi (1)	RW	32	0x0000 00C0 + (0x60 * i)	0x4A05 60C0 + (0x60 * i)
DMA4_COLORi (1)	RW	32	0x0000 00C4 + (0x60 * i)	0x4A05 60C4 + (0x60 * i)
DMA4_CDPi (1)	RW	32	0x0000 00D0 + (0x60 * i)	0x4A05 60D0 + (0x60 * i)
DMA4_CNDPi (1)	RW	32	0x0000 00D4 + (0x60 * i)	0x4A05 60D4 + (0x60 * i)
DMA4_CCDNi (1)	RW	32	0x0000 00D8 + (0x60 * i)	0x4A05 60D8 + (0x60 * i)

(1) i = 0 to 31
(2) j = 0 to 3

16.1.6.2.2 DMA_SYSTEM Register Description

Note

Some registers have no reset value (marked with -) because of hardware implementation in memory. Software must ensure the correct programming of these registers, if needed.

Shadow registers are used to read run-time registers such as CCEN, CCFN, CDAC, and CSAC. Typically, when accessed in 8-bit or 16-bit access for two consecutive accesses, the value of the previous registers can change. A shadow register holds the entire value to let the next access recover the remaining 24 or 16 bits.

For non-32-bit transactions, start reading or writing from the LSByte first to enable the register update. There is no issue for 32-bit read-write transactions.

Table 16-21. DMA4_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A05 6000	**Instance**	DMA_SYSTEM
Description	This register contains the DMA revision code		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Reserved, Write 0's for future compatibility. Read returns 0	R	TI internal Data

Table 16-22. DMA4_IRQSTATUS_Lj

Address Offset	0x0000 0008 + (0x4 * j)	**Index**	j = 0 to 3
Physical Address	0x4A05 6008 + (0x4 * j)	**Instance**	DMA_SYSTEM
Description	The interrupt status register regroups all the status of the DMA_SYSTEM channels that can generate an interrupt over line Lj.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH_31_0_Lj																															

Bits	Field Name	Description	Type	Reset
31:0	CH_31_0_Lj	Channel 31 Interrupt on Lj: When an interrupt is seen on the line Lj the status of a interrupting channel i is read in the bit field i. Read 0x0: Channel Interrupt Lj false Write 0x0: Channel Interrupt Lj status bit unchanged Write 0x1: Channel Interrupt Lj status bit is reset Read 0x1: Channel Interrupt Lj true (pending)	RW W1toClr	0x0000 0000

Table 16-23. DMA4_IRQENABLE_Lj

Address Offset	0x0000 0018 + (0x4 * j)	**Index**	j = 0 to 3
Physical Address	0x4A05 6018 + (0x4 * j)	**Instance**	DMA_SYSTEM
Description	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on line Lj		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CH_31_0_Lj_EN

Bits	Field Name	Description	Type	Reset
31:0	CH_31_0_Lj_EN	Channel Interrupt on Lj mask/unmask : to Mask/Unmask a channel i interrupt on Lj the user writes 0/1 on the bit field i. 0x0: Channel Interrupt Lj is masked 0x1: Channel Interrupt Lj generates an interrupt when it occurs	RW	0x0000 0000

Table 16-24. DMA4_SYSSTATUS

Address Offset	0x0000 0028		
Physical Address	0x4A05 6028	**Instance**	DMA_SYSTEM
Description	The register provides status information about the module excluding the interrupt status information (see interrupt status register)		
Type	R		

31:1	0
RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved for module-specific status information	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is on-going Read 0x1: Reset completed	R	1

Table 16-25. DMA4_OCP_SYSCONFIG

Address Offset	0x0000 002C		
Physical Address	0x4A05 602C	**Instance**	DMA_SYSTEM
Description	DMA system configuration register		
Type	RW		

31:14	13:12	11:10	9:8	7:6	5	4:3	2	1	0
RESERVED	MIDLEMODE	RESERVED	CLOCKACTIVITY	RESERVED	EMUFREE	SIDLEMODE	RESERVED	RESERVED	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x00000

Bits	Field Name	Description	Type	Reset
13:12	MIDLEMODE	Read write power management, standby/wait control 0x0: Force-standby: MStandby is asserted only when all the DMA channels are disabled 0x1: No-Standby: MStandby is never asserted 0x2: Smart-Standby: MStandby is asserted if at least one of the following two conditions is satisfied: 1. All the channels are disabled, OR 2. There is no non-synchronized channel enabled AND [if hardware synchronized channel is enabled, then no DMA request input is asserted and no requests are pending to be serviced]. 0x3: Reserved	RW	0x0
11:10	RESERVED	Reserved for clocks activities extension	RW	0x0
9:8	CLOCKACTIVITY	Clocks activities during wake-up Bit 8: Interface clock 0x0: Interface clock can be switched-off Bit 9: Functional clock 0x0: Functional clock can be switched-off	R	0x0
7:6	RESERVED	Write 0's for future compatibility. Read returns 0	RW	0x0
5	EMUFREE	Enable sensitivity to MSuspend 0x0: DMA4 freezes its internal logic upon MSuspend assertion 0x1: DMA4 ignores the MSuspend input	RW	0
4:3	SIDLEMODE	Configuration port power management, Idle req/ack control 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Idle acknowledge is given by DMA4 if all of the conditions are true: 1. All the channels are disabled. 2. If hardware synchronized channel is enabled, then no DMA request input is asserted and no requests are pending to be serviced. 3. All transactions are completed on all the DMA ports. 4.No interrupts are pending to be serviced. 0x3: Reserved. Do not use	RW	0x0
2	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0
1	RESERVED	Reserved for non-GP devices	RW	0
0	AUTOIDLE	Internal interface clock gating strategy 0x0: Interface clock is free running 0x1: Automatic interface clock gating strategy is applied, based on the interface activity.	RW	0

Table 16-26. DMA4_CAPS_0

Address Offset	0x0000 0064		
Physical Address	0x4A05 6064	**Instance**	DMA_SYSTEM
Description	DMA Capabilities Register 0 LSW		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LINK_LIST_CPBLTY_TYPE4	LINK_LIST_CPBLTY_TYPE123	CONST_FILL_CPBLTY	TRANSPARENT_BLT_CPBLTY	RESERVED

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Write 0's for future compatibility. Read returns 0	RW	0x000
21	LINK_LIST_CPBLTY_TYPE4	Link List capability for type4 descriptor capability	R	0
20	LINK_LIST_CPBLTY_TYPE123	Link List capability for type123 descriptor capability	R	1
19	CONST_FILL_CPBLTY	Constant_Fill_Capability Read 0x0: No LCH supports constant fill copy Read 0x1: any LCH supports constant fill copy	R	1
18	TRANSPARENT_BLT_CPBLTY	Transparent_BLT_Capability Read 0x0: No LCH supports transparent BLT copy Read 0x1: any LCH supports transparent BLT copy	R	1
17:0	RESERVED	Write 0's for future compatibility. Read returns 0	RW	0x00000

Table 16-27. DMA4_CAPS_2

Address Offset	0x0000 006C		
Physical Address	0x4A05 606C	**Instance**	DMA_SYSTEM
Description	DMA Capabilities Register 2		
Type	R		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9	8	7	6	5	4	3	2	1	0
RESERVED	SEPARATE_SRC_AND_DST_INDEX_CPBLTY	DST_DOUBLE_INDEX_ADRS_CPBLTY	DST_SINGLE_INDEX_ADRS_CPBLTY	DST_POST_INCRMNT_ADRS_CPBLTY	DST_CONST_ADRS_CPBLTY	SRC_DOUBLE_INDEX_ADRS_CPBLTY	SRC_SINGLE_INDEX_ADRS_CPBLTY	SRC_POST_INCREMENT_ADRS_CPBLTY	SRC_CONST_ADRS_CPBLTY

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0's for future compatibility. Read returns 0	R	0x000000

Bits	Field Name	Description	Type	Reset
8	SEPARATE_SRC_AND_DST_INDEX_CPBLTY	Separate_source/destination_index_capability Read 0x0: Does not support separate src/dst index for 2D addressing Read 0x1: Supports separate src/dest index for 2D addressing	R	1
7	DST_DOUBLE_INDEX_ADRS_CPBLTY	Destination_double_index_address_capability Read 0x0: Does not support double index address mode on the destination port Read 0x1: Supports double index address mode on the destination port	R	1
6	DST_SINGLE_INDEX_ADRS_CPBLTY	Destination_single_index_address_capability Read 0x0: Does not support single index address mode on the destination port Read 0x1: Supports single index address mode on the destination port	R	1
5	DST_POST_INCRMNT_ADRS_CPBLTY	Destination_post_increment_address_capability Read 0x0: Does not supports post-increment address mode in the destination port Read 0x1: Supports post-increment address mode in the destination port	R	1
4	DST_CONST_ADRS_CPBLTY	Destination_constant_address_capability Read 0x0: Does not supports constant address mode in the destination port Read 0x1: Supports constant address mode in the destination port	R	1
3	SRC_DOUBLE_INDEX_ADRS_CPBLTY	Source_double_index_address_capability Read 0x0: Does not support double index address mode on the source port Read 0x1: Supports double index address mode on the source port	R	1
2	SRC_SINGLE_INDEX_ADRS_CPBLTY	Source_single_index_address_capability Read 0x0: Does not support single index address mode on the source port Read 0x1: Supports single index address mode in the source port	R	1
1	SRC_POST_INCREMENT_ADRS_CPBLTY	Source_post_increment_address_capability Read 0x0: Does not supports post-increment address mode in the source port Read 0x1: Supports post-increment address mode in the source port	R	1
0	SRC_CONST_ADRS_CPBLTY	Source_constant_address_capability Read 0x0: Does not supports constant address mode in the source port Read 0x1: Supports constant address mode in the source port	R	1

Table 16-28. DMA4_CAPS_3

Address Offset	0x0000 0070		
Physical Address	0x4A05 6070	**Instance**	DMA_SYSTEM
Description	DMA Capabilities Register 3		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	BLOCK_SYNCHR_CPBLTY	PKT_SYNCHR_CPBLTY	CHANNEL_CHANINIG_CPBLTY	CHANNEL_INTERLEAVE_CPBLTY	RESERVED	FRAME_SYNCHR_CPBLTY	ELMNT_SYNCHR_CPBLTY

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Read returns 0	R	0x000000
7	BLOCK_SYNCHR_CPBLTY	Block_synchronization_capability Read 0x0: Does not support synchronization transfer on block boundary Read 0x1: Supports synchronization transfer on block boundary	R	1
6	PKT_SYNCHR_CPBLTY	Packet_synchronization_capability Read 0x0: Does not support synchronization transfer on packet boundary Read 0x1: Supports synchronization transfer on packet boundary	R	1
5	CHANNEL_CHANINIG_CPBLTY	Channel_Chaninig_capability Read 0x0: Does not support Channel Chaninig capability Read 0x1: Supports Channel Chaninig capability	R	1
4	CHANNEL_INTERLEAVE_CPBLTY	Channel_interleave_capability Read 0x0: Does not support Channel interleave capability Read 0x1: Supports Channel_interleave capability	R	1
3:2	RESERVED		R	0x0
1	FRAME_SYNCHR_CPBLTY	Frame_synchronization_capability Read 0x0: Does not support synchronization transfer on Frame boundary Read 0x1: Supports synchronization transfer on Frame boundary	R	1
0	ELMNT_SYNCHR_CPBLTY	Element_synchronization_capability Read 0x0: Does not support synchronization transfer on Element boundary Read 0x1: Supports synchronization transfer on Element boundary	R	1

Table 16-29. DMA4_CAPS_4

Address Offset	0x0000 0074		
Physical Address	0x4A05 6074	**Instance**	DMA_SYSTEM
Description	DMA Capabilities Register 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	EOSB_INTERRUPT_CPBLTY	RESERVED	DRAIN_END_INTERRUPT_CPBLTY	MISALIGNED_ADRS_ERR_INTERRUPT_CPBLTY	SUPERVISOR_ERR_INTERRUPT_CPBLTY	RESERVED	TRANS_ERR_INTERRUPT_CPBLTY	PKT_INTERRUPT_CPBLTY	SYNC_STATUS_CPBLTY	BLOCK_INTERRUPT_CPBLTY	LAST_FRAME_INTERRUPT_CPBLTY	FRAME_INTERRUPT_CPBLTY	HALF_FRAME_INTERRUPT_CPBLTY	EVENT_DROP_INTERRUPT_CPBLTY	RESERVED

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00000
14	EOSB_INTERRUPT_CPBLTY	End of Super Block detection capability.	R	1
13	RESERVED	Reserved for non-GP devices	R	1
12	DRAIN_END_INTERRUPT_CPBLTY	Drain End detection capability.	R	1
11	MISALIGNED_ADRS_ERR_INTERRUPT_CPBLTY	Misaligned error detection capability.	R	1
10	SUPERVISOR_ERR_INTERRUPT_CPBLTY	Supervisor error detection capability.	R	1
9	RESERVED	Reserved for non-GP devices	R	1
8	TRANS_ERR_INTERRUPT_CPBLTY	Transaction error detection capability.	R	1
7	PKT_INTERRUPT_CPBLTY	End of Packet detection capability. Read 0x0: Does not support end of packet interrupt generation capability Read 0x1: Supports end of packet interrupt generation capability	R	1
6	SYNC_STATUS_CPBLTY	Sync_status_capability Read 0x0: Does not support synchronized transfer status bit generation Read 0x1: Supports synchronized transfer status bit generation	R	1
5	BLOCK_INTERRUPT_CPBLTY	End of block detection capability. Read 0x0: Does not support end of block interrupt generation capability Read 0x1: Supports end of block interrupt generation capability	R	1
4	LAST_FRAME_INTERRUPT_CPBLTY	Start of last frame detection capability. Read 0x0: Does not support last frame interrupt generation capability Read 0x1: Supports last frame interrupt generation capability	R	1

Bits	Field Name	Description	Type	Reset
3	FRAME_INTERRUPT_CPBLTY	End of frame detection capability. Read 0x0: Does not support end of frame interrupt generation capability Read 0x1: Supports end of frame interrupt generation capability	R	1
2	HALF_FRAME_INTERRUPT_CPBLTY	Detection capability of the half of frame end. Read 0x0: Does not support half of frame interrupt generation capability Read 0x1: Supports half of frame interrupt generation capability	R	1
1	EVENT_DROP_INTERRUPT_CPBLTY	Request collision detection capability. Read 0x0: Does not support event drop interrupt generation capability Read 0x1: Supports event drop interrupt generation capability	R	1
0	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0

Table 16-30. DMA4_GCR

Address Offset	0x0000 0078		
Physical Address	0x4A05 6078	**Instance**	DMA_SYSTEM
Description	FIFO sharing between high and low priority channel. The Maximum per channel FIFO depth is bounded by the low and high channel FIFO budget. The high respectively low priority channels maximum burst size must be less than the min (high respectively low priority channel FIFO budget, per channel maximum FIFO depth)		
Type	RW		

31:25	24	23:16	15:14	13:12	11:8	7:0
RESERVED	CHANNEL_ID_GATE	ARBITRATION_RATE	HI_LO_FIFO_BUDGET	HI_THREAD_RESERVED	RESERVED	MAX_CHANNEL_FIFO_DEPTH

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00
24	CHANNEL_ID_GATE	Gates the Channel ID bus monitoring on both Read and Write ports 0x0: Gates the Channel ID qualifiers on both Read and Write Ports 0x1: Does not gate the Channel ID qualifiers on both Read and Write Ports	RW	0x0
23:16	ARBITRATION_RATE	Arbitration switching rate between prioritized and regular channel queues	RW	0x01

Bits	Field Name	Description	Type	Reset
15:14	HI_LO_FIFO_BUDGET	Allow to have a separate Global FIFO budget for high and low priority channels. For Hi priority Channel: (Per_channel_Maximum FIFO depth + 1) x Number of active High priority Channel =< High Budget FIFO For Low priority channel: (Per_channel_Maximum FIFO depth + 1) x Number of active Low priority Channel =< Low Budget FIFO 0x0: no fixed budget for neither higher nor lower priority channel 0x1: 75% of FIFO for low priority and 25% for high priority channels 0x2: 25% of FIFO for low priority and 75% for high priority channels 0x3: 50% of FIFO for low priority and 50% for high priority channels	RW	0x0
13:12	HI_THREAD_RESERVED	Allow thread reservation for high priority channel on both read and write ports. 0x0: No ThreadID is reserved on the Read Port for high priority channels. No ThreadID is reserved on the Write Port for high priority channels. 0x1: Read Port ThreadID 0 is reserved for high priority channels. Write Port ThreadID 0 is reserved for high priority channels. 0x2: Read port ThreadID 0 and ThreadID 1 are reserved for high priority channels. Write Port ThreadID 0 is reserved for high priority channels. 0x3: Read PortThreadID 0, ThreadID 1 and ThreadID 2 are reserved for high priority channels. Write Port ThreadID 0 is reserved for high priority channels.	RW	0x0
11:8	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0
7:0	MAX_CHANNEL_FIFO_DEPTH	Maximum FIFO depth allocated to one logical channel. Maximum FIFO depth can not be 0x0. It should be at least 0x1 or greater. Note that If channel limit is less than destination burst size enough data will not be accumulated in the data FIFO and it will never be sent out on the WR port. The burst size should be less than the FIFO limit specified in this bit field.	RW	0x10

Table 16-31. DMA4_CCRi

Address Offset	0x0000 0080 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 6080 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	WRITE_PRIORITY	BUFFERING_DISABLE	SEL_SRC_DST_SYNC	PREFETCH	SUPERVISOR	RESERVED	SYNCHRO_CONTROL_UPPER	BS	TRANSPARENT_COPY_ENABLE	CONST_FILL_ENABLE	DST_AMODE	SRC_AMODE	RESERVED	WR_ACTIVE	RD_ACTIVE	SUSPEND_SENSITIVE	ENABLE	READ_PRIORITY	FS	SYNCHRO_CONTROL

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0
29:27	RESERVED	Reserved for non-GP devices	RW	0x0
26	WRITE_PRIORITY	Channel priority on the Write side 0x0: Channel has low priority on the write side during the arbitration process. 0x1: Channel has high priority on write sided during the arbitration process.	RW	0
25	BUFFERING_DISABLE	This bit allows to disable the default buffering functionality when transfer is source synchronized. 0x0: Buffering is enabled across element/packet when source is synchronized to element, packet, frame or blocks. 0x1: Buffering is disabled across element/packet when source is synchronized to element, packet, frame or blocks.	RW	-
24	SEL_SRC_DST_SYNC	Specifies that element, packet, frame or block transfer (depending on CCR.bs and CCR.fs) is triggered by the source or the destination on the DMA request 0x0: Transfer is triggered by the destination. If synch on packet the packet element number is specified in the CDFI register. 0x1: Transfer is triggered by the source. If synchronized on packet the packet element number is specified in the CSFI register.	RW	-
23	PREFETCH	Enables the prefetch mode 0x0: Prefetch mode is disabled. When Sel_Src_Dst_Sync=1 transfers are buffered and pipelined between DMA requests. 0x1: Prefetch mode is enabled. Prefetch mode is active only when destination is synchronized. It is software user responsibility not to have at the same time Prefetch=1 when Sel_Src_Dst_Sync=1. This mode is not supported.	RW	0
22	SUPERVISOR	Enables the supervisor mode 0x0: Supervisor mode is disabled. 0x1: Supervisor mode is enabled.	RW	0
21	RESERVED	Reserved for non-GP devices	RW	0
20:19	SYNCHRO_CONTROL_UPPER	Channel Synchronization control upper (used in conjunction with the 5 bits of synchro channel DMA4_CCRi[4:0]) Used in conjunction, as 2 MSB, with the 5 bits of the synchro channel bit field.	RW	0b00
18	BS	Block synchronization This bit used in conjunction with the fs to see how the DMA request is serviced in a synchronized transfer.	RW	-

Bits	Field Name	Description	Type	Reset
17	TRANSPARENT_COPY_ENABLE	Transparent copy enable 0x0: Transparent copy mode is disabled. 0x1: Transparent copy mode is enabled.	RW	-
16	CONST_FILL_ENABLE	Constant fill enable 0x0: Constant fill mode is disabled. 0x1: Constant fill mode is enabled.	RW	0
15:14	DST_AMODE	Selects the addressing mode on the Write Port of a channel. 0x0: Constant address mode 0x1: Post-incremented address mode 0x2: Single index address mode 0x3: Double index address mode	RW	0bxx
13:12	SRC_AMODE	Selects the addressing mode on the Read Port of a channel. 0x0: Constant address mode 0x1: Post-incremented address mode 0x2: Single index address mode 0x3: Double index address mode	RW	0bxx
11	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
10	WR_ACTIVE	Indicates if the channel write context is active or not Read 0x0: Channel is not active on the write port. Read 0x1: Channel is currently active on the write port.	R	0
9	RD_ACTIVE	Indicates if the channel read context is active or not Read 0x0: Channel is not active on the read port. Read 0x1: Channel is currently active on the read port.	R	0
8	SUSPEND_SENSITIVE	Logical channel suspend enable bit 0x0: The channel ignores the MSuspend even if EMUFree is set to 0. 0x1: If EMUFree is set to 0 and MSuspend comes in then all current OCP services (single transaction or burst transaction as specified in the corresponding CSDP register) have to be completed before stopping processing any more transactions.	RW	0
7	ENABLE	Logical channel enable. It is SW responsibility to clear the CSR register and the IRQSTATUS bit for the different interrupt lines before enabling the channel. 0x0: The logical channel is disabled. 0x1: The logical channel is enabled.	RW	0
6	READ_PRIORITY	Channel priority on the read side 0x0: Channel has low priority on the read side during the arbitration process. 0x1: Channel has high priority on read sided during the arbitration process.	RW	0
5	FS	Frame synchronization This bit used in conjunction with the BS to see how the DMA request is serviced in a synchronized transfer FS = 0 and BS = 0: An element is transferred once a DMA request is made. FS = 0 and BS = 1: An entire block is transferred once a DMA request is made. FS = 1 and BS = 0: An entire frame is transferred once a DMA request is made. FS = 1 and BS = 1: A packet is transferred once a DMA request is made. All these different transfers can be interleaved on the port with other DMA requests.	RW	-

Bits	Field Name	Description	Type	Reset
4:0	SYNCHRO_CONTROL	Channel synchronization control This bit field used in conjunction with the second_level_ synchro_control_upper (as 2 MSB) 0000000 : Is reserved for non synchronized LCH transfer xxxxxxx (from 1 to 127)There are 127 possible DMA request to assign to any LCH. **Note:** The channel synchronization control registers are 1-based. For example, to enable the S_DMA_1 request, DMA4_CCR[4:0] SYNCHRO_CONTROL must be set to 0x2 (DMA request number + 1).	RW	0b00000

Table 16-32. DMA4_CLNK_CTRLi

Address Offset	0x0000 0084 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 6084 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Link Control Register		
Type	RW		

31–16	15	14–5	4–0
RESERVED	ENABLE_LNK	RESERVED	NEXTLCH_ID

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15	ENABLE_LNK	Enables or disable the channel linking. 0x0: Channel linking mode is disabled When set on the fly to 0 the current channel will complete the transfer and stops the chain linking 0x1: Channel linking mode is enabled. The logical channel defined in the NextLCH_ID is enabled at the end of the current transfer	RW	0
14:5	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x000
4:0	NEXTLCH_ID	Defines the NextLCh_ID, which is used to build logical channel chaining queue.	RW	0bxxxxx

Table 16-33. DMA4_CICRi

Address Offset	0x0000 0088 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 6088 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Interrupt Control Register		
Type	RW		

31–15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	SUPER_BLOCK_IE	RESERVED	DRAIN_IE	MISALIGNED_ERR_IE	SUPERVISOR_ERR_IE	RESERVED	TRANS_ERR_IE	PKT_IE	RESERVED	BLOCK_IE	LAST_IE	FRAME_IE	HALF_IE	DROP_IE	RESERVED

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00000
14	SUPER_BLOCK_IE	Enables the end of super block interrupt	RW	-

Bits	Field Name	Description	Type	Reset
13	RESERVED	Reserved for non-GP devices	RW	1
12	DRAIN_IE	Enables the end of draining interrupt	RW	0
11	MISALIGNED_ERR_IE	Enables the address misaligned error event interrupt 0x0: Disables the misaligned address error event interrupt 0x1: Enables the misaligned address error event interrupt	RW	-
10	SUPERVISOR_ERR_IE	Enables the supervisor transaction error event interrupt 0x0: Disables the supervisor transaction error event interrupt 0x1: Enables the supervisor transaction error event interrupt	RW	1
9	RESERVED	Reserved for non-GP devices	RW	1
8	TRANS_ERR_IE	Enables the transaction error event interrupt 0x0: Disables the transaction error event interrupt 0x1: Enables the transaction error event interrupt	RW	-
7	PKT_IE	Enables the end of Packet interrupt 0x0: Disables the end of Packet transfer interrupt 0x1: Enables the end of Packet transfer interrupt	RW	-
6	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0
5	BLOCK_IE	Enables the end of block interrupt 0x0: Disables the end of block interrupt 0x1: Enables the end of block interrupt	RW	-
4	LAST_IE	Last frame interrupt enable (start of last frame) 0x0: Disables the last frame interrupt 0x1: Enables the last frame interrupt	RW	-
3	FRAME_IE	Frame interrupt enable (end of frame) 0x0: Disables the end of frame interrupt 0x1: Enables the end of frame interrupt	RW	-
2	HALF_IE	Enables or disables the half frame interrupt. 0x0: Disables the half frame interrupt 0x1: Enables the half frame interrupt	RW	-
1	DROP_IE	Synchronization event drop interrupt enable (request collision) 0x0: Disables the event drop interrupt 0x1: Enables the event drop interrupt	RW	0
0	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0

Table 16-34. DMA4_CSRi

Address Offset	0x0000 008C + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 608C + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RESERVED	SUPER_BLOCK	RESERVED	DRAIN_END	MISALIGNED_ADRS_ERR	SUPERVISOR_ERR	RESERVED	TRANS_ERR	PKT	SYNC	BLOCK	LAST	FRAME	HALF	DROP	RESERVED

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
16:15	RESERVED	Reserved for debug (Monitor descriptor/data load phase), Write 0's for future compatibility, Read returns 0	RW	0x0
14	SUPER_BLOCK	End of Super block event Read 0x0: The current Super block transfer has not been finished Write 0x0: Status bit unchanged Read 0x1: The current Super block has been transferred Write 0x1: Status bit is reset	RW W1toClr	0
13	RESERVED	Reserved for non-GP devices	RW	0
12	DRAIN_END	End of channel draining Read 0x0: No drain end in the current transfer Write 0x0: Status bit unchanged Read 0x1: The current channel draining is completed Write 0x1: Status bit is reset	RW W1toClr	0
11	MISALIGNED_ADRS_ERR	Misaligned address error event Read 0x0: No address error Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: An address error has been occurred	RW W1toClr	0
10	SUPERVISOR_ERR	Supervisor transaction error event Read 0x0: No supervisor transaction error Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: A supervisor transaction error has been occurred	RW W1toClr	0
9	RESERVED	Reserved for non-GP devices	RW	0
8	TRANS_ERR	Transaction error event Read 0x0: No transaction error Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: A transaction error has been occurred	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
7	PKT	End of Packet transfer Read 0x0: The current packet transfer has not been finished Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The current packet has been transferred	RW W1toClr	0
6	SYNC	Synchronization status of a channel. Read 0x0: Logical channel is not scheduled or servicing a non synchronized DMA request. Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: Logical channel is servicing a synchronized DMA request	RW W1toClr	0
5	BLOCK	End of block event Read 0x0: The current block transfer has not been finished Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The current block has been transferred	RW W1toClr	0
4	LAST	Last frame (start of last frame) Read 0x0: The start of the last frame to transfer is not reached Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The start of the last frame to transfer is reached	RW W1toClr	0
3	FRAME	End of frame event Read 0x0: The end of current transferred frame is not reached Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The end of current transferred frame is reached	RW W1toClr	0
2	HALF	Half of frame event. Read 0x0: The half of current transferred frame is not reached Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The half of current transferred frame is reached	RW W1toClr	0
1	DROP	Synchronization event drop occured during the transfer Read 0x0: No synchronization collision Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: A synchronization collision has been occurred	RW W1toClr	0
0	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0

Table 16-35. DMA4_CSDPi

Address Offset	0x0000 0090 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 6090 + (0x60 * i)	**Instance**	DMA_SYSTEM

Table 16-35. DMA4_CSDPi (continued)

Description	Channel Source Destination Parameters
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td>SRC_ENDIAN</td><td>SRC_ENDIAN_LOCK</td><td>DST_ENDIAN</td><td>DST_ENDIAN_LOCK</td><td colspan="2">WRITE_MODE</td><td colspan="2">DST_BURST_EN</td><td>DST_PACKED</td><td colspan="4">RESERVED</td><td colspan="2">SRC_BURST_EN</td><td>SRC_PACKED</td><td colspan="4">RESERVED</td><td colspan="2">DATA_TYPE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x000
21	SRC_ENDIAN	Channel source endianness control 0x0: Source has Little Endian type 0x1: Source has Big Endian type	RW	-
20	SRC_ENDIAN_LOCK	Endianness Lock 0x0: Endianness adapt 0x1: Endianness lock	RW	-
19	DST_ENDIAN	Channel Destination endianness control 0x0: Destination has Little Endian type 0x1: Destination has Big Endian type	RW	-
18	DST_ENDIAN_LOCK	Endianness Lock 0x0: Endianness adapt 0x1: Endianness lock	RW	-
17:16	WRITE_MODE	Used to enable writing mode without posting or with posting 0x0: Write None Posted (WRNP) 0x1: Write (Posted) 0x2: All transaction are mapped on the Write command as posted except for the last transaction in the transfer mapped on a Write None Posted 0x3: Undefined	RW	0bxx
15:14	DST_BURST_EN	Used to enable bursting on the Write Port. Smaller burst size than the programmed burst size is also allowed 0x0: single access 0x1: 16 bytes or 4x32-bit / 2x64-bit burst access 0x2: 32 bytes or 8x32-bit / 4x64-bit burst access 0x3: 64 bytes or 16x32-bit / 8x64-bit burst access	RW	0b00
13	DST_PACKED	Destination receives packed data. 0x0: The destination target is non packed 0x1: The destination target is packed	RW	-
12:9	RESERVED	Write the reset value. Read returns reset value	RW	0x-
8:7	SRC_BURST_EN	Used to enable bursting on the Read Port. Smaller burst size than the programmed burst size is also allowed 0x0: single access 0x1: 16 bytes or 4x32-bit / 2x64-bit burst access 0x2: 32 bytes or 8x32-bit / 4x64-bit burst access 0x3: 64 bytes or 16x32-bit / 8x64-bit burst access	RW	0bxx

Bits	Field Name	Description	Type	Reset
6	SRC_PACKED	Source provides packed data. 0x0: The source target is non packed 0x1: The source target is packed	RW	-
5:2	RESERVED	Write the reset value. Read returns reset value	RW	0x-
1:0	DATA_TYPE	Defines the type of the data moved in the channel. 0x0: 8 bits scalar 0x1: 16 bits scalar 0x2: 32 bits scalar 0x3: Reserved	RW	0bxx

Table 16-36. DMA4_CENi

Address Offset	0x0000 0094 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 6094 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Element Number		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">CHANNEL_ELMNT_NBR</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00
23:0	CHANNEL_ELMNT_NBR	Number of elements within a frame (unsigned) to transfer	RW	0x------

Table 16-37. DMA4_CFNi

Address Offset	0x0000 0098 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 6098 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Frame Number		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">CHANNEL_FRAME_NBR</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CHANNEL_FRAME_NBR	Number of frames within the block to be transferred (unsigned)	RW	0x----

Table 16-38. DMA4_CSSAi

Address Offset	0x0000 009C + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 609C + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Source Start Address		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">SRC_START_ADRS</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	SRC_START_ADRS	32 bits of the source start address	RW	0x--------

Table 16-39. DMA4_CDSAi

Address Offset	0x0000 00A0 + (0x60 * i)	**index:**	i = 0 to 31

Table 16-39. DMA4_CDSAi (continued)

Physical Address	0x4A05 60A0 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Destination Start Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST_START_ADRS																															

Bits	Field Name	Description	Type	Reset
31:0	DST_START_ADRS	32 bits of the destination start address	RW	0x--------

Table 16-40. DMA4_CSEIi

Address Offset	0x0000 00A4 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60A4 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Source Element Index (Signed)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CHANNEL_SRC_ELMNT_INDEX															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CHANNEL_SRC_ELMNT_INDEX	Channel source element index	RW	0x----

Table 16-41. DMA4_CSFIi

Address Offset	0x0000 00A8 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60A8 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Source Frame Index (Signed) or 16-bit Packet size		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH_SRC_FRM_INDEX_OR_16BIT_PKT_ELNT_NBR																															

Bits	Field Name	Description	Type	Reset
31:0	CH_SRC_FRM_INDEX_OR_16BIT_PKT_ELNT_NBR	Channel source frame index value if source address is in double index mode. Or if fs=bs=1 and DMA_CCR[SEL_SRC_DST_SYNC] = 1; the bit field [15:0] gives the number of element in packet. The field [31:16] is unused for the packet size.	RW	0x--------

Table 16-42. DMA4_CDEIi

Address Offset	0x0000 00AC + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60AC + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Destination Element Index (Signed)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CHANNEL_DST_ELMNT_INDEX															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CHANNEL_DST_ELMNT_INDEX	Channel destination element index	RW	0x----

Table 16-43. DMA4_CDFIi

Address Offset	0x0000 00B0 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60B0 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Destination Frame Index (Signed) or 16-bit Packet size		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH_DST_FRM_IDX_OR_16BIT_PKT_ELNT_NBR																															

Bits	Field Name	Description	Type	Reset
31:0	CH_DST_FRM_IDX_OR_16BIT_PKT_ELNT_NBR	Channel destination frame index value if destination address is in double index mode. Or if fs=bs=1 and DMA_CCR[SEL_SRC_DST_SYNC]=0; the bit field [15:0] gives the number of element in packet. The field [31:16] is unused for the packet size..	RW	0x--------

Table 16-44. DMA4_CSACi

Address Offset	0x0000 00B4 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60B4 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Source Address Value. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_ELMNT_ADRS																															

Bits	Field Name	Description	Type	Reset
31:0	SRC_ELMNT_ADRS	Current source address counter value	R	0x--------

Table 16-45. DMA4_CDACi

Address Offset	0x0000 00B8 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60B8 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Destination Address Value. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST_ELMNT_ADRS																															

Bits	Field Name	Description	Type	Reset
31:0	DST_ELMNT_ADRS	Current destination address counter value	RW	0x--------

Table 16-46. DMA4_CCENi

Address Offset	0x0000 00BC + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60BC + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Current Transferred Element Number in the current frame. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CURRENT_ELMNT_NBR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00

Bits	Field Name	Description	Type	Reset
23:0	CURRENT_ELMNT_NBR	Channel current transferred element number in the current frame	RW	0x------

Table 16-47. DMA4_CCFNi

Address Offset	0x0000 00C0 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60C0 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel Current Transferred Frame Number in the current transfer. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
Type	RW		

31-16	15-0
RESERVED	CURRENT_FRAME_NBR

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CURRENT_FRAME_NBR	Channel current transferred frame number in the current transfer	RW	0x----

Table 16-48. DMA4_COLORi

Address Offset	0x0000 00C4 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60C4 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	Channel DMA COLOR KEY /SOLID COLOR		
Type	RW		

31-24	23-0
RESERVED	CH_BLT_FRGRND_COLOR_OR_SOLID_COLOR_PTRN

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x–
23:0	CH_BLT_FRGRND_COLOR_OR_SOLID_COLOR_PTRN	Color key or solid color pattern: The pattern is replicated according to the data type. If the data-type is 8-bit the pattern is replicated 4 times to fill the register in order to enhance processing when data is packed at the graphic module input. The same reasoning for 16-bit data-type.	RW	0x------

Table 16-49. DMA4_CDPi

Address Offset	0x0000 00D0 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60D0 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	This register controls the various parameters of the link list mechanism		
Type	RW		

31-11	10	9-8	7	6-4	3-2	1-0
RESERVED	FAST	TRANSFER_MODE	PAUSE_LINK_LIST	NEXT_DESCRIPTOR_TYPE	SRC_VALID	DEST_VALID

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x00000

Bits	Field Name	Description	Type	Reset
10	FAST	Sets the fast-start mode for linked list descriptor types 1, 2 and 3 0x0: No fast-start mode 0x1: Fast-start mode is enabled.	RW	0x0
9:8	TRANSFER_MODE	Enable linked-list transfer mode 0x0: Normal transfer mode is used. 0x1: Linked-list channel mode for type 1, 2, or 3 descriptor is used. 0x2: Undefined 0x3: Undefined	RW	0x0
7	PAUSE_LINK_LIST	Suspend the linked-list transfer at completion of the current block transfer. 0x0: Linked list is active. 0x1: Linked list is suspended at the boundary of next descriptor loading.	RW	0x0
6:4	NEXT_DESCRIPTOR_TYPE	Next Descriptor Type 0x0: Undefined 0x1: Next descriptor is of type 1. 0x2: Next descriptor is of type 2. 0x3: Next descriptor is of type 3. 0x4: Undefined 0x5: Undefined 0x6: Undefined 0x7: Undefined	RW	0x-
3:2	SRC_VALID	Source address valid 0x0: The source address is not present in the next descriptor and continuous incrementing is enabled. 0x1: The source address must be reloaded in the next descriptor transfer. 0x2: The source start address is not present in the next descriptor. But will reload the one from configuration memory which belongs to the previous descriptor. 0x3: Undefined addressing mode	RW	0x-
1:0	DEST_VALID	Destination address valid 0x0: The destination address is not present in the next descriptor and continuous incrementing is enabled. 0x1: The destination address must be reloaded in the next descriptor transfer. 0x2: The destination start address is not present in the next descriptor. But will reload the one from configuration memory which belongs to the previous descriptor. 0x3: Undefined addressing mode	RW	0x-

Table 16-50. DMA4_CNDPi

Address Offset	0x0000 00D4 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60D4 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description	This register contains the Next descriptor Address Pointer for the link list Mechanism		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NEXT_DESCRIPTOR_POINTER	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	NEXT_DESCRIPTOR_POINTER	This register contains the Next descriptor Address Pointer for the link list Mechanism	RW	0bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1:0	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x0

Table 16-51. DMA4_CCDNi

Address Offset	0x0000 00D8 + (0x60 * i)	**index:**	i = 0 to 31
Physical Address	0x4A05 60D8 + (0x60 * i)	**Instance**	DMA_SYSTEM
Description			
Type	RW		

31–16	15–0
RESERVED	CURRENT_DESCRIPTOR_NBR

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x0000
15:0	CURRENT_DESCRIPTOR_NBR	This register when read contains the current active descriptor number in the link list. This register is Read/write to allow user initialization.	RW	0x----

16.2 Enhanced DMA

This chapter describes the Enhanced Direct Memory Access (EDMA) controller.

16.2.1 EDMA Module Overview

The enhanced direct memory access module, also called EDMA, performs high-performance data transfers between two slave points, memories and peripheral devices without microprocessor unit (MPU) or digital signal processor (DSP) support during transfer. EDMA transfer is programmed through a logical EDMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

The EDMA can also perform transfers between external memories and between Device subsystems internal memories, with some performance loss caused by resource sharing between the read and write ports.

EDMA controller is based on two major principal blocks:
- EDMA third-party channel controller (EDMA_TPCC)
- EDMA third-party transfer controller (EDMA_TPTC)

The **TPCC** is a high flexible Channel Controller. It serves as an user interface and an event interface for the EDMA controller. The EDMA_TPCC serves to prioritize incoming software requests or events from peripherals and submits transfer requests (TRs) to the transfer controller.

The **TPTC** performs read and write transfers by EDMA ports to the slave peripherals as programmed in the "Active" and "Pending" set of the registers. The transfer controllers are responsible for data movement and issue read/write commands to the source and destination addresses that are programmed for a given transfer in the EDMA_TPCC.

The SoC integrates the following EDMA instances:

- One system-level EDMA
- One DSP internal EDMA (per DSP)
- One EVE internal EDMA (per EVE)

Each of these EDMA modules consists of:

- One TPCC instance
- Two TPTC instances

Note

All EDMA modules in the SoC are functionally identical. Note that some of the configuration parameters may be different for the various EDMA instances (see Section 16.2.1.3, *EDMA Controllers Configuration*).

This chapter is mostly focused on describing the system-level EDMA module (in terms of configuration and integration in the SoC). For details on DSPx_EDMA / EVEx_EDMA integration, see their respective chapters.

Figure 16-12 shows an overview of the EDMA module.

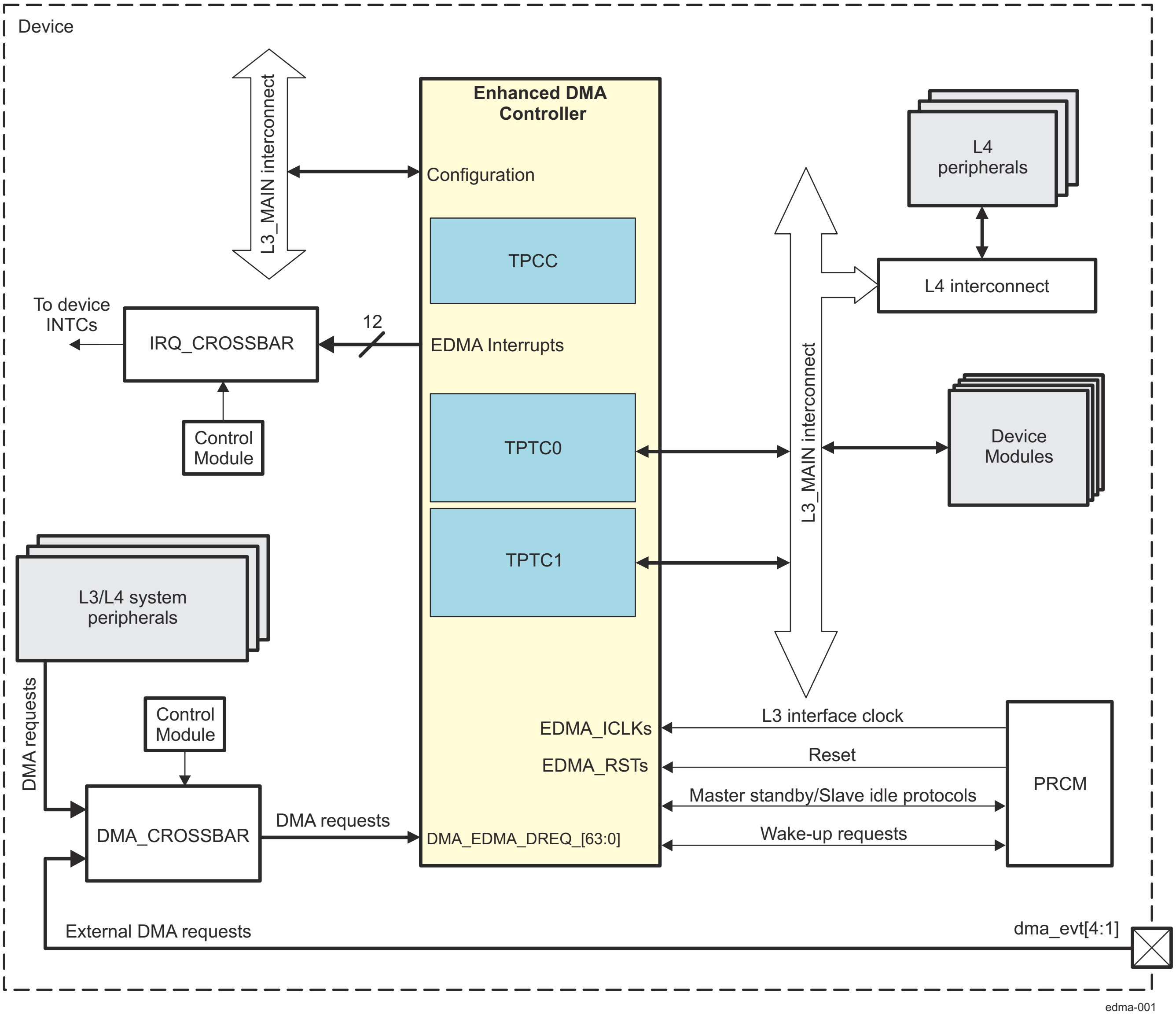


Figure 16-12. EDMA module Overview

The device CPUs can configure the EDMA controller blocks through the L3_MAIN interconnect.

16.2.1.1 EDMA Features

The **EDMA_TPCC** channel controller has following features:

- Fully orthogonal transfer description:
  - Three transfer dimensions.
  - A-synchronized transfers: one-dimension serviced per event.
  - AB-synchronized transfers: two-dimensions serviced per event.
  - Independent indexes on source and destination.
  - Chaining feature allows a 3-D transfer based on a single event.
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes.
  - Linking mechanism allows automatic PaRAM set update.
  - Chaining allows multiple transfers to execute with one event.
- Interrupt generation for the following:

  - Transfer completion.
  - Error conditions.
- Debug visibility:
  - Queue water marking/threshold.
  - Error and status recording to facilitate debug.
- 64 DMA request channels:
  - Event synchronization.
  - Manual synchronization (CPU(s) write to event set registers EDMA_TPCC_ESR and EDMA_TPCC_ESRH).
  - Chain synchronization (completion of one transfer triggers another transfer).
- Eight QDMA channels:
  - QDMA channels trigger automatically upon writing to a parameter RAM (PaRAM) set entry.
  - Support for programmable QDMA channel to PaRAM mapping.
- 512 PaRAM sets:
  - Each PaRAM set can be used for a DMA channel, QDMA channel, or link set.
- Two transfer controllers/event queues.
- 16 event entries per event queue.
- Memory protection support:
  - Proxy memory protection for TR submission.
  - Active memory protection for accesses to PaRAM and registers.

The **EDMA_TPTC** transfer controller has the following features:
- Two transfer controllers (TC).
- 128-bit wide read and write ports per TC.
- Up to four in-flight transfer requests (TRs).
- Programmable priority level.
- Supports two-dimensional transfers with independent indexes on source and destination (EDMA_TPCC manages the 3rd dimension).
- Support for increment or constant addressing mode transfers.
- Interrupt and error support.
- Memory-Mapped Register (MMR) bit fields are fixed position in 32-bit MMR regardless of endianness.

16.2.1.2

EDMA controller uses the shared MMU1 module for transfering to and from DSP module. This provides several benefits including:
- Protection of Host CPU memory regions from accidental corruption by EDMA TPTCs.
- Direct allocation of buffers in user space without the need for translation between CPU and DSP applications utilizing EDMA TPTCs.

Accesses by the EDMA TPTCs (both TPTC0 and TPTC1) may optionally be routed through the MMU1.

The TPTC0 and TPTC1 routing allows EDMA transfer controller to be used to perform transfers using only the virtual addresses of the associated buffers.

For more information about MMU1 module refer to Chapter 20 *Memory Management Units*.

16.2.1.3 EDMA Controllers Configuration

Table 16-52 summarizes the configuration for each of the EDMA channel controllers present on the SoC.

Table 16-52. EDMA Channel Controllers Configuration

Parameter	SYS_EDMA CC Configuration	DSPx_EDMA CC Configuration	EVEx_EDMA CC Configuration
Number of DMA channels (NUM_DMACH)	64	64	16
Number of QDMA channels (NUM_QDMACH)	8	8	8
Number of interrupt channels (NUM_INTCH)	64	64	16

Table 16-52. EDMA Channel Controllers Configuration (continued)

Parameter	SYS_EDMA CC Configuration	DSPx_EDMA CC Configuration	EVEx_EDMA CC Configuration
Number of PaRAM set entries (NUM_PARAMENTRY)	512	128	128
Number of event queues (NUM_EVQUE)	2	2	2
Number of transfer controllers (NUM_TC)	2	2	2
Memory protection existence (MPEXIST)	Yes	Yes	Yes
Number of memory protection and shadow regions (NUM_REGIONS)	8	8	8
Channel mapping existence (CHMAPEXIST)	Yes	Yes	Yes

Table 16-53 summarizes the configuration of each of the EDMA transfer controllers present on the SoC.

Table 16-53. EDMA Transfer Controllers Configuration

Parameter	SYS_EDMA TC0 / TC1 Configuration	DSPx_EDMA TC0 / TC1 Configuration	EVEx_EDMA TC0 / TC1 Configuration
Data FIFO size (FIFOSIZE)	1024 bytes	2048 bytes	2048 bytes
Bus width (BUSBYTE)	16 bytes	16 bytes	16 bytes
Number of destination FIFO register sets (DSTREGDEPTH)	4 entries	4 entries	4 entries
Default burst size (DBS)	Defined by CTRL_CORE_CONTROL_IO_1 register	Defined by DSP_SYS_BUS_CONFIG register	Defined by EVE_BUS_CONFIG register

16.2.2 EDMA Controller Environment

The EDMA controller supports external DMA requests through the dma_evt[4:1] pins (see Table 16-54). A logical channel can be configured to respond to an external synchronization request.

Table 16-54. External EDMA Request Signals

Pin Name	DMA_CROSSBAR Input	Signal Name	I/O(1)	Description	Module Reset Value
dma_evt1	DMA_CROSSBAR_2	EXT_SYS_DREQ_0	I	External DMA request 0 (system expansion)	Z
dma_evt2	DMA_CROSSBAR_3	EXT_SYS_DREQ_1	I	External DMA request 1 (system expansion)	Z
dma_evt3	DMA_CROSSBAR_167	EXT_SYS_DREQ_2	I	External DMA request 2 (system expansion)	Z
dma_evt4	DMA_CROSSBAR_168	EXT_SYS_DREQ_3	I	External DMA request 3 (system expansion)	Z

(1) I = Input, O = Output

Figure 16-13 shows an example of how to use the external hardware DMA request pins in the EDMA environment.

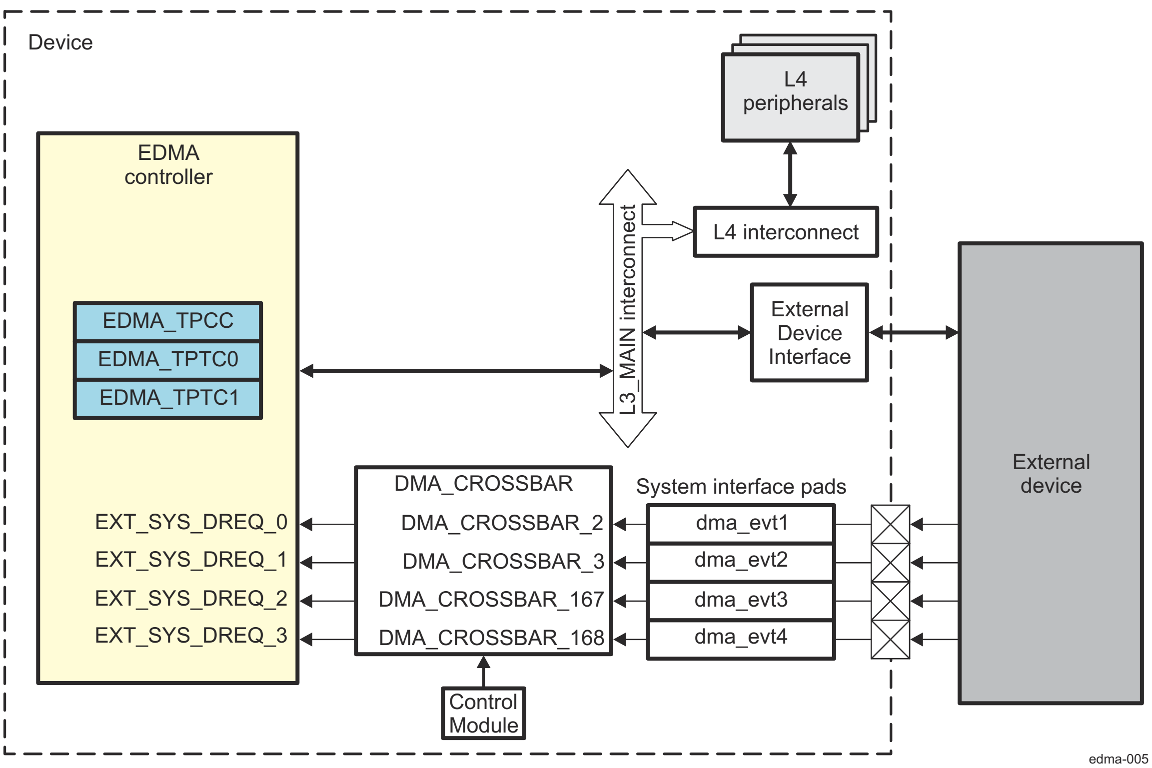


Figure 16-13. Example of External DMA Requests Use

An external device can use the external DMA request pins to start a logical channel transfer. The transfer can be a memory-to-memory transfer in which the source memory is in the external device.

By default, the external DMA request signals are not available on external pins after a cold reset. For more information about multiplexing out the two signal lines to pins, refer to *Control Module*.

All 64 DMA request lines are transition sensitive.

16.2.3 EDMA Controller Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 16-14 shows the EDMA controller integration.

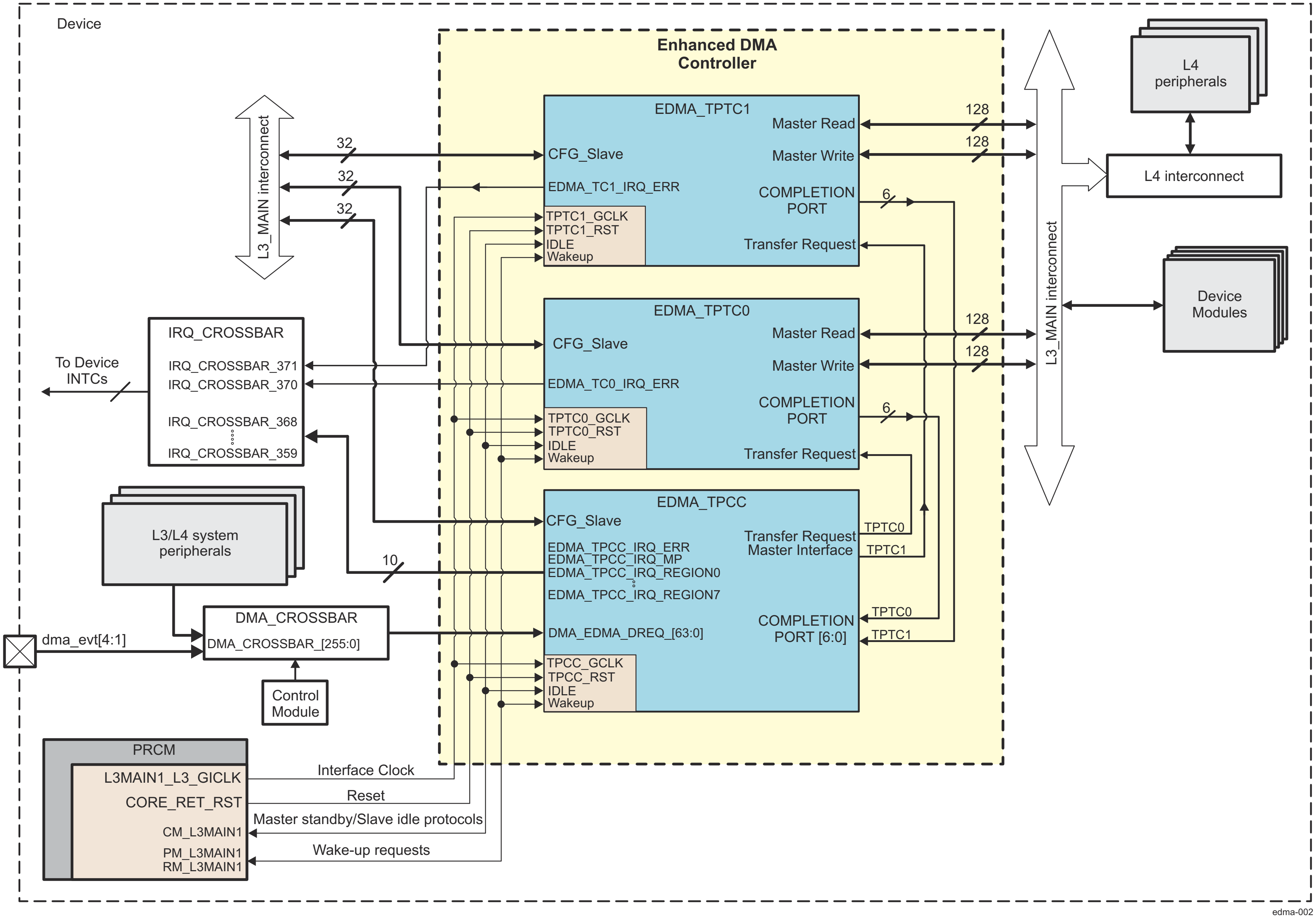


Figure 16-14. EDMA Controller Integration

Table 16-55 through Table 16-57 summarize the integration of the module in the device.

Table 16-55. EDMA Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
EDMA_TPCC	PD_COREAON	L3_MAIN
EDMA_TPTC0		
EDMA_TPTC1		

Table 16-56. EDMA Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EDMA_TPCC	EDMA_TPCC_GCLK	L3MAIN1_L3_GICLK	PRCM	Interface clock. It supports the configuration port. For information about PRCM clock gating and management, see *Power, Reset, and Clock Management*.
EDMA_TPTC0	EDMA_TPTC0_GCLK			
EDMA_TPTC1	EDMA_TPTC1_GCLK			
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EDMA_TPCC	EDMA_TPCC_RST	CORE_RET_RST	PRCM	Hardware retention reset. It initializes all internal logic of the EDMA contrller modules, all global registers, and some of the per-channel registers, implemented in flip-flops. However, all remaining per-channel registers are memory-based, and, therefore, are not reset (have undefined values). Thus, when programming a channel for the first time, all bits that have undefined reset values must be configured before enabling the channel. For information about PRCM reset sources and distribution, see *Power, Reset, and Clock Management*.
EDMA_TPTC0	EDMA_TPTC0_RST			
EDMA_TPTC1	EDMA_TPTC1_RST			

Table 16-57. EDMA Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR INPUT**	**Default mapping**	**Description**
EDMA_TPCC	EDMA_TPCC_IRQ_ERR	IRQ_CROSSBAR_359	-	TPCC error interrupt. This IRQ source signal is not mapped by default to any device INTC. For more information about INTC refer to *Interrupt Controllers*.
	EDMA_TPCC_IRQ_MP	IRQ_CROSSBAR_360	-	TPCC memory protection interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION0	IRQ_CROSSBAR_361	-	TPCC Region 0 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 16-57. EDMA Hardware Requests (continued)

	EDMA_TPCC_IRQ_REGION1	IRQ_CROSSBAR_362	-	TPCC Region 1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION2	IRQ_CROSSBAR_363	-	TPCC Region 2 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION3	IRQ_CROSSBAR_364	-	TPCC Region 3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION4	IRQ_CROSSBAR_365	-	TPCC Region 4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION5	IRQ_CROSSBAR_366	-	TPCC Region 5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION6	IRQ_CROSSBAR_367	-	TPCC Region 6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION7	IRQ_CROSSBAR_368	-	TPCC Region 7 interrupt. This IRQ source signal is not mapped by default to any device INTC.
EDMA_TPTC0	EDMA_TC0_IRQ_ERR	IRQ_CROSSBAR_370	-	TPTC0 error interrupt. This IRQ source signal is not mapped by default to any device INTC.
EDMA_TPTC1	EDMA_TC1_IRQ_ERR	IRQ_CROSSBAR_371	-	TPTC1 error interrupt. This IRQ source signal is not mapped by default to any device INTC.

Note

The “**Default Mapping**” column in Table 16-57 *EDMA Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR or DMA_CROSSBAR modules.
For more information about the IRQ_CROSSBAR and DMA_CROSSBAR modules, see sections: *IRQ_CROSSBAR Module Functional Description* and *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For a description of the interrupt source, see *EDMA interrupts*.

16.2.3.1 EDMA Requests to the EDMA Controller

Table 16-58 lists the default DMA sources for the EDMA controller. In addition, the EDMA inputs (DMA_EDMA_DREQ_[63:0]) can alternatively be sourced through the associated DMA_CROSSBAR from one of the 256 multiplexed device DMA sources listed in Table 16-6. The CTRL_CORE_DMA_EDMA_DREQ_y_z registers (where *y* and *z* are indexes of EDMA input lines) in the Control Module are used to select between the default DMA sources and the multiplexed DMA sources.

Table 16-58. EDMA Default Request Mapping

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_EDMA_DREQ_0	1	CTRL_CORE_DMA_EDMA_DREQ_0_1[7:0]	1	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_1	2	CTRL_CORE_DMA_EDMA_DREQ_0_1[23:16]	2	EXT_SYS_DREQ_0	External DMA request 0 (system expansion)
DMA_EDMA_DREQ_2	3	CTRL_CORE_DMA_EDMA_DREQ_2_3[7:0]	3	EXT_SYS_DREQ_1	External DMA request 1 (system expansion)
DMA_EDMA_DREQ_3	4	CTRL_CORE_DMA_EDMA_DREQ_2_3[23:16]	4	GPMC_DREQ	GPMC data transmit request from prefetch engine
DMA_EDMA_DREQ_4	5	CTRL_CORE_DMA_EDMA_DREQ_4_5[7:0]	5	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_5	6	CTRL_CORE_DMA_EDMA_DREQ_4_5[23:16]	6	DISPC_DREQ	Frame update request
DMA_EDMA_DREQ_6	7	CTRL_CORE_DMA_EDMA_DREQ_6_7[7:0]	7	CT_TBR_DREQ	DEBUG subsystem CT_TBR request
DMA_EDMA_DREQ_7	8	CTRL_CORE_DMA_EDMA_DREQ_6_7[23:16]	8	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_8	9	CTRL_CORE_DMA_EDMA_DREQ_8_9[7:0]	9	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_9	10	CTRL_CORE_DMA_EDMA_DREQ_8_9[23:16]	10	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_10	11	CTRL_CORE_DMA_EDMA_DREQ_10_11[7:0]	11	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_11	12	CTRL_CORE_DMA_EDMA_DREQ_10_11[23:16]	12	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_12	13	CTRL_CORE_DMA_EDMA_DREQ_12_13[7:0]	13	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_13	14	CTRL_CORE_DMA_EDMA_DREQ_12_13[23:16]	14	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_14	15	CTRL_CORE_DMA_EDMA_DREQ_14_15[7:0]	15	MCSPI3_DREQ_TX0	McSPI3 transmit request channel 0
DMA_EDMA_DREQ_15	16	CTRL_CORE_DMA_EDMA_DREQ_14_15[23:16]	16	MCSPI3_DREQ_RX0	McSPI3 receive request channel 0
DMA_EDMA_DREQ_16	17	CTRL_CORE_DMA_EDMA_DREQ_16_17[7:0]	17	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_17	18	CTRL_CORE_DMA_EDMA_DREQ_16_17[23:16]	18	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_18	19	CTRL_CORE_DMA_EDMA_DREQ_18_19[7:0]	19	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_19	20	CTRL_CORE_DMA_EDMA_DREQ_18_19[23:16]	20	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_20	21	CTRL_CORE_DMA_EDMA_DREQ_20_21[7:0]	21	Reserved	Reserved by default but can be remapped to a valid DMA source

Table 16-58. EDMA Default Request Mapping (continued)

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_EDMA_DREQ_21	22	CTRL_CORE_DMA_EDMA_DREQ_20_21[23:16]	22	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_22	23	CTRL_CORE_DMA_EDMA_DREQ_22_23[7:0]	23	MCSPI3_DREQ_TX1	McSPI3 transmit request channel 1
DMA_EDMA_DREQ_23	24	CTRL_CORE_DMA_EDMA_DREQ_22_23[23:16]	24	MCSPI3_DREQ_RX1	McSPI3 receive request channel 1
DMA_EDMA_DREQ_24	25	CTRL_CORE_DMA_EDMA_DREQ_24_25[7:0]	25	I2C3_DREQ_TX	I2C3 transmit request
DMA_EDMA_DREQ_25	26	CTRL_CORE_DMA_EDMA_DREQ_24_25[23:16]	26	I2C3_DREQ_RX	I2C3 receive request
DMA_EDMA_DREQ_26	27	CTRL_CORE_DMA_EDMA_DREQ_26_27[7:0]	27	I2C1_DREQ_TX	I2C1 transmit request
DMA_EDMA_DREQ_27	28	CTRL_CORE_DMA_EDMA_DREQ_26_27[23:16]	28	I2C1_DREQ_RX	I2C1 receive request
DMA_EDMA_DREQ_28	29	CTRL_CORE_DMA_EDMA_DREQ_28_29[7:0]	29	I2C2_DREQ_TX	I2C2 transmit request
DMA_EDMA_DREQ_29	30	CTRL_CORE_DMA_EDMA_DREQ_28_29[23:16]	30	I2C2_DREQ_RX	I2C2 receive request
DMA_EDMA_DREQ_30	31	CTRL_CORE_DMA_EDMA_DREQ_30_31[7:0]	31	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_31	32	CTRL_CORE_DMA_EDMA_DREQ_30_31[23:16]	32	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_32	33	CTRL_CORE_DMA_EDMA_DREQ_32_33[7:0]	33	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_33	34	CTRL_CORE_DMA_EDMA_DREQ_32_33[23:16]	34	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_34	35	CTRL_CORE_DMA_EDMA_DREQ_34_35[7:0]	35	MCSPI1_DREQ_TX0	McSPI1 transmit request channel 0
DMA_EDMA_DREQ_35	36	CTRL_CORE_DMA_EDMA_DREQ_34_35[23:16]	36	MCSPI1_DREQ_RX0	McSPI1 receive request channel 0
DMA_EDMA_DREQ_36	37	CTRL_CORE_DMA_EDMA_DREQ_36_37[7:0]	37	MCSPI1_DREQ_TX1	McSPI1 transmit request channel 1
DMA_EDMA_DREQ_37	38	CTRL_CORE_DMA_EDMA_DREQ_36_37[23:16]	38	MCSPI1_DREQ_RX1	McSPI1 receive request channel 1
DMA_EDMA_DREQ_38	39	CTRL_CORE_DMA_EDMA_DREQ_38_39[7:0]	39	MCSPI1_DREQ_TX2	McSPI1 transmit request channel 2
DMA_EDMA_DREQ_39	40	CTRL_CORE_DMA_EDMA_DREQ_38_39[23:16]	40	MCSPI1_DREQ_RX2	McSPI1 receive request channel 2
DMA_EDMA_DREQ_40	41	CTRL_CORE_DMA_EDMA_DREQ_40_41[7:0]	41	MCSPI1_DREQ_TX3	McSPI1 transmit request channel 3
DMA_EDMA_DREQ_41	42	CTRL_CORE_DMA_EDMA_DREQ_40_41[23:16]	42	MCSPI1_DREQ_RX3	McSPI1 receive request channel 3
DMA_EDMA_DREQ_42	43	CTRL_CORE_DMA_EDMA_DREQ_42_43[7:0]	43	MCSPI2_DREQ_TX0	McSPI2 transmit request channel 0
DMA_EDMA_DREQ_43	44	CTRL_CORE_DMA_EDMA_DREQ_42_43[23:16]	44	MCSPI2_DREQ_RX0	McSPI2 receive request channel 0
DMA_EDMA_DREQ_44	45	CTRL_CORE_DMA_EDMA_DREQ_44_45[7:0]	45	MCSPI2_DREQ_TX1	McSPI2 transmit request channel 1

Table 16-58. EDMA Default Request Mapping (continued)

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_EDMA_DREQ_45	46	CTRL_CORE_DMA_EDMA_DREQ_44_45[23:16]	46	MCSPI2_DREQ_RX1	McSPI2 receive request channel 1
DMA_EDMA_DREQ_46	47	CTRL_CORE_DMA_EDMA_DREQ_46_47[7:0]	47	MMC2_DREQ_TX	MMC2 transmit request
DMA_EDMA_DREQ_47	48	CTRL_CORE_DMA_EDMA_DREQ_46_47[23:16]	48	MMC2_DREQ_RX	MMC2 receive request
DMA_EDMA_DREQ_48	49	CTRL_CORE_DMA_EDMA_DREQ_48_49[7:0]	49	UART1_DREQ_TX	UART1 transmit request
DMA_EDMA_DREQ_49	50	CTRL_CORE_DMA_EDMA_DREQ_48_49[23:16]	50	UART1_DREQ_RX	UART1 receive request
DMA_EDMA_DREQ_50	51	CTRL_CORE_DMA_EDMA_DREQ_50_51[7:0]	51	UART2_DREQ_TX	UART2 transmit request
DMA_EDMA_DREQ_51	52	CTRL_CORE_DMA_EDMA_DREQ_50_51[23:16]	52	UART2_DREQ_RX	UART2 receive request
DMA_EDMA_DREQ_52	53	CTRL_CORE_DMA_EDMA_DREQ_52_53[7:0]	53	UART3_DREQ_TX	UART3 transmit request
DMA_EDMA_DREQ_53	54	CTRL_CORE_DMA_EDMA_DREQ_52_53[23:16]	54	UART3_DREQ_RX	UART3 receive request
DMA_EDMA_DREQ_54	55	CTRL_CORE_DMA_EDMA_DREQ_54_55[7:0]	55	UART4_DREQ_TX	UART4 transmit request
DMA_EDMA_DREQ_55	56	CTRL_CORE_DMA_EDMA_DREQ_54_55[23:16]	56	UART4_DREQ_RX	UART4 receive request
DMA_EDMA_DREQ_56	57	CTRL_CORE_DMA_EDMA_DREQ_56_57[7:0]	57	MMC4_DREQ_TX	MMC4 transmit request
DMA_EDMA_DREQ_57	58	CTRL_CORE_DMA_EDMA_DREQ_56_57[23:16]	58	MMC4_DREQ_RX	MMC4 receive request
DMA_EDMA_DREQ_58	59	CTRL_CORE_DMA_EDMA_DREQ_58_59[7:0]	59	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_59	60	CTRL_CORE_DMA_EDMA_DREQ_58_59[23:16]	60	Reserved	Reserved by default but can be remapped to a valid DMA source
DMA_EDMA_DREQ_60	61	CTRL_CORE_DMA_EDMA_DREQ_60_61[7:0]	61	MMC1_DREQ_TX	MMC1 transmit request
DMA_EDMA_DREQ_61	62	CTRL_CORE_DMA_EDMA_DREQ_60_61[23:16]	62	MMC1_DREQ_RX	MMC1 receive request
DMA_EDMA_DREQ_62	63	CTRL_CORE_DMA_EDMA_DREQ_62_63[7:0]	63	UART5_DREQ_TX	UART5 transmit request
DMA_EDMA_DREQ_63	64	CTRL_CORE_DMA_EDMA_DREQ_62_63[23:16]	64	UART5_DREQ_RX	UART5 receive request

16.2.4 EDMA Controller Functional Description

This chapter discusses the architecture of the EDMA controller.

16.2.4.1 Block Diagram

Figure 16-15 shows the functional block diagram of the EDMA controller.

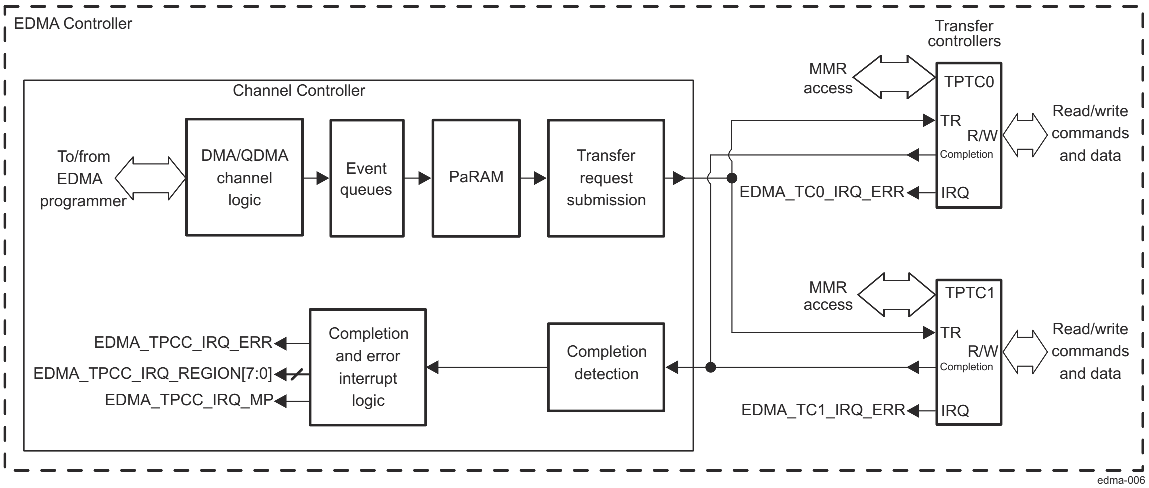


Figure 16-15. EDMA Controller Block Diagram

16.2.4.1.1 Third-Party Channel Controller

The TPCC is the EDMA transfer scheduler responsible for scheduling, arbitrating, and issuing user programmed transfers to the two TPTCs.

Figure 16-16 shows a functional block diagram of the EDMA channel controller (EDMA_TPCC).

The main blocks of the EDMA_TPCC are as follows:

- Parameter RAM (PaRAM): The PaRAM maintains parameter sets for channel and reload parameter sets. The PaRAM must be written with the transfer context for the desired channels and link parameter sets. EDMA_TPCC processes and sets based on a trigger event and submits a transfer request (TR) to the transfer controllers.
- EDMA event and interrupt processing registers: Allows mapping of events to parameter sets, enable/disable events, enable/disable interrupt conditions, and clearing interrupts.
- Completion detection: The completion detect block detects completion of transfers by the EDMA_TPTCs or slave peripherals. The completion of transfers can be used optionally to chain trigger new transfers or to assert interrupts.
- Event queues: Event queues form the interface between the event detection logic and the transfer request submission logic.
- Memory protection registers: Memory protection registers define the accesses (privilege level and requestor(s)) that are allowed to access the DMA channel shadow region view(s) and regions of PaRAM.

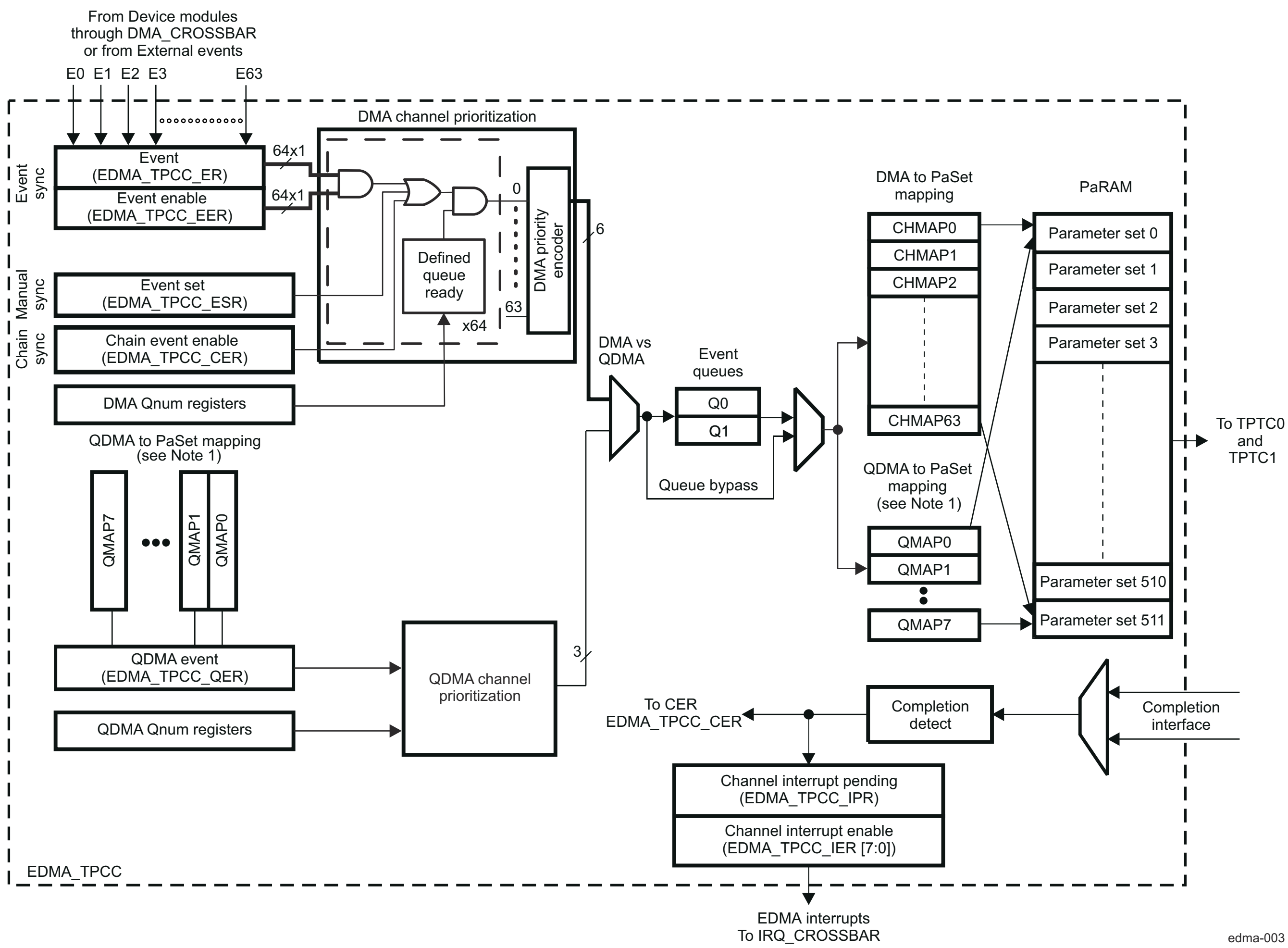
Other functions include the following:

- Region registers: Region registers allow DMA resources (DMA channels and interrupts) to be assigned to unique regions that different EDMA programmers own (for example, MPU or DSPs).
- Debug registers: Debug registers allow debug visibility by providing registers to read the queue status, controller status, and missed event status.

The EDMA_TPCC includes two channel types: DMA channels (64 channels) and QDMA channels (8 channels).

Each channel is associated with a given event queue/transfer controller and with a given PaRAM set. The main difference between a DMA channel and a QDMA channel is the method that the system uses to trigger transfers.

Figure 16-16 is a block diagram of the EDMA_TPCC.



A. Although it is depicted twice in Figure 16-16, there is only one physical register set for the QDMA to PaRAM set mapping block.

Figure 16-16. EDMA Channel Controller Block Diagram

The EDMA_TPCC supports up to 64 DMA channels and up to 8 QDMA channels. These channels are identical, except for how they are triggered:

- DMA channels are triggered by external events (such as McSPI modules TX event and McSPI modules RX event) by the event set registers EDMA_TPCC_ESR and EDMA_TPCC_ESRH, or through chaining register EDMA_TPCC_CER.
- QDMA channels are triggered automatically (auto-triggered) by the CPU. QDMAs allow a minimum number of linear writes to be issued to the TPCC to force a series of transfers to occur.

The TPCC arbitrates among pending DMA and QDMA events with a fixed [64:1] and [8:1] priority encoder for these events, respectively (a low channel number corresponds to a high priority).

DMA events are always higher priority than QDMA events. The higher-priority event is placed in the event queue to await submission to the transfer controllers, which occurs at the earliest opportunity. Each event queue is serviced in FIFO order, with a maximum of 16 queued events per event queue. If more than one TPTC is ready to be programmed with a transmission request (TR), the event queues are serviced with fixed priority: Q0 is higher than Q1. When an event is ready to be queued and the event queue and the TC channel are empty, the event bypasses the event queue and goes directly to the PaRAM processing logic for submission to the appropriate TC. If the transfer request TR bus or PaRAM processing are busy, the bypass path is not used. The bypass is not used to dequeue for a higher-priority event.

Events are extracted from the event queue when the EDMA_TPTC is available for a new TR to be programmed into the EDMA_TPTC (signaled with the empty signal, indicating an empty program register set). As an event is extracted from the event queue, the associated PaRAM entry is processed and submitted to the TPTC as a TR. The TPCC updates the appropriate counts and addresses in the PaRAM entry in anticipation of the next trigger event for that PaRAM entry.

The EDMA_TPCC also has an error detection logic that causes an error interrupt generation on various error conditions (for example: missed events EDMA_TPCC_EMR and EDMA_TPCC_EMRH registers, exceeding event queue thresholds in EDMA_TPCC_CCERR register, etc.).

16.2.4.1.2 Third-Party Transfer Controller

The TPTC module is the EDMA transfer engine that generates transfers as programmed in dedicated working registers, using two dedicated master ports: a read-only port and a write-only port.

Figure 16-17 shows a functional block diagram and of the EDMA transfer controller (EDMA_TPTC) and its connection to the EDMA_TPCC.

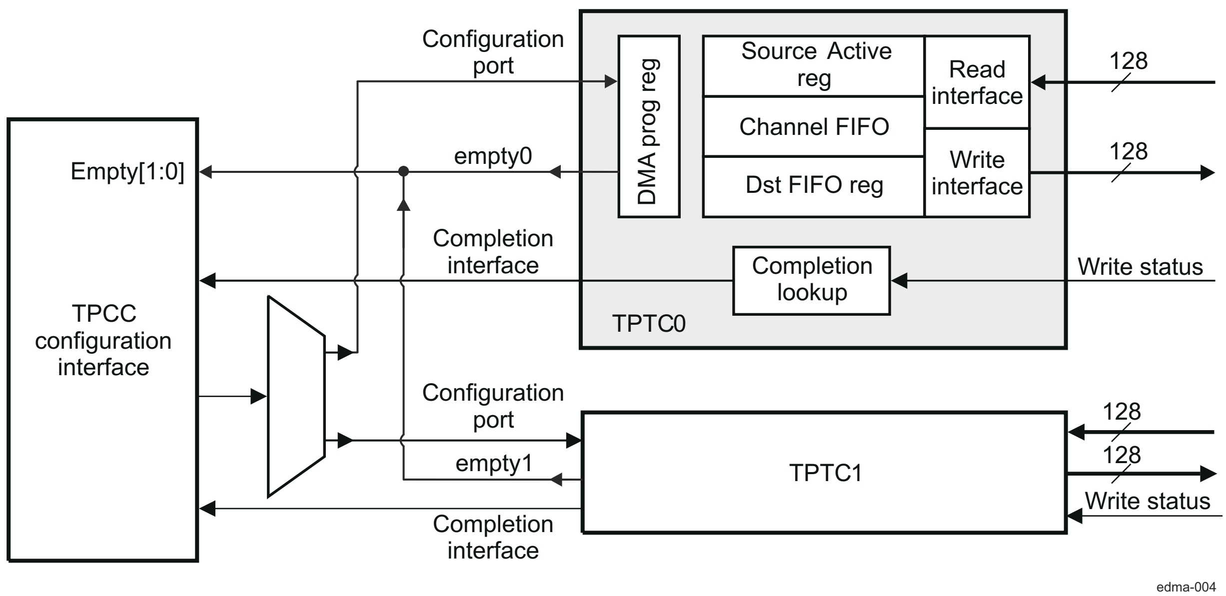


Figure 16-17. TPTC Block Diagram

Note

The port data bus width of the instances of the TPTC is fixed at 128 bits.

Two instances of the EDMA_TPTC generate concurrent traffic on the L3_MAIN interconnect. Each TC controller consists of the following components:

- DMA Program Register Set: Stores the context for the DMA transfer that is loaded into the active register set when the current active register set completes. The CPU or TPCC programs the Program Register Set, not the active register set. For typical standalone operation, the CPU programs the Program Register while the TC services the Active register set. The Program Register set includes ownership control such that CPU software and the EDMA stay synchronized relative to one another.
- Source Active Register Set : Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in progress in the Read Controller. The Active register set is split into independent Source and Destination, because the source interconnect controller and the distant interconnect controller operate independently of one another.

- Destination FIFO Register Set: Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in progress, or pending, in the Write Controller. The pending register must allow the source controller to begin processing a new TR while the distant register set processes the previous TR.
- Channel FIFO: Temporary holding buffer for in-flight data. The read return data of the source peripheral is stored in the Data FIFO, and then is written to the destination peripheral by the write command/data bus.
- Read Controller/Interconnect Read Interface: The Interconnect read interface issues optimally sized read commands to the source peripheral, based on a burst size of 128 bytes and available landing space in the channel FIFO.
- Write controller/Interconnect Write interface: The local interconnect write interface issues optimally sized write commands to the destination peripheral, based on a burst size of 128 bytes and available data in the channel FIFO.
- Completion interface: sends completion codes to the EDMA_TPCC when a transfer completes and generates interrupts and chained events in the TPCC module.
- Configuration port: Slave interface that provides read/write access to program registers and read access to all memory-mapped TPTC registers.

When one EDMA_TPTC module is idle and receive its first TR, DMA program register set receives the TR, where it transitions to the DMA source active set and the destination FIFO register set immediately. The second TR (if pending from EDMA_TPCC) is loaded into the DMA program set, ensuring it can start as soon as possible when the active transfer completes. As soon as the current active set is exhausted, the TR is loaded from the DMA program register set into the DMA source active register set as well as to the appropriate entry in the destination FIFO register set.

The read controller issues read commands controlled by the rules of command fragmentation and optimization. These are issued only when the data FIFO has space available for the data read. When sufficient data is in the data FIFO, the write controller starts issuing a write command again following the rules for command fragmentation and optimization.

Depending on the number of entries, the read controller can process up to two or four transfer requests ahead of the destination subject to the amount of free data FIFO.

16.2.4.2 Types of EDMA controller Transfers

An EDMA transfer is always defined in terms of three dimensions. Figure 16-18 shows the three dimensions used by EDMA controller transfers. These three dimensions are defined as:

- 1st Dimension or Array (A): The 1st dimension in a transfer consists of EDMA_TPCC_ABCNT_n[15:0] ACNT contiguous bytes.
- 2nd Dimension or Frame (B): The 2nd dimension in a transfer consists of EDMA_TPCC_ABCNT_n[31:16] BCNT arrays of ACNT bytes. Each array transfer in the 2nd dimension is separated from each other by an index programmed using bit-fields EDMA_TPCC_BIDX_n[15:0] SBIDX or EDMA_TPCC_BIDX_n[31:16] DBIDX.
- 3rd Dimension or Block (C): The 3rd dimension in a transfer consists of CCNT frames of BCNT arrays of ACNT bytes. The Count for 3rd Dimension is defined in register EDMA_TPCC_CCNT_n[15:0] CCNT. Each transfer in the 3rd dimension is separated from the previous by an index programmed using EDMA_TPCC_CIDX_n[15:0] SCIDX or EDMA_TPCC_CIDX_n[31:16] DCIDX.

Note

The reference point for the index depends on the synchronization type. The amount of data transferred upon receipt of a trigger/synchronization event is controlled by the synchronization types (EDMA_TPCC_OPT_n[2] SYNCDIM bit). For these three dimensions, only two synchronization types are supported: A-synchronized transfers and AB-synchronized transfers.

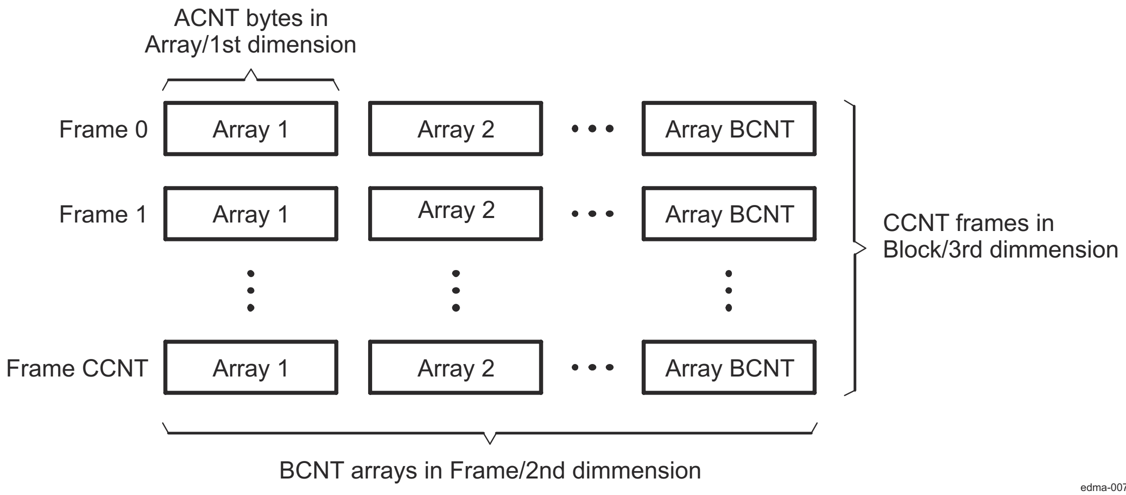


Figure 16-18. Definition of ACNT, BCNT, and CCNT

16.2.4.2.1 A-Synchronized Transfers

In an A-synchronized transfer, each EDMA sync event initiates the transfer of the 1st dimension of EDMA_TPCC_ABCNT_n[15:0] ACNT bytes, or one array of ACNT bytes. Each event/TR packet conveys the transfer information for one array only. Thus, BCNT × CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA_TPCC_BIDX_n[15:0]SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX, as shown in Figure 16-19, where the start address of Array N is equal to the start address of Array N – 1 plus source (SRC) or destination (DST) in EDMA_TPCC_BIDX_n register.

Frames are always separated by EDMA_TPCC_CIDX_n[15:0] SCIDX and EDMA_TPCC_CIDX_n[31:16] DCIDX. For A-synchronized transfers, after the frame is exhausted, the address is updated by adding SRCCIDX/ DSTCIDX to the beginning address of the last array in the frame. As in Figure 16-19, SRCCIDX / DSTCIDX is the difference between the start of Frame 0 Array 3 to the start of Frame 1 Array 0.

Figure 16-19 shows an A-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of n (ACNT) bytes. In this example, a total of 12 sync events (BCNT × CCNT) exhaust a PaRAM set. See Figure 16-19 for details on parameter set updates.

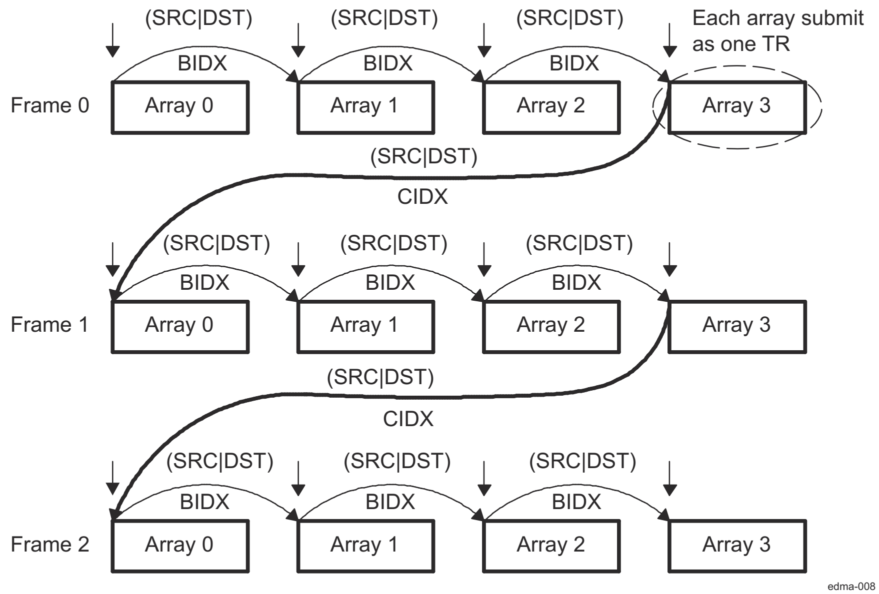


Figure 16-19. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)

16.2.4.2.2 AB-Synchronized Transfers

In a AB-synchronized transfer, each EDMA sync event initiates the transfer of 2 dimensions or one frame. Each event/TR packet conveys information for one entire frame of BCNT_n arrays of ACNT_n bytes. Thus, EDMA_TPCC_CCNT_n events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA_TPCC_BIDX_n[15:0] SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX as shown in Figure 16-20. Frames are always separated by SRCCIDX and DSTCIDX.

Note that for AB-synchronized transfers, after a TR for the frame is submitted, the address update is to add EDMA_TPCC_CIDX_n[15:0] SCIDX / EDMA_TPCC_CIDX_n[31:16] DCIDX to the beginning address of the beginning array in the frame. This is different from A-synchronized transfers where the address is updated by adding SRCCIDX/DSTCIDX to the start address of the last array in the frame. See Section 16.2.4.3.6 *Parameter Set Updates* for details on parameter set updates.

Figure 16-20 shows an AB-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of *n* (ACNT) bytes. In this example, a total of 3 sync events (CCNT) exhaust a PaRAM set; that is, a total of 3 transfers of 4 arrays each completes the transfer.

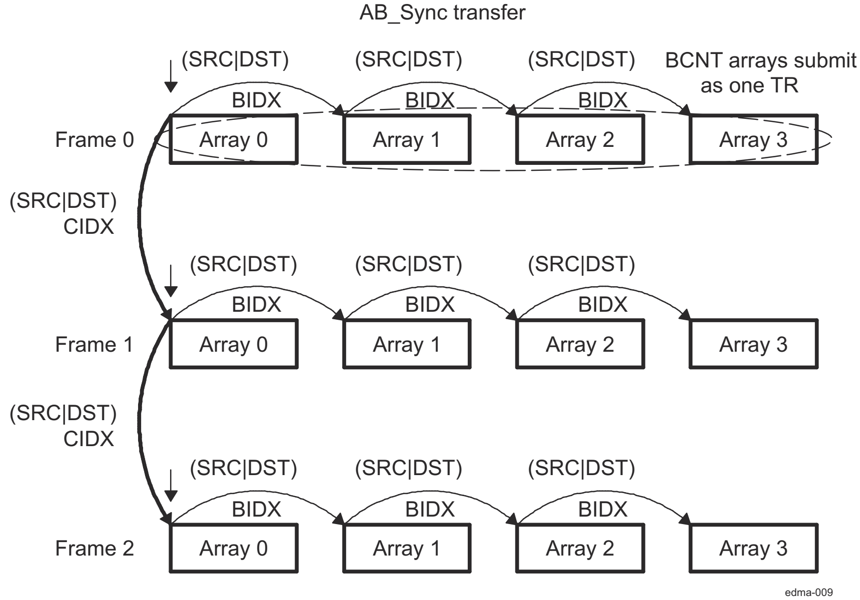


Figure 16-20. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)

Note

ABC-synchronized transfers are not directly supported. It can be logically achieved by chaining between multiple AB-synchronized transfers.

Note

VCP does not support Const/FIFO mode DMA transfers. The EDMA should be configured for AB-Synchronized transfer with ACNT = 8, BCNT = number of elements.

16.2.4.3 Parameter RAM (PaRAM)

The EDMA controller is a RAM-based architecture. The transfer context (source/destination addresses, count, indexes, etc.) for DMA or QDMA channels is programmed in a parameter RAM table in EDMA_TPCC. The PaRAM table is segmented into multiple PaRAM sets. Each PaRAM set includes eight four-byte PaRAM set entries (32-bytes total per PaRAM set), which includes typical DMA transfer parameters such as source address, destination address, transfer counts, indexes, options, etc.

The PaRAM structure supports flexible ping-pong, circular buffering, channel chaining, and auto-reloading (linking).

The contents of the PaRAM include the following:

- 512 PaRAM sets
- 64 channels that are direct mapped and can be used as link or QDMA sets if not used for DMA channels
- 64 channels remain for link or QDMA sets

By default, all channels map to PaRAM set to 0, they should be remapped before use by EDMA_TPCC_DCHMAPN_m and EDMA_TPCC_QCHMAPN_j registers.

Table 16-59. EDMA Parameter RAM Contents

PaRAM Set Number	Base Address	Parameters(1)
0	EDMA Base Address + 4000h to EDMA Base Address + 401Fh	PaRAM set 0
1	EDMA Base Address + 4020h to EDMA Base Address + 403Fh	PaRAM set 1
2	EDMA Base Address + 4040h to EDMA Base Address + 405Fh	PaRAM set 2
3	EDMA Base Address + 4060h to EDMA Base Address + 407Fh	PaRAM set 3
4	EDMA Base Address + 4080h to EDMA Base Address + 409Fh	PaRAM set 4
5	EDMA Base Address + 40A0h to EDMA Base Address + 40BFh	PaRAM set 5
6	EDMA Base Address + 40C0h to EDMA Base Address + 40DFh	PaRAM set 6
7	EDMA Base Address + 40E0h to EDMA Base Address + 40FFh	PaRAM set 7
8	EDMA Base Address + 4100h to EDMA Base Address + 411Fh	PaRAM set 8
9	EDMA Base Address + 4120h to EDMA Base Address + 413Fh	PaRAM set 9
...	...	...
63	EDMA Base Address + 47E0h to EDMA Base Address + 47FFh	PaRAM set 63
64	EDMA Base Address + 4800h to EDMA Base Address + 481Fh	PaRAM set 64
65	EDMA Base Address + 4820h to EDMA Base Address + 483Fh	PaRAM set 65
...	...	...
510	EDMA Base Address + 7FC0h to EDMA Base Address + 7FDFh	PaRAM set 510
511	EDMA Base Address + 7FE0h to EDMA Base Address + 7FFFh	PaRAM set 511

(1) The device has 8 QDMA channels that can be mapped to any parameter set number from 0 to 511.

16.2.4.3.1 PaRAM

Each parameter set of PaRAM is organized into eight 32-bit words or 32 bytes, as shown in Figure 16-21 and described in Table 16-60. Each PaRAM set consists of 16-bit and 32-bit parameters.

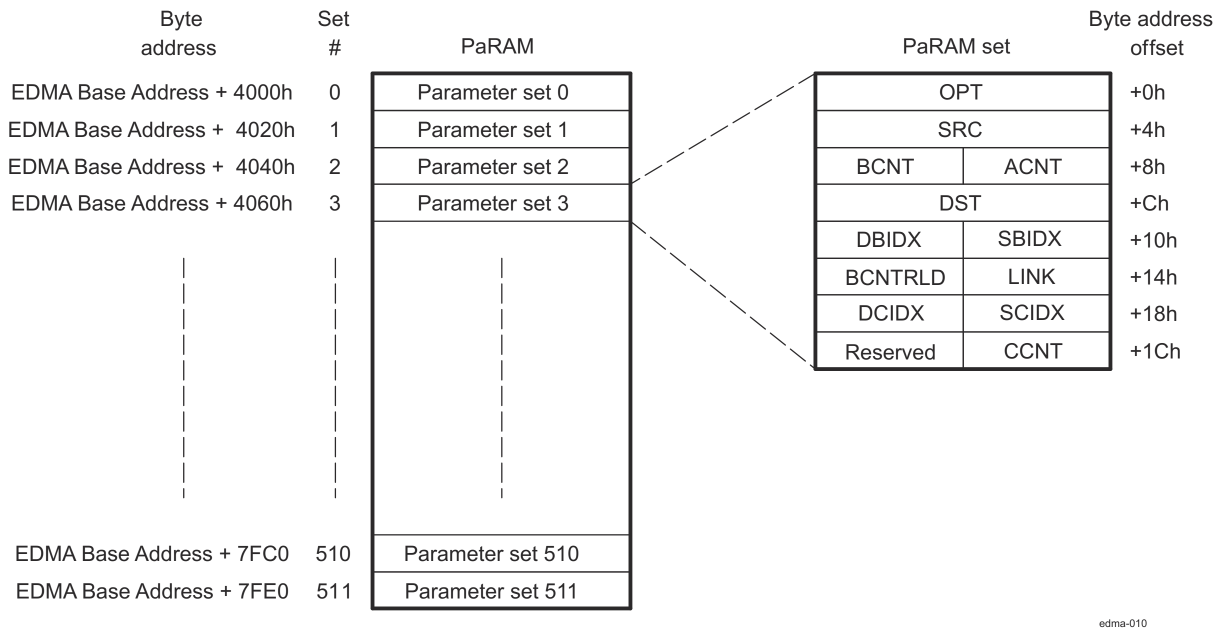


Figure 16-21. PaRAM Set

Table 16-60. EDMA Channel Parameter Description

Offset Address (bytes)	Acronym	Parameter	Description
0h	OPT	Channel Options EDMA_TPCC_OPT_n register	Transfer configuration options
4h	SRC	Channel Source Address EDMA_TPCC_SRC_n register	The byte address from which data is transferred
8h[(1)]	ACNT	Count for 1st Dimension EDMA_TPCC_ABCNT_n[15:0] ACNT bit-field.	Unsigned value specifying the number of contiguous bytes within an array (first dimension of the transfer). Valid values range from 1 to 65 535.
	BCNT	Count for 2nd Dimension EDMA_TPCC_ABCNT_n[31:16] BCNT bit-field.	Unsigned value specifying the number of arrays in a frame, where an array is ACNT bytes. Valid values range from 1 to 65 535.
Ch	DST	Channel Destination Address EDMA_TPCC_DST_n register	The byte address to which data is transferred
10h[(1)]	SBIDX	Source BCNT Index EDMA_TPCC_BIDX_n[15:0] SBIDX bit-field.	Signed value specifying the byte address offset between source arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
	DBIDX	Destination BCNT Index EDMA_TPCC_BIDX_n[31:16] DBIDX bit-field.	Signed value specifying the byte address offset between destination arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
14h[(1)]	LINK	Link Address EDMA_TPCC_LNK_n[15:0] LINK bit-field	The PaRAM address containing the PaRAM set to be linked (copied from) when the current PaRAM set is exhausted. A value of FFFFh specifies a null link.
	BCNTRLD	BCNT Reload EDMA_TPCC_LNK_n[31:16] BCNTRLD bit-field	The count value used to reload BCNT when BCNT decrements to 0 (TR is submitted for the last array in 2nd dimension). Only relevant in A-synchronized transfers.
18h[(1)]	SCIDX	Source CCNT index. EDMA_TPCC_CIDX_n[15:0] SCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last source array in a frame to the beginning of the first source array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first source array in a frame to the beginning of the first source array in the next frame.
	DCIDX	Destination CCNT index. EDMA_TPCC_CIDX_n[31:16] DCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last destination array in a frame to the beginning of the first destination array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first destination array in a frame to the beginning of the first destination array in the next frame.
1Ch	CCNT	Count for 3rd Dimension. EDMA_TPCC_CCNT_n[15:0] CCNT bit-field.	Unsigned value specifying the number of frames in a block, where a frame is BCNT arrays of ACNT bytes. Valid values range from 1 to 65 535.
	Reserved	Reserved	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

(1) If OPT, SRC, or DST is the trigger word for a QDMA transfer, then a 32-bit access to that field is required. Furthermore, it is recommended to perform only 32-bit accesses on the parameter RAM for best code compatibility. For example, switching the endianness of the processor will swap addresses of the 16-bit fields, but 32-bit accesses avoid the issue entirely.

16.2.4.3.2 EDMA Channel PaRAM Set Entry Fields

16.2.4.3.2.1 Channel Options Parameter (OPT)

For detailed information about the channel options parameter, see the EDMA_TPCC_OPT_n register description in Section 16.2.7.2.2.1, *EDMA_TPCC Register Description*.

16.2.4.3.2.2 Channel Source Address (SRC)

The 32-bit source address parameter specifies the starting byte address of the source. For SAM in increment mode, there are no alignment restrictions imposed by EDMA. For SAM in constant addressing mode, it must program the source address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA_TPTC returns an error. Refer to Section 16.2.4.12.3 *Error Generation* for additional details.

16.2.4.3.2.3 Channel Destination Address (DST)

The 32-bit destination address parameter specifies the starting byte address of the destination. For DAM in increment mode, there are no alignment restrictions imposed by EDMA. For DAM in constant addressing mode, it must program the destination address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA_TPTC returns an error. Refer to Section 16.2.4.12.3 *Error Generation* for additional details.

16.2.4.3.2.4 Count for 1st Dimension (ACNT)

EDMA_TPCC_ABCNT_n[15:0] ACNT represents the number of bytes within the 1st dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65 535. Therefore, the maximum number of bytes in an array is 65 535 bytes (64K – 1 bytes). ACNT must be greater than or equal to 1 for a TR to be submitted to EDMA_TPTC. A transfer with ACNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

Refer to Section 16.2.4.3.5 *Dummy Versus Null Transfer Comparison* and Section 16.2.4.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

16.2.4.3.2.5 Count for 2nd Dimension (BCNT)

EDMA_TPCC_ABCNT_n[15:0] BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT are between 1 and 65 535. Therefore, the maximum number of arrays in a frame is 65 535 (64K – 1 arrays). A transfer with BCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

Refer to Section 16.2.4.3.5 *Dummy Versus Null Transfer Comparison* and Section 16.2.4.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

16.2.4.3.2.6 Count for 3rd Dimension (CCNT)

EDMA_TPCC_CCNT_n[15:0] CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT are between 1 and 65 535. Therefore, the maximum number of frames in a block is 65 535 (64K – 1 frames). A transfer with CCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

A CCNT value of 0 is considered either a null or dummy transfer.

Refer to Section 16.2.4.3.5 *Dummy Versus Null Transfer Comparison* and Section 16.2.4.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

16.2.4.3.2.7 BCNT Reload (BCNTRLD)

EDMA_TPCC_LNK_n[31:16] BCNTRLD is a 16-bit unsigned value used to reload the EDMA_TPCC_ABCNT_n[15:0] BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-synchronized transfers. In this case, the EDMA_TPCC decrements the BCNT value by 1 on each TR submission. When BCNT reaches 0, the EDMA_TPCC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value.

For AB-synchronized transfers, the EDMA_TPCC submits the BCNT in the TR and the EDMA_TPTC decrements BCNT appropriately. For AB-synchronized transfers, BCNTRLD is not used.

16.2.4.3.2.8 Source B Index (SBIDX)

EDMA_TPCC_BIDX_n[15:0] SBIDX is a 16-bit signed value (2s complement) used for source address modification between each array in the 2nd dimension. Valid values for EDMA_TPCC_BIDX_n[15:0] SBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-synchronized and AB-synchronized transfers. Some examples:

- EDMA_TPCC_BIDX_n[15:0] SBIDX = 0000h (0): no address offset from the beginning of an array to the beginning of the next array. All arrays are fixed to the same beginning address.
- EDMA_TPCC_BIDX_n[15:0] SBIDX = 0003h (+3): the address offset from the beginning of an array to the beginning of the next array in a frame is 3 bytes. For example, if the current array begins at address 1000h, the next array begins at 1003h.
- EDMA_TPCC_BIDX_n[15:0] SBIDX = FFFFh (–1): the address offset from the beginning of an array to the beginning of the next array in a frame is –1 byte. For example, if the current array begins at address 5054h, the next array begins at 5053h.

16.2.4.3.2.9 Destination B Index (DBIDX)

EDMA_TPCC_BIDX_n[31:16] DBIDX is a 16-bit signed value (2s complement) used for destination address modification between each array in the 2nd dimension. Valid values for EDMA_TPCC_BIDX_n[31:16] DBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-synchronized and AB-synchronized transfers. Refer to Section 16.2.4.3.2.8 *Source B Index (SBIDX)* for examples.

16.2.4.3.2.10 Source C Index (SCIDX)

EDMA_TPCC_CIDX_n[15:0] SCIDX is a 16-bit signed value (2s complement) used for source address modification in the 3rd dimension. Valid values for EDMA_TPCC_CIDX_n[15:0] SCIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

Note

When SCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 16-19), while the current array in an AB-synchronized transfer is the first array in the frame (Figure 16-20).

16.2.4.3.2.11 Destination C Index (DCIDX)

EDMA_TPCC_CIDX_n[31:16] DCIDX is a 16-bit signed value (2s complement) used for destination address modification in the 3rd dimension. Valid values are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array TR in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

Note

When DCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 16-19), while the current array in a AB-synchronized transfer is the first array in the frame (Figure 16-20).

16.2.4.3.2.12 Link Address (LINK)

The EDMA_TPCC provides a mechanism, called linking, to reload the current PaRAM set upon its natural termination (that is, after the count fields are decremented to 0) with a new PaRAM set. The 16-bit parameter EDMA_TPCC_LNK_n[15:0] LINK specifies the byte address offset in the PaRAM from which the EDMA_TPCC loads/reloads the next PaRAM set during linking.

It must program the link address to point to a valid aligned 32-byte PaRAM set. The 5 LSBs of the LINK field should be cleared to 0.

The EDMA_TPCC ignores the upper 2 bits of the LINK entry, allowing the flexibility of programming the link address as either an absolute/literal byte address or use the PaRAM-base-relative offset address. Therefore, if it use the literal address with a range from 4000h to 7FFFh, it will be treated as a PaRAM-base-relative value of 0000h to 3FFFh.

It should check that the programed value in the EDMA_TPCC_LNK_n[15:0] LINK field is correctly, so that link update is requested from a PaRAM address that falls in the range of the available PaRAM addresses on the device.

Value of FFFFh in EDMA_TPCC_LNK_n[15:0] LINK bit-field is referred to as a NULL link that should cause the EDMA_TPCC to perform an internal write of 0 to all entries of the current PaRAM set, except for the EDMA_TPCC_LNK_n[15:0] LINK field is set to FFFFh. Also, see Section 16.2.4.5 *Completion of a DMA Transfer* for details on terminating a transfer.

16.2.4.3.3 Null PaRAM Set

A null PaRAM set is defined as a PaRAM set where all count fields (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, and EDMA_TPCC_CCNT_n[15:0] CCNT) are cleared to 0. If a PaRAM set associated with a channel is a NULL set, then when serviced by the EDMA_TPCC, the bit corresponding to the channel is set in the associated event missed register (EDMA_TPCC_EMR, EDMA_TPCC_EMRH, or EDMA_TPCC_QEMR). This bit remains set in the associated secondary event register (EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER).
This implies that any future events on the same channel are ignored by the EDMA_TPCC and it is required to clear the bit in EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER for the channel. This is considered an error condition, since events are not expected on a channel that is configured as a null transfer.

16.2.4.3.4 Dummy PaRAM Set

A dummy PaRAM set is defined as a PaRAM set where at least one of the count fields (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, or EDMA_TPCC_CCNT_n[15:0] CCNT) is cleared to 0 and at least one of the count fields is nonzero.

If a PaRAM set associated with a channel is a dummy set, then when serviced by the EDMA_TPCC, it will not set the bit corresponding to the channel (DMA/QDMA) in the event missed register (EDMA_TPCC_EMR, EDMA_TPCC_EMRH, or EDMA_TPCC_QEMR) and the secondary event register (EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER) bit gets cleared similar to a normal transfer. Future events on that channel are serviced. A dummy transfer is a legal transfer of 0 bytes.

16.2.4.3.5 Dummy Versus Null Transfer Comparison

There are some differences in the way the EDMA_TPCC logic treats a dummy versus a null transfer request. A null transfer request is an error condition, but a dummy transfer is a legal transfer of 0 bytes. A null transfer causes an error bit (E*n*) in EDMA_TPCC_EMR to get set and the E*n* bit in EDMA_TPCC_SER remains set, essentially preventing any further transfers on that channel without clearing the associated error registers.

Table 16-61 summarizes the conditions and effects of null and dummy transfer requests.

Table 16-61. Dummy and Null Transfer Request

Feature	Null TR	Dummy TR
EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR is set	Yes	No
EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER remains set	Yes	No

Table 16-61. Dummy and Null Transfer Request (continued)

Feature	Null TR	Dummy TR
Link update (STATIC = 0 in EDMA_TPCC_OPT_n)	Yes	Yes
EDMA_TPCC_QER is set	Yes	Yes
EDMA_TPCC_IPR / EDMA_TPCC_IPRH, EDMA_TPCC_CER / EDMA_TPCC_CERH is set using early completion	Yes	Yes

16.2.4.3.6 Parameter Set Updates

When a TR is submitted for a given DMA/QDMA channel and its corresponding PaRAM set, the EDMA_TPCC is responsible for updating the PaRAM set in anticipation of the next trigger event. For events that are not final, this includes address and count updates; for final events, this includes the link update.

The specific PaRAM set entries that are updated depend on the channel's synchronization type (A-synchronized or B-synchronized) and the current state of the PaRAM set. A B-update refers to the decrementing of EDMA_TPCC_ABCNT_n[31:16] BCNT in the case of A-synchronized transfers after the submission of successive TRs. A C-update refers to the decrementing of CCNT in the case of A-synchronized transfers after BCNT TRs for EDMA_TPCC_ABCNT_n[15:0] ACNT byte transfers have submitted. For AB-synchronized transfers, a C-update refers to the decrementing of EDMA_TPCC_CCNT_n[15:0] CCNT after submission of every transfer request.

Refer to Table 16-62 for details and conditions on the parameter updates. A link update occurs when the PaRAM set is exhausted, as described in Section 16.2.4.3.7 *Linking Transfers*.

After the TR is read from the PaRAM (and is in process of being submitted to EDMA_TPTC), the following fields are updated if needed:

- A-synchronized: BCNT, CCNT, SRC, DST.
- AB-synchronized: CCNT, SRC, DST.

The following fields are not updated (except for during linking, where all fields are overwritten by the link PaRAM set):

- A-synchronized: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_LNK_n[31:16] BCNTRLD, EDMA_TPCC_BIDX_n[15:0] SBIDX, EDMA_TPCC_BIDX_n[31:16] DBIDX, EDMA_TPCC_CIDX_n[15:0] SCIDX, EDMA_TPCC_CIDX_n[31:16] DCIDX, EDMA_TPCC_OPT_n, EDMA_TPCC_LNK_n[15:0]LINK.
- AB-synchronized: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, EDMA_TPCC_LNK_n[31:16] BCNTRLD, EDMA_TPCC_BIDX_n[15:0] SBIDX, EDMA_TPCC_BIDX_n[31:16] DBIDX, EDMA_TPCC_CIDX_n[15:0] SCIDX, EDMA_TPCC_CIDX_n[31:16] DCIDX, EDMA_TPCC_OPT_n, EDMA_TPCC_LNK_n[15:0]LINK.

Note

PaRAM updates only pertain to the information that is needed to properly submit the next transfer request to the EDMA_TPTC. Updates that occur while data is moved within a transfer request are tracked within the transfer controller, and is detailed in Section 16.2.4.12 *EDMA Transfer Controller (EDMA_TPTC)*. For A-synchronized transfers, the EDMA_TPCC always submits a TRP for EDMA_TPCC_ABCNT_n[15:0] ACNT bytes (EDMA_TPCC_ABCNT_n[31:16] BCNT = 1 and EDMA_TPCC_CCNT_n[15:0] CCNT = 1). For AB-synchronized transfers, the EDMA_TPCC always submits a TRP for EDMA_TPCC_ABCNT_n[15:0] ACNT bytes of BCNT arrays (EDMA_TPCC_CCNT_n[15:0] CCNT = 1). The EDMA_TPTC is responsible for updating source and destination addresses within the array based on EDMA_TPCC_ABCNT_n[15:0] ACNT and EDMA_TPCC_OPT_n[10:8] FWID. For AB-synchronized transfers, the EDMA_TPTC is also responsible to update source and destination addresses between arrays based on EDMA_TPCC_BIDX_n[15:0] SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX.

Table 16-62 shows the details of parameter updates that occur within EDMA_TPCC for A-synchronized and AB-synchronized transfers.

Table 16-62. Parameter Updates in EDMA_TPCC (for Non-Null, Non-Dummy PaRAM Set)

	A-Synchronized Transfer			AB-Synchronized Transfer		
	B-Update	C-Update	Link Update	B-Update	C-Update	Link Update
Condition:	BCNT > 1	BCNT == 1 && CCNT > 1	BCNT == 1 && CCNT == 1	N/A	EDMA_TPCC_CCNT_n[15:0] CCNT > 1	EDMA_TPCC_CCNT_n[15:0] CCNT == 1
SRC	+= SBIDX	+= SCIDX	= Link.EDMA_TPCC_SRC_n	in EDMA_TPTC	+= SCIDX	= Link.EDMA_TPCC_SRC_n
DST	+= DBIDX	+= DCIDX	= Link.EDMA_TPCC_DST_n	in EDMA_TPTC	+= DCIDX	= Link.EDMA_TPCC_DST_n
ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT
BCNT	–= 1	= BCNTRLD	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT	in EDMA_TPTC	N/A	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT
CCNT	None	–= 1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT	in EDMA_TPTC	-=1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT
SBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPTC	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
SCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPTC	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0]LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0]LINK
BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD
OPT[(1)]	None	None	= LINK.EDMA_TPCC_OPT_n	None	None	= LINK.EDMA_TPCC_OPT_n

(1) In all cases, no updates occur if EDMA_TPCC_OPT_n[3] STATIC == 1 for the current PaRAM set.

Note

The EDMA_TPCC includes no special hardware to detect when an indexed address update calculation overflows/underflows. The address update will wrap across boundaries as programmed by the user. It should ensure that no transfer is allowed to cross internal port boundaries between peripherals. A single TR must target a single source/destination slave endpoint.

16.2.4.3.7 Linking Transfers

The EDMA_TPCC provides a mechanism known as linking, which allows the entire PaRAM set to be reloaded from a location within the PaRAM memory map (for both DMA and QDMA channels). Linking is especially useful for maintaining ping-pong buffers, circular buffering, and repetitive/continuous transfers with no CPU intervention. Upon completion of a transfer, the current transfer parameters are reloaded with the parameter set pointed that the 16-bit link address field of the current parameter set points to. Linking only occurs when the EDMA_TPCC_OPT_n[3] STATIC bit is cleared.

Note

It should always link a transfer (EDMA or QDMA) to another useful transfer. If it must terminate a transfer, then link the transfer to a NULL parameter set. Refer to Section 16.2.4.3.3 *Null PaRAM Set*.

The link update occurs after the current PaRAM set event parameters have been exhausted. An event's parameters are exhausted when the EDMA channel controller has submitted all of the transfers that are associated with the PaRAM set.

A link update occurs for null and dummy transfers depending on the state of the EDMA_TPCC_OPT_n[3] STATIC bit and the EDMA_TPCC_LNK_n[15:0] LINK field. In both cases (null or dummy), if the value of EDMA_TPCC_LNK_n[15:0] LINK is FFFFh, then a null PaRAM set (with all 0s and EDMA_TPCC_LNK_n[15:0] LINK set to FFFFh) is written to the current PaRAM set.
Similarly, if EDMA_TPCC_LNK_n[15:0] LINK is set to a value other than FFFFh, then the appropriate PaRAM location that EDMA_TPCC_LNK_n[15:0] LINK points to is copied to the current PaRAM set.

Once the channel completion conditions are met for an event, the transfer parameters that are located at the link address are loaded into the current DMA or QDMA channel’s associated parameter set. This indicates that the EDMA_TPCC reads the entire set (eight words) from the PaRAM set specified by EDMA_TPCC_LNK_n[15:0] LINK and writes all eight words to the PaRAM set that is associated with the current channel. Figure 16-22 shows an example of a linked transfer.

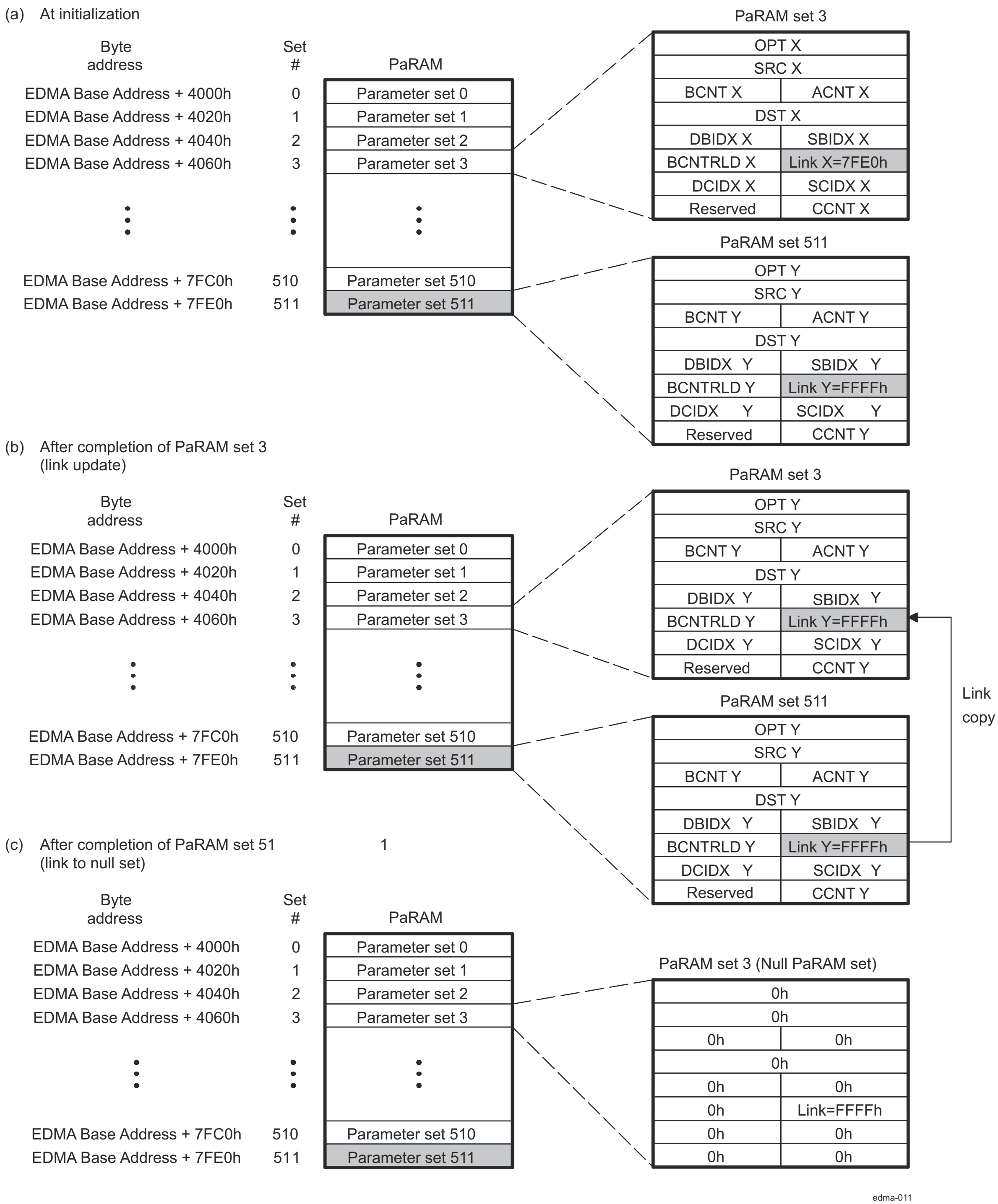


Figure 16-22. Linked Transfer

Any PaRAM set in the PaRAM can be used as a link/reload parameter set. The PaRAM sets associated with peripheral synchronization events (refer to Section 16.2.4.6 *Event, Channel, and PaRAM Mapping*) only use for linking if the corresponding events are disabled.

If a PaRAM set location is defined as a QDMA channel PaRAM set (by EDMA_TPCC_QCHMAPN_j register), then copying the link PaRAM set into the current QDMA channel PaRAM set is recognized as a trigger event. It is latched in EDMA_TPCC_QER because a write to the trigger word was performed. This feature is used to create a linked list of transfers using a single QDMA channel and multiple PaRAM sets. Refer to Section 16.2.4.4.2 *QDMA Channels*.

Linking to itself replicates the behavior of auto-initialization, thus facilitating the use of circular buffering and repetitive transfers. After an EDMA channel exhausts its current PaRAM set, it reloads all of the parameter set entries from another PaRAM set, which is initialized with values that are identical to the original PaRAM set. Figure 16-23 shows an example of a linked to self transfer. Here, the PaRAM set 511 has the link field pointing to the address of parameter set 511 (linked to self).

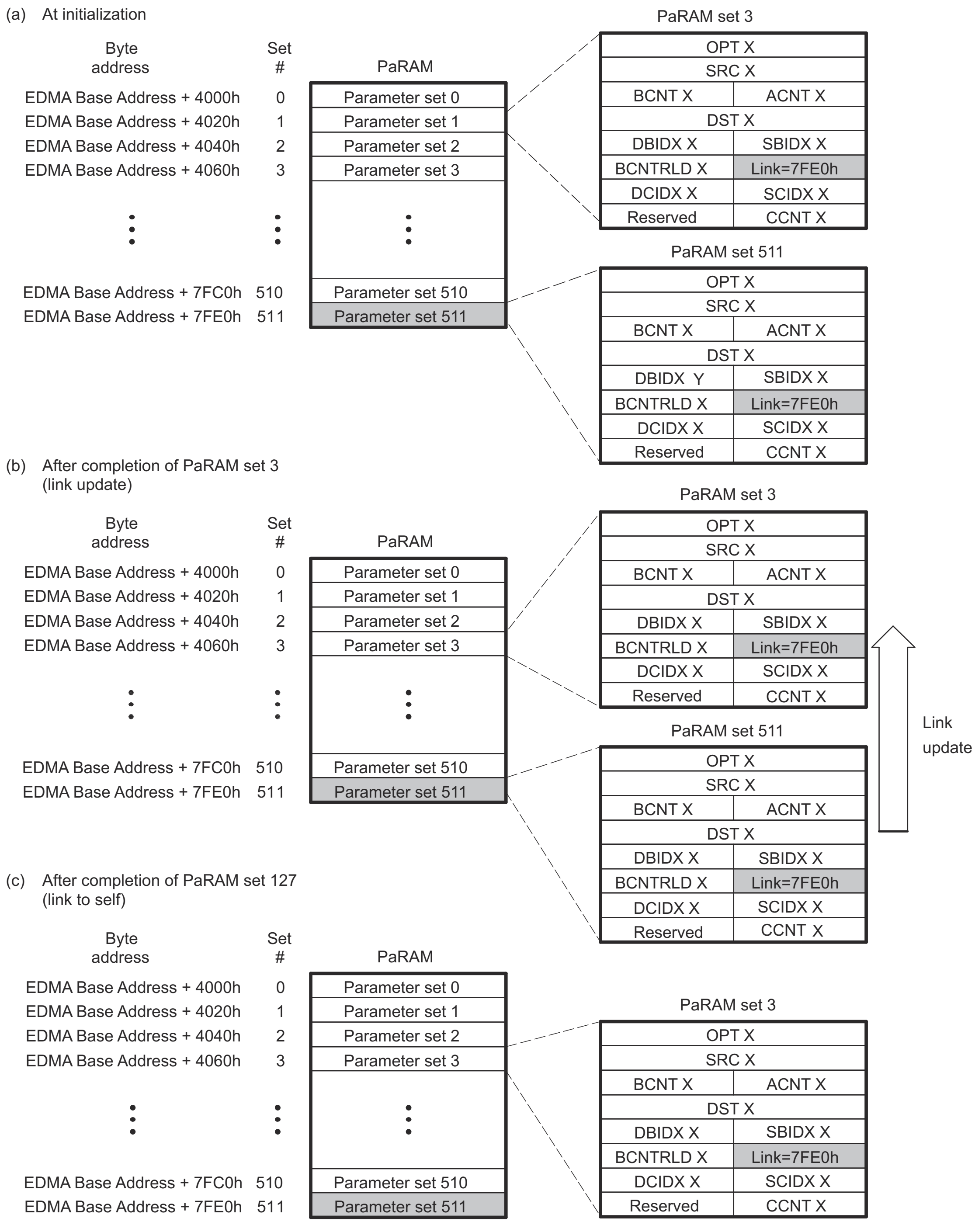


Figure 16-23. Link-to-Self Transfer

Note

If the in EDMA_TPCC_OPT_n[3] STATIC bit is set for a PaRAM set, then link updates are not performed.

16.2.4.3.8 Constant Addressing Mode Transfers/Alignment Issues

If either EDMA_TPCC_OPT_n[0] SAM or EDMA_TPCC_OPT_n[1] DAM is set (constant addressing mode), then the source or destination address must be aligned to a 256-bit aligned address, respectively, and the corresponding EDMA_TPCC_BIDX_n is an even multiple of 32 bytes (256 bits). The EDMA_TPCC does not recognize errors here, but the EDMA_TPTC asserts an error if this is not true. Refer to Section 16.2.4.12.3 *Error Generation*.

Note

The constant addressing (CONST) mode has limited applicability. The EDMA is configured for the constant addressing mode (EDMA_TPCC_OPT_n[0] SAM / EDMA_TPCC_OPT_n[1] DAM = 1) only if the transfer source or destination (on-chip memory, off-chip memory controllers, slave peripherals) support the constant addressing mode. If the constant addressing mode is not supported, the similar logical transfer can be achieved using the increment (INCR) mode (EDMA_TPCC_OPT_n[0] SAM / EDMA_TPCC_OPT_n[1] DAM =0) by appropriately programming the count and indices values.

16.2.4.3.9 Element Size

The EDMA controller does not use element-size and element-indexing. Instead, all transfers are defined in terms of all three dimensions: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, and EDMA_TPCC_CCNT_n[15:0] CCNT. An element-indexed transfer is logically achieved by programming EDMA_TPCC_ABCNT_n[15:0] ACNT to the size of the element and EDMA_TPCC_ABCNT_n[31:16] BCNT to the number of elements that need to be transferred. For example: If there are 16-bit audio data and 256 audio samples that must be transferred to a serial port, therefore the EDMA_TPCC_ABCNT_n[15:0] ACNT = 2 (2 bytes) and EDMA_TPCC_ABCNT_n[31:16] BCNT = 256.

16.2.4.4 Initiating a DMA Transfer

There are multiple ways to initiate a programmed data transfer using the EDMA_TPCC channel controller. Transfers on DMA channels are initiated by three sources.

They are listed as follows:

- **Event-triggered transfer request** (this is the typical usage of EDMA controller): A peripheral, system, or externally-generated event triggers a transfer request.
- **Manually-triggered transfer request:**The CPU manually triggers a transfer by writing a 1 to the corresponding bit in the event set registers (EDMA_TPCC_ESR / EDMA_TPCC_ESRH).
- **Chain-triggered transfer request:** A transfer is triggered on the completion of another transfer or sub-transfer.

Transfers on QDMA channels are initiated by two sources. They are as follows:

- **Auto-triggered transfer request:** Writing to the programmed trigger word triggers a transfer.
- **Link-triggered transfer requests:** Writing to the trigger word triggers the transfer when linking occurs.

16.2.4.4.1 DMA Channel

16.2.4.4.1.1 Event-Triggered Transfer Request

When an event is asserted from a peripheral or device pins, it gets latched in the corresponding bit of the event register (EDMA_TPCC_ER[31:0] E*n* = 1). For more information about peripheral events to EDMA events mapping, refer to *the device data manual*.
If the corresponding event in the event enable register (EDMA_TPCC_EER) is enabled (EDMA_TPCC_EER[31:0] E*n* = 1), then the EDMA_TPCC prioritizes and queues the event in the appropriate

event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

If the PaRAM set is valid (not a NULL set), then a transfer request packet (TRP) is submitted to the EDMA_TPTC and the EDMA_TPCC_ER[31:0] E*n* bit is cleared. At this point, a new event can be safely received by the EDMA_TPCC.

If the PaRAM set associated with the channel is a NULL set (see Section 16.2.4.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_ER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register (EDMA_TPCC_EMR[31:0] E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include cleaning the event missed error before re-triggering the DMA channel.

When an event is received, the corresponding event bit in the event register is set (EDMA_TPCC_ER[31:0] E*n* = 1), regardless of the state of EDMA_TPCC_EER[31:0] E*n*. If the event is disabled when an external event is received (EDMA_TPCC_ER[31:0] E*n* = 1 and EDMA_TPCC_EER[31:0] E*n* = 0), the EDMA_TPCC_ER[31:0] E*n* bit remains set. If the event is subsequently enabled (EDMA_TPCC_EER[31:0] E*n* = 1), then the pending event is processed by the EDMA_TPCC and the TR is processed/submitted, after which the EDMA_TPCC_ER[31:0] E*n* bit is cleared.

If an event is being processed (prioritized or is in the event queue) and another sync event is received for the same channel prior to the original being cleared (EDMA_TPCC_ER[31:0] E*n* != 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EDMA_TPCC_EMR[31:0] E*n* = 1).

16.2.4.4.1.2 Manually-Triggered Transfer Request

The CPU or any peripheral device module initiates a DMA transfer by writing to the event set register EDMA_TPCC_ESR. Writing a 1 to an event bit in the EDMA_TPCC_ESR results in the event being prioritized/ queued in the appropriate event queue, regardless of the state of the EDMA_TPCC_EER[31:0] E*n* bit. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 16.2.4.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_ER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA_TPCC_EMR[31:0] E*n* = 1 to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include clearing the event missed error before re-triggering the DMA channel.

If an event is being processed (prioritized or is in the event queue) and the same channel is manually set by a write to the corresponding channel bit of the event set register EDMA_TPCC_ESR[31:0] E*n* = 1 prior to the original being cleared EDMA_TPCC_ESR[31:0] E*n* = 0, then the second event is registered as a missed event in the corresponding bit of the event missed register EDMA_TPCC_EMR[31:0] E*n* = 1.

16.2.4.4.1.3 Chain-Triggered Transfer Request

Chaining is a mechanism by which the completion of one transfer automatically sets the event for another channel. When a chained completion code is detected, the value of which is dictated by the transfer completion code EDMA_TPCC_OPT_n[17:12] TCC of the PaRAM set associated with the channel, it results in the corresponding bit in the chained event register EDMA_TPCC_CER to be set EDMA_TPCC_CER[31:0] E[TCC] = 1).

Once a bit is set in EDMA_TPCC_CER, the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 16.2.4.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_CER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA_TPCC_EMR[31:0] E*n* = 1 to indicate that the event was discarded due to a null TR being serviced. In this case, the error condition must be cleared before the DMA channel can be re-triggered. Good programming practices might include clearing the event missed error before re-triggering the DMA channel.

If a chaining event is being processed (prioritized or queued) and another chained event is received for the same channel prior to the original being cleared EDMA_TPCC_CER[31:0] E*n* != 0), then the second chained event is registered as a missed event in the corresponding channel bit of the event missed register EDMA_TPCC_EMR[31:0] E*n* = 1.

Note

Chained event registers EDMA_TPCC_CER, event registers EDMA_TPCC_ER, and event set registers EDMA_TPCC_ESR operate independently. An event E*n* can be triggered by any of the trigger sources (event-triggered, manually-triggered, or chain-triggered).

16.2.4.4.2 QDMA Channels

16.2.4.4.2.1 Auto-triggered and Link-Triggered Transfer Request

QDMA-based transfer requests are issued when a QDMA event gets latched in the QDMA event register EDMA_TPCC_QER[31:0] E*n* = 1. A bit corresponding to a QDMA channel is set in the QDMA event register EDMA_TPCC_QER when the following occurs:

- A CPU (or any device module) write occurs to a PaRAM address that is defined as a QDMA channel trigger word (programmed in the QDMA channel mapping register EDMA_TPCC_QCHMAPN_j for the particular QDMA channel and the QDMA channel is enabled via the QDMA event enable register EDMA_TPCC_QEER[31:0] E*n* = 1.
- EDMA_TPCC performs a link update on a PaRAM set address that is configured as a QDMA channel matches EDMA_TPCC_QCHMAPN_j settings and the corresponding channel is enabled via the QDMA event enable register EDMA_TPCC_QEER[31:0] E*n* = 1.

Once a bit is set in EDMA_TPCC_QER, the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If a bit is already set in EDMA_TPCC_QER[31:0] E*n* = 1 and a second QDMA event for the same QDMA channel occurs prior to the original being cleared, the second QDMA event gets captured in the QDMA event miss register EDMA_TPCC_QEMR[7:0] E*n* = 1.

16.2.4.4.3 Comparison Between DMA and QDMA Channels

The primary difference between DMA and QDMA channels is the event/channel synchronization.

QDMA events are either auto-triggered or link triggered. Auto-triggering allows QDMA channels to be triggered by CPU(s) with a minimum number of linear writes to PaRAM. Link triggering allows a linked list of transfers to be executed, using a single QDMA PaRAM set and multiple link PaRAM sets.

A QDMA transfer is triggered when a CPU (or other device modules) writes to the trigger word of the QDMA channel parameter set (auto-triggered) or when the EDMA_TPCC performs a link update on a PaRAM set that has been mapped to a QDMA channel (link triggered).

Note

The CPUs triggered (manually triggered) DMA channels, in addition to writing to the PaRAM set, it is required to write to the event set register EDMA_TPCC_ESR to kick-off the transfer.

QDMA channels are typically for cases where a single event accomplishes a complete transfer since the CPU (or other device modules) must reprogram some portion of the QDMA PaRAM set in order to re-trigger the channel. QDMA transfers are programmed with EDMA_TPCC_ABCNT_n[31:0] BCNT =1 and EDMA_TPCC_CCNT_n[15:0] CCNT = 1 for A-synchronized transfers, and EDMA_TPCC_CCNT_n[15:0] CCNT = 1 for AB-synchronized transfers.

Additionally, since linking is also supported (if EDMA_TPCC_OPT_n[3] STATIC = 0) for QDMA transfers, it allows to initiate a linked list of QDMAs, so when EDMA_TPCC copies over a link PaRAM set (including the write to the trigger word), the current PaRAM set mapped to the QDMA channel automatically recognizes as a valid QDMA event and initiate another set of transfers as specified by the linked set.

16.2.4.5 Completion of a DMA Transfer

A parameter set for a given channel is complete when the required number of transfer requests is submitted (based on receiving the number of synchronization events). The expected number of TRs for a non-null/non-dummy transfer is shown in Table 16-63 for both synchronization types along with state of the PaRAM set prior to the final TR being submitted. When the counts (EDMA_TPCC_ABCNT_n[31:0] BCNT and/or EDMA_TPCC_CCNT_n[15:0] CCNT) are this value, the next TR results in:

- Final chaining or interrupt codes sent by the transfer controllers (instead of intermediate).
- Link updates (linking to either null or another valid link set).

Table 16-63. Expected Number of Transfers for Non-Null Transfer

Sync Mode	Counts at time 0	Total # Transfers	Counts prior to final TR
A-synchronized	ACNT BCNT CCNT	(BCNT × CCNT) TRs of ACNT bytes each	EDMA_TPCC_ABCNT_n[31:0] BCNT == 1 && EDMA_TPCC_CCNT_n[15:0] CCNT == 1
AB-synchronized	ACNT BCNT CCNT	CCNT TRs for ACNT × BCNT bytes each	EDMA_TPCC_CCNT_n[15:0] CCNT == 1

The PaRAM OPT field must program with a specific transfer completion code TCC or EDMA_TPCC_OPT_n[17:12] TCC along with the other EDMA_TPCC_OPT_n fields ([22] TCCHEN, [20] TCINTEN, [23] ITCCHEN, and [21] ITCINTEN bits) to indicate whether the completion code is to be used for generating a chained event or/and for generating an interrupt upon completion of a transfer.

The specific EDMA_TPCC_OPT_n[17:12] TCC value (6-bit binary value) programmed dictates which of the 64-bits in the chain event register EDMA_TPCC_CER [TCC] and/or interrupt pending register EDMA_TPCC_IPR [TCC] is set.

It can selectively program whether the transfer controller sends back completion codes on completion of the final transfer request (TR) of a parameter set EDMA_TPCC_OPT_n[22] TCCHEN or EDMA_TPCC_OPT_n[20] TCINTEN, for all but the final transfer request (TR) of a parameter set EDMA_TPCC_OPT_n[23] ITCCHEN or EDMA_TPCC_OPT_n[21] ITCINTEN), or for all TRs of a parameter set (both). Refer to Section 16.2.4.8 *Chaining EDMA Channels* for details on chaining (intermediate/final chaining) and Section 16.2.4.9 *EDMA Interrupts* for details on intermediate/final interrupt completion.

A completion detection interface exists between the EDMA channel controller and transfer controller(s). This interface sends back information from the transfer controller to the channel controller to indicate that a specific transfer is completed. Completion of a transfer is used for generating chained events and/or generating interrupts to the CPU(s).

All DMA/QDMA PaRAM sets must also specify a link address value. For repetitive transfers such as ping-pong buffers, the link address value must point to another predefined PaRAM set. Alternatively, a non-repetitive transfer must set the link address value to the null link value. The null link value is defined as FFFFh. Refer to Section 16.2.4.3.7 *Linking Transfers* for more details.

Note

Any incoming events that are mapped to a null PaRAM set results in an error condition. The error condition must clear before the corresponding channel is used again. Refer to Section 16.2.4.3.5 *Dummy Versus Null Transfer Comparison*.

There are three ways the EDMA_TPCC gets updated/informed about a transfer completion: normal completion, early completion, and dummy/null completion. This applies to both chained events and completion interrupt generation.

16.2.4.5.1 Normal Completion

In normal completion mode EDMA_TPCC_OPT_n[11] TCCMODE = 0, the transfer or sub-transfer is considered to be complete when the EDMA channel controller receives the completion codes from the EDMA transfer controller. In this mode, the completion code to the channel controller is posted by the transfer controller after it receives a signal from the destination peripheral. Normal completion is typically used to generate an interrupt to inform the CPU that a set of data is ready for processing.

16.2.4.5.2 Early Completion

In early completion mode EDMA_TPCC_OPT_n[11] TCCMODE = 1, the transfer is considered to be complete when the EDMA channel controller submits the transfer request (TR) to the EDMA transfer controller. In this mode, the channel controller generates the completion code internally. Early completion is typically useful for chaining, as it allows subsequent transfers to be chained-triggered while the previous transfer is still in progress within the transfer controller, maximizing the overall throughput of the set of the transfers.

16.2.4.5.3 Dummy or Null Completion

This is a variation of early completion. Dummy or null completion is associated with a dummy set Section 16.2.4.3.4 or null set Section 16.2.4.3.3. In both cases, the EDMA channel controller does not submit the associated transfer request to the EDMA transfer controller(s). However, if the set (dummy/null) has the OPT field programmed to return completion code (intermediate/final interrupt/chaining completion), then it sets the appropriate bits in the interrupt pending registers EDMA_TPCC_IPR and EDMA_TPCC_IPRH or chained event register EDMA_TPCC_CER and EDMA_TPCC_CERH. The internal early completion path is used by the channel controller to return the completion codes internally (that is, EDMA_TPCC generates the completion code).

16.2.4.6 Event, Channel, and PaRAM Mapping

Several of the 64 DMA channels are tied to a specific hardware event, thus allowing events from device peripherals or external hardware (via the dma_evt[4:1] pins) to trigger transfers. A DMA channel typically requests a data transfer when it receives its event (apart from manually-triggered, chain-triggered, and other transfers). The amount of data transferred per synchronization event depends on the channel's configuration (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, EDMA_TPCC_CCNT_n[15:0] CCNT, etc.) and the synchronization type (A-synchronized or AB-synchronized).

The association of an event to a channel is fixed within the EDMA Channel Controller, that is, each DMA channel has one specific event associated with it. The EDMA event crossbar can be used to select which level events (of which there are more than 64) are mapped to the 64 input events to the EDMA Channel Controller. The default mapping and event crossbar mapping are defined in Section 16.2.3.1, *EDMA Requests to the EDMA Controller*. The event crossbar mapping is controlled by the device Control Modules registers refer to *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

In an application, if a channel does not use the associated synchronization event or if it does not have an associated synchronization event (unused), that channel can be used for manually-triggered or chained-triggered transfers, for linking/reloading, or as a QDMA channel.

16.2.4.6.1 DMA Channel to PaRAM Mapping

The mapping between the DMA channel numbers and the PaRAM sets is programmable (see Table 16-59). The DMA channel mapping registers EDMA_TPCC_DCHMAPN_m in the EDMA_TPCC provide programmability that allows the DMA channels to be mapped to any of the PaRAM sets in the PaRAM memory map. Figure 16-24 illustrates the use of EDMA_TPCC_DCHMAPN_m. There is one EDMA_TPCC_DCHMAPN_m register per channel.

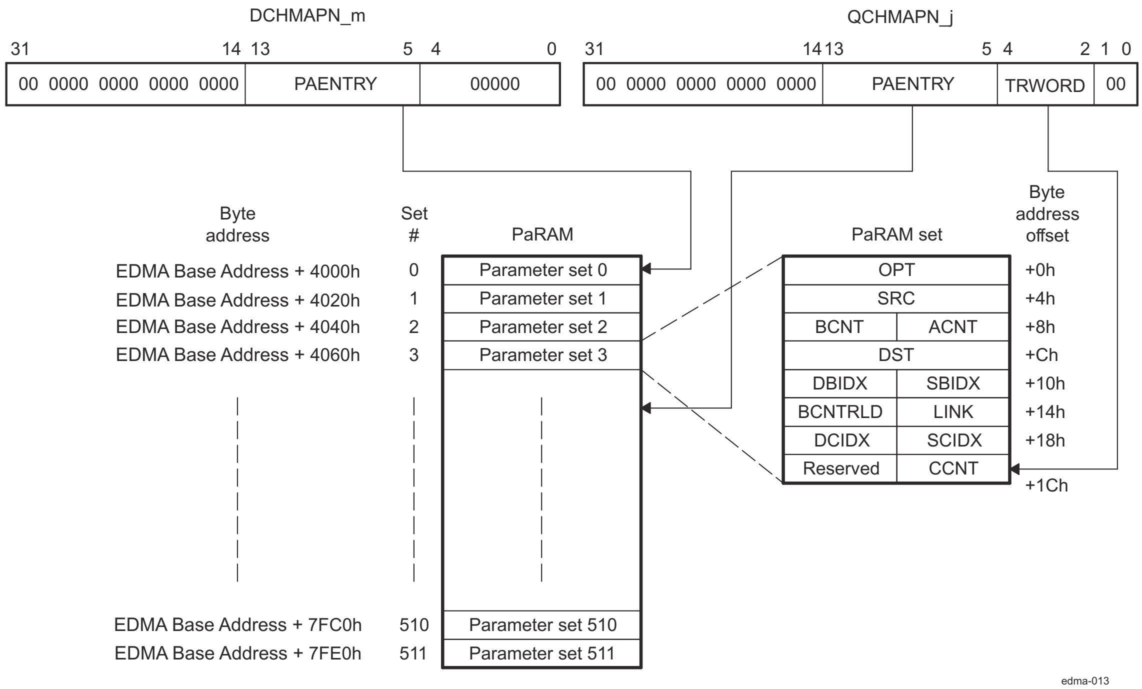


Figure 16-24. DMA Channel and QDMA Channel to PaRAM Mapping

16.2.4.6.2 QDMA Channel to PaRAM Mapping

The mapping between the QDMA channels and the PaRAM sets is programmable. The QDMA channel mapping register EDMA_TPCC_QCHMAPN_j in the EDMA_TPCC allows to map the QDMA channels to any of the PaRAM sets in the PaRAM memory map. Figure 16-25 illustrates the use of EDMA_TPCC_QCHMAPN_j.

EDMA_TPCC_QCHMAPN_j[4:2] TRWORD bit-field allows to program the trigger word in the PaRAM set for the QDMA channel. A trigger word is one of the eight words in the PaRAM set. For a QDMA transfer to occur, a valid TR synchronization event for EDMA_TPCC is a write to the trigger word in the PaRAM set pointed to by EDMA_TPCC_QCHMAPN_j for a particular QDMA channel. By default, QDMA channels are mapped to PaRAM set 0.

It must appropriately re-map PaRAM set 0 before use.

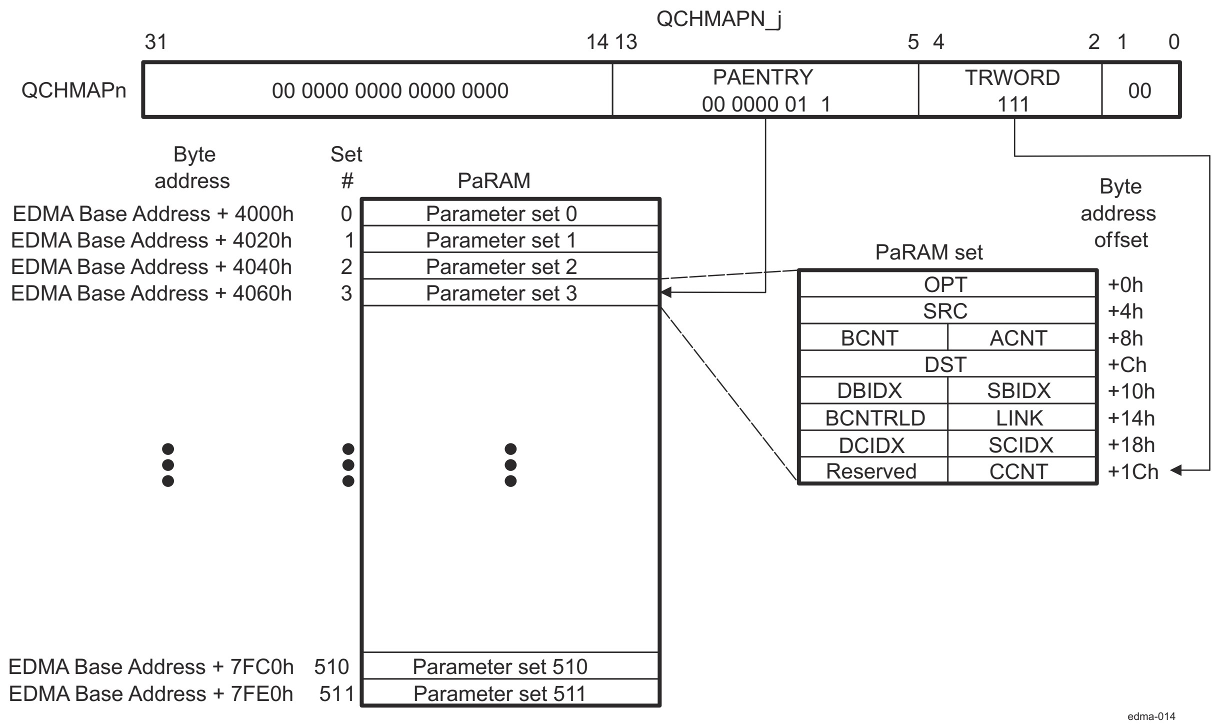


Figure 16-25. QDMA Channel to PaRAM Mapping

16.2.4.7 EDMA Channel Controller Regions

The EDMA channel controller divides its address space into eight regions. Individual channel resources are assigned to a specific region, where each region is typically assigned to a specific device module uses the EDMA controller.

Application software can use regions or to ignore them altogether. It can be used active memory protection in conjunction with regions so that only a specific device module which uses the EDMA (for example, privilege identification) or privilege level (for example, user vs. supervisor) is allowed access to a given region, and thus to a given DMA or QDMA channel. This allows robust system-level DMA code where each EDMA initiator only modifies the state of the assigned resources. Memory protection is described in Section 16.2.4.10 *Memory Protection*.

16.2.4.7.1 Region Overview

The EDMA channel controller memory-mapped registers are divided in three main categories:

1. Global registers
2. Global region channel registers
3. Shadow region channel registers

The global registers are located at a single/fixed location in the EDMA_TPCC memory map. These registers control EDMA resource mapping and provide debug visibility and error tracking information.

The channel registers (including DMA, QDMA, and interrupt registers) are accessible via the global channel region address range, or in the shadow *n* channel region address range(s). For example, the event enable register EDMA_TPCC_EER is visible at the global address of EDMA Base Address + 1020h or region addresses of EDMA Base Address + 2020h for region 0, EDMA Base Address + 2220h for region 1, … EDMA Base Address + 2E20h for region 7.

The DMA region access enable registers EDMA_TPCC_DRAEM_k and the QDMA region access enable registers EDMA_TPCC_QRAEN_k control the underlying control register bits that are accessible via the shadow region address space (except for EDMA_TPCC_IEVAL and EDMA_TPCC_IEVAL_RN_k registers). Table 16-64 lists the registers in the shadow region memory map. Refer to EDMA_TPCC register summary Table 16-80 for the complete global and shadow region memory maps.

Table 16-64. Shadow Region Registers

EDMA_TPCC_DRAEM_k	EDMA_TPCC_DRAEHM_k	EDMA_TPCC_QRAEN_k
EDMA_TPCC_ER	EDMA_TPCC_ERH	EDMA_TPCC_QER
EDMA_TPCC_ECR	EDMA_TPCC_ECRH	EDMA_TPCC_QEER
EDMA_TPCC_ESR	EDMA_TPCC_ESRH	EDMA_TPCC_QEECR
EDMA_TPCC_CER	EDMA_TPCC_CERH	EDMA_TPCC_QEESR
EDMA_TPCC_EER	EDMA_TPCC_EERH	
EDMA_TPCC_EECR	EDMA_TPCC_EECRH	
EDMA_TPCC_EESR	EDMA_TPCC_EESRH	
EDMA_TPCC_SER	EDMA_TPCC_SERH	
EDMA_TPCC_SECR	EDMA_TPCC_SECRH	
EDMA_TPCC_IER	EDMA_TPCC_IERH	
EDMA_TPCC_IECR	EDMA_TPCC_IECRH	
EDMA_TPCC_IESR	EDMA_TPCC_IESRH	
EDMA_TPCC_IPR	EDMA_TPCC_IPRH	
EDMA_TPCC_ICR	EDMA_TPCC_ICRH	
Register not affected by DRAE\DRAEH		
EDMA_TPCC_IEVAL		
EDMA_TPCC_IEVAL_RN_k		

Figure 16-26 illustrates the conceptual view of the regions.

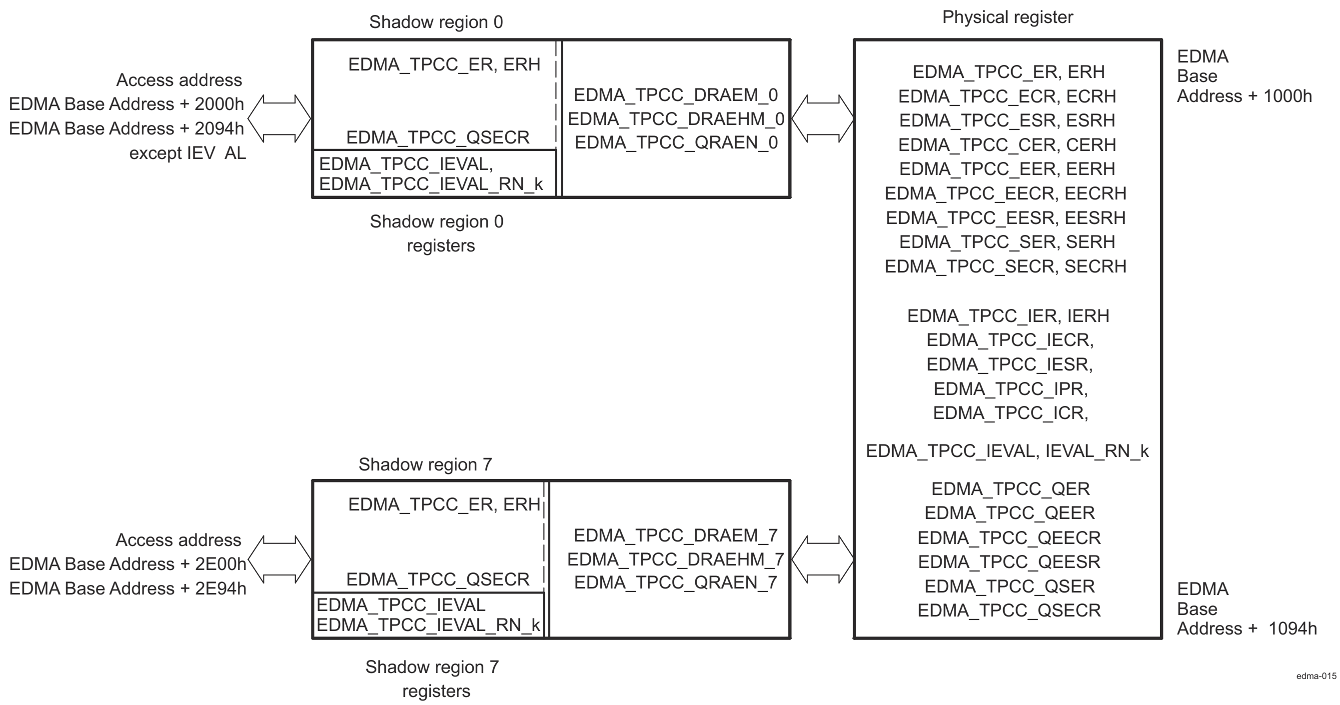


Figure 16-26. Shadow Region Registers

16.2.4.7.2 Channel Controller Regions

There are eight EDMA shadow regions (and associated memory maps). Associated with each shadow region are a set of registers defining which channels and interrupt completion codes belong to that region. These registers are user-programmed per region to assign ownership of the DMA/QDMA channels to a region.

- EDMA_TPCC_DRAEM_k and EDMA_TPCC_DRAEHM_k: One register pair exists for each of the shadow regions. The number of bits in each register pair matches the number of DMA channels (64 DMA channels). These registers need to be programmed to assign ownership of DMA channels and interrupt (or EDMA_TPCC_OPT_n[17:12] TCC codes) to the respective region. Accesses to DMA and interrupt registers via the shadow region address view are filtered through the DRAEM/DRAEHM pair. A value of 1 in the corresponding EDMA_TPCC_DRAEM_k[31:0] / EDMA_TPCC_DRAEHM_k[31:0] bit implies that the corresponding DMA interrupt channel is accessible; a value of 0 in the corresponding EDMA_TPCC_DRAEM_k[31:0] / EDMA_TPCC_DRAEHM_k[31:0] bit forces writes to be discarded and returns a value of 0 for reads.
- EDMA_TPCC_QRAEN_k: One register exists for every region. The number of bits in each register matches the number of QDMA channels (4 QDMA channels). These registers must be programmed to assign ownership of QDMA channels to the respective region. To enable a channel in a shadow region using shadow region 0 EDMA_TPCC_QEER, the corresponding bits in QRAE must be set or writing into EDMA_TPCC_QEESR there will be no the desired effect.
- EDMA_TPCC_MPPAN_k and EDMA_TPCC_MPPAG: One register exists for every region. This register defines the privilege level, requestor, and types of accesses allowed to a region's memory-mapped registers.

It is typical for an application to have a unique assignment of QDMA/DMA channels (and, therefore, a given bit position) to a given region.

The use of shadow regions allows restricted access to EDMA resources (DMA channels, QDMA channels, TCC, interrupts) by tasks in a system by setting or clearing bits in the EDMA_TPCC_DRAEM_k / EDMA_TPCC_QRAEN_k registers.

If exclusive access to any given channel / TCC code is required for a region, then only that region's EDMA_TPCC_DRAEM_k / EDMA_TPCC_QRAEN_k have the associated bit set.

Example 16-1. Resource Pool Division Across Two Regions

This example illustrates a resource pool division across two regions, assuming region 0 must be allocated 16 DMA channels (0-15) and 1 QDMA channel (0) and 32 TCC codes (0-15 and 48-63).

Region 1 needs to be allocated 16 DMA channels (16-32) and the remaining 7 QDMA channels (1-7) and TCC codes (16-47).

EDMA_TPCC_DRAEM_k should be equal to the OR of the bits that are required for the DMA channels and the TCC codes:

```
Region 0: DRAEHM, DRAEM = 0xFFFF0000, 0x0000FFFF QRAEN = 0x0000001
Region 1: DRAEHM, DRAEM = 0x0000FFFF, 0xFFFF0000 QRAEN = 0x00000FE
```

16.2.4.7.3 Region Interrupts

In addition to the EDMA_TPCC global completion interrupt, there is an additional completion interrupt line that is associated with every shadow region. Along with the interrupt enable register EDMA_TPCC_IER, DRAEM acts as a secondary interrupt enable for the respective shadow region interrupts. Refer to Table 16-57 *Hardware Request* for more information about EDMA Interrupts.

16.2.4.8 Chaining EDMA Channels

The channel chaining capability for the EDMA allows the completion of an EDMA channel transfer to trigger another EDMA channel transfer. The purpose is to allow the ability to chain several events through one event occurrence.

Chaining is different from linking (Section 16.2.4.3.7 *Linking Transfers*). The EDMA link feature reloads the current channel parameter set with the linked parameter set. The EDMA chaining feature does not modify or update any channel parameter set. It provides a synchronization event to the chained channel (see Section 16.2.4.4.1.3 *Chain-Triggered Transfer Request*).

Chaining is achieved at either final transfer completion or intermediate transfer completion, or both, of the current channel. Consider a channel *m* (DMA/QDMA) required to chain to channel *n*. Channel number *n* (0-63) needs to be programmed into the EDMA_TPCC_OPT_n[17:12] TCC bit-field of channel *m* channel options parameter (OPT) set.

- If final transfer completion chaining EDMA_TPCC_OPT_n[22] TCCHEN = 1 is enabled, the chain-triggered event occurs after the submission of the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).
- If intermediate transfer completion chaining EDMA_TPCC_OPT_n[23] ITCCHEN = 1 is enabled, the chain-triggered event occurs after every transfer request, except the last of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion chaining (EDMA_TPCC_OPT_n[22] TCCHEN = 1 and EDMA_TPCC_OPT_n[23] ITCCHEN = 1) are enabled, then the chain-trigger event occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 16-65 illustrates the number of chain event triggers occurring in different synchronized scenarios. Consider channel 31 programmed with EDMA_TPCC_ABCNT_n[15:0] ACNT = 3, EDMA_TPCC_ABCNT_n[31:16] BCNT = 4, EDMA_TPCC_CCNT_n[15:0] CCNT = 5, and EDMA_TPCC_OPT_n[17:12] TCC = 30.

Table 16-65. Chain Event Triggers

	(Number of chained event triggers on channel 30)	
Options	**A-Synchronized**	**AB-Synchronized**
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 0	1 (Owing to the last TR)	1 (Owing to the last TR)
EDMA_TPCC_OPT_n[22] TCCHEN = 0, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	19 (Owing to all but the last TR)	4 (Owing to all but the last TR)

Table 16-65. Chain Event Triggers (continued)

	(Number of chained event triggers on channel 30)	
Options	**A-Synchronized**	**AB-Synchronized**
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	20 (Owing to a total of 20 TRs)	5 (Owing to a total of 5 TRs)

16.2.4.9 EDMA Interrupts

The EDMA interrupts are divided into 2 categories: transfer completion interrupts and error interrupts.

There are nine region interrupts, eight shadow regions and one global region. The transfer completion interrupts are listed in Table 16-66. The transfer completion interrupts and the error interrupts from the transfer controllers are all routed to the device interrupt controllers INTCs through the inputs of the IRQ_CROSSBAR module.

Table 16-66. EDMA Transfer Completion Interrupts

Name	Description
EDMA_TPCC_INT0	EDMA_TPCC Transfer Completion Interrupt Shadow Region 0
EDMA_TPCC_INT1	EDMA_TPCC Transfer Completion Interrupt Shadow Region 1
EDMA_TPCC_INT2	EDMA_TPCC Transfer Completion Interrupt Shadow Region 2
EDMA_TPCC_INT3	EDMA_TPCC Transfer Completion Interrupt Shadow Region 3
EDMA_TPCC_INT4	EDMA_TPCC Transfer Completion Interrupt Shadow Region 4
EDMA_TPCC_INT5	EDMA_TPCC Transfer Completion Interrupt Shadow Region 5
EDMA_TPCC_INT6	EDMA_TPCC Transfer Completion Interrupt Shadow Region 6
EDMA_TPCC_INT7	EDMA_TPCC Transfer Completion Interrupt Shadow Region 7

Table 16-67. EDMA Error Interrupts

Name	Description
EDMA_TPCC_ERRINT	EDMA_TPCC Error Interrupt
EDMA_TPCC_MPINT	EDMA_TPCC Memory Protection Interrupt
EDMA_TC0_ERRINT	TC0 Error Interrupt
EDMA_TC1_ERRINT	TC1 Error Interrupt

16.2.4.9.1 Transfer Completion Interrupts

The EDMA_TPCC is responsible for generating transfer completion interrupts to the CPU(s) (and other EDMA masters). The EDMA generates a single completion interrupt per shadow region, as well as one for the global region on behalf of all 64 channels. The various control registers and bit fields facilitate EDMA interrupt generation.

The software architecture must either use the global interrupt or the shadow interrupts, but not both.

The transfer completion code EDMA_TPCC_OPT_n[17:12] TCC value is directly mapped to the bits of the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH.

For example, if EDMA_TPCC_OPT_n[17:12] TCC = 10 0001b, EDMA_TPCC_IPRH[1] is set after transfer completion, and results in interrupt generation to the CPU(s) if the completion interrupt is enabled for the CPU. See Section 16.2.4.9.1.1 *Enabling Transfer Completion Interrupts* for details about enabling EDMA transfer completion interrupts.

When a completion code is returned (as a result of early or normal completions), the corresponding bit in EDMA_TPCC_IPR / EDMA_TPCC_IPRH registers is set if transfer completion interrupt (final/intermediate) is enabled in the channel options parameter (OPT) for a PaRAM set associated with the transfer.

Table 16-68. Transfer Complete Code (TCC) to EDMA_TPCC Interrupt Mapping

TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPR Bit Set	TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPRH Bit Set(1)
0	EDMA_TPCC_IPR[0]	20h	EDMA_TPCC_IPR[32] / EDMA_TPCC_IPRH[0]
1	EDMA_TPCC_IPR[1]	21h	EDMA_TPCC_IPR[33] / EDMA_TPCC_IPRH[1]
2h	EDMA_TPCC_IPR[2]	22h	EDMA_TPCC_IPR[34] / EDMA_TPCC_IPRH[2]
3h	EDMA_TPCC_IPR[3]	23h	EDMA_TPCC_IPR[35] / EDMA_TPCC_IPRH[3]
4h	EDMA_TPCC_IPR[4]	24h	EDMA_TPCC_IPR[36] / EDMA_TPCC_IPRH[4]
...	...	...	...
1Eh	EDMA_TPCC_IPR[30]	3Eh	EDMA_TPCC_IPR[62] / EDMA_TPCC_IPRH[30]
1Fh	EDMA_TPCC_IPR[31]	3Fh	EDMA_TPCC_IPR[63] / EDMA_TPCC_IPRH[31]

(1) Bit fields EDMA_TPCC_IPR [32-63] correspond to bits 0 to 31 in EDMA_TPCC_IPRH, respectively.

The transfer completion code (TCC) can program to any value for a DMA/QDMA channel. A direct relation between the channel number and the transfer completion code value does not need to exist. This allows multiple channels having the same transfer completion code value to cause a CPU to execute the same interrupt service routine (ISR) for different channels.

If the channel is used in the context of a shadow region and it intends for the shadow region interrupt to be asserted, then ensure that the bit corresponding to the TCC code is enabled in EDMA_TPCC_IER / EDMA_TPCC_IERH and in the corresponding shadow region's DMA region access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k).

Interrupt generation can be enabled at either final transfer completion or intermediate transfer completion, or both. Consider channel *m* as an example.

- If the final transfer interrupt (EDMA_TPCC_OPT_n[20] TCINTEN = 1) is enabled, the interrupt occurs after the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).
- If the intermediate transfer interrupt (EDMA_TPCC_OPT_n[21] ITCINTEN = 1) is enabled, the interrupt occurs after every transfer request, except the last TR of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion interrupts (EDMA_TPCC_OPT_n[20] TCINTEN = 1, and EDMA_TPCC_OPT_n[21] ITCINTEN = 1) are enabled, then the interrupt occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 16-69 shows the number of interrupts that occur in different synchronized scenarios. Consider channel 31, programmed with ABCNT_n[15:0] ACNT = 3, EDMA_TPCC_ABCNT_n[31:16] BCNT = 4, EDMA_TPCC_CCNT_n[15:0]CCNT = 5, and EDMA_TPCC_OPT_n[17:12] TCC = 30.

Table 16-69. Number of Interrupts

Options	A-Synchronized	AB-Synchronized
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 0	1 (Last TR)	1 (Last TR)
EDMA_TPCC_OPT_n[20] TCINTEN = 0, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	19 (All but the last TR)	4 (All but the last TR)

Table 16-69. Number of Interrupts (continued)

Options	A-Synchronized	AB-Synchronized
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	20 (All TRs)	5 (All TRs)

16.2.4.9.1.1 Enabling Transfer Completion Interrupts

For the EDMA channel controller to assert a transfer completion to the external environment, the interrupts must be enabled in the EDMA_TPCC. This is in addition to setting up the EDMA_TPCC_OPT_n[20] TCINTEN and EDMA_TPCC_OPT_n[21] ITCINTEN bits of the associated PaRAM set.

The EDMA channel controller has interrupt enable registers EDMA_TPCC_IER / EDMA_TPCC_IERH and each bit location in EDMA_TPCC_IER / EDMA_TPCC_IERH serves as a primary enable for the corresponding interrupt pending registers EDMA_TPCC_IPR / EDMA_TPCC_IPRH.

All of the interrupt registers (EDMA_TPCC_IER, EDMA_TPCC_IESR, EDMA_TPCC_IECR, and EDMA_TPCC_IPR) are either manipulated from the global DMA channel region, or by the DMA channel shadow regions. The shadow regions provide a view to the same set of physical registers that are in the global region.

The EDMA channel controller has a hierarchical completion interrupt scheme that uses a single set of interrupt pending registers EDMA_TPCC_IPR / EDMA_TPCC_IPRH and single set of interrupt enable registers EDMA_TPCC_IER / EDMA_TPCC_IERH. The programmable DMA region access enable registers EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k provides a second level of interrupt masking. The global region interrupt output is gated based on the enable mask that is provided by EDMA_TPCC_IER / EDMA_TPCC_IERH, see Figure 16-27

The region interrupt outputs are gated by EDMA_TPCC_IER and the specific EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k associated with the region.

Figure 16-27 shows the Interrupt diagram of the EDMA controller.

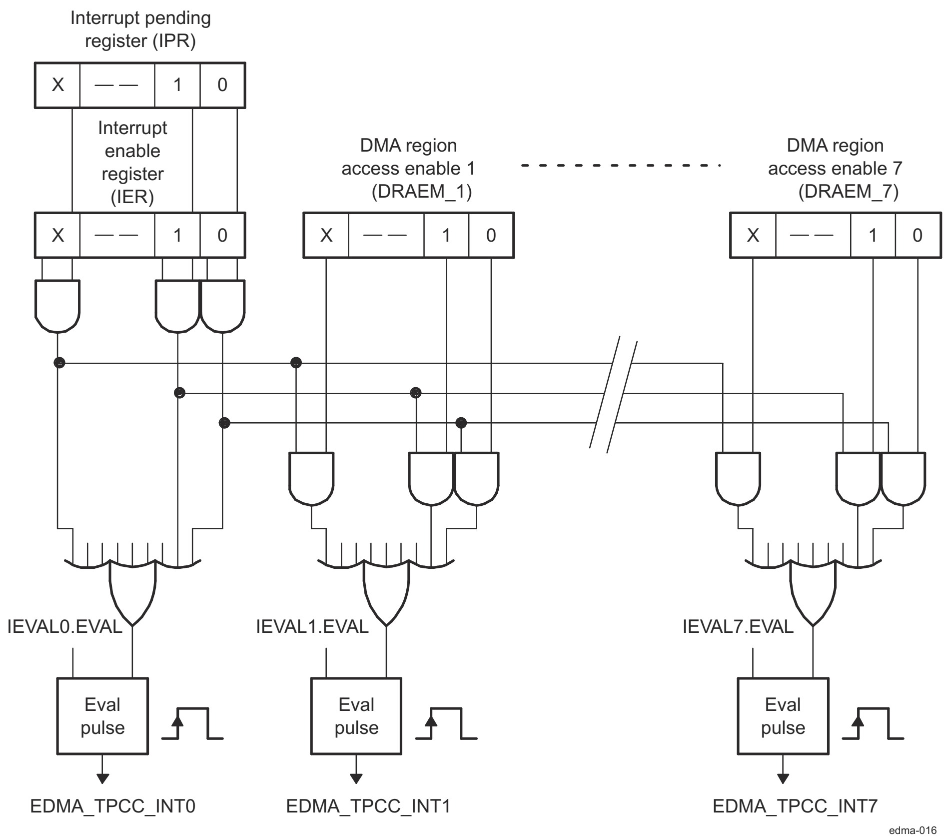


Figure 16-27. Interrupt Diagram

The EDMA_TPCC generates the transfer completion interrupts that are associated with each shadow region, the following conditions must be true:

- EDMA_TPCC_INT0: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_0[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_0[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_0[31] E63)
- EDMA_TPCC_INT1: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_1[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_1[1] E1) | …| (EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_1[31] E63)
- EDMA_TPCC_INT2: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_2[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_2[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_2[31] E63)....
- Up to EDMA_TPCC_INT7: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_7[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_7[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_7[31] E63)

Note

The EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k for all regions are expected to be set up at system initialization and to remain static for an extended period of time. The interrupt enable registers are used for dynamic enable/disable of individual interrupts.

Because there is no relation between the EDMA_TPCC_OPT_n[17:12] TCC value and the DMA/ QDMA channel, it is possible, the DMA channel 0 to have the EDMA_TPCC_OPT_n[17:12] TCC = 63 in its associated PaRAM set. This mean that if a transfer completion interrupt is enabled (EDMA_TPCC_OPT_n[20] TCINTEN or EDMA_TPCC_OPT_n[21] ITCINTEN is set), then based on the TCC value, EDMA_TPCC_IPRH[31] E63 is set up on completion. For proper channel operations and interrupt generation using the shadow region map - program the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k that is associated with the shadow region to have read/write access to both bit 0 (corresponding to channel 0) and bit 63 (corresponding to EDMA_TPCC_IPRH bit that is set upon completion).

16.2.4.9.1.2 Clearing Transfer Completion Interrupts

Transfer completion interrupts that are latched to the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH) are cleared by writing a 1 to the corresponding bit in the interrupt pending clear register (EDMA_TPCC_ICR / EDMA_TPCC_ICRH). For example, a write of 1 to EDMA_TPCC_ICR[0] E0 clears a pending interrupt in EDMA_TPCC_IPR[0] E0.

If an incoming transfer completion code TCC (EDMA_TPCC_OPT_n[17:12] TCC) gets latched to a bit in EDMA_TPCC_IPR / EDMA_TPCC_IPRH, then additional bits that get set due to a subsequent transfer completion does not result in asserting the EDMA_TPCC completion interrupt. In order for the completion interrupt to be pulsed, the required transition is from a state where no enabled interrupts are set to a state where at least one enabled interrupt is set.

16.2.4.9.2 EDMA Interrupt Servicing

Upon completion of a transfer (early or normal completion), the EDMA channel controller sets the appropriate bit in the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH), as the transfer completion codes specify. If the completion interrupts are appropriately enabled, then the CPU enters the interrupt service routine (ISR) when the completion interrupt is asserted.

After servicing the interrupt, the ISR should clear the corresponding bit in EDMA_TPCC_IPR/ EDMA_TPCC_IPRH, thereby enabling recognition of future interrupts. The EDMA_TPCC only asserts additional completion interrupts when all EDMA_TPCC_IPR / EDMA_TPCC_IPRH bits clear.

When one interrupt is serviced many other transfer completions may result in additional bits being set in EDMA_TPCC_IPR / EDMA_TPCC_IPRH, thereby resulting in additional interrupts. Each of the bits in EDMA_TPCC_IPR / EDMA_TPCC_IPRH may need different types of service therefore, the ISR must check all pending interrupts and continue until all of the posted interrupts are serviced appropriately.

Examples of pseudo code for a CPU interrupt service routine for an EDMA_TPCC completion interrupt are shown in Example 16-2 and Example 16-3.

The ISR routine in Example 16-2 is more exhaustive and incurs a higher latency.

Example 16-2. Interrupt Servicing

The pseudo code:

1. Reads the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH.
2. Performs the operations needed.
3. Writes to the interrupt pending clear register EDMA_TPCC_ICR / EDMA_TPCC_ICRH to clear the corresponding EDMA_TPCC_IPR / EDMA_TPCC_IPRH bit(s).
4. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH again:

a. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is not equal to 0, repeat from step 2 (implies occurrence of new event between step 2 to step 4).
b. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is equal to 0, assure that all of the enabled interrupts are inactive.

Note

An event may occur during step 4 while the EDMA_TPCC_IPR / EDMA_TPCC_IPRH bits are read as 0 and the application is still in the interrupt service routine. If this happens, a new interrupt is recorded in the device interrupt controller and a new interrupt generates as soon as the application exits in the interrupt service routine.

16.2.4.9.3

Example 16-3 is less rigorous, with less burden on the software in polling for set interrupt bits, but can occasionally cause a race condition as mentioned above.

Example 16-3. Interrupt Servicing

If any enabled and pending (possibly lower priority) interrupts are left, force the interrupt logic to reassert the interrupt pulse by setting the EDMA_TPCC_IEVAL[0] EVAL bit in the interrupt evaluation register.

The pseudo code is as follows:

1. Enters ISR.
2. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH.
3. For the condition that is set in EDMA_TPCC_IPR / EDMA_TPCC_IPRH:
   a. Service interrupt as the application requires.
   b. Clear the bit for serviced conditions (others may still be set, and other transfers may have resulted in returning the TCC to EDMA_TPCC after step 2).
4. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH prior to exiting the ISR:
   a. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is equal to 0, then exit the ISR.
   b. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is not equal to 0, then set EDMA_TPCC_IEVAL so that upon exit of ISR, a new interrupt triggers if any enabled interrupts are still pending.

16.2.4.9.4 Interrupt Evaluation Operations

The EDMA_TPCC has interrupt evaluate registers EDMA_TPCC_IEVAL that exist in the global region and in each shadow region. The registers in the shadow region are the only registers in the DMA channel shadow region memory map that are not affected by the settings for the DMA region access enable registers EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k. Writing a 1 to the EDMA_TPCC_IEVAL[0] EVAL bit in the registers that are associated with a particular shadow region results in pulsing the associated region interrupt (global or shadow), if any enabled interrupt (via EDMA_TPCC_IER / EDMA_TPCC_IERH) is still pending EDMA_TPCC_IPR / EDMA_TPCC_IPRH. This register assures that the CPU does not miss the interrupts (or the EDMA master associated with the shadow region) if the software architecture chooses not to use all interrupts. Refer to Example 16-3 about the use of EDMA_TPCC_IEVAL in the EDMA interrupt service routine (ISR).

Similarly an error evaluation register EDMA_TPCC_EEVAL exists in the global region. Writing a 1 to the EDMA_TPCC_EEVAL[0] EVAL bit causes the pulsing of the error interrupt if any pending errors are in EDMA_TPCC_EMR / EDMA_TPCC_EMRH, EDMA_TPCC_QEMR, or EDMA_TPCC_CCERR. See Section 16.2.4.9.5 *Error Interrupts* for additional information regarding error interrupts.

Note

While using EDMA_TPCC_IEVAL for shadow region completion interrupts, check that the EDMA_TPCC_IEVAL operated upon is from that particular shadow region memory map.

16.2.4.9.5 Error Interrupts

The EDMA_TPCC error registers provide the capability to differentiate error conditions (event missed, threshold exceed, etc.). Additionally, setting the error bits in these registers results in asserting the EDMA_TPCC error interrupt. If the EDMA_TPCC error interrupt is enabled in the device interrupt controller(s), then it allows the CPU(s) to handle the error conditions.

The EDMA_TPCC has a single error interrupt (EDMA_TPCC_ERRINT) that is asserted for all EDMA_TPCC error conditions. There are four conditions that cause the error interrupt:

- DMA missed events: for all 64 DMA channels. DMA missed events are latched in the event missed registers EDMA_TPCC_EMR / EDMA_TPCC_EMRH.
- QDMA missed events: for all 8 QDMA channels. QDMA missed events are latched in the QDMA event missed register EDMA_TPCC_QEMR.
- Threshold exceed: for all event queues. These are latched in EDMA_TPCC error register EDMA_TPCC_CCERR.
- TCC error: for outstanding transfer requests that are expected to return completion code EDMA_TPCC_OPT_n[22] TCCHEN or EDMA_TPCC_OPT_n[23] TCINTEN bit is set to 1, exceeding the maximum limit of 63. This is also latched in the EDMA_TPCC error register EDMA_TPCC_CCERR.

Figure 16-28 illustrates the EDMA_TPCC error interrupt generation operation.

If any of the bits are set in the error registers due to any error condition, the EDMA_TPCC_ERRINT is always asserted, as there are no enables for masking these error events. Similar to transfer completion interrupts (EDMA_TPCC_INT), the error interrupt also only pulses when the error interrupt condition transitions from no errors being set to at least one error being set. If additional error events are latched prior to the original error bits clearing, the EDMA_TPCC does not generate additional interrupt.

To reduce the burden on the software, there is an error evaluate register EDMA_TPCC_EEVAL that allows re-evaluation of pending set error events/bits, similar to the interrupt evaluate register EDMA_TPCC_IEVAL. Unlike the EDMA_TPCC_IEVAL functionality, the EDMA_TPCC_EEVAL register must be written with ‘1’ after any error interrupts are serviced (even when all pending errors are cleared) in order for subsequent errors to trigger a new interrupt.

Note

It is good practice to enable the error interrupt in the device interrupt controller and to associate an interrupt service routine with it to address the various error conditions appropriately. Doing so puts less burden on the software (polling for error status), it provides a good debug mechanism for unexpected error conditions.

16.2.4.9.6

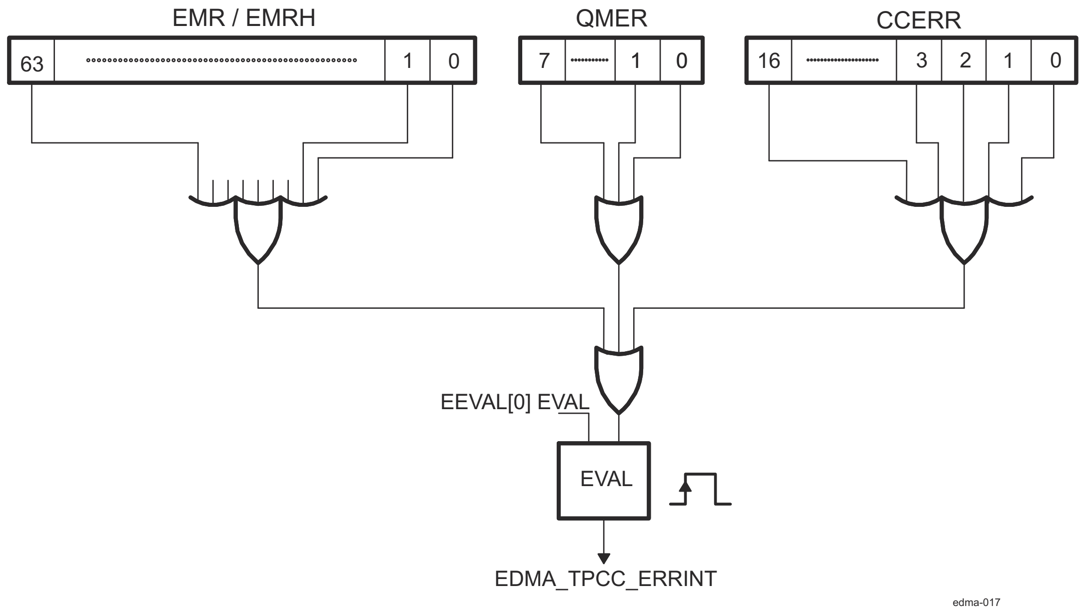


Figure 16-28. Error Interrupt Operation

16.2.4.10 Memory Protection

The EDMA channel controller supports two kinds of memory protection: active and proxy.

16.2.4.10.1 Active Memory Protection

Active memory protection is a feature that allows or prevents read and write accesses to the EDMA_TPCC registers. Active memory protection is achieved by a set of memory protection permissions attribute EDMA_TPCC_MPPAN_k registers.

The EDMA_TPCC register map is divided into three categories:

- a global region.
- a global channel region.
- eight shadow regions.

Each shadow region consists of the respective shadow region registers and the associated PaRAM. For more detailed information regarding the contents of a shadow region, refer to Table 16-80 *EDMA_TPCC Registers Mapping Summary*.

Each of the eight shadow regions has an associated EDMA_TPCC_MPPAN_k registers that defines the specific requestor(s) and types of requests that are allowed to the regions resources.

The global channel region is also protected with a memory-mapped register EDMA_TPCC_MPPAG. The EDMA_TPCC_MPPAG applies to the global region and to the global channel region, except the other EDMA_TPCC_MPPAN_k registers themselves.

Table 16-70 shows the accesses that are allowed or not allowed to the EDMA_TPCC_MPPAG and EDMA_TPCC_MPPAN_k. The active memory protection uses the EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID attributes of the EDMA peripheral modules. TheEDMA_TPCC_OPT_n[31] PRIV is the privilege level (i.e., user vs. supervisor).

The EDMA_TPCC_OPT_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an EDMA peripheral modules.

Table 16-70. Allowed Accesses

Access	Supervisor	User
Read	Yes	Yes
Write	Yes	No

Table 16-71 describes the EDMA_TPCC_MPPAN_k register mapping for the shadow regions (which includes shadow region registers and PaRAM addresses).

The region-based EDMA_TPCC_MPPAN_k registers are used to protect accesses to the DMA shadow regions and the associated region PaRAM. Because there are eight regions, there are eight EDMA_TPCC_MPPAN_k region registers (MPPA[0-7]).

Table 16-71. MPPA Registers to Region Assignment

Register	Registers Protect	Address Range	PaRAM Protect[(1)]	Address Range
EDMA_TPCC_MPPAG	Global Range	0000h-1FFCh	N/A	N/A
EDMA_TPCC_MPPAN_k. MPPAN_0	DMA Shadow 0	2000h-21FCh	1st octant	4000h-47FCh
MPPAN_1	DMA Shadow 1	2200h-23FCh	2nd octant	4800h-4FFCh
MPPAN_2	DMA Shadow 2	2400h-25FCh	3rd octant	5000h-57FCh
MPPAN_3	DMA Shadow 3	2600h-27FCh	4th octant	5800h-5FFCh
MPPAN_4	DMA Shadow 4	2800h-29FCh	5th octant	6000h-67FCh
MPPAN_5	DMA Shadow 5	2A00h-2BFCh	6th octant	6800h-6FFCh
MPPAN_6	DMA Shadow 6	2C00h-2DFCh	7th octant	7000h-77FCh

Table 16-71. MPPA Registers to Region Assignment (continued)

Register	Registers Protect	Address Range	PaRAM Protect(1)	Address Range
MPPAN_7	DMA Shadow 7	2E00h-2FFCh	8th octant	7800h-7FFCh

(1) The PARAM region is divided into 8 regions referred to as an octant.

Example *Access denied.*

Write access to shadow region 7's event enable set register EDMA_TPCC_EESR:

1. The original value of the event enable register EDMA_TPCC_EER at address offset 0x1020 is 0x0.
2. The EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[7] NS is set to prevent user level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 0, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 0), but it allows supervisor level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1) with a privilege ID of 0. (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0 attempts to perform a user-level write of a value of 0xFF00FF00 to shadow region 7's event enable set register EDMA_TPCC_EESR at address offset 0x2E30.

Note

The EDMA_TPCC_EER is a read-only register and the only way that write to it is by writing to the EDMA_TPCC_EESR. There is only one physical register for EDMA_TPCC_EER, EDMA_TPCC_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 0, though the privilege ID of the write access is set to 0, the access is not allowed and the EDMA_TPCC_EER is not written too.

Table 16-72. Example Access Denied

Register	Value	Description
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Value in EDMA_TPCC_EER to begin with.
EDMA_TPCC_EESR (offset 0x2E30)	0xFF00 FF00 ↓	Value attempted to be written to shadow region 7's EDMA_TPCC_EESR. This is done by an EDMA connected device module with a privilege level of User and Privilege ID of 0.
EDMA_TPCC_MPPAN_k (offset 0x082C)	0x0000 04B0	Memory Protection Filter EDMA_TPCC_MPPAN_k[10] AID0 = 1, EDMA_TPCC_MPPAN_k[1] UW = 0, EDMA_TPCC_MPPAN_k[2] UR = 0, EDMA_TPCC_MPPAN_k[4] SW = 1, EDMA_TPCC_MPPAN_k[5] SR = 1.
	X	Access Denied
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Final value of EDMA_TPCC_EER

Example *Access Allowed*

Write access to shadow region 7's event enable set register EDMA_TPCC_EESR:

1. The original value of the event enable register EDMA_TPCC_EER at address offset 0x1020 is 0x0.
2. The EDMA_TPCC_MPPAN_k.EDMA_TPCC_MPPAN_7 is set to allow user-level accesses (EDMA_TPCC_MPPAN_k.EDMA_TPCC_MPPAN_7[1] UW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 1) and supervisor-level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1) with a privilege ID of 0. (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0, attempts to perform a user-level write of a value of 0xABCD0123 to shadow region 7's event enable set register EDMA_TPCC_EESR at address offset 0x2E30.

Note

The EDMA_TPCC_EER is a read-only register and the only way that write to it is by writing to the EDMA_TPCC_EESR. There is only one physical register for EDMA_TPCC_EER, EDMA_TPCC_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 1 and EDMA_TPCC_MPPAN_k. MPPAN_7[10] AID0 = 1, the user-level write access is allowed.
5. The accesse to shadow region registers are masked by their respective EDMA_TPCC_DRAEM_k register. In this example, the EDMA_TPCC_DRAEM_k. EDMA_TPCC_DRAEM_7 is set of 0x9FF00FC2.
6. The value finally written to EDMA_TPCC_EER is 0x8BC00102.

Table 16-73. Example Access Allowed

Register	Value	Description
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Value in EER to begin with.
EDMA_TPCC_EESR (offset 0x2E30)	0xFF00 FF00	Value attempted to be written to shadow region 7's EESR. This is done by an EDMA peripheral module with a privilege level of User and Privilege ID of 0.
EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7 (offset 0x082C)	0x0000 04B3	Memory Protection Filter EDMA_TPCC_MPPAN_k[10] AID = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1.
	√ ↓	Access allowed.
EDMA_TPCC_DRAEM_k. EDMA_TPCC_DRAEM_7 (offset 0x0378)	0x9FF0 0FC2 ↓	DMA Region Access Enable Filter
EDMA_TPCC_EESR (offset 0x2E30)	0x8BC0 0102 ↓	Value written to shadow region 7's EESR. This is done by an EDMA peripheral module with a privilege level of User and a Privilege ID of 0.
EDMA_TPCC_EER (offset 0x1020)	↓ 0xBC0 0102	Final value of EER.

16.2.4.10.2 Proxy Memory Protection

Proxy memory protection allows an EDMA transfer programmed by a given peripheral module connected to EDMA, to have its permissions travel with the transfer through the EDMA_TPTC. The permissions travel along with the read transactions to the source and the write transactions to the destination endpoints. The EDMA_TPCC_OPT_n[31] PRIV bit and EDMA_TPCC_OPT_n[27:24] PRIVID bit is set with the peripheral module's PRIV value and PRIVID values, respectively, when any part of the PaRAM set is written.

The EDMA_TPCC_OPT_n[31] PRIV is the privilege level (i.e., user vs. supervisor). The EDMA_TPCC_OPT_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an peripheral module connected to EDMA.

These options are part of the TR that are submitted to the transfer controller. The transfer controller uses the above values on their respective read and write command bus so that the target endpoints can perform memory protection checks based on these values.

Consider a parameter set that is programmed by a CPU in user privilege level for a simple transfer with the source buffer on an L2 page and the destination buffer on an L1D page. The EDMA_TPCC_OPT_n[31] PRIV is 0 for user-level and the CPU has a EDMA_TPCC_OPT_n[27:24] PRIVID to 0.

The PaRAM set is shown in Figure 16-29.

Figure 16-29. PaRAM Set Content for Proxy Memory Protection Example

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0007h		Channel Options Parameter (OPT)	
009F 0000h		Channel Source Address (SRC)	
0001h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
00F0 7800h		Channel Destination Address (DST)	
0001h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT_n) Content

31	30	29-28	27-24	23	22	21	20	19-18	17-16
0	0	00	0000	0	0	0	1	00	00
PRIV	Rsvd	Rsvd	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15-12	11	10-8	7-4	3	2	1	0
0000	0	000	0000	0	1	1	1
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

edma-018

Figure 16-30. Channel Options Parameter (OPT) Example

The EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID information travels along with the read and write requests that are issued to the source and destination memories.

For example, if the access attributes that are associated with the L2 page with the source buffer only allow supervisor read, write accesses EDMA_TPCC_MPPAN_k[4] SW and EDMA_TPCC_MPPAN_k[5] SR, the user-level read request above is refused. Similarly, if the access attributes that are associated with the L1D page with the destination buffer only allow supervisor read and write accesses (EDMA_TPCC_MPPAN_k[4] SW, EDMA_TPCC_MPPAN_k[5] SR), the user-level write request above is refused. For the transfer to succeed, the source and destination pages must have user-read and user-write permissions, respectively, along with allowing accesses from a PRIVID = 0.

Because the privilege level and privilege identification travel with the read and write requests, EDMA acts as a proxy.

Figure 16-31 illustrates the propagation of EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID at the boundaries of all the interacting entities (CPU, EDMA_TPCC, EDMA_TPTCs, and slave memories).

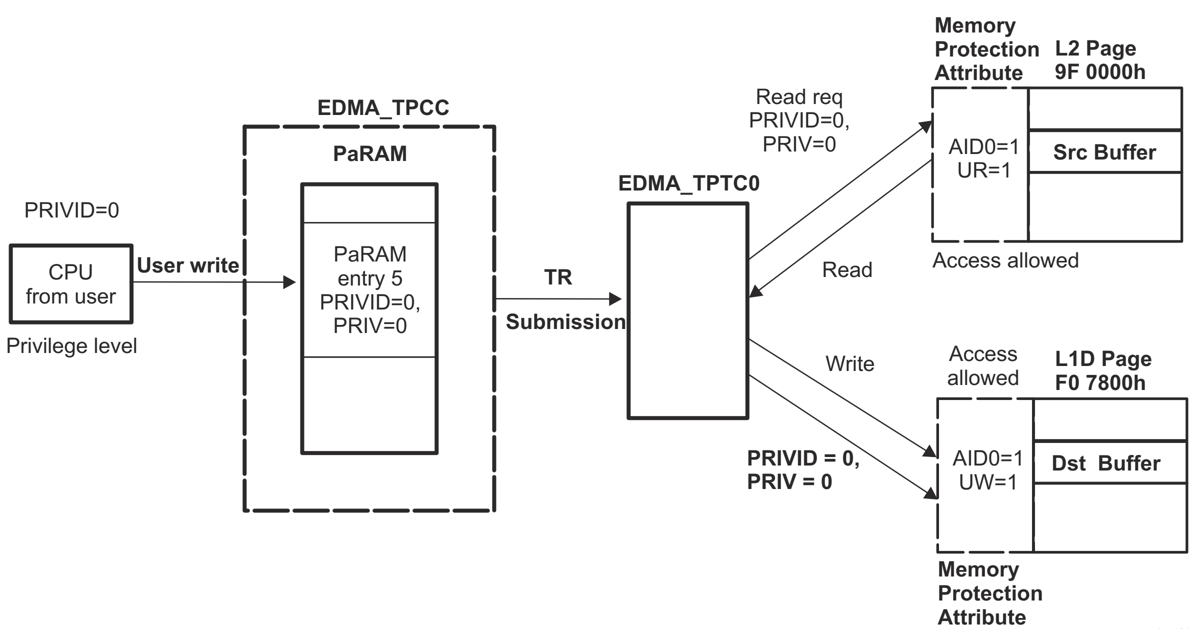


Figure 16-31. Proxy Memory Protection Example

16.2.4.11 Event Queue(s)

Event queues are a part of the EDMA channel controller. Event queues form the interface between the event detection logic in the EDMA_TPCC and the transfer request (TR) submission logic of the EDMA_TPCC. Each queue is 16 entries deep. Each event queue can queue a maximum of 16 events. If there are more than 16 events, then the events that cannot find a place in the event queue remain set in the associated event register and the CPU does not stall.

There are two event queues for the device: Queue0, Queue1. Events in Queue0 result in submission of its associated transfer requests (TRs) to TC0. The transfer requests that are associated with events in Queue1 are submitted to TC1.

An event that wins prioritization against other DMA and/or QDMA pending events is placed at the tail of the appropriate event queue. Each event queue is serviced in FIFO order. Once the event reaches the head of its queue and the corresponding transfer controller is ready to receive another TR, the event is de-queued and the PaRAM set corresponding to the de-queued event is processed and submitted as a transfer request packet (TRP) to the associated EDMA transfer controller.

Queue0 has highest priority and Queue1 has the lowest priority, if Queue0 and Queue1 both have at least one event entry and if both TC0 and TC1 can accept transfer requests, then the event in Queue0 is de-queued first and its associated PaRAM set is processed and submitted as a transfer request (TR) to TC0.

Refer to Section 16.2.4.11.4 for system-level performance considerations. All of the event entries in all of the event queues are software readable (not writeable) by accessing the event entry registers EDMA_TPCC_Q0E_p and EDMA_TPCC_Q1E_p. Each event entry register characterizes the queued event in terms of the type of event (manual, event, chained or auto-triggered) and the event number. Refer to Section 16.2.7.2.2.1 *EDMA_TPCC Register Description* for Q0E_p / Q1E_p descriptions of the bit fields.

16.2.4.11.1 DMA/QDMA Channel to Event Queue Mapping

Each of the 64 DMA channels and eight QDMA channels are programmed independently to map to a specific queue, using the DMA queue number register EDMA_TPCC_DMAQNUMN_k and the QDMA queue number register EDMA_TPCC_QDMAQNUM. The mapping of DMA/QDMA channels is critical to achieving the desired performance level for the EDMA and most importantly, in meeting real-time deadlines. Refer to Section 16.2.4.11.4 *System-level Performance Considerations*.

Note

If an event is ready to be queued and both the event queue and the EDMA transfer controller that is associated to the event queue are empty, then the event bypasses the event queue, and moves the PaRAM processing logic, and eventually to the transfer request submission logic for submission to the EDMA_TPTC. In this case, the event is not logged in the event queue status registers.

16.2.4.11.2 Queue RAM Debug Visibility

There are two event queues and each queue has 16 entries. These 16 entries are managed in a circular FIFO. There is a queue status register EDMA_TPCC_QSTATN_i associated with each queue. These along with all of the 16 entries per queue can be read via registers EDMA_TPCC_QSTATN_i and Q0E_p / Q1E_p, respectively.

These registers provide user visibility.

The event queue entry register (Q*x*E*y* Q0E_p / Q1E_p) uniquely identifies the specific event type (event-triggered, manually-triggered, chain-triggered, and QDMA events) along with the event number (for all DMA/ QDMA event channels) that are in the queue or have been de-queued (passed through the queue).

Each of the 16 entries in the event queue are read using the EDMA_TPCC memory-mapped register. To see the history of the last 16 TRs that have been processed by the EDMA on a given queue, read the event queue registers. This provides user/software visibility and is helpful for debugging real-time issues (typically post-mortem), involving multiple events and event sources.

The queue status register (QSTAT*n* EDMA_TPCC_QSTATN_i) includes fields for the start pointer EDMA_TPCC_QSTATN_i[3:0] STRTPTR which provides the offset to the head entry of an event. It also includes a field called EDMA_TPCC_QSTATN_i[12:8] NUMVAL that provides the total number of valid entries residing in the event queue at a given instance of time. The EDMA_TPCC_QSTATN_i[3:0] STRTPTR is used to index appropriately into the 16 event entries. EDMA_TPCC_QSTATN_i[12:8] NUMVAL number of entries starting from STRTPTR are indicative of events still queued in the respective queue. The remaining entry must be read to determine what's already de-queued and submitted to the associated transfer controller.

16.2.4.11.3 Queue Resource Tracking

The EDMA_TPCC event queue includes watermarking/threshold logic that allows to keep track of maximum usage of all event queues. This is useful for debugging real-time deadline violations that may result from head-of-line blocking on a given EDMA event queue.

The maximum number of events are programed that the queue up in an event queue by programming the threshold value (between 0 to 15) in the queue watermark threshold A register EDMA_TPCC_QWMTHRA. The maximum queue usage is recorded actively in the watermark EDMA_TPCC_QSTATN_i[20:16] WM field of the queue status register, that keeps getting updated based on a comparison of number of valid entries, which is also visible in the EDMA_TPCC_QSTATN_i[12:8] NUMVAL bit and the maximum number of entries.

If the queue usage is exceeded, this status is visible in the EDMA_TPCC registers: the QTHRXCD*n* bits in the channel controller error register EDMA_TPCC_CCERR[7:0] and the EDMA_TPCC_QSTATN_i[24] THRXCD bit, where *n* stands for the event queue number. Any bits that are set in EDMA_TPCC_CCERR also generate an EDMA_TPCC error interrupt.

16.2.4.11.4 Performance Considerations

The main system bus infrastructure (L3) arbitrates bus requests from all of the masters (TCs, CPU(S), and other bus masters) to the shared slave resources (peripherals and memories).

The priorities of transfer requests (read and write commands) from the EDMA transfer controllers with respect to other masters within the device IRQ_CROSSBAR are programmed using the Control Module registers. The EDMA_TPCC_QUEPRI register has no affect.

Therefore, the priority of unloading queues has a secondary affect compared to the priority of the transfers as they are executed by the EDMA_TPTC (dictated by the priority set using the Control Module registers, refer to *Control Module Register Manual* in *Control Module* chapter).

16.2.4.12 EDMA Transfer Controller (EDMA_TPTC)

The EDMA channel controller is the user-interface of the EDMA and the EDMA transfer controller (EDMA_TPTC) is the data movement engine of the EDMA controller. The EDMA_TPCC submits transfer requests (TR) to the EDMA_TPTC and the EDMA_TPTC performs the data transfers dictated by the TR, so the EDMA_TPTC is a slave to the EDMA_TPCC.

16.2.4.12.1 Architecture Details

16.2.4.12.1.1 Command Fragmentation

The TC read and write controllers in conjunction with the source and destination register sets are responsible for issuing optimally-sized reads and writes to the slave endpoints. An optimally-sized command is defined by the transfer controller default burst size (DBS), which is defined in Section 16.2.4.12.5 *EDMA_TPTC Configuration*.

The EDMA_TPTC attempts to issue the largest possible command size as limited by the DBS value or the EDMA_TPCC_ABCNT_n[15:0] ACNT and EDMA_TPCC_ABCNT_n[31:16] BCNT value of the TR. EDMA_TPTC obeys the following rules:

- The read/write controllers always issue commands less than or equal to the DBS value.
- The first command of a 1D transfer command always aligns the address of subsequent commands to the DBS value.

Table 16-74 lists the TR segmentation rules that are followed by the EDMA_TPTC. In summary, if the EDMA_TPCC_ABCNT_n[15:0] ACNT value is larger than the DBS value, then the EDMA_TPTC breaks the

EDMA_TPCC_ABCNT_n[15:0] ACNT array into DBS-sized commands to the source/destination addresses. Each EDMA_TPCC_ABCNT_n[31:16] BCNT number of arrays are then serviced in succession.

For BCNT arrays of ACNT bytes (that is, a 2D transfer), if the EDMA_TPCC_ABCNT_n[15:0] ACNT value is less than or equal to the DBS value, then the TR may be optimized into a 1D-transfer in order to maximize efficiency. The optimization takes place if the EDMA_TPTC recognizes that the 2D-transfer is organized as a single dimension (EDMA_TPCC_ABCNT_n[15:0] ACNT == EDMA_TPCC_BIDX_n) and the ACNT value is a power of 2.

Table 16-74 lists conditions in which the optimizations are performed.

Table 16-74. Read/Write Command Optimization Rules

ACNT ≤ DBS	ACNT is power of 2	BIDX = ACNT	BCNT ≤ 1023	SAM/DAM = Increment	Description
Yes	Yes	Yes	Yes	Yes	Optimized
No	x	x	x	x	Not Optimized
x	No	x	x	x	Not Optimized
x	x	No	x	x	Not Optimized
x	x	x	No	x	Not Optimized
x	x	x	x	No	Not Optimized

16.2.4.12.1.2 TR Pipelining

TR pipelining refers to the ability of the source active set to proceed ahead of the destination active set. Essentially, the reads for a given TR may already be in progress while the writes of a previous TR may not have completed.

The number of outstanding TRs is limited by the number of destination FIFO register entries.

TR pipelining is useful for maintaining throughput on back-to-back small TRs. It minimizes the startup overhead because reads start in the background of a previous TR writes.

Example 16-4. Command Fragmentation (DBS = 64)

The pseudo code:

1. EDMA_TPTCn_PCNT[15:0] ACNT = 8, EDMA_TPTCn_PCNT[31:16] BCNT = 8, EDMA_TPTCn_PBIDX[15:0] SBIDX = 8, EDMA_TPTCn_PBIDX[31:16] DBIDX = 10, EDMA_TPTCn_PSRC[31:0] SADDR = 64, EDMA_TPTCn_SADST[31:0] DADDR = 191

   Read Controller: This is optimized from a 2D-transfer to a 1D-transfer such that the read side is equivalent to EDMA_TPTCn_PCNT[15:0] ACNT = 64, EDMA_TPTCn_PCNT[31:16] BCNT = 1.

   Cmd0 = 64 byte

   Write Controller: Because DBIDX != ACNT, it is not optimized.

   Cmd0 = 8 byte, Cmd1 = 8 byte, Cmd2 = 8 byte, Cmd3 = 8 byte, Cmd4 = 8 byte, Cmd5 = 8 byte, Cmd6 = 8 byte, Cmd7 = 8 byte.
2. EDMA_TPTCn_PCNT[15:0] ACNT=128, EDMA_TPTCn_PCNT[31:16] BCNT = 1, EDMA_TPTCn_PSRC[31:0] SADDR = 63, EDMA_TPTCn_SADST[31:0] DADDR = 513

   Read Controller: Read address is not aligned.

   Cmd0 = 1 byte, (now the SADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 63 bytes

   Write Controller: The write address is also not aligned.

   Cmd0 = 63 bytes, (now the DADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 1 byte

16.2.4.12.1.3 Performance Tuning

By default, reads are as issued as fast as possible. In some cases, the reads issued by the EDMA_TPTC could fill the available command buffering for a slave, delaying other (potentially higher priority) masters from successfully submitting commands to that slave. The rate at which read commands are issued by the EDMA_TPTC is controlled by the EDMA_TPTCn_RDRATE register. The EDMA_TPTCn_RDRATE register defines the number of cycles that the EDMA_TPTC read controller waits before issuing subsequent commands for a given TR, thus minimizing the chance of the EDMA_TPTC consuming all available slave resources. The EDMA_TPTCn_RDRATE[2:0] RDRATE value must be set to a relatively small value if the transfer controller is targeted for high priority transfers and to a higher value if the transfer controller is targeted for low priority transfers.

In contrast, the Write Interface does not have any performance turning knobs because writes always have an interval between commands as write commands are submitted along with the associated write data.

16.2.4.12.2 Memory Protection

The transfer controller plays an important role in handling proxy memory protection. There are two access properties associated with a transfer: for instance, the privilege id (system-wide identification assigned to a master) of the master initiating the transfer, and the privilege level (user versus supervisor) used to program the transfer. This information is maintained in the PaRAM set when it is programmed in the channel controller. When a TR is submitted to the transfer controller, this information is made available to the EDMA_TPTC and used by the EDMA_TPTC while issuing read and write commands. The read or write commands have the same privilege identification, and privilege level as that programmed in the EDMA transfer in the channel controller.

16.2.4.12.3 Error Generation

Errors are generated if enabled under three conditions:

- EDMA_TPTC detection of an error signaled by the source or destination address.
- Attempt to read or write to an invalid address in the configuration memory map.
- Detection of a constant addressing mode TR violating the constant addressing mode transfer rules (the source/destination addresses and source/destination indexes must be aligned to 32 bytes).

Either or all error types may be disabled. If an error bit is set and enabled, the error interrupt for the concerned transfer controller is generated.

16.2.4.12.4 Debug Features

The DMA program register set, DMA source active register set, and the destination FIFO register set are used to derive a brief history of TRs serviced through the transfer controller.

Additionally, the EDMA_TPTC status register EDMA_TPTCn_TCSTAT has dedicated bit fields to indicate the ongoing activity within different parts of the transfer controller:

- The EDMA_TPTCn_TCSTAT[1] SRCACTV bit indicates whether the source active set is active.
- The EDMA_TPTCn_TCSTAT[6:4] DSTACTV bit indicates the number of TRs resident in the destination register active set at a given instance.
- The EDMA_TPTCn_TCSTAT[0] PROGBUSY bit indicates whether a valid TR is present in the DMA program set.

Note

If the TRs are in progression, it must realize that there is a chance that the values read from the EDMA_TPTC status registers will be inconsistent since the EDMA_TPTC changes the values of these registers due to ongoing activities.

It is recommended that to ensure no additional submission of TRs to the EDMA_TPTC in order to facilitate ease of debug.

16.2.4.12.4.1 Destination FIFO Register Pointer

The destination FIFO register pointer is implemented as a circular buffer with the start pointer being EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR and a buffer depth of usually 2 or 4. The EDMA_TPTC maintains two important status details in EDMA_TPTCn_TCSTAT that are used during advanced debugging, if necessary. The EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR is a start pointer, the index to the head of the destination FIFO register. The EDMA_TPTCn_TCSTAT[6:4] DSTACTV is a counter for the number of valid (occupied) entries. These registers are used to get a brief history of transfers.

Examples of some register field values and their interpretation:

- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x0 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x0 implies that no TRs are stored in the destination FIFO register.
- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x1 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 1 and the second pending TR is read from the destination FIFO register entry 2.
- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x3 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 3 and the second pending TR is read from the destination FIFO register entry 0.

16.2.4.12.5 EDMA_TPTC Configuration

Table 16-75 provides the configuration of the individual EDMA transfer controllers present on the device. The DBS for each transfer controller is defined by Control Module register (CTRL_CORE_CONTROL_IO_1) settings.

Table 16-75. EDMA Transfer Controller Configurations

Name	TC0	TC1
EDMA_TPTCn_TCCFG[2:0] FIFOSIZE	1024 bytes	1024 bytes

Table 16-75. EDMA Transfer Controller Configurations (continued)

Name	TC0	TC1
EDMA_TPTCn_TCCFG[5:4] BUSWIDTH	16 bytes	16 bytes
EDMA_TPTCn_TCCFG[9:8] DSTREGDEPTH	4 entries	4 entries
DBS	Defined by CTRL_CORE_CONTROL_IO_1[9:8] TC0_DEFAULT_BURST_SIZE	Defined by CTRL_CORE_CONTROL_IO_1[13:12] TC1_DEFAULT_BURST_SIZE

16.2.4.13 Event Dataflow

This section summarizes the data flow of a single event, from the time the event is latched to the channel controller to the time the transfer completion code is returned. The following steps list the sequence of EDMA_TPCC activity:

1. Event is asserted from an external source (peripheral or external interrupt). This also is similar for a manually-triggered, chained-triggered, or QDMA-triggered event. The event is latched into the EDMA_TPCC_ER[31:0]E*n* / EDMA_TPCC_ERH[31:0] E*n* (or EDMA_TPCC_CER[31:0] E*n* / EDMA_TPCC_CERH[31:0] E*n*, EDMA_TPCC_ESR[31:0] E*n* / EDMA_TPCC_ESRH[31:0] E*n*, EDMA_TPCC_QER[7:0] E*n*) bit.
2. Once an event is prioritized and queued into the appropriate event queue, the EDMA_TPCC_SER[31:0] E*n* \ EDMA_TPCC_SERH[31:0] E*n* (or EDMA_TPCC_QSER[7:0] E*n*) bit is set to inform the event prioritization / processing logic to disregard this event since it is already in the queue. Alternatively, if the transfer controller and the event queue are empty, then the event bypasses the queue.
3. The EDMA_TPCC processing and the submission logic evaluates the appropriate PaRAM set and determines whether it is a non-null and non-dummy transfer request (TR).
4. The EDMA_TPCC clears the EDMA_TPCC_ER[31:0] E*n*/ EDMA_TPCC_ERH[31:0] E*n* (or EDMA_TPCC_CER[31:0] E*n* / EDMA_TPCC_CERH[31:0] E*n*, EDMA_TPCC_ESR[31:0]E*n* / EDMA_TPCC_ESRH[31:0] E*n*, EDMA_TPCC_QER[31:0] E*n*) bit and the EDMA_TPCC_SER[31:0] E*n* / EDMA_TPCC_SERH[31:0] E*n* bit as soon as it determines the TR is non-null. In the case of a null set, the EDMA_TPCC_SER[31:0] E*n* / EDMA_TPCC_SERH[31:0] E*n* bit remains set. It submits the non-null/non-dummy TR to the associated transfer controller. If the TR was programmed for early completion, the EDMA_TPCC immediately sets the interrupt pending register (EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC] - 32).
5. If the TR was programmed for normal completion, the EDMA_TPCC sets the interrupt pending register (EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC]) when the EDMA_TPTC informs the EDMA_TPCC about completion of the transfer (returns transfer completion codes).
6. The EDMA_TPCC programs the associated EDMA_TPTC's Program Register Set with the TR.
7. The TR is then passed to the Source Active set and the DST FIFO Register Set, if both the register sets are available.
8. The Read Controller processes the TR by issuing read commands to the source slave endpoint. The Read Data lands in the Data FIFO of the EDMA_TPTC*n*.
9. As soon as sufficient data is available, the Write Controller begins processing the TR by issuing write commands to the destination slave endpoint.
10. This continues until the TR completes and the EDMA_TPTC*n* then signals completion status to the EDMA_TPCC.

16.2.4.14 EDMA controller Prioritization

The EDMA controller has many implementation rules to deal with concurrent events/channels, transfers, etc. The following subsections detail various arbitration details whenever there might be occurrence of concurrent activity. Figure 16-32 shows the different places EDMA priorities come into play.

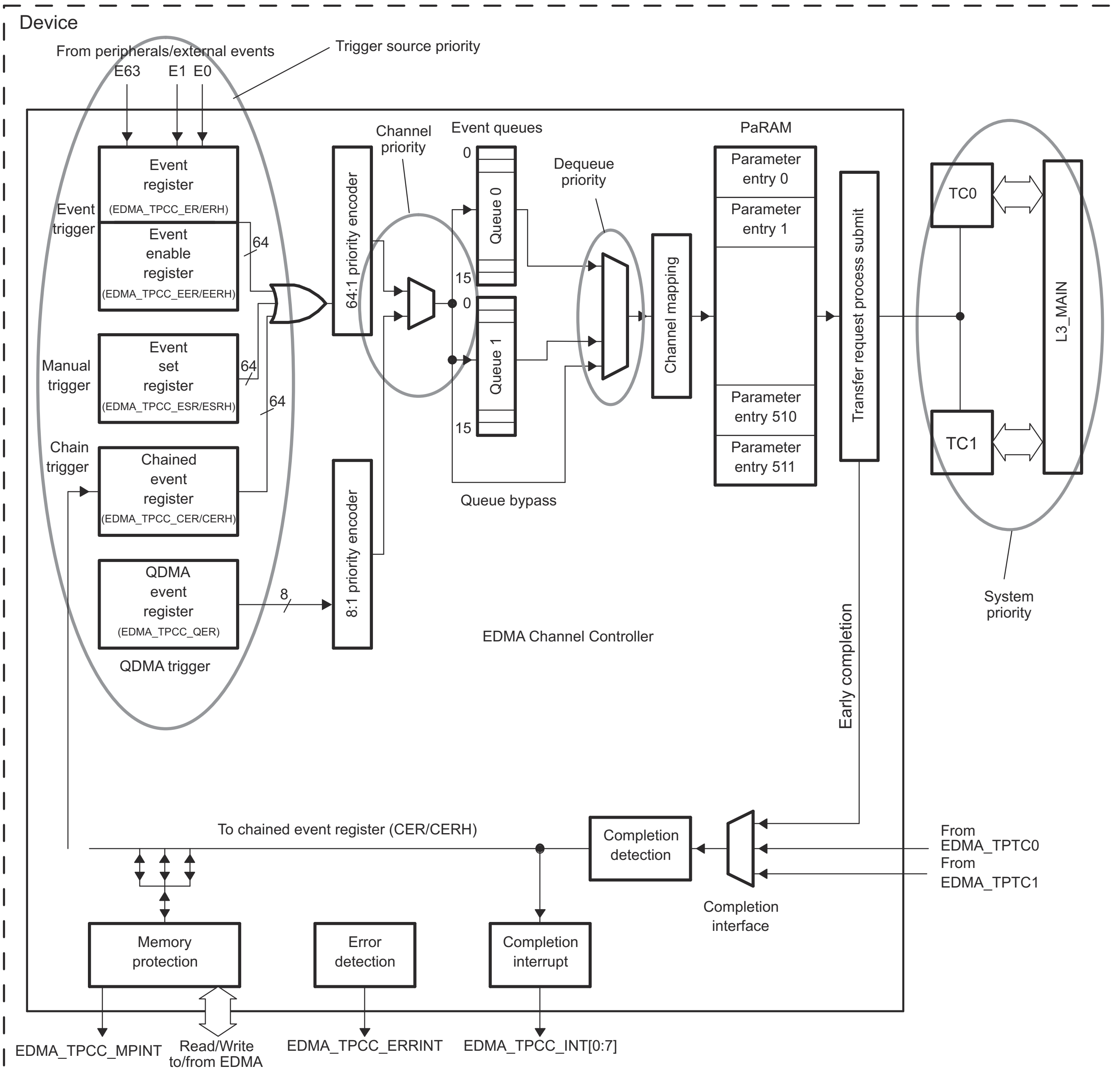


Figure 16-32. EDMA Prioritization

16.2.4.14.1 Channel Priority

The EDMA event registers EDMA_TPCC_ER and EDMA_TPCC_ERH capture up to 64 events, the QDMA event register EDMA_TPCC_QER captures QDMA events for all QDMA channels therefore, it is possible for events to occur simultaneously on the DMA/QDMA event inputs. For events arriving simultaneously, the event associated with the lowest channel number is prioritized for submission to the event queues (for DMA events, channel 0 has the highest priority and channel 63 has the lowest priority, for QDMA events, channel 0 has the highest priority and channel 7 has the lowest priority). This mechanism only sorts simultaneous events for submission to the event queues.

If a DMA and QDMA event occurs simultaneously, the DMA event always has prioritization against the QDMA event for submission to the event queues.

16.2.4.14.2 Trigger Source Priority

If a EDMA channel is associated with more than one trigger source (event trigger, manual trigger, and chain trigger), and if multiple events are set simultaneously for the same channel (EDMA_TPCC_ER[31:0] E*n* = 1, EDMA_TPCC_ESR[31:0] E*n* = 1 , EDMA_TPCC_CER[31:0] E*n* = 1) , then the EDMA_TPCC always services these events in the following priority order: event trigger (via EDMA_TPCC_ER) is higher priority than chain trigger (via EDMA_TPCC_CER) and chain trigger is higher priority than manual trigger (via EDMA_TPCC_ESR).

This implies that if for channel 0, both EDMA_TPCC_ER[0] E0 = 1 and EDMA_TPCC_CER[0] E0 = 1 at the same time, then the EDMA_TPCC_ER[0] E0 event is always queued before the EDMA_TPCC_CER[0] E0 event.

16.2.4.14.3 Dequeue Priority

The priority of the associated transfer request (TR) is further mitigated by which event queue is being used for event submission (dictated by EDMA_TPCC_DMAQNUMN_k and EDMA_TPCC_QDMAQNUM). For submission of a TR to the transfer request, events need to be de-queued from the event queues. Queue 0 has the highest dequeue priority and queue 1 the lowest.

16.2.4.15 EDMA Power, Reset and Clock Management

16.2.4.15.1 Clock and Power Management

The EDMA channel controller and transfer controller are clocked from L3MIAN1_L3_GICLK interface clock. The EDMA system runs at the L3 clock frequency.

The Auto clock gating for the EDMA_TPCC module is controlled by the EDMA_TPCC_CLKGDIS[0] CLKGDIS bit at the module level.

The L3MIAN1_L3_GICLK interface clock to EDMA controller's modules are controlled by the following registers in the PRCM module:

- PRCM.CM_L3MAIN1_TPCC_CLKCTRL - manages EDMA_TPCC module clock.
- PRCM.CM_L3MAIN1_TPTC1_CLKCTRL - manages EDMA_TPTC0 module clock.
- PRCM.CM_L3MAIN1_TPTC2_CLKCTRL - manages EDMA_TPTC1 module clock.

EDMA_TPCC and EDMA_TPTC0 and EDMA_TPTC1 modules have wakeup depandaces to several device modules. The wakeup dependency based on EDMA modules service requests are controlled by registers in PRCM module:

- PRCM.PM_L3MAIN1_TPCC_WKDEP - controls wakeup dependency based on TPCC service requests.
- PRCM.PM_L3MAIN1_TPTC1_WKDEP - controls wakeup dependency based on TPTC0 service requests.
- PRCM.PM_L3MAIN1_TPTC2_WKDEP - controls wakeup dependency based on TPTC1 service requests.

The EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1 can be placed in reduced-power modes to conserve power during periods of low activity. The power management of the peripheral is controlled by the PRCM module. The PRCM acts as a master controller for power management for all peripherals on the device.

The EDMA controller can be idled on receiving a clock stop request from the PRCM. The requests to EDMA_TPCC and EDMA_TPTC0 and EDMA_TPTC1 are separate. In general, it should be verified that there are no pending activities in the EDMA controller

Note

When EDMA controler modules no longer require the interface clock, software can disable it at the PRCM level by configuring the MODULEMODE bit field (PRCM.CM_L3MAIN1_TPCC_CLKCTRL[1:0], PRCM.CM_L3MAIN1_TPTC1_CLKCTRL[1:0], PRCM.CM_L3MAIN1_TPTC2_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At the PRCM level, when all the conditions to shut off the L3MIAN1_L3_GICLK clock are met the PRCM module automatically launches a hardware handshake protocol to ensure EDMA modules are ready to have this clock switched off. Namely, the PRCM module asserts an IDLE request to the EDMA modules. For more information, refer to *Power, Reset, and Clock Management*.

16.2.4.15.2 Reset Considerations

A hardware resets the EDMA (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1) and the EDMA configuration registers. The PaRAM memory contents are undefined after device reset and it should not rely on parameters to be reset to a known state. The PaRAM entry must be initialized to a desired value before it is used. The EDMA modules are reset by CORE_RET_RST reset signal from PRCM.

16.2.4.16 Emulation Considerations

During debug when using the emulator, the CPU(s) may be halted on an execute packet boundary for single-stepping, benchmarking, profiling, or other debug purposes. During an emulation halt, the EDMA channel controller and transfer controller operations continue. Events continue to be latched and processed and transfer requests continue to be submitted and serviced.

Since EDMA is involved in servicing multiple master and slave peripherals, it is not feasible to have an independent behavior of the EDMA for emulation halts. EDMA functionality would be coupled with the peripherals it is servicing, which might have different behavior during emulation halts.

16.2.5 EDMA Transfer Examples

The EDMA channel controller performs a variety of transfers depending on the parameter configuration. The following sections provide a description and PaRAM configuration for some typical use case scenarios.

16.2.5.1 Block Move Example

The most basic transfer performed by the EDMA is a block move. During device operation it is often necessary to transfer a block of data from one location to another, usually between on-chip and off-chip memory.

In this example, a section of data is to be copied from external memory to internal L2 SRAM as shown in Figure 16-33.

The source address for the transfer is set to the start of the data block in external memory, and the destination address is set to the start of the data block in L2. If the data block is less than 64K bytes, the PaRAM configuration shown in Figure 16-34 holds true with the synchronization type set to A-synchronized and indexes cleared to 0. If the amount of data is greater than 64K bytes, EDMA_TPCC_ABCNT_n[31:16] BCNT and the B-indexes need to be set appropriately with the synchronization type set to AB-synchronized. The EDMA_TPCC_OPT_n[3] STATIC bit is set to prevent linking.

This transfer example may also be set up using QDMA. For successive transfer submissions, of a similar nature, the number of cycles used to submit the transfer are fewer depending on the number of changing transfer parameters. The QDMA trigger word must be programed to be the highest numbered offset in the PaRAM set that undergoes change.

Figure 16-34 shows the parameters Block Move transfer.

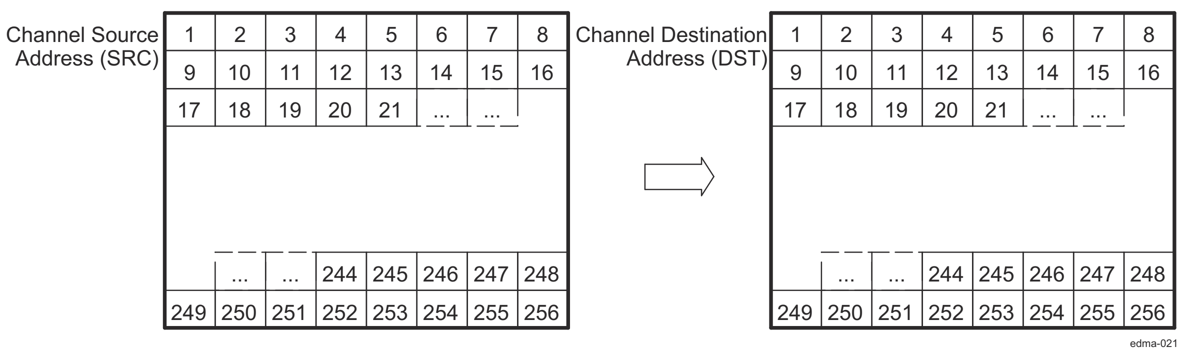


Figure 16-33. Block Move Example

Figure 16-34. Block Move Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0008h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0001h	0100h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0000h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b)Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[3] STATIC = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

16.2.5.2 Subframe Extraction Example

The EDMA can efficiently extract a small frame of data from a larger frame of data. By performing a 2D-to-1D transfer, the EDMA retrieves a portion of data for the CPU to process. In this example, a 640 × 480-pixel frame of video data is stored in external memory. Each pixel is represented by a 16-bit halfword. The CPU extracts a 16 × 12-pixel subframe of the image for processing. To facilitate more efficient processing time by the CPU, the EDMA places the subframe in internal L2 SRAM. Figure 16-35 shows the transfer of a subframe from external memory to L2.

The same PaRAM entry options are used for QDMA channels, as well as DMA channels. The EDMA_TPCC_OPT_n[3] STATIC bit is set to prevent linking. For successive transfers, only changed parameters need to be programmed before triggering the channel.

Figure 16-36 shows the parameters for Subframe Extraction transfer.

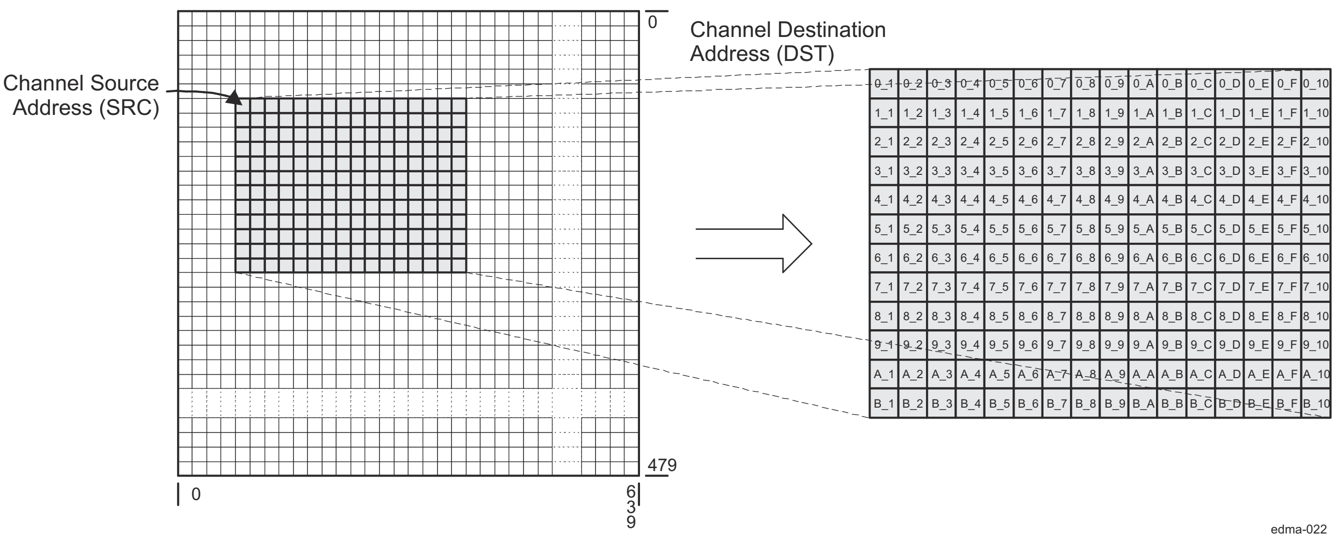


Figure 16-35. Subframe Extraction Transfer

Figure 16-36. Subframe Extraction Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 000Ch		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
000Ch	0020h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0020h	0500h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b)Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[3] STATIC = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

16.2.5.3 Data Sorting Example

Many applications require the use of multiple data arrays, it is often desirable to have the arrays arranged such that the first elements of each array are adjacent, the second elements are adjacent, and so on. Often this is not how the data is presented to the device. Either data is transferred via a peripheral with the data arrays arriving one after the other or the arrays are located in memory with each array occupying a portion of contiguous memory spaces. For these instances, the EDMA can reorganize the data into the desired format.

To determine the parameter set values, the following need to be considered:

- ACNT - Program this to be the size in bytes of an element.
- BCNT - Program this to be the number of elements in a frame.
- CCNT - Program this to be the number of frames.
- SBIDX - Program this to be the size of the element or ACNT.
- DBIDX - CCNT × ACNT
- SCIDX - ACNT × BCNT
- DCIDX - ACNT

The synchronization type needs to be AB-synchronized and the EDMA_TPCC_OPT_n[3] STATIC bit is 0 to allow updates to the parameter set. It is advised to use normal EDMA channels for sorting.

It is not possible to sort this with a single trigger event. Instead, the channel can be programmed to be chained to itself. After BCNT elements get sorted, intermediate chaining could be used to trigger the channel again causing the transfer of the next BCNT elements and so on. Figure 16-38 shows the parameter set programming for this transfer, assuming channel 0 and an element size of 4 bytes.

Figure 16-37 shows the Data Sorting transfer

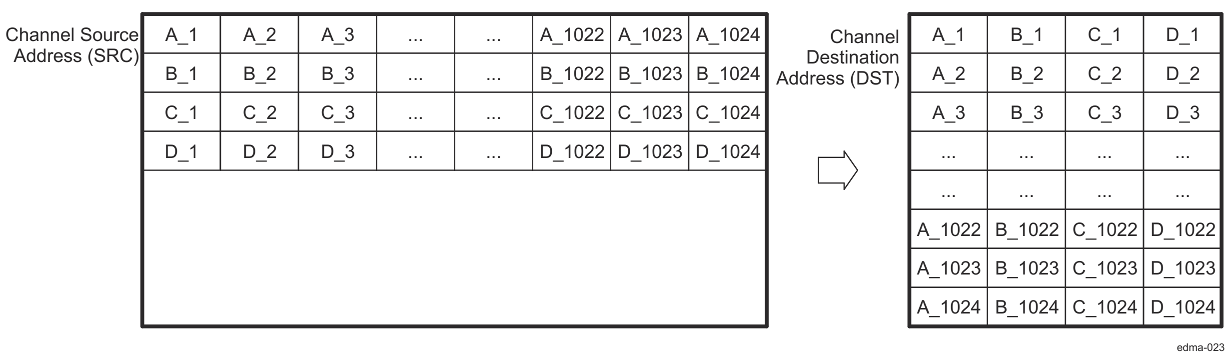


Figure 16-37. Data Sorting Example

Figure 16-38. Data Sorting Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0090 0004h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0400h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0010h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0004h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1
- EDMA_TPCC_OPT_n[23] ITCCHEN = 0x1

16.2.5.4 Peripheral Servicing Example

The EDMA channel controller also services peripherals in the background of CPU operation, without requiring any CPU intervention. Through proper initialization of the EDMA channels, they can be configured to continuously service on-chip and off-chip peripherals throughout the device operation. Each event available to the EDMA has its own dedicated channel, and all channels operate simultaneously. The only requirements are to use the proper channel for a particular transfer and to enable the channel event in the event enable register EDMA_TPCC_EER. When programming an EDMA channel to service a peripheral, it is necessary to know how data is to be presented to the processor. Data is always provided with some kind of synchronization event as either one element per event (non-bursting) or multiple elements per event (bursting).

16.2.5.4.1 Non-bursting Peripherals

Non-bursting peripherals include the on-chip multichannel audio serial port (McASP) and many external devices, such as codecs. Regardless of the peripheral, the EDMA channel configuration is the same.

The McASP transmit and receive data streams are treated independently by the EDMA. The transmit and receive data streams can have completely different counts, data sizes, and formats.

To transfer the incoming data stream to its proper location in DDR memory, the EDMA channel must be set up for a 1D-to-1D transfer with A-synchronization. Because an event (AREVT) is generated for every word as it arrives, it is necessary to have the EDMA issue the transfer request for each element individually. Figure 16-40 shows the parameters for this transfer. The source address of the EDMA channel is set to the data port address (DAT) for McASP, and the destination address is set to the start of the data block in DDR. Because the address of serializer buffer is fixed, the source B index is cleared to 0 (no modification) and the destination B index is set to 0x2 (increment).

Based on the premise that serial data is typically a high priority, the EDMA channel should be programmed to be on queue 0.

Figure 16-39 shows servicing incoming McASP data.

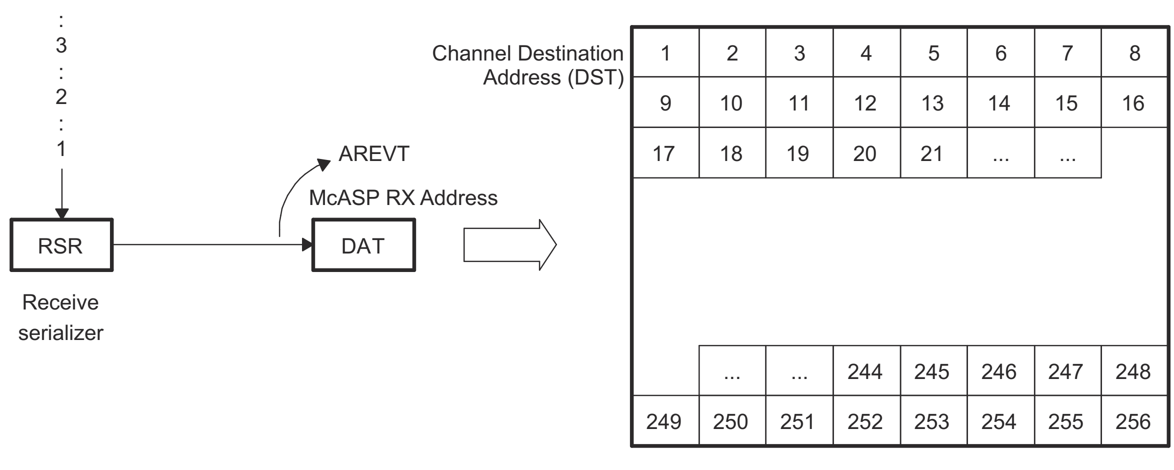


Figure 16-39. Servicing Incoming McASP Data Example

Figure 16-40. Servicing Incoming McASP Data Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0000h		Channel Options Parameter (OPT)	
McASP RX Address		Channel Source Address (SRC)	
0100h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0004h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

16.2.5.4.2 Bursting Peripherals

Higher bandwidth applications require that multiple data elements be presented to the processor core for every synchronization event. This frame of data can either be from multiple sources that are working simultaneously or from a single high-throughput peripheral that streams data to/from the processor.

In this example, a port is receiving a video frame from a camera and presenting it to the DSP one array at a time. The video image is 640 × 480 pixels, with each pixel represented by a 16-bit element. The image is to be stored in external memory.

To transfer data from an external peripheral to an external buffer one array at a time based on EVT*n* , channel *n* must be configured. Due to the nature of the data (a video frame made up of arrays of pixels) the destination is essentially a 2D entity. Figure 16-42 shows the parameters to service the incoming data with a 1D-to-2D transfer using AB-synchronization. The source address is set to the location of the video framer peripheral, and the destination address is set to the start of the data buffer. Because the input address is static, the EDMA_TPCC_BDIX_n[15:0] SBIDX is 0 (no modification to the source address). The destination is made up of arrays of contiguous, linear elements; therefore, the EDMA_TPCC_BIDX_n[31:16] DBIDX is set to pixel size, 2 bytes. EDMA_TPCC_ABCNT_n[15:0] ANCT is equal to the pixel size, 2 bytes. EDMA_TPCC_ABCNT_n[31:16] BCNT is set to the number of pixels in an array, 640. EDMA_TPCC_CCNT_n[15:0] CCNT is equal to the total number of arrays in the block, 480. EDMA_TPCC_CIDX_n[15:0] SCIDX is 0 because the source address undergoes no increment. The EDMA_TPCC_CIDX_n[31:16] DCIDX is equal to the difference between the starting addresses of each array. Because one pixel is 16 bits (2 bytes), EDMA_TPCC_CIDX_n[31:16] DCIDX is equal to 640 × 2.

Figure 16-41 shows Bursting Peripherals Transfer.

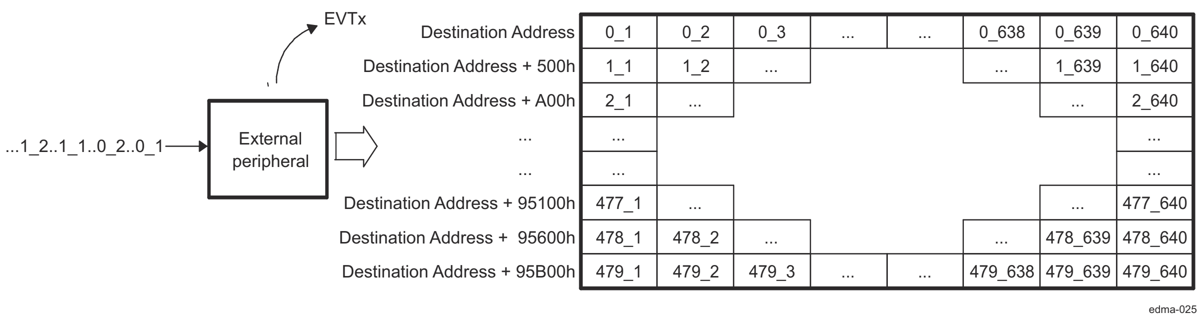


Figure 16-41. Servicing Peripheral Burst Example

Figure 16-42. Servicing Peripheral Burst Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0004h		Channel Options Parameter (OPT)	
Channel Source Address		Channel Source Address (SRC)	
0280h	0002h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address		Channel Destination Address (DST)	
0002h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0500h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	01E0h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

16.2.5.4.3 Continuous Operation

Configuring an EDMA channel to receive a single frame of data is useful, and is applicable to some systems. A majority of the time, however, data is going to be continuously transmitted and received throughout the entire operation of the processor. In this case, it is necessary to implement some form of linking such that the EDMA channels continuously reload the necessary parameter sets. In this example, McASP is configured to transmit and receive data on a T1 array. To simplify the example, only two channels are active for both transmit and receive data streams. Each channel receives packets of 128 elements. The packets are transferred from the serial port to internal memory and from internal memory to the serial port, as shown in Figure 16-43.

The McASP generates AREVT for every element received and generates AXEVT for every element transmitted. To service the data streams, them DMA channels associated with the McASP must be setup for 1D-to-1D transfers with A-synchronization.

Figure 16-44 shows the parameter entries for the channel for these transfers. To service the McASP continuously throughout DSP operation, the channels must be linked to a duplicate PaRAM set in the PaRAM. After all frames have been transferred, the EDMA channels reload and continue.

Figure 16-45 shows the reload parameters for the channel.

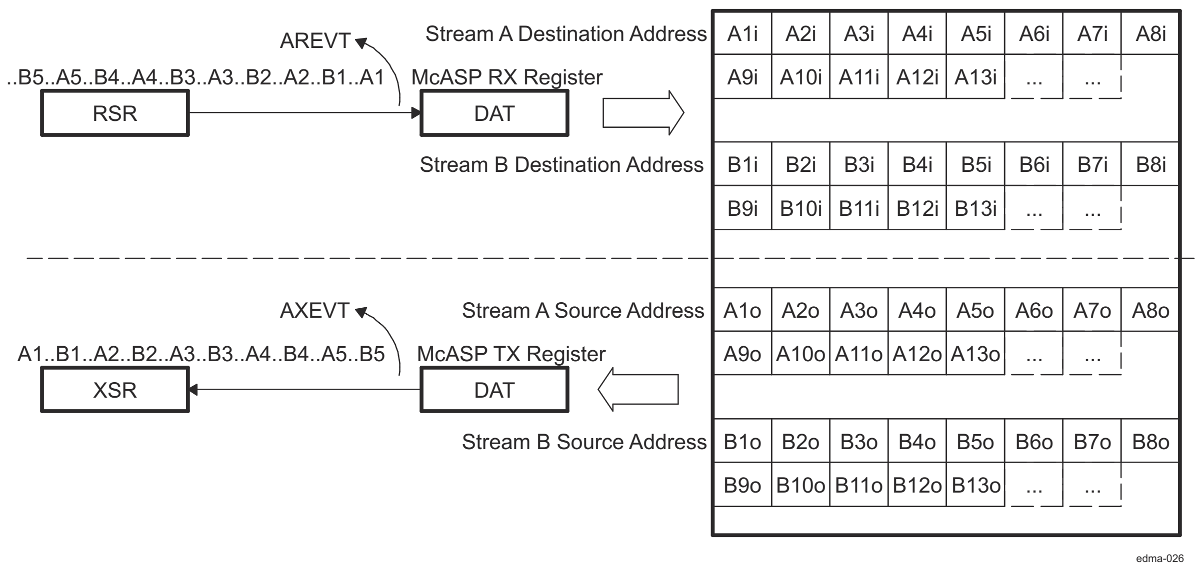


Figure 16-43. Servicing Continuous McASP Data Example

16.2.5.4.3.1 Receive Channel

EDMA channel 15 services the incoming data stream of McASP. The source address is set to that of the receive serializer buffer, and the destination address is set to the first element of the data block. Because there are two data channels being serviced, A and B, they are to be located separately within the L2 SRAM.

To facilitate continuous operation, a copy of the PaRAM set for the channel is placed in PaRAM set 64. The LINK option is set and the link address is provided in the PaRAM set. Upon exhausting the channel 15 parameter set, the parameters located at the link address are loaded into the channel 15 parameter set and operation continues. This function continues throughout device operation until halted by the CPU.

16.2.5.4.3.2 Transmit Channel

EDMA channel 12 services the outgoing data stream of McASP. In this case the destination address needs no update, hence, the parameter set changes accordingly. Linking is also used to allow continuous operation by the EDMA channel, with duplicate PaRAM set entries at PaRAM set 65.

16.2.5.4.3.3

Figure 16-44. Servicing Continuous McASP Data Example PaRAM Configuration

(a) EDMA Parameters for Receive Channel (PaRAM Set 15) being Linked to PaRAM Set 64

Parameter Contents		Parameter	
0010 0000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 15)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

(c) EDMA Parameters for Transmit Channel (PaRAM Set 12) being Linked to PaRAM Set 65

Parameter Contents		Parameter	
0010 1000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4860h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 12)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

Figure 16-45. Servicing Continuous McASP Data Example Reload PaRAM Configuration

(a) EDMA Reload Parameters (PaRAM Set 64) for Receive Channel

Parameter Contents		Parameter	
0010 0000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 64)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

(c) EDMA Reload Parameters (PaRAM Set 65) for Transmit Channel

Parameter Contents		Parameter	
0010 1000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4860h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 65)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

16.2.5.4.4 Ping-Pong Buffering

Although the previous configuration allows the EDMA to service a peripheral continuously, it presents a number of restrictions to the CPU. Because the input and output buffers are continuously being filled/emptied, the CPU must match the pace of the EDMA very closely to process the data. The EDMA receive data must always be placed in memory before the CPU accesses it, and the CPU must provide the output data before the EDMA transfers it. Though not impossible, this is an unnecessary challenge. It is particularly difficult in a 2-level cache scheme.

Ping-pong buffering is a simple technique that allows the CPU activity to be distanced from the EDMA activity. This means that there are multiple (usually two) sets of data buffers for all incoming and outgoing data streams. While the EDMA transfers the data into and out of the ping buffers, the CPU manipulates the data in the pong buffers. When both CPU and EDMA activity completes, they switch. The EDMA then writes over the old input data and transfers the new output data. Figure 16-46 shows the ping-pong scheme for this example.

To change the continuous operation example, such that a ping-pong buffering scheme is used, the EDMA channels need only a moderate change. Instead of one parameter set, there are two; one for transferring data to/from the ping buffers and one for transferring data to/from the pong buffers. As soon as one transfer completes, the channel loads the PaRAM set for the other and the data transfers continue. Figure 16-47 shows the EDMA channel configuration required.

Each channel has two parameter sets, ping and pong. The EDMA channel is initially loaded with the ping parameters (Figure 16-47). The link address for the ping set is set to the PaRAM offset of the pong parameter set (Figure 16-48). The link address for the pong set is set to the PaRAM offset of the ping parameter set (Figure 16-49). The channel options, count values, and index values are all identical between the ping and pong parameters for each channel. The only differences are the link address provided and the address of the data buffer.

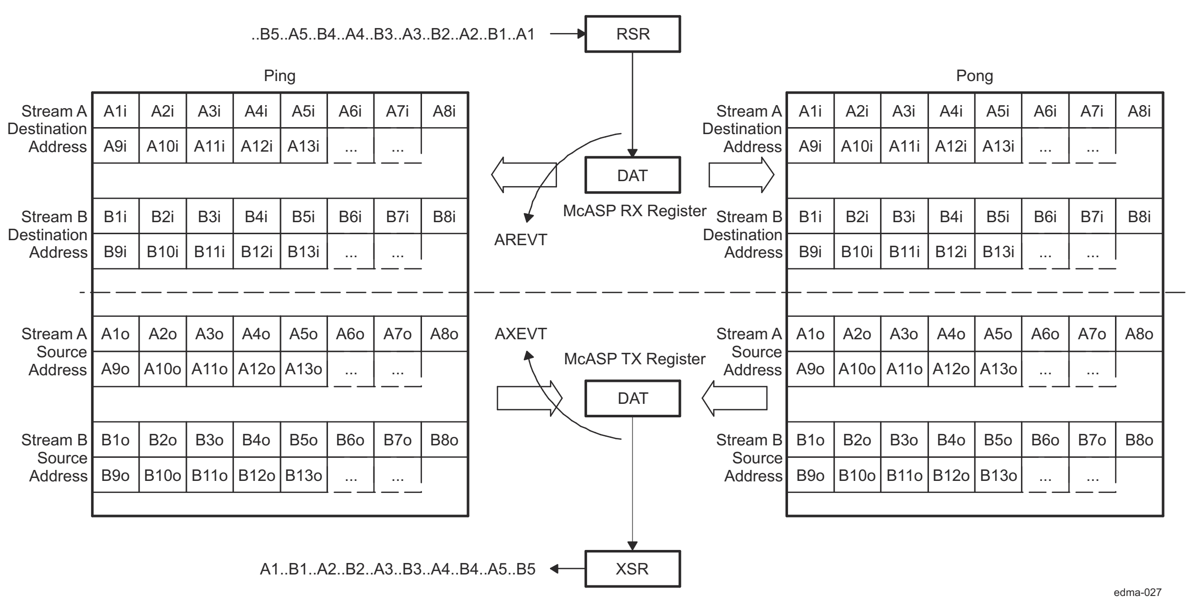


Figure 16-46. Ping-Pong Buffering for McASP Data Example

16.2.5.4.4.1 Synchronization with the CPU

To utilize the ping-pong buffering technique, the system must signal the CPU when to begin to access the new data set. After the CPU finishes processing an input buffer (ping), it waits for the EDMA to complete before switching to the alternate (pong) buffer. In this example, both channels provide their channel numbers as their report word and set the EDMA_TPCC_OPT_n[20] TCINTEN bit to generate an interrupt after completion. When channel 15 fills an input buffer, the E15 bit in the interrupt pending register EDMA_TPCC_IPR is set; when channel 12 empties an output buffer, the E12 bit in EDMA_TPCC_IPR is set. The CPU must manually clear these bits. With the channel parameters set, the CPU polls EDMA_TPCC_IPR to determine when to switch. The EDMA and CPU could alternatively be configured such that the channel completion interrupts the CPU. By doing this, the CPU could service a background task while waiting for the EDMA to complete.

Figure 16-47. Ping-Pong Buffering for McASP Example PaRAM Configuration

(a) EDMA Parameters for Channel 15 (Using PaRAM Set 15 Linked to Pong Set 64)

Parameter Contents		Parameter	
0010 D000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Channel 15

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

(c) EDMA Parameters for Channel 12 (Using PaRAM Set 12 Linked to Pong Set 65)

Parameter Contents		Parameter	
0010 C000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4840h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Channel 12

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

Figure 16-48. Ping-Pong Buffering for McASP Example Pong PaRAM Configuration

(a) EDMA Pong Parameters for Channel 15 at Set 64 Linked to Set 65

Parameter Contents		Parameter	
0010 D000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4820h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Pong Parameters for Channel 12 at Set 66 Linked to Set 67

Parameter Contents		Parameter	
0010 C000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4860h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

Figure 16-49. Ping-Pong Buffering for McASP Example Ping PaRAM Configuration

(a) EDMA Ping Parameters for Channel 15 at Set 65 Linked to Set 64

Parameter Contents		Parameter	
0010 D000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Ping Parameters for Channel 12 at Set 67 Linked to Set 66

Parameter Contents		Parameter	
0010 C000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0080h	4840h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

16.2.5.4.5 Transfer Chaining Examples

The following examples explain the intermediate transfer complete chaining function.

16.2.5.4.5.1 Servicing Input/Output FIFOs with a Single Event

Many systems require the use of a pair of external FIFOs that must be serviced at the same rate. One FIFO buffers data input, and the other buffers data output. The EDMA channels that service these FIFOs can be set up for AB-synchronized transfers. While each FIFO is serviced with a different set of parameters, both can be signaled from a single event.

For example, an external interrupt pin can be tied to the status flags of one of the FIFOs. When this event arrives, the EDMA needs to perform servicing for both the input and output streams. Without the intermediate transfer complete chaining feature this would require two events, and thus two external interrupt pins. The intermediate transfer complete chaining feature allows the use of a single external event (for example, a GPIO event). Figure 16-50 shows the EDMA setup and illustration for this example.

A GPIO event (in this case, GPINT0) triggers an array transfer. Upon completion of each intermediate array transfer of channel 48, intermediate transfer complete chaining sets the E8 bit (specified by TCC of 8) in the chained event register EDMA_TPCC_CER and provides a synchronization event to channel 8. Upon completion of the last array transfer of channel 48, transfer complete chaining—not intermediate transfer complete chaining—sets the E8 bit in EDMA_TPCC_CER (specified by EDMA_TPCC_OPT_n[11] TCCMODE: TCC) and provides a synchronization event to channel 8. The completion of channel 8 sets the I8 bit (specified by EDMA_TPCC_OPT_n[11] TCCMODE: TCC) in the interrupt pending register EDMA_TPCC_IPR, which can generate an interrupt to the CPU, if the I8 bit in the interrupt enable register EDMA_TPCC_IER is set.

Figure 16-50 shows the Intermediate Transfer Completion Chaining Example.

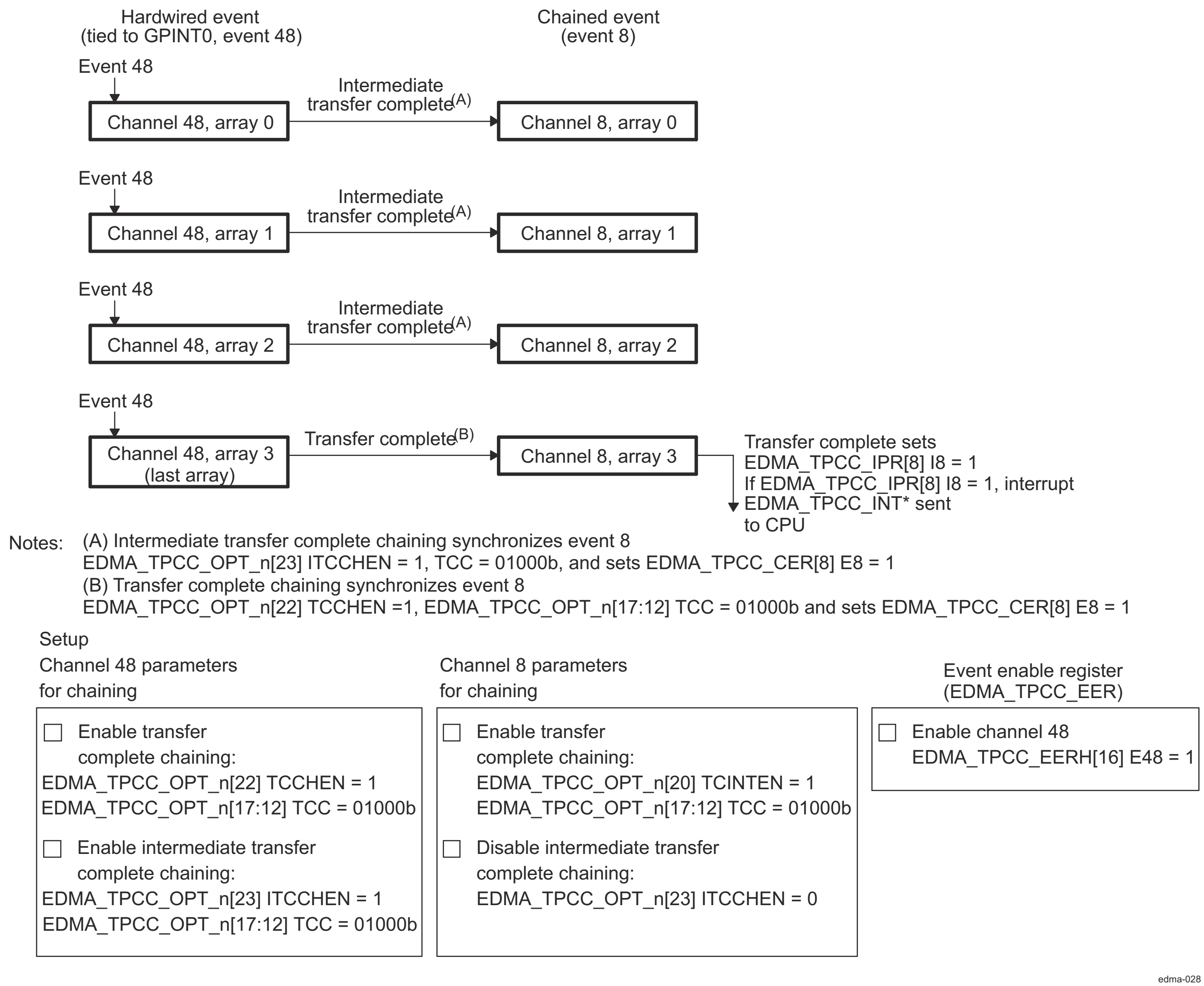


Figure 16-50. Intermediate Transfer Completion Chaining Example

16.2.5.4.5.2 Breaking Up Large Transfers with Intermediate Chaining

Another feature of intermediate transfer chaining EDMA_TPCC_OPT_n[23] ITCCHEN is for breaking up large transfers. A large transfer may lock out other transfers of the same priority level for the duration of the transfer. For example, a large transfer on queue 0 from the internal memory to the external memory using the EMIF may starve other EDMA transfers on the same queue. In addition, this large high-priority transfer may prevent the EMIF for a long duration to service other lower priority transfers. When a large transfer is considered to be high priority, it should be split into multiple smaller transfers. Figure 16-51 shows the EDMA setup and illustration of an example single large block transfer.

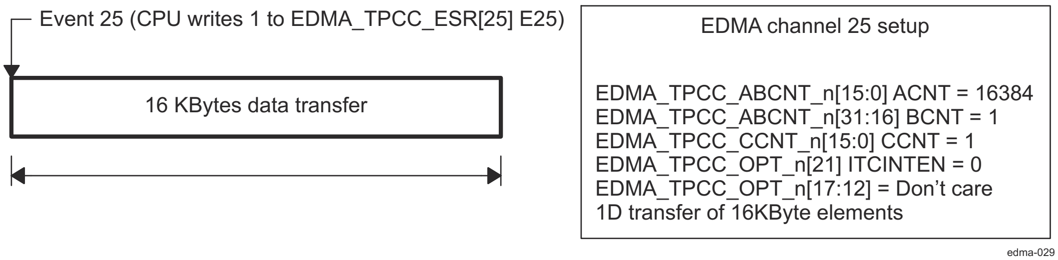


Figure 16-51. Single Large Block Transfer Example

The intermediate transfer chaining enable EDMA_TPCC_OPT_n[23] ITCCHEN provides a method to break up a large transfer into smaller transfers. For example, to move a single large block of memory (16K bytes), the EDMA performs an A-synchronized transfer. The element count is set to a reasonable value, where reasonable derives from the amount of time it would take to move this smaller amount of data. Assume 1 Kbyte is a reasonable small transfer in this example. The EDMA is set up to transfer 16 arrays of 1 Kbyte elements, for a total of 16K byte elements. The EDMA_TPCC_OPT_n[17:12] TCC field in the channel options parameter (OPT) is set to the same value as the channel number and EDMA_TPCC_OPT_n[23] ITCCHEN are set. In this example, EDMA channel 25 is used and EDMA_TPCC_OPT_n[17:12] TCC is also set to 25. The EDMA_TPCC_OPT_n[20] TCINTEN may also be set to trigger interrupt 25 when the last 1 Kbyte array is transferred. The CPU starts the EDMA transfer by writing to the appropriate bit of the event set register EDMA_TPCC_ESR[25] E25. The EDMA transfers the first 1 Kbyte array. Upon completion of the first array, intermediate transfer complete code chaining generates a synchronization event to channel 25, a value specified by the EDMA_TPCC_OPT_n[17:12] TCC field. This intermediate transfer completion chaining event causes EDMA channel 25 to transfer the next 1 Kbyte array. This process continues until the transfer parameters are exhausted, at which point the EDMA has completed the 16K byte transfer. This method breaks up a large transfer into smaller packets, thus providing natural time slices in the transfer such that other events may be processed. Figure 16-52 shows the EDMA setup and illustration of the broken up smaller packet transfers.

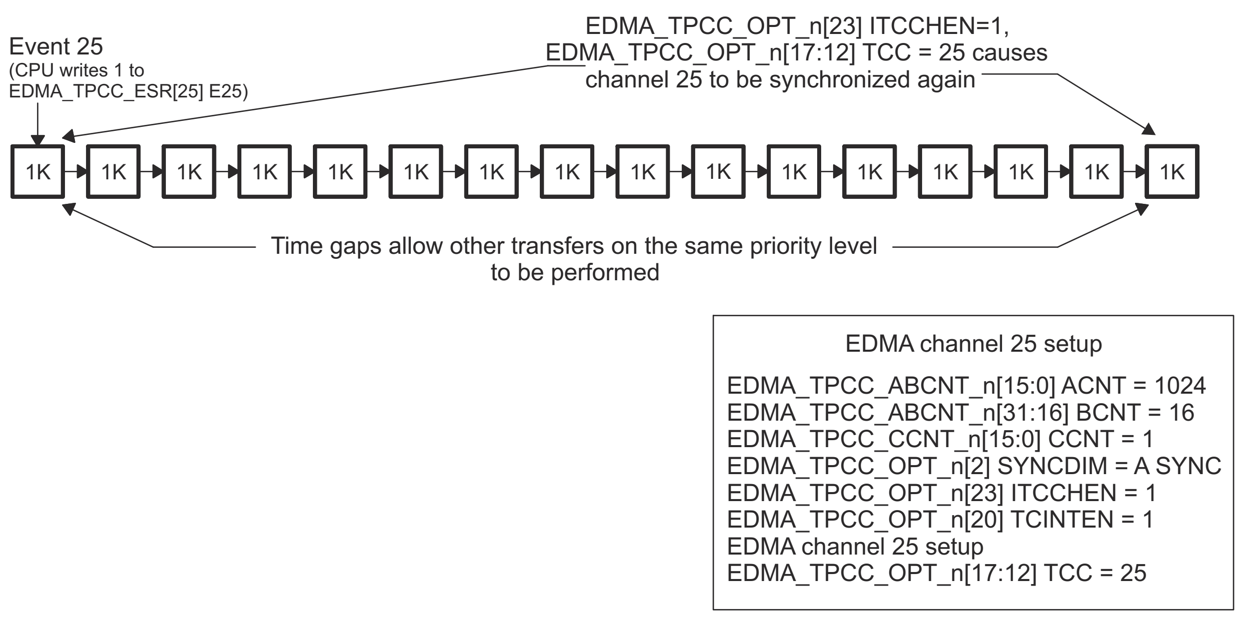


Figure 16-52. Smaller Packet Data Transfers Example

16.2.5.5 Setting Up an EDMA Transfer

16.2.5.5.1

The following list provides a quick guide for the typical steps involved in setting up a transfer.

1. Initiating a DMA/QDMA channel
   a. Determine the type of channel (QDMA or DMA) to be used.
   b. Channel mapping
      i. If using a QDMA channel, program the EDMA_TPCC_QCHMAPN_j with the parameter set number to which the channel maps and the trigger word.
      ii. If using a DMA channel, program the EDMA_TPCC_DCHMAPN_m with the parameter set number to which the channel maps.
   c. If the channel is being used in the context of a shadow region, ensure the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k for the region is properly set up to allow read write accesses to bits in the event registers and interrupt registers in the Shadow region memory map. The subsequent steps in this process should be done using the respective shadow region registers. (Shadow region descriptions and usage are provided in Section 16.2.4.7.1.)
   d. Determine the type of triggering used.
      i. If external events are used for triggering (DMA channels), enable the respective event in EDMA_TPCC_EER / EDMA_TPCC_EERH by writing into EDMA_TPCC_EESR / EDMA_TPCC_EESRH.
      ii. If QDMA Channel is used, enable the channel in EDMA_TPCC_QEER by writing into EDMA_TPCC_QEESR.
   e. Queue setup
      i. If a QDMA channel is used, set up the EDMA_TPCC_QDMAQNUM to map the channel to the respective event queue.
      ii. If a DMA channel is used, set up the EDMA_TPCC_DMAQNUMN_k to map the event to the respective event queue.
2. Parameter set setup
   a. Program the PaRAM set number associated with the channel. Note that

Note

If it is a QDMA channel, the PaRAM entry that is configured as trigger word is written to last. Alternatively, enable the QDMA channel (step 1-b-ii above) just before the write to the trigger word.

3. Interrupt setup
   a. Enable the interrupt in the EDMA_TPCC_IER / EDMA_TPCC_IERH by writing into EDMA_TPCC_IESR / EDMA_TPCC_IESRH.
   b. Ensure that the EDMA_TPCC completion interrupt (either the global or the shadow region interrupt) is enabled properly in the device interrupt controller.
   c. Ensure the EDMA_TPCC completion interrupt (this refers to either the Global interrupt or the shadow region interrupt) is enabled properly in the Device Interrupt controller.
   d. Set up the interrupt controller properly to receive the expected EDMA interrupt.
4. Initiate transfer
   a. This step is highly dependent on the event trigger source:
      i. If the source is an external event coming from a peripheral, the peripheral will be enabled to start generating relevant EDMA events that can be latched to the EDMA_TPCC_ER transfer.
      ii. For QDMA events, writes to the trigger word (step 2-a above) will initiate the transfer.
      iii. Manually triggered transfers will be initiated by writes to the Event Set Registers EDMA_TPCC_ESR / EDMA_TPCC_ESRH.
      iv. Chained-trigger events initiate when a previous transfer returns a transfer completion code equal to the chained channel number.
5. Wait for completion
   a. If the interrupts are enabled as mentioned in step 3 above, then the EDMA_TPCC will generate a completion interrupt to the CPU whenever transfer completion results in setting the corresponding bits

in the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH. The set bits must be cleared in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH by writing to corresponding bit in EDMA_TPCC_ICR / EDMA_TPCC_ICRH.

b. If polling for completion (interrupts not enabled in the device controller), then the application code can wait on the expected bits to be set in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH. Again, the set bits in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH must be manually cleared via EDMA_TPCC_ICR / EDMA_TPCC_ICRH before the next set of transfers is performed for the same transfer completion code values.

16.2.6 EDMA Debug Checklist and Programming Tips

This section lists some tips to keep in mind while debugging applications using the EDMA controller.

16.2.6.1 EDMA Debug Checklist

Table 16-76 provides some common issues and their probable causes and resolutions.

Table 16-76. Debug Checklist

Issue	Description/Solution
The transfer associated with the channel does not happen. The channel does not get serviced.	The EDMA_TPCC may not service a transfer request, even though the associated PaRAM set is programmed appropriately. Check for the following: 1) Verify that events are enabled, i.e., if an external/peripheral event is latched in Event Registers EDMA_TPCC_ER / EDMA_TPCC_ERH, check that the event is enabled in the Event Enable Registers EDMA_TPCC_EER / EDMA_TPCC_EERH. Similarly, for QDMA channels, check that QDMA events are appropriately enabled in the QDMA Event Enable Register EDMA_TPCC_QEER. 2) Verify that the DMA or QDMA Secondary Event Register EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER bits corresponding to the particular event or channel are not set.
The Secondary Event Registers bits are set, not allowing additional transfers to occur on a channel.	It is possible that a trigger event was received when the parameter set associated with the channel/event was a NULL set for a previous transfer on the channel. This is typical in two cases: 1) QDMA channels: Typically if the parameter set is non-static and expected to be terminated by a NULL set (i.e., EDMA_TPCC_OPT_n[3] STATIC = 0x0, EDMA_TPCC_LNK_n[15:0] LINK = 0xFFFF), the parameter set is updated with a NULL set after submission of the last TR. Because QDMA channels are auto-triggered, this update caused the generation of an event. An event generated for a NULL set causes an error condition and results in setting the bits corresponding to the QDMA channel in the EDMA_TPCC_QEMR and EDMA_TPCC_QSER. This will disable further prioritization of the channel. 2) DMA channels used in a continuous mode: The peripheral may be set up to continuously generate infinite events (for instance, in case of McASP, every time the data shifts out from the DXR register, it generates an XEVT). The parameter set may be programmed to expect only a finite number of events and to be terminated by a NULL link. After the expected number of events, the parameter set is reloaded with a NULL parameter set. Because the peripheral will generate additional events, an error condition is set in the EDMA_TPCC_SER[31:0] E*n* and EDMA_TPCC_EMR[31:0] E*n* set, preventing further event prioritization. Check the number of events received is limited to the expected number of events for which the parameter set is programmed, or check the bits corresponding to particular channel or event are not set in the Secondary event registers (EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER) and Event Missed Registers (EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR) before trying to perform subsequent transfers for the event/channel.

Table 16-76. Debug Checklist (continued)

Issue	Description/Solution
Completion interrupts are not asserted, or no further interrupts are received after the first completion interrupt.	Check the following: 1) The interrupt generation is enabled in the EDMA_TPCC_OPT_n of the associated PaRAM set (EDMA_TPCC_OPT_n[20] TCINTEN = 0x1 and/or EDMA_TPCC_OPT_n[20] ITCINTEN = 0x1). 2) The interrupts are enabled in the EDMA Channel Controller, via the Interrupt Enable Registers (EDMA_TPCC_IER / EDMA_TPCC_IERH). 3) The corresponding interrupts are enabled in the device interrupt controller. 4) The set interrupts are cleared in the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH) before exiting the transfer completion interrupt service routine (ISR). See Section 16.2.4.9.1.2 *Clearing Transfer Complerion Interrupts* for details on writing EDMA ISRs. 5) If working with shadow region interrupts, make sure that the DMA Region Access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k) are set up properly, because the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers act as secondary enables for shadow region completion interrupts, along with the EDMA_TPCC_IER / EDMA_TPCC_IERH registers. If working with shadow region interrupts, make sure that the bits corresponding to the transfer completion code EDMA_TPCC_OPT_n[17:12] TCC value are also enabled in the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers. For instance, if the PaRAM set associated with Channel 0 returns a completion code of 63 EDMA_TPCC_OPT_n[17:12] TCC = 63, ensure that EDMA_TPCC_DRAEHM_k[31] E63 is also set for a shadow region completion interrupt because the interrupt pending register bit set will be EDMA_TPCC_IPRH[31] I63 (not EDMA_TPCC_IPR[0] I0).

16.2.6.2 EDMA Programming Tips

1. For several registers, the setting and clearing of bits needs to be done via separate dedicated registers. For example, the Event Register (EDMA_TPCC_ER / EDMA_TPCC_ERH) can only be cleared by writing a 1 to the corresponding bits in the Event Clear Registers (EDMA_TPCC_ECR / EDMA_TPCC_ECRH). Similarly, the Event Enable Register (EDMA_TPCC_EER / EDMA_TPCC_EERH) bits can only be set with writing of 0x1 to the Event Enable Set Registers (EDMA_TPCC_EESR / EDMA_TPCC_EESRH) and cleared with writing of 0x1 to the corresponding bits in the Event Enable Clear Register (EDMA_TPCC_EECR / EDMA_TPCC_EECRH).
2. Writes to the shadow region memory maps are governed by region access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k / EDMA_TPCC_QRAEN_k). If the appropriate channels are not enabled in these registers, read/write access to the shadow region memory map is not enabled.
3. When working with shadow region completion interrupts, ensure that the DMA Region Access Registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k) for every region are set in a mutually exclusive way (unless it is a requirement for an application). If there is an overlap in the allocated channels and transfer completion codes (setting of Interrupt Pending Register bits) in the region resource allocation, it results in multiple shadow region completion interrupts.
   For example, if EDMA_TPCC_DRAEM_k.DRAEM_0[0] E0 and EDMA_TPCC_DRAEM_k.DRAEM_1[0] E0 are both set, then on completion of a transfer that returns a TCC = 0x0, they will generate both shadow region 0 and 1 completion interrupts.
4. While programming a non-dummy parameter set, ensure the EDMA_TPCC_CCNT_n[15:0] CCNT is not left to zero.
5. Enable the EDMA_TPCC error interrupt in the device controller and attach an interrupt service routine (ISR) to ensure that error conditions are not missed in an application and are appropriately addressed with the ISR.
6. Depending on the application, it can want to break large transfers into smaller transfers and use self-chaining to prevent starvation of other events in an event queue.
7. In applications where a large transfer is broken into sets of small transfers using chaining or other methods, it chooses to use the early chaining option to reduce the time between the sets of transfers and increase the throughput.
   However, keep in mind that with early completion, all data might have not been received at the end point when completion is reported because the EDMA_TPCC internally signals completion when the TR is submitted to the EDMA_TPTC, potentially before any data has been transferred.
8. The event queue entries can be observed to determine the last few events if there is a system failure (provided the entries were not bypassed).

16.2.7 EDMA Register Manual

16.2.7.1 EDMA Instance Summary

Table 16-77shows the L3_MAIN base address and address space for the EDMA module instances.

Table 16-77. EDMA Instance Summary

Module Name	Base Address (L3_MAIN Access)	Size
SYS_EDMA_TPCC	0x4330 0000	1 MB
SYS_EDMA_TPTC0	0x4340 0000	1 MB
SYS_EDMA_TPTC1	0x4350 0000	1 MB
DSP1_EDMA_TPCC	0x40D1 0000	32 KB
DSP1_EDMA_TPTC0	0x40D0 5000	4 KB
DSP1_EDMA_TPTC1	0x40D0 6000	4 KB
DSP2_EDMA_TPCC	0x4151 0000	32 KB
DSP2_EDMA_TPTC0	0x4150 5000	4 KB
DSP2_EDMA_TPTC1	0x4150 6000	4 KB
EVE1_EDMA_TPCC	0x420A 0000	32 KB
EVE2_EDMA_TPCC	0x421A 0000	32 KB
EVE1_EDMA_TPTC0	0x4208 6000	4 KB
EVE2_EDMA_TPTC0	0x4218 6000	4 KB
EVE1_EDMA_TPTC1	0x4208 7000	4 KB
EVE2_EDMA_TPTC1	0x4218 7000	4 KB

Table 16-78 lists the base addresses for DSP internal (private) access to its embedded TPCC / TPTC modules.

Table 16-78. DSP Private Access EDMA Instance Summary

Module Name	Base Address	Size
DSP_EDMA_TPCC	0x01D1 0000	32 KB
DSP_EDMA_TPTC0	0x01D0 5000	4 KB
DSP_EDMA_TPTC1	0x01D0 6000	4 KB

Table 16-79 lists the base addresses for EVE internal (private) access to its embedded TPCC / TPTC modules.

Table 16-79. EVE EDMA Instance Summary (Private Access)

Module Name	Base Address (EVE Private Access)	Size
EVE_EDMA_TPCC	0x400A 0000	32 KB
EVE_EDMA_TPTC0	0x4008 6000	4 KB
EVE_EDMA_TPTC1	0x4008 7000	4 KB

16.2.7.2 EDMA Registers

16.2.7.2.1 EDMA Register Summary

Table 16-80 through Table 16-90 summarize the EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1 registers.

Note

All EDMA modules in the SoC are functionally identical. Note that some of the configuration parameters may be different for the various EDMA instances (see Section 16.2.1.3, *EDMA Controllers Configuration*).

Table 16-80. System EDMA_TPCC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x4330 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x4330 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x4330 00FC
EDMA_TPCC_DCHMAPN_m (5)	RW	32	0x0000 0100 + (0x4 * m)	0x4330 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j (2)	RW	32	0x0000 0200 + (0x4 * j)	0x4330 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUMN_k (3)	RW	32	0x0000 0240 + (0x4 * k)	0x4330 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x4330 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x4330 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x4330 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x4330 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x4330 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x4330 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x4330 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x4330 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x4330 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x4330 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x4330 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x4330 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	0x4330 0340 + (0x8 * k)
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	0x4330 0344 + (0x8 * k)
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x4330 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 *pl)	0x4330 0400 + (0x4 * p)
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 * p)	0x4330 0440 + (0x4 * p)
EDMA_TPCC_QSTATN_i (1)	R	32	0x0000 0600 + (0x4 * i)	0x4330 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x4330 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x4330 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x4330 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x4330 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x4330 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x4330 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x4330 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x4330 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x4330 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x4330 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x4330 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x4330 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x4330 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x4330 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x4330 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x4330 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x4330 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x4330 1018
EDMA_TPCC_CERH	R	32	0x0000 101C	0x4330 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x4330 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x4330 1024

Table 16-80. System EDMA_TPCC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_EECR	W	32	0x0000 1028	0x4330 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x4330 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x4330 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x4330 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x4330 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x4330 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x4330 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x4330 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x4330 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x4330 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x4330 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x4330 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x4330 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x4330 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x4330 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x4330 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x4330 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x4330 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x4330 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x4330 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x4330 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x4330 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x4330 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x4330 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x4330 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x4330 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x4330 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x4330 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x4330 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x4330 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x4330 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x4330 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x4330 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x4330 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x4330 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x4330 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x4330 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x4330 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x4330 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x4330 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x4330 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k (3)	W	32	0x0000 2040 + (0x200 * k)	0x4330 2040 + (0x200 * k)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x4330 2044 + (0x200 * k)
EDMA_TPCC_IER_RN_k (3)	R	32	0x0000 2050 + (0x200 * k)	0x4330 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k (3)	R	32	0x0000 2054 + (0x200 * k)	0x4330 2054 + (0x200 * k)

Table 16-80. System EDMA_TPCC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_IECR_RN_k [(3)]	W	32	0x0000 2058 + (0x200 * k)	0x4330 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k [(3)]	W	32	0x0000 205C + (0x200 * k)	0x4330 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k [(3)]	W	32	0x0000 2060 + (0x200 * k)	0x4330 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k [(3)]	W	32	0x0000 2064 + (0x200 * k)	0x4330 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k [(3)]	R	32	0x0000 2068 + (0x200 * k)	0x4330 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k [(3)]	R	32	0x0000 206C + (0x200 * k)	0x4330 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k [(3)]	W	32	0x0000 2070 + (0x200 * k)	0x4330 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k [(3)]	W	32	0x0000 2074 + (0x200 * k)	0x4330 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k [(3)]	W	32	0x0000 2078 + (0x200 * k)	0x4330 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k [(3)]	R	32	0x0000 2080 + (0x200 * k)	0x4330 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k [(3)]	R	32	0x0000 2084 + (0x200 * k)	0x4330 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k [(3)]	W	32	0x0000 2088 + (0x200 * k)	0x4330 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k [(3)]	W	32	0x0000 208C + (0x200 * k)	0x4330 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k [(3)]	R	32	0x0000 2090 + (0x200 * k)	0x4330 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k [(3)]	W	32	0x0000 2094 + (0x200 * k)	0x4330 2094 + (0x200 * k)
EDMA_TPCC_OPT_n [(6)]	RW	32	0x0000 4000 + (0x20 * n)	0x4330 4000 + (0x20 * n)
EDMA_TPCC_SRC_n [(6)]	RW	32	0x0000 4004 + (0x20 * n)	0x4330 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n [(6)]	RW	32	0x0000 4008 + (0x20 * n)	0x4330 4008 + (0x20 * n)
EDMA_TPCC_DST_n [(6)]	RW	32	0x0000 400C + (0x20 * n)	0x4330 400C + (0x20 * n)
EDMA_TPCC_BIDX_n [(6)]	RW	32	0x0000 4010 + (0x20 * n)	0x4330 4010 + (0x20 * n)
EDMA_TPCC_LNK_n [(6)]	RW	32	0x0000 4014 + (0x20 * n)	0x4330 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n [(6)]	RW	32	0x0000 4018 + (0x20 * n)	0x4330 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n [(6)]	RW	32	0x0000 401C + (0x20 * n)	0x4330 401C + (0x20 * n)

(1) i = 0 to 1 for SYS_EDMA_TPCC
(2) j = 0 to 7 for SYS_EDMA_TPCC
(3) k = 0 to 7 for SYS_EDMA_TPCC
(4) p = 0 to 15 for SYS_EDMA_TPCC
(5) m = 0 to 63 for SYS_EDMA_TPCC
(6) n = 0 to 512 for SYS_EDMA_TPCC

Table 16-81. DSP1 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x40D1 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x40D1 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x40D1 00FC
EDMA_TPCC_DCHMAPN_m [(5)]	RW	32	0x0000 0100 + (0x4 * m)	0x40D1 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j [(2)]	RW	32	0x0000 0200 + (0x4 * j)	0x40D1 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUMN_k [(3)]	RW	32	0x0000 0240 + (0x4 * k)	0x40D1 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x40D1 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x40D1 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x40D1 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x40D1 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x40D1 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x40D1 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x40D1 030C

Table 16-81. DSP1 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x40D1 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x40D1 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x40D1 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x40D1 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x40D1 0320
EDMA_TPCC_DRAEM_k [3]	RW	32	0x0000 0340 + (0x8 * k)	0x40D1 0340 + (0x8 * k)
EDMA_TPCC_DRAEHM_k [3]	RW	32	0x0000 0344 + (0x8 * k)	0x40D1 0344 + (0x8 * k)
EDMA_TPCC_QRAEN_k [3]	RW	32	0x0000 0380 + (0x4 * k)	0x40D1 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p [4]	R	32	0x0000 0400 + (0x4 *pl)	0x40D1 0400 + (0x4 * p)
EDMA_TPCC_Q1E_p [4]	R	32	0x0000 0440 + (0x4 * p)	0x40D1 0440 + (0x4 * p)
EDMA_TPCC_QSTATN_i [1]	R	32	0x0000 0600 + (0x4 * i)	0x40D1 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x40D1 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x40D1 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x40D1 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x40D1 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x40D1 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x40D1 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x40D1 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x40D1 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x40D1 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x40D1 080C
EDMA_TPCC_MPPAN_k [3]	RW	32	0x0000 0810 + (0x4 * k)	0x40D1 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x40D1 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x40D1 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x40D1 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x40D1 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x40D1 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x40D1 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x40D1 1018
EDMA_TPCC_CERH	R	32	0x0000 101C	0x40D1 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x40D1 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x40D1 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x40D1 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x40D1 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x40D1 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x40D1 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x40D1 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x40D1 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x40D1 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x40D1 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x40D1 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x40D1 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x40D1 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x40D1 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x40D1 1060

Table 16-81. DSP1 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x40D1 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x40D1 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x40D1 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x40D1 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x40D1 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x40D1 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x40D1 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x40D1 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x40D1 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x40D1 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x40D1 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x40D1 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x40D1 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x40D1 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x40D1 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x40D1 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x40D1 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x40D1 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x40D1 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x40D1 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x40D1 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x40D1 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x40D1 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x40D1 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x40D1 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x40D1 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x40D1 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x40D1 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k (3)	W	32	0x0000 2040 + (0x200 * k)	0x40D1 2040 + (0x200 * k)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x40D1 2044 + (0x200 * k)
EDMA_TPCC_IER_RN_k (3)	R	32	0x0000 2050 + (0x200 * k)	0x40D1 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k (3)	R	32	0x0000 2054 + (0x200 * k)	0x40D1 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k (3)	W	32	0x0000 2058 + (0x200 * k)	0x40D1 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k (3)	W	32	0x0000 205C + (0x200 * k)	0x40D1 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k (3)	W	32	0x0000 2060 + (0x200 * k)	0x40D1 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k (3)	W	32	0x0000 2064 + (0x200 * k)	0x40D1 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k (3)	R	32	0x0000 2068 + (0x200 * k)	0x40D1 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k (3)	R	32	0x0000 206C + (0x200 * k)	0x40D1 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k (3)	W	32	0x0000 2070 + (0x200 * k)	0x40D1 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k (3)	W	32	0x0000 2074 + (0x200 * k)	0x40D1 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k (3)	W	32	0x0000 2078 + (0x200 * k)	0x40D1 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k (3)	R	32	0x0000 2080 + (0x200 * k)	0x40D1 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k (3)	R	32	0x0000 2084 + (0x200 * k)	0x40D1 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k (3)	W	32	0x0000 2088 + (0x200 * k)	0x40D1 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k (3)	W	32	0x0000 208C + (0x200 * k)	0x40D1 208C + (0x200 * k)

Table 16-81. DSP1 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_QSER_RN_k (3)	R	32	0x0000 2090 + (0x200 * k)	0x40D1 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k (3)	W	32	0x0000 2094 + (0x200 * k)	0x40D1 2094 + (0x200 * k)
EDMA_TPCC_OPT_n (6)	RW	32	0x0000 4000 + (0x20 * n)	0x40D1 4000 + (0x20 * n)
EDMA_TPCC_SRC_n (6)	RW	32	0x0000 4004 + (0x20 * n)	0x40D1 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n (6)	RW	32	0x0000 4008 + (0x20 * n)	0x40D1 4008 + (0x20 * n)
EDMA_TPCC_DST_n (6)	RW	32	0x0000 400C + (0x20 * n)	0x40D1 400C + (0x20 * n)
EDMA_TPCC_BIDX_n (6)	RW	32	0x0000 4010 + (0x20 * n)	0x40D1 4010 + (0x20 * n)
EDMA_TPCC_LNK_n (6)	RW	32	0x0000 4014 + (0x20 * n)	0x40D1 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n (6)	RW	32	0x0000 4018 + (0x20 * n)	0x40D1 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n (6)	RW	32	0x0000 401C + (0x20 * n)	0x40D1 401C + (0x20 * n)

(1) i = 0 to 1 for DSP1_EDMA_TPCC
(2) j = 0 to 7 for DSP1_EDMA_TPCC
(3) k = 0 to 7 for DSP1_EDMA_TPCC
(4) p = 0 to 15 for DSP1_EDMA_TPCC
(5) m = 0 to 63 for DSP1_EDMA_TPCC
(6) n = 0 to 127 for DSP1_EDMA_TPCC

Table 16-82. DSP2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x4151 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x4151 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x4151 00FC
EDMA_TPCC_DCHMAPN_m (5)	RW	32	0x0000 0100 + (0x4 * m)	0x4151 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j (2)	RW	32	0x0000 0200 + (0x4 * j)	0x4151 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUM_k (3)	RW	32	0x0000 0240 + (0x4 * k)	0x4151 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x4151 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x4151 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x4151 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x4151 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x4151 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x4151 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x4151 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x4151 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x4151 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x4151 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x4151 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x4151 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	0x4151 0340 + (0x8 * k)
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	0x4151 0344 + (0x8 * k)
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x4151 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 * p)	0x4151 0400 + (0x4 * p)
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 * p)	0x4151 0440 + (0x4 * p)
EDMA_TPCC_QSTATN_i (1)	R	32	0x0000 0600 + (0x4 * i)	0x4151 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x4151 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x4151 0624

Table 16-82. DSP2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x4151 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x4151 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x4151 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x4151 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x4151 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x4151 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x4151 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x4151 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x4151 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x4151 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x4151 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x4151 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x4151 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x4151 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x4151 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x4151 1018
EDMA_TPCC_CERH	R	32	0x0000 101C	0x4151 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x4151 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x4151 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x4151 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x4151 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x4151 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x4151 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x4151 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x4151 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x4151 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x4151 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x4151 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x4151 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x4151 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x4151 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x4151 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x4151 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x4151 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x4151 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x4151 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x4151 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x4151 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x4151 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x4151 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x4151 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x4151 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x4151 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x4151 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x4151 2000 + (0x200 * k)

Table 16-82. DSP2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x4151 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x4151 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x4151 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x4151 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x4151 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x4151 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x4151 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x4151 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x4151 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x4151 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x4151 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x4151 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x4151 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x4151 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x4151 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k (3)	W	32	0x0000 2040 + (0x200 * k)	0x4151 2040 + (0x200 * k)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x4151 2044 + (0x200 * k)
EDMA_TPCC_IER_RN_k (3)	R	32	0x0000 2050 + (0x200 * k)	0x4151 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k (3)	R	32	0x0000 2054 + (0x200 * k)	0x4151 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k (3)	W	32	0x0000 2058 + (0x200 * k)	0x4151 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k (3)	W	32	0x0000 205C + (0x200 * k)	0x4151 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k (3)	W	32	0x0000 2060 + (0x200 * k)	0x4151 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k (3)	W	32	0x0000 2064 + (0x200 * k)	0x4151 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k (3)	R	32	0x0000 2068 + (0x200 * k)	0x4151 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k (3)	R	32	0x0000 206C + (0x200 * k)	0x4151 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k (3)	W	32	0x0000 2070 + (0x200 * k)	0x4151 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k (3)	W	32	0x0000 2074 + (0x200 * k)	0x4151 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k (3)	W	32	0x0000 2078 + (0x200 * k)	0x4151 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k (3)	R	32	0x0000 2080 + (0x200 * k)	0x4151 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k (3)	R	32	0x0000 2084 + (0x200 * k)	0x4151 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k (3)	W	32	0x0000 2088 + (0x200 * k)	0x4151 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k (3)	W	32	0x0000 208C + (0x200 * k)	0x4151 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k (3)	R	32	0x0000 2090 + (0x200 * k)	0x4151 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k (3)	W	32	0x0000 2094 + (0x200 * k)	0x4151 2094 + (0x200 * k)
EDMA_TPCC_OPT_n (6)	RW	32	0x0000 4000 + (0x20 * n)	0x4151 4000 + (0x20 * n)
EDMA_TPCC_SRC_n (6)	RW	32	0x0000 4004 + (0x20 * n)	0x4151 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n (6)	RW	32	0x0000 4008 + (0x20 * n)	0x4151 4008 + (0x20 * n)
EDMA_TPCC_DST_n (6)	RW	32	0x0000 400C + (0x20 * n)	0x4151 400C + (0x20 * n)
EDMA_TPCC_BIDX_n (6)	RW	32	0x0000 4010 + (0x20 * n)	0x4151 4010 + (0x20 * n)
EDMA_TPCC_LNK_n (6)	RW	32	0x0000 4014 + (0x20 * n)	0x4151 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n (6)	RW	32	0x0000 4018 + (0x20 * n)	0x4151 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n (6)	RW	32	0x0000 401C + (0x20 * n)	0x4151 401C + (0x20 * n)

(1) i = 0 to 1 for DSP2_EDMA_TPCC
(2) j = 0 to 7 for DSP2_EDMA_TPCC
(3) k = 0 to 7 for DSP2_EDMA_TPCC

(4) p = 0 to 15 for DSP2_EDMA_TPCC
(5) m = 0 to 63 for DSP2_EDMA_TPCC
(6) n = 0 to 127 for DSP2_EDMA_TPCC

Table 16-83. DSP EDMA_TPCC Registers Mapping Summary (Private Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPCC Physical Address (DSP Private Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x01D1 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x01D1 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x01D1 00FC
EDMA_TPCC_DCHMAPN_m (5)	RW	32	0x0000 0100 + (0x4 * m)	0x01D1 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j (2)	RW	32	0x0000 0200 + (0x4 * j)	0x01D1 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUMN_k (3)	RW	32	0x0000 0240 + (0x4 * k)	0x01D1 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x01D1 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x01D1 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x01D1 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x01D1 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x01D1 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x01D1 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x01D1 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x01D1 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x01D1 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x01D1 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x01D1 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x01D1 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	0x01D1 0340 + (0x8 * k)
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	0x01D1 0344 + (0x8 * k)
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x01D1 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 *pl)	0x01D1 0400 + (0x4 * p)
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 * p)	0x01D1 0440 + (0x4 * p)
EDMA_TPCC_QSTATN_i (1)	R	32	0x0000 0600 + (0x4 * i)	0x01D1 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x01D1 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x01D1 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x01D1 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x01D1 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x01D1 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x01D1 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x01D1 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x01D1 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x01D1 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x01D1 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x01D1 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x01D1 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x01D1 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x01D1 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x01D1 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x01D1 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x01D1 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x01D1 1018

Table 16-83. DSP EDMA_TPCC Registers Mapping Summary (Private Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPCC Physical Address (DSP Private Access)
EDMA_TPCC_CERH	R	32	0x0000 101C	0x01D1 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x01D1 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x01D1 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x01D1 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x01D1 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x01D1 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x01D1 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x01D1 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x01D1 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x01D1 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x01D1 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x01D1 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x01D1 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x01D1 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x01D1 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x01D1 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x01D1 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x01D1 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x01D1 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x01D1 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x01D1 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x01D1 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x01D1 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x01D1 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x01D1 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x01D1 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x01D1 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x01D1 1094
EDMA_TPCC_ER_RN_k [3]	R	32	0x0000 2000 + (0x200 * k)	0x01D1 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k [3]	R	32	0x0000 2004 + (0x200 * k)	0x01D1 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k [3]	W	32	0x0000 2008 + (0x200 * k)	0x01D1 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k [3]	W	32	0x0000 200C + (0x200 * k)	0x01D1 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k [3]	W	32	0x0000 2010 + (0x200 * k)	0x01D1 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k [3]	W	32	0x0000 2014 + (0x200 * k)	0x01D1 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k [3]	R	32	0x0000 2018 + (0x200 * k)	0x01D1 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k [3]	R	32	0x0000 201C + (0x200 * k)	0x01D1 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k [3]	R	32	0x0000 2020 + (0x200 * k)	0x01D1 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k [3]	R	32	0x0000 2024 + (0x200 * k)	0x01D1 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k [3]	W	32	0x0000 2028 + (0x200 * k)	0x01D1 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k [3]	W	32	0x0000 202C + (0x200 * k)	0x01D1 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k [3]	W	32	0x0000 2030 + (0x200 * k)	0x01D1 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k [3]	W	32	0x0000 2034 + (0x200 * k)	0x01D1 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k [3]	R	32	0x0000 2038 + (0x200 * k)	0x01D1 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k [3]	R	32	0x0000 203C + (0x200 * k)	0x01D1 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k [3]	W	32	0x0000 2040 + (0x200 * k)	0x01D1 2040 + (0x200 * k)

Table 16-83. DSP EDMA_TPCC Registers Mapping Summary (Private Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPCC Physical Address (DSP Private Access)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x01D1 2044 + (0x200 * k)
EDMA_TPCC_IER_RN_k (3)	R	32	0x0000 2050 + (0x200 * k)	0x01D1 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k (3)	R	32	0x0000 2054 + (0x200 * k)	0x01D1 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k (3)	W	32	0x0000 2058 + (0x200 * k)	0x01D1 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k (3)	W	32	0x0000 205C + (0x200 * k)	0x01D1 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k (3)	W	32	0x0000 2060 + (0x200 * k)	0x01D1 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k (3)	W	32	0x0000 2064 + (0x200 * k)	0x01D1 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k (3)	R	32	0x0000 2068 + (0x200 * k)	0x01D1 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k (3)	R	32	0x0000 206C + (0x200 * k)	0x01D1 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k (3)	W	32	0x0000 2070 + (0x200 * k)	0x01D1 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k (3)	W	32	0x0000 2074 + (0x200 * k)	0x01D1 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k (3)	W	32	0x0000 2078 + (0x200 * k)	0x01D1 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k (3)	R	32	0x0000 2080 + (0x200 * k)	0x01D1 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k (3)	R	32	0x0000 2084 + (0x200 * k)	0x01D1 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k (3)	W	32	0x0000 2088 + (0x200 * k)	0x01D1 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k (3)	W	32	0x0000 208C + (0x200 * k)	0x01D1 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k (3)	R	32	0x0000 2090 + (0x200 * k)	0x01D1 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k (3)	W	32	0x0000 2094 + (0x200 * k)	0x01D1 2094 + (0x200 * k)
EDMA_TPCC_OPT_n (6)	RW	32	0x0000 4000 + (0x20 * n)	0x01D1 4000 + (0x20 * n)
EDMA_TPCC_SRC_n (6)	RW	32	0x0000 4004 + (0x20 * n)	0x01D1 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n (6)	RW	32	0x0000 4008 + (0x20 * n)	0x01D1 4008 + (0x20 * n)
EDMA_TPCC_DST_n (6)	RW	32	0x0000 400C + (0x20 * n)	0x01D1 400C + (0x20 * n)
EDMA_TPCC_BIDX_n (6)	RW	32	0x0000 4010 + (0x20 * n)	0x01D1 4010 + (0x20 * n)
EDMA_TPCC_LNK_n (6)	RW	32	0x0000 4014 + (0x20 * n)	0x01D1 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n (6)	RW	32	0x0000 4018 + (0x20 * n)	0x01D1 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n (6)	RW	32	0x0000 401C + (0x20 * n)	0x01D1 401C + (0x20 * n)

(1) i = 0 to 1 for DSP_EDMA_TPCC
(2) j = 0 to 7 for DSP_EDMA_TPCC
(3) k = 0 to 7 for DSP_EDMA_TPCC
(4) p = 0 to 15 for DSP_EDMA_TPCC
(5) m = 0 to 63 for DSP_EDMA_TPCC
(6) n = 0 to 127 for DSP_EDMA_TPCC

Table 16-84. EVE1 and EVE2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPCC Physical Address (L3_MAIN Access)	EVE2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x420A 0000	0x421A 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x420A 0004	0x421A 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x420A 00FC	0x421A 00FC
EDMA_TPCC_DCHMAPN_m (5)	RW	32	0x0000 0100 + (0x4 * m)	0x420A 0100 + (0x4 * m)	0x421A 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j (2)	RW	32	0x0000 0200 + (0x4 * j)	0x420A 0200 + (0x4 * j)	0x421A 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUM_k (3)	RW	32	0x0000 0240 + (0x4 * k)	0x420A 0240 + (0x4 * k)	0x421A 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x420A 0260	0x421A 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x420A 0280	0x421A 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x420A 0284	0x421A 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x420A 0300	0x421A 0300

Table 16-84. EVE1 and EVE2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPCC Physical Address (L3_MAIN Access)	EVE2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x420A 0304	0x421A 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x420A 0308	0x421A 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x420A 030C	0x421A 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x420A 0310	0x421A 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x420A 0314	0x421A 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x420A 0318	0x421A 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x420A 031C	0x421A 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x420A 0320	0x421A 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	-	-
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	-	-
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x420A 0380 + (0x4 * k)	0x421A 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 * p)	-	-
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 *pl)	-	-
EDMA_TPCC_QSTATN_i (1)	R	32	0x0000 0600 + (0x4 * i)	0x420A 0600 + (0x4 * i)	0x421A 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x420A 0620	0x421A 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x420A 0624	0x421A 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x420A 0640	0x421A 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x420A 0700	0x421A 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x420A 0704	0x421A 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x420A 0708	0x421A 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x420A 0800	0x421A 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x420A 0804	0x421A 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x420A 0808	0x421A 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x420A 080C	0x421A 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x420A 0810 + (0x4 * k)	0x421A 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x420A 1000	0x421A 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x420A 1004	0x421A 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x420A 1008	0x421A 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x420A 100C	0x421A 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x420A 1010	0x421A 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x420A 1014	0x421A 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x420A 1018	0x421A 1018
EDMA_TPCC_CERH	R	32	0x0000 101C	0x420A 101C	0x421A 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x420A 1020	0x421A 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x420A 1024	0x421A 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x420A 1028	0x421A 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x420A 102C	0x421A 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x420A 1030	0x421A 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x420A 1034	0x421A 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x420A 1038	0x421A 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x420A 103C	0x421A 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x420A 1040	0x421A 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x420A 1044	0x421A 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x420A 1050	0x421A 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x420A 1054	0x421A 1054

Table 16-84. EVE1 and EVE2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPCC Physical Address (L3_MAIN Access)	EVE2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_IECR	W	32	0x0000 1058	0x420A 1058	0x421A 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x420A 105C	0x421A 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x420A 1060	0x421A 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x420A 1064	0x421A 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x420A 1068	0x421A 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x420A 106C	0x421A 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x420A 1070	0x421A 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x420A 1074	0x421A 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x420A 1078	0x421A 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x420A 1080	0x421A 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x420A 1084	0x421A 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x420A 1088	0x421A 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x420A 108C	0x421A 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x420A 1090	0x421A 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x420A 1094	0x421A 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x420A 2000 + (0x200 * k)	0x421A 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x420A 2004 + (0x200 * k)	0x421A 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x420A 2008 + (0x200 * k)	0x421A 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x420A 200C + (0x200 * k)	0x421A 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x420A 2010 + (0x200 * k)	0x421A 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x420A 2014 + (0x200 * k)	0x421A 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x420A 2018 + (0x200 * k)	0x421A 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x420A 201C + (0x200 * k)	0x421A 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x420A 2020 + (0x200 * k)	0x421A 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x420A 2024 + (0x200 * k)	0x421A 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x420A 2028 + (0x200 * k)	0x421A 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x420A 202C + (0x200 * k)	0x421A 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x420A 2030 + (0x200 * k)	0x421A 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x420A 2034 + (0x200 * k)	0x421A 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x420A 2038 + (0x200 * k)	0x421A 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x420A 203C + (0x200 * k)	0x421A 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k (3)	W	32	0x0000 2040 + (0x200 * k)	0x420A 2040 + (0x200 * k)	0x421A 2040 + (0x200 * k)

Table 16-84. EVE1 and EVE2 EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPCC Physical Address (L3_MAIN Access)	EVE2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x420A 2044 + (0x200 * k)	0x421A 2044 + (0x200 * k)
EDMA_TPCC_IER_RN_k (3)	R	32	0x0000 2050 + (0x200 * k)	0x420A 2050 + (0x200 * k)	0x421A 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k (3)	R	32	0x0000 2054 + (0x200 * k)	0x420A 2054 + (0x200 * k)	0x421A 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k (3)	W	32	0x0000 2058 + (0x200 * k)	0x420A 2058 + (0x200 * k)	0x421A 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k (3)	W	32	0x0000 205C + (0x200 * k)	0x420A 205C + (0x200 * k)	0x421A 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k (3)	W	32	0x0000 2060 + (0x200 * k)	0x420A 2060 + (0x200 * k)	0x421A 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k (3)	W	32	0x0000 2064 + (0x200 * k)	0x420A 2064 + (0x200 * k)	0x421A 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k (3)	R	32	0x0000 2068 + (0x200 * k)	0x420A 2068 + (0x200 * k)	0x421A 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k (3)	R	32	0x0000 206C + (0x200 * k)	0x420A 206C + (0x200 * k)	0x421A 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k (3)	W	32	0x0000 2070 + (0x200 * k)	0x420A 2070 + (0x200 * k)	0x421A 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k (3)	W	32	0x0000 2074 + (0x200 * k)	0x420A 2074 + (0x200 * k)	0x421A 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k (3)	W	32	0x0000 2078 + (0x200 * k)	0x420A 2078 + (0x200 * k)	0x421A 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k (3)	R	32	0x0000 2080 + (0x200 * k)	0x420A 2080 + (0x200 * k)	0x421A 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k (3)	R	32	0x0000 2084 + (0x200 * k)	0x420A 2084 + (0x200 * k)	0x421A 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k (3)	W	32	0x0000 2088 + (0x200 * k)	0x420A 2088 + (0x200 * k)	0x421A 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k (3)	W	32	0x0000 208C + (0x200 * k)	0x420A 208C + (0x200 * k)	0x421A 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k (3)	R	32	0x0000 2090 + (0x200 * k)	0x420A 2090 + (0x200 * k)	0x421A 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k (3)	W	32	0x0000 2094 + (0x200 * k)	0x420A 2094 + (0x200 * k)	0x421A 2094 + (0x200 * k)
EDMA_TPCC_OPT_n (6)	RW	32	0x0000 4000 + (0x20 * n)	0x420A 4000 + (0x20 * n)	0x421A 4000 + (0x20 * n)
EDMA_TPCC_SRC_n (6)	RW	32	0x0000 4004 + (0x20 * n)	0x420A 4004 + (0x20 * n)	0x421A 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n (6)	RW	32	0x0000 4008 + (0x20 * n)	0x420A 4008 + (0x20 * n)	0x421A 4008 + (0x20 * n)
EDMA_TPCC_DST_n (6)	RW	32	0x0000 400C + (0x20 * n)	0x420A 400C + (0x20 * n)	0x421A 400C + (0x20 * n)
EDMA_TPCC_BIDX_n (6)	RW	32	0x0000 4010 + (0x20 * n)	0x420A 4010 + (0x20 * n)	0x421A 4010 + (0x20 * n)
EDMA_TPCC_LNK_n (6)	RW	32	0x0000 4014 + (0x20 * n)	0x420A 4014 + (0x20 * n)	0x421A 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n (6)	RW	32	0x0000 4018 + (0x20 * n)	0x420A 4018 + (0x20 * n)	0x421A 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n (6)	RW	32	0x0000 401C + (0x20 * n)	0x420A 401C + (0x20 * n)	0x421A 401C + (0x20 * n)

(1) i = 0 to 1 for EVE1_EDMA_TPCC and EVE2_EDMA_TPCC
(2) j = 0 to 7 for EVE1_EDMA_TPCC and EVE2_EDMA_TPCC
(3) k = 0 to 7 for EVE1_EDMA_TPCC and EVE2_EDMA_TPCC
(4) p = 0 to 1 for EVE1_EDMA_TPCC and EVE2_EDMA_TPCC
(5) m = 0 to 15 for EVE1_EDMA_TPCC and EVE2_EDMA_TPCC
(6) n = 0 to 127 for EVE1_EDMA_TPCC and EVE2_EDMA_TPCC

Note

The value for "n" is from 0 to 1 in the Table 16-85. It corresponds of the Transfer Controller (EDMA_TPTC0 and EDMA_TPTC1) instances in the device.

Table 16-85. System EDMA TPTC0 and EDMA TPTC1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPTC0 Physical Address (L3_MAIN Access)	SYS_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	0x4340 0000	0x4350 0000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x4340 0004	0x4350 0004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x4340 0100	0x4350 0100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x4340 0104	0x4350 0104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x4340 0108	0x4350 0108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x4340 010C	0x4350 010C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x4340 0110	0x4350 0110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x4340 0120	0x4350 0120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x4340 0124	0x4350 0124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x4340 0128	0x4350 0128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x4340 012C	0x4350 012C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x4340 0130	0x4350 0130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x4340 0140	0x4350 0140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x4340 0200	0x4350 0200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x4340 0204	0x4350 0204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x4340 0208	0x4350 0208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x4340 020C	0x4350 020C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x4340 0210	0x4350 0210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x4340 0214	0x4350 0214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x4340 0240	0x4350 0240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x4340 0244	0x4350 0244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x4340 0248	0x4350 0248
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x4340 024C	0x4350 024C
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x4340 0250	0x4350 0250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x4340 0254	0x4350 0254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x4340 0258	0x4350 0258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x4340 025C	0x4350 025C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x4340 0260	0x4350 0260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x4340 0280	0x4350 0280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x4340 0284	0x4350 0284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x4340 0288	0x4350 0288
EDMA_TPTCn_DFOPTi (1)	R	32	0x0000 0300 + (0x40 * i)	0x4340 0300 + (0x40 * i)	0x4350 0300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x4340 0304 + (0x40 * i)	0x4350 0304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x4340 0308 + (0x40 * i)	0x4350 0308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x4340 030C + (0x40 * i)	0x4350 030C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x4340 0310 + (0x40 * i)	0x4350 0310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x4340 0314 + (0x40 * i)	0x4350 0314 + (0x40 * i)

(1) i = 0 to 1 for SYS_EDMA_TPTC0 and SYS_EDMA_TPTC1

Table 16-86. DSP1 EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPTC0 Physical Address (L3_MAIN Access)	DSP1_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	0x40D0 5000	0x40D0 6000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x40D0 5004	0x40D0 6004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x40D0 5100	0x40D0 6100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x40D0 5104	0x40D0 6104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x40D0 5108	0x40D0 6108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x40D0 510C	0x40D0 610C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x40D0 5110	0x40D0 6110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x40D0 5120	0x40D0 6120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x40D0 5124	0x40D0 6124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x40D0 5128	0x40D0 6128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x40D0 512C	0x40D0 612C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x40D0 5130	0x40D0 6130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x40D0 5140	0x40D0 6140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x40D0 5200	0x40D0 6200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x40D0 5204	0x40D0 6204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x40D0 5208	0x40D0 6208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x40D0 520C	0x40D0 620C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x40D0 5210	0x40D0 6210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x40D0 5214	0x40D0 6214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x40D0 5240	0x40D0 6240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x40D0 5244	0x40D0 6244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x40D0 5248	0x40D0 6248
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x40D0 524C	0x40D0 624C
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x40D0 5250	0x40D0 6250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x40D0 5254	0x40D0 6254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x40D0 5258	0x40D0 6258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x40D0 525C	0x40D0 625C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x40D0 5260	0x40D0 6260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x40D0 5280	0x40D0 6280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x40D0 5284	0x40D0 6284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x40D0 5288	0x40D0 6288
EDMA_TPTCn_DFOPTi (1)	R	32	0x0000 0300 + (0x40 * i)	0x40D0 5300 + (0x40 * i)	0x40D0 6300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x40D0 5304 + (0x40 * i)	0x40D0 6304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x40D0 5308 + (0x40 * i)	0x40D0 6308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x40D0 530C + (0x40 * i)	0x40D0 630C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x40D0 5310 + (0x40 * i)	0x40D0 6310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x40D0 5314 + (0x40 * i)	0x40D0 6314 + (0x40 * i)

(1) i = 0 to 1 for DSP1_EDMA_TPTC0 and DSP1_EDMA_TPTC1

Table 16-87. DSP2 EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_EDMA_TPTC0 Physical Address (L3_MAIN Access)	DSP2_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	0x4150 5000	0x4150 6000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x4150 5004	0x4150 6004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x4150 5100	0x4150 6100

Table 16-87. DSP2 EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_EDMA_TPTC0 Physical Address (L3_MAIN Access)	DSP2_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x4150 5104	0x4150 6104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x4150 5108	0x4150 6108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x4150 510C	0x4150 610C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x4150 5110	0x4150 6110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x4150 5120	0x4150 6120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x4150 5124	0x4150 6124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x4150 5128	0x4150 6128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x4150 512C	0x4150 612C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x4150 5130	0x4150 6130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x4150 5140	0x4150 6140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x4150 5200	0x4150 6200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x4150 5204	0x4150 6204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x4150 5208	0x4150 6208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x4150 520C	0x4150 620C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x4150 5210	0x4150 6210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x4150 5214	0x4150 6214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x4150 5240	0x4150 6240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x4150 5244	0x4150 6244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x4150 5248	0x4150 6248
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x4150 524C	0x4150 624C
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x4150 5250	0x4150 6250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x4150 5254	0x4150 6254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x4150 5258	0x4150 6258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x4150 525C	0x4150 625C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x4150 5260	0x4150 6260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x4150 5280	0x4150 6280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x4150 5284	0x4150 6284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x4150 5288	0x4150 6288
EDMA_TPTCn_DFOPTi (1)	R	32	0x0000 0300 + (0x40 * i)	0x4150 5300 + (0x40 * i)	0x4150 6300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x4150 5304 + (0x40 * i)	0x4150 6304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x4150 5308 + (0x40 * i)	0x4150 6308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x4150 530C + (0x40 * i)	0x4150 630C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x4150 5310 + (0x40 * i)	0x4150 6310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x4150 5314 + (0x40 * i)	0x4150 6314 + (0x40 * i)

(1) i = 0 to 1 for DSP2_EDMA_TPTC0 and DSP2_EDMA_TPTC1

Table 16-88. DSP EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (Private Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPTC0 Physical Address (DSP Private Access)	DSP_EDMA_TPTC1 Physical Address (DSP Private Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	0x01D0 5000	0x01D0 6000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x01D0 5004	0x01D0 6004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x01D0 5100	0x01D0 6100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x01D0 5104	0x01D0 6104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x01D0 5108	0x01D0 6108

Table 16-88. DSP EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (Private Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPTC0 Physical Address (DSP Private Access)	DSP_EDMA_TPTC1 Physical Address (DSP Private Access)
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x01D0 510C	0x01D0 610C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x01D0 5110	0x01D0 6110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x01D0 5120	0x01D0 6120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x01D0 5124	0x01D0 6124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x01D0 5128	0x01D0 6128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x01D0 512C	0x01D0 612C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x01D0 5130	0x01D0 6130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x01D0 5140	0x01D0 6140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x01D0 5200	0x01D0 6200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x01D0 5204	0x01D0 6204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x01D0 5208	0x01D0 6208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x01D0 520C	0x01D0 620C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x01D0 5210	0x01D0 6210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x01D0 5214	0x01D0 6214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x01D0 5240	0x01D0 6240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x01D0 5244	0x01D0 6244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x01D0 5248	0x01D0 6248
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x01D0 524C	0x01D0 624C
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x01D0 5250	0x01D0 6250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x01D0 5254	0x01D0 6254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x01D0 5258	0x01D0 6258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x01D0 525C	0x01D0 625C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x01D0 5260	0x01D0 6260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x01D0 5280	0x01D0 6280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x01D0 5284	0x01D0 6284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x01D0 5288	0x01D0 6288
EDMA_TPTCn_DFOPTi (1)	R	32	0x0000 0300 + (0x40 * i)	0x01D0 5300 + (0x40 * i)	0x01D0 6300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x01D0 5304 + (0x40 * i)	0x01D0 6304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x01D0 5308 + (0x40 * i)	0x01D0 6308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x01D0 530C + (0x40 * i)	0x01D0 630C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x01D0 5310 + (0x40 * i)	0x01D0 6310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x01D0 5314 + (0x40 * i)	0x01D0 6314 + (0x40 * i)

(1) i = 0 to 1 for DSP_EDMA_TPTC0 and DSP_EDMA_TPTC1

Table 16-89. EVE1 and EVE2 EDMA_TPTC0 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPTC0 Physical Address (L3_MAIN Access)	EVE2_EDMA_TPTC0 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	-	-
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	-	-
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x4208 6100	0x4218 6100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x4208 6104	0x4218 6104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x4208 6108	0x4218 6108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x4208 610C	0x4218 610C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x4208 6110	0x4218 6110

Table 16-89. EVE1 and EVE2 EDMA_TPTC0 Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPTC0 Physical Address (L3_MAIN Access)	EVE2_EDMA_TPTC0 Physical Address (L3_MAIN Access)
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x4208 6120	0x4218 6120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x4208 6124	0x4218 6124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x4208 6128	0x4218 6128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x4208 612C	0x4218 612C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x4208 6130	0x4218 6130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x4208 6140	0x4218 6140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x4208 6200	0x4218 6200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x4208 6204	0x4218 6204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x4208 6208	0x4218 6208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x4208 620C	0x4218 620C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x4208 6210	0x4218 6210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x4208 6214	0x4218 6214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x4208 6240	0x4218 6240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x4208 6244	0x4218 6244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x4208 6248	0x4218 6248
EDMA_TPTCn_SADST	R	32	0x0000 024C	-	-
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x4208 6250	0x4218 6250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x4208 6254	0x4218 6254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x4208 6258	0x4218 6258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x4208 625C	0x4218 625C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x4208 6260	0x4218 6260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x4208 6280	0x4218 6280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x4208 6284	0x4218 6284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	-	-
EDMA_TPTCn_DFOPTi (1)	R	32	0x0000 0300 + (0x40 * i)	0x4208 6300 + (0x40 * i)	0x4218 6300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x4208 6304 + (0x40 * i)	0x4218 6304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x4208 6308 + (0x40 * i)	0x4218 6308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x4208 630C + (0x40 * i)	0x4218 630C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x4208 6310 + (0x40 * i)	0x4218 6310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x4208 6314 + (0x40 * i)	0x4218 6314 + (0x40 * i)

(1) i = 0 to 1 for EVE1_EDMA_TPTC0 and EVE2_EDMA_TPTC0

Table 16-90. EVE1 and EVE2 EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPTC1 Physical Address (L3_MAIN Access)	EVE2_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	-	-
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	-	-
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x4208 7100	0x4218 7100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x4208 7104	0x4218 7104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x4208 7108	0x4218 7108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x4208 710C	0x4218 710C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x4208 7110	0x4218 7110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x4208 7120	0x4218 7120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x4208 7124	0x4218 7124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x4208 7128	0x4218 7128

Table 16-90. EVE1 and EVE2 EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_EDMA_TPTC1 Physical Address (L3_MAIN Access)	EVE2_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x4208 712C	0x4218 712C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x4208 7130	0x4218 7130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x4208 7140	0x4218 7140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x4208 7200	0x4218 7200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x4208 7204	0x4218 7204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x4208 7208	0x4218 7208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x4208 720C	0x4218 720C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x4208 7210	0x4218 7210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x4208 7214	0x4218 7214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x4208 7240	0x4218 7240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x4208 7244	0x4218 7244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x4208 7248	0x4218 7248
EDMA_TPTCn_SADST	R	32	0x0000 024C	-	-
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x4208 7250	0x4218 7250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x4208 7254	0x4218 7254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x4208 7258	0x4218 7258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x4208 725C	0x4218 725C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x4208 7260	0x4218 7260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x4208 7280	0x4218 7280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x4208 7284	0x4218 7284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	-	-
EDMA_TPTCn_DFOPTi (1)	R	32	0x0000 0300 + (0x40 * i)	0x4208 7300 + (0x40 * i)	0x4218 7300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x4208 7304 + (0x40 * i)	0x4218 7304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x4208 7308 + (0x40 * i)	0x4218 7308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x4208 730C + (0x40 * i)	0x4218 730C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x4208 7310 + (0x40 * i)	0x4218 7310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x4208 7314 + (0x40 * i)	0x4218 7314 + (0x40 * i)

(1) i = 0 to 1 for EVE1_EDMA_TPTC1 and EVE2_EDMA_TPTC1

16.2.7.2.2 EDMA Register Description

16.2.7.2.2.1 EDMA_TPCC Register Description

Table 16-91 through Table 16-203 describe the EDMA_TPCC module registers.

Table 16-91. EDMA_TPCC_PID

Address Offset	0x0000 0000		
Physical Address	0x4330 0000 0x40D1 0000 0x4151 0000 0x01D1 0000 0x420A 0000 0x421A 0000	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Peripheral ID Register		

Table 16-91. EDMA_TPCC_PID (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 16-92. EDMA_TPCC_CCCFG

Address Offset	0x0000 0004		
Physical Address	0x4330 0004 0x40D1 0004 0x4151 0004 0x01D1 0004 0x420A 0004 0x421A 0004	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	CC Configuration Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MPEXIST	CHMAPEXIST	RESERVED		NUMREGN		RESERVED	NUMTC			RESERVED	NUMPAENTRY			RESERVED	NUMINTCH			RESERVED	NUMQDMACH			RESERVED	NUMDMACH		

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reads return 0's	R	0x0
25	MPEXIST	Memory Protection Existence 0x0: No memory protection 0x1: Memory Protection logic included	R	See Table 16-52
24	CHMAPEXIST	Channel Mapping Existence 0x0: No Channel mapping 0x1: Channel mapping logic included	R	See Table 16-52
23:22	RESERVED	Reads return 0's	R	0x0
21:20	NUMREGN	Number of MP and Shadow regions 0x0: 0 Regions 0x1: 2 Regions 0x2: 4 Regions 0x3: 8 Regions	R	See Table 16-52
19	RESERVED	Reads return 0's	R	0x0
18:16	NUMTC	Number of Queues/Number of TCs 0x0: 1 TC/Event Queue 0x1: 2 TC/Event Queue 0x2: 3 TC/Event Queue 0x3: 4 TC/Event Queue 0x4: 5 TC/Event Queue 0x5: 6 TC/Event Queue 0x6: 7 TC/Event Queue 0x7: 8 TC/Event Queue	R	See Table 16-52

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reads return 0's	R	0x0
14:12	NUMPAENTRY	Number of PaRAM entries 0x0: 16 entries 0x1: 32 entries 0x2: 64 entries 0x3: 128 entries 0x4: 256 entries 0x5: 512 entries	R	See Table 16-52
11	RESERVED	Reads return 0's	R	0x0
10:8	NUMINTCH	Number of Interrupt Channels 0x1: 8 Interrupt channels 0x2: 16 Interrupt channels 0x3: 32 Interrupt channels 0x4: 64 Interrupt channels	R	See Table 16-52
7	RESERVED	reads return 0's	R	0x0
6:4	NUMQDMACH	Number of QDMA Channels 0x0: No QDMA Channels 0x1: 2 QDMA Channels 0x2: 4 QDMA Channels 0x3: 6 QDMA Channels 0x4: 8 QDMA Channels	R	See Table 16-52
3	RESERVED	reads return 0's	R	0x0
2:0	NUMDMACH	Number of DMA Channels 0x0: No DMA Channels 0x1: 4 DMA Channels 0x2: 8 DMA Channels 0x3: 16 DMA Channels 0x4: 32 DMA Channels 0x5: 64 DMA Channels	R	See Table 16-52

Table 16-93. EDMA_TPCC_CLKGDIS

Address Offset	0x0000 00FC		
Physical Address	0x4330 00FC 0x40D1 00FC 0x4151 00FC 0x01D1 00FC 0x420A 00FC 0x421A 00FC	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Auto Clock Gate Disable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKGDIS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	CLKGDIS	Auto Clock Gate Disable	RW	0x0

Table 16-94. EDMA_TPCC_DCHMAPN_m

Address Offset	0x0000 0100 + (0x4 * m)		
Physical Address	0x4330 0100 + (0x4 * m) 0x40D1 0100 + (0x4 * m) 0x4151 0100 + (0x4 * m) 0x01D1 0100 + (0x4 * m) 0x420A 0100 + (0x4 * m) 0x421A 0100 + (0x4 * m)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	DMA Channel N Mapping Register		
Type	RW		

31:14	13:5	4:0
RESERVED	PAENTRY	RESERVED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x0
13:5	PAENTRY	PaRAM Entry number for DMA Channel N.	RW	0x0
4:0	RESERVED	Reserved	R	0x0

Table 16-95. EDMA_TPCC_QCHMAPN_j

Address Offset	0x0000 0200 + (0x4 * j)		
Physical Address	0x4330 0200 + (0x4 * j) 0x40D1 0200 + (0x4 * j) 0x4151 0200 + (0x4 * j) 0x01D1 0200 + (0x4 * j) 0x420A 0200 + (0x4 * j) 0x421A 0200 + (0x4 * j)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Channel N Mapping Register		
Type	RW		

31:14	13:5	4:2	1:0
RESERVED	PAENTRY	TRWORD	RESERVED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x0
13:5	PAENTRY	PaRAM Entry number for QDMA Channel N.	RW	0x0
4:2	TRWORD	TRWORD points to the specific trigger word of the PaRAM Entry defined by PAENTRY. A write to the trigger word results in a QDMA Event being recognized.	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 16-96. EDMA_TPCC_DMAQNUMN_k

Address Offset	0x0000 0240 + (0x4 * k)		
Physical Address	0x4330 0240 + (0x4 * k) 0x40D1 0240 + (0x4 * k) 0x4151 0240 + (0x4 * k) 0x01D1 0240 + (0x4 * k) 0x420A 0240 + (0x4 * k) 0x421A 0240 + (0x4 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	DMA Queue Number Register n Contains the Event queue number to be used for the corresponding DMA Channel.		

Table 16-96. EDMA_TPCC_DMAQNUMN_k (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	E7			RESERVED	E6			RESERVED	E5			RESERVED	E4			RESERVED	E3			RESERVED	E2			RESERVED	E1			RESERVED	E0		

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0x0
30:28	E7	DMA Queue Number for event #7	RW	0x0
27	RESERVED	Reserved	R	0x0
26:24	E6	DMA Queue Number for event #6	RW	0x0
23	RESERVED	Reserved	R	0x0
22:20	E5	DMA Queue Number for event #5	RW	0x0
19	RESERVED	Reserved	R	0x0
18:16	E4	DMA Queue Number for event #4	RW	0x0
15	RESERVED	Reserved	R	0x0
14:12	E3	DMA Queue Number for event #3	RW	0x0
11	RESERVED	Reserved	R	0x0
10:8	E2	DMA Queue Number for event #2	RW	0x0
7	RESERVED	Reserved	R	0x0
6:4	E1	DMA Queue Number for event #1	RW	0x0
3	RESERVED	Reserved	R	0x0
2:0	E0	DMA Queue Number for event #0	RW	0x0

Table 16-97. EDMA_TPCC_QDMAQNUM

Address Offset	0x0000 0260		
Physical Address	0x4330 0260 0x40D1 0260 0x4151 0260 0x01D1 0260 0x420A 0260 0x421A 0260	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Queue Number Register Contains the Event queue number to be used for the corresponding QDMA Channel.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	E7			RESERVED	E6			RESERVED	E5			RESERVED	E4			RESERVED	E3			RESERVED	E2			RESERVED	E1			RESERVED	E0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	E7	QDMA Queue Number for event #7	RW	0x0
27	RESERVED		R	0x0
26:24	E6	QDMA Queue Number for event #6	RW	0x0
23	RESERVED		R	0x0
22:20	E5	QDMA Queue Number for event #5	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	E4	QDMA Queue Number for event #4	RW	0x0
15	RESERVED		R	0x0
14:12	E3	QDMA Queue Number for event #3	RW	0x0
11	RESERVED		R	0x0
10:8	E2	QDMA Queue Number for event #2	RW	0x0
7	RESERVED		R	0x0
6:4	E1	QDMA Queue Number for event #1	RW	0x0
3	RESERVED		R	0x0
2:0	E0	QDMA Queue Number for event #0	RW	0x0

Table 16-98. EDMA_TPCC_QUETCMAP

Address Offset	0x0000 0280		
Physical Address	0x4330 0280 0x40D1 0280 0x4151 0280 0x01D1 0280 0x420A 0280 0x421A 0280	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Queue to TC Mapping		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									TCNUMQ1			RESERVED	TCNUMQ0		

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	TCNUMQ1	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.	RW	0x1
3	RESERVED	Reserved	R	0x0
2:0	TCNUMQ0	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.	RW	0x0

Table 16-99. EDMA_TPCC_QUEPRI

Address Offset	0x0000 0284		
Physical Address	0x4330 0284 0x40D1 0284 0x4151 0284 0x01D1 0284 0x420A 0284 0x421A 0284	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Queue Priority		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									PRIQ1			RESERVED	PRIQ0		

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
6:4	PRIQ1	Priority Level for Queue 1 Dictates the priority level used for the OPTIONS field programmation for Qn TRs. Sets the priority used for TC read and write commands.	RW	0x0
3	RESERVED	Reserved	R	0x0
2:0	PRIQ0	Priority Level for Queue 0 Dictates the priority level used for the OPTIONS field programmation for Qn TRs. Sets the priority used for TC read and write commands.	RW	0x0

Table 16-100. EDMA_TPCC_EMR

Address Offset	0x0000 0300		
Physical Address	0x4330 0300 0x40D1 0300 0x4151 0300 0x01D1 0300 0x420A 0300 0x421A 0300	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Missed Register: The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER), Set Events (ESR), and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including EDMA_TPCC_QEMR / EDMA_TPCC_CCERR) were previously clear), then an error will be signaled with TPCC error interrupt.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event Missed #31	R	0x0
30	E30	Event Missed #30	R	0x0
29	E29	Event Missed #29	R	0x0
28	E28	Event Missed #28	R	0x0
27	E27	Event Missed #27	R	0x0
26	E26	Event Missed #26	R	0x0
25	E25	Event Missed #25	R	0x0
24	E24	Event Missed #24	R	0x0
23	E23	Event Missed #23	R	0x0
22	E22	Event Missed #22	R	0x0
21	E21	Event Missed #21	R	0x0
20	E20	Event Missed #20	R	0x0
19	E19	Event Missed #19	R	0x0
18	E18	Event Missed #18	R	0x0
17	E17	Event Missed #17	R	0x0
16	E16	Event Missed #16	R	0x0
15	E15	Event Missed #15	R	0x0
14	E14	Event Missed #14	R	0x0
13	E13	Event Missed #13	R	0x0
12	E12	Event Missed #12	R	0x0
11	E11	Event Missed #11	R	0x0
10	E10	Event Missed #10	R	0x0
9	E9	Event Missed #9	R	0x0

Bits	Field Name	Description	Type	Reset
8	E8	Event Missed #8	R	0x0
7	E7	Event Missed #7	R	0x0
6	E6	Event Missed #6	R	0x0
5	E5	Event Missed #5	R	0x0
4	E4	Event Missed #4	R	0x0
3	E3	Event Missed #3	R	0x0
2	E2	Event Missed #2	R	0x0
1	E1	Event Missed #1	R	0x0
0	E0	Event Missed #0	R	0x0

Table 16-101. EDMA_TPCC_EMRH

Address Offset	0x0000 0304		
Physical Address	0x4330 0304 0x40D1 0304 0x4151 0304 0x01D1 0304 0x420A 0304 0x421A 0304	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Missed Register (High Part): The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER), Set Events (ESR), and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including EDMA_TPCC_QEMR / EDMA_TPCC_CCERR) were previously clear), then an error will be signaled with TPCC error interrupt.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event Missed #63	R	0x0
30	E62	Event Missed #62	R	0x0
29	E61	Event Missed #61	R	0x0
28	E60	Event Missed #60	R	0x0
27	E59	Event Missed #59	R	0x0
26	E58	Event Missed #58	R	0x0
25	E57	Event Missed #57	R	0x0
24	E56	Event Missed #56	R	0x0
23	E55	Event Missed #55	R	0x0
22	E54	Event Missed #54	R	0x0
21	E53	Event Missed #53	R	0x0
20	E52	Event Missed #52	R	0x0
19	E51	Event Missed #51	R	0x0
18	E50	Event Missed #50	R	0x0
17	E49	Event Missed #49	R	0x0
16	E48	Event Missed #48	R	0x0
15	E47	Event Missed #47	R	0x0
14	E46	Event Missed #46	R	0x0
13	E45	Event Missed #45	R	0x0
12	E44	Event Missed #44	R	0x0

Bits	Field Name	Description	Type	Reset
11	E43	Event Missed #43	R	0x0
10	E42	Event Missed #42	R	0x0
9	E41	Event Missed #41	R	0x0
8	E40	Event Missed #40	R	0x0
7	E39	Event Missed #39	R	0x0
6	E38	Event Missed #38	R	0x0
5	E37	Event Missed #37	R	0x0
4	E36	Event Missed #36	R	0x0
3	E35	Event Missed #35	R	0x0
2	E34	Event Missed #34	R	0x0
1	E33	Event Missed #33	R	0x0
0	E32	Event Missed #32	R	0x0

Table 16-102. EDMA_TPCC_EMCR

Address Offset	0x0000 0308		
Physical Address	0x4330 0308 0x40D1 0308 0x4151 0308 0x01D1 0308 0x420A 0308 0x421A 0308	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Missed Clear Register: CPU write of '1' to the EDMA_TPCC_EMCR.En bit causes the EDMA_TPCC_EMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event Missed Clear #31	W	0x0
30	E30	Event Missed Clear #30	W	0x0
29	E29	Event Missed Clear #29	W	0x0
28	E28	Event Missed Clear #28	W	0x0
27	E27	Event Missed Clear #27	W	0x0
26	E26	Event Missed Clear #26	W	0x0
25	E25	Event Missed Clear #25	W	0x0
24	E24	Event Missed Clear #24	W	0x0
23	E23	Event Missed Clear #23	W	0x0
22	E22	Event Missed Clear #22	W	0x0
21	E21	Event Missed Clear #21	W	0x0
20	E20	Event Missed Clear #20	W	0x0
19	E19	Event Missed Clear #19	W	0x0
18	E18	Event Missed Clear #18	W	0x0
17	E17	Event Missed Clear #17	W	0x0
16	E16	Event Missed Clear #16	W	0x0
15	E15	Event Missed Clear #15	W	0x0
14	E14	Event Missed Clear #14	W	0x0
13	E13	Event Missed Clear #13	W	0x0

Bits	Field Name	Description	Type	Reset
12	E12	Event Missed Clear #12	W	0x0
11	E11	Event Missed Clear #11	W	0x0
10	E10	Event Missed Clear #10	W	0x0
9	E9	Event Missed Clear #9	W	0x0
8	E8	Event Missed Clear #8	W	0x0
7	E7	Event Missed Clear #7	W	0x0
6	E6	Event Missed Clear #6	W	0x0
5	E5	Event Missed Clear #5	W	0x0
4	E4	Event Missed Clear #4	W	0x0
3	E3	Event Missed Clear #3	W	0x0
2	E2	Event Missed Clear #2	W	0x0
1	E1	Event Missed Clear #1	W	0x0
0	E0	Event Missed Clear #0	W	0x0

Table 16-103. EDMA_TPCC_EMCRH

Address Offset	0x0000 030C		
Physical Address	0x4330 030C 0x40D1 030C 0x4151 030C 0x01D1 030C 0x420A 030C 0x421A 030C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Missed Clear Register (High Part): CPU write of '1' to the EDMA_TPCC_EMCR.En bit causes the EDMA_TPCC_EMR.En bit to be cleared. CPU write of '0' has no effect. All error bits must be cleared before additional error interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event Missed Clear #63	W	0x0
30	E62	Event Missed Clear #62	W	0x0
29	E61	Event Missed Clear #61	W	0x0
28	E60	Event Missed Clear #60	W	0x0
27	E59	Event Missed Clear #59	W	0x0
26	E58	Event Missed Clear #58	W	0x0
25	E57	Event Missed Clear #57	W	0x0
24	E56	Event Missed Clear #56	W	0x0
23	E55	Event Missed Clear #55	W	0x0
22	E54	Event Missed Clear #54	W	0x0
21	E53	Event Missed Clear #53	W	0x0
20	E52	Event Missed Clear #52	W	0x0
19	E51	Event Missed Clear #51	W	0x0
18	E50	Event Missed Clear #50	W	0x0
17	E49	Event Missed Clear #49	W	0x0
16	E48	Event Missed Clear #48	W	0x0
15	E47	Event Missed Clear #47	W	0x0
14	E46	Event Missed Clear #46	W	0x0

Bits	Field Name	Description	Type	Reset
13	E45	Event Missed Clear #45	W	0x0
12	E44	Event Missed Clear #44	W	0x0
11	E43	Event Missed Clear #43	W	0x0
10	E42	Event Missed Clear #42	W	0x0
9	E41	Event Missed Clear #41	W	0x0
8	E40	Event Missed Clear #40	W	0x0
7	E39	Event Missed Clear #39	W	0x0
6	E38	Event Missed Clear #38	W	0x0
5	E37	Event Missed Clear #37	W	0x0
4	E36	Event Missed Clear #36	W	0x0
3	E35	Event Missed Clear #35	W	0x0
2	E34	Event Missed Clear #34	W	0x0
1	E33	Event Missed Clear #33	W	0x0
0	E32	Event Missed Clear #32	W	0x0

Table 16-104. EDMA_TPCC_QEMR

Address Offset	0x0000 0310		
Physical Address	0x4330 0310 0x40D1 0310 0x4151 0310 0x01D1 0310 0x420A 0310 0x421A 0310	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Event Missed Register: The QDMA Event Missed register is set if 2 QDMA events are detected without the first event being cleared or if a Null TR is serviced. If any bit in the EDMA_TPCC_QEMR register is set (and all errors (including EDMA_TPCC_EMR / EDMA_TPCC_CCERR) were previously clear), then an error will be signaled with TPCC error interrupt.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event Missed #7	R	0x0
6	E6	Event Missed #6	R	0x0
5	E5	Event Missed #5	R	0x0
4	E4	Event Missed #4	R	0x0
3	E3	Event Missed #3	R	0x0
2	E2	Event Missed #2	R	0x0
1	E1	Event Missed #1	R	0x0
0	E0	Event Missed #0	R	0x0

Table 16-105. EDMA_TPCC_QEMCR

Address Offset	0x0000 0314		
Physical Address	0x4330 0314 0x40D1 0314 0x4151 0314 0x01D1 0314 0x420A 0314 0x421A 0314	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC

Table 16-105. EDMA_TPCC_QEMCR (continued)

Description	QDMA Event Missed Clear Register: CPU write of '1' to the EDMA_TPCC_QEMCR.En bit causes the EDMA_TPCC_QEMR.En bit to be cleared. CPU write of '0' has no effect. All error bits must be cleared before additional error interrupts will be asserted by CC.
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event Missed Clear #7	W	0x0
6	E6	Event Missed Clear #6	W	0x0
5	E5	Event Missed Clear #5	W	0x0
4	E4	Event Missed Clear #4	W	0x0
3	E3	Event Missed Clear #3	W	0x0
2	E2	Event Missed Clear #2	W	0x0
1	E1	Event Missed Clear #1	W	0x0
0	E0	Event Missed Clear #0	W	0x0

Table 16-106. EDMA_TPCC_CCERR

Address Offset	0x0000 0318		
Physical Address	0x4330 0318 0x40D1 0318 0x4151 0318 0x01D1 0318 0x420A 0318 0x421A 0318	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	CC Error Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															TCERR	RESERVED								QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0
16	TCERR	Transfer Completion Code Error 0x0: Total number of allowed TCCs outstanding has not been reached. 0x1: Total number of allowed TCCs has been reached. TCERR can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors were previously clear), then an error will be signaled with TPCC error interrupt.	R	0x0
15:8	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
7	QTHRXCD7	Queue Threshold Error for Q7 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD7 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
6	QTHRXCD6	Queue Threshold Error for Q6 0x0 : Watermark/threshold has not been exceeded. 0x1 : Watermark/threshold has been exceeded. QTHRXCD6 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
5	QTHRXCD5	Queue Threshold Error for Q5 0x0 : Watermark/threshold has not been exceeded. 0x1 : Watermark/threshold has been exceeded. QTHRXCD5 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
4	QTHRXCD4	Queue Threshold Error for Q4 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD4 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
3	QTHRXCD3	Queue Threshold Error for Q3 0x0: Watermark/threshold has not been exceeded. 0x1 : Watermark/threshold has been exceeded. QTHRXCD3 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
2	QTHRXCD2	Queue Threshold Error for Q2 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD2 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0

Bits	Field Name	Description	Type	Reset
1	QTHRXCD1	Queue Threshold Error for Q1 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD1 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
0	QTHRXCD0	Queue Threshold Error for Q0: 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD0 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/ EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0

Table 16-107. EDMA_TPCC_CCERRCLR

Address Offset	0x0000 031C		
Physical Address	0x4330 031C 0x40D1 031C 0x4151 031C 0x01D1 031C 0x420A 031C 0x421A 031C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	CC Error Clear Register		
Type	W		

31:17	16	15:8	7	6	5	4	3	2	1	0
RESERVED	TCERR	RESERVED	QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0
16	TCERR	Clear Error for EDMA_TPCC_CCERR[16] TR. Write 0x1 to clear the value of EDMA_TPCC_CCERR[16] TCERR. Write 0x0 have no affect.	W	0x0
15:8	RESERVED	Reserved	R	0x0
7	QTHRXCD7	Clear error for EDMA_TPCC_CCERR[7]QTHRXCD7 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT7.WM, QSTAT7.THRXCD, EDMA_TPCC_CCERR[7] QTHRXCD7	W	0x0
6	QTHRXCD6	Clear error for EDMA_TPCC_CCERR[6] QTHRXCD6 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT6.WM, QSTAT6.THRXCD, EDMA_TPCC_CCERR[6]QTHRXCD6	W	0x0

Bits	Field Name	Description	Type	Reset
5	QTHRXCD5	Clear error for EDMA_TPCC_CCERR[5] QTHRXCD5 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT5.WM, QSTAT5.THRXCD, EDMA_TPCC_CCERR[5]QTHRXCD5	W	0x0
4	QTHRXCD4	Clear error for EDMA_TPCC_CCERR[4] QTHRXCD4: Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT4.WM, QSTAT4.THRXCD, EDMA_TPCC_CCERR[4] QTHRXCD4	W	0x0
3	QTHRXCD3	Clear error for EDMA_TPCC_CCERR[3] QTHRXCD3 Write 0x1 to clear the values of QSTAT3.WM, QSTAT3.THRXCD, EDMA_TPCC_CCERR[3] QTHRXCD3 Write 0x0 have no affect.	W	0x0
2	QTHRXCD2	Clear error for EDMA_TPCC_CCERR[2] QTHRXCD2 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT2.WM, QSTAT2.THRXCD, EDMA_TPCC_CCERR[2] QTHRXCD2	W	0x0
1	QTHRXCD1	Clear error for EDMA_TPCC_CCERR[1] QTHRXCD1 Write 0x1 to clear the values of QSTAT1.WM, QSTAT1.THRXCD, EDMA_TPCC_CCERR[1] QTHRXCD1 Write 0x0 have no affect.	W	0x0
0	QTHRXCD0	Clear error for EDMA_TPCC_CCERR[0] QTHRXCD0 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT0.WM, QSTAT0.THRXCD, EDMA_TPCC_CCERR[0] QTHRXCD0	W	0x0

Table 16-108. EDMA_TPCC_EEVAL

Address Offset	0x0000 0320		
Physical Address	0x4330 0320 0x40D1 0320 0x4151 0320 0x01D1 0320 0x420A 0320 0x421A 0320	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Error Eval Register		
Type	W		

31–2	1	0
RESERVED	SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x000000
1	SET	Error Interrupt Set CPU writes 0x0 has no effect. CPU writes 0x1 to the SET bit causes the TPCC error interrupt to be pulsed regardless of state of EDMA_TPCC_EMR/EDMA_TPCC_EMRH, EDMA_TPCC_QEMR, or EDMA_TPCC_CCERR.	W	0x0

Bits	Field Name	Description	Type	Reset
0	EVAL	Error Interrupt Evaluate CPU writes 0x0 has no effect. CPU writes 0x1 to the EVAL bit causes the TPCC error interrupt to be pulsed if any errors have not been cleared in the EDMA_TPCC_EMR/EDMA_TPCC_EMRH, EDMA_TPCC_QEMR, or EDMA_TPCC_CCERR registers. The CPU must also write 0x1 after any error interrupts are serviced in order for subsequent interrupts to be asserted.	W	0x0

Table 16-109. EDMA_TPCC_DRAEM_k

Address Offset	0x0000 0340 + (0x8 * k)		
Physical Address	0x4330 0340 + (0x8 * k) 0x40D1 0340 + (0x8 * k) 0x4151 0340 + (0x8 * k) 0x01D1 0340 + (0x8 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC
Description	DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	DMA Region Access enable for Region M, bit #31	RW	0x0
30	E30	DMA Region Access enable for Region M, bit #30	RW	0x0
29	E29	DMA Region Access enable for Region M, bit #29	RW	0x0
28	E28	DMA Region Access enable for Region M, bit #28	RW	0x0
27	E27	DMA Region Access enable for Region M, bit #27	RW	0x0
26	E26	DMA Region Access enable for Region M, bit #26	RW	0x0
25	E25	DMA Region Access enable for Region M, bit #25	RW	0x0
24	E24	DMA Region Access enable for Region M, bit #24	RW	0x0
23	E23	DMA Region Access enable for Region M, bit #23	RW	0x0
22	E22	DMA Region Access enable for Region M, bit #22	RW	0x0
21	E21	DMA Region Access enable for Region M, bit #21	RW	0x0
20	E20	DMA Region Access enable for Region M, bit #20	RW	0x0
19	E19	DMA Region Access enable for Region M, bit #19	RW	0x0
18	E18	DMA Region Access enable for Region M, bit #18	RW	0x0
17	E17	DMA Region Access enable for Region M, bit #17	RW	0x0
16	E16	DMA Region Access enable for Region M, bit #16	RW	0x0
15	E15	DMA Region Access enable for Region M, bit #15	RW	0x0
14	E14	DMA Region Access enable for Region M, bit #14	RW	0x0
13	E13	DMA Region Access enable for Region M, bit #13	RW	0x0
12	E12	DMA Region Access enable for Region M, bit #12	RW	0x0
11	E11	DMA Region Access enable for Region M, bit #11	RW	0x0
10	E10	DMA Region Access enable for Region M, bit #10	RW	0x0
9	E9	DMA Region Access enable for Region M, bit #9	RW	0x0

Bits	Field Name	Description	Type	Reset
8	E8	DMA Region Access enable for Region M, bit #8	RW	0x0
7	E7	DMA Region Access enable for Region M, bit #7	RW	0x0
6	E6	DMA Region Access enable for Region M, bit #6	RW	0x0
5	E5	DMA Region Access enable for Region M, bit #5	RW	0x0
4	E4	DMA Region Access enable for Region M, bit #4	RW	0x0
3	E3	DMA Region Access enable for Region M, bit #3	RW	0x0
2	E2	DMA Region Access enable for Region M, bit #2	RW	0x0
1	E1	DMA Region Access enable for Region M, bit #1	RW	0x0
0	E0	DMA Region Access enable for Region M, bit #0	RW	0x0

Table 16-110. EDMA_TPCC_DRAEHM_k

Address Offset	0x0000 0344 + (0x8 * k)		
Physical Address	0x4330 0344 + (0x8 * k) 0x40D1 0344 + (0x8 * k) 0x4151 0344 + (0x8 * k) 0x01D1 0344 + (0x8 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC
Description	DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt. En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	DMA Region Access enable for Region M, bit #63	RW	0x0
30	E62	DMA Region Access enable for Region M, bit #62	RW	0x0
29	E61	DMA Region Access enable for Region M, bit #61	RW	0x0
28	E60	DMA Region Access enable for Region M, bit #60	RW	0x0
27	E59	DMA Region Access enable for Region M, bit #59	RW	0x0
26	E58	DMA Region Access enable for Region M, bit #58	RW	0x0
25	E57	DMA Region Access enable for Region M, bit #57	RW	0x0
24	E56	DMA Region Access enable for Region M, bit #56	RW	0x0
23	E55	DMA Region Access enable for Region M, bit #55	RW	0x0
22	E54	DMA Region Access enable for Region M, bit #54	RW	0x0
21	E53	DMA Region Access enable for Region M, bit #53	RW	0x0
20	E52	DMA Region Access enable for Region M, bit #52	RW	0x0
19	E51	DMA Region Access enable for Region M, bit #51	RW	0x0
18	E50	DMA Region Access enable for Region M, bit #50	RW	0x0
17	E49	DMA Region Access enable for Region M, bit #49	RW	0x0
16	E48	DMA Region Access enable for Region M, bit #48	RW	0x0
15	E47	DMA Region Access enable for Region M, bit #47	RW	0x0

Bits	Field Name	Description	Type	Reset
14	E46	DMA Region Access enable for Region M, bit #46	RW	0x0
13	E45	DMA Region Access enable for Region M, bit #45	RW	0x0
12	E44	DMA Region Access enable for Region M, bit #44	RW	0x0
11	E43	DMA Region Access enable for Region M, bit #43	RW	0x0
10	E42	DMA Region Access enable for Region M, bit #42	RW	0x0
9	E41	DMA Region Access enable for Region M, bit #41	RW	0x0
8	E40	DMA Region Access enable for Region M, bit #40	RW	0x0
7	E39	DMA Region Access enable for Region M, bit #39	RW	0x0
6	E38	DMA Region Access enable for Region M, bit #38	RW	0x0
5	E37	DMA Region Access enable for Region M, bit #37	RW	0x0
4	E36	DMA Region Access enable for Region M, bit #36	RW	0x0
3	E35	DMA Region Access enable for Region M, bit #35	RW	0x0
2	E34	DMA Region Access enable for Region M, bit #34	RW	0x0
1	E33	DMA Region Access enable for Region M, bit #33	RW	0x0
0	E32	DMA Region Access enable for Region M, bit #32	RW	0x0

Table 16-111. EDMA_TPCC_QRAEN_k

Address Offset	0x0000 0380 + (0x4 * k)		
Physical Address	0x4330 0380 + (0x4 * k) 0x40D1 0380 + (0x4 * k) 0x4151 0380 + (0x4 * k) 0x01D1 0380 + (0x4 * k) 0x420A 0380 + (0x4 * k) 0x421A 0380 + (0x4 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any QDMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any QDMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region n interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	QDMA Region Access enable for Region M, bit #7	RW	0x0
6	E6	QDMA Region Access enable for Region M, bit #6	RW	0x0
5	E5	QDMA Region Access enable for Region M, bit #5	RW	0x0
4	E4	QDMA Region Access enable for Region M, bit #4	RW	0x0
3	E3	QDMA Region Access enable for Region M, bit #3	RW	0x0
2	E2	QDMA Region Access enable for Region M, bit #2	RW	0x0
1	E1	QDMA Region Access enable for Region M, bit #1	RW	0x0
0	E0	QDMA Region Access enable for Region M, bit #0	RW	0x0

Table 16-112. EDMA_TPCC_Q0E_p

Address Offset	0x0000 0400 + (0x4 * l)		
Physical Address	0x4330 0400 + (0x4 * p) 0x40D1 0400 + (0x4 * p) 0x4151 0400 + (0x4 * p) 0x01D1 0400 + (0x4 * p)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC

Table 16-112. EDMA_TPCC_Q0E_p (continued)

Description	Event Queue Entries Diagram for Queue 0 - Entry 0 through Entry 15
Type	R

31–8	7–6	5–0
RESERVED	ETYPE	ENUM

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:6	ETYPE	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.	R	0x0
5:0	ENUM	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (EDMA_TPCC_ER/EDMA_TPCC_ESR/ EDMA_TPCC_CER), ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (EDMA_TPCC_QER), ENUM will range between 0 and NUM_QDMACH (up to 7).	R	0x0

Table 16-113. EDMA_TPCC_Q1E_p

Address Offset	0x0000 0440 + (0x4 * I)		
Physical Address	0x4330 0440 + (0x4 * p) 0x40D1 0440 + (0x4 * p) 0x4151 0440 + (0x4 * p) 0x01D1 0440 + (0x4 * p)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC
Description	Event Queue Entries Diagram for Queue 1 - Entry 0 through Entry 15		
Type	R		

31–8	7–6	5–0
RESERVED	ETYPE	ENUM

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:6	ETYPE	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.	R	0x0
5:0	ENUM	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (EDMA_TPCC_ER / EDMA_TPCC_ESR / EDMA_TPCC_CER), ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (EDMA_TPCC_QER), ENUM will range between 0 and NUM_QDMACH (up to 7).	R	0x0

Table 16-114. EDMA_TPCC_QSTATN_i

Address Offset	0x0000 0600 + (0x4 * i)		
Physical Address	0x4330 0600 + (0x4 * i) 0x40D1 0600 + (0x4 * i) 0x4151 0600 + (0x4 * i) 0x01D1 0600 + (0x4 * i) 0x420A 0600 + (0x4 * i) 0x421A 0600 + (0x4 * i)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QSTATn Register Set		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	THRXCD	RESERVED	WM	RESERVED	NUMVAL	RESERVED	STRTPTR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R Returns 0's	0x0
24	THRXCD	Threshold Exceeded 0x0 : Threshold specified by QWMTHR(A|B).Qn has not been exceeded. 0x1 : Threshold specified by QWMTHR(A|B).Qn has been exceeded. THRXCD is cleared via EDMA_TPCC_CCERR. WMCLRn bit.	R	0x0
23:21	RESERVED	Reserved	R Returns 0's	0x0
20:16	WM	Watermark for Maximum Queue Usage: Watermark tracks the most entries that have been in QueueN since reset or since the last time that the watermark (WM) was cleared. QSTATn. WM is cleared via EDMA_TPCC_CCERR.WMCLRn bit. Legal values: 0x0: empty 0x10: full	R	0x0
15:13	RESERVED	Reserved	Returns 0's	0x0
12:8	NUMVAL	Number of Valid Entries in QueueN: Represents the total number of entries residing in the Queue Manager FIFO at a given instant. Always enabled. Legal values: = 0x0 (empty) to 0x10 (full) 0x0: empty 0x10: full	R	0x0
7:4	RESERVED	Reserved	Returns 0's	0x0
3:0	STRTPTR	Start Pointer: Represents the offset to the head entry of QueueN, in units of *entries*. Always enabled. Legal values: 0x0: 0th entry 0xF: 15th entry	R	0x0

Table 16-115. EDMA_TPCC_QWMTHRA

Address Offset	0x0000 0620		
Physical Address	0x4330 0620 0x40D1 0620 0x4151 0620 0x01D1 0620 0x420A 0620 0x421A 0620	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Queue Threshold A, for Q[3:0]: EDMA_TPCC_CCERR.QTHRXCDn and QSTATn[24] THRXCD error bit is set when the number of Events in QueueN at an instant in time (visible via QSTATn[12:8] NUMVAL) equals or exceeds the value specified by EDMA_TPCC_QWMTHRA.Qn. Legal values = 0x0 (ever used?) to 0x10 (ever full?) A value of 0x11 disables threshold errors.		

Table 16-115. EDMA_TPCC_QWMTHRA (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			Q3					RESERVED			Q2					RESERVED			Q1					RESERVED			Q0				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:24	Q3	Queue Threshold for Q3 value	RW	0x10
23:21	RESERVED	Reserved	R	0x0
20:16	Q2	Queue Threshold for Q2 value	RW	0x10
15:13	RESERVED	Reserved	R	0x0
12:8	Q1	Queue Threshold for Q1 value	RW	0x10
7:5	RESERVED	Reserved	R	0x0
4:0	Q0	Queue Threshold for Q0 value	RW	0x10

Table 16-116. EDMA_TPCC_QWMTHRB

Address Offset	0x0000 0624		
Physical Address	0x4330 0624 0x40D1 0624 0x4151 0624 0x01D1 0624 0x420A 0624 0x421A 0624	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Queue Threshold B, for Q[7:4]: EDMA_TPCC_CCERR.QTHRXCDn and QSTATn[24]THRXCD error bit is set when the number of Events in QueueN at an instant in time (visible via QSTATn[12:8] NUMVAL) equals or exceeds the value specified by QWMTHRB.Qn. Legal values = 0x0 (ever used?) to 0x10 (ever full?) A value of 0x11 disables threshold errors.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			Q7					RESERVED			Q6					RESERVED			Q5					RESERVED			Q4				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:24	Q7	Queue Threshold for Q7 value (unused in the context of IVAHD)	RW	0x10
23:21	RESERVED	Reserved	R	0x0
20:16	Q6	Queue Threshold for Q6 value (unused in the context of IVAHD)	RW	0x10
15:13	RESERVED	Reserved	R	0x0
12:8	Q5	Queue Threshold for Q5 value (unused in the context of IVAHD)	RW	0x10
7:5	RESERVED	Reserved	R	0x0
4:0	Q4	Queue Threshold for Q4 value (unused in the context of IVAHD)	RW	0x10

Table 16-117. EDMA_TPCC_CCSTAT

Address Offset	0x0000 0640

Table 16-117. EDMA_TPCC_CCSTAT (continued)

Physical Address	0x4330 0640 0x40D1 0640 0x4151 0640 0x01D1 0640 0x420A 0640 0x421A 0640	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	CC Status Register		
Type	R		

31:24	23	22	21	20	19	18	17	16	15:14	13:8	7:5	4	3	2	1	0
RESERVED	QUEACTV7	QUEACTV6	QUEACTV5	QUEACTV4	QUEACTV3	QUEACTV2	QUEACTV1	QUEACTV0	RESERVED	COMPACTV	RESERVED	ACTV	RESERVED	TRACTV	QEVTACTV	EVTACTV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	reads return 0's	R	0x0
23	QUEACTV7	Queue 7 Active 0x0: No Evts are queued in Q7 0x1: At least one TR is queued in Q7.	R	0x0
22	QUEACTV6	Queue 6 Active 0x0: No Evts are queued in Q6. 0x1: At least one TR is queued in Q6.	R	0x0
21	QUEACTV5	Queue 5 Active 0x0: No Evts are queued in Q5 0x1: At least one TR is queued in Q5.	R	0x0
20	QUEACTV4	Queue 4 Active 0x0: No Evts are queued in Q4. 0x1: At least one TR is queued in Q4.	R	0x0
19	QUEACTV3	Queue 3 Active 0x0: No Evts are queued in Q3. 0x1: At least one TR is queued in Q3.	R	0x0
18	QUEACTV2	Queue 2 Active QUEACTV2 = 0 : No Evts are queued in Q2. QUEACTV2 = 1 : At least one TR is queued in Q2. 0x0: 0x1:	R	0x0
17	QUEACTV1	Queue 1 Active 0x0: No Evts are queued in Q1. 0x1: At least one TR is queued in Q1.	R	0x0
16	QUEACTV0	Queue 0 Active 0x0: No Evts are queued in Q0. 0x1: At least one TR is queued in Q0.	R	0x0
15:14	RESERVED	Reserved	R reads return 0's	0x0

Bits	Field Name	Description	Type	Reset
13:8	COMPACTV	Completion Request Active: Counter that tracks the total number of completion requests submitted to the TC. The counter increments when a TR is submitted with TCINTEN or TCCHEN set to '1'. The counter decrements for every valid completion code received from any of the external TCs. The CC will not service new TRs if COMPACTV count is already at the limit. 0x0: No completion requests outstanding. 0x1: Total of '1' completion request outstanding. ... 0x3F: Total of 63 completion requests are outstanding. No additional TRs will be submitted until count is less than 63.	R	0x0
7:5	RESERVED	reads return 0's	R	0x0
4	ACTV	Channel Controller Active Channel Controller Active is a logical-OR of each of the *ACTV signals. The ACTV bit must remain high through the life of a: 0x0: Channel is idle. 0x1: Channel is busy.	R	0x0
3	RESERVED	reads return 0's	R	0x0
2	TRACTV	Transfer Request Active TRACTV = 0 : Transfer Request processing/submission logic is inactive. TRACTV = 1 : Transfer Request processing/submission logic is active. 0x0: 0x1:	R	0x0
1	QEVTACTV	QDMA Event Active 0x0: No enabled QDMA Events are active within the CC. 0x1: At least one enabled DMA Event (EDMA_TPCC_ER, EDMA_TPCC_EER, EDMA_TPCC_ESR, EDMA_TPCC_CER) is active within the CC.	R	0x0
0	EVTACTV	DMA Event Active 0x0: No enabled DMA Events are active within the CC. 0x1: At least one enabled DMA Event (EDMA_TPCC_ER, EDMA_TPCC_EER, EDMA_TPCC_ESR, EDMA_TPCC_CER) is active within the CC.	R	0x0

Table 16-118. EDMA_TPCC_AETCTL

Address Offset	0x0000 0700		
Physical Address	0x4330 0700 0x40D1 0700 0x4151 0700 0x01D1 0700 0x420A 0700 0x421A 0700	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Advanced Event Trigger Control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EN	RESERVED	ENDINT	RESERVED	TYPE	STRTEVT

Bits	Field Name	Description	Type	Reset
31	EN	AET Enable 0x0: AET event generation is disabled. 0x1: AET event generation is enabled.	RW	0x0
30:14	RESERVED	Reserved	R	0x0
13:8	ENDINT	AET End Interrupt: Dictates the completion interrupt number that will force the tpcc_aet signal to be deasserted (low)	RW	0x0
7	RESERVED	Reserved	R	0x0
6	TYPE	AET Event Type 0x0: Event specified by STARTEVT applies to DMA Events (set by EDMA_TPCC_ER, EDMA_TPCC_ESR, or EDMA_TPCC_CER) 0x1: Event specified by STARTEVT applies to QDMA Events	RW	0x0
5:0	STRTEVT	AET Start Event: Dictates the Event Number that will force the tpcc_aet signal to be asserted (high)	RW	0x0

Table 16-119. EDMA_TPCC_AETSTAT

Address Offset	0x0000 0704		
Physical Address	0x4330 0704 0x40D1 0704 0x4151 0704 0x01D1 0704 0x420A 0704 0x421A 0704	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Advanced Event Trigger Stat		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STAT

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R Return 0's	0x0
0	STAT	AET Status 0x0: tpcc_aet is currently low. 0x1: tpcc_aet is currently high.	R	0x0

Table 16-120. EDMA_TPCC_AETCMD

Address Offset	0x0000 0708		
Physical Address	0x4330 0708 0x40D1 0708 0x4151 0708 0x01D1 0708 0x420A 0708 0x421A 0708	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	AET Command		
Type	W		

31–1	0
RESERVED	CLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	CLR	AET Clear command CPU writes 0x0 has no effect. CPU writes 0x1 to the CLR bit causes the tpcc_aet output signal and EDMA_TPCC_AETSTAT[0]STAT register to be cleared.	W	0x0

Table 16-121. EDMA_TPCC_MPFAR

Address Offset	0x0000 0800		
Physical Address	0x4330 0800 0x40D1 0800 0x4151 0800 0x01D1 0800 0x420A 0800 0x421A 0800	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	MMemory Protection Fault Address		
Type	R		

31–0
FADDR

Bits	Field Name	Description	Type	Reset
31:0	FADDR	Fault Address: 32-bit read-only status register containing the faulting address when a mMemory protection violation is detected. This register can only be cleared via the EDMA_TPCC_MPFCR.	R	0x0

Table 16-122. EDMA_TPCC_MPFSR

Address Offset	0x0000 0804		
Physical Address	0x4330 0804 0x40D1 0804 0x4151 0804 0x01D1 0804 0x420A 0804 0x421A 0804	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Memory Protection Fault Status Register		
Type	R		

31–13	12–9	8–6	5	4	3	2	1	0
RESERVED	FID	RESERVED	SRE	SWE	SXE	URE	UWE	UXE

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R Returns 0	0x0
12:9	FID	Faulted ID: FID register contains valid info if any of the MP error bits (UXE, UWE, URE, SXE, SWE, SRE) are non-zero (i.e., if an error has been detected.) The FID field contains the VBus PrivID for the specific request/requestor that resulted in a MP Error.	R	0x0

Bits	Field Name	Description	Type	Reset
8:6	RESERVED	Reserved	R Returns 0	0x0
5	SRE	Supervisor Read Error 0x0: No error detected. 0x1: Supervisor level task attempted to Read from a MP Page without SR permissions.	R	0x0
4	SWE	Supervisor Write Error 0x0: No error detected. 0x1: Supervisor level task attempted to Write to a MP Page without SW permissions.	R	0x0
3	SXE	Supervisor Execute Error 0x0: No error detected. 0x1: Supervisor level task attempted to Execute from a MP Page without SX permissions.	R	0x0
2	URE	User Read Error 0x0: No error detected. 0x1: User level task attempted to Read from a MP Page without UR permissions.	R	0x0
1	UWE	User Write Error 0x0: No error detected. 0x1: User level task attempted to Write to a MP Page without UW permissions.	R	0x0
0	UXE	User Execute Error 0x0: No error detected 0x1: User level task attempted to Execute from a MP Page without UX permissions.	R	0x0

Table 16-123. EDMA_TPCC_MPFCR

Address Offset	0x0000 0808		
Physical Address	0x4330 0808 0x40D1 0808 0x4151 0808 0x01D1 0808 0x420A 0808 0x421A 0808	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Memory Protection Fault Command Register		
Type	W		

31:1	0
RESERVED	MPFCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MPFCLR	Fault Clear register CPU writes 0x0: has no effect CPU writes 0x1: to the MPFCLR bit causes any error conditions stored in EDMA_TPCC_MPFAR and EDMA_TPCC_MPFSR registers to be cleared.	W	0x0

Table 16-124. EDMA_TPCC_MPPAG

Address Offset	0x0000 080C		
Physical Address	0x4330 080C 0x40D1 080C 0x4151 080C 0x01D1 080C 0x420A 080C 0x421A 080C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Memory Protection Page Attribute for Global registers		
Type	RW		

31:16	15	14	13	12	11	10	9	8:7	6	5	4	3	2	1	0
RESERVED	AID5	AID4	AID3	AID2	AID1	AID0	EXT	RESERVED	SR	SW	SX	UR	UW	UX	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15	AID5	Allowed ID 5 0x0: VBus requests with PrivID == '5' are not allowed regardless of permission settings (UW, UR, SW, SR).0 0x1: VBus requests with PrivID == '5' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
14	AID4	Allowed ID 4 0x0: VBus requests with PrivID == '4' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '4' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
13	AID3	Allowed ID 3 0x0: VBus requests with PrivID == '3' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '3' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
12	AID2	Allowed ID 2 0x0: VBus requests with PrivID == '2' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '2' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
11	AID1	Allowed ID 1 0x0: VBus requests with PrivID == '1' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '1' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
10	AID0	Allowed ID 0 0x0: VBus requests with PrivID == '0' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '0' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1

Bits	Field Name	Description	Type	Reset
9	EXT	External Allowed ID 0x0: VBus requests with PrivID = '6' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID = '6' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
8:6	RESERVED	Reserved	R	0x1
5	SR	Supervisor Read permission 0x0: Supervisor read accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
4	SW	Supervisor Write permission 0x0: Supervisor write accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
3	SX	Supervisor Execute permission 0x0: Supervisor execute accesses are not allowed 0x1: Supervisor execute accesses are allowed	RW	0x0
2	UR	User Read permission 0x0: User read accesses are not allowed 0x1: User write accesses are allowed	RW	0x1
1	UW	User Write permission 0x0: User write accesses are not allowed 0x1: User write accesses are allowed	RW	0x1
0	UX	User Execute permission 0x0: User execute accesses are not allowed 0x1: User execute accesses are allowed	RW	0x0

Table 16-125. EDMA_TPCC_MPPAN_k

Address Offset	0x0000 0810 + (0x4 * k)		
Physical Address	0x4330 0810 + (0x4 * k) 0x40D1 0810 + (0x4 * k) 0x4151 0810 + (0x4 * k) 0x01D1 0810 + (0x4 * k) 0x420A 0810 + (0x4 * k) 0x421A 0810 + (0x4 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	P Permission Attribute for DMA Region n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AID5	AID4	AID3	AID2	AID1	AID0	EXT	RESERVED			SR	SW	SX	UR	UW	UX

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15	AID5	Allowed ID 5 0x0: VBus requests with PrivID == '5' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '5' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1

Bits	Field Name	Description	Type	Reset
14	AID4	Allowed ID 4 0x0: VBus requests with PrivID == '4' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '4' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
13	AID3	Allowed ID 3 0x0: VBus requests with PrivID == '3' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '3' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
12	AID2	Allowed ID 2 0x0: VBus requests with PrivID == '2' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '2' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
11	AID1	Allowed ID 1 0x0: VBus requests with PrivID == '1' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '1' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
10	AID0	Allowed ID 0: AID0 = 0 : VBus requests with PrivID == '0' are not allowed regardless of permission settings (UW, UR, SW, SR). AID0 = 1 : VBus requests with PrivID == '0' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR). 0x0: 0x1:	RW	0x1
9	EXT	External Allowed ID 0x0: VBus requests with PrivID = '6' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID = '6' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
8:6	RESERVED	Reserved	R	0x0
5	SR	Supervisor Read permission 0x0: Supervisor read accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
4	SW	Supervisor Write permission 0x0: Supervisor write accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
3	SX	Supervisor Execute permission 0x0: Supervisor execute accesses are not allowed 0x1: Supervisor execute accesses are allowed	RW	0x0
2	UR	User Read permission 0x0: User read accesses are not allowed 0x1: User write accesses are allowed	RW	0x1

Bits	Field Name	Description	Type	Reset
1	UW	User Write permission 0x0: User write accesses are not allowed 0x1: User write accesses are allowed	RW	0x1
0	UX	User Execute permission 0x0: User execute accesses are not allowed 0x0: User execute accesses are allowed	RW	0x0

Table 16-126. EDMA_TPCC_ER

Address Offset	0x0000 1000		
Physical Address	0x4330 1000 0x40D1 1000 0x4151 1000 0x01D1 1000 0x420A 1000 0x421A 1000	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Register: If EDMA_TPCC_ER.En bit is set and the EDMA_TPCC_EER.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ER.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EDMA_TPCC_EER.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EER register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECR pseudo-register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0

Bits	Field Name	Description	Type	Reset
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-127. EDMA_TPCC_ERH

Address Offset	0x0000 1004		
Physical Address	0x4330 1004 0x40D1 1004 0x4151 1004 0x01D1 1004 0x420A 1004 0x421A 1004	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Register (High Part): If EDMA_TPCC_ERH.En bit is set and the EDMA_TPCC_EERH.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ERH.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EDMA_TPCC_EERH.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EERH register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECRH pseudo-register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0

Bits	Field Name	Description	Type	Reset
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 16-128. EDMA_TPCC_ECR

Address Offset	0x0000 1008		
Physical Address	0x4330 1008 0x40D1 1008 0x4151 1008 0x01D1 1008 0x420A 1008 0x421A 1008	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Clear Register: CPU write of '1' to the EDMA_TPCC_ECR.En bit causes the EDMA_TPCC_ER.En bit to be cleared. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0

Bits	Field Name	Description	Type	Reset
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-129. EDMA_TPCC_ECRH

Address Offset	0x0000 100C		
Physical Address	0x4330 100C 0x40D1 100C 0x4151 100C 0x01D1 100C 0x420A 100C 0x421A 100C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Clear Register (High Part): CPU write of '1' to the EDMA_TPCC_ECRH.En bit causes the EDMA_TPCC_ERH.En bit to be cleared. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0

Bits	Field Name	Description	Type	Reset
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-130. EDMA_TPCC_ESR

Address Offset	0x0000 1010		
Physical Address	0x4330 1010 0x40D1 1010 0x4151 1010 0x01D1 1010 0x420A 1010 0x421A 1010	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Set Register: CPU write of '1' to the EDMA_TPCC_ESR.En bit causes the EDMA_TPCC_ER.En bit to be set. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0

Bits	Field Name	Description	Type	Reset
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-131. EDMA_TPCC_ESRH

Address Offset	0x0000 1014		
Physical Address	0x4330 1014 0x40D1 1014 0x4151 1014 0x01D1 1014 0x420A 1014 0x421A 1014	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Set Register (High Part) CPU write of '1' to the EDMA_TPCC_ESRH.En bit causes the EDMA_TPCC_ERH.En bit to be set. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0

Bits	Field Name	Description	Type	Reset
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-132. EDMA_TPCC_CER

Address Offset	0x0000 1018		
Physical Address	0x4330 1018 0x40D1 1018 0x4151 1018 0x01D1 1018 0x420A 1018 0x421A 1018	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Chained Event Register: If EDMA_TPCC_CER.En bit is set (regardless of state of EDMA_TPCC_EER.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CER.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CER.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0

Bits	Field Name	Description	Type	Reset
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-133. EDMA_TPCC_CERH

Address Offset	0x0000 101C		
Physical Address	0x4330 101C 0x40D1 101C 0x4151 101C 0x01D1 101C 0x420A 101C 0x421A 101C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Chained Event Register (High Part): If EDMA_TPCC_CERH.En bit is set (regardless of state of EDMA_TPCC_EERH.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CERH.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CERH.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 16-134. EDMA_TPCC_EER

Address Offset	0x0000 1020		
Physical Address	0x4330 1020 0x40D1 1020 0x4151 1020 0x01D1 1020 0x420A 1020 0x421A 1020	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Register: Enables DMA transfers for EDMA_TPCC_ER.En pending events. EDMA_TPCC_ER.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CER) or Event Set Register (EDMA_TPCC_ESR). Note that if a bit is set in EDMA_TPCC_ER.En while EDMA_TPCC_EER.En is disabled, no action is taken. If EDMA_TPCC_EER.En is enabled at a later point (and EDMA_TPCC_ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EER.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESR and can be disabled via writes to EDMA_TPCC_EECR register. EDMA_TPCC_EER.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EER.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.		

Table 16-134. EDMA_TPCC_EER (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-135. EDMA_TPCC_EERH

Address Offset	0x0000 1024		
Physical Address	0x4330 1024 0x40D1 1024 0x4151 1024 0x01D1 1024 0x420A 1024 0x421A 1024	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC

Table 16-135. EDMA_TPCC_EERH (continued)

Description	Event Enable Register (High Part): Enables DMA transfers for EDMA_TPCC_ERH.En pending events. EDMA_TPCC_ERH.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CERH) or Event Set Register (EDMA_TPCC_ESRH). Note that if a bit is set in EDMA_TPCC_ERH.En while EDMA_TPCC_EERH.En is disabled, no action is taken. If EDMA_TPCC_EERH.En is enabled at a later point (and EDMA_TPCC_ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EERH.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESRH and can be disabled via writes to EDMA_TPCC_EECRH register. EDMA_TPCC_EERH.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EERH.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0

Bits	Field Name	Description	Type	Reset
0	E32	Event #32	R	0x0

Table 16-136. EDMA_TPCC_EECR

Address Offset	0x0000 1028		
Physical Address	0x4330 1028 0x40D1 1028 0x4151 1028 0x01D1 1028 0x420A 1028 0x421A 1028	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Clear Register CPU writes of '1' to the EDMA_TPCC_EECR.En bit causes the EDMA_TPCC_EER.En bit to be cleared. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0

Bits	Field Name	Description	Type	Reset
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-137. EDMA_TPCC_EECRH

Address Offset	0x0000 102C		
Physical Address	0x4330 102C 0x40D1 102C 0x4151 102C 0x01D1 102C 0x420A 102C 0x421A 102C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Clear Register (High Part) CPU writes of '1' to the EDMA_TPCC_EECRH.En bit causes the EERH.En bit to be cleared. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0

Bits	Field Name	Description	Type	Reset
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-138. EDMA_TPCC_EESR

Address Offset	0x0000 1030		
Physical Address	0x4330 1030 0x40D1 1030 0x4151 1030 0x01D1 1030 0x420A 1030 0x421A 1030	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Set Register CPU write of '1' to the EDMA_TPCC_EESR.En bit causes the EDMA_TPCC_EER.En bit to be set. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0

Bits	Field Name	Description	Type	Reset
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-139. EDMA_TPCC_EESRH

Address Offset	0x0000 1034		
Physical Address	0x4330 1034 0x40D1 1034 0x4151 1034 0x01D1 1034 0x420A 1034 0x421A 1034	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Set Register (High Part) CPU writes of '1' to the EDMA_TPCC_EESRH.En bit causes the EDMA_TPCC_EERH.En bit to be set. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0

Bits	Field Name	Description	Type	Reset
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-140. EDMA_TPCC_SER

Address Offset	0x0000 1038		
Physical Address	0x4330 1038 0x40D1 1038 0x4151 1038 0x01D1 1038 0x420A 1038 0x421A 1038	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Secondary Event Register The secondary event register is used along with the Event Register (EDMA_TPCC_ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0

Bits	Field Name	Description	Type	Reset
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-141. EDMA_TPCC_SERH

Address Offset	0x0000 103C		
Physical Address	0x4330 103C 0x40D1 103C 0x4151 103C 0x01D1 103C 0x420A 103C 0x421A 103C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Secondary Event Register (High Part) The secondary event register is used along with the Event Register (EDMA_TPCC_ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0

Bits	Field Name	Description	Type	Reset
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 16-142. EDMA_TPCC_SECR

Address Offset	0x0000 1040		
Physical Address	0x4330 1040 0x40D1 1040 0x4151 1040 0x01D1 1040 0x420A 1040 0x421A 1040	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Secondary Event Clear Register The secondary event clear register is used to clear the status of the EDMA_TPCC_SER registers. CPU write of '1' to the EDMA_TPCC_SECR.En bit clears the EDMA_TPCC_SER register. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0

Bits	Field Name	Description	Type	Reset
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-143. EDMA_TPCC_SECRH

Address Offset	0x0000 1044		
Physical Address	0x4330 1044 0x40D1 1044 0x4151 1044 0x01D1 1044 0x420A 1044 0x421A 1044	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Secondary Event Clear Register (High Part) The secondary event clear register is used to clear the status of the EDMA_TPCC_SERH registers. CPU write of '1' to the EDMA_TPCC_SECRH.En bit clears the EDMA_TPCC_SERH register. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0

Bits	Field Name	Description	Type	Reset
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-144. EDMA_TPCC_IER

Address Offset	0x0000 1050		
Physical Address	0x4330 1050 0x40D1 1050 0x4151 1050 0x01D1 1050 0x420A 1050 0x421A 1050	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Register EDMA_TPCC_IER.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESR and can be disabled via writes to EDMA_TPCC_IECR register. EDMA_TPCC_IER. In = 0: EDMA_TPCC_IPR.In is NOT enabled for interrupts. EDMA_TPCC_IER. In = 1: EDMA_TPCC_IPR.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0

Bits	Field Name	Description	Type	Reset
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0
7	I7	Interrupt associated with TCC #7	R	0x0
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0
0	I0	Interrupt associated with TCC #0	R	0x0

Table 16-145. EDMA_TPCC_IERH

Address Offset	0x0000 1054		
Physical Address	0x4330 1054 0x40D1 1054 0x4151 1054 0x01D1 1054 0x420A 1054 0x421A 1054	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Register (High Part) EDMA_TPCC_IERH.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESRH and can be disabled via writes to EDMA_TPCC_IECRH register. EDMA_TPCC_IERH. In = 0: EDMA_TPCC_IPRH.In is NOT enabled for interrupts. EDMA_TPCC_IERH. In = 1: EDMA_TPCC_IPRH.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0

Bits	Field Name	Description	Type	Reset
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 16-146. EDMA_TPCC_IECR

Address Offset	0x0000 1058		
Physical Address	0x4330 1058 0x40D1 1058 0x4151 1058 0x01D1 1058 0x420A 1058 0x421A 1058	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Clear Register CPU writes of '1' to the EDMA_TPCC_IECR.In bit causes the EDMA_TPCC_IER.In bit to be cleared. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0

Bits	Field Name	Description	Type	Reset
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 16-147. EDMA_TPCC_IECRH

Address Offset	0x0000 105C		
Physical Address	0x4330 105C 0x40D1 105C 0x4151 105C 0x01D1 105C 0x420A 105C 0x421A 105C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Clear Register (High Part) CPU write of '1' to the EDMA_TPCC_IECRH.In bit causes the EDMA_TPCC_IERH.In bit to be cleared. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0

Bits	Field Name	Description	Type	Reset
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 16-148. EDMA_TPCC_IESR

Address Offset	0x0000 1060		
Physical Address	0x4330 1060 0x40D1 1060 0x4151 1060 0x01D1 1060 0x420A 1060 0x421A 1060	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Set Register CPU write of '1' to the EDMA_TPCC_IESR.In bit causes the EDMA_TPCC_IESR.In bit to be set. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0

Bits	Field Name	Description	Type	Reset
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 16-149. EDMA_TPCC_IESRH

Address Offset	0x0000 1064		
Physical Address	0x4330 1064 0x40D1 1064 0x4151 1064 0x01D1 1064 0x420A 1064 0x421A 1064	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Set Register (High Part) CPU write of '1' to the EDMA_TPCC_IESRH.In bit causes the EDMA_TPCC_IESRH.In bit to be set. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0

Bits	Field Name	Description	Type	Reset
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 16-150. EDMA_TPCC_IPR

Address Offset	0x0000 1068		
Physical Address	0x4330 1068 0x40D1 1068 0x4151 1068 0x01D1 1068 0x420A 1068 0x421A 1068	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Pending Register EDMA_TPCC_IPR.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPR.In bit is cleared via software by writing a '1' to EDMA_TPCC_ICR.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0

Bits	Field Name	Description	Type	Reset
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0
7	I7	Interrupt associated with TCC #7	R	0x0
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0
0	I0	Interrupt associated with TCC #0	R	0x0

Table 16-151. EDMA_TPCC_IPRH

Address Offset	0x0000 106C		
Physical Address	0x4330 106C 0x40D1 106C 0x4151 106C 0x01D1 106C 0x420A 106C 0x421A 106C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Pending Register (High Part) EDMA_TPCC_IPRH.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPRH. In bit is cleared via software by writing a '1' to EDMA_TPCC_ICRH.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0

Bits	Field Name	Description	Type	Reset
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 16-152. EDMA_TPCC_ICR

Address Offset	0x0000 1070		
Physical Address	0x4330 1070 0x40D1 1070 0x4151 1070 0x01D1 1070 0x420A 1070 0x421A 1070	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Clear Register CPU write of '1' to the EDMA_TPCC_ICR.In bit causes the EDMA_TPCC_IPR.In bit to be cleared. CPU write of '0' has no effect. All EDMA_TPCC_IPR.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0

Bits	Field Name	Description	Type	Reset
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 16-153. EDMA_TPCC_ICRH

Address Offset	0x0000 1074		
Physical Address	0x4330 1074 0x40D1 1074 0x4151 1074 0x01D1 1074 0x420A 1074 0x421A 1074	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Clear Register (High Part) CPU write of '1' to the EDMA_TPCC_ICRH.In bit causes the EDMA_TPCC_IPRH.In bit to be cleared. CPU write of '0' has no effect. All EDMA_TPCC_IPRH.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0

Bits	Field Name	Description	Type	Reset
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 16-154. EDMA_TPCC_IEVAL

Address Offset	0x0000 1078		
Physical Address	0x4330 1078 0x40D1 1078 0x4151 1078 0x01D1 1078 0x420A 1078 0x421A 1078	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Eval Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
1	SET	Interrupt Set: CPU write of '1' to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (EDMA_TPCC_IPRn). CPU write of '0' has no effect.	W	0x0
0	EVAL	Interrupt Evaluate: CPU write of '1' to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (EDMA_TPCC_IPRn). CPU write of '0' has no effect.	W	0x0

Table 16-155. EDMA_TPCC_QER

Address Offset	0x0000 1080		
Physical Address	0x4330 1080 0x40D1 1080 0x4151 1080 0x01D1 1080 0x420A 1080 0x421A 1080	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Event Register: If EDMA_TPCC_QER.En bit is set, then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. EDMA_TPCC_QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. EDMA_TPCC_QER.En bit is cleared when the corresponding event is prioritized and serviced. EDMA_TPCC_QER.En is also cleared when user writes a '1' to the EDMA_TPCC_QSECR.En bit. If the EDMA_TPCC_QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and EDMA_TPCC_QEER register is set, then the corresponding bit in the QDMA Event Missed Register is set.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-156. EDMA_TPCC_QEER

Address Offset	0x0000 1084		
Physical Address	0x4330 1084 0x40D1 1084 0x4151 1084 0x01D1 1084 0x420A 1084 0x421A 1084	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC

Table 16-156. EDMA_TPCC_QEER (continued)

Description	QDMA Event Enable Register Enabled/disabled QDMA address comparator for QDMA Channel N. EDMA_TPCC_QEER.En is not directly writeable. QDMA channels can be enabled via writes to EDMA_TPCC_QEESR and can be disabled via writes to EDMA_TPCC_QEECR register. EDMA_TPCC_QEER.En = 1, The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in EDMA_TPCC_QER.En. EDMA_TPCC_QEER.En = 0, The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in EDMA_TPCC_QER.En.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-157. EDMA_TPCC_QEECR

Address Offset	0x0000 1088		
Physical Address	0x4330 1088 0x40D1 1088 0x4151 1088 0x01D1 1088 0x420A 1088 0x421A 1088	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Event Enable Clear Register CPU write of '1' to the EDMA_TPCC_QEECR.En bit causes the EDMA_TPCC_QEER.En bit to be cleared. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-158. EDMA_TPCC_QEESR

Address Offset	0x0000 108C		
Physical Address	0x4330 108C 0x40D1 108C 0x4151 108C 0x01D1 108C 0x420A 108C 0x421A 108C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Event Enable Set Register CPU write of '1' to the EDMA_TPCC_QEESR.En bit causes the EDMA_TPCC_QEESR.En bit to be set. CPU write of '0' has no effect..		
Type	W		

31:8	7	6	5	4	3	2	1	0
RESERVED	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-159. EDMA_TPCC_QSER

Address Offset	0x0000 1090		
Physical Address	0x4330 1090 0x40D1 1090 0x4151 1090 0x01D1 1090 0x420A 1090 0x421A 1090	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Secondary Event Register The QDMA secondary event register is used along with the QDMA Event Register (EDMA_TPCC_QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31:8	7	6	5	4	3	2	1	0
RESERVED	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0

Bits	Field Name	Description	Type	Reset
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-160. EDMA_TPCC_QSECR

Address Offset	0x0000 1094		
Physical Address	0x4330 1094 0x40D1 1094 0x4151 1094 0x01D1 1094 0x420A 1094 0x421A 1094	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Secondary Event Clear Register The secondary event clear register is used to clear the status of the EDMA_TPCC_QSER and EDMA_TPCC_QER register (note that this is slightly different than the EDMA_TPCC_SER operation, which does not clear the EDMA_TPCC_ER.En register). CPU write of '1' to the EDMA_TPCC_QSECR.En bit clears the EDMA_TPCC_QSER.En and EDMA_TPCC_QER.En register fields. CPU write of '0' has no effect..		
Type	W		

31–8	7	6	5	4	3	2	1	0
RESERVED	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-161. EDMA_TPCC_ER_RN_k

Address Offset	0x0000 2000 + (0x200 * k)		
Physical Address	0x4330 2000 + (0x200 * k) 0x40D1 2000 + (0x200 * k) 0x4151 2000 + (0x200 * k) 0x01D1 2000 + (0x200 * k) 0x420A 2000 + (0x200 * k) 0x421A 2000 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Register If EDMA_TPCC_ER.En bit is set and the EDMA_TPCC_EER.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ER.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EDMA_TPCC_EER.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EER register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECR pseudo-register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-162. EDMA_TPCC_ERH_RN_k

Address Offset	0x0000 2004 + (0x200 * k)		
Physical Address	0x4330 2004 + (0x200 * k) 0x40D1 2004 + (0x200 * k) 0x4151 2004 + (0x200 * k) 0x01D1 2004 + (0x200 * k) 0x420A 2004 + (0x200 * k) 0x421A 2004 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC

Table 16-162. EDMA_TPCC_ERH_RN_k (continued)

Description	Event Register (High Part) If EDMA_TPCC_ERH.En bit is set and the EDMA_TPCC_EERH.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ERH.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EERH.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EERH register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECRH pseudo-register.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 16-163. EDMA_TPCC_ECR_RN_k

Address Offset	0x0000 2008 + (0x200 * k)		
Physical Address	0x4330 2008 + (0x200 * k) 0x40D1 2008 + (0x200 * k) 0x4151 2008 + (0x200 * k) 0x01D1 2008 + (0x200 * k) 0x420A 2008 + (0x200 * k) 0x421A 2008 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Clear Register CPU write of '1' to the EDMA_TPCC_ECR.En bit causes the EDMA_TPCC_ER.En bit to be cleared. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-164. EDMA_TPCC_ECRH_RN_k

Address Offset	0x0000 200C + (0x200 * k)		
Physical Address	0x4330 200C + (0x200 * k) 0x40D1 200C + (0x200 * k) 0x4151 200C + (0x200 * k) 0x01D1 200C + (0x200 * k) 0x420A 200C + (0x200 * k) 0x421A 200C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Clear Register (High Part) CPU write of '1' to the EDMA_TPCC_ECRH.En bit causes the EDMA_TPCC_ERH.En bit to be cleared. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-165. EDMA_TPCC_ESR_RN_k

Address Offset	0x0000 2010 + (0x200 * k)		
Physical Address	0x4330 2010 + (0x200 * k) 0x40D1 2010 + (0x200 * k) 0x4151 2010 + (0x200 * k) 0x01D1 2010 + (0x200 * k) 0x420A 2010 + (0x200 * k) 0x421A 2010 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Set Register CPU write of '1' to the EDMA_TPCC_ESR.En bit causes the EDMA_TPCC_ER.En bit to be set. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-166. EDMA_TPCC_ESRH_RN_k

Address Offset	0x0000 2014 + (0x200 * k)		
Physical Address	0x4330 2014 + (0x200 * k) 0x40D1 2014 + (0x200 * k) 0x4151 2014 + (0x200 * k) 0x01D1 2014 + (0x200 * k) 0x420A 2014 + (0x200 * k) 0x421A 2014 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Set Register (High Part) CPU write of '1' to the EDMA_TPCC_ESRH.En bit causes the EDMA_TPCC_ERH.En bit to be set. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-167. EDMA_TPCC_CER_RN_k

Address Offset	0x0000 2018 + (0x200 * k)		
Physical Address	0x4330 2018 + (0x200 * k) 0x40D1 2018 + (0x200 * k) 0x4151 2018 + (0x200 * k) 0x01D1 2018 + (0x200 * k) 0x420A 2018 + (0x200 * k) 0x421A 2018 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Chained Event Register If EDMA_TPCC_CER.En bit is set (regardless of state of EDMA_TPCC_EER.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CER.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CER.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0

Bits	Field Name	Description	Type	Reset
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-168. EDMA_TPCC_CERH_RN_k

Address Offset	0x0000 201C + (0x200 * k)		
Physical Address	0x4330 201C + (0x200 * k) 0x40D1 201C + (0x200 * k) 0x4151 201C + (0x200 * k) 0x01D1 201C + (0x200 * k) 0x420A 201C + (0x200 * k) 0x421A 201C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Chained Event Register (High Part) If EDMA_TPCC_CERH.En bit is set (regardless of state of EDMA_TPCC_EERH.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CERH.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CERH.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0

Bits	Field Name	Description	Type	Reset
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 16-169. EDMA_TPCC_EER_RN_k

Address Offset	0x0000 2020 + (0x200 * k)		
Physical Address	0x4330 2020 + (0x200 * k) 0x40D1 2020 + (0x200 * k) 0x4151 2020 + (0x200 * k) 0x01D1 2020 + (0x200 * k) 0x420A 2020 + (0x200 * k) 0x421A 2020 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Register Enables DMA transfers for EDMA_TPCC_ER.En pending events. EDMA_TPCC_ER.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CER) or Event Set Register (EDMA_TPCC_ESR). **NOTE: If a bit is set in EDMA_TPCC_ER.En while EDMA_TPCC_EER.En is disabled, no action is taken.** If EDMA_TPCC_EER.En is enabled at a later point (and EDMA_TPCC_ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EER.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESR and can be disabled via writes to EDMA_TPCC_EECR register. EDMA_TPCC_EER.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EER.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0

Bits	Field Name	Description	Type	Reset
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-170. EDMA_TPCC_EERH_RN_k

Address Offset	0x0000 2024 + (0x200 * k)		
Physical Address	0x4330 2024 + (0x200 * k) 0x40D1 2024 + (0x200 * k) 0x4151 2024 + (0x200 * k) 0x01D1 2024 + (0x200 * k) 0x420A 2024 + (0x200 * k) 0x421A 2024 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Register (High Part) Enables DMA transfers for EDMA_TPCC_ERH.En pending events. EDMA_TPCC_ERH.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CERH) or Event Set Register (EDMA_TPCC_ESRH). **NOTE: If a bit is set in EDMA_TPCC_ERH.En while EDMA_TPCC_EERH.En is disabled, no action is taken.** If EDMA_TPCC_EERH.En is enabled at a later point (and EDMA_TPCC_ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EERH.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESRH and can be disabled via writes to EDMA_TPCC_EECRH register. EDMA_TPCC_EERH.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EERH.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0

Bits	Field Name	Description	Type	Reset
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 16-171. EDMA_TPCC_EECR_RN_k

Address Offset	0x0000 2028 + (0x200 * k)		
Physical Address	0x4330 2028 + (0x200 * k) 0x40D1 2028 + (0x200 * k) 0x4151 2028 + (0x200 * k) 0x01D1 2028 + (0x200 * k) 0x420A 2028 + (0x200 * k) 0x421A 2028 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Clear Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EECR.En bit causes the EDMA_TPCC_EER.En bit to be cleared.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0

Bits	Field Name	Description	Type	Reset
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-172. EDMA_TPCC_EECRH_RN_k

Address Offset	0x0000 202C + (0x200 * k)		
Physical Address	0x4330 202C + (0x200 * k) 0x40D1 202C + (0x200 * k) 0x4151 202C + (0x200 * k) 0x01D1 202C + (0x200 * k) 0x420A 202C + (0x200 * k) 0x421A 202C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Clear Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EECRH.En bit causes the EDMA_TPCC_EERH.En bit to be cleared.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0

Bits	Field Name	Description	Type	Reset
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-173. EDMA_TPCC_EESR_RN_k

Address Offset	0x0000 2030 + (0x200 * k)		
Physical Address	0x4330 2030 + (0x200 * k) 0x40D1 2030 + (0x200 * k) 0x4151 2030 + (0x200 * k) 0x01D1 2030 + (0x200 * k) 0x420A 2030 + (0x200 * k) 0x421A 2030 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Set Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EESR.En bit causes the EDMA_TPCC_EER.En bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0

Bits	Field Name	Description	Type	Reset
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-174. EDMA_TPCC_EESRH_RN_k

Address Offset	0x0000 2034 + (0x200 * k)		
Physical Address	0x4330 2034 + (0x200 * k) 0x40D1 2034 + (0x200 * k) 0x4151 2034 + (0x200 * k) 0x01D1 2034 + (0x200 * k) 0x420A 2034 + (0x200 * k) 0x421A 2034 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Event Enable Set Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EESRH.En bit causes the EDMA_TPCC_EERH.En bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0

Bits	Field Name	Description	Type	Reset
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-175. EDMA_TPCC_SER_RN_k

Address Offset	0x0000 2038 + (0x200 * k)		
Physical Address	0x4330 2038 + (0x200 * k) 0x40D1 2038 + (0x200 * k) 0x4151 2038 + (0x200 * k) 0x01D1 2038 + (0x200 * k) 0x420A 2038 + (0x200 * k) 0x421A 2038 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Secondary Event Register The secondary event register is used along with the Event Register (EDMA_TPCC_ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-176. EDMA_TPCC_SERH_RN_k

Address Offset	0x0000 203C + (0x200 * k)		
Physical Address	0x4330 203C + (0x200 * k) 0x40D1 203C + (0x200 * k) 0x4151 203C + (0x200 * k) 0x01D1 203C + (0x200 * k) 0x420A 203C + (0x200 * k) 0x421A 203C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Secondary Event Register (High Part) The secondary event register is used along with the Event Register (EDMA_TPCC_ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		

Table 16-176. EDMA_TPCC_SERH_RN_k (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 16-177. EDMA_TPCC_SECR_RN_k

Address Offset	0x0000 2040 + (0x200 * k)		
Physical Address	0x4330 2040 + (0x200 * k) 0x40D1 2040 + (0x200 * k) 0x4151 2040 + (0x200 * k) 0x01D1 2040 + (0x200 * k) 0x420A 2040 + (0x200 * k) 0x421A 2040 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC

Table 16-177. EDMA_TPCC_SECR_RN_k (continued)

Description	Secondary Event Clear Register The secondary event clear register is used to clear the status of the EDMA_TPCC_SER registers. CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_SECR.En bit clears the EDMA_TPCC_SER register.
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-178. EDMA_TPCC_SECRH_RN_k

Address Offset	0x0000 2044 + (0x200 * k)

Table 16-178. EDMA_TPCC_SECRH_RN_k (continued)

Physical Address	0x4330 2044 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC
	0x40D1 2044 + (0x200 * k)		DSP1_EDMA_TPCC
	0x4151 2044 + (0x200 * k)		DSP2_EDMA_TPCC
	0x01D1 2044 + (0x200 * k)		DSP_EDMA_TPCC
	0x420A 2044 + (0x200 * k)		EVE1_EDMA_TPCC
	0x421A 2044 + (0x200 * k)		EVE2_EDMA_TPCC
Description	Secondary Event Clear Register (High Part) The secondary event clear register is used to clear the status of the EDMA_TPCC_SERH registers. CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_SECRH.En bit clears the EDMA_TPCC_SERH register.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 16-179. EDMA_TPCC_IER_RN_k

Address Offset	0x0000 2050 + (0x200 * k)		
Physical Address	0x4330 2050 + (0x200 * k) 0x40D1 2050 + (0x200 * k) 0x4151 2050 + (0x200 * k) 0x01D1 2050 + (0x200 * k) 0x420A 2050 + (0x200 * k) 0x421A 2050 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Register EDMA_TPCC_IER.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESR and can be disabled via writes to EDMA_TPCC_IECR register. EDMA_TPCC_IER.In = 0: EDMA_TPCC_IPR.In is NOT enabled for interrupts. EDMA_TPCC_IER.In = 1: EDMA_TPCC_IPR.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0
7	I7	Interrupt associated with TCC #7	R	0x0
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0

Bits	Field Name	Description	Type	Reset
0	I0	Interrupt associated with TCC #0	R	0x0

Table 16-180. EDMA_TPCC_IERH_RN_k

Address Offset	0x0000 2054 + (0x200 * k)		
Physical Address	0x4330 2054 + (0x200 * k) 0x40D1 2054 + (0x200 * k) 0x4151 2054 + (0x200 * k) 0x01D1 2054 + (0x200 * k) 0x420A 2054 + (0x200 * k) 0x421A 2054 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Register (High Part) EDMA_TPCC_IERH.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESRH and can be disabled via writes to EDMA_TPCC_IECRH register. EDMA_TPCC_IERH.In = 0: EDMA_TPCC_IPRH.In is NOT enabled for interrupts. EDMA_TPCC_IERH.In = 1: EDMA_TPCC_IPRH.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0

Bits	Field Name	Description	Type	Reset
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 16-181. EDMA_TPCC_IECR_RN_k

Address Offset	0x0000 2058 + (0x200 * k)		
Physical Address	0x4330 2058 + (0x200 * k) 0x40D1 2058 + (0x200 * k) 0x4151 2058 + (0x200 * k) 0x01D1 2058 + (0x200 * k) 0x420A 2058 + (0x200 * k) 0x421A 2058 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Clear Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IECR.In bit causes the EDMA_TPCC_IER.In bit to be cleared.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0

Bits	Field Name	Description	Type	Reset
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 16-182. EDMA_TPCC_IECRH_RN_k

Address Offset	0x0000 205C + (0x200 * k)		
Physical Address	0x4330 205C + (0x200 * k) 0x40D1 205C + (0x200 * k) 0x4151 205C + (0x200 * k) 0x01D1 205C + (0x200 * k) 0x420A 205C + (0x200 * k) 0x421A 205C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Clear Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IECRH.In bit causes the EDMA_TPCC_IERH.In bit to be cleared.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0

Bits	Field Name	Description	Type	Reset
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 16-183. EDMA_TPCC_IESR_RN_k

Address Offset	0x0000 2060 + (0x200 * k)		
Physical Address	0x4330 2060 + (0x200 * k) 0x40D1 2060 + (0x200 * k) 0x4151 2060 + (0x200 * k) 0x01D1 2060 + (0x200 * k) 0x420A 2060 + (0x200 * k) 0x421A 2060 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Set Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IESR.In bit causes the EDMA_TPCC_IESR.In bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0

Bits	Field Name	Description	Type	Reset
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 16-184. EDMA_TPCC_IESRH_RN_k

Address Offset	0x0000 2064 + (0x200 * k)		
Physical Address	0x4330 2064 + (0x200 * k) 0x40D1 2064 + (0x200 * k) 0x4151 2064 + (0x200 * k) 0x01D1 2064 + (0x200 * k) 0x420A 2064 + (0x200 * k) 0x421A 2064 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Int Enable Set Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IESRH.In bit causes the EDMA_TPCC_IESRH.In bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0

Bits	Field Name	Description	Type	Reset
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 16-185. EDMA_TPCC_IPR_RN_k

Address Offset	0x0000 2068 + (0x200 * k)		
Physical Address	0x4330 2068 + (0x200 * k) 0x40D1 2068 + (0x200 * k) 0x4151 2068 + (0x200 * k) 0x01D1 2068 + (0x200 * k) 0x420A 2068 + (0x200 * k) 0x421A 2068 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Pending Register EDMA_TPCC_IPR.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPR.In bit is cleared via software by writing a '1' to EDMA_TPCC_ICR.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0

Bits	Field Name	Description	Type	Reset
7	I7	Interrupt associated with TCC #7	R	0x0
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0
0	I0	Interrupt associated with TCC #0	R	0x0

Table 16-186. EDMA_TPCC_IPRH_RN_k

Address Offset	0x0000 206C + (0x200 * k)		
Physical Address	0x4330 206C + (0x200 * k) 0x40D1 206C + (0x200 * k) 0x4151 206C + (0x200 * k) 0x01D1 206C + (0x200 * k) 0x420A 206C + (0x200 * k) 0x421A 206C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Pending Register (High Part) EDMA_TPCC_IPRH.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPRH.In bit is cleared via software by writing a '1' to EDMA_TPCC_ICRH.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0

Bits	Field Name	Description	Type	Reset
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 16-187. EDMA_TPCC_ICR_RN_k

Address Offset	0x0000 2070 + (0x200 * k)		
Physical Address	0x4330 2070 + (0x200 * k) 0x40D1 2070 + (0x200 * k) 0x4151 2070 + (0x200 * k) 0x01D1 2070 + (0x200 * k) 0x420A 2070 + (0x200 * k) 0x421A 2070 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Clear Register CPU writes of '0' has no effect. CPU write of '1' to the EDMA_TPCC_ICR.In bit causes the EDMA_TPCC_IPR.In bit to be cleared. All EDMA_TPCC_IPR.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0

Bits	Field Name	Description	Type	Reset
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 16-188. EDMA_TPCC_ICRH_RN_k

Address Offset	0x0000 2074 + (0x200 * k)		
Physical Address	0x4330 2074 + (0x200 * k) 0x40D1 2074 + (0x200 * k) 0x4151 2074 + (0x200 * k) 0x01D1 2074 + (0x200 * k) 0x420A 2074 + (0x200 * k) 0x421A 2074 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Clear Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_ICRH.In bit causes the EDMA_TPCC_IPRH.In bit to be cleared. All EDMA_TPCC_IPRH.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0

Bits	Field Name	Description	Type	Reset
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 16-189. EDMA_TPCC_IEVAL_RN_k

Address Offset	0x0000 2078 + (0x200 * k)		
Physical Address	0x4330 2078 + (0x200 * k) 0x40D1 2078 + (0x200 * k) 0x4151 2078 + (0x200 * k) 0x01D1 2078 + (0x200 * k) 0x420A 2078 + (0x200 * k) 0x421A 2078 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Interrupt Eval Register		
Type	W		

31–2	1	0
RESERVED	SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	SET	Interrupt Set CPU writes 0x0 has no effect. CPU writes 0x1 to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (EDMA_TPCC_IPRn).	W	0x0
0	EVAL	Interrupt Evaluate CPU writes 0x0 has no effect. CPU writes 0x1 to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (EDMA_TPCC_IPRn).	W	0x0

Table 16-190. EDMA_TPCC_QER_RN_k

Address Offset	0x0000 2080 + (0x200 * k)		
Physical Address	0x4330 2080 + (0x200 * k) 0x40D1 2080 + (0x200 * k) 0x4151 2080 + (0x200 * k) 0x01D1 2080 + (0x200 * k) 0x420A 2080 + (0x200 * k) 0x421A 2080 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC

Table 16-190. EDMA_TPCC_QER_RN_k (continued)

Description	QDMA Event Register If EDMA_TPCC_QER.En bit is set, then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. EDMA_TPCC_QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. EDMA_TPCC_QER.En bit is cleared when the corresponding event is prioritized and serviced. EDMA_TPCC_QER.En is also cleared when user writes a '1' to the EDMA_TPCC_QSECR.En bit. If the EDMA_TPCC_QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and EDMA_TPCC_QEER register is set, then the corresponding bit in the QDMA Event Missed Register is set.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-191. EDMA_TPCC_QEER_RN_k

Address Offset	0x0000 2084 + (0x200 * k)		
Physical Address	0x4330 2084 + (0x200 * k) 0x40D1 2084 + (0x200 * k) 0x4151 2084 + (0x200 * k) 0x01D1 2084 + (0x200 * k) 0x420A 2084 + (0x200 * k) 0x421A 2084 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Event Enable Register Enabled/disabled QDMA address comparator for QDMA Channel N. EDMA_TPCC_QEER.En is not directly writeable. The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in EDMA_TPCC_QER.En. EDMA_TPCC_QEER.En = 0, The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in EDMA_TPCC_QER.En. QDMA channels can be enabled via writes to EDMA_TPCC_QEESR and can be disabled via writes to EDMA_TPCC_QEECR register. EDMA_TPCC_QEER.En = 1,		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0

Bits	Field Name	Description	Type	Reset
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-192. EDMA_TPCC_QEECR_RN_k

Address Offset	0x0000 2088 + (0x200 * k)		
Physical Address	0x4330 2088 + (0x200 * k) 0x40D1 2088 + (0x200 * k) 0x4151 2088 + (0x200 * k) 0x01D1 2088 + (0x200 * k) 0x420A 2088 + (0x200 * k) 0x421A 2088 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Event Enable Clear Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_QEECR.En bit causes the EDMA_TPCC_QEER.En bit to be cleared.		
Type	W		

31:8	7	6	5	4	3	2	1	0
RESERVED	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-193. EDMA_TPCC_QEESR_RN_k

Address Offset	0x0000 208C + (0x200 * k)		
Physical Address	0x4330 208C + (0x200 * k) 0x40D1 208C + (0x200 * k) 0x4151 208C + (0x200 * k) 0x01D1 208C + (0x200 * k) 0x420A 208C + (0x200 * k) 0x421A 208C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Event Enable Set Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_QEESR.En bit causes the EDMA_TPCC_QEESR.En bit to be set.		
Type	W		

31:8	7	6	5	4	3	2	1	0
RESERVED	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0

Bits	Field Name	Description	Type	Reset
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-194. EDMA_TPCC_QSER_RN_k

Address Offset	0x0000 2090 + (0x200 * k)		
Physical Address	0x4330 2090 + (0x200 * k) 0x40D1 2090 + (0x200 * k) 0x4151 2090 + (0x200 * k) 0x01D1 2090 + (0x200 * k) 0x420A 2090 + (0x200 * k) 0x421A 2090 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Secondary Event Register The QDMA secondary event register is used along with the QDMA Event Register (EDMA_TPCC_QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 16-195. EDMA_TPCC_QSECR_RN_k

Address Offset	0x0000 2094 + (0x200 * k)		
Physical Address	0x4330 2094 + (0x200 * k) 0x40D1 2094 + (0x200 * k) 0x4151 2094 + (0x200 * k) 0x01D1 2094 + (0x200 * k) 0x420A 2094 + (0x200 * k) 0x421A 2094 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	QDMA Secondary Event Clear Register CPU write of '0' has no effect. The secondary event clear register is used to clear the status of the EDMA_TPCC_QSER and EDMA_TPCC_QER register (note that this is slightly different than the EDMA_TPCC_SER operation, which does not clear the EDMA_TPCC_ER.En register). CPU write of '1' to the EDMA_TPCC_QSECR.En bit clears the EDMA_TPCC_QSER.En and EDMA_TPCC_QER.En register fields.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	write 0's for future compatibility	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 16-196. EDMA_TPCC_OPT_n

Address Offset	0x0000 4000 + (0x20 * n)		
Physical Address	0x4330 4000 + (0x20 * n) 0x40D1 4000 + (0x20 * n) 0x4151 4000 + (0x20 * n) 0x01D1 4000 + (0x20 * n) 0x420A 4000 + (0x20 * n) 0x421A 4000 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Options Parameter		
Type	RW		

31	30:28	27:24	23	22	21	20	19	18	17:12	11	10:8	7:4	3	2	1	0
PRIV	RESERVED	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	WIMODE	RESERVED	TCC	TCCMODE	FWID	RESERVED	STATIC	SYNCDIM	DAM	SAM

Bits	Field Name	Description	Type	Reset
31	PRIV	Privilege level privilege level (supervisor vs. user) for the host/cpu/dma that programmed this PaRAM Entry. Value is set with the vbus priv value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus. 0x0: User level privilege 0x1: Supervisor level privilege	R	0x0
30:28	RESERVED	Reserved	R	0x0
27:24	PRIVID	Privilege ID Privilege ID for the external host/cpu/dma that programmed this PaRAM Entry. This value is set with the vbus privid value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus.	R	0x0
23	ITCCHEN	Intermediate transfer completion chaining enable 0x0: Intermediate transfer complete chaining is disabled. 0x1: Intermediate transfer complete chaining is enabled.	RW	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	RW	0x0

Bits	Field Name	Description	Type	Reset
21	ITCINTEN	Intermediate transfer completion interrupt enable 0x0: Intermediate transfer complete interrupt is disabled. 0x1: Intermediate transfer complete interrupt is enabled (corresponding EDMA_TPCC_IER[TCC] bit must be set to 1 to generate interrupt)	RW	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled (corresponding EDMA_TPCC_IER[TCC] bit must be set to 1 to generate interrupt)	RW	0x0
19	WIMODE	Backward compatibility mode 0x0: Normal operation 0x1: WI Backwards Compatibility mode, forces BCNT to be adjusted by '1' upon TR submission (0 means 1, 1 means 2, ...) and forces ACNT to be treated as a word-count (left shifted by 2 by hardware to create byte cnt for TR submission)	RW	0x0
18	RESERVED	Reserved	R	0x0
17:12	TCC	Transfer Complete Code The 6-bit code is used to set the relevant bit in EDMA_TPCC_CER (bit EDMA_TPCC_CER[TCC]) for chaining or in EDMA_TPCC_IER (bit EDMA_TPCC_IER[TCC]) for interrupts.	RW	0x0
11	TCCMODE	Transfer complete code mode: Indicates the point at which a transfer is considered completed. Applies to both chaining and interrupt. 0x0: Normal Completion. A transfer is considered completed after the transfer parameters are returned to the CC from the TC (which was returned from the peripheral) 0x1: Early Completion, A transfer is considered completed after the CC submits a TR to the TC. CC generates completion code internally.	RW	0x0
10:8	FWID	FIFO width: Applies if either SAM or DAM is set to FIFO mode. Pass-thru to TC.	RW	0x0
7:4	RESERVED	Reserved	R	0x0
3	STATIC	Static Entry 0x0: Entry is updated as normal 0x1: Entry is static, Count and Address updates are not updated after TRP is submitted. Linking is not performed.	RW	0x0
2	SYNCDIM	Transfer Synchronization Dimension: 0x0: A-Sync, Each event triggers the transfer of ACNT elements. 0x1: AB-Sync, Each event triggers the transfer of BCNT arrays of ACNT elements.	RW	0x0
1	DAM	Destination Address Mode: Destination Address Mode within an array. Pass-thru to TC. 0x0: INCR, Dst addressing within an array increments. Dst is not a FIFO. 0x1: FIFO, Dst addressing within an array wraps around upon reaching FIFO width.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	SAM	Source Address Mode: Source Address Mode within an array. Pass-thru to TC. 0x0: INCR, Src addressing within an array increments. Source is not a FIFO. 0x1: FIFO, Src addressing within an array wraps around upon reaching FIFO width.	RW	0x0

Table 16-197. EDMA_TPCC_SRC_n

Address Offset	0x0000 4004 + (0x20 * n)		
Physical Address	0x4330 4004 + (0x20 * n) 0x40D1 4004 + (0x20 * n) 0x4151 4004 + (0x20 * n) 0x01D1 4004 + (0x20 * n) 0x420A 4004 + (0x20 * n) 0x421A 4004 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Source Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC																															

Bits	Field Name	Description	Type	Reset
31:0	SRC	Source Address The 32-bit source address parameters specify the starting byte address of the source. If SAM is set to FIFO mode then the user should program the Source address to be aligned to the value specified by the EDMA_TPCC_OPT_n[10:8] FWID field. No errors are recognized here but TC will assert error if this is not true.	RW	0x0

Table 16-198. EDMA_TPCC_ABCNT_n

Address Offset	0x0000 4008 + (0x20 * n)		
Physical Address	0x4330 4008 + (0x20 * n) 0x40D1 4008 + (0x20 * n) 0x4151 4008 + (0x20 * n) 0x01D1 4008 + (0x20 * n) 0x420A 4008 + (0x20 * n) 0x421A 4008 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	A and B byte count		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															

Bits	Field Name	Description	Type	Reset
31:16	BCNT	BCNT: Count for 2nd Dimension: BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT can be anywhere between 1 and 65535. Therefore, the maximum number of arrays in a frame is 65535 (64K-1 arrays). BCNT=1 means 1 array in the frame, and BCNT=0 means 0 arrays in the frame. In normal mode, a BCNT of '0' is considered as either a Null or Dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the EDMA_TPCC_OPT_n.WIMODE bit is set, then the programmed BCNT value will be incremented by '1' before submission to TC. I.e., 0 means 1, 1 means 2, 2 means 3, ..., 0xFFFE means 0xFFFF. A value of 0xFFFF is an illegal value that will be treated as a Null TR.	RW	0x0
15:0	ACNT	ACNT: number of bytes in 1st dimension: ACNT represents the number of bytes within the first dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65535. Therefore, the maximum number of bytes in an array is 65535 bytes (64K-1 bytes). ACNT must be greater than or equal to '1' for a TR to be submitted to TC. An ACNT of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the EDMA_TPCC_OPT_n.WIMODE bit is set then the ACNT field represents a word count. The CC must internally multiply by 4 to translate the word count to a byte count prior to submission to the TC. The 2 MSBs of the 16-bit ACNT are reserved and should always be written as 'b00 by the user. If user writes a value other than 0, it will still be treated as 0 since the multiply-by-4 operation (to translate between a word count and a byte count) will drop the 2 msbits. For dummy and null transfer definition, the ACNT definition will disregard the 2 msbits. I.e., a programmed ACNT value of 0x8000 in WI-mode will be treated as 0 byte transfer, resulting in null or dummy operation dependent on the state of BCNT and CCNT.	RW	0x0

Table 16-199. EDMA_TPCC_DST_n

Address Offset	0x0000 400C + (0x20 * n)		
Physical Address	0x4330 400C + (0x20 * n) 0x40D1 400C + (0x20 * n) 0x4151 400C + (0x20 * n) 0x01D1 400C + (0x20 * n) 0x420A 400C + (0x20 * n) 0x421A 400C + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Destination Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST																															

Bits	Field Name	Description	Type	Reset
31:0	DST	Destination Address: The 32-bit destination address parameters specify the starting byte address of the destination. If DAM is set to FIFO mode then the user should program the Destination address to be aligned to the value specified by the EDMA_TPCC_OPT_n.FWID field. No errors are recognized here but TC will assert error if this is not true.	RW	0x0

Table 16-200. EDMA_TPCC_BIDX_n

Address Offset	0x0000 4010 + (0x20 * n)		
Physical Address	0x4330 4010 + (0x20 * n) 0x40D1 4010 + (0x20 * n) 0x4151 4010 + (0x20 * n) 0x01D1 4010 + (0x20 * n) 0x420A 4010 + (0x20 * n) 0x421A 4010 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Destination 2nd Dimension Index: DBIDX is a 16-bit signed value (2's complement) used for destination address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-Sync and AB-Sync transfers.	RW	0x0
15:0	SBIDX	Source 2nd Dimension Index: SBIDX is a 16-bit signed value (2's complement) used for source address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-sync and AB-sync transfers.	RW	0x0

Table 16-201. EDMA_TPCC_LNK_n

Address Offset	0x0000 4014 + (0x20 * n)		
Physical Address	0x4330 4014 + (0x20 * n) 0x40D1 4014 + (0x20 * n) 0x4151 4014 + (0x20 * n) 0x01D1 4014 + (0x20 * n) 0x420A 4014 + (0x20 * n) 0x421A 4014 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Link and Reload parameters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNTRLD																LINK															

Bits	Field Name	Description	Type	Reset
31:16	BCNTRLD	BCNT Reload: BCNTRLD is a 16-bit unsigned value used to reload the BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-Sync'ed transfers. In this case, the CC decrements the BCNT value by one on each TR submission. When BCNT (conceptually) reaches zero, then the CC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value. For AB-synchronized transfers, the CC submits the BCNT in the TR and therefore the TC is responsible to keep track of BCNT, not thus BCNTRLD is a don't care field.	RW	0x0
15:0	LINK	Link Address: The CC provides a mechanism to reload the current PaRAM Entry upon its natural termination (i.e., after count fields are decremented to '0') with a new PaRAM Entry. This is called 'linking'. The 16-bit parameter LINK specifies the byte address offset in the PaRAM from which the CC loads/reloads the next PaRAM entry in the link. The CC should disregard the value in the upper 2 bits of the LINK field as well as the lower 5-bits of the LINK field. The upper two bits are ignored such that the user can program either the 'literal' byte address of the LINK parameter or the 'PaRAM base-relative' address of the link field. Therefore, if the user uses the literal address with a range from 0x4000 to 0x7FFF, it will be treated as a PaRAM-base-relative value of 0x0000 to 0x3FFF. The lower-5 bits are ignored and treated as 'b00000, thereby guaranteeing that all Link pointers point to a 32-byte aligned PaRAM entry. In the latter case (5-lsbs), behavior is undefined for the user (i.e., don't have to test it). In the former case (2 msbs), user should be able to take advantage of this feature (i.e., do have to test it). If a Link Update is requested to a PaRAM address that is beyond the actual range of implemented PaRAM, then the Link will be treated as a Null Linkand all 0s plus 0xFFFF will be written to the current entry location. A LINK value of 0xFFFF is referred to as a NULL link which should cause the CC to write 0x0 to all entries of the current PaRAM Entry except for the LINK field which is set to 0xFFFF. The Priv/Privid state is overwritten to 0x0 when linking. MSBs and LSBS should not be masked when comparing against the 0xFFFF value. I.e., a value of 0x3FFE is a non-NULL PaRAM link field.	RW	0x0

Table 16-202. EDMA_TPCC_CIDX_n

Address Offset	0x0000 4018 + (0x20 * n)		
Physical Address	0x4330 4018 + (0x20 * n) 0x40D1 4018 + (0x20 * n) 0x4151 4018 + (0x20 * n) 0x01D1 4018 + (0x20 * n) 0x420A 4018 + (0x20 * n) 0x421A 4018 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	Source and destination frame indexes		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCIDX																SCIDX															

Bits	Field Name	Description	Type	Reset
31:16	DCIDX	Destination Frame Index: DCIDX is a 16-bit signed value (2's complement) used for destination address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when DCIDX is applied, the current array in an A-sync transfer is the last array in the frame, while the current array in a ABsync transfer is the first array in the frame.	RW	0x0
15:0	SCIDX	Source Frame Index: SCIDX is a 16-bit signed value (2's complement) used for source address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when SCIDX is applied, the current array in an A-sync transfer is the last array in the frame, while the current array in a AB-sync transfer is the first array in the frame.	RW	0x0

Table 16-203. EDMA_TPCC_CCNT_n

Address Offset	0x0000 401C + (0x20 * n)		
Physical Address	0x4330 401C + (0x20 * n) 0x40D1 401C + (0x20 * n) 0x4151 401C + (0x20 * n) 0x01D1 401C + (0x20 * n) 0x420A 401C + (0x20 * n) 0x421A 401C + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE1_EDMA_TPCC EVE2_EDMA_TPCC
Description	C byte count		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CCNT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW	0x0
15:0	CCNT	CCNT: Count for 3rd Dimension: CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT can be anywhere between 1 and 65535. Therefore, the maximum number of frames in a block is 65535 (64K-1 frames). CCNT of '1' means '1' frame in the block, and CCNT of '0' means '0' frames in the block. A CCNT value of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. WIMODE has no affect on CCNT operation.	RW	0x0

16.2.7.2.2.2 EDMA_TPTC0 and EDMA_TPTC1 Register Description

Table 16-204 through Table 16-240 describe the individual EDMA_TPTC0 and EDMA_TPTC1 module registers.

Table 16-204. EDMA_TPTCn_PID

Address Offset	0x0000 0000

Table 16-204. EDMA_TPTCn_PID (continued)

Physical Address	0x4340 0000 0x4350 0000 0x40D0 5000 0x40D0 6000 0x4150 5000 0x4150 6000 0x01D0 5000 0x01D0 6000	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	Peripheral ID Register		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

Table 16-205. EDMA_TPTCn_TCCFG

Address Offset	0x0000 0004		
Physical Address	0x4340 0004 0x4350 0004 0x40D0 5004 0x40D0 6004 0x4150 5004 0x4150 6004 0x01D0 5004 0x01D0 6004	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	TC Configuration Register		
Type	R		

31–10	9–8	7–6	5–4	3	2–0
RESERVED	DREGDEPTH	RESERVED	BUSWIDTH	RESERVED	FIFOSIZE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reads return 0's	R	0x0
9:8	DREGDEPTH	Dst Register FIFO Depth Parameterization 0x0: 1 entry 0x1: 2 entries 0x2: 4 entries	R	See Table 16-53
7:6	RESERVED	Reads return 0's	R	0x0
5:4	BUSWIDTH	Bus Width Parameterization 0x0: 32-bit 0x1: 64-bit 0x2: 128-bit	R	See Table 16-53
3	RESERVED	Reads return 0's	R	0x0

Bits	Field Name	Description	Type	Reset
2:0	FIFOSIZE	Fifo Size Parameterization 0x0: 32 byte FIFO 0x1: 64 byte FIFO 0x2: 128 byte FIFO 0x3: 256 byte FIFO 0x4: 512 byte FIFO	R	See Table 16-53

Table 16-206. EDMA_TPTCn_TCSTAT

Address Offset	0x0000 0100		
Physical Address	0x4340 0100 0x4350 0100 0x40D0 5100 0x40D0 6100 0x4150 5100 0x4150 6100 0x01D0 5100 0x01D0 6100 0x4208 6100 0x4218 6100 0x4208 7100 0x4218 7100	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	TC Status Register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="19">RESERVED</td><td colspan="2">DFSTR
TPTR</td><td colspan="2">RESE
RVED</td><td>AC
TV</td><td>RE
SE
RV
ED</td><td colspan="3">DSTACTV</td><td>RE
SE
RV
ED</td><td>W
SA
CT
V</td><td>SR
CA
CT
V</td><td>PR
O
G
BU
SY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R Return 0's	0x0
12:11	DFSTRTPTR	Dst FIFO Start Pointer Represents the offset to the head entry of Dst Register FIFO, in units of *entries*. Legal values = 0x0 to 0x3	R	0x0
10:9	RESERVED	Reserved	R Return 0's	0x0
8	ACTV	Channel Active Channel Active is a logical-OR of each of the *BUSY/ACTV signals. The ACTV bit must remain high through the life of a TR. 0x0: Channel is idle 0x1: Channel is busy	R	0x1
7	RESERVED	Reserved	R Return 0's	0x0
6:4	DSTACTV	Destination Active State Specifies the number of TRs that are resident in the Dst Register FIFO at a given instant. Legal values are constrained by the DSTREGDEPTH parameter.	R	0x0
3	RESERVED	Reserved	R Return 0's	0x0

Bits	Field Name	Description	Type	Reset
2	WSACTV	Write Status Active 0x0: Write status is not pending. Write status has been received for all previously issued write commands. 0x1: Write Status is pending. Write status has not been received for all previously issued write commands.	R	0x0
1	SRCACTV	Source Active State 0x0: Source Active set is idle. Any TR written to Prog Set will immediately transition to Source Active set as long as the Dst FIFO Set is not full (DSTFULL == 1). 0x1: Source Active set is busy either performing read transfers or waiting to perform read transfers for current Transfer Request.	R	0x0
0	PROGBUSY	Program Register Set Busy 0x0: Program set idle and is available for programming. 0x1: Program set busy. User should poll for PROGBUSY equal to '0' prior to re-programming the Program Register set.	R	0x0

Table 16-207. EDMA_TPTCn_INTSTAT

Address Offset	0x0000 0104		
Physical Address	0x4340 0104 0x4350 0104 0x40D0 5104 0x40D0 6104 0x4150 5104 0x4150 6104 0x01D0 5104 0x01D0 6104 0x4208 6104 0x4218 6104 0x4208 7104 0x4218 7104	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Interrupt Status Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TRDONE	PROGEMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R Return 0's	0x0
1	TRDONE	TR Done Event Status 0x0: Condition not detected. 0x1: Set when TC has completed a Transfer Request. TRDONE should be set when the write status is returned for the final write of a TR. Cleared when write '1' to INTCLR.TRDONE register bit.	R	0x0

Bits	Field Name	Description	Type	Reset
0	PROGEMPTY	Program Set Empty Event Status 0x0: Condition not detected 0x1: Set when Program Register set transitions to empty state. Cleared when write '1' to EDMA_TPTCn_INTCLR[0] PROGEMPTY register bit.	R	0x0

Table 16-208. EDMA_TPTCn_INTEN

Address Offset	0x0000 0108		
Physical Address	0x4340 0108 0x4350 0108 0x40D0 5108 0x40D0 6108 0x4150 5108 0x4150 6108 0x01D0 5108 0x01D0 6108 0x4208 6108 0x4218 6108 0x4208 7108 0x4218 7108	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Interrupt Enable Register		
Type	RW		

31:2	1	0
RESERVED	TRDONE	PROGEMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	TRDONE	TR Done Event Enable 0x0: TRDONE Event is disabled. 0x1: TRDONE Event is enabled, and contributes to interrupt generation	RW	0x0
0	PROGEMPTY	Program Set Empty Event Enable 0x0: PROGEMPTY Event is disabled. 0x1: PROGEMPTY Event is enabled, and contributes to interrupt generation	RW	0x0

Table 16-209. EDMA_TPTCn_INTCLR

Address Offset	0x0000 010C		
Physical Address	0x4340 010C 0x4350 010C 0x40D0 510C 0x40D0 610C 0x4150 510C 0x4150 610C 0x01D0 510C 0x01D0 610C 0x4208 610C 0x4218 610C 0x4208 710C 0x4218 710C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1

Table 16-209. EDMA_TPTCn_INTCLR (continued)

Description	Interrupt Clear Register
Type	W

31–2	1	0
RESERVED	TRDONE	PROGEMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	TRDONE	TR Done Event Clear Write 0x0: have no effect. Write 0x1: Clear	W	0x0
0	PROGEMPTY	Program Set Empty Event Clear Write 0x0: have no effect. Write 0x1: Clear	W	0x0

Table 16-210. EDMA_TPTCn_INTCMD

Address Offset	0x0000 0110		
Physical Address	0x4340 0110 0x4350 0110 0x40D0 5110 0x40D0 6110 0x4150 5110 0x4150 6110 0x01D0 5110 0x01D0 6110 0x4208 6110 0x4218 6110 0x4208 7110 0x4218 7110	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Interrupt Command Register		
Type	W		

31–2	1	0
RESERVED	SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	SET	Set TPTC interrupt Write 0x0: have no affect. Write 0x1: SET causes TPTC interrupt to be pulsed unconditionally	W	0x0
0	EVAL	Evaluate state of TPTC interrupt Write 0x0: have no affect. 0x1: causes TPTC interrupt to be pulsed if any of the EDMA_TPTCn_INTSTAT bits are set to '1'.	W	0x0

Table 16-211. EDMA_TPTCn_ERRSTAT

Address Offset	0x0000 0120

Table 16-211. EDMA_TPTCn_ERRSTAT (continued)

Physical Address	0x4340 0120 0x4350 0120 0x40D0 5120 0x40D0 6120 0x4150 5120 0x4150 6120 0x01D0 5120 0x01D0 6120 0x4208 6120 0x4218 6120 0x4208 7120 0x4218 7120	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Error Status Register		
Type	R		

31:4	3	2	1	0
RESERVED	MMRAERR	TRERR	RESERVED	BUSERR

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R Return 0's	0x0
3	MMRAERR	MR Address Error 0x0: Condition not detected 0x1: User attempted to read or write to invalid address configuration mMemory map. (Is only be set for non-emulation accesses). No additional error information is recorded.	R	0x0
2	TRERR	TR Error: TR detected that violates FIFO Mode transfer (SAM or DAM is '1') alignment rules or has ACNT or BCNT == 0. No additional error information is recorded.	R	0x0
1	RESERVED	Reserved	R Return 0's	0x0
0	BUSERR	Bus Error Event 0x0: Condition not detected. 0x1: TC has detected an error code on the write response bus or read response bus. Error information is stored in Error Details Register (EDMA_TPTCn_ERRDET).	R	0x0

Table 16-212. EDMA_TPTCn_ERREN

Address Offset	0x0000 0124		
Physical Address	0x4340 0124 0x4350 0124 0x40D0 5124 0x40D0 6124 0x4150 5124 0x4150 6124 0x01D0 5124 0x01D0 6124 0x4208 6124 0x4218 6124 0x4208 7124 0x4218 7124	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Error Enable Register		

Table 16-212. EDMA_TPTCn_ERREN (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MMRAERR	TRERR	RESERVED	BUSERR

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R Return 0's	0x0
3	MMRAERR	Interrupt enable for EDMA_TPTCn_ERRSTAT[3] MMRAERR 0x0: BUSERR is disabled 0x1: MMRAERR is enabled, and contributes to the TPTC error interrupt generation.	RW	0x0
2	TRERR	Interrupt enable for EDMA_TPTCn_ERRSTAT[2] TRERR 0x0: BUSERR is disabled. 0x1: TRERR is enabled, and contributes to the TPTC error interrupt generation.	RW	0x0
1	RESERVED	Reserved	R Return 0's	0x0
0	BUSERR	Interrupt enable for EDMA_TPTCn_ERRSTAT[0] BUSERR 0x0: BUSERR is disabled. 0x1: BUSERR is enabled, and contributes to the TPTC error interrupt generation.	RW	0x0

Table 16-213. EDMA_TPTCn_ERRCLR

Address Offset	0x0000 0128		
Physical Address	0x4340 0128 0x4350 0128 0x40D0 5128 0x40D0 6128 0x4150 5128 0x4150 6128 0x01D0 5128 0x01D0 6128 0x4208 6128 0x4218 6128 0x4208 7128 0x4218 7128	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Error Clear Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MMRAERR	TRERR	RESERVED	BUSERR

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R Return 0's	0x0

Bits	Field Name	Description	Type	Reset
3	MMRAERR	Interrupt clear for EDMA_TPTCn_ERRSTAT[3] MMRAERR Write 0x0: have no effect Write 0x1: to clear EDMA_TPTCn_ERRSTAT[3] MMRAERR bit. Write of '1' to EDMA_TPTCn_ERRSTAT[3] MMRAERR does not clear the ERRDET register.	W	0x0
2	TRERR	Interrupt clear for EDMA_TPTCn_ERRSTAT[2] TRERR Write 0x0: have no effect Write 0x1: to clear EDMA_TPTCn_ERRSTAT[2] TRERR bit. Write of '1' to EDMA_TPTCn_ERRSTAT[2] TRERR does not clear the ERRDET register.	W	0x0
1	RESERVED	Reserved	R Return 0's	0x0
0	BUSERR	Interrupt clear for EDMA_TPTCn_ERRSTAT[0] BUSERR Write 0x0: have no effect Write 0x1: to clear EDMA_TPTCn_ERRSTAT[0] BUSERR bit Write of '1' to EDMA_TPTCn_ERRSTAT[0] BUSERR clears the ERRDET register.	W	0x0

Table 16-214. EDMA_TPTCn_ERRDET

Address Offset	0x0000 012C		
Physical Address	0x4340 012C 0x4350 012C 0x40D0 512C 0x40D0 612C 0x4150 512C 0x4150 612C 0x01D0 512C 0x01D0 612C 0x4208 612C 0x4218 612C 0x4208 712C 0x4218 712C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Error Details Register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td>TCCHEN</td><td>TCINTEN</td><td colspan="2">RESERVED</td><td colspan="6">TCC</td><td colspan="4">RESERVED</td><td colspan="4">STAT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R Return 0's	0x0
17	TCCHEN	Contains the EDMA_TPCC_OPT_n[17] TCCHEN value programmed by the user for the Read or Write transaction that resulted in an error.	R	0x0
16	TCINTEN	Contains the EDMA_TPCC_OPT_n[16] TCINTEN value programmed by the user for the Read or Write transaction that resulted in an error.	R	0x0
15:14	RESERVED	Reserved	R Return 0's	0x0

Bits	Field Name	Description	Type	Reset
13:8	TCC	Transfer Complete Code: Contains the EDMA_TPCC_OPT_n[13:8] TCC value programmed by the user for the Read or Write transaction that resulted in an error.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0
3:0	STAT	Transaction Status: Stores the non-zero status/error code that was detected on the read status or write status bus. MS-bit effectively serves as the read vs. write error code. If read status and write status are returned on the same cycle, then the TC chooses non-zero version. If both are non-zero then write status is treated as higher priority. Encoding of errors matches the CBA spec and is summarized here: 0xF =	R	0x0

Table 16-215. EDMA_TPTCn_ERRCMD

Address Offset	0x0000 0130		
Physical Address	0x4340 0130 0x4350 0130 0x40D0 5130 0x40D0 6130 0x4150 5130 0x4150 6130 0x01D0 5130 0x01D0 6130 0x4208 6130 0x4218 6130 0x4208 7130 0x4218 7130	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Error Command Register		
Type	W		

31–2	1	0
RESERVED	SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	SET	Set TPTC error interrupt Write 0x0: have no affect Write 0x1: to SET causes TPTC error interrupt to be pulsed unconditionally.	W	0x0
0	EVAL	Evaluate state of TPTC error interrupt Write of '1' Write 0x0: have no affect Write 0x1: to EVAL causes TPTC error interrupt to be pulsed if any of the EDMA_TPTCn_ERRSTAT bits are set to '1'.	W	0x0

Table 16-216. EDMA_TPTCn_RDRATE

Address Offset	0x0000 0140

Table 16-216. EDMA_TPTCn_RDRATE (continued)

Physical Address	0x4340 0140 0x4350 0140 0x40D0 5140 0x40D0 6140 0x4150 5140 0x4150 6140 0x01D0 5140 0x01D0 6140 0x4208 6140 0x4218 6140 0x4208 7140 0x4218 7140	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Read Rate Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													RDRATE		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0
2:0	RDRATE	Read Rate Control: Controls the number of cycles between read commands. This is a global setting that applies to all TRs for this TC.	RW	0x0

Table 16-217. EDMA_TPTCn_POPT

Address Offset	0x0000 0200		
Physical Address	0x4340 0200 0x4350 0200 0x40D0 5200 0x40D0 6200 0x4150 5200 0x4150 6200 0x01D0 5200 0x01D0 6200 0x4208 6200 0x4218 6200 0x4208 7200 0x4218 7200	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Program Set Options		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									TCCHEN	RESERVED	TCINTEN	RESERVED		TCC						RESERVED	FWID			RESERVED	PRI			RESERVED		DAM	SAM

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	RW	0x0
21	RESERVED	Reserved	R	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled.	RW	0x0
19:18	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
17:12	TCC	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or EDMA_TPCC_IPR of the TPCC module.	RW	0x0
11	RESERVED	Reserved	R	0x0
10:8	FWID	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.	RW	0x0
7	RESERVED	Reserved	R	0x0
6:4	PRI	Transfer Priority: 0x0: Priority 0 - Highest priority 0x1: Priority 1 ... 0x7: Priority 7 - Lowest priority	RW	0x0
3:2	RESERVED	Reserved	R	0x0
1	DAM	Destination Address Mode within an array 0x0: INCR, Destination addressing within an array increments. 0x1: FIFO, Destination addressing within an array wraps around upon reaching FIFO width.	RW	0x0
0	SAM	Source Address Mode within an array 0x0: INCR, Source addressing within an array increments. 0x1: FIFO, Source addressing within an array wraps around upon reaching FIFO width.	RW	0x0

Table 16-218. EDMA_TPTCn_PSRC

Address Offset	0x0000 0204		
Physical Address	0x4340 0204 0x4350 0204 0x40D0 5204 0x40D0 6204 0x4150 5204 0x4150 6204 0x01D0 5204 0x01D0 6204 0x4208 6204 0x4218 6204 0x4208 7204 0x4218 7204	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Program Set Source Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															

Bits	Field Name	Description	Type	Reset
31:0	SADDR	Source address for Program Register Set	RW	0x0

Table 16-219. EDMA_TPTCn_PCNT

Address Offset	0x0000 0208

Table 16-219. EDMA_TPTCn_PCNT (continued)

Physical Address	0x4340 0208 0x4350 0208 0x40D0 5208 0x40D0 6208 0x4150 5208 0x4150 6208 0x01D0 5208 0x01D0 6208 0x4208 6208 0x4218 6208 0x4208 7208 0x4218 7208	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Program Set Count		
Type	RW		

31–16	15–0
BCNT	ACNT

Bits	Field Name	Description	Type	Reset
31:16	BCNT	B-Dimension count. Number of arrays to be transferred, where each array is ACNT in length.	RW	0x0
15:0	ACNT	A-Dimension count. Number of bytes to be transferred in first dimension.	RW	0x0

Table 16-220. EDMA_TPTCn_PDST

Address Offset	0x0000 020C		
Physical Address	0x4340 020C 0x4350 020C 0x40D0 520C 0x40D0 620C 0x4150 520C 0x4150 620C 0x01D0 520C 0x01D0 620C 0x4208 620C 0x4218 620C 0x4208 720C 0x4218 720C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Program Set Destination Address		
Type	RW		

31–0
DADDR

Bits	Field Name	Description	Type	Reset
31:0	DADDR	Destination address for Program Register Set	RW	0x0

Table 16-221. EDMA_TPTCn_PBIDX

Address Offset	0x0000 0210

Table 16-221. EDMA_TPTCn_PBIDX (continued)

Physical Address	0x4340 0210 0x4350 0210 0x40D0 5210 0x40D0 6210 0x4150 5210 0x4150 6210 0x01D0 5210 0x01D0 6210 0x4208 6210 0x4218 6210 0x4208 7210 0x4218 7210	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Program Set B-Dim Idx		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Dest B-Idx for Program Register Set: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used, regardless of whether DAM is Increment or FIFO mode.	RW	0x0
15:0	SBIDX	Source B-Idx for Program Register Set: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used, regardless of whether SAM is Increment or FIFO mode.	RW	0x0

Table 16-222. EDMA_TPTCn_PMPPRXY

Address Offset	0x0000 0214		
Physical Address	0x4340 0214 0x4350 0214 0x40D0 5214 0x40D0 6214 0x4150 5214 0x4150 6214 0x01D0 5214 0x01D0 6214 0x4208 6214 0x4218 6214 0x4208 7214 0x4218 7214	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Program Set Memory Protect Proxy		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PRIV	RESERVED				PRIVID			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R Return 0's	0x0

Bits	Field Name	Description	Type	Reset
8	PRIV	Privilege Level 0x0: User level privilege 0x1: Supervisor level privilege EDMA_TPTCn_PMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the PRIV of the external host that sets up the DMA transaction.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0
3:0	PRIVID	Privilege ID: EDMA_TPTCn_PMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the privid of the external host that sets up the DMA transaction.	R	0x0

Table 16-223. EDMA_TPTCn_SAOPT

Address Offset	0x0000 0240		
Physical Address	0x4340 0240 0x4350 0240 0x40D0 5240 0x40D0 6240 0x4150 5240 0x4150 6240 0x01D0 5240 0x01D0 6240 0x4208 6240 0x4218 6240 0x4208 7240 0x4218 7240	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Source Actve Set Options		
Type	R		

31:23	22	21	20	19:18	17:12	11	10:8	7	6:4	3:2	1	0
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED	TCC	RESERVED	FWID	RESERVED	PRI	RESERVED	DAM	SAM

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R Return 0's	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	R	0x0

Bits	Field Name	Description	Type	Reset
21	RESERVED	Reserved	R Return 0's	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled.	R	0x0
19:18	RESERVED	Reserved	R Return 0's	0x0
17:12	TCC	Transfer Complete Code The 6-bit code is used to set the relevant bit in EDMA_TPCC_CER or EDMA_TPCC_IPR of the TPCC module.	R	0x0
11	RESERVED	Reserved	R Return 0's	0x0
10:8	FWID	FIFO width control Applies if either SAM or DAM is set to FIFO mode.	R	0x0
7	RESERVED	Reserved	R Return 0's	0x0
6:4	PRI	Transfer Priority 0x0: Priority 0 - Highest priority 0x1: Priority 1 ... 0x7: Priority 7 - Lowest priority	R	0x0
3:2	RESERVED	Reserved	R Return 0's	0x0
1	DAM	Destination Address Mode within an array 0x0: INCR, Destination addressing within an array increments. 0x1: FIFO, Destination addressing within an array wraps around upon reaching FIFO width.	R	0x0
0	SAM	Source Address Mode within an array 0x0: INCR, Source addressing within an array increments. 0X1: FIFO, Source addressing within an array wraps around upon reaching FIFO width.	R	0x0

Table 16-224. EDMA_TPTCn_SASRC

Address Offset	0x0000 0244		
Physical Address	0x4340 0244 0x4350 0244 0x40D0 5244 0x40D0 6244 0x4150 5244 0x4150 6244 0x01D0 5244 0x01D0 6244 0x4208 6244 0x4218 6244 0x4208 7244 0x4218 7244	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Source Active Set Source Address		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															

Bits	Field Name	Description	Type	Reset
31:0	SADDR	Source address for Source Active Register Set: Initial value is copied from EDMA_TPTCn_PSRC.SADDR. TC updates value according to source addressing mode (EDMA_TPCC_OPT_n.SAM) and/or source index value (BIDX.SBIDX) after each read command is issued. When a TR is complete, the final value should be the address of the last read command issued.	R	0x0

Table 16-225. EDMA_TPTCn_SACNT

Address Offset	0x0000 0248		
Physical Address	0x4340 0248 0x4350 0248 0x40D0 5248 0x40D0 6248 0x4150 5248 0x4150 6248 0x01D0 5248 0x01D0 6248 0x4208 6248 0x4218 6248 0x4208 7248 0x4218 7248	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Source Active Set Count		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															

Bits	Field Name	Description	Type	Reset
31:16	BCNT	B-Dimension count: Number of arrays to be transferred, where each array is ACNT in length. Count Remaining for Source Active Register Set. Represents the amount of data remaining to be read. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each read command is issued. Final value should be 0 when TR is complete.	R	0x0
15:0	ACNT	A-Dimension count: Number of bytes to be transferred in first dimension. Count Remaining for Source Active Register Set. Represents the amount of data remaining to be read. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each read command is issued. Final value should be 0 when TR is complete.	R	0x0

Table 16-226. EDMA_TPTCn_SADST

Address Offset	0x0000 024C		
Physical Address	0x4340 024C 0x4350 024C 0x40D0 524C 0x40D0 624C 0x4150 524C 0x4150 624C 0x01D0 524C 0x01D0 624C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	Source Active Destination Address Register Reserved, return 0x0 w/o AERROR		

Table 16-226. EDMA_TPTCn_SADST (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															

Bits	Field Name	Description	Type	Reset
31:0	DADDR	Destination address is not applicable for Source Active Register Set. Reads return 0x0	R	0x0

Table 16-227. EDMA_TPTCn_SABIDX

Address Offset	0x0000 0250		
Physical Address	0x4340 0250 0x4350 0250 0x40D0 5250 0x40D0 6250 0x4150 5250 0x4150 6250 0x01D0 5250 0x01D0 6250 0x4208 6250 0x4218 6250 0x4208 7250 0x4218 7250	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Source Active Set B-Dim Idx		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Destination B-Idx for Source Active Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used, regardless of whether DAM is Increment or FIFO mode.	R	0x0
15:0	SBIDX	Source B-Idx for Source Active Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used, regardless of whether SAM is Increment or FIFO mode.	R	0x0

Table 16-228. EDMA_TPTCn_SAMPPRXY

Address Offset	0x0000 0254		
Physical Address	0x4340 0254 0x4350 0254 0x40D0 5254 0x40D0 6254 0x4150 5254 0x4150 6254 0x01D0 5254 0x01D0 6254 0x4208 6254 0x4218 6254 0x4208 7254 0x4218 7254	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1

Table 16-228. EDMA_TPTCn_SAMPPRXY (continued)

Description	Source Active Set Mem Protect Proxy
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PRIV	RESERVED				PRIVID			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R Return 0's	0x0
8	PRIV	Privilege Level 0x0: User level privilege 0x1: Supervisor level privilege SAMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the PRIV of the external host that sets up the DMA transaction.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0
3:0	PRIVID	Privilege ID SAMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the privid of the external host that sets up the DMA transaction.	R	0x0

Table 16-229. EDMA_TPTCn_SACNTRLD

Address Offset	0x0000 0258		
Physical Address	0x4340 0258 0x4350 0258 0x40D0 5258 0x40D0 6258 0x4150 5258 0x4150 6258 0x01D0 5258 0x01D0 6258 0x4208 6258 0x4218 6258 0x4208 7258 0x4218 7258	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Source Active Set Count Reload		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACNTRLD															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R Return 0's	0x0

Bits	Field Name	Description	Type	Reset
15:0	ACNTRLD	A-Cnt Reload value for Source Active Register set. Value copied from EDMA_TPTCn_PCNT[15:0] ACNT: Represents the originally programmed value of ACNT. The Reload value is used to reinitialize ACNT after each array is serviced (i.e., ACNT decrements to 0). by the Src offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT bytes)	R	0x0

Table 16-230. EDMA_TPTCn_SASRCBREF

Address Offset	0x0000 025C		
Physical Address	0x4340 025C 0x4350 025C 0x40D0 525C 0x40D0 625C 0x4150 525C 0x4150 625C 0x01D0 525C 0x01D0 625C 0x4208 625C 0x4218 625C 0x4208 725C 0x4218 725C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Source Active Set Source Address A-Reference		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	SADDRBREF	Source address reference for Source Active Register Set: Represents the starting address for the array currently being read. The next array's starting address is calculated as the 'reference address' plus the 'source b-idx' value.	R	0x0

Table 16-231. EDMA_TPTCn_SADSTBREF

Address Offset	0x0000 0260		
Physical Address	0x4340 0260 0x4350 0260 0x40D0 5260 0x40D0 6260 0x4150 5260 0x4150 6260 0x01D0 5260 0x01D0 6260 0x4208 6260 0x4218 6260 0x4208 7260 0x4218 7260	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Source Active Destination Address B-Reference Register Reserved, return 0x0 w/o AERROR		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	DADDRBREF	Destination address reference is not applicable for Src Active Register Set. Reads return 0x0.	R	0x0

Table 16-232. EDMA_TPTCn_DFCNTRLD

Address Offset	0x0000 0280		
Physical Address	0x4340 0280 0x4350 0280 0x40D0 5280 0x40D0 6280 0x4150 5280 0x4150 6280 0x01D0 5280 0x01D0 6280 0x4208 6280 0x4218 6280 0x4208 7280 0x4218 7280	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Destination FIFO Set Count Reload		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACNTRLD															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R Return 0's	0x0
15:0	ACNTRLD	A-Cnt Reload value for Destination FIFO Register set. Value copied from EDMA_TPTCn_PCNT[15:0] ACNT: Represents the originally programmed value of ACNT. The Reload value is used to reinitialize ACNT after each array is serviced (i.e., ACNT decrements to 0). by the Src offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT bytes)	R	0x0

Table 16-233. EDMA_TPTCn_DFSRCBREF

Address Offset	0x0000 0284		
Physical Address	0x4340 0284 0x4350 0284 0x40D0 5284 0x40D0 6284 0x4150 5284 0x4150 6284 0x01D0 5284 0x01D0 6284 0x4208 6284 0x4218 6284 0x4208 7284 0x4218 7284	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Destination FIFO Set Destination Address B Reference Reserved, return 0x0 w/o AERROR		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	SADDRBREF	Source address reference is not applicable for Dst FIFO Register Set. Reads return 0x0.	R	0x0

Table 16-234. EDMA_TPTCn_DFDSTBREF

Address Offset	0x0000 0288		
Physical Address	0x4340 0288 0x4350 0288 0x40D0 5288 0x40D0 6288 0x4150 5288 0x4150 6288 0x01D0 5288 0x01D0 6288	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	Destination FIFO Set Destination Address A-Reference		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	DADDRBREF	Destination address reference for Dst FIFO Register Set: Represents the starting address for the array currently being written. The next array's starting address is calculated as the 'reference address' plus the 'dest bidx' value.	R	0x0

Table 16-235. EDMA_TPTCn_DFOPTi

Address Offset	0x0000 0300 + (0x40 * i)		
Physical Address	0x4340 0300 + (0x40 * i) 0x4350 0300 + (0x40 * i) 0x40D0 5300 + (0x40 * i) 0x40D0 6300 + (0x40 * i) 0x4150 5300 + (0x40 * i) 0x4150 6300 + (0x40 * i) 0x01D0 5300 + (0x40 * i) 0x01D0 6300 + (0x40 * i) 0x4208 6300 + (0x40 * i) 0x4218 6300 + (0x40 * i) 0x4208 7300 + (0x40 * i) 0x4218 7300 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Destination FIFO Set Options		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									TCCHEN	RESERVED	TCINTEN	RESERVED		TCC						RESERVED	FWID			RESERVED	PRI			RESERVED		DAM	SAM

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R Return 0's	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	R	0x0

Bits	Field Name	Description	Type	Reset
21	RESERVED	Reserved	R Return 0's	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled.	R	0x0
19:18	RESERVED	Reserved	R Return 0's	0x0
17:12	TCC	Transfer Complete Code The 6-bit code is used to set the relevant bit in CER or EDMA_TPCC_IPR of the TPCC module.	R	0x0
11	RESERVED	Reserved	R Return 0's	0x0
10:8	FWID	FIFO width control Applies if either SAM or DAM is set to FIFO mode.	R	0x0
7	RESERVED	Reserved	R Return 0's	0x0
6:4	PRI	Transfer Priority 0x0: Priority 0 - Highest priority 0x1: Priority 1 ... 0x7: Priority 7 - Lowest priority	R	0x0
3:2	RESERVED	Reserved	R Return 0's	0x0
1	DAM	Destination Address Mode within an array 0x0: INCR, Dst addressing within an array increments. 0x1: FIFO, Dst addressing within an array wraps around upon reaching FIFO width.	R	0x0
0	SAM	Source Address Mode within an array 0x0: INCR, Source addressing within an array increments. 0x1: FIFO, Source addressing within an array wraps around upon reaching FIFO width.	R	0x0

Table 16-236. EDMA_TPTCn_DFSRCi

Address Offset	0x0000 0304 + (0x40 * i)		
Physical Address	0x4340 0304 + (0x40 * i) 0x4350 0304 + (0x40 * i) 0x40D0 5304 + (0x40 * i) 0x40D0 6304 + (0x40 * i) 0x4150 5304 + (0x40 * i) 0x4150 6304 + (0x40 * i) 0x01D0 5304 + (0x40 * i) 0x01D0 6304 + (0x40 * i) 0x4208 6304 + (0x40 * i) 0x4218 6304 + (0x40 * i) 0x4208 7304 + (0x40 * i) 0x4218 7304 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Destination FIFO source address register Reserved, return 0x0 w/o AERROR		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															

Bits	Field Name	Description	Type	Reset
31:0	SADDR	Source address is not applicable for Dst FIFO Register Set: Reads return 0x0.	R	0x0

Table 16-237. EDMA_TPTCn_DFCNTi

Address Offset	0x0000 0308 + (0x40 * i)		
Physical Address	0x4340 0308 + (0x40 * i) 0x4350 0308 + (0x40 * i) 0x40D0 5308 + (0x40 * i) 0x40D0 6308 + (0x40 * i) 0x4150 5308 + (0x40 * i) 0x4150 6308 + (0x40 * i) 0x01D0 5308 + (0x40 * i) 0x01D0 6308 + (0x40 * i) 0x4208 6308 + (0x40 * i) 0x4218 6308 + (0x40 * i) 0x4208 7308 + (0x40 * i) 0x4218 7308 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	Destination FIFO count register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															

Bits	Field Name	Description	Type	Reset
31:16	BCNT	B-Count Remaining for Dst Register Set: Number of arrays to be transferred, where each array is ACNT in length. Represents the amount of data remaining to be written. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.	R	0x0
15:0	ACNT	A-Count Remaining for Dst Register Set: Number of bytes to be transferred in first dimension. Represents the amount of data remaining to be written. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.	R	0x0

Table 16-238. EDMA_TPTCn_DFDSTi

Address Offset	0x0000 030C + (0x40 * i)		
Physical Address	0x4340 030C + (0x40 * i) 0x4350 030C + (0x40 * i) 0x40D0 530C + (0x40 * i) 0x40D0 630C + (0x40 * i) 0x4150 530C + (0x40 * i) 0x4150 630C + (0x40 * i) 0x01D0 530C + (0x40 * i) 0x01D0 630C + (0x40 * i) 0x4208 630C + (0x40 * i) 0x4218 630C + (0x40 * i) 0x4208 730C + (0x40 * i) 0x4218 730C + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	The destination FIFO destination address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DADDR

Bits	Field Name	Description	Type	Reset
31:0	DADDR	Destination address for Dst FIFO Register Set: Initial value is copied from EDMA_TPTCn_PDST[31:0] DADDR. TC updates value according to destination addressing mode (EDMA_TPCC_OPT_n. SAM) and/or dest index value (BIDX. DBIDX) after each write command is issued. When a TR is complete, the final value should be the address of the last write command issued.	R	0x0

Table 16-239. EDMA_TPTCn_DFBIDXi

Address Offset	0x0000 0310 + (0x40 * i)		
Physical Address	0x4340 0310 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0
	0x4350 0310 + (0x40 * i)		SYS_EDMA_TPTC1
	0x40D0 5310 + (0x40 * i)		DSP1_EDMA_TPTC0
	0x40D0 6310 + (0x40 * i)		DSP1_EDMA_TPTC1
	0x4150 5310 + (0x40 * i)		DSP2_EDMA_TPTC0
	0x4150 6310 + (0x40 * i)		DSP2_EDMA_TPTC1
	0x01D0 5310 + (0x40 * i)		DSP_EDMA_TPTC0
	0x01D0 6310 + (0x40 * i)		DSP_EDMA_TPTC1
	0x4208 6310 + (0x40 * i)		EVE1_EDMA_TPTC0
	0x4218 6310 + (0x40 * i)		EVE2_EDMA_TPTC0
	0x4208 7310 + (0x40 * i)		EVE1_EDMA_TPTC1
	0x4218 7310 + (0x40 * i)		EVE2_EDMA_TPTC1
Description	The destination FIFO B-index register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Dest B-Idx for Dest FIFO Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used, regardless of whether DAM is Increment or FIFO mode.	R	0x0
15:0	SBIDX	Dest B-Idx for Dest FIFO Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used, regardless of whether SAM is Increment or FIFO mode.	R	0x0

Table 16-240. EDMA_TPTCn_DFMPPRXYi

Address Offset	0x0000 0314 + (0x40 * i)

Table 16-240. EDMA_TPTCn_DFMPPRXYi (continued)

Physical Address	0x4340 0314 + (0x40 * i) 0x4350 0314 + (0x40 * i) 0x40D0 5314 + (0x40 * i) 0x40D0 6314 + (0x40 * i) 0x4150 5314 + (0x40 * i) 0x4150 6314 + (0x40 * i) 0x01D0 5314 + (0x40 * i) 0x01D0 6314 + (0x40 * i) 0x4208 6314 + (0x40 * i) 0x4218 6314 + (0x40 * i) 0x4208 7314 + (0x40 * i) 0x4218 7314 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC0 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE1_EDMA_TPTC0 EVE2_EDMA_TPTC0 EVE1_EDMA_TPTC1 EVE2_EDMA_TPTC1
Description	The destination FIFO memory protection proxy register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PRIV	RESERVED				PRIVID			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R Return 0's	0x0
8	PRIV	Privilege Level 0x0: User level privilege 0x1: Supervisor level privilege DFMPPRXY0.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the PRIV of the external host that sets up the DMA transaction.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0
3:0	PRIVID	Privilege ID: DFMPPRXY0.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the privid of the external host that sets up the DMA transaction.	R	0x0

Chapter 17

Interrupt Controllers

This chapter describes the interrupt controllers (INTCs) in the device.

17.1 Interrupt Controllers Overview 3968
17.2 Interrupt Controllers Environment 3970
17.3 Interrupt Controllers Integration 3972
17.4 Interrupt Controllers Functional Description 4014

17.1 Interrupt Controllers Overview

The device has a large number of interrupts to service the needs of its many peripherals and subsystems. The MPU, DSP (x2), IPU (x2), and EVE (x2) subsystems are capable of servicing these interrupts via their integrated interrupt controllers. In addition, each processor's interrupt controller is preceded by an Interrupt Controller Crossbar (IRQ_CROSSBAR) that provides flexibility in mapping the device interrupts to processor interrupt inputs. For more information about IRQ crossbar, see Chapter 18, *Control Module*.

Dual Cortex®-A15 MPU Subsystem Interrupt Controller (MPU_INTC)

The MPU_INTC module (also called Generalized Interrupt Controller [GIC]) is a single functional unit that is integrated in the Arm® Cortex-A15 multiprocessor core (MPCore) alongside Cortex-A15 processors. It provides:

- 160 hardware interrupt inputs
- Generation of interrupts by software
- Prioritization of interrupts
- Masking of any interrupts
- Distribution of the interrupts to the target Cortex-A15 processor(s)
- Tracking the status of interrupts

Each Cortex-A15 processor supports three main groups of interrupt sources, with each interrupt source having a unique ID:

- *Software Generated Interrupts (SGIs):* SGIs are generated by writing to the Cortex-A15 Software Generated Interrupt Register (GICD_SGIR). A maximum of 16 SGIs (ID0–ID15) can be generated for each CPU interface. An SGI has edge-triggered properties. The software triggering of the interrupt is equivalent to the edge transition of the interrupt signal on a peripheral input.
- *Private Peripheral Interrupts (PPIs):* A PPI is an interrupt generated by a peripheral that is specific to a single processor. Although interrupts ID16–ID31 are dedicated to PPIs in general, only seven PPIs are actually used for each CPU interface (ID25–ID31). Interrupts ID16–ID24 are reserved (not used).
- *Shared Peripheral Interrupts (SPIs):* SPIs are triggered by events generated on associated interrupt input lines. In this device, the GIC is configured to support 160 SPIs corresponding to its external IRQS[159:0] signals. SPIs start at ID32 and their mapping is presented in Section 17.3.1.

For detailed information about this module and description of SGIs and PPIs, see the Arm *Cortex-A15 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

C66x DSP Subsystem Interrupt Controller (DSPx_INTC, where x = 1, 2)

There are two Digital Signal Processing (DSP) subsystems in the device - DSP1, and DSP2. Each DSP subsystem integrates an interrupt controller - DSPx_INTC, which interfaces the system events to the C66x core interrupt and exceptions inputs. It combines up to 128 interrupts into 12 prioritized interrupts presented to the C66x CPU.

For detailed information about this module, see Chapter 5, *DSP Subsystem*.

Dual Cortex-M4 IPU Subsystem Interrupt Controller (IPUx_Cx_INTC, where x = 1, 2)

There are two Image Processing Unit (IPU) subsystems in the device - IPU1, and IPU2. Each IPU subsystem integrates two Arm Cortex-M4 cores.

A Nested Vectored Interrupt Controller (NVIC) is integrated within each Cortex-M4. The interrupt mapping is the same (per IPU) for the two cores to facilitate parallel processing. The NVIC supports:

- 64 external interrupts (in addition to 16 Cortex-M4 internal interrupts), which are dynamically prioritized with 16 levels of priority defined for each core
- Low-latency exception and interrupt handling
- Prioritization and handling of exceptions
- Control of the local power management
- Debug accesses to the processor core

For detailed information about this module, refer to Arm *Cortex-M4 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

Note

IPUx_Cx_INTC is a unified name for the following interrupt controllers:

- IPU1_C0_INTC - NVIC in first Cortex-M4 core of IPU1
- IPU1_C1_INTC - NVIC in second Cortex-M4 core of IPU1
- IPU2_C0_INTC - NVIC in first Cortex-M4 core of IPU2
- IPU2_C1_INTC - NVIC in second Cortex-M4 core of IPU2

EVE Subsystem Interrupt Controller (EVEx_INTC, where x = 1 to 2)

There are two Embedded Video Engine (EVE) subsystems in the device - EVE1, and EVE2. Each EVE subsystem integrates an interrupt controller - EVEx_INTC, which handles incoming interrupts, merging them with internal interrupt sources to drive ARP32's interrupt inputs. It also allows ARP32 to generate outgoing interrupts or events to synchronize with other system processors and EDMA.

The EVEx_INTC supports up to 32 active-high level interrupt inputs. Its architecture allows both hardware and software prioritization.

For detailed information about this module, see Section 8.1, *Embedded Vision Engine*.

Figure 17-1 shows the top-level block diagram of the interrupt controllers in this device.

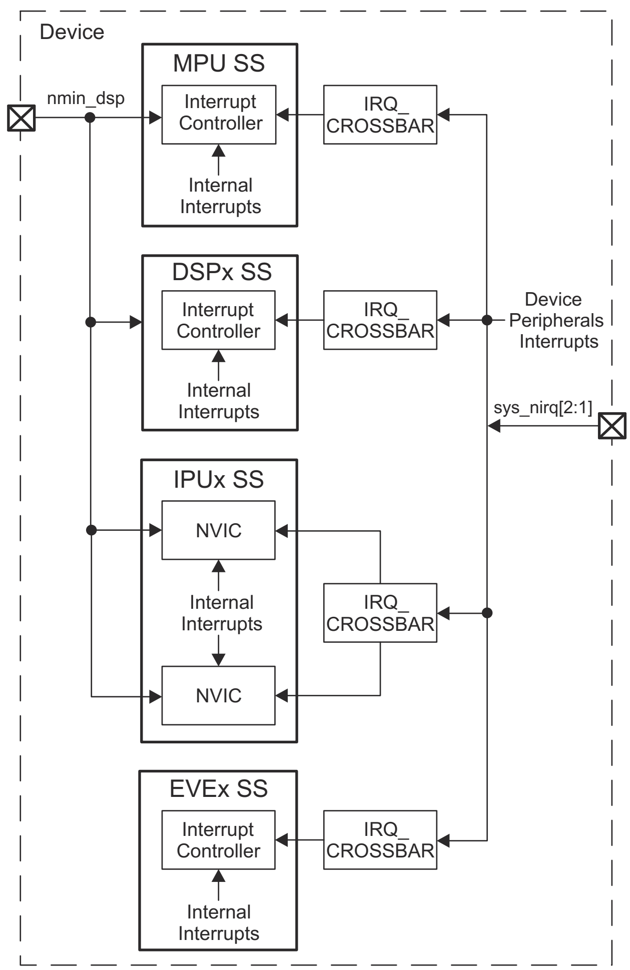


Figure 17-1. Interrupt Controllers in the Device

17.2 Interrupt Controllers Environment

Figure 17-2 shows the relationship between the device INTCs and external interrupts.

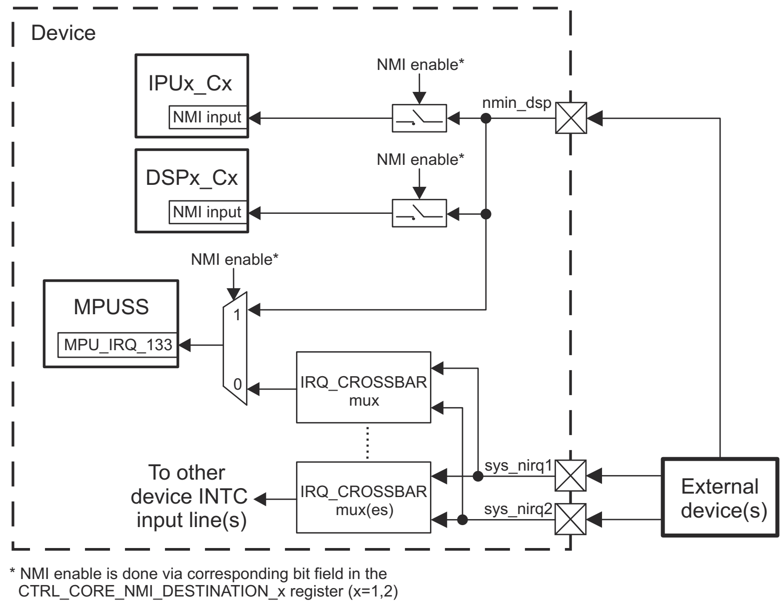


Figure 17-2. Interrupts From External Devices

Table 17-1 describes the signals that can be used by external devices to generate interrupts to the device INTCs.

Table 17-1. Interrupts From External Devices

Device Pin	I/O[(1)]	Description
sys_nirq1	I	External devices can use these pins to generate a system wake-up interrupt event to any device INTC. The user must take care to configure the corresponding IRQ_CROSSBAR properly (via Control Module).
sys_nirq2	I	

Table 17-1. Interrupts From External Devices (continued)

Device Pin	I/O[(1)]	Description
nmin_dsp	I	External device can use this pin to generate a non-maskable interrupt (NMI) event to the following device processors: - IPU1_C0, IPU1_C1. Upon enable (via Control Module), the NMI is routed directly to the NMI input of Cortex M4 core. Note that NMI can be enabled separately for IPU1_C0 and IPU1_C1. - IPU2_C0, IPU2_C1. Upon enable (via Control Module), the NMI is routed directly to the NMI input of Cortex M4 core. Note that NMI can be enabled separately for IPU2_C0 and IPU2_C1. - DSP1, DSP2. Upon enable (via Control Module), the NMI is routed directly to the NMI input of C66x CPU. The signal from the **nmin_dsp** pin can also be mapped to the MPU_INTC (MPU_IRQ_133 input) but it would be treated by Cortex-A15 as a general interrupt and not as a non-maskable interrupt. The Cortex-A15 processor does not provide NMI input. The Control Module registers CTRL_CORE_NMI_DESTINATION_1 and CTRL_CORE_NMI_DESTINATION_2 are used to route the **NMI** signal to the corresponding device processor subsystems as follows: • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_1[23:16] IPU2_C1 enables NMI mapping to IPU2_C1; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_1[15:8] IPU2_C0 enables NMI mapping to IPU2_C0; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_1[7:0] IPU1_C1 enables NMI mapping to IPU1_C1; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_2[31:24] IPU1_C0 enables NMI mapping to IPU1_C0; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_2[23:16] DSP2 enables NMI mapping to DSP2; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_2[15:8] DSP1 enables NMI mapping to DSP1; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_2[7:0] MPU enables NMI mapping to MPU_IRQ_133 interrupt line of MPU_INTC; writing 0x0 to this bit field disables the NMI mapping and IRQ_CROSSBAR mux is mapped to MPU_IRQ_133 instead. For more information about CTRL_CORE_NMI_DESTINATION_1 and CTRL_CORE_NMI_DESTINATION_2 registers, see Chapter 18, *Control Module*.

(1) I = Input, O = Output

Note

External devices can also use the GPIO modules to generate an interrupt to the device INTCs. For more information, see Chapter 27, *General-Purpose Interface*.

17.3 Interrupt Controllers Integration

Table 17-2 through Table 17-8 present the default interrupt mapping of the device INTCs. The mapping of device interrupts to IRQ_CROSSBAR inputs is presented in Table 17-9.

Note

All device interrupts (external to the MPU, DSP [x2], IPU [x2], and EVE [x2] subsystems) are active-high, level-sensitive.

CAUTION

A single interrupt source can be physically mapped to multiple INTCs. With multiple-mapped interrupts, it is strongly recommended to unmask each interrupt source in only one INTC at a time.

17.3.1 Interrupt Requests to MPU_INTC

Table 17-2 lists the default interrupt sources for the MPU_INTC. In addition, interrupts MPU_IRQ_4, MPU_IRQ_[130:7], MPU_IRQ_[138:133], and MPU_IRQ_[159:141] can alternatively be sourced through the MPU's IRQ_CROSSBAR from one of the 420 multiplexed device interrupts listed in Table 17-9. The CTRL_CORE_MPU_IRQ_y_z registers (where *y* and *z* are indexes of INTC input lines) in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Note

The interrupts listed in Table 17-2 are also called Shared Peripheral Interrupts (SPIs) in Arm Cortex-A15 terminology. That is, the MPU_IRQ_[159:0] interrupt inputs correspond to the **IRQS[N:0]** GIC signals (N = 159 for this device), described in the Arm *Cortex-A15 MPCore Processor Technical Reference Manual*.

The association between the Shared Peripheral Interrupts (MPU_IRQ_0 – MPU_IRQ_159) and the GIC inputs (ID32 – ID191) is shown in the first column of this table.

Table 17-2. MPU_INTC Default Interrupt Mapping

IRQ Input Line (GIC ID Number) (1)	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_0 (ID32)	N/A	N/A	N/A	MPU_CLUSTER_IRQ_INTERR	Illegal writes to interrupt controller memory map region
MPU_IRQ_1 (ID33)	N/A	N/A	N/A	CS_CTI_MPU_C0_IRQ	TRIGOUT[6] of Cross Trigger Interface 0 (CTI0)
MPU_IRQ_2 (ID34)	N/A	N/A	N/A	CS_CTI_MPU_C1_IRQ	TRIGOUT[6] of Cross Trigger Interface 1 (CTI1)
MPU_IRQ_3 (ID35)	N/A	N/A	N/A	MPU_CLUSTER_IRQ_AXIERR	Error indication for AXI write transactions with a BRESP error condition
MPU_IRQ_4 (ID36)	1	CTRL_CORE_MPU_IRQ_4_7[8:0]	1	ELM_IRQ	Error location process completion interrupt
MPU_IRQ_5 (ID37)	N/A	N/A	N/A	WD_TIMER_MPU_C0_IRQ_WARN	MPU_WD_TIMER channel 0 warning interrupt
MPU_IRQ_6 (ID38)	N/A	N/A	N/A	WD_TIMER_MPU_C1_IRQ_WARN	MPU_WD_TIMER channel 1 warning interrupt
MPU_IRQ_7 (ID39)	2	CTRL_CORE_MPU_IRQ_4_7[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_8 (ID40)	3	CTRL_CORE_MPU_IRQ_8_9[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 18.4.6.14.3.
MPU_IRQ_9 (ID41)	4	CTRL_CORE_MPU_IRQ_8_9[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
MPU_IRQ_10 (ID42)	5	CTRL_CORE_MPU_IRQ_10_11[8:0] (not functional)	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error(2)
MPU_IRQ_11 (ID43)	6	CTRL_CORE_MPU_IRQ_10_11[24:16]	6	PRM_IRQ_MPU	PRCM interrupt to MPU
MPU_IRQ_12 (ID44)	7	CTRL_CORE_MPU_IRQ_12_13[8:0]	7	DMA_SYSTEM_IRQ_0	System DMA interrupt 0
MPU_IRQ_13 (ID45)	8	CTRL_CORE_MPU_IRQ_12_13[24:16]	8	DMA_SYSTEM_IRQ_1	System DMA interrupt 1
MPU_IRQ_14 (ID46)	9	CTRL_CORE_MPU_IRQ_14_15[8:0]	9	DMA_SYSTEM_IRQ_2	System DMA interrupt 2
MPU_IRQ_15 (ID47)	10	CTRL_CORE_MPU_IRQ_14_15[24:16]	10	DMA_SYSTEM_IRQ_3	System DMA interrupt 3
MPU_IRQ_16 (ID48)	11	CTRL_CORE_MPU_IRQ_16_17[8:0]	11	L3_MAIN_IRQ_STAT_ALARM	L3_MAIN statistic collector alarm interrupt
MPU_IRQ_17 (ID49)	12	CTRL_CORE_MPU_IRQ_16_17[24:16]	12	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_18 (ID50)	13	CTRL_CORE_MPU_IRQ_18_19[8:0]	13	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_19 (ID51)	14	CTRL_CORE_MPU_IRQ_18_19[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_20 (ID52)	15	CTRL_CORE_MPU_IRQ_20_21[8:0]	15	GPMC_IRQ	GPMC interrupt
MPU_IRQ_21 (ID53)	16	CTRL_CORE_MPU_IRQ_20_21[24:16]	16	GPU_IRQ	GPU interrupt
MPU_IRQ_22 (ID54)	17	CTRL_CORE_MPU_IRQ_22_23[8:0]	17	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_23 (ID55)	18	CTRL_CORE_MPU_IRQ_22_23[24:16]	18	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_24 (ID56)	19	CTRL_CORE_MPU_IRQ_24_25[8:0]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_25 (ID57)	20	CTRL_CORE_MPU_IRQ_24_25[24:16]	20	DISPC_IRQ	Display controller interrupt
MPU_IRQ_26 (ID58)	21	CTRL_CORE_MPU_IRQ_26_27[8:0]	21	MAILBOX1_IRQ_USER0	Mailbox 1 user 0 interrupt
MPU_IRQ_27 (ID59)	22	CTRL_CORE_MPU_IRQ_26_27[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_28 (ID60)	23	CTRL_CORE_MPU_IRQ_28_29[8:0]	23	DSP1_IRQ_MMU0	DSP1 MMU0 interrupt

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_29 (ID61)	24	CTRL_CORE_MPU_IRQ_28_29[24:16]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
MPU_IRQ_30 (ID62)	25	CTRL_CORE_MPU_IRQ_30_31[8:0]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
MPU_IRQ_31 (ID63)	26	CTRL_CORE_MPU_IRQ_30_31[24:16]	26	GPIO3_IRQ_1	GPIO3 interrupt 1
MPU_IRQ_32 (ID64)	27	CTRL_CORE_MPU_IRQ_32_33[8:0]	27	GPIO4_IRQ_1	GPIO4 interrupt 1
MPU_IRQ_33 (ID65)	28	CTRL_CORE_MPU_IRQ_32_33[24:16]	28	GPIO5_IRQ_1	GPIO5 interrupt 1
MPU_IRQ_34 (ID66)	29	CTRL_CORE_MPU_IRQ_34_35[8:0]	29	GPIO6_IRQ_1	GPIO6 interrupt 1
MPU_IRQ_35 (ID67)	30	CTRL_CORE_MPU_IRQ_34_35[24:16]	30	GPIO7_IRQ_1	GPIO7 interrupt 1
MPU_IRQ_36 (ID68)	31	CTRL_CORE_MPU_IRQ_36_37[8:0]	31	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_37 (ID69)	32	CTRL_CORE_MPU_IRQ_36_37[24:16]	32	TIMER1_IRQ	TIMER1 interrupt
MPU_IRQ_38 (ID70)	33	CTRL_CORE_MPU_IRQ_38_39[8:0]	33	TIMER2_IRQ	TIMER2 interrupt
MPU_IRQ_39 (ID71)	34	CTRL_CORE_MPU_IRQ_38_39[24:16]	34	TIMER3_IRQ	TIMER3 interrupt
MPU_IRQ_40 (ID72)	35	CTRL_CORE_MPU_IRQ_40_41[8:0]	35	TIMER4_IRQ	TIMER4 interrupt
MPU_IRQ_41 (ID73)	36	CTRL_CORE_MPU_IRQ_40_41[24:16]	36	TIMER5_IRQ	TIMER5 interrupt
MPU_IRQ_42 (ID74)	37	CTRL_CORE_MPU_IRQ_42_43[8:0]	37	TIMER6_IRQ	TIMER6 interrupt
MPU_IRQ_43 (ID75)	38	CTRL_CORE_MPU_IRQ_42_43[24:16]	38	TIMER7_IRQ	TIMER7 interrupt
MPU_IRQ_44 (ID76)	39	CTRL_CORE_MPU_IRQ_44_45[8:0]	39	TIMER8_IRQ	TIMER8 interrupt
MPU_IRQ_45 (ID77)	40	CTRL_CORE_MPU_IRQ_44_45[24:16]	40	TIMER9_IRQ	TIMER9 interrupt
MPU_IRQ_46 (ID78)	41	CTRL_CORE_MPU_IRQ_46_47[8:0]	41	TIMER10_IRQ	TIMER10 interrupt
MPU_IRQ_47 (ID79)	42	CTRL_CORE_MPU_IRQ_46_47[24:16]	42	TIMER11_IRQ	TIMER11 interrupt
MPU_IRQ_48 (ID80)	43	CTRL_CORE_MPU_IRQ_48_49[8:0]	43	MCSPI4_IRQ	McSPI4 interrupt
MPU_IRQ_49 (ID81)	44	CTRL_CORE_MPU_IRQ_48_49[24:16]	44	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_50 (ID82)	45	CTRL_CORE_MPU_IRQ_50_51[8:0]	45	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_51 (ID83)	46	CTRL_CORE_MPU_IRQ_50_51[24:16]	46	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_52 (ID84)	47	CTRL_CORE_MPU_IRQ_52_53[8:0]	47	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_53 (ID85)	48	CTRL_CORE_MPU_IRQ_52_53[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_54 (ID86)	49	CTRL_CORE_MPU_IRQ_54_55[8:0]	49	SATA_IRQ	SATA interrupt
MPU_IRQ_55 (ID87)	50	CTRL_CORE_MPU_IRQ_54_55[24:16]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_56 (ID88)	51	CTRL_CORE_MPU_IRQ_56_57[8:0]	51	I2C1_IRQ	I2C1 interrupt
MPU_IRQ_57 (ID89)	52	CTRL_CORE_MPU_IRQ_56_57[24:16]	52	I2C2_IRQ	I2C2 interrupt
MPU_IRQ_58 (ID90)	53	CTRL_CORE_MPU_IRQ_58_59[8:0]	53	HDQ1W_IRQ	HDQ1W interrupt
MPU_IRQ_59 (ID91)	54	CTRL_CORE_MPU_IRQ_58_59[24:16]	54	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_60 (ID92)	55	CTRL_CORE_MPU_IRQ_60_61[8:0]	55	I2C5_IRQ	I2C5 interrupt
MPU_IRQ_61 (ID93)	56	CTRL_CORE_MPU_IRQ_60_61[24:16]	56	I2C3_IRQ	I2C3 interrupt
MPU_IRQ_62 (ID94)	57	CTRL_CORE_MPU_IRQ_62_63[8:0]	57	I2C4_IRQ	I2C4 interrupt
MPU_IRQ_63 (ID95)	58	CTRL_CORE_MPU_IRQ_62_63[24:16]	58	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_64 (ID96)	59	CTRL_CORE_MPU_IRQ_64_65[8:0]	59	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_65 (ID97)	60	CTRL_CORE_MPU_IRQ_64_65[24:16]	60	MCSPI1_IRQ	McSPI1 interrupt
MPU_IRQ_66 (ID98)	61	CTRL_CORE_MPU_IRQ_66_67[8:0]	61	MCSPI2_IRQ	McSPI2 interrupt
MPU_IRQ_67 (ID99)	62	CTRL_CORE_MPU_IRQ_66_67[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_68 (ID100)	63	CTRL_CORE_MPU_IRQ_68_69[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_69 (ID101)	64	CTRL_CORE_MPU_IRQ_68_69[24:16]	64	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_70 (ID102)	65	CTRL_CORE_MPU_IRQ_70_71[8:0]	65	UART4_IRQ	UART4 interrupt
MPU_IRQ_71 (ID103)	66	CTRL_CORE_MPU_IRQ_70_71[24:16]	66	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_72 (ID104)	67	CTRL_CORE_MPU_IRQ_72_73[8:0]	67	UART1_IRQ	UART1 interrupt
MPU_IRQ_73 (ID105)	68	CTRL_CORE_MPU_IRQ_72_73[24:16]	68	UART2_IRQ	UART2 interrupt
MPU_IRQ_74 (ID106)	69	CTRL_CORE_MPU_IRQ_74_75[8:0]	69	UART3_IRQ	UART3 interrupt
MPU_IRQ_75 (ID107)	70	CTRL_CORE_MPU_IRQ_74_75[24:16]	70	PBIAS_IRQ	MMC1 PBIAS interrupt (controlled via device Control Module)
MPU_IRQ_76 (ID108)	71	CTRL_CORE_MPU_IRQ_76_77[8:0]	71	USB1_IRQ_INTR0	USB1 interrupt 0
MPU_IRQ_77 (ID109)	72	CTRL_CORE_MPU_IRQ_76_77[24:16]	72	USB1_IRQ_INTR1	USB1 interrupt 1
MPU_IRQ_78 (ID110)	73	CTRL_CORE_MPU_IRQ_78_79[8:0]	73	USB2_IRQ_INTR0	USB2 interrupt 0
MPU_IRQ_79 (ID111)	74	CTRL_CORE_MPU_IRQ_78_79[24:16]	74	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_80 (ID112)	75	CTRL_CORE_MPU_IRQ_80_81[8:0]	75	WD_TIMER2_IRQ	WD_TIMER2 interrupt
MPU_IRQ_81 (ID113)	76	CTRL_CORE_MPU_IRQ_80_81[24:16]	76	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_82 (ID114)	77	CTRL_CORE_MPU_IRQ_82_83[8:0]	77	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_83 (ID115)	78	CTRL_CORE_MPU_IRQ_82_83[24:16]	78	MMC1_IRQ	MMC1 interrupt
MPU_IRQ_84 (ID116)	79	CTRL_CORE_MPU_IRQ_84_85[8:0]	79	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_85 (ID117)	80	CTRL_CORE_MPU_IRQ_84_85[24:16]	80	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_86 (ID118)	81	CTRL_CORE_MPU_IRQ_86_87[8:0]	81	MMC2_IRQ	MMC2 interrupt
MPU_IRQ_87 (ID119)	82	CTRL_CORE_MPU_IRQ_86_87[24:16]	82	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_88 (ID120)	83	CTRL_CORE_MPU_IRQ_88_89[8:0]	83	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_89 (ID121)	84	CTRL_CORE_MPU_IRQ_88_89[24:16]	84	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_90 (ID122)	85	CTRL_CORE_MPU_IRQ_90_91[8:0]	85	DEBUGSS_IRQ_CT_UART	CT_UART interrupt generated when data ready on RX or when TX empty
MPU_IRQ_91 (ID123)	86	CTRL_CORE_MPU_IRQ_90_91[24:16]	86	MCSPI3_IRQ	McSPI3 interrupt
MPU_IRQ_92 (ID124)	87	CTRL_CORE_MPU_IRQ_92_93[8:0]	87	USB2_IRQ_INTR1	USB2 interrupt 1

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_93 (ID125)	88	CTRL_CORE_MPU_IRQ_92_93[24:16]	88	USB3_IRQ_INTR0	USB3 interrupt 0
MPU_IRQ_94 (ID126)	89	CTRL_CORE_MPU_IRQ_94_95[8:0]	89	MMC3_IRQ	MMC3 interrupt
MPU_IRQ_95 (ID127)	90	CTRL_CORE_MPU_IRQ_94_95[24:16]	90	TIMER12_IRQ	TIMER12 interrupt
MPU_IRQ_96 (ID128)	91	CTRL_CORE_MPU_IRQ_96_97[8:0]	91	MMC4_IRQ	MMC4 interrupt
MPU_IRQ_97 (ID129)	92	CTRL_CORE_MPU_IRQ_96_97[24:16]	92	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_98 (ID130)	93	CTRL_CORE_MPU_IRQ_98_99[8:0]	93	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_99 (ID131)	94	CTRL_CORE_MPU_IRQ_98_99[24:16]	94	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_100 (ID132)	95	CTRL_CORE_MPU_IRQ_100_101[8:0]	395	IPU1_IRQ_MMU	IPU1 MMU interrupt
MPU_IRQ_101 (ID133)	96	CTRL_CORE_MPU_IRQ_100_101[24:16]	96	HDMI_IRQ	HDMI interrupt
MPU_IRQ_102 (ID134)	97	CTRL_CORE_MPU_IRQ_102_103[8:0]	97	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_103 (ID135)	98	CTRL_CORE_MPU_IRQ_102_103[24:16]	98	IVA_IRQ_SYNC_1	IVA ICONT2 sync interrupt
MPU_IRQ_104 (ID136)	99	CTRL_CORE_MPU_IRQ_104_105[8:0]	99	IVA_IRQ_SYNC_0	IVA ICONT1 sync interrupt
MPU_IRQ_105 (ID137)	100	CTRL_CORE_MPU_IRQ_104_105[24:16]	100	UART5_IRQ	UART5 interrupt
MPU_IRQ_106 (ID138)	101	CTRL_CORE_MPU_IRQ_106_107[8:0]	101	UART6_IRQ	UART6 interrupt
MPU_IRQ_107 (ID139)	102	CTRL_CORE_MPU_IRQ_106_107[24:16]	102	IVA_IRQ_MAILBOX_0	IVA mailbox user 0 interrupt
MPU_IRQ_108 (ID140)	103	CTRL_CORE_MPU_IRQ_108_109[8:0]	103	McASP1_IRQ_AREVT	McASP1 receive interrupt
MPU_IRQ_109 (ID141)	104	CTRL_CORE_MPU_IRQ_108_109[24:16]	104	McASP1_IRQ_AXEVT	McASP1 transmit interrupt
MPU_IRQ_110 (ID142)	105	CTRL_CORE_MPU_IRQ_110_111[8:0]	105	EMIF1_IRQ	EMIF1 interrupt
MPU_IRQ_111 (ID143)	106	CTRL_CORE_MPU_IRQ_110_111[24:16]	106	EMIF2_IRQ	EMIF2 interrupt
MPU_IRQ_112 (ID144)	107	CTRL_CORE_MPU_IRQ_112_113[8:0]	107	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_113 (ID145)	108	CTRL_CORE_MPU_IRQ_112_113[24:16]	108	DMM_IRQ	DMM interrupt
MPU_IRQ_114 (ID146)	109	CTRL_CORE_MPU_IRQ_114_115[8:0]	109	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_115 (ID147)	110	CTRL_CORE_MPU_IRQ_114_115[24:16]	110	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_116 (ID148)	111	CTRL_CORE_MPU_IRQ_116_117[8:0]	111	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_117 (ID149)	112	CTRL_CORE_MPU_IRQ_116_117[24:16]	112	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_118 (ID150)	113	CTRL_CORE_MPU_IRQ_118_119[8:0]	113	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_119 (ID151)	114	CTRL_CORE_MPU_IRQ_118_119[24:16]	114	EXT_SYS_IRQ_2	External interrupt (active low) via sys_nirq2 pin
MPU_IRQ_120 (ID152)	115	CTRL_CORE_MPU_IRQ_120_121[8:0]	115	KBD_IRQ	Keyboard controller interrupt
MPU_IRQ_121 (ID153)	116	CTRL_CORE_MPU_IRQ_120_121[24:16]	116	GPIO8_IRQ_1	GPIO8 interrupt 1
MPU_IRQ_122 (ID154)	117	CTRL_CORE_MPU_IRQ_122_123[8:0]	117	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_123 (ID155)	118	CTRL_CORE_MPU_IRQ_122_123[24:16]	118	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_124 (ID156)	119	CTRL_CORE_MPU_IRQ_124_125[8:0]	119	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_125 (ID157)	120	CTRL_CORE_MPU_IRQ_124_125[24:16]	120	BB2D_IRQ	BB2D interrupt
MPU_IRQ_126 (ID158)	121	CTRL_CORE_MPU_IRQ_126_127[8:0]	121	CTRL_MODULE_CORE_IRQ_ THERMAL_ALERT	CTRL_MODULE thermal alert interrupt
MPU_IRQ_127 (ID159)	122	CTRL_CORE_MPU_IRQ_126_127[24:16]	122	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_128 (ID160)	123	CTRL_CORE_MPU_IRQ_128_129[8:0]	123	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_129 (ID161)	124	CTRL_CORE_MPU_IRQ_128_129[24:16]	124	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_130 (ID162)	125	CTRL_CORE_MPU_IRQ_130_133[8:0]	125	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_131 (ID163)	N/A	N/A	N/A	MPU_CLUSTER_IRQ_PMU_C0	MPU core 0 PMU interrupt
MPU_IRQ_132 (ID164)	N/A	N/A	N/A	MPU_CLUSTER_IRQ_PMU_C1	MPU core 1 PMU interrupt

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_133 (ID165)	126	CTRL_CORE_MPU_IRQ_130_133[24:16]	0	Reserved	Reserved by default but can be remapped to: • a valid interrupt source (through CTRL_CORE_MPU_IRQ_130_133[24:16] • the nmin_dsp pin (through CTRL_MOD_CTRL_CORE_NMI_DESTINATION_2[7:0] MPU). See Section 17.2, *Interrupt Controllers Environment* for details.
MPU_IRQ_134 (ID166)	127	CTRL_CORE_MPU_IRQ_134_135[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_135 (ID167)	128	CTRL_CORE_MPU_IRQ_134_135[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_136 (ID168)	129	CTRL_CORE_MPU_IRQ_136_137[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_137 (ID169)	130	CTRL_CORE_MPU_IRQ_136_137[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_138 (ID170)	131	CTRL_CORE_MPU_IRQ_138_139[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_139 (ID171)	N/A	N/A	N/A	WD_TIMER_MPU_C0_IRQ	MPU_WD_TIMER channel 0 timeout interrupt (watchdog reset)
MPU_IRQ_140 (ID172)	N/A	N/A	N/A	WD_TIMER_MPU_C1_IRQ	MPU_WD_TIMER channel 1 timeout interrupt (watchdog reset)
MPU_IRQ_141 (ID173)	134	CTRL_CORE_MPU_IRQ_140_141[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_142 (ID174)	135	CTRL_CORE_MPU_IRQ_142_143[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_143 (ID175)	136	CTRL_CORE_MPU_IRQ_142_143[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_144 (ID176)	137	CTRL_CORE_MPU_IRQ_144_145[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_145 (ID177)	138	CTRL_CORE_MPU_IRQ_144_145[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_146 (ID178)	139	CTRL_CORE_MPU_IRQ_146_147[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-2. MPU_INTC Default Interrupt Mapping (continued)

IRQ Input Line (GIC ID Number) (1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
MPU_IRQ_147 (ID179)	140	CTRL_CORE_MPU_IRQ_146_147[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_148 (ID180)	141	CTRL_CORE_MPU_IRQ_148_149[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_149 (ID181)	142	CTRL_CORE_MPU_IRQ_148_149[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_150 (ID182)	143	CTRL_CORE_MPU_IRQ_150_151[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_151 (ID183)	144	CTRL_CORE_MPU_IRQ_150_151[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_152 (ID184)	145	CTRL_CORE_MPU_IRQ_152_153[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_153 (ID185)	146	CTRL_CORE_MPU_IRQ_152_153[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_154 (ID186)	147	CTRL_CORE_MPU_IRQ_154_155[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_155 (ID187)	148	CTRL_CORE_MPU_IRQ_154_155[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_156 (ID188)	149	CTRL_CORE_MPU_IRQ_156_157[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_157 (ID189)	150	CTRL_CORE_MPU_IRQ_156_157[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_158 (ID190)	151	CTRL_CORE_MPU_IRQ_158_159[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
MPU_IRQ_159 (ID191)	152	CTRL_CORE_MPU_IRQ_158_159[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source

(1) This column shows the association between the Shared Peripheral Interrupts (MPU_IRQ_0 – MPU_IRQ_159) and the GIC inputs (ID32 – ID191).
(2) The L3_MAIN_IRQ_APP_ERR interrupt is directly mapped to the MPU_IRQ_10 line, bypassing the crossbar instance dedicated to this line. No other interrupt source can be mapped to MPU_IRQ_10.

Note

The "**IRQ_CROSSBAR Default Input Index**" column of Table 17-2 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding MPU_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_MPU_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to MPU_INTC inputs. For example, the MPU_IRQ_4_7[8:0] bit field is used to configure which device interrupt would be mapped to the MPU_IRQ_4 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to MPU_IRQ_4 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the MPU subsystem. There is no IRQ_CROSSBAR dedicated to the associated MPU_INTC input line and therefore, the user cannot change its default mapping.

17.3.2 Interrupt Requests to DSP1_INTC

Table 17-3 lists the default interrupt sources for the DSP1_INTC. In addition, interrupts DSP1_IRQ_32 through DSP1_IRQ_95 can alternatively be sourced through the DSP1's IRQ_CROSSBAR from one of the 420 multiplexed device interrupts listed in Table 17-9. The CTRL_CORE_DSP1_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 17-3. DSP1_INTC Default Interrupt Mapping

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_0	N/A	N/A	N/A	CGEM_IRQ_0	CGEM Internal Interrupt
DSP1_IRQ_1	N/A	N/A	N/A	CGEM_IRQ_1	CGEM Internal Interrupt
DSP1_IRQ_2	N/A	N/A	N/A	CGEM_IRQ_2	CGEM Internal Interrupt
DSP1_IRQ_3	N/A	N/A	N/A	CGEM_IRQ_3	CGEM Internal Interrupt
DSP1_IRQ_4	N/A	N/A	N/A	CGEM_IRQ_4	CGEM Internal Interrupt
DSP1_IRQ_5	N/A	N/A	N/A	CGEM_IRQ_5	CGEM Internal Interrupt
DSP1_IRQ_6	N/A	N/A	N/A	CGEM_IRQ_6	CGEM Internal Interrupt
DSP1_IRQ_7	N/A	N/A	N/A	CGEM_IRQ_7	CGEM Internal Interrupt
DSP1_IRQ_8	N/A	N/A	N/A	CGEM_IRQ_8	CGEM Internal Interrupt
DSP1_IRQ_9	N/A	N/A	N/A	CGEM_IRQ_9	CGEM Internal Interrupt
DSP1_IRQ_10	N/A	N/A	N/A	CGEM_IRQ_10	CGEM Internal Interrupt
DSP1_IRQ_11	N/A	N/A	N/A	CGEM_IRQ_11	CGEM Internal Interrupt
DSP1_IRQ_12	N/A	N/A	N/A	CGEM_IRQ_12	CGEM Internal Interrupt
DSP1_IRQ_13	N/A	N/A	N/A	CGEM_IRQ_13	CGEM Internal Interrupt

Table 17-3. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_14	N/A	N/A	N/A	CGEM_IRQ_14	CGEM Internal Interrupt
DSP1_IRQ_15	N/A	N/A	N/A	CGEM_IRQ_15	CGEM Internal Interrupt
DSP1_IRQ_16	N/A	N/A	N/A	TPCC_INTG	EDMA CC global interrupt
DSP1_IRQ_17	N/A	N/A	N/A	TPCC_INT0	EDMA CC region 0 interrupt
DSP1_IRQ_18	N/A	N/A	N/A	TPCC_INT1	EDMA CC region 1 interrupt
DSP1_IRQ_19	N/A	N/A	N/A	TPCC_INT2	EDMA CC region 2 interrupt
DSP1_IRQ_20	N/A	N/A	N/A	TPCC_INT3	EDMA CC region 3 interrupt
DSP1_IRQ_21	N/A	N/A	N/A	FW0_FUNC_ERROR	Firewall 0 func access error
DSP1_IRQ_22	N/A	N/A	N/A	FW0_DEBUG_ERROR	Firewall 0 debug access error
DSP1_IRQ_23	N/A	N/A	N/A	FW1_FUNC_ERROR	Firewall 1 func access error
DSP1_IRQ_24	N/A	N/A	N/A	FW1_DEBUG_ERROR	Firewall 1 debug access error
DSP1_IRQ_25	N/A	N/A	N/A	MMU0_INT	DSP MMU0 Interrupt
DSP1_IRQ_26	N/A	N/A	N/A	MMU1_INT	DSP MMU1 Interrupt
DSP1_IRQ_27	N/A	N/A	N/A	TPCC_ERRINT	EDMA CC error interrupt
DSP1_IRQ_28	N/A	N/A	N/A	TPTC_ERRINT0	EDMA TC0 error interrupt
DSP1_IRQ_29	N/A	N/A	N/A	TPTC_ERRINT1	EDMA TC1 error interrupt
DSP1_IRQ_30	N/A	N/A	N/A	NOC_ERRINT	Interconnect error interrupt
DSP1_IRQ_31	NA	N/A	N/A	EDMA_WAKE_INT	EDMA wakeup interrupt
DSP1_IRQ_32	1	CTRL_CORE_DSP1_IRQ_32_33[8:0]	1	ELM_IRQ	Error location process completion interrupt
DSP1_IRQ_33	2	CTRL_CORE_DSP1_IRQ_32_33[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin
DSP1_IRQ_34	3	CTRL_CORE_DSP1_IRQ_34_35[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 18.4.6.14.3.
DSP1_IRQ_35	4	CTRL_CORE_DSP1_IRQ_34_35[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
DSP1_IRQ_36	5	CTRL_CORE_DSP1_IRQ_36_37[8:0]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
DSP1_IRQ_37	6	CTRL_CORE_DSP1_IRQ_36_37[24:16]	6	PRM_IRQ_MPU	PRCM interrupt to MPU

Table 17-3. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_38	7	CTRL_CORE_DSP1_IRQ_38_39[8:0]	7	DMA_SYSTEM_IRQ_0	System DMA interrupt 0
DSP1_IRQ_39	8	CTRL_CORE_DSP1_IRQ_38_39[24:16]	8	DMA_SYSTEM_IRQ_1	System DMA interrupt 1
DSP1_IRQ_40	9	CTRL_CORE_DSP1_IRQ_40_41[8:0]	9	DMA_SYSTEM_IRQ_2	System DMA interrupt 2
DSP1_IRQ_41	10	CTRL_CORE_DSP1_IRQ_40_41[24:16]	10	DMA_SYSTEM_IRQ_3	System DMA interrupt 3
DSP1_IRQ_42	11	CTRL_CORE_DSP1_IRQ_42_43[8:0]	11	L3_MAIN_IRQ_STAT_ALARM	L3_MAIN statistic collector alarm interrupt
DSP1_IRQ_43	12	CTRL_CORE_DSP1_IRQ_42_43[24:16]	12	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_44	13	CTRL_CORE_DSP1_IRQ_44_45[8:0]	13	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_45	14	CTRL_CORE_DSP1_IRQ_44_45[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_46	15	CTRL_CORE_DSP1_IRQ_46_47[8:0]	15	GPMC_IRQ	GPMC interrupt
DSP1_IRQ_47	16	CTRL_CORE_DSP1_IRQ_46_47[24:16]	16	GPU_IRQ	GPU interrupt
DSP1_IRQ_48	17	CTRL_CORE_DSP1_IRQ_48_49[8:0]	17	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_49	18	CTRL_CORE_DSP1_IRQ_48_49[24:16]	18	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_50	19	CTRL_CORE_DSP1_IRQ_50_51[8:0]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_51	20	CTRL_CORE_DSP1_IRQ_50_51[24:16]	20	DISPC_IRQ	Display controller interrupt
DSP1_IRQ_52	21	CTRL_CORE_DSP1_IRQ_52_53[8:0]	21	MAILBOX1_IRQ_USER0	Mailbox 1 user 0 interrupt
DSP1_IRQ_53	22	CTRL_CORE_DSP1_IRQ_52_53[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_54	23	CTRL_CORE_DSP1_IRQ_54_55[8:0]	23	DSP1_IRQ_MMU0	DSP1 MMU0 interrupt
DSP1_IRQ_55	24	CTRL_CORE_DSP1_IRQ_54_55[24:16]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
DSP1_IRQ_56	25	CTRL_CORE_DSP1_IRQ_56_57[8:0]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
DSP1_IRQ_57	26	CTRL_CORE_DSP1_IRQ_56_57[24:16]	26	GPIO3_IRQ_1	GPIO3 interrupt 1
DSP1_IRQ_58	27	CTRL_CORE_DSP1_IRQ_58_59[8:0]	27	GPIO4_IRQ_1	GPIO4 interrupt 1
DSP1_IRQ_59	28	CTRL_CORE_DSP1_IRQ_58_59[24:16]	28	GPIO5_IRQ_1	GPIO5 interrupt 1
DSP1_IRQ_60	29	CTRL_CORE_DSP1_IRQ_60_61[8:0]	29	GPIO6_IRQ_1	GPIO6 interrupt 1
DSP1_IRQ_61	30	CTRL_CORE_DSP1_IRQ_60_61[24:16]	30	GPIO7_IRQ_1	GPIO7 interrupt 1

Table 17-3. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_62	31	CTRL_CORE_DSP1_IRQ_62_63[8:0]	31	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_63	32	CTRL_CORE_DSP1_IRQ_62_63[24:16]	32	TIMER1_IRQ	TIMER1 interrupt
DSP1_IRQ_64	33	CTRL_CORE_DSP1_IRQ_64_65[8:0]	33	TIMER2_IRQ	TIMER2 interrupt
DSP1_IRQ_65	34	CTRL_CORE_DSP1_IRQ_64_65[24:16]	34	TIMER3_IRQ	TIMER3 interrupt
DSP1_IRQ_66	35	CTRL_CORE_DSP1_IRQ_66_67[8:0]	35	TIMER4_IRQ	TIMER4 interrupt
DSP1_IRQ_67	36	CTRL_CORE_DSP1_IRQ_66_67[24:16]	36	TIMER5_IRQ	TIMER5 interrupt
DSP1_IRQ_68	37	CTRL_CORE_DSP1_IRQ_68_69[8:0]	37	TIMER6_IRQ	TIMER6 interrupt
DSP1_IRQ_69	38	CTRL_CORE_DSP1_IRQ_68_69[24:16]	38	TIMER7_IRQ	TIMER7 interrupt
DSP1_IRQ_70	39	CTRL_CORE_DSP1_IRQ_70_71[8:0]	39	TIMER8_IRQ	TIMER8 interrupt
DSP1_IRQ_71	40	CTRL_CORE_DSP1_IRQ_70_71[24:16]	40	TIMER9_IRQ	TIMER9 interrupt
DSP1_IRQ_72	41	CTRL_CORE_DSP1_IRQ_72_73[8:0]	41	TIMER10_IRQ	TIMER10 interrupt
DSP1_IRQ_73	42	CTRL_CORE_DSP1_IRQ_72_73[24:16]	42	TIMER11_IRQ	TIMER11 interrupt
DSP1_IRQ_74	43	CTRL_CORE_DSP1_IRQ_74_75[8:0]	43	MCSPI4_IRQ	McSPI4 interrupt
DSP1_IRQ_75	44	CTRL_CORE_DSP1_IRQ_74_75[24:16]	44	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_76	45	CTRL_CORE_DSP1_IRQ_76_77[8:0]	45	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_77	46	CTRL_CORE_DSP1_IRQ_76_77[24:16]	46	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_78	47	CTRL_CORE_DSP1_IRQ_78_79[8:0]	47	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_79	48	CTRL_CORE_DSP1_IRQ_78_79[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_80	49	CTRL_CORE_DSP1_IRQ_80_81[8:0]	49	SATA_IRQ	SATA interrupt
DSP1_IRQ_81	50	CTRL_CORE_DSP1_IRQ_80_81[24:16]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_82	51	CTRL_CORE_DSP1_IRQ_82_83[8:0]	51	I2C1_IRQ	I2C1 interrupt
DSP1_IRQ_83	52	CTRL_CORE_DSP1_IRQ_82_83[24:16]	52	I2C2_IRQ	I2C2 interrupt
DSP1_IRQ_84	53	CTRL_CORE_DSP1_IRQ_84_85[8:0]	53	HDQ1W_IRQ	HDQ1W interrupt
DSP1_IRQ_85	54	CTRL_CORE_DSP1_IRQ_84_85[24:16]	54	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-3. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_86	55	CTRL_CORE_DSP1_IRQ_86_87[8:0]	55	I2C5_IRQ	I2C5 interrupt
DSP1_IRQ_87	56	CTRL_CORE_DSP1_IRQ_86_87[24:16]	56	I2C3_IRQ	I2C3 interrupt
DSP1_IRQ_88	57	CTRL_CORE_DSP1_IRQ_88_89[8:0]	57	I2C4_IRQ	I2C4 interrupt
DSP1_IRQ_89	58	CTRL_CORE_DSP1_IRQ_88_89[24:16]	58	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_90	59	CTRL_CORE_DSP1_IRQ_90_91[8:0]	59	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_91	60	CTRL_CORE_DSP1_IRQ_90_91[24:16]	60	MCSPI1_IRQ	McSPI1 interrupt
DSP1_IRQ_92	61	CTRL_CORE_DSP1_IRQ_92_93[8:0]	61	MCSPI2_IRQ	McSPI2 interrupt
DSP1_IRQ_93	62	CTRL_CORE_DSP1_IRQ_92_93[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_94	63	CTRL_CORE_DSP1_IRQ_94_95[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_95	64	CTRL_CORE_DSP1_IRQ_94_95[24:16]	64	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_96	N/A	N/A	N/A	CGEM_IRQ_16	CGEM Internal Interrupt
DSP1_IRQ_97	N/A	N/A	N/A	CGEM_IRQ_17	CGEM Internal Interrupt
DSP1_IRQ_98	N/A	N/A	N/A	CGEM_IRQ_18	CGEM Internal Interrupt
DSP1_IRQ_99	NA	N/A	N/A	CGEM_IRQ_19	CGEM Internal Interrupt
DSP1_IRQ_100	N/A	N/A	N/A	CGEM_IRQ_20	CGEM Internal Interrupt
DSP1_IRQ_101	N/A	N/A	N/A	CGEM_IRQ_21	CGEM Internal Interrupt
DSP1_IRQ_102	N/A	N/A	N/A	CGEM_IRQ_22	CGEM Internal Interrupt
DSP1_IRQ_103	N/A	N/A	N/A	CGEM_IRQ_23	CGEM Internal Interrupt
DSP1_IRQ_104	N/A	N/A	N/A	CGEM_IRQ_24	CGEM Internal Interrupt
DSP1_IRQ_105	NA	N/A	N/A	CGEM_IRQ_25	CGEM Internal Interrupt
DSP1_IRQ_106	N/A	N/A	N/A	CGEM_IRQ_26	CGEM Internal Interrupt
DSP1_IRQ_107	N/A	N/A	N/A	CGEM_IRQ_27	CGEM Internal Interrupt
DSP1_IRQ_108	N/A	N/A	N/A	CGEM_IRQ_28	CGEM Internal Interrupt
DSP1_IRQ_109	N/A	N/A	N/A	CGEM_IRQ_29	CGEM Internal Interrupt

Table 17-3. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_110	N/A	N/A	N/A	CGEM_IRQ_30	CGEM Internal Interrupt
DSP1_IRQ_111	N/A	N/A	N/A	CGEM_IRQ_31	CGEM Internal Interrupt
DSP1_IRQ_112	N/A	N/A	N/A	CGEM_IRQ_32	CGEM Internal Interrupt
DSP1_IRQ_113	N/A	N/A	N/A	CGEM_IRQ_33	CGEM Internal Interrupt
DSP1_IRQ_114	N/A	N/A	N/A	CGEM_IRQ_34	CGEM Internal Interrupt
DSP1_IRQ_115	N/A	N/A	N/A	CGEM_IRQ_35	CGEM Internal Interrupt
DSP1_IRQ_116	N/A	N/A	N/A	CGEM_IRQ_36	CGEM Internal Interrupt
DSP1_IRQ_117	N/A	N/A	N/A	CGEM_IRQ_37	CGEM Internal Interrupt
DSP1_IRQ_118	N/A	N/A	N/A	CGEM_IRQ_38	CGEM Internal Interrupt
DSP1_IRQ_119	NA	N/A	N/A	CGEM_IRQ_39	CGEM Internal Interrupt
DSP1_IRQ_120	N/A	N/A	N/A	CGEM_IRQ_40	CGEM Internal Interrupt
DSP1_IRQ_121	N/A	N/A	N/A	CGEM_IRQ_41	CGEM Internal Interrupt
DSP1_IRQ_122	N/A	N/A	N/A	CGEM_IRQ_42	CGEM Internal Interrupt
DSP1_IRQ_123	N/A	N/A	N/A	CGEM_IRQ_43	CGEM Internal Interrupt
DSP1_IRQ_124	N/A	N/A	N/A	CGEM_IRQ_44	CGEM Internal Interrupt
DSP1_IRQ_125	N/A	N/A	N/A	CGEM_IRQ_45	CGEM Internal Interrupt
DSP1_IRQ_126	N/A	N/A	N/A	CGEM_IRQ_46	CGEM Internal Interrupt
DSP1_IRQ_127	N/A	N/A	N/A	CGEM_IRQ_47	CGEM Internal Interrupt

Note

The "**IRQ_CROSSBAR Default Input Index**" column of Table 17-3 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding DSP1_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_DSP1_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to DSP1_INTC inputs. For example, the DSP1_IRQ_32_33[8:0] bit field is used to configure which device interrupt would be mapped to the DSP1_IRQ_32 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to DSP1_IRQ_32 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the DSP1 subsystem. There is no IRQ_CROSSBAR dedicated to the associated DSP1_INTC input line and therefore, the user cannot change its default mapping.

17.3.3 Interrupt Requests to DSP2_INTC

Table 17-4 lists the default interrupt sources for the DSP2_INTC. In addition, interrupts DSP2_IRQ_32 through DSP2_IRQ_95 can alternatively be sourced through the DSP2's IRQ_CROSSBAR from one of the 420 multiplexed device interrupts listed in Table 17-9. The CTRL_CORE_DSP2_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 17-4. DSP2_INTC Default Interrupt Mapping

IRQ Input Line	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_0	N/A	N/A	N/A	CGEM_IRQ_0	CGEM Internal Interrupt
DSP2_IRQ_1	N/A	N/A	N/A	CGEM_IRQ_1	CGEM Internal Interrupt
DSP2_IRQ_2	N/A	N/A	N/A	CGEM_IRQ_2	CGEM Internal Interrupt
DSP2_IRQ_3	N/A	N/A	N/A	CGEM_IRQ_3	CGEM Internal Interrupt
DSP2_IRQ_4	N/A	N/A	N/A	CGEM_IRQ_4	CGEM Internal Interrupt
DSP2_IRQ_5	N/A	N/A	N/A	CGEM_IRQ_5	CGEM Internal Interrupt
DSP2_IRQ_6	N/A	N/A	N/A	CGEM_IRQ_6	CGEM Internal Interrupt
DSP2_IRQ_7	N/A	N/A	N/A	CGEM_IRQ_7	CGEM Internal Interrupt
DSP2_IRQ_8	N/A	N/A	N/A	CGEM_IRQ_8	CGEM Internal Interrupt
DSP2_IRQ_9	N/A	N/A	N/A	CGEM_IRQ_9	CGEM Internal Interrupt
DSP2_IRQ_10	N/A	N/A	N/A	CGEM_IRQ_10	CGEM Internal Interrupt
DSP2_IRQ_11	N/A	N/A	N/A	CGEM_IRQ_11	CGEM Internal Interrupt
DSP2_IRQ_12	N/A	N/A	N/A	CGEM_IRQ_12	CGEM Internal Interrupt
DSP2_IRQ_13	N/A	N/A	N/A	CGEM_IRQ_13	CGEM Internal Interrupt

Table 17-4. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_14	N/A	N/A	N/A	CGEM_IRQ_14	CGEM Internal Interrupt
DSP2_IRQ_15	N/A	N/A	N/A	CGEM_IRQ_15	CGEM Internal Interrupt
DSP2_IRQ_16	N/A	N/A	N/A	TPCC_INTG	EDMA CC global interrupt
DSP2_IRQ_17	N/A	N/A	N/A	TPCC_INT0	EDMA CC region 0 interrupt
DSP2_IRQ_18	N/A	N/A	N/A	TPCC_INT1	EDMA CC region 1 interrupt
DSP2_IRQ_19	N/A	N/A	N/A	TPCC_INT2	EDMA CC region 2 interrupt
DSP2_IRQ_20	N/A	N/A	N/A	TPCC_INT3	EDMA CC region 3 interrupt
DSP2_IRQ_21	N/A	N/A	N/A	FW0_FUNC_ERROR	Firewall 0 func access error
DSP2_IRQ_22	N/A	N/A	N/A	FW0_DEBUG_ERROR	Firewall 0 debug access error
DSP2_IRQ_23	N/A	N/A	N/A	FW1_FUNC_ERROR	Firewall 1 func access error
DSP2_IRQ_24	N/A	N/A	N/A	FW1_DEBUG_ERROR	Firewall 1 debug access error
DSP2_IRQ_25	N/A	N/A	N/A	MMU0_INT	DSP MMU0 Interrupt
DSP2_IRQ_26	N/A	N/A	N/A	MMU1_INT	DSP MMU1 Interrupt
DSP2_IRQ_27	N/A	N/A	N/A	TPCC_ERRINT	EDMA CC error interrupt
DSP2_IRQ_28	N/A	N/A	N/A	TPTC_ERRINT0	EDMA TC0 error interrupt
DSP2_IRQ_29	N/A	N/A	N/A	TPTC_ERRINT1	EDMA TC1 error interrupt
DSP2_IRQ_30	N/A	N/A	N/A	NOC_ERRINT	Interconnect error interrupt
DSP2_IRQ_31	NA	N/A	N/A	EDMA_WAKE_INT	EDMA wakeup interrupt
DSP2_IRQ_32	1	CTRL_CORE_DSP2_IRQ_32_33[8:0]	1	ELM_IRQ	Error location process completion interrupt
DSP2_IRQ_33	2	CTRL_CORE_DSP2_IRQ_32_33[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin
DSP2_IRQ_34	3	CTRL_CORE_DSP2_IRQ_34_35[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 18.4.6.14.3.
DSP2_IRQ_35	4	CTRL_CORE_DSP2_IRQ_34_35[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
DSP2_IRQ_36	5	CTRL_CORE_DSP2_IRQ_36_37[8:0]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
DSP2_IRQ_37	6	CTRL_CORE_DSP2_IRQ_36_37[24:16]	6	PRM_IRQ_MPU	PRCM interrupt to MPU

Table 17-4. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_38	7	CTRL_CORE_DSP2_IRQ_38_39[8:0]	7	DMA_SYSTEM_IRQ_0	System DMA interrupt 0
DSP2_IRQ_39	8	CTRL_CORE_DSP2_IRQ_38_39[24:16]	8	DMA_SYSTEM_IRQ_1	System DMA interrupt 1
DSP2_IRQ_40	9	CTRL_CORE_DSP2_IRQ_40_41[8:0]	9	DMA_SYSTEM_IRQ_2	System DMA interrupt 2
DSP2_IRQ_41	10	CTRL_CORE_DSP2_IRQ_40_41[24:16]	10	DMA_SYSTEM_IRQ_3	System DMA interrupt 3
DSP2_IRQ_42	11	CTRL_CORE_DSP2_IRQ_42_43[8:0]	11	L3_MAIN_IRQ_STAT_ALARM	L3_MAIN statistic collector alarm interrupt
DSP2_IRQ_43	12	CTRL_CORE_DSP2_IRQ_42_43[24:16]	12	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_44	13	CTRL_CORE_DSP2_IRQ_44_45[8:0]	13	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_45	14	CTRL_CORE_DSP2_IRQ_44_45[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_46	15	CTRL_CORE_DSP2_IRQ_46_47[8:0]	15	GPMC_IRQ	GPMC interrupt
DSP2_IRQ_47	16	CTRL_CORE_DSP2_IRQ_46_47[24:16]	16	GPU_IRQ	GPU interrupt
DSP2_IRQ_48	17	CTRL_CORE_DSP2_IRQ_48_49[8:0]	17	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_49	18	CTRL_CORE_DSP2_IRQ_48_49[24:16]	18	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_50	19	CTRL_CORE_DSP2_IRQ_50_51[8:0]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_51	20	CTRL_CORE_DSP2_IRQ_50_51[24:16]	20	DISPC_IRQ	Display controller interrupt
DSP2_IRQ_52	21	CTRL_CORE_DSP2_IRQ_52_53[8:0]	21	MAILBOX1_IRQ_USER0	Mailbox 1 user 0 interrupt
DSP2_IRQ_53	22	CTRL_CORE_DSP2_IRQ_52_53[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_54	23	CTRL_CORE_DSP2_IRQ_54_55[8:0]	23	DSP2_IRQ_MMU0	DSP1 MMU0 interrupt
DSP2_IRQ_55	24	CTRL_CORE_DSP2_IRQ_54_55[24:16]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
DSP2_IRQ_56	25	CTRL_CORE_DSP2_IRQ_56_57[8:0]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
DSP2_IRQ_57	26	CTRL_CORE_DSP2_IRQ_56_57[24:16]	26	GPIO3_IRQ_1	GPIO3 interrupt 1
DSP2_IRQ_58	27	CTRL_CORE_DSP2_IRQ_58_59[8:0]	27	GPIO4_IRQ_1	GPIO4 interrupt 1
DSP2_IRQ_59	28	CTRL_CORE_DSP2_IRQ_58_59[24:16]	28	GPIO5_IRQ_1	GPIO5 interrupt 1
DSP2_IRQ_60	29	CTRL_CORE_DSP2_IRQ_60_61[8:0]	29	GPIO6_IRQ_1	GPIO6 interrupt 1
DSP2_IRQ_61	30	CTRL_CORE_DSP2_IRQ_60_61[24:16]	30	GPIO7_IRQ_1	GPIO7 interrupt 1

Table 17-4. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_62	31	CTRL_CORE_DSP2_IRQ_62_63[8:0]	31	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_63	32	CTRL_CORE_DSP2_IRQ_62_63[24:16]	32	TIMER1_IRQ	TIMER1 interrupt
DSP2_IRQ_64	33	CTRL_CORE_DSP2_IRQ_64_65[8:0]	33	TIMER2_IRQ	TIMER2 interrupt
DSP2_IRQ_65	34	CTRL_CORE_DSP2_IRQ_64_65[24:16]	34	TIMER3_IRQ	TIMER3 interrupt
DSP2_IRQ_66	35	CTRL_CORE_DSP2_IRQ_66_67[8:0]	35	TIMER4_IRQ	TIMER4 interrupt
DSP2_IRQ_67	36	CTRL_CORE_DSP2_IRQ_66_67[24:16]	36	TIMER5_IRQ	TIMER5 interrupt
DSP2_IRQ_68	37	CTRL_CORE_DSP2_IRQ_68_69[8:0]	37	TIMER6_IRQ	TIMER6 interrupt
DSP2_IRQ_69	38	CTRL_CORE_DSP2_IRQ_68_69[24:16]	38	TIMER7_IRQ	TIMER7 interrupt
DSP2_IRQ_70	39	CTRL_CORE_DSP2_IRQ_70_71[8:0]	39	TIMER8_IRQ	TIMER8 interrupt
DSP2_IRQ_71	40	CTRL_CORE_DSP2_IRQ_70_71[24:16]	40	TIMER9_IRQ	TIMER9 interrupt
DSP2_IRQ_72	41	CTRL_CORE_DSP2_IRQ_72_73[8:0]	41	TIMER10_IRQ	TIMER10 interrupt
DSP2_IRQ_73	42	CTRL_CORE_DSP2_IRQ_72_73[24:16]	42	TIMER11_IRQ	TIMER11 interrupt
DSP2_IRQ_74	43	CTRL_CORE_DSP2_IRQ_74_75[8:0]	43	MCSPI4_IRQ	McSPI4 interrupt
DSP2_IRQ_75	44	CTRL_CORE_DSP2_IRQ_74_75[24:16]	44	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_76	45	CTRL_CORE_DSP2_IRQ_76_77[8:0]	45	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_77	46	CTRL_CORE_DSP2_IRQ_76_77[24:16]	46	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_78	47	CTRL_CORE_DSP2_IRQ_78_79[8:0]	47	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_79	48	CTRL_CORE_DSP2_IRQ_78_79[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_80	49	CTRL_CORE_DSP2_IRQ_80_81[8:0]	49	SATA_IRQ	SATA interrupt
DSP2_IRQ_81	50	CTRL_CORE_DSP2_IRQ_80_81[24:16]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_82	51	CTRL_CORE_DSP2_IRQ_82_83[8:0]	51	I2C1_IRQ	I2C1 interrupt
DSP2_IRQ_83	52	CTRL_CORE_DSP2_IRQ_82_83[24:16]	52	I2C2_IRQ	I2C2 interrupt
DSP2_IRQ_84	53	CTRL_CORE_DSP2_IRQ_84_85[8:0]	53	HDQ1W_IRQ	HDQ1W interrupt
DSP2_IRQ_85	54	CTRL_CORE_DSP2_IRQ_84_85[24:16]	54	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-4. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_86	55	CTRL_CORE_DSP2_IRQ_86_87[8:0]	55	I2C5_IRQ	I2C5 interrupt
DSP2_IRQ_87	56	CTRL_CORE_DSP2_IRQ_86_87[24:16]	56	I2C3_IRQ	I2C3 interrupt
DSP2_IRQ_88	57	CTRL_CORE_DSP2_IRQ_88_89[8:0]	57	I2C4_IRQ	I2C4 interrupt
DSP2_IRQ_89	58	CTRL_CORE_DSP2_IRQ_88_89[24:16]	58	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_90	59	CTRL_CORE_DSP2_IRQ_90_91[8:0]	59	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_91	60	CTRL_CORE_DSP2_IRQ_90_91[24:16]	60	MCSPI1_IRQ	McSPI1 interrupt
DSP2_IRQ_92	61	CTRL_CORE_DSP2_IRQ_92_93[8:0]	61	MCSPI2_IRQ	McSPI2 interrupt
DSP2_IRQ_93	62	CTRL_CORE_DSP2_IRQ_92_93[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_94	63	CTRL_CORE_DSP2_IRQ_94_95[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_95	64	CTRL_CORE_DSP2_IRQ_94_95[24:16]	64	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_96	N/A	N/A	N/A	CGEM_IRQ_16	CGEM Internal Interrupt
DSP2_IRQ_97	N/A	N/A	N/A	CGEM_IRQ_17	CGEM Internal Interrupt
DSP2_IRQ_98	N/A	N/A	N/A	CGEM_IRQ_18	CGEM Internal Interrupt
DSP2_IRQ_99	NA	N/A	N/A	CGEM_IRQ_19	CGEM Internal Interrupt
DSP2_IRQ_100	N/A	N/A	N/A	CGEM_IRQ_20	CGEM Internal Interrupt
DSP2_IRQ_101	N/A	N/A	N/A	CGEM_IRQ_21	CGEM Internal Interrupt
DSP2_IRQ_102	N/A	N/A	N/A	CGEM_IRQ_22	CGEM Internal Interrupt
DSP2_IRQ_103	N/A	N/A	N/A	CGEM_IRQ_23	CGEM Internal Interrupt
DSP2_IRQ_104	N/A	N/A	N/A	CGEM_IRQ_24	CGEM Internal Interrupt
DSP2_IRQ_105	NA	N/A	N/A	CGEM_IRQ_25	CGEM Internal Interrupt
DSP2_IRQ_106	N/A	N/A	N/A	CGEM_IRQ_26	CGEM Internal Interrupt
DSP2_IRQ_107	N/A	N/A	N/A	CGEM_IRQ_27	CGEM Internal Interrupt
DSP2_IRQ_108	N/A	N/A	N/A	CGEM_IRQ_28	CGEM Internal Interrupt
DSP2_IRQ_109	N/A	N/A	N/A	CGEM_IRQ_29	CGEM Internal Interrupt

Table 17-4. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_ 110	N/A	N/A	N/A	CGEM_IRQ_30	CGEM Internal Interrupt
DSP2_IRQ_ 111	N/A	N/A	N/A	CGEM_IRQ_31	CGEM Internal Interrupt
DSP2_IRQ_ 112	N/A	N/A	N/A	CGEM_IRQ_32	CGEM Internal Interrupt
DSP2_IRQ_ 113	N/A	N/A	N/A	CGEM_IRQ_33	CGEM Internal Interrupt
DSP2_IRQ_ 114	N/A	N/A	N/A	CGEM_IRQ_34	CGEM Internal Interrupt
DSP2_IRQ_ 115	N/A	N/A	N/A	CGEM_IRQ_35	CGEM Internal Interrupt
DSP2_IRQ_ 116	N/A	N/A	N/A	CGEM_IRQ_36	CGEM Internal Interrupt
DSP2_IRQ_ 117	N/A	N/A	N/A	CGEM_IRQ_37	CGEM Internal Interrupt
DSP2_IRQ_ 118	N/A	N/A	N/A	CGEM_IRQ_38	CGEM Internal Interrupt
DSP2_IRQ_ 119	NA	N/A	N/A	CGEM_IRQ_39	CGEM Internal Interrupt
DSP2_IRQ_ 120	N/A	N/A	N/A	CGEM_IRQ_40	CGEM Internal Interrupt
DSP2_IRQ_ 121	N/A	N/A	N/A	CGEM_IRQ_41	CGEM Internal Interrupt
DSP2_IRQ_ 122	N/A	N/A	N/A	CGEM_IRQ_42	CGEM Internal Interrupt
DSP2_IRQ_ 123	N/A	N/A	N/A	CGEM_IRQ_43	CGEM Internal Interrupt
DSP2_IRQ_ 124	N/A	N/A	N/A	CGEM_IRQ_44	CGEM Internal Interrupt
DSP2_IRQ_ 125	N/A	N/A	N/A	CGEM_IRQ_45	CGEM Internal Interrupt
DSP2_IRQ_ 126	N/A	N/A	N/A	CGEM_IRQ_46	CGEM Internal Interrupt
DSP2_IRQ_ 127	N/A	N/A	N/A	CGEM_IRQ_47	CGEM Internal Interrupt

Note

The "**IRQ_CROSSBAR Default Input Index**" column of Table 17-4 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding DSP2_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_DSP2_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to DSP2_INTC inputs. For example, the DSP2_IRQ_32_33[8:0] bit field is used to configure which device interrupt would be mapped to the DSP2_IRQ_32 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to DSP2_IRQ_32 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the DSP2 subsystem. There is no IRQ_CROSSBAR dedicated to the associated DSP2_INTC input line and therefore, the user cannot change its default mapping.

17.3.4 Interrupt Requests to IPU1_Cx_INTC

Table 17-5 lists the default interrupt sources for the IPU1_Cx_INTC. In addition, device interrupts IPU1_IRQ_23 through IPU1_IRQ_79 can alternatively be sourced through the IPU1's IRQ_CROSSBAR from one of the 420 multiplexed device interrupts listed in Table 17-9. The CTRL_CORE_IPU1_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 17-5. IPU1_Cx_INTC Default Interrupt Mapping

IRQ Input Line[(1)]	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU1_IRQ_0	N/A	N/A	N/A	Reserved	MSP initial value in exception vector table
IPU1_IRQ_1	N/A	N/A	N/A	RESET_IRQ	Reset
IPU1_IRQ_2	N/A	N/A	N/A	NMI_IRQ	External NMI input (interrupt from **nmin_dsp** device pad)
IPU1_IRQ_3	N/A	N/A	N/A	HARD_FAULT_IRQ	All fault conditions, if the fault handle is not enabled
IPU1_IRQ_4	N/A	N/A	N/A	MEM_MANAGE_FAULT_IRQ	Memory management fault; access to illegal locations
IPU1_IRQ_5	N/A	N/A	N/A	BUS_FAULT_IRQ	Bus error (on AHB intf)
IPU1_IRQ_6	N/A	N/A	N/A	USAGE_FAULT_IRQ	Program error
IPU1_IRQ_7	N/A	N/A	N/A	Reserved	Reserved
IPU1_IRQ_8	N/A	N/A	N/A	Reserved	Reserved
IPU1_IRQ_9	NA	N/A	N/A	Reserved	Reserved
IPU1_IRQ_10	N/A	N/A	N/A	Reserved	Reserved
IPU1_IRQ_11	N/A	N/A	N/A	SV_CALL_IRQ	Service system Call
IPU1_IRQ_12	N/A	N/A	N/A	DEBUG_MON_IRQ	BP, WP or external debug req.
IPU1_IRQ_13	N/A	N/A	N/A	Reserved	Reserved

Table 17-5. IPU1_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line(1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU1_IRQ_14	N/A	N/A	N/A	PEND_SV_IRQ	Pendable request for system device
IPU1_IRQ_15	N/A	N/A	N/A	SYS_TICK_TIMER_IRQ	System Tick Timer
IPU1_IRQ_16	N/A	N/A	N/A	XLATE_MMU_FAULT_IRQ	xlate_mmu_fault (from L2 MMU)
IPU1_IRQ_17	N/A	N/A	N/A	UNICACHE_MMU_IRQ	Unicache or MMU maintenance complete
IPU1_IRQ_18	N/A	N/A	N/A	CTM_TIM_EVENT1_IRQ	CTM timer event (timer #1)
IPU1_IRQ_19	N/A	N/A	N/A	HWSEM_M4_IRQ	Semaphore interrupt (1 to each core)
IPU1_IRQ_20	N/A	N/A	N/A	ICE_NEMU_IRQ	ICECrusher (1 to each core)
IPU1_IRQ_21	N/A	N/A	N/A	IPU_IMP_FAULT_IRQ	Ducati imprecise fault (from interconnect)
IPU1_IRQ_22	N/A	N/A	N/A	CTM_TIM_EVENT2_IRQ	CTM timer event (timer #2)
IPU1_IRQ_23	1	CTRL_CORE_IPU1_IRQ_23_24[8:0]	20	DISPC_IRQ	Display controller interrupt
IPU1_IRQ_24	2	CTRL_CORE_IPU1_IRQ_23_24[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_25	3	CTRL_CORE_IPU1_IRQ_25_26[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_26	4	CTRL_CORE_IPU1_IRQ_25_26[24:16]	96	HDMI_IRQ	HDMI interrupt
IPU1_IRQ_27	5	CTRL_CORE_IPU1_IRQ_27_28[8:0]	126	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_28	6	CTRL_CORE_IPU1_IRQ_27_28[24:16]	127	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_29	7	CTRL_CORE_IPU1_IRQ_29_30[8:0]	128	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_30	8	CTRL_CORE_IPU1_IRQ_29_30[24:16]	129	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_31	9	CTRL_CORE_IPU1_IRQ_31_32[8:0]	130	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_32	10	CTRL_CORE_IPU1_IRQ_31_32[24:16]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_33	11	CTRL_CORE_IPU1_IRQ_33_34[8:0]	131	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_34	12	CTRL_CORE_IPU1_IRQ_33_34[24:16]	7	DMA_SYSTEM_IRQ_0	System DMA interrupt 0
IPU1_IRQ_35	13	CTRL_CORE_IPU1_IRQ_35_36[8:0]	8	DMA_SYSTEM_IRQ_1	System DMA interrupt 1
IPU1_IRQ_36	14	CTRL_CORE_IPU1_IRQ_35_36[24:16]	9	DMA_SYSTEM_IRQ_2	System DMA interrupt 2
IPU1_IRQ_37	15	CTRL_CORE_IPU1_IRQ_37_38[8:0]	10	DMA_SYSTEM_IRQ_3	System DMA interrupt 3
IPU1_IRQ_38	16	CTRL_CORE_IPU1_IRQ_37_38[24:16]	132	IVA_IRQ_MAILBOX_2	IVA mailbox user 2 interrupt

Table 17-5. IPU1_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[(1)]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU1_IRQ_39	17	CTRL_CORE_IPU1_IRQ_39_40[8:0]	98	IVA_IRQ_SYNC_1	IVA ICONT2 sync interrupt
IPU1_IRQ_40	18	CTRL_CORE_IPU1_IRQ_39_40[24:16]	99	IVA_IRQ_SYNC_0	IVA ICONT1 sync interrupt
IPU1_IRQ_41	19	CTRL_CORE_IPU1_IRQ_41_42[8:0]	51	I2C1_IRQ	I2C1 interrupt
IPU1_IRQ_42	20	CTRL_CORE_IPU1_IRQ_41_42[24:16]	52	I2C2_IRQ	I2C2 interrupt
IPU1_IRQ_43	21	CTRL_CORE_IPU1_IRQ_43_44[8:0]	56	I2C3_IRQ	I2C3 interrupt
IPU1_IRQ_44	22	CTRL_CORE_IPU1_IRQ_43_44[24:16]	57	I2C4_IRQ	I2C4 interrupt
IPU1_IRQ_45	23	CTRL_CORE_IPU1_IRQ_45_46[8:0]	69	UART3_IRQ	UART3 interrupt
IPU1_IRQ_46	24	CTRL_CORE_IPU1_IRQ_45_46[24:16]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
IPU1_IRQ_47	25	CTRL_CORE_IPU1_IRQ_47_48[8:0]	133	PRM_IRQ_IPU1	PRCM interrupt to IPU1
IPU1_IRQ_48	26	CTRL_CORE_IPU1_IRQ_47_48[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_49	27	CTRL_CORE_IPU1_IRQ_49_50[8:0]	66	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_50	28	CTRL_CORE_IPU1_IRQ_49_50[24:16]	134	MAILBOX1_IRQ_USER2	Mailbox 1 user 2 interrupt
IPU1_IRQ_51	29	CTRL_CORE_IPU1_IRQ_51_52[8:0]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
IPU1_IRQ_52	30	CTRL_CORE_IPU1_IRQ_51_52[24:16]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
IPU1_IRQ_53	31	CTRL_CORE_IPU1_IRQ_53_54[8:0]	34	TIMER3_IRQ	TIMER3 interrupt
IPU1_IRQ_54	32	CTRL_CORE_IPU1_IRQ_53_54[24:16]	35	TIMER4_IRQ	TIMER4 interrupt
IPU1_IRQ_55	33	CTRL_CORE_IPU1_IRQ_55_56[8:0]	40	TIMER9_IRQ	TIMER9 interrupt
IPU1_IRQ_56	34	CTRL_CORE_IPU1_IRQ_55_56[24:16]	42	TIMER11_IRQ	TIMER11 interrupt
IPU1_IRQ_57	35	CTRL_CORE_IPU1_IRQ_57_58[8:0]	60	MCSPI1_IRQ	McSPI1 interrupt
IPU1_IRQ_58	36	CTRL_CORE_IPU1_IRQ_57_58[24:16]	61	MCSPI2_IRQ	McSPI2 interrupt
IPU1_IRQ_59	37	CTRL_CORE_IPU1_IRQ_59_60[8:0]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_60	38	CTRL_CORE_IPU1_IRQ_59_60[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_61	39	CTRL_CORE_IPU1_IRQ_61_62[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_62	40	CTRL_CORE_IPU1_IRQ_61_62[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_63	41	CTRL_CORE_IPU1_IRQ_63_64[8:0]	83	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-5. IPU1_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[(1)]	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU1_IRQ_64	42	CTRL_CORE_IPU1_IRQ_63_64[24:16]	108	DMM_IRQ	DMM interrupt
IPU1_IRQ_65	43	CTRL_CORE_IPU1_IRQ_65_66[8:0]	120	BB2D_IRQ	BB2D interrupt
IPU1_IRQ_66	44	CTRL_CORE_IPU1_IRQ_65_66[24:16]	78	MMC1_IRQ	MMC1 interrupt
IPU1_IRQ_67	45	CTRL_CORE_IPU1_IRQ_67_68[8:0]	81	MMC2_IRQ	MMC2 interrupt
IPU1_IRQ_68	46	CTRL_CORE_IPU1_IRQ_67_68[24:16]	89	MMC3_IRQ	MMC3 interrupt
IPU1_IRQ_69	47	CTRL_CORE_IPU1_IRQ_69_70[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_70	48	CTRL_CORE_IPU1_IRQ_69_70[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_71	49	CTRL_CORE_IPU1_IRQ_71_72[8:0]	119	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_72	50	CTRL_CORE_IPU1_IRQ_71_72[24:16]	118	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_73	51	CTRL_CORE_IPU1_IRQ_73_74[8:0]	72	USB1_IRQ_INTR1	USB1 interrupt 1
IPU1_IRQ_74	52	CTRL_CORE_IPU1_IRQ_73_74[24:16]	73	USB2_IRQ_INTR0	USB2 interrupt 0
IPU1_IRQ_75	53	CTRL_CORE_IPU1_IRQ_75_76[8:0]	117	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_76	54	CTRL_CORE_IPU1_IRQ_75_76[24:16]	87	USB2_IRQ_INTR1	USB2 interrupt 1
IPU1_IRQ_77	55	CTRL_CORE_IPU1_IRQ_77_78[8:0]	88	USB3_IRQ_INTR0	USB3 interrupt 0
IPU1_IRQ_78	56	CTRL_CORE_IPU1_IRQ_77_78[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU1_IRQ_79	57	CTRL_CORE_IPU1_IRQ_79_80[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source

(1) This column shows the number of the corresponding exception type.

Note

Exceptions/interrupts IPU1_IRQ_[15:0] are all internal to the Cortex-M4 core.

Exceptions/interrupts IPU1_IRQ_[79:16] are all external to the Cortex-M4 core – that is, the first Cortex-M4 external interrupt is mapped to IPU1_IRQ_16 (exception #16), and the last (sixty-fourth) Cortex-M4 external interrupt is mapped to IPU1_IRQ_79 (exception #79).

For more information about Cortex-M4 exception types, refer to Arm *Cortex®-M4 Devices Generic User Guide* (available at http://infocenter.arm.com/help/index.jsp).

Note

The "**IRQ_CROSSBAR Default Input Index**" column of Table 17-5 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding IPU1_Cx_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_IPU1_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to IPU1_Cx_INTC inputs. For example, the IPU1_IRQ_23_24[8:0] bit field is used to configure which device interrupt would be mapped to the IPU1_IRQ_23 line. The reset value of this bit field is 0x14, meaning that DISPC_IRQ would be mapped to IPU1_IRQ_23 by default because it is connected to the IRQ_CROSSBAR_20 input.

'N/A' in this column means that the corresponding interrupt is internal to the IPU1 subsystem. There is no IRQ_CROSSBAR dedicated to the associated IPU1_Cx_INTC input line and therefore, the user cannot change its default mapping.

The CTRL_CORE_IPU1_IRQ_y_z registers control the IRQ_CROSSBAR settings for both NVICs in the IPU1 subsystem. That is, it is not possible to map different interrupts to the same interrupt input of the NVICs in IPU1.

17.3.5 Interrupt Requests to IPU2_Cx_INTC

Table 17-6 lists the default interrupt sources for the IPU2_Cx_INTC. In addition, device interrupts IPU2_IRQ_23 through IPU2_IRQ_79 can alternatively be sourced through the IPU2's IRQ_CROSSBAR from one of the 420 multiplexed device interrupts listed in Table 17-9. The CTRL_CORE_IPU2_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 17-6. IPU2_Cx_INTC Default Interrupt Mapping

IRQ Input Line[(1)]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU2_IRQ_ 0	N/A	N/A	N/A	Reserved	MSP initial value in exception vector table
IPU2_IRQ_ 1	N/A	N/A	N/A	RESET_IRQ	Reset
IPU2_IRQ_ 2	N/A	N/A	N/A	NMI_IRQ	External NMI input (interrupt from **nmin_dsp** device pad)
IPU2_IRQ_ 3	N/A	N/A	N/A	HARD_FAULT_IRQ	All fault conditions, if the fault handle is not enabled
IPU2_IRQ_ 4	N/A	N/A	N/A	MEM_MANAGE_FAULT _IRQ	Memory management fault; access to illegal locations
IPU2_IRQ_ 5	N/A	N/A	N/A	BUS_FAULT_IRQ	Bus error (on AHB intf)
IPU2_IRQ_ 6	N/A	N/A	N/A	USAGE_FAULT_IRQ	Program error
IPU2_IRQ_ 7	N/A	N/A	N/A	Reserved	Reserved
IPU2_IRQ_ 8	N/A	N/A	N/A	Reserved	Reserved
IPU2_IRQ_ 9	NA	N/A	N/A	Reserved	Reserved
IPU2_IRQ_ 10	N/A	N/A	N/A	Reserved	Reserved
IPU2_IRQ_ 11	N/A	N/A	N/A	SV_CALL_IRQ	Service system Call
IPU2_IRQ_ 12	N/A	N/A	N/A	DEBUG_MON_IRQ	BP, WP or external debug req.

Table 17-6. IPU2_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[(1)]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU2_IRQ_ 13	N/A	N/A	N/A	Reserved	Reserved
IPU2_IRQ_ 14	N/A	N/A	N/A	PEND_SV_IRQ	Pendable request for system device
IPU2_IRQ_ 15	N/A	N/A	N/A	SYS_TICK_TIMER_IRQ	System Tick Timer
IPU2_IRQ_ 16	N/A	N/A	N/A	XLATE_MMU_FAULT_IRQ	xlate_mmu_fault (from L2 MMU)
IPU2_IRQ_ 17	N/A	N/A	N/A	UNICACHE_MMU_IRQ	Unicache or MMU maintenance complete
IPU2_IRQ_ 18	N/A	N/A	N/A	CTM_TIM_EVENT1_IRQ	CTM timer event (timer #1)
IPU2_IRQ_ 19	N/A	N/A	N/A	HWSEM_M4_IRQ	Semaphore interrupt (1 to each core)
IPU2_IRQ_ 20	N/A	N/A	N/A	ICE_NEMU_IRQ	ICECrusher (1 to each core)
IPU2_IRQ_ 21	N/A	N/A	N/A	IPU_IMP_FAULT_IRQ	Ducati imprecise fault (from interconnect)
IPU2_IRQ_ 22	N/A	N/A	N/A	CTM_TIM_EVENT2_IRQ	CTM timer event (timer #2)
IPU2_IRQ_ 23	1	CTRL_CORE_IPU2_IRQ_23_24[8:0]	20	DISPC_IRQ	Display controller interrupt
IPU2_IRQ_ 24	2	CTRL_CORE_IPU2_IRQ_23_24[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 25	3	CTRL_CORE_IPU2_IRQ_25_26[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 26	4	CTRL_CORE_IPU2_IRQ_25_26[24:16]	96	HDMI_IRQ	HDMI interrupt
IPU2_IRQ_ 27	5	CTRL_CORE_IPU2_IRQ_27_28[8:0]	126	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 28	6	CTRL_CORE_IPU2_IRQ_27_28[24:16]	127	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 29	7	CTRL_CORE_IPU2_IRQ_29_30[8:0]	128	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 30	8	CTRL_CORE_IPU2_IRQ_29_30[24:16]	129	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 31	9	CTRL_CORE_IPU2_IRQ_31_32[8:0]	130	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 32	10	CTRL_CORE_IPU2_IRQ_31_32[24:16]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 33	11	CTRL_CORE_IPU2_IRQ_33_34[8:0]	131	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_ 34	12	CTRL_CORE_IPU2_IRQ_33_34[24:16]	7	DMA_SYSTEM_IRQ_0	System DMA interrupt 0
IPU2_IRQ_ 35	13	CTRL_CORE_IPU2_IRQ_35_36[8:0]	8	DMA_SYSTEM_IRQ_1	System DMA interrupt 1
IPU2_IRQ_ 36	14	CTRL_CORE_IPU2_IRQ_35_36[24:16]	9	DMA_SYSTEM_IRQ_2	System DMA interrupt 2
IPU2_IRQ_ 37	15	CTRL_CORE_IPU2_IRQ_37_38[8:0]	10	DMA_SYSTEM_IRQ_3	System DMA interrupt 3

Table 17-6. IPU2_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line(1)	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU2_IRQ_38	16	CTRL_CORE_IPU2_IRQ_37_38[24:16]	132	IVA_IRQ_MAILBOX_2	IVA mailbox user 2 interrupt
IPU2_IRQ_39	17	CTRL_CORE_IPU2_IRQ_39_40[8:0]	98	IVA_IRQ_SYNC_1	IVA ICONT2 sync interrupt
IPU2_IRQ_40	18	CTRL_CORE_IPU2_IRQ_39_40[24:16]	99	IVA_IRQ_SYNC_0	IVA ICONT1 sync interrupt
IPU2_IRQ_41	19	CTRL_CORE_IPU2_IRQ_41_42[8:0]	51	I2C1_IRQ	I2C1 interrupt
IPU2_IRQ_42	20	CTRL_CORE_IPU2_IRQ_41_42[24:16]	52	I2C2_IRQ	I2C2 interrupt
IPU2_IRQ_43	21	CTRL_CORE_IPU2_IRQ_43_44[8:0]	56	I2C3_IRQ	I2C3 interrupt
IPU2_IRQ_44	22	CTRL_CORE_IPU2_IRQ_43_44[24:16]	57	I2C4_IRQ	I2C4 interrupt
IPU2_IRQ_45	23	CTRL_CORE_IPU2_IRQ_45_46[8:0]	69	UART3_IRQ	UART3 interrupt
IPU2_IRQ_46	24	CTRL_CORE_IPU2_IRQ_45_46[24:16]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
IPU2_IRQ_47	25	CTRL_CORE_IPU2_IRQ_47_48[8:0]	133	PRM_IRQ_IPU1	PRCM interrupt to IPU2
IPU2_IRQ_48	26	CTRL_CORE_IPU2_IRQ_47_48[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_49	27	CTRL_CORE_IPU2_IRQ_49_50[8:0]	66	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_50	28	CTRL_CORE_IPU2_IRQ_49_50[24:16]	134	MAILBOX1_IRQ_USER2	Mailbox 1 user 2 interrupt
IPU2_IRQ_51	29	CTRL_CORE_IPU2_IRQ_51_52[8:0]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
IPU2_IRQ_52	30	CTRL_CORE_IPU2_IRQ_51_52[24:16]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
IPU2_IRQ_53	31	CTRL_CORE_IPU2_IRQ_53_54[8:0]	34	TIMER3_IRQ	TIMER3 interrupt
IPU2_IRQ_54	32	CTRL_CORE_IPU2_IRQ_53_54[24:16]	35	TIMER4_IRQ	TIMER4 interrupt
IPU2_IRQ_55	33	CTRL_CORE_IPU2_IRQ_55_56[8:0]	40	TIMER9_IRQ	TIMER9 interrupt
IPU2_IRQ_56	34	CTRL_CORE_IPU2_IRQ_55_56[24:16]	42	TIMER11_IRQ	TIMER11 interrupt
IPU2_IRQ_57	35	CTRL_CORE_IPU2_IRQ_57_58[8:0]	60	MCSPI1_IRQ	McSPI1 interrupt
IPU2_IRQ_58	36	CTRL_CORE_IPU2_IRQ_57_58[24:16]	61	MCSPI2_IRQ	McSPI2 interrupt
IPU2_IRQ_59	37	CTRL_CORE_IPU2_IRQ_59_60[8:0]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_60	38	CTRL_CORE_IPU2_IRQ_59_60[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_61	39	CTRL_CORE_IPU2_IRQ_61_62[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_62	40	CTRL_CORE_IPU2_IRQ_61_62[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-6. IPU2_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[(1)]	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU2_IRQ_63	41	CTRL_CORE_IPU2_IRQ_63_64[8:0]	83	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_64	42	CTRL_CORE_IPU2_IRQ_63_64[24:16]	108	DMM_IRQ	DMM interrupt
IPU2_IRQ_65	43	CTRL_CORE_IPU2_IRQ_65_66[8:0]	120	BB2D_IRQ	BB2D interrupt
IPU2_IRQ_66	44	CTRL_CORE_IPU2_IRQ_65_66[24:16]	78	MMC1_IRQ	MMC1 interrupt
IPU2_IRQ_67	45	CTRL_CORE_IPU2_IRQ_67_68[8:0]	81	MMC2_IRQ	MMC2 interrupt
IPU2_IRQ_68	46	CTRL_CORE_IPU2_IRQ_67_68[24:16]	89	MMC3_IRQ	MMC3 interrupt
IPU2_IRQ_69	47	CTRL_CORE_IPU2_IRQ_69_70[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_70	48	CTRL_CORE_IPU2_IRQ_69_70[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_71	49	CTRL_CORE_IPU2_IRQ_71_72[8:0]	119	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_72	50	CTRL_CORE_IPU2_IRQ_71_72[24:16]	118	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_73	51	CTRL_CORE_IPU2_IRQ_73_74[8:0]	72	USB1_IRQ_INTR1	USB1 interrupt 1
IPU2_IRQ_74	52	CTRL_CORE_IPU2_IRQ_73_74[24:16]	73	USB2_IRQ_INTR0	USB2 interrupt 0
IPU2_IRQ_75	53	CTRL_CORE_IPU2_IRQ_75_76[8:0]	117	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_76	54	CTRL_CORE_IPU2_IRQ_75_76[24:16]	87	USB2_IRQ_INTR1	USB2 interrupt 1
IPU2_IRQ_77	55	CTRL_CORE_IPU2_IRQ_77_78[8:0]	88	USB3_IRQ_INTR0	USB3 interrupt 0
IPU2_IRQ_78	56	CTRL_CORE_IPU2_IRQ_77_78[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU2_IRQ_79	57	CTRL_CORE_IPU2_IRQ_79_80[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source

(1) This column shows the number of the corresponding exception type.

Note

Exceptions/interrupts IPU2_IRQ_[15:0] are all internal to the Cortex-M4 core.

Exceptions/interrupts IPU2_IRQ_[79:16] are all external to the Cortex-M4 core – that is, the first Cortex-M4 external interrupt is mapped to IPU2_IRQ_16 (exception #16), and the last (sixty-fourth) Cortex-M4 external interrupt is mapped to IPU2_IRQ_79 (exception #79).

For more information about Cortex-M4 exception types, refer to Arm *Cortex®-M4 Devices Generic User Guide* (available at http://infocenter.arm.com/help/index.jsp).

Note

The "**IRQ_CROSSBAR Default Input Index**" column of Table 17-6 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding IPU2_Cx_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_IPU2_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to IPU2_Cx_INTC inputs. For example, the IPU2_IRQ_23_24[8:0] bit field is used to configure which device interrupt would be mapped to the IPU2_IRQ_23 line. The reset value of this bit field is 0x14, meaning that DISPC_IRQ would be mapped to IPU2_IRQ_23 by default because it is connected to the IRQ_CROSSBAR_20 input.

'N/A' in this column means that the corresponding interrupt is internal to the IPU2 subsystem. There is no IRQ_CROSSBAR dedicated to the associated IPU2_Cx_INTC input line and therefore, the user cannot change its default mapping.

The CTRL_CORE_IPU2_IRQ_y_z registers control the IRQ_CROSSBAR settings for both NVICs in the IPU2 subsystem. That is, it is not possible to map different interrupts to the same interrupt input of the NVICs in IPU2.

17.3.6 Interrupt Requests to EVE1_INTC1

Table 17-7 lists the default interrupt sources for the EVE1_INTC1. In addition, device interrupts EVE1_IRQ_0 through EVE1_IRQ_7 can alternatively be sourced through the EVE1's IRQ_CROSSBAR from one of the 420 multiplexed device interrupts listed in Table 17-9. The CTRL_CORE_EVE1_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 17-7. EVE1_INTC1 Default Interrupt Mapping

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
EVE1_IRQ_0	1	CTRL_CORE_EVE1_IRQ_0_1[8:0]	1	ELM_IRQ	Error location process completion interrupt
EVE1_IRQ_1	2	CTRL_CORE_EVE1_IRQ_0_1[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin
EVE1_IRQ_2	3	CTRL_CORE_EVE1_IRQ_2_3[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 18.4.6.14.3.
EVE1_IRQ_3	4	CTRL_CORE_EVE1_IRQ_2_3[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
EVE1_IRQ_4	5	CTRL_CORE_EVE1_IRQ_4_5[8:0]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
EVE1_IRQ_5	6	CTRL_CORE_EVE1_IRQ_4_5[24:16]	6	PRM_IRQ_MPU	PRCM interrupt to MPU
EVE1_IRQ_6	7	CTRL_CORE_EVE1_IRQ_6_7[8:0]	7	DMA_SYSTEM_IRQ_0	System DMA interrupt 0
EVE1_IRQ_7	8	CTRL_CORE_EVE1_IRQ_6_7[24:16]	8	DMA_SYSTEM_IRQ_1	System DMA interrupt 1
EVE1_IRQ_8	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_9	N/A	N/A	N/A	EVE2_GP8	EVE2 GP8 interrupt
EVE1_IRQ_10	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_11	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-7. EVE1_INTC1 Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
EVE1_IRQ_12	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_13	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_14	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_15	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_16	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_17	N/A	N/A	N/A	EVE2_MBX2_INT1	EVE2 Mailbox 2 Interrupt 1
EVE1_IRQ_18	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_19	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_20	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_21	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_22	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_23	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_24	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_25	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_26	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_27	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_28	N/A	N/A	N/A	EVE1_MBX2_INT0	EVE1 Mailbox 2 Interrupt 0
EVE1_IRQ_29	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_30	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE1_IRQ_31	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source

Note

The "**IRQ_CROSSBAR Default Input Index**" column of Table 17-7 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding EVE1_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_EVE1_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to EVE1_INTC inputs. For example, the EVE1_IRQ_0_1[8:0] bit field is used to configure which device interrupt would be mapped to the EVE1_IRQ_0 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to EVE1_IRQ_0 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the EVE1 subsystem. There is no IRQ_CROSSBAR dedicated to the associated EVE1_INTC input line and therefore, the user cannot change its default mapping.

17.3.7 Interrupt Requests to EVE2_INTC1

Table 17-8 lists the default interrupt sources for the EVE2_INTC1. In addition, device interrupts EVE2_IRQ_0 through EVE2_IRQ_7 can alternatively be sourced through the EVE2's IRQ_CROSSBAR from one of the 420 multiplexed device interrupts listed in Table 17-9. The CTRL_CORE_EVE2_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 17-8. EVE2_INTC1 Default Interrupt Mapping

IRQ Input Line	IRQ_CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
EVE2_IRQ_0	1	CTRL_CORE_EVE2_IRQ_0_1[8:0]	1	ELM_IRQ	Error location process completion interrupt
EVE2_IRQ_1	2	CTRL_CORE_EVE2_IRQ_0_1[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin
EVE2_IRQ_2	3	CTRL_CORE_EVE2_IRQ_2_3[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 18.4.6.14.3.
EVE2_IRQ_3	4	CTRL_CORE_EVE2_IRQ_2_3[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
EVE2_IRQ_4	5	CTRL_CORE_EVE2_IRQ_4_5[8:0]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
EVE2_IRQ_5	6	CTRL_CORE_EVE2_IRQ_4_5[24:16]	6	PRM_IRQ_MPU	PRCM interrupt to MPU
EVE2_IRQ_6	7	CTRL_CORE_EVE2_IRQ_6_7[8:0]	7	DMA_SYSTEM_IRQ_0	System DMA interrupt 0
EVE2_IRQ_7	8	CTRL_CORE_EVE2_IRQ_6_7[24:16]	8	DMA_SYSTEM_IRQ_1	System DMA interrupt 1
EVE2_IRQ_8	N/A	N/A	N/A	EVE1_GP1	EVE1 GP1 interrupt
EVE2_IRQ_9	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_10	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_11	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_12	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 17-8. EVE2_INTC1 Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
EVE2_IRQ_13	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_14	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_15	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_16	N/A	N/A	N/A	EVE1_MBX2_INT1	EVE1 Mailbox 2 Interrupt 1
EVE2_IRQ_17	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_18	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_19	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_20	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_21	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_22	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_23	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_24	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_25	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_26	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_27	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_28	N/A	N/A	N/A	EVE2_MBX2_INT0	EVE2 Mailbox 2 Interrupt 0
EVE2_IRQ_29	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_30	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE2_IRQ_31	N/A	N/A	N/A	Reserved	Reserved by default but can be remapped to a valid interrupt source

Note

The "**IRQ_CROSSBAR Default Input Index**" column of Table 17-8 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding EVE2_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_EVE2_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to EVE2_INTC inputs. For example, the EVE2_IRQ_0_1[8:0] bit field is used to configure which device interrupt would be mapped to the EVE2_IRQ_0 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to EVE2_IRQ_0 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the EVE2 subsystem. There is no IRQ_CROSSBAR dedicated to the associated EVE2_INTC input line and therefore, the user cannot change its default mapping.

17.3.8 Mapping of Device Interrupts to IRQ_CROSSBAR Inputs

Table 17-9 shows the individual connection between all module IRQs and all IRQ_CROSSBAR inputs.

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_0	Reserved	Reserved	Reserved
IRQ_CROSSBAR_1	ELM_IRQ	ELM	Error location process completion interrupt
IRQ_CROSSBAR_2	EXT_SYS_IRQ_1	External system	External interrupt (active low) via sys_nirq1 pin
IRQ_CROSSBAR_3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	CTRL_MODULE_CORE	Combined firewall error interrupt. For more information, see *Firewall Error Status Registers*
IRQ_CROSSBAR_4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN	L3_MAIN debug error
IRQ_CROSSBAR_5	L3_MAIN_IRQ_APP_ERR	L3_MAIN	L3_MAIN application or non-attributable error
IRQ_CROSSBAR_6	PRM_IRQ_MPU	PRM	PRCM interrupt to MPU
IRQ_CROSSBAR_7	DMA_SYSTEM_IRQ_0	DMA_SYSTEM	System DMA interrupt 0
IRQ_CROSSBAR_8	DMA_SYSTEM_IRQ_1	DMA_SYSTEM	System DMA interrupt 1
IRQ_CROSSBAR_9	DMA_SYSTEM_IRQ_2	DMA_SYSTEM	System DMA interrupt 2
IRQ_CROSSBAR_10	DMA_SYSTEM_IRQ_3	DMA_SYSTEM	System DMA interrupt 3
IRQ_CROSSBAR_11	L3_MAIN_IRQ_STAT_ALARM	L3_MAIN	L3_MAIN statistic collector alarm interrupt
IRQ_CROSSBAR_12	Reserved	Reserved	Reserved
IRQ_CROSSBAR_13	Reserved	Reserved	Reserved
IRQ_CROSSBAR_14	Reserved	Reserved	Reserved
IRQ_CROSSBAR_15	GPMC_IRQ	GPMC	GPMC interrupt
IRQ_CROSSBAR_16	GPU_IRQ	GPU	GPU interrupt
IRQ_CROSSBAR_17	Reserved	Reserved	Reserved
IRQ_CROSSBAR_18	Reserved	Reserved	Reserved
IRQ_CROSSBAR_19	Reserved	Reserved	Reserved
IRQ_CROSSBAR_20	DISPC_IRQ	DISPC	Display controller interrupt
IRQ_CROSSBAR_21	MAILBOX1_IRQ_USER0	MAILBOX1	Mailbox 1 user 0 interrupt
IRQ_CROSSBAR_22	Reserved	Reserved	Reserved
IRQ_CROSSBAR_23	DSP1_IRQ_MMU0	DSP1	DSP1 MMU0 interrupt
IRQ_CROSSBAR_24	GPIO1_IRQ_1	GPIO1	GPIO1 interrupt 1
IRQ_CROSSBAR_25	GPIO2_IRQ_1	GPIO2	GPIO2 interrupt 1
IRQ_CROSSBAR_26	GPIO3_IRQ_1	GPIO3	GPIO3 interrupt 1
IRQ_CROSSBAR_27	GPIO4_IRQ_1	GPIO4	GPIO4 interrupt 1

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_28	GPIO5_IRQ_1	GPIO5	GPIO5 interrupt 1
IRQ_CROSSBAR_29	GPIO6_IRQ_1	GPIO6	GPIO6 interrupt 1
IRQ_CROSSBAR_30	GPIO7_IRQ_1	GPIO7	GPIO7 interrupt 1
IRQ_CROSSBAR_31	Reserved	Reserved	Reserved
IRQ_CROSSBAR_32	TIMER1_IRQ	TIMER1	TIMER1 interrupt
IRQ_CROSSBAR_33	TIMER2_IRQ	TIMER2	TIMER2 interrupt
IRQ_CROSSBAR_34	TIMER3_IRQ	TIMER3	TIMER3 interrupt
IRQ_CROSSBAR_35	TIMER4_IRQ	TIMER4	TIMER4 interrupt
IRQ_CROSSBAR_36	TIMER5_IRQ	TIMER5	TIMER5 interrupt
IRQ_CROSSBAR_37	TIMER6_IRQ	TIMER6	TIMER6 interrupt
IRQ_CROSSBAR_38	TIMER7_IRQ	TIMER7	TIMER7 interrupt
IRQ_CROSSBAR_39	TIMER8_IRQ	TIMER8	TIMER8 interrupt
IRQ_CROSSBAR_40	TIMER9_IRQ	TIMER9	TIMER9 interrupt
IRQ_CROSSBAR_41	TIMER10_IRQ	TIMER10	TIMER10 interrupt
IRQ_CROSSBAR_42	TIMER11_IRQ	TIMER11	TIMER11 interrupt
IRQ_CROSSBAR_43	MCSPI4_IRQ	MCSPI4	McSPI4 interrupt
IRQ_CROSSBAR_44	Reserved	Reserved	Reserved
IRQ_CROSSBAR_45	Reserved	Reserved	Reserved
IRQ_CROSSBAR_46	Reserved	Reserved	Reserved
IRQ_CROSSBAR_47	Reserved	Reserved	Reserved
IRQ_CROSSBAR_48	Reserved	Reserved	Reserved
IRQ_CROSSBAR_49	SATA_IRQ	SATA	SATA interrupt
IRQ_CROSSBAR_50	Reserved	Reserved	Reserved
IRQ_CROSSBAR_51	I2C1_IRQ	I2C1	I2C1 interrupt
IRQ_CROSSBAR_52	I2C2_IRQ	I2C2	I2C2 interrupt
IRQ_CROSSBAR_53	HDQ1W_IRQ	HDQ1W	HDQ1W interrupt
IRQ_CROSSBAR_54	Reserved	Reserved	Reserved
IRQ_CROSSBAR_55	I2C5_IRQ	I2C5	I2C5 interrupt
IRQ_CROSSBAR_56	I2C3_IRQ	I2C3	I2C3 interrupt
IRQ_CROSSBAR_57	I2C4_IRQ	I2C4	I2C4 interrupt
IRQ_CROSSBAR_58	Reserved	Reserved	Reserved
IRQ_CROSSBAR_59	Reserved	Reserved	Reserved
IRQ_CROSSBAR_60	MCSPI1_IRQ	MCSPI1	McSPI1 interrupt
IRQ_CROSSBAR_61	MCSPI2_IRQ	MCSPI2	McSPI2 interrupt
IRQ_CROSSBAR_62	Reserved	Reserved	Reserved
IRQ_CROSSBAR_63	Reserved	Reserved	Reserved
IRQ_CROSSBAR_64	Reserved	Reserved	Reserved
IRQ_CROSSBAR_65	UART4_IRQ	UART4	UART4 interrupt
IRQ_CROSSBAR_66	Reserved	Reserved	Reserved
IRQ_CROSSBAR_67	UART1_IRQ	UART1	UART1 interrupt
IRQ_CROSSBAR_68	UART2_IRQ	UART2	UART2 interrupt
IRQ_CROSSBAR_69	UART3_IRQ	UART3	UART3 interrupt
IRQ_CROSSBAR_70	PBIAS_IRQ	MMC1 PBIAS Cell	MMC1 PBIAS interrupt (controlled via device Control Module)
IRQ_CROSSBAR_71	USB1_IRQ_INTR0	USB1	USB1 interrupt 0

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_72	USB1_IRQ_INTR1	USB1	USB1 interrupt 1
IRQ_CROSSBAR_73	USB2_IRQ_INTR0	USB2	USB2 interrupt 0
IRQ_CROSSBAR_74	Reserved	Reserved	Reserved
IRQ_CROSSBAR_75	WD_TIMER2_IRQ	WD_TIMER2	WD_TIMER2 interrupt
IRQ_CROSSBAR_76	Reserved	Reserved	Reserved
IRQ_CROSSBAR_77	Reserved	Reserved	Reserved
IRQ_CROSSBAR_78	MMC1_IRQ	MMC1	MMC1 interrupt
IRQ_CROSSBAR_79	Reserved	Reserved	Reserved
IRQ_CROSSBAR_80	Reserved	Reserved	Reserved
IRQ_CROSSBAR_81	MMC2_IRQ	MMC2	MMC2 interrupt
IRQ_CROSSBAR_82	Reserved	Reserved	Reserved
IRQ_CROSSBAR_83	Reserved	Reserved	Reserved
IRQ_CROSSBAR_84	Reserved	Reserved	Reserved
IRQ_CROSSBAR_85	DEBUGSS_IRQ_CT_UART	DEBUGSS	CT_UART interrupt generated when data ready on RX or when TX empty
IRQ_CROSSBAR_86	MCSPI3_IRQ	MCSPI3	McSPI3 interrupt
IRQ_CROSSBAR_87	USB2_IRQ_INTR1	USB2	USB2 interrupt 1
IRQ_CROSSBAR_88	USB3_IRQ_INTR0	USB3	USB3 interrupt 0
IRQ_CROSSBAR_89	MMC3_IRQ	MMC3	MMC3 interrupt
IRQ_CROSSBAR_90	TIMER12_IRQ	TIMER12	TIMER12 interrupt
IRQ_CROSSBAR_91	MMC4_IRQ	MMC4	MMC4 interrupt
IRQ_CROSSBAR_92	Reserved	Reserved	Reserved
IRQ_CROSSBAR_93	Reserved	Reserved	Reserved
IRQ_CROSSBAR_94	Reserved	Reserved	Reserved
IRQ_CROSSBAR_95	Reserved	Reserved	Reserved
IRQ_CROSSBAR_96	HDMI_IRQ	HDMI	HDMI interrupt
IRQ_CROSSBAR_97	Reserved	Reserved	Reserved
IRQ_CROSSBAR_98	IVA_IRQ_SYNC_1	IVA	IVA ICONT2 sync interrupt
IRQ_CROSSBAR_99	IVA_IRQ_SYNC_0	IVA	IVA ICONT1 sync interrupt
IRQ_CROSSBAR_100	UART5_IRQ	UART5	UART5 interrupt
IRQ_CROSSBAR_101	UART6_IRQ	UART6	UART6 interrupt
IRQ_CROSSBAR_102	IVA_IRQ_MAILBOX_0	IVA	IVA mailbox user 0 interrupt
IRQ_CROSSBAR_103	McASP1_IRQ_AREVT	McASP1	McASP1 receive interrupt
IRQ_CROSSBAR_104	McASP1_IRQ_AXEVT	McASP1	McASP1 transmit interrupt
IRQ_CROSSBAR_105	EMIF1_IRQ	EMIF1	EMIF1 interrupt
IRQ_CROSSBAR_106	EMIF2_IRQ	EMIF2	EMIF2 interrupt
IRQ_CROSSBAR_107	Reserved	Reserved	Reserved
IRQ_CROSSBAR_108	DMM_IRQ	DMM	DMM interrupt
IRQ_CROSSBAR_109	Reserved	Reserved	Reserved
IRQ_CROSSBAR_110	Reserved	Reserved	Reserved
IRQ_CROSSBAR_111	Reserved	Reserved	Reserved
IRQ_CROSSBAR_112	Reserved	Reserved	Reserved
IRQ_CROSSBAR_113	Reserved	Reserved	Reserved
IRQ_CROSSBAR_114	EXT_SYS_IRQ_2	External system	External interrupt (active low) via sys_nirq2 pin
IRQ_CROSSBAR_115	KBD_IRQ	KBD	Keyboard controller interrupt

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_116	GPIO8_IRQ_1	GPIO8	GPIO8 interrupt 1
IRQ_CROSSBAR_117	Reserved	Reserved	Reserved
IRQ_CROSSBAR_118	Reserved	Reserved	Reserved
IRQ_CROSSBAR_119	Reserved	Reserved	Reserved
IRQ_CROSSBAR_120	BB2D_IRQ	BB2D	BB2D interrupt
IRQ_CROSSBAR_121	CTRL_MODULE_CORE_IRQ_THERMAL_ALERT	CTRL_MODULE	CTRL_MODULE thermal alert interrupt
IRQ_CROSSBAR_[122:131]	Reserved	Reserved	Reserved
IRQ_CROSSBAR_132	IVA_IRQ_MAILBOX_2	IVA	IVA mailbox user 2 interrupt
IRQ_CROSSBAR_133	PRM_IRQ_IPU1	PRM	PRCM interrupt to IPU1
IRQ_CROSSBAR_134	MAILBOX1_IRQ_USER2	MAILBOX1	Mailbox 1 user 2 interrupt
IRQ_CROSSBAR_135	MAILBOX1_IRQ_USER1	MAILBOX1	Mailbox 1 user 1 interrupt
IRQ_CROSSBAR_136	IVA_IRQ_MAILBOX_1	IVA	IVA mailbox user 1 interrupt
IRQ_CROSSBAR_137	PRM_IRQ_DSP1	PRM	PRCM interrupt to DSP1
IRQ_CROSSBAR_138	GPIO1_IRQ_2	GPIO1	GPIO1 interrupt 2
IRQ_CROSSBAR_139	GPIO2_IRQ_2	GPIO2	GPIO2 interrupt 2
IRQ_CROSSBAR_140	GPIO3_IRQ_2	GPIO3	GPIO3 interrupt 2
IRQ_CROSSBAR_141	GPIO4_IRQ_2	GPIO4	GPIO4 interrupt 2
IRQ_CROSSBAR_142	GPIO5_IRQ_2	GPIO5	GPIO5 interrupt 2
IRQ_CROSSBAR_143	GPIO6_IRQ_2	GPIO6	GPIO6 interrupt 2
IRQ_CROSSBAR_144	Reserved	Reserved	Reserved
IRQ_CROSSBAR_145	DSP1_IRQ_MMU1	DSP1	DSP1 MMU1 interrupt
IRQ_CROSSBAR_146	DSP2_IRQ_MMU0	DSP2	DSP2 MMU0 interrupt
IRQ_CROSSBAR_147	DSP2_IRQ_MMU1	DSP2	DSP2 MMU1 interrupt
IRQ_CROSSBAR_148	McASP2_IRQ_AREVT	McASP2	McASP2 receive interrupt
IRQ_CROSSBAR_149	McASP2_IRQ_AXEVT	McASP2	McASP2 transmit interrupt
IRQ_CROSSBAR_150	McASP3_IRQ_AREVT	McASP3	McASP3 receive interrupt
IRQ_CROSSBAR_151	McASP3_IRQ_AXEVT	McASP3	McASP3 transmit interrupt
IRQ_CROSSBAR_152	McASP4_IRQ_AREVT	McASP4	McASP4 receive interrupt
IRQ_CROSSBAR_153	McASP4_IRQ_AXEVT	McASP4	McASP4 transmit interrupt
IRQ_CROSSBAR_154	McASP5_IRQ_AREVT	McASP5	McASP5 receive interrupt
IRQ_CROSSBAR_155	McASP5_IRQ_AXEVT	McASP5	McASP5 transmit interrupt
IRQ_CROSSBAR_156	McASP6_IRQ_AREVT	McASP6	McASP6 receive interrupt
IRQ_CROSSBAR_157	McASP6_IRQ_AXEVT	McASP6	McASP6 transmit interrupt
IRQ_CROSSBAR_158	McASP7_IRQ_AREVT	McASP7	McASP7 receive interrupt
IRQ_CROSSBAR_159	McASP7_IRQ_AXEVT	McASP7	McASP7 transmit interrupt
IRQ_CROSSBAR_160	McASP8_IRQ_AREVT	McASP8	McASP8 receive interrupt
IRQ_CROSSBAR_161	McASP8_IRQ_AXEVT	McASP8	McASP8 transmit interrupt
IRQ_CROSSBAR_162	Reserved	Reserved	Reserved
IRQ_CROSSBAR_163	Reserved	Reserved	Reserved
IRQ_CROSSBAR_164	OCMC_RAM1_IRQ	OCMC_RAM1	OCMC_RAM1 interrupt
IRQ_CROSSBAR_165	OCMC_RAM2_IRQ	OCMC_RAM2	OCMC_RAM2 interrupt
IRQ_CROSSBAR_166	OCMC_RAM3_IRQ	OCMC_RAM3	OCMC_RAM3 interrupt
IRQ_CROSSBAR_167	Reserved	Reserved	Reserved

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_168	EVE1_IRQ_OUT0	EVE1	EVE1 output interrupt 0
IRQ_CROSSBAR_169	EVE1_IRQ_OUT1	EVE1	EVE1 output interrupt 1
IRQ_CROSSBAR_170	EVE1_IRQ_OUT2	EVE1	EVE1 output interrupt 2
IRQ_CROSSBAR_171	EVE1_IRQ_OUT3	EVE1	EVE1 output interrupt 3
IRQ_CROSSBAR_172	EVE2_IRQ_OUT0	EVE2	EVE2 output interrupt 0
IRQ_CROSSBAR_173	EVE2_IRQ_OUT1	EVE2	EVE2 output interrupt 1
IRQ_CROSSBAR_174	EVE2_IRQ_OUT2	EVE2	EVE2 output interrupt 2
IRQ_CROSSBAR_175	EVE2_IRQ_OUT3	EVE2	EVE2 output interrupt 3
IRQ_CROSSBAR_[176:203]	Reserved	Reserved	Reserved
IRQ_CROSSBAR_204	PWMSS1_IRQ_ePWM0_TZINT	PWMSS1	eHRPWM0 TZ interrupt
IRQ_CROSSBAR_205	PWMSS2_IRQ_ePWM1_TZINT	PWMSS2	eHRPWM1 TZ interrupt
IRQ_CROSSBAR_206	PWMSS3_IRQ_ePWM2_TZINT	PWMSS3	eHRPWM2 TZ interrupt
IRQ_CROSSBAR_207	PWMSS1_IRQ_ePWM0INT	PWMSS1	eHRPWM0 event/interrupt
IRQ_CROSSBAR_208	PWMSS2_IRQ_ePWM1INT	PWMSS2	eHRPWM1 event/interrupt
IRQ_CROSSBAR_209	PWMSS3_IRQ_ePWM2INT	PWMSS3	eHRPWM2 event/interrupt
IRQ_CROSSBAR_210	PWMSS1_IRQ_eQEP0INT	PWMSS1	eQEP0 event/interrupt
IRQ_CROSSBAR_211	PWMSS2_IRQ_eQEP1INT	PWMSS2	eQEP1 event/interrupt
IRQ_CROSSBAR_212	PWMSS3_IRQ_eQEP2INT	PWMSS3	eQEP2 event/interrupt
IRQ_CROSSBAR_213	PWMSS1_IRQ_eCAP0INT	PWMSS1	eCAP0 event/interrupt
IRQ_CROSSBAR_214	PWMSS2_IRQ_eCAP1INT	PWMSS2	eCAP1 event/interrupt
IRQ_CROSSBAR_215	PWMSS3_IRQ_eCAP2INT	PWMSS3	eCAP2 event/interrupt
IRQ_CROSSBAR_216	Reserved	Reserved	Reserved
IRQ_CROSSBAR_217	RTC_SS_IRQ_ALARM	RTC_SS	RTC_SS alarm interrupt
IRQ_CROSSBAR_218	UART7_IRQ	UART7	UART7 interrupt
IRQ_CROSSBAR_219	UART8_IRQ	UART8	UART8 interrupt
IRQ_CROSSBAR_220	UART9_IRQ	UART9	UART9 interrupt
IRQ_CROSSBAR_221	UART10_IRQ	UART10	UART10 interrupt
IRQ_CROSSBAR_222	DCAN1_IRQ_INT0	DCAN1	DCAN1 interrupt 0
IRQ_CROSSBAR_223	DCAN1_IRQ_INT1	DCAN1	DCAN1 interrupt 1
IRQ_CROSSBAR_224	DCAN1_IRQ_PARITY	DCAN1	DCAN1 parity interrupt
IRQ_CROSSBAR_225	DCAN2_IRQ_INT0	DCAN2	DCAN2 interrupt 0
IRQ_CROSSBAR_226	DCAN2_IRQ_INT1	DCAN2	DCAN2 interrupt 1
IRQ_CROSSBAR_227	DCAN2_IRQ_PARITY	DCAN2	DCAN2 parity interrupt
IRQ_CROSSBAR_228	MLB_IRQ_SYS_INT0	MLB	MLB sys interrupt 0
IRQ_CROSSBAR_229	MLB_IRQ_SYS_INT1	MLB	MLB sys interrupt 1
IRQ_CROSSBAR_230	VCP1_IRQ_INT	VCP1	VCP1 interrupt
IRQ_CROSSBAR_231	VCP2_IRQ_INT	VCP2	VCP2 interrupt
IRQ_CROSSBAR_232	PCIe_SS1_IRQ_INT0	PCIe_SS1	PCIe_SS1 interrupt 0
IRQ_CROSSBAR_233	PCIe_SS1_IRQ_INT1	PCIe_SS1	PCIe_SS1 interrupt 1
IRQ_CROSSBAR_234	Reserved	Reserved	Reserved
IRQ_CROSSBAR_235	Reserved	Reserved	Reserved
IRQ_CROSSBAR_236	Reserved	Reserved	Reserved
IRQ_CROSSBAR_237	MAILBOX2_IRQ_USER0	MAILBOX2	Mailbox 2 user 0 interrupt
IRQ_CROSSBAR_238	MAILBOX2_IRQ_USER1	MAILBOX2	Mailbox 2 user 1 interrupt

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_239	MAILBOX2_IRQ_USER2	MAILBOX2	Mailbox 2 user 2 interrupt
IRQ_CROSSBAR_240	MAILBOX2_IRQ_USER3	MAILBOX2	Mailbox 2 user 3 interrupt
IRQ_CROSSBAR_241	MAILBOX3_IRQ_USER0	MAILBOX3	Mailbox 3 user 0 interrupt
IRQ_CROSSBAR_242	MAILBOX3_IRQ_USER1	MAILBOX3	Mailbox 3 user 1 interrupt
IRQ_CROSSBAR_243	MAILBOX3_IRQ_USER2	MAILBOX3	Mailbox 3 user 2 interrupt
IRQ_CROSSBAR_244	MAILBOX3_IRQ_USER3	MAILBOX3	Mailbox 3 user 3 interrupt
IRQ_CROSSBAR_245	MAILBOX4_IRQ_USER0	MAILBOX4	Mailbox 4 user 0 interrupt
IRQ_CROSSBAR_246	MAILBOX4_IRQ_USER1	MAILBOX4	Mailbox 4 user 1 interrupt
IRQ_CROSSBAR_247	MAILBOX4_IRQ_USER2	MAILBOX4	Mailbox 4 user 2 interrupt
IRQ_CROSSBAR_248	MAILBOX4_IRQ_USER3	MAILBOX4	Mailbox 4 user 3 interrupt
IRQ_CROSSBAR_249	MAILBOX5_IRQ_USER0	MAILBOX5	Mailbox 5 user 0 interrupt
IRQ_CROSSBAR_250	MAILBOX5_IRQ_USER1	MAILBOX5	Mailbox 5 user 1 interrupt
IRQ_CROSSBAR_251	MAILBOX5_IRQ_USER2	MAILBOX5	Mailbox 5 user 2 interrupt
IRQ_CROSSBAR_252	MAILBOX5_IRQ_USER3	MAILBOX5	Mailbox 5 user 3 interrupt
IRQ_CROSSBAR_253	MAILBOX6_IRQ_USER0	MAILBOX6	Mailbox 6 user 0 interrupt
IRQ_CROSSBAR_254	MAILBOX6_IRQ_USER1	MAILBOX6	Mailbox 6 user 1 interrupt
IRQ_CROSSBAR_255	MAILBOX6_IRQ_USER2	MAILBOX6	Mailbox 6 user 2 interrupt
IRQ_CROSSBAR_256	MAILBOX6_IRQ_USER3	MAILBOX6	Mailbox 6 user 3 interrupt
IRQ_CROSSBAR_257	MAILBOX7_IRQ_USER0	MAILBOX7	Mailbox 7 user 0 interrupt
IRQ_CROSSBAR_258	MAILBOX7_IRQ_USER1	MAILBOX7	Mailbox 7 user 1 interrupt
IRQ_CROSSBAR_259	MAILBOX7_IRQ_USER2	MAILBOX7	Mailbox 7 user 2 interrupt
IRQ_CROSSBAR_260	MAILBOX7_IRQ_USER3	MAILBOX7	Mailbox 7 user 3 interrupt
IRQ_CROSSBAR_261	MAILBOX8_IRQ_USER0	MAILBOX8	Mailbox 8 user 0 interrupt
IRQ_CROSSBAR_262	MAILBOX8_IRQ_USER1	MAILBOX8	Mailbox 8 user 1 interrupt
IRQ_CROSSBAR_263	MAILBOX8_IRQ_USER2	MAILBOX8	Mailbox 8 user 2 interrupt
IRQ_CROSSBAR_264	MAILBOX8_IRQ_USER3	MAILBOX8	Mailbox 8 user 3 interrupt
IRQ_CROSSBAR_265	MAILBOX9_IRQ_USER0	MAILBOX9	Mailbox 9 user 0 interrupt
IRQ_CROSSBAR_266	MAILBOX9_IRQ_USER1	MAILBOX9	Mailbox 9 user 1 interrupt
IRQ_CROSSBAR_267	MAILBOX9_IRQ_USER2	MAILBOX9	Mailbox 9 user 2 interrupt
IRQ_CROSSBAR_268	MAILBOX9_IRQ_USER3	MAILBOX9	Mailbox 9 user 3 interrupt
IRQ_CROSSBAR_269	MAILBOX10_IRQ_USER0	MAILBOX10	Mailbox 10 user 0 interrupt
IRQ_CROSSBAR_270	MAILBOX10_IRQ_USER1	MAILBOX10	Mailbox 10 user 1 interrupt
IRQ_CROSSBAR_271	MAILBOX10_IRQ_USER2	MAILBOX10	Mailbox 10 user 2 interrupt
IRQ_CROSSBAR_272	MAILBOX10_IRQ_USER3	MAILBOX10	Mailbox 10 user 3 interrupt
IRQ_CROSSBAR_273	MAILBOX11_IRQ_USER0	MAILBOX11	Mailbox 11 user 0 interrupt
IRQ_CROSSBAR_274	MAILBOX11_IRQ_USER1	MAILBOX11	Mailbox 11 user 1 interrupt
IRQ_CROSSBAR_275	MAILBOX11_IRQ_USER2	MAILBOX11	Mailbox 11 user 2 interrupt
IRQ_CROSSBAR_276	MAILBOX11_IRQ_USER3	MAILBOX11	Mailbox 11 user 3 interrupt
IRQ_CROSSBAR_277	MAILBOX12_IRQ_USER0	MAILBOX12	Mailbox 12 user 0 interrupt
IRQ_CROSSBAR_278	MAILBOX12_IRQ_USER1	MAILBOX12	Mailbox 12 user 1 interrupt
IRQ_CROSSBAR_279	MAILBOX12_IRQ_USER2	MAILBOX12	Mailbox 12 user 2 interrupt
IRQ_CROSSBAR_280	MAILBOX12_IRQ_USER3	MAILBOX12	Mailbox 12 user 3 interrupt
IRQ_CROSSBAR_281	EVE1_IRQ_TPCC_REGION1	EVE1	EVE1 TPCC region 1 interrupt
IRQ_CROSSBAR_282	EVE1_IRQ_TPCC_REGION2	EVE1	EVE1 TPCC region 2 interrupt

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_283	EVE1_IRQ_TPCC_REGION3	EVE1	EVE1 TPCC region 3 interrupt
IRQ_CROSSBAR_284	EVE1_IRQ_MBX0_USER1	EVE1	EVE1 mailbox 0 user 1 interrupt
IRQ_CROSSBAR_285	EVE1_IRQ_MBX0_USER2	EVE1	EVE1 mailbox 0 user 2 interrupt
IRQ_CROSSBAR_286	EVE1_IRQ_MBX0_USER3	EVE1	EVE1 mailbox 0 user 3 interrupt
IRQ_CROSSBAR_287	EVE1_IRQ_MBX1_USER1	EVE1	EVE1 mailbox 1 user 1 interrupt
IRQ_CROSSBAR_288	EVE1_IRQ_MBX1_USER2	EVE1	EVE1 mailbox 1 user 2 interrupt
IRQ_CROSSBAR_289	EVE1_IRQ_MBX1_USER3	EVE1	EVE1 mailbox 1 user 3 interrupt
IRQ_CROSSBAR_290	EVE2_IRQ_TPCC_REGION1	EVE2	EVE2 TPCC region 1 interrupt
IRQ_CROSSBAR_291	EVE2_IRQ_TPCC_REGION2	EVE2	EVE2 TPCC region 2 interrupt
IRQ_CROSSBAR_292	EVE2_IRQ_TPCC_REGION3	EVE2	EVE2 TPCC region 3 interrupt
IRQ_CROSSBAR_293	EVE2_IRQ_MBX0_USER1	EVE2	EVE2 mailbox 0 user 1 interrupt
IRQ_CROSSBAR_294	EVE2_IRQ_MBX0_USER2	EVE2	EVE2 mailbox 0 user 2 interrupt
IRQ_CROSSBAR_295	EVE2_IRQ_MBX0_USER3	EVE2	EVE2 mailbox 0 user 3 interrupt
IRQ_CROSSBAR_296	EVE2_IRQ_MBX1_USER1	EVE2	EVE2 mailbox 1 user 1 interrupt
IRQ_CROSSBAR_297	EVE2_IRQ_MBX1_USER2	EVE2	EVE2 mailbox 1 user 2 interrupt
IRQ_CROSSBAR_298	EVE2_IRQ_MBX1_USER3	EVE2	EVE2 mailbox 1 user 3 interrupt
IRQ_CROSSBAR_[299:316]	Reserved	Reserved	Reserved
IRQ_CROSSBAR_317	DSP1_IRQ_TPCC_ERR	DSP1	DSP1 TPCC error interrupt
IRQ_CROSSBAR_318	DSP1_IRQ_TPCC_GLOBAL	DSP1	DSP1 TPCC global interrupt
IRQ_CROSSBAR_319	DSP1_IRQ_TPCC_REGION0	DSP1	DSP1 TPCC region 0 interrupt
IRQ_CROSSBAR_320	DSP1_IRQ_TPCC_REGION1	DSP1	DSP1 TPCC region 1 interrupt
IRQ_CROSSBAR_321	DSP1_IRQ_TPCC_REGION2	DSP1	DSP1 TPCC region 2 interrupt
IRQ_CROSSBAR_322	DSP1_IRQ_TPCC_REGION3	DSP1	DSP1 TPCC region 3 interrupt
IRQ_CROSSBAR_323	DSP1_IRQ_TPCC_REGION4	DSP1	DSP1 TPCC region 4 interrupt
IRQ_CROSSBAR_324	DSP1_IRQ_TPCC_REGION5	DSP1	DSP1 TPCC region 5 interrupt
IRQ_CROSSBAR_325	DSP2_IRQ_TPCC_ERR	DSP2	DSP2 TPCC error interrupt
IRQ_CROSSBAR_326	DSP2_IRQ_TPCC_GLOBAL	DSP2	DSP2 TPCC global interrupt
IRQ_CROSSBAR_327	DSP2_IRQ_TPCC_REGION0	DSP2	DSP2 TPCC region 0 interrupt
IRQ_CROSSBAR_328	DSP2_IRQ_TPCC_REGION1	DSP2	DSP2 TPCC region 1 interrupt
IRQ_CROSSBAR_329	DSP2_IRQ_TPCC_REGION2	DSP2	DSP2 TPCC region 2 interrupt
IRQ_CROSSBAR_330	DSP2_IRQ_TPCC_REGION3	DSP2	DSP2 TPCC region 3 interrupt
IRQ_CROSSBAR_331	DSP2_IRQ_TPCC_REGION4	DSP2	DSP2 TPCC region 4 interrupt
IRQ_CROSSBAR_332	DSP2_IRQ_TPCC_REGION5	DSP2	DSP2 TPCC region 5 interrupt
IRQ_CROSSBAR_333	MMU1_IRQ	MMU1	Top level MMU1 interrupt
IRQ_CROSSBAR_334	GMAC_SW_IRQ_RX_THRESH_PULSE	GMAC_SW	GMAC_SW receive threshold interrupt
IRQ_CROSSBAR_335	GMAC_SW_IRQ_RX_PULSE	GMAC_SW	GMAC_SW receive interrupt
IRQ_CROSSBAR_336	GMAC_SW_IRQ_TX_PULSE	GMAC_SW	GMAC_SW transmit interrupt
IRQ_CROSSBAR_337	GMAC_SW_IRQ_MISC_PULSE	GMAC_SW	GMAC_SW miscellaneous interrupt
IRQ_CROSSBAR_338	Reserved	Reserved	Reserved
IRQ_CROSSBAR_339	TIMER13_IRQ	TIMER13	TIMER13 interrupt
IRQ_CROSSBAR_340	TIMER14_IRQ	TIMER14	TIMER14 interrupt
IRQ_CROSSBAR_341	TIMER15_IRQ	TIMER15	TIMER15 interrupt
IRQ_CROSSBAR_342	TIMER16_IRQ	TIMER16	TIMER16 interrupt

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_343	QSPI_IRQ	QSPI	QSPI interrupt
IRQ_CROSSBAR_344	USB3_IRQ_INTR1	USB3	USB3 interrupt 1
IRQ_CROSSBAR_345	USB4_IRQ_INTR0	USB4	USB4 interrupt 0
IRQ_CROSSBAR_346	USB4_IRQ_INTR1	USB4	USB4 interrupt 1
IRQ_CROSSBAR_347	GPIO7_IRQ_2	GPIO7	GPIO7 interrupt 2
IRQ_CROSSBAR_348	GPIO8_IRQ_2	GPIO8	GPIO8 interrupt 2
IRQ_CROSSBAR_349	Reserved	Reserved	Reserved
IRQ_CROSSBAR_350	Reserved	Reserved	Reserved
IRQ_CROSSBAR_351	VIP1_IRQ_1	VIP1	VIP1 interrupt 1
IRQ_CROSSBAR_352	VIP2_IRQ_1	VIP2	VIP2 interrupt 1
IRQ_CROSSBAR_353	VIP3_IRQ_1	VIP3	VIP3 interrupt 1
IRQ_CROSSBAR_354	VPE_IRQ	VPE	VPE interrupt
IRQ_CROSSBAR_355	PCIe_SS2_IRQ_INT0	PCIe_SS2	PCIe_SS2 interrupt 0
IRQ_CROSSBAR_356	PCIe_SS2_IRQ_INT1	PCIe_SS2	PCIe_SS2 interrupt 1
IRQ_CROSSBAR_357	Reserved	Reserved	Reserved
IRQ_CROSSBAR_358	Reserved	Reserved	Reserved
IRQ_CROSSBAR_359	EDMA_TPCC_IRQ_ERR	EDMA TPCC	EDMA TPCC error interrupt
IRQ_CROSSBAR_360	EDMA_TPCC_IRQ_MP	EDMA TPCC	EDMA TPCC memory protection interrupt
IRQ_CROSSBAR_361	EDMA_TPCC_IRQ_REGION0	EDMA TPCC	EDMA TPCC region 0 interrupt
IRQ_CROSSBAR_362	EDMA_TPCC_IRQ_REGION1	EDMA TPCC	EDMA TPCC region 1 interrupt
IRQ_CROSSBAR_363	EDMA_TPCC_IRQ_REGION2	EDMA TPCC	EDMA TPCC region 2 interrupt
IRQ_CROSSBAR_364	EDMA_TPCC_IRQ_REGION3	EDMA TPCC	EDMA TPCC region 3 interrupt
IRQ_CROSSBAR_365	EDMA_TPCC_IRQ_REGION4	EDMA TPCC	EDMA TPCC region 4 interrupt
IRQ_CROSSBAR_366	EDMA_TPCC_IRQ_REGION5	EDMA TPCC	EDMA TPCC region 5 interrupt
IRQ_CROSSBAR_367	EDMA_TPCC_IRQ_REGION6	EDMA TPCC	EDMA TPCC region 6 interrupt
IRQ_CROSSBAR_368	EDMA_TPCC_IRQ_REGION7	EDMA TPCC	EDMA TPCC region 7 interrupt
IRQ_CROSSBAR_369	MMU2_IRQ	MMU2	Top level MMU2 interrupt
IRQ_CROSSBAR_370	EDMA_TC0_IRQ_ERR	EDMA TC0	EDMA TPTC0 error interrupt
IRQ_CROSSBAR_371	EDMA_TC1_IRQ_ERR	EDMA TC1	EDMA TPTC1 error interrupt
IRQ_CROSSBAR_372	OCMC_RAM1_IRQ_CBUF	OCMC_RAM1	OCMC_RAM1 CBUF interrupt
IRQ_CROSSBAR_373	OCMC_RAM2_IRQ_CBUF	OCMC_RAM2	OCMC_RAM2 CBUF interrupt
IRQ_CROSSBAR_374	OCMC_RAM3_IRQ_CBUF	OCMC_RAM3	OCMC_RAM3 CBUF interrupt
IRQ_CROSSBAR_375	DSP1_IRQ_TPCC_REGION6	DSP1	DSP1 TPCC region 6 interrupt
IRQ_CROSSBAR_376	DSP1_IRQ_TPCC_REGION7	DSP1	DSP1 TPCC region 7 interrupt
IRQ_CROSSBAR_377	DSP2_IRQ_TPCC_REGION6	DSP2	DSP2 TPCC region 6 interrupt
IRQ_CROSSBAR_378	DSP2_IRQ_TPCC_REGION7	DSP2	DSP2 TPCC region 7 interrupt
IRQ_CROSSBAR_379	MAILBOX13_IRQ_USER0	MAILBOX13	Mailbox 13 user 0 interrupt
IRQ_CROSSBAR_380	MAILBOX13_IRQ_USER1	MAILBOX13	Mailbox 13 user 1 interrupt
IRQ_CROSSBAR_381	MAILBOX13_IRQ_USER2	MAILBOX13	Mailbox 13 user 2 interrupt
IRQ_CROSSBAR_382	MAILBOX13_IRQ_USER3	MAILBOX13	Mailbox 13 user 3 interrupt
IRQ_CROSSBAR_383	Reserved	Reserved	Reserved
IRQ_CROSSBAR_384	Reserved	Reserved	Reserved
IRQ_CROSSBAR_385	Reserved	Reserved	Reserved
IRQ_CROSSBAR_386	PRM_IRQ_IPU2	PRM	PRCM interrupt to IPU2

Table 17-9. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_387	PRM_IRQ_DSP2	PRM	PRCM interrupt to DSP2
IRQ_CROSSBAR_388	PRM_IRQ_EVE1	PRM	PRCM interrupt to EVE1
IRQ_CROSSBAR_389	PRM_IRQ_EVE2	PRM	PRCM interrupt to EVE2
IRQ_CROSSBAR_390	Reserved	Reserved	Reserved
IRQ_CROSSBAR_391	Reserved	Reserved	Reserved
IRQ_CROSSBAR_392	VIP1_IRQ_2	VIP1	VIP1 interrupt 2
IRQ_CROSSBAR_393	VIP2_IRQ_2	VIP2	VIP2 interrupt 2
IRQ_CROSSBAR_394	VIP3_IRQ_2	VIP3	VIP3 interrupt 2
IRQ_CROSSBAR_395	IPU1_IRQ_MMU	IPU1	IPU1 MMU interrupt
IRQ_CROSSBAR_396	IPU2_IRQ_MMU	IPU2	IPU2 MMU interrupt
IRQ_CROSSBAR_397	MLB_IRQ	MLB	MLB interrupt
IRQ_CROSSBAR_398	EVE1_IRQ_TPCC_REGION4	EVE1	EVE1 TPCC region 4 interrupt
IRQ_CROSSBAR_399	EVE2_IRQ_TPCC_REGION4	EVE2	EVE2 TPCC region 4 interrupt
IRQ_CROSSBAR_[400:419]	Reserved	Reserved	Reserved

17.4 Interrupt Controllers Functional Description

For detailed information about each device INTC (including functional description and registers descriptions), see the TRM chapters and Arm documents referenced in Section 17.1, *Interrupt Controllers Overview*.

Chapter 18

Control Module

This chapter describes the system control module for the device.

18.1 Control Module Overview 4016
18.2 Control Module Environment 4018
18.3 Control Module Integration 4019
18.4 Control Module Functional Description 4021
18.5 Control Module Register Manual 4059
18.6 IODELAYCONFIG Module Integration 4671
18.7 IODELAYCONFIG Module Register Manual 4672

18.1 Control Module Overview

The control module allows software control of the various operation modes supported by the device. It is composed of two submodules. The CTRL_MODULE_CORE submodule which resides in the COREAON power domain and the CTRL_MODULE_WKUP submodule which resides in the WKUPAON power domain. These two submodules represent a set of registers which are used to control the device I/O ports and also various kinds of settings related to the different device operation modes and also to its internal modules.

The CTRL_MODULE_CORE submodule has registers for the following features:

- Pad configuration with following controls:
  - Pad I/O multiplexing
  - Pad pullup and pulldown configuration
  - Pad wake-up detection enabling
  - Pad wake-up event status
  - Pad input buffer enable
  - Pad slew rate control
- Device thermal management control and status registers
- PBIAS cell and MMC1 I/O cells control
- IRQ_CROSSBAR and DMA_CROSSBAR control
- Control the priority of initiator accesses to the external SDRAM
- Control the priority of initiators connected to L3_MAIN interconnect
- Memory region lock registers
- Mapping of the device non-maskable interrupt (NMI) to respective cores
- Controls for the DDR2/DDR3 I/O Cells
- Controls for the DDR2/DDR3 associated vref-generation cells
- AVS Class 0 associated registers
- ABB associated registers
- PCIe related registers
- Standard eFuse logic
- Other miscellaneous functions:
  - Status of the system boot settings
  - DSP1 and DSP2 reset vector address
  - Settings assoaciated with USB, SATA, and HDMI PHYs
  - DSS PLLs multiplexing and enabling
  - Force MPU write nonposted transactions
  - Firewalls error status
  - Settings related to different peripheral modules
  - Others

The CTRL_MODULE_WKUP submodule has registers for the following features:

- Basic EMIF configuration settings
- XTAL Oscilator control
- Efuse I/O compensation
- Other functions

Figure 18-1 represents an overview block diagram of the control module.

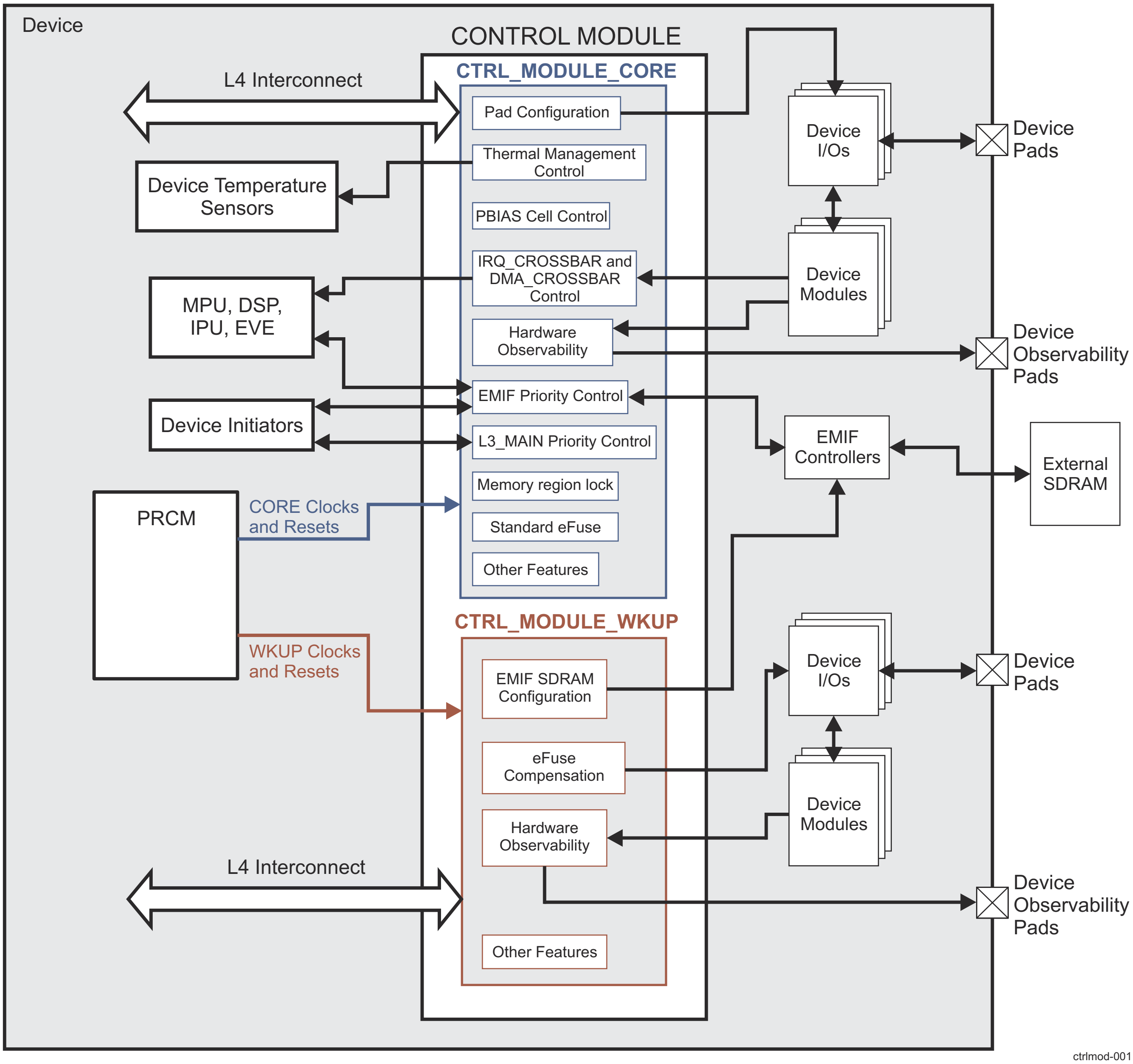


Figure 18-1. Control Module Overview Block Diagram

18.2 Control Module Environment

One of the control module functions is to control the multiplexing of certain signals from the device internal modules. Using the correct pad configuration settings, the control module maps these observability lines to the device boundary and a set of user selected internal module signals are available to be observed on the device pads. In addition, there are also two lines intended for observation of any of the device interrupt requests connected to the inputs of the IRQ_CROSSBAR module and two lines intended for observation of any of the DMA requests connected to the inputs of the DMA_CROSSBAR module. For more information, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description* and Section 18.4.6.5, *DMA_CROSSBAR Module Functional Description*.

An external non-maskable interrupt signal is also present. Using the appropriate registers the NMI can be mapped to different device cores. For more information, see Section 18.4.6.9, *NMI Mapping to respective cores*.

Figure 18-2 is an overview of the control module environment.

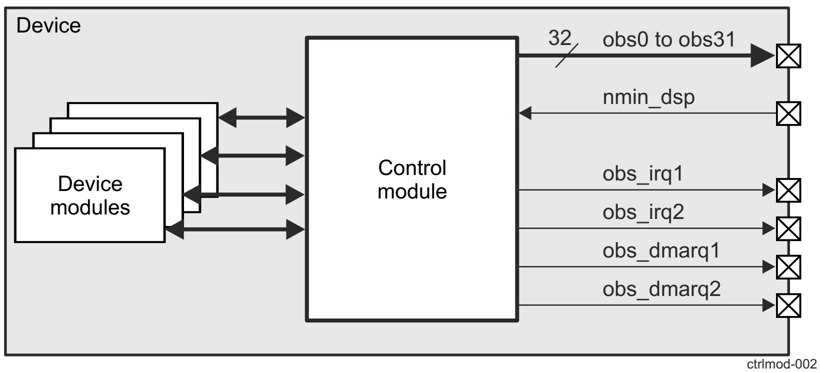


Figure 18-2. Control Module Environment

Table 18-1 shows the NMI input signal and observability outputs used to observe different signals of the device modules.

Table 18-1. Control Module I/O Description

Signal	I/O(1)	Description
obs[31:0]	O	32 internal hardware observability signals
nmin_dsp	I	External non-maskable interrupt signal
obs_irq1	O	Two signals intended for IRQ observation
obs_irq2	O	
obs_dmarq1	O	Two signals intended for DREQ observation
obs_dmarq2	O	

(1) I = Input; O = Output; I/O = Bidirectional

18.3 Control Module Integration

Figure 18-3 shows the integration of the control module in the device.

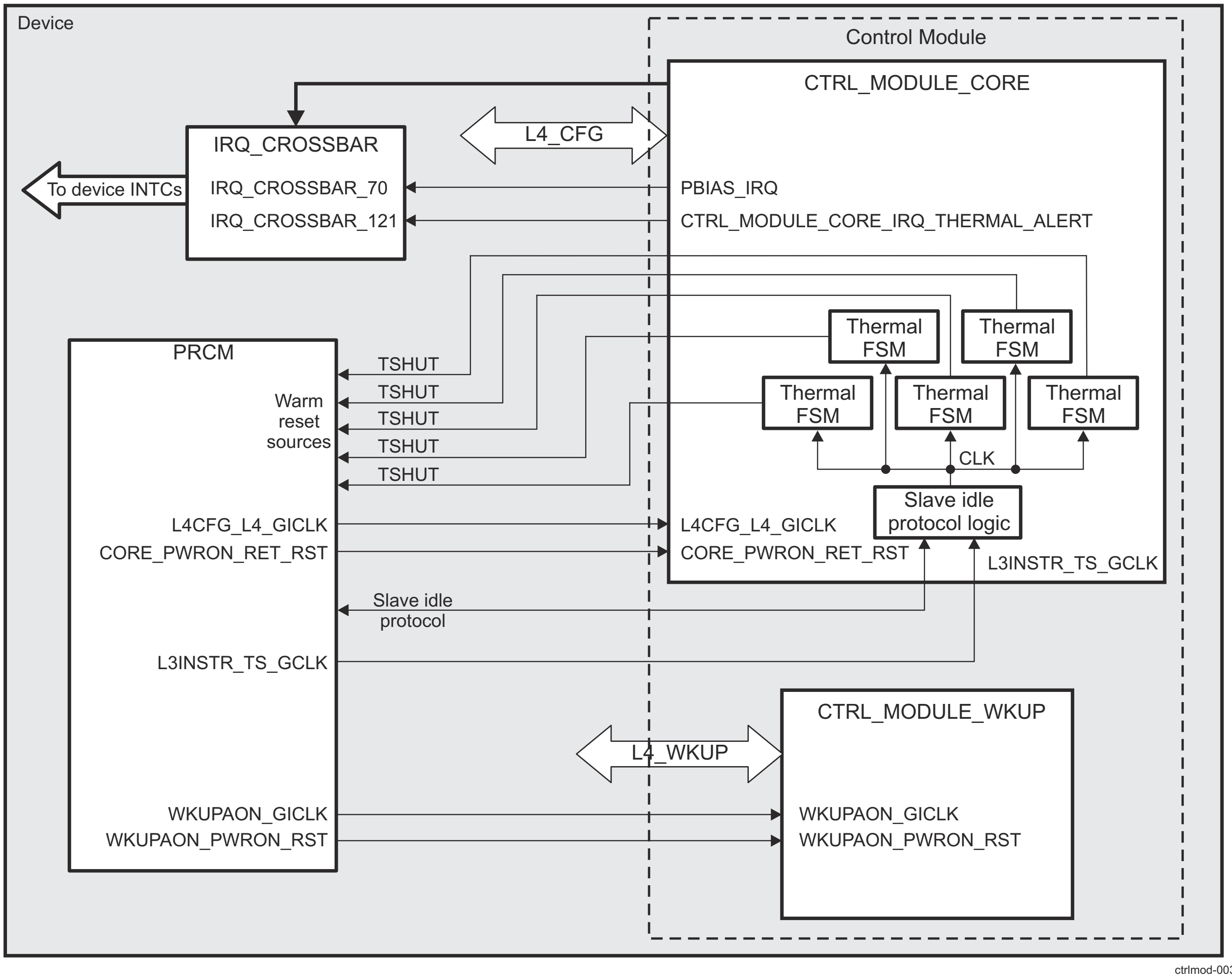


Figure 18-3. Control Module Integration

Table 18-2 through Table 18-4 summarize the integration of the Control Module in the device.

Table 18-2. Control Module Integration Attributes

Submodule	Attributes	
	Power Domain	Interconnect
CTRL_MODULE_CORE	PD_COREAON	L4_CFG
CTRL_MODULE_WKUP	PD_WKUPAON	L4_WKUP

Table 18-3. Control Module Clocks and Resets

Clocks				
Submodule	Destination Signal Name	Source Signal Name	Source	Description

Table 18-3. Control Module Clocks and Resets (continued)

CTRL_MODULE_CORE	L4CFG_L4_GICLK	L4CFG_L4_GICLK	PRCM	Interface clock to the CTRL_MODULE_CORE submodule
	L3INSTR_TS_GCLK	L3INSTR_TS_GCLK	PRCM	Common functional clock for the five thermal FSMs instantiated in the CTRL_MODULE_CORE submodule
CTRL_MODULE_WKUP	WKUPAON_GICLK	WKUPAON_GICLK	PRCM	Interface clock to the CTRL_MODULE_WKUP submodule
		Resets		
CTRL_MODULE_CORE	CORE_PWRON_RET_RST	CORE_PWRON_RET_RST	PRCM	Internal power-on reset (POR) affecting the CTRL_MODULE_CORE submodule
CTRL_MODULE_WKUP	WKUPAON_PWRON_RST	WKUPAON_PWRON_RST	PRCM	Internal POR affecting the CTRL_MODULE_WKUP submodule

Table 18-4. Control Module Hardware Requests

Interrupt Requests				
Submodule	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
CTRL_MODULE_CORE	PBIAS_IRQ	IRQ_CROSSBAR_70	MPU_IRQ_75	Interrupt signal generated by the PBIAS cell when the MMC1 I/Os supply voltage is not equal to the bias voltage generated by the PBIAS cell
	CTRL_MODULE_CORE_IRQ_THERMAL_ALERT	IRQ_CROSSBAR_121	MPU_IRQ_126	Thermal alert interrupt signal generated when one of the five thermal sensors goes over the tepmerature threshold value

Note

The “**Default Mapping**” column in Table 18-4 *Control Module Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*.
For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

The integration of the control module is the following:

- No wake-up request generation
- No DMA request generation
- No master standby protocol with the PRCM
- Slave idle protocol, between the CTRL_MODULE_CORE submodule and the PRCM, related only to the L3INSTR_TS_GCLK
- One interrupt request to the IRQ_CROSSBAR module
- Five (thermal shutdown) TSHUT signals used as PRCM warm reset sources
- Two clocks and one reset signal to the CTRL_MODULE_CORE submodule
- One clock and one reset signal to the CTRL_MODULE_WKUP submodule

18.4 Control Module Functional Description

18.4.1 Control Module Clock Configuration

There is no software control over the L4CFG_L4_GICLK and WKUPAON_GICLK clocks neither in the control module nor in the PRCM module. The L4CFG_L4_GICLK clock is automatically gated when there is no access to the CTRL_MODULE_CORE registers and the WKUPAON_GICLK is automatically gated when there is no access to the CTRL_MODULE_WKUP registers. There are clock activity status bits for these two clocks in the PRCM module.

The L3INSTR_TS_GCLK is controlled by the CTRL_CORE_BANDGAP_MASK_1[31:30] SIDLEMODE bit field. For more information, see Section 18.4.6.2.5.

The L3INSTR_TS_GCLK has also the following software controls located in the PRCM module:
- Status: see Chapter 3, *Power, Reset, and Clock Management*
- Divider ratio: see Chapter 3, *Power, Reset, and Clock Management*

18.4.2 Control Module Resets

The control module is not sensitive to software reset. It does not respond to global warm reset too. The control module is reset only by the internal POR (global cold reset).

Despite the previously stated that control module is not sensitive to global warm reset all CTRL_CORE_PAD_x registers are exception of this rule and are sensitive to global warm reset. All other control module registers are sensitive only to global cold reset.

The PRCM provides the CORE_PWRON_RET_RST POR signal to the CTRL_MODULE_CORE and the WKUPAON_PWRON_RST POR signal to the CTRL_MODULE_WKUP. For more information, see Section 3.5.5, *Reset Domains*, in Chapter 3, *Power, Reset, and Clock Management*.

18.4.3 Control Module Power Management

18.4.3.1 Power Management Protocols

The control module, which is slave on the L4 interconnect, does not support master standby or slave idle protocols for handshaking with the PRCM. Only the five thermal FSMs support slave idle protocol used to control their common functional clock, that is the L3INSTR_TS_GCLK clock.

18.4.4 Hardware Requests

The control module does not generate DMA and wake-up requests. The CTRL_MODULE_CORE submodule generates only two IRQs to the IRQ_CROSSBAR module. The first one is the PBIAS_IRQ and the second one is the CTRL_MODULE_CORE_IRQ_THERMAL_ALERT. For more information, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*.

18.4.5 Control Module Initialization

The control module responds to the internal POR. During device initialization, only modules used at boot time are associated with the pads. Other module inputs are internally tied and output buffers are turned off. After POR, software must set the pad configuration registers to appropriate values according to the desired device configuration.

The CTRL_CORE_BOOTSTRAP[15:0] bit field reflects the state of the sysboot[15:0] pads captured at POR in the PRCM module.

18.4.6 Functional Description Of The Various Register Types In CTRL_MODULE_CORE Submodule

The following sections describe in detail the purpose of the various kinds of registers and register groups which reside in the CTRL_MODULE_CORE submodule.

18.4.6.1 Pad Configuration

18.4.6.1.1 Pad Configuration Registers

The pad configuration registers are used to configure most of the device pads. Each pad configuration register is assotiated only with one pad. The name of each register is formed by the corresponding pad name and the prefix "CTRL_CORE_PAD_". Almost all pad configuration registers have same bits. In some of these registers certain bits cannot be present. Figure 18-4 shows the general case in which all the pad configuration register bits are present. Table 18-5 describes these bits.

After POR, software must set the pad configuration registers to appropriate values depending on the desired device configuration. Configuration of the pad configuration registers is normally done as part of the IO delay recalibration sequence described in Section 18.4.6.1.8, *IO Delay Recalibration*.

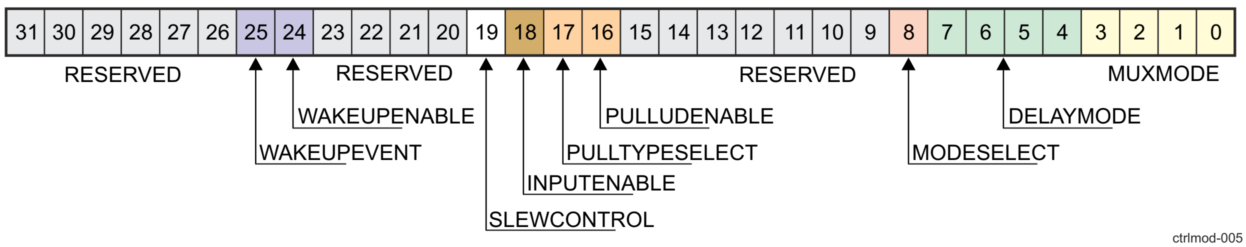


Figure 18-4. Pad Configuration Register Bits

Table 18-5. Description Of The Pad Configuration Register Bits

Bit/Bit Field	Bit Meaning		Description
	0b0	0b1	
WAKEUPEVENT	Wake-up event is not detected	Wake-up event is detected	Wake-up event status for a given pad. Indicates whenever the pad state is changed (0->1 or 1->0). In addition, when a wake-up event from a given I/O cell is received, the PRCM will wake up the MPU by switching-on its power domain and clocks and then generating an interrupt.
WAKEUPENABLE	Disable I/O wake-up function	Enable I/O wake-up function	Enables wake-up detection on input
SLEWCONTROL	Fast slew is selected	Slow slew is selected	Selects the slew rate for a given pad. The slew rate should be set to the value specified in the device Data Manual for a given mode of operation.
INPUTENABLE[1]	Receive mode is disabled. The pad is configured in output mode only.	Receive mode is enabled. The pad is configured in bidirectional mode.	Enables the input buffer of a given I/O
PULLTYPESELECT	Weak pull-down resistor is selected	Weak pull-up resistor is selected	Weak pull-up or weak pull-down resistor selection for a given pad
PULLUDENABLE	Weak pull-up/pull-down resistor is enabled	Weak pull-up/pull-down resistor is disabled	Enables weak pull-up/pull-down feature of a given pad
MODESELECT	Default IO Timing Mode is used	A Virtual or Manual IO Timing Mode is used.	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, *Virtual IO Timing Modes*. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, *Manual IO Timing Modes*.

Table 18-5. Description Of The Pad Configuration Register Bits (continued)

Bit/Bit Field	Bit Meaning		Description
	0b0	0b1	
DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, *Virtual IO Timing Modes* for details.		
MUXMODE	This bit field selects the desired Multiplexing Mode for the pad. See section "*Multiplexing Characteristics*" of the Data Manual for a list of available functions for each pad.		

(1) To enable/disable the input buffers of the mmc1_* pads the ACTIVE bits of the corresponding CTRL_CORE_PAD_x registers are used. These registers have ACTIVE bits instead of INPUTENABLE bits.

Note

The default SLEWCONTROL settings in each pad configuration register must be used to guarantee timings, unless specific instructions otherwise are given in the individual timing sub-sections of the device Data Manual. The only exception is when the MUXMODE is configured to select a vout*_* signal. In this case the corresponding SLEWCONTROL bit is recommended to be set to slow slew instead of the default fast slew. FAST slew setting is allowed, but results in faster edge rates on the VOUTn bus, higher power/ground noise, and higher EMI emissions compared to SLOW slew rate.

Note

It must be taken into account that when an external pull resistor is desired, the internal pull resistors must be disabled by software because they are enabled by default.

Note

GPIO pins can directly snoop device pads even if those are configured in the other modes. For GPIO output user needs to set pad mux.

The following signals require CTRL_CORE_PAD_x to be programmed with INPUTENABLE=1 for retiming purposes:

- mmc2_clk, mmc3_clk, and mmc4_clk
- gpmc_clk
- i2cj_scl where j=1-5
- spim_sclk where m=1-4
- mcaspi_aclkx, mcaspi_ahclkx and mcaspi_aclkr signals where i=1-8

18.4.6.1.1.1 Permanent PU/PD disabling (SR 2.0 only)

This section applies only to SR2.0.

The internal pull resistors of pads gpmc_a[27:24, 22:19] can be permanently disabled by pulling sysboot15 high during power-on-reset deassertion. When sysboot15 = 1, the corresponding PULLUDENABLE bit does not apply anymore. In this case it is a don't care bit whatever value set. This functionality is intended to be used when mmc2_dat* signals are configured on the gpmc_a[27:24, 22:19] pads. In this case, if sysboot15 = 1, external pull-up resistors with values as per JEDEC eMMC specification (JESD84-B451) must be connected to these pads. If gpmc_a[27:24, 22:19] are connected to non-multiplexed NOR flash used for booting and sysboot15 = 1, external pull-down resistors are mandatory on these pads otherwise boot is not possible. During NOR flash run-time accesses the external pull-down resistors are not needed.

18.4.6.1.2 Pull Selection

There is no automatic gating control to ensure that internal weak pull-up or pull-down resistors on a pad are disconnected whenever pad is configured as output. If a pad is always configured in output mode, it is recommended for user software to disable any internal pull resistor tied to it to avoid unnecessary consumption.

Table 18-6 describes the software controls available for pad internal pull-up and pull-down resistors in the control module pad configuration registers.

Table 18-6. Pull Selection

PULL		Pad Behavior
PULLTYPESELECT	PULLUDENABLE	
0	1	Pull-down selected but not activated
0	0	Pull-down selected and activated
1	1	Pull-up selected but not activated
1	0	Pull-up selected and activated

18.4.6.1.3 Pad multiplexing

Many of the Device pads support pad multiplexing. This means that their function can be independently chosen from two or more options. The selection of functions available on each pad is enumerated in the "*Multiplexing Characteristics*" section of the Data Manual. The desired function is selected via the MUXMODE field of the associated pad configuration register.

By default, the MUXMODE field of most device pads is set to 0xF, which means the pads are in Hi-Z. Only MUXMODE values which correspond to defined functions should be used.

Note

When setting the MUXMODE of any mmc1_* pad, the following register fields should also be programmed:

- CTRL_CORE_CONTROL_PBIAS[27] SDCARD_BIAS_PWRDNZ = 1
- CTRL_CORE_CONTROL_PBIAS[26] SDCARD_IO_PWRDNZ = 1
- If MUXMODE=0 is not selected, configure CTRL_CORE_CONTROL_PBIAS[21] SDCARD_BIAS_VMODE
  – Clear to 0 if vddshv8 = 1.8V
  – Set to 1 if vddshv8 = 3.3V

18.4.6.1.4 IOSETs

An IO signal may have multiplexing options across two or more device pads. In many cases, the user is allowed to select any combination of IO signal multiplexing options to use for an interface. But in some cases, specific combinations of multiplexing options must be selected to guarantee the Timing and Switching Characteristics in the *Data Manual*. These specific combinations are called IOSETs, and they generally represent layout-friendly groups of pads that are pinned-out in close proximity to each other. IOSETs are defined in the Data Manual for interfaces that require them.

18.4.6.1.5 Virtual IO Timing Modes

When operating a pad in certain modes, a Virtual IO Timing Mode must be selected to ensure that IO timings are met. The modes requiring Virtual IO Timing Modes are described in *Virtual Functions Mapping* tables within the "*Timing Requirements and Switching Characteristics*" section of the Data Manual. These tables list each pad associated with a specific Virtual IO Timing Mode, along with the DELAYMODE setting required for that pad.

To select a Virtual IO Timing Mode, both the MODESELECT field of each associated pad configuration register must be set to 0b1 and DELAYMODE field of each associated pad configuration register must be set to match the *Virtual Function Mapping* tables in the Data Manual.

Selection of all Virtual IO Timings Modes should be done as part of the IO Delay Recalibration Sequence described in Section 18.4.6.1.8, *IO Delay Recalibration*.

18.4.6.1.6 Manual IO Timing Modes

When operating a pad in certain modes, a Manual IO Timing Mode must be configured to ensure that IO timings are met. The modes requiring Manual IO Timing Modes are described in *Manual Functions Mapping* tables within the "*Timing Requirements and Switching Characteristics*" section of the Data Manual. These tables list each pad associated with a specific Manual IO Timing Mode, along with the required A_DELAY and G_DELAY values that should be used to calculate the correct values to be set in the CFG_x_IN, CFG_x_OEN, and CFG_x_OUT registers associated with that pad. For more information regarding these registers, see Section 18.7, *IODELAYCONFIG Module Register Manual*

To select a Manual IO Timing Mode, the MODESELECT field of each associated pad configuration register must be set to 0b1 and the associated CFG_x_IN, CFG_x_OEN, and CFG_x_OUT registers should be set via the following sequence:

1. Compute CDPE and FPDE based on the following equations using values from the IODELAYCONFIG module registers (the register values must be converted to decimal before calculating):

$$\text{CDPE} = \frac{10^*(\text{CONFIG\_REG\_3[15:0] COARSE\_REF\_COUNT}^*\text{CONFIG\_REG\_2[15:0] REFCLK\_PERIOD})}{(2^*(\text{CONFIG\_REG\_3[31:16] COARSE\_DELAY\_COUNT}^*88))}$$

$$\text{FDPE} = \frac{10^*(\text{CONFIG\_REG\_4[15:0] FINE\_REF\_COUNT}^*\text{CONFIG\_REG\_2[15:0] REFCLK\_PERIOD})}{(2^*(\text{CONFIG\_REG\_4[31:16] FINE\_DELAY\_COUNT}^*264))}$$

ctrlmod-014

2. For each pad that requires a Manual IO Timing Mode, perform the following setups to calculate the values required to be programmed into each of the associated CFG_x_IN, CFG_x_OEN, and CFG_x_OUT IODELAYCONFIG module registers:
   a. Calculate the Coarse Values and Fine Values corresponding to the A_DELAY and G_DELAY values listed in the Data Manual for the associated Manual IO Timing Mode as follows:

$$\text{G\_DELAY\_COARSE} = \text{floor}\left(\frac{\text{G\_DELAY}}{920}\right)$$

$$\text{G\_DELAY\_FINE} = \frac{\text{floor(G\_DELAY mod 920)}^*10}{60}$$

$$\text{A\_DELAY\_COARSE} = \text{floor}\left(\frac{\text{A\_DELAY}}{\text{CDPE}}\right)$$

$$\text{A\_DELAY\_FINE} = \frac{\text{floor(A\_DELAY mod CDPE)}^*10}{\text{FDPE}}$$

ctrlmod-015

   b. Calculate the required number of Coarse Elements and Fine Elements:

$$\text{COARSE\_ELEMENTS} = \text{G\_DELAY\_COARSE} + \text{A\_DELAY\_COARSE}$$

$$\text{FINE\_ELEMENTS} = \frac{\text{G\_DELAY\_FINE} + \text{A\_DELAY\_FINE}}{10}$$

ctrlmod-016

   c. If FINE_ELEMENTS > 22, recalculate COARSE_ELEMENTS and FINE_ELEMENTS as follows:

$$\text{COARSE\_ELEMENTS} = \frac{\text{TOTAL\_DELAY}}{\text{CDPE}}$$

$$\text{FINE\_ELEMENTS} = \frac{\text{TOTAL\_DELAY mod CDPE}}{\text{FDPE}}$$

Where TOTAL_DELAY = (Original COARSE_ELEMENTS * CDPE) + (Original FINE_ELEMENTS * FDPE)

ctrlmod-017

   d. Convert COARSE_ELEMENTS and FINE_ELEMENTS to 5-bit binary values COARSE_ELEMENTS_5b and FINE_ELEMENETS_5b
   e. Write associated CFG_x register with 0x29400 + COARSE_ELEMENTS_5b<<5 + FINE_ELEMENETs_5b

Selection of all Manual IO Timing Modes should be done as part of the IO Delay Recalibration Sequence described in Section 18.4.6.1.8, *IO Delay Recalibration*.

18.4.6.1.7 Isolation Requirements

When reprogramming the MUXMODE, DELAYMODE, and MODESELECT fields of a pad configuration register or the CFG_x_IN, CFG_x_OEN, and CFG_x_OUT registers in the IODELAYCONFIG Module, there is potential for a glitch on the corresponding IO. Therefore, the device IOs should be isolated when writing to any of these fields. Isolating the IOs forces them to tri-state output with internal pulls holding their previous levels. De-isolating the IOs returns their control to the selected peripheral modules.

Note

While the IOs are Isolated, code must execute only from internal RAM and the isolated IO interfaces should not be actively used.

Isolation is a global event that affects all LVCMOS IOs except for the following:

- JTAG IOs
- on_off
- rstoutn

Isolation/De-isolation of the device IOs is accomplished via the following sequences:

Isolation Sequence:

1. Ensure that the CTRL_CORE_CONTROL_PBIAS[27] SDCARD_BIAS_PWRDNZ and CTRL_CORE_CONTROL_PBIAS[26] SDCARD_IO_PWRDNZ bits are set to 1
2. Write 1 to the PRM_IO_PMCTRL[0] ISOCLK_OVERRIDE bit
3. Poll the PRM_IO_PMCTRL[1] ISOCLK_STATUS bit until it reads 1
4. Write 1 to the CTRL_CORE_SMA_SW_0[2] ISOLATE bit
5. Read the CTRL_CORE_SMA_SW_0[2] ISOLATE bit for timing purposes only
6. Write the PRM_IO_PMCTRL[0] ISOCLK_OVERRIDE bit to 0
7. Poll the PRM_IO_PMCTRL[1] ISOCLK_STATUS bit until it reads 0

De-isolation Sequence:

1. Write 1 to the PRM_IO_PMCTRL[0] ISOCLK_OVERRIDE bit
2. Poll the PRM_IO_PMCTRL[1] ISOCLK_STATUS bit until it reads 1
3. Write 0 to the CTRL_CORE_SMA_SW_0[2] ISOLATE bit
4. Read the CTRL_CORE_SMA_SW_0[2] ISOLATE bit for timing purposes only
5. Write the PRM_IO_PMCTRL[0] ISOCLK_OVERRIDE bit to 0
6. Poll the PRM_IO_PMCTRL[1] ISOCLK_STATUS bit until it reads 0

Configuration of the pad configuration registers is normally done as part of the IO Delay Recalibration Sequence described in Section 18.4.6.1.8, *IO Delay Recalibration*, which references the above Isolation/De-isolation sequences.

18.4.6.1.8 IO Delay Recalibration

After adjusting the AVS voltage for VDD_CORE_L voltage domain, an IO Delay Recalibration Sequence must be followed to ensure device IO timings are met. The IO Delay Recalibration Sequence is as follows:

IO Delay Recalibration Sequence:

1. Complete the AVS voltage change on the VDD_CORE_L voltage domain and ensure voltage has stabilized to the new AVS target voltage
2. Unlock the registers used by this sequence as follows:
   a. Write 0x2FF1AC2B to register CTRL_CORE_MMR_LOCK_1 of the Control Module
   b. Write 0x6F361E05 to register CTRL_CORE_MMR_LOCK_5 of the Control Module
   c. Write 0x0000AAAA to register CONFIG_REG_8 of the IODELAYCONFIG Module
3. Perform IO Delay Calibration

a. Write register field CONFIG_REG_2[15:0] REFCLK_PERIOD of the IODELAYCONFIG Module with the L4_ICLK clock period in ps divided by 10 (or, equivalently, L3_ICLK clock period in ps divided by 5), then rounded down to the closest integer
   i. L4_ICLK = 133MHz requires a value of 0x2EF
b. Write 1 to register field CONFIG_REG_0[0] CALIBRATION_START of the IODELAYCONFIG module to initiate the calibration.
c. Poll register field CONFIG_REG_0[0] CALIBRATION_START for 0, indicating calibration is complete.

4. Isolate the device IOs via the Isolation Sequence described in Section 18.4.6.1.7 *Isolation Requirements*
5. Update the delay mechanism for each IO with new calibrated values:
   a. Write 1 to register field CONFIG_REG_0[1] ROM_READ of the IODELAYCONFIG module to initiate a reload of calibrated delay values for all IOs.
   b. Poll register field CONFIG_REG_0[1] ROM_READ for 0, indicating reload is complete.
6. Configure the pad configuration register (CTRL_CORE_PAD_x) for each IO with the desired MUXMODE, DELAYMODE, and MODESELECT settings.
7. Configure all required Manual IO Timing Modes as described in Section 18.4.6.1.6, *Manual IO Timing Modes*
8. De-isolate the device IOs via the de-isolation sequence described in Section 18.4.6.1.7 *Isolation Requirements*
9. Relock the registers used by this sequence:
   a. Write 0x1A1C8144 to register CTRL_CORE_MMR_LOCK_1
   b. Write 0x143F832C to register CTRL_CORE_MMR_LOCK_5
   c. Write 0x0000AAAB to register CONFIG_REG_8

18.4.6.2 Thermal Management Related Registers

There are five temperature sensors on the device die. Each sensor is associated only with one voltage domain and is also a part of a VBGAPTS cell. This cell has a 10-bit ADC. The ADC converts the temperature values into digital output values proportional to the temperature measured. Each VBGAPTS cell is controlled by a dedicated FSM referred to as thermal FSM. The registers associated with each FSM reside in the CTRL_MODULE_CORE submodule. All FSMs are clocked by the L3INSTR_TS_GCLK clock. The PRCM module controlls this clock through the slave idle protocol.

The device thermal management related registers can be split into the following classes:

- Temperature sensors control registers
- Registers for the thermal alert comparators
- Temperature timestamp registers
- Other registers used for:
  – controlling the FIFOs
  – controlling the clock provided to the five FSMs

Figure 18-5 shows the block diagram of the device thermal management.

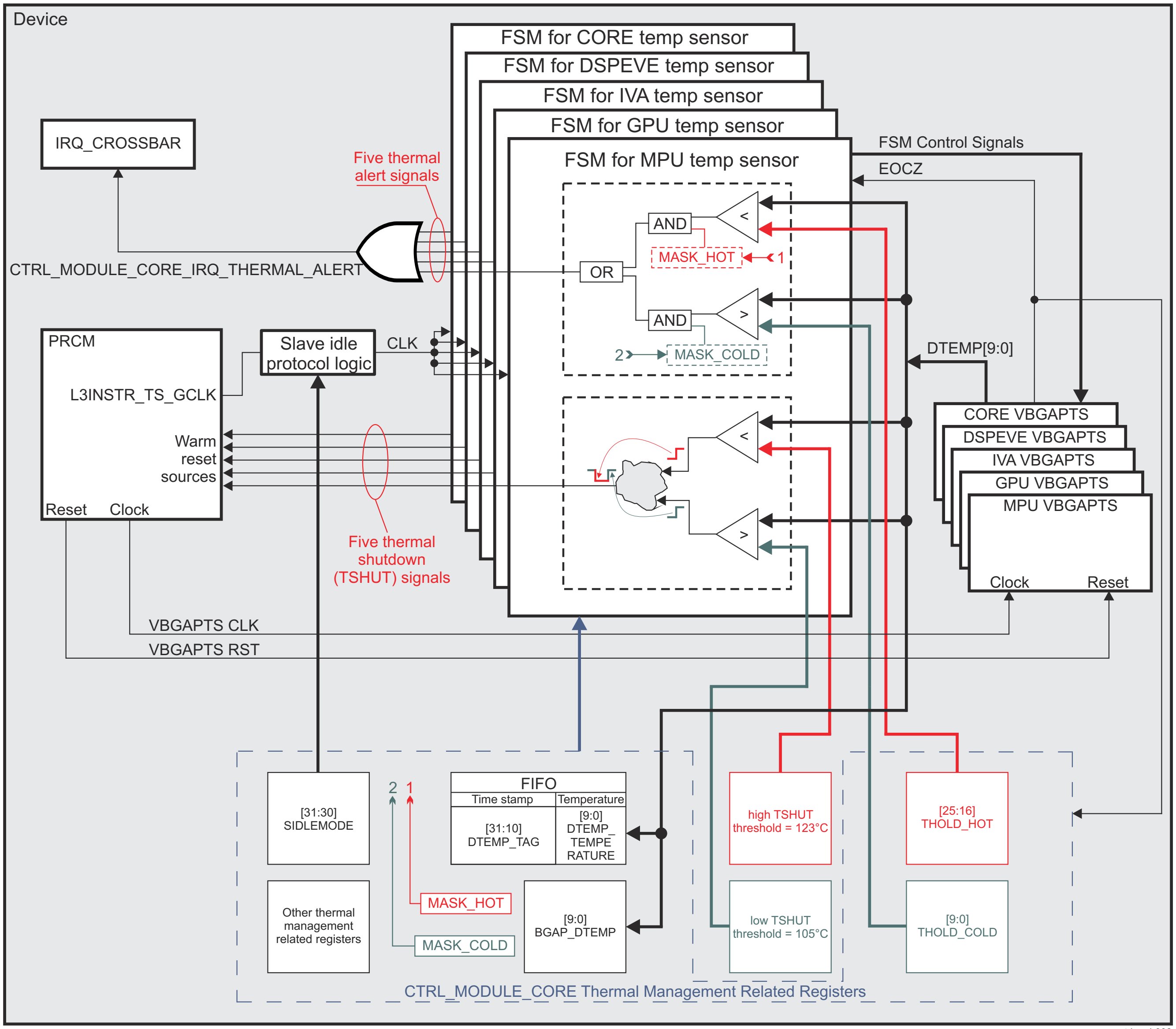


Figure 18-5. Thermal Management Functional Block Diagram

Table 18-7 describes the signals related to the device thermal management logic.

Table 18-7. Thermal Management Signals Description

Signal	I/O[(1)]	Description
EOCZ	O	End of conversion signal. When low, this signal indicates that the value of DTEMP[9:0] is valid.
DTEMP[9:0]	O	Temperature data from the temperature sensor. This value is valid when EOCZ is low.
VBGAPTS CLK	I	Functional clock from the WKUPAON power domain used by the temperature sensor during temperature conversion.
THERMAL ALERT	O	The five thermal alert outputs from each thermal FSM are ORed and then mapped as an interrupt request to the IRQ_CROSSBAR module. Software uses this interrupt to implement the device thermal management policy.

Table 18-7. Thermal Management Signals Description (continued)

Signal	I/O(1)	Description
TSHUT	O	Each of the five thermal shutdown signals is mapped to the PRCM and is used as a warm reset signal. These overheat protection signals are high during normal operation and go low during thermal shutdown event.

(1) I = Input; O = Output

The ADC values which correspond to the current temperature are listed in Table 18-10.

18.4.6.2.1 Temperature Sensors Control Registers

Each VBGAPTS cell works in continuous conversion mode controlled by the corresponding FSM. This means that the temperature is measured at regular time intervals. The start of temperature measurement is initiated automatically by each FSM after it goes out of reset state. To control the main delay between two measurements the CTRL_CORE_BANDGAP_MASK_1[29:27] COUNTER_DELAY bit field is used. After this delay expires the five FSMs automatically start a temperature conversion.

The conversion is complete when the CTRL_CORE_TEMP_SENSOR_x[10] BGAP_EOCZ_x status bits are set to 0x0. After this the valid temperature is written automatically by each FSM in the CTRL_CORE_TEMP_SENSOR_x[9:0] BGAP_DTEMP_x bit fields, and then software is able to read it from the corresponding register. After writing the valid temperature values the five FSMs wait one clock cycle and start another conversion cycle.

For details regarding the CTRL_CORE_TEMP_SENSOR_x registers, see Table 18-9.

18.4.6.2.2 Registers For The Thermal Alert Comparators

There is a comparator block responsible for the thermal alert function of the CTRL_MODULE_CORE thermal management logic. This comparator block is composed of two comparators. One dedicated to low temperature threshold and the other one to high temperature threshold. Because of the five temperature sensors there are also five comparator blocks. Software can configure the low temperature threshold through the CTRL_CORE_BANDGAP_THRESHOLD_x[9:0] THOLD_COLD_x bit fields and high temperature threshold through the CTRL_CORE_BANDGAP_THRESHOLD_x[25:16] THOLD_HOT_x bit fields. The values which have to be loaded in these bit fields are the same as those listed in Table 18-10. For example, to set the high temperature threshold value to 120°C, the THOLD_HOT filed must be loaded with value of 931 (0x3A3).

The thermal alert logic provides also a capability to mask the outputs of the comparators associated with low and high temperature thresholds. The low temperature threshold comparator outputs are masked by setting to 0x0 the corresponding MASK_COLD_x bits in the CTRL_CORE_BANDGAP_MASK_1 and CTRL_CORE_BANDGAP_MASK_2 registers. The high temperature threshold comparator outputs are masked by setting to 0x0 the corresponding MASK_HOT_x bits in CTRL_CORE_BANDGAP_MASK_1 and CTRL_CORE_BANDGAP_MASK_2 registers.

The masked low and high temperature threshold outputs are ORed once and as a result a single thermal alert signal is produced. As there are five temperature sensors there are also five thermal alert output signals produced. These five signals are then ORed (see Figure 18-5) in the CTRL_MODULE_CORE_IRQ_THERMAL_ALERT signal, which delivers an interrupt to the IRQ_CROSSBAR_121 input line of the IRQ_CROSSBAR module. Software can use this interrupt to implement the device thermal management policy. The CTRL_MODULE_CORE_IRQ_THERMAL_ALERT signal is also routed to the CTRL_CORE_BANDGAP_STATUS_1[31] ALERT bit. Value of 0x1 indicates that the CTRL_MODULE_CORE_IRQ_THERMAL_ALERT signal is asserted.

The non masked (raw) comparator outputs are available for reading through the corresponding bits in the CTRL_CORE_BANDGAP_STATUS_1 and CTRL_CORE_BANDGAP_STATUS_2 registers. The low temperature threshold comparator outputs are read through the COLD_x bits and the high temperature threshold comparator outputs through the HOT_x bits.

Figure 18-6 shows the behavior of the thermal alert logic.

For details regarding the CTRL_CORE_BANDGAP_THRESHOLD_x registers, see Table 18-9.

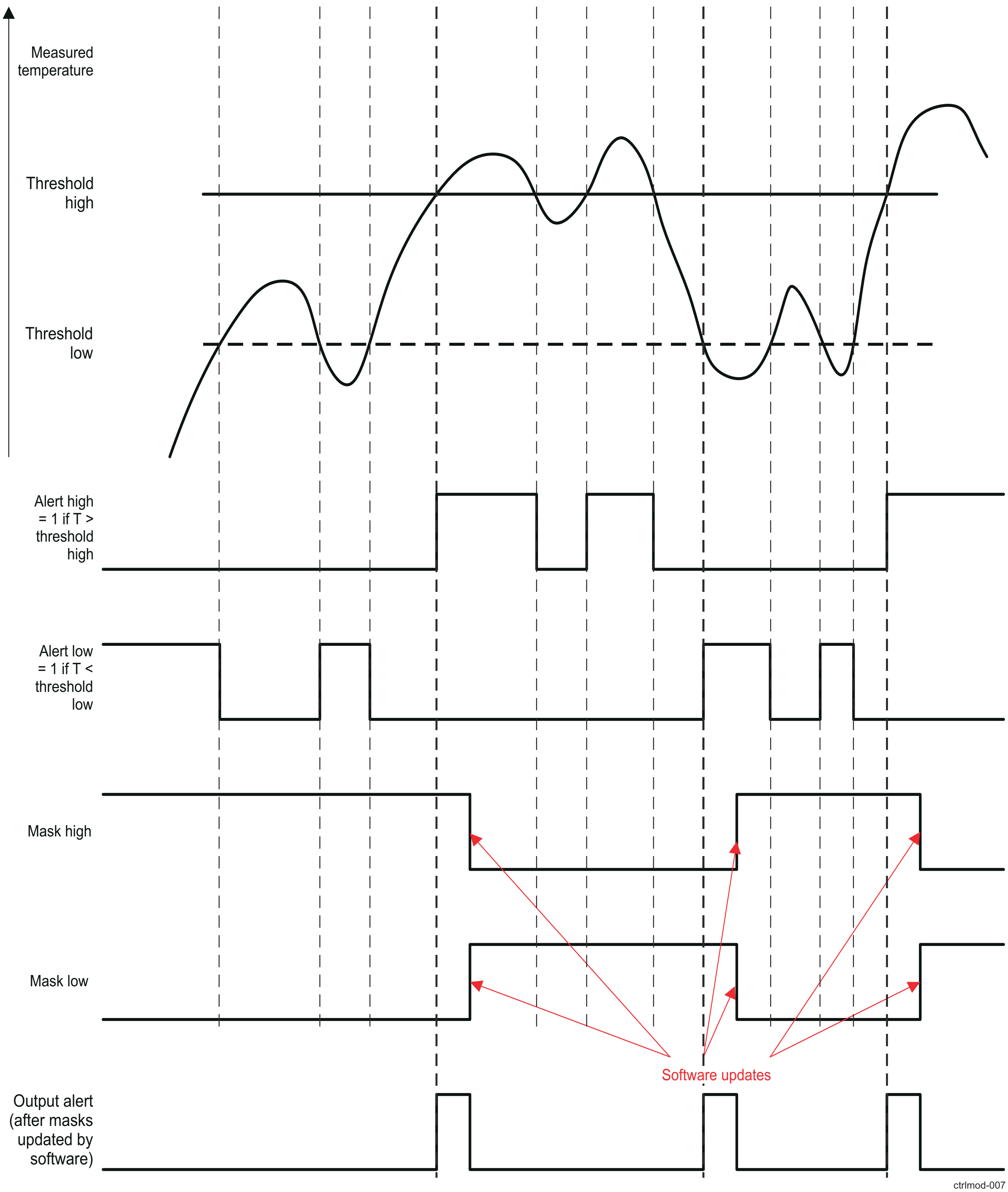


Figure 18-6. Behavior Of The Thermal Alert Logic

18.4.6.2.3 Thermal Shutdown Comparators

There is also a comparator block responsible for the thermal shutdown (TSHUT) function of the CTRL_MODULE_CORE thermal management logic. This comparator block is also composed of two comparators. One dedicated to low TSHUT threshold and the other one to high TSHUT threshold. Because of the five temperature sensors there are also five comparator blocks. The comparator outputs for the low and high TSHUT thresholds of each comparator block are connected to a logic which then generates a single TSHUT signal. All five TSHUT signals are tied to the PRCM and used as warm reset sources. For details, see Figure 18-5.

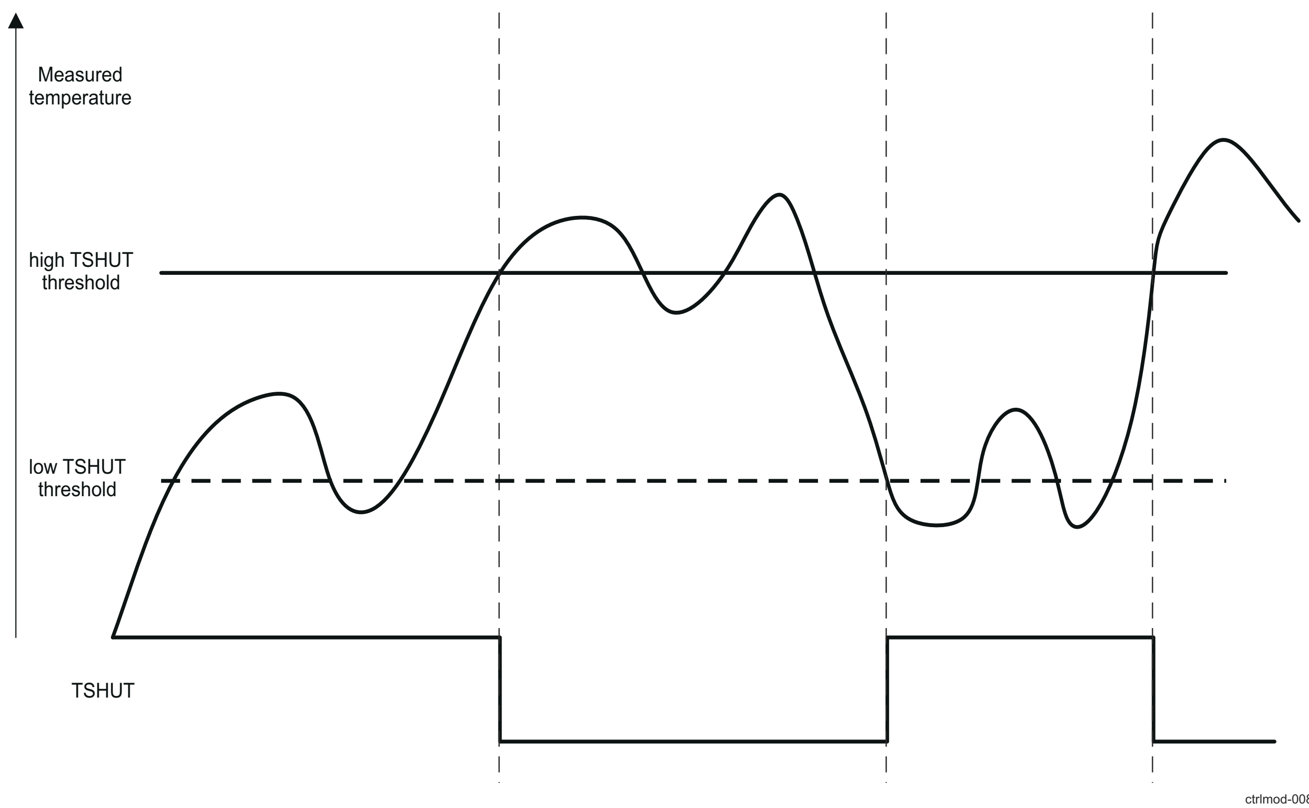
The values of the five low and high TSHUT thresholds are fixed and can be neither overridden nor read by software. The value for the high TSHUT threshold is 123°C (assuming ±2°C temperature sensor accuracy), and the low TSHUT threshold is 105°C.

When the high TSHUT threshold is reached, the TSHUT output is activated. To deactivate each TSHUT output, the temperature must go below the value of each low TSHUT threshold.

Each of the five TSHUT signals is used as an overheat protection.

Figure 18-7 shows the behavior of the TSHUT logic.

Figure 18-7. Behavior Of The Thermal Shutdown Logic



18.4.6.2.4 Temperature Timestamp Registers

Each time one of the CTRL_CORE_TEMP_SENSOR_x[9:0] BGAP_DTEMP_x bit fields is updated with new temperature value, this value is also automatically stored into a 5-level deep FIFO and a timestamp is registered too. There are five FIFOs used to store a brief history for the last few temperature measurements and are also dedicated to temperature timestamping feature. Each FIFO has two fields. The first one is 10 bits wide, 5 levels deep, and is intended to store the temperature values for the last five measurements. The second field is 22 bits wide, 5 levels deep, and acts like a counter for the number of temperature measurements. Each FIFO is composed of the following registers:

- CTRL_CORE_DTEMP_x_0
- CTRL_CORE_DTEMP_x_1
- CTRL_CORE_DTEMP_x_2
- CTRL_CORE_DTEMP_x_3
- CTRL_CORE_DTEMP_x_4

Table 18-8 shows a generic description of the five FIFOs.

For details regarding the five CTRL_CORE_DTEMP_x registers, see Table 18-9.

Table 18-8. FIFOs Generic Description

FIFO Levels	Second FIFO Field (22 Bits) – Timestamp	Description	First FIFO field (10 Bits) – Temperature	Description
	Bits [31:10]		Bits [9:0]	
Level 1	DTEMP_TAG_x_0	Indicates the number of temperature measurements	DTEMP_TEMPERATURE_x_0	Indicates the last measured temperature value (the most recent sample)
Level 2	DTEMP_TAG_x_1	Indicates the number of temperature measurements minus one (DTEMP_TEMPERATURE_x_0 – 1)	DTEMP_TEMPERATURE_x_1	Indicates the penultimate measured temperature value
Level 3	DTEMP_TAG_x_2	Indicates the number of temperature measurements minus two (DTEMP_TEMPERATURE_x_0 – 2)	DTEMP_TEMPERATURE_x_2	Indicates temperature value measured before DTEMP_TEMPERATURE_x_1
Level 4	DTEMP_TAG_x_3	Indicates the number of temperature measurements minus three (DTEMP_TEMPERATURE_x_0 – 3)	DTEMP_TEMPERATURE_x_3	Indicates temperature value measured before DTEMP_TEMPERATURE_x_2
Level 5	DTEMP_TAG_x_4	Indicates the number of temperature measurements minus four (DTEMP_TEMPERATURE_x_0 – 4)	DTEMP_TEMPERATURE_x_4	Indicates temperature value measured before DTEMP_TEMPERATURE_x_3 (the oldest sample)

Note

DTEMP_TAG_x_4 increments its value with one after each fifth temperature measurement.

DTEMP_TAG_x_3 increments its value with one after each fourth temperature measurement.

DTEMP_TAG_x_2 increments its value with one after each third temperature measurement.

DTEMP_TAG_x_1 increments its value with one after each second temperature measurement.

DTEMP_TAG_x_0 increments its value with one after each temperature measurement.

18.4.6.2.5 Other Thermal Management Related Registers

- **Controlling the FIFOs:**

  Software can stop a certain FIFO to update with new temperature and timestamp values when setting to 0x1 one of the FREEZE_x bits in the CTRL_CORE_BANDGAP_MASK_1 and CTRL_CORE_BANDGAP_MASK_2 registers. These 5 bits are automatically cleared by hardware after the FIFOs are cleared.

Each FIFO is cleared by setting to 0x1 one of the CLEAR_x bits in the CTRL_CORE_BANDGAP_MASK_1 and CTRL_CORE_BANDGAP_MASK_2 registers. These 5 bits are also automatically set by hardware to 0x0 after the FIFOs clearing procedure completes.

- **Controlling the clock provided to the five FSMs:**

  The five FSMs comply with the PRCM slave idle protocol. They share common functional clock (L3INSTR_TS_GCLK), which is automatically gated by PRCM depending on the value of the CTRL_CORE_BANDGAP_MASK_1[31:30] SIDLEMODE bit field. L3INSTR_TS_GCLK clock is also enabled automatically by the PRCM module.

18.4.6.2.6 Summary of the Thermal Management Related Registers

Table 18-9 summarizes all the thermal management related registers.

Table 18-9. Summary of the Thermal Management Related Registers

Register Name	Description	Access
CTRL_CORE_TEMP_SENSOR_MPU CTRL_CORE_TEMP_SENSOR_GPU CTRL_CORE_TEMP_SENSOR_CORE CTRL_CORE_TEMP_SENSOR_IVA CTRL_CORE_TEMP_SENSOR_DSPEVE	Temperature sensor control registers	RW
CTRL_CORE_BANDGAP_THRESHOLD_MPU CTRL_CORE_BANDGAP_THRESHOLD_GPU CTRL_CORE_BANDGAP_THRESHOLD_CORE CTRL_CORE_BANDGAP_THRESHOLD_IVA CTRL_CORE_BANDGAP_THRESHOLD_DSPEVE	Registers for the thermal alert comparators	RW
CTRL_CORE_BANDGAP_TSHUT_MPU CTRL_CORE_BANDGAP_TSHUT_GPU CTRL_CORE_BANDGAP_TSHUT_CORE CTRL_CORE_BANDGAP_TSHUT_IVA CTRL_CORE_BANDGAP_TSHUT_DSPEVE	Registers for the thermal shutdown comparators	RW

Table 18-9. Summary of the Thermal Management Related Registers (continued)

Register Name	Description	Access
CTRL_CORE_DTEMP_MPU_0 CTRL_CORE_DTEMP_MPU_1 CTRL_CORE_DTEMP_MPU_2 CTRL_CORE_DTEMP_MPU_3 CTRL_CORE_DTEMP_MPU_4	Temperature timestamp registers	RO
CTRL_CORE_DTEMP_GPU_0 CTRL_CORE_DTEMP_GPU_1 CTRL_CORE_DTEMP_GPU_2 CTRL_CORE_DTEMP_GPU_3 CTRL_CORE_DTEMP_GPU_4		
CTRL_CORE_DTEMP_CORE_0 CTRL_CORE_DTEMP_CORE_1 CTRL_CORE_DTEMP_CORE_2 CTRL_CORE_DTEMP_CORE_3 CTRL_CORE_DTEMP_CORE_4		
CTRL_CORE_DTEMP_IVA_0 CTRL_CORE_DTEMP_IVA_1 CTRL_CORE_DTEMP_IVA_2 CTRL_CORE_DTEMP_IVA_3 CTRL_CORE_DTEMP_IVA_4		
CTRL_CORE_DTEMP_DSPEVE_0 CTRL_CORE_DTEMP_DSPEVE_1 CTRL_CORE_DTEMP_DSPEVE_2 CTRL_CORE_DTEMP_DSPEVE_3 CTRL_CORE_DTEMP_DSPEVE_4		
CTRL_CORE_BANDGAP_STATUS_1 CTRL_CORE_BANDGAP_STATUS_2	Registers with status bits for the non masked comparator outputs and the thermal alert signal.	RO
CTRL_CORE_BANDGAP_MASK_1 CTRL_CORE_BANDGAP_MASK_2	Registers used to mask the comparator outputs for the low and high thermal alert thresholds. These registers are also used to control the FIFOs and the clock provided to the five FSMs.	RW

18.4.6.2.7 ADC Values Versus Temperature

Table 18-10 provides all the valid ADC values which correspond to the temperature measured which is read from the CTRL_CORE_TEMP_SENSOR_x[9:0] BGAP_DTEMP_x bit fields. Table 18-10 also provides the values for the low and high temperature thresholds which are configurable through the CTRL_CORE_BANDGAP_THRESHOLD_x[9:0] THOLD_COLD_x and CTRL_CORE_BANDGAP_THRESHOLD_x[25:16] THOLD_HOT_x bit fields.

Table 18-10. ADC Values Versus Temperature

ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature	
	From	To		From	To		From	To		From	To
0-539	Outside region of operation		641	0.8	1.2	743	43.4	44	845	85.2	85.6
540	-40	-40	642	1.2	1.6	744	44	44.4	846	85.6	86
541	-40	-40	643	1.6	2	745	44.4	44.8	847	86	86.4
542	-40	-40	644	2	2.4	746	44.8	45.2	848	86.4	86.8

Table 18-10. ADC Values Versus Temperature (continued)

ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature	
543	-40	-40	645	2.4	2.8	747	45.2	45.6	849	86.8	87.2
544	-40	-39.6	646	2.8	3.2	748	45.6	46	850	87.2	87.6
545	-39.6	-39.2	647	3.2	3.6	749	46	46.4	851	87.6	88
546	-39.2	-38.8	648	3.6	4.2	750	46.4	46.8	852	88	88.4
547	-38.8	-38.4	649	4.2	4.8	751	46.8	47.2	853	88.4	88.8
548	-38.4	-38	650	4.8	5.2	752	47.2	47.6	854	88.8	89.2
549	-38	-37.6	651	5.2	5.6	753	47.6	48	855	89.2	89.6
550	-37.6	-37.2	652	5.6	6	754	48	48.4	856	89.6	90
551	-37.2	-36.8	653	6	6.4	755	48.4	48.8	857	90	90.4
552	-36.8	-36.4	654	6.4	6.8	756	48.8	49.2	858	90.4	90.8
553	-36.4	-36	655	6.8	7.2	757	49.2	49.6	859	90.8	91.2
554	-36	-35.6	656	7.2	7.6	758	49.6	50	860	91.2	91.6
555	-35.6	-35	657	7.6	8	759	50	50.4	861	91.6	92
556	-35	-34.4	658	8	8.4	760	50.4	50.8	862	92	92.4
557	-34.4	-34	659	8.4	8.8	761	50.8	51.2	863	92.4	92.8
558	-34	-33.6	660	8.8	9.2	762	51.2	51.6	864	92.8	93.2
559	-33.6	-33.2	661	9.2	9.6	763	51.6	52	865	93.2	93.6
560	-33.2	-32.8	662	9.6	10	764	52	52.4	866	93.6	94
561	-32.8	-32.4	663	10	10.4	765	52.4	52.8	867	94	94.4
562	-32.4	-32	664	10.4	10.8	766	52.8	53.2	868	94.4	94.8
563	-32	-31.6	665	10.8	11.2	767	53.2	53.6	869	94.8	95.2
564	-31.6	-31.2	666	11.2	11.6	768	53.6	54	870	95.2	95.6
565	-31.2	-30.8	667	11.6	12	769	54	54.4	871	95.6	96
566	-30.8	-30.4	668	12	12.4	770	54.4	54.8	872	96	96.4
567	-30.4	-30	669	12.4	13	771	54.8	55.2	873	96.4	96.8
568	-30	-29.6	670	13	13.6	772	55.2	55.6	874	96.8	97.2
569	-29.6	-29.2	671	13.6	14	773	55.6	56.2	875	97.2	97.8
570	-29.2	-28.8	672	14	14.4	774	56.2	56.8	876	97.8	98.4
571	-28.8	-28.4	673	14.4	14.8	775	56.8	57.2	877	98.4	98.8
572	-28.4	-28	674	14.8	15.2	776	57.2	57.6	878	98.8	99.2
573	-28	-27.4	675	15.2	15.6	777	57.6	58	879	99.2	99.6
574	-27.4	-26.8	676	15.6	16	778	58	58.4	880	99.6	100
575	-26.8	-26.4	677	16	16.4	779	58.4	58.8	881	100	100.4
576	-26.4	-26	678	16.4	16.8	780	58.8	59.2	882	100.4	100.8
577	-26	-25.6	679	16.8	17.2	781	59.2	59.6	883	100.8	101.2
578	-25.6	-25.2	680	17.2	17.6	782	59.6	60	884	101.2	101.6
579	-25.2	-24.8	681	17.6	18	783	60	60.4	885	101.6	102
580	-24.8	-24.4	682	18	18.4	784	60.4	60.8	886	102	102.4
581	-24.4	-24	683	18.4	18.8	785	60.8	61.2	887	102.4	102.8
582	-24	-23.6	684	18.8	19.2	786	61.2	61.6	888	102.8	103.2
583	-23.6	-23.2	685	19.2	19.6	787	61.6	62	889	103.2	103.6
584	-23.2	-22.8	686	19.6	20	788	62	62.4	890	103.6	104
585	-22.8	-22.4	687	20	20.4	789	62.4	62.8	891	104	104.4
586	-22.4	-22	688	20.4	20.8	790	62.8	63.2	892	104.4	104.8

Table 18-10. ADC Values Versus Temperature (continued)

ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature	
587	-22	-21.6	689	20.8	21.2	791	63.2	63.6	893	104.8	105.2
588	-21.6	-21.2	690	21.2	21.6	792	63.6	64	894	105.2	105.6
589	-21.2	-20.8	691	21.6	22.2	793	64	64.4	895	105.6	106
590	-20.8	-20.2	692	22.2	22.8	794	64.4	64.8	896	106	106.4
591	-20.2	-19.6	693	22.8	23.2	795	64.8	65.2	897	106.4	106.8
592	-19.6	-19.2	694	23.2	23.6	796	65.2	65.6	898	106.8	107.2
593	-19.2	-18.8	695	23.6	24	797	65.6	66	899	107.2	107.6
594	-18.8	-18.4	696	24	24.4	798	66	66.4	900	107.6	108
595	-18.4	-18	697	24.4	24.8	799	66.4	66.8	901	108	108.4
596	-18	-17.6	698	24.8	25.2	800	66.8	67.2	902	108.4	108.8
597	-17.6	-17.2	699	25.2	25.6	801	67.2	67.6	903	108.8	109.2
598	-17.2	-16.8	700	25.6	26	802	67.6	68	904	109.2	109.6
599	-16.8	-16.4	701	26	26.4	803	68	68.4	905	109.6	110
600	-16.4	-16	702	26.4	26.8	804	68.4	68.8	906	110	110.4
601	-16	-15.6	703	26.8	27.2	805	68.8	69.2	907	110.4	110.8
602	-15.6	-15.2	704	27.2	27.6	806	69.2	69.6	908	110.8	111.2
603	-15.2	-14.8	705	27.6	28	807	69.6	70	909	111.2	111.6
604	-14.8	-14.4	706	28	28.4	808	70	70.4	910	111.6	112
605	-14.4	-14	707	28.4	28.8	809	70.4	70.8	911	112	112.4
606	-14	-13.6	708	28.8	29.2	810	70.8	71.2	912	112.4	112.8
607	-13.6	-13.2	709	29.2	29.6	811	71.2	71.8	913	112.8	113.2
608	-13.2	-12.8	710	29.6	30	812	71.8	72.4	914	113.2	113.6
609	-12.8	-12.2	711	30	30.4	813	72.4	72.8	915	113.6	114
610	-12.2	-11.6	712	30.4	30.8	814	72.8	73.2	916	114	114.4
611	-11.6	-11.2	713	30.8	31.2	815	73.2	73.6	917	114.4	114.8
612	-11.2	-10.8	714	31.2	31.6	816	73.6	74	918	114.8	115.2
613	-10.8	-10.4	715	31.6	32.2	817	74	74.4	919	115.2	115.6
614	-10.4	-10	716	32.2	32.8	818	74.4	74.8	920	115.6	116
615	-10	-9.6	717	32.8	33.2	819	74.8	75.2	921	116	116.4
616	-9.6	-9.2	718	33.2	33.6	820	75.2	75.6	922	116.4	116.8
617	-9.2	-8.8	719	33.6	34	821	75.6	76	923	116.8	117.2
618	-8.8	-8.4	720	34	34.4	822	76	76.4	924	117.2	117.6
619	-8.4	-8	721	34.4	34.8	823	76.4	76.8	925	117.6	118
620	-8	-7.6	722	34.8	35.2	824	76.8	77.2	926	118	118.4
621	-7.6	-7.2	723	35.2	35.6	825	77.2	77.6	927	118.4	118.8
622	-7.2	-6.8	724	35.6	36	826	77.6	78	928	118.8	119.2
623	-6.8	-6.4	725	36	36.4	827	78	78.4	929	119.2	119.6
624	-6.4	-6	726	36.4	36.8	828	78.4	78.8	930	119.6	120
625	-6	-5.6	727	36.8	37.2	829	78.8	79.2	931	120	120.4
626	-5.6	-5.2	728	37.2	37.6	830	79.2	79.6	932	120.4	120.8
627	-5.2	-4.8	729	37.6	38	831	79.6	80	933	120.8	121.2
628	-4.8	-4.2	730	38	38.4	832	80	80.4	934	121.2	121.6
629	-4.2	-3.6	731	38.4	38.8	833	80.4	80.8	935	121.6	122
630	-3.6	-3.2	732	38.8	39.2	834	80.8	81.2	936	122	122.4

Table 18-10. ADC Values Versus Temperature (continued)

ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature	
631	-3.2	-2.8	733	39.2	39.6	835	81.2	81.6	937	122.4	122.8
632	-2.8	-2.4	734	39.6	40	836	81.6	82	938	122.8	123.2
633	-2.4	-2	735	40	40.4	837	82	82.4	939	123.2	123.6
634	-2	-1.6	736	40.4	40.8	838	82.4	82.8	940	123.6	124
635	-1.6	-1.2	737	40.8	41.2	839	82.8	83.2	941	124	124.4
636	-1.2	-0.8	738	41.2	41.6	840	83.2	83.6	942	124.4	124.8
637	-0.8	-0.4	739	41.6	42	841	83.6	84	943	124.8	125
638	-0.4	0	740	42	42.4	842	84	84.4	945	125	125
639	0	0.4	741	42.4	42.8	843	84.4	84.8	946-1023	Outside region of operation	
640	0.4	0.8	742	42.8	43.4	844	84.8	85.2			

18.4.6.3 PBIAS Cell And MMC1 I/O Cells Control Registers

The PBIAS cell is associated with device MMC1 interface and used together with the MMC1 I/O cells. Its purpose is to provide bias voltage to the MMC1 I/O cells. Without this bias voltage these I/O cells can not function properly.

The PBIAS cell is controlled by software using the CTRL_CORE_CONTROL_PBIAS register. The CTRL_CORE_CONTROL_HYST_1 register is used for hysteresis and drive strength control of the MMC1 I/O cells.

Figure 18-8 shows the PBIAS cell with the control bits and connections between it, the control module and the MMC1 I/O cells.

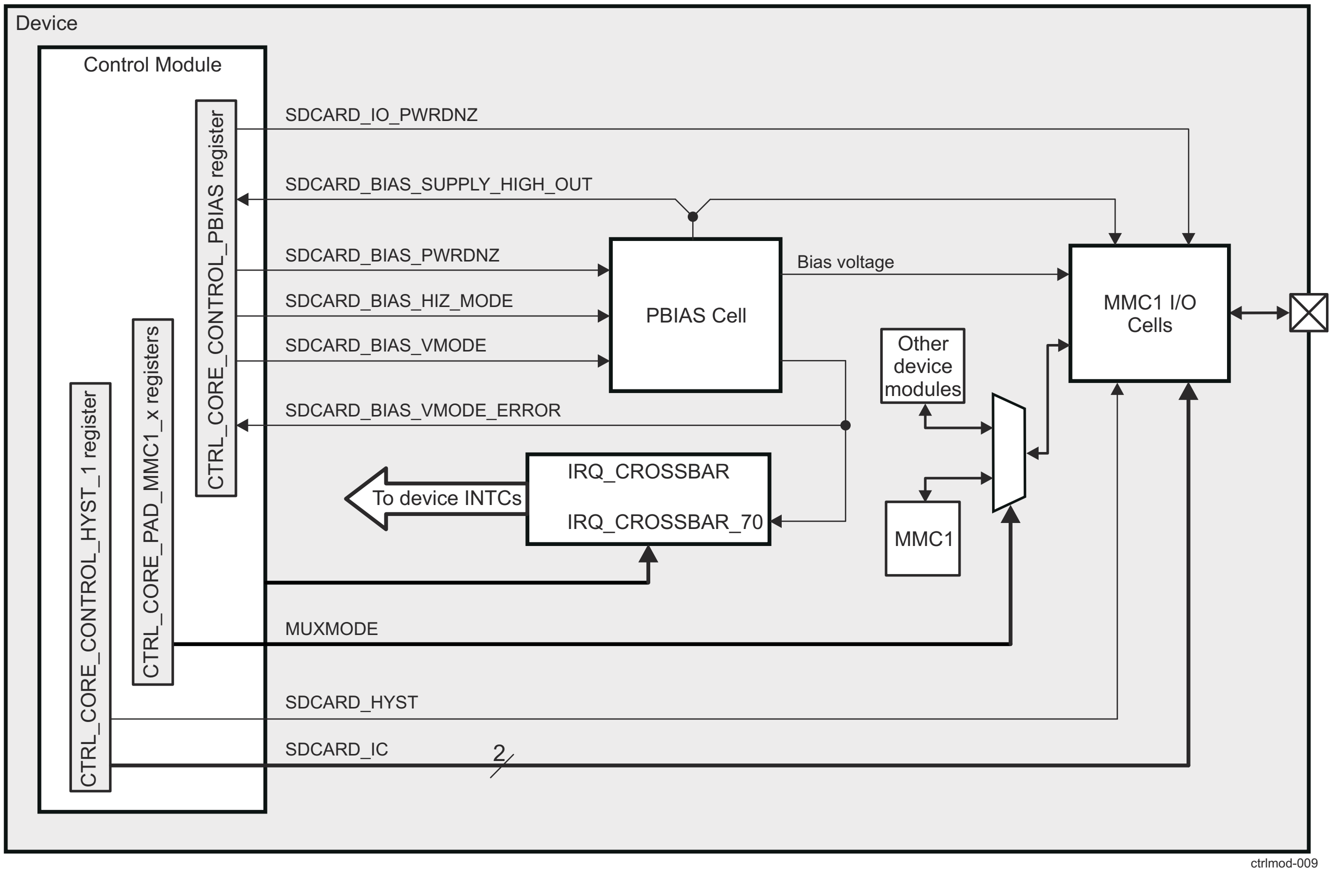


Figure 18-8. PBIAS Cell And Its Connections

Table 18-11 describes PBIAS cell and MMC1 I/O cells control bits.

Table 18-11. Control Bits For the PBIAS and MMC1 I/O Cells

Control bits for PBIAS cell and MMC1 I/O cells	Reset Value	Description
CTRL_CORE_CONTROL_PBIAS[27] SDCARD_BIAS_PWRDNZ	0x0	This bit turns ON and OFF the PBIAS cell. Software must keep it to 0x0 whenever the PBIAS cell supply voltage is ramping up/down or changing. Thus the PBIAS cell is protected. This bit should be set to 0x1 only after the PBIAS cell supply voltage is stable. This action power up the PBIAS cell.
CTRL_CORE_CONTROL_PBIAS[26] SDCARD_IO_PWRDNZ	0x0	This bit turns ON and OFF the MMC1 I/O cells. Software must keep it to 0x0 whenever the MMC1 IOs supply voltage is ramping up/down or changing. Thus the MMC1 I/O cells are protected. When this bit is set to 0x0, the MMC1 pads are floating. This bit should be set to 0x1 only after the MMC1 IOs supply voltage is stable. This action powers up the MMC1 I/O cells.
CTRL_CORE_CONTROL_PBIAS[25] SDCARD_BIAS_HIZ_MODE	0x0	When this bit is set to 0x1, the PBIAS cell output is in high impedance state and the SDCARD_BIAS_VMODE_ERROR bit sets automatically to 0x1. When SDCARD_BIAS_HIZ_MODE is set to 0x0, the PBIAS cell is in normal operation mode.
CTRL_CORE_CONTROL_PBIAS[24] SDCARD_BIAS_SUPPLY_HI_OUT	0x0	This is status bit indicating whether the MMC1 I/O cells supply voltage is equal to 1,8V or 3,3V.

Table 18-11. Control Bits For the PBIAS and MMC1 I/O Cells (continued)

Control bits for PBIAS cell and MMC1 I/O cells	Reset Value	Description
CTRL_CORE_CONTROL_PBIAS[23] SDCARD_BIAS_VMODE_ERROR	0x0	Status bit which indicates, during PBIAS cell normal operation mode, whether the voltage defined by the SDCARD_BIAS_VMODE bit is equal to the MMC1 I/Os supply voltage or not. If not, this bit is automatically set to 0x1 4µs after the voltage detection. It is also mapped to the IRQ_CROSSBAR_70 input line and is used as an interrupt source. If both voltage values are not equal an interrupt is generated.
CTRL_CORE_CONTROL_PBIAS[21] SDCARD_BIAS_VMODE	0x1	By controlling this bit software tells the PBIAS cell whether the MMC1 I/O cells supply voltage is equal to 1,8V or 3,3V. Its reset value indicates that the MMC1 I/Os voltage level is 3,3V.
CTRL_CORE_CONTROL_HYST_1[31] SDCARD_HYST	0x1	Hysteresis enabling/disabling for the MMC1 I/Os input buffer.
CTRL_CORE_CONTROL_HYST_1[30:29] SDCARD_IC	0x0	Impedance control (drive strength) for the MMC1 I/Os output buffer.

When the MMC1 signals are not used, that is MUXMODE different than 0x0 selected, the PBIAS cell must also be configured for proper work of the other interface signals multiplexed on the MMC1 I/O cells. For example, if MUXMODE = 0xE (gpio6_21 to gpio6_26 signals selected) the PBIAS cell and MMC1 I/O cells must be configured. This means that both the cells must be powerd and the appropriate PBAIS cell control bits must also be configured.

The MMC1 interface pads are the following:

- mmc1_clk
- mmc1_cmd
- mmc1_dat[3:0]
- mmc1_sdcd
- mmc1_sdwp

All of these pads except the mmc1_sdcd and mmc1_sdwp pads are associated with the PBIAS cell and the CTRL_CORE_CONTROL_PBIAS register and are also controlled by the CTRL_CORE_CONTROL_HYST_1 register.

The PBIAS cell and the MMC1 I/O cells are powered externally through the vddshv8 ball.

The PBIAS cell must be programmed according to the MMC1 I/O cells supply voltage. For details, see Table 18-12.

Table 18-12. PBIAS Cell Voltage Configuration

CTRL_CORE_CONTROL_PBIAS[21] SDCARD_BIAS_VMODE Bit Configuration(1)	PBIAS Cell and MMC1 I/O Cells Supply Voltage	Type of Operation
1.8V (0x0)	1.8V	Normal 1.8V operation
1.8V (0x0)	3.3V	Damaging configuration(2)
3.3V (0x1)	1.8V	Damaging configuration(2)
3.3V (0x1)	3.3V	Normal 3.3V operation

(1) For damaging configuration, hardware system protection is provided to prevent deterioration of the associated MMC1 I/Os.
(2) These modes must not be used.

Table 18-13 summarizes the generation of the CTRL_CORE_CONTROL_PBIAS[23] SDCARD_BIAS_VMODE_ERROR status flag, which depends on the various combinations of bits in the CTRL_CORE_CONTROL_PBIAS register. When this flag sets to 0x1, it is recommended the MMC1 I/O cells to be powered down by setting to 0x0 the CTRL_CORE_CONTROL_PBIAS[26] SDCARD_IO_PWRDNZ bit.

Table 18-13. PBIAS Cell Error Signal Truth Table

Programmed Voltage Level (SDCARD_BIAS_VMODE)	SDCARD_BIAS_SUPPLY_HI_OUT	SDCARD_BIAS_HIZ_MODE	PWRDNZ Bits	SDCARD_BIAS_VMODE_ERROR
0	0	X	0	0
0	0	0	1	0
0	1	0	1	1
X	X	1	1	1
1	0	X	0	0
1	0	0	1	1
1	1	0	1	0

SDCARD_BIAS_VMODE_ERROR = 0x1 shows that the programmed voltage level is not the same as the voltage indicated by the SDCARD_BIAS_SUPPLY_HI_OUT bit or high impedance mode is selected.

SDCARD_BIAS_VMODE_ERROR = 0x0 shows that the programmed voltage level is the same as the voltage indicated by the SDCARD_BIAS_SUPPLY_HI_OUT bit or it is not considered because SDCARD_PWRDNZ = 0x0.

18.4.6.4 IRQ_CROSSBAR Module Functional Description

There is an IRQ_CROSSBAR module in the device, which is controlled by registers in the CTRL_MODULE_CORE submodule. The IRQ_CROSSBAR is able to map any of its input signals to any of its outputs. This module is associated with the device interrupt sources. The IRQs from all the device modules are connected to the IRQ_CROSSBAR inputs. Each module IRQ is connected only to one cross-bar input. Each output of the IRQ_CROSSBAR module is connected only to one interrupt line of certain interrupt controller (INTC). Thus, the device IRQs are mapped to the device INTCs through the IRQ_CROSSBAR. Some of these IRQs are mapped by default to certain interrupt lines of one of the device INTCs, but there are IRQs which are not mapped by default to any interrupt line of any device INTC. All IRQs, connected to the IRQ_CROSSBAR inputs, can be remapped to other interrupt lines of the different device INTCs through the CTRL_CORE_X_IRQ_B_A registers. Each of these registers has a structure described in Table 18-14.

Table 18-14. Generic Description of the CTRL_CORE_X_IRQ_B_A IRQ_CROSSBAR Control Registers

Bits	Field Name	Description	Type	Note
31:25	RESERVED		R	
24:16	X_IRQ_A	Selects an interrupt source signal for the X_IRQ_A INTC line 0x0: Reserved 0x1: Maps IRQ_CROSSBAR input 1 to X_IRQ_A INTC line 0x2: Maps IRQ_CROSSBAR input 2 to X_IRQ_A INTC line 0x-: .... 0x64: Maps IRQ_CROSSBAR input 100 to X_IRQ_A INTC line 0x-: .... 0x190 to 0x1FF: Reserved	RW	X is summarization. It is equal to: • MPU • DSP1 • DSP2 • IPU1 • IPU2 • EVE1 • EVE2 X shows the module name to which interrupt controller all inputs of the IRQ_CROSSBAR module can be mapped. A is also summarization. It shows the number of the line for the corresponding INTC. A is equal to 0, 1, 2, ..., and so on, depending on the count of the INTC lines controlled by the IRQ_CROSSBAR module. For more details, see Table 18-15.
15:9	RESERVED		R	

Table 18-14. Generic Description of the CTRL_CORE_X_IRQ_B_A IRQ_CROSSBAR Control Registers (continued)

Bits	Field Name	Description	Type	Note
8:0	X_IRQ_B	Selects an interrupt source signal for the X_IRQ_B INTC line 0x0: Reserved 0x1: Maps IRQ_CROSSBAR input 1 to X_IRQ_B INTC line 0x2: Maps IRQ_CROSSBAR input 2 to X_IRQ_B INTC line 0x-: .... 0x64: Maps IRQ_CROSSBAR input 100 to X_IRQ_B INTC line 0x-: .... 0x190 to 0x1FF: Reserved	RW	B is summarization. It shows the number of the line for the corresponding INTC. B is equal to 0, 1, 2, ..., and so on, depending on the count of the INTC lines controlled by the IRQ_CROSSBAR module.

Figure 18-9 represents the way in which the IRQ_CROSSBAR module works. It shows the device modules and their IRQs connected to the IRQ_CROSSBAR inputs, the structure of the cross-bar and its outputs connected to the device INTCs.

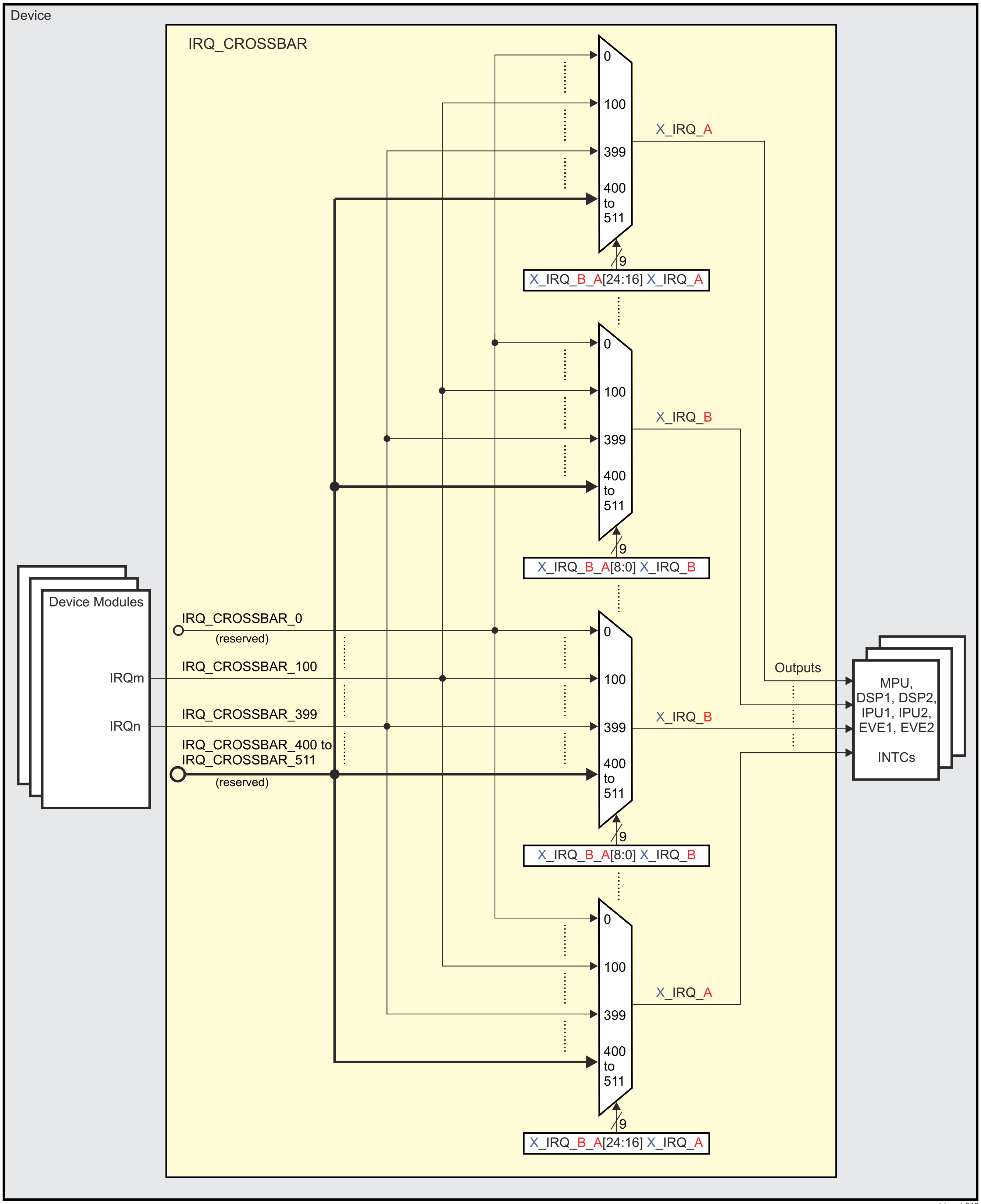


Figure 18-9. IRQ_CROSSBAR Module Functional Diagram

Each IRQ_CROSSBAR control register has two 9-bit fields. Each 9-bit field is associated only with one interrupt line. Through this 9-bit field any of the IRQs connected to the IRQ_CROSSBAR inputs can be mapped to the INTC line associated with this 9-bit field. For example, the register CTRL_CORE_MPU_IRQ_74_75 is associated with MPU_IRQ_74 and MPU_IRQ_75. The 9-bit field CTRL_CORE_MPU_IRQ_74_75[24:16] MPU_IRQ_75 is associated only with MPU_IRQ_75 interrupt line of the MPU INTC. Setting this bit field to any other value different than its reset value will map another module IRQ to the MPU_IRQ_75 interrupt line. The default (reset) value of this bit field is 0x46 which corresponds to the IRQ_CROSSBAR_70 input. The PBIAS_IRQ is connected to this cross-bar input. Setting another register to 0x46 will cause the PBIAS_IRQ to be mapped to another INTC line. For example, if the CTRL_CORE_DSP1_IRQ_40_41[8:0] DSP1_IRQ_40 is set to 0x46 the PBIAS_IRQ will be mapped to the DSP1_IRQ_40 interrupt line. The same logic also applies to the other interrupt lines of the device INTCs.

In addition, not all of the interrupt lines of a given INTC are controlled by the IRQ_CROSSBAR registers. This means, that there are interrupt lines which are not connected to any IRQ_CROSSBAR output and thus only one IRQ is connected to these interrupt lines. In the most common case these lines are used by interrupt sources internal or specific for the corresponding module, which has an INTC.

Table 18-15 shows which lines of each INTC are associated with the IRQ_CROSSBAR control registers. The rest of the INTC lines (not listed in the table) cannot be controlled by the cross-bar registers.

Table 18-15. Interrupt Lines Associated With The IRQ_CROSSBAR Control Registers

MPU INTC Lines	DSP INTC Lines	IPU INTC Lines	EVE INTC Lines
MPU_IRQ_4, MPU_IRQ_7 to MPU_IRQ_130, MPU_IRQ_133 to MPU_IRQ_159	DSP1_IRQ_32 to DSP1_IRQ_95, DSP2_IRQ_32 to DSP2_IRQ_95	IPU1_IRQ_23 to IPU1_IRQ_79, IPU2_IRQ_23 to IPU2_IRQ_79	EVE1_IRQ_0 to EVE1_IRQ_7, EVE2_IRQ_0 to EVE2_IRQ_7

The individual connection between all module IRQs and all IRQ_CROSSBAR inputs is shown in Section 17.3.8, *Mapping of Device Interrupts to IRQ_CROSSBAR Inputs* of Chapter 17, *Interrupt Controllers*.

In addition, the CTRL_CORE_OVS_IRQ_IO_MUX register is used to select for observation on two external pads any IRQ connected to the IRQ_CROSSBAR inputs. Using the CTRL_CORE_OVS_IRQ_IO_MUX[17:9] OVS_IRQ_IO_MUX_2 bit field all IRQs can be mapped to the obs_irq2 signal. The CTRL_CORE_OVS_IRQ_IO_MUX[8:0] OVS_IRQ_IO_MUX_1 bit field maps all IRQs to the obs_irq1 signal. For example, setting the CTRL_CORE_OVS_IRQ_IO_MUX[8:0] OVS_IRQ_IO_MUX_1 to 0x18 maps the GPIO1_IRQ_1 to the obs_irq1 line and thus this IRQ can be observed.

18.4.6.5 DMA_CROSSBAR Module Functional Description

There is a DMA_CROSSBAR module in the device, which is controlled by registers in the CTRL_MODULE_CORE submodule. The DMA_CROSSBAR is able to map any of its input signals to any of its outputs. This module is associated with the device DMA request source signals. The DREQs from all the device modules are connected to the DMA_CROSSBAR inputs. Each DREQ signal is connected only to one DMA cross-bar input. Each output of the DMA_CROSSBAR module is connected only to one input line of the device DMA modules. The DMAs associated with the DMA_CROSSBAR are the following:

- DMA_SYSTEM
- DMA_EDMA
- DMA_DSP1_EDMA
- DMA_DSP2_EDMA

All DREQs, connected to the DMA_CROSSBAR inputs, can be remapped to other lines of the these DMA modules through the CTRL_CORE_DMA_X_DREQ_B_A registers. Each of these registers has a structure described in Table 18-16.

Table 18-16. Generic Description of the CTRL_CORE_DMA_X_DREQ_B_A DMA_CROSSBAR Control Registers

Bits	Field Name	Description	Type	Note
31:24	RESERVED		R	

Table 18-16. Generic Description of the CTRL_CORE_DMA_X_DREQ_B_A DMA_CROSSBAR Control Registers (continued)

Bits	Field Name	Description	Type	Note
23:16	DMA_X_DREQ_A_IRQ_A	Selects a DMA request source signal for the DMA_X_DREQ_A_IRQ_A DMA line 0x0: Reserved 0x1: Reserved 0x2: Maps DMA_CROSSBAR input 2 to DMA_X_DREQ_A_IRQ_A DMA line 0x3: Maps DMA_CROSSBAR input 3 to DMA_X_DREQ_A_IRQ_A DMA line 0x-: .... 0x32: Maps DMA_CROSSBAR input 50 to DMA_X_DREQ_A_IRQ_A DMA line 0x-: .... 0xCC to 0xFF: Reserved	RW	X is summarization. It is equal to: • SYSTEM • EDMA • DSP1_EDMA • DSP2_EDMA X shows to which DMA all inputs of the DMA_CROSSBAR module can be mapped. A is also summarization. It shows the number of the DREQ line for the corresponding DMA module. A is equal to 0, 1, 2, ..., and so on, depending on the count of the DMA lines controlled by the DMA_CROSSBAR module. For more details, see Table 18-17.
15:8	RESERVED		R	
7:0	DMA_X_DREQ_B_IRQ_B	Selects a DMA request source signal for the DMA_X_DREQ_B_IRQ_B DMA line 0x0: Reserved 0x1: Reserved 0x2: Maps DMA_CROSSBAR input 2 to DMA_X_DREQ_B_IRQ_B DMA line 0x3: Maps DMA_CROSSBAR input 3 to DMA_X_DREQ_B_IRQ_B DMA line 0x-: .... 0x32: Maps DMA_CROSSBAR input 50 to DMA_X_DREQ_B_IRQ_B DMA line 0x-: .... 0xCC to 0xFF: Reserved	RW	B is also summarization. It shows the number of the DREQ line for the corresponding DMA module. B is equal to 0, 1, 2, ..., and so on, depending on the count of the DMA lines controlled by the DMA_CROSSBAR module. For more details, see Table 18-17.

Figure 18-10 represents the way in which the DMA_CROSSBAR module works. It shows the device modules and their DREQs connected to the DMA_CROSSBAR inputs, the structure of the cross-bar and its outputs connected to the device DMA modules.

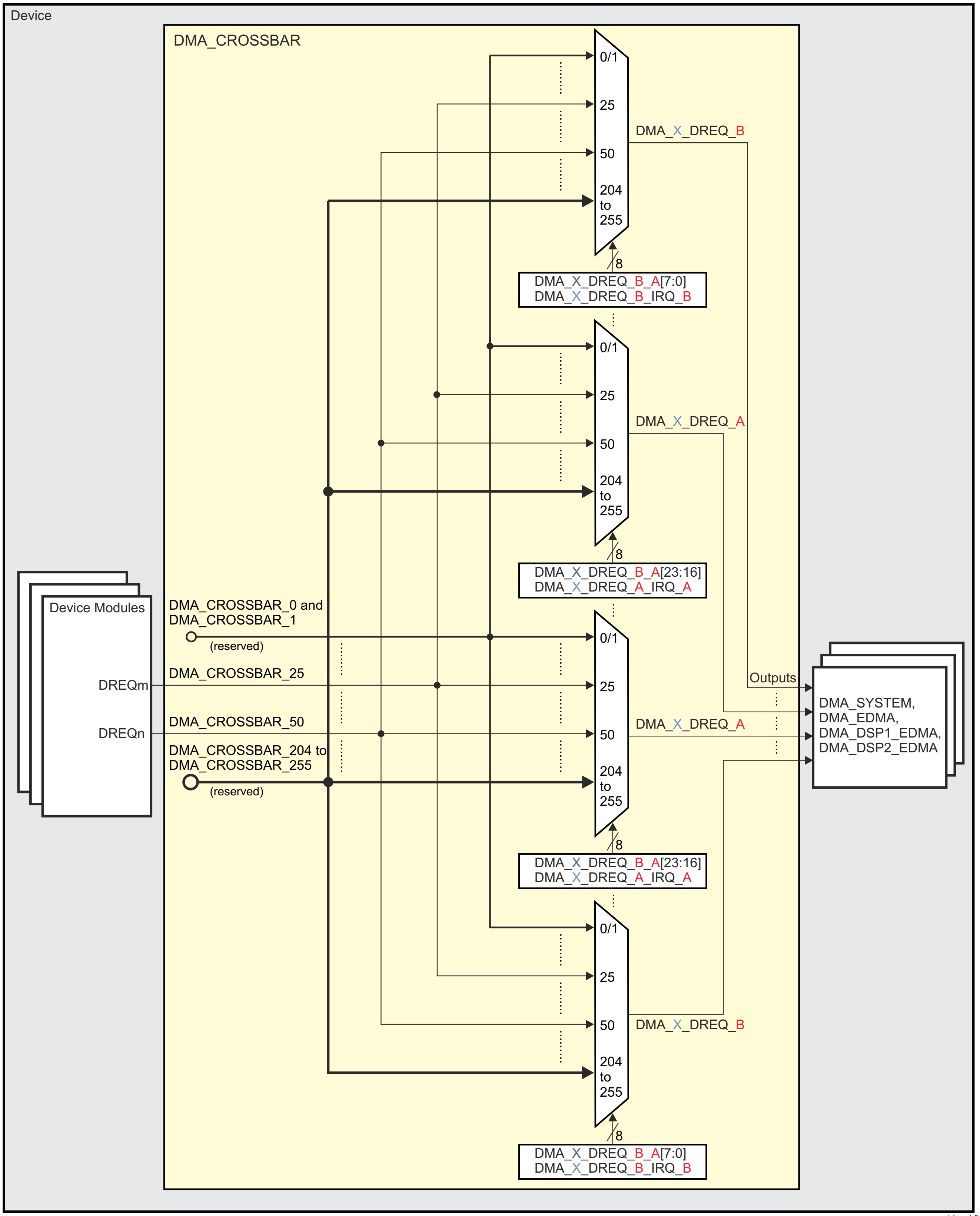


Figure 18-10. DMA_CROSSBAR Module Functional Diagram

Each DMA_CROSSBAR control register has two 8-bit fields. Each 8-bit field is associated only with one line of certain DMA module. Through this 8-bit field any of the DREQs connected to the DMA_CROSSBAR inputs can be mapped to the DMA line associated with this 8-bit field. For example, the register CTRL_CORE_DMA_EDMA_DREQ_62_63 is associated with DMA_EDMA_DREQ_62 and DMA_EDMA_DREQ_63 lines. The 8-bit field CTRL_CORE_DMA_EDMA_DREQ_62_63[7:0] DMA_EDMA_DREQ_62_IRQ_62 is associated only with DMA_EDMA_DREQ_62 line of the DMA_EDMA module. Setting this bit field to any other value different than its reset value will map another DREQ from the device modules to the DMA_EDMA_DREQ_62 line. The default (reset) value of this bit field is 0x3F which corresponds to the DMA_CROSSBAR_63 input. The UART5_DREQ_TX is connected to this cross-bar input. Setting another register to 0x3F will cause the UART5_DREQ_TX to be mapped to another DMA line. For example, if the CTRL_CORE_DMA_EDMA_DREQ_30_31[23:16] DMA_EDMA_DREQ_31_IRQ_31 is set to 0x3F the UART5_DREQ_TX will be mapped to the DMA_EDMA_DREQ_31 line. The same logic also applies to the other lines of the device DMAs.

Table 18-17 shows which lines of each DMA are associated with the DMA_CROSSBAR control registers. The rest of the lines (not listed in the table) cannot be controlled by the cross-bar registers.

Table 18-17. DREQ Lines Associated With The DMA_CROSSBAR Control Registers

DMA_SYSTEM DREQ Lines	DMA_EDMA DREQ Lines	DMA_DSP1_EDMA DREQ Lines	DMA_DSP2_EDMA DREQ Lines
DMA_SYSTEM_DREQ_0 to DMA_SYSTEM_DREQ_126	DMA_EDMA_DREQ_0 to DMA_EDMA_DREQ_63	DMA_DSP1_DREQ_0 to DMA_DSP1_DREQ_19	DMA_DSP2_DREQ_0 to DMA_DSP2_DREQ_19

The individual connection between all module DREQs and all DMA_CROSSBAR inputs is shown in Section 16.1.3.2, *Mapping of DMA Requests to DMA_CROSSBAR Inputs* of Section 16.1, *System DMA*.

In addition, the CTRL_CORE_OVS_DMARQ_IO_MUX register is used to select for observation on two external pads any DREQ connected to the DMA_CROSSBAR inputs. Using the CTRL_CORE_OVS_DMARQ_IO_MUX[15:8] OVS_DMARQ_IO_MUX_2 bit field all DREQs can be mapped to the obs_dmarq2 signal. The CTRL_CORE_OVS_DMARQ_IO_MUX[7:0] OVS_DMARQ_IO_MUX_1 bit field maps all DREQs to the obs_dmarq1 signal. For example, setting the CTRL_CORE_OVS_DMARQ_IO_MUX[7:0] OVS_DMARQ_IO_MUX_1 to 0x6 maps the DISPC_DREQ to the obs_dmarq1 line and thus this DREQ can be observed.

18.4.6.6 SDRAM Initiator Priority Registers

The CTRL_CORE_EMIF_INITIATOR_PRIORITY_1 to CTRL_CORE_EMIF_INITIATOR_PRIORITY_6 registers are intended to control the priority of each initiator accessing the two EMIFs. Each 3-bit field in these registers is associated only with one initiator. Setting this bit field to 0x0 means that the corresponding initiator has a highest priority over the others and setting it to 0x7 is for lowest priority. This feature is useful in case of concurrent access to the external SDRAM from several initiators.

Note

The priorities configured through the CTRL_CORE_EMIF_INITIATOR_PRIORITY_1 to CTRL_CORE_EMIF_INITIATOR_PRIORITY_6 registers have affect only at the L3 switch levels and are always overridden at DMM level by the priorities configured thorough the DMM_PEG_PRIO_k registers.

18.4.6.7 L3_MAIN Initiator Priority Registers

The CTRL_CORE_L3_INITIATOR_PRESSURE_1 to CTRL_CORE_L3_INITIATOR_PRESSURE_6 registers are used for controlling the priority of certain initiators on the L3_MAIN. Each 2-bit field in these registers is associated only with one initiator. Setting this bit field to 0x3 means that the traffic of this initiator has highest proiroty over the other traffics. A value of 0x0 is for lowest priority. Through these registers a dynamic priority escalation for the following L3_MAIN initiators is provided:

- MPU

- DSP1
- DSP2
- IPU1
- IPU2
- GPU P1
- GPU P2
- SATA
- MMC1
- MMC2
- USB1
- USB2
- USB3
- USB4

18.4.6.8 Memory Region Lock Registers

There are five registers used to lock different memory regions of CTRL_MODULE_CORE memory mapped space. A memory region is locked, means that all wite accesses to this region are ignored. Writing a value unique for each register will lock certain memory region and writing another unique value results in unlocking of the same region. These five registers can lock the entire memory space of the CTRL_MODULE_CORE submodule. Table 18-18 gives more details.

Table 18-18. Memory Region Lock Registers

Register	Memory Space to Lock	Groups of Registers Associated With This Memory Region	Lock/Unlock (register reset value)
CTRL_CORE_MMR_LOCK_1	Region from 0x0000 0100 to 0x0000 079F	Thermal management related registers, EMIF initiator priority, L3_MAIN initiator pressure (priority), standard eFuse and other registers	locked
CTRL_CORE_MMR_LOCK_2	Region from 0x0000 07A0 to 0x0000 0D9F	IRQ_CROSSBAR and DMA_CROSSBAR registers	locked
CTRL_CORE_MMR_LOCK_3	Region from 0x0000 0DA0 to 0x0000 0FFF	A few registers associated with device I/Os	locked
CTRL_CORE_MMR_LOCK_4	Region from 0x0000 1000 to 0x0000 13FF	Reserved range	locked
CTRL_CORE_MMR_LOCK_5	Region from 0x0000 1400 to 0x0000 1FFF	Mainly pad configuration registers	locked

Note

By default the entire CTRL_MODULE_CORE memory space is locked but the ROM code unlocks it by writing corresponding unlock values to all CTRL_CORE_MMR_LOCK_x registers.

18.4.6.9 NMI Mapping To Respective Cores

Two registers CTRL_CORE_NMI_DESTINATION_1 and CTRL_CORE_NMI_DESTINATION_2 are intended to map the external non-maskable interrupt (NMI) to certain of the device host processors. Writing 0x1 into a bit field of these registers enables the NMI to be mapped to the corresponding processor associated with this bit field. Writing 0x0 disables the NMI mapping to this processor.

18.4.6.10 Software Controls for the DDR2/DDR3 I/O Cells

There are two types of I/O cells associated with the DDR2/DDR3 interface. These are single-ended and differential I/O cells.

These I/O cells have the following software controls which reside in registers of the CTRL_MODULE_CORE:

- Output impedance controls - I[2:0]
- Slew rate controls - SR[2:0]

- Weak driver controls - WD[1:0]

The I and SR controls apply when the I/Os operate as outputs. The WD controls apply when the I/Os operate as both inputs or outputs.

The bits I[2:0] are used for programming the desired impedance value of the output buffer. Table 18-19 describes the I[2:0] controls which are valid for pull-up and pull-down outputs.

Table 18-19. Output Impedance Controls - I[2:0]

I[2]	I[1]	I[0]	Drive Setting Name	Output Impedance
0	0	0	Imp80	80 Ohms
0	0	1	Imp60	60 Ohms
0	1	0	Imp48	48 Ohms
0	1	1	Imp40	40 Ohms
1	0	0	Imp34	34 Ohms
1	0	1	Reserved	Reserved
1	1	0	Reserved	Reserved
1	1	1	Reserved	Reserved

To achieve optimal noise/speed trade-off the slew rate of the output signal can also be programmed using the slew rate control bits SR[2:0] shown in Table 18-20. These SR settings do not affect the DC drive strength of the output buffer. They only control its turn-on time.

Table 18-20. Slew Rate Controls - SR[2:0]

SR[2]	SR[1]	SR[0]	Turn-On Time Level	Note
0	0	0	fastest	All 8 values are valid.
...	...	...		
...	...	...		
1	1	1	slowest	

Weak pull-up, pull-down or keeper option for device DDR2/3 pads is enabled through the WD[1:0] bits. The weak pull-up or pull-down option is used to define the pad state (high or low) when no signal is driving the pad. The weak keeper option is used to maintain the previous output value when nothing is driving the pad. Table 18-21 describes the WD controls. They are used for avoiding floating pads on the DDR2/DDR3 interface.

Table 18-21. Weak Driver Controls - WD[1:0]

WD[1]	WD[0]	Single-Ended Operation	Differential Pair Operation	
			padp	padn
0	0	Pull logic disabled	Pull logic disabled	Pull logic disabled
0	1	Weak pullup enabled	Weak pullup enabled	Weak pulldown enabled
1	0	Weak pulldown enabled	Weak pulldown enabled	Weak pullup enabled
1	1	Weak keeper enabled	Weak keeper enabled	Weak keeper enabled

Note

To avoid unnecessary power consumption, software must overwrite the Weak Driver (WD) reset values by setting them to 0x0.

It must be taken into account that the I, SR and WD software controls apply for several pads combined in groups, and not for a single pad. For example, writing 0x1 to the CTRL_CORE_CONTROL_DDRCACH1_0[17:16] DDR3CH1_PART5A_WD bit field enables the weak pull-up resistors for all 16 pads in the PART5A group. These are the ddr1_a[15:0] pads. Table 18-22 shows the I, SR and WD controls for the different DDR2/DDR3 pad groups.

Table 18-22. Software Group Controls for the DDR2/DDR3 Pads

DDR2/DDR3 Interface I/O Group Controls	Group Name	Pads in a Group
EMIF1 Pads		
CTRL_CORE_CONTROL_DDRCACH1_0[31:29] DDR3CH1_PART0_I CTRL_CORE_CONTROL_DDRCACH1_0[28:26] DDR3CH1_PART0_SR CTRL_CORE_CONTROL_DDRCACH1_0[25:24] DDR3CH1_PART0_WD	PART0	ddr1_casn, ddr1_rasn, ddr1_rst, ddr1_wen, ddr1_csn[0], ddr1_cke, ddr1_odt[0]
CTRL_CORE_CONTROL_DDRCACH1_0[23:21] DDR3CH1_PART5A_I CTRL_CORE_CONTROL_DDRCACH1_0[20:18] DDR3CH1_PART5A_SR CTRL_CORE_CONTROL_DDRCACH1_0[17:16] DDR3CH1_PART5A_WD	PART5A	ddr1_a[15:0]
CTRL_CORE_CONTROL_DDRCACH1_0[15:13] DDR3CH1_PART5B_I CTRL_CORE_CONTROL_DDRCACH1_0[12:10] DDR3CH1_PART5B_SR CTRL_CORE_CONTROL_DDRCACH1_0[9:8] DDR3CH1_PART5B_WD	PART5B	ddr1_ba[0], ddr1_ba[1], ddr1_ba[2]
CTRL_CORE_CONTROL_DDRCACH1_0[7:5] DDR3CH1_PART6_I CTRL_CORE_CONTROL_DDRCACH1_0[4:2] DDR3CH1_PART6_SR CTRL_CORE_CONTROL_DDRCACH1_0[1:0] DDR3CH1_PART6_WD	PART6	ddr1_ck, ddr1_nck
CTRL_CORE_CONTROL_DDRCH1_0[31:29] DDRCH1_PART1A_I CTRL_CORE_CONTROL_DDRCH1_0[28:26] DDRCH1_PART1A_SR CTRL_CORE_CONTROL_DDRCH1_0[25:24] DDRCH1_PART1A_WD	PART1A	ddr1_d[7:0], ddr1_dqm[0]
CTRL_CORE_CONTROL_DDRCH1_0[23:21] DDRCH1_PART1B_I CTRL_CORE_CONTROL_DDRCH1_0[20:18] DDRCH1_PART1B_SR CTRL_CORE_CONTROL_DDRCH1_0[17:16] DDRCH1_PART1B_WD	PART1B	ddr1_dqs[0], ddr1_dqsn[0]
CTRL_CORE_CONTROL_DDRCH1_0[15:13] DDRCH1_PART2A_I CTRL_CORE_CONTROL_DDRCH1_0[12:10] DDRCH1_PART2A_SR CTRL_CORE_CONTROL_DDRCH1_0[9:8] DDRCH1_PART2A_WD	PART2A	ddr1_d[15:8], ddr1_dqm[1]
CTRL_CORE_CONTROL_DDRCH1_0[7:5] DDRCH1_PART2B_I CTRL_CORE_CONTROL_DDRCH1_0[4:2] DDRCH1_PART2B_SR CTRL_CORE_CONTROL_DDRCH1_0[1:0] DDRCH1_PART2B_WD	PART2B	ddr1_dqs[1], ddr1_dqsn[1]

Table 18-22. Software Group Controls for the DDR2/DDR3 Pads (continued)

DDR2/DDR3 Interface I/O Group Controls	Group Name	Pads in a Group
EMIF1 Pads		
CTRL_CORE_CONTROL_DDRCH1_1[31:29] DDRCH1_PART3A_I CTRL_CORE_CONTROL_DDRCH1_1[28:26] DDRCH1_PART3A_SR CTRL_CORE_CONTROL_DDRCH1_1[25:24] DDRCH1_PART3A_WD	PART3A	ddr1_d[23:16], ddr1_dqm[2]
CTRL_CORE_CONTROL_DDRCH1_1[23:21] DDRCH1_PART3B_I CTRL_CORE_CONTROL_DDRCH1_1[20:18] DDRCH1_PART3B_SR CTRL_CORE_CONTROL_DDRCH1_1[17:16] DDRCH1_PART3B_WD	PART3B	ddr1_dqs[2], ddr1_dqsn[2]
CTRL_CORE_CONTROL_DDRCH1_1[15:13] DDRCH1_PART4A_I CTRL_CORE_CONTROL_DDRCH1_1[12:10] DDRCH1_PART4A_SR CTRL_CORE_CONTROL_DDRCH1_1[9:8] DDRCH1_PART4A_WD	PART4A	ddr1_d[31:24], ddr1_dqm[3]
CTRL_CORE_CONTROL_DDRCH1_1[7:5] DDRCH1_PART4B_I CTRL_CORE_CONTROL_DDRCH1_1[4:2] DDRCH1_PART4B_SR CTRL_CORE_CONTROL_DDRCH1_1[1:0] DDRCH1_PART4B_WD	PART4B	ddr1_dqs[3], ddr1_dqsn[3]
CTRL_CORE_CONTROL_DDRCH1_2[23:21] DDRCH1_PART7A_I CTRL_CORE_CONTROL_DDRCH1_2[20:18] DDRCH1_PART7A_SR CTRL_CORE_CONTROL_DDRCH1_2[17:16] DDRCH1_PART7A_WD	PART7A	ddr1_ecc_d[7:0], ddr1_dqm_ecc
CTRL_CORE_CONTROL_DDRCH1_2[15:13] DDRCH1_PART7B_I CTRL_CORE_CONTROL_DDRCH1_2[12:10] DDRCH1_PART7B_SR CTRL_CORE_CONTROL_DDRCH1_2[9:8] DDRCH1_PART7B_WD	PART7B	ddr1_dqs_ecc, ddr1_dqsn_ecc
EMIF2 Pads		
CTRL_CORE_CONTROL_DDRCACH2_0[31:29] DDR3CH2_PART0_I CTRL_CORE_CONTROL_DDRCACH2_0[28:26] DDR3CH2_PART0_SR CTRL_CORE_CONTROL_DDRCACH2_0[25:24] DDR3CH2_PART0_WD	PART0	ddr2_casn, ddr2_rasn, ddr2_rst, ddr2_wen, ddr2_csn[0], ddr2_cke, ddr2_odt[0]
CTRL_CORE_CONTROL_DDRCACH2_0[23:21] DDR3CH2_PART5A_I CTRL_CORE_CONTROL_DDRCACH2_0[20:18] DDR3CH2_PART5A_SR CTRL_CORE_CONTROL_DDRCACH2_0[17:16] DDR3CH2_PART5A_WD	PART5A	ddr2_a[15:0]

Table 18-22. Software Group Controls for the DDR2/DDR3 Pads (continued)

DDR2/DDR3 Interface I/O Group Controls	Group Name	Pads in a Group
EMIF1 Pads		
CTRL_CORE_CONTROL_DDRCACH2_0[15:13] DDR3CH2_PART5B_I CTRL_CORE_CONTROL_DDRCACH2_0[12:10] DDR3CH2_PART5B_SR CTRL_CORE_CONTROL_DDRCACH2_0[9:8] DDR3CH2_PART5B_WD	PART5B	ddr2_ba[0], ddr2_ba[1], ddr2_ba[2]
CTRL_CORE_CONTROL_DDRCACH2_0[7:5] DDR3CH2_PART6_I CTRL_CORE_CONTROL_DDRCACH2_0[4:2] DDR3CH2_PART6_SR CTRL_CORE_CONTROL_DDRCACH2_0[1:0] DDR3CH2_PART6_WD	PART6	ddr2_ck, ddr2_nck
CTRL_CORE_CONTROL_DDRCH2_0[31:29] DDRCH2_PART1A_I CTRL_CORE_CONTROL_DDRCH2_0[28:26] DDRCH2_PART1A_SR CTRL_CORE_CONTROL_DDRCH2_0[25:24] DDRCH2_PART1A_WD	PART1A	ddr2_d[7:0], ddr2_dqm[0]
CTRL_CORE_CONTROL_DDRCH2_0[23:21] DDRCH2_PART1B_I CTRL_CORE_CONTROL_DDRCH2_0[20:18] DDRCH2_PART1B_SR CTRL_CORE_CONTROL_DDRCH2_0[17:16] DDRCH2_PART1B_WD	PART1B	ddr2_dqs[0], ddr2_dqsn[0]
CTRL_CORE_CONTROL_DDRCH2_0[15:13] DDRCH2_PART2A_I CTRL_CORE_CONTROL_DDRCH2_0[12:10] DDRCH2_PART2A_SR CTRL_CORE_CONTROL_DDRCH2_0[9:8] DDRCH2_PART2A_WD	PART2A	ddr2_d[15:8], ddr2_dqm[1]
CTRL_CORE_CONTROL_DDRCH2_0[7:5] DDRCH2_PART2B_I CTRL_CORE_CONTROL_DDRCH2_0[4:2] DDRCH2_PART2B_SR CTRL_CORE_CONTROL_DDRCH2_0[1:0] DDRCH2_PART2B_WD	PART2B	ddr2_dqs[1], ddr2_dqsn[1]
CTRL_CORE_CONTROL_DDRCH2_1[31:29] DDRCH2_PART3A_I CTRL_CORE_CONTROL_DDRCH2_1[28:26] DDRCH2_PART3A_SR CTRL_CORE_CONTROL_DDRCH2_1[25:24] DDRCH2_PART3A_WD	PART3A	ddr2_d[23:16], ddr2_dqm[2]
CTRL_CORE_CONTROL_DDRCH2_1[23:21] DDRCH2_PART3B_I CTRL_CORE_CONTROL_DDRCH2_1[20:18] DDRCH2_PART3B_SR CTRL_CORE_CONTROL_DDRCH2_1[17:16] DDRCH2_PART3B_WD	PART3B	ddr2_dqs[2], ddr2_dqsn[2]

Table 18-22. Software Group Controls for the DDR2/DDR3 Pads (continued)

DDR2/DDR3 Interface I/O Group Controls	Group Name	Pads in a Group
	EMIF1 Pads	
CTRL_CORE_CONTROL_DDRCH2_1[15:13] DDRCH2_PART4A_I CTRL_CORE_CONTROL_DDRCH2_1[12:10] DDRCH2_PART4A_SR CTRL_CORE_CONTROL_DDRCH2_1[9:8] DDRCH2_PART4A_WD	PART4A	ddr2_d[31:24], ddr2_dqm[3]
CTRL_CORE_CONTROL_DDRCH2_1[7:5] DDRCH2_PART4B_I CTRL_CORE_CONTROL_DDRCH2_1[4:2] DDRCH2_PART4B_SR CTRL_CORE_CONTROL_DDRCH2_1[1:0] DDRCH2_PART4B_WD	PART4B	ddr2_dqs[3], ddr2_dqsn[3]

18.4.6.11 Reference Voltage for the Device DDR2/DDR3 Receivers

The device DDR2/DDR3 input buffers work in so-called Vref-based receiver mode. In this mode, the buffers act like differential comparators with positive terminal connected to a device pad which receives signals from DDR2/DDR3 memory and negative terminal connected to a source of reference voltage.

To work properly, a reference voltage must be provided to the device DDR2/DDR3 input buffers. There are two Vref-generation cells per memory channel in the device intended to supply this internal reference voltage.

Figure 18-11 shows the Vref-generation cells in the device.

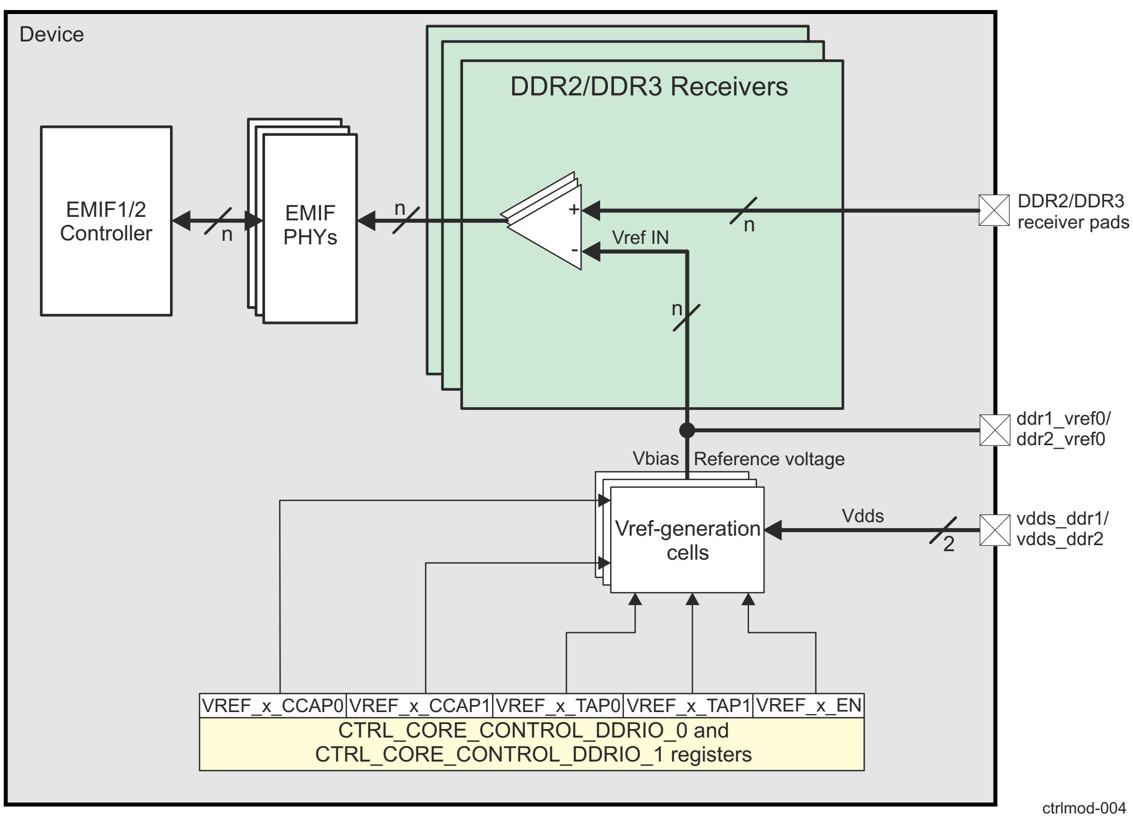


Figure 18-11. Vref-Generation Cells and Their Controls

Both the Vref-generation cells associated with EMIF1 are powerd through the vdds_ddr1 ball. Both the Vref-generation cells associated with EMIF2 are powerd through the vdds_ddr2 ball. For more information, see the device data manual.

The control bits for the Vref-generation cells reside in the CTRL_CORE_CONTROL_DDRIO_0 and CTRL_CORE_CONTROL_DDRIO_1 registers. There are VREF_x_TAP[1:0] control bits which set the output drive capability of the Vref cells. Table 18-23 lists the possible options for selection of load current sourced from the output of each Vref-generation cell.

Table 18-23. Vref Cell Load Current Selection

VREF_x_TAP1	VREF_x_TAP0	Description
0	0	2-μA load current
0	1	4-μA load current
1	0	8-μA load current
1	1	32-μA load current

According to the noise environment, the user can choose to filter the supplied reference voltage. Two coupling capacitors internal to each Vref-generation cell are available and configurable through the VREF_x_CCAP[1:0] control bits in the CTRL_CORE_CONTROL_DDRIO_0 and CTRL_CORE_CONTROL_DDRIO_1 registers, as specified in Table 18-24.

Table 18-24. Vref Cell Coupling Capacitor Selection

VREF_x_CCAP1	VREF_x_CCAP0	Capacitor
0	0	No capacitor connected.
0	1	One capacitor connected between Vbias and ground.(1)
1	0	One capacitor connected between Vbias and Vdds.(2)
1	1	One capacitor connected between Vbias and ground and one capacitor connected between Vbias and Vdds.

(1) Vbias is the output of the Vref-generation cell which provides the reference voltage.
(2) Vdds is the power supply voltage of the Vref-generation cell.

The Vref-gereration cells can be enabled by setting to 0x1 the VREF_x_EN bits in the CTRL_CORE_CONTROL_DDRIO_0 and CTRL_CORE_CONTROL_DDRIO_1 registers. These cells can be disabled (for leakage improvement and when not in use) by clearing the same VREF_x_EN bits.

Table 18-25 shows the Vref-generation cells control bits and the DDR2/DDR3 pads used as receivers to which the corresponding Vref cell supplies reference voltage.

Table 18-25. Controls for the Vref-Generation Cells Versus DDR2/DDR3 Receiver Pads

Vref-generation Cell Control Bits for Memory Channel 1	DDR2/DDR3 Vref Cell Associated Pads Used as Receivers (pads for memory channel 1)
CTRL_CORE_CONTROL_DDRIO_0[19] DDRCH1_VREF_DQ0_INT_CCAP0	ddr1_d[7:0], ddr1_d[15:8]
CTRL_CORE_CONTROL_DDRIO_0[18] DDRCH1_VREF_DQ0_INT_CCAP1	
CTRL_CORE_CONTROL_DDRIO_0[17] DDRCH1_VREF_DQ0_INT_TAP0	
CTRL_CORE_CONTROL_DDRIO_0[16] DDRCH1_VREF_DQ0_INT_TAP1	
CTRL_CORE_CONTROL_DDRIO_0[15] DDRCH1_VREF_DQ0_INT_EN	

Table 18-25. Controls for the Vref-Generation Cells Versus DDR2/DDR3 Receiver Pads (continued)

CTRL_CORE_CONTROL_DDRIO_0[14] DDRCH1_VREF_DQ1_INT_CCAP0	ddr1_d[23:16], ddr1_d[31:24], ddr1_ecc_d[7:0]
CTRL_CORE_CONTROL_DDRIO_0[13] DDRCH1_VREF_DQ1_INT_CCAP1	
CTRL_CORE_CONTROL_DDRIO_0[12] DDRCH1_VREF_DQ1_INT_TAP0	
CTRL_CORE_CONTROL_DDRIO_0[11] DDRCH1_VREF_DQ1_INT_TAP1	
CTRL_CORE_CONTROL_DDRIO_0[10] DDRCH1_VREF_DQ1_INT_EN	
Vref-generation Cell Control Bits for Memory Channel 2	**DDR2/DDR3 Vref Cell Associated Pads Used as Receivers (pads for memory channel 2)**
CTRL_CORE_CONTROL_DDRIO_1[26] DDRCH2_VREF_DQ0_INT_CCAP0	ddr2_d[7:0], ddr2_d[15:8]
CTRL_CORE_CONTROL_DDRIO_1[25] DDRCH2_VREF_DQ0_INT_CCAP1	
CTRL_CORE_CONTROL_DDRIO_1[24] DDRCH2_VREF_DQ0_INT_TAP0	
CTRL_CORE_CONTROL_DDRIO_1[23] DDRCH2_VREF_DQ0_INT_TAP1	
CTRL_CORE_CONTROL_DDRIO_1[22] DDRCH2_VREF_DQ0_INT_EN	
CTRL_CORE_CONTROL_DDRIO_1[21] DDRCH2_VREF_DQ1_INT_CCAP0	ddr2_d[23:16], ddr2_d[31:24]
CTRL_CORE_CONTROL_DDRIO_1[20] DDRCH2_VREF_DQ1_INT_CCAP1	
CTRL_CORE_CONTROL_DDRIO_1[19] DDRCH2_VREF_DQ1_INT_TAP0	
CTRL_CORE_CONTROL_DDRIO_1[18] DDRCH2_VREF_DQ1_INT_TAP1	
CTRL_CORE_CONTROL_DDRIO_1[17] DDRCH2_VREF_DQ1_INT_EN	

18.4.6.12 AVS Class 0 Associated Registers

AVS Class 0 attempts to normalize the power consumption across all devices by lowering the operating voltage of certain voltage rails. This procedure of lowering the voltage should be performed in the boot loader after ROM code. The new voltage to be set for each AVS Class 0 supported voltage rail should be read from the eFuse using dedicated registers. Then the power supply output voltage is adjusted to this new voltage value.

The following voltage rails support AVS Class 0:

- vdd_iva
- vdd_dspeve
- vdd
- vdd_gpu
- vdd_mpu

Note

For descriptions of the voltage rails previously listed see the *"Power Supply Signal Descriptions"* table in the device Data Manual.

Table 18-26 shows all registers associated with AVS Class 0. The corresponding AVS Class 0 voltage value can be read from the 12 LSbits of each of the listed registers. They contain the voltage value in hex format. To find the actual value in mV a conversion from hex to decimal value is needed. For example, if the value read from CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_2[11:0] STD_FUSE_OPP_VMIN_MPU_2 is 0x041F, then this corresponds to 1055 mV.

Table 18-26. Registers Associated With AVS Class 0 Voltage

Physical Address	Bit Field Containing the AVS Class 0 Voltage Value	Voltage Rail	Supported OPP
0x4A00 25CC	CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_2[11:0] STD_FUSE_OPP_VMIN_IVA_2	vdd_iva	OPP_NOM
0x4A00 25D0	CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_3[11:0] STD_FUSE_OPP_VMIN_IVA_3		OPP_OD
0x4A00 25D4	CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_4[11:0] STD_FUSE_OPP_VMIN_IVA_4		OPP_HIGH
0x4A00 25C4	CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_5[11:0] STD_FUSE_OPP_VMIN_IVA_5		OPP_PLUS
0x4A00 25E0	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2[11:0] STD_FUSE_OPP_VMIN_DSPEVE_2	vdd_dspeve	OPP_NOM
0x4A00 25E4	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3[11:0] STD_FUSE_OPP_VMIN_DSPEVE_3		OPP_OD
0x4A00 25E8	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4[11:0] STD_FUSE_OPP_VMIN_DSPEVE_4		OPP_HIGH
0x4A00 25D8	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5[11:0] STD_FUSE_OPP_VMIN_DSPEVE_5		OPP_PLUS
0x4A00 25F4	CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2[11:0] STD_FUSE_OPP_VMIN_CORE_2	vdd	OPP_NOM
0x4A00 3B08	CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_2[11:0] STD_FUSE_OPP_VMIN_GPU_2	vdd_gpu	OPP_NOM
0x4A00 3B0C	CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_3[11:0] STD_FUSE_OPP_VMIN_GPU_3		OPP_OD
0x4A00 3B10	CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_4[11:0] STD_FUSE_OPP_VMIN_GPU_4		OPP_HIGH
0x4A00 3B14	CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_5[11:0] STD_FUSE_OPP_VMIN_GPU_5		OPP_PLUS
0x4A00 3B1C	CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_1[11:0] STD_FUSE_OPP_VMIN_MPU_1	vdd_mpu	OPP_LOW
0x4A00 3B20	CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_2[11:0] STD_FUSE_OPP_VMIN_MPU_2		OPP_NOM
0x4A00 3B24	CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_3[11:0] STD_FUSE_OPP_VMIN_MPU_3		OPP_OD
0x4A00 3B28	CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_4[11:0] STD_FUSE_OPP_VMIN_MPU_4		OPP_HIGH
0x4A00 3B2C	CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_5[11:0] STD_FUSE_OPP_VMIN_MPU_5		OPP_PLUS

Some of the OPPs listed in Table 18-26 may not be supported for some devices. In these cases the voltage values in the corresponding AVS Class 0 registers can be disregarded.

Note

For more information about the supported OPPs, see the "*Operating Performance Points*" section of the device Data Manual.

In some cases, the AVS Class 0 voltage that is read from the CTRL_CORE_STD_FUSE_OPP_VMIN_xxx_y registers has a value between two incremental voltage steps of the power supply. If such a case occurs, the higher voltage value should be selected.

If several AVS Class 0 supported voltage rails are combined with each other, then all corresponding registers should be read and the highest value should be selected. The power supply of the combined rails should be changed to this highest voltage value.

Note

For a list of the supported voltage rail combinations, see the device Data Manual.

Figure 18-12 shows a general example of how AVS Class 0 should be performed.

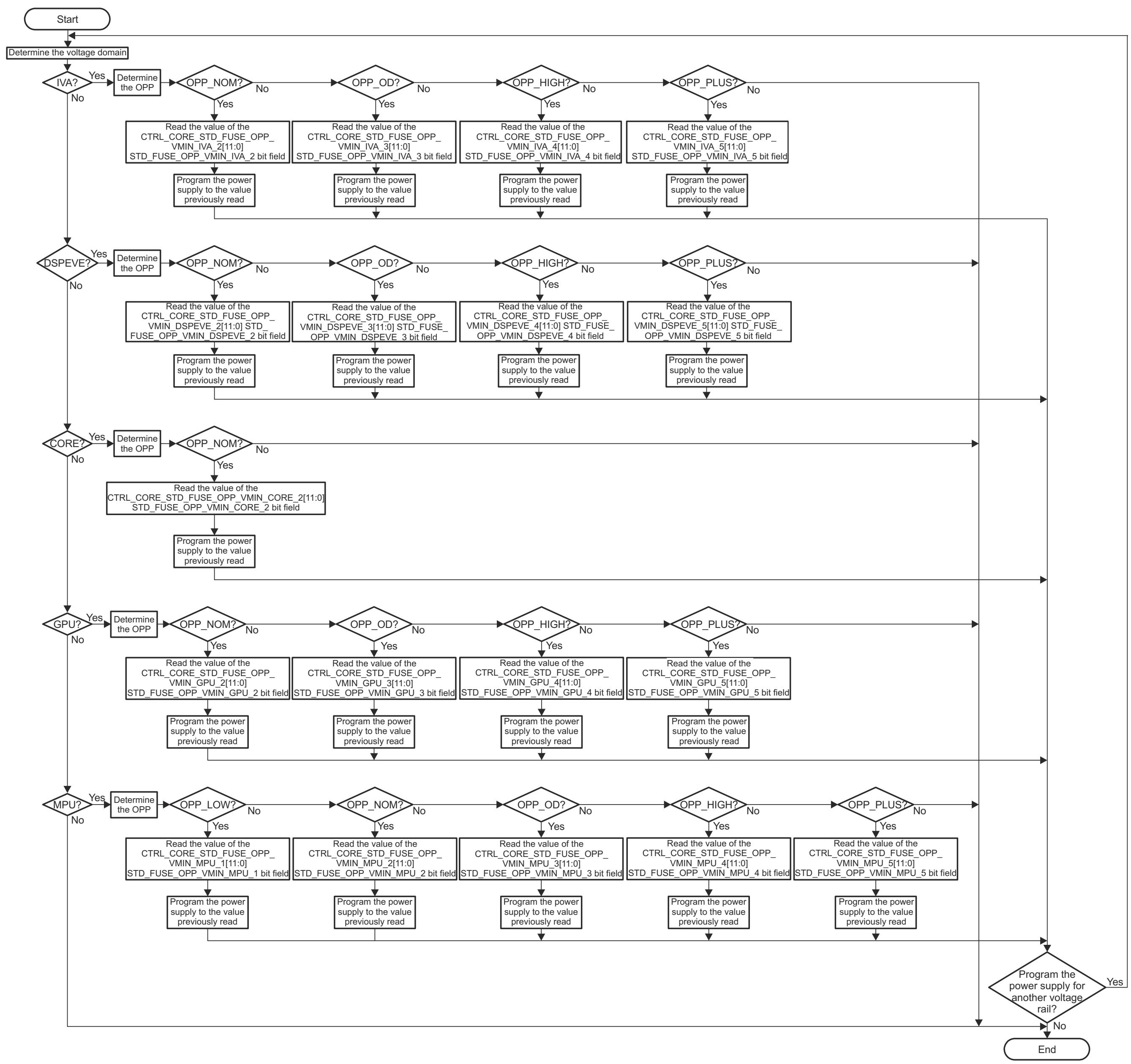


Figure 18-12. AVS Class 0 Procedure

18.4.6.13 ABB Associated Registers

When ABB is needed the following registers should be used:

- CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL
- CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL
- CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL
- CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL
- The CTRL_CORE_STD_FUSE_OPP_VMIN_xxx_y registers depending on the OPP. There are different values for each OPP stored in each one of these registers.

The ABB LDO target value can be read from CTRL_CORE_STD_FUSE_OPP_VMIN_xxx_y[24:20] VSETABB bits, if CTRL_CORE_STD_FUSE_OPP_VMIN_xxx_y[25] ABBEN is set to 0x1. Then this value should be written to bits FBB_VSET_OUT (bits [4:0]) of registers CTRL_CORE/WKUP_LDOVBB_x_VOLTAGE_CTRL. The ABB LDO target value applies when CTRL_CORE/WKUP_LDOVBB_x_VOLTAGE_CTRL[10] FBB_ MUX_CTRL is set to 0x1, that is, FBB_VSET_OUT is used as a target bias voltage. When ABB is bypassed (not used) the FBB_VSET_OUT bits should be loaded with 0x0 and FBB_ MUX_CTRL should be set to 0x0.

It must be taken into account that the ABB LDO target value depends on the OPP. For example, concerning the MPU voltage domain:

- in case of OPP_NOM the value from CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_2[24:20] VSETABB should be used.
- in case of OPP_OD the value from CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_3[24:20] VSETABB should be used.

Note

Before enabling the ABB LDO in FBB mode the FBB_VSET_OUT and FBB_ MUX_CTRL bits must be programmed. The FBB_VSET_OUT bits must be loaded with a value which corresponds to the relevant OPP.

For more details regarding ABB see Section 3.8.3.4 *ABB LDO Programming sequence*.

18.4.6.14 Registers For Other Miscellaneous Functions

18.4.6.14.1 System Boot Status Settings

The CTRL_CORE_BOOTSTRAP register is a status register which indicates the state of the sysboot0 to sysboot15 input signals. Their purpose is to select the boot interface, the device source clock configuration and also other boot related settings.

Note

For proper device operation, sysboot14 must be tied to vss . For SR1.1, sysboot15 must be tied to vdd, but for SR2.0 it is configurable. For more information, see Section 18.4.6.1.1.1, *Permanent PU/PD disabling (SR 2.0 only)* in Chapter 18, *Control Module*.

18.4.6.14.2 Force MPU Write Nonposted Transactions

The control module provides a way for software to force all writes from the MPU subsystem to the L3_MAIN to be nonposted regardless of the attributes of the transactions coming from the MPU. This is done by setting to 0x1 the CTRL_CORE_MPU_FORCEWRNP[0] MPU_FORCEWRNP bit. This bit must not be changed until the transfer completes.

18.4.6.14.3 Firewall Error Status Registers

There are four status registers which show when there is a firewall error. The CTRL_CORE_SEC_ERR_STATUS_FUNC_1 and CTRL_CORE_SEC_ERR_STATUS_FUNC_2 registers are used in device normal operation mode. The x_FW_ERROR bits from these two registers are combined into a single interrupt signal sent to the IRQ_CROSSBAR module as shown in Figure 18-13. The CTRL_CORE_SEC_ERR_STATUS_DEBUG_1 and CTRL_CORE_SEC_ERR_STATUS_DEBUG_2 registers are used when the device is in debug mode. These two registers have bits same as in the CTRL_CORE_SEC_ERR_STATUS_FUNC_1 and CTRL_CORE_SEC_ERR_STATUS_FUNC_2 registers.

All bits in these registers are cleared when the ERROR_LOG_k and L4_IA_ERROR_LOG_L registers are cleared.

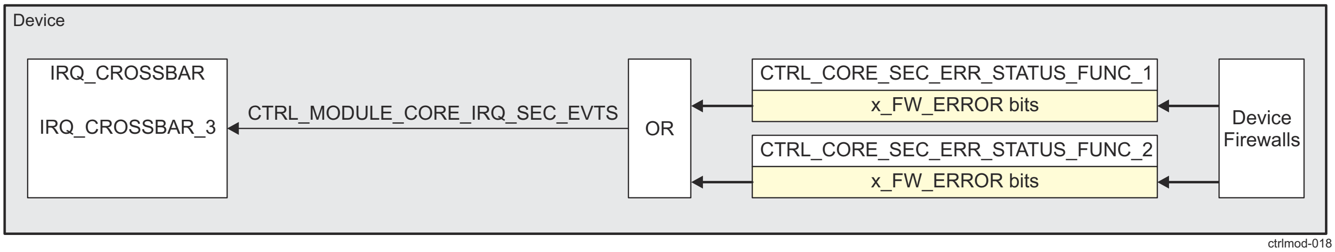


Figure 18-13. Combined Firewall Error Interrupt

18.4.6.14.4 Settings Related To Different Peripheral Modules

The CTRL_CORE_CONTROL_IO_1 and CTRL_CORE_CONTROL_IO_2 registers have controls for specific settings of several device peripheral modules.

18.4.7 Functional Description Of The Various Register Types In CTRL_MODULE_WKUP Submodule

The following sections describe in detail the purpose of the various kinds of registers and register groups which reside in the CTRL_MODULE_WKUP submodule.

18.4.7.1 Registers For Basic EMIF Configuration

The CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG and CTRL_WKUP_SECURE_EMIF2_SDRAM_CONFIG registers have bits which determine the basic settings of the two EMIF controllers. These are, for example, settings like CAS write latency, SDRAM drive strength, SDRAM termination resistor values, SDRAM type and others.

The CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG bit field values are exported upon POR to the EMIF1.EMIF_SDRAM_CONFIG register.

The CTRL_WKUP_SECURE_EMIF2_SDRAM_CONFIG bit field values are exported upon POR to the EMIF2.EMIF_SDRAM_CONFIG register.

The CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT and CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT registers are associated with DDR PHY controls, ODT values for device DDR I/Os, some leveling related parameters and others.

18.5 Control Module Register Manual

18.5.1 Control Module Instance Summary

Table 18-27. CONTROL MODULE Instance Summary

Module Name	Module Base Address	Size
CTRL_MODULE_CORE	0x4A00 2000	8 KiB
CTRL_MODULE_WKUP	0x4AE0 C000	4 KiB

18.5.2 CTRL_MODULE_CORE Registers

18.5.3 CTRL_MODULE_CORE Register Summary

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
RESERVED_k (k = 0 to 76)	R	32	0x0000 0000 + (k*4)	0x4A00 2000 + (k*4)
CTRL_CORE_STATUS	R	32	0x0000 0134	0x4A00 2134
RESERVED	R	32	0x0000 0138	0x4A00 2138
RESERVED	R	32	0x0000 013C	0x4A00 213C
RESERVED	R	32	0x0000 0140	0x4A00 2140
RESERVED	R	32	0x0000 0144	0x4A00 2144
CTRL_CORE_SEC_ERR_STATUS_FUNC_1	RW	32	0x0000 0148	0x4A00 2148
RESERVED	R	32	0x0000 014C	0x4A00 214C
CTRL_CORE_SEC_ERR_STATUS_DEBUG_1	RW	32	0x0000 0150	0x4A00 2150
RESERVED	R	32	0x0000 0154	0x4A00 2154
RESERVED	R	32	0x0000 0158	0x4A00 2158
CTRL_CORE_MPU_FORCEWRNP	RW	32	0x0000 015C	0x4A00 215C
RESERVED	R	32	0x0000 0160	0x4A00 2160
RESERVED	R	32	0x0000 0164	0x4A00 2164
RESERVED	R	32	0x0000 0168	0x4A00 2168
RESERVED	R	32	0x0000 016C	0x4A00 216C
RESERVED	R	32	0x0000 0170	0x4A00 2170
RESERVED	R	32	0x0000 0174	0x4A00 2174
RESERVED	R	32	0x0000 0178	0x4A00 2178
RESERVED	R	32	0x0000 017C	0x4A00 217C
RESERVED	R	32	0x0000 0180	0x4A00 2180
RESERVED	R	32	0x0000 0184	0x4A00 2184
RESERVED	R	32	0x0000 0188	0x4A00 2188
RESERVED	R	32	0x0000 018C	0x4A00 218C
RESERVED	R	32	0x0000 0190	0x4A00 2190
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_0	R	32	0x0000 0194	0x4A00 2194
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_1	R	32	0x0000 0198	0x4A00 2198
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_2	R	32	0x0000 019C	0x4A00 219C
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_3	R	32	0x0000 01A0	0x4A00 21A0
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_4	R	32	0x0000 01A4	0x4A00 21A4
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_5	R	32	0x0000 01A8	0x4A00 21A8
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_0	R	32	0x0000 01AC	0x4A00 21AC
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_1	R	32	0x0000 01B0	0x4A00 21B0
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_2	R	32	0x0000 01B4	0x4A00 21B4
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_3	R	32	0x0000 01B8	0x4A00 21B8
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_4	R	32	0x0000 01BC	0x4A00 21BC

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_5	R	32	0x0000 01C0	0x4A00 21C0
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_6	R	32	0x0000 01C4	0x4A00 21C4
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_7	R	32	0x0000 01C8	0x4A00 21C8
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_0	R	32	0x0000 01CC	0x4A00 21CC
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_1	R	32	0x0000 01D0	0x4A00 21D0
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_2	R	32	0x0000 01D4	0x4A00 21D4
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_3	R	32	0x0000 01D8	0x4A00 21D8
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_4	R	32	0x0000 01DC	0x4A00 21DC
CTRL_CORE_STD_FUSE_OPP_BGAP_GPU	R	32	0x0000 01E0	0x4A00 21E0
CTRL_CORE_STD_FUSE_OPP_BGAP_MPU	R	32	0x0000 01E4	0x4A00 21E4
CTRL_CORE_STD_FUSE_OPP_BGAP_CORE	R	32	0x0000 01E8	0x4A00 21E8
CTRL_CORE_STD_FUSE_OPP_BGAP_MPU23	R	32	0x0000 01EC	0x4A00 21EC
RESERVED_x (x = 0 to 11)	R	32	0x0000 01F0	0x4A00 21F0
CTRL_CORE_STD_FUSE_MPK_0	R	32	0x0000 0220	0x4A00 2220
CTRL_CORE_STD_FUSE_MPK_1	R	32	0x0000 0224	0x4A00 2224
CTRL_CORE_STD_FUSE_MPK_2	R	32	0x0000 0228	0x4A00 2228
CTRL_CORE_STD_FUSE_MPK_3	R	32	0x0000 022C	0x4A00 222C
CTRL_CORE_STD_FUSE_MPK_4	R	32	0x0000 0230	0x4A00 2230
CTRL_CORE_STD_FUSE_MPK_5	R	32	0x0000 0234	0x4A00 2234
CTRL_CORE_STD_FUSE_MPK_6	R	32	0x0000 0238	0x4A00 2238
CTRL_CORE_STD_FUSE_MPK_7	R	32	0x0000 023C	0x4A00 223C
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_0	R	32	0x0000 0240	0x4A00 2240
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_1	R	32	0x0000 0244	0x4A00 2244
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_2	R	32	0x0000 0248	0x4A00 2248
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_3	R	32	0x0000 024C	0x4A00 224C
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_4	R	32	0x0000 0250	0x4A00 2250
CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_5	R	32	0x0000 0254	0x4A00 2254
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_0	R	32	0x0000 0258	0x4A00 2258
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_1	R	32	0x0000 025C	0x4A00 225C
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_2	R	32	0x0000 0260	0x4A00 2260
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_3	R	32	0x0000 0264	0x4A00 2264
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_4	R	32	0x0000 0268	0x4A00 2268
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_5	R	32	0x0000 026C	0x4A00 226C
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_6	R	32	0x0000 0270	0x4A00 2270
CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_7	R	32	0x0000 0274	0x4A00 2274
RESERVED_v (v = 0 to 16)	R	32	0x0000 0278 + (v*4)	0x4A00 2278 + (v*4)
CTRL_CORE_CUST_FUSE_SWRV_0	R	32	0x0000 02BC	0x4A00 22BC
CTRL_CORE_CUST_FUSE_SWRV_1	R	32	0x0000 02C0	0x4A00 22C0
CTRL_CORE_CUST_FUSE_SWRV_2	R	32	0x0000 02C4	0x4A00 22C4
CTRL_CORE_CUST_FUSE_SWRV_3	R	32	0x0000 02C8	0x4A00 22C8
CTRL_CORE_CUST_FUSE_SWRV_4	R	32	0x0000 02CC	0x4A00 22CC
CTRL_CORE_CUST_FUSE_SWRV_5	R	32	0x0000 02D0	0x4A00 22D0
CTRL_CORE_CUST_FUSE_SWRV_6	R	32	0x0000 02D4	0x4A00 22D4
RESERVED	R	32	0x0000 02D8	0x4A00 22D8
RESERVED	R	32	0x0000 02DC	0x4A00 22DC
RESERVED	R	32	0x0000 02E0	0x4A00 22E0
RESERVED	R	32	0x0000 02E4	0x4A00 22E4
RESERVED	R	32	0x0000 02E8	0x4A00 22E8
RESERVED	R	32	0x0000 02EC	0x4A00 22EC

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_DEV_CONF	RW	32	0x0000 0300	0x4A00 2300
RESERVED	R	32	0x0000 0304	0x4A00 2304
CTRL_CORE_TEMP_SENSOR_MPU	R	32	0x0000 032C	0x4A00 232C
CTRL_CORE_TEMP_SENSOR_GPU	R	32	0x0000 0330	0x4A00 2330
CTRL_CORE_TEMP_SENSOR_CORE	R	32	0x0000 0334	0x4A00 2334
RESERVED	R	32	0x0000 033C	0x4A00 233C
RESERVED	R	32	0x0000 0340	0x4A00 2340
RESERVED	R	32	0x0000 0344	0x4A00 2344
CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR	RW	32	0x0000 0358	0x4A00 2358
CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR	RW	32	0x0000 035C	0x4A00 235C
CTRL_CORE_HWOBS_CONTROL	RW	32	0x0000 0360	0x4A00 2360
RESERVED	R	32	0x0000 0364	0x4A00 2364
RESERVED	R	32	0x0000 0368	0x4A00 2368
RESERVED	R	32	0x0000 036C	0x4A00 236C
CTRL_CORE_PHY_POWER_USB	RW	32	0x0000 0370	0x4A00 2370
CTRL_CORE_PHY_POWER_SATA	RW	32	0x0000 0374	0x4A00 2374
CTRL_CORE_BANDGAP_MASK_1	RW	32	0x0000 0380	0x4A00 2380
CTRL_CORE_BANDGAP_THRESHOLD_MPU	RW	32	0x0000 0384	0x4A00 2384
CTRL_CORE_BANDGAP_THRESHOLD_GPU	RW	32	0x0000 0388	0x4A00 2388
CTRL_CORE_BANDGAP_THRESHOLD_CORE	RW	32	0x0000 038C	0x4A00 238C
CTRL_CORE_BANDGAP_TSHUT_MPU	RW	32	0x0000 0390	0x4A00 2390
CTRL_CORE_BANDGAP_TSHUT_GPU	RW	32	0x0000 0394	0x4A00 2394
CTRL_CORE_BANDGAP_TSHUT_CORE	RW	32	0x0000 0398	0x4A00 2398
RESERVED	R	32	0x0000 039C	0x4A00 239C
RESERVED	R	32	0x0000 03A0	0x4A00 23A0
RESERVED	R	32	0x0000 03A4	0x4A00 23A4
CTRL_CORE_BANDGAP_STATUS_1	R	32	0x0000 03A8	0x4A00 23A8
CTRL_CORE_SATA_EXT_MODE	RW	32	0x0000 03AC	0x4A00 23AC
RESERVED	R	32	0x0000 03B0	0x4A00 23B0
RESERVED	R	32	0x0000 03B4	0x4A00 23B4
RESERVED	R	32	0x0000 03B8	0x4A00 23B8
RESERVED	R	32	0x0000 03BC	0x4A00 23BC
CTRL_CORE_DTEMP_MPU_0	R	32	0x0000 03C0	0x4A00 23C0
CTRL_CORE_DTEMP_MPU_1	R	32	0x0000 03C4	0x4A00 23C4
CTRL_CORE_DTEMP_MPU_2	R	32	0x0000 03C8	0x4A00 23C8
CTRL_CORE_DTEMP_MPU_3	R	32	0x0000 03CC	0x4A00 23CC
CTRL_CORE_DTEMP_MPU_4	R	32	0x0000 03D0	0x4A00 23D0
CTRL_CORE_DTEMP_GPU_0	R	32	0x0000 03D4	0x4A00 23D4
CTRL_CORE_DTEMP_GPU_1	R	32	0x0000 03D8	0x4A00 23D8
CTRL_CORE_DTEMP_GPU_2	R	32	0x0000 03DC	0x4A00 23DC
CTRL_CORE_DTEMP_GPU_3	R	32	0x0000 03E0	0x4A00 23E0
CTRL_CORE_DTEMP_GPU_4	R	32	0x0000 03E4	0x4A00 23E4
CTRL_CORE_DTEMP_CORE_0	R	32	0x0000 03E8	0x4A00 23E8
CTRL_CORE_DTEMP_CORE_1	R	32	0x0000 03EC	0x4A00 23EC
CTRL_CORE_DTEMP_CORE_2	R	32	0x0000 03F0	0x4A00 23F0
CTRL_CORE_DTEMP_CORE_3	R	32	0x0000 03F4	0x4A00 23F4
CTRL_CORE_DTEMP_CORE_4	R	32	0x0000 03F8	0x4A00 23F8
CTRL_CORE_SMA_SW_0	RW	32	0x0000 03FC	0x4A00 23FC
RESERVED	R	32	0x0000 0400	0x4A00 2400

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
RESERVED	R	32	0x0000 0404	0x4A00 2404
RESERVED	R	32	0x0000 0408	0x4A00 2408
RESERVED	R	32	0x0000 040C	0x4A00 240C
CTRL_CORE_SEC_ERR_STATUS_FUNC_2	RW	32	0x0000 0414	0x4A00 2414
RESERVED	R	32	0x0000 0418	0x4A00 2418
CTRL_CORE_SEC_ERR_STATUS_DEBUG_2	RW	32	0x0000 041C	0x4A00 241C
CTRL_CORE_EMIF_INITIATOR_PRIORITY_1	RW	32	0x0000 0420	0x4A00 2420
CTRL_CORE_EMIF_INITIATOR_PRIORITY_2	RW	32	0x0000 0424	0x4A00 2424
CTRL_CORE_EMIF_INITIATOR_PRIORITY_3	RW	32	0x0000 0428	0x4A00 2428
CTRL_CORE_EMIF_INITIATOR_PRIORITY_4	RW	32	0x0000 042C	0x4A00 242C
CTRL_CORE_EMIF_INITIATOR_PRIORITY_5	RW	32	0x0000 0430	0x4A00 2430
CTRL_CORE_EMIF_INITIATOR_PRIORITY_6	RW	32	0x0000 0434	0x4A00 2434
RESERVED	R	32	0x0000 0438	0x4A00 2438
CTRL_CORE_L3_INITIATOR_PRESSURE_1	RW	32	0x0000 043C	0x4A00 243C
CTRL_CORE_L3_INITIATOR_PRESSURE_2	RW	32	0x0000 0440	0x4A00 2440
RESERVED	R	32	0x0000 0444	0x4A00 2444
CTRL_CORE_L3_INITIATOR_PRESSURE_4	RW	32	0x0000 0448	0x4A00 2448
CTRL_CORE_L3_INITIATOR_PRESSURE_5	RW	32	0x0000 044C	0x4A00 244C
CTRL_CORE_L3_INITIATOR_PRESSURE_6	RW	32	0x0000 0450	0x4A00 2450
RESERVED	R	32	0x0000 0454	0x4A00 2454
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_0	R	32	0x0000 0458	0x4A00 2458
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_1	R	32	0x0000 045C	0x4A00 245C
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_2	R	32	0x0000 0460	0x4A00 2460
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_3	R	32	0x0000 0464	0x4A00 2464
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_4	R	32	0x0000 0468	0x4A00 2468
CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL	RW	32	0x0000 046C	0x4A00 246C
CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL	RW	32	0x0000 0470	0x4A00 2470
RESERVED_c (c = 0 to 28)	R	32	0x0000 0474 + (c*4)	0x4A00 2474 + (c*4)
CTRL_CORE_CUST_FUSE_UID_0	R	32	0x0000 04E8	0x4A00 24E8
CTRL_CORE_CUST_FUSE_UID_1	R	32	0x0000 04EC	0x4A00 24EC
CTRL_CORE_CUST_FUSE_UID_2	R	32	0x0000 04F0	0x4A00 24F0
CTRL_CORE_CUST_FUSE_UID_3	R	32	0x0000 04F4	0x4A00 24F4
CTRL_CORE_CUST_FUSE_UID_4	R	32	0x0000 04F8	0x4A00 24F8
CTRL_CORE_CUST_FUSE_UID_5	R	32	0x0000 04FC	0x4A00 24FC
CTRL_CORE_CUST_FUSE_UID_6	R	32	0x0000 0500	0x4A00 2500
RESERVED	R	32	0x0000 0504	0x4A00 2504
CTRL_CORE_CUST_FUSE_PCIE_ID_0	R	32	0x0000 0508	0x4A00 2508
RESERVED	R	32	0x0000 050C	0x4A00 250C
CTRL_CORE_CUST_FUSE_USB_ID_0	R	32	0x0000 0510	0x4A00 2510
CTRL_CORE_MAC_ID_SW_0	R	32	0x0000 0514	0x4A00 2514
CTRL_CORE_MAC_ID_SW_1	R	32	0x0000 0518	0x4A00 2518
CTRL_CORE_MAC_ID_SW_2	R	32	0x0000 051C	0x4A00 251C
CTRL_CORE_MAC_ID_SW_3	R	32	0x0000 0520	0x4A00 2520
RESERVED_d (d = 0 to 3)	R	32	0x0000 0524 + (d*4)	0x4A00 2524 + (d*4)
CTRL_CORE_SMA_SW_1	RW	32	0x0000 0534	0x4A00 2534
CTRL_CORE_DSS_PLL_CONTROL	RW	32	0x0000 0538	0x4A00 2538
RESERVED	R	32	0x0000 053C	0x4A00 253C
CTRL_CORE_MMR_LOCK_1	RW	32	0x0000 0540	0x4A00 2540

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_MMR_LOCK_2	RW	32	0x0000 0544	0x4A00 2544
CTRL_CORE_MMR_LOCK_3	RW	32	0x0000 0548	0x4A00 2548
CTRL_CORE_MMR_LOCK_4	RW	32	0x0000 054C	0x4A00 254C
CTRL_CORE_MMR_LOCK_5	RW	32	0x0000 0550	0x4A00 2550
CTRL_CORE_CONTROL_IO_1	RW	32	0x0000 0554	0x4A00 2554
CTRL_CORE_CONTROL_IO_2	RW	32	0x0000 0558	0x4A00 2558
CTRL_CORE_CONTROL_DSP1_RST_VECT	RW	32	0x0000 055C	0x4A00 255C
CTRL_CORE_CONTROL_DSP2_RST_VECT	RW	32	0x0000 0560	0x4A00 2560
CTRL_CORE_STD_FUSE_OPP_BGAP_DSPEVE	R	32	0x0000 0564	0x4A00 2564
CTRL_CORE_STD_FUSE_OPP_BGAP_IVA	R	32	0x0000 0568	0x4A00 2568
CTRL_CORE_LDOSRAM_DSPEVE_VOLTAGE_CTRL	RW	32	0x0000 056C	0x4A00 256C
CTRL_CORE_LDOSRAM_IVA_VOLTAGE_CTRL	RW	32	0x0000 0570	0x4A00 2570
CTRL_CORE_TEMP_SENSOR_DSPEVE	R	32	0x0000 0574	0x4A00 2574
CTRL_CORE_TEMP_SENSOR_IVA	R	32	0x0000 0578	0x4A00 2578
CTRL_CORE_BANDGAP_MASK_2	RW	32	0x0000 057C	0x4A00 257C
CTRL_CORE_BANDGAP_THRESHOLD_DSPEVE	RW	32	0x0000 0580	0x4A00 2580
CTRL_CORE_BANDGAP_THRESHOLD_IVA	RW	32	0x0000 0584	0x4A00 2584
CTRL_CORE_BANDGAP_TSHUT_DSPEVE	RW	32	0x0000 0588	0x4A00 2588
CTRL_CORE_BANDGAP_TSHUT_IVA	RW	32	0x0000 058C	0x4A00 258C
RESERVED	R	32	0x0000 0590	0x4A00 2590
RESERVED	R	32	0x0000 0594	0x4A00 2594
CTRL_CORE_BANDGAP_STATUS_2	R	32	0x0000 0598	0x4A00 2598
CTRL_CORE_DTEMP_DSPEVE_0	R	32	0x0000 059C	0x4A00 259C
CTRL_CORE_DTEMP_DSPEVE_1	R	32	0x0000 05A0	0x4A00 25A0
CTRL_CORE_DTEMP_DSPEVE_2	R	32	0x0000 05A4	0x4A00 25A4
CTRL_CORE_DTEMP_DSPEVE_3	R	32	0x0000 05A8	0x4A00 25A8
CTRL_CORE_DTEMP_DSPEVE_4	R	32	0x0000 05AC	0x4A00 25AC
CTRL_CORE_DTEMP_IVA_0	R	32	0x0000 05B0	0x4A00 25B0
CTRL_CORE_DTEMP_IVA_1	R	32	0x0000 05B4	0x4A00 25B4
CTRL_CORE_DTEMP_IVA_2	R	32	0x0000 05B8	0x4A00 25B8
CTRL_CORE_DTEMP_IVA_3	R	32	0x0000 05BC	0x4A00 25BC
CTRL_CORE_DTEMP_IVA_4	R	32	0x0000 05C0	0x4A00 25C0
CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_5	R	32	0x0000 05C4	0x4A00 25C4
RESERVED	R	32	0x0000 05C8	0x4A00 25C8
CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_2	R	32	0x0000 05CC	0x4A00 25CC
CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_3	R	32	0x0000 05D0	0x4A00 25D0
CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_4	R	32	0x0000 05D4	0x4A00 25D4
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5	R	32	0x0000 05D8	0x4A00 25D8
RESERVED	R	32	0x0000 05DC	0x4A00 25DC
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2	R	32	0x0000 05E0	0x4A00 25E0
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3	R	32	0x0000 05E4	0x4A00 25E4
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4	R	32	0x0000 05E8	0x4A00 25E8
RESERVED	R	32	0x0000 05EC	0x4A00 25EC
RESERVED	R	32	0x0000 05F0	0x4A00 25F0
CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2	R	32	0x0000 05F4	0x4A00 25F4
RESERVED	R	32	0x0000 05F8	0x4A00 25F8
RESERVED	R	32	0x0000 05FC	0x4A00 25FC
RESERVED_m (m = 0 to 31)	R	32	0x0000 0600 + (m*4)	0x4A00 2600 + (m*4)
CTRL_CORE_LDOSRAM_CORE_2_VOLTAGE_CTRL	RW	32	0x0000 0680	0x4A00 2680

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_LDOSRAM_CORE_3_VOLTAGE_CTRL	RW	32	0x0000 0684	0x4A00 2684
RESERVED	R	32	0x0000 0688	0x4A00 2688
CTRL_CORE_NMI_DESTINATION_1	RW	32	0x0000 068C	0x4A00 268C
CTRL_CORE_NMI_DESTINATION_2	RW	32	0x0000 0690	0x4A00 2690
RESERVED	R	32	0x0000 0694	0x4A00 2694
CTRL_CORE_IP_PRESSURE	RW	32	0x0000 0698	0x4A00 2698
RESERVED	R	32	0x0000 069C	0x4A00 269C
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_0	R	32	0x0000 06A0	0x4A00 26A0
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_1	R	32	0x0000 06A4	0x4A00 26A4
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_2	R	32	0x0000 06A8	0x4A00 26A8
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_3	R	32	0x0000 06AC	0x4A00 26AC
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_4	R	32	0x0000 06B0	0x4A00 26B0
CTRL_CORE_CUST_FUSE_SWRV_7	R	32	0x0000 06B4	0x4A00 26B4
CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0	R	32	0x0000 06B8	0x4A00 26B8
CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1	R	32	0x0000 06BC	0x4A00 26BC
CTRL_CORE_PCIE_POWER_STATE	RW	32	0x0000 06C0	0x4A00 26C0
CTRL_CORE_BOOTSTRAP	R	32	0x0000 06C4	0x4A00 26C4
CTRL_CORE_MLB_SIG_IO_CTRL	RW	32	0x0000 06C8	0x4A00 26C8
CTRL_CORE_MLB_DAT_IO_CTRL	RW	32	0x0000 06CC	0x4A00 26CC
CTRL_CORE_MLB_CLK_BG_CTRL	RW	32	0x0000 06D0	0x4A00 26D0
RESERVED_n (n = 0 to 48)	R	32	0x0000 06DC + (n*4)	0x4A00 26DC + (n*4)
CTRL_CORE_EVE1_IRQ_0_1	RW	32	0x0000 07A0	0x4A00 27A0
CTRL_CORE_EVE1_IRQ_2_3	RW	32	0x0000 07A4	0x4A00 27A4
CTRL_CORE_EVE1_IRQ_4_5	RW	32	0x0000 07A8	0x4A00 27A8
CTRL_CORE_EVE1_IRQ_6_7	RW	32	0x0000 07AC	0x4A00 27AC
CTRL_CORE_EVE2_IRQ_0_1	RW	32	0x0000 07B0	0x4A00 27B0
CTRL_CORE_EVE2_IRQ_2_3	RW	32	0x0000 07B4	0x4A00 27B4
CTRL_CORE_EVE2_IRQ_4_5	RW	32	0x0000 07B8	0x4A00 27B8
CTRL_CORE_EVE2_IRQ_6_7	RW	32	0x0000 07BC	0x4A00 27BC
RESERVED	R	32	0x0000 07C0	0x4A00 27C0
RESERVED	R	32	0x0000 07C4	0x4A00 27C4
RESERVED	R	32	0x0000 07C8	0x4A00 27C8
RESERVED	R	32	0x0000 07CC	0x4A00 27CC
RESERVED	R	32	0x0000 07D0	0x4A00 27D0
RESERVED	R	32	0x0000 07D4	0x4A00 27D4
RESERVED	R	32	0x0000 07D8	0x4A00 27D8
RESERVED	R	32	0x0000 07DC	0x4A00 27DC
CTRL_CORE_IPU1_IRQ_23_24	RW	32	0x0000 07E0	0x4A00 27E0
CTRL_CORE_IPU1_IRQ_25_26	RW	32	0x0000 07E4	0x4A00 27E4
CTRL_CORE_IPU1_IRQ_27_28	RW	32	0x0000 07E8	0x4A00 27E8
CTRL_CORE_IPU1_IRQ_29_30	RW	32	0x0000 07EC	0x4A00 27EC
CTRL_CORE_IPU1_IRQ_31_32	RW	32	0x0000 07F0	0x4A00 27F0
CTRL_CORE_IPU1_IRQ_33_34	RW	32	0x0000 07F4	0x4A00 27F4
CTRL_CORE_IPU1_IRQ_35_36	RW	32	0x0000 07F8	0x4A00 27F8
CTRL_CORE_IPU1_IRQ_37_38	RW	32	0x0000 07FC	0x4A00 27FC
CTRL_CORE_IPU1_IRQ_39_40	RW	32	0x0000 0800	0x4A00 2800
CTRL_CORE_IPU1_IRQ_41_42	RW	32	0x0000 0804	0x4A00 2804
CTRL_CORE_IPU1_IRQ_43_44	RW	32	0x0000 0808	0x4A00 2808
CTRL_CORE_IPU1_IRQ_45_46	RW	32	0x0000 080C	0x4A00 280C

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_IPU1_IRQ_47_48	RW	32	0x0000 0810	0x4A00 2810
CTRL_CORE_IPU1_IRQ_49_50	RW	32	0x0000 0814	0x4A00 2814
CTRL_CORE_IPU1_IRQ_51_52	RW	32	0x0000 0818	0x4A00 2818
CTRL_CORE_IPU1_IRQ_53_54	RW	32	0x0000 081C	0x4A00 281C
CTRL_CORE_IPU1_IRQ_55_56	RW	32	0x0000 0820	0x4A00 2820
CTRL_CORE_IPU1_IRQ_57_58	RW	32	0x0000 0824	0x4A00 2824
CTRL_CORE_IPU1_IRQ_59_60	RW	32	0x0000 0828	0x4A00 2828
CTRL_CORE_IPU1_IRQ_61_62	RW	32	0x0000 082C	0x4A00 282C
CTRL_CORE_IPU1_IRQ_63_64	RW	32	0x0000 0830	0x4A00 2830
CTRL_CORE_IPU1_IRQ_65_66	RW	32	0x0000 0834	0x4A00 2834
CTRL_CORE_IPU1_IRQ_67_68	RW	32	0x0000 0838	0x4A00 2838
CTRL_CORE_IPU1_IRQ_69_70	RW	32	0x0000 083C	0x4A00 283C
CTRL_CORE_IPU1_IRQ_71_72	RW	32	0x0000 0840	0x4A00 2840
CTRL_CORE_IPU1_IRQ_73_74	RW	32	0x0000 0844	0x4A00 2844
CTRL_CORE_IPU1_IRQ_75_76	RW	32	0x0000 0848	0x4A00 2848
CTRL_CORE_IPU1_IRQ_77_78	RW	32	0x0000 084C	0x4A00 284C
CTRL_CORE_IPU1_IRQ_79_80	RW	32	0x0000 0850	0x4A00 2850
CTRL_CORE_IPU2_IRQ_23_24	RW	32	0x0000 0854	0x4A00 2854
CTRL_CORE_IPU2_IRQ_25_26	RW	32	0x0000 0858	0x4A00 2858
CTRL_CORE_IPU2_IRQ_27_28	RW	32	0x0000 085C	0x4A00 285C
CTRL_CORE_IPU2_IRQ_29_30	RW	32	0x0000 0860	0x4A00 2860
CTRL_CORE_IPU2_IRQ_31_32	RW	32	0x0000 0864	0x4A00 2864
CTRL_CORE_IPU2_IRQ_33_34	RW	32	0x0000 0868	0x4A00 2868
CTRL_CORE_IPU2_IRQ_35_36	RW	32	0x0000 086C	0x4A00 286C
CTRL_CORE_IPU2_IRQ_37_38	RW	32	0x0000 0870	0x4A00 2870
CTRL_CORE_IPU2_IRQ_39_40	RW	32	0x0000 0874	0x4A00 2874
CTRL_CORE_IPU2_IRQ_41_42	RW	32	0x0000 0878	0x4A00 2878
CTRL_CORE_IPU2_IRQ_43_44	RW	32	0x0000 087C	0x4A00 287C
CTRL_CORE_IPU2_IRQ_45_46	RW	32	0x0000 0880	0x4A00 2880
CTRL_CORE_IPU2_IRQ_47_48	RW	32	0x0000 0884	0x4A00 2884
CTRL_CORE_IPU2_IRQ_49_50	RW	32	0x0000 0888	0x4A00 2888
CTRL_CORE_IPU2_IRQ_51_52	RW	32	0x0000 088C	0x4A00 288C
CTRL_CORE_IPU2_IRQ_53_54	RW	32	0x0000 0890	0x4A00 2890
CTRL_CORE_IPU2_IRQ_55_56	RW	32	0x0000 0894	0x4A00 2894
CTRL_CORE_IPU2_IRQ_57_58	RW	32	0x0000 0898	0x4A00 2898
CTRL_CORE_IPU2_IRQ_59_60	RW	32	0x0000 089C	0x4A00 289C
CTRL_CORE_IPU2_IRQ_61_62	RW	32	0x0000 08A0	0x4A00 28A0
CTRL_CORE_IPU2_IRQ_63_64	RW	32	0x0000 08A4	0x4A00 28A4
CTRL_CORE_IPU2_IRQ_65_66	RW	32	0x0000 08A8	0x4A00 28A8
CTRL_CORE_IPU2_IRQ_67_68	RW	32	0x0000 08AC	0x4A00 28AC
CTRL_CORE_IPU2_IRQ_69_70	RW	32	0x0000 08B0	0x4A00 28B0
CTRL_CORE_IPU2_IRQ_71_72	RW	32	0x0000 08B4	0x4A00 28B4
CTRL_CORE_IPU2_IRQ_73_74	RW	32	0x0000 08B8	0x4A00 28B8
CTRL_CORE_IPU2_IRQ_75_76	RW	32	0x0000 08BC	0x4A00 28BC
CTRL_CORE_IPU2_IRQ_77_78	RW	32	0x0000 08C0	0x4A00 28C0
CTRL_CORE_IPU2_IRQ_79_80	RW	32	0x0000 08C4	0x4A00 28C4
RESERVED_y (y = 0 to 31)	R	32	0x0000 08C8 + (y*4)	0x4A00 28C8 + (y*4)
CTRL_CORE_DSP1_IRQ_32_33	RW	32	0x0000 0948	0x4A00 2948
CTRL_CORE_DSP1_IRQ_34_35	RW	32	0x0000 094C	0x4A00 294C

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_DSP1_IRQ_36_37	RW	32	0x0000 0950	0x4A00 2950
CTRL_CORE_DSP1_IRQ_38_39	RW	32	0x0000 0954	0x4A00 2954
CTRL_CORE_DSP1_IRQ_40_41	RW	32	0x0000 0958	0x4A00 2958
CTRL_CORE_DSP1_IRQ_42_43	RW	32	0x0000 095C	0x4A00 295C
CTRL_CORE_DSP1_IRQ_44_45	RW	32	0x0000 0960	0x4A00 2960
CTRL_CORE_DSP1_IRQ_46_47	RW	32	0x0000 0964	0x4A00 2964
CTRL_CORE_DSP1_IRQ_48_49	RW	32	0x0000 0968	0x4A00 2968
CTRL_CORE_DSP1_IRQ_50_51	RW	32	0x0000 096C	0x4A00 296C
CTRL_CORE_DSP1_IRQ_52_53	RW	32	0x0000 0970	0x4A00 2970
CTRL_CORE_DSP1_IRQ_54_55	RW	32	0x0000 0974	0x4A00 2974
CTRL_CORE_DSP1_IRQ_56_57	RW	32	0x0000 0978	0x4A00 2978
CTRL_CORE_DSP1_IRQ_58_59	RW	32	0x0000 097C	0x4A00 297C
CTRL_CORE_DSP1_IRQ_60_61	RW	32	0x0000 0980	0x4A00 2980
CTRL_CORE_DSP1_IRQ_62_63	RW	32	0x0000 0984	0x4A00 2984
CTRL_CORE_DSP1_IRQ_64_65	RW	32	0x0000 0988	0x4A00 2988
CTRL_CORE_DSP1_IRQ_66_67	RW	32	0x0000 098C	0x4A00 298C
CTRL_CORE_DSP1_IRQ_68_69	RW	32	0x0000 0990	0x4A00 2990
CTRL_CORE_DSP1_IRQ_70_71	RW	32	0x0000 0994	0x4A00 2994
CTRL_CORE_DSP1_IRQ_72_73	RW	32	0x0000 0998	0x4A00 2998
CTRL_CORE_DSP1_IRQ_74_75	RW	32	0x0000 099C	0x4A00 299C
CTRL_CORE_DSP1_IRQ_76_77	RW	32	0x0000 09A0	0x4A00 29A0
CTRL_CORE_DSP1_IRQ_78_79	RW	32	0x0000 09A4	0x4A00 29A4
CTRL_CORE_DSP1_IRQ_80_81	RW	32	0x0000 09A8	0x4A00 29A8
CTRL_CORE_DSP1_IRQ_82_83	RW	32	0x0000 09AC	0x4A00 29AC
CTRL_CORE_DSP1_IRQ_84_85	RW	32	0x0000 09B0	0x4A00 29B0
CTRL_CORE_DSP1_IRQ_86_87	RW	32	0x0000 09B4	0x4A00 29B4
CTRL_CORE_DSP1_IRQ_88_89	RW	32	0x0000 09B8	0x4A00 29B8
CTRL_CORE_DSP1_IRQ_90_91	RW	32	0x0000 09BC	0x4A00 29BC
CTRL_CORE_DSP1_IRQ_92_93	RW	32	0x0000 09C0	0x4A00 29C0
CTRL_CORE_DSP1_IRQ_94_95	RW	32	0x0000 09C4	0x4A00 29C4
CTRL_CORE_DSP2_IRQ_32_33	RW	32	0x0000 09C8	0x4A00 29C8
CTRL_CORE_DSP2_IRQ_34_35	RW	32	0x0000 09CC	0x4A00 29CC
CTRL_CORE_DSP2_IRQ_36_37	RW	32	0x0000 09D0	0x4A00 29D0
CTRL_CORE_DSP2_IRQ_38_39	RW	32	0x0000 09D4	0x4A00 29D4
CTRL_CORE_DSP2_IRQ_40_41	RW	32	0x0000 09D8	0x4A00 29D8
CTRL_CORE_DSP2_IRQ_42_43	RW	32	0x0000 09DC	0x4A00 29DC
CTRL_CORE_DSP2_IRQ_44_45	RW	32	0x0000 09E0	0x4A00 29E0
CTRL_CORE_DSP2_IRQ_46_47	RW	32	0x0000 09E4	0x4A00 29E4
CTRL_CORE_DSP2_IRQ_48_49	RW	32	0x0000 09E8	0x4A00 29E8
CTRL_CORE_DSP2_IRQ_50_51	RW	32	0x0000 09EC	0x4A00 29EC
CTRL_CORE_DSP2_IRQ_52_53	RW	32	0x0000 09F0	0x4A00 29F0
CTRL_CORE_DSP2_IRQ_54_55	RW	32	0x0000 09F4	0x4A00 29F4
CTRL_CORE_DSP2_IRQ_56_57	RW	32	0x0000 09F8	0x4A00 29F8
CTRL_CORE_DSP2_IRQ_58_59	RW	32	0x0000 09FC	0x4A00 29FC
CTRL_CORE_DSP2_IRQ_60_61	RW	32	0x0000 0A00	0x4A00 2A00
CTRL_CORE_DSP2_IRQ_62_63	RW	32	0x0000 0A04	0x4A00 2A04
CTRL_CORE_DSP2_IRQ_64_65	RW	32	0x0000 0A08	0x4A00 2A08
CTRL_CORE_DSP2_IRQ_66_67	RW	32	0x0000 0A0C	0x4A00 2A0C
CTRL_CORE_DSP2_IRQ_68_69	RW	32	0x0000 0A10	0x4A00 2A10

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_DSP2_IRQ_70_71	RW	32	0x0000 0A14	0x4A00 2A14
CTRL_CORE_DSP2_IRQ_72_73	RW	32	0x0000 0A18	0x4A00 2A18
CTRL_CORE_DSP2_IRQ_74_75	RW	32	0x0000 0A1C	0x4A00 2A1C
CTRL_CORE_DSP2_IRQ_76_77	RW	32	0x0000 0A20	0x4A00 2A20
CTRL_CORE_DSP2_IRQ_78_79	RW	32	0x0000 0A24	0x4A00 2A24
CTRL_CORE_DSP2_IRQ_80_81	RW	32	0x0000 0A28	0x4A00 2A28
CTRL_CORE_DSP2_IRQ_82_83	RW	32	0x0000 0A2C	0x4A00 2A2C
CTRL_CORE_DSP2_IRQ_84_85	RW	32	0x0000 0A30	0x4A00 2A30
CTRL_CORE_DSP2_IRQ_86_87	RW	32	0x0000 0A34	0x4A00 2A34
CTRL_CORE_DSP2_IRQ_88_89	RW	32	0x0000 0A38	0x4A00 2A38
CTRL_CORE_DSP2_IRQ_90_91	RW	32	0x0000 0A3C	0x4A00 2A3C
CTRL_CORE_DSP2_IRQ_92_93	RW	32	0x0000 0A40	0x4A00 2A40
CTRL_CORE_DSP2_IRQ_94_95	RW	32	0x0000 0A44	0x4A00 2A44
CTRL_CORE_MPU_IRQ_4_7	RW	32	0x0000 0A48	0x4A00 2A48
CTRL_CORE_MPU_IRQ_8_9	RW	32	0x0000 0A4C	0x4A00 2A4C
CTRL_CORE_MPU_IRQ_10_11	RW	32	0x0000 0A50	0x4A00 2A50
CTRL_CORE_MPU_IRQ_12_13	RW	32	0x0000 0A54	0x4A00 2A54
CTRL_CORE_MPU_IRQ_14_15	RW	32	0x0000 0A58	0x4A00 2A58
CTRL_CORE_MPU_IRQ_16_17	RW	32	0x0000 0A5C	0x4A00 2A5C
CTRL_CORE_MPU_IRQ_18_19	RW	32	0x0000 0A60	0x4A00 2A60
CTRL_CORE_MPU_IRQ_20_21	RW	32	0x0000 0A64	0x4A00 2A64
CTRL_CORE_MPU_IRQ_22_23	RW	32	0x0000 0A68	0x4A00 2A68
CTRL_CORE_MPU_IRQ_24_25	RW	32	0x0000 0A6C	0x4A00 2A6C
CTRL_CORE_MPU_IRQ_26_27	RW	32	0x0000 0A70	0x4A00 2A70
CTRL_CORE_MPU_IRQ_28_29	RW	32	0x0000 0A74	0x4A00 2A74
CTRL_CORE_MPU_IRQ_30_31	RW	32	0x0000 0A78	0x4A00 2A78
CTRL_CORE_MPU_IRQ_32_33	RW	32	0x0000 0A7C	0x4A00 2A7C
CTRL_CORE_MPU_IRQ_34_35	RW	32	0x0000 0A80	0x4A00 2A80
CTRL_CORE_MPU_IRQ_36_37	RW	32	0x0000 0A84	0x4A00 2A84
CTRL_CORE_MPU_IRQ_38_39	RW	32	0x0000 0A88	0x4A00 2A88
CTRL_CORE_MPU_IRQ_40_41	RW	32	0x0000 0A8C	0x4A00 2A8C
CTRL_CORE_MPU_IRQ_42_43	RW	32	0x0000 0A90	0x4A00 2A90
CTRL_CORE_MPU_IRQ_44_45	RW	32	0x0000 0A94	0x4A00 2A94
CTRL_CORE_MPU_IRQ_46_47	RW	32	0x0000 0A98	0x4A00 2A98
CTRL_CORE_MPU_IRQ_48_49	RW	32	0x0000 0A9C	0x4A00 2A9C
CTRL_CORE_MPU_IRQ_50_51	RW	32	0x0000 0AA0	0x4A00 2AA0
CTRL_CORE_MPU_IRQ_52_53	RW	32	0x0000 0AA4	0x4A00 2AA4
CTRL_CORE_MPU_IRQ_54_55	RW	32	0x0000 0AA8	0x4A00 2AA8
CTRL_CORE_MPU_IRQ_56_57	RW	32	0x0000 0AAC	0x4A00 2AAC
CTRL_CORE_MPU_IRQ_58_59	RW	32	0x0000 0AB0	0x4A00 2AB0
CTRL_CORE_MPU_IRQ_60_61	RW	32	0x0000 0AB4	0x4A00 2AB4
CTRL_CORE_MPU_IRQ_62_63	RW	32	0x0000 0AB8	0x4A00 2AB8
CTRL_CORE_MPU_IRQ_64_65	RW	32	0x0000 0ABC	0x4A00 2ABC
CTRL_CORE_MPU_IRQ_66_67	RW	32	0x0000 0AC0	0x4A00 2AC0
CTRL_CORE_MPU_IRQ_68_69	RW	32	0x0000 0AC4	0x4A00 2AC4
CTRL_CORE_MPU_IRQ_70_71	RW	32	0x0000 0AC8	0x4A00 2AC8
CTRL_CORE_MPU_IRQ_72_73	RW	32	0x0000 0ACC	0x4A00 2ACC
CTRL_CORE_MPU_IRQ_74_75	RW	32	0x0000 0AD0	0x4A00 2AD0
CTRL_CORE_MPU_IRQ_76_77	RW	32	0x0000 0AD4	0x4A00 2AD4

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_MPU_IRQ_78_79	RW	32	0x0000 0AD8	0x4A00 2AD8
CTRL_CORE_MPU_IRQ_80_81	RW	32	0x0000 0ADC	0x4A00 2ADC
CTRL_CORE_MPU_IRQ_82_83	RW	32	0x0000 0AE0	0x4A00 2AE0
CTRL_CORE_MPU_IRQ_84_85	RW	32	0x0000 0AE4	0x4A00 2AE4
CTRL_CORE_MPU_IRQ_86_87	RW	32	0x0000 0AE8	0x4A00 2AE8
CTRL_CORE_MPU_IRQ_88_89	RW	32	0x0000 0AEC	0x4A00 2AEC
CTRL_CORE_MPU_IRQ_90_91	RW	32	0x0000 0AF0	0x4A00 2AF0
CTRL_CORE_MPU_IRQ_92_93	RW	32	0x0000 0AF4	0x4A00 2AF4
CTRL_CORE_MPU_IRQ_94_95	RW	32	0x0000 0AF8	0x4A00 2AF8
CTRL_CORE_MPU_IRQ_96_97	RW	32	0x0000 0AFC	0x4A00 2AFC
CTRL_CORE_MPU_IRQ_98_99	RW	32	0x0000 0B00	0x4A00 2B00
CTRL_CORE_MPU_IRQ_100_101	RW	32	0x0000 0B04	0x4A00 2B04
CTRL_CORE_MPU_IRQ_102_103	RW	32	0x0000 0B08	0x4A00 2B08
CTRL_CORE_MPU_IRQ_104_105	RW	32	0x0000 0B0C	0x4A00 2B0C
CTRL_CORE_MPU_IRQ_106_107	RW	32	0x0000 0B10	0x4A00 2B10
CTRL_CORE_MPU_IRQ_108_109	RW	32	0x0000 0B14	0x4A00 2B14
CTRL_CORE_MPU_IRQ_110_111	RW	32	0x0000 0B18	0x4A00 2B18
CTRL_CORE_MPU_IRQ_112_113	RW	32	0x0000 0B1C	0x4A00 2B1C
CTRL_CORE_MPU_IRQ_114_115	RW	32	0x0000 0B20	0x4A00 2B20
CTRL_CORE_MPU_IRQ_116_117	RW	32	0x0000 0B24	0x4A00 2B24
CTRL_CORE_MPU_IRQ_118_119	RW	32	0x0000 0B28	0x4A00 2B28
CTRL_CORE_MPU_IRQ_120_121	RW	32	0x0000 0B2C	0x4A00 2B2C
CTRL_CORE_MPU_IRQ_122_123	RW	32	0x0000 0B30	0x4A00 2B30
CTRL_CORE_MPU_IRQ_124_125	RW	32	0x0000 0B34	0x4A00 2B34
CTRL_CORE_MPU_IRQ_126_127	RW	32	0x0000 0B38	0x4A00 2B38
CTRL_CORE_MPU_IRQ_128_129	RW	32	0x0000 0B3C	0x4A00 2B3C
CTRL_CORE_MPU_IRQ_130_133	RW	32	0x0000 0B40	0x4A00 2B40
CTRL_CORE_MPU_IRQ_134_135	RW	32	0x0000 0B44	0x4A00 2B44
CTRL_CORE_MPU_IRQ_136_137	RW	32	0x0000 0B48	0x4A00 2B48
CTRL_CORE_MPU_IRQ_138_139	RW	32	0x0000 0B4C	0x4A00 2B4C
CTRL_CORE_MPU_IRQ_140_141	RW	32	0x0000 0B50	0x4A00 2B50
CTRL_CORE_MPU_IRQ_142_143	RW	32	0x0000 0B54	0x4A00 2B54
CTRL_CORE_MPU_IRQ_144_145	RW	32	0x0000 0B58	0x4A00 2B58
CTRL_CORE_MPU_IRQ_146_147	RW	32	0x0000 0B5C	0x4A00 2B5C
CTRL_CORE_MPU_IRQ_148_149	RW	32	0x0000 0B60	0x4A00 2B60
CTRL_CORE_MPU_IRQ_150_151	RW	32	0x0000 0B64	0x4A00 2B64
CTRL_CORE_MPU_IRQ_152_153	RW	32	0x0000 0B68	0x4A00 2B68
CTRL_CORE_MPU_IRQ_154_155	RW	32	0x0000 0B6C	0x4A00 2B6C
CTRL_CORE_MPU_IRQ_156_157	RW	32	0x0000 0B70	0x4A00 2B70
CTRL_CORE_MPU_IRQ_158_159	RW	32	0x0000 0B74	0x4A00 2B74
CTRL_CORE_DMA_SYSTEM_DREQ_0_1	RW	32	0x0000 0B78	0x4A00 2B78
CTRL_CORE_DMA_SYSTEM_DREQ_2_3	RW	32	0x0000 0B7C	0x4A00 2B7C
CTRL_CORE_DMA_SYSTEM_DREQ_4_5	RW	32	0x0000 0B80	0x4A00 2B80
CTRL_CORE_DMA_SYSTEM_DREQ_6_7	RW	32	0x0000 0B84	0x4A00 2B84
CTRL_CORE_DMA_SYSTEM_DREQ_8_9	RW	32	0x0000 0B88	0x4A00 2B88
CTRL_CORE_DMA_SYSTEM_DREQ_10_11	RW	32	0x0000 0B8C	0x4A00 2B8C
CTRL_CORE_DMA_SYSTEM_DREQ_12_13	RW	32	0x0000 0B90	0x4A00 2B90
CTRL_CORE_DMA_SYSTEM_DREQ_14_15	RW	32	0x0000 0B94	0x4A00 2B94
CTRL_CORE_DMA_SYSTEM_DREQ_16_17	RW	32	0x0000 0B98	0x4A00 2B98

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_DMA_SYSTEM_DREQ_18_19	RW	32	0x0000 0B9C	0x4A00 2B9C
CTRL_CORE_DMA_SYSTEM_DREQ_20_21	RW	32	0x0000 0BA0	0x4A00 2BA0
CTRL_CORE_DMA_SYSTEM_DREQ_22_23	RW	32	0x0000 0BA4	0x4A00 2BA4
CTRL_CORE_DMA_SYSTEM_DREQ_24_25	RW	32	0x0000 0BA8	0x4A00 2BA8
CTRL_CORE_DMA_SYSTEM_DREQ_26_27	RW	32	0x0000 0BAC	0x4A00 2BAC
CTRL_CORE_DMA_SYSTEM_DREQ_28_29	RW	32	0x0000 0BB0	0x4A00 2BB0
CTRL_CORE_DMA_SYSTEM_DREQ_30_31	RW	32	0x0000 0BB4	0x4A00 2BB4
CTRL_CORE_DMA_SYSTEM_DREQ_32_33	RW	32	0x0000 0BB8	0x4A00 2BB8
CTRL_CORE_DMA_SYSTEM_DREQ_34_35	RW	32	0x0000 0BBC	0x4A00 2BBC
CTRL_CORE_DMA_SYSTEM_DREQ_36_37	RW	32	0x0000 0BC0	0x4A00 2BC0
CTRL_CORE_DMA_SYSTEM_DREQ_38_39	RW	32	0x0000 0BC4	0x4A00 2BC4
CTRL_CORE_DMA_SYSTEM_DREQ_40_41	RW	32	0x0000 0BC8	0x4A00 2BC8
CTRL_CORE_DMA_SYSTEM_DREQ_42_43	RW	32	0x0000 0BCC	0x4A00 2BCC
CTRL_CORE_DMA_SYSTEM_DREQ_44_45	RW	32	0x0000 0BD0	0x4A00 2BD0
CTRL_CORE_DMA_SYSTEM_DREQ_46_47	RW	32	0x0000 0BD4	0x4A00 2BD4
CTRL_CORE_DMA_SYSTEM_DREQ_48_49	RW	32	0x0000 0BD8	0x4A00 2BD8
CTRL_CORE_DMA_SYSTEM_DREQ_50_51	RW	32	0x0000 0BDC	0x4A00 2BDC
CTRL_CORE_DMA_SYSTEM_DREQ_52_53	RW	32	0x0000 0BE0	0x4A00 2BE0
CTRL_CORE_DMA_SYSTEM_DREQ_54_55	RW	32	0x0000 0BE4	0x4A00 2BE4
CTRL_CORE_DMA_SYSTEM_DREQ_56_57	RW	32	0x0000 0BE8	0x4A00 2BE8
CTRL_CORE_DMA_SYSTEM_DREQ_58_59	RW	32	0x0000 0BEC	0x4A00 2BEC
CTRL_CORE_DMA_SYSTEM_DREQ_60_61	RW	32	0x0000 0BF0	0x4A00 2BF0
CTRL_CORE_DMA_SYSTEM_DREQ_62_63	RW	32	0x0000 0BF4	0x4A00 2BF4
CTRL_CORE_DMA_SYSTEM_DREQ_64_65	RW	32	0x0000 0BF8	0x4A00 2BF8
CTRL_CORE_DMA_SYSTEM_DREQ_66_67	RW	32	0x0000 0BFC	0x4A00 2BFC
CTRL_CORE_DMA_SYSTEM_DREQ_68_69	RW	32	0x0000 0C00	0x4A00 2C00
CTRL_CORE_DMA_SYSTEM_DREQ_70_71	RW	32	0x0000 0C04	0x4A00 2C04
CTRL_CORE_DMA_SYSTEM_DREQ_72_73	RW	32	0x0000 0C08	0x4A00 2C08
CTRL_CORE_DMA_SYSTEM_DREQ_74_75	RW	32	0x0000 0C0C	0x4A00 2C0C
CTRL_CORE_DMA_SYSTEM_DREQ_76_77	RW	32	0x0000 0C10	0x4A00 2C10
CTRL_CORE_DMA_SYSTEM_DREQ_78_79	RW	32	0x0000 0C14	0x4A00 2C14
CTRL_CORE_DMA_SYSTEM_DREQ_80_81	RW	32	0x0000 0C18	0x4A00 2C18
CTRL_CORE_DMA_SYSTEM_DREQ_82_83	RW	32	0x0000 0C1C	0x4A00 2C1C
CTRL_CORE_DMA_SYSTEM_DREQ_84_85	RW	32	0x0000 0C20	0x4A00 2C20
CTRL_CORE_DMA_SYSTEM_DREQ_86_87	RW	32	0x0000 0C24	0x4A00 2C24
CTRL_CORE_DMA_SYSTEM_DREQ_88_89	RW	32	0x0000 0C28	0x4A00 2C28
CTRL_CORE_DMA_SYSTEM_DREQ_90_91	RW	32	0x0000 0C2C	0x4A00 2C2C
CTRL_CORE_DMA_SYSTEM_DREQ_92_93	RW	32	0x0000 0C30	0x4A00 2C30
CTRL_CORE_DMA_SYSTEM_DREQ_94_95	RW	32	0x0000 0C34	0x4A00 2C34
CTRL_CORE_DMA_SYSTEM_DREQ_96_97	RW	32	0x0000 0C38	0x4A00 2C38
CTRL_CORE_DMA_SYSTEM_DREQ_98_99	RW	32	0x0000 0C3C	0x4A00 2C3C
CTRL_CORE_DMA_SYSTEM_DREQ_100_101	RW	32	0x0000 0C40	0x4A00 2C40
CTRL_CORE_DMA_SYSTEM_DREQ_102_103	RW	32	0x0000 0C44	0x4A00 2C44
CTRL_CORE_DMA_SYSTEM_DREQ_104_105	RW	32	0x0000 0C48	0x4A00 2C48
CTRL_CORE_DMA_SYSTEM_DREQ_106_107	RW	32	0x0000 0C4C	0x4A00 2C4C
CTRL_CORE_DMA_SYSTEM_DREQ_108_109	RW	32	0x0000 0C50	0x4A00 2C50
CTRL_CORE_DMA_SYSTEM_DREQ_110_111	RW	32	0x0000 0C54	0x4A00 2C54
CTRL_CORE_DMA_SYSTEM_DREQ_112_113	RW	32	0x0000 0C58	0x4A00 2C58
CTRL_CORE_DMA_SYSTEM_DREQ_114_115	RW	32	0x0000 0C5C	0x4A00 2C5C

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_DMA_SYSTEM_DREQ_116_117	RW	32	0x0000 0C60	0x4A00 2C60
CTRL_CORE_DMA_SYSTEM_DREQ_118_119	RW	32	0x0000 0C64	0x4A00 2C64
CTRL_CORE_DMA_SYSTEM_DREQ_120_121	RW	32	0x0000 0C68	0x4A00 2C68
CTRL_CORE_DMA_SYSTEM_DREQ_122_123	RW	32	0x0000 0C6C	0x4A00 2C6C
CTRL_CORE_DMA_SYSTEM_DREQ_124_125	RW	32	0x0000 0C70	0x4A00 2C70
CTRL_CORE_DMA_SYSTEM_DREQ_126_127	RW	32	0x0000 0C74	0x4A00 2C74
CTRL_CORE_DMA_EDMA_DREQ_0_1	RW	32	0x0000 0C78	0x4A00 2C78
CTRL_CORE_DMA_EDMA_DREQ_2_3	RW	32	0x0000 0C7C	0x4A00 2C7C
CTRL_CORE_DMA_EDMA_DREQ_4_5	RW	32	0x0000 0C80	0x4A00 2C80
CTRL_CORE_DMA_EDMA_DREQ_6_7	RW	32	0x0000 0C84	0x4A00 2C84
CTRL_CORE_DMA_EDMA_DREQ_8_9	RW	32	0x0000 0C88	0x4A00 2C88
CTRL_CORE_DMA_EDMA_DREQ_10_11	RW	32	0x0000 0C8C	0x4A00 2C8C
CTRL_CORE_DMA_EDMA_DREQ_12_13	RW	32	0x0000 0C90	0x4A00 2C90
CTRL_CORE_DMA_EDMA_DREQ_14_15	RW	32	0x0000 0C94	0x4A00 2C94
CTRL_CORE_DMA_EDMA_DREQ_16_17	RW	32	0x0000 0C98	0x4A00 2C98
CTRL_CORE_DMA_EDMA_DREQ_18_19	RW	32	0x0000 0C9C	0x4A00 2C9C
CTRL_CORE_DMA_EDMA_DREQ_20_21	RW	32	0x0000 0CA0	0x4A00 2CA0
CTRL_CORE_DMA_EDMA_DREQ_22_23	RW	32	0x0000 0CA4	0x4A00 2CA4
CTRL_CORE_DMA_EDMA_DREQ_24_25	RW	32	0x0000 0CA8	0x4A00 2CA8
CTRL_CORE_DMA_EDMA_DREQ_26_27	RW	32	0x0000 0CAC	0x4A00 2CAC
CTRL_CORE_DMA_EDMA_DREQ_28_29	RW	32	0x0000 0CB0	0x4A00 2CB0
CTRL_CORE_DMA_EDMA_DREQ_30_31	RW	32	0x0000 0CB4	0x4A00 2CB4
CTRL_CORE_DMA_EDMA_DREQ_32_33	RW	32	0x0000 0CB8	0x4A00 2CB8
CTRL_CORE_DMA_EDMA_DREQ_34_35	RW	32	0x0000 0CBC	0x4A00 2CBC
CTRL_CORE_DMA_EDMA_DREQ_36_37	RW	32	0x0000 0CC0	0x4A00 2CC0
CTRL_CORE_DMA_EDMA_DREQ_38_39	RW	32	0x0000 0CC4	0x4A00 2CC4
CTRL_CORE_DMA_EDMA_DREQ_40_41	RW	32	0x0000 0CC8	0x4A00 2CC8
CTRL_CORE_DMA_EDMA_DREQ_42_43	RW	32	0x0000 0CCC	0x4A00 2CCC
CTRL_CORE_DMA_EDMA_DREQ_44_45	RW	32	0x0000 0CD0	0x4A00 2CD0
CTRL_CORE_DMA_EDMA_DREQ_46_47	RW	32	0x0000 0CD4	0x4A00 2CD4
CTRL_CORE_DMA_EDMA_DREQ_48_49	RW	32	0x0000 0CD8	0x4A00 2CD8
CTRL_CORE_DMA_EDMA_DREQ_50_51	RW	32	0x0000 0CDC	0x4A00 2CDC
CTRL_CORE_DMA_EDMA_DREQ_52_53	RW	32	0x0000 0CE0	0x4A00 2CE0
CTRL_CORE_DMA_EDMA_DREQ_54_55	RW	32	0x0000 0CE4	0x4A00 2CE4
CTRL_CORE_DMA_EDMA_DREQ_56_57	RW	32	0x0000 0CE8	0x4A00 2CE8
CTRL_CORE_DMA_EDMA_DREQ_58_59	RW	32	0x0000 0CEC	0x4A00 2CEC
CTRL_CORE_DMA_EDMA_DREQ_60_61	RW	32	0x0000 0CF0	0x4A00 2CF0
CTRL_CORE_DMA_EDMA_DREQ_62_63	RW	32	0x0000 0CF4	0x4A00 2CF4
CTRL_CORE_DMA_DSP1_DREQ_0_1	RW	32	0x0000 0CF8	0x4A00 2CF8
CTRL_CORE_DMA_DSP1_DREQ_2_3	RW	32	0x0000 0CFC	0x4A00 2CFC
CTRL_CORE_DMA_DSP1_DREQ_4_5	RW	32	0x0000 0D00	0x4A00 2D00
CTRL_CORE_DMA_DSP1_DREQ_6_7	RW	32	0x0000 0D04	0x4A00 2D04
CTRL_CORE_DMA_DSP1_DREQ_8_9	RW	32	0x0000 0D08	0x4A00 2D08
CTRL_CORE_DMA_DSP1_DREQ_10_11	RW	32	0x0000 0D0C	0x4A00 2D0C
CTRL_CORE_DMA_DSP1_DREQ_12_13	RW	32	0x0000 0D10	0x4A00 2D10
CTRL_CORE_DMA_DSP1_DREQ_14_15	RW	32	0x0000 0D14	0x4A00 2D14
CTRL_CORE_DMA_DSP1_DREQ_16_17	RW	32	0x0000 0D18	0x4A00 2D18
CTRL_CORE_DMA_DSP1_DREQ_18_19	RW	32	0x0000 0D1C	0x4A00 2D1C
CTRL_CORE_DMA_DSP2_DREQ_0_1	RW	32	0x0000 0D20	0x4A00 2D20

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_DMA_DSP2_DREQ_2_3	RW	32	0x0000 0D24	0x4A00 2D24
CTRL_CORE_DMA_DSP2_DREQ_4_5	RW	32	0x0000 0D28	0x4A00 2D28
CTRL_CORE_DMA_DSP2_DREQ_6_7	RW	32	0x0000 0D2C	0x4A00 2D2C
CTRL_CORE_DMA_DSP2_DREQ_8_9	RW	32	0x0000 0D30	0x4A00 2D30
CTRL_CORE_DMA_DSP2_DREQ_10_11	RW	32	0x0000 0D34	0x4A00 2D34
CTRL_CORE_DMA_DSP2_DREQ_12_13	RW	32	0x0000 0D38	0x4A00 2D38
CTRL_CORE_DMA_DSP2_DREQ_14_15	RW	32	0x0000 0D3C	0x4A00 2D3C
CTRL_CORE_DMA_DSP2_DREQ_16_17	RW	32	0x0000 0D40	0x4A00 2D40
CTRL_CORE_DMA_DSP2_DREQ_18_19	RW	32	0x0000 0D44	0x4A00 2D44
RESERVED	R	32	0x0000 0D48	0x4A00 2D48
CTRL_CORE_OVS_DMARQ_IO_MUX	RW	32	0x0000 0D4C	0x4A00 2D4C
CTRL_CORE_OVS_IRQ_IO_MUX	RW	32	0x0000 0D50	0x4A00 2D50
RESERVED_q (q = 0 to 42)	R	32	0x0000 0D54 + (q*4)	0x4A00 2D54 + (q*4)
CTRL_CORE_CONTROL_PBIAS	RW	32	0x0000 0E00	0x4A00 2E00
RESERVED	R	32	0x0000 0E04	0x4A00 2E04
CTRL_CORE_CONTROL_HDMI_TX_PHY	RW	32	0x0000 0E0C	0x4A00 2E0C
RESERVED	R	32	0x0000 0E14	0x4A00 2E14
RESERVED	R	32	0x0000 0E18	0x4A00 2E18
CTRL_CORE_CONTROL_USB2PHYCORE	RW	32	0x0000 0E1C	0x4A00 2E1C
CTRL_CORE_CONTROL_HDMI_1	RW	32	0x0000 0E20	0x4A00 2E20
RESERVED	RW	32	0x0000 0E24	0x4A00 2E24
CTRL_CORE_CONTROL_DDRCACH1_0	RW	32	0x0000 0E30	0x4A00 2E30
CTRL_CORE_CONTROL_DDRCACH2_0	RW	32	0x0000 0E34	0x4A00 2E34
CTRL_CORE_CONTROL_DDRCH1_0	RW	32	0x0000 0E38	0x4A00 2E38
CTRL_CORE_CONTROL_DDRCH1_1	RW	32	0x0000 0E3C	0x4A00 2E3C
CTRL_CORE_CONTROL_DDRCH2_0	RW	32	0x0000 0E40	0x4A00 2E40
CTRL_CORE_CONTROL_DDRCH2_1	RW	32	0x0000 0E44	0x4A00 2E44
CTRL_CORE_CONTROL_DDRCH1_2	RW	32	0x0000 0E48	0x4A00 2E48
RESERVED	R	32	0x0000 0E4C	0x4A00 2E4C
CTRL_CORE_CONTROL_DDRIO_0	RW	32	0x0000 0E50	0x4A00 2E50
CTRL_CORE_CONTROL_DDRIO_1	RW	32	0x0000 0E54	0x4A00 2E54
RESERVED	R	32	0x0000 0E58	0x4A00 2E58
CTRL_CORE_CONTROL_HYST_1	RW	32	0x0000 0E5C	0x4A00 2E5C
RESERVED	R	32	0x0000 0E60	0x4A00 2E60
RESERVED	R	32	0x0000 0E64	0x4A00 2E64
CTRL_CORE_CONTROL_SPARE_RW	RW	32	0x0000 0E68	0x4A00 2E68
RESERVED	R	32	0x0000 0E6C	0x4A00 2E6C
RESERVED	R	32	0x0000 0E70	0x4A00 2E70
CTRL_CORE_SRCOMP_NORTH_SIDE	RW	32	0x0000 0E74	0x4A00 2E74
CTRL_CORE_SRCOMP_SOUTH_SIDE	R	32	0x0000 0E78	0x4A00 2E78
RESERVED_p (p = 0 to 352)	R	32	0x0000 0E7C + (p*4)	0x4A00 2E7C + (p*4)
CTRL_CORE_PAD_GPMC_AD0	RW	32	0x0000 1400	0x4A00 3400
CTRL_CORE_PAD_GPMC_AD1	RW	32	0x0000 1404	0x4A00 3404
CTRL_CORE_PAD_GPMC_AD2	RW	32	0x0000 1408	0x4A00 3408
CTRL_CORE_PAD_GPMC_AD3	RW	32	0x0000 140C	0x4A00 340C
CTRL_CORE_PAD_GPMC_AD4	RW	32	0x0000 1410	0x4A00 3410
CTRL_CORE_PAD_GPMC_AD5	RW	32	0x0000 1414	0x4A00 3414
CTRL_CORE_PAD_GPMC_AD6	RW	32	0x0000 1418	0x4A00 3418

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_PAD_GPMC_AD7	RW	32	0x0000 141C	0x4A00 341C
CTRL_CORE_PAD_GPMC_AD8	RW	32	0x0000 1420	0x4A00 3420
CTRL_CORE_PAD_GPMC_AD9	RW	32	0x0000 1424	0x4A00 3424
CTRL_CORE_PAD_GPMC_AD10	RW	32	0x0000 1428	0x4A00 3428
CTRL_CORE_PAD_GPMC_AD11	RW	32	0x0000 142C	0x4A00 342C
CTRL_CORE_PAD_GPMC_AD12	RW	32	0x0000 1430	0x4A00 3430
CTRL_CORE_PAD_GPMC_AD13	RW	32	0x0000 1434	0x4A00 3434
CTRL_CORE_PAD_GPMC_AD14	RW	32	0x0000 1438	0x4A00 3438
CTRL_CORE_PAD_GPMC_AD15	RW	32	0x0000 143C	0x4A00 343C
CTRL_CORE_PAD_GPMC_A0	RW	32	0x0000 1440	0x4A00 3440
CTRL_CORE_PAD_GPMC_A1	RW	32	0x0000 1444	0x4A00 3444
CTRL_CORE_PAD_GPMC_A2	RW	32	0x0000 1448	0x4A00 3448
CTRL_CORE_PAD_GPMC_A3	RW	32	0x0000 144C	0x4A00 344C
CTRL_CORE_PAD_GPMC_A4	RW	32	0x0000 1450	0x4A00 3450
CTRL_CORE_PAD_GPMC_A5	RW	32	0x0000 1454	0x4A00 3454
CTRL_CORE_PAD_GPMC_A6	RW	32	0x0000 1458	0x4A00 3458
CTRL_CORE_PAD_GPMC_A7	RW	32	0x0000 145C	0x4A00 345C
CTRL_CORE_PAD_GPMC_A8	RW	32	0x0000 1460	0x4A00 3460
CTRL_CORE_PAD_GPMC_A9	RW	32	0x0000 1464	0x4A00 3464
CTRL_CORE_PAD_GPMC_A10	RW	32	0x0000 1468	0x4A00 3468
CTRL_CORE_PAD_GPMC_A11	RW	32	0x0000 146C	0x4A00 346C
CTRL_CORE_PAD_GPMC_A12	RW	32	0x0000 1470	0x4A00 3470
CTRL_CORE_PAD_GPMC_A13	RW	32	0x0000 1474	0x4A00 3474
CTRL_CORE_PAD_GPMC_A14	RW	32	0x0000 1478	0x4A00 3478
CTRL_CORE_PAD_GPMC_A15	RW	32	0x0000 147C	0x4A00 347C
CTRL_CORE_PAD_GPMC_A16	RW	32	0x0000 1480	0x4A00 3480
CTRL_CORE_PAD_GPMC_A17	RW	32	0x0000 1484	0x4A00 3484
CTRL_CORE_PAD_GPMC_A18	RW	32	0x0000 1488	0x4A00 3488
CTRL_CORE_PAD_GPMC_A19	RW	32	0x0000 148C	0x4A00 348C
CTRL_CORE_PAD_GPMC_A20	RW	32	0x0000 1490	0x4A00 3490
CTRL_CORE_PAD_GPMC_A21	RW	32	0x0000 1494	0x4A00 3494
CTRL_CORE_PAD_GPMC_A22	RW	32	0x0000 1498	0x4A00 3498
CTRL_CORE_PAD_GPMC_A23	RW	32	0x0000 149C	0x4A00 349C
CTRL_CORE_PAD_GPMC_A24	RW	32	0x0000 14A0	0x4A00 34A0
CTRL_CORE_PAD_GPMC_A25	RW	32	0x0000 14A4	0x4A00 34A4
CTRL_CORE_PAD_GPMC_A26	RW	32	0x0000 14A8	0x4A00 34A8
CTRL_CORE_PAD_GPMC_A27	RW	32	0x0000 14AC	0x4A00 34AC
CTRL_CORE_PAD_GPMC_CS1	RW	32	0x0000 14B0	0x4A00 34B0
CTRL_CORE_PAD_GPMC_CS0	RW	32	0x0000 14B4	0x4A00 34B4
CTRL_CORE_PAD_GPMC_CS2	RW	32	0x0000 14B8	0x4A00 34B8
CTRL_CORE_PAD_GPMC_CS3	RW	32	0x0000 14BC	0x4A00 34BC
CTRL_CORE_PAD_GPMC_CLK	RW	32	0x0000 14C0	0x4A00 34C0
CTRL_CORE_PAD_GPMC_ADVN_ALE	RW	32	0x0000 14C4	0x4A00 34C4
CTRL_CORE_PAD_GPMC_OEN_REN	RW	32	0x0000 14C8	0x4A00 34C8
CTRL_CORE_PAD_GPMC_WEN	RW	32	0x0000 14CC	0x4A00 34CC
CTRL_CORE_PAD_GPMC_BEN0	RW	32	0x0000 14D0	0x4A00 34D0
CTRL_CORE_PAD_GPMC_BEN1	RW	32	0x0000 14D4	0x4A00 34D4
CTRL_CORE_PAD_GPMC_WAIT0	RW	32	0x0000 14D8	0x4A00 34D8
CTRL_CORE_PAD_VIN1A_CLK0	RW	32	0x0000 14DC	0x4A00 34DC

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_PAD_VIN1B_CLK1	RW	32	0x0000 14E0	0x4A00 34E0
CTRL_CORE_PAD_VIN1A_DE0	RW	32	0x0000 14E4	0x4A00 34E4
CTRL_CORE_PAD_VIN1A_FLD0	RW	32	0x0000 14E8	0x4A00 34E8
CTRL_CORE_PAD_VIN1A_HSYNC0	RW	32	0x0000 14EC	0x4A00 34EC
CTRL_CORE_PAD_VIN1A_VSYNC0	RW	32	0x0000 14F0	0x4A00 34F0
CTRL_CORE_PAD_VIN1A_D0	RW	32	0x0000 14F4	0x4A00 34F4
CTRL_CORE_PAD_VIN1A_D1	RW	32	0x0000 14F8	0x4A00 34F8
CTRL_CORE_PAD_VIN1A_D2	RW	32	0x0000 14FC	0x4A00 34FC
CTRL_CORE_PAD_VIN1A_D3	RW	32	0x0000 1500	0x4A00 3500
CTRL_CORE_PAD_VIN1A_D4	RW	32	0x0000 1504	0x4A00 3504
CTRL_CORE_PAD_VIN1A_D5	RW	32	0x0000 1508	0x4A00 3508
CTRL_CORE_PAD_VIN1A_D6	RW	32	0x0000 150C	0x4A00 350C
CTRL_CORE_PAD_VIN1A_D7	RW	32	0x0000 1510	0x4A00 3510
CTRL_CORE_PAD_VIN1A_D8	RW	32	0x0000 1514	0x4A00 3514
CTRL_CORE_PAD_VIN1A_D9	RW	32	0x0000 1518	0x4A00 3518
CTRL_CORE_PAD_VIN1A_D10	RW	32	0x0000 151C	0x4A00 351C
CTRL_CORE_PAD_VIN1A_D11	RW	32	0x0000 1520	0x4A00 3520
CTRL_CORE_PAD_VIN1A_D12	RW	32	0x0000 1524	0x4A00 3524
CTRL_CORE_PAD_VIN1A_D13	RW	32	0x0000 1528	0x4A00 3528
CTRL_CORE_PAD_VIN1A_D14	RW	32	0x0000 152C	0x4A00 352C
CTRL_CORE_PAD_VIN1A_D15	RW	32	0x0000 1530	0x4A00 3530
CTRL_CORE_PAD_VIN1A_D16	RW	32	0x0000 1534	0x4A00 3534
CTRL_CORE_PAD_VIN1A_D17	RW	32	0x0000 1538	0x4A00 3538
CTRL_CORE_PAD_VIN1A_D18	RW	32	0x0000 153C	0x4A00 353C
CTRL_CORE_PAD_VIN1A_D19	RW	32	0x0000 1540	0x4A00 3540
CTRL_CORE_PAD_VIN1A_D20	RW	32	0x0000 1544	0x4A00 3544
CTRL_CORE_PAD_VIN1A_D21	RW	32	0x0000 1548	0x4A00 3548
CTRL_CORE_PAD_VIN1A_D22	RW	32	0x0000 154C	0x4A00 354C
CTRL_CORE_PAD_VIN1A_D23	RW	32	0x0000 1550	0x4A00 3550
CTRL_CORE_PAD_VIN2A_CLK0	RW	32	0x0000 1554	0x4A00 3554
CTRL_CORE_PAD_VIN2A_DE0	RW	32	0x0000 1558	0x4A00 3558
CTRL_CORE_PAD_VIN2A_FLD0	RW	32	0x0000 155C	0x4A00 355C
CTRL_CORE_PAD_VIN2A_HSYNC0	RW	32	0x0000 1560	0x4A00 3560
CTRL_CORE_PAD_VIN2A_VSYNC0	RW	32	0x0000 1564	0x4A00 3564
CTRL_CORE_PAD_VIN2A_D0	RW	32	0x0000 1568	0x4A00 3568
CTRL_CORE_PAD_VIN2A_D1	RW	32	0x0000 156C	0x4A00 356C
CTRL_CORE_PAD_VIN2A_D2	RW	32	0x0000 1570	0x4A00 3570
CTRL_CORE_PAD_VIN2A_D3	RW	32	0x0000 1574	0x4A00 3574
CTRL_CORE_PAD_VIN2A_D4	RW	32	0x0000 1578	0x4A00 3578
CTRL_CORE_PAD_VIN2A_D5	RW	32	0x0000 157C	0x4A00 357C
CTRL_CORE_PAD_VIN2A_D6	RW	32	0x0000 1580	0x4A00 3580
CTRL_CORE_PAD_VIN2A_D7	RW	32	0x0000 1584	0x4A00 3584
CTRL_CORE_PAD_VIN2A_D8	RW	32	0x0000 1588	0x4A00 3588
CTRL_CORE_PAD_VIN2A_D9	RW	32	0x0000 158C	0x4A00 358C
CTRL_CORE_PAD_VIN2A_D10	RW	32	0x0000 1590	0x4A00 3590
CTRL_CORE_PAD_VIN2A_D11	RW	32	0x0000 1594	0x4A00 3594
CTRL_CORE_PAD_VIN2A_D12	RW	32	0x0000 1598	0x4A00 3598
CTRL_CORE_PAD_VIN2A_D13	RW	32	0x0000 159C	0x4A00 359C
CTRL_CORE_PAD_VIN2A_D14	RW	32	0x0000 15A0	0x4A00 35A0

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_PAD_VIN2A_D15	RW	32	0x0000 15A4	0x4A00 35A4
CTRL_CORE_PAD_VIN2A_D16	RW	32	0x0000 15A8	0x4A00 35A8
CTRL_CORE_PAD_VIN2A_D17	RW	32	0x0000 15AC	0x4A00 35AC
CTRL_CORE_PAD_VIN2A_D18	RW	32	0x0000 15B0	0x4A00 35B0
CTRL_CORE_PAD_VIN2A_D19	RW	32	0x0000 15B4	0x4A00 35B4
CTRL_CORE_PAD_VIN2A_D20	RW	32	0x0000 15B8	0x4A00 35B8
CTRL_CORE_PAD_VIN2A_D21	RW	32	0x0000 15BC	0x4A00 35BC
CTRL_CORE_PAD_VIN2A_D22	RW	32	0x0000 15C0	0x4A00 35C0
CTRL_CORE_PAD_VIN2A_D23	RW	32	0x0000 15C4	0x4A00 35C4
CTRL_CORE_PAD_VOUT1_CLK	RW	32	0x0000 15C8	0x4A00 35C8
CTRL_CORE_PAD_VOUT1_DE	RW	32	0x0000 15CC	0x4A00 35CC
CTRL_CORE_PAD_VOUT1_FLD	RW	32	0x0000 15D0	0x4A00 35D0
CTRL_CORE_PAD_VOUT1_HSYNC	RW	32	0x0000 15D4	0x4A00 35D4
CTRL_CORE_PAD_VOUT1_VSYNC	RW	32	0x0000 15D8	0x4A00 35D8
CTRL_CORE_PAD_VOUT1_D0	RW	32	0x0000 15DC	0x4A00 35DC
CTRL_CORE_PAD_VOUT1_D1	RW	32	0x0000 15E0	0x4A00 35E0
CTRL_CORE_PAD_VOUT1_D2	RW	32	0x0000 15E4	0x4A00 35E4
CTRL_CORE_PAD_VOUT1_D3	RW	32	0x0000 15E8	0x4A00 35E8
CTRL_CORE_PAD_VOUT1_D4	RW	32	0x0000 15EC	0x4A00 35EC
CTRL_CORE_PAD_VOUT1_D5	RW	32	0x0000 15F0	0x4A00 35F0
CTRL_CORE_PAD_VOUT1_D6	RW	32	0x0000 15F4	0x4A00 35F4
CTRL_CORE_PAD_VOUT1_D7	RW	32	0x0000 15F8	0x4A00 35F8
CTRL_CORE_PAD_VOUT1_D8	RW	32	0x0000 15FC	0x4A00 35FC
CTRL_CORE_PAD_VOUT1_D9	RW	32	0x0000 1600	0x4A00 3600
CTRL_CORE_PAD_VOUT1_D10	RW	32	0x0000 1604	0x4A00 3604
CTRL_CORE_PAD_VOUT1_D11	RW	32	0x0000 1608	0x4A00 3608
CTRL_CORE_PAD_VOUT1_D12	RW	32	0x0000 160C	0x4A00 360C
CTRL_CORE_PAD_VOUT1_D13	RW	32	0x0000 1610	0x4A00 3610
CTRL_CORE_PAD_VOUT1_D14	RW	32	0x0000 1614	0x4A00 3614
CTRL_CORE_PAD_VOUT1_D15	RW	32	0x0000 1618	0x4A00 3618
CTRL_CORE_PAD_VOUT1_D16	RW	32	0x0000 161C	0x4A00 361C
CTRL_CORE_PAD_VOUT1_D17	RW	32	0x0000 1620	0x4A00 3620
CTRL_CORE_PAD_VOUT1_D18	RW	32	0x0000 1624	0x4A00 3624
CTRL_CORE_PAD_VOUT1_D19	RW	32	0x0000 1628	0x4A00 3628
CTRL_CORE_PAD_VOUT1_D20	RW	32	0x0000 162C	0x4A00 362C
CTRL_CORE_PAD_VOUT1_D21	RW	32	0x0000 1630	0x4A00 3630
CTRL_CORE_PAD_VOUT1_D22	RW	32	0x0000 1634	0x4A00 3634
CTRL_CORE_PAD_VOUT1_D23	RW	32	0x0000 1638	0x4A00 3638
CTRL_CORE_PAD_MDIO_MCLK	RW	32	0x0000 163C	0x4A00 363C
CTRL_CORE_PAD_MDIO_D	RW	32	0x0000 1640	0x4A00 3640
CTRL_CORE_PAD_RMII_MHZ_50_CLK	RW	32	0x0000 1644	0x4A00 3644
CTRL_CORE_PAD_UART3_RXD	RW	32	0x0000 1648	0x4A00 3648
CTRL_CORE_PAD_UART3_TXD	RW	32	0x0000 164C	0x4A00 364C
CTRL_CORE_PAD_RGMII0_TXC	RW	32	0x0000 1650	0x4A00 3650
CTRL_CORE_PAD_RGMII0_TXCTL	RW	32	0x0000 1654	0x4A00 3654
CTRL_CORE_PAD_RGMII0_TXD3	RW	32	0x0000 1658	0x4A00 3658
CTRL_CORE_PAD_RGMII0_TXD2	RW	32	0x0000 165C	0x4A00 365C
CTRL_CORE_PAD_RGMII0_TXD1	RW	32	0x0000 1660	0x4A00 3660
CTRL_CORE_PAD_RGMII0_TXD0	RW	32	0x0000 1664	0x4A00 3664

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_PAD_RGMII0_RXC	RW	32	0x0000 1668	0x4A00 3668
CTRL_CORE_PAD_RGMII0_RXCTL	RW	32	0x0000 166C	0x4A00 366C
CTRL_CORE_PAD_RGMII0_RXD3	RW	32	0x0000 1670	0x4A00 3670
CTRL_CORE_PAD_RGMII0_RXD2	RW	32	0x0000 1674	0x4A00 3674
CTRL_CORE_PAD_RGMII0_RXD1	RW	32	0x0000 1678	0x4A00 3678
CTRL_CORE_PAD_RGMII0_RXD0	RW	32	0x0000 167C	0x4A00 367C
CTRL_CORE_PAD_USB1_DRVVBUS	RW	32	0x0000 1680	0x4A00 3680
CTRL_CORE_PAD_USB2_DRVVBUS	RW	32	0x0000 1684	0x4A00 3684
CTRL_CORE_PAD_GPIO6_14	RW	32	0x0000 1688	0x4A00 3688
CTRL_CORE_PAD_GPIO6_15	RW	32	0x0000 168C	0x4A00 368C
CTRL_CORE_PAD_GPIO6_16	RW	32	0x0000 1690	0x4A00 3690
CTRL_CORE_PAD_XREF_CLK0	RW	32	0x0000 1694	0x4A00 3694
CTRL_CORE_PAD_XREF_CLK1	RW	32	0x0000 1698	0x4A00 3698
CTRL_CORE_PAD_XREF_CLK2	RW	32	0x0000 169C	0x4A00 369C
CTRL_CORE_PAD_XREF_CLK3	RW	32	0x0000 16A0	0x4A00 36A0
CTRL_CORE_PAD_MCASP1_ACLKX	RW	32	0x0000 16A4	0x4A00 36A4
CTRL_CORE_PAD_MCASP1_FSX	RW	32	0x0000 16A8	0x4A00 36A8
CTRL_CORE_PAD_MCASP1_ACLKR	RW	32	0x0000 16AC	0x4A00 36AC
CTRL_CORE_PAD_MCASP1_FSR	RW	32	0x0000 16B0	0x4A00 36B0
CTRL_CORE_PAD_MCASP1_AXR0	RW	32	0x0000 16B4	0x4A00 36B4
CTRL_CORE_PAD_MCASP1_AXR1	RW	32	0x0000 16B8	0x4A00 36B8
CTRL_CORE_PAD_MCASP1_AXR2	RW	32	0x0000 16BC	0x4A00 36BC
CTRL_CORE_PAD_MCASP1_AXR3	RW	32	0x0000 16C0	0x4A00 36C0
CTRL_CORE_PAD_MCASP1_AXR4	RW	32	0x0000 16C4	0x4A00 36C4
CTRL_CORE_PAD_MCASP1_AXR5	RW	32	0x0000 16C8	0x4A00 36C8
CTRL_CORE_PAD_MCASP1_AXR6	RW	32	0x0000 16CC	0x4A00 36CC
CTRL_CORE_PAD_MCASP1_AXR7	RW	32	0x0000 16D0	0x4A00 36D0
CTRL_CORE_PAD_MCASP1_AXR8	RW	32	0x0000 16D4	0x4A00 36D4
CTRL_CORE_PAD_MCASP1_AXR9	RW	32	0x0000 16D8	0x4A00 36D8
CTRL_CORE_PAD_MCASP1_AXR10	RW	32	0x0000 16DC	0x4A00 36DC
CTRL_CORE_PAD_MCASP1_AXR11	RW	32	0x0000 16E0	0x4A00 36E0
CTRL_CORE_PAD_MCASP1_AXR12	RW	32	0x0000 16E4	0x4A00 36E4
CTRL_CORE_PAD_MCASP1_AXR13	RW	32	0x0000 16E8	0x4A00 36E8
CTRL_CORE_PAD_MCASP1_AXR14	RW	32	0x0000 16EC	0x4A00 36EC
CTRL_CORE_PAD_MCASP1_AXR15	RW	32	0x0000 16F0	0x4A00 36F0
CTRL_CORE_PAD_MCASP2_ACLKX	RW	32	0x0000 16F4	0x4A00 36F4
CTRL_CORE_PAD_MCASP2_FSX	RW	32	0x0000 16F8	0x4A00 36F8
CTRL_CORE_PAD_MCASP2_ACLKR	RW	32	0x0000 16FC	0x4A00 36FC
CTRL_CORE_PAD_MCASP2_FSR	RW	32	0x0000 1700	0x4A00 3700
CTRL_CORE_PAD_MCASP2_AXR0	RW	32	0x0000 1704	0x4A00 3704
CTRL_CORE_PAD_MCASP2_AXR1	RW	32	0x0000 1708	0x4A00 3708
CTRL_CORE_PAD_MCASP2_AXR2	RW	32	0x0000 170C	0x4A00 370C
CTRL_CORE_PAD_MCASP2_AXR3	RW	32	0x0000 1710	0x4A00 3710
CTRL_CORE_PAD_MCASP2_AXR4	RW	32	0x0000 1714	0x4A00 3714
CTRL_CORE_PAD_MCASP2_AXR5	RW	32	0x0000 1718	0x4A00 3718
CTRL_CORE_PAD_MCASP2_AXR6	RW	32	0x0000 171C	0x4A00 371C
CTRL_CORE_PAD_MCASP2_AXR7	RW	32	0x0000 1720	0x4A00 3720
CTRL_CORE_PAD_MCASP3_ACLKX	RW	32	0x0000 1724	0x4A00 3724
CTRL_CORE_PAD_MCASP3_FSX	RW	32	0x0000 1728	0x4A00 3728

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_PAD_MCASP3_AXR0	RW	32	0x0000 172C	0x4A00 372C
CTRL_CORE_PAD_MCASP3_AXR1	RW	32	0x0000 1730	0x4A00 3730
CTRL_CORE_PAD_MCASP4_ACLKX	RW	32	0x0000 1734	0x4A00 3734
CTRL_CORE_PAD_MCASP4_FSX	RW	32	0x0000 1738	0x4A00 3738
CTRL_CORE_PAD_MCASP4_AXR0	RW	32	0x0000 173C	0x4A00 373C
CTRL_CORE_PAD_MCASP4_AXR1	RW	32	0x0000 1740	0x4A00 3740
CTRL_CORE_PAD_MCASP5_ACLKX	RW	32	0x0000 1744	0x4A00 3744
CTRL_CORE_PAD_MCASP5_FSX	RW	32	0x0000 1748	0x4A00 3748
CTRL_CORE_PAD_MCASP5_AXR0	RW	32	0x0000 174C	0x4A00 374C
CTRL_CORE_PAD_MCASP5_AXR1	RW	32	0x0000 1750	0x4A00 3750
CTRL_CORE_PAD_MMC1_CLK	RW	32	0x0000 1754	0x4A00 3754
CTRL_CORE_PAD_MMC1_CMD	RW	32	0x0000 1758	0x4A00 3758
CTRL_CORE_PAD_MMC1_DAT0	RW	32	0x0000 175C	0x4A00 375C
CTRL_CORE_PAD_MMC1_DAT1	RW	32	0x0000 1760	0x4A00 3760
CTRL_CORE_PAD_MMC1_DAT2	RW	32	0x0000 1764	0x4A00 3764
CTRL_CORE_PAD_MMC1_DAT3	RW	32	0x0000 1768	0x4A00 3768
CTRL_CORE_PAD_MMC1_SDCD	RW	32	0x0000 176C	0x4A00 376C
CTRL_CORE_PAD_MMC1_SDWP	RW	32	0x0000 1770	0x4A00 3770
CTRL_CORE_PAD_GPIO6_10	RW	32	0x0000 1774	0x4A00 3774
CTRL_CORE_PAD_GPIO6_11	RW	32	0x0000 1778	0x4A00 3778
CTRL_CORE_PAD_MMC3_CLK	RW	32	0x0000 177C	0x4A00 377C
CTRL_CORE_PAD_MMC3_CMD	RW	32	0x0000 1780	0x4A00 3780
CTRL_CORE_PAD_MMC3_DAT0	RW	32	0x0000 1784	0x4A00 3784
CTRL_CORE_PAD_MMC3_DAT1	RW	32	0x0000 1788	0x4A00 3788
CTRL_CORE_PAD_MMC3_DAT2	RW	32	0x0000 178C	0x4A00 378C
CTRL_CORE_PAD_MMC3_DAT3	RW	32	0x0000 1790	0x4A00 3790
CTRL_CORE_PAD_MMC3_DAT4	RW	32	0x0000 1794	0x4A00 3794
CTRL_CORE_PAD_MMC3_DAT5	RW	32	0x0000 1798	0x4A00 3798
CTRL_CORE_PAD_MMC3_DAT6	RW	32	0x0000 179C	0x4A00 379C
CTRL_CORE_PAD_MMC3_DAT7	RW	32	0x0000 17A0	0x4A00 37A0
CTRL_CORE_PAD_SPI1_SCLK	RW	32	0x0000 17A4	0x4A00 37A4
CTRL_CORE_PAD_SPI1_D1	RW	32	0x0000 17A8	0x4A00 37A8
CTRL_CORE_PAD_SPI1_D0	RW	32	0x0000 17AC	0x4A00 37AC
CTRL_CORE_PAD_SPI1_CS0	RW	32	0x0000 17B0	0x4A00 37B0
CTRL_CORE_PAD_SPI1_CS1	RW	32	0x0000 17B4	0x4A00 37B4
CTRL_CORE_PAD_SPI1_CS2	RW	32	0x0000 17B8	0x4A00 37B8
CTRL_CORE_PAD_SPI1_CS3	RW	32	0x0000 17BC	0x4A00 37BC
CTRL_CORE_PAD_SPI2_SCLK	RW	32	0x0000 17C0	0x4A00 37C0
CTRL_CORE_PAD_SPI2_D1	RW	32	0x0000 17C4	0x4A00 37C4
CTRL_CORE_PAD_SPI2_D0	RW	32	0x0000 17C8	0x4A00 37C8
CTRL_CORE_PAD_SPI2_CS0	RW	32	0x0000 17CC	0x4A00 37CC
CTRL_CORE_PAD_DCAN1_TX	RW	32	0x0000 17D0	0x4A00 37D0
CTRL_CORE_PAD_DCAN1_RX	RW	32	0x0000 17D4	0x4A00 37D4
RESERVED	R	32	0x0000 17D8	0x4A00 37D8
RESERVED	R	32	0x0000 17DC	0x4A00 37DC
CTRL_CORE_PAD_UART1_RXD	RW	32	0x0000 17E0	0x4A00 37E0
CTRL_CORE_PAD_UART1_TXD	RW	32	0x0000 17E4	0x4A00 37E4
CTRL_CORE_PAD_UART1_CTSN	RW	32	0x0000 17E8	0x4A00 37E8
CTRL_CORE_PAD_UART1_RTSN	RW	32	0x0000 17EC	0x4A00 37EC

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_ CORE Base Address
CTRL_CORE_PAD_UART2_RXD	RW	32	0x0000 17F0	0x4A00 37F0
CTRL_CORE_PAD_UART2_TXD	RW	32	0x0000 17F4	0x4A00 37F4
CTRL_CORE_PAD_UART2_CTSN	RW	32	0x0000 17F8	0x4A00 37F8
CTRL_CORE_PAD_UART2_RTSN	RW	32	0x0000 17FC	0x4A00 37FC
CTRL_CORE_PAD_I2C1_SDA	RW	32	0x0000 1800	0x4A00 3800
CTRL_CORE_PAD_I2C1_SCL	RW	32	0x0000 1804	0x4A00 3804
CTRL_CORE_PAD_I2C2_SDA	RW	32	0x0000 1808	0x4A00 3808
CTRL_CORE_PAD_I2C2_SCL	RW	32	0x0000 180C	0x4A00 380C
RESERVED	R	32	0x0000 1810	0x4A00 3810
RESERVED	R	32	0x0000 1814	0x4A00 3814
CTRL_CORE_PAD_WAKEUP0	RW	32	0x0000 1818	0x4A00 3818
CTRL_CORE_PAD_WAKEUP1	RW	32	0x0000 181C	0x4A00 381C
CTRL_CORE_PAD_WAKEUP2	RW	32	0x0000 1820	0x4A00 3820
CTRL_CORE_PAD_WAKEUP3	RW	32	0x0000 1824	0x4A00 3824
CTRL_CORE_PAD_ON_OFF	RW	32	0x0000 1828	0x4A00 3828
CTRL_CORE_PAD_RTC_PORZ	RW	32	0x0000 182C	0x4A00 382C
CTRL_CORE_PAD_TMS	RW	32	0x0000 1830	0x4A00 3830
CTRL_CORE_PAD_TDI	RW	32	0x0000 1834	0x4A00 3834
CTRL_CORE_PAD_TDO	RW	32	0x0000 1838	0x4A00 3838
CTRL_CORE_PAD_TCLK	RW	32	0x0000 183C	0x4A00 383C
CTRL_CORE_PAD_TRSTN	RW	32	0x0000 1840	0x4A00 3840
CTRL_CORE_PAD_RTCK	RW	32	0x0000 1844	0x4A00 3844
CTRL_CORE_PAD_EMU0	RW	32	0x0000 1848	0x4A00 3848
CTRL_CORE_PAD_EMU1	RW	32	0x0000 184C	0x4A00 384C
RESERVED	R	32	0x0000 1850	0x4A00 3850
RESERVED	R	32	0x0000 1854	0x4A00 3854
RESERVED	R	32	0x0000 1858	0x4A00 3858
CTRL_CORE_PAD_RESETN	RW	32	0x0000 185C	0x4A00 385C
CTRL_CORE_PAD_NMIN_DSP	RW	32	0x0000 1860	0x4A00 3860
CTRL_CORE_PAD_RSTOUTN	RW	32	0x0000 1864	0x4A00 3864
CTRL_CORE_PADCONF_WAKEUPEVENT_0	R	32	0x0000 1868	0x4A00 3868
CTRL_CORE_PADCONF_WAKEUPEVENT_1	R	32	0x0000 186C	0x4A00 386C
CTRL_CORE_PADCONF_WAKEUPEVENT_2	R	32	0x0000 1870	0x4A00 3870
CTRL_CORE_PADCONF_WAKEUPEVENT_3	R	32	0x0000 1874	0x4A00 3874
CTRL_CORE_PADCONF_WAKEUPEVENT_4	R	32	0x0000 1878	0x4A00 3878
CTRL_CORE_PADCONF_WAKEUPEVENT_5	R	32	0x0000 187C	0x4A00 387C
CTRL_CORE_PADCONF_WAKEUPEVENT_6	R	32	0x0000 1880	0x4A00 3880
CTRL_CORE_PADCONF_WAKEUPEVENT_7	R	32	0x0000 1884	0x4A00 3884
CTRL_CORE_PADCONF_WAKEUPEVENT_8	R	32	0x0000 1888	0x4A00 3888
RESERVED_j (j= 0 to 63)	R	32	0x0000 1A00 + (j*4)	0x4A00 3A00 + (j*4)
RESERVED	R	32	0x0000 1B00	0x4A00 3B00
RESERVED	R	32	0x0000 1B04	0x4A00 3B04
CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_2	R	32	0x0000 1B08	0x4A00 3B08
CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_3	R	32	0x0000 1B0C	0x4A00 3B0C
CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_4	R	32	0x0000 1B10	0x4A00 3B10
CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_5	R	32	0x0000 1B14	0x4A00 3B14
RESERVED	R	32	0x0000 1B18	0x4A00 3B18
CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_1	R	32	0x0000 1B1C	0x4A00 3B1C
CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_2	R	32	0x0000 1B20	0x4A00 3B20

Table 18-28. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_3	R	32	0x0000 1B24	0x4A00 3B24
CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_4	R	32	0x0000 1B28	0x4A00 3B28
CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_5	R	32	0x0000 1B2C	0x4A00 3B2C
RESERVED	R	32	0x0000 1B30	0x4A00 3B30
RESERVED	R	32	0x0000 1B34	0x4A00 3B34
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_0	R	32	0x0000 1B38	0x4A00 3B38
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_1	R	32	0x0000 1B3C	0x4A00 3B3C
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_2	R	32	0x0000 1B40	0x4A00 3B40
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_3	R	32	0x0000 1B44	0x4A00 3B44
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_4	R	32	0x0000 1B48	0x4A00 3B48
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_0	R	32	0x0000 1B4C	0x4A00 3B4C
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_1	R	32	0x0000 1B50	0x4A00 3B50
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_2	R	32	0x0000 1B54	0x4A00 3B54
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_3	R	32	0x0000 1B58	0x4A00 3B58
CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_4	R	32	0x0000 1B5C	0x4A00 3B5C
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_0	R	32	0x0000 1B60	0x4A00 3B60
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_1	R	32	0x0000 1B64	0x4A00 3B64
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_2	R	32	0x0000 1B68	0x4A00 3B68
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_3	R	32	0x0000 1B6C	0x4A00 3B6C
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_4	R	32	0x0000 1B70	0x4A00 3B70
CTRL_CORE_LDOSRAM_CORE_4_VOLTAGE_CTRL	RW	32	0x0000 1B74	0x4A00 3B74
CTRL_CORE_LDOSRAM_CORE_5_VOLTAGE_CTRL	RW	32	0x0000 1B78	0x4A00 3B78
CTRL_CORE_LDOSRAM_DSPEVE_2_VOLTAGE_CTRL	RW	32	0x0000 1B7C	0x4A00 3B7C
RESERVED_i (i = 0 to 32)	R	32	0x0000 1B80 + (i*4)	0x4A00 3B80 + (i*4)
CTRL_CORE_SMA_SW_2	RW	32	0x0000 1C04	0x4A00 3C04
CTRL_CORE_SMA_SW_3	RW	32	0x0000 1C08	0x4A00 3C08
RESERVED	R	32	0x0000 1C0C	0x4A00 3C0C
RESERVED	R	32	0x0000 1C10	0x4A00 3C10
CTRL_CORE_SMA_SW_6	RW	32	0x0000 1C14	0x4A00 3C14
CTRL_CORE_SMA_SW_7	RW	32	0x0000 1C18	0x4A00 3C18
CTRL_CORE_SMA_SW_8	RW	32	0x0000 1C1C	0x4A00 3C1C
CTRL_CORE_SMA_SW_9	RW	32	0x0000 1C20	0x4A00 3C20
CTRL_CORE_PCIESS1_PCS1	RW	32	0x0000 1C24	0x4A00 3C24
CTRL_CORE_PCIESS1_PCS2	RW	32	0x0000 1C28	0x4A00 3C28
CTRL_CORE_PCIESS2_PCS1	RW	32	0x0000 1C2C	0x4A00 3C2C
CTRL_CORE_PCIESS2_PCS2	RW	32	0x0000 1C30	0x4A00 3C30
CTRL_CORE_PCIE_PCS	RW	32	0x0000 1C34	0x4A00 3C34
CTRL_CORE_PCIE_PCS_REVISION	R	32	0x0000 1C38	0x4A00 3C38
CTRL_CORE_PCIE_CONTROL	RW	32	0x0000 1C3C	0x4A00 3C3C
CTRL_CORE_PHY_POWER_PCIESS1	RW	32	0x0000 1C40	0x4A00 3C40
CTRL_CORE_PHY_POWER_PCIESS2	RW	32	0x0000 1C44	0x4A00 3C44

18.5.4 CTRL_MODULE_CORE Register Description

Table 18-29. CTRL_CORE_STATUS

Address Offset	0x0000 0134		
Physical Address	0x4A00 2134	**Instance**	CTRL_MODULE_CORE
Description	Control Module Status Register		

Table 18-29. CTRL_CORE_STATUS (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							DEVICE_TYPE			RESERVED					

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:6	DEVICE_TYPE	Device type captured at reset time. Device type value sampled at power-on reset. Read 0x3 = General Purpose (GP)	R	0x3
5:0	RESERVED	Reserved	R	0x0

Table 18-30. CTRL_CORE_SEC_ERR_STATUS_FUNC_1

Address Offset	0x0000 0148		
Physical Address	0x4A00 2148	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status functional Register 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		EVE2_FW_ERROR	EVE1_FW_ERROR	RESERVED				BB2D_FW_ERROR	L4_WAKEUP_FW_ERROR	RESERVED			DEBUGSS_FW_ERROR	L4_CONFIG_FW_ERROR	L4_PERIPH1_FW_ERROR	RESERVED	DSS_FW_ERROR	GPU_FW_ERROR	RESERVED						IVAHD_SL2_FW_ERROR	IPU1_FW_ERROR	IVAHD_FW_ERROR	EMIF_FW_ERROR	GPMC_FW_ERROR	L3RAM1_FW_ERROR	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	EVE2_FW_ERROR	EVE2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
28	EVE1_FW_ERROR	EVE1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
27:24	RESERVED		R	0x0
23	BB2D_FW_ERROR	BB2D firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
22	L4_WAKEUP_FW_ERROR	L4 wakeup firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
21:19	RESERVED		R	0x0
18	DEBUGSS_FW_ERROR	DebugSS firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
17	L4_CONFIG_FW_ERROR	L4 config firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	L4_PERIPH1_FW_ERROR	L4 periph1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15	RESERVED		R	0x0
14	DSS_FW_ERROR	DSS firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
13	GPU_FW_ERROR	GPU firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
12:7	RESERVED		R	0x0
6	IVAHD_SL2_FW_ERROR	IVAHD SL2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
5	IPU1_FW_ERROR	IPU1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	IVAHD_FW_ERROR	IVAHD firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
3	EMIF_FW_ERROR	EMIF firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
2	GPMC_FW_ERROR	GPMC firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
1	L3RAM1_FW_ERROR	L3RAM1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
0	RESERVED		R	0x0

Table 18-31. CTRL_CORE_SEC_ERR_STATUS_DEBUG_1

Address Offset	0x0000 0150		
Physical Address	0x4A00 2150	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status Debug Register 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	EVE2_DBGFW_ERROR	EVE1_DBGFW_ERROR	RESERVED	BB2D_DBGFW_ERROR	L4_WAKEUP_DBGFW_ERROR	RESERVED	DEBUGSS_DBGFW_ERROR	L4_CONFIG_DBGFW_ERROR	L4_PERIPH1_DBGFW_ERROR	RESERVED	DSS_DBGFW_ERROR	GPU_DBGFW_ERROR	RESERVED	IVAHD_SL2_DBGFW_ERROR	IPU1_DBGFW_ERROR	IVAHD_DBGFW_ERROR	EMIF_DBGFW_ERROR	GPMC_DBGFW_ERROR	L3RAM1_DBGFW_ERROR	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	EVE2_DBGFW_ERROR	EVE2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
28	EVE1_DBGFW_ERROR	EVE1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
27:24	RESERVED		R	0x0
23	BB2D_DBGFW_ERROR	BB2D firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
22	L4_WAKEUP_DBGFW_ERROR	L4 wakeup firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
21:19	RESERVED		R	0x0
18	DEBUGSS_DBGFW_ERROR	DebugSS firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
17	L4_CONFIG_DBGFW_ERROR	L4 config firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	L4_PERIPH1_DBGFW_ERROR	L4 periph1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15	RESERVED		R	0x0
14	DSS_DBGFW_ERROR	DSS debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
13	GPU_DBGFW_ERROR	GPU debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
12:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6	IVAHD_SL2_DBGFW_ERROR	IVAHD SL2 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
5	IPU1_DBGFW_ERROR	IPU1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	IVAHD_DBGFW_ERROR	IVAHD debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
3	EMIF_DBGFW_ERROR	EMIF debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
2	GPMC_DBGFW_ERROR	GPMC debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
1	L3RAM1_DBGFW_ERROR	L3RAM1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
0	RESERVED		R	0x0

Table 18-32. CTRL_CORE_MPU_FORCEWRNP

Address Offset	0x0000 015C		
Physical Address	0x4A00 215C	**Instance**	CTRL_MODULE_CORE
Description	FORCE WRITE NON POSTED		
Type	RW		

31:1	0
RESERVED	MPU_FORCEWRNP

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MPU_FORCEWRNP	Force mpu write non posted transactions 0x0 = disable force wrnp 0x1 = force wrnp	RW	0x0

Table 18-33. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_0

Address Offset	0x0000 0194		
Physical Address	0x4A00 2194	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STD_FUSE_OPP_VDD_GPU_0

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_0		R	0x0

Table 18-34. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_1

Address Offset	0x0000 0198		
Physical Address	0x4A00 2198	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_1		R	0x0

Table 18-35. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_2

Address Offset	0x0000 019C		
Physical Address	0x4A00 219C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_2		R	0x0

Table 18-36. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_3

Address Offset	0x0000 01A0		
Physical Address	0x4A00 21A0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_3		R	0x0

Table 18-37. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_4

Address Offset	0x0000 01A4		
Physical Address	0x4A00 21A4	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		

Table 18-37. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_4 (continued)

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_4		R	0x0

Table 18-38. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_5

Address Offset 0x0000 01A8

Physical Address 0x4A00 21A8 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_GPU [191:160]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_5																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_5		R	0x0

Table 18-39. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_0

Address Offset 0x0000 01AC

Physical Address 0x4A00 21AC **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_MPU [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_0		R	0x0

Table 18-40. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_1

Address Offset 0x0000 01B0

Physical Address 0x4A00 21B0 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_MPU [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_1		R	0x0

Table 18-41. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_2

Address Offset 0x0000 01B4

Table 18-41. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_2 (continued)

Physical Address	0x4A00 21B4	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_2		R	0x0

Table 18-42. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_3

Address Offset	0x0000 01B8		
Physical Address	0x4A00 21B8	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_3		R	0x0

Table 18-43. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_4

Address Offset	0x0000 01BC		
Physical Address	0x4A00 21BC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_4		R	0x0

Table 18-44. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_5

Address Offset	0x0000 01C0		
Physical Address	0x4A00 21C0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [191:160]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_5																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_5		R	0x0

Table 18-45. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_6

Address Offset	0x0000 01C4		
Physical Address	0x4A00 21C4	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [223:192]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_6																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_6		R	0x0

Table 18-46. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_7

Address Offset	0x0000 01C8		
Physical Address	0x4A00 21C8	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [255:224]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_7																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_7		R	0x0

Table 18-47. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_0

Address Offset	0x0000 01CC		
Physical Address	0x4A00 21CC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_0		R	0x0

Table 18-48. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_1

Address Offset	0x0000 01D0		
Physical Address	0x4A00 21D0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STD_FUSE_OPP_VDD_CORE_1

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_1		R	0x0

Table 18-49. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_2

Address Offset	0x0000 01D4		
Physical Address	0x4A00 21D4	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_2		R	0x0

Table 18-50. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_3

Address Offset	0x0000 01D8		
Physical Address	0x4A00 21D8	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_3		R	0x0

Table 18-51. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_4

Address Offset	0x0000 01DC		
Physical Address	0x4A00 21DC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_4		R	0x0

Table 18-52. CTRL_CORE_STD_FUSE_OPP_BGAP_GPU

Address Offset	0x0000 01E0		
Physical Address	0x4A00 21E0	**Instance**	CTRL_MODULE_CORE
Description	Trim values for GPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.		

Table 18-52. CTRL_CORE_STD_FUSE_OPP_BGAP_GPU (continued)

Type R

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
STD_FUSE_OPP_BGAP_GPU_0	STD_FUSE_OPP_BGAP_GPU_1	STD_FUSE_OPP_BGAP_GPU_2	STD_FUSE_OPP_BGAP_GPU_3

Bits	Field Name	Description	Type	Reset
31:24	STD_FUSE_OPP_BGAP_GPU_0	Trim values for GPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
23:16	STD_FUSE_OPP_BGAP_GPU_1	Trim values for GPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
15:8	STD_FUSE_OPP_BGAP_GPU_2	Trim values for GPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
7:0	STD_FUSE_OPP_BGAP_GPU_3	Trim values for GPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Table 18-53. CTRL_CORE_STD_FUSE_OPP_BGAP_MPU

Address Offset 0x0000 01E4

Physical Address 0x4A00 21E4 **Instance** CTRL_MODULE_CORE

Description Trim values for MPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.

Type R

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
STD_FUSE_OPP_BGAP_MPU_0	STD_FUSE_OPP_BGAP_MPU_1	STD_FUSE_OPP_BGAP_MPU_2	STD_FUSE_OPP_BGAP_MPU_3

Bits	Field Name	Description	Type	Reset
31:24	STD_FUSE_OPP_BGAP_MPU_0	Trim values for MPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
23:16	STD_FUSE_OPP_BGAP_MPU_1	Trim values for MPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
15:8	STD_FUSE_OPP_BGAP_MPU_2	Trim values for MPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
7:0	STD_FUSE_OPP_BGAP_MPU_3	Trim values for MPU associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Table 18-54. CTRL_CORE_STD_FUSE_OPP_BGAP_CORE

Address Offset 0x0000 01E8

Physical Address 0x4A00 21E8 **Instance** CTRL_MODULE_CORE

Description Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.

Type R

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
STD_FUSE_OPP_BGAP_CORE_0	STD_FUSE_OPP_BGAP_CORE_1	STD_FUSE_OPP_BGAP_CORE_2	STD_FUSE_OPP_BGAP_CORE_3

Bits	Field Name	Description	Type	Reset
31:24	STD_FUSE_OPP_BGAP_CORE_0	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
23:16	STD_FUSE_OPP_BGAP_CORE_1	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Bits	Field Name	Description	Type	Reset
15:8	STD_FUSE_OPP_BGAP_CORE_2	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
7:0	STD_FUSE_OPP_BGAP_CORE_3	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Table 18-55. CTRL_CORE_STD_FUSE_OPP_BGAP_MPU23

Address Offset	0x0000 01EC		
Physical Address	0x4A00 21EC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP BGAP. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_BGAP_MPU3																STD_FUSE_OPP_BGAP_MPU2															

Bits	Field Name	Description	Type	Reset
31:16	STD_FUSE_OPP_BGAP_MPU3		R	0x0
15:0	STD_FUSE_OPP_BGAP_MPU2		R	0x0

Table 18-56. CTRL_CORE_STD_FUSE_MPK_0

Address Offset	0x0000 0220		
Physical Address	0x4A00 2220	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_0		R	0x0

Table 18-57. CTRL_CORE_STD_FUSE_MPK_1

Address Offset	0x0000 0224		
Physical Address	0x4A00 2224	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_1		R	0x0

Table 18-58. CTRL_CORE_STD_FUSE_MPK_2

Address Offset	0x0000 0228		
Physical Address	0x4A00 2228	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		

Table 18-58. CTRL_CORE_STD_FUSE_MPK_2 (continued)

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_2		R	0x0

Table 18-59. CTRL_CORE_STD_FUSE_MPK_3

Address Offset 0x0000 022C

Physical Address 0x4A00 222C **Instance** CTRL_MODULE_CORE

Description Standard Fuse keys. Root_public_key_hash [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_3		R	0x0

Table 18-60. CTRL_CORE_STD_FUSE_MPK_4

Address Offset 0x0000 0230

Physical Address 0x4A00 2230 **Instance** CTRL_MODULE_CORE

Description Standard Fuse keys. Root_public_key_hash [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_4		R	0x0

Table 18-61. CTRL_CORE_STD_FUSE_MPK_5

Address Offset 0x0000 0234

Physical Address 0x4A00 2234 **Instance** CTRL_MODULE_CORE

Description Standard Fuse keys. Root_public_key_hash [191:160]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_5																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_5		R	0x0

Table 18-62. CTRL_CORE_STD_FUSE_MPK_6

Address Offset 0x0000 0238

Physical Address 0x4A00 2238 **Instance** CTRL_MODULE_CORE

Description Standard Fuse keys. Root_public_key_hash [223:192]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Table 18-62. CTRL_CORE_STD_FUSE_MPK_6 (continued)

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_6																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_6		R	0x0

Table 18-63. CTRL_CORE_STD_FUSE_MPK_7

Address Offset 0x0000 023C

Physical Address 0x4A00 223C **Instance** CTRL_MODULE_CORE

Description Standard Fuse keys. Root_public_key_hash [255:224]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_7																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_7		R	0x0

Table 18-64. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_0

Address Offset 0x0000 0240

Physical Address 0x4A00 2240 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_GPU [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_LVT_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_LVT_0		R	0x0

Table 18-65. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_1

Address Offset 0x0000 0244

Physical Address 0x4A00 2244 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_GPU [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_LVT_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_LVT_1		R	0x0

Table 18-66. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_2

Address Offset 0x0000 0248

Physical Address 0x4A00 2248 **Instance** CTRL_MODULE_CORE

Table 18-66. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_2 (continued)

Description Standard Fuse OPP VDD_GPU [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_LVT_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_LVT_2		R	0x0

Table 18-67. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_3

Address Offset 0x0000 024C

Physical Address 0x4A00 224C **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_GPU [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_LVT_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_LVT_3		R	0x0

Table 18-68. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_4

Address Offset 0x0000 0250

Physical Address 0x4A00 2250 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_GPU [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_LVT_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_LVT_4		R	0x0

Table 18-69. CTRL_CORE_STD_FUSE_OPP_VDD_GPU_LVT_5

Address Offset 0x0000 0254

Physical Address 0x4A00 2254 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_GPU [191:160]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_GPU_LVT_5																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_GPU_LVT_5		R	0x0

Table 18-70. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_0

Address Offset	0x0000 0258		
Physical Address	0x4A00 2258	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_LVT_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_0		R	0x0

Table 18-71. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_1

Address Offset	0x0000 025C		
Physical Address	0x4A00 225C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_LVT_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_1		R	0x0

Table 18-72. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_2

Address Offset	0x0000 0260		
Physical Address	0x4A00 2260	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_LVT_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_2		R	0x0

Table 18-73. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_3

Address Offset	0x0000 0264		
Physical Address	0x4A00 2264	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_LVT_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_3		R	0x0

Table 18-74. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_4

Address Offset	0x0000 0268		
Physical Address	0x4A00 2268	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_LVT_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_4		R	0x0

Table 18-75. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_5

Address Offset	0x0000 026C		
Physical Address	0x4A00 226C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [191:160]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_LVT_5																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_5		R	0x0

Table 18-76. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_6

Address Offset	0x0000 0270		
Physical Address	0x4A00 2270	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [223:192]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_MPU_LVT_6																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_6		R	0x0

Table 18-77. CTRL_CORE_STD_FUSE_OPP_VDD_MPU_LVT_7

Address Offset	0x0000 0274		
Physical Address	0x4A00 2274	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_MPU [255:224]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STD_FUSE_OPP_VDD_MPU_LVT_7

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_MPU_LVT_7		R	0x0

Table 18-78. CTRL_CORE_CUST_FUSE_SWRV_0

Address Offset	0x0000 02BC		
Physical Address	0x4A00 22BC	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [031:000] (16 bits upper Redundant field) [FIELD OVERFLOW]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_0																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_0		R	0x0

Table 18-79. CTRL_CORE_CUST_FUSE_SWRV_1

Address Offset	0x0000 02C0		
Physical Address	0x4A00 22C0	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [063:032] (16 bits upper Redundant field) [FIELD F]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_1																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_1		R	0x0

Table 18-80. CTRL_CORE_CUST_FUSE_SWRV_2

Address Offset	0x0000 02C4		
Physical Address	0x4A00 22C4	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [095:064] (16 bits upper Redundant field) [FIELD E]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_2																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_2		R	0x0

Table 18-81. CTRL_CORE_CUST_FUSE_SWRV_3

Address Offset	0x0000 02C8		
Physical Address	0x4A00 22C8	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [127:096] (16 bits upper Redundant field) [FIELD D]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		

Table 18-81. CTRL_CORE_CUST_FUSE_SWRV_3 (continued)

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_3																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_3		R	0x0

Table 18-82. CTRL_CORE_CUST_FUSE_SWRV_4

Address Offset 0x0000 02CC

Physical Address 0x4A00 22CC **Instance** CTRL_MODULE_CORE

Description Customer Fuse keys. Software Version Control [159:127] (16 bits upper Redundant field) [FIELD C]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_4																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_4		R	0x0

Table 18-83. CTRL_CORE_CUST_FUSE_SWRV_5

Address Offset 0x0000 02D0

Physical Address 0x4A00 22D0 **Instance** CTRL_MODULE_CORE

Description Customer Fuse keys. Software Version Control [191:160] (16 bits upper Redundant field) [FIELD B]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_5																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_5		R	0x0

Table 18-84. CTRL_CORE_CUST_FUSE_SWRV_6

Address Offset 0x0000 02D4

Physical Address 0x4A00 22D4 **Instance** CTRL_MODULE_CORE

Description Customer Fuse keys. Software Version Control [223:192] (16 bits upper Redundant field) [FIELD A]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_6																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_6		R	0x0

Table 18-85. CTRL_CORE_DEV_CONF

Address Offset 0x0000 0300

Physical Address 0x4A00 2300 **Instance** CTRL_MODULE_CORE

Table 18-85. CTRL_CORE_DEV_CONF (continued)

Description	This register is used to power down the USB2_PHY1
Type	RW

31–1	0
RESERVED	USBPHY_PD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	USBPHY_PD	Power down the entire USB2_PHY1 (data, common module and UTMI). 0x0: Normal operation 0x1: Power down the USB2_PHY1	RW	0x0

Table 18-86. CTRL_CORE_TEMP_SENSOR_MPU

Address Offset	0x0000 032C		
Physical Address	0x4A00 232C	**Instance**	CTRL_MODULE_CORE
Description	Control VBGAPTS temperature sensor and thermal comparator shutdown register		
Type	R		

31–12	11	10	9–0
RESERVED	BGAP_TMPSOFF_MPU	BGAP_EOCZ_MPU	BGAP_DTEMP_MPU

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x0
11	BGAP_TMPSOFF_MPU	This bit indicates the temperature sensor state. 0x0: temperature sensor is ON 0x1: temperature sensor is OFF **NOTE:** Software doesn't take care of this bit to get the temperature data. Only the BGAP_EOCZ_MPU bit is needed.	R	0x1
10	BGAP_EOCZ_MPU	ADC End of Conversion. Active low, when BGAP_DTEMP_MPU is valid.	R	0x0
9:0	BGAP_DTEMP_MPU	Temperature data from the ADC. Valid if EOCZ is low.	R	0x0

Table 18-87. CTRL_CORE_TEMP_SENSOR_GPU

Address Offset	0x0000 0330		
Physical Address	0x4A00 2330	**Instance**	CTRL_MODULE_CORE
Description	Control VBGAPTS temperature sensor and thermal comparator shutdown register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	BGAP_TMPSOFF_GPU	BGAP_EOCZ_GPU	BGAP_DTEMP_GPU

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x0
11	BGAP_TMPSOFF_GPU	This bit indicates the temperature sensor state. 0x0: temperature sensor is ON 0x1: temperature sensor is OFF **NOTE:** Software doesn't take care of this bit to get the temperature data. Only the BGAP_EOCZ_GPU bit is needed.	R	0x1
10	BGAP_EOCZ_GPU	ADC End of Conversion. Active low, when BGAP_DTEMP_GPU is valid.	R	0x0
9:0	BGAP_DTEMP_GPU	Temperature data from the ADC. Valid if EOCZ is low.	R	0x0

Table 18-88. CTRL_CORE_TEMP_SENSOR_CORE

Address Offset	0x0000 0334		
Physical Address	0x4A00 2334	**Instance**	CTRL_MODULE_CORE
Description	Control VBGAPTS temperature sensor and thermal comparator shutdown register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				BGAP_TMPSOFF_CORE	BGAP_EOCZ_CORE	BGAP_DTEMP_CORE									

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x0
11	BGAP_TMPSOFF_CORE	This bit indicates the temperature sensor state. 0x0: temperature sensor is ON 0x1: temperature sensor is OFF **NOTE:** Software doesn't take care of this bit to get the temperature data. Only the BGAP_EOCZ_CORE bit is needed.	R	0x1
10	BGAP_EOCZ_CORE	ADC End of Conversion. Active low, when BGAP_DTEMP_CORE is valid.	R	0x0
9:0	BGAP_DTEMP_CORE	Temperature data from the ADC. Valid if EOCZ is low.	R	0x0

Table 18-89. CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR

Address Offset	0x0000 0358		
Physical Address	0x4A00 2358	**Instance**	CTRL_MODULE_CORE

Table 18-89. CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR (continued)

Description	Cortex M4 register
Type	RW

31–20	19–0
RESERVED	CORTEX_M4_MMUADDRTRANSLTR

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x0
19:0	CORTEX_M4_MMUADDR TRANSLTR	Used to save the IPU AMMU translated/boot address	RW	0x0

Table 18-90. CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR

Address Offset	0x0000 035C		
Physical Address	0x4A00 235C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–20	19–0
RESERVED	CORTEX_M4_MMUADDRLOGICTR

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x0
19:0	CORTEX_M4_MMUADDR LOGICTR	Used to save the IPU AMMU logical source address	RW	0x0

Table 18-91. CTRL_CORE_HWOBS_CONTROL

Address Offset	0x0000 0360		
Physical Address	0x4A00 2360	**Instance**	CTRL_MODULE_CORE
Description	HW observability control. This register enables or disables HW observability outputs (to save power primarily)		
Type	RW		

31–19	18–14	13–9	8	7–3	2	1	0
RESERVED	HWOBS_CLKDIV_SEL_2	HWOBS_CLKDIV_SEL_1	RESERVED	HWOBS_CLKDIV_SEL	HWOBS_ALL_ZERO_MODE	HWOBS_ALL_ONE_MODE	HWOBS_MACRO_ENABLE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved	R	0x0
18:14	HWOBS_CLKDIV_SEL_2	Clock divider selection on po_hwobs(2). 0x1 = output is not divided 0x2 = output is divided by 2 0x4 = output is divided by 4 0x8 = output is divided by 8 0x10 = output is divided by 16	RW	0x0

Bits	Field Name	Description	Type	Reset
13:9	HWOBS_CLKDIV_SEL_1	Clock divider selection on po_hwobs(1). 0x1 = output is not divided 0x2 = output is divided by 2 0x4 = output is divided by 4 0x8 = output is divided by 8 0x10 = output is divided by 16	RW	0x0
8	RESERVED	Reserved	R	0x0
7:3	HWOBS_CLKDIV_SEL	Clock divider selection on po_hwobs(0). 0x1 = output is not divided 0x2 = output is divided by 2 0x4 = output is divided by 4 0x8 = output is divided by 8 0x10 = output is divided by 16	RW	0x0
2	HWOBS_ALL_ZERO_MODE	Used to gate observable signals. When set all outputs are set to zero (can be used to check the path from HW observability to external pads). 0x0 = hw observability ports are not gated 0x1 = hw observability ports are all set to 0	RW	0x0
1	HWOBS_ALL_ONE_MODE	Used to gate observable signals. When set all outputs are set to one (can be used to check the path from HW observability to external pads). 0x0 = hw observability ports are not gated 0x1 = hw observability ports are all set to 1	RW	0x0
0	HWOBS_MACRO_ENABLE	Used to gate observable signals coming from macros using the 32:bit HWOBS bus definition. When deasserted all outputs of the HWOBS busdef are set to zero. 0x0 = hw observability ports from macros are gated and set to zero 0x1 = hw observability ports from macros are not gated	RW	0x0

Table 18-92. CTRL_CORE_PHY_POWER_USB

Address Offset	0x0000 0370		
Physical Address	0x4A00 2370	**Instance**	CTRL_MODULE_CORE
Description	phy_power_usb		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USB_PWRCTL_CLK_FREQ										USB_PWRCTL_CLK_CMD								RESERVED													

Bits	Field Name	Description	Type	Reset
31:22	USB_PWRCTL_CLK_FREQ	Frequency of SYSCLK1 in MHz (rounded). For example, for 20MHz, program 0x14.	RW	0x0

Bits	Field Name	Description	Type	Reset
21:14	USB_PWRCTL_CLK_CMD	Powers up/down the USB3_PHY_TX and USB3_PHY_RX modules. This bit field is also used for partially power down these TX and RX modules. Each bit has the following meaning: Bit[14] - 0x1: Powers-up the USB3_PHY_RX Bit[15] - 0x1: Powers-up the USB3_PHY_TX Bit[16] - A don't care bit. Not used. Bit[17] - A don't care bit. Not used. Bit[18] - 0x1: Disables the synchronized power-up of USB3_PHY_TX with USB3_PHY_RX. The TX power-up is independent of the RX power-up. Bit[19] - 0x1: Disables the automatic power-cycling of USB3_PHY_RX in P3 power state when PLL_CLK stops and starts. Bit[20] - 0x1: Partially powers-down the USB3_PHY_RX when it is in P3 power state. DCC, Phase interpolator, Equalizer are disabled. Bit[21] - A don't care bit. Not used.	RW	0x0
13:0	RESERVED	Reserved	R	0x0

Table 18-93. CTRL_CORE_PHY_POWER_SATA

Address Offset	0x0000 0374		
Physical Address	0x4A00 2374	**Instance**	CTRL_MODULE_CORE
Description	phy_power_sata		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SATA_PWRCTL_CLK_FREQ										SATA_PWRCTL_CLK_CMD								RESERVED													

Bits	Field Name	Description	Type	Reset
31:22	SATA_PWRCTL_CLK_FREQ	Frequency of SYSCLK1 in MHz (rounded). For example, for 20MHz, program 0x14.	RW	0x0
21:14	SATA_PWRCTL_CLK_CMD	Powers up/down the SATA_PHY_TX and SATA_PHY_RX modules. 0x0: Powers down SATA_PHY_TX and SATA_PHY_RX 0x1: Powers up SATA_PHY_RX 0x2: Powers up SATA_PHY_TX 0x3: Powers up SATA_PHY_TX and SATA_PHY_RX 0x4-0xFF: Reserved	RW	0x0
13:0	RESERVED	Reserved	R	0x0

Table 18-94. CTRL_CORE_BANDGAP_MASK_1

Address Offset	0x0000 0380		
Physical Address	0x4A00 2380	**Instance**	CTRL_MODULE_CORE
Description	bgap_mask		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SIDLEMODE	COUNTER_DELAY	RESERVED	FREEZE_CORE	FREEZE_GPU	FREEZE_MPU	CLEAR_CORE	CLEAR_GPU	CLEAR_MPU	RESERVED	MASK_HOT_CORE	MASK_COLD_CORE	MASK_HOT_GPU	MASK_COLD_GPU	MASK_HOT_MPU	MASK_COLD_MPU

Bits	Field Name	Description	Type	Reset
31:30	SIDLEMODE	sidlemode for bandgap 0x0 = No Idle 0x1 = Force Idle 0x2 = Smart Idle 0x3 = Reserved	RW	0x0
29:27	COUNTER_DELAY	Counter delay 0x0 = Imediat 0x1 = Delay of 1ms 0x2 = Delay of 10ms 0x3 = Delay of 100ms 0x4 = Delay of 250ms 0x5 = Delay of 500ms	RW	0x0
26:24	RESERVED		R	0x0
23	FREEZE_CORE	Freeze the FIFO CORE 0x0 = No operation 0x1 = Freeze the FIFO	RW	0x0
22	FREEZE_GPU	Freeze the FIFO GPU 0x0 = No operation 0x1 = Freeze the FIFO	RW	0x0
21	FREEZE_MPU	Freeze the FIFO MPU 0x0 = No operation 0x1 = Freeze the FIFO	RW	0x0
20	CLEAR_CORE	Reset the FIFO CORE 0x0 = No operation 0x1 = Reset the FIFO	RW	0x0
19	CLEAR_GPU	Reset the FIFO GPU 0x0 = No operation 0x1 = Reset the FIFO	RW	0x0
18	CLEAR_MPU	Reset the FIFO MPU 0x0 = No operation 0x1 = Reset the FIFO	RW	0x0
17:6	RESERVED		R	0x0
5	MASK_HOT_CORE	Mask for hot event CORE 0x0 = hot event is masked 0x1 = hot event is not masked	RW	0x0
4	MASK_COLD_CORE	Mask for cold event CORE 0x0 = cold event is masked 0x1 = cold event is not masked	RW	0x0
3	MASK_HOT_GPU	Mask for hot event GPU 0x0 = hot event is masked 0x1 = hot event is not masked	RW	0x0

Bits	Field Name	Description	Type	Reset
2	MASK_COLD_GPU	Mask for cold event GPU 0x0 = cold event is masked 0x1 = cold event is not masked	RW	0x0
1	MASK_HOT_MPU	Mask for hot event MPU 0x0 = hot event is masked 0x1 = hot event is not masked	RW	0x0
0	MASK_COLD_MPU	Mask for cold event MPU 0x0 = cold event is masked 0x1 = cold event is not masked	RW	0x0

Table 18-95. CTRL_CORE_BANDGAP_THRESHOLD_MPU

Address Offset	0x0000 0384		
Physical Address	0x4A00 2384	**Instance**	CTRL_MODULE_CORE
Description	BGAP THRESHOLD MPU		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	THOLD_HOT_MPU	RESERVED	THOLD_COLD_MPU

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	THOLD_HOT_MPU	Value for the high temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0
15:10	RESERVED		R	0x0
9:0	THOLD_COLD_MPU	Value for the low temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0

Table 18-96. CTRL_CORE_BANDGAP_THRESHOLD_GPU

Address Offset	0x0000 0388		
Physical Address	0x4A00 2388	**Instance**	CTRL_MODULE_CORE
Description	BGAP THRESHOLD MM		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	THOLD_HOT_GPU	RESERVED	THOLD_COLD_GPU

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	THOLD_HOT_GPU	Value for the high temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0
15:10	RESERVED		R	0x0
9:0	THOLD_COLD_GPU	Value for the low temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0

Table 18-97. CTRL_CORE_BANDGAP_THRESHOLD_CORE

Address Offset	0x0000 038C		
Physical Address	0x4A00 238C	**Instance**	CTRL_MODULE_CORE
Description	BGAP THRESHOLD CORE		

Table 18-97. CTRL_CORE_BANDGAP_THRESHOLD_CORE (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						THOLD_HOT_CORE										RESERVED						THOLD_COLD_CORE									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	THOLD_HOT_CORE	Value for the high temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0
15:10	RESERVED		R	0x0
9:0	THOLD_COLD_CORE	Value for the low temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0

Table 18-98. CTRL_CORE_BANDGAP_TSHUT_MPU

Address Offset 0x0000 0390

Physical Address 0x4A00 2390 **Instance** CTRL_MODULE_CORE

Description BGAP TSHUT THRESHOLD MPU

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSHUT_MUXCTRL_MPU	RESERVED					THOLD_HOT_MPU										RESERVED						TSHUT_COLD_MPU									

Bits	Field Name	Description	Type	Reset
31	TSHUT_MUXCTRL_MPU	Writing a ‘1’ to this field allows SW to override the TSHUT_HOT and TSHUT_COLD values that are set by default in efuse	RW	0x0
30:26	RESERVED		R	0x0
25:16	TSHUT_HOT_MPU	Controls the TSHUT_HOT reset threshold, which protects the device from thermal runaway and potential device damage. The register defaults to 123°C. **Software override of this register value is not recommended, and should only be done with extreme caution as damage to the device can occur above the default setting.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	TSHUT_COLD_MPU	Controls the TSHUT_COLD reset threshold, which is the limit where the TSHUT comparator releases the device from reset after cooling from a TSHUT condition. The register defaults to 105°C. **Software override of this register value is not recommended, and should only be done with extreme caution.**	RW	0x0

Table 18-99. CTRL_CORE_BANDGAP_TSHUT_GPU

Address Offset 0x0000 0394

Physical Address 0x4A00 2394 **Instance** CTRL_MODULE_CORE

Description BGAP TSHUT THRESHOLD GPU

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TSHUT_MUXCTRL_GPU	RESERVED	TSHUT_HOT_GPU	RESERVED	TSHUT_COLD_GPU

Bits	Field Name	Description	Type	Reset
31	TSHUT_MUXCTRL_GPU	Writing a '1' to this field allows SW to override the TSHUT_HOT and TSHUT_COLD values that are set by default in efuse.	RW	0x0
30:26	RESERVED		R	0x0
25:16	TSHUT_HOT_GPU	Controls the TSHUT_HOT reset threshold, which protects the device from thermal runaway and potential device damage. The register defaults to 123°C. **Software override of this register value is not recommended, and should only be done with extreme caution as damage to the device can occur above the default setting**	RW	0x0
15:10	RESERVED		R	0x0
9:0	TSHUT_COLD_GPU	Controls the TSHUT_COLD reset threshold, which is the limit where the TSHUT comparator releases the device from reset after cooling from a TSHUT condition. The register defaults to 105°C. **Software override of this register value is not recommended, and should only be done with extreme caution.**	RW	0x0

Table 18-100. CTRL_CORE_BANDGAP_TSHUT_CORE

Address Offset	0x0000 0398		
Physical Address	0x4A00 2398	**Instance**	CTRL_MODULE_CORE
Description	BGAP TSHUT THRESHOLD CORE		
Type	R		

31	30 29 28 27 26	25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10	9 8 7 6 5 4 3 2 1 0
TSHUT_MUXCTRL_CORE	RESERVED	TSHUT_HOT_CORE	RESERVED	TSHUT_COLD_CORE

Bits	Field Name	Description	Type	Reset
31	TSHUT_MUXCTRL_CORE	Writing a '1' to this field allows SW to override the TSHUT_HOT and TSHUT_COLD values that are set by default in efuse.	RW	0x0
30:26	RESERVED		R	0x0
25:16	TSHUT_HOT_CORE	Controls the TSHUT_HOT reset threshold, which protects the device from thermal runaway and potential device damage. The register defaults to 123°C. **Software override of this register value is not recommended, and should only be done with extreme caution as damage to the device can occur above the default setting.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	TSHUT_COLD_CORE	Controls the TSHUT_COLD reset threshold, which is the limit where the TSHUT comparator releases the device from reset after cooling from a TSHUT condition. The register defaults to 105°C. **Software override of this register value is not recommended, and should only be done with extreme caution.**	RW	0x0

Table 18-101. CTRL_CORE_BANDGAP_STATUS_1

Address Offset	0x0000 03A8		
Physical Address	0x4A00 23A8	**Instance**	CTRL_MODULE_CORE
Description	BGAP STATUS		
Type	R		

31	30-6	5	4	3	2	1	0
ALERT	RESERVED	HOT_CORE	COLD_CORE	HOT_GPU	COLD_GPU	HOT_MPU	COLD_MPU

Bits	Field Name	Description	Type	Reset
31	ALERT	Alert temperature when '1'	R	0x0
30:6	RESERVED		R	0x0
5	HOT_CORE	Event for hot temperature mpu bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
4	COLD_CORE	Event for cold temperature mpu bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
3	HOT_GPU	Event for hot temperature gpu bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
2	COLD_GPU	Event for cold temperature gpu bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
1	HOT_MPU	Event for hot temperature core bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
0	COLD_MPU	Event for cold temperature core bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0

Table 18-102. CTRL_CORE_SATA_EXT_MODE

Address Offset	0x0000 03AC		
Physical Address	0x4A00 23AC	**Instance**	CTRL_MODULE_CORE
Description	SATA EXTENDED MODE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SATA_EXTENDED_MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SATA_EXTENDED_MODE	sata extended mode 0x0 = no extended mode 0x1 = extended mode	RW	0x0

Table 18-103. CTRL_CORE_DTEMP_MPU_0

Address Offset	0x0000 03C0		
Physical Address	0x4A00 23C0	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE MPU DOMAIN. Most recent sample		
Type	R		

31-10	9-0
DTEMP_TAG_MPU_0	DTEMP_TEMPERATURE_MPU_0

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_MPU_0	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_MPU_0	temperature	R	0x0

Table 18-104. CTRL_CORE_DTEMP_MPU_1

Address Offset	0x0000 03C4		
Physical Address	0x4A00 23C4	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE MPU DOMAIN		
Type	R		

31-10	9-0
DTEMP_TAG_MPU_1	DTEMP_TEMPERATURE_MPU_1

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_MPU_1	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_MPU_1	temperature	R	0x0

Table 18-105. CTRL_CORE_DTEMP_MPU_2

Address Offset	0x0000 03C8		
Physical Address	0x4A00 23C8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE MPU DOMAIN		
Type	R		

31-10	9-0
DTEMP_TAG_MPU_2	DTEMP_TEMPERATURE_MPU_2

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_MPU_2	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_MPU_2	temperature	R	0x0

Table 18-106. CTRL_CORE_DTEMP_MPU_3

Address Offset	0x0000 03CC		
Physical Address	0x4A00 23CC	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE MPU DOMAIN		
Type	R		

31:10	9:0
DTEMP_TAG_MPU_3	DTEMP_TEMPERATURE_MPU_3

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_MPU_3	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_MPU_3	temperature	R	0x0

Table 18-107. CTRL_CORE_DTEMP_MPU_4

Address Offset	0x0000 03D0		
Physical Address	0x4A00 23D0	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE MPU DOMAIN. Oldest sample		
Type	R		

31:10	9:0
DTEMP_TAG_MPU_4	DTEMP_TEMPERATURE_MPU_4

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_MPU_4	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_MPU_4	temperature	R	0x0

Table 18-108. CTRL_CORE_DTEMP_GPU_0

Address Offset	0x0000 03D4		
Physical Address	0x4A00 23D4	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE GPU DOMAIN. Most recent sample.		
Type	R		

31:10	9:0
DTEMP_TAG_GPU_0	DTEMP_TEMPERATURE_GPU_0

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_GPU_0	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_GPU_0	temperature	R	0x0

Table 18-109. CTRL_CORE_DTEMP_GPU_1

Address Offset	0x0000 03D8		
Physical Address	0x4A00 23D8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE GPU DOMAIN.		

Table 18-109. CTRL_CORE_DTEMP_GPU_1 (continued)

Type R

31–10	9–0
DTEMP_TAG_GPU_1	DTEMP_TEMPERATURE_GPU_1

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_GPU_1	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_GPU_1	temperature	R	0x0

Table 18-110. CTRL_CORE_DTEMP_GPU_2

Address Offset 0x0000 03DC
Physical Address 0x4A00 23DC **Instance** CTRL_MODULE_CORE
Description TAGGED TEMPERATURE GPU DOMAIN.
Type R

31–10	9–0
DTEMP_TAG_GPU_2	DTEMP_TEMPERATURE_GPU_2

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_GPU_2	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_GPU_2	temperature	R	0x0

Table 18-111. CTRL_CORE_DTEMP_GPU_3

Address Offset 0x0000 03E0
Physical Address 0x4A00 23E0 **Instance** CTRL_MODULE_CORE
Description TAGGED TEMPERATURE GPU DOMAIN.
Type R

31–10	9–0
DTEMP_TAG_GPU_3	DTEMP_TEMPERATURE_GPU_3

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_GPU_3	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_GPU_3	temperature	R	0x0

Table 18-112. CTRL_CORE_DTEMP_GPU_4

Address Offset 0x0000 03E4
Physical Address 0x4A00 23E4 **Instance** CTRL_MODULE_CORE
Description TAGGED TEMPERATURE GPU DOMAIN. Oldest sample.
Type R

31–10	9–0
DTEMP_TAG_GPU_4	DTEMP_TEMPERATURE_GPU_4

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_GPU_4	tag. Indicate number of times in the bgap state machine.	R	0x0

Bits	Field Name	Description	Type	Reset
9:0	DTEMP_TEMPERATURE_GPU_4	temperature	R	0x0

Table 18-113. CTRL_CORE_DTEMP_CORE_0

Address Offset	0x0000 03E8		
Physical Address	0x4A00 23E8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN. Most recent sample.		
Type	R		

31:10	9:0
DTEMP_TAG_CORE_0	DTEMP_TEMPERATURE_CORE_0

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_0	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_0	temperature	R	0x0

Table 18-114. CTRL_CORE_DTEMP_CORE_1

Address Offset	0x0000 03EC		
Physical Address	0x4A00 23EC	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN		
Type	R		

31:10	9:0
DTEMP_TAG_CORE_1	DTEMP_TEMPERATURE_CORE_1

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_1	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_1	temperature	R	0x0

Table 18-115. CTRL_CORE_DTEMP_CORE_2

Address Offset	0x0000 03F0		
Physical Address	0x4A00 23F0	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN		
Type	R		

31:10	9:0
DTEMP_TAG_CORE_2	DTEMP_TEMPERATURE_CORE_2

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_2	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_2	temperature	R	0x0

Table 18-116. CTRL_CORE_DTEMP_CORE_3

Address Offset	0x0000 03F4		
Physical Address	0x4A00 23F4	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_CORE_3																						DTEMP_TEMPERATURE_CORE_3									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_3	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_3	temperature	R	0x0

Table 18-117. CTRL_CORE_DTEMP_CORE_4

Address Offset	0x0000 03F8		
Physical Address	0x4A00 23F8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN. Oldest sample.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_CORE_4																						DTEMP_TEMPERATURE_CORE_4									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_4	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_4	temperature	R	0x0

Table 18-118. CTRL_CORE_SMA_SW_0

Address Offset	0x0000 03FC		
Physical Address	0x4A00 23FC	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													SATA_PLL_SOFT_RESET	RESERVED															ISOLATE	EMIF2_CKE_GATING_CTRL	EMIF1_CKE_GATING_CTRL

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	SATA_PLL_SOFT_RESET	Software reset control for SATA PLL	RW	0x0
17:3	RESERVED		R	0x0
2	ISOLATE	This bit is used during the isolation/de-isolation sequence described in *Isolation Requirements*.	RW	0x0
1	EMIF2_CKE_GATING_CTRL	Forces the EMIF2 CKE pad to tri-state. 0x0: The CKE pad is not in tri-state and can be controlled by EMIF2 0x1: The CKE pad is in tri-state	RW	0x0

Bits	Field Name	Description	Type	Reset
0	EMIF1_CKE_GATING_CTRL	Forces the EMIF1 CKE pad to tri-state. 0x0: The CKE pad is not in tri-state and can be controlled by EMIF1 0x1: The CKE pad is in tri-state	RW	0x0

Table 18-119. CTRL_CORE_SEC_ERR_STATUS_FUNC_2

Address Offset	0x0000 0414		
Physical Address	0x4A00 2414	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status functional Register 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TC1_EDMA_FW_ERROR	RESERVED			QSPI_FW_ERROR	RESERVED				TPCC_EDMA_FW_ERROR	TC0_EDMA_FW_ERROR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		MCASP3_FW_ERROR	MCASP2_FW_ERROR	MCASP1_FW_ERROR	VCP2_FW_ERROR	VCP1_FW_ERROR	PCIESS2_FW_ERROR	PCIESS1_FW_ERROR	IPU2_FW_ERROR	L4_PERIPH3_FW_ERROR	L4_PERIPH2_FW_ERROR	L3RAM3_FW_ERROR	L3RAM2_FW_ERROR	DSP2_FW_ERROR	DSP1_FW_ERROR

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	TC1_EDMA_FW_ERROR	EDMA TC1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
25:23	RESERVED		R	0x0
22	QSPI_FW_ERROR	QSPI firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
21:18	RESERVED		R	
17	TPCC_EDMA_FW_ERROR	EDMA TPCC firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	TC0_EDMA_FW_ERROR	EDMA TC0 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15:14	RESERVED		R	
13	MCASP3_FW_ERROR	McASP3 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
12	MCASP2_FW_ERROR	McASP2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
11	MCASP1_FW_ERROR	McASP1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
10	VCP2_FW_ERROR	VCP2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
9	VCP1_FW_ERROR	VCP1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
8	PCIESS2_FW_ERROR	PCIeSS2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
7	PCIESS1_FW_ERROR	PCIeSS1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
6	IPU2_FW_ERROR	IPU2 firewall. 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
5	L4_PERIPH3_FW_ERROR	L4 periph3 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	L4_PERIPH2_FW_ERROR	L4 periph2 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
3	L3RAM3_FW_ERROR	L3RAM3 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
2	L3RAM2_FW_ERROR	L3RAM2 target firewall. 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
1	DSP2_FW_ERROR	DSP2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
0	DSP1_FW_ERROR	DSP1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Table 18-120. CTRL_CORE_SEC_ERR_STATUS_DEBUG_2

Address Offset	0x0000 041C		
Physical Address	0x4A00 241C	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status debug Register 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	TC1_EDMA_DBGFW_ERROR	RESERVED	QSPI_DBGFW_ERROR	RESERVED	TPCC_EDMA_DBGFW_ERROR	TC0_EDMA_DBGFW_ERROR	RESERVED	MCASP3_DBGFW_ERROR	MCASP2_DBGFW_ERROR	MCASP1_DBGFW_ERROR	VCP2_DBGFW_ERROR	VCP1_DBGFW_ERROR	PCIESS2_DBGFW_ERROR	PCIESS1_DBGFW_ERROR	IPU2_DBGFW_ERROR	L4_PERIPH3_DBGFW_ERROR	L4_PERIPH2_DBGFW_ERROR	L3RAM3_DBGFW_ERROR	L3RAM2_DBGFW_ERROR	DSP2_DBGFW_ERROR	DSP1_DBGFW_ERROR

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	TC1_EDMA_DBGFW_ERROR	EDMA TC1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
25:23	RESERVED		R	0x0
22	QSPI_DBGFW_ERROR	QSPI debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
21:18	RESERVED		R	0x0
17	TPCC_EDMA_DBGFW_ERROR	EDMA TPCC debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	TC0_EDMA_DBGFW_ERROR	EDMA TC0 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15:14	RESERVED		R	0x0
13	MCASP3_DBGFW_ERROR	McASP3 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
12	MCASP2_DBGFW_ERROR	McASP2 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
11	MCASP1_DBGFW_ERROR	McASP1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
10	VCP2_DBGFW_ERROR	VCP2 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
9	VCP1_DBGFW_ERROR	VCP1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
8	PCIESS2_DBGFW_ERROR	PCIeSS2 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
7	PCIESS1_DBGFW_ERROR	PCIeSS1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
6	IPU2_DBGFW_ERROR	IPU2 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
5	L4_PERIPH3_DBGFW_ERROR	L4 periph3 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	L4_PERIPH2_DBGFW_ERROR	L4 periph2 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
3	L3RAM3_DBGFW_ERROR	L3RAM3 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
2	L3RAM2_DBGFW_ERROR	L3RAM2 target debug firewall. 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
1	DSP2_DBGFW_ERROR	DSP2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
0	DSP1_DBGFW_ERROR	DSP1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Table 18-121. CTRL_CORE_EMIF_INITIATOR_PRIORITY_1

Address Offset	0x0000 0420		
Physical Address	0x4A00 2420	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="3">MPU_EMIF_PRIORITY</td><td colspan="9">RESERVED</td><td colspan="3">DSP1_MDMA_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">DSP1_CFG_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">DSP1_EDMA_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">DSP2_EDMA_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">DSP2_CFG_EMIF_PRIORITY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	MPU_EMIF_PRIORITY	MPU priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
27:19	RESERVED		R	0x88
18:16	DSP1_MDMA_EMIF_PRIORITY	DSP1 MDMA priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
15	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
14:12	DSP1_CFG_EMIF_PRIORITY	DSP1 CFG priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	DSP1_EDMA_EMIF_PRIORITY	DSP1 EDMA priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7	RESERVED		R	0x0
6:4	DSP2_EDMA_EMIF_PRIORITY	DSP2 EDMA priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
3	RESERVED		R	0x0
2:0	DSP2_CFG_EMIF_PRIORITY	DSP2 CFG priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4

Table 18-122. CTRL_CORE_EMIF_INITIATOR_PRIORITY_2

Address Offset	0x0000 0424		
Physical Address	0x4A00 2424	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	DSP2_MDMA_EMIF_PRIORITY			RESERVED	IVA_ICONT1_EMIF_PRIORITY			RESERVED					EVE1_TC0_EMIF_PRIORITY			RESERVED	EVE2_TC0_EMIF_PRIORITY			RESERVED											

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	DSP2_MDMA_EMIF_PRIORITY	DSP2 MDMA priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
27	RESERVED		R	0x0
26:24	IVA_ICONT1_EMIF_PRIORITY	IVA ICONT1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
23:19	RESERVED		R	0x0
18:16	EVE1_TC0_EMIF_PRIORITY	EVE1 TC0 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
15	RESERVED		R	0x0
14:12	EVE2_TC0_EMIF_PRIORITY	EVE2 TC0 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11:0	RESERVED		R	0x444

Table 18-123. CTRL_CORE_EMIF_INITIATOR_PRIORITY_3

Address Offset	0x0000 0428		
Physical Address	0x4A00 2428	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

31:19	18:16	15	14:12	11	10:8	7:3	2:0
RESERVED	IPU1_EMIF_PRIORITY	RESERVED	IPU2_EMIF_PRIORITY	RESERVED	DMA_SYSTEM_EMIF_PRIORITY	RESERVED	EDMA_TC0_EMIF_PRIORITY

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x888
18:16	IPU1_EMIF_PRIORITY	IPU1 priority setting 0x0 = highest prioroty 0x7 = lowest priority	RW	0x4
15	RESERVED		R	0x0
14:12	IPU2_EMIF_PRIORITY	IPU2 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	DMA_SYSTEM_EMIF_PRIORITY	DMA SYSTEM priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7:3	RESERVED		R	0x8
2:0	EDMA_TC0_EMIF_PRIORITY	EDMA TC0 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4

Table 18-124. CTRL_CORE_EMIF_INITIATOR_PRIORITY_4

Address Offset	0x0000 042C		
Physical Address	0x4A00 242C	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

31	30:28	27	26:24	23	22:20	19	18:16	15	14:12	11	10:8	7	6:4	3	2:0
RESERVED	EDMA_TC1_EMIF_PRIORITY	RESERVED	DSS_EMIF_PRIORITY	RESERVED	MLB_MMU1_EMIF_PRIORITY	RESERVED	PCIESS1_EMIF_PRIORITY	RESERVED	PCIESS2_EMIF_PRIORITY	RESERVED	VIP1_P1_P2_EMIF_PRIORITY	RESERVED	VIP2_P1_P2_EMIF_PRIORITY	RESERVED	VIP3_P1_P2_EMIF_PRIORITY

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	EDMA_TC1_EMIF_PRIORITY	EDMA TC1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
27	RESERVED		R	0x0
26:24	DSS_EMIF_PRIORITY	DSS priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
23	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
22:20	MLB_MMU1_EMIF_PRIORITY	MLB, MMU1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
19	RESERVED		R	0x0
18:16	PCIESS1_EMIF_PRIORITY	PCIeSS1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
15	RESERVED		R	0x0
14:12	PCIESS2_EMIF_PRIORITY	PCIeSS2 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	VIP1_P1_P2_EMIF_PRIORITY	VIP1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7	RESERVED		R	0x0
6:4	VIP2_P1_P2_EMIF_PRIORITY	VIP2 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
3	RESERVED		R	0x0
2:0	VIP3_P1_P2_EMIF_PRIORITY	VIP3 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4

Table 18-125. CTRL_CORE_EMIF_INITIATOR_PRIORITY_5

Address Offset	0x0000 0430		
Physical Address	0x4A00 2430	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="3">VPE_P1_P2_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">MMC1_GPU_P1_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">MMC2_GPU_P2_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">BB2D_P1_P2_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">GMAC_SW_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">USB1_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">USB2_EMIF_PRIORITY</td><td>RESERVED</td><td colspan="3">USB3_EMIF_PRIORITY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	VPE_P1_P2_EMIF_PRIORITY	VPE priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
27	RESERVED		R	0x0
26:24	MMC1_GPU_P1_EMIF_PRIORITY	MMC1, GPU P1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
23	RESERVED		R	0x0
22:20	MMC2_GPU_P2_EMIF_PRIORITY	MMC2, GPU P2 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4

Bits	Field Name	Description	Type	Reset
19	RESERVED		R	0x0
18:16	BB2D_P1_P2_EMIF_PRIORITY	BB2D priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
15	RESERVED		R	0x0
14:12	GMAC_SW_EMIF_PRIORITY	GMAC_SW priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	USB1_EMIF_PRIORITY	USB1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7	RESERVED		R	0x0
6:4	USB2_EMIF_PRIORITY	USB2 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
3	RESERVED		R	0x0
2:0	USB3_EMIF_PRIORITY	USB3 priority setting 0x0 = highest priority 0x7 = lowest priorty	RW	0x4

Table 18-126. CTRL_CORE_EMIF_INITIATOR_PRIORITY_6

Address Offset	0x0000 0434		
Physical Address	0x4A00 2434	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

31	30:28	27:15	14:12	11	10:8	7	6:4	3:0
RESERVED	USB4_EMIF_PRIORITY	RESERVED	SATA_EMIF_PRIORITY	RESERVED	EVE1_TC1_EMIF_PRIORITY	RESERVED	EVE2_TC1_EMIF_PRIORITY	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	USB4_EMIF_PRIORITY	USB4 priority setting 0x0 = highest priority 0x7 = lowest prority	RW	0x4
27:15	RESERVED		R	0x888
14:12	SATA_EMIF_PRIORITY	SATA priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	EVE1_TC1_EMIF_PRIORITY	EVE1 TC1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:4	EVE2_TC1_EMIF_PRIORITY	EVE2 TC1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
3:0	RESERVED		R	0x4

Table 18-127. CTRL_CORE_L3_INITIATOR_PRESSURE_1

Address Offset	0x0000 043C		
Physical Address	0x4A00 243C	**Instance**	CTRL_MODULE_CORE
Description	Register for pressure settings for L3 arbitration		
Type	RW		

31:28	27:26	25:19	18:17	16:11	10:9	8:0
RESERVED	MPU_L3_PRESSURE	RESERVED	DSP1_CFG_L3_PRESSURE	RESERVED	DSP2_CFG_L3_PRESSURE	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	MPU_L3_PRESSURE	MPU pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
25:19	RESERVED		R	0x0
18:17	DSP1_CFG_L3_PRESSURE	DSP1 CFG pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
16:11	RESERVED		R	0x0
10:9	DSP2_CFG_L3_PRESSURE	DSP2 CFG pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
8:0	RESERVED		R	0x0

Table 18-128. CTRL_CORE_L3_INITIATOR_PRESSURE_2

Address Offset	0x0000 0440		
Physical Address	0x4A00 2440	**Instance**	CTRL_MODULE_CORE
Description	Register for pressure settings for L3 arbitration		
Type	RW		

31:14	13:12	11	10:9	8:0
RESERVED	IPU1_L3_PRESSURE	RESERVED	IPU2_L3_PRESSURE	RESERVED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	IPU1_L3_PRESSURE	IPU1 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:9	IPU2_L3_PRESSURE	IPU2 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
8:0	RESERVED		R	0x0

Table 18-129. CTRL_CORE_L3_INITIATOR_PRESSURE_4

Address Offset	0x0000 0448		
Physical Address	0x4A00 2448	**Instance**	CTRL_MODULE_CORE
Description	Register for pressure settings for L3 arbitration		
Type	RW		

31:25	24:23	22	21:20	19:0
RESERVED	GPU_P1_L3_PRESSURE	RESERVED	GPU_P2_L3_PRESSURE	RESERVED

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x0
24:23	GPU_P1_L3_PRESSURE	GPU P1 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
22	RESERVED		R	0x0
21:20	GPU_P2_L3_PRESSURE	GPU P2 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
19:0	RESERVED		R	0x0

Table 18-130. CTRL_CORE_L3_INITIATOR_PRESSURE_5

Address Offset	0x0000 044C		
Physical Address	0x4A00 244C	**Instance**	CTRL_MODULE_CORE
Description	Register for pressure settings for L3 arbitration		
Type	RW		

31:5	4:3	2	1:0
RESERVED	SATA_L3_PRESSURE	RESERVED	MMC1_L3_PRESSURE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:3	SATA_L3_PRESSURE	SATA pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
2	RESERVED		R	0x0
1:0	MMC1_L3_PRESSURE	MMC1 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0

Table 18-131. CTRL_CORE_L3_INITIATOR_PRESSURE_6

Address Offset	0x0000 0450		
Physical Address	0x4A00 2450	**Instance**	CTRL_MODULE_CORE
Description	Register for pressure settings for L3 arbitration		
Type	RW		

31:19	18:17	16:15	14	13:12	11	10:9	8	7:6	5:0
RESERVED	MMC2_L3_PRESSURE	USB1_L3_PRESSURE	RESERVED	USB2_L3_PRESSURE	RESERVED	USB3_L3_PRESSURE	RESERVED	USB4_L3_PRESSURE	RESERVED

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:17	MMC2_L3_PRESSURE	MMC2 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
16:15	USB1_L3_PRESSURE	USB1 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
14	RESERVED		R	0x0
13:12	USB2_L3_PRESSURE	USB2 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
11	RESERVED		R	0x0
10:9	USB3_L3_PRESSURE	USB3 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
8	RESERVED		R	0x0
7:6	USB4_L3_PRESSURE	USB4 pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
5:0	RESERVED		R	0x0

Table 18-132. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_0

Address Offset	0x0000 0458		
Physical Address	0x4A00 2458	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_iva [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31:0
STD_FUSE_OPP_VDD_IVA_0

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_0		R	0x0

Table 18-133. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_1

Address Offset	0x0000 045C		
Physical Address	0x4A00 245C	**Instance**	CTRL_MODULE_CORE

Table 18-133. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_1 (continued)

Description	Standard Fuse OPP VDD_iva [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_1		R	0x0

Table 18-134. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_2

Address Offset	0x0000 0460		
Physical Address	0x4A00 2460	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_iva [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_2		R	0x0

Table 18-135. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_3

Address Offset	0x0000 0464		
Physical Address	0x4A00 2464	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_iva [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_3		R	0x0

Table 18-136. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_4

Address Offset	0x0000 0468		
Physical Address	0x4A00 2468	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_iva [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_4		R	0x0

Table 18-137. CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL

Address Offset	0x0000 046C		
Physical Address	0x4A00 246C	**Instance**	CTRL_MODULE_CORE
Description	DSPEVE Voltage Body Bias LDO Control register		
Type	RW		

31:11	10	9:5	4:0
RESERVED	LDOVBBDSPEVE_FBB_MUX_CTRL	LDOVBBDSPEVE_FBB_VSET_IN	LDOVBBDSPEVE_FBB_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LDOVBBDSPEVE_FBB_MUX_CTRL	Override control of EFUSE Forward Body Bias voltage value 0x0 = efuse value is used 0x1 = override value is used	RW	0x0
9:5	LDOVBBDSPEVE_FBB_VSET_IN	EFUSE Forward Body Bias voltage value	R	0x0
4:0	LDOVBBDSPEVE_FBB_VSET_OUT	Override value for Forward Body Bias voltage. If ABB is used, depending on the OPP this bit field should be loaded with a value read from one of the CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_x[24:20] VSETABB bit fields. This value applies if LDOVBBDSPEVE_FBB_MUX_CTRL is set to 0x1.	RW	0x0

Table 18-138. CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL

Address Offset	0x0000 0470		
Physical Address	0x4A00 2470	**Instance**	CTRL_MODULE_CORE
Description	IVA Voltage Body Bias LDO Control register		
Type	RW		

31:11	10	9:5	4:0
RESERVED	LDOVBBIVA_FBB_MUX_CTRL	LDOVBBIVA_FBB_VSET_IN	LDOVBBIVA_FBB_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LDOVBBIVA_FBB_MUX_CTRL	Override control of EFUSE Forward Body Bias voltage value 0x0 = efuse value is used 0x1 = override value is used	RW	0x0
9:5	LDOVBBIVA_FBB_VSET_IN	EFUSE Forward Body Bias voltage value	R	0x0
4:0	LDOVBBIVA_FBB_VSET_OUT	Override value for Forward Body Bias voltage. If ABB is used, depending on the OPP this bit field should be loaded with a value read from one of the CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_x[24:20] VSETABB bit fields. This value applies if LDOVBBIVA_FBB_MUX_CTRL is set to 0x1.	RW	0x0

Table 18-139. CTRL_CORE_CUST_FUSE_UID_0

Address Offset	0x0000 04E8		
Physical Address	0x4A00 24E8	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [031:000] (16 bits upper Redundant field) [FIELD OVERFLOW]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_0																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_0		R	0x0

Table 18-140. CTRL_CORE_CUST_FUSE_UID_1

Address Offset	0x0000 04EC		
Physical Address	0x4A00 24EC	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [063:032] (16 bits upper Redundant field) [FIELD F]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_1																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_1		R	0x0

Table 18-141. CTRL_CORE_CUST_FUSE_UID_2

Address Offset	0x0000 04F0		
Physical Address	0x4A00 24F0	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [095:064] (16 bits upper Redundant field) [FIELD E]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_2																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_2		R	0x0

Table 18-142. CTRL_CORE_CUST_FUSE_UID_3

Address Offset	0x0000 04F4		
Physical Address	0x4A00 24F4	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [127:096] (16 bits upper Redundant field) [FIELD D]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_3																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_3		R	0x0

Table 18-143. CTRL_CORE_CUST_FUSE_UID_4

Address Offset	0x0000 04F8		
Physical Address	0x4A00 24F8	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [159:127] (16 bits upper Redundant field) [FIELD C]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_4																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_4		R	0x0

Table 18-144. CTRL_CORE_CUST_FUSE_UID_5

Address Offset	0x0000 04FC		
Physical Address	0x4A00 24FC	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [191:160] (16 bits upper Redundant field) [FIELD B]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_5																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_5		R	0x0

Table 18-145. CTRL_CORE_CUST_FUSE_UID_6

Address Offset	0x0000 0500		
Physical Address	0x4A00 2500	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [223:192] (16 bits upper Redundant field) [FIELD A]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CUST_FUSE_UID_6

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_6		R	0x0

Table 18-146. CTRL_CORE_CUST_FUSE_PCIE_ID_0

Address Offset	0x0000 0508		
Physical Address	0x4A00 2508	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. PCIe ID [031:000] (16 bits upper Redundant field) [FIELD OVERFLOW]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_PCIE_ID_0																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_PCIE_ID_0		R	0x0

Table 18-147. CTRL_CORE_CUST_FUSE_USB_ID_0

Address Offset	0x0000 0510		
Physical Address	0x4A00 2510	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. USB ID [031:000] (16 bits upper Redundant field) [FIELD OVERFLOW]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_USB_ID_0																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_USB_ID_0		R	0x0

Table 18-148. CTRL_CORE_MAC_ID_SW_0

Address Offset	0x0000 0514		
Physical Address	0x4A00 2514	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_1 [63:32].		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							STD_FUSE_MAC_ID_SW_0																								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	STD_FUSE_MAC_ID_SW_0	This bit field contains the last three octets (NIC specific) of the MAC address of the GMAC_SW port 0. Bits [23:16] contain the 4th octet of the MAC address. Bits [15:8] contain the 5th octet of the MAC address. Bits [7:0] contain the last (6th) octet of the MAC address.	R	0x0

Table 18-149. CTRL_CORE_MAC_ID_SW_1

Address Offset	0x0000 0518		
Physical Address	0x4A00 2518	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_1 [31:0].		

Table 18-149. CTRL_CORE_MAC_ID_SW_1 (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							STD_FUSE_MAC_ID_SW_1																								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	STD_FUSE_MAC_ID_SW_1	This bit field contains the first three octets (the OUI) of the MAC address of the GMAC_SW port 0. Bits [23:16] contain the first octet of the MAC address. Bits [15:8] contain the second octet of the MAC address. Bits [7:0] contain the third octet of the MAC address.	R	0x0

Table 18-150. CTRL_CORE_MAC_ID_SW_2

Address Offset	0x0000 051C		
Physical Address	0x4A00 251C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_2 [63:32].		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							STD_FUSE_MAC_ID_SW_2																								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	STD_FUSE_MAC_ID_SW_2	This bit field contains the last three octets (NIC specific) of the MAC address of the GMAC_SW port 1. Bits [23:16] contain the 4th octet of the MAC address. Bits [15:8] contain the 5th octet of the MAC address. Bits [7:0] contain the last (6th) octet of the MAC address.	R	0x0

Table 18-151. CTRL_CORE_MAC_ID_SW_3

Address Offset	0x0000 0520		
Physical Address	0x4A00 2520	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_2 [31:0].		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							STD_FUSE_MAC_ID_SW_3																								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	STD_FUSE_MAC_ID_SW_3	This bit field contains the first three octets (the OUI) of the MAC address of the GMAC_SW port 1. Bits [23:16] contain the first octet of the MAC address. Bits [15:8] contain the second octet of the MAC address. Bits [7:0] contain the third octet of the MAC address.	R	0x0

Table 18-152. CTRL_CORE_SMA_SW_1

Address Offset	0x0000 0534		
Physical Address	0x4A00 2534	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RGMII2_ID_MODE_N	RGMII1_ID_MODE_N	DSS_CH2_ON_OFF	DSS_CH1_ON_OFF	DSS_CH0_ON_OFF	DSS_CH2_IPC	DSS_CH1_IPC	DSS_CH0_IPC	DSS_CH2_RF	DSS_CH1_RF	DSS_CH0_RF	RESERVED	VIP3_CLK_INV_PORT_1A	VIP3_CLK_INV_PORT_2A	VPE_CLK_DIV_BY_2_EN	VIP2_CLK_INV_PORT_2B	VIP2_CLK_INV_PORT_1B	VIP2_CLK_INV_PORT_2A	VIP2_CLK_INV_PORT_1A	VIP1_CLK_INV_PORT_2B	VIP1_CLK_INV_PORT_1B	VIP1_CLK_INV_PORT_2A	VIP1_CLK_INV_PORT_1A

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	RGMII2_ID_MODE_N	Ethernet RGMII port 2 internal delay on transmit (SR2.0) 0x0: Internal delay enabled 0x1: Internal delay disabled	RW	0x0
25	RGMII1_ID_MODE_N	Ethernet RGMII port 1 internal delay on transmit (SR2.0) 0x0: Internal delay enabled 0x1: Internal delay disabled	RW	0x0
24	DSS_CH2_ON_OFF	DSS Channel 2 Pixel clock control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit DSS_CH2_RF	RW	0x0
23	DSS_CH1_ON_OFF	DSS Channel 1 Pixel clock control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit DSS_CH1_RF	RW	0x0
22	DSS_CH0_ON_OFF	DSS Channel 0 Pixel clock control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit DSS_CH0_RF	RW	0x0
21	DSS_CH2_IPC	DSS Channel 2 IPC control 0x0: Data is driven on the LCD data lines on the rising edge of the pixel clock 0x1: Data is driven on the LCD data lines on the falling edge of the pixel clock	RW	0x0
20	DSS_CH1_IPC	DSS Channel 1 IPC control 0x0: Data is driven on the LCD data lines on the rising edge of the pixel clock 0x1: Data is driven on the LCD data lines on the falling edge of the pixel clock	RW	0x0
19	DSS_CH0_IPC	DSS Channel 0 IPC control 0x0: Data is driven on the LCD data lines on the rising edge of the pixel clock 0x1: Data is driven on the LCD data lines on the falling edge of the pixel clock	RW	0x0
18	DSS_CH2_RF	DSS Channel 2 Rise/Fall control 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit DSS_CH2_ON_OFF set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit DSS_CH2_ON_OFF set to 1)	RW	0x0
17	DSS_CH1_RF	DSS Channel 1 Rise/Fall control 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit DSS_CH1_ON_OFF set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit DSS_CH1_ON_OFF set to 1)	RW	0x0

Bits	Field Name	Description	Type	Reset
16	DSS_CH0_RF	DSS Channel 0 Rise/Fall control 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit DSS_CH0_ON_OFF set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit DSS_CH0_ON_OFF set to 1)	RW	0x0
15:11	RESERVED		R	0x0
10	VIP3_CLK_INV_PORT_1A	VIP3 Slice 1 Clock inversion for Port A enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
9	VIP3_CLK_INV_PORT_2A	VIP3 Slice 0 Clock inversion for Port A enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
8	VPE_CLK_DIV_BY_2_EN	Selects alternative clock source for VPE. 0x0: Default clock source from DPLL_CORE is selected 0x1: Alternative clock source from DPLL_VIDEO1 is selected	RW	0x0
7	VIP2_CLK_INV_PORT_2B	VIP2 Slice 1 Clock inversion for Port B enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
6	VIP2_CLK_INV_PORT_1B	VIP2 Slice 0 Clock inversion for Port B enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
5	VIP2_CLK_INV_PORT_2A	VIP2 Slice 1 Clock inversion for Port A enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
4	VIP2_CLK_INV_PORT_1A	VIP2 Slice 0 Clock inversion for Port A enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
3	VIP1_CLK_INV_PORT_2B	VIP1 Slice 1 Clock inversion for Port B enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
2	VIP1_CLK_INV_PORT_1B	VIP1 Slice 0 Clock inversion for Port B enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
1	VIP1_CLK_INV_PORT_2A	VIP1 Slice 1 Clock inversion for Port A enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0
0	VIP1_CLK_INV_PORT_1A	VIP1 Slice 0 Clock inversion for Port A enable 0x0: clock inversion is disabled 0x1: clock inversion is enabled	RW	0x0

Table 18-153. CTRL_CORE_DSS_PLL_CONTROL

Address Offset	0x0000 0538		
Physical Address	0x4A00 2538	**Instance**	CTRL_MODULE_CORE
Description	DSS PLLs Mux control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SDVENC_CLK_SELECTION	DSI1_C_CLK1_SELECTION	DSI1_B_CLK1_SELECTION	DSI1_A_CLK1_SELECTION	PLL_HDMI_DSS_CONTROL_DISABLE	PLL_VIDEO2_DSS_CONTROL_DISABLE	PLL_VIDEO1_DSS_CONTROL_DISABLE

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	RW	0x0
10:9	SDVENC_CLK_SELECTION	SDVENC_CLK mux configuration 0x0: HDMI_CLK 0x1: DPLL_VIDEO1_HSDIVIDER_clkout3	RW	0x1
8:7	DSI1_C_CLK1_SELECTION	DSI1_C_CLK1 mux configuration 0x0: DPLL_VIDEO2 0x1: DPLL_VIDEO1 0x2: DPLL_HDMI	RW	0x1
6:5	DSI1_B_CLK1_SELECTION	DSI1_B_CLK1 mux configuration 0x0: DPLL_VIDEO1 0x1: DPLL_VIDEO2 0x2: DPLL_HDMI 0x3: DPLL_ABE	RW	0x1
4:3	DSI1_A_CLK1_SELECTION	DSI1_A_CLK1 mux configuration 0x0: DPLL_VIDEO1 0x1: DPLL_HDMI	RW	0x1
2	PLL_HDMI_DSS_CONTROL_DISABLE	HDMI PLL disable 0x0: PLL enabled 0x1: PLL disabled	RW	0x1
1	PLL_VIDEO2_DSS_CONTROL_DISABLE	VIDEO2 PLL disable 0x0: PLL enabled 0x1: PLL disabled	RW	0x1
0	PLL_VIDEO1_DSS_CONTROL_DISABLE	VIDEO1 PLL disable 0x0: PLL enabled 0x1: PLL disabled	RW	0x1

Table 18-154. CTRL_CORE_MMR_LOCK_1

Address Offset	0x0000 0540		
Physical Address	0x4A00 2540	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 0100 and ending at address offset 0x0000 079F		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MMR_LOCK_1

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_1	Lock value for region 0x0000 0100 to 0x0000 079F 0x1A1C8144 = lock value 0x2FF1AC2B = unlock value	RW	0x1A1C8144

Table 18-155. CTRL_CORE_MMR_LOCK_2

Address Offset	0x0000 0544		
Physical Address	0x4A00 2544	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 07A0 and ending at address offset 0x0000 0D9F		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_2																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_2	Lock value for region 0x0000 07A0 to 0x0000 0D9F 0xFDF45530 = lock value 0xF757FDC0 = unlock value	RW	0xFDF45530

Table 18-156. CTRL_CORE_MMR_LOCK_3

Address Offset	0x0000 0548		
Physical Address	0x4A00 2548	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 0DA0 and ending at address offset 0x0000 0FFF		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_3																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_3	Lock value for region 0x0000 0DA0 to 0x0000 0FFF 0x1AE6E320 = lock value 0xE2BC3A6D = unlock value	RW	0x1AE6E320

Table 18-157. CTRL_CORE_MMR_LOCK_4

Address Offset	0x0000 054C		
Physical Address	0x4A00 254C	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 1000 and ending at address offset 0x0000 13FF		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_4																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_4	Lock value for region 0x0000 1000 to 0x0000 13FF 0x2FFA927C = lock value 0x1EBF131D = unlock value	RW	0x2FFA927C

Table 18-158. CTRL_CORE_MMR_LOCK_5

Address Offset	0x0000 0550		
Physical Address	0x4A00 2550	**Instance**	CTRL_MODULE_CORE

Table 18-158. CTRL_CORE_MMR_LOCK_5 (continued)

Description	Register to lock memory region starting at address offset 0x0000 1400 and ending at address offset 0x0000 1FFF
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_5																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_5	Lock value for region 0x0000 1400 to 0x0000 1FFF 0x143F832C = lock value 0x6F361E05 = unlock value	RW	0x143F832C

Table 18-159. CTRL_CORE_CONTROL_IO_1

Address Offset	0x0000 0554		
Physical Address	0x4A00 2554	**Instance**	CTRL_MODULE_CORE
Description	Register to configure some IP level signals		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											MMU2_DISABLE	RESERVED			MMU1_DISABLE	RESERVED		TC1_DEFAULT_BURST_SIZE		RESERVED		TC0_DEFAULT_BURST_SIZE		RESERVED		GMII2_SEL		RESERVED		GMII1_SEL	

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20	MMU2_DISABLE	MMU2 DISABLE setting	RW	0x0
19:17	RESERVED		R	0x0
16	MMU1_DISABLE	MMU1 DISABLE setting	RW	0x0
15:14	RESERVED		R	0x0
13:12	TC1_DEFAULT_BURST_SIZE	EDMA TC1 Default Burst Size (DBS) setting 0x0: 16 byte burst 0x1: 32 byte burst 0x2: 64 byte burst 0x3: 128 byte burst	RW	0x3
11:10	RESERVED		R	0x0
9:8	TC0_DEFAULT_BURST_SIZE	EDMA TC0 Default Burst Size (DBS) setting 0x0: 16 byte burst 0x1: 32 byte burst 0x2: 64 byte burst 0x3: 128 byte burst	RW	0x3
7:6	RESERVED		R	0x0
5:4	GMII2_SEL	GMII2 selection setting 0x0: GMII/MII 0x1: RMII 0x2: RGMII 0x3: Reserved	RW	0x0
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	GMII1_SEL	GMII1 selection setting 0x0: GMII/MII 0x1: RMII 0x2: RGMII 0x3: Reserved	RW	0x0

Table 18-160. CTRL_CORE_CONTROL_IO_2

Address Offset	0x0000 0558		
Physical Address	0x4A00 2558	**Instance**	CTRL_MODULE_CORE
Description	Register to configure some IP level signals		
Type	RW		

31-24	23	22	21	20	19-14	13	12-11	10-8	7-6	5	4	3	2	1	0
RESERVED	GMAC_RESET_ISOLATION_ENABLE	PWMSS3_TBCLKEN	PWMSS2_TBCLKEN	PWMSS1_TBCLKEN	RESERVED	PCIE_1LANE_2LANE_SELECTION	RESERVED	QSPI_MEMMAPPED_CS	RESERVED	DCAN2_RAMINIT_START	DSS_DESHDCP_DISABLE	DCAN1_RAMINIT_START	DCAN2_RAMINIT_DONE	DCAN1_RAMINIT_DONE	DSS_DESHDCP_CLKEN

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	GMAC_RESET_ISOLATION_ENABLE	Reset isolation enable setting 0x0 = Reset is not isolated 0x1 = Reset is isolated	RW	0x0
22	PWMSS3_TBCLKEN	PWMSS3 CLOCK ENABLE setting	RW	0x0
21	PWMSS2_TBCLKEN	PWMSS2 CLOCK ENABLE setting	RW	0x0
20	PWMSS1_TBCLKEN	PWMSS1 CLOCK ENABLE setting	RW	0x0
19:14	RESERVED		R	0x0
13	PCIE_1LANE_2LANE_SELECTION	Reserved	RW	0x0
12:11	RESERVED		R	0x0
10:8	QSPI_MEMMAPPED_CS	QSPI CS MAPPING setting. 0x0: The QSPI configuration registers are accessed 0x1: An external device connected to CS0 is accessed 0x2: An external device connected to CS1 is accessed 0x3: An external device connected to CS2 is accessed 0x4-0x7: An external device connected to CS3 is accessed	RW	0x0
7:6	RESERVED		R	0x0
5	DCAN2_RAMINIT_START	DCAN2 RAM INIT START setting To initialize DCAN2 RAM, the bit should be set to 0x1. It is not auto cleared by hardware. **Note**: If DCAN RAMINIT sequence needs to be redone, this bit should be first cleared and then set again.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DSS_DESHDCP_DISABLE	DSS DESHDCP DISABLE setting	RW	0x0
3	DCAN1_RAMINIT_START	DCAN1 RAM INIT START setting To initialize DCAN1 RAM, the bit should be set to 0x1. It is not auto cleared by hardware. **Note**: If DCAN RAMINIT sequence needs to be redone, this bit should be first cleared and then set again.	RW	0x0
2	DCAN2_RAMINIT_DONE	DCAN2 RAM INIT DONE status	RW	0x0
1	DCAN1_RAMINIT_DONE	DCAN1 RAM INIT DONE status	RW	0x0
0	DSS_DESHDCP_CLKEN	DSS DESHDCP CLOCK ENABLE setting	RW	0x0

Table 18-161. CTRL_CORE_CONTROL_DSP1_RST_VECT

Address Offset	0x0000 055C		
Physical Address	0x4A00 255C	**Instance**	CTRL_MODULE_CORE
Description	Register for storing DSP1 reset vector		
Type	RW		

31:27	26:24	23:22	21:0
RESERVED	DSP1_NUM_MM	RESERVED	DSP1_RST_VECT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	RW	0x0
26:24	DSP1_NUM_MM	Number of DSP instances in the SoC 0x1 = 1 0x2 = 2	RW	0x0
23:22	RESERVED		R	0x0
21:0	DSP1_RST_VECT	DSP1 reset vector address	RW	0x0

Table 18-162. CTRL_CORE_CONTROL_DSP2_RST_VECT

Address Offset	0x0000 0560		
Physical Address	0x4A00 2560	**Instance**	CTRL_MODULE_CORE
Description	Register for storing DSP2 reset vector		
Type	RW		

31:27	26:24	23:22	21:0
RESERVED	DSP2_NUM_MM	RESERVED	DSP2_RST_VECT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	RW	0x0
26:24	DSP2_NUM_MM	Number of DSP instances in the SoC 0x1 = 1 0x2 = 2	RW	0x0
23:22	RESERVED		R	0x0
21:0	DSP2_RST_VECT	DSP2 reset vector address	RW	0x0

Table 18-163. CTRL_CORE_STD_FUSE_OPP_BGAP_DSPEVE

Address Offset	0x0000 0564		
Physical Address	0x4A00 2564	**Instance**	CTRL_MODULE_CORE

Table 18-163. CTRL_CORE_STD_FUSE_OPP_BGAP_DSPEVE (continued)

Description Trim values for DSPEVE associated bandgap. Contains TI Internal information, not intended for application use.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STD_FUSE_OPP_BGAP_DSPEVE_0								STD_FUSE_OPP_BGAP_DSPEVE_1							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x-
15:8	STD_FUSE_OPP_BGAP_DSPEVE_0	Trim values for DSPEVE associated bandgap. Contains TI Internal information, not intended for application use.	R	0x-
7:0	STD_FUSE_OPP_BGAP_DSPEVE_1	Trim values for DSPEVE associated bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Table 18-164. CTRL_CORE_STD_FUSE_OPP_BGAP_IVA

Address Offset 0x0000 0568

Physical Address 0x4A00 2568 **Instance** CTRL_MODULE_CORE

Description Trim values for IVA associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_BGAP_IVA_0								STD_FUSE_OPP_BGAP_IVA_1								STD_FUSE_OPP_BGAP_IVA_2								STD_FUSE_OPP_BGAP_IVA_3							

Bits	Field Name	Description	Type	Reset
31:24	STD_FUSE_OPP_BGAP_IVA_0	Trim values for IVA associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
23:16	STD_FUSE_OPP_BGAP_IVA_1	Trim values for IVA associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
15:8	STD_FUSE_OPP_BGAP_IVA_2	Trim values for IVA associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
7:0	STD_FUSE_OPP_BGAP_IVA_3	Trim values for IVA associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Table 18-165. CTRL_CORE_LDOSRAM_DSPEVE_VOLTAGE_CTRL

Address Offset 0x0000 056C

Physical Address 0x4A00 256C **Instance** CTRL_MODULE_CORE

Description DSPEVE SRAM LDO Control register

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LDOSRAMDSPEVE_RETMODE_MUX_CTRL	LDOSRAMDSPEVE_RETMODE_VSET_IN	LDOSRAMDSPEVE_RETMODE_VSET_OUT	RESERVED	LDOSRAMDSPEVE_ACTMODE_MUX_CTRL	LDOSRAMDSPEVE_ACTMODE_VSET_IN	LDOSRAMDSPEVE_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMDSPEVE_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMDSPEVE_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMDSPEVE_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMDSPEVE_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMDSPEVE_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMDSPEVE_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-166. CTRL_CORE_LDOSRAM_IVA_VOLTAGE_CTRL

Address Offset	0x0000 0570		
Physical Address	0x4A00 2570	**Instance**	CTRL_MODULE_CORE
Description	IVA SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LDOSRAMIVA_RETMODE_MUX_CTRL	LDOSRAMIVA_RETMODE_VSET_IN	LDOSRAMIVA_RETMODE_VSET_OUT	RESERVED	LDOSRAMIVA_ACTMODE_MUX_CTRL	LDOSRAMIVA_ACTMODE_VSET_IN	LDOSRAMIVA_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMIVA_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMIVA_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMIVA_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMIVA_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMIVA_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMIVA_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-167. CTRL_CORE_TEMP_SENSOR_DSPEVE

Address Offset	0x0000 0574		
Physical Address	0x4A00 2574	**Instance**	CTRL_MODULE_CORE
Description	Control VBGAPTS temperature sensor and thermal comparator shutdown register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				BGAP_TMPSOFF_DSPEVE	BGAP_EOCZ_DSPEVE	BGAP_DTEMP_DSPEVE									

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11	BGAP_TMPSOFF_DSPEVE	This bit indicates the temperature sensor state. 0x0: temperature sensor is ON 0x1: temperature sensor is OFF **NOTE:** Software doesn't take care of this bit to get the temperature data. Only the BGAP_EOCZ_DSPEVE bit is needed.	R	0x1
10	BGAP_EOCZ_DSPEVE	ADC End of Conversion. Active low, when BGAP_DTEMP_DSPEVE is valid.	R	0x0
9:0	BGAP_DTEMP_DSPEVE	Temperature data from the ADC. Valid if EOCZ is low.	R	0x0

Table 18-168. CTRL_CORE_TEMP_SENSOR_IVA

Address Offset	0x0000 0578		
Physical Address	0x4A00 2578	**Instance**	CTRL_MODULE_CORE
Description	Control VBGAPTS temperature sensor and thermal comparator shutdown register		
Type	R		

31:12	11	10	9:0
RESERVED	BGAP_TMPSOFF_IVA	BGAP_EOCZ_IVA	BGAP_DTEMP_IVA

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	BGAP_TMPSOFF_IVA	This bit indicates the temperature sensor state. 0x0: temperature sensor is ON 0x1: temperature sensor is OFF **NOTE:** Software doesn't take care of this bit to get the temperature data. Only the BGAP_EOCZ_IVA bit is needed.	R	0x1
10	BGAP_EOCZ_IVA	ADC End of Conversion. Active low, when BGAP_DTEMP_IVA is valid.	R	0x0
9:0	BGAP_DTEMP_IVA	Temperature data from the ADC. Valid if EOCZ is low.	R	0x0

Table 18-169. CTRL_CORE_BANDGAP_MASK_2

Address Offset	0x0000 057C		
Physical Address	0x4A00 257C	**Instance**	CTRL_MODULE_CORE
Description	bgap_mask		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FREEZE_IVA	FREEZE_DSPEVE	RESERVED	CLEAR_IVA	CLEAR_DSPEVE	RESERVED	MASK_HOT_IVA	MASK_COLD_IVA	MASK_HOT_DSPEVE	MASK_COLD_DSPEVE

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	FREEZE_IVA	Freeze the FIFO IVA 0x0 = No operation 0x1 = Freeze the FIFO	RW	0x0
21	FREEZE_DSPEVE	Freeze the FIFO DSPEVE 0x0 = No operation 0x1 = Freeze the FIFO	RW	0x0
20	RESERVED		R	0x0
19	CLEAR_IVA	Reset the FIFO IVA 0x0 = No operation 0x1 = Reset the FIFO	RW	0x0
18	CLEAR_DSPEVE	Reset the FIFO DSPEVE 0x0 = No operation 0x1 = Reset the FIFO	RW	0x0
17:4	RESERVED		R	0x0
3	MASK_HOT_IVA	Mask for hot event IVA 0x0 = hot event is masked 0x1 = hot event is not masked	RW	0x0
2	MASK_COLD_IVA	Mask for cold event IVA 0x0 = cold event is masked 0x1 = cold event is not masked	RW	0x0
1	MASK_HOT_DSPEVE	Mask for hot event DSPEVE 0x0 = hot event is masked 0x1 = hot event is not masked	RW	0x0
0	MASK_COLD_DSPEVE	Mask for cold event DSPEVE 0x0 = cold event is masked 0x1 = cold event is not masked	RW	0x0

Table 18-170. CTRL_CORE_BANDGAP_THRESHOLD_DSPEVE

Address Offset	0x0000 0580		
Physical Address	0x4A00 2580	**Instance**	CTRL_MODULE_CORE
Description	BGAP THRESHOLD DSPEVE		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	THOLD_HOT_DSPEVE	RESERVED	THOLD_COLD_DSPEVE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	THOLD_HOT_DSPEVE	Value for the high temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	THOLD_COLD_DSPEVE	Value for the low temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0

Table 18-171. CTRL_CORE_BANDGAP_THRESHOLD_IVA

Address Offset	0x0000 0584		
Physical Address	0x4A00 2584	**Instance**	CTRL_MODULE_CORE
Description	BGAP THRESHOLD IVA		
Type	RW		

31:26	25:16	15:10	9:0
RESERVED	THOLD_HOT_IVA	RESERVED	THOLD_COLD_IVA

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	THOLD_HOT_IVA	Value for the high temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0
15:10	RESERVED		R	0x0
9:0	THOLD_COLD_IVA	Value for the low temperature threshold. The values for loading this bit field are listed in Table 18-10.	RW	0x0

Table 18-172. CTRL_CORE_BANDGAP_TSHUT_DSPEVE

Address Offset	0x0000 0588		
Physical Address	0x4A00 2588	**Instance**	CTRL_MODULE_CORE
Description	BGAP TSHUT THRESHOLD IVA		
Type	R		

31	30:26	25:16	15:10	9:0
TSHUT_MUXCTRL_DSPEVE	RESERVED	TSHUT_HOT_DSPEVE	RESERVED	TSHUT_COLD_DSPEVE

Bits	Field Name	Description	Type	Reset
31	TSHUT_MUXCTRL_DSPEVE	Writing a ‘1’ to this field allows SW to override the TSHUT_HOT and TSHUT_COLD values that are set by default in efuse.	RW	0x0
30:26	RESERVED		R	0x0
25:16	TSHUT_HOT_DSPEVE	Controls the TSHUT_HOT reset threshold, which protects the device from thermal runaway and potential device damage. The register defaults to 123°C. **Software override of this register value is not recommended, and should only be done with extreme caution as damage to the device can occur above the default setting.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	TSHUT_COLD_DSPEVE	Controls the TSHUT_COLD reset threshold, which is the limit where the TSHUT comparator releases the device from reset after cooling from a TSHUT condition. The register defaults to 105°C. **Software override of this register value is not recommended, and should only be done with extreme caution.**	RW	0x0

Table 18-173. CTRL_CORE_BANDGAP_TSHUT_IVA

Address Offset	0x0000 058C		
Physical Address	0x4A00 258C	**Instance**	CTRL_MODULE_CORE
Description	BGAP TSHUT THRESHOLD IVA		
Type	R		

31	30:26	25:16	15:10	9:0
TSHUT_MUXCTRL_IVA	RESERVED	TSHUT_HOT_IVA	RESERVED	TSHUT_COLD_IVA

Bits	Field Name	Description	Type	Reset
31	TSHUT_MUXCTRL_IVA	Writing a '1' to this field allows SW to override the TSHUT_HOT and TSHUT_COLD values that are set by default in efuse.	RW	0x0
30:26	RESERVED		R	0x0
25:16	TSHUT_HOT_IVA	Controls the TSHUT_HOT reset threshold, which protects the device from thermal runaway and potential device damage. The register defaults to 123°C. **Software override of this register value is not recommended, and should only be done with extreme caution as damage to the device can occur above the default setting.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	TSHUT_COLD_IVA	Controls the TSHUT_COLD reset threshold, which is the limit where the TSHUT comparator releases the device from reset after cooling from a TSHUT condition. The register defaults to 105°C. **Software override of this register value is not recommended, and should only be done with extreme caution.**	RW	0x0

Table 18-174. CTRL_CORE_BANDGAP_STATUS_2

Address Offset	0x0000 0598		
Physical Address	0x4A00 2598	**Instance**	CTRL_MODULE_CORE
Description	BGAP STATUS		
Type	R		

31:4	3	2	1	0
RESERVED	HOT_IVA	COLD_IVA	HOT_DSPEVE	COLD_DSPEVE

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	HOT_IVA	Event for hot temperature iva bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
2	COLD_IVA	Event for cold temperature iva bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
1	HOT_DSPEVE	Event for hot temperature dspeve bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
0	COLD_DSPEVE	Event for cold temperature dspeve bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0

Table 18-175. CTRL_CORE_DTEMP_DSPEVE_0

Address Offset	0x0000 059C		
Physical Address	0x4A00 259C	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE DSPEVE DOMAIN. Most recent sample		
Type	R		

31:10	9:0
DTEMP_TAG_DSPEVE_0	DTEMP_TEMPERATURE_DSPEVE_0

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_DSPEVE_0	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_DSPEVE_0	temperature	R	0x0

Table 18-176. CTRL_CORE_DTEMP_DSPEVE_1

Address Offset	0x0000 05A0		
Physical Address	0x4A00 25A0	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE DSPEVE DOMAIN		
Type	R		

31:10	9:0
DTEMP_TAG_DSPEVE_1	DTEMP_TEMPERATURE_DSPEVE_1

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_DSPEVE_1	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_DSPEVE_1	temperature	R	0x0

Table 18-177. CTRL_CORE_DTEMP_DSPEVE_2

Address Offset	0x0000 05A4		
Physical Address	0x4A00 25A4	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE DSPEVE DOMAIN		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DTEMP_TAG_DSPEVE_2	DTEMP_TEMPERATURE_DSPEVE_2

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_DSPEVE_2	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_DSPEVE_2	temperature	R	0x0

Table 18-178. CTRL_CORE_DTEMP_DSPEVE_3

Address Offset	0x0000 05A8		
Physical Address	0x4A00 25A8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE DSPEVE DOMAIN		
Type	R		

31–10	9–0
DTEMP_TAG_DSPEVE_3	DTEMP_TEMPERATURE_DSPEVE_3

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_DSPEVE_3	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_DSPEVE_3	temperature	R	0x0

Table 18-179. CTRL_CORE_DTEMP_DSPEVE_4

Address Offset	0x0000 05AC		
Physical Address	0x4A00 25AC	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE DSPEVE DOMAIN. Oldest sample		
Type	R		

31–10	9–0
DTEMP_TAG_DSPEVE_4	DTEMP_TEMPERATURE_DSPEVE_4

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_DSPEVE_4	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_DSPEVE_4	temperature	R	0x0

Table 18-180. CTRL_CORE_DTEMP_IVA_0

Address Offset	0x0000 05B0		
Physical Address	0x4A00 25B0	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE IVA DOMAIN. Most recent sample		
Type	R		

31–10	9–0
DTEMP_TAG_IVA_0	DTEMP_TEMPERATURE_IVA_0

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_IVA_0	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_IVA_0	temperature	R	0x0

Table 18-181. CTRL_CORE_DTEMP_IVA_1

Address Offset	0x0000 05B4		
Physical Address	0x4A00 25B4	**Instance**	CTRL_MODULE_CORE

Table 18-181. CTRL_CORE_DTEMP_IVA_1 (continued)

Description	TAGGED TEMPERATURE IVA DOMAIN
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_IVA_1																						DTEMP_TEMPERATURE_IVA_1									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_IVA_1	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_IVA_1	temperature	R	0x0

Table 18-182. CTRL_CORE_DTEMP_IVA_2

Address Offset	0x0000 05B8		
Physical Address	0x4A00 25B8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE IVA DOMAIN		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_IVA_2																						DTEMP_TEMPERATURE_IVA_2									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_IVA_2	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_IVA_2	temperature	R	0x0

Table 18-183. CTRL_CORE_DTEMP_IVA_3

Address Offset	0x0000 05BC		
Physical Address	0x4A00 25BC	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE IVA DOMAIN		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_IVA_3																						DTEMP_TEMPERATURE_IVA_3									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_IVA_3	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_IVA_3	temperature	R	0x0

Table 18-184. CTRL_CORE_DTEMP_IVA_4

Address Offset	0x0000 05C0		
Physical Address	0x4A00 25C0	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE IVA DOMAIN. Oldest sample		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_IVA_4																						DTEMP_TEMPERATURE_IVA_4									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_IVA_4	tag. Indicate number of times in the bgap state machine.	R	0x0

Bits	Field Name	Description	Type	Reset
9:0	DTEMP_TEMPERATURE_IVA_4	temperature	R	0x0

Table 18-185. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_5

Address offset	0x0000 05C4		
Physical Address	0x0000 05C4	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_PLUS. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_IVA_5

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_PLUS which has to be written to the CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL [4:0] LDOVBBIVA_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_IVA_5	AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_PLUS. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-186. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_2

Address offset	0x0000 05CC		
Physical Address	0x4A00 25CC	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_NOM. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_IVA_2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_NOM which has to be written to the CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL [4:0] LDOVBBIVA_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_IVA_2	AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_NOM. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-187. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_3

Address offset	0x0000 05D0		
Physical Address	0x4A00 25D0	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_OD. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_IVA_3

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_OD which has to be written to the CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL [4:0] LDOVBBIVA_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_IVA_3	AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_OD. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-188. CTRL_CORE_STD_FUSE_OPP_VMIN_IVA_4

Address offset	0x0000 05D4		
Physical Address	0x4A00 25D4	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_HIGH. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_IVA_4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_HIGH which has to be written to the CTRL_CORE_LDOVBB_IVA_VOLTAGE_CTRL [4:0] LDOVBBIVA_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_IVA_4	AVS Class 0 voltage value for the vdd_iva voltage rail when running at OPP_HIGH. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-189. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5

Address offset	0x0000 05D8		
Physical Address	0x0000 05D8	**Instance**	CTRL_MODULE_CORE

Table 18-189. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5 (continued)

Description	This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_PLUS. This register also stores information about ABB configuration for that OPP.
Type	R

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_DSPEVE_5

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_PLUS which has to be written to the CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL [4:0] LDOVBBDSPEVE_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_5	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_PLUS. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-190. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2

Address offset	0x0000 05E0		
Physical Address	0x4A00 25E0	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_NOM. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_DSPEVE_2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_NOM which has to be written to the CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL [4:0] LDOVBBDSPEVE_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_2	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_NOM. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-191. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3

Address offset	0x0000 05E4		
Physical Address	0x4A00 25E4	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_OD. This register also stores information about ABB configuration for that OPP.		

Table 18-191. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3 (continued)

Type R

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_DSPEVE_3

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_OD which has to be written to the CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL [4:0] LDOVBBDSPEVE_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_3	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_OD. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-192. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4

Address offset 0x0000 05E8
Physical Address 0x4A00 25E8 **Instance** CTRL_MODULE_CORE
Description This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_HIGH. This register also stores information about ABB configuration for that OPP.
Type R

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_DSPEVE_4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_HIGH which has to be written to the CTRL_CORE_LDOVBB_DSPEVE_VOLTAGE_CTRL [4:0] LDOVBBDSPEVE_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_4	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_HIGH. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-193. CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2

Address Offset 0x0000 05F4
Physical Address 0x4A00 25F4 **Instance** CTRL_MODULE_CORE
Description This register contains the AVS Class 0 voltage value for the vdd voltage rail when running at OPP_NOM.
Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STD_FUSE_OPP_VMIN_CORE_2

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_CORE_2	AVS Class 0 voltage value for the vdd voltage rail when running at OPP_NOM. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-194. CTRL_CORE_LDOSRAM_CORE_2_VOLTAGE_CTRL

Address Offset	0x0000 0680		
Physical Address	0x4A00 2680	**Instance**	CTRL_MODULE_CORE
Description	CORE 2nd SRAM LDO Control register		
Type	RW		

31 30 29 28 27	26	25 24 23 22 21	20 19 18 17 16	15 14 13 12 11	10	9 8 7 6 5	4 3 2 1 0
RESERVED	LDOSRAMCORE_2_RETMODE_MUX_CTRL	LDOSRAMCORE_2_RETMODE_VSET_IN	LDOSRAMCORE_2_RETMODE_VSET_OUT	RESERVED	LDOSRAMCORE_2_ACTMODE_MUX_CTRL	LDOSRAMCORE_2_ACTMODE_VSET_IN	LDOSRAMCORE_2_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_2_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_2_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_2_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_2_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_2_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_2_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-195. CTRL_CORE_LDOSRAM_CORE_3_VOLTAGE_CTRL

Address Offset	0x0000 0684		
Physical Address	0x4A00 2684	**Instance**	CTRL_MODULE_CORE

Table 18-195. CTRL_CORE_LDOSRAM_CORE_3_VOLTAGE_CTRL (continued)

Description	CORE 3rd SRAM LDO Control register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDOSRAMCORE_3_RETMODE_MUX_CTRL	LDOSRAMCORE_3_RETMODE_VSET_IN					LDOSRAMCORE_3_RETMODE_VSET_OUT					RESERVED					LDOSRAMCORE_3_ACTMODE_MUX_CTRL	LDOSRAMCORE_3_ACTMODE_VSET_IN					LDOSRAMCORE_3_ACTMODE_VSET_OUT				

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_3_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_3_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_3_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_3_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_3_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_3_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-196. CTRL_CORE_NMI_DESTINATION_1

Address Offset	0x0000 068C		
Physical Address	0x4A00 268C	**Instance**	CTRL_MODULE_CORE
Description	Register for routing NMI interrupt to respective cores		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU2_C1								IPU2_C0								IPU1_C1							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
23:16	IPU2_C1	Enable IPU2 CORE1 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
15:8	IPU2_C0	Enable IPU2 CORE0 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
7:0	IPU1_C1	Enable IPU1 CORE1 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0

Table 18-197. CTRL_CORE_NMI_DESTINATION_2

Address Offset	0x0000 0690		
Physical Address	0x4A00 2690	**Instance**	CTRL_MODULE_CORE
Description	Register for routing NMI interrupt to respective cores		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPU1_C0								DSP2								DSP1								MPU							

Bits	Field Name	Description	Type	Reset
31:24	IPU1_C0	Enable IPU1 CORE0 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
23:16	DSP2	Enable DSP2 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
15:8	DSP1	Enable DSP1 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
7:0	MPU	Comes from Efuse (MPU_EN) 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0

Table 18-198. CTRL_CORE_IP_PRESSURE

Address Offset	0x0000 0698		
Physical Address	0x4A00 2698	**Instance**	CTRL_MODULE_CORE
Description	Register to override the L3 pressure setting for the MLB module		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													MLB_L3_PRESSURE_ENABLE	MLB_L3_PRESSURE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	MLB_L3_PRESSURE_ENABLE	Override enable for the MLB L3 pressure setting 0x0 = Overriding of the L3 pressure setting for the MLB module is disabled 0x1 = Overriding of the L3 pressure setting for the MLB module is enabled	RW	0x0
1:0	MLB_L3_PRESSURE	MLB L3 pressure setting 0x0 = Lowest 0x3 = Highest	RW	0x0

Table 18-199. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_0

Address Offset	0x0000 06A0		
Physical Address	0x4A00 26A0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_0		R	0x0

Table 18-200. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_1

Address Offset	0x0000 06A4		
Physical Address	0x4A00 26A4	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_1		R	0x0

Table 18-201. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_2

Address Offset	0x0000 06A8		
Physical Address	0x4A00 26A8	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_2		R	0x0

Table 18-202. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_3

Address Offset	0x0000 06AC		
Physical Address	0x4A00 26AC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_3		R	0x0

Table 18-203. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_4

Address Offset	0x0000 06B0		
Physical Address	0x4A00 26B0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_4		R	0x0

Table 18-204. CTRL_CORE_CUST_FUSE_SWRV_7

Address Offset	0x0000 06B4		
Physical Address	0x4A00 26B4	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. SWRV [31:0] (16 bits upper Redundant field) [FIELD A]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_7																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_7		R	0x0

Table 18-205. CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0

Address Offset	0x0000 06B8		
Physical Address	0x4A00 26B8	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse Calibration override value [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0		R	0x0

Table 18-206. CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1

Address Offset	0x0000 06BC		
Physical Address	0x4A00 26BC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse Calibration override value [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31–0
STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1		R	0x0

Table 18-207. CTRL_CORE_PCIE_POWER_STATE

Address Offset	0x0000 06C0		
Physical Address	0x4A00 26C0	**Instance**	CTRL_MODULE_CORE
Description	Register to PCIe related controls		
Type	RW		

31	30	29–26	25–16	15–0
BYPASS_EN_APLL_PCIE	CLKOOUTEN_APLL_PCIE	RESERVED	EFUSE_TRIM_ACS_PCIE	EFUSE_TRIM_PCIE_PLL

Bits	Field Name	Description	Type	Reset
31	BYPASS_EN_APLL_PCIE	Bypass enable bit setting for APLL_PCIe	RW	0x0
30	CLKOOUTEN_APLL_PCIE	Clock output enable bit setting for APLL_PCIe	RW	0x0
29:26	RESERVED		R	0x0
25:16	EFUSE_TRIM_ACS_PCIE	MMR override capability for ACS_PCIe efuse trim bits	RW	0x0
15:0	EFUSE_TRIM_PCIE_PLL	MMR override capability for PCIe PLL efuse trim bits	RW	0x0

Table 18-208. CTRL_CORE_BOOTSTRAP

Address Offset	0x0000 06C4		
Physical Address	0x4A00 26C4	**Instance**	CTRL_MODULE_CORE
Description	Register to view all the sysboot settings		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DSP_CLOCK_DIVIDER	RESERVED	BOOTDEVICESIZE	MUXCS0DEVICE	BOOTWAITEN	SPEEDSELECT	SYSBOOT_76	BOOTMODE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DSP_CLOCK_DIVIDER	SR1.1 Only: Divide factor for DSP clock 0x0: DSP_CLK2 is selected. Not supported on this SoC. 0x1: DSP_CLK3 is selected SR2.0 Only: Permanently disables the internal PU/PD resistors on pads gpmc_a[27:24, 22:19]. 0x0: Internal pull-down resistors are enabled 0x1: Internal pull-down resistors are permanently disabled	R	0x0
14	RESERVED	For proper device operation, a value of 0 is required on the sysboot14 pad.	R	0x0
13	BOOTDEVICESIZE	Select the size of the flash device on CS0. 0x0: 8-bit 0x1: 16-bit	R	0x0
12:11	MUXCS0DEVICE	Select IC boot sequence to be executed from a multiplexed address and data device attached to CS0. 0x0: Non-muxed device attached 0x1: Addr-Data Mux device attached 0x2: Reserved 0x3: Reserved	R	0x0
10	BOOTWAITEN	Enable the monitoring on CS0 of the wait pin at IC reset release time for read accesses. 0x0: Wait pin is not monitored for read accesses 0x1: Wait pin is monitored for read accesses	R	0x0
9:8	SPEEDSELECT	Indicates the SYS_CLK1 frequency (from osc0). Note that the internal FUNC_32K_CLK is equal to SYS_CLK1/610, which is nominally 32.7869 kHz with 20 MHz clock. 0x0: Reserved 0x1: 20 MHz 0x2: 27 MHz 0x3: 19.2 MHz	R	0x0
7:6	SYSBOOT_76	Sector offset for the location of the redundant SBL images in QSPI. 0x0: 64 KB offset 0x1: 128 KB offset 0x2: 256 KB offset 0x3: 512 KB offset	R	0x0
5:0	BOOTMODE	SYSBOOT mode	R	0x0

Table 18-209. CTRL_CORE_MLB_SIG_IO_CTRL

Address Offset	0x0000 06C8		
Physical Address	0x4A00 26C8	**Instance**	CTRL_MODULE_CORE

Table 18-209. CTRL_CORE_MLB_SIG_IO_CTRL (continued)

Description	Register to set the MLB's SIG IO characteristics
Type	RW

31:22	21:16	15:14	13:8	7	6	5	4	3	2:0
RESERVED	SIG_NC_IN	RESERVED	SIG_PC_IN	RESERVED	SIG_REMOVE_SKEW	SIG_PWRDNRX	SIG_PWRDNTX	SIG_EN_EXT_RES	RESERVED

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:16	SIG_NC_IN	efuse trim for Nmos impedance	RW	0x0
15:14	RESERVED		R	0x0
13:8	SIG_PC_IN	efuse trim for Pmos impedance	RW	0x0
7	RESERVED		R	0x0
6	SIG_REMOVE_SKEW	Adjust for skew generated by the receiver due to asymmetric inputs. 0x0: skew compensation is disabled 0x1: skew compensation is enabled	RW	0x0
5	SIG_PWRDNRX	powerdown receiver, active high 0x0 = Powered ON 0x1 = Powered OFF	RW	0x1
4	SIG_PWRDNTX	powerdown transmitter, active high 0x0 = Powered ON 0x1 = Powered OFF	RW	0x1
3	SIG_EN_EXT_RES	disables internal resistors 0x0 = Disabled 0x1 = Enabled	RW	0x0
2:0	RESERVED		R	0x0

Table 18-210. CTRL_CORE_MLB_DAT_IO_CTRL

Address Offset	0x0000 06CC		
Physical Address	0x4A00 26CC	**Instance**	CTRL_MODULE_CORE
Description	Register to set the MLB's DAT IO characteristics		
Type	RW		

31:22	21:16	15:14	13:8	7	6	5	4	3	2:0
RESERVED	DAT_NC_IN	RESERVED	DAT_PC_IN	RESERVED	DAT_REMOVE_SKEW	DAT_PWRDNRX	DAT_PWRDNTX	DAT_EN_EXT_RES	RESERVED

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21:16	DAT_NC_IN	efuse trim for Nmos impedance	RW	0x0
15:14	RESERVED		R	0x0
13:8	DAT_PC_IN	efuse trim for Pmos impedance	RW	0x0
7	RESERVED		R	0x0
6	DAT_REMOVE_SKEW	Adjust for skew generated by the receiver due to asymmetric inputs. 0x0: skew compensation is disabled 0x1: skew compensation is enabled	RW	0x0
5	DAT_PWRDNRX	powerdown receiver, active high 0x0 = Powered ON 0x1 = Powered OFF	RW	0x1
4	DAT_PWRDNTX	powerdown transmitter, active high 0x0 = Powered ON 0x1 = Powered OFF	RW	0x1
3	DAT_EN_EXT_RES	Enable/disable internal resistors 0x0 = Disabled 0x1 = Enabled	RW	0x0
2:0	RESERVED		R	0x0

Table 18-211. CTRL_CORE_MLB_CLK_BG_CTRL

Address Offset	0x0000 06D0		
Physical Address	0x4A00 26D0	**Instance**	CTRL_MODULE_CORE
Description	Register to set the MLB's clock receiver IO and bandgap characteristics		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															T_HYSTERISIS_EN	RESERVED								BG_TRIM						BG_PWRDN	CLK_PWRDN

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	T_HYSTERISIS_EN	Hysterisis enable 0x0: Disabled 0x1: Enabled	RW	0x0
15:8	RESERVED		R	0x0
7:2	BG_TRIM	Trim values for MLB bandgap	RW	0x0
1	BG_PWRDN	MLB bandgap cell enable. 0x0: The MLB bandgap cell is powered (enabled) 0x1: The MLB bandgap cell is disabled	RW	0x0
0	CLK_PWRDN	Enable the MLB differential clock receiver. 0x0: MLB differential clock receiver is enabled 0x1: MLB differential clock receiver is disabled	RW	0x1

Table 18-212. CTRL_CORE_EVE1_IRQ_0_1

Address Offset	0x0000 07A0		
Physical Address	0x4A00 27A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	EVE1_IRQ_1	RESERVED	EVE1_IRQ_0

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_1		RW	0x2
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_0		RW	0x1

Table 18-213. CTRL_CORE_EVE1_IRQ_2_3

Address Offset	0x0000 07A4		
Physical Address	0x4A00 27A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	EVE1_IRQ_3	RESERVED	EVE1_IRQ_2

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_3		RW	0x4
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_2		RW	0x3

Table 18-214. CTRL_CORE_EVE1_IRQ_4_5

Address Offset	0x0000 07A8		
Physical Address	0x4A00 27A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	EVE1_IRQ_5	RESERVED	EVE1_IRQ_4

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_5		RW	0x6
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_4		RW	0x5

Table 18-215. CTRL_CORE_EVE1_IRQ_6_7

Address Offset	0x0000 07AC		
Physical Address	0x4A00 27AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							EVE1_IRQ_7									RESERVED							EVE1_IRQ_6								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_7		RW	0x8
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_6		RW	0x7

Table 18-216. CTRL_CORE_EVE2_IRQ_0_1

Address Offset	0x0000 07B0		
Physical Address	0x4A00 27B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							EVE2_IRQ_1									RESERVED							EVE2_IRQ_0								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE2_IRQ_1		RW	0x2
15:9	RESERVED		R	0x0
8:0	EVE2_IRQ_0		RW	0x1

Table 18-217. CTRL_CORE_EVE2_IRQ_2_3

Address Offset	0x0000 07B4		
Physical Address	0x4A00 27B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							EVE2_IRQ_3									RESERVED							EVE2_IRQ_2								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE2_IRQ_3		RW	0x4
15:9	RESERVED		R	0x0
8:0	EVE2_IRQ_2		RW	0x3

Table 18-218. CTRL_CORE_EVE2_IRQ_4_5

Address Offset	0x0000 07B8		
Physical Address	0x4A00 27B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							EVE2_IRQ_5									RESERVED							EVE2_IRQ_4								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE2_IRQ_5		RW	0x6

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	EVE2_IRQ_4		RW	0x5

Table 18-219. CTRL_CORE_EVE2_IRQ_6_7

Address Offset	0x0000 07BC		
Physical Address	0x4A00 27BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	EVE2_IRQ_7	RESERVED	EVE2_IRQ_6

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE2_IRQ_7		RW	0x8
15:9	RESERVED		R	0x0
8:0	EVE2_IRQ_6		RW	0x7

Table 18-220. CTRL_CORE_IPU1_IRQ_23_24

Address Offset	0x0000 07E0		
Physical Address	0x4A00 27E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_24	RESERVED	IPU1_IRQ_23

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_24		RW	0x30
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_23		RW	0x14

Table 18-221. CTRL_CORE_IPU1_IRQ_25_26

Address Offset	0x0000 07E4		
Physical Address	0x4A00 27E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_26	RESERVED	IPU1_IRQ_25

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_26		RW	0x60
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_25		RW	0x0

Table 18-222. CTRL_CORE_IPU1_IRQ_27_28

Address Offset	0x0000 07E8

Table 18-222. CTRL_CORE_IPU1_IRQ_27_28 (continued)

Physical Address 0x4A00 27E8 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_28	RESERVED	IPU1_IRQ_27

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_28		RW	0x7F
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_27		RW	0x7E

Table 18-223. CTRL_CORE_IPU1_IRQ_29_30

Address Offset 0x0000 07EC

Physical Address 0x4A00 27EC **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_30	RESERVED	IPU1_IRQ_29

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_30		RW	0x81
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_29		RW	0x80

Table 18-224. CTRL_CORE_IPU1_IRQ_31_32

Address Offset 0x0000 07F0

Physical Address 0x4A00 27F0 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_32	RESERVED	IPU1_IRQ_31

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_32		RW	0x13
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_31		RW	0x82

Table 18-225. CTRL_CORE_IPU1_IRQ_33_34

Address Offset 0x0000 07F4

Physical Address 0x4A00 27F4 **Instance** CTRL_MODULE_CORE

Description

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IPU1_IRQ_34	RESERVED	IPU1_IRQ_33

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_34		RW	0x7
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_33		RW	0x83

Table 18-226. CTRL_CORE_IPU1_IRQ_35_36

Address Offset	0x0000 07F8		
Physical Address	0x4A00 27F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	IPU1_IRQ_36	RESERVED	IPU1_IRQ_35

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_36		RW	0x9
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_35		RW	0x8

Table 18-227. CTRL_CORE_IPU1_IRQ_37_38

Address Offset	0x0000 07FC		
Physical Address	0x4A00 27FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	IPU1_IRQ_38	RESERVED	IPU1_IRQ_37

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_38		RW	0x84
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_37		RW	0xA

Table 18-228. CTRL_CORE_IPU1_IRQ_39_40

Address Offset	0x0000 0800		
Physical Address	0x4A00 2800	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	IPU1_IRQ_40	RESERVED	IPU1_IRQ_39

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_40		RW	0x63
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:0	IPU1_IRQ_39		RW	0x62

Table 18-229. CTRL_CORE_IPU1_IRQ_41_42

Address Offset	0x0000 0804		
Physical Address	0x4A00 2804	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_42	RESERVED	IPU1_IRQ_41

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_42		RW	0x34
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_41		RW	0x33

Table 18-230. CTRL_CORE_IPU1_IRQ_43_44

Address Offset	0x0000 0808		
Physical Address	0x4A00 2808	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_44	RESERVED	IPU1_IRQ_43

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_44		RW	0x39
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_43		RW	0x38

Table 18-231. CTRL_CORE_IPU1_IRQ_45_46

Address Offset	0x0000 080C		
Physical Address	0x4A00 280C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_46	RESERVED	IPU1_IRQ_45

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_46		RW	0x5
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_45		RW	0x45

Table 18-232. CTRL_CORE_IPU1_IRQ_47_48

Address Offset	0x0000 0810		
Physical Address	0x4A00 2810	**Instance**	CTRL_MODULE_CORE

Table 18-232. CTRL_CORE_IPU1_IRQ_47_48 (continued)

Description	
Type	RW

31–25	24–16	15–9	8–0
RESERVED	IPU1_IRQ_48	RESERVED	IPU1_IRQ_47

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_48		RW	0xE
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_47		RW	0x85

Table 18-233. CTRL_CORE_IPU1_IRQ_49_50

Address Offset	0x0000 0814		
Physical Address	0x4A00 2814	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	IPU1_IRQ_50	RESERVED	IPU1_IRQ_49

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_50		RW	0x86
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_49		RW	0x42

Table 18-234. CTRL_CORE_IPU1_IRQ_51_52

Address Offset	0x0000 0818		
Physical Address	0x4A00 2818	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	IPU1_IRQ_52	RESERVED	IPU1_IRQ_51

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_52		RW	0x19
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_51		RW	0x18

Table 18-235. CTRL_CORE_IPU1_IRQ_53_54

Address Offset	0x0000 081C		
Physical Address	0x4A00 281C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	IPU1_IRQ_54	RESERVED	IPU1_IRQ_53

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_54		RW	0x23
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_53		RW	0x22

Table 18-236. CTRL_CORE_IPU1_IRQ_55_56

Address Offset	0x0000 0820		
Physical Address	0x4A00 2820	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_56	RESERVED	IPU1_IRQ_55

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_56		RW	0x2A
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_55		RW	0x28

Table 18-237. CTRL_CORE_IPU1_IRQ_57_58

Address Offset	0x0000 0824		
Physical Address	0x4A00 2824	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_58	RESERVED	IPU1_IRQ_57

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_58		RW	0x3D
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_57		RW	0x3C

Table 18-238. CTRL_CORE_IPU1_IRQ_59_60

Address Offset	0x0000 0828		
Physical Address	0x4A00 2828	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_60	RESERVED	IPU1_IRQ_59

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_60		RW	0x0
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_59		RW	0x32

Table 18-239. CTRL_CORE_IPU1_IRQ_61_62

Address Offset	0x0000 082C		
Physical Address	0x4A00 282C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_62	RESERVED	IPU1_IRQ_61

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_62		RW	0x16
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_61		RW	0x0

Table 18-240. CTRL_CORE_IPU1_IRQ_63_64

Address Offset	0x0000 0830		
Physical Address	0x4A00 2830	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_64	RESERVED	IPU1_IRQ_63

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_64		RW	0x6C
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_63		RW	0x53

Table 18-241. CTRL_CORE_IPU1_IRQ_65_66

Address Offset	0x0000 0834		
Physical Address	0x4A00 2834	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_66	RESERVED	IPU1_IRQ_65

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_66		RW	0x4E
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_65		RW	0x78

Table 18-242. CTRL_CORE_IPU1_IRQ_67_68

Address Offset	0x0000 0838		
Physical Address	0x4A00 2838	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU1_IRQ_68									RESERVED							IPU1_IRQ_67								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_68		RW	0x59
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_67		RW	0x51

Table 18-243. CTRL_CORE_IPU1_IRQ_69_70

Address Offset	0x0000 083C		
Physical Address	0x4A00 283C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU1_IRQ_70									RESERVED							IPU1_IRQ_69								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_70		RW	0x0
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_69		RW	0x0

Table 18-244. CTRL_CORE_IPU1_IRQ_71_72

Address Offset	0x0000 0840		
Physical Address	0x4A00 2840	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU1_IRQ_72									RESERVED							IPU1_IRQ_71								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_72		RW	0x76
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_71		RW	0x77

Table 18-245. CTRL_CORE_IPU1_IRQ_73_74

Address Offset	0x0000 0844		
Physical Address	0x4A00 2844	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU1_IRQ_74									RESERVED							IPU1_IRQ_73								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_74		RW	0x49

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_73		RW	0x48

Table 18-246. CTRL_CORE_IPU1_IRQ_75_76

Address Offset	0x0000 0848		
Physical Address	0x4A00 2848	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_76	RESERVED	IPU1_IRQ_75

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_76		RW	0x57
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_75		RW	0x75

Table 18-247. CTRL_CORE_IPU1_IRQ_77_78

Address Offset	0x0000 084C		
Physical Address	0x4A00 284C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU1_IRQ_78	RESERVED	IPU1_IRQ_77

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_78		RW	0x3E
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_77		RW	0x58

Table 18-248. CTRL_CORE_IPU1_IRQ_79_80

Address Offset	0x0000 0850		
Physical Address	0x4A00 2850	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:9	8:0
RESERVED	IPU1_IRQ_79

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	IPU1_IRQ_79		RW	0x3F

Table 18-249. CTRL_CORE_IPU2_IRQ_23_24

Address Offset	0x0000 0854		
Physical Address	0x4A00 2854	**Instance**	CTRL_MODULE_CORE
Description			

Table 18-249. CTRL_CORE_IPU2_IRQ_23_24 (continued)

Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_24	RESERVED	IPU2_IRQ_23

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_24		RW	0x30
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_23		RW	0x14

Table 18-250. CTRL_CORE_IPU2_IRQ_25_26

Address Offset 0x0000 0858
Physical Address 0x4A00 2858 **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_26	RESERVED	IPU2_IRQ_25

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_26		RW	0x60
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_25		RW	0x0

Table 18-251. CTRL_CORE_IPU2_IRQ_27_28

Address Offset 0x0000 085C
Physical Address 0x4A00 285C **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_28	RESERVED	IPU2_IRQ_27

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_28		RW	0x7F
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_27		RW	0x7E

Table 18-252. CTRL_CORE_IPU2_IRQ_29_30

Address Offset 0x0000 0860
Physical Address 0x4A00 2860 **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_30	RESERVED	IPU2_IRQ_29

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_30		RW	0x81
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_29		RW	0x80

Table 18-253. CTRL_CORE_IPU2_IRQ_31_32

Address Offset	0x0000 0864		
Physical Address	0x4A00 2864	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_32	RESERVED	IPU2_IRQ_31

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_32		RW	0x13
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_31		RW	0x82

Table 18-254. CTRL_CORE_IPU2_IRQ_33_34

Address Offset	0x0000 0868		
Physical Address	0x4A00 2868	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_34	RESERVED	IPU2_IRQ_33

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_34		RW	0x7
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_33		RW	0x83

Table 18-255. CTRL_CORE_IPU2_IRQ_35_36

Address Offset	0x0000 086C		
Physical Address	0x4A00 286C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_36	RESERVED	IPU2_IRQ_35

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_36		RW	0x9
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_35		RW	0x8

Table 18-256. CTRL_CORE_IPU2_IRQ_37_38

Address Offset	0x0000 0870		
Physical Address	0x4A00 2870	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_38	RESERVED	IPU2_IRQ_37

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_38		RW	0x84
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_37		RW	0xA

Table 18-257. CTRL_CORE_IPU2_IRQ_39_40

Address Offset	0x0000 0874		
Physical Address	0x4A00 2874	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_40	RESERVED	IPU2_IRQ_39

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_40		RW	0x63
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_39		RW	0x62

Table 18-258. CTRL_CORE_IPU2_IRQ_41_42

Address Offset	0x0000 0878		
Physical Address	0x4A00 2878	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_42	RESERVED	IPU2_IRQ_41

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_42		RW	0x34
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_41		RW	0x33

Table 18-259. CTRL_CORE_IPU2_IRQ_43_44

Address Offset	0x0000 087C		
Physical Address	0x4A00 287C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU2_IRQ_44									RESERVED							IPU2_IRQ_43								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_44		RW	0x39
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_43		RW	0x38

Table 18-260. CTRL_CORE_IPU2_IRQ_45_46

Address Offset	0x0000 0880		
Physical Address	0x4A00 2880	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU2_IRQ_46									RESERVED							IPU2_IRQ_45								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_46		RW	0x5
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_45		RW	0x45

Table 18-261. CTRL_CORE_IPU2_IRQ_47_48

Address Offset	0x0000 0884		
Physical Address	0x4A00 2884	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU2_IRQ_48									RESERVED							IPU2_IRQ_47								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_48		RW	0xE
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_47		RW	0x85

Table 18-262. CTRL_CORE_IPU2_IRQ_49_50

Address Offset	0x0000 0888		
Physical Address	0x4A00 2888	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU2_IRQ_50									RESERVED							IPU2_IRQ_49								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_50		RW	0x86

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_49		RW	0x42

Table 18-263. CTRL_CORE_IPU2_IRQ_51_52

Address Offset	0x0000 088C		
Physical Address	0x4A00 288C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU2_IRQ_52									RESERVED							IPU2_IRQ_51								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_52		RW	0x19
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_51		RW	0x18

Table 18-264. CTRL_CORE_IPU2_IRQ_53_54

Address Offset	0x0000 0890		
Physical Address	0x4A00 2890	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU2_IRQ_54									RESERVED							IPU2_IRQ_53								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_54		RW	0x23
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_53		RW	0x22

Table 18-265. CTRL_CORE_IPU2_IRQ_55_56

Address Offset	0x0000 0894		
Physical Address	0x4A00 2894	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IPU2_IRQ_56									RESERVED							IPU2_IRQ_55								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_56		RW	0x2A
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_55		RW	0x28

Table 18-266. CTRL_CORE_IPU2_IRQ_57_58

Address Offset	0x0000 0898

Table 18-266. CTRL_CORE_IPU2_IRQ_57_58 (continued)

Physical Address 0x4A00 2898 **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_58	RESERVED	IPU2_IRQ_57

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_58		RW	0x3D
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_57		RW	0x3C

Table 18-267. CTRL_CORE_IPU2_IRQ_59_60

Address Offset 0x0000 089C
Physical Address 0x4A00 289C **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_60	RESERVED	IPU2_IRQ_59

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_60		RW	0x0
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_59		RW	0x32

Table 18-268. CTRL_CORE_IPU2_IRQ_61_62

Address Offset 0x0000 08A0
Physical Address 0x4A00 28A0 **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_62	RESERVED	IPU2_IRQ_61

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_62		RW	0x16
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_61		RW	0x0

Table 18-269. CTRL_CORE_IPU2_IRQ_63_64

Address Offset 0x0000 08A4
Physical Address 0x4A00 28A4 **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IPU2_IRQ_64	RESERVED	IPU2_IRQ_63

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_64		RW	0x6C
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_63		RW	0x53

Table 18-270. CTRL_CORE_IPU2_IRQ_65_66

Address Offset	0x0000 08A8		
Physical Address	0x4A00 28A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	IPU2_IRQ_66	RESERVED	IPU2_IRQ_65

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_66		RW	0x4E
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_65		RW	0x78

Table 18-271. CTRL_CORE_IPU2_IRQ_67_68

Address Offset	0x0000 08AC		
Physical Address	0x4A00 28AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	IPU2_IRQ_68	RESERVED	IPU2_IRQ_67

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_68		RW	0x59
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_67		RW	0x51

Table 18-272. CTRL_CORE_IPU2_IRQ_69_70

Address Offset	0x0000 08B0		
Physical Address	0x4A00 28B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	IPU2_IRQ_70	RESERVED	IPU2_IRQ_69

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_70		RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:0	IPU2_IRQ_69		RW	0x0

Table 18-273. CTRL_CORE_IPU2_IRQ_71_72

Address Offset	0x0000 08B4		
Physical Address	0x4A00 28B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_72	RESERVED	IPU2_IRQ_71

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_72		RW	0x76
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_71		RW	0x77

Table 18-274. CTRL_CORE_IPU2_IRQ_73_74

Address Offset	0x0000 08B8		
Physical Address	0x4A00 28B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_74	RESERVED	IPU2_IRQ_73

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_74		RW	0x49
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_73		RW	0x48

Table 18-275. CTRL_CORE_IPU2_IRQ_75_76

Address Offset	0x0000 08BC		
Physical Address	0x4A00 28BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_76	RESERVED	IPU2_IRQ_75

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_76		RW	0x57
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_75		RW	0x75

Table 18-276. CTRL_CORE_IPU2_IRQ_77_78

Address Offset	0x0000 08C0		
Physical Address	0x4A00 28C0	**Instance**	CTRL_MODULE_CORE

Table 18-276. CTRL_CORE_IPU2_IRQ_77_78 (continued)

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	IPU2_IRQ_78	RESERVED	IPU2_IRQ_77

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU2_IRQ_78		RW	0x3E
15:9	RESERVED		R	0x0
8:0	IPU2_IRQ_77		RW	0x58

Table 18-277. CTRL_CORE_IPU2_IRQ_79_80

Address Offset 0x0000 08C4

Physical Address 0x4A00 28C4 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:9	8:0
RESERVED	IPU2_IRQ_79

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	IPU2_IRQ_79		RW	0x3F

Table 18-278. CTRL_CORE_DSP1_IRQ_32_33

Address Offset 0x0000 0948

Physical Address 0x4A00 2948 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_33	RESERVED	DSP1_IRQ_32

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_33		RW	0x2
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_32		RW	0x1

Table 18-279. CTRL_CORE_DSP1_IRQ_34_35

Address Offset 0x0000 094C

Physical Address 0x4A00 294C **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_35	RESERVED	DSP1_IRQ_34

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
24:16	DSP1_IRQ_35		RW	0x4
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_34		RW	0x3

Table 18-280. CTRL_CORE_DSP1_IRQ_36_37

Address Offset	0x0000 0950		
Physical Address	0x4A00 2950	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_37	RESERVED	DSP1_IRQ_36

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_37		RW	0x6
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_36		RW	0x5

Table 18-281. CTRL_CORE_DSP1_IRQ_38_39

Address Offset	0x0000 0954		
Physical Address	0x4A00 2954	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_39	RESERVED	DSP1_IRQ_38

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_39		RW	0x8
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_38		RW	0x7

Table 18-282. CTRL_CORE_DSP1_IRQ_40_41

Address Offset	0x0000 0958		
Physical Address	0x4A00 2958	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_41	RESERVED	DSP1_IRQ_40

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_41		RW	0xA
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_40		RW	0x9

Table 18-283. CTRL_CORE_DSP1_IRQ_42_43

Address Offset	0x0000 095C		
Physical Address	0x4A00 295C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_43	RESERVED	DSP1_IRQ_42

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_43		RW	0xC
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_42		RW	0xB

Table 18-284. CTRL_CORE_DSP1_IRQ_44_45

Address Offset	0x0000 0960		
Physical Address	0x4A00 2960	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_45	RESERVED	DSP1_IRQ_44

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_45		RW	0xE
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_44		RW	0xD

Table 18-285. CTRL_CORE_DSP1_IRQ_46_47

Address Offset	0x0000 0964		
Physical Address	0x4A00 2964	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_47	RESERVED	DSP1_IRQ_46

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_47		RW	0x10
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_46		RW	0xF

Table 18-286. CTRL_CORE_DSP1_IRQ_48_49

Address Offset	0x0000 0968		
Physical Address	0x4A00 2968	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP1_IRQ_49									RESERVED							DSP1_IRQ_48								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_49		RW	0x12
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_48		RW	0x11

Table 18-287. CTRL_CORE_DSP1_IRQ_50_51

Address Offset	0x0000 096C		
Physical Address	0x4A00 296C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP1_IRQ_51									RESERVED							DSP1_IRQ_50								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_51		RW	0x14
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_50		RW	0x13

Table 18-288. CTRL_CORE_DSP1_IRQ_52_53

Address Offset	0x0000 0970		
Physical Address	0x4A00 2970	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP1_IRQ_53									RESERVED							DSP1_IRQ_52								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_53		RW	0x16
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_52		RW	0x15

Table 18-289. CTRL_CORE_DSP1_IRQ_54_55

Address Offset	0x0000 0974		
Physical Address	0x4A00 2974	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP1_IRQ_55									RESERVED							DSP1_IRQ_54								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_55		RW	0x18

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_54		RW	0x17

Table 18-290. CTRL_CORE_DSP1_IRQ_56_57

Address Offset	0x0000 0978		
Physical Address	0x4A00 2978	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	DSP1_IRQ_57	RESERVED	DSP1_IRQ_56

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_57		RW	0x1A
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_56		RW	0x19

Table 18-291. CTRL_CORE_DSP1_IRQ_58_59

Address Offset	0x0000 097C		
Physical Address	0x4A00 297C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	DSP1_IRQ_59	RESERVED	DSP1_IRQ_58

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_59		RW	0x1C
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_58		RW	0x1B

Table 18-292. CTRL_CORE_DSP1_IRQ_60_61

Address Offset	0x0000 0980		
Physical Address	0x4A00 2980	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	DSP1_IRQ_61	RESERVED	DSP1_IRQ_60

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_61		RW	0x1E
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_60		RW	0x1D

Table 18-293. CTRL_CORE_DSP1_IRQ_62_63

Address Offset	0x0000 0984

Table 18-293. CTRL_CORE_DSP1_IRQ_62_63 (continued)

Physical Address	0x4A00 2984	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_63	RESERVED	DSP1_IRQ_62

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_63		RW	0x20
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_62		RW	0x1F

Table 18-294. CTRL_CORE_DSP1_IRQ_64_65

Address Offset	0x0000 0988		
Physical Address	0x4A00 2988	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_65	RESERVED	DSP1_IRQ_64

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_65		RW	0x22
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_64		RW	0x21

Table 18-295. CTRL_CORE_DSP1_IRQ_66_67

Address Offset	0x0000 098C		
Physical Address	0x4A00 298C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_67	RESERVED	DSP1_IRQ_66

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_67		RW	0x24
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_66		RW	0x23

Table 18-296. CTRL_CORE_DSP1_IRQ_68_69

Address Offset	0x0000 0990		
Physical Address	0x4A00 2990	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DSP1_IRQ_69	RESERVED	DSP1_IRQ_68

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_69		RW	0x26
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_68		RW	0x25

Table 18-297. CTRL_CORE_DSP1_IRQ_70_71

Address Offset	0x0000 0994		
Physical Address	0x4A00 2994	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP1_IRQ_71									RESERVED							DSP1_IRQ_70								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_71		RW	0x28
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_70		RW	0x27

Table 18-298. CTRL_CORE_DSP1_IRQ_72_73

Address Offset	0x0000 0998		
Physical Address	0x4A00 2998	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP1_IRQ_73									RESERVED							DSP1_IRQ_72								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_73		RW	0x2A
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_72		RW	0x29

Table 18-299. CTRL_CORE_DSP1_IRQ_74_75

Address Offset	0x0000 099C		
Physical Address	0x4A00 299C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP1_IRQ_75									RESERVED							DSP1_IRQ_74								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_75		RW	0x2C
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:0	DSP1_IRQ_74		RW	0x2B

Table 18-300. CTRL_CORE_DSP1_IRQ_76_77

Address Offset	0x0000 09A0		
Physical Address	0x4A00 29A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_77	RESERVED	DSP1_IRQ_76

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_77		RW	0x2E
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_76		RW	0x2D

Table 18-301. CTRL_CORE_DSP1_IRQ_78_79

Address Offset	0x0000 09A4		
Physical Address	0x4A00 29A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_79	RESERVED	DSP1_IRQ_78

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_79		RW	0x30
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_78		RW	0x2F

Table 18-302. CTRL_CORE_DSP1_IRQ_80_81

Address Offset	0x0000 09A8		
Physical Address	0x4A00 29A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_81	RESERVED	DSP1_IRQ_80

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_81		RW	0x32
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_80		RW	0x31

Table 18-303. CTRL_CORE_DSP1_IRQ_82_83

Address Offset	0x0000 09AC		
Physical Address	0x4A00 29AC	**Instance**	CTRL_MODULE_CORE

Table 18-303. CTRL_CORE_DSP1_IRQ_82_83 (continued)

Description

Type RW

31–25	24–16	15–9	8–0
RESERVED	DSP1_IRQ_83	RESERVED	DSP1_IRQ_82

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_83		RW	0x34
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_82		RW	0x33

Table 18-304. CTRL_CORE_DSP1_IRQ_84_85

Address Offset 0x0000 09B0

Physical Address 0x4A00 29B0 **Instance** CTRL_MODULE_CORE

Description

Type RW

31–25	24–16	15–9	8–0
RESERVED	DSP1_IRQ_85	RESERVED	DSP1_IRQ_84

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_85		RW	0x36
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_84		RW	0x35

Table 18-305. CTRL_CORE_DSP1_IRQ_86_87

Address Offset 0x0000 09B4

Physical Address 0x4A00 29B4 **Instance** CTRL_MODULE_CORE

Description

Type RW

31–25	24–16	15–9	8–0
RESERVED	DSP1_IRQ_87	RESERVED	DSP1_IRQ_86

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_87		RW	0x38
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_86		RW	0x37

Table 18-306. CTRL_CORE_DSP1_IRQ_88_89

Address Offset 0x0000 09B8

Physical Address 0x4A00 29B8 **Instance** CTRL_MODULE_CORE

Description

Type RW

31–25	24–16	15–9	8–0
RESERVED	DSP1_IRQ_89	RESERVED	DSP1_IRQ_88

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_89		RW	0x3A
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_88		RW	0x39

Table 18-307. CTRL_CORE_DSP1_IRQ_90_91

Address Offset	0x0000 09BC		
Physical Address	0x4A00 29BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_91	RESERVED	DSP1_IRQ_90

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_91		RW	0x3C
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_90		RW	0x3B

Table 18-308. CTRL_CORE_DSP1_IRQ_92_93

Address Offset	0x0000 09C0		
Physical Address	0x4A00 29C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_93	RESERVED	DSP1_IRQ_92

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_93		RW	0x3E
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_92		RW	0x3D

Table 18-309. CTRL_CORE_DSP1_IRQ_94_95

Address Offset	0x0000 09C4		
Physical Address	0x4A00 29C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP1_IRQ_95	RESERVED	DSP1_IRQ_94

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_95		RW	0x40
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_94		RW	0x3F

Table 18-310. CTRL_CORE_DSP2_IRQ_32_33

Address Offset	0x0000 09C8		
Physical Address	0x4A00 29C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_33	RESERVED	DSP2_IRQ_32

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_33		RW	0x2
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_32		RW	0x1

Table 18-311. CTRL_CORE_DSP2_IRQ_34_35

Address Offset	0x0000 09CC		
Physical Address	0x4A00 29CC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_35	RESERVED	DSP2_IRQ_34

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_35		RW	0x4
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_34		RW	0x3

Table 18-312. CTRL_CORE_DSP2_IRQ_36_37

Address Offset	0x0000 09D0		
Physical Address	0x4A00 29D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_37	RESERVED	DSP2_IRQ_36

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_37		RW	0x6
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_36		RW	0x5

Table 18-313. CTRL_CORE_DSP2_IRQ_38_39

Address Offset	0x0000 09D4		
Physical Address	0x4A00 29D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP2_IRQ_39									RESERVED							DSP2_IRQ_38								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_39		RW	0x8
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_38		RW	0x7

Table 18-314. CTRL_CORE_DSP2_IRQ_40_41

Address Offset 0x0000 09D8
Physical Address 0x4A00 29D8 **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP2_IRQ_41									RESERVED							DSP2_IRQ_40								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_41		RW	0xA
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_40		RW	0x9

Table 18-315. CTRL_CORE_DSP2_IRQ_42_43

Address Offset 0x0000 09DC
Physical Address 0x4A00 29DC **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP2_IRQ_43									RESERVED							DSP2_IRQ_42								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_43		RW	0xC
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_42		RW	0xB

Table 18-316. CTRL_CORE_DSP2_IRQ_44_45

Address Offset 0x0000 09E0
Physical Address 0x4A00 29E0 **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP2_IRQ_45									RESERVED							DSP2_IRQ_44								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_45		RW	0xE

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_44		RW	0xD

Table 18-317. CTRL_CORE_DSP2_IRQ_46_47

Address Offset	0x0000 09E4		
Physical Address	0x4A00 29E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_47	RESERVED	DSP2_IRQ_46

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_47		RW	0x10
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_46		RW	0xF

Table 18-318. CTRL_CORE_DSP2_IRQ_48_49

Address Offset	0x0000 09E8		
Physical Address	0x4A00 29E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_49	RESERVED	DSP2_IRQ_48

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_49		RW	0x12
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_48		RW	0x11

Table 18-319. CTRL_CORE_DSP2_IRQ_50_51

Address Offset	0x0000 09EC		
Physical Address	0x4A00 29EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_51	RESERVED	DSP2_IRQ_50

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_51		RW	0x14
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_50		RW	0x13

Table 18-320. CTRL_CORE_DSP2_IRQ_52_53

Address Offset	0x0000 09F0

Table 18-320. CTRL_CORE_DSP2_IRQ_52_53 (continued)

Physical Address	0x4A00 29F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_53	RESERVED	DSP2_IRQ_52

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_53		RW	0x16
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_52		RW	0x15

Table 18-321. CTRL_CORE_DSP2_IRQ_54_55

Address Offset	0x0000 09F4		
Physical Address	0x4A00 29F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_55	RESERVED	DSP2_IRQ_54

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_55		RW	0x18
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_54		RW	0x17

Table 18-322. CTRL_CORE_DSP2_IRQ_56_57

Address Offset	0x0000 09F8		
Physical Address	0x4A00 29F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_57	RESERVED	DSP2_IRQ_56

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_57		RW	0x1A
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_56		RW	0x19

Table 18-323. CTRL_CORE_DSP2_IRQ_58_59

Address Offset	0x0000 09FC		
Physical Address	0x4A00 29FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DSP2_IRQ_59	RESERVED	DSP2_IRQ_58

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_59		RW	0x1C
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_58		RW	0x1B

Table 18-324. CTRL_CORE_DSP2_IRQ_60_61

Address Offset 0x0000 0A00
Physical Address 0x4A00 2A00 **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_61	RESERVED	DSP2_IRQ_60

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_61		RW	0x1E
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_60		RW	0x1D

Table 18-325. CTRL_CORE_DSP2_IRQ_62_63

Address Offset 0x0000 0A04
Physical Address 0x4A00 2A04 **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_63	RESERVED	DSP2_IRQ_62

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_63		RW	0x20
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_62		RW	0x1F

Table 18-326. CTRL_CORE_DSP2_IRQ_64_65

Address Offset 0x0000 0A08
Physical Address 0x4A00 2A08 **Instance** CTRL_MODULE_CORE
Description
Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_65	RESERVED	DSP2_IRQ_64

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_65		RW	0x22
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:0	DSP2_IRQ_64		RW	0x21

Table 18-327. CTRL_CORE_DSP2_IRQ_66_67

Address Offset	0x0000 0A0C		
Physical Address	0x4A00 2A0C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	DSP2_IRQ_67	RESERVED	DSP2_IRQ_66

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_67		RW	0x24
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_66		RW	0x23

Table 18-328. CTRL_CORE_DSP2_IRQ_68_69

Address Offset	0x0000 0A10		
Physical Address	0x4A00 2A10	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	DSP2_IRQ_69	RESERVED	DSP2_IRQ_68

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_69		RW	0x26
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_68		RW	0x25

Table 18-329. CTRL_CORE_DSP2_IRQ_70_71

Address Offset	0x0000 0A14		
Physical Address	0x4A00 2A14	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	DSP2_IRQ_71	RESERVED	DSP2_IRQ_70

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_71		RW	0x28
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_70		RW	0x27

Table 18-330. CTRL_CORE_DSP2_IRQ_72_73

Address Offset	0x0000 0A18		
Physical Address	0x4A00 2A18	**Instance**	CTRL_MODULE_CORE

Table 18-330. CTRL_CORE_DSP2_IRQ_72_73 (continued)

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_73	RESERVED	DSP2_IRQ_72

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_73		RW	0x2A
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_72		RW	0x29

Table 18-331. CTRL_CORE_DSP2_IRQ_74_75

Address Offset 0x0000 0A1C

Physical Address 0x4A00 2A1C **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_75	RESERVED	DSP2_IRQ_74

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_75		RW	0x2C
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_74		RW	0x2B

Table 18-332. CTRL_CORE_DSP2_IRQ_76_77

Address Offset 0x0000 0A20

Physical Address 0x4A00 2A20 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_77	RESERVED	DSP2_IRQ_76

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_77		RW	0x2E
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_76		RW	0x2D

Table 18-333. CTRL_CORE_DSP2_IRQ_78_79

Address Offset 0x0000 0A24

Physical Address 0x4A00 2A24 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_79	RESERVED	DSP2_IRQ_78

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_79		RW	0x30
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_78		RW	0x2F

Table 18-334. CTRL_CORE_DSP2_IRQ_80_81

Address Offset	0x0000 0A28		
Physical Address	0x4A00 2A28	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_81	RESERVED	DSP2_IRQ_80

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_81		RW	0x32
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_80		RW	0x31

Table 18-335. CTRL_CORE_DSP2_IRQ_82_83

Address Offset	0x0000 0A2C		
Physical Address	0x4A00 2A2C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_83	RESERVED	DSP2_IRQ_82

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_83		RW	0x34
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_82		RW	0x33

Table 18-336. CTRL_CORE_DSP2_IRQ_84_85

Address Offset	0x0000 0A30		
Physical Address	0x4A00 2A30	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_85	RESERVED	DSP2_IRQ_84

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_85		RW	0x36
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_84		RW	0x35

Table 18-337. CTRL_CORE_DSP2_IRQ_86_87

Address Offset	0x0000 0A34		
Physical Address	0x4A00 2A34	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_87	RESERVED	DSP2_IRQ_86

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_87		RW	0x38
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_86		RW	0x37

Table 18-338. CTRL_CORE_DSP2_IRQ_88_89

Address Offset	0x0000 0A38		
Physical Address	0x4A00 2A38	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_89	RESERVED	DSP2_IRQ_88

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_89		RW	0x3A
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_88		RW	0x39

Table 18-339. CTRL_CORE_DSP2_IRQ_90_91

Address Offset	0x0000 0A3C		
Physical Address	0x4A00 2A3C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	DSP2_IRQ_91	RESERVED	DSP2_IRQ_90

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_91		RW	0x3C
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_90		RW	0x3B

Table 18-340. CTRL_CORE_DSP2_IRQ_92_93

Address Offset	0x0000 0A40		
Physical Address	0x4A00 2A40	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP2_IRQ_93									RESERVED							DSP2_IRQ_92								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_93		RW	0x3E
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_92		RW	0x3D

Table 18-341. CTRL_CORE_DSP2_IRQ_94_95

Address Offset	0x0000 0A44		
Physical Address	0x4A00 2A44	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSP2_IRQ_95									RESERVED							DSP2_IRQ_94								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_95		RW	0x40
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_94		RW	0x3F

Table 18-342. CTRL_CORE_MPU_IRQ_4_7

Address Offset	0x0000 0A48		
Physical Address	0x4A00 2A48	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_7									RESERVED							MPU_IRQ_4								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_7		RW	0x2
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_4		RW	0x1

Table 18-343. CTRL_CORE_MPU_IRQ_8_9

Address Offset	0x0000 0A4C		
Physical Address	0x4A00 2A4C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_9									RESERVED							MPU_IRQ_8								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_9		RW	0x4

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_8		RW	0x3

Table 18-344. CTRL_CORE_MPU_IRQ_10_11

Address Offset	0x0000 0A50		
Physical Address	0x4A00 2A50	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_11	RESERVED	MPU_IRQ_10

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_11		RW	0x6
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_10	**Note** NOTE: This bit field is not functional	RW	0x5

Table 18-345. CTRL_CORE_MPU_IRQ_12_13

Address Offset	0x0000 0A54		
Physical Address	0x4A00 2A54	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_13	RESERVED	MPU_IRQ_12

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_13		RW	0x8
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_12		RW	0x7

Table 18-346. CTRL_CORE_MPU_IRQ_14_15

Address Offset	0x0000 0A58		
Physical Address	0x4A00 2A58	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_15	RESERVED	MPU_IRQ_14

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_15		RW	0xA
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_14		RW	0x9

Table 18-347. CTRL_CORE_MPU_IRQ_16_17

Address Offset	0x0000 0A5C		
Physical Address	0x4A00 2A5C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9	8 7 6 5 4 3 2 1 0
RESERVED	MPU_IRQ_17	RESERVED	MPU_IRQ_16

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_17		RW	0xC
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_16		RW	0xB

Table 18-348. CTRL_CORE_MPU_IRQ_18_19

Address Offset	0x0000 0A60		
Physical Address	0x4A00 2A60	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9	8 7 6 5 4 3 2 1 0
RESERVED	MPU_IRQ_19	RESERVED	MPU_IRQ_18

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_19		RW	0xE
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_18		RW	0xD

Table 18-349. CTRL_CORE_MPU_IRQ_20_21

Address Offset	0x0000 0A64		
Physical Address	0x4A00 2A64	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9	8 7 6 5 4 3 2 1 0
RESERVED	MPU_IRQ_21	RESERVED	MPU_IRQ_20

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_21		RW	0x10
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_20		RW	0xF

Table 18-350. CTRL_CORE_MPU_IRQ_22_23

Address Offset	0x0000 0A68		
Physical Address	0x4A00 2A68	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_23									RESERVED							MPU_IRQ_22								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_23		RW	0x12
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_22		RW	0x11

Table 18-351. CTRL_CORE_MPU_IRQ_24_25

Address Offset	0x0000 0A6C		
Physical Address	0x4A00 2A6C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_25									RESERVED							MPU_IRQ_24								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_25		RW	0x14
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_24		RW	0x13

Table 18-352. CTRL_CORE_MPU_IRQ_26_27

Address Offset	0x0000 0A70		
Physical Address	0x4A00 2A70	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_27									RESERVED							MPU_IRQ_26								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_27		RW	0x16
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_26		RW	0x15

Table 18-353. CTRL_CORE_MPU_IRQ_28_29

Address Offset	0x0000 0A74		
Physical Address	0x4A00 2A74	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_29									RESERVED							MPU_IRQ_28								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_29		RW	0x18

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_28		RW	0x17

Table 18-354. CTRL_CORE_MPU_IRQ_30_31

Address Offset	0x0000 0A78		
Physical Address	0x4A00 2A78	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_31	RESERVED	MPU_IRQ_30

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_31		RW	0x1A
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_30		RW	0x19

Table 18-355. CTRL_CORE_MPU_IRQ_32_33

Address Offset	0x0000 0A7C		
Physical Address	0x4A00 2A7C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_33	RESERVED	MPU_IRQ_32

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_33		RW	0x1C
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_32		RW	0x1B

Table 18-356. CTRL_CORE_MPU_IRQ_34_35

Address Offset	0x0000 0A80		
Physical Address	0x4A00 2A80	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_35	RESERVED	MPU_IRQ_34

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_35		RW	0x1E
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_34		RW	0x1D

Table 18-357. CTRL_CORE_MPU_IRQ_36_37

Address Offset 0x0000 0A84

Table 18-357. CTRL_CORE_MPU_IRQ_36_37 (continued)

Physical Address 0x4A00 2A84 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_37	RESERVED	MPU_IRQ_36

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_37		RW	0x20
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_36		RW	0x1F

Table 18-358. CTRL_CORE_MPU_IRQ_38_39

Address Offset 0x0000 0A88

Physical Address 0x4A00 2A88 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_39	RESERVED	MPU_IRQ_38

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_39		RW	0x22
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_38		RW	0x21

Table 18-359. CTRL_CORE_MPU_IRQ_40_41

Address Offset 0x0000 0A8C

Physical Address 0x4A00 2A8C **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_41	RESERVED	MPU_IRQ_40

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_41		RW	0x24
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_40		RW	0x23

Table 18-360. CTRL_CORE_MPU_IRQ_42_43

Address Offset 0x0000 0A90

Physical Address 0x4A00 2A90 **Instance** CTRL_MODULE_CORE

Description

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MPU_IRQ_43	RESERVED	MPU_IRQ_42

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_43		RW	0x26
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_42		RW	0x25

Table 18-361. CTRL_CORE_MPU_IRQ_44_45

Address Offset	0x0000 0A94		
Physical Address	0x4A00 2A94	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	MPU_IRQ_45	RESERVED	MPU_IRQ_44

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_45		RW	0x28
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_44		RW	0x27

Table 18-362. CTRL_CORE_MPU_IRQ_46_47

Address Offset	0x0000 0A98		
Physical Address	0x4A00 2A98	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	MPU_IRQ_47	RESERVED	MPU_IRQ_46

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_47		RW	0x2A
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_46		RW	0x29

Table 18-363. CTRL_CORE_MPU_IRQ_48_49

Address Offset	0x0000 0A9C		
Physical Address	0x4A00 2A9C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	MPU_IRQ_49	RESERVED	MPU_IRQ_48

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_49		RW	0x2C
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:0	MPU_IRQ_48		RW	0x2B

Table 18-364. CTRL_CORE_MPU_IRQ_50_51

Address Offset	0x0000 0AA0		
Physical Address	0x4A00 2AA0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_51									RESERVED							MPU_IRQ_50								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_51		RW	0x2E
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_50		RW	0x2D

Table 18-365. CTRL_CORE_MPU_IRQ_52_53

Address Offset	0x0000 0AA4		
Physical Address	0x4A00 2AA4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_53									RESERVED							MPU_IRQ_52								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_53		RW	0x30
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_52		RW	0x2F

Table 18-366. CTRL_CORE_MPU_IRQ_54_55

Address Offset	0x0000 0AA8		
Physical Address	0x4A00 2AA8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_55									RESERVED							MPU_IRQ_54								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_55		RW	0x32
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_54		RW	0x31

Table 18-367. CTRL_CORE_MPU_IRQ_56_57

Address Offset	0x0000 0AAC		
Physical Address	0x4A00 2AAC	**Instance**	CTRL_MODULE_CORE

Table 18-367. CTRL_CORE_MPU_IRQ_56_57 (continued)

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_57	RESERVED	MPU_IRQ_56

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_57		RW	0x34
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_56		RW	0x33

Table 18-368. CTRL_CORE_MPU_IRQ_58_59

Address Offset 0x0000 0AB0

Physical Address 0x4A00 2AB0 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_59	RESERVED	MPU_IRQ_58

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_59		RW	0x36
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_58		RW	0x35

Table 18-369. CTRL_CORE_MPU_IRQ_60_61

Address Offset 0x0000 0AB4

Physical Address 0x4A00 2AB4 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_61	RESERVED	MPU_IRQ_60

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_61		RW	0x38
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_60		RW	0x37

Table 18-370. CTRL_CORE_MPU_IRQ_62_63

Address Offset 0x0000 0AB8

Physical Address 0x4A00 2AB8 **Instance** CTRL_MODULE_CORE

Description

Type RW

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_63	RESERVED	MPU_IRQ_62

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_63		RW	0x3A
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_62		RW	0x39

Table 18-371. CTRL_CORE_MPU_IRQ_64_65

Address Offset	0x0000 0ABC		
Physical Address	0x4A00 2ABC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_65	RESERVED	MPU_IRQ_64

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_65		RW	0x3C
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_64		RW	0x3B

Table 18-372. CTRL_CORE_MPU_IRQ_66_67

Address Offset	0x0000 0AC0		
Physical Address	0x4A00 2AC0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_67	RESERVED	MPU_IRQ_66

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_67		RW	0x3E
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_66		RW	0x3D

Table 18-373. CTRL_CORE_MPU_IRQ_68_69

Address Offset	0x0000 0AC4		
Physical Address	0x4A00 2AC4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_69	RESERVED	MPU_IRQ_68

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_69		RW	0x40
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_68		RW	0x3F

Table 18-374. CTRL_CORE_MPU_IRQ_70_71

Address Offset	0x0000 0AC8		
Physical Address	0x4A00 2AC8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_71	RESERVED	MPU_IRQ_70

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_71		RW	0x42
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_70		RW	0x41

Table 18-375. CTRL_CORE_MPU_IRQ_72_73

Address Offset	0x0000 0ACC		
Physical Address	0x4A00 2ACC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_73	RESERVED	MPU_IRQ_72

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_73		RW	0x44
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_72		RW	0x43

Table 18-376. CTRL_CORE_MPU_IRQ_74_75

Address Offset	0x0000 0AD0		
Physical Address	0x4A00 2AD0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_75	RESERVED	MPU_IRQ_74

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_75		RW	0x46
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_74		RW	0x45

Table 18-377. CTRL_CORE_MPU_IRQ_76_77

Address Offset	0x0000 0AD4		
Physical Address	0x4A00 2AD4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_77									RESERVED							MPU_IRQ_76								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_77		RW	0x48
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_76		RW	0x47

Table 18-378. CTRL_CORE_MPU_IRQ_78_79

Address Offset 0x0000 0AD8
Physical Address 0x4A00 2AD8 **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_79									RESERVED							MPU_IRQ_78								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_79		RW	0x4A
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_78		RW	0x49

Table 18-379. CTRL_CORE_MPU_IRQ_80_81

Address Offset 0x0000 0ADC
Physical Address 0x4A00 2ADC **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_81									RESERVED							MPU_IRQ_80								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_81		RW	0x4C
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_80		RW	0x4B

Table 18-380. CTRL_CORE_MPU_IRQ_82_83

Address Offset 0x0000 0AE0
Physical Address 0x4A00 2AE0 **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_83									RESERVED							MPU_IRQ_82								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_83		RW	0x4E

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_82		RW	0x4D

Table 18-381. CTRL_CORE_MPU_IRQ_84_85

Address Offset	0x0000 0AE4		
Physical Address	0x4A00 2AE4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9	8 7 6 5 4 3 2 1 0
RESERVED	MPU_IRQ_85	RESERVED	MPU_IRQ_84

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_85		RW	0x50
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_84		RW	0x4F

Table 18-382. CTRL_CORE_MPU_IRQ_86_87

Address Offset	0x0000 0AE8		
Physical Address	0x4A00 2AE8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9	8 7 6 5 4 3 2 1 0
RESERVED	MPU_IRQ_87	RESERVED	MPU_IRQ_86

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_87		RW	0x52
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_86		RW	0x51

Table 18-383. CTRL_CORE_MPU_IRQ_88_89

Address Offset	0x0000 0AEC		
Physical Address	0x4A00 2AEC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9	8 7 6 5 4 3 2 1 0
RESERVED	MPU_IRQ_89	RESERVED	MPU_IRQ_88

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_89		RW	0x54
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_88		RW	0x53

Table 18-384. CTRL_CORE_MPU_IRQ_90_91

Address Offset	0x0000 0AF0

Table 18-384. CTRL_CORE_MPU_IRQ_90_91 (continued)

Physical Address	0x4A00 2AF0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_91	RESERVED	MPU_IRQ_90

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_91		RW	0x56
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_90		RW	0x55

Table 18-385. CTRL_CORE_MPU_IRQ_92_93

Address Offset	0x0000 0AF4		
Physical Address	0x4A00 2AF4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_93	RESERVED	MPU_IRQ_92

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_93		RW	0x58
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_92		RW	0x57

Table 18-386. CTRL_CORE_MPU_IRQ_94_95

Address Offset	0x0000 0AF8		
Physical Address	0x4A00 2AF8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_95	RESERVED	MPU_IRQ_94

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_95		RW	0x5A
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_94		RW	0x59

Table 18-387. CTRL_CORE_MPU_IRQ_96_97

Address Offset	0x0000 0AFC		
Physical Address	0x4A00 2AFC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MPU_IRQ_97	RESERVED	MPU_IRQ_96

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_97		RW	0x5C
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_96		RW	0x5B

Table 18-388. CTRL_CORE_MPU_IRQ_98_99

Address Offset	0x0000 0B00		
Physical Address	0x4A00 2B00	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_99	RESERVED	MPU_IRQ_98

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_99		RW	0x5E
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_98		RW	0x5D

Table 18-389. CTRL_CORE_MPU_IRQ_100_101

Address Offset	0x0000 0B04		
Physical Address	0x4A00 2B04	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_101	RESERVED	MPU_IRQ_100

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_101		RW	0x60
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_100		RW	0x18B

Table 18-390. CTRL_CORE_MPU_IRQ_102_103

Address Offset	0x0000 0B08		
Physical Address	0x4A00 2B08	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_103	RESERVED	MPU_IRQ_102

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_103		RW	0x62
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:0	MPU_IRQ_102		RW	0x61

Table 18-391. CTRL_CORE_MPU_IRQ_104_105

Address Offset	0x0000 0B0C		
Physical Address	0x4A00 2B0C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_105	RESERVED	MPU_IRQ_104

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_105		RW	0x64
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_104		RW	0x63

Table 18-392. CTRL_CORE_MPU_IRQ_106_107

Address Offset	0x0000 0B10		
Physical Address	0x4A00 2B10	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_107	RESERVED	MPU_IRQ_106

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_107		RW	0x66
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_106		RW	0x65

Table 18-393. CTRL_CORE_MPU_IRQ_108_109

Address Offset	0x0000 0B14		
Physical Address	0x4A00 2B14	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_109	RESERVED	MPU_IRQ_108

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_109		RW	0x68
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_108		RW	0x67

Table 18-394. CTRL_CORE_MPU_IRQ_110_111

Address Offset	0x0000 0B18		
Physical Address	0x4A00 2B18	**Instance**	CTRL_MODULE_CORE

Table 18-394. CTRL_CORE_MPU_IRQ_110_111 (continued)

Description	
Type	RW

31–25	24–16	15–9	8–0
RESERVED	MPU_IRQ_111	RESERVED	MPU_IRQ_110

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_111		RW	0x6A
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_110		RW	0x69

Table 18-395. CTRL_CORE_MPU_IRQ_112_113

Address Offset	0x0000 0B1C		
Physical Address	0x4A00 2B1C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	MPU_IRQ_113	RESERVED	MPU_IRQ_112

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_113		RW	0x6C
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_112		RW	0x6B

Table 18-396. CTRL_CORE_MPU_IRQ_114_115

Address Offset	0x0000 0B20		
Physical Address	0x4A00 2B20	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	MPU_IRQ_115	RESERVED	MPU_IRQ_114

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_115		RW	0x6E
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_114		RW	0x6D

Table 18-397. CTRL_CORE_MPU_IRQ_116_117

Address Offset	0x0000 0B24		
Physical Address	0x4A00 2B24	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–25	24–16	15–9	8–0
RESERVED	MPU_IRQ_117	RESERVED	MPU_IRQ_116

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_117		RW	0x70
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_116		RW	0x6F

Table 18-398. CTRL_CORE_MPU_IRQ_118_119

Address Offset	0x0000 0B28		
Physical Address	0x4A00 2B28	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_119									RESERVED							MPU_IRQ_118								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_119		RW	0x72
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_118		RW	0x71

Table 18-399. CTRL_CORE_MPU_IRQ_120_121

Address Offset	0x0000 0B2C		
Physical Address	0x4A00 2B2C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_121									RESERVED							MPU_IRQ_120								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_121		RW	0x74
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_120		RW	0x73

Table 18-400. CTRL_CORE_MPU_IRQ_122_123

Address Offset	0x0000 0B30		
Physical Address	0x4A00 2B30	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_123									RESERVED							MPU_IRQ_122								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_123		RW	0x76
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_122		RW	0x75

Table 18-401. CTRL_CORE_MPU_IRQ_124_125

Address Offset	0x0000 0B34		
Physical Address	0x4A00 2B34	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_125	RESERVED	MPU_IRQ_124

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_125		RW	0x78
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_124		RW	0x77

Table 18-402. CTRL_CORE_MPU_IRQ_126_127

Address Offset	0x0000 0B38		
Physical Address	0x4A00 2B38	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_127	RESERVED	MPU_IRQ_126

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_127		RW	0x7A
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_126		RW	0x79

Table 18-403. CTRL_CORE_MPU_IRQ_128_129

Address Offset	0x0000 0B3C		
Physical Address	0x4A00 2B3C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_129	RESERVED	MPU_IRQ_128

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_129		RW	0x7C
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_128		RW	0x7B

Table 18-404. CTRL_CORE_MPU_IRQ_130_133

Address Offset	0x0000 0B40		
Physical Address	0x4A00 2B40	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_133									RESERVED							MPU_IRQ_130								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_133		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_130		RW	0x7D

Table 18-405. CTRL_CORE_MPU_IRQ_134_135

Address Offset	0x0000 0B44		
Physical Address	0x4A00 2B44	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_135									RESERVED							MPU_IRQ_134								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_135		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_134		RW	0x0

Table 18-406. CTRL_CORE_MPU_IRQ_136_137

Address Offset	0x0000 0B48		
Physical Address	0x4A00 2B48	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_137									RESERVED							MPU_IRQ_136								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_137		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_136		RW	0x0

Table 18-407. CTRL_CORE_MPU_IRQ_138_139

Address Offset	0x0000 0B4C		
Physical Address	0x4A00 2B4C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_139									RESERVED							MPU_IRQ_138								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_139		RW	0x0

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_138		RW	0x0

Table 18-408. CTRL_CORE_MPU_IRQ_140_141

Address Offset	0x0000 0B50		
Physical Address	0x4A00 2B50	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_141	RESERVED	MPU_IRQ_140

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_141		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_140		RW	0x0

Table 18-409. CTRL_CORE_MPU_IRQ_142_143

Address Offset	0x0000 0B54		
Physical Address	0x4A00 2B54	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_143	RESERVED	MPU_IRQ_142

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_143		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_142		RW	0x0

Table 18-410. CTRL_CORE_MPU_IRQ_144_145

Address Offset	0x0000 0B58		
Physical Address	0x4A00 2B58	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_145	RESERVED	MPU_IRQ_144

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_145		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_144		RW	0x0

Table 18-411. CTRL_CORE_MPU_IRQ_146_147

Address Offset 0x0000 0B5C

Table 18-411. CTRL_CORE_MPU_IRQ_146_147 (continued)

Physical Address	0x4A00 2B5C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_147	RESERVED	MPU_IRQ_146

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_147		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_146		RW	0x0

Table 18-412. CTRL_CORE_MPU_IRQ_148_149

Address Offset	0x0000 0B60		
Physical Address	0x4A00 2B60	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_149	RESERVED	MPU_IRQ_148

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_149		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_148		RW	0x0

Table 18-413. CTRL_CORE_MPU_IRQ_150_151

Address Offset	0x0000 0B64		
Physical Address	0x4A00 2B64	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:25	24:16	15:9	8:0
RESERVED	MPU_IRQ_151	RESERVED	MPU_IRQ_150

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_151		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_150		RW	0x0

Table 18-414. CTRL_CORE_MPU_IRQ_152_153

Address Offset	0x0000 0B68		
Physical Address	0x4A00 2B68	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MPU_IRQ_153	RESERVED	MPU_IRQ_152

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_153		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_152		RW	0x0

Table 18-415. CTRL_CORE_MPU_IRQ_154_155

Address Offset	0x0000 0B6C		
Physical Address	0x4A00 2B6C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_155									RESERVED							MPU_IRQ_154								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_155		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_154		RW	0x0

Table 18-416. CTRL_CORE_MPU_IRQ_156_157

Address Offset	0x0000 0B70		
Physical Address	0x4A00 2B70	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_157									RESERVED							MPU_IRQ_156								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_157		RW	0x0
15:9	RESERVED		R	0x0
8:0	MPU_IRQ_156		RW	0x0

Table 18-417. CTRL_CORE_MPU_IRQ_158_159

Address Offset	0x0000 0B74		
Physical Address	0x4A00 2B74	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							MPU_IRQ_159									RESERVED							MPU_IRQ_158								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	MPU_IRQ_159		RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:0	MPU_IRQ_158		RW	0x0

Table 18-418. CTRL_CORE_DMA_SYSTEM_DREQ_0_1

Address Offset	0x0000 0B78		
Physical Address	0x4A00 2B78	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_1_IRQ_1	RESERVED	DMA_SYSTEM_DREQ_0_IRQ_0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_1_IRQ_1		RW	0x2
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_0_IRQ_0		RW	0x1

Table 18-419. CTRL_CORE_DMA_SYSTEM_DREQ_2_3

Address Offset	0x0000 0B7C		
Physical Address	0x4A00 2B7C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_3_IRQ_3	RESERVED	DMA_SYSTEM_DREQ_2_IRQ_2

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_3_IRQ_3		RW	0x4
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_2_IRQ_2		RW	0x3

Table 18-420. CTRL_CORE_DMA_SYSTEM_DREQ_4_5

Address Offset	0x0000 0B80		
Physical Address	0x4A00 2B80	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_5_IRQ_5	RESERVED	DMA_SYSTEM_DREQ_4_IRQ_4

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_5_IRQ_5		RW	0x6
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_SYSTEM_DREQ_4_IRQ_4		RW	0x5

Table 18-421. CTRL_CORE_DMA_SYSTEM_DREQ_6_7

Address Offset	0x0000 0B84		
Physical Address	0x4A00 2B84	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_7_IRQ_7	RESERVED	DMA_SYSTEM_DREQ_6_IRQ_6

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_7_IRQ_7		RW	0x8
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_6_IRQ_6		RW	0x7

Table 18-422. CTRL_CORE_DMA_SYSTEM_DREQ_8_9

Address Offset	0x0000 0B88		
Physical Address	0x4A00 2B88	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_9_IRQ_9	RESERVED	DMA_SYSTEM_DREQ_8_IRQ_8

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_9_IRQ_9		RW	0xA
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_8_IRQ_8		RW	0x9

Table 18-423. CTRL_CORE_DMA_SYSTEM_DREQ_10_11

Address Offset	0x0000 0B8C		
Physical Address	0x4A00 2B8C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_11_IRQ_11	RESERVED	DMA_SYSTEM_DREQ_10_IRQ_10

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_11_IRQ_11		RW	0xC
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_SYSTEM_DREQ_10_IRQ_10		RW	0xB

Table 18-424. CTRL_CORE_DMA_SYSTEM_DREQ_12_13

Address Offset	0x0000 0B90		
Physical Address	0x4A00 2B90	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_13_IRQ_13	RESERVED	DMA_SYSTEM_DREQ_12_IRQ_12

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_13_IRQ_13		RW	0xE
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_12_IRQ_12		RW	0xD

Table 18-425. CTRL_CORE_DMA_SYSTEM_DREQ_14_15

Address Offset	0x0000 0B94		
Physical Address	0x4A00 2B94	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_15_IRQ_15	RESERVED	DMA_SYSTEM_DREQ_14_IRQ_14

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_15_IRQ_15		RW	0x10
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_14_IRQ_14		RW	0xF

Table 18-426. CTRL_CORE_DMA_SYSTEM_DREQ_16_17

Address Offset	0x0000 0B98		
Physical Address	0x4A00 2B98	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_17_IRQ_17	RESERVED	DMA_SYSTEM_DREQ_16_IRQ_16

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
23:16	DMA_SYSTEM_DREQ_17_IRQ_17		RW	0x12
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_16_IRQ_16		RW	0x11

Table 18-427. CTRL_CORE_DMA_SYSTEM_DREQ_18_19

Address Offset	0x0000 0B9C		
Physical Address	0x4A00 2B9C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_19_IRQ_19	RESERVED	DMA_SYSTEM_DREQ_18_IRQ_18

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_19_IRQ_19		RW	0x14
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_18_IRQ_18		RW	0x13

Table 18-428. CTRL_CORE_DMA_SYSTEM_DREQ_20_21

Address Offset	0x0000 0BA0		
Physical Address	0x4A00 2BA0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_21_IRQ_21	RESERVED	DMA_SYSTEM_DREQ_20_IRQ_20

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_21_IRQ_21		RW	0x16
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_20_IRQ_20		RW	0x15

Table 18-429. CTRL_CORE_DMA_SYSTEM_DREQ_22_23

Address Offset	0x0000 0BA4		
Physical Address	0x4A00 2BA4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_23_IRQ_23	RESERVED	DMA_SYSTEM_DREQ_22_IRQ_22

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_23_IRQ_23		RW	0x18
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_22_IRQ_22		RW	0x17

Table 18-430. CTRL_CORE_DMA_SYSTEM_DREQ_24_25

Address Offset	0x0000 0BA8		
Physical Address	0x4A00 2BA8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_25_IRQ_25	RESERVED	DMA_SYSTEM_DREQ_24_IRQ_24

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_25_IRQ_25		RW	0x1A
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_24_IRQ_24		RW	0x19

Table 18-431. CTRL_CORE_DMA_SYSTEM_DREQ_26_27

Address Offset	0x0000 0BAC		
Physical Address	0x4A00 2BAC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_27_IRQ_27	RESERVED	DMA_SYSTEM_DREQ_26_IRQ_26

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_27_IRQ_27		RW	0x1C
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_26_IRQ_26		RW	0x1B

Table 18-432. CTRL_CORE_DMA_SYSTEM_DREQ_28_29

Address Offset	0x0000 0BB0		
Physical Address	0x4A00 2BB0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DMA_SYSTEM_DREQ_29_IRQ_29	RESERVED	DMA_SYSTEM_DREQ_28_IRQ_28

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_29_IRQ_29		RW	0x1E
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_28_IRQ_28		RW	0x1D

Table 18-433. CTRL_CORE_DMA_SYSTEM_DREQ_30_31

Address Offset	0x0000 0BB4		
Physical Address	0x4A00 2BB4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_31_IRQ_31								RESERVED								DMA_SYSTEM_DREQ_30_IRQ_30							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_31_IRQ_31		RW	0x20
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_30_IRQ_30		RW	0x1F

Table 18-434. CTRL_CORE_DMA_SYSTEM_DREQ_32_33

Address Offset	0x0000 0BB8		
Physical Address	0x4A00 2BB8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_33_IRQ_33								RESERVED								DMA_SYSTEM_DREQ_32_IRQ_32							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_33_IRQ_33		RW	0x22
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_32_IRQ_32		RW	0x21

Table 18-435. CTRL_CORE_DMA_SYSTEM_DREQ_34_35

Address Offset	0x0000 0BBC		
Physical Address	0x4A00 2BBC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_35_IRQ_35								RESERVED								DMA_SYSTEM_DREQ_34_IRQ_34							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_35_IRQ_35		RW	0x24
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_34_IRQ_34		RW	0x23

Table 18-436. CTRL_CORE_DMA_SYSTEM_DREQ_36_37

Address Offset	0x0000 0BC0		
Physical Address	0x4A00 2BC0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_37_IRQ_37								RESERVED								DMA_SYSTEM_DREQ_36_IRQ_36							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_37_IRQ_37		RW	0x26
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_36_IRQ_36		RW	0x25

Table 18-437. CTRL_CORE_DMA_SYSTEM_DREQ_38_39

Address Offset	0x0000 0BC4		
Physical Address	0x4A00 2BC4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_39_IRQ_39								RESERVED								DMA_SYSTEM_DREQ_38_IRQ_38							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_39_IRQ_39		RW	0x28
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_38_IRQ_38		RW	0x27

Table 18-438. CTRL_CORE_DMA_SYSTEM_DREQ_40_41

Address Offset	0x0000 0BC8		
Physical Address	0x4A00 2BC8	**Instance**	CTRL_MODULE_CORE
Description			

Table 18-438. CTRL_CORE_DMA_SYSTEM_DREQ_40_41 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_41_IRQ_41								RESERVED								DMA_SYSTEM_DREQ_40_IRQ_40							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_41_IRQ_41		RW	0x2A
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_40_IRQ_40		RW	0x29

Table 18-439. CTRL_CORE_DMA_SYSTEM_DREQ_42_43

Address Offset	0x0000 0BCC		
Physical Address	0x4A00 2BCC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_43_IRQ_43								RESERVED								DMA_SYSTEM_DREQ_42_IRQ_42							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_43_IRQ_43		RW	0x2C
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_42_IRQ_42		RW	0x2B

Table 18-440. CTRL_CORE_DMA_SYSTEM_DREQ_44_45

Address Offset	0x0000 0BD0		
Physical Address	0x4A00 2BD0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_45_IRQ_45								RESERVED								DMA_SYSTEM_DREQ_44_IRQ_44							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_45_IRQ_45		RW	0x2E
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_44_IRQ_44		RW	0x2D

Table 18-441. CTRL_CORE_DMA_SYSTEM_DREQ_46_47

Address Offset	0x0000 0BD4		
Physical Address	0x4A00 2BD4	**Instance**	CTRL_MODULE_CORE

Table 18-441. CTRL_CORE_DMA_SYSTEM_DREQ_46_47 (continued)

Description	
Type	RW

31–24	23–16	15–8	7–0
RESERVED	DMA_SYSTEM_DREQ_47_IRQ_47	RESERVED	DMA_SYSTEM_DREQ_46_IRQ_46

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_47_IRQ_47		RW	0x30
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_46_IRQ_46		RW	0x2F

Table 18-442. CTRL_CORE_DMA_SYSTEM_DREQ_48_49

Address Offset	0x0000 0BD8		
Physical Address	0x4A00 2BD8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_SYSTEM_DREQ_49_IRQ_49	RESERVED	DMA_SYSTEM_DREQ_48_IRQ_48

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_49_IRQ_49		RW	0x32
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_48_IRQ_48		RW	0x31

Table 18-443. CTRL_CORE_DMA_SYSTEM_DREQ_50_51

Address Offset	0x0000 0BDC		
Physical Address	0x4A00 2BDC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_SYSTEM_DREQ_51_IRQ_51	RESERVED	DMA_SYSTEM_DREQ_50_IRQ_50

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_51_IRQ_51		RW	0x34
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_50_IRQ_50		RW	0x33

Table 18-444. CTRL_CORE_DMA_SYSTEM_DREQ_52_53

Address Offset	0x0000 0BE0

Table 18-444. CTRL_CORE_DMA_SYSTEM_DREQ_52_53 (continued)

Physical Address	0x4A00 2BE0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_53_IRQ_53	RESERVED	DMA_SYSTEM_DREQ_52_IRQ_52

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_53_IRQ_53		RW	0x36
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_52_IRQ_52		RW	0x35

Table 18-445. CTRL_CORE_DMA_SYSTEM_DREQ_54_55

Address Offset	0x0000 0BE4		
Physical Address	0x4A00 2BE4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_55_IRQ_55	RESERVED	DMA_SYSTEM_DREQ_54_IRQ_54

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_55_IRQ_55		RW	0x38
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_54_IRQ_54		RW	0x37

Table 18-446. CTRL_CORE_DMA_SYSTEM_DREQ_56_57

Address Offset	0x0000 0BE8		
Physical Address	0x4A00 2BE8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_57_IRQ_57	RESERVED	DMA_SYSTEM_DREQ_56_IRQ_56

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_57_IRQ_57		RW	0x3A
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_56_IRQ_56		RW	0x39

Table 18-447. CTRL_CORE_DMA_SYSTEM_DREQ_58_59

Address Offset	0x0000 0BEC		
Physical Address	0x4A00 2BEC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_59_IRQ_59	RESERVED	DMA_SYSTEM_DREQ_58_IRQ_58

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_59_IRQ_59		RW	0x3C
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_58_IRQ_58		RW	0x3B

Table 18-448. CTRL_CORE_DMA_SYSTEM_DREQ_60_61

Address Offset	0x0000 0BF0		
Physical Address	0x4A00 2BF0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_61_IRQ_61	RESERVED	DMA_SYSTEM_DREQ_60_IRQ_60

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_61_IRQ_61		RW	0x3E
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_60_IRQ_60		RW	0x3D

Table 18-449. CTRL_CORE_DMA_SYSTEM_DREQ_62_63

Address Offset	0x0000 0BF4		
Physical Address	0x4A00 2BF4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_63_IRQ_63	RESERVED	DMA_SYSTEM_DREQ_62_IRQ_62

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_63_IRQ_63		RW	0x40
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_SYSTEM_DREQ_62_IRQ_62		RW	0x3F

Table 18-450. CTRL_CORE_DMA_SYSTEM_DREQ_64_65

Address Offset	0x0000 0BF8		
Physical Address	0x4A00 2BF8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_65_IRQ_65	RESERVED	DMA_SYSTEM_DREQ_64_IRQ_64

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_65_IRQ_65		RW	0x42
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_64_IRQ_64		RW	0x41

Table 18-451. CTRL_CORE_DMA_SYSTEM_DREQ_66_67

Address Offset	0x0000 0BFC		
Physical Address	0x4A00 2BFC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_67_IRQ_67	RESERVED	DMA_SYSTEM_DREQ_66_IRQ_66

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_67_IRQ_67		RW	0x44
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_66_IRQ_66		RW	0x43

Table 18-452. CTRL_CORE_DMA_SYSTEM_DREQ_68_69

Address Offset	0x0000 0C00		
Physical Address	0x4A00 2C00	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_69_IRQ_69	RESERVED	DMA_SYSTEM_DREQ_68_IRQ_68

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
23:16	DMA_SYSTEM_DREQ_69_IRQ_69		RW	0x46
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_68_IRQ_68		RW	0x45

Table 18-453. CTRL_CORE_DMA_SYSTEM_DREQ_70_71

Address Offset	0x0000 0C04		
Physical Address	0x4A00 2C04	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_71_IRQ_71	RESERVED	DMA_SYSTEM_DREQ_70_IRQ_70

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_71_IRQ_71		RW	0x48
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_70_IRQ_70		RW	0x47

Table 18-454. CTRL_CORE_DMA_SYSTEM_DREQ_72_73

Address Offset	0x0000 0C08		
Physical Address	0x4A00 2C08	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_73_IRQ_73	RESERVED	DMA_SYSTEM_DREQ_72_IRQ_72

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_73_IRQ_73		RW	0x4A
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_72_IRQ_72		RW	0x49

Table 18-455. CTRL_CORE_DMA_SYSTEM_DREQ_74_75

Address Offset	0x0000 0C0C		
Physical Address	0x4A00 2C0C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_75_IRQ_75	RESERVED	DMA_SYSTEM_DREQ_74_IRQ_74

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_75_IRQ_75		RW	0x4C
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_74_IRQ_74		RW	0x4B

Table 18-456. CTRL_CORE_DMA_SYSTEM_DREQ_76_77

Address Offset	0x0000 0C10		
Physical Address	0x4A00 2C10	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_77_IRQ_77								RESERVED								DMA_SYSTEM_DREQ_76_IRQ_76							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_77_IRQ_77		RW	0x4E
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_76_IRQ_76		RW	0x4D

Table 18-457. CTRL_CORE_DMA_SYSTEM_DREQ_78_79

Address Offset	0x0000 0C14		
Physical Address	0x4A00 2C14	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_79_IRQ_79								RESERVED								DMA_SYSTEM_DREQ_78_IRQ_78							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_79_IRQ_79		RW	0x50
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_78_IRQ_78		RW	0x4F

Table 18-458. CTRL_CORE_DMA_SYSTEM_DREQ_80_81

Address Offset	0x0000 0C18		
Physical Address	0x4A00 2C18	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DMA_SYSTEM_DREQ_81_IRQ_81	RESERVED	DMA_SYSTEM_DREQ_80_IRQ_80

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_81_IRQ_81		RW	0x52
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_80_IRQ_80		RW	0x51

Table 18-459. CTRL_CORE_DMA_SYSTEM_DREQ_82_83

Address Offset	0x0000 0C1C		
Physical Address	0x4A00 2C1C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_83_IRQ_83	RESERVED	DMA_SYSTEM_DREQ_82_IRQ_82

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_83_IRQ_83		RW	0x54
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_82_IRQ_82		RW	0x53

Table 18-460. CTRL_CORE_DMA_SYSTEM_DREQ_84_85

Address Offset	0x0000 0C20		
Physical Address	0x4A00 2C20	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_85_IRQ_85	RESERVED	DMA_SYSTEM_DREQ_84_IRQ_84

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_85_IRQ_85		RW	0x56
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_84_IRQ_84		RW	0x55

Table 18-461. CTRL_CORE_DMA_SYSTEM_DREQ_86_87

Address Offset	0x0000 0C24		
Physical Address	0x4A00 2C24	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_87_IRQ_87								RESERVED								DMA_SYSTEM_DREQ_86_IRQ_86							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_87_IRQ_87		RW	0x58
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_86_IRQ_86		RW	0x57

Table 18-462. CTRL_CORE_DMA_SYSTEM_DREQ_88_89

Address Offset	0x0000 0C28		
Physical Address	0x4A00 2C28	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_89_IRQ_89								RESERVED								DMA_SYSTEM_DREQ_88_IRQ_88							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_89_IRQ_89		RW	0x5A
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_88_IRQ_88		RW	0x59

Table 18-463. CTRL_CORE_DMA_SYSTEM_DREQ_90_91

Address Offset	0x0000 0C2C		
Physical Address	0x4A00 2C2C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_SYSTEM_DREQ_91_IRQ_91								RESERVED								DMA_SYSTEM_DREQ_90_IRQ_90							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_91_IRQ_91		RW	0x5C
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_90_IRQ_90		RW	0x5B

Table 18-464. CTRL_CORE_DMA_SYSTEM_DREQ_92_93

Address Offset	0x0000 0C30		
Physical Address	0x4A00 2C30	**Instance**	CTRL_MODULE_CORE
Description			

Table 18-464. CTRL_CORE_DMA_SYSTEM_DREQ_92_93 (continued)

Type	RW

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_93_IRQ_93	RESERVED	DMA_SYSTEM_DREQ_92_IRQ_92

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_93_IRQ_93		RW	0x5E
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_92_IRQ_92		RW	0x5D

Table 18-465. CTRL_CORE_DMA_SYSTEM_DREQ_94_95

Address Offset	0x0000 0C34		
Physical Address	0x4A00 2C34	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_95_IRQ_95	RESERVED	DMA_SYSTEM_DREQ_94_IRQ_94

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_95_IRQ_95		RW	0x60
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_94_IRQ_94		RW	0x5F

Table 18-466. CTRL_CORE_DMA_SYSTEM_DREQ_96_97

Address Offset	0x0000 0C38		
Physical Address	0x4A00 2C38	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_97_IRQ_97	RESERVED	DMA_SYSTEM_DREQ_96_IRQ_96

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_97_IRQ_97		RW	0x62
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_96_IRQ_96		RW	0x61

Table 18-467. CTRL_CORE_DMA_SYSTEM_DREQ_98_99

Address Offset	0x0000 0C3C		
Physical Address	0x4A00 2C3C	**Instance**	CTRL_MODULE_CORE

Table 18-467. CTRL_CORE_DMA_SYSTEM_DREQ_98_99 (continued)

Description

Type RW

31–24	23–16	15–8	7–0
RESERVED	DMA_SYSTEM_DREQ_99_IRQ_99	RESERVED	DMA_SYSTEM_DREQ_98_IRQ_98

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_99_IRQ_99		RW	0x64
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_98_IRQ_98		RW	0x63

Table 18-468. CTRL_CORE_DMA_SYSTEM_DREQ_100_101

Address Offset 0x0000 0C40

Physical Address 0x4A00 2C40 **Instance** CTRL_MODULE_CORE

Description

Type RW

31–24	23–16	15–8	7–0
RESERVED	DMA_SYSTEM_DREQ_101_IRQ_101	RESERVED	DMA_SYSTEM_DREQ_100_IRQ_100

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_101_IRQ_101		RW	0x66
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_100_IRQ_100		RW	0x65

Table 18-469. CTRL_CORE_DMA_SYSTEM_DREQ_102_103

Address Offset 0x0000 0C44

Physical Address 0x4A00 2C44 **Instance** CTRL_MODULE_CORE

Description

Type RW

31–24	23–16	15–8	7–0
RESERVED	DMA_SYSTEM_DREQ_103_IRQ_103	RESERVED	DMA_SYSTEM_DREQ_102_IRQ_102

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_103_IRQ_103		RW	0x68
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_102_IRQ_102		RW	0x67

Table 18-470. CTRL_CORE_DMA_SYSTEM_DREQ_104_105

Address Offset 0x0000 0C48

Table 18-470. CTRL_CORE_DMA_SYSTEM_DREQ_104_105 (continued)

Physical Address	0x4A00 2C48	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_105_IRQ_105	RESERVED	DMA_SYSTEM_DREQ_104_IRQ_104

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_105_IRQ_105		RW	0x6A
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_104_IRQ_104		RW	0x69

Table 18-471. CTRL_CORE_DMA_SYSTEM_DREQ_106_107

Address Offset	0x0000 0C4C		
Physical Address	0x4A00 2C4C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_107_IRQ_107	RESERVED	DMA_SYSTEM_DREQ_106_IRQ_106

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_107_IRQ_107		RW	0x6C
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_106_IRQ_106		RW	0x6B

Table 18-472. CTRL_CORE_DMA_SYSTEM_DREQ_108_109

Address Offset	0x0000 0C50		
Physical Address	0x4A00 2C50	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_SYSTEM_DREQ_109_IRQ_109	RESERVED	DMA_SYSTEM_DREQ_108_IRQ_108

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_109_IRQ_109		RW	0x6E
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_108_IRQ_108		RW	0x6D

Table 18-473. CTRL_CORE_DMA_SYSTEM_DREQ_110_111

Address Offset	0x0000 0C54		
Physical Address	0x4A00 2C54	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_111_IRQ_111	RESERVED	DMA_SYSTEM_DREQ_110_IRQ_110

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_111_IRQ_111		RW	0x70
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_110_IRQ_110		RW	0x6F

Table 18-474. CTRL_CORE_DMA_SYSTEM_DREQ_112_113

Address Offset	0x0000 0C58		
Physical Address	0x4A00 2C58	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_113_IRQ_113	RESERVED	DMA_SYSTEM_DREQ_112_IRQ_112

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_113_IRQ_113		RW	0x72
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_112_IRQ_112		RW	0x71

Table 18-475. CTRL_CORE_DMA_SYSTEM_DREQ_114_115

Address Offset	0x0000 0C5C		
Physical Address	0x4A00 2C5C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_115_IRQ_115	RESERVED	DMA_SYSTEM_DREQ_114_IRQ_114

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_115_IRQ_115		RW	0x74
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_SYSTEM_DREQ_114_IRQ_114		RW	0x73

Table 18-476. CTRL_CORE_DMA_SYSTEM_DREQ_116_117

Address Offset	0x0000 0C60		
Physical Address	0x4A00 2C60	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_117_IRQ_117	RESERVED	DMA_SYSTEM_DREQ_116_IRQ_116

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_117_IRQ_117		RW	0x76
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_116_IRQ_116		RW	0x75

Table 18-477. CTRL_CORE_DMA_SYSTEM_DREQ_118_119

Address Offset	0x0000 0C64		
Physical Address	0x4A00 2C64	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_119_IRQ_119	RESERVED	DMA_SYSTEM_DREQ_118_IRQ_118

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_119_IRQ_119		RW	0x78
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_118_IRQ_118		RW	0x77

Table 18-478. CTRL_CORE_DMA_SYSTEM_DREQ_120_121

Address Offset	0x0000 0C68		
Physical Address	0x4A00 2C68	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_121_IRQ_121	RESERVED	DMA_SYSTEM_DREQ_120_IRQ_120

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
23:16	DMA_SYSTEM_DREQ_121_IRQ_121		RW	0x7A
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_120_IRQ_120		RW	0x79

Table 18-479. CTRL_CORE_DMA_SYSTEM_DREQ_122_123

Address Offset	0x0000 0C6C		
Physical Address	0x4A00 2C6C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_123_IRQ_123	RESERVED	DMA_SYSTEM_DREQ_122_IRQ_122

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_123_IRQ_123		RW	0x7C
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_122_IRQ_122		RW	0x7B

Table 18-480. CTRL_CORE_DMA_SYSTEM_DREQ_124_125

Address Offset	0x0000 0C70		
Physical Address	0x4A00 2C70	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_SYSTEM_DREQ_125_IRQ_125	RESERVED	DMA_SYSTEM_DREQ_124_IRQ_124

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_SYSTEM_DREQ_125_IRQ_125		RW	0x7E
15:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_124_IRQ_124		RW	0x7D

Table 18-481. CTRL_CORE_DMA_SYSTEM_DREQ_126_127

Address Offset	0x0000 0C74		
Physical Address	0x4A00 2C74	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:8	7:0
RESERVED	DMA_SYSTEM_DREQ_126_IRQ_126

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	DMA_SYSTEM_DREQ_126_IRQ_126		RW	0x7F

Table 18-482. CTRL_CORE_DMA_EDMA_DREQ_0_1

Address Offset	0x0000 0C78		
Physical Address	0x4A00 2C78	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_1_IRQ_1	RESERVED	DMA_EDMA_DREQ_0_IRQ_0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_1_IRQ_1		RW	0x2
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_0_IRQ_0		RW	0x1

Table 18-483. CTRL_CORE_DMA_EDMA_DREQ_2_3

Address Offset	0x0000 0C7C		
Physical Address	0x4A00 2C7C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_3_IRQ_3	RESERVED	DMA_EDMA_DREQ_2_IRQ_2

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_3_IRQ_3		RW	0x4
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_2_IRQ_2		RW	0x3

Table 18-484. CTRL_CORE_DMA_EDMA_DREQ_4_5

Address Offset	0x0000 0C80		
Physical Address	0x4A00 2C80	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_5_IRQ_5	RESERVED	DMA_EDMA_DREQ_4_IRQ_4

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_5_IRQ_5		RW	0x6
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_4_IRQ_4		RW	0x5

Table 18-485. CTRL_CORE_DMA_EDMA_DREQ_6_7

Address Offset	0x0000 0C84		
Physical Address	0x4A00 2C84	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_7_IRQ_7	RESERVED	DMA_EDMA_DREQ_6_IRQ_6

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_7_IRQ_7		RW	0x8
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_6_IRQ_6		RW	0x7

Table 18-486. CTRL_CORE_DMA_EDMA_DREQ_8_9

Address Offset	0x0000 0C88		
Physical Address	0x4A00 2C88	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_9_IRQ_9	RESERVED	DMA_EDMA_DREQ_8_IRQ_8

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_9_IRQ_9		RW	0xA
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_8_IRQ_8		RW	0x9

Table 18-487. CTRL_CORE_DMA_EDMA_DREQ_10_11

Address Offset	0x0000 0C8C		
Physical Address	0x4A00 2C8C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_11_IRQ_11	RESERVED	DMA_EDMA_DREQ_10_IRQ_10

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_11_IRQ_11		RW	0xC
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_10_IRQ_10		RW	0xB

Table 18-488. CTRL_CORE_DMA_EDMA_DREQ_12_13

Address Offset	0x0000 0C90		
Physical Address	0x4A00 2C90	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_13_IRQ_13	RESERVED	DMA_EDMA_DREQ_12_IRQ_12

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_13_IRQ_13		RW	0xE
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_12_IRQ_12		RW	0xD

Table 18-489. CTRL_CORE_DMA_EDMA_DREQ_14_15

Address Offset	0x0000 0C94		
Physical Address	0x4A00 2C94	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_15_IRQ_15	RESERVED	DMA_EDMA_DREQ_14_IRQ_14

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_15_IRQ_15		RW	0x10
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_14_IRQ_14		RW	0xF

Table 18-490. CTRL_CORE_DMA_EDMA_DREQ_16_17

Address Offset	0x0000 0C98		
Physical Address	0x4A00 2C98	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_17_IRQ_17	RESERVED	DMA_EDMA_DREQ_16_IRQ_16

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_17_IRQ_17		RW	0x12
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_16_IRQ_16		RW	0x11

Table 18-491. CTRL_CORE_DMA_EDMA_DREQ_18_19

Address Offset	0x0000 0C9C		
Physical Address	0x4A00 2C9C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_19_IRQ_19	RESERVED	DMA_EDMA_DREQ_18_IRQ_18

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_19_IRQ_19		RW	0x14
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_18_IRQ_18		RW	0x13

Table 18-492. CTRL_CORE_DMA_EDMA_DREQ_20_21

Address Offset	0x0000 0CA0		
Physical Address	0x4A00 2CA0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_21_IRQ_21	RESERVED	DMA_EDMA_DREQ_20_IRQ_20

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_21_IRQ_21		RW	0x16
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_20_IRQ_20		RW	0x15

Table 18-493. CTRL_CORE_DMA_EDMA_DREQ_22_23

Address Offset	0x0000 0CA4		
Physical Address	0x4A00 2CA4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_23_IRQ_23	RESERVED	DMA_EDMA_DREQ_22_IRQ_22

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_23_IRQ_23		RW	0x18
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_22_IRQ_22		RW	0x17

Table 18-494. CTRL_CORE_DMA_EDMA_DREQ_24_25

Address Offset	0x0000 0CA8		
Physical Address	0x4A00 2CA8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_25_IRQ_25	RESERVED	DMA_EDMA_DREQ_24_IRQ_24

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_25_IRQ_25		RW	0x1A
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_24_IRQ_24		RW	0x19

Table 18-495. CTRL_CORE_DMA_EDMA_DREQ_26_27

Address Offset	0x0000 0CAC		
Physical Address	0x4A00 2CAC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_27_IRQ_27	RESERVED	DMA_EDMA_DREQ_26_IRQ_26

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_27_IRQ_27		RW	0x1C
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_26_IRQ_26		RW	0x1B

Table 18-496. CTRL_CORE_DMA_EDMA_DREQ_28_29

Address Offset	0x0000 0CB0		
Physical Address	0x4A00 2CB0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_29_IRQ_29	RESERVED	DMA_EDMA_DREQ_28_IRQ_28

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_29_IRQ_29		RW	0x1E
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_28_IRQ_28		RW	0x1D

Table 18-497. CTRL_CORE_DMA_EDMA_DREQ_30_31

Address Offset	0x0000 0CB4		
Physical Address	0x4A00 2CB4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_31_IRQ_31	RESERVED	DMA_EDMA_DREQ_30_IRQ_30

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_31_IRQ_31		RW	0x20
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_30_IRQ_30		RW	0x1F

Table 18-498. CTRL_CORE_DMA_EDMA_DREQ_32_33

Address Offset	0x0000 0CB8		
Physical Address	0x4A00 2CB8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_33_IRQ_33	RESERVED	DMA_EDMA_DREQ_32_IRQ_32

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_33_IRQ_33		RW	0x22
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_32_IRQ_32		RW	0x21

Table 18-499. CTRL_CORE_DMA_EDMA_DREQ_34_35

Address Offset	0x0000 0CBC		
Physical Address	0x4A00 2CBC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_35_IRQ_35	RESERVED	DMA_EDMA_DREQ_34_IRQ_34

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_35_IRQ_35		RW	0x24
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_34_IRQ_34		RW	0x23

Table 18-500. CTRL_CORE_DMA_EDMA_DREQ_36_37

Address Offset	0x0000 0CC0		
Physical Address	0x4A00 2CC0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_37_IRQ_37	RESERVED	DMA_EDMA_DREQ_36_IRQ_36

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_37_IRQ_37		RW	0x26
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_36_IRQ_36		RW	0x25

Table 18-501. CTRL_CORE_DMA_EDMA_DREQ_38_39

Address Offset	0x0000 0CC4		
Physical Address	0x4A00 2CC4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_39_IRQ_39	RESERVED	DMA_EDMA_DREQ_38_IRQ_38

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_39_IRQ_39		RW	0x28
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_38_IRQ_38		RW	0x27

Table 18-502. CTRL_CORE_DMA_EDMA_DREQ_40_41

Address Offset	0x0000 0CC8		
Physical Address	0x4A00 2CC8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_41_IRQ_41	RESERVED	DMA_EDMA_DREQ_40_IRQ_40

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_41_IRQ_41		RW	0x2A
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_40_IRQ_40		RW	0x29

Table 18-503. CTRL_CORE_DMA_EDMA_DREQ_42_43

Address Offset	0x0000 0CCC		
Physical Address	0x4A00 2CCC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_43_IRQ_43	RESERVED	DMA_EDMA_DREQ_42_IRQ_42

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_43_IRQ_43		RW	0x2C
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_42_IRQ_42		RW	0x2B

Table 18-504. CTRL_CORE_DMA_EDMA_DREQ_44_45

Address Offset	0x0000 0CD0		
Physical Address	0x4A00 2CD0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_45_IRQ_45	RESERVED	DMA_EDMA_DREQ_44_IRQ_44

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_45_IRQ_45		RW	0x2E
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_44_IRQ_44		RW	0x2D

Table 18-505. CTRL_CORE_DMA_EDMA_DREQ_46_47

Address Offset	0x0000 0CD4		
Physical Address	0x4A00 2CD4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_47_IRQ_47	RESERVED	DMA_EDMA_DREQ_46_IRQ_46

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_47_IRQ_47		RW	0x30
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_46_IRQ_46		RW	0x2F

Table 18-506. CTRL_CORE_DMA_EDMA_DREQ_48_49

Address Offset	0x0000 0CD8		
Physical Address	0x4A00 2CD8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_49_IRQ_49	RESERVED	DMA_EDMA_DREQ_48_IRQ_48

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_49_IRQ_49		RW	0x32
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_48_IRQ_48		RW	0x31

Table 18-507. CTRL_CORE_DMA_EDMA_DREQ_50_51

Address Offset	0x0000 0CDC		
Physical Address	0x4A00 2CDC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_51_IRQ_51	RESERVED	DMA_EDMA_DREQ_50_IRQ_50

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_51_IRQ_51		RW	0x34
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_50_IRQ_50		RW	0x33

Table 18-508. CTRL_CORE_DMA_EDMA_DREQ_52_53

Address Offset	0x0000 0CE0		
Physical Address	0x4A00 2CE0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_EDMA_DREQ_53_IRQ_53	RESERVED	DMA_EDMA_DREQ_52_IRQ_52

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_53_IRQ_53		RW	0x36
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_52_IRQ_52		RW	0x35

Table 18-509. CTRL_CORE_DMA_EDMA_DREQ_54_55

Address Offset	0x0000 0CE4		
Physical Address	0x4A00 2CE4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_55_IRQ_55	RESERVED	DMA_EDMA_DREQ_54_IRQ_54

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_55_IRQ_55		RW	0x38
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_54_IRQ_54		RW	0x37

Table 18-510. CTRL_CORE_DMA_EDMA_DREQ_56_57

Address Offset	0x0000 0CE8		
Physical Address	0x4A00 2CE8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_57_IRQ_57	RESERVED	DMA_EDMA_DREQ_56_IRQ_56

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_57_IRQ_57		RW	0x3A
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_56_IRQ_56		RW	0x39

Table 18-511. CTRL_CORE_DMA_EDMA_DREQ_58_59

Address Offset	0x0000 0CEC		
Physical Address	0x4A00 2CEC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_EDMA_DREQ_59_IRQ_59	RESERVED	DMA_EDMA_DREQ_58_IRQ_58

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_59_IRQ_59		RW	0x3C
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_EDMA_DREQ_58_IRQ_58		RW	0x3B

Table 18-512. CTRL_CORE_DMA_EDMA_DREQ_60_61

Address Offset	0x0000 0CF0		
Physical Address	0x4A00 2CF0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_61_IRQ_61								RESERVED								DMA_EDMA_DREQ_60_IRQ_60							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_61_IRQ_61		RW	0x3E
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_60_IRQ_60		RW	0x3D

Table 18-513. CTRL_CORE_DMA_EDMA_DREQ_62_63

Address Offset	0x0000 0CF4		
Physical Address	0x4A00 2CF4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_63_IRQ_63								RESERVED								DMA_EDMA_DREQ_62_IRQ_62							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_63_IRQ_63		RW	0x40
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_62_IRQ_62		RW	0x3F

Table 18-514. CTRL_CORE_DMA_DSP1_DREQ_0_1

Address Offset	0x0000 0CF8		
Physical Address	0x4A00 2CF8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_1_IRQ_1								RESERVED								DMA_DSP1_DREQ_0_IRQ_0							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_1_IRQ_1		RW	0x81
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_DSP1_DREQ_0_IRQ_0		RW	0x80

Table 18-515. CTRL_CORE_DMA_DSP1_DREQ_2_3

Address Offset	0x0000 0CFC		
Physical Address	0x4A00 2CFC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_DSP1_DREQ_3_IRQ_3	RESERVED	DMA_DSP1_DREQ_2_IRQ_2

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_3_IRQ_3		RW	0x83
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_2_IRQ_2		RW	0x82

Table 18-516. CTRL_CORE_DMA_DSP1_DREQ_4_5

Address Offset	0x0000 0D00		
Physical Address	0x4A00 2D00	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_DSP1_DREQ_5_IRQ_5	RESERVED	DMA_DSP1_DREQ_4_IRQ_4

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_5_IRQ_5		RW	0x85
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_4_IRQ_4		RW	0x84

Table 18-517. CTRL_CORE_DMA_DSP1_DREQ_6_7

Address Offset	0x0000 0D04		
Physical Address	0x4A00 2D04	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_DSP1_DREQ_7_IRQ_7	RESERVED	DMA_DSP1_DREQ_6_IRQ_6

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_7_IRQ_7		RW	0x87
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_DSP1_DREQ_6_IRQ_6		RW	0x86

Table 18-518. CTRL_CORE_DMA_DSP1_DREQ_8_9

Address Offset	0x0000 0D08		
Physical Address	0x4A00 2D08	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP1_DREQ_9_IRQ_9	RESERVED	DMA_DSP1_DREQ_8_IRQ_8

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_9_IRQ_9		RW	0x89
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_8_IRQ_8		RW	0x88

Table 18-519. CTRL_CORE_DMA_DSP1_DREQ_10_11

Address Offset	0x0000 0D0C		
Physical Address	0x4A00 2D0C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP1_DREQ_11_IRQ_11	RESERVED	DMA_DSP1_DREQ_10_IRQ_10

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_11_IRQ_11		RW	0x8B
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_10_IRQ_10		RW	0x8A

Table 18-520. CTRL_CORE_DMA_DSP1_DREQ_12_13

Address Offset	0x0000 0D10		
Physical Address	0x4A00 2D10	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP1_DREQ_13_IRQ_13	RESERVED	DMA_DSP1_DREQ_12_IRQ_12

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_13_IRQ_13		RW	0x8D
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_DSP1_DREQ_12_IRQ_12		RW	0x8C

Table 18-521. CTRL_CORE_DMA_DSP1_DREQ_14_15

Address Offset	0x0000 0D14		
Physical Address	0x4A00 2D14	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_DSP1_DREQ_15_IRQ_15	RESERVED	DMA_DSP1_DREQ_14_IRQ_14

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_15_IRQ_15		RW	0x8F
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_14_IRQ_14		RW	0x8E

Table 18-522. CTRL_CORE_DMA_DSP1_DREQ_16_17

Address Offset	0x0000 0D18		
Physical Address	0x4A00 2D18	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_DSP1_DREQ_17_IRQ_17	RESERVED	DMA_DSP1_DREQ_16_IRQ_16

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_17_IRQ_17		RW	0x9B
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_16_IRQ_16		RW	0x9A

Table 18-523. CTRL_CORE_DMA_DSP1_DREQ_18_19

Address Offset	0x0000 0D1C		
Physical Address	0x4A00 2D1C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–24	23–16	15–8	7–0
RESERVED	DMA_DSP1_DREQ_19_IRQ_19	RESERVED	DMA_DSP1_DREQ_18_IRQ_18

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_19_IRQ_19		RW	0x9D
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_DSP1_DREQ_18_IRQ_18		RW	0x9C

Table 18-524. CTRL_CORE_DMA_DSP2_DREQ_0_1

Address Offset	0x0000 0D20		
Physical Address	0x4A00 2D20	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP2_DREQ_1_IRQ_1	RESERVED	DMA_DSP2_DREQ_0_IRQ_0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_1_IRQ_1		RW	0x81
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_0_IRQ_0		RW	0x80

Table 18-525. CTRL_CORE_DMA_DSP2_DREQ_2_3

Address Offset	0x0000 0D24		
Physical Address	0x4A00 2D24	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP2_DREQ_3_IRQ_3	RESERVED	DMA_DSP2_DREQ_2_IRQ_2

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_3_IRQ_3		RW	0x83
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_2_IRQ_2		RW	0x82

Table 18-526. CTRL_CORE_DMA_DSP2_DREQ_4_5

Address Offset	0x0000 0D28		
Physical Address	0x4A00 2D28	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP2_DREQ_5_IRQ_5	RESERVED	DMA_DSP2_DREQ_4_IRQ_4

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_5_IRQ_5		RW	0x85
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_DSP2_DREQ_4_IRQ_4		RW	0x84

Table 18-527. CTRL_CORE_DMA_DSP2_DREQ_6_7

Address Offset	0x0000 0D2C		
Physical Address	0x4A00 2D2C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP2_DREQ_7_IRQ_7	RESERVED	DMA_DSP2_DREQ_6_IRQ_6

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_7_IRQ_7		RW	0x87
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_6_IRQ_6		RW	0x86

Table 18-528. CTRL_CORE_DMA_DSP2_DREQ_8_9

Address Offset	0x0000 0D30		
Physical Address	0x4A00 2D30	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP2_DREQ_9_IRQ_9	RESERVED	DMA_DSP2_DREQ_8_IRQ_8

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_9_IRQ_9		RW	0x89
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_8_IRQ_8		RW	0x88

Table 18-529. CTRL_CORE_DMA_DSP2_DREQ_10_11

Address Offset	0x0000 0D34		
Physical Address	0x4A00 2D34	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:16	15:8	7:0
RESERVED	DMA_DSP2_DREQ_11_IRQ_11	RESERVED	DMA_DSP2_DREQ_10_IRQ_10

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_11_IRQ_11		RW	0x8B
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_DSP2_DREQ_10_IRQ_10		RW	0x8A

Table 18-530. CTRL_CORE_DMA_DSP2_DREQ_12_13

Address Offset	0x0000 0D38		
Physical Address	0x4A00 2D38	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_DSP2_DREQ_13_IRQ_13	RESERVED	DMA_DSP2_DREQ_12_IRQ_12

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_13_IRQ_13		RW	0x8D
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_12_IRQ_12		RW	0x8C

Table 18-531. CTRL_CORE_DMA_DSP2_DREQ_14_15

Address Offset	0x0000 0D3C		
Physical Address	0x4A00 2D3C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_DSP2_DREQ_15_IRQ_15	RESERVED	DMA_DSP2_DREQ_14_IRQ_14

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_15_IRQ_15		RW	0x8F
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_14_IRQ_14		RW	0x8E

Table 18-532. CTRL_CORE_DMA_DSP2_DREQ_16_17

Address Offset	0x0000 0D40		
Physical Address	0x4A00 2D40	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DMA_DSP2_DREQ_17_IRQ_17	RESERVED	DMA_DSP2_DREQ_16_IRQ_16

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_17_IRQ_17		RW	0x9B
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7:0	DMA_DSP2_DREQ_16_IRQ_16		RW	0x9A

Table 18-533. CTRL_CORE_DMA_DSP2_DREQ_18_19

Address Offset	0x0000 0D44		
Physical Address	0x4A00 2D44	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_19_IRQ_19								RESERVED								DMA_DSP2_DREQ_18_IRQ_18							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_19_IRQ_19		RW	0x9D
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_18_IRQ_18		RW	0x9C

Table 18-534. CTRL_CORE_OVS_DMARQ_IO_MUX

Address Offset	0x0000 0D4C		
Physical Address	0x4A00 2D4C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OVS_DMARQ_IO_MUX_2								OVS_DMARQ_IO_MUX_1							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	OVS_DMARQ_IO_MUX_2		RW	0x0
7:0	OVS_DMARQ_IO_MUX_1		RW	0x0

Table 18-535. CTRL_CORE_OVS_IRQ_IO_MUX

Address Offset	0x0000 0D50		
Physical Address	0x4A00 2D50	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														OVS_IRQ_IO_MUX_2									OVS_IRQ_IO_MUX_1								

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:9	OVS_IRQ_IO_MUX_2		RW	0x0
8:0	OVS_IRQ_IO_MUX_1		RW	0x0

Table 18-536. CTRL_CORE_CONTROL_PBIAS

Address Offset	0x0000 0E00		
Physical Address	0x4A00 2E00	**Instance**	CTRL_MODULE_CORE

Table 18-536. CTRL_CORE_CONTROL_PBIAS (continued)

Description	PBIASLITE control
Type	RW

31 30 29 28	27	26	25	24	23	22	21	20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	SDCARD_BIAS_PWRDNZ	SDCARD_IO_PWRDNZ	SDCARD_BIAS_HIZ_MODE	SDCARD_BIAS_SUPPLY_HI_OUT	SDCARD_BIAS_VMODE_ERROR	RESERVED	SDCARD_BIAS_VMODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	SDCARD_BIAS_PWRDNZ	PWRDNZ control to SDCARD BIAS 0x0 = This signal is used to protect SDCARD BIAS when VDDS is not stable 0x1 = SW keep this bit to 1'b1 after VDDS stabilizing	RW	0x0
26	SDCARD_IO_PWRDNZ	PWRDNZ control to SDCARD IO 0x0 = This signal is used to protect SDCARD IOs when VDDS is not stable 0x1 = SW keep this bit to 1'b1 after VDDS stabilizing	RW	0x0
25	SDCARD_BIAS_HIZ_MODE	HIZ_MODE from SDCARD PBIAS 0x0 = PBIAS in normal operation mode 0x1 = PBIAS output is in high impedance state	RW	0x0
24	SDCARD_BIAS_SUPPLY_HI_OUT	SUPPLY_HI_OUT from SDCARD PBIAS 0x0 = VDDS = 1.8V 0x1 = VDDS = 3.3V	R	0x0
23	SDCARD_BIAS_VMODE_ERROR	VMODE ERROR from SDCARD PBIAS 0x0 = VMODE level is same as SUPPLY_HI_OUT 0x1 = VMODE level is not same as SUPPLY_HI_OUT	R	0x0
22	RESERVED		R	0x0
21	SDCARD_BIAS_VMODE	VMODE control to SDCARD PBIAS 0x0 = VDDS = 1.8V 0x1 = VDDS = 3.3V	RW	0x1
20:0	RESERVED		R	0x0

Table 18-537. CTRL_CORE_CONTROL_HDMI_TX_PHY

Address Offset	0x0000 0E0C		
Physical Address	0x4A00 2E0C	**Instance**	CTRL_MODULE_CORE
Description	HDMI TX PHY control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31	30	29	28	27:0
RESERVED	HDMITXPHY_TXVALID	HDMITXPHY_ENBYPASSCLK	HDMITXPHY_PD_PULLUPDET	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	HDMITXPHY_TXVALID	0x1= Valid data on the HDMI_TXPHY input data interface , sampled on the rising edge of TMDSCLK	RW	0x0
29	HDMITXPHY_ENBYPASSCLK	0x1 = Enables the HFBYPASSCLK to be used in place of the HFBITCLK	RW	0x0
28	HDMITXPHY_PD_PULLUPDET	0x0 = Set this bit to 0x0 if RX connection is required to be detected, even when HDMI_TXPHY is powered down 0x1 = Disables the low power RX detection functionality	RW	0x1
27:0	RESERVED		R	0x0

Table 18-538. CTRL_CORE_CONTROL_USB2PHYCORE

Address Offset	0x0000 0E1C		
Physical Address	0x4A00 2E1C	**Instance**	CTRL_MODULE_CORE
Description	This register is related to the USB2_PHY1.		
Type	RW		

31	30	29	28	27	26	25	24	23:21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4:0
USB2PHY_AUTORESUME_EN	USB2PHY_DISCHGDET	USB2PHY_GPIOMODE	USB2PHY_CHG_DET_EXT_CTL	USB2PHY_RDM_PD_CHGDET_EN	USB2PHY_RDP_PU_CHGDET_EN	USB2PHY_CHG_VSRC_EN	USB2PHY_CHG_ISINK_EN	USB2PHY_CHG_DET_STATUS	USB2PHY_CHG_DET_DM_COMP	USB2PHY_CHG_DET_DP_COMP	USB2PHY_DATADET	USB2PHY_SINKONDP	USB2PHY_SRCONDM	USB2PHY_RESTARTCHGDET	USB2PHY_CHGDETDONE	USB2PHY_CHGDETECTED	USB2PHY_MCPCPUEN	USB2PHY_MCPCMODEEN	USB2PHY_RESETDONEMCLK	USB2PHY_UTMIRESETDONE	RESERVED	USB2PHY_DATAPOLARITYN	USBDPLL_FREQLOCK	USB2PHY_RESETDONETCLK	RESERVED

Bits	Field Name	Description	Type	Reset
31	USB2PHY_AUTORESUME_EN	Auto resume enable 0x0 = disable autoresume 0x1 = enable autoresume	RW	0x0
30	USB2PHY_DISCHGDET	Disable charger detect 0x0 = charger detect function enabled 0x1 = charger detect function disabled	RW	0x1
29	USB2PHY_GPIOMODE	GPIO mode 0x0 = USB mode enabled 0x1 = GPIO mode enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
28	USB2PHY_CHG_DET_EXT_CTL	Charge detect external control 0x0 = charger detect internal state machine used 0x1 = charge detect statemachine is bypassed	RW	0x0
27	USB2PHY_RDM_PD_CHGDET_EN	DM Pull down control 0x0 = PD disabled 0x1 = PD enabled	RW	0x0
26	USB2PHY_RDP_PU_CHGDET_EN	DP Pull up control 0x0 = PU disabled 0x1 = PU enabled	RW	0x0
25	USB2PHY_CHG_VSRC_EN	VSRC enable on DP line:Host charger case 0x0 = disable VSRC drive on DP 0x1 = drives VSRC 600mV on DP line	RW	0x0
24	USB2PHY_CHG_ISINK_EN	ISINK enable on DM line:Host charger case 0x0 = disable the isink on DM 0x1 = enables the ISINK (100uA) on DM line	RW	0x0
23:21	USB2PHY_CHG_DET_STATUS	Status of charger detection 0x0 = Wait state 0x1 = No contact 0x2 = PS/2 0x3 = Unknown error 0x4 = Dedicated charger 0x5 = HOST charger 0x6 = PC 0x7 = Interrupt	R	0x0
20	USB2PHY_CHG_DET_DM_COMP	Output of the comparator on DM during the resistor host detect protocol 0x0 = DM line is below 0.75V to 0.95V 0x1 = DM line is above 0.75V to 0.95V	R	0x0
19	USB2PHY_CHG_DET_DP_COMP	Output of the comparator on DP during the resistor host detect protocol 0x0 = DP line is below 0.75V to 0.95V 0x1 = DP line is above 0.75V to 0.95V	R	0x0
18	USB2PHY_DATADET	Output of the charger detect comparator 0x0 = DM line is below 0.25V to 0.4V 0x1 = DM line is above 0.25V to 0.4V	R	0x0
17	USB2PHY_SINKONDP	When '1' current sink is connected to DP instead of DM 0x0 = Default value 0x1 = enables the ISINK on DP instead of DM	RW	0x0
16	USB2PHY_SRCONDM	When '1' voltage source is connected to DP instead of DM 0x0 = Default value 0x1 = enable the VSRC on DM instead of DP	RW	0x0
15	USB2PHY_RESTARTCHGDET	restartchgdet = '1' for 1 msec cause the CD_START to reset 0x0 = Default value 0x1 = a high pulse of 1 msec causes the charger detect to restart on negative edge of restartchgdet	RW	0x0
14	USB2PHY_CHGDETDONE	Status indicates that charger detection protocol is over 0x0 = charger detection protocol is not over 0x1 = charger detection protocol is over	R	0x0

Bits	Field Name	Description	Type	Reset
13	USB2PHY_CHGDETECTED	Output of the charger detection protocol 0x0 = charger not detected 0x1 = charger detected	R	0x0
12	USB2PHY_MCPCPUEN	MCPC Pull up enable 0x0 = disable the MCPC pull up 0x1 = enable the 4.7K to10K pull up on receive line DP when datapolarityn is 0 and DM when datapolarityn is 1	RW	0x0
11	USB2PHY_MCPCMODEEN	MCPC Mode enable 0x0 = disable MCPC mode 0x1 = enable MCPC mode	RW	0x0
10	USB2PHY_RESETDONEMCLK	OCP reset status 0x0 = OCP domain is in reset 0x1 = OCP domain is out of reset	R	0x0
9	USB2PHY_UTMIRESETDONE	UTMI FSM reset status 0x0 = UTMI FSMs are in reset 0x1 = UTMI FSMs are out of reset	R	0x0
8	RESERVED		R	0x0
7	USB2PHY_DATAPOLARITYN	Data polarity 0x0 = DP functionality is on DP and DM funcationality is on DM 0x1 = DP functionality is on DM and DM functionality is on DP	RW	0x0
6	USBDPLL_FREQLOCK	Status from USB DPLL	R	0x0
5	USB2PHY_RESETDONETCLK	resetdonetclk status from USB2PHY	R	0x0
4:0	RESERVED		R	0x0

Table 18-539. CTRL_CORE_CONTROL_HDMI_1

Address Offset	0x0000 0E20		
Physical Address	0x4A00 2E20	**Instance**	CTRL_MODULE_CORE
Description	HDMI pads control 1		
Type	RW		

31	30	29	28	27	26	25-0
HDMI_DDC_SDA_GLFENB	HDMI_DDC_SDA_PULLUPRESX	HDMI_DDC_SCL_GLFENB	HDMI_DDC_SCL_PULLUPRESX	HDMI_DDC_SDA_HSMODE	HDMI_DDC_SCL_HSMODE	RESERVED

Bits	Field Name	Description	Type	Reset
31	HDMI_DDC_SDA_GLFENB	Active_high glitch free operation enable pin for hdmi_ddc_sda receiver 0x0: Disabled 0x1: Enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
30	HDMI_DDC_SDA_PULLUPRESX	Active_low internal pull_up resistor enabled for hdmi_ddc_sda 0x0: Enabled 0x1: Disabled	RW	0x0
29	HDMI_DDC_SCL_GLFENB	Active_high glitch free operation enable pin for hdmi_ddc_scl receiver 0x0: Disabled 0x1: Enabled	RW	0x0
28	HDMI_DDC_SCL_PULLUPRESX	Active_low internal pull_up resistor enabled for hdmi_ddc_scl 0x0: Enabled 0x1: Disabled	RW	0x0
27	HDMI_DDC_SDA_HSMODE	Active-high selection for I2C High-Speed mode 0x0: Disabled 0x1: Enabled	RW	0x0
26	HDMI_DDC_SCL_HSMODE	Active-high selection for I2C High-Speed mode 0x0: Disabled 0x1: Enabled	RW	0x0
25:0	RESERVED		R	0x0

Table 18-540. CTRL_CORE_CONTROL_DDRCACH1_0

Address Offset	0x0000 0E30		
Physical Address	0x4A00 2E30	**Instance**	CTRL_MODULE_CORE
Description	ddrcaCH1 control		
Type	RW		

31 30 29	28 27 26	25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6 5	4 3 2	1 0
DDRCH1_PART0_I	DDRCH1_PART0_SR	DDRCH1_PART0_WD	DDRCH1_PART5A_I	DDRCH1_PART5A_SR	DDRCH1_PART5A_WD	DDRCH1_PART5B_I	DDRCH1_PART5B_SR	DDRCH1_PART5B_WD	DDRCH1_PART6_I	DDRCH1_PART6_SR	DDRCH1_PART6_WD

Bits	Field Name	Description	Type	Reset
31:29	DDRCH1_PART0_I	PART0 Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH1_PART0_SR	PART0 Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
25:24	DDRCH1_PART0_WD	PART0 Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH1_PART5A_I	PART5A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART5A_SR	PART5A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
17:16	DDRCH1_PART5A_WD	PART5A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART5B_I	PART5B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2

Bits	Field Name	Description	Type	Reset
12:10	DDRCH1_PART5B_SR	PART5B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH1_PART5B_WD	PART5B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH1_PART6_I	PART6 Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
4:2	DDRCH1_PART6_SR	PART6 Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
1:0	DDRCH1_PART6_WD	PART6 Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 18-541. CTRL_CORE_CONTROL_DDRCACH2_0

Address Offset	0x0000 0E34		
Physical Address	0x4A00 2E34	**Instance**	CTRL_MODULE_CORE
Description	ddrcaCH2 control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRCH2_PART0_I			DDRCH2_PART0_SR			DDRCH2_PART0_WD		DDRCH2_PART5A_I			DDRCH2_PART5A_SR			DDRCH2_PART5A_WD		DDRCH2_PART5B_I			DDRCH2_PART5B_SR			DDRCH2_PART5B_WD		DDRCH2_PART6_I			DDRCH2_PART6_SR			DDRCH2_PART6_WD	

Bits	Field Name	Description	Type	Reset
31:29	DDRCH2_PART0_I	PART0 Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH2_PART0_SR	PART0 Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
25:24	DDRCH2_PART0_WD	PART0 Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH2_PART5A_I	PART5A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH2_PART5A_SR	PART5A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
17:16	DDRCH2_PART5A_WD	PART5A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH2_PART5B_I	PART5B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH2_PART5B_SR	PART5B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH2_PART5B_WD	PART5B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH2_PART6_I	PART6 Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2

Bits	Field Name	Description	Type	Reset
4:2	DDRCH2_PART6_SR	PART6 Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
1:0	DDRCH2_PART6_WD	PART6 Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 18-542. CTRL_CORE_CONTROL_DDRCH1_0

Address Offset	0x0000 0E38		
Physical Address	0x4A00 2E38	**Instance**	CTRL_MODULE_CORE
Description	DDRCH1 control 0		
Type	RW		

31 30 29	28 27 26	25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6 5	4 3 2	1 0
DDRCH1_PART1A_I	DDRCH1_PART1A_SR	DDRCH1_PART1A_WD	DDRCH1_PART1B_I	DDRCH1_PART1B_SR	DDRCH1_PART1B_WD	DDRCH1_PART2A_I	DDRCH1_PART2A_SR	DDRCH1_PART2A_WD	DDRCH1_PART2B_I	DDRCH1_PART2B_SR	DDRCH1_PART2B_WD

Bits	Field Name	Description	Type	Reset
31:29	DDRCH1_PART1A_I	PART1A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH1_PART1A_SR	PART1A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
25:24	DDRCH1_PART1A_WD	PART1A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH1_PART1B_I	PART1B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART1B_SR	PART1B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
17:16	DDRCH1_PART1B_WD	PART1B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART2A_I	PART2A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2

Bits	Field Name	Description	Type	Reset
12:10	DDRCH1_PART2A_SR	PART2A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH1_PART2A_WD	PART2A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH1_PART2B_I	PART2B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
4:2	DDRCH1_PART2B_SR	PART2B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
1:0	DDRCH1_PART2B_WD	PART2B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 18-543. CTRL_CORE_CONTROL_DDRCH1_1

Address Offset	0x0000 0E3C		
Physical Address	0x4A00 2E3C	**Instance**	CTRL_MODULE_CORE
Description	DDRCH1 control 1		

Table 18-543. CTRL_CORE_CONTROL_DDRCH1_1 (continued)

Type	RW

31 30 29	28 27 26	25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6 5	4 3 2	1 0
DDRCH1_PART3A_I	DDRCH1_PART3A_SR	DDRCH1_PART3A_WD	DDRCH1_PART3B_I	DDRCH1_PART3B_SR	DDRCH1_PART3B_WD	DDRCH1_PART4A_I	DDRCH1_PART4A_SR	DDRCH1_PART4A_WD	DDRCH1_PART4B_I	DDRCH1_PART4B_SR	DDRCH1_PART4B_WD

Bits	Field Name	Description	Type	Reset
31:29	DDRCH1_PART3A_I	PART3A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH1_PART3A_SR	PART3A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
25:24	DDRCH1_PART3A_WD	PART3A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH1_PART3B_I	PART3B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART3B_SR	PART3B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
17:16	DDRCH1_PART3B_WD	PART3B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART4A_I	PART4A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH1_PART4A_SR	PART4A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH1_PART4A_WD	PART4A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH1_PART4B_I	PART4B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2

Bits	Field Name	Description	Type	Reset
4:2	DDRCH1_PART4B_SR	PART4B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
1:0	DDRCH1_PART4B_WD	PART4B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 18-544. CTRL_CORE_CONTROL_DDRCH2_0

Address Offset	0x0000 0E40		
Physical Address	0x4A00 2E40	**Instance**	CTRL_MODULE_CORE
Description	DDRCH2 control 0		
Type	RW		

31 30 29	28 27 26	25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6 5	4 3 2	1 0
DDRCH2_PART1A_I	DDRCH2_PART1A_SR	DDRCH2_PART1A_WD	DDRCH2_PART1B_I	DDRCH2_PART1B_SR	DDRCH2_PART1B_WD	DDRCH2_PART2A_I	DDRCH2_PART2A_SR	DDRCH2_PART2A_WD	DDRCH2_PART2B_I	DDRCH2_PART2B_SR	DDRCH2_PART2B_WD

Bits	Field Name	Description	Type	Reset
31:29	DDRCH2_PART1A_I	PART1A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH2_PART1A_SR	PART1A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
25:24	DDRCH2_PART1A_WD	PART1A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH2_PART1B_I	PART1B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH2_PART1B_SR	PART1B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
17:16	DDRCH2_PART1B_WD	PART1B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH2_PART2A_I	PART2A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2

Bits	Field Name	Description	Type	Reset
12:10	DDRCH2_PART2A_SR	PART2A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH2_PART2A_WD	PART2A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH2_PART2B_I	PART2B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
4:2	DDRCH2_PART2B_SR	PART2B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
1:0	DDRCH2_PART2B_WD	PART2B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 18-545. CTRL_CORE_CONTROL_DDRCH2_1

Address Offset	0x0000 0E44		
Physical Address	0x4A00 2E44	**Instance**	CTRL_MODULE_CORE
Description	DDRCH2 control 1		

Table 18-545. CTRL_CORE_CONTROL_DDRCH2_1 (continued)

Type RW

31 30 29	28 27 26	25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6 5	4 3 2	1 0
DDRCH2_PART3A_I	DDRCH2_PART3A_SR	DDRCH2_PART3A_WD	DDRCH2_PART3B_I	DDRCH2_PART3B_SR	DDRCH2_PART3B_WD	DDRCH2_PART4A_I	DDRCH2_PART4A_SR	DDRCH2_PART4A_WD	DDRCH2_PART4B_I	DDRCH2_PART4B_SR	DDRCH2_PART4B_WD

Bits	Field Name	Description	Type	Reset
31:29	DDRCH2_PART3A_I	PART3A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH2_PART3A_SR	PART3A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
25:24	DDRCH2_PART3A_WD	PART3A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH2_PART3B_I	PART3B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH2_PART3B_SR	PART3B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
17:16	DDRCH2_PART3B_WD	PART3B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH2_PART4A_I	PART4A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH2_PART4A_SR	PART4A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH2_PART4A_WD	PART4A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH2_PART4B_I	PART4B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2

Bits	Field Name	Description	Type	Reset
4:2	DDRCH2_PART4B_SR	PART4B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
1:0	DDRCH2_PART4B_WD	PART4B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 18-546. CTRL_CORE_CONTROL_DDRCH1_2

Address Offset	0x0000 0E48		
Physical Address	0x4A00 2E48	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:24	23:21	20:18	17:16	15:13	12:10	9:8	7:0
RESERVED	DDRCH1_PART7A_I	DDRCH1_PART7A_SR	DDRCH1_PART7A_WD	DDRCH1_PART7B_I	DDRCH1_PART7B_SR	DDRCH1_PART7B_WD	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:21	DDRCH1_PART7A_I	PART7A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART7A_SR	PART7A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
17:16	DDRCH1_PART7A_WD	PART7A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART7B_I	PART7B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH1_PART7B_SR	PART7B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH1_PART7B_WD	PART7B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:0	RESERVED		R	0x0

Table 18-547. CTRL_CORE_CONTROL_DDRIO_0

Address Offset	0x0000 0E50		
Physical Address	0x4A00 2E50	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DDRCH1_VREF_DQ0_INT_CCAP0	DDRCH1_VREF_DQ0_INT_CCAP1	DDRCH1_VREF_DQ0_INT_TAP0	DDRCH1_VREF_DQ0_INT_TAP1	DDRCH1_VREF_DQ0_INT_EN	DDRCH1_VREF_DQ1_INT_CCAP0	DDRCH1_VREF_DQ1_INT_CCAP1	DDRCH1_VREF_DQ1_INT_TAP0	DDRCH1_VREF_DQ1_INT_TAP1	DDRCH1_VREF_DQ1_INT_EN	RESERVED

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	DDRCH1_VREF_DQ0_INT_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x1
18	DDRCH1_VREF_DQ0_INT_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x0
17	DDRCH1_VREF_DQ0_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x0
16	DDRCH1_VREF_DQ0_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x1
15	DDRCH1_VREF_DQ0_INT_EN	Enable 0x0: Disabled 0x1: Enabled	RW	0x1
14	DDRCH1_VREF_DQ1_INT_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x1
13	DDRCH1_VREF_DQ1_INT_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x0
12	DDRCH1_VREF_DQ1_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x0
11	DDRCH1_VREF_DQ1_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x1
10	DDRCH1_VREF_DQ1_INT_EN	Enable 0x0: Disabled 0x1: Enabled	RW	0x1
9:0	RESERVED		R	0x260

Table 18-548. CTRL_CORE_CONTROL_DDRIO_1

Address Offset	0x0000 0E54		
Physical Address	0x4A00 2E54	**Instance**	CTRL_MODULE_CORE

Table 18-548. CTRL_CORE_CONTROL_DDRIO_1 (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					DDRCH2_VREF_DQ0_INT_CCAP0	DDRCH2_VREF_DQ0_INT_CCAP1	DDRCH2_VREF_DQ0_INT_TAP0	DDRCH2_VREF_DQ0_INT_TAP1	DDRCH2_VREF_DQ0_INT_EN	DDRCH2_VREF_DQ1_INT_CCAP0	DDRCH2_VREF_DQ1_INT_CCAP1	DDRCH2_VREF_DQ1_INT_TAP0	DDRCH2_VREF_DQ1_INT_TAP1	DDRCH2_VREF_DQ1_INT_EN	RESERVED																

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	DDRCH2_VREF_DQ0_INT_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x1
25	DDRCH2_VREF_DQ0_INT_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x0
24	DDRCH2_VREF_DQ0_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x0
23	DDRCH2_VREF_DQ0_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x1
22	DDRCH2_VREF_DQ0_INT_EN	Enable 0x0: Disabled 0x1: Enabled	RW	0x1
21	DDRCH2_VREF_DQ1_INT_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x1
20	DDRCH2_VREF_DQ1_INT_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x0
19	DDRCH2_VREF_DQ1_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x0
18	DDRCH2_VREF_DQ1_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x1
17	DDRCH2_VREF_DQ1_INT_EN	Enable 0x0: Disabled 0x1: Enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
16:0	RESERVED		R	0x13000

Table 18-549. CTRL_CORE_CONTROL_HYST_1

Address Offset	0x0000 0E5C		
Physical Address	0x4A00 2E5C	**Instance**	CTRL_MODULE_CORE
Description	Register for hysteresis and impedance control of the MMC1 pads. Effective when corresponding MUXMODE field is not configured for MMC operation.		
Type	RW		

31	30:29	28:0
SDCARD_HYST	SDCARD_IC	RESERVED

Bits	Field Name	Description	Type	Reset
31	SDCARD_HYST	hysteresis control for sdcard 0x0 = Disabled 0x1 = Enabled	RW	0x1
30:29	SDCARD_IC	Drive strength control for MMC1 pads In 3.3V signaling mode: 0x0: 50 Ohms Drive Strength 0x1: 33 Ohms Drive Strength 0x2: 66 Ohms Drive Strength 0x3: Reserved In 1.8V signaling mode: 0x0: 44 Ohms Drive Strength 0x1: 33 Ohms Drive Strength 0x2: 58 Ohms Drive Strength 0x3: 100 Ohms Drive Strength	RW	0x0
28:0	RESERVED		R	0x0

Table 18-550. CTRL_CORE_CONTROL_SPARE_RW

Address Offset	0x0000 0E68		
Physical Address	0x4A00 2E68	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CORE_CONTROL_SPARE_RW	CORE_CONTROL_SPARE_RW_MMC2_LOOPBACK	CORE_CONTROL_SPARE_RW_MMC1_LOOPBACK

Bits	Field Name	Description	Type	Reset
31:2	CORE_CONTROL_SPARE_RW	Spare bits	RW	0x0
1	CORE_CONTROL_SPARE_RW_MMC2_LOOPBACK	Selects the source of loopback clock for mmc2_clk. 0x0: Loopback clock from the I/O pad is selected 0x1: Internal loopback clock is selected	RW	0x0
0	CORE_CONTROL_SPARE_RW_MMC1_LOOPBACK	Selects the source of loopback clock for mmc1_clk. 0x0: Loopback clock from the I/O pad is selected 0x1: Internal loopback clock is selected	RW	0x0

Table 18-551. CTRL_CORE_SRCOMP_NORTH_SIDE

Address Offset	0x0000 0E74		
Physical Address	0x4A00 2E74	**Instance**	CTRL_MODULE_CORE
Description	This register is related to the USB2_PHY2.		
Type	RW		

Bit	Field
31	RESERVED
30	USB2PHY_AUTORESUME_EN
29	USB2PHY_DISCHGDET
28	USB2PHY_PD
27:21	RESERVED
20	USB2PHY_CHG_DET_DM_COMP
19	USB2PHY_CHG_DET_DP_COMP
18	USB2PHY_DATADET
17	USB2PHY_CHGDETDONE
16	USB2PHY_CHGDETECTED
15	USB2PHY_RESETDONEMCLK
14	USB2PHY_UTMIRESETDONE
13	USBDPLL_FREQLOCK
12	USB2PHY_RESETDONETCLK
11	USB2PHY_GPIOMODE
10	USB2PHY_CHG_DET_EXT_CTL
9	USB2PHY_RDM_PD_CHGDET_EN
8	USB2PHY_RDP_PU_CHGDET_EN
7	USB2PHY_CHG_VSRC_EN
6	USB2PHY_CHG_ISINK_EN
5	USB2PHY_SINKONDP
4	USB2PHY_SRCONDM
3	USB2PHY_RESTARTCHGDET
2	USB2PHY_MCPCPUEN
1	USB2PHY_MCMODEEN
0	USB2PHY_DATAPOLARITYN

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	USB2PHY_AUTORESUME_EN	Auto resume enable 0x0: disable autoresume 0x1: enable autoresume	RW	0x0
29	USB2PHY_DISCHGDET	Disable charger detect 0x0: charger detect function enabled 0x1: charger detect function disabled	RW	0x1
28	USB2PHY_PD	Power down the entire USB2_PHY2 (data, common module and UTMI). 0x0: Normal operation 0x1: Power down the USB2_PHY2	RW	0x0
27:21	RESERVED		R	0x0
20	USB2PHY_CHG_DET_DM_COMP	Output of the comparator on DM during the resistor host detect protocol. 0x0: DM line is below 0.75V to 0.95V 0x1: DM line is above 0.75V to 0.95V	R	0x0
19	USB2PHY_CHG_DET_DP_COMP	Output of the comparator on DP during the resistor host detect protocol 0x0: DP line is below 0.75V to 0.95V 0x1: DP line is above 0.75V to 0.95V	R	0x0
18	USB2PHY_DATADET	Output of the charger detect comparator 0x0: DM line is below 0.25V to 0.4V 0x1: DM line is above 0.25V to 0.4V	R	0x0
17	USB2PHY_CHGDETDONE	Status indicates that charger detection protocol is over 0x0: charger detection protocol is not over 0x1: charger detection protocol is over	R	0x0
16	USB2PHY_CHGDETECTED	Output of the charger detection protocol 0x0: charger not detected 0x1: charger detected	R	0x0
15	USB2PHY_RESETDONEMCLK	OCP reset status 0x0: OCP domain is in reset 0x1: OCP domain is out of reset	R	0x0
14	USB2PHY_UTMIRESETDONE	UTMI FSM reset status 0x0: UTMI FSMs are in reset 0x1: UTMI FSMs are out of reset	R	0x0
13	USBDPLL_FREQLOCK	Status from USB DPLL	R	0x0
12	USB2PHY_RESETDONETCLK	resetdonetclk status from USB2_PHY2	R	0x0
11	USB2PHY_GPIOMODE	GPIO mode 0x0: USB mode enabled 0x1: GPIO mode enabled	RW	0x0
10	USB2PHY_CHG_DET_EXT_CTL	Charge detect external control 0x0: charger detect internal state machine used 0x1: charge detect statemachine is bypassed	RW	0x0
9	USB2PHY_RDM_PD_CHGDET_EN	DM Pull down control 0x0: PD disabled 0x1: PD enabled	RW	0x0
8	USB2PHY_RDP_PU_CHGDET_EN	DP Pull up control 0x0: PU disabled 0x1: PU enabled	RW	0x0
7	USB2PHY_CHG_VSRC_EN	VSRC enable on DP line: Host charger case 0x0: disable VSRC drive on DP 0x1: drives VSRC 600mV on DP line	RW	0x0
6	USB2PHY_CHG_ISINK_EN	ISINK enable on DM line: Host charger case 0x0: disable the ISINK on DM 0x1: enables the ISINK (100µA) on DM line	RW	0x0

Bits	Field Name	Description	Type	Reset
5	USB2PHY_SINKONDP	When '1' current sink is connected to DP instead of DM 0x0: Default value 0x1: enables the ISINK on DP instead of DM	RW	0x0
4	USB2PHY_SRCONDM	When '1' voltage source is connected to DP instead of DM 0x0: Default value 0x1: enable the VSRC on DM instead of DP	RW	0x0
3	USB2PHY_RESTARTCHGDET	restartchgdet: '1' for 1 msec cause the CD_START to reset 0x0: Default value 0x1: a high pulse of 1 msec causes the charger detect to restart on negative edge of restartchgdet	RW	0x0
2	USB2PHY_MCPCPUEN	MCPC Pull up enable 0x0: disable the MCPC pull up 0x1: enable the 4.7K to10K pull up on receive line DP when datapolarityn is 0 and DM when datapolarityn is 1	RW	0x0
1	USB2PHY_MCPCMODEEN	MCPC Mode enable 0x0: disable MCPC mode 0x1: enable MCPC mode	RW	0x0
0	USB2PHY_DATAPOLARITYN	Data polarity 0x0: DP functionality is on DP and DM funcationality is on DM 0x1: DP functionality is on DM and DM functionality is on DP	RW	0x0

Table 18-552. CTRL_CORE_SRCOMP_SOUTH_SIDE

Address Offset	0x0000 0E78		
Physical Address	0x4A00 2E78	**Instance**	CTRL_MODULE_CORE
Description	This register is related to the USB2_PHY2.		
Type	R		

31:15	14:12	11:0
RESERVED	USB2PHY_CHG_DET_STATUS	RESERVED

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:12	USB2PHY_CHG_DET_STATUS	Status of charger detection 0x0: Wait state 0x1: No contact 0x2: PS/2 0x3: Unknown error 0x4: Dedicated charger 0x5: HOST charger 0x6: PC 0x7: Interrupt	R	0x0
11:0	RESERVED		R	0x0

Table 18-553. CTRL_CORE_PAD_GPMC_AD0

Address Offset	0x0000 1400		
Physical Address	0x4A00 3400	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_AD0_WAKEUPEVENT	GPMC_AD0_WAKEUPENABLE	RESERVED	GPMC_AD0_SLEWCONTROL	GPMC_AD0_INPUTENABLE	GPMC_AD0_PULLTYPESELECT	GPMC_AD0_PULLUDENABLE	RESERVED	GPMC_AD0_MODESELECT	GPMC_AD0_DELAYMODE	GPMC_AD0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD0_MUXMODE	0x0: gpmc_ad0 0x2: vin3a_d0 0x3: vout3_d0 0xE: gpio1_6 0xF: sysboot0	RW	0xF

Table 18-554. CTRL_CORE_PAD_GPMC_AD1

Address Offset	0x0000 1404

Table 18-554. CTRL_CORE_PAD_GPMC_AD1 (continued)

Physical Address	0x4A00 3404	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD1_WAKEUPEVENT	GPMC_AD1_WAKEUPENABLE	RESERVED	GPMC_AD1_SLEWCONTROL	GPMC_AD1_INPUTENABLE	GPMC_AD1_PULLTYPESELECT	GPMC_AD1_PULLUDENABLE	RESERVED	GPMC_AD1_MODESELECT	GPMC_AD1_DELAYMODE	GPMC_AD1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_AD1_MUXMODE	0x0: gpmc_ad1 0x2: vin3a_d1 0x3: vout3_d1 0xE: gpio1_7 0xF: sysboot1	RW	0xF

Table 18-555. CTRL_CORE_PAD_GPMC_AD2

Address Offset	0x0000 1408		
Physical Address	0x4A00 3408	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD2_WAKEUPEVENT	GPMC_AD2_WAKEUPENABLE	RESERVED	GPMC_AD2_SLEWCONTROL	GPMC_AD2_INPUTENABLE	GPMC_AD2_PULLTYPESELECT	GPMC_AD2_PULLUDENABLE	RESERVED	GPMC_AD2_MODESELECT	GPMC_AD2_DELAYMODE	GPMC_AD2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_AD2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD2_MUXMODE	0x0: gpmc_ad2 0x2: vin3a_d2 0x3: vout3_d2 0xE: gpio1_8 0xF: sysboot2	RW	0xF

Table 18-556. CTRL_CORE_PAD_GPMC_AD3

Address Offset	0x0000 140C		
Physical Address	0x4A00 340C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31-26	25	24	23-20	19	18	17	16	15-9	8	7-4	3-0
RESERVED	GPMC_AD3_WAKEUPEVENT	GPMC_AD3_WAKEUPENABLE	RESERVED	GPMC_AD3_SLEWCONTROL	GPMC_AD3_INPUTENABLE	GPMC_AD3_PULLTYPESELECT	GPMC_AD3_PULLUDENABLE	RESERVED	GPMC_AD3_MODESELECT	GPMC_AD3_DELAYMODE	GPMC_AD3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	GPMC_AD3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD3_MUXMODE	0x0: gpmc_ad3 0x2: vin3a_d3 0x3: vout3_d3 0xE: gpio1_9 0xF: sysboot3	RW	0xF

Table 18-557. CTRL_CORE_PAD_GPMC_AD4

Address Offset	0x0000 1410		
Physical Address	0x4A00 3410	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	GPMC_AD4_WAKEUPEVENT
24	GPMC_AD4_WAKEUPENABLE
23:19	RESERVED
19	GPMC_AD4_SLEWCONTROL
18	GPMC_AD4_INPUTENABLE
17	GPMC_AD4_PULLTYPESELECT
16	GPMC_AD4_PULLUDENABLE
15:9	RESERVED
8	GPMC_AD4_MODESELECT
7:4	GPMC_AD4_DELAYMODE
3:0	GPMC_AD4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	RESERVED		R	0x0
19	GPMC_AD4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD4_MUXMODE	0x0: gpmc_ad4 0x2: vin3a_d4 0x3: vout3_d4 0xE: gpio1_10 0xF: sysboot4	RW	0xF

Table 18-558. CTRL_CORE_PAD_GPMC_AD5

Address Offset	0x0000 1414		
Physical Address	0x4A00 3414	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD5_WAKEUPEVENT	GPMC_AD5_WAKEUPENABLE	RESERVED	GPMC_AD5_SLEWCONTROL	GPMC_AD5_INPUTENABLE	GPMC_AD5_PULLTYPESELECT	GPMC_AD5_PULLUDENABLE	RESERVED	GPMC_AD5_MODESELECT	GPMC_AD5_DELAYMODE	GPMC_AD5_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD5_MUXMODE	0x0: gpmc_ad5 0x2: vin3a_d5 0x3: vout3_d5 0xE: gpio1_11 0xF: sysboot5	RW	0xF

Table 18-559. CTRL_CORE_PAD_GPMC_AD6

Address Offset	0x0000 1418		
Physical Address	0x4A00 3418	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_AD6_WAKEUPEVENT	GPMC_AD6_WAKEUPENABLE	RESERVED	GPMC_AD6_SLEWCONTROL	GPMC_AD6_INPUTENABLE	GPMC_AD6_PULLTYPESELECT	GPMC_AD6_PULLUDENABLE	RESERVED	GPMC_AD6_MODESELECT	GPMC_AD6_DELAYMODE	GPMC_AD6_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD6_MUXMODE	0x0: gpmc_ad6 0x2: vin3a_d6 0x3: vout3_d6 0xE: gpio1_12 0xF: sysboot6	RW	0xF

Table 18-560. CTRL_CORE_PAD_GPMC_AD7

Address Offset	0x0000 141C

Table 18-560. CTRL_CORE_PAD_GPMC_AD7 (continued)

Physical Address	0x4A00 341C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD7_WAKEUPEVENT	GPMC_AD7_WAKEUPENABLE	RESERVED	GPMC_AD7_SLEWCONTROL	GPMC_AD7_INPUTENABLE	GPMC_AD7_PULLTYPESELECT	GPMC_AD7_PULLUDENABLE	RESERVED	GPMC_AD7_MODESELECT	GPMC_AD7_DELAYMODE	GPMC_AD7_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_AD7_MUXMODE	0x0: gpmc_ad7 0x2: vin3a_d7 0x3: vout3_d7 0xE: gpio1_13 0xF: sysboot7	RW	0xF

Table 18-561. CTRL_CORE_PAD_GPMC_AD8

Address Offset	0x0000 1420		
Physical Address	0x4A00 3420	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						GPMC_AD8_WAKEUPEVENT	GPMC_AD8_WAKEUPENABLE	RESERVED				GPMC_AD8_SLEWCONTROL	GPMC_AD8_INPUTENABLE	GPMC_AD8_PULLTYPESELECT	GPMC_AD8_PULLUDENABLE	RESERVED							GPMC_AD8_MODESELECT	GPMC_AD8_DELAYMODE				GPMC_AD8_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD8_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD8_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_AD8_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD8_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD8_MUXMODE	0x0: gpmc_ad8 0x2: vin3a_d8 0x3: vout3_d8 0xE: gpio7_18 0xF: sysboot8	RW	0xF

Table 18-562. CTRL_CORE_PAD_GPMC_AD9

Address Offset	0x0000 1424		
Physical Address	0x4A00 3424	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						GPMC_AD9_WAKEUPEVENT	GPMC_AD9_WAKEUPENABLE	RESERVED				GPMC_AD9_SLEWCONTROL	GPMC_AD9_INPUTENABLE	GPMC_AD9_PULLTYPESELECT	GPMC_AD9_PULLUDENABLE	RESERVED							GPMC_AD9_MODESELECT	GPMC_AD9_DELAYMODE				GPMC_AD9_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD9_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD9_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	GPMC_AD9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD9_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD9_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD9_MUXMODE	0x0: gpmc_ad9 0x2: vin3a_d9 0x3: vout3_d9 0xE: gpio7_19 0xF: sysboot9	RW	0xF

Table 18-563. CTRL_CORE_PAD_GPMC_AD10

Address Offset	0x0000 1428		
Physical Address	0x4A00 3428	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD10_WAKEUPEVENT	GPMC_AD10_WAKEUPENABLE	RESERVED	GPMC_AD10_SLEWCONTROL	GPMC_AD10_INPUTENABLE	GPMC_AD10_PULLTYPESELECT	GPMC_AD10_PULLUDENABLE	RESERVED	GPMC_AD10_MODESELECT	GPMC_AD10_DELAYMODE	GPMC_AD10_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD10_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD10_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	RESERVED		R	0x0
19	GPMC_AD10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD10_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD10_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD10_MUXMODE	0x0: gpmc_ad10 0x2: vin3a_d10 0x3: vout3_d10 0xE: gpio7_28 0xF: sysboot10	RW	0xF

Table 18-564. CTRL_CORE_PAD_GPMC_AD11

Address Offset	0x0000 142C		
Physical Address	0x4A00 342C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD11_WAKEUPEVENT	GPMC_AD11_WAKEUPENABLE	RESERVED	GPMC_AD11_SLEWCONTROL	GPMC_AD11_INPUTENABLE	GPMC_AD11_PULLTYPESELECT	GPMC_AD11_PULLUDENABLE	RESERVED	GPMC_AD11_MODESELECT	GPMC_AD11_DELAYMODE	GPMC_AD11_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD11_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD11_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD11_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD11_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD11_MUXMODE	0x0: gpmc_ad11 0x2: vin3a_d11 0x3: vout3_d11 0xE: gpio7_29 0xF: sysboot11	RW	0xF

Table 18-565. CTRL_CORE_PAD_GPMC_AD12

Address Offset	0x0000 1430		
Physical Address	0x4A00 3430	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_AD12_WAKEUPEVENT	GPMC_AD12_WAKEUPENABLE	RESERVED	GPMC_AD12_SLEWCONTROL	GPMC_AD12_INPUTENABLE	GPMC_AD12_PULLTYPESELECT	GPMC_AD12_PULLUDENABLE	RESERVED	GPMC_AD12_MODESELECT	GPMC_AD12_DELAYMODE	GPMC_AD12_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD12_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD12_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD12_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD12_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD12_MUXMODE	0x0: gpmc_ad12 0x2: vin3a_d12 0x3: vout3_d12 0xE: gpio1_18 0xF: sysboot12	RW	0xF

Table 18-566. CTRL_CORE_PAD_GPMC_AD13

Address Offset	0x0000 1434

Table 18-566. CTRL_CORE_PAD_GPMC_AD13 (continued)

Physical Address	0x4A00 3434	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD13_WAKEUPEVENT	GPMC_AD13_WAKEUPENABLE	RESERVED	GPMC_AD13_SLEWCONTROL	GPMC_AD13_INPUTENABLE	GPMC_AD13_PULLTYPESELECT	GPMC_AD13_PULLUDENABLE	RESERVED	GPMC_AD13_MODESELECT	GPMC_AD13_DELAYMODE	GPMC_AD13_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD13_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD13_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD13_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD13_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_AD13_MUXMODE	0x0: gpmc_ad13 0x2: vin3a_d13 0x3: vout3_d13 0xE: gpio1_19 0xF: sysboot13	RW	0xF

Table 18-567. CTRL_CORE_PAD_GPMC_AD14

Address Offset	0x0000 1438		
Physical Address	0x4A00 3438	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_AD14_WAKEUPEVENT	GPMC_AD14_WAKEUPENABLE	RESERVED	GPMC_AD14_SLEWCONTROL	GPMC_AD14_INPUTENABLE	GPMC_AD14_PULLTYPESELECT	GPMC_AD14_PULLUDENABLE	RESERVED	GPMC_AD14_MODESELECT	GPMC_AD14_DELAYMODE	GPMC_AD14_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD14_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD14_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_AD14_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD14_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD14_MUXMODE	0x0: gpmc_ad14 0x2: vin3a_d14 0x3: vout3_d14 0xE: gpio1_20 0xF: sysboot14	RW	0xF

Table 18-568. CTRL_CORE_PAD_GPMC_AD15

Address Offset	0x0000 143C		
Physical Address	0x4A00 343C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	GPMC_AD15_WAKEUPEVENT
24	GPMC_AD15_WAKEUPENABLE
23:20	RESERVED
19	GPMC_AD15_SLEWCONTROL
18	GPMC_AD15_INPUTENABLE
17	GPMC_AD15_PULLTYPESELECT
16	GPMC_AD15_PULLUDENABLE
15:9	RESERVED
8	GPMC_AD15_MODESELECT
7:4	GPMC_AD15_DELAYMODE
3:0	GPMC_AD15_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_AD15_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_AD15_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_AD15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	GPMC_AD15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	GPMC_AD15_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_AD15_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_AD15_MUXMODE	0x0: gpmc_ad15 0x2: vin3a_d15 0x3: vout3_d15 0xE: gpio1_21 0xF: sysboot15	RW	0xF

Table 18-569. CTRL_CORE_PAD_GPMC_A0

Address Offset	0x0000 1440		
Physical Address	0x4A00 3440	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A0_WAKEUPEVENT	GPMC_A0_WAKEUPENABLE	RESERVED	GPMC_A0_SLEWCONTROL	GPMC_A0_INPUTENABLE	GPMC_A0_PULLTYPESELECT	GPMC_A0_PULLUDENABLE	RESERVED	GPMC_A0_MODESELECT	GPMC_A0_DELAYMODE	GPMC_A0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	GPMC_A0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A0_MUXMODE	0x0: gpmc_a0 0x2: vin3a_d16 0x3: vout3_d16 0x4: vin4a_d0 0x6: vin4b_d0 0x7: i2c4_scl 0x8: uart5_rxd 0xE: gpio7_3 0xF: Driver off	RW	0xF

Table 18-570. CTRL_CORE_PAD_GPMC_A1

Address Offset	0x0000 1444		
Physical Address	0x4A00 3444	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_A1_WAKEUPEVENT	GPMC_A1_WAKEUPENABLE	RESERVED	GPMC_A1_SLEWCONTROL	GPMC_A1_INPUTENABLE	GPMC_A1_PULLTYPESELECT	GPMC_A1_PULLUDENABLE	RESERVED	GPMC_A1_MODESELECT	GPMC_A1_DELAYMODE	GPMC_A1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_A1_MUXMODE	0x0: gpmc_a1 0x2: vin3a_d17 0x3: vout3_d17 0x4: vin4a_d1 0x6: vin4b_d1 0x7: i2c4_sda 0x8: uart5_txd 0xE: gpio7_4 0xF: Driver off	RW	0xF

Table 18-571. CTRL_CORE_PAD_GPMC_A2

Address Offset	0x0000 1448		
Physical Address	0x4A00 3448	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A2_WAKEUPEVENT	GPMC_A2_WAKEUPENABLE	RESERVED	GPMC_A2_SLEWCONTROL	GPMC_A2_INPUTENABLE	GPMC_A2_PULLTYPESELECT	GPMC_A2_PULLUDENABLE	RESERVED	GPMC_A2_MODESELECT	GPMC_A2_DELAYMODE	GPMC_A2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_A2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A2_MUXMODE	0x0: gpmc_a2 0x2: vin3a_d18 0x3: vout3_d18 0x4: vin4a_d2 0x6: vin4b_d2 0x7: uart7_rxd 0x8: uart5_ctsn 0xE: gpio7_5 0xF: Driver off	RW	0xF

Table 18-572. CTRL_CORE_PAD_GPMC_A3

Address Offset	0x0000 144C		
Physical Address	0x4A00 344C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A3_WAKEUPEVENT	GPMC_A3_WAKEUPENABLE	RESERVED	GPMC_A3_SLEWCONTROL	GPMC_A3_INPUTENABLE	GPMC_A3_PULLTYPESELECT	GPMC_A3_PULLUDENABLE	RESERVED	GPMC_A3_MODESELECT	GPMC_A3_DELAYMODE	GPMC_A3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	GPMC_A3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A3_MUXMODE	0x0: gpmc_a3 0x1: qspi1_cs2 0x2: vin3a_d19 0x3: vout3_d19 0x4: vin4a_d3 0x6: vin4b_d3 0x7: uart7_txd 0x8: uart5_rtsn 0xE: gpio7_6 0xF: Driver off	RW	0xF

Table 18-573. CTRL_CORE_PAD_GPMC_A4

Address Offset	0x0000 1450		
Physical Address	0x4A00 3450	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_A4_WAKEUPEVENT	GPMC_A4_WAKEUPENABLE	RESERVED	GPMC_A4_SLEWCONTROL	GPMC_A4_INPUTENABLE	GPMC_A4_PULLTYPESELECT	GPMC_A4_PULLUDENABLE	RESERVED	GPMC_A4_MODESELECT	GPMC_A4_DELAYMODE	GPMC_A4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_A4_MUXMODE	0x0: gpmc_a4 0x1: qspi1_cs3 0x2: vin3a_d20 0x3: vout3_d20 0x4: vin4a_d4 0x6: vin4b_d4 0x7: i2c5_scl 0x8: uart6_rxd 0xE: gpio1_26 0xF: Driver off	RW	0xF

Table 18-574. CTRL_CORE_PAD_GPMC_A5

Address Offset	0x0000 1454		
Physical Address	0x4A00 3454	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A5_WAKEUPEVENT	GPMC_A5_WAKEUPENABLE	RESERVED	GPMC_A5_SLEWCONTROL	GPMC_A5_INPUTENABLE	GPMC_A5_PULLTYPESELECT	GPMC_A5_PULLUDENABLE	RESERVED	GPMC_A5_MODESELECT	GPMC_A5_DELAYMODE	GPMC_A5_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_A5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A5_MUXMODE	0x0: gpmc_a5 0x2: vin3a_d21 0x3: vout3_d21 0x4: vin4a_d5 0x6: vin4b_d5 0x7: i2c5_sda 0x8: uart6_txd 0xE: gpio1_27 0xF: Driver off	RW	0xF

Table 18-575. CTRL_CORE_PAD_GPMC_A6

Address Offset	0x0000 1458		
Physical Address	0x4A00 3458	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A6_WAKEUPEVENT	GPMC_A6_WAKEUPENABLE	RESERVED	GPMC_A6_SLEWCONTROL	GPMC_A6_INPUTENABLE	GPMC_A6_PULLTYPESELECT	GPMC_A6_PULLUDENABLE	RESERVED	GPMC_A6_MODESELECT	GPMC_A6_DELAYMODE	GPMC_A6_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	GPMC_A6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A6_MUXMODE	0x0: gpmc_a6 0x2: vin3a_d22 0x3: vout3_d22 0x4: vin4a_d6 0x6: vin4b_d6 0x7: uart8_rxd 0x8: uart6_ctsn 0xE: gpio1_28 0xF: Driver off	RW	0xF

Table 18-576. CTRL_CORE_PAD_GPMC_A7

Address Offset	0x0000 145C		
Physical Address	0x4A00 345C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_A7_WAKEUPEVENT	GPMC_A7_WAKEUPENABLE	RESERVED	GPMC_A7_SLEWCONTROL	GPMC_A7_INPUTENABLE	GPMC_A7_PULLTYPESELECT	GPMC_A7_PULLUDENABLE	RESERVED	GPMC_A7_MODESELECT	GPMC_A7_DELAYMODE	GPMC_A7_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_A7_MUXMODE	0x0: gpmc_a7 0x2: vin3a_d23 0x3: vout3_d23 0x4: vin4a_d7 0x6: vin4b_d7 0x7: uart8_txd 0x8: uart6_rtsn 0xE: gpio1_29 0xF: Driver off	RW	0xF

Table 18-577. CTRL_CORE_PAD_GPMC_A8

Address Offset	0x0000 1460		
Physical Address	0x4A00 3460	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A8_WAKEUPEVENT	GPMC_A8_WAKEUPENABLE	RESERVED	GPMC_A8_SLEWCONTROL	GPMC_A8_INPUTENABLE	GPMC_A8_PULLTYPESELECT	GPMC_A8_PULLUDENABLE	RESERVED	GPMC_A8_MODESELECT	GPMC_A8_DELAYMODE	GPMC_A8_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A8_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A8_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_A8_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A8_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A8_MUXMODE	0x0: gpmc_a8 0x2: vin3a_hsync0 0x3: vout3_hsync 0x6: vin4b_hsync1 0x7: timer12 0x8: spi4_sclk 0xE: gpio1_30 0xF: Driver off	RW	0xF

Table 18-578. CTRL_CORE_PAD_GPMC_A9

Address Offset	0x0000 1464		
Physical Address	0x4A00 3464	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>GPMC_A9_WAKEUPEVENT</td><td>GPMC_A9_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>GPMC_A9_SLEWCONTROL</td><td>GPMC_A9_INPUTENABLE</td><td>GPMC_A9_PULLTYPESELECT</td><td>GPMC_A9_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>GPMC_A9_MODESELECT</td><td colspan="4">GPMC_A9_DELAYMODE</td><td colspan="4">GPMC_A9_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A9_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A9_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	GPMC_A9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A9_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A9_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A9_MUXMODE	0x0: gpmc_a9 0x2: vin3a_vsync0 0x3: vout3_vsync 0x6: vin4b_vsync1 0x7: timer11 0x8: spi4_d1 0xE: gpio1_31 0xF: Driver off	RW	0xF

Table 18-579. CTRL_CORE_PAD_GPMC_A10

Address Offset	0x0000 1468		
Physical Address	0x4A00 3468	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						GPMC_A10_WAKEUPEVENT	GPMC_A10_WAKEUPENABLE	RESERVED				GPMC_A10_SLEWCONTROL	GPMC_A10_INPUTENABLE	GPMC_A10_PULLTYPESELECT	GPMC_A10_PULLUDENABLE	RESERVED							GPMC_A10_MODESELECT	GPMC_A10_DELAYMODE				GPMC_A10_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A10_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A10_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A10_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A10_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A10_MUXMODE	0x0: gpmc_a10 0x2: vin3a_de0 0x3: vout3_de 0x6: vin4b_clk1 0x7: timer10 0x8: spi4_d0 0xE: gpio2_0 0xF: Driver off	RW	0xF

Table 18-580. CTRL_CORE_PAD_GPMC_A11

Address Offset	0x0000 146C		
Physical Address	0x4A00 346C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_A11_WAKEUPEVENT	GPMC_A11_WAKEUPENABLE	RESERVED	GPMC_A11_SLEWCONTROL	GPMC_A11_INPUTENABLE	GPMC_A11_PULLTYPESELECT	GPMC_A11_PULLUDENABLE	RESERVED	GPMC_A11_MODESELECT	GPMC_A11_DELAYMODE	GPMC_A11_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A11_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A11_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A11_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A11_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_A11_MUXMODE	0x0: gpmc_a11 0x2: vin3a_fld0 0x3: vout3_fld 0x4: vin4a_fld0 0x6: vin4b_de1 0x7: timer9 0x8: spi4_cs0 0xE: gpio2_1 0xF: Driver off	RW	0xF

Table 18-581. CTRL_CORE_PAD_GPMC_A12

Address Offset	0x0000 1470		
Physical Address	0x4A00 3470	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	GPMC_A12_WAKEUPEVENT	GPMC_A12_WAKEUPENABLE	RESERVED	GPMC_A12_SLEWCONTROL	GPMC_A12_INPUTENABLE	GPMC_A12_PULLTYPESELECT	GPMC_A12_PULLUDENABLE	RESERVED	GPMC_A12_MODESELECT	GPMC_A12_DELAYMODE	GPMC_A12_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A12_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A12_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_A12_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A12_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A12_MUXMODE	0x0: gpmc_a12 0x4: vin4a_clk0 0x5: gpmc_a0 0x6: vin4b_fld1 0x7: timer8 0x8: spi4_cs1 0x9: dma_evt1 0xE: gpio2_2 0xF: Driver off	RW	0xF

Table 18-582. CTRL_CORE_PAD_GPMC_A13

Address Offset	0x0000 1474		
Physical Address	0x4A00 3474	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A13_WAKEUPEVENT	GPMC_A13_WAKEUPENABLE	RESERVED	GPMC_A13_SLEWCONTROL	GPMC_A13_INPUTENABLE	GPMC_A13_PULLTYPESELECT	GPMC_A13_PULLUDENABLE	RESERVED	GPMC_A13_MODESELECT	GPMC_A13_DELAYMODE	GPMC_A13_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A13_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A13_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	GPMC_A13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A13_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A13_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A13_MUXMODE	0x0: gpmc_a13 0x1: qspi1_rtclk 0x4: vin4a_hsync0 0x7: timer7 0x8: spi4_cs2 0x9: dma_evt2 0xE: gpio2_3 0xF: Driver off	RW	0xF

Table 18-583. CTRL_CORE_PAD_GPMC_A14

Address Offset	0x0000 1478		
Physical Address	0x4A00 3478	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_A14_WAKEUPEVENT	GPMC_A14_WAKEUPENABLE	RESERVED	GPMC_A14_SLEWCONTROL	GPMC_A14_INPUTENABLE	GPMC_A14_PULLTYPESELECT	GPMC_A14_PULLUDENABLE	RESERVED	GPMC_A14_MODESELECT	GPMC_A14_DELAYMODE	GPMC_A14_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A14_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A14_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A14_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A14_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A14_MUXMODE	0x0: gpmc_a14 0x1: qspi1_d3 0x4: vin4a_vsync0 0x7: timer6 0x8: spi4_cs3 0xE: gpio2_4 0xF: Driver off	RW	0xF

Table 18-584. CTRL_CORE_PAD_GPMC_A15

Address Offset	0x0000 147C		
Physical Address	0x4A00 347C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A15_WAKEUPEVENT	GPMC_A15_WAKEUPENABLE	RESERVED	GPMC_A15_SLEWCONTROL	GPMC_A15_INPUTENABLE	GPMC_A15_PULLTYPESELECT	GPMC_A15_PULLUDENABLE	RESERVED	GPMC_A15_MODESELECT	GPMC_A15_DELAYMODE	GPMC_A15_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A15_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A15_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A15_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A15_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_A15_MUXMODE	0x0: gpmc_a15 0x1: qspi1_d2 0x4: vin4a_d8 0x7: timer5 0xE: gpio2_5 0xF: Driver off	RW	0xF

Table 18-585. CTRL_CORE_PAD_GPMC_A16

Address Offset	0x0000 1480		
Physical Address	0x4A00 3480	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A16_WAKEUPEVENT	GPMC_A16_WAKEUPENABLE	RESERVED	GPMC_A16_SLEWCONTROL	GPMC_A16_INPUTENABLE	GPMC_A16_PULLTYPESELECT	GPMC_A16_PULLUDENABLE	RESERVED	GPMC_A16_MODESELECT	GPMC_A16_DELAYMODE	GPMC_A16_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A16_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A16_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A16_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A16_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A16_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A16_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_A16_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A16_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A16_MUXMODE	0x0: gpmc_a16 0x1: qspi1_d0 0x4: vin4a_d9 0xE: gpio2_6 0xF: Driver off	RW	0xF

Table 18-586. CTRL_CORE_PAD_GPMC_A17

Address Offset	0x0000 1484		
Physical Address	0x4A00 3484	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	GPMC_A17_WAKEUPEVENT
24	GPMC_A17_WAKEUPENABLE
23:20	RESERVED
19	GPMC_A17_SLEWCONTROL
18	GPMC_A17_INPUTENABLE
17	GPMC_A17_PULLTYPESELECT
16	GPMC_A17_PULLUDENABLE
15:9	RESERVED
8	GPMC_A17_MODESELECT
7:4	GPMC_A17_DELAYMODE
3:0	GPMC_A17_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A17_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A17_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A17_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A17_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	GPMC_A17_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A17_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A17_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A17_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A17_MUXMODE	0x0: gpmc_a17 0x1: qspi1_d1 0x4: vin4a_d10 0xE: gpio2_7 0xF: Driver off	RW	0xF

Table 18-587. CTRL_CORE_PAD_GPMC_A18

Address Offset	0x0000 1488		
Physical Address	0x4A00 3488	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	GPMC_A18_WAKEUPEVENT
24	GPMC_A18_WAKEUPENABLE
23:20	RESERVED
19	GPMC_A18_SLEWCONTROL
18	GPMC_A18_INPUTENABLE
17	GPMC_A18_PULLTYPESELECT
16	GPMC_A18_PULLUDENABLE
15:9	RESERVED
8	GPMC_A18_MODESELECT
7:4	GPMC_A18_DELAYMODE
3:0	GPMC_A18_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A18_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A18_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	RESERVED		R	0x0
19	GPMC_A18_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A18_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A18_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A18_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A18_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A18_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A18_MUXMODE	0x0: gpmc_a18 0x1: qspi1_sclk 0x4: vin4a_d11 0xE: gpio2_8 0xF: Driver off	RW	0xF

Table 18-588. CTRL_CORE_PAD_GPMC_A19

Address Offset	0x0000 148C		
Physical Address	0x4A00 348C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						GPMC_A19_WAKEUPEVENT	GPMC_A19_WAKEUPENABLE	RESERVED				GPMC_A19_SLEWCONTROL	GPMC_A19_INPUTENABLE	GPMC_A19_PULLTYPESELECT	GPMC_A19_PULLUDENABLE	RESERVED							GPMC_A19_MODESELECT	GPMC_A19_DELAYMODE				GPMC_A19_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A19_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A19_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A19_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A19_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A19_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A19_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A19_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A19_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A19_MUXMODE	0x0: gpmc_a19 0x1: mmc2_dat4 0x2: gpmc_a13 0x4: vin4a_d12 0x6: vin3b_d0 0xE: gpio2_9 0xF: Driver off	RW	0xF

Table 18-589. CTRL_CORE_PAD_GPMC_A20

Address Offset	0x0000 1490		
Physical Address	0x4A00 3490	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_A20_WAKEUPEVENT	GPMC_A20_WAKEUPENABLE	RESERVED	GPMC_A20_SLEWCONTROL	GPMC_A20_INPUTENABLE	GPMC_A20_PULLTYPESELECT	GPMC_A20_PULLUDENABLE	RESERVED	GPMC_A20_MODESELECT	GPMC_A20_DELAYMODE	GPMC_A20_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A20_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A20_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A20_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A20_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A20_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A20_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A20_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A20_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A20_MUXMODE	0x0: gpmc_a20 0x1: mmc2_dat5 0x2: gpmc_a14 0x4: vin4a_d13 0x6: vin3b_d1 0xE: gpio2_10 0xF: Driver off	RW	0xF

Table 18-590. CTRL_CORE_PAD_GPMC_A21

Address Offset	0x0000 1494		
Physical Address	0x4A00 3494	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A21_WAKEUPEVENT	GPMC_A21_WAKEUPENABLE	RESERVED	GPMC_A21_SLEWCONTROL	GPMC_A21_INPUTENABLE	GPMC_A21_PULLTYPESELECT	GPMC_A21_PULLUDENABLE	RESERVED	GPMC_A21_MODESELECT	GPMC_A21_DELAYMODE	GPMC_A21_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A21_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A21_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A21_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A21_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A21_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A21_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A21_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A21_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_A21_MUXMODE	0x0: gpmc_a21 0x1: mmc2_dat6 0x2: gpmc_a15 0x4: vin4a_d14 0x6: vin3b_d2 0xE: gpio2_11 0xF: Driver off	RW	0xF

Table 18-591. CTRL_CORE_PAD_GPMC_A22

Address Offset	0x0000 1498		
Physical Address	0x4A00 3498	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A22_WAKEUPEVENT	GPMC_A22_WAKEUPENABLE	RESERVED	GPMC_A22_SLEWCONTROL	GPMC_A22_INPUTENABLE	GPMC_A22_PULLTYPESELECT	GPMC_A22_PULLUDENABLE	RESERVED	GPMC_A22_MODESELECT	GPMC_A22_DELAYMODE	GPMC_A22_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A22_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A22_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A22_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A22_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A22_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A22_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_A22_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A22_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A22_MUXMODE	0x0: gpmc_a22 0x1: mmc2_dat7 0x2: gpmc_a16 0x4: vin4a_d15 0x6: vin3b_d3 0xE: gpio2_12 0xF: Driver off	RW	0xF

Table 18-592. CTRL_CORE_PAD_GPMC_A23

Address Offset	0x0000 149C		
Physical Address	0x4A00 349C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A23_WAKEUPEVENT	GPMC_A23_WAKEUPENABLE	RESERVED	GPMC_A23_SLEWCONTROL	GPMC_A23_INPUTENABLE	GPMC_A23_PULLTYPESELECT	GPMC_A23_PULLUDENABLE	RESERVED	GPMC_A23_MODESELECT	GPMC_A23_DELAYMODE	GPMC_A23_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A23_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A23_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A23_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	GPMC_A23_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A23_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A23_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A23_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A23_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A23_MUXMODE	0x0: gpmc_a23 0x1: mmc2_clk 0x2: gpmc_a17 0x4: vin4a_fld0 0x6: vin3b_d4 0xE: gpio2_13 0xF: Driver off	RW	0xF

Table 18-593. CTRL_CORE_PAD_GPMC_A24

Address Offset	0x0000 14A0		
Physical Address	0x4A00 34A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A24_WAKEUPEVENT	GPMC_A24_WAKEUPENABLE	RESERVED	GPMC_A24_SLEWCONTROL	GPMC_A24_INPUTENABLE	GPMC_A24_PULLTYPESELECT	GPMC_A24_PULLUDENABLE	RESERVED	GPMC_A24_MODESELECT	GPMC_A24_DELAYMODE	GPMC_A24_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	GPMC_A24_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A24_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A24_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A24_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A24_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A24_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A24_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A24_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A24_MUXMODE	0x0: gpmc_a24 0x1: mmc2_dat0 0x2: gpmc_a18 0x6: vin3b_d5 0xE: gpio2_14 0xF: Driver off	RW	0xF

Table 18-594. CTRL_CORE_PAD_GPMC_A25

Address Offset	0x0000 14A4		
Physical Address	0x4A00 34A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_A25_WAKEUPEVENT	GPMC_A25_WAKEUPENABLE	RESERVED	GPMC_A25_SLEWCONTROL	GPMC_A25_INPUTENABLE	GPMC_A25_PULLTYPESELECT	GPMC_A25_PULLUDENABLE	RESERVED	GPMC_A25_MODESELECT	GPMC_A25_DELAYMODE	GPMC_A25_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A25_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A25_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A25_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A25_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A25_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A25_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A25_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A25_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A25_MUXMODE	0x0: gpmc_a25 0x1: mmc2_dat1 0x2: gpmc_a19 0x6: vin3b_d6 0xE: gpio2_15 0xF: Driver off	RW	0xF

Table 18-595. CTRL_CORE_PAD_GPMC_A26

Address Offset	0x0000 14A8		
Physical Address	0x4A00 34A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_A26_WAKEUPEVENT	GPMC_A26_WAKEUPENABLE	RESERVED	GPMC_A26_SLEWCONTROL	GPMC_A26_INPUTENABLE	GPMC_A26_PULLTYPESELECT	GPMC_A26_PULLUDENABLE	RESERVED	GPMC_A26_MODESELECT	GPMC_A26_DELAYMODE	GPMC_A26_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A26_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A26_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A26_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A26_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A26_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A26_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_A26_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A26_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_A26_MUXMODE	0x0: gpmc_a26 0x1: mmc2_dat2 0x2: gpmc_a20 0x6: vin3b_d7 0xE: gpio2_16 0xF: Driver off	RW	0xF

Table 18-596. CTRL_CORE_PAD_GPMC_A27

Address Offset	0x0000 14AC		
Physical Address	0x4A00 34AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
Field	RESERVED	GPMC_A27_WAKEUPEVENT	GPMC_A27_WAKEUPENABLE	RESERVED	GPMC_A27_SLEWCONTROL	GPMC_A27_INPUTENABLE	GPMC_A27_PULLTYPESELECT	GPMC_A27_PULLUDENABLE	RESERVED	GPMC_A27_MODESELECT	GPMC_A27_DELAYMODE	GPMC_A27_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_A27_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_A27_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_A27_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_A27_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_A27_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_A27_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_A27_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_A27_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_A27_MUXMODE	0x0: gpmc_a27 0x1: mmc2_dat3 0x2: gpmc_a21 0x6: vin3b_hsync1 0xE: gpio2_17 0xF: Driver off	RW	0xF

Table 18-597. CTRL_CORE_PAD_GPMC_CS1

Address Offset	0x0000 14B0		
Physical Address	0x4A00 34B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_CS1_WAKEUPEVENT	GPMC_CS1_WAKEUPENABLE	RESERVED	GPMC_CS1_SLEWCONTROL	GPMC_CS1_INPUTENABLE	GPMC_CS1_PULLTYPESELECT	GPMC_CS1_PULLUDENABLE	RESERVED	GPMC_CS1_MODESELECT	GPMC_CS1_DELAYMODE	GPMC_CS1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_CS1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_CS1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_CS1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	GPMC_CS1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_CS1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_CS1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_CS1_MUXMODE	0x0: gpmc_cs1 0x1: mmc2_cmd 0x2: gpmc_a22 0x4: vin4a_de0 0x6: vin3b_vsync1 0xE: gpio2_18 0xF: Driver off	RW	0xF

Table 18-598. CTRL_CORE_PAD_GPMC_CS0

Address Offset	0x0000 14B4		
Physical Address	0x4A00 34B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_CS0_WAKEUPEVENT	GPMC_CS0_WAKEUPENABLE	RESERVED	GPMC_CS0_SLEWCONTROL	GPMC_CS0_INPUTENABLE	GPMC_CS0_PULLTYPESELECT	GPMC_CS0_PULLUDENABLE	RESERVED	GPMC_CS0_MODESELECT	GPMC_CS0_DELAYMODE	GPMC_CS0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	GPMC_CS0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_CS0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_CS0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_CS0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_CS0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_CS0_MUXMODE	0x0: gpmc_cs0 0xE: gpio2_19 0xF: Driver off	RW	0xF

Table 18-599. CTRL_CORE_PAD_GPMC_CS2

Address Offset	0x0000 14B8		
Physical Address	0x4A00 34B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_CS2_WAKEUPEVENT	GPMC_CS2_WAKEUPENABLE	RESERVED	GPMC_CS2_SLEWCONTROL	GPMC_CS2_INPUTENABLE	GPMC_CS2_PULLTYPESELECT	GPMC_CS2_PULLUDENABLE	RESERVED	GPMC_CS2_MODESELECT	GPMC_CS2_DELAYMODE	GPMC_CS2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_CS2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_CS2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_CS2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_CS2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_CS2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_CS2_MUXMODE	0x0: gpmc_cs2 0x1: qspi1_cs0 0xE: gpio2_20 0xF: Driver off	RW	0xF

Table 18-600. CTRL_CORE_PAD_GPMC_CS3

Address Offset	0x0000 14BC		
Physical Address	0x4A00 34BC	**Instance**	CTRL_MODULE_CORE

Table 18-600. CTRL_CORE_PAD_GPMC_CS3 (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						GPMC_CS3_WAKEUPEVENT	GPMC_CS3_WAKEUPENABLE	RESERVED				GPMC_CS3_SLEWCONTROL	GPMC_CS3_INPUTENABLE	GPMC_CS3_PULLTYPESELECT	GPMC_CS3_PULLUDENABLE	RESERVED							GPMC_CS3_MODESELECT	GPMC_CS3_DELAYMODE				GPMC_CS3_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_CS3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_CS3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_CS3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_CS3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_CS3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_CS3_MUXMODE	0x0: gpmc_cs3 0x1: qspi1_cs1 0x2: vin3a_clk0 0x3: vout3_clk 0x5: gpmc_a1 0xE: gpio2_21 0xF: Driver off	RW	0xF

Table 18-601. CTRL_CORE_PAD_GPMC_CLK

Address Offset	0x0000 14C0		
Physical Address	0x4A00 34C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_CLK_WAKEUPEVENT	GPMC_CLK_WAKEUPENABLE	RESERVED	GPMC_CLK_SLEWCONTROL	GPMC_CLK_INPUTENABLE	GPMC_CLK_PULLTYPESELECT	GPMC_CLK_PULLUDENABLE	RESERVED	GPMC_CLK_MODESELECT	GPMC_CLK_DELAYMODE	GPMC_CLK_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_CLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_CLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_CLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_CLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_CLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_CLK_MUXMODE	0x0: gpmc_clk 0x1: gpmc_cs7 0x2: clkout1 0x3: gpmc_wait1 0x4: vin4a_hsync0 0x5: vin4a_de0 0x6: vin3b_clk1 0x7: timer4 0x8: i2c3_scl 0x9: dma_evt1 0xE: gpio2_22 0xF: Driver off	RW	0xF

Table 18-602. CTRL_CORE_PAD_GPMC_ADVN_ALE

Address Offset	0x0000 14C4		
Physical Address	0x4A00 34C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_ADVN_ALE_WAKEUPEVENT	GPMC_ADVN_ALE_WAKEUPENABLE	RESERVED	GPMC_ADVN_ALE_SLEWCONTROL	GPMC_ADVN_ALE_INPUTENABLE	GPMC_ADVN_ALE_PULLTYPESELECT	GPMC_ADVN_ALE_PULLUDENABLE	RESERVED	GPMC_ADVN_ALE_MODESELECT	GPMC_ADVN_ALE_DELAYMODE	GPMC_ADVN_ALE_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_ADVN_ALE_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0

Bits	Field Name	Description	Type	Reset
24	GPMC_ADVN_ALE_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_ADVN_ALE_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_ADVN_ALE_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_ADVN_ALE_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_ADVN_ALE_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_ADVN_ALE_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_ADVN_ALE_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_ADVN_ALE_MUXMODE	0x0: gpmc_advn_ale 0x1: gpmc_cs6 0x2: clkout2 0x3: gpmc_wait1 0x4: vin4a_vsync0 0x5: gpmc_a2 0x6: gpmc_a23 0x7: timer3 0x8: i2c3_sda 0x9: dma_evt2 0xE: gpio2_23 0xF: Driver off	RW	0xF

Table 18-603. CTRL_CORE_PAD_GPMC_OEN_REN

Address Offset	0x0000 14C8		
Physical Address	0x4A00 34C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_OEN_REN_WAKEUPEVENT	GPMC_OEN_REN_WAKEUPENABLE	RESERVED	GPMC_OEN_REN_SLEWCONTROL	GPMC_OEN_REN_INPUTENABLE	GPMC_OEN_REN_PULLTYPESELECT	GPMC_OEN_REN_PULLUDENABLE	RESERVED	GPMC_OEN_REN_MODESELECT	GPMC_OEN_REN_DELAYMODE	GPMC_OEN_REN_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_OEN_REN_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_OEN_REN_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_OEN_REN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_OEN_REN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_OEN_REN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_OEN_REN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_OEN_REN_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_OEN_REN_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_OEN_REN_MUXMODE	0x0: gpmc_oen_ren 0xE: gpio2_24 0xF: Driver off	RW	0xF

Table 18-604. CTRL_CORE_PAD_GPMC_WEN

Address Offset	0x0000 14CC		
Physical Address	0x4A00 34CC	**Instance**	CTRL_MODULE_CORE

Table 18-604. CTRL_CORE_PAD_GPMC_WEN (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						GPMC_WEN_WAKEUPEVENT	GPMC_WEN_WAKEUPENABLE	RESERVED				GPMC_WEN_SLEWCONTROL	GPMC_WEN_INPUTENABLE	GPMC_WEN_PULLTYPESELECT	GPMC_WEN_PULLUDENABLE	RESERVED							GPMC_WEN_MODESELECT	GPMC_WEN_DELAYMODE				GPMC_WEN_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_WEN_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_WEN_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_WEN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_WEN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_WEN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_WEN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_WEN_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_WEN_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_WEN_MUXMODE	0x0: gpmc_wen 0xE: gpio2_25 0xF: Driver off	RW	0xF

Table 18-605. CTRL_CORE_PAD_GPMC_BEN0

Address Offset	0x0000 14D0		
Physical Address	0x4A00 34D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPMC_BEN0_WAKEUPEVENT	GPMC_BEN0_WAKEUPENABLE	RESERVED	GPMC_BEN0_SLEWCONTROL	GPMC_BEN0_INPUTENABLE	GPMC_BEN0_PULLTYPESELECT	GPMC_BEN0_PULLUDENABLE	RESERVED	GPMC_BEN0_MODESELECT	GPMC_BEN0_DELAYMODE	GPMC_BEN0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_BEN0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_BEN0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_BEN0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_BEN0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_BEN0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_BEN0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPMC_BEN0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_BEN0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_BEN0_MUXMODE	0x0: gpmc_ben0 0x1: gpmc_cs4 0x3: vin1b_hsync1 0x6: vin3b_de1 0x7: timer2 0x9: dma_evt3 0xE: gpio2_26 0xF: Driver off	RW	0xF

Table 18-606. CTRL_CORE_PAD_GPMC_BEN1

Address Offset	0x0000 14D4		
Physical Address	0x4A00 34D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	GPMC_BEN1_WAKEUPEVENT	GPMC_BEN1_WAKEUPENABLE	RESERVED	GPMC_BEN1_SLEWCONTROL	GPMC_BEN1_INPUTENABLE	GPMC_BEN1_PULLTYPESELECT	GPMC_BEN1_PULLUDENABLE	RESERVED	GPMC_BEN1_MODESELECT	GPMC_BEN1_DELAYMODE	GPMC_BEN1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_BEN1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_BEN1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	GPMC_BEN1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_BEN1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_BEN1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_BEN1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_BEN1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_BEN1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_BEN1_MUXMODE	0x0: gpmc_ben1 0x1: gpmc_cs5 0x3: vin1b_de1 0x4: vin3b_clk1 0x5: gpmc_a3 0x6: vin3b_fld1 0x7: timer1 0x9: dma_evt4 0xE: gpio2_27 0xF: Driver off	RW	0xF

Table 18-607. CTRL_CORE_PAD_GPMC_WAIT0

Address Offset	0x0000 14D8		
Physical Address	0x4A00 34D8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPMC_WAIT0_WAKEUPEVENT	GPMC_WAIT0_WAKEUPENABLE	RESERVED	GPMC_WAIT0_SLEWCONTROL	GPMC_WAIT0_INPUTENABLE	GPMC_WAIT0_PULLTYPESELECT	GPMC_WAIT0_PULLUDENABLE	RESERVED	GPMC_WAIT0_MODESELECT	GPMC_WAIT0_DELAYMODE	GPMC_WAIT0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPMC_WAIT0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPMC_WAIT0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPMC_WAIT0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	GPMC_WAIT0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_WAIT0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_WAIT0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPMC_WAIT0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPMC_WAIT0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPMC_WAIT0_MUXMODE	0x0: gpmc_wait0 0xE: gpio2_28 0xF: Driver off	RW	0xF

Table 18-608. CTRL_CORE_PAD_VIN1A_CLK0

Address Offset	0x0000 14DC		
Physical Address	0x4A00 34DC	**Instance**	CTRL_MODULE_CORE

Table 18-608. CTRL_CORE_PAD_VIN1A_CLK0 (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIN1A_CLK0_WAKEUPEVENT	VIN1A_CLK0_WAKEUPENABLE	RESERVED				VIN1A_CLK0_SLEWCONTROL	VIN1A_CLK0_INPUTENABLE	VIN1A_CLK0_PULLTYPESELECT	VIN1A_CLK0_PULLUDENABLE	RESERVED							VIN1A_CLK0_MODESELECT	VIN1A_CLK0_DELAYMODE				VIN1A_CLK0_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_CLK0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_CLK0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_CLK0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_CLK0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_CLK0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_CLK0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_CLK0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_CLK0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_CLK0_MUXMODE	0x0: vin1a_clk0 0x3: vout3_d16 0x4: vout3_fld 0xE: gpio2_30 0xF: Driver off	RW	0xF

Table 18-609. CTRL_CORE_PAD_VIN1B_CLK1

Address Offset	0x0000 14E0		
Physical Address	0x4A00 34E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31-26	25	24	23-20	19	18	17	16	15-9	8	7-4	3-0
RESERVED	VIN1B_CLK1_WAKEUPEVENT	VIN1B_CLK1_WAKEUPENABLE	RESERVED	VIN1B_CLK1_SLEWCONTROL	VIN1B_CLK1_INPUTENABLE	VIN1B_CLK1_PULLTYPESELECT	VIN1B_CLK1_PULLUDENABLE	RESERVED	VIN1B_CLK1_MODESELECT	VIN1B_CLK1_DELAYMODE	VIN1B_CLK1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1B_CLK1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1B_CLK1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1B_CLK1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	VIN1B_CLK1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1B_CLK1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1B_CLK1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN1B_CLK1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1B_CLK1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1B_CLK1_MUXMODE	0x0: vin1b_clk1 0x6: vin3a_clk0 0xE: gpio2_31 0xF: Driver off	RW	0xF

Table 18-610. CTRL_CORE_PAD_VIN1A_DE0

Address Offset	0x0000 14E4		
Physical Address	0x4A00 34E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_DE0_WAKEUPEVENT	VIN1A_DE0_WAKEUPENABLE	RESERVED	VIN1A_DE0_SLEWCONTROL	VIN1A_DE0_INPUTENABLE	VIN1A_DE0_PULLTYPESELECT	VIN1A_DE0_PULLUDENABLE	RESERVED	VIN1A_DE0_MODESELECT	VIN1A_DE0_DELAYMODE	VIN1A_DE0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_DE0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_DE0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_DE0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_DE0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	VIN1A_DE0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_DE0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_DE0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_DE0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_DE0_MUXMODE	0x0: vin1a_de0 0x1: vin1b_hsync1 0x3: vout3_d17 0x4: vout3_de 0x5: uart7_rxd 0x7: timer16 0x8: spi3_sclk 0x9: kbd_row0 0xA: eQEP1A_in 0xE: gpio3_0 0xF: Driver off	RW	0xF

Table 18-611. CTRL_CORE_PAD_VIN1A_FLD0

Address Offset	0x0000 14E8		
Physical Address	0x4A00 34E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>VIN1A_FLD0_WAKEUPEVENT</td><td>VIN1A_FLD0_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>VIN1A_FLD0_SLEWCONTROL</td><td>VIN1A_FLD0_INPUTENABLE</td><td>VIN1A_FLD0_PULLTYPESELECT</td><td>VIN1A_FLD0_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>VIN1A_FLD0_MODESELECT</td><td colspan="4">VIN1A_FLD0_DELAYMODE</td><td colspan="4">VIN1A_FLD0_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_FLD0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_FLD0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_FLD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_FLD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_FLD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_FLD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_FLD0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_FLD0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_FLD0_MUXMODE	0x0: vin1a_fld0 0x1: vin1b_vsync1 0x4: vout3_clk 0x5: uart7_txd 0x7: timer15 0x8: spi3_d1 0x9: kbd_row1 0xA: eQEP1B_in 0xE: gpio3_1 0xF: Driver off	RW	0xF

Table 18-612. CTRL_CORE_PAD_VIN1A_HSYNC0

Address Offset	0x0000 14EC		
Physical Address	0x4A00 34EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN1A_HSYNC0_WAKEUPEVENT	VIN1A_HSYNC0_WAKEUPENABLE	RESERVED	VIN1A_HSYNC0_SLEWCONTROL	VIN1A_HSYNC0_INPUTENABLE	VIN1A_HSYNC0_PULLTYPESELECT	VIN1A_HSYNC0_PULLUDENABLE	RESERVED	VIN1A_HSYNC0_MODESELECT	VIN1A_HSYNC0_DELAYMODE	VIN1A_HSYNC0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_HSYNC0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_HSYNC0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_HSYNC0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_HSYNC0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_HSYNC0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_HSYNC0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_HSYNC0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_HSYNC0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_HSYNC0_MUXMODE	0x0: vin1a_hsync0 0x1: vin1b_fld1 0x4: vout3_hsync 0x5: uart7_ctsn 0x7: timer14 0x8: spi3_d0 0xA: eQEP1_index 0xE: gpio3_2 0xF: Driver off	RW	0xF

Table 18-613. CTRL_CORE_PAD_VIN1A_VSYNC0

Address Offset	0x0000 14F0		
Physical Address	0x4A00 34F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIN1A_VSYNC0_WAKEUPEVENT	VIN1A_VSYNC0_WAKEUPENABLE	RESERVED				VIN1A_VSYNC0_SLEWCONTROL	VIN1A_VSYNC0_INPUTENABLE	VIN1A_VSYNC0_PULLTYPESELECT	VIN1A_VSYNC0_PULLUDENABLE	RESERVED							VIN1A_VSYNC0_MODESELECT	VIN1A_VSYNC0_DELAYMODE				VIN1A_VSYNC0_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_VSYNC0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_VSYNC0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_VSYNC0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_VSYNC0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_VSYNC0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_VSYNC0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN1A_VSYNC0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_VSYNC0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_VSYNC0_MUXMODE	0x0: vin1a_vsync0 0x1: vin1b_de1 0x4: vout3_vsync 0x5: uart7_rtsn 0x7: timer13 0x8: spi3_cs0 0xA: eQEP1_strobe 0xE: gpio3_3 0xF: Driver off	RW	0xF

Table 18-614. CTRL_CORE_PAD_VIN1A_D0

Address Offset	0x0000 14F4		
Physical Address	0x4A00 34F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D0_WAKEUPEVENT	VIN1A_D0_WAKEUPENABLE	RESERVED	VIN1A_D0_SLEWCONTROL	VIN1A_D0_INPUTENABLE	VIN1A_D0_PULLTYPESELECT	VIN1A_D0_PULLUDENABLE	RESERVED	VIN1A_D0_MODESELECT	VIN1A_D0_DELAYMODE	VIN1A_D0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VIN1A_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D0_MUXMODE	0x0: vin1a_d0 0x3: vout3_d7 0x4: vout3_d23 0x5: uart8_rxd 0xA: ehrpwm1A 0xE: gpio3_4 0xF: Driver off	RW	0xF

Table 18-615. CTRL_CORE_PAD_VIN1A_D1

Address Offset	0x0000 14F8		
Physical Address	0x4A00 34F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D1_WAKEUPEVENT	VIN1A_D1_WAKEUPENABLE	RESERVED	VIN1A_D1_SLEWCONTROL	VIN1A_D1_INPUTENABLE	VIN1A_D1_PULLTYPESELECT	VIN1A_D1_PULLUDENABLE	RESERVED	VIN1A_D1_MODESELECT	VIN1A_D1_DELAYMODE	VIN1A_D1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0

Bits	Field Name	Description	Type	Reset
24	VIN1A_D1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D1_MUXMODE	0x0: vin1a_d1 0x3: vout3_d6 0x4: vout3_d22 0x5: uart8_txd 0xA: ehrpwm1B 0xE: gpio3_5 0xF: Driver off	RW	0xF

Table 18-616. CTRL_CORE_PAD_VIN1A_D2

Address Offset	0x0000 14FC		
Physical Address	0x4A00 34FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN1A_D2_WAKEUPEVENT	VIN1A_D2_WAKEUPENABLE	RESERVED	VIN1A_D2_SLEWCONTROL	VIN1A_D2_INPUTENABLE	VIN1A_D2_PULLTYPESELECT	VIN1A_D2_PULLUDENABLE	RESERVED	VIN1A_D2_MODESELECT	VIN1A_D2_DELAYMODE	VIN1A_D2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D2_MUXMODE	0x0: vin1a_d2 0x3: vout3_d5 0x4: vout3_d21 0x5: uart8_ctsn 0xA: ehrpwm1_tripzone_input 0xE: gpio3_6 0xF: Driver off	RW	0xF

Table 18-617. CTRL_CORE_PAD_VIN1A_D3

Address Offset	0x0000 1500

Table 18-617. CTRL_CORE_PAD_VIN1A_D3 (continued)

Physical Address	0x4A00 3500	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D3_WAKEUPEVENT	VIN1A_D3_WAKEUPENABLE	RESERVED	VIN1A_D3_SLEWCONTROL	VIN1A_D3_INPUTENABLE	VIN1A_D3_PULLTYPESELECT	VIN1A_D3_PULLUDENABLE	RESERVED	VIN1A_D3_MODESELECT	VIN1A_D3_DELAYMODE	VIN1A_D3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D3_MUXMODE	0x0: vin1a_d3 0x3: vout3_d4 0x4: vout3_d20 0x5: uart8_rtsn 0xA: eCAP1_in_PWM1_out 0xE: gpio3_7 0xF: Driver off	RW	0xF

Table 18-618. CTRL_CORE_PAD_VIN1A_D4

Address Offset	0x0000 1504		
Physical Address	0x4A00 3504	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	VIN1A_D4_WAKEUPEVENT	VIN1A_D4_WAKEUPENABLE	RESERVED	VIN1A_D4_SLEWCONTROL	VIN1A_D4_INPUTENABLE	VIN1A_D4_PULLTYPESELECT	VIN1A_D4_PULLUDENABLE	RESERVED	VIN1A_D4_MODESELECT	VIN1A_D4_DELAYMODE	VIN1A_D4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN1A_D4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D4_MUXMODE	0x0: vin1a_d4 0x3: vout3_d3 0x4: vout3_d19 0xA: ehrpwm1_synci 0xE: gpio3_8 0xF: Driver off	RW	0xF

Table 18-619. CTRL_CORE_PAD_VIN1A_D5

Address Offset	0x0000 1508		
Physical Address	0x4A00 3508	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>VIN1A_D5_WAKEUPEVENT</td><td>VIN1A_D5_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>VIN1A_D5_SLEWCONTROL</td><td>VIN1A_D5_INPUTENABLE</td><td>VIN1A_D5_PULLTYPESELECT</td><td>VIN1A_D5_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>VIN1A_D5_MODESELECT</td><td colspan="4">VIN1A_D5_DELAYMODE</td><td colspan="4">VIN1A_D5_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	VIN1A_D5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D5_MUXMODE	0x0: vin1a_d5 0x3: vout3_d2 0x4: vout3_d18 0xA: ehrpwm1_synco 0xE: gpio3_9 0xF: Driver off	RW	0xF

Table 18-620. CTRL_CORE_PAD_VIN1A_D6

Address Offset	0x0000 150C		
Physical Address	0x4A00 350C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D6_WAKEUPEVENT	VIN1A_D6_WAKEUPENABLE	RESERVED	VIN1A_D6_SLEWCONTROL	VIN1A_D6_INPUTENABLE	VIN1A_D6_PULLTYPESELECT	VIN1A_D6_PULLUDENABLE	RESERVED	VIN1A_D6_MODESELECT	VIN1A_D6_DELAYMODE	VIN1A_D6_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	RESERVED		R	0x0
19	VIN1A_D6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D6_MUXMODE	0x0: vin1a_d6 0x3: vout3_d1 0x4: vout3_d17 0xA: eQEP2A_in 0xE: gpio3_10 0xF: Driver off	RW	0xF

Table 18-621. CTRL_CORE_PAD_VIN1A_D7

Address Offset	0x0000 1510		
Physical Address	0x4A00 3510	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D7_WAKEUPEVENT	VIN1A_D7_WAKEUPENABLE	RESERVED	VIN1A_D7_SLEWCONTROL	VIN1A_D7_INPUTENABLE	VIN1A_D7_PULLTYPESELECT	VIN1A_D7_PULLUDENABLE	RESERVED	VIN1A_D7_MODESELECT	VIN1A_D7_DELAYMODE	VIN1A_D7_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D7_MUXMODE	0x0: vin1a_d7 0x3: vout3_d0 0x4: vout3_d16 0xA: eQEP2B_in 0xE: gpio3_11 0xF: Driver off	RW	0xF

Table 18-622. CTRL_CORE_PAD_VIN1A_D8

Address Offset	0x0000 1514		
Physical Address	0x4A00 3514	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN1A_D8_WAKEUPEVENT	VIN1A_D8_WAKEUPENABLE	RESERVED	VIN1A_D8_SLEWCONTROL	VIN1A_D8_INPUTENABLE	VIN1A_D8_PULLTYPESELECT	VIN1A_D8_PULLUDENABLE	RESERVED	VIN1A_D8_MODESELECT	VIN1A_D8_DELAYMODE	VIN1A_D8_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D8_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D8_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D8_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D8_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D8_MUXMODE	0x0: vin1a_d8 0x1: vin1b_d7 0x4: vout3_d15 0x9: kbd_row2 0xA: eQEP2_index 0xE: gpio3_12 0xF: Driver off	RW	0xF

Table 18-623. CTRL_CORE_PAD_VIN1A_D9

Address Offset	0x0000 1518

Table 18-623. CTRL_CORE_PAD_VIN1A_D9 (continued)

Physical Address	0x4A00 3518	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D9_WAKEUPEVENT	VIN1A_D9_WAKEUPENABLE	RESERVED	VIN1A_D9_SLEWCONTROL	VIN1A_D9_INPUTENABLE	VIN1A_D9_PULLTYPESELECT	VIN1A_D9_PULLUDENABLE	RESERVED	VIN1A_D9_MODESELECT	VIN1A_D9_DELAYMODE	VIN1A_D9_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D9_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D9_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D9_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D9_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D9_MUXMODE	0x0: vin1a_d9 0x1: vin1b_d6 0x4: vout3_d14 0x9: kbd_row3 0xA: eQEP2_strobe 0xE: gpio3_13 0xF: Driver off	RW	0xF

Table 18-624. CTRL_CORE_PAD_VIN1A_D10

Address Offset	0x0000 151C		
Physical Address	0x4A00 351C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D10_WAKEUPEVENT	VIN1A_D10_WAKEUPENABLE	RESERVED	VIN1A_D10_SLEWCONTROL	VIN1A_D10_INPUTENABLE	VIN1A_D10_PULLTYPESELECT	VIN1A_D10_PULLUDENABLE	RESERVED	VIN1A_D10_MODESELECT	VIN1A_D10_DELAYMODE	VIN1A_D10_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D10_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D10_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN1A_D10_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D10_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D10_MUXMODE	0x0: vin1a_d10 0x1: vin1b_d5 0x4: vout3_d13 0x9: kbd_row4 0xE: gpio3_14 0xF: Driver off	RW	0xF

Table 18-625. CTRL_CORE_PAD_VIN1A_D11

Address Offset	0x0000 1520		
Physical Address	0x4A00 3520	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	VIN1A_D11_WAKEUPEVENT	VIN1A_D11_WAKEUPENABLE	RESERVED	VIN1A_D11_SLEWCONTROL	VIN1A_D11_INPUTENABLE	VIN1A_D11_PULLTYPESELECT	VIN1A_D11_PULLUDENABLE	RESERVED	VIN1A_D11_MODESELECT	VIN1A_D11_DELAYMODE	VIN1A_D11_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D11_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D11_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	VIN1A_D11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D11_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D11_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D11_MUXMODE	0x0: vin1a_d11 0x1: vin1b_d4 0x4: vout3_d12 0x5: gpmc_a23 0x9: kbd_row5 0xE: gpio3_15 0xF: Driver off	RW	0xF

Table 18-626. CTRL_CORE_PAD_VIN1A_D12

Address Offset	0x0000 1524		
Physical Address	0x4A00 3524	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D12_WAKEUPEVENT	VIN1A_D12_WAKEUPENABLE	RESERVED	VIN1A_D12_SLEWCONTROL	VIN1A_D12_INPUTENABLE	VIN1A_D12_PULLTYPESELECT	VIN1A_D12_PULLUDENABLE	RESERVED	VIN1A_D12_MODESELECT	VIN1A_D12_DELAYMODE	VIN1A_D12_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D12_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0

Bits	Field Name	Description	Type	Reset
24	VIN1A_D12_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D12_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D12_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D12_MUXMODE	0x0: vin1a_d12 0x1: vin1b_d3 0x2: usb3_ulpi_d7 0x4: vout3_d11 0x5: gpmc_a24 0x9: kbd_row6 0xE: gpio3_16 0xF: Driver off	RW	0xF

Table 18-627. CTRL_CORE_PAD_VIN1A_D13

Address Offset	0x0000 1528		
Physical Address	0x4A00 3528	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN1A_D13_WAKEUPEVENT	VIN1A_D13_WAKEUPENABLE	RESERVED	VIN1A_D13_SLEWCONTROL	VIN1A_D13_INPUTENABLE	VIN1A_D13_PULLTYPESELECT	VIN1A_D13_PULLUDENABLE	RESERVED	VIN1A_D13_MODESELECT	VIN1A_D13_DELAYMODE	VIN1A_D13_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D13_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D13_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D13_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D13_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D13_MUXMODE	0x0: vin1a_d13 0x1: vin1b_d2 0x2: usb3_ulpi_d6 0x4: vout3_d10 0x5: gpmc_a25 0x9: kbd_row7 0xE: gpio3_17 0xF: Driver off	RW	0xF

Table 18-628. CTRL_CORE_PAD_VIN1A_D14

Address Offset	0x0000 152C		
Physical Address	0x4A00 352C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D14_WAKEUPEVENT	VIN1A_D14_WAKEUPENABLE	RESERVED	VIN1A_D14_SLEWCONTROL	VIN1A_D14_INPUTENABLE	VIN1A_D14_PULLTYPESELECT	VIN1A_D14_PULLUDENABLE	RESERVED	VIN1A_D14_MODESELECT	VIN1A_D14_DELAYMODE	VIN1A_D14_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D14_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D14_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D14_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D14_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D14_MUXMODE	0x0: vin1a_d14 0x1: vin1b_d1 0x2: usb3_ulpi_d5 0x4: vout3_d9 0x5: gpmc_a26 0x9: kbd_row8 0xE: gpio3_18 0xF: Driver off	RW	0xF

Table 18-629. CTRL_CORE_PAD_VIN1A_D15

Address Offset	0x0000 1530		
Physical Address	0x4A00 3530	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D15_WAKEUPEVENT	VIN1A_D15_WAKEUPENABLE	RESERVED	VIN1A_D15_SLEWCONTROL	VIN1A_D15_INPUTENABLE	VIN1A_D15_PULLTYPESELECT	VIN1A_D15_PULLUDENABLE	RESERVED	VIN1A_D15_MODESELECT	VIN1A_D15_DELAYMODE	VIN1A_D15_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D15_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D15_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN1A_D15_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D15_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D15_MUXMODE	0x0: vin1a_d15 0x1: vin1b_d0 0x2: usb3_ulpi_d4 0x4: vout3_d8 0x5: gpmc_a27 0x9: kbd_col0 0xE: gpio3_19 0xF: Driver off	RW	0xF

Table 18-630. CTRL_CORE_PAD_VIN1A_D16

Address Offset	0x0000 1534		
Physical Address	0x4A00 3534	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	VIN1A_D16_WAKEUPEVENT
24	VIN1A_D16_WAKEUPENABLE
23:20	RESERVED
19	VIN1A_D16_SLEWCONTROL
18	VIN1A_D16_INPUTENABLE
17	VIN1A_D16_PULLTYPESELECT
16	VIN1A_D16_PULLUDENABLE
15:9	RESERVED
8	VIN1A_D16_MODESELECT
7:4	VIN1A_D16_DELAYMODE
3:0	VIN1A_D16_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D16_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D16_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D16_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VIN1A_D16_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D16_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D16_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D16_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D16_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D16_MUXMODE	0x0: vin1a_d16 0x1: vin1b_d7 0x2: usb3_ulpi_d3 0x4: vout3_d7 0x6: vin3a_d0 0x9: kbd_col1 0xE: gpio3_20 0xF: Driver off	RW	0xF

Table 18-631. CTRL_CORE_PAD_VIN1A_D17

Address Offset	0x0000 1538		
Physical Address	0x4A00 3538	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D17_WAKEUPEVENT	VIN1A_D17_WAKEUPENABLE	RESERVED	VIN1A_D17_SLEWCONTROL	VIN1A_D17_INPUTENABLE	VIN1A_D17_PULLTYPESELECT	VIN1A_D17_PULLUDENABLE	RESERVED	VIN1A_D17_MODESELECT	VIN1A_D17_DELAYMODE	VIN1A_D17_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	VIN1A_D17_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D17_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D17_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D17_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D17_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D17_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D17_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D17_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D17_MUXMODE	0x0: vin1a_d17 0x1: vin1b_d6 0x2: usb3_ulpi_d2 0x4: vout3_d6 0x6: vin3a_d1 0x9: kbd_col2 0xE: gpio3_21 0xF: Driver off	RW	0xF

Table 18-632. CTRL_CORE_PAD_VIN1A_D18

Address Offset	0x0000 153C		
Physical Address	0x4A00 353C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN1A_D18_WAKEUPEVENT	VIN1A_D18_WAKEUPENABLE	RESERVED	VIN1A_D18_SLEWCONTROL	VIN1A_D18_INPUTENABLE	VIN1A_D18_PULLTYPESELECT	VIN1A_D18_PULLUDENABLE	RESERVED	VIN1A_D18_MODESELECT	VIN1A_D18_DELAYMODE	VIN1A_D18_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D18_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D18_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D18_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D18_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D18_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D18_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D18_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D18_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D18_MUXMODE	0x0: vin1a_d18 0x1: vin1b_d5 0x2: usb3_ulpi_d1 0x4: vout3_d5 0x6: vin3a_d2 0x9: kbd_col3 0xE: gpio3_22 0xF: Driver off	RW	0xF

Table 18-633. CTRL_CORE_PAD_VIN1A_D19

Address Offset	0x0000 1540		
Physical Address	0x4A00 3540	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	VIN1A_D19_WAKEUPEVENT	VIN1A_D19_WAKEUPENABLE	RESERVED	VIN1A_D19_SLEWCONTROL	VIN1A_D19_INPUTENABLE	VIN1A_D19_PULLTYPESELECT	VIN1A_D19_PULLUDENABLE	RESERVED	VIN1A_D19_MODESELECT	VIN1A_D19_DELAYMODE	VIN1A_D19_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D19_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D19_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D19_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D19_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D19_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D19_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D19_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D19_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D19_MUXMODE	0x0: vin1a_d19 0x1: vin1b_d4 0x2: usb3_ulpi_d0 0x4: vout3_d4 0x6: vin3a_d3 0x9: kbd_col4 0xE: gpio3_23 0xF: Driver off	RW	0xF

Table 18-634. CTRL_CORE_PAD_VIN1A_D20

Address Offset	0x0000 1544		
Physical Address	0x4A00 3544	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D20_WAKEUPEVENT	VIN1A_D20_WAKEUPENABLE	RESERVED	VIN1A_D20_SLEWCONTROL	VIN1A_D20_INPUTENABLE	VIN1A_D20_PULLTYPESELECT	VIN1A_D20_PULLUDENABLE	RESERVED	VIN1A_D20_MODESELECT	VIN1A_D20_DELAYMODE	VIN1A_D20_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D20_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D20_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D20_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D20_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D20_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D20_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN1A_D20_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D20_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D20_MUXMODE	0x0: vin1a_d20 0x1: vin1b_d3 0x2: usb3_ulpi_nxt 0x4: vout3_d3 0x6: vin3a_d4 0x9: kbd_col5 0xE: gpio3_24 0xF: Driver off	RW	0xF

Table 18-635. CTRL_CORE_PAD_VIN1A_D21

Address Offset	0x0000 1548		
Physical Address	0x4A00 3548	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN1A_D21_WAKEUPEVENT	VIN1A_D21_WAKEUPENABLE	RESERVED	VIN1A_D21_SLEWCONTROL	VIN1A_D21_INPUTENABLE	VIN1A_D21_PULLTYPESELECT	VIN1A_D21_PULLUDENABLE	RESERVED	VIN1A_D21_MODESELECT	VIN1A_D21_DELAYMODE	VIN1A_D21_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D21_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D21_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D21_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VIN1A_D21_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D21_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D21_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D21_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D21_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D21_MUXMODE	0x0: vin1a_d21 0x1: vin1b_d2 0x2: usb3_ulpi_dir 0x4: vout3_d2 0x6: vin3a_d5 0x9: kbd_col6 0xE: gpio3_25 0xF: Driver off	RW	0xF

Table 18-636. CTRL_CORE_PAD_VIN1A_D22

Address Offset	0x0000 154C		
Physical Address	0x4A00 354C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31-26	25	24	23-20	19	18	17	16	15-9	8	7-4	3-0
RESERVED	VIN1A_D22_WAKEUPEVENT	VIN1A_D22_WAKEUPENABLE	RESERVED	VIN1A_D22_SLEWCONTROL	VIN1A_D22_INPUTENABLE	VIN1A_D22_PULLTYPESELECT	VIN1A_D22_PULLUDENABLE	RESERVED	VIN1A_D22_MODESELECT	VIN1A_D22_DELAYMODE	VIN1A_D22_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	VIN1A_D22_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D22_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D22_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D22_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D22_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D22_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D22_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D22_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D22_MUXMODE	0x0: vin1a_d22 0x1: vin1b_d1 0x2: usb3_ulpi_stp 0x4: vout3_d1 0x6: vin3a_d6 0x9: kbd_col7 0xE: gpio3_26 0xF: Driver off	RW	0xF

Table 18-637. CTRL_CORE_PAD_VIN1A_D23

Address Offset	0x0000 1550		
Physical Address	0x4A00 3550	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN1A_D23_WAKEUPEVENT	VIN1A_D23_WAKEUPENABLE	RESERVED	VIN1A_D23_SLEWCONTROL	VIN1A_D23_INPUTENABLE	VIN1A_D23_PULLTYPESELECT	VIN1A_D23_PULLUDENABLE	RESERVED	VIN1A_D23_MODESELECT	VIN1A_D23_DELAYMODE	VIN1A_D23_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN1A_D23_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN1A_D23_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN1A_D23_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D23_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D23_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D23_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN1A_D23_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN1A_D23_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN1A_D23_MUXMODE	0x0: vin1a_d23 0x1: vin1b_d0 0x2: usb3_ulpi_clk 0x4: vout3_d0 0x6: vin3a_d7 0x9: kbd_col8 0xE: gpio3_27 0xF: Driver off	RW	0xF

Table 18-638. CTRL_CORE_PAD_VIN2A_CLK0

Address Offset	0x0000 1554		
Physical Address	0x4A00 3554	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_CLK0_WAKEUPEVENT	VIN2A_CLK0_WAKEUPENABLE	RESERVED	VIN2A_CLK0_SLEWCONTROL	VIN2A_CLK0_INPUTENABLE	VIN2A_CLK0_PULLTYPESELECT	VIN2A_CLK0_PULLUDENABLE	RESERVED	VIN2A_CLK0_MODESELECT	VIN2A_CLK0_DELAYMODE	VIN2A_CLK0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_CLK0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_CLK0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_CLK0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_CLK0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_CLK0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_CLK0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_CLK0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_CLK0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN2A_CLK0_MUXMODE	0x0: vin2a_clk0 0x4: vout2_fld 0x5: emu5 0x9: kbd_row0 0xA: eQEP1A_in 0xE: gpio3_28 0xF: Driver off	RW	0xF

Table 18-639. CTRL_CORE_PAD_VIN2A_DE0

Address Offset	0x0000 1558		
Physical Address	0x4A00 3558	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_DE0_WAKEUPEVENT	VIN2A_DE0_WAKEUPENABLE	RESERVED	VIN2A_DE0_SLEWCONTROL	VIN2A_DE0_INPUTENABLE	VIN2A_DE0_PULLTYPESELECT	VIN2A_DE0_PULLUDENABLE	RESERVED	VIN2A_DE0_MODESELECT	VIN2A_DE0_DELAYMODE	VIN2A_DE0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_DE0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_DE0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_DE0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_DE0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_DE0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_DE0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN2A_DE0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_DE0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_DE0_MUXMODE	0x0: vin2a_de0 0x1: vin2a_fld0 0x2: vin2b_fld1 0x3: vin2b_de1 0x4: vout2_de 0x5: emu6 0x9: kbd_row1 0xA: eQEP1B_in 0xE: gpio3_29 0xF: Driver off	RW	0xF

Table 18-640. CTRL_CORE_PAD_VIN2A_FLD0

Address Offset	0x0000 155C		
Physical Address	0x4A00 355C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_FLD0_WAKEUPEVENT	VIN2A_FLD0_WAKEUPENABLE	RESERVED	VIN2A_FLD0_SLEWCONTROL	VIN2A_FLD0_INPUTENABLE	VIN2A_FLD0_PULLTYPESELECT	VIN2A_FLD0_PULLUDENABLE	RESERVED	VIN2A_FLD0_MODESELECT	VIN2A_FLD0_DELAYMODE	VIN2A_FLD0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_FLD0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_FLD0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VIN2A_FLD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_FLD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_FLD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_FLD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_FLD0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_FLD0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_FLD0_MUXMODE	0x0: vin2a_fld0 0x2: vin2b_clk1 0x4: vout2_clk 0x5: emu7 0xA: eQEP1_index 0xE: gpio3_30 0xF: Driver off	RW	0xF

Table 18-641. CTRL_CORE_PAD_VIN2A_HSYNC0

Address Offset	0x0000 1560		
Physical Address	0x4A00 3560	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIN2A_HSYNC0_WAKEUPEVENT	VIN2A_HSYNC0_WAKEUPENABLE	RESERVED				VIN2A_HSYNC0_SLEWCONTROL	VIN2A_HSYNC0_INPUTENABLE	VIN2A_HSYNC0_PULLTYPESELECT	VIN2A_HSYNC0_PULLUDENABLE	RESERVED							VIN2A_HSYNC0_MODESELECT	VIN2A_HSYNC0_DELAYMODE				VIN2A_HSYNC0_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_HSYNC0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_HSYNC0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_HSYNC0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_HSYNC0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_HSYNC0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_HSYNC0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_HSYNC0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_HSYNC0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_HSYNC0_MUXMODE	0x0: vin2a_hsync0 0x3: vin2b_hsync1 0x4: vout2_hsync 0x5: emu8 0x7: uart9_rxd 0x8: spi4_sclk 0x9: kbd_row2 0xA: eQEP1_strobe 0xE: gpio3_31 0xF: Driver off	RW	0xF

Table 18-642. CTRL_CORE_PAD_VIN2A_VSYNC0

Address Offset	0x0000 1564		
Physical Address	0x4A00 3564	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN2A_VSYNC0_WAKEUPEVENT	VIN2A_VSYNC0_WAKEUPENABLE	RESERVED	VIN2A_VSYNC0_SLEWCONTROL	VIN2A_VSYNC0_INPUTENABLE	VIN2A_VSYNC0_PULLTYPESELECT	VIN2A_VSYNC0_PULLUDENABLE	RESERVED	VIN2A_VSYNC0_MODESELECT	VIN2A_VSYNC0_DELAYMODE	VIN2A_VSYNC0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_VSYNC0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_VSYNC0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_VSYNC0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_VSYNC0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_VSYNC0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_VSYNC0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_VSYNC0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_VSYNC0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN2A_VSYNC0_MUXMODE	0x0: vin2a_vsync0 0x3: vin2b_vsync1 0x4: vout2_vsync 0x5: emu9 0x7: uart9_txd 0x8: spi4_d1 0x9: kbd_row3 0xA: ehrpwm1A 0xE: gpio4_0 0xF: Driver off	RW	0xF

Table 18-643. CTRL_CORE_PAD_VIN2A_D0

Address Offset	0x0000 1568		
Physical Address	0x4A00 3568	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D0_WAKEUPEVENT	VIN2A_D0_WAKEUPENABLE	RESERVED	VIN2A_D0_SLEWCONTROL	VIN2A_D0_INPUTENABLE	VIN2A_D0_PULLTYPESELECT	VIN2A_D0_PULLUDENABLE	RESERVED	VIN2A_D0_MODESELECT	VIN2A_D0_DELAYMODE	VIN2A_D0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN2A_D0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D0_MUXMODE	0x0: vin2a_d0 0x4: vout2_d23 0x5: emu10 0x7: uart9_ctsn 0x8: spi4_d0 0x9: kbd_row4 0xA: ehrpwm1B 0xE: gpio4_1 0xF: Driver off	RW	0xF

Table 18-644. CTRL_CORE_PAD_VIN2A_D1

Address Offset	0x0000 156C		
Physical Address	0x4A00 356C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>VIN2A_D1_WAKEUPEVENT</td><td>VIN2A_D1_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>VIN2A_D1_SLEWCONTROL</td><td>VIN2A_D1_INPUTENABLE</td><td>VIN2A_D1_PULLTYPESELECT</td><td>VIN2A_D1_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>VIN2A_D1_MODESELECT</td><td colspan="4">VIN2A_D1_DELAYMODE</td><td colspan="4">VIN2A_D1_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VIN2A_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D1_MUXMODE	0x0: vin2a_d1 0x4: vout2_d22 0x5: emu11 0x7: uart9_rtsn 0x8: spi4_cs0 0x9: kbd_row5 0xA: ehrpwm1_tripzone_input 0xE: gpio4_2 0xF: Driver off	RW	0xF

Table 18-645. CTRL_CORE_PAD_VIN2A_D2

Address Offset	0x0000 1570		
Physical Address	0x4A00 3570	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D2_WAKEUPEVENT	VIN2A_D2_WAKEUPENABLE	RESERVED	VIN2A_D2_SLEWCONTROL	VIN2A_D2_INPUTENABLE	VIN2A_D2_PULLTYPESELECT	VIN2A_D2_PULLUDENABLE	RESERVED	VIN2A_D2_MODESELECT	VIN2A_D2_DELAYMODE	VIN2A_D2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	VIN2A_D2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D2_MUXMODE	0x0: vin2a_d2 0x4: vout2_d21 0x5: emu12 0x8: uart10_rxd 0x9: kbd_row6 0xA: eCAP1_in_PWM1_out 0xE: gpio4_3 0xF: Driver off	RW	0xF

Table 18-646. CTRL_CORE_PAD_VIN2A_D3

Address Offset	0x0000 1574		
Physical Address	0x4A00 3574	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN2A_D3_WAKEUPEVENT	VIN2A_D3_WAKEUPENABLE	RESERVED	VIN2A_D3_SLEWCONTROL	VIN2A_D3_INPUTENABLE	VIN2A_D3_PULLTYPESELECT	VIN2A_D3_PULLUDENABLE	RESERVED	VIN2A_D3_MODESELECT	VIN2A_D3_DELAYMODE	VIN2A_D3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D3_MUXMODE	0x0: vin2a_d3 0x4: vout2_d20 0x5: emu13 0x8: uart10_txd 0x9: kbd_col0 0xA: ehrpwm1_synci 0xE: gpio4_4 0xF: Driver off	RW	0xF

Table 18-647. CTRL_CORE_PAD_VIN2A_D4

Address Offset	0x0000 1578		
Physical Address	0x4A00 3578	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D4_WAKEUPEVENT	VIN2A_D4_WAKEUPENABLE	RESERVED	VIN2A_D4_SLEWCONTROL	VIN2A_D4_INPUTENABLE	VIN2A_D4_PULLTYPESELECT	VIN2A_D4_PULLUDENABLE	RESERVED	VIN2A_D4_MODESELECT	VIN2A_D4_DELAYMODE	VIN2A_D4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN2A_D4_MUXMODE	0x0: vin2a_d4 0x4: vout2_d19 0x5: emu14 0x8: uart10_ctsn 0x9: kbd_col1 0xA: ehrpwm1_synco 0xE: gpio4_5 0xF: Driver off	RW	0xF

Table 18-648. CTRL_CORE_PAD_VIN2A_D5

Address Offset	0x0000 157C		
Physical Address	0x4A00 357C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D5_WAKEUPEVENT	VIN2A_D5_WAKEUPENABLE	RESERVED	VIN2A_D5_SLEWCONTROL	VIN2A_D5_INPUTENABLE	VIN2A_D5_PULLTYPESELECT	VIN2A_D5_PULLUDENABLE	RESERVED	VIN2A_D5_MODESELECT	VIN2A_D5_DELAYMODE	VIN2A_D5_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN2A_D5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D5_MUXMODE	0x0: vin2a_d5 0x4: vout2_d18 0x5: emu15 0x8: uart10_rtsn 0x9: kbd_col2 0xA: eQEP2A_in 0xE: gpio4_6 0xF: Driver off	RW	0xF

Table 18-649. CTRL_CORE_PAD_VIN2A_D6

Address Offset	0x0000 1580		
Physical Address	0x4A00 3580	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>VIN2A_D6_WAKEUPEVENT</td><td>VIN2A_D6_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>VIN2A_D6_SLEWCONTROL</td><td>VIN2A_D6_INPUTENABLE</td><td>VIN2A_D6_PULLTYPESELECT</td><td>VIN2A_D6_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>VIN2A_D6_MODESELECT</td><td colspan="4">VIN2A_D6_DELAYMODE</td><td colspan="4">VIN2A_D6_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VIN2A_D6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D6_MUXMODE	0x0: vin2a_d6 0x4: vout2_d17 0x5: emu16 0x8: mii1_rxd1 0x9: kbd_col3 0xA: eQEP2B_in 0xE: gpio4_7 0xF: Driver off	RW	0xF

Table 18-650. CTRL_CORE_PAD_VIN2A_D7

Address Offset	0x0000 1584		
Physical Address	0x4A00 3584	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D7_WAKEUPEVENT	VIN2A_D7_WAKEUPENABLE	RESERVED	VIN2A_D7_SLEWCONTROL	VIN2A_D7_INPUTENABLE	VIN2A_D7_PULLTYPESELECT	VIN2A_D7_PULLUDENABLE	RESERVED	VIN2A_D7_MODESELECT	VIN2A_D7_DELAYMODE	VIN2A_D7_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	VIN2A_D7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D7_MUXMODE	0x0: vin2a_d7 0x4: vout2_d16 0x5: emu17 0x8: mii1_rxd2 0x9: kbd_col4 0xA: eQEP2_index 0xE: gpio4_8 0xF: Driver off	RW	0xF

Table 18-651. CTRL_CORE_PAD_VIN2A_D8

Address Offset	0x0000 1588		
Physical Address	0x4A00 3588	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN2A_D8_WAKEUPEVENT	VIN2A_D8_WAKEUPENABLE	RESERVED	VIN2A_D8_SLEWCONTROL	VIN2A_D8_INPUTENABLE	VIN2A_D8_PULLTYPESELECT	VIN2A_D8_PULLUDENABLE	RESERVED	VIN2A_D8_MODESELECT	VIN2A_D8_DELAYMODE	VIN2A_D8_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D8_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D8_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D8_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D8_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D8_MUXMODE	0x0: vin2a_d8 0x4: vout2_d15 0x5: emu18 0x8: mii1_rxd3 0x9: kbd_col5 0xA: eQEP2_strobe 0xE: gpio4_9 0xF: Driver off	RW	0xF

Table 18-652. CTRL_CORE_PAD_VIN2A_D9

Address Offset	0x0000 158C		
Physical Address	0x4A00 358C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D9_WAKEUPEVENT	VIN2A_D9_WAKEUPENABLE	RESERVED	VIN2A_D9_SLEWCONTROL	VIN2A_D9_INPUTENABLE	VIN2A_D9_PULLTYPESELECT	VIN2A_D9_PULLUDENABLE	RESERVED	VIN2A_D9_MODESELECT	VIN2A_D9_DELAYMODE	VIN2A_D9_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D9_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D9_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D9_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D9_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN2A_D9_MUXMODE	0x0: vin2a_d9 0x4: vout2_d14 0x5: emu19 0x8: mii1_rxd0 0x9: kbd_col6 0xA: ehrpwm2A 0xE: gpio4_10 0xF: Driver off	RW	0xF

Table 18-653. CTRL_CORE_PAD_VIN2A_D10

Address Offset	0x0000 1590		
Physical Address	0x4A00 3590	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIN2A_D10_WAKEUPEVENT	VIN2A_D10_WAKEUPENABLE	RESERVED				VIN2A_D10_SLEWCONTROL	VIN2A_D10_INPUTENABLE	VIN2A_D10_PULLTYPESELECT	VIN2A_D10_PULLUDENABLE	RESERVED							VIN2A_D10_MODESELECT	VIN2A_D10_DELAYMODE				VIN2A_D10_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D10_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D10_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN2A_D10_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D10_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D10_MUXMODE	0x0: vin2a_d10 0x3: mdio_mclk 0x4: vout2_d13 0x9: kbd_col7 0xA: ehrpwm2B 0xE: gpio4_11 0xF: Driver off	RW	0xF

Table 18-654. CTRL_CORE_PAD_VIN2A_D11

Address Offset	0x0000 1594		
Physical Address	0x4A00 3594	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D11_WAKEUPEVENT	VIN2A_D11_WAKEUPENABLE	RESERVED	VIN2A_D11_SLEWCONTROL	VIN2A_D11_INPUTENABLE	VIN2A_D11_PULLTYPESELECT	VIN2A_D11_PULLUDENABLE	RESERVED	VIN2A_D11_MODESELECT	VIN2A_D11_DELAYMODE	VIN2A_D11_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D11_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D11_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VIN2A_D11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D11_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D11_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D11_MUXMODE	0x0: vin2a_d11 0x3: mdio_d 0x4: vout2_d12 0x9: kbd_row7 0xA: ehrpwm2_tripzone_input 0xE: gpio4_12 0xF: Driver off	RW	0xF

Table 18-655. CTRL_CORE_PAD_VIN2A_D12

Address Offset	0x0000 1598		
Physical Address	0x4A00 3598	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	VIN2A_D12_WAKEUPEVENT	VIN2A_D12_WAKEUPENABLE	RESERVED	VIN2A_D12_SLEWCONTROL	VIN2A_D12_INPUTENABLE	VIN2A_D12_PULLTYPESELECT	VIN2A_D12_PULLUDENABLE	RESERVED	VIN2A_D12_MODESELECT	VIN2A_D12_DELAYMODE	VIN2A_D12_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	VIN2A_D12_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D12_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D12_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D12_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D12_MUXMODE	0x0: vin2a_d12 0x3: rgmii1_txc 0x4: vout2_d11 0x8: mii1_rxclk 0x9: kbd_col8 0xA: eCAP2_in_PWM2_out 0xE: gpio4_13 0xF: Driver off	RW	0xF

Table 18-656. CTRL_CORE_PAD_VIN2A_D13

Address Offset	0x0000 159C		
Physical Address	0x4A00 359C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN2A_D13_WAKEUPEVENT	VIN2A_D13_WAKEUPENABLE	RESERVED	VIN2A_D13_SLEWCONTROL	VIN2A_D13_INPUTENABLE	VIN2A_D13_PULLTYPESELECT	VIN2A_D13_PULLUDENABLE	RESERVED	VIN2A_D13_MODESELECT	VIN2A_D13_DELAYMODE	VIN2A_D13_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D13_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D13_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D13_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D13_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D13_MUXMODE	0x0: vin2a_d13 0x3: rgmii1_txctl 0x4: vout2_d10 0x8: mii1_rxdv 0x9: kbd_row8 0xA: eQEP3A_in 0xE: gpio4_14 0xF: Driver off	RW	0xF

Table 18-657. CTRL_CORE_PAD_VIN2A_D14

Address Offset	0x0000 15A0		
Physical Address	0x4A00 35A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D14_WAKEUPEVENT	VIN2A_D14_WAKEUPENABLE	RESERVED	VIN2A_D14_SLEWCONTROL	VIN2A_D14_INPUTENABLE	VIN2A_D14_PULLTYPESELECT	VIN2A_D14_PULLUDENABLE	RESERVED	VIN2A_D14_MODESELECT	VIN2A_D14_DELAYMODE	VIN2A_D14_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D14_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D14_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D14_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D14_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN2A_D14_MUXMODE	0x0: vin2a_d14 0x3: rgmii1_txd3 0x4: vout2_d9 0x8: mii1_txclk 0xA: eQEP3B_in 0xE: gpio4_15 0xF: Driver off	RW	0xF

Table 18-658. CTRL_CORE_PAD_VIN2A_D15

Address Offset	0x0000 15A4		
Physical Address	0x4A00 35A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D15_WAKEUPEVENT	VIN2A_D15_WAKEUPENABLE	RESERVED	VIN2A_D15_SLEWCONTROL	VIN2A_D15_INPUTENABLE	VIN2A_D15_PULLTYPESELECT	VIN2A_D15_PULLUDENABLE	RESERVED	VIN2A_D15_MODESELECT	VIN2A_D15_DELAYMODE	VIN2A_D15_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D15_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D15_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN2A_D15_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D15_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D15_MUXMODE	0x0: vin2a_d15 0x3: rgmii1_txd2 0x4: vout2_d8 0x8: mii1_txd0 0xA: eQEP3_index 0xE: gpio4_16 0xF: Driver off	RW	0xF

Table 18-659. CTRL_CORE_PAD_VIN2A_D16

Address Offset	0x0000 15A8		
Physical Address	0x4A00 35A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIN2A_D16_WAKEUPEVENT	VIN2A_D16_WAKEUPENABLE	RESERVED				VIN2A_D16_SLEWCONTROL	VIN2A_D16_INPUTENABLE	VIN2A_D16_PULLTYPESELECT	VIN2A_D16_PULLUDENABLE	RESERVED							VIN2A_D16_MODESELECT	VIN2A_D16_DELAYMODE				VIN2A_D16_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D16_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D16_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D16_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VIN2A_D16_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D16_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D16_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D16_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D16_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D16_MUXMODE	0x0: vin2a_d16 0x2: vin2b_d7 0x3: rgmii1_txd1 0x4: vout2_d7 0x6: vin3a_d8 0x8: mii1_txd1 0xA: eQEP3_strobe 0xE: gpio4_24 0xF: Driver off	RW	0xF

Table 18-660. CTRL_CORE_PAD_VIN2A_D17

Address Offset	0x0000 15AC		
Physical Address	0x4A00 35AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	VIN2A_D17_WAKEUPEVENT	VIN2A_D17_WAKEUPENABLE	RESERVED	VIN2A_D17_SLEWCONTROL	VIN2A_D17_INPUTENABLE	VIN2A_D17_PULLTYPESELECT	VIN2A_D17_PULLUDENABLE	RESERVED	VIN2A_D17_MODESELECT	VIN2A_D17_DELAYMODE	VIN2A_D17_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D17_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D17_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D17_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D17_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D17_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D17_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D17_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D17_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D17_MUXMODE	0x0: vin2a_d17 0x2: vin2b_d6 0x3: rgmii1_txd0 0x4: vout2_d6 0x6: vin3a_d9 0x8: mii1_txd2 0xA: ehrpwm3A 0xE: gpio4_25 0xF: Driver off	RW	0xF

Table 18-661. CTRL_CORE_PAD_VIN2A_D18

Address Offset	0x0000 15B0		
Physical Address	0x4A00 35B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN2A_D18_WAKEUPEVENT	VIN2A_D18_WAKEUPENABLE	RESERVED	VIN2A_D18_SLEWCONTROL	VIN2A_D18_INPUTENABLE	VIN2A_D18_PULLTYPESELECT	VIN2A_D18_PULLUDENABLE	RESERVED	VIN2A_D18_MODESELECT	VIN2A_D18_DELAYMODE	VIN2A_D18_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D18_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D18_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D18_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D18_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D18_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D18_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D18_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D18_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN2A_D18_MUXMODE	0x0: vin2a_d18 0x2: vin2b_d5 0x3: rgmii1_rxc 0x4: vout2_d5 0x6: vin3a_d10 0x8: mii1_txd3 0xA: ehrpwm3B 0xE: gpio4_26 0xF: Driver off	RW	0xF

Table 18-662. CTRL_CORE_PAD_VIN2A_D19

Address Offset	0x0000 15B4		
Physical Address	0x4A00 35B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D19_WAKEUPEVENT	VIN2A_D19_WAKEUPENABLE	RESERVED	VIN2A_D19_SLEWCONTROL	VIN2A_D19_INPUTENABLE	VIN2A_D19_PULLTYPESELECT	VIN2A_D19_PULLUDENABLE	RESERVED	VIN2A_D19_MODESELECT	VIN2A_D19_DELAYMODE	VIN2A_D19_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D19_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D19_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D19_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D19_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D19_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D19_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN2A_D19_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D19_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D19_MUXMODE	0x0: vin2a_d19 0x2: vin2b_d4 0x3: rgmii1_rxctl 0x4: vout2_d4 0x6: vin3a_d11 0x8: mii1_txer 0xA: ehrpwm3_tripzone_input 0xE: gpio4_27 0xF: Driver off	RW	0xF

Table 18-663. CTRL_CORE_PAD_VIN2A_D20

Address Offset	0x0000 15B8		
Physical Address	0x4A00 35B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D20_WAKEUPEVENT	VIN2A_D20_WAKEUPENABLE	RESERVED	VIN2A_D20_SLEWCONTROL	VIN2A_D20_INPUTENABLE	VIN2A_D20_PULLTYPESELECT	VIN2A_D20_PULLUDENABLE	RESERVED	VIN2A_D20_MODESELECT	VIN2A_D20_DELAYMODE	VIN2A_D20_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D20_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D20_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VIN2A_D20_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D20_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D20_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D20_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D20_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D20_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D20_MUXMODE	0x0: vin2a_d20 0x2: vin2b_d3 0x3: rgmii1_rxd3 0x4: vout2_d3 0x5: vin3a_de0 0x6: vin3a_d12 0x8: mii1_rxer 0xA: eCAP3_in_PWM3_out 0xE: gpio4_28 0xF: Driver off	RW	0xF

Table 18-664. CTRL_CORE_PAD_VIN2A_D21

Address Offset	0x0000 15BC		
Physical Address	0x4A00 35BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VIN2A_D21_WAKEUPEVENT	VIN2A_D21_WAKEUPENABLE	RESERVED	VIN2A_D21_SLEWCONTROL	VIN2A_D21_INPUTENABLE	VIN2A_D21_PULLTYPESELECT	VIN2A_D21_PULLUDENABLE	RESERVED	VIN2A_D21_MODESELECT	VIN2A_D21_DELAYMODE	VIN2A_D21_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D21_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D21_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D21_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D21_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D21_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D21_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D21_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D21_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN2A_D21_MUXMODE	0x0: vin2a_d21 0x2: vin2b_d2 0x3: rgmii1_rxd2 0x4: vout2_d2 0x5: vin3a_fld0 0x6: vin3a_d13 0x8: mii1_col 0xE: gpio4_29 0xF: Driver off	RW	0xF

Table 18-665. CTRL_CORE_PAD_VIN2A_D22

Address Offset	0x0000 15C0		
Physical Address	0x4A00 35C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D22_WAKEUPEVENT	VIN2A_D22_WAKEUPENABLE	RESERVED	VIN2A_D22_SLEWCONTROL	VIN2A_D22_INPUTENABLE	VIN2A_D22_PULLTYPESELECT	VIN2A_D22_PULLUDENABLE	RESERVED	VIN2A_D22_MODESELECT	VIN2A_D22_DELAYMODE	VIN2A_D22_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D22_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D22_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VIN2A_D22_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D22_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D22_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D22_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VIN2A_D22_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D22_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D22_MUXMODE	0x0: vin2a_d22 0x2: vin2b_d1 0x3: rgmii1_rxd1 0x4: vout2_d1 0x5: vin3a_hsync0 0x6: vin3a_d14 0x8: mii1_crs 0xE: gpio4_30 0xF: Driver off	RW	0xF

Table 18-666. CTRL_CORE_PAD_VIN2A_D23

Address Offset	0x0000 15C4		
Physical Address	0x4A00 35C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VIN2A_D23_WAKEUPEVENT	VIN2A_D23_WAKEUPENABLE	RESERVED	VIN2A_D23_SLEWCONTROL	VIN2A_D23_INPUTENABLE	VIN2A_D23_PULLTYPESELECT	VIN2A_D23_PULLUDENABLE	RESERVED	VIN2A_D23_MODESELECT	VIN2A_D23_DELAYMODE	VIN2A_D23_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIN2A_D23_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VIN2A_D23_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VIN2A_D23_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_D23_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_D23_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_D23_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VIN2A_D23_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VIN2A_D23_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VIN2A_D23_MUXMODE	0x0: vin2a_d23 0x2: vin2b_d0 0x3: rgmii1_rxd0 0x4: vout2_d0 0x5: vin3a_vsync0 0x6: vin3a_d15 0x8: mii1_txen 0xE: gpio4_31 0xF: Driver off	RW	0xF

Table 18-667. CTRL_CORE_PAD_VOUT1_CLK

Address Offset	0x0000 15C8		
Physical Address	0x4A00 35C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_CLK_WAKEUPEVENT	VOUT1_CLK_WAKEUPENABLE	RESERVED	VOUT1_CLK_SLEWCONTROL	VOUT1_CLK_INPUTENABLE	VOUT1_CLK_PULLTYPESELECT	VOUT1_CLK_PULLUDENABLE	RESERVED	VOUT1_CLK_MODESELECT	VOUT1_CLK_DELAYMODE	VOUT1_CLK_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_CLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_CLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_CLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_CLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_CLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_CLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_CLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_CLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_CLK_MUXMODE	0x0: vout1_clk 0x3: vin4a_fld0 0x4: vin3a_fld0 0x8: spi3_cs0 0xE: gpio4_19 0xF: Driver off	RW	0xF

Table 18-668. CTRL_CORE_PAD_VOUT1_DE

Address Offset	0x0000 15CC		
Physical Address	0x4A00 35CC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_DE_WAKEUPEVENT	VOUT1_DE_WAKEUPENABLE	RESERVED	VOUT1_DE_SLEWCONTROL	VOUT1_DE_INPUTENABLE	VOUT1_DE_PULLTYPESELECT	VOUT1_DE_PULLUDENABLE	RESERVED	VOUT1_DE_MODESELECT	VOUT1_DE_DELAYMODE	VOUT1_DE_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_DE_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_DE_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_DE_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_DE_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_DE_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_DE_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_DE_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_DE_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VOUT1_DE_MUXMODE	0x0: vout1_de 0x3: vin4a_de0 0x4: vin3a_de0 0x8: spi3_d1 0xE: gpio4_20 0xF: Driver off	RW	0xF

Table 18-669. CTRL_CORE_PAD_VOUT1_FLD

Address Offset	0x0000 15D0		
Physical Address	0x4A00 35D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_FLD_WAKEUPEVENT	VOUT1_FLD_WAKEUPENABLE	RESERVED	VOUT1_FLD_SLEWCONTROL	VOUT1_FLD_INPUTENABLE	VOUT1_FLD_PULLTYPESELECT	VOUT1_FLD_PULLUDENABLE	RESERVED	VOUT1_FLD_MODESELECT	VOUT1_FLD_DELAYMODE	VOUT1_FLD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_FLD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_FLD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_FLD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_FLD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_FLD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_FLD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VOUT1_FLD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_FLD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_FLD_MUXMODE	0x0: vout1_fld 0x3: vin4a_clk0 0x4: vin3a_clk0 0x8: spi3_cs1 0xE: gpio4_21 0xF: Driver off	RW	0xF

Table 18-670. CTRL_CORE_PAD_VOUT1_HSYNC

Address Offset	0x0000 15D4		
Physical Address	0x4A00 35D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_HSYNC_WAKEUPEVENT	VOUT1_HSYNC_WAKEUPENABLE	RESERVED	VOUT1_HSYNC_SLEWCONTROL	VOUT1_HSYNC_INPUTENABLE	VOUT1_HSYNC_PULLTYPESELECT	VOUT1_HSYNC_PULLUDENABLE	RESERVED	VOUT1_HSYNC_MODESELECT	VOUT1_HSYNC_DELAYMODE	VOUT1_HSYNC_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_HSYNC_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_HSYNC_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_HSYNC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VOUT1_HSYNC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_HSYNC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_HSYNC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_HSYNC_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_HSYNC_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_HSYNC_MUXMODE	0x0: vout1_hsync 0x3: vin4a_hsync0 0x4: vin3a_hsync0 0x8: spi3_d0 0xE: gpio4_22 0xF: Driver off	RW	0xF

Table 18-671. CTRL_CORE_PAD_VOUT1_VSYNC

Address Offset	0x0000 15D8		
Physical Address	0x4A00 35D8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	VOUT1_VSYNC_WAKEUPEVENT
24	VOUT1_VSYNC_WAKEUPENABLE
23:20	RESERVED
19	VOUT1_VSYNC_SLEWCONTROL
18	VOUT1_VSYNC_INPUTENABLE
17	VOUT1_VSYNC_PULLTYPESELECT
16	VOUT1_VSYNC_PULLUDENABLE
15:9	RESERVED
8	VOUT1_VSYNC_MODESELECT
7:4	VOUT1_VSYNC_DELAYMODE
3:0	VOUT1_VSYNC_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	VOUT1_VSYNC_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_VSYNC_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_VSYNC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_VSYNC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_VSYNC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_VSYNC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_VSYNC_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_VSYNC_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_VSYNC_MUXMODE	0x0: vout1_vsync 0x3: vin4a_vsync0 0x4: vin3a_vsync0 0x8: spi3_sclk 0xE: gpio4_23 0xF: Driver off	RW	0xF

Table 18-672. CTRL_CORE_PAD_VOUT1_D0

Address Offset	0x0000 15DC		
Physical Address	0x4A00 35DC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_D0_WAKEUPEVENT	VOUT1_D0_WAKEUPENABLE	RESERVED	VOUT1_D0_SLEWCONTROL	VOUT1_D0_INPUTENABLE	VOUT1_D0_PULLTYPESELECT	VOUT1_D0_PULLUDENABLE	RESERVED	VOUT1_D0_MODESELECT	VOUT1_D0_DELAYMODE	VOUT1_D0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D0_MUXMODE	0x0: vout1_d0 0x2: uart5_rxd 0x3: vin4a_d16 0x4: vin3a_d16 0x8: spi3_cs2 0xE: gpio8_0 0xF: Driver off	RW	0xF

Table 18-673. CTRL_CORE_PAD_VOUT1_D1

Address Offset	0x0000 15E0		
Physical Address	0x4A00 35E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D1_WAKEUPEVENT	VOUT1_D1_WAKEUPENABLE	RESERVED	VOUT1_D1_SLEWCONTROL	VOUT1_D1_INPUTENABLE	VOUT1_D1_PULLTYPESELECT	VOUT1_D1_PULLUDENABLE	RESERVED	VOUT1_D1_MODESELECT	VOUT1_D1_DELAYMODE	VOUT1_D1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VOUT1_D1_MUXMODE	0x0: vout1_d1 0x2: uart5_txd 0x3: vin4a_d17 0x4: vin3a_d17 0xE: gpio8_1 0xF: Driver off	RW	0xF

Table 18-674. CTRL_CORE_PAD_VOUT1_D2

Address Offset	0x0000 15E4		
Physical Address	0x4A00 35E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D2_WAKEUPEVENT	VOUT1_D2_WAKEUPENABLE	RESERVED	VOUT1_D2_SLEWCONTROL	VOUT1_D2_INPUTENABLE	VOUT1_D2_PULLTYPESELECT	VOUT1_D2_PULLUDENABLE	RESERVED	VOUT1_D2_MODESELECT	VOUT1_D2_DELAYMODE	VOUT1_D2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VOUT1_D2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D2_MUXMODE	0x0: vout1_d2 0x2: emu2 0x3: vin4a_d18 0x4: vin3a_d18 0x5: obs0 0x6: obs16 0x7: obs_irq1 0xE: gpio8_2 0xF: Driver off	RW	0xF

Table 18-675. CTRL_CORE_PAD_VOUT1_D3

Address Offset	0x0000 15E8		
Physical Address	0x4A00 35E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
Field	RESERVED	VOUT1_D3_WAKEUPEVENT	VOUT1_D3_WAKEUPENABLE	RESERVED	VOUT1_D3_SLEWCONTROL	VOUT1_D3_INPUTENABLE	VOUT1_D3_PULLTYPESELECT	VOUT1_D3_PULLUDENABLE	RESERVED	VOUT1_D3_MODESELECT	VOUT1_D3_DELAYMODE	VOUT1_D3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VOUT1_D3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D3_MUXMODE	0x0: vout1_d3 0x2: emu5 0x3: vin4a_d19 0x4: vin3a_d19 0x5: obs1 0x6: obs17 0x7: obs_dmarq1 0xE: gpio8_3 0xF: Driver off	RW	0xF

Table 18-676. CTRL_CORE_PAD_VOUT1_D4

Address Offset	0x0000 15EC		
Physical Address	0x4A00 35EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_D4_WAKEUPEVENT	VOUT1_D4_WAKEUPENABLE	RESERVED	VOUT1_D4_SLEWCONTROL	VOUT1_D4_INPUTENABLE	VOUT1_D4_PULLTYPESELECT	VOUT1_D4_PULLUDENABLE	RESERVED	VOUT1_D4_MODESELECT	VOUT1_D4_DELAYMODE	VOUT1_D4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D4_MUXMODE	0x0: vout1_d4 0x2: emu6 0x3: vin4a_d20 0x4: vin3a_d20 0x5: obs2 0x6: obs18 0xE: gpio8_4 0xF: Driver off	RW	0xF

Table 18-677. CTRL_CORE_PAD_VOUT1_D5

Address Offset	0x0000 15F0		
Physical Address	0x4A00 35F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
Field	RESERVED	VOUT1_D5_WAKEUPEVENT	VOUT1_D5_WAKEUPENABLE	RESERVED	VOUT1_D5_SLEWCONTROL	VOUT1_D5_INPUTENABLE	VOUT1_D5_PULLTYPESELECT	VOUT1_D5_PULLUDENABLE	RESERVED	VOUT1_D5_MODESELECT	VOUT1_D5_DELAYMODE	VOUT1_D5_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VOUT1_D5_MUXMODE	0x0: vout1_d5 0x2: emu7 0x3: vin4a_d21 0x4: vin3a_d21 0x5: obs3 0x6: obs19 0xE: gpio8_5 0xF: Driver off	RW	0xF

Table 18-678. CTRL_CORE_PAD_VOUT1_D6

Address Offset	0x0000 15F4		
Physical Address	0x4A00 35F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D6_WAKEUPEVENT	VOUT1_D6_WAKEUPENABLE	RESERVED	VOUT1_D6_SLEWCONTROL	VOUT1_D6_INPUTENABLE	VOUT1_D6_PULLTYPESELECT	VOUT1_D6_PULLUDENABLE	RESERVED	VOUT1_D6_MODESELECT	VOUT1_D6_DELAYMODE	VOUT1_D6_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VOUT1_D6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D6_MUXMODE	0x0: vout1_d6 0x2: emu8 0x3: vin4a_d22 0x4: vin3a_d22 0x5: obs4 0x6: obs20 0xE: gpio8_6 0xF: Driver off	RW	0xF

Table 18-679. CTRL_CORE_PAD_VOUT1_D7

Address Offset	0x0000 15F8		
Physical Address	0x4A00 35F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>VOUT1_D7_WAKEUPEVENT</td><td>VOUT1_D7_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>VOUT1_D7_SLEWCONTROL</td><td>VOUT1_D7_INPUTENABLE</td><td>VOUT1_D7_PULLTYPESELECT</td><td>VOUT1_D7_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>VOUT1_D7_MODESELECT</td><td colspan="4">VOUT1_D7_DELAYMODE</td><td colspan="4">VOUT1_D7_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	VOUT1_D7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D7_MUXMODE	0x0: vout1_d7 0x2: emu9 0x3: vin4a_d23 0x4: vin3a_d23 0xE: gpio8_7 0xF: Driver off	RW	0xF

Table 18-680. CTRL_CORE_PAD_VOUT1_D8

Address Offset	0x0000 15FC		
Physical Address	0x4A00 35FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VOUT1_D8_WAKEUPEVENT	VOUT1_D8_WAKEUPENABLE	RESERVED				VOUT1_D8_SLEWCONTROL	VOUT1_D8_INPUTENABLE	VOUT1_D8_PULLTYPESELECT	VOUT1_D8_PULLUDENABLE	RESERVED							VOUT1_D8_MODESELECT	VOUT1_D8_DELAYMODE				VOUT1_D8_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D8_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0

Bits	Field Name	Description	Type	Reset
24	VOUT1_D8_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D8_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D8_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D8_MUXMODE	0x0: vout1_d8 0x2: uart6_rxd 0x3: vin4a_d8 0x4: vin3a_d8 0xE: gpio8_8 0xF: Driver off	RW	0xF

Table 18-681. CTRL_CORE_PAD_VOUT1_D9

Address Offset	0x0000 1600		
Physical Address	0x4A00 3600	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_D9_WAKEUPEVENT	VOUT1_D9_WAKEUPENABLE	RESERVED	VOUT1_D9_SLEWCONTROL	VOUT1_D9_INPUTENABLE	VOUT1_D9_PULLTYPESELECT	VOUT1_D9_PULLUDENABLE	RESERVED	VOUT1_D9_MODESELECT	VOUT1_D9_DELAYMODE	VOUT1_D9_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D9_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D9_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D9_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D9_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D9_MUXMODE	0x0: vout1_d9 0x2: uart6_txd 0x3: vin4a_d9 0x4: vin3a_d9 0xE: gpio8_9 0xF: Driver off	RW	0xF

Table 18-682. CTRL_CORE_PAD_VOUT1_D10

Address Offset	0x0000 1604

Table 18-682. CTRL_CORE_PAD_VOUT1_D10 (continued)

Physical Address	0x4A00 3604	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	VOUT1_D10_WAKEUPEVENT	VOUT1_D10_WAKEUPENABLE	RESERVED	VOUT1_D10_SLEWCONTROL	VOUT1_D10_INPUTENABLE	VOUT1_D10_PULLTYPESELECT	VOUT1_D10_PULLUDENABLE	RESERVED	VOUT1_D10_MODESELECT	VOUT1_D10_DELAYMODE	VOUT1_D10_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D10_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D10_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D10_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D10_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VOUT1_D10_MUXMODE	0x0: vout1_d10 0x2: emu3 0x3: vin4a_d10 0x4: vin3a_d10 0x5: obs5 0x6: obs21 0x7: obs_irq2 0xE: gpio8_10 0xF: Driver off	RW	0xF

Table 18-683. CTRL_CORE_PAD_VOUT1_D11

Address Offset	0x0000 1608		
Physical Address	0x4A00 3608	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D11_WAKEUPEVENT	VOUT1_D11_WAKEUPENABLE	RESERVED	VOUT1_D11_SLEWCONTROL	VOUT1_D11_INPUTENABLE	VOUT1_D11_PULLTYPESELECT	VOUT1_D11_PULLUDENABLE	RESERVED	VOUT1_D11_MODESELECT	VOUT1_D11_DELAYMODE	VOUT1_D11_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D11_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D11_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VOUT1_D11_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D11_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D11_MUXMODE	0x0: vout1_d11 0x2: emu10 0x3: vin4a_d11 0x4: vin3a_d11 0x5: obs6 0x6: obs22 0x7: obs_dmarq2 0xE: gpio8_11 0xF: Driver off	RW	0xF

Table 18-684. CTRL_CORE_PAD_VOUT1_D12

Address Offset	0x0000 160C		
Physical Address	0x4A00 360C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D12_WAKEUPEVENT	VOUT1_D12_WAKEUPENABLE	RESERVED	VOUT1_D12_SLEWCONTROL	VOUT1_D12_INPUTENABLE	VOUT1_D12_PULLTYPESELECT	VOUT1_D12_PULLUDENABLE	RESERVED	VOUT1_D12_MODESELECT	VOUT1_D12_DELAYMODE	VOUT1_D12_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D12_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D12_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VOUT1_D12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D12_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D12_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D12_MUXMODE	0x0: vout1_d12 0x2: emu11 0x3: vin4a_d12 0x4: vin3a_d12 0x5: obs7 0x6: obs23 0xE: gpio8_12 0xF: Driver off	RW	0xF

Table 18-685. CTRL_CORE_PAD_VOUT1_D13

Address Offset	0x0000 1610		
Physical Address	0x4A00 3610	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_D13_WAKEUPEVENT	VOUT1_D13_WAKEUPENABLE	RESERVED	VOUT1_D13_SLEWCONTROL	VOUT1_D13_INPUTENABLE	VOUT1_D13_PULLTYPESELECT	VOUT1_D13_PULLUDENABLE	RESERVED	VOUT1_D13_MODESELECT	VOUT1_D13_DELAYMODE	VOUT1_D13_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D13_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D13_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D13_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D13_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VOUT1_D13_MUXMODE	0x0: vout1_d13 0x2: emu12 0x3: vin4a_d13 0x4: vin3a_d13 0x5: obs8 0x6: obs24 0xE: gpio8_13 0xF: Driver off	RW	0xF

Table 18-686. CTRL_CORE_PAD_VOUT1_D14

Address Offset	0x0000 1614		
Physical Address	0x4A00 3614	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D14_WAKEUPEVENT	VOUT1_D14_WAKEUPENABLE	RESERVED	VOUT1_D14_SLEWCONTROL	VOUT1_D14_INPUTENABLE	VOUT1_D14_PULLTYPESELECT	VOUT1_D14_PULLUDENABLE	RESERVED	VOUT1_D14_MODESELECT	VOUT1_D14_DELAYMODE	VOUT1_D14_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D14_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D14_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VOUT1_D14_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D14_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D14_MUXMODE	0x0: vout1_d14 0x2: emu13 0x3: vin4a_d14 0x4: vin3a_d14 0x5: obs9 0x6: obs25 0xE: gpio8_14 0xF: Driver off	RW	0xF

Table 18-687. CTRL_CORE_PAD_VOUT1_D15

Address Offset	0x0000 1618		
Physical Address	0x4A00 3618	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
Field	RESERVED	VOUT1_D15_WAKEUPEVENT	VOUT1_D15_WAKEUPENABLE	RESERVED	VOUT1_D15_SLEWCONTROL	VOUT1_D15_INPUTENABLE	VOUT1_D15_PULLTYPESELECT	VOUT1_D15_PULLUDENABLE	RESERVED	VOUT1_D15_MODESELECT	VOUT1_D15_DELAYMODE	VOUT1_D15_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D15_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D15_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VOUT1_D15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D15_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D15_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D15_MUXMODE	0x0: vout1_d15 0x2: emu14 0x3: vin4a_d15 0x4: vin3a_d15 0x5: obs10 0x6: obs26 0xE: gpio8_15 0xF: Driver off	RW	0xF

Table 18-688. CTRL_CORE_PAD_VOUT1_D16

Address Offset	0x0000 161C		
Physical Address	0x4A00 361C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_D16_WAKEUPEVENT	VOUT1_D16_WAKEUPENABLE	RESERVED	VOUT1_D16_SLEWCONTROL	VOUT1_D16_INPUTENABLE	VOUT1_D16_PULLTYPESELECT	VOUT1_D16_PULLUDENABLE	RESERVED	VOUT1_D16_MODESELECT	VOUT1_D16_DELAYMODE	VOUT1_D16_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D16_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D16_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D16_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D16_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D16_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D16_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D16_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D16_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D16_MUXMODE	0x0: vout1_d16 0x2: uart7_rxd 0x3: vin4a_d0 0x4: vin3a_d0 0xE: gpio8_16 0xF: Driver off	RW	0xF

Table 18-689. CTRL_CORE_PAD_VOUT1_D17

Address Offset	0x0000 1620		
Physical Address	0x4A00 3620	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D17_WAKEUPEVENT	VOUT1_D17_WAKEUPENABLE	RESERVED	VOUT1_D17_SLEWCONTROL	VOUT1_D17_INPUTENABLE	VOUT1_D17_PULLTYPESELECT	VOUT1_D17_PULLUDENABLE	RESERVED	VOUT1_D17_MODESELECT	VOUT1_D17_DELAYMODE	VOUT1_D17_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D17_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D17_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D17_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D17_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D17_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D17_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D17_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D17_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VOUT1_D17_MUXMODE	0x0: vout1_d17 0x2: uart7_txd 0x3: vin4a_d1 0x4: vin3a_d1 0xE: gpio8_17 0xF: Driver off	RW	0xF

Table 18-690. CTRL_CORE_PAD_VOUT1_D18

Address Offset	0x0000 1624		
Physical Address	0x4A00 3624	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	VOUT1_D18_WAKEUPEVENT	VOUT1_D18_WAKEUPENABLE	RESERVED	VOUT1_D18_SLEWCONTROL	VOUT1_D18_INPUTENABLE	VOUT1_D18_PULLTYPESELECT	VOUT1_D18_PULLUDENABLE	RESERVED	VOUT1_D18_MODESELECT	VOUT1_D18_DELAYMODE	VOUT1_D18_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D18_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D18_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D18_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D18_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D18_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D18_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VOUT1_D18_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D18_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D18_MUXMODE	0x0: vout1_d18 0x2: emu4 0x3: vin4a_d2 0x4: vin3a_d2 0x5: obs11 0x6: obs27 0xE: gpio8_18 0xF: Driver off	RW	0xF

Table 18-691. CTRL_CORE_PAD_VOUT1_D19

Address Offset	0x0000 1628		
Physical Address	0x4A00 3628	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	VOUT1_D19_WAKEUPEVENT	VOUT1_D19_WAKEUPENABLE	RESERVED	VOUT1_D19_SLEWCONTROL	VOUT1_D19_INPUTENABLE	VOUT1_D19_PULLTYPESELECT	VOUT1_D19_PULLUDENABLE	RESERVED	VOUT1_D19_MODESELECT	VOUT1_D19_DELAYMODE	VOUT1_D19_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D19_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D19_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VOUT1_D19_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D19_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D19_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D19_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D19_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D19_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D19_MUXMODE	0x0: vout1_d19 0x2: emu15 0x3: vin4a_d3 0x4: vin3a_d3 0x5: obs12 0x6: obs28 0xE: gpio8_19 0xF: Driver off	RW	0xF

Table 18-692. CTRL_CORE_PAD_VOUT1_D20

Address Offset	0x0000 162C		
Physical Address	0x4A00 362C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_D20_WAKEUPEVENT	VOUT1_D20_WAKEUPENABLE	RESERVED	VOUT1_D20_SLEWCONTROL	VOUT1_D20_INPUTENABLE	VOUT1_D20_PULLTYPESELECT	VOUT1_D20_PULLUDENABLE	RESERVED	VOUT1_D20_MODESELECT	VOUT1_D20_DELAYMODE	VOUT1_D20_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D20_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D20_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D20_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D20_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D20_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D20_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D20_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D20_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	VOUT1_D20_MUXMODE	0x0: vout1_d20 0x2: emu16 0x3: vin4a_d4 0x4: vin3a_d4 0x5: obs13 0x6: obs29 0xE: gpio8_20 0xF: Driver off	RW	0xF

Table 18-693. CTRL_CORE_PAD_VOUT1_D21

Address Offset	0x0000 1630		
Physical Address	0x4A00 3630	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VOUT1_D21_WAKEUPEVENT	VOUT1_D21_WAKEUPENABLE	RESERVED				VOUT1_D21_SLEWCONTROL	VOUT1_D21_INPUTENABLE	VOUT1_D21_PULLTYPESELECT	VOUT1_D21_PULLUDENABLE	RESERVED							VOUT1_D21_MODESELECT	VOUT1_D21_DELAYMODE				VOUT1_D21_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D21_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D21_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D21_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D21_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D21_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D21_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	VOUT1_D21_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D21_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D21_MUXMODE	0x0: vout1_d21 0x2: emu17 0x3: vin4a_d5 0x4: vin3a_d5 0x5: obs14 0x6: obs30 0xE: gpio8_21 0xF: Driver off	RW	0xF

Table 18-694. CTRL_CORE_PAD_VOUT1_D22

Address Offset	0x0000 1634		
Physical Address	0x4A00 3634	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VOUT1_D22_WAKEUPEVENT	VOUT1_D22_WAKEUPENABLE	RESERVED				VOUT1_D22_SLEWCONTROL	VOUT1_D22_INPUTENABLE	VOUT1_D22_PULLTYPESELECT	VOUT1_D22_PULLUDENABLE	RESERVED							VOUT1_D22_MODESELECT	VOUT1_D22_DELAYMODE				VOUT1_D22_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D22_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D22_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	VOUT1_D22_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D22_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D22_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D22_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D22_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D22_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D22_MUXMODE	0x0: vout1_d22 0x2: emu18 0x3: vin4a_d6 0x4: vin3a_d6 0x5: obs15 0x6: obs31 0xE: gpio8_22 0xF: Driver off	RW	0xF

Table 18-695. CTRL_CORE_PAD_VOUT1_D23

Address Offset	0x0000 1638		
Physical Address	0x4A00 3638	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	VOUT1_D23_WAKEUPEVENT	VOUT1_D23_WAKEUPENABLE	RESERVED	VOUT1_D23_SLEWCONTROL	VOUT1_D23_INPUTENABLE	VOUT1_D23_PULLTYPESELECT	VOUT1_D23_PULLUDENABLE	RESERVED	VOUT1_D23_MODESELECT	VOUT1_D23_DELAYMODE	VOUT1_D23_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VOUT1_D23_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	VOUT1_D23_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	VOUT1_D23_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D23_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D23_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D23_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	VOUT1_D23_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	VOUT1_D23_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	VOUT1_D23_MUXMODE	0x0: vout1_d23 0x2: emu19 0x3: vin4a_d7 0x4: vin3a_d7 0x8: spi3_cs3 0xE: gpio8_23 0xF: Driver off	RW	0xF

Table 18-696. CTRL_CORE_PAD_MDIO_MCLK

Address Offset	0x0000 163C		
Physical Address	0x4A00 363C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MDIO_MCLK_WAKEUPEVENT	MDIO_MCLK_WAKEUPENABLE	RESERVED	MDIO_MCLK_SLEWCONTROL	MDIO_MCLK_INPUTENABLE	MDIO_MCLK_PULLTYPESELECT	MDIO_MCLK_PULLUDENABLE	RESERVED	MDIO_MCLK_MODESELECT	MDIO_MCLK_DELAYMODE	MDIO_MCLK_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MDIO_MCLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MDIO_MCLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MDIO_MCLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MDIO_MCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MDIO_MCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MDIO_MCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MDIO_MCLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MDIO_MCLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MDIO_MCLK_MUXMODE	0x0: mdio_mclk 0x1: uart3_rtsn 0x3: mii0_col 0x4: vin2a_clk0 0x5: vin4b_clk1 0xE: gpio5_15 0xF: Driver off	RW	0xF

Table 18-697. CTRL_CORE_PAD_MDIO_D

Address Offset	0x0000 1640		
Physical Address	0x4A00 3640	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	MDIO_D_WAKEUPEVENT	MDIO_D_WAKEUPENABLE	RESERVED	MDIO_D_SLEWCONTROL	MDIO_D_INPUTENABLE	MDIO_D_PULLTYPESELECT	MDIO_D_PULLUDENABLE	RESERVED	MDIO_D_MODESELECT	MDIO_D_DELAYMODE	MDIO_D_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MDIO_D_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MDIO_D_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MDIO_D_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MDIO_D_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MDIO_D_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MDIO_D_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MDIO_D_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MDIO_D_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MDIO_D_MUXMODE	0x0: mdio_d 0x1: uart3_ctsn 0x3: mii0_txer 0x4: vin2a_d0 0x5: vin4b_d0 0xE: gpio5_16 0xF: Driver off	RW	0xF

Table 18-698. CTRL_CORE_PAD_RMII_MHZ_50_CLK

Address Offset	0x0000 1644		
Physical Address	0x4A00 3644	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	RMII_MHZ_50_CLK_WAKEUPEVENT	RMII_MHZ_50_CLK_WAKEUPENABLE	RESERVED	RMII_MHZ_50_CLK_SLEWCONTROL	RMII_MHZ_50_CLK_INPUTENABLE	RMII_MHZ_50_CLK_PULLTYPESELECT	RMII_MHZ_50_CLK_PULLUDENABLE	RESERVED	RMII_MHZ_50_CLK_MODESELECT	RMII_MHZ_50_CLK_DELAYMODE	RMII_MHZ_50_CLK_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RMII_MHZ_50_CLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RMII_MHZ_50_CLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	RMII_MHZ_50_CLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RMII_MHZ_50_CLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RMII_MHZ_50_CLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RMII_MHZ_50_CLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RMII_MHZ_50_CLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RMII_MHZ_50_CLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RMII_MHZ_50_CLK_MUXMODE	0x0: RMII_MHZ_50_CLK 0x4: vin2a_d11 0xE: gpio5_17 0xF: Driver off	RW	0xF

Table 18-699. CTRL_CORE_PAD_UART3_RXD

Address Offset	0x0000 1648		
Physical Address	0x4A00 3648	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	UART3_RXD_WAKEUPEVENT	UART3_RXD_WAKEUPENABLE	RESERVED	UART3_RXD_SLEWCONTROL	UART3_RXD_INPUTENABLE	UART3_RXD_PULLTYPESELECT	UART3_RXD_PULLUDENABLE	RESERVED	UART3_RXD_MODESELECT	UART3_RXD_DELAYMODE	UART3_RXD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	UART3_RXD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART3_RXD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART3_RXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	UART3_RXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART3_RXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	UART3_RXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART3_RXD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART3_RXD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART3_RXD_MUXMODE	0x0: uart3_rxd 0x2: rmii1_crs 0x3: mii0_rxdv 0x4: vin2a_d1 0x5: vin4b_d1 0x7: spi3_sclk 0xE: gpio5_18 0xF: Driver off	RW	0xF

Table 18-700. CTRL_CORE_PAD_UART3_TXD

Address Offset	0x0000 164C		
Physical Address	0x4A00 364C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	UART3_TXD_WAKEUPEVENT	UART3_TXD_WAKEUPENABLE	RESERVED	UART3_TXD_SLEWCONTROL	UART3_TXD_INPUTENABLE	UART3_TXD_PULLTYPESELECT	UART3_TXD_PULLUDENABLE	RESERVED	UART3_TXD_MODESELECT	UART3_TXD_DELAYMODE	UART3_TXD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART3_TXD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART3_TXD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART3_TXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	UART3_TXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART3_TXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	UART3_TXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART3_TXD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART3_TXD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	UART3_TXD_MUXMODE	0x0: uart3_txd 0x2: rmii1_rxer 0x3: mii0_rxclk 0x4: vin2a_d2 0x5: vin4b_d2 0x7: spi3_d1 0x8: spi4_cs1 0xE: gpio5_19 0xF: Driver off	RW	0xF

Table 18-701. CTRL_CORE_PAD_RGMII0_TXC

Address Offset	0x0000 1650		
Physical Address	0x4A00 3650	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>RGMII0_TXC_WAKEUPEVENT</td><td>RGMII0_TXC_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>RGMII0_TXC_SLEWCONTROL</td><td>RGMII0_TXC_INPUTENABLE</td><td>RGMII0_TXC_PULLTYPESELECT</td><td>RGMII0_TXC_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>RGMII0_TXC_MODESELECT</td><td colspan="4">RGMII0_TXC_DELAYMODE</td><td colspan="4">RGMII0_TXC_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_TXC_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_TXC_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_TXC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RGMII0_TXC_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_TXC_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_TXC_MUXMODE	0x0: rgmii0_txc 0x1: uart3_ctsn 0x2: rmii1_rxd1 0x3: mii0_rxd3 0x4: vin2a_d3 0x5: vin4b_d3 0x6: usb4_ulpi_clk 0x7: spi3_d0 0x8: spi4_cs2 0xE: gpio5_20 0xF: Driver off	RW	0xF

Table 18-702. CTRL_CORE_PAD_RGMII0_TXCTL

Address Offset	0x0000 1654		
Physical Address	0x4A00 3654	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						RGMII0_TXCTL_WAKEUPEVENT	RGMII0_TXCTL_WAKEUPENABLE	RESERVED				RGMII0_TXCTL_SLEWCONTROL	RGMII0_TXCTL_INPUTENABLE	RGMII0_TXCTL_PULLTYPESELECT	RGMII0_TXCTL_PULLUDENABLE	RESERVED							RGMII0_TXCTL_MODESELECT	RGMII0_TXCTL_DELAYMODE				RGMII0_TXCTL_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_TXCTL_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0

Bits	Field Name	Description	Type	Reset
24	RGMII0_TXCTL_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_TXCTL_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXCTL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXCTL_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXCTL_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_TXCTL_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_TXCTL_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_TXCTL_MUXMODE	0x0: rgmii0_txctl 0x1: uart3_rtsn 0x2: rmii1_rxd0 0x3: mii0_rxd2 0x4: vin2a_d4 0x5: vin4b_d4 0x6: usb4_ulpi_stp 0x7: spi3_cs0 0x8: spi4_cs3 0xE: gpio5_21 0xF: Driver off	RW	0xF

Table 18-703. CTRL_CORE_PAD_RGMII0_TXD3

Address Offset	0x0000 1658		
Physical Address	0x4A00 3658	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RGMII0_TXD3_WAKEUPEVENT	RGMII0_TXD3_WAKEUPENABLE	RESERVED	RGMII0_TXD3_SLEWCONTROL	RGMII0_TXD3_INPUTENABLE	RGMII0_TXD3_PULLTYPESELECT	RGMII0_TXD3_PULLUDENABLE	RESERVED	RGMII0_TXD3_MODESELECT	RGMII0_TXD3_DELAYMODE	RGMII0_TXD3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_TXD3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_TXD3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_TXD3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_TXD3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_TXD3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	RGMII0_TXD3_MUXMODE	0x0: rgmii0_txd3 0x1: rmii0_crs 0x3: mii0_crs 0x4: vin2a_de0 0x5: vin4b_de1 0x6: usb4_ulpi_dir 0x7: spi4_sclk 0x8: uart4_rxd 0xE: gpio5_22 0xF: Driver off	RW	0xF

Table 18-704. CTRL_CORE_PAD_RGMII0_TXD2

Address Offset	0x0000 165C		
Physical Address	0x4A00 365C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>RGMII0_TXD2_WAKEUPEVENT</td><td>RGMII0_TXD2_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>RGMII0_TXD2_SLEWCONTROL</td><td>RGMII0_TXD2_INPUTENABLE</td><td>RGMII0_TXD2_PULLTYPESELECT</td><td>RGMII0_TXD2_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>RGMII0_TXD2_MODESELECT</td><td colspan="4">RGMII0_TXD2_DELAYMODE</td><td colspan="4">RGMII0_TXD2_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_TXD2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_TXD2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_TXD2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RGMII0_TXD2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_TXD2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_TXD2_MUXMODE	0x0: rgmii0_txd2 0x1: rmii0_rxer 0x3: mii0_rxer 0x4: vin2a_hsync0 0x5: vin4b_hsync1 0x6: usb4_ulpi_nxt 0x7: spi4_d1 0x8: uart4_txd 0xE: gpio5_23 0xF: Driver off	RW	0xF

Table 18-705. CTRL_CORE_PAD_RGMII0_TXD1

Address Offset	0x0000 1660		
Physical Address	0x4A00 3660	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	RGMII0_TXD1_WAKEUPEVENT	RGMII0_TXD1_WAKEUPENABLE	RESERVED	RGMII0_TXD1_SLEWCONTROL	RGMII0_TXD1_INPUTENABLE	RGMII0_TXD1_PULLTYPESELECT	RGMII0_TXD1_PULLUDENABLE	RESERVED	RGMII0_TXD1_MODESELECT	RGMII0_TXD1_DELAYMODE	RGMII0_TXD1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_TXD1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_TXD1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	RGMII0_TXD1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_TXD1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_TXD1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_TXD1_MUXMODE	0x0: rgmii0_txd1 0x1: rmii0_rxd1 0x3: mii0_rxd1 0x4: vin2a_vsync0 0x5: vin4b_vsync1 0x6: usb4_ulpi_d0 0x7: spi4_d0 0x8: uart4_ctsn 0xE: gpio5_24 0xF: Driver off	RW	0xF

Table 18-706. CTRL_CORE_PAD_RGMII0_TXD0

Address Offset	0x0000 1664		
Physical Address	0x4A00 3664	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RGMII0_TXD0_WAKEUPEVENT	RGMII0_TXD0_WAKEUPENABLE	RESERVED	RGMII0_TXD0_SLEWCONTROL	RGMII0_TXD0_INPUTENABLE	RGMII0_TXD0_PULLTYPESELECT	RGMII0_TXD0_PULLUDENABLE	RESERVED	RGMII0_TXD0_MODESELECT	RGMII0_TXD0_DELAYMODE	RGMII0_TXD0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_TXD0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_TXD0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_TXD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_TXD0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_TXD0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	RGMII0_TXD0_MUXMODE	0x0: rgmii0_txd0 0x1: rmii0_rxd0 0x3: mii0_rxd0 0x4: vin2a_d10 0x6: usb4_ulpi_d1 0x7: spi4_cs0 0x8: uart4_rtsn 0xE: gpio5_25 0xF: Driver off	RW	0xF

Table 18-707. CTRL_CORE_PAD_RGMII0_RXC

Address Offset	0x0000 1668		
Physical Address	0x4A00 3668	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	RGMII0_RXC_WAKEUPEVENT	RGMII0_RXC_WAKEUPENABLE	RESERVED	RGMII0_RXC_SLEWCONTROL	RGMII0_RXC_INPUTENABLE	RGMII0_RXC_PULLTYPESELECT	RGMII0_RXC_PULLUDENABLE	RESERVED	RGMII0_RXC_MODESELECT	RGMII0_RXC_DELAYMODE	RGMII0_RXC_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_RXC_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_RXC_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_RXC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RGMII0_RXC_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_RXC_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_RXC_MUXMODE	0x0: rgmii0_rxc 0x2: rmii1_txen 0x3: mii0_txclk 0x4: vin2a_d5 0x5: vin4b_d5 0x6: usb4_ulpi_d2 0xE: gpio5_26 0xF: Driver off	RW	0xF

Table 18-708. CTRL_CORE_PAD_RGMII0_RXCTL

Address Offset	0x0000 166C		
Physical Address	0x4A00 366C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	RGMII0_RXCTL_WAKEUPEVENT	RGMII0_RXCTL_WAKEUPENABLE	RESERVED	RGMII0_RXCTL_SLEWCONTROL	RGMII0_RXCTL_INPUTENABLE	RGMII0_RXCTL_PULLTYPESELECT	RGMII0_RXCTL_PULLUDENABLE	RESERVED	RGMII0_RXCTL_MODESELECT	RGMII0_RXCTL_DELAYMODE	RGMII0_RXCTL_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_RXCTL_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_RXCTL_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	RGMII0_RXCTL_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXCTL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXCTL_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXCTL_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_RXCTL_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_RXCTL_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_RXCTL_MUXMODE	0x0: rgmii0_rxctl 0x2: rmii1_txd1 0x3: mii0_txd3 0x4: vin2a_d6 0x5: vin4b_d6 0x6: usb4_ulpi_d3 0xE: gpio5_27 0xF: Driver off	RW	0xF

Table 18-709. CTRL_CORE_PAD_RGMII0_RXD3

Address Offset	0x0000 1670		
Physical Address	0x4A00 3670	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RGMII0_RXD3_WAKEUPEVENT	RGMII0_RXD3_WAKEUPENABLE	RESERVED	RGMII0_RXD3_SLEWCONTROL	RGMII0_RXD3_INPUTENABLE	RGMII0_RXD3_PULLTYPESELECT	RGMII0_RXD3_PULLUDENABLE	RESERVED	RGMII0_RXD3_MODESELECT	RGMII0_RXD3_DELAYMODE	RGMII0_RXD3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_RXD3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_RXD3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_RXD3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXD3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_RXD3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_RXD3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	RGMII0_RXD3_MUXMODE	0x0: rgmii0_rxd3 0x2: rmii1_txd0 0x3: mii0_txd2 0x4: vin2a_d7 0x5: vin4b_d7 0x6: usb4_ulpi_d4 0xE: gpio5_28 0xF: Driver off	RW	0xF

Table 18-710. CTRL_CORE_PAD_RGMII0_RXD2

Address Offset	0x0000 1674		
Physical Address	0x4A00 3674	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	RGMII0_RXD2_WAKEUPEVENT	RGMII0_RXD2_WAKEUPENABLE	RESERVED	RGMII0_RXD2_SLEWCONTROL	RGMII0_RXD2_INPUTENABLE	RGMII0_RXD2_PULLTYPESELECT	RGMII0_RXD2_PULLUDENABLE	RESERVED	RGMII0_RXD2_MODESELECT	RGMII0_RXD2_DELAYMODE	RGMII0_RXD2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_RXD2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_RXD2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_RXD2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXD2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	RGMII0_RXD2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_RXD2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_RXD2_MUXMODE	0x0: rgmii0_rxd2 0x1: rmii0_txen 0x3: mii0_txen 0x4: vin2a_d8 0x6: usb4_ulpi_d5 0xE: gpio5_29 0xF: Driver off	RW	0xF

Table 18-711. CTRL_CORE_PAD_RGMII0_RXD1

Address Offset	0x0000 1678		
Physical Address	0x4A00 3678	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	RGMII0_RXD1_WAKEUPEVENT
24	RGMII0_RXD1_WAKEUPENABLE
23:20	RESERVED
19	RGMII0_RXD1_SLEWCONTROL
18	RGMII0_RXD1_INPUTENABLE
17	RGMII0_RXD1_PULLTYPESELECT
16	RGMII0_RXD1_PULLUDENABLE
15:9	RESERVED
8	RGMII0_RXD1_MODESELECT
7:4	RGMII0_RXD1_DELAYMODE
3:0	RGMII0_RXD1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RGMII0_RXD1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_RXD1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_RXD1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	RGMII0_RXD1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_RXD1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_RXD1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_RXD1_MUXMODE	0x0: rgmii0_rxd1 0x1: rmii0_txd1 0x3: mii0_txd1 0x4: vin2a_d9 0x6: usb4_ulpi_d6 0xE: gpio5_30 0xF: Driver off	RW	0xF

Table 18-712. CTRL_CORE_PAD_RGMII0_RXD0

Address Offset	0x0000 167C		
Physical Address	0x4A00 367C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						RGMII0_RXD0_WAKEUPEVENT	RGMII0_RXD0_WAKEUPENABLE	RESERVED				RGMII0_RXD0_SLEWCONTROL	RGMII0_RXD0_INPUTENABLE	RGMII0_RXD0_PULLTYPESELECT	RGMII0_RXD0_PULLUDENABLE	RESERVED							RGMII0_RXD0_MODESELECT	RGMII0_RXD0_DELAYMODE				RGMII0_RXD0_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	RGMII0_RXD0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RGMII0_RXD0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RGMII0_RXD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	RGMII0_RXD0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RGMII0_RXD0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RGMII0_RXD0_MUXMODE	0x0: rgmii0_rxd0 0x1: rmii0_txd0 0x3: mii0_txd0 0x4: vin2a_fld0 0x5: vin4b_fld1 0x6: usb4_ulpi_d7 0xE: gpio5_31 0xF: Driver off	RW	0xF

Table 18-713. CTRL_CORE_PAD_USB1_DRVVBUS

Address Offset	0x0000 1680		
Physical Address	0x4A00 3680	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	USB1_DRVVBUS_WAKEUPEVENT	USB1_DRVVBUS_WAKEUPENABLE	RESERVED	USB1_DRVVBUS_SLEWCONTROL	USB1_DRVVBUS_INPUTENABLE	USB1_DRVVBUS_PULLTYPESELECT	USB1_DRVVBUS_PULLUDENABLE	RESERVED	USB1_DRVVBUS_MODESELECT	USB1_DRVVBUS_DELAYMODE	USB1_DRVVBUS_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	USB1_DRVVBUS_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	USB1_DRVVBUS_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	USB1_DRVVBUS_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	USB1_DRVVBUS_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	USB1_DRVVBUS_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	USB1_DRVVBUS_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	USB1_DRVVBUS_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	USB1_DRVVBUS_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	USB1_DRVVBUS_MUXMODE	0x0: usb1_drvvbus 0x7: timer16 0xE: gpio6_12 0xF: Driver off	RW	0xF

Table 18-714. CTRL_CORE_PAD_USB2_DRVVBUS

Address Offset	0x0000 1684		
Physical Address	0x4A00 3684	**Instance**	CTRL_MODULE_CORE

Table 18-714. CTRL_CORE_PAD_USB2_DRVVBUS (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						USB2_DRVVBUS_WAKEUPEVENT	USB2_DRVVBUS_WAKEUPENABLE	RESERVED				USB2_DRVVBUS_SLEWCONTROL	USB2_DRVVBUS_INPUTENABLE	USB2_DRVVBUS_PULLTYPESELECT	USB2_DRVVBUS_PULLUDENABLE	RESERVED							USB2_DRVVBUS_MODESELECT	USB2_DRVVBUS_DELAYMODE				USB2_DRVVBUS_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	USB2_DRVVBUS_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	USB2_DRVVBUS_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	USB2_DRVVBUS_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	USB2_DRVVBUS_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	USB2_DRVVBUS_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	USB2_DRVVBUS_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	USB2_DRVVBUS_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	USB2_DRVVBUS_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	USB2_DRVVBUS_MUXMODE	0x0: usb2_drvvbus 0x7: timer15 0xE: gpio6_13 0xF: Driver off	RW	0xF

Table 18-715. CTRL_CORE_PAD_GPIO6_14

Address Offset	0x0000 1688		
Physical Address	0x4A00 3688	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPIO6_14_WAKEUPEVENT	GPIO6_14_WAKEUPENABLE	RESERVED	GPIO6_14_SLEWCONTROL	GPIO6_14_INPUTENABLE	GPIO6_14_PULLTYPESELECT	GPIO6_14_PULLUDENABLE	RESERVED	GPIO6_14_MODESELECT	GPIO6_14_DELAYMODE	GPIO6_14_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPIO6_14_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPIO6_14_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPIO6_14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPIO6_14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPIO6_14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPIO6_14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPIO6_14_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPIO6_14_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPIO6_14_MUXMODE	0x0: gpio6_14 0x1: mcasp1_axr8 0x2: dcan2_tx 0x3: uart10_rxd 0x6: vout2_hsync 0x8: vin4a_hsync0 0x9: i2c3_sda 0xA: timer1 0xE: gpio6_14 0xF: Driver off	RW	0xF

Table 18-716. CTRL_CORE_PAD_GPIO6_15

Address Offset	0x0000 168C		
Physical Address	0x4A00 368C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPIO6_15_WAKEUPEVENT	GPIO6_15_WAKEUPENABLE	RESERVED	GPIO6_15_SLEWCONTROL	GPIO6_15_INPUTENABLE	GPIO6_15_PULLTYPESELECT	GPIO6_15_PULLUDENABLE	RESERVED	GPIO6_15_MODESELECT	GPIO6_15_DELAYMODE	GPIO6_15_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPIO6_15_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPIO6_15_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	GPIO6_15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPIO6_15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPIO6_15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPIO6_15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPIO6_15_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPIO6_15_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPIO6_15_MUXMODE	0x0: gpio6_15 0x1: mcasp1_axr9 0x2: dcan2_rx 0x3: uart10_txd 0x6: vout2_vsync 0x8: vin4a_vsync0 0x9: i2c3_scl 0xA: timer2 0xE: gpio6_15 0xF: Driver off	RW	0xF

Table 18-717. CTRL_CORE_PAD_GPIO6_16

Address Offset	0x0000 1690		
Physical Address	0x4A00 3690	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	GPIO6_16_WAKEUPEVENT	GPIO6_16_WAKEUPENABLE	RESERVED	GPIO6_16_SLEWCONTROL	GPIO6_16_INPUTENABLE	GPIO6_16_PULLTYPESELECT	GPIO6_16_PULLUDENABLE	RESERVED	GPIO6_16_MODESELECT	GPIO6_16_DELAYMODE	GPIO6_16_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPIO6_16_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPIO6_16_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPIO6_16_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPIO6_16_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPIO6_16_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPIO6_16_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPIO6_16_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPIO6_16_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPIO6_16_MUXMODE	0x0: gpio6_16 0x1: mcasp1_axr10 0x6: vout2_fld 0x8: vin4a_fld0 0x9: clkout1 0xA: timer3 0xE: gpio6_16 0xF: Driver off	RW	0xF

Table 18-718. CTRL_CORE_PAD_XREF_CLK0

Address Offset	0x0000 1694		
Physical Address	0x4A00 3694	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	XREF_CLK0_WAKEUPEVENT	XREF_CLK0_WAKEUPENABLE	RESERVED	XREF_CLK0_SLEWCONTROL	XREF_CLK0_INPUTENABLE	XREF_CLK0_PULLTYPESELECT	XREF_CLK0_PULLUDENABLE	RESERVED	XREF_CLK0_MODESELECT	XREF_CLK0_DELAYMODE	XREF_CLK0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	XREF_CLK0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	XREF_CLK0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	XREF_CLK0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	XREF_CLK0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	XREF_CLK0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	XREF_CLK0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	XREF_CLK0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	XREF_CLK0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	XREF_CLK0_MUXMODE	0x0: xref_clk0 0x1: mcasp2_axr8 0x2: mcasp1_axr4 0x3: mcasp1_ahclkx 0x4: mcasp5_ahclkx 0x5: atl_clk0 0x7: vin6a_d0 0x8: hdq0 0x9: clkout2 0xA: timer13 0xE: gpio6_17 0xF: Driver off	RW	0xF

Table 18-719. CTRL_CORE_PAD_XREF_CLK1

Address Offset	0x0000 1698		
Physical Address	0x4A00 3698	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	XREF_CLK1_WAKEUPEVENT	XREF_CLK1_WAKEUPENABLE	RESERVED	XREF_CLK1_SLEWCONTROL	XREF_CLK1_INPUTENABLE	XREF_CLK1_PULLTYPESELECT	XREF_CLK1_PULLUDENABLE	RESERVED	XREF_CLK1_MODESELECT	XREF_CLK1_DELAYMODE	XREF_CLK1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	XREF_CLK1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	XREF_CLK1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	XREF_CLK1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	XREF_CLK1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	XREF_CLK1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	XREF_CLK1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8	XREF_CLK1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	XREF_CLK1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	XREF_CLK1_MUXMODE	0x0: xref_clk1 0x1: mcasp2_axr9 0x2: mcasp1_axr5 0x3: mcasp2_ahclkx 0x4: mcasp6_ahclkx 0x5: atl_clk1 0x7: vin6a_clk0 0xA: timer14 0xE: gpio6_18 0xF: Driver off	RW	0xF

Table 18-720. CTRL_CORE_PAD_XREF_CLK2

Address Offset	0x0000 169C		
Physical Address	0x4A00 369C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	XREF_CLK2_WAKEUPEVENT
24	XREF_CLK2_WAKEUPENABLE
23:20	RESERVED
19	XREF_CLK2_SLEWCONTROL
18	XREF_CLK2_INPUTENABLE
17	XREF_CLK2_PULLTYPESELECT
16	XREF_CLK2_PULLUDENABLE
15:9	RESERVED
8	XREF_CLK2_MODESELECT
7:4	XREF_CLK2_DELAYMODE
3:0	XREF_CLK2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	XREF_CLK2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	XREF_CLK2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	RESERVED		R	0x0
19	XREF_CLK2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	XREF_CLK2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	XREF_CLK2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	XREF_CLK2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	XREF_CLK2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	XREF_CLK2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	XREF_CLK2_MUXMODE	0x0: xref_clk2 0x1: mcasp2_axr10 0x2: mcasp1_axr6 0x3: mcasp3_ahclkx 0x4: mcasp7_ahclkx 0x5: atl_clk2 0x6: vout2_clk 0x8: vin4a_clk0 0xA: timer15 0xE: gpio6_19 0xF: Driver off	RW	0xF

Table 18-721. CTRL_CORE_PAD_XREF_CLK3

Address Offset	0x0000 16A0		
Physical Address	0x4A00 36A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	XREF_CLK3_WAKEUPEVENT	XREF_CLK3_WAKEUPENABLE	RESERVED	XREF_CLK3_SLEWCONTROL	XREF_CLK3_INPUTENABLE	XREF_CLK3_PULLTYPESELECT	XREF_CLK3_PULLUDENABLE	RESERVED	XREF_CLK3_MODESELECT	XREF_CLK3_DELAYMODE	XREF_CLK3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	XREF_CLK3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	XREF_CLK3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	XREF_CLK3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	XREF_CLK3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	XREF_CLK3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	XREF_CLK3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	XREF_CLK3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	XREF_CLK3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	XREF_CLK3_MUXMODE	0x0: xref_clk3 0x1: mcasp2_axr11 0x2: mcasp1_axr7 0x3: mcasp4_ahclkx 0x4: mcasp8_ahclkx 0x5: atl_clk3 0x6: vout2_de 0x7: hdq0 0x8: vin4a_de0 0x9: clkout3 0xA: timer16 0xE: gpio6_20 0xF: Driver off	RW	0xF

Table 18-722. CTRL_CORE_PAD_MCASP1_ACLKX

Address Offset	0x0000 16A4		
Physical Address	0x4A00 36A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_ACLKX_WAKEUPEVENT	MCASP1_ACLKX_WAKEUPENABLE	RESERVED	MCASP1_ACLKX_SLEWCONTROL	MCASP1_ACLKX_INPUTENABLE	MCASP1_ACLKX_PULLTYPESELECT	MCASP1_ACLKX_PULLUDENABLE	RESERVED	MCASP1_ACLKX_MODESELECT	MCASP1_ACLKX_DELAYMODE	MCASP1_ACLKX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_ACLKX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_ACLKX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_ACLKX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP1_ACLKX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_ACLKX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
16	MCASP1_ACLKX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_ACLKX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_ACLKX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_ACLKX_MUXMODE	0x0: mcasp1_aclkx 0x7: vin6a_fld0 0xA: i2c3_sda 0xE: gpio7_31 0xF: Driver off	RW	0xF

Table 18-723. CTRL_CORE_PAD_MCASP1_FSX

Address Offset	0x0000 16A8		
Physical Address	0x4A00 36A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31-26	25	24	23-20	19	18	17	16	15-9	8	7-4	3-0
RESERVED	MCASP1_FSX_WAKEUPEVENT	MCASP1_FSX_WAKEUPENABLE	RESERVED	MCASP1_FSX_SLEWCONTROL	MCASP1_FSX_INPUTENABLE	MCASP1_FSX_PULLTYPESELECT	MCASP1_FSX_PULLUDENABLE	RESERVED	MCASP1_FSX_MODESELECT	MCASP1_FSX_DELAYMODE	MCASP1_FSX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_FSX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_FSX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MCASP1_FSX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_FSX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_FSX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_FSX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_FSX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_FSX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_FSX_MUXMODE	0x0: mcasp1_fsx 0x7: vin6a_de0 0xA: i2c3_scl 0xE: gpio7_30 0xF: Driver off	RW	0xF

Table 18-724. CTRL_CORE_PAD_MCASP1_ACLKR

Address Offset	0x0000 16AC		
Physical Address	0x4A00 36AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MCASP1_ACLKR_WAKEUPEVENT	MCASP1_ACLKR_WAKEUPENABLE	RESERVED				MCASP1_ACLKR_SLEWCONTROL	MCASP1_ACLKR_INPUTENABLE	MCASP1_ACLKR_PULLTYPESELECT	MCASP1_ACLKR_PULLUDENABLE	RESERVED							MCASP1_ACLKR_MODESELECT	MCASP1_ACLKR_DELAYMODE				MCASP1_ACLKR_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_ACLKR_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_ACLKR_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_ACLKR_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP1_ACLKR_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_ACLKR_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_ACLKR_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_ACLKR_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_ACLKR_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_ACLKR_MUXMODE	0x0: mcasp1_aclkr 0x1: mcasp7_axr2 0x6: vout2_d0 0x8: vin4a_d0 0xA: i2c4_sda 0xE: gpio5_0 0xF: Driver off	RW	0xF

Table 18-725. CTRL_CORE_PAD_MCASP1_FSR

Address Offset	0x0000 16B0		
Physical Address	0x4A00 36B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP1_FSR_WAKEUPEVENT	MCASP1_FSR_WAKEUPENABLE	RESERVED	MCASP1_FSR_SLEWCONTROL	MCASP1_FSR_INPUTENABLE	MCASP1_FSR_PULLTYPESELECT	MCASP1_FSR_PULLUDENABLE	RESERVED	MCASP1_FSR_MODESELECT	MCASP1_FSR_DELAYMODE	MCASP1_FSR_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_FSR_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_FSR_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_FSR_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP1_FSR_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_FSR_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_FSR_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_FSR_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_FSR_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_FSR_MUXMODE	0x0: mcasp1_fsr 0x1: mcasp7_axr3 0x6: vout2_d1 0x8: vin4a_d1 0xA: i2c4_scl 0xE: gpio5_1 0xF: Driver off	RW	0xF

Table 18-726. CTRL_CORE_PAD_MCASP1_AXR0

Address Offset	0x0000 16B4		
Physical Address	0x4A00 36B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–26	25	24	23–20	19	18	17	16	15–9	8	7–4	3–0
RESERVED	MCASP1_AXR0_WAKEUPEVENT	MCASP1_AXR0_WAKEUPENABLE	RESERVED	MCASP1_AXR0_SLEWCONTROL	MCASP1_AXR0_INPUTENABLE	MCASP1_AXR0_PULLTYPESELECT	MCASP1_AXR0_PULLUDENABLE	RESERVED	MCASP1_AXR0_MODESELECT	MCASP1_AXR0_DELAYMODE	MCASP1_AXR0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP1_AXR0_MUXMODE	0x0: mcasp1_axr0 0x3: uart6_rxd 0x7: vin6a_vsync0 0xA: i2c5_sda 0xE: gpio5_2 0xF: Driver off	RW	0xF

Table 18-727. CTRL_CORE_PAD_MCASP1_AXR1

Address Offset	0x0000 16B8		
Physical Address	0x4A00 36B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20	19	18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	MCASP1_AXR1_WAKEUPEVENT	MCASP1_AXR1_WAKEUPENABLE	RESERVED	MCASP1_AXR1_SLEWCONTROL	MCASP1_AXR1_INPUTENABLE	MCASP1_AXR1_PULLTYPESELECT	MCASP1_AXR1_PULLUDENABLE	RESERVED	MCASP1_AXR1_MODESELECT	MCASP1_AXR1_DELAYMODE	MCASP1_AXR1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP1_AXR1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR1_MUXMODE	0x0: mcasp1_axr1 0x3: uart6_txd 0x7: vin6a_hsync0 0xA: i2c5_scl 0xE: gpio5_3 0xF: Driver off	RW	0xF

Table 18-728. CTRL_CORE_PAD_MCASP1_AXR2

Address Offset	0x0000 16BC		
Physical Address	0x4A00 36BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_AXR2_WAKEUPEVENT	MCASP1_AXR2_WAKEUPENABLE	RESERVED	MCASP1_AXR2_SLEWCONTROL	MCASP1_AXR2_INPUTENABLE	MCASP1_AXR2_PULLTYPESELECT	MCASP1_AXR2_PULLUDENABLE	RESERVED	MCASP1_AXR2_MODESELECT	MCASP1_AXR2_DELAYMODE	MCASP1_AXR2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	MCASP1_AXR2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR2_MUXMODE	0x0: mcasp1_axr2 0x1: mcasp6_axr2 0x3: uart6_ctsn 0x6: vout2_d2 0x8: vin4a_d2 0xE: gpio5_4 0xF: Driver off	RW	0xF

Table 18-729. CTRL_CORE_PAD_MCASP1_AXR3

Address Offset	0x0000 16C0		
Physical Address	0x4A00 36C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_AXR3_WAKEUPEVENT	MCASP1_AXR3_WAKEUPENABLE	RESERVED	MCASP1_AXR3_SLEWCONTROL	MCASP1_AXR3_INPUTENABLE	MCASP1_AXR3_PULLTYPESELECT	MCASP1_AXR3_PULLUDENABLE	RESERVED	MCASP1_AXR3_MODESELECT	MCASP1_AXR3_DELAYMODE	MCASP1_AXR3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	MCASP1_AXR3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP1_AXR3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR3_MUXMODE	0x0: mcasp1_axr3 0x1: mcasp6_axr3 0x3: uart6_rtsn 0x6: vout2_d3 0x8: vin4a_d3 0xE: gpio5_5 0xF: Driver off	RW	0xF

Table 18-730. CTRL_CORE_PAD_MCASP1_AXR4

Address Offset	0x0000 16C4		
Physical Address	0x4A00 36C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP1_AXR4_WAKEUPEVENT	MCASP1_AXR4_WAKEUPENABLE	RESERVED	MCASP1_AXR4_SLEWCONTROL	MCASP1_AXR4_INPUTENABLE	MCASP1_AXR4_PULLTYPESELECT	MCASP1_AXR4_PULLUDENABLE	RESERVED	MCASP1_AXR4_MODESELECT	MCASP1_AXR4_DELAYMODE	MCASP1_AXR4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP1_AXR4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR4_MUXMODE	0x0: mcasp1_axr4 0x1: mcasp4_axr2 0x6: vout2_d4 0x8: vin4a_d4 0xE: gpio5_6 0xF: Driver off	RW	0xF

Table 18-731. CTRL_CORE_PAD_MCASP1_AXR5

Address Offset	0x0000 16C8		
Physical Address	0x4A00 36C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>MCASP1_AXR5_WAKEUPEVENT</td><td>MCASP1_AXR5_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>MCASP1_AXR5_SLEWCONTROL</td><td>MCASP1_AXR5_INPUTENABLE</td><td>MCASP1_AXR5_PULLTYPESELECT</td><td>MCASP1_AXR5_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>MCASP1_AXR5_MODESELECT</td><td colspan="4">MCASP1_AXR5_DELAYMODE</td><td colspan="4">MCASP1_AXR5_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP1_AXR5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP1_AXR5_MUXMODE	0x0: mcasp1_axr5 0x1: mcasp4_axr3 0x6: vout2_d5 0x8: vin4a_d5 0xE: gpio5_7 0xF: Driver off	RW	0xF

Table 18-732. CTRL_CORE_PAD_MCASP1_AXR6

Address Offset	0x0000 16CC		
Physical Address	0x4A00 36CC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_AXR6_WAKEUPEVENT	MCASP1_AXR6_WAKEUPENABLE	RESERVED	MCASP1_AXR6_SLEWCONTROL	MCASP1_AXR6_INPUTENABLE	MCASP1_AXR6_PULLTYPESELECT	MCASP1_AXR6_PULLUDENABLE	RESERVED	MCASP1_AXR6_MODESELECT	MCASP1_AXR6_DELAYMODE	MCASP1_AXR6_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP1_AXR6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP1_AXR6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR6_MUXMODE	0x0: mcasp1_axr6 0x1: mcasp5_axr2 0x6: vout2_d6 0x8: vin4a_d6 0xE: gpio5_8 0xF: Driver off	RW	0xF

Table 18-733. CTRL_CORE_PAD_MCASP1_AXR7

Address Offset	0x0000 16D0		
Physical Address	0x4A00 36D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MCASP1_AXR7_WAKEUPEVENT	MCASP1_AXR7_WAKEUPENABLE	RESERVED				MCASP1_AXR7_SLEWCONTROL	MCASP1_AXR7_INPUTENABLE	MCASP1_AXR7_PULLTYPESELECT	MCASP1_AXR7_PULLUDENABLE	RESERVED							MCASP1_AXR7_MODESELECT	MCASP1_AXR7_DELAYMODE				MCASP1_AXR7_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	MCASP1_AXR7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR7_MUXMODE	0x0: mcasp1_axr7 0x1: mcasp5_axr3 0x6: vout2_d7 0x8: vin4a_d7 0xA: timer4 0xE: gpio5_9 0xF: Driver off	RW	0xF

Table 18-734. CTRL_CORE_PAD_MCASP1_AXR8

Address Offset	0x0000 16D4		
Physical Address	0x4A00 36D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_AXR8_WAKEUPEVENT	MCASP1_AXR8_WAKEUPENABLE	RESERVED	MCASP1_AXR8_SLEWCONTROL	MCASP1_AXR8_INPUTENABLE	MCASP1_AXR8_PULLTYPESELECT	MCASP1_AXR8_PULLUDENABLE	RESERVED	MCASP1_AXR8_MODESELECT	MCASP1_AXR8_DELAYMODE	MCASP1_AXR8_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	MCASP1_AXR8_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR8_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR8_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR8_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR8_MUXMODE	0x0: mcasp1_axr8 0x1: mcasp6_axr0 0x3: spi3_sclk 0x7: vin6a_d15 0xA: timer5 0xE: gpio5_10 0xF: Driver off	RW	0xF

Table 18-735. CTRL_CORE_PAD_MCASP1_AXR9

Address Offset	0x0000 16D8		
Physical Address	0x4A00 36D8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP1_AXR9_WAKEUPEVENT	MCASP1_AXR9_WAKEUPENABLE	RESERVED	MCASP1_AXR9_SLEWCONTROL	MCASP1_AXR9_INPUTENABLE	MCASP1_AXR9_PULLTYPESELECT	MCASP1_AXR9_PULLUDENABLE	RESERVED	MCASP1_AXR9_MODESELECT	MCASP1_AXR9_DELAYMODE	MCASP1_AXR9_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR9_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR9_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR9_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR9_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR9_MUXMODE	0x0: mcasp1_axr9 0x1: mcasp6_axr1 0x3: spi3_d1 0x7: vin6a_d14 0xA: timer6 0xE: gpio5_11 0xF: Driver off	RW	0xF

Table 18-736. CTRL_CORE_PAD_MCASP1_AXR10

Address Offset	0x0000 16DC		
Physical Address	0x4A00 36DC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>MCASP1_AXR10_WAKEUPEVENT</td><td>MCASP1_AXR10_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>MCASP1_AXR10_SLEWCONTROL</td><td>MCASP1_AXR10_INPUTENABLE</td><td>MCASP1_AXR10_PULLTYPESELECT</td><td>MCASP1_AXR10_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>MCASP1_AXR10_MODESELECT</td><td colspan="4">MCASP1_AXR10_DELAYMODE</td><td colspan="4">MCASP1_AXR10_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR10_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR10_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR10_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR10_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP1_AXR10_MUXMODE	0x0: mcasp1_axr10 0x1: mcasp6_aclkx 0x2: mcasp6_aclkr 0x3: spi3_d0 0x7: vin6a_d13 0xA: timer7 0xE: gpio5_12 0xF: Driver off	RW	0xF

Table 18-737. CTRL_CORE_PAD_MCASP1_AXR11

Address Offset	0x0000 16E0		
Physical Address	0x4A00 36E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_AXR11_WAKEUPEVENT	MCASP1_AXR11_WAKEUPENABLE	RESERVED	MCASP1_AXR11_SLEWCONTROL	MCASP1_AXR11_INPUTENABLE	MCASP1_AXR11_PULLTYPESELECT	MCASP1_AXR11_PULLUDENABLE	RESERVED	MCASP1_AXR11_MODESELECT	MCASP1_AXR11_DELAYMODE	MCASP1_AXR11_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR11_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR11_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP1_AXR11_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR11_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR11_MUXMODE	0x0: mcasp1_axr11 0x1: mcasp6_fsx 0x2: mcasp6_fsr 0x3: spi3_cs0 0x7: vin6a_d12 0xA: timer8 0xE: gpio4_17 0xF: Driver off	RW	0xF

Table 18-738. CTRL_CORE_PAD_MCASP1_AXR12

Address Offset	0x0000 16E4		
Physical Address	0x4A00 36E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MCASP1_AXR12_WAKEUPEVENT	MCASP1_AXR12_WAKEUPENABLE	RESERVED				MCASP1_AXR12_SLEWCONTROL	MCASP1_AXR12_INPUTENABLE	MCASP1_AXR12_PULLTYPESELECT	MCASP1_AXR12_PULLUDENABLE	RESERVED							MCASP1_AXR12_MODESELECT	MCASP1_AXR12_DELAYMODE				MCASP1_AXR12_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR12_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR12_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MCASP1_AXR12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR12_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR12_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR12_MUXMODE	0x0: mcasp1_axr12 0x1: mcasp7_axr0 0x3: spi3_cs1 0x7: vin6a_d11 0xA: timer9 0xE: gpio4_18 0xF: Driver off	RW	0xF

Table 18-739. CTRL_CORE_PAD_MCASP1_AXR13

Address Offset	0x0000 16E8		
Physical Address	0x4A00 36E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP1_AXR13_WAKEUPEVENT	MCASP1_AXR13_WAKEUPENABLE	RESERVED	MCASP1_AXR13_SLEWCONTROL	MCASP1_AXR13_INPUTENABLE	MCASP1_AXR13_PULLTYPESELECT	MCASP1_AXR13_PULLUDENABLE	RESERVED	MCASP1_AXR13_MODESELECT	MCASP1_AXR13_DELAYMODE	MCASP1_AXR13_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR13_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR13_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR13_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR13_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR13_MUXMODE	0x0: mcasp1_axr13 0x1: mcasp7_axr1 0x7: vin6a_d10 0xA: timer10 0xE: gpio6_4 0xF: Driver off	RW	0xF

Table 18-740. CTRL_CORE_PAD_MCASP1_AXR14

Address Offset	0x0000 16EC		
Physical Address	0x4A00 36EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_AXR14_WAKEUPEVENT	MCASP1_AXR14_WAKEUPENABLE	RESERVED	MCASP1_AXR14_SLEWCONTROL	MCASP1_AXR14_INPUTENABLE	MCASP1_AXR14_PULLTYPESELECT	MCASP1_AXR14_PULLUDENABLE	RESERVED	MCASP1_AXR14_MODESELECT	MCASP1_AXR14_DELAYMODE	MCASP1_AXR14_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR14_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR14_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP1_AXR14_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR14_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP1_AXR14_MUXMODE	0x0: mcasp1_axr14 0x1: mcasp7_aclkx 0x2: mcasp7_aclkr 0x7: vin6a_d9 0xA: timer11 0xE: gpio6_5 0xF: Driver off	RW	0xF

Table 18-741. CTRL_CORE_PAD_MCASP1_AXR15

Address Offset	0x0000 16F0		
Physical Address	0x4A00 36F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP1_AXR15_WAKEUPEVENT	MCASP1_AXR15_WAKEUPENABLE	RESERVED	MCASP1_AXR15_SLEWCONTROL	MCASP1_AXR15_INPUTENABLE	MCASP1_AXR15_PULLTYPESELECT	MCASP1_AXR15_PULLUDENABLE	RESERVED	MCASP1_AXR15_MODESELECT	MCASP1_AXR15_DELAYMODE	MCASP1_AXR15_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP1_AXR15_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP1_AXR15_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP1_AXR15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP1_AXR15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP1_AXR15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP1_AXR15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP1_AXR15_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP1_AXR15_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP1_AXR15_MUXMODE	0x0: mcasp1_axr15 0x1: mcasp7_fsx 0x2: mcasp7_fsr 0x7: vin6a_d8 0xA: timer12 0xE: gpio6_6 0xF: Driver off	RW	0xF

Table 18-742. CTRL_CORE_PAD_MCASP2_ACLKX

Address Offset	0x0000 16F4		
Physical Address	0x4A00 36F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_ACLKX_WAKEUPEVENT	MCASP2_ACLKX_WAKEUPENABLE	RESERVED	MCASP2_ACLKX_SLEWCONTROL	MCASP2_ACLKX_INPUTENABLE	MCASP2_ACLKX_PULLTYPESELECT	MCASP2_ACLKX_PULLUDENABLE	RESERVED	MCASP2_ACLKX_MODESELECT	MCASP2_ACLKX_DELAYMODE	MCASP2_ACLKX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_ACLKX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_ACLKX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_ACLKX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	MCASP2_ACLKX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_ACLKX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_ACLKX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_ACLKX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_ACLKX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_ACLKX_MUXMODE	0x0: mcasp2_aclkx 0x7: vin6a_d7 0xF: Driver off	RW	0xF

Table 18-743. CTRL_CORE_PAD_MCASP2_FSX

Address Offset	0x0000 16F8		
Physical Address	0x4A00 36F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_FSX_WAKEUPEVENT	MCASP2_FSX_WAKEUPENABLE	RESERVED	MCASP2_FSX_SLEWCONTROL	MCASP2_FSX_INPUTENABLE	MCASP2_FSX_PULLTYPESELECT	MCASP2_FSX_PULLUDENABLE	RESERVED	MCASP2_FSX_MODESELECT	MCASP2_FSX_DELAYMODE	MCASP2_FSX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_FSX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_FSX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	RESERVED		R	0x0
19	MCASP2_FSX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP2_FSX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_FSX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_FSX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_FSX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_FSX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_FSX_MUXMODE	0x0: mcasp2_fsx 0x7: vin6a_d6 0xF: Driver off	RW	0xF

Table 18-744. CTRL_CORE_PAD_MCASP2_ACLKR

Address Offset	0x0000 16FC		
Physical Address	0x4A00 36FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MCASP2_ACLKR_WAKEUPEVENT	MCASP2_ACLKR_WAKEUPENABLE	RESERVED				MCASP2_ACLKR_SLEWCONTROL	MCASP2_ACLKR_INPUTENABLE	MCASP2_ACLKR_PULLTYPESELECT	MCASP2_ACLKR_PULLUDENABLE	RESERVED							MCASP2_ACLKR_MODESELECT	MCASP2_ACLKR_DELAYMODE				MCASP2_ACLKR_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	MCASP2_ACLKR_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_ACLKR_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_ACLKR_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_ACLKR_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_ACLKR_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_ACLKR_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_ACLKR_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_ACLKR_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_ACLKR_MUXMODE	0x0: mcasp2_aclkr 0x1: mcasp8_axr2 0x6: vout2_d8 0x8: vin4a_d8 0xF: Driver off	RW	0xF

Table 18-745. CTRL_CORE_PAD_MCASP2_FSR

Address Offset	0x0000 1700		
Physical Address	0x4A00 3700	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP2_FSR_WAKEUPEVENT	MCASP2_FSR_WAKEUPENABLE	RESERVED	MCASP2_FSR_SLEWCONTROL	MCASP2_FSR_INPUTENABLE	MCASP2_FSR_PULLTYPESELECT	MCASP2_FSR_PULLUDENABLE	RESERVED	MCASP2_FSR_MODESELECT	MCASP2_FSR_DELAYMODE	MCASP2_FSR_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_FSR_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_FSR_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_FSR_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_FSR_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_FSR_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_FSR_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_FSR_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_FSR_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_FSR_MUXMODE	0x0: mcasp2_fsr 0x1: mcasp8_axr3 0x6: vout2_d9 0x8: vin4a_d9 0xF: Driver off	RW	0xF

Table 18-746. CTRL_CORE_PAD_MCASP2_AXR0

Address Offset	0x0000 1704

Table 18-746. CTRL_CORE_PAD_MCASP2_AXR0 (continued)

Physical Address	0x4A00 3704	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–26	25	24	23–20	19	18	17	16	15–9	8	7–4	3–0
RESERVED	MCASP2_AXR0_WAKEUPEVENT	MCASP2_AXR0_WAKEUPENABLE	RESERVED	MCASP2_AXR0_SLEWCONTROL	MCASP2_AXR0_INPUTENABLE	MCASP2_AXR0_PULLTYPESELECT	MCASP2_AXR0_PULLUDENABLE	RESERVED	MCASP2_AXR0_MODESELECT	MCASP2_AXR0_DELAYMODE	MCASP2_AXR0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_AXR0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_AXR0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_AXR0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_AXR0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP2_AXR0_MUXMODE	0x0: mcasp2_axr0 0x6: vout2_d10 0x8: vin4a_d10 0xF: Driver off	RW	0xF

Table 18-747. CTRL_CORE_PAD_MCASP2_AXR1

Address Offset	0x0000 1708		
Physical Address	0x4A00 3708	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_AXR1_WAKEUPEVENT	MCASP2_AXR1_WAKEUPENABLE	RESERVED	MCASP2_AXR1_SLEWCONTROL	MCASP2_AXR1_INPUTENABLE	MCASP2_AXR1_PULLTYPESELECT	MCASP2_AXR1_PULLUDENABLE	RESERVED	MCASP2_AXR1_MODESELECT	MCASP2_AXR1_DELAYMODE	MCASP2_AXR1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_AXR1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_AXR1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_AXR1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP2_AXR1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_AXR1_MUXMODE	0x0: mcasp2_axr1 0x6: vout2_d11 0x8: vin4a_d11 0xF: Driver off	RW	0xF

Table 18-748. CTRL_CORE_PAD_MCASP2_AXR2

Address Offset	0x0000 170C		
Physical Address	0x4A00 370C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_AXR2_WAKEUPEVENT	MCASP2_AXR2_WAKEUPENABLE	RESERVED	MCASP2_AXR2_SLEWCONTROL	MCASP2_AXR2_INPUTENABLE	MCASP2_AXR2_PULLTYPESELECT	MCASP2_AXR2_PULLUDENABLE	RESERVED	MCASP2_AXR2_MODESELECT	MCASP2_AXR2_DELAYMODE	MCASP2_AXR2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_AXR2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP2_AXR2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	MCASP2_AXR2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_AXR2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_AXR2_MUXMODE	0x0: mcasp2_axr2 0x1: mcasp3_axr2 0x7: vin6a_d5 0xE: gpio6_8 0xF: Driver off	RW	0xF

Table 18-749. CTRL_CORE_PAD_MCASP2_AXR3

Address Offset	0x0000 1710		
Physical Address	0x4A00 3710	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_AXR3_WAKEUPEVENT	MCASP2_AXR3_WAKEUPENABLE	RESERVED	MCASP2_AXR3_SLEWCONTROL	MCASP2_AXR3_INPUTENABLE	MCASP2_AXR3_PULLTYPESELECT	MCASP2_AXR3_PULLUDENABLE	RESERVED	MCASP2_AXR3_MODESELECT	MCASP2_AXR3_DELAYMODE	MCASP2_AXR3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	RESERVED		R	0x0
19	MCASP2_AXR3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP2_AXR3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_AXR3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_AXR3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_AXR3_MUXMODE	0x0: mcasp2_axr3 0x1: mcasp3_axr3 0x7: vin6a_d4 0xE: gpio6_9 0xF: Driver off	RW	0xF

Table 18-750. CTRL_CORE_PAD_MCASP2_AXR4

Address Offset	0x0000 1714		
Physical Address	0x4A00 3714	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_AXR4_WAKEUPEVENT	MCASP2_AXR4_WAKEUPENABLE	RESERVED	MCASP2_AXR4_SLEWCONTROL	MCASP2_AXR4_INPUTENABLE	MCASP2_AXR4_PULLTYPESELECT	MCASP2_AXR4_PULLUDENABLE	RESERVED	MCASP2_AXR4_MODESELECT	MCASP2_AXR4_DELAYMODE	MCASP2_AXR4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_AXR4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_AXR4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_AXR4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_AXR4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_AXR4_MUXMODE	0x0: mcasp2_axr4 0x1: mcasp8_axr0 0x6: vout2_d12 0x8: vin4a_d12 0xE: gpio1_4 0xF: Driver off	RW	0xF

Table 18-751. CTRL_CORE_PAD_MCASP2_AXR5

Address Offset	0x0000 1718		
Physical Address	0x4A00 3718	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP2_AXR5_WAKEUPEVENT	MCASP2_AXR5_WAKEUPENABLE	RESERVED	MCASP2_AXR5_SLEWCONTROL	MCASP2_AXR5_INPUTENABLE	MCASP2_AXR5_PULLTYPESELECT	MCASP2_AXR5_PULLUDENABLE	RESERVED	MCASP2_AXR5_MODESELECT	MCASP2_AXR5_DELAYMODE	MCASP2_AXR5_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_AXR5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_AXR5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_AXR5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_AXR5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_AXR5_MUXMODE	0x0: mcasp2_axr5 0x1: mcasp8_axr1 0x6: vout2_d13 0x8: vin4a_d13 0xE: gpio6_7 0xF: Driver off	RW	0xF

Table 18-752. CTRL_CORE_PAD_MCASP2_AXR6

Address Offset	0x0000 171C		
Physical Address	0x4A00 371C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_AXR6_WAKEUPEVENT	MCASP2_AXR6_WAKEUPENABLE	RESERVED	MCASP2_AXR6_SLEWCONTROL	MCASP2_AXR6_INPUTENABLE	MCASP2_AXR6_PULLTYPESELECT	MCASP2_AXR6_PULLUDENABLE	RESERVED	MCASP2_AXR6_MODESELECT	MCASP2_AXR6_DELAYMODE	MCASP2_AXR6_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_AXR6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_AXR6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_AXR6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP2_AXR6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP2_AXR6_MUXMODE	0x0: mcasp2_axr6 0x1: mcasp8_aclkx 0x2: mcasp8_aclkr 0x6: vout2_d14 0x8: vin4a_d14 0xE: gpio2_29 0xF: Driver off	RW	0xF

Table 18-753. CTRL_CORE_PAD_MCASP2_AXR7

Address Offset	0x0000 1720		
Physical Address	0x4A00 3720	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP2_AXR7_WAKEUPEVENT	MCASP2_AXR7_WAKEUPENABLE	RESERVED	MCASP2_AXR7_SLEWCONTROL	MCASP2_AXR7_INPUTENABLE	MCASP2_AXR7_PULLTYPESELECT	MCASP2_AXR7_PULLUDENABLE	RESERVED	MCASP2_AXR7_MODESELECT	MCASP2_AXR7_DELAYMODE	MCASP2_AXR7_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP2_AXR7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP2_AXR7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP2_AXR7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP2_AXR7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP2_AXR7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP2_AXR7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP2_AXR7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP2_AXR7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP2_AXR7_MUXMODE	0x0: mcasp2_axr7 0x1: mcasp8_fsx 0x2: mcasp8_fsr 0x6: vout2_d15 0x8: vin4a_d15 0xE: gpio1_5 0xF: Driver off	RW	0xF

Table 18-754. CTRL_CORE_PAD_MCASP3_ACLKX

Address Offset	0x0000 1724		
Physical Address	0x4A00 3724	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP3_ACLKX_WAKEUPEVENT	MCASP3_ACLKX_WAKEUPENABLE	RESERVED	MCASP3_ACLKX_SLEWCONTROL	MCASP3_ACLKX_INPUTENABLE	MCASP3_ACLKX_PULLTYPESELECT	MCASP3_ACLKX_PULLUDENABLE	RESERVED	MCASP3_ACLKX_MODESELECT	MCASP3_ACLKX_DELAYMODE	MCASP3_ACLKX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP3_ACLKX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP3_ACLKX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP3_ACLKX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	MCASP3_ACLKX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP3_ACLKX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP3_ACLKX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP3_ACLKX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP3_ACLKX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP3_ACLKX_MUXMODE	0x0: mcasp3_aclkx 0x1: mcasp3_aclkr 0x2: mcasp2_axr12 0x3: uart7_rxd 0x7: vin6a_d3 0xE: gpio5_13 0xF: Driver off	RW	0xF

Table 18-755. CTRL_CORE_PAD_MCASP3_FSX

Address Offset	0x0000 1728		
Physical Address	0x4A00 3728	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP3_FSX_WAKEUPEVENT	MCASP3_FSX_WAKEUPENABLE	RESERVED	MCASP3_FSX_SLEWCONTROL	MCASP3_FSX_INPUTENABLE	MCASP3_FSX_PULLTYPESELECT	MCASP3_FSX_PULLUDENABLE	RESERVED	MCASP3_FSX_MODESELECT	MCASP3_FSX_DELAYMODE	MCASP3_FSX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	MCASP3_FSX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP3_FSX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP3_FSX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP3_FSX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP3_FSX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP3_FSX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP3_FSX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP3_FSX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP3_FSX_MUXMODE	0x0: mcasp3_fsx 0x1: mcasp3_fsr 0x2: mcasp2_axr13 0x3: uart7_txd 0x7: vin6a_d2 0xE: gpio5_14 0xF: Driver off	RW	0xF

Table 18-756. CTRL_CORE_PAD_MCASP3_AXR0

Address Offset	0x0000 172C		
Physical Address	0x4A00 372C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP3_AXR0_WAKEUPEVENT	MCASP3_AXR0_WAKEUPENABLE	RESERVED	MCASP3_AXR0_SLEWCONTROL	MCASP3_AXR0_INPUTENABLE	MCASP3_AXR0_PULLTYPESELECT	MCASP3_AXR0_PULLUDENABLE	RESERVED	MCASP3_AXR0_MODESELECT	MCASP3_AXR0_DELAYMODE	MCASP3_AXR0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP3_AXR0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP3_AXR0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP3_AXR0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP3_AXR0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP3_AXR0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP3_AXR0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP3_AXR0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP3_AXR0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP3_AXR0_MUXMODE	0x0: mcasp3_axr0 0x2: mcasp2_axr14 0x3: uart7_ctsn 0x4: uart5_rxd 0x7: vin6a_d1 0xF: Driver off	RW	0xF

Table 18-757. CTRL_CORE_PAD_MCASP3_AXR1

Address Offset	0x0000 1730		
Physical Address	0x4A00 3730	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MCASP3_AXR1_WAKEUPEVENT	MCASP3_AXR1_WAKEUPENABLE	RESERVED				MCASP3_AXR1_SLEWCONTROL	MCASP3_AXR1_INPUTENABLE	MCASP3_AXR1_PULLTYPESELECT	MCASP3_AXR1_PULLUDENABLE	RESERVED							MCASP3_AXR1_MODESELECT	MCASP3_AXR1_DELAYMODE				MCASP3_AXR1_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP3_AXR1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP3_AXR1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP3_AXR1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCASP3_AXR1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP3_AXR1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP3_AXR1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP3_AXR1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP3_AXR1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP3_AXR1_MUXMODE	0x0: mcasp3_axr1 0x2: mcasp2_axr15 0x3: uart7_rtsn 0x4: uart5_txd 0x7: vin6a_d0 0x9: vin5a_fld0 0xF: Driver off	RW	0xF

Table 18-758. CTRL_CORE_PAD_MCASP4_ACLKX

Address Offset	0x0000 1734		
Physical Address	0x4A00 3734	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP4_ACLKX_WAKEUPEVENT	MCASP4_ACLKX_WAKEUPENABLE	RESERVED	MCASP4_ACLKX_SLEWCONTROL	MCASP4_ACLKX_INPUTENABLE	MCASP4_ACLKX_PULLTYPESELECT	MCASP4_ACLKX_PULLUDENABLE	RESERVED	MCASP4_ACLKX_MODESELECT	MCASP4_ACLKX_DELAYMODE	MCASP4_ACLKX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP4_ACLKX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP4_ACLKX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP4_ACLKX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP4_ACLKX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP4_ACLKX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP4_ACLKX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP4_ACLKX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP4_ACLKX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP4_ACLKX_MUXMODE	0x0: mcasp4_aclkx 0x1: mcasp4_aclkr 0x2: spi3_sclk 0x3: uart8_rxd 0x4: i2c4_sda 0x6: vout2_d16 0x8: vin4a_d16 0x9: vin5a_d15 0xF: Driver off	RW	0xF

Table 18-759. CTRL_CORE_PAD_MCASP4_FSX

Address Offset	0x0000 1738		
Physical Address	0x4A00 3738	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	MCASP4_FSX_WAKEUPEVENT
24	MCASP4_FSX_WAKEUPENABLE
23:20	RESERVED
19	MCASP4_FSX_SLEWCONTROL
18	MCASP4_FSX_INPUTENABLE
17	MCASP4_FSX_PULLTYPESELECT
16	MCASP4_FSX_PULLUDENABLE
15:9	RESERVED
8	MCASP4_FSX_MODESELECT
7:4	MCASP4_FSX_DELAYMODE
3:0	MCASP4_FSX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP4_FSX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP4_FSX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MCASP4_FSX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP4_FSX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP4_FSX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP4_FSX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP4_FSX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP4_FSX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP4_FSX_MUXMODE	0x0: mcasp4_fsx 0x1: mcasp4_fsr 0x2: spi3_d1 0x3: uart8_txd 0x4: i2c4_scl 0x6: vout2_d17 0x8: vin4a_d17 0x9: vin5a_d14 0xF: Driver off	RW	0xF

Table 18-760. CTRL_CORE_PAD_MCASP4_AXR0

Address Offset	0x0000 173C		
Physical Address	0x4A00 373C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP4_AXR0_WAKEUPEVENT	MCASP4_AXR0_WAKEUPENABLE	RESERVED	MCASP4_AXR0_SLEWCONTROL	MCASP4_AXR0_INPUTENABLE	MCASP4_AXR0_PULLTYPESELECT	MCASP4_AXR0_PULLUDENABLE	RESERVED	MCASP4_AXR0_MODESELECT	MCASP4_AXR0_DELAYMODE	MCASP4_AXR0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP4_AXR0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP4_AXR0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP4_AXR0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP4_AXR0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP4_AXR0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP4_AXR0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP4_AXR0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP4_AXR0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP4_AXR0_MUXMODE	0x0: mcasp4_axr0 0x2: spi3_d0 0x3: uart8_ctsn 0x4: uart4_rxd 0x6: vout2_d18 0x8: vin4a_d18 0x9: vin5a_d13 0xF: Driver off	RW	0xF

Table 18-761. CTRL_CORE_PAD_MCASP4_AXR1

Address Offset	0x0000 1740		
Physical Address	0x4A00 3740	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP4_AXR1_WAKEUPEVENT	MCASP4_AXR1_WAKEUPENABLE	RESERVED	MCASP4_AXR1_SLEWCONTROL	MCASP4_AXR1_INPUTENABLE	MCASP4_AXR1_PULLTYPESELECT	MCASP4_AXR1_PULLUDENABLE	RESERVED	MCASP4_AXR1_MODESELECT	MCASP4_AXR1_DELAYMODE	MCASP4_AXR1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP4_AXR1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP4_AXR1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP4_AXR1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP4_AXR1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP4_AXR1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP4_AXR1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP4_AXR1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP4_AXR1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP4_AXR1_MUXMODE	0x0: mcasp4_axr1 0x2: spi3_cs0 0x3: uart8_rtsn 0x4: uart4_txd 0x6: vout2_d19 0x8: vin4a_d19 0x9: vin5a_d12 0xF: Driver off	RW	0xF

Table 18-762. CTRL_CORE_PAD_MCASP5_ACLKX

Address Offset	0x0000 1744		
Physical Address	0x4A00 3744	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP5_ACLKX_WAKEUPEVENT	MCASP5_ACLKX_WAKEUPENABLE	RESERVED	MCASP5_ACLKX_SLEWCONTROL	MCASP5_ACLKX_INPUTENABLE	MCASP5_ACLKX_PULLTYPESELECT	MCASP5_ACLKX_PULLUDENABLE	RESERVED	MCASP5_ACLKX_MODESELECT	MCASP5_ACLKX_DELAYMODE	MCASP5_ACLKX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP5_ACLKX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP5_ACLKX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MCASP5_ACLKX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP5_ACLKX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP5_ACLKX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP5_ACLKX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP5_ACLKX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP5_ACLKX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP5_ACLKX_MUXMODE	0x0: mcasp5_aclkx 0x1: mcasp5_aclkr 0x2: spi4_sclk 0x3: uart9_rxd 0x4: i2c5_sda 0x5: mlb_clk 0x6: vout2_d20 0x8: vin4a_d20 0x9: vin5a_d11 0xF: Driver off	RW	0xF

Table 18-763. CTRL_CORE_PAD_MCASP5_FSX

Address Offset	0x0000 1748		
Physical Address	0x4A00 3748	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MCASP5_FSX_WAKEUPEVENT	MCASP5_FSX_WAKEUPENABLE	RESERVED	MCASP5_FSX_SLEWCONTROL	MCASP5_FSX_INPUTENABLE	MCASP5_FSX_PULLTYPESELECT	MCASP5_FSX_PULLUDENABLE	RESERVED	MCASP5_FSX_MODESELECT	MCASP5_FSX_DELAYMODE	MCASP5_FSX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP5_FSX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP5_FSX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP5_FSX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP5_FSX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP5_FSX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP5_FSX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP5_FSX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP5_FSX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MCASP5_FSX_MUXMODE	0x0: mcasp5_fsx 0x1: mcasp5_fsr 0x2: spi4_d1 0x3: uart9_txd 0x4: i2c5_scl 0x6: vout2_d21 0x8: vin4a_d21 0x9: vin5a_d10 0xF: Driver off	RW	0xF

Table 18-764. CTRL_CORE_PAD_MCASP5_AXR0

Address Offset	0x0000 174C		
Physical Address	0x4A00 374C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP5_AXR0_WAKEUPEVENT	MCASP5_AXR0_WAKEUPENABLE	RESERVED	MCASP5_AXR0_SLEWCONTROL	MCASP5_AXR0_INPUTENABLE	MCASP5_AXR0_PULLTYPESELECT	MCASP5_AXR0_PULLUDENABLE	RESERVED	MCASP5_AXR0_MODESELECT	MCASP5_AXR0_DELAYMODE	MCASP5_AXR0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP5_AXR0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP5_AXR0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MCASP5_AXR0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP5_AXR0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP5_AXR0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP5_AXR0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MCASP5_AXR0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP5_AXR0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP5_AXR0_MUXMODE	0x0: mcasp5_axr0 0x2: spi4_d0 0x3: uart9_ctsn 0x4: uart3_rxd 0x5: mlb_sig 0x6: vout2_d22 0x8: vin4a_d22 0x9: vin5a_d9 0xF: Driver off	RW	0xF

Table 18-765. CTRL_CORE_PAD_MCASP5_AXR1

Address Offset	0x0000 1750		
Physical Address	0x4A00 3750	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MCASP5_AXR1_WAKEUPEVENT	MCASP5_AXR1_WAKEUPENABLE	RESERVED	MCASP5_AXR1_SLEWCONTROL	MCASP5_AXR1_INPUTENABLE	MCASP5_AXR1_PULLTYPESELECT	MCASP5_AXR1_PULLUDENABLE	RESERVED	MCASP5_AXR1_MODESELECT	MCASP5_AXR1_DELAYMODE	MCASP5_AXR1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MCASP5_AXR1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MCASP5_AXR1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MCASP5_AXR1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MCASP5_AXR1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCASP5_AXR1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MCASP5_AXR1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MCASP5_AXR1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MCASP5_AXR1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MCASP5_AXR1_MUXMODE	0x0: mcasp5_axr1 0x2: spi4_cs0 0x3: uart9_rtsn 0x4: uart3_txd 0x5: mlb_dat 0x6: vout2_d23 0x8: vin4a_d23 0x9: vin5a_d8 0xF: Driver off	RW	0xF

Table 18-766. CTRL_CORE_PAD_MMC1_CLK

Address Offset	0x0000 1754		
Physical Address	0x4A00 3754	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC1_CLK_WAKEUPEVENT	MMC1_CLK_WAKEUPENABLE	RESERVED	MMC1_CLK_ACTIVE	MMC1_CLK_PULLTYPESELECT	MMC1_CLK_PULLUDENABLE	RESERVED	MMC1_CLK_MODESELECT	MMC1_CLK_DELAYMODE	MMC1_CLK_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	MMC1_CLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_CLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED	Reserved	R	0x0
18	MMC1_CLK_ACTIVE	Controls enabling/disabling of the input buffer. 0x0: Input buffer is disabled 0x1: Input buffer is enabled	RW	0x1
17	MMC1_CLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC1_CLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED	Reserved	R	0x0
8	MMC1_CLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_CLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC1_CLK_MUXMODE	0x0: mmc1_clk 0xE: gpio6_21 0xF: Driver off	RW	0xF

Table 18-767. CTRL_CORE_PAD_MMC1_CMD

Address Offset	0x0000 1758		
Physical Address	0x4A00 3758	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC1_CMD_WAKEUPEVENT	MMC1_CMD_WAKEUPENABLE	RESERVED	MMC1_CMD_ACTIVE	MMC1_CMD_PULLTYPESELECT	MMC1_CMD_PULLUDENABLE	RESERVED	MMC1_CMD_MODESELECT	MMC1_CMD_DELAYMODE	MMC1_CMD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	MMC1_CMD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_CMD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED	Reserved	R	0x0
18	MMC1_CMD_ACTIVE	Controls enabling/disabling of the input buffer. 0x0: Input buffer is disabled 0x1: Input buffer is enabled	RW	0x1
17	MMC1_CMD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC1_CMD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED	Reserved	R	0x0
8	MMC1_CMD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_CMD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC1_CMD_MUXMODE	0x0: mmc1_cmd 0xE: gpio6_22 0xF: Driver off	RW	0xF

Table 18-768. CTRL_CORE_PAD_MMC1_DAT0

Address Offset	0x0000 175C		
Physical Address	0x4A00 375C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC1_DAT0_WAKEUPEVENT	MMC1_DAT0_WAKEUPENABLE	RESERVED	MMC1_DAT0_ACTIVE	MMC1_DAT0_PULLTYPESELECT	MMC1_DAT0_PULLUDENABLE	RESERVED	MMC1_DAT0_MODESELECT	MMC1_DAT0_DELAYMODE	MMC1_DAT0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	MMC1_DAT0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_DAT0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED	Reserved	R	0x0
18	MMC1_DAT0_ACTIVE	Controls enabling/disabling of the input buffer. 0x0: Input buffer is disabled 0x1: Input buffer is enabled	RW	0x1
17	MMC1_DAT0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC1_DAT0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED	Reserved	R	0x0
8	MMC1_DAT0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_DAT0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC1_DAT0_MUXMODE	0x0: mmc1_dat0 0xE: gpio6_23 0xF: Driver off	RW	0xF

Table 18-769. CTRL_CORE_PAD_MMC1_DAT1

Address Offset	0x0000 1760		
Physical Address	0x4A00 3760	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC1_DAT1_WAKEUPEVENT	MMC1_DAT1_WAKEUPENABLE	RESERVED	MMC1_DAT1_ACTIVE	MMC1_DAT1_PULLTYPESELECT	MMC1_DAT1_PULLUDENABLE	RESERVED	MMC1_DAT1_MODESELECT	MMC1_DAT1_DELAYMODE	MMC1_DAT1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	MMC1_DAT1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_DAT1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED	Reserved	R	0x0
18	MMC1_DAT1_ACTIVE	Controls enabling/disabling of the input buffer. 0x0: Input buffer is disabled 0x1: Input buffer is enabled	RW	0x1
17	MMC1_DAT1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC1_DAT1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED	Reserved	R	0x0
8	MMC1_DAT1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_DAT1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC1_DAT1_MUXMODE	0x0: mmc1_dat1 0xE: gpio6_24 0xF: Driver off	RW	0xF

Table 18-770. CTRL_CORE_PAD_MMC1_DAT2

Address Offset	0x0000 1764		
Physical Address	0x4A00 3764	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC1_DAT2_WAKEUPEVENT	MMC1_DAT2_WAKEUPENABLE	RESERVED	MMC1_DAT2_ACTIVE	MMC1_DAT2_PULLTYPESELECT	MMC1_DAT2_PULLUDENABLE	RESERVED	MMC1_DAT2_MODESELECT	MMC1_DAT2_DELAYMODE	MMC1_DAT2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC1_DAT2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_DAT2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED		R	0x0
18	MMC1_DAT2_ACTIVE	Controls enabling/disabling of the input buffer. 0x0: Input buffer is disabled 0x1: Input buffer is enabled	RW	0x1
17	MMC1_DAT2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC1_DAT2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC1_DAT2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_DAT2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC1_DAT2_MUXMODE	0x0: mmc1_dat2 0xE: gpio6_25 0xF: Driver off	RW	0xF

Table 18-771. CTRL_CORE_PAD_MMC1_DAT3

Address Offset	0x0000 1768		
Physical Address	0x4A00 3768	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC1_DAT3_WAKEUPEVENT	MMC1_DAT3_WAKEUPENABLE	RESERVED	MMC1_DAT3_ACTIVE	MMC1_DAT3_PULLTYPESELECT	MMC1_DAT3_PULLUDENABLE	RESERVED	MMC1_DAT3_MODESELECT	MMC1_DAT3_DELAYMODE	MMC1_DAT3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC1_DAT3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_DAT3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED		R	0x0
18	MMC1_DAT3_ACTIVE	Controls enabling/disabling of the input buffer. 0x0: Input buffer is disabled 0x1: Input buffer is enabled	RW	0x1
17	MMC1_DAT3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC1_DAT3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC1_DAT3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_DAT3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC1_DAT3_MUXMODE	0x0: mmc1_dat3 0xE: gpio6_26 0xF: Driver off	RW	0xF

Table 18-772. CTRL_CORE_PAD_MMC1_SDCD

Address Offset	0x0000 176C		
Physical Address	0x4A00 376C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC1_SDCD_WAKEUPEVENT	MMC1_SDCD_WAKEUPENABLE	RESERVED	MMC1_SDCD_SLEWCONTROL	MMC1_SDCD_INPUTENABLE	MMC1_SDCD_PULLTYPESELECT	MMC1_SDCD_PULLUDENABLE	RESERVED	MMC1_SDCD_MODESELECT	MMC1_SDCD_DELAYMODE	MMC1_SDCD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC1_SDCD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_SDCD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC1_SDCD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MMC1_SDCD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC1_SDCD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC1_SDCD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC1_SDCD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_SDCD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC1_SDCD_MUXMODE	0x0: mmc1_sdcd 0x3: uart6_rxd 0x4: i2c4_sda 0xE: gpio6_27 0xF: Driver off	RW	0xF

Table 18-773. CTRL_CORE_PAD_MMC1_SDWP

Address Offset	0x0000 1770		
Physical Address	0x4A00 3770	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MMC1_SDWP_WAKEUPEVENT	MMC1_SDWP_WAKEUPENABLE	RESERVED	MMC1_SDWP_SLEWCONTROL	MMC1_SDWP_INPUTENABLE	MMC1_SDWP_PULLTYPESELECT	MMC1_SDWP_PULLUDENABLE	RESERVED	MMC1_SDWP_MODESELECT	MMC1_SDWP_DELAYMODE	MMC1_SDWP_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC1_SDWP_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC1_SDWP_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC1_SDWP_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MMC1_SDWP_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC1_SDWP_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	MMC1_SDWP_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC1_SDWP_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC1_SDWP_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MMC1_SDWP_MUXMODE	0x0: mmc1_sdwp 0x3: uart6_txd 0x4: i2c4_scl 0xE: gpio6_28 0xF: Driver off	RW	0xF

Table 18-774. CTRL_CORE_PAD_GPIO6_10

Address Offset	0x0000 1774		
Physical Address	0x4A00 3774	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	GPIO6_10_WAKEUPEVENT	GPIO6_10_WAKEUPENABLE	RESERVED	GPIO6_10_SLEWCONTROL	GPIO6_10_INPUTENABLE	GPIO6_10_PULLTYPESELECT	GPIO6_10_PULLUDENABLE	RESERVED	GPIO6_10_MODESELECT	GPIO6_10_DELAYMODE	GPIO6_10_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPIO6_10_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPIO6_10_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	GPIO6_10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPIO6_10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPIO6_10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPIO6_10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	GPIO6_10_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPIO6_10_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPIO6_10_MUXMODE	0x0: gpio6_10 0x1: mdio_mclk 0x2: i2c3_sda 0x3: usb3_ulpi_d7 0x4: vin2b_hsync1 0x9: vin5a_clk0 0xA: ehrpwm2A 0xE: gpio6_10 0xF: Driver off	RW	0xF

Table 18-775. CTRL_CORE_PAD_GPIO6_11

Address Offset	0x0000 1778		
Physical Address	0x4A00 3778	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

Bits	Field
31:26	RESERVED
25	GPIO6_11_WAKEUPEVENT
24	GPIO6_11_WAKEUPENABLE
23:20	RESERVED
19	GPIO6_11_SLEWCONTROL
18	GPIO6_11_INPUTENABLE
17	GPIO6_11_PULLTYPESELECT
16	GPIO6_11_PULLUDENABLE
15:9	RESERVED
8	GPIO6_11_MODESELECT
7:4	GPIO6_11_DELAYMODE
3:0	GPIO6_11_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	GPIO6_11_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	GPIO6_11_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	GPIO6_11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPIO6_11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPIO6_11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPIO6_11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	GPIO6_11_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	GPIO6_11_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	GPIO6_11_MUXMODE	0x0: gpio6_11 0x1: mdio_d 0x2: i2c3_scl 0x3: usb3_ulpi_d6 0x4: vin2b_vsync1 0x9: vin5a_de0 0xA: ehrpwm2B 0xE: gpio6_11 0xF: Driver off	RW	0xF

Table 18-776. CTRL_CORE_PAD_MMC3_CLK

Address Offset	0x0000 177C		
Physical Address	0x4A00 377C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC3_CLK_WAKEUPEVENT	MMC3_CLK_WAKEUPENABLE	RESERVED	MMC3_CLK_SLEWCONTROL	MMC3_CLK_INPUTENABLE	MMC3_CLK_PULLTYPESELECT	MMC3_CLK_PULLUDENABLE	RESERVED	MMC3_CLK_MODESELECT	MMC3_CLK_DELAYMODE	MMC3_CLK_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_CLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_CLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC3_CLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_CLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_CLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_CLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC3_CLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_CLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC3_CLK_MUXMODE	0x0: mmc3_clk 0x3: usb3_ulpi_d5 0x4: vin2b_d7 0x9: vin5a_d7 0xA: ehrpwm2_tripzone_input 0xE: gpio6_29 0xF: Driver off	RW	0xF

Table 18-777. CTRL_CORE_PAD_MMC3_CMD

Address Offset	0x0000 1780		
Physical Address	0x4A00 3780	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MMC3_CMD_WAKEUPEVENT	MMC3_CMD_WAKEUPENABLE	RESERVED	MMC3_CMD_SLEWCONTROL	MMC3_CMD_INPUTENABLE	MMC3_CMD_PULLTYPESELECT	MMC3_CMD_PULLUDENABLE	RESERVED	MMC3_CMD_MODESELECT	MMC3_CMD_DELAYMODE	MMC3_CMD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_CMD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_CMD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC3_CMD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_CMD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_CMD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_CMD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC3_CMD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_CMD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MMC3_CMD_MUXMODE	0x0: mmc3_cmd 0x1: spi3_sclk 0x3: usb3_ulpi_d4 0x4: vin2b_d6 0x9: vin5a_d6 0xA: eCAP2_in_PWM2_out 0xE: gpio6_30 0xF: Driver off	RW	0xF

Table 18-778. CTRL_CORE_PAD_MMC3_DAT0

Address Offset	0x0000 1784		
Physical Address	0x4A00 3784	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MMC3_DAT0_WAKEUPEVENT	MMC3_DAT0_WAKEUPENABLE	RESERVED	MMC3_DAT0_SLEWCONTROL	MMC3_DAT0_INPUTENABLE	MMC3_DAT0_PULLTYPESELECT	MMC3_DAT0_PULLUDENABLE	RESERVED	MMC3_DAT0_MODESELECT	MMC3_DAT0_DELAYMODE	MMC3_DAT0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC3_DAT0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MMC3_DAT0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC3_DAT0_MUXMODE	0x0: mmc3_dat0 0x1: spi3_d1 0x2: uart5_rxd 0x3: usb3_ulpi_d3 0x4: vin2b_d5 0x9: vin5a_d5 0xA: eQEP3A_in 0xE: gpio6_31 0xF: Driver off	RW	0xF

Table 18-779. CTRL_CORE_PAD_MMC3_DAT1

Address Offset	0x0000 1788		
Physical Address	0x4A00 3788	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MMC3_DAT1_WAKEUPEVENT	MMC3_DAT1_WAKEUPENABLE	RESERVED	MMC3_DAT1_SLEWCONTROL	MMC3_DAT1_INPUTENABLE	MMC3_DAT1_PULLTYPESELECT	MMC3_DAT1_PULLUDENABLE	RESERVED	MMC3_DAT1_MODESELECT	MMC3_DAT1_DELAYMODE	MMC3_DAT1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MMC3_DAT1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC3_DAT1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC3_DAT1_MUXMODE	0x0: mmc3_dat1 0x1: spi3_d0 0x2: uart5_txd 0x3: usb3_ulpi_d2 0x4: vin2b_d4 0x9: vin5a_d4 0xA: eQEP3B_in 0xE: gpio7_0 0xF: Driver off	RW	0xF

Table 18-780. CTRL_CORE_PAD_MMC3_DAT2

Address Offset	0x0000 178C		
Physical Address	0x4A00 378C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC3_DAT2_WAKEUPEVENT	MMC3_DAT2_WAKEUPENABLE	RESERVED	MMC3_DAT2_SLEWCONTROL	MMC3_DAT2_INPUTENABLE	MMC3_DAT2_PULLTYPESELECT	MMC3_DAT2_PULLUDENABLE	RESERVED	MMC3_DAT2_MODESELECT	MMC3_DAT2_DELAYMODE	MMC3_DAT2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC3_DAT2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC3_DAT2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MMC3_DAT2_MUXMODE	0x0: mmc3_dat2 0x1: spi3_cs0 0x2: uart5_ctsn 0x3: usb3_ulpi_d1 0x4: vin2b_d3 0x9: vin5a_d3 0xA: eQEP3_index 0xE: gpio7_1 0xF: Driver off	RW	0xF

Table 18-781. CTRL_CORE_PAD_MMC3_DAT3

Address Offset	0x0000 1790		
Physical Address	0x4A00 3790	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MMC3_DAT3_WAKEUPEVENT	MMC3_DAT3_WAKEUPENABLE	RESERVED	MMC3_DAT3_SLEWCONTROL	MMC3_DAT3_INPUTENABLE	MMC3_DAT3_PULLTYPESELECT	MMC3_DAT3_PULLUDENABLE	RESERVED	MMC3_DAT3_MODESELECT	MMC3_DAT3_DELAYMODE	MMC3_DAT3_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC3_DAT3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MMC3_DAT3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC3_DAT3_MUXMODE	0x0: mmc3_dat3 0x1: spi3_cs1 0x2: uart5_rtsn 0x3: usb3_ulpi_d0 0x4: vin2b_d2 0x9: vin5a_d2 0xA: eQEP3_strobe 0xE: gpio7_2 0xF: Driver off	RW	0xF

Table 18-782. CTRL_CORE_PAD_MMC3_DAT4

Address Offset	0x0000 1794		
Physical Address	0x4A00 3794	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MMC3_DAT4_WAKEUPEVENT	MMC3_DAT4_WAKEUPENABLE	RESERVED	MMC3_DAT4_SLEWCONTROL	MMC3_DAT4_INPUTENABLE	MMC3_DAT4_PULLTYPESELECT	MMC3_DAT4_PULLUDENABLE	RESERVED	MMC3_DAT4_MODESELECT	MMC3_DAT4_DELAYMODE	MMC3_DAT4_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT4_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT4_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MMC3_DAT4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC3_DAT4_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT4_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC3_DAT4_MUXMODE	0x0: mmc3_dat4 0x1: spi4_sclk 0x2: uart10_rxd 0x3: usb3_ulpi_nxt 0x4: vin2b_d1 0x9: vin5a_d1 0xA: ehrpwm3A 0xE: gpio1_22 0xF: Driver off	RW	0xF

Table 18-783. CTRL_CORE_PAD_MMC3_DAT5

Address Offset	0x0000 1798		
Physical Address	0x4A00 3798	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MMC3_DAT5_WAKEUPEVENT	MMC3_DAT5_WAKEUPENABLE	RESERVED	MMC3_DAT5_SLEWCONTROL	MMC3_DAT5_INPUTENABLE	MMC3_DAT5_PULLTYPESELECT	MMC3_DAT5_PULLUDENABLE	RESERVED	MMC3_DAT5_MODESELECT	MMC3_DAT5_DELAYMODE	MMC3_DAT5_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT5_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT5_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC3_DAT5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC3_DAT5_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT5_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	MMC3_DAT5_MUXMODE	0x0: mmc3_dat5 0x1: spi4_d1 0x2: uart10_txd 0x3: usb3_ulpi_dir 0x4: vin2b_d0 0x9: vin5a_d0 0xA: ehrpwm3B 0xE: gpio1_23 0xF: Driver off	RW	0xF

Table 18-784. CTRL_CORE_PAD_MMC3_DAT6

Address Offset	0x0000 179C		
Physical Address	0x4A00 379C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	MMC3_DAT6_WAKEUPEVENT	MMC3_DAT6_WAKEUPENABLE	RESERVED	MMC3_DAT6_SLEWCONTROL	MMC3_DAT6_INPUTENABLE	MMC3_DAT6_PULLTYPESELECT	MMC3_DAT6_PULLUDENABLE	RESERVED	MMC3_DAT6_MODESELECT	MMC3_DAT6_DELAYMODE	MMC3_DAT6_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT6_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT6_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	MMC3_DAT6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	MMC3_DAT6_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT6_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC3_DAT6_MUXMODE	0x0: mmc3_dat6 0x1: spi4_d0 0x2: uart10_ctsn 0x3: usb3_ulpi_stp 0x4: vin2b_de1 0x9: vin5a_hsync0 0xA: ehrpwm3_tripzone_input 0xE: gpio1_24 0xF: Driver off	RW	0xF

Table 18-785. CTRL_CORE_PAD_MMC3_DAT7

Address Offset	0x0000 17A0		
Physical Address	0x4A00 37A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MMC3_DAT7_WAKEUPEVENT	MMC3_DAT7_WAKEUPENABLE	RESERVED				MMC3_DAT7_SLEWCONTROL	MMC3_DAT7_INPUTENABLE	MMC3_DAT7_PULLTYPESELECT	MMC3_DAT7_PULLUDENABLE	RESERVED							MMC3_DAT7_MODESELECT	MMC3_DAT7_DELAYMODE				MMC3_DAT7_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	MMC3_DAT7_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	MMC3_DAT7_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	MMC3_DAT7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	MMC3_DAT7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MMC3_DAT7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MMC3_DAT7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	MMC3_DAT7_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	MMC3_DAT7_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	MMC3_DAT7_MUXMODE	0x0: mmc3_dat7 0x1: spi4_cs0 0x2: uart10_rtsn 0x3: usb3_ulpi_clk 0x4: vin2b_clk1 0x9: vin5a_vsync0 0xA: eCAP3_in_PWM3_out 0xE: gpio1_25 0xF: Driver off	RW	0xF

Table 18-786. CTRL_CORE_PAD_SPI1_SCLK

Address Offset	0x0000 17A4		
Physical Address	0x4A00 37A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>SPI1_SCLK_WAKEUPEVENT</td><td>SPI1_SCLK_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>SPI1_SCLK_SLEWCONTROL</td><td>SPI1_SCLK_INPUTENABLE</td><td>SPI1_SCLK_PULLTYPESELECT</td><td>SPI1_SCLK_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>SPI1_SCLK_MODESELECT</td><td colspan="4">SPI1_SCLK_DELAYMODE</td><td colspan="4">SPI1_SCLK_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI1_SCLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI1_SCLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI1_SCLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_SCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_SCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI1_SCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI1_SCLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI1_SCLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI1_SCLK_MUXMODE	0x0: spi1_sclk 0xE: gpio7_7 0xF: Driver off	RW	0xF

Table 18-787. CTRL_CORE_PAD_SPI1_D1

Address Offset	0x0000 17A8		
Physical Address	0x4A00 37A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>SPI1_D1_WAKEUPEVENT</td><td>SPI1_D1_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>SPI1_D1_SLEWCONTROL</td><td>SPI1_D1_INPUTENABLE</td><td>SPI1_D1_PULLTYPESELECT</td><td>SPI1_D1_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>SPI1_D1_MODESELECT</td><td colspan="4">SPI1_D1_DELAYMODE</td><td colspan="4">SPI1_D1_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI1_D1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI1_D1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI1_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI1_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI1_D1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI1_D1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI1_D1_MUXMODE	0x0: spi1_d1 0xE: gpio7_8 0xF: Driver off	RW	0xF

Table 18-788. CTRL_CORE_PAD_SPI1_D0

Address Offset	0x0000 17AC		
Physical Address	0x4A00 37AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	SPI1_D0_WAKEUPEVENT	SPI1_D0_WAKEUPENABLE	RESERVED	SPI1_D0_SLEWCONTROL	SPI1_D0_INPUTENABLE	SPI1_D0_PULLTYPESELECT	SPI1_D0_PULLUDENABLE	RESERVED	SPI1_D0_MODESELECT	SPI1_D0_DELAYMODE	SPI1_D0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI1_D0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI1_D0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI1_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI1_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI1_D0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI1_D0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI1_D0_MUXMODE	0x0: spi1_d0 0xE: gpio7_9 0xF: Driver off	RW	0xF

Table 18-789. CTRL_CORE_PAD_SPI1_CS0

Address Offset	0x0000 17B0		
Physical Address	0x4A00 37B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SPI1_CS0_WAKEUPEVENT	SPI1_CS0_WAKEUPENABLE	RESERVED	SPI1_CS0_SLEWCONTROL	SPI1_CS0_INPUTENABLE	SPI1_CS0_PULLTYPESELECT	SPI1_CS0_PULLUDENABLE	RESERVED	SPI1_CS0_MODESELECT	SPI1_CS0_DELAYMODE	SPI1_CS0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI1_CS0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI1_CS0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI1_CS0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_CS0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_CS0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI1_CS0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI1_CS0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI1_CS0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI1_CS0_MUXMODE	0x0: spi1_cs0 0xE: gpio7_10 0xF: Driver off	RW	0xF

Table 18-790. CTRL_CORE_PAD_SPI1_CS1

Address Offset	0x0000 17B4		
Physical Address	0x4A00 37B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						SPI1_CS1_WAKEUPEVENT	SPI1_CS1_WAKEUPENABLE	RESERVED				SPI1_CS1_SLEWCONTROL	SPI1_CS1_INPUTENABLE	SPI1_CS1_PULLTYPESELECT	SPI1_CS1_PULLUDENABLE	RESERVED							SPI1_CS1_MODESELECT	SPI1_CS1_DELAYMODE				SPI1_CS1_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI1_CS1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI1_CS1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI1_CS1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_CS1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_CS1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI1_CS1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI1_CS1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI1_CS1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI1_CS1_MUXMODE	0x0: spi1_cs1 0x2: sata1_led 0x3: spi2_cs1 0xE: gpio7_11 0xF: Driver off	RW	0xF

Table 18-791. CTRL_CORE_PAD_SPI1_CS2

Address Offset	0x0000 17B8		
Physical Address	0x4A00 37B8	**Instance**	CTRL_MODULE_CORE

Table 18-791. CTRL_CORE_PAD_SPI1_CS2 (continued)

Description	
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>SPI1_CS2_WAKEUPEVENT</td><td>SPI1_CS2_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>SPI1_CS2_SLEWCONTROL</td><td>SPI1_CS2_INPUTENABLE</td><td>SPI1_CS2_PULLTYPESELECT</td><td>SPI1_CS2_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>SPI1_CS2_MODESELECT</td><td colspan="4">SPI1_CS2_DELAYMODE</td><td colspan="4">SPI1_CS2_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI1_CS2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI1_CS2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI1_CS2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI1_CS2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_CS2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI1_CS2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI1_CS2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI1_CS2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	SPI1_CS2_MUXMODE	0x0: spi1_cs2 0x1: uart4_rxd 0x2: mmc3_sdcd 0x3: spi2_cs2 0x4: dcan2_tx 0x5: mdio_mclk 0x6: hdmi1_hpd 0xE: gpio7_12 0xF: Driver off	RW	0xF

Table 18-792. CTRL_CORE_PAD_SPI1_CS3

Address Offset	0x0000 17BC		
Physical Address	0x4A00 37BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						SPI1_CS3_WAKEUPEVENT	SPI1_CS3_WAKEUPENABLE	RESERVED				SPI1_CS3_SLEWCONTROL	SPI1_CS3_INPUTENABLE	SPI1_CS3_PULLTYPESELECT	SPI1_CS3_PULLUDENABLE	RESERVED							SPI1_CS3_MODESELECT	SPI1_CS3_DELAYMODE				SPI1_CS3_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI1_CS3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI1_CS3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI1_CS3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI1_CS3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_CS3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI1_CS3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	SPI1_CS3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI1_CS3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI1_CS3_MUXMODE	0x0: spi1_cs3 0x1: uart4_txd 0x2: mmc3_sdwp 0x3: spi2_cs3 0x4: dcan2_rx 0x5: mdio_d 0x6: hdmi1_cec 0xE: gpio7_13 0xF: Driver off	RW	0xF

Table 18-793. CTRL_CORE_PAD_SPI2_SCLK

Address Offset	0x0000 17C0		
Physical Address	0x4A00 37C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	SPI2_SCLK_WAKEUPEVENT	SPI2_SCLK_WAKEUPENABLE	RESERVED	SPI2_SCLK_SLEWCONTROL	SPI2_SCLK_INPUTENABLE	SPI2_SCLK_PULLTYPESELECT	SPI2_SCLK_PULLUDENABLE	RESERVED	SPI2_SCLK_MODESELECT	SPI2_SCLK_DELAYMODE	SPI2_SCLK_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI2_SCLK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI2_SCLK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI2_SCLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	SPI2_SCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_SCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI2_SCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI2_SCLK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI2_SCLK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI2_SCLK_MUXMODE	0x0: spi2_sclk 0x1: uart3_rxd 0xE: gpio7_14 0xF: Driver off	RW	0xF

Table 18-794. CTRL_CORE_PAD_SPI2_D1

Address Offset	0x0000 17C4		
Physical Address	0x4A00 37C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	SPI2_D1_WAKEUPEVENT	SPI2_D1_WAKEUPENABLE	RESERVED	SPI2_D1_SLEWCONTROL	SPI2_D1_INPUTENABLE	SPI2_D1_PULLTYPESELECT	SPI2_D1_PULLUDENABLE	RESERVED	SPI2_D1_MODESELECT	SPI2_D1_DELAYMODE	SPI2_D1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI2_D1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI2_D1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	SPI2_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI2_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI2_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI2_D1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI2_D1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI2_D1_MUXMODE	0x0: spi2_d1 0x1: uart3_txd 0xE: gpio7_15 0xF: Driver off	RW	0xF

Table 18-795. CTRL_CORE_PAD_SPI2_D0

Address Offset	0x0000 17C8		
Physical Address	0x4A00 37C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	SPI2_D0_WAKEUPEVENT	SPI2_D0_WAKEUPENABLE	RESERVED	SPI2_D0_SLEWCONTROL	SPI2_D0_INPUTENABLE	SPI2_D0_PULLTYPESELECT	SPI2_D0_PULLUDENABLE	RESERVED	SPI2_D0_MODESELECT	SPI2_D0_DELAYMODE	SPI2_D0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI2_D0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0

Bits	Field Name	Description	Type	Reset
24	SPI2_D0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI2_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI2_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI2_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI2_D0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI2_D0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI2_D0_MUXMODE	0x0: spi2_d0 0x1: uart3_ctsn 0x2: uart5_rxd 0xE: gpio7_16 0xF: Driver off	RW	0xF

Table 18-796. CTRL_CORE_PAD_SPI2_CS0

Address Offset	0x0000 17CC		
Physical Address	0x4A00 37CC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>SPI2_CS0_WAKEUPEVENT</td><td>SPI2_CS0_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>SPI2_CS0_SLEWCONTROL</td><td>SPI2_CS0_INPUTENABLE</td><td>SPI2_CS0_PULLTYPESELECT</td><td>SPI2_CS0_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>SPI2_CS0_MODESELECT</td><td colspan="4">SPI2_CS0_DELAYMODE</td><td colspan="4">SPI2_CS0_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	SPI2_CS0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	SPI2_CS0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	SPI2_CS0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI2_CS0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_CS0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI2_CS0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	SPI2_CS0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	SPI2_CS0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	SPI2_CS0_MUXMODE	0x0: spi2_cs0 0x1: uart3_rtsn 0x2: uart5_txd 0xE: gpio7_17 0xF: Driver off	RW	0xF

Table 18-797. CTRL_CORE_PAD_DCAN1_TX

Address Offset	0x0000 17D0		
Physical Address	0x4A00 37D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DCAN1_TX_WAKEUPEVENT	DCAN1_TX_WAKEUPENABLE	RESERVED	DCAN1_TX_SLEWCONTROL	DCAN1_TX_INPUTENABLE	DCAN1_TX_PULLTYPESELECT	DCAN1_TX_PULLUDENABLE	RESERVED	DCAN1_TX_MODESELECT	DCAN1_TX_DELAYMODE	DCAN1_TX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	DCAN1_TX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	DCAN1_TX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	DCAN1_TX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	DCAN1_TX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	DCAN1_TX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	DCAN1_TX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	DCAN1_TX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	DCAN1_TX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	DCAN1_TX_MUXMODE	0x0: dcan1_tx 0x2: uart8_rxd 0x3: mmc2_sdcd 0x6: hdmi1_hpd 0xE: gpio1_14 0xF: Driver off	RW	0xF

Table 18-798. CTRL_CORE_PAD_DCAN1_RX

Address Offset	0x0000 17D4

Table 18-798. CTRL_CORE_PAD_DCAN1_RX (continued)

Physical Address	0x4A00 37D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	DCAN1_RX_WAKEUPEVENT	DCAN1_RX_WAKEUPENABLE	RESERVED	DCAN1_RX_SLEWCONTROL	DCAN1_RX_INPUTENABLE	DCAN1_RX_PULLTYPESELECT	DCAN1_RX_PULLUDENABLE	RESERVED	DCAN1_RX_MODESELECT	DCAN1_RX_DELAYMODE	DCAN1_RX_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	DCAN1_RX_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	DCAN1_RX_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	DCAN1_RX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	DCAN1_RX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	DCAN1_RX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	DCAN1_RX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	DCAN1_RX_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	DCAN1_RX_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	DCAN1_RX_MUXMODE	0x0: dcan1_rx 0x2: uart8_txd 0x3: mmc2_sdwp 0x4: sata1_led 0x6: hdmi1_cec 0xE: gpio1_15 0xF: Driver off	RW	0xF

Table 18-799. CTRL_CORE_PAD_UART1_RXD

Address Offset	0x0000 17E0		
Physical Address	0x4A00 37E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	UART1_RXD_WAKEUPEVENT	UART1_RXD_WAKEUPENABLE	RESERVED	UART1_RXD_SLEWCONTROL	UART1_RXD_INPUTENABLE	UART1_RXD_PULLTYPESELECT	UART1_RXD_PULLUDENABLE	RESERVED	UART1_RXD_MODESELECT	UART1_RXD_DELAYMODE	UART1_RXD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART1_RXD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART1_RXD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART1_RXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	UART1_RXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART1_RXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART1_RXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	UART1_RXD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART1_RXD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART1_RXD_MUXMODE	0x0: uart1_rxd 0x3: mmc4_sdcd 0xE: gpio7_22 0xF: Driver off	RW	0xF

Table 18-800. CTRL_CORE_PAD_UART1_TXD

Address Offset	0x0000 17E4		
Physical Address	0x4A00 37E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	UART1_TXD_WAKEUPEVENT	UART1_TXD_WAKEUPENABLE	RESERVED	UART1_TXD_SLEWCONTROL	UART1_TXD_INPUTENABLE	UART1_TXD_PULLTYPESELECT	UART1_TXD_PULLUDENABLE	RESERVED	UART1_TXD_MODESELECT	UART1_TXD_DELAYMODE	UART1_TXD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART1_TXD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART1_TXD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART1_TXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	UART1_TXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	UART1_TXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART1_TXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART1_TXD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART1_TXD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART1_TXD_MUXMODE	0x0: uart1_txd 0x3: mmc4_sdwp 0xE: gpio7_23 0xF: Driver off	RW	0xF

Table 18-801. CTRL_CORE_PAD_UART1_CTSN

Address Offset	0x0000 17E8		
Physical Address	0x4A00 37E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	UART1_CTSN_WAKEUPEVENT	UART1_CTSN_WAKEUPENABLE	RESERVED	UART1_CTSN_SLEWCONTROL	UART1_CTSN_INPUTENABLE	UART1_CTSN_PULLTYPESELECT	UART1_CTSN_PULLUDENABLE	RESERVED	UART1_CTSN_MODESELECT	UART1_CTSN_DELAYMODE	UART1_CTSN_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART1_CTSN_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART1_CTSN_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	UART1_CTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART1_CTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART1_CTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART1_CTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART1_CTSN_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART1_CTSN_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART1_CTSN_MUXMODE	0x0: uart1_ctsn 0x2: uart9_rxd 0x3: mmc4_clk 0xE: gpio7_24 0xF: Driver off	RW	0xF

Table 18-802. CTRL_CORE_PAD_UART1_RTSN

Address Offset	0x0000 17EC		
Physical Address	0x4A00 37EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	UART1_RTSN_WAKEUPEVENT	UART1_RTSN_WAKEUPENABLE	RESERVED	UART1_RTSN_SLEWCONTROL	UART1_RTSN_INPUTENABLE	UART1_RTSN_PULLTYPESELECT	UART1_RTSN_PULLUDENABLE	RESERVED	UART1_RTSN_MODESELECT	UART1_RTSN_DELAYMODE	UART1_RTSN_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	UART1_RTSN_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART1_RTSN_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART1_RTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART1_RTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART1_RTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART1_RTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART1_RTSN_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART1_RTSN_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART1_RTSN_MUXMODE	0x0: uart1_rtsn 0x2: uart9_txd 0x3: mmc4_cmd 0xE: gpio7_25 0xF: Driver off	RW	0xF

Table 18-803. CTRL_CORE_PAD_UART2_RXD

Address Offset	0x0000 17F0		
Physical Address	0x4A00 37F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	UART2_RXD_WAKEUPEVENT	UART2_RXD_WAKEUPENABLE	RESERVED	UART2_RXD_SLEWCONTROL	UART2_RXD_INPUTENABLE	UART2_RXD_PULLTYPESELECT	UART2_RXD_PULLUDENABLE	RESERVED	UART2_RXD_MODESELECT	UART2_RXD_DELAYMODE	UART2_RXD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART2_RXD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART2_RXD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART2_RXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART2_RXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART2_RXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART2_RXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART2_RXD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART2_RXD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART2_RXD_MUXMODE	0x0: Reserved 0x1: uart3_ctsn 0x2: uart3_rctx 0x3: mmc4_dat0 0x4: uart2_rxd 0x5: uart1_dcdn 0xE: gpio7_26 0xF: Driver off	RW	0xF

Table 18-804. CTRL_CORE_PAD_UART2_TXD

Address Offset	0x0000 17F4		
Physical Address	0x4A00 37F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	UART2_TXD_WAKEUPEVENT	UART2_TXD_WAKEUPENABLE	RESERVED	UART2_TXD_SLEWCONTROL	UART2_TXD_INPUTENABLE	UART2_TXD_PULLTYPESELECT	UART2_TXD_PULLUDENABLE	RESERVED	UART2_TXD_MODESELECT	UART2_TXD_DELAYMODE	UART2_TXD_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART2_TXD_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART2_TXD_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART2_TXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART2_TXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART2_TXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART2_TXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART2_TXD_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART2_TXD_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	UART2_TXD_MUXMODE	0x0: uart2_txd 0x1: uart3_rtsn 0x2: uart3_sd 0x3: mmc4_dat1 0x4: uart2_txd 0x5: uart1_dsrn 0xE: gpio7_27 0xF: Driver off	RW	0xF

Table 18-805. CTRL_CORE_PAD_UART2_CTSN

Address Offset	0x0000 17F8		
Physical Address	0x4A00 37F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	UART2_CTSN_WAKEUPEVENT	UART2_CTSN_WAKEUPENABLE	RESERVED	UART2_CTSN_SLEWCONTROL	UART2_CTSN_INPUTENABLE	UART2_CTSN_PULLTYPESELECT	UART2_CTSN_PULLUDENABLE	RESERVED	UART2_CTSN_MODESELECT	UART2_CTSN_DELAYMODE	UART2_CTSN_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART2_CTSN_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART2_CTSN_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART2_CTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART2_CTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART2_CTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART2_CTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	UART2_CTSN_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART2_CTSN_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART2_CTSN_MUXMODE	0x0: uart2_ctsn 0x2: uart3_rxd 0x3: mmc4_dat2 0x4: uart10_rxd 0x5: uart1_dtrn 0xE: gpio1_16 0xF: Driver off	RW	0xF

Table 18-806. CTRL_CORE_PAD_UART2_RTSN

Address Offset	0x0000 17FC		
Physical Address	0x4A00 37FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31–26	25	24	23–20	19	18	17	16	15–9	8	7–4	3–0
RESERVED	UART2_RTSN_WAKEUPEVENT	UART2_RTSN_WAKEUPENABLE	RESERVED	UART2_RTSN_SLEWCONTROL	UART2_RTSN_INPUTENABLE	UART2_RTSN_PULLTYPESELECT	UART2_RTSN_PULLUDENABLE	RESERVED	UART2_RTSN_MODESELECT	UART2_RTSN_DELAYMODE	UART2_RTSN_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	UART2_RTSN_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	UART2_RTSN_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	UART2_RTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	UART2_RTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART2_RTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART2_RTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	UART2_RTSN_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	UART2_RTSN_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	UART2_RTSN_MUXMODE	0x0: uart2_rtsn 0x1: uart3_txd 0x2: uart3_irtx 0x3: mmc4_dat3 0x4: uart10_txd 0x5: uart1_rin 0xE: gpio1_17 0xF: Driver off	RW	0xF

Table 18-807. CTRL_CORE_PAD_I2C1_SDA

Address Offset	0x0000 1800		
Physical Address	0x4A00 3800	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						I2C1_SDA_WAKEUPEVENT	I2C1_SDA_WAKEUPENABLE	RESERVED					I2C1_SDA_INPUTENABLE	I2C1_SDA_PULLTYPESELECT	I2C1_SDA_PULLUDENABLE	RESERVED															

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	I2C1_SDA_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	I2C1_SDA_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED		R	0x0
18	I2C1_SDA_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	I2C1_SDA_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	I2C1_SDA_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-808. CTRL_CORE_PAD_I2C1_SCL

Address Offset	0x0000 1804		
Physical Address	0x4A00 3804	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:19	18	17	16	15:0
RESERVED	I2C1_SCL_WAKEUPEVENT	I2C1_SCL_WAKEUPENABLE	RESERVED	I2C1_SCL_INPUTENABLE	I2C1_SCL_PULLTYPESELECT	I2C1_SCL_PULLUDENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	I2C1_SCL_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	I2C1_SCL_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED		R	0x0
18	I2C1_SCL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	I2C1_SCL_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	I2C1_SCL_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-809. CTRL_CORE_PAD_I2C2_SDA

Address Offset	0x0000 1808		
Physical Address	0x4A00 3808	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:19	18	17	16	15:4	3:0
RESERVED	I2C2_SDA_WAKEUPEVENT	I2C2_SDA_WAKEUPENABLE	RESERVED	I2C2_SDA_INPUTENABLE	I2C2_SDA_PULLTYPESELECT	I2C2_SDA_PULLUDENABLE	RESERVED	I2C2_SDA_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	I2C2_SDA_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	I2C2_SDA_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED		R	0x0
18	I2C2_SDA_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	I2C2_SDA_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	I2C2_SDA_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	I2C2_SDA_MUXMODE	0x0: i2c2_sda 0x1: hdmi1_ddc_scl 0xF: Driver off	RW	0xF

Table 18-810. CTRL_CORE_PAD_I2C2_SCL

Address Offset	0x0000 180C		
Physical Address	0x4A00 380C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	I2C2_SCL_WAKEUPEVENT	I2C2_SCL_WAKEUPENABLE	RESERVED	I2C2_SCL_INPUTENABLE	I2C2_SCL_PULLTYPESELECT	I2C2_SCL_PULLUDENABLE	RESERVED	I2C2_SCL_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	I2C2_SCL_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	I2C2_SCL_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:19	RESERVED		R	0x0
18	I2C2_SCL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	I2C2_SCL_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	I2C2_SCL_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	I2C2_SCL_MUXMODE	0x0: i2c2_scl 0x1: hdmi1_ddc_sda 0xF: Driver off	RW	0xF

Table 18-811. CTRL_CORE_PAD_WAKEUP0

Address Offset	0x0000 1818		
Physical Address	0x4A00 3818	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:18	17	16	15:9	8	7:4	3:0
RESERVED	WAKEUP0_WAKEUPEVENT	WAKEUP0_WAKEUPENABLE	RESERVED	WAKEUP0_PULLTYPESELECT	WAKEUP0_PULLUDENABLE	RESERVED	WAKEUP0_MODESELECT	WAKEUP0_DELAYMODE	WAKEUP0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	WAKEUP0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	WAKEUP0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:18	RESERVED		R	0x0
17	WAKEUP0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	WAKEUP0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	WAKEUP0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	WAKEUP0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	WAKEUP0_MUXMODE	0x0: Wakeup0 0x1: dcan1_rx 0xE: gpio1_0 0xF: Driver off	RW	0xF

Table 18-812. CTRL_CORE_PAD_WAKEUP1

Address Offset	0x0000 181C		
Physical Address	0x4A00 381C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26	25	24	23 22 21 20 19 18	17	16	15 14 13 12 11 10 9	8	7 6 5 4	3 2 1 0
RESERVED	WAKEUP1_WAKEUPEVENT	WAKEUP1_WAKEUPENABLE	RESERVED	WAKEUP1_PULLTYPESELECT	WAKEUP1_PULLUDENABLE	RESERVED	WAKEUP1_MODESELECT	WAKEUP1_DELAYMODE	WAKEUP1_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	WAKEUP1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	WAKEUP1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:18	RESERVED		R	0x0
17	WAKEUP1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	WAKEUP1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	WAKEUP1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	WAKEUP1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	WAKEUP1_MUXMODE	0x0: Wakeup1 0x1: dcan2_rx 0xE: gpio1_1 0xF: Driver off	RW	0xF

Table 18-813. CTRL_CORE_PAD_WAKEUP2

Address Offset	0x0000 1820		
Physical Address	0x4A00 3820	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:18	17	16	15:9	8	7:4	3:0
RESERVED	WAKEUP2_WAKEUPEVENT	WAKEUP2_WAKEUPENABLE	RESERVED	WAKEUP2_PULLTYPESELECT	WAKEUP2_PULLUDENABLE	RESERVED	WAKEUP2_MODESELECT	WAKEUP2_DELAYMODE	WAKEUP2_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	WAKEUP2_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	WAKEUP2_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:18	RESERVED		R	0x0
17	WAKEUP2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	WAKEUP2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	WAKEUP2_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	WAKEUP2_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	WAKEUP2_MUXMODE	0x0: Wakeup2 0x1: sys_nirq2 0xE: gpio1_2 0xF: Driver off	RW	0xF

Table 18-814. CTRL_CORE_PAD_WAKEUP3

Address Offset	0x0000 1824		
Physical Address	0x4A00 3824	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>WAKEUP3_WAKEUPEVENT</td><td>WAKEUP3_WAKEUPENABLE</td><td colspan="6">RESERVED</td><td>WAKEUP3_PULLTYPESELECT</td><td>WAKEUP3_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>WAKEUP3_MODESELECT</td><td colspan="4">WAKEUP3_DELAYMODE</td><td colspan="4">WAKEUP3_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
25	WAKEUP3_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	WAKEUP3_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:18	RESERVED		R	0x0
17	WAKEUP3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	WAKEUP3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	WAKEUP3_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	WAKEUP3_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	WAKEUP3_MUXMODE	0x0: Wakeup3 0x1: sys_nirq1 0xE: gpio1_3 0xF: Driver off	RW	0xF

Table 18-815. CTRL_CORE_PAD_ON_OFF

Address Offset	0x0000 1828		
Physical Address	0x4A00 3828	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:18	17	16	15:0
RESERVED	ON_OFF_PULLTYPESELECT	ON_OFF_PULLUDENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	ON_OFF_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1

Bits	Field Name	Description	Type	Reset
16	ON_OFF_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-816. CTRL_CORE_PAD_RTC_PORZ

Address Offset	0x0000 182C		
Physical Address	0x4A00 382C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														RTC_PORZ_PULLTYPESELECT	RTC_PORZ_PULLUDENABLE	RESERVED															

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	RTC_PORZ_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RTC_PORZ_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:0	RESERVED		R	0x0

Table 18-817. CTRL_CORE_PAD_TMS

Address Offset	0x0000 1830		
Physical Address	0x4A00 3830	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												TMS_SLEWCONTROL	TMS_INPUTENABLE	TMS_PULLTYPESELECT	TMS_PULLUDENABLE	RESERVED															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	TMS_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	TMS_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TMS_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	TMS_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-818. CTRL_CORE_PAD_TDI

Address Offset	0x0000 1834		
Physical Address	0x4A00 3834	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	TDI_WAKEUPEVENT	TDI_WAKEUPENABLE	RESERVED	TDI_SLEWCONTROL	TDI_INPUTENABLE	TDI_PULLTYPESELECT	TDI_PULLUDENABLE	RESERVED	TDI_MODESELECT	TDI_DELAYMODE	TDI_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	TDI_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	TDI_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	TDI_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	TDI_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TDI_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	TDI_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	TDI_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	TDI_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	TDI_MUXMODE	0x0: tdi 0xE: gpio8_27	RW	0x0

Table 18-819. CTRL_CORE_PAD_TDO

Address Offset	0x0000 1838		
Physical Address	0x4A00 3838	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	TDO_WAKEUPEVENT	TDO_WAKEUPENABLE	RESERVED	TDO_SLEWCONTROL	TDO_INPUTENABLE	TDO_PULLTYPESELECT	TDO_PULLUDENABLE	RESERVED	TDO_MODESELECT	TDO_DELAYMODE	TDO_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	TDO_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	TDO_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	TDO_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	TDO_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TDO_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	TDO_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	TDO_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	TDO_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	TDO_MUXMODE	0x0: tdo 0xE: gpio8_28	RW	0x0

Table 18-820. CTRL_CORE_PAD_TCLK

Address Offset	0x0000 183C		
Physical Address	0x4A00 383C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:19	18	17	16	15:0
RESERVED	TCLK_INPUTENABLE	TCLK_PULLTYPESELECT	TCLK_PULLUDENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	TCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	TCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-821. CTRL_CORE_PAD_TRSTN

Address Offset	0x0000 1840		
Physical Address	0x4A00 3840	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	TRSTN_SLEWCONTROL	TRSTN_INPUTENABLE	TRSTN_PULLTYPESELECT	TRSTN_PULLUDENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	TRSTN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	TRSTN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TRSTN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	TRSTN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-822. CTRL_CORE_PAD_RTCK

Address Offset	0x0000 1844		
Physical Address	0x4A00 3844	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>RTCK_WAKEUPEVENT</td><td>RTCK_WAKEUPENABLE</td><td colspan="4">RESERVED</td><td>RTCK_SLEWCONTROL</td><td>RTCK_INPUTENABLE</td><td>RTCK_PULLTYPESELECT</td><td>RTCK_PULLUDENABLE</td><td colspan="7">RESERVED</td><td>RTCK_MODESELECT</td><td colspan="4">RTCK_DELAYMODE</td><td colspan="4">RTCK_MUXMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	RTCK_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	RTCK_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	RTCK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	RTCK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RTCK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RTCK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:9	RESERVED		R	0x0
8	RTCK_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	RTCK_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	RTCK_MUXMODE	0x0: rtck 0xE: gpio8_29	RW	0x0

Table 18-823. CTRL_CORE_PAD_EMU0

Address Offset	0x0000 1848		
Physical Address	0x4A00 3848	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:26	25	24	23:20	19	18	17	16	15:9	8	7:4	3:0
RESERVED	EMU0_WAKEUPEVENT	EMU0_WAKEUPENABLE	RESERVED	EMU0_SLEWCONTROL	EMU0_INPUTENABLE	EMU0_PULLTYPESELECT	EMU0_PULLUDENABLE	RESERVED	EMU0_MODESELECT	EMU0_DELAYMODE	EMU0_MUXMODE

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	EMU0_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	EMU0_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	EMU0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	EMU0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	EMU0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	EMU0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	EMU0_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	EMU0_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	EMU0_MUXMODE	0x0: emu0 0xE: gpio8_30	RW	0x0

Table 18-824. CTRL_CORE_PAD_EMU1

Address Offset	0x0000 184C		
Physical Address	0x4A00 384C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						EMU1_WAKEUPEVENT	EMU1_WAKEUPENABLE	RESERVED				EMU1_SLEWCONTROL	EMU1_INPUTENABLE	EMU1_PULLTYPESELECT	EMU1_PULLUDENABLE	RESERVED							EMU1_MODESELECT	EMU1_DELAYMODE				EMU1_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	EMU1_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	EMU1_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:20	RESERVED		R	0x0
19	EMU1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	EMU1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	EMU1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	EMU1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:9	RESERVED		R	0x0
8	EMU1_MODESELECT	Selects between the Default IO Timing Mode and a Virtual or Manual IO Timing Mode. Refer to the device Data Manual for definition of the required settings for a given mode of operation. When this bit is 0b1, a Virtual IO Timing Mode can be selected via the DELAYMODE field of this register, as described in Section 18.4.6.1.5, Virtual IO Timing Modes. Manual IO Timing Modes are selected via the procedure described in Section 18.4.6.1.6, Manual IO Timing Modes. 0x0: Default IO Timing Mode is used 0x1: A Virtual or Manual IO Timing Mode is used	RW	0x0
7:4	EMU1_DELAYMODE	This bit field selects the Virtual Timing Mode used when the MODESELECT bit is set to 0b1. See Section 18.4.6.1.5, Virtual IO Timing Modes for details.	RW	0x0
3:0	EMU1_MUXMODE	0x0: emu1 0xE: gpio8_31	RW	0x0

Table 18-825. CTRL_CORE_PAD_RESETN

Address Offset	0x0000 185C		
Physical Address	0x4A00 385C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:18	17	16	15:0
RESERVED	RESETN_PULLTYPESELECT	RESETN_PULLUDENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	RESETN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	RESETN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-826. CTRL_CORE_PAD_NMIN_DSP

Address Offset	0x0000 1860		
Physical Address	0x4A00 3860	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>NMIN_WAKEUPEVENT</td><td>NMIN_WAKEUPENABLE</td><td colspan="6">RESERVED</td><td>NMIN_PULLTYPESELECT</td><td>NMIN_PULLUDENABLE</td><td colspan="16">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	NMIN_WAKEUPEVENT	0x0: No wakeup event detected 0x1: Wakeup event detected	R	0x0
24	NMIN_WAKEUPENABLE	0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
23:18	RESERVED		R	0x0
17	NMIN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	NMIN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-827. CTRL_CORE_PAD_RSTOUTN

Address Offset	0x0000 1864		
Physical Address	0x4A00 3864	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td>RSTOUTN_PULLTYPESELECT</td><td>RSTOUTN_PULLUDENABLE</td><td colspan="16">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	RSTOUTN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
16	RSTOUTN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 18-828. CTRL_CORE_PADCONF_WAKEUPEVENT_0

Address Offset	0x0000 1868		
Physical Address	0x4A00 3868	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_A15_DUPLICATEWAKEUPEVENT	GPMC_A14_DUPLICATEWAKEUPEVENT	GPMC_A13_DUPLICATEWAKEUPEVENT	GPMC_A12_DUPLICATEWAKEUPEVENT	GPMC_A11_DUPLICATEWAKEUPEVENT	GPMC_A10_DUPLICATEWAKEUPEVENT	GPMC_A9_DUPLICATEWAKEUPEVENT	GPMC_A8_DUPLICATEWAKEUPEVENT	GPMC_A7_DUPLICATEWAKEUPEVENT	GPMC_A6_DUPLICATEWAKEUPEVENT	GPMC_A5_DUPLICATEWAKEUPEVENT	GPMC_A4_DUPLICATEWAKEUPEVENT	GPMC_A3_DUPLICATEWAKEUPEVENT	GPMC_A2_DUPLICATEWAKEUPEVENT	GPMC_A1_DUPLICATEWAKEUPEVENT	GPMC_A0_DUPLICATEWAKEUPEVENT	GPMC_AD15_DUPLICATEWAKEUPEVENT	GPMC_AD14_DUPLICATEWAKEUPEVENT	GPMC_AD13_DUPLICATEWAKEUPEVENT	GPMC_AD12_DUPLICATEWAKEUPEVENT	GPMC_AD11_DUPLICATEWAKEUPEVENT	GPMC_AD10_DUPLICATEWAKEUPEVENT	GPMC_AD9_DUPLICATEWAKEUPEVENT	GPMC_AD8_DUPLICATEWAKEUPEVENT	GPMC_AD7_DUPLICATEWAKEUPEVENT	GPMC_AD6_DUPLICATEWAKEUPEVENT	GPMC_AD5_DUPLICATEWAKEUPEVENT	GPMC_AD4_DUPLICATEWAKEUPEVENT	GPMC_AD3_DUPLICATEWAKEUPEVENT	GPMC_AD2_DUPLICATEWAKEUPEVENT	GPMC_AD1_DUPLICATEWAKEUPEVENT	GPMC_AD0_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	GPMC_A15_DUPLICATE WAKEUPEVENT		R	0x0
30	GPMC_A14_DUPLICATE WAKEUPEVENT		R	0x0
29	GPMC_A13_DUPLICATE WAKEUPEVENT		R	0x0
28	GPMC_A12_DUPLICATE WAKEUPEVENT		R	0x0
27	GPMC_A11_DUPLICATE WAKEUPEVENT		R	0x0
26	GPMC_A10_DUPLICATE WAKEUPEVENT		R	0x0
25	GPMC_A9_DUPLICATEW AKEUPEVENT		R	0x0
24	GPMC_A8_DUPLICATEW AKEUPEVENT		R	0x0
23	GPMC_A7_DUPLICATEW AKEUPEVENT		R	0x0
22	GPMC_A6_DUPLICATEW AKEUPEVENT		R	0x0
21	GPMC_A5_DUPLICATEW AKEUPEVENT		R	0x0
20	GPMC_A4_DUPLICATEW AKEUPEVENT		R	0x0
19	GPMC_A3_DUPLICATEW AKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
18	GPMC_A2_DUPLICATEWAKEUPEVENT		R	0x0
17	GPMC_A1_DUPLICATEWAKEUPEVENT		R	0x0
16	GPMC_A0_DUPLICATEWAKEUPEVENT		R	0x0
15	GPMC_AD15_DUPLICATEWAKEUPEVENT		R	0x0
14	GPMC_AD14_DUPLICATEWAKEUPEVENT		R	0x0
13	GPMC_AD13_DUPLICATEWAKEUPEVENT		R	0x0
12	GPMC_AD12_DUPLICATEWAKEUPEVENT		R	0x0
11	GPMC_AD11_DUPLICATEWAKEUPEVENT		R	0x0
10	GPMC_AD10_DUPLICATEWAKEUPEVENT		R	0x0
9	GPMC_AD9_DUPLICATEWAKEUPEVENT		R	0x0
8	GPMC_AD8_DUPLICATEWAKEUPEVENT		R	0x0
7	GPMC_AD7_DUPLICATEWAKEUPEVENT		R	0x0
6	GPMC_AD6_DUPLICATEWAKEUPEVENT		R	0x0
5	GPMC_AD5_DUPLICATEWAKEUPEVENT		R	0x0
4	GPMC_AD4_DUPLICATEWAKEUPEVENT		R	0x0
3	GPMC_AD3_DUPLICATEWAKEUPEVENT		R	0x0
2	GPMC_AD2_DUPLICATEWAKEUPEVENT		R	0x0
1	GPMC_AD1_DUPLICATEWAKEUPEVENT		R	0x0
0	GPMC_AD0_DUPLICATEWAKEUPEVENT		R	0x0

Table 18-829. CTRL_CORE_PADCONF_WAKEUPEVENT_1

Address Offset	0x0000 186C		
Physical Address	0x4A00 386C	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

VIN1A_D2_DUPLICATEWAKEUPEVENT	VIN1A_D1_DUPLICATEWAKEUPEVENT	VIN1A_D0_DUPLICATEWAKEUPEVENT	VIN1A_VSYNC0_DUPLICATEWAKEUPEVENT	VIN1A_HSYNC0_DUPLICATEWAKEUPEVENT	VIN1A_FLD0_DUPLICATEWAKEUPEVENT	VIN1A_DE0_DUPLICATEWAKEUPEVENT	VIN1B_CLK1_DUPLICATEWAKEUPEVENT	VIN1A_CLK0_DUPLICATEWAKEUPEVENT	GPMC_WAIT0_DUPLICATEWAKEUPEVENT	GPMC_BEN1_DUPLICATEWAKEUPEVENT	GPMC_BEN0_DUPLICATEWAKEUPEVENT	GPMC_WEN_DUPLICATEWAKEUPEVENT	GPMC_OEN_REN_DUPLICATEWAKEUPEVENT	GPMC_ADVN_ALE_DUPLICATEWAKEUPEVENT	GPMC_CLK_DUPLICATEWAKEUPEVENT	GPMC_CS3_DUPLICATEWAKEUPEVENT	GPMC_CS2_DUPLICATEWAKEUPEVENT	GPMC_CS0_DUPLICATEWAKEUPEVENT	GPMC_CS1_DUPLICATEWAKEUPEVENT	GPMC_A27_DUPLICATEWAKEUPEVENT	GPMC_A26_DUPLICATEWAKEUPEVENT	GPMC_A25_DUPLICATEWAKEUPEVENT	GPMC_A24_DUPLICATEWAKEUPEVENT	GPMC_A23_DUPLICATEWAKEUPEVENT	GPMC_A22_DUPLICATEWAKEUPEVENT	GPMC_A21_DUPLICATEWAKEUPEVENT	GPMC_A20_DUPLICATEWAKEUPEVENT	GPMC_A19_DUPLICATEWAKEUPEVENT	GPMC_A18_DUPLICATEWAKEUPEVENT	GPMC_A17_DUPLICATEWAKEUPEVENT	GPMC_A16_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	VIN1A_D2_DUPLICATEWAKEUPEVENT		R	0x0
30	VIN1A_D1_DUPLICATEWAKEUPEVENT		R	0x0
29	VIN1A_D0_DUPLICATEWAKEUPEVENT		R	0x0
28	VIN1A_VSYNC0_DUPLICATEWAKEUPEVENT		R	0x0
27	VIN1A_HSYNC0_DUPLICATEWAKEUPEVENT		R	0x0
26	VIN1A_FLD0_DUPLICATEWAKEUPEVENT		R	0x0
25	VIN1A_DE0_DUPLICATEWAKEUPEVENT		R	0x0
24	VIN1B_CLK1_DUPLICATEWAKEUPEVENT		R	0x0
23	VIN1A_CLK0_DUPLICATEWAKEUPEVENT		R	0x0
22	GPMC_WAIT0_DUPLICATEWAKEUPEVENT		R	0x0
21	GPMC_BEN1_DUPLICATEWAKEUPEVENT		R	0x0
20	GPMC_BEN0_DUPLICATEWAKEUPEVENT		R	0x0
19	GPMC_WEN_DUPLICATEWAKEUPEVENT		R	0x0
18	GPMC_OEN_REN_DUPLICATEWAKEUPEVENT		R	0x0
17	GPMC_ADVN_ALE_DUPLICATEWAKEUPEVENT		R	0x0
16	GPMC_CLK_DUPLICATEWAKEUPEVENT		R	0x0
15	GPMC_CS3_DUPLICATEWAKEUPEVENT		R	0x0
14	GPMC_CS2_DUPLICATEWAKEUPEVENT		R	0x0
13	GPMC_CS0_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
12	GPMC_CS1_DUPLICATE WAKEUPEVENT		R	0x0
11	GPMC_A27_DUPLICATE WAKEUPEVENT		R	0x0
10	GPMC_A26_DUPLICATE WAKEUPEVENT		R	0x0
9	GPMC_A25_DUPLICATE WAKEUPEVENT		R	0x0
8	GPMC_A24_DUPLICATE WAKEUPEVENT		R	0x0
7	GPMC_A23_DUPLICATE WAKEUPEVENT		R	0x0
6	GPMC_A22_DUPLICATE WAKEUPEVENT		R	0x0
5	GPMC_A21_DUPLICATE WAKEUPEVENT		R	0x0
4	GPMC_A20_DUPLICATE WAKEUPEVENT		R	0x0
3	GPMC_A19_DUPLICATE WAKEUPEVENT		R	0x0
2	GPMC_A18_DUPLICATE WAKEUPEVENT		R	0x0
1	GPMC_A17_DUPLICATE WAKEUPEVENT		R	0x0
0	GPMC_A16_DUPLICATE WAKEUPEVENT		R	0x0

Table 18-830. CTRL_CORE_PADCONF_WAKEUPEVENT_2

Address Offset	0x0000 1870		
Physical Address	0x4A00 3870	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24
VIN2A_D5_DUPLICATEWAKEUPEVENT	VIN2A_D4_DUPLICATEWAKEUPEVENT	VIN2A_D3_DUPLICATEWAKEUPEVENT	VIN2A_D2_DUPLICATEWAKEUPEVENT	VIN2A_D1_DUPLICATEWAKEUPEVENT	VIN2A_D0_DUPLICATEWAKEUPEVENT	VIN2A_VSYNC0_DUPLICATEWAKEUPEVENT	VIN2A_HSYNC0_DUPLICATEWAKEUPEVENT

23	22	21	20	19	18	17	16
VIN2A_FLD0_DUPLICATEWAKEUPEVENT	VIN2A_DE0_DUPLICATEWAKEUPEVENT	VIN2A_CLK0_DUPLICATEWAKEUPEVENT	VIN1A_D23_DUPLICATEWAKEUPEVENT	VIN1A_D22_DUPLICATEWAKEUPEVENT	VIN1A_D21_DUPLICATEWAKEUPEVENT	VIN1A_D20_DUPLICATEWAKEUPEVENT	VIN1A_D19_DUPLICATEWAKEUPEVENT

15	14	13	12	11	10	9	8
VIN1A_D18_DUPLICATEWAKEUPEVENT	VIN1A_D17_DUPLICATEWAKEUPEVENT	VIN1A_D16_DUPLICATEWAKEUPEVENT	VIN1A_D15_DUPLICATEWAKEUPEVENT	VIN1A_D14_DUPLICATEWAKEUPEVENT	VIN1A_D13_DUPLICATEWAKEUPEVENT	VIN1A_D12_DUPLICATEWAKEUPEVENT	VIN1A_D11_DUPLICATEWAKEUPEVENT

7	6	5	4	3	2	1	0
VIN1A_D10_DUPLICATEWAKEUPEVENT	VIN1A_D9_DUPLICATEWAKEUPEVENT	VIN1A_D8_DUPLICATEWAKEUPEVENT	VIN1A_D7_DUPLICATEWAKEUPEVENT	VIN1A_D6_DUPLICATEWAKEUPEVENT	VIN1A_D5_DUPLICATEWAKEUPEVENT	VIN1A_D4_DUPLICATEWAKEUPEVENT	VIN1A_D3_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	VIN2A_D5_DUPLICATEW AKEUPEVENT		R	0x0
30	VIN2A_D4_DUPLICATEW AKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
29	VIN2A_D3_DUPLICATEWAKEUPEVENT		R	0x0
28	VIN2A_D2_DUPLICATEWAKEUPEVENT		R	0x0
27	VIN2A_D1_DUPLICATEWAKEUPEVENT		R	0x0
26	VIN2A_D0_DUPLICATEWAKEUPEVENT		R	0x0
25	VIN2A_VSYNC0_DUPLICATEWAKEUPEVENT		R	0x0
24	VIN2A_HSYNC0_DUPLICATEWAKEUPEVENT		R	0x0
23	VIN2A_FLD0_DUPLICATEWAKEUPEVENT		R	0x0
22	VIN2A_DE0_DUPLICATEWAKEUPEVENT		R	0x0
21	VIN2A_CLK0_DUPLICATEWAKEUPEVENT		R	0x0
20	VIN1A_D23_DUPLICATEWAKEUPEVENT		R	0x0
19	VIN1A_D22_DUPLICATEWAKEUPEVENT		R	0x0
18	VIN1A_D21_DUPLICATEWAKEUPEVENT		R	0x0
17	VIN1A_D20_DUPLICATEWAKEUPEVENT		R	0x0
16	VIN1A_D19_DUPLICATEWAKEUPEVENT		R	0x0
15	VIN1A_D18_DUPLICATEWAKEUPEVENT		R	0x0
14	VIN1A_D17_DUPLICATEWAKEUPEVENT		R	0x0
13	VIN1A_D16_DUPLICATEWAKEUPEVENT		R	0x0
12	VIN1A_D15_DUPLICATEWAKEUPEVENT		R	0x0
11	VIN1A_D14_DUPLICATEWAKEUPEVENT		R	0x0
10	VIN1A_D13_DUPLICATEWAKEUPEVENT		R	0x0
9	VIN1A_D12_DUPLICATEWAKEUPEVENT		R	0x0
8	VIN1A_D11_DUPLICATEWAKEUPEVENT		R	0x0
7	VIN1A_D10_DUPLICATEWAKEUPEVENT		R	0x0
6	VIN1A_D9_DUPLICATEWAKEUPEVENT		R	0x0
5	VIN1A_D8_DUPLICATEWAKEUPEVENT		R	0x0
4	VIN1A_D7_DUPLICATEWAKEUPEVENT		R	0x0
3	VIN1A_D6_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
2	VIN1A_D5_DUPLICATEWAKEUPEVENT		R	0x0
1	VIN1A_D4_DUPLICATEWAKEUPEVENT		R	0x0
0	VIN1A_D3_DUPLICATEWAKEUPEVENT		R	0x0

Table 18-831. CTRL_CORE_PADCONF_WAKEUPEVENT_3

Address Offset	0x0000 1874		
Physical Address	0x4A00 3874	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VOUT1_D8_DUPLICATEWAKEUPEVENT	VOUT1_D7_DUPLICATEWAKEUPEVENT	VOUT1_D6_DUPLICATEWAKEUPEVENT	VOUT1_D5_DUPLICATEWAKEUPEVENT	VOUT1_D4_DUPLICATEWAKEUPEVENT	VOUT1_D3_DUPLICATEWAKEUPEVENT	VOUT1_D2_DUPLICATEWAKEUPEVENT	VOUT1_D1_DUPLICATEWAKEUPEVENT	VOUT1_D0_DUPLICATEWAKEUPEVENT	VOUT1_VSYNC_DUPLICATEWAKEUPEVENT	VOUT1_HSYNC_DUPLICATEWAKEUPEVENT	VOUT1_FLD_DUPLICATEWAKEUPEVENT	VOUT1_DE_DUPLICATEWAKEUPEVENT	VOUT1_CLK_DUPLICATEWAKEUPEVENT	VIN2A_D23_DUPLICATEWAKEUPEVENT	VIN2A_D22_DUPLICATEWAKEUPEVENT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIN2A_D21_DUPLICATEWAKEUPEVENT	VIN2A_D20_DUPLICATEWAKEUPEVENT	VIN2A_D19_DUPLICATEWAKEUPEVENT	VIN2A_D18_DUPLICATEWAKEUPEVENT	VIN2A_D17_DUPLICATEWAKEUPEVENT	VIN2A_D16_DUPLICATEWAKEUPEVENT	VIN2A_D15_DUPLICATEWAKEUPEVENT	VIN2A_D14_DUPLICATEWAKEUPEVENT	VIN2A_D13_DUPLICATEWAKEUPEVENT	VIN2A_D12_DUPLICATEWAKEUPEVENT	VIN2A_D11_DUPLICATEWAKEUPEVENT	VIN2A_D10_DUPLICATEWAKEUPEVENT	VIN2A_D9_DUPLICATEWAKEUPEVENT	VIN2A_D8_DUPLICATEWAKEUPEVENT	VIN2A_D7_DUPLICATEWAKEUPEVENT	VIN2A_D6_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	VOUT1_D8_DUPLICATEWAKEUPEVENT		R	0x0
30	VOUT1_D7_DUPLICATEWAKEUPEVENT		R	0x0
29	VOUT1_D6_DUPLICATEWAKEUPEVENT		R	0x0
28	VOUT1_D5_DUPLICATEWAKEUPEVENT		R	0x0
27	VOUT1_D4_DUPLICATEWAKEUPEVENT		R	0x0
26	VOUT1_D3_DUPLICATEWAKEUPEVENT		R	0x0
25	VOUT1_D2_DUPLICATEWAKEUPEVENT		R	0x0
24	VOUT1_D1_DUPLICATEWAKEUPEVENT		R	0x0
23	VOUT1_D0_DUPLICATEWAKEUPEVENT		R	0x0
22	VOUT1_VSYNC_DUPLICATEWAKEUPEVENT		R	0x0
21	VOUT1_HSYNC_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
20	VOUT1_FLD_DUPLICATE WAKEUPEVENT		R	0x0
19	VOUT1_DE_DUPLICATE WAKEUPEVENT		R	0x0
18	VOUT1_CLK_DUPLICATE WAKEUPEVENT		R	0x0
17	VIN2A_D23_DUPLICATE WAKEUPEVENT		R	0x0
16	VIN2A_D22_DUPLICATE WAKEUPEVENT		R	0x0
15	VIN2A_D21_DUPLICATE WAKEUPEVENT		R	0x0
14	VIN2A_D20_DUPLICATE WAKEUPEVENT		R	0x0
13	VIN2A_D19_DUPLICATE WAKEUPEVENT		R	0x0
12	VIN2A_D18_DUPLICATE WAKEUPEVENT		R	0x0
11	VIN2A_D17_DUPLICATE WAKEUPEVENT		R	0x0
10	VIN2A_D16_DUPLICATE WAKEUPEVENT		R	0x0
9	VIN2A_D15_DUPLICATE WAKEUPEVENT		R	0x0
8	VIN2A_D14_DUPLICATE WAKEUPEVENT		R	0x0
7	VIN2A_D13_DUPLICATE WAKEUPEVENT		R	0x0
6	VIN2A_D12_DUPLICATE WAKEUPEVENT		R	0x0
5	VIN2A_D11_DUPLICATE WAKEUPEVENT		R	0x0
4	VIN2A_D10_DUPLICATE WAKEUPEVENT		R	0x0
3	VIN2A_D9_DUPLICATEW AKEUPEVENT		R	0x0
2	VIN2A_D8_DUPLICATEW AKEUPEVENT		R	0x0
1	VIN2A_D7_DUPLICATEW AKEUPEVENT		R	0x0
0	VIN2A_D6_DUPLICATEW AKEUPEVENT		R	0x0

Table 18-832. CTRL_CORE_PADCONF_WAKEUPEVENT_4

Address Offset	0x0000 1878		
Physical Address	0x4A00 3878	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RGMII0_RXD0_DUPLICATEWAKEUPEVENT	RGMII0_RXD1_DUPLICATEWAKEUPEVENT	RGMII0_RXD2_DUPLICATEWAKEUPEVENT	RGMII0_RXD3_DUPLICATEWAKEUPEVENT	RGMII0_RXCTL_DUPLICATEWAKEUPEVENT	RGMII0_RXC_DUPLICATEWAKEUPEVENT	RGMII0_TXD0_DUPLICATEWAKEUPEVENT	RGMII0_TXD1_DUPLICATEWAKEUPEVENT	RGMII0_TXD2_DUPLICATEWAKEUPEVENT	RGMII0_TXD3_DUPLICATEWAKEUPEVENT	RGMII0_TXCTL_DUPLICATEWAKEUPEVENT	RGMII0_TXC_DUPLICATEWAKEUPEVENT	UART3_TXD_DUPLICATEWAKEUPEVENT	UART3_RXD_DUPLICATEWAKEUPEVENT	RMII_MHZ_50_CLK_DUPLICATEWAKEUPEVENT	MDIO_D_DUPLICATEWAKEUPEVENT	MDIO_MCLK_DUPLICATEWAKEUPEVENT	VOUT1_D23_DUPLICATEWAKEUPEVENT	VOUT1_D22_DUPLICATEWAKEUPEVENT	VOUT1_D21_DUPLICATEWAKEUPEVENT	VOUT1_D20_DUPLICATEWAKEUPEVENT	VOUT1_D19_DUPLICATEWAKEUPEVENT	VOUT1_D18_DUPLICATEWAKEUPEVENT	VOUT1_D17_DUPLICATEWAKEUPEVENT	VOUT1_D16_DUPLICATEWAKEUPEVENT	VOUT1_D15_DUPLICATEWAKEUPEVENT	VOUT1_D14_DUPLICATEWAKEUPEVENT	VOUT1_D13_DUPLICATEWAKEUPEVENT	VOUT1_D12_DUPLICATEWAKEUPEVENT	VOUT1_D11_DUPLICATEWAKEUPEVENT	VOUT1_D10_DUPLICATEWAKEUPEVENT	VOUT1_D9_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	RGMII0_RXD0_DUPLICATEWAKEUPEVENT		R	0x0
30	RGMII0_RXD1_DUPLICATEWAKEUPEVENT		R	0x0
29	RGMII0_RXD2_DUPLICATEWAKEUPEVENT		R	0x0
28	RGMII0_RXD3_DUPLICATEWAKEUPEVENT		R	0x0
27	RGMII0_RXCTL_DUPLICATEWAKEUPEVENT		R	0x0
26	RGMII0_RXC_DUPLICATEWAKEUPEVENT		R	0x0
25	RGMII0_TXD0_DUPLICATEWAKEUPEVENT		R	0x0
24	RGMII0_TXD1_DUPLICATEWAKEUPEVENT		R	0x0
23	RGMII0_TXD2_DUPLICATEWAKEUPEVENT		R	0x0
22	RGMII0_TXD3_DUPLICATEWAKEUPEVENT		R	0x0
21	RGMII0_TXCTL_DUPLICATEWAKEUPEVENT		R	0x0
20	RGMII0_TXC_DUPLICATEWAKEUPEVENT		R	0x0
19	UART3_TXD_DUPLICATEWAKEUPEVENT		R	0x0
18	UART3_RXD_DUPLICATEWAKEUPEVENT		R	0x0
17	RMII_MHZ_50_CLK_DUPLICATEWAKEUPEVENT		R	0x0
16	MDIO_D_DUPLICATEWAKEUPEVENT		R	0x0
15	MDIO_MCLK_DUPLICATEWAKEUPEVENT		R	0x0
14	VOUT1_D23_DUPLICATEWAKEUPEVENT		R	0x0
13	VOUT1_D22_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
12	VOUT1_D21_DUPLICATE WAKEUPEVENT		R	0x0
11	VOUT1_D20_DUPLICATE WAKEUPEVENT		R	0x0
10	VOUT1_D19_DUPLICATE WAKEUPEVENT		R	0x0
9	VOUT1_D18_DUPLICATE WAKEUPEVENT		R	0x0
8	VOUT1_D17_DUPLICATE WAKEUPEVENT		R	0x0
7	VOUT1_D16_DUPLICATE WAKEUPEVENT		R	0x0
6	VOUT1_D15_DUPLICATE WAKEUPEVENT		R	0x0
5	VOUT1_D14_DUPLICATE WAKEUPEVENT		R	0x0
4	VOUT1_D13_DUPLICATE WAKEUPEVENT		R	0x0
3	VOUT1_D12_DUPLICATE WAKEUPEVENT		R	0x0
2	VOUT1_D11_DUPLICATE WAKEUPEVENT		R	0x0
1	VOUT1_D10_DUPLICATE WAKEUPEVENT		R	0x0
0	VOUT1_D9_DUPLICATE WAKEUPEVENT		R	0x0

Table 18-833. CTRL_CORE_PADCONF_WAKEUPEVENT_5

Address Offset	0x0000 187C		
Physical Address	0x4A00 387C	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24
MCASP2_ACLKR_DUPLICATEWAKEUPEVENT	MCASP2_FSX_DUPLICATEWAKEUPEVENT	MCASP2_ACLKX_DUPLICATEWAKEUPEVENT	MCASP1_AXR15_DUPLICATEWAKEUPEVENT	MCASP1_AXR14_DUPLICATEWAKEUPEVENT	MCASP1_AXR13_DUPLICATEWAKEUPEVENT	MCASP1_AXR12_DUPLICATEWAKEUPEVENT	MCASP1_AXR11_DUPLICATEWAKEUPEVENT

23	22	21	20	19	18	17	16
MCASP1_AXR10_DUPLICATEWAKEUPEVENT	MCASP1_AXR9_DUPLICATEWAKEUPEVENT	MCASP1_AXR8_DUPLICATEWAKEUPEVENT	MCASP1_AXR7_DUPLICATEWAKEUPEVENT	MCASP1_AXR6_DUPLICATEWAKEUPEVENT	MCASP1_AXR5_DUPLICATEWAKEUPEVENT	MCASP1_AXR4_DUPLICATEWAKEUPEVENT	MCASP1_AXR3_DUPLICATEWAKEUPEVENT

15	14	13	12	11	10	9	8
MCASP1_AXR2_DUPLICATEWAKEUPEVENT	MCASP1_AXR1_DUPLICATEWAKEUPEVENT	MCASP1_AXR0_DUPLICATEWAKEUPEVENT	MCASP1_FSR_DUPLICATEWAKEUPEVENT	MCASP1_ACLKR_DUPLICATEWAKEUPEVENT	MCASP1_FSX_DUPLICATEWAKEUPEVENT	MCASP1_ACLKX_DUPLICATEWAKEUPEVENT	XREF_CLK3_DUPLICATEWAKEUPEVENT

7	6	5	4	3	2	1	0
XREF_CLK2_DUPLICATEWAKEUPEVENT	XREF_CLK1_DUPLICATEWAKEUPEVENT	XREF_CLK0_DUPLICATEWAKEUPEVENT	GPIO6_16_DUPLICATEWAKEUPEVENT	GPIO6_15_DUPLICATEWAKEUPEVENT	GPIO6_14_DUPLICATEWAKEUPEVENT	USB2_DRVVBUS_DUPLICATEWAKEUPEVENT	USB1_DRVVBUS_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	MCASP2_ACLKR_DUPLI CATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
30	MCASP2_FSX_DUPLICATEWAKEUPEVENT		R	0x0
29	MCASP2_ACLKX_DUPLICATEWAKEUPEVENT		R	0x0
28	MCASP1_AXR15_DUPLICATEWAKEUPEVENT		R	0x0
27	MCASP1_AXR14_DUPLICATEWAKEUPEVENT		R	0x0
26	MCASP1_AXR13_DUPLICATEWAKEUPEVENT		R	0x0
25	MCASP1_AXR12_DUPLICATEWAKEUPEVENT		R	0x0
24	MCASP1_AXR11_DUPLICATEWAKEUPEVENT		R	0x0
23	MCASP1_AXR10_DUPLICATEWAKEUPEVENT		R	0x0
22	MCASP1_AXR9_DUPLICATEWAKEUPEVENT		R	0x0
21	MCASP1_AXR8_DUPLICATEWAKEUPEVENT		R	0x0
20	MCASP1_AXR7_DUPLICATEWAKEUPEVENT		R	0x0
19	MCASP1_AXR6_DUPLICATEWAKEUPEVENT		R	0x0
18	MCASP1_AXR5_DUPLICATEWAKEUPEVENT		R	0x0
17	MCASP1_AXR4_DUPLICATEWAKEUPEVENT		R	0x0
16	MCASP1_AXR3_DUPLICATEWAKEUPEVENT		R	0x0
15	MCASP1_AXR2_DUPLICATEWAKEUPEVENT		R	0x0
14	MCASP1_AXR1_DUPLICATEWAKEUPEVENT		R	0x0
13	MCASP1_AXR0_DUPLICATEWAKEUPEVENT		R	0x0
12	MCASP1_FSR_DUPLICATEWAKEUPEVENT		R	0x0
11	MCASP1_ACLKR_DUPLICATEWAKEUPEVENT		R	0x0
10	MCASP1_FSX_DUPLICATEWAKEUPEVENT		R	0x0
9	MCASP1_ACLKX_DUPLICATEWAKEUPEVENT		R	0x0
8	XREF_CLK3_DUPLICATEWAKEUPEVENT		R	0x0
7	XREF_CLK2_DUPLICATEWAKEUPEVENT		R	0x0
6	XREF_CLK1_DUPLICATEWAKEUPEVENT		R	0x0
5	XREF_CLK0_DUPLICATEWAKEUPEVENT		R	0x0
4	GPIO6_16_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
3	GPIO6_15_DUPLICATEWAKEUPEVENT		R	0x0
2	GPIO6_14_DUPLICATEWAKEUPEVENT		R	0x0
1	USB2_DRVVBUS_DUPLICATEWAKEUPEVENT		R	0x0
0	USB1_DRVVBUS_DUPLICATEWAKEUPEVENT		R	0x0

Table 18-834. CTRL_CORE_PADCONF_WAKEUPEVENT_6

Address Offset	0x0000 1880		
Physical Address	0x4A00 3880	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMC3_CLK_DUPLICATEWAKEUPEVENT	GPIO6_11_DUPLICATEWAKEUPEVENT	GPIO6_10_DUPLICATEWAKEUPEVENT	MMC1_SDWP_DUPLICATEWAKEUPEVENT	MMC1_SDCD_DUPLICATEWAKEUPEVENT	MMC1_DAT3_DUPLICATEWAKEUPEVENT	MMC1_DAT2_DUPLICATEWAKEUPEVENT	MMC1_DAT1_DUPLICATEWAKEUPEVENT	MMC1_DAT0_DUPLICATEWAKEUPEVENT	MMC1_CMD_DUPLICATEWAKEUPEVENT	MMC1_CLK_DUPLICATEWAKEUPEVENT	MCASP5_AXR1_DUPLICATEWAKEUPEVENT	MCASP5_AXR0_DUPLICATEWAKEUPEVENT	MCASP5_FSX_DUPLICATEWAKEUPEVENT	MCASP5_ACLKX_DUPLICATEWAKEUPEVENT	MCASP4_AXR1_DUPLICATEWAKEUPEVENT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCASP4_AXR0_DUPLICATEWAKEUPEVENT	MCASP4_FSX_DUPLICATEWAKEUPEVENT	MCASP4_ACLKX_DUPLICATEWAKEUPEVENT	MCASP3_AXR1_DUPLICATEWAKEUPEVENT	MCASP3_AXR0_DUPLICATEWAKEUPEVENT	MCASP3_FSX_DUPLICATEWAKEUPEVENT	MCASP3_ACLKX_DUPLICATEWAKEUPEVENT	MCASP2_AXR7_DUPLICATEWAKEUPEVENT	MCASP2_AXR6_DUPLICATEWAKEUPEVENT	MCASP2_AXR5_DUPLICATEWAKEUPEVENT	MCASP2_AXR4_DUPLICATEWAKEUPEVENT	MCASP2_AXR3_DUPLICATEWAKEUPEVENT	MCASP2_AXR2_DUPLICATEWAKEUPEVENT	MCASP2_AXR1_DUPLICATEWAKEUPEVENT	MCASP2_AXR0_DUPLICATEWAKEUPEVENT	MCASP2_FSR_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	MMC3_CLK_DUPLICATEWAKEUPEVENT		R	0x0
30	GPIO6_11_DUPLICATEWAKEUPEVENT		R	0x0
29	GPIO6_10_DUPLICATEWAKEUPEVENT		R	0x0
28	MMC1_SDWP_DUPLICATEWAKEUPEVENT		R	0x0
27	MMC1_SDCD_DUPLICATEWAKEUPEVENT		R	0x0
26	MMC1_DAT3_DUPLICATEWAKEUPEVENT		R	0x0
25	MMC1_DAT2_DUPLICATEWAKEUPEVENT		R	0x0
24	MMC1_DAT1_DUPLICATEWAKEUPEVENT		R	0x0
23	MMC1_DAT0_DUPLICATEWAKEUPEVENT		R	0x0
22	MMC1_CMD_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
21	MMC1_CLK_DUPLICATE WAKEUPEVENT		R	0x0
20	MCASP5_AXR1_DUPLIC ATEWAKEUPEVENT		R	0x0
19	MCASP5_AXR0_DUPLIC ATEWAKEUPEVENT		R	0x0
18	MCASP5_FSX_DUPLICA TEWAKEUPEVENT		R	0x0
17	MCASP5_ACLKX_DUPLI CATEWAKEUPEVENT		R	0x0
16	MCASP4_AXR1_DUPLIC ATEWAKEUPEVENT		R	0x0
15	MCASP4_AXR0_DUPLIC ATEWAKEUPEVENT		R	0x0
14	MCASP4_FSX_DUPLICA TEWAKEUPEVENT		R	0x0
13	MCASP4_ACLKX_DUPLI CATEWAKEUPEVENT		R	0x0
12	MCASP3_AXR1_DUPLIC ATEWAKEUPEVENT		R	0x0
11	MCASP3_AXR0_DUPLIC ATEWAKEUPEVENT		R	0x0
10	MCASP3_FSX_DUPLICA TEWAKEUPEVENT		R	0x0
9	MCASP3_ACLKX_DUPLI CATEWAKEUPEVENT		R	0x0
8	MCASP2_AXR7_DUPLIC ATEWAKEUPEVENT		R	0x0
7	MCASP2_AXR6_DUPLIC ATEWAKEUPEVENT		R	0x0
6	MCASP2_AXR5_DUPLIC ATEWAKEUPEVENT		R	0x0
5	MCASP2_AXR4_DUPLIC ATEWAKEUPEVENT		R	0x0
4	MCASP2_AXR3_DUPLIC ATEWAKEUPEVENT		R	0x0
3	MCASP2_AXR2_DUPLIC ATEWAKEUPEVENT		R	0x0
2	MCASP2_AXR1_DUPLIC ATEWAKEUPEVENT		R	0x0
1	MCASP2_AXR0_DUPLIC ATEWAKEUPEVENT		R	0x0
0	MCASP2_FSR_DUPLICA TEWAKEUPEVENT		R	0x0

Table 18-835. CTRL_CORE_PADCONF_WAKEUPEVENT_7

Address Offset	0x0000 1884		
Physical Address	0x4A00 3884	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

UART2_RTSN_DUPLICATEWAKEUPEVENT	UART2_CTSN_DUPLICATEWAKEUPEVENT	UART2_TXD_DUPLICATEWAKEUPEVENT	UART2_RXD_DUPLICATEWAKEUPEVENT	UART1_RTSN_DUPLICATEWAKEUPEVENT	UART1_CTSN_DUPLICATEWAKEUPEVENT	UART1_TXD_DUPLICATEWAKEUPEVENT	UART1_RXD_DUPLICATEWAKEUPEVENT	DCAN2_RX_DUPLICATEWAKEUPEVENT	DCAN2_TX_DUPLICATEWAKEUPEVENT	DCAN1_RX_DUPLICATEWAKEUPEVENT	DCAN1_TX_DUPLICATEWAKEUPEVENT	SPI2_CS0_DUPLICATEWAKEUPEVENT	SPI2_D0_DUPLICATEWAKEUPEVENT	SPI2_D1_DUPLICATEWAKEUPEVENT	SPI2_SCLK_DUPLICATEWAKEUPEVENT	SPI1_CS3_DUPLICATEWAKEUPEVENT	SPI1_CS2_DUPLICATEWAKEUPEVENT	SPI1_CS1_DUPLICATEWAKEUPEVENT	SPI1_CS0_DUPLICATEWAKEUPEVENT	SPI1_D0_DUPLICATEWAKEUPEVENT	SPI1_D1_DUPLICATEWAKEUPEVENT	SPI1_SCLK_DUPLICATEWAKEUPEVENT	MMC3_DAT7_DUPLICATEWAKEUPEVENT	MMC3_DAT6_DUPLICATEWAKEUPEVENT	MMC3_DAT5_DUPLICATEWAKEUPEVENT	MMC3_DAT4_DUPLICATEWAKEUPEVENT	MMC3_DAT3_DUPLICATEWAKEUPEVENT	MMC3_DAT2_DUPLICATEWAKEUPEVENT	MMC3_DAT1_DUPLICATEWAKEUPEVENT	MMC3_DAT0_DUPLICATEWAKEUPEVENT	MMC3_CMD_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	UART2_RTSN_DUPLICATEWAKEUPEVENT		R	0x0
30	UART2_CTSN_DUPLICATEWAKEUPEVENT		R	0x0
29	UART2_TXD_DUPLICATEWAKEUPEVENT		R	0x0
28	UART2_RXD_DUPLICATEWAKEUPEVENT		R	0x0
27	UART1_RTSN_DUPLICATEWAKEUPEVENT		R	0x0
26	UART1_CTSN_DUPLICATEWAKEUPEVENT		R	0x0
25	UART1_TXD_DUPLICATEWAKEUPEVENT		R	0x0
24	UART1_RXD_DUPLICATEWAKEUPEVENT		R	0x0
23	DCAN2_RX_DUPLICATEWAKEUPEVENT		R	0x0
22	DCAN2_TX_DUPLICATEWAKEUPEVENT		R	0x0
21	DCAN1_RX_DUPLICATEWAKEUPEVENT		R	0x0
20	DCAN1_TX_DUPLICATEWAKEUPEVENT		R	0x0
19	SPI2_CS0_DUPLICATEWAKEUPEVENT		R	0x0
18	SPI2_D0_DUPLICATEWAKEUPEVENT		R	0x0
17	SPI2_D1_DUPLICATEWAKEUPEVENT		R	0x0
16	SPI2_SCLK_DUPLICATEWAKEUPEVENT		R	0x0
15	SPI1_CS3_DUPLICATEWAKEUPEVENT		R	0x0
14	SPI1_CS2_DUPLICATEWAKEUPEVENT		R	0x0
13	SPI1_CS1_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
12	SPI1_CS0_DUPLICATEWAKEUPEVENT		R	0x0
11	SPI1_D0_DUPLICATEWAKEUPEVENT		R	0x0
10	SPI1_D1_DUPLICATEWAKEUPEVENT		R	0x0
9	SPI1_SCLK_DUPLICATEWAKEUPEVENT		R	0x0
8	MMC3_DAT7_DUPLICATEWAKEUPEVENT		R	0x0
7	MMC3_DAT6_DUPLICATEWAKEUPEVENT		R	0x0
6	MMC3_DAT5_DUPLICATEWAKEUPEVENT		R	0x0
5	MMC3_DAT4_DUPLICATEWAKEUPEVENT		R	0x0
4	MMC3_DAT3_DUPLICATEWAKEUPEVENT		R	0x0
3	MMC3_DAT2_DUPLICATEWAKEUPEVENT		R	0x0
2	MMC3_DAT1_DUPLICATEWAKEUPEVENT		R	0x0
1	MMC3_DAT0_DUPLICATEWAKEUPEVENT		R	0x0
0	MMC3_CMD_DUPLICATEWAKEUPEVENT		R	0x0

Table 18-836. CTRL_CORE_PADCONF_WAKEUPEVENT_8

Address Offset	0x0000 1888		
Physical Address	0x4A00 3888	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31:19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	NMIN_DUPLICATEWAKEUPEVENT	EMU4_DUPLICATEWAKEUPEVENT	EMU3_DUPLICATEWAKEUPEVENT	EMU2_DUPLICATEWAKEUPEVENT	EMU1_DUPLICATEWAKEUPEVENT	EMU0_DUPLICATEWAKEUPEVENT	RTCK_DUPLICATEWAKEUPEVENT	TDO_DUPLICATEWAKEUPEVENT	TDI_DUPLICATEWAKEUPEVENT	WAKEUP3_DUPLICATEWAKEUPEVENT	WAKEUP2_DUPLICATEWAKEUPEVENT	WAKEUP1_DUPLICATEWAKEUPEVENT	WAKEUP0_DUPLICATEWAKEUPEVENT	I2C3_SCL_DUPLICATEWAKEUPEVENT	I2C3_SDA_DUPLICATEWAKEUPEVENT	I2C2_SCL_DUPLICATEWAKEUPEVENT	I2C2_SDA_DUPLICATEWAKEUPEVENT	I2C1_SCL_DUPLICATEWAKEUPEVENT	I2C1_SDA_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	NMIN_DUPLICATEWAKEUPEVENT		R	0x0
17	EMU4_DUPLICATEWAKEUPEVENT		R	0x0

Bits	Field Name	Description	Type	Reset
16	EMU3_DUPLICATEWAKEUPEVENT		R	0x0
15	EMU2_DUPLICATEWAKEUPEVENT		R	0x0
14	EMU1_DUPLICATEWAKEUPEVENT		R	0x0
13	EMU0_DUPLICATEWAKEUPEVENT		R	0x0
12	RTCK_DUPLICATEWAKEUPEVENT		R	0x0
11	TDO_DUPLICATEWAKEUPEVENT		R	0x0
10	TDI_DUPLICATEWAKEUPEVENT		R	0x0
9	WAKEUP3_DUPLICATEWAKEUPEVENT		R	0x0
8	WAKEUP2_DUPLICATEWAKEUPEVENT		R	0x0
7	WAKEUP1_DUPLICATEWAKEUPEVENT		R	0x0
6	WAKEUP0_DUPLICATEWAKEUPEVENT		R	0x0
5	I2C3_SCL_DUPLICATEWAKEUPEVENT		R	0x0
4	I2C3_SDA_DUPLICATEWAKEUPEVENT		R	0x0
3	I2C2_SCL_DUPLICATEWAKEUPEVENT		R	0x0
2	I2C2_SDA_DUPLICATEWAKEUPEVENT		R	0x0
1	I2C1_SCL_DUPLICATEWAKEUPEVENT		R	0x0
0	I2C1_SDA_DUPLICATEWAKEUPEVENT		R	0x0

Table 18-837. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_2

Address offset	0x0000 1B08		
Physical Address	0x4A00 3B08	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_NOM. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_GPU_2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-

Bits	Field Name	Description	Type	Reset
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_NOM which has to be written to the CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL [4:0] LDOVBBGPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_GPU_2	AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_NOM. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-838. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_3

Address offset 0x0000 1B0C

Physical Address 0x4A00 3B0C **Instance** CTRL_MODULE_CORE

Description This register contains the AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_OD. This register also stores information about ABB configuration for that OPP.

Type R

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_GPU_3

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_OD which has to be written to the CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL [4:0] LDOVBBGPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_GPU_3	AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_OD. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-839. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_4

Address Offset 0x0000 1B10

Physical Address 0x4A00 3B10 **Instance** CTRL_MODULE_CORE

Description This register contains the AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_HIGH. This register also stores information about ABB configuration for that OPP.

Type R

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_GPU_4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-

Bits	Field Name	Description	Type	Reset
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_HIGH which has to be written to the CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL [4:0] LDOVBBGPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_GPU_4	AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_HIGH. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-840. CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_5

Address Offset	0x0000 1B14		
Physical Address	0x4A00 3B14	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_PLUS. This register also stores information about ABB configuration for that OPP.		
Type	R		

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_PLUS which has to be written to the CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL[4:0] LDOVBBGPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_GPU_5	AVS Class 0 voltage value for the vdd_gpu voltage rail when running at OPP_PLUS. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-841. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_1

Address offset	0x0000 1B1C		
Physical Address	0x4A00 3B1C	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_LOW. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_MPU_1

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_LOW which has to be written to the CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL[4:0] LDOVBBMPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-

Bits	Field Name	Description	Type	Reset
11:0	STD_FUSE_OPP_VMIN_MPU_1	AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_LOW. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-842. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_2

Address offset	0x0000 1B20		
Physical Address	0x4A00 3B20	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_NOM. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_MPU_2

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_NOM which has to be written to the CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL[4:0] LDOVBBMPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_MPU_2	AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_NOM. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-843. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_3

Address offset	0x0000 1B24		
Physical Address	0x4A00 3B24	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_OD. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_MPU_3

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_OD which has to be written to the CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL[4:0] LDOVBBMPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-

Bits	Field Name	Description	Type	Reset
11:0	STD_FUSE_OPP_VMIN_MPU_3	AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_OD. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-844. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_4

Address Offset	0x0000 1B28		
Physical Address	0x4A00 3B28	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_HIGH. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_MPU_4

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_HIGH which has to be written to the CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL[4:0] LDOVBBMPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_MPU_4	AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_HIGH. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-845. CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_5

Address Offset	0x0000 1B2C		
Physical Address	0x4A00 3B2C	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_PLUS. This register also stores information about ABB configuration for that OPP.		
Type	R		

31:26	25	24:20	19:12	11:0
RESERVED	ABBEN	VSETABB	RESERVED	STD_FUSE_OPP_VMIN_MPU_5

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25	ABBEN	0x0: ABB is disabled 0x1: ABB is enabled	R	0x-
24:20	VSETABB	This bit field shows the ABB LDO target value for OPP_PLUS which has to be written to the CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL[4:0] LDOVBBMPU_FBB_VSET_OUT bit field, if ABB is enabled.	R	0x-
19:12	RESERVED	Reserved	R	0x-

Bits	Field Name	Description	Type	Reset
11:0	STD_FUSE_OPP_VMIN_MPU_5	AVS Class 0 voltage value for the vdd_mpu voltage rail when running at OPP_PLUS. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 18-846. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_0

Address Offset	0x0000 1B38		
Physical Address	0x4A00 3B38	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_0		R	0x0

Table 18-847. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_1

Address Offset	0x0000 1B3C		
Physical Address	0x4A00 3B3C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_1		R	0x0

Table 18-848. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_2

Address Offset	0x0000 1B40		
Physical Address	0x4A00 3B40	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_2		R	0x0

Table 18-849. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_3

Address Offset	0x0000 1B44		
Physical Address	0x4A00 3B44	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_3		R	0x0

Table 18-850. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_4

Address Offset 0x0000 1B48

Physical Address 0x4A00 3B48 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_GPU [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_4		R	0x0

Table 18-851. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_0

Address Offset 0x0000 1B4C

Physical Address 0x4A00 3B4C **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_IVA [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_LVT_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_LVT_0		R	0x0

Table 18-852. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_1

Address Offset 0x0000 1B50

Physical Address 0x4A00 3B50 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_IVA [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_LVT_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_LVT_1		R	0x0

Table 18-853. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_2

Address Offset 0x0000 1B54

Physical Address 0x4A00 3B54 **Instance** CTRL_MODULE_CORE

Description Standard Fuse OPP VDD_IVA [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.

Table 18-853. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_2 (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_LVT_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_LVT_2		R	0x0

Table 18-854. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_3

Address Offset	0x0000 1B58		
Physical Address	0x4A00 3B58	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_IVA [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_LVT_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_LVT_3		R	0x0

Table 18-855. CTRL_CORE_STD_FUSE_OPP_VDD_IVA_LVT_4

Address Offset	0x0000 1B5C		
Physical Address	0x4A00 3B5C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_IVA [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_IVA_LVT_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_IVA_LVT_4		R	0x0

Table 18-856. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_0

Address Offset	0x0000 1B60		
Physical Address	0x4A00 3B60	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_0		R	0x0

Table 18-857. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_1

Address Offset	0x0000 1B64

Table 18-857. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_1 (continued)

Physical Address	0x4A00 3B64	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_1		R	0x0

Table 18-858. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_2

Address Offset	0x0000 1B68		
Physical Address	0x4A00 3B68	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_2		R	0x0

Table 18-859. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_3

Address Offset	0x0000 1B6C		
Physical Address	0x4A00 3B6C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_3		R	0x0

Table 18-860. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_4

Address Offset	0x0000 1B70		
Physical Address	0x4A00 3B70	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_4		R	0x0

Table 18-861. CTRL_CORE_LDOSRAM_CORE_4_VOLTAGE_CTRL

Address Offset	0x0000 1B74		
Physical Address	0x4A00 3B74	**Instance**	CTRL_MODULE_CORE
Description	CORE 4th SRAM LDO Control register		
Type	RW		

31:27	26	25:21	20:16	15:11	10	9:5	4:0
RESERVED	LDOSRAMCORE_4_RETMODE_MUX_CTRL	LDOSRAMCORE_4_RETMODE_VSET_IN	LDOSRAMCORE_4_RETMODE_VSET_OUT	RESERVED	LDOSRAMCORE_4_ACTMODE_MUX_CTRL	LDOSRAMCORE_4_ACTMODE_VSET_IN	LDOSRAMCORE_4_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_4_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_4_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_4_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_4_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_4_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_4_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-862. CTRL_CORE_LDOSRAM_CORE_5_VOLTAGE_CTRL

Address Offset	0x0000 1B78		
Physical Address	0x4A00 3B78	**Instance**	CTRL_MODULE_CORE
Description	CORE 5th SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LDOSRAMCORE_5_RETMODE_MUX_CTRL	LDOSRAMCORE_5_RETMODE_VSET_IN	LDOSRAMCORE_5_RETMODE_VSET_OUT	RESERVED	LDOSRAMCORE_5_ACTMODE_MUX_CTRL	LDOSRAMCORE_5_ACTMODE_VSET_IN	LDOSRAMCORE_5_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_5_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_5_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_5_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_5_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_5_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_5_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-863. CTRL_CORE_LDOSRAM_DSPEVE_2_VOLTAGE_CTRL

Address Offset	0x0000 1B7C		
Physical Address	0x4A00 3B7C	**Instance**	CTRL_MODULE_CORE
Description	DSPEVE 2nd SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LDOSRAMDSPEVE_2_RETMODE_MUX_CTRL	LDOSRAMDSPEVE_2_RETMODE_VSET_IN	LDOSRAMDSPEVE_2_RETMODE_VSET_OUT	RESERVED	LDOSRAMDSPEVE_2_ACTMODE_MUX_CTRL	LDOSRAMDSPEVE_2_ACTMODE_VSET_IN	LDOSRAMDSPEVE_2_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMDSPEVE_2_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMDSPEVE_2_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMDSPEVE_2_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMDSPEVE_2_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMDSPEVE_2_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMDSPEVE_2_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-864. CTRL_CORE_SMA_SW_2

Address Offset	0x0000 1C04		
Physical Address	0x4A00 3C04	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_2																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_2	OCP spare register	RW	0x0

Table 18-865. CTRL_CORE_SMA_SW_3

Address Offset	0x0000 1C08		
Physical Address	0x4A00 3C08	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		

Table 18-865. CTRL_CORE_SMA_SW_3 (continued)

Type	RW

31–0
SMA_SW_3

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_3	OCP spare register	RW	0x0

Table 18-866. CTRL_CORE_SMA_SW_6

Address Offset	0x0000 1C14		
Physical Address	0x4A00 3C14	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31:29	28:27	26:18	17:16	15:9	8	7:1	0
RESERVED	PLLEN_CONTROL	RESERVED	PCIE_TX_RX_CONTROL	RESERVED	RMII_CLK_SETTING	RESERVED	MUXSEL_32K_CLKIN

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:27	PLLEN_CONTROL	PLLEN control setting. Bit [28] – Controls the CLKOUT of DPLL_USB_OTG 0x0: CLKOUT is disabled. 0x1: CLKOUT is enabled. Bit [27] – Controls the CLKOUT of DPLL_SATA 0x0: CLKOUT is disabled. 0x1: CLKOUT is enabled.	RW	0x0
26:18	RESERVED		R	0x0
17:16	PCIE_TX_RX_CONTROL	PCIe RX and TX control of ACSPCIe. 0x0: ACSPCIe Power Down Mode 0x1: ACSPCIe TX Mode 0x2: ACSPCIe RX Mode 0x3: Reserved	RW	0x0
15:9	RESERVED		R	0x0
8	RMII_CLK_SETTING	RMII CLK setting 0x0: Internal clock from DPLL_GMAC 0x1: External clock from RMII_MHZ_50_CLK pin	RW	0x0
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	MUXSEL_32K_CLKIN	Setting for mux to select 32KHz clock input to PRCM. This bit must NOT be modified by software. The 32kHz clock selection is done through the device sysboot[9:8] signals.	RW	0x0

Table 18-867. CTRL_CORE_SMA_SW_7

Address Offset	0x0000 1C18		
Physical Address	0x4A00 3C18	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														MMU1_ABORT_ENABLE	MMU2_ABORT_ENABLE	RESERVED		EDMA_TC1_WR_MMU_ROUTE_ENABLE	EDMA_TC1_RD_MMU_ROUTE_ENABLE	EDMA_TC0_WR_MMU_ROUTE_ENABLE	EDMA_TC0_RD_MMU_ROUTE_ENABLE	PCIE_SS2_MMU_ROUTE_ENABLE	PCIE_SS1_MMU_ROUTE_ENABLE	RESERVED						PCIE_SS2_AXI2OCP_LEGACY_MODE_ENABLE	PCIE_SS1_AXI2OCP_LEGACY_MODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0
17	MMU1_ABORT_ENABLE	MMU1 abort enable	RW	0x0
16	MMU2_ABORT_ENABLE	MMU2 abort enable	RW	0x0
15:14	RESERVED	Reserved	R	0x0
13	EDMA_TC1_WR_MMU_ROUTE_ENABLE	EDMA TC1 WR traffic MMU route enable	RW	0x0
12	EDMA_TC1_RD_MMU_ROUTE_ENABLE	EDMA TC1 RD traffic MMU route enable	RW	0x0
11	EDMA_TC0_WR_MMU_ROUTE_ENABLE	EDMA TC0 WR traffic MMU route enable	RW	0x0
10	EDMA_TC0_RD_MMU_ROUTE_ENABLE	EDMA TC0 RD traffic MMU route enable	RW	0x0
9	PCIE_SS2_MMU_ROUTE_ENABLE	PCIe_SS2 MMU route enable	RW	0x0
8	PCIE_SS1_MMU_ROUTE_ENABLE	PCIe_SS1 MMU route enable	RW	0x0
7:2	RESERVED	Reserved	R	0x0
1	PCIE_SS2_AXI2OCP_LEGACY_MODE_ENABLE	PCIe_SS2 AXI2OCP legacy mode enable	RW	0x0
0	PCIE_SS1_AXI2OCP_LEGACY_MODE_ENABLE	PCIe_SS1 AXI2OCP legacy mode enable	RW	0x0

Table 18-868. CTRL_CORE_SMA_SW_8

Address Offset	0x0000 1C1C		
Physical Address	0x4A00 3C1C	**Instance**	CTRL_MODULE_CORE
Description	Test control inputs used by the module		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCIE_PLL_TEST_INPUT_1																															

Bits	Field Name	Description	Type	Reset
31:0	PCIE_PLL_TEST_INPUT_1	Test control inputs used by the module	RW	0x0

Table 18-869. CTRL_CORE_SMA_SW_9

Address Offset	0x0000 1C20		
Physical Address	0x4A00 3C20	**Instance**	CTRL_MODULE_CORE
Description	Test control inputs used by the module		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCIE_PLL_TEST_INPUT_2																															

Bits	Field Name	Description	Type	Reset
31:0	PCIE_PLL_TEST_INPUT_2	Test control inputs used by the module	RW	0x0

Table 18-870. CTRL_CORE_PCIESS1_PCS1

Address Offset	0x0000 1C24		
Physical Address	0x4A00 3C24	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCIESS1_PCS_TEST_TXDATA										PCIESS1_PCS_ERR_BIT_EN										PCIESS1_PCS_CFG_HOLDOFF								PCIESS1_PCS_DET_DELAY			

Bits	Field Name	Description	Type	Reset
31:22	PCIESS1_PCS_TEST_TXDATA		RW	0x0
21:12	PCIESS1_PCS_ERR_BIT_EN		RW	0x0
11:4	PCIESS1_PCS_CFG_HOLDOFF		RW	0x0
3:0	PCIESS1_PCS_DET_DELAY		RW	0x1

Table 18-871. CTRL_CORE_PCIESS1_PCS2

Address Offset	0x0000 1C28		
Physical Address	0x4A00 3C28	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PCIESS1_PCS_CFG_SYNC	PCIESS1_PCS_CFG_EQ_FUNC	PCIESS1_PCS_CFG_EQ_HOLD	PCIESS1_PCS_CFG_EQ_INIT	PCIESS1_PCS_TEST_OSEL	RESERVED	PCIESS1_PCS_TEST_LSEL	RESERVED	PCIESS1_PCS_ERR_MODE	PCIESS1_PCS_L1_SLEEP	PCIESS1_PCS_TEST_MODE	PCIESS1_PCS_ERR_LN_EN	RESERVED	PCIESS1_PCS_SHORT_TIMES

Bits	Field Name	Description	Type	Reset
31:27	PCIESS1_PCS_CFG_SYNC		RW	0x0
26:23	PCIESS1_PCS_CFG_EQ_FUNC		RW	0x0
22:19	PCIESS1_PCS_CFG_EQ_HOLD		RW	0x0
18:15	PCIESS1_PCS_CFG_EQ_INIT		RW	0x0
14:12	PCIESS1_PCS_TEST_OSEL		RW	0x0
11:10	RESERVED		R	0x0
9	PCIESS1_PCS_TEST_LSEL		RW	0x0
8	RESERVED		R	0x0
7:6	PCIESS1_PCS_ERR_MODE		RW	0x0
5	PCIESS1_PCS_L1_SLEEP		RW	0x0
4	PCIESS1_PCS_TEST_MODE		RW	0x0
3:2	PCIESS1_PCS_ERR_LN_EN		RW	0x0
1	RESERVED		R	0x0
0	PCIESS1_PCS_SHORT_TIMES		RW	0x0

Table 18-872. CTRL_CORE_PCIESS2_PCS1

Address Offset	0x0000 1C2C		
Physical Address	0x4A00 3C2C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:22	21:12	11:4	3:0
PCIESS2_PCS_TEST_TXDATA	PCIESS2_PCS_ERR_BIT_EN	PCIESS2_PCS_CFG_HOLDOFF	PCIESS2_PCS_DET_DELAY

Bits	Field Name	Description	Type	Reset
31:22	PCIESS2_PCS_TEST_TXDATA		RW	0x0
21:12	PCIESS2_PCS_ERR_BIT_EN		RW	0x0
11:4	PCIESS2_PCS_CFG_HOLDOFF		RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	PCIESS2_PCS_DET_DELAY		RW	0x1

Table 18-873. CTRL_CORE_PCIESS2_PCS2

Address Offset	0x0000 1C30		
Physical Address	0x4A00 3C30	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCIESS2_PCS_CFG_SYNC					PCIESS2_PCS_CFG_EQ_FUNC				PCIESS2_PCS_CFG_EQ_HOLD				PCIESS2_PCS_CFG_EQ_INIT				PCIESS2_PCS_TEST_OSEL			RESERVED		PCIESS2_PCS_TEST_LSEL	RESERVED	PCIESS2_PCS_ERR_MODE		PCIESS2_PCS_L1_SLEEP	PCIESS2_PCS_TEST_MODE	PCIESS2_PCS_ERR_LN_EN		RESERVED	PCIESS2_PCS_SHORT_TIMES

Bits	Field Name	Description	Type	Reset
31:27	PCIESS2_PCS_CFG_SYNC		RW	0x0
26:23	PCIESS2_PCS_CFG_EQ_FUNC		RW	0x0
22:19	PCIESS2_PCS_CFG_EQ_HOLD		RW	0x0
18:15	PCIESS2_PCS_CFG_EQ_INIT		RW	0x0
14:12	PCIESS2_PCS_TEST_OSEL		RW	0x0
11:10	RESERVED		R	0x0
9	PCIESS2_PCS_TEST_LSEL		RW	0x0
8	RESERVED		R	0x0
7:6	PCIESS2_PCS_ERR_MODE		RW	0x0
5	PCIESS2_PCS_L1_SLEEP		RW	0x0
4	PCIESS2_PCS_TEST_MODE		RW	0x0
3:2	PCIESS2_PCS_ERR_LN_EN		RW	0x0
1	RESERVED		R	0x0
0	PCIESS2_PCS_SHORT_TIMES		RW	0x0

Table 18-874. CTRL_CORE_PCIE_PCS

Address Offset	0x0000 1C34		
Physical Address	0x4A00 3C34	**Instance**	CTRL_MODULE_CORE
Description			

Table 18-874. CTRL_CORE_PCIE_PCS (continued)

Type	RW

31:24	23:16	15:0
RESERVED	PCIESS_PCS_RC_DELAY_COUNT	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:16	PCIESS_PCS_RC_DELAY_COUNT	Set to 0x96 for proper functional and compliance-mode behavior on both PCIESS1 and PCIESS2.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 18-875. CTRL_CORE_PCIE_PCS_REVISION

Address Offset	0x0000 1C38		
Physical Address	0x4A00 3C38	**Instance**	CTRL_MODULE_CORE
Description	pcs_revision		
Type	R		

31:26	25:23	22:20	19:0
RESERVED	PCIESS2_PCS_REVISION	PCIESS1_PCS_REVISION	RESERVED

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:23	PCIESS2_PCS_REVISION		R	0x0
22:20	PCIESS1_PCS_REVISION		R	0x0
19:0	RESERVED		R	0x0

Table 18-876. CTRL_CORE_PCIE_CONTROL

Address Offset	0x0000 1C3C		
Physical Address	0x4A00 3C3C	**Instance**	CTRL_MODULE_CORE
Description	serdes control selection PCIE C0 (0 default) vs PCIE B1 (1)		
Type	RW		

31:3	2	1	0
RESERVED	PCIE_B1C0_MODE_SEL	RESERVED	PCIE_B0_B1_TSYNCEN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	PCIE_B1C0_MODE_SEL	0x0: PCIESS1 x1 Mode and/or PCIESS2 x1 Mode 0x1: PCIESS1 x2 Mode, PCIESS2 Unused	RW	0x0
1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	PCIE_B0_B1_TSYNCEN	0x0: PCIESS1 x1 Mode and/or PCIESS2 x1 Mode 0x1: PCIESS1 x2 Mode, PCIESS2 Unused	RW	0x0

Table 18-877. CTRL_CORE_PHY_POWER_PCIESS1

Address Offset	0x0000 1C40		
Physical Address	0x4A00 3C40	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:22	21:14	13:0
PCIESS1_PWRCTL_CLKFREQ	PCIESS1_PWRCTL_CMD	RESERVED

Bits	Field Name	Description	Type	Reset
31:22	PCIESS1_PWRCTL_CLKFREQ	Frequency of SYSCLK1 in MHz (rounded). For example, for 20MHz, program 0x14.	RW	0x0
21:14	PCIESS1_PWRCTL_CMD	Powers up/down the PCIESS1_PHY_TX and PCIESS1_PHY_RX modules. 0x0: Powers down PCIESS1_PHY_TX and PCIESS1_PHY_RX 0x1: Powers up PCIESS1_PHY_RX 0x2: Powers up PCIESS1_PHY_TX 0x3: Powers up PCIESS1_PHY_TX and PCIESS1_PHY_RX 0x4-0xFF: Reserved	RW	0x0
13:0	RESERVED	Reserved	R	0x0

Table 18-878. CTRL_CORE_PHY_POWER_PCIESS2

Address Offset	0x0000 1C44		
Physical Address	0x4A00 3C44	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31:22	21:14	13:0
PCIESS2_PWRCTL_CLKFREQ	PCIESS2_PWRCTL_CMD	RESERVED

Bits	Field Name	Description	Type	Reset
31:22	PCIESS2_PWRCTL_CLKFREQ	Frequency of SYSCLK1 in MHz (rounded). For example, for 20MHz, program 0x14.	RW	0x0
21:14	PCIESS2_PWRCTL_CMD	Powers up/down the PCIESS2_PHY_TX and PCIESS2_PHY_RX modules. 0x0: Powers down PCIESS2_PHY_TX and PCIESS2_PHY_RX 0x1: Powers up PCIESS2_PHY_RX 0x2: Powers up PCIESS2_PHY_TX 0x3: Powers up PCIESS2_PHY_TX and PCIESS2_PHY_RX 0x4-0xFF: Reserved	RW	0x0
13:0	RESERVED	Reserved	R	0x0

18.5.5 CTRL_MODULE_WKUP Registers

18.5.6 CTRL_MODULE_WKUP Register Summary

Table 18-879. CTRL_MODULE_WKUP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE _WKUP Base Address
RESERVED_a (a = 0 to 63)	R	32	0x0000 0000 + (a*4)	0x4AE0 C000 + (a*4)
CTRL_WKUP_SEC_CTRL	RW	32	0x0000 0100	0x4AE0 C100
RESERVED	R	32	0x0000 0104	0x4AE0 C104
CTRL_WKUP_SEC_TAP	RW	32	0x0000 0108	0x4AE0 C108
CTRL_WKUP_OCPREG_SPARE	RW	32	0x0000 010C	0x4AE0 C10C
CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG	RW	32	0x0000 0110	0x4AE0 C110
RESERVED	R	32	0x0000 0114	0x4AE0 C114
CTRL_WKUP_SECURE_EMIF2_SDRAM_CONFIG	RW	32	0x0000 0118	0x4AE0 C118
RESERVED_i (i = 0 to 6)	R	32	0x0000 011C + (i*4)	0x4AE0 C11C + (i*4)
CTRL_WKUP_STD_FUSE_USB_CONF	R	32	0x0000 0138	0x4AE0 C138
CTRL_WKUP_STD_FUSE_CONF	R	32	0x0000 013C	0x4AE0 C13C
RESERVED	R	32	0x0000 0140	0x4AE0 C140
CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT	RW	32	0x0000 0144	0x4AE0 C144
CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT	RW	32	0x0000 0148	0x4AE0 C148
CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1	R	32	0x0000 014C	0x4AE0 C14C
CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT_2	R	32	0x0000 0150	0x4AE0 C150
CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL	RW	32	0x0000 0154	0x4AE0 C154
CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL	RW	32	0x0000 0158	0x4AE0 C158
CTRL_WKUP_LDOSRAM_GPU_VOLTAGE_CTRL	RW	32	0x0000 015C	0x4AE0 C15C
CTRL_WKUP_LDOSRAM_MPU_VOLTAGE_CTRL	RW	32	0x0000 0160	0x4AE0 C160
CTRL_WKUP_LDOSRAM_CORE_VOLTAGE_CTRL	RW	32	0x0000 0164	0x4AE0 C164
CTRL_WKUP_LDOSRAM_MPU_2_VOLTAGE_CTRL	RW	32	0x0000 0168	0x4AE0 C168
RESERVED_j (j = 0 to 36)	R	32	0x0000 016C + (j*4)	0x4AE0 C16C + (j*4)
CTRL_WKUP_STD_FUSE_DIE_ID_0	R	32	0x0000 0200	0x4AE0 C200
CTRL_WKUP_ID_CODE	R	32	0x0000 0204	0x4AE0 C204
CTRL_WKUP_STD_FUSE_DIE_ID_1	R	32	0x0000 0208	0x4AE0 C208
CTRL_WKUP_STD_FUSE_DIE_ID_2	R	32	0x0000 020C	0x4AE0 C20C
CTRL_WKUP_STD_FUSE_DIE_ID_3	R	32	0x0000 0210	0x4AE0 C210
CTRL_WKUP_STD_FUSE_PROD_ID_0	R	32	0x0000 0214	0x4AE0 C214
RESERVED_k (k = 0 to 292)	R	32	0x0000 0218 + (k*4)	0x4AE0 C218 + (k*4)
CTRL_WKUP_CONTROL_XTAL_OSCILLATOR	RW	32	0x0000 05AC	0x4AE0 C5AC
RESERVED	R	32	0x0000 05B0	0x4AE0 C5B0
RESERVED	R	32	0x0000 05B4	0x4AE0 C5B4
RESERVED	R	32	0x0000 05B8	0x4AE0 C5B8
RESERVED	R	32	0x0000 05BC	0x4AE0 C5BC
RESERVED	R	32	0x0000 05C0	0x4AE0 C5C0
RESERVED	R	32	0x0000 05C4	0x4AE0 C5C4
CTRL_WKUP_EFUSE_1	RW	32	0x0000 05C8	0x4AE0 C5C8
CTRL_WKUP_EFUSE_2	RW	32	0x0000 05CC	0x4AE0 C5CC
CTRL_WKUP_EFUSE_3	RW	32	0x0000 05D0	0x4AE0 C5D0
CTRL_WKUP_EFUSE_4	RW	32	0x0000 05D4	0x4AE0 C5D4

Table 18-879. CTRL_MODULE_WKUP Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_WKUP Base Address
RESERVED_m (m = 0 to 7)	R	32	0x0000 05D8 + (m*4)	0x4AE0 C5D8 + (m*4)
CTRL_WKUP_EFUSE_13	RW	32	0x0000 05F8	0x4AE0 C5F8

18.5.7 CTRL_MODULE_WKUP Register Description

Table 18-880. CTRL_WKUP_SEC_CTRL

Address Offset	0x0000 0100		
Physical Address	0x4AE0 C100	**Instance**	CTRL_MODULE_WKUP
Description	Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECCTRLWRDISABLE	RESERVED																										SECURE_EMIF_CONFIG_RO_EN	RESERVED			

Bits	Field Name	Description	Type	Reset
31	SECCTRLWRDISABLE	Control Register write disable control. 0x0 = Write in this register is allowed 0x1 = Write in this register is forbidden	RW	0x0
30:5	RESERVED		R	0x0
4	SECURE_EMIF_CONFIG_RO_EN	Access mode for registers: CTRL_WKUP_EMIF1_SDRAM_CONFIG CTRL_WKUP_EMIF2_SDRAM_CONFIG 0x0 = These registers are RW 0x1 = These registers are RO	RW	0x0
3:0	RESERVED		R	0x0

Table 18-881. CTRL_WKUP_SEC_TAP

Address Offset	0x0000 0108		
Physical Address	0x4AE0 C108	**Instance**	CTRL_MODULE_WKUP
Description	TAP controllers register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECTAPWR_DISABLE	RESERVED				RESERVED	RESERVED												IPU2_TAPENABLE	DSP2_TAPENABLE	JTAG_EXT_TAPENABLE	IVA_TAPENABLE	MPUGLOBALDEBUG_ENABLE	RESERVED				IEEE1500_ENABLE	P1500_ENABLE	IPU1_TAPENABLE	DSP1_TAPENABLE	DAP_TAPENABLE

Bits	Field Name	Description	Type	Reset
31	SECTAPWR_DISABLE	TAP controllers register write disable control 0x0: Write in this register is allowed 0x1: Write in this register is forbidden	RW Woco	0x0
30:27	RESERVED		R	0x0
26	RESERVED	Reserved. This bit must not be modified.	RW	0x1
25:14	RESERVED		R	0x0
13	IPU2_TAPENABLE	IPU2 TAP control 0x0: IPU2 TAP controller is disabled 0x1: IPU2 TAP controller is enabled	RW	0x1
12	DSP2_TAPENABLE	DSP2 TAP control 0x0: DSP2 TAP controller is disabled 0x1: DSP2 TAP controller is enabled	RW	0x1
11	JTAGEXT_TAPENABLE	External JTAG expansion TAP control. 0x0: external JTAG TAP controller is disabled 0x1: external JTAG TAP controller is enabled	RW	0x1
10	IVA_TAPENABLE	IVA TAP control 0x0: IVA TAP controller is disabled 0x1: IVA TAP controller is enabled	RW	0x1
9	MPUGLOBALDEBUG_ENABLE	MPU TAP control 0x0: MPU TAP controller is disabled 0x1: MPU TAP controller is enabled	RW	0x1
8:5	RESERVED		R	0x0
4	IEEE1500_ENABLE	IEEE1500 and P1500 access enable 0x0: P1500 controller is disabled 0x1: P1500 controller is enabled	RW W1toClr	0x1
3	P1500_ENABLE	P1500 access enable 0x0: P1500 controller is disabled 0x1: P1500 controller is enabled	RW	0x1
2	IPU1_TAPENABLE	IPU1 TAP control 0x0: IPU1 TAP controller is disabled 0x1: IPU1 TAP controller is enabled	RW	0x1
1	DSP1_TAPENABLE	DSP1 TAP control 0x0: DSP1 TAP controller is disabled 0x1: DSP1 TAP controller is enabled	RW	0x1
0	DAP_TAPENABLE	DAP TAP control 0x0: DAP TAP controller is disabled 0x1: DAP TAP controller is enabled	RW	0x1

Table 18-882. CTRL_WKUP_OCPREG_SPARE

Address Offset	0x0000 010C		
Physical Address	0x4AE0 C10C	**Instance**	CTRL_MODULE_WKUP
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OCPREG_SPARE31	OCPREG_SPARE30	OCPREG_SPARE29	OCPREG_SPARE28	OCPREG_SPARE27	OCPREG_SPARE26	OCPREG_SPARE25	OCPREG_SPARE24	OCPREG_SPARE23	OCPREG_SPARE22	OCPREG_SPARE21	OCPREG_SPARE20	OCPREG_SPARE19	OCPREG_SPARE18	OCPREG_SPARE17	OCPREG_SPARE16	OCPREG_SPARE15	OCPREG_SPARE14	OCPREG_SPARE13	OCPREG_SPARE12	OCPREG_SPARE11	OCPREG_SPARE10	OCPREG_SPARE9	OCPREG_SPARE8	OCPREG_SPARE7	OCPREG_SPARE6	OCPREG_SPARE5	OCPREG_SPARE4	OCPREG_SPARE3	OCPREG_SPARE2	OCPREG_SPARE1	RESERVED

Bits	Field Name	Description	Type	Reset
31	OCPREG_SPARE31	OCP spare register 31	RW	0x0
30	OCPREG_SPARE30	OCP spare register 30	RW	0x0
29	OCPREG_SPARE29	OCP spare register 29	RW	0x0
28	OCPREG_SPARE28	OCP spare register 28	RW	0x0
27	OCPREG_SPARE27	OCP spare register 27	RW	0x0
26	OCPREG_SPARE26	OCP spare register 26	RW	0x0
25	OCPREG_SPARE25	OCP spare register 25	RW	0x0
24	OCPREG_SPARE24	OCP spare register 24	RW	0x0
23	OCPREG_SPARE23	OCP spare register 23	RW	0x0
22	OCPREG_SPARE22	OCP spare register 22	RW	0x0
21	OCPREG_SPARE21	OCP spare register 21	RW	0x0
20	OCPREG_SPARE20	OCP spare register 20	RW	0x0
19	OCPREG_SPARE19	OCP spare register 19	RW	0x0
18	OCPREG_SPARE18	OCP spare register 18	RW	0x0
17	OCPREG_SPARE17	OCP spare register 17	RW	0x0
16	OCPREG_SPARE16	OCP spare register 16	RW	0x0
15	OCPREG_SPARE15	OCP spare register 15	RW	0x0
14	OCPREG_SPARE14	OCP spare register 14	RW	0x0
13	OCPREG_SPARE13	OCP spare register 13	RW	0x0
12	OCPREG_SPARE12	OCP spare register 12	RW	0x0
11	OCPREG_SPARE11	OCP spare register 11	RW	0x0
10	OCPREG_SPARE10	OCP spare register 10	RW	0x0
9	OCPREG_SPARE9	OCP spare register 9	RW	0x0
8	OCPREG_SPARE8	OCP spare register 8	RW	0x0
7	OCPREG_SPARE7	OCP spare register 7	RW	0x0
6	OCPREG_SPARE6	OCP spare register 6	RW	0x0
5	OCPREG_SPARE5	OCP spare register 5	RW	0x0
4	OCPREG_SPARE4	OCP spare register 4	RW	0x0
3	OCPREG_SPARE3	OCP spare register 3	RW	0x0
2	OCPREG_SPARE2	OCP spare register 2	RW	0x0
1	OCPREG_SPARE1	OCP spare register 1	RW	0x0
0	RESERVED		R	0x0

Table 18-883. CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG

Address Offset	0x0000 0110

Table 18-883. CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG (continued)

Physical Address	0x4AE0 C110	**Instance**	CTRL_MODULE_WKUP
Description	EMIF1 SDRAM configuration register. Its values are exported to EMIF_SDRAM_CONFIG register at POR. For bit field descriptions see EMIF_SDRAM_CONFIG register in *EMIF Controller*, in *Memory Subsystem*. Write to this register is allowed if the CTRL_WKUP_SEC_CTRL[4] SECURE_EMIF_CONFIG_RO_EN bit is set to 0x0 (default).		
Type	RW		

31 30 29	28 27	26 25 24	23	22 21	20	19 18	17 16	15 14	13 12 11 10	9 8 7	6 5 4	3	2 1 0
RESERVED	EMIF1_SDRAM_IBANK_POS	EMIF1_SDRAM_DDR_TERM	EMIF1_SDRAM_DDR2_DDQS	EMIF1_SDRAM_DYN_ODT	EMIF1_SDRAM_DDR_DISABLE_DLL	EMIF1_SDRAM_DRIVE	EMIF1_SDRAM_CWL	RESERVED	EMIF1_SDRAM_CL	EMIF1_SDRAM_ROWSIZE	EMIF1_SDRAM_IBANK	RESERVED	EMIF1_SDRAM_PAGESIZE

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:27	EMIF1_SDRAM_IBANK_POS	Internal bank position.	RW	0x0
26:24	EMIF1_SDRAM_DDR_TERM	DDR2 and DDR3 termination resistor value.	RW	0x0
23	EMIF1_SDRAM_DDR2_DDQS	DDR2 differential DQS enable.	RW	0x1
22:21	EMIF1_SDRAM_DYN_ODT	DDR3 Dynamic ODT.	RW	0x0
20	EMIF1_SDRAM_DDR_DISABLE_DLL	Disable DLL select.	RW	0x0
19:18	EMIF1_SDRAM_DRIVE	SDRAM drive strength.	RW	0x0
17:16	EMIF1_SDRAM_CWL	DDR3 CAS Write latency.	RW	0x0
15:14	RESERVED		R	0x0
13:10	EMIF1_SDRAM_CL	CAS Latency.	RW	0x0
9:7	EMIF1_SDRAM_ROWSIZE	Row Size.	RW	0x0
6:4	EMIF1_SDRAM_IBANK	Internal Bank setup.	RW	0x0
3	RESERVED		R	0x0
2:0	EMIF1_SDRAM_PAGESIZE	Page Size.	RW	0x0

Table 18-884. CTRL_WKUP_SECURE_EMIF2_SDRAM_CONFIG

Address Offset	0x0000 0118		
Physical Address	0x4AE0 C118	**Instance**	CTRL_MODULE_WKUP
Description	EMIF2 SDRAM register. Its values are exported to EMIF_SDRAM_CONFIG register at POR. For bit field descriptions see EMIF_SDRAM_CONFIG register in *EMIF Controller*, in *Memory Subsystem*. Write to this register is allowed if the CTRL_WKUP_SEC_CTRL[4] SECURE_EMIF_CONFIG_RO_EN bit is set to 0x0 (default).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EMIF2_SDRAM_IBANK_POS		EMIF2_SDRAM_DDR_TERM			EMIF2_SDRAM_DDR2_DDQS	EMIF2_SDRAM_DYN_ODT		EMIF2_SDRAM_DDR_DISABLE_DLL	EMIF2_SDRAM_DRIVE		EMIF2_SDRAM_CWL		RESERVED		EMIF2_SDRAM_CL				EMIF2_SDRAM_ROWSIZE			EMIF2_SDRAM_IBANK			RESERVED	EMIF2_SDRAM_PAGESIZE		

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:27	EMIF2_SDRAM_IBANK_POS	Internal bank position.	RW	0x0
26:24	EMIF2_SDRAM_DDR_TERM	DDR2 and DDR3 termination resistor value.	RW	0x0
23	EMIF2_SDRAM_DDR2_DDQS	DDR2 differential DQS enable.	RW	0x1
22:21	EMIF2_SDRAM_DYN_ODT	DDR3 Dynamic ODT.	RW	0x0
20	EMIF2_SDRAM_DDR_DISABLE_DLL	Disable DLL select.	RW	0x0
19:18	EMIF2_SDRAM_DRIVE	SDRAM drive strength.	RW	0x0
17:16	EMIF2_SDRAM_CWL	DDR3 CAS Write latency.	RW	0x0
15:14	RESERVED		R	0x0
13:10	EMIF2_SDRAM_CL	CAS Latency.	RW	0x0
9:7	EMIF2_SDRAM_ROWSIZE	Row Size.	RW	0x0
6:4	EMIF2_SDRAM_IBANK	Internal Bank setup.	RW	0x0
3	RESERVED		R	0x0
2:0	EMIF2_SDRAM_PAGESIZE	Page Size.	RW	0x0

Table 18-885. CTRL_WKUP_STD_FUSE_USB_CONF

Address Offset	0x0000 0138		
Physical Address	0x4AE0 C138	**Instance**	CTRL_MODULE_WKUP
Description	Standard Fuse conf [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USB_PROD_ID																USB_VENDOR_ID															

Bits	Field Name	Description	Type	Reset
31:16	USB_PROD_ID	USB Product Identification	R	0x0
15:0	USB_VENDOR_ID	USB Vendor Identification	R	0x0

Table 18-886. CTRL_WKUP_STD_FUSE_CONF

Address Offset	0x0000 013C		
Physical Address	0x4AE0 C13C	**Instance**	CTRL_MODULE_WKUP
Description	Standard Fuse conf [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										STD_FUSE_EMIF2_INITREF_DEF_DIS	STD_FUSE_EMIF2_DDR3_LPDDR2N	STD_FUSE_EMIF1_INITREF_DEF_DIS	STD_FUSE_EMIF1_DDR3_LPDDR2N	RESERVED	STD_FUSE_HDCP_ENABLE	RESERVED			STD_FUSE_CH_SPEEDUP_DISABLE	RESERVED							STD_FUSE_SGX540_3D_CLOCK_SOURCE	STD_FUSE_SGX540_3D_DISABLE	RESERVED		

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x-
21	STD_FUSE_EMIF2_INITREF_DEF_DIS	Disable EMIF2 DDR refresh and initialization sequence 0x1 = refresh and initialization sequence are disabled 0x0 = refresh and initialization sequence are enabled	R	0x-
20	STD_FUSE_EMIF2_DDR3_LPDDR2N	EMIF2 DDR3 0x1= DDR3 configured 0x0 = reserved	R	0x-
19	STD_FUSE_EMIF1_INITREF_DEF_DIS	Disable EMIF1 DDR refresh and initialization sequence 0x1 = refresh and initialization sequence are disabled 0x0 = refresh and initialization sequence are enabled	R	0x-
18	STD_FUSE_EMIF1_DDR3_LPDDR2N	EMIF1 DDR3 0x1 = DDR3 configured 0x0 = reserved	R	0x-
17	RESERVED		R	0x-
16	STD_FUSE_HDCP_ENABLE	Enable hdcp 0x0 = enables hdcp 0x1 = disables hdcp	R	0x-
15:13	RESERVED		R	0x-
12	STD_FUSE_CH_SPEEDUP_DISABLE	ROM code settings for configuration header block and speedup block. Only SW access (no hardware access). 0x0 = enables CH and speedup 0x1 = disables CH and speedup	R	0x-

Bits	Field Name	Description	Type	Reset
11:5	RESERVED		R	0x-
4	STD_FUSE_SGX540_3D_CLOCK_SOURCE	Functional clock selection for the 3D accelerator engine 0x0 = GPU is fully enabled (DPLL_CORE/PER) 0x1 = GPU is partially enabled (DPLL_PER/8 max)	R	0x-
3	STD_FUSE_SGX540_3D_DISABLE	Disable the 3D accelerator engine 0x1 = SGX is disabled 0x0 = SGX is enabled	R	0x-
2:0	RESERVED		R	0x-

Table 18-887. CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT

Address Offset	0x0000 0144		
Physical Address	0x4AE0 C144	**Instance**	CTRL_MODULE_WKUP
Description	SLICE register for emif1 and emif2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EMIF1_NARROW_ONLY	EMIF1_EN_ECC	EMIF1_REG_PHY_NUM_OF_SAMPLES		EMIF1_REG_PHY_SEL_LOGIC	EMIF1_REG_PHY_ALL_DQ_MPR_RD_RESP	EMIF1_REG_PHY_OUTPUT_STATUS_SELECT			RESERVED	EMIF1_SDRAM_DISABLE_RESET	EMIF1_PHY_RD_LOCAL_ODT		RESERVED	EMIF1_DFI_CLOCK_PHASE_CTRL	EMIF1_EN_SLICE_2	EMIF1_EN_SLICE_1	EMIF1_EN_SLICE_0

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	EMIF1_NARROW_ONLY	EMIF1 operates in narrow mode, to allow for data macros to be powered down to save power 0x0 = narrow mode disabled 0x1 = narrow mode enabled	RW	0x0
16	EMIF1_EN_ECC	EMIF1 ECC enable 0x0 = ECC is disabled 0x1 = ECC is enabled	RW	0x0
15:14	EMIF1_REG_PHY_NUM_OF_SAMPLES	Controls the number of DQ samples required for read leveling. The recommended setting for full leveling is 0x3 (128 samples) . 0x0 = 4 samples 0x1 = 8 samples. 0x2 = 16 samples 0x3 = 128 samples	RW	0x0
13	EMIF1_REG_PHY_SEL_LOGIC	Selects an algorithm for read leveling. The use of algorithm 1 (set by default) is recommended. 0x0 = Algorithm 1 is used 0x1 = Algorithm 2 is used	RW	0x0

Bits	Field Name	Description	Type	Reset
12	EMIF1_REG_PHY_ALL_DQ_MPR_RD_RESP	Analysis method of DQ bits during read leveling. 0x0: if the DRAM provides a read response on only one DQ bit (this can be any bit, since in this mode all 8 DQ bits are OR-ed together). This is the default setting and works with all memory types (memories send responses on all DQ bits or on a single DQ bit). 0x1: if the DRAM provides a read response on all DQ bits.	RW	0x0
11:9	EMIF1_REG_PHY_OUTPUT_STATUS_SELECT	Selects the status to be observed on the outputs of the DDR PHYs through CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1 register. 0x0 = selects phy_reg_rdlvl_start_ratio[7:0] 0x1 = selects phy_reg_rdlvl_start_ratio[15:8] 0x2 = selects phy_reg_rdlvl_end_ratio[7:0] 0x3 = selects phy_reg_rdlvl_end_ratio[15:8]	RW	0x0
8	RESERVED		R	0x1
7	EMIF1_SDRAM_DISABLE_RESET	DDR3 SDRAM reset disable. 0x0 = DDR3 SDRAM reset signal is enabled. It can be asserted by EMIF 0x1 = DDR3 SDRAM reset signal is disabled. It is forbidden to EMIF to assert it.	RW	0x0
6:5	EMIF1_PHY_RD_LOCAL_ODT	Control of ODT (on – die termination) settings for the device DDR I/Os. ODT is enabled only during read operations when termination is required. 0x0 = ODT disabled 0x1= 60 Ohms 0x2 = 80 Ohms 0x3 =120 Ohms	RW	0x0
4	RESERVED		RW	0x0
3	EMIF1_DFI_CLOCK_PHASE_CTRL	EMIF_FICLK clock phase control (shifting by 180°). For normal operation this bit must always be set to 0x0 (disabled).	RW	0x0
2	EMIF1_EN_SLICE_2	Enable command PHY 2. When using DDR3 this bit can be set to 0x0 or 0x1. For lower power consumption 0x0 is used.	RW	0x1
1	EMIF1_EN_SLICE_1	Enable command PHY 1. 0x1 is the mandatory setting if DDR3 is used. EMIF1_EN_SLICE_0 and EMIF1_EN_SLICE_1 have to be programmed with the same value.	RW	0x1
0	EMIF1_EN_SLICE_0	Enable command PHY 0. 0x1 is the mandatory setting if DDR3 is used. EMIF1_EN_SLICE_0 and EMIF1_EN_SLICE_1 have to be programmed with the same value.	RW	0x1

Table 18-888. CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT

Address Offset	0x0000 0148		
Physical Address	0x4AE0 C148	**Instance**	CTRL_MODULE_WKUP
Description	SLICE register for emif1 and emif2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	EMIF2_NARROW_ONLY	RESERVED	EMIF2_REG_PHY_NUM_OF_SAMPLES	EMIF2_REG_PHY_SEL_LOGIC	EMIF2_REG_PHY_ALL_DQ_MPR_RD_RESP	EMIF2_REG_PHY_OUTPUT_STATUS_SELECT	RESERVED	EMIF2_SDRAM_DISABLE_RESET	EMIF2_PHY_RD_LOCAL_ODT	RESERVED	EMIF2_DFI_CLOCK_PHASE_CTRL	EMIF2_EN_SLICE_2	EMIF2_EN_SLICE_1	EMIF2_EN_SLICE_0

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	EMIF2_NARROW_ONLY	EMIF2 operates in narrow mode, to allow for data macros to be powered down to save power 0x0 = narrow mode disabled 0x1 = narrow mode enabled	RW	0x0
16	RESERVED		R	0x0
15:14	EMIF2_REG_PHY_NUM_OF_SAMPLES	Controls the number of DQ samples required for read leveling. The recommended setting for full leveling is 0x3 (128 samples) . 0x0 = 4 samples 0x1 = 8 samples. 0x2 = 16 samples 0x3 = 128 samples	RW	0x0
13	EMIF2_REG_PHY_SEL_LOGIC	Selects an algorithm for read leveling. The use of algorithm 1 (set by default) is recommended. 0x0 = Algorithm 1 is used 0x1 = Algorithm 2 is used	RW	0x0
12	EMIF2_REG_PHY_ALL_DQ_MPR_RD_RESP	Analysis method of DQ bits during read leveling. 0x0: if the DRAM provides a read response on only one DQ bit (this can be any bit, since in this mode all 8 DQ bits are OR-ed together). This is the default setting and works with all memory types (memories send responses on all DQ bits or on a single DQ bit). 0x1: if the DRAM provides a read response on all DQ bits.	RW	0x0
11:9	EMIF2_REG_PHY_OUTPUT_STATUS_SELECT	Selects the status to be observed on the outputs of the DDR PHYs through CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT_2 register. 0x0 = selects phy_reg_rdlvl_start_ratio[7:0] 0x1 = selects phy_reg_rdlvl_start_ratio[15:8] 0x2 = selects phy_reg_rdlvl_end_ratio[7:0] 0x3 = selects phy_reg_rdlvl_end_ratio[15:8]	RW	0x0
8	RESERVED		R	0x1
7	EMIF2_SDRAM_DISABLE_RESET	DDR3 SDRAM reset disable. 0x0 = DDR3 SDRAM reset signal is enabled. It can be asserted by EMIF 0x1 = DDR3 SDRAM reset signal is disabled. It is forbidden to EMIF to assert it.	RW	0x0

Bits	Field Name	Description	Type	Reset
6:5	EMIF2_PHY_RD_LOCAL_ODT	Control of ODT (on – die termination) settings for the device DDR I/Os. ODT is enabled only during read operations when termination is required. 0x0 = ODT disabled 0x1= 60 Ohms 0x2 = 80 Ohms 0x3 =120 Ohms	RW	0x0
4	RESERVED		R	0x0
3	EMIF2_DFI_CLOCK_PHASE_CTRL	EMIF_FICLK clock phase control (shifting by 180°). For normal operation this bit must always be set to 0x0 (disabled).	RW	0x0
2	EMIF2_EN_SLICE_2	Enable command PHY 2. When using DDR3 this bit can be set to 0x0 or 0x1. For lower power consumption 0x0 is used.	RW	0x1
1	EMIF2_EN_SLICE_1	Enable command PHY 1. 0x1 is the mandatory setting if DDR3 is used. EMIF2_EN_SLICE_0 and EMIF2_EN_SLICE_1 have to be programmed with the same value.	RW	0x1
0	EMIF2_EN_SLICE_0	Enable command PHY 0. 0x1 is the mandatory setting if DDR3 is used. EMIF2_EN_SLICE_0 and EMIF2_EN_SLICE_1 have to be programmed with the same value.	RW	0x1

Table 18-889. CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1

Address Offset	0x0000 014C		
Physical Address	0x4AE0 C14C	**Instance**	CTRL_MODULE_WKUP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMIF1_PHY_REG_READ_DATA_EYE_LVL																															

Bits	Field Name	Description	Type	Reset
31:0	EMIF1_PHY_REG_READ_DATA_EYE_LVL		R	0x0

Table 18-890. CTRL_WKUP_EMIF2_SDRAM_CONFIG_EXT_2

Address Offset	0x0000 0150		
Physical Address	0x4AE0 C150	**Instance**	CTRL_MODULE_WKUP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMIF2_PHY_REG_READ_DATA_EYE_LVL																															

Bits	Field Name	Description	Type	Reset
31:0	EMIF2_PHY_REG_READ_DATA_EYE_LVL		R	0x0

Table 18-891. CTRL_WKUP_LDOVBB_GPU_VOLTAGE_CTRL

Address Offset	0x0000 0154		
Physical Address	0x4AE0 C154	**Instance**	CTRL_MODULE_WKUP
Description	GPU Voltage Body Bias LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LDOVBBGPU_FBB_MUX_CTRL	LDOVBBGPU_FBB_VSET_IN					LDOVBBGPU_FBB_VSET_OUT				

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LDOVBBGPU_FBB_MUX_CTRL	Override control of EFUSE Forward Body Bias voltage value 0x0 = efuse value is used 0x1 = override value is used	RW	0x0
9:5	LDOVBBGPU_FBB_VSET_IN	EFUSE Forward Body Bias voltage value	R	0x0
4:0	LDOVBBGPU_FBB_VSET_OUT	Override value for Forward Body Bias voltage. If ABB is used, depending on the OPP this bit field should be loaded with a value read from one of the CTRL_CORE_STD_FUSE_OPP_VMIN_GPU_x[24:20] VSETABB bit fields. This value applies if LDOVBBGPU_FBB_MUX_CTRL is set to 0x1.	RW	0x0

Table 18-892. CTRL_WKUP_LDOVBB_MPU_VOLTAGE_CTRL

Address Offset	0x0000 0158		
Physical Address	0x4AE0 C158	**Instance**	CTRL_MODULE_WKUP
Description	MPU Voltage Body Bias LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LDOVBBMPU_FBB_MUX_CTRL	LDOVBBMPU_FBB_VSET_IN					LDOVBBMPU_FBB_VSET_OUT				

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LDOVBBMPU_FBB_MUX_CTRL	Override control of EFUSE Forward Body Bias voltage value 0x0 = efuse value is used 0x1 = override value is used	RW	0x0
9:5	LDOVBBMPU_FBB_VSET_IN	EFUSE Forward Body Bias voltage value	R	0x0

Bits	Field Name	Description	Type	Reset
4:0	LDOVBBMPU_FBB_VSET_OUT	Override value for Forward Body Bias voltage. If ABB is used, depending on the OPP this bit field should be loaded with a value read from one of the CTRL_CORE_STD_FUSE_OPP_VMIN_MPU_x[24:20] VSETABB bit fields. This value applies if LDOVBBMPU_FBB_MUX_CTRL is set to 0x1.	RW	0x0

Table 18-893. CTRL_WKUP_LDOSRAM_GPU_VOLTAGE_CTRL

Address Offset	0x0000 015C		
Physical Address	0x4AE0 C15C	**Instance**	CTRL_MODULE_WKUP
Description	GPU SRAM LDO Control register		
Type	RW		

31:27	26	25:21	20:16	15:11	10	9:5	4:0
RESERVED	LDOSRAMGPU_RETMODE_MUX_CTRL	LDOSRAMGPU_RETMODE_VSET_IN	LDOSRAMGPU_RETMODE_VSET_OUT	RESERVED	LDOSRAMGPU_ACTMODE_MUX_CTRL	LDOSRAMGPU_ACTMODE_VSET_IN	LDOSRAMGPU_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMGPU_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMGPU_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMGPU_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMGPU_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMGPU_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMGPU_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-894. CTRL_WKUP_LDOSRAM_MPU_VOLTAGE_CTRL

Address Offset	0x0000 0160		
Physical Address	0x4AE0 C160	**Instance**	CTRL_MODULE_WKUP
Description	MPU SRAM LDO Control register		

Table 18-894. CTRL_WKUP_LDOSRAM_MPU_VOLTAGE_CTRL (continued)

Type	RW

31:27	26	25:21	20:16	15:11	10	9:5	4:0
RESERVED	LDOSRAMMPU_RETMODE_MUX_CTRL	LDOSRAMMPU_RETMODE_VSET_IN	LDOSRAMMPU_RETMODE_VSET_OUT	RESERVED	LDOSRAMMPU_ACTMODE_MUX_CTRL	LDOSRAMMPU_ACTMODE_VSET_IN	LDOSRAMMPU_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMMPU_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMMPU_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMMPU_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMMPU_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMMPU_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMMPU_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-895. CTRL_WKUP_LDOSRAM_CORE_VOLTAGE_CTRL

Address Offset	0x0000 0164		
Physical Address	0x4AE0 C164	**Instance**	CTRL_MODULE_WKUP
Description	Core SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LDOSRAMCORE_RETMODE_MUX_CTRL	LDOSRAMCORE_RETMODE_VSET_IN	LDOSRAMCORE_RETMODE_VSET_OUT	RESERVED	LDOSRAMCORE_ACTMODE_MUX_CTRL	LDOSRAMCORE_ACTMODE_VSET_IN	LDOSRAMCORE_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-896. CTRL_WKUP_LDOSRAM_MPU_2_VOLTAGE_CTRL

Address Offset	0x0000 0168		
Physical Address	0x4AE0 C168	**Instance**	CTRL_MODULE_WKUP
Description	MPU 2nd SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LDOSRAMMPU_2_RETMODE_MUX_CTRL	LDOSRAMMPU_2_RETMODE_VSET_IN	LDOSRAMMPU_2_RETMODE_VSET_OUT	RESERVED	LDOSRAMMPU_2_ACTMODE_MUX_CTRL	LDOSRAMMPU_2_ACTMODE_VSET_IN	LDOSRAMMPU_2_ACTMODE_VSET_OUT

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMMPU_2_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMMPU_2_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMMPU_2_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMMPU_2_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMMPU_2_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMMPU_2_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 18-897. CTRL_WKUP_STD_FUSE_DIE_ID_0

Address Offset	0x0000 0200		
Physical Address	0x4AE0 C200	**Instance**	CTRL_MODULE_WKUP
Description	Die ID Register : Part 0. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_0		R	0x0

Table 18-898. CTRL_WKUP_ID_CODE

Address Offset	0x0000 0204		
Physical Address	0x4AE0 C204	**Instance**	CTRL_MODULE_WKUP
Description	ID_CODE Key Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_IDCODE																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_IDCODE		R	0x0

Table 18-899. CTRL_WKUP_STD_FUSE_DIE_ID_1

Address Offset	0x0000 0208		
Physical Address	0x4AE0 C208	**Instance**	CTRL_MODULE_WKUP
Description	Die ID Register : Part 1. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_1		R	0x0

Table 18-900. CTRL_WKUP_STD_FUSE_DIE_ID_2

Address Offset	0x0000 020C		
Physical Address	0x4AE0 C20C	**Instance**	CTRL_MODULE_WKUP
Description	Die ID Register : Part 2. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_2		R	0x0

Table 18-901. CTRL_WKUP_STD_FUSE_DIE_ID_3

Address Offset	0x0000 0210		
Physical Address	0x4AE0 C210	**Instance**	CTRL_MODULE_WKUP
Description	Die ID Register : Part 3. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_3		R	0x0

Table 18-902. CTRL_WKUP_STD_FUSE_PROD_ID_0

Address Offset	0x0000 0214		
Physical Address	0x4AE0 C214	**Instance**	CTRL_MODULE_WKUP
Description	Prod ID Register : Part 0. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STD_FUSE_PROD_ID

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_PROD_ID		R	0x0

Table 18-903. CTRL_WKUP_CONTROL_XTAL_OSCILLATOR

Address Offset	0x0000 05AC		
Physical Address	0x4AE0 C5AC	**Instance**	CTRL_MODULE_WKUP
Description	XTAL OSCILLATOR control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSCILLATOR0_BOOST	OSCILLATOR0_OS_OUT	OSCILLATOR1_BOOST	OSCILLATOR1_OS_OUT	RESERVED																											

Bits	Field Name	Description	Type	Reset
31	OSCILLATOR0_BOOST	Fast startup control of OSC0 0x0 = Fast startup is disabled 0x1 = Fast startup is enabled	RW	0x1
30	OSCILLATOR0_OS_OUT	Oscillator output of OSC0 0x0 = low to high transition in BOOST mode 0x1 = BOOST is disabled	R	0x0
29	OSCILLATOR1_BOOST	Fast startup control of OSC1 0x0 = Fast startup is disabled 0x1 = Fast startup is enabled	RW	0x1
28	OSCILLATOR1_OS_OUT	Oscillator output of OSC1 0x0 = low to high transition in BOOST mode 0x1 = BOOST is disabled	R	0x0
27:0	RESERVED		R	0x0

Table 18-904. CTRL_WKUP_EFUSE_1

Address Offset	0x0000 05C8		
Physical Address	0x4AE0 C5C8	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DDRDIFF_PTV_NORTH_SIDE_N5	DDRDIFF_PTV_NORTH_SIDE_N4	DDRDIFF_PTV_NORTH_SIDE_N3	DDRDIFF_PTV_NORTH_SIDE_N2	DDRDIFF_PTV_NORTH_SIDE_N1	DDRDIFF_PTV_NORTH_SIDE_N0	DDRDIFF_PTV_NORTH_SIDE_P5	DDRDIFF_PTV_NORTH_SIDE_P4	DDRDIFF_PTV_NORTH_SIDE_P3	DDRDIFF_PTV_NORTH_SIDE_P2	DDRDIFF_PTV_NORTH_SIDE_P1	DDRDIFF_PTV_NORTH_SIDE_P0	DDRDIFF_PTV_EAST_SIDE_N5	DDRDIFF_PTV_EAST_SIDE_N4	DDRDIFF_PTV_EAST_SIDE_N3	DDRDIFF_PTV_EAST_SIDE_N2	DDRDIFF_PTV_EAST_SIDE_N1	DDRDIFF_PTV_EAST_SIDE_N0	DDRDIFF_PTV_EAST_SIDE_P5	DDRDIFF_PTV_EAST_SIDE_P4	DDRDIFF_PTV_EAST_SIDE_P3	DDRDIFF_PTV_EAST_SIDE_P2	DDRDIFF_PTV_EAST_SIDE_P1	DDRDIFF_PTV_EAST_SIDE_P0	RESERVED

Bits	Field Name	Description	Type	Reset
31	DDRDIFF_PTV_NORTH_SIDE_N5		RW	0x0
30	DDRDIFF_PTV_NORTH_SIDE_N4		RW	0x0
29	DDRDIFF_PTV_NORTH_SIDE_N3		RW	0x0
28	DDRDIFF_PTV_NORTH_SIDE_N2		RW	0x0
27	DDRDIFF_PTV_NORTH_SIDE_N1		RW	0x0
26	DDRDIFF_PTV_NORTH_SIDE_N0		RW	0x0
25	DDRDIFF_PTV_NORTH_SIDE_P5		RW	0x0
24	DDRDIFF_PTV_NORTH_SIDE_P4		RW	0x0
23	DDRDIFF_PTV_NORTH_SIDE_P3		RW	0x0
22	DDRDIFF_PTV_NORTH_SIDE_P2		RW	0x0
21	DDRDIFF_PTV_NORTH_SIDE_P1		RW	0x0
20	DDRDIFF_PTV_NORTH_SIDE_P0		RW	0x0
19	DDRDIFF_PTV_EAST_SIDE_N5		RW	0x0
18	DDRDIFF_PTV_EAST_SIDE_N4		RW	0x0
17	DDRDIFF_PTV_EAST_SIDE_N3		RW	0x0
16	DDRDIFF_PTV_EAST_SIDE_N2		RW	0x0
15	DDRDIFF_PTV_EAST_SIDE_N1		RW	0x0
14	DDRDIFF_PTV_EAST_SIDE_N0		RW	0x0
13	DDRDIFF_PTV_EAST_SIDE_P5		RW	0x0
12	DDRDIFF_PTV_EAST_SIDE_P4		RW	0x0

Bits	Field Name	Description	Type	Reset
11	DDRDIFF_PTV_EAST_SIDE_P3		RW	0x0
10	DDRDIFF_PTV_EAST_SIDE_P2		RW	0x0
9	DDRDIFF_PTV_EAST_SIDE_P1		RW	0x0
8	DDRDIFF_PTV_EAST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 18-905. CTRL_WKUP_EFUSE_2

Address Offset	0x0000 05CC		
Physical Address	0x4AE0 C5CC	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRDIFF_PTV_SOUTH_SIDE_N5	DDRDIFF_PTV_SOUTH_SIDE_N4	DDRDIFF_PTV_SOUTH_SIDE_N3	DDRDIFF_PTV_SOUTH_SIDE_N2	DDRDIFF_PTV_SOUTH_SIDE_N1	DDRDIFF_PTV_SOUTH_SIDE_N0	DDRDIFF_PTV_SOUTH_SIDE_P5	DDRDIFF_PTV_SOUTH_SIDE_P4	DDRDIFF_PTV_SOUTH_SIDE_P3	DDRDIFF_PTV_SOUTH_SIDE_P2	DDRDIFF_PTV_SOUTH_SIDE_P1	DDRDIFF_PTV_SOUTH_SIDE_P0	DDRDIFF_PTV_WEST_SIDE_N5	DDRDIFF_PTV_WEST_SIDE_N4	DDRDIFF_PTV_WEST_SIDE_N3	DDRDIFF_PTV_WEST_SIDE_N2	DDRDIFF_PTV_WEST_SIDE_N1	DDRDIFF_PTV_WEST_SIDE_N0	DDRDIFF_PTV_WEST_SIDE_P5	DDRDIFF_PTV_WEST_SIDE_P4	DDRDIFF_PTV_WEST_SIDE_P3	DDRDIFF_PTV_WEST_SIDE_P2	DDRDIFF_PTV_WEST_SIDE_P1	DDRDIFF_PTV_WEST_SIDE_P0	RESERVED							

Bits	Field Name	Description	Type	Reset
31	DDRDIFF_PTV_SOUTH_SIDE_N5		RW	0x0
30	DDRDIFF_PTV_SOUTH_SIDE_N4		RW	0x0
29	DDRDIFF_PTV_SOUTH_SIDE_N3		RW	0x0
28	DDRDIFF_PTV_SOUTH_SIDE_N2		RW	0x0
27	DDRDIFF_PTV_SOUTH_SIDE_N1		RW	0x0
26	DDRDIFF_PTV_SOUTH_SIDE_N0		RW	0x0
25	DDRDIFF_PTV_SOUTH_SIDE_P5		RW	0x0
24	DDRDIFF_PTV_SOUTH_SIDE_P4		RW	0x0
23	DDRDIFF_PTV_SOUTH_SIDE_P3		RW	0x0
22	DDRDIFF_PTV_SOUTH_SIDE_P2		RW	0x0
21	DDRDIFF_PTV_SOUTH_SIDE_P1		RW	0x0

Bits	Field Name	Description	Type	Reset
20	DDRDIFF_PTV_SOUTH_SIDE_P0		RW	0x0
19	DDRDIFF_PTV_WEST_SIDE_N5		RW	0x0
18	DDRDIFF_PTV_WEST_SIDE_N4		RW	0x0
17	DDRDIFF_PTV_WEST_SIDE_N3		RW	0x0
16	DDRDIFF_PTV_WEST_SIDE_N2		RW	0x0
15	DDRDIFF_PTV_WEST_SIDE_N1		RW	0x0
14	DDRDIFF_PTV_WEST_SIDE_N0		RW	0x0
13	DDRDIFF_PTV_WEST_SIDE_P5		RW	0x0
12	DDRDIFF_PTV_WEST_SIDE_P4		RW	0x0
11	DDRDIFF_PTV_WEST_SIDE_P3		RW	0x0
10	DDRDIFF_PTV_WEST_SIDE_P2		RW	0x0
9	DDRDIFF_PTV_WEST_SIDE_P1		RW	0x0
8	DDRDIFF_PTV_WEST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 18-906. CTRL_WKUP_EFUSE_3

Address Offset	0x0000 05D0		
Physical Address	0x4AE0 C5D0	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7:0
DDRSE_PTV_NORTH_SIDE_N5	DDRSE_PTV_NORTH_SIDE_N4	DDRSE_PTV_NORTH_SIDE_N3	DDRSE_PTV_NORTH_SIDE_N2	DDRSE_PTV_NORTH_SIDE_N1	DDRSE_PTV_NORTH_SIDE_N0	DDRSE_PTV_NORTH_SIDE_P5	DDRSE_PTV_NORTH_SIDE_P4	DDRSE_PTV_NORTH_SIDE_P3	DDRSE_PTV_NORTH_SIDE_P2	DDRSE_PTV_NORTH_SIDE_P1	DDRSE_PTV_NORTH_SIDE_P0	DDRSE_PTV_EAST_SIDE_N5	DDRSE_PTV_EAST_SIDE_N4	DDRSE_PTV_EAST_SIDE_N3	DDRSE_PTV_EAST_SIDE_N2	DDRSE_PTV_EAST_SIDE_N1	DDRSE_PTV_EAST_SIDE_N0	DDRSE_PTV_EAST_SIDE_P5	DDRSE_PTV_EAST_SIDE_P4	DDRSE_PTV_EAST_SIDE_P3	DDRSE_PTV_EAST_SIDE_P2	DDRSE_PTV_EAST_SIDE_P1	DDRSE_PTV_EAST_SIDE_P0	RESERVED

Bits	Field Name	Description	Type	Reset
31	DDRSE_PTV_NORTH_SIDE_N5		RW	0x0
30	DDRSE_PTV_NORTH_SIDE_N4		RW	0x0

Bits	Field Name	Description	Type	Reset
29	DDRSE_PTV_NORTH_SIDE_N3		RW	0x0
28	DDRSE_PTV_NORTH_SIDE_N2		RW	0x0
27	DDRSE_PTV_NORTH_SIDE_N1		RW	0x0
26	DDRSE_PTV_NORTH_SIDE_N0		RW	0x0
25	DDRSE_PTV_NORTH_SIDE_P5		RW	0x0
24	DDRSE_PTV_NORTH_SIDE_P4		RW	0x0
23	DDRSE_PTV_NORTH_SIDE_P3		RW	0x0
22	DDRSE_PTV_NORTH_SIDE_P2		RW	0x0
21	DDRSE_PTV_NORTH_SIDE_P1		RW	0x0
20	DDRSE_PTV_NORTH_SIDE_P0		RW	0x0
19	DDRSE_PTV_EAST_SIDE_N5		RW	0x0
18	DDRSE_PTV_EAST_SIDE_N4		RW	0x0
17	DDRSE_PTV_EAST_SIDE_N3		RW	0x0
16	DDRSE_PTV_EAST_SIDE_N2		RW	0x0
15	DDRSE_PTV_EAST_SIDE_N1		RW	0x0
14	DDRSE_PTV_EAST_SIDE_N0		RW	0x0
13	DDRSE_PTV_EAST_SIDE_P5		RW	0x0
12	DDRSE_PTV_EAST_SIDE_P4		RW	0x0
11	DDRSE_PTV_EAST_SIDE_P3		RW	0x0
10	DDRSE_PTV_EAST_SIDE_P2		RW	0x0
9	DDRSE_PTV_EAST_SIDE_P1		RW	0x0
8	DDRSE_PTV_EAST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 18-907. CTRL_WKUP_EFUSE_4

Address Offset	0x0000 05D4		
Physical Address	0x4AE0 C5D4	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DDRSE_PTV_SOUTH_SIDE_N5	DDRSE_PTV_SOUTH_SIDE_N4	DDRSE_PTV_SOUTH_SIDE_N3	DDRSE_PTV_SOUTH_SIDE_N2	DDRSE_PTV_SOUTH_SIDE_N1	DDRSE_PTV_SOUTH_SIDE_N0	DDRSE_PTV_SOUTH_SIDE_P5	DDRSE_PTV_SOUTH_SIDE_P4	DDRSE_PTV_SOUTH_SIDE_P3	DDRSE_PTV_SOUTH_SIDE_P2	DDRSE_PTV_SOUTH_SIDE_P1	DDRSE_PTV_SOUTH_SIDE_P0	DDRSE_PTV_WEST_SIDE_N5	DDRSE_PTV_WEST_SIDE_N4	DDRSE_PTV_WEST_SIDE_N3	DDRSE_PTV_WEST_SIDE_N2	DDRSE_PTV_WEST_SIDE_N1	DDRSE_PTV_WEST_SIDE_N0	DDRSE_PTV_WEST_SIDE_P5	DDRSE_PTV_WEST_SIDE_P4	DDRSE_PTV_WEST_SIDE_P3	DDRSE_PTV_WEST_SIDE_P2	DDRSE_PTV_WEST_SIDE_P1	DDRSE_PTV_WEST_SIDE_P0	RESERVED

Bits	Field Name	Description	Type	Reset
31	DDRSE_PTV_SOUTH_SIDE_N5		RW	0x0
30	DDRSE_PTV_SOUTH_SIDE_N4		RW	0x0
29	DDRSE_PTV_SOUTH_SIDE_N3		RW	0x0
28	DDRSE_PTV_SOUTH_SIDE_N2		RW	0x0
27	DDRSE_PTV_SOUTH_SIDE_N1		RW	0x0
26	DDRSE_PTV_SOUTH_SIDE_N0		RW	0x0
25	DDRSE_PTV_SOUTH_SIDE_P5		RW	0x0
24	DDRSE_PTV_SOUTH_SIDE_P4		RW	0x0
23	DDRSE_PTV_SOUTH_SIDE_P3		RW	0x0
22	DDRSE_PTV_SOUTH_SIDE_P2		RW	0x0
21	DDRSE_PTV_SOUTH_SIDE_P1		RW	0x0
20	DDRSE_PTV_SOUTH_SIDE_P0		RW	0x0
19	DDRSE_PTV_WEST_SIDE_N5		RW	0x0
18	DDRSE_PTV_WEST_SIDE_N4		RW	0x0
17	DDRSE_PTV_WEST_SIDE_N3		RW	0x0
16	DDRSE_PTV_WEST_SIDE_N2		RW	0x0
15	DDRSE_PTV_WEST_SIDE_N1		RW	0x0
14	DDRSE_PTV_WEST_SIDE_N0		RW	0x0
13	DDRSE_PTV_WEST_SIDE_P5		RW	0x0
12	DDRSE_PTV_WEST_SIDE_P4		RW	0x0
11	DDRSE_PTV_WEST_SIDE_P3		RW	0x0

Bits	Field Name	Description	Type	Reset
10	DDRSE_PTV_WEST_SIDE_P2		RW	0x0
9	DDRSE_PTV_WEST_SIDE_P1		RW	0x0
8	DDRSE_PTV_WEST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 18-908. CTRL_WKUP_EFUSE_13

Address Offset	0x0000 05F8		
Physical Address	0x4AE0 C5F8	**Instance**	CTRL_MODULE_WKUP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDIO1833_PTV_N5	SDIO1833_PTV_N4	SDIO1833_PTV_N3	SDIO1833_PTV_N2	SDIO1833_PTV_N1	SDIO1833_PTV_N0	SDIO1833_PTV_P5	SDIO1833_PTV_P4	SDIO1833_PTV_P3	SDIO1833_PTV_P2	SDIO1833_PTV_P1	SDIO1833_PTV_P0	RESERVED																			

Bits	Field Name	Description	Type	Reset
31	SDIO1833_PTV_N5		RW	0x0
30	SDIO1833_PTV_N4		RW	0x0
29	SDIO1833_PTV_N3		RW	0x0
28	SDIO1833_PTV_N2		RW	0x0
27	SDIO1833_PTV_N1		RW	0x0
26	SDIO1833_PTV_N0		RW	0x0
25	SDIO1833_PTV_P5		RW	0x0
24	SDIO1833_PTV_P4		RW	0x0
23	SDIO1833_PTV_P3		RW	0x0
22	SDIO1833_PTV_P2		RW	0x0
21	SDIO1833_PTV_P1		RW	0x0
20	SDIO1833_PTV_P0		RW	0x0
19:0	RESERVED		R	0x0

18.6 IODELAYCONFIG Module Integration

Figure 18-14 shows the integration of the IODELAYCONFIG module in the device.

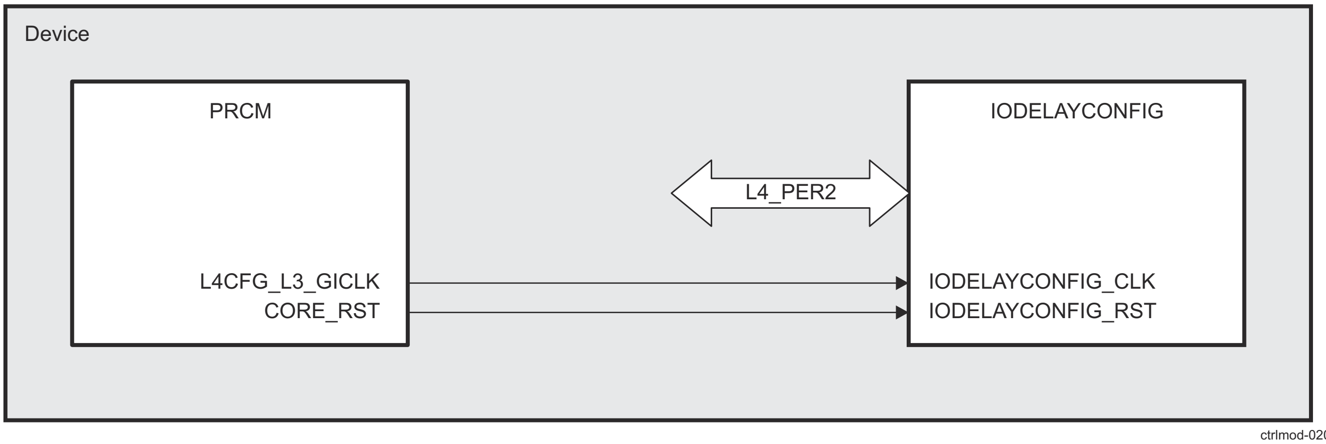


Figure 18-14. IODELAYCONFIG Integration

Table 18-909 and Table 18-910 summarize the integration of the IODELAYCONFIG module in the device.

Table 18-909. IODELAYCONFIG Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
IODELAYCONFIG	PD_COREAON	L4_PER2

Table 18-910. IODELAYCONFIG Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
IODELAYCONFIG	IODELAYCONFIG_CLK	L4CFG_L3_GICLK	PRCM	IODELAYCONFIG module clock
Resets				
IODELAYCONFIG	IODELAYCONFIG_RST	CORE_RST	PRCM	IODELAYCONFIG module reset

18.7 IODELAYCONFIG Module Register Manual

18.7.1 IODELAYCONFIG Module Instance Summary

Table 18-911. IODELAYCONFIG Module Instance Summary

Module Name	Module Base Address	Size
IODELAYCONFIG	0x4844 A000	4KiB

18.7.2 IODELAYCONFIG Registers

18.7.3 IODELAYCONFIG Register Summary

Table 18-912. IODELAYCONFIG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
RESERVED	R	32	0x0000 0000	0x4844 A000
RESERVED	R	32	0x0000 0004	0x4844 A004
RESERVED	R	32	0x0000 0008	0x4844 A008
CONFIG_REG_0	RW	32	0x0000 000C	0x4844 A00C
RESERVED	R	32	0x0000 0010	0x4844 A010
CONFIG_REG_2	RW	32	0x0000 0014	0x4844 A014
CONFIG_REG_3	RW	32	0x0000 0018	0x4844 A018
CONFIG_REG_4	RW	32	0x0000 001C	0x4844 A01C
RESERVED	R	32	0x0000 0020	0x4844 A020
RESERVED	R	32	0x0000 0024	0x4844 A024
RESERVED	R	32	0x0000 0028	0x4844 A028
CONFIG_REG_8	RW	32	0x0000 002C	0x4844 A02C
CFG_RMII_MHZ_50_CLK_IN	RW	32	0x0000 0030	0x4844 A030
CFG_RMII_MHZ_50_CLK_OEN	RW	32	0x0000 0034	0x4844 A034
CFG_RMII_MHZ_50_CLK_OUT	RW	32	0x0000 0038	0x4844 A038
CFG_WAKEUP0_IN	RW	32	0x0000 003C	0x4844 A03C
CFG_WAKEUP0_OEN	RW	32	0x0000 0040	0x4844 A040
CFG_WAKEUP0_OUT	RW	32	0x0000 0044	0x4844 A044
CFG_WAKEUP1_IN	RW	32	0x0000 0048	0x4844 A048
CFG_WAKEUP1_OEN	RW	32	0x0000 004C	0x4844 A04C
CFG_WAKEUP1_OUT	RW	32	0x0000 0050	0x4844 A050
CFG_WAKEUP2_IN	RW	32	0x0000 0054	0x4844 A054
CFG_WAKEUP2_OEN	RW	32	0x0000 0058	0x4844 A058
CFG_WAKEUP2_OUT	RW	32	0x0000 005C	0x4844 A05C
CFG_WAKEUP3_IN	RW	32	0x0000 0060	0x4844 A060
CFG_WAKEUP3_OEN	RW	32	0x0000 0064	0x4844 A064
CFG_WAKEUP3_OUT	RW	32	0x0000 0068	0x4844 A068
CFG_DCAN1_RX_IN	RW	32	0x0000 006C	0x4844 A06C
CFG_DCAN1_RX_OEN	RW	32	0x0000 0070	0x4844 A070
CFG_DCAN1_RX_OUT	RW	32	0x0000 0074	0x4844 A074
CFG_DCAN1_TX_IN	RW	32	0x0000 0078	0x4844 A078
CFG_DCAN1_TX_OEN	RW	32	0x0000 007C	0x4844 A07C
CFG_DCAN1_TX_OUT	RW	32	0x0000 0080	0x4844 A080
CFG_DCAN2_RX_IN	RW	32	0x0000 0084	0x4844 A084
CFG_DCAN2_RX_OEN	RW	32	0x0000 0088	0x4844 A088
CFG_DCAN2_RX_OUT	RW	32	0x0000 008C	0x4844 A08C
CFG_DCAN2_TX_IN	RW	32	0x0000 0090	0x4844 A090
CFG_DCAN2_TX_OEN	RW	32	0x0000 0094	0x4844 A094

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_DCAN2_TX_OUT	RW	32	0x0000 0098	0x4844 A098
CFG_EMU0_IN	RW	32	0x0000 009C	0x4844 A09C
CFG_EMU0_OEN	RW	32	0x0000 00A0	0x4844 A0A0
CFG_EMU0_OUT	RW	32	0x0000 00A4	0x4844 A0A4
CFG_EMU1_IN	RW	32	0x0000 00A8	0x4844 A0A8
CFG_EMU1_OEN	RW	32	0x0000 00AC	0x4844 A0AC
CFG_EMU1_OUT	RW	32	0x0000 00B0	0x4844 A0B0
CFG_EMU2_IN	RW	32	0x0000 00B4	0x4844 A0B4
CFG_EMU2_OEN	RW	32	0x0000 00B8	0x4844 A0B8
CFG_EMU2_OUT	RW	32	0x0000 00BC	0x4844 A0BC
CFG_EMU3_IN	RW	32	0x0000 00C0	0x4844 A0C0
CFG_EMU3_OEN	RW	32	0x0000 00C4	0x4844 A0C4
CFG_EMU3_OUT	RW	32	0x0000 00C8	0x4844 A0C8
CFG_EMU4_IN	RW	32	0x0000 00CC	0x4844 A0CC
CFG_EMU4_OEN	RW	32	0x0000 00D0	0x4844 A0D0
CFG_EMU4_OUT	RW	32	0x0000 00D4	0x4844 A0D4
CFG_GPIO6_10_IN	RW	32	0x0000 00D8	0x4844 A0D8
CFG_GPIO6_10_OEN	RW	32	0x0000 00DC	0x4844 A0DC
CFG_GPIO6_10_OUT	RW	32	0x0000 00E0	0x4844 A0E0
CFG_GPIO6_11_IN	RW	32	0x0000 00E4	0x4844 A0E4
CFG_GPIO6_11_OEN	RW	32	0x0000 00E8	0x4844 A0E8
CFG_GPIO6_11_OUT	RW	32	0x0000 00EC	0x4844 A0EC
CFG_GPIO6_14_IN	RW	32	0x0000 00F0	0x4844 A0F0
CFG_GPIO6_14_OEN	RW	32	0x0000 00F4	0x4844 A0F4
CFG_GPIO6_14_OUT	RW	32	0x0000 00F8	0x4844 A0F8
CFG_GPIO6_15_IN	RW	32	0x0000 00FC	0x4844 A0FC
CFG_GPIO6_15_OEN	RW	32	0x0000 0100	0x4844 A100
CFG_GPIO6_15_OUT	RW	32	0x0000 0104	0x4844 A104
CFG_GPIO6_16_IN	RW	32	0x0000 0108	0x4844 A108
CFG_GPIO6_16_OEN	RW	32	0x0000 010C	0x4844 A10C
CFG_GPIO6_16_OUT	RW	32	0x0000 0110	0x4844 A110
CFG_GPMC_A0_IN	RW	32	0x0000 0114	0x4844 A114
CFG_GPMC_A0_OEN	RW	32	0x0000 0118	0x4844 A118
CFG_GPMC_A0_OUT	RW	32	0x0000 011C	0x4844 A11C
CFG_GPMC_A10_IN	RW	32	0x0000 0120	0x4844 A120
CFG_GPMC_A10_OEN	RW	32	0x0000 0124	0x4844 A124
CFG_GPMC_A10_OUT	RW	32	0x0000 0128	0x4844 A128
CFG_GPMC_A11_IN	RW	32	0x0000 012C	0x4844 A12C
CFG_GPMC_A11_OEN	RW	32	0x0000 0130	0x4844 A130
CFG_GPMC_A11_OUT	RW	32	0x0000 0134	0x4844 A134
CFG_GPMC_A12_IN	RW	32	0x0000 0138	0x4844 A138
CFG_GPMC_A12_OEN	RW	32	0x0000 013C	0x4844 A13C
CFG_GPMC_A12_OUT	RW	32	0x0000 0140	0x4844 A140
CFG_GPMC_A13_IN	RW	32	0x0000 0144	0x4844 A144
CFG_GPMC_A13_OEN	RW	32	0x0000 0148	0x4844 A148
CFG_GPMC_A13_OUT	RW	32	0x0000 014C	0x4844 A14C
CFG_GPMC_A14_IN	RW	32	0x0000 0150	0x4844 A150
CFG_GPMC_A14_OEN	RW	32	0x0000 0154	0x4844 A154
CFG_GPMC_A14_OUT	RW	32	0x0000 0158	0x4844 A158

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_GPMC_A15_IN	RW	32	0x0000 015C	0x4844 A15C
CFG_GPMC_A15_OEN	RW	32	0x0000 0160	0x4844 A160
CFG_GPMC_A15_OUT	RW	32	0x0000 0164	0x4844 A164
CFG_GPMC_A16_IN	RW	32	0x0000 0168	0x4844 A168
CFG_GPMC_A16_OEN	RW	32	0x0000 016C	0x4844 A16C
CFG_GPMC_A16_OUT	RW	32	0x0000 0170	0x4844 A170
CFG_GPMC_A17_IN	RW	32	0x0000 0174	0x4844 A174
CFG_GPMC_A17_OEN	RW	32	0x0000 0178	0x4844 A178
CFG_GPMC_A17_OUT	RW	32	0x0000 017C	0x4844 A17C
CFG_GPMC_A18_IN	RW	32	0x0000 0180	0x4844 A180
CFG_GPMC_A18_OEN	RW	32	0x0000 0184	0x4844 A184
CFG_GPMC_A18_OUT	RW	32	0x0000 0188	0x4844 A188
CFG_GPMC_A19_IN	RW	32	0x0000 018C	0x4844 A18C
CFG_GPMC_A19_OEN	RW	32	0x0000 0190	0x4844 A190
CFG_GPMC_A19_OUT	RW	32	0x0000 0194	0x4844 A194
CFG_GPMC_A1_IN	RW	32	0x0000 0198	0x4844 A198
CFG_GPMC_A1_OEN	RW	32	0x0000 019C	0x4844 A19C
CFG_GPMC_A1_OUT	RW	32	0x0000 01A0	0x4844 A1A0
CFG_GPMC_A20_IN	RW	32	0x0000 01A4	0x4844 A1A4
CFG_GPMC_A20_OEN	RW	32	0x0000 01A8	0x4844 A1A8
CFG_GPMC_A20_OUT	RW	32	0x0000 01AC	0x4844 A1AC
CFG_GPMC_A21_IN	RW	32	0x0000 01B0	0x4844 A1B0
CFG_GPMC_A21_OEN	RW	32	0x0000 01B4	0x4844 A1B4
CFG_GPMC_A21_OUT	RW	32	0x0000 01B8	0x4844 A1B8
CFG_GPMC_A22_IN	RW	32	0x0000 01BC	0x4844 A1BC
CFG_GPMC_A22_OEN	RW	32	0x0000 01C0	0x4844 A1C0
CFG_GPMC_A22_OUT	RW	32	0x0000 01C4	0x4844 A1C4
CFG_GPMC_A23_IN	RW	32	0x0000 01C8	0x4844 A1C8
CFG_GPMC_A23_OEN	RW	32	0x0000 01CC	0x4844 A1CC
CFG_GPMC_A23_OUT	RW	32	0x0000 01D0	0x4844 A1D0
CFG_GPMC_A24_IN	RW	32	0x0000 01D4	0x4844 A1D4
CFG_GPMC_A24_OEN	RW	32	0x0000 01D8	0x4844 A1D8
CFG_GPMC_A24_OUT	RW	32	0x0000 01DC	0x4844 A1DC
CFG_GPMC_A25_IN	RW	32	0x0000 01E0	0x4844 A1E0
CFG_GPMC_A25_OEN	RW	32	0x0000 01E4	0x4844 A1E4
CFG_GPMC_A25_OUT	RW	32	0x0000 01E8	0x4844 A1E8
CFG_GPMC_A26_IN	RW	32	0x0000 01EC	0x4844 A1EC
CFG_GPMC_A26_OEN	RW	32	0x0000 01F0	0x4844 A1F0
CFG_GPMC_A26_OUT	RW	32	0x0000 01F4	0x4844 A1F4
CFG_GPMC_A27_IN	RW	32	0x0000 01F8	0x4844 A1F8
CFG_GPMC_A27_OEN	RW	32	0x0000 01FC	0x4844 A1FC
CFG_GPMC_A27_OUT	RW	32	0x0000 0200	0x4844 A200
CFG_GPMC_A2_IN	RW	32	0x0000 0204	0x4844 A204
CFG_GPMC_A2_OEN	RW	32	0x0000 0208	0x4844 A208
CFG_GPMC_A2_OUT	RW	32	0x0000 020C	0x4844 A20C
CFG_GPMC_A3_IN	RW	32	0x0000 0210	0x4844 A210
CFG_GPMC_A3_OEN	RW	32	0x0000 0214	0x4844 A214
CFG_GPMC_A3_OUT	RW	32	0x0000 0218	0x4844 A218
CFG_GPMC_A4_IN	RW	32	0x0000 021C	0x4844 A21C

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_GPMC_A4_OEN	RW	32	0x0000 0220	0x4844 A220
CFG_GPMC_A4_OUT	RW	32	0x0000 0224	0x4844 A224
CFG_GPMC_A5_IN	RW	32	0x0000 0228	0x4844 A228
CFG_GPMC_A5_OEN	RW	32	0x0000 022C	0x4844 A22C
CFG_GPMC_A5_OUT	RW	32	0x0000 0230	0x4844 A230
CFG_GPMC_A6_IN	RW	32	0x0000 0234	0x4844 A234
CFG_GPMC_A6_OEN	RW	32	0x0000 0238	0x4844 A238
CFG_GPMC_A6_OUT	RW	32	0x0000 023C	0x4844 A23C
CFG_GPMC_A7_IN	RW	32	0x0000 0240	0x4844 A240
CFG_GPMC_A7_OEN	RW	32	0x0000 0244	0x4844 A244
CFG_GPMC_A7_OUT	RW	32	0x0000 0248	0x4844 A248
CFG_GPMC_A8_IN	RW	32	0x0000 024C	0x4844 A24C
CFG_GPMC_A8_OEN	RW	32	0x0000 0250	0x4844 A250
CFG_GPMC_A8_OUT	RW	32	0x0000 0254	0x4844 A254
CFG_GPMC_A9_IN	RW	32	0x0000 0258	0x4844 A258
CFG_GPMC_A9_OEN	RW	32	0x0000 025C	0x4844 A25C
CFG_GPMC_A9_OUT	RW	32	0x0000 0260	0x4844 A260
CFG_GPMC_AD0_IN	RW	32	0x0000 0264	0x4844 A264
CFG_GPMC_AD0_OEN	RW	32	0x0000 0268	0x4844 A268
CFG_GPMC_AD0_OUT	RW	32	0x0000 026C	0x4844 A26C
CFG_GPMC_AD10_IN	RW	32	0x0000 0270	0x4844 A270
CFG_GPMC_AD10_OEN	RW	32	0x0000 0274	0x4844 A274
CFG_GPMC_AD10_OUT	RW	32	0x0000 0278	0x4844 A278
CFG_GPMC_AD11_IN	RW	32	0x0000 027C	0x4844 A27C
CFG_GPMC_AD11_OEN	RW	32	0x0000 0280	0x4844 A280
CFG_GPMC_AD11_OUT	RW	32	0x0000 0284	0x4844 A284
CFG_GPMC_AD12_IN	RW	32	0x0000 0288	0x4844 A288
CFG_GPMC_AD12_OEN	RW	32	0x0000 028C	0x4844 A28C
CFG_GPMC_AD12_OUT	RW	32	0x0000 0290	0x4844 A290
CFG_GPMC_AD13_IN	RW	32	0x0000 0294	0x4844 A294
CFG_GPMC_AD13_OEN	RW	32	0x0000 0298	0x4844 A298
CFG_GPMC_AD13_OUT	RW	32	0x0000 029C	0x4844 A29C
CFG_GPMC_AD14_IN	RW	32	0x0000 02A0	0x4844 A2A0
CFG_GPMC_AD14_OEN	RW	32	0x0000 02A4	0x4844 A2A4
CFG_GPMC_AD14_OUT	RW	32	0x0000 02A8	0x4844 A2A8
CFG_GPMC_AD15_IN	RW	32	0x0000 02AC	0x4844 A2AC
CFG_GPMC_AD15_OEN	RW	32	0x0000 02B0	0x4844 A2B0
CFG_GPMC_AD15_OUT	RW	32	0x0000 02B4	0x4844 A2B4
CFG_GPMC_AD1_IN	RW	32	0x0000 02B8	0x4844 A2B8
CFG_GPMC_AD1_OEN	RW	32	0x0000 02BC	0x4844 A2BC
CFG_GPMC_AD1_OUT	RW	32	0x0000 02C0	0x4844 A2C0
CFG_GPMC_AD2_IN	RW	32	0x0000 02C4	0x4844 A2C4
CFG_GPMC_AD2_OEN	RW	32	0x0000 02C8	0x4844 A2C8
CFG_GPMC_AD2_OUT	RW	32	0x0000 02CC	0x4844 A2CC
CFG_GPMC_AD3_IN	RW	32	0x0000 02D0	0x4844 A2D0
CFG_GPMC_AD3_OEN	RW	32	0x0000 02D4	0x4844 A2D4
CFG_GPMC_AD3_OUT	RW	32	0x0000 02D8	0x4844 A2D8
CFG_GPMC_AD4_IN	RW	32	0x0000 02DC	0x4844 A2DC
CFG_GPMC_AD4_OEN	RW	32	0x0000 02E0	0x4844 A2E0

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_GPMC_AD4_OUT	RW	32	0x0000 02E4	0x4844 A2E4
CFG_GPMC_AD5_IN	RW	32	0x0000 02E8	0x4844 A2E8
CFG_GPMC_AD5_OEN	RW	32	0x0000 02EC	0x4844 A2EC
CFG_GPMC_AD5_OUT	RW	32	0x0000 02F0	0x4844 A2F0
CFG_GPMC_AD6_IN	RW	32	0x0000 02F4	0x4844 A2F4
CFG_GPMC_AD6_OEN	RW	32	0x0000 02F8	0x4844 A2F8
CFG_GPMC_AD6_OUT	RW	32	0x0000 02FC	0x4844 A2FC
CFG_GPMC_AD7_IN	RW	32	0x0000 0300	0x4844 A300
CFG_GPMC_AD7_OEN	RW	32	0x0000 0304	0x4844 A304
CFG_GPMC_AD7_OUT	RW	32	0x0000 0308	0x4844 A308
CFG_GPMC_AD8_IN	RW	32	0x0000 030C	0x4844 A30C
CFG_GPMC_AD8_OEN	RW	32	0x0000 0310	0x4844 A310
CFG_GPMC_AD8_OUT	RW	32	0x0000 0314	0x4844 A314
CFG_GPMC_AD9_IN	RW	32	0x0000 0318	0x4844 A318
CFG_GPMC_AD9_OEN	RW	32	0x0000 031C	0x4844 A31C
CFG_GPMC_AD9_OUT	RW	32	0x0000 0320	0x4844 A320
CFG_GPMC_ADVN_ALE_IN	RW	32	0x0000 0324	0x4844 A324
CFG_GPMC_ADVN_ALE_OEN	RW	32	0x0000 0328	0x4844 A328
CFG_GPMC_ADVN_ALE_OUT	RW	32	0x0000 032C	0x4844 A32C
CFG_GPMC_BEN0_IN	RW	32	0x0000 0330	0x4844 A330
CFG_GPMC_BEN0_OEN	RW	32	0x0000 0334	0x4844 A334
CFG_GPMC_BEN0_OUT	RW	32	0x0000 0338	0x4844 A338
CFG_GPMC_BEN1_IN	RW	32	0x0000 033C	0x4844 A33C
CFG_GPMC_BEN1_OEN	RW	32	0x0000 0340	0x4844 A340
CFG_GPMC_BEN1_OUT	RW	32	0x0000 0344	0x4844 A344
CFG_GPMC_CLK_IN	RW	32	0x0000 0348	0x4844 A348
CFG_GPMC_CLK_OEN	RW	32	0x0000 034C	0x4844 A34C
CFG_GPMC_CLK_OUT	RW	32	0x0000 0350	0x4844 A350
CFG_GPMC_CS0_IN	RW	32	0x0000 0354	0x4844 A354
CFG_GPMC_CS0_OEN	RW	32	0x0000 0358	0x4844 A358
CFG_GPMC_CS0_OUT	RW	32	0x0000 035C	0x4844 A35C
CFG_GPMC_CS1_IN	RW	32	0x0000 0360	0x4844 A360
CFG_GPMC_CS1_OEN	RW	32	0x0000 0364	0x4844 A364
CFG_GPMC_CS1_OUT	RW	32	0x0000 0368	0x4844 A368
CFG_GPMC_CS2_IN	RW	32	0x0000 036C	0x4844 A36C
CFG_GPMC_CS2_OEN	RW	32	0x0000 0370	0x4844 A370
CFG_GPMC_CS2_OUT	RW	32	0x0000 0374	0x4844 A374
CFG_GPMC_CS3_IN	RW	32	0x0000 0378	0x4844 A378
CFG_GPMC_CS3_OEN	RW	32	0x0000 037C	0x4844 A37C
CFG_GPMC_CS3_OUT	RW	32	0x0000 0380	0x4844 A380
CFG_GPMC_OEN_REN_IN	RW	32	0x0000 0384	0x4844 A384
CFG_GPMC_OEN_REN_OEN	RW	32	0x0000 0388	0x4844 A388
CFG_GPMC_OEN_REN_OUT	RW	32	0x0000 038C	0x4844 A38C
CFG_GPMC_WAIT0_IN	RW	32	0x0000 0390	0x4844 A390
CFG_GPMC_WAIT0_OEN	RW	32	0x0000 0394	0x4844 A394
CFG_GPMC_WAIT0_OUT	RW	32	0x0000 0398	0x4844 A398
CFG_GPMC_WEN_IN	RW	32	0x0000 039C	0x4844 A39C
CFG_GPMC_WEN_OEN	RW	32	0x0000 03A0	0x4844 A3A0
CFG_GPMC_WEN_OUT	RW	32	0x0000 03A4	0x4844 A3A4

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_MCASP1_ACLKR_IN	RW	32	0x0000 03A8	0x4844 A3A8
CFG_MCASP1_ACLKR_OEN	RW	32	0x0000 03AC	0x4844 A3AC
CFG_MCASP1_ACLKR_OUT	RW	32	0x0000 03B0	0x4844 A3B0
CFG_MCASP1_ACLKX_IN	RW	32	0x0000 03B4	0x4844 A3B4
CFG_MCASP1_ACLKX_OEN	RW	32	0x0000 03B8	0x4844 A3B8
CFG_MCASP1_ACLKX_OUT	RW	32	0x0000 03BC	0x4844 A3BC
CFG_MCASP1_AXR0_IN	RW	32	0x0000 03C0	0x4844 A3C0
CFG_MCASP1_AXR0_OEN	RW	32	0x0000 03C4	0x4844 A3C4
CFG_MCASP1_AXR0_OUT	RW	32	0x0000 03C8	0x4844 A3C8
CFG_MCASP1_AXR10_IN	RW	32	0x0000 03CC	0x4844 A3CC
CFG_MCASP1_AXR10_OEN	RW	32	0x0000 03D0	0x4844 A3D0
CFG_MCASP1_AXR10_OUT	RW	32	0x0000 03D4	0x4844 A3D4
CFG_MCASP1_AXR11_IN	RW	32	0x0000 03D8	0x4844 A3D8
CFG_MCASP1_AXR11_OEN	RW	32	0x0000 03DC	0x4844 A3DC
CFG_MCASP1_AXR11_OUT	RW	32	0x0000 03E0	0x4844 A3E0
CFG_MCASP1_AXR12_IN	RW	32	0x0000 03E4	0x4844 A3E4
CFG_MCASP1_AXR12_OEN	RW	32	0x0000 03E8	0x4844 A3E8
CFG_MCASP1_AXR12_OUT	RW	32	0x0000 03EC	0x4844 A3EC
CFG_MCASP1_AXR13_IN	RW	32	0x0000 03F0	0x4844 A3F0
CFG_MCASP1_AXR13_OEN	RW	32	0x0000 03F4	0x4844 A3F4
CFG_MCASP1_AXR13_OUT	RW	32	0x0000 03F8	0x4844 A3F8
CFG_MCASP1_AXR14_IN	RW	32	0x0000 03FC	0x4844 A3FC
CFG_MCASP1_AXR14_OEN	RW	32	0x0000 0400	0x4844 A400
CFG_MCASP1_AXR14_OUT	RW	32	0x0000 0404	0x4844 A404
CFG_MCASP1_AXR15_IN	RW	32	0x0000 0408	0x4844 A408
CFG_MCASP1_AXR15_OEN	RW	32	0x0000 040C	0x4844 A40C
CFG_MCASP1_AXR15_OUT	RW	32	0x0000 0410	0x4844 A410
CFG_MCASP1_AXR1_IN	RW	32	0x0000 0414	0x4844 A414
CFG_MCASP1_AXR1_OEN	RW	32	0x0000 0418	0x4844 A418
CFG_MCASP1_AXR1_OUT	RW	32	0x0000 041C	0x4844 A41C
CFG_MCASP1_AXR2_IN	RW	32	0x0000 0420	0x4844 A420
CFG_MCASP1_AXR2_OEN	RW	32	0x0000 0424	0x4844 A424
CFG_MCASP1_AXR2_OUT	RW	32	0x0000 0428	0x4844 A428
CFG_MCASP1_AXR3_IN	RW	32	0x0000 042C	0x4844 A42C
CFG_MCASP1_AXR3_OEN	RW	32	0x0000 0430	0x4844 A430
CFG_MCASP1_AXR3_OUT	RW	32	0x0000 0434	0x4844 A434
CFG_MCASP1_AXR4_IN	RW	32	0x0000 0438	0x4844 A438
CFG_MCASP1_AXR4_OEN	RW	32	0x0000 043C	0x4844 A43C
CFG_MCASP1_AXR4_OUT	RW	32	0x0000 0440	0x4844 A440
CFG_MCASP1_AXR5_IN	RW	32	0x0000 0444	0x4844 A444
CFG_MCASP1_AXR5_OEN	RW	32	0x0000 0448	0x4844 A448
CFG_MCASP1_AXR5_OUT	RW	32	0x0000 044C	0x4844 A44C
CFG_MCASP1_AXR6_IN	RW	32	0x0000 0450	0x4844 A450
CFG_MCASP1_AXR6_OEN	RW	32	0x0000 0454	0x4844 A454
CFG_MCASP1_AXR6_OUT	RW	32	0x0000 0458	0x4844 A458
CFG_MCASP1_AXR7_IN	RW	32	0x0000 045C	0x4844 A45C
CFG_MCASP1_AXR7_OEN	RW	32	0x0000 0460	0x4844 A460
CFG_MCASP1_AXR7_OUT	RW	32	0x0000 0464	0x4844 A464
CFG_MCASP1_AXR8_IN	RW	32	0x0000 0468	0x4844 A468

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_MCASP1_AXR8_OEN	RW	32	0x0000 046C	0x4844 A46C
CFG_MCASP1_AXR8_OUT	RW	32	0x0000 0470	0x4844 A470
CFG_MCASP1_AXR9_IN	RW	32	0x0000 0474	0x4844 A474
CFG_MCASP1_AXR9_OEN	RW	32	0x0000 0478	0x4844 A478
CFG_MCASP1_AXR9_OUT	RW	32	0x0000 047C	0x4844 A47C
CFG_MCASP1_FSR_IN	RW	32	0x0000 0480	0x4844 A480
CFG_MCASP1_FSR_OEN	RW	32	0x0000 0484	0x4844 A484
CFG_MCASP1_FSR_OUT	RW	32	0x0000 0488	0x4844 A488
CFG_MCASP1_FSX_IN	RW	32	0x0000 048C	0x4844 A48C
CFG_MCASP1_FSX_OEN	RW	32	0x0000 0490	0x4844 A490
CFG_MCASP1_FSX_OUT	RW	32	0x0000 0494	0x4844 A494
CFG_MCASP2_ACLKR_IN	RW	32	0x0000 0498	0x4844 A498
CFG_MCASP2_ACLKR_OEN	RW	32	0x0000 049C	0x4844 A49C
CFG_MCASP2_ACLKR_OUT	RW	32	0x0000 04A0	0x4844 A4A0
CFG_MCASP2_ACLKX_IN	RW	32	0x0000 04A4	0x4844 A4A4
CFG_MCASP2_ACLKX_OEN	RW	32	0x0000 04A8	0x4844 A4A8
CFG_MCASP2_ACLKX_OUT	RW	32	0x0000 04AC	0x4844 A4AC
CFG_MCASP2_AXR0_IN	RW	32	0x0000 04B0	0x4844 A4B0
CFG_MCASP2_AXR0_OEN	RW	32	0x0000 04B4	0x4844 A4B4
CFG_MCASP2_AXR0_OUT	RW	32	0x0000 04B8	0x4844 A4B8
CFG_MCASP2_AXR1_IN	RW	32	0x0000 04BC	0x4844 A4BC
CFG_MCASP2_AXR1_OEN	RW	32	0x0000 04C0	0x4844 A4C0
CFG_MCASP2_AXR1_OUT	RW	32	0x0000 04C4	0x4844 A4C4
CFG_MCASP2_AXR2_IN	RW	32	0x0000 04C8	0x4844 A4C8
CFG_MCASP2_AXR2_OEN	RW	32	0x0000 04CC	0x4844 A4CC
CFG_MCASP2_AXR2_OUT	RW	32	0x0000 04D0	0x4844 A4D0
CFG_MCASP2_AXR3_IN	RW	32	0x0000 04D4	0x4844 A4D4
CFG_MCASP2_AXR3_OEN	RW	32	0x0000 04D8	0x4844 A4D8
CFG_MCASP2_AXR3_OUT	RW	32	0x0000 04DC	0x4844 A4DC
CFG_MCASP2_AXR4_IN	RW	32	0x0000 04E0	0x4844 A4E0
CFG_MCASP2_AXR4_OEN	RW	32	0x0000 04E4	0x4844 A4E4
CFG_MCASP2_AXR4_OUT	RW	32	0x0000 04E8	0x4844 A4E8
CFG_MCASP2_AXR5_IN	RW	32	0x0000 04EC	0x4844 A4EC
CFG_MCASP2_AXR5_OEN	RW	32	0x0000 04F0	0x4844 A4F0
CFG_MCASP2_AXR5_OUT	RW	32	0x0000 04F4	0x4844 A4F4
CFG_MCASP2_AXR6_IN	RW	32	0x0000 04F8	0x4844 A4F8
CFG_MCASP2_AXR6_OEN	RW	32	0x0000 04FC	0x4844 A4FC
CFG_MCASP2_AXR6_OUT	RW	32	0x0000 0500	0x4844 A500
CFG_MCASP2_AXR7_IN	RW	32	0x0000 0504	0x4844 A504
CFG_MCASP2_AXR7_OEN	RW	32	0x0000 0508	0x4844 A508
CFG_MCASP2_AXR7_OUT	RW	32	0x0000 050C	0x4844 A50C
CFG_MCASP2_FSR_IN	RW	32	0x0000 0510	0x4844 A510
CFG_MCASP2_FSR_OEN	RW	32	0x0000 0514	0x4844 A514
CFG_MCASP2_FSR_OUT	RW	32	0x0000 0518	0x4844 A518
CFG_MCASP2_FSX_IN	RW	32	0x0000 051C	0x4844 A51C
CFG_MCASP2_FSX_OEN	RW	32	0x0000 0520	0x4844 A520
CFG_MCASP2_FSX_OUT	RW	32	0x0000 0524	0x4844 A524
CFG_MCASP3_ACLKX_IN	RW	32	0x0000 0528	0x4844 A528
CFG_MCASP3_ACLKX_OEN	RW	32	0x0000 052C	0x4844 A52C

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_MCASP3_ACLKX_OUT	RW	32	0x0000 0530	0x4844 A530
CFG_MCASP3_AXR0_IN	RW	32	0x0000 0534	0x4844 A534
CFG_MCASP3_AXR0_OEN	RW	32	0x0000 0538	0x4844 A538
CFG_MCASP3_AXR0_OUT	RW	32	0x0000 053C	0x4844 A53C
CFG_MCASP3_AXR1_IN	RW	32	0x0000 0540	0x4844 A540
CFG_MCASP3_AXR1_OEN	RW	32	0x0000 0544	0x4844 A544
CFG_MCASP3_AXR1_OUT	RW	32	0x0000 0548	0x4844 A548
CFG_MCASP3_FSX_IN	RW	32	0x0000 054C	0x4844 A54C
CFG_MCASP3_FSX_OEN	RW	32	0x0000 0550	0x4844 A550
CFG_MCASP3_FSX_OUT	RW	32	0x0000 0554	0x4844 A554
CFG_MCASP4_ACLKX_IN	RW	32	0x0000 0558	0x4844 A558
CFG_MCASP4_ACLKX_OEN	RW	32	0x0000 055C	0x4844 A55C
CFG_MCASP4_ACLKX_OUT	RW	32	0x0000 0560	0x4844 A560
CFG_MCASP4_AXR0_IN	RW	32	0x0000 0564	0x4844 A564
CFG_MCASP4_AXR0_OEN	RW	32	0x0000 0568	0x4844 A568
CFG_MCASP4_AXR0_OUT	RW	32	0x0000 056C	0x4844 A56C
CFG_MCASP4_AXR1_IN	RW	32	0x0000 0570	0x4844 A570
CFG_MCASP4_AXR1_OEN	RW	32	0x0000 0574	0x4844 A574
CFG_MCASP4_AXR1_OUT	RW	32	0x0000 0578	0x4844 A578
CFG_MCASP4_FSX_IN	RW	32	0x0000 057C	0x4844 A57C
CFG_MCASP4_FSX_OEN	RW	32	0x0000 0580	0x4844 A580
CFG_MCASP4_FSX_OUT	RW	32	0x0000 0584	0x4844 A584
CFG_MCASP5_ACLKX_IN	RW	32	0x0000 0588	0x4844 A588
CFG_MCASP5_ACLKX_OEN	RW	32	0x0000 058C	0x4844 A58C
CFG_MCASP5_ACLKX_OUT	RW	32	0x0000 0590	0x4844 A590
CFG_MCASP5_AXR0_IN	RW	32	0x0000 0594	0x4844 A594
CFG_MCASP5_AXR0_OEN	RW	32	0x0000 0598	0x4844 A598
CFG_MCASP5_AXR0_OUT	RW	32	0x0000 059C	0x4844 A59C
CFG_MCASP5_AXR1_IN	RW	32	0x0000 05A0	0x4844 A5A0
CFG_MCASP5_AXR1_OEN	RW	32	0x0000 05A4	0x4844 A5A4
CFG_MCASP5_AXR1_OUT	RW	32	0x0000 05A8	0x4844 A5A8
CFG_MCASP5_FSX_IN	RW	32	0x0000 05AC	0x4844 A5AC
CFG_MCASP5_FSX_OEN	RW	32	0x0000 05B0	0x4844 A5B0
CFG_MCASP5_FSX_OUT	RW	32	0x0000 05B4	0x4844 A5B4
CFG_MDIO_D_IN	RW	32	0x0000 05B8	0x4844 A5B8
CFG_MDIO_D_OEN	RW	32	0x0000 05BC	0x4844 A5BC
CFG_MDIO_D_OUT	RW	32	0x0000 05C0	0x4844 A5C0
CFG_MDIO_MCLK_IN	RW	32	0x0000 05C4	0x4844 A5C4
CFG_MDIO_MCLK_OEN	RW	32	0x0000 05C8	0x4844 A5C8
CFG_MDIO_MCLK_OUT	RW	32	0x0000 05CC	0x4844 A5CC
CFG_MLBP_CLK_N_IN	RW	32	0x0000 05D0	0x4844 A5D0
CFG_MLBP_CLK_N_OEN	RW	32	0x0000 05D4	0x4844 A5D4
CFG_MLBP_CLK_N_OUT	RW	32	0x0000 05D8	0x4844 A5D8
CFG_MLBP_CLK_P_IN	RW	32	0x0000 05DC	0x4844 A5DC
CFG_MLBP_CLK_P_OEN	RW	32	0x0000 05E0	0x4844 A5E0
CFG_MLBP_CLK_P_OUT	RW	32	0x0000 05E4	0x4844 A5E4
CFG_MLBP_DAT_N_IN	RW	32	0x0000 05E8	0x4844 A5E8
CFG_MLBP_DAT_N_OEN	RW	32	0x0000 05EC	0x4844 A5EC
CFG_MLBP_DAT_N_OUT	RW	32	0x0000 05F0	0x4844 A5F0

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_MLBP_DAT_P_IN	RW	32	0x0000 05F4	0x4844 A5F4
CFG_MLBP_DAT_P_OEN	RW	32	0x0000 05F8	0x4844 A5F8
CFG_MLBP_DAT_P_OUT	RW	32	0x0000 05FC	0x4844 A5FC
CFG_MLBP_SIG_N_IN	RW	32	0x0000 0600	0x4844 A600
CFG_MLBP_SIG_N_OEN	RW	32	0x0000 0604	0x4844 A604
CFG_MLBP_SIG_N_OUT	RW	32	0x0000 0608	0x4844 A608
CFG_MLBP_SIG_P_IN	RW	32	0x0000 060C	0x4844 A60C
CFG_MLBP_SIG_P_OEN	RW	32	0x0000 0610	0x4844 A610
CFG_MLBP_SIG_P_OUT	RW	32	0x0000 0614	0x4844 A614
CFG_MMC1_CLK_IN	RW	32	0x0000 0618	0x4844 A618
CFG_MMC1_CLK_OEN	RW	32	0x0000 061C	0x4844 A61C
CFG_MMC1_CLK_OUT	RW	32	0x0000 0620	0x4844 A620
CFG_MMC1_CMD_IN	RW	32	0x0000 0624	0x4844 A624
CFG_MMC1_CMD_OEN	RW	32	0x0000 0628	0x4844 A628
CFG_MMC1_CMD_OUT	RW	32	0x0000 062C	0x4844 A62C
CFG_MMC1_DAT0_IN	RW	32	0x0000 0630	0x4844 A630
CFG_MMC1_DAT0_OEN	RW	32	0x0000 0634	0x4844 A634
CFG_MMC1_DAT0_OUT	RW	32	0x0000 0638	0x4844 A638
CFG_MMC1_DAT1_IN	RW	32	0x0000 063C	0x4844 A63C
CFG_MMC1_DAT1_OEN	RW	32	0x0000 0640	0x4844 A640
CFG_MMC1_DAT1_OUT	RW	32	0x0000 0644	0x4844 A644
CFG_MMC1_DAT2_IN	RW	32	0x0000 0648	0x4844 A648
CFG_MMC1_DAT2_OEN	RW	32	0x0000 064C	0x4844 A64C
CFG_MMC1_DAT2_OUT	RW	32	0x0000 0650	0x4844 A650
CFG_MMC1_DAT3_IN	RW	32	0x0000 0654	0x4844 A654
CFG_MMC1_DAT3_OEN	RW	32	0x0000 0658	0x4844 A658
CFG_MMC1_DAT3_OUT	RW	32	0x0000 065C	0x4844 A65C
CFG_MMC1_SDCD_IN	RW	32	0x0000 0660	0x4844 A660
CFG_MMC1_SDCD_OEN	RW	32	0x0000 0664	0x4844 A664
CFG_MMC1_SDCD_OUT	RW	32	0x0000 0668	0x4844 A668
CFG_MMC1_SDWP_IN	RW	32	0x0000 066C	0x4844 A66C
CFG_MMC1_SDWP_OEN	RW	32	0x0000 0670	0x4844 A670
CFG_MMC1_SDWP_OUT	RW	32	0x0000 0674	0x4844 A674
CFG_MMC3_CLK_IN	RW	32	0x0000 0678	0x4844 A678
CFG_MMC3_CLK_OEN	RW	32	0x0000 067C	0x4844 A67C
CFG_MMC3_CLK_OUT	RW	32	0x0000 0680	0x4844 A680
CFG_MMC3_CMD_IN	RW	32	0x0000 0684	0x4844 A684
CFG_MMC3_CMD_OEN	RW	32	0x0000 0688	0x4844 A688
CFG_MMC3_CMD_OUT	RW	32	0x0000 068C	0x4844 A68C
CFG_MMC3_DAT0_IN	RW	32	0x0000 0690	0x4844 A690
CFG_MMC3_DAT0_OEN	RW	32	0x0000 0694	0x4844 A694
CFG_MMC3_DAT0_OUT	RW	32	0x0000 0698	0x4844 A698
CFG_MMC3_DAT1_IN	RW	32	0x0000 069C	0x4844 A69C
CFG_MMC3_DAT1_OEN	RW	32	0x0000 06A0	0x4844 A6A0
CFG_MMC3_DAT1_OUT	RW	32	0x0000 06A4	0x4844 A6A4
CFG_MMC3_DAT2_IN	RW	32	0x0000 06A8	0x4844 A6A8
CFG_MMC3_DAT2_OEN	RW	32	0x0000 06AC	0x4844 A6AC
CFG_MMC3_DAT2_OUT	RW	32	0x0000 06B0	0x4844 A6B0
CFG_MMC3_DAT3_IN	RW	32	0x0000 06B4	0x4844 A6B4

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_MMC3_DAT3_OEN	RW	32	0x0000 06B8	0x4844 A6B8
CFG_MMC3_DAT3_OUT	RW	32	0x0000 06BC	0x4844 A6BC
CFG_MMC3_DAT4_IN	RW	32	0x0000 06C0	0x4844 A6C0
CFG_MMC3_DAT4_OEN	RW	32	0x0000 06C4	0x4844 A6C4
CFG_MMC3_DAT4_OUT	RW	32	0x0000 06C8	0x4844 A6C8
CFG_MMC3_DAT5_IN	RW	32	0x0000 06CC	0x4844 A6CC
CFG_MMC3_DAT5_OEN	RW	32	0x0000 06D0	0x4844 A6D0
CFG_MMC3_DAT5_OUT	RW	32	0x0000 06D4	0x4844 A6D4
CFG_MMC3_DAT6_IN	RW	32	0x0000 06D8	0x4844 A6D8
CFG_MMC3_DAT6_OEN	RW	32	0x0000 06DC	0x4844 A6DC
CFG_MMC3_DAT6_OUT	RW	32	0x0000 06E0	0x4844 A6E0
CFG_MMC3_DAT7_IN	RW	32	0x0000 06E4	0x4844 A6E4
CFG_MMC3_DAT7_OEN	RW	32	0x0000 06E8	0x4844 A6E8
CFG_MMC3_DAT7_OUT	RW	32	0x0000 06EC	0x4844 A6EC
CFG_RGMII0_RXC_IN	RW	32	0x0000 06F0	0x4844 A6F0
CFG_RGMII0_RXC_OEN	RW	32	0x0000 06F4	0x4844 A6F4
CFG_RGMII0_RXC_OUT	RW	32	0x0000 06F8	0x4844 A6F8
CFG_RGMII0_RXCTL_IN	RW	32	0x0000 06FC	0x4844 A6FC
CFG_RGMII0_RXCTL_OEN	RW	32	0x0000 0700	0x4844 A700
CFG_RGMII0_RXCTL_OUT	RW	32	0x0000 0704	0x4844 A704
CFG_RGMII0_RXD0_IN	RW	32	0x0000 0708	0x4844 A708
CFG_RGMII0_RXD0_OEN	RW	32	0x0000 070C	0x4844 A70C
CFG_RGMII0_RXD0_OUT	RW	32	0x0000 0710	0x4844 A710
CFG_RGMII0_RXD1_IN	RW	32	0x0000 0714	0x4844 A714
CFG_RGMII0_RXD1_OEN	RW	32	0x0000 0718	0x4844 A718
CFG_RGMII0_RXD1_OUT	RW	32	0x0000 071C	0x4844 A71C
CFG_RGMII0_RXD2_IN	RW	32	0x0000 0720	0x4844 A720
CFG_RGMII0_RXD2_OEN	RW	32	0x0000 0724	0x4844 A724
CFG_RGMII0_RXD2_OUT	RW	32	0x0000 0728	0x4844 A728
CFG_RGMII0_RXD3_IN	RW	32	0x0000 072C	0x4844 A72C
CFG_RGMII0_RXD3_OEN	RW	32	0x0000 0730	0x4844 A730
CFG_RGMII0_RXD3_OUT	RW	32	0x0000 0734	0x4844 A734
CFG_RGMII0_TXC_IN	RW	32	0x0000 0738	0x4844 A738
CFG_RGMII0_TXC_OEN	RW	32	0x0000 073C	0x4844 A73C
CFG_RGMII0_TXC_OUT	RW	32	0x0000 0740	0x4844 A740
CFG_RGMII0_TXCTL_IN	RW	32	0x0000 0744	0x4844 A744
CFG_RGMII0_TXCTL_OEN	RW	32	0x0000 0748	0x4844 A748
CFG_RGMII0_TXCTL_OUT	RW	32	0x0000 074C	0x4844 A74C
CFG_RGMII0_TXD0_IN	RW	32	0x0000 0750	0x4844 A750
CFG_RGMII0_TXD0_OEN	RW	32	0x0000 0754	0x4844 A754
CFG_RGMII0_TXD0_OUT	RW	32	0x0000 0758	0x4844 A758
CFG_RGMII0_TXD1_IN	RW	32	0x0000 075C	0x4844 A75C
CFG_RGMII0_TXD1_OEN	RW	32	0x0000 0760	0x4844 A760
CFG_RGMII0_TXD1_OUT	RW	32	0x0000 0764	0x4844 A764
CFG_RGMII0_TXD2_IN	RW	32	0x0000 0768	0x4844 A768
CFG_RGMII0_TXD2_OEN	RW	32	0x0000 076C	0x4844 A76C
CFG_RGMII0_TXD2_OUT	RW	32	0x0000 0770	0x4844 A770
CFG_RGMII0_TXD3_IN	RW	32	0x0000 0774	0x4844 A774
CFG_RGMII0_TXD3_OEN	RW	32	0x0000 0778	0x4844 A778

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_RGMII0_TXD3_OUT	RW	32	0x0000 077C	0x4844 A77C
CFG_RTCK_IN	RW	32	0x0000 0780	0x4844 A780
CFG_RTCK_OEN	RW	32	0x0000 0784	0x4844 A784
CFG_RTCK_OUT	RW	32	0x0000 0788	0x4844 A788
CFG_SPI1_CS0_IN	RW	32	0x0000 078C	0x4844 A78C
CFG_SPI1_CS0_OEN	RW	32	0x0000 0790	0x4844 A790
CFG_SPI1_CS0_OUT	RW	32	0x0000 0794	0x4844 A794
CFG_SPI1_CS1_IN	RW	32	0x0000 0798	0x4844 A798
CFG_SPI1_CS1_OEN	RW	32	0x0000 079C	0x4844 A79C
CFG_SPI1_CS1_OUT	RW	32	0x0000 07A0	0x4844 A7A0
CFG_SPI1_CS2_IN	RW	32	0x0000 07A4	0x4844 A7A4
CFG_SPI1_CS2_OEN	RW	32	0x0000 07A8	0x4844 A7A8
CFG_SPI1_CS2_OUT	RW	32	0x0000 07AC	0x4844 A7AC
CFG_SPI1_CS3_IN	RW	32	0x0000 07B0	0x4844 A7B0
CFG_SPI1_CS3_OEN	RW	32	0x0000 07B4	0x4844 A7B4
CFG_SPI1_CS3_OUT	RW	32	0x0000 07B8	0x4844 A7B8
CFG_SPI1_D0_IN	RW	32	0x0000 07BC	0x4844 A7BC
CFG_SPI1_D0_OEN	RW	32	0x0000 07C0	0x4844 A7C0
CFG_SPI1_D0_OUT	RW	32	0x0000 07C4	0x4844 A7C4
CFG_SPI1_D1_IN	RW	32	0x0000 07C8	0x4844 A7C8
CFG_SPI1_D1_OEN	RW	32	0x0000 07CC	0x4844 A7CC
CFG_SPI1_D1_OUT	RW	32	0x0000 07D0	0x4844 A7D0
CFG_SPI1_SCLK_IN	RW	32	0x0000 07D4	0x4844 A7D4
CFG_SPI1_SCLK_OEN	RW	32	0x0000 07D8	0x4844 A7D8
CFG_SPI1_SCLK_OUT	RW	32	0x0000 07DC	0x4844 A7DC
CFG_SPI2_CS0_IN	RW	32	0x0000 07E0	0x4844 A7E0
CFG_SPI2_CS0_OEN	RW	32	0x0000 07E4	0x4844 A7E4
CFG_SPI2_CS0_OUT	RW	32	0x0000 07E8	0x4844 A7E8
CFG_SPI2_D0_IN	RW	32	0x0000 07EC	0x4844 A7EC
CFG_SPI2_D0_OEN	RW	32	0x0000 07F0	0x4844 A7F0
CFG_SPI2_D0_OUT	RW	32	0x0000 07F4	0x4844 A7F4
CFG_SPI2_D1_IN	RW	32	0x0000 07F8	0x4844 A7F8
CFG_SPI2_D1_OEN	RW	32	0x0000 07FC	0x4844 A7FC
CFG_SPI2_D1_OUT	RW	32	0x0000 0800	0x4844 A800
CFG_SPI2_SCLK_IN	RW	32	0x0000 0804	0x4844 A804
CFG_SPI2_SCLK_OEN	RW	32	0x0000 0808	0x4844 A808
CFG_SPI2_SCLK_OUT	RW	32	0x0000 080C	0x4844 A80C
CFG_TDI_IN	RW	32	0x0000 0810	0x4844 A810
CFG_TDI_OEN	RW	32	0x0000 0814	0x4844 A814
CFG_TDI_OUT	RW	32	0x0000 0818	0x4844 A818
CFG_TDO_IN	RW	32	0x0000 081C	0x4844 A81C
CFG_TDO_OEN	RW	32	0x0000 0820	0x4844 A820
CFG_TDO_OUT	RW	32	0x0000 0824	0x4844 A824
CFG_TMS_IN	RW	32	0x0000 0828	0x4844 A828
CFG_TMS_OEN	RW	32	0x0000 082C	0x4844 A82C
CFG_TMS_OUT	RW	32	0x0000 0830	0x4844 A830
CFG_TRSTN_IN	RW	32	0x0000 0834	0x4844 A834
CFG_TRSTN_OEN	RW	32	0x0000 0838	0x4844 A838
CFG_TRSTN_OUT	RW	32	0x0000 083C	0x4844 A83C

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_UART1_CTSN_IN	RW	32	0x0000 0840	0x4844 A840
CFG_UART1_CTSN_OEN	RW	32	0x0000 0844	0x4844 A844
CFG_UART1_CTSN_OUT	RW	32	0x0000 0848	0x4844 A848
CFG_UART1_RTSN_IN	RW	32	0x0000 084C	0x4844 A84C
CFG_UART1_RTSN_OEN	RW	32	0x0000 0850	0x4844 A850
CFG_UART1_RTSN_OUT	RW	32	0x0000 0854	0x4844 A854
CFG_UART1_RXD_IN	RW	32	0x0000 0858	0x4844 A858
CFG_UART1_RXD_OEN	RW	32	0x0000 085C	0x4844 A85C
CFG_UART1_RXD_OUT	RW	32	0x0000 0860	0x4844 A860
CFG_UART1_TXD_IN	RW	32	0x0000 0864	0x4844 A864
CFG_UART1_TXD_OEN	RW	32	0x0000 0868	0x4844 A868
CFG_UART1_TXD_OUT	RW	32	0x0000 086C	0x4844 A86C
CFG_UART2_CTSN_IN	RW	32	0x0000 0870	0x4844 A870
CFG_UART2_CTSN_OEN	RW	32	0x0000 0874	0x4844 A874
CFG_UART2_CTSN_OUT	RW	32	0x0000 0878	0x4844 A878
CFG_UART2_RTSN_IN	RW	32	0x0000 087C	0x4844 A87C
CFG_UART2_RTSN_OEN	RW	32	0x0000 0880	0x4844 A880
CFG_UART2_RTSN_OUT	RW	32	0x0000 0884	0x4844 A884
CFG_UART2_RXD_IN	RW	32	0x0000 0888	0x4844 A888
CFG_UART2_RXD_OEN	RW	32	0x0000 088C	0x4844 A88C
CFG_UART2_RXD_OUT	RW	32	0x0000 0890	0x4844 A890
CFG_UART2_TXD_IN	RW	32	0x0000 0894	0x4844 A894
CFG_UART2_TXD_OEN	RW	32	0x0000 0898	0x4844 A898
CFG_UART2_TXD_OUT	RW	32	0x0000 089C	0x4844 A89C
CFG_UART3_RXD_IN	RW	32	0x0000 08A0	0x4844 A8A0
CFG_UART3_RXD_OEN	RW	32	0x0000 08A4	0x4844 A8A4
CFG_UART3_RXD_OUT	RW	32	0x0000 08A8	0x4844 A8A8
CFG_UART3_TXD_IN	RW	32	0x0000 08AC	0x4844 A8AC
CFG_UART3_TXD_OEN	RW	32	0x0000 08B0	0x4844 A8B0
CFG_UART3_TXD_OUT	RW	32	0x0000 08B4	0x4844 A8B4
CFG_USB1_DRVVBUS_IN	RW	32	0x0000 08B8	0x4844 A8B8
CFG_USB1_DRVVBUS_OEN	RW	32	0x0000 08BC	0x4844 A8BC
CFG_USB1_DRVVBUS_OUT	RW	32	0x0000 08C0	0x4844 A8C0
CFG_USB2_DRVVBUS_IN	RW	32	0x0000 08C4	0x4844 A8C4
CFG_USB2_DRVVBUS_OEN	RW	32	0x0000 08C8	0x4844 A8C8
CFG_USB2_DRVVBUS_OUT	RW	32	0x0000 08CC	0x4844 A8CC
CFG_VIN1A_CLK0_IN	RW	32	0x0000 08D0	0x4844 A8D0
CFG_VIN1A_CLK0_OEN	RW	32	0x0000 08D4	0x4844 A8D4
CFG_VIN1A_CLK0_OUT	RW	32	0x0000 08D8	0x4844 A8D8
CFG_VIN1A_D0_IN	RW	32	0x0000 08DC	0x4844 A8DC
CFG_VIN1A_D0_OEN	RW	32	0x0000 08E0	0x4844 A8E0
CFG_VIN1A_D0_OUT	RW	32	0x0000 08E4	0x4844 A8E4
CFG_VIN1A_D10_IN	RW	32	0x0000 08E8	0x4844 A8E8
CFG_VIN1A_D10_OEN	RW	32	0x0000 08EC	0x4844 A8EC
CFG_VIN1A_D10_OUT	RW	32	0x0000 08F0	0x4844 A8F0
CFG_VIN1A_D11_IN	RW	32	0x0000 08F4	0x4844 A8F4
CFG_VIN1A_D11_OEN	RW	32	0x0000 08F8	0x4844 A8F8
CFG_VIN1A_D11_OUT	RW	32	0x0000 08FC	0x4844 A8FC
CFG_VIN1A_D12_IN	RW	32	0x0000 0900	0x4844 A900

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_VIN1A_D12_OEN	RW	32	0x0000 0904	0x4844 A904
CFG_VIN1A_D12_OUT	RW	32	0x0000 0908	0x4844 A908
CFG_VIN1A_D13_IN	RW	32	0x0000 090C	0x4844 A90C
CFG_VIN1A_D13_OEN	RW	32	0x0000 0910	0x4844 A910
CFG_VIN1A_D13_OUT	RW	32	0x0000 0914	0x4844 A914
CFG_VIN1A_D14_IN	RW	32	0x0000 0918	0x4844 A918
CFG_VIN1A_D14_OEN	RW	32	0x0000 091C	0x4844 A91C
CFG_VIN1A_D14_OUT	RW	32	0x0000 0920	0x4844 A920
CFG_VIN1A_D15_IN	RW	32	0x0000 0924	0x4844 A924
CFG_VIN1A_D15_OEN	RW	32	0x0000 0928	0x4844 A928
CFG_VIN1A_D15_OUT	RW	32	0x0000 092C	0x4844 A92C
CFG_VIN1A_D16_IN	RW	32	0x0000 0930	0x4844 A930
CFG_VIN1A_D16_OEN	RW	32	0x0000 0934	0x4844 A934
CFG_VIN1A_D16_OUT	RW	32	0x0000 0938	0x4844 A938
CFG_VIN1A_D17_IN	RW	32	0x0000 093C	0x4844 A93C
CFG_VIN1A_D17_OEN	RW	32	0x0000 0940	0x4844 A940
CFG_VIN1A_D17_OUT	RW	32	0x0000 0944	0x4844 A944
CFG_VIN1A_D18_IN	RW	32	0x0000 0948	0x4844 A948
CFG_VIN1A_D18_OEN	RW	32	0x0000 094C	0x4844 A94C
CFG_VIN1A_D18_OUT	RW	32	0x0000 0950	0x4844 A950
CFG_VIN1A_D19_IN	RW	32	0x0000 0954	0x4844 A954
CFG_VIN1A_D19_OEN	RW	32	0x0000 0958	0x4844 A958
CFG_VIN1A_D19_OUT	RW	32	0x0000 095C	0x4844 A95C
CFG_VIN1A_D1_IN	RW	32	0x0000 0960	0x4844 A960
CFG_VIN1A_D1_OEN	RW	32	0x0000 0964	0x4844 A964
CFG_VIN1A_D1_OUT	RW	32	0x0000 0968	0x4844 A968
CFG_VIN1A_D20_IN	RW	32	0x0000 096C	0x4844 A96C
CFG_VIN1A_D20_OEN	RW	32	0x0000 0970	0x4844 A970
CFG_VIN1A_D20_OUT	RW	32	0x0000 0974	0x4844 A974
CFG_VIN1A_D21_IN	RW	32	0x0000 0978	0x4844 A978
CFG_VIN1A_D21_OEN	RW	32	0x0000 097C	0x4844 A97C
CFG_VIN1A_D21_OUT	RW	32	0x0000 0980	0x4844 A980
CFG_VIN1A_D22_IN	RW	32	0x0000 0984	0x4844 A984
CFG_VIN1A_D22_OEN	RW	32	0x0000 0988	0x4844 A988
CFG_VIN1A_D22_OUT	RW	32	0x0000 098C	0x4844 A98C
CFG_VIN1A_D23_IN	RW	32	0x0000 0990	0x4844 A990
CFG_VIN1A_D23_OEN	RW	32	0x0000 0994	0x4844 A994
CFG_VIN1A_D23_OUT	RW	32	0x0000 0998	0x4844 A998
CFG_VIN1A_D2_IN	RW	32	0x0000 099C	0x4844 A99C
CFG_VIN1A_D2_OEN	RW	32	0x0000 09A0	0x4844 A9A0
CFG_VIN1A_D2_OUT	RW	32	0x0000 09A4	0x4844 A9A4
CFG_VIN1A_D3_IN	RW	32	0x0000 09A8	0x4844 A9A8
CFG_VIN1A_D3_OEN	RW	32	0x0000 09AC	0x4844 A9AC
CFG_VIN1A_D3_OUT	RW	32	0x0000 09B0	0x4844 A9B0
CFG_VIN1A_D4_IN	RW	32	0x0000 09B4	0x4844 A9B4
CFG_VIN1A_D4_OEN	RW	32	0x0000 09B8	0x4844 A9B8
CFG_VIN1A_D4_OUT	RW	32	0x0000 09BC	0x4844 A9BC
CFG_VIN1A_D5_IN	RW	32	0x0000 09C0	0x4844 A9C0
CFG_VIN1A_D5_OEN	RW	32	0x0000 09C4	0x4844 A9C4

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_VIN1A_D5_OUT	RW	32	0x0000 09C8	0x4844 A9C8
CFG_VIN1A_D6_IN	RW	32	0x0000 09CC	0x4844 A9CC
CFG_VIN1A_D6_OEN	RW	32	0x0000 09D0	0x4844 A9D0
CFG_VIN1A_D6_OUT	RW	32	0x0000 09D4	0x4844 A9D4
CFG_VIN1A_D7_IN	RW	32	0x0000 09D8	0x4844 A9D8
CFG_VIN1A_D7_OEN	RW	32	0x0000 09DC	0x4844 A9DC
CFG_VIN1A_D7_OUT	RW	32	0x0000 09E0	0x4844 A9E0
CFG_VIN1A_D8_IN	RW	32	0x0000 09E4	0x4844 A9E4
CFG_VIN1A_D8_OEN	RW	32	0x0000 09E8	0x4844 A9E8
CFG_VIN1A_D8_OUT	RW	32	0x0000 09EC	0x4844 A9EC
CFG_VIN1A_D9_IN	RW	32	0x0000 09F0	0x4844 A9F0
CFG_VIN1A_D9_OEN	RW	32	0x0000 09F4	0x4844 A9F4
CFG_VIN1A_D9_OUT	RW	32	0x0000 09F8	0x4844 A9F8
CFG_VIN1A_DE0_IN	RW	32	0x0000 09FC	0x4844 A9FC
CFG_VIN1A_DE0_OEN	RW	32	0x0000 0A00	0x4844 AA00
CFG_VIN1A_DE0_OUT	RW	32	0x0000 0A04	0x4844 AA04
CFG_VIN1A_FLD0_IN	RW	32	0x0000 0A08	0x4844 AA08
CFG_VIN1A_FLD0_OEN	RW	32	0x0000 0A0C	0x4844 AA0C
CFG_VIN1A_FLD0_OUT	RW	32	0x0000 0A10	0x4844 AA10
CFG_VIN1A_HSYNC0_IN	RW	32	0x0000 0A14	0x4844 AA14
CFG_VIN1A_HSYNC0_OEN	RW	32	0x0000 0A18	0x4844 AA18
CFG_VIN1A_HSYNC0_OUT	RW	32	0x0000 0A1C	0x4844 AA1C
CFG_VIN1A_VSYNC0_IN	RW	32	0x0000 0A20	0x4844 AA20
CFG_VIN1A_VSYNC0_OEN	RW	32	0x0000 0A24	0x4844 AA24
CFG_VIN1A_VSYNC0_OUT	RW	32	0x0000 0A28	0x4844 AA28
CFG_VIN1B_CLK1_IN	RW	32	0x0000 0A2C	0x4844 AA2C
CFG_VIN1B_CLK1_OEN	RW	32	0x0000 0A30	0x4844 AA30
CFG_VIN1B_CLK1_OUT	RW	32	0x0000 0A34	0x4844 AA34
CFG_VIN2A_CLK0_IN	RW	32	0x0000 0A38	0x4844 AA38
CFG_VIN2A_CLK0_OEN	RW	32	0x0000 0A3C	0x4844 AA3C
CFG_VIN2A_CLK0_OUT	RW	32	0x0000 0A40	0x4844 AA40
CFG_VIN2A_D0_IN	RW	32	0x0000 0A44	0x4844 AA44
CFG_VIN2A_D0_OEN	RW	32	0x0000 0A48	0x4844 AA48
CFG_VIN2A_D0_OUT	RW	32	0x0000 0A4C	0x4844 AA4C
CFG_VIN2A_D10_IN	RW	32	0x0000 0A50	0x4844 AA50
CFG_VIN2A_D10_OEN	RW	32	0x0000 0A54	0x4844 AA54
CFG_VIN2A_D10_OUT	RW	32	0x0000 0A58	0x4844 AA58
CFG_VIN2A_D11_IN	RW	32	0x0000 0A5C	0x4844 AA5C
CFG_VIN2A_D11_OEN	RW	32	0x0000 0A60	0x4844 AA60
CFG_VIN2A_D11_OUT	RW	32	0x0000 0A64	0x4844 AA64
CFG_VIN2A_D12_IN	RW	32	0x0000 0A68	0x4844 AA68
CFG_VIN2A_D12_OEN	RW	32	0x0000 0A6C	0x4844 AA6C
CFG_VIN2A_D12_OUT	RW	32	0x0000 0A70	0x4844 AA70
CFG_VIN2A_D13_IN	RW	32	0x0000 0A74	0x4844 AA74
CFG_VIN2A_D13_OEN	RW	32	0x0000 0A78	0x4844 AA78
CFG_VIN2A_D13_OUT	RW	32	0x0000 0A7C	0x4844 AA7C
CFG_VIN2A_D14_IN	RW	32	0x0000 0A80	0x4844 AA80
CFG_VIN2A_D14_OEN	RW	32	0x0000 0A84	0x4844 AA84
CFG_VIN2A_D14_OUT	RW	32	0x0000 0A88	0x4844 AA88

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_VIN2A_D15_IN	RW	32	0x0000 0A8C	0x4844 AA8C
CFG_VIN2A_D15_OEN	RW	32	0x0000 0A90	0x4844 AA90
CFG_VIN2A_D15_OUT	RW	32	0x0000 0A94	0x4844 AA94
CFG_VIN2A_D16_IN	RW	32	0x0000 0A98	0x4844 AA98
CFG_VIN2A_D16_OEN	RW	32	0x0000 0A9C	0x4844 AA9C
CFG_VIN2A_D16_OUT	RW	32	0x0000 0AA0	0x4844 AAA0
CFG_VIN2A_D17_IN	RW	32	0x0000 0AA4	0x4844 AAA4
CFG_VIN2A_D17_OEN	RW	32	0x0000 0AA8	0x4844 AAA8
CFG_VIN2A_D17_OUT	RW	32	0x0000 0AAC	0x4844 AAAC
CFG_VIN2A_D18_IN	RW	32	0x0000 0AB0	0x4844 AAB0
CFG_VIN2A_D18_OEN	RW	32	0x0000 0AB4	0x4844 AAB4
CFG_VIN2A_D18_OUT	RW	32	0x0000 0AB8	0x4844 AAB8
CFG_VIN2A_D19_IN	RW	32	0x0000 0ABC	0x4844 AABC
CFG_VIN2A_D19_OEN	RW	32	0x0000 0AC0	0x4844 AAC0
CFG_VIN2A_D19_OUT	RW	32	0x0000 0AC4	0x4844 AAC4
CFG_VIN2A_D1_IN	RW	32	0x0000 0AC8	0x4844 AAC8
CFG_VIN2A_D1_OEN	RW	32	0x0000 0ACC	0x4844 AACC
CFG_VIN2A_D1_OUT	RW	32	0x0000 0AD0	0x4844 AAD0
CFG_VIN2A_D20_IN	RW	32	0x0000 0AD4	0x4844 AAD4
CFG_VIN2A_D20_OEN	RW	32	0x0000 0AD8	0x4844 AAD8
CFG_VIN2A_D20_OUT	RW	32	0x0000 0ADC	0x4844 AADC
CFG_VIN2A_D21_IN	RW	32	0x0000 0AE0	0x4844 AAE0
CFG_VIN2A_D21_OEN	RW	32	0x0000 0AE4	0x4844 AAE4
CFG_VIN2A_D21_OUT	RW	32	0x0000 0AE8	0x4844 AAE8
CFG_VIN2A_D22_IN	RW	32	0x0000 0AEC	0x4844 AAEC
CFG_VIN2A_D22_OEN	RW	32	0x0000 0AF0	0x4844 AAF0
CFG_VIN2A_D22_OUT	RW	32	0x0000 0AF4	0x4844 AAF4
CFG_VIN2A_D23_IN	RW	32	0x0000 0AF8	0x4844 AAF8
CFG_VIN2A_D23_OEN	RW	32	0x0000 0AFC	0x4844 AAFC
CFG_VIN2A_D23_OUT	RW	32	0x0000 0B00	0x4844 AB00
CFG_VIN2A_D2_IN	RW	32	0x0000 0B04	0x4844 AB04
CFG_VIN2A_D2_OEN	RW	32	0x0000 0B08	0x4844 AB08
CFG_VIN2A_D2_OUT	RW	32	0x0000 0B0C	0x4844 AB0C
CFG_VIN2A_D3_IN	RW	32	0x0000 0B10	0x4844 AB10
CFG_VIN2A_D3_OEN	RW	32	0x0000 0B14	0x4844 AB14
CFG_VIN2A_D3_OUT	RW	32	0x0000 0B18	0x4844 AB18
CFG_VIN2A_D4_IN	RW	32	0x0000 0B1C	0x4844 AB1C
CFG_VIN2A_D4_OEN	RW	32	0x0000 0B20	0x4844 AB20
CFG_VIN2A_D4_OUT	RW	32	0x0000 0B24	0x4844 AB24
CFG_VIN2A_D5_IN	RW	32	0x0000 0B28	0x4844 AB28
CFG_VIN2A_D5_OEN	RW	32	0x0000 0B2C	0x4844 AB2C
CFG_VIN2A_D5_OUT	RW	32	0x0000 0B30	0x4844 AB30
CFG_VIN2A_D6_IN	RW	32	0x0000 0B34	0x4844 AB34
CFG_VIN2A_D6_OEN	RW	32	0x0000 0B38	0x4844 AB38
CFG_VIN2A_D6_OUT	RW	32	0x0000 0B3C	0x4844 AB3C
CFG_VIN2A_D7_IN	RW	32	0x0000 0B40	0x4844 AB40
CFG_VIN2A_D7_OEN	RW	32	0x0000 0B44	0x4844 AB44
CFG_VIN2A_D7_OUT	RW	32	0x0000 0B48	0x4844 AB48
CFG_VIN2A_D8_IN	RW	32	0x0000 0B4C	0x4844 AB4C

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_VIN2A_D8_OEN	RW	32	0x0000 0B50	0x4844 AB50
CFG_VIN2A_D8_OUT	RW	32	0x0000 0B54	0x4844 AB54
CFG_VIN2A_D9_IN	RW	32	0x0000 0B58	0x4844 AB58
CFG_VIN2A_D9_OEN	RW	32	0x0000 0B5C	0x4844 AB5C
CFG_VIN2A_D9_OUT	RW	32	0x0000 0B60	0x4844 AB60
CFG_VIN2A_DE0_IN	RW	32	0x0000 0B64	0x4844 AB64
CFG_VIN2A_DE0_OEN	RW	32	0x0000 0B68	0x4844 AB68
CFG_VIN2A_DE0_OUT	RW	32	0x0000 0B6C	0x4844 AB6C
CFG_VIN2A_FLD0_IN	RW	32	0x0000 0B70	0x4844 AB70
CFG_VIN2A_FLD0_OEN	RW	32	0x0000 0B74	0x4844 AB74
CFG_VIN2A_FLD0_OUT	RW	32	0x0000 0B78	0x4844 AB78
CFG_VIN2A_HSYNC0_IN	RW	32	0x0000 0B7C	0x4844 AB7C
CFG_VIN2A_HSYNC0_OEN	RW	32	0x0000 0B80	0x4844 AB80
CFG_VIN2A_HSYNC0_OUT	RW	32	0x0000 0B84	0x4844 AB84
CFG_VIN2A_VSYNC0_IN	RW	32	0x0000 0B88	0x4844 AB88
CFG_VIN2A_VSYNC0_OEN	RW	32	0x0000 0B8C	0x4844 AB8C
CFG_VIN2A_VSYNC0_OUT	RW	32	0x0000 0B90	0x4844 AB90
CFG_VOUT1_CLK_IN	RW	32	0x0000 0B94	0x4844 AB94
CFG_VOUT1_CLK_OEN	RW	32	0x0000 0B98	0x4844 AB98
CFG_VOUT1_CLK_OUT	RW	32	0x0000 0B9C	0x4844 AB9C
CFG_VOUT1_D0_IN	RW	32	0x0000 0BA0	0x4844 ABA0
CFG_VOUT1_D0_OEN	RW	32	0x0000 0BA4	0x4844 ABA4
CFG_VOUT1_D0_OUT	RW	32	0x0000 0BA8	0x4844 ABA8
CFG_VOUT1_D10_IN	RW	32	0x0000 0BAC	0x4844 ABAC
CFG_VOUT1_D10_OEN	RW	32	0x0000 0BB0	0x4844 ABB0
CFG_VOUT1_D10_OUT	RW	32	0x0000 0BB4	0x4844 ABB4
CFG_VOUT1_D11_IN	RW	32	0x0000 0BB8	0x4844 ABB8
CFG_VOUT1_D11_OEN	RW	32	0x0000 0BBC	0x4844 ABBC
CFG_VOUT1_D11_OUT	RW	32	0x0000 0BC0	0x4844 ABC0
CFG_VOUT1_D12_IN	RW	32	0x0000 0BC4	0x4844 ABC4
CFG_VOUT1_D12_OEN	RW	32	0x0000 0BC8	0x4844 ABC8
CFG_VOUT1_D12_OUT	RW	32	0x0000 0BCC	0x4844 ABCC
CFG_VOUT1_D13_IN	RW	32	0x0000 0BD0	0x4844 ABD0
CFG_VOUT1_D13_OEN	RW	32	0x0000 0BD4	0x4844 ABD4
CFG_VOUT1_D13_OUT	RW	32	0x0000 0BD8	0x4844 ABD8
CFG_VOUT1_D14_IN	RW	32	0x0000 0BDC	0x4844 ABDC
CFG_VOUT1_D14_OEN	RW	32	0x0000 0BE0	0x4844 ABE0
CFG_VOUT1_D14_OUT	RW	32	0x0000 0BE4	0x4844 ABE4
CFG_VOUT1_D15_IN	RW	32	0x0000 0BE8	0x4844 ABE8
CFG_VOUT1_D15_OEN	RW	32	0x0000 0BEC	0x4844 ABEC
CFG_VOUT1_D15_OUT	RW	32	0x0000 0BF0	0x4844 ABF0
CFG_VOUT1_D16_IN	RW	32	0x0000 0BF4	0x4844 ABF4
CFG_VOUT1_D16_OEN	RW	32	0x0000 0BF8	0x4844 ABF8
CFG_VOUT1_D16_OUT	RW	32	0x0000 0BFC	0x4844 ABFC
CFG_VOUT1_D17_IN	RW	32	0x0000 0C00	0x4844 AC00
CFG_VOUT1_D17_OEN	RW	32	0x0000 0C04	0x4844 AC04
CFG_VOUT1_D17_OUT	RW	32	0x0000 0C08	0x4844 AC08
CFG_VOUT1_D18_IN	RW	32	0x0000 0C0C	0x4844 AC0C
CFG_VOUT1_D18_OEN	RW	32	0x0000 0C10	0x4844 AC10

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_VOUT1_D18_OUT	RW	32	0x0000 0C14	0x4844 AC14
CFG_VOUT1_D19_IN	RW	32	0x0000 0C18	0x4844 AC18
CFG_VOUT1_D19_OEN	RW	32	0x0000 0C1C	0x4844 AC1C
CFG_VOUT1_D19_OUT	RW	32	0x0000 0C20	0x4844 AC20
CFG_VOUT1_D1_IN	RW	32	0x0000 0C24	0x4844 AC24
CFG_VOUT1_D1_OEN	RW	32	0x0000 0C28	0x4844 AC28
CFG_VOUT1_D1_OUT	RW	32	0x0000 0C2C	0x4844 AC2C
CFG_VOUT1_D20_IN	RW	32	0x0000 0C30	0x4844 AC30
CFG_VOUT1_D20_OEN	RW	32	0x0000 0C34	0x4844 AC34
CFG_VOUT1_D20_OUT	RW	32	0x0000 0C38	0x4844 AC38
CFG_VOUT1_D21_IN	RW	32	0x0000 0C3C	0x4844 AC3C
CFG_VOUT1_D21_OEN	RW	32	0x0000 0C40	0x4844 AC40
CFG_VOUT1_D21_OUT	RW	32	0x0000 0C44	0x4844 AC44
CFG_VOUT1_D22_IN	RW	32	0x0000 0C48	0x4844 AC48
CFG_VOUT1_D22_OEN	RW	32	0x0000 0C4C	0x4844 AC4C
CFG_VOUT1_D22_OUT	RW	32	0x0000 0C50	0x4844 AC50
CFG_VOUT1_D23_IN	RW	32	0x0000 0C54	0x4844 AC54
CFG_VOUT1_D23_OEN	RW	32	0x0000 0C58	0x4844 AC58
CFG_VOUT1_D23_OUT	RW	32	0x0000 0C5C	0x4844 AC5C
CFG_VOUT1_D2_IN	RW	32	0x0000 0C60	0x4844 AC60
CFG_VOUT1_D2_OEN	RW	32	0x0000 0C64	0x4844 AC64
CFG_VOUT1_D2_OUT	RW	32	0x0000 0C68	0x4844 AC68
CFG_VOUT1_D3_IN	RW	32	0x0000 0C6C	0x4844 AC6C
CFG_VOUT1_D3_OEN	RW	32	0x0000 0C70	0x4844 AC70
CFG_VOUT1_D3_OUT	RW	32	0x0000 0C74	0x4844 AC74
CFG_VOUT1_D4_IN	RW	32	0x0000 0C78	0x4844 AC78
CFG_VOUT1_D4_OEN	RW	32	0x0000 0C7C	0x4844 AC7C
CFG_VOUT1_D4_OUT	RW	32	0x0000 0C80	0x4844 AC80
CFG_VOUT1_D5_IN	RW	32	0x0000 0C84	0x4844 AC84
CFG_VOUT1_D5_OEN	RW	32	0x0000 0C88	0x4844 AC88
CFG_VOUT1_D5_OUT	RW	32	0x0000 0C8C	0x4844 AC8C
CFG_VOUT1_D6_IN	RW	32	0x0000 0C90	0x4844 AC90
CFG_VOUT1_D6_OEN	RW	32	0x0000 0C94	0x4844 AC94
CFG_VOUT1_D6_OUT	RW	32	0x0000 0C98	0x4844 AC98
CFG_VOUT1_D7_IN	RW	32	0x0000 0C9C	0x4844 AC9C
CFG_VOUT1_D7_OEN	RW	32	0x0000 0CA0	0x4844 ACA0
CFG_VOUT1_D7_OUT	RW	32	0x0000 0CA4	0x4844 ACA4
CFG_VOUT1_D8_IN	RW	32	0x0000 0CA8	0x4844 ACA8
CFG_VOUT1_D8_OEN	RW	32	0x0000 0CAC	0x4844 ACAC
CFG_VOUT1_D8_OUT	RW	32	0x0000 0CB0	0x4844 ACB0
CFG_VOUT1_D9_IN	RW	32	0x0000 0CB4	0x4844 ACB4
CFG_VOUT1_D9_OEN	RW	32	0x0000 0CB8	0x4844 ACB8
CFG_VOUT1_D9_OUT	RW	32	0x0000 0CBC	0x4844 ACBC
CFG_VOUT1_DE_IN	RW	32	0x0000 0CC0	0x4844 ACC0
CFG_VOUT1_DE_OEN	RW	32	0x0000 0CC4	0x4844 ACC4
CFG_VOUT1_DE_OUT	RW	32	0x0000 0CC8	0x4844 ACC8
CFG_VOUT1_FLD_IN	RW	32	0x0000 0CCC	0x4844 ACCC
CFG_VOUT1_FLD_OEN	RW	32	0x0000 0CD0	0x4844 ACD0
CFG_VOUT1_FLD_OUT	RW	32	0x0000 0CD4	0x4844 ACD4

Table 18-912. IODELAYCONFIG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IODELAYCONFIG Base Address
CFG_VOUT1_HSYNC_IN	RW	32	0x0000 0CD8	0x4844 ACD8
CFG_VOUT1_HSYNC_OEN	RW	32	0x0000 0CDC	0x4844 ACDC
CFG_VOUT1_HSYNC_OUT	RW	32	0x0000 0CE0	0x4844 ACE0
CFG_VOUT1_VSYNC_IN	RW	32	0x0000 0CE4	0x4844 ACE4
CFG_VOUT1_VSYNC_OEN	RW	32	0x0000 0CE8	0x4844 ACE8
CFG_VOUT1_VSYNC_OUT	RW	32	0x0000 0CEC	0x4844 ACEC
CFG_XREF_CLK0_IN	RW	32	0x0000 0CF0	0x4844 ACF0
CFG_XREF_CLK0_OEN	RW	32	0x0000 0CF4	0x4844 ACF4
CFG_XREF_CLK0_OUT	RW	32	0x0000 0CF8	0x4844 ACF8
CFG_XREF_CLK1_IN	RW	32	0x0000 0CFC	0x4844 ACFC
CFG_XREF_CLK1_OEN	RW	32	0x0000 0D00	0x4844 AD00
CFG_XREF_CLK1_OUT	RW	32	0x0000 0D04	0x4844 AD04
CFG_XREF_CLK2_IN	RW	32	0x0000 0D08	0x4844 AD08
CFG_XREF_CLK2_OEN	RW	32	0x0000 0D0C	0x4844 AD0C
CFG_XREF_CLK2_OUT	RW	32	0x0000 0D10	0x4844 AD10
CFG_XREF_CLK3_IN	RW	32	0x0000 0D14	0x4844 AD14
CFG_XREF_CLK3_OEN	RW	32	0x0000 0D18	0x4844 AD18
CFG_XREF_CLK3_OUT	RW	32	0x0000 0D1C	0x4844 AD1C

18.7.4 IODELAYCONFIG Register Description

Table 18-913. CONFIG_REG_0

Address Offset	0x0000 000C		
Physical Address	0x4844 A00C	**Instance**	IODELAYCONFIG
Description	Calibration Control Register		
Type	RW		

31–2	1	0
RESERVED	ROM_READ	CALIBRATION_START

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	ROM_READ	Triggers complete ROM read when '1' is written. Cleared when ROM read is complete.	RW	0x0
0	CALIBRATION_START	Triggers hardware calibration when '1' is written. Cleared when hardware completes calibration.	RW	0x0

Table 18-914. Register Call Summary for Register CONFIG_REG_0

Control Module Functional Description

- IO Delay Recalibration: [0] [1] [2] [3]

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [4]

Table 18-915. CONFIG_REG_2

Address Offset	0x0000 0014		
Physical Address	0x4844 A014	**Instance**	IODELAYCONFIG
Description	Reference Clock Period Register.		
Type	RW		

31:16	15:0
RESERVED	REFCLK_PERIOD

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	REFCLK_PERIOD	15:0 stores the binary equivalent of reference clock period in units of 10ps. This value (along with calibration results) is used for computing the coarse/fine element delay Example: 0xF0 means 2400ps.	RW	0x21D2

Table 18-916. Register Call Summary for Register CONFIG_REG_2

Control Module Functional Description

- IO Delay Recalibration: [0]

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [1]

Table 18-917. CONFIG_REG_3

Address Offset	0x0000 0018		
Physical Address	0x4844 A018	**Instance**	IODELAYCONFIG
Description	coarse calibration results register		
Type	RW		

31:16	15:0
COARSE_DELAY_COUNT	COARSE_REF_COUNT

Bits	Field Name	Description	Type	Reset
31:16	COARSE_DELAY_COUNT	Results of 16 bit counter clocked by "delay line oscillator" clock during calibration.	RW	0x0
15:0	COARSE_REF_COUNT	Results of 16 bit counter clocked by "reference" clock during coarse calibration.	RW	0x0

Table 18-918. Register Call Summary for Register CONFIG_REG_3

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-919. CONFIG_REG_4

Address Offset	0x0000 001C		
Physical Address	0x4844 A01C	**Instance**	IODELAYCONFIG
Description	fine calibration results register		
Type	RW		

31:16	15:0
FINE_DELAY_COUNT	FINE_REF_COUNT

Bits	Field Name	Description	Type	Reset
31:16	FINE_DELAY_COUNT	Results of 16 bit counter clocked by "delay line oscillator" clock during fine calibration.	RW	0x0
15:0	FINE_REF_COUNT	Results of 16 bit counter clocked by "reference" clock during fine calibration.	RW	0x0

Table 18-920. Register Call Summary for Register CONFIG_REG_4

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-921. CONFIG_REG_8

Address Offset	0x0000 002C		
Physical Address	0x4844 A02C	**Instance**	IODELAYCONFIG
Description	Global Lock Register.		
Type	RW		

31:1	0
RESERVED	GLOBAL_LOCK_BIT

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	GLOBAL_LOCK_BIT	Global Lock Bit Register. A '1' in this bit protects the writes to MMRs that store delay line select values. A '0' in this bit indicates that MMRs that store delay line select values are writeable. To write a '0' to this bit, signature of 0x5555 must be used on the MSB bits 16:1 of mdata.	RW	0x1

Table 18-922. Register Call Summary for Register CONFIG_REG_8

Control Module Functional Description

- IO Delay Recalibration: [0] [1]

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [2]

Table 18-923. CFG_RMII_MHZ_50_CLK_IN

Address Offset	0x0000 0030		
Physical Address	0x4844 A030	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_RMII_MHZ_50_CLK_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-924. Register Call Summary for Register CFG_RMII_MHZ_50_CLK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-925. CFG_RMII_MHZ_50_CLK_OEN

Address Offset	0x0000 0034		
Physical Address	0x4844 A034	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_RMII_MHZ_50_CLK_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-926. Register Call Summary for Register CFG_RMII_MHZ_50_CLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-927. CFG_RMII_MHZ_50_CLK_OUT

Address Offset	0x0000 0038		
Physical Address	0x4844 A038	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_RMII_MHZ_50_CLK_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-928. Register Call Summary for Register CFG_RMII_MHZ_50_CLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-929. CFG_WAKEUP0_IN

Address Offset	0x0000 003C		
Physical Address	0x4844 A03C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-930. Register Call Summary for Register CFG_WAKEUP0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-931. CFG_WAKEUP0_OEN

Address Offset	0x0000 0040		
Physical Address	0x4844 A040	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-932. Register Call Summary for Register CFG_WAKEUP0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-933. CFG_WAKEUP0_OUT

Address Offset	0x0000 0044		
Physical Address	0x4844 A044	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup0_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-934. Register Call Summary for Register CFG_WAKEUP0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-935. CFG_WAKEUP1_IN

Address Offset	0x0000 0048		
Physical Address	0x4844 A048	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup1_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-936. Register Call Summary for Register CFG_WAKEUP1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-937. CFG_WAKEUP1_OEN

Address Offset	0x0000 004C		
Physical Address	0x4844 A04C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup1_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-938. Register Call Summary for Register CFG_WAKEUP1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-939. CFG_WAKEUP1_OUT

Address Offset	0x0000 0050		
Physical Address	0x4844 A050	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup1_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-940. Register Call Summary for Register CFG_WAKEUP1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-941. CFG_WAKEUP2_IN

Address Offset	0x0000 0054		
Physical Address	0x4844 A054	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-942. Register Call Summary for Register CFG_WAKEUP2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-943. CFG_WAKEUP2_OEN

Address Offset	0x0000 0058		
Physical Address	0x4844 A058	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-944. Register Call Summary for Register CFG_WAKEUP2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-945. CFG_WAKEUP2_OUT

Address Offset	0x0000 005C		
Physical Address	0x4844 A05C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-946. Register Call Summary for Register CFG_WAKEUP2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-947. CFG_WAKEUP3_IN

Address Offset	0x0000 0060		
Physical Address	0x4844 A060	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-948. Register Call Summary for Register CFG_WAKEUP3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-949. CFG_WAKEUP3_OEN

Address Offset	0x0000 0064		
Physical Address	0x4844 A064	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup3_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-950. Register Call Summary for Register CFG_WAKEUP3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-951. CFG_WAKEUP3_OUT

Address Offset	0x0000 0068		
Physical Address	0x4844 A068	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_Wakeup3_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-952. Register Call Summary for Register CFG_WAKEUP3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-953. CFG_DCAN1_RX_IN

Address Offset	0x0000 006C		
Physical Address	0x4844 A06C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan1_rx_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-954. Register Call Summary for Register CFG_DCAN1_RX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-955. CFG_DCAN1_RX_OEN

Address Offset	0x0000 0070		
Physical Address	0x4844 A070	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan1_rx_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-956. Register Call Summary for Register CFG_DCAN1_RX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-957. CFG_DCAN1_RX_OUT

Address Offset	0x0000 0074		
Physical Address	0x4844 A074	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan1_rx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-958. Register Call Summary for Register CFG_DCAN1_RX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-959. CFG_DCAN1_TX_IN

Address Offset	0x0000 0078		
Physical Address	0x4844 A078	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan1_tx_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-960. Register Call Summary for Register CFG_DCAN1_TX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-961. CFG_DCAN1_TX_OEN

Address Offset	0x0000 007C		
Physical Address	0x4844 A07C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan1_tx_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-962. Register Call Summary for Register CFG_DCAN1_TX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-963. CFG_DCAN1_TX_OUT

Address Offset	0x0000 0080		
Physical Address	0x4844 A080	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan1_tx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-964. Register Call Summary for Register CFG_DCAN1_TX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-965. CFG_DCAN2_RX_IN

Address Offset	0x0000 0084		
Physical Address	0x4844 A084	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan2_rx_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-966. Register Call Summary for Register CFG_DCAN2_RX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-967. CFG_DCAN2_RX_OEN

Address Offset	0x0000 0088		
Physical Address	0x4844 A088	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan2_rx_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-968. Register Call Summary for Register CFG_DCAN2_RX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-969. CFG_DCAN2_RX_OUT

Address Offset	0x0000 008C		
Physical Address	0x4844 A08C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan2_rx_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-970. Register Call Summary for Register CFG_DCAN2_RX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-971. CFG_DCAN2_TX_IN

Address Offset	0x0000 0090		
Physical Address	0x4844 A090	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan2_tx_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-972. Register Call Summary for Register CFG_DCAN2_TX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-973. CFG_DCAN2_TX_OEN

Address Offset	0x0000 0094		
Physical Address	0x4844 A094	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan2_tx_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-974. Register Call Summary for Register CFG_DCAN2_TX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-975. CFG_DCAN2_TX_OUT

Address Offset	0x0000 0098		
Physical Address	0x4844 A098	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_dcan2_tx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-976. Register Call Summary for Register CFG_DCAN2_TX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-977. CFG_EMU0_IN

Address Offset	0x0000 009C		
Physical Address	0x4844 A09C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-978. Register Call Summary for Register CFG_EMU0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-979. CFG_EMU0_OEN

Address Offset	0x0000 00A0		
Physical Address	0x4844 A0A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-980. Register Call Summary for Register CFG_EMU0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-981. CFG_EMU0_OUT

Address Offset	0x0000 00A4		
Physical Address	0x4844 A0A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu0_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-982. Register Call Summary for Register CFG_EMU0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-983. CFG_EMU1_IN

Address Offset	0x0000 00A8		
Physical Address	0x4844 A0A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu1_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-984. Register Call Summary for Register CFG_EMU1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-985. CFG_EMU1_OEN

Address Offset	0x0000 00AC		
Physical Address	0x4844 A0AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu1_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-986. Register Call Summary for Register CFG_EMU1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-987. CFG_EMU1_OUT

Address Offset	0x0000 00B0		
Physical Address	0x4844 A0B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu1_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-988. Register Call Summary for Register CFG_EMU1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-989. CFG_EMU2_IN

Address Offset	0x0000 00B4		
Physical Address	0x4844 A0B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-990. Register Call Summary for Register CFG_EMU2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-991. CFG_EMU2_OEN

Address Offset	0x0000 00B8		
Physical Address	0x4844 A0B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-992. Register Call Summary for Register CFG_EMU2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-993. CFG_EMU2_OUT

Address Offset	0x0000 00BC		
Physical Address	0x4844 A0BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-994. Register Call Summary for Register CFG_EMU2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-995. CFG_EMU3_IN

Address Offset	0x0000 00C0		
Physical Address	0x4844 A0C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-996. Register Call Summary for Register CFG_EMU3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-997. CFG_EMU3_OEN

Address Offset	0x0000 00C4		
Physical Address	0x4844 A0C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-998. Register Call Summary for Register CFG_EMU3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-999. CFG_EMU3_OUT

Address Offset	0x0000 00C8		
Physical Address	0x4844 A0C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1000. Register Call Summary for Register CFG_EMU3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1001. CFG_EMU4_IN

Address Offset	0x0000 00CC		
Physical Address	0x4844 A0CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu4_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1002. Register Call Summary for Register CFG_EMU4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1003. CFG_EMU4_OEN

Address Offset	0x0000 00D0		
Physical Address	0x4844 A0D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu4_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1004. Register Call Summary for Register CFG_EMU4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1005. CFG_EMU4_OUT

Address Offset	0x0000 00D4		
Physical Address	0x4844 A0D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_emu4_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1006. Register Call Summary for Register CFG_EMU4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1007. CFG_GPIO6_10_IN

Address Offset	0x0000 00D8		
Physical Address	0x4844 A0D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_10_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1008. Register Call Summary for Register CFG_GPIO6_10_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1009. CFG_GPIO6_10_OEN

Address Offset	0x0000 00DC		
Physical Address	0x4844 A0DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_10_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1010. Register Call Summary for Register CFG_GPIO6_10_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1011. CFG_GPIO6_10_OUT

Address Offset	0x0000 00E0		
Physical Address	0x4844 A0E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_10_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1012. Register Call Summary for Register CFG_GPIO6_10_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1013. CFG_GPIO6_11_IN

Address Offset	0x0000 00E4		
Physical Address	0x4844 A0E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_11_in interface		
Type	RW		

31-18	17-12	11	10	9-0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1014. Register Call Summary for Register CFG_GPIO6_11_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1015. CFG_GPIO6_11_OEN

Address Offset	0x0000 00E8		
Physical Address	0x4844 A0E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_11_oen interface		
Type	RW		

31-18	17-12	11	10	9-0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1016. Register Call Summary for Register CFG_GPIO6_11_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1017. CFG_GPIO6_11_OUT

Address Offset	0x0000 00EC		
Physical Address	0x4844 A0EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_11_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1018. Register Call Summary for Register CFG_GPIO6_11_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1019. CFG_GPIO6_14_IN

Address Offset	0x0000 00F0		
Physical Address	0x4844 A0F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_14_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1020. Register Call Summary for Register CFG_GPIO6_14_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1021. CFG_GPIO6_14_OEN

Address Offset	0x0000 00F4		
Physical Address	0x4844 A0F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_14_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1022. Register Call Summary for Register CFG_GPIO6_14_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1023. CFG_GPIO6_14_OUT

Address Offset	0x0000 00F8		
Physical Address	0x4844 A0F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_14_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1024. Register Call Summary for Register CFG_GPIO6_14_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1025. CFG_GPIO6_15_IN

Address Offset	0x0000 00FC		
Physical Address	0x4844 A0FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_15_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1026. Register Call Summary for Register CFG_GPIO6_15_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1027. CFG_GPIO6_15_OEN

Address Offset	0x0000 0100		
Physical Address	0x4844 A100	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_15_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1028. Register Call Summary for Register CFG_GPIO6_15_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1029. CFG_GPIO6_15_OUT

Address Offset	0x0000 0104		
Physical Address	0x4844 A104	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_15_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1030. Register Call Summary for Register CFG_GPIO6_15_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1031. CFG_GPIO6_16_IN

Address Offset	0x0000 0108		
Physical Address	0x4844 A108	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_16_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1032. Register Call Summary for Register CFG_GPIO6_16_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1033. CFG_GPIO6_16_OEN

Address Offset	0x0000 010C		
Physical Address	0x4844 A10C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_16_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1034. Register Call Summary for Register CFG_GPIO6_16_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1035. CFG_GPIO6_16_OUT

Address Offset	0x0000 0110		
Physical Address	0x4844 A110	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpio6_16_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1036. Register Call Summary for Register CFG_GPIO6_16_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1037. CFG_GPMC_A0_IN

Address Offset	0x0000 0114		
Physical Address	0x4844 A114	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1038. Register Call Summary for Register CFG_GPMC_A0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1039. CFG_GPMC_A0_OEN

Address Offset	0x0000 0118		
Physical Address	0x4844 A118	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1040. Register Call Summary for Register CFG_GPMC_A0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1041. CFG_GPMC_A0_OUT

Address Offset	0x0000 011C		
Physical Address	0x4844 A11C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1042. Register Call Summary for Register CFG_GPMC_A0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1043. CFG_GPMC_A10_IN

Address Offset	0x0000 0120		
Physical Address	0x4844 A120	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a10_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1044. Register Call Summary for Register CFG_GPMC_A10_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1045. CFG_GPMC_A10_OEN

Address Offset	0x0000 0124		
Physical Address	0x4844 A124	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a10_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1046. Register Call Summary for Register CFG_GPMC_A10_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1047. CFG_GPMC_A10_OUT

Address Offset	0x0000 0128		
Physical Address	0x4844 A128	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a10_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1048. Register Call Summary for Register CFG_GPMC_A10_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1049. CFG_GPMC_A11_IN

Address Offset	0x0000 012C		
Physical Address	0x4844 A12C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a11_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1050. Register Call Summary for Register CFG_GPMC_A11_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1051. CFG_GPMC_A11_OEN

Address Offset	0x0000 0130		
Physical Address	0x4844 A130	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a11_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1052. Register Call Summary for Register CFG_GPMC_A11_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1053. CFG_GPMC_A11_OUT

Address Offset	0x0000 0134		
Physical Address	0x4844 A134	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a11_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1054. Register Call Summary for Register CFG_GPMC_A11_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1055. CFG_GPMC_A12_IN

Address Offset	0x0000 0138		
Physical Address	0x4844 A138	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a12_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1056. Register Call Summary for Register CFG_GPMC_A12_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1057. CFG_GPMC_A12_OEN

Address Offset	0x0000 013C		
Physical Address	0x4844 A13C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a12_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1058. Register Call Summary for Register CFG_GPMC_A12_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1059. CFG_GPMC_A12_OUT

Address Offset	0x0000 0140		
Physical Address	0x4844 A140	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a12_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1060. Register Call Summary for Register CFG_GPMC_A12_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1061. CFG_GPMC_A13_IN

Address Offset	0x0000 0144		
Physical Address	0x4844 A144	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a13_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1062. Register Call Summary for Register CFG_GPMC_A13_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1063. CFG_GPMC_A13_OEN

Address Offset	0x0000 0148		
Physical Address	0x4844 A148	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a13_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1064. Register Call Summary for Register CFG_GPMC_A13_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1065. CFG_GPMC_A13_OUT

Address Offset	0x0000 014C		
Physical Address	0x4844 A14C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a13_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1066. Register Call Summary for Register CFG_GPMC_A13_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1067. CFG_GPMC_A14_IN

Address Offset	0x0000 0150		
Physical Address	0x4844 A150	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a14_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1068. Register Call Summary for Register CFG_GPMC_A14_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1069. CFG_GPMC_A14_OEN

Address Offset	0x0000 0154		
Physical Address	0x4844 A154	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a14_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1070. Register Call Summary for Register CFG_GPMC_A14_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1071. CFG_GPMC_A14_OUT

Address Offset	0x0000 0158		
Physical Address	0x4844 A158	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a14_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1072. Register Call Summary for Register CFG_GPMC_A14_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1073. CFG_GPMC_A15_IN

Address Offset	0x0000 015C		
Physical Address	0x4844 A15C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a15_in interface		
Type	RW		

31-18	17-12	11	10	9-0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1074. Register Call Summary for Register CFG_GPMC_A15_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1075. CFG_GPMC_A15_OEN

Address Offset	0x0000 0160		
Physical Address	0x4844 A160	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a15_oen interface		
Type	RW		

31-18	17-12	11	10	9-0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1076. Register Call Summary for Register CFG_GPMC_A15_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1077. CFG_GPMC_A15_OUT

Address Offset	0x0000 0164		
Physical Address	0x4844 A164	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a15_out interface		
Type	RW		

Bits 31–18	Bits 17–12	Bit 11	Bit 10	Bits 9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1078. Register Call Summary for Register CFG_GPMC_A15_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1079. CFG_GPMC_A16_IN

Address Offset	0x0000 0168		
Physical Address	0x4844 A168	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a16_in interface		
Type	RW		

Bits 31–18	Bits 17–12	Bit 11	Bit 10	Bits 9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1080. Register Call Summary for Register CFG_GPMC_A16_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1081. CFG_GPMC_A16_OEN

Address Offset	0x0000 016C		
Physical Address	0x4844 A16C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a16_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1082. Register Call Summary for Register CFG_GPMC_A16_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1083. CFG_GPMC_A16_OUT

Address Offset	0x0000 0170		
Physical Address	0x4844 A170	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a16_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1084. Register Call Summary for Register CFG_GPMC_A16_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1085. CFG_GPMC_A17_IN

Address Offset	0x0000 0174		
Physical Address	0x4844 A174	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a17_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1086. Register Call Summary for Register CFG_GPMC_A17_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1087. CFG_GPMC_A17_OEN

Address Offset	0x0000 0178		
Physical Address	0x4844 A178	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a17_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1088. Register Call Summary for Register CFG_GPMC_A17_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1089. CFG_GPMC_A17_OUT

Address Offset	0x0000 017C		
Physical Address	0x4844 A17C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a17_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1090. Register Call Summary for Register CFG_GPMC_A17_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1091. CFG_GPMC_A18_IN

Address Offset	0x0000 0180		
Physical Address	0x4844 A180	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a18_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1092. Register Call Summary for Register CFG_GPMC_A18_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1093. CFG_GPMC_A18_OEN

Address Offset	0x0000 0184		
Physical Address	0x4844 A184	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a18_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1094. Register Call Summary for Register CFG_GPMC_A18_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1095. CFG_GPMC_A18_OUT

Address Offset	0x0000 0188		
Physical Address	0x4844 A188	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a18_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1096. Register Call Summary for Register CFG_GPMC_A18_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1097. CFG_GPMC_A19_IN

Address Offset	0x0000 018C		
Physical Address	0x4844 A18C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a19_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1098. Register Call Summary for Register CFG_GPMC_A19_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1099. CFG_GPMC_A19_OEN

Address Offset	0x0000 0190		
Physical Address	0x4844 A190	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a19_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1100. Register Call Summary for Register CFG_GPMC_A19_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1101. CFG_GPMC_A19_OUT

Address Offset	0x0000 0194		
Physical Address	0x4844 A194	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a19_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1102. Register Call Summary for Register CFG_GPMC_A19_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1103. CFG_GPMC_A1_IN

Address Offset	0x0000 0198		
Physical Address	0x4844 A198	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1104. Register Call Summary for Register CFG_GPMC_A1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1105. CFG_GPMC_A1_OEN

Address Offset	0x0000 019C		
Physical Address	0x4844 A19C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1106. Register Call Summary for Register CFG_GPMC_A1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1107. CFG_GPMC_A1_OUT

Address Offset	0x0000 01A0		
Physical Address	0x4844 A1A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1108. Register Call Summary for Register CFG_GPMC_A1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1109. CFG_GPMC_A20_IN

Address Offset	0x0000 01A4		
Physical Address	0x4844 A1A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a20_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1110. Register Call Summary for Register CFG_GPMC_A20_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1111. CFG_GPMC_A20_OEN

Address Offset	0x0000 01A8		
Physical Address	0x4844 A1A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a20_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1112. Register Call Summary for Register CFG_GPMC_A20_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1113. CFG_GPMC_A20_OUT

Address Offset	0x0000 01AC		
Physical Address	0x4844 A1AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a20_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1114. Register Call Summary for Register CFG_GPMC_A20_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1115. CFG_GPMC_A21_IN

Address Offset	0x0000 01B0		
Physical Address	0x4844 A1B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a21_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1116. Register Call Summary for Register CFG_GPMC_A21_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1117. CFG_GPMC_A21_OEN

Address Offset	0x0000 01B4		
Physical Address	0x4844 A1B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a21_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1118. Register Call Summary for Register CFG_GPMC_A21_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1119. CFG_GPMC_A21_OUT

Address Offset	0x0000 01B8		
Physical Address	0x4844 A1B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a21_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1120. Register Call Summary for Register CFG_GPMC_A21_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1121. CFG_GPMC_A22_IN

Address Offset	0x0000 01BC		
Physical Address	0x4844 A1BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a22_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1122. Register Call Summary for Register CFG_GPMC_A22_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1123. CFG_GPMC_A22_OEN

Address Offset	0x0000 01C0		
Physical Address	0x4844 A1C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a22_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1124. Register Call Summary for Register CFG_GPMC_A22_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1125. CFG_GPMC_A22_OUT

Address Offset	0x0000 01C4		
Physical Address	0x4844 A1C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a22_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1126. Register Call Summary for Register CFG_GPMC_A22_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1127. CFG_GPMC_A23_IN

Address Offset	0x0000 01C8		
Physical Address	0x4844 A1C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a23_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1128. Register Call Summary for Register CFG_GPMC_A23_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1129. CFG_GPMC_A23_OEN

Address Offset	0x0000 01CC		
Physical Address	0x4844 A1CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a23_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1130. Register Call Summary for Register CFG_GPMC_A23_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1131. CFG_GPMC_A23_OUT

Address Offset	0x0000 01D0		
Physical Address	0x4844 A1D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a23_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1132. Register Call Summary for Register CFG_GPMC_A23_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1133. CFG_GPMC_A24_IN

Address Offset	0x0000 01D4		
Physical Address	0x4844 A1D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a24_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1134. Register Call Summary for Register CFG_GPMC_A24_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1135. CFG_GPMC_A24_OEN

Address Offset	0x0000 01D8		
Physical Address	0x4844 A1D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a24_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1136. Register Call Summary for Register CFG_GPMC_A24_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1137. CFG_GPMC_A24_OUT

Address Offset	0x0000 01DC		
Physical Address	0x4844 A1DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a24_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1138. Register Call Summary for Register CFG_GPMC_A24_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1139. CFG_GPMC_A25_IN

Address Offset	0x0000 01E0		
Physical Address	0x4844 A1E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a25_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1140. Register Call Summary for Register CFG_GPMC_A25_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1141. CFG_GPMC_A25_OEN

Address Offset	0x0000 01E4		
Physical Address	0x4844 A1E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a25_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1142. Register Call Summary for Register CFG_GPMC_A25_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1143. CFG_GPMC_A25_OUT

Address Offset	0x0000 01E8		
Physical Address	0x4844 A1E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a25_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1144. Register Call Summary for Register CFG_GPMC_A25_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1145. CFG_GPMC_A26_IN

Address Offset	0x0000 01EC		
Physical Address	0x4844 A1EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a26_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1146. Register Call Summary for Register CFG_GPMC_A26_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1147. CFG_GPMC_A26_OEN

Address Offset	0x0000 01F0		
Physical Address	0x4844 A1F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a26_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1148. Register Call Summary for Register CFG_GPMC_A26_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1149. CFG_GPMC_A26_OUT

Address Offset	0x0000 01F4		
Physical Address	0x4844 A1F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a26_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1150. Register Call Summary for Register CFG_GPMC_A26_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1151. CFG_GPMC_A27_IN

Address Offset	0x0000 01F8		
Physical Address	0x4844 A1F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a27_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1152. Register Call Summary for Register CFG_GPMC_A27_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1153. CFG_GPMC_A27_OEN

Address Offset	0x0000 01FC		
Physical Address	0x4844 A1FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a27_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1154. Register Call Summary for Register CFG_GPMC_A27_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1155. CFG_GPMC_A27_OUT

Address Offset	0x0000 0200		
Physical Address	0x4844 A200	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a27_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1156. Register Call Summary for Register CFG_GPMC_A27_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1157. CFG_GPMC_A2_IN

Address Offset	0x0000 0204		
Physical Address	0x4844 A204	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1158. Register Call Summary for Register CFG_GPMC_A2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1159. CFG_GPMC_A2_OEN

Address Offset	0x0000 0208		
Physical Address	0x4844 A208	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1160. Register Call Summary for Register CFG_GPMC_A2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1161. CFG_GPMC_A2_OUT

Address Offset	0x0000 020C		
Physical Address	0x4844 A20C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a2_out interface		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18	17 16 15 14 13 12	11	10	9 8 7 6 5 4 3 2 1 0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1162. Register Call Summary for Register CFG_GPMC_A2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1163. CFG_GPMC_A3_IN

Address Offset	0x0000 0210		
Physical Address	0x4844 A210	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a3_in interface		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18	17 16 15 14 13 12	11	10	9 8 7 6 5 4 3 2 1 0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1164. Register Call Summary for Register CFG_GPMC_A3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1165. CFG_GPMC_A3_OEN

Address Offset	0x0000 0214		
Physical Address	0x4844 A214	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1166. Register Call Summary for Register CFG_GPMC_A3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1167. CFG_GPMC_A3_OUT

Address Offset	0x0000 0218		
Physical Address	0x4844 A218	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1168. Register Call Summary for Register CFG_GPMC_A3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1169. CFG_GPMC_A4_IN

Address Offset	0x0000 021C		
Physical Address	0x4844 A21C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a4_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1170. Register Call Summary for Register CFG_GPMC_A4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1171. CFG_GPMC_A4_OEN

Address Offset	0x0000 0220		
Physical Address	0x4844 A220	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a4_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1172. Register Call Summary for Register CFG_GPMC_A4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1173. CFG_GPMC_A4_OUT

Address Offset	0x0000 0224		
Physical Address	0x4844 A224	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a4_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1174. Register Call Summary for Register CFG_GPMC_A4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1175. CFG_GPMC_A5_IN

Address Offset	0x0000 0228		
Physical Address	0x4844 A228	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a5_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1176. Register Call Summary for Register CFG_GPMC_A5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1177. CFG_GPMC_A5_OEN

Address Offset	0x0000 022C		
Physical Address	0x4844 A22C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a5_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1178. Register Call Summary for Register CFG_GPMC_A5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1179. CFG_GPMC_A5_OUT

Address Offset	0x0000 0230		
Physical Address	0x4844 A230	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a5_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1180. Register Call Summary for Register CFG_GPMC_A5_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1181. CFG_GPMC_A6_IN

Address Offset	0x0000 0234		
Physical Address	0x4844 A234	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a6_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1182. Register Call Summary for Register CFG_GPMC_A6_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1183. CFG_GPMC_A6_OEN

Address Offset	0x0000 0238		
Physical Address	0x4844 A238	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a6_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1184. Register Call Summary for Register CFG_GPMC_A6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1185. CFG_GPMC_A6_OUT

Address Offset	0x0000 023C		
Physical Address	0x4844 A23C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a6_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1186. Register Call Summary for Register CFG_GPMC_A6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1187. CFG_GPMC_A7_IN

Address Offset	0x0000 0240		
Physical Address	0x4844 A240	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a7_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1188. Register Call Summary for Register CFG_GPMC_A7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1189. CFG_GPMC_A7_OEN

Address Offset	0x0000 0244		
Physical Address	0x4844 A244	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a7_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1190. Register Call Summary for Register CFG_GPMC_A7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1191. CFG_GPMC_A7_OUT

Address Offset	0x0000 0248		
Physical Address	0x4844 A248	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a7_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1192. Register Call Summary for Register CFG_GPMC_A7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1193. CFG_GPMC_A8_IN

Address Offset	0x0000 024C		
Physical Address	0x4844 A24C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a8_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1194. Register Call Summary for Register CFG_GPMC_A8_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1195. CFG_GPMC_A8_OEN

Address Offset	0x0000 0250		
Physical Address	0x4844 A250	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a8_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1196. Register Call Summary for Register CFG_GPMC_A8_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1197. CFG_GPMC_A8_OUT

Address Offset	0x0000 0254		
Physical Address	0x4844 A254	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a8_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1198. Register Call Summary for Register CFG_GPMC_A8_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1199. CFG_GPMC_A9_IN

Address Offset	0x0000 0258		
Physical Address	0x4844 A258	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a9_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1200. Register Call Summary for Register CFG_GPMC_A9_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1201. CFG_GPMC_A9_OEN

Address Offset	0x0000 025C		
Physical Address	0x4844 A25C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a9_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1202. Register Call Summary for Register CFG_GPMC_A9_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1203. CFG_GPMC_A9_OUT

Address Offset	0x0000 0260		
Physical Address	0x4844 A260	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_a9_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1204. Register Call Summary for Register CFG_GPMC_A9_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1205. CFG_GPMC_AD0_IN

Address Offset	0x0000 0264		
Physical Address	0x4844 A264	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1206. Register Call Summary for Register CFG_GPMC_AD0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1207. CFG_GPMC_AD0_OEN

Address Offset	0x0000 0268		
Physical Address	0x4844 A268	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1208. Register Call Summary for Register CFG_GPMC_AD0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1209. CFG_GPMC_AD0_OUT

Address Offset	0x0000 026C		
Physical Address	0x4844 A26C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad0_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1210. Register Call Summary for Register CFG_GPMC_AD0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1211. CFG_GPMC_AD10_IN

Address Offset	0x0000 0270		
Physical Address	0x4844 A270	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad10_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1212. Register Call Summary for Register CFG_GPMC_AD10_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1213. CFG_GPMC_AD10_OEN

Address Offset	0x0000 0274		
Physical Address	0x4844 A274	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad10_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1214. Register Call Summary for Register CFG_GPMC_AD10_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1215. CFG_GPMC_AD10_OUT

Address Offset	0x0000 0278		
Physical Address	0x4844 A278	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad10_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1216. Register Call Summary for Register CFG_GPMC_AD10_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1217. CFG_GPMC_AD11_IN

Address Offset	0x0000 027C		
Physical Address	0x4844 A27C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad11_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1218. Register Call Summary for Register CFG_GPMC_AD11_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1219. CFG_GPMC_AD11_OEN

Address Offset	0x0000 0280		
Physical Address	0x4844 A280	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad11_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1220. Register Call Summary for Register CFG_GPMC_AD11_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1221. CFG_GPMC_AD11_OUT

Address Offset	0x0000 0284		
Physical Address	0x4844 A284	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad11_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1222. Register Call Summary for Register CFG_GPMC_AD11_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1223. CFG_GPMC_AD12_IN

Address Offset	0x0000 0288		
Physical Address	0x4844 A288	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad12_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1224. Register Call Summary for Register CFG_GPMC_AD12_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1225. CFG_GPMC_AD12_OEN

Address Offset	0x0000 028C		
Physical Address	0x4844 A28C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad12_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1226. Register Call Summary for Register CFG_GPMC_AD12_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1227. CFG_GPMC_AD12_OUT

Address Offset	0x0000 0290		
Physical Address	0x4844 A290	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad12_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1228. Register Call Summary for Register CFG_GPMC_AD12_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1229. CFG_GPMC_AD13_IN

Address Offset	0x0000 0294		
Physical Address	0x4844 A294	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad13_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1230. Register Call Summary for Register CFG_GPMC_AD13_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1231. CFG_GPMC_AD13_OEN

Address Offset	0x0000 0298		
Physical Address	0x4844 A298	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad13_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1232. Register Call Summary for Register CFG_GPMC_AD13_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1233. CFG_GPMC_AD13_OUT

Address Offset	0x0000 029C		
Physical Address	0x4844 A29C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad13_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1234. Register Call Summary for Register CFG_GPMC_AD13_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1235. CFG_GPMC_AD14_IN

Address Offset	0x0000 02A0		
Physical Address	0x4844 A2A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad14_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1236. Register Call Summary for Register CFG_GPMC_AD14_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1237. CFG_GPMC_AD14_OEN

Address Offset	0x0000 02A4		
Physical Address	0x4844 A2A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad14_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1238. Register Call Summary for Register CFG_GPMC_AD14_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1239. CFG_GPMC_AD14_OUT

Address Offset	0x0000 02A8		
Physical Address	0x4844 A2A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad14_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1240. Register Call Summary for Register CFG_GPMC_AD14_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1241. CFG_GPMC_AD15_IN

Address Offset	0x0000 02AC		
Physical Address	0x4844 A2AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad15_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1242. Register Call Summary for Register CFG_GPMC_AD15_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1243. CFG_GPMC_AD15_OEN

Address Offset	0x0000 02B0		
Physical Address	0x4844 A2B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad15_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1244. Register Call Summary for Register CFG_GPMC_AD15_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1245. CFG_GPMC_AD15_OUT

Address Offset	0x0000 02B4		
Physical Address	0x4844 A2B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad15_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1246. Register Call Summary for Register CFG_GPMC_AD15_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1247. CFG_GPMC_AD1_IN

Address Offset	0x0000 02B8		
Physical Address	0x4844 A2B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad1_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1248. Register Call Summary for Register CFG_GPMC_AD1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1249. CFG_GPMC_AD1_OEN

Address Offset	0x0000 02BC		
Physical Address	0x4844 A2BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1250. Register Call Summary for Register CFG_GPMC_AD1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1251. CFG_GPMC_AD1_OUT

Address Offset	0x0000 02C0		
Physical Address	0x4844 A2C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1252. Register Call Summary for Register CFG_GPMC_AD1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1253. CFG_GPMC_AD2_IN

Address Offset	0x0000 02C4		
Physical Address	0x4844 A2C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1254. Register Call Summary for Register CFG_GPMC_AD2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1255. CFG_GPMC_AD2_OEN

Address Offset	0x0000 02C8		
Physical Address	0x4844 A2C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1256. Register Call Summary for Register CFG_GPMC_AD2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1257. CFG_GPMC_AD2_OUT

Address Offset	0x0000 02CC		
Physical Address	0x4844 A2CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad2_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1258. Register Call Summary for Register CFG_GPMC_AD2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1259. CFG_GPMC_AD3_IN

Address Offset	0x0000 02D0		
Physical Address	0x4844 A2D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad3_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1260. Register Call Summary for Register CFG_GPMC_AD3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1261. CFG_GPMC_AD3_OEN

Address Offset	0x0000 02D4		
Physical Address	0x4844 A2D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1262. Register Call Summary for Register CFG_GPMC_AD3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1263. CFG_GPMC_AD3_OUT

Address Offset	0x0000 02D8		
Physical Address	0x4844 A2D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1264. Register Call Summary for Register CFG_GPMC_AD3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1265. CFG_GPMC_AD4_IN

Address Offset	0x0000 02DC		
Physical Address	0x4844 A2DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad4_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1266. Register Call Summary for Register CFG_GPMC_AD4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1267. CFG_GPMC_AD4_OEN

Address Offset	0x0000 02E0		
Physical Address	0x4844 A2E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad4_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1268. Register Call Summary for Register CFG_GPMC_AD4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1269. CFG_GPMC_AD4_OUT

Address Offset	0x0000 02E4		
Physical Address	0x4844 A2E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad4_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1270. Register Call Summary for Register CFG_GPMC_AD4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1271. CFG_GPMC_AD5_IN

Address Offset	0x0000 02E8		
Physical Address	0x4844 A2E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad5_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1272. Register Call Summary for Register CFG_GPMC_AD5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1273. CFG_GPMC_AD5_OEN

Address Offset	0x0000 02EC		
Physical Address	0x4844 A2EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad5_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1274. Register Call Summary for Register CFG_GPMC_AD5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1275. CFG_GPMC_AD5_OUT

Address Offset	0x0000 02F0		
Physical Address	0x4844 A2F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad5_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1276. Register Call Summary for Register CFG_GPMC_AD5_OUT

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1277. CFG_GPMC_AD6_IN

Address Offset	0x0000 02F4		
Physical Address	0x4844 A2F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad6_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1278. Register Call Summary for Register CFG_GPMC_AD6_IN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1279. CFG_GPMC_AD6_OEN

Address Offset	0x0000 02F8		
Physical Address	0x4844 A2F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad6_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1280. Register Call Summary for Register CFG_GPMC_AD6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1281. CFG_GPMC_AD6_OUT

Address Offset	0x0000 02FC		
Physical Address	0x4844 A2FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad6_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1282. Register Call Summary for Register CFG_GPMC_AD6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1283. CFG_GPMC_AD7_IN

Address Offset	0x0000 0300		
Physical Address	0x4844 A300	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad7_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1284. Register Call Summary for Register CFG_GPMC_AD7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1285. CFG_GPMC_AD7_OEN

Address Offset	0x0000 0304		
Physical Address	0x4844 A304	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad7_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1286. Register Call Summary for Register CFG_GPMC_AD7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1287. CFG_GPMC_AD7_OUT

Address Offset	0x0000 0308		
Physical Address	0x4844 A308	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad7_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1288. Register Call Summary for Register CFG_GPMC_AD7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1289. CFG_GPMC_AD8_IN

Address Offset	0x0000 030C		
Physical Address	0x4844 A30C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad8_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1290. Register Call Summary for Register CFG_GPMC_AD8_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1291. CFG_GPMC_AD8_OEN

Address Offset	0x0000 0310		
Physical Address	0x4844 A310	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad8_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1292. Register Call Summary for Register CFG_GPMC_AD8_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1293. CFG_GPMC_AD8_OUT

Address Offset	0x0000 0314		
Physical Address	0x4844 A314	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad8_out interface		
Type	RW		

31-18	17-12	11	10	9-0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1294. Register Call Summary for Register CFG_GPMC_AD8_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1295. CFG_GPMC_AD9_IN

Address Offset	0x0000 0318		
Physical Address	0x4844 A318	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad9_in interface		
Type	RW		

31-18	17-12	11	10	9-0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1296. Register Call Summary for Register CFG_GPMC_AD9_IN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1297. CFG_GPMC_AD9_OEN

Address Offset	0x0000 031C		
Physical Address	0x4844 A31C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad9_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1298. Register Call Summary for Register CFG_GPMC_AD9_OEN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1299. CFG_GPMC_AD9_OUT

Address Offset	0x0000 0320		
Physical Address	0x4844 A320	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ad9_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1300. Register Call Summary for Register CFG_GPMC_AD9_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1301. CFG_GPMC_ADVN_ALE_IN

Address Offset	0x0000 0324		
Physical Address	0x4844 A324	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_advn_ale_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1302. Register Call Summary for Register CFG_GPMC_ADVN_ALE_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1303. CFG_GPMC_ADVN_ALE_OEN

Address Offset	0x0000 0328		
Physical Address	0x4844 A328	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_advn_ale_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1304. Register Call Summary for Register CFG_GPMC_ADVN_ALE_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1305. CFG_GPMC_ADVN_ALE_OUT

Address Offset	0x0000 032C		
Physical Address	0x4844 A32C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_advn_ale_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1306. Register Call Summary for Register CFG_GPMC_ADVN_ALE_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1307. CFG_GPMC_BEN0_IN

Address Offset	0x0000 0330		
Physical Address	0x4844 A330	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ben0_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1308. Register Call Summary for Register CFG_GPMC_BEN0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1309. CFG_GPMC_BEN0_OEN

Address Offset	0x0000 0334		
Physical Address	0x4844 A334	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ben0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1310. Register Call Summary for Register CFG_GPMC_BEN0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1311. CFG_GPMC_BEN0_OUT

Address Offset	0x0000 0338		
Physical Address	0x4844 A338	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ben0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1312. Register Call Summary for Register CFG_GPMC_BEN0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1313. CFG_GPMC_BEN1_IN

Address Offset	0x0000 033C		
Physical Address	0x4844 A33C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ben1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1314. Register Call Summary for Register CFG_GPMC_BEN1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1315. CFG_GPMC_BEN1_OEN

Address Offset	0x0000 0340		
Physical Address	0x4844 A340	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ben1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1316. Register Call Summary for Register CFG_GPMC_BEN1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1317. CFG_GPMC_BEN1_OUT

Address Offset	0x0000 0344		
Physical Address	0x4844 A344	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_ben1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1318. Register Call Summary for Register CFG_GPMC_BEN1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1319. CFG_GPMC_CLK_IN

Address Offset	0x0000 0348		
Physical Address	0x4844 A348	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_clk_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1320. Register Call Summary for Register CFG_GPMC_CLK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1321. CFG_GPMC_CLK_OEN

Address Offset	0x0000 034C		
Physical Address	0x4844 A34C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_clk_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1322. Register Call Summary for Register CFG_GPMC_CLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1323. CFG_GPMC_CLK_OUT

Address Offset	0x0000 0350		
Physical Address	0x4844 A350	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_clk_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1324. Register Call Summary for Register CFG_GPMC_CLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1325. CFG_GPMC_CS0_IN

Address Offset	0x0000 0354		
Physical Address	0x4844 A354	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1326. Register Call Summary for Register CFG_GPMC_CS0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1327. CFG_GPMC_CS0_OEN

Address Offset	0x0000 0358		
Physical Address	0x4844 A358	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1328. Register Call Summary for Register CFG_GPMC_CS0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1329. CFG_GPMC_CS0_OUT

Address Offset	0x0000 035C		
Physical Address	0x4844 A35C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1330. Register Call Summary for Register CFG_GPMC_CS0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1331. CFG_GPMC_CS1_IN

Address Offset	0x0000 0360		
Physical Address	0x4844 A360	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1332. Register Call Summary for Register CFG_GPMC_CS1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1333. CFG_GPMC_CS1_OEN

Address Offset	0x0000 0364		
Physical Address	0x4844 A364	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs1_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1334. Register Call Summary for Register CFG_GPMC_CS1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1335. CFG_GPMC_CS1_OUT

Address Offset	0x0000 0368		
Physical Address	0x4844 A368	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs1_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1336. Register Call Summary for Register CFG_GPMC_CS1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1337. CFG_GPMC_CS2_IN

Address Offset	0x0000 036C		
Physical Address	0x4844 A36C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1338. Register Call Summary for Register CFG_GPMC_CS2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1339. CFG_GPMC_CS2_OEN

Address Offset	0x0000 0370		
Physical Address	0x4844 A370	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1340. Register Call Summary for Register CFG_GPMC_CS2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1341. CFG_GPMC_CS2_OUT

Address Offset	0x0000 0374		
Physical Address	0x4844 A374	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1342. Register Call Summary for Register CFG_GPMC_CS2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1343. CFG_GPMC_CS3_IN

Address Offset	0x0000 0378		
Physical Address	0x4844 A378	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1344. Register Call Summary for Register CFG_GPMC_CS3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1345. CFG_GPMC_CS3_OEN

Address Offset	0x0000 037C		
Physical Address	0x4844 A37C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs3_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1346. Register Call Summary for Register CFG_GPMC_CS3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1347. CFG_GPMC_CS3_OUT

Address Offset	0x0000 0380		
Physical Address	0x4844 A380	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_cs3_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1348. Register Call Summary for Register CFG_GPMC_CS3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1349. CFG_GPMC_OEN_REN_IN

Address Offset	0x0000 0384		
Physical Address	0x4844 A384	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_oen_ren_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1350. Register Call Summary for Register CFG_GPMC_OEN_REN_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1351. CFG_GPMC_OEN_REN_OEN

Address Offset	0x0000 0388		
Physical Address	0x4844 A388	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_oen_ren_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1352. Register Call Summary for Register CFG_GPMC_OEN_REN_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1353. CFG_GPMC_OEN_REN_OUT

Address Offset	0x0000 038C		
Physical Address	0x4844 A38C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_oen_ren_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1354. Register Call Summary for Register CFG_GPMC_OEN_REN_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1355. CFG_GPMC_WAIT0_IN

Address Offset	0x0000 0390		
Physical Address	0x4844 A390	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_wait0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1356. Register Call Summary for Register CFG_GPMC_WAIT0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1357. CFG_GPMC_WAIT0_OEN

Address Offset	0x0000 0394		
Physical Address	0x4844 A394	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_wait0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1358. Register Call Summary for Register CFG_GPMC_WAIT0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1359. CFG_GPMC_WAIT0_OUT

Address Offset	0x0000 0398		
Physical Address	0x4844 A398	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_wait0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1360. Register Call Summary for Register CFG_GPMC_WAIT0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1361. CFG_GPMC_WEN_IN

Address Offset	0x0000 039C		
Physical Address	0x4844 A39C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_wen_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1362. Register Call Summary for Register CFG_GPMC_WEN_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1363. CFG_GPMC_WEN_OEN

Address Offset	0x0000 03A0		
Physical Address	0x4844 A3A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_wen_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1364. Register Call Summary for Register CFG_GPMC_WEN_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1365. CFG_GPMC_WEN_OUT

Address Offset	0x0000 03A4		
Physical Address	0x4844 A3A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_gpmc_wen_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1366. Register Call Summary for Register CFG_GPMC_WEN_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1367. CFG_MCASP1_ACLKR_IN

Address Offset	0x0000 03A8		
Physical Address	0x4844 A3A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_aclkr_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1368. Register Call Summary for Register CFG_MCASP1_ACLKR_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1369. CFG_MCASP1_ACLKR_OEN

Address Offset	0x0000 03AC		
Physical Address	0x4844 A3AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_aclkr_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1370. Register Call Summary for Register CFG_MCASP1_ACLKR_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1371. CFG_MCASP1_ACLKR_OUT

Address Offset	0x0000 03B0		
Physical Address	0x4844 A3B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_aclkr_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1372. Register Call Summary for Register CFG_MCASP1_ACLKR_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1373. CFG_MCASP1_ACLKX_IN

Address Offset	0x0000 03B4		
Physical Address	0x4844 A3B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_aclkx_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1374. Register Call Summary for Register CFG_MCASP1_ACLKX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1375. CFG_MCASP1_ACLKX_OEN

Address Offset	0x0000 03B8		
Physical Address	0x4844 A3B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_aclkx_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1376. Register Call Summary for Register CFG_MCASP1_ACLKX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1377. CFG_MCASP1_ACLKX_OUT

Address Offset	0x0000 03BC		
Physical Address	0x4844 A3BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_aclkx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1378. Register Call Summary for Register CFG_MCASP1_ACLKX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1379. CFG_MCASP1_AXR0_IN

Address Offset	0x0000 03C0		
Physical Address	0x4844 A3C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1380. Register Call Summary for Register CFG_MCASP1_AXR0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1381. CFG_MCASP1_AXR0_OEN

Address Offset	0x0000 03C4		
Physical Address	0x4844 A3C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr0_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1382. Register Call Summary for Register CFG_MCASP1_AXR0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1383. CFG_MCASP1_AXR0_OUT

Address Offset	0x0000 03C8		
Physical Address	0x4844 A3C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr0_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1384. Register Call Summary for Register CFG_MCASP1_AXR0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1385. CFG_MCASP1_AXR10_IN

Address Offset	0x0000 03CC		
Physical Address	0x4844 A3CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr10_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1386. Register Call Summary for Register CFG_MCASP1_AXR10_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1387. CFG_MCASP1_AXR10_OEN

Address Offset	0x0000 03D0		
Physical Address	0x4844 A3D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr10_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1388. Register Call Summary for Register CFG_MCASP1_AXR10_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1389. CFG_MCASP1_AXR10_OUT

Address Offset	0x0000 03D4		
Physical Address	0x4844 A3D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr10_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1390. Register Call Summary for Register CFG_MCASP1_AXR10_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1391. CFG_MCASP1_AXR11_IN

Address Offset	0x0000 03D8		
Physical Address	0x4844 A3D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr11_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1392. Register Call Summary for Register CFG_MCASP1_AXR11_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1393. CFG_MCASP1_AXR11_OEN

Address Offset	0x0000 03DC		
Physical Address	0x4844 A3DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr11_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1394. Register Call Summary for Register CFG_MCASP1_AXR11_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1395. CFG_MCASP1_AXR11_OUT

Address Offset	0x0000 03E0		
Physical Address	0x4844 A3E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr11_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1396. Register Call Summary for Register CFG_MCASP1_AXR11_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1397. CFG_MCASP1_AXR12_IN

Address Offset	0x0000 03E4		
Physical Address	0x4844 A3E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr12_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1398. Register Call Summary for Register CFG_MCASP1_AXR12_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1399. CFG_MCASP1_AXR12_OEN

Address Offset	0x0000 03E8		
Physical Address	0x4844 A3E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr12_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1400. Register Call Summary for Register CFG_MCASP1_AXR12_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1401. CFG_MCASP1_AXR12_OUT

Address Offset	0x0000 03EC		
Physical Address	0x4844 A3EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr12_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1402. Register Call Summary for Register CFG_MCASP1_AXR12_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1403. CFG_MCASP1_AXR13_IN

Address Offset	0x0000 03F0		
Physical Address	0x4844 A3F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr13_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1404. Register Call Summary for Register CFG_MCASP1_AXR13_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1405. CFG_MCASP1_AXR13_OEN

Address Offset	0x0000 03F4		
Physical Address	0x4844 A3F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr13_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1406. Register Call Summary for Register CFG_MCASP1_AXR13_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1407. CFG_MCASP1_AXR13_OUT

Address Offset	0x0000 03F8		
Physical Address	0x4844 A3F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr13_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1408. Register Call Summary for Register CFG_MCASP1_AXR13_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1409. CFG_MCASP1_AXR14_IN

Address Offset	0x0000 03FC		
Physical Address	0x4844 A3FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr14_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1410. Register Call Summary for Register CFG_MCASP1_AXR14_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1411. CFG_MCASP1_AXR14_OEN

Address Offset	0x0000 0400		
Physical Address	0x4844 A400	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr14_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1412. Register Call Summary for Register CFG_MCASP1_AXR14_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1413. CFG_MCASP1_AXR14_OUT

Address Offset	0x0000 0404		
Physical Address	0x4844 A404	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr14_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1414. Register Call Summary for Register CFG_MCASP1_AXR14_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1415. CFG_MCASP1_AXR15_IN

Address Offset	0x0000 0408		
Physical Address	0x4844 A408	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr15_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1416. Register Call Summary for Register CFG_MCASP1_AXR15_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1417. CFG_MCASP1_AXR15_OEN

Address Offset	0x0000 040C		
Physical Address	0x4844 A40C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr15_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1418. Register Call Summary for Register CFG_MCASP1_AXR15_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1419. CFG_MCASP1_AXR15_OUT

Address Offset	0x0000 0410		
Physical Address	0x4844 A410	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr15_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1420. Register Call Summary for Register CFG_MCASP1_AXR15_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1421. CFG_MCASP1_AXR1_IN

Address Offset	0x0000 0414		
Physical Address	0x4844 A414	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1422. Register Call Summary for Register CFG_MCASP1_AXR1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1423. CFG_MCASP1_AXR1_OEN

Address Offset	0x0000 0418		
Physical Address	0x4844 A418	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1424. Register Call Summary for Register CFG_MCASP1_AXR1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1425. CFG_MCASP1_AXR1_OUT

Address Offset	0x0000 041C		
Physical Address	0x4844 A41C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr1_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1426. Register Call Summary for Register CFG_MCASP1_AXR1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1427. CFG_MCASP1_AXR2_IN

Address Offset	0x0000 0420		
Physical Address	0x4844 A420	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr2_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1428. Register Call Summary for Register CFG_MCASP1_AXR2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1429. CFG_MCASP1_AXR2_OEN

Address Offset	0x0000 0424		
Physical Address	0x4844 A424	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1430. Register Call Summary for Register CFG_MCASP1_AXR2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1431. CFG_MCASP1_AXR2_OUT

Address Offset	0x0000 0428		
Physical Address	0x4844 A428	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1432. Register Call Summary for Register CFG_MCASP1_AXR2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1433. CFG_MCASP1_AXR3_IN

Address Offset	0x0000 042C		
Physical Address	0x4844 A42C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1434. Register Call Summary for Register CFG_MCASP1_AXR3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1435. CFG_MCASP1_AXR3_OEN

Address Offset	0x0000 0430		
Physical Address	0x4844 A430	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1436. Register Call Summary for Register CFG_MCASP1_AXR3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1437. CFG_MCASP1_AXR3_OUT

Address Offset	0x0000 0434		
Physical Address	0x4844 A434	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1438. Register Call Summary for Register CFG_MCASP1_AXR3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1439. CFG_MCASP1_AXR4_IN

Address Offset	0x0000 0438		
Physical Address	0x4844 A438	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr4_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1440. Register Call Summary for Register CFG_MCASP1_AXR4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1441. CFG_MCASP1_AXR4_OEN

Address Offset	0x0000 043C		
Physical Address	0x4844 A43C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr4_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1442. Register Call Summary for Register CFG_MCASP1_AXR4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1443. CFG_MCASP1_AXR4_OUT

Address Offset	0x0000 0440		
Physical Address	0x4844 A440	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr4_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1444. Register Call Summary for Register CFG_MCASP1_AXR4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1445. CFG_MCASP1_AXR5_IN

Address Offset	0x0000 0444		
Physical Address	0x4844 A444	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr5_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1446. Register Call Summary for Register CFG_MCASP1_AXR5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1447. CFG_MCASP1_AXR5_OEN

Address Offset	0x0000 0448		
Physical Address	0x4844 A448	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr5_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1448. Register Call Summary for Register CFG_MCASP1_AXR5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1449. CFG_MCASP1_AXR5_OUT

Address Offset	0x0000 044C		
Physical Address	0x4844 A44C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr5_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1450. Register Call Summary for Register CFG_MCASP1_AXR5_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1451. CFG_MCASP1_AXR6_IN

Address Offset	0x0000 0450		
Physical Address	0x4844 A450	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr6_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1452. Register Call Summary for Register CFG_MCASP1_AXR6_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1453. CFG_MCASP1_AXR6_OEN

Address Offset	0x0000 0454		
Physical Address	0x4844 A454	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr6_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1454. Register Call Summary for Register CFG_MCASP1_AXR6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1455. CFG_MCASP1_AXR6_OUT

Address Offset	0x0000 0458		
Physical Address	0x4844 A458	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr6_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1456. Register Call Summary for Register CFG_MCASP1_AXR6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1457. CFG_MCASP1_AXR7_IN

Address Offset	0x0000 045C		
Physical Address	0x4844 A45C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr7_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1458. Register Call Summary for Register CFG_MCASP1_AXR7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1459. CFG_MCASP1_AXR7_OEN

Address Offset	0x0000 0460		
Physical Address	0x4844 A460	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr7_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1460. Register Call Summary for Register CFG_MCASP1_AXR7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1461. CFG_MCASP1_AXR7_OUT

Address Offset	0x0000 0464		
Physical Address	0x4844 A464	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr7_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1462. Register Call Summary for Register CFG_MCASP1_AXR7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1463. CFG_MCASP1_AXR8_IN

Address Offset	0x0000 0468		
Physical Address	0x4844 A468	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr8_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1464. Register Call Summary for Register CFG_MCASP1_AXR8_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1465. CFG_MCASP1_AXR8_OEN

Address Offset	0x0000 046C		
Physical Address	0x4844 A46C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr8_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1466. Register Call Summary for Register CFG_MCASP1_AXR8_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1467. CFG_MCASP1_AXR8_OUT

Address Offset	0x0000 0470		
Physical Address	0x4844 A470	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr8_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1468. Register Call Summary for Register CFG_MCASP1_AXR8_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1469. CFG_MCASP1_AXR9_IN

Address Offset	0x0000 0474		
Physical Address	0x4844 A474	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr9_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1470. Register Call Summary for Register CFG_MCASP1_AXR9_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1471. CFG_MCASP1_AXR9_OEN

Address Offset	0x0000 0478		
Physical Address	0x4844 A478	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr9_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1472. Register Call Summary for Register CFG_MCASP1_AXR9_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1473. CFG_MCASP1_AXR9_OUT

Address Offset	0x0000 047C		
Physical Address	0x4844 A47C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_axr9_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1474. Register Call Summary for Register CFG_MCASP1_AXR9_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1475. CFG_MCASP1_FSR_IN

Address Offset	0x0000 0480		
Physical Address	0x4844 A480	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_fsr_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1476. Register Call Summary for Register CFG_MCASP1_FSR_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1477. CFG_MCASP1_FSR_OEN

Address Offset	0x0000 0484		
Physical Address	0x4844 A484	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_fsr_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1478. Register Call Summary for Register CFG_MCASP1_FSR_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1479. CFG_MCASP1_FSR_OUT

Address Offset	0x0000 0488		
Physical Address	0x4844 A488	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_fsr_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1480. Register Call Summary for Register CFG_MCASP1_FSR_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1481. CFG_MCASP1_FSX_IN

Address Offset	0x0000 048C		
Physical Address	0x4844 A48C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_fsx_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1482. Register Call Summary for Register CFG_MCASP1_FSX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1483. CFG_MCASP1_FSX_OEN

Address Offset	0x0000 0490		
Physical Address	0x4844 A490	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_fsx_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1484. Register Call Summary for Register CFG_MCASP1_FSX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1485. CFG_MCASP1_FSX_OUT

Address Offset	0x0000 0494		
Physical Address	0x4844 A494	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp1_fsx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1486. Register Call Summary for Register CFG_MCASP1_FSX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1487. CFG_MCASP2_ACLKR_IN

Address Offset	0x0000 0498		
Physical Address	0x4844 A498	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_aclkr_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1488. Register Call Summary for Register CFG_MCASP2_ACLKR_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1489. CFG_MCASP2_ACLKR_OEN

Address Offset	0x0000 049C		
Physical Address	0x4844 A49C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_aclkr_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1490. Register Call Summary for Register CFG_MCASP2_ACLKR_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1491. CFG_MCASP2_ACLKR_OUT

Address Offset	0x0000 04A0		
Physical Address	0x4844 A4A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_aclkr_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1492. Register Call Summary for Register CFG_MCASP2_ACLKR_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1493. CFG_MCASP2_ACLKX_IN

Address Offset	0x0000 04A4		
Physical Address	0x4844 A4A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_aclkx_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1494. Register Call Summary for Register CFG_MCASP2_ACLKX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1495. CFG_MCASP2_ACLKX_OEN

Address Offset	0x0000 04A8		
Physical Address	0x4844 A4A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_aclkx_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1496. Register Call Summary for Register CFG_MCASP2_ACLKX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1497. CFG_MCASP2_ACLKX_OUT

Address Offset	0x0000 04AC		
Physical Address	0x4844 A4AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_aclkx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1498. Register Call Summary for Register CFG_MCASP2_ACLKX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1499. CFG_MCASP2_AXR0_IN

Address Offset	0x0000 04B0		
Physical Address	0x4844 A4B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1500. Register Call Summary for Register CFG_MCASP2_AXR0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1501. CFG_MCASP2_AXR0_OEN

Address Offset	0x0000 04B4		
Physical Address	0x4844 A4B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1502. Register Call Summary for Register CFG_MCASP2_AXR0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1503. CFG_MCASP2_AXR0_OUT

Address Offset	0x0000 04B8		
Physical Address	0x4844 A4B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1504. Register Call Summary for Register CFG_MCASP2_AXR0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1505. CFG_MCASP2_AXR1_IN

Address Offset	0x0000 04BC		
Physical Address	0x4844 A4BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1506. Register Call Summary for Register CFG_MCASP2_AXR1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1507. CFG_MCASP2_AXR1_OEN

Address Offset	0x0000 04C0		
Physical Address	0x4844 A4C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1508. Register Call Summary for Register CFG_MCASP2_AXR1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1509. CFG_MCASP2_AXR1_OUT

Address Offset	0x0000 04C4		
Physical Address	0x4844 A4C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1510. Register Call Summary for Register CFG_MCASP2_AXR1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1511. CFG_MCASP2_AXR2_IN

Address Offset	0x0000 04C8		
Physical Address	0x4844 A4C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1512. Register Call Summary for Register CFG_MCASP2_AXR2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1513. CFG_MCASP2_AXR2_OEN

Address Offset	0x0000 04CC		
Physical Address	0x4844 A4CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1514. Register Call Summary for Register CFG_MCASP2_AXR2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1515. CFG_MCASP2_AXR2_OUT

Address Offset	0x0000 04D0		
Physical Address	0x4844 A4D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1516. Register Call Summary for Register CFG_MCASP2_AXR2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1517. CFG_MCASP2_AXR3_IN

Address Offset	0x0000 04D4		
Physical Address	0x4844 A4D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr3_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1518. Register Call Summary for Register CFG_MCASP2_AXR3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1519. CFG_MCASP2_AXR3_OEN

Address Offset	0x0000 04D8		
Physical Address	0x4844 A4D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr3_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1520. Register Call Summary for Register CFG_MCASP2_AXR3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1521. CFG_MCASP2_AXR3_OUT

Address Offset	0x0000 04DC		
Physical Address	0x4844 A4DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1522. Register Call Summary for Register CFG_MCASP2_AXR3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1523. CFG_MCASP2_AXR4_IN

Address Offset	0x0000 04E0		
Physical Address	0x4844 A4E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr4_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1524. Register Call Summary for Register CFG_MCASP2_AXR4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1525. CFG_MCASP2_AXR4_OEN

Address Offset	0x0000 04E4		
Physical Address	0x4844 A4E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr4_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1526. Register Call Summary for Register CFG_MCASP2_AXR4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1527. CFG_MCASP2_AXR4_OUT

Address Offset	0x0000 04E8		
Physical Address	0x4844 A4E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr4_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1528. Register Call Summary for Register CFG_MCASP2_AXR4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1529. CFG_MCASP2_AXR5_IN

Address Offset	0x0000 04EC		
Physical Address	0x4844 A4EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr5_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1530. Register Call Summary for Register CFG_MCASP2_AXR5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1531. CFG_MCASP2_AXR5_OEN

Address Offset	0x0000 04F0		
Physical Address	0x4844 A4F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr5_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1532. Register Call Summary for Register CFG_MCASP2_AXR5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1533. CFG_MCASP2_AXR5_OUT

Address Offset	0x0000 04F4		
Physical Address	0x4844 A4F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr5_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1534. Register Call Summary for Register CFG_MCASP2_AXR5_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1535. CFG_MCASP2_AXR6_IN

Address Offset	0x0000 04F8		
Physical Address	0x4844 A4F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr6_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1536. Register Call Summary for Register CFG_MCASP2_AXR6_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1537. CFG_MCASP2_AXR6_OEN

Address Offset	0x0000 04FC		
Physical Address	0x4844 A4FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr6_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1538. Register Call Summary for Register CFG_MCASP2_AXR6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1539. CFG_MCASP2_AXR6_OUT

Address Offset	0x0000 0500		
Physical Address	0x4844 A500	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr6_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1540. Register Call Summary for Register CFG_MCASP2_AXR6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1541. CFG_MCASP2_AXR7_IN

Address Offset	0x0000 0504		
Physical Address	0x4844 A504	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr7_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1542. Register Call Summary for Register CFG_MCASP2_AXR7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1543. CFG_MCASP2_AXR7_OEN

Address Offset	0x0000 0508		
Physical Address	0x4844 A508	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr7_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1544. Register Call Summary for Register CFG_MCASP2_AXR7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1545. CFG_MCASP2_AXR7_OUT

Address Offset	0x0000 050C		
Physical Address	0x4844 A50C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_axr7_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1546. Register Call Summary for Register CFG_MCASP2_AXR7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1547. CFG_MCASP2_FSR_IN

Address Offset	0x0000 0510		
Physical Address	0x4844 A510	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_fsr_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1548. Register Call Summary for Register CFG_MCASP2_FSR_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1549. CFG_MCASP2_FSR_OEN

Address Offset	0x0000 0514		
Physical Address	0x4844 A514	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_fsr_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1550. Register Call Summary for Register CFG_MCASP2_FSR_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1551. CFG_MCASP2_FSR_OUT

Address Offset	0x0000 0518		
Physical Address	0x4844 A518	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_fsr_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1552. Register Call Summary for Register CFG_MCASP2_FSR_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1553. CFG_MCASP2_FSX_IN

Address Offset	0x0000 051C		
Physical Address	0x4844 A51C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_fsx_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1554. Register Call Summary for Register CFG_MCASP2_FSX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1555. CFG_MCASP2_FSX_OEN

Address Offset	0x0000 0520		
Physical Address	0x4844 A520	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_fsx_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1556. Register Call Summary for Register CFG_MCASP2_FSX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1557. CFG_MCASP2_FSX_OUT

Address Offset	0x0000 0524		
Physical Address	0x4844 A524	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp2_fsx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1558. Register Call Summary for Register CFG_MCASP2_FSX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1559. CFG_MCASP3_ACLKX_IN

Address Offset	0x0000 0528		
Physical Address	0x4844 A528	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_aclkx_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1560. Register Call Summary for Register CFG_MCASP3_ACLKX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1561. CFG_MCASP3_ACLKX_OEN

Address Offset	0x0000 052C		
Physical Address	0x4844 A52C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_aclkx_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1562. Register Call Summary for Register CFG_MCASP3_ACLKX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1563. CFG_MCASP3_ACLKX_OUT

Address Offset	0x0000 0530		
Physical Address	0x4844 A530	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_aclkx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1564. Register Call Summary for Register CFG_MCASP3_ACLKX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1565. CFG_MCASP3_AXR0_IN

Address Offset	0x0000 0534		
Physical Address	0x4844 A534	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_axr0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1566. Register Call Summary for Register CFG_MCASP3_AXR0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1567. CFG_MCASP3_AXR0_OEN

Address Offset	0x0000 0538		
Physical Address	0x4844 A538	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_axr0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1568. Register Call Summary for Register CFG_MCASP3_AXR0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1569. CFG_MCASP3_AXR0_OUT

Address Offset	0x0000 053C		
Physical Address	0x4844 A53C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_axr0_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1570. Register Call Summary for Register CFG_MCASP3_AXR0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1571. CFG_MCASP3_AXR1_IN

Address Offset	0x0000 0540		
Physical Address	0x4844 A540	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_axr1_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1572. Register Call Summary for Register CFG_MCASP3_AXR1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1573. CFG_MCASP3_AXR1_OEN

Address Offset	0x0000 0544		
Physical Address	0x4844 A544	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_axr1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1574. Register Call Summary for Register CFG_MCASP3_AXR1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1575. CFG_MCASP3_AXR1_OUT

Address Offset	0x0000 0548		
Physical Address	0x4844 A548	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_axr1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1576. Register Call Summary for Register CFG_MCASP3_AXR1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1577. CFG_MCASP3_FSX_IN

Address Offset	0x0000 054C		
Physical Address	0x4844 A54C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_fsx_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1578. Register Call Summary for Register CFG_MCASP3_FSX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1579. CFG_MCASP3_FSX_OEN

Address Offset	0x0000 0550		
Physical Address	0x4844 A550	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_fsx_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1580. Register Call Summary for Register CFG_MCASP3_FSX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1581. CFG_MCASP3_FSX_OUT

Address Offset	0x0000 0554		
Physical Address	0x4844 A554	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp3_fsx_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1582. Register Call Summary for Register CFG_MCASP3_FSX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1583. CFG_MCASP4_ACLKX_IN

Address Offset	0x0000 0558		
Physical Address	0x4844 A558	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_aclkx_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1584. Register Call Summary for Register CFG_MCASP4_ACLKX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1585. CFG_MCASP4_ACLKX_OEN

Address Offset	0x0000 055C		
Physical Address	0x4844 A55C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_aclkx_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1586. Register Call Summary for Register CFG_MCASP4_ACLKX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1587. CFG_MCASP4_ACLKX_OUT

Address Offset	0x0000 0560		
Physical Address	0x4844 A560	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_aclkx_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1588. Register Call Summary for Register CFG_MCASP4_ACLKX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1589. CFG_MCASP4_AXR0_IN

Address Offset	0x0000 0564		
Physical Address	0x4844 A564	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_axr0_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1590. Register Call Summary for Register CFG_MCASP4_AXR0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1591. CFG_MCASP4_AXR0_OEN

Address Offset	0x0000 0568		
Physical Address	0x4844 A568	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_axr0_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1592. Register Call Summary for Register CFG_MCASP4_AXR0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1593. CFG_MCASP4_AXR0_OUT

Address Offset	0x0000 056C		
Physical Address	0x4844 A56C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_axr0_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1594. Register Call Summary for Register CFG_MCASP4_AXR0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1595. CFG_MCASP4_AXR1_IN

Address Offset	0x0000 0570		
Physical Address	0x4844 A570	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_axr1_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1596. Register Call Summary for Register CFG_MCASP4_AXR1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1597. CFG_MCASP4_AXR1_OEN

Address Offset	0x0000 0574		
Physical Address	0x4844 A574	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_axr1_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1598. Register Call Summary for Register CFG_MCASP4_AXR1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1599. CFG_MCASP4_AXR1_OUT

Address Offset	0x0000 0578		
Physical Address	0x4844 A578	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_axr1_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1600. Register Call Summary for Register CFG_MCASP4_AXR1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1601. CFG_MCASP4_FSX_IN

Address Offset	0x0000 057C		
Physical Address	0x4844 A57C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_fsx_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1602. Register Call Summary for Register CFG_MCASP4_FSX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1603. CFG_MCASP4_FSX_OEN

Address Offset	0x0000 0580		
Physical Address	0x4844 A580	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_fsx_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1604. Register Call Summary for Register CFG_MCASP4_FSX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1605. CFG_MCASP4_FSX_OUT

Address Offset	0x0000 0584		
Physical Address	0x4844 A584	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp4_fsx_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1606. Register Call Summary for Register CFG_MCASP4_FSX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1607. CFG_MCASP5_ACLKX_IN

Address Offset	0x0000 0588		
Physical Address	0x4844 A588	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_aclkx_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1608. Register Call Summary for Register CFG_MCASP5_ACLKX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1609. CFG_MCASP5_ACLKX_OEN

Address Offset	0x0000 058C		
Physical Address	0x4844 A58C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_aclkx_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1610. Register Call Summary for Register CFG_MCASP5_ACLKX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1611. CFG_MCASP5_ACLKX_OUT

Address Offset	0x0000 0590		
Physical Address	0x4844 A590	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_aclkx_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1612. Register Call Summary for Register CFG_MCASP5_ACLKX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1613. CFG_MCASP5_AXR0_IN

Address Offset	0x0000 0594		
Physical Address	0x4844 A594	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_axr0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1614. Register Call Summary for Register CFG_MCASP5_AXR0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1615. CFG_MCASP5_AXR0_OEN

Address Offset	0x0000 0598		
Physical Address	0x4844 A598	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_axr0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1616. Register Call Summary for Register CFG_MCASP5_AXR0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1617. CFG_MCASP5_AXR0_OUT

Address Offset	0x0000 059C		
Physical Address	0x4844 A59C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_axr0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1618. Register Call Summary for Register CFG_MCASP5_AXR0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1619. CFG_MCASP5_AXR1_IN

Address Offset	0x0000 05A0		
Physical Address	0x4844 A5A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_axr1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1620. Register Call Summary for Register CFG_MCASP5_AXR1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1621. CFG_MCASP5_AXR1_OEN

Address Offset	0x0000 05A4		
Physical Address	0x4844 A5A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_axr1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1622. Register Call Summary for Register CFG_MCASP5_AXR1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1623. CFG_MCASP5_AXR1_OUT

Address Offset	0x0000 05A8		
Physical Address	0x4844 A5A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_axr1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1624. Register Call Summary for Register CFG_MCASP5_AXR1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1625. CFG_MCASP5_FSX_IN

Address Offset	0x0000 05AC		
Physical Address	0x4844 A5AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_fsx_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1626. Register Call Summary for Register CFG_MCASP5_FSX_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1627. CFG_MCASP5_FSX_OEN

Address Offset	0x0000 05B0		
Physical Address	0x4844 A5B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_fsx_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1628. Register Call Summary for Register CFG_MCASP5_FSX_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1629. CFG_MCASP5_FSX_OUT

Address Offset	0x0000 05B4		
Physical Address	0x4844 A5B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mcasp5_fsx_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1630. Register Call Summary for Register CFG_MCASP5_FSX_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1631. CFG_MDIO_D_IN

Address Offset	0x0000 05B8		
Physical Address	0x4844 A5B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mdio_d_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1632. Register Call Summary for Register CFG_MDIO_D_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1633. CFG_MDIO_D_OEN

Address Offset	0x0000 05BC		
Physical Address	0x4844 A5BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mdio_d_oen interface		
Type	RW		

Bits 31–18	Bits 17–12	Bit 11	Bit 10	Bits 9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1634. Register Call Summary for Register CFG_MDIO_D_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1635. CFG_MDIO_D_OUT

Address Offset	0x0000 05C0		
Physical Address	0x4844 A5C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mdio_d_out interface		
Type	RW		

Bits 31–18	Bits 17–12	Bit 11	Bit 10	Bits 9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1636. Register Call Summary for Register CFG_MDIO_D_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1637. CFG_MDIO_MCLK_IN

Address Offset	0x0000 05C4		
Physical Address	0x4844 A5C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mdio_mclk_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1638. Register Call Summary for Register CFG_MDIO_MCLK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1639. CFG_MDIO_MCLK_OEN

Address Offset	0x0000 05C8		
Physical Address	0x4844 A5C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mdio_mclk_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1640. Register Call Summary for Register CFG_MDIO_MCLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1641. CFG_MDIO_MCLK_OUT

Address Offset	0x0000 05CC		
Physical Address	0x4844 A5CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mdio_mclk_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1642. Register Call Summary for Register CFG_MDIO_MCLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1643. CFG_MLBP_CLK_N_IN

Address Offset	0x0000 05D0		
Physical Address	0x4844 A5D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_clk_n_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1644. Register Call Summary for Register CFG_MLBP_CLK_N_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1645. CFG_MLBP_CLK_N_OEN

Address Offset	0x0000 05D4		
Physical Address	0x4844 A5D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_clk_n_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1646. Register Call Summary for Register CFG_MLBP_CLK_N_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1647. CFG_MLBP_CLK_N_OUT

Address Offset	0x0000 05D8		
Physical Address	0x4844 A5D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_clk_n_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1648. Register Call Summary for Register CFG_MLBP_CLK_N_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1649. CFG_MLBP_CLK_P_IN

Address Offset	0x0000 05DC		
Physical Address	0x4844 A5DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_clk_p_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1650. Register Call Summary for Register CFG_MLBP_CLK_P_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1651. CFG_MLBP_CLK_P_OEN

Address Offset	0x0000 05E0		
Physical Address	0x4844 A5E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_clk_p_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1652. Register Call Summary for Register CFG_MLBP_CLK_P_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1653. CFG_MLBP_CLK_P_OUT

Address Offset	0x0000 05E4		
Physical Address	0x4844 A5E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_clk_p_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1654. Register Call Summary for Register CFG_MLBP_CLK_P_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1655. CFG_MLBP_DAT_N_IN

Address Offset	0x0000 05E8		
Physical Address	0x4844 A5E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_dat_n_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1656. Register Call Summary for Register CFG_MLBP_DAT_N_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1657. CFG_MLBP_DAT_N_OEN

Address Offset	0x0000 05EC		
Physical Address	0x4844 A5EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_dat_n_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1658. Register Call Summary for Register CFG_MLBP_DAT_N_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1659. CFG_MLBP_DAT_N_OUT

Address Offset	0x0000 05F0		
Physical Address	0x4844 A5F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_dat_n_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1660. Register Call Summary for Register CFG_MLBP_DAT_N_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1661. CFG_MLBP_DAT_P_IN

Address Offset	0x0000 05F4		
Physical Address	0x4844 A5F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_dat_p_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1662. Register Call Summary for Register CFG_MLBP_DAT_P_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1663. CFG_MLBP_DAT_P_OEN

Address Offset	0x0000 05F8		
Physical Address	0x4844 A5F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_dat_p_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1664. Register Call Summary for Register CFG_MLBP_DAT_P_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1665. CFG_MLBP_DAT_P_OUT

Address Offset	0x0000 05FC		
Physical Address	0x4844 A5FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_dat_p_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1666. Register Call Summary for Register CFG_MLBP_DAT_P_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1667. CFG_MLBP_SIG_N_IN

Address Offset	0x0000 0600		
Physical Address	0x4844 A600	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_sig_n_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1668. Register Call Summary for Register CFG_MLBP_SIG_N_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1669. CFG_MLBP_SIG_N_OEN

Address Offset	0x0000 0604		
Physical Address	0x4844 A604	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_sig_n_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1670. Register Call Summary for Register CFG_MLBP_SIG_N_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1671. CFG_MLBP_SIG_N_OUT

Address Offset	0x0000 0608		
Physical Address	0x4844 A608	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_sig_n_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1672. Register Call Summary for Register CFG_MLBP_SIG_N_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1673. CFG_MLBP_SIG_P_IN

Address Offset	0x0000 060C		
Physical Address	0x4844 A60C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_sig_p_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1674. Register Call Summary for Register CFG_MLBP_SIG_P_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1675. CFG_MLBP_SIG_P_OEN

Address Offset	0x0000 0610		
Physical Address	0x4844 A610	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_sig_p_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1676. Register Call Summary for Register CFG_MLBP_SIG_P_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1677. CFG_MLBP_SIG_P_OUT

Address Offset	0x0000 0614		
Physical Address	0x4844 A614	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mlbp_sig_p_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1678. Register Call Summary for Register CFG_MLBP_SIG_P_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1679. CFG_MMC1_CLK_IN

Address Offset	0x0000 0618		
Physical Address	0x4844 A618	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_clk_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1680. Register Call Summary for Register CFG_MMC1_CLK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1681. CFG_MMC1_CLK_OEN

Address Offset	0x0000 061C		
Physical Address	0x4844 A61C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_clk_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1682. Register Call Summary for Register CFG_MMC1_CLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1683. CFG_MMC1_CLK_OUT

Address Offset	0x0000 0620		
Physical Address	0x4844 A620	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_clk_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1684. Register Call Summary for Register CFG_MMC1_CLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1685. CFG_MMC1_CMD_IN

Address Offset	0x0000 0624		
Physical Address	0x4844 A624	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_cmd_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1686. Register Call Summary for Register CFG_MMC1_CMD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1687. CFG_MMC1_CMD_OEN

Address Offset	0x0000 0628		
Physical Address	0x4844 A628	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_cmd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1688. Register Call Summary for Register CFG_MMC1_CMD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1689. CFG_MMC1_CMD_OUT

Address Offset	0x0000 062C		
Physical Address	0x4844 A62C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_cmd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1690. Register Call Summary for Register CFG_MMC1_CMD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1691. CFG_MMC1_DAT0_IN

Address Offset	0x0000 0630		
Physical Address	0x4844 A630	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1692. Register Call Summary for Register CFG_MMC1_DAT0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1693. CFG_MMC1_DAT0_OEN

Address Offset	0x0000 0634		
Physical Address	0x4844 A634	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat0_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1694. Register Call Summary for Register CFG_MMC1_DAT0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1695. CFG_MMC1_DAT0_OUT

Address Offset	0x0000 0638		
Physical Address	0x4844 A638	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat0_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1696. Register Call Summary for Register CFG_MMC1_DAT0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1697. CFG_MMC1_DAT1_IN

Address Offset	0x0000 063C		
Physical Address	0x4844 A63C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat1_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1698. Register Call Summary for Register CFG_MMC1_DAT1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1699. CFG_MMC1_DAT1_OEN

Address Offset	0x0000 0640		
Physical Address	0x4844 A640	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat1_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1700. Register Call Summary for Register CFG_MMC1_DAT1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1701. CFG_MMC1_DAT1_OUT

Address Offset	0x0000 0644		
Physical Address	0x4844 A644	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1702. Register Call Summary for Register CFG_MMC1_DAT1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1703. CFG_MMC1_DAT2_IN

Address Offset	0x0000 0648		
Physical Address	0x4844 A648	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1704. Register Call Summary for Register CFG_MMC1_DAT2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1705. CFG_MMC1_DAT2_OEN

Address Offset	0x0000 064C		
Physical Address	0x4844 A64C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat2_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1706. Register Call Summary for Register CFG_MMC1_DAT2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1707. CFG_MMC1_DAT2_OUT

Address Offset	0x0000 0650		
Physical Address	0x4844 A650	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat2_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1708. Register Call Summary for Register CFG_MMC1_DAT2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1709. CFG_MMC1_DAT3_IN

Address Offset	0x0000 0654		
Physical Address	0x4844 A654	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1710. Register Call Summary for Register CFG_MMC1_DAT3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1711. CFG_MMC1_DAT3_OEN

Address Offset	0x0000 0658		
Physical Address	0x4844 A658	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1712. Register Call Summary for Register CFG_MMC1_DAT3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1713. CFG_MMC1_DAT3_OUT

Address Offset	0x0000 065C		
Physical Address	0x4844 A65C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_dat3_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1714. Register Call Summary for Register CFG_MMC1_DAT3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1715. CFG_MMC1_SDCD_IN

Address Offset	0x0000 0660		
Physical Address	0x4844 A660	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_sdcd_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1716. Register Call Summary for Register CFG_MMC1_SDCD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1717. CFG_MMC1_SDCD_OEN

Address Offset	0x0000 0664		
Physical Address	0x4844 A664	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_sdcd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1718. Register Call Summary for Register CFG_MMC1_SDCD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1719. CFG_MMC1_SDCD_OUT

Address Offset	0x0000 0668		
Physical Address	0x4844 A668	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_sdcd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1720. Register Call Summary for Register CFG_MMC1_SDCD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1721. CFG_MMC1_SDWP_IN

Address Offset	0x0000 066C		
Physical Address	0x4844 A66C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_sdwp_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1722. Register Call Summary for Register CFG_MMC1_SDWP_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1723. CFG_MMC1_SDWP_OEN

Address Offset	0x0000 0670		
Physical Address	0x4844 A670	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_sdwp_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1724. Register Call Summary for Register CFG_MMC1_SDWP_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1725. CFG_MMC1_SDWP_OUT

Address Offset	0x0000 0674		
Physical Address	0x4844 A674	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc1_sdwp_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1726. Register Call Summary for Register CFG_MMC1_SDWP_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1727. CFG_MMC3_CLK_IN

Address Offset	0x0000 0678		
Physical Address	0x4844 A678	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_clk_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1728. Register Call Summary for Register CFG_MMC3_CLK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1729. CFG_MMC3_CLK_OEN

Address Offset	0x0000 067C		
Physical Address	0x4844 A67C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_clk_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1730. Register Call Summary for Register CFG_MMC3_CLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1731. CFG_MMC3_CLK_OUT

Address Offset	0x0000 0680		
Physical Address	0x4844 A680	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_clk_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1732. Register Call Summary for Register CFG_MMC3_CLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1733. CFG_MMC3_CMD_IN

Address Offset	0x0000 0684		
Physical Address	0x4844 A684	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_cmd_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1734. Register Call Summary for Register CFG_MMC3_CMD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1735. CFG_MMC3_CMD_OEN

Address Offset	0x0000 0688		
Physical Address	0x4844 A688	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_cmd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1736. Register Call Summary for Register CFG_MMC3_CMD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1737. CFG_MMC3_CMD_OUT

Address Offset	0x0000 068C		
Physical Address	0x4844 A68C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_cmd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1738. Register Call Summary for Register CFG_MMC3_CMD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1739. CFG_MMC3_DAT0_IN

Address Offset	0x0000 0690		
Physical Address	0x4844 A690	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1740. Register Call Summary for Register CFG_MMC3_DAT0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1741. CFG_MMC3_DAT0_OEN

Address Offset	0x0000 0694		
Physical Address	0x4844 A694	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1742. Register Call Summary for Register CFG_MMC3_DAT0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1743. CFG_MMC3_DAT0_OUT

Address Offset	0x0000 0698		
Physical Address	0x4844 A698	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1744. Register Call Summary for Register CFG_MMC3_DAT0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1745. CFG_MMC3_DAT1_IN

Address Offset	0x0000 069C		
Physical Address	0x4844 A69C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1746. Register Call Summary for Register CFG_MMC3_DAT1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1747. CFG_MMC3_DAT1_OEN

Address Offset	0x0000 06A0		
Physical Address	0x4844 A6A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1748. Register Call Summary for Register CFG_MMC3_DAT1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1749. CFG_MMC3_DAT1_OUT

Address Offset	0x0000 06A4		
Physical Address	0x4844 A6A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat1_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1750. Register Call Summary for Register CFG_MMC3_DAT1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1751. CFG_MMC3_DAT2_IN

Address Offset	0x0000 06A8		
Physical Address	0x4844 A6A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat2_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1752. Register Call Summary for Register CFG_MMC3_DAT2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1753. CFG_MMC3_DAT2_OEN

Address Offset	0x0000 06AC		
Physical Address	0x4844 A6AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat2_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1754. Register Call Summary for Register CFG_MMC3_DAT2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1755. CFG_MMC3_DAT2_OUT

Address Offset	0x0000 06B0		
Physical Address	0x4844 A6B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat2_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1756. Register Call Summary for Register CFG_MMC3_DAT2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1757. CFG_MMC3_DAT3_IN

Address Offset	0x0000 06B4		
Physical Address	0x4844 A6B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1758. Register Call Summary for Register CFG_MMC3_DAT3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1759. CFG_MMC3_DAT3_OEN

Address Offset	0x0000 06B8		
Physical Address	0x4844 A6B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1760. Register Call Summary for Register CFG_MMC3_DAT3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1761. CFG_MMC3_DAT3_OUT

Address Offset	0x0000 06BC		
Physical Address	0x4844 A6BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat3_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1762. Register Call Summary for Register CFG_MMC3_DAT3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1763. CFG_MMC3_DAT4_IN

Address Offset	0x0000 06C0		
Physical Address	0x4844 A6C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat4_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1764. Register Call Summary for Register CFG_MMC3_DAT4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1765. CFG_MMC3_DAT4_OEN

Address Offset	0x0000 06C4		
Physical Address	0x4844 A6C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat4_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1766. Register Call Summary for Register CFG_MMC3_DAT4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1767. CFG_MMC3_DAT4_OUT

Address Offset	0x0000 06C8		
Physical Address	0x4844 A6C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat4_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1768. Register Call Summary for Register CFG_MMC3_DAT4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1769. CFG_MMC3_DAT5_IN

Address Offset	0x0000 06CC		
Physical Address	0x4844 A6CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat5_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1770. Register Call Summary for Register CFG_MMC3_DAT5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1771. CFG_MMC3_DAT5_OEN

Address Offset	0x0000 06D0		
Physical Address	0x4844 A6D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat5_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1772. Register Call Summary for Register CFG_MMC3_DAT5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1773. CFG_MMC3_DAT5_OUT

Address Offset	0x0000 06D4		
Physical Address	0x4844 A6D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat5_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1774. Register Call Summary for Register CFG_MMC3_DAT5_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1775. CFG_MMC3_DAT6_IN

Address Offset	0x0000 06D8		
Physical Address	0x4844 A6D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat6_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1776. Register Call Summary for Register CFG_MMC3_DAT6_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1777. CFG_MMC3_DAT6_OEN

Address Offset	0x0000 06DC		
Physical Address	0x4844 A6DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat6_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1778. Register Call Summary for Register CFG_MMC3_DAT6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1779. CFG_MMC3_DAT6_OUT

Address Offset	0x0000 06E0		
Physical Address	0x4844 A6E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat6_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1780. Register Call Summary for Register CFG_MMC3_DAT6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1781. CFG_MMC3_DAT7_IN

Address Offset	0x0000 06E4		
Physical Address	0x4844 A6E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat7_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1782. Register Call Summary for Register CFG_MMC3_DAT7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1783. CFG_MMC3_DAT7_OEN

Address Offset	0x0000 06E8		
Physical Address	0x4844 A6E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat7_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1784. Register Call Summary for Register CFG_MMC3_DAT7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1785. CFG_MMC3_DAT7_OUT

Address Offset	0x0000 06EC		
Physical Address	0x4844 A6EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_mmc3_dat7_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1786. Register Call Summary for Register CFG_MMC3_DAT7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1787. CFG_RGMII0_RXC_IN

Address Offset	0x0000 06F0		
Physical Address	0x4844 A6F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxc_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1788. Register Call Summary for Register CFG_RGMII0_RXC_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1789. CFG_RGMII0_RXC_OEN

Address Offset	0x0000 06F4		
Physical Address	0x4844 A6F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxc_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1790. Register Call Summary for Register CFG_RGMII0_RXC_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1791. CFG_RGMII0_RXC_OUT

Address Offset	0x0000 06F8		
Physical Address	0x4844 A6F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxc_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1792. Register Call Summary for Register CFG_RGMII0_RXC_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1793. CFG_RGMII0_RXCTL_IN

Address Offset	0x0000 06FC		
Physical Address	0x4844 A6FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxctl_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1794. Register Call Summary for Register CFG_RGMII0_RXCTL_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1795. CFG_RGMII0_RXCTL_OEN

Address Offset	0x0000 0700		
Physical Address	0x4844 A700	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxctl_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1796. Register Call Summary for Register CFG_RGMII0_RXCTL_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1797. CFG_RGMII0_RXCTL_OUT

Address Offset	0x0000 0704		
Physical Address	0x4844 A704	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxctl_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1798. Register Call Summary for Register CFG_RGMII0_RXCTL_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1799. CFG_RGMII0_RXD0_IN

Address Offset	0x0000 0708		
Physical Address	0x4844 A708	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1800. Register Call Summary for Register CFG_RGMII0_RXD0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1801. CFG_RGMII0_RXD0_OEN

Address Offset	0x0000 070C		
Physical Address	0x4844 A70C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1802. Register Call Summary for Register CFG_RGMII0_RXD0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1803. CFG_RGMII0_RXD0_OUT

Address Offset	0x0000 0710		
Physical Address	0x4844 A710	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1804. Register Call Summary for Register CFG_RGMII0_RXD0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1805. CFG_RGMII0_RXD1_IN

Address Offset	0x0000 0714		
Physical Address	0x4844 A714	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1806. Register Call Summary for Register CFG_RGMII0_RXD1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1807. CFG_RGMII0_RXD1_OEN

Address Offset	0x0000 0718		
Physical Address	0x4844 A718	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1808. Register Call Summary for Register CFG_RGMII0_RXD1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1809. CFG_RGMII0_RXD1_OUT

Address Offset	0x0000 071C		
Physical Address	0x4844 A71C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1810. Register Call Summary for Register CFG_RGMII0_RXD1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1811. CFG_RGMII0_RXD2_IN

Address Offset	0x0000 0720		
Physical Address	0x4844 A720	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1812. Register Call Summary for Register CFG_RGMII0_RXD2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1813. CFG_RGMII0_RXD2_OEN

Address Offset	0x0000 0724		
Physical Address	0x4844 A724	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1814. Register Call Summary for Register CFG_RGMII0_RXD2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1815. CFG_RGMII0_RXD2_OUT

Address Offset	0x0000 0728		
Physical Address	0x4844 A728	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1816. Register Call Summary for Register CFG_RGMII0_RXD2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1817. CFG_RGMII0_RXD3_IN

Address Offset	0x0000 072C		
Physical Address	0x4844 A72C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1818. Register Call Summary for Register CFG_RGMII0_RXD3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1819. CFG_RGMII0_RXD3_OEN

Address Offset	0x0000 0730		
Physical Address	0x4844 A730	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1820. Register Call Summary for Register CFG_RGMII0_RXD3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1821. CFG_RGMII0_RXD3_OUT

Address Offset	0x0000 0734		
Physical Address	0x4844 A734	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_rxd3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1822. Register Call Summary for Register CFG_RGMII0_RXD3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1823. CFG_RGMII0_TXC_IN

Address Offset	0x0000 0738		
Physical Address	0x4844 A738	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txc_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1824. Register Call Summary for Register CFG_RGMII0_TXC_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1825. CFG_RGMII0_TXC_OEN

Address Offset	0x0000 073C		
Physical Address	0x4844 A73C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txc_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1826. Register Call Summary for Register CFG_RGMII0_TXC_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1827. CFG_RGMII0_TXC_OUT

Address Offset	0x0000 0740		
Physical Address	0x4844 A740	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txc_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1828. Register Call Summary for Register CFG_RGMII0_TXC_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1829. CFG_RGMII0_TXCTL_IN

Address Offset	0x0000 0744		
Physical Address	0x4844 A744	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txctl_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1830. Register Call Summary for Register CFG_RGMII0_TXCTL_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1831. CFG_RGMII0_TXCTL_OEN

Address Offset	0x0000 0748		
Physical Address	0x4844 A748	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txctl_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1832. Register Call Summary for Register CFG_RGMII0_TXCTL_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1833. CFG_RGMII0_TXCTL_OUT

Address Offset	0x0000 074C		
Physical Address	0x4844 A74C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txctl_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1834. Register Call Summary for Register CFG_RGMII0_TXCTL_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1835. CFG_RGMII0_TXD0_IN

Address Offset	0x0000 0750		
Physical Address	0x4844 A750	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1836. Register Call Summary for Register CFG_RGMII0_TXD0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1837. CFG_RGMII0_TXD0_OEN

Address Offset	0x0000 0754		
Physical Address	0x4844 A754	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1838. Register Call Summary for Register CFG_RGMII0_TXD0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1839. CFG_RGMII0_TXD0_OUT

Address Offset	0x0000 0758		
Physical Address	0x4844 A758	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1840. Register Call Summary for Register CFG_RGMII0_TXD0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1841. CFG_RGMII0_TXD1_IN

Address Offset	0x0000 075C		
Physical Address	0x4844 A75C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1842. Register Call Summary for Register CFG_RGMII0_TXD1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1843. CFG_RGMII0_TXD1_OEN

Address Offset	0x0000 0760		
Physical Address	0x4844 A760	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1844. Register Call Summary for Register CFG_RGMII0_TXD1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1845. CFG_RGMII0_TXD1_OUT

Address Offset	0x0000 0764		
Physical Address	0x4844 A764	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd1_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1846. Register Call Summary for Register CFG_RGMII0_TXD1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1847. CFG_RGMII0_TXD2_IN

Address Offset	0x0000 0768		
Physical Address	0x4844 A768	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd2_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1848. Register Call Summary for Register CFG_RGMII0_TXD2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1849. CFG_RGMII0_TXD2_OEN

Address Offset	0x0000 076C		
Physical Address	0x4844 A76C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd2_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td colspan="1">RESERVED</td><td colspan="1">LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1850. Register Call Summary for Register CFG_RGMII0_TXD2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1851. CFG_RGMII0_TXD2_OUT

Address Offset	0x0000 0770		
Physical Address	0x4844 A770	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd2_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td colspan="1">RESERVED</td><td colspan="1">LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1852. Register Call Summary for Register CFG_RGMII0_TXD2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1853. CFG_RGMII0_TXD3_IN

Address Offset	0x0000 0774		
Physical Address	0x4844 A774	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd3_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1854. Register Call Summary for Register CFG_RGMII0_TXD3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1855. CFG_RGMII0_TXD3_OEN

Address Offset	0x0000 0778		
Physical Address	0x4844 A778	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd3_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1856. Register Call Summary for Register CFG_RGMII0_TXD3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1857. CFG_RGMII0_TXD3_OUT

Address Offset	0x0000 077C		
Physical Address	0x4844 A77C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rgmii0_txd3_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1858. Register Call Summary for Register CFG_RGMII0_TXD3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1859. CFG_RTCK_IN

Address Offset	0x0000 0780		
Physical Address	0x4844 A780	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rtck_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1860. Register Call Summary for Register CFG_RTCK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1861. CFG_RTCK_OEN

Address Offset	0x0000 0784		
Physical Address	0x4844 A784	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rtck_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1862. Register Call Summary for Register CFG_RTCK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1863. CFG_RTCK_OUT

Address Offset	0x0000 0788		
Physical Address	0x4844 A788	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_rtck_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1864. Register Call Summary for Register CFG_RTCK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1865. CFG_SPI1_CS0_IN

Address Offset	0x0000 078C		
Physical Address	0x4844 A78C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1866. Register Call Summary for Register CFG_SPI1_CS0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1867. CFG_SPI1_CS0_OEN

Address Offset	0x0000 0790		
Physical Address	0x4844 A790	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1868. Register Call Summary for Register CFG_SPI1_CS0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1869. CFG_SPI1_CS0_OUT

Address Offset	0x0000 0794		
Physical Address	0x4844 A794	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1870. Register Call Summary for Register CFG_SPI1_CS0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1871. CFG_SPI1_CS1_IN

Address Offset	0x0000 0798		
Physical Address	0x4844 A798	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1872. Register Call Summary for Register CFG_SPI1_CS1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1873. CFG_SPI1_CS1_OEN

Address Offset	0x0000 079C		
Physical Address	0x4844 A79C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs1_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1874. Register Call Summary for Register CFG_SPI1_CS1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1875. CFG_SPI1_CS1_OUT

Address Offset	0x0000 07A0		
Physical Address	0x4844 A7A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs1_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1876. Register Call Summary for Register CFG_SPI1_CS1_OUT

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1877. CFG_SPI1_CS2_IN

Address Offset	0x0000 07A4		
Physical Address	0x4844 A7A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs2_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RE
SE
RV
ED</td><td>LO
CK
_B
IT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1878. Register Call Summary for Register CFG_SPI1_CS2_IN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1879. CFG_SPI1_CS2_OEN

Address Offset	0x0000 07A8		
Physical Address	0x4844 A7A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs2_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RE
SE
RV
ED</td><td>LO
CK
_B
IT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1880. Register Call Summary for Register CFG_SPI1_CS2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1881. CFG_SPI1_CS2_OUT

Address Offset	0x0000 07AC		
Physical Address	0x4844 A7AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs2_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1882. Register Call Summary for Register CFG_SPI1_CS2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1883. CFG_SPI1_CS3_IN

Address Offset	0x0000 07B0		
Physical Address	0x4844 A7B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs3_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1884. Register Call Summary for Register CFG_SPI1_CS3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1885. CFG_SPI1_CS3_OEN

Address Offset	0x0000 07B4		
Physical Address	0x4844 A7B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1886. Register Call Summary for Register CFG_SPI1_CS3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1887. CFG_SPI1_CS3_OUT

Address Offset	0x0000 07B8		
Physical Address	0x4844 A7B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_cs3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1888. Register Call Summary for Register CFG_SPI1_CS3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1889. CFG_SPI1_D0_IN

Address Offset	0x0000 07BC		
Physical Address	0x4844 A7BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_d0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1890. Register Call Summary for Register CFG_SPI1_D0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1891. CFG_SPI1_D0_OEN

Address Offset	0x0000 07C0		
Physical Address	0x4844 A7C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_d0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1892. Register Call Summary for Register CFG_SPI1_D0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1893. CFG_SPI1_D0_OUT

Address Offset	0x0000 07C4		
Physical Address	0x4844 A7C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_d0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1894. Register Call Summary for Register CFG_SPI1_D0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1895. CFG_SPI1_D1_IN

Address Offset	0x0000 07C8		
Physical Address	0x4844 A7C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_d1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1896. Register Call Summary for Register CFG_SPI1_D1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1897. CFG_SPI1_D1_OEN

Address Offset	0x0000 07CC		
Physical Address	0x4844 A7CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_d1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1898. Register Call Summary for Register CFG_SPI1_D1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1899. CFG_SPI1_D1_OUT

Address Offset	0x0000 07D0		
Physical Address	0x4844 A7D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_d1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1900. Register Call Summary for Register CFG_SPI1_D1_OUT

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1901. CFG_SPI1_SCLK_IN

Address Offset	0x0000 07D4		
Physical Address	0x4844 A7D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_sclk_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1902. Register Call Summary for Register CFG_SPI1_SCLK_IN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1903. CFG_SPI1_SCLK_OEN

Address Offset	0x0000 07D8		
Physical Address	0x4844 A7D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_sclk_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1904. Register Call Summary for Register CFG_SPI1_SCLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1905. CFG_SPI1_SCLK_OUT

Address Offset	0x0000 07DC		
Physical Address	0x4844 A7DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi1_sclk_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1906. Register Call Summary for Register CFG_SPI1_SCLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1907. CFG_SPI2_CS0_IN

Address Offset	0x0000 07E0		
Physical Address	0x4844 A7E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_cs0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1908. Register Call Summary for Register CFG_SPI2_CS0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1909. CFG_SPI2_CS0_OEN

Address Offset	0x0000 07E4		
Physical Address	0x4844 A7E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_cs0_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1910. Register Call Summary for Register CFG_SPI2_CS0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1911. CFG_SPI2_CS0_OUT

Address Offset	0x0000 07E8		
Physical Address	0x4844 A7E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_cs0_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1912. Register Call Summary for Register CFG_SPI2_CS0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1913. CFG_SPI2_D0_IN

Address Offset	0x0000 07EC		
Physical Address	0x4844 A7EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_d0_in interface		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18	17 16 15 14 13 12	11	10	9 8 7 6 5 4 3 2 1 0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1914. Register Call Summary for Register CFG_SPI2_D0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1915. CFG_SPI2_D0_OEN

Address Offset	0x0000 07F0		
Physical Address	0x4844 A7F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_d0_oen interface		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18	17 16 15 14 13 12	11	10	9 8 7 6 5 4 3 2 1 0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1916. Register Call Summary for Register CFG_SPI2_D0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1917. CFG_SPI2_D0_OUT

Address Offset	0x0000 07F4		
Physical Address	0x4844 A7F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_d0_out interface		
Type	RW		

Bits 31–18	Bits 17–12	Bit 11	Bit 10	Bits 9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1918. Register Call Summary for Register CFG_SPI2_D0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1919. CFG_SPI2_D1_IN

Address Offset	0x0000 07F8		
Physical Address	0x4844 A7F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_d1_in interface		
Type	RW		

Bits 31–18	Bits 17–12	Bit 11	Bit 10	Bits 9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1920. Register Call Summary for Register CFG_SPI2_D1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1921. CFG_SPI2_D1_OEN

Address Offset	0x0000 07FC		
Physical Address	0x4844 A7FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_d1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1922. Register Call Summary for Register CFG_SPI2_D1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1923. CFG_SPI2_D1_OUT

Address Offset	0x0000 0800		
Physical Address	0x4844 A800	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_d1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1924. Register Call Summary for Register CFG_SPI2_D1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1925. CFG_SPI2_SCLK_IN

Address Offset	0x0000 0804		
Physical Address	0x4844 A804	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_sclk_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1926. Register Call Summary for Register CFG_SPI2_SCLK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1927. CFG_SPI2_SCLK_OEN

Address Offset	0x0000 0808		
Physical Address	0x4844 A808	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_sclk_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1928. Register Call Summary for Register CFG_SPI2_SCLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1929. CFG_SPI2_SCLK_OUT

Address Offset	0x0000 080C		
Physical Address	0x4844 A80C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_spi2_sclk_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1930. Register Call Summary for Register CFG_SPI2_SCLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1931. CFG_TDI_IN

Address Offset	0x0000 0810		
Physical Address	0x4844 A810	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tdi_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1932. Register Call Summary for Register CFG_TDI_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1933. CFG_TDI_OEN

Address Offset	0x0000 0814		
Physical Address	0x4844 A814	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tdi_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1934. Register Call Summary for Register CFG_TDI_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1935. CFG_TDI_OUT

Address Offset	0x0000 0818		
Physical Address	0x4844 A818	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tdi_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1936. Register Call Summary for Register CFG_TDI_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1937. CFG_TDO_IN

Address Offset	0x0000 081C		
Physical Address	0x4844 A81C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tdo_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1938. Register Call Summary for Register CFG_TDO_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1939. CFG_TDO_OEN

Address Offset	0x0000 0820		
Physical Address	0x4844 A820	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tdo_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1940. Register Call Summary for Register CFG_TDO_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1941. CFG_TDO_OUT

Address Offset	0x0000 0824		
Physical Address	0x4844 A824	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tdo_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1942. Register Call Summary for Register CFG_TDO_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1943. CFG_TMS_IN

Address Offset	0x0000 0828		
Physical Address	0x4844 A828	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tms_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1944. Register Call Summary for Register CFG_TMS_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1945. CFG_TMS_OEN

Address Offset	0x0000 082C		
Physical Address	0x4844 A82C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tms_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1946. Register Call Summary for Register CFG_TMS_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1947. CFG_TMS_OUT

Address Offset	0x0000 0830		
Physical Address	0x4844 A830	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_tms_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1948. Register Call Summary for Register CFG_TMS_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1949. CFG_TRSTN_IN

Address Offset	0x0000 0834		
Physical Address	0x4844 A834	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_trstn_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1950. Register Call Summary for Register CFG_TRSTN_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1951. CFG_TRSTN_OEN

Address Offset	0x0000 0838		
Physical Address	0x4844 A838	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_trstn_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1952. Register Call Summary for Register CFG_TRSTN_OEN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1953. CFG_TRSTN_OUT

Address Offset	0x0000 083C		
Physical Address	0x4844 A83C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_trstn_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1954. Register Call Summary for Register CFG_TRSTN_OUT

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1955. CFG_UART1_CTSN_IN

Address Offset	0x0000 0840		
Physical Address	0x4844 A840	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_ctsn_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1956. Register Call Summary for Register CFG_UART1_CTSN_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1957. CFG_UART1_CTSN_OEN

Address Offset	0x0000 0844		
Physical Address	0x4844 A844	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_ctsn_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1958. Register Call Summary for Register CFG_UART1_CTSN_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1959. CFG_UART1_CTSN_OUT

Address Offset	0x0000 0848		
Physical Address	0x4844 A848	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_ctsn_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1960. Register Call Summary for Register CFG_UART1_CTSN_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1961. CFG_UART1_RTSN_IN

Address Offset	0x0000 084C		
Physical Address	0x4844 A84C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_rtsn_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1962. Register Call Summary for Register CFG_UART1_RTSN_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1963. CFG_UART1_RTSN_OEN

Address Offset	0x0000 0850		
Physical Address	0x4844 A850	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_rtsn_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1964. Register Call Summary for Register CFG_UART1_RTSN_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1965. CFG_UART1_RTSN_OUT

Address Offset	0x0000 0854		
Physical Address	0x4844 A854	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_rtsn_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1966. Register Call Summary for Register CFG_UART1_RTSN_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1967. CFG_UART1_RXD_IN

Address Offset	0x0000 0858		
Physical Address	0x4844 A858	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_rxd_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1968. Register Call Summary for Register CFG_UART1_RXD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1969. CFG_UART1_RXD_OEN

Address Offset	0x0000 085C		
Physical Address	0x4844 A85C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_rxd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1970. Register Call Summary for Register CFG_UART1_RXD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1971. CFG_UART1_RXD_OUT

Address Offset	0x0000 0860		
Physical Address	0x4844 A860	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_rxd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1972. Register Call Summary for Register CFG_UART1_RXD_OUT

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1973. CFG_UART1_TXD_IN

Address Offset	0x0000 0864		
Physical Address	0x4844 A864	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_txd_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1974. Register Call Summary for Register CFG_UART1_TXD_IN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-1975. CFG_UART1_TXD_OEN

Address Offset	0x0000 0868		
Physical Address	0x4844 A868	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_txd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1976. Register Call Summary for Register CFG_UART1_TXD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1977. CFG_UART1_TXD_OUT

Address Offset	0x0000 086C		
Physical Address	0x4844 A86C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart1_txd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1978. Register Call Summary for Register CFG_UART1_TXD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1979. CFG_UART2_CTSN_IN

Address Offset	0x0000 0870		
Physical Address	0x4844 A870	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_ctsn_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1980. Register Call Summary for Register CFG_UART2_CTSN_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1981. CFG_UART2_CTSN_OEN

Address Offset	0x0000 0874		
Physical Address	0x4844 A874	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_ctsn_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1982. Register Call Summary for Register CFG_UART2_CTSN_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1983. CFG_UART2_CTSN_OUT

Address Offset	0x0000 0878		
Physical Address	0x4844 A878	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_ctsn_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1984. Register Call Summary for Register CFG_UART2_CTSN_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1985. CFG_UART2_RTSN_IN

Address Offset	0x0000 087C		
Physical Address	0x4844 A87C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_rtsn_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1986. Register Call Summary for Register CFG_UART2_RTSN_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1987. CFG_UART2_RTSN_OEN

Address Offset	0x0000 0880		
Physical Address	0x4844 A880	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_rtsn_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1988. Register Call Summary for Register CFG_UART2_RTSN_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1989. CFG_UART2_RTSN_OUT

Address Offset	0x0000 0884		
Physical Address	0x4844 A884	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_rtsn_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1990. Register Call Summary for Register CFG_UART2_RTSN_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1991. CFG_UART2_RXD_IN

Address Offset	0x0000 0888		
Physical Address	0x4844 A888	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_rxd_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1992. Register Call Summary for Register CFG_UART2_RXD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1993. CFG_UART2_RXD_OEN

Address Offset	0x0000 088C		
Physical Address	0x4844 A88C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_rxd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1994. Register Call Summary for Register CFG_UART2_RXD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1995. CFG_UART2_RXD_OUT

Address Offset	0x0000 0890		
Physical Address	0x4844 A890	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_rxd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1996. Register Call Summary for Register CFG_UART2_RXD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1997. CFG_UART2_TXD_IN

Address Offset	0x0000 0894		
Physical Address	0x4844 A894	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_txd_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-1998. Register Call Summary for Register CFG_UART2_TXD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-1999. CFG_UART2_TXD_OEN

Address Offset	0x0000 0898		
Physical Address	0x4844 A898	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_txd_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2000. Register Call Summary for Register CFG_UART2_TXD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2001. CFG_UART2_TXD_OUT

Address Offset	0x0000 089C		
Physical Address	0x4844 A89C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart2_txd_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2002. Register Call Summary for Register CFG_UART2_TXD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2003. CFG_UART3_RXD_IN

Address Offset	0x0000 08A0		
Physical Address	0x4844 A8A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart3_rxd_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2004. Register Call Summary for Register CFG_UART3_RXD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2005. CFG_UART3_RXD_OEN

Address Offset	0x0000 08A4		
Physical Address	0x4844 A8A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart3_rxd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2006. Register Call Summary for Register CFG_UART3_RXD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2007. CFG_UART3_RXD_OUT

Address Offset	0x0000 08A8		
Physical Address	0x4844 A8A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart3_rxd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2008. Register Call Summary for Register CFG_UART3_RXD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2009. CFG_UART3_TXD_IN

Address Offset	0x0000 08AC		
Physical Address	0x4844 A8AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart3_txd_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2010. Register Call Summary for Register CFG_UART3_TXD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2011. CFG_UART3_TXD_OEN

Address Offset	0x0000 08B0		
Physical Address	0x4844 A8B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart3_txd_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2012. Register Call Summary for Register CFG_UART3_TXD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2013. CFG_UART3_TXD_OUT

Address Offset	0x0000 08B4		
Physical Address	0x4844 A8B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_uart3_txd_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2014. Register Call Summary for Register CFG_UART3_TXD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2015. CFG_USB1_DRVVBUS_IN

Address Offset	0x0000 08B8		
Physical Address	0x4844 A8B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_usb1_drvvbus_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2016. Register Call Summary for Register CFG_USB1_DRVVBUS_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2017. CFG_USB1_DRVVBUS_OEN

Address Offset	0x0000 08BC		
Physical Address	0x4844 A8BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_usb1_drvvbus_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2018. Register Call Summary for Register CFG_USB1_DRVVBUS_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2019. CFG_USB1_DRVVBUS_OUT

Address Offset	0x0000 08C0		
Physical Address	0x4844 A8C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_usb1_drvvbus_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2020. Register Call Summary for Register CFG_USB1_DRVVBUS_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2021. CFG_USB2_DRVVBUS_IN

Address Offset	0x0000 08C4		
Physical Address	0x4844 A8C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_usb2_drvvbus_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2022. Register Call Summary for Register CFG_USB2_DRVVBUS_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2023. CFG_USB2_DRVVBUS_OEN

Address Offset	0x0000 08C8		
Physical Address	0x4844 A8C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_usb2_drvvbus_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2024. Register Call Summary for Register CFG_USB2_DRVVBUS_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2025. CFG_USB2_DRVVBUS_OUT

Address Offset	0x0000 08CC		
Physical Address	0x4844 A8CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_usb2_drvvbus_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2026. Register Call Summary for Register CFG_USB2_DRVVBUS_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2027. CFG_VIN1A_CLK0_IN

Address Offset	0x0000 08D0		
Physical Address	0x4844 A8D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_clk0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2028. Register Call Summary for Register CFG_VIN1A_CLK0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2029. CFG_VIN1A_CLK0_OEN

Address Offset	0x0000 08D4		
Physical Address	0x4844 A8D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_clk0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2030. Register Call Summary for Register CFG_VIN1A_CLK0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2031. CFG_VIN1A_CLK0_OUT

Address Offset	0x0000 08D8		
Physical Address	0x4844 A8D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_clk0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2032. Register Call Summary for Register CFG_VIN1A_CLK0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2033. CFG_VIN1A_D0_IN

Address Offset	0x0000 08DC		
Physical Address	0x4844 A8DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2034. Register Call Summary for Register CFG_VIN1A_D0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2035. CFG_VIN1A_D0_OEN

Address Offset	0x0000 08E0		
Physical Address	0x4844 A8E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2036. Register Call Summary for Register CFG_VIN1A_D0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2037. CFG_VIN1A_D0_OUT

Address Offset	0x0000 08E4		
Physical Address	0x4844 A8E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d0_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2038. Register Call Summary for Register CFG_VIN1A_D0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2039. CFG_VIN1A_D10_IN

Address Offset	0x0000 08E8		
Physical Address	0x4844 A8E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d10_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2040. Register Call Summary for Register CFG_VIN1A_D10_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2041. CFG_VIN1A_D10_OEN

Address Offset	0x0000 08EC		
Physical Address	0x4844 A8EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d10_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2042. Register Call Summary for Register CFG_VIN1A_D10_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2043. CFG_VIN1A_D10_OUT

Address Offset	0x0000 08F0		
Physical Address	0x4844 A8F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d10_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2044. Register Call Summary for Register CFG_VIN1A_D10_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2045. CFG_VIN1A_D11_IN

Address Offset	0x0000 08F4		
Physical Address	0x4844 A8F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d11_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2046. Register Call Summary for Register CFG_VIN1A_D11_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2047. CFG_VIN1A_D11_OEN

Address Offset	0x0000 08F8		
Physical Address	0x4844 A8F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d11_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2048. Register Call Summary for Register CFG_VIN1A_D11_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2049. CFG_VIN1A_D11_OUT

Address Offset	0x0000 08FC		
Physical Address	0x4844 A8FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d11_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2050. Register Call Summary for Register CFG_VIN1A_D11_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2051. CFG_VIN1A_D12_IN

Address Offset	0x0000 0900		
Physical Address	0x4844 A900	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d12_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2052. Register Call Summary for Register CFG_VIN1A_D12_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2053. CFG_VIN1A_D12_OEN

Address Offset	0x0000 0904		
Physical Address	0x4844 A904	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d12_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2054. Register Call Summary for Register CFG_VIN1A_D12_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2055. CFG_VIN1A_D12_OUT

Address Offset	0x0000 0908		
Physical Address	0x4844 A908	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d12_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2056. Register Call Summary for Register CFG_VIN1A_D12_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2057. CFG_VIN1A_D13_IN

Address Offset	0x0000 090C		
Physical Address	0x4844 A90C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d13_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2058. Register Call Summary for Register CFG_VIN1A_D13_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2059. CFG_VIN1A_D13_OEN

Address Offset	0x0000 0910		
Physical Address	0x4844 A910	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d13_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2060. Register Call Summary for Register CFG_VIN1A_D13_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2061. CFG_VIN1A_D13_OUT

Address Offset	0x0000 0914		
Physical Address	0x4844 A914	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d13_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2062. Register Call Summary for Register CFG_VIN1A_D13_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2063. CFG_VIN1A_D14_IN

Address Offset	0x0000 0918		
Physical Address	0x4844 A918	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d14_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2064. Register Call Summary for Register CFG_VIN1A_D14_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2065. CFG_VIN1A_D14_OEN

Address Offset	0x0000 091C		
Physical Address	0x4844 A91C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d14_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2066. Register Call Summary for Register CFG_VIN1A_D14_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2067. CFG_VIN1A_D14_OUT

Address Offset	0x0000 0920		
Physical Address	0x4844 A920	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d14_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2068. Register Call Summary for Register CFG_VIN1A_D14_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2069. CFG_VIN1A_D15_IN

Address Offset	0x0000 0924		
Physical Address	0x4844 A924	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d15_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2070. Register Call Summary for Register CFG_VIN1A_D15_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2071. CFG_VIN1A_D15_OEN

Address Offset	0x0000 0928		
Physical Address	0x4844 A928	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d15_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2072. Register Call Summary for Register CFG_VIN1A_D15_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2073. CFG_VIN1A_D15_OUT

Address Offset	0x0000 092C		
Physical Address	0x4844 A92C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d15_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2074. Register Call Summary for Register CFG_VIN1A_D15_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2075. CFG_VIN1A_D16_IN

Address Offset	0x0000 0930		
Physical Address	0x4844 A930	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d16_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2076. Register Call Summary for Register CFG_VIN1A_D16_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2077. CFG_VIN1A_D16_OEN

Address Offset	0x0000 0934		
Physical Address	0x4844 A934	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d16_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2078. Register Call Summary for Register CFG_VIN1A_D16_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2079. CFG_VIN1A_D16_OUT

Address Offset	0x0000 0938		
Physical Address	0x4844 A938	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d16_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2080. Register Call Summary for Register CFG_VIN1A_D16_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2081. CFG_VIN1A_D17_IN

Address Offset	0x0000 093C		
Physical Address	0x4844 A93C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d17_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2082. Register Call Summary for Register CFG_VIN1A_D17_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2083. CFG_VIN1A_D17_OEN

Address Offset	0x0000 0940		
Physical Address	0x4844 A940	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d17_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2084. Register Call Summary for Register CFG_VIN1A_D17_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2085. CFG_VIN1A_D17_OUT

Address Offset	0x0000 0944		
Physical Address	0x4844 A944	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d17_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2086. Register Call Summary for Register CFG_VIN1A_D17_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2087. CFG_VIN1A_D18_IN

Address Offset	0x0000 0948		
Physical Address	0x4844 A948	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d18_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2088. Register Call Summary for Register CFG_VIN1A_D18_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2089. CFG_VIN1A_D18_OEN

Address Offset	0x0000 094C		
Physical Address	0x4844 A94C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d18_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2090. Register Call Summary for Register CFG_VIN1A_D18_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2091. CFG_VIN1A_D18_OUT

Address Offset	0x0000 0950		
Physical Address	0x4844 A950	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d18_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2092. Register Call Summary for Register CFG_VIN1A_D18_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2093. CFG_VIN1A_D19_IN

Address Offset	0x0000 0954		
Physical Address	0x4844 A954	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d19_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2094. Register Call Summary for Register CFG_VIN1A_D19_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2095. CFG_VIN1A_D19_OEN

Address Offset	0x0000 0958		
Physical Address	0x4844 A958	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d19_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2096. Register Call Summary for Register CFG_VIN1A_D19_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2097. CFG_VIN1A_D19_OUT

Address Offset	0x0000 095C		
Physical Address	0x4844 A95C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d19_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2098. Register Call Summary for Register CFG_VIN1A_D19_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2099. CFG_VIN1A_D1_IN

Address Offset	0x0000 0960		
Physical Address	0x4844 A960	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2100. Register Call Summary for Register CFG_VIN1A_D1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2101. CFG_VIN1A_D1_OEN

Address Offset	0x0000 0964		
Physical Address	0x4844 A964	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d1_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2102. Register Call Summary for Register CFG_VIN1A_D1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2103. CFG_VIN1A_D1_OUT

Address Offset	0x0000 0968		
Physical Address	0x4844 A968	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d1_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2104. Register Call Summary for Register CFG_VIN1A_D1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2105. CFG_VIN1A_D20_IN

Address Offset	0x0000 096C		
Physical Address	0x4844 A96C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d20_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2106. Register Call Summary for Register CFG_VIN1A_D20_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2107. CFG_VIN1A_D20_OEN

Address Offset	0x0000 0970		
Physical Address	0x4844 A970	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d20_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2108. Register Call Summary for Register CFG_VIN1A_D20_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2109. CFG_VIN1A_D20_OUT

Address Offset	0x0000 0974		
Physical Address	0x4844 A974	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d20_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2110. Register Call Summary for Register CFG_VIN1A_D20_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2111. CFG_VIN1A_D21_IN

Address Offset	0x0000 0978		
Physical Address	0x4844 A978	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d21_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2112. Register Call Summary for Register CFG_VIN1A_D21_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2113. CFG_VIN1A_D21_OEN

Address Offset	0x0000 097C		
Physical Address	0x4844 A97C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d21_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2114. Register Call Summary for Register CFG_VIN1A_D21_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2115. CFG_VIN1A_D21_OUT

Address Offset	0x0000 0980		
Physical Address	0x4844 A980	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d21_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2116. Register Call Summary for Register CFG_VIN1A_D21_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2117. CFG_VIN1A_D22_IN

Address Offset	0x0000 0984		
Physical Address	0x4844 A984	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d22_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2118. Register Call Summary for Register CFG_VIN1A_D22_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2119. CFG_VIN1A_D22_OEN

Address Offset	0x0000 0988		
Physical Address	0x4844 A988	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d22_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2120. Register Call Summary for Register CFG_VIN1A_D22_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2121. CFG_VIN1A_D22_OUT

Address Offset	0x0000 098C		
Physical Address	0x4844 A98C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d22_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2122. Register Call Summary for Register CFG_VIN1A_D22_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2123. CFG_VIN1A_D23_IN

Address Offset	0x0000 0990		
Physical Address	0x4844 A990	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d23_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2124. Register Call Summary for Register CFG_VIN1A_D23_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2125. CFG_VIN1A_D23_OEN

Address Offset	0x0000 0994		
Physical Address	0x4844 A994	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d23_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2126. Register Call Summary for Register CFG_VIN1A_D23_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2127. CFG_VIN1A_D23_OUT

Address Offset	0x0000 0998		
Physical Address	0x4844 A998	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d23_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2128. Register Call Summary for Register CFG_VIN1A_D23_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2129. CFG_VIN1A_D2_IN

Address Offset	0x0000 099C		
Physical Address	0x4844 A99C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2130. Register Call Summary for Register CFG_VIN1A_D2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2131. CFG_VIN1A_D2_OEN

Address Offset	0x0000 09A0		
Physical Address	0x4844 A9A0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2132. Register Call Summary for Register CFG_VIN1A_D2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2133. CFG_VIN1A_D2_OUT

Address Offset	0x0000 09A4		
Physical Address	0x4844 A9A4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2134. Register Call Summary for Register CFG_VIN1A_D2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2135. CFG_VIN1A_D3_IN

Address Offset	0x0000 09A8		
Physical Address	0x4844 A9A8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2136. Register Call Summary for Register CFG_VIN1A_D3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2137. CFG_VIN1A_D3_OEN

Address Offset	0x0000 09AC		
Physical Address	0x4844 A9AC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d3_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2138. Register Call Summary for Register CFG_VIN1A_D3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2139. CFG_VIN1A_D3_OUT

Address Offset	0x0000 09B0		
Physical Address	0x4844 A9B0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2140. Register Call Summary for Register CFG_VIN1A_D3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2141. CFG_VIN1A_D4_IN

Address Offset	0x0000 09B4		
Physical Address	0x4844 A9B4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d4_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2142. Register Call Summary for Register CFG_VIN1A_D4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2143. CFG_VIN1A_D4_OEN

Address Offset	0x0000 09B8		
Physical Address	0x4844 A9B8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d4_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2144. Register Call Summary for Register CFG_VIN1A_D4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2145. CFG_VIN1A_D4_OUT

Address Offset	0x0000 09BC		
Physical Address	0x4844 A9BC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d4_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2146. Register Call Summary for Register CFG_VIN1A_D4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2147. CFG_VIN1A_D5_IN

Address Offset	0x0000 09C0		
Physical Address	0x4844 A9C0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d5_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2148. Register Call Summary for Register CFG_VIN1A_D5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2149. CFG_VIN1A_D5_OEN

Address Offset	0x0000 09C4		
Physical Address	0x4844 A9C4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d5_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2150. Register Call Summary for Register CFG_VIN1A_D5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2151. CFG_VIN1A_D5_OUT

Address Offset	0x0000 09C8		
Physical Address	0x4844 A9C8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d5_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2152. Register Call Summary for Register CFG_VIN1A_D5_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2153. CFG_VIN1A_D6_IN

Address Offset	0x0000 09CC		
Physical Address	0x4844 A9CC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d6_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2154. Register Call Summary for Register CFG_VIN1A_D6_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2155. CFG_VIN1A_D6_OEN

Address Offset	0x0000 09D0		
Physical Address	0x4844 A9D0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d6_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2156. Register Call Summary for Register CFG_VIN1A_D6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2157. CFG_VIN1A_D6_OUT

Address Offset	0x0000 09D4		
Physical Address	0x4844 A9D4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d6_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2158. Register Call Summary for Register CFG_VIN1A_D6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2159. CFG_VIN1A_D7_IN

Address Offset	0x0000 09D8		
Physical Address	0x4844 A9D8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d7_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2160. Register Call Summary for Register CFG_VIN1A_D7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2161. CFG_VIN1A_D7_OEN

Address Offset	0x0000 09DC		
Physical Address	0x4844 A9DC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d7_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2162. Register Call Summary for Register CFG_VIN1A_D7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2163. CFG_VIN1A_D7_OUT

Address Offset	0x0000 09E0		
Physical Address	0x4844 A9E0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d7_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2164. Register Call Summary for Register CFG_VIN1A_D7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2165. CFG_VIN1A_D8_IN

Address Offset	0x0000 09E4		
Physical Address	0x4844 A9E4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d8_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2166. Register Call Summary for Register CFG_VIN1A_D8_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2167. CFG_VIN1A_D8_OEN

Address Offset	0x0000 09E8		
Physical Address	0x4844 A9E8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d8_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2168. Register Call Summary for Register CFG_VIN1A_D8_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2169. CFG_VIN1A_D8_OUT

Address Offset	0x0000 09EC		
Physical Address	0x4844 A9EC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d8_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2170. Register Call Summary for Register CFG_VIN1A_D8_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2171. CFG_VIN1A_D9_IN

Address Offset	0x0000 09F0		
Physical Address	0x4844 A9F0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d9_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2172. Register Call Summary for Register CFG_VIN1A_D9_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2173. CFG_VIN1A_D9_OEN

Address Offset	0x0000 09F4		
Physical Address	0x4844 A9F4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d9_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2174. Register Call Summary for Register CFG_VIN1A_D9_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2175. CFG_VIN1A_D9_OUT

Address Offset	0x0000 09F8		
Physical Address	0x4844 A9F8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_d9_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2176. Register Call Summary for Register CFG_VIN1A_D9_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2177. CFG_VIN1A_DE0_IN

Address Offset	0x0000 09FC		
Physical Address	0x4844 A9FC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_de0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2178. Register Call Summary for Register CFG_VIN1A_DE0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2179. CFG_VIN1A_DE0_OEN

Address Offset	0x0000 0A00		
Physical Address	0x4844 AA00	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_de0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2180. Register Call Summary for Register CFG_VIN1A_DE0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2181. CFG_VIN1A_DE0_OUT

Address Offset	0x0000 0A04		
Physical Address	0x4844 AA04	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_de0_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2182. Register Call Summary for Register CFG_VIN1A_DE0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2183. CFG_VIN1A_FLD0_IN

Address Offset	0x0000 0A08		
Physical Address	0x4844 AA08	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_fld0_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2184. Register Call Summary for Register CFG_VIN1A_FLD0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2185. CFG_VIN1A_FLD0_OEN

Address Offset	0x0000 0A0C		
Physical Address	0x4844 AA0C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_fld0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2186. Register Call Summary for Register CFG_VIN1A_FLD0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2187. CFG_VIN1A_FLD0_OUT

Address Offset	0x0000 0A10		
Physical Address	0x4844 AA10	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_fld0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2188. Register Call Summary for Register CFG_VIN1A_FLD0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2189. CFG_VIN1A_HSYNC0_IN

Address Offset	0x0000 0A14		
Physical Address	0x4844 AA14	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_hsync0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2190. Register Call Summary for Register CFG_VIN1A_HSYNC0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2191. CFG_VIN1A_HSYNC0_OEN

Address Offset	0x0000 0A18		
Physical Address	0x4844 AA18	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_hsync0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2192. Register Call Summary for Register CFG_VIN1A_HSYNC0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2193. CFG_VIN1A_HSYNC0_OUT

Address Offset	0x0000 0A1C		
Physical Address	0x4844 AA1C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_hsync0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2194. Register Call Summary for Register CFG_VIN1A_HSYNC0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2195. CFG_VIN1A_VSYNC0_IN

Address Offset	0x0000 0A20		
Physical Address	0x4844 AA20	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_vsync0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2196. Register Call Summary for Register CFG_VIN1A_VSYNC0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2197. CFG_VIN1A_VSYNC0_OEN

Address Offset	0x0000 0A24		
Physical Address	0x4844 AA24	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_vsync0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2198. Register Call Summary for Register CFG_VIN1A_VSYNC0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2199. CFG_VIN1A_VSYNC0_OUT

Address Offset	0x0000 0A28		
Physical Address	0x4844 AA28	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1a_vsync0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2200. Register Call Summary for Register CFG_VIN1A_VSYNC0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2201. CFG_VIN1B_CLK1_IN

Address Offset	0x0000 0A2C		
Physical Address	0x4844 AA2C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1b_clk1_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2202. Register Call Summary for Register CFG_VIN1B_CLK1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2203. CFG_VIN1B_CLK1_OEN

Address Offset	0x0000 0A30		
Physical Address	0x4844 AA30	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1b_clk1_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2204. Register Call Summary for Register CFG_VIN1B_CLK1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2205. CFG_VIN1B_CLK1_OUT

Address Offset	0x0000 0A34		
Physical Address	0x4844 AA34	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin1b_clk1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2206. Register Call Summary for Register CFG_VIN1B_CLK1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2207. CFG_VIN2A_CLK0_IN

Address Offset	0x0000 0A38		
Physical Address	0x4844 AA38	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_clk0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2208. Register Call Summary for Register CFG_VIN2A_CLK0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2209. CFG_VIN2A_CLK0_OEN

Address Offset	0x0000 0A3C		
Physical Address	0x4844 AA3C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_clk0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2210. Register Call Summary for Register CFG_VIN2A_CLK0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2211. CFG_VIN2A_CLK0_OUT

Address Offset	0x0000 0A40		
Physical Address	0x4844 AA40	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_clk0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2212. Register Call Summary for Register CFG_VIN2A_CLK0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2213. CFG_VIN2A_D0_IN

Address Offset	0x0000 0A44		
Physical Address	0x4844 AA44	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d0_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2214. Register Call Summary for Register CFG_VIN2A_D0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2215. CFG_VIN2A_D0_OEN

Address Offset	0x0000 0A48		
Physical Address	0x4844 AA48	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d0_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2216. Register Call Summary for Register CFG_VIN2A_D0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2217. CFG_VIN2A_D0_OUT

Address Offset	0x0000 0A4C		
Physical Address	0x4844 AA4C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2218. Register Call Summary for Register CFG_VIN2A_D0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2219. CFG_VIN2A_D10_IN

Address Offset	0x0000 0A50		
Physical Address	0x4844 AA50	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d10_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2220. Register Call Summary for Register CFG_VIN2A_D10_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2221. CFG_VIN2A_D10_OEN

Address Offset	0x0000 0A54		
Physical Address	0x4844 AA54	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d10_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2222. Register Call Summary for Register CFG_VIN2A_D10_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2223. CFG_VIN2A_D10_OUT

Address Offset	0x0000 0A58		
Physical Address	0x4844 AA58	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d10_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2224. Register Call Summary for Register CFG_VIN2A_D10_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2225. CFG_VIN2A_D11_IN

Address Offset	0x0000 0A5C		
Physical Address	0x4844 AA5C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d11_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2226. Register Call Summary for Register CFG_VIN2A_D11_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2227. CFG_VIN2A_D11_OEN

Address Offset	0x0000 0A60		
Physical Address	0x4844 AA60	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d11_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2228. Register Call Summary for Register CFG_VIN2A_D11_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2229. CFG_VIN2A_D11_OUT

Address Offset	0x0000 0A64		
Physical Address	0x4844 AA64	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d11_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2230. Register Call Summary for Register CFG_VIN2A_D11_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2231. CFG_VIN2A_D12_IN

Address Offset	0x0000 0A68		
Physical Address	0x4844 AA68	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d12_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2232. Register Call Summary for Register CFG_VIN2A_D12_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2233. CFG_VIN2A_D12_OEN

Address Offset	0x0000 0A6C		
Physical Address	0x4844 AA6C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d12_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2234. Register Call Summary for Register CFG_VIN2A_D12_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2235. CFG_VIN2A_D12_OUT

Address Offset	0x0000 0A70		
Physical Address	0x4844 AA70	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d12_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2236. Register Call Summary for Register CFG_VIN2A_D12_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2237. CFG_VIN2A_D13_IN

Address Offset	0x0000 0A74		
Physical Address	0x4844 AA74	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d13_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2238. Register Call Summary for Register CFG_VIN2A_D13_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2239. CFG_VIN2A_D13_OEN

Address Offset	0x0000 0A78		
Physical Address	0x4844 AA78	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d13_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2240. Register Call Summary for Register CFG_VIN2A_D13_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2241. CFG_VIN2A_D13_OUT

Address Offset	0x0000 0A7C		
Physical Address	0x4844 AA7C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d13_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2242. Register Call Summary for Register CFG_VIN2A_D13_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2243. CFG_VIN2A_D14_IN

Address Offset	0x0000 0A80		
Physical Address	0x4844 AA80	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d14_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2244. Register Call Summary for Register CFG_VIN2A_D14_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2245. CFG_VIN2A_D14_OEN

Address Offset	0x0000 0A84		
Physical Address	0x4844 AA84	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d14_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2246. Register Call Summary for Register CFG_VIN2A_D14_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2247. CFG_VIN2A_D14_OUT

Address Offset	0x0000 0A88		
Physical Address	0x4844 AA88	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d14_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2248. Register Call Summary for Register CFG_VIN2A_D14_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2249. CFG_VIN2A_D15_IN

Address Offset	0x0000 0A8C		
Physical Address	0x4844 AA8C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d15_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2250. Register Call Summary for Register CFG_VIN2A_D15_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2251. CFG_VIN2A_D15_OEN

Address Offset	0x0000 0A90		
Physical Address	0x4844 AA90	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d15_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2252. Register Call Summary for Register CFG_VIN2A_D15_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2253. CFG_VIN2A_D15_OUT

Address Offset	0x0000 0A94		
Physical Address	0x4844 AA94	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d15_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2254. Register Call Summary for Register CFG_VIN2A_D15_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2255. CFG_VIN2A_D16_IN

Address Offset	0x0000 0A98		
Physical Address	0x4844 AA98	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d16_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2256. Register Call Summary for Register CFG_VIN2A_D16_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2257. CFG_VIN2A_D16_OEN

Address Offset	0x0000 0A9C		
Physical Address	0x4844 AA9C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d16_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2258. Register Call Summary for Register CFG_VIN2A_D16_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2259. CFG_VIN2A_D16_OUT

Address Offset	0x0000 0AA0		
Physical Address	0x4844 AAA0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d16_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2260. Register Call Summary for Register CFG_VIN2A_D16_OUT

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-2261. CFG_VIN2A_D17_IN

Address Offset	0x0000 0AA4		
Physical Address	0x4844 AAA4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d17_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2262. Register Call Summary for Register CFG_VIN2A_D17_IN

IODELAYCONFIG Module Register Manual
- IODELAYCONFIG Register Summary: [0]

Table 18-2263. CFG_VIN2A_D17_OEN

Address Offset	0x0000 0AA8		
Physical Address	0x4844 AAA8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d17_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2264. Register Call Summary for Register CFG_VIN2A_D17_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2265. CFG_VIN2A_D17_OUT

Address Offset	0x0000 0AAC		
Physical Address	0x4844 AAAC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d17_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2266. Register Call Summary for Register CFG_VIN2A_D17_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2267. CFG_VIN2A_D18_IN

Address Offset	0x0000 0AB0		
Physical Address	0x4844 AAB0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d18_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2268. Register Call Summary for Register CFG_VIN2A_D18_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2269. CFG_VIN2A_D18_OEN

Address Offset	0x0000 0AB4		
Physical Address	0x4844 AAB4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d18_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2270. Register Call Summary for Register CFG_VIN2A_D18_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2271. CFG_VIN2A_D18_OUT

Address Offset	0x0000 0AB8		
Physical Address	0x4844 AAB8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d18_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2272. Register Call Summary for Register CFG_VIN2A_D18_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2273. CFG_VIN2A_D19_IN

Address Offset	0x0000 0ABC		
Physical Address	0x4844 AABC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d19_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2274. Register Call Summary for Register CFG_VIN2A_D19_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2275. CFG_VIN2A_D19_OEN

Address Offset	0x0000 0AC0		
Physical Address	0x4844 AAC0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d19_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2276. Register Call Summary for Register CFG_VIN2A_D19_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2277. CFG_VIN2A_D19_OUT

Address Offset	0x0000 0AC4		
Physical Address	0x4844 AAC4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d19_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2278. Register Call Summary for Register CFG_VIN2A_D19_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2279. CFG_VIN2A_D1_IN

Address Offset	0x0000 0AC8		
Physical Address	0x4844 AAC8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d1_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2280. Register Call Summary for Register CFG_VIN2A_D1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2281. CFG_VIN2A_D1_OEN

Address Offset	0x0000 0ACC		
Physical Address	0x4844 AACC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d1_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2282. Register Call Summary for Register CFG_VIN2A_D1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2283. CFG_VIN2A_D1_OUT

Address Offset	0x0000 0AD0		
Physical Address	0x4844 AAD0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d1_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2284. Register Call Summary for Register CFG_VIN2A_D1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2285. CFG_VIN2A_D20_IN

Address Offset	0x0000 0AD4		
Physical Address	0x4844 AAD4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d20_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2286. Register Call Summary for Register CFG_VIN2A_D20_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2287. CFG_VIN2A_D20_OEN

Address Offset	0x0000 0AD8		
Physical Address	0x4844 AAD8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d20_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2288. Register Call Summary for Register CFG_VIN2A_D20_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2289. CFG_VIN2A_D20_OUT

Address Offset	0x0000 0ADC		
Physical Address	0x4844 AADC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d20_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2290. Register Call Summary for Register CFG_VIN2A_D20_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2291. CFG_VIN2A_D21_IN

Address Offset	0x0000 0AE0		
Physical Address	0x4844 AAE0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d21_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2292. Register Call Summary for Register CFG_VIN2A_D21_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2293. CFG_VIN2A_D21_OEN

Address Offset	0x0000 0AE4		
Physical Address	0x4844 AAE4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d21_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2294. Register Call Summary for Register CFG_VIN2A_D21_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2295. CFG_VIN2A_D21_OUT

Address Offset	0x0000 0AE8		
Physical Address	0x4844 AAE8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d21_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2296. Register Call Summary for Register CFG_VIN2A_D21_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2297. CFG_VIN2A_D22_IN

Address Offset	0x0000 0AEC		
Physical Address	0x4844 AAEC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d22_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2298. Register Call Summary for Register CFG_VIN2A_D22_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2299. CFG_VIN2A_D22_OEN

Address Offset	0x0000 0AF0		
Physical Address	0x4844 AAF0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d22_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2300. Register Call Summary for Register CFG_VIN2A_D22_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2301. CFG_VIN2A_D22_OUT

Address Offset	0x0000 0AF4		
Physical Address	0x4844 AAF4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d22_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2302. Register Call Summary for Register CFG_VIN2A_D22_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2303. CFG_VIN2A_D23_IN

Address Offset	0x0000 0AF8		
Physical Address	0x4844 AAF8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d23_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2304. Register Call Summary for Register CFG_VIN2A_D23_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2305. CFG_VIN2A_D23_OEN

Address Offset	0x0000 0AFC		
Physical Address	0x4844 AAFC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d23_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2306. Register Call Summary for Register CFG_VIN2A_D23_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2307. CFG_VIN2A_D23_OUT

Address Offset	0x0000 0B00		
Physical Address	0x4844 AB00	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d23_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2308. Register Call Summary for Register CFG_VIN2A_D23_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2309. CFG_VIN2A_D2_IN

Address Offset	0x0000 0B04		
Physical Address	0x4844 AB04	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2310. Register Call Summary for Register CFG_VIN2A_D2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2311. CFG_VIN2A_D2_OEN

Address Offset	0x0000 0B08		
Physical Address	0x4844 AB08	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2312. Register Call Summary for Register CFG_VIN2A_D2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2313. CFG_VIN2A_D2_OUT

Address Offset	0x0000 0B0C		
Physical Address	0x4844 AB0C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2314. Register Call Summary for Register CFG_VIN2A_D2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2315. CFG_VIN2A_D3_IN

Address Offset	0x0000 0B10		
Physical Address	0x4844 AB10	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d3_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2316. Register Call Summary for Register CFG_VIN2A_D3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2317. CFG_VIN2A_D3_OEN

Address Offset	0x0000 0B14		
Physical Address	0x4844 AB14	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d3_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2318. Register Call Summary for Register CFG_VIN2A_D3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2319. CFG_VIN2A_D3_OUT

Address Offset	0x0000 0B18		
Physical Address	0x4844 AB18	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d3_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2320. Register Call Summary for Register CFG_VIN2A_D3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2321. CFG_VIN2A_D4_IN

Address Offset	0x0000 0B1C		
Physical Address	0x4844 AB1C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d4_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2322. Register Call Summary for Register CFG_VIN2A_D4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2323. CFG_VIN2A_D4_OEN

Address Offset	0x0000 0B20		
Physical Address	0x4844 AB20	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d4_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2324. Register Call Summary for Register CFG_VIN2A_D4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2325. CFG_VIN2A_D4_OUT

Address Offset	0x0000 0B24		
Physical Address	0x4844 AB24	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d4_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2326. Register Call Summary for Register CFG_VIN2A_D4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2327. CFG_VIN2A_D5_IN

Address Offset	0x0000 0B28		
Physical Address	0x4844 AB28	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d5_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2328. Register Call Summary for Register CFG_VIN2A_D5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2329. CFG_VIN2A_D5_OEN

Address Offset	0x0000 0B2C		
Physical Address	0x4844 AB2C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d5_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2330. Register Call Summary for Register CFG_VIN2A_D5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2331. CFG_VIN2A_D5_OUT

Address Offset	0x0000 0B30		
Physical Address	0x4844 AB30	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d5_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2332. Register Call Summary for Register CFG_VIN2A_D5_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2333. CFG_VIN2A_D6_IN

Address Offset	0x0000 0B34		
Physical Address	0x4844 AB34	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d6_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2334. Register Call Summary for Register CFG_VIN2A_D6_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2335. CFG_VIN2A_D6_OEN

Address Offset	0x0000 0B38		
Physical Address	0x4844 AB38	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d6_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2336. Register Call Summary for Register CFG_VIN2A_D6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2337. CFG_VIN2A_D6_OUT

Address Offset	0x0000 0B3C		
Physical Address	0x4844 AB3C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d6_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2338. Register Call Summary for Register CFG_VIN2A_D6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2339. CFG_VIN2A_D7_IN

Address Offset	0x0000 0B40		
Physical Address	0x4844 AB40	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d7_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2340. Register Call Summary for Register CFG_VIN2A_D7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2341. CFG_VIN2A_D7_OEN

Address Offset	0x0000 0B44		
Physical Address	0x4844 AB44	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d7_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2342. Register Call Summary for Register CFG_VIN2A_D7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2343. CFG_VIN2A_D7_OUT

Address Offset	0x0000 0B48		
Physical Address	0x4844 AB48	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d7_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2344. Register Call Summary for Register CFG_VIN2A_D7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2345. CFG_VIN2A_D8_IN

Address Offset	0x0000 0B4C		
Physical Address	0x4844 AB4C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d8_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2346. Register Call Summary for Register CFG_VIN2A_D8_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2347. CFG_VIN2A_D8_OEN

Address Offset	0x0000 0B50		
Physical Address	0x4844 AB50	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d8_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2348. Register Call Summary for Register CFG_VIN2A_D8_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2349. CFG_VIN2A_D8_OUT

Address Offset	0x0000 0B54		
Physical Address	0x4844 AB54	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d8_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2350. Register Call Summary for Register CFG_VIN2A_D8_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2351. CFG_VIN2A_D9_IN

Address Offset	0x0000 0B58		
Physical Address	0x4844 AB58	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d9_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2352. Register Call Summary for Register CFG_VIN2A_D9_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2353. CFG_VIN2A_D9_OEN

Address Offset	0x0000 0B5C		
Physical Address	0x4844 AB5C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d9_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2354. Register Call Summary for Register CFG_VIN2A_D9_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2355. CFG_VIN2A_D9_OUT

Address Offset	0x0000 0B60		
Physical Address	0x4844 AB60	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_d9_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2356. Register Call Summary for Register CFG_VIN2A_D9_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2357. CFG_VIN2A_DE0_IN

Address Offset	0x0000 0B64		
Physical Address	0x4844 AB64	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_de0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2358. Register Call Summary for Register CFG_VIN2A_DE0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2359. CFG_VIN2A_DE0_OEN

Address Offset	0x0000 0B68		
Physical Address	0x4844 AB68	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_de0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2360. Register Call Summary for Register CFG_VIN2A_DE0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2361. CFG_VIN2A_DE0_OUT

Address Offset	0x0000 0B6C		
Physical Address	0x4844 AB6C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_de0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2362. Register Call Summary for Register CFG_VIN2A_DE0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2363. CFG_VIN2A_FLD0_IN

Address Offset	0x0000 0B70		
Physical Address	0x4844 AB70	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_fld0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2364. Register Call Summary for Register CFG_VIN2A_FLD0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2365. CFG_VIN2A_FLD0_OEN

Address Offset	0x0000 0B74		
Physical Address	0x4844 AB74	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_fld0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2366. Register Call Summary for Register CFG_VIN2A_FLD0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2367. CFG_VIN2A_FLD0_OUT

Address Offset	0x0000 0B78		
Physical Address	0x4844 AB78	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_fld0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2368. Register Call Summary for Register CFG_VIN2A_FLD0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2369. CFG_VIN2A_HSYNC0_IN

Address Offset	0x0000 0B7C		
Physical Address	0x4844 AB7C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_hsync0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2370. Register Call Summary for Register CFG_VIN2A_HSYNC0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2371. CFG_VIN2A_HSYNC0_OEN

Address Offset	0x0000 0B80		
Physical Address	0x4844 AB80	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_hsync0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2372. Register Call Summary for Register CFG_VIN2A_HSYNC0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2373. CFG_VIN2A_HSYNC0_OUT

Address Offset	0x0000 0B84		
Physical Address	0x4844 AB84	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_hsync0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2374. Register Call Summary for Register CFG_VIN2A_HSYNC0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2375. CFG_VIN2A_VSYNC0_IN

Address Offset	0x0000 0B88		
Physical Address	0x4844 AB88	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_vsync0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2376. Register Call Summary for Register CFG_VIN2A_VSYNC0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2377. CFG_VIN2A_VSYNC0_OEN

Address Offset	0x0000 0B8C		
Physical Address	0x4844 AB8C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_vsync0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2378. Register Call Summary for Register CFG_VIN2A_VSYNC0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2379. CFG_VIN2A_VSYNC0_OUT

Address Offset	0x0000 0B90		
Physical Address	0x4844 AB90	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vin2a_vsync0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2380. Register Call Summary for Register CFG_VIN2A_VSYNC0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2381. CFG_VOUT1_CLK_IN

Address Offset	0x0000 0B94		
Physical Address	0x4844 AB94	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_clk_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2382. Register Call Summary for Register CFG_VOUT1_CLK_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2383. CFG_VOUT1_CLK_OEN

Address Offset	0x0000 0B98		
Physical Address	0x4844 AB98	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_clk_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2384. Register Call Summary for Register CFG_VOUT1_CLK_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2385. CFG_VOUT1_CLK_OUT

Address Offset	0x0000 0B9C		
Physical Address	0x4844 AB9C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_clk_out interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2386. Register Call Summary for Register CFG_VOUT1_CLK_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2387. CFG_VOUT1_D0_IN

Address Offset	0x0000 0BA0		
Physical Address	0x4844 ABA0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d0_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2388. Register Call Summary for Register CFG_VOUT1_D0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2389. CFG_VOUT1_D0_OEN

Address Offset	0x0000 0BA4		
Physical Address	0x4844 ABA4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2390. Register Call Summary for Register CFG_VOUT1_D0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2391. CFG_VOUT1_D0_OUT

Address Offset	0x0000 0BA8		
Physical Address	0x4844 ABA8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2392. Register Call Summary for Register CFG_VOUT1_D0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2393. CFG_VOUT1_D10_IN

Address Offset	0x0000 0BAC		
Physical Address	0x4844 ABAC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d10_in interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2394. Register Call Summary for Register CFG_VOUT1_D10_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2395. CFG_VOUT1_D10_OEN

Address Offset	0x0000 0BB0		
Physical Address	0x4844 ABB0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d10_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2396. Register Call Summary for Register CFG_VOUT1_D10_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2397. CFG_VOUT1_D10_OUT

Address Offset	0x0000 0BB4		
Physical Address	0x4844 ABB4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d10_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2398. Register Call Summary for Register CFG_VOUT1_D10_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2399. CFG_VOUT1_D11_IN

Address Offset	0x0000 0BB8		
Physical Address	0x4844 ABB8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d11_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2400. Register Call Summary for Register CFG_VOUT1_D11_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2401. CFG_VOUT1_D11_OEN

Address Offset	0x0000 0BBC		
Physical Address	0x4844 ABBC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d11_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2402. Register Call Summary for Register CFG_VOUT1_D11_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2403. CFG_VOUT1_D11_OUT

Address Offset	0x0000 0BC0		
Physical Address	0x4844 ABC0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d11_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2404. Register Call Summary for Register CFG_VOUT1_D11_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2405. CFG_VOUT1_D12_IN

Address Offset	0x0000 0BC4		
Physical Address	0x4844 ABC4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d12_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2406. Register Call Summary for Register CFG_VOUT1_D12_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2407. CFG_VOUT1_D12_OEN

Address Offset	0x0000 0BC8		
Physical Address	0x4844 ABC8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d12_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2408. Register Call Summary for Register CFG_VOUT1_D12_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2409. CFG_VOUT1_D12_OUT

Address Offset	0x0000 0BCC		
Physical Address	0x4844 ABCC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d12_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2410. Register Call Summary for Register CFG_VOUT1_D12_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2411. CFG_VOUT1_D13_IN

Address Offset	0x0000 0BD0		
Physical Address	0x4844 ABD0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d13_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2412. Register Call Summary for Register CFG_VOUT1_D13_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2413. CFG_VOUT1_D13_OEN

Address Offset	0x0000 0BD4		
Physical Address	0x4844 ABD4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d13_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2414. Register Call Summary for Register CFG_VOUT1_D13_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2415. CFG_VOUT1_D13_OUT

Address Offset	0x0000 0BD8		
Physical Address	0x4844 ABD8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d13_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2416. Register Call Summary for Register CFG_VOUT1_D13_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2417. CFG_VOUT1_D14_IN

Address Offset	0x0000 0BDC		
Physical Address	0x4844 ABDC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d14_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2418. Register Call Summary for Register CFG_VOUT1_D14_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2419. CFG_VOUT1_D14_OEN

Address Offset	0x0000 0BE0		
Physical Address	0x4844 ABE0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d14_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2420. Register Call Summary for Register CFG_VOUT1_D14_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2421. CFG_VOUT1_D14_OUT

Address Offset	0x0000 0BE4		
Physical Address	0x4844 ABE4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d14_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2422. Register Call Summary for Register CFG_VOUT1_D14_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2423. CFG_VOUT1_D15_IN

Address Offset	0x0000 0BE8		
Physical Address	0x4844 ABE8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d15_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2424. Register Call Summary for Register CFG_VOUT1_D15_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2425. CFG_VOUT1_D15_OEN

Address Offset	0x0000 0BEC		
Physical Address	0x4844 ABEC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d15_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2426. Register Call Summary for Register CFG_VOUT1_D15_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2427. CFG_VOUT1_D15_OUT

Address Offset	0x0000 0BF0		
Physical Address	0x4844 ABF0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d15_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2428. Register Call Summary for Register CFG_VOUT1_D15_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2429. CFG_VOUT1_D16_IN

Address Offset	0x0000 0BF4		
Physical Address	0x4844 ABF4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d16_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2430. Register Call Summary for Register CFG_VOUT1_D16_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2431. CFG_VOUT1_D16_OEN

Address Offset	0x0000 0BF8		
Physical Address	0x4844 ABF8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d16_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2432. Register Call Summary for Register CFG_VOUT1_D16_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2433. CFG_VOUT1_D16_OUT

Address Offset	0x0000 0BFC		
Physical Address	0x4844 ABFC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d16_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2434. Register Call Summary for Register CFG_VOUT1_D16_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2435. CFG_VOUT1_D17_IN

Address Offset	0x0000 0C00		
Physical Address	0x4844 AC00	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d17_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2436. Register Call Summary for Register CFG_VOUT1_D17_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2437. CFG_VOUT1_D17_OEN

Address Offset	0x0000 0C04		
Physical Address	0x4844 AC04	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d17_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2438. Register Call Summary for Register CFG_VOUT1_D17_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2439. CFG_VOUT1_D17_OUT

Address Offset	0x0000 0C08		
Physical Address	0x4844 AC08	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d17_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2440. Register Call Summary for Register CFG_VOUT1_D17_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2441. CFG_VOUT1_D18_IN

Address Offset	0x0000 0C0C		
Physical Address	0x4844 AC0C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d18_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2442. Register Call Summary for Register CFG_VOUT1_D18_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2443. CFG_VOUT1_D18_OEN

Address Offset	0x0000 0C10		
Physical Address	0x4844 AC10	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d18_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2444. Register Call Summary for Register CFG_VOUT1_D18_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2445. CFG_VOUT1_D18_OUT

Address Offset	0x0000 0C14		
Physical Address	0x4844 AC14	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d18_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2446. Register Call Summary for Register CFG_VOUT1_D18_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2447. CFG_VOUT1_D19_IN

Address Offset	0x0000 0C18		
Physical Address	0x4844 AC18	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d19_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2448. Register Call Summary for Register CFG_VOUT1_D19_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2449. CFG_VOUT1_D19_OEN

Address Offset	0x0000 0C1C		
Physical Address	0x4844 AC1C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d19_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2450. Register Call Summary for Register CFG_VOUT1_D19_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2451. CFG_VOUT1_D19_OUT

Address Offset	0x0000 0C20		
Physical Address	0x4844 AC20	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d19_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2452. Register Call Summary for Register CFG_VOUT1_D19_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2453. CFG_VOUT1_D1_IN

Address Offset	0x0000 0C24		
Physical Address	0x4844 AC24	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2454. Register Call Summary for Register CFG_VOUT1_D1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2455. CFG_VOUT1_D1_OEN

Address Offset	0x0000 0C28		
Physical Address	0x4844 AC28	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2456. Register Call Summary for Register CFG_VOUT1_D1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2457. CFG_VOUT1_D1_OUT

Address Offset	0x0000 0C2C		
Physical Address	0x4844 AC2C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2458. Register Call Summary for Register CFG_VOUT1_D1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2459. CFG_VOUT1_D20_IN

Address Offset	0x0000 0C30		
Physical Address	0x4844 AC30	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d20_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2460. Register Call Summary for Register CFG_VOUT1_D20_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2461. CFG_VOUT1_D20_OEN

Address Offset	0x0000 0C34		
Physical Address	0x4844 AC34	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d20_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2462. Register Call Summary for Register CFG_VOUT1_D20_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2463. CFG_VOUT1_D20_OUT

Address Offset	0x0000 0C38		
Physical Address	0x4844 AC38	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d20_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2464. Register Call Summary for Register CFG_VOUT1_D20_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2465. CFG_VOUT1_D21_IN

Address Offset	0x0000 0C3C		
Physical Address	0x4844 AC3C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d21_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2466. Register Call Summary for Register CFG_VOUT1_D21_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2467. CFG_VOUT1_D21_OEN

Address Offset	0x0000 0C40		
Physical Address	0x4844 AC40	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d21_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2468. Register Call Summary for Register CFG_VOUT1_D21_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2469. CFG_VOUT1_D21_OUT

Address Offset	0x0000 0C44		
Physical Address	0x4844 AC44	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d21_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2470. Register Call Summary for Register CFG_VOUT1_D21_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2471. CFG_VOUT1_D22_IN

Address Offset	0x0000 0C48		
Physical Address	0x4844 AC48	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d22_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2472. Register Call Summary for Register CFG_VOUT1_D22_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2473. CFG_VOUT1_D22_OEN

Address Offset	0x0000 0C4C		
Physical Address	0x4844 AC4C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d22_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2474. Register Call Summary for Register CFG_VOUT1_D22_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2475. CFG_VOUT1_D22_OUT

Address Offset	0x0000 0C50		
Physical Address	0x4844 AC50	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d22_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2476. Register Call Summary for Register CFG_VOUT1_D22_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2477. CFG_VOUT1_D23_IN

Address Offset	0x0000 0C54		
Physical Address	0x4844 AC54	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d23_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2478. Register Call Summary for Register CFG_VOUT1_D23_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2479. CFG_VOUT1_D23_OEN

Address Offset	0x0000 0C58		
Physical Address	0x4844 AC58	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d23_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2480. Register Call Summary for Register CFG_VOUT1_D23_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2481. CFG_VOUT1_D23_OUT

Address Offset	0x0000 0C5C		
Physical Address	0x4844 AC5C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d23_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2482. Register Call Summary for Register CFG_VOUT1_D23_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2483. CFG_VOUT1_D2_IN

Address Offset	0x0000 0C60		
Physical Address	0x4844 AC60	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2484. Register Call Summary for Register CFG_VOUT1_D2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2485. CFG_VOUT1_D2_OEN

Address Offset	0x0000 0C64		
Physical Address	0x4844 AC64	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d2_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2486. Register Call Summary for Register CFG_VOUT1_D2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2487. CFG_VOUT1_D2_OUT

Address Offset	0x0000 0C68		
Physical Address	0x4844 AC68	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d2_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2488. Register Call Summary for Register CFG_VOUT1_D2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2489. CFG_VOUT1_D3_IN

Address Offset	0x0000 0C6C		
Physical Address	0x4844 AC6C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d3_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2490. Register Call Summary for Register CFG_VOUT1_D3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2491. CFG_VOUT1_D3_OEN

Address Offset	0x0000 0C70		
Physical Address	0x4844 AC70	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d3_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2492. Register Call Summary for Register CFG_VOUT1_D3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2493. CFG_VOUT1_D3_OUT

Address Offset	0x0000 0C74		
Physical Address	0x4844 AC74	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2494. Register Call Summary for Register CFG_VOUT1_D3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2495. CFG_VOUT1_D4_IN

Address Offset	0x0000 0C78		
Physical Address	0x4844 AC78	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d4_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2496. Register Call Summary for Register CFG_VOUT1_D4_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2497. CFG_VOUT1_D4_OEN

Address Offset	0x0000 0C7C		
Physical Address	0x4844 AC7C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d4_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2498. Register Call Summary for Register CFG_VOUT1_D4_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2499. CFG_VOUT1_D4_OUT

Address Offset	0x0000 0C80		
Physical Address	0x4844 AC80	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d4_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2500. Register Call Summary for Register CFG_VOUT1_D4_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2501. CFG_VOUT1_D5_IN

Address Offset	0x0000 0C84		
Physical Address	0x4844 AC84	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d5_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2502. Register Call Summary for Register CFG_VOUT1_D5_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2503. CFG_VOUT1_D5_OEN

Address Offset	0x0000 0C88		
Physical Address	0x4844 AC88	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d5_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2504. Register Call Summary for Register CFG_VOUT1_D5_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2505. CFG_VOUT1_D5_OUT

Address Offset	0x0000 0C8C		
Physical Address	0x4844 AC8C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d5_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2506. Register Call Summary for Register CFG_VOUT1_D5_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2507. CFG_VOUT1_D6_IN

Address Offset	0x0000 0C90		
Physical Address	0x4844 AC90	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d6_in interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2508. Register Call Summary for Register CFG_VOUT1_D6_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2509. CFG_VOUT1_D6_OEN

Address Offset	0x0000 0C94		
Physical Address	0x4844 AC94	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d6_oen interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2510. Register Call Summary for Register CFG_VOUT1_D6_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2511. CFG_VOUT1_D6_OUT

Address Offset	0x0000 0C98		
Physical Address	0x4844 AC98	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d6_out interface		
Type	RW		

31–18	17–12	11	10	9–0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2512. Register Call Summary for Register CFG_VOUT1_D6_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2513. CFG_VOUT1_D7_IN

Address Offset	0x0000 0C9C		
Physical Address	0x4844 AC9C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d7_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2514. Register Call Summary for Register CFG_VOUT1_D7_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2515. CFG_VOUT1_D7_OEN

Address Offset	0x0000 0CA0		
Physical Address	0x4844 ACA0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d7_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2516. Register Call Summary for Register CFG_VOUT1_D7_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2517. CFG_VOUT1_D7_OUT

Address Offset	0x0000 0CA4		
Physical Address	0x4844 ACA4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d7_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2518. Register Call Summary for Register CFG_VOUT1_D7_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2519. CFG_VOUT1_D8_IN

Address Offset	0x0000 0CA8		
Physical Address	0x4844 ACA8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d8_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2520. Register Call Summary for Register CFG_VOUT1_D8_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2521. CFG_VOUT1_D8_OEN

Address Offset	0x0000 0CAC		
Physical Address	0x4844 ACAC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d8_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2522. Register Call Summary for Register CFG_VOUT1_D8_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2523. CFG_VOUT1_D8_OUT

Address Offset	0x0000 0CB0		
Physical Address	0x4844 ACB0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d8_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2524. Register Call Summary for Register CFG_VOUT1_D8_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2525. CFG_VOUT1_D9_IN

Address Offset	0x0000 0CB4		
Physical Address	0x4844 ACB4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d9_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2526. Register Call Summary for Register CFG_VOUT1_D9_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2527. CFG_VOUT1_D9_OEN

Address Offset	0x0000 0CB8		
Physical Address	0x4844 ACB8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d9_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2528. Register Call Summary for Register CFG_VOUT1_D9_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2529. CFG_VOUT1_D9_OUT

Address Offset	0x0000 0CBC		
Physical Address	0x4844 ACBC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_d9_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2530. Register Call Summary for Register CFG_VOUT1_D9_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2531. CFG_VOUT1_DE_IN

Address Offset	0x0000 0CC0		
Physical Address	0x4844 ACC0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_de_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2532. Register Call Summary for Register CFG_VOUT1_DE_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2533. CFG_VOUT1_DE_OEN

Address Offset	0x0000 0CC4		
Physical Address	0x4844 ACC4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_de_oen interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2534. Register Call Summary for Register CFG_VOUT1_DE_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2535. CFG_VOUT1_DE_OUT

Address Offset	0x0000 0CC8		
Physical Address	0x4844 ACC8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_de_out interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SIGNATURE						RESERVED	LOCK_BIT	BINARY_DELAY									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2536. Register Call Summary for Register CFG_VOUT1_DE_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2537. CFG_VOUT1_FLD_IN

Address Offset	0x0000 0CCC		
Physical Address	0x4844 ACCC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_fld_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2538. Register Call Summary for Register CFG_VOUT1_FLD_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2539. CFG_VOUT1_FLD_OEN

Address Offset	0x0000 0CD0		
Physical Address	0x4844 ACD0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_fld_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2540. Register Call Summary for Register CFG_VOUT1_FLD_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2541. CFG_VOUT1_FLD_OUT

Address Offset	0x0000 0CD4		
Physical Address	0x4844 ACD4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_fld_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2542. Register Call Summary for Register CFG_VOUT1_FLD_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2543. CFG_VOUT1_HSYNC_IN

Address Offset	0x0000 0CD8		
Physical Address	0x4844 ACD8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_hsync_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2544. Register Call Summary for Register CFG_VOUT1_HSYNC_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2545. CFG_VOUT1_HSYNC_OEN

Address Offset	0x0000 0CDC		
Physical Address	0x4844 ACDC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_hsync_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2546. Register Call Summary for Register CFG_VOUT1_HSYNC_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2547. CFG_VOUT1_HSYNC_OUT

Address Offset	0x0000 0CE0		
Physical Address	0x4844 ACE0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_hsync_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2548. Register Call Summary for Register CFG_VOUT1_HSYNC_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2549. CFG_VOUT1_VSYNC_IN

Address Offset	0x0000 0CE4		
Physical Address	0x4844 ACE4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_vsync_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2550. Register Call Summary for Register CFG_VOUT1_VSYNC_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2551. CFG_VOUT1_VSYNC_OEN

Address Offset	0x0000 0CE8		
Physical Address	0x4844 ACE8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_vsync_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2552. Register Call Summary for Register CFG_VOUT1_VSYNC_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2553. CFG_VOUT1_VSYNC_OUT

Address Offset	0x0000 0CEC		
Physical Address	0x4844 ACEC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_vout1_vsync_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2554. Register Call Summary for Register CFG_VOUT1_VSYNC_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2555. CFG_XREF_CLK0_IN

Address Offset	0x0000 0CF0		
Physical Address	0x4844 ACF0	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk0_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2556. Register Call Summary for Register CFG_XREF_CLK0_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2557. CFG_XREF_CLK0_OEN

Address Offset	0x0000 0CF4		
Physical Address	0x4844 ACF4	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk0_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2558. Register Call Summary for Register CFG_XREF_CLK0_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2559. CFG_XREF_CLK0_OUT

Address Offset	0x0000 0CF8		
Physical Address	0x4844 ACF8	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk0_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2560. Register Call Summary for Register CFG_XREF_CLK0_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2561. CFG_XREF_CLK1_IN

Address Offset	0x0000 0CFC		
Physical Address	0x4844 ACFC	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk1_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2562. Register Call Summary for Register CFG_XREF_CLK1_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2563. CFG_XREF_CLK1_OEN

Address Offset	0x0000 0D00		
Physical Address	0x4844 AD00	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk1_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2564. Register Call Summary for Register CFG_XREF_CLK1_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2565. CFG_XREF_CLK1_OUT

Address Offset	0x0000 0D04		
Physical Address	0x4844 AD04	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk1_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2566. Register Call Summary for Register CFG_XREF_CLK1_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2567. CFG_XREF_CLK2_IN

Address Offset	0x0000 0D08		
Physical Address	0x4844 AD08	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk2_in interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2568. Register Call Summary for Register CFG_XREF_CLK2_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2569. CFG_XREF_CLK2_OEN

Address Offset	0x0000 0D0C		
Physical Address	0x4844 AD0C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk2_oen interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2570. Register Call Summary for Register CFG_XREF_CLK2_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2571. CFG_XREF_CLK2_OUT

Address Offset	0x0000 0D10		
Physical Address	0x4844 AD10	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk2_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2572. Register Call Summary for Register CFG_XREF_CLK2_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2573. CFG_XREF_CLK3_IN

Address Offset	0x0000 0D14		
Physical Address	0x4844 AD14	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk3_in interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2574. Register Call Summary for Register CFG_XREF_CLK3_IN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2575. CFG_XREF_CLK3_OEN

Address Offset	0x0000 0D18		
Physical Address	0x4844 AD18	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk3_oen interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="6">SIGNATURE</td><td>RESERVED</td><td>LOCK_BIT</td><td colspan="10">BINARY_DELAY</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2576. Register Call Summary for Register CFG_XREF_CLK3_OEN

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Table 18-2577. CFG_XREF_CLK3_OUT

Address Offset	0x0000 0D1C		
Physical Address	0x4844 AD1C	**Instance**	IODELAYCONFIG
Description	Delay Select Value in binary coded form for cfg_xref_clk3_out interface		
Type	RW		

31:18	17:12	11	10	9:0
RESERVED	SIGNATURE	RESERVED	LOCK_BIT	BINARY_DELAY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:12	SIGNATURE	Write to this register will succeed only if data on these bits carries a signature of 6'h29 (6'b101001)	RW	0x0
11	RESERVED		R	0x0
10	LOCK_BIT	When '1', prevents HW update to this MMR. When '0', allows HW update of this MMR.	RW	0x0
9:0	BINARY_DELAY	Delay Select Value in binary coded form. Indicates the number of coarse (9:5) and fine (4:0) delay elements to be used on the delay line connected to this pad.	RW	0x0

Table 18-2578. Register Call Summary for Register CFG_XREF_CLK3_OUT

IODELAYCONFIG Module Register Manual

- IODELAYCONFIG Register Summary: [0]

Chapter 19
Mailbox

This chapter describes the mailbox module in the device.

19.1 Mailbox Overview........ 5107
19.2 Mailbox Integration........ 5107
19.3 Mailbox Functional Description........ 5115
19.4 Mailbox Programming Guide........ 5121
19.5 Mailbox Register Manual........ 5124

19.1 Mailbox Overview

Communication between the on-chip processors of the device uses a queued mailbox-interrupt mechanism.

The queued mailbox-interrupt mechanism allows the software to establish a communication channel between two processors through a set of registers and associated interrupt signals by sending and receiving messages (mailboxes).

The device implements the following mailbox types:

- System mailbox:
  - Number of instances: 13
  - Used for communication between: MPU, DSP1, DSP2, IPU1, and IPU2 subsystems
  - Reference name: MAILBOX(1..13)
- IVA mailbox:
  - Number of instances: 1
  - Used for communication between: IVA local user (ICONT1, or ICONT2) and three external users (selected among MPU, DSP1, DSP2, IPU1, and IPU2 subsystems)
  - Reference name: IVA_MBOX
- EVEx mailbox (where x = 1, 2):
  - Number of instances: 3 (per EVE)
  - Used for communication between:
    - EVEx local user (ARP32) and three external users (selected among MPU, DSP1, DSP2, IPU1, and IPU2) - EVEx_MBOX0 and EVEx_MBOX1 are dedicated for this communication
    - EVE1 and EVE2 local users - EVEx_MBOX2 is dedicated for this communication
  - Reference name: EVEx_MBOX(0..2)

Each mailbox module supports the following features:

- Parameters configurable at design time (see Table 19-1):
  - Number of users
  - Number of mailbox message queues
  - Number of messages (FIFO depth) for each message queue
- 32-bit message width
- Message reception and queue-not-full notification using interrupts
- Support of 16-/32-bit addressing scheme
- Power management support

Table 19-1 shows the configuartion of the mailbox modules in the device.

Table 19-1. Mailbox Configuration in the Device

<table>
<tr><th rowspan="3">Module Parameters</th><th colspan="4">Mailbox Type</th></tr>
<tr><th colspan="2">System Mailbox</th><th rowspan="2">IVA Mailbox</th><th rowspan="2">EVEx Mailbox (0..2)</th></tr>
<tr><th>MAILBOX1</th><th>MAILBOX2..13</th></tr>
<tr><td>Number of users</td><td>3</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>Number of mailbox message queues</td><td>8</td><td>12</td><td>6</td><td>16</td></tr>
<tr><td>Number of messages (FIFO depth) for each message queue</td><td colspan="2">4</td><td>4</td><td>4</td></tr>
</table>

19.2 Mailbox Integration

This section describes the mailbox integration in the device, including information about clocks, resets, and hardware requests.

19.2.1 System MAILBOX Integration

The integration of MAILBOX1 in the device differs from the integration of the rest of the system mailboxes (MAILBOX2..13). Figure 19-1 and Figure 19-2 show the MAILBOX1 and MAILBOX2..13 integration, respectively.

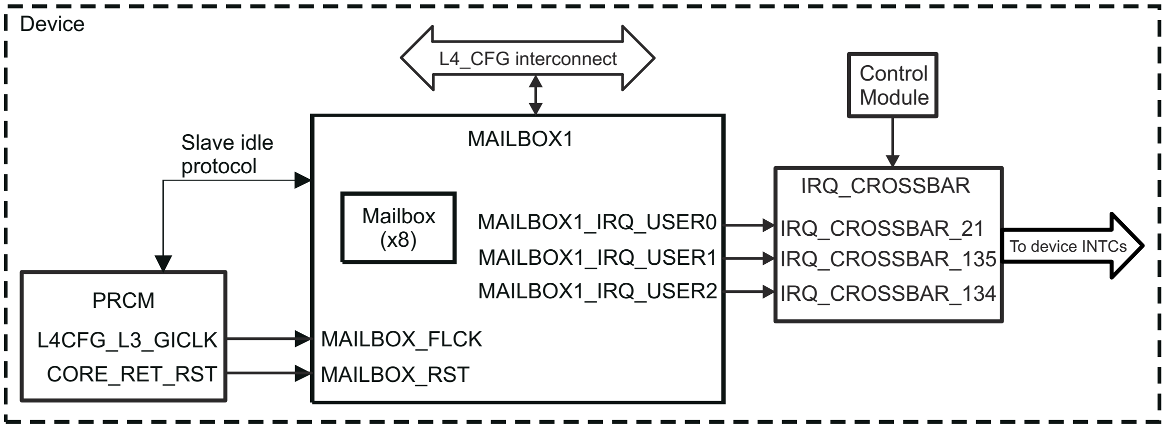


Figure 19-1. MAILBOX1 Integration

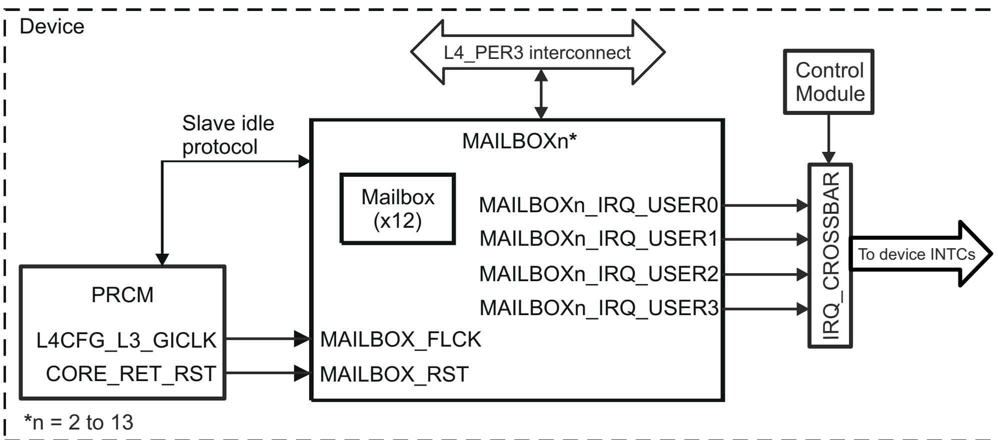


Figure 19-2. MAILBOX2..13 Integration

Note

For more information about the Slave idle protocol, see *Module-Level Clock Management* in *Power, Reset, and Clock Management*.

Table 19-2 through Table 19-4 summarize the MAILBOX(1..13) integration in the device.

Table 19-2. MAILBOX Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
MAILBOX1	PD_COREAON	N/A	L4_CFG
MAILBOX2..13	PD_COREAON	N/A	L4_PER3

Table 19-3. MAILBOX Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description

Table 19-3. MAILBOX Clocks and Resets (continued)

MAILBOX	MAILBOX_FCLK	L4CFG_L3_GICLK	PRCM	MAILBOX interface clock. This clock is used for all interface and functional operations.

Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
MAILBOX	MAILBOX_RST	CORE_RET_RST	PRCM	MAILBOX hardware reset. This reset is asynchronously applied to the MAILBOX registers.

Table 19-4. MAILBOX Hardware Requests

Interrupt Requests

Module Instance	Interrupt Name (Source)	IRQ_CROSSBAR Input (Destination)	Default Mapping	Description
MAILBOX1	MAILBOX1_IRQ_USER0	IRQ_CROSSBAR_21	MPU_IRQ_26; DSP1_IRQ_52; DSP2_IRQ_52;	MAILBOX1 user 0 interrupt request.
	MAILBOX1_IRQ_USER1	IRQ_CROSSBAR_135	–	MAILBOX1 user 1 interrupt request.
	MAILBOX1_IRQ_USER2	IRQ_CROSSBAR_134	IPU1_IRQ_50; IPU2_IRQ_50;	MAILBOX1 user 2 interrupt request.
MAILBOX2	MAILBOX2_IRQ_USER0	IRQ_CROSSBAR_237	–	MAILBOX2 user 0 interrupt request.
	MAILBOX2_IRQ_USER1	IRQ_CROSSBAR_238	–	MAILBOX2 user 1 interrupt request.
	MAILBOX2_IRQ_USER2	IRQ_CROSSBAR_239	–	MAILBOX2 user 2 interrupt request.
	MAILBOX2_IRQ_USER3	IRQ_CROSSBAR_240	–	MAILBOX2 user 3 interrupt request.
MAILBOX3	MAILBOX3_IRQ_USER0	IRQ_CROSSBAR_241	–	MAILBOX3 user 0 interrupt request.
	MAILBOX3_IRQ_USER1	IRQ_CROSSBAR_242	–	MAILBOX3 user 1 interrupt request.
	MAILBOX3_IRQ_USER2	IRQ_CROSSBAR_243	–	MAILBOX3 user 2 interrupt request.
	MAILBOX3_IRQ_USER3	IRQ_CROSSBAR_244	–	MAILBOX3 user 3 interrupt request.
MAILBOX4	MAILBOX4_IRQ_USER0	IRQ_CROSSBAR_245	–	MAILBOX4 user 0 interrupt request.
	MAILBOX4_IRQ_USER1	IRQ_CROSSBAR_246	–	MAILBOX4 user 1 interrupt request.
	MAILBOX4_IRQ_USER2	IRQ_CROSSBAR_247	–	MAILBOX4 user 2 interrupt request.
	MAILBOX4_IRQ_USER3	IRQ_CROSSBAR_248	–	MAILBOX4 user 3 interrupt request.
MAILBOX5	MAILBOX5_IRQ_USER0	IRQ_CROSSBAR_249	–	MAILBOX5 user 0 interrupt request.
	MAILBOX5_IRQ_USER1	IRQ_CROSSBAR_250	–	MAILBOX5 user 1 interrupt request.
	MAILBOX5_IRQ_USER2	IRQ_CROSSBAR_251	–	MAILBOX5 user 2 interrupt request.
	MAILBOX5_IRQ_USER3	IRQ_CROSSBAR_252	–	MAILBOX5 user 3 interrupt request.
MAILBOX6	MAILBOX6_IRQ_USER0	IRQ_CROSSBAR_253	–	MAILBOX6 user 0 interrupt request.
	MAILBOX6_IRQ_USER1	IRQ_CROSSBAR_254	–	MAILBOX6 user 1 interrupt request.
	MAILBOX6_IRQ_USER2	IRQ_CROSSBAR_255	–	MAILBOX6 user 2 interrupt request.
	MAILBOX6_IRQ_USER3	IRQ_CROSSBAR_256	–	MAILBOX6 user 3 interrupt request.
MAILBOX7	MAILBOX7_IRQ_USER0	IRQ_CROSSBAR_257	–	MAILBOX7 user 0 interrupt request.
	MAILBOX7_IRQ_USER1	IRQ_CROSSBAR_258	–	MAILBOX7 user 1 interrupt request.
	MAILBOX7_IRQ_USER2	IRQ_CROSSBAR_259	–	MAILBOX7 user 2 interrupt request.
	MAILBOX7_IRQ_USER3	IRQ_CROSSBAR_260	–	MAILBOX7 user 3 interrupt request.
MAILBOX8	MAILBOX8_IRQ_USER0	IRQ_CROSSBAR_261	–	MAILBOX8 user 0 interrupt request.
	MAILBOX8_IRQ_USER1	IRQ_CROSSBAR_262	–	MAILBOX8 user 1 interrupt request.
	MAILBOX8_IRQ_USER2	IRQ_CROSSBAR_263	–	MAILBOX8 user 2 interrupt request.
	MAILBOX8_IRQ_USER3	IRQ_CROSSBAR_264	–	MAILBOX8 user 3 interrupt request.
MAILBOX9	MAILBOX9_IRQ_USER0	IRQ_CROSSBAR_265	–	MAILBOX9 user 0 interrupt request.
	MAILBOX9_IRQ_USER1	IRQ_CROSSBAR_266	–	MAILBOX9 user 1 interrupt request.
	MAILBOX9_IRQ_USER2	IRQ_CROSSBAR_267	–	MAILBOX9 user 2 interrupt request.

Table 19-4. MAILBOX Hardware Requests (continued)

	MAILBOX9_IRQ_USER3	IRQ_CROSSBAR_268	–	MAILBOX9 user 3 interrupt request.
MAILBOX10	MAILBOX10_IRQ_USER0	IRQ_CROSSBAR_269	–	MAILBOX10 user 0 interrupt request.
	MAILBOX10_IRQ_USER1	IRQ_CROSSBAR_270	–	MAILBOX10 user 1 interrupt request.
	MAILBOX10_IRQ_USER2	IRQ_CROSSBAR_271	–	MAILBOX10 user 2 interrupt request.
	MAILBOX10_IRQ_USER3	IRQ_CROSSBAR_272	–	MAILBOX10 user 3 interrupt request.
MAILBOX11	MAILBOX11_IRQ_USER0	IRQ_CROSSBAR_273	–	MAILBOX11 user 0 interrupt request.
	MAILBOX11_IRQ_USER1	IRQ_CROSSBAR_274	–	MAILBOX11 user 1 interrupt request.
	MAILBOX11_IRQ_USER2	IRQ_CROSSBAR_275	–	MAILBOX11 user 2 interrupt request.
	MAILBOX11_IRQ_USER3	IRQ_CROSSBAR_276	–	MAILBOX11 user 3 interrupt request.
MAILBOX12	MAILBOX12_IRQ_USER0	IRQ_CROSSBAR_277	–	MAILBOX12 user 0 interrupt request.
	MAILBOX12_IRQ_USER1	IRQ_CROSSBAR_278	–	MAILBOX12 user 1 interrupt request.
	MAILBOX12_IRQ_USER2	IRQ_CROSSBAR_279	–	MAILBOX12 user 2 interrupt request.
	MAILBOX12_IRQ_USER3	IRQ_CROSSBAR_280	–	MAILBOX12 user 3 interrupt request.
MAILBOX13	MAILBOX13_IRQ_USER0	IRQ_CROSSBAR_379	–	MAILBOX13 user 0 interrupt request.
	MAILBOX13_IRQ_USER1	IRQ_CROSSBAR_380	–	MAILBOX13 user 1 interrupt request.
	MAILBOX13_IRQ_USER2	IRQ_CROSSBAR_381	–	MAILBOX13 user 2 interrupt request.
	MAILBOX13_IRQ_USER3	IRQ_CROSSBAR_382	–	MAILBOX13 user 3 interrupt request.
		No DMA Requests		

Note

The "**Default Mapping**" column in Table 19-4, Table 19-7 , and Table 19-10 shows the default mapping of module interrupts. These interrupts can also be mapped to other input lines of each device interrupt controller through the IRQ_CROSSBAR module.

For more information about the IRQ_CROSSBAR module, see *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For information about interrupt source description, see *Mailbox Interrupt Requests*.

19.2.2 IVA Mailbox Integration

Figure 19-3 shows the IVA mailbox integration.

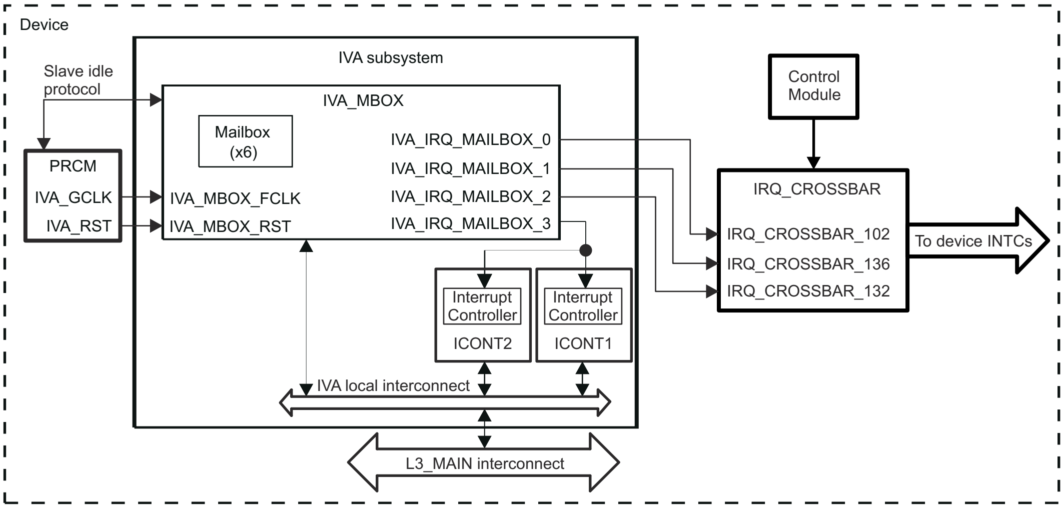


Figure 19-3. IVA Mailbox Integration

Note

The IVA_IRQ_MAILBOX_3 interrupt is mapped on both ICONT1 and ICONT2 interrupt controllers. It is recommended that this interrupt is unmasked at only one one interrupt controller at a time. For more information, see *IVA Imaging Controller*.

Table 19-5 through Table 19-7 summarize the IVA_MBOX integration in the device.

Table 19-5. IVA_MBOX Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
IVA_MBOX	PD_IVA	N/A	IVA local interconnect

Table 19-6. IVA_MBOX Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
IVA_MBOX	IVA_MBOX_FCLK	IVA_GCLK	PRCM	IVA_MBOX interface clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
IVA_MBOX	IVA_MBOX_RST	IVA_RST	PRCM	IVA_MBOX hardware reset. This reset is asynchronously applied to the IVA_MBOX registers.

Table 19-7. IVA_MBOX Hardware Requests

Interrupt Requests				
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**
IVA_MBOX	IVA_IRQ_MAILBOX_0	IRQ_CROSSBAR_102	MPU_IRQ_107	IVA_MBOX user 0 interrupt request.
	IVA_IRQ_MAILBOX_1	IRQ_CROSSBAR_136	–	IVA_MBOX user 1 interrupt request.
	IVA_IRQ_MAILBOX_2	IRQ_CROSSBAR_132	IPU1_IRQ_38; IPU2_IRQ_38	IVA_MBOX user 2 interrupt request.
	IVA_IRQ_MAILBOX_3	–	ICONT1_IRQ_11; ICONT2_IRQ_11	IVA_MBOX user 3 interrupt request.

Table 19-7. IVA_MBOX Hardware Requests (continued)

No DMA Requests

19.2.3 EVE Mailbox Integration

Figure 19-4 and Figure 19-5 show the EVE1_MBOX and EVE2_MBOX integration, respectively.

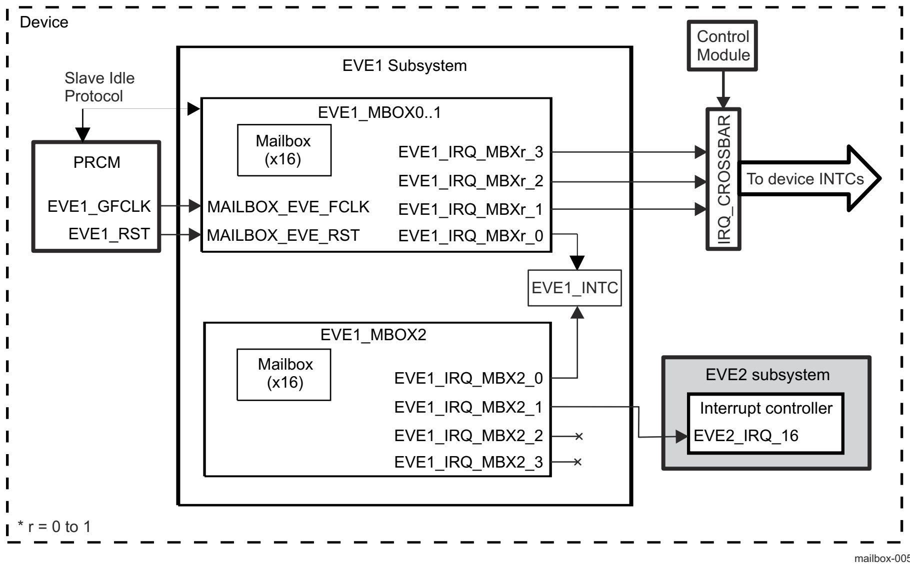


Figure 19-4. EVE1_MBOX Integration

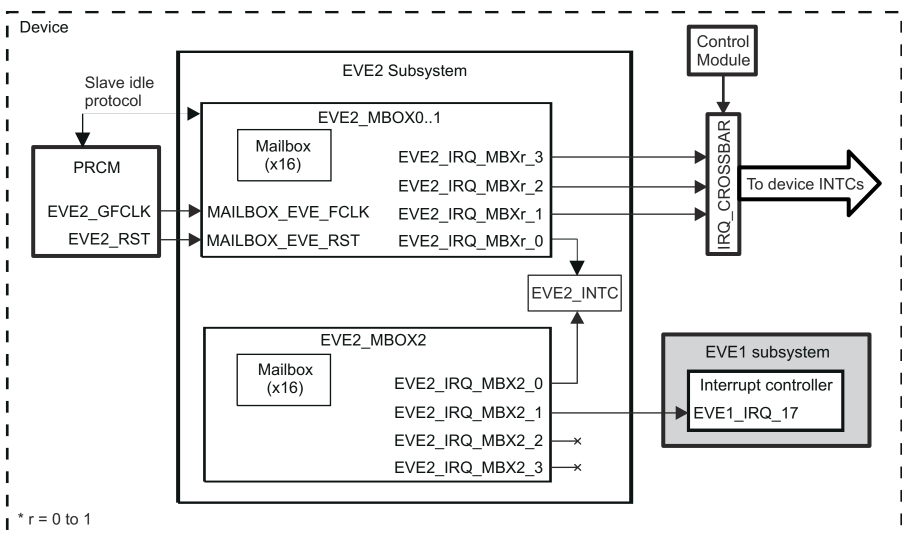


Figure 19-5. EVE2_MBOX Integration

Table 19-8 through Table 19-10 summarize the EVEx_MBOX(0..2) integration in the device.

Table 19-8. EVEx_MBOX Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
EVE1_MBOX	PD_EVE1	N/A	EVE1 local interconnect
EVE2_MBOX	PD_EVE2	N/A	EVE2 local interconnect

Table 19-9. EVEx_MBOX Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EVE1_MBOX	MAILBOX_EVE_FCLK	EVE1_GCLK	PRCM	EVE1_MBOX interface clock. This clock is used for all interface and functional operations.
EVE2_MBOX	MAILBOX_EVE_FCLK	EVE2_GCLK	PRCM	EVE2_MBOX interface clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EVE1_MBOX	MAILBOX_EVE_RST	EVE1_RST	PRCM	EVE1_MBOX hardware reset. This reset is asynchronously applied to the EVE1_MBOX registers.
EVE2_MBOX	MAILBOX_EVE_RST	EVE2_RST	PRCM	EVE2_MBOX hardware reset. This reset is asynchronously applied to the EVE2_MBOX registers.

Table 19-10. EVEx_MBOX Hardware Requests

Interrupt Requests				
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**
EVE1_MBOX0	EVE1_IRQ_MBX0_USER0	TBD	–	EVE1_MBOX0 user 0 interrupt request.
	EVE1_IRQ_MBX0_USER1	IRQ_CROSSBAR_284	–	EVE1_MBOX0 user 1 interrupt request.
	EVE1_IRQ_MBX0_USER2	IRQ_CROSSBAR_285	–	EVE1_MBOX0 user 2 interrupt request.
	EVE1_IRQ_MBX0_USER3	IRQ_CROSSBAR_286	–	EVE1_MBOX0 user 3 interrupt request.
EVE1_MBOX1	EVE1_IRQ_MBX1_USER0	TBD	–	EVE1_MBOX1 user 0 interrupt request.
	EVE1_IRQ_MBX1_USER1	IRQ_CROSSBAR_287	–	EVE1_MBOX1 user 1 interrupt request.
	EVE1_IRQ_MBX1_USER2	IRQ_CROSSBAR_288	–	EVE1_MBOX1 user 2 interrupt request.
	EVE1_IRQ_MBX1_USER3	IRQ_CROSSBAR_289	–	EVE1_MBOX1 user 3 interrupt request.
EVE1_MBOX2	EVE1_IRQ_MBX2_USER0	–	EVE1_IRQ_28	EVE1_MBOX2 user 0 interrupt request.
	EVE1_IRQ_MBX2_USER1	–	EVE2_IRQ_16	EVE1_MBOX2 user 1 interrupt request.
	EVE1_IRQ_MBX2_USER2	–	Not connected	EVE1_MBOX2 user 2 interrupt request.
	EVE1_IRQ_MBX2_USER3	–	Not connected	EVE1_MBOX2 user 3 interrupt request.
EVE2_MBOX0	EVE2_IRQ_MBX0_USER0	TBD	–	EVE2_MBOX0 user 0 interrupt request.
	EVE2_IRQ_MBX0_USER1	IRQ_CROSSBAR_293	–	EVE2_MBOX0 user 1 interrupt request.
	EVE2_IRQ_MBX0_USER2	IRQ_CROSSBAR_294	–	EVE2_MBOX0 user 2 interrupt request.
	EVE2_IRQ_MBX0_USER3	IRQ_CROSSBAR_295	–	EVE2_MBOX0 user 3 interrupt request.
EVE2_MBOX1	EVE2_IRQ_MBX1_USER0	TBD	–	EVE2_MBOX1 user 0 interrupt request.

Table 19-10. EVEx_MBOX Hardware Requests (continued)

	EVE2_IRQ_MBX1_USER1	IRQ_CROSSBAR_296	–	EVE2_MBOX1 user 1 interrupt request.
	EVE2_IRQ_MBX1_USER2	IRQ_CROSSBAR_297	–	EVE2_MBOX1 user 2 interrupt request.
	EVE2_IRQ_MBX1_USER3	IRQ_CROSSBAR_298	–	EVE2_MBOX1 user 3 interrupt request.
EVE2_MBOX2	EVE2_IRQ_MBX2_USER0	–	EVE2_IRQ_28	EVE2_MBOX2 user 0 interrupt request.
	EVE2_IRQ_MBX2_USER1	–	EVE1_IRQ_17	EVE2_MBOX2 user 1 interrupt request.
	EVE2_IRQ_MBX2_USER2	–	Not connected	EVE2_MBOX2 user 2 interrupt request.
	EVE2_IRQ_MBX2_USER3	–	Not connected	EVE2_MBOX2 user 3 interrupt request.
		No DMA Requests		

19.3 Mailbox Functional Description

Note

The functionality of all mailbox instances in the device is the same and is described in this section.

Note

In this chapter, *u* is the user number and *m* is the mailbox number as follows:

- For MAILBOX1: *u* = 0 to 2 and *m* = 0 to 7;
- For MAILBOX2..13: *u* = 0 to 3 and *m* = 0 to 11;
- For IVA_MBOX: *u* = 0 to 3 and *m* = 0 to 5;
- For EVEx_MBOX: *u* = 0 to 3 and *m* = 0 to 15;

The mailbox module provides a means of communication through message queues among the users. The individual mailbox modules, or FIFOs, can associate (or de-associate) with any of the processors using the MAILBOX_IRQENABLE_SET_u (or MAILBOX_IRQENABLE_CLR_u) register.

Table 19-11 shows the potential users of the mailbox modules in the device.

Table 19-11. Mailbox Users in the Device

<table>
<tr><th colspan="3" rowspan="2">Mailbox Type</th><th colspan="4">Users</th></tr>
<tr><th>User 0</th><th>User 1</th><th>User 2</th><th>User 3</th></tr>
<tr><td rowspan="2">System Mailbox</td><td colspan="2">MAILBOX1</td><td colspan="3">Any of: MPU, DSP1, DSP2, IPU1, IPU2</td><td>–</td></tr>
<tr><td colspan="2">MAILBOX2..13</td><td colspan="4">Any of: MPU, DSP1, DSP2, IPU1, IPU2</td></tr>
<tr><td colspan="3">IVA Mailbox</td><td colspan="3">Any of: MPU, DSP1, DSP2, IPU1, IPU2</td><td>IVA local - ICONT, or ICONT2</td></tr>
<tr><td rowspan="3">EVEx Mailbox</td><td colspan="2">EVEx_MBOX0, EVEx_MBOX1</td><td>TBD</td><td colspan="3">Any of: MPU, DSP1, DSP2, IPU1, IPU2</td></tr>
<tr><td rowspan="2">EVEx_ MBOX2</td><td>EVE1_MBOX2</td><td>ARP32 of EVE1</td><td>ARP32 of EVE2</td><td colspan="2">–</td></tr>
<tr><td>EVE2_MBOX2</td><td>ARP32 of EVE2</td><td>ARP32 of EVE1</td><td colspan="2">–</td></tr>
</table>

Note

It is software responsibility to select a user by mapping (via IRQ_CROSSBAR) the corresponding mailbox interrupt to the interrupt controller of the appropriate processor subsystem.

Each user has a dedicated interrupt signal from the corresponding mailbox module instance and dedicated interrupt enabling and status registers.
Each MAILBOX_IRQSTATUS_RAW_u/MAILBOX_IRQSTATUS_CLR_u interrupt status register corresponds to a particular user.

19.3.1 Mailbox Block Diagram

19.3.1.1

Figure 19-6 shows the mailbox block diagram.

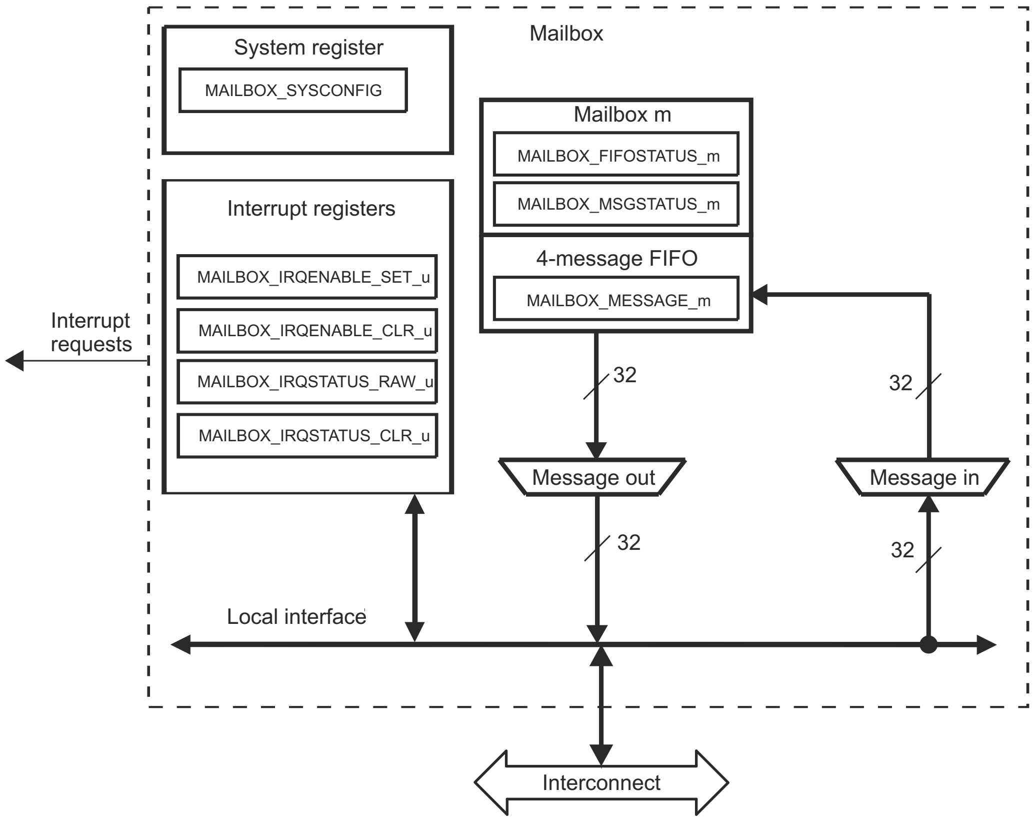


Figure 19-6. Mailbox Block Diagram

19.3.2 Mailbox Software Reset

The mailbox module supports a software reset through the MAILBOX_SYSCONFIG[0] SOFTRESET bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Reading the MAILBOX_SYSCONFIG[0] SOFTRESET bit gives the status of the software reset:

- Read 1: the software reset is on-going.
- Read 0: the software reset is complete.

The software must ensure that the software reset completes before doing mailbox operations.

19.3.3 Mailbox Power Management

Table 19-12 describes power-management features available for the mailbox module.

Table 19-12. Local Power Management Features

Feature	Registers	Description
Clock autogating	N/A	Feature not available
Slave idle modes	MAILBOX_SYSCONFIG[3:2] SIDLEMODE bit field	Force-idle, no-idle and smart-idle modes are available
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	N/A	Feature not available

The mailbox module can be configured using the MAILBOX_SYSCONFIG[3:2] SIDLEMODE bit field to one of the following acknowledgment modes:

- Force-idle mode (SIDLEMODE = 0x0): The mailbox module immediately enters the idle state on receiving a low-power-mode request from the PRCM module. In this mode, the software must ensure that there are no asserted output interrupts before requesting this mode to go into the idle state.
- No-idle mode (SIDLEMODE = 0x1): The mailbox module never enters the idle state.
- Smart-idle mode (SIDLEMODE = 0x2): After receiving a low-power-mode request from the PRCM module, the mailbox module enters the idle state only after all asserted output interrupts are acknowledged.

19.3.4 Mailbox Interrupt Requests

An interrupt request allows the user of the mailbox to be notified when a message is received or when the message queue is not full. There is one interrupt per user.

Table 19-13 lists the event flags, and their mask, that can cause module interrupts.

Table 19-13. Interrupt Events

Non-Maskable Event Flag[(1)]	Maskable Event Flag	Event Mask Bit	Event Unmask Bit	Description
MAILBOX_IRQSTATUS_RAW_u[0+m*2] NEWMSGSTATUSUUMBm	MAILBOX_IRQSTATUS_CLR_u[0+m*2] NEWMSGSTATUSUUMBm	MAILBOX_IRQENABLE_CLR_u[0+m*2] NEWMSGSTATUSUUMBm	MAILBOX_IRQENABLE_SET_u[0+m*2] NEWMSGSTATUSUUMBm	Mailbox *m* receives a new message.
MAILBOX_IRQSTATUS_RAW_u[1+m*2] NOTFULLSTATUSUUMBm	MAILBOX_IRQSTATUS_CLR_u[1+m*2] NOTFULLSTATUSUUMBm	MAILBOX_IRQENABLE_CLR_u[1+m*2] NOTFULLSTATUSUUMBm	MAILBOX_IRQENABLE_SET_u[1+m*2] NOTFULLSTATUSUUMBm	Mailbox *m* message queue is not full.

(1) MAILBOX_IRQSTATUS_RAW_u register is mostly used for debug purposes.

> **CAUTION**
>
> Once an event generating the interrupt request has been processed by the software, it must be cleared by writing a logical 1 in the corresponding bit of the MAILBOX_IRQSTATUS_CLR_u register.
>
> Writing a logical 1 in a bit of the MAILBOX_IRQSTATUS_CLR_u register will also clear to 0 the corresponding bit in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

An event can generate an interrupt request when a logical 1 is written to the corresponding unmask bit in the MAILBOX_IRQENABLE_SET_u register. Events are reported in the appropriate MAILBOX_IRQSTATUS_CLR_u and MAILBOX_IRQSTATUS_RAW_u registers.

An event stops generating interrupt requests when a logical 1 is written to the corresponding mask bit in the MAILBOX_IRQENABLE_CLR_u register. Events are only reported in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

In case of the MAILBOX_IRQSTATUS_RAW_u register, the event is reported in the corresponding bit even if the interrupt request generation is disabled for this event.

19.3.5 Mailbox Assignment

19.3.5.1 Description

To assign a receiver to a mailbox, set the new message interrupt enable bit corresponding to the desired mailbox in the MAILBOX_IRQENABLE_SET_u register. The receiver reads the MAILBOX_MESSAGE_m register to retrieve a message from the mailbox.

An alternate method for the receiver that does not use the interrupts is to poll the MAILBOX_FIFOSTATUS_m and/or MAILBOX_MSGSTATUS_m registers to know when to send or retrieve a message to or from the mailbox. This method does not require assigning a receiver to a mailbox. Because this method does not include the explicit assignment of the mailbox, the software must avoid having multiple receivers use the same mailbox, which can result in incoherency.

To assign a sender to a mailbox, set the queue-not-full interrupt enable bit of the desired mailbox in the MAILBOX_IRQENABLE_SET_u register, where *u* is the number of the sending user. However, direct allocation of a mailbox to a sender is not recommended because it can cause the sending processor to be constantly interrupted.

It is recommended that register polling be used to:

- Check the status of either the MAILBOX_FIFOSTATUS_m or MAILBOX_MSGSTATUS_m registers
- Write the message to the corresponding MAILBOX_MESSAGE_m register, if space is available.

The sender might use the queue-not-full interrupt when the initial mailbox status check indicates the mailbox is full. In this case, the sender can enable the queue-not-full interrupt for its mailbox in the appropriate MAILBOX_IRQENABLE_SET_u register. This allows the sender to be notified by interrupt only when a FIFO queue has at least one available entry.

Reading the MAILBOX_IRQSTATUS_CLR_u register determines the status of the new message and the queue-not-full interrupts for a particular user. Writing 1 to the corresponding bit in the MAILBOX_IRQSTATUS_CLR_u register acknowledges, and subsequently clears, an interrupt.

CAUTION

Assigning multiple senders or multiple receivers to the same mailbox is not recommended.

19.3.6 Sending and Receiving Messages

19.3.6.1 Description

When a 32-bit message is written to the MAILBOX_MESSAGE_m register, the message is appended into the FIFO queue. This queue holds four messages. If the queue is full, the message is discarded.

Queue overflow can be avoided by first reading the MAILBOX_FIFOSTATUS_m register to check that the mailbox message queue is not full before writing a new message to it.

Reading the MAILBOX_MESSAGE_m register returns the message at the beginning of the FIFO queue and removes it from the queue. If the FIFO queue is empty when the MAILBOX_MESSAGE_m register is read, the value 0 is returned.

The new message interrupt is asserted when at least one message is in the mailbox message FIFO queue. To determine the number of messages in the mailbox message FIFO queue, read the MAILBOX_MSGSTATUS_m register.

19.3.7 16-Bit Register Access

19.3.7.1 Description

So that 16-bit processors can access the mailbox module, the module allows 16-bit register read and write access, with restrictions for the MAILBOX_MESSAGE_m registers. The 16-bit half-words are organized in little endian fashion; that is, the least-significant 16 bits are at the low address and the most-significant 16 bits are at the high address (low address + 0x02).

All mailbox module registers can be read or written to directly using individual 16-bit accesses with no restriction on interleaving, except the MAILBOX_MESSAGE_m registers, which must always be accessed by either single 32-bit accesses or two consecutive 16-bit accesses.

CAUTION

When using 16-bit accesses to the MAILBOX_MESSAGE_m registers, the order of access must be the least-significant half-word first (low address) and the most-significant half-word last (high address). This requirement is because of the update operation by the message FIFO of the MAILBOX_MSGSTATUS_m registers. The update of the FIFO queue contents and the associated status registers and possible interrupt generation occurs only when the most-significant 16 bits of a MAILBOX_MESSAGE_m are accessed.

19.3.8 Example of Communication

Figure 19-7 shows an example of communication between MPU and DSP1 subsystems.

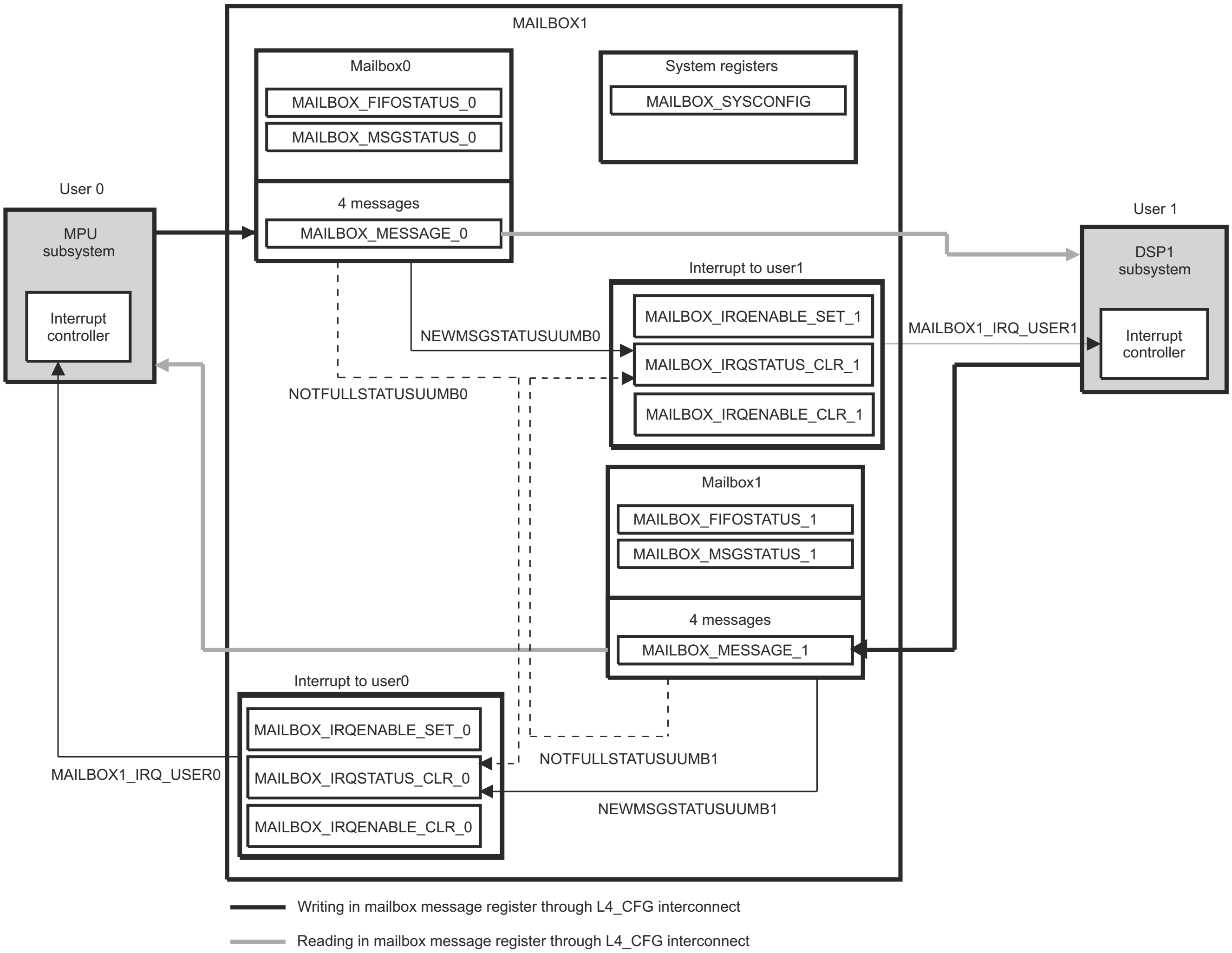


Figure 19-7. Example of Communication

19.4 Mailbox Programming Guide

19.4.1 Mailbox Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the mailbox module.

19.4.1.1 Global Initialization

19.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the mailbox module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the mailbox.

See Section 19.2, *Mailbox Integration*, for further information.

Table 19-14. Global Initialization of Surrounding Modules for MAILBOX

Surrounding Modules	Comments
PRCM	MAILBOX functional/interface clock must be enabled.
Interrupt Controllers	MPU, or IPU1, or IPU2, or DSP1, or DSP2 interrupt controller must be configured to enable the interrupt request generation to the corresponding subsystem.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

Table 19-15. Global Initialization of Surrounding Modules for IVA_MBOX

Surrounding Modules	Comments
PRCM	IVA_MBOX functional/interface clock must be enabled.
Interrupt Controllers	MPU, or IPU1, or IPU2, or DSP1, or DSP2, or IVA ICONT1/ICONT2 interrupt controller must be configured to enable the interrupt request generation to the corresponding subsystem.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

Table 19-16. Global Initialization of Surrounding Modules for EVEx_MBOX

Surrounding Modules	Comments
PRCM	EVEx_MBOX functional/interface clock must be enabled.
Interrupt Controllers	MPU, or IPU1, or IPU2, or DSP1, or DSP2, or EVE interrupt controller must be configured to enable the interrupt request generation to the corresponding subsystem.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

19.4.1.1.2 Mailbox Global Initialization

19.4.1.1.2.1 Main Sequence - Mailbox Global Initialization

This procedure initializes the mailbox module after a power-on or software reset.

Table 19-17. Mailbox Global Initialization

Step	Register/ Bit Field / Programming Model	Value
Perform a software reset	MAILBOX_SYSCONFIG[0] SOFTRESET	0x1
Wait until reset is complete	MAILBOX_SYSCONFIG[0] SOFTRESET	= 0x0
Set idle mode configuration	MAILBOX_SYSCONFIG[3:2] SIDLEMODE	0x-

19.4.1.2 Mailbox Operational Modes Configuration

19.4.1.2.1 Mailbox Processing modes

19.4.1.2.1.1 Main Sequence - Sending a Message (Polling Method)

Table 19-18. Sending a Message (Polling Method)

Step	Register/ Bit Field / Programming Model	Value
IF : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x1
Wait until at least one message slot is available	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x0
ELSE		
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ENDIF		

19.4.1.2.1.2 Main Sequence - Sending a Message (Interrupt Method)

Table 19-19. Sending a Message (Interrupt Method)

Step	Register/ Bit Field / Programming Model	Value
IF : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x1
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[1+ m*2]	0x1
User (processor) can perform another task until interrupt occurs See Section 19.4.1.3.1 for interrupt handling in sending mode		
ELSE		
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ENDIF		

19.4.1.2.1.3 Main Sequence - Receiving a Message (Polling Method)

Table 19-20. Receiving a Message (Polling Method)

Step	Register/ Bit Field / Programming Model	Value
IF : Number of messages is not equal to 0	MAILBOX_MSGSTATUS_m[2:0] NBOFMSGMB	!= 0x0
Read message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ENDIF		

19.4.1.2.1.4 Main Sequence - Receiving a Message (Interrupt Method)

Table 19-21. Receiving a Message (Interrupt Method)

Step	Register/ Bit Field / Programming Model	Value
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[0 + m*2]	0x1
User (processor) can perform another task until interrupt occurs See Section 19.4.1.3.2 for interrupt handling in receiving mode		

19.4.1.3 Mailbox Events Servicing

19.4.1.3.1 Events Servicing in Sending Mode

Table 19-22 describes the events servicing in sending mode.

Table 19-22. Events Servicing in Sending Mode

Step	Register/ Bit Field / Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	0x1

Table 19-22. Events Servicing in Sending Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	0x1

19.4.1.3.2 Events Servicing in Receiving Mode

Table 19-23 describes the events servicing in receiving mode.

Table 19-23. Events Servicing in Receiving Mode

Step	Register/ Bit Field / Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	0x1
IF : Number of messages is not equal to 0 ?	MAILBOX_MSGSTATUS_m[2:0] NBOFMSGMB	!= 0x0
Read message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ELSE		
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	0x1
ENDIF		

19.5 Mailbox Register Manual

19.5.1 Mailbox Instance Summary

Table 19-24. Mailbox Instance Summary

Module Name	L3_MAIN Base Address	L4_CFG Base Address	L4_PER3 Base Address	Size
MAILBOX1	–	0x4A0F 4000	–	4 KiB
MAILBOX2	–	–	0x4883 A000	4 KiB
MAILBOX3	–	–	0x4883 C000	4 KiB
MAILBOX4	–	–	0x4883 E000	4 KiB
MAILBOX5	–	–	0x4884 0000	4 KiB
MAILBOX6	–	–	0x4884 2000	4 KiB
MAILBOX7	–	–	0x4884 4000	4 KiB
MAILBOX8	–	–	0x4884 6000	4 KiB
MAILBOX9	–	–	0x4885 E000	4 KiB
MAILBOX10	–	–	0x4886 0000	4 KiB
MAILBOX11	–	–	0x4886 2000	4 KiB
MAILBOX12	–	–	0x4886 4000	4 KiB
MAILBOX13	–	–	0x4880 2000	4 KiB
IVA_MBOX	0x5A05 A800	–	–	4 KiB
EVE1_MBOX0	0x4208 B000	–	–	4 KiB
EVE1_MBOX1	0x4208 C000	–	–	4 KiB
EVE1_MBOX2	0x4208 D000	–	–	4 KiB
EVE2_MBOX0	0x4218 B000	–	–	4 KiB
EVE2_MBOX1	0x4218 C000	–	–	4 KiB
EVE2_MBOX2	0x4218 D000	–	–	4 KiB

19.5.2 Mailbox Registers

19.5.2.1 Mailbox Register Summary

Table 19-25. MAILBOX Registers Mapping Summary (1/5)

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX1 L4_CFG Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4A0F 4000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4A0F 4010
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x4A0F 4040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x4A0F 4080 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x4A0F 40C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u (2)	RW	32	0x0000 0100 + (0x10 * u)	0x4A0F 4100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u (2)	RW	32	0x0000 0104 + (0x10 * u)	0x4A0F 4104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u (2)	RW	32	0x0000 0108 + (0x10 * u)	0x4A0F 4108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u (2)	RW	32	0x0000 010C + (0x10 * u)	0x4A0F 410C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4A0F 4140

(1) m = 0 to 7
(2) u = 0 to 2

Table 19-26. MAILBOX Registers Mapping Summary (2/5)

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX2 L4_PER3 Physical Address	MAILBOX3 L4_PER3 Physical Address	MAILBOX4 L4_PER3 Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4883 A000	0x4883 C000	0x4883 E000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4883 A010	0x4883 C010	0x4883 E010
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x4883 A040 + (0x4 * m)	0x4883 C040 + (0x4 * m)	0x4883 E040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x4883 A080 + (0x4 * m)	0x4883 C080 + (0x4 * m)	0x4883 E080 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x4883 A0C0 + (0x4 * m)	0x4883 C0C0 + (0x4 * m)	0x4883 E0C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u (2)	RW	32	0x0000 0100 + (0x10 * u)	0x4883 A100 + (0x10 * u)	0x4883 C100 + (0x10 * u)	0x4883 E100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u (2)	RW	32	0x0000 0104 + (0x10 * u)	0x4883 A104 + (0x10 * u)	0x4883 C104 + (0x10 * u)	0x4883 E104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u (2)	RW	32	0x0000 0108 + (0x10 * u)	0x4883 A108 + (0x10 * u)	0x4883 C108 + (0x10 * u)	0x4883 E108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u (2)	RW	32	0x0000 010C + (0x10 * u)	0x4883 A10C + (0x10 * u)	0x4883 C10C + (0x10 * u)	0x4883 E10C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4883 A140	0x4883 C140	0x4883 E140

(1) m = 0 to 11
(2) u = 0 to 3

Table 19-27. MAILBOX Registers Mapping Summary (3/5)

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX5 L4_PER3 Physical Address	MAILBOX6 L4_PER3 Physical Address	MAILBOX7 L4_PER3 Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4884 0000	0x4884 2000	0x4884 4000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4884 0010	0x4884 2010	0x4884 4010
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x4884 0040 + (0x4 * m)	0x4884 2040 + (0x4 * m)	0x4884 4040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x4884 0080 + (0x4 * m)	0x4884 2080 + (0x4 * m)	0x4884 4080 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x4884 00C0 + (0x4 * m)	0x4884 20C0 + (0x4 * m)	0x4884 40C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u (2)	RW	32	0x0000 0100 + (0x10 * u)	0x4884 0100 + (0x10 * u)	0x4884 2100 + (0x10 * u)	0x4884 4100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u (2)	RW	32	0x0000 0104 + (0x10 * u)	0x4884 0104 + (0x10 * u)	0x4884 2104 + (0x10 * u)	0x4884 4104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u (2)	RW	32	0x0000 0108 + (0x10 * u)	0x4884 0108 + (0x10 * u)	0x4884 2108 + (0x10 * u)	0x4884 4108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u (2)	RW	32	0x0000 010C + (0x10 * u)	0x4884 010C + (0x10 * u)	0x4884 210C + (0x10 * u)	0x4884 410C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4884 0140	0x4884 2140	0x4884 4140

(1) m = 0 to 11
(2) u = 0 to 3

Table 19-28. MAILBOX Registers Mapping Summary (4/5)

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX8 L4_PER3 Physical Address	MAILBOX9 L4_PER3 Physical Address	MAILBOX10 L4_PER3 Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4884 6000	0x4885 E000	0x4886 0000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4884 6010	0x4885 E010	0x4886 0010

Table 19-28. MAILBOX Registers Mapping Summary (4/5) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX8 L4_PER3 Physical Address	MAILBOX9 L4_PER3 Physical Address	MAILBOX10 L4_PER3 Physical Address
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x4884 6040 + (0x4 * m)	0x4885 E040 + (0x4 * m)	0x4886 0040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x4884 6080 + (0x4 * m)	0x4885 E080 + (0x4 * m)	0x4886 0080 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x4884 60C0 + (0x4 * m)	0x4885 E0C0 + (0x4 * m)	0x4886 00C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u (2)	RW	32	0x0000 0100 + (0x10 * u)	0x4884 6100 + (0x10 * u)	0x4885 E100 + (0x10 * u)	0x4886 0100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u (2)	RW	32	0x0000 0104 + (0x10 * u)	0x4884 6104 + (0x10 * u)	0x4885 E104 + (0x10 * u)	0x4886 0104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u (2)	RW	32	0x0000 0108 + (0x10 * u)	0x4884 6108 + (0x10 * u)	0x4885 E108 + (0x10 * u)	0x4886 0108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u (2)	RW	32	0x0000 010C + (0x10 * u)	0x4884 610C + (0x10 * u)	0x4885 E10C + (0x10 * u)	0x4886 010C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4884 6140	0x4885 E140	0x4886 0140

(1) m = 0 to 11
(2) u = 0 to 3

Table 19-29. MAILBOX Registers Mapping Summary (5/5)

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX11 L4_PER3 Physical Address	MAILBOX12 L4_PER3 Physical Address	MAILBOX13 L4_PER3 Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4886 2000	0x4886 4000	0x4880 2000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4886 2010	0x4886 4010	0x4880 2010
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x4886 2040 + (0x4 * m)	0x4886 4040 + (0x4 * m)	0x4880 2040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x4886 2080 + (0x4 * m)	0x4886 4080 + (0x4 * m)	0x4880 2080 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x4886 20C0 + (0x4 * m)	0x4886 40C0 + (0x4 * m)	0x4880 20C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u (2)	RW	32	0x0000 0100 + (0x10 * u)	0x4886 2100 + (0x10 * u)	0x4886 4100 + (0x10 * u)	0x4880 2100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u (2)	RW	32	0x0000 0104 + (0x10 * u)	0x4886 2104 + (0x10 * u)	0x4886 4104 + (0x10 * u)	0x4880 2104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u (2)	RW	32	0x0000 0108 + (0x10 * u)	0x4886 2108 + (0x10 * u)	0x4886 4108 + (0x10 * u)	0x4880 2108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u (2)	RW	32	0x0000 010C + (0x10 * u)	0x4886 210C + (0x10 * u)	0x4886 410C + (0x10 * u)	0x4880 210C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4886 2140	0x4886 4140	0x4880 2140

(1) m = 0 to 11
(2) u = 0 to 3

Table 19-30. IVA_MBOX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IVA_MBOX L3_MAIN Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x5A05 A800
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x5A05 A810
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x5A05 A840 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x5A05 A880 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x5A05 A8C0 + (0x4 * m)

Table 19-30. IVA_MBOX Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IVA_MBOX L3_MAIN Physical Address
MAILBOX_IRQSTATUS_RAW_u [2]	RW	32	0x0000 0100 + (0x10 * u)	0x5A05 A900 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u [2]	RW	32	0x0000 0104 + (0x10 * u)	0x5A05 A904 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u [2]	RW	32	0x0000 0108 + (0x10 * u)	0x5A05 A908 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u [2]	RW	32	0x0000 010C + (0x10 * u)	0x5A05 A90C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x5A05 A940

(1) m = 0 to 5
(2) u = 0 to 3

Table 19-31. EVE1_MBOX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_MBOX0 L3_MAIN Physical Address	EVE1_MBOX1 L3_MAIN Physical Address	EVE1_MBOX2 L3_MAIN Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4208 B000	0x4208 C000	0x4208 D000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4208 B010	0x4208 C010	0x4208 D010
MAILBOX_MESSAGE_m [1]	RW	32	0x0000 0040 + (0x4 * m)	0x4208 B040 + (0x4 * m)	0x4208 C040 + (0x4 * m)	0x4208 D040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m [1]	R	32	0x0000 0080 + (0x4 * m)	0x4208 B080 + (0x4 * m)	0x4208 C080 + (0x4 * m)	0x4208 D080 + (0x4 * m)
MAILBOX_MSGSTATUS_m [1]	R	32	0x0000 00C0 + (0x4 * m)	0x4208 B0C0 + (0x4 * m)	0x4208 C0C0 + (0x4 * m)	0x4208 D0C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u [2]	RW	32	0x0000 0100 + (0x10 * u)	0x4208 B100 + (0x10 * u)	0x4208 C100 + (0x10 * u)	0x4208 D100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u [2]	RW	32	0x0000 0104 + (0x10 * u)	0x4208 B104 + (0x10 * u)	0x4208 C104 + (0x10 * u)	0x4208 D104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u [2]	RW	32	0x0000 0108 + (0x10 * u)	0x4208 B108 + (0x10 * u)	0x4208 C108 + (0x10 * u)	0x4208 D108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u [2]	RW	32	0x0000 010C + (0x10 * u)	0x4208 B10C + (0x10 * u)	0x4208 C10C + (0x10 * u)	0x4208 D10C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4208 B140	0x4208 C140	0x4208 D140

(1) m = 0 to 15
(2) u = 0 to 3

Table 19-32. EVE2_MBOX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE2_MBOX0 L3_MAIN Physical Address	EVE2_MBOX1 L3_MAIN Physical Address	EVE2_MBOX2 L3_MAIN Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4218 B000	0x4218 C000	0x4218 D000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4218 B010	0x4218 C010	0x4218 D010
MAILBOX_MESSAGE_m [1]	RW	32	0x0000 0040 + (0x4 * m)	0x4218 B040 + (0x4 * m)	0x4218 C040 + (0x4 * m)	0x4218 D040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m [1]	R	32	0x0000 0080 + (0x4 * m)	0x4218 B080 + (0x4 * m)	0x4218 C080 + (0x4 * m)	0x4218 D080 + (0x4 * m)
MAILBOX_MSGSTATUS_m [1]	R	32	0x0000 00C0 + (0x4 * m)	0x4218 B0C0 + (0x4 * m)	0x4218 C0C0 + (0x4 * m)	0x4218 D0C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u [2]	RW	32	0x0000 0100 + (0x10 * u)	0x4218 B100 + (0x10 * u)	0x4218 C100 + (0x10 * u)	0x4218 D100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u [2]	RW	32	0x0000 0104 + (0x10 * u)	0x4218 B104 + (0x10 * u)	0x4218 C104 + (0x10 * u)	0x4218 D104 + (0x10 * u)

Table 19-32. EVE2_MBOX Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE2_MBOX0 L3_MAIN Physical Address	EVE2_MBOX1 L3_MAIN Physical Address	EVE2_MBOX2 L3_MAIN Physical Address
MAILBOX_IRQENABLE_SET_u (2)	RW	32	0x0000 0108 + (0x10 * u)	0x4218 B108 + (0x10 * u)	0x4218 C108 + (0x10 * u)	0x4218 D108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u (2)	RW	32	0x0000 010C + (0x10 * u)	0x4218 B10C + (0x10 * u)	0x4218 C10C + (0x10 * u)	0x4218 D10C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4218 B140	0x4218 C140	0x4218 D140

(1) m = 0 to 15
(2) u = 0 to 3

19.5.2.2 Mailbox Register Description

Table 19-33. MAILBOX_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A0F 4000 0x5A05 A800	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	This register contains the IP revision code		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

Table 19-34. MAILBOX_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A0F 4010 0x5A05 A810	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	This register controls the various parameters of the communication interface		
Type	RW		

31–4	3–2	1	0
RESERVED	SIDLEMODE	RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW	0x0000000
3:2	SIDLEMODE	Idle Mode 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module based on the internal activity of the module 0x3: reserved do not use	RW	0x2
1	RESERVED	Reserved	RW	0

Bits	Field Name	Description	Type	Reset
0	SOFTRESET	Softreset Read 0x0: Soft/Hard reset done Write 0x0: No action Read 0x1: Reset is ongoing Write 0x1: Start the soft reset sequence	RW	0

Table 19-35. MAILBOX_MESSAGE_m

Address Offset	0x0000 0040 + (0x4 * m)	**Index**	m = 0 to 7 (MAILBOX1), or m = 0 to 11 (MAILBOX2..13), or m = 0 to 5 (IVA_MBOX), or m = 0 to 15 (EVEx_MBOX)
Physical Address	0x4A0F 4040 + (0x4 * m) 0x5A05 A840 + (0x4 * m)	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.		
Type	RW		

31–0
MESSAGEVALUEMBM

Bits	Field Name	Description	Type	Reset
31:0	MESSAGEVALUEMBM	Message in Mailbox	RW	0x0000 0000

Table 19-36. MAILBOX_FIFOSTATUS_m

Address Offset	0x0000 0080 + (0x4 * m)	**Index**	m = 0 to 7 (MAILBOX1), or m = 0 to 11 (MAILBOX2..13), or m = 0 to 5 (IVA_MBOX), or m = 0 to 15 (EVEx_MBOX)
Physical Address	0x4A0F 4080 + (0x4 * m) 0x5A05 A880 + (0x4 * m)	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	The FIFO status register has the status related to the mailbox internal FIFO		
Type	R		

31–1	0
RESERVED	FIFOFULLMBM

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x0000 0000
0	FIFOFULLMBM	Full flag for Mailbox Read 0x0: Mailbox FIFO is not full Read 0x1: Mailbox FIFO is full	R	0

Table 19-37. MAILBOX_MSGSTATUS_m

Address Offset	0x0000 00C0 + (0x4 * m)	**Index**	m = 0 to 7 (MAILBOX1), or m = 0 to 11 (MAILBOX2..13), or m = 0 to 5 (IVA_MBOX), or m = 0 to 15 (EVEx_MBOX)
Physical Address	0x4A0F 40C0 + (0x4 * m) 0x5A05 A8C0 + (0x4 * m)	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3

Table 19-37. MAILBOX_MSGSTATUS_m (continued)

Description	The message status register has the status of the messages in the mailbox.
Type	R

31–3	2–0
RESERVED	NBOFMSGMBM

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved. Read returns 0	R	0x0000 0000
2:0	NBOFMSGMBM	Number of unread messages in Mailbox Note: Limited to four messages per mailbox.	R	0x00

Table 19-38. MAILBOX_IRQSTATUS_RAW_u

Address Offset	0x0000 0100 + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2..13, IVA_MBOX, EVEx_MBOX)
Physical Address	0x4A0F 4100 + (0x10 * u) 0x5A05 A900 + (0x10 * u)	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	The interrupt status register has the raw status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit sets this bit. This register is mainly used for debug purpose.		
Type	RW		

31	30	29	28	27	26	25	24
NOTFULLSTATUSUUMB15	NEWMSGSTATUSUUMB15	NOTFULLSTATUSUUMB14	NEWMSGSTATUSUUMB14	NOTFULLSTATUSUUMB13	NEWMSGSTATUSUUMB13	NOTFULLSTATUSUUMB12	NEWMSGSTATUSUUMB12

23	22	21	20	19	18	17	16
NOTFULLSTATUSUUMB11	NEWMSGSTATUSUUMB11	NOTFULLSTATUSUUMB10	NEWMSGSTATUSUUMB10	NOTFULLSTATUSUUMB9	NEWMSGSTATUSUUMB9	NOTFULLSTATUSUUMB8	NEWMSGSTATUSUUMB8

15	14	13	12	11	10	9	8
NOTFULLSTATUSUUMB7	NEWMSGSTATUSUUMB7	NOTFULLSTATUSUUMB6	NEWMSGSTATUSUUMB6	NOTFULLSTATUSUUMB5	NEWMSGSTATUSUUMB5	NOTFULLSTATUSUUMB4	NEWMSGSTATUSUUMB4

7	6	5	4	3	2	1	0
NOTFULLSTATUSUUMB3	NEWMSGSTATUSUUMB3	NOTFULLSTATUSUUMB2	NEWMSGSTATUSUUMB2	NOTFULLSTATUSUUMB1	NEWMSGSTATUSUUMB1	NOTFULLSTATUSUUMB0	NEWMSGSTATUSUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLSTATUSUUMB15	NotFull Status bit for User u, Mailbox 15 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
30	NEWMSGSTATUSUUMB15	NewMessage Status bit for User u, Mailbox 15 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
29	NOTFULLSTATUSUUMB14	NotFull Status bit for User u, Mailbox 14 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1

Bits	Field Name	Description	Type	Reset
28	NEWMSGSTATUSUUMB14	NewMessage Status bit for User u, Mailbox 14 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
27	NOTFULLSTATUSUUMB13	NotFull Status bit for User u, Mailbox 13 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
26	NEWMSGSTATUSUUMB13	NewMessage Status bit for User u, Mailbox 13 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
25	NOTFULLSTATUSUUMB12	NotFull Status bit for User u, Mailbox 12 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
24	NEWMSGSTATUSUUMB12	NewMessage Status bit for User u, Mailbox 12 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
23	NOTFULLSTATUSUUMB11	NotFull Status bit for User u, Mailbox 11 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
22	NEWMSGSTATUSUUMB11	NewMessage Status bit for User u, Mailbox 11 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
21	NOTFULLSTATUSUUMB10	NotFull Status bit for User u, Mailbox 10 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
20	NEWMSGSTATUSUUMB10	NewMessage Status bit for User u, Mailbox 10 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0

Bits	Field Name	Description	Type	Reset
19	NOTFULLSTATUSUUMB9	NotFull Status bit for User u, Mailbox 9 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
18	NEWMSGSTATUSUUMB9	NewMessage Status bit for User u, Mailbox 9 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
17	NOTFULLSTATUSUUMB8	NotFull Status bit for User u, Mailbox 8 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
16	NEWMSGSTATUSUUMB8	NewMessage Status bit for User u, Mailbox 8 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
15	NOTFULLSTATUSUUMB7	NotFull Status bit for User u, Mailbox 7 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
14	NEWMSGSTATUSUUMB7	NewMessage Status bit for User u, Mailbox 7 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
13	NOTFULLSTATUSUUMB6	NotFull Status bit for User u, Mailbox 6 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
12	NEWMSGSTATUSUUMB6	NewMessage Status bit for User u, Mailbox 6 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
11	NOTFULLSTATUSUUMB5	NotFull Status bit for User u, Mailbox 5 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1

Bits	Field Name	Description	Type	Reset
10	NEWMSGSTATUSUUMB5	NewMessage Status bit for User u, Mailbox 5 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
9	NOTFULLSTATUSUUMB4	NotFull Status bit for User u, Mailbox 4 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
8	NEWMSGSTATUSUUMB4	NewMessage Status bit for User u, Mailbox 4 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
7	NOTFULLSTATUSUUMB3	NotFull Status bit for User u, Mailbox 3 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
6	NEWMSGSTATUSUUMB3	NewMessage Status bit for User u, Mailbox 3 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
5	NOTFULLSTATUSUUMB2	NotFull Status bit for User u, Mailbox 2 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
4	NEWMSGSTATUSUUMB2	NewMessage Status bit for User u, Mailbox 2 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
3	NOTFULLSTATUSUUMB1	NotFull Status bit for User u, Mailbox 1 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
2	NEWMSGSTATUSUUMB1	NewMessage Status bit for User u, Mailbox 1 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0

Bits	Field Name	Description	Type	Reset
1	NOTFULLSTATUSUUMB0	NotFull Status bit for User u, Mailbox 0 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
0	NEWMSGSTATUSUUMB0	NewMessage Status bit for User u, Mailbox 0 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0

Table 19-39. MAILBOX_IRQSTATUS_CLR_u

Address Offset	0x0000 0104 + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2..13, IVA_MBOX, EVEx_MBOX)
Physical Address	0x4A0F 4104 + (0x10 * u) 0x5A05 A904 + (0x10 * u)	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOTFULLSTATUSENUUMB15	NEWMSGSTATUSENUUMB15	NOTFULLSTATUSENUUMB14	NEWMSGSTATUSENUUMB14	NOTFULLSTATUSENUUMB13	NEWMSGSTATUSENUUMB13	NOTFULLSTATUSENUUMB12	NEWMSGSTATUSENUUMB12	NOTFULLSTATUSENUUMB11	NEWMSGSTATUSENUUMB11	NOTFULLSTATUSENUUMB10	NEWMSGSTATUSENUUMB10	NOTFULLSTATUSENUUMB9	NEWMSGSTATUSENUUMB9	NOTFULLSTATUSENUUMB8	NEWMSGSTATUSENUUMB8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTFULLSTATUSENUUMB7	NEWMSGSTATUSENUUMB7	NOTFULLSTATUSENUUMB6	NEWMSGSTATUSENUUMB6	NOTFULLSTATUSENUUMB5	NEWMSGSTATUSENUUMB5	NOTFULLSTATUSENUUMB4	NEWMSGSTATUSENUUMB4	NOTFULLSTATUSENUUMB3	NEWMSGSTATUSENUUMB3	NOTFULLSTATUSENUUMB2	NEWMSGSTATUSENUUMB2	NOTFULLSTATUSENUUMB1	NEWMSGSTATUSENUUMB1	NOTFULLSTATUSENUUMB0	NEWMSGSTATUSENUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLSTATUSENUUMB15	NotFull Status bit for User u, Mailbox 15 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
30	NEWMSGSTATUSENUUMB15	NewMessage Status bit for User u, Mailbox 15 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
29	NOTFULLSTATUSENUUMB14	NotFull Status bit for User u, Mailbox 14 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
28	NEWMSGSTATUSENUUMB14	NewMessage Status bit for User u, Mailbox 14 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
27	NOTFULLSTATUSENUUMB13	NotFull Status bit for User u, Mailbox 13 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
26	NEWMSGSTATUSENUUMB13	NewMessage Status bit for User u, Mailbox 13 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
25	NOTFULLSTATUSENUUMB12	NotFull Status bit for User u, Mailbox 12 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
24	NEWMSGSTATUSENUUMB12	NewMessage Status bit for User u, Mailbox 12 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
23	NOTFULLSTATUSENUUMB11	NotFull Status bit for User u, Mailbox 11 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
22	NEWMSGSTATUSENUUMB11	NewMessage Status bit for User u, Mailbox 11 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
21	NOTFULLSTATUSENUUMB10	NotFull Status bit for User u, Mailbox 10 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
20	NEWMSGSTATUSENUUMB10	NewMessage Status bit for User u, Mailbox 10 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
19	NOTFULLSTATUSENUUMB9	NotFull Status bit for User u, Mailbox 9 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
18	NEWMSGSTATUSENUUMB9	NewMessage Status bit for User u, Mailbox 9 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
17	NOTFULLSTATUSENUUMB8	NotFull Status bit for User u, Mailbox 8 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
16	NEWMSGSTATUSENUUMB8	NewMessage Status bit for User u, Mailbox 8 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
15	NOTFULLSTATUSENUUMB7	NotFull Status bit for User u, Mailbox 7 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
14	NEWMSGSTATUSENUUMB7	NewMessage Status bit for User u, Mailbox 7 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
13	NOTFULLSTATUSENUUMB6	NotFull Status bit for User u, Mailbox 6 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
12	NEWMSGSTATUSENUUMB6	NewMessage Status bit for User u, Mailbox 6 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
11	NOTFULLSTATUSENUUMB5	NotFull Status bit for User u, Mailbox 5 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
10	NEWMSGSTATUSENUUMB5	NewMessage Status bit for User u, Mailbox 5 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
9	NOTFULLSTATUSENUUMB4	NotFull Status bit for User u, Mailbox 4 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
8	NEWMSGSTATUSENUUMB4	NewMessage Status bit for User u, Mailbox 4 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
7	NOTFULLSTATUSENUUMB3	NotFull Status bit for User u, Mailbox 3 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
6	NEWMSGSTATUSENUUMB3	NewMessage Status bit for User u, Mailbox 3 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
5	NOTFULLSTATUSENUUMB2	NotFull Status bit for User u, Mailbox 2 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
4	NEWMSGSTATUSENUUMB2	NewMessage Status bit for User u, Mailbox 2 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
3	NOTFULLSTATUSENUUMB1	NotFull Status bit for User u, Mailbox 1 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
2	NEWMSGSTATUSENUUMB1	NewMessage Status bit for User u, Mailbox 1 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
1	NOTFULLSTATUSENUUMB0	NotFull Status bit for User u, Mailbox 0 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
0	NEWMSGSTATUSENUUMB0	NewMessage Status bit for User u, Mailbox 0 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0

Table 19-40. MAILBOX_IRQENABLE_SET_u

Address Offset	0x0000 0108 + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2..13, IVA_MBOX, EVEx_MBOX)
Physical Address	0x4A0F 4108 + (0x10 * u) 0x5A05 A908 + (0x10 * u)	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOTFULLENABLEUUMB15	NEWMSGENABLEUUMB15	NOTFULLENABLEUUMB14	NEWMSGENABLEUUMB14	NOTFULLENABLEUUMB13	NEWMSGENABLEUUMB13	NOTFULLENABLEUUMB12	NEWMSGENABLEUUMB12	NOTFULLENABLEUUMB11	NEWMSGENABLEUUMB11	NOTFULLENABLEUUMB10	NEWMSGENABLEUUMB10	NOTFULLENABLEUUMB9	NEWMSGENABLEUUMB9	NOTFULLENABLEUUMB8	NEWMSGENABLEUUMB8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTFULLENABLEUUMB7	NEWMSGENABLEUUMB7	NOTFULLENABLEUUMB6	NEWMSGENABLEUUMB6	NOTFULLENABLEUUMB5	NEWMSGENABLEUUMB5	NOTFULLENABLEUUMB4	NEWMSGENABLEUUMB4	NOTFULLENABLEUUMB3	NEWMSGENABLEUUMB3	NOTFULLENABLEUUMB2	NEWMSGENABLEUUMB2	NOTFULLENABLEUUMB1	NEWMSGENABLEUUMB1	NOTFULLENABLEUUMB0	NEWMSGENABLEUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLENABLEUUMB15	NotFull Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
30	NEWMSGENABLEUUMB15	NewMessage Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
29	NOTFULLENABLEUUMB14	NotFull Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
28	NEWMSGENABLEUUMB14	NewMessage Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
27	NOTFULLENABLEUUMB13	NotFull Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
26	NEWMSGENABLEUUMB13	NewMessage Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
25	NOTFULLENABLEUUMB12	NotFull Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
24	NEWMSGENABLEUUMB12	NewMessage Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
23	NOTFULLENABLEUUMB11	NotFull Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
22	NEWMSGENABLEUUMB11	NewMessage Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
21	NOTFULLENABLEUUMB10	NotFull Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
20	NEWMSGENABLEUUMB10	NewMessage Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
19	NOTFULLENABLEUUMB9	NotFull Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
18	NEWMSGENABLEUUMB9	NewMessage Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
17	NOTFULLENABLEUUMB8	NotFull Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
16	NEWMSGENABLEUUMB8	NewMessage Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
15	NOTFULLENABLEUUMB7	NotFull Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
14	NEWMSGENABLEUUMB7	NewMessage Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
13	NOTFULLENABLEUUMB6	NotFull Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
12	NEWMSGENABLEUUMB6	NewMessage Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
11	NOTFULLENABLEUUMB5	NotFull Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
10	NEWMSGENABLEUUMB5	NewMessage Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
9	NOTFULLENABLEUUMB4	NotFull Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
8	NEWMSGENABLEUUMB4	NewMessage Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
7	NOTFULLENABLEUUMB3	NotFull Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
6	NEWMSGENABLEUUMB3	NewMessage Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
5	NOTFULLENABLEUUMB2	NotFull Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
4	NEWMSGENABLEUUMB2	NewMessage Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
3	NOTFULLENABLEUUMB1	NotFull Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
2	NEWMSGENABLEUUMB1	NewMessage Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
1	NOTFULLENABLEUUMB0	NotFull Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
0	NEWMSGENABLEUUMB0	NewMessage Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Table 19-41. MAILBOX_IRQENABLE_CLR_u

Address Offset	0x0000 010C + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2..13, IVA_MBOX, EVEx_MBOX)
Physical Address	0x4A0F 410C + (0x10 * u) 0x5A05 A90C + (0x10 * u)	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOTFULLENABLEUUMB15	NEWMSGENABLEUUMB15	NOTFULLENABLEUUMB14	NEWMSGENABLEUUMB14	NOTFULLENABLEUUMB13	NEWMSGENABLEUUMB13	NOTFULLENABLEUUMB12	NEWMSGENABLEUUMB12	NOTFULLENABLEUUMB11	NEWMSGENABLEUUMB11	NOTFULLENABLEUUMB10	NEWMSGENABLEUUMB10	NOTFULLENABLEUUMB9	NEWMSGENABLEUUMB9	NOTFULLENABLEUUMB8	NEWMSGENABLEUUMB8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTFULLENABLEUUMB7	NEWMSGENABLEUUMB7	NOTFULLENABLEUUMB6	NEWMSGENABLEUUMB6	NOTFULLENABLEUUMB5	NEWMSGENABLEUUMB5	NOTFULLENABLEUUMB4	NEWMSGENABLEUUMB4	NOTFULLENABLEUUMB3	NEWMSGENABLEUUMB3	NOTFULLENABLEUUMB2	NEWMSGENABLEUUMB2	NOTFULLENABLEUUMB1	NEWMSGENABLEUUMB1	NOTFULLENABLEUUMB0	NEWMSGENABLEUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLENABLEUUMB15	NotFull Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
30	NEWMSGENABLEUUMB15	NewMessage Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
29	NOTFULLENABLEUUMB14	NotFull Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
28	NEWMSGENABLEUUMB14	NewMessage Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
27	NOTFULLENABLEUUMB13	NotFull Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
26	NEWMSGENABLEUUMB13	NewMessage Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
25	NOTFULLENABLEUUMB12	NotFull Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
24	NEWMSGENABLEUUMB12	NewMessage Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
23	NOTFULLENABLEUUMB11	NotFull Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
22	NEWMSGENABLEUUMB11	NewMessage Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
21	NOTFULLENABLEUUMB10	NotFull Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
20	NEWMSGENABLEUUMB10	NewMessage Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
19	NOTFULLENABLEUUMB9	NotFull Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
18	NEWMSGENABLEUUMB9	NewMessage Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
17	NOTFULLENABLEUUMB8	NotFull Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
16	NEWMSGENABLEUUMB8	NewMessage Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
15	NOTFULLENABLEUUMB7	NotFull Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
14	NEWMSGENABLEUUMB7	NewMessage Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
13	NOTFULLENABLEUUMB6	NotFull Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
12	NEWMSGENABLEUUMB6	NewMessage Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
11	NOTFULLENABLEUUMB5	NotFull Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
10	NEWMSGENABLEUUMB5	NewMessage Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
9	NOTFULLENABLEUUMB4	NotFull Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
8	NEWMSGENABLEUUMB4	NewMessage Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
7	NOTFULLENABLEUUMB3	NotFull Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
6	NEWMSGENABLEUUMB3	NewMessage Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
5	NOTFULLENABLEUUMB2	NotFull Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
4	NEWMSGENABLEUUMB2	NewMessage Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
3	NOTFULLENABLEUUMB1	NotFull Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
2	NEWMSGENABLEUUMB1	NewMessage Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
1	NOTFULLENABLEUUMB0	NotFull Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
0	NEWMSGENABLEUUMB0	NewMessage Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Note

For each interrupt status and enable register (MAILBOX_IRQSTATUS_RAW_u, MAILBOX_IRQSTATUS_CLR_u, MAILBOX_IRQENABLE_SET_u and MAILBOX_IRQENABLE_CLR_u):

- Bits [31:0] have the given meaning for the EVEx_MBOX0..2 instances.
- Bits [31:24] are RESERVED for the MAILBOX2..13 instances.
- Bits [31:16] are RESERVED for the MAILBOX1 instance.
- Bits [31:12] are RESERVED for the IVA_MBOX instance.

Table 19-42. MAILBOX_IRQ_EOI

Address Offset	0x0000 0140		
Physical Address	0x4A0F 4140 0x5A05 A940	**Instance**	MAILBOX1_CFG_L4 IVA_MBOX_MAIN_L3
Description	This register is used for the software EOI clearance of the pulse. This register being write only gives 0 on read.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EOIVAL	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0's for future compatibility. Reads returns 0	W	0x0
1:0	EOIVAL	EOI value 0x0: EOI val first bit 0x1: EOI val second bit	W	0x0

Chapter 20

Memory Management Units

This chapter describes the memory management units (MMUs) in the device.

20.1 MMU Overview 5148
20.2 MMU Integration 5151
20.3 MMU Functional Description 5153
20.4 MMU Low-level Programming Models 5165
20.5 MMU Register Manual 5168

20.1 MMU Overview

A memory management unit (MMU) is a hardware component responsible for handling accesses to memory requested by a processing unit, DMA controller, or other bus requestor. MMU functions include:

- Translation of initiator internal (virtual) addresses to physical addresses (that is, virtual memory management)
- Preventing an initiator from making accesses to unmapped pages of the system memory

This device includes the following MMUs:

- Two top-level (system) MMUs:
  - MMU1 dedicated to EDMA Transfer Controller 0 (TC0), and EDMA Transfer Controller 1 (TC1)
  - MMU2 dedicated to PCIe_SS1, and PCIeSS2
- One MMU inside the MPU (dual Cortex®-A15) subsystem – MPU_MMU. This MMU is integrated in the Cortex-A15 processor.
- Two MMUs inside each of the DSP1, and DSP2 subsystems:
  - DSP1 subsystem: DSP1_MMU0, and DSP1_MMU1
  - DSP2 subsystem: DSP2_MMU0, and DSP2_MMU1
- Two MMUs inside each of the EVE1, and EVE2 subsystems:
  - EVE1 subsystem: EVE1_MMU0, and EVE1_MMU1
  - EVE2 subsystem: EVE2_MMU0, and EVE2_MMU1
- Two MMUs inside each of the IPU1, and IPU2 subsystems:
  - L1 unicache MMU – IPU1_UNICACHE_MMU, and IPU2_UNICACHE_MMU
  - L2 MMU – IPU1_MMU, and IPU2_MMU
- One MMU inside the 3D GPU (dual SGX544 core) subsystem
- One MMU inside the BB2D subsystem

Note

There is a Physical Address Translator (PAT) module in the Dynamic Memory Manager (DMM), which has similar to the MMU functionality. For more information about this module, see Section 15.2, *Dynamic Memory Manager*.

Note

This chapter provides a detailed description of the following MMUs:

- System MMUs
- DSP MMUs
- EVE MMUs
- IPU L2 MMU

System MMUs, DSP MMUs and EVE MMUs are fully identical from functional perspective. IPU L2 MMU is different in that it does not support “bypass” functionality.

For more information about:

- Cortex-A15 MMU, see the Arm® *Cortex®-A15 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).
- IPU unicache MMU, see Chapter 7, *Dual Cortex-M4 IPU Subsystem*.

Figure 20-1 and Figure 20-2 show an overview of system MMU1 and MMU2, respectively. In summary, requests initiated by a given requestor (EDMA TC0 and TC1 [both read and write ports] for system MMU1; PCIe_SS1 and PCIe_SS2 for system MMU2) can optionally be routed through the corresponding system MMU. Each requestor’s use (or not) of the MMU is independently controllable via the Control Module CTRL_CORE_SMA_SW_7 register bitfields. It is recommended that this register is set during system initialization and remain static.

If the MMU loopback path is enabled for a given requestor, requests will be routed via the L3_MAIN interconnect to the MMU and will again go through the L3 interconnect to the requested physical address. If the MMU loopback path is disabled for a given requestor, those bus requests go directly through the L3_MAIN interconnect to the requested physical address, thus minimizing bus request latency.

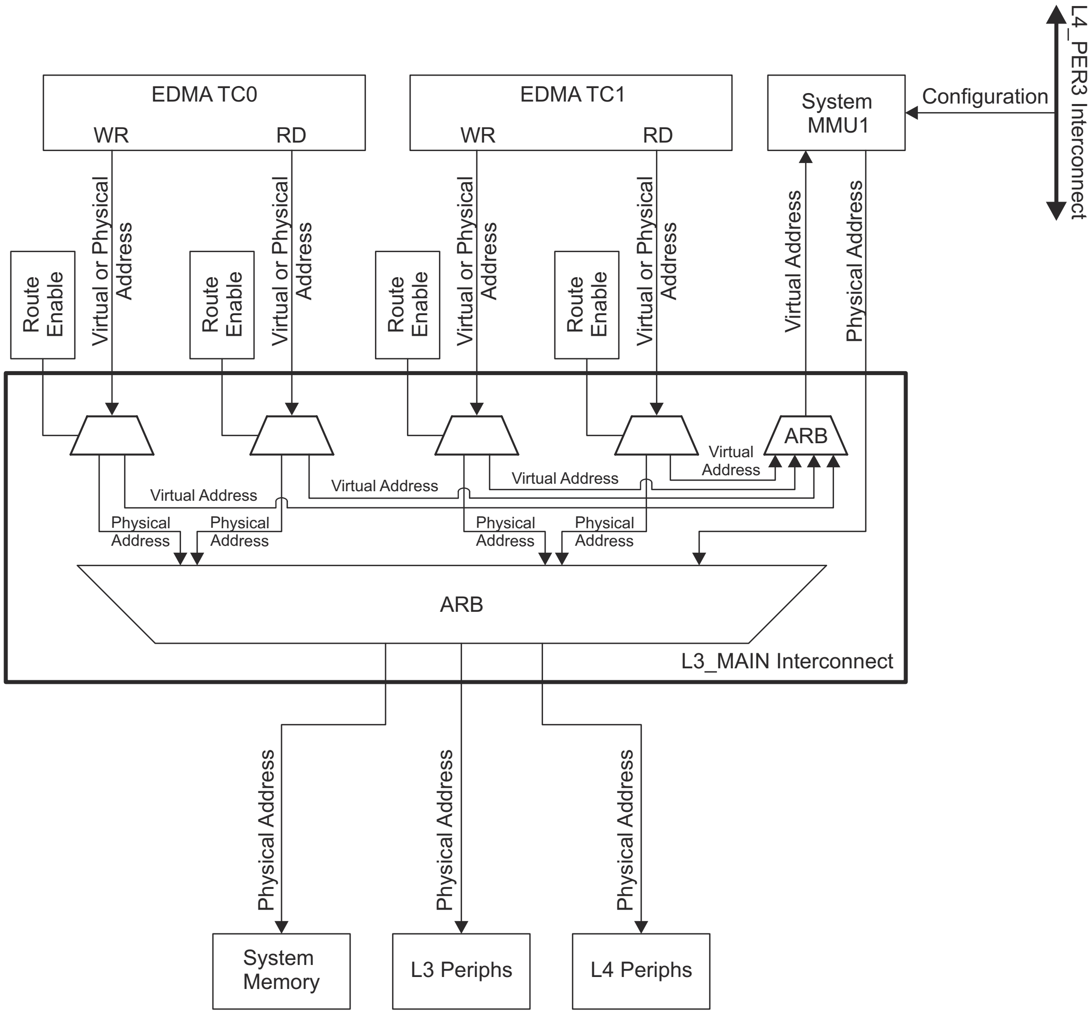


Figure 20-1. System MMU1 Overview

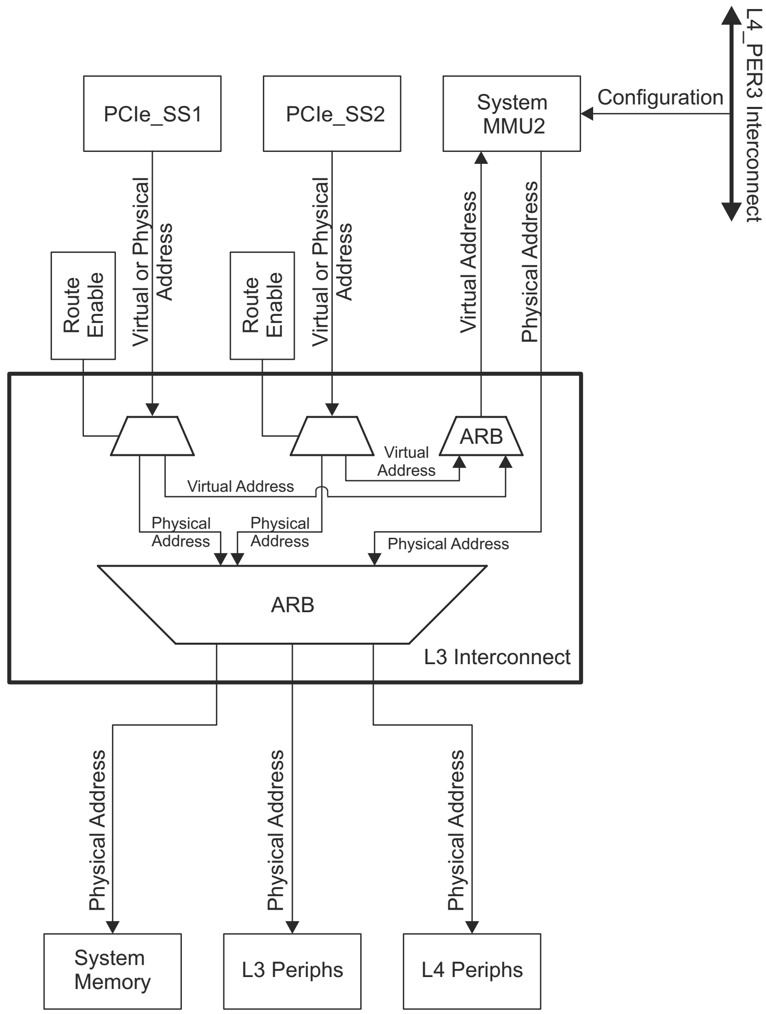


Figure 20-2. System MMU2 Overview

20.2 MMU Integration

This section describes the system MMUs integration in the device, including information about clocks, resets, and hardware requests. For more information about DSP, EVE, and IPU MMUs integration, refer to their respective chapters.

Figure 20-3 and Figure 20-4 show system MMU1, and MMU2 integration, respectively.

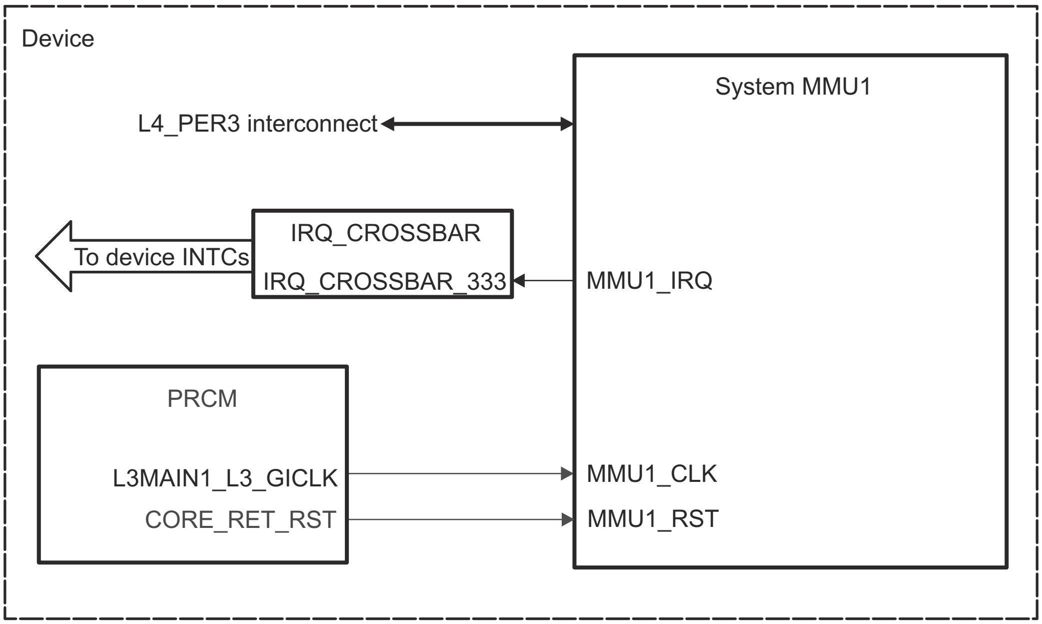


Figure 20-3. System MMU1 Integration

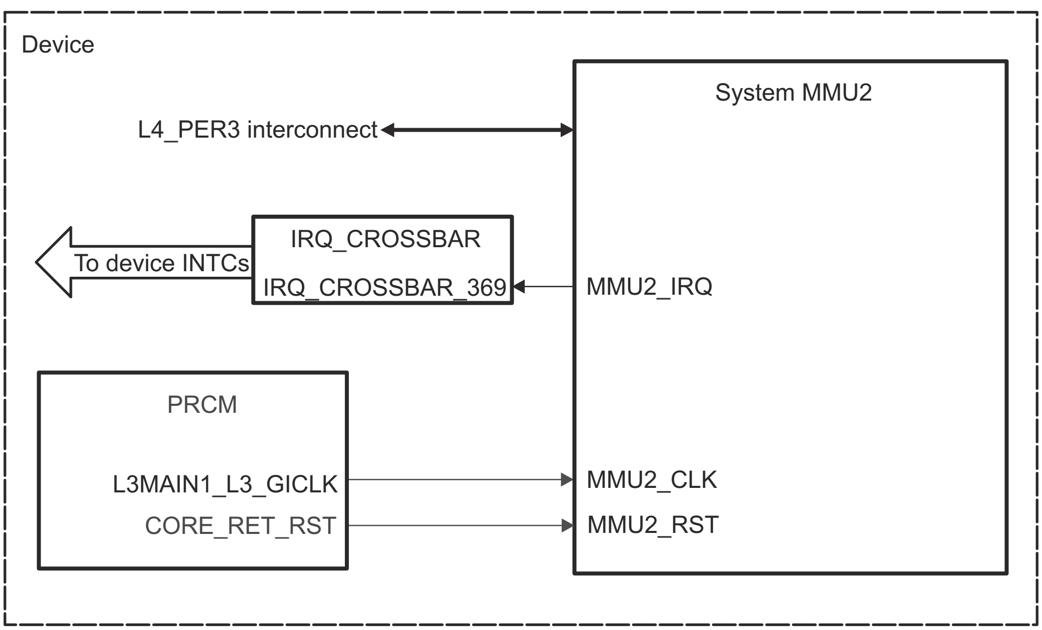


Figure 20-4. System MMU2 Integration

Table 20-1 through Table 20-3 summarize the system MMUs integration.

Table 20-1. MMU Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
System MMU1	PD_COREAON	L4_PER3
System MMU2	PD_COREAON	L4_PER3

Table 20-2. MMU Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
System MMU1	MMU1_CLK	L3MAIN1_L3_GICLK	PRCM	System MMU1 interface/functional clock. This clock is used for all interface and functional operations.
System MMU2	MMU2_CLK	L3MAIN1_L3_GICLK	PRCM	System MMU2 interface/functional clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
System MMU1	MMU1_RST	CORE_RET_RST	PRCM	System MMU1 hardware reset. This reset is asynchronously applied to the MMU1 internal registers.
System MMU2	MMU2_RST	CORE_RET_RST	PRCM	System MMU2 hardware reset. This reset is asynchronously applied to the MMU2 internal registers.

Table 20-3. MMU Hardware Requests

Interrupt Requests				
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**
System MMU1	MMU1_IRQ	IRQ_CROSSBAR_333	–	System MMU1 interrupt.
System MMU2	MMU2_IRQ	IRQ_CROSSBAR_369	–	System MMU2 interrupt.
No DMA Requests				

Note

For a description of the interrupt sources, see Section 20.3.4, *MMU Interrupt Requests*.

20.3 MMU Functional Description

20.3.1 MMU Block Diagram

The MMU manages the virtual to physical address translation for external addresses. Figure 20-5 shows the MMU block diagram.

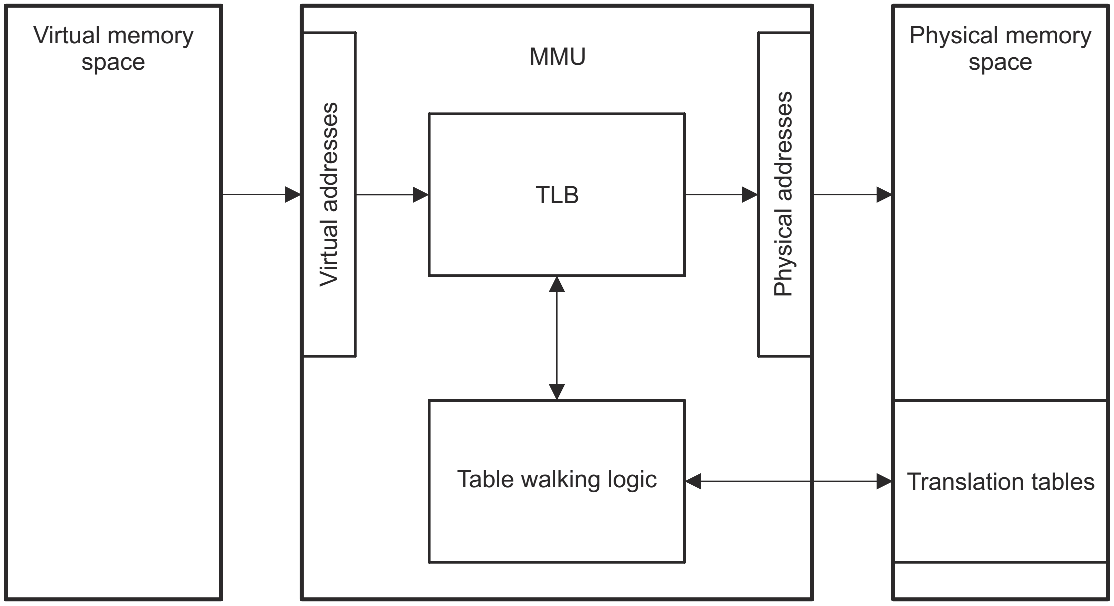


Figure 20-5. MMU Block Diagram

Each table entry describes the translation of one contiguous memory region. For a description of the structure of these tables, see Section 20.3.1.2, *Translation Tables*.

Two major functional units exist in the MMU to provide address translation automatically based on the table entries:

- The table walker automatically retrieves the correct translation table entry for a requested translation. If two-level translation is used (for the translation of small memory pages), the table walker also automatically reads the required second-level translation table entry. The two-level translation is described later in the chapter.
- The translation look-aside buffer (TLB) stores recently used translation entries, acting like a cache of the translation table.

20.3.1.1 MMU Address Translation Process

Whenever an address translation is requested (that is, for every access with the MMU enabled), the MMU first checks whether the translation is contained in the TLB, which acts like a cache storing recent translations. The TLB can also be programmed manually to ensure that time-critical data can be translated without delay.

If the requested translation is not in the TLB, the table-walking logic retrieves this translation from the translation table(s), and then updates the TLB. The address translation is then performed. Figure 20-6 summarizes the process.

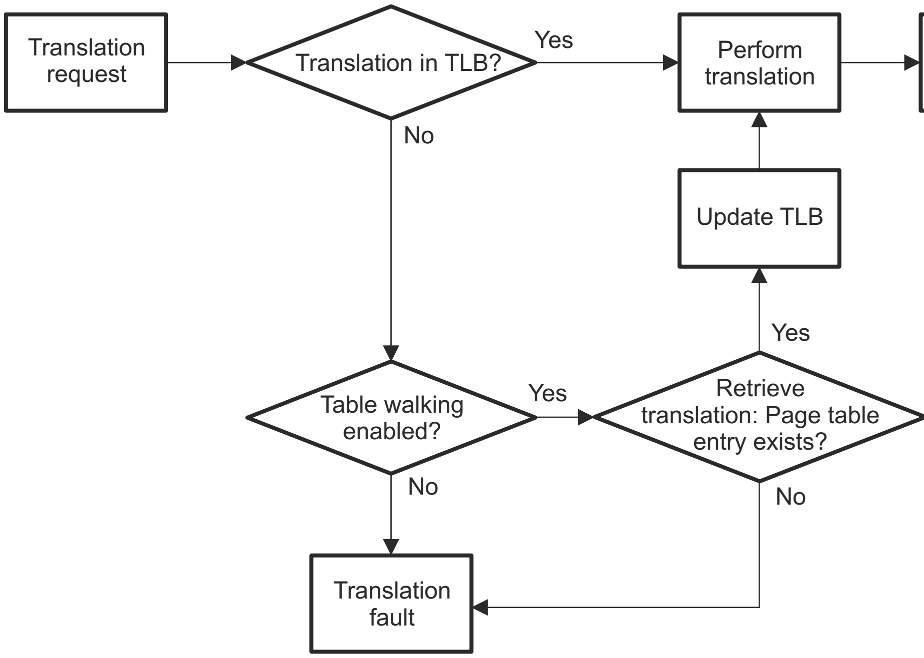


Figure 20-6. Translation Process

20.3.1.2 Translation Tables

The translation of virtual to physical addresses is based on entries in translation tables that define the following properties:

- Address translation, that is, the correspondence between virtual and physical addresses
- Size of the memory region the entry translates

The virtual addresses index the translation tables. Each virtual address corresponds to exactly one entry in the translation table.

20.3.1.2.1 Translation Table Hierarchy

When developing a table-based address translation scheme, one of the most important design parameters is the memory page size described by each translation table entry. MMU instances support 4-KiB and 64-KiB pages, a 1-MiB section, and a 16-MiB supersection. Using bigger page sizes means a smaller translation table.

Using a smaller page size greatly increases the efficiency of dynamic memory allocation and defragmentation. That is why many operating systems (OSs) can operate on memory blocks as small as 4 KiB; however, the smaller size implies a more complex table structure.

A quick calculation shows that using 4 KiB memory pages with one translation table would require one million entries to span the entire 4-GiB address range. The table itself would be 32 MiB, a size that is not feasible.

However, using bigger pages reduces the flexibility of typical OS memory management. Implementing a two-level hierarchy reconciles these two requirements. Within this hierarchy, one first-level translation table describes the translation properties based on 1 MiB memory regions.

Each of the entries in this first-level translation table can specify the following:

- The translation properties for a big memory section. This memory section can be either 1 MiB (section) or 16 MiB (supersection). In this case, all translation parameters are specified in the first-level translation table entry.
- A pointer to a second-level translation table that specifies individual translation properties based on smaller pages within the 1-MiB page of memory. These pages can be either 64 KiB (large page) or 4 KiB (small page). In this case, the actual translation parameters are specified in the second-level translation table entry. The first-level translation table entry specifies only the base address of the second-level translation table.

This hierarchical approach means that additional translation information for smaller pages must be provided only when the pages are actually used. Figure 20-7 shows the hierarchy.

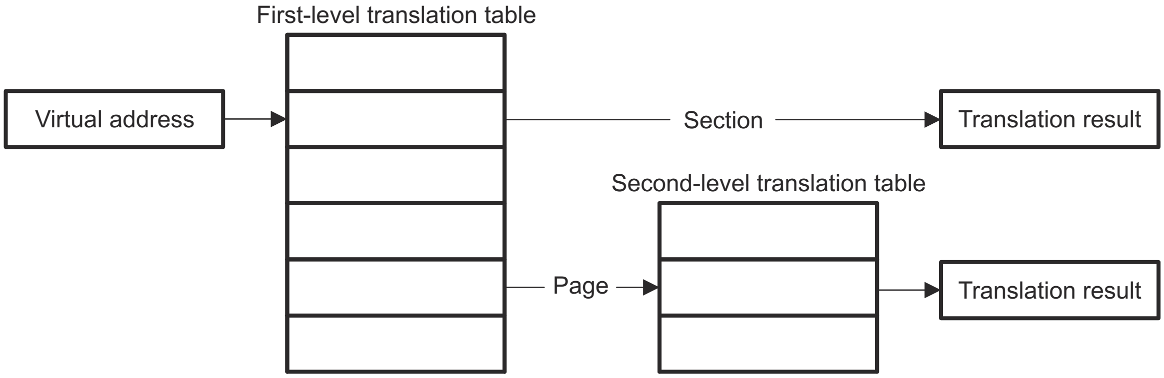


Figure 20-7. Translation Hierarchy

The structure of the first and second-level translation tables and their entries are described in more detail in Section 20.3.1.2.2, *First-Level Translation Table*, and Section 20.3.1.2.3, *Two-Level Translation*.

20.3.1.2.2 First-Level Translation Table

The first-level translation table describes the translation properties for 1-MiB sections. To describe a 4-GiB address range requires 4096 32-bit entries (so-called first-level descriptors).

The first-level translation table start address must be aligned on a multiple of the table size with a 128-byte minimum. Consequently, an alignment of at least 16K bytes is required for a complete 4096-entry table; that is, at least the last fourteen address bits must be zero.

The start address of the first-level translation table is specified by the so-called translation table base. The table is indexed by the upper 12-bits of the virtual address. Figure 20-8 shows this mechanism.

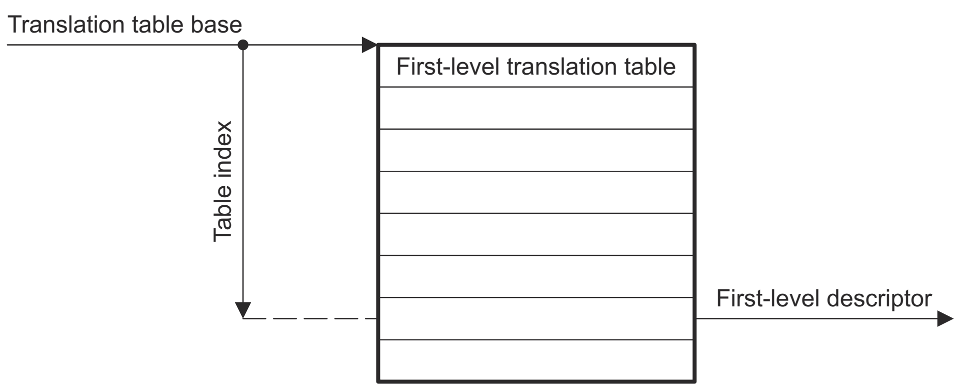


Figure 20-8. First-level Descriptor Address Calculation

To summarize, the translation table base and the translation table index together define the first-level descriptor address. Figure 20-9 outlines the precise mechanism used to calculate this address.

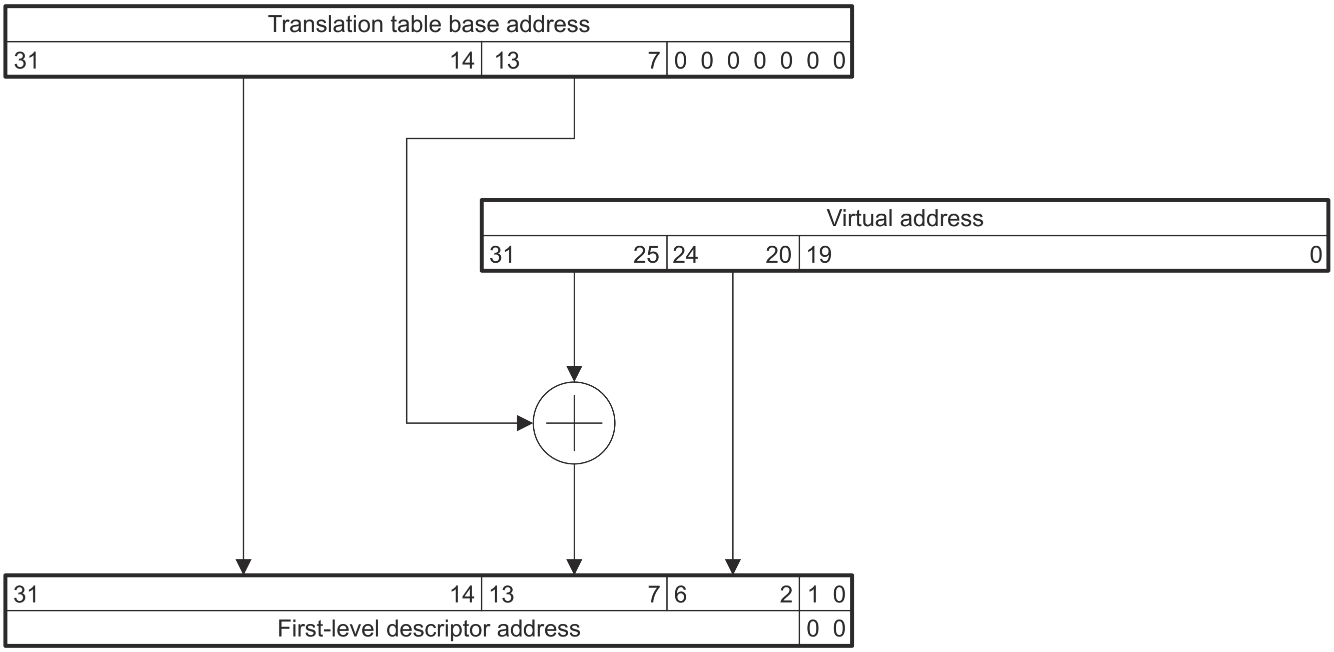


Figure 20-9. Detailed First-Level Descriptor Address Calculation

As an example of this mechanism, consider a translation table base address of 0x8000:0000 and a virtual address of 0x1234:5678. In this case, the first-level descriptor address is 0x8000:0000 + (0x123 << 2) = 0x8000:048C.

20.3.1.2.2.1 First-Level Descriptor Format

Each first-level descriptor provides either the complete address translation for 1-MiB or 16-MiB sections or provides a pointer to a second-level translation table for 4 KiB or 64 KiB pages. Table 20-4 shows the first-level descriptor format.

Table 20-4. First-Level Descriptor Format

First-Level Descriptor Format								
31:24	23:20	19	18	17:10	9:2	1	0	
X						0	0	Fault
Second-Level Translation Table Base Address					X	0	1	Page
Section Base Address		X	0	X		1	0	Section
Supersection Base Address	X		1	X		1	0	Supersection
X						1	1	Fault

X = Don't care. Set to 0 for future compatibility.

20.3.1.2.2.2 First-Level Page Descriptor Format

If a translation granularity smaller than 1 MiB is required, a two-level translation process is used. In this case, the first-level block descriptor specifies only the start address of a second-level translation table. The second-level translation table entries specify the actual translation properties.

20.3.1.2.2.3 First-Level Section Descriptor Format

Each section descriptor in the first-level translation table specifies the complete translation properties for a 1-MiB section or a 16-MiB supersection.

Note

Supersection descriptors must be repeated 16 times, because each descriptor in the first-level translation table describes 1 MiB of memory. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

20.3.1.2.2.4 Section Translation Summary

Sections and supersections can be translated based solely on the information in the first-level translation table. Figure 20-10 summarizes the address translation process for a section.

Figure 20-10. Section Translation Summary

20.3.1.2.2.5 Supersection Translation Summary

The translation of a supersection is similar to the translation of a section. The difference is that for a supersection only bits 31 to 24 index into the first-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a supersection.

Figure 20-11 shows the translation mechanism for a supersection.

Figure 20-11. Supersection Translation Summary

20.3.1.2.3 Two-Level Translation

Two-level translation is used when fine-grain granularity is required, that is, when memory sections smaller than 1 MiB are needed. In this case, the first-level descriptor provides a pointer to the base address of a second-level translation table. This second-level table is indexed by bits 19 to 12 of the virtual address. Figure 20-12 shows this indexing mechanism.

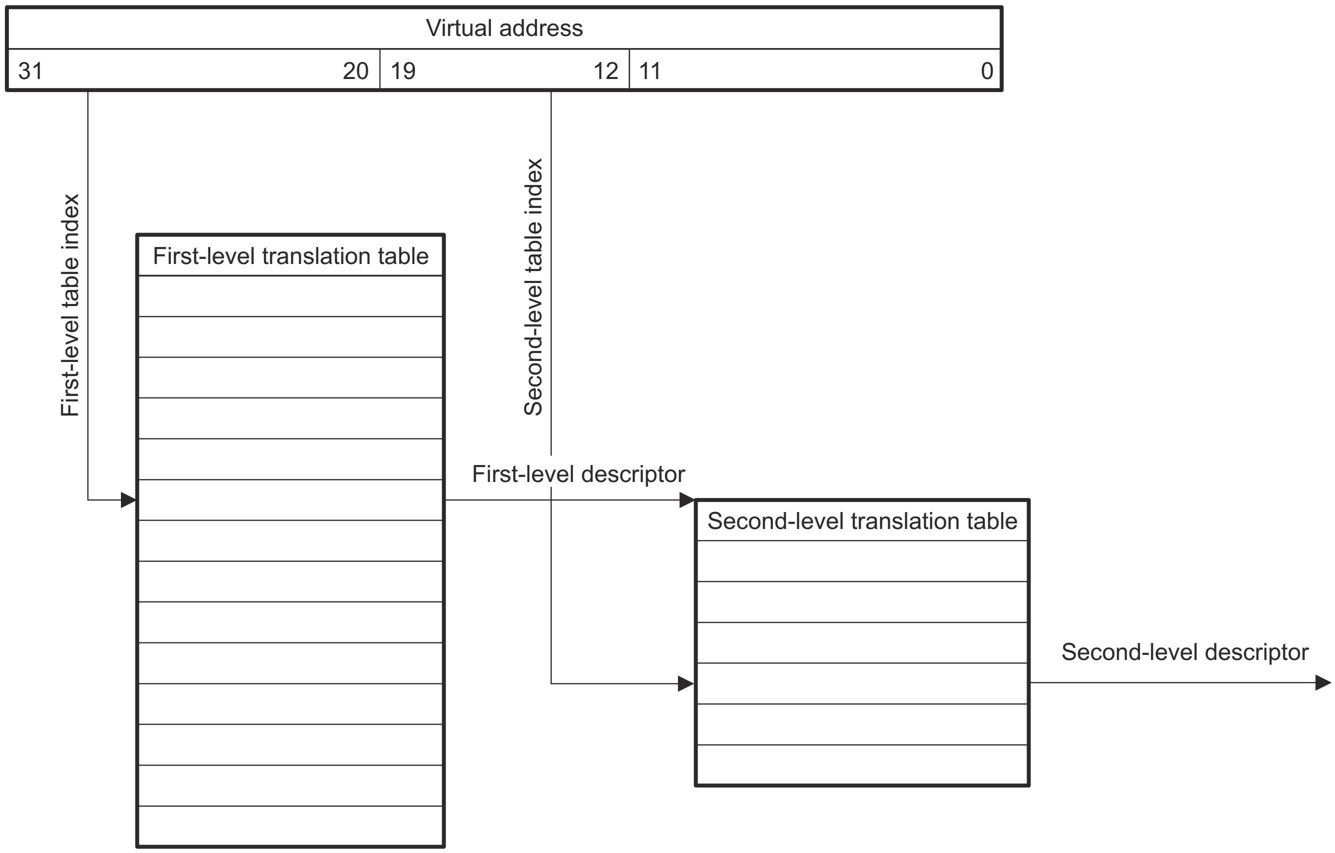


Figure 20-12. Two-Level Translation

Each second-level translation table describes the translation of 1 MiB of address space in pages of 64 KiB (large page) or 4 KiB (small page). It consists of 256 second-level descriptors describing 4 KiB each.

Note

In the case of a large page, the same descriptor must be repeated 16 times. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

20.3.1.2.3.1 Second-Level Descriptor Format

Similar to first-level section descriptors, second-level descriptors provide all of the necessary information for the translation of a large or small page. Table 20-5 shows the format of second-level descriptors.

Table 20-5. Second-Level Descriptor Format

Second-Level Descriptor Format					
31:16	15:12	11:2	1	0	
X			0	0	Fault
Large Page Base Address	X		0	1	Large Page
Small Page Base Address		X	1	X	Small Page

X = Don't care. Set to 0 for future compatibility.

20.3.1.2.3.2 Small Page Translation Summary

Figure 20-13 summarizes the translation process for small pages.

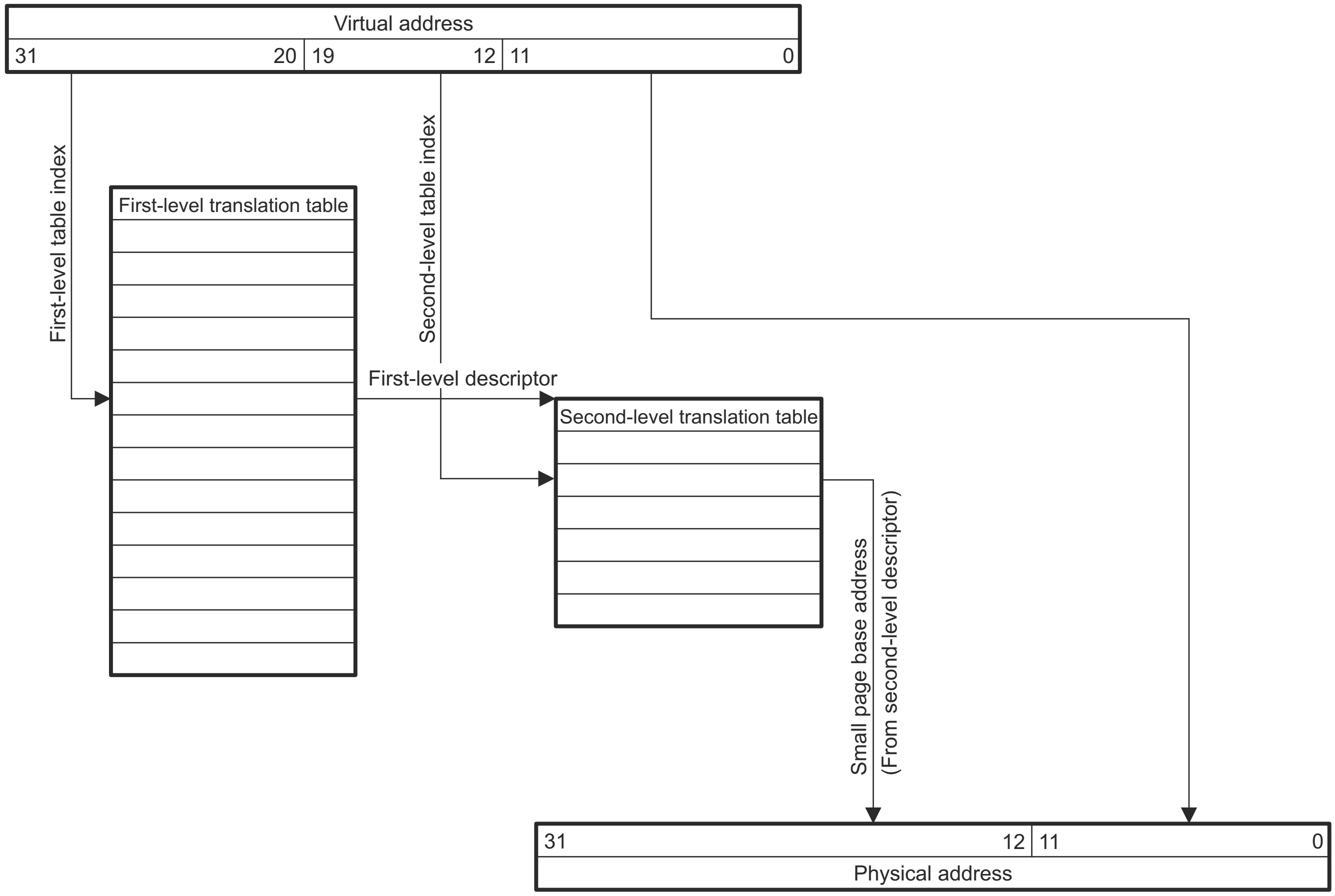


Figure 20-13. Small Page Translation Summary

20.3.1.2.3.3 Large Page Translation Summary

The translation of a large page is similar to the translation of a small page. The difference is that, for a large page, only bits 19 to 16 index into the second-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a large page. This is shown in Figure 20-14.

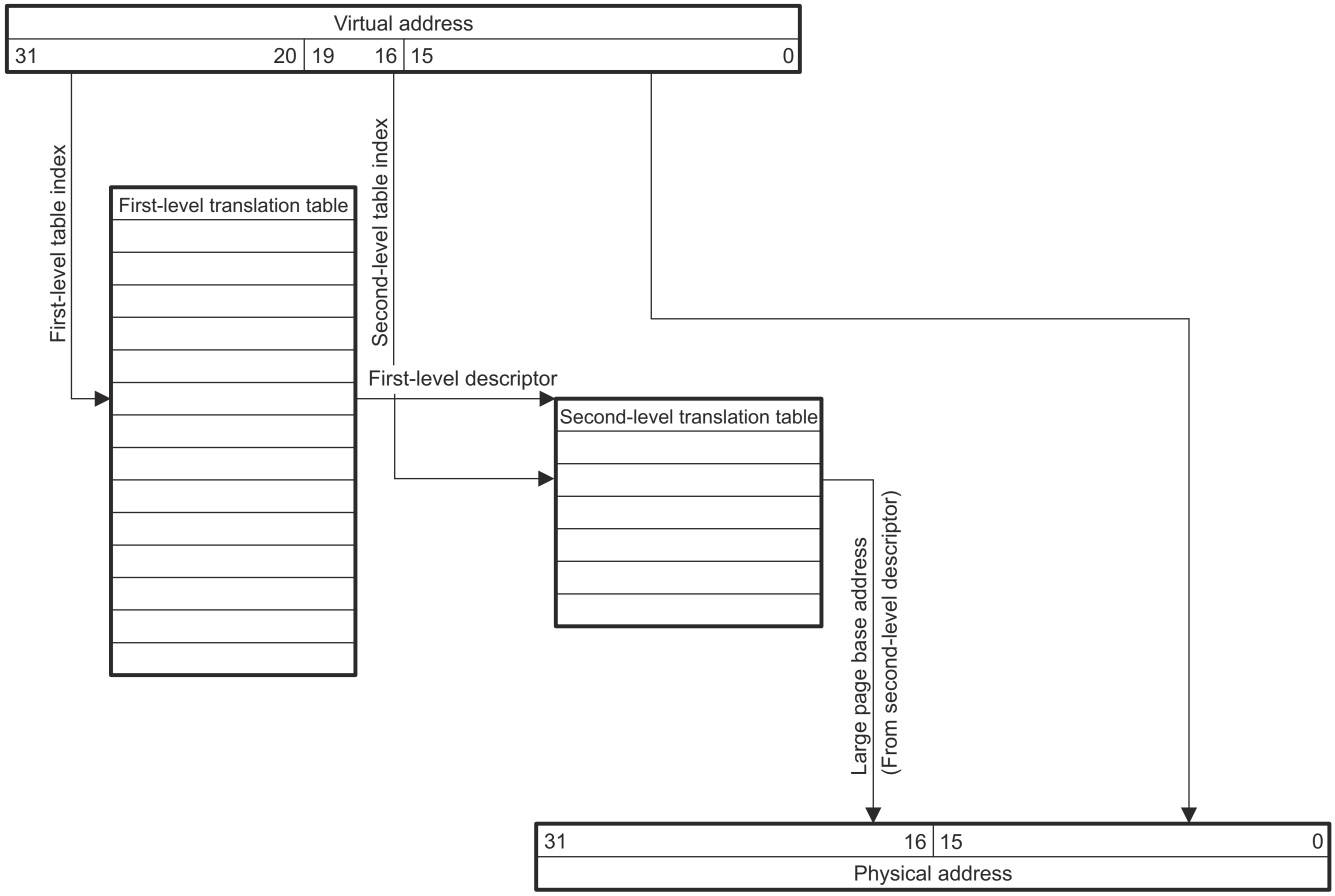


Figure 20-14. Large Page Translation Summary

20.3.1.3 Translation Lookaside Buffer

Translating virtual addresses to physical addresses is required for each memory access in systems using an MMU. To accelerate this translation process, a cache, or TLB, holds the result of recent translations.

For every translation, the MMU internal logic first checks whether the requested translation is already cached in the TLB. If the translation is cached, this translation is used; otherwise the translation is retrieved from the translation tables and the TLB is updated. If the TLB is full, one of its entries must be replaced. This entry is selected on a random basis.

The first *n* TLB entries, where *n* < *Total Number N of TLB Entries*, can be protected (locked) against being overwritten by setting the TLB base pointer to *n*. When this mechanism is used, only unprotected entries can be overwritten. The victim pointer indicates the next TLB entry to be written. Figure 20-15 shows an example of the TLB with N TLB entries (ranging from 0 to N-1). The base pointer contains the value "3" protecting Entry 0, Entry 1, and Entry 2 and the victim pointer points to the next TLB entry to be updated.

Note

The last TLB entry (Entry N-1) always remains unprotected.

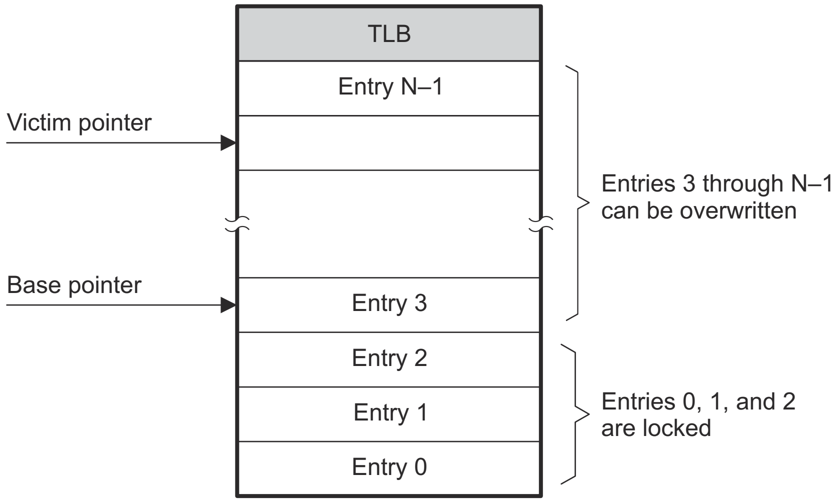


Figure 20-15. TLB Entry Lock Mechanism

The table walking logic automatically writes the TLB entries. The entries can also be manually written, which is done typically to ensure that the translation of time-critical data accesses is already present in the TLB so that they execute as fast as possible. The entries must be locked to prevent them from being overwritten.

20.3.1.3.1 TLB Entry Format

TLB entries consist of two parts:

- The Content Addressable Memory (CAM) part contains the virtual address tag used to determine if a virtual address translation is in the TLB. The TLB acts like a fully associative cache addressed by the virtual address tag. The CAM part also contains the section/page size, as well as the preserved and the valid parameters. See the MMU_CAM register table for more details.
- The Random Addressable Memory (RAM) part contains the address translation that belongs to the virtual address tag. See the MMU_RAM register table for more details.

The valid parameter specifies whether an entry is valid or not. The preserved parameter determines the behavior of an entry in the event of a TLB flush. If an entry is set as preserved, it is not deleted when a TLB is flushed, that is, when MMU_GFLUSH[0] GLOBALFLUSH is set to 1. Preserved entries must be deleted manually. Section 20.3.1.2.2, *First-Level Translation Table* describes the procedure to delete TLB entries.

Figure 20-17 shows the TLB entry structure.

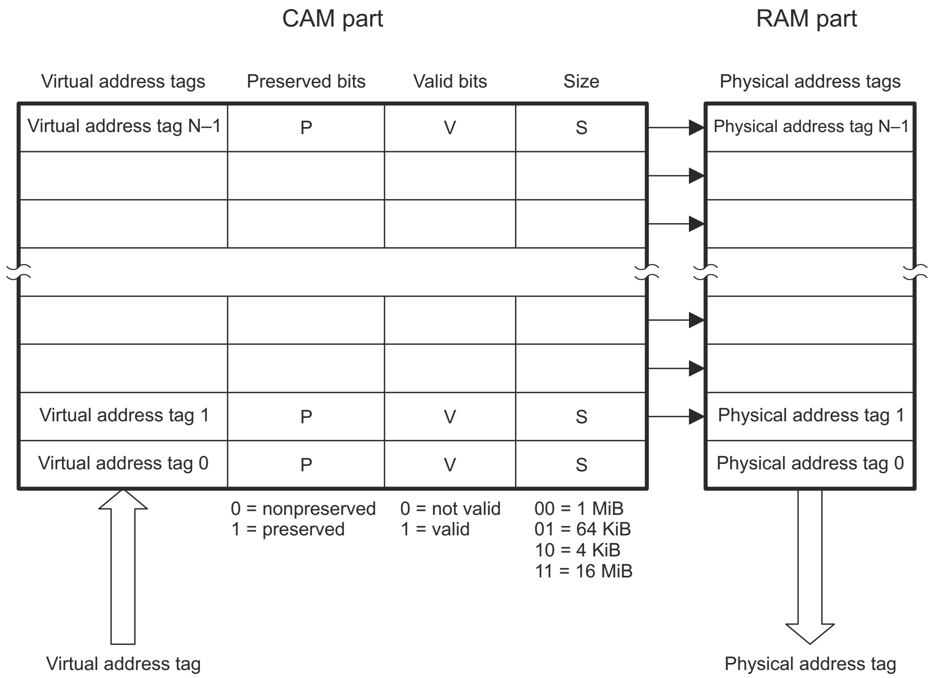


Figure 20-16. TLB Entry Structure

20.3.1.4 No Translation (Bypass) Regions

The MMU provides support for up to four user programmable regions where there is no address translation. Any access to a region specified by the MMU_BYPASS_REGIONx_ADDR and MMU_BYPASS_REGIONx_SIZE registers (where x = 1 to 4) will have no virtual to physical address translation.

20.3.2 MMU Software Reset

To perform a software reset, write 1 in the MMU_SYSCONFIG[1] SOFTRESET bit. The MMU_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the MMU_SYSCONFIG[1] SOFTRESET bit is automatically reset. The software must ensure that the software reset completes before doing MMU operations. When an MMU instance is released from reset, its TLB is empty and the MMU is disabled.

20.3.3 MMU Power Management

Table 20-6 describes the power-management features available for the MMU modules.

Table 20-6. MMU Local Power Management Features

Feature	Register
Idle modes	MMU_SYSCONFIG[4:3] IDLEMODE
Clock activity	MMU_SYSCONFIG[9:8] CLOCKACTIVITY
Clock autogating	MMU_SYSCONFIG[0] AUTOIDLE

Note

The MMU_SYSCONFIG[9:8] CLOCKACTIVITY bit field is read only.

20.3.4 MMU Interrupt Requests

Table 20-7. MMU Events

Event Flag	Event Mask	Description
MMU_IRQSTATUS[4] MULTIHITFAULT	MMU_IRQENABLE[4] MULTIHITFAULT	Error due to multiple matches in the TLB

Table 20-7. MMU Events (continued)

Event Flag	Event Mask	Description
MMU_IRQSTATUS[3] TABLEWALKFAULT	MMU_IRQENABLE[3] TABLEWALKFAULT	Error due to error response received during a Table Walk
MMU_IRQSTATUS[2] EMUMISS	MMU_IRQENABLE[2] EMUMISS	Error due to unrecoverable TLB miss during debug (hardware TWL disabled)
MMU_IRQSTATUS[1] TRANSLATIONFAULT	MMU_IRQENABLE[1] TRANSLATIONFAULT	Error due to invalid descriptor in the translation tables (translation fault)
MMU_IRQSTATUS[0] TLBMISS	MMU_IRQENABLE[0] TLBMISS	Error due to unrecoverable TLB miss (hardware TWL disabled)

20.3.5 MMU Error Handling

Table 20-8 summarizes the intended operation for real and potential error conditions.

Table 20-8. Error Handling

Item	Condition	Action
1	Table-walk read has an error response.	Treat generally the same as a translation fault, but set the TableWalkFault interrupt status bit to aid in diagnosis
2	MMU is disabled during table-walk.	Not permitted; can result in loss of the current transaction but must not deadlock the MMU. Avoid this condition by first disabling the table-walk logic and then polling the TWLRunning bit to ensure that no table walk is pending
3	MMU is disabled during an address translation.	Not permitted; can result in access to an unintended location, but must not deadlock MMU. This condition should be avoided by ensuring that no accesses are pending.
4	TLB is accessed during an address translation or a table walk.	Reading permitted; write should be done with care to ensure that the TLB is self-consistent at all times that a translation can occur.
5	TLB is flushed during address translation or a table walk.	Permitted; the flush is processed first, followed by the TWL update.
6	MMU is disabled while an interrupt is pending.	Not permitted; all pending interrupts should be processed before disabling the MMU.
7	Interrupt is not enabled and a fault/miss happens during translation.	If MMU_GPR[0] FAULT_INTR_DIS = 1 : Error response is sent back. If MMU_GPR[0] FAULT_INTR_DIS = 0 : • Mreqdebug = 1 (debug access) : Error response is sent back • Mreqdebug = 0 (application access) : Error response is not sent. MMU is waiting for TLB to be updated through config port. But since the interrupt is not asserted, system does not know there was a fault. This results in deadlock. Software must take care of this by enabling interrupts.

20.4 MMU Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

20.4.1 Global Initialization

20.4.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the MMU module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMU. For more information, see Section 20.2, *MMU Module Integration* .

Table 20-9. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Enable MMU interface/functional clock.
(optional) Interrupt controller(s)	Configure device interrupt controller(s) to enable the interrupts from MMU.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

20.4.1.2 MMU Global Initialization

20.4.1.2.1 Main Sequence - MMU Global Initialization

Figure 20-17 shows the procedure to initialize the MMU after a power-on or software reset.

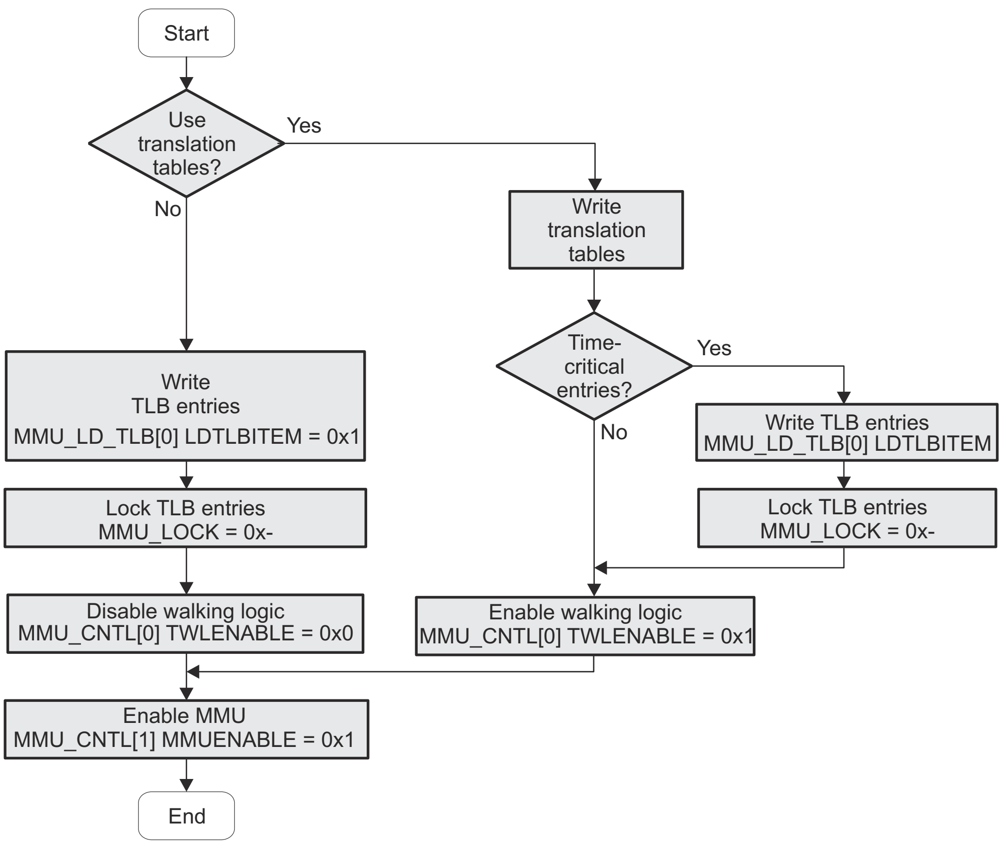


Figure 20-17. MMU Global Initialization

20.4.1.2.2 Subsequence - Configure a TLB entry

Table 20-10. Configure a TLB Entry

Step	Register / Bit Field / Programming Model	Value
Load the Virtual Address Tag	MMU_CAM[31:12] VATAG	0x-
Protect the TLB entry against flush	MMU_CAM[3] P	0x1
Validate the TLB entry	MMU_CAM[2] V	0x1
Define the page size	MMU_CAM[1:0] PAGESIZE	0x-

20.4.1.3 Operational Modes Configuration

20.4.1.3.1 Main Sequence - Writing TLB Entries Statically

Writing TLB entries statically avoids the need to write translation tables in memory and is commonly used for relatively small address spaces. This method ensures that the translation of time-critical data accesses execute as fast as possible with entries already present in the TLB. These entries must be locked to prevent them from being overwritten.

Table 20-11. MMU Writing TLB Entries Statically

Step	Register/ Bit Field / Programming Model	Value
Execute software reset	MMU_SYSCONFIG[1] SOFTRESET	0x1
Wait for reset to complete	MMU_SYSSTATUS[0] RESETDONE	=0x1
Enable power saving via automatic interface clock gating	MMU_SYSCONFIG[0] AUTOIDLE	0x1
Configure TLB entries	See Table 20-10	
Load the physical Address of the page	MMU_RAM[31:12] PHYSICALADDRESS	0x-
Specify the TLB entry you want to write	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Load the specified entry in the TLB	MMU_LD_TLB[0] LDTLBITEM	0x1
Enable multihit fault and TLB miss	MMU_IRQENABLE[4] MULTIHITFAULT	0x1
	MMU_IRQENABLE[0] TLBMISS	0x1
Enable memory translations	MMU_CNTL[1] MMUENABLE	0x1

20.4.1.3.2 Main Sequence - Protecting TLB Entries

The first *n* TLB entries (with *n* < total number of TLB entries) can be protected from being overwritten with new translations. This is useful to ensure that certain commonly used or time-critical translations are always in the TLB and do not require retrieval using the table walking process.

Table 20-12. Protecting TLB Entries

Step	Registe/Bit Field/Programming Model	Value
Locks the TLB entries	MMU_LOCK[14:10] BASEVALUE	0x-

20.4.1.3.3 Main Sequence - Deleting TLB Entries

Two mechanisms exist to delete TLB entries. All unpreserved TLB entries, that is, TLB entries written with the preserved bit set to zero, can be deleted by invoking a TLB flush. The preserved bit should only be used on protected TLB entries, as it does not prevent replacement by the table walking logic.

Table 20-13. Deleting TLB Entries

Step	Registet / Bit Field / Programming Model	Value
Flush all nonprotected TLB entries	MMU_GFLUSH[0] GLOBALFLUSH	0x1
Flush all TLB entries specified by the CAM register	MMU_FLUSH_ENTRY[0] FLUSHENTRY	0x1

20.4.1.3.4 Main Sequence - Read TLB Entries

TLB entries can be read by the programmer to determine the TLB content at runtime.

Table 20-14. Read TLB Entries

Step	Register / Bit Field / Programming Model	Value
Set the current victim pointer	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Read RAM parts of the TLB entry	MMU_READ_RAM	
Read CAM parts of the TLB entry	MMU_READ_CAM	

20.5 MMU Register Manual

20.5.1 MMU Instance Summary

Table 20-15. MMU Instance Summary

Module Name	Base Address (L3/L4 Access)	Base Address (CPU Private Access)	Size
System MMU1	0x4881 C000	–	176 Bytes
System MMU2	0x4881 E000	–	176 Bytes
DSP1_MMU0	0x40D0 1000	0x01D0 1000	176 Bytes
DSP1_MMU1	0x40D0 2000	0x01D0 2000	176 Bytes
DSP2_MMU0	0x4150 1000	0x01D0 1000	176 Bytes
DSP2_MMU1	0x4150 2000	0x01D0 2000	176 Bytes
EVE1_MMU0	0x4208 1000	0x4008 1000	176 Bytes
EVE1_MMU1	0x4208 2000	0x4008 2000	176 Bytes
EVE2_MMU0	0x4218 1000	0x4008 1000	176 Bytes
EVE2_MMU1	0x4218 2000	0x4008 2000	176 Bytes
IPU1_MMU	0x5888 2000	0x5508 2000	176 Bytes
IPU2_MMU	0x5508 2000	0x5508 2000	176 Bytes

20.5.2 MMU Registers

20.5.2.1 MMU Register Summary

Table 20-16. System MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	System MMU1 Physical Address (L4_PER3 Access)	System MMU2 Physical Address (L4_PER3 Access)
MMU_REVISION	R	32	0x0000 0000	0x4881 C000	0x4881 E000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4881 C010	0x4881 E010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4881 C014	0x4881 E014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4881 C018	0x4881 E018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4881 C01C	0x4881 E01C
MMU_WALKING_ST	R	32	0x0000 0040	0x4881 C040	0x4881 E040
MMU_CNTL	RW	32	0x0000 0044	0x4881 C044	0x4881 E044
MMU_FAULT_AD	R	32	0x0000 0048	0x4881 C048	0x4881 E048
MMU_TTB	RW	32	0x0000 004C	0x4881 C04C	0x4881 E04C
MMU_LOCK	RW	32	0x0000 0050	0x4881 C050	0x4881 E050
MMU_LD_TLB	RW	32	0x0000 0054	0x4881 C054	0x4881 E054
MMU_CAM	RW	32	0x0000 0058	0x4881 C058	0x4881 E058
MMU_RAM	RW	32	0x0000 005C	0x4881 C05C	0x4881 E05C
MMU_GFLUSH	RW	32	0x0000 0060	0x4881 C060	0x4881 E060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4881 C064	0x4881 E064
MMU_READ_CAM	R	32	0x0000 0068	0x4881 C068	0x4881 E068
MMU_READ_RAM	R	32	0x0000 006C	0x4881 C06C	0x4881 E06C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4881 C070	0x4881 E070
RESERVED	R	32	0x0000 0074	0x4881 C074	0x4881 E074
RESERVED	R	32	0x0000 0078	0x4881 C078	0x4881 E078
RESERVED	R	32	0x0000 007C	0x4881 C07C	0x4881 E07C
MMU_FAULT_PC	R	32	0x0000 0080	0x4881 C080	0x4881 E080

Table 20-16. System MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	System MMU1 Physical Address (L4_PER3 Access)	System MMU2 Physical Address (L4_PER3 Access)
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4881 C084	0x4881 E084
MMU_GPR	RW	32	0x0000 0088	0x4881 C088	0x4881 E088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4881 C090	0x4881 E090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4881 C094	0x4881 E094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4881 C098	0x4881 E098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4881 C09C	0x4881 E09C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4881 C0A0	0x4881 E0A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4881 C0A4	0x4881 E0A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4881 C0A8	0x4881 E0A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4881 C0AC	0x4881 E0AC

Table 20-17. DSP1 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_MMU0 Physical Address (L3_MAIN Access)	DSP1_MMU0 Physical Address (DSP1 Private Access)	DSP1_MMU1 Physical Address (L3_MAIN Access)	DSP1_MMU1 Physical Address (DSP1 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x40D0 1000	0x01D0 1000	0x40D0 2000	0x01D0 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x40D0 1010	0x01D0 1010	0x40D0 2010	0x01D0 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x40D0 1014	0x01D0 1014	0x40D0 2014	0x01D0 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x40D0 1018	0x01D0 1018	0x40D0 2018	0x01D0 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x40D0 101C	0x01D0 101C	0x40D0 201C	0x01D0 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x40D0 1040	0x01D0 1040	0x40D0 2040	0x01D0 2040
MMU_CNTL	RW	32	0x0000 0044	0x40D0 1044	0x01D0 1044	0x40D0 2044	0x01D0 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x40D0 1048	0x01D0 1048	0x40D0 2048	0x01D0 2048
MMU_TTB	RW	32	0x0000 004C	0x40D0 104C	0x01D0 104C	0x40D0 204C	0x01D0 204C
MMU_LOCK	RW	32	0x0000 0050	0x40D0 1050	0x01D0 1050	0x40D0 2050	0x01D0 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x40D0 1054	0x01D0 1054	0x40D0 2054	0x01D0 2054
MMU_CAM	RW	32	0x0000 0058	0x40D0 1058	0x01D0 1058	0x40D0 2058	0x01D0 2058
MMU_RAM	RW	32	0x0000 005C	0x40D0 105C	0x01D0 105C	0x40D0 205C	0x01D0 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x40D0 1060	0x01D0 1060	0x40D0 2060	0x01D0 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x40D0 1064	0x01D0 1064	0x40D0 2064	0x01D0 2064
MMU_READ_CAM	R	32	0x0000 0068	0x40D0 1068	0x01D0 1068	0x40D0 2068	0x01D0 2068
MMU_READ_RAM	R	32	0x0000 006C	0x40D0 106C	0x01D0 106C	0x40D0 206C	0x01D0 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x40D0 1070	0x01D0 1070	0x40D0 2070	0x01D0 2070
RESERVED	R	32	0x0000 0074	0x40D0 1074	0x01D0 1074	0x40D0 2074	0x01D0 2074
RESERVED	R	32	0x0000 0078	0x40D0 1078	0x01D0 1078	0x40D0 2078	0x01D0 2078
RESERVED	R	32	0x0000 007C	0x40D0 107C	0x01D0 107C	0x40D0 207C	0x01D0 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x40D0 1080	0x01D0 1080	0x40D0 2080	0x01D0 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x40D0 1084	0x01D0 1084	0x40D0 2084	0x01D0 2084
MMU_GPR	RW	32	0x0000 0088	0x40D0 1088	0x01D0 1088	0x40D0 2088	0x01D0 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x40D0 1090	0x01D0 1090	0x40D0 2090	0x01D0 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x40D0 1094	0x01D0 1094	0x40D0 2094	0x01D0 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x40D0 1098	0x01D0 1098	0x40D0 2098	0x01D0 2098

Table 20-17. DSP1 MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_MMU0 Physical Address (L3_MAIN Access)	DSP1_MMU0 Physical Address (DSP1 Private Access)	DSP1_MMU1 Physical Address (L3_MAIN Access)	DSP1_MMU1 Physical Address (DSP1 Private Access)
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x40D0 109C	0x01D0 109C	0x40D0 209C	0x01D0 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x40D0 10A0	0x01D0 10A0	0x40D0 20A0	0x01D0 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x40D0 10A4	0x01D0 10A4	0x40D0 20A4	0x01D0 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x40D0 10A8	0x01D0 10A8	0x40D0 20A8	0x01D0 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x40D0 10AC	0x01D0 10AC	0x40D0 20AC	0x01D0 20AC

Table 20-18. DSP2 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_MMU0 Physical Address (L3_MAIN Access)	DSP2_MMU0 Physical Address (DSP2 Private Access)	DSP2_MMU1 Physical Address (L3_MAIN Access)	DSP2_MMU1 Physical Address (DSP2 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4150 1000	0x01D0 1000	0x4150 2000	0x01D0 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4150 1010	0x01D0 1010	0x4150 2010	0x01D0 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4150 1014	0x01D0 1014	0x4150 2014	0x01D0 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4150 1018	0x01D0 1018	0x4150 2018	0x01D0 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4150 101C	0x01D0 101C	0x4150 201C	0x01D0 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4150 1040	0x01D0 1040	0x4150 2040	0x01D0 2040
MMU_CNTL	RW	32	0x0000 0044	0x4150 1044	0x01D0 1044	0x4150 2044	0x01D0 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4150 1048	0x01D0 1048	0x4150 2048	0x01D0 2048
MMU_TTB	RW	32	0x0000 004C	0x4150 104C	0x01D0 104C	0x4150 204C	0x01D0 204C
MMU_LOCK	RW	32	0x0000 0050	0x4150 1050	0x01D0 1050	0x4150 2050	0x01D0 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4150 1054	0x01D0 1054	0x4150 2054	0x01D0 2054
MMU_CAM	RW	32	0x0000 0058	0x4150 1058	0x01D0 1058	0x4150 2058	0x01D0 2058
MMU_RAM	RW	32	0x0000 005C	0x4150 105C	0x01D0 105C	0x4150 205C	0x01D0 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4150 1060	0x01D0 1060	0x4150 2060	0x01D0 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4150 1064	0x01D0 1064	0x4150 2064	0x01D0 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4150 1068	0x01D0 1068	0x4150 2068	0x01D0 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4150 106C	0x01D0 106C	0x4150 206C	0x01D0 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4150 1070	0x01D0 1070	0x4150 2070	0x01D0 2070
RESERVED	R	32	0x0000 0074	0x4150 1074	0x01D0 1074	0x4150 2074	0x01D0 2074
RESERVED	R	32	0x0000 0078	0x4150 1078	0x01D0 1078	0x4150 2078	0x01D0 2078
RESERVED	R	32	0x0000 007C	0x4150 107C	0x01D0 107C	0x4150 207C	0x01D0 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4150 1080	0x01D0 1080	0x4150 2080	0x01D0 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4150 1084	0x01D0 1084	0x4150 2084	0x01D0 2084
MMU_GPR	RW	32	0x0000 0088	0x4150 1088	0x01D0 1088	0x4150 2088	0x01D0 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4150 1090	0x01D0 1090	0x4150 2090	0x01D0 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4150 1094	0x01D0 1094	0x4150 2094	0x01D0 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4150 1098	0x01D0 1098	0x4150 2098	0x01D0 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4150 109C	0x01D0 109C	0x4150 209C	0x01D0 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4150 10A0	0x01D0 10A0	0x4150 20A0	0x01D0 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4150 10A4	0x01D0 10A4	0x4150 20A4	0x01D0 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4150 10A8	0x01D0 10A8	0x4150 20A8	0x01D0 20A8

Table 20-18. DSP2 MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_MMU0 Physical Address (L3_MAIN Access)	DSP2_MMU0 Physical Address (DSP2 Private Access)	DSP2_MMU1 Physical Address (L3_MAIN Access)	DSP2_MMU1 Physical Address (DSP2 Private Access)
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4150 10AC	0x01D0 10AC	0x4150 20AC	0x01D0 20AC

Table 20-19. EVE1 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_MMU0 Physical Address (L3_MAIN Access)	EVE1_MMU0 Physical Address (EVE1 Private Access)	EVE1_MMU1 Physical Address (L3_MAIN Access)	EVE1_MMU1 Physical Address (EVE1 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4208 1000	0x4008 1000	0x4208 2000	0x4008 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4208 1010	0x4008 1010	0x4208 2010	0x4008 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4208 1014	0x4008 1014	0x4208 2014	0x4008 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4208 1018	0x4008 1018	0x4208 2018	0x4008 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4208 101C	0x4008 101C	0x4208 201C	0x4008 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4208 1040	0x4008 1040	0x4208 2040	0x4008 2040
MMU_CNTL	RW	32	0x0000 0044	0x4208 1044	0x4008 1044	0x4208 2044	0x4008 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4208 1048	0x4008 1048	0x4208 2048	0x4008 2048
MMU_TTB	RW	32	0x0000 004C	0x4208 104C	0x4008 104C	0x4208 204C	0x4008 204C
MMU_LOCK	RW	32	0x0000 0050	0x4208 1050	0x4008 1050	0x4208 2050	0x4008 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4208 1054	0x4008 1054	0x4208 2054	0x4008 2054
MMU_CAM	RW	32	0x0000 0058	0x4208 1058	0x4008 1058	0x4208 2058	0x4008 2058
MMU_RAM	RW	32	0x0000 005C	0x4208 105C	0x4008 105C	0x4208 205C	0x4008 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4208 1060	0x4008 1060	0x4208 2060	0x4008 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4208 1064	0x4008 1064	0x4208 2064	0x4008 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4208 1068	0x4008 1068	0x4208 2068	0x4008 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4208 106C	0x4008 106C	0x4208 206C	0x4008 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4208 1070	0x4008 1070	0x4208 2070	0x4008 2070
RESERVED	R	32	0x0000 0074	0x4208 1074	0x4008 1074	0x4208 2074	0x4008 2074
RESERVED	R	32	0x0000 0078	0x4208 1078	0x4008 1078	0x4208 2078	0x4008 2078
RESERVED	R	32	0x0000 007C	0x4208 107C	0x4008 107C	0x4208 207C	0x4008 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4208 1080	0x4008 1080	0x4208 2080	0x4008 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4208 1084	0x4008 1084	0x4208 2084	0x4008 2084
MMU_GPR	RW	32	0x0000 0088	0x4208 1088	0x4008 1088	0x4208 2088	0x4008 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4208 1090	0x4008 1090	0x4208 2090	0x4008 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4208 1094	0x4008 1094	0x4208 2094	0x4008 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4208 1098	0x4008 1098	0x4208 2098	0x4008 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4208 109C	0x4008 109C	0x4208 209C	0x4008 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4208 10A0	0x4008 10A0	0x4208 20A0	0x4008 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4208 10A4	0x4008 10A4	0x4208 20A4	0x4008 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4208 10A8	0x4008 10A8	0x4208 20A8	0x4008 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4208 10AC	0x4008 10AC	0x4208 20AC	0x4008 20AC

Table 20-20. EVE2 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE2_MMU0 Physical Address (L3_MAIN Access)	EVE2_MMU0 Physical Address (EVE2 Private Access)	EVE2_MMU1 Physical Address (L3_MAIN Access)	EVE2_MMU1 Physical Address (EVE2 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4218 1000	0x4008 1000	0x4218 2000	0x4008 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4218 1010	0x4008 1010	0x4218 2010	0x4008 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4218 1014	0x4008 1014	0x4218 2014	0x4008 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4218 1018	0x4008 1018	0x4218 2018	0x4008 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4218 101C	0x4008 101C	0x4218 201C	0x4008 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4218 1040	0x4008 1040	0x4218 2040	0x4008 2040
MMU_CNTL	RW	32	0x0000 0044	0x4218 1044	0x4008 1044	0x4218 2044	0x4008 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4218 1048	0x4008 1048	0x4218 2048	0x4008 2048
MMU_TTB	RW	32	0x0000 004C	0x4218 104C	0x4008 104C	0x4218 204C	0x4008 204C
MMU_LOCK	RW	32	0x0000 0050	0x4218 1050	0x4008 1050	0x4218 2050	0x4008 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4218 1054	0x4008 1054	0x4218 2054	0x4008 2054
MMU_CAM	RW	32	0x0000 0058	0x4218 1058	0x4008 1058	0x4218 2058	0x4008 2058
MMU_RAM	RW	32	0x0000 005C	0x4218 105C	0x4008 105C	0x4218 205C	0x4008 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4218 1060	0x4008 1060	0x4218 2060	0x4008 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4218 1064	0x4008 1064	0x4218 2064	0x4008 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4218 1068	0x4008 1068	0x4218 2068	0x4008 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4218 106C	0x4008 106C	0x4218 206C	0x4008 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4218 1070	0x4008 1070	0x4218 2070	0x4008 2070
RESERVED	R	32	0x0000 0074	0x4218 1074	0x4008 1074	0x4218 2074	0x4008 2074
RESERVED	R	32	0x0000 0078	0x4218 1078	0x4008 1078	0x4218 2078	0x4008 2078
RESERVED	R	32	0x0000 007C	0x4218 107C	0x4008 107C	0x4218 207C	0x4008 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4218 1080	0x4008 1080	0x4218 2080	0x4008 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4218 1084	0x4008 1084	0x4218 2084	0x4008 2084
MMU_GPR	RW	32	0x0000 0088	0x4218 1088	0x4008 1088	0x4218 2088	0x4008 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4218 1090	0x4008 1090	0x4218 2090	0x4008 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4218 1094	0x4008 1094	0x4218 2094	0x4008 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4218 1098	0x4008 1098	0x4218 2098	0x4008 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4218 109C	0x4008 109C	0x4218 209C	0x4008 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4218 10A0	0x4008 10A0	0x4218 20A0	0x4008 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4218 10A4	0x4008 10A4	0x4218 20A4	0x4008 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4218 10A8	0x4008 10A8	0x4218 20A8	0x4008 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4218 10AC	0x4008 10AC	0x4218 20AC	0x4008 20AC

Table 20-21. IPU MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU1_MMU Physical Address (L3_MAIN Access)	IPU1_MMU Physical Address (IPU1 Private Access)	IPU2_MMU Physical Address (L3_MAIN Access)	IPU2_MMU Physical Address (IPU2 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x5888 2000	0x5508 2000	0x5508 2000	0x5508 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x5888 2010	0x5508 2010	0x5508 2010	0x5508 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x5888 2014	0x5508 2014	0x5508 2014	0x5508 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x5888 2018	0x5508 2018	0x5508 2018	0x5508 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x5888 201C	0x5508 201C	0x5508 201C	0x5508 201C

Table 20-21. IPU MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IPU1_MMU Physical Address (L3_MAIN Access)	IPU1_MMU Physical Address (IPU1 Private Access)	IPU2_MMU Physical Address (L3_MAIN Access)	IPU2_MMU Physical Address (IPU2 Private Access)
MMU_WALKING_ST	R	32	0x0000 0040	0x5888 2040	0x5508 2040	0x5508 2040	0x5508 2040
MMU_CNTL	RW	32	0x0000 0044	0x5888 2044	0x5508 2044	0x5508 2044	0x5508 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x5888 2048	0x5508 2048	0x5508 2048	0x5508 2048
MMU_TTB	RW	32	0x0000 004C	0x5888 204C	0x5508 204C	0x5508 204C	0x5508 204C
MMU_LOCK	RW	32	0x0000 0050	0x5888 2050	0x5508 2050	0x5508 2050	0x5508 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x5888 2054	0x5508 2054	0x5508 2054	0x5508 2054
MMU_CAM	RW	32	0x0000 0058	0x5888 2058	0x5508 2058	0x5508 2058	0x5508 2058
MMU_RAM	RW	32	0x0000 005C	0x5888 205C	0x5508 205C	0x5508 205C	0x5508 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x5888 2060	0x5508 2060	0x5508 2060	0x5508 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x5888 2064	0x5508 2064	0x5508 2064	0x5508 2064
MMU_READ_CAM	R	32	0x0000 0068	0x5888 2068	0x5508 2068	0x5508 2068	0x5508 2068
MMU_READ_RAM	R	32	0x0000 006C	0x5888 206C	0x5508 206C	0x5508 206C	0x5508 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x5888 2070	0x5508 2070	0x5508 2070	0x5508 2070
RESERVED	R	32	0x0000 0074	0x5888 2074	0x5508 2074	0x5508 2074	0x5508 2074
RESERVED	R	32	0x0000 0078	0x5888 2078	0x5508 2078	0x5508 2078	0x5508 2078
RESERVED	R	32	0x0000 007C	0x5888 207C	0x5508 207C	0x5508 207C	0x5508 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x5888 2080	0x5508 2080	0x5508 2080	0x5508 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x5888 2084	0x5508 2084	0x5508 2084	0x5508 2084
MMU_GPR	RW	32	0x0000 0088	0x5888 2088	0x5508 2088	0x5508 2088	0x5508 2088

20.5.2.2 MMU Register Description

Table 20-22. MMU_REVISION

Address Offset	0x0000 0000		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the IP revision code		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REVISION</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See [1]

(1) TI internal data

Table 20-23. MMU_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register controls the various parameters of the OCP interface		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

RESERVED	CLOCKACTIVITY	RESERVED	IDLEMODE	RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x000000
9:8	CLOCKACTIVITY	Clock activity during wake-up mode 0x0: Functional and OCP clocks can be switched off	R	0x0
7:5	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x0
4:3	IDLEMODE	Idle mode 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module 0x3: Reserved. Do not use	RW	0x0
2	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x0
1	SOFTRESET	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0 Write 0x0: No functional effect Write 0x1: The module is reset	W	0x0
0	AUTOIDLE	Internal OCP clock gating strategy 0x0: OCP clock is free-running 0x1: Automatic interconnect clock gating strategy is applied, based on the interconnect interface activity	RW	0x0

Table 20-24. MMU_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register provides status information about the module, excluding the interrupt status information		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x000000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset in on-going Read 0x1: Reset completed	R	-

Table 20-25. MMU_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This interrupt status register regroups all the status of the module internal events that can generate an interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											MULTIHITFAULT	TABLEWALKFAULT	EMUMISS	TRANSLATIONFAULT	TLBMISS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility. Read returns 0	R	0x0000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB Read 0x0: MultiHitFault false Write 0x0: MultiHitFault status bit unchanged Write 0x1: MultiHitFault status bit is reset Read 0x1: MultiHitFault is true ('pending')	RW (W1toClr)	0x0
3	TABLEWALKFAULT	Error response received during a Table Walk Read 0x0: TableWalkFault false Write 0x0: TableWalkFault status bit unchanged Write 0x1: TableWalkFault status bit is reset Read 0x1: TableWalkFault is true ('pending')	RW (W1toClr)	0x0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) Read 0x0: EMUMiss false Write 0x0: EMUMiss status bit unchanged Write 0x1: EMUMiss status bit is reset Read 0x1: EMUMiss is true ('pending')	RW (W1toClr)	0x0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) Read 0x0: TranslationFault false Write 0x0: TranslationFault status bit unchanged Write 0x1: TranslationFault status bit is reset Read 0x1: TranslationFault is true ('pending')	RW (W1toClr)	0x0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) Read 0x0: TLBMiss false Write 0x0: TLBMiss status bit unchanged Write 0x1: TLBMiss status bit is reset Read 0x1: TLBMiss is true ('pending')	RW (W1toClr)	0x0

Table 20-26. MMU_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MULTIHITFAULT	TABLEWALKFAULT	EMUMISS	TRANSLATIONFAULT	TLBMISS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility Read returns 0	R	0x0000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB 0x0: MultiHitFault is masked 0x1: MultiHitFault event generates an interrupt if occurs	RW	0x0
3	TABLEWALKFAULT	Error response received during a Table Walk 0x0: TableWalkFault is masked 0x1: TableWalkFault event generates an interrupt if occurs	RW	0x0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) 0x0: EMUMiss interrupt is masked 0x1: EMUMiss event generates an interrupt when it occurs	RW	0x0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) 0x0: TranslationFault is masked 0x1: TranslationFault event generates an interrupt if occurs	RW	0x0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) 0x0: TLBMiss interrupt is masked 0x1: TLBMiss event generates an interrupt when if occurs	RW	0x0

Table 20-27. MMU_WALKING_ST

Address Offset	0x0000 0040		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register provides status information about the table walking logic		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TWLRUNNING

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0	R	0x0000 0000
0	TWLRUNNING	Table Walking Logic is running Read 0x0: TWL Completed Read 0x1: TWL Running	R	0x0

Table 20-28. MMU_CNTL

Address Offset	0x0000 0044		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register programs the MMU features		
Type	RW		

31–4	3	2	1	0
RESERVED	EMUTLBUPDATE	TWLENABLE	MMUENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000000
3	EMUTLBUPDATE	Enable TLB update on emulator table walk 0x0: Emulator TLB update disabled 0x1: Emulator TLB update enabled	RW	0x0
2	TWLENABLE	Table Walking Logic enable 0x0: TWL disabled 0x1: TWL enabled	RW	0x0
1	MMUENABLE	MMU enable 0x0: MMU disabled 0x1: MMU enabled	RW	0x0
0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 20-29. MMU_FAULT_AD

Address Offset	0x0000 0048		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the virtual address that generated the interrupt		
Type	R		

31–0
FAULTADDRESS

Bits	Field Name	Description	Type	Reset
31:0	FAULTADDRESS	Virtual address of the access that generated a fault	R	0x0000 0000

Table 20-30. MMU_TTB

Address Offset	0x0000 004C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the Translation Table Base address		
Type	RW		

31–7	6–0
TTBADDRESS	RESERVED

Bits	Field Name	Description	Type	Reset
31:7	TTBADDRESS	Translation Table Base Address	RW	0x0000000
6:0	RESERVED	Write 0's for future compatibility. Reads retiurn 0	R	0x00

Table 20-31. MMU_LOCK

Address Offset	0x0000 0050		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register locks some of the TLB entries		
Type	RW		

31:15	14:10	9	8:4	3:0
RESERVED	BASEVALUE	RESERVED	CURRENTVICTIM	RESERVED

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00000
14:10	BASEVALUE	Locked entries base value.	RW	0x00
9	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
8:4	CURRENTVICTIM	Current entry to be updated either by the TWL or by the software. Write value : TLB entry to be updated by software Read value : TLB entry that will be updated by table walk logic. This will be same as BASEVALUE when there are no tablewalks.	RW	0x00
3:0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 20-32. MMU_LD_TLB

Address Offset	0x0000 0054		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register loads a TLB entry (CAM+RAM)		
Type	RW		

31:1	0
RESERVED	LDTLBITEM

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	LDTLBITEM	Write (load) data in the TLB. Reads return 0. Write 0x0: No functional effect Write 0x1: Load TLB data	W	0x0

Table 20-33. MMU_CAM

Address Offset	0x0000 0058		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register holds a CAM entry		
Type	RW		

31:12	11:4	3	2	1:0
VATAG	RESERVED	P	V	PAGESIZE

Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	RW	0x00000
11:4	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
3	P	Preserved bit 0x0: TLB entry may be flushed 0x1: TLB entry is protected against flush	RW	0x0
2	V	Valid bit 0x0: TLB entry is invalid 0x1: TLB entry is valid	RW	0x0
1:0	PAGESIZE	Page size 0x0: Section (1 MiB) 0x1: Large page (64 KiB) 0x2: Small page (4 KiB) 0x3: Supersection (16 MiB)	RW	0x0

Table 20-34. MMU_RAM

Address Offset	0x0000 005C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains bits [31:12] of the physical address to be written to a TLB entry pointed to by CURRENTVICTIM field of MMU_LOCK register.		
Type	RW		

31:12	11:0
PHYSICALADDRESS	RESERVED

Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	RW	0x00000
11:0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 20-35. MMU_GFLUSH

Address Offset	0x0000 0060		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register flushes all the non-protected TLB entries		
Type	RW		

31:1	0
RESERVED	GLOBALFLUSH

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	GLOBALFLUSH	Flush all the non-protected TLB entries when set. Reads return 0. Write 0x0: No functional effect Write 0x1: Flush all the non-protected TLB entries	W	0x0

Table 20-36. MMU_FLUSH_ENTRY

Address Offset	0x0000 0064

Table 20-36. MMU_FLUSH_ENTRY (continued)

Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register flushes the entry pointed to by the CAM virtual address		
Type	RW		

31:1	0
RESERVED	FLUSHENTRY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	FLUSHENTRY	Flush the TLB entry pointed by the virtual address (VATag) in MMU_CAM register, even if this entry is set protected. Reads return 0. Write 0x0: No functional effect Write 0x1: Flush all the TLB entries specified by the CAM register	W	0x0

Table 20-37. MMU_READ_CAM

Address Offset	0x0000 0068		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register reads CAM data from a CAM entry		
Type	R		

31:12	11:4	3	2	1:0
VATAG	RESERVED	P	V	PAGESIZE

Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	R	0x00000
11:4	RESERVED	Reads return 0	R	0x00
3	P	Preserved bit Read 0x0: TLB entry may be flushed Read 0x1: TLB entry is protected against flush	R	0x0
2	V	Valid bit Read 0x0: TLB entry is invalid Read 0x1: TLB entry is valid	R	0x0
1:0	PAGESIZE	Page size Read 0x0: Section (1 MiB) Read 0x1: Large page (64 KiB) Read 0x2: Small page (4 KiB) Read 0x3: Supersection (16 MiB)	R	0x0

Table 20-38. MMU_READ_RAM

Address Offset	0x0000 006C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register reads bits [31:12] of the physical address from the TLB entry pointed to by CURRENTVICTIM field of the MMU_LOCK register.		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="20">PHYSICALADDRESS</td><td colspan="12">RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	R	0x00000
11:0	RESERVED	Reads return 0	R	0x0

Table 20-39. MMU_EMU_FAULT_AD

Address Offset	0x0000 0070		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the last virtual address of a fault caused by the debugger		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">EMUFAULTADDRESS</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	EMUFAULTADDRESS	Virtual address of the last emulator access that generated a fault	R	0x0000 0000

Table 20-40. MMU_FAULT_PC

Address Offset	0x0000 0080		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	Typically CPU program counter value of instruction generating MMU fault. The address value is captured at MMU_EMU_FAULT_AD[31:0] EMUFAULTADDRESS. Data-Read-access : corresponding PC. Data-write-access : not perfect accuracy due to posted-write.		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">PC</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	PC	Typically CPU program counter value of instruction generating MMU fault	R	0x0000 0000

Table 20-41. MMU_FAULT_STATUS

Address Offset	0x0000 0084		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	Fault status register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td colspan="5">MMU_FAULT_TRANS_ID</td><td>RD_WR</td><td colspan="2">MMU_FAULT_TYPE</td><td>FAULTINDICATION</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000

Bits	Field Name	Description	Type	Reset
8:4	MMU_FAULT_TRANS_ID	MtagID of the transaction that caused fault	R	0x0
3	RD_WR	Indicates read or write 0x0: Write 0x1: Read	R	0x0
2:1	MMU_FAULT_TYPE	MReqInfo[1:0] is captured as fault type	R	0x0
0	FAULTINDICATION	Indicates an MMU fault	RW (W1toClr)	0x0

Table 20-42. MMU_GPR

Address Offset	0x0000 0088		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	General purpose register		
Type	RW		

31–16	15–1	0
GPO	RESERVED	FAULT_INTR_DIS

Bits	Field Name	Description	Type	Reset
31:16	GPO	General purpose output sent out as MMU output	RW	0x0000
15:1	RESERVED	Reserved	R	0x0000
0	FAULT_INTR_DIS	Disable generation of interrupt on fault. Error response is returned instead on the slave port	RW	0x0

Table 20-43. MMU_BYPASS_REGION1_ADDR

Address Offset	0x0000 0090		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the start address of the first NO TRANSLATION REGION		
Type	RW		

31–16	15–0
START_ADDR	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION1_SIZE	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-44. MMU_BYPASS_REGION1_SIZE

Address Offset	0x0000 0094		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of first NO TRANSLATION REGION		
Type	RW		

31–4	3–0
RESERVED	SIZE

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 20-45. MMU_BYPASS_REGION2_ADDR

Address Offset	0x0000 0098		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the start address of the second NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-46. MMU_BYPASS_REGION2_SIZE

Address Offset	0x0000 009C		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of second NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 20-47. MMU_BYPASS_REGION3_ADDR

Address Offset	0x0000 00A0		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the start address of the third NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-48. MMU_BYPASS_REGION3_SIZE

Address Offset	0x0000 00A4		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of third NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 20-49. MMU_BYPASS_REGION4_ADDR

Address Offset	0x0000 00A8		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the start address of the fourth NO TRANSLATION REGION		
Type	RW		

31–16	15–0
START_ADDR	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 20-50. MMU_BYPASS_REGION4_SIZE

Address Offset	0x0000 00AC		
Physical Address	See Section 20.5.2.1	**Instance**	See Table 20-15
Description	This register contains the size of fourth NO TRANSLATION REGION		
Type	RW		

31–4	3–0
RESERVED	SIZE

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Chapter 21
Spinlock

This chapter describes the Spinlock module of the device.

21.1 Spinlock Overview 5188
21.2 Spinlock Integration 5189
21.3 Spinlock Functional Description 5190
21.4 Spinlock Programming Guide 5192
21.5 Spinlock Register Manual 5194

21.1 Spinlock Overview

The Spinlock module provides hardware assistance for synchronizing the processes running on multiple processors in the device:

- Dual Cortex®-A15 microprocessor unit (MPU) subsystem
- Digital signal processor (DSP) subsystems – DSP1 and DSP2
- Dual Cortex-M4 image processing unit (IPU) subsystems – IPU1 and IPU2

The Spinlock module implements 256 spinlocks (or hardware semaphores), which provide an efficient way to perform a lock operation of a device resource using a single read-access, avoiding the need of a read-modify-write bus transfer that the programmable cores are not capable of.

Figure 21-1 shows an overview of the Spinlock module.

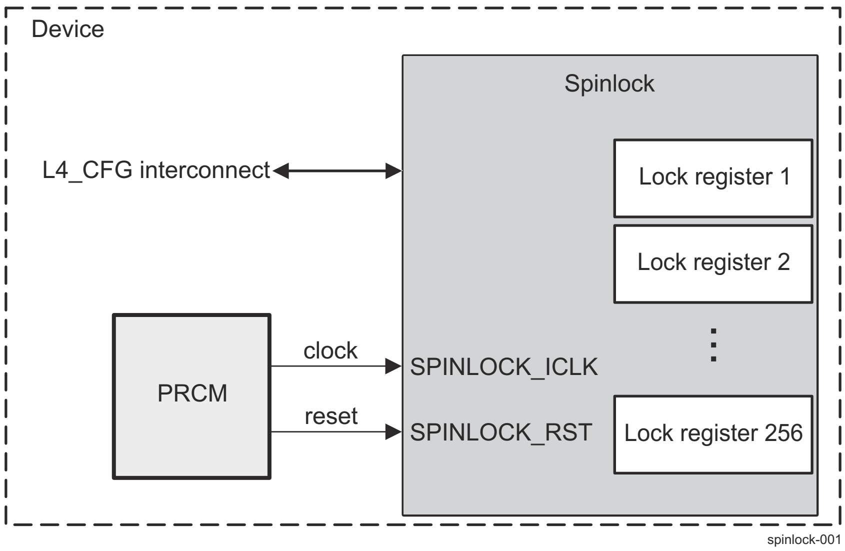


Figure 21-1. Spinlock Overview

21.2 Spinlock Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 21-2 shows the Spinlock integration.

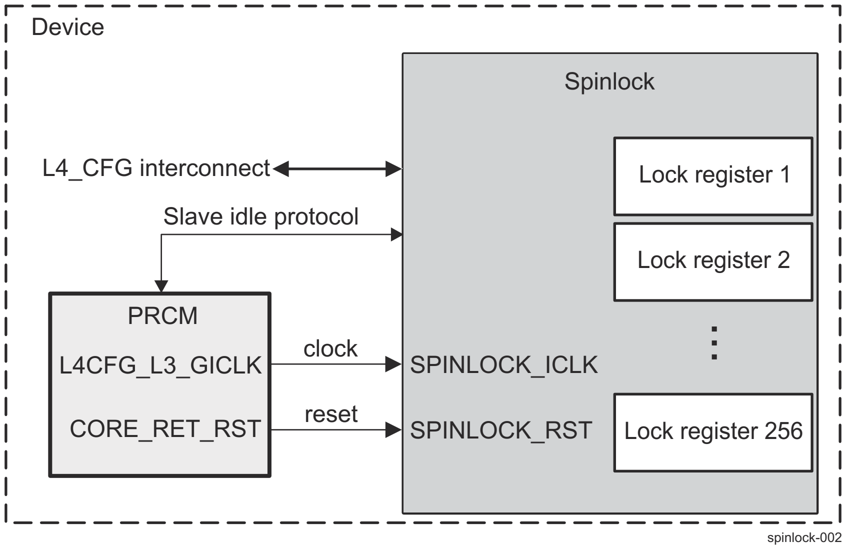


Figure 21-2. Spinlock Integration

Note

For more information about the Slave idle protocol and the wake-up request, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

Table 21-1 and Table 21-2 summarize the integration of the module in the device.

Table 21-1. Spinlock Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
SPINLOCK	PD_COREAON	L4_CFG

Table 21-2. Spinlock Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
SPINLOCK	SPINLOCK_ICLK	L4CFG_L3_GICLK	PRCM	Spinlock interface/functional clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
SPINLOCK	SPINLOCK_RST	CORE_RET_RST	PRCM	Spinlock hardware reset. This reset is asynchronously applied to the Spinlock internal registers.

Note

The Spinlock module does not support any interrupt and DMA requests.

21.3 Spinlock Functional Description

21.3.1 Spinlock Software Reset

The Spinlock module can be reset by software through the SPINLOCK_SYSCONFIG[1] SOFTRESET bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. The SPINLOCK_SYSTATUS[0] RESETDONE bit can be polled to check the reset status (reading 1 indicates that reset sequence is done; reading 0 indicates that reset sequence is in progress). The software must ensure that the software reset completes before doing Spinlock operations.

21.3.2 Spinlock Power Management

Table 21-3 describes power-management features available to the Spinlock module.

Table 21-3. Spinlock Local Power Management Features

Feature	Registers	Description
Clock auto gating	SPINLOCK_SYSCONFIG[0] AUTOGATING bit	This bit indicates that the Spinlock module uses an automatic internal interface clock gating strategy, based on interface activity.
Global wake-up enable	SPINLOCK_SYSCONFIG[2] ENAWAKEUP bit	This bit indicates that the wake-up generation feature (at Spinlock module level) is disabled.
Slave idle modes	SPINLOCK_SYSCONFIG[4:3] SIDLEMODE bit field	This bit field indicates that the Spinlock module uses smart-idle mode.

Note

All Spinlock local power management features are non-configurable – that is, their respective bit fields are read-only and only show the actual hardware implementation.

For information about source clock gating and sleep/wake-up transitions description, see *Clock Domain-Level Clock Management*, in *Power, Reset, and Clock Management*. For descriptions of EnaWakeUp, and IdleMode features, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

The Spinlock module is normally idle, except when processing a request from its slave interface port. The smart-idle mode acknowledges idle requests from the PRCM only when the module is prepared to go idle. The Spinlock module is always ready to go idle if it does not have any request that it is processing.

21.3.3 About Spinlocks

Spinlocks are present to solve the need for synchronization and mutual exclusion between heterogeneous processors and those not operating under a single, shared operating system. There is no alternative mechanism to accomplish these operations between processors in separate subsystems.

Spinlocks are not the best way to synchronize between tasks or threads on one CPU. Instead, spinlocks are for use in synchronization between different subsystems in the device that don't have any other means of hardware-based synchronization.

Spinlocks do not solve all system synchronization issues. They have limited applicability and should be used with care to implement higher level synchronization protocols.

A spinlock is appropriate for mutual exclusion for access to a shared data structure. It should be used only when:

1. The time to hold the lock is predictable and small (for example, a maximum hold time of less than 200 CPU cycles may be acceptable).
2. The locking task cannot be preempted, suspended, or interrupted while holding the lock (this would make the hold time large and unpredictable).
3. The lock is lightly contended, that is the chance of any other process (or processor) trying to acquire the lock while it is held is small.

If these conditions are met, then the locking code can retry a failed attempt to acquire the lock until success.

If the conditions are not met, then a spinlock is not a good candidate. One alternative is to use a spinlock for critical section control (engineered to meet the conditions) to implement a higher level semaphore that can support preemption, notification, timeout or other higher level properties.

21.3.4 Spinlock Functional Operation

The Spinlock module supports 256 spinlocks. It accepts only a single command at a time and processes the command fully before accepting the next command. A lock is requested by reading the SPINLOCK_LOCK_REG_i[0] TAKEN bit. There are two states: Taken (SPINLOCK_LOCK_REG_i[0] TAKEN = 1) or Not Taken (SPINLOCK_LOCK_REG_i[0] TAKEN = 0).

When the status of lock *i* (where i = 0 to 255) is Not Taken (free), a read from the SPINLOCK_LOCK_REG_i register returns 0 and sets the lock to Taken (locked). When the status of lock *i* is Taken, a read returns 1 and does not change the state of the lock.

A write to the SPINLOCK_LOCK_REG_i register does not change the state of lock, unless when writing 0 when the lock is in Taken state. By doing this, the requester frees the lock.

CAUTION

Only 32-bit reads and writes are supported.

Figure 21-3 shows the SPINLOCK_LOCK_REG_i register state diagram.

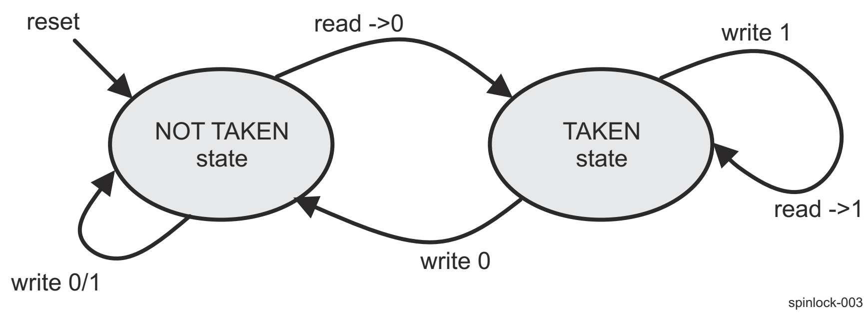


Figure 21-3. Lock Register State Diagram

Note

- There is no support to ensure that a lock register is locked and unlocked by the same process. This must be ensured in software.
- There is no support to check that the same initiator that acquired the lock is the one that is freeing the lock.

21.4 Spinlock Programming Guide

21.4.1 Spinlock Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

21.4.1.1 Surrounding Modules Global Initialization

This procedure initializes the surrounding modules when the Spinlock module is used for the first time after a device reset.

Table 21-4. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Spinlock interface clock must be enabled. For more information, see *CD_L4_CFG Clock Domain*, in *Power, Reset, and Clock Management*.
Interconnect	For more information about the L4_CFG interconnect configuration, see *L4 Interconnects*.

21.4.1.2 Basic Spinlock Operations

The main spinlock operations are:

- Clear all the Taken spinlocks (only after a system bug recovery)
- Take a spinlock
- Release spinlock

21.4.1.2.1 Spinlocks Clearing After a System Bug Recovery

Module initialization (after reset) is not needed, except after system bug recovery. The following table presents the Spinlock initialization after a system bug recovery. Software should store 0 into each of the SPINLOCK_LOCK_REG_i registers at system startup to insure that all locks are initialized to Not Taken.

Table 21-5. Spinlock System Bug Recovery

Step	Register	Value
IF: SPINLOCK_SYSTATUS[0] IU0 = = 1?	SPINLOCK_SYSTATUS[0] IU0	
Free the 256 locks	SPINLOCK_LOCK_REG_i[0] TAKEN (i = 0 to 255)	0x0
END		

21.4.1.2.2 Take and Release Spinlock

This procedure configures the take and release (free) operations for the Spinlock module. A spinlock should only be held with interrupts disabled. So, before attempting to obtain the spinlock, software should disable interrupts. Then it should read the SPINLOCK_LOCK_REG_i[0] TAKEN bit to attempt to obtain the lock. If it succeeds, it should proceed directly through the critical section then unlock and re-enable interrupts. If the acquisition attempt fails, the acquisition should be reattempted. To prevent unknown interrupt disabled time, interrupts should be re-enabled and then disabled before reattempting to acquire the lock. Figure 21-4 shows the described above procedure.

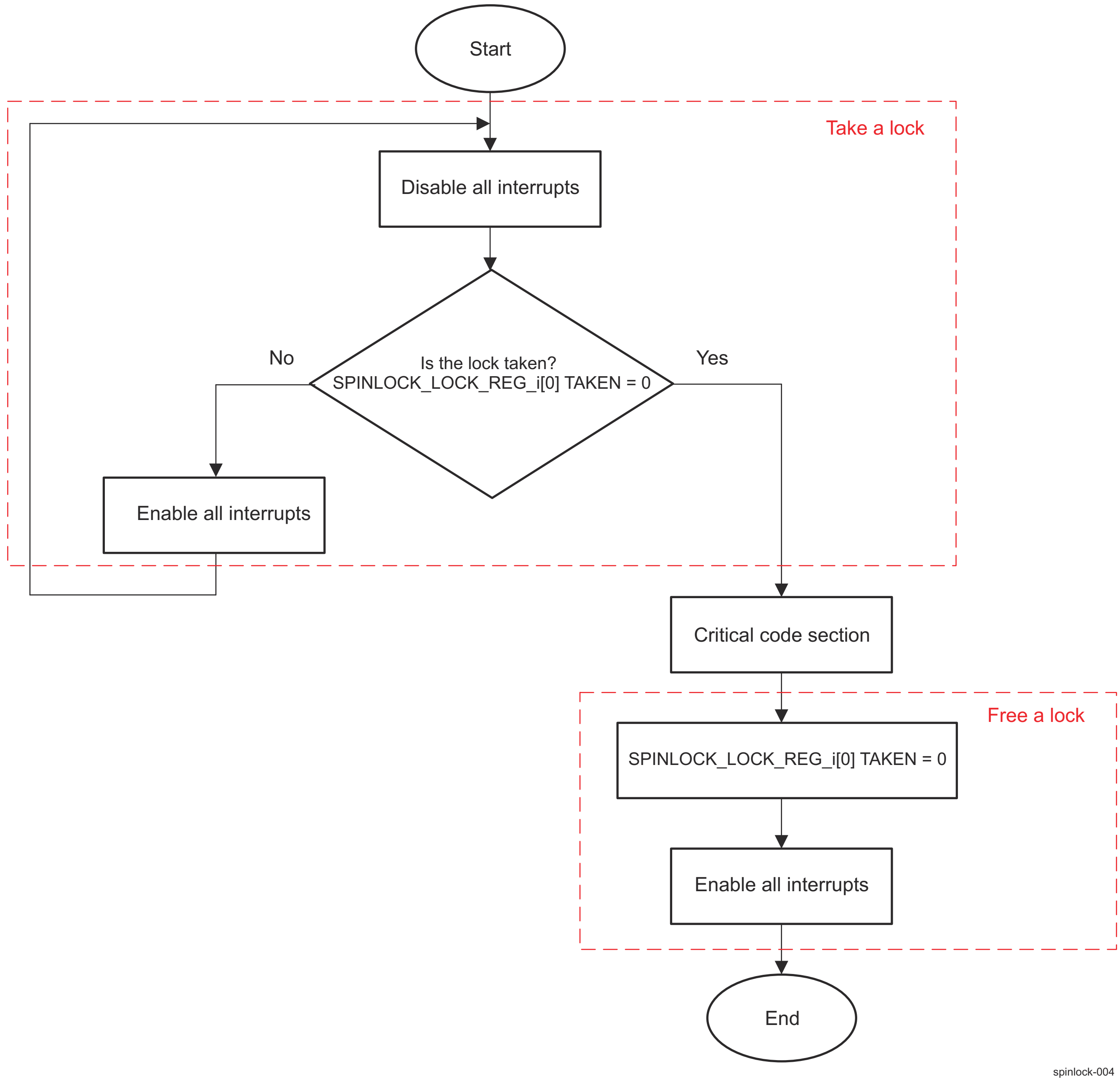


Figure 21-4. Take and Release Spinlock

Table 21-6. Register Call Summary

Register Name
SPINLOCK_LOCK_REG_i[0] TAKEN

Table 21-7. Subprocess Call Summary

Subprocess Name	Cross Reference
Disable Interrupts	For information about disabling/enabling interrupts in MPU_INTC, see the Arm *Cortex-A15 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp). For information about disabling/enabling interrupts in IPU_INTC, see the Arm *Cortex-M4 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp). For information about disabling/enabling interrupts in DSP_INTC, see *DSP Subsystem*.
Enable Interrupts	

21.5 Spinlock Register Manual

21.5.1 Spinlock Instance Summary

Table 21-8. Spinlock Instance Summary

Module Name	L4_CFG Base Address	Size
Spinlock	0x4A0F 6000	4KiB

21.5.2 Spinlock Registers

21.5.2.1 Spinlock Register Summary

Table 21-9. Spinlock Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG Physical Address
SPINLOCK_REVISION	R	32	0x0000 0000	0x4A0F 6000
SPINLOCK_SYSCONFIG	RW	32	0x0000 0010	0x4A0F 6010
SPINLOCK_SYSTATUS	R	32	0x0000 0014	0x4A0F 6014
SPINLOCK_LOCK_REG_i [(1)]	RW	32	0x0000 0800 + (0x4 * i)	0x4A0F 6800 + (0x4 * i)

(1) i = 0 to 255

21.5.2.2 Spinlock Register Description

Table 21-10. SPINLOCK_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A0F 6000	**Instance**	Spinlock
Description	This register contains the IP revision code		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REVISION</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 21-11. SPINLOCK_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A0F 6010	**Instance**	Spinlock
Description	This register controls the various parameters of the OCP interface. Note that most fields are read-only.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="27">RESERVED</td><td colspan="2">SIDLEMODE</td><td>ENWAKEUP</td><td>SOFTRESET</td><td>AUTOGATING</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved. Reads return 0.	R	0x0000000

Bits	Field Name	Description	Type	Reset
4:3	SIDLEMODE	Slave interface power management (IDLE request/ acknowledgement control). Read 0x0: Force-idle. IDLE request is acknowledged unconditionally and immediately. Read 0x1: No-idle. IDLE request is never acknowledged. Read 0x2: Smart-idle. IDLE request acknowledgement is based on the internal module activity. Read 0x3: Reserved. Do not use.	R	0x2
2	ENWAKEUP	Asynchronous wakeup gereration. Read 0x0: Wakeup generation is disabled. Read 0x1: Wakeup generation is enabled.	R	0
1	SOFTRESET	Module software reset. Write 0x0: No action Write 0x1: Start soft reset sequence	W	0
0	AUTOGATING	Internal interface clock gating strategy. Read 0x0: Interface clock is not gated when the interface is idle. Read 0x1: Automatic internal OCP clock gating strategy is applied, based on the OCP interface activity.	R	1

Table 21-12. SPINLOCK_SYSTATUS

Address Offset	0x0000 0014		
Physical Address	0x4A0F 6014	**Instance**	Spinlock
Description	This register provides status information about this instance of the Spinlock module.		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">NUMLOCKS</td><td colspan="8">RESERVED</td><td>IU7</td><td>IU6</td><td>IU5</td><td>IU4</td><td>IU3</td><td>IU2</td><td>IU1</td><td>IU0</td><td colspan="7">RESERVED</td><td>RESETDONE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:24	NUMLOCKS	Number of lock registers implemeted. Read 0x1: This instance has 32 lock registers. Read 0x2: This instance has 64 lock registers. Read 0x4: This instance has 128 lock registers. Read 0x8: This instance has 256 lock registers.	R	0x08
23:16	RESERVED	Reserved. Reads return 0.	R	0x00
15	IU7	In-Use flag 0, covering lock registers 224 - 255. Read 0x0: All lock registers 224 - 255 are in the Not Taken state. Read 0x1: At least one of the lock registers 224 - 255 is in the Taken state.	R	0
14	IU6	In-Use flag 0, covering lock registers 192 - 223. Read 0x0: All lock registers 192 - 223 are in the Not Taken state. Read 0x1: At least one of the lock registers 192 - 223 is in the Taken state.	R	0

Bits	Field Name	Description	Type	Reset
13	IU5	In-Use flag 0, covering lock registers 160 - 191. Read 0x0: All lock registers 160 - 191 are in the Not Taken state. Read 0x1: At least one of the lock registers 160 - 191 is in the Taken state.	R	0
12	IU4	In-Use flag 0, covering lock registers 128 - 159. Read 0x0: All lock registers 128 - 159 are in the Not Taken state. Read 0x1: At least one of the lock registers 128 - 159 is in the Taken state.	R	0
11	IU3	In-Use flag 0, covering lock registers 96 - 127. Read 0x0: All lock registers 96 - 127 are in the Not Taken state. Read 0x1: At least one of the lock registers 96 - 127 is in the Taken state.	R	0
10	IU2	In-Use flag 0, covering lock registers 64 - 95. Read 0x0: All lock registers 64 - 95 are in the Not Taken state. Read 0x1: At least one of the lock registers 64 - 95 is in the Taken state.	R	0
9	IU1	In-Use flag 0, covering lock registers 32 - 63. Read 0x0: All lock registers 32 - 63 are in the Not Taken state. Read 0x1: At least one of the lock registers 32 - 63 is in the Taken state.	R	0
8	IU0	In-Use flag 0, covering lock registers 0 - 31. Read 0x0: All lock registers 0 - 31 are in the Not Taken state. Read 0x1: At least one of the lock registers 0 - 31 is in the Taken state.	R	0
7:1	RESERVED	Reserved. Reads return 0.	R	0x00
0	RESETDONE	Reset done status. Read 0x0: Reset in progress. Read 0x1: Reset is completed.	R	1

Table 21-13. SPINLOCK_LOCK_REG_i

Address Offset	0x0000 0800	**index**	i = 0 to 255
Physical Address	0x4A0F 6800 + (0x4 * i)	**Instance**	Spinlock
Description	This register contains the state of one lock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TAKEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved. Reads return 0. Writes are ignored.	R	0x0000 0000

Bits	Field Name	Description	Type	Reset
0	TAKEN	Lock State Read 0x0: Lock was previously Not Taken (free). The requester is granted the lock. Write 0x0: Set the lock to Not Taken (free). Read 0x1: Lock was previously Taken. The requester is not granted the lock and must retry. Write 0x1: No update to the lock value.	RW	0

Chapter 22

Timers

This chapter describes the timer modules for the device.

22.1 Timers Overview 5199
22.2 General-Purpose Timers 5200
22.3 32-kHz Synchronized Timer (COUNTER_32K) 5249
22.4 Watchdog Timer 5256

22.1 Timers Overview

The device includes several types of timers used by the system software, including 16 general-purpose (GP) timers, one watchdog timer, and a 32-kHz synchronized timer (COUNTER_32K). Figure 22-1 is a high-level block diagram of the device timers.

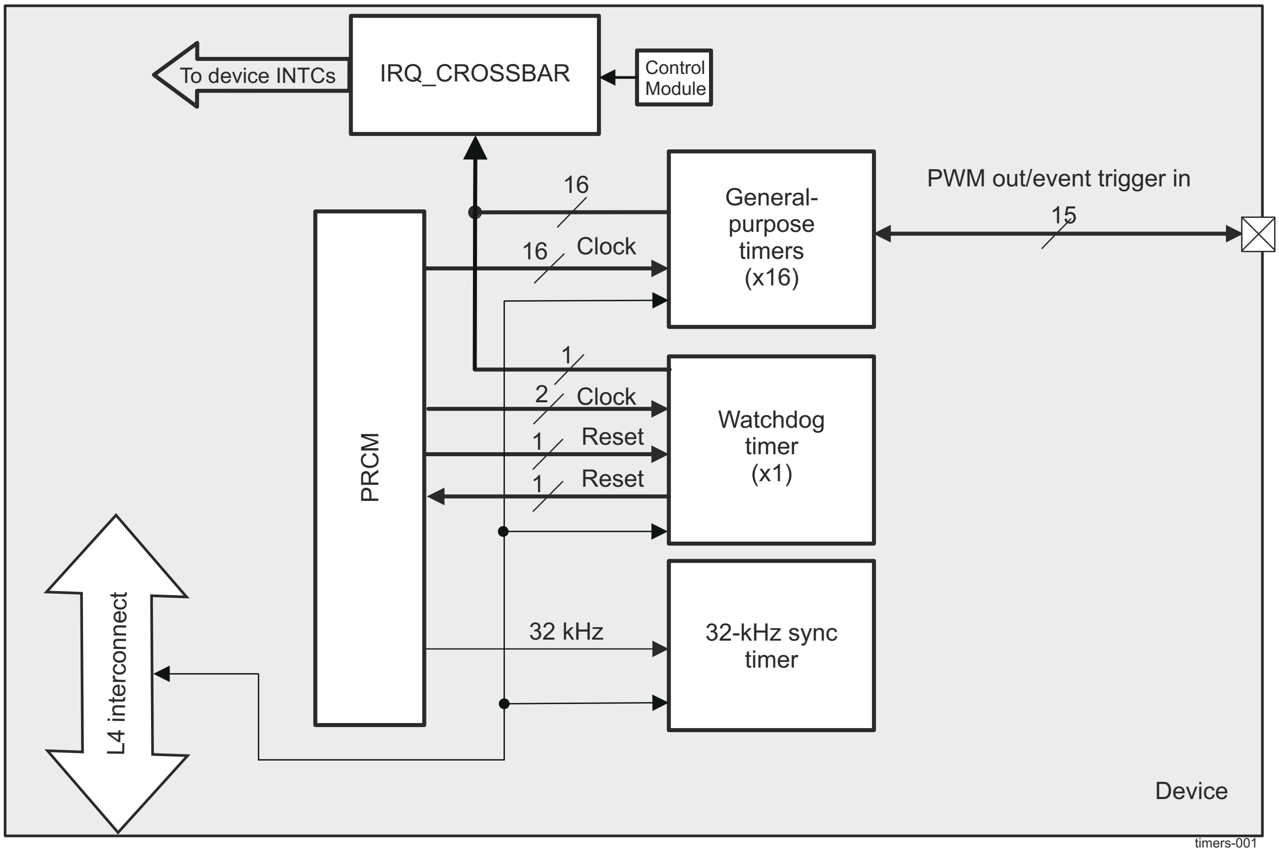


Figure 22-1. Timers Overview

The watchdog timer is clocked with 32-kHz clocks. The 32-kHz sync timer, which is reset only at power up, provides the operating system (OS) with a stable timing source that stores the relative time since the last power cycle of the product. The fifteen GP timers, which are useful as basic timers, are included to generate time-stamp-based interrupts to the system software or to use as a source of pulse-width modulation (PWM) signals.

22.2 General-Purpose Timers

22.2.1 General-Purpose Timers Overview

The device has 16 GP timers: TIMER1 through TIMER16.

- TIMER1(1ms tick): has its event capture pin tied to 32KHz clock and can be used to gauge the system clock input and detects its frequency among 19.2, 20, or 27 MHz. It includes specific functions to generate accurate tick interrupts to the operating system
- TIMER2 and TIMER10: (1ms tick timers): they include specific functions to generate accurate tick interrupts to the operating system

Each timer (except TIMER12) can be clocked from the system clock (19.2, 20, or 27 MHz) or the 32-kHz clock. The selection of clock source is made at the power, reset, and clock management (PRCM) module level. TIMER12 can be clocked only from the internal oscillator (on-die oscillator). For more information, see *PRM Clock Source*.

Each timer provides an interrupt via the device IRQ_CROSSBAR.

Each timer is connected to external pin by its PWM output or its event capture input pin (for external timer triggering). Figure 22-2 is an overview of the GP timers.

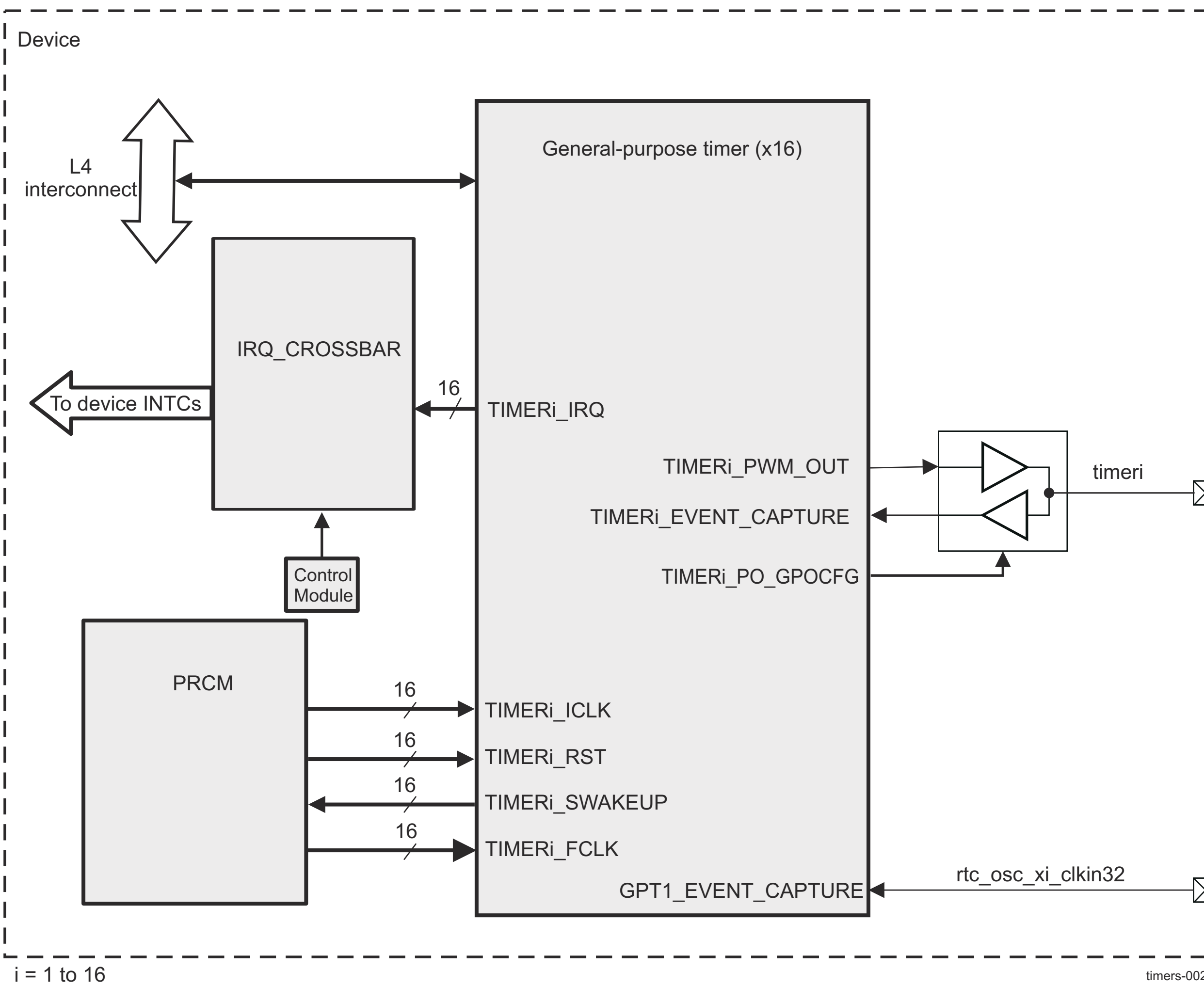


Figure 22-2. GP Timers Overview

22.2.1.1 GP Timer Features

The following are the main features of the GP timer controllers:

- Level 4 (L4) slave interface support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 10-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode supported
- Interrupts generated on overflow, compare, and capture
- Free-running 32-bit upward counter
- Compare and capture modes
- Autoreload mode
- Start/stop mode
- Programmable divider clock source (2^n, where n = [0:8])
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated GP output signal for using the TIMERi_GPO_CFG signal
- On-the-fly read/write register (while counting)
- 1-ms tick with 32.768-Hz functional clock generated (only TIMER1, TIMER2, and TIMER10)

22.2.2 GP Timer Environment

22.2.2.1 GP Timer External System Interface

Each timer can send or receive stimulus to/from the external (off-chip) system. In the device all timers are configured to output a PWM pulse or receive an external event signal used as a trigger to capture the current timer count. TIMER1 is also configured to receive an event trigger input (GPT1_EVENT_CAPTURE) tied to the internal 32-kHz clock. This event signal gauges the system clock input, detecting its frequency among 19.2, 20, or 27 MHz.

Figure 22-3 shows the external system interface for the GP timers, and Table 22-1 describes the GP timer inputs and outputs.

Note

Software control must ensure that MUX mode is configured to select the timeri (where i = 1 to 16) signal on only one pad. Other pads on which the same signal is multiplexed must be configured in safe mode or non-dmtimer mode to avoid two different pads driving the same signal.

For more information about the configuration of the timeri I/O pads, see *Pad Configuration Registers*.

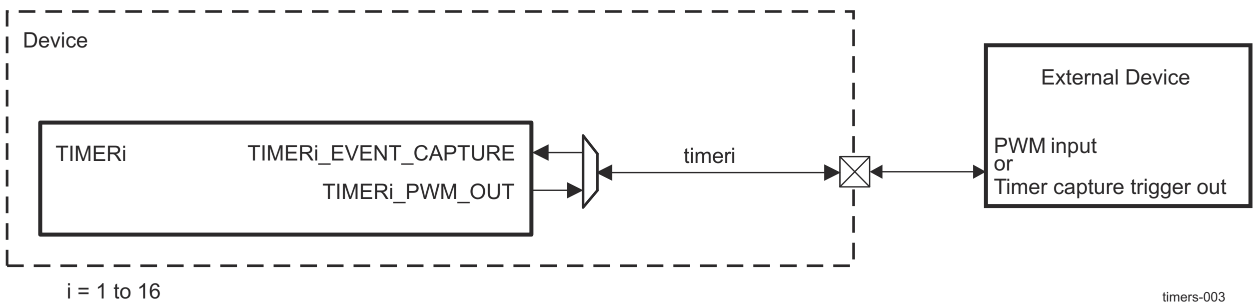


Figure 22-3. GP Timers External System Interface

Table 22-1. Input/Output Description

Pin Name	Type [(1)]	Reset Value	Signal Name	Description
timer1	I/O	0	TIMER1_PI_EVENT_CAPTURE TIMER1_PWM_OUT	TIMER1 trigger input/ PWM output
timer2	I/O	0	TIMER2_PI_EVENT_CAPTURE TIMER2_PWM_OUT	TIMER2 trigger input/ PWM output
timer3	I/O	0	TIMER3_PI_EVENT_CAPTURE TIMER3_PWM_OUT	TIMER3 trigger input/ PWM output
timer4	I/O	0	TIMER4_PI_EVENT_CAPTURE TIMER4_PWM_OUT	TIMER4 trigger input/ PWM output
timer5	I/O	0	TIMER5_PI_EVENT_CAPTURE TIMER5_PWM_OUT	TIMER5 trigger input/ PWM output
timer6	I/O	0	TIMER6_PI_EVENT_CAPTURE TIMER6_PWM_OUT	TIMER6 trigger input/ PWM output
timer7	I/O	0	TIMER7_PI_EVENT_CAPTURE TIMER7_PWM_OUT	TIMER7 trigger input/ PWM output
timer8	I/O	0	TIMER8_PI_EVENT_CAPTURE TIMER8_PWM_OUT	TIMER8 trigger input/ PWM output
timer9	I/O	0	TIMER9_PI_EVENT_CAPTURE TIMER9_PWM_OUT	TIMER9 trigger input/ PWM output
timer10	I/O	0	TIMER10_PI_EVENT_CAPTURE TIMER10_PWM_OUT	TIMER10 trigger input/ PWM output
timer11	I/O	0	TIMER11_PI_EVENT_CAPTURE TIMER11_PWM_OUT	TIMER11 trigger input/ PWM output

Table 22-1. Input/Output Description (continued)

Pin Name	Type [(1)]	Reset Value	Signal Name	Description
timer12	I/O	0	TIMER12_PI_EVENT_CAPTURE TIMER12_PWM_OUT	TIMER11 trigger input/ PWM output
timer13	I/O	0	TIMER13_PI_EVENT_CAPTURE TIMER13_PWM_OUT	TIMER13 trigger input/ PWM output
timer14	I/O	0	TIMER14_PI_EVENT_CAPTURE TIMER14_PWM_OUT	TIMER14 trigger input/ PWM output
timer15	I/O	0	TIMER15_PI_EVENT_CAPTURE TIMER15_PWM_OUT	TIMER15 trigger input/ PWM output
timer16	I/O	0	TIMER16_PI_EVENT_CAPTURE TIMER16_PWM_OUT	TIMER16 trigger input/ PWM output

(1) When configured for that function; I = Input, O = Output

Note

Each TIMERi_PO_GPOCFG signal is used to as an output enable to control the function of the TIMERi pin (where i = 1 to 16) as the PWM output (PO_GPOCFG =0) or capture input (PO_GPOCFG = 1).

22.2.3 GP Timer Integration

Figure 22-4 shows the integration of the GP timer in the device.

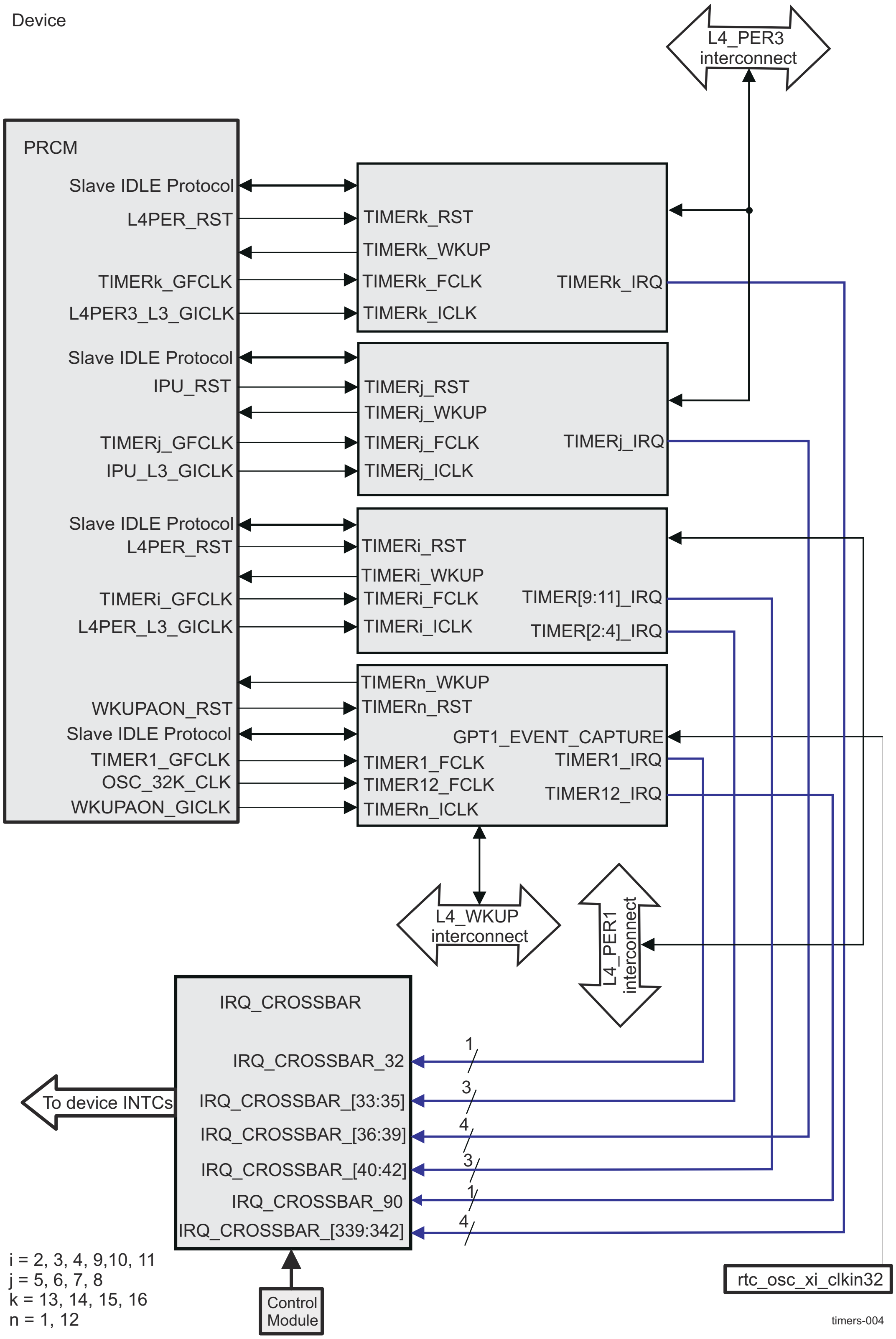


Figure 22-4. GP Timer Integration

Table 22-2 through Table 22-4 summarize the integration of the module in the device.

Table 22-2. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
TIMER1	PD_WKUPAON	Yes	L4_WKUP
TIMER2	PD_COREAON	Yes	L4_PER1
TIMER3	PD_COREAON	Yes	L4_PER1
TIMER4	PD_COREAON	Yes	L4_PER1
TIMER5	PD_COREAON	Yes	L4_PER3
TIMER6	PD_COREAON	Yes	L4_PER3
TIMER7	PD_COREAON	Yes	L4_PER3
TIMER8	PD_COREAON	Yes	L4_PER3
TIMER9	PD_COREAON	Yes	L4_PER1
TIMER10	PD_COREAON	Yes	L4_PER1
TIMER11	PD_COREAON	Yes	L4_PER1
TIMER12	PD_WKUPAON	Yes	L4_WKUP
TIMER13	PD_COREAON	Yes	L4_PER3
TIMER14	PD_COREAON	Yes	L4_PER3
TIMER15	PD_COREAON	Yes	L4_PER3
TIMER16	PD_COREAON	Yes	L4_PER3

Table 22-3. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
TIMER1	TIMER1_FCLK	TIMER1_GFCLK	PRCM	TIMER1 functional clock
	TIMER1_ICLK	WKUPAON_GICLK	PRCM	TIMER1 interface clock
TIMER2	TIMER2_FCLK	TIMER2_GFCLK	PRCM	TIMER2 functional clock
	TIMER2_ICLK	L4PER_L3_GICLK	PRCM	TIMER2 interface clock
TIMER3	TIMER3_FCLK	TIMER3_GFCLK	PRCM	TIMER3 functional clock
	TIMER3_ICLK	L4PER_L3_GICLK	PRCM	TIMER3 interface clock
TIMER4	TIMER4_FCLK	TIMER4_GFCLK	PRCM	TIMER4 functional clock
	TIMER4_ICLK	L4PER_L3_GICLK	PRCM	TIMER4 interface clock
TIMER5	TIMER5_FCLK	TIMER5_GFCLK	PRCM	TIMER5 functional clock
	TIMER5_ICLK	IPU_L3_GICLK	PRCM	TIMER5 interface clock
TIMER6	TIMER6_FCLK	TIMER6_GFCLK	PRCM	TIMER6 functional clock
	TIMER6_ICLK	IPU_L3_GICLK	PRCM	TIMER6 interface clock
TIMER7	TIMER7_FCLK	TIMER7_GFCLK	PRCM	TIMER7 functional clock
	TIMER7_ICLK	IPU_L3_GICLK	PRCM	TIMER7 interface clock
TIMER8	TIMER8_FCLK	TIMER8_GFCLK	PRCM	TIMER8 functional clock
	TIMER8_ICLK	IPU_L3_GICLK	PRCM	TIMER8 interface clock
TIMER9	TIMER9_FCLK	TIMER9_GFCLK	PRCM	TIMER9 functional clock
	TIMER9_ICLK	L4PER_L3_GICLK	PRCM	TIMER9 interface clock
TIMER10	TIMER10_FCLK	TIMER10_GFCLK	PRCM	TIMER10 functional clock
	TIMER10_ICLK	L4PER_L3_GICLK	PRCM	TIMER10 interface clock
TIMER11	TIMER11_FCLK	TIMER11_GFCLK	PRCM	TIMER11 functional clock
	TIMER11_ICLK	L4PER_L3_GICLK	PRCM	TIMER11 interface clock
TIMER12	TIMER12_FCLK	OSC_32K_CLK[1]	PRCM	TIMER12 functional clock
	TIMER12_ICLK	WKUPAON_GICLK	PRCM	TIMER12 interface clock

Table 22-3. Clocks and Resets (continued)

TIMER13	TIMER13_FCLK	TIMER13_GFCLK	PRCM	TIMER13 functional clock
	TIMER13_ICLK	L4PER3_L3_GICLK	PRCM	TIMER13 interface clock
TIMER14	TIMER14_FCLK	TIMER14_GFCLK	PRCM	TIMER14 functional clock
	TIMER14_ICLK	L4PER3_L3_GICLK	PRCM	TIMER14 interface clock
TIMER15	TIMER15_FCLK	TIMER15_GFCLK	PRCM	TIMER15 functional clock
	TIMER15_ICLK	L4PER3_L3_GICLK	PRCM	TIMER15 interface clock
TIMER16	TIMER16_FCLK	TIMER16_GFCLK	PRCM	TIMER16 functional clock
	TIMER16_ICLK	L4PER3_L3_GICLK	PRCM	TIMER16 interface clock
		Resets		
TIMER1	TIMER1_RST	WKUPAON_RST	PRM	Reset to TIMER1
TIMER2	TIMER2_RST	L4PER_RST	PRM	Reset to TIMER2
TIMER3	TIMER3_RST	L4PER_RST	PRM	Reset to TIMER3
TIMER4	TIMER4_RST	L4PER_RST	PRM	Reset to TIMER4
TIMER5	TIMER5_RST	IPU_RST	PRM	Reset to TIMER5
TIMER6	TIMER6_RST	IPU_RST	PRM	Reset to TIMER6
TIMER7	TIMER7_RST	IPU_RST	PRM	Reset to TIMER7
TIMER8	TIMER8_RST	IPU_RST	PRM	Reset to TIMER8
TIMER9	TIMER9_RST	L4PER_RST	PRM	Reset to TIMER9
TIMER10	TIMER10_RST	L4PER_RST	PRM	Reset to TIMER10
TIMER11	TIMER11_RST	L4PER_RST	PRM	Reset to TIMER11
TIMER12	TIMER12_RST	WKUPAON_RST	PRM	Reset to TIMER12
TIMER13	TIMER13_RST	L4PER_RST	PRM	Reset to TIMER13
TIMER14	TIMER14_RST	L4PER_RST	PRM	Reset to TIMER14
TIMER15	TIMER15_RST	L4PER_RST	PRM	Reset to TIMER15
TIMER16	TIMER16_RST	L4PER_RST	PRM	Reset to TIMER16

(1) The OSC_32K_CLK clock, provided by the On-die 32K RC Osc is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

Table 22-4. GP Timers Hardware Requests

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
		Interrupt Requests		
TIMER1	TIMER1_IRQ	IRQ_CROSSBAR_32	MPU_IRQ_37	TIMER1 interrupt
			DSP1_IRQ_63	
			DSP2_IRQ_63	
TIMER2	TIMER2_IRQ	IRQ_CROSSBAR_33	MPU_IRQ_38	TIMER2 interrupt
			DSP1_IRQ_64	
			DSP2_IRQ_64	
TIMER3	TIMER3_IRQ	IRQ_CROSSBAR_34	MPU_IRQ_39	TIMER3 interrupt
			DSP1_IRQ_65	
			DSP2_IRQ_65	
			IPU1_IRQ_53	
			IPU2_IRQ_53	
TIMER4	TIMER4_IRQ	IRQ_CROSSBAR_35	MPU_IRQ_40	TIMER4 interrupt
			DSP1_IRQ_66	
			DSP2_IRQ_66	
			IPU1_IRQ_54	
			IPU2_IRQ_54	

Table 22-4. GP Timers Hardware Requests (continued)

TIMER5	TIMER5_IRQ	IRQ_CROSSBAR_36	MPU_IRQ_41 DSP1_IRQ_67 DSP2_IRQ_67	TIMER5 interrupt
TIMER6	TIMER6_IRQ	IRQ_CROSSBAR_37	MPU_IRQ_42 DSP1_IRQ_68 DSP2_IRQ_68	TIMER6 interrupt
TIMER7	TIMER7_IRQ	IRQ_CROSSBAR_38	MPU_IRQ_43 DSP1_IRQ_69 DSP2_IRQ_69	TIMER7 interrupt
TIMER8	TIMER8_IRQ	IRQ_CROSSBAR_39	MPU_IRQ_44 DSP1_IRQ_70 DSP2_IRQ_70	TIMER8 interrupt
TIMER9	TIMER9_IRQ	IRQ_CROSSBAR_40	MPU_IRQ_45 DSP1_IRQ_71 DSP2_IRQ_71 IPU1_IRQ_55 IPU2_IRQ_55	TIMER9 interrupt
TIMER10	TIMER10_IRQ	IRQ_CROSSBAR_41	MPU_IRQ_46 DSP1_IRQ_72 DSP2_IRQ_72	TIMER10 interrupt
TIMER11	TIMER11_IRQ	IRQ_CROSSBAR_42	MPU_IRQ_47 DSP1_IRQ_73 DSP2_IRQ_73 IPU1_IRQ_56 IPU2_IRQ_56	TIMER11 interrupt
TIMER12	TIMER12_IRQ	IRQ_CROSSBAR_90	MPU_IRQ_95	TIMER12 interrupt
TIMER13	TIMER13_IRQ	IRQ_CROSSBAR_339	-	TIMER13 interrupt. This IRQ source signal is not mapped by default to any device INTC
TIMER14	TIMER14_IRQ	IRQ_CROSSBAR_340	-	TIMER14 interrupt. This IRQ source signal is not mapped by default to any device INTC
TIMER15	TIMER15_IRQ	IRQ_CROSSBAR_341	-	TIMER15 interrupt. This IRQ source signal is not mapped by default to any device INTC
TIMER16	TIMER16_IRQ	IRQ_CROSSBAR_342	-	TIMER16 interrupt. This IRQ source signal is not mapped by default to any device INTC
No DMA Requests				

Note

The “**Default Mapping**” column in Table 22-4 *GP Timers Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For the description of the interrupt source, see Section 22.2.4.5, *GP Timer Interrupt*.

22.2.4 GP Timer Functional Description

Each GP timer contains a free-running upward counter with autoreload capability on overflow. The timer counter can be read and written on-the-fly (while counting). Each GP timer includes compare logic to allow an interrupt event on a programmable counter matching value. A dedicated output signal can be pulsed or toggled on either an overflow or a match event. This offers time-stamp trigger signaling or PWM signal sources. A dedicated input signal can be used to trigger an automatic timer counter capture or an interrupt event on a programmable input signal transition. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged into one module interrupt line and one wake-up line.

Each internal interrupt source can be independently enabled and disabled by a dedicated bit in the IRQSTATUS_SET and IRQSTATUS_CLR register for the interrupt features, and a dedicated bit of the IRQWAKEEN register for the wake-up of TIMER1, TIMER2, and TIMER10. In addition, these timers have a mechanism implemented to generate an accurate tick interrupt.

For all other internal interrupt source can be independently enabled and disabled through the IRQENABLE_SET and IRQENABLE_CLR registers.

For each GP timer implemented in the device, there are two possible clock sources:

- 32-kHz clock
- System clock

Selection of the input clock source is done in the registers in the PRCM configuration (see Section 22.2.1, *GP Timer Overview*).

Each GP timer supports three functional modes:

- Timer mode
- Capture mode
- Compare mode

The capture and compare modes are disabled by default after core reset.

22.2.4.1 GP Timer Block Diagram

Figure 22-5 is a block diagram of the common GP timers, and Figure 22-6 is a block diagram of the GP timers with a 1-ms tick generation module.

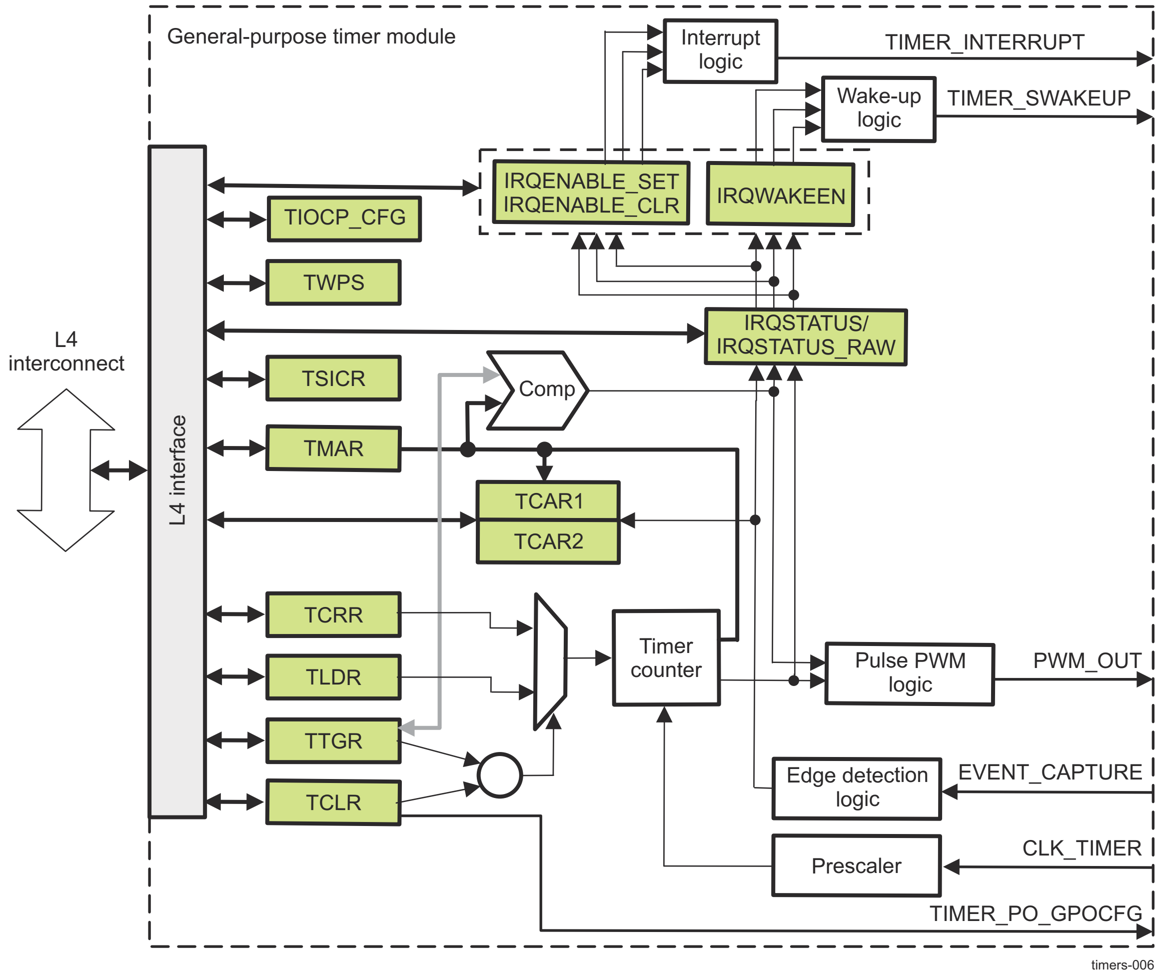


Figure 22-5. Block Diagram of TIMER3 Through TIMER9 and TIMER11 Through TIMER16

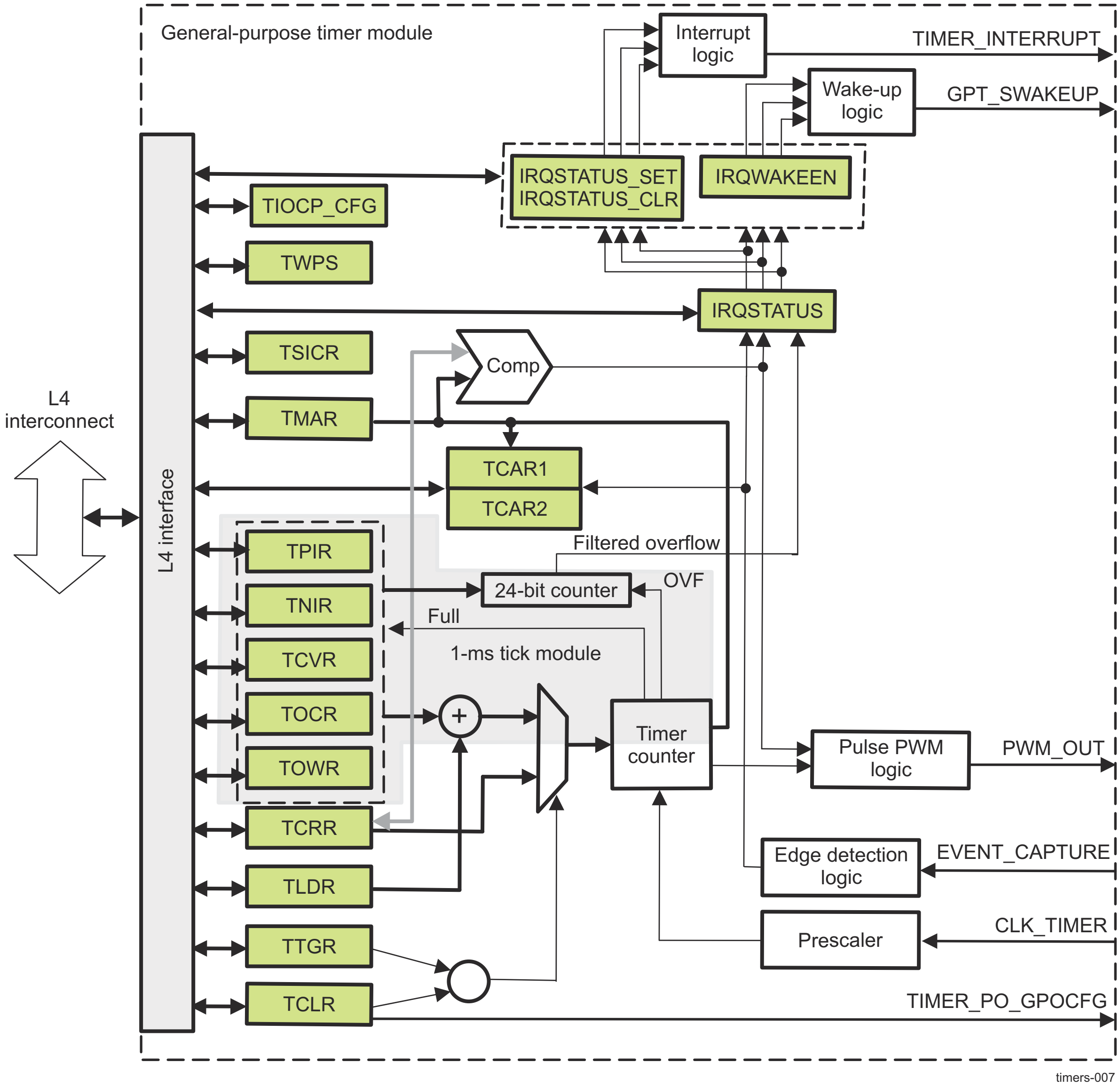


Figure 22-6. Block Diagram of TIMER1, TIMER2 and TIMER10

22.2.4.2 TIMER1, TIMER2 and TIMER10 Power Management

At the PRCM module level, when all conditions to shut off the functional or interface output clocks in the PRCM module are met (see *Clock Domain-Level Clock Management*), the PRCM module automatically launches a hardware handshake protocol to ensure the GP timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the GP timer.

Although this handshake is a hardware function and is out of software control, the way the GP timer acknowledges the PRCM IDLE request is configurable through the TIOCP_CFG[3:2] IDLEMODE bit field.

Table 22-5 lists the IDLEMODE settings and the related acknowledgment modes.

Table 22-5. IDLEMODE Settings

IDLEMODE Value	Selected Mode	Description
00	Force-idle	The GP timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent the loss of data when the clock is switched off.
01	No-idle	The GP timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. It is not efficient from a power-saving perspective, because it does not allow the PRCM output clock to be shut off, and thus the power domain to be set to a lower power state.
10	Smart-idle	The GP timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach to efficient system power management.
11	Smart-idleWakeup	The module behaves like in Smart-idle mode, with the exception, that it can issue a wake-up request in sleep mode, if the functional clock is not cut off.

22.2.4.2.1 Wake-Up Capability

If the TIOCP_CFG[3:2] IDLEMODE bit field sets the smart-idle mode or smart-idle with wakeup mode, the timer evaluates its internal capability to have the interface clock switched off. When there is no further internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters sleep mode, ready to issue a wake-up request if is configured in smart-idle with wakeup mode.

Figure 22-7 shows the wake-up request generation. For more information about the GP timer clock control, *Clock Management Functional Description*, in *Power, Reset, and Clock Management*.

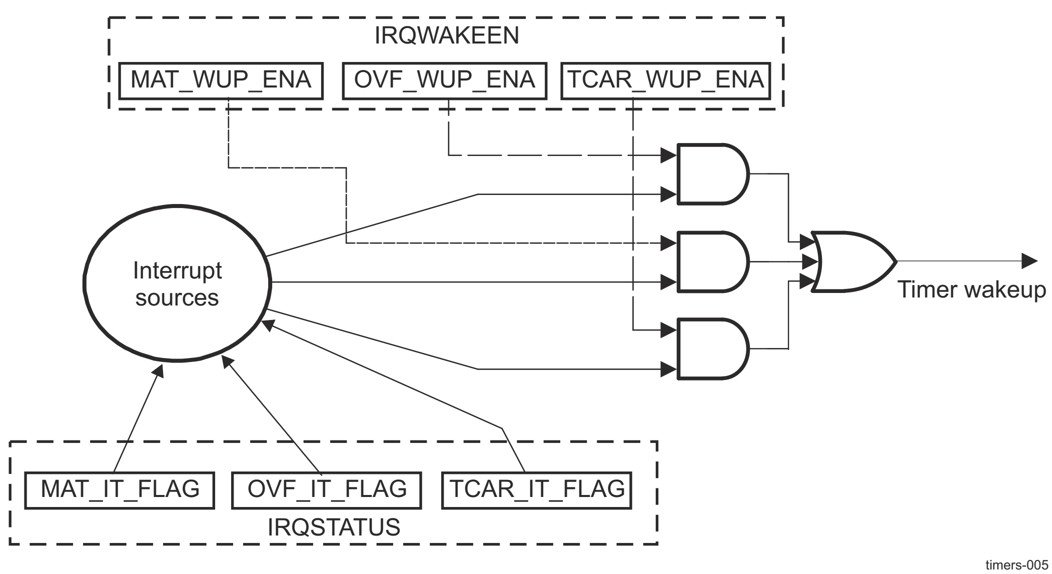


Figure 22-7. Wake-Up Request Generation

For TIMER1, TIMER2 and TIMER10, the timer wake-up-enable register allows masking of the expected source of the wake-up event that generates a wake-up request. The register is synchronously programmed with the interface clock before the PRCM module sends an idle mode request. The expected source of the wake-up event is an overflow (TCRR), a timer match (the compare result of TCRR and TMAR matches the counter value), and a timer capture (detection of an external pulse transition of the correct polarity on the TIMER_EVENT_CAPTURE).

When the wake-up event is issued, the associated interrupt status bit is set in the timer status register (IRQSTATUS). The pending wake-up event is reset when the set status bit is overwritten with 1.

Note

The status bit must be reset to re-enter idle mode.

22.2.4.3 Power Management of Other GP Timers

At the PRCM module level, when all conditions to shut off the functional or interface output clocks of the PRCM module are met (see *Clock Domain-Level Clock Management*), the PRCM module automatically launches a hardware handshake protocol to ensure the GP timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the GP timer.

Although this handshake is a hardware function and is out of software control, the way the GP timer acknowledges the PRCM IDLE request is configurable through the TIOCP_CFG[3:2] IDLEMODE bit field.

Table 22-6 lists the IDLEMODE settings and the related acknowledgment modes.

Table 22-6. IDLEMODE Settings

IDLEMODE Value	Selected Mode	Description
00	Force-idle	The GP timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent the loss of data when the clock is switched off.
01	No-idle	The GP timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. It is not efficient from a power-saving perspective, however, because it does not allow the PRCM output clock to be shut off and thus the power domain to be set to a lower power state.
10	Smart-idle	The GP timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach to efficient system power management.
11	Smart-idleWakeup	The module behaves like in Smart-idle mode, with the exception, that it can issue a wake-up request in sleep mode, if the functional clock is not cut off.

22.2.4.3.1 Wake-Up Capability

If the TIOCP_CFG[3:2] IDLEMODE bit field sets the smart-idle mode or smart-idle with wakeup mode, the timer evaluates its internal capability to have the interface clock switched off. When there is no further internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters sleep mode, ready to issue a wake-up request if is configured in smart-idle with wakeup mode. This wake-up request is sent only if the IRQWAKEEN[2:0] bit field enables the timer wake-up capability.

Figure 22-8 shows the wake-up request generation. For more information about the GP timer clock control, see *Clock Management Functional Description*, in *Power, Reset, and Clock Management*.

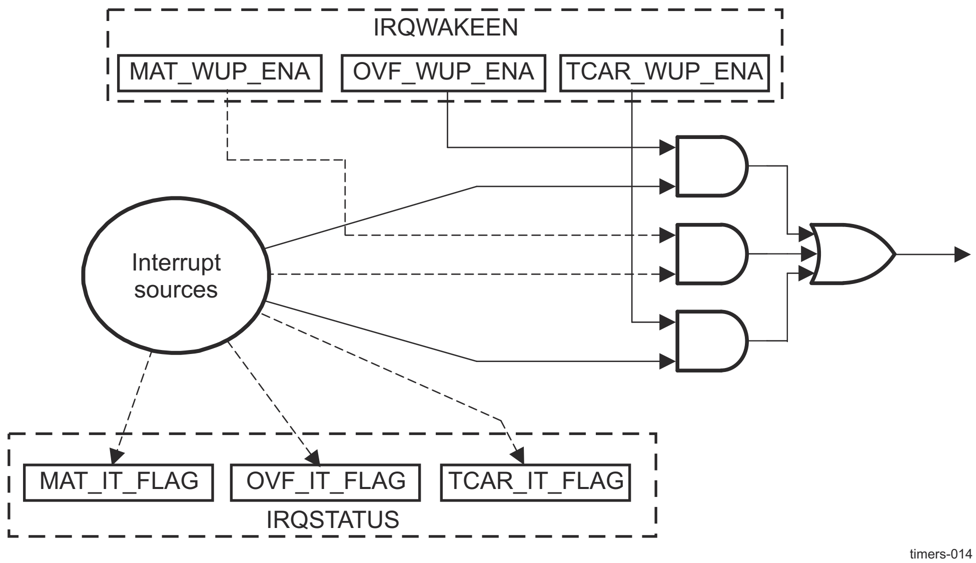


Figure 22-8. Wake-Up Request Generation

When the wake-up event is issued, the associated interrupt status bit is set in the timer status register (IRQSTATUS). The pending wake-up event is reset when the set status bit is overwritten with 1.

Note

The status bit must be reset to re-enter idle mode.

22.2.4.4 Software Reset

Two bits can generate a software reset of the GP timer:

- TIOCP_CFG[0] SOFTRESET
- TSICR[1] SFT

For both bits, all read accesses return 0.

The TIOCP_CFG[0] SOFTRESET bit allows resetting of the functional and interface domains. The TSICR[1] SFT bit allows resetting the functional part of the GP timer.

Before accessing or using the GP timer, the local host must ensure that both internal resets are released by reading the TIOCP_CFG[0] SOFTRESET bit. This bit monitors the internal reset status.

22.2.4.5 GP Timer Interrupts

The timer can issue an overflow interrupt, a timer match interrupt, and a timer capture interrupt. Each internal interrupt source can be independently enabled and disabled in the interrupt-enable register (IRQSTATUS_SET for TIMER1/2/10 and IRQENABLE_SET for other timers) and disabled in the interrupt-disable register (IRQSTATUS_CLR for TIMER1/2/10 and IRQENABLE_CLR for other timers). When the interrupt event is issued, the associated interrupt status bit is set in the timer status register (IRQSTATUS).

22.2.4.6 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the timer control register (the TCLR[0] ST bit). The timer counter register (TCRR) can be loaded when stopped or on-the-fly (while counting). TCRR can be loaded directly by a TCRR write access with a new timer value. TCRR can also be loaded with the value held in the timer load register (TLDR) by a trigger register (TTGR) write access. The loading of TCRR is done regardless of the written value of TTGR. The value of TCRR can be read when stopped or captured on-the-fly by a TCRR read access. The timer is stopped and the counter value is set to 0 when the module reset is asserted. The timer is maintained at stop after the reset is released.

In one-shot mode (the TCLR[1] AR bit is set to 0), the counter is stopped after counting overflow occurs (the counter value remains at 0).

When the autoreload mode is enabled (the TCLR[1] AR bit is set to 1), TCRR is reloaded with the value of TLDR after a counting overflow occurs.

CAUTION

Do not put the overflow value (0xFFFF FFFF) in the TLDR register because it can lead to undesirable results.

An interrupt can be issued on overflow if the overflow interrupt-enable bit is set in the timer interrupt-enable register (the IRQSTATUS_SET[1] OVF_EN_FLAG bit is set to 1 for TIMER1/2/10 and the IRQENABLE_SET[1]OVF_EN_FLAG bit is set to 1 for other timers). A dedicated output pin (timer PWM) can be programmed in the TCLR[12] PT bit through the TCLR[11:10] (PT and TRG bits) to generate one positive pulse (prescaler duration) or to invert the current value (toggle mode) when an overflow occurs. The TCLR[12] PT bit selects pulse/toggle modulation (the TCLR[11:10] TRG bit field selects trigger mode).

Figure 22-9 shows the TCRR timing value.

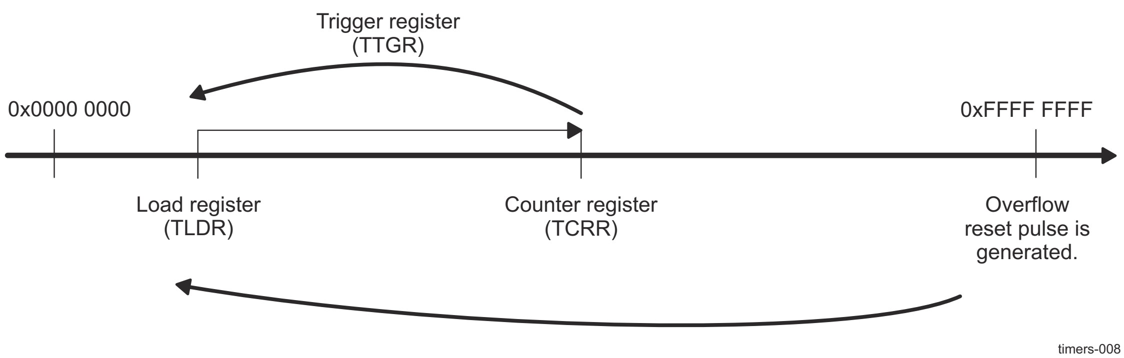


Figure 22-9. TCRR Timing Value

22.2.4.6.1 1-ms Tick Generation (Only TIMER1, TIMER2 and TIMER10)

The interrupt period is not exactly 1 ms, because the timer input clock is 32.768 Hz. If the clock counts up to 32, it obtains a 0.977-ms period; if it counts up to 33, it obtains a 1.007-ms period. For large granularity, the error is cumulative and can generate important deviations from the standard value.

To minimize the error between a true 1-ms tick and the tick generated by the 32.768-Hz timer, the sequencing of periods less than 1 ms and periods greater than 1 ms must be shuffled. An additional 1-ms block is used to correct this error. See Figure 22-10.

In this implementation, the increment sequencing is automatically managed by the timer to minimize the error. The user must define only the value of the timer positive increment register (the TPIR[31:0] POSITIVE_INC_VALUE bit field) and the timer negative increment register (the TNIR[31:0] NEGATIVE_INC_VALUE bit field). An automatic adaptation mechanism is used to simplify the programming model.

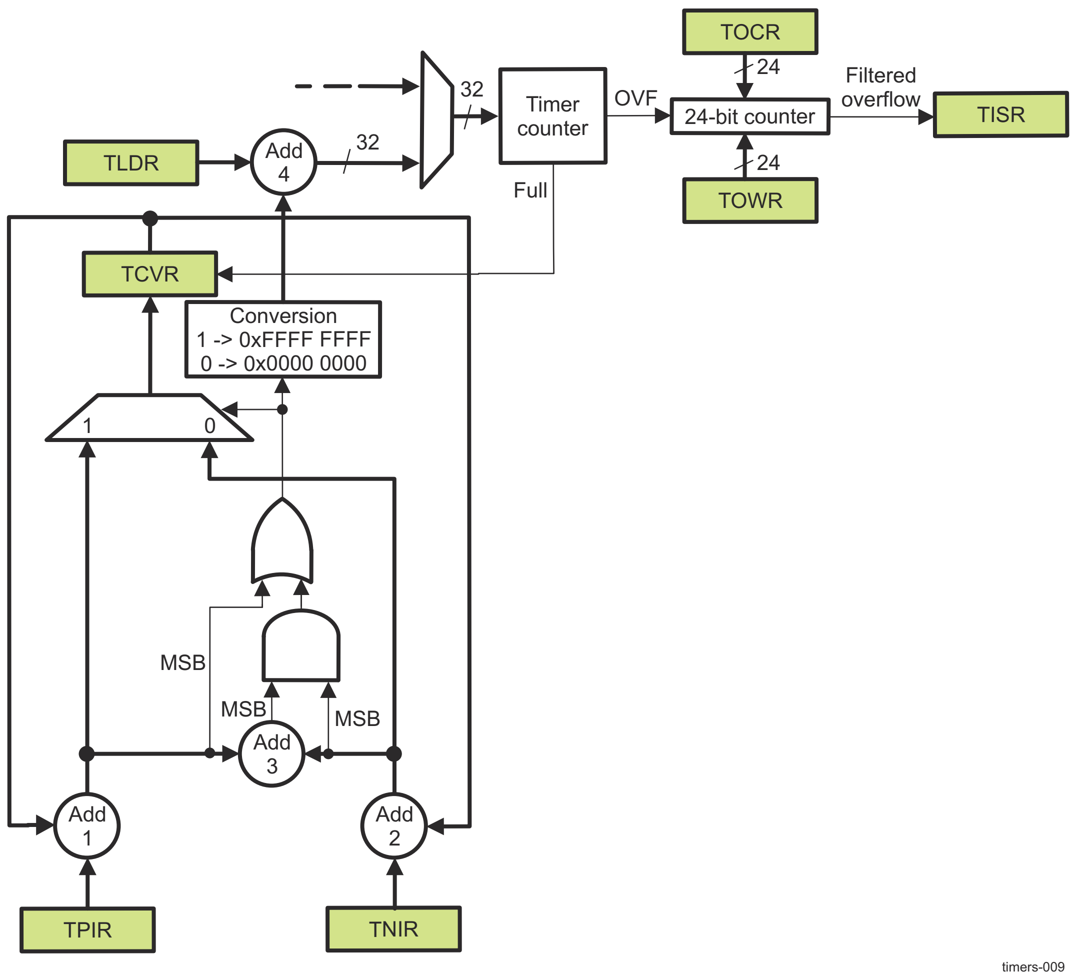


Figure 22-10. Block Diagram of the 1-ms Tick Module

The TPIR, TNIR, and TCVR registers and adders Add1, Add2, and Add3 are used to define whether the next value loaded in the timer counter register (the TCRR[31:0] TIMER_COUNTER bit field) is the value of the TLDR[31:0] LOAD_VALUE bit field (period less than 1 ms) or the value of TLDR[31:0] LOAD_VALUE –1 (period greater than 1 ms).

Table 22-7 lists the value loaded in the TCRR according to the sign of the result of Add1, Add2, and Add3.

MSB = 0: Positive value; MSB = 1: Negative value

Table 22-7. Value Loaded in TCRR to Generate 1-ms Tick

Add1 MSB	Add2 MSB	Add3 MSB	Value of TCRR Register
0	0	0	TLDR[31:0] LOAD_VALUE bit field
0	0	1	TLDR[31:0] LOAD_VALUE bit field
0	1	0	TLDR[31:0] LOAD_VALUE bit field
0	1	1	TLDR[31:0] LOAD_VALUE –1
1	0	0	N/A
1	0	1	N/A
1	1	0	TLDR[31:0] LOAD_VALUE –1
1	1	1	TLDR[31:0] LOAD_VALUE –1

The values of the TPIR and TNIR registers are calculated using the following formulas:

- Positive increment value = ((INTEGER[F_{clk} × T_{tick}] + 1) × 1e6) – (F_{clk} × T_{tick} × 1e6)
- Negative increment value = (INTEGER[F_{clk} × T_{tick}] × 1e6) – (F_{clk} × T_{tick} × 1e6)

Note

F_{clk} clock frequency (kHz)

T_{tick} tick period (ms)

The timer overflow counter register (TOCR) and the timer overflow wrapping register (TOWR) are used to filter interrupts. When the timer overflows, it increments the 24-bit TOCR. When the values in the 24-bit TOCR match the values in the 24-bit TOWR and the timer overflow is asserted, the TOCR is reset and an interrupt is generated to the IRQSTATUS register.

Note

TOWR has to be set to requested value. For example, if no interrupt needs to be masked TOWR must be set ot 0, if one interrupt needs to TOWR must be set to 1, if two interrupt need to be masked TOWR must be set to 2 and so on.

It is important to have in mind that the case when FFFFFF interrupt need to be masked is not possible.

With the conversion block in reset state (the positive increment register, negative increment register, and counter value register are zeroed), the programming model and the behavior of TIMER1, TIMER2, and TIMER10 remain unchanged.

For 1-ms tick with a 32.768-Hz clock:

- TPIR[31:0] POSITIVE_INC_VALUE = 232,000
- TNIR[31:0] NEGATIVE_INC_VALUE = –768,000
- TLDR[31:0] LOAD_VALUE = 0xFFFF FFE0

Note

Any value of the tick period can be generated with the appropriate value of the TPIR, TNIR, and TLDR.

By default, the TPIR, TNIR, TCVR, TOCR, and TOWR and the associated logic are in reset mode (all 0s) and have no effect on the programming model.

22.2.4.7 Capture Mode Functionality

When a transition is detected on the module input pin (EVENT_CAPTURE), the timer value in the TCRR can be captured and saved in the TCAR1 or TCAR2 register function of the mode selected in the TCLR[13] CAPT_MODE bit. The edge detection circuitry monitors transitions on the input pin (EVENT_CAPTURE).

The rising edge, falling edge, or both, can be selected in the TCLR[9:8] TCM bit field to trigger the timer counter capture. The module sets the IRQSTATUS[2] TCAR_IT_FLAG bit when an active edge is detected, and at the same time, the counter value TCRR is stored in timer capture register TCAR1 or TCAR2, as follows:

- If the TCLR[13] CAPT_MODE bit is 0, on the first enabled capture event, the value of the counter register is saved in the TCAR1 register, and the next events are ignored (no update on the TCAR1 register and no interrupt triggering) until the detection logic is reset or the IRQSTATUS[2] TCAR_IT_FLAG is cleared by writing 1 to it.
- If the TCLR[13] CAPT_MODE bit is 1, on the first enabled capture event, the value of the counter register is saved in the TCAR1 register, and on the second enabled capture event, the value of the counter register is saved in the TCAR2 register. If a capture interrupt is enabled, the interrupt triggers on the second event capture. All other events are ignored (no update on TCAR1/TCAR2 and no interrupt triggering) until the

detection logic is reset or the IRQSTATUS[2] TCAR_IT_FLAG bit is cleared by writing 1 to it. This mechanism is useful for period calculation of a clock, if that clock is connected to the EVENT_CAPTURE input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served. The IRQSTATUS[2] TCAR_IT_FLAG bit is cleared by writing 1 to it or when the edge detection mode bits (the TCLR[9:8] TCM bit field) are changed from no-capture mode detection to any other mode. The timer functional clock (input to prescaler) is used to sample the input pin (EVENT_CAPTURE). A negative or positive pulse input can be detected when the pulse time is greater than the functional clock period. An interrupt is issued on edge detection if the capture interrupt-enable bit is set in the IRQSTATUS_SET[2] TCAR_EN_FLAG bit (for TIMER1/2/10) or in the IRQENABLE_SET[2] TCAR_EN_FLAG bit (for other timers). See the examples in Figure 22-11 and Figure 22-12.

In Figure 22-11, the value of the TCLR[9:8] TCM bit field is 0b01, and the TCLR[13] CAPT_MODE bit is 0. Only the rising edge of EVENT_CAPTURE triggers a capture in the TCAR1 and TCAR2 registers, and only the TCAR1 register updates.

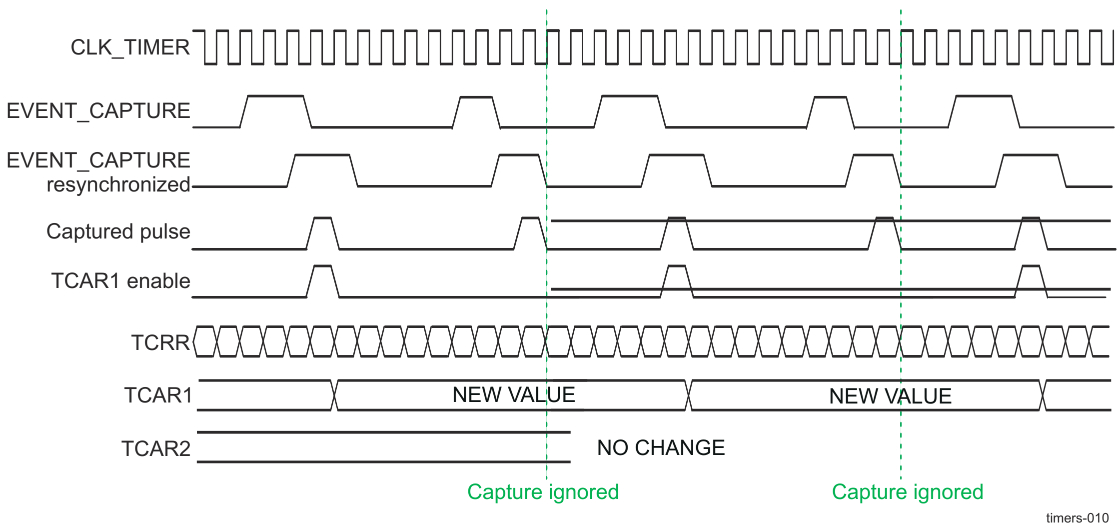


Figure 22-11. Capture Wave Example for TCLR[13] CAPT_MODE = 0

In Figure 22-12, the value of the TCLR[9:8] TCM bit field is 0b01, and the TCLR[13] CAPT_MODE bit is 1. Only the rising edge of EVENT_CAPTURE triggers a capture in the TCAR1 register on the first enabled event, and the TCAR2 register updates on the second enabled event.

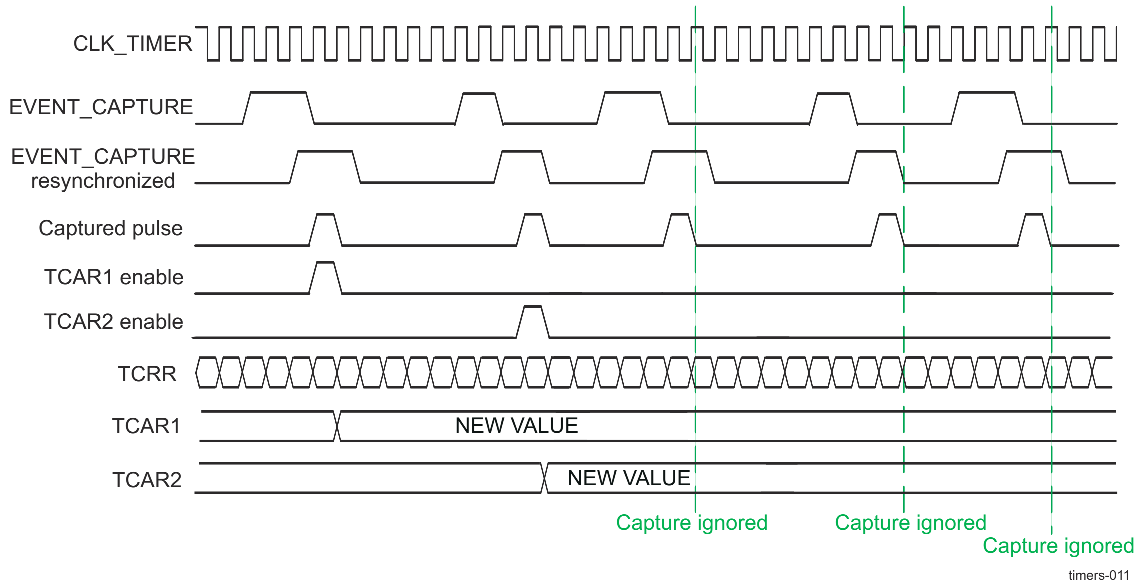


Figure 22-12. Capture Wave Example for TCLR[13] CAPT_MODE = 1

22.2.4.8 Compare Mode Functionality

When the compare-enable register TCLR[6] CE bit is set to 1, the timer value (the TCRR[31:0] TIMER_COUNTER bit field) is continuously compared to the value held in the timer match register (TMAR). The value of the TMAR[31:0] COMPARE_VALUE bit field can be loaded at any time (timer counting or stopped). When the TCRR and the TMAR values match, an interrupt is issued, if the IRQSTATUS_SET[0] MAT_EN_FLAG bit (for TIMER1, TIMER2, and TIMER10), or the IRQENABLE_SET[0] MAT_EN_FLAG bit (for other timers) is set.

To prevent any unwanted interrupts due to reset value matching effect, write a compare value to the TMAR before setting the TCLR[6] CE bit.

The dedicated output pin (timer PWM) can be programmed in the TCLR[12] PT bit through the TCLR[11:10] TRG bit field to generate one positive pulse (timer clock duration) or to invert the current value (toggle mode) when an overflow or a match occurs.

22.2.4.9 Prescaler Functionality

A prescaler can be used to divide the timer counter input clock frequency. The prescaler is enabled when the TCLR[5] PRE bit is set. The TCLR[4:2] PTV bit field sets the 2^n division ratio (prescaler value is $2^{(PTV + 1)}$. The prescaler counter is reset when the timer counter is stopped or reloaded on-the-fly.

Table 22-8 lists the prescaler/timer reload values versus contexts.

Table 22-8. Prescaler/Timer Reload Values Versus Contexts

Context	Prescaler	Timer Counter
Overflow (when autoreload is on)	Reset	TLDR[31:0]
TCRR write	Reset	TCRR[31:0]
TTGR write	Reset	TLDR[31:0]
Stop	Reset	Frozen

22.2.4.10 Pulse-Width Modulation

The timer can be configured to provide a programmable PWM output. The timer PWM output pin can be configured to toggle on an event. The TCLR[11:10] TRG bit field determines on which register value the PWM pin toggles. Either overflow or both overflow and match can be selected to toggle the timer PWM pin when a compare condition occurs.

Note

In toggle mode, when TCLR[11:10] TRG = 0x2 (overflow and match), the first event that toggles the PWM line is an overflow event. If a match event occurs first, it does not toggle the PWM line (see Figure 22-14).

The TCLR[7] SCPWM bit can be programmed to set or clear the timer PWM output signal only while the counter is stopped or the trigger is off. This allows setting the output pin to a known state before modulation starts. Modulation synchronously stops when the TCLR[11:10] TRG bit field is cleared and overflow occurs. This allows fixing a deterministic state of the output pin when modulation stops.

In Figure 22-13, the internal overflow pulse is set each time the (0xFFFF FFFF – TLDR[31:0] LOAD_VALUE + 1) value is reached, and the internal match pulse is set when the counter reaches the value of TMAR. Depending on the value of the TCLR[12] PT bit and TCLR[11:10] TRG bit field, the timer provides pulse or PWM event on the output pin (timer PWM).

The TLDR and TMAR must keep values below the overflow value (0xFFFF FFFF) by at least two units. If the PWM trigger events are both overflow and match, the difference between the values kept in the TMAR and the value in the TLDR must be at least two units. When match event is used, the compare mode TCLR[6] CE bit must be set.

In Figure 22-13, the TCLR[7] SCPWM bit is set to 0. In Figure 22-14, the TCLR[7] SCPWM bit is set to 1. To obtain the desired wave form, start the counter at 0xFFFF FFFE value (to ensure an overflow first) or adjust the line polarity (TCLR[7] SCPWM bit).

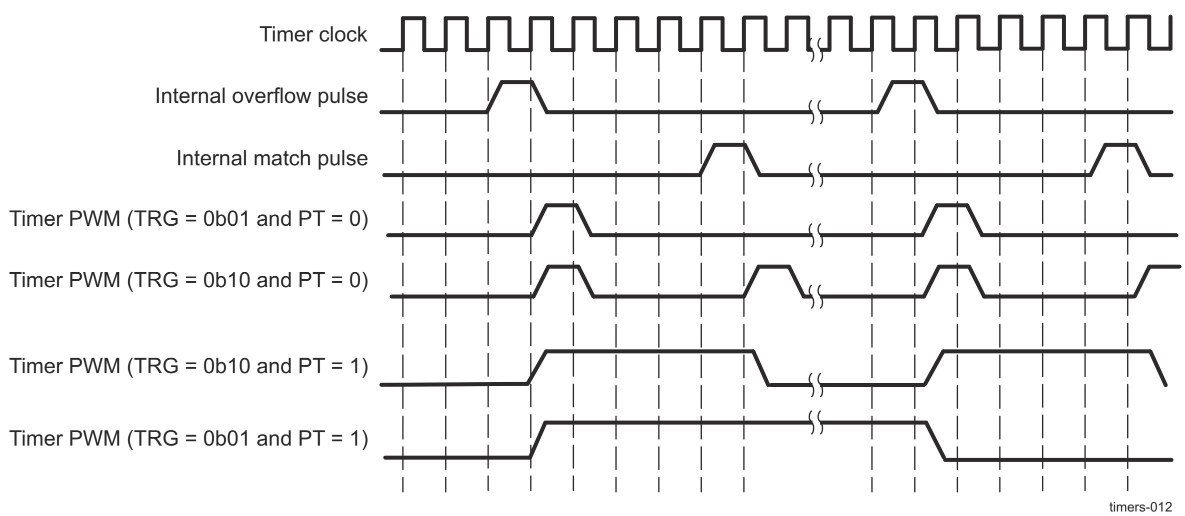


Figure 22-13. Timing Diagram of PWM With TCLR[7] SCPWM Bit = 0

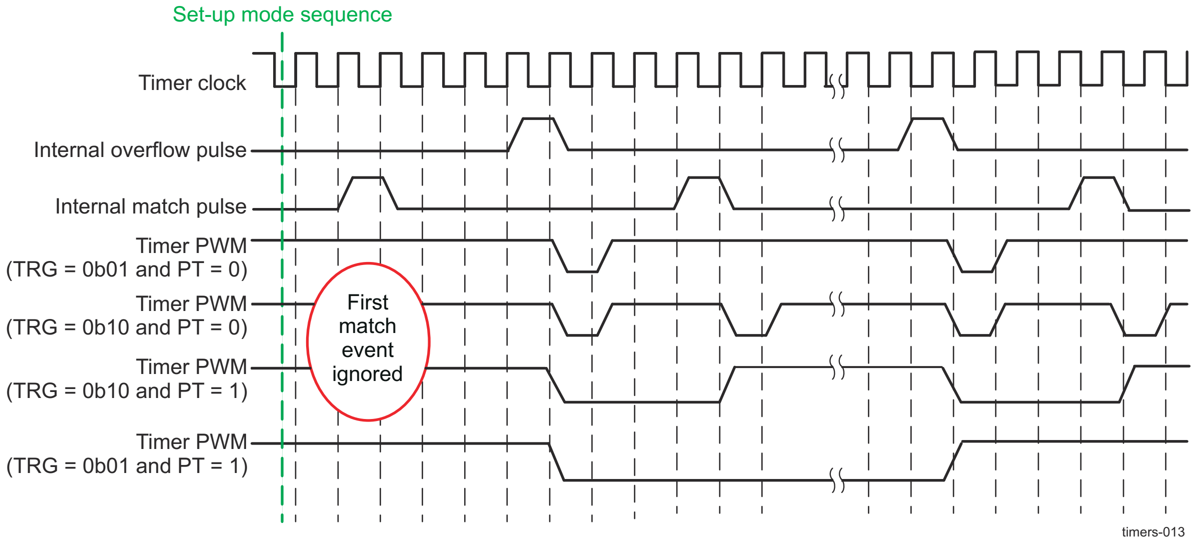


Figure 22-14. Timing Diagram of PWM With TCLR[7] SCPWM Bit = 1

22.2.4.11 Timer Counting Rate

The timer rate is defined by the following values:

- Value of the prescaler fields (the TCLR[5] PRE bit and TCLR[4:2] PTV bit field)
- Value loaded into the TLDR

Table 22-9 lists the prescaler clock ratio values.

Table 22-9. Prescaler Clock Ratio Values

TCLR[5] PRE	TCLR[4:2] PTV	Divisor (PS)
0	X	1
1	0	2
1	1	4
1	2	8
1	3	16
1	4	32
1	5	64
1	6	128
1	7	256

Thus, the timer overflow rate is expressed as:

OVF_Rate = (0xFFFF FFFF – TLDR + 1) × (timer-functional clock period) × PS

With (timer-functional clock period) = 1/(timer-functional clock frequency) and PS = $2^{(PTV + 1)}$ if prescaler is enabled, or PS = 1 if prescaler is disabled.

CAUTION

Internal resynchronization causes any write to the TCLR[1] ST bit to have some latency before the register is updated:

2.5 × functional clock cycles write_TIMER_TCLR_latency 3.5 × functional clock cycles

Remember to consider this latency whenever the timer must be started or stopped by a software change to the TCLR[1] ST bit.

CAUTION

- In non-PWM mode, TLDR must be maintained at less than or equal to 0xFFFF FFFE.
- In PWM mode, TLDR must be maintained at less than or equal to 0xFFFF FFFD.

For example, with a timer clock input of 32 kHz and the TCLR[5] PRE bit set to 0, the timer output period is as listed in Table 22-10.

Table 22-10. Value and Corresponding Interrupt Period

TLDR[31:0] LOAD_VALUE	Interrupt Period
0x0000 0000	37 h
0xFFFF 0000	2 s
0xFFFF FFF0	500 µs
0xFFFF FFFE	62.5 µs

22.2.4.12 Timer Under Emulation

During emulation mode, the timer continues to run according to the value of the TIOCP_ CFG[1] EMUFREE bit.

If the TIOCP_CFG[1] EMUFREE bit is set to 1, timer execution is not stopped in emulation mode and the interrupt is still generated when overflow or match is reached.

If the TIOCP_CFG[1] EMUFREE bit is set to 0, the prescaler and timer are frozen and both resume on exit from emulation mode. The asynchronous external input pin (timerx_pwm_evt, where x = [8:11]) is internally synchronized on two timer-clock rising edges.

22.2.4.13 Accessing GP Timer Registers

All accesses are nonposted until software reconfiguration. All registers are 32 bits wide, accessible through the OCP interface with 16- or 32-bit access (read/write).

Any 16-bit write access must be least-significant bit (LSB) first, and the second write access must be most-significant bit (MSB) first. Write operations to the following GP timer registers can skip the MSB access if it is not necessary to update the 16 MSBs of the register:

- TIDR (all GP timers)
- TIOCP_CFG (all GP timers)
- IRQSTATUS_SET (GP timers 1, 2, 10)
- IRQSTATUS_RAW (all GP timers)
- IRQSTATUS (all GP timers)
- IRQENABLE_SET(all GP timers except 1,2,10)
- IRQENABLE_CLR(all GP timers except 1,2,10)
- IRQSTATUS_SET (GP timers 1, 2, 10)
- IRQSTATUS_CLR (GP timers 1, 2, 10)
- IRQWAKEEN (all GP timers)
- TSICR (all GP timers)

Write operations to the following functional registers must be complete (the MSB must be written even if the MSB data is not used):

- TCLR (all GP timers)
- TCRR (all GP timers)
- TLDR (all GP timers)
- TTGR (all GP timers)
- TMAR (all GP timers)
- TPIR (GP timers 1, 2, 10)
- TNIR (GP timers 1, 2, 10)
- TCVR (GP timers 1, 2, 10)
- TOCR (GP timers 1, 2, 10)

- TOWR (GP timers1, 2, 10)

The following L4 synchronous registers are not affected by the posted/nonposted mode selection; the write/read operation is effective and acknowledged (command accepted) after one L4 clock cycle from command assertion:

- TIDR
- TIOCP_CFG
- IRQSTATUS
- IRQSTATUS_RAW
- IRQENABLE_SET
- IRQENABLE_CLR
- IRQSTATUS_SET
- IRQSTATUS_CLR
- IRQWAKEEN
- TWPS
- TSICR

22.2.4.13.1 Writing to Timer Registers

The host uses the OCP interface to write to the following registers synchronously with the timer interface clock:

- TLDR
- TCRR
- TCLR
- TIOCP_CFG
- IRQSTATUS
- IRQENABLE_SET
- IRQENABLE_CLR
- IRQWAKEEN
- TTGR
- TSICR
- TMAR

TIMER1, TIMER2, and TIMER10 also have the following registers:

- IRQSTATUS_SET
- IRQSTATUS_CLR
- TPIR
- TNIR
- TCVR
- TOCR
- TOWR

In 16-bit access mode, the 16 LSBs must be written before writing to the 16 MSBs.

22.2.4.13.1.1 Write Posting Synchronization Mode

This mode is used if the TSICR[2] POSTED bit is set to 1.

This mode uses a posted write scheme to update any internal register (TCLR, TCRR, TLDR, TTGR, TMAR, and TPIR, TNIR, TCVR, TOCR, and TOWR for TIMER1, TIMER2, and TIMER10). Therefore, the write transaction is immediately acknowledged on the open-core protocol (OCP) interface, although the effective write operation occurs later because of a resynchronization in the timer clock domain. The advantage is that neither the interconnect nor the device that requested the write transaction is stalled.

For each register, a status bit is provided in the timer write-posted status (TWPS) register. In this mode, it is mandatory that software check this status bit before any write access. If a write is attempted to a register with a previous access pending, the previous access is discarded without notice.

The timer module updates the value of the timer counter register synchronously with the OCP clock. Consequently, any read access to TCRR does not add any resynchronization latency; the current value is always available.

Note

Because the overflow IRQ is generated when the value of TCRR reaches 0xFFFF FFFF, and not when it changes its value to the value after overflow, it is necessary to wait a delay of (1 × PS × timer functional clock period) before any read access to TCRR to ensure a correct reading of its content.

Note

If TTGR register is written during posted write to TCRR, the value to be written to TCRR will be discarded.

If a posted write to TCVR is started, the user must not write to TPIR or TNIR before the TCVR write is finished, because the value of TCVR is re-evaluated, so both the value to be written, and the recalculated value will be discarded.

If a write access is pending for a register, reading from this register does not yield a correct result. Software synchronization must be used to avoid incorrect results.

Functional frequency range: freq(timer clock) < freq(OCP interface clock) / 4.

22.2.4.13.1.2 Write Nonposting Synchronization Mode

This mode is used if the TSICR[2] POSTED bit is set to 0 (default value). It uses a nonposted write scheme to update any internal register. Therefore, the write transaction is not acknowledged on the L4 interface until the effective write operation occurs after the resynchronization in the timer functional clock domain. The drawback is that the interconnect and the device that requested the write transaction are stalled during this period.

The same full resynchronization scheme is used for a read transaction, and the same stall period applies. A register read following a write to the same register is always coherent.

This mode is functional regardless of the ratio between the OCP interface frequency and the timer clock frequency.

22.2.4.13.2 Reading From Timer Counter Registers

In 16-bit access mode, reading the 16 LSBs from the timer counter registers (TCRR, TCAR1, and TCAR2) captures the current timer counter value. This must be followed by reading the 16 MSBs. The synchronization schemes for read posted and read non-posted transactions are the same as the corresponded write transactions described before.

Note

LSB/MSB accesses cannot be interleaved (that is, the sequence LSB register 1, LSB register 2, MSB register 1, MSB register 2 is not supported).

The TCRR is a 32-bit “atomic datum” and its 16-bit capture is done on the 16-bit LSB first to allow atomic LSB16 + MSB16 capture. This capture scheme is also performed for the TCAR1 and TCAR2 registers as they can be changed due to internal processes too. DSP 16 bit accesses can be interleaved with MCU 32 bit accesses.

Note

Reading of counter value of GPTimer5 through GPTimer8 should be done with delay of 10 L4_PER clock cycles when the functional clock source is 32kHz and the IPU CD is in the hardware AUTO state.

Note

Reading of counter value of GPTimer5 through GPTimer8 can be done without any delay when the functional clock source is 32kHz and the IPU CD is in software wakeup mode, or the static dependency between IPU and MPU is enabled.

22.2.4.13.2.1 Read Posted

This mode is functional whatever the ratio between the OCP interface frequency and the functional clock frequency are. The recommended functional frequency range is freq(timer) < freq(OCP) / 4.

Read posted mode is used if TSICR[2] POSTED = 0x1 or TSICR[3] READ_MODE is set to 0. This mode uses a posted-read scheme for reading any internal timer register. The read transaction is immediately acknowledged on the OCP interface, and the value to be read has been resynchronized. With this method, neither the interconnect nor the device that requested the read transaction are stalled.

Read posted mode applies to TCRR, TCAR1, TCAR2, TCVR, and TOWR, which need resynchronization from functional to OCP clock domains.

22.2.4.13.2.2 Read Non-Posted

This mode is functional regardless of the ratio between the OCP interface frequency and the functional clock frequency. Recommended functional frequency range is freq(timer) >= freq(OCP) / 4.

Read non-posted mode is used if TSICR[2] POSTED = 0x0 and TSICR[3] READ_MODE = 0x1. This mode uses a non-posted read scheme for reading internal timer registers. The read transaction is not acknowledged on the OCP interface until the effective read operation occurs, after the resynchronization in the timer clock domain. The result is that both the interconnect and the device that requested the read transaction are stalled during this period.

This mode applies to TCRR, TCAR1, TCAR2, TCVR, and TOWR, which need resynchronization from functional to OCP clock domains.

22.2.4.14 Posted Mode Selection

A choice between two synchronization modes is made taking into account the frequency ratio and the stall periods that can be supported by the system, without impacting the global performance.

The posted mode selection applies only to registers that require synchronization on or from the timer clock domain. For write operation, the registers affected by posted and non-posted selection are TCLR, TLDR, TCRR, TTGR, TMAR, TPIR, TNIR, TCVR, TOCR, and TOWR. For read operation, the registers affected by this selection are: TCRR, TCAR1, TCAR2, TCVR, and TOWR.

The OCP clock domain synchronous registers TIDR, TIOCP_CFG, IRQSTATUS, IRQSTATUS_SET, IRQWAKEEN, TWPS, and TSICR are not affected by posted and non-posted mode selection. The operation (read or write) is effective and acknowledged after one OCP clock cycle from the command assertion.

The configuration of posted or non-posted mode can be changed (overwritten) by software by writing in TSICR[2] POSTED bit. The TSICR[3] READ_MODE defines how the read operation is performed when the module is configured in non-posted mode (see TSICR). The following cases are possible:

- TSICR[2] POSTED = 0x1 and TSICR[3] READ_MODE = x (don't care): read and write operations are expected in posted mode.
- TSICR[2] POSTED = 0x0 and TSICR[3] READ_MODE = 0x0: the write operation is executed in non-posted mode and read is executed in posted mode.

- TSICR[2]POSTED = 0x0 and TSICR[3] READ_MODE = 0x1: write is executed in non-posted mode and read is executed in non-posted mode.

22.2.5 GP Timer Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

22.2.5.1 Global Initialization

22.2.5.1.1 Global Initialization of Surrounding Modules

This section identifies the requirements for initializing the surrounding modules when the GP timer module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the GP timer. For more information, see Section 22.2.3, *GP Timers Integration*, and Section 22.2.2, *GP Timers Environment*. Table 22-11 summarizes the surrounding modules.

Table 22-11. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. For more information about the module configuration, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.
Control module	The module-specific pad muxing must be set in the control module. For more information about the module configuration, see *Pad Configuration Registers,* in *Control Module*.
IRQ_CROSSBAR	The IRQ_CROSSBAR configuration must be done to enable the interrupts from the GP timer module. See *IRQ_CROSSBAR Module Functional Description*.

22.2.5.1.2 GP Timer Module Global Initialization

22.2.5.1.2.1 Main Sequence – GP Timer Module Global Initialization

Table 22-12 identifies the main steps for initializing the GP timer module when the module is to be used for the first time.

Table 22-12. GP Timer Module Global Initialization

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	TIOCP_CFG[0] SOFTRESET	0x1
Wait until reset release?	TIOCP_CFG[0] SOFTRESET	0x0
Configure idle mode.	TIOCP_CFG[3:2] IDLEMODE	0x-
Enable wake-up interrupt events.	IRQWAKEEN[2:0]	0x-
Select posted mode.	TSICR[2] POSTED	0x-

22.2.5.2 Operational Mode Configuration

22.2.5.2.1 GP Timer Mode

22.2.5.2.1.1 Main Sequence – GP Timer Mode Configuration

Table 22-13 lists the steps in the GP timer mode configuration.

Table 22-13. GP Timer Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1
Enable overflow interrupt.	IRQSTATUS_SET[1] OVF_EN_FLAG[(1)] or IRQENABLE_SET[1] OVF_EN_FLAG[(2)]	0x1
Load timer counter value.	TCRR	0x-
Load timer load value.	TLDR	0x-
Start the timer.	TCLR[0] ST	0x1

(1) Applies only to TIMER1, TIMER2, and TIMER10.
(2) Applies to TIMER3 through TIMER 9 and TIMER11 through TIMER16.

22.2.5.2.2 GP Timer Compare Mode

22.2.5.2.2.1 Main Sequence – GP Timer Compare Mode Configuration

Table 22-14 lists the steps in the GP timer compare mode configuration.

Table 22-14. GP Timer Compare Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1
Enable match interrupt.	IRQSTATUS_SET[0] MAT_EN_FLAG(1) or IRQENABLE_SET[0] MAT_EN_FLAG(2)	0x1
Load timer counter value.	TCRR	0x-
Load timer compare value.	TMAR	0x-
Enable compare mode.	TCLR[6] CE	0x1
Start the timer.	TCLR[0] ST	0x1

(1) Applies only to TIMER1, TIMER2, and TIMER10.
(2) Applies to TIMER3 through TIMER9 and TIMER11 through TIMER16.

22.2.5.2.3 GP Timer Capture Mode

22.2.5.2.3.1 Main Sequence – GP Timer Capture Mode Configuration

Table 22-15 lists the steps in the GP timer capture mode configuration.

Table 22-15. GP Timer Capture Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Initialize capture mode.	See Section 22.2.5.2.3.2.	
Enable capture interrupt.	IRQENABLE_SET[2] TCAR_EN_FLAG (2) or IRQSTATUS_SET[2] TCAR_EN_FLAG (1)	0x1
Start the timer.	TCLR[0] ST	0x1
Detect event.	See Section 22.2.5.2.3.3.	

(1) Applies only to TIMER1, TIMER2, and TIMER10.
(2) Applies only to TIMER3 through TIMER9 and TIMER11 through TIMER16.

22.2.5.2.3.2 Subsequence – Initialize Capture Mode

Table 22-16 lists the steps to initialize capture mode.

Table 22-16. Initialize Capture Mode

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1
Select TIMERi (where i = 2 to 11 and 13 to 16). Capture input at device pin timeri.	TCLR[14] GPO_CFG	0x1
Select single or second event capture.	TCLR[13] CAPT_MODE	0x-
Select transition capture mode.	TCLR[9:8] TCM	0x-

22.2.5.2.3.3 Subsequence – Detect Event

Table 22-17 lists the steps in detecting an event.

Table 22-17. Detect Event

Step	Register/Bit Field/Programming Model	Value
Wait until event detected?	IRQSTATUS[2] TCAR_IT_FLAG	= 0x1

Table 22-17. Detect Event (continued)

Step	Register/Bit Field/Programming Model	Value
Read timer capture value.	TCAR1 and/or TCAR2	
Clear capture interrupt request.	IRQSTATUS[2] TCAR_IT_FLAG	0x1

22.2.5.2.4 GP Timer PWM Mode

22.2.5.2.4.1 Main Sequence – GP Timer PWM Mode Configuration

Table 22-18 lists the steps in the GP timer PWM mode configuration.

Table 22-18. GP Timer PWM Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1
Select trigger output mode.	TCLR[11:10] TRG	0x-
Select pulse or toggle modulation PWM mode.	TCLR[12] PT	0x-
Select TIMERi (where i = 2 to 11 and 13 to 16) PWM output at device pin timeri.	TCLR[14] GPO_CFG	0x0
Configure PWM output pin default value.	TCLR[7] SCPWM	0x-
Load timer load value.	TLDR	0x-
Load timer compare value.	TMAR	0x-
Enable compare.	TCLR[6] CE	0x1
Start the timer.	TCLR[0] ST	0x1

22.2.6 GP Timer Register Manual

22.2.6.1 GP Timer Instance Summary

Table 22-19 lists the base address and block size for the GP timer module instances.

Table 22-19. GP Timer Instance Summary

Module Name	Base Address L4_PER1 Interconnect	Base Address L4_PER3 Interconnect	Base Address L4_WKUP Interconnect	Size
TIMER2	0x4803 2000	–	–	112 Bytes
TIMER3	0x4803 4000	–	–	92 Bytes
TIMER4	0x4803 6000	–	–	92 Bytes
TIMER9	0x4803 E000	–	–	92 Bytes
TIMER10	0x4808 6000	–	–	112 Bytes
TIMER11	0x4808 8000	–	–	92 Bytes
TIMER5	–	0x4882 0000	--	92 Bytes
TIMER6	–	0x4882 2000	--	92 Bytes
TIMER7	–	0x4882 4000	--	92 Bytes
TIMER8	–	0x4882 6000	--	92 Bytes
TIMER13	--	0x4882 8000	--	92 Bytes
TIMER14	--	0x4882 A000	--	92 Bytes
TIMER15	--	0x4882 C000	--	92 Bytes
TIMER16	--	0x4882 E000	--	92 Bytes
TIMER12	-	-	0x4AE2 0000	92 bytes
TIMER1	--	–	0x4AE1 8000	112 Bytes

22.2.6.2 GP Timer Registers

22.2.6.2.1 GP Timer Register Summary

CAUTION

The GP timer registers are limited to 32- and 16-bit data accesses; 8-bit access is not allowed and can corrupt the register content.

Table 22-20 through Table 22-25 provide the register summary and associated offset addresses for the 15 GP timer internal registers.

Table 22-20. TIMER1, TIMER2, and TIMER10 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER1 Physical Address L4_WKUP Interconnect	TIMER2 Physical Address L4_PER1 Interconnect	TIMER10 Physical Address L4_PER1 Interconnect
TIDR	RO	32	0x0000 0000	0x4AE1 8000	0x4803 2000	0x4808 6000
TIOCP_CFG	RW	32	0x0000 0010	0x4AE1 8010	0x4803 2010	0x4808 6010
IRQ_EOI	RW	32	0x0000 0020	0x4AE1 8020	0x4803 2020	0x4808 6020
IRQSTATUS_RAW	RW	32	0X0000 0024	0x4AE1 8024	0x4803 2024	0x4808 6024
IRQSTATUS	RW	32	0X0000 0028	0x4AE1 8028	0x4803 2028	0x4808 6028
IRQSTATUS_SET	RW	32	0X0000 002C	0x4AE1 802C	0x4803 202C	0x4808 602C
IRQSTATUS_CLR	RW	32	0X0000 0030	0x4AE1 8030	0x4803 2030	0x4808 6030
IRQWAKEEN	RW	32	0X0000 0034	0x4AE1 8034	0x4803 2034	0x4808 6034
TCLR	RW	32	0x0000 0038	0x4AE1 8038	0x4803 2038	0x4808 6038
TCRR	RW	32	0x0000 003C	0x4AE1 803C	0x4803 203C	0x4808 603C
TLDR	RW	32	0x0000 0040	0x4AE1 8040	0x4803 2040	0x4808 6040
TTGR	RW	32	0x0000 0044	0x4AE1 8044	0x4803 2044	0x4808 6044
TWPS	RO	32	0x0000 0048	0x4AE1 8048	0x4803 2048	0x4808 6048
TMAR	RW	32	0x0000 004C	0x4AE1 804C	0x4803 204C	0x4808 604C
TCAR1	RO	32	0x0000 0050	0x4AE1 8050	0x4803 2050	0x4808 6050
TSICR	RW	32	0x0000 0054	0x4AE1 8054	0x4803 2054	0x4808 6054
TCAR2	RO	32	0x0000 0058	0x4AE1 8058	0x4803 2058	0x4808 6058
TPIR	RW	32	0x0000 005C	0x4AE1 805C	0x4803 205C	0x4808 605C
TNIR	RW	32	0x0000 0060	0x4AE1 8060	0x4803 2060	0x4808 6060
TCVR	RW	32	0x0000 0064	0x4AE1 8064	0x4803 2064	0x4808 6064
TOCR	RW	32	0x0000 0068	0x4AE1 8068	0x4803 2068	0x4808 6068
TOWR	RW	32	0x0000 006C	0x4AE1 806C	0x4803 206C	0x4808 606C

Table 22-21. TIMER3 and TIMER4 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER3 Physical Address L4_PER1 Interconnect	TIMER4 Physical Address L4_PER1 Interconnect
TIDR	R	32	0x0000 0000	0x4803 4000	0x4803 6000
TIOCP_CFG	RW	32	0x0000 0010	0x4803 4010	0x4803 6010
IRQ_EOI	RW	32	0x0000 0020	0x4803 4020	0x4803 6020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4803 4024	0x4803 6024
IRQSTATUS	RW	32	0x0000 0028	0x4803 4028	0x4803 6028
IRQENABLE_SET	RW	32	0x0000 002C	0x4803 402C	0x4803 602C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4803 4030	0x4803 6030
IRQWAKEEN	RW	32	0x0000 0034	0x4803 4034	0x4803 6034
TCLR	RW	32	0x0000 0038	0x4803 4038	0x4803 6038
TCRR	RW	32	0x0000 003C	0x4803 403C	0x4803 603C
TLDR	RW	32	0x0000 0040	0x4803 4040	0x4803 6040
TTGR	RW	32	0x0000 0044	0x4803 4044	0x4803 6044
TWPS	R	32	0x0000 0048	0x4803 4048	0x4803 6048
TMAR	RW	32	0x0000 004C	0x4803 404C	0x4803 604C
TCAR1	R	32	0x0000 0050	0x4803 4050	0x4803 6050
TSICR	RW	32	0x0000 0054	0x4803 4054	0x4803 6054
TCAR2	R	32	0x0000 0058	0x4803 4058	0x4803 6058

Table 22-22. TIMER5, TIMER6 and TIMER7 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER5 Physical Address L4_PER3 Interconnect	TIMER6 Physical Address L4_PER3 Interconnect	TIMER7 Physical Address L4_PER3 Interconnect
TIDR	R	32	0x0000 0000	0x4882 0000	0x4882 2000	0x4882 4000
TIOCP_CFG	RW	32	0x0000 0010	0x4882 0010	0x4882 2010	0x4882 4010
IRQ_EOI	RW	32	0x0000 0020	0x4882 0020	0x4882 2020	0x4882 4020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4882 0024	0x4882 2024	0x4882 4024
IRQSTATUS	RW	32	0x0000 0028	0x4882 0028	0x4882 2028	0x4882 4028
IRQENABLE_SET	RW	32	0x0000 002C	0x4882 002C	0x4882 202C	0x4882 402C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4882 0030	0x4882 2030	0x4882 4030
IRQWAKEEN	RW	32	0x0000 0034	0x4882 0034	0x4882 2034	0x4882 4034
TCLR	RW	32	0x0000 0038	0x4882 0038	0x4882 2038	0x4882 4038
TCRR	RW	32	0x0000 003C	0x4882 003C	0x4882 203C	0x4882 403C
TLDR	RW	32	0x0000 0040	0x4882 0040	0x4882 2040	0x4882 4040
TTGR	RW	32	0x0000 0044	0x4882 0044	0x4882 2044	0x4882 4044
TWPS	R	32	0x0000 0048	0x4882 0048	0x4882 2048	0x4882 4048
TMAR	RW	32	0x0000 004C	0x4882 004C	0x4882 204C	0x4882 404C
TCAR1	R	32	0x0000 0050	0x4882 0050	0x4882 2050	0x4882 4050
TSICR	RW	32	0x0000 0054	0x4882 0054	0x4882 2054	0x4882 4054
TCAR2	R	32	0x0000 0058	0x4882 0058	0x4882 2058	0x4882 4058

Table 22-23. TIMER8, TIMER9 and TIMER11 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER8 Physical Address L4_PER3 Interconnect	TIMER9 Physical Address L4_PER1 Interconnect	TIMER11 Physical Address L4_PER1 Interconnect
TIDR	R	32	0x0000 0000	0x4882 6000	0x4803 E000	0x4808 8000
TIOCP_CFG	RW	32	0x0000 0010	0x4882 6010	0x4803 E010	0x4808 8010
IRQ_EOI	RW	32	0x0000 0020	0x4882 6020	0x4803 E020	0x4808 8020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4882 6024	0x4803 E024	0x4808 8024
IRQSTATUS	RW	32	0x0000 0028	0x4882 6028	0x4803 E028	0x4808 8028
IRQENABLE_SET	RW	32	0x0000 002C	0x4882 602C	0x4803 E02C	0x4808 802C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4882 6030	0x4803 E030	0x4808 8030
IRQWAKEEN	RW	32	0x0000 0034	0x4882 6034	0x4803 E034	0x4808 8034
TCLR	RW	32	0x0000 0038	0x4882 6038	0x4803 E038	0x4808 8038
TCRR	RW	32	0x0000 003C	0x4882 603C	0x4803 E03C	0x4808 803C
TLDR	RW	32	0x0000 0040	0x4882 6040	0x4803 E040	0x4808 8040
TTGR	RW	32	0x0000 0044	0x4882 6044	0x4803 E044	0x4808 8044
TWPS	R	32	0x0000 0048	0x4882 6048	0x4803 E048	0x4808 8048
TMAR	RW	32	0x0000 004C	0x4882 604C	0x4803 E04C	0x4808 804C
TCAR1	R	32	0x0000 0050	0x4882 6050	0x4803 E050	0x4808 8050
TSICR	RW	32	0x0000 0054	0x4882 6054	0x4803 E054	0x4808 8054
TCAR2	R	32	0x0000 0058	0x4882 6058	0x4803 E058	0x4808 8058

Table 22-24. TIMER13, TIMER14 and TIMER15 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER13 Physical Address L4_PER3 Interconnect	TIMER14 Physical Address L4_PER3 Interconnect	TIMER15 Physical Address L4_PER3 Interconnect
TIDR	R	32	0x0000 0000	0x4882 8000	0x4882 A000	0x4882 C000
TIOCP_CFG	RW	32	0x0000 0010	0x4882 8010	0x4882 A010	0x4882 C010
IRQ_EOI	RW	32	0x0000 0020	0x4882 8020	0x4882 A020	0x4882 C020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4882 8024	0x4882 A024	0x4882 C024
IRQSTATUS	RW	32	0x0000 0028	0x4882 8028	0x4882 A028	0x4882 C028
IRQENABLE_SET	RW	32	0x0000 002C	0x4882 802C	0x4882 A02C	0x4882 C02C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4882 8030	0x4882 A030	0x4882 C030
IRQWAKEEN	RW	32	0x0000 0034	0x4882 8034	0x4882 A034	0x4882 C034
TCLR	RW	32	0x0000 0038	0x4882 8038	0x4882 A038	0x4882 C038
TCRR	RW	32	0x0000 003C	0x4882 803C	0x4882 A03C	0x4882 C03C
TLDR	RW	32	0x0000 0040	0x4882 8040	0x4882 A040	0x4882 C040
TTGR	RW	32	0x0000 0044	0x4882 8044	0x4882 A044	0x4882 C044
TWPS	R	32	0x0000 0048	0x4882 8048	0x4882 A048	0x4882 C048
TMAR	RW	32	0x0000 004C	0x4882 804C	0x4882 A04C	0x4882 C04C
TCAR1	R	32	0x0000 0050	0x4882 8050	0x4882 A050	0x4882 C050
TSICR	RW	32	0x0000 0054	0x4882 8054	0x4882 A054	0x4882 C054
TCAR2	R	32	0x0000 0058	0x4882 8058	0x4882 A058	0x4882 C058

Table 22-25. TIMER16 and TIMER12 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER16 Physical Address L4_PER3 Interconnect	TIMER12 Physical Address L4_WKUP Interconnect
TIDR	R	32	0x0000 0000	0x4882 E000	0x4AE2 0000
TIOCP_CFG	RW	32	0x0000 0010	0x4882 E010	0x4AE2 0010
IRQ_EOI	RW	32	0x0000 0020	0x4882 E020	0x4AE2 0020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4882 E024	0x4AE2 0024
IRQSTATUS	RW	32	0x0000 0028	0x4882 E028	0x4AE2 0028
IRQENABLE_SET	RW	32	0x0000 002C	0x4882 E02C	0x4AE2 002C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4882 E030	0x4AE2 0030
IRQWAKEEN	RW	32	0x0000 0034	0x4882 E034	0x4AE2 0034
TCLR	RW	32	0x0000 0038	0x4882 E038	0x4AE2 0038
TCRR	RW	32	0x0000 003C	0x4882 E03C	0x4AE2 003C
TLDR	RW	32	0x0000 0040	0x4882 E040	0x4AE2 0040
TTGR	RW	32	0x0000 0044	0x4882 E044	0x4AE2 0044
TWPS	R	32	0x0000 0048	0x4882 E048	0x4AE2 0048
TMAR	RW	32	0x0000 004C	0x4882 E04C	0x4AE2 004C
TCAR1	R	32	0x0000 0050	0x4882 E050	0x4AE2 0050
TSICR	RW	32	0x0000 0054	0x4882 E054	0x4AE2 0054
TCAR2	R	32	0x0000 0058	0x4882 E058	0x4AE2 0058

22.2.6.2.2 GP Timer Register Description

Table 22-26 through Table 22-49 describe the individual GP timer registers.

Table 22-26. TIDR

Address Offset	0x0000 0000

Table 22-26. TIDR (continued)

Physical Address	0x4AE1 8000	**Instance**	TIMER1_WKUP_L4
	0x4803 2000		TIMER2_PER1_L4
	0x4808 6000		TIMER10_PER1_L4
	0x4803 4000		TIMER3_PER1_L4
	0x4803 6000		TIMER4_PER1_L4
	0x4882 0000		TIMER5_PER3_L4
	0x4882 2000		TIMER6_PER3_L4
	0x4882 4000		TIMER7_PER3_L4
	0x4882 6000		TIMER8_PER3_L4
	0x4803 E000		TIMER9_PER1_L4
	0x4808 8000		TIMER11_PER1_L4
	0x4882 8000		TIMER13_PER3_L4
	0x4882 A000		TIMER14_PER3_L4
	0x4882 C000		TIMER15_PER3_L4
	0x4882 E000		TIMER16_PER3_L4
	0x4AE2 0000		TIMER12_WKUP_L4
Description	This read-only register contains the revision number of the module. A write to this register has no effect. This register is used by software to track features, bugs, and compatibility.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31: 0	REVISION	IP Revision	R	0x– [1]

(1) TI internal data

Table 22-27. TIOCP_CFG

Address Offset	0x0000 0010		
Physical Address	0x4AE1 8010	**Instance**	TIMER1_WKUP_L4
	0x4803 2010		TIMER2_PER1_L4
	0x4808 6010		TIMER10_PER1_L4
	0x4803 4010		TIMER3_PER1_L4
	0x4803 6010		TIMER4_PER1_L4
	0x4882 0010		TIMER5_PER3_L4
	0x4882 2010		TIMER6_PER3_L4
	0x4882 4010		TIMER7_PER3_L4
	0x4882 6010		TIMER8_PER3_L4
	0x4803 E010		TIMER9_PER1_L4
	0x4808 8010		TIMER11_PER1_L4
	0x4882 8010		TIMER13_PER3_L4
	0x4882 A010		TIMER14_PER3_L4
	0x4882 C010		TIMER15_PER3_L4
	0x4882 E010		TIMER16_PER3_L4
	0x4AE2 0010		TIMER12_WKUP_L4
Description	This register controls the various parameters of the L4 interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		EMUFREE	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000

Bits	Field Name	Description	Type	Reset
3:2	IDLEMODE	Power management, req/ack control 0x0: Force-idle mode: local target idle state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the IP module internal requirements. Back-up mode, for debug only. 0x1: No-idle mode: local target never enters idle state. Back-up mode, for debug only. 0x2: Smart-idle mode: local target idle state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module should not generate (IRQ- or DMA-request-related) wake-up events. 0x3: Smart-idle wake-up-capable mode: local target idle state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. Mode is relevant only if the appropriate IP module *swake-up* output(s) is (are) implemented.	RW	0x0
1	EMUFREE	Emulation mode 0x0: The timer is frozen in emulation mode (PINSUSPENDN signal active). 0x1: The timer runs free, regardless of PINSUSPENDN value.	RW	0
0	SOFTRESET	Software reset 0x0: Read 0: reset done, no pending action Write 0: No action 0x1: Read 1: initiate software reset Write 1: Reset ongoing	RW	0

Table 22-28. IRQ_EOI

Address Offset	0x0000 0020		
Physical Address	0x4AE1 8020 0x4803 2020 0x4808 6020 0x4803 4020 0x4803 6020 0x4882 0020 0x4882 2020 0x4882 4020 0x4882 6020 0x4803 E020 0x4808 8020 0x4882 8020 0x4882 A020 0x4882 C020 0x4882 E020 0x4AE2 0020	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if a new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LINE_NUMBER	Write the number of the interrupt line to apply a SW EOI to it. Note that there is only a single line (i.e. number 0). Read: Read always returns 0 Write 0: SW EOI on interrupt line Write 1: No action	RW	0x0

Table 22-29. IRQSTATUS_RAW

Address Offset	0x0000 0024		
Physical Address	0x4AE1 8024	**Instance**	TIMER1_WKUP_L4
	0x4803 2024		TIMER2_PER1_L4
	0x4808 6024		TIMER10_PER1_L4
	0x4803 4024		TIMER3_PER1_L4
	0x4803 6024		TIMER4_PER1_L4
	0x4882 0024		TIMER5_PER3_L4
	0x4882 2024		TIMER6_PER3_L4
	0x4882 4024		TIMER7_PER3_L4
	0x4882 6024		TIMER8_PER3_L4
	0x4803 E024		TIMER9_PER1_L4
	0x4808 8024		TIMER11_PER1_L4
	0x4882 8024		TIMER13_PER3_L4
	0x4882 A024		TIMER14_PER3_L4
	0x4882 C024		TIMER15_PER3_L4
	0x4882 E024		TIMER16_PER3_L4
	0x4AE2 0024		TIMER12_WKUP_L4
Description	Component interrupt-request status. Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
Type	RW		

31–3	2	1	0
RESERVED	TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_IT_FLAG	IRQ status for capture Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0
1	OVF_IT_FLAG	IRQ status for overflow Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0

Bits	Field Name	Description	Type	Reset
0	MAT_IT_FLAG	IRQ status for match Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0

Table 22-30. IRQSTATUS

Address Offset	0x0000 0028		
Physical Address	0x4AE1 8028 0x4803 2028 0x4808 6028 0x4803 4028 0x4803 6028 0x4882 0028 0x4882 2028 0x4882 4028 0x4882 6028 0x4803 E028 0x4808 8028 0x4882 8028 0x4882 A028 0x4882 C028 0x4882 E028 0x4AE2 0028	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	Component interrupt-request status. Check the corresponding secondary status register. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
Type	RW		

31–3	2	1	0
RESERVED	TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_IT_FLAG	IRQ status for capture Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0
1	OVF_IT_FLAG	IRQ status for overflow Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0
0	MAT_IT_FLAG	IRQ status for match Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0

Table 22-31. IRQENABLE_SET

Address Offset	0x0000 002C

Table 22-31. IRQENABLE_SET (continued)

Physical Address	0x4803 402C 0x4803 602C 0x4882 002C 0x4882 202C 0x4882 402C 0x4882 602C 0x4803 E02C 0x4808 802C 0x4882 802C 0x4882 A02C 0x4882 C02C 0x4882 E02C 0x4AE2 002C	**Instance**	TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	Component interrupt-request enable. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31:3	2	1	0
RESERVED	TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled Write 1: Set IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0

Table 22-32. IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	0x4803 4030 0x4803 6030 0x4882 0030 0x4882 2030 0x4882 4030 0x4882 6030 0x4803 E030 0x4808 8030 0x4882 8030 0x4882 A030 0x4882 C030 0x4882 E030 0x4AE2 0030	**Instance**	TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	Component interrupt-request enable. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		

Table 22-32. IRQENABLE_CLR (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0

Table 22-33. IRQWAKEEN

Address Offset	0x0000 0034		
Physical Address	0x4AE1 8034 0x4803 2034 0x4808 6034 0x4803 4034 0x4803 6034 0x4882 0034 0x4882 2034 0x4882 4034 0x4882 6034 0x4803 E034 0x4808 8034 0x4882 8034 0x4882 A034 0x4882 C034 0x4882 E034 0x4AE2 0034	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	Wake-up-enabled events taking place when module is idle should generate an asynchronous wake-up.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_WUP_ENA	OVF_WUP_ENA	MAT_WUP_ENA

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_WUP_ENA	Wake-up generation for compare 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0
1	OVF_WUP_ENA	Wake-up generation for overflow 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0
0	MAT_WUP_ENA	Wake-up generation for match 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0

Table 22-34. TCLR

Address Offset	0x0000 0038		
Physical Address	0x4AE1 8038 0x4803 2038 0x4808 6038 0x4803 4038 0x4803 6038 0x4882 0038 0x4882 2038 0x4882 4038 0x4882 6038 0x4803 E038 0x4808 8038 0x4882 8038 0x4882 A038 0x4882 C038 0x4882 E038 0x4AE2 0038	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	This register controls optional features specific to the timer functionality.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	GPO_CFG	CAPT_MODE	PT	TRG		TCM		SCPWM	CE	PRE	PTV			AR	ST

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	GPO_CFG	General-purpose output - this register directly drives the PO_GPOCFG output. For specific use of the GPO_CFG bit, see *GP Timer External System Interface*. 0x0: PO_GPOCFG drives 0. 0x1: PO_GPOCFG drives 1.	RW	0
13	CAPT_MODE	Capture mode select bit (first/second) 0x0: Single capture: Capture the first enabled capture event in TCAR1. 0x1: Capture on second event: Capture the second enabled capture event in TCAR1 and the second enabled capture event in TCAR2.	RW	0
12	PT	Pulse or toggle mode on TIMERi_PWM_out output pin 0x0: Pulse modulation 0x1: Toggle modulation	RW	0

Bits	Field Name	Description	Type	Reset
11:10	TRG	Trigger output mode on TIMERi_PWM_out output pin 0x0: No trigger 0x1: Trigger on overflow. 0x2: Trigger on overflow and match. 0x3: Reserved	RW	0x0
9:8	TCM	Transition capture mode on TIMERi_EVENT_CAPTURE input pin (When the TCM field passed from (00) to any other combination, the TCAR_IT_FLAG and the edge detection logic are cleared.) 0x0: No capture 0x1: Capture on rising edges of TIMERi_EVENT_CAPTURE pin 0x2: Capture on falling edges of TIMERi_EVENT_CAPTURE pin 0x3: Capture on both edges of TIMERi_EVENT_CAPTURE pin	RW	0x0
7	SCPWM	Pulse width modulation output pin default setting This bit must be set or clear while the timer is stopped or the trigger is off. 0x0: Clear the TIMERi_PWM_out output pin and select positive pulse for pulse mode. 0x1: Set the TIMERi_PWM_out output pin and select negative pulse for pulse mode.	RW	0
6	CE	Compare enable 0x0: Compare mode is disable. 0x1: Compare mode is enable.	RW	0
5	PRE	Prescaler enable 0x0: The TIMER clock input pin clocks the counter. 0x1: The divided input pin clocks the counter.	RW	0
4:2	PTV	Prescale clock timer value The timer counter is prescaled with the value $2^{(PTV+1)}$. Example: PTV = 3, counter increases value (if started) after 16 functional clock periods.	RW	0x0
1	AR	Autoreload mode 0x0: One shot timer 0x1: Autoreload timer	RW	0
0	ST	Start/stop timer control 0x0: Stop timer: Only the counter is frozen. If one-shot mode selected (AR =0), this bit is automatically reset by internal logic when the counter is overflowed. 0x1: Start timer	RW	0

Table 22-35. TCRR

Address Offset	0x0000 003C

Table 22-35. TCRR (continued)

Physical Address	0x4AE1 803C 0x4803 203C 0x4808 603C 0x4803 403C 0x4803 603C 0x4882 003C 0x4882 203C 0x4882 403C 0x4882 603C 0x4803 E03C 0x4808 803C 0x4882 803C 0x4882 A03C 0x4882 C03C 0x4882 E03C 0x4AE2 003C	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	This register holds the value of the internal counter.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER_COUNTER																															

Bits	Field Name	Description	Type	Reset
31:0	TIMER_COUNTER	Value of TIMER counter	RW	0x0000 0000

Table 22-36. TLDR

Address Offset	0x0000 0040		
Physical Address	0x4AE1 8040 0x4803 2040 0x4808 6040 0x4803 4040 0x4803 6040 0x4882 0040 0x4882 2040 0x4882 4040 0x4882 6040 0x4803 E040 0x4808 8040 0x4882 8040 0x4882 A040 0x4882 C040 0x4882 E040 0x4AE2 0040	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	This register holds the timer load value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	LOAD_VALUE	Timer counter value loaded on overflow in autoreload mode or on TTGR write access. LOAD_VALUE must be different than the timer overflow value (0xFFFF FFFF).	RW	0x0000 0000

Table 22-37. TTGR

Address Offset	0x0000 0044

Table 22-37. TTGR (continued)

Physical Address	0x4AE1 8044 0x4803 2044 0x4808 6044 0x4803 4044 0x4803 6044 0x4882 0044 0x4882 2044 0x4882 4044 0x4882 6044 0x4803 E044 0x4808 8044 0x4882 8044 0x4882 A044 0x4882 C044 0x4882 E044 0x4AE2 0044	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	The read value of this register is always 0xFFFF FFFF.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTGR_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	TTGR_VALUE	Writing to the TTGR register causes the TCRR to be loaded from TLDR and the prescaler counter to be cleared. Reload is done regardless of the AR field value of the TCLR register.	RW Rreturns1s	0xFFFF FFFF

Table 22-38. TWPS

Address Offset	0x0000 0048		
Physical Address	0x4AE1 8048 0x4803 2048 0x4808 6048 0x4803 4048 0x4803 6048 0x4882 0048 0x4882 2048 0x4882 4048 0x4882 6048 0x4803 E048 0x4808 8048 0x4882 8048 0x4882 A048 0x4882 C048 0x4882 E048 0x4AE2 0048	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	This register contains the write posting bits for all writable functional registers.		
Type	R		

31:10	9	8	7	6	5	4	3	2	1	0
RESERVED	W_PEND_TOWR	W_PEND_TOCR	W_PEND_TCVR	W_PEND_TNIR	W_PEND_TPIR	W_PEND_TMAR	W_PEND_TTGR	W_PEND_TLDR	W_PEND_TCRR	W_PEND_TCLR

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0000000

Bits	Field Name	Description	Type	Reset
9	W_PEND_TOWR	Write pending for the TOWR register Read 1: Write pending Read 0: No write pending	R	0
8	W_PEND_TOCR	Write pending for the TOCR register Read 1: Write pending Read 0: No write pending	R	0
7	W_PEND_TCVR	Write pending for the TCVR register Read 1: Write pending Read 0: No write pending	R	0
6	W_PEND_TNIR	Write pending for the TNIR register Read 1: Negative increment register write pending Read 0: No negative increment register write pending	R	0
5	W_PEND_TPIR	Write pending for the TPIR register Read 1: Positive increment register write pending Read 0: No positive increment register write pending	R	0
4	W_PEND_TMAR	When equal to 1, a write is pending to the TMAR register.	R	0
3	W_PEND_TTGR	When equal to 1, a write is pending to the TTGR register.	R	0
2	W_PEND_TLDR	When equal to 1, a write is pending to the TLDR register.	R	0
1	W_PEND_TCRR	When equal to 1, a write is pending to the TCRR register.	R	0
0	W_PEND_TCLR	When equal to 1, a write is pending to the TCLR register.	R	0

Table 22-39. TMAR

Address Offset	0x0000 004C		
Physical Address	0x4AE1 804C	**Instance**	TIMER1_WKUP_L4
	0x4803 204C		TIMER2_PER1_L4
	0x4808 604C		TIMER10_PER1_L4
	0x4803 404C		TIMER3_PER1_L4
	0x4803 604C		TIMER4_PER1_L4
	0x4882 004C		TIMER5_PER3_L4
	0x4882 204C		TIMER6_PER3_L4
	0x4882 404C		TIMER7_PER3_L4
	0x4882 604C		TIMER8_PER3_L4
	0x4803 E04C		TIMER9_PER1_L4
	0x4808 804C		TIMER11_PER1_L4
	0x4882 804C		TIMER13_PER3_L4
	0x4882 A04C		TIMER14_PER3_L4
	0x4882 C04C		TIMER15_PER3_L4
	0x4882 E04C		TIMER16_PER3_L4
	0x4AE2 004C		TIMER12_WKUP_L4
Description	The compare logic consists of a 32-bit-wide, read/write data TMAR register and logic to compare counter.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMPARE_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COMPARE_VALUE	Value to be compared to the timer counter	RW	0x0000 0000

Table 22-40. TCAR1

Address Offset	0x0000 0050

Table 22-40. TCAR1 (continued)

Physical Address	0x4AE1 8050 0x4803 2050 0x4808 6050 0x4803 4050 0x4803 6050 0x4882 0050 0x4882 2050 0x4882 4050 0x4882 6050 0x4803 E050 0x4808 8050 0x4882 8050 0x4882 A050 0x4882 C050 0x4882 E050 0x4AE2 0050	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	This register holds the first captured value of the counter register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_VALUE1																															

Bits	Field Name	Description	Type	Reset
31:0	CAPTURE_VALUE1	First timer counter value captured on an external event trigger	R	0x0000 0000

Table 22-41. TSICR

Address Offset	0x0000 0054		
Physical Address	0x4AE1 8054 0x4803 2054 0x4808 6054 0x4803 4054 0x4803 6054 0x4882 0054 0x4882 2054 0x4882 4054 0x4882 6054 0x4803 E054 0x4808 8054 0x4882 8054 0x4882 A054 0x4882 C054 0x4882 E054 0x4AE2 0054	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	Timer synchronous interface control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												READ_MODE	POSTED	SFT	RESERVED

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x000 0000
3	READ_MODE	Select posted/non-posted mode for read operation: 0x0: When the module is configured in non-posted mode (POSTED = '0'), the read operation is executed as read posted.	RW	0

Bits	Field Name	Description	Type	Reset
		0x1: When the module is configured in non-posted mode (POSTED = '0'), the read operation is executed as read non-posted. NOTE: When the module is configured in posted mode (POSTED = '1'), this bit is not used. NOTE: For GP TIMER1, TIMER2 and TIMER10 this bit is write only.		
2	POSTED	Posted mode selection 0x0: Posted mode inactive: Delay the command accept output signal. 0x1: Posted mode active	RW	0
1	SFT	This bit resets all the functional part of the module. 0x0: Software reset is disabled. 0x1: Software reset is enabled.	RW	0
0	RESERVED	Reserved	R	0

Table 22-42. TCAR2

Address Offset	0x0000 0058		
Physical Address	0x4AE1 8058 0x4803 2058 0x4808 6058 0x4803 4058 0x4803 6058 0x4882 0058 0x4882 2058 0x4882 4058 0x4882 6058 0x4803 E058 0x4808 8058 0x4882 8058 0x4882 A058 0x4882 C058 0x4882 E058 0x4AE2 0058	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4 TIMER3_PER1_L4 TIMER4_PER1_L4 TIMER5_PER3_L4 TIMER6_PER3_L4 TIMER7_PER3_L4 TIMER8_PER3_L4 TIMER9_PER1_L4 TIMER11_PER1_L4 TIMER13_PER3_L4 TIMER14_PER3_L4 TIMER15_PER3_L4 TIMER16_PER3_L4 TIMER12_WKUP_L4
Description	This register holds the second captured value of the counter register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_VALUE2																															

Bits	Field Name	Description	Type	Reset
31:0	CAPTURE_VALUE2	Second timer counter value captured on an external event trigger	R	0x0000 0000

22.2.6.2.3 TIMER1, TIMER2, and TIMER10 Register Description

Table 22-43. TPIR

Address Offset	0x0000 005C		
Physical Address	0x4AE1 805C 0x4803 205C 0x4808 605C	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4
Description	This register is used for 1-ms tick generation. The TPIR register holds the value of the positive increment. The value of this register is added to the value of TCVR to determine whether next value loaded in TCRR is the subperiod value or the overperiod value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

POSITIVE_INC_VALUE

Bits	Field Name	Description	Type	Reset
31:0	POSITIVE_INC_VALUE	Value of the positive increment	RW	0x0000 0000

Table 22-44. TNIR

Address Offset	0x0000 0060		
Physical Address	0x4AE1 8060 0x4803 2060 0x4808 6060	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4
Description	This register is used for 1-ms tick generation. The TNIR register holds the value of the negative increment. The value of this register is added to the value of the TCVR to determine whether next value loaded in TCRR is the subperiod value or the overperiod value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEGATIVE_INV_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	NEGATIVE_INV_VALUE	Value of the negative increment	RW	0x0000 0000

Table 22-45. TCVR

Address Offset	0x0000 0064		
Physical Address	0x4AE1 8064 0x4803 2064 0x4808 6064	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4
Description	This register is used for 1-ms tick generation. The TCVR register determines whether next value loaded in TCRR is the subperiod value or the overperiod value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER_VALUE	Value of CVR counter	RW	0x0000 0000

Table 22-46. TOCR

Address Offset	0x0000 0068		
Physical Address	0x4AE1 8068 0x4803 2068 0x4808 6068	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4
Description	This register is used to mask the tick interrupt for a selected number of ticks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OVF_COUNTER_VALUE																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0.	R	0x00
23:0	OVF_COUNTER_VALUE	Number of overflow events	RW	0x000000

Table 22-47. TOWR

Address Offset	0x0000 006C

Table 22-47. TOWR (continued)

Physical Address	0x4AE1 806C 0x4803 206C 0x4808 606C	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4
Description	This register holds the number of masked overflow interrupts.		
Type	RW		

31–24	23–0
RESERVED	OVF_WRAPPING_VALUE

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0.	R	0x00
23:0	OVF_WRAPPING_VALUE	Number of masked interrupts	RW	0x000000

Table 22-48. IRQSTATUS_SET

Address Offset	0x0000 002C		
Physical Address	0x4AE1 802C 0x4803 202C 0x4808 602C	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4
Description	Component interrupt-request enable. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31–3	2	1	0
RESERVED	TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled Write 1: Set IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0

Table 22-49. IRQSTATUS_CLR

Address Offset	0x0000 0030		
Physical Address	0x4AE1 8030 0x4803 2030 0x4808 6030	**Instance**	TIMER1_WKUP_L4 TIMER2_PER1_L4 TIMER10_PER1_L4
Description	Component interrupt-request enable. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		

Table 22-49. IRQSTATUS_CLR (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0

22.3 32-kHz Synchronized Timer (COUNTER_32K)

22.3.1 32-kHz Synchronized Timer Overview

The 32-kHz synchronized timer (COUNTER_32K) is a 32-bit counter clocked by the falling edge of the 32-kHz system clock.

Figure 22-15 is the block diagram of the 32-kHz synchronized timer.

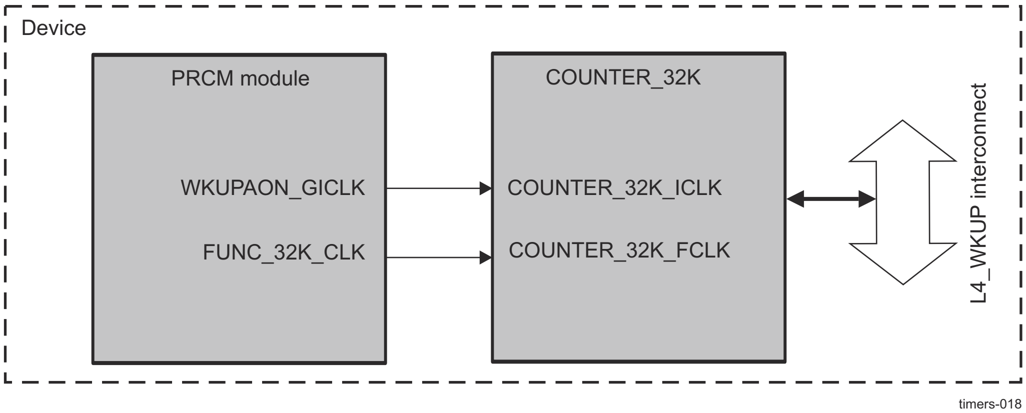


Figure 22-15. 32-kHz Synchronized Timer Block Diagram

22.3.1.1 32-kHz Synchronized Timer Features

The main features of the 32-kHz synchronized timer controller are:

- L4 slave interface (OCP) support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 16-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode not supported
- Only read operations are supported on the module registers; no write operation is supported (no error/no action on write).
- Free-running 32-bit upward counter
- Start and keep counting after power-on reset
- Automatic roll over to 0; highest value reached: 0xFFFF FFFF
- On-the-fly read (while counting)

22.3.2 32-kHz Synchronized Timer Integration

The synchronized timer is accessible only through the L4_WKUP interface.

Table 22-50 through Table 22-52 summarize the integration of the module in the device.

Table 22-50. COUNTER_32K Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
COUNTER_32K	PD_WKUPAON	No	L4_WKUP

Table 22-51. COUNTER_32K Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
COUNTER_32K	COUNTER_32K _FCLK	FUNC_32K_CLK	PRCM	COUNTER_32K functional 32KHz clock
COUNTER_32K	COUNTER_32K _ICLK	WKUPAON_GICLK	PRCM	COUNTER_32K interface clock
Resets				
COUNTER_32K	COUNTER_32K _NRESPWRON	WKUPAON_SYS_PWRON _RST	PRM	Reset to COUNTER_32K. Reset all internal logic, running on COUNTER_32K_FCLK
COUNTER_32K	COUNTER_32K _OCPRESETN	WKUPAON_RST	PRM	Reset to COUNTER_32K. Reset all L4 interface logic, running on COUNTER_32K_ICLK.

Table 22-52. COUNTER_32K Hardware Requests

No Interrupt Requests
No DMA Requests

22.3.3 32-kHz Synchronized Timer Functional Description

The synchronized timer is a counter that starts on the rising edge of an external asynchronous signal (COUNTER_32K_NRESPWRON). When COUNTER_32K_NRESPWRON is released (on the rising edge of COUNTER_32K_FCLK), the counter starts counting up from the reset value of the counter register on the falling edge of the 32-kHz COUNTER_32K_FCLK clock after three inverted 32-kHz clock periods. After reaching its highest value, the counter wraps back to 0 and starts counting again with no additional delay.

Figure 22-16 shows the reset synchronization timing diagram.

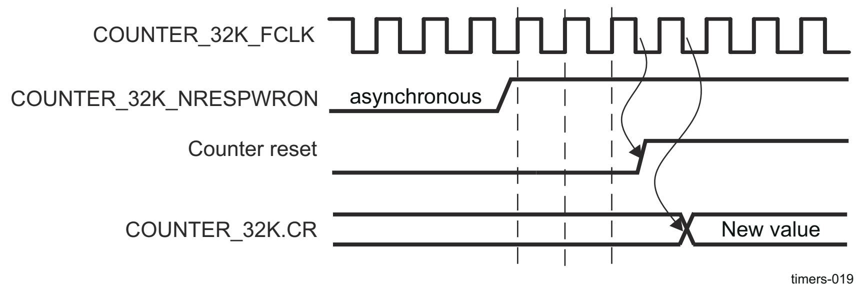


Figure 22-16. Reset Resynchronization Timing Diagram

Figure 22-17 is the block diagram of the synchronized timer.

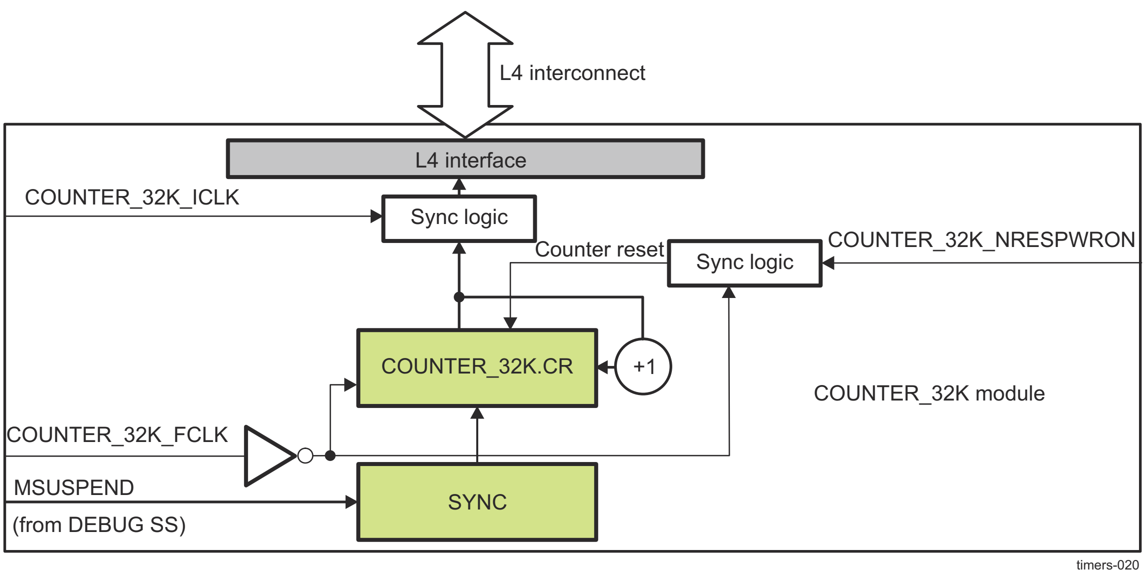


Figure 22-17. CONTER_32K Block Diagram

The sync logic ensures the correctness of the read transaction by synchronizing the counter register read access on COUNTER_32K_ICLK, because the COUNTER_32K_ICLK clock signal is completely asynchronous with COUNTER_32K_FCLK. The COUNTER_32K_NRESPWRON input resets the counter register (CR). The inverted COUNTER_32K_FCLK clocks the counter register CR.

The counting is temporally stopped when MSUSPEND control signal (coming from DEBUG SS) is asserted.

22.3.3.1 Reading the 32-kHz Synchronized Timer

The counter register (CR) is 32 bits wide. For correct count capture, it must be accessed as 16-bit LSB access first and 16-bit MSB access next. The value of the counter is read through the L4 interconnect slave interface. Internal synchronization logic allows the reading of the counter value with COUNTER_32K_ICLK while the counter is running. The time latency to read the synchronized counter register is one COUNTER_32K_ICLK clock period.

The user can select between two syncronization schemes by setting SYSCONFIG[0]SYNCMODE bit.

- SYSCONFIG[0]SYSCMODE = 0x0 - default. In this mode COUNTER_32K timer uses Gray encode/decode scheme. When the L4 interface sends a LSB16 read request command, the 32 bit coded value is registered directly to the interface domain. Due to the characteristics of this encoding if a read command arrives during a count up event either the old or the new value of CR is captured, not a transient value. The captured value will be decoded and send on the SDATA bus. The MSB16 read command reads upper 16 bits of the 32 bit value of counter register captured during the last LSB16 read access.
- SYSCONFIG[0]SYNCMODE = 0x1 - legacy synchronization scheme. In this mode the value of the CR is synchronized to the OCP domain at every count up event. This synchronization is possible because the COUNTER_32K_ICLK is much faster that the 32KHz clock (COUNTER_32K_FCLK). The drawback of this method is that if the interface clock is switched back after wake up from idle mode, the syncronized value will be updated only at the next count up event, until then it will be incorrect.

22.3.4 COUNTER_32K Timer Register Manual

Table 22-53 lists the base address and block size for the 32-kHz synchronized timer. It is memory-mapped to the L4 peripheral bus memory space.

Table 22-53. COUNTER_32K Timer Instance Summary

Module Name	Module Base Address	Size
L4_WKUP_COUNTER_32K	0x4AE0 4000	52 Bytes

22.3.4.1 COUNTER_32K Timer Register Mapping Summary

CAUTION

The 32-kHz synchronized timer registers are limited to 32- and 16-bit data accesses; 8-bit access is not allowed and can corrupt the register content.

Table 22-54 lists the 32-kHz synchronized timer registers. Table 22-55 through Table 22-59 describe the register bits.

Table 22-54. COUNTER_32K Timer Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_WKUP_COUNTER_32K Base Address
REVISION	R	32	0x0000 0000	0x4AE0 4000
SYSCONFIG	RW	32	0x0000 0010	0x4AE0 4010
CR	R	32	0x0000 0030	0x4AE0 4030

22.3.4.2 COUNTER_32K Timer Register Description

Table 22-55. REVISION

Address Offset	0x0 0000
Physical Address	0x4AE0 4000
Description	This register contains the sync counter IP revision code.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	0x-[1]

(1) TI internal data

Table 22-56. Register Call Summary for Register REVISION

32-kHz Synchronized Timer (COUNTER_32K)

- COUNTER_32K Timer Register Mapping Summary: [0]
- COUNTER_32K Timer Register Description: [1]

Table 22-57. SYSCONFIG

Address Offset	0x0 0010
Physical Address	0x4AE0 4010
Description	This register is used for idle modes only.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											IDLEMODE		Reserved		SYNCMODE

Bits	Field Name	Description	Type	Reset
31:5	Reserved	Reads return 0.	R	0x0
4:3	IDLEMODE	Power management REQ/ACK control 0x0: Force-idle. An IDLE request is acknowledged unconditionally. 0x1: No-idle. An IDLE request is never acknowledged. 0x2: Reserved 0x3: Reserved	RW	0x0
2:1	Reserved	Reads return 0.	R	0x0
0	SYNCMODE	Synchronization scheme 0x0 Gray synchronization scheme. Ensures that a stable value of the CR register is read. 0x1 Legacy synchronization scheme.	RW	0x0

Table 22-58. Register Call Summary for Register SYSCONFIG

32-kHz Synchronized Timer (COUNTER_32K)

- Reading the 32-kHz Synchronized Timer: [0] [1] [2]
- COUNTER_32K Timer Register Mapping Summary: [3]

Table 22-59. CR

Address Offset	0x0 0030
Physical Address	0x4AE0 4030
Description	This register contains the 32-kHz sync counter value.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER_VALUE	Counter register value	R	0x00000003

Table 22-60. Register Call Summary for Register CR

32-kHz Synchronized Timer (COUNTER_32K)

- 32-kHz Synchronized Timer Functional Description: [0]
- Reading the 32-kHz Synchronized Timer: [1] [2] [3]
- COUNTER_32K Timer Register Mapping Summary: [4]
- COUNTER_32K Timer Register Description: [5]

22.4 Watchdog Timer

22.4.1 Watchdog Timer Overview

The device includes one instance of the 32-bit watchdog timer: WD_TIMER2. Figure 22-18 shows how the timer is connected in the device.

The watchdog timer is an upward counter capable of generating a pulse on the reset pin and an interrupt to the device system modules following an overflow condition. The WD_TIMER2 timer serves resets to the PRCM module (its interrupt outputs are unused).

WD_TIMER2 is located in the PD_WKUPAON power domain, and can run when the device is in lowest power state above RTC mode (all power domains are off except always-on (AON), WKUP, and RTC domains).

The watchdog timer can be accessed, loaded, and cleared by registers through the L4_WKUP interface. The watchdog timer has the 32-kHz clock for its timer clock input. WD_TIMER2 directly generates a warm reset condition on overflow.

WD_TIMER2 connects to a single target agent port on the L4_WKUP interconnect.

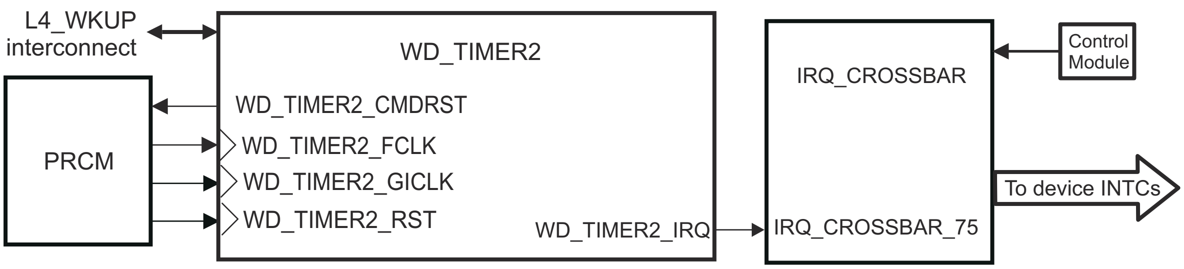


Figure 22-18. Watchdog Timer Block Diagram

Table 22-61 lists the default state of the watchdog timer in the device.

Table 22-61. Watchdog Timer Default State

Timer	Default State	
WD_TIMER2	Enabled	Not running

Note

The default state of the watchdog timer described in Table 22-61 is considered to be its state immediately after ROM code execution. For more information, see *Initialization*.

22.4.1.1 Watchdog Timer Features

The main features of the watchdog timer controllers are:

- L4 slave interface support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 11-bit address bus width
  - Burst mode not supported
  - Write nonposted mode supported
- Free-running 32-bit upward counter
- Programmable divider clock source (2^n where n = [0:7])
- On-the-fly read/write register (while counting)
- Subset programming model of the GP timer
- The watchdog timer is reset either on power on or after a warm reset before it starts counting.
- Reset or interrupt actions when a timer overflow condition occurs

- The watchdog timer generates a reset or an interrupt in its hardware integration.

22.4.2 Watchdog Timer Integration

Figure 22-19 shows the integration of the watchdog timers in the device.

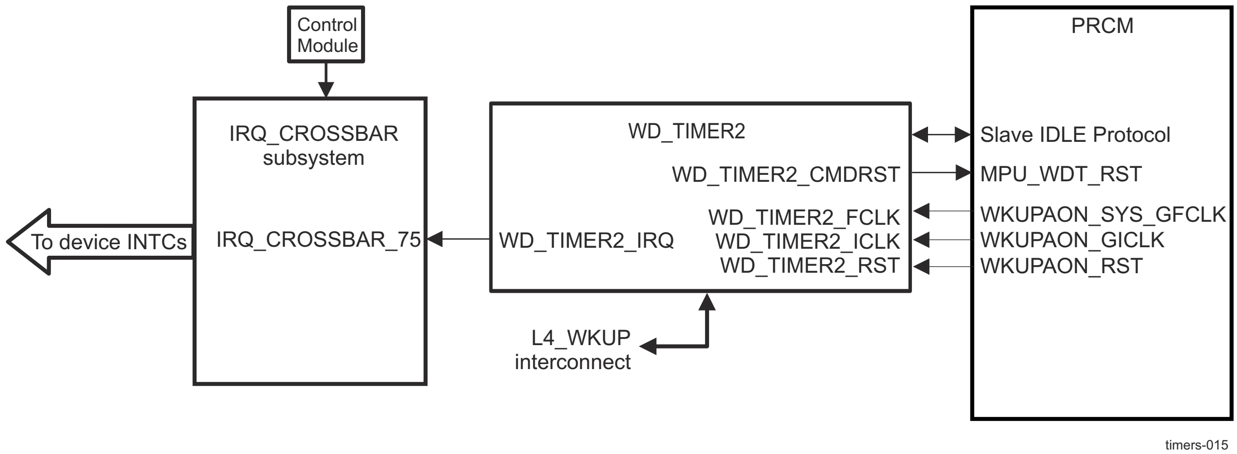


Figure 22-19. Watchdog Timer Integration

Table 22-62 through Table 22-64 summarize the integration of the module in the device.

Table 22-62. Watchdog Timer Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
WD_TIMER2	PD_WKUPAON	Yes	L4_WKUP

Table 22-63. Watchdog Timer Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
WD_TIMER2	WD_TIMER2_FCLK	WKUPAON_SYS_GFCLK	PRCM	WD_TIMER2 functional clock
WD_TIMER2	WD_TIMER2_ICLK	WKUPAON_GICLK	PRCM	WD_TIMER2 interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
WD_TIMER2	WD_TIMER2_RST	WKUPAON_RST	PRM	Reset to WD_TIMER2
	MPU_WDT_RST	WD_TIMER2_CMDRST	WD_TIMER2	Reset to MPU

Note

WD_TIMER2 is reset on power on or after a warm reset before it starts counting.

Table 22-64. Watchdog Timer Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
WD_TIMER2	WD_TIMER2_IRQ	IRQ_CROSSBAR_75	MPU_IRQ_80	WD_TIMER2 interrupt request
No DMA Requests				

Note

The "**Default Mapping**" column in Table 22-64 *Watchdog Timer Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

22.4.3 Watchdog Timer Functional Description

22.4.3.1 Power Management

There are two clock domains in the watchdog timer:

- Functional clock domain: WD_TIMER2_FCLK is the watchdog timer functional clock. It is used to clock the watchdog timer internal logic.
- Interface clock domain: WD_TIMER2_ICLK is the watchdog timer interface clock. It is used to synchronize the watchdog timer L4 port to the L4 interconnect. All accesses from the interconnect are synchronous to WD_TIMER2_ICLK.

Table 22-63 lists the source clocks for the watchdog timer in the device. For more information about clock control and domains, see *Clock Management Functional Description,* in *Power, Reset, and Clock Management*.

From a global system power-management perspective, when the watchdog timer clocks is no longer required, the watchdog timer can be deactivated at the PRCM module level in the corresponding registers.

At the PRCM module level, when the conditions to shut off the PRCM module functional or interface output clocks are met (for more information, see *Clock Domain-Level Clock Management*), the PRCM module automatically launches a hardware handshake protocol to ensure the watchdog timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the watchdog timer.

Although this handshake is a hardware function and out of software control, the way on which the watchdog timer acknowledges the PRCM IDLE request is configurable through the WDSC[4:3] IDLEMODE bit field. Table 22-65 lists the settings and related acknowledgment modes of the IDLEMODE bit field.

Table 22-65. IDLEMODE Settings

IDLEMODE Value	Selected Mode	Description
00	Force-idle	The watchdog timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent loss of data when the clock is switched off.
01	No-idle	The watchdog timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. It is not efficient from a power-saving perspective, however, because it does not allow the PRCM module output clock to be shut off and thus the power domain to be set to a lower power state.
10	Smart-idle	The watchdog timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach for efficient system power management.
11	Smart-idle wakeup-capable mode	The watchdog timer acknowledges the IDLE request, basing its decision on its internal activity. The timer generates (IRQ-request-related) wake-up events when in IDLE state if the WIRQWAKEEN[1:0] bit field is set to 1.

22.4.3.1.1 Wake-Up Capability

If the WDSC[4:3] IDLEMODE bit field sets smart-idle wakeup-capable mode (= 0 × 3), the timer evaluates its internal capability to have the interface clock switched off. When there is no more internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters into sleep mode, ready to issue a wake-up request. This wake-up request is sent only if the WIRQWAKEEN[0] OVF_WK_ENA and/or the WIRQWAKEEN[1] DLY_WK_ENA bits enable the overflow and/or the delay wake-up capability.

22.4.3.2 Interrupts

Table 22-66 list the event flags, and their masks that cause module interrupts.

Table 22-66. Watchdog Timer Events

Event Flag	Event Mask	Mapping	Comments
WIRQSTAT[0] EVENT_OVF	WIRQENSET/WIRQENCLR[0] OVF_IT_ENA	WD_TIMER2_IRQ	Watchdog timer overflow

Table 22-66. Watchdog Timer Events (continued)

Event Flag	Event Mask	Mapping	Comments
WIRQSTAT[1] EVENT_DLY	WIRQENSET/WIRQENCLR[1] DLY_IT_ENA	WD_TIMER2_IRQ	Watchdog delay value reached

22.4.3.3 General Watchdog Timer Operation

The watchdog timer is based on an upward 32-bit counter coupled with a prescaler. The counter overflow is signaled through two independent signals: a simple reset signal and an interrupt signal, both active low. The use of these signals depends on whether they are connected or not. For this information, see Figure 22-18. The interrupt generation mechanism is controlled through the WIRQENSET/WIRQENCLR and WIRQSTAT registers.

The prescaler ratio can be set from 1 to 128 by accessing the WCLR[4:2] PTV bit field and the WCLR[5] PRE bit of the watchdog control register (WCLR).

The current timer value can be accessed on-the-fly by reading the watchdog timer counter register (WCRR), modified by accessing the watchdog timer load register (WLDR) (no on-the-fly update), or reloaded by following a specific reload sequence on the watchdog timer trigger register (WTGR). A start/stop sequence applied to the watchdog timer start/stop register (WSPR) can start and stop the watchdog timers.

Figure 22-20 is a functional block diagram of the watchdog timer.

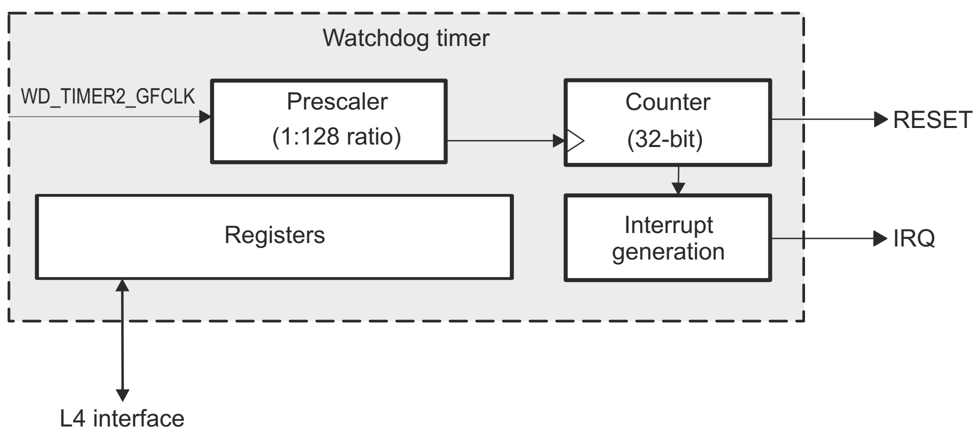


Figure 22-20. 32-Bit Watchdog Timer Functional Block Diagram

22.4.3.4 Reset Context

The watchdog timer is enabled after reset. Table 22-67 lists the default reset values of the two watchdog timer load registers (WLDR) and prescaler ratios (the WCLR[4:2] PTV bit field). To get these values, software must read the corresponding WCLR[4:2] PTV bit field and the 32-bit register to retrieve the static configuration of the module.

Table 22-67. Count and Prescaler Default Reset Values

Timer	WLDR Reset Value	PTV Reset Value
WD_TIMER2	0xFFFB 0000	0

22.4.3.5 Overflow/Reset Generation

When the WCRR overflows, an active-low reset pulse is generated to the PRCM module. This pulse is one prescaled timer clock cycle wide and occurs at the same time as the timer counter overflow.

After reset generation, the counter is automatically reloaded with the value stored in the WLDR and the prescaler is reset (the prescaler ratio remains unchanged). When the reset pulse output is generated, the timer counter begins incrementing again.

Figure 22-21 shows a general functional view of the watchdog timers.

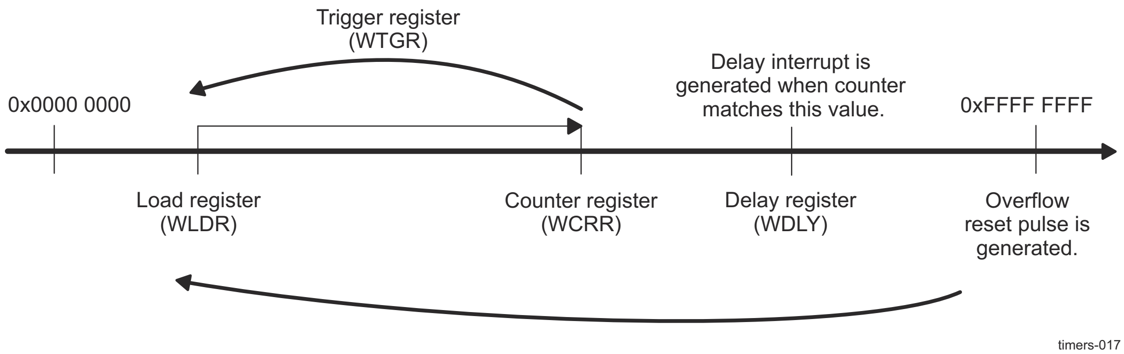


Figure 22-21. Watchdog Timers General Functional View

22.4.3.6 Prescaler Value/Timer Reset Frequency

The watchdog timer is composed of a prescaler stage and a timer counter.

The timer rate is defined by the following values:

- Value of the prescaler fields (the WCLR[5] PRE bit and the WCLR[4:2] PTV bit field)
- Value loaded into WLDR

The prescaler stage is clocked with the timer clock and acts as a clock divider for the timer counter stage. The ratio is managed by accessing the ratio definition field (the WCLR [4:2] PTV bit field) and is enabled with the WCLR[5] PRE bit.

Table 22-68 lists the prescaler clock ratio values.

Table 22-68. Prescaler Clock Ratio Values

WCLR[5] PRE	WCLR[4:2] PTV	Clock Divider (PS)
0	X	1
1	0	1
1	1	2
1	2	4
1	3	8
1	4	16
1	5	32
1	6	64
1	7	128

Thus the watchdog timer overflow rate is expressed as:

OVF_Rate = (0xFFFF FFFF – WLDR + 1) × (timer clock period) × PS

where wd-functional clock period = 1 / (timer clock frequency) and PS = $2^{(PTV)}$

CAUTION

Internal resynchronization causes some latency in any software write to WSPR before WSPR is updated with the programmed value:

1.5 × functional clock cycles ≤ write_WD_TIMER_WSPR_latency ≤ 2.5 × functional clock cycles

Remember to consider this latency whenever the watchdog timer must be started or stopped.

For example, for a timer clock input of 32 kHz with a prescaler ratio value of 0x1 (clock divided by 2) and WCLR[5] PRE = 1 (clock divider enabled), the reset period is as listed in Table 22-69.

Table 22-69. Reset Period Examples

WLDR Value	Reset Period
0x0000 0000	74 h 56 min
0xFFFF 0000	4 s
0xFFFF FFF0	1 ms
0xFFFF FFFF	62.5 µs

CAUTION

- Ensure that the reloaded value allows the correct operation of the application. When a watchdog timer is enabled, software must periodically trigger a reload before the counter overflows. Hence, the value of the WLDR[31:0] bit field must be chosen according to the ongoing activity preceding the watchdog reload.
- Due to design reasons, WLDR[31:0] = 0xFFFF FFFF is a special case, although such value of WLDR is meaningless. When WLDR is programmed with the overflow value, a triggering event generates a reset/interrupt one functional clock cycle later, even if the watchdog timer is stopped.

Table 22-70 lists the default reset periods for the watchdog timer.

Table 22-70. Default Watchdog Timer Reset Periods

Watchdog Timers	Clock Source	Default Reset Period
WD_TIMER2	32 kHz	10 s

22.4.3.7 Triggering a Timer Reload

To reload the timer counter and reset the prescaler before reaching overflow, a reload command is executed by accessing the WTGR using a specific reload sequence.

The specific reload sequence is performed whenever the written value on the WTGR differs from its previous value. In this case, reload is executed in the same way as an overflow autoreload, but without the generation of a reset pulse.

The timer counter is loaded with the value of the watchdog timer load register (the WLDR[31:0] TIMER_LOAD bit field), and the prescaler is reset.

22.4.3.8 Start/Stop Sequence for Watchdog Timer (Using the WSPR Register)

To start and stop a watchdog timer, access must be made through the WSPR using a specific sequence.

To disable the timer, follow this sequence:

1. Write 0xXXXX AAAA in the WSPR.
2. Write 0xXXXX 5555 in the WSPR.

To enable the timer, follow this sequence:

1. Write 0xXXXX BBBB in the WSPR.
2. Write 0xXXXX 4444 in the WSPR.

All other write sequences on the WSPR have no effect on the start/stop feature of the module.

22.4.3.9 Modifying Timer Count/Load Values and Prescaler Setting

To modify the timer counter value (the WCRR), the prescaler ratio (the WCLR[4:2] PTV bit field), delay configuration value (the WDLY[31:0] DLY_VALUE bit field), or the load value (the WLDR[31:0] TIMER_LOAD bit field), the watchdog timer must be disabled by using the start/stop sequence (the WSPR).

After a write access, the load register value and prescaler ratio registers are updated immediately, but new values are considered only after the next consecutive counter overflow or after a new trigger command (the WTGR).

22.4.3.10 Watchdog Counter Register Access Restriction (WCRR)

A 32-bit shadow register is implemented to read a coherent value of the WCRR because the WCRR is directly related to the timer counter value and is updated on the timer clock (WD_TIMER_FCLK). The shadow register is updated by a 16-bit LSB read command.

Note

Although the L4 clock (WD_TIMER_ICLK) is completely asynchronous with the timer clock (WD_TIMER_FCLK), some synchronization is performed to ensure that the value of the WCRR is not read while it is being incremented.

When 32-bit read access is performed, the shadow register is not updated. Read access is performed directly from the accessed register.

To ensure that a coherent value is read inside the WCRR, the first read access is to the lower 16 bits (offset = 0x08), followed by read access to the upper 16 bits (offset = 0x0A).

22.4.3.11 Watchdog Timer Interrupt Generation

When an interrupt source occurs, the interrupt status bit (the WISR[0] OVF_IT_FLAG or WISR[1] DLY_IT_FLAG bit) is set to 1. The output interrupt line (WD_TIMER_IRQ) is asserted (active high) when status (the EVENT_xxx bit) and enable (the xxx_IT_ENA bit) flags are set to 1; the order is not relevant. Writing 1 to the enable bit (the status is already set at 1) also triggers the interrupt in the normal order (enable first, status next). The pending interrupt event is cleared when the set status bit is overwritten by a value of 1 by a write command in the WISR. Reading the WISR and writing the value back allows a fast interrupt acknowledge process.

The watchdog timer issues an overflow interrupt if this interrupt is enabled in the watchdog interrupt-enable register (WIER[0] OVF_IT_ENA = 1). When the overflow occurs, the interrupt status bit (the WISR[0] OVF_IT_FLAG bit) is set to 1. The output interrupt line (WD_TIMER_IRQ) is asserted (active high) when status (OVF_IT_FLAG) and enable (OVF_IT_ENA) flags are set to 1; the order is not relevant. This interrupt can be disabled by setting the WIER [0] OVF_IT_ENA bit to 1.

The watchdog can issue the delay interrupt if this interrupt is enabled in the interrupt-enable register (WIER[1] DLY_IT_ENA = 1). When the counter is running and the counter value matches the value stored in the delay configuration register (WDLY), the corresponding interrupt status bit is set in the watchdog status register (WISR) and the output interrupt line is asserted (active high) when the flag (DLY_IT_FLAG) and enable (DLY_IT_ENA) bits are set to 1 in the WISR and WIER registers, respectively; the order (normally enable, then flag) is not relevant. This interrupt can be disabled by setting the WIER[1] DLY_IT_ENA bit to 1.

Note

Writing 0 to the WISR[0] OVF_IT_FLAG or WISR[1] DLY_IT_FLAG bit has no effect.

The two clock domains are resynchronized because the interrupt event is generated on the functional clock domain (WD_TIMERi_FCLK) during the updating of the interrupt status register.

The WDLY register is used to specify the value of the delay configuration register. The delay time to interrupt is the difference between the reload value stored in the counter load register (WLDR) and the programmed value in this register (WDLY).

Use the following formula to estimate the delay time:

Delay time period = (WDLY – WLDR + 1) × Timer clock period × Clock divider

Where:

- Timer clock period = 1 / (Timer clock frequency)
- Clock divider = 2PTV

If the counter value (WCRR) reaches the programmed value (WDLY), the status bit (EVENT_DLY) is set in the interrupt status register (WIRQSTAT), and an interrupt occurs if the corresponding enable bit is set in the interrupt enable register (WIRQENSET).

CAUTION

If the reload event occurs (after a triggering sequence or after a reset sequence) before reaching the programmed value (the WDLY[31:0] WDLY_VALUE bit field), no interrupt is generated.

Also, no interrupt is generated if the value programmed in the WDLY register is less than the value stored in the WLDR.

22.4.3.12 Watchdog Timer Under Emulation

During emulation mode, the watchdog timer can or cannot continue to run, according to the value of the WDSC[5] EMUFREE bit of the system configuration register (WDSC).

- When EMUFREE is 1, watchdog timer execution is not stopped and a reset pulse is still generated when overflow is reached.
- When EMUFREE is 0, the counters (prescaler/timer) are frozen and incrementation restarts after exiting emulation mode.

22.4.3.13 Accessing Watchdog Timer Registers

Posted/nonposted selection applies only to functional registers that require synchronization on/from the timer functional clock domain (WD_TIMER2_FCLK). For write/read operation, the following registers are affected:

- WCLR
- WCRR
- WLDR
- WTGR
- WDLY
- WSPR

The timer interface clock domain synchronous registers are not affected by the posted/nonposted selection; the write/read operation is effective and acknowledged (command accepted) after one WD_TIMER_ICLK cycle from the command assertion. The timer interface clock domain synchronous registers are:

- WIDR
- WDSC
- WDST
- WISR
- WIER
- WWER
- WWPS

Note

Accesses to WD_TIMER2 is posted.

22.4.4 Watchdog Timer Low-Level Programming Model

This section covers the low-level hardware programming sequences for the configuration and use of the module.

22.4.4.1 Global Initialization

22.4.4.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the watchdog timer is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the watchdog timer (see Table 22-71). For more information, see Section 22.4.2, *Watchdog Timer Integration.*

Table 22-71. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. For more information about the module configuration, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing must be set in the control module. For more information about the module configuration, see *Pad Configuration Registers* in *Control Module*.
IRQ_CROSSBAR	The IRQ_CROSSBAR configuration must be performed to enable the interrupts from the watchdog timer. See *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

22.4.4.1.2 Watchdog Timer Module Global Initialization

22.4.4.1.2.1 Main Sequence – Watchdog Timer Module Global Initialization

Table 22-72 lists the steps for initializing the watchdog timer module when the module is to be used for the first time.

Table 22-72. Watchdog Timer Module Global Initialization

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	WDSC[1] SOFTRESET	0x1
Wait until reset release?	WDSC[1] SOFTRESET	0x0
Configure idle mode.	WDSC[4:3] IDLEMODE	0x-
Enable delay wakeup.	WIRQWAKEEN[1] DLY_WK_ENA	0x1
Enable overflow wakeup.	WIRQWAKEEN[0] OVF_WK_ENA	0x1
Enable delay interrupt.	WIRQENSET[1] ENABLE_DLY	0x1
Enable overflow interrupt.	WIRQENSET[0] ENABLE_OVF	0x1

22.4.4.2 Operational Mode Configuration

22.4.4.2.1 Watchdog Timer Basic Configuration

22.4.4.2.1.1 Main Sequence – Watchdog Timer Basic Configuration

Table 22-73 lists the steps for the basic configuration of the watchdog timer.

Table 22-73. Watchdog Timer Basic Configuration

Step	Register/Bit Field/Programming Model	Value
Disable the watchdog timer.	See Section 22.4.4.2.1.2.	
Set prescaler value.	WCLR[4:2] PTV	0x-
Enable prescaler.	WCLR[5] PRE	0x1
Load delay configuration value.	WDLY	0x-
Load timer counter value.	WCRR	0x-
Enable the watchdog timer.	See Section 22.4.4.2.1.3.	

22.4.4.2.1.2 Subsequence – Disable the Watchdog Timer

Table 22-74 lists the steps to disable the watchdog timer.

Table 22-74. Disable the Watchdog Timer

Step	Register/Bit Field/Programming Model	Value
Write disable sequence Data1.	WSPR	0xXXXX AAAA
Write disable sequence Data2.	WSPR	0xXXXX 5555

22.4.4.2.1.3 Subsequence – Enable the Watchdog Timer

Table 22-75 lists the steps to enable the watchdog timer.

Table 22-75. Enable the Watchdog Timer

Step	Register/Bit Field/Programming Model	Value
Write enable sequence Data1.	WSPR	0xXXXX BBBB
Write enable sequence Data2.	WSPR	0xXXXX 4444

22.4.5 Watchdog Timer Register Manual

22.4.5.1 Watchdog Timer Instance Summary

Table 22-76 lists the base address and address space for the watchdog timer module instances.

Table 22-76. Watchdog Timer Instance Summary

Module Name	Module Base Address L4_WKUP Interconnect	Size
WD_TIMER2	0x4AE1 4000	104 Bytes

Note

Private access is access that does not use the L4 interconnect.

22.4.5.2 Watchdog Timer Registers

22.4.5.2.1 Watchdog Timer Register Summary

CAUTION

The watchdog timers registers are limited to 32- and 16-bit data accesses; 8-bit access is not allowed and can corrupt register content.

Table 22-77 lists the WD_TIMER2 registers.

Table 22-77. WD_TIMER2 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WD_TIMER2 Physical Address L4_WKUP Interconnect
WIDR	R	32	0x0000 0000	0x4AE1 4000
WDSC	RW	32	0x0000 0010	0x4AE1 4010
WDST	R	32	0x0000 0014	0x4AE1 4014
WISR	RW	32	0x0000 0018	0x4AE1 4018
WIER	RW	32	0x0000 001C	0x4AE1 401C
WWER	RW	32	0x0000 0020	0x4AE1 4020
WCLR	RW	32	0x0000 0024	0x4AE1 4024
WCRR	RW	32	0x0000 0028	0x4AE1 4028
WLDR	RW	32	0x0000 002C	0x4AE1 402C
WTGR	RW	32	0x0000 0030	0x4AE1 4030
WWPS	R	32	0x0000 0034	0x4AE1 4034
WDLY	RW	32	0x0000 0044	0x4AE1 4044
WSPR	RW	32	0x0000 0048	0x4AE1 4048
WIRQEOI	RW	32	0x0000 0050	0x4AE1 4050
WIRQSTATRAW	RW	32	0x0000 0054	0x4AE1 4054
WIRQSTAT	RW	32	0x0000 0058	0x4AE1 4058
WIRQENSET	RW	32	0x0000 005C	0x4AE1 405C
WIRQENCLR	RW	32	0x0000 0060	0x4AE1 4060
WIRQWAKEEN	RW	32	0x0000 0064	0x4AE1 4064

22.4.5.2.2

Note

- The WISR and WIRQSTATRAW registers have the same functionality. The WISR register is used for software backward compatibility.
- The WIER and WIRQENSET/WIRQENCLR registers have the same functionality. The WIER register is used for software backward compatibility.
- The WWER and WIRQWAKEEN registers have the same functionality. The WWER is used for software backward compatibility.
- The WIRQSTATRAW and WIRQSTAT registers give the same information when read. The WIRQSTATRAW register is used for debug.

22.4.5.2.3 Watchdog Timer Register Description

Table 22-78 through Table 22-96 describe the watchdog timer registers.

Table 22-78. WIDR

Address Offset	0x0000 0000		
Physical Address	0x4AE1 4000	**Instance**	WD_TIMER2
Description	IP revision identifier		
Type	R		

31–0
REV

Bits	Field Name	Description	Type	Reset
31:0	REV	IP Revision	R	0x– [(1)]

(1) TI internal data.

Table 22-79. WDSC

Address Offset	0x0000 0010		
Physical Address	0x4AE1 4010	**Instance**	WD_TIMER2
Description	This register controls the various parameters of the L4 interface.		
Type	RW		

31–6	5	4–3	2	1	0
RESERVED	EMUFREE	IDLEMODE	RESERVED	SOFTRESET	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000000
5	EMUFREE	Emulation mode 0x0: Timer counter frozen in emulation 0x1: Timer counter free-running in emulation	RW	0
4:3	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target IDLE state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the IP module internal requirements. Backup mode, for debug only. 0x1: No-idle mode: local target never enters IDLE state. Backup mode, for debug only. 0x2: Smart-idle mode: local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module should not generate (IRQ- or DMA-request-related) wake-up events. 0x3: Smart-idle wake-up-capable mode: local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. Mode is relevant only if the appropriate IP module *swake-up* output(s) is (are) implemented.	RW	0x2
2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0

Bits	Field Name	Description	Type	Reset
1	SOFTRESET	Software reset. (Optional) Read 0x0: Reset done, no pending action Write 0x0: No action Write 0x1: Initiate software reset. Read 0x1: Reset (software or other) ongoing	RW	0
0	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0

Table 22-80. WDST

Address Offset	0x0000 0014		
Physical Address	0x4AE1 4014	**Instance**	WD_TIMER2
Description	This register provides status information about the module.		
Type	R		

31:1	0
RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0.	R	0x0000 0000
0	RESETDONE	Internal module reset monitoring Read 0x0: Internal module reset is ongoing. Read 0x1: Reset completed	R	1

Table 22-81. WISR

Address Offset	0x0000 0018		
Physical Address	0x4AE1 4018	**Instance**	WD_TIMER2
Description	This register shows which interrupt events are pending inside the module.		
Type	RW		

31:2	1	0
RESERVED	DLY_IT_FLAG	OVF_IT_FLAG

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0.	R	0x0000 0000
1	DLY_IT_FLAG	Pending delay interrupt status. Read 0x0: No delay interrupt pending Write 0x0: Status unchanged Write 0x1: Status bit cleared Read 0x1: Delay interrupt pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	OVF_IT_FLAG	Pending overflow interrupt status. Read 0x0: No overflow interrupt pending Write 0x0: Status unchanged Write 0x1: Status bit cleared Read 0x1: Overflow interrupt pending	RW W1toClr	0

Table 22-82. WIER

Address Offset	0x0000 001C		
Physical Address	0x4AE1 401C	**Instance**	WD_TIMER2
Description	This register controls (enable/disable) the interrupt events.		
Type	RW		

31:2	1	0
RESERVED	DLY_IT_ENA	OVF_IT_ENA

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0.	R	0x0000 0000
1	DLY_IT_ENA	Delay interrupt enable/disable 0x0: Disable delay interrupt. 0x1: Enable delay interrupt.	RW	0
0	OVF_IT_ENA	Overflow interrupt enable/disable 0x0: Disable overflow interrupt. 0x1: Enable overflow interrupt.	RW	0

Table 22-83. WWER

Address Offset	0x0000 0020		
Physical Address	0x4AE1 4020	**Instance**	WD_TIMER2
Description	This register controls (enable/disable) the wake-up events.		
Type	RW		

31:2	1	0
RESERVED	DLY_WK_ENA	OVF_WK_ENA

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	DLY_WK_ENA	Delay wake-up enable 0x0: Disable delay wakeup. 0x1: Enable delay wakeup.	RW	0

Bits	Field Name	Description	Type	Reset
0	OVF_WK_ENA	Overflow wake-up enable 0x0: Disable overflow wakeup. 0x1: Enable overflow wakeup.	RW	0

Table 22-84. WCLR

Address Offset	0x0000 0024		
Physical Address	0x4AE1 4024	**Instance**	WD_TIMER2
Description	This register controls the prescaler stage of the counter.		
Type	RW		

31–6	5	4–2	1–0
RESERVED	PRE	PTV	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reads return 0.	R	0x0000000
5	PRE	Prescaler enable/disable configuration 0x0: Prescaler disabled 0x1: Prescaler enabled	RW	1
4:2	PTV	Prescaler value The timer counter is prescaled with the value: 2^{PTV}. Example: PTV = 3 -> counter increases value if started after 8 functional clock periods. On reset, it is loaded from PI_PTV_RESET_VALUE input port.	RW	0x0
1:0	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0

Table 22-85. WCRR

Address Offset	0x0000 0028		
Physical Address	0x4AE1 4028	**Instance**	WD_TIMER2
Description	This register holds the value of the internal counter.		
Type	RW		

31–0
TIMER_COUNTER

Bits	Field Name	Description	Type	Reset
31:0	TIMER_COUNTER	Value of the timer counter register	RW	0x0000 0000

Table 22-86. WLDR

Address Offset	0x0000 002C		
Physical Address	0x4AE1 402C	**Instance**	WD_TIMER2
Description	This register holds the timer load value.		
Type	RW		

31–0
TIMER_LOAD

Bits	Field Name	Description	Type	Reset
31:0	TIMER_LOAD	Value of the timer load register	RW	0x0000 0000

Table 22-87. WTGR

Address Offset	0x0000 0030		
Physical Address	0x4AE1 4030	**Instance**	WD_TIMER2
Description	Writing a different value than the one already written in this register does a watchdog counter reload.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTGR_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	TTGR_VALUE	Value of the trigger register	RW	0x0000 0000

Table 22-88. WWPS

Address Offset	0x0000 0034		
Physical Address	0x4AE1 4034	**Instance**	WD_TIMER2
Description	This register contains the write posting bits for all writeable functional registers.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										W_PEND_WDLY	W_PEND_WSPR	W_PEND_WTGR	W_PEND_WLDR	W_PEND_WCRR	W_PEND_WCLR

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000000
5	W_PEND_WDLY	Write pending for register WDLY Read 0x0: No register write pending Read 0x1: Register write pending	R	0
4	W_PEND_WSPR	Write pending for register WSPR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
3	W_PEND_WTGR	Write pending for register WTGR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
2	W_PEND_WLDR	Write pending for register WLDR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
1	W_PEND_WCRR	Write pending for register WCRR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
0	W_PEND_WCLR	Write pending for register WCLR Read 0x0: No register write pending Read 0x1: Register write pending	R	0

Table 22-89. WDLY

Address Offset	0x0000 0044		
Physical Address	0x4AE1 4044	**Instance**	WD_TIMER2

Table 22-89. WDLY (continued)

Description	This register holds the delay value that controls the internal pre-overflow event detection.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDLY_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	WDLY_VALUE	Value of the delay register	RW	0x0000 0000

Table 22-90. WSPR

Address Offset	0x0000 0048		
Physical Address	0x4AE1 4048	**Instance**	WD_TIMER2
Description	This register holds the start-stop value that controls the internal start-stop FSM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WSPR_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	WSPR_VALUE	Value of the start-stop register	RW	0x0000 0000

Table 22-91. WIRQEOI

Address Offset	0x0000 0050		
Physical Address	0x4AE1 4050	**Instance**	WD_TIMER2
Description	Software End Of Interrupt		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0.	R	0x0
0	LINE_NUMBER	EOI for interrupt output line Reads always 0 (no EOI memory)	RW	0x0

Table 22-92. WIRQSTATRAW

Address Offset	0x0000 0054		
Physical Address	0x4AE1 4054	**Instance**	WD_TIMER2
Description	IRQ unmasked status, status set per-event raw interrupt status vector, line 0. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	EVENT_DLY	EVENT_OVF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	EVENT_DLY	Settable raw status for delay event Read 0x0: No event pending Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending	RW W1toSet	0
0	EVENT_OVF	Settable raw status for overflow event Read 0x0: No event pending Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending	RW W1toSet	0

Table 22-93. WIRQSTAT

Address Offset	0x0000 0058		
Physical Address	0x4AE1 4058	**Instance**	WD_TIMER2
Description	IRQ masked status, status clear per-event enabled interrupt status vector, line 0. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EVENT_DLY	EVENT_OVF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	EVENT_DLY	Clearable, enabled status for delay event Read 0x0: No (enabled) event pending Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending	RW W1toClr	0
0	EVENT_OVF	Clearable, enabled status for overflow event Read 0x0: No (enabled) event pending Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending	RW W1toClr	0

Table 22-94. WIRQENSET

Address Offset	0x0000 005C		
Physical Address	0x4AE1 405C	**Instance**	WD_TIMER2
Description	IRQ enable set per-event interrupt enable bit vector, line 0. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		

Table 22-94. WIRQENSET (continued)

Type	RW

31–2	1	0
RESERVED	ENABLE_DLY	ENABLE_OVF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	ENABLE_DLY	Enable for delay event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Enable interrupt. Read 0x1: Interrupt enabled	RW W1toSet	0
0	ENABLE_OVF	Enable for overflow event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Enable interrupt. Read 0x1: Interrupt enabled	RW W1toSet	0

Table 22-95. WIRQENCLR

Address Offset	0x0000 0060		
Physical Address	0x4AE1 4060	**Instance**	WD_TIMER2
Description	IRQ enable clear per-event interrupt enable bit vector, line 0. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
Type	RW		

31–2	1	0
RESERVED	ENABLE_DLY	ENABLE_OVF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	ENABLE_DLY	Enable for delay event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Disable interrupt. Read 0x1: Interrupt enabled	RW W1toClr	0
0	ENABLE_OVF	Enable for overflow event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Disable interrupt. Read 0x1: Interrupt enabled	RW W1toClr	0

Table 22-96. WIRQWAKEEN

Address Offset	0x0000 0064		
Physical Address	0x4AE1 4064	**Instance**	WD_TIMER2

Table 22-96. WIRQWAKEEN (continued)

Description	This register controls (enable/disable) the wake-up events.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DLY_WK_ENA	OVF_WK_ENA

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	DLY_WK_ENA	Enable delay wake-up 0x0: Disable delay wakeup 0x1: Enable delay wakeup	RW	0
0	OVF_WK_ENA	Enable overflow wakeup 0x0: Disable overflow wakeup 0x1: Enable overflow wakeup	RW	0

Chapter 23

Real-Time Clock (RTC)

This chapter provides a functional presentation of the Real-Time Clock (RTC) module.

23.1 RTC Overview 5280
23.2 RTC Environment 5282
23.3 RTC Integration 5283
23.4 RTC Functional Description 5285
23.5 RTC Low-Level Programming Guide 5293
23.6 RTC Register Manual 5294

23.1 RTC Overview

The real-time clock is a precise timer which can generate interrupts on intervals specified by the user. Interrupts can occur every second, minute, hour, or day. The clock can track the passage of real time for durations of several years, provided it has a sufficient power source the entire time.

The basic purpose of the RTC is to keep time of day. The other equally important purpose of the RTC is for Digital Rights management. Some degree of tamper proofing is needed to ensure that simply stopping, resetting, or corrupting the RTC does not go unnoticed so that if this occurs, the application can re-acquire the time of day from a trusted source. Another purpose of the RTC is to wake up the rest of the chip from a power-down state.

Alarms are available to interrupt the host processor at a particular time, or at periodic intervals, such as once per minute or once per day. In addition, the RTC can interrupt the CPU every time the calendar and time registers are updated, or at programmable periodic intervals.

Figure 23-1 shows the RTC functional block diagram.

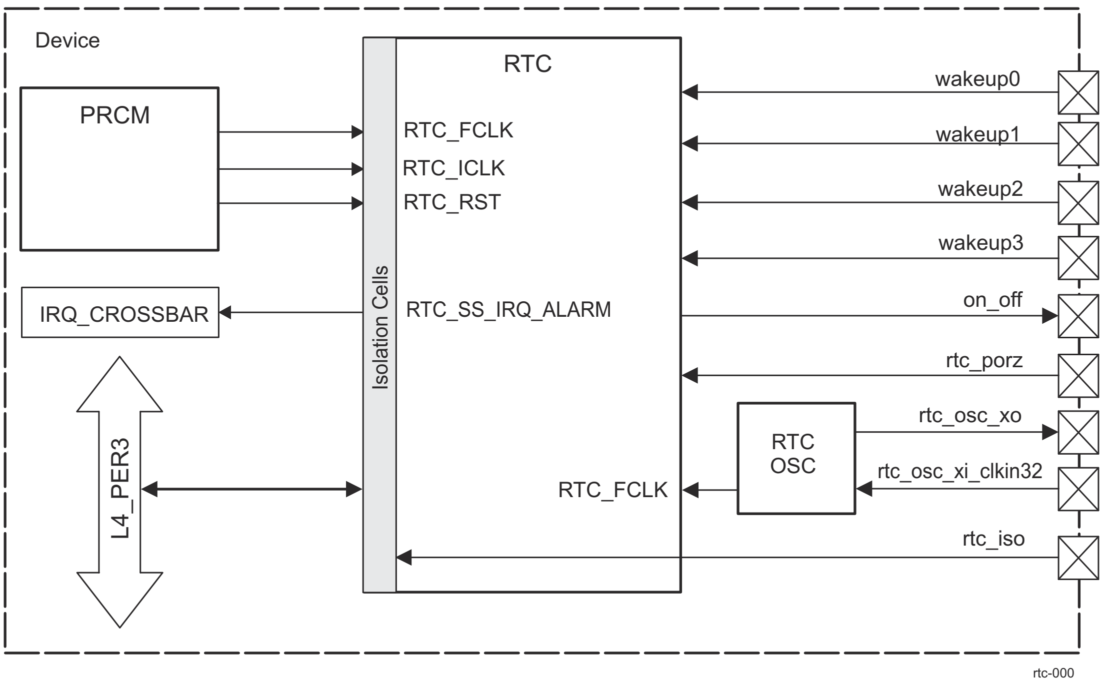


Figure 23-1. RTC Block Diagram

23.1.1 RTC Features

The main features of the RTC are:

- 100-year calendar
- Counts seconds, minutes, hours, day of the week, date, month, and year with leap year compensation
- Binary-coded decimal (BCD) representation of time, calendar and alarm
- Alarm interrupt
- Periodic interrupt
- Single interrupt to the host processor
- External 32-kHz clock
- Isolated RTC voltage domain which remains powered during standby (RTC) mode while the rest of chip supplies are turned off
- RTC voltage domain has 3 × 32-bit read/write scratch registers retained in RTC modes
- RTC supports four wakeup input pins:
  - Each input has a filter to detect high or low transition with pulse duration.

  - Each pin has the following controls retained in standby mode and modified and read by control module in ON mode.
    - Enable bit (decides if the pin causes wakeup or not)
    - Edge sensitivity (rising, falling, or both causing wakeup)
    - Wake-up flag (sticky bits indicating wake-up source)
  - Allow paths for optional input to GPIO if not used for wakeup.
- PMIC enable output pin to indicate external switched MAIN domain supplies to turn on or off based on wakeup state

23.2 RTC Environment

23.2.1 RTC External Interface

The RTC is connected to four external system wake-up signals. The RTC also can operate on external 32-kHz clock and has dedicated power-on reset pin. An external control of the RTC isolation cells is available to isolate RTC from other power domains when the PD_RTC is the only powered domain.

The RTC module can be configured to produce a power-up/down control signal for the power-management chip (PMIC, if present).

Table 23-1 summarizes the external signals received and produced by the RTC module.

Table 23-1. RTC External Signals

Device-Level Pin	Module-Level Signal	I/O	Description
rtc_osc_xi_clkin32	RTC_32K_CLK	I	RTC clock input
rtc_osc_xo	—(1)	O	32k oscillator crystal driver output. Unused if clock is provided from external source (crystal not connected)
wakeup0	EXT_WAKEUP0	I	External wake-up signal
wakeup1	EXT_WAKEUP1	I	External wake-up signal
wakeup2	EXT_WAKEUP2	I	External wake-up signal
wakeup3	EXT_WAKEUP3	I	External wake-up signal
on_off	PMIC_PWR_ENABLE	O	Companion PMIC control output (on/off) (optional)
rtc_porz	RTC_POR_ARST	I	External power-on-reset input (optional)
rtc_iso	—(2)	I	External control of isolation cells. This signal must be kept low (0) during RTC mode and high (1) after the device power supplies ramped-up. This can typically be achieved by connecting rtc_iso to the porz (not rtc_porz) with appropriate voltage level translation if necessary.

(1) Not a RTC module's signal. Crystal driver output from oscillator.
(2) Not a RTC module's signal. Controls isolation cells.

Figure 23-2 shows the described signals.

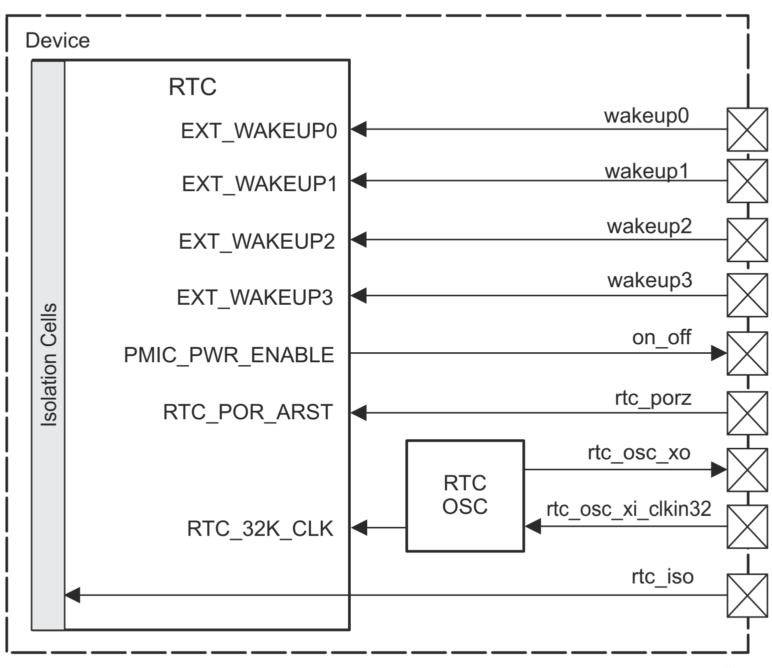


Figure 23-2. RTC External Signals

23.3 RTC Integration

Figure 23-3 shows the integration of the Real-time clock subsystem inside the device.

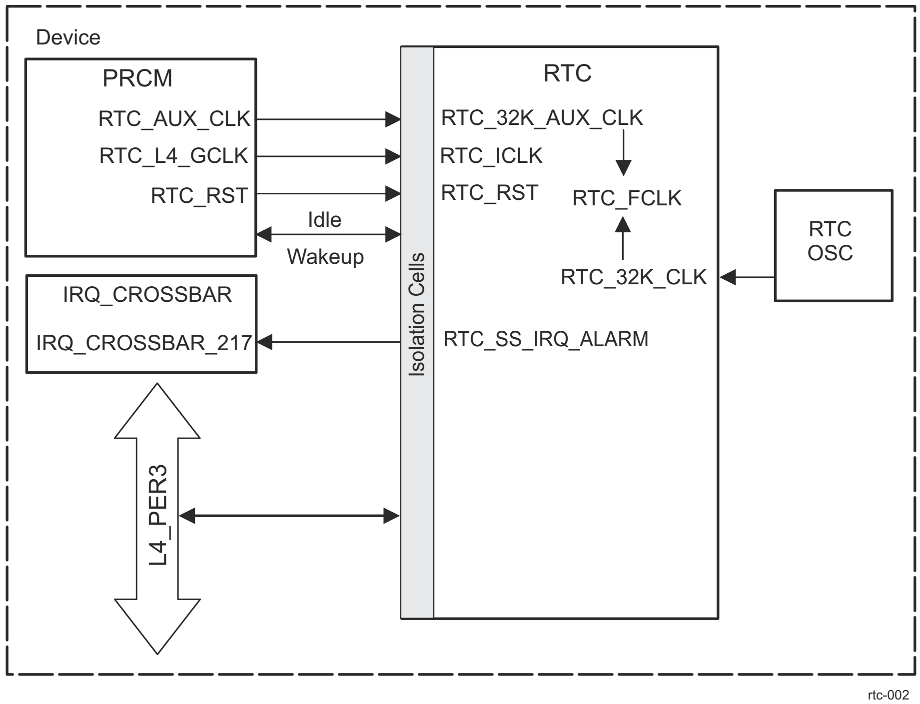


Figure 23-3. RTC Module Integration

Table 23-2. RTC Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
RTC	PD_RTC	Yes	L4_PER3

Table 23-3. RTC Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
RTC	RTC_32K_AUX_CLK	RTC_AUX_CLK	PRCM	RTC functional clock from PRCM
	RTC_32K_CLK	rtc_osc_xi	32k RTC Osc	RTC functional clock from oscillator/pin
	RTC_ICLK	RTC_L4_GICLK	PRCM	RTC interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
RTC	RTC_RST	RTC_RST	PRCM	Real-time clock subsystem PRCM reset signal

Table 23-4. RTC Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR**	**Default Mapping**	**Description**
RTC	RTC_SS_IRQ_ALARM	IRQ_CROSSBAR_217	-	Alarm interrupt going to device IRQ_CROSSBAR.
No DMA Requests				

Note

The **Default Mapping** column in Table 23-4, *RTC Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other inputs of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

23.4 RTC Functional Description

This section defines the module interrupt capabilities and requirements.

Figure 23-4 shows the functional block diagram of the RTC module.

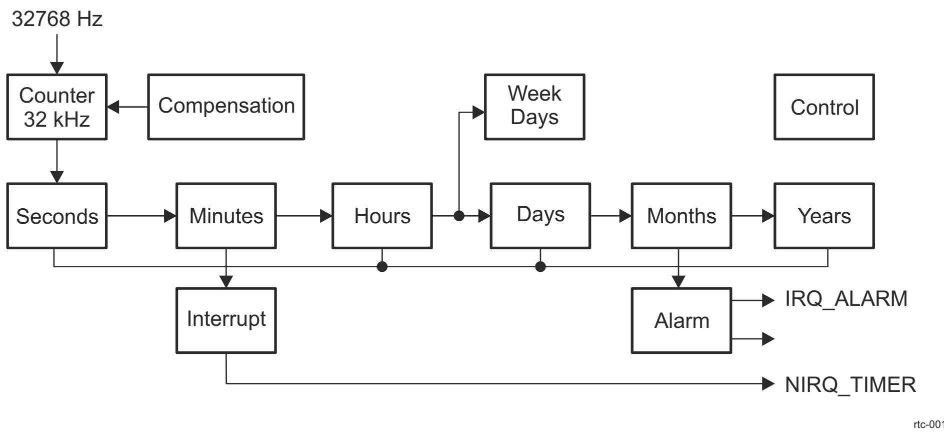


Figure 23-4. RTC Functional Block Diagram

23.4.1 Clock Source

The clock reference for the RTC is an internal 32-kHz clock signal (RTC_AUX_CLK from PRCM) or an external clock source of the 32.768 Hz frequency on the RTC_32K_CLK pin.

The RTC_DISABLE bit in the control register (RTC_CTRL_REG) can be set to save power; however, the RTC_DISABLE bit is not to be cleared once it has been set. If the application requires the RTC module to stop and continue, the STOP_RTC bit in RTC_CTRL_REG is used instead.

The selection between external and internal clock is done through the RTC_OSC_REG register. Setting the RTC_OSC_REG[3] K32CLK_SEL = 1 selects the 32-kHz external/oscillator clock (RTC_32K_CLK), setting it to 0 selects the PRCM clock (RTC_AUX_CLK). RTC_OSC_REG[2] RES_SELECT enables the internal feedback resistor. By modifying the RTC_OSC_REG[4] OSC32K_GZ bit, software enables or disables the 32-kHz oscillator.

If the RTC module or RTC oscillator is not used, the rtc_osc_xi_clkin32 device pin should be held low and rtc_osc_xo should be left unconnected. For more information about rtc pin connection if RTC is not used in the design, see the device-specific data manual.

23.4.2 Interrupt Support

23.4.2.1 CPU Interrupts

The RTC generates two interrupt outputs:

- timer_intr is a timer interrupt.
- alarm_intr is an alarm interrupt.

These two interrupts are OR-ed inside the RTC subsystem and a single interrupt (RTC_SS_IRQ_ALARM) is passed to the IRQ_CROSSBAR module.

Note

Both interrupt outputs support high-level and high-pulse.

23.4.2.2 Interrupt Description

23.4.2.2.1 Timer Interrupt (timer_intr)

The timer interrupt can be generated periodically: every second, every minute, every hour, or every day (see RTC_INTERRUPTS_REG[1:0] for a description of how to set this up). The IT_TIMER bit of RTC_INTERRUPTS_REG enables this interrupt. The timer interrupt is active-low.

The RTC_STATUS_REG bits are only updated at each new interrupt and occur according to Table 23-5. For example, bit 2 (SEC) is set every time one second has passed. Bit 2 is also set when 1 minute has passed, because the completion of 1 minute also marks the completion of 1 second (from 59 seconds to 60 seconds). The same holds true for hours and days: each of them also corresponds to the passing of a second.

Conversely, bit 5 (DAY) is always set when a day has passed. It might also be set when an hour, minute, or second has passed. However, this only occurs when the elapsed hour, minute, or second corresponds to the start of a new day.

Table 23-5. Interrupt Trigger Events

STATUS_REG Bit	One Day Has Passed	One Hour Has Passed	One Minute Has Passed	One Second Has Passed
[5] (DAY)	1	0/1[(1)]	0/1[(1)]	0/1[(1)]
[4] (HOUR)	1	1	0/1[(1)]	0/1[(1)]
[3] (MIN)	1	1	1	0/1[(1)]
[2] (SEC)	1	1	1	1

(1) This event is triggered only when the elapsed time unit (for example, Day) corresponds to the passage of another unit (for example, Seconds). For example, when the clock ticks from 00:23:59:59 (days : hours : minutes : seconds) to 01:00:00:00.

23.4.2.2.2 Alarm Interrupt (alarm_intr)

The alarm interrupt can be generated when the time set into the Timer/Calendar (TC) ALARM registers (see Section 23.4.3.2) is exactly the same as in the TC registers. This interrupt is then generated, if the IT_ALARM bit in the interrupt register (RTC_INTERRUPTS_REG) is set. This interrupt is low-level sensitive. The RTC_STATUS_REG[6] ALARM bit indicates that alarm interrupt has occurred. This interrupt is disabled by setting the RTC_STATUS_REG[6] ALARM bit to 1.

To set up an alarm:

- Modify the RTC_ALARM_SECONDS_REG, RTC_ALARM_MINUTES_REG, RTC_ALARM_HOURS_REG, RTC_ALARM_DAYS_REG, RTC_ALARM_MONTHS_REG, and RTC_ALARM_YEARS_REG registers to the exact time an alarm is to be generated.
- Set the IT_ALARM bit in RTC_INTERRUPTS_REG to enable the alarm interrupt.

When the RTC_PMIC_REG[3:0] EXT_WAKEUP_EN and RTC_PMIC_REG[16] PWR_ENABLE_EN bits are set to 0x1, the PMIC_PWR_ENABLE is controlled by EXT_WAKEUP[3:0] ALARM and ALARM2.

The ALARM2 is set by writing the exact time and date to generate an alarm in RTC_ALARM2_SECONDS_REG, RTC_ALARM2_MINUTES_REG, RTC_ALARM2_HOURS_REG, RTC_ALARM2_DAYS_REG, RTC_ALARM2_MONTHS_REG, and RTC_ALARM2_YEARS_REG registers.

23.4.3 RTC Programming/Usage Guide

23.4.3.1 Time/Calendar Data Format

The time and calendar data in the RTC is stored in BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. Four bits are assigned to each BCD digit in most time/calendar registers; however, some of the register fields are shorter, because the range of valid numbers may be limited. For example, only three bits are required to represent the day of the week (RTC_WEEKS_REG), because only BCD numbers 1 through 7 are required. The following time and calendar registers are supported (BCD Format):

- RTC_SECONDS_REG — Second Count (00–59)
- RTC_MINUTES_REG — Minute Count (00–59)
- RTC_HOURS_REG — Hour Count (12HR: 01–12; 24HR: 00–23)
- RTC_DAYS_REG — Day of the Month Count (01–31)

- RTC_WEEKS_REG — Day of the Week (0–6: SUN = 0)
- RTC_MONTHS_REG — Month Count (01–12; JAN = 1)
- RTC_YEARS_REG — Year Count (00–99)

23.4.3.2 Register Access

The three register types are as follows and each has its own access constraints:

- TC registers and TC alarm registers
  – TC registers (see Section 23.4.3.1)
  – TC alarm registers:
    - RTC_ALARM_SECONDS_REG
    - RTC_ALARM_MINUTES_REG
    - RTC_ALARM_HOURS_REG
    - RTC_ALARM_DAYS_REG
    - RTC_ALARM_MONTHS_REG
    - RTC_ALARM_YEARS_REG
    - RTC_ALARM2_SECONDS_REG
    - RTC_ALARM2_MINUTES_REG
    - RTC_ALARM2_HOURS_REG
    - RTC_ALARM2_DAYS_REG
    - RTC_ALARM2_MONTHS_REG
    - RTC_ALARM2_YEARS_REG
- General registers:
  – RTC_CTRL_REG
  – RTC_STATUS_REG
  – RTC_INTERRUPTS_REG
  – RTC_SCRATCH0_REG
  – RTC_SCRATCH1_REG
  – RTC_SCRATCH2_REG
  – RTC_KICK0_REG
  – RTC_KICK1_REG
  – RTC_REVISION_REG
  – RTC_SYSCONFIG_REG
  – RTC_IRQWAKEEN
  – RTC_PMIC_REG
  – RTC_RTL_DEBOUNCE_REG
  – RTC_OSC_REG
- Compensation registers:
  – RTC_COMP_LSB_REG
  – RTC_COMP_MSB_REG

23.4.3.3 Register Spurious Write Protection

The module also contains a kicker mechanism (see Figure 23-5) to prevent any spurious writes from changing the register values.

Note

This mechanism requires two register writes to the KICK0 and KICK1 registers with exact data values before the kicker lock mechanism is released. Once released, the registers are writeable. The KICK0 data is 83E7 0B13h; the KICK1 data is 95A4 F1E0h. The mechanism remains in an unlocked state until a reset or invalid data pattern is written to one of the RTC_KICK0_REG or RTC_KICK1_REG registers.

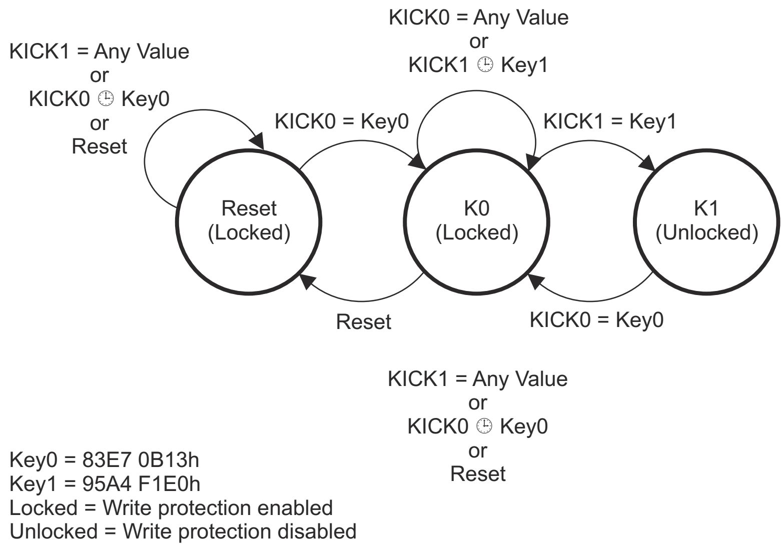


Figure 23-5. Kick Register State Machine Diagram

- S0 is the Reset/Idle state
- S1 is an write cycle of 83E7 0B13h at KICK0 completed state
- S2 is the UNLOCK register WRT state
- S0 → S1 when write cycle of 83E7 0B13h at KICK0
- S1 → S2 when write cycle of 95A4 F1E0h at KICK1
- S1 → S0 when reset event
- S2 → S0 when reset event or write cycle of NOT 83E7 0B13h at KICK0 or write cycle at KICK1
- S2 → S1 when write cycle of 83E7 0B13h at KICK0

23.4.3.4 Reading the Timer/Calendar (TC) Registers

The TC registers have a read-show register. The reading of the SECONDS register updates all TC registers. For example, the Year is updated only on a reading of the SECONDS register. The TC registers are updated every second as the time changes. During a read of the SECONDS register, the RTC copies the current values of the time/date registers into shadow read registers. This isolation assures that the CPU can capture all the time/date values when the SECONDS read request occurs and is not subject to changing register values from time updates.

If desired, the RTC also provides a one-time-triggered minute-rounding feature to round the MINUTE:SECOND registers to the nearest minute (with zero seconds). This feature is enabled by setting the ROUND_30S bit in the control register (RTC_CTRL_REG). The RTC automatically rounds the time values to the nearest minute the next time the SECONDS register is read.

Note

Software should always read the SECONDS register first. However, software does not have to poll any status bit to determine when to read the TC registers. Table 23-6 defines the TC set that gets shadowed.

Table 23-6. RTC Register Names and Values

Time Unit	Range	Remarks
Year	00 to 99	
Month	01 to 12	

Table 23-6. RTC Register Names and Values (continued)

Time Unit	Range	Remarks
Day	01 to 31	Months 1, 3, 5, 7, 8, 10, 12
	01 to 30	Months 4, 6, 9, 11
	01 to 29	Month 2 (leap year)
	01 to 28	Month 2 (common year)
Week	00 to 06	Day of week
Hour	00 to 23	24-hour mode
	01 to 12	AM/PM mode
Minute	00 to 59	
Seconds	00 to 59	

23.4.3.4.1 Rounding Seconds

Time can be rounded to the closest minute, by setting the ROUND_30S bit of the control register (RTC_CTRL_REG). When this bit is set, TC values are set to the closest minute value at the next second. The ROUND_30S bit is automatically cleared when rounding time is performed.

Example:

- If current time is 10H59M45S, round operation changes time to 11H00M00S.
- If current time is 10H59M29S, round operation changes time to 10H59M00S.

23.4.3.5 Modifying the TC Registers

To write correct data from and to the TC and TC alarm registers and read the TC alarm registers, the MPU must first read the BUSY bit of the status register (RTC_STATUS_REG) until BUSY is equal to zero. Once the BUSY flag is zero, there is a 15-μs access period in which the MPU can program the TC and TC alarm registers. Once the 15-μs access period passes, the BUSY flag must again be read from the STATUS register as previously described. If the MPU accesses the TC registers outside of the access period, then the access is not assured.

The MPU can access the RTC_STATUS_REG and the RTC_CTRL_REG registers at any time, with the exception of the RTC_CTRL_REG[5] bit, which can only be changed when the RTC is stopped. The MPU can stop the RTC by clearing the STOP_RTC bit of RTC_CTRL_REG. After clearing this bit, the RUN bit in RTC_STATUS_REG must be checked to verify the RTC has stopped. Once this is confirmed, the TC values can be updated. After the values have been updated, the RTC can be restarted by resetting the STOP_RTC bit.

Note

After writing to a TC register, software must wait four clock cycles before reading the value from the register. If this wait time is not observed and the TC register is accessed, then old data will be read from the register.

CAUTION

To remove any possibility of interrupting the register read process, which introduces a risk of violating the authorized 15-μs access period, TI recommends disabling all incoming interrupts during the register read process.

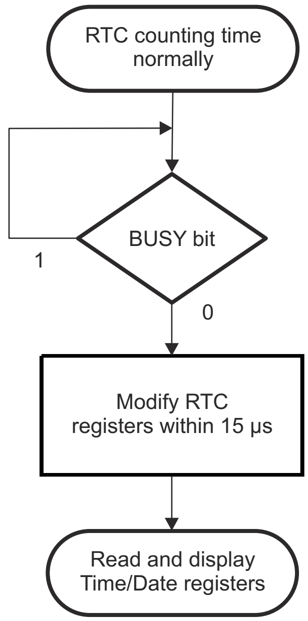


Figure 23-6. Flow Control for Updating RTC Registers

23.4.3.5.1 General Registers

The MPU can access the RTC_STATUS_REG and the RTC_CTRL_REG registers at any time (except the CTRL_REG[5] bit, which must be changed only when the RTC is stopped). For the RTC_INTERRUPTS_REG register, the MPU should respect the available access period to prevent spurious interrupt.

The RTC_DISABLE bit of the RTC_CTRL_REG register must only be used to completely disable the RTC function. When this bit is set, the 32-kHz clock is gated, and the RTC is frozen. From this point, resetting the RTC_DISABLE bit to zero can lead to unexpected behavior. To save power, this bit should only be used if the RTC function is unwanted in the application.

23.4.3.6 Crystal Compensation

To compensate for any inaccuracy of the 32-kHz oscillator the MPU can perform a calibration of the oscillator frequency, calculate the drift compensation versus a 1-hour period, and load the compensation registers with the drift compensation value. Auto compensation is enabled by the AUTO_COMP bit in the RTC_CTRL_REG register. If the value of the COMP_REG registers (RTC_COMP_LSB_REG and RTC_COMP_MSB_REG) is positive, compensation occurs after the second change event. COMP_REG cycles are removed from the next second. If the COMP_REG value is negative, compensation occurs before the second change event. COMP_REG cycles are added to the current second. This enables compensation with a 1 32-kHz period accuracy each hour. The waveform in Figure 23-7 summarizes positive and negative compensation effect.

Access to the RTC_COMP_MSB_REG and RTC_COMP_LSB_REG registers must respect the available access period. These registers should not be updated during compensation (the first second of each hour), but it is alright to update them during the second preceding a compensation event. For example, the MPU could load the compensation value into these registers after each hour event, during an available access period.

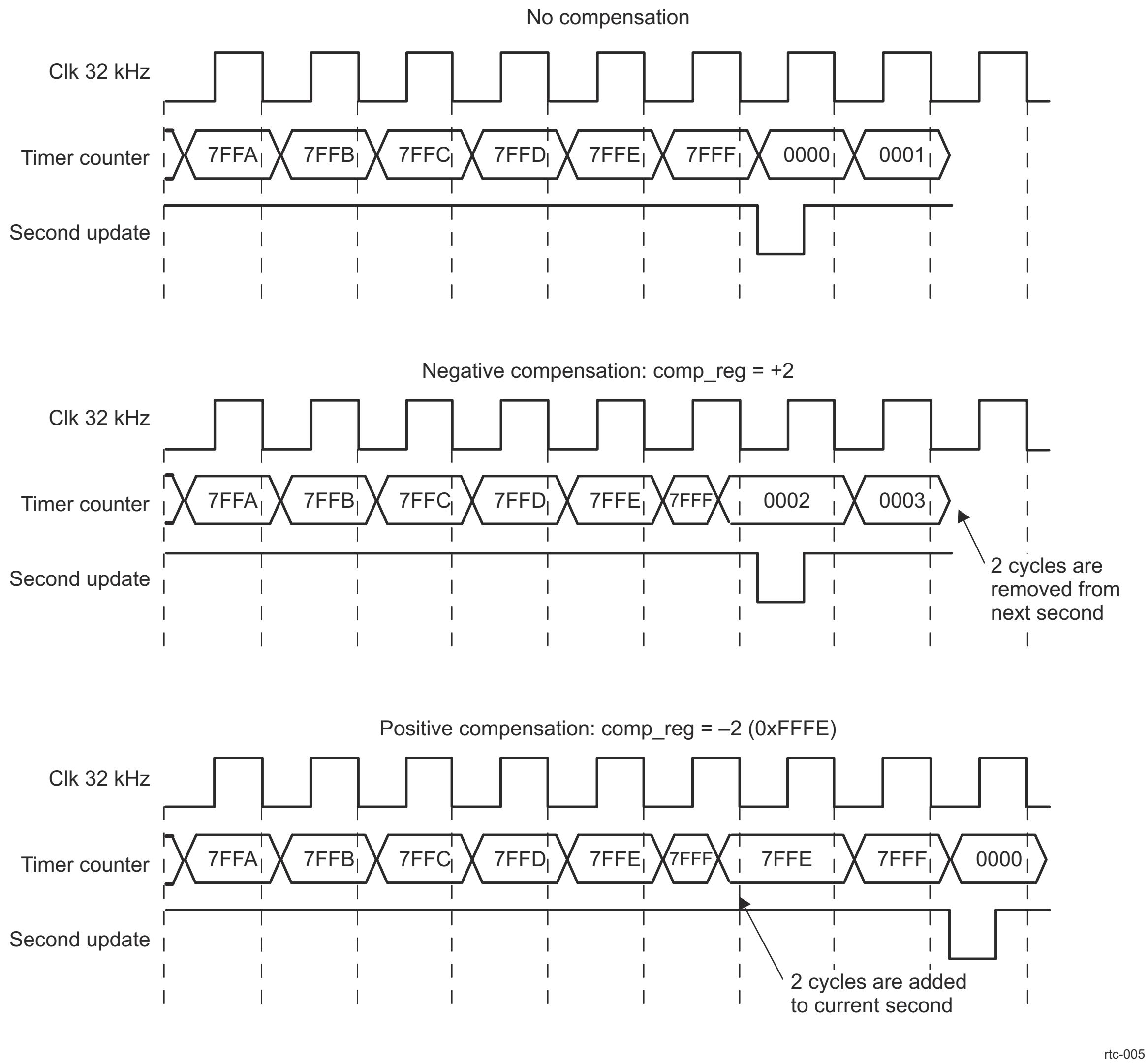


Figure 23-7. Compensation Illustration

23.4.4 Scratch Registers

The RTC provides three general-purpose registers (RTC_SCRATCHx_REG) that can be used to store 32-bit words -- these registers have no functional purpose for the RTC. Software using the RTC may find the SCRATCHx_REG registers to be useful in indicating RTC states. For example, the RTC_SCRATCHx_REG (RTC_SCRATCH0_REG, RTC_SCRATCH1_REG, RTC_SCRATCH2_REG) registers may be used to indicate write-protection lock status or unintentional power downs. To indicate write-protection, the software should write a unique value to one of the RTC_SCRATCHx_REG registers when write-protection is disabled and another unique value when write-protection is enabled again. In this way, the lock-status of the registers can be determined quickly by reading the RTC_SCRATCHx_REG register. To indicate unintentional power downs, the software should write a unique value to one of the RTC_SCRATCHx_REG registers when RTC is configured and enabled. If the RTC is unintentionally powered down, the value written to the RTC_SCRATCHx_REG register is cleared.

23.4.5 Debouncing

The debounce timer uses the 32768-Hz clock or the clock coming from the external oscillator, depending on the SW configurations for the RTC. It allows choosing the timing or accuracy of the "debouncing".

A register receives a bit from the reference pin. Software then chooses the timing if it uses the debouncing feature like a timer, or it chooses the accuracy if it uses the debouncing like a real debouncing. The debouncing is finished when the reference pin stays the same value, defined in RTC_RTL_DEBOUNCE_REG[7:0] DEBOUNCE_REG, for a defined time.

If the RTC module uses the internal 32-kHz clock then setting the RTC_RTL_DEBOUNCE_REG[7:0] DEBOUNCE_REG to 0x0 results in a debounce time of 30.52us. Otherwise (DEBOUNCE_REG = n) the debounce time is 30.52μs × (n + 1). The debounce time may be different, depending on, whether the module uses the internal 32 kHz clock, or an external clock source.

23.4.6 Power Management

23.4.6.1 Device-Level Power Management

The RTC supports the power idle protocol. The RTC has two SWakeup ports: one for the alarm event and one for a timer event.

When the RTC is in IDLE mode, the clock is turned off and the 32 kHz clock remains on. The time and calendar continue to count in IDLE mode. When the RTC is placed back in FUNCTIONAL mode, the TC registers can be read. The RTC idle mode configuration is done through the RTC_SYSCONFIG_REG[1:0] IDLEMODE bit (see Table 23-7).

Table 23-7. RTC Idle Configuration

IDLEMODE	Description
0x0	Force IDLE: idle state of the local target follows (acknowledges) the system IDLE request unconditionally
0x1	No-idle mode: RTC never enters idle state
0x2	Smart-idle mode: RTC subsystem waits for all interrupts/events to be serviced before entering in IDLE mode.
0x3	Smart-idle wakeup-capable mode: RTC eventually acknowledges the system IDLE requests, depending on its internal requirements, RTC can generate wakeup events when in idle state.

The Alarm SWakeup event can be used to wake up the RTC when it is in idle state. To do so, the alarm must be set and enabled before RTC enters idle state. Also the wakeup generation for alarm/timer event must be set (bits [1:0] in the RTC_IRQWAKEEN register). Once this is done, the SWakeup event occurs when the alarm event triggers.

Note

SWakeup is not periodic, using Timer SWakeup event to wake up the RTC when in idle state is not recommended. Use Alarm SWakeup instead.

There are two idle modes: smart-idle mode and smart-idle wakeup-capable mode. The RTC should be configured to smart-idle wakeup-capable mode in order to use the SWakeup events.

23.4.6.2 Subsystem-Level Power Management — PMIC Mode

The RTC generates PMIC_PWR_ENABLE control, which can be used to control an external PMIC. The control of the associated PMIC wakeups and alarms is done through the RTC_PMIC_REG[22:0] bits.

23.5 RTC Low-Level Programming Guide

23.5.1 Global Initialization

23.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the real time clock is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the RTC.

Table 23-8. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. for more information about the module configuration. For more information about the module configuration, see *Module Level Clock Management*, in *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing must be set in the control module. For more information about the module configuration, see *Pad Configuration Registers* in *Control Module*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

23.5.1.2 RTC Module Global Initialization

23.5.1.2.1 Main Sequence – RTC Module Global Initialization

Table 23-9 lists the steps for initializing the RTC module when the module is to be used for the first time.

Table 23-9. RTC Module Global Initialization

Step	Register/Bit Field/Programming Model	Value
Write 0x83E7 0B13 to the Kick0 Register to unlock register writes	RTC_KICK0_REG	0x83E70B13
Write 0x95A4 F1E0 to the Kick1 Register to unlock register writes	RTC_KICK1_REG	0x95A4F1E0
Select the clock source (internal or external 32-kHz clock)	RTC_OSC_REG[3] 32KCLK_SEL	-
If external clock is selected, start the oscillator. Wait for oscillator to stabilize.	RTC_OSC_REG[4] OSC32K_GZ	0
Enable the 32-kHz clock	RTC_OSC_REG[6] 32KCLK_EN	1
Configure slave idle mode	RTC_SYSCONFIG_REG [1:0] IDLEMODE	0x-
Enable interrupts, if needed	RTC_INTERRUPTS_REG[4:0]	0x-
Enable wakeup interrupts, if needed	RTC_IRQWAKEEN[1:0]	0x-
Set current time	RTC_SECONDS_REG[6:0]	BCD
	RTC_MINUTES_REG[6:0]	BCD
	RTC_HOURS_REG[5:0]	BCD
Set current date	RTC_DAYS_REG[5:0]	BCD
	RTC_MONTHS_REG[4:0]	BCD
	RTC_YEARS_REG[7:0]	BCD
	RTC_WEEKS_REG[2:0]	BCD
Run the real-time clock	RTC_CTRL_REG[0] STOP_RTC	1

23.6 RTC Register Manual

23.6.1 RTC Instance Summary

Table 23-10. RTC Instance Summary

Module Name	Base Address	Size
RTC_SS	0x4883 8000	160 bytes

23.6.2 RTC_SS Registers

23.6.2.1 RTC_SS Register Summary

Table 23-11. RTC_SS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	RTC_SS Physical Address
RTC_SECONDS_REG	RW	32	0x0000 0000	0x4883 8000
RTC_MINUTES_REG	RW	32	0x0000 0004	0x4883 8004
RTC_HOURS_REG	RW	32	0x0000 0008	0x4883 8008
RTC_DAYS_REG	RW	32	0x0000 000C	0x4883 800C
RTC_MONTHS_REG	RW	32	0x0000 0010	0x4883 8010
RTC_YEARS_REG	RW	32	0x0000 0014	0x4883 8014
RTC_WEEKS_REG	RW	32	0x0000 0018	0x4883 8018
RTC_ALARM_SECONDS_REG	RW	32	0x0000 0020	0x4883 8020
RTC_ALARM_MINUTES_REG	RW	32	0x0000 0024	0x4883 8024
RTC_ALARM_HOURS_REG	RW	32	0x0000 0028	0x4883 8028
RTC_ALARM_DAYS_REG	RW	32	0x0000 002C	0x4883 802C
RTC_ALARM_MONTHS_REG	RW	32	0x0000 0030	0x4883 8030
RTC_ALARM_YEARS_REG	RW	32	0x0000 0034	0x4883 8034
RTC_CTRL_REG	RW	32	0x0000 0040	0x4883 8040
RTC_STATUS_REG	RW	32	0x0000 0044	0x4883 8044
RTC_INTERRUPTS_REG	RW	32	0x0000 0048	0x4883 8048
RTC_COMP_LSB_REG	RW	32	0x0000 004C	0x4883 804C
RTC_COMP_MSB_REG	RW	32	0x0000 0050	0x4883 8050
RTC_OSC_REG	RW	32	0x0000 0054	0x4883 8054
RTC_SCRATCH0_REG	RW	32	0x0000 0060	0x4883 8060
RTC_SCRATCH1_REG	RW	32	0x0000 0064	0x4883 8064
RTC_SCRATCH2_REG	RW	32	0x0000 0068	0x4883 8068
RTC_KICK0_REG	RW	32	0x0000 006C	0x4883 806C
RTC_KICK1_REG	RW	32	0x0000 0070	0x4883 8070
RTC_REVISION_REG	R	32	0x0000 0074	0x4883 8074
RTC_SYSCONFIG_REG	RW	32	0x0000 0078	0x4883 8078
RTC_IRQWAKEEN	RW	32	0x0000 007C	0x4883 807C
RTC_ALARM2_SECONDS_REG	RW	32	0x0000 0080	0x4883 8080
RTC_ALARM2_MINUTES_REG	RW	32	0x0000 0084	0x4883 8084
RTC_ALARM2_HOURS_REG	RW	32	0x0000 0088	0x4883 8088
RTC_ALARM2_DAYS_REG	RW	32	0x0000 008C	0x4883 808C
RTC_ALARM2_MONTHS_REG	RW	32	0x0000 0090	0x4883 8090
RTC_ALARM2_YEARS_REG	RW	32	0x0000 0094	0x4883 8094
RTC_PMIC_REG	RW	32	0x0000 0098	0x4883 8098
RTC_RTL_DEBOUNCE_REG	RW	32	0x0000 009C	0x4883 809C

23.6.2.2 RTC_SS Register Description

Table 23-12. RTC_SECONDS_REG

Address Offset	0x0000 0000

Table 23-12. RTC_SECONDS_REG (continued)

Physical Address	0x4883 8000	**Instance**	RTC_SS
Description	Used to program the required seconds value of the current time. Seconds are stored in BCD format, the decimal numbers are encoded with their binary equivalent. That is, if seconds value is 45, then SEC0 = 5 and SEC1 = 4.		
Type	RW		

31–7	6–4	3–0
RESERVED	SEC1	SEC0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	SEC1	Second digit of seconds. Range is 0 to 5	RW	0x0
3:0	SEC0	First digit of seconds. Range is 0 to 9	RW	0x0

Table 23-13. Register Call Summary for Register RTC_SECONDS_REG

RTC Functional Description
- Time/Calendar Data Format: [0]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-14. RTC_MINUTES_REG

Address Offset	0x0000 0004		
Physical Address	0x4883 8004	**Instance**	RTC_SS
Description	Used to program the required minutes value of the current time. Minutes are stored in BCD format, the decimal numbers are encoded with their binary equivalent. That is, if minutes value is 32, then MIN0 = 2 and MIN1 = 3.		
Type	RW		

31–7	6–4	3–0
RESERVED	MIN1	MIN0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	MIN1	Second digit of minutes Range is 0 to 5	RW	0x0
3:0	MIN0	First digit of minutes Range is 0 to 9	RW	0x0

Table 23-15. Register Call Summary for Register RTC_MINUTES_REG

RTC Functional Description
- Time/Calendar Data Format: [0]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-16. RTC_HOURS_REG

Address Offset	0x0000 0008		
Physical Address	0x4883 8008	**Instance**	RTC_SS

Table 23-16. RTC_HOURS_REG (continued)

Description	Used to program the hours value of the current time. Hours are stored in BCD format, the decimal numbers are encoded with their binary equivalent. That is, if hour is 18, then HOUR0 = 8 and HOUR1 = 1.
Type	RW

31–8	7	6	5–4	3–0
RESERVED	PM_NAM	RESERVED	HOUR1	HOUR0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	PM_NAM	Only used in PM_AM mode (otherwise 0) 0 = AM 1 = PM	RW	0x0
6	RESERVED	Reserved	R	0x0
5:4	HOUR1	Second digit of hours Range is 0 to 2	RW	0x0
3:0	HOUR0	First digit of hours Range is 0 to 9	RW	0x0

Table 23-17. Register Call Summary for Register RTC_HOURS_REG

RTC Functional Description

- Time/Calendar Data Format: [0]

RTC Low-Level Programming Guide

- RTC Module Global Initialization: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-18. RTC_DAYS_REG

Address Offset	0x0000 000C		
Physical Address	0x4883 800C	**Instance**	RTC_SS
Description	Used to program the day of the month value of the current date. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the day value of the date is 28, DAY0 is set as 8 and DAY1 is set as 2.		
Type	RW		

31–6	5–4	3–0
RESERVED	DAY1	DAY0

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:4	DAY1	Second digit of days Range from 0 to 3	RW	0x0
3:0	DAY0	First digit of days Range from 0 to 9	RW	0x1

Table 23-19. Register Call Summary for Register RTC_DAYS_REG

RTC Functional Description

- Time/Calendar Data Format: [0]

RTC Low-Level Programming Guide

- RTC Module Global Initialization: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-20. RTC_MONTHS_REG

Address Offset	0x0000 0010		
Physical Address	0x4883 8010	**Instance**	RTC_SS
Description	MONTHS_REG is used to set the month in the year value of the current date. Months are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.		
Type	RW		

31:5	4	3:0
RESERVED	MONTH1	MONTH0

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4	MONTH1	Second digit of months Range from 0 to 1	RW	0x0
3:0	MONTH0	First digit of months Range from 0 to 9	RW	0x1

Table 23-21. Register Call Summary for Register RTC_MONTHS_REG

RTC Functional Description
- Time/Calendar Data Format: [0]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-22. RTC_YEARS_REG

Address Offset	0x0000 0014		
Physical Address	0x4883 8014	**Instance**	RTC_SS
Description	YEARS_REG is used to program the year value of the current date. The year value is represented by only the last two digits and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The year 1979 is programmed as 79 with YEAR0 set as 9 and YEAR1 set as 7.		
Type	RW		

31:8	7:4	3:0
RESERVED	YEAR1	YEAR0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:4	YEAR1	Second digit of Years Range from 0 to 9	RW	0x0
3:0	YEAR0	First digit of Years Range from 0 to 9	RW	0x0

Table 23-23. Register Call Summary for Register RTC_YEARS_REG

RTC Functional Description
- Time/Calendar Data Format: [0]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-24. RTC_WEEKS_REG

Address Offset	0x0000 0018		
Physical Address	0x4883 8018	**Instance**	RTC_SS
Description	WEEKS_REG is used to program the day of the week value of the current date. The day of the week is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.		
Type	RW		

31:3	2:0
RESERVED	WEEK

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0
2:0	WEEK	First digit of Days in a week Range from 0 (Sunday) to 6 (Saturday)	RW	0x0

Table 23-25. Register Call Summary for Register RTC_WEEKS_REG

RTC Functional Description

- Time/Calendar Data Format: [0] [1]

RTC Low-Level Programming Guide

- RTC Module Global Initialization: [2]

RTC Register Manual

- RTC_SS Register Summary: [3]

Table 23-26. RTC_ALARM_SECONDS_REG

Address Offset	0x0000 0020		
Physical Address	0x4883 8020	**Instance**	RTC_SS
Description	ALARM_SECONDS_REG is used to program the seconds value for the alarm interrupt. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.		
Type	RW		

31:7	6:4	3:0
RESERVED	ALARM_SEC1	ALARM_SEC0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	ALARM_SEC1	Second digit of seconds Range is 0 to 5	RW	0x0
3:0	ALARM_SEC0	First digit of seconds Range is 0 to 9	RW	0x0

Table 23-27. Register Call Summary for Register RTC_ALARM_SECONDS_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-28. RTC_ALARM_MINUTES_REG

Address Offset	0x0000 0024		
Physical Address	0x4883 8024	**Instance**	RTC_SS

Table 23-28. RTC_ALARM_MINUTES_REG (continued)

Description	ALARM_MINUTES_REG is used to program the minute value for the alarm interrupt. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="25">RESERVED</td><td colspan="3">ALARM_MIN1</td><td colspan="4">ALARM_MIN0</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	ALARM_MIN1	Second digit of minutes Range is 0 to 5	RW	0x0
3:0	ALARM_MIN0	First digit of minutes Range is 0 to 9	RW	0x0

Table 23-29. Register Call Summary for Register RTC_ALARM_MINUTES_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-30. RTC_ALARM_HOURS_REG

Address Offset	0x0000 0028		
Physical Address	0x4883 8028	**Instance**	RTC_SS
Description	ALARM_HOURS_REG is used to program the hour value for the alarm interrupt. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td>ALARM_PM_NAM</td><td>RESERVED</td><td colspan="2">ALARM_HOUR1</td><td colspan="4">ALARM_HOUR0</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	ALARM_PM_NAM	Only used in PM_AM mode (otherwise 0) 0 = AM 1 = PM	RW	0x0
6	RESERVED	Reserved	R	0x0
5:4	ALARM_HOUR1	Second digit of hours. Range is 0 to 2	RW	0x0
3:0	ALARM_HOUR0	First digit of hours. Range is 0 to 9	RW	0x0

Table 23-31. Register Call Summary for Register RTC_ALARM_HOURS_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-32. RTC_ALARM_DAYS_REG

Address Offset	0x0000 002C		
Physical Address	0x4883 802C	**Instance**	RTC_SS
Description	ALARM_DAYS_REG is used to program the day of the month value for the alarm interrupt. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.		
Type	RW		

31:6	5:4	3:0
RESERVED	ALARM_DAY1	ALARM_DAY0

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:4	ALARM_DAY1	Second digit for days. Range from 0 to 3	RW	0x0
3:0	ALARM_DAY0	First digit for days. Range from 0 to 9	RW	0x1

Table 23-33. Register Call Summary for Register RTC_ALARM_DAYS_REG

RTC Functional Description
- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-34. RTC_ALARM_MONTHS_REG

Address Offset	0x0000 0030		
Physical Address	0x4883 8030	**Instance**	RTC_SS
Description	ALARM_MONTHS_REG is used to program the month in the year value for the alarm interrupt. The month is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent		
Type	RW		

31:5	4	3:0
RESERVED	ALARM_MONTH1	ALARM_MONTH0

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4	ALARM_MONTH1	Second digit of months. Range from 0 to 1	RW	0x0
3:0	ALARM_MONTH0	First digit of months. Range from 0 to 9	RW	0x1

Table 23-35. Register Call Summary for Register RTC_ALARM_MONTHS_REG

RTC Functional Description
- Interrupt Description: [0]
- Register Access: [1]

Table 23-35. Register Call Summary for Register RTC_ALARM_MONTHS_REG (continued)

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-36. RTC_ALARM_YEARS_REG

Address Offset	0x0000 0034		
Physical Address	0x4883 8034	**Instance**	RTC_SS
Description	ALARM_YEARS_REG is used to program the year for the alarm interrupt. Only the last two digits are used to represent the year and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent		
Type	RW		

31:8	7:4	3:0
RESERVED	ALARM_YEAR1	ALARM_YEAR0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:4	ALARM_YEAR1	Second digit of years. Range from 0 to 9	RW	0x0
3:0	ALARM_YEAR0	First digit of years. Range from 0 to 9	RW	0x0

Table 23-37. Register Call Summary for Register RTC_ALARM_YEARS_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-38. RTC_CTRL_REG

Address Offset	0x0000 0040		
Physical Address	0x4883 8040	**Instance**	RTC_SS
Description	CTRL_REG contains the controls to enable/disable RTC.		
Type	RW		

31:7	6	5	4	3	2	1	0
RESERVED	RTC_DISABLE	SET_32_COUNTER	TEST_MODE	MODE_12_24	AUTO_COMP	ROUND_30S	STOP_RTC

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6	RTC_DISABLE	0: RTC enable 1: RTC disable (no 32-kHz clock)	RW	0x0
5	SET_32_COUNTER	0: No action 1: Set the 32-kHz counter with comp_reg value	RW	0x0
4	TEST_MODE	0: Functional mode 1: Test mode (Auto compensation is enabled when the 32-kHz counter reaches its end.)	RW	0x0

Bits	Field Name	Description	Type	Reset
3	MODE_12_24	0: 24-hour mode 1: 12-hour mode (PM-AM mode)	RW	0x0
2	AUTO_COMP	0: No auto compensation 1: Auto compensation enabled	RW	0x0
1	ROUND_30S	0: No update 1: When a 1 is written, the time is rounded to the closest minute.	RW	0x0
0	STOP_RTC	0: RTC is frozen. 1: RTC is running.	RW	0x0

Table 23-39. Register Call Summary for Register RTC_CTRL_REG

RTC Functional Description

- Clock Source: [0] [1]
- Register Access: [2]
- Reading the Timer/Calendar (TC) Registers: [3] [4]
- Modifying the TC Registers: [5] [6] [7] [8] [9]
- Crystal Compensation: [10]

RTC Low-Level Programming Guide

- RTC Module Global Initialization: [11]

RTC Register Manual

- RTC_SS Register Summary: [12]

Table 23-40. RTC_STATUS_REG

Address Offset	0x0000 0044		
Physical Address	0x4883 8044	**Instance**	RTC_SS
Description	RTC STATUS_REG contains bits that signal the status of interrupts, events to the processor. Status for the alarm interrupt and timer events are notified by the register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ALARM2	ALARM	EVENT_1D	EVENT_1H	EVENT_1M	EVENT_1S	RUN	BUSY

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	ALARM2	Indicates that an alarm2 interrupt has been generated. Software must wait 31 µs before it clears this status to allow PMIC_PWR_ENABLE 1 - 0 transmission.	RW	0x0
6	ALARM	Indicates that an alarm interrupt has been generated. Writing 1 to the bit clears the interrupt.	RW	0x0
5	EVENT_1D	One day has occurred.	R	0x0
4	EVENT_1H	One hour has occurred.	R	0x0
3	EVENT_1M	One minute has occurred.	R	0x0
2	EVENT_1S	One second has occurred.	R	0x0
1	RUN	0: RTC is frozen. 1: RTC is running.	R	0x0

Bits	Field Name	Description	Type	Reset
0	BUSY	0: Updating event in more than 15 µs. 1: Updating event. This bit will give the status of RTC module. The time and alarm registers can be modified only when this bit is 0.	R	0x0

Table 23-41. Register Call Summary for Register RTC_STATUS_REG

RTC Functional Description
- Interrupt Description: [0] [1] [2]
- Register Access: [3]
- Modifying the TC Registers: [4] [5] [6] [7]

RTC Register Manual
- RTC_SS Register Summary: [8]

Table 23-42. RTC_INTERRUPTS_REG

Address Offset	0x0000 0048		
Physical Address	0x4883 8048	**Instance**	RTC_SS
Description	INTERRUPTS_REG is used to enable or disable RTC from generating interrupts. The timer interrupt and alarm interrupt can be controlled using this register.		
Type	RW		

31–5	4	3	2	1–0
RESERVED	IT_ALARM2	IT_ALARM	IT_TIMER	EVERY

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4	IT_ALARM2	Enable one interrupt when the alarm value is reached (TC ALARM2 registers) by the TC registers	RW	0x0
3	IT_ALARM	Enable one interrupt when the alarm value is reached (TC ALARM registers) by the TC registers	RW	0x0
2	IT_TIMER	Enable periodic interrupt 0 = interrupt disabled 1 = interrupt enabled	RW	0x0
1:0	EVERY	Interrupt period: 0 - every second 1 - every minute 2 - every hour 3 - every day	RW	0x0

Table 23-43. Register Call Summary for Register RTC_INTERRUPTS_REG

RTC Functional Description
- Interrupt Description: [0] [1] [2] [3]
- Register Access: [4]
- Modifying the TC Registers: [5]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [6]

RTC Register Manual
- RTC_SS Register Summary: [7]

Table 23-44. RTC_COMP_LSB_REG

Address Offset	0x0000 004C		
Physical Address	0x4883 804C	**Instance**	RTC_SS
Description	COMP_LSB_REG is used to program the LSB value of the 32-kHz periods to be added to the 32-kHz counter every hour. This is used to compensate the oscillator drift. The COMP_LSB_REG works with the compensation (MSB) register (COMP_MSB_REG). The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to OCCUR.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RTC_COMP_LSB							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:0	RTC_COMP_LSB	Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour.	RW	0x0

Table 23-45. Register Call Summary for Register RTC_COMP_LSB_REG

RTC Functional Description

- Register Access: [0]
- Crystal Compensation: [1] [2]

RTC Register Manual

- RTC_SS Register Summary: [3]

Table 23-46. RTC_COMP_MSB_REG

Address Offset	0x0000 0050		
Physical Address	0x4883 8050	**Instance**	RTC_SS
Description	COMP_MSB_REG is used to program the MSB value of the 32-kHz periods to be added to the 32-kHz counter every hour. This is used to compensate the oscillator drift. The COMP_MSB_REG works with the compensation (LSB) register (COMP_LSB_REG) to set the hourly oscillator compensation value. The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to OCCUR.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RTC_COMP_MSB							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:0	RTC_COMP_MSB	Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour	RW	0x0

Table 23-47. Register Call Summary for Register RTC_COMP_MSB_REG

RTC Functional Description

- Register Access: [0]
- Crystal Compensation: [1] [2]

RTC Register Manual

- RTC_SS Register Summary: [3]

Table 23-48. RTC_OSC_REG

Address Offset	0x0000 0054		
Physical Address	0x4883 8054	**Instance**	RTC_SS
Description	OSC_REG is used to program the oscillator resistance value, and to select and enable the clock source.		

Table 23-48. RTC_OSC_REG (continued)

Type	RW

31–7	6	5	4	3	2	1	0
RESERVED	K32CLK_EN	RESERVED	OSC32K_GZ	K32CLK_SEL	RES_SELECT	SW2	SW1

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6	K32CLK_EN	32-kHz clock enable post clock mux of RTC_32K_AUX_CLK and RTC_32K_CLK.	RW	0x0
5	RESERVED	Reserved	R	0x0
4	OSC32K_GZ	Disable the oscillator and applies high impedance to the output. 0: Enable 1: Disabled and high impedance	RW	0x1
3	K32CLK_SEL	32-kHz clock source select. 0: Selects internal clock source, namely RTC_32K_AUX_CLK, from PRCM. 1: Selects external clock source namely RTC_32K_CLK, from the 32-kHz Oscillator.	RW	0x0
2	RES_SELECT	External feedback resistor selection. 0: Internal 1: External	RW	0x0
1	SW2	Inverter size adjustment.	RW	0x0
0	SW1	Inverter size adjustment.	RW	0x0

Table 23-49. Register Call Summary for Register RTC_OSC_REG

RTC Functional Description
- Clock Source: [0] [1] [2]
- Register Access: [3]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [4] [5]

RTC Register Manual
- RTC_SS Register Summary: [6]

Table 23-50. RTC_SCRATCH0_REG

Address Offset	0x0000 0060		
Physical Address	0x4883 8060	**Instance**	RTC_SS
Description	Used to hold some required values for the RTC register.		
Type	RW		

31–0
RTCSCRATCH0

Bits	Field Name	Description	Type	Reset
31:0	RTCSCRATCH0	Scratch registers, available to program.	RW	0x0

Table 23-51. Register Call Summary for Register RTC_SCRATCH0_REG

RTC Functional Description
- Register Access: [0]
- Scratch Registers: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-52. RTC_SCRATCH1_REG

Address Offset	0x0000 0064		
Physical Address	0x4883 8064	**Instance**	RTC_SS
Description	Used to hold some required values for the RTC register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCSCRATCH1																															

Bits	Field Name	Description	Type	Reset
31:0	RTCSCRATCH1	Scratch registers, available to program	RW	0x0

Table 23-53. Register Call Summary for Register RTC_SCRATCH1_REG

RTC Functional Description
- Register Access: [0]
- Scratch Registers: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-54. RTC_SCRATCH2_REG

Address Offset	0x0000 0068		
Physical Address	0x4883 8068	**Instance**	RTC_SS
Description	Used to hold some required values for the RTC register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCSCRATCH2																															

Bits	Field Name	Description	Type	Reset
31:0	RTCSCRATCH2	Scratch registers, available to program.	RW	0x0

Table 23-55. Register Call Summary for Register RTC_SCRATCH2_REG

RTC Functional Description
- Register Access: [0]
- Scratch Registers: [1]

RTC Register Manual
- RTC_SS Register Summary: [2]

Table 23-56. RTC_KICK0_REG

Address Offset	0x0000 006C		
Physical Address	0x4883 806C	**Instance**	RTC_SS
Description	The KICK0 register allows writing to unlock the kick0 data. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0, followed by the value of 95A4 F1E0h written to KICK1. RTC register write protection is enabled when any value is written to KICK0		

Table 23-56. RTC_KICK0_REG (continued)

Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KICK0																															

Bits	Field Name	Description	Type	Reset
31:0	KICK0	Kick0 data.	W	0x0

Table 23-57. Register Call Summary for Register RTC_KICK0_REG

RTC Functional Description
- Register Access: [0]
- OCP MMR Spurious Write Protection: [1]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [2]

RTC Register Manual
- RTC_SS Register Summary: [3]

Table 23-58. RTC_KICK1_REG

Address Offset	0x0000 0070		
Physical Address	0x4883 8070	**Instance**	RTC_SS
Description	Kick1 data. The KICK1 register allows writing to unlock the kick1 data and the kicker mechanism to write to other registers. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0, followed by the value of 95A4 F1E0h written to KICK1.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KICK1																															

Bits	Field Name	Description	Type	Reset
31:0	KICK1	Kick1 data.	W	0x0

Table 23-59. Register Call Summary for Register RTC_KICK1_REG

RTC Functional Description
- Register Access: [0]
- OCP MMR Spurious Write Protection: [1]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [2]

RTC Register Manual
- RTC_SS Register Summary: [3]

Table 23-60. RTC_REVISION_REG

Address Offset	0x0000 0074		
Physical Address	0x4883 8074	**Instance**	RTC_SS
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-

Table 23-61. Register Call Summary for Register RTC_REVISION_REG

RTC Functional Description

- Register Access: [0]

RTC Register Manual

- RTC_SS Register Summary: [1]

Table 23-62. RTC_SYSCONFIG_REG

Address Offset	0x0000 0078		
Physical Address	0x4883 8078	**Instance**	RTC_SS
Description			
Type	RW		

31–2	1–0
RESERVED	IDLEMODE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1:0	IDLEMODE	Configuration of the local target state management mode, By definition target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e., regardless of the IP module's internal requirements; Backup mode, for debug only. 0x1: No-idle mode: local target never enters idle state, Backup mode, for debug only. 0x2: Smart-idle mode: local target's state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements, IP module shall not generate (IRQ- or DMArequest- related) wakeup events. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges the system's idle requests, depending on the IP module's internal requirements, IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state, Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.	RW	0x2

Table 23-63. Register Call Summary for Register RTC_SYSCONFIG_REG

RTC Functional Description

- Register Access: [0]
- Device-Level Power Management: [1]

RTC Low-Level Programming Guide

- RTC Module Global Initialization: [2]

RTC Register Manual

- RTC_SS Register Summary: [3]

Table 23-64. RTC_IRQWAKEEN

Address Offset	0x0000 007C

Table 23-64. RTC_IRQWAKEEN (continued)

Physical Address	0x4883 807C	**Instance**	RTC_SS
Description			
Type	RW		

31–2	1	0
RESERVED	ALARM_WAKEEN	TIMMER_WAKEEN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	RW	0x0
1	ALARM_WAKEEN	Wakeup generation for event Alarm 0: Wakeup disabled 1: Wakeup enable	RW	0x0
0	TIMMER_WAKEEN	Wakeup generation for event Timer 0: Wakeup disabled 1: Wakeup enable Timer wakeup should not get used.	RW	0x0

Table 23-65. Register Call Summary for Register RTC_IRQWAKEEN

RTC Functional Description
- Register Access: [0]
- Device-Level Power Management: [1]

RTC Low-Level Programming Guide
- RTC Module Global Initialization: [2]

RTC Register Manual
- RTC_SS Register Summary: [3]

Table 23-66. RTC_ALARM2_SECONDS_REG

Address Offset	0x0000 0080		
Physical Address	0x4883 8080	**Instance**	RTC_SS
Description	ALARM2_SECONDS_REG is used to program the seconds value of the ALARM2 time		
Type	RW		

31–7	6–4	3–0
RESERVED	ALARM2_SEC1	ALARM2_SEC0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	ALARM2_SEC1	Second digit of seconds Range is 0 to 5	RW	0x0
3:0	ALARM2_SEC0	First digit of seconds Range is 0 to 9	RW	0x0

Table 23-67. Register Call Summary for Register RTC_ALARM2_SECONDS_REG

RTC Functional Description
- Interrupt Description: [0]
- Register Access: [1]

Table 23-67. Register Call Summary for Register RTC_ALARM2_SECONDS_REG (continued)

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-68. RTC_ALARM2_MINUTES_REG

Address Offset	0x0000 0084		
Physical Address	0x4883 8084	**Instance**	RTC_SS
Description			
Type	RW		

31:7	6:4	3:0
RESERVED	ALARM2_MIN1	ALARM2_MIN0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	ALARM2_MIN1	Second digit of minutes Range is 0 to 5	RW	0x0
3:0	ALARM2_MIN0	First digit of minutes Range is 0 to 9	RW	0x0

Table 23-69. Register Call Summary for Register RTC_ALARM2_MINUTES_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-70. RTC_ALARM2_HOURS_REG

Address Offset	0x0000 0088		
Physical Address	0x4883 8088	**Instance**	RTC_SS
Description			
Type	RW		

31:8	7	6	5:4	3:0
RESERVED	ALARM2_PM_NAM	RESERVED	ALARM2_HOUR1	ALARM2_HOUR0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	ALARM2_PM_NAM	Only used in PM_AM mode (otherwise 0) 0 = AM 1 = PM	RW	0x0
6	RESERVED	Reserved	R	0x0
5:4	ALARM2_HOUR1	Second digit of hours Range is 0 to 2	RW	0x0
3:0	ALARM2_HOUR0	First digit of hours Range is 0 to 9	RW	0x0

Table 23-71. Register Call Summary for Register RTC_ALARM2_HOURS_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

Table 23-71. Register Call Summary for Register RTC_ALARM2_HOURS_REG (continued)

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-72. RTC_ALARM2_DAYS_REG

Address Offset	0x0000 008C		
Physical Address	0x4883 808C	**Instance**	RTC_SS
Description			
Type	RW		

31:6	5:4	3:0
RESERVED	ALARM_DAY1	ALARM_DAY0

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:4	ALARM_DAY1	Second digit for days Range from 0 to 3	RW	0x0
3:0	ALARM_DAY0	First digit for days Range from 0 to 9	RW	0x1

Table 23-73. Register Call Summary for Register RTC_ALARM2_DAYS_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-74. RTC_ALARM2_MONTHS_REG

Address Offset	0x0000 0090		
Physical Address	0x4883 8090	**Instance**	RTC_SS
Description			
Type	RW		

31:5	4	3:0
RESERVED	ALARM2_MONTH1	ALARM2_MONTH0

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4	ALARM2_MONTH1	Second digit of months Range from 0 to 1	RW	0x0
3:0	ALARM2_MONTH0	First digit of months Range from 0 to 9	RW	0x1

Table 23-75. Register Call Summary for Register RTC_ALARM2_MONTHS_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

Table 23-75. Register Call Summary for Register RTC_ALARM2_MONTHS_REG (continued)

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-76. RTC_ALARM2_YEARS_REG

Address Offset	0x0000 0094		
Physical Address	0x4883 8094	**Instance**	RTC_SS
Description			
Type	RW		

31:8	7:4	3:0
RESERVED	ALARM2_YEAR1	ALARM2_YEAR0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:4	ALARM2_YEAR1	Second digit of Years Range from 0 to 9	RW	0x0
3:0	ALARM2_YEAR0	First digit of Years Range from 0 to 9	RW	0x0

Table 23-77. Register Call Summary for Register RTC_ALARM2_YEARS_REG

RTC Functional Description

- Interrupt Description: [0]
- Register Access: [1]

RTC Register Manual

- RTC_SS Register Summary: [2]

Table 23-78. RTC_PMIC_REG

Address Offset	0x0000 0098		
Physical Address	0x4883 8098	**Instance**	RTC_SS
Description			
Type	RW		

31:23	22:19	18:17	16	15:12	11:8	7:4	3:0
RESERVED	EXT_WAKEUP_POL_HL	PWR_ENABLE_SM	PWR_ENABLE_EN	EXT_WAKEUP_STATUS	EXT_WAKEUP_DB_EN	EXT_WAKEUP_POL	EXT_WAKEUP_EN

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	RW	0x0
22:19	EXT_WAKEUP_POL_HL	External wakeup inputs polarity enable for Active High and Active Low 0: Disabled 1: Enabled, Active High and Active Low EXT_WAKEUP_POL_HL[0] controls ext_wakeup0; EXT_WAKEUP_POL_HL[N] controls ext_wakeup n ; Note when enabled EXT_WAKEUP_POL_HL overrides EXT_WAKEUP_POL 1	RW	0x0

Bits	Field Name	Description	Type	Reset
18:17	PWR_ENABLE_SM	Power State Machine state 00 = Idle/Default 01 = Shutdown (ALARM2 and pwr_enable enable is set, note 31uS latency from ALARM2 event.) 10 = Time based wakeup (ALARM status is set) 11 = External event based wakeup (one or more bit set in EXT_WAKEUP_STATUS)	RW	0x0
16	PWR_ENABLE_EN	pwr_enable enable 0: Disable When Disabled, PMIC_PWR_ENABLE will always be drive 1, ON state 1: Enable When enabled: PMIC_PWR_ENABLE will be controlled by ext_wakeup 3:0 , alarm , and alarm2. ON - OFF (Turn OFF) By ALARM2 event OFF - ON (TURN ON) By ALARM event OR ext_wakeup n event 1	RW	0x0
15:12	EXT_WAKEUP_STATUS	External wakeup status 0: External wakeup event has not occurred 1: External wakeup event has occurred Wrt 1 to Clear EXT_WAKEUP_STATUS[0] status of ext_wakeup0 event EXT_WAKEUP_STATUS[N] status of ext_wakeup n event. SW must clear the events before PMIC_PWR_ENABLE can go from 1 - 0. 1	RW	0x0
11:8	EXT_WAKEUP_DB_EN	External wakeup debounce enabled 0: Disable 1: Enable EXT_WAKEUP_DB_EN[0] controls ext_wakeup0; EXT_WAKEUP_DB_EN[N] controls ext_wakeup n ; When enabled RTL_DEBOUNCE_REG defines the debounce time. 1	RW	0x0
7:4	EXT_WAKEUP_POL	External wakeup inputs polarity 0: Active High 1: Active Low EXT_WAKEUP_POL[0] controls ext_wakeup0; EXT_WAKEUP_POL[N] controls ext_wakeup n ; 1	RW	0x0
3:0	EXT_WAKEUP_EN	Enable External wakeup inputs 0: Ext Wakeup disabled 1: Ext Wakeup enable EXT_WAKEUP_EN[0] controls ext_wakeup0; EXT_WAKEUP_EN[N] controls ext_wakeup n ; 1	RW	0x0

1. n=0 to 3

Table 23-79. Register Call Summary for Register RTC_PMIC_REG

RTC Functional Description

- Interrupt Description: [0] [1]
- Register Access: [2]
- Subsystem-Level Power Management — PMIC Mode: [3]

RTC Register Manual

- RTC_SS Register Summary: [4]

Table 23-80. RTC_RTL_DEBOUNCE_REG

Address Offset	0x0000 009C		
Physical Address	0x4883 809C	**Instance**	RTC_SS
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DEBOUNCE_REG							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x0
7:0	DEBOUNCE_REG	Debounce time, see Section 23.4.5 for details.	RW	0x0

Table 23-81. Register Call Summary for Register RTC_RTL_DEBOUNCE_REG

RTC Functional Description

- Register Access: [0]
- Debouncing: [1] [2]

RTC Register Manual

- RTC_SS Register Summary: [3]

Chapter 24

Serial Communication Interfaces

This chapter describes the features and operation of the device serial communication interfaces (SCI).

24.1 Multimaster High-Speed I²C Controller 5316
24.2 HDQ/1-Wire 5382
24.3 UART/IrDA/CIR 5400
24.4 Multichannel Serial Peripheral Interface 5504
24.5 Quad Serial Peripheral Interface 5577
24.6 Multichannel Audio Serial Port 5606
24.7 SuperSpeed USB DRD 5746
24.8 SATA Controller 5761
24.9 PCIe Controller 5827
24.10 DCAN 5998
24.11 Gigabit Ethernet Switch (GMAC_SW) 6093
24.12 Media Local Bus (MLB) 6333

24.1 Multimaster High-Speed I^{2}C Controller

This section describes the high-speed inter-integrated circuit (I^{2}C) controller modules in the device.

Note

Not all I^{2}C instances support HS-mode operation. See the Device Data Manual for details.

24.1.1 HS I^{2}C Overview

The device contains five multimaster high-speed (HS) inter-integrated circuit (I^{2}C) controllers (I2C*i* modules, where *i* = 1, 2 ,3, 4, 5) each of which provides an interface between a local host (LH), such as a digital signal processor (DSP), and any I^{2}C-bus-compatible device that connects through the I^{2}C serial bus. External components attached to the I^{2}C bus can serially transmit and receive up to 8 bits of data to and from the LH device through the 2-wire I^{2}C interface.

Each multimaster HS I^{2}C controller can be configured to act like a slave or master I^{2}C-compatible device.

I2C1 and I2C2 controllers have dedicated I2C compliant open drain buffers, and support Fast mode (up to 400Kbps). I2C3, I2C4 and I2C5 controllers are multiplexed with standard LVCMOS IO and connected to emulate open drain. I^{2}C emulation is achieved by configuring the LVCMOS buffers to output Hi-Z instead of driving high when transmitting logic 1. These controllers support HS mode (up to 3.4Mbps). For the specific IO timing characteristics of the different I^{2}C instances, see the device data manual.

Figure 24-1 shows the I^{2}C.

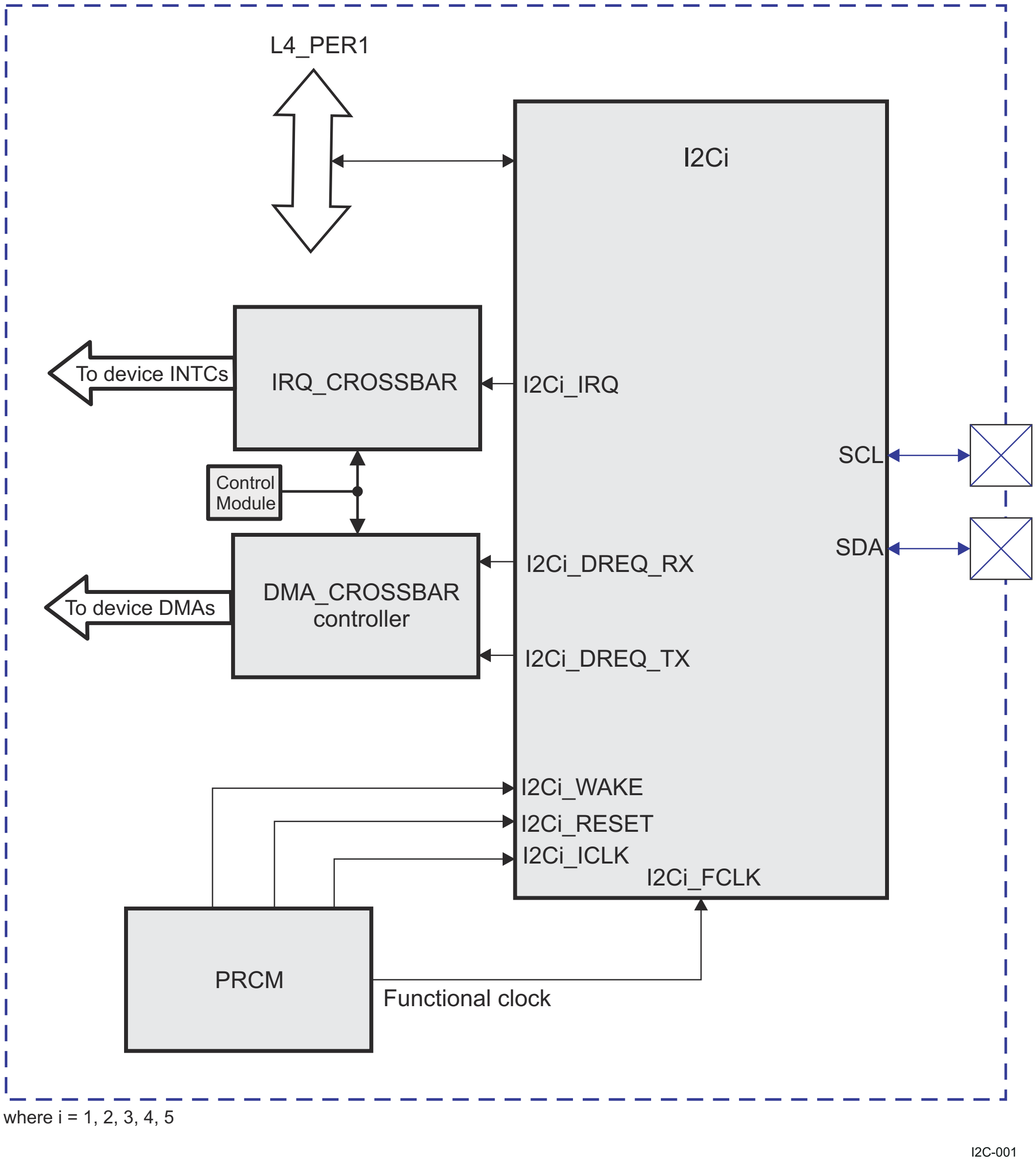


Figure 24-1. HS I²C Controllers

The multimaster HS I²C controllers have the following features:

- Compliant with Philips I²C specification version 2.1
- Supports a standard mode (up to 100 kbps) and fast mode (up to 400 kbps)
- Supports HS mode for transfer up to 3.4 Mbps (only for I2C3, I2C4 and I2C5)
- 7-bit and 10-bit device addressing modes
- General call
- Start/Restart/Stop
- Multimaster transmitter/slave receiver mode
- Multimaster receiver/slave transmitter mode
- Combined master transmit/receive and receive/transmit mode
- Built-in FIFOs (16 bytes) for buffered read or write
- Module enable/disable capability
- Programmable multislave channel (responds to four separate addresses)
- Programmable clock generation

- 8-bit-wide data access
- Designed for low power consumption
- Implement Auto Idle mechanism
- Implement Idle Request/Idle Acknowledge handshake mechanism
- Support for asynchronous wakeup mechanism
- Two direct memory access (DMA) channels
- Wide interrupt capability

24.1.2 HS I²C Environment

This section describes the HS I²C application fields from an environment point of view (external connections). It describes HS I²C connectivity options, lists all possible interfaces, and describes the protocol and data format used in each case.

24.1.2.1 HS I²C Typical Application

Figure 24-2 shows the multimaster HS I²C controllers and their related connections with I²C-compliant devices.

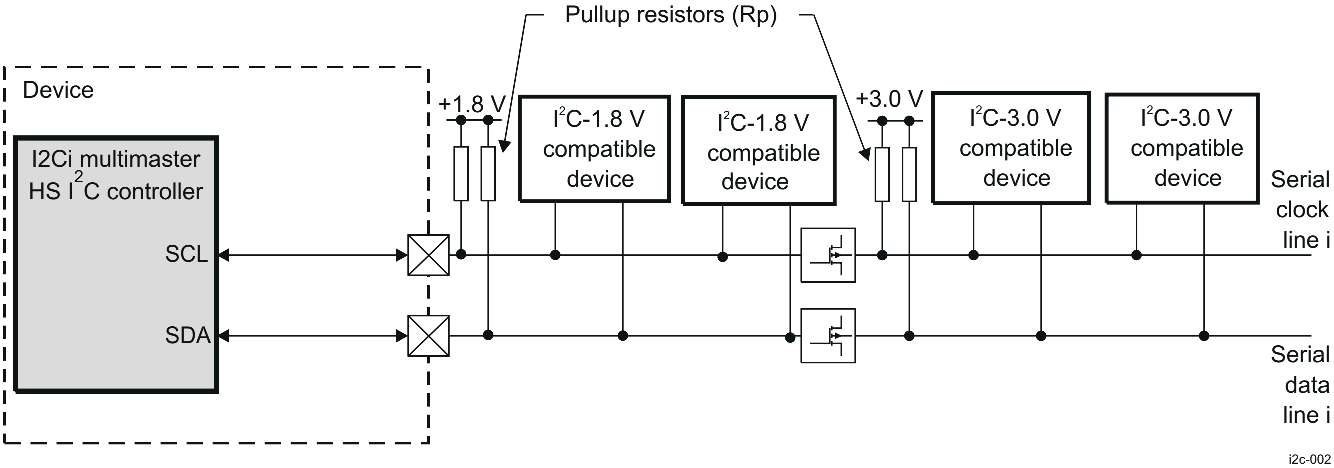


Figure 24-2. HS I²C and Typical Connections to I²C Devices

24.1.2.1.1 HS I²C Pins for Typical Connections in I²C Mode

Figure 24-3 shows the multimaster HS I²C controller pins used for typical connections with I²C devices.

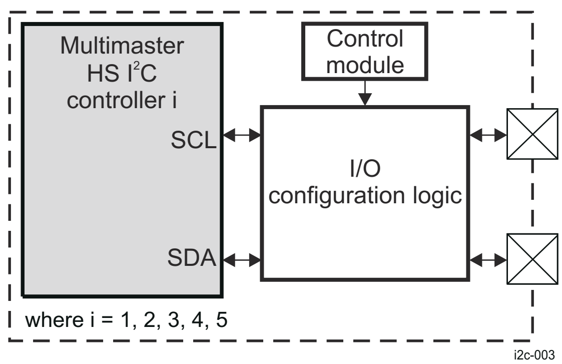


Figure 24-3. HS I²C Interface Signals

24.1.2.1.2 HS I²C Interface Typical Connections

Table 24-1 lists the pins associated with the I²C interface.

Table 24-1. HS I²C Input/Output

Signal	Device Level Signal	I/O(1)	Description	Reset Value
SCL	i2c*j*_scl(2)	I/O	I²C serial clock line. Open-drain output buffer.	1
SDA	i2c*j*_sda(2)	I/O	I²C serial data line. Open-drain output buffer.	1
SCL	i2c*i*_scl(3)	I/O	I²C serial clock line. Emulated open-drain output buffer.	1
SDA	i2c*i*_sda(3)	I/O	I²C serial data line. Emulated open-drain output buffer.	1

(1) I = Input; O = Output
(2) j= 1 and 2
(3) i= 3 to 5

Note

For the i2c*j*_scl and i2c*i*_scl signals to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers should be set to 0x1 because of retiming purposes.

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. Refer to the sections *Pad Functional Multiplexing* of the chapter *Control Module*, for more information.

24.1.2.2 HS I^{2}C Typical Connection Protocol and Data Format

24.1.2.2.1 HS I^{2}C Serial Data Format

The I^{2}C controller operates in 8-bit word data format (byte write access supported for the last access). Each byte transmitted or received on the serial data line is 8 bits long. The number of bytes that can be transmitted or received is not restricted. The data is transferred with the most-significant bit (MSB) first. In receiver mode, each byte is followed by an acknowledge bit from the I^{2}C. Figure 24-4 shows a typical I^{2}C communication format.

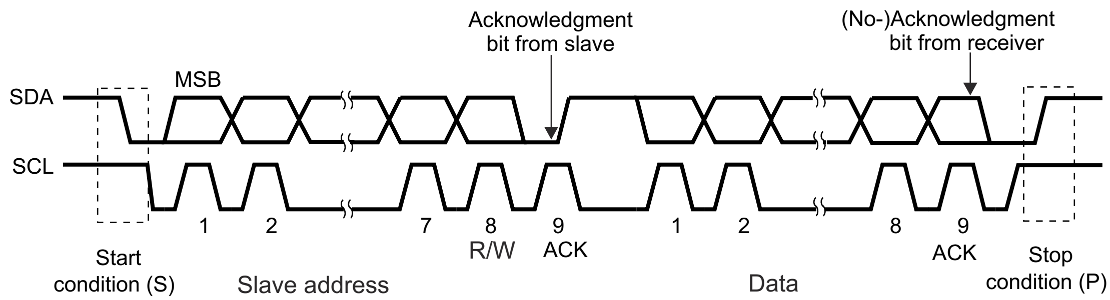


Figure 24-4. HS I^{2}C Data Transfer

24.1.2.2.2 HS I^{2}C Data Validity

The data on the serial data line (SDA) must be stable during the high period of the serial clock line. The high and low states of the data line can change only when the clock signal on the serial clock line (SCL) is low.

Figure 24-5 is an example of data validity requirements.

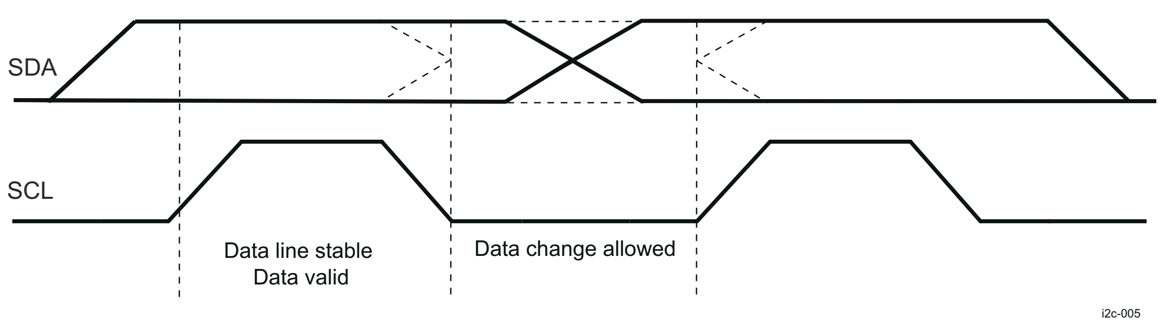


Figure 24-5. HS I^{2}C Bit Transfer on the I^{2}C Bus

24.1.2.2.3 HS I^{2}C Start and Stop Conditions

The I^{2}C module generates start (S) and stop (P) conditions when it is configured as a master.

- An S condition is a high-to-low transition on the serial data line while serial clock line is high.
- A P condition is a low-to-high transition on the serial data line while serial clock line is high.

The bus is considered busy after the S condition (the I2C*i*. [12] BB bit is 1 to indicate that the bus is busy) and free after the P condition (the I2C*i*.I2C_IRQSTATUS_RAW [12] BB bit is 0 to indicate that the bus is free).

Figure 24-6 shows the waveforms that occur during an S and a P condition.

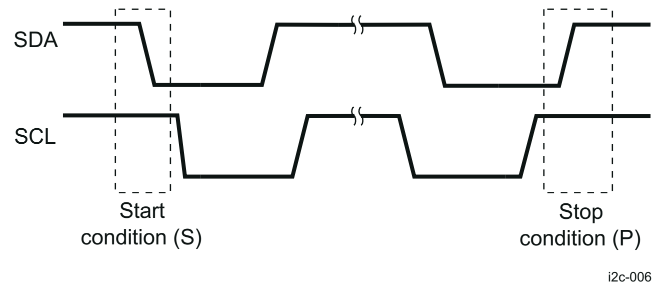


Figure 24-6. HS I²C S and P Condition Events

Note

I²C controller does not support messages non-compliant with I2C standard. Void messages are non-standard I²C messages and will lockup the controller. A void message is a START condition followed by a STOP condition, in other words, while the bus is free the SDA line is pulled low (START) and then released (STOP). This would result in a timeout (software) of the next master transfer which would never complete. A soft reset of the controller is recommended for recovery.

24.1.2.2.4 HS I²C Addressing

The I²C module supports two data formats in fast/standard (F/S) and HS modes:

- 7-bit/10-bit addressing format
- 7-bit/10-bit addressing format with repeated start (Sr) condition

24.1.2.2.4.1 Data Transfer Formats in F/S Mode

Figure 24-7 shows the I²C data transfer formats in F/S mode.

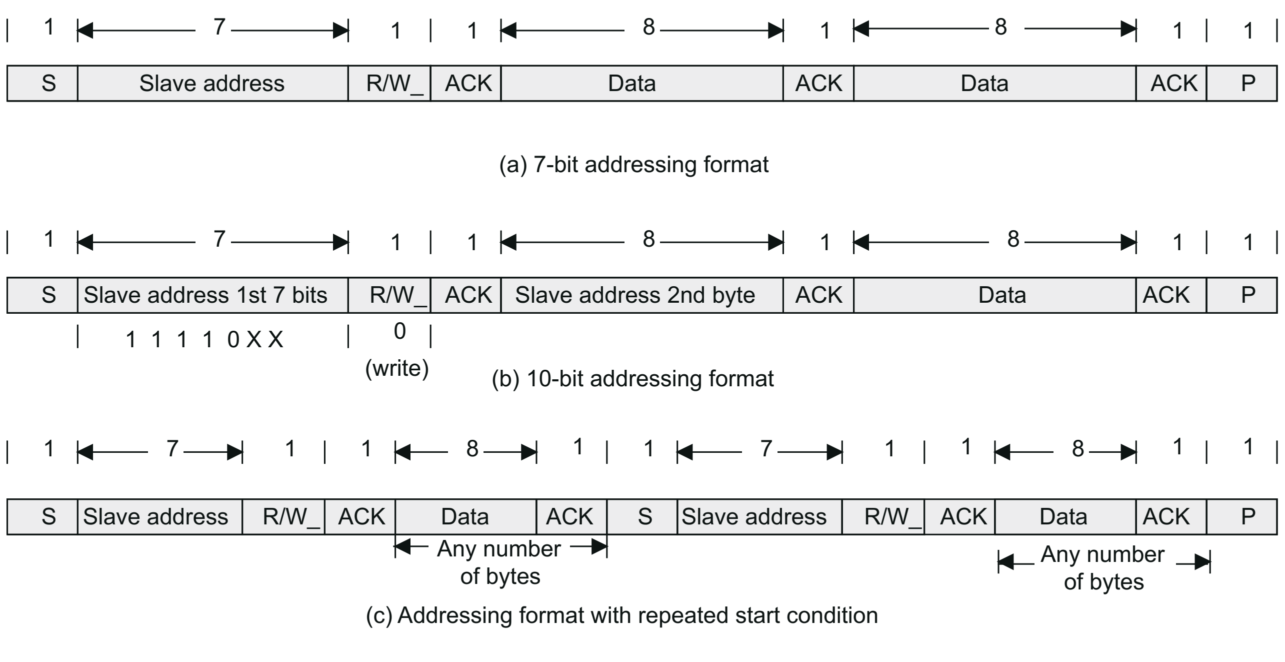


Figure 24-7. HS I²C Data Transfer Formats in F/S Mode

The first word after an S condition consists of 8 bits. In acknowledge mode, an extra dedicated acknowledgment bit is inserted after each byte.

In addressing formats with 7-bit addresses, the first byte is composed of 7 MSB slave address bits and 1 least-significant bit (LSB) R/W_ bit.

The LSB R/W_ bit of the address byte indicates the transmission direction of the data bytes that follow it. If R/W_ is 0, the master writes data to the selected slave; if it is 1, the master reads data from the slave.

In addressing formats with 10-bit addresses, the structure of the first byte is 11110XXY, where XX is the two MSBs of the 10-bit addresses, and Y is the R/W_ bit. If the R/W_ bit is 0, the next byte contains the last 8 bits of the slave address. If the R/W_ bit is 1, the next byte contains data transmitted from the slave to the master.

24.1.2.2.4.2 Data Transfer Format in HS Mode

Figure 24-8 shows the I^2C data transfer format in HS mode.

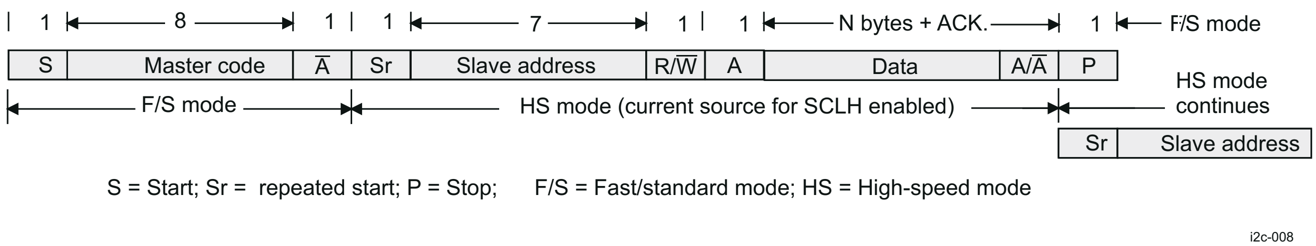


Figure 24-8. HS I^2C Data Transfer in HS Mode

Each multimaster HS I^2C controller can also operate in HS mode. In this case, after the S condition, the module, which is in F/S mode, writes the master code address (000001XXX, where XXX is the variable portion of the master code) on the bus. No device connected on the same bus acknowledges this address. The module switches the clock to the HS clock and after an Sr condition, and sends the slave address and the data, as shown in Figure 24-8.

24.1.2.2.5 HS I^2C Master Transmitter

In master transmitter mode, data assembled in one of the previously described data formats is shifted out on the serial data line SDA in sync with the self-generated clock pulses on the serial clock line SCL. The clock pulses are inhibited and SCL is held low when the intervention of the processor is required (XUDF) after a byte is transmitted.

24.1.2.2.6 HS I^2C Master Receiver

Master receiver mode can be entered only from master transmitter mode. With any of the address formats (a), (b), or (c) (see Figure 24-7), if R/W_ is high, the module enters master receiver mode after the slave address byte and bit R/W_ are transmitted. Serial data bits received on bus line SDAI are shifted in synchronization with the self-generated clock pulses on SCLI.

24.1.2.2.7 HS I^2C Slave Transmitter

Slave transmitter mode can be entered only from slave receiver mode. With any of the address formats (a), (b), or (c) (see Figure 24-7), the slave transmitter is entered if the slave address byte is the same as its own address and bit R/W_ is transmitted, if R/W_ is high. The slave transmitter shifts the serial data out on the data line SDA in sync with the clock pulses that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the LH is required (XUDF).

24.1.2.2.8 HS I^2C Slave Receiver

In this mode, serial data bits received on the bus line SDA are shifted-in in sync with the clock pulses on SCL that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the LH is required (ROVR) after a byte is received.

24.1.2.2.9 HS I²C Bus Arbitration

If two or more master transmitters start a transmission on the same bus almost simultaneously, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial bus by the competing transmitters. When a transmitter senses that a high signal it has presented on the bus has been overruled by a low signal, it switches to the slave receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration lost interrupt. Figure 24-9 shows the arbitration procedure between two devices. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. If two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

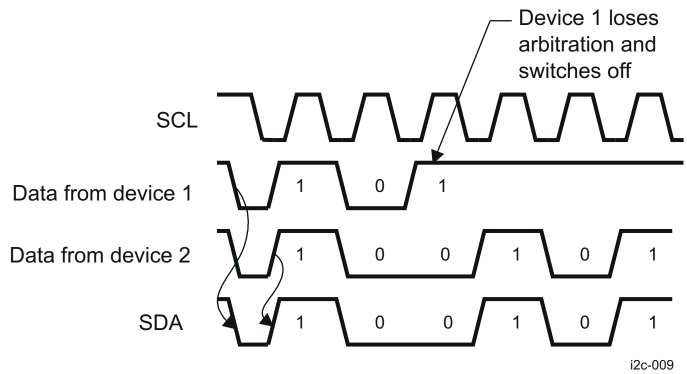


Figure 24-9. HS I²C Arbitration Between Master Transmitters

24.1.2.2.10 HS I²C Clock Generation and Synchronization

Under normal conditions, only one master device generates the clock signal, SCL. However, there are two or more master devices during the arbitration procedure, and the clock must be synchronized so that the data output can be compared. The wired-AND property of the clock line means that a device that first generates a low period of the clock line overrules the other devices. At this high/low transition, the clock generators of the other devices are forced to start generation of their own low period. The clock line is then held low by the device with the longest low period, while the other devices that finish their low periods must wait for the clock line to be released before starting their high periods. A synchronized signal on the clock line is thus obtained, where the slowest device determines the length of the low period and the fastest device determines the length of the high period. If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the WAIT-state. In this way a slave can slow down a fast master and the slow device can create enough time to store a received byte or prepare a byte to be transmitted (clock stretching).

Note

In case the SCL or SDA lines are stuck low, a bus clearing operation is supported:

- If the clock line (SCL) is stuck low, the preferential procedure is to reset the bus using the hardware reset signal if your I²C devices have hardware reset inputs. If the I²C devices do not have hardware reset inputs, cycle power to the devices to activate the mandatory internal power-on reset (POR) circuit.
- If the data line (SDA) is stuck low, the master should send nine clock pulses. The device that held the bus low should release it sometime within those nine clocks. If not, then use the hardware reset or cycle power to clear the bus.

Figure 24-10 shows clock synchronization.

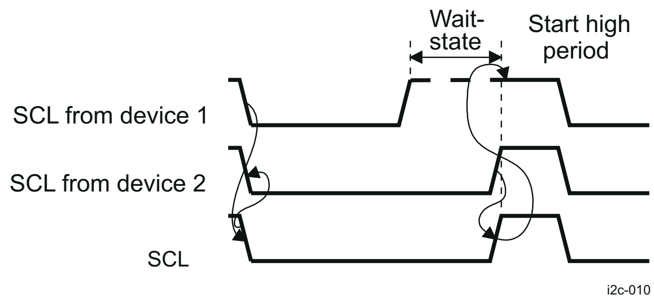


Figure 24-10. HS I^{2}C Clock Generators Synchronization

24.1.3 HS I²C Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 24-11 shows the integration of the five HS I²C controllers in the device.

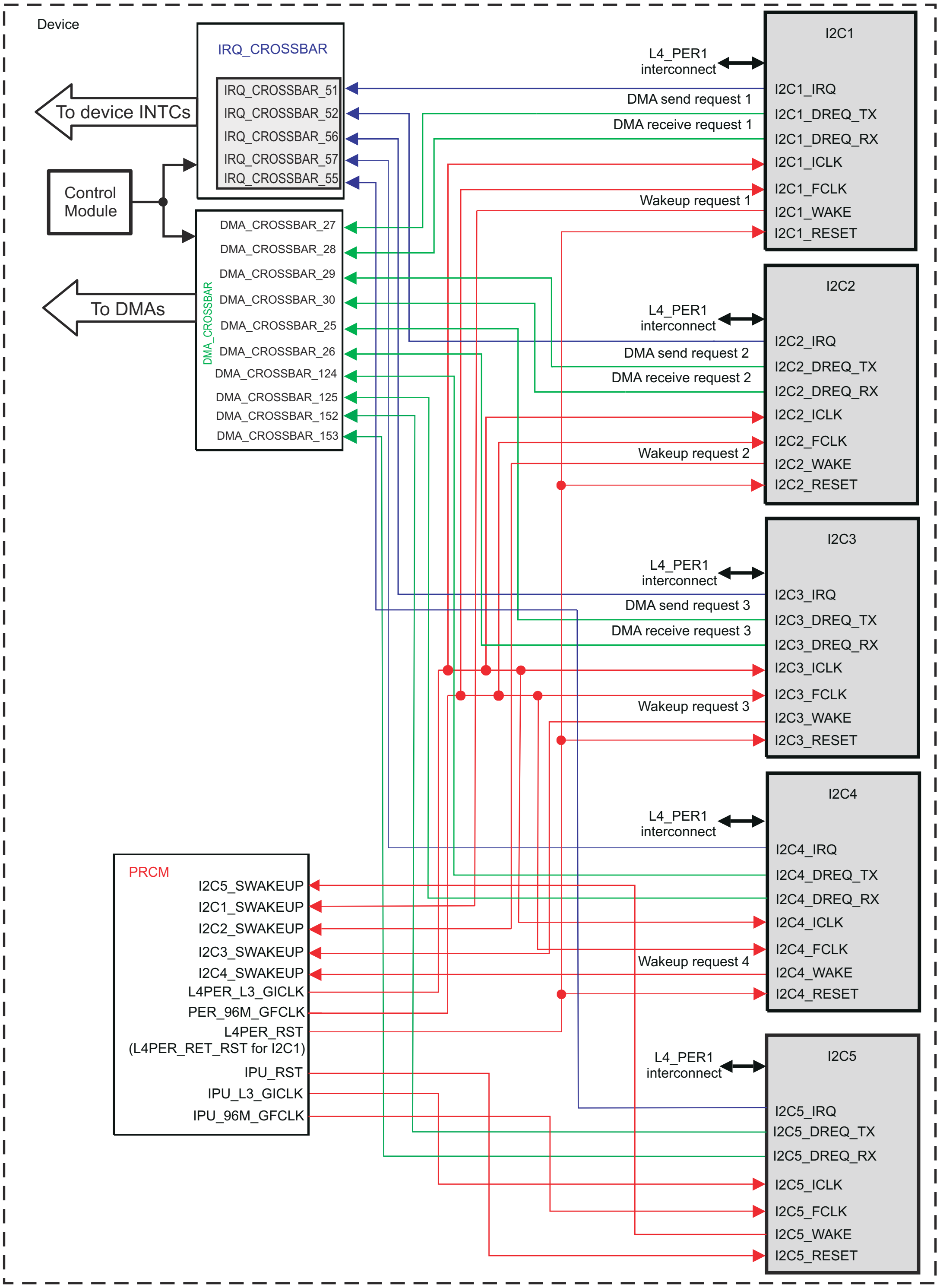


Figure 24-11. HS I²C Integration

Note

For more information about the slave idle protocol and the wake-up request, see *Power Management Functional Description*, in *Power, Reset, and Clock Management*.

Table 24-2 through Table 24-4 summarize the integration of the module in the device.

Table 24-2. HS I²C Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
I2C1	PD_COREAON	L4_PER1
I2C2	PD_COREAON	L4_PER1
I2C3	PD_COREAON	L4_PER1
I2C4	PD_COREAON	L4_PER1
I2C5	PD_COREAON	L4_PER1

Table 24-3. HS I²C Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
I2C1	I2C1_ICLK	L4PER_L3_GICLK	PRCM	I2C1 interface clock
	I2C1_FCLK	PER_96M_GFCLK	PRCM	I2C1 functional clock
I2C2	I2C2_ICLK	L4PER_L3_GICLK	PRCM	I2C2 interface clock
	I2C2_FCLK	PER_96M_GFCLK	PRCM	I2C2 functional clock
I2C3	I2C3_ICLK	L4PER_L3_GICLK	PRCM	I2C3 interface clock
	I2C3_FCLK	PER_96M_GFCLK	PRCM	I2C3 functional clock
I2C4	I2C4_ICLK	L4PER_L3_GICLK	PRCM	I2C4 interface clock
	I2C4_FCLK	PER_96M_GFCLK	PRCM	I2C4 functional clock
I2C5	I2C5_ICLK	IPU_L3_GICLK	PRCM	I2C5 interface clock
	I2C5_FCLK	IPU_96M_GFCLK	PRCM	I2C5 functional clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
I2C1	I2C1_RESET	L4PER_RET_RST	PRCM	I2C1 reset
I2C2	I2C2_RESET	L4PER_RST	PRCM	I2C2 reset
I2C3	I2C3_RESET	L4PER_RST	PRCM	I2C3 reset
I2C4	I2C4_RESET	L4PER_RST	PRCM	I2C4 reset
I2C5	I2C5_RESET	IPU_RST	PRCM	I2C5 reset

Table 24-4. HS I²C Hardware Requests

Interrupt Requests				
Module Instance	Source Signal Name	IRQ_CROSSBAR Input	Default Mapping	Description
I2C1	I2C1_IRQ	IRQ_CROSSBAR_51	MPU_IRQ_56	I2C1 interrupt request
			DSP1_IRQ_82	
			DSP2_IRQ_82	
			IPU1_IRQ_41	
			IPU2_IRQ_41	
I2C2	I2C2_IRQ	IRQ_CROSSBAR_52	MPU_IRQ_57	I2C2 interrupt request
			DSP1_IRQ_83	
			DSP2_IRQ_83	

Table 24-4. HS I²C Hardware Requests (continued)

			IPU1_IRQ_42	
			IPU2_IRQ_42	
I2C3	I2C3_IRQ	IRQ_CROSSBAR_56	MPU_IRQ_61	I2C3 interrupt request
			DSP1_IRQ_87	
			DSP2_IRQ_87	
			IPU1_IRQ_43	
			IPU2_IRQ_43	
I2C4	I2C4_IRQ	IRQ_CROSSBAR_57	MPU_IRQ_62	I2C4 interrupt request
			DSP1_IRQ_88	
			DSP2_IRQ_88	
			IPU1_IRQ_44	
			IPU2_IRQ_44	
I2C5	I2C5_IRQ	IRQ_CROSSBAR_55	MPU_IRQ_60	I2C5 interrupt request
			DSP1_IRQ_86	
			DSP2_IRQ_86	
		DMA Requests		
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
I2C1	I2C1_DREQ_TX	DMA_CROSSBAR_27	DMA_EDMA_DREQ_26	I2C1 DMA transmit request
			DMA_SYSTEM_DREQ_26	
	I2C1_DREQ_RX	DMA_CROSSBAR_28	DMA_EDMA_DREQ_27	I2C1 DMA receive request
			DMA_SYSTEM_DREQ_27	
I2C2	I2C2_DREQ_TX	DMA_CROSSBAR_29	DMA_EDMA_DREQ_28	I2C2 DMA transmit request
			DMA_SYSTEM_DREQ_28	
	I2C2_DREQ_RX	DMA_CROSSBAR_30	DMA_EDMA_DREQ_29	I2C2 DMA receive request
			DMA_SYSTEM_DREQ_29	
I2C3	I2C3_DREQ_TX	DMA_CROSSBAR_25	DMA_EDMA_DREQ_24	I2C3 DMA transmit request
			DMA_SYSTEM_DREQ_24	
	I2C3_DREQ_RX	DMA_CROSSBAR_26	DMA_EDMA_DREQ_25	I2C3 DMA receive request
			DMA_SYSTEM_DREQ_25	
I2C4	I2C4_DREQ_TX	DMA_CROSSBAR_124	DMA_SYSTEM_DREQ_123	I2C4 DMA transmit request
	I2C4_DREQ_RX	DMA_CROSSBAR_125	DMA_SYSTEM_DREQ_124	I2C4 DMA receive request
I2C5	I2C5_DREQ_TX	DMA_CROSSBAR_152	-	I2C5 DMA transmit request
	I2C5_DREQ_RX	DMA_CROSSBAR_153	-	I2C5 DMA receive request

Note

The **Default Mapping** column in Table 24-4 *HS I^2C Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.
For more information about the device DMA_SYSTEM module, see *System DMA*.
For more information about the device EDMA module, see *Enhanced DMA*.

Note

- For a description of interrupt source, see Table 24-10.
- For a description of DMA source, see Table 24-11.

24.1.4 HS I²C Functional Description

24.1.4.1 HS I²C Block Diagram

Figure 24-12 is the multimaster I²C HS controller block diagram.

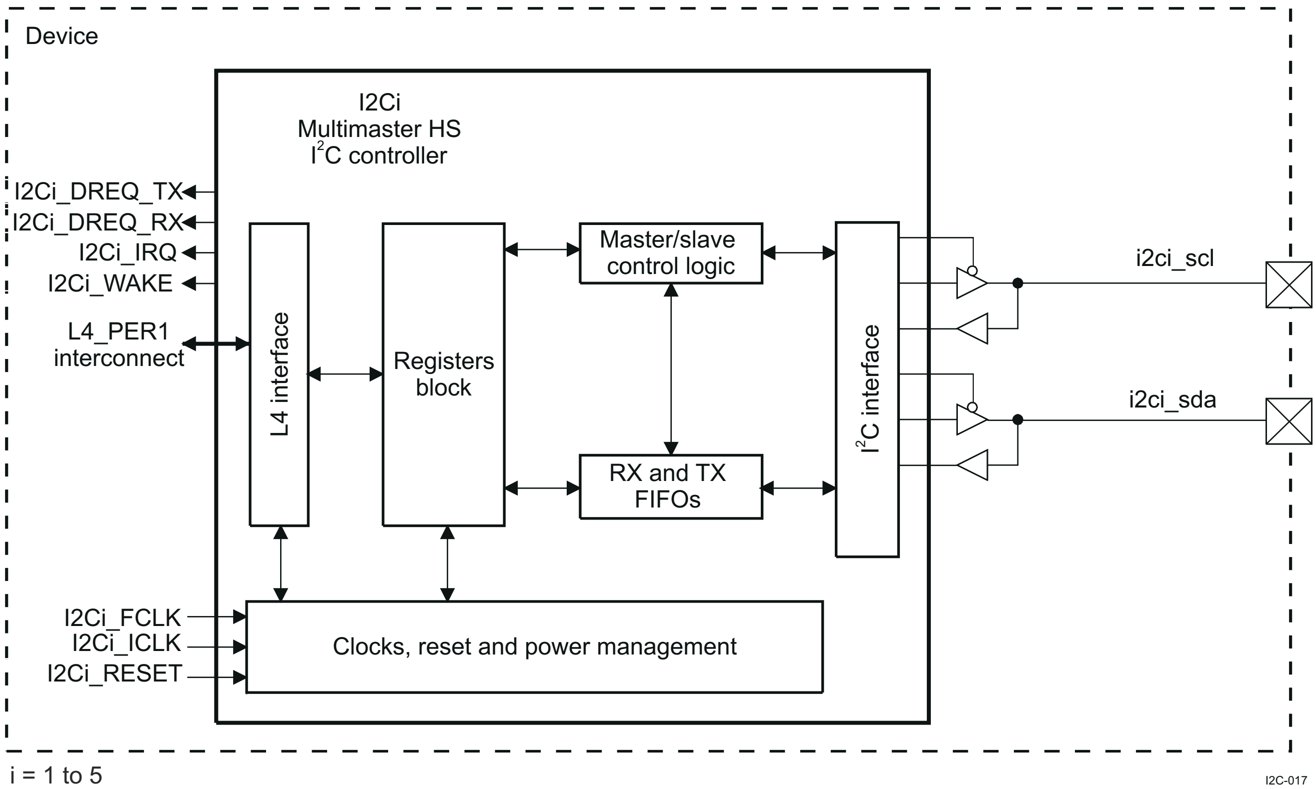


Figure 24-12. HS I²C Block Diagram

The five multimaster HS I²C controllers can be configured in F/S I²C mode or HS I²C mode. The operation mode is selected by configuring the I2C*i*.I2C_CON[13:12] OPMODE bit field. Table 24-5 lists the available operation modes.

Table 24-5. HS I²C Operation Mode Selection

Operation Mode	Value of I2C*i*.I2C_CON[13:12] OPMODE
F/S I²C	0x0
HS I²C	0x1
Reserved	0x2
Reserved (not used)	0x3

24.1.4.2 HS I²C Clocks

24.1.4.2.1 HS I²C Clocking

Figure 24-13 shows the I²C clock generation of the HS I²C controllers.

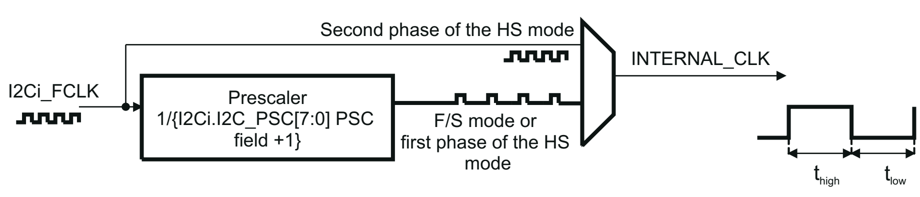


Figure 24-13. HS I²C Clock Generation

Each multimaster HS I^2C controller uses the I2C*i*_FCLK functional clock in the PRCM module. The internal sampling clock I2C*i*_INTERNAL_CLK is generated by dividing the functional clock by the I2C*i*.I2C_PSC[7:0] PSC bit field value + 1 in F/S mode, or in the first phase of HS mode; or by directly using the functional clock in the second phase of HS mode (prescaler is bypassed).

The low time of the I2C*i*_SCLL signal is determined by the I2C*i*.I2C_SCLL[7:0] SCLL bit field in F/S mode and in the first phase of HS mode; or by the I2C*i*.I2C_SCLL[15:8] HSSCLL bit field in the second phase of HS mode.

The high time of the I2C*i*_SCLL signal is determined by the I2C*i*.I2C_SCLH[7:0] SCLH bit field in F/S mode and in the first phase of HS mode; or by the I2C*i*.I2C_SCLH[15:8] HSSCLH bit field in the second phase of HS mode.

Table 24-6 lists the t_{LOW} and t_{high} values in master mode only (in slave mode, the I^2C controller does not generate the I^2C clock).

Table 24-6. HS I^2C t_{LOW} and t_{high} Values of the I^2C Clock

Mode	I2C*i* Clock	t_{LOW}	t_{high}
F/S or HS first phase	I2C*i*_INTERNAL_CLK = I2C*i*_FCLK / (I2C*i*.I2C_PSC[7:0] PSC bit field + 1)	(I2C*i*.I2C_SCLL[7:0] SCLL bit field value + 7) x I2C*i*_INTERNAL_CLK period	(I2Ci.I2C_SCLH [7:0] SCLH bit filed value + 5) x I2Ci_INTERNAL_CLK period
HS second phase	I2C*i*_FCLK	(I2C*i*.I2C_SCLL[15:8] HSSCLL bit field value + 7) x I2C*i*_FCLK period	(I2Ci.I2CSCLL [15:8] HSSCLH bit field value + 5) x I2Ci_FCLK period

Note

For HS mode, the I2C*i*.I2C_SCLL[15:8] HSSCLL and I2C*i*.I2C_SCLL[7:0] SCLL bit fields must be programmed (the first phase of an HS transaction is performed at F/S speed).

For HS mode, the I2C*i*.I2C_SCLH[15:8] HSSCLH and I2C*i*.I2C_SCLH[7:0] SCLH bit fields must be programmed (the first phase of an HS transaction is performed at F/S speed).

Note

The equations in Table 24-6 give the SCLL timing values for SCLL/SCLH/HSSCLL/HSSCLH at HS I^2C controller outputs. Actual tlow and thigh periods may vary depending on the board (the load capacitance on the SCLL signal). If necessary, any adjustments to the SCLL/SCLH/HSSCLL/HSSCLH values must be determined by measurements of actual SCL signal on the board.

CAUTION

During active mode (the I2C*i*.I2C_CON[15] I2C_EN bit is set to 1), make no changes to the I2C*i*.I2C_SCLL and I2C*i*.I2C_SCLH registers. Changes may result in unpredictable behavior.

Table 24-7 lists the register values for obtaining the maximum I^2C bit rates and the maximum period of the filtered spikes in F/S mode and HS mode.

Table 24-7. HS I^2C Register Values for Maximum I^2C Bit Rates in I^2C F/S, I^2C HS Modes

[1]	I^2C Mode for I2C*i*,			Description
	Standard Mode	Fast Mode	High-Speed Mode[2]	
I2C*i*_FCLK frequency (MHz)	96			

Table 24-7. HS I²C Register Values for Maximum I²C Bit Rates in I²C F/S, I²C HS Modes (continued)

[1]	I²C Mode for I2C*i*,			Description
I2C *i* .I2C_PSC[7:0] PSC	**23**	**9**	**1**	Prescaler value for F/S and HS modes
I2C*i*_INTERNAL_CLK frequency (MHz)	4	9.6	96	
I2C *i* .I2C_SCLL[7:0] SCLL	**13**	**5**	**115**	Value for F/S mode and first phase of HS mode
I2C *i* .I2C_SCLH[7:0] SCLH	**15**	**7**	**113**	Value for F/S mode and first phase of HS mode
Maximum bit rate (Mbps)	0.1	0.4	0.4	F/S mode and first phase in HS mode maximum bit rate
Maximum filter period (ns)	250	104.2	10	
I2C *i* .I2C_SCLL[15:8] HSSCLL			**12**	Values for second phase of HS mode
I2C *i* .I2C_SCLH[15:8] HSSCLH			**5**	Values for second phase of HS mode
HS mode maximum bit rate (Mbps)			3.31	HS mode maximum bit rate
Maximum filter period (ns)			10	

(1) Programmable fields are in bold.
(2) Supported only on I2C3, I2C4 and I2C5.

Note

This table presents informative values only for the configuration parameters and the I²C bus performance obtained according to these values. The delays added by the analog pads are not considered in these figures.

Note

For I2Ci (where i=1, 2, 3, 4, 5)

I2Ci_INTERNAL_CLK freq = I2Ci_FCLK / (PSC +1)

F/S filter period = 1 / I2Ci_INTERNAL_CLK

HS filter period = 1 / I2Ci_FCLK freq

HS bit rate = I2Ci_FCLK freq / (HSSCLL+ 7 + HSSCLH + 5)

FS bit rate = I2Ci_INTERNAL_CLK / (SCLL+ 7 + SCLH + 5)

24.1.4.2.2 HS I²C Automatic Blocking of the I²C Clock Feature

This feature offers the possibility for the LH to command the blocking of the I²C clock after the slave addressing phase, when the I²C controller is addressed by an external master device using a certain Own Address.

The release of the I²C clock can be performed independently for each Own Address (I2C*i*.I2C_OA, and I2C*i*.I2C_OAx registers, where *i* = 1 to 5, x = 1, 2, 3) by deasserting the corresponding bit in the I2C*i*.I2C_SBLOCK register.

24.1.4.3 HS I²C Software Reset

Each multimaster HS I²C controller supports the software reset by accessing the I2C*i*.I2C_SYSC[1] SRST bit (1: reset; 0: normal mode).

The software reset status can be checked by accessing the I2C*i*.I2C_SYSS[0] RDONE bit (1: reset is done; 0: reset is ongoing).

To do a software reset, the following steps must be done:

1. Ensure that the module is disabled (clear the I2C*i*.I2C_CON[15] I2C_EN bit to 0).
2. Set the I2C*i*.I2C_SYSC[1] SRST bit to 1.
3. Enable the module by setting I2C*i*.I2C_CON[15] I2C_EN bit to 1.
4. Check the I2C*i*.I2C_SYSS[0] RDONE bit until it is set to 1 to indicate the software reset is complete.

Note

The I2C*i*.I2C_CON[15] I2C_EN bit can hold the functional clock domain of the multimaster HS I^{2}C controller in reset after the device reset has been released. When the system bus reset is removed, this bit remains cleared. The functional part of the I^{2}C controller is held in reset state while this bit is 0, and all configuration registers can be accessed.

The I2C*i*.I2C_CON[15] I2C_EN bit must be set to 1 to enable the functional part of the I^{2}C controller.

The I2C*i*.I2C_SYSS[0] RDONE bit is asserted only after the module is enabled by setting the I2C*i*.I2C_CON[15] I2C_EN bit to 1.

24.1.4.4 HS I^{2}C Power Management

Table 24-8 describes power-management features available for the multimaster HS I^{2}C controllers.

Note

- For information about source clock gating and sleep/wake-up transitions description, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.
- For descriptions of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

Table 24-8. HS I^{2}C Local Power-Management Features

Feature	Registers	Description
Clock auto gating	I2C*i*.I2C_SYSC[0] AUTOIDLE	This bit allows a local power optimization inside the module.
Slave idle modes	I2C*i*.I2C_SYSC[4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available.
Clock activity	I2C*i*.I2C_SYSC[9:8] CLOCKACTIVITY	For configuration details, see Table 24-9.
Global wake-up enable	I2C*i*.I2C_SYSC[2] ENAWAKEUP	This bit enables the wake-up feature at module level.

Note

The voltage controllers, in which the HS I^{2}C controller is implemented, have no idle request/ acknowledge mechanism. The idle modes for the voltage controllers are directly managed by the PRCM module.

Table 24-9. HS I^{2}C Clock Activity Settings

I2C*i*.I2C_SYSC[9:8] CLOCKACTIVITY	Clock State When Module is in IDLE State		Features Available/Unavailable When Module is in IDLE State
	I2C*i*_ICLK	I2C*i*_ FCLK	
00	OFF	OFF	Both clocks are disabled.
10	OFF	ON	Interface clock is disabled; functional clock is enabled
01	ON	OFF	Functional clock is disabled; interface clock is enabled
11	ON	ON	Both clocks are enabled.

CAUTION

The PRCM module has no hardware means of reading the settings of CLOCKACTIVITY. Thus, software must ensure consistent programming between the I2C CLOCKACTIVITY and I2C clock PRCM control bits. For a description of the ClockActivity feature, see *Module-Level Clock Management* in *Power, Reset, and Clock Management*.

24.1.4.5 HS I2C Interrupt Requests

Table 24-10 lists the event flags, and their mask, that can cause module interrupts.

Table 24-10. HS I2C Events

Event Flag	Event Unmask	Event Mask	Description
I2C*i*.I2C_IRQSTATUS[0] AL	I2C*i*.I2C_IRQENABLE_SET[0] AL_IE	I2C*i*.I2C_IRQENABLE_CLR [0] AL_IE	Arbitration lost. This bit is automatically set by the hardware when it loses the arbitration in master transmit mode, an interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[1] NACK	I2C*i*.I2C_IRQENABLE_SET[1] NACK_IE	I2C*i*.I2C_IRQENABLE_CLR [1] NACK_IE	No acknowledgement. Bit is set when No Acknowledge is received, an interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[2] ARDY	I2C*i*.I2C_IRQENABLE_SET[2] ARDY_IE	I2C*i*.I2C_IRQENABLE_CLR [2] ARDY_IE	Register access ready. When set to 1 it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[3] RRDY	I2C*i*.I2C_IRQENABLE_SET[3] RRDY_IE	I2C*i*.I2C_IRQENABLE_CLR [3] RRDY_IE	Receive data ready. Set to 1 by core when in receiver mode, a new data can be read. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[4] XRDY	I2C*i*.I2C_IRQENABLE_SET[4] XRDY_IE	I2C*i*.I2C_IRQENABLE_CLR [4] XRDY_IE	Transmit data ready. Set to 1 by core when transmitter is ready for new data. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[5] GC	I2C*i*.I2C_IRQENABLE_SET[5] GC_IE	I2C*i*.I2C_IRQENABLE_CLR [5] GC_IE	General call. Set to 1 by core when General Call address was detected. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[6] STC	I2C*i*.I2C_IRQENABLE_SET[6] STC_IE	I2C*i*.I2C_IRQENABLE_CLR [6] STC_IE	Start condition detected. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[7] AERR	I2C*i*.I2C_IRQENABLE_SET[7] AERR_IE	I2C*i*.I2C_IRQENABLE_CLR [7] AERR_IE	Bus Access Error. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[8] BF	I2C*i*.I2C_IRQENABLE_SET[8] BF_IE	I2C*i*.I2C_IRQENABLE_CLR [8] BF_IE	Bus free. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[9] AAS	I2C*i*.I2C_IRQENABLE_SET[9] AAS_IE	I2C*i*.I2C_IRQENABLE_CLR [9] AAS_IE	Address recognized as slave. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[10] XUDF	I2C*i*.I2C_IRQENABLE_SET [10] XUDF_IE	I2C*i*.I2C_IRQENABLE_CLR [10] XUDF_IE	Transmit underflow. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[11] ROVR	I2C*i*.I2C_IRQENABLE_SET [11] ROVR_IE	I2C*i*.I2C_IRQENABLE_CLR [11] ROVR_IE	Receive overrun. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[12] BB	N/A	N/A	Bus busy indicator
I2C*i*.I2C_IRQSTATUS[13] RDR	I2C*i*.I2C_IRQENABLE_SET [13] RDR_IE	I2C*i*.I2C_IRQENABLE_CLR [13] RDR_IE	Receive draining. An interrupt is signaled to the host.
I2C*i*.I2C_IRQSTATUS[14] XDR	I2C*i*.I2C_IRQENABLE_SET [14] XDR_IE	I2C*i*.I2C_IRQENABLE_CLR [14] XDR_IE	Transmit draining. An interrupt is signaled to the host.

24.1.4.6 HS I2C DMA Requests

Each multimaster HS I2C controller can generate two DMA requests to the device DMA controllers through the DMA_CROSSBAR module. Table 24-11 lists the DMA requests. For information about DMA generation, see Section 24.1.4.8.3, *HS I2C FIFO DMA Mode (I2C Mode Only)*.

Table 24-11. HS I2C DMA Requests

Name	Source	Description
I2C1_DREQ_TX	I2C1	I2C1 DMA write request to inform the DMAs to write new data in the I2C1.I2C_DATA[7:0] DATA bit field
I2C1_DREQ_RX	I2C1	I2C1 DMA read request to inform the DMAs to read the data in the I2C1.I2C_DATA[7:0] DATA bit field
I2C2_DREQ_TX	I2C2	I2C2 DMA write request to inform the DMAs to write new data in the I2C2.I2C_DATA[7:0] DATA bit field
I2C2_DREQ_RX	I2C2	I2C2 DMA read request to inform the DMAs to read the data in the I2C2.I2C_DATA[7:0] DATA bit field
I2C3_DREQ_TX	I2C3	I2C3 DMA write request to inform the DMAs to write new data in the I2C3.I2C_DATA[7:0] DATA bit field
I2C3_DREQ_RX	I2C3	I2C3 DMA read request to inform the DMAs to read the data in the I2C3.I2C_DATA[7:0] DATA bit field
I2C4_DREQ_TX	I2C4	I2C4 DMA write request to inform the DMAs to write new data in the I2C4.I2C_DATA[7:0] DATA bit field
I2C4_DREQ_RX	I2C4	I2C4 DMA read request to inform the DMAs to read the data in the I2C4.I2C_DATA[7:0] DATA bit field
I2C5_DREQ_TX	I2C5	I2C5 DMA write request to inform the DMAs to write new data in the I2C5.I2C_DATA[7:0] DATA bit field
I2C5_DREQ_RX	I2C5	I2C5 DMA read request to inform the DMAs to read the data in the I2C5.I2C_DATA[7:0] DATA bit field

Note

For more information about I2C*i*_DREQ_TX and I2C*i*_DREQ_RX (where *i* = 1 to 5)signals mapping to DMA_CROSSBAR, see Section 16.1.3.2 Section 16.1.3.2, *Mapping of DMA Requests to DMA_CROSSBAR Inputs*, in Table 24-4, *HS I2C Hardware Requests*.

24.1.4.7 HS I2C Programmable Multislave Channel Feature

This feature allows each multimaster HS I2C controller to be addressed using four separate Own Addresses configured in the I2C*i*.I2C_OA and I2C*i*.I2C_OAx registers (where = 1 to 5, x = 1, 2, 3). An additional register (I2C*i*.I2C_ACTOA) is used to indicate to the LH which address is used by the external master to communicate with the I2C controller.

Each Own Address can be independently configured in 7-bit or 10-bit mode by setting the corresponding bit (I2C*i*.I2C_CON[7] XOA0, I2C*i*.I2C_CON[6] XOA1, I2C*i*.I2C_CON[5] XOA2, or I2C*i*.I2C_CON[4] XOA3).

24.1.4.8 HS I2C FIFO Management

Each multimaster HS I2C controller implements two internal 8-bit FIFOs, the RX and TX FIFOs.

The depth of the RX and TX FIFOs can be checked by reading the I2C*i*.I2C_BUFSTAT[15:14] FIFODEPTH bit field (0x0: 8 bytes, 0x1: 16 bytes, 0x2: 32 bytes, and 0x3: 64 bytes).

24.1.4.8.1 HS I2C FIFO Interrupt Mode

In FIFO interrupt mode (relevant interrupts enabled by the I2C*i*.I2C_IRQENABLE_SET register), an interrupt signal informs the processor of the receiver and transmitter status. These interrupts are raised when the RX/TX FIFO thresholds (defined by the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1 for the RX FIFO or the I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1 for the TX FIFO) are reached; the interrupt signals instruct the

LH to transfer data to the destination (from the I²C controller in receive mode and/or from any source to the I²C controller FIFO in transmit mode).

Figure 24-14 and Figure 24-15 show receive and transmit operations, respectively, from a FIFO management point of view.

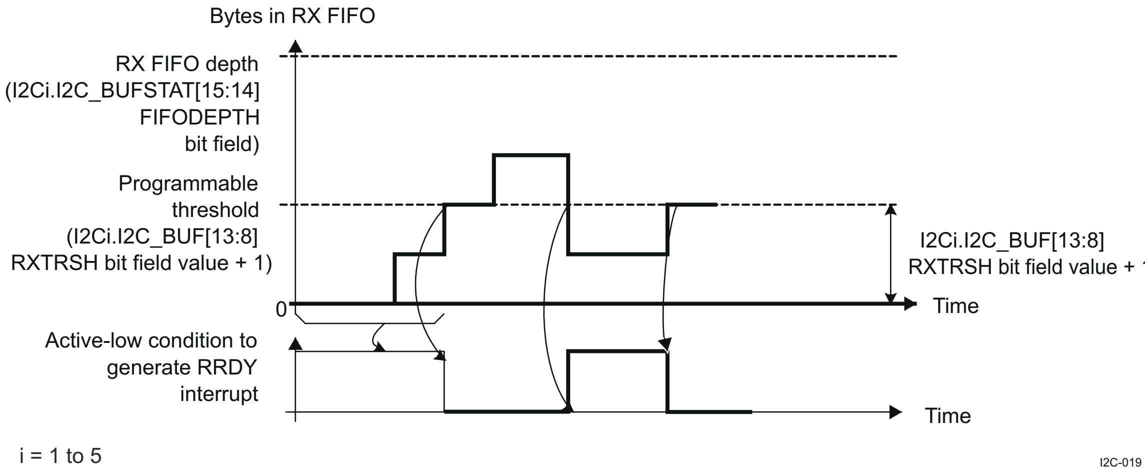


Figure 24-14. HS I²C Receive FIFO Interrupt Request Generation

In Figure 24-14 , the RRDY interrupt condition shows that the condition for generating an RRDY interrupt is achieved. The interrupt request is generated when this signal is active, and it can be cleared only by the LH by writing 1 in the corresponding bit. If the condition is still present after clearing the previous interrupt, another interrupt request is generated.

In receive mode, an RRDY interrupt is generated as soon as the FIFO reaches its receive threshold (I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1). The interrupt can be deasserted only when the LH has handled enough bytes to make the number of bytes in the RX FIFO lower than the programmed threshold. For each interrupt, the LH can be configured to read a number of bytes equal to the value of the RX FIFO threshold.

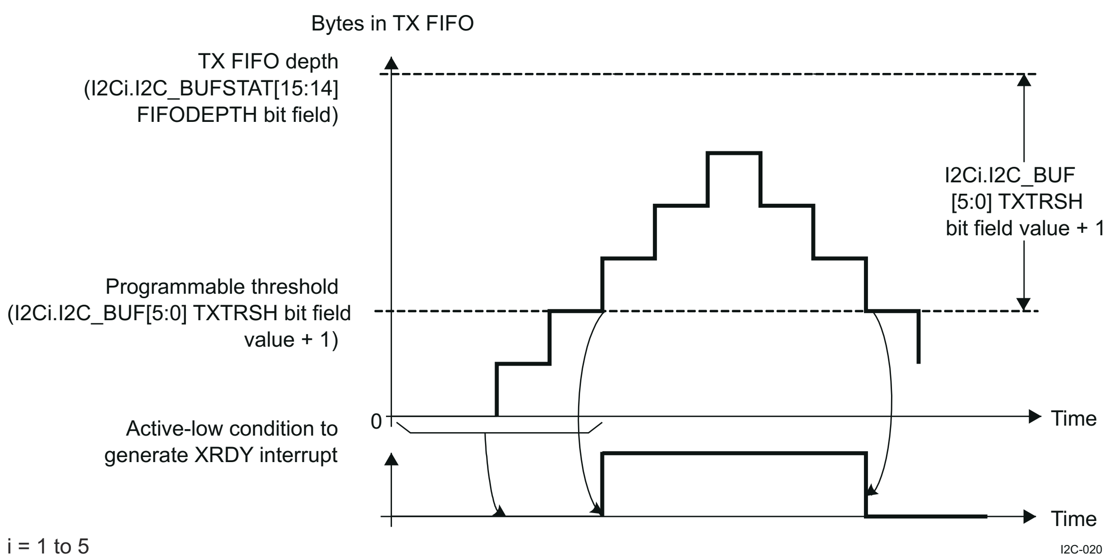


Figure 24-15. HS I²C Transmit FIFO Interrupt Request Generation

In Figure 24-15 , the XRDY interrupt condition shows that the condition for generating an XRDY interrupt is achieved. The interrupt request is generated when TX FIFO is empty or when the TX FIFO threshold is not reached, and the LH can clear the XRDY status bit by setting the I2C*i*.I2C_IRQENABLE_CLR [4] XRDY_IE bit to 1 after transmitting the configured number of bytes. If the condition is still present after clearing the previous interrupt, another interrupt request is generated.

In interrupt mode, the module offers two options for the LH application to handle the interrupts:

- When detecting an interrupt request (XRDY or RRDY type), the LH can write/read 1 data byte to/from the TX/RX FIFO and then clear the interrupt. The module reasserts the interrupt until the interrupt condition is not met.
- When detecting an interrupt request (XRDY or RRDY type), the LH can be programmed to write/read the amount of data bytes specified by the corresponding FIFO threshold (I2C_BUF[5:0] TXTRSH + 1 or I2C_BUF[5:0] RXTRSH + 1). In this case, the interrupt condition is cleared and the next interrupt is asserted again when the XRDY or RRDY condition is met again.

If the second-interrupt-serving approach is used, an additional mechanism (draining feature) is implemented for cases where the transfer length is not a multiple of the FIFO threshold value (see Section 24.1.4.8.4, *Draining Feature [I²C Mode Only]*).

Note

In slave transmit mode (the I2C*i*.I2C_CON[10] MST bit is cleared and the I2C*i*.I2C_CON[9] TRX bit is set to 1), the draining feature must not be used, because the transfer length is not known at configuration time, and the external master can end the transfer at any point by not acknowledging 1 data byte. If the draining feature is used in slave transmit mode, data can remain in the TX FIFO without being transmitted over the I²C bus. In this case, the TX FIFO must be cleared by setting the I2C*i*.I2C_BUF[6] TXFIFO_CLR bit.

24.1.4.8.2 HS I²C FIFO Polling Mode

In FIFO polling mode (the I2C*i*.I2C_IRQENABLE_SET [4] XRDY_IE and I2C*i*.I2C_IRQENABLE_SET [3] RRDY_IE bits are disabled), the status of the module (receiver or transmitter) can be checked by polling the I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY and the I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bits (the I2C*i*.I2C_IRQSTATUS_RAW [13] RDR and I2C*i*.I2C_IRQSTATUS_RAW [14] XDR bits can also be polled if the draining feature is enabled). The I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY and I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bits accurately reflect the interrupt conditions described in the discussion of FIFO interrupt mode.

24.1.4.8.3 HS I²C FIFO DMA Mode

In receive mode, a DMA request is generated by the I2C*i*_DREQ_RX signal as soon as the RX FIFO exceeds its threshold level (the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1). This request is deasserted when the number of bytes defined by the threshold level is read by the DMA controller.

Figure 24-16 shows the DMA request generation in receive mode.

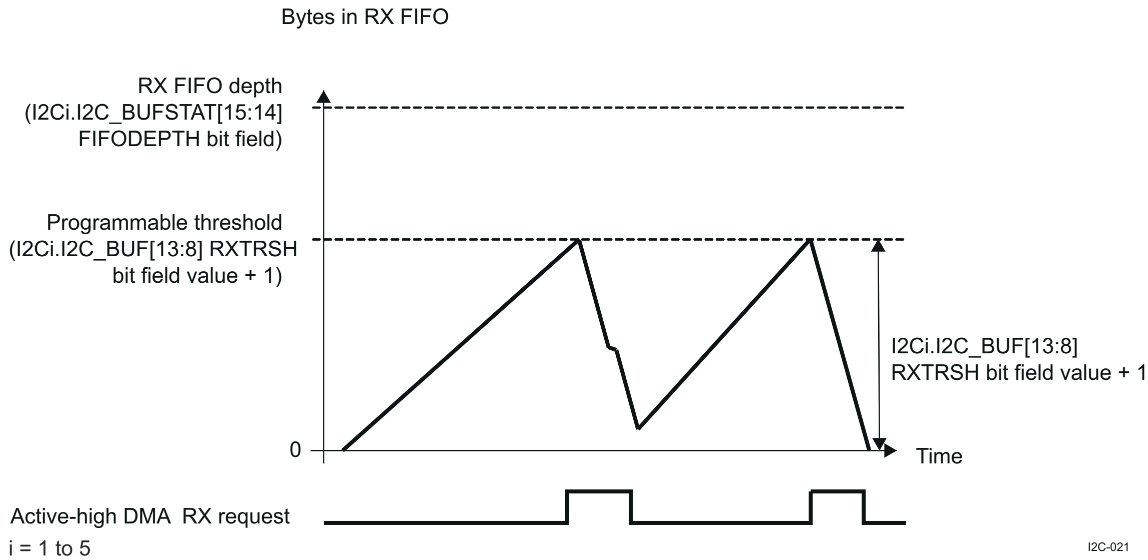


Figure 24-16. HS I²C Receive FIFO DMA Request Generation

In transmit mode, a DMA request is automatically asserted by the I2C*i*_DREQ_TX signal when the TX FIFO is empty. This request is deasserted when the number of bytes (the I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1) is written in the FIFO by the DMA controller. If an insufficient number of bytes is written, the DMA request remains active. Figure 24-17 and Figure 24-18 show the DMA TX transfers with different values for the I2C*i*.I2C_BUF[5:0] TXTRSH bit field.

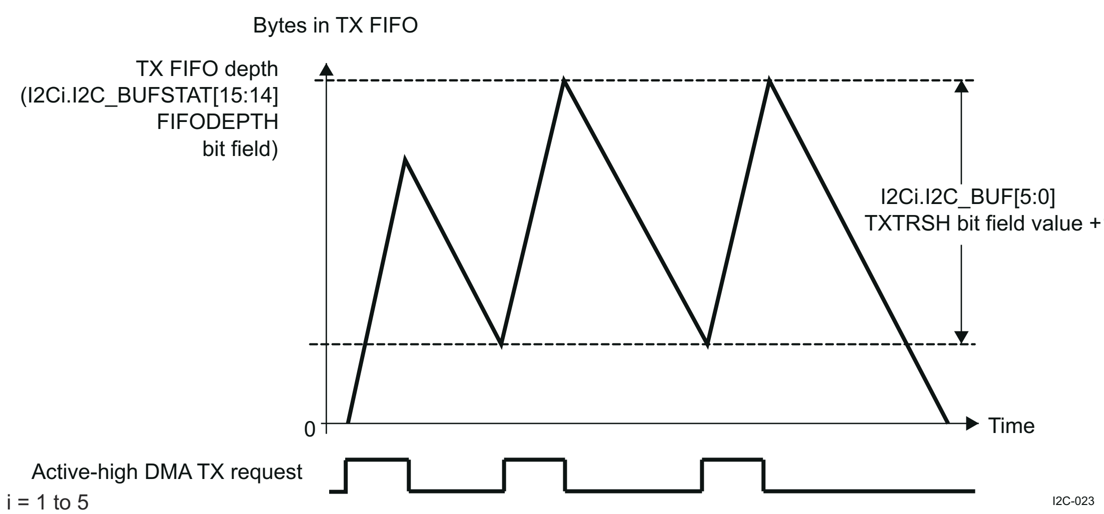


Figure 24-17. HS I²C Transmit FIFO Request Generation (High Threshold)

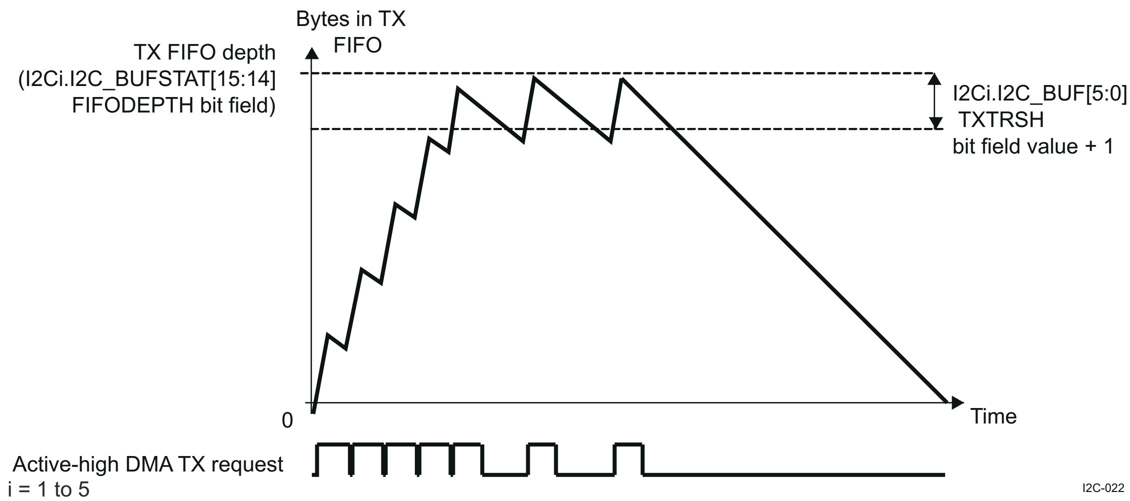


Figure 24-18. HS I²C Transmit FIFO Request Generation (Low Threshold)

The I2C module provides the possibility to the user to clear the RX or TX FIFO, by setting the I2Ci.I2C_BUF[14]RXFIFO_CLR and I2Ci.I2C_BUF[6]TXFIFO_CLR registers, which act like software reset for the FIFOs. In DMA mode, these bits will also reset the DMA state machines.

The FIFO clearing feature can be used when the following conditions are met:

1. The module is configured as a transmitter
2. The external receiver responds with a NACK in the middle of the transfer
3. There is still data in TX FIFO waiting to be transferred

24.1.4.8.4 HS I²C Draining Feature

The draining feature is implemented to handle the end of a transfer whose length is not a multiple of the FIFO threshold values (the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1 for the RX threshold and the I2C*i*.I2C_BUF[5:0] TXTRSH field value + 1 for the TX threshold). It can also transfer the remaining number of bytes (because the threshold is not reached).

This feature prevents the LH or the DMA controller from trying more FIFO accesses than necessary (for example, to generate at the end of a transfer a DMA RX request having fewer bytes in the FIFO than the configured DMA transfer length). Otherwise, an AERR interrupt is generated by the I2C*i*.I2C_IRQSTATUS_RAW [7] AERR bit.

The draining mechanism generates an interrupt using the I2C*i*.I2C_IRQSTATUS_RAW [13] RDR or I2C*i*.I2C_IRQSTATUS_RAW [14] XDR bit at the end of the transfer, informing the LH that it must check the amount of data left to be transferred (the I2C*i*.I2C_BUFSTAT[13:8] RXSTAT or I2C*i*.I2C_BUFSTAT[5:0] TXSTAT bit fields) and enable the draining feature of the DMA controller by reconfiguring the DMA transfer length according to this value (when the DMA mode is enabled) or perform only the required number of data accesses (when the DMA mode is disabled).

In receive mode (master or slave), if the RX FIFO threshold (the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1) is not reached, but the transfer ends on the I²C bus and data remains in the RX FIFO (less than the threshold), the receive draining interrupt (the I2C*i*.I2C_IRQSTATUS_RAW [13] RDR bit) is asserted to inform the LH that it can read the amount of data in the RX FIFO (the I2C*i*.I2C_BUFSTAT[13:8] RXSTAT bit field). The LH performs a number of data read accesses equal to the I2C*i*.I2C_BUFSTAT[13:8] RXSTAT bit field (interrupt or polling mode), or reconfigures the DMA controller with the required value to drain the FIFO.

In master transmit mode, if the TX FIFO threshold (the I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1) is not reached, but the amount of data remaining to be written in the TX FIFO is less than the threshold, the transmit draining interrupt (the I2C*i*.I2C_IRQSTATUS_RAW [14] XDR bit) is asserted to inform the LH that it can read the amount of data remaining to be written in the TX FIFO (the I2C*i*.I2C_BUFSTAT[5:0] TXSTAT bit field). The LH must write the required number of data bytes specified by the I2C*i*.I2C_BUFSTAT[5:0] TXSTAT bit field value or reconfigure the DMA controller with the value required to transfer the last bytes to the FIFO.

In master mode, the LH can alternately not check the values of the I2C*i*.I2C_BUFSTAT[5:0] TXSTAT and I2C*i*.I2C_BUFSTAT[13:8] RXSTAT bit fields, because it can obtain this information internally (by computing the I2C*i*.I2C_CNT[15:0] DATACOUNT bit field value modulo I2C*i*.I2C_BUF[13:8] RXTRSH or I2C*i*.I2C_BUF[5:0] TXTRSH).

By default, the draining feature is disabled; it can be enabled using the I2C*i*.I2C_IRQENABLE_SET [14] XDR_IE or I2C*i*.I2C_IRQENABLE_SET [13] RDR_IE bits (default disabled) only for transfers with lengths not equal to the threshold values (I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1 for the TX threshold or the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1 for the RX threshold).

24.1.4.9 HS I^{2}C Noise Filter

The noise filter is used to suppress any noise that is 50 ns or less in case of F/S operation modes, and any noise that is 10 ns or less in case of HS mode operation. The noise filter is always one period of the I2C*i*_INTERNAL_CLK clock. This way, for HS mode operation (prescaler bypassed), the filter suppresses spikes of less than 10.4 ns.

For standard mode (for example, the I2C*i*.I2C_PSC[7:0] PSC bit field = 4), the maximum width of suppressed spikes is 46.1 ns.

To ensure correct filtering, the prescaler must be programmed accordingly by the 2C*i*.I2C_PSC[7:0] PSC bit field.

24.1.4.10 HS I^{2}C System Test Mode

A system test mode is available for multimaster HS I^{2}C controller module testing. This mode is enabled by setting the I2C*i*.I2C_SYSTEST[15] ST_EN bit to 1. When this bit is cleared to 0, the I^{2}C controller is configured in normal operation mode.

In system test mode, the I2C*i*.I2C_SYSTEST [13:12] TMODE bit field selects the type of test. Table 24-12 lists the tests available for the multimaster HS I^{2}C controllers.

Table 24-12. HS I^{2}C List of Tests

I2C*i*.I2C_SYSTEST[13:12] TMODE	Test	Description
00	Functional mode	Normal operation mode
01	Reserved (not used)	
10	Test of i2c*i*_scl serial clock line	The i2c*i*_scl line is driven with a permanent clock as if mastered with the parameters set in the I2C*i*.I2C_PSC, I2C*i*.I2C_SCLL, and I2C*i*.I2C_SCLH registers.
11	Loop-back mode + i2c*i*_scl/ i2c*i*_sda I/O	In master transmit mode only, data transmitted out of the I2C*i*.I2C_DATA register (write action) is received in the same I2C*i*.I2C_DATA register through an internal path through the FIFO buffers. The DMA and interrupt requests are normally generated if they are enabled. Moreover, the i2c*i*_scl and i2c*i*_sda are controlled with the I2C*i*.I2C_SYSTEST[3:0] bits.

Note

When the I2C*i*.I2C_SYSTEST[13:12] TMODE bit field is set to 11, the I^{2}C controller must be configured in I^{2}C F/S (I2C*i*.I2C_CON[13:12] OPMODE set to 00) or I^{2}C HS mode (I2C*i*.I2C_CON[13:12] OPMODE set to 01).

Note

In normal operation mode (the I2C*i*.I2C_SYSTEST[15] ST_EN bit cleared to 0), the I2C*i*.I2C_SYSTEST[3:0] bits that control the i2c*i*_scl, i2c*i*_sda lines in system test mode are read-only bits.

In system test mode (the I2C*i*.I2C_SYSTEST[15] ST_EN bit set to 1), the I2C*i*.I2C_IRQSTATUS_RAW.XRDY, I2C_IRQSTATUS_RAW.RRDY, I2C_IRQSTATUS_RAW.XUDF, I2C_IRQSTATUS_RAW.ROVR, I2C_IRQSTATUS_RAW.ARDY and I2C_IRQSTATUS_RAW.NACK status bits can be set to 1 when the I2C*i*.I2C_SYSTEST[11] SSB bit is set to 1. Clearing the I2C*i*.I2C_SYSTEST[11] SSB bit to 0 does not clear the I2C*i*.I2C_IRQSTATUS_RAW bits to 0. The I2C*i*.I2C_IRQSTATUS_RAW bit field can be cleared to 0 only by writing 1 in the corresponding bits.

24.1.5 HS I²C Programming Guide

24.1.5.1 HS I²C Low-Level Programming Models

24.1.5.1.1 HS I²C Programming Model

This section describes the programming model of the multimaster HS I²C controllers configured in I²C mode.

24.1.5.1.1.1 Main Program

24.1.5.1.1.1.1 Configure the Module Before Enabling the I²C Controller

Before enabling the I²C controller, perform the following steps:

1. Enable the functional and interface clocks (see Table 24-3).
2. Program the prescaler to obtain an approximately 12-MHz internal sampling clock by programming the corresponding value in the I2C*i*.I2C_PSC[7:0] PSC bit field. This value depends on the frequency of the functional clock (I2C*i*_FCLK).
3. Program the I2C*i*.I2C_SCLL[7:0] SCLL and I2C*i*.I2C_SCLH[7:0] SCLH bit fields to obtain a bit rate of 100 kbps or 400 kbps. These values depend on the internal sampling clock frequency (see Table 24-6).
4. (Optional) Program the I2C*i*.I2C_SCLL[15:8] HSSCLL and I2C*i*.I2C_SCLH[15:8] HSSCLH bit fields to obtain a bit rate of 400 kbps or 3.4 Mbps (for the second phase of HS mode). These values depend on the internal sampling clock frequency (see Table 24-6).
5. Configure the Own Address of the I²C controller by storing it in the I2C*i*.I2C_OA register. Up to four Own Addresses can be programmed in the I2C*i*.I2C_OA and I2C*i*.I2C_OAx registers (where x = 1, 2, 3) for each I²C controller.

Note

For a 10-bit address, set the corresponding expand Own Address bit in the I2C*i*.I2C_CON register.

6. Set the TX threshold (in transmitter mode) and the RX threshold (in receiver mode) by setting the I2C*i*.I2C_BUF[5:0] TXTRSH bit field to (TX threshold – 1) and the I2C*i*.I2C_BUF[13:8] RXTRSH bit field to (RX threshold – 1), where the TX and RX thresholds are greater than or equal to 1.
7. Take the I²C controller out of reset by setting the I2C*i*.I2C_CON[15] I2C_EN bit to 1.

24.1.5.1.1.1.2 Initialize the I²C Controller

To initialize the I²C controller, perform the following steps:

1. Configure the I2C*i*.I2C_CON register:
   - For master or slave mode, set the I2C*i*.I2C_CON[10] MST bit (0: slave; 1: master).
   - For transmitter or receiver mode, set the I2C*i*.I2C_CON[9] TRX bit (0: receiver; 1: transmitter).
2. If using an interrupt to transmit and receive data, set the corresponding bit in the I2C*i*.I2C_IRQENABLE_SET register to 1 (the I2C*i*.I2C_IRQENABLE_SET [4] XRDY_IE bit for the transmit interrupt, the I2C*i*.I2C_IRQENABLE_SET [3] RRDY bit for the receive interrupt).
3. If using DMA to receive and transmit data, set the corresponding bit in the I2C*i*.I2C_BUF register to 1 (the I2C*i*.I2C_BUF[15] RDMA_EN bit for the receive DMA channel, the I2C*i*.I2C_BUF[7] XDMA_EN bit for the transmit DMA channel).

24.1.5.1.1.1.3 Configure Slave Address and the Data Control Register

In master mode, configure the slave address register by programming the I2C*i*.I2C_SA[9:0] SA bit field and the number of data bytes (I²C data payload) associated with the transfer by programming the I2C*i*.I2C_CNT[15:0] DCOUNT bit field.

Note

For a 10-bit address, set the I2C*i*.I2C_CON[8] XSA bit to 1.

24.1.5.1.1.1.4 Initiate a Transfer

Poll the I2C*i*.I2C_IRQSTATUS_RAW [12] BB bit. If it is cleared to 0 (bus not busy), configure the I2C*i*.I2C_CON[0] STT and I2C*i*.I2C_CON[1] STP bits. To initiate a transfer, the I2C*i*.I2C_CON[0] STT bit must be set to 1, and it is not mandatory to set the I2C*i*.I2C_CON[1] STP bit to 1.

24.1.5.1.1.1.5 Receive Data

Poll the I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bit, or use the RRDY interrupt (the I2C*i*.I2C_IRQENABLE_SET [3] RRDY_IE bit must be set to 1) or the DMA RX channel (the I2C*i*.I2C_BUF[15] RDMA_EN bit must be set to 1 together with I2C_DMARXENABLE_SET) to read the receive data in the I2C*i*.I2C_DATA register.

If the transfer length does not equal the RX FIFO threshold (the I2C*i*.I2C_BUF[13:8] RTRSH bit field + 1), use the draining feature (enable the RDR interrupt by setting the I2C*i*.I2C_IRQENABLE_SET [13] RDR_IE bit to 1).

Note

In receive mode only, the I2C*i*.I2C_IRQSTATUS_RAW [11] ROVR (receive overrun) bit indicates whether the receiver has experienced overrun. An overrun condition occurs when the shift register and the RX FIFO are full. An overrun condition does not result in data loss; the I^{2}C controller simply holds i2c*i*_scl to low to prevent other bytes from being received.

The I2C*i*.I2C_IRQSTATUS_RAW[7] AERR bit is set to 1 when a read access is performed in the I2C*i*.I2C_DATA register while the RX FIFO is empty. The corresponding interrupt can be enabled by setting the I2C*i*.I2C_IRQENABLE_SET [7] AERR_IE bit to 1.

24.1.5.1.1.1.6 Transmit Data

Poll the I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY bit, or use the XRDY interrupt (the I2C*i*.I2C_IRQENABLE_SET [4] XRDY_IE bit must be set to 1) or the DMA TX channel (the I2C*i*.I2C_BUF[7] XDMA_EN bit must be set to 1 together with I2C_DMATXENABLE_SET) to write data to the I2C*i*.I2C_DATA register.

If the transfer length does not equal the TX FIFO threshold (the I2C*i*.I2C_BUF[5:0] TXTRSH bit field + 1), use the draining feature (enable the XDR interrupt by setting the I2C*i*.I2C_IRQENABLE_SET [14] XDR_IE bit to 1).

Note

In transmit mode only, the I2C*i*.I2C_IRQSTATUS_RAW [10] XUDF bit indicates whether the transmitter has experienced underflow.

In master transmit mode, underflow occurs when the shift register and the TX FIFO are empty and there are still some bytes to transmit (the value of the I2C*i*.I2C_CNT[15:0] DCOUNT bit field is not 0).

In slave transmit mode, underflow occurs when the shift register and the TX FIFO are empty and the external I^{2}C master device still requests data bytes to be read.

The I2C*i*.I2C_IRQSTATUS_RAW [7] AERR bit is set to 1 when a write access is performed in the I2C*i*.I2C_DATA register while the TX FIFO is full. The corresponding interrupt can be enabled by setting the I2C*i*.I2C_IRQENABLE_SET [7] AERR_IE bit to 1.

24.1.5.1.1.2 Interrupt Subroutine Sequence

1. Test for arbitration lost (the I2C*i*.I2C_IRQSTATUS_RAW [0] AL bit) and resolve accordingly.
2. Test for no acknowledgment (the I2C*i*.I2C_IRQSTATUS_RAW [1] NACK bit) and resolve accordingly.
3. Test for register access ready (the I2C*i*.I2C_IRQSTATUS_RAW [2] ARDY bit) and resolve accordingly.
4. Test for receive data ready (the I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bit) and resolve accordingly.
5. Test for transmit data ready (the I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY bit) and resolve accordingly.
6. Test for general call (the I2C*i*.I2C_IRQSTATUS_RAW [5] GC bit) and resolve accordingly.
7. Test for start (S) condition (the I2C*i*.I2C_IRQSTATUS_RAW [6] STC bit) and resolve accordingly. For this test, the functional clock must be inactive.
8. Test for access error (the I2C*i*.I2C_IRQSTATUS_RAW [7] AERR bit) and resolve accordingly.

9. Test for bus free (the I2C*i*.I2C_IRQSTATUS_RAW [8] BF bit) and resolve accordingly.

24.1.5.1.1.3 Programming Flow-Diagrams

Figure 24-19 through Figure 24-27 are procedure flow charts for programming the F/S and HS I^{2}C modes.

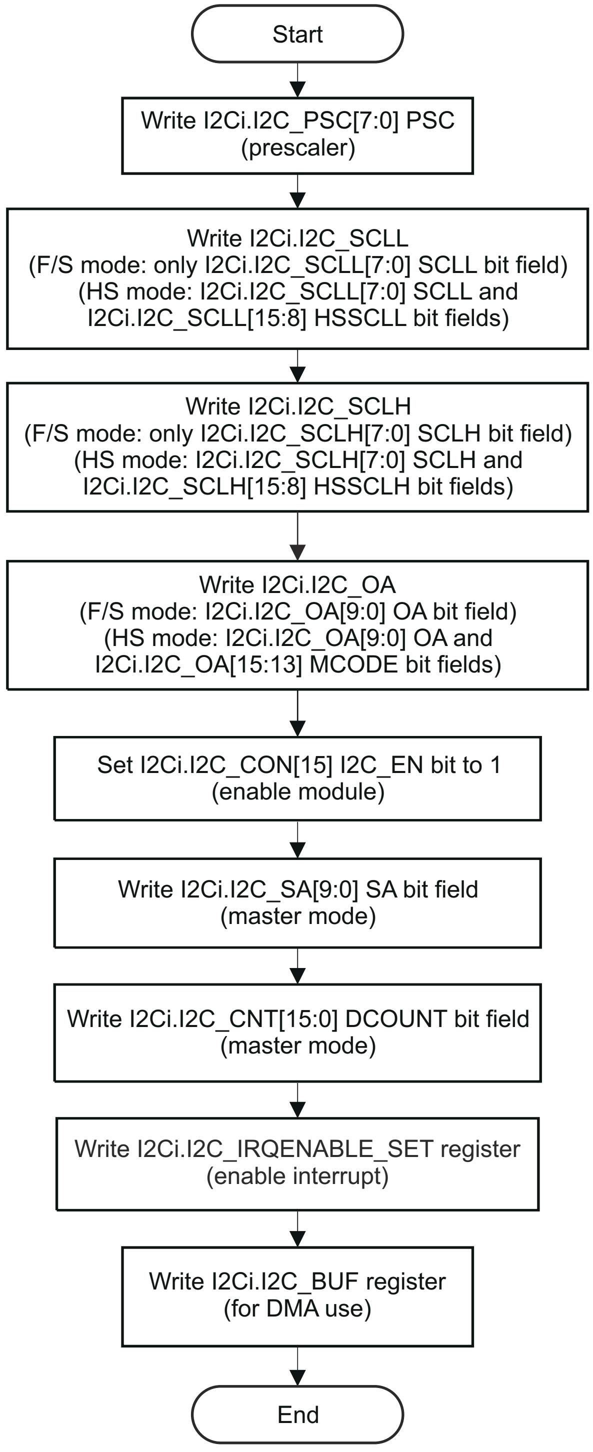


Figure 24-19. HS I^{2}C Setup Procedure

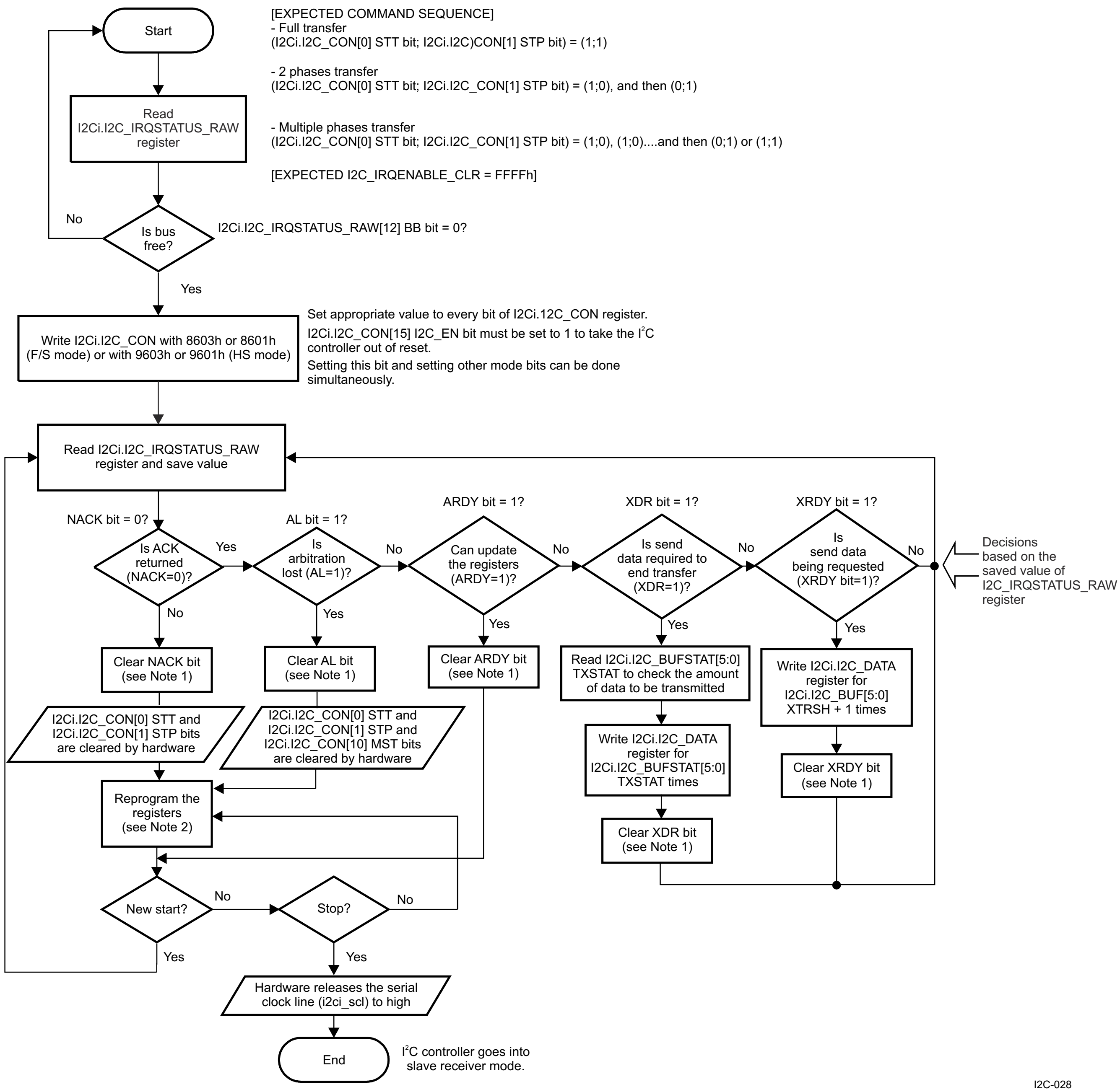
Table 24-13. Subprocess Call Summary for Sequence – I²C Setup Procedure

Subprocess Name	Cross-Reference
Pad configuration	See *PAD Configuration Registers* in *Control Module*

Table 24-14. HS I²C Register Call Summary for Sequence – Setup Procedure

Register Name	Register Name	Register Name
I2C*i*.I2C_PSC (1)	I2C*i*.I2C_CON (1)	I2C*i*.I2C_BUF (1)
I2C*i*.I2C_SCLL (1)	I2C*i*.I2C_SA (1)	I2C*i*.I2C_IRQENABLE_SET (1)
I2C*i*.I2C_SCLH (1)	I2C*i*.I2C_CNT (1)	I2C*i*.I2C_OA (1)

(1) *i* = 1 to 5



A. The NACK, AL, ARDY, XDR, and XRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

B. Reprogram the registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Figure 24-20. HS I^{2}C Master Transmitter Mode, Polling Method, in F/S and HS Modes

Note

The FIFO clearing can be made when the module is configured as transmitter, the receiver send a NACK in the middle of the transfer, and there is still data in the FIFO.

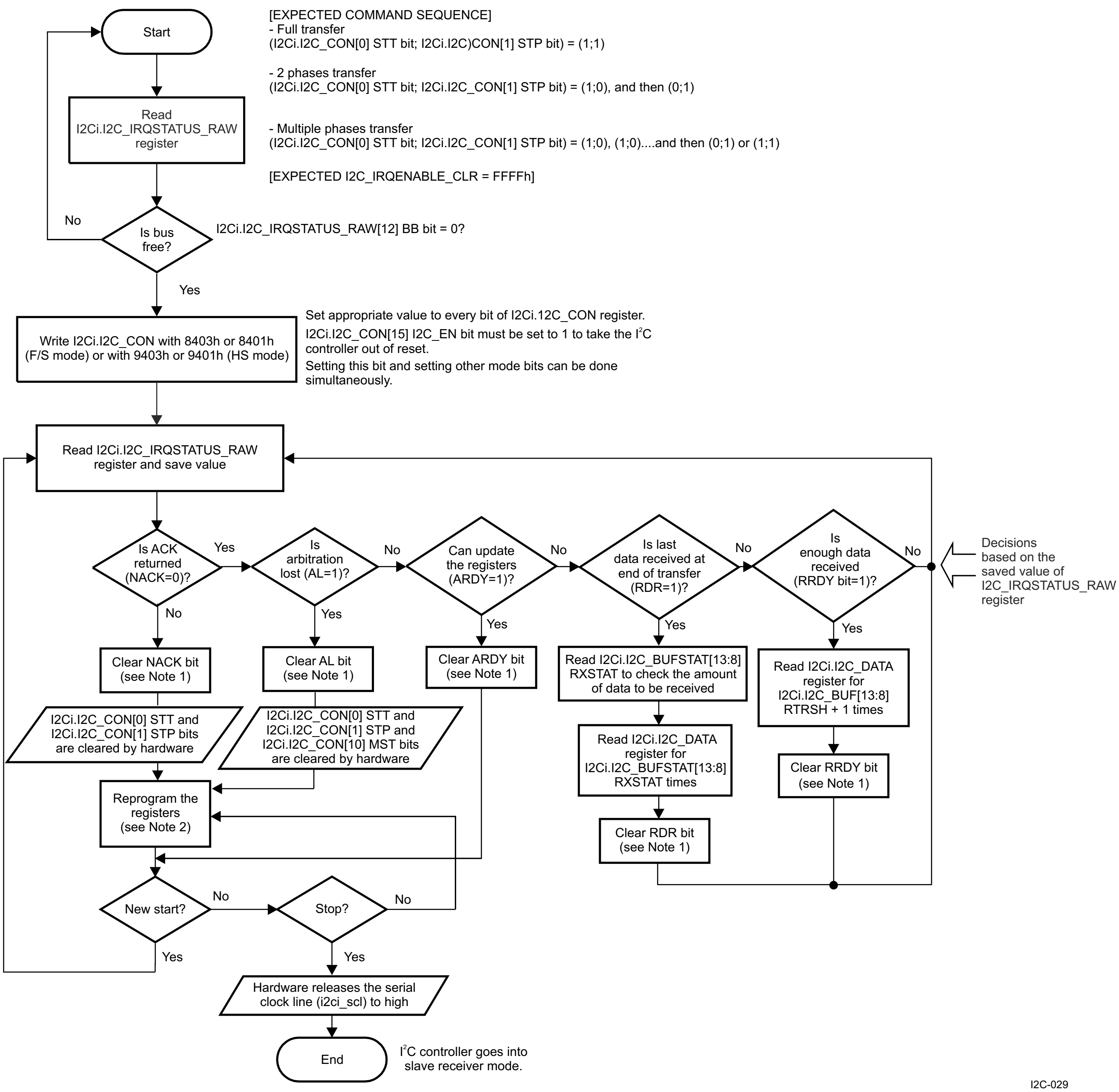
Note

In HS mode, the Sr condition and clock frequency switching are automatically generated by the multimaster HS I^{2}C controller.

Table 24-15. HS I^{2}C Register Call Summary for Sequence – Master Transmitter Mode, Polling Method, in F/S and HS Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW [(1)]	I2C*i*.I2C_DATA [(1)]	I2C*i*.I2C_CON [(1)]
I2C*i*.I2C_BUFSTAT [(1)]	I2C*i*.I2C_CON [(1)]	I2C*i*.I2C_BUF [(1)]

(1) *i* = 1 to 5



A. The NACK, AL, ARDY, RDR, and RRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

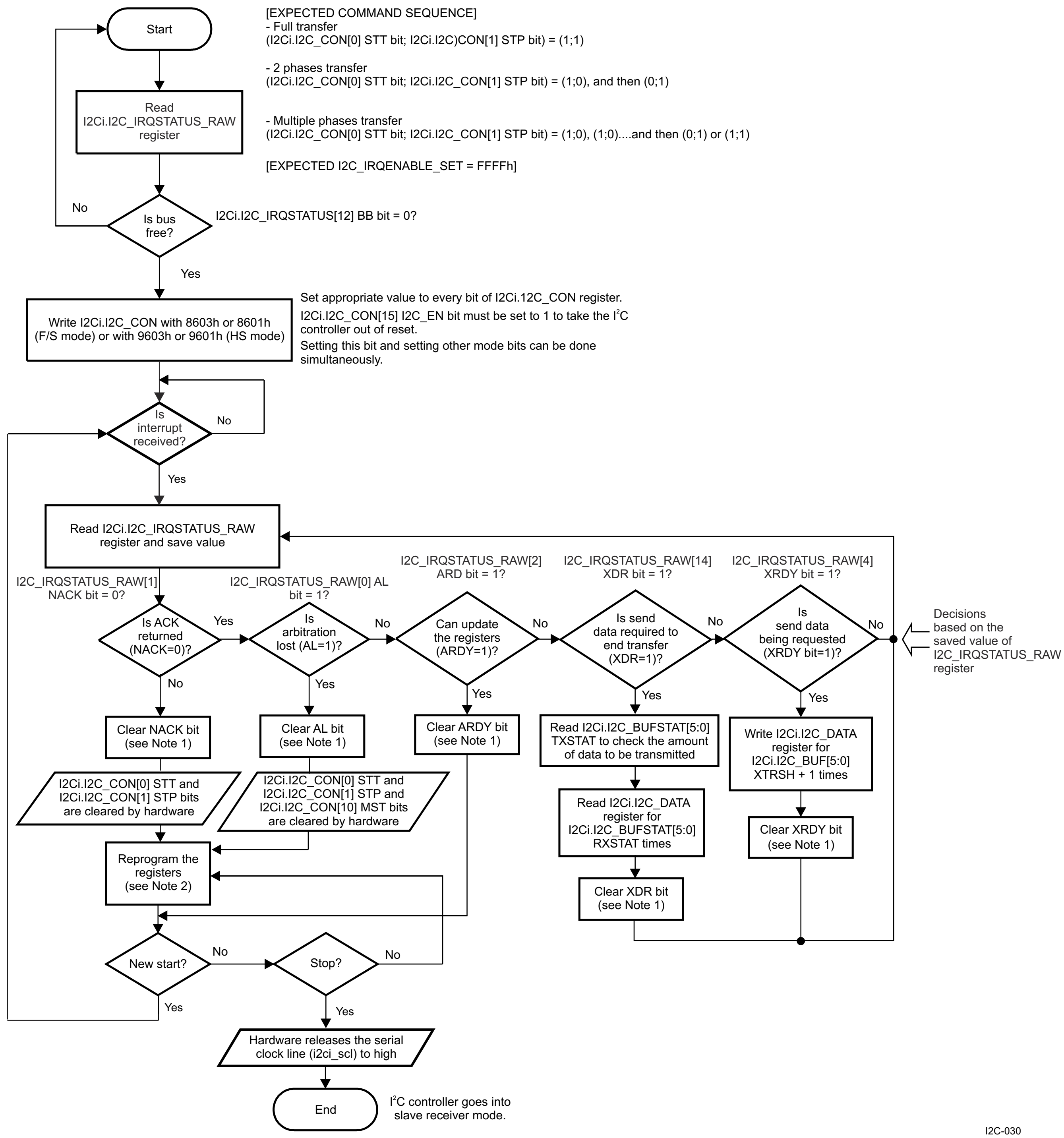
B. Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Figure 24-21. HS I^{2}C Master Receiver Mode, Polling Method, in F/S and HS Modes

Table 24-16. HS I^{2}C Register Call Summary for Sequence – Master Receiver Mode, Polling Method, in F/S and HS Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)	I2C*i*.I2C_BUF (1)
I2C*i*i.I2C_CON (1)	I2C*i*.I2C_DATA (1)	

(1) *i* = 1 to 5



A. The NACK, AL, ARDY, XDR, and XRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

B. Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Figure 24-22. HS I²C Master Transmitter Mode, Interrupt Method, in F/S and HS Modes

Note

The FIFO clearing can be made when the module is configured as transmitter, the receiver send a NACK in the middle of the transfer, and there is still data in the FIFO.

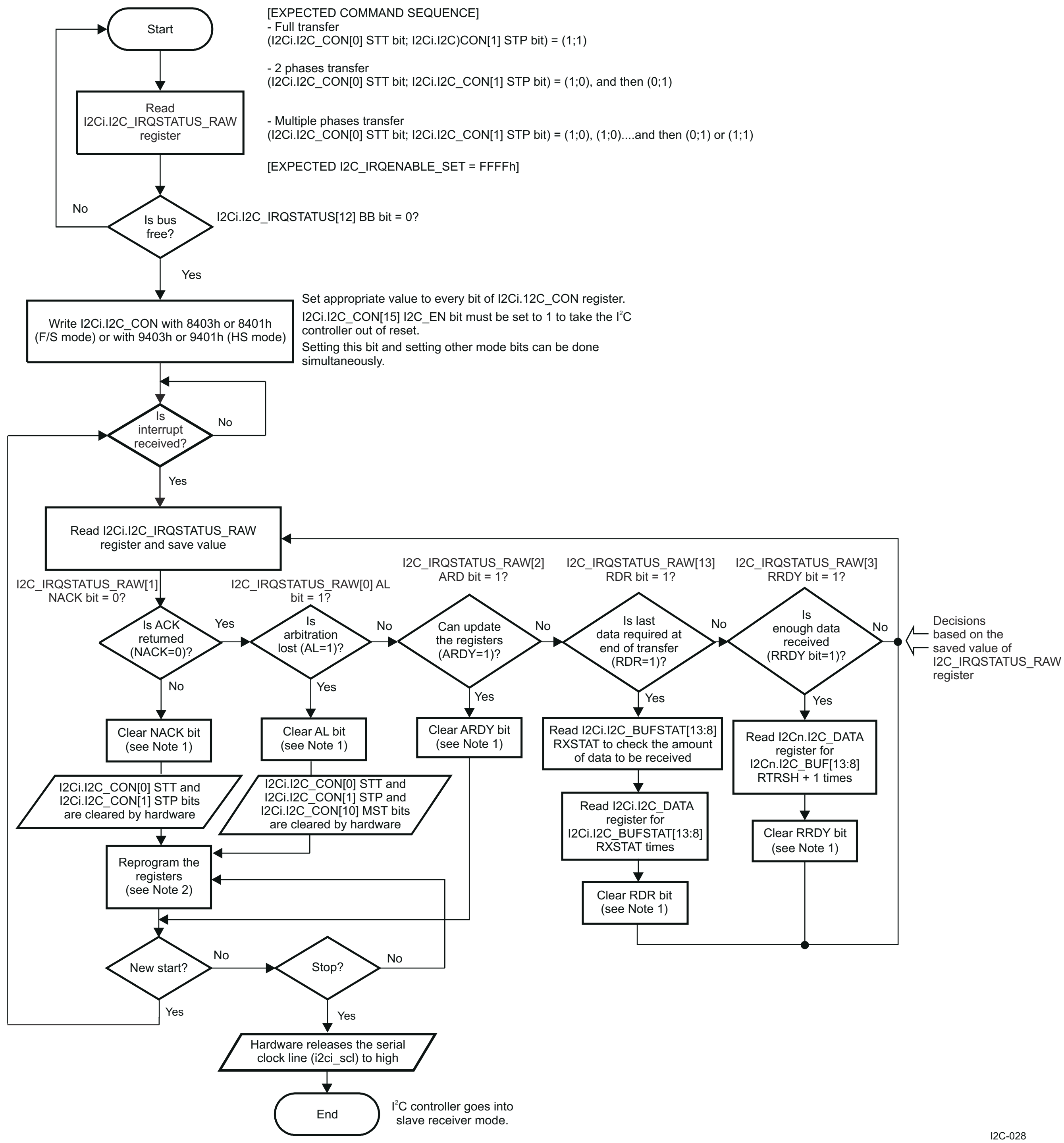
Note

In HS mode, the Sr condition and clock frequency switching are automatically generated by the multimaster HS I^{2}C controller.

Table 24-17. HS I^{2}C Register Call Summary for Sequence – Master Transmitter Mode, Interrupt Method, in F/S and HS Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)	I2C*i*.I2C_BUF (1)
I2C*i*.I2C_CON (1)	I2C*i*.I2C_DATA (1)	

(1) *i* = 1 to 5



A. The NACK, AL, ARDY, RDR, and RRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*I2C_IRQSTATUS register.

B. Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Figure 24-23. HS I²C Master Receiver Mode, Interrupt Method, in F/S and HS Modes

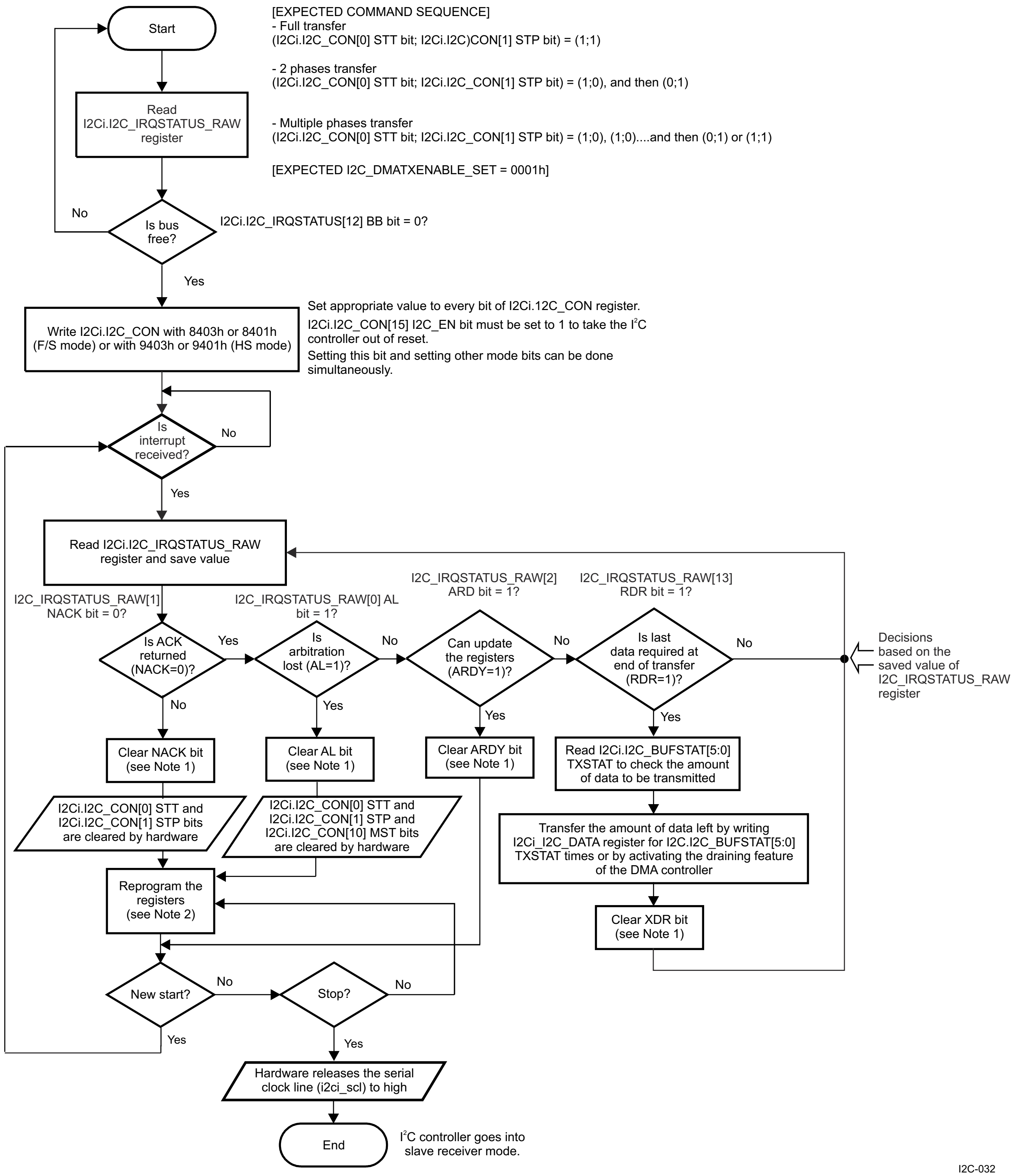
Table 24-18. HS I²C Register Call Summary for Sequence – Master Receiver Mode, Interrupt Method, in F/S and HS Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW [(1)]	I2C*i*.I2C_BUFSTAT [(1)]	I2C*i*.I2C_BUF [(1)]

Table 24-18. HS I²C Register Call Summary for Sequence – Master Receiver Mode, Interrupt Method, in F/S and HS Modes (continued)

Register Name	Register Name	Register Name
I2C*i*.I2C_CON [(1)]	I2C*i*.I2C_DATA [(1)]	

(1) *i* = 1 to 5



A. The NACK, AL, ARDY, and XDR bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

B. Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2Ci.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Figure 24-24. HS I²C Master Transmitter Mode, DMA Method in F/S and HS Modes

---

Note

The FIFO clearing can be made when the module is configured as transmitter, the receiver send a NACK in the middle of the transfer, and there is still data in the FIFO.

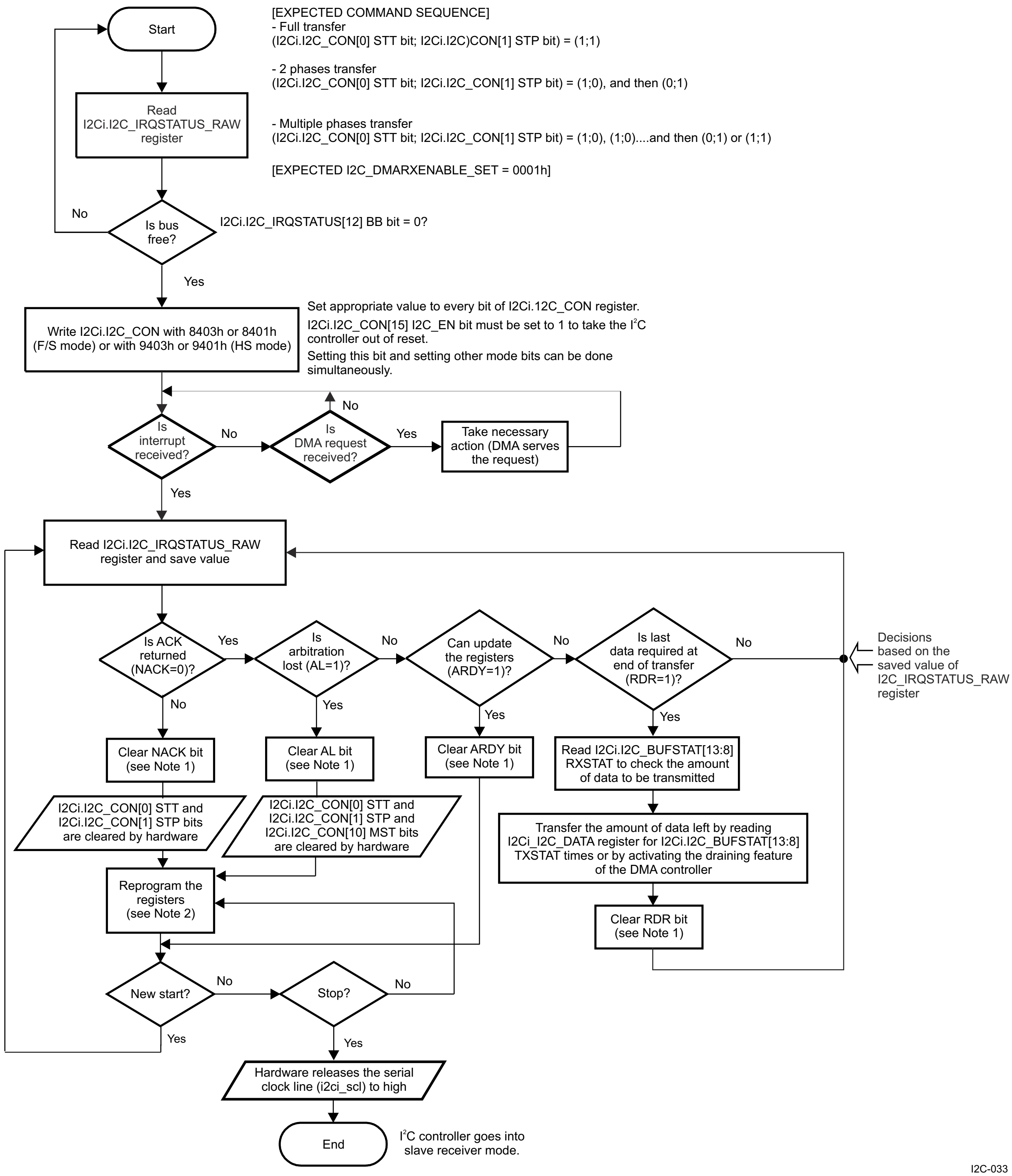
Note

In HS mode, the Sr condition and clock frequency switching are automatically generated by the multimaster HS I²C controller.

Table 24-19. HS I²C Register Call Summary for Sequence – Master Transmitter Mode, DMA Method in F/S and HS Modes

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW [(1)]	I2Ci.I2C_BUFSTAT [(1)]
I2C*i*.I2C_CON [(1)]	I2Ci.I2C_DATA [(1)]

(1) *i* = 1 to 5



A. The NACK, AL, ARDY, and RDR bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

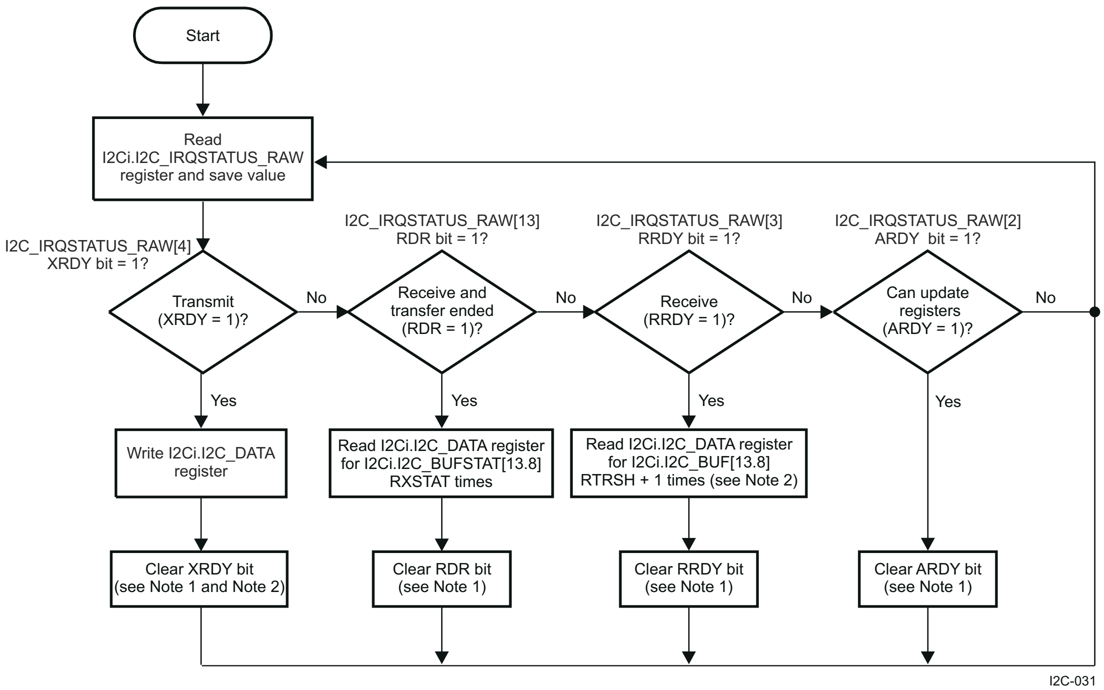
B. Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Figure 24-25. HS I^{2}C Master Receiver Mode, DMA Method in F/S and HS Modes

Table 24-20. HS I²C Register Call Summary for Sequence – Master Receiver Mode, DMA Method in F/S and HS Modes

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)
I2Ci.I2C_CON (1)	I2C*i*.I2C_DATA (1)

(1) *i* = 1 to 5



A. The XRDY, RDR, RRDY, and ARDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

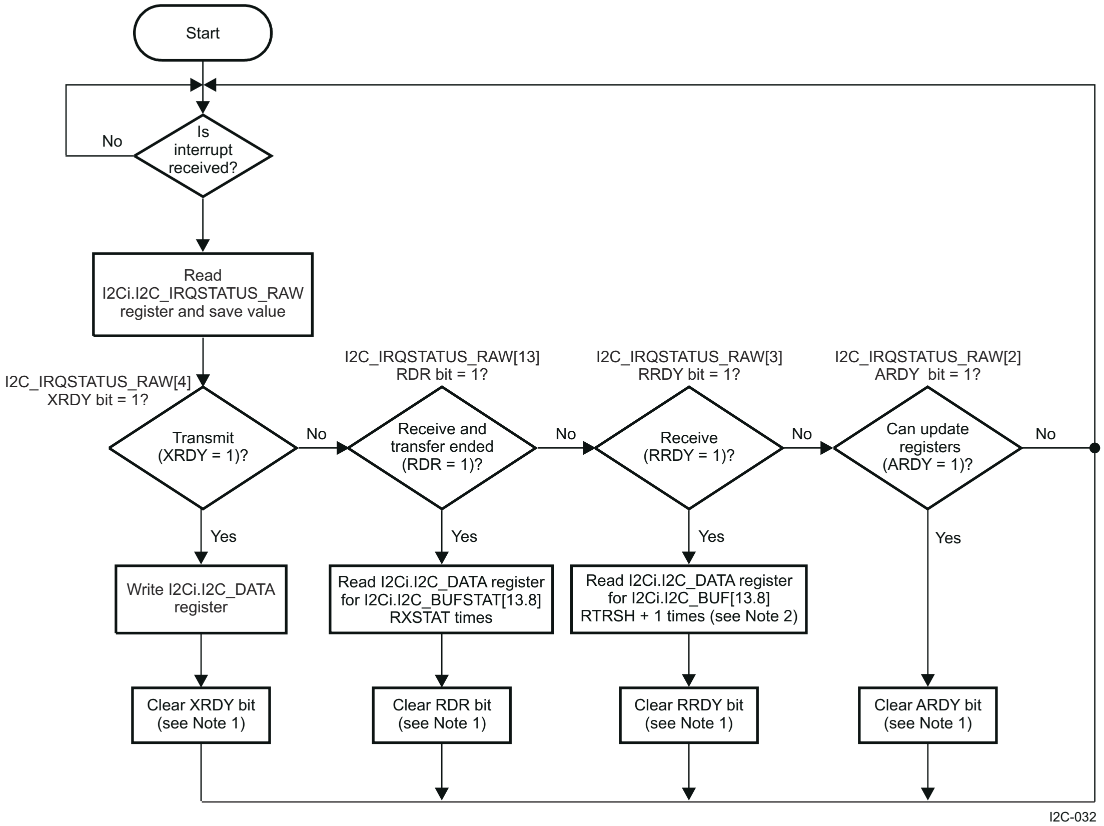
B. In slave transmitter mode, the amount of data requested by the external master I²C device is unknown; thus, the I2C*i*.I2C_BUF[5:0] XTRSH bit field must be configured to 0x0 (TX threshold = 1).

Figure 24-26. HS I²C Slave Transmitter/Receiver Mode, Polling

Table 24-21. HS I²C Register Call Summary for Sequence – Slave Transmitter/Receiver Mode, Polling

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)
I2C*i*.I2C_DATA (1)	I2C*i*.I2C_BUF (1)

(1) *i* = 1 to 5



A. The XRDY, RDR, RRDY, and ARDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

B. In slave transmitter mode, the amount of data requested by the external master I^{2}C device is unknown; thus, the I2C*i*.I2C_BUF[5:0] XTRSH bit field must be configured to 0x0 (TX threshold = 1).

Figure 24-27. HS I^{2}C Slave Transmitter/Receiver Mode, Interrupt

Table 24-22. HS I^{2}C Register Call Summary for Sequence – Slave Transmitter/Receiver Mode, Interrupt

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)
I2C*i*.I2C_DATA (1)	I2C*i*.I2C_BUF (1)

(1) *i* = 1 to 5

24.1.6 HS I2C Register Manual

24.1.6.1 HS I2C Instance Summary

Table 24-23 lists the base address and block size for the HS I2C module instances.

Table 24-23. HS I2C Instance Summary

Module Name	Module Base Address	Size
I2C1	0x4807 0000	214 Bytes
I2C2	0x4807 2000	214 Bytes
I2C3	0x4806 0000	214 Bytes
I2C4	0x4807 A000	214 Bytes
I2C5	0x4807 C000	214 Bytes

24.1.6.2 HS I2C Registers

24.1.6.2.1 HS I2C Register Summary

Table 24-24 and Table 24-25 provide the register summary and associated offset addresses for the five HS I2C internal registers.

Table 24-24. HS I2C Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	I2C1 Physical Address	I2C2 Physical Address	I2C3 Physical Address
I2C_REVNB_LO	R	16	0x0000 0000	0x4807 0000	0x4807 2000	0x4806 0000
I2C_REVNB_HI	R	16	0x0000 0004	0x4807 0004	0x4807 2004	0x4806 0004
I2C_SYSC	RW	16	0x0000 0010	0x4807 0010	0x4807 2010	0x4806 0010
I2C_EOI	W	16	0x0000 0020	0x4807 0020	0x4807 2020	0x4806 0020
I2C_IRQSTATUS_RAW	RW	16	0x0000 0024	0x4807 0024	0x4807 2024	0x4806 0024
I2C_IRQSTATUS	RW	16	0x0000 0028	0x4807 0028	0x4807 2028	0x4806 0028
I2C_IRQENABLE_SET	RW	16	0x0000 002C	0x4807 002C	0x4807 202C	0x4806 002C
I2C_IRQENABLE_CLR	RW	16	0x0000 0030	0x4807 0030	0x4807 2030	0x4806 0030
I2C_WE	RW	16	0x0000 0034	0x4807 0034	0x4807 2034	0x4806 0034
I2C_DMARXENABLE_SET	RW	16	0x0000 0038	0x4807 0038	0x4807 2038	0x4806 0038
I2C_DMATXENABLE_SET	RW	16	0x0000 003C	0x4807 003C	0x4807 203C	0x4806 003C
I2C_DMARXENABLE_CLR	RW	16	0x0000 0040	0x4807 0040	0x4807 2040	0x4806 0040
I2C_DMATXENABLE_CLR	RW	16	0x0000 0044	0x4807 0044	0x4807 2044	0x4806 0044
I2C_DMARXWAKE_EN	RW	16	0x0000 0048	0x4807 0048	0x4807 2048	0x4806 0048
I2C_DMATXWAKE_EN	RW	16	0x0000 004C	0x4807 004C	0x4807 204C	0x4806 004C
RESERVED	RW	16	0x0000 0084	0x4807 0084	0x4807 2084	0x4806 0084
RESERVED	RW	16	0x0000 0088	0x4807 0088	0x4807 2088	0x4806 0088
I2C_SYSS	RW	16	0x0000 0090	0x4807 0090	0x4807 2090	0x4806 0090
I2C_BUF	RW	16	0x0000 0094	0x4807 0094	0x4807 2094	0x4806 0094
I2C_CNT	RW	16	0x0000 0098	0x4807 0098	0x4807 2098	0x4806 0098
I2C_DATA	RW	16	0x0000 009C	0x4807 009C	0x4807 209C	0x4806 009C
I2C_CON	RW	16	0x0000 00A4	0x4807 00A4	0x4807 20A4	0x4806 00A4
I2C_OA	RW	16	0x0000 00A8	0x4807 00A8	0x4807 20A8	0x4806 00A8
I2C_SA	RW	16	0x0000 00AC	0x4807 00AC	0x4807 20AC	0x4806 00AC
I2C_PSC	RW	16	0x0000 00B0	0x4807 00B0	0x4807 20B0	0x4806 00B0
I2C_SCLL	RW	16	0x0000 00B4	0x4807 00B4	0x4807 20B4	0x4806 00B4

Table 24-24. HS I²C Registers Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	I2C1 Physical Address	I2C2 Physical Address	I2C3 Physical Address
I2C_SCLH	RW	16	0x0000 00B8	0x4807 00B8	0x4807 20B8	0x4806 00B8
I2C_SYSTEST	RW	16	0x0000 00BC	0x4807 00BC	0x4807 20BC	0x4806 00BC
I2C_BUFSTAT	R	16	0x0000 00C0	0x4807 00C0	0x4807 20C0	0x4806 00C0
I2C_OA1	RW	16	0x0000 00C4	0x4807 00C4	0x4807 20C4	0x4806 00C4
I2C_OA2	RW	16	0x0000 00C8	0x4807 00C8	0x4807 20C8	0x4806 00C8
I2C_OA3	RW	16	0x0000 00CC	0x4807 00CC	0x4807 20CC	0x4806 00CC
I2C_ACTOA	R	16	0x0000 00D0	0x4807 00D0	0x4807 20D0	0x4806 00D0
I2C_SBLOCK	RW	16	0x0000 00D4	0x4807 00D4	0x4807 20D4	0x4806 00D4

Table 24-25. HS I²C Registers Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	I2C4 Physical Address	I2C5 Physical Address
I2C_REVNB_LO	R	16	0x0000 0000	0x4807 A000	0x4807 C000
I2C_REVNB_HI	R	16	0x0000 0004	0x4807 A004	0x4807 C004
I2C_SYSC	RW	16	0x0000 0010	0x4807 A010	0x4807 C010
I2C_EOI	W	16	0x0000 0020	0x4807 A020	0x4807 C020
I2C_IRQSTATUS_RAW	RW	16	0x0000 0024	0x4807 A024	0x4807 C024
I2C_IRQSTATUS	RW	16	0x0000 0028	0x4807 A028	0x4807 C028
I2C_IRQENABLE_SET	RW	16	0x0000 002C	0x4807 A02C	0x4807 C02C
I2C_IRQENABLE_CLR	RW	16	0x0000 0030	0x4807 A030	0x4807 C030
I2C_WE	RW	16	0x0000 0034	0x4807 A034	0x4807 C034
I2C_DMARXENABLE_SET	RW	16	0x0000 0038	0x4807 A038	0x4807 C038
I2C_DMATXENABLE_SET	RW	16	0x0000 003C	0x4807 A03C	0x4807 C03C
I2C_DMARXENABLE_CLR	RW	16	0x0000 0040	0x4807 A040	0x4807 C040
I2C_DMATXENABLE_CLR	RW	16	0x0000 0044	0x4807 A044	0x4807 C044
I2C_DMARXWAKE_EN	RW	16	0x0000 0048	0x4807 A048	0x4807 C048
I2C_DMATXWAKE_EN	RW	16	0x0000 004C	0x4807 A04C	0x4807 C04C
RESERVED	RW	16	0x0000 0084	0x4807 A084	0x4807 C084
RESERVED	RW	16	0x0000 0088	0x4807 A088	0x4807 C088
I2C_SYSS	RW	16	0x0000 0090	0x4807 A090	0x4807 C090
I2C_BUF	RW	16	0x0000 0094	0x4807 A094	0x4807 C094
I2C_CNT	RW	16	0x0000 0098	0x4807 A098	0x4807 C098
I2C_DATA	RW	16	0x0000 009C	0x4807 A09C	0x4807 C09C
I2C_CON	RW	16	0x0000 00A4	0x4807 A0A4	0x4807 C0A4
I2C_OA	RW	16	0x0000 00A8	0x4807 A0A8	0x4807 C0A8
I2C_SA	RW	16	0x0000 00AC	0x4807 A0AC	0x4807 C0AC
I2C_PSC	RW	16	0x0000 00B0	0x4807 A0B0	0x4807 C0B0
I2C_SCLL	RW	16	0x0000 00B4	0x4807 A0B4	0x4807 C0B4
I2C_SCLH	RW	16	0x0000 00B8	0x4807 A0B8	0x4807 C0B8
I2C_SYSTEST	RW	16	0x0000 00BC	0x4807 A0BC	0x4807 C0BC
I2C_BUFSTAT	R	16	0x0000 00C0	0x4807 A0C0	0x4807 C0C0
I2C_OA1	RW	16	0x0000 00C4	0x4807 A0C4	0x4807 C0C4
I2C_OA2	RW	16	0x0000 00C8	0x4807 A0C8	0x4807 C0C8
I2C_OA3	RW	16	0x0000 00CC	0x4807 A0CC	0x4807 C0CC
I2C_ACTOA	R	16	0x0000 00D0	0x4807 A0D0	0x4807 C0D0

Table 24-25. HS I²C Registers Mapping Summary 2 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	I2C4 Physical Address	I2C5 Physical Address
I2C_SBLOCK	RW	16	0x0000 00D4	0x4807 A0D4	0x4807 C0D4

24.1.6.2.2 HS I²C Register Description

Table 24-26 through Table 24-57 describe the individual HS I²C registers.

Table 24-26. I2C_REVNB_LO

Address Offset	0x0000 0000		
Physical Address	0x4806 0000 0x4807 0000 0x4807 2000 0x4807 A000 0x4807 C000	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Module Revision Identifier Used by software to track features, bugs, and compatibility		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION															

Bits	Field Name	Description	Type	Reset
15:0	REVISION	IP Revision	R	TI internal data

Table 24-27. I2C_REVNB_HI

Address Offset	0x0000 0004		
Physical Address	0x4806 0004 0x4807 0004 0x4807 2004 0x4807 A004 0x4807 C004	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Module Revision Identifeir Used by software to track features, bugs, and compatibility		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION															

Bits	Field Name	Description	Type	Reset
15:0	REVISION	IP Revision	R	TI internal data

Table 24-28. I2C_SYSC

Address Offset	0x0000 0010		
Physical Address	0x4806 0010 0x4807 0010 0x4807 2010 0x4807 A010 0x4807 C010	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	System Configuration register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CLKACTIVITY	RESERVED	IDLEMODE	ENAWAKEUP	SRST	AUTOIDLE

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:8	CLKACTIVITY	Clock Activity selection bits 0x0: Both clocks can be cut off 0x1: Only OCP clock must be kept active; system clock can be cut off 0x2: Only system clock must be kept active; OCP clock can be cut off 0x3: Both clocks must be kept active	RW	0x0
7:5	RESERVED	Reads return 0.	R	0x0
4:3	IDLEMODE	Idle Mode selection bits 0x0: Force Idle mode 0x1: No Idle mode 0x2: Smart Idle mode 0x3: Smart-idle wakeup-capable mode	RW	0x0
2	ENAWAKEUP	Enable Wakeup control bit 0x0: Wakeup mechanism is disabled 0x1: Wakeup mechanism is enabled	RW	0
1	SRST	SoftReset bit 0x0: Normal mode 0x1: The module is reset	RW	0
0	AUTOIDLE	Autoidle bit 0x0: Auto Idle mechanism is disabled 0x1: Auto Idle mechanism is enabled	RW	1

Table 24-29. I2C_EOI

Address Offset	0x0000 0020		
Physical Address	0x4806 0020 0x4807 0020 0x4807 2020 0x4807 A020 0x4807 C020	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	End Of Interrupt number specification		
Type	W		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															LINE_NUMBER

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write number of interrupt output.	W	0x0

Table 24-30. I2C_IRQSTATUS_RAW

Address Offset	0x0000 0024

Table 24-30. I2C_IRQSTATUS_RAW (continued)

Physical Address	0x4806 0024 0x4807 0024 0x4807 2024 0x4807 A024 0x4807 C024	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event raw interrupt status vector. The raw status is set even if event is not enabled. Write 1 to set the (raw) status. Used mostly for debug		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	BB	ROVR	XUDF	AAS	BF	AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR	Transmit draining IRQ status. 0x0: Transmit draining inactive. 0x1: Transmit draining enabled.	RW	0
13	RDR	Receive draining IRQ status. 0x0: Receive draining inactive. 0x1: Receive draining enabled.	RW	0
12	BB	Bus busy status. Writing into this bit has no effect. Read 0x1: Bus is occupied. Read 0x0: Bus is free.	R	0
11	ROVR	Receive overrun status. Writing into this bit has no effect. Read 0x1: Receiver overrun. Read 0x0: Normal operation.	RW	0
10	XUDF	Transmit underflow status. Writing into this bit has no effect. Read 0x1: Transmit underflow. Read 0x0: Normal operation.	RW	0
9	AAS	Address recognized as slave IRQ status. 0x0: No action. 0x1: Address recognized.	RW	0
8	BF	Bus Free IRQ status. 0x0: No action. 0x1: Bus Free.	RW	0
7	AERR	Access Error IRQ status. 0x0: No action. 0x1: Access Error.	RW	0
6	STC	Start Condition IRQ status. 0x0: No action. 0x1: Start Condition detected.	RW	0

Bits	Field Name	Description	Type	Reset
5	GC	General call IRQ status. Set to 1 by core when General call address detected and interrupt signaled to IRQ_Crossbar. 0x0: No general call detected. 0x1: General call address detected.	RW	0
4	XRDY	Transmit data ready IRQ status. Set to 1 by core when transmitter and when new data is requested. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. 0x0: Transmision ongoing. 0x1: Transmit data ready.	RW	0
3	RRDY	Receive data ready IRQ status. Set to 1 by core when receiver mode, a new data is able to be read. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. 0x0: No data available. 0x1: Receive data available.	RW	0
2	ARDY	Register access ready IRQ status. When set to 1 it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to IRQ_Crossbar. 0x0: Module busy. 0x1: Access ready.	RW	0
1	NACK	No acknowledgement IRQ status. Bit is set when No Acknowledge has been received, an interrupt is signaled to IRQ_Crossbar. 0x0: Normal operation. 0x1: Not Acknowledge detected.	RW	0
0	AL	Arbitration lost IRQ status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to IRQ_Crossbar. During reads, it always returns 0. 0x0: Normal operation. 0x1: Arbitration lost detected.	RW	0

Table 24-31. I2C_IRQSTATUS

Address Offset	0x0000 0028		
Physical Address	0x4806 0028 0x4807 0028 0x4807 2028 0x4807 A028 0x4807 C028	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event enabled interrupt status vector		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	BB	ROVR	XUDF	AAS	BF	AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR	Transmit draining IRQ enabled status. 0x0: Transmit draining inactive. 0x1: Transmit draining enabled.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
13	RDR	Receive draining IRQ enabled status. 0x0: Receive draining inactive. 0x1: Receive draining enabled.	RW W1toClr	0
12	BB	Bus busy enabled status. Writing into this bit has no effect. Read 0x1: Bus is occupied. Read 0x0: Bus is free.	R	0
11	ROVR	Receive overrun enabled status. Writing into this bit has no effect. Read 0x1: Receiver overrun. Read 0x0: Normal operation.	RW W1toClr	0
10	XUDF	Transmit underflow enabled status. Writing into this bit has no effect. Read 0x1: Transmit underflow. Read 0x0: Normal operation.	RW W1toClr	0
9	AAS	Address recognized as slave IRQ enabled status. 0x0: No action. 0x1: Address recognized.	RW W1toClr	0
8	BF	Bus Free IRQ enabled status. 0x0: No action. 0x1: Bus Free.	RW W1toClr	0
7	AERR	Access Error IRQ enabled status. 0x0: No action. 0x1: Access Error.	RW W1toClr	0
6	STC	Start Condition IRQ enabled status. 0x0: No action. 0x1: Start Condition detected.	RW W1toClr	0
5	GC	General call IRQ enabled status. Set to 1 by core when General call address detected and interrupt signaled to IRQ_Crossbar. Write 1 to clear. 0x0: No general call detected. 0x1: General call address detected.	RW W1toClr	0
4	XRDY	Transmit data ready IRQ enabled status. Set to 1 by core when transmitter and when new data is requested. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. Write 1 to clear. 0x0: Transmision ongoing. 0x1: Transmit data ready.	RW W1toClr	0
3	RRDY	Receive data ready IRQ enabled status. Set to 1 by core when receiver mode, a new data is able to be read. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. Write 1 to clear. 0x0: No data available. 0x1: Receive data available.	RW W1toClr	0
2	ARDY	Register access ready IRQ enabled status. When set to 1 it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to IRQ_Crossbar. Write 1 to clear. 0x0: Module busy. 0x1: Access ready.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	NACK	No acknowledgement IRQ enabled status. Bit is set when No Acknowledge has been received, an interrupt is signaled to IRQ_Crossbar. Write 1 to clear this bit. 0x0: Normal operation. 0x1: Not Acknowledge detected.	RW W1toClr	0
0	AL	Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to IRQ_Crossbar. During reads, it always returns 0. 0x0: Normal operation. 0x1: Arbitration lost detected.	RW W1toClr	0

Table 24-32. I2C_IRQENABLE_SET

Address Offset	0x0000 002C		
Physical Address	0x4806 002C 0x4807 002C 0x4807 202C 0x4807 A02C 0x4807 C02C	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event interrupt enable bit vector.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	AAS_IE	BF_IE	AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	R	0
14	XDR_IE	Transmit Draining interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW[XDR]. Read: 0x0: Transmit Draining interrupt disabled 0x1: Transmit Draining interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Transmit Draining interrupt	RW W1toSet	0
13	RDR_IE	Receive Draining interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [RDR]. Read: 0x0: Receive Draining interrupt disabled 0x1: Receive Draining interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Receive Draining interrupt	RW W1toSet	0
12	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
11	ROVR	Receive overrun enable set. Read: 0x0: Receive overrun interrupt disabled 0x1: Receive overrun interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Receive overrun interrupt	RW W1toSet	0
10	XUDF	Transmit underflow enable set. Read: 0x0: Transmit underflow interrupt disabled 0x1: Transmit underflow interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Transmit underflow interrupt	RW W1toSet	0
9	AAS_IE	Addressed as Slave interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AAS]. Read: 0x0: Addressed as Slave interrupt disabled 0x1: Addressed as Slave interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Addressed as Slave interrupt	RW W1toSet	0
8	BF_IE	Bus Free interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [BF]. Read: 0x0: Bus Free interrupt disabled 0x1: Bus Free interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Bus Free interrupt	RW W1toSet	0
7	AERR_IE	Access Error interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AERR]. Read: 0x0: Access Error interrupt disabled 0x1: Access Error interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Access Error interrupt	RW W1toSet	0
6	STC_IE	Start Condition interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [STC]. Read: 0x0: Start Condition interrupt disabled 0x1: Start Condition interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Start Condition interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
5	GC_IE	General call Interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [GC] Read: 0x0: General call interrupt disabled 0x1: General call interrupt enabled Write: 0x0: Has no effect 0x1: Enables the General call interrupt	RW W1toSet	0
4	XRDY_IE	Transmit data ready interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [XRDY] Read: 0x0: Transmit data ready interrupt disabled 0x1: Transmit data ready interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Transmit data ready interrupt	RW W1toSet	0
3	RRDY_IE	Receive data ready interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTTUS_RAW [RRDY] Read: 0x0: Receive data ready interrupt disabled 0x1: Receive data ready interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Receive data ready interrupt	RW W1toSet	0
2	ARDY_IE	Register access ready interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [ARDY] Read: 0x0: Register access ready interrupt disabled 0x1: Register access ready interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Register access ready interrupt	RW W1toSet	0
1	NACK_IE	No acknowledgement interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [NACK] Read: 0x0: Not Acknowledge interrupt disabled 0x1: Not Acknowledge interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Not Acknowledge interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
0	AL_IE	Arbitration lost interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AL] Read: 0x0: Arbitration lost interrupt disabled 0x1: Arbitration lost interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Arbitration lost interrupt	RW W1toSet	0

Table 24-33. I2C_IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	0x4806 0030 0x4807 0030 0x4807 2030 0x4807 A030 0x4807 C030	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event interrupt clear bit vector.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	AAS_IE	BF_IE	AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	R	0
14	XDR_IE	Transmit Draining interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [XDR]. Read: 0x0: Transmit Draining interrupt disabled 0x1: Transmit Draining interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Transmit Draining interrupt	RW W1toClr	0
13	RDR_IE	Receive Draining interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTAUS_RAW [RDR]. Read: 0x0: Receive Draining interrupt disabled 0x1: Receive Draining interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Receive Draining interrupt	RW W1toClr	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun enable clear. Read: 0x0: Receive overrun interrupt disabled 0x1: Receive overrun interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Receive overrun interrupt	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
10	XUDF	Transmit underflow enable clear. Read: 0x0: Transmit underflow interrupt disabled 0x1: Transmit underflow interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Transmit underflow interrupt	RW W1toClr	0
9	AAS_IE	Addressed as Slave interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AAS]. Read: 0x0: Addressed as Slave interrupt disabled 0x1: Addressed as Slave interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Addressed as Slave interrupt	RW W1toClr	0
8	BF_IE	Bus Free interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [BF]. Read: 0x0: Bus Free interrupt disabled 0x1: Bus Free interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Bus Free interrupt	RW W1toClr	0
7	AERR_IE	Access Error interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AERR]. Read: 0x0: Access Error interrupt disabled 0x1: Access Error interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Access Error interrupt	RW W1toClr	0
6	STC_IE	Start Condition interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [STC]. Read: 0x0: Start Condition interrupt disabled 0x1: Start Condition interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Start Condition interrupt	RW W1toClr	0
5	GC_IE	General call Interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [GC] Read: 0x0: General call interrupt disabled 0x1: General call interrupt enabled Write: 0x0: Has no effect 0x1: Disables the General call interrupt	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
4	XRDY_IE	Transmit data ready interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [XRDY] Read: 0x0: Transmit data ready interrupt disabled 0x1: Transmit data ready interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Transmit data ready interrupt	RW W1toClr	0
3	RRDY_IE	Receive data ready interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTAUS_RAW [RRDY] Read: 0x0: Receive data ready interrupt disabled 0x1: Receive data ready interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Receive data ready interrupt	RW W1toClr	0
2	ARDY_IE	Register access ready interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [ARDY] Read: 0x0: Register access ready interrupt disabled 0x1: Register access ready interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Register access ready interrupt	RW W1toClr	0
1	NACK_IE	No acknowledgement interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [NACK] Read: 0x0: Not Acknowledge interrupt disabled 0x1: Not Acknowledge interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Not Acknowledge interrupt	RW W1toClr	0
0	AL_IE	Arbitration lost interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW[AL] Read: 0x0: Arbitration lost interrupt disabled 0x1: Arbitration lost interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Arbitration lost interrupt	RW W1toClr	0

Table 24-34. I2C_WE

Address Offset	0x0000 0034		
Physical Address	0x4806 0034 0x4807 0034 0x4807 2034 0x4807 A034 0x4807 C034	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5

Table 24-34. I2C_WE (continued)

Description	I2C wakeup enable vector.
Type	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0

Bits	Field Name	Description	Type	Reset
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

Table 24-35. I2C_DMARXENABLE_SET

Address Offset	0x0000 0038		
Physical Address	0x4806 0038 0x4807 0038 0x4807 2038 0x4807 A038 0x4807 C038	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event DMA RX enable set.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMARX_ENABLE_SET

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMARX_ENABLE_SET	Receive DMA channel enable set.	RW	0

Table 24-36. I2C_DMATXENABLE_SET

Address Offset	0x0000 003C		
Physical Address	0x4806 003C 0x4807 003C 0x4807 203C 0x4807 A03C 0x4807 C03C	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event DMA TX enable set.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMATX_ENABLE_SET

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMATX_ENABLE_SET	Transmit DMA channel enable set.	RW	0

Table 24-37. I2C_DMARXENABLE_CLR

Address Offset	0x0000 0040		
Physical Address	0x4806 0040 0x4807 0040 0x4807 2040 0x4807 A040 0x4807 C040	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event DMA RX enable clear.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DMARX_ENABLE_CLEAR

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMARX_ENABLE_CLEAR	Receive DMA channel enable clear.	RW	0

Table 24-38. I2C_DMATXENABLE_CLR

Address Offset	0x0000 0044		
Physical Address	0x4806 0044 0x4807 0044 0x4807 2044 0x4807 A044 0x4807 C044	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event DMA TX enable clear.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMATX_ENABLE_CLEAR

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMATX_ENABLE_CLEAR	Transmit DMA channel enable clear.	RW	0

Table 24-39. I2C_DMARXWAKE_EN

Address Offset	0x0000 0048		
Physical Address	0x4806 0048 0x4807 0048 0x4807 2048 0x4807 A048 0x4807 C048	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event DMA RX wakeup enable.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0

Bits	Field Name	Description	Type	Reset
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

Table 24-40. I2C_DMATXWAKE_EN

Address Offset	0x0000 004C		
Physical Address	0x4806 004C 0x4807 004C 0x4807 204C 0x4807 A04C 0x4807 C04C	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Per-event DMA TX wakeup enable.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

Table 24-41. I2C_SYSS

Address Offset	0x0000 0090		
Physical Address	0x4806 0090 0x4807 0090 0x4807 2090 0x4807 A090 0x4807 C090	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	System Status register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RDONE

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	RDONE	Reset done bit Read 0x1: Reset completed Read 0x0: Internal module reset in on-going	RW	1

Table 24-42. I2C_BUF

Address Offset	0x0000 0094		
Physical Address	0x4806 0094 0x4807 0094 0x4807 2094 0x4807 A094 0x4807 C094	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Buffer Configuration register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDMA_EN	RXFIFO_CLR	RXTRSH						XDMA_EN	TXFIFO_CLR	TXTRSH					

Bits	Field Name	Description	Type	Reset
15	RDMA_EN	Receive DMA channel enable 0x0: Receive DMA channel disabled 0x1: Receive DMA channel enabled	RW	0
14	RXFIFO_CLR	Receive FIFO clear 0x0: Normal mode 0x1: Rx FIFO is reset	RW	0
13:8	RXTRSH	Threshold value for FIFO buffer in RX mode	RW	0x00
7	XDMA_EN	Transmit DMA channel enable 0x0: Transmit DMA channel disabled 0x1: Transmit DMA channel enabled	RW	0
6	TXFIFO_CLR	Transmit FIFO clear 0x0: Normal mode 0x1: Tx FIFO is reset	RW	0
5:0	TXTRSH	Threshold value for FIFO buffer in TX mode	RW	0x00

Table 24-43. I2C_CNT

Address Offset	0x0000 0098		
Physical Address	0x4806 0098 0x4807 0098 0x4807 2098 0x4807 A098 0x4807 C098	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Data counter register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCOUNT															

Bits	Field Name	Description	Type	Reset
15:0	DCOUNT	Data count	RW	0x0000

Table 24-44. I2C_DATA

Address Offset	0x0000 009C		
Physical Address	0x4806 009C 0x4807 009C 0x4807 209C 0x4807 A09C 0x4807 C09C	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Data access register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DATA							

Bits	Field Name	Description	Type	Reset
15:8	RESERVED	Reserved	R	0x00
7:0	DATA	Transmit/Receive data FIFO endpoint	RW	0x--

Table 24-45. I2C_CON

Address Offset	0x0000 00A4		
Physical Address	0x4806 00A4 0x4807 00A4 0x4807 20A4 0x4807 A0A4 0x4807 C0A4	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C configuration register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C_EN	RESERVED	OPMODE		STB	MST	TRX	XSA	XOA0	XOA1	XOA2	XOA3	RESERVED		STP	STT

Bits	Field Name	Description	Type	Reset
15	I2C_EN	I2C module enable. 0x0: Controller in reset. FIFO are cleared and status bits are set to their default value 0x1: Module enabled	RW	0
14	RESERVED	Reserved	R	0
13:12	OPMODE	Operation mode selection. 0x0: I2C Fast/Standard mode. 0x1: I2C High Speed mode. 0x3: Reserved. 0x2: Reserved	RW	0x0
11	STB	Start byte mode (master mode only). 0x0: Normal mode 0x1: Start byte mode	RW	0
10	MST	Master/slave mode. 0x0: Slave mode 0x1: Master mode	RW	0

Bits	Field Name	Description	Type	Reset
9	TRX	Transmitter/Receiver mode (master mode only). 0x0: Receiver mode 0x1: Transmitter mode	RW	0
8	XSA	Expand Slave address. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
7	XOA0	Expand Own address 0. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
6	XOA1	Expand Own address 1. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
5	XOA2	Expand Own address 2. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
4	XOA3	Expand Own address 3. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
3:2	RESERVED	Reserved	R	0x0
1	STP	Stop condition (master mode only). 0x0: No action or stop condition detected 0x1: Stop condition queried	RW	0
0	STT	Start condition (master mode only). 0x0: No action or start condition detected 0x1: Start condition queried	RW	0

Table 24-46. I2C_OA

Address Offset	0x0000 00A8		
Physical Address	0x4806 00A8 0x4807 00A8 0x4807 20A8 0x4807 A0A8 0x4807 C0A8	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	Own address register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCODE			RESERVED			OA									

Bits	Field Name	Description	Type	Reset
15:13	MCODE	Master Code	RW	0x0
12:10	RESERVED	Reserved	R	0x0
9:0	OA	Own address	RW	0x000

Table 24-47. I2C_SA

Address Offset	0x0000 00AC		
Physical Address	0x4806 00AC 0x4807 00AC 0x4807 20AC 0x4807 A0AC 0x4807 C0AC	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5

Table 24-47. I2C_SA (continued)

Description	Slave address register
Type	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						SA									

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	SA	Slave address	RW	0x3FF

Table 24-48. I2C_PSC

Address Offset	0x0000 00B0		
Physical Address	0x4806 00B0 0x4807 00B0 0x4807 20B0 0x4807 A0B0 0x4807 C0B0	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C Clock Prescaler Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PSC							

Bits	Field Name	Description	Type	Reset
15:8	RESERVED	Reserved	R	0x00
7:0	PSC	Fast/Standard mode prescale sampling clock divider value 0x0: Divide by 1 0x1: Divide by 2 .... 0xFF: Divide by 256	RW	0x00

Table 24-49. I2C_SCLL

Address Offset	0x0000 00B4		
Physical Address	0x4806 00B4 0x4807 00B4 0x4807 20B4 0x4807 A0B4 0x4807 C0B4	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C SCL Low Time Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSSCLL								SCLL							

Bits	Field Name	Description	Type	Reset
15:8	HSSCLL	High speed mode SCL low time The value of the bit field is automatically increased by 7.	RW	0x00
7:0	SCLL	Fast/standard mode SCL low time The value of the bit field is automatically increased by 7.	RW	0x00

Table 24-50. I2C_SCLH

Address Offset	0x0000 00B8

Table 24-50. I2C_SCLH (continued)

Physical Address	0x4806 00B8 0x4807 00B8 0x4807 20B8 0x4807 A0B8 0x4807 C0B8	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C SCL High Time Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSSCLH								SCLH							

Bits	Field Name	Description	Type	Reset
15:8	HSSCLH	High speed mode SCL high time The value of the bit field is automatically increased by 5.	RW	0x00
7:0	SCLH	Fast/standard mode SCL high time The value of the bit field is automatically increased by 5.	RW	0x00

Table 24-51. I2C_SYSTEST

Address Offset	0x0000 00BC		
Physical Address	0x4806 00BC 0x4807 00BC 0x4807 20BC 0x4807 A0BC 0x4807 C0BC	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C System Test Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_EN	FREE	TMODE		SSB	RESERVED		SCL_I_FUNC	SCL_O_FUNC	SDA_I_FUNC	SDA_O_FUNC	RESERVED	SCL_I	SCL_O	SDA_I	SDA_O

Bits	Field Name	Description	Type	Reset
15	ST_EN	System test enable. 0x0: Normal mode. All others bits in register are read only 0x1: System test enabled. Permit other system test registers bits to be set	RW	0
14	FREE	Free running mode (on breakpoint) 0x0: Stop mode (on breakpoint condition). If Master mode, it stops after completion of the ongoing bit transfer. In slave mode, it stops during the phase transfer when 1 byte is completely transmitted/received. 0x1: Free running mode	RW	0
13:12	TMODE	Test mode select. 0x0: Functional mode (default) 0x1: Reserved 0x3: Loop back mode select + SDA/SCL IO mode select 0x2: Test of SCL counters (SCLL, SCLH, PSC). SCL provides a permanent clock with master mode.	RW	0x0
11	SSB	Set all status bits in I2C_IRQSTATUS_RAW [14:0]. 0x0: No action 0x1: Set interrupt status bits to 1.	RW	0

Bits	Field Name	Description	Type	Reset
10:9	RESERVED	Reserved	R	0x0
8	SCL_I_FUNC	SCL line input value (functional mode). Read 0x1: Read 1 from SCL line Read 0x0: Read 0 from SCL line	R	1
7	SCL_O_FUNC	SCL line output value (functional mode). Read 0x1: Driven 1 on SCL line Read 0x0: Driven 0 on SCL line	R	1
6	SDA_I_FUNC	SDA line input value (functional mode). Read 0x1: Read 1 from SDA line Read 0x0: Read 0 from SDA line	R	1
5	SDA_O_FUNC	SDA line output value (functional mode). Read 0x1: Driven 1 to SDA line Read 0x0: Driven 0 to SDA line	R	1
4	RESERVED	Reserved	R	0
3	SCL_I	SCL line sense input value Read 0x1: Read 1 from SCL line Read 0x0: Read 0 from SCL line	R	0
2	SCL_O	SCL line drive output value. 0x0: Write 0 to SCL line 0x1: Write 1 to SCL line	RW	0
1	SDA_I	SDA line sense input value. Read 0x1: Read 1 from SDA line Read 0x0: Read 0 from SDA line	R	0
0	SDA_O	SDA line drive output value. 0x0: Write 0 to SDA line 0x1: Write 1 to SDA line	RW	0

Table 24-52. I2C_BUFSTAT

Address Offset	0x0000 00C0		
Physical Address	0x4806 00C0 0x4807 00C0 0x4807 20C0 0x4807 A0C0 0x4807 C0C0	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C Buffer Status Register.		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFODEPTH		RXSTAT						RESERVED		TXSTAT					

Bits	Field Name	Description	Type	Reset
15:14	FIFODEPTH	Internal FIFO buffers depth. Read 0x0: 8-bytes FIFO. Read 0x1: 16-bytes FIFO. Read 0x2: 32-bytes FIFO. Read 0x3: 64-bytes FIFO.	R	0x1
13:8	RXSTAT	RX Buffer Status	R	0x00
7:6	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
5:0	TXSTAT	TX Buffer Status.	R	0x00

Table 24-53. I2C_OA1

Address Offset	0x0000 00C4		
Physical Address	0x4806 00C4 0x4807 00C4 0x4807 20C4 0x4807 A0C4 0x4807 C0C4	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C Own Address 1 Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						OA1									

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	OA1	Own address 1	RW	0x000

Table 24-54. I2C_OA2

Address Offset	0x0000 00C8		
Physical Address	0x4806 00C8 0x4807 00C8 0x4807 20C8 0x4807 A0C8 0x4807 C0C8	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C Own Address 2 Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						OA2									

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	OA2	Own address 2	RW	0x000

Table 24-55. I2C_OA3

Address Offset	0x0000 00CC		
Physical Address	0x4806 00CC 0x4807 00CC 0x4807 20CC 0x4807 A0CC 0x4807 C0CC	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C Own Address 3 Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						OA3									

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	OA3	Own address 3	RW	0x000

Table 24-56. I2C_ACTOA

Address Offset	0x0000 00D0		
Physical Address	0x4806 00D0 0x4807 00D0 0x4807 20D0 0x4807 A0D0 0x4807 C0D0	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C Active Own Address Register.		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OA3_ACT	OA2_ACT	OA1_ACT	OA0_ACT

Bits	Field Name	Description	Type	Reset
15:4	RESERVED	Reserved	R	0x000
3	OA3_ACT	Own Address 3 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
2	OA2_ACT	Own Address 2 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
1	OA1_ACT	Own Address 1 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
0	OA0_ACT	Own Address 0 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0

Table 24-57. I2C_SBLOCK

Address Offset	0x0000 00D4		
Physical Address	0x4806 00D4 0x4807 00D4 0x4807 20D4 0x4807 A0D4 0x4807 C0D4	**Instance**	I2C3 I2C1 I2C2 I2C4 I2C5
Description	I2C Clock Blocking Enable Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OA3_EN	OA2_EN	OA1_EN	OA0_EN

Bits	Field Name	Description	Type	Reset
15:4	RESERVED	Reserved	R	0x000
3	OA3_EN	Enable I2C Clock Blocking for Own Address 3. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
2	OA2_EN	Enable I2C Clock Blocking for Own Address 2. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0

Bits	Field Name	Description	Type	Reset
1	OA1_EN	Enable I2C Clock Blocking for Own Address 1. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
0	OA0_EN	Enable I2C Clock Blocking for Own Address 0. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0

24.2 HDQ/1-Wire

This section describes the HDQ™/ 1-Wire® interface for the device.

24.2.1 HDQ1W Overview

The HDQ1W module implements the hardware protocol of the master functions of the TI/Benchmarq HDQ and the Dallas Semiconductor 1-Wire® protocols. These protocols use a single wire for communication between the master (HDQ1W controller) and the slaves (HDQ/1-Wire external compliant devices).

Figure 24-28 shows the HDQ1W.

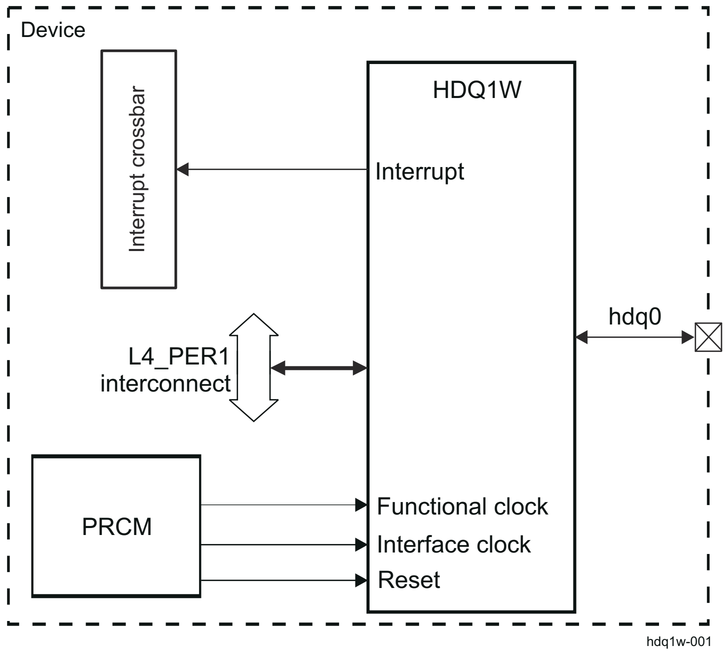


Figure 24-28. HDQ1W Overview

The HDQ1W has a generic L4 interface and is intended to be used in an interrupt-driven fashion. The 1-pin interface is implemented as an open-drain output at the device level.

The main features supported by the HDQ1W are the following:

- Benchmarq HDQ protocol
- Dallas Semiconductor 1-Wire protocol
- Power-down mode

The HDQ1W provides a communication rate of 5 Kbps over an address space of 128 bytes.

A typical application of the HDQ1W is the communication with battery monitor (gas gauge) integrated circuits.

24.2.2 HDQ1W Environment

24.2.2.1 HDQ1W Functional Modes

The HDQ1W has two main modes: HDQ and 1-Wire. Each of these modes includes idle, active, and power-down submodes. Table 24-58 lists the HDQ1W functional modes, and Figure 24-29 shows an overview of a typical application.

Table 24-58. Functional Modes

Functions	Description
HDQ	Benchmarq HDQ protocol
1-Wire	Dallas Semiconductor 1-Wire protocol

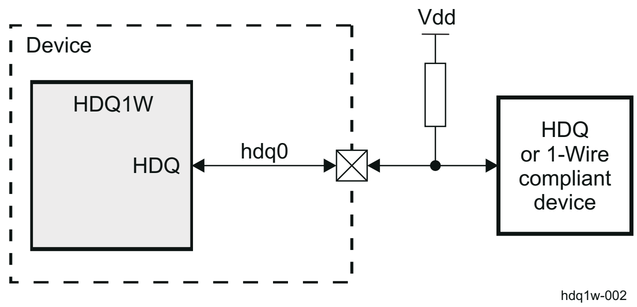


Figure 24-29. HDQ1W Typical Application System Overview

An external pullup is required, because the two protocols use a return-to-1 mechanism (that is, after any command by any of the connected devices, the line is pulled to a logical high level).

The HDQ1W operates according to a command structure that is programmed into transmit command registers (as described in Section 24.2.5.1.2, *HDQ1W Low-level Programming Model*).

The 1-Wire mode runs at slower speeds than the capabilities of the mode.

Table 24-59 describes the external signal of the HDQ/1-Wire compliant module.

Table 24-59. I/O Description

Signal	I/O(1)	Description	Value at Reset
hdq_sio	I/O	Serial data input/output. Output is open drain type.	HiZ (pulled to 1 by pullup)

(1) I = Input; O = Output

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see *Pad Configuration Registers* in *Control Module*.

24.2.2.2 HDQ and 1-Wire (SDQ) Protocols

24.2.2.2.1 HDQ Protocol Initialization (Default)

In HDQ mode, the firmware does not require the host to create an initialization pulse to the slave. However, the slave can be reset by using an initialization pulse (also referred to as a break pulse). The initialization pulse is generated by setting the HDQ_CTRL_STATUS[2] INITIALIZATION bit and then setting the HDQ_CTRL_STATUS[4] GO bit. The slave does not respond with a presence pulse as it does in the 1-Wire protocol.

The HDQ is a command-based protocol in which the host sends a command byte to the slave. The command directs the slave either to store the next eight bits of data received to a register specified by the command byte (write operation) or to output the eight bits of data from a register specified by the command byte (read operation). The master implementation is a simple byte engine. Sending of the ID, command/address, and data is controlled by firmware. The master engine provides only a single HDQ_TX_DATA register.

The command and data bytes consist of a stream of eight bits with a maximum transmission rate of 5 Kbps. The least-significant bit (LSB) of a command or data byte is transmitted first. If a communication time-out occurs between the host and the slave (for example, if the host waits longer than the specified time for the slave to respond, or if this is the first access command), then the host must send an initialization pulse (BREAK) before sending the command again.

The slave detects a break when the HDQ pin is driven to a logic-low state for a specified break time $t_{(B)}$ or greater. The HDQ pin then returns to its normal ready-high logic state for a specified break-recovery time $t_{(BR)}$. The slave is then ready for a command from the host processor. Figure 24-30 shows this behavior.

An interrupt condition indicates a TX-complete, an RX-complete, or a time-out condition. Reading the interrupt status register clears all interrupt conditions. Only one interrupt signal is sent to the microcontroller, and only one overall mask bit can enable or disable the interrupt. The interrupt conditions cannot be individually masked.

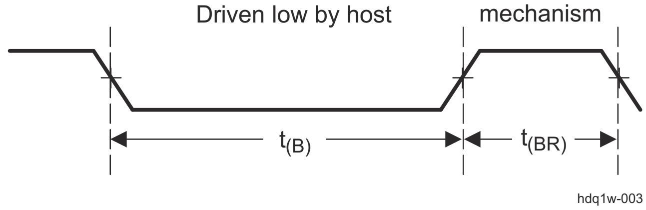


Figure 24-30. HDQ Break-Pulse Timing Diagram

24.2.2.2.2 1-Wire (SDQ) Protocol Initialization

In 1-Wire (SDQ) protocol, the host first sends an initialization pulse (by pulling the line to a logic-low state) and then waits for the slave to respond with a presence pulse before enabling any communication sequence.

As for the initialization pulse, the presence pulse is a low-level pulse on the line initiated by the slave. The timing diagram in Figure 24-31 shows the 1-Wire (SDQ) reset sequence.

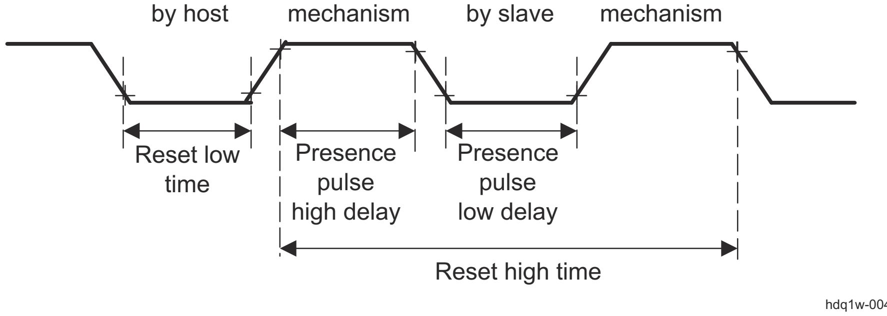


Figure 24-31. 1-Wire (SDQ) Reset Timing Diagram

The host drives the line to a logic-low state for a minimum of reset low time. Once the slave detects this pulse, it must drive the line to a logic-low state within the presence pulse high delay for a minimum period of presence pulse low time.

If the slave does not respond within this interval of time, a time-out event occurs and no transaction can be initiated. The host must initiate the reset sequence again before sending any command to the slave.

On the other hand, if the slave sends back its presence pulse within the specified time interval, the communication can be enabled after the reset high time.

24.2.2.2.3 Communication Sequence (HDQ and 1-Wire Protocols)

The description in this section applies to both protocols.

After a successful break pulse (HDQ mode) or initialization sequence (1-Wire protocol), the host and slave are ready for bit transmission. Each bit to transmit (either from the host to the slave or from the slave to the host) is preceded by a low-going edge on the line, as shown in Figure 24-32.

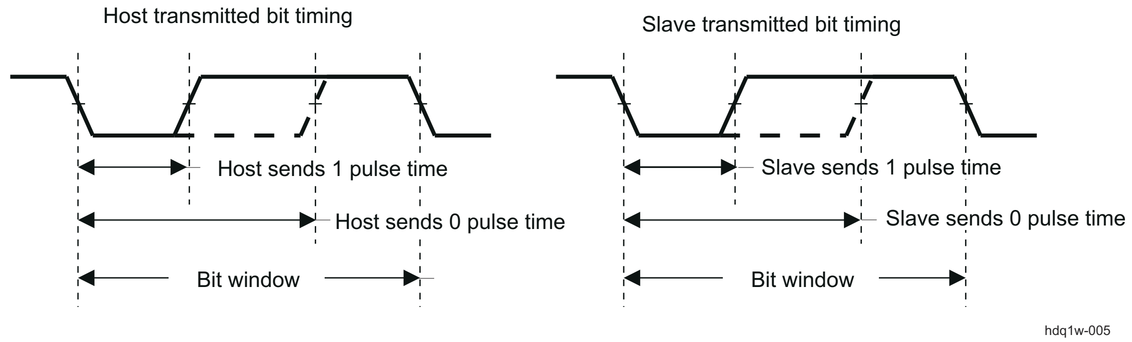


Figure 24-32. HDQ1W Transmitted Bit Timing

The return-to-1 data-bit frame consists of three distinct sections. The first section starts the transmission when either the slave or the host takes the line to a logic-low state. The next section is the actual data transmission in which the data must be valid during a specified period of time after the negative edge that starts the communication. The final section stops the transmission by returning the HDQ/1-Wire line to a logic-high state. Communication with an HDQ/1-Wire slave always occurs with the LSB being transmitted first.

The command byte of the HDQ/1-Wire protocols consists of eight contiguous valid command bits. The command byte contains two fields: R/W command and address. The R/W bit of the command byte determines whether the command is a read or a write, and the address field containing bits AD6-AD0 indicates the address to be read or written. Table 24-60 lists the command byte values.

Table 24-60. HDQ/1-Wire Command Byte

7	6	5	4	3	2	1	0
R/W	AD6	AD5	AD4	AD3	AD2	AD1	AD0

R/W Indicates whether the command byte is a read or a write. A 1 indicates a write command; the following eight bits must be written to the register specified by the address field of the command byte. A 0 indicates that the command is a read. On a read command, the slave outputs the requested register contents.

AD6-AD0 Represent the seven bits labeled AD6-AD0 containing the address portion of the register to be accessed. The communication sequence example in Figure 24-33 shows a read command at address 0x55; the received data is 0x65.

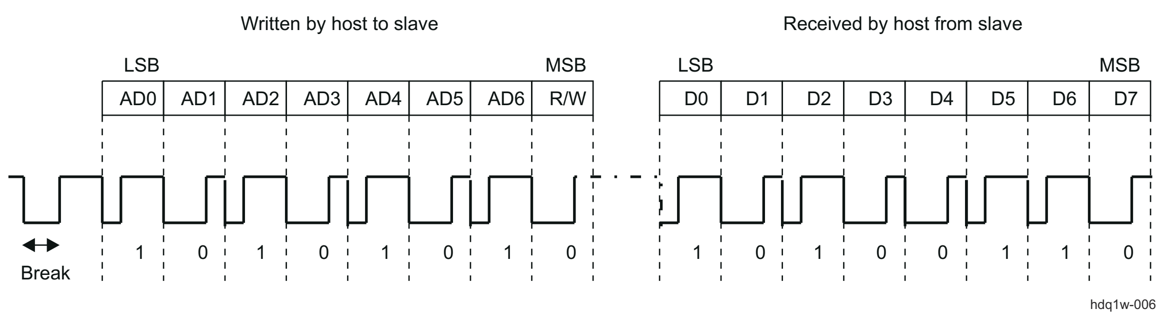


Figure 24-33. HDQ/1-Wire Communication Sequence

24.2.3 HDQ1W Integration

Figure 24-34 shows HDQ1W integration in the device.

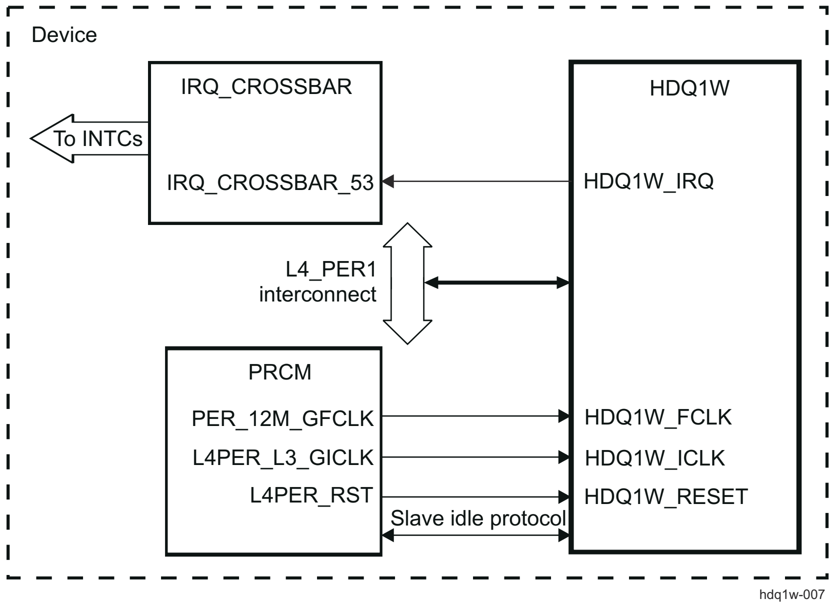


Figure 24-34. HDQ1W Integration

Table 24-61 through Table 24-63 summarize the integration of the module in the device.

Table 24-61. HDQ1W Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
HDQ1W	PD_COREAON	No	L4_PER1

Table 24-62. HDQ1W Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
HDQ1W	HDQ1W_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	HDQ1W_FCLK	PER_12M_GFCLK	PRCM	Functional clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
HDQ1W	HDQ1W_RESET	L4PER_RST	PRCM	HDQ1W reset signal

Table 24-63. HDQ1W Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
HDQ1W	HDQ1W_IRQ	IRQ_CROSSBAR_53	MPU_IRQ_58	HDQ1W interrupt request
			DSP1_IRQ_84	
			DSP2_IRQ_84	

Note

The **“Default Mapping”** column in Table 24-63, *HDQ1W Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.

For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

24.2.4 HDQ1W Functional Description

The HDQ1W works with HDQ and 1-Wire protocols. The protocols use a single wire to establish communication between the master and the slave. Both protocols use a return-to-1 mechanism; that is, after any command is driven, the line is pulled to a high level. This mechanism requires an external pullup.

24.2.4.1 HDQ1W Block Diagram

Figure 24-35 is the HDQ1W block diagram.

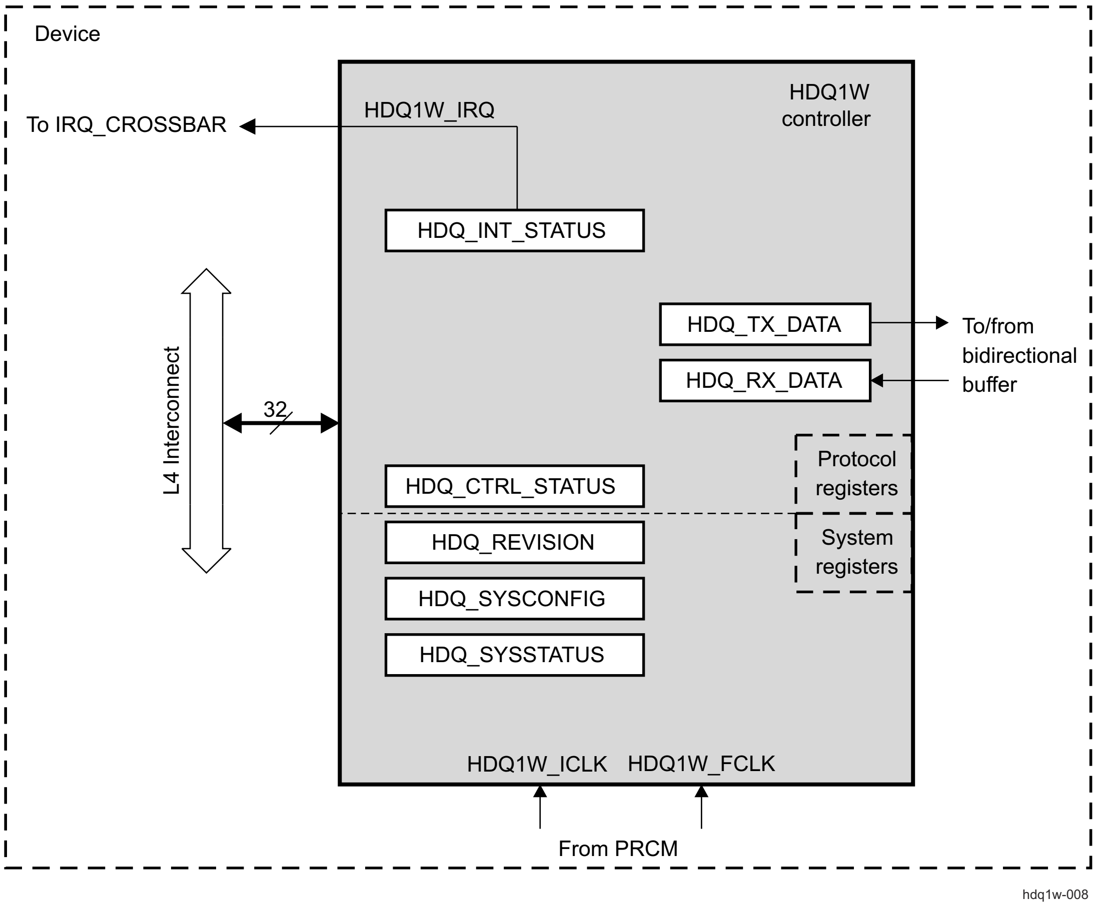


Figure 24-35. HDQ1W Block Diagram

The HDQ_CTRL_STATUS[0] MODE bit allows selection between the HDQ and 1-Wire protocols. This bit is assumed static for design purposes. The configuration is in HDQ mode by default.

Figure 24-36 shows the protocol-dedicated register scheme.

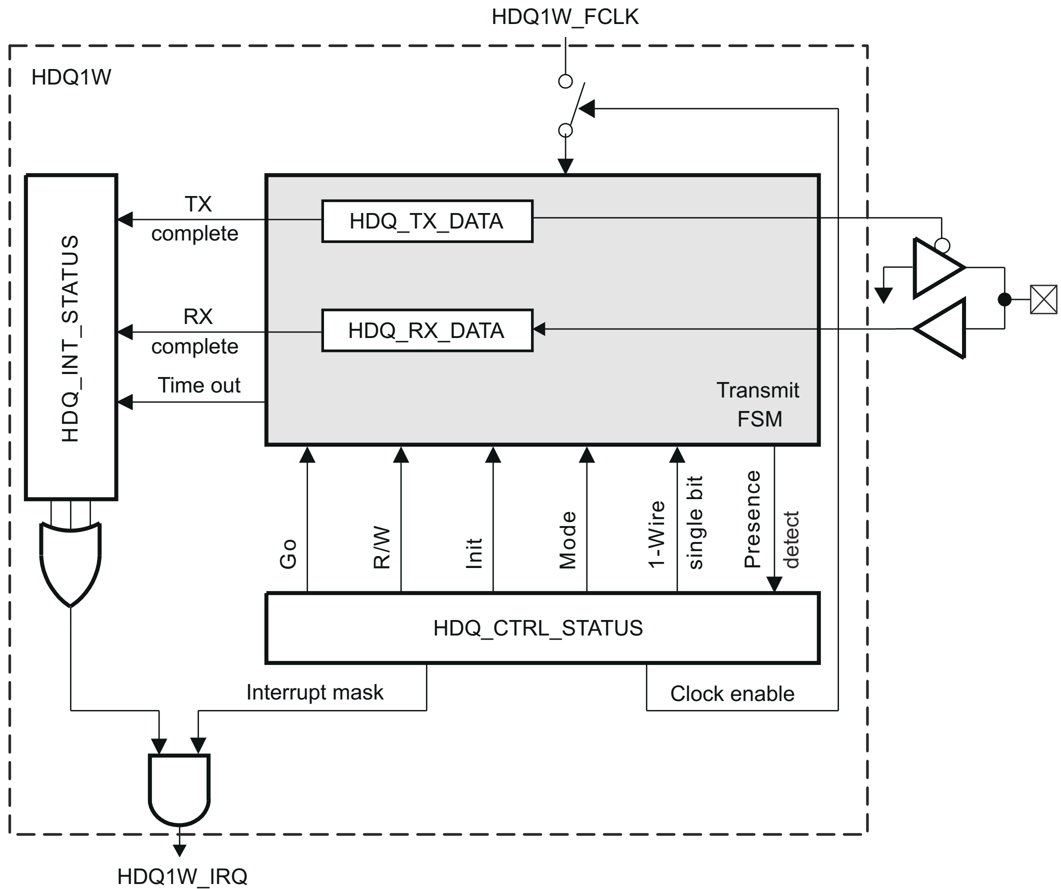


Figure 24-36. Protocol Registers Description

The receive and transmit operations of the HDQ1W module are performed with respect to the timing of the slower HDQ protocol. When the 1-Wire protocol is used, it runs at lower speed than its full capabilities, but is still able to meet the timing requirements and practical considerations.

24.2.4.2 HDQ1W Clocking Configuration

24.2.4.2.1 HDQ1W Clocks

The HDQ1W module operates from two clocks: a functional clock (HDQ1W_FCLK) and an interface clock (HDQ1W_ICLK). When these clocks are set in the PRCM module, the following rule must be observed: HDQ1W_ICLK ≥ HDQ1W_FCLK.

- The HDQ1W_FCLK functional clock is a fixed clock provided by the PRCM module. It is used to clock the internal module logic.

  For more information about the clock and the PRCM register settings, see *CD_L4PER1 Clock Domain*, in chapter *Power, Reset, and Clock Management*.

  When the HDQ1W no longer requires the HDQ1W_FCLK, the software can disable it at the PRCM level. The clock is effectively cut, provided the other modules that receive it do not require it either.
- The HDQ1W_ICLK interface clock runs at L4 interconnect clock speed and is used to trigger access to the HDQ1W L4 interface.

  When the HDQ1W no longer requires the HDQ1W_ICLK (no transfer is in progress), the software can disable it at the PRCM level. The clock is effectively cut, provided the other modules that receive it do not require it either.

24.2.4.3 HDQ1W Hardware and Software Reset

Global reset of the module is done at the power, reset, and clock management (PRCM) module level (for more information, see *CD_L4PER1 Clock Domain*) or by setting the HDQ_SYSCONFIG[1] SOFTRESET bit to 1. Setting this bit enables an active software reset functionality equivalent to a hardware reset. The HDQ1W_FCLK functional clock must be enabled from the PRCM level and locally through the HDQ_CTRL_STATUS[5] CLOCKENABLE bit (set to 1) for the software reset to complete.

24.2.4.4 HDQ1W Power Management

Table 24-64 describes power-management features available to the HDQ1W.

Table 24-64. Local Power-Management Features

Feature	Registers	Description
Clock auto gating	HDQ_SYSCONFIG [0] AUTOIDLE bit	Auto-idle mode
Slave idle modes	N/A	N/A
Clock enable	HDQ_CTRL_STATUS[5] CLOCKENABLE bit	Power-down mode
Master standby modes	N/A	N/A
Global wake-up enable	N/A	N/A
Wake-up sources enable	N/A	N/A

24.2.4.4.1 Auto-Idle Mode

The HDQ1W provides an auto-idle function in its interconnect clock domain.

The interconnect clock auto-idle power-saving mode is enabled or disabled through the HDQ_SYSCONFIG[0] AUTOIDLE bit. When this mode is enabled and there is no activity on the interconnect interface, the interconnect clock (HDQ1W_ICLK) is disabled inside the module, thereby reducing power consumption. When there is new activity on the interconnect interface, the interconnect clock is restarted with no latency penalty. This mode is disabled by default after a reset.

The auto-idle mode can be enabled in order to reduce power consumption.

24.2.4.4.2 Power-Down Mode

The HDQ1W also provides a power-saving function in its functional clock domain.

Setting the CLOCKENABLE bit in the control and status register (HDQ_CTRL_STATUS[5] CLOCKENABLE bit) to 0 shuts off the functional clock (HDQ1W_FCLK) to the state-machine. The state-machine is reset when the functional clock is disabled; if any transaction is ongoing, it is aborted into the reset state.

Before shutting off the functional clock, the software must wait for transaction-complete interrupt. In write operation the software must check whether the interrupt was generated after address/command byte was sent or after data byte was sent. The functional clock must not be shut off after address/command byte is sent; otherwise, the data is not written to the slave.

The register values are not affected by disabling the functional clock.

24.2.4.4.3

CAUTION

There is no hardware mechanism to prevent cutting off the HDQ1W clocks while the module is performing a transfer. This would result in loss of data being transferred.

24.2.4.5 HDQ Interrupt Requests

The HDQ1W can generate one interrupt:

- HDQ1W_IRQ: Table 24-65 lists the events that can generate this interrupt.

Table 24-65. Events

Event Flag	Event Mask	Sync	Sensitivity	Description
HDQ_INT_STATUS[2] TXCOMPLETE	HDQ_CTRL_STATUS[31] INTERRUPTMASK	Yes	Level	A write operation of one byte was completed.
HDQ_INT_STATUS[1] RXCOMPLETE	HDQ_CTRL_STATUS[31] INTERRUPTMASK	Yes	Level	A byte has been successfully read.
HDQ_INT_STATUS[0] TIMEOUT	HDQ_CTRL_STATUS[31] INTERRUPTMASK	Yes	Level	After a read command initiated by the host, the slave did not pull the line low within the specified time.

24.2.4.6 HDQ Mode (Default)

24.2.4.6.1 HDQ Mode Features

The HDQ mode supports the following:

- Benchmarq HDQ protocol
- Power-down mode

24.2.4.6.2 Description

In the HDQ mode, there is no need for the host to create an initialization pulse to the slave. However, the host can reset the slave by using an initialization pulse (also known as a break pulse). Setting the HDQ_CTRL_STATUS[2] INITIALIZATION bit and then setting the HDQ_CTRL_STATUS[4] GO bit creates this pulse by pulling the line down for a defined duration. When the slave receives the pulse, it is ready for communication but does not respond with a presence pulse.

In a typical write operation, two bytes are sent to the slave. The first byte corresponds to the command/address byte, and the second byte corresponds to the data to be written.

In a typical read operation, the host sends a command/address byte and the slave returns a byte of data.

The master is implemented to send and receive bytes. Sending the command/address and data is controlled by the firmware. The master provides only a single data TX register.

The HDQ protocol is a return-to-1 protocol. Consequently, after a byte is sent to the slave (either command/address + data for a write, or just command/address for a read), the host pulls the line up. The line is set to the high-impedance state in the device and an external pullup brings it to a logical high level.

In the case of a read operation, the slave also drives the line to a logic-low state before sending the requested data.

If the host initiates a read and does not receive data within a specified interval of time (that is, the slave does not drive the line low within this interval), the HDQ_INT_STATUS[0] TIMEOUT bit is set, thereby indicating a read failure. The TIMEOUT bit remains set until the host reads the interrupt status register (HDQ_INT_STATUS).

An interrupt condition indicates either a TX-complete, an RX-complete, or a time-out on a transaction. The corresponding bit is set in the interrupt status register (HDQ_INT_STATUS). This register is cleared as soon as it is read.

Only one interrupt signal is sent, and only an overall mask can enable or disable the interrupts. These interrupts cannot be individually masked.

24.2.4.6.3 Single-Bit Mode

In HDQ mode, the single-bit mode (HDQ_CTRL_STATUS[7] ONE_WIRE_SINGLE_BIT bit set to 1) has no effect because the HDQ protocol supports only byte transfers.

24.2.4.6.4 Interrupt Conditions

The HDQ1W provides the following interrupt status:

- Transmission complete:

A write operation of one byte was completed. Successful or failed completion is not indicated, because there is no acknowledgment from the slave in HDQ protocol. This interrupt condition is cleared by reading the interrupt status register (HDQ_INT_STATUS).

- Read complete:

  In HDQ mode, the interrupt status indicates that a byte has been successfully read. This interrupt condition is cleared by reading the interrupt status register (HDQ_INT_STATUS).

- Presence detect/time-out:

  In HDQ mode, the interrupt status indicates that after a read command initiated by the host, the slave did not pull the line low within the specified time. This interrupt condition is cleared by reading the interrupt status register (HDQ_INT_STATUS).

  In HDQ mode, a time-out condition is also used to indicate the successful completion of a break pulse. That is, if the master has sent the break pulse, it is indicated with a time-out instead of a TX-complete.

Only one interrupt is generated to the host CPU based on any of these interrupt conditions. A read operation on the interrupt status register clears all the interrupt status bits that were previously set.

24.2.4.7 1-Wire Mode

24.2.4.7.1 1-Wire Mode Features

The 1-Wire mode supports the following:

- Dallas Semiconductor 1-Wire protocol
- Power-down mode
- Single-bit mode

24.2.4.7.2 Description

The 1-Wire mode requires an initialization pulse to be sent to the slave(s) connected on the interface. If a slave is present, it responds with a presence pulse.

The initialization pulse is sent when the HDQ_CTRL_STATUS[2] INITIALIZATION bit is set and the HDQ_CTRL_STATUS[4] GO bit is set afterwards.

When the slave receives the initialization pulse, it sends back its presence pulse by pulling down the line for a defined duration. The module detects this low-level pulse and sets the HDQ_CTRL_STATUS[3] PRESENCEDETECT bit.

In a similar way, if a presence pulse is not received from the slave after an initialization pulse is sent, the PRESENCEDETECT bit remains cleared.

Whether or not a presence pulse is detected after an initialization pulse is sent, the HDQ_INT_STATUS[0] TIMEOUT bit is set and an interrupt condition is generated.

In 1-Wire mode, the generated interrupt condition means the maximum time allowed for receiving the response has elapsed and the software must check the PRESENCEDETECT bit to determine whether or not there was a presence pulse.

The INITIALIZATION bit is cleared at the end of the initialization pulse at the same time as the TIMEOUT bit is set. The TIMEOUT bit is cleared when the interrupt status register (HDQ_INT_STATUS) is read.

For read operations, 1-Wire is a bit-by-bit protocol, which means the slave must be clocked by the host for each bit of the byte to read.

The line is pulled up at the end of the command/address byte. On the first read, the host creates a low-going edge to initiate a bit read. The line is then pulled up (pulled to the high-impedance state by the host and set to a high logical level by the external pullup) and the slave either drives the line low to transmit a 0, or does not drive the line to transmit a 1. This sequence is repeated for each bit to read.

The first bit the host receives is the LSB, and the last bit is the most-significant bit (MSB) in the receive data register (HDQ_RX_DATA).

An interrupt condition indicates either a TX-complete, an RX-complete, or a time-out condition (that is, the time allowed for the slave to indicate its presence has elapsed). A read operation on the interrupt status register clears the interrupt conditions previously set. As in the HDQ mode, only one interrupt signal is sent to the host CPU. Only an overall mask bit can enable or disable the interrupt (the interrupt conditions cannot be masked individually).

24.2.4.7.3 1-Wire Single-Bit Mode Operation

A single-bit mode can be entered by setting the appropriate bit in the control and status register (ONE_WIRE_SINGLE_BIT bit HDQ_CTRL_STATUS[7]). In this mode, only one bit of data at a time is transferred between the master and the slave. After the bit is transferred, an interrupt is generated (that is, there is an RX-complete for a read operation and a TX-complete for a write operation). The ONE_WIRE_SINGLE_BIT bit is cleared by hardware after every single bit is received. Software must set this bit to re-enable reception in single-bit mode. Bit 0 of the RX register (HDQ_RX_DATA) is updated each time a bit is received from the slave; bit 0 of the TX register (HDQ_TX_DATA) contains the bit to be sent.

24.2.4.7.4 Interrupt Conditions

The HDQ1W provides the following interrupt status:

- Transmission complete:

  A write operation of one byte was completed. Successful or failed completion is not indicated, because there is no acknowledgment from the slave in 1-Wire protocol. This interrupt condition is cleared by reading the interrupt status register (HDQ_INT_STATUS).
- Read complete:

  In 1-Wire mode, the interrupt status indicates that a byte has been successfully read. This interrupt condition is cleared by reading the interrupt status register (HDQ_INT_STATUS).
- Presence detect/time-out:

  In 1-Wire mode, the interrupt status indicates that it is now valid to check the PRESENCEDETECT bit. This interrupt condition is cleared by reading the interrupt status register (HDQ_INT_STATUS).

Only one interrupt is generated to the host CPU based on any of these interrupt conditions. A read operation on the interrupt status register clears all interrupt status bits that were previously set.

24.2.4.7.5 Status Flags

The presence-condition-detected status flag is contained in the HDQ_CTRL_STATUS[3] PRESENCEDETECT bit. This is valid only in 1-Wire mode. The flag is updated when the HDQ_INT_STATUS[0] TIMEOUT bit is set. Therefore, its correct value shows only after the interrupt is generated. The firmware must wait for the time-out condition; otherwise, the flag keeps its previous value and is undefined.

24.2.4.8 BITFSM Delay

The return-to-one mechanism on the HDQ/1-Wire bus is a simple pull-up resistor. Consequently, an excessive wire load of the HDQ/1-Wire bus can cause a significant delay to the bus rise time. This can prevent the module state machine from working correctly by reading back an improper value caused by the line delay. To correct such condition by software, it is possible to configure the HDQ_CTRL_STATUS[10:8] BITFSM register bitfield with the expected line delay. This way the module state machine waits the proper time interval, before reading back the line value. The delay can be adjusted in 1.33 µs steps. The default value of BITFSM = 0x0 corresponds to 1.33 µs delay.

Bus delay can be calculated as follows: $T_{delay} \approx 2.2 \times R_{pullup} \times C_{line}$

See more information in the *Device Data Manual*.

24.2.5 HDQ1W Low-Level Programming Model

This section describes the low-level hardware programming sequences for configuration and usage of the module. The basic protocol functions, such as slave initialization (reset), read-byte, and write-byte operations, are described. For a description of the high-level functions, see the HDQ/1-Wire protocol documentation and the slave device datasheet.

24.2.5.1 Global Initialization

24.2.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the HDQ1W module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the HDQ1W. Refer to the HDQ1W Module Integration and Environment Sections for further information.

Table 24-66. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. The interface clock must not be slower than the functional clock (HDQ1W_ICLK ≥ HDQ1W_FCLK). For more information about the module configuration, see *Power, Reset, and Clock Management*.
Control Module	Module specific pad muxing and pullup must be set in the control module. If an external pullup is used, the internal pullup/pulldown must be disabled. For more information about the module configuration, see *Control Module*.
IRQ_CROSSBAR/INTC	Interrupt crossbar and interrupt controller configuration must be done to enable the interrupts from HDQ1W module. See *Interrupt Controllers*, and *IRQ_CROSSBAR Module Functional Description*.

24.2.5.1.2 HDQ1W Module Global Initialization

Table 24-67. HDQ1W Module Global Initialization

Step	Register/Bit Field/Programming Model	Value
Initiate software reset.	HDQ_SYSCONFIG[1] SOFTRESET	0x1
Disable power-down mode.	HDQ_CTRL_STATUS[5] CLOCKENABLE	0x1
Wait until reset complete?	HDQ_SYSSTATUS[0] RESETDONE	= 0x1
Disable power-down mode.	HDQ_CTRL_STATUS[5] CLOCKENABLE	0x1
Configure auto-idle mode.	HDQ_SYSCONFIG[0] AUTOIDLE	0x-

24.2.5.2 HDQ Operational Modes Configuration

24.2.5.2.1 Main Sequence - HDQ Write Operation Mode

Table 24-68. HDQ Mode Selection

Step	Register/Bit Field/Programming Model	Value
Select HDQ mode.	HDQ_CTRL_STATUS[0] MODE	0x0
Enable interrupt generation.	HDQ_CTRL_STATUS[6] INTERRUPTMASK	0x1
Initialize HDQ slave.	See Section 24.2.5.2.2.1	

Table 24-69. HDQ Write Operation Mode

Step	Register/Bit Field/Programming Model	Value
Write command/address or data value.	HDQ_TX_DATA[7:0]	0x-
Select write operation.	HDQ_CTRL_STATUS[1] DIR	0x0
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Reading HDQ_INT_STATUS clears interrupt conditions.	HDQ_INT_STATUS[2] TXCOMPLETE	0x1

24.2.5.2.2 Main Sequence - HDQ Read Operation Mode

24.2.5.2.2.1 Sub-sequence - Initialize HDQ Slave

Table 24-70. HDQ Read Operation Mode

Step	Register/Bit Field/ Programming Model	Value
Select read operation.	HDQ_CTRL_STATUS[1] DIR	0x1
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Read and store HDQ_INT_STATUS. Reading HDQ_INT_STATUS clears interrupt conditions.	HDQ_INT_STATUS	0x-
IF: Read operation successful?	HDQ_INT_STATUS[1] RXCOMPLETE	= 0x1
	HDQ_INT_STATUS[0] TIMEOUT	= 0x0
Get received data.	HDQ_RX_DATA[7:0]	0x-
ENDIF		

Table 24-71. Initialize HDQ Slave

Step	Register/Bit Field/Programming Model	Value
Send Initialization Pulse	HDQ_CTRL_STATUS[2] INITIALIZATION	0x1
Send Command	HDQ_CTRL_STATUS[4] GO	0x1

24.2.5.3 1-Wire Operational Modes Configuration

24.2.5.3.1 Main Sequence - 1-Wire Write Operation Mode

Table 24-72. 1-Wire Mode Selection

Step	Register/Bit Field/Programming Model	Value
Reset HDQ1W module.	See Section 24.2.5.1.2.	
Select 1-Wire mode.	HDQ_CTRL_STATUS[0] MODE	0x1
Enable interrupt generation.	HDQ_CTRL_STATUS[6] INTERRUPTMASK	0x1
Initialize 1-Wire slave, check for slave presence.	See Section 24.2.5.3.3.	

Table 24-73. 1-Wire Write Operation Mode

Step	Register/Bit Field/ Programming Model	Value
Write ID/command or data value.	HDQ_TX_DATA[7:0]	0x-
Select write operation.	HDQ_CTRL_STATUS[1] DIR	0x0
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Reading HDQ_INT_STATUS clears interrupt conditions.	HDQ_INT_STATUS[2] TXCOMPLETE	0x1

24.2.5.3.2 Main Sequence - 1-Wire Read Operation Mode

Table 24-74. 1-Wire Read Operation Mode

Step	Register/Bit Field/Programming Model	Value
Select read operation.	HDQ_CTRL_STATUS[1] DIR	0x1
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Read and store HDQ_INT_STATUS. Reading HDQ_INT_STATUS clears interrupt conditions.	HDQ_INT_STATUS	0x-
IF: Read operation successful?	HDQ_INT_STATUS[1] RXCOMPLETE	= 0x1
Get received data.	HDQ_RX_DATA[7:0]	0x-
ENDIF		

24.2.5.3.3 Sub-sequence - Initialize 1-Wire Slave

Table 24-75. Initialize 1-Wire Slave

Step	Register/Bit Field/Programming Model	Value
Select sending initialization pulse operation.	HDQ_CTRL_STATUS[2] INITIALIZATION	0x1
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
IF: Presence pulse detected?	HDQ_INT_STATUS[0] TIMEOUT	= 0x1
	HDQ_CTRL_STATUS[3] PRESENCEDETECT	= 0x1
Slave is present and initialized.		
ELSE		
Repeat initialization subsequence.		
ENDIF		

24.2.6 HDQ1W Register Manual

24.2.6.1 HDQ1W Instance Summary

Table 24-76. HDQ1W Instance Summary

Module Name	Base Address	Size
HDQ1W	0x480B 2000	4 KiB

24.2.6.2 HDQ1W Registers

CAUTION

The following rules must be observed when accessing the module registers:

- A read from the HDQ_INT_STATUS register or the HDQ_RX_DATA register is not allowed unless the processor has been interrupted by the module.
- After the release of the GO bit in the HDQ_CTRL_STATUS register, no access to the HDQ_TX_DATA or HDQ_CTRL_STATUS register is allowed until the processor has been interrupted by the module.
- Polling of the HDQ_INT_STATUS register by software to determine whether an interrupt was generated is not allowed.

CAUTION

The HDQ1W registers are limited to 32-bit data accesses; 16-bit and 8-bit data accesses are not allowed and can corrupt register content.

24.2.6.2.1 HDQ1W Register Summary

Table 24-77. HDQ1W Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
HDQ_REVISION	R	32	0x0000 0000	0x480B 2000
HDQ_TX_DATA	RW	32	0x0000 0004	0x480B 2004
HDQ_RX_DATA	R	32	0x0000 0008	0x480B 2008
HDQ_CTRL_STATUS	RW	32	0x0000 000C	0x480B 200C
HDQ_INT_STATUS	R	32	0x0000 0010	0x480B 2010
HDQ_SYSCONFIG	RW	32	0x0000 0014	0x480B 2014
HDQ_SYSSTATUS	R	32	0x0000 0018	0x480B 2018

24.2.6.2.2 HDQ1W Register Description

Table 24-78. HDQ_REVISION

Address Offset	0x0000 0000		
Physical Address	0x480B 2000	**Instance**	HDQ1W
Description	This register contains the IP revision code		
Type	R		

31:8	7:0
RESERVED	REV

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 24-79. HDQ_TX_DATA

Address Offset	0x0000 0004		
Physical Address	0x480B 2004	**Instance**	HDQ1W
Description	This register contains the data to be transmitted.		
Type	RW		

31:8	7:0
RESERVED	TX_DATA

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads returns 0	R	0x000000
7:0	TX_DATA	Transmit data (used in both HDQ and 1-Wire modes)	RW	0x00

Table 24-80. HDQ_RX_DATA

Address Offset	0x0000 0008		
Physical Address	0x480B 2008	**Instance**	HDQ1W
Description	This register contains the data to be received.		
Type	R		

31:8	7:0
RESERVED	RX_DATA

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads returns 0	R	0x000000
7:0	RX_DATA	Receive data (used in both HDQ and 1-Wire modes)	R	0x00

Table 24-81. HDQ_CTRL_STATUS

Address Offset	0x0000 000C		
Physical Address	0x480B 200C	**Instance**	HDQ1W
Description	This register provides status information about the module.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	BITFSM	ONE_WIRE_SINGLE_BIT	INTERRUPTMASK	CLOCKENABLE	GO	PRESENCEDETECT	INITIALIZATION	DIR	MODE

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reads returns 0	RW	0x000000
10:8	BITFSM	BITFSM delay value in 1.33 µs steps. 0x0 value corresponds to 1.33 µs.	RW	0x00
7	ONE_WIRE_SINGLE_BIT	Single-bit mode for 1-Wire 0x0: Disabled 0x1: Enabled	RW	0
6	INTERRUPTMASK	Interrupt masking bit 0x0: Interrupts disable 0x1: Interrupts enable	RW	0
5	CLOCKENABLE	Power-down mode bit 0x0: Clock disable (power down) 0x1: Clock enable	RW	0
4	GO	Go bit. Write 1 to start the appropriate operation. Bit returns to 0 after the operation is complete.	RW	0
3	PRESENCEDETECT	Slave presence indicator. Actual only just after initialization time-out. Used in 1-Wire mode. Read-only flag. 0x0: No slave detected 0x1: Slave detected	R	0
2	INITIALIZATION	Write 1 to send initialization pulse. Bit returns to 0 after pulse is sent.	RW	0
1	DIR	DIR bit, determines if next command is read or write 0x0: Write 0x1: Read	RW	0
0	MODE	Mode selection bit 0x0: HDQ mode 0x1: 1-Wire mode	RW	0

Table 24-82. HDQ_INT_STATUS

Address Offset	0x0000 0010		
Physical Address	0x480B 2010	**Instance**	HDQ1W
Description	This register controls interrupts status		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TXCOMPLETE	RXCOMPLETE	TIMEOUT

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads returns 0	R	0x0000 0000

Bits	Field Name	Description	Type	Reset
2	TXCOMPLETE	TX-complete interrupt flag. Set to 1 if cause of interrupt. Set to 0 when register read.	R	0
1	RXCOMPLETE	Read-complete interrupt flag. Set to 1 if cause of interrupt. Set to 0 when register read.	R	0
0	TIMEOUT	Presence detect/timeout interrupt flag. In 1-Wire mode, set to 1 if slave's presence detected. In HDQ mode, set to 1 if timeout on read occurs. Set to 0 when register read.	R	0

Table 24-83. HDQ_SYSCONFIG

Address Offset	0x0000 0014		
Physical Address	0x480B 2014	**Instance**	HDQ1W
Description	This register controls various bits		
Type	RW		

31–2	1	0
RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0	RW	0x0000 0000
1	SOFTRESET	Start soft reset sequence. 0x0: Disabled 0x1: Enabled	RW	0
0	AUTOIDLE	Interconnect idle. 0x0: Module clock is free-running. 0x1: Module is in power saving mode: Clock is running only when module is accessed or inside logic is in function to process events.	RW	0

Table 24-84. HDQ_SYSSTATUS

Address Offset	0x0000 0018		
Physical Address	0x480B 2018	**Instance**	HDQ1W
Description	This register monitors the reset sequence.		
Type	R		

31–1	0
RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x0000 0000
0	RESETDONE	Reset monitoring. 0x0: The module is currently performing its reset. When the module is in power-down mode, set to 0 to indicate this fact. 0x1: The module has finished its reset.	R	1

24.3 UART/IrDA/CIR

This chapter describes the function, operation, and configuration of the universal asynchronous receiver/transmitter (UART)/infrared data association (IrDA)/consumer infrared (CIR) module in the device.

24.3.1 UART/IrDA/CIR Overview

The UART is a simple L4 slave peripheral that utilizes the DMA_SYSTEM or EDMA for data transfer or IRQ polling via CPU. There are 10 UART modules in the device. Only one UART supports IrDA features. Each UART can be used for configuration and data exchange with a number of external peripheral devices or interprocessor communication between devices.

Figure 24-37 shows the UART module overview.

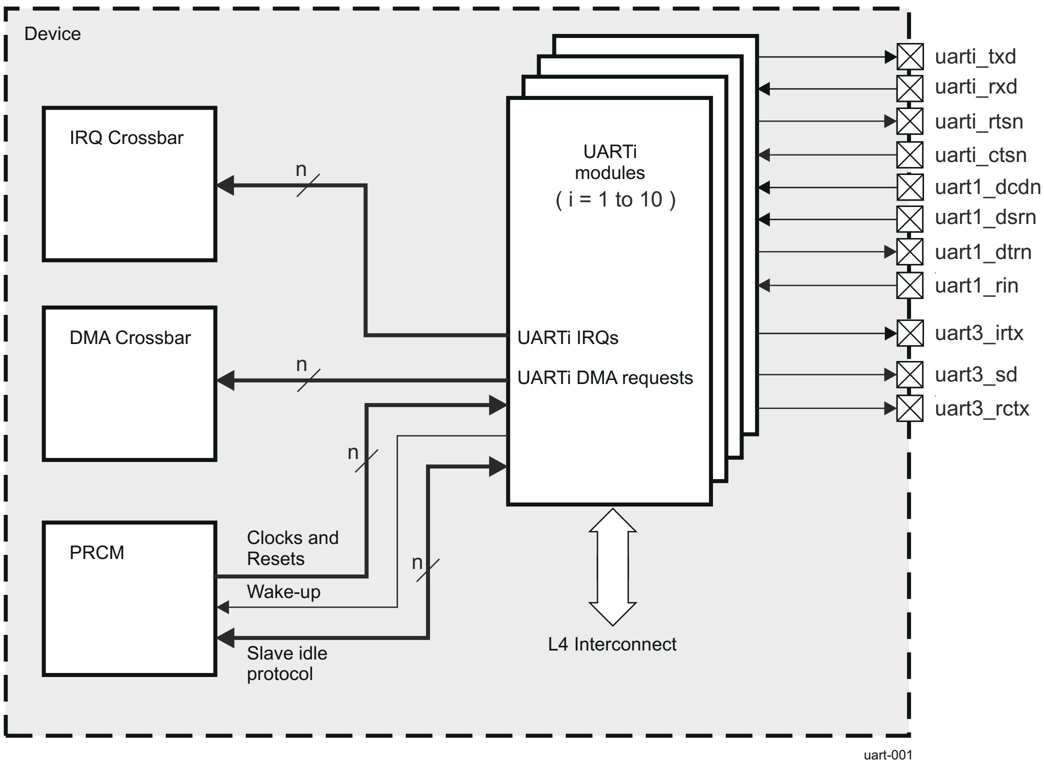


Figure 24-37. UART Overview

24.3.1.1 UART Features

The UARTi (where i = 1 to 10) include the following features:

- 16C750 compatibility
- 64-byte FIFO buffer for receiver and 64-byte FIFO for transmitter
- Programmable interrupt trigger levels for FIFOs
- Baud generation based on programmable divisors N (where N = 1…16,384) operating from a fixed functional clock of 48 MHz or 192 MHz

Oversampling is programmed by software as 16 or 13. Thus, the baud rate computation is one of two options:

- Baud rate = (functional clock / 16) / N
- Baud rate = (functional clock / 13) / N
- This software programming mode enables higher baud rates with the same error amount without changing the clock source
- Break character detection and generation
- Configurable data format:

  – Data bit: 5, 6, 7, or 8 bits
  – Parity bit: Even, odd, none
  – Stop-bit: 1, 1.5, 2 bit(s)
- Flow control: Hardware (RTS/CTS) or software (XON/XOFF)
- The 48 MHz functional clock option allows baud rates up to 3.6Mbps
- The 192 MHz functional clock option allows baud rates up to 12Mbps
- UART1 module has extended modem control signals (DCD, RI, DTR, DSR)
- UART3 supports IrDA

24.3.1.2 IrDA Features

UART3 supports the following IrDA key features:

- Support of IrDA 1.4 slow infrared (SIR), medium infrared (MIR), and fast infrared (FIR) communications:
  – Frame formatting: Addition of variable beginning-of-frame (xBOF) characters and end-of-frame (EOF) characters
  – Uplink/downlink cyclic redundancy check (CRC) generation/detection
  – Asynchronous transparency (automatic insertion of break character)
  – Eight-entry status FIFO (with selectable trigger levels) to monitor frame length and frame errors
  – Framing error, CRC error, illegal symbol (FIR), and abort pattern (SIR, MIR) detection

24.3.1.3 CIR Features

The CIR mode uses a variable pulse-width modulation (PWM) technique (based on multiples of a programmable *t* period) to encompass the various formats of infrared encoding for remote-control applications. The CIR logic transmits data packets based on a user-definable frame structure and packet content.

The CIR (UART3 only) includes the following features to provide CIR support for remote-control applications:

- Transmit mode only (receive mode is not supported)
- Free data format (supports any remote-control private standards)
- Selectable bit rate
- Configurable carrier frequency
- 1/2, 5/12, 1/3, or 1/4 carrier duty cycle

24.3.2 UART/IrDA/CIR Environment

This section describes the UART/IrDA/CIR connection with an external device.

- The UART interface is described in Section 24.3.2.1.
- The IrDA interface is described in Section 24.3.2.2, *IrDA Functional Interfaces*.
- The CIR interface is described in Section 24.3.2.3, *CIR Functional Interfaces*.

24.3.2.1 UART Interface

24.3.2.1.1 System Using UART Communication With Hardware Handshake

Each UART instance can be easily connected to the UART port of an external IC (see Figure 24-38).

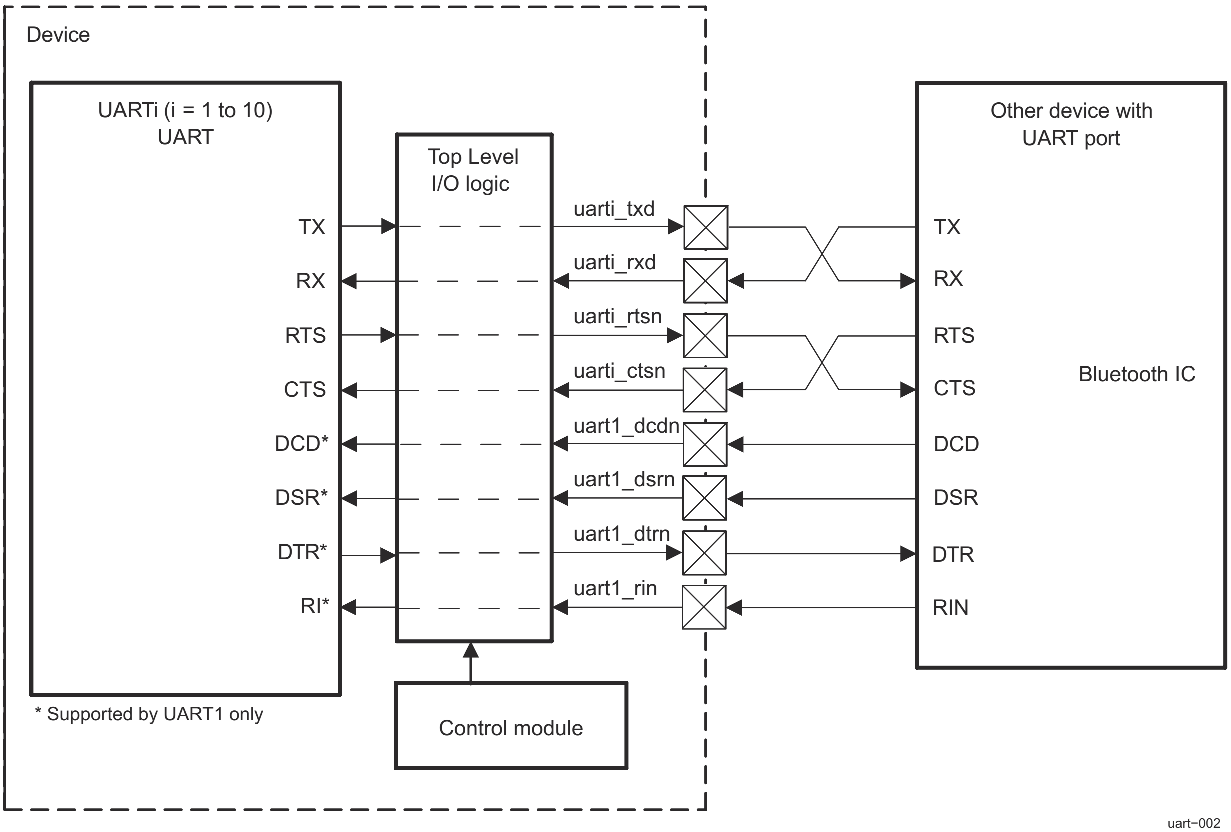


Figure 24-38. UART Mode Bus System Overview

24.3.2.1.2 UART Interface Description

Table 24-85 lists the UART interface input/output (I/O) signals.

Table 24-85. UART Interface Signals

Signal	I/O(1)	Description	Module Level Reset Value
UARTi Interface Signals(2)			
uarti_rxd	I	Serial data input.	HiZ
uarti_txd	O	Serial data output Because this pin is active high in IrDA mode and the output is muxed, this pin is set to low on reset (when the UARTi.UART_MDR1[2:0] bit field is set to 0x7) and takes the defined inactive level of that signal corresponding to when and how the UARTi.UART_MDR1 register is programmed; that is, the output is 1 (inactive for UART modem modes) and 0 (inactive for IrDA modes).	1
uarti_ctsn	I	Clear to send	HiZ

Table 24-85. UART Interface Signals (continued)

Signal	I/O[(1)]	Description	Module Level Reset Value
		Active-low modem status signal. Reading the UARTi.UART_MSR[4] NCTS_STS bit checks the condition of uarti_ctsn. Reading the UARTi.UART_MSR[0] CTS_STS bit checks a change of state of uarti_ctsn since the last read of the modem status register. The auto-CTS mode uses uarti_ctsn to control the transmitter.	
uarti_rtsn	O	Request to send When active (low), the module is ready to receive data. Setting the UARTi.UART_MCR[1] RTS bit activates uarti_rtsn, which becomes inactive as the result of a module reset, loopback mode, or clearing the UARTi.UART_MCR[1] RTS bit. In auto-RTS mode, uarti_rtsn becomes inactive as a result of the receiver threshold logic.	1
UART1 Modem Signals			
uart1_dcdn	I	Data Carrier Detect Active-low modem status signal. The condition of uart1_dcdn can be checked by reading UART_MSR[7] NCD_STS register bit. Any change in its state can be detected by reading UART_MSR[3] DCD_STS bit.	HiZ
uart1_dsrn	I	Data Set Ready Active-low modem status signal. Reading UART_MSR[5] NDSR_STS register bit checks the condition of uart1_dsrn. Reading UART_MSR[1] DSR_STS bit checks a change of state of uart1_dsrn since the last read of the UART_MSR register.	HiZ
uart1_dtrn	O	Data Terminal Ready When active (low), this signal informs the modem that the module is ready to communicate. It is activated by setting UART_MCR[0] DTR register bit.	1
uart1_rin	I	Ring Indicator Active-low modem status signal. The condition of uart1_rin can be checked by reading UART_MSR[6] NRI_STS register bit. Any change in its state can be detected by reading UART_MSR[2] RI_STS bit.	HiZ

(1) I = Input; O = Output; I/O = Bidirectional
(2) i = 1 to 10

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. For more information on control module settings, see *Pad Configuration Registers*, in *Control Module*.

24.3.2.1.3 UART Protocol and Data Format

The UART device operates in three modes:

- UART 16x (<= 230.4 kbps)
- UART 16x with autobauding (>= 1200 bps and >= 115.2 kbps)
- UART 13x (>= 460.8 kbps)

CAUTION

To be used as a UART, the operating mode must be programmed appropriately in the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to select UART, IrDA, or CIR mode.

The UART uses a wired interface for serial communication with a remote device.

The UART is functionally compatible with the TL16C750 UART and earlier designs such as the TL16C550.

Figure 24-39 shows the UART frame data format.

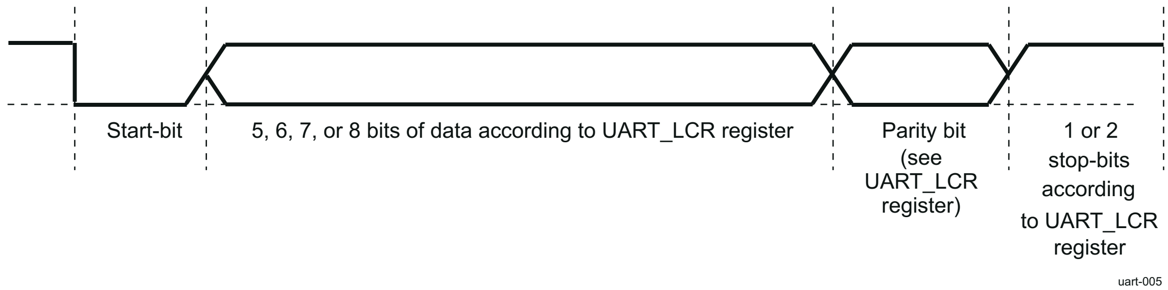


Figure 24-39. UART Frame Data Format

24.3.2.2 IrDA Functional Interfaces

24.3.2.2.1 System Using IrDA Communication Protocol

As Figure 24-40 shows, UART3 can be connected to an external infrared transceiver in the IrDA modes (FIR, SIR, and MIR).

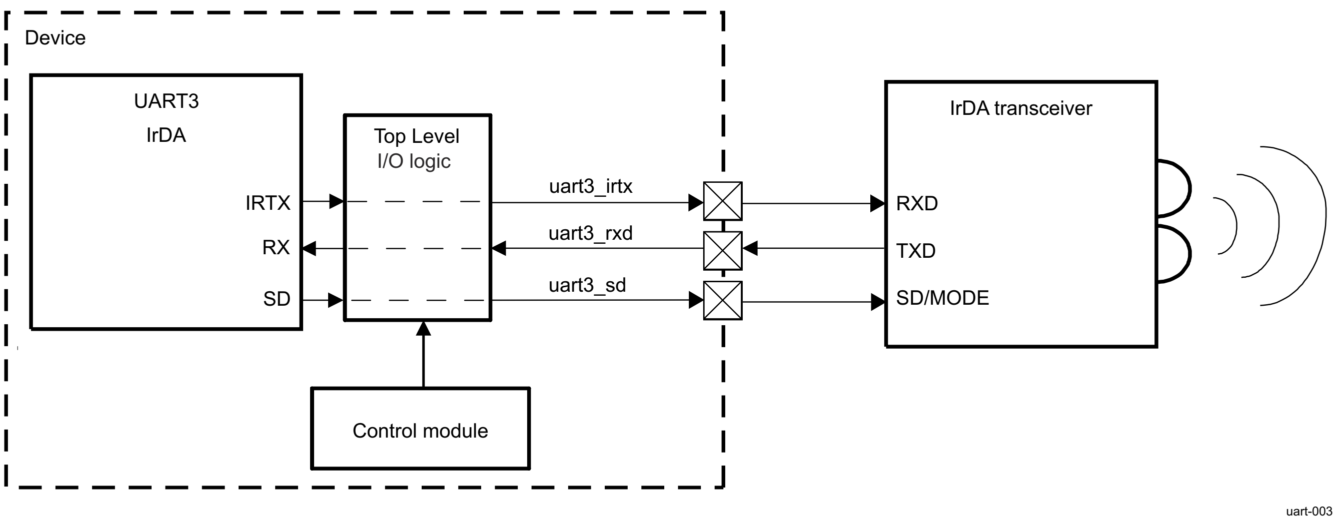


Figure 24-40. IrDA System Overview

24.3.2.2.2 IrDA Interface Description

Table 24-86 lists the IrDA interface I/O signals.

Table 24-86. IrDA I/O Signals

Signal	I/O(1)	Description	Reset
IrDA Signals			
uart3_rxd	I	Serial data input	HiZ
uart3_irtx	O	Serial data output in IrDA modes (SIR, MIR, and FIR). In other modes, this pin is set to the reset value (inactive state).	0
uart3_sd	O	SD mode is used to configure the transceivers. The SD pinout is an inverted value of the UART3.UART_ACREG[6] SD_MOD bit.	1

(1) I = Input; O = Output

24.3.2.2.3 IrDA Protocol and Data Format

24.3.2.2.3.1 SIR Mode

In SIR mode, data is transferred between the MPUand peripheral devices at speeds of up to 115,200 baud. A SIR transmit frame begins with start flags (a single 0xC0, a multiple 0xC0, or a single 0xC0 preceded by a number of 0xFF flags), followed by frame data and a CRC-16, and ends with a stop flag (0xC1).

The bit format for a single word uses 1 start-bit, 8 data bits, and 1 stop-bit, and is unaffected by the use and settings of the UART3.UART_LCR register.

The UART3.UART_BLR[6] XBOF_TYPE bit selects whether the 0xC0 or 0xFF start patterns are used when multiple start flags are required.

The SIR transmit state-machine attaches start flags, CRC-16, and stop flags, and checks the outgoing data to establish whether data transparency is required.

SIR transparency is carried out if the outgoing data between the start and stop flags contains 0xC0, 0xC1, or 0x7D. If one of these start flags is about to be transmitted, the SIR state-machine sends an escape character (0x7D), inverts the fifth bit of the real data to be sent, and then sends this data immediately after the 0x7D character.

The SIR receive state-machine recovers the receive clock, removes the start flags and any transparency from the incoming data, and determines the frame boundary with reception of the stop flag. The SIR state-machine also checks for errors such as a frame abort (0x7D character followed immediately by a 0xC1 stop flag without transparency), a CRC error, or a frame-length error. At the end of a frame reception, the MPUreads the line status register (UART3.UART_LSR_IRDA) to find possible errors of the received frame.

Note

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. See the description of the UART3.UART_ACREG[5] DIS_IR_RX bit. This applies to all three modes: SIR, MIR, and FIR.

Infrared output in SIR mode can be 1.6-μs or 3/16 encoding, selected by the UART3.UART_ACREG[7] PULSE_TYPE bit. In 1.6-μs encoding, the infrared pulse width is 1.6 μs; and in 3/16th encoding, the infrared pulse width is 3/16th of a bit duration (1/baud rate).

For back-to-back frames, the transmitting device must send at least two start flags at the start of each frame.

Note

Reception supports variable-length stop-bits.

24.3.2.2.3.1.1 Frame Format

Figure 24-41 shows the IrDA SIR frame format.

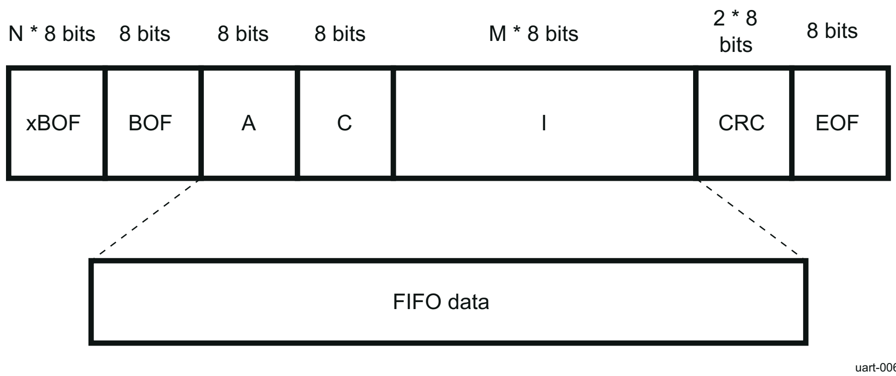


Figure 24-41. IrDA SIR Frame Format

The CRC is applied on the address (A), control (C), and information (I) bytes.

Note

The two words of CRC are written to the FIFO in reception.

24.3.2.2.3.1.2 Asynchronous Transparency

Before transmitting a byte, the UART IrDA controller examines each byte of the payload and the CRC field (between BOF and EOF). For each byte equal to 0xC0 (BOF), 0xC1 (EOF), or 0x7D (control escape), the controller performs certain tasks:

- In transmission:
  - Inserts a control escape (CE) byte preceding the byte
  - Complements bit 5 of the byte (that is, exclusive ORs the byte with 0x20)

  The byte sent for the CRC computation is the initial byte written in the TX FIFO (before the XOR with 0x20).
- In reception:

  For the A, C, I, and CRC fields:
  - Compares the byte with the CE byte; if they are not equal, sends the byte to the CRC detector and stores it in the RX FIFO.
  - If the byte is equal to the CE byte, discards the CE byte
  - Complements bit 5 of the byte following the CE
  - Sends the complemented byte to the CRC detector and stores it in the RX FIFO

24.3.2.2.3.1.3 Abort Sequence

The transmitter can prematurely close a frame (abort) by sending the sequence 0x7DC1. The abort pattern closes the frame without a CRC field or an ending flag.

When a 0x7D character that is followed immediately by a 0xC1 character is received without transparency, the receiver treats the frame as an aborted frame.

24.3.2.2.3.1.4 Pulse Shaping

The SIR mode supports the 3/16 and the 1.6-μs pulse duration methods. The UART3.UART_ACREG[7] PULSE_TYPE bit selects the pulse-width method in transmit mode.

24.3.2.2.3.1.5 Encoder

Serial data from the transmit state-machine are encoded to transmit data to the optoelectronics. While the TX FIFO output is high, the uart3_irtx line is always low, and the counter used to form a pulse on uart3_irtx is cleared continuously.

After the TX FIFO output resets to 0, uart3_irtx rises on the falling edge of the seventh 16XCLK. On the falling edge of the tenth 16XCLK pulse, uart3_irtx falls, creating a 3-clock-wide pulse. While the TX FIFO output stays low, a pulse is transmitted during the seventh clock to the tenth clock of each 16-clock bit cycle.

Figure 24-42 shows the IrDA SIR encoding mechanism.

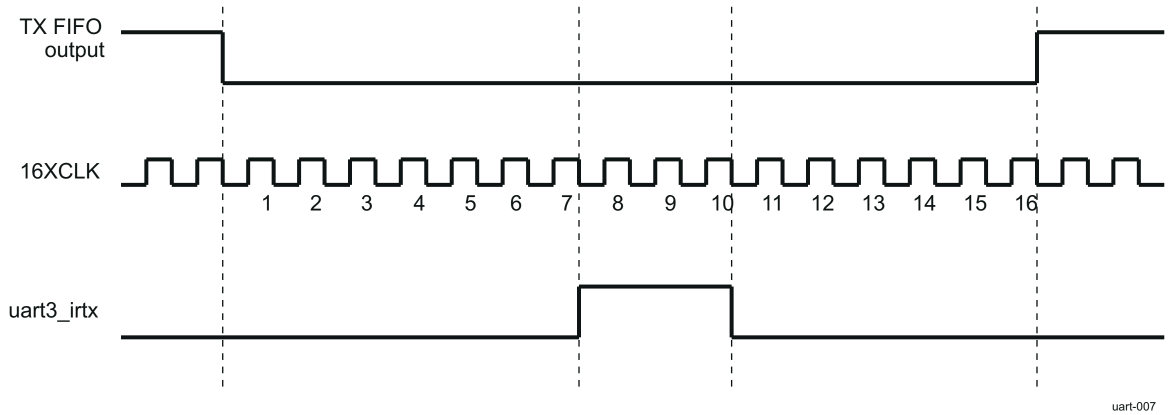


Figure 24-42. IrDA SIR Encoding Mechanism

24.3.2.2.3.1.6 Decoder

After reset, the RX FIFO input is high and the 4-bit counter is cleared. When a rising edge is detected on RX, the RX FIFO input falls on the next rising edge of 16XCLK with sufficient setup time. The RX FIFO input stays low for 16 cycles (16XCLK) and then returns to high as required by the IrDA specification. As long as no pulses (rising edges) are detected on the RX, the RX FIFO input remains high.

Figure 24-43 shows the IrDA SIR decoding mechanism.

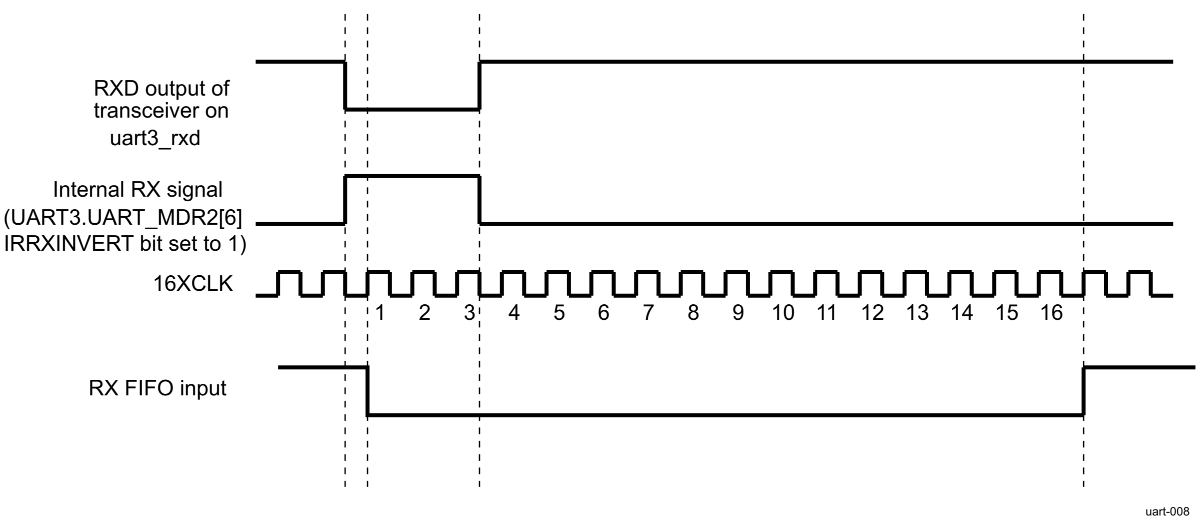


Figure 24-43. IrDA SIR Decoding Mechanism

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. The operation of the uart3_rxd input can be disabled using the UART3.UART_ACREG[5] DIS_IR_RX bit. The UART3.UART_MDR2[6] IRRXINVERT bit can invert the signal from the transceiver (RXD) pin to the IR RX logic in the UART. This inversion is performed by default.

24.3.2.2.3.1.7 IR Address Checking

In all IR modes, when address checking is enabled by setting the UART3.UART_EFR[1:0] bit field (see Table 24-87), only frames intended for the device are written to the RX FIFO. This is to avoid receiving frames not meant for this device in a multipoint infrared environment. To program two frame addresses that the UART3 receives in IrDA mode, use the UART3.UART_XON1_ADDR1[7:0] and UART3.UART_XON2_ADDR2[7:0] bit fields.

Table 24-87. UART_EFR[1:0] IR Address Checking Options

UART_EFR[1]	UART_EFR[0]	IR Address Checking
0	0	All address-checking operations disabled
0	1	Only address 1 checking enabled
1	0	Only address 2 checking enabled
1	1	All address-checking operations enabled

24.3.2.2.3.2 SIR Free-Format Mode

To allow complete software flexibility when transmitting and receiving infrared data packets, the SIR free-format (FF) mode is a subfunction of the existing SIR mode. In FF mode, all frames going to and from the FIFO buffers are untouched with respect to appending and removing control characters and CRC values.

The FF mode corresponds to a UART mode with a pulse modulation of 3/16 of baud rate pulse width.

For example, a normal SIR packet has BOF control and CRC error-checking data appended (transmitting) or removed (receiving) from the data going to and from the FIFOs.

Figure 24-44 shows SIR FF mode.

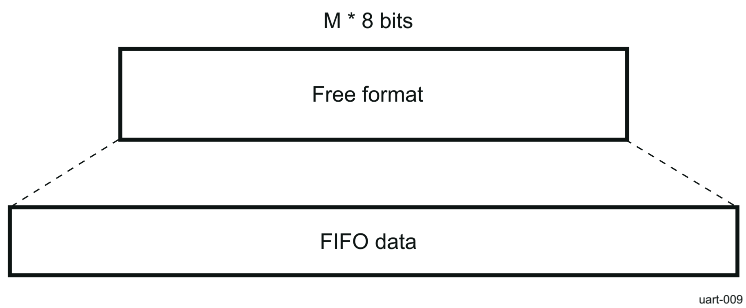


Figure 24-44. SIR FF Mode

In SIR FF mode, the MPU software must construct (that is, encode and decode) the entire FIFO data packet.

24.3.2.2.3.3 MIR Mode

In MIR mode, data is transferred between the MPUand the peripheral devices at 0.576 or 1.152 Mbps. A MIR transmit frame starts with at least two start flags, followed by a frame data and a CRC-16, and ends with a stop flag (see Figure 24-45).

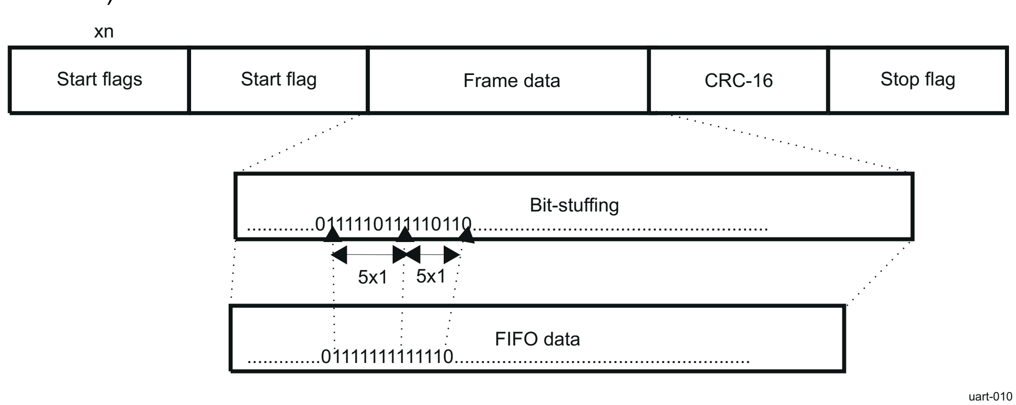


Figure 24-45. MIR Transmit Frame Format

On transmit, the MIR state-machine attaches start flags, a CRC-16, and stop flags, as in SIR mode. All fields are transmitted least-significant bit (LSB) of each byte first.

In MIR mode:

- The state-machine looks for consecutive 1s in the frame data and automatically inserts 0 after five consecutive 1s (this is called bit-stuffing).
- 0x7E is used for start and stop flags (unambiguously, not data, because of bit-stuffing).
- An abort sequence requires a minimum of seven consecutive 1s (unambiguously, not data, because of bit-stuffing).
- Back-to-back frames are allowed with three or more stop flags between them. If two consecutive frames are not back to back, the gap between the last stop flag of the first frame and the start flag of the second frame must be separated by at least seven bit durations.

On receive, the MIR receive state-machine recovers the receive clock, removes the start flags, destuffs the incoming data, and determines the frame boundary with reception of the stop flag. The state-machine also checks for errors such as frame abort, CRC error, and frame-length error. At the end of a frame reception, the MPU reads the line status register (UART3.UART_LSR_IRDA) to detect errors of the received frame.

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware.

24.3.2.2.3.3.1 MIR Encoder/Decoder

To meet the MIR baud rate tolerance of 0.1 percent with a 48-MHz clock input, a 42-41-42 encoding/decoding adjustment is performed. The reference start point is the first start flag, and the 42-41-42 cyclic pattern is repeated until the stop flag is sent or detected.

The jitter created this way is within MIR tolerances. The pulse width is not exactly 1/4, but it is within the tolerances defined by IrDA specifications.

Figure 24-46 shows the MIR baud rate adjustment mechanism.

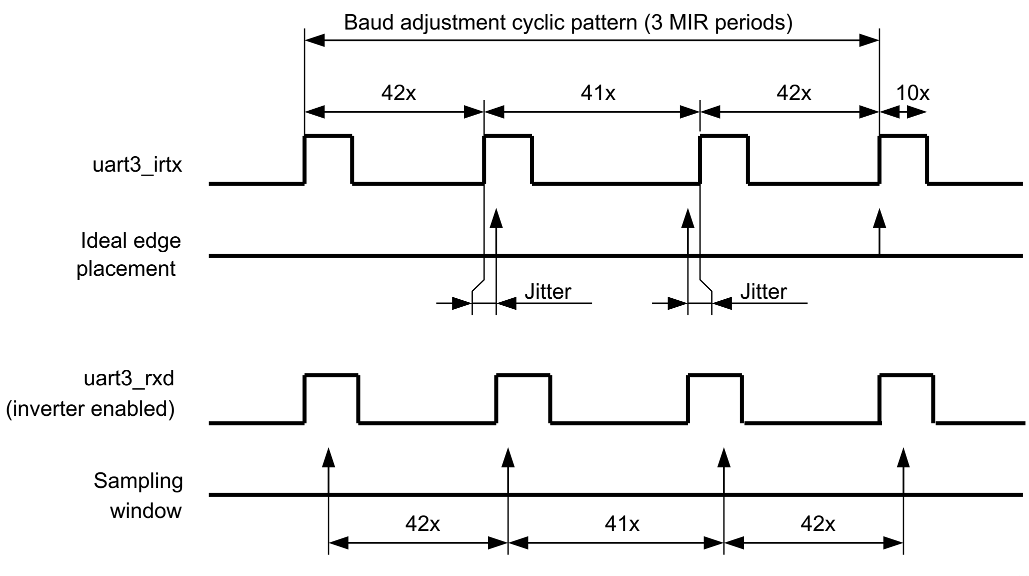


Figure 24-46. MIR Baud Rate Adjustment Mechanism

24.3.2.2.3.3.2 SIP Generation

In the MIR and FIR operation modes, the transmitter must send a serial infrared interaction pulse (SIP) at least once every 500 ms. The SIP informs slow devices (operating in SIR mode) that the medium is occupied.

Figure 24-47 shows the SIP.

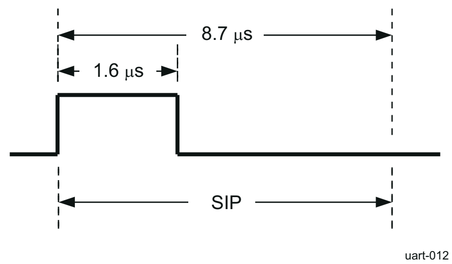


Figure 24-47. SIP

24.3.2.2.3.4 FIR Mode

In FIR mode, data is transferred between the MPU and the peripheral devices at 4 Mbps. A FIR transmit frame starts with a preamble that is followed by a start flag, frame data, CRC-32, and ends with a stop flag.

Figure 24-48 shows the FIR transmit frame format.

Figure 24-48. FIR Transmit Frame Format

Preamble (16x)	Start flag	Frame data	CRC-32	Stop flag

On transmit, the FIR transmit state-machine attaches the preamble, start flag, CRC-32, and stop flag. An abort sequence requires at least two transmissions of 0000. Back-to-back frames are allowed, but each frame must be complete.

The state-machine also encodes the transmit data into 4-PPM format (see Table 24-88) and generates the SIP (see Section 24.3.2.2.3.3.2, *SIP Generation*).

Table 24-88. 4-PPM Format

Data Bit Pair (Bin)	4-PPM Data Symbol (Bin)
00	1000
01	0100
10	0010
11	0001

The four symbols described in Table 24-88 are the legal, encoded data symbols. All other combinations are illegal for encoding data. Some of these illegal symbols are used in the definition of the preamble, start flag, and stop flag because they are unambiguously not data (see Table 24-89).

Table 24-89. FIR Preamble, Start Flag, and Stop Flag

Frame Part	Transmitted Frame (Bin)
Preamble	1000 0000 1010 1000 (16 repeated transmissions)
Start flag	0000 1100 0000 1100 0110 0000 0110 0000
Stop flag	0000 1100 0000 1100 0000 0110 0000 0110

All fields are transmitted LSBs of each byte first (see Table 24-90).

Table 24-90. FIR Data Byte Transmission Order Example

Data Byte (Hex)	Data Byte Pair (Bin)	4-PPM Data Symbol (Bin)	Transmission Order
0x0B	00	1000	4
	00	1000	3
	10	0010	2
	11	0001	1

On receive, the FIR receive state-machine recovers the receive clock, removes the preamble and the start flag, decodes the 4-PPM incoming data, and determines the frame boundary with reception of the stop flag. The state-machine also checks for errors such as illegal symbol, CRC error, and frame-length error. At the end of a frame reception, the MPU reads the line status register (UART3.UART_LSR_IRDA) to detect errors of the received frame.

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware.

24.3.2.3 CIR Functional Interfaces

24.3.2.3.1 System Using CIR Communication Protocol With Remote Control

UART3 can be connected to an external infrared transceiver in CIR mode (see Figure 24-49).

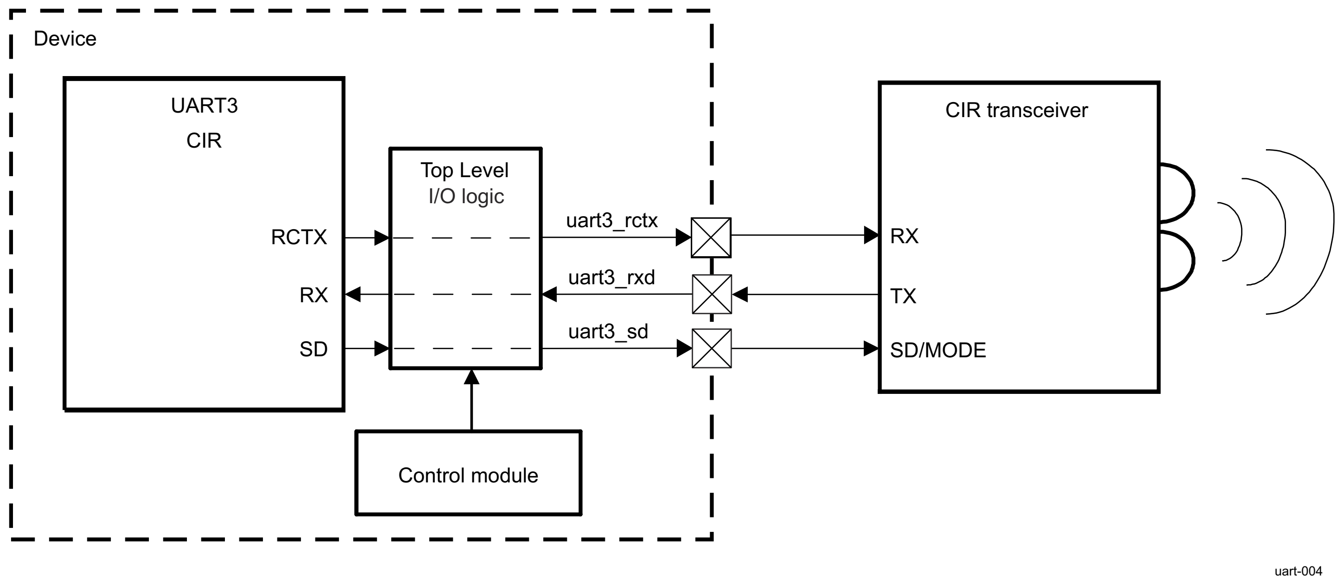


Figure 24-49. CIR System Overview

24.3.2.3.2 CIR Interface Description

Table 24-91 lists the CIR interface I/O signals.

Table 24-91. CIR I/O Signals

Signal	I/O(1)	Description	Reset
CIR Signals			
uart3_rxd	I	Serial data input	HiZ
uart3_rctx	O	Serial data output in CIR mode. In other modes, this pin is set to the reset value (inactive state).	0
uart3_sd	O	SD mode is used to configure the transceivers. The SD pinout is an inverted value of the UART3.UART_ACREG[6] SD_MOD bit.	1

(1) I = Input; O = Output

24.3.2.3.3 CIR Protocol and Data Format

In CIR mode, the infrared operation functions as a programmable (universal) remote control.

CIR mode uses a variable PWM technique (based on multiples of a programmable *t* period) to encompass the various formats of infrared encoding for remote-control applications. The CIR logic transmits data packets based on user-defined frame structure and packet content.

24.3.2.3.3.1 Carrier Modulation

Each modulated pulse that constitutes a digit is a train of on/off pulses (see Figure 24-50).

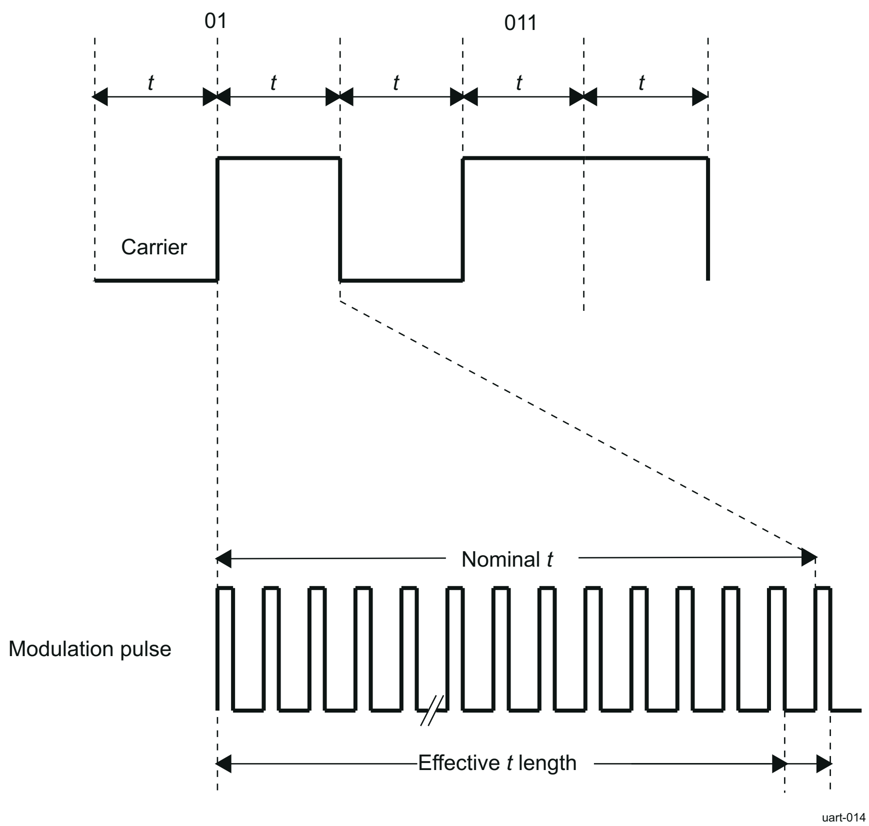


Figure 24-50. CIR Pulse Modulation

24.3.2.3.3.2 Pulse Duty Cycle

The programmer can choose one of four duty cycles for modulation pulses by setting the appropriate value in the UART3.UART_MDR2[5:4] CIR_PULSE_MODE bit field (1/4, 1/3, 5/12, or 1/2).

Figure 24-51 shows the CIR modulation duty cycles.

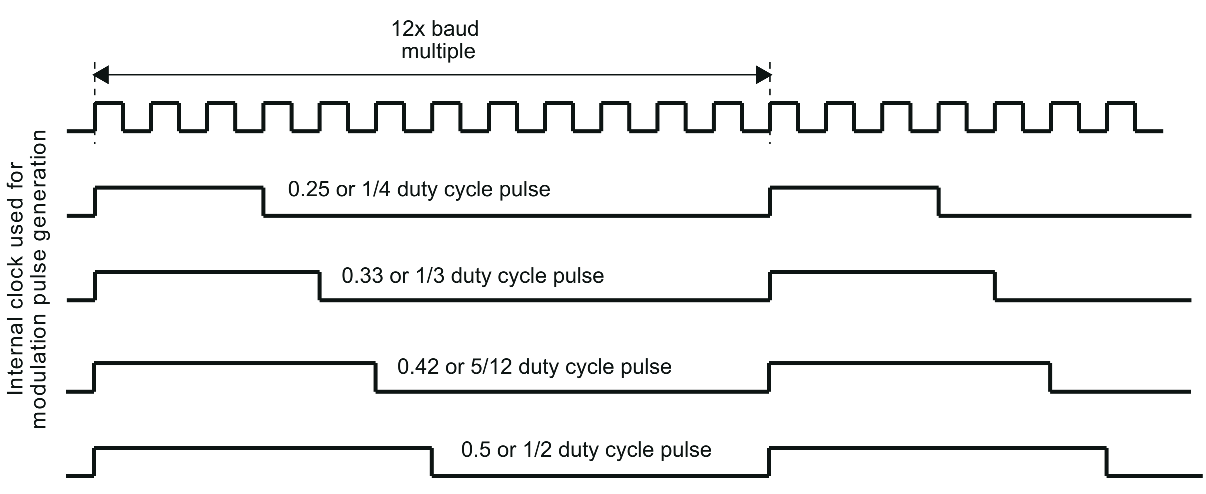


Figure 24-51. CIR Modulation Duty Cycle

The transmission logic ensures that all pulses are transmitted completely (no cutoff during transmission). While transmitting continuous bytes back-to-back, no delay is inserted between 2 transmitted bytes. Thus, software must handle the delay between consecutively transmitted bytes if the receiving end requires it.

24.3.2.3.3.3 Consumer IR Encoding/Decoding

There are two methods of encoding for remote-control applications:

- Pulse duration encoding (time-extended bit forms): A variable pulse distance, or duration, in which the difference between logic 1 and logic 0 is the length of the pulse width
- Biphase encoding: The encoding of logic 0 and logic 1 is in the change of signal level from 1 to 0 or 0 to 1, respectively.

Japanese manufacturers favor pulse duration encoding; European manufacturers favor biphase encoding.

CIR mode uses a completely flexible free-format encoding in which 1 is transmitted from the TX FIFO as a modulated pulse with duration *t*.

Similarly, 0 is transmitted as a blank duration T. The MPUconstructs and deciphers the protocol of the data. For example, the RC-5 protocol using Manchester encoding can be emulated as using a 01 pair for 1 and a 10 pair for 0 (see Figure 24-52).

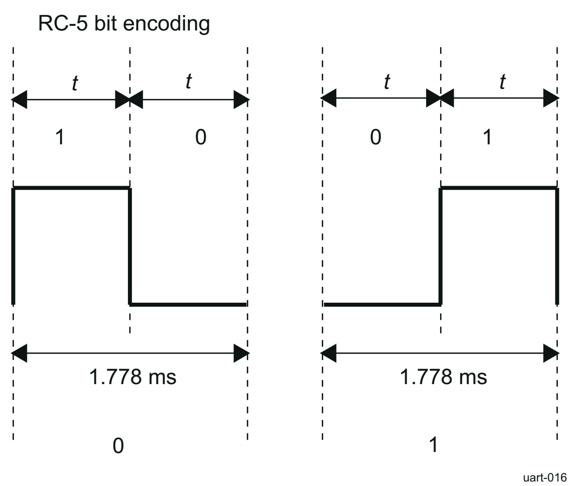


Figure 24-52. RC-5 Bit Encoding

Because CIR mode logic does not impose a fixed format for infrared packets of data, the MPUsoftware can define the format using simple data structures that are then modulated into an industry standard, such as RC-5 or SIRC. To send a sequence of 0101 in RC-5, the MPUsoftware must write an 8-bit binary character of 10011001 to the data FIFO of the UART.

For SIRC, the modulation length (multiples of *t*) is used to distinguish between 1 and 0. The subsequent SIRC digits show the difference in encoding between this and, for example, RC-5. The pulse width is extended for one digit.

Figure 24-53 shows SIRC bit encoding.

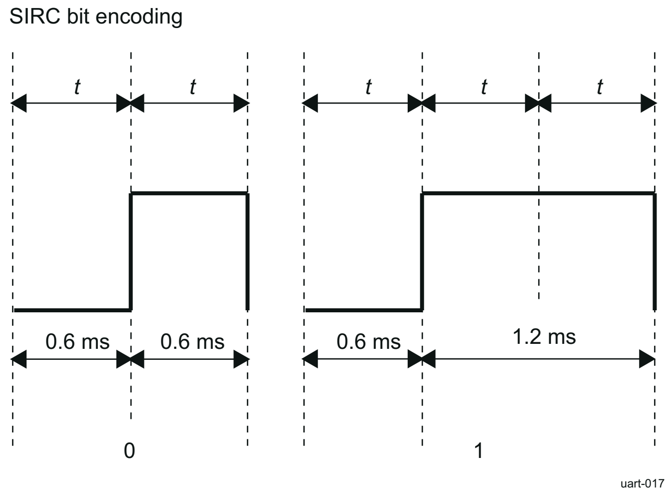


Figure 24-53. SIRC Bit Encoding

To construct comprehensive packets constituting remote-control commands, the MPU software must combine a number of 8-bit data characters in a sequence that follows one of the universally accepted formats.

Figure 24-54 shows a standard RC-5 frame as detected by UART3 in CIR mode (the SIRC format follows this). Each field in RC-5 can be considered as two *t* pulses (digital bits) from the TX FIFO.

S1	S2	T	A4	A3	A2	A1	A0	C5	C4	C3	C2	C1	C0

uart-018

Figure 24-54. RC-5 Standard Packet Format

Where:

S1, S2:	Start-bits (always 1)
T:	Toggle bit
A4..A0:	Address (or system) bits
C5..C0:	Command bits

The toggle bit T changes when a new command is transmitted to detect when the same key is pressed twice (effectively receiving the same data from the host consecutively). A brief delay in the transmission of the same command is detected by the use of the toggle bit because a code is sent while the MPUtransmits characters to the UART for transmission. The address bits define the machine or device for which the infrared transmission is intended, and the command defines the operation.

To accommodate an extended RC-5 format, the S2 bit is replaced by an additional command bit (C6) that lets the command range increase to 7 bits. This format is known as the extended RC-5 format.

SIRC encoding uses the duration of modulation for mark and space; therefore, the duration of data bits in the standard frame length varies.

Figure 24-55 shows the packet format and bit encoding. As Figure 24-56 shows, 1 start-bit of 2.4 ms and control codes are followed by data that constitute the entire frame.

S	C0	C1	C2	C3	C4	C5	C6	D0	D1	D2	D3	D4

uart-019

Figure 24-55. SIRC Packet Format

Note

The encoding must take a standard duration, but the contents of the data can vary. This implies that the control software for sending and receiving data packets must exercise a scheme of interpacket delay, where successive packets can be sent only after a real-time delay expires.

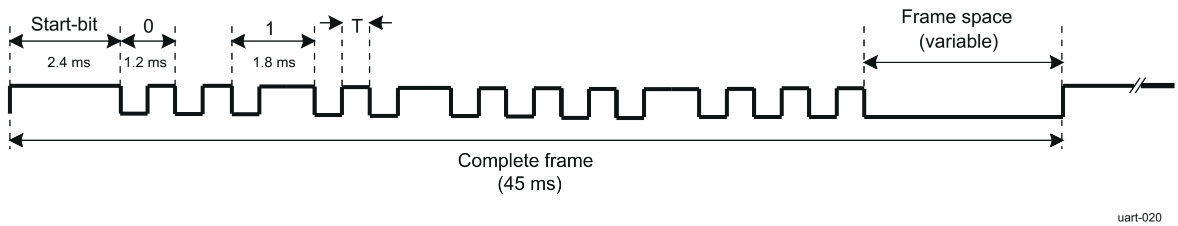


Figure 24-56. SIRC Bit Transmission Example

This document does not describe all encoding methods and techniques; the previous information discusses the considerations required to employ different encoding methods for different industry-standard protocols. See industry-standard documentation for specific methods of encoding and protocol use.

24.3.3 UART/IrDA/CIR Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 24-57 shows the device internal connections with related modules for UART functions.

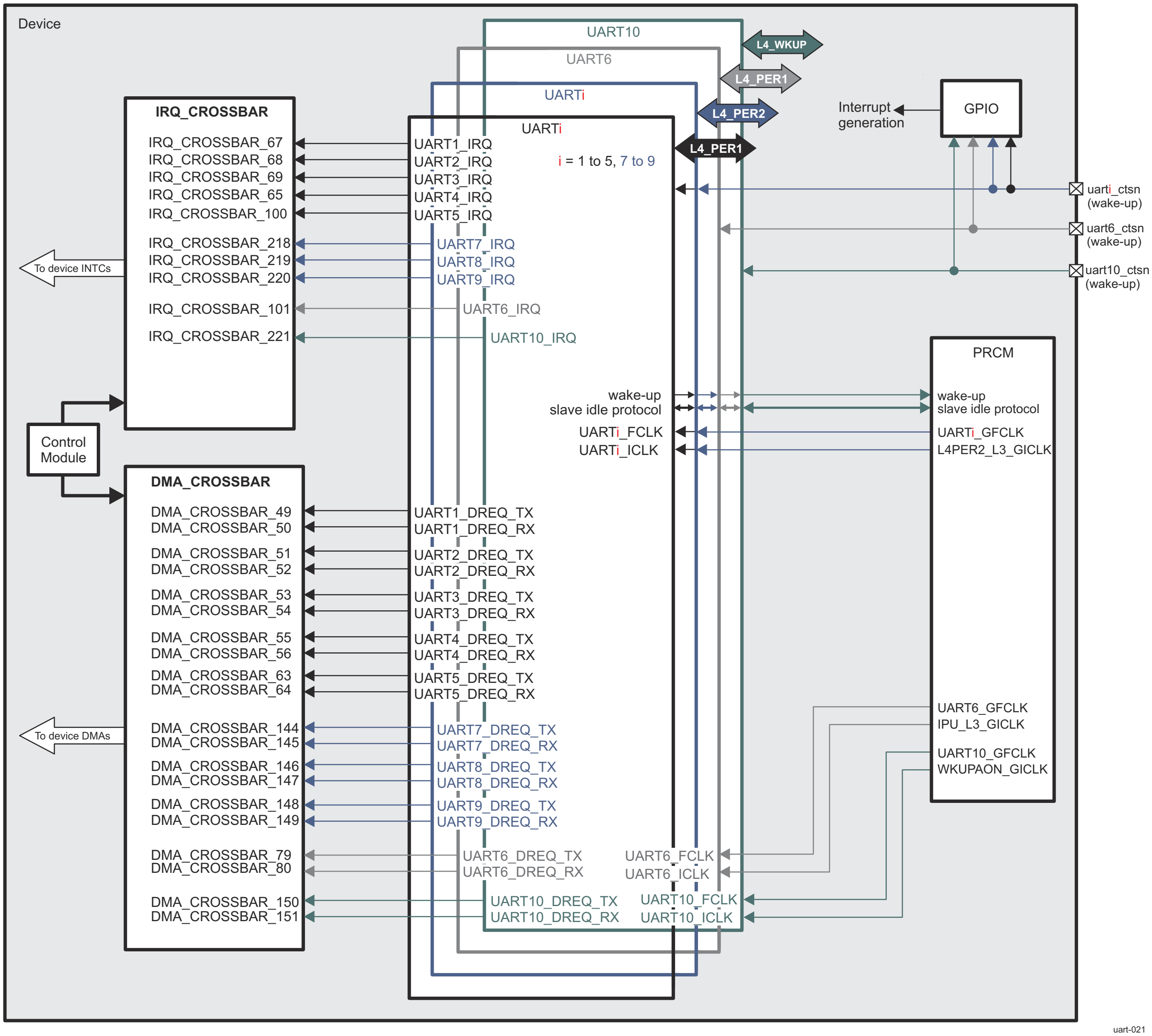


Figure 24-57. UART/IrDA/CIR Integration

Note

For more information about the master standby and slave idle protocols and the wake-up request, see *Clock Domain-Level Clock Management*, in *Power, Reset, and Clock Management*

24.3.3.1

Table 24-92 through Table 24-94 summarize the integration of the module in the device.

Table 24-92. UART Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
UART1	PD_COREAON	Yes	L4_PER1
UART2	PD_COREAON	Yes	L4_PER1
UART3	PD_COREAON	Yes	L4_PER1
UART4	PD_COREAON	Yes	L4_PER1
UART5	PD_COREAON	Yes	L4_PER1
UART6	PD_COREAON	Yes	L4_PER1
UART7	PD_COREAON	Yes	L4_PER2
UART8	PD_COREAON	Yes	L4_PER2
UART9	PD_COREAON	Yes	L4_PER2
UART10	PD_WKUPAON	Yes	L4_WKUP

Table 24-93. UART Clocks and Resets

Clocks				
Module Instance	Destination Signal	Source Signal	Source	Description
UART1	UART1_ICLK	L4PER_L3_GICLK	PRCM	UART1 interface clock
	UART1_FCLK	UART1_GFCLK	PRCM	UART1 functional clock
UART2	UART2_ICLK	L4PER_L3_GICLK	PRCM	UART2 interface clock
	UART2_FCLK	UART2_GFCLK	PRCM	UART2 functional clock
UART3	UART3_ICLK	L4PER_L3_GICLK	PRCM	UART3 interface clock
	UART3_FCLK	UART3_GFCLK	PRCM	UART3 functional clock
UART4	UART4_ICLK	L4PER_L3_GICLK	PRCM	UART4 interface clock
	UART4_FCLK	UART4_GFCLK	PRCM	UART4 functional clock
UART5	UART5_ICLK	L4PER_L3_GICLK	PRCM	UART5 interface clock
	UART5_FCLK	UART5_GFCLK	PRCM	UART5 functional clock
UART6	UART6_ICLK	IPU_L3_GICLK	PRCM	UART6 interface clock
	UART6_FCLK	UART6_GFCLK	PRCM	UART6 functional clock
UART7	UART7_ICLK	L4PER2_L3_GICLK	PRCM	UART7 interface clock
	UART7_FCLK	UART7_GFCLK	PRCM	UART7 functional clock
UART8	UART8_ICLK	L4PER2_L3_GICLK	PRCM	UART8 interface clock
	UART8_FCLK	UART8_GFCLK	PRCM	UART8 functional clock
UART9	UART9_ICLK	L4PER2_L3_GICLK	PRCM	UART9 interface clock
	UART9_FCLK	UART9_GFCLK	PRCM	UART9 functional clock
UART10	UART10_ICLK	WKUPAON_GICLK	PRCM	UART10 interface clock
	UART10_FCLK	UART10_GFCLK	PRCM	UART10 functional clock
Resets				
Module Instance	Destination Signal	Source Signal	Source	Description
UART1	UART1_RST	L4PER_RET_RST	PRCM	UART1 reset
UART2	UART2_RST	L4PER_RET_RST	PRCM	UART2 reset
UART3	UART3_RST	L4PER_RET_RST	PRCM	UART3 reset
UART4	UART4_RST	L4PER_RET_RST	PRCM	UART4 reset
UART5	UART5_RST	L4PER_RET_RST	PRCM	UART5 reset
UART6	UART6_RST	IPU_RET_RST	PRCM	UART6 reset
UART7	UART7_RST	L4PER_RET_RST	PRCM	UART7 reset

Table 24-93. UART Clocks and Resets (continued)

UART8	UART8_RST	L4PER_RET_RST	PRCM	UART8 reset
UART9	UART9_RST	L4PER_RET_RST	PRCM	UART9 reset
UART10	UART10_RST	WKUPAON_RST	PRCM	UART10 reset

Table 24-94. UART Hardware Requests

Interrupt Requests

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR input	Default Mapping	Description
UART1	UART1_IRQ	IRQ_CROSSBAR_67	MPU_IRQ_72	UART 1 IRQ line
UART2	UART2_IRQ	IRQ_CROSSBAR_68	MPU_IRQ_73	UART 2 IRQ line
UART3	UART3_IRQ	IRQ_CROSSBAR_69	MPU_IRQ_74, IPU1_IRQ_45	UART 3 IRQ line
UART4	UART4_IRQ	IRQ_CROSSBAR_65	MPU_IRQ_70	UART 4 IRQ line
UART5	UART5_IRQ	IRQ_CROSSBAR_100	MPU_IRQ_105	UART 5 IRQ line
UART6	UART6_IRQ	IRQ_CROSSBAR_101	MPU_IRQ_106	UART 6 IRQ line
UART7	UART7_IRQ	IRQ_CROSSBAR_218	-	UART 7 IRQ line
UART8	UART8_IRQ	IRQ_CROSSBAR_219	-	UART 8 IRQ line
UART9	UART9_IRQ	IRQ_CROSSBAR_220	-	UART 9 IRQ line
UART10	UART10_IRQ	IRQ_CROSSBAR_221	-	UART 10 IRQ line

Direct Memory Access (DMA) Requests

Module Instance	Source Signal Name	Destination DMA_CROSSBAR input	Default Mapping	Description
UART1	UART1_DREQ_TX	DMA_CROSSBAR_49	DMA_SYSTEM_DREQ_48 DMA_EDMA_DREQ_48	UART 1 – transmit request
	UART1_DREQ_RX	DMA_CROSSBAR_50	DMA_SYSTEM_DREQ_49 DMA_EDMA_DREQ_49	UART 1 – receive request
UART2	UART2_DREQ_TX	DMA_CROSSBAR_51	DMA_SYSTEM_DREQ_50 DMA_EDMA_DREQ_50	UART 2 – transmit request
	UART2_DREQ_RX	DMA_CROSSBAR_52	DMA_SYSTEM_DREQ_51 DMA_EDMA_DREQ_51	UART 2 – receive request
UART3	UART3_DREQ_TX	DMA_CROSSBAR_53	DMA_SYSTEM_DREQ_52 DMA_EDMA_DREQ_52	UART 3 – transmit request
	UART3_DREQ_RX	DMA_CROSSBAR_54	DMA_SYSTEM_DREQ_53 DMA_EDMA_DREQ_53	UART 3 – receive request
UART4	UART4_DREQ_TX	DMA_CROSSBAR_55	DMA_SYSTEM_DREQ_54 DMA_EDMA_DREQ_54	UART 4 – transmit request
	UART4_DREQ_RX	DMA_CROSSBAR_56	DMA_SYSTEM_DREQ_55 DMA_EDMA_DREQ_55	UART 4 – receive request
UART5	UART5_DREQ_TX	DMA_CROSSBAR_63	DMA_SYSTEM_DREQ_62 DMA_EDMA_DREQ_62	UART 5 – transmit request
	UART5_DREQ_RX	DMA_CROSSBAR_64	DMA_SYSTEM_DREQ_63 DMA_EDMA_DREQ_63	UART 5 – receive request
UART6	UART6_DREQ_TX	DMA_CROSSBAR_79	DMA_SYSTEM_DREQ_78	UART 6 – transmit request
	UART6_DREQ_RX	DMA_CROSSBAR_80	DMA_SYSTEM_DREQ_79	UART 6 – receive request
UART7	UART7_DREQ_TX	DMA_CROSSBAR_144	-	UART 7 – transmit request.
	UART7_DREQ_RX	DMA_CROSSBAR_145	-	UART 7 – receive request
UART8	UART8_DREQ_TX	DMA_CROSSBAR_146	-	UART 8 – transmit request
	UART8_DREQ_RX	DMA_CROSSBAR_147	-	UART 8 – receive request
UART9	UART9_DREQ_TX	DMA_CROSSBAR_148	-	UART 9 – transmit request
	UART9_DREQ_RX	DMA_CROSSBAR_149	-	UART 9 – receive request
UART10	UART10_DREQ_TX	DMA_CROSSBAR_150	-	UART 10 – transmit request

Table 24-94. UART Hardware Requests (continued)

UART10_DREQ_RX	DMA_CROSSBAR_151	-	UART 10 – receive request

Note

The **Default Mapping** column in Table 24-94, *UART Hardware Requests*, shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.

For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

For more information about the device DMA_SYSTEM module, see *System DMA*.

For more information about the device EDMA module, see *Enhanced DMA*.

24.3.4 UART/IrDA/CIR Functional Description

24.3.4.1 Block Diagram

The UART/IrDA/CIR module can be divided into three main blocks:

- FIFO management
- Mode selection
- Protocol formatting

FIFO management is common to all functions and enables the transmission and reception of data from the host processor point of view.

There are two modes:

- Function mode: Routes the data to the chosen function (UART, IrDA, or CIR) and enables the mechanism corresponding to the chosen function
- Register mode: Enables conditional access to registers

For more information about mode configuration, see Section 24.3.4.7, *Mode Selection*.

Protocol formatting has three subcategories:

- Clock generation: The 48-MHz input clock generates all necessary clocks.
- Data formatting: Each function uses its own state-machine that is responsible for the transition between FIFO data and frame data associated with it.
- Interrupt management: Different interrupt types are generated depending on the chosen function. In each mode, when an interrupt is generated, the UART_IIR register indicates the interrupt type.
  - UART mode interrupts: Seven interrupts prioritized in six different levels
  - IrDA mode interrupts: Eight interrupts. The interrupt line is activated when any interrupt is generated (there is no priority).
  - CIR mode interrupts: A subset of existing IrDA mode interrupts is used.

In parallel with these functional blocks, a power-saving strategy exists for each function.

Figure 24-58 is the UART/IrDA/CIR block diagram.

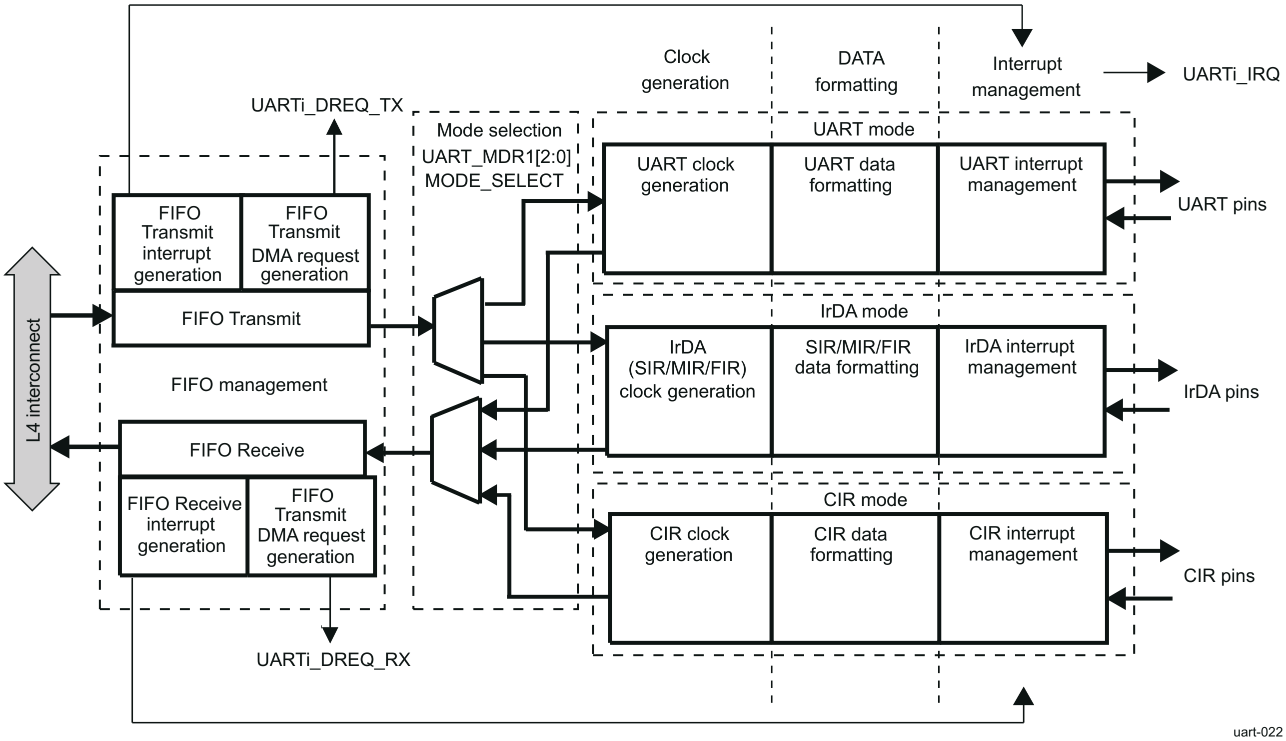


Figure 24-58. UART/IrDA/CIR Functional Block Diagram

24.3.4.2 Clock Configuration

Each UART uses a 48-MHz functional clock for its logic and to generate external interface signals. Each UART uses an interface clock for register accesses. The PRCM module generates and controls all these clocks (for more information, see *Clock Domain-Level Clock Management*, in *Power, Reset, and Clock Management*).

The idle and wake-up processes use a handshake protocol between the PRCM and the UART (for a description of the protocol, see *Clock Domain Level Clock Management*, in *Power, Reset, and Clock Management*). The UARTi.UART_SYSC[4:3] IDLEMODE bit field controls UART idle mode.

24.3.4.3 Software Reset

The UARTi.UART_SYSC[1] SOFTRESET bit controls the software reset; setting this bit to 1 triggers a software reset functionally equivalent to hardware reset.

24.3.4.4 Power Management

24.3.4.4.1 UART Mode Power Management

24.3.4.4.1.1 Module Power Saving

In UART modes, sleep mode is enabled by setting the UARTi.UART_IER[4] SLEEP_MODE bit to 1 (when the UARTi.UART_EFR[4] ENHANCED_EN bit is set to 1).

Sleep mode is entered when all of the following conditions exist:

- The serial data input line, uarti_rxd, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- The only pending interrupts are THR interrupts.

Sleep mode is a good way to lower UART power consumption, but this state can be achieved only when the UART is set to modem mode. Therefore, even if the UART has no key role functionally, it must be initialized in a functional mode to take advantage of sleep mode.

In sleep mode, the module clock and baud rate clock are stopped internally. Because most registers are clocked by these clocks, this greatly reduces power consumption. The module wakes up when a change is detected on the uarti_rxd line, when data is written to the TX FIFO, and when there is a change in the state of the modem input pins.

An interrupt can be generated on a wake-up event by setting the UARTi.UART_SCR[4] RX_CTS_WU_EN bit to 1. To understand how to manage the interrupt, see Section 24.3.4.5.1.2, *Wake-Up Interrupt*.

Note

There must be no writing to the divisor latches, UARTi.UART_DLL and UARTi.UART_DLH, to set the baud clock (BCLK) while in sleep mode. It is advisable to disable sleep mode using the UARTi.UART_IER[4] SLEEP_MODE bit before writing to the UARTi.UART_DLL or UARTi.UART_DLH register.

24.3.4.4.1.2 System Power Saving

Sleep and auto-idle modes are embedded power-saving features. Power-reduction techniques can be applied at the system level by shutting down certain internal clock and power domains of the device.

The UART supports an idle req/idle ack handshaking protocol used at the system level to shut down the UART clocks in a clean and controlled manner and to switch the UART from interrupt-generation mode to wake-up generation mode for unmasked events (see the UARTi.UART_SYSC[2] ENAWAKEUP bit and the UARTi.UART_WER register).

For more information, see *Module Level Clock Management*, *Power, Reset, and Clock Management*.

24.3.4.4.2 IrDA Mode Power Management (UART3 Only)

24.3.4.4.2.1 Module Power Saving

In IrDA modes, sleep mode is enabled by setting the UART3.UART_MDR1[3] IR_SLEEP bit to 1.

Sleep mode is entered when all of the following conditions exist:

- The serial data input line, RXD, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- No interrupts are pending except THR interrupts.

The module wakes up when a change is detected on the RXD line or when data is written to the TX FIFO.

24.3.4.4.2.2 System Power Saving

System power saving for the IrDA mode has the same function as for the UART mode (see Section 24.3.4.4.1.2, *System Power Saving*).

24.3.4.4.3 CIR Mode Power Management (UART3 Only)

24.3.4.4.3.1 Module Power Saving

Module power saving for the CIR mode has the same function as for the IrDA mode (see Section 24.3.4.4.2.1, *Module Power Saving*).

24.3.4.4.3.2 System Power Saving

System power saving for the CIR mode has the same function as for the UART mode (see Section 24.3.4.4.1.2, *System Power Saving*).

24.3.4.4.4 Local Power Management

Table 24-95 describes power-management features available for the UART.

Note

For information about source clock gating and the sleep/wake-up transitions description, see *Module Level Clock Management*, *Power, Reset, and Clock Management*.

Table 24-95. Local Power-Management Features

Feature	Registers	Description
Clock autogating	UART_SYSC[0] AUTOIDLE	This bit allows local power optimization in the module by gating the UARTi_ICLK clock on interface activity or gating the UARTi_FCLK clock on internal activity.
Slave idle modes	UART_SYSC[4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available.
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	UART_SYSC[2] ENAWAKEUP	This bit enables the wake-up feature at module level.
Wake-Up sources enable	N/A	Feature not available

24.3.4.5 Interrupt Requests

24.3.4.5.1 UART Mode Interrupt Management

24.3.4.5.1.1 UART Interrupts

UART mode includes seven possible interrupts prioritized to six levels.

When an interrupt is generated, the interrupt identification register (UARTi.UART_IIR) sets the UARTi.UART_IIR[0] IT_PENDING bit to 0 to indicate that an interrupt is pending, and indicates the type of interrupt through the UARTi.UART_IIR[5:1] bit field. Table 24-96 summarizes the interrupt control functions.

Table 24-96. UART Mode Interrupts

IIR[5:0]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Method
000001	N/A	No Interrupt	N/A	N/A
000110	1	Receiver line status	OE, FE, PE, or BI errors occur in characters in the RX FIFO.	FE, PE, BI: Read the UART_RHR register. OE: Read the UART_LSR register.
001100	2	RX time-out	Stale data in RX FIFO	Read the UART_RHR register.
000100	2	RHR interrupt	DRDY (data ready) (FIFO disabled) RX FIFO above trigger level (FIFO enabled)	Read the UART_RHR register until the interrupt condition disappears
000010	3	THR interrupt	TFE (THR empty) (FIFO disabled) TX FIFO below trigger level (FIFO enabled)	Write to the UART_THR until the interrupt condition disappears
000000	4	Modem status	See the UART_MSR register.	Read the MSR register
010000	5	XOFF interrupt/ special character interrupt	Receive XOFF characters/special character	Receive XON character(s), if XOFF interrupt/read of the UART_IIR register, if special character interrupt
100000	6	CTS, RTS	RTS pin or CTS pin change state from active (low) to inactive (high)	Read the UART_IIR register

For the receiver-line status interrupt, the RX_FIFO_STS bit (UARTi.UART_LSR[7]) generates the interrupt.

For the XOFF interrupt, if an XOFF flow character detection caused the interrupt, the interrupt is cleared by an XON flow character detection. If special character detection caused the interrupt, the interrupt is cleared by a read of the UARTi.UART_IIR register.

24.3.4.5.1.2 Wake-Up Interrupt

Wake-up interrupt is a special interrupt that works differently from other interrupts. This interrupt is enabled when the UARTi.UART_SCR[4] RX_CTS_WU_EN bit is set to 1. The UARTi.UART_IIR register is not modified when this occurs; the UART3.UART_SSR[1] RX_CTS_WU_STS bit must be checked to detect a wake-up event.

When a wake-up interrupt occurs, it can be cleared only by resetting the UARTi.UART_SCR[4] RX_CTS_WU_EN bit. This bit must be reenabled (set to 1) after the current wake-up interrupt event is processed to detect the next incoming wake-up event.

24.3.4.5.2 IrDA Mode Interrupt Management

24.3.4.5.2.1 IrDA Interrupts

The IrDA function generates interrupts. All interrupts can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (UART3.UART_IER_IRDA). The interrupt status of the device can be checked by reading the interrupt identification register (UART3.UART_IIR_IRDA).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different UART3.UART_IER_IRDA and UART3.UART_IIR_IRDA mappings, depending on the selected mode.

IrDA modes have eight possible interrupts (see Table 24-97). The interrupt line is activated when any interrupt is generated (there is no priority).

Table 24-97. IrDA Mode Interrupts

IIR_IRDA Bit	Interrupt Type	Interrupt Source	Interrupt Reset Method
0	RHR interrupt	DRDY (data ready) (FIFO disabled) RX FIFO above trigger level (FIFO enabled)	Read the UART_RHR register until the interrupt condition disappears.
1	THR interrupt	TFE (THR empty) (FIFO disabled) TX FIFO below trigger level (FIFO enabled)	Write to the UART_THR until the interrupt condition disappears.
2	Last byte in RX FIFO	Last byte of frame in RX FIFO is available to be read at the RHR port.	Read the UART_RHR register.
3	RX overrun	Write to the UART_RHR register when the RX FIFO is full.	Read UART_RESUME register.
4	Status FIFO interrupt	Status FIFO triggers level reached.	Read STATUS FIFO.
5	TX status	THR empty before EOF sent. Last bit of transmission of the IrDA frame occurred, but with an underrun error OR Transmission of the last bit of the IrDA frame completed successfully.	Read the UART_RESUME register OR Read the UART_IIR_IRDA register.
6	Receiver line status interrupt	CRC, ABORT, or frame-length error is written into the STATUS FIFO.	Read the STATUS FIFO (read until empty - maximum of eight reads required).
7	Received EOF	Received end-of-frame	Read the UART_IIR_IRDA register.

24.3.4.5.2.2 Wake-Up Interrupts

The wake-up interrupt for IrDA mode has the same function as that for UART mode (see Section 24.3.4.5.1.2, *Wake-Up Interrupt*).

CAUTION

Wake-up interface implementation in this mode is based on the UARTi_SIDLEACK low-to-high transition instead of the UARTi_SIDLEACK state.

This does not ensure wake-up event generation as expected when configured in smart-idle mode, and the system wakes up for a short period.

24.3.4.5.3 CIR Mode Interrupt Management

24.3.4.5.3.1 CIR Interrupts

The CIR function generates interrupts that can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (UART3.UART_IER_CIR). The interrupt status of the device can be checked by reading the interrupt identification register (UART3.UART_IIR_CIR).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different UART3.UART_IER_CIR and UART3.UART_IIR_CIR mappings, depending on the selected mode.

Table 24-98 lists the interrupt modes to be maintained. In CIR mode, the sole purpose of the UART3.UART_IIR_CIR[5] bit is to indicate that the last bit of infrared data was passed to the uart3_rctx pin.

Table 24-98. CIR Mode Interrupts

IIR_CIR Bit Number	Interrupt Type	Interrupt Source	Interrupt Reset Method
0	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
1	THR interrupt	TFE (THR empty) (FIFO disabled) TX FIFO below trigger level (FIFO enabled)	Write to the THR register until the interrupt condition disappears
2	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
3	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
4	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
5	TX status	Transmission of the last bit of the frame is complete successfully	Read the IIR_CIR register
6	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
7	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode

24.3.4.5.3.2 Wake-Up Interrupts

The wake-up interrupt for CIR mode has the same function as that for UART mode (see Section 24.3.4.5.1.2, *Wake-Up Interrupt*).

24.3.4.6 FIFO Management

The FIFO is accessed by reading and writing the UARTi.UART_RHR and UARTi.UART_THR registers. Parameters are controlled using the FIFO control register (UARTi.UART_FCR) and supplementary control register (UARTi.UART_SCR). Reading the UARTi.UART_SSR[0] TX_FIFO_FULL bit at 1 means the FIFO is full.

The UARTi.UART_TLR register controls the FIFO trigger level, which enables DMA and interrupt generation. After reset, transmit (TX) and receive (RX) FIFOs are disabled; thus, the trigger level is the default value of 1 byte. Figure 24-59 shows the FIFO management registers.

Note

Data in the UARTi.UART_RHR register is not overwritten when an overflow occurs.

Note

The UARTi.UART_SFLSR, UARTi.UART_SFREGL, and UARTi.UART_SFREGH status registers are used in IrDA mode only. For information about their use, see Section 24.3.4.8.2.3, *IrDA Data Formatting*.

Note

Bits UARTI.UART_FCR[2] TX_FIFO_CLEAR and UARTI.UART_FCR[1] RX_FIFO_CLEAR are automatically cleared by hardware after 4* UARTi_ICLK + 5* UARTi_FCLK clock cycles. This delay is needed to finish the resetting of the corresponding FIFO and DMA control registers.

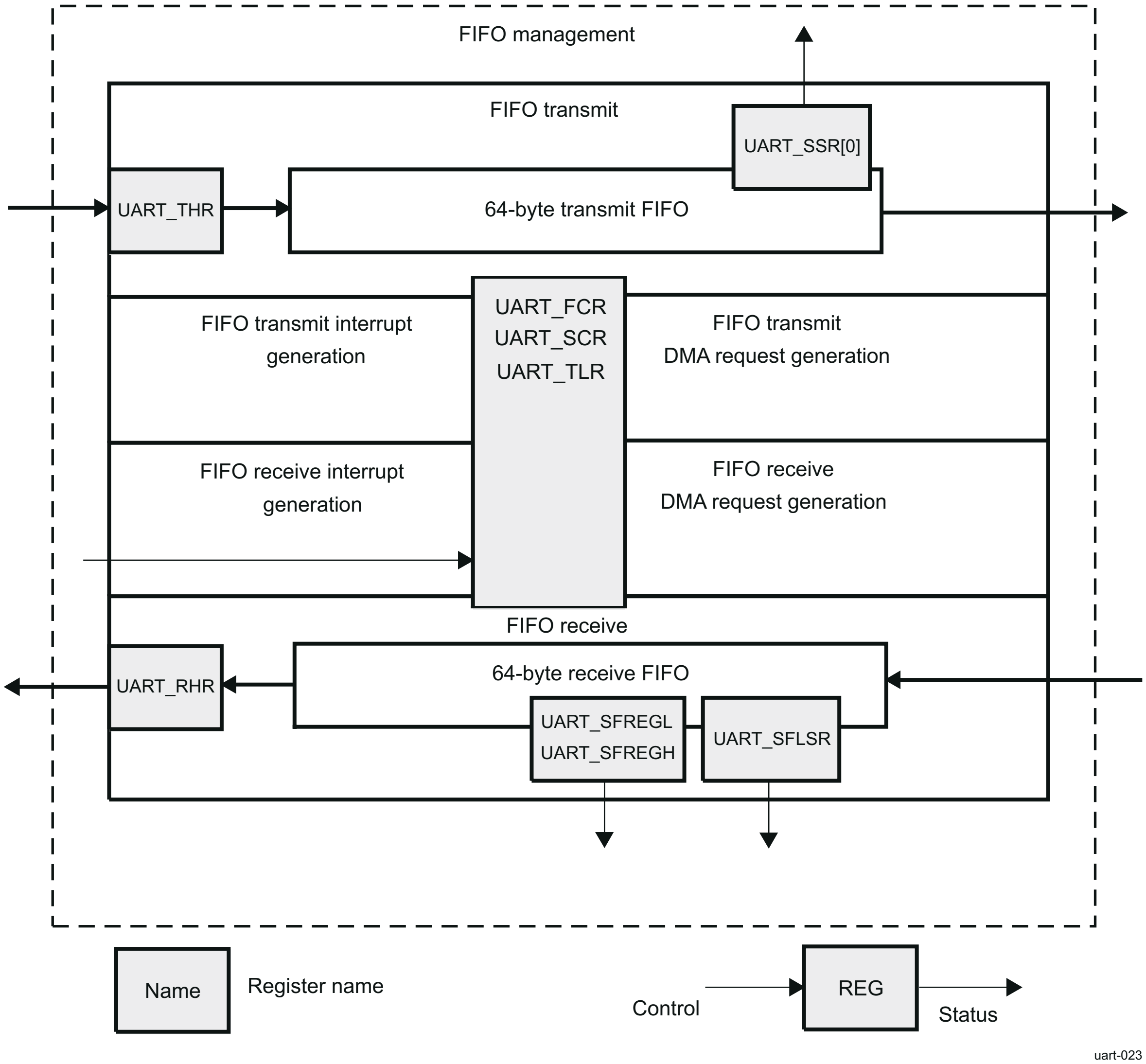


Figure 24-59. FIFO Management Registers

24.3.4.6.1 FIFO Trigger

24.3.4.6.1.1 Transmit FIFO Trigger

Table 24-99 lists the TX FIFO trigger level settings.

Table 24-99. TX FIFO Trigger Level Setting Summary

SCR[6]	TLR[3:0]	TX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field (8,16, 32, or 56 spaces)
0	!= 0x0	Defined by the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field (from 4 to 60 spaces with a granularity of 4 spaces)
1	Value	Defined by the concatenated value of TX_FIFO_TRIG_DMA and TX_FIFO_TRIG (from 1 to 63 spaces with a granularity of 1 space) **Note:** The combination of TX_FIFO_TRIG_DMA = 0x0 and TX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of 1 space required). All zeros result in unpredictable behavior.

24.3.4.6.1.2 Receive FIFO Trigger

Table 24-100 lists the RX FIFO trigger-level settings.

Table 24-100. RX FIFO Trigger-Level Setting Summary

SCR[7]	TLR[7:4]	RX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field (8,16, 56, or 60 characters)
0	!= 0x0	Defined by the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA bit field (from 4 to 60 characters with a granularity of 4 characters)
1	Value	Defined by the concatenated value of RX_FIFO_TRIG_DMA and RX_FIFO_TRIG (from 1 to 63 characters with a granularity of 1 character) **Note:** The combination of RX_FIFO_TRIG_DMA = 0x0 and RX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of 1 character required). All zeros result in unpredictable behavior.

The receive threshold is programmed using the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START and UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit fields:

- Trigger levels from 0 to 60 bytes are available with a granularity of 4 (trigger level = 4 × [4-bit register value]).
- To ensure correct device operation, ensure that RX_FIFO_TRIG_HALT > RX_FIFO_TRIG when auto-RTS is enabled.

  Delay = [4 + 16 × (1 + CHAR_LENGTH + Parity + Stop – 0.5)] × Baud_rate + 4 × FCLK

  Note

  The RTS signal is deasserted after the UART module receives the data over RX_FIFO_TRIG_HALT. Delay means how long the UART module takes to deassert the RTS signal after reaching RX_FIFO_TRIG_HALT.

- In FIFO interrupt mode with flow control, ensure that the trigger level to HALT transmission is greater than or equal to the RX FIFO trigger level (the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist, because a DMA request is sent when a byte is received.

24.3.4.6.2 FIFO Interrupt Mode

In FIFO interrupt mode (the FIFO control register UARTi.UART_FCR[0] FIFO_EN bit is set to 1 and relevant interrupts are enabled by the UARTi.UART_IER register), an interrupt signal informs the processor of the status of the receiver and transmitter. These interrupts are raised when the RX/TX FIFO threshold (the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA and UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit fields or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG and UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit fields, respectively) is reached.

The interrupt signals instruct the MPU to transfer data to the destination (from the UART in receive mode and/or from any source to the UART FIFO in transmit mode).

When UART flow control is enabled with interrupt capabilities, the UART flow control FIFO threshold (the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field) must be greater than or equal to the RX FIFO threshold.

Figure 24-60 shows the generation of the RX FIFO interrupt request.

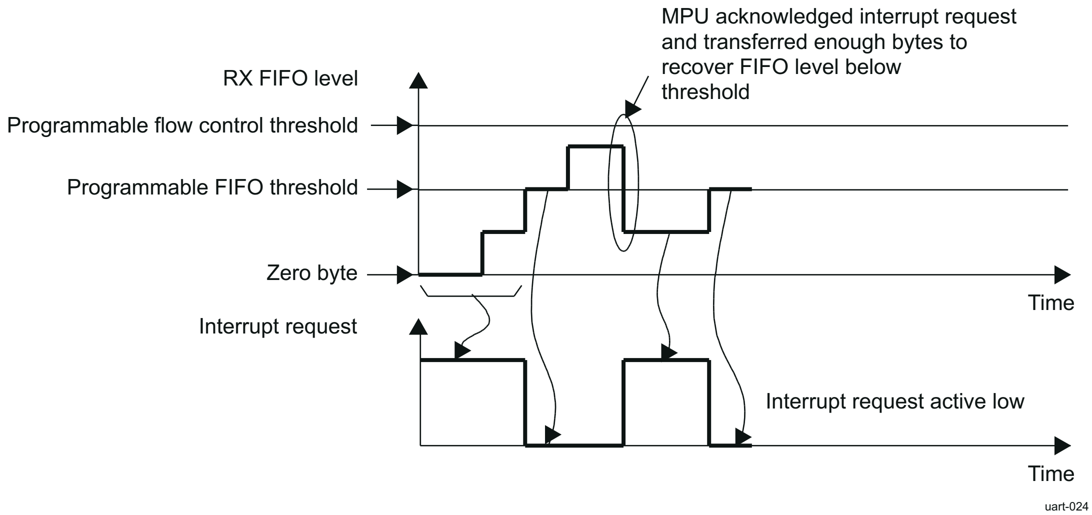


Figure 24-60. RX FIFO Interrupt Request Generation

In receive mode, no interrupt is generated until the RX FIFO reaches its threshold. Once low, the interrupt can be deasserted only when the MPU has handled enough bytes to put the FIFO level below threshold. The flow control threshold is set at a higher value than the FIFO threshold.

Figure 24-61 shows the generation of the TX FIFO interrupt request.

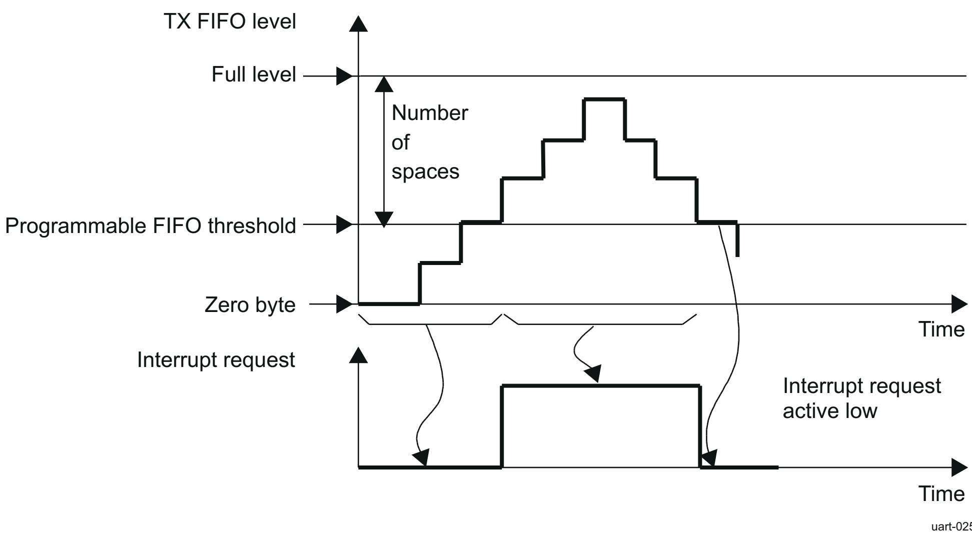


Figure 24-61. TX FIFO Interrupt Request Generation

In transmit mode, an interrupt request is automatically asserted when the TX FIFO is empty. This request is deasserted when the TX FIFO crosses the threshold level. The interrupt line is deasserted until a sufficient number of elements is transmitted to go below the TX FIFO threshold.

24.3.4.6.3 FIFO Polled Mode Operation

In FIFO polled mode (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0 and the relevant interrupts are disabled by the UARTi.UART_IER register), the status of the receiver and transmitter can be checked by polling the line status register (UARTi.UART_LSR).

This mode is an alternative to the FIFO interrupt mode of operation in which the status of the receiver and transmitter is automatically determined by sending interrupts to the MPU.

24.3.4.6.4 FIFO DMA Mode Operation

Although the DMA operation includes four modes (DMA modes 0 through 3), the information in Table 24-94, *UART Hardware Requests*, assumes that mode 1 is used. (Mode 2 and mode 3 are legacy modes that use only one DMA request for each module.)

In mode 2, the remaining DMA request is used for RX. In mode 3, the remaining DMA request is used for TX.

DMA requests in mode 2 and mode 3 use the UARTi_DREQ_TX signals (where i = 1 to 10).

The UARTi_DREQ_RX signals are not used by the module in mode 2 and mode 3:

The DMA mode and signals usage can be selected as follows:

- When the UARTi.UART_SCR[0] DMA_MODE_CTL bit is set to 0, setting the UARTi.UART_FCR[3] DMA_MODE bit to 0 enables DMA mode 0. Setting the UARTi.UART_FCR[3] DMA_MODE bit to 1 enables DMA mode 1.
- When the UARTi.UART_SCR[0] DMA_MODE_CTL bit is set to 1, the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field determines DMA mode 0 to mode 3 based on the supplementary control register (SCR) description.

For example:

- If no DMA operation is desired, set theUARTi.UART_SCR[0] DMA_MODE_CTL bit to 1 and the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field to 0x0. (The DMA_MODE bit is discarded.)
- If DMA mode 1 is desired, set the UARTi.UART_SCR[0] DMA_MODE_CTL bit to 0 and the UARTi.UART_FCR[3] DMA_MODE bit to 1, or set the UARTi.UART_SCR[0] DMA_MODE_CTL bit to 1 and the SCR[2:1] DMA_MODE_2 bit field to 01. (The UARTi.UART_FCR[3] DMA_MODE bit is discarded.)

If the FIFOs are disabled (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0), the DMA occurs in single-character transfers.

When DMA mode 0 is programmed, the signals associated with DMA operation are not active.

Depending on UARTi.UART_MDR3[2] SET_DMA_TX_THRESHOLD, the threshold can be programmed different ways:

- SET_TX_DMA_THRESHOLD = 1:

  The threshold value will be the value of the TX_DMA_THRESHOLD register. If SET_TX_DMA_THRESHOLD + TX trigger spaces 64, then the default method of threshold is used: threshold value = TX FIFO size.
- SET_TX_DMA_THRESHOLD = 0:

  The threshold value = TX FIFO size TX trigger space. The TX DMA line is asserted if the TX FIFO level is lower then the threshold. It remains asserted until TX trigger spaces number of bytes are written into the FIFO. The DMA line is then deasserted and the FIFO level is compared with the threshold value.

24.3.4.6.4.1 DMA sequence to disable TX DMA

In order to disable TX DMA if it is not needed anymore (e.g. all transfers are done and UART idle mode is desired), the following sequence must be use

1. DMA mode 1 is set (both TX/RX DMA) by registers UARTi.UART_SCR[0] = 0 and UARTi.UART_FCR[3] = 1:
   a. Set the UARTi.UART_SCR[2:1] DMA_MODE_2 bit fields to 01 (DMA mode 1)
   b. Set the UARTi.UART_SCR[0] DMA_MODE_CTL bit to1 (this setting of UARTi.UART_SCR[0] DMA_MODE-CTL will ignores UARTi.UART_FCR[3] DMA_MODE_CTL bit)

   Note

   It is strongly suggested to do steps 'a' and 'b' in two separate write in order to avoid malfunction of the device.

   c. Set the UARTi.UART_FCR[3] DMA_MODE bit to 0. It is not necessary but suggested to avoid restore of DMA mode 1 during accidental reset of UARTi.UART_SCR[0] DMA_MODE_CTL bit. Be sure that all data was read out from RX FIFO and if it possible disable the RX side. In UART mode the RTS/CTS or XOFF/XON protocol can be used. In IrDA modes RX can be forcibly disabled by setting UARTi.UART_ACREG[5] DIS_IR_RX bit

Note

There can be RX DATA loss during the next steps if all DATA was not read out or there was an ongoing reception!

d. Set the UARTi.UART_FCR[2:1] DMA_MODE bit field to 11 (clear TX and RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO).
e. Set the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field to 10 (DMA mode 2, RX only).
f. Set the UARTi.UART_FCR[2:1] DMA_MODE bit field to 11 (clear TX and RX FIFO and the DMA request again).
g. Set the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field to 00 (no DMA) or keep 10 if RX DMA is needed.

2. DMA mode 1 is set (both TX/RX DMA) by registers UARTi.UART_FCR[3] = 0 and UARTi.UART_SCR[0] = 1, UARTi.UART_SCR[2:1] = 01. It is almost the same as above, but steps ‘a’, and ‘b’ can be skipped:
   a. Set the UARTi.UART_FCR[3] DMA_MODE bit to 0. It is not necessary but suggested to avoid restore of DMA mode 1 during accidental reset of UARTi.UART_SCR[0] DMA_MODE_CTL bit. Be sure that all data was read out from RX FIFO and if it possible disable the RX side. In UART mode the RTS/CTS or XOFF/XON protocol can be used. In IrDA modes RX can be forcibly disabled by setting UARTi.UART_ACREG[5] DIS_IR_RX bit

   Note

   There can be RX DATA loss during the next steps if all DATA was not read out or there was an ongoing reception!

   b. Set the UARTi.UART_FCR[2:1] DMA_MODE bit field to 11 (clear TX and RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO).
   c. Set the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field to 10 (DMA mode 2, RX only).
   d. Set the UARTi.UART_FCR[2:1] DMA_MODE bit field to 11 (clear TX and RX FIFO and the DMA request again).
   e. Set the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field to 00 (no DMA) or keep 10 if RX DMA is needed.
3. DMA mode 3 is set (TX DMA only) by registers UARTi.UART_FCR[3] = 0 and UARTi.UART_SCR[0] = 1, UARTi.UART_SCR[2:1] = 11. It is the same as above:
   a. Set the UARTi.UART_FCR[3] DMA_MODE bit to 0. It is not necessary but suggested to avoid restore of DMA mode 1 during accidental reset of UARTi.UART_SCR[0] DMA_MODE_CTL bit. Be sure that all data was read out from RX FIFO and if it possible disable the RX side. In UART mode the RTS/CTS or XOFF/XON protocol can be used. In IrDA modes RX can be forcibly disabled by setting UARTi.UART_ACREG[5] DIS_IR_RX bit

   Note

   There can be RX DATA loss during the next steps if all DATA was not read out or there was an ongoing reception!

   b. Set the UARTi.UART_FCR[2:1] DMA_MODE bit field to 11 (clear TX and RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO).
   c. Set the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field to 10 (DMA mode 2, RX only).
   d. Set the UARTi.UART_FCR[2:1] DMA_MODE bit field to 11 (clear TX and RX FIFO and the DMA request again).
   e. Set the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field to 00 (no DMA) or keep 10 if RX DMA is needed.

24.3.4.6.4.2 DMA Transfers (DMA Mode 1, 2, or 3)

Figure 24-62 through Figure 24-65 show the supported DMA operations.

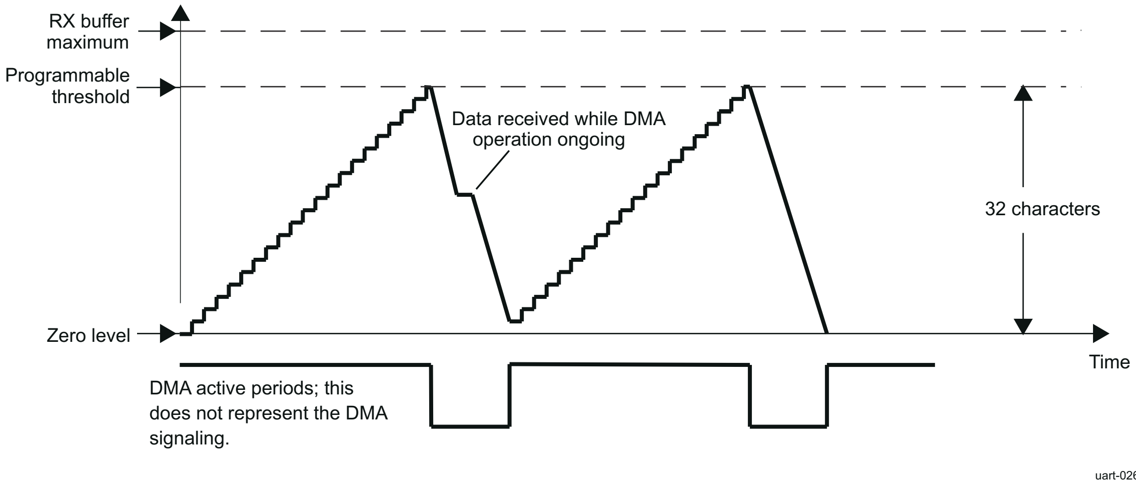


Figure 24-62. Receive FIFO DMA Request Generation (32 Characters)

In receive mode, a DMA request is generated when the RX FIFO reaches its threshold level defined in the trigger level register (UARTi.UART_TLR). This request is deasserted when the number of bytes defined by the threshold level is read by the device DMA controllers.

In transmit mode, a DMA request is automatically asserted when the TX FIFO is empty. This request is deasserted when the number of bytes defined by the number of spaces in the UARTi.UART_TLR register is written by the device DMA controllers. If an insufficient number of characters is written, the DMA request stays active.

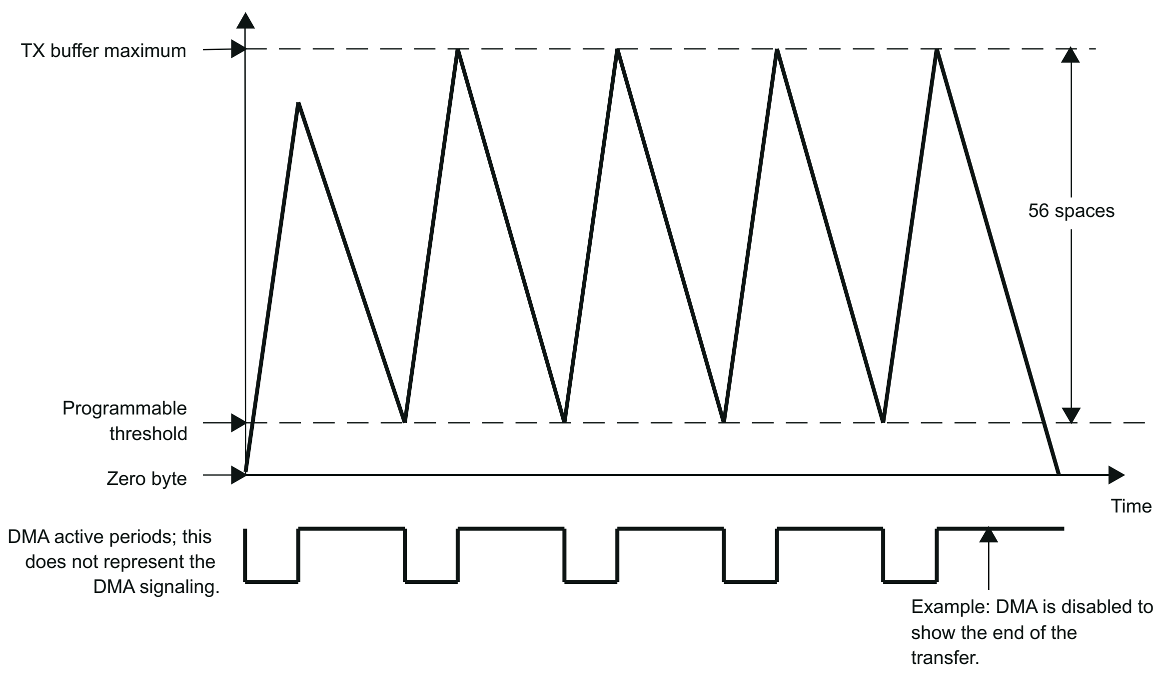


Figure 24-63. Transmit FIFO DMA Request Generation (56 Spaces)

The DMA request is again asserted if the FIFO can receive the number of bytes defined by the UARTi.UART_TLR register.

The threshold can be programmed in a number of ways. Figure 24-63 shows a DMA transfer operating with a space setting of 56 that can arise from using the auto settings in the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field or the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field concatenated with the TX_FIFO_TRIG bit field.

The setting of 56 spaces in the UART/IrDA/CIR module must correlate with the settings of the device DMA controllers, so that the buffer does not overflow (program the DMA request size of the LH controller to equal the number of spaces in the UART/IrDA/CIR module).

Figure 24-64 shows an example with eight spaces to show the buffer level crossing the space threshold. The LH DMA controller settings must correspond to those of the UART/IrDA/CIR module.

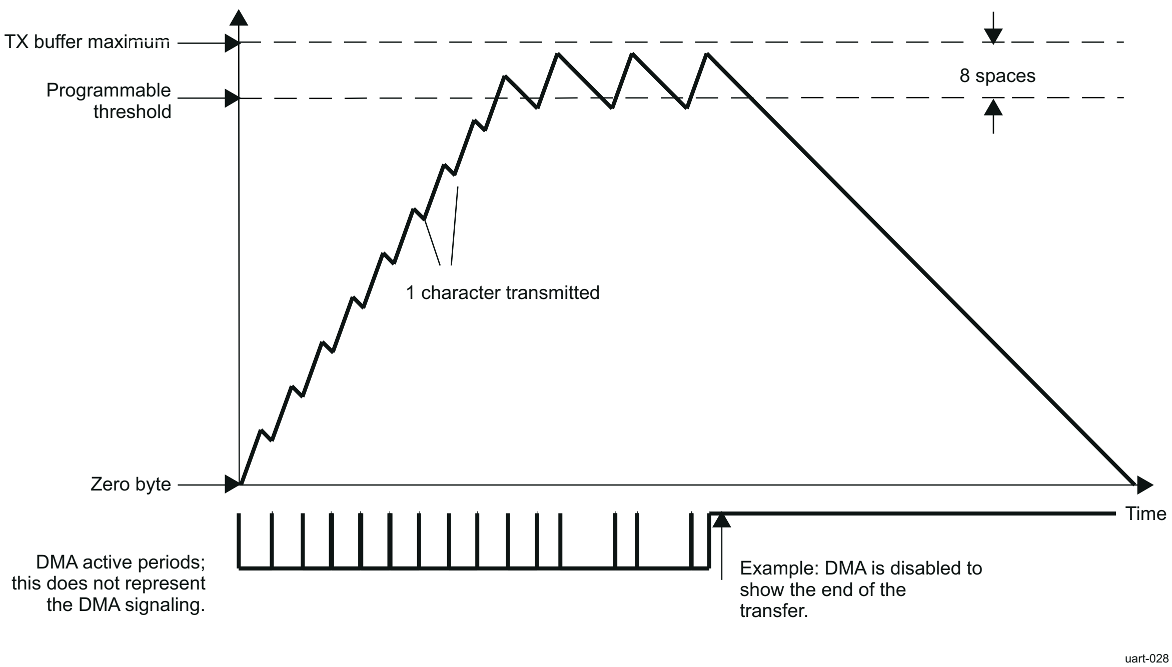


Figure 24-64. Transmit FIFO DMA Request Generation (8 Spaces)

The next example shows the setting of one space that uses the DMA for each transfer of one character to the transmit buffer (see Figure 24-65). The buffer is filled faster than the baud rate at which data is transmitted to the TX pin. Eventually, the buffer is completely full and the DMA operations stop transferring data to the transmit buffer.

On two occasions, the buffer holds the maximum amount of data words; shortly after this, the DMA is disabled to show the slower transmission of the data words to the TX pin. Eventually, the buffer is emptied at the rate specified by the baud rate settings of the UARTi.UART_DLL and UARTi.UART_DLH registers.

The DMA settings must correspond to the system LH DMA controller settings to ensure correct operation of this logic.

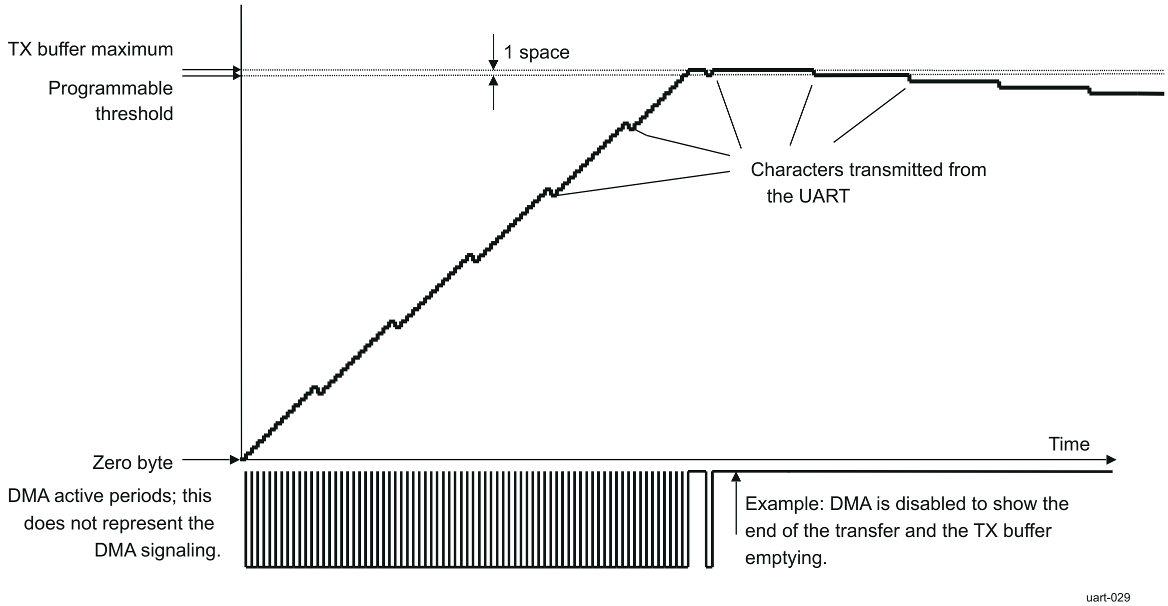


Figure 24-65. Transmit FIFO DMA Request Generation (1 Space)

The final example illustrates the setting of eight spaces but setting the TX DMA threshold directly by setting UART_MDR3[1] NONDEFAULT_FREQ bit and UART_TX_DMA_THRESHOLD register (see Figure 24-66). In the example, UART_TX_DMA_THRESHOLD[5:0] TX_DMA_THRESHOLD = 3 and the trigger level is 8. The buffer is filled at a faster rate than the BAUD rate transmits data to the TX pin. The buffer is filled with 8 bytes and the DMA operations stop transferring data to the transmit buffer. When the buffer is emptied to the threshold level by transmission, the DMA operation activates again to fill the buffer with 8 bytes.

Eventually, the buffer will be emptied at the rate specified by the BAUD Rate settings of the UART_DLL and UART_DLH registers.

If the selected threshold level + trigger level exceeds max buffer size, then the original TX DMA threshold method is used to prevent TX overrun, regardless of the UART_MDR3[1] NONDEFAULT_FREQ value.

The DMA settings should correspond to the system Local Host DMA controller settings in order to ensure the correct operation of this logic.

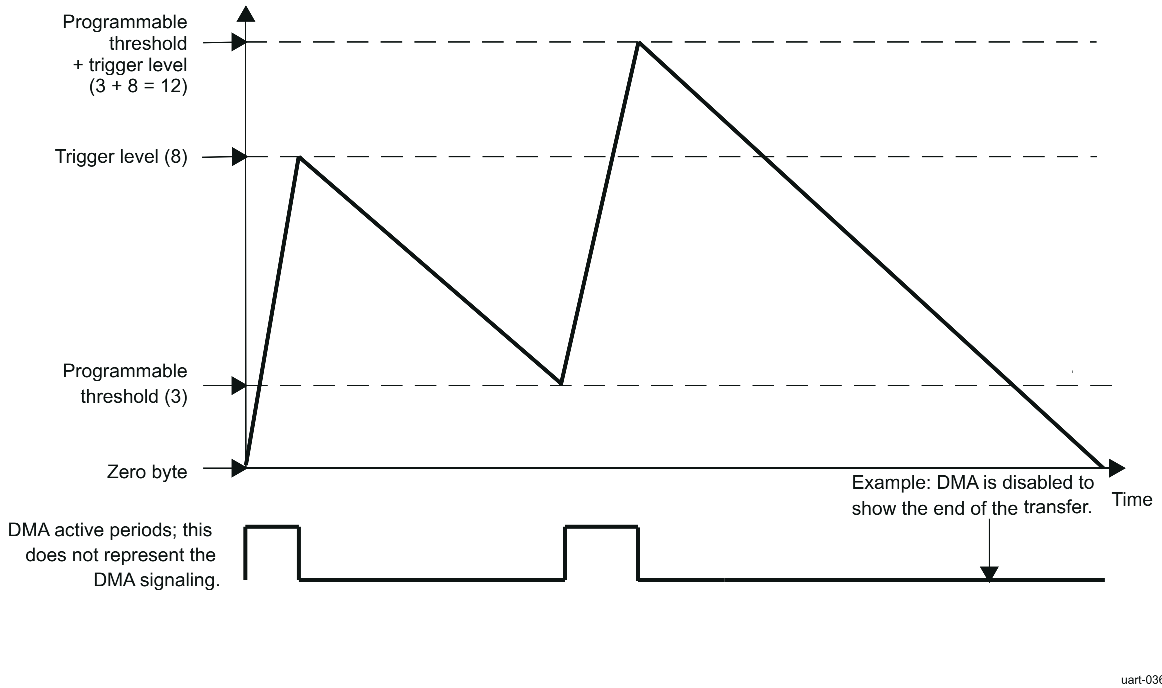


Figure 24-66. Transmit FIFO DMA Request Generation Using Direct TX DMA Threshold Programming. (Threshold = 3; Spaces = 8)

24.3.4.6.4.3 DMA Transmission

Figure 24-67 shows DMA transmission.

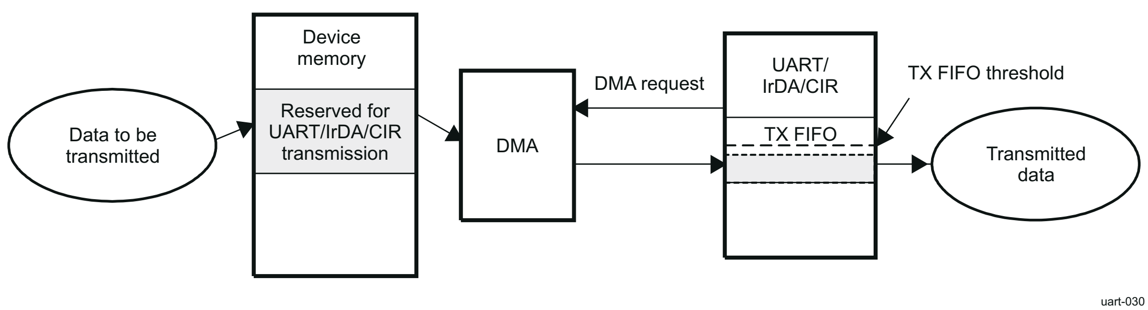


Figure 24-67. DMA Transmission

1. Data to be transmitted are put in the device memory reserved for UART/IrDA/CIR transmission by the DMA:
    a. Until the TX FIFO trigger level is not reached, a DMA request is generated
    b. An element (1 byte) is transferred from the SDRAM to the TX FIFO at each DMA request (DMA element synchronization).
2. Data in the TX FIFO are automatically transmitted.
3. The end of the transmission is signaled by the UARTi.UART_THR empty (TX FIFO empty).

Note

In IrDA mode, the transmission does not end immediately after the TX FIFO empties, at which point the last data byte, the CRC field, and the stop flag still must be transmitted; thus, the end of transmission occurs a few milliseconds after the UARTi.UART_THR register empties.

24.3.4.6.4.4 DMA Reception

Figure 24-68 shows DMA reception.

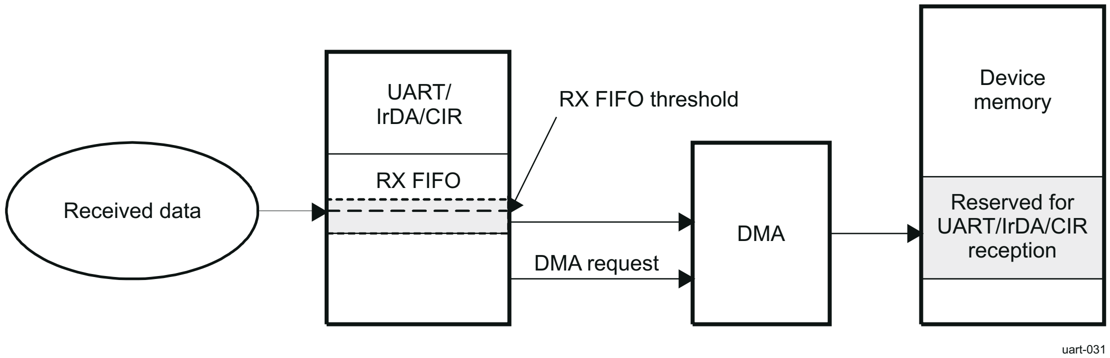


Figure 24-68. DMA Reception

1. Enable the reception.
2. Received data are put in the RX FIFO.
3. Data are transferred from the RX FIFO to the device memory by the DMA:
   a. At each received byte, the RX FIFO trigger level (one character) is reached and a DMA request is generated.
   b. An element (1 byte) is transferred from the RX FIFO to the SDRAM at each DMA request (DMA element synchronization).
4. The end of the reception is signaled by the EOF interrupt.

24.3.4.7 Mode Selection

24.3.4.7.1 Register Access Modes

24.3.4.7.1.1 Operational Mode and Configuration Modes

Register access depends on the register access mode, although register access modes are not correlated to functional mode selection. Three different modes are available:

- Operational mode
- Configuration mode A
- Configuration mode B

Operational mode is the selected mode when the function is active; serial data transfer can be performed in this mode.

Configuration mode A and configuration mode B are used during module initialization steps. These modes enable access to configuration registers, which are hidden in the operational mode. The modes are used when the module is inactive (no serial data transfer processed) and only for initialization or reconfiguration of the module.

The value of the UARTi.UART_LCR register determines the register access mode (see Table 24-101).

Table 24-101. UART/IrDA/CIR Register Access Mode Programming (Using UART_LCR)

Mode	Condition
Configuration mode A	UART_LCR[7] = 0x1 and UART_LCR[7:0] != 0xBF
Configuration mode B	UART_LCR[7] = 0x1 and UART_LCR[7:0] = 0xBF
Operational mode	UART_LCR[7] = 0x0

24.3.4.7.1.2 Register Access Submode

In each access register mode (operational mode or configuration mode A/B), some register accesses are conditional on the programming of a submode (MSR_SPR, TCR_TLR, and XOFF). These registers are identified in Section 24.3.6, *UART/IrDA/CIR Register Manual*.

Table 24-102 through Table 24-104 summarize the register access submodes.

Table 24-102. Subconfiguration Mode A Summary

Mode	Condition
MSR_SPR	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

Table 24-103. Subconfiguration Mode B Summary

Mode	Condition
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1
XOFF	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)

Table 24-104. Suboperational Mode Summary

Mode	Condition
MSR_SPR	UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

24.3.4.7.1.3 Registers Available for the Register Access Modes

Table 24-105 lists the names of the register bits in each access register mode. Gray shading indicates that the register does not depend on the register access mode (available in all modes).

Table 24-105. UART/IrDA/CIR Register Access Mode Overview

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER	UART_IER
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR	UART_FCR
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x010	UART_MCR	UART_MCR	UART_XON1_ADDR1	UART_XON1_ADDR1	UART_MCR	UART_MCR
0x014	UART_LSR	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR	–
0x018	UART_MSR / UART_TCR	UART_TCR	UART_TCR / UART_XOFF1	UART_TCR / UART_XOFF1	UART_MSR / UART_TCR	UART_TCR
0x01C	UART_SPR / UART_TLR	UART_SPR / UART_TLR	UART_TLR / UART_XOFF2	UART_TLR / UART_XOFF2	UART_SPR / UART_TLR	UART_SPR / UART_TLR
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL
0x02C	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH
0x030	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH
0x038	UART_UASR	–	UART_UASR	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR

Table 24-105. UART/IrDA/CIR Register Access Mode Overview (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

24.3.4.7.2 UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection

To select a mode, set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field (see Table 24-106).

Table 24-106. UART Mode Selection

Value	Mode
0x0:	UART 16x mode
0x1:	SIR mode (UART3 only)
0x2:	UART 16x auto-baud
0x3:	UART 13x mode
0x4:	MIR mode (UART3 only)
0x5:	FIR mode (UART3 only)
0x6:	CIR mode (UART3 only)

MODE_SELECT is effective when the module is in operational mode (see Section 24.3.4.7.1, *Register Access Modes*).

24.3.4.7.2.1 Registers Available for the UART Function

Only the registers listed in Table 24-107 are used for the UART function.

Table 24-107. UART Mode Register Overview

Address Offset	Registers[(1) (2)]					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER (UART)	UART_IER (UART)
0x008	UART_IIR	UART_FCR	UART_EFR [4]	UART_EFR [4]	UART_IIR (UART)	UART_FCR (UART)
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR

Table 24-107. UART Mode Register Overview (continued)

Address Offset	Registers(1) (2)					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x010	UART_MCR	UART_MCR	UART_XON1_ADDR1	UART_XON1_ADDR1	UART_MCR	UART_MCR
0x014	UART_LSR (UART)	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR (UART)	–
0x018	UART_MSR/ UART_TCR	UART_TCR	UART_XOFF1/ UART_TCR	UART_XOFF1/ UART_TCR	UART_MSR/ UART_TCR	UART_TCR
0x01C	UART_TLR/ UART_SPR	UART_TLR/ UART_SPR	UART_TLR/ UART_XOFF2	UART_TLR/ UART_XOFF2	UART_TLR/ UART_SPR	UART_TLR/ UART_SPR
0x020	UART_MDR1	UART_MDR1 [2:0]	UART_MDR1 [2:0]	UART_MDR1 [2:0]	UART_MDR1 [2:0]	UART_MDR1 [2:0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	–	–	–	–	–	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	UART_UASR	–	UART_UASR	–	–	–
0x03C	–	–	–	–	–	–
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	–	–
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	–	–	–	–	–	–
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

(1) REGISTER_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions (described separately in Section 24.3.6, *UART/IrDA/CIR Register Manual*).
(2) REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

24.3.4.7.2.2 Registers Available for the IrDA Function (UART3 Only)

Only the registers listed in Table 24-108 are used for the IrDA function.

Table 24-108. IrDA Mode Register Overview

Address Offset	Registers(1) (2)					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER (IrDA)	UART_IER (IrDA)

Table 24-108. IrDA Mode Register Overview (continued)

Address Offset	Registers(1) (2)					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x008	UART_IIR	UART_FCR	UART_EFR [4]	UART_EFR [4]	UART_IIR (IrDA)	UART_FCR (IrDA)
0x00C	UART_LCR [7]	UART_LCR [7]	UART_LCR [7]	UART_LCR [7]	UART_LCR [7]	UART_LCR [7]
0x010	–	–	UART_XON1_ADDR1	UART_XON1_ADDR1	–	–
0x014	UART_LSR (IrDA)	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR (IrDA)	–
0x018	UART_MSR/ UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/ UART_TCR	UART_TCR
0x01C	UART_TLR/ UART_SPR	UART_TLR/ UART_SPR	UART_TLR	UART_TLR	UART_TLR/ UART_SPR	UART_TLR/ UART_SPR
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL
0x02C	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH
0x030	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH
0x038	–	–	–	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]
0x060	–	–	–	–	–	–
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

(1) REGISTER_NAME(IrDA) notation indicates that the register exists for other functions (UART or CIR), but fields have different meanings for other functions (described separately in Section 24.3.6, *UART/IrDA/CIR Register Manual*).
(2) REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the IrDA function.

24.3.4.7.2.3 Registers Available for the CIR Function (UART3 Only)

Only the registers listed in Table 24-109 are used for the CIR function.

Table 24-109. CIR Mode Register Overview

Address Offset	Registers[(1) (2)]					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	–	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER (CIR)	UART_IER (CIR)
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR (CIR)	UART_FCR (CIR)
0x00C	UART_LCR	UART_LCR [7]	UART_LCR [7]	UART_LCR [7]	UART_LCR [7]	UART_LCR [7]
0x010	–	–	–	–	–	–
0x014	UART_LSR (CIR)	–	–	–	UART_LSR (CIR)	–
0x018	UART_MSR/ UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/ UART_TCR	UART_TCR
0x01C	UART_TLR/ UART_SPR	UART_TLR/ UART_SPR	UART_TLR	UART_TLR	UART_TLR/ UART_SPR	UART_TLR/ UART_SPR
0x020	UART_MDR1 [3:0]	UART_MDR1 [3:0]	UART_MDR1 [3:0]	UART_MDR1 [3:0]	UART_MDR1 [3:0]	UART_MDR1 [3:0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	UART_RESUME	–	UART_RESUME	–	UART_RESUME	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	–	–	–	–	–	–
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]	UART_WER [6:4]
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

(1) REGISTER_NAME(CIR) notation indicates that the register exists for other functions (IrDA or UART), but fields have different meanings for other functions (described separately in Section 24.3.6, *UART/IrDA/CIR Register Manual*).
(2) REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the CIR function.

24.3.4.8 Protocol Formatting

24.3.4.8.1 UART Mode

24.3.4.8.1.1 UART Clock Generation: Baud Rate Generation

The UART function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 24-69 shows the baud rate generator and associated controls.

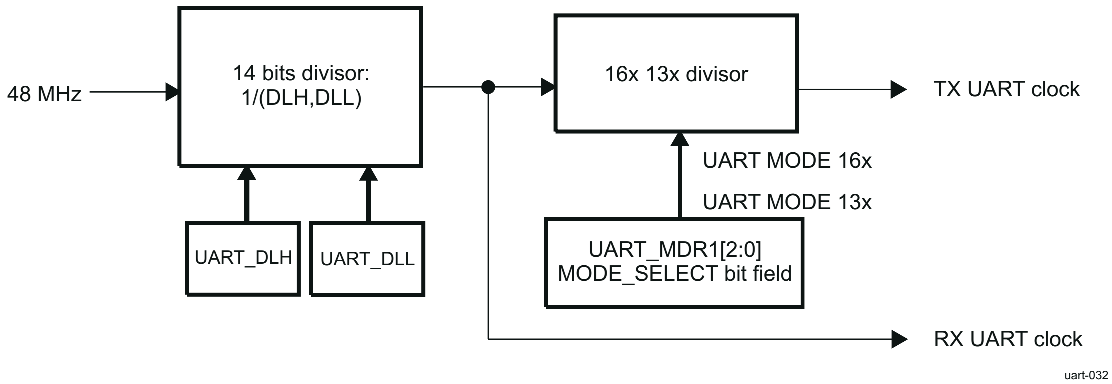


Figure 24-69. Baud Rate Generation

CAUTION

Before initializing or modifying clock parameter controls (UARTi.UART_DLH, UARTi.UART_DLL), MODE_SELECT = DISABLE (UARTi.UART_MDR1[2:0]) must be set to 0x7. Failure to observe this rule can result in unpredictable module behavior.

24.3.4.8.1.2 Choosing the Appropriate Divisor Value

Two divisor values are:

- UART 16x mode: Divisor value = Operating frequency / (16x baud rate)
- UART 13x mode: Divisor value = Operating frequency / (13x baud rate)

Table 24-110 and Table 24-111 describe the UART baud rate settings.

Table 24-110. UART Baud Rate Settings (48-MHz Clock)

Baud Rate	Baud Multiple	DLH, DLL (Decimal)	DLH, DLL (Hex)	Actual Baud Rate	Error (%)
0.3 kbps	16x	10000	0x27, 0x10	0.3 kbps	0
0.6 kbps	16x	5000	0x13, 0x88	0.6 kbps	0
1.2 kbps	16x	2500	0x09, 0xC4	1.2 kbps	0
2.4 kbps	16x	1250	0x04, 0xE2	2.4 kbps	0
4.8 kbps	16x	625	0x02, 0x71	4.8 kbps	0
9.6 kbps	16x	312	0x01, 0x38	9.6153 kbps	+0.16
14.4 kbps	16x	208	0x00, 0xD0	14.423 kbps	+0.16
19.2 kbps	16x	156	0x00, 0x9C	19.231 kbps	+0.16
28.8 kbps	16x	104	0x00, 0x68	28.846 kbps	+0.16
38.4 kbps	16x	78	0x00, 0x4E	38.462 kbps	+0.16
57.6 kbps	16x	52	0x00, 0x34	57.692 kbps	+0.16
115.2 kbps	16x	26	0x00, 0x1A	115.38 kbps	+0.16
230.4 kbps	16x	13	0x00, 0x0D	230.77 kbps	+0.16
460.8 kbps	13x	8	0x00, 0x08	461.54 kbps	+0.16
921.6 kbps	13x	4	0x00, 0x04	923.08 kbps	+0.16
1.843 Mbps	13x	2	0x00, 0x02	1.846 Mbps	+0.16
3.0 Mbps	16x	1	0x00, 0x01	3.0 Mbps	0
3.6884 Mbps	13x	1	0x00, 0x01	3.6923 Mbps	+0.16

Table 24-111. UART Baud Rate Settings (192-MHz Clock)

Baud Rate	Baud Multiple	DLH, DLL (Decimal)	DLH, DLL (Hex)	Actual Baud Rate	Error (%)
1.2 kbps	16x	10000	0x27, 0x10	1.2 kbps	0
2.4 kbps	16x	5000	0x13, 0x88	2.4 kbps	0
4.8 kbps	16x	2500	0x09, 0xC4	4.8 kbps	0
9.6 kbps	16x	1250	0x04, 0xE2	9.6 kbps	0
14.4 kbps	16x	834	0x03, 0x42	14.388 kbps	-0.08
19.2 kbps	16x	625	0x02, 0x71	19.2 kbps	0
28.8 kbps	16x	417	0x01, 0xA1	28.777 kbps	-0.08
38.4 kbps	16x	312	0x01, 0x38	38.462 kbps	+0.16
57.6 kbps	16x	208	0x00, 0xD0	57.692 kbps	+0.16
115.2 kbps	16x	104	0x00, 0x68	115.385 kbps	+0.16
230.4 kbps	16x	52	0x00, 0x34	230.769 kbps	+0.16
460.8 kbps	16x	26	0x00, 0x1A	461.538 kbps	+0.16
921.6 kbps	16x	13	0x00, 0x0D	923.077 kbps	+0.16
1.8432 Mbps	13x	8	0x00, 0x08	1.846154 Mbps	+0.16
3.0 Mbps	16x	4	0x00, 0x04	3.0 Mbps	0
3.6864 Mbps	13x	4	0x00, 0x04	3.692308 Mbps	+0.16
7.3728 Mbps	13x	2	0x00, 0x02	7.384615 Mbps	+0.16
12.0 Mbps	16x	1	0x00, 0x01	12.0 Mbps	0

24.3.4.8.1.3 UART Data Formatting

The UART can use hardware flow control to manage transmission and reception. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the RTS output and CTS input signals.

The UART is enhanced with the autobauding function. In control mode, autobauding lets the speed, the number of bits per character, and the parity selected be set automatically.

24.3.4.8.1.3.1 Frame Formatting

When autobauding is not used, frame format attributes must be defined in the UARTi.UART_LCR register.

Character length is specified using the UARTi.UART_LCR[1:0] CHAR_LENGTH bit field.

The number of stop-bits is specified using the UARTi.UART_LCR[2] NB_STOP bit.

The parity bit is programmed using the UARTi.UART_LCR[5:3] PARITY_EN, PARITY_TYPE_1, and PARITY_TYPE_2 bit fields (see Table 24-112).

Table 24-112. UART Parity Bit Encoding

PARITY_EN	PARITY_TYPE_1	PARITY_TYPE_2	Parity
0	N/A	N/A	No parity
1	0	0	Odd parity
1	1	0	Even parity
1	0	1	Forced 1
1	1	1	Forced 0

24.3.4.8.1.3.2 Hardware Flow Control

Hardware flow control is composed of auto-CTS and auto-RTS. Auto-CTS and auto-RTS can be enabled and disabled independently by programming the UARTi.UART_EFR[7:6] AUTO_CTS_EN and AUTO_RTS_EN bit fields, respectively.

With auto-CTS, uarti_ctsn must be active before the module can transmit data.

Auto-RTS activates the uarti_rtsn output only when there is enough room in the RX FIFO to receive data. It deactivates the uarti_rtsn output when the RX FIFO is sufficiently full. The HALT and RESTORE trigger levels in the UARTi.UART_TCR register determine the levels at which uarti_rtsn is activated and deactivated.

If auto-CTS and auto-RTS are enabled, data transmission does not occur unless the RX FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If auto-CTS and auto-RTS are not enabled, overrun errors occur if the transmit data rate exceeds the RX FIFO latency.

- Auto-RTS:

  Auto-RTS data flow control originates in the receiver block. The RX FIFO trigger levels used in auto-RTS are stored in the UARTi.UART_TCR register. uarti_rtsn is active if the RX FIFO level is below the HALT trigger level in the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field. When the RX FIFO HALT trigger level is reached, uarti_rtsn is deasserted. The sending device (for example, another UART) can send an additional byte after the trigger level is reached because it may not recognize the deassertion of RTS until it begins sending the additional byte.

  uarti_rtsn is automatically reasserted when the RX FIFO reaches the RESUME trigger level programmed by the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field. This reassertion requests the sending device to resume transmission.

  In this case, uarti_rtsn is an active-low signal.
- Auto-CTS:

  The transmitter circuitry checks uarti_ctsn before sending the next data byte. When uarti_ctsn is active, the transmitter sends the next byte. To stop the transmitter from sending the next byte, uarti_ctsn must be deasserted before the middle of the last stop-bit currently sent.

  The auto-CTS function reduces interrupts to the host system. When auto-CTS flow control is enabled, the uarti_ctsn state changes do not have to trigger host interrupts because the device automatically controls its own transmitter. Without auto-CTS, the transmitter sends any data present in the transmit FIFO, and a receiver overrun error can result.

  In this case, uarti_ctsn is an active-low signal.

24.3.4.8.1.3.3 Software Flow Control

Software flow control is enabled through the enhanced feature register (UARTi.UART_EFR) and the modem control register (UARTi.UART_MCR). Different combinations of software flow control can be enabled by setting different combinations of the UARTi.UART_EFR[3:0] bit field (see Table 24-113).

Two other enhanced features relate to software flow control:

- XON-any function (UARTi.UART_MCR[5]): Operation resumes after receiving any character after the XOFF character is recognized. If special character detect is enabled and special character is received after XOFF1, it does not resume transmission. The special character is stored in the RX FIFO.

  Note

  The XON-any character is written into the RX FIFO even if it is a software flow character.
- Special character (UARTi.UART_EFR[5]): Incoming data is compared to XOFF2. When the special character is detected, the XOFF interrupt (UARTi.UART_IIR[4]) is set, but it does not halt transmission. The XOFF interrupt is cleared by a read of UARTi.UART_IIR. The special character is transferred to the RX FIFO. Special character does not work with XON2, XOFF2, or sequential XOFFs.

Table 24-113. UART_EFR[3:0] Software Flow Control Options

Bit 3	Bit 2	Bit 1	Bit 0	TX, RX Software Flow Controls
0	0	X	X	No transmit flow control
1	0	X	X	Transmit XON1, XOFF1
0	1	X	X	Transmit XON2, XOFF2

Table 24-113. UART_EFR[3:0] Software Flow Control Options (continued)

Bit 3	Bit 2	Bit 1	Bit 0	TX, RX Software Flow Controls
1	1	X	X	Transmit XON1, XON2: XOFF1, XOFF2[(1)]
X	X	0	0	No receive flow control
X	X	1	0	Receiver compares XON1, XOFF1
X	X	0	1	Receiver compares XON2, XOFF2
X	X	1	1	Receiver compares XON1, XON2: XOFF1, XOFF2[(1)]

(1) In these cases, the XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2.
XON1 is defined in the UARTi.UART_XON1_ADDR1[7:0] XON_WORD1 bit field. XON2 is defined in the UARTi.UART_XON2_ADDR2[7:0] XON_WORD2 bit field.
XOFF1 is defined in the UARTi.UART_XOFF1[7:0] XOFF_WORD1 bit field. XOFF2 is defined in the UARTi.UART_XOFF2[7:0] XOFF_WORD2 bit field.

24.3.4.8.1.3.3.1 Receive (RX)

When software flow control operation is enabled, the UART compares incoming data with XOFF1/2 programmed characters (in certain cases, XOFF1 and XOFF2 must be received sequentially). When the correct XOFF characters are received, transmission stops after transmission of the current character completes. Detection of XOFF also sets the UARTi.UART_IIR[4] bit (if enabled by UARTi.UART_IER[5]) and causes the interrupt line to go low.

To resume transmission, an XON1/2 character must be received (in certain cases, XON1 and XON2 must be received sequentially). When the correct XON characters are received, the UARTi.UART_IIR[4] bit is cleared and the XOFF interrupt disappears.

Note

When a parity, framing, or break error occurs while receiving a software flow control character, this character is treated as normal data and is written to the RX FIFO.

When XON-any and special character detect are disabled and software flow control is enabled, no valid XON or XOFF characters are written to the RX FIFO. For example, when UARTi.UART_EFR[1:0] = 0x2, if XON1 and XOFF1 characters are received, they are not written to the RX FIFO.

When pairs of software flow characters are programmed to be received sequentially (UARTi.UART_EFR[1:0] = 0x3), the software flow characters are not written to the RX FIFO if they are received sequentially. However, received XON1/XOFF1 characters must be written to the RX FIFO if the subsequent character is not XON2/XOFF2.

24.3.4.8.1.3.3.2 Transmit (TX)

Two XOFF1 characters are transmitted when the RX FIFO passes the trigger level programmed by UARTi.UART_TCR[3:0]. As soon as the RX FIFO reaches the trigger level programmed by UARTi.UART_TCR[7:4], two XON1 characters are sent, so the data transfer recovers.

Note

If software flow control is disabled after an XOFF character is sent, the module transmits XON characters automatically to enable normal transmission.

The transmission of XOFF(s)/XON(s) follows the same protocol as transmission of an ordinary byte from the TX FIFO. This means that even if the word length is 5, 6, or 7 characters, the 5, 6, or 7 LSBs of XOFF1/2 and XON1/2 are transmitted. The 5, 6, or 7 bits of a character are seldom transmitted, but this function is included to maintain compatibility with earlier designs.

It is assumed that software flow control and hardware flow control are never enabled simultaneously.

24.3.4.8.1.3.4 Autobauding Modes

In autobauding mode, the UART can extract transfer characteristics (speed, length, and parity) from an "at" (AT) command (ASCII code). These characteristics are used to receive data after an AT and to send data.

The following AT commands are valid:

AT	DATA	<CR>
at	DATA	<CR>
A/		
a/		

A line break during the acquisition of the sequence AT is not recognized, and an echo function is not implemented in hardware.

A/ and a/ are not used to extract characteristics, but they must be recognized because of their special meaning. A/ or a/ is used to instruct the software to repeat the last received AT command; therefore, an a/ always follows an AT, and transfer characteristics are not expected to change between an AT and an a/.

When a valid AT is received, AT and all subsequent data, including the final <CR> (0x0D), are saved to the RX FIFO. The autobaud state-machine waits for the next valid AT command. If an a/ (A/) is received, the a/ (A/) is saved in the RX FIFO and the state-machine waits for the next valid AT command.

On the first successful detection of the baud rate, the UART activates an interrupt to signify that the AT (upper or lower case) sequence is detected. The UARTi.UART_UASR register reflects the correct settings for the baud rate detected. Interrupt activity can continue in this fashion when a subsequent character is received. Therefore, it is recommended that the software enable the RHR interrupt when using the autobaud mode.

The following settings are detected in autobaud mode with a module clock of 48 MHz:

- Speed:
  - 115.2K baud
  - 57.6K baud
  - 38.4K baud
  - 28.8K baud
  - 19.2K baud
  - 14.4K baud
  - 9.6K baud
  - 4.8K baud
  - 2.4K baud
  - 1.2K baud
- Length: 7 or 8 bits
- Parity: Odd, even, or space

Note

The combination of 7-bit character plus space parity is not supported.

Autobauding mode is selected when the UARTi.UART_MDR1[2:0] MODE_SELECT bit field is set to 0x2. In UART autobauding mode, UARTi.UART_DLL, UARTi.UART_DLH, and UARTi.UART_LCR[5:0] bit field settings are not used; instead, UASR is updated with the configuration detected by the autobauding logic.

UASR Autobauding Status Register Use

This register is used to set up transmission according to the characteristics of the previous reception instead of the UARTi.UART_LCR, UARTi.UART_DLL, and UARTi.UART_DLH registers when the UART is in autobauding mode.

To reset the autobauding hardware (to start a new AT detection) or to set the UART in standard mode (no autobaud), the UARTi.UART_MDR1[2:0] MODE_ SELECT bit field must be set to reset state (0x7) and then to the UART in autobauding mode (0x2) or to the UART in standard mode (0x0).

Use limitation:

- Only 7- and 8-bit characters (5- and 6-bit not supported)
- 7-bit character with space parity not supported
- Baud rate between 1200 and 115,200 bps (10 possibilities)

24.3.4.8.1.3.5 Error Detection

When the UARTi.UART_LSR register is read, the UARTi.UART_LSR[4:2] bit field reflects the error bits (BI: break condition, FE: framing error, PE: parity error) of the character at the top of the RX FIFO (the next character to be read). Therefore, reading the UARTi.UART_LSR register and then reading the UARTi.UART_RHR register identifies errors in a character.

Reading the UARTi.UART_RHR register updates the BI, FE, and PE bits (see Table 24-96 for the UART mode interrupts).

The UARTi.UART_LSR[7] RX_FIFO_STS bit is set when there is an error in the RX FIFO and is cleared only when no errors remain in the RX FIFO.

Note

Reading the UARTi.UART_LSR register does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the UARTi.UART_RHR register.

Reading the UARTi.UART_LSR register clears the OE bit if it is set (see Table 24-96 for the UART mode interrupts).

24.3.4.8.1.3.6 Overrun During Receive

Overrun during receive occurs if the RX state-machine tries to write data into the RX FIFO when it is already full. When overrun occurs, the device interrupts the MPUwith the UARTi.UART_IIR[5:1] IT_TYPE bit field set to 0x3 (receiver line status error) and discards the remaining portion of the frame.

Overrun also causes an internal flag to be set, which disables further reception. Before the next frame can be received, the MPU must:

- Reset the RX FIFO.
- Read the UARTi.UART_RESUME register, which clears the internal flag.

24.3.4.8.1.3.7 Time-Out and Break Conditions

24.3.4.8.1.3.7.1 Time-Out Counter

An RX idle condition is detected when the receiver line (uarti_rxd) is high for a time that equals 4x the programmed word length + 12 bits. uarti_rxd is sampled midway through each bit.

For sleep mode, the counter is reset when there is activity on uarti_rxd.

For the time-out interrupt, the counter counts only when there is data in the RX FIFO, and the count is reset when there is activity on uarti_rxd or when the UARTi.UART_RHR register is read.

24.3.4.8.1.3.7.2 Break Condition

When a break condition occurs, uarti_txd is pulled low. A break condition is activated by setting the UARTi.UART_LCR[6] BREAK_EN bit. The break condition is not aligned on word stream (a break condition can occur in the middle of a character). The only way to send a break condition on a full character is:

1. Reset the TX FIFO (if enabled).
2. Wait for the transmit shift register to empty (the UARTi.UART_LSR[6] TX_SR_E bit is set to 1).
3. Take a guard time according to stop-bit definition.

4. Set the BREAK_EN bit to 1.

The break condition is asserted while the BREAK_EN bit is set to 1.

The time-out counter and break condition apply only to UART modem operation and not to IrDA/CIR mode operation.

24.3.4.8.2 IrDA Mode (UART3 Only)

24.3.4.8.2.1 IrDA Clock Generation: Baud Generator

The IrDA function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 24-70 shows the baud rate generator and associated controls.

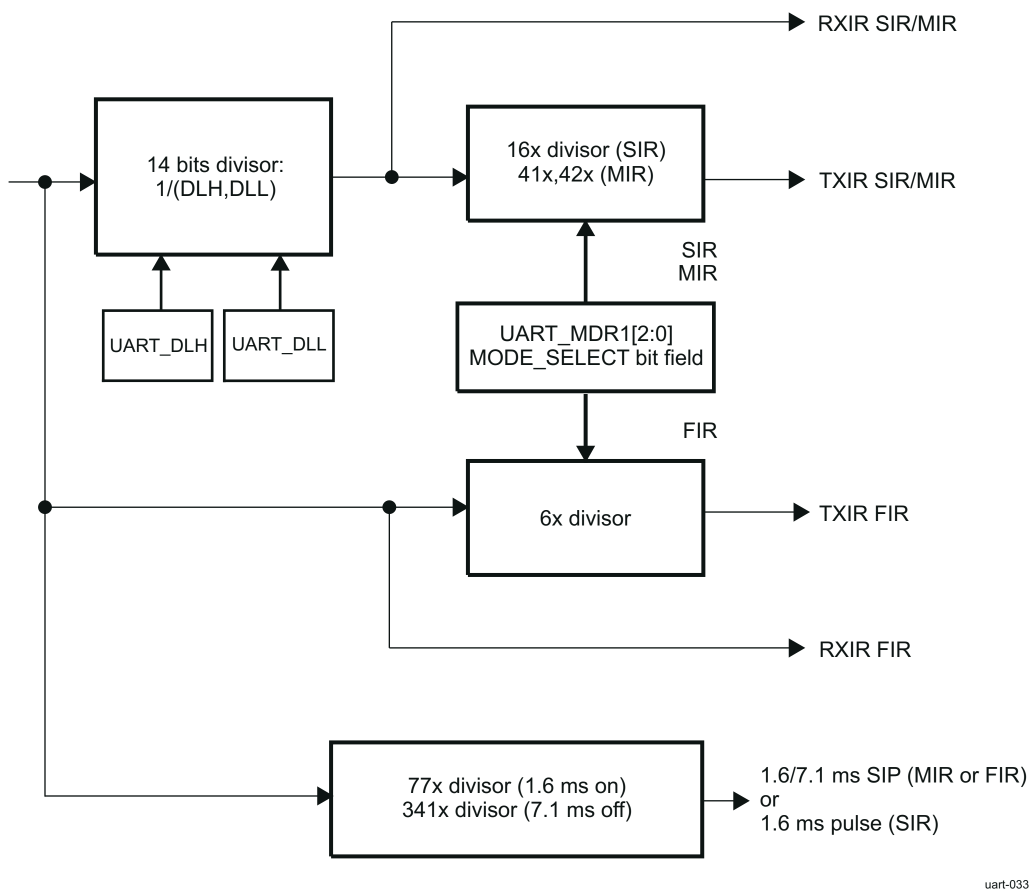


Figure 24-70. Baud Rate Generator

CAUTION

Before initializing or modifying clock parameter controls (UARTi.UART_DLH, UARTi.UART_DLL), MODE_SELECT=DISABLE (UARTi.UART_MDR1[2:0]) must be set to 0x7). Failure to observe this rule can result in unpredictable module behavior.

24.3.4.8.2.2 Choosing the Appropriate Divisor Value

Three divisor values are:

- SIR mode: Divisor value = Operating frequency/(16x baud rate)
- MIR mode: Divisor value = Operating frequency/(41x/42x baud rate)
- FIR mode: Divisor value = None

Table 24-114 lists the IrDA baud rate settings.

Table 24-114. IrDA Baud Rate Settings

Baud Rate	IR Mode	Baud Multiple	Encoding	DLH, DLL (Decimal)	Actual Baud Rate	Error (%)	Source Jitter (%)	Pulse Duration
2.4 kbps	SIR	16x	3/16	1250	2.4 kbps	0	0	78.1 µs
9.6 kbps	SIR	16x	3/16	312	9.6153 kbps	+0.16	0	19.5 µs
19.2 kbps	SIR	16x	3/16	156	19.231 kbps	+0.16	0	9.75 µs
38.4 kbps	SIR	16x	3/16	78	38.462 kbps	+0.16	0	4.87 µs
57.6 kbps	SIR	16x	3/16	52	57.692 kbps	+0.16	0	3.25 µs
115.2 kbps	SIR	16x	3/16	26	115.38 kbps	+0.16	0	1.62 µs
0.576 Mbps	MIR	41x/42x	1/4	2	0.5756 Mbps[1]	0	+1.63/-0.80	416 ns
1.152 Mbps	MIR	41x/42x	1/4	1	1.1511 Mbps[1]	0	+1.63/-0.80	208 ns
4 Mbps	FIR	6x	4 PPM	–	4 Mbps	0	0	125 ns

(1) Average value

Note

Baud rate error and source jitter table values do not include 48-MHz reference clock error and jitter.

24.3.4.8.2.3 IrDA Data Formatting

The methods described in this section apply to all IrDA modes (SIR, MIR, and FIR).

24.3.4.8.2.3.1 IR RX Polarity Control

The UART3.UART_MDR2[6] IRRXINVERT bit provides the flexibility to invert the uart3_rxd pin in the UART to ensure that the protocol at the output of the transceiver has the same polarity at module level. By default, the uart3_rxd pin is inverted because most transceivers invert the IR receive pin.

24.3.4.8.2.3.2 IrDA Reception Control

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware.

Operation of the uart3_rxd input can be disabled by the UART3.UART_ACREG[5] DIS_IR_RX bit.

24.3.4.8.2.3.3 IR Address Checking

In all IR modes, when address checking is enabled, only frames intended for the device are written to the RX FIFO. This restriction avoids receiving frames not meant for this device in a multipoint infrared environment. It is possible to program two frame addresses that the UART IrDA receives, with the UART3.UART_XON1_ADDR1[7:0] XON_WORD1 and UART3.UART_XON2_ADDR2[7:0] XON_WORD2 bit fields.

Setting the UART_EFR[0] bit to 1 selects address1 checking. Setting the UART_EFR[1] bit to 1 selects address2 checking. Setting the UART_EFR[1:0] bit field to 0 disables all address checking operations. If both bits are set, the incoming frame is checked for private and public addresses.

If address checking is disabled, all received frames write to the RX FIFO.

24.3.4.8.2.3.4 Frame Closing

A transmission frame can be terminated in two ways:

- Frame-length method: Set the UART3.UART_MDR1[7] FRAME_END_MODE bit to 0. The MPU writes the value of the frame length to the UART3.UART_TXFLH and UART3.UART_TXFLL registers. The device automatically attaches end flags to the frame when the number of bytes transmitted equals the value of the frame length.

- Set-EOT bit method: Set the FRAME_END_MODE bit to 1. The MPU writes 1 to the UART3.UART_ACREG[0] EOT bit just before it writes the last byte to the TX FIFO. When the MPU writes the last byte to the TX FIFO, the device internally sets the tag bit for that character in the TX FIFO. As the TX state-machine reads data from the TX FIFO, it uses this tag-bit information to attach end flags and correctly terminate the frame.

24.3.4.8.2.3.5 Store and Controlled Transmission

In store and controlled transmission (SCT) mode, the MPU starts writing data to the TX FIFO. Then, after writing a part of a frame (for a bigger frame) or an entire frame (a small frame; that is, a supervisory frame), the MPU writes 1 to the UART3.UART_ACREG[2] SCTX_EN bit (deferred TX start) to start transmission.

SCT mode is enabled by setting the UART3.UART_MDR1[5] SCT bit to 1. This transmission method differs from normal mode, in which data transmission starts immediately after data is written to the TX FIFO. SCT mode is useful for sending short frames without TX underrun.

24.3.4.8.2.3.6 Error Detection

When the UART3.UART_LSR register is read, the UART3.UART_LSR[4:2] bit field reflects the error bits [FL, CRC, ABORT] of the frame at the top of the STATUS FIFO (the next frame status to be read).

The error is triggered by an interrupt (for IrDA mode interrupts, see Table 24-97). The STATUS FIFO must be read until empty (a maximum of eight reads is required).

24.3.4.8.2.3.7 Underrun During Transmission

Underrun during transmission occurs when the TX FIFO is empty before the end of the frame is transmitted. When underrun occurs, the device closes the frame with end flags but attaches an incorrect CRC value. The receiving device detects a CRC error and discards the frame; it can then ask for a retransmission.

Underrun also causes an internal flag to be set, which disables additional transmissions. Before the next frame can be transmitted, the MPU must:

- Reset the TX FIFO.
- Read the UART3.UART_RESUME register, which clears the internal flag.

This function can be disabled by the UART3.UART_ACREG[4] DIS_TX_UNDERRUN bit, compensated by the extension of the stop-bit in transmission if the TX FIFO is empty.

24.3.4.8.2.3.8 Overrun During Receive

Overrun during receive for the IrDA mode has the same function as that for the UART mode (see Section 24.3.4.8.1.3.6, *Overrun During Receive*).

24.3.4.8.2.3.9 Status FIFO

In IrDA modes, a status FIFO records the received frame status. When a complete frame is received, the length of the frame and the error bits associated with the frame are written to the status FIFO.

Reading the UART3.UART_SFREGH[3:0] MSB and UART3.UART_SFREGL[3:0] (LSB) bit fields obtains the frame length. The frame error status is read in the UART3.UART_SFLSR register. Reading the UART3.UART_SFLSR register increments the status FIFO read pointer. Because the status FIFO is eight entries deep, it can hold the status of eight frames.

The MPU uses the frame-length information to locate the frame boundary in the received frame data. The MPU can screen bad frames using the error status information and can later request the sender to resend only the bad frames.

This status FIFO can be used effectively in DMA mode because the MPUmust be interrupted only when the programmed status FIFO trigger level is reached, not each time a frame is received.

24.3.4.8.2.4 SIR Mode Data Formatting

This section provides specific instructions for SIR mode programming.

24.3.4.8.2.4.1 Abort Sequence

The transmitter can prematurely close a frame (abort) by sending the sequence 0x7DC1. The abort pattern closes the frame without a CRC field or an ending flag.

A transmission frame can be aborted by setting the UART3.UART_ACREG[1] ABORT_EN bit to 1. When this bit is set to 1, 0x7D and 0xC1 are transmitted and the frame is not terminated with CRC or stop flags.

When a 0x7D character followed immediately by a 0xC1 character is received without transparency, the receiver treats a frame as an aborted frame.

> **CAUTION**
>
> When the TX FIFO is not empty and the UART3.UART_MDR1[5] SCT bit is set to 1, the UART IrDA starts a new transfer with data of a previous frame when the aborted frame is sent. Therefore, the TX FIFO must be reset before sending an aborted frame.

24.3.4.8.2.4.2 Pulse Shaping

SIR mode supports the 3/16 or the 1.6-μs pulse duration methods. The UART3.UART_ACREG[7] PULSE_TYPE bit selects the pulse width method in the transmit mode.

24.3.4.8.2.4.3 SIR Free Format Programming

The SIR FF mode is selected by setting the module in the UART mode (UART3.UART_MDR1[2:0] MODE_SELECT = 0x0) and the UART3.UART_MDR2[3] PULSE bit to 1 to allow pulse shaping.

Because the bit format stays the same, some UART mode configuration registers must be set at specific values:

- UART3.UART_LCR[1:0] CHAR_LENGTH bit field = 0x3 (8 data bits)
- UART3.UART_LCR[2] NB_STOP bit = 0x0 (1 stop-bit)
- UART3.UART_LCR[3] PARITY_EN bit = 0x0 (no parity)

The UART mode interrupts are used for the SIR FF mode, but many are not relevant (XOFF, RTS, CTS, modem status register, etc.).

24.3.4.8.2.5 MIR and FIR Mode Data Formatting

This section describes common instructions for FIR and MIR mode programming.

At the end of a frame reception, the MPU reads the line status register (UART3.UART_LSR) to detect errors in the received frame.

When the UART3.UART_MDR1[6] SIP_MODE bit is set to 1, the TX state-machine always sends one SIP at the end of a transmission frame. However, when the SIP_MODE bit is set to 0, SIP transmission depends on the UART3.UART_ACREG[3] SEND_SIP bit.

The MPU can set the SEND_SIP bit at least once every 500 ms. The advantage of this approach over the default approach is that the TX state-machine does not have to send the SIP at the end of each frame, thus reducing the overhead required.

24.3.4.8.3 CIR Mode (UART3 Only)

24.3.4.8.3.1 CIR Mode Clock Generation

Depending on the encoding method (variable pulse distance/biphase), the MPU must develop a data structure that combines 1 and 0 with a *t* period to encode the complete frame to transmit. This can then be transmitted to the infrared output with a modulation method, as shown in Figure 24-71.

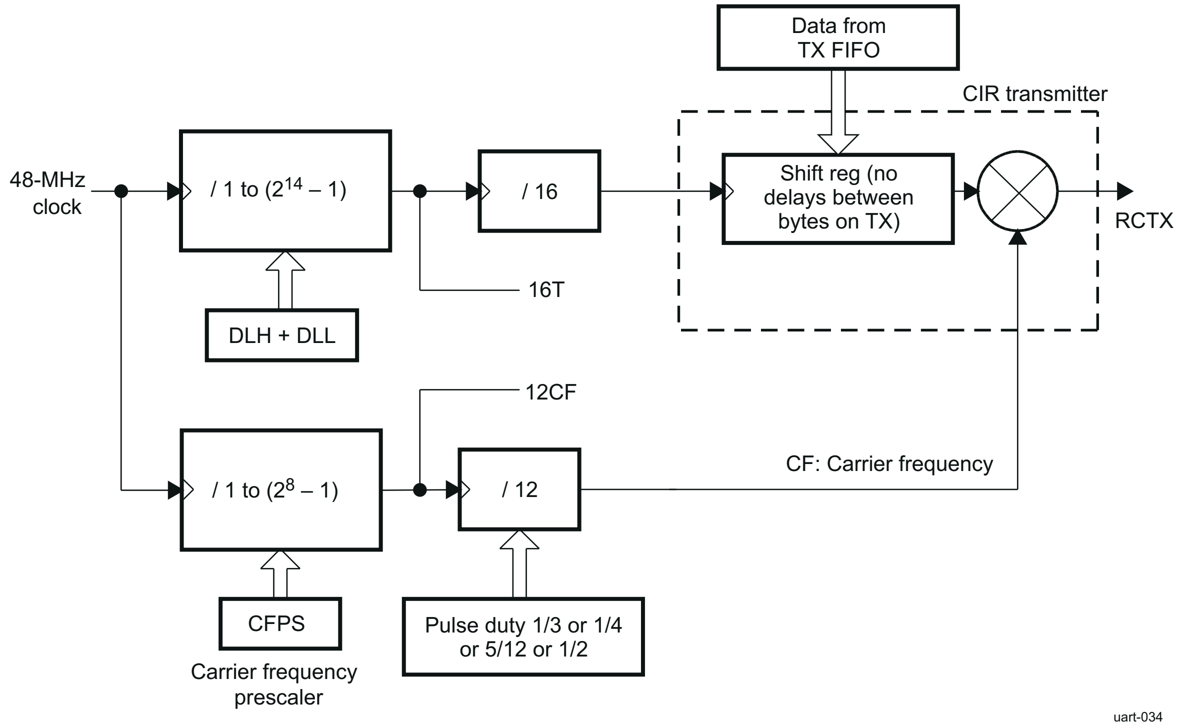


Figure 24-71. CIR Mode Block Components

Based on the requested modulation frequency, the UART3.UART_CFPS register must be set with the correct dividing value to provide an accurate pulse frequency:

Dividing value = (FCLK / 12) / MODfreq

Where:

FCLK = System clock frequency (48 MHz)

12 = Real value of baud multiple

MODfreq = Effective frequency of the modulation (MHz)

Example: For a targeted modulation frequency of 36 kHz, the value of CFPS must be set to 0x7 (decimal), which provides a modulation frequency of 36.04 kHz.

Note

The UART3.UART_CFPS register starts with a reset value of 105 (decimal), which translates to a frequency of 38.1 kHz.

The duty cycle of these pulses is user-defined by the pulse duty register bits in the UART3.UART_MDR2 register. Table 24-115 shows the duty cycle.

Table 24-115. Duty Cycle

UART_MDR2[5:4]	Duty Cycle (High-Level)
00	1/4
01	1/3
10	5/12
11	1/2

24.3.4.8.3.2 CIR Data Formatting

The methods described in this section apply to all CIR modes.

24.3.4.8.3.2.1 IR RX Polarity Control

The IR RX polarity control for CIR mode has the same function as that for IrDA mode (see Section 24.3.4.8.2.3.1, *IR RX Polarity Control*).

24.3.4.8.3.2.2 CIR Transmission

In transmission, the MPU software must exercise an element of real-time control to transmit data packets, each of which must be emitted at a constant delay from the start-bits of each individual packet. Thus, when sending a series of packets, the packet-to-packet delay must respect a specific delay. Two methods can be used to control this delay:

- Filling the TX FIFO with a number of zero bits that are transmitted with a *t* period
- Using an external system timer to control the delay between each start-of-frame or between the end of a frame and the start of the next one. This can be performed by:
  - Controlling the start of the frame using the UART3.UART_MDR1[5] SCT bit and the UART3.UART_ACREG[2] SCTX_EN bit, depending on the timer status
  - Using the UART3.UART_IIR[5] TX_STATUS_IT interrupt bit to preload the next frame in the TX FIFO and to control the start of the timer (in case of control delay between the end of a frame and the start of the next frame)

24.3.5 UART/IrDA/CIR Basic Programming Model

This section describes the procedure for operating the UART with FIFO and DMA or interrupts. This three-part procedure ensures the quick start of the UART. It does not cover every UART feature.

The first programming model covers software reset of the UART. The second programming model describes FIFO and DMA configuration. The last programming model describes protocol, baud rate, and interrupt configuration.

Note

Each programming model can be used independently of the other two; for instance, reconfiguring the FIFOs and DMA settings only.

Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see Section 24.3.4.7.1, *Register Access Modes*.

24.3.5.1 Global Initialization

24.3.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the UART/IrDA/CIR module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the UART/IrDA/CIR.

For more information, see Section 24.3.3, *UART/IrDA/CIR Integration*.

Table 24-116. Global Initialization of Surrounding Modules for UART/IrDA/CIR

Surrounding Modules	Comments
PRCM	UART functional and interface clocks must be enabled. For more information, see *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing must be set in the control module. For more information, see *Control Module*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*
Interrupt controllers	Device INTCs must be configured to enable the interrupt request generation. For information about enabling interrupts, see *Interrupt Controllers*.
DMA_CROSSBAR	DMA_CROSSBAR configuration must be done to allow module DREQs to be mapped to certain device DMA line. For more information see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
DMA controllers	DMA controllers configuration must be done to enable the module DMA channel request. See Chapter 16, *DMA Controllers*
Interconnects	For information about the L4 interconnect configuration, see *Interconnect*.

24.3.5.1.2 UART/IrDA/CIR Module Global Initialization

The procedure in Table 24-117 can be used to initialize UART when performing software reset.

Table 24-117. UART/IrDA/CIR Global Initialization

Step	Register/Bit Field/Programming Model	Value
Perform a software reset.	UART_SYSC[1] SOFTRESET	1
Wait until reset is finished.	UART_SYSS[0] RESETDONE	=1

24.3.5.2 Mode selection

Table 24-118 describes how to set different register access mode.

Table 24-118. Configure Register Access Mode

Step	Register/Bit Field/Programming Model	Value
Set the register access mode A	UART_LCR[7] DIV_EN	1
	UART_LCR[7:0]	≠0xBF
Set the register access mode B	UART_LCR[7:0]	0xBF
Set the operational mode	UART_LCR[7] DIV_EN	0

24.3.5.3 Submode selection

This section describes how to set different register access submode.

Table 24-119. Configure Register Access Submode TCR_TLR

Step	Register/Bit Field/Programming Model	Value
Configure the submode TCR_TLR:		
Configure mode B	see Table 24-118	
Enable writing to register bits UART_MCR[6:5]	UART_EFR[4] ENHANCED_EN	1
Configure mode A	see Table 24-118	0x1
Set the submode TCR_TLR	UART_MCR[6] TCR_TLR	1

Table 24-120. Configure Register Access Submode MSR_SPR

Step	Register/Bit Field/Programming Model	Value
First option to configure the submode MSR_SPR:		
Configure mode B	see Table 24-118	
Set the submode MSR_SPR	UART_EFR[4] ENHANCED_EN	0
Second option to configure the submode MSR_SPR:		
Configure mode B	see Table 24-118	
Enable writing to register bits UART_MCR[6:5]	UART_EFR[4] ENHANCED_EN	1
Set the submode MSR_SPR	UART_MCR[6] TCR_TLR	0

Table 24-121. Configure Register Access Submode XOFF

Step	Register/Bit Field/Programming Model	Value
Configure of the XOFF:		
Configure B	see Table 24-118	
Set the submode XOFF	UART_EFR[4] ENHANCED_EN	0

24.3.5.4 Load FIFO trigger and DMA mode settings

24.3.5.4.1 DMA mode Settings

To enable and configure DMA mode, perform the following steps:

Table 24-122. DMA Mode Settings

Step	Register/Bit Field/Programming Model	Value
Set the option of DMA mode configuration	UART_SCR[0] DMA_MODE_CTL	-
IF Configure DMA mode 0 and 1	UART_SCR[0] DMA_MODE_CTL	=0
Select the DMA mode, for more information see Section 24.3.4.6.4	UART_FCR[3] DMA_MODE	-
IF Configure DMA mode from 0 to 3	UART_SCR[0] DMA_MODE_CTL	=1
Select the DMA mode, for more information see Section 24.3.4.6.4	UART_SCR[2:1] DMA_MODE_2	-

24.3.5.4.2 FIFO Trigger Settings

In this section is described configuration and settings of FIFO trigger level, which enable DMA and interrupt generation.

Table 24-123. Load FIFO Triggers Defined by the FCR

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 24-119	0x-
Set the desire RX FIFO trigger level	UART_FCR[5:4] TX_FIFO_TRIG	0x-
Set the desire TX FIFO trigger level	UART_FCR[7:6] RX_FIFO_TRIG	0x-

Table 24-124. Load FIFO Triggers Defined by the TLR

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 24-119	0x-
Set the desire RX FIFO trigger level	UART_TLR[7:4] RX_FIFO_TRIG_DMA	0x-
Set the desire TX FIFO trigger level	UART_TLR[3:0] TX_FIFO_TRIG_DMA	0x-

Table 24-125. Load FIFO Triggers Defined by the Concatenated Value

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 24-119	0x-
Set the register bit	UART_SCR[7] RX_TRIG_GRANU1	1
Set the desire RX FIFO trigger level	UART_TLR[7:4] RX_FIFO_TRIG_DMA UART_FCR[7:6] RX_FIFO_TRIG	0x-
Set the register bit	UART_SCR[6] TX_TRIG_GRANU1	1
Set the desire TX FIFO trigger level	UART_TLR[3:0] TX_FIFO_TRIG_DMA UART_FCR[5:4] TX_FIFO_TRIG	0x-

24.3.5.5 Protocol, Baud rate and interrupt settings

24.3.5.5.1 Baud rate settings

Table 24-126. Baud Rate Settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Switch to register configuration mode B	see Table 24-118	
Enable access to UART_IER[7:4]	UART_EFR[4] ENHANCED_EN	1
Switch register operational mode	see Table 24-118	
Disable sleep mode	UART_IER[4] SLEEP_MODE	0
Switch to register configuration mode A or B	see Table 24-118	
Set the appropriate divisor value	UART_DLL[7:0] CLOCK_LSB UART_DLH[5:0] CLOCK_MSB	0x-

24.3.5.5.2 Interrupt settings

Table 24-127. Interrupt Settings

Step	Register/Bit Field/Programming Model	Value
Switch to register configuration mode B	see Table 24-118	0x7
Enable access to UART_IER[7:4]	UART_EFR[4] ENHANCED_EN	1
Switch register operational mode	see Table 24-118	

Table 24-127. Interrupt Settings (continued)

Step	Register/Bit Field/Programming Model	Value
Set the desired interrupt configuration (0: Disable the interrupt; 1: Enable the interrupt)	UART_IER[7] CTS_IT	0x-
	UART_IER[6] RTS_IT	
	UART_IER[5] XOFF_IT	
	UART_IER[4] SLEEP_MODE	
	UART_IER[3] MODEM_STS_IT	
	UART_IER[2] LINE_STS_IT	
	UART_IER[1] THR_IT	
	UART_IER[0] RHR_IT	

24.3.5.5.3 Protocol settings

Load the desired protocol formatting (parity, stop-bit, character length) and switch to register operational mode.

Table 24-128. Protocol Settings

Step	Register/Bit Field/Programming Model	Value
Load desired protocol formatting, see Section 24.3.4.8.1.3.1 , *Frame Formatting*	UART_LCR[5] PARITY_TYPE_2	0x-
	UART_LCR[4] PARITY_TYPE_1	
	UART_LCR[3] PARITY_EN	
	UART_LCR[2] NB_STOP	
	UART_LCR[1:0] PARITY_LENGTH	
Switch to register operational mode	UART_LCR[7] DIV_EN	0
	UART_LCR[6] BREAK_EN	

24.3.5.5.4 UART/IrDA(SIR/MIR/FIR)/CIR

Table 24-129. UART/IrDA/CIR Mode Selection

Step	Register/Bit Field/Programming Model	Value
Load the desired UART mode, see Section 24.3.4.7.2, *UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection*	UART_MDR1[2:0] MODE_SELECT	0x-

24.3.5.6 Hardware and Software Flow Control Configuration

This section describes the programming steps to enable and configure hardware and software flow control. Hardware and software flow control cannot be used at the same time.

24.3.5.6.1 Hardware Flow Control Configuration

Table 24-130. Hardware Flow Control Configuration

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 24-119	0x7
Load the start and halt trigger value.	UART_TCR[7:4] AUTO_RTS_START	0x-
	UART_TCR[3:0] AUTO_RTS_HALT	
Enable or disable receive and transmit hardware flow control mode.	UART_EFR[7] AUTO_CTS_EN	0x-
	UART_EFR[6] AUTO_RTS_EN	

24.3.5.6.2 Software Flow Control Configuration

Table 24-131. Software Flow Control Configuration

Step	Register/Bit Field/Programming Model	Value
Set the register access submode XOFF	see Table 24-121	

Table 24-131. Software Flow Control Configuration (continued)

Step	Register/Bit Field/Programming Model	Value
Load the software control characters	UART_XON1_ADDR1[7:0] XON_WORD1	0x-
	UART_XON2_ADDR2[7:0] XON_WORD2	
	UART_XOFF1[7:0] XOFF_WORD1	
	UART_XOFF2[7:0] XOFF_WORD2	
Set the register access submode TCR_TLR	see Table 24-119	
Enable or disable XON any function (0: Disable; 1: Enable).	UART_MCR[5] XON_EN	--
Load start and halt trigger value for software flow control	UART_TCR[7:4] AUTO_RTS_START	0x-
	UART_TCR[3:0] AUTO_RTS_HALT	
Enable or disable special character function (0: Disable; 1: Enable)	UART_EFR[5] SPEC_CHAR	0x-
Set the software flow control mode	UART_EFR[3:0] SW_FLOW_CONTROL	0x-

24.3.5.7 IrDA Programming Model (UART3 Only)

24.3.5.7.1 SIR mode

24.3.5.7.1.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with parity forced to 1, baud rate = 115.2 kbps, FIFOs disabled, 2 stop-bits, and 8-bit word length:

Table 24-132. SIR Mode Receive Settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Grant access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x80
Load the baud rate(115.2 Kbps)	UART_DLL[7:0] CLOCK_LSB	0x1A
	UART_DLH[5:0] CLOCK_MSB	0x00
Set SIR mode	UART_MDR1[2:0] MODE_SELECT	0x1
Disable access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x00
Enable the UART_RHR interrupt	UART_IER_IRDA[0] RHR_IT	1

24.3.5.7.1.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 6-byte frame with no parity, baud rate = 115.2 kbps, FIFOs disabled, 3/16 encoding, 2 stop-bits, and 7-bit word length:

Table 24-133. SIR Mode Transmit Settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Grant access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x80
Load the baud rate (115.2 Kbps)	UART_DLL[7:0] CLOCK_LSB	0x1A
	UART_DLH[5:0] CLOCK_MSB	0x00
Set SIR mode	UART_MDR1[2:0] MODE_SELECT	0x1
Disable access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x00
Force DTR output to active	UART_MCR[0] DTR	1
Enable the THR interrupt	UART_IER_IRDA[1] THR_IT	0x1
Set transmit frame length to 6 bytes	UART_TXFLL[7:0] TXFLL	0x06
Set the seven starts of frame transmission	UART_EBLR[7:0] EBLR	0x08
Set SIR pulse width to be 1.6 µs	UART_ACREG[7] PULSE_TYPE	1

24.3.5.7.2 MIR mode

24.3.5.7.2.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with no parity, baud rate = 1.152 Mpbs, and FIFOs disabled.

Table 24-134. MIR Mode Receive Settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Grant access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x80
Load the baud rate (1.152bps)	UART_DLL[7:0] CLOCK_LSB	0x01
	UART_DLH[5:0] CLOCK_MSB	0x00
Set MIR mode	UART_MDR1[2:0] MODE_SELECT	0x4
Disable access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x00
Force outputs DTR and RTS to active	UART_MCR[1:0]	0x3
Enable the UART_RHR interrupt	UART_IER_IRDA[0] RHR_IT	1

24.3.5.7.2.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 60-byte frame with no parity, baud rate = 1.152 Mpbs, and FIFOs disabled.

Table 24-135. MIR Mode Transmit Settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Grant access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x80
Load the baud rate (115.2 kbps)	UART_DLL[7:0] CLOCK_LSB	0x01
	UART_DLH[5:0] CLOCK_MSB	0x00
Set SIR mode	UART_MDR1[2:0] MODE_SELECT	0x4
Disable access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x00
Force DTR output to active	UART_MCR[0] DTR	1
Enable the THR interrupt	UART_IER_IRDA[1] THR_IT	0x1
Set transmit frame length to 60 bytes	UART_TXFLL[7:0] TXFLL	0x3C
Set the eight additional starts of frame transmission	UART_EBLR[7:0] EBLR	0x08
SIP is sent at the end of transmission	UART_ACREG[3] SEND_SIP	1

24.3.5.7.3 FIR mode

24.3.5.7.3.1 Receive

The following programming model explains how to program the module to receive the IrDA frame with no parity, baud rate = 4 Mbps, FIFOs enabled, 8-bit word length.

Table 24-136. FIR Mode Receive Settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Grant access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x80
Enable access to change UART_FCR[0]	UART_DLL[7:0] CLOCK_LSB	0x0
	UART_DLH[7:0] CLOCK_MSB	
FIFO clear and enable	UART_FCR[2:0]	0x7
Set the FIFO trigger level	see Section 24.3.5.4, *Load FIFO trigger and DMA mode settings*	

Table 24-136. FIR Mode Receive Settings (continued)

Step	Register/Bit Field/Programming Model	Value
Set FIR mode	UART_MDR1[2:0] MODE_SELECT	0x5
Set frame length	UART_RXFLL[7:0] RXFLL	0xA
Disable access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x00
Enable the UART_RHR interrupt	UART_IER_IRDA[0] RHR_IT	1

24.3.5.7.3.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 4-byte frame with no parity, baud rate = 4 Mbps, FIFOs enabled, and 8-bit word length.

Table 24-137. FIR Mode Transmit Settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Grant access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x80
Enable access to change UART_FCR[0]	UART_DLL[7:0] CLOCK_LSB	0x0
	UART_DLH[5:0] CLOCK_MSB	
FIFO clear and enable	UART_FCR[2:0]	0x7
Set the FIFO trigger level	see Section 24.3.5.4, *Load FIFO trigger and DMA mode settings*	
Set FIR mode	UART_MDR1[2:0] MODE_SELECT	0x1
Disable access to the UART_DLL and UART_DLH registers	UART_LCR[7:0]	0x00
Set FIR mode and enable auto-SIP mode	UART_MDR1[7:0]	0x45
Set frame length	UART_TXFLL[7:0] TXFLL	0x4
	UART_TXFLH[7:0] TXFLH	0x0
Force DTR output to active	UART_MCR[0] DTR	1
Enable the THR interrupt	UART_IER_IRDA[1] THR_IT	1
Set the eight additional starts of frame transmission	UART_EBLR[7:0] EBLR	0x08
SIP is sent at the end of transmission	UART_ACREG[3] SEND_SIP	1

24.3.6 UART/IrDA/CIR Register Manual

24.3.6.1 UART/IrDA/CIR Instance Summary

Table 24-138. UART/IrDA/CIR Instance Summary

Module Name	Module Base Address	Size
UART1	0x4806 A000	136 Bytes
UART2	0x4806 C000	136 Bytes
UART3	0x4802 0000	136 Bytes
UART4	0x4806 E000	136 Bytes
UART5	0x4806 6000	136 Bytes
UART6	0x4806 8000	136 Bytes
UART7	0x4842 0000	136 Bytes
UART8	0x4842 2000	136 Bytes
UART9	0x4842 4000	136 Bytes
UART10	0x4AE2 B000	136 Bytes

24.3.6.2 UART/IrDA/CIR Registers

24.3.6.2.1 UART/IrDA/CIR Register Summary

Table 24-139. UART/IrDA/CIR Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	UART1 Base Address	UART2 Base Address	UART3 Base Address
UART_THR	W	32	0x0000 0000	0x4806 A000	0x4806 C000	0x4802 0000
UART_DLL	RW	32	0x0000 0000	0x4806 A000	0x4806 C000	0x4802 0000
UART_RHR	R	32	0x0000 0000	0x4806 A000	0x4806 C000	0x4802 0000
UART_IER_CIR	RW	32	0x0000 0004	0x4806 A004	0x4806 C004	0x4802 0004
UART_IER	RW	32	0x0000 0004	0x4806 A004	0x4806 C004	0x4802 0004
UART_DLH	RW	32	0x0000 0004	0x4806 A004	0x4806 C004	0x4802 0004
UART_IER_IRDA	RW	32	0x0000 0004	0x4806 A004	0x4806 C004	0x4802 0004
UART_IIR_CIR	R	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_FCR	W	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_IIR	R	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_EFR	RW	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_IIR_IRDA	R	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_LCR	RW	32	0x0000 000C	0x4806 A00C	0x4806 C00C	0x4802 000C
UART_XON1_ADDR1	RW	32	0x0000 0010	0x4806 A010	0x4806 C010	0x4802 0010
UART_MCR	RW	32	0x0000 0010	0x4806 A010	0x4806 C010	0x4802 0010
UART_LSR	R	32	0x0000 0014	0x4806 A014	0x4806 C014	0x4802 0014
UART_XON2_ADDR2	RW	32	0x0000 0014	0x4806 A014	0x4806 C014	0x4802 0014
UART_LSR_CIR	R	32	0x0000 0014	0x4806 A014	0x4806 C014	0x4802 0014
UART_LSR_IRDA	R	32	0x0000 0014	0x4806 A014	0x4806 C014	0x4802 0014
UART_XOFF1	RW	32	0x0000 0018	0x4806 A018	0x4806 C018	0x4802 0018
UART_MSR	R	32	0x0000 0018	0x4806 A018	0x4806 C018	0x4802 0018
UART_TCR	RW	32	0x0000 0018	0x4806 A018	0x4806 C018	0x4802 0018
UART_XOFF2	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C	0x4802 001C
UART_TLR	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C	0x4802 001C
UART_SPR	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C	0x4802 001C
UART_MDR1	RW	32	0x0000 0020	0x4806 A020	0x4806 C020	0x4802 0020
UART_MDR2	RW	32	0x0000 0024	0x4806 A024	0x4806 C024	0x4802 0024

Table 24-139. UART/IrDA/CIR Registers Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	UART1 Base Address	UART2 Base Address	UART3 Base Address
UART_SFLSR	R	32	0x0000 0028	0x4806 A028	0x4806 C028	0x4802 0028
UART_TXFLL	W	32	0x0000 0028	0x4806 A028	0x4806 C028	0x4802 0028
UART_RESUME	R	32	0x0000 002C	0x4806 A02C	0x4806 C02C	0x4802 002C
UART_TXFLH	W	32	0x0000 002C	0x4806 A02C	0x4806 C02C	0x4802 002C
UART_SFREGL	R	32	0x0000 0030	0x4806 A030	0x4806 C030	0x4802 0030
UART_RXFLL	W	32	0x0000 0030	0x4806 A030	0x4806 C030	0x4802 0030
UART_RXFLH	W	32	0x0000 0034	0x4806 A034	0x4806 C034	0x4802 0034
UART_SFREGH	R	32	0x0000 0034	0x4806 A034	0x4806 C034	0x4802 0034
UART_BLR	RW	32	0x0000 0038	0x4806 A038	0x4806 C038	0x4802 0038
UART_UASR	R	32	0x0000 0038	0x4806 A038	0x4806 C038	0x4802 0038
UART_ACREG	RW	32	0x0000 003C	0x4806 A03C	0x4806 C03C	0x4802 003C
UART_SCR	RW	32	0x0000 0040	0x4806 A040	0x4806 C040	0x4802 0040
UART_SSR	RW	32	0x0000 0044	0x4806 A044	0x4806 C044	0x4802 0044
UART_EBLR	RW	32	0x0000 0048	0x4806 A048	0x4806 C048	0x4802 0048
UART_MVR	R	32	0x0000 0050	0x4806 A050	0x4806 C050	0x4802 0050
UART_SYSC	RW	32	0x0000 0054	0x4806 A054	0x4806 C054	0x4802 0054
UART_SYSS	R	32	0x0000 0058	0x4806 A058	0x4806 C058	0x4802 0058
UART_WER	RW	32	0x0000 005C	0x4806 A05C	0x4806 C05C	0x4802 005C
UART_CFPS	RW	32	0x0000 0060	0x4806 A060	0x4806 C060	0x4802 0060
UART_RXFIFO_LVL	R	32	0x0000 0064	0x4806 A064	0x4806 C064	0x4802 0064
UART_TXFIFO_LVL	R	32	0x0000 0068	0x4806 A068	0x4806 C068	0x4802 0068
UART_IER2	RW	32	0x0000 006C	0x4806 A06C	0x4806 C06C	0x4802 006C
UART_ISR2	RW	32	0x0000 0070	0x4806 A070	0x4806 C070	0x4802 0070
UART_FREQ_SEL	RW	32	0x0000 0074	0x4806 A074	0x4806 C074	0x4802 0074
RESERVED	R	32	0x0000 0078	0x4806 A078	0x4806 C078	0x4802 0078
RESERVED	R	32	0x0000 007C	0x4806 A07C	0x4806 C07C	0x4802 007C
UART_MDR3	RW	32	0x0000 0080	0x4806 A080	0x4806 C080	0x4802 0080
UART_TX_DMA_THRESHOLD	RW	32	0x0000 0084	0x4806 A084	0x4806 C084	0x4802 0084

Table 24-140. UART/IrDA/CIR Registers Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	UART4 Base Address	UART5 Base Address	UART6 Base Address
UART_THR	W	32	0x0000 0000	0x4806 E000	0x4806 6000	0x4806 8000
UART_DLL	RW	32	0x0000 0000	0x4806 E000	0x4806 6000	0x4806 8000
UART_RHR	R	32	0x0000 0000	0x4806 E000	0x4806 6000	0x4806 8000
UART_IER_CIR	RW	32	0x0000 0004	0x4806 E004	0x4806 6004	0x4806 8004
UART_IER	RW	32	0x0000 0004	0x4806 E004	0x4806 6004	0x4806 8004
UART_DLH	RW	32	0x0000 0004	0x4806 E004	0x4806 6004	0x4806 8004
UART_IER_IRDA	RW	32	0x0000 0004	0x4806 E004	0x4806 6004	0x4806 8004
UART_IIR_CIR	R	32	0x0000 0008	0x4806 E008	0x4806 6008	0x4806 8008
UART_FCR	W	32	0x0000 0008	0x4806 E008	0x4806 6008	0x4806 8008
UART_IIR	R	32	0x0000 0008	0x4806 E008	0x4806 6008	0x4806 8008
UART_EFR	RW	32	0x0000 0008	0x4806 E008	0x4806 6008	0x4806 8008
UART_IIR_IRDA	R	32	0x0000 0008	0x4806 E008	0x4806 6008	0x4806 8008
UART_LCR	RW	32	0x0000 000C	0x4806 E00C	0x4806 600C	0x4806 800C

Table 24-140. UART/IrDA/CIR Registers Mapping Summary 2 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	UART4 Base Address	UART5 Base Address	UART6 Base Address
UART_XON1_ADDR1	RW	32	0x0000 0010	0x4806 E010	0x4806 6010	0x4806 8010
UART_MCR	RW	32	0x0000 0010	0x4806 E010	0x4806 6010	0x4806 8010
UART_LSR	R	32	0x0000 0014	0x4806 E014	0x4806 6014	0x4806 8014
UART_XON2_ADDR2	RW	32	0x0000 0014	0x4806 E014	0x4806 6014	0x4806 8014
UART_LSR_CIR	R	32	0x0000 0014	0x4806 E014	0x4806 6014	0x4806 8014
UART_LSR_IRDA	R	32	0x0000 0014	0x4806 E014	0x4806 6014	0x4806 8014
UART_XOFF1	RW	32	0x0000 0018	0x4806 E018	0x4806 6018	0x4806 8018
UART_MSR	R	32	0x0000 0018	0x4806 E018	0x4806 6018	0x4806 8018
UART_TCR	RW	32	0x0000 0018	0x4806 E018	0x4806 6018	0x4806 8018
UART_XOFF2	RW	32	0x0000 001C	0x4806 E01C	0x4806 601C	0x4806 801C
UART_TLR	RW	32	0x0000 001C	0x4806 E01C	0x4806 601C	0x4806 801C
UART_SPR	RW	32	0x0000 001C	0x4806 E01C	0x4806 601C	0x4806 801C
UART_MDR1	RW	32	0x0000 0020	0x4806 E020	0x4806 6020	0x4806 8020
UART_MDR2	RW	32	0x0000 0024	0x4806 E024	0x4806 6024	0x4806 8024
UART_SFLSR	R	32	0x0000 0028	0x4806 E028	0x4806 6028	0x4806 8028
UART_TXFLL	W	32	0x0000 0028	0x4806 E028	0x4806 6028	0x4806 8028
UART_RESUME	R	32	0x0000 002C	0x4806 E02C	0x4806 602C	0x4806 802C
UART_TXFLH	W	32	0x0000 002C	0x4806 E02C	0x4806 602C	0x4806 802C
UART_SFREGL	R	32	0x0000 0030	0x4806 E030	0x4806 6030	0x4806 8030
UART_RXFLL	W	32	0x0000 0030	0x4806 E030	0x4806 6030	0x4806 8030
UART_RXFLH	W	32	0x0000 0034	0x4806 E034	0x4806 6034	0x4806 8034
UART_SFREGH	R	32	0x0000 0034	0x4806 E034	0x4806 6034	0x4806 8034
UART_BLR	RW	32	0x0000 0038	0x4806 E038	0x4806 6038	0x4806 8038
UART_UASR	R	32	0x0000 0038	0x4806 E038	0x4806 6038	0x4806 8038
UART_ACREG	RW	32	0x0000 003C	0x4806 E03C	0x4806 603C	0x4806 803C
UART_SCR	RW	32	0x0000 0040	0x4806 E040	0x4806 6040	0x4806 8040
UART_SSR	RW	32	0x0000 0044	0x4806 E044	0x4806 6044	0x4806 8044
UART_EBLR	RW	32	0x0000 0048	0x4806 E048	0x4806 6048	0x4806 8048
UART_MVR	R	32	0x0000 0050	0x4806 E050	0x4806 6050	0x4806 8050
UART_SYSC	RW	32	0x0000 0054	0x4806 E054	0x4806 6054	0x4806 8054
UART_SYSS	R	32	0x0000 0058	0x4806 E058	0x4806 6058	0x4806 8058
UART_WER	RW	32	0x0000 005C	0x4806 E05C	0x4806 605C	0x4806 805C
UART_CFPS	RW	32	0x0000 0060	0x4806 E060	0x4806 6060	0x4806 8060
UART_RXFIFO_LVL	R	32	0x0000 0064	0x4806 E064	0x4806 6064	0x4806 8064
UART_TXFIFO_LVL	R	32	0x0000 0068	0x4806 E068	0x4806 6068	0x4806 8068
UART_IER2	RW	32	0x0000 006C	0x4806 E06C	0x4806 606C	0x4806 806C
UART_ISR2	RW	32	0x0000 0070	0x4806 E070	0x4806 6070	0x4806 8070
UART_FREQ_SEL	RW	32	0x0000 0074	0x4806 E074	0x4806 6074	0x4806 8074
RESERVED	R	32	0x0000 0078	0x4806 E078	0x4806 6078	0x4806 8078
RESERVED	R	32	0x0000 007C	0x4806 E07C	0x4806 607C	0x4806 807C
UART_MDR3	RW	32	0x0000 0080	0x4806 E080	0x4806 6080	0x4806 8080
UART_TX_DMA_THRESHOLD	RW	32	0x0000 0084	0x4806 E084	0x4806 6084	0x4806 8084

Table 24-141. UART/IrDA/CIR Registers Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	UART7 Base Address	UART8 Base Address
UART_THR	W	32	0x0000 0000	0x4842 0000	0x4842 2000
UART_DLL	RW	32	0x0000 0000	0x4842 0000	0x4842 2000
UART_RHR	R	32	0x0000 0000	0x4842 0000	0x4842 2000
UART_IER_CIR	RW	32	0x0000 0004	0x4842 0004	0x4842 2004
UART_IER	RW	32	0x0000 0004	0x4842 0004	0x4842 2004
UART_DLH	RW	32	0x0000 0004	0x4842 0004	0x4842 2004
UART_IER_IRDA	RW	32	0x0000 0004	0x4842 0004	0x4842 2004
UART_IIR_CIR	R	32	0x0000 0008	0x4842 0008	0x4842 2008
UART_FCR	W	32	0x0000 0008	0x4842 0008	0x4842 2008
UART_IIR	R	32	0x0000 0008	0x4842 0008	0x4842 2008
UART_EFR	RW	32	0x0000 0008	0x4842 0008	0x4842 2008
UART_IIR_IRDA	R	32	0x0000 0008	0x4842 0008	0x4842 2008
UART_LCR	RW	32	0x0000 000C	0x4842 000C	0x4842 200C
UART_XON1_ADDR1	RW	32	0x0000 0010	0x4842 0010	0x4842 2010
UART_MCR	RW	32	0x0000 0010	0x4842 0010	0x4842 2010
UART_LSR	R	32	0x0000 0014	0x4842 0014	0x4842 2014
UART_XON2_ADDR2	RW	32	0x0000 0014	0x4842 0014	0x4842 2014
UART_LSR_CIR	R	32	0x0000 0014	0x4842 0014	0x4842 2014
UART_LSR_IRDA	R	32	0x0000 0014	0x4842 0014	0x4842 2014
UART_XOFF1	RW	32	0x0000 0018	0x4842 0018	0x4842 2018
UART_MSR	R	32	0x0000 0018	0x4842 0018	0x4842 2018
UART_TCR	RW	32	0x0000 0018	0x4842 0018	0x4842 2018
UART_XOFF2	RW	32	0x0000 001C	0x4842 001C	0x4842 201C
UART_TLR	RW	32	0x0000 001C	0x4842 001C	0x4842 201C
UART_SPR	RW	32	0x0000 001C	0x4842 001C	0x4842 201C
UART_MDR1	RW	32	0x0000 0020	0x4842 0020	0x4842 2020
UART_MDR2	RW	32	0x0000 0024	0x4842 0024	0x4842 2024
UART_SFLSR	R	32	0x0000 0028	0x4842 0028	0x4842 2028
UART_TXFLL	W	32	0x0000 0028	0x4842 0028	0x4842 2028
UART_RESUME	R	32	0x0000 002C	0x4842 002C	0x4842 202C
UART_TXFLH	W	32	0x0000 002C	0x4842 002C	0x4842 202C
UART_SFREGL	R	32	0x0000 0030	0x4842 0030	0x4842 2030
UART_RXFLL	W	32	0x0000 0030	0x4842 0030	0x4842 2030
UART_RXFLH	W	32	0x0000 0034	0x4842 0034	0x4842 2034
UART_SFREGH	R	32	0x0000 0034	0x4842 0034	0x4842 2034
UART_BLR	RW	32	0x0000 0038	0x4842 0038	0x4842 2038
UART_UASR	R	32	0x0000 0038	0x4842 0038	0x4842 2038
UART_ACREG	RW	32	0x0000 003C	0x4842 003C	0x4842 203C
UART_SCR	RW	32	0x0000 0040	0x4842 0040	0x4842 2040
UART_SSR	RW	32	0x0000 0044	0x4842 0044	0x4842 2044
UART_EBLR	RW	32	0x0000 0048	0x4842 0048	0x4842 2048
UART_MVR	R	32	0x0000 0050	0x4842 0050	0x4842 2050
UART_SYSC	RW	32	0x0000 0054	0x4842 0054	0x4842 2054
UART_SYSS	R	32	0x0000 0058	0x4842 0058	0x4842 2058
UART_WER	RW	32	0x0000 005C	0x4842 005C	0x4842 205C

Table 24-141. UART/IrDA/CIR Registers Mapping Summary 3 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	UART7 Base Address	UART8 Base Address
UART_CFPS	RW	32	0x0000 0060	0x4842 0060	0x4842 2060
UART_RXFIFO_LVL	R	32	0x0000 0064	0x4842 0064	0x4842 2064
UART_TXFIFO_LVL	R	32	0x0000 0068	0x4842 0068	0x4842 2068
UART_IER2	RW	32	0x0000 006C	0x4842 006C	0x4842 206C
UART_ISR2	RW	32	0x0000 0070	0x4842 0070	0x4842 2070
UART_FREQ_SEL	RW	32	0x0000 0074	0x4842 0074	0x4842 2074
RESERVED	R	32	0x0000 0078	0x4842 0078	0x4842 2078
RESERVED	R	32	0x0000 007C	0x4842 007C	0x4842 207C
UART_MDR3	RW	32	0x0000 0080	0x4842 0080	0x4842 2080
UART_TX_DMA_THRESHOLD	RW	32	0x0000 0084	0x4842 0084	0x4842 2084

Table 24-142. UART/IrDA/CIR Registers Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	UART9 Base Address	UART10 Base Address
UART_THR	W	32	0x0000 0000	0x4842 4000	0x4AE2 B000
UART_DLL	RW	32	0x0000 0000	0x4842 4000	0x4AE2 B000
UART_RHR	R	32	0x0000 0000	0x4842 4000	0x4AE2 B000
UART_IER_CIR	RW	32	0x0000 0004	0x4842 4004	0x4AE2 B004
UART_IER	RW	32	0x0000 0004	0x4842 4004	0x4AE2 B004
UART_DLH	RW	32	0x0000 0004	0x4842 4004	0x4AE2 B004
UART_IER_IRDA	RW	32	0x0000 0004	0x4842 4004	0x4AE2 B004
UART_IIR_CIR	R	32	0x0000 0008	0x4842 4008	0x4AE2 B008
UART_FCR	W	32	0x0000 0008	0x4842 4008	0x4AE2 B008
UART_IIR	R	32	0x0000 0008	0x4842 4008	0x4AE2 B008
UART_EFR	RW	32	0x0000 0008	0x4842 4008	0x4AE2 B008
UART_IIR_IRDA	R	32	0x0000 0008	0x4842 4008	0x4AE2 B008
UART_LCR	RW	32	0x0000 000C	0x4842 400C	0x4AE2 B00C
UART_XON1_ADDR1	RW	32	0x0000 0010	0x4842 4010	0x4AE2 B010
UART_MCR	RW	32	0x0000 0010	0x4842 4010	0x4AE2 B010
UART_LSR	R	32	0x0000 0014	0x4842 4014	0x4AE2 B014
UART_XON2_ADDR2	RW	32	0x0000 0014	0x4842 4014	0x4AE2 B014
UART_LSR_CIR	R	32	0x0000 0014	0x4842 4014	0x4AE2 B014
UART_LSR_IRDA	R	32	0x0000 0014	0x4842 4014	0x4AE2 B014
UART_XOFF1	RW	32	0x0000 0018	0x4842 4018	0x4AE2 B018
UART_MSR	R	32	0x0000 0018	0x4842 4018	0x4AE2 B018
UART_TCR	RW	32	0x0000 0018	0x4842 4018	0x4AE2 B018
UART_XOFF2	RW	32	0x0000 001C	0x4842 401C	0x4AE2 B01C
UART_TLR	RW	32	0x0000 001C	0x4842 401C	0x4AE2 B01C
UART_SPR	RW	32	0x0000 001C	0x4842 401C	0x4AE2 B01C
UART_MDR1	RW	32	0x0000 0020	0x4842 4020	0x4AE2 B020
UART_MDR2	RW	32	0x0000 0024	0x4842 4024	0x4AE2 B024
UART_SFLSR	R	32	0x0000 0028	0x4842 4028	0x4AE2 B028
UART_TXFLL	W	32	0x0000 0028	0x4842 4028	0x4AE2 B028
UART_RESUME	R	32	0x0000 002C	0x4842 402C	0x4AE2 B02C
UART_TXFLH	W	32	0x0000 002C	0x4842 402C	0x4AE2 B02C
UART_SFREGL	R	32	0x0000 0030	0x4842 4030	0x4AE2 B030

Table 24-142. UART/IrDA/CIR Registers Mapping Summary 4 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	UART9 Base Address	UART10 Base Address
UART_RXFLL	W	32	0x0000 0030	0x4842 4030	0x4AE2 B030
UART_RXFLH	W	32	0x0000 0034	0x4842 4034	0x4AE2 B034
UART_SFREGH	R	32	0x0000 0034	0x4842 4034	0x4AE2 B034
UART_BLR	RW	32	0x0000 0038	0x4842 4038	0x4AE2 B038
UART_UASR	R	32	0x0000 0038	0x4842 4038	0x4AE2 B038
UART_ACREG	RW	32	0x0000 003C	0x4842 403C	0x4AE2 B03C
UART_SCR	RW	32	0x0000 0040	0x4842 4040	0x4AE2 B040
UART_SSR	RW	32	0x0000 0044	0x4842 4044	0x4AE2 B044
UART_EBLR	RW	32	0x0000 0048	0x4842 4048	0x4AE2 B048
UART_MVR	R	32	0x0000 0050	0x4842 4050	0x4AE2 B050
UART_SYSC	RW	32	0x0000 0054	0x4842 4054	0x4AE2 B054
UART_SYSS	R	32	0x0000 0058	0x4842 4058	0x4AE2 B058
UART_WER	RW	32	0x0000 005C	0x4842 405C	0x4AE2 B05C
UART_CFPS	RW	32	0x0000 0060	0x4842 4060	0x4AE2 B060
UART_RXFIFO_LVL	R	32	0x0000 0064	0x4842 4064	0x4AE2 B064
UART_TXFIFO_LVL	R	32	0x0000 0068	0x4842 4068	0x4AE2 B068
UART_IER2	RW	32	0x0000 006C	0x4842 406C	0x4AE2 B06C
UART_ISR2	RW	32	0x0000 0070	0x4842 4070	0x4AE2 B070
UART_FREQ_SEL	RW	32	0x0000 0074	0x4842 4074	0x4AE2 B074
RESERVED	R	32	0x0000 0078	0x4842 4078	0x4AE2 B078
RESERVED	R	32	0x0000 007C	0x4842 407C	0x4AE2 B07C
UART_MDR3	RW	32	0x0000 0080	0x4842 4080	0x4AE2 B080
UART_TX_DMA_THRESHOLD	RW	32	0x0000 0084	0x4842 4084	0x4AE2 B084

24.3.6.2.2 UART/IrDA/CIR Register Description

Table 24-143. UART_THR

Address Offset	0x0000 0000		
Physical Address	0x4806 A000 0x4806 C000 0x4802 0000 0x4806 E000 0x4806 6000 0x4806 8000 0x4842 0000 0x4842 2000 0x4842 4000 0x4AE2 B000	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	The transmitter section consists of the transmit holding register (THR) and the transmit shift register. The THR is a 64-byte FIFO. The local host (LH) writes data to the THR. The data is placed in the transmit shift register where it is shifted out serially on the TX output. If the FIFO is disabled, location 0 of the FIFO stores the data.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								THR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:0	THR	Transmit holding register	W	0x-

Table 24-144. UART_RHR

Address Offset	0x0000 0000		
Physical Address	0x4806 A000 0x4806 C000 0x4802 0000 0x4806 E000 0x4806 6000 0x4806 8000 0x4842 0000 0x4842 2000 0x4842 4000 0x4AE2 B000	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	The receiver section consists of the receiver holding register (RHR) and the receiver shift register. The RHR is a 64-byte FIFO. The receiver shift register receives serial data from RX input. The data is converted to parallel data and moved to the RHR. If the FIFO is disabled, location 0 of the FIFO stores the single data character. **Note:** If an overflow occurs, the data in the RHR is not overwritten.		
Type	R		

31–8	7–0
RESERVED	RHR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0	R	0x000000
7:0	RHR	Receive holding register	R	0x-

Table 24-145. UART_DLL

Address Offset	0x0000 0000		
Physical Address	0x4806 A000 0x4806 C000 0x4802 0000 0x4806 E000 0x4806 6000 0x4806 8000 0x4842 0000 0x4842 2000 0x4842 4000 0x4AE2 B000	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	This register, with UART_DLH, stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor. DLL stores the least-significant part of the divisor.		
Type	RW		

31–8	7–0
RESERVED	CLOCK_LSB

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	CLOCK_LSB	Stores the 8-bit LSB divisor value	RW	0x00

Table 24-146. UART_IER

Address Offset	0x0000 0004

Table 24-146. UART_IER (continued)

Physical Address	0x4806 A004	**Instance**	UART1
	0x4806 C004		UART2
	0x4802 0004		UART3
	0x4806 E004		UART4
	0x4806 6004		UART5
	0x4806 8004		UART6
	0x4842 0004		UART7
	0x4842 2004		UART8
	0x4842 4004		UART9
	0x4AE2 B004		UART10
Description	Interrupt enable register		
Type	RW		

31–8	7	6	5	4	3	2	1	0
RESERVED	CTS_IT	RTS_IT	XOFF_IT	SLEEP_MODE	MODEM_STS_IT	LINE_STS_IT	THR_IT	RHR_IT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	CTS_IT	0x0: Disables the CTS* interrupt 0x1: Enables the CTS* interrupt	RW	0
6	RTS_IT	0x0: Disables the RTS* interrupt 0x1: Enables the RTS* interrupt	RW	0
5	XOFF_IT	0x0: Disables the XOFF interrupt 0x1: Enables the XOFF interrupt	RW	0
4	SLEEP_MODE	0x0: Disables sleep mode 0x1: Enables sleep mode (stop baud rate clock when the module is inactive)	RW	0
3	MODEM_STS_IT	0x0: Disables the modem status register interrupt 0x1: Enables the modem status register interrupt	RW	0
2	LINE_STS_IT	0x0: Disables the receiver line status interrupt 0x1: Enables the receiver line status interrupt	RW	0
1	THR_IT	0x0: Disables the THR interrupt 0x1: Enables the THR interrupt	RW	0
0	RHR_IT	0x0: Disables the RHR interrupt and time-out interrupt 0x1: Enables the RHR interrupt and time-out interrupt	RW	0

Table 24-147. UART_IER_IRDA

Address Offset	0x0000 0004

Table 24-147. UART_IER_IRDA (continued)

Physical Address	0x4806 A004	**Instance**	UART1
	0x4806 C004		UART2
	0x4802 0004		UART3
	0x4806 E004		UART4
	0x4806 6004		UART5
	0x4806 8004		UART6
	0x4842 0004		UART7
	0x4842 2004		UART8
	0x4842 4004		UART9
	0x4AE2 B004		UART10

Description
There are 8 types of interrupt in these modes, received EOF, LSR interrupt, TX status, status FIFO interrupt, RX overrun, last byte in RX FIFO, THR interrupt and RHR interrupt and they can be enabled/disabled individually.
Note: The TX_STATUS_IT interrupt reflects two possible conditions. The UART_MDR2[0] should be read to determine the status in the event of this interrupt.

Type
RW

31–8	7	6	5	4	3	2	1	0
RESERVED	EOF_IT	LINE_STS_IT	TX_STATUS_IT	STS_FIFO_TRIG_IT	RX_OVERRUN_IT	LAST_RX_BYTE_IT	THR_IT	RHR_IT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	EOF_IT	0x0: Disables the received EOF interrupt 0x1: Enables the received EOF interrupt	RW	0
6	LINE_STS_IT	0x0: Disables the receiver line status interrupt 0x1: Enables the receiver line status interrupt	RW	0
5	TX_STATUS_IT	0x0: Disables the TX status interrupt 0x1: Enables the TX status interrupt	RW	0
4	STS_FIFO_TRIG_IT	0x0: Disables status FIFO trigger level interrupt 0x1: Enables status FIFO trigger level interrupt	RW	0
3	RX_OVERRUN_IT	0x0: Disables the RX overrun interrupt 0x1: Enables the RX overrun interrupt	RW	0
2	LAST_RX_BYTE_IT	0x0: Disables the last byte of frame in RX FIFO interrupt 0x1: Enables the last byte of frame in RX FIFO interrupt	RW	0
1	THR_IT	0x0: Disables the THR interrupt 0x1: Enables the THR interrupt	RW	0
0	RHR_IT	0x0: Disables the RHR interrupt and time-out interrupt 0x1: Enables the RHR interrupt and time-out interrupt	RW	0

Table 24-148. UART_IER_CIR

Address Offset
0x0000 0004

Table 24-148. UART_IER_CIR (continued)

Physical Address	0x4806 A004 0x4806 C004 0x4802 0004 0x4806 E004 0x4806 6004 0x4806 8004 0x4842 0004 0x4842 2004 0x4842 4004 0x4AE2 B004	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	There are 6 types of interrupt in these modes, TX status, status FIFO interrupt, RX overrun, last byte in RX FIFO, THR interrupt and RHR interrupt and they can be enabled/disabled individually. Notes: The RX_STOP_IT interrupt is generated based on the value set in the BOF Length register (UART_EBLR). In IR-CIR mode, contrary to the IR-IRDA mode, the TX_STATUS_IT has only one meaning corresponding to the case UART_MDR2[0] = 0.		
Type	RW		

31:8	7:6	5	4	3	2	1	0
RESERVED	RESERVED	TX_STATUS_IT	RESERVED	RX_OVERRUN_IT	RX_STOP_IT	THR_IT	RHR_IT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:6	RESERVED	Not used in CIR mode	RW	0x0
5	TX_STATUS_IT	0x0: Disables the TX status interrupt 0x1: Enables the TX status interrupt	RW	0
4	RESERVED	Not used in CIR mode	RW	0
3	RX_OVERRUN_IT	0x0: Disables the RX overrun interrupt 0x1: Enables the RX overrun interrupt	RW	0
2	RX_STOP_IT	0x0: Disables the receive stop interrupt 0x1: Enables the receive stop interrupt	RW	0
1	THR_IT	0x0: Disables the THR interrupt 0x1: Enables the THR interrupt	RW	0
0	RHR_IT	0x0: Disables the RHR interrupt 0x1: Enables the RHR interrupt	RW	0

Table 24-149. UART_DLH

Address Offset	0x0000 0004		
Physical Address	0x4806 A004 0x4806 C004 0x4802 0004 0x4806 E004 0x4806 6004 0x4806 8004 0x4842 0004 0x4842 2004 0x4842 4004 0x4AE2 B004	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10

Table 24-149. UART_DLH (continued)

Description	This register, with UART_DLL, stores the 14-bit divisor for generating the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor. DLL stores the least-significant part of the divisor.
Type	RW

31–8	7–6	5–0
RESERVED	RESERVED	CLOCK_MSB

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:6	RESERVED	Read returns 0. Write has no effect.	RW	0x0
5:0	CLOCK_MSB	Stores the 6-bit MSB divisor value	RW	0x00

Table 24-150. UART_IIR

Address Offset	0x0000 0008		
Physical Address	0x4806 A008	**Instance**	UART1
	0x4806 C008		UART2
	0x4802 0008		UART3
	0x4806 E008		UART4
	0x4806 6008		UART5
	0x4806 8008		UART6
	0x4842 0008		UART7
	0x4842 2008		UART8
	0x4842 4008		UART9
	0x4AE2 B008		UART10
Description	Interrupt identification register. The IIR is a read-only register that provides the source of the interrupt in a prioritized manner.		
Type	R		

31–8	7–6	5–1	0
RESERVED	FCR_MIRROR	IT_TYPE	IT_PENDING

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	R	0x000000
7:6	FCR_MIRROR	Mirror the contents of UART_FCR[0] on both bits.	R	0x0
5:1	IT_TYPE	Read 0x0: Modem interrupt. Priority = 4 Read 0x1: THR interrupt. Priority = 3 Read 0x2: RHR interrupt. Priority = 2 Read 0x3: Receiver line status error. Priority = 3 Read 0x6: Rx time-out. Priority = 2 Read 0x8: XOFF/special character. Priority = 5 Read 0x10: CTS, RTS, DSR change state from active (low) to inactive (high) Priority = 6	R	0x00
0	IT_PENDING	Read 0x0: An interrupt is pending. Read 0x1: No interrupt is pending.	R	1

Table 24-151. UART_IIR_IRDA

Address Offset	0x0000 0008

Table 24-151. UART_IIR_IRDA (continued)

Physical Address	0x4806 A008 0x4806 C008 0x4802 0008 0x4806 E008 0x4806 6008 0x4806 8008 0x4842 0008 0x4842 2008 0x4842 4008 0x4AE2 B008	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	The interrupt line is activated whenever one of the 8 interrupts is active.		
Type	R		

31–8	7	6	5	4	3	2	1	0
RESERVED	EOF_IT	LINE_STS_IT	TX_STATUS_IT	STS_FIFO_IT	RX_OE_IT	RX_FIFO_LAST_BYTE_IT	THR_IT	RHR_IT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	R	0x000000
7	EOF_IT	Read 0x0: Receive EOF interrupt inactive Read 0x1: Received EOF interrupt active	R	0
6	LINE_STS_IT	Read 0x0: Receiver line status interrupt inactive Read 0x1: Receiver line status interrupt active	R	0
5	TX_STATUS_IT	Read 0x0: TX status interrupt inactive Read 0x1: TX status interrupt active	R	0
4	STS_FIFO_IT	Read 0x0: Status FIFO trigger level interrupt inactive Read 0x1: Status FIFO trigger level interrupt active	R	0
3	RX_OE_IT	Read 0x0: RX overrun interrupt inactive Read 0x1: RX overrun interrupt active	R	0
2	RX_FIFO_LAST_BYTE_IT	Read 0x0: Last byte of frame in RX FIFO interrupt inactive Read 0x1: Last byte of frame in RX FIFO interrupt active	R	0
1	THR_IT	Read 0x0: THR interrupt inactive Read 0x1: THR interrupt active	R	0
0	RHR_IT	Read 0x0: RHR interrupt inactive Read 0x1: RHR interrupt active	R	1

Table 24-152. UART_IIR_CIR

Address Offset	0x0000 0008

Table 24-152. UART_IIR_CIR (continued)

Physical Address	0x4806 A008 0x4806 C008 0x4802 0008 0x4806 E008 0x4806 6008 0x4806 8008 0x4842 0008 0x4842 2008 0x4842 4008 0x4AE2 B008	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	The interrupt line is activated whenever one of the 6 interrupts is active.		
Type	R		

31–8	7–6	5	4	3	2	1	0
RESERVED	RESERVED	TX_STATUS_IT	RESERVED	RX_OE_IT	RX_STOP_IT	THR_IT	RHR_IT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	R	0x000000
7:6	RESERVED	Not used in CIR mode	R	0x0
5	TX_STATUS_IT	Read 0x0: TX status interrupt inactive Read 0x1: TX status interrupt active	R	0
4	RESERVED	Not used in CIR mode	R	0
3	RX_OE_IT	Read 0x0: RX overrun interrupt inactive Read 0x1: RX overrun interrupt active	R	0
2	RX_STOP_IT	Read 0x0: Receive stop interrupt inactive Read 0x1: Receive stop interrupt active	R	0
1	THR_IT	Read 0x0: THR interrupt inactive Read 0x1: THR interrupt active	R	0
0	RHR_IT	Read 0x0: RHR interrupt inactive Read 0x1: RHR interrupt active	R	0

Table 24-153. UART_FCR

Address Offset	0x0000 0008		
Physical Address	0x4806 A008 0x4806 C008 0x4802 0008 0x4806 E008 0x4806 6008 0x4806 8008 0x4842 0008 0x4842 2008 0x4842 4008 0x4AE2 B008	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	FIFO control register **Notes:** Bits 4 and 5 can only be written to when UART_EFR[4] = 1. Bits 0 and 3 can be changed only when the baud clock is not running (DLL and DLH set to 0). See Table 24-99 for UART_FCR[5:4] setting restriction when UART_SCR[6] = 1. See Table 24-100 for UART_FCR[7:6] setting restriction when UART_SCR[7] = 1.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RX_FIFO_TRIG	TX_FIFO_TRIG	DMA_MODE	TX_FIFO_CLEAR	RX_FIFO_CLEAR	FIFO_EN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:6	RX_FIFO_TRIG	Sets the trigger level for the RX FIFO: If UART_SCR[7] = 0 and UART_TLR[7:4] = 0000: 00: 8 characters 01: 16 characters 10: 56 characters 11: 60 characters If UART_SCR[7] = 0 and UART_TLR[7:4] != 0000, RX_FIFO_TRIG is not considered. If UART_SCR[7] = 1, RX_FIFO_TRIG is 2 LSBs of the trigger level (1-63 on 6 bits) with the granularity 1.	W	0x0
5:4	TX_FIFO_TRIG	Sets the trigger level for the TX FIFO: If UART_SCR[6] = 0 and UART_TLR[3:0] = 0000: 00: 8 spaces 01: 16 spaces 10: 32 spaces 11: 56 spaces If UART_SCR[6] = 0 and UART_TLR[3:0] != 0000, TX_FIFO_TRIG is not considered. If UART_SCR[6] = 1, TX_FIFO_TRIG is 2 LSBs of the trigger level (1-63 on 6 bits) with the granularity 1	W	0x0
3	DMA_MODE	This register is considered if UART_SCR[0] = 0. Write 0x0: DMA_MODE 0 (No DMA) Write 0x1: DMA_MODE 1 (UART_nDMA_REQ[0] in TX (UARTi_DREQ_TX), UART_nDMA_REQ[1] in RX (UARTi_DREQ_RX))	W	0
2	TX_FIFO_CLEAR	Write 0x0: No change Write 0x1: Clears the TX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.	W	0
1	RX_FIFO_CLEAR	Write 0x0: No change Write 0x1: Clears the RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.	W	0
0	FIFO_EN	Write 0x0: Disables the transmit and RX FIFOs. The transmit and receive holding registers are 1-byte FIFOs. Write 0x1: Enables the transmit and RX FIFOs. The transmit and receive holding registers are 64-byte FIFOs.	W	0

Table 24-154. UART_EFR

Address Offset	0x0000 0008		
Physical Address	0x4806 A008 0x4806 C008 0x4802 0008 0x4806 E008 0x4806 6008 0x4806 8008 0x4842 0008 0x4842 2008 0x4842 4008 0x4AE2 B008	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Enhanced feature register		

Table 24-154. UART_EFR (continued)

	This register enables or disables enhanced features. Most of the enhanced functions apply only to UART modes, but UART_EFR[4] enables write accesses to UART_FCR[5:4], the TX trigger level, which is also used in IrDA modes.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								AUTO_CTS_EN	AUTO_RTS_EN	SPECIAL_CHAR_DETECT	ENHANCED_EN	SW_FLOW_CONTROL			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	AUTO_CTS_EN	Auto-CTS enable bit 0x0: Normal operation 0x1: Auto-CTS flow control is enabled. Transmission is halted when the CTS* pin is high (inactive).	RW	0
6	AUTO_RTS_EN	Auto-RTS enable bit 0x0: Normal operation 0x1: Auto-RTS flow control is enabled. RTS* pin goes high (inactive) when the RX FIFO HALT trigger level, UART_TCR[3:0], is reached, and goes low (active) when the RX FIFO RESTORE transmission trigger level is reached.	RW	0
5	SPECIAL_CHAR_DETECT	0x0: Normal operation 0x1: Special character detect enable. Received data is compared with XOFF2 data. If a match occurs, the received data is transferred to the RX FIFO and the UART_IIR[4] bit is set to 1 to indicate that a special character was detected.	RW	0
4	ENHANCED_EN	Enhanced functions write enable bit 0x0: Disables writing to IER bits 4-7, UART_FCR bits 4-5, and UART_MCR bits 5-7. 0x1: Enables writing to IER bits 4-7, UART_FCR bits 4-5, and UART_MCR bits 5-7.	RW	0
3:0	SW_FLOW_CONTROL	Combinations of software flow control can be selected by programming bit 3 - bit 0. See Table 24-113.	RW	0x0

Table 24-155. UART_LCR

Address Offset	0x0000 000C		
Physical Address	0x4806 A00C	**Instance**	UART1
	0x4806 C00C		UART2
	0x4802 000C		UART3
	0x4806 E00C		UART4
	0x4806 600C		UART5
	0x4806 800C		UART6
	0x4842 000C		UART7
	0x4842 200C		UART8
	0x4842 400C		UART9
	0x4AE2 B00C		UART10
Description	Line control register		

Table 24-155. UART_LCR (continued)

	LCR[6:0] define transmission and reception parameters. **Note:** When LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1.
Type	RW

31–8	7	6	5	4	3	2	1–0
RESERVED	DIV_EN	BREAK_EN	PARITY_TYPE2	PARITY_TYPE1	PARITY_EN	NB_STOP	CHAR_LENGTH

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	DIV_EN	0x0: Normal operating condition 0x1: Divisor latch enable. Allows access to UART_DLL, UART_DLH, and other registers (see Table 24-101).	RW	0
6	BREAK_EN	Break control bit 0x0: Normal operating condition 0x1: Forces the transmitter output to go low to alert the communication terminal	RW	0
5	PARITY_TYPE2	Selects the forced parity format (if UART_LCR[3] = 1). If UART_LCR[5] = 1 and UART_LCR[4] = 0, the parity bit is forced to 1 in the transmitted and received data. If UART_LCR[5] = 1 and UART_LCR[4] = 1, the parity bit is forced to 0 in the transmitted and received data. **UART_LCR[3] / UART_LCR[4] / UART_LCR[5] / Parity** 0 / N/A / N/A / No parity 1 / 0 / 0 / Odd parity 1 / 1 / 0 / Even parity 1 / 0 / 1 / Forced 1 1 / 1 / 1 / Forced 0	RW	0
4	PARITY_TYPE1	0x0: Odd parity is generated (if UART_LCR[3] = 1). 0x1: Even parity is generated (if UART_LCR[3] = 1).	RW	0
3	PARITY_EN	0x0: No parity 0x1: A parity bit is generated during transmission and the receiver checks for received parity.	RW	0
2	NB_STOP	Specifies the number of stop-bits 0x0: 1 stop-bit (word length = 5, 6, 7, 8) 0x1: 1.5 stop-bits (word length = 5) 2 stop-bits (word length = 6, 7, 8)	RW	0
1:0	CHAR_LENGTH	Specifies the word length to be transmitted or received 0x0: 5 bits 0x1: 6 bits 0x2: 7 bits 0x3: 8 bits	RW	0x0

Table 24-156. UART_XON1_ADDR1

Address Offset	0x0000 0010

Table 24-156. UART_XON1_ADDR1 (continued)

Physical Address	0x4806 A010 0x4806 C010 0x4802 0010 0x4806 E010 0x4806 6010 0x4806 8010 0x4842 0010 0x4842 2010 0x4842 4010 0x4AE2 B010	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	UART mode: XON1 character, IrDA mode: ADDR1 address		
Type	RW		

31–8	7–0
RESERVED	XON_WORD1

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XON_WORD1	Stores the 8-bit XON1 character in UART modes and ADDR1 address 1 for IrDA modes	RW	0x00

Table 24-157. UART_MCR

Address Offset	0x0000 0010		
Physical Address	0x4806 A010 0x4806 C010 0x4802 0010 0x4806 E010 0x4806 6010 0x4806 8010 0x4842 0010 0x4842 2010 0x4842 4010 0x4AE2 B010	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Modem control register MCR[3:0] controls the interface with the modem, data set, or peripheral device that emulates the modem.		
Type	RW		

31–8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	TCR_TLR	XON_EN	LOOPBACK_EN	CD_STS_CH	RI_STS_CH	RTS	DTR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	RESERVED	Read returns 0. Write has no effect.	RW	0
6	TCR_TLR	0x0: No action 0x1: Enables access to the UART_TCR and UART_TLR registers	RW	0
5	XON_EN	0x0: Disable XON any function. 0x1: Enable XON any function.	RW	0
4	LOOPBACK_EN	0x0: Normal operating mode	RW	0

Bits	Field Name	Description	Type	Reset
		0x1: Enable local loopback mode (internal). In this mode, the MCR[3:0] signals are looped back into the UART_MSR[7:4] bit field. The transmit output is looped back to the receive input internally.		
3	CD_STS_CH	0x0: In loopback, forces DCD* input high and IRQ outputs to inactive state 0x1: In loopback, forces DCD* input low and IRQ outputs to inactive state	RW	0
2	RI_STS_CH	0x0: In loopback, forces RI* input high 0x1: In loopback, forces RI* input low	RW	0
1	RTS	In loopback, controls the UART_MSR[4] bit. If auto-RTS is enabled, the RTS* output is controlled by hardware flow control. 0x0: Force RTS* output to inactive (high). 0x1: Force RTS* output to active (low).	RW	0
0	DTR	0x0: Force DTR* output to inactive (high). 0x1: Force DTR* output to active (low).	RW	0

Table 24-158. UART_LSR

Address Offset	0x0000 0014		
Physical Address	0x4806 A014 0x4806 C014 0x4802 0014 0x4806 E014 0x4806 6014 0x4806 8014 0x4842 0014 0x4842 2014 0x4842 4014 0x4AE2 B014	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Line status register		
Type	R		

31:8	7	6	5	4	3	2	1	0
RESERVED	RX_FIFO_STS	TX_SR_E	TX_FIFO_E	RX_BI	RX_FE	RX_PE	RX_OE	RX_FIFO_E

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	RX_FIFO_STS	Read 0x0: Normal operation Read 0x1: At least one parity error, framing error, or break indication in the RX FIFO. Bit 7 is cleared when no more errors are present in the RX FIFO.	R	0
6	TX_SR_E	Read 0x0: Transmitter hold (TX FIFO) and shift registers are not empty. Read 0x1: Transmitter hold (TX FIFO) and shift registers are empty.	R	1
5	TX_FIFO_E	Read 0x0: Transmit hold register (TX FIFO) is not empty. Read 0x1: Transmit hold register (TX FIFO) is empty. The transmission is not necessarily complete.	R	1
4	RX_BI	Read 0x0: No break condition	R	0

Bits	Field Name	Description	Type	Reset
		Read 0x1: A break was detected while the data from the RX FIFO was received (for example, RX input was low for one character + 1 bit time frame).		
3	RX_FE	Read 0x0: No framing error in data RX FIFO Read 0x1: Framing error occurred in data from RX FIFO (received data did not have a valid stop-bit).	R	0
2	RX_PE	Read 0x0: No parity error in data from RX FIFO Read 0x1: Parity error in data from RX FIFO	R	0
1	RX_OE	Read 0x0: No overrun error Read 0x1: Overrun error occurred. Set when the character in the receive shift register is not transferred to the RX FIFO. This occurs only when the RX FIFO is full.	R	0
0	RX_FIFO_E	Read 0x0: No data in the RX FIFO Read 0x1: At least one data character in the RX FIFO	R	0

Table 24-159. UART_LSR_IRDA

Address Offset	0x0000 0014		
Physical Address	0x4806 A014 0x4806 C014 0x4802 0014 0x4806 E014 0x4806 6014 0x4806 8014 0x4842 0014 0x4842 2014 0x4842 4014 0x4AE2 B014	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	When the LSR is read, LSR[4:2] reflect the error bits [FL, CRC, ABORT] of the frame at the top of the STATUS FIFO (next frame status to be read).		
Type	R		

31–8	7	6	5	4	3	2	1	0
RESERVED	THR_EMPTY	STS_FIFO_FULL	RX_LAST_BYTE	FRAME_TOO_LONG	ABORT	CRC	STS_FIFO_E	RX_FIFO_E

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	THR_EMPTY	Read 0x0: Transmit holding register (TX FIFO) is not empty. Read 0x1: Transmit hold register (TX FIFO) is empty. The transmission is not necessarily complete.	R	1
6	STS_FIFO_FULL	Read 0x0: Status FIFO not full Read 0x1: Status FIFO full	R	0
5	RX_LAST_BYTE	Read 0x0: The RX FIFO (RHR) does not contain the last byte of the frame to be read.	R	0

Bits	Field Name	Description	Type	Reset
		Read 0x1: The RX FIFO (RHR) contains the last byte of the frame to be read. This bit is set only when the last byte of a frame is available to be read. It determines the frame boundary. It is cleared on a single read of the LSR register. See the note below.		
4	FRAME_TOO_LONG	Read 0x0: No frame-too-long error in frame Read 0x1: Frame-too-long error in the frame at the top of the STATUS FIFO, (next character to be read). This bit is set to 1 when a frame exceeding the maximum length (set by RXFLH and RXFLL registers) is received. When this error is detected, current frame reception is terminated. Reception is stopped until the next START flag is detected.	R	0
3	ABORT	Read 0x0: No abort pattern error in frame Read 0x1: Abort pattern is received. SIR and MIR: Abort pattern FIR: Illegal symbol	R	0
2	CRC	Read 0x0: No CRC error in frame Read 0x1: CRC error in the frame at the top of the STATUS FIFO (next character to be read)	R	0
1	STS_FIFO_E	Read 0x0: Status FIFO not empty Read 0x1: Status FIFO empty	R	1
0	RX_FIFO_E	Read 0x0: No data in the RX FIFO Read 0x1: At least one data character in the RX FIFO	R	1

Table 24-160. UART_LSR_CIR

Address Offset	0x0000 0014		
Physical Address	0x4806 A014	**Instance**	UART1
	0x4806 C014		UART2
	0x4802 0014		UART3
	0x4806 E014		UART4
	0x4806 6014		UART5
	0x4806 8014		UART6
	0x4842 0014		UART7
	0x4842 2014		UART8
	0x4842 4014		UART9
	0x4AE2 B014		UART10
Description	Line status register in CIR mode		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								THR_EMPTY	RESERVED	RX_STOP	RESERVED				RX_FIFO_E

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	THR_EMPTY	Read 0x0: Transmit holding register (TX FIFO) is not empty. Read 0x1: Transmit hold register (TX FIFO) is empty. The transmission is not necessarily complete.	R	1
6	RESERVED	Not used in CIR mode	R	0

Bits	Field Name	Description	Type	Reset
5	RX_STOP	The RX_STOP is generated based on the value set in the BOF Length register (UART_EBLR). It is cleared on a single read of the UART_LSR register. Read 0x0: Reception is ongoing or waiting for a new frame. Read 0x1: Reception is complete.	R	0
4:1	RESERVED	Not used in CIR mode	R	0x0
0	RX_FIFO_E	Read 0x0: At least one data character in the RX FIFO Read 0x1: No data in the RX FIFO	R	1

Table 24-161. UART_XON2_ADDR2

Address Offset	0x0000 0014		
Physical Address	0x4806 A014 0x4806 C014 0x4802 0014 0x4806 E014 0x4806 6014 0x4806 8014 0x4842 0014 0x4842 2014 0x4842 4014 0x4AE2 B014	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Stores the 8-bit XON2 character in UART moldes and ADDR2 address 2 for IrDA modes		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XON_WORD2							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XON_WORD2	Stores the 8-bit XON2 character in UART modes and ADDR2 address 2 for IrDA modes	RW	0x00

Table 24-162. UART_TCR

Address Offset	0x0000 0018		
Physical Address	0x4806 A018 0x4806 C018 0x4802 0018 0x4806 E018 0x4806 6018 0x4806 8018 0x4842 0018 0x4842 2018 0x4842 4018 0x4AE2 B018	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Transmission control register This register stores the RX FIFO threshold levels to start/stop transmission during hardware/software flow control. Notes: Trigger levels from 0 to 60 bytes are available with a granularity of 4. (Trigger level = 4 x [4-bit register value]) The programmer must ensure that UART_TCR[3:0] > UART_TCR[7:4] when auto-RTS or software flow control is enabled to avoid a mis-operation of the device. In FIFO interrupt mode with flow control, the programmer must ensure that the trigger level to halt transmission is greater than or equal to the RX FIFO trigger level (UART_TLR[7:4] or UART_FCR[7:6]); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist because a DMA request is sent each time a byte is received.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RX_FIFO_TRIG_START	RX_FIFO_TRIG_HALT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:4	RX_FIFO_TRIG_START	RX FIFO trigger level to RESTORE transmission (0 - 60)	RW	0x0
3:0	RX_FIFO_TRIG_HALT	RX FIFO trigger level to HALT transmission (0 - 60)	RW	0xF

Table 24-163. UART_XOFF1

Address Offset	0x0000 0018		
Physical Address	0x4806 A018 0x4806 C018 0x4802 0018 0x4806 E018 0x4806 6018 0x4806 8018 0x4842 0018 0x4842 2018 0x4842 4018 0x4AE2 B018	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	UART mode XOFF1 character		
Type	RW		

31:8	7:0
RESERVED	XOFF_WORD1

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XOFF_WORD1	Stores the 8-bit XOFF1 character used in UART modes	RW	0x00

Table 24-164. UART_MSR

Address Offset	0x0000 0018		
Physical Address	0x4806 A018 0x4806 C018 0x4802 0018 0x4806 E018 0x4806 6018 0x4806 8018 0x4842 0018 0x4842 2018 0x4842 4018 0x4AE2 B018	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Modem status register. UART mode only. This register provides information about the current state of the control lines from the modem, data set, or peripheral device to the LH. It also indicates when a control input from the modem changes state.		
Type	R		

31:8	7	6	5	4	3	2	1	0
RESERVED	NCD_STS	NRI_STS	NDSR_STS	NCTS_STS	DCD_STS	RI_STS	DSR_STS	CTS_STS

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	NCD_STS	This bit is the complement of the DCD* input. In loopback mode, it is equivalent to UART_MCR[3].	R	-

Bits	Field Name	Description	Type	Reset
6	NRI_STS	This bit is the complement of the RI* input. In loopback mode, it is equivalent to UART_MCR[2].	R	-
5	NDSR_STS	This bit is the complement of the DSR* input. In loopback mode, it is equivalent to UART_MCR[0].	R	-
4	NCTS_STS	This bit is the complement of the CTS* input. In loopback mode, it is equivalent to UART_MCR[1].	R	-
3	DCD_STS	Indicates that DCD* input (or UART_MCR[3] in loopback) changed. Cleared on a read.	R	0
2	RI_STS	Indicates that RI* input (or UART_MCR[2] in loopback) changed state from low to high. Cleared on a read.	R	0
1	DSR_STS	Read 0x1: Indicates that DSR* input (or UART_MCR[0] in loopback) changed state. Cleared on a read.	R	0
0	CTS_STS	Read 0x1: Indicates that CTS* input (or UART_MCR[1] in loopback) changed state. Cleared on a read.	R	0

Table 24-165. UART_SPR

Address Offset	0x0000 001C		
Physical Address	0x4806 A01C 0x4806 C01C 0x4802 001C 0x4806 E01C 0x4806 601C 0x4806 801C 0x4842 001C 0x4842 201C 0x4842 401C 0x4AE2 B01C	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Scratchpad register This read/write register does not control the module. It is a scratchpad register to be used by the programmer to hold temporary data.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SPR_WORD							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	SPR_WORD	Scratchpad register	RW	0x00

Table 24-166. UART_TLR

Address Offset	0x0000 001C		
Physical Address	0x4806 A01C 0x4806 C01C 0x4802 001C 0x4806 E01C 0x4806 601C 0x4806 801C 0x4842 001C 0x4842 201C 0x4842 401C 0x4AE2 B01C	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Trigger level register This register stores the programmable transmit and RX FIFO trigger levels for DMA and IRQ generation.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RX_FIFO_TRIG_DMA	TX_FIFO_TRIG_DMA

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:4	RX_FIFO_TRIG_DMA	Receive FIFO trigger level	RW	0x0
3:0	TX_FIFO_TRIG_DMA	Transmit FIFO trigger level	RW	0x0

Table 24-167. UART_XOFF2

Address Offset	0x0000 001C		
Physical Address	0x4806 A01C 0x4806 C01C 0x4802 001C 0x4806 E01C 0x4806 601C 0x4806 801C 0x4842 001C 0x4842 201C 0x4842 401C 0x4AE2 B01C	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	UART mode XOFF2 character		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XOFF_WORD2							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XOFF_WORD2	Stores the 8-bit XOFF2 character used in UART modes.	RW	0x00

Table 24-168. UART_MDR1

Address Offset	0x0000 0020		
Physical Address	0x4806 A020 0x4806 C020 0x4802 0020 0x4806 E020 0x4806 6020 0x4806 8020 0x4842 0020 0x4842 2020 0x4842 4020 0x4AE2 B020	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Mode definition register 1 The mode of operation can be programmed by writing to MDR1[2:0] and therefore the MDR1 must be programmed on startup after configuration of the configuration registers (UART_DLL, UART_DLH, and UART_LCR). The value of MDR1[2:0] must not be changed again during normal operation. **Note:** If the module is disabled by setting the MODE_SELECT field to 111, interrupt requests can still be generated unless disabled through the interrupt enable register (UART_IER). In this case, UART mode interrupts are visible. Reading the interrupt identification register (UART_IIR) shows UART mode interrupt flags.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FRAME_END_MODE	SIP_MODE	SCT	SET_TXIR	IR_SLEEP	MODE_SELECT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	FRAME_END_MODE	IrDA mode only 0x0: Frame-length method 0x1: Set EOT bit method	RW	0
6	SIP_MODE	MIR/FIR modes only 0x0: Manual SIP mode: SIP is generated with the control of UART_ACREG[3]. 0x1: Automatic SIP mode: SIP is generated after each transmission.	RW	0
5	SCT	Store and control the transmission. 0x0: Starts the infrared transmission when a value is written to UART_THR 0x1: Starts the infrared transmission with the control of UART_ACREG[2]. **Note:** Before starting any transmission, there must be no reception ongoing.	RW	0
4	SET_TXIR	Used to configure the infrared transceiver 0x0: a) No action if UART_MDR2[7] = 0 b) TXIR pin output is forced low if UART_MDR2[7] = 1. 0x1: IRTX pin output is forced high (not dependent on UART_MDR2[7] value).	RW	0
3	IR_SLEEP	0x0: IrDA/CIR sleep mode disabled 0x1: IrDA/CIR sleep mode enabled	RW	0
2:0	MODE_SELECT	0x0: UART 16x mode 0x1: SIR mode 0x2: UART 16x auto-baud 0x3: UART 13x mode 0x4: MIR mode 0x5: FIR mode 0x6: CIR mode 0x7: Disable (default state)	RW	0x7

Table 24-169. UART_MDR2

Address Offset	0x0000 0024		
Physical Address	0x4806 A024 0x4806 C024 0x4802 0024 0x4806 E024 0x4806 6024 0x4806 8024 0x4842 0024 0x4842 2024 0x4842 4024 0x4AE2 B024	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10

Table 24-169. UART_MDR2 (continued)

Description	Mode definition register 2 IR-IrDA and IR-CIR modes only. UART_MDR2[0] describes the status of the interrupt in UART_IIR[5]. The IRTX_UNDERRUN bit should be read after an UART_IIR[5] TX_STATUS_IT interrupt. The bits [2:1] of this register set the trigger level for the frame status FIFO (8 entries) and must be programmed before the mode is programmed in UART_MDR1[2:0]. **Note:** The UART_MDR2[6] gives the flexibility to invert the RX pin in the UART to ensure that the protocol at the input of the transceiver module has the same polarity at module level. By default, the RX pin is inverted because most transceivers invert the IR receive pin.
Type	RW

31:8	7	6	5:4	3	2:1	0
RESERVED	SET_TXIR_ALT	IRRXINVERT	CIR_PULSE_MODE	UART_PULSE	STS_FIFO_TRIG	IRTX_UNDERRUN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	SET_TXIR_ALT	Provide alternate function for UART_MDR1[4] (SET_TXIR). 0x0: Normal mode 0x1: Alternate mode for SET_TXIR	R	0
6	IRRXINVERT	IR mode only (IrDA and CIR). Invert RX pin in the module before the voting or sampling system logic of the infrared block. This does not affect the RX path in UART modem modes. 0x0: Inversion is performed. 0x1: No inversion is performed.	RW	0
5:4	CIR_PULSE_MODE	CIR pulse modulation definition. Defines high level of the pulse width associated with a digit: 0x0: Pulse width of 3 from 12 cycles 0x1: Pulse width of 4 from 12 cycles 0x2: Pulse width of 5 from 12 cycles 0x3: Pulse width of 6 from 12 cycles	RW	0x0
3	UART_PULSE	UART mode only. Allows pulse shaping in UART mode. 0x0: Normal UART mode 0x1: UART mode with a pulse shaping	RW	0
2:1	STS_FIFO_TRIG	IR-IrDA mode only. Frame status FIFO threshold select: 0x0: 1 entry 0x1: 4 entries 0x2: 7 entries 0x3: 8 entries	RW	0x0

Bits	Field Name	Description	Type	Reset
0	IRTX_UNDERRUN	IrDA transmission status interrupt. When the UART_IIR[5] interrupt occurs, the meaning of the interrupt is: Read 0x0: The last bit of the frame transmitted successfully without error. Read 0x1: An underrun occurred. The last bit of the frame was transmitted but with an underrun error. The bit is reset to 0 when the UART_RESUME register is read.	R	0

Table 24-170. UART_SFLSR

Address Offset	0x0000 0028		
Physical Address	0x4806 A028 0x4806 C028 0x4802 0028 0x4806 E028 0x4806 6028 0x4806 8028 0x4842 0028 0x4842 2028 0x4842 4028 0x4AE2 B028	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Status FIFO line status register IrDA modes only. Reading this register effectively reads frame status information from the status FIFO (this register does not physically exist). Reading this register increments the status FIFO read pointer (UART_SFREGL and UART_SFREGH must be read first).		
Type	R		

31:8	7:5	4	3	2	1	0
RESERVED	RESERVED	OE_ERROR	FRAME_TOO_LONG_ERROR	ABORT_DETECT	CRC_ERROR	RESERVED

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:5	RESERVED	Read returns 0.	R	0x0
4	OE_ERROR	Read 0x1: Overrun error in RX FIFO when frame at top of RX FIFO was received **Note:** Top of RX FIFO = Next frame to be read from RX FIFO	R	-
3	FRAME_TOO_LONG_ERROR	Read 0x1: Frame-length too long error in frame at top of RX FIFO	R	-
2	ABORT_DETECT	Read 0x1: Abort pattern detected in frame at top of RX FIFO	R	-
1	CRC_ERROR	Read 0x1: CRC error in frame at top of RX FIFO	R	-
0	RESERVED		R	0

Table 24-171. UART_TXFLL

Address Offset	0x0000 0028

Table 24-171. UART_TXFLL (continued)

Physical Address	0x4806 A028 0x4806 C028 0x4802 0028 0x4806 E028 0x4806 6028 0x4806 8028 0x4842 0028 0x4842 2028 0x4842 4028 0x4AE2 B028	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Transmit frame length register low IrDA modes only. The UART_TXFLL and UART_TXFLH registers hold the 13-bit transmit frame length (expressed in bytes). UART_TXFLL holds the LSBs and UART_TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used.		
Type	W		

31:8	7:0
RESERVED	TXFLL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:0	TXFLL	LSB register used to specify the frame length	W	0x00

Table 24-172. UART_RESUME

Address Offset	0x0000 002C		
Physical Address	0x4806 A02C 0x4806 C02C 0x4802 002C 0x4806 E02C 0x4806 602C 0x4806 802C 0x4842 002C 0x4842 202C 0x4842 402C 0x4AE2 B02C	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	IR-IrDA and IR-CIR modes only. This register is used to clear internal flags, which halt transmission/ reception when an underrun/overrun error occurs. Reading this register resumes the halted operation. This register does not physically exist and reads always as 0x00.		
Type	R		

31:8	7:0
RESERVED	RESUME

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	RESUME	Dummy read to restart the TX or RX	R	0x00

Table 24-173. UART_TXFLH

Address Offset	0x0000 002C

Table 24-173. UART_TXFLH (continued)

Physical Address	0x4806 A02C 0x4806 C02C 0x4802 002C 0x4806 E02C 0x4806 602C 0x4806 802C 0x4842 002C 0x4842 202C 0x4842 402C 0x4AE2 B02C	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Transmit frame length register high IrDA modes only. The UART_TXFLL and UART_TXFLH registers hold the 13-bit transmit frame length (expressed in bytes). UART_TXFLL holds the LSBs and UART_TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used.		
Type	W		

31–8	7–5	4–0
RESERVED	RESERVED	TXFLH

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:5	RESERVED	Write has no effect.	W	0x0
4:0	TXFLH	MSB register used to specify the frame length	W	0x00

Table 24-174. UART_SFREGL

Address Offset	0x0000 0030		
Physical Address	0x4806 A030 0x4806 C030 0x4802 0030 0x4806 E030 0x4806 6030 0x4806 8030 0x4842 0030 0x4842 2030 0x4842 4030 0x4AE2 B030	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Status FIFO register low IrDA modes only. The frame lengths of received frames are written into the status FIFO. This information can be read by reading the UART_SFREGL and UART_SFREGH registers (these registers do not physically exist). The LSBs are read from UART_SFREGL and the MSBs are read from UART_SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers should be read before the pointer is incremented by reading the UART_SFLSR register.		
Type	R		

31–8	7–0
RESERVED	SFREGL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	SFREGL	LSB part of the frame length	R	0x-

Table 24-175. UART_RXFLL

Address Offset	0x0000 0030

Table 24-175. UART_RXFLL (continued)

Physical Address	0x4806 A030 0x4806 C030 0x4802 0030 0x4806 E030 0x4806 6030 0x4806 8030 0x4842 0030 0x4842 2030 0x4842 4030 0x4AE2 B030	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Received frame length register low IrDA modes only. The UART_RXFLL and UART_RXFLH registers hold the 12-bit receive maximum frame length. UART_RXFLL holds the LSBs and UART_RXFLH holds the MSBs. If the intended maximum receive frame length is n bytes, program the UART_RXFLL and UART_RXFLH registers to be n + 3 in SIR or MIR modes and n + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; 2 bytes are associated with the FIR stop flag).		
Type	W		

31–8	7–0
RESERVED	RXFLL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:0	RXFLL	LSB register used to specify the frame length in reception	W	0x00

Table 24-176. UART_SFREGH

Address Offset	0x0000 0034		
Physical Address	0x4806 A034 0x4806 C034 0x4802 0034 0x4806 E034 0x4806 6034 0x4806 8034 0x4842 0034 0x4842 2034 0x4842 4034 0x4AE2 B034	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Status FIFO register high IrDA modes only. The frame lengths of received frames are written into the status FIFO. This information can be read by reading the UART_SFREGL and UART_SFREGH registers (these registers do not physically exist). The LSBs are read from UART_SFREGL and the MSBs are read from UART_SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers should be read before the pointer is incremented by reading the UART_SFLSR register.		
Type	R		

31–8	7–4	3–0
RESERVED	RESERVED	SFREGH

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:4	RESERVED	Read returns 0.	R	0x0
3:0	SFREGH	MSB part of the frame length	R	0x-

Table 24-177. UART_RXFLH

Address Offset	0x0000 0034

Table 24-177. UART_RXFLH (continued)

Physical Address	0x4806 A034 0x4806 C034 0x4802 0034 0x4806 E034 0x4806 6034 0x4806 8034 0x4842 0034 0x4842 2034 0x4842 4034 0x4AE2 B034	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Received frame length register high IrDA modes only. The UART_RXFLL and UART_RXFLH registers hold the 12-bit receive maximum frame length. UART_RXFLL holds the LSBs and UART_RXFLH holds the MSBs. If the intended maximum receive frame length is n bytes, program the UART_RXFLL and UART_RXFLH to be n + 3 in SIR or MIR modes and n + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; 2 bytes are associated with the FIR stop flag).		
Type	W		

31–8	7–4	3–0
RESERVED	RESERVED	RXFLH

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:4	RESERVED	Write has no effect.	W	0x0
3:0	RXFLH	MSB register used to specify the frame length in reception	W	0x0

Table 24-178. UART_BLR

Address Offset	0x0000 0038		
Physical Address	0x4806 A038 0x4806 C038 0x4802 0038 0x4806 E038 0x4806 6038 0x4806 8038 0x4842 0038 0x4842 2038 0x4842 4038 0x4AE2 B038	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	BOF control register IrDA modes only. The UART_BLR[6] bit selects whether 0xC0 or 0xFF start patterns are to be used, when multiple start flags are required in SIR mode. If only one start flag is required, this is always 0xC0. If n start flags are required, (-1) 0xC0 or (-1) 0xFF flags are sent, followed by a single 0xC0 flag (immediately preceding the first data byte).		
Type	RW		

31–8	7	6	5–0
RESERVED	STS_FIFO_RESET	XBOF_TYPE	RESERVED

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	STS_FIFO_RESET	Status FIFO reset. This bit is self-clearing.	RW	0

Bits	Field Name	Description	Type	Reset
6	XBOF_TYPE	SIR xBOF select 0x0: 0xFF 0x1: 0xC0	RW	1
5:0	RESERVED	Read returns 0.	R	0x00

Table 24-179. UART_UASR

Address Offset	0x0000 0038		
Physical Address	0x4806 A038 0x4806 C038 0x4802 0038 0x4806 E038 0x4806 6038 0x4806 8038 0x4842 0038 0x4842 2038 0x4842 4038 0x4AE2 B038	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	UART autobauding status register UART autobauding mode only. This status register returns the speed, the number of bits by characters, and the type of the parity in UART autobauding mode. In autobauding mode, the input frequency of the UART modem must be fixed to 48 MHz. Any other module clock frequency results in incorrect baud rate recognition. **Note:** When the UART is in autobauding mode, this register, instead of the UART_LCR, UART_DLL, and UART_DLH registers, is used to set up transmission according to the characteristics of the previous reception. To reset the autobauding hardware (to start a new AT detection), set UART_MDR1[2:0] to 111 (reset value), then set UART_MDR1[2:1] to 010 (UART in autobaud mode). To set the UART to standard mode (no autobaud), set UART_MDR1[2:1] to 000.		
Type	R		

31:8	7:6	5	4:0
RESERVED	PARITY_TYPE	BIT_BY_CHAR	SPEED

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:6	PARITY_TYPE	Read 0x0: No parity identified Read 0x1: Parity space Read 0x2: Even parity Read 0x3: Odd parity	R	0x0
5	BIT_BY_CHAR	Read 0x0: 7-bit character identified Read 0x1: 8-bit character identified	R	0
4:0	SPEED	Used to report the speed identified Read 0x0: No speed identified Read 0x1: 115,200 baud Read 0x2: 57,600 baud Read 0x3: 38,400 baud Read 0x4: 28,800 baud Read 0x5: 19,200 baud Read 0x6: 14,400 baud Read 0x7: 9,600 baud	R	0x00

Bits	Field Name	Description	Type	Reset
		Read 0x8: 4,800 baud Read 0x9: 2,400 baud Read 0xA: 1,200 baud		

Table 24-180. UART_ACREG

Address Offset	0x0000 003C		
Physical Address	0x4806 A03C 0x4806 C03C 0x4802 003C 0x4806 E03C 0x4806 603C 0x4806 803C 0x4842 003C 0x4842 203C 0x4842 403C 0x4AE2 B03C	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Auxiliary control register. IR-IrDA and IR-CIR modes only.		
Type	RW		

31–8	7	6	5	4	3	2	1	0
RESERVED	PULSE_TYPE	SD_MOD	DIS_IR_RX	DIS_TX_UNDERRUN	SEND_SIP	SCTX_EN	ABORT_EN	EOT_EN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	PULSE_TYPE	SIR pulse width select 0x0: 3/16 of baud-rate pulse width 0x1: 1.6 μs	RW	0
6	SD_MOD	Primary output used to configure transceivers. Connected to the SD/MODE input pin of IrDA transceivers. 0x0: SD pin is set to high. 0x1: SD pin is set to low.	RW	0
5	DIS_IR_RX	0x0: Normal operation (RX input automatically disabled during transmit but enabled outside of transmit operation) 0x1: Disables RX input (permanent state - independent of transmit)	RW	0
4	DIS_TX_UNDERRUN	It is recommended to disable TX FIFO underrun capability by masking corresponding underrun interrupt. When disabling underrun by setting UART_ACREG[4] = 1, garbage data is sent over TX line. 0x0: Long stop-bits cannot be transmitted; TX underrun is enabled. 0x1: Long stop-bits can be transmitted; TX underrun is disabled.	RW	0

Bits	Field Name	Description	Type	Reset
3	SEND_SIP	MIR/FIR modes only. Send serial infrared interaction pulse (SIP). If this bit is set during an MIR/FIR transmission, the SIP is sent at the end of it. This bit is cleared automatically at the end of the SIP transmission. 0x0: No action 0x1: Send SIP pulse.	RW	0
2	SCTX_EN	Store and controlled TX start. When UART_MDR1[5] = 1 and the LH writes 1 to this bit, the TX state-machine starts frame transmission. This bit is self-clearing.	RW	0
1	ABORT_EN	Frame abort. The LH can intentionally abort transmission of a frame by writing 1 to this bit. Neither the end flag nor the CRC bits are appended to the frame. If TX FIFO is not empty and UART_MDR1[5] = 1, UART IrDA starts a new transfer with data of the previous frame when the abort frame is sent. Therefore, TX FIFO must be reset before sending an abort frame.	RW	0
0	EOT_EN	EOT (end of transmission) bit. The LH writes 1 to this bit just before it writes the last byte to the TX FIFO in set-EOT bit frame closing method. This bit is cleared automatically when the LH writes to the THR (TX FIFO).	RW	0

Table 24-181. UART_SCR

Address Offset	0x0000 0040		
Physical Address	0x4806 A040	**Instance**	UART1
	0x4806 C040		UART2
	0x4802 0040		UART3
	0x4806 E040		UART4
	0x4806 6040		UART5
	0x4806 8040		UART6
	0x4842 0040		UART7
	0x4842 2040		UART8
	0x4842 4040		UART9
	0x4AE2 B040		UART10
Description	Supplementary control register **Note:** Bit 4 enables the wake-up interrupt, but this interrupt is not mapped into the UART_IIR register. Therefore, when an interrupt occurs and there is no interrupt pending in the UART_IIR register, the UART_SSR[1] bit must be checked. To clear the wake-up interrupt, bit UART_SCR[4] must be reset to 0.		
Type	RW		

31–8	7	6	5	4	3	2–1	0
RESERVED	RX_TRIG_GRANU1	TX_TRIG_GRANU1	DSR_IT	RX_CTS_DSR_WAKE_UP_ENABLE	TX_EMPTY_CTL_IT	DMA_MODE_2	DMA_MODE_CTL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	RX_TRIG_GRANU1	0x0: Disables the granularity of 1 for trigger RX level	RW	0

Bits	Field Name	Description	Type	Reset
		0x1: Enables the granularity of 1 for trigger RX level		
6	TX_TRIG_GRANU1	0x0: Disables the granularity of 1 for trigger TX level 0x1: Enables the granularity of 1 for trigger TX level	RW	0
5	DSR_IT	0x0: Disables DSR* interrupt 0x1: Enables DSR* interrupt	RW	0
4	RX_CTS_DSR_WAKE_UP_ENABLE	0x0: Disables the wake-up interrupt and clears SSR[1] 0x1: Waits for a falling edge of pins RX, CTS*, or DSR* to generate an interrupt	RW	0
3	TX_EMPTY_CTL_IT	0x0: Normal mode for THR interrupt (see UART mode interrupts table) 0x1: The THR interrupt is generated when TX FIFO and TX shift register are empty.	RW	0
2:1	DMA_MODE_2	Used to specify the DMA mode valid if the UART_SCR[0] bit = 1 0x0: DMA mode 0 (no DMA) 0x1: DMA mode 1 (UART_nDMA_REQ[0] in TX, UART_nDMA_REQ[1] in RX) 0x2: DMA mode 2 (UART_nDMA_REQ[0] in RX) 0x3: DMA mode 3 (UART_nDMA_REQ[0] in TX)	RW	0x0
0	DMA_MODE_CTL	0x0: The DMA_MODE is set with UART_FCR[3]. 0x1: The DMA_MODE is set with UART_SCR[2:1].	RW	0

Table 24-182. UART_SSR

Address Offset	0x0000 0044		
Physical Address	0x4806 A044 0x4806 C044 0x4802 0044 0x4806 E044 0x4806 6044 0x4806 8044 0x4842 0044 0x4842 2044 0x4842 4044 0x4AE2 B044	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Supplementary status register **Note:** Bit 1 is reset only when UART_SCR[4] is reset to 0.		
Type	RW		

31:8	7:3	2	1	0
RESERVED	RESERVED	DMA_COUNTER_RST	RX_CTS_DSR_WAKE_UP_STS	TX_FIFO_FULL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000

Bits	Field Name	Description	Type	Reset
7:3	RESERVED	Read returns 0.	R	0x00
2	DMA_COUNTER_RST	0x0: The DMA counter will not be reset if the corresponding FIFO is reset (through UART_FCR[1] or UART_FCR[2]). 0x1: The DMA counter will be reset if corresponding FIFO is reset (through UART_FCR[1] or UART_FCR[2]).	RW	1
1	RX_CTS_DSR_WAKE_UP_STS	Read 0x0: No falling edge event on RX, CTS*, and DSR* Read 0x1: A falling edge occurred on RX, CTS*, or DSR*.	R	0
0	TX_FIFO_FULL	Read 0x0: TX FIFO is not full. Read 0x1: TX FIFO is full.	R	0

Table 24-183. UART_EBLR

Address Offset	0x0000 0048		
Physical Address	0x4806 A048	**Instance**	UART1
	0x4806 C048		UART2
	0x4802 0048		UART3
	0x4806 E048		UART4
	0x4806 6048		UART5
	0x4806 8048		UART6
	0x4842 0048		UART7
	0x4842 2048		UART8
	0x4842 4048		UART9
	0x4AE2 B048		UART10
Description	BOF length register IR-IrDA and IR-CIR modes only. In IR-IrDA SIR operation, this register specifies the number of BOF + xBOFs to transmit. Value set into this register must account for the BOF character; therefore, to send only one BOF with no XBOF, this register must be set to 1. To send one BOF with N XBOF, this register must be set to N + 1. The value 0 sends 1 BOF plus 255 XBOF. In IR-IrDA MIR mode, this register specifies the number of additional start flags (MIR protocol mandates a minimum of 2 start flags). In IR-CIR mode, this register specifies the number of consecutive 0s to be received before generating the RX_STOP interrupt (UART_IIR[2]). All received 0s are stored in the RX FIFO. When the register is set to 0, this feature is deactivated and always in reception state, which can be disabled by setting the UART_ACREG[5] to 1. **Note:** If the RX_STOP interrupt occurs before a byte boundary, the remaining bits of the last byte are filled with 0s and passed into the RX FIFO.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EBLR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	EBLR	IR-IrDA mode: This register allows definition of up to 176 xBOFs, the maximum required by IrDA specification. IR-CIR mode: This register specifies the number of consecutive 0s to be received before generating the RX_STOP interrupt (UART_IIR[2]). 0x00: Feature disabled 0x01: Generate RX_STOP interrupt after receiving one zero bit. ... 0xFF: Generate RX_STOP interrupt after receiving 255 zero bits.	RW	0x00

Table 24-184. UART_MVR

Address Offset	0x0000 0050

Table 24-184. UART_MVR (continued)

Physical Address	0x4806 A050 0x4806 C050 0x4802 0050 0x4806 E050 0x4806 6050 0x4806 8050 0x4842 0050 0x4842 2050 0x4842 4050 0x4AE2 B050	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Module version register The reset value is fixed by hardware and corresponds to the RTL revision of this module. A reset has no effect on the value returned.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															

Bits	Field Name	Description	Type	Reset
31:0	REV	Revision number	R	0x-- TI internal data

Table 24-185. UART_SYSC

Address Offset	0x0000 0054		
Physical Address	0x4806 A054 0x4806 C054 0x4802 0054 0x4806 E054 0x4806 6054 0x4806 8054 0x4842 0054 0x4842 2054 0x4842 4054 0x4AE2 B054	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	System configuration register The AUTOIDLE bit controls a power-saving technique to reduce the logic power consumption of the open-core protocol (OCP) interface. When the feature is enabled, the clock is gated off until an OCP command for this device is detected. When the software reset bit is set high, it causes a full device reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED			IDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:5	RESERVED	Read returns 0.	R	0x0

Bits	Field Name	Description	Type	Reset
4:3	IDLEMODE	Power management req/ack control ref: OCP Design Guidelines Version 1.1 0x0: Force-idle: Idle request is acknowledged unconditionally. 0x1: No-idle: Idle request is never acknowledged. 0x2: Smart-idle: Idle request is acknowledged based in module internal activity. 0x3: Smart-idle Wake-up: Acknowledgement to an idle request is given based in the internal activity of the module. The module is allowed to generate wake-up request.	RW	0x0
2	ENAWAKEUP	Wake-up feature control 0x0: Wakeup is disabled. 0x1: Wake-up capability is enabled.	RW	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. This bit is automatically reset by the hardware. Read returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	Internal OCP clock gating strategy 0x0: Clock is running. 0x1: Automatic OCP clock gating strategy is applied, based on OCP interface activity	RW	0

Table 24-186. UART_SYSS

Address Offset	0x0000 0058		
Physical Address	0x4806 A058 0x4806 C058 0x4802 0058 0x4806 E058 0x4806 6058 0x4806 8058 0x4842 0058 0x4842 2058 0x4842 4058 0x4AE2 B058	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	System status register		
Type	R		

31–8	7–1	0
RESERVED	RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:1	RESERVED	Read returns 0.	R	0x00
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is ongoing. Read 0x1: Reset complete	R	0

Table 24-187. UART_WER

Address Offset	0x0000 005C

Table 24-187. UART_WER (continued)

Physical Address	0x4806 A05C	**Instance**	UART1
	0x4806 C05C		UART2
	0x4802 005C		UART3
	0x4806 E05C		UART4
	0x4806 605C		UART5
	0x4806 805C		UART6
	0x4842 005C		UART7
	0x4842 205C		UART8
	0x4842 405C		UART9
	0x4AE2 B05C		UART10
Description	Wake-up enable register The UART wake-up enable register is used to mask and unmask a UART event that would subsequently notify the system. An event is any activity in the logic that could cause an interrupt and/or an activity that would require the system to wake up. Even if the wakeup is disabled for certain events, if these events are also an interrupt to the UART, the UART registers the interrupt.		
Type	RW		

31:8	7	6	5	4	3	2	1	0
RESERVED	TX_WAKEUP_EN	EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT	EVENT_5_RHR_INTERRUPT	EVENT_4_RX_ACTIVITY	EVENT_3_DCD_CD_ACTIVITY	EVENT_2_RI_ACTIVITY	EVENT_1_DSR_ACTIVITY	EVENT_0_CTS_ACTIVITY

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	TX_WAKEUP_EN	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system: it can be THR_IT or TX_DMA request and/or TX_STATUS_IT.	RW	1
6	EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
5	EVENT_5_RHR_INTERRUPT	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
4	EVENT_4_RX_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
3	EVENT_3_DCD_CD_ACTIVITY	0x0: Event is not allowed to wake up the system 0x1: Event can wake up the system	RW	1
2	EVENT_2_RI_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
1	EVENT_1_DSR_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1

Bits	Field Name	Description	Type	Reset
0	EVENT_0_CTS_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1

Table 24-188. UART_CFPS

Address Offset	0x0000 0060		
Physical Address	0x4806 A060 0x4806 C060 0x4802 0060 0x4806 E060 0x4806 6060 0x4806 8060 0x4842 0060 0x4842 2060 0x4842 4060 0x4AE2 B060	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Carrier frequency prescaler Because the consumer IR works at modulation rates of 30 to 56.8 kHz, the 48-MHz clock must be prescaled before the clock can drive the IR logic. This register sets the divisor rate to give a range to accommodate the remote-control requirements in baud multiples of 12x. The value of the CFPS at reset is 0105 decimal, which equals 38.1 kHz output from starting conditions. The 48-MHz carrier is prescaled by the CFPS, which is then divided by the 12x baud multiple.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CFPS							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	CFPS	System clock frequency prescaler at (12x multiple). Examples for CFPS values: **Target Freq (kHz)** / **CFPS (decimal)** / **Actual Freq (kHz)** 30 / 133 / 30.08 32.75 / 122 / 32.79 36 / 111 / 36.04 36.7 / 109 / 36.69 38* / 105 / 38.1 40 / 100 / 40 56.8 / 70 / 57.14 *configured at reset to this value **Note:** CFPS = 0 is not supported.	RW	0x69

Table 24-189. UART_RXFIFO_LVL

Address Offset	0x0000 0064		
Physical Address	0x4806 A064 0x4806 C064 0x4802 0064 0x4806 E064 0x4806 6064 0x4806 8064 0x4842 0064 0x4842 2064 0x4842 4064 0x4AE2 B064	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Level of the RX FIFO		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RXFIFO_LVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	RXFIFO_LVL	Shows the number of received bytes in the RX FIFO	R	0x00

Table 24-190. UART_TXFIFO_LVL

Address Offset	0x0000 0068		
Physical Address	0x4806 A068 0x4806 C068 0x4802 0068 0x4806 E068 0x4806 6068 0x4806 8068 0x4842 0068 0x4842 2068 0x4842 4068 0x4AE2 B068	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Level of the TX FIFO		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TXFIFO_LVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	TXFIFO_LVL	Shows the number of written bytes in the TX FIFO	R	0x00

Table 24-191. UART_IER2

Address Offset	0x0000 006C		
Physical Address	0x4806 A06C 0x4806 C06C 0x4802 006C 0x4806 E06C 0x4806 606C 0x4806 806C 0x4842 006C 0x4842 206C 0x4842 406C 0x4AE2 B06C	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Enables RX/TX FIFOs empty corresponding interrupts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EN_TXFIFO_EMPTY	EN_RXFIFO_EMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000

Bits	Field Name	Description	Type	Reset
1	EN_TXFIFO_EMPTY	Enables TX FIFO empty corresponding interrupt 0x0: Disables EN_TXFIFO_EMPTY interrupt 0x1: Enables EN_TXFIFO_EMPTY interrupt	RW	0
0	EN_RXFIFO_EMPTY	Enables RX FIFO empty corresponding interrupt 0x0: Disables EN_RXFIFO_EMPTY interrupt 0x1: Enables EN_RXFIFO_EMPTY interrupt	RW	0

Table 24-192. UART_ISR2

Address Offset	0x0000 0070		
Physical Address	0x4806 A070 0x4806 C070 0x4802 0070 0x4806 E070 0x4806 6070 0x4806 8070 0x4842 0070 0x4842 2070 0x4842 4070 0x4AE2 B070	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Status of RX/TX FIFOs empty corresponding interrupts		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2	1	0
RESERVED	TXFIFO_EMPTY_STS	RXFIFO_EMPTY_STS

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
1	TXFIFO_EMPTY_STS	Used to generate interrupt if the TX_FIFO is empty (software flow control) 0x0: TXFIFO_EMPTY interrupt not pending. 0x1: TXFIFO_EMPTY interrupt pending.	RW	1
0	RXFIFO_EMPTY_STS	Used to generate interrupt if the RX_FIFO is empty (software flow control) 0x0: RXFIFO_EMPTY interrupt not pending. 0x1: RXFIFO_EMPTY interrupt pending.	RW	1

Table 24-193. UART_FREQ_SEL

Address Offset	0x0000 0074		
Physical Address	0x4806 A074 0x4806 C074 0x4802 0074 0x4806 E074 0x4806 6074 0x4806 8074 0x4842 0074 0x4842 2074 0x4842 4074 0x4AE2 B074	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Sample per bit selector		

Table 24-193. UART_FREQ_SEL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FREQ_SEL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
7:0	FREQ_SEL	Sets the sample per bit if nondefault frequency is used. UART_MDR3[1] must be set to 1 after this value is set. Must be equal to or higher than 6.	RW	0x1A

Table 24-194. UART_MDR3

Address Offset	0x0000 0080		
Physical Address	0x4806 A080 0x4806 C080 0x4802 0080 0x4806 E080 0x4806 6080 0x4806 8080 0x4842 0080 0x4842 2080 0x4842 4080 0x4AE2 B080	**Instance**	UART1 UART2 UART3 UART4 UART5 UART6 UART7 UART8 UART9 UART10
Description	Mode definition register 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													SET_DMA_TX_THRESHOLD	NONDEFAULT_FREQ	DISABLE_CIR_RX_DEMOD

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
2	SET_DMA_TX_THRESHOLD	Enable to set different TXDMA threshold in UART_TX_DMA_THRESHOLD register.	RW	0
1	NONDEFAULT_FREQ	Used to enable the NONDEFAULT fclk frequencies. 0x0: Disables using NONDEFAULT fclk frequencies. 0x1: Enables using NONDEFAULT fclk frequencies (set UART_FREQ_SEL and UART_DLH/UART_DLL).	RW	0
0	DISABLE_CIR_RX_DEMOD	Used to enable CIR RX demodulation. 0x0: Enables CIR RX demodulation. 0x1: Disables CIR RX demodulation.	RW	0

Table 24-195. UART_TX_DMA_THRESHOLD

Address Offset	0x0000 0084

Table 24-195. UART_TX_DMA_THRESHOLD (continued)

Physical Address	0x4806 A084	**Instance**	UART1
	0x4806 C084		UART2
	0x4802 0084		UART3
	0x4806 E084		UART4
	0x4806 6084		UART5
	0x4806 8084		UART6
	0x4842 0084		UART7
	0x4842 2084		UART8
	0x4842 4084		UART9
	0x4AE2 B084		UART10
Description	Use to manually set the TX DMA threshold level. UART_MDR3[2] SET_TX_DMA_THRESHOLD must be 1 and must be value + tx_trigger_level = 64 (TX FIFO size). If not, 64-tx_trigger_level will be used without modifying the value of this register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										TX_DMA_THRESHOLD					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0x0000000
5:0	TX_DMA_THRESHOLD	Used to manually set the TX DMA threshold level	RW	0x00

24.4 Multichannel Serial Peripheral Interface

This section describes the four Multichannel Serial Peripheral Interface (McSPI) modules for the device.

24.4.1 McSPI Overview

The SPI is a master/slave synchronous serial bus. There are four separate McSPI modules (McSPI1, McSPI2, McSPI3, and McSPI4) in the device (see Figure 24-72). All these four modules support up to four external devices (four chip selects) and are able to work as both master and slave.

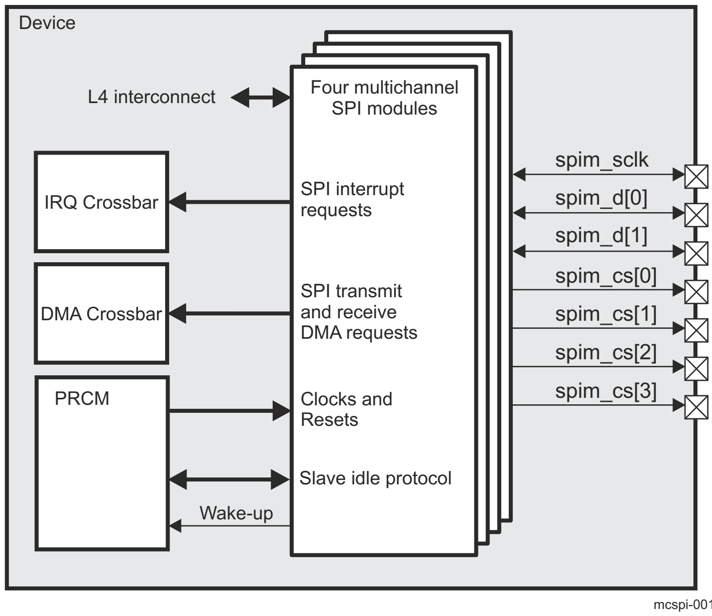


Figure 24-72. Multichannel SPI Modules

The McSPI modules include the following main features:

- Serial clock with programmable frequency, polarity, and phase for each channel
- Wide selection of SPI word lengths, ranging from 4 to 32 bits
- Up to four master channels, or single channel in slave mode
- Master multichannel mode:
  - Full duplex/half duplex
  - Transmit-only/receive-only/transmit-and-receive modes
  - Flexible input/output (I/O) port controls per channel
  - Programmable clock granularity
  - SPI configuration per channel. This means, clock definition, polarity enabling and word width
- Single interrupt line for multiple interrupt source events
- Power management through wake-up capabilities
- Enable the addition of a programmable start-bit for SPI transfer per channel (start-bit mode)
- Supports start-bit write command
- Supports start-bit pause and break sequence
- Programmable timing control between chip select and external clock generation
- Built-in FIFO available for a single channel

24.4.2 McSPI Environment

24.4.2.1 Basic McSPI Pins for Master Mode

Figure 24-73 shows all of the McSPI interface signals in master mode.

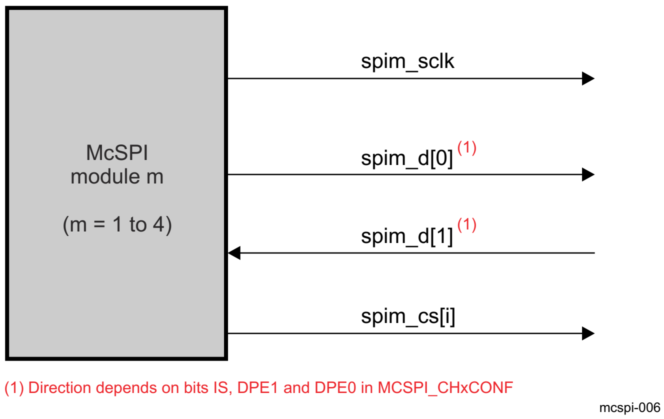


Figure 24-73. McSPI Interface Signals in Master Mode

Table 24-196 describes the McSPI I/O in master mode.

Table 24-196. McSPI I/O Description (Master Mode)

Device-Level Signal Name	Module Signal Name	I/O[1]	Description
spim_sclk	SPICLK	O	SPIm module serial clock output
spim_d[0]	SPIDAT[0]	O[2]	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[16] DPE0.
spim_d[1]	SPIDAT[1]	I[2]	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[17] DPE1.
spim_cs[i]	SPIEN[x]	O	SPIm module chip-select i output

(1) I = Input; O = Output
(2) Example configuration only.

Note

For the spim_sclk signals to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers should be set to 0x1 because of retiming purposes.

24.4.2.2 Basic McSPI Pins for Slave Mode

Figure 24-74 shows all of the McSPI interface signals in slave mode.

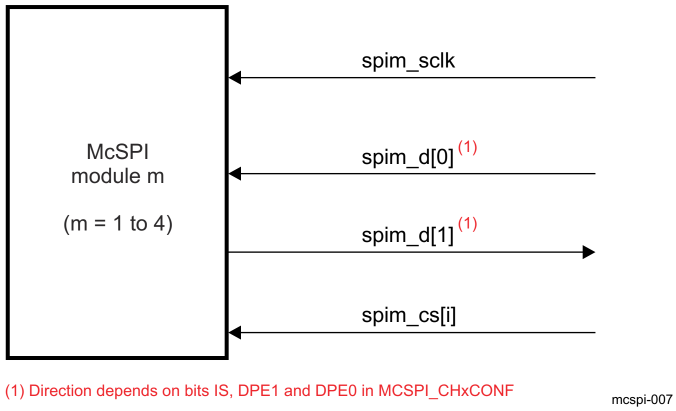


Figure 24-74. McSPI Interface Signals in Slave Mode

Table 24-197 describes the McSPI I/O in slave mode.

Table 24-197. McSPI I/O Description (Slave Mode)

Device-Level Signal Name	Module Signal Name	I/O[1]	Description
spim_sclk	SPICLK	I	McSPIm module serial clock input
spim_d[0]	SPIDAT[0]	I[2]	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[16] DPE0.
spim_d[1]	SPIDAT[1]	O[2]	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[17] DPE1.
spim_cs[i]	SPIEN[x]	I	McSPIm module chip-select i input. Can be selected through MCSPI_CH0CONF[22:21] SPIENSLV bit field

(1) I = Input; O = Output
(2) Example configuration only.

24.4.2.3 Multichannel SPI Protocol and Data Format

The synchronous SPI protocol allows a master device to initiate serial data transfers to a slave device. A slave select line (SPIEN[x]) allows selection of an individual slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities.

McSPI offers the flexibility to modify the following parameters to adapt to the device features:

- Word length

  McSPI supports any SPI word ranging from 4 bits to 32 bits long (the MCSPI_CHxCONF[11:7] WL bit field).

  SPI word length can be changed between transmissions to allow the master device to communicate with peripheral slaves that have different requirements.
- SPI enable (SPIEN[x], for channel x of instance m)

  The polarity of the SPI enable signals is programmable (the MCSPI_CHxCONF[6] EPOL bit). SPIEN[x] signals can be active high or low.

  Assertion of the SPIEN[x] signals is programmable and can be done manually or automatically. The manual assertion mode is available in single master mode only. SPIEN[x] can be kept active between words with the MCSPI_CHxCONF[20] FORCE bit.

  Two consecutive words for two different slave devices can go along with active SPIEN[x] signals with different polarity.
- Programmable start-bit

In start-bit mode a start-bit is added before the SPI word length to indicate how the next SPI word must be handled. The start-bit is enabled by setting the MCSPI_CHxCONF[23] SBE bit to 1. The MCSPI_CHxCONF[24] SBPOL bit defines the polarity of the start-bit.

- Programmable SPI clock
  - Bit rate

    In master mode, the baud rate of the SPI serial clock is programmable using the 48-MHz reference clock (from the power, reset, and clock managment [PRCM] module). Table 24-198 lists the SPICLK bit rates obtained for data transfer when programming the clock divider (the MCSPI_CHxCONF[5:2] CLKD bit field).

Table 24-198. SPI Master Clock Rates

Divider	Clock Rate
1	48 MHz[(1)]
2	24 MHz[(1)]
4	12 MHz
8	6 MHz
16	3 MHz
32	1.5 MHz
64	750 kHz
128	375 kHz
256	~187 kHz
512	~93.7 kHz
1024	~46.8 kHz
2048	~23.4 kHz
4096	~11.7 kHz

(1) These frequencies are not necessarily supported by all SPI modules. For more information, see the *Timing Requirements and Switching Characteristics* chapter in the device data manual.

  - Polarity and phase

    The polarity (the MCSPI_CHxCONF[1] POL bit) and the phase (the MCSPI_CHxCONF[0] PHA bit) of the SPI serial clock (SPICLK) are configurable to offer four combinations. Software selects the right combination, depending on the device. See Table 24-199 and Figure 24-75.

Table 24-199. Phase and Polarity Combinations

Polarity (POL)	Phase (PHA)	SPI Mode	Description
0	0	Mode 0	SPICLK is inactive low and sampling occurs at the rising edge.
0	1	Mode 1	SPICLK is inactive low and sampling occurs at the falling edge.
1	0	Mode 2	SPICLK is inactive high and sampling occurs at the falling edge.
1	1	Mode 3	SPICLK is inactive high and sampling occurs at the rising edge.

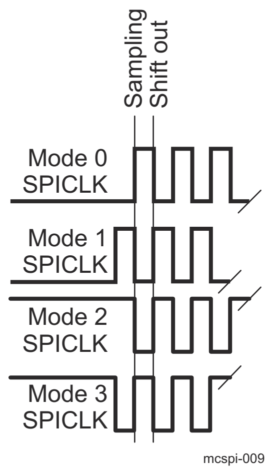


Figure 24-75. Phase and Polarity Combinations

24.4.2.3.1 Transfer Format

In master and slave modes, the McSPI drives the data lines when SPIEN[x] is asserted.

Each word is transmitted starting with the most-significant bit (MSB).

This section explains the two cases of data transmission determined by the clock phase (PHA) and the type of data transmission using a start-bit (SBE) called the start-bit mode:

- Transmission in mode 0 and mode 2 (PHA = 0)

  When PHA = 0, the first bit of the SPI word to transmit (on the master or the slave data output pin) is valid one-half cycle of SPICLK after the assertion of SPIEN.

  Therefore, the first edge of the SPICLK line is used by the master to sample the first data bit sent by the slave. On the same edge, the first data bit sent by the master is sampled by the slave.

  On the next SPICLK edge, the received data bit is shifted into the receive shift register and a new data bit is transmitted on the serial data line.

  This process continues for a number of pulses on the SPICLK line defined by the SPI word length programmed in the master device, with data being latched on odd-numbered edges and shifted on even-numbered edges. See Figure 24-76.

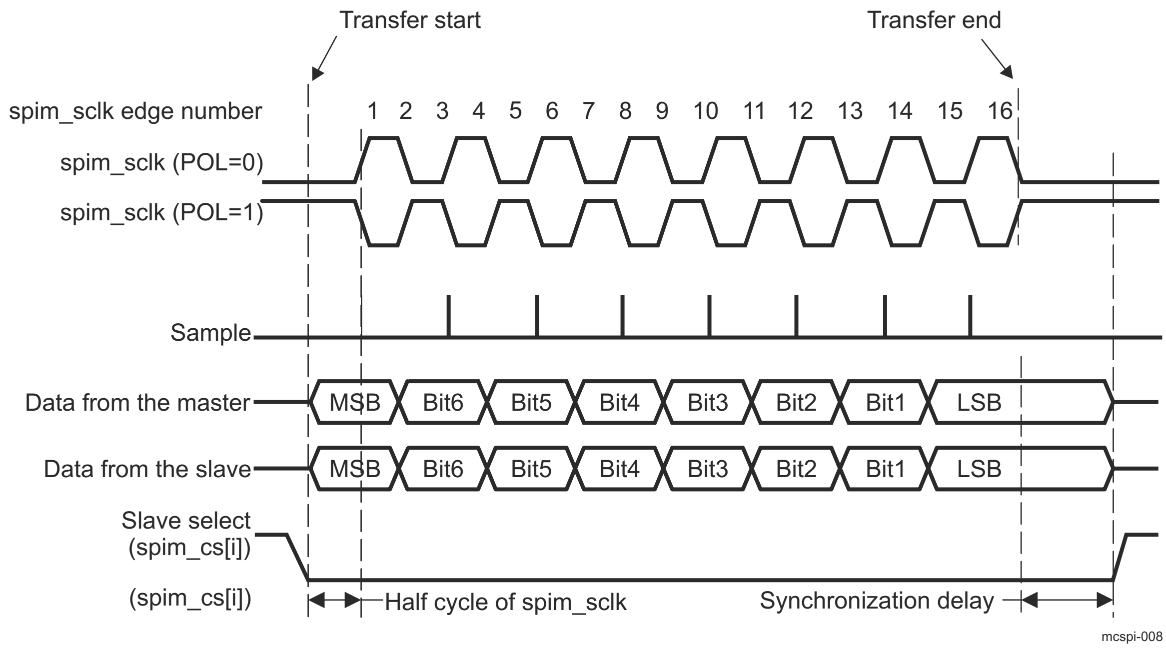


Figure 24-76. Full-Duplex Transfer Format With PHA = 0

- Transmission in mode 1 and mode 3 (PHA = 1)

  When PHA = 1, the first bit of the SPI word to transmit (on the master or the slave data output pin) is valid on the following SPICLK edge (one-half cycle later). This is the sampling edge for the master and slave. A synchronization delay is added between the activation of SPIEN[x] and the first SPICLK edge.

  The received data bit is shifted into the shift register on the third SPICLK edge.

  This process continues for a number of pulses on the SPICLK line defined by the SPI word length programmed in the master device, with data being latched on even-numbered edges and shifted on odd-numbered edges.

Note

The minimum synchronization delay is one cycle of SPICLK, if the frequency of SPICLK equals the frequency of SPIm_FCLK (McSPIm functional clock) in master mode. The minimum synchronization delay is one-half cycle of SPICLK, if the frequency of SPICLK is lower than the frequency of SPIm_FCLK in the master and slave modes.

- Transmission with a start-bit (SBE = 1)

  When the MCSPI_CHxCONF[23] SBE bit is set to 1, a start-bit is added before the MSB to indicate whether the next SPI word must be handled as a command or as data.

  Figure 24-77 shows an example of a data transfer with an extra start-bit.

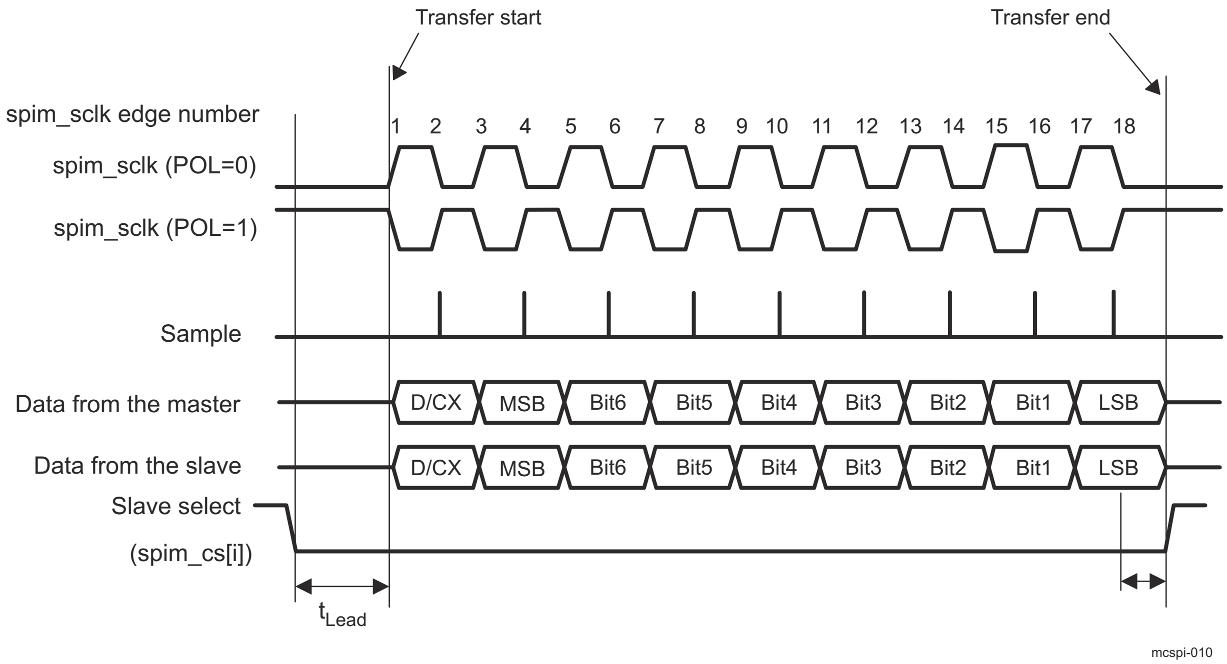


Figure 24-77. Extended SPI Transfer With a Start-Bit (SBE = 1)

24.4.2.4 SPI in Master Mode

Figure 24-78 shows a case in master mode (full-duplex) where the McSPI module is connected with two slave devices.

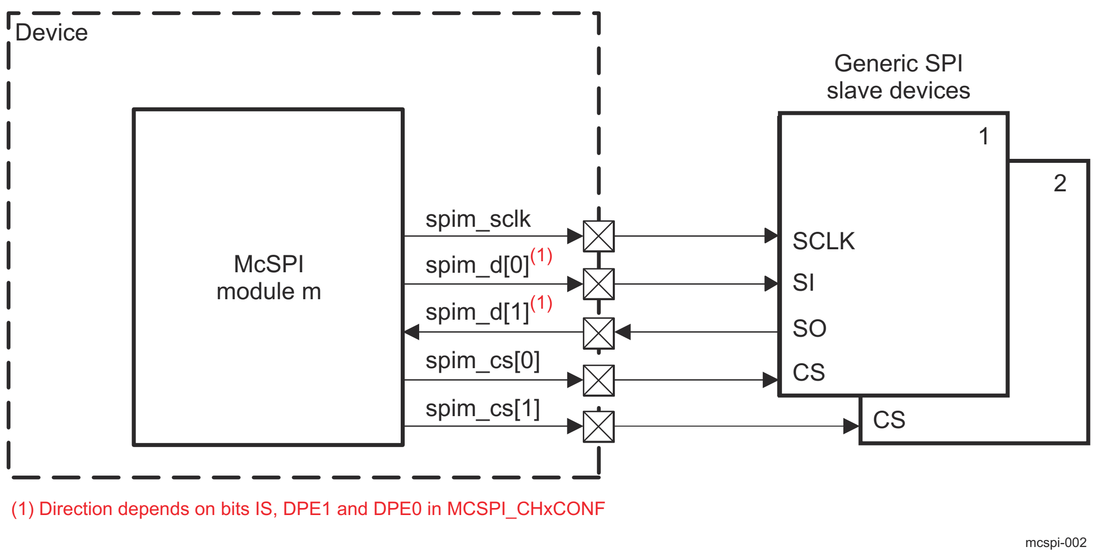


Figure 24-78. McSPI Master Mode (Full Duplex)

Figure 24-79 shows the master single mode, which can also be configured in receive-only mode.

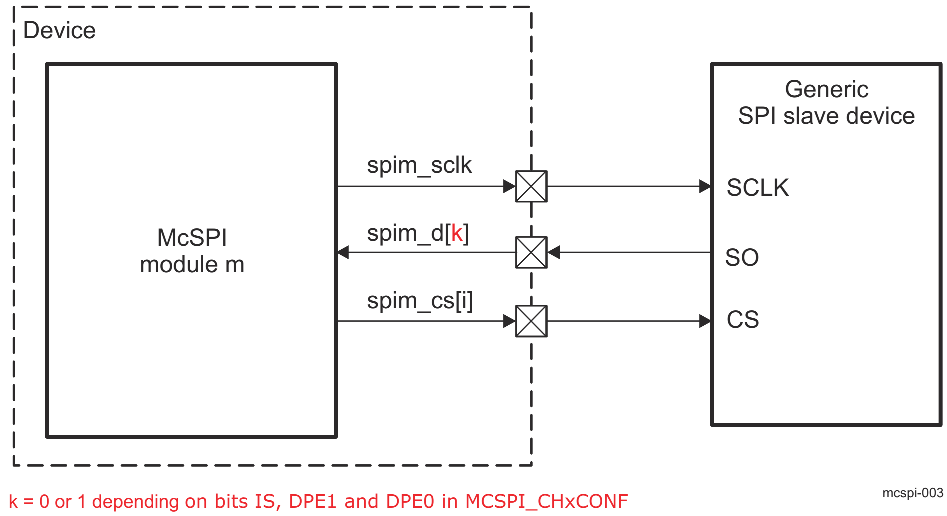


Figure 24-79. McSPI Master Single Mode (Receive Only)

24.4.2.5 SPI in Slave Mode

Figure 24-80 shows a case in slave mode (full-duplex).

Note

Only channel 0 can be configured as slave, but the chip-enable signal can be connected to any SPIEN[x] pin and then rerouted internally to channel 0 (the MCSPI_CHxCONF[22:21] SPIENSLV bit field [where x = 0]). For more information, see Section 24.4.4.4, *Slave Mode*.

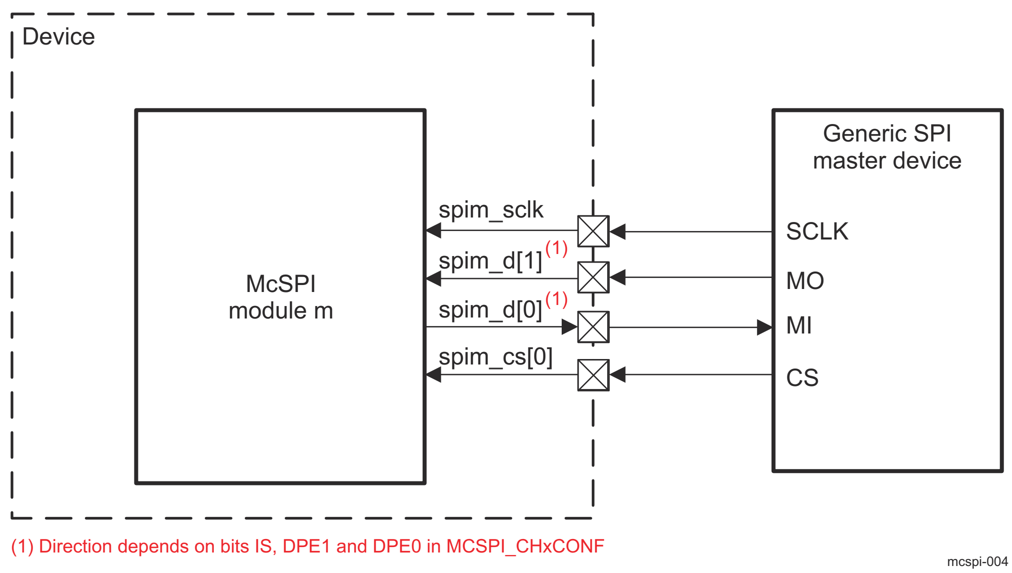


Figure 24-80. McSPI Slave Mode (Full Duplex)

Figure 24-81 shows the slave single mode, which can also be configured in transmit-only mode.

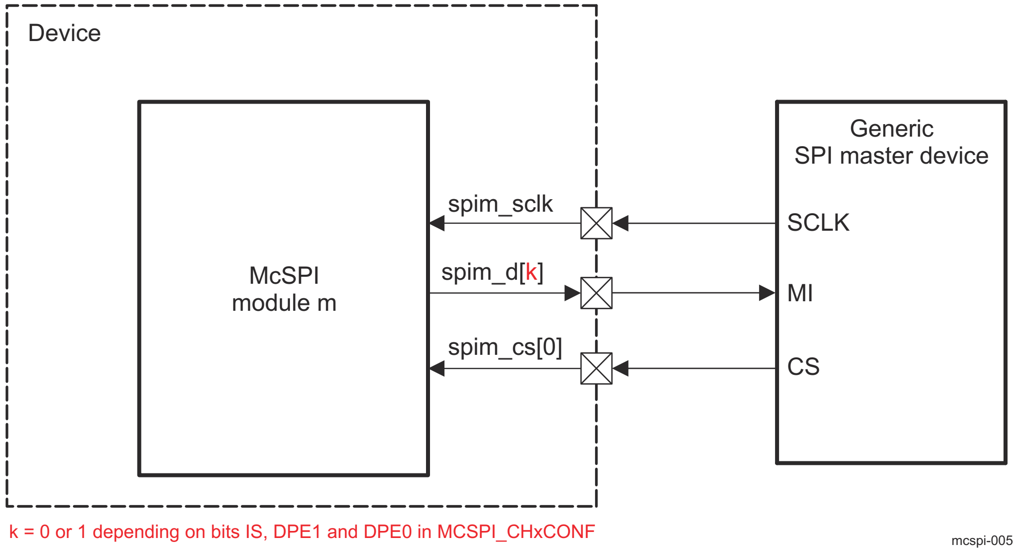


Figure 24-81. McSPI Slave Single Mode (Transmit Only)

24.4.3 McSPI Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 24-82 shows McSPI integration.

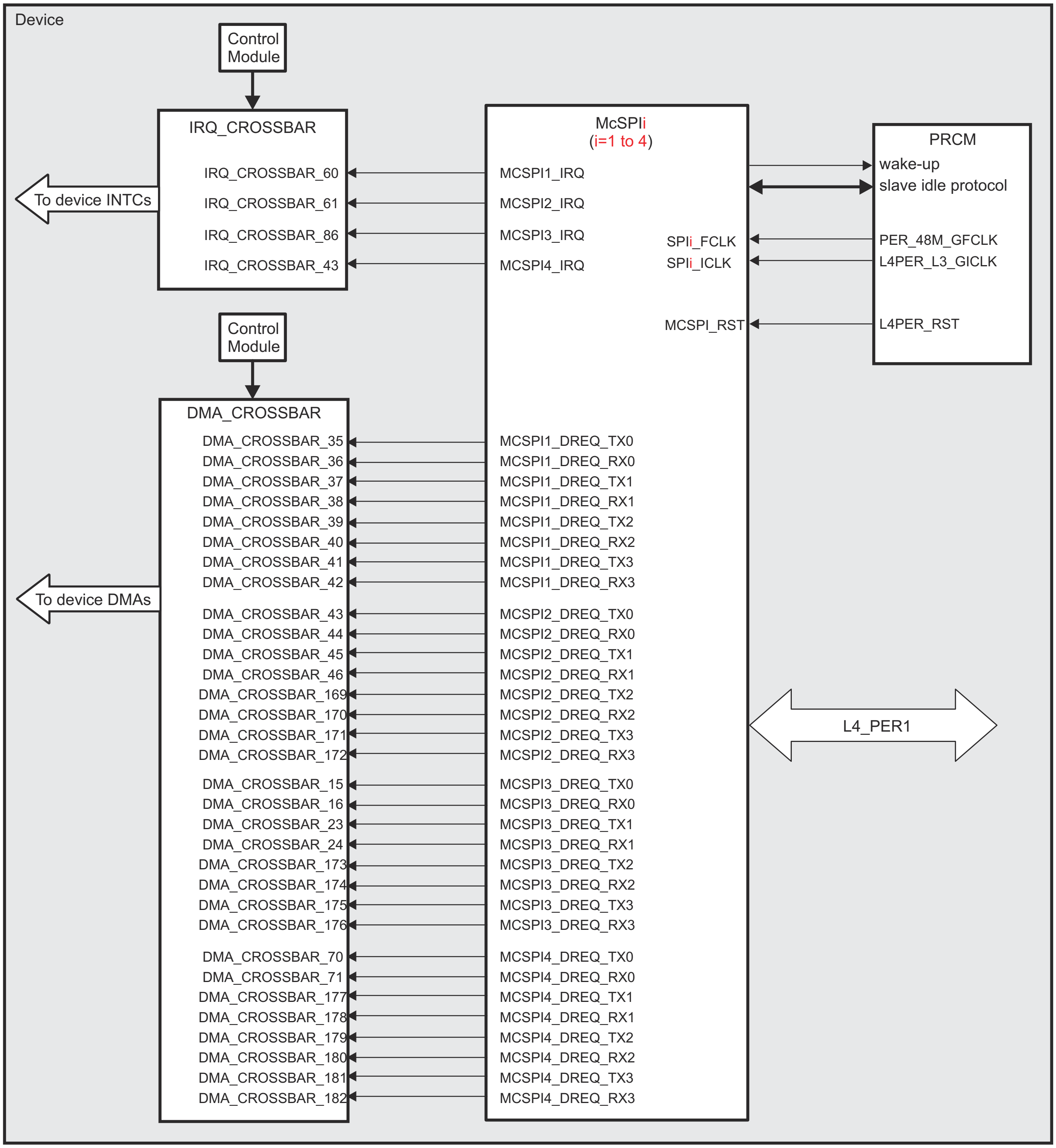


Figure 24-82. McSPI Integration

Table 24-200 through Table 24-202 summarize the integration of the module in the device.

Table 24-200. McSPI Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
McSPI1	PD_COREAON	L4_PER1
McSPI2	PD_COREAON	L4_PER1
McSPI3	PD_COREAON	L4_PER1
McSPI4	PD_COREAON	L4_PER1

Table 24-201. McSPI Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
McSPI1	SPI1_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI1_FCLK	PER_48M_GFCLK	PRCM	Functional clock
McSPI2	SPI2_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI2_FCLK	PER_48M_GFCLK	PRCM	Functional clock
McSPI3	SPI3_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI3_FCLK	PER_48M_GFCLK	PRCM	Functional clock
McSPI4	SPI4_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI4_FCLK	PER_48M_GFCLK	PRCM	Functional clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
McSPI1	MCSPI_RST	L4PER_RST	PRCM	McSPI1 reset signal
McSPI2	MCSPI_RST	L4PER_RST	PRCM	McSPI2 reset signal
McSPI3	MCSPI_RST	L4PER_RST	PRCM	McSPI3 reset signal
McSPI4	MCSPI_RST	L4PER_RST	PRCM	McSPI4 reset signal

Table 24-202. McSPI Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
McSPI1	MCSPI1_IRQ	IRQ_CROSSBAR_60	MPU_IRQ_65	McSPI module 1 interrupt request
			DSP1_IRQ_91	
			DSP2_IRQ_91	
			IPU1_IRQ_57	
			IPU2_IRQ_57	
McSPI2	MCSPI2_IRQ	IRQ_CROSSBAR_61	MPU_IRQ_66	McSPI module 2 interrupt request
			DSP1_IRQ_92	
			DSP2_IRQ_92	
			IPU1_IRQ_58	
			IPU2_IRQ_58	
McSPI3	MCSPI3_IRQ	IRQ_CROSSBAR_86	MPU_IRQ_91	McSPI module 3 interrupt request
McSPI4	MCSPI4_IRQ	IRQ_CROSSBAR_43	MPU_IRQ_48	McSPI module 4 interrupt request
			DSP1_IRQ_74	
			DSP2_IRQ_74	
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**

Table 24-202. McSPI Hardware Requests (continued)

McSPI1	MCSPI1_DREQ_TX0	DMA_CROSSBAR_35	DMA_SYSTEM_DREQ_34 DMA_EDMA_DREQ_34	McSPI module 1 - transmit request channel 0
	MCSPI1_DREQ_RX0	DMA_CROSSBAR_36	DMA_SYSTEM_DREQ_35 DMA_EDMA_DREQ_35	McSPI module 1 - receive request channel 0
	MCSPI1_DREQ_TX1	DMA_CROSSBAR_37	DMA_SYSTEM_DREQ_36 DMA_EDMA_DREQ_36	McSPI module 1 - transmit request channel 1
	MCSPI1_DREQ_RX1	DMA_CROSSBAR_38	DMA_SYSTEM_DREQ_37 DMA_EDMA_DREQ_37	McSPI module 1 - receive request channel 1
	MCSPI1_DREQ_TX2	DMA_CROSSBAR_39	DMA_SYSTEM_DREQ_38 DMA_EDMA_DREQ_38	McSPI module 1 - transmit request channel 2
	MCSPI1_DREQ_RX2	DMA_CROSSBAR_40	DMA_SYSTEM_DREQ_39 DMA_EDMA_DREQ_39	McSPI module 1 - receive request channel 2
	MCSPI1_DREQ_TX3	DMA_CROSSBAR_41	DMA_SYSTEM_DREQ_40 DMA_EDMA_DREQ_40	McSPI module 1 - transmit request channel 3
	MCSPI1_DREQ_RX3	DMA_CROSSBAR_42	DMA_SYSTEM_DREQ_41 DMA_EDMA_DREQ_41	McSPI module 1 - receive request channel 3
McSPI2	MCSPI2_DREQ_TX0	DMA_CROSSBAR_43	DMA_SYSTEM_DREQ_42 DMA_EDMA_DREQ_42	McSPI module 2 - transmit request channel 0
	MCSPI2_DREQ_RX0	DMA_CROSSBAR_44	DMA_SYSTEM_DREQ_43 DMA_EDMA_DREQ_43	McSPI module 2 - receive request channel 0
	MCSPI2_DREQ_TX1	DMA_CROSSBAR_45	DMA_SYSTEM_DREQ_44 DMA_EDMA_DREQ_44	McSPI module 2 - transmit request channel 1
	MCSPI2_DREQ_RX1	DMA_CROSSBAR_46	DMA_SYSTEM_DREQ_45 DMA_EDMA_DREQ_45	McSPI module 2 - receive request channel 1
	MCSPI2_DREQ_TX2	DMA_CROSSBAR_169	-	McSPI module 2 - transmit request channel 2
	MCSPI2_DREQ_RX2	DMA_CROSSBAR_170	-	McSPI module 2 - receive request channel 2
	MCSPI2_DREQ_TX3	DMA_CROSSBAR_171	-	McSPI module 2 - transmit request channel 3
	MCSPI2_DREQ_RX3	DMA_CROSSBAR_172	-	McSPI module 2 - receive request channel 3
McSPI3	MCSPI3_DREQ_TX0	DMA_CROSSBAR_15	DMA_SYSTEM_DREQ_14 DMA_EDMA_DREQ_14	McSPI module 3 - transmit request channel 0
	MCSPI3_DREQ_RX0	DMA_CROSSBAR_16	DMA_SYSTEM_DREQ_15 DMA_EDMA_DREQ_15	McSPI module 3 - receive request channel 0
	MCSPI3_DREQ_TX1	DMA_CROSSBAR_23	DMA_SYSTEM_DREQ_22 DMA_EDMA_DREQ_22	McSPI module 3 - transmit request channel 1
	MCSPI3_DREQ_RX1	DMA_CROSSBAR_24	DMA_SYSTEM_DREQ_23 DMA_EDMA_DREQ_23	McSPI module 3 - receive request channel 1
	MCSPI3_DREQ_TX2	DMA_CROSSBAR_173	-	McSPI module 3 - transmit request channel 2
	MCSPI3_DREQ_RX2	DMA_CROSSBAR_174	-	McSPI module 3 - receive request channel 2
	MCSPI3_DREQ_TX3	DMA_CROSSBAR_175	-	McSPI module 3 - transmit request channel 3
	MCSPI3_DREQ_RX3	DMA_CROSSBAR_176	-	McSPI module 3 - receive request channel 3

Table 24-202. McSPI Hardware Requests (continued)

McSPI4	MCSPI4_DREQ_TX0	DMA_CROSSBAR_70	DMA_SYSTEM_DREQ_69	McSPI module 4 - transmit request channel 0
	MCSPI4_DREQ_RX0	DMA_CROSSBAR_71	DMA_SYSTEM_DREQ_70	McSPI module 4 - receive request channel 0
	MCSPI4_DREQ_TX1	DMA_CROSSBAR_177	-	McSPI module 4 - transmit request channel 1
	MCSPI4_DREQ_RX1	DMA_CROSSBAR_178	-	McSPI module 4 - receive request channel 1
	MCSPI4_DREQ_TX2	DMA_CROSSBAR_179	-	McSPI module 4 - transmit request channel 2
	MCSPI4_DREQ_RX2	DMA_CROSSBAR_180	-	McSPI module 4 - receive request channel 2
	MCSPI4_DREQ_TX3	DMA_CROSSBAR_181	-	McSPI module 4 - transmit request channel 3
	MCSPI4_DREQ_RX3	DMA_CROSSBAR_182	-	McSPI module 4 - receive request channel 3

Note

The Default Mapping column in Table 24-202 *McSPI Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.

Note

For more information about the device interrupt controllers, see *Interrupt Controllers*.
For more information about the device DMA_SYSTEM module, see *System DMA*.
For more information about the device EDMA module, see *Enhanced DMA*.

24.4.4 McSPI Functional Description

24.4.4.1 McSPI Block Diagram

Figure 24-83 shows the McSPI module.

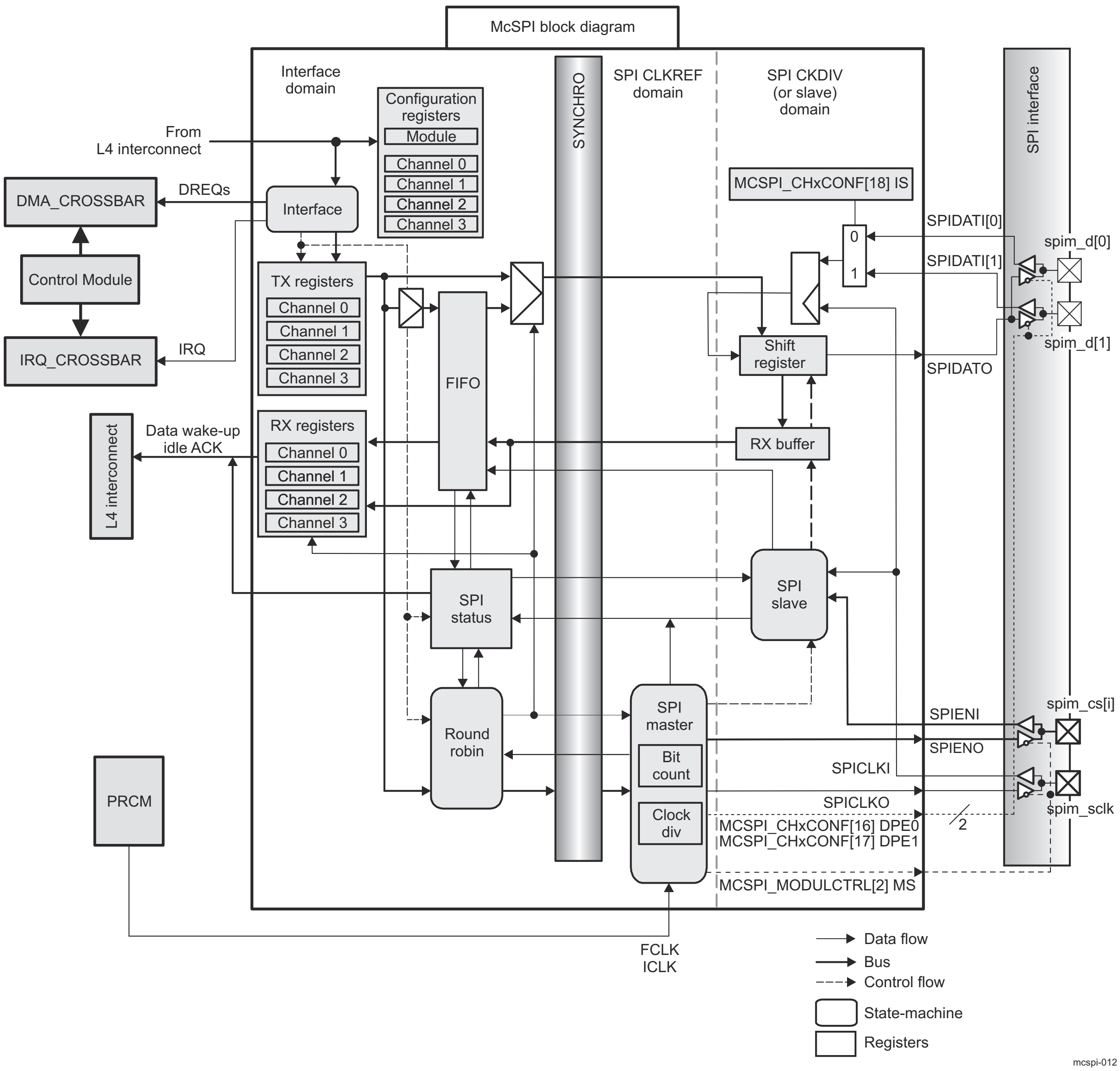


Figure 24-83. McSPI Block Diagram

24.4.4.2 Reset

The McSPI module can be reset either by hardware or by software reset. All configuration registers and all state machines are reset by the hardware reset signal (MCSPI_RST). McSPI can be reset by software through the MCSPI_SYSCONFIG[1] SOFTRESET bit. This bit has the same impact on the module as the hardware reset signal. The only exception is that the MCSPI_SYSCONFIG register is not affected by that software reset.

24.4.4.3 Master Mode

24.4.4.3.1 Master Mode Features

The McSPI master mode supports multichannel communication with up to four independent SPI communication channel contexts. The McSPI initiates a data transfer on the data lines (SPIDAT[0] and SPIDAT[1]) and generates clock (SPICLK) and control (SPIEN) signals.

Connected to multiple external devices, the McSPI exchanges data with one SPI device at a time through two main modes (available in slave mode):

- Two-data-pins interface mode (transmit-and-receive mode for full-duplex transmission)
- Single-data-pin interface mode (recommended for half-duplex transmission)

Note

There is a fixed chip select line allocation in multichannel master mode. Channel x SPIEN[x] is mapped to spim_cs[i] pin.

Two DMA request events (read and write) allow synchronized accesses of the DMA controller with the activity of McSPI.

Three interrupt events can be used for data transmission and reception in master mode (for more information about interrupts, see Section 24.4.4.7.1, *Interrupt Events in Master Mode*).

24.4.4.3.2 Master Transmit-and-Receive Mode (Full Duplex)

In full-duplex transmission, data is transmitted (shifted out serially on SPIDAT[0]) and received (shifted in serially on SPIDAT[1]) simultaneously on separate data lines.

The master transmit-and-receive mode is programmable per channel (the MCSPI_CHxCONF[13:12] TRM bit field).

Channel access to the shift registers for transmission/reception is based on the MCSPI_TXx transmitter register state, the MCSPI_RXx receiver register state, and round-robin arbitration.

Channels that meet the following rules are included in the round-robin list of active channels scheduled for transmission and/or reception. The arbiter skips channels that do not meet the rules and searches in the rotation for the next enabled channel.

- **Rule 1:** Only enabled channels (the MCSPI_CHxCTRL[0] EN bit) can be scheduled for transmission and/or reception.
- **Rule 2:** If its MCSPI_TXx transmitter register is not empty (the MCSPI_CHxSTAT[1] TXS bit), an enabled channel can be scheduled when the shift register is assigned. If the MCSPI_TXx register is empty when the shift register is assigned, the TXx_UNDERFLOW event is activated, and the next enabled channel with new data to transmit is scheduled (see also transmit-only mode).
- **Rule 3:** An enabled channel can be scheduled if its receive register is not full (the MCSPI_CHxSTAT[0] RXS bit) when the shift register is assigned (see also receive-only mode). Therefore, the MCSPI_RXx register cannot be overwritten. The SPI1.MCSPI_IRQSTATUS[3] RX0_OVERFLOW bit is never set to this mode.

When SPI word transfer completes (the MCSPI_CHxSTAT[2] EOT bit is set), the updated MCSPI_TXx register of the next scheduled channel is loaded into the shift register. The serialization (transmit-and-receive) starts depending on the channel communication configuration. When serialization completes, the received data transfers to the channel receive register.

The serial clock (SPICLK) synchronizes shifting and sampling of the information on the two serial data lines (SPIDAT[0] and SPIDAT[1]). Each time a bit transfers out from the master, 1 bit transfers in from the slave.

Figure 24-84 shows an example of a full-duplex system with a master device (McSPI module *m*) on the left and a slave device on the right. After eight cycles of the serial clock SPICLK, WordA transfers from the master to the slave. At the same time, WordB transfers from the slave to the master.

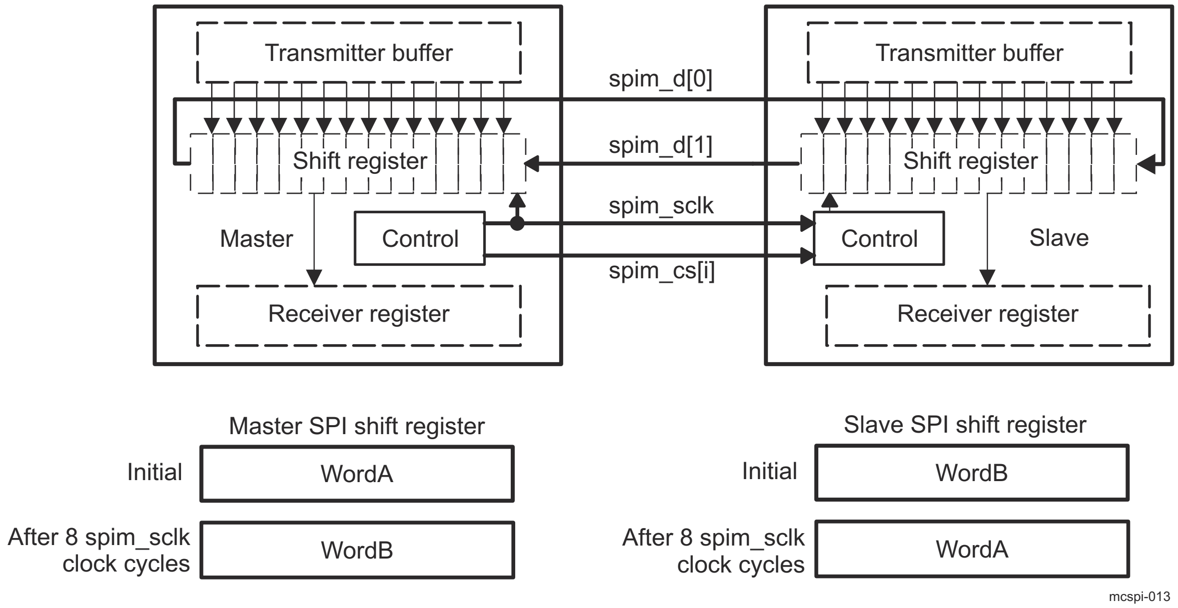


Figure 24-84. SPI Full-Duplex Transmission (Example)

24.4.4.3.3 Master Transmit-Only Mode (Half Duplex)

The master transmit-only mode prevents the microprocessor unit (MPU) from reading the MCSPI_RXx register (minimizing data movement) when only transmission is meaningful.

The master transmit-only mode is programmable per channel (the MCSPI_CHxCONF[13:12] TRM bit field). Transmission starts only after data is loaded into the MCSPI_TXx register.

Rule 1 and Rule 2, defined in Section 24.4.4.3.2, apply in this mode.

Rule 3, defined in Section 24.4.4.3.2, does not apply.

In master transmit-only mode, the MCSPI_RXx register state FULL does not prevent transmission and the MCSPI_RXx register is always overwritten with the new SPI word. This event is not significant when only transmission is meaningful. Thus, the RX0_OVERFLOW bit in the MCSPI_IRQSTATUS register is never set in this mode.

The hardware automatically disables the RX_FULL interrupt and the DMA read requests.

The transfer status is given by the MCSPI_CHxSTAT[2] EOT bit.

24.4.4.3.4 Master Receive-Only Mode (Half Duplex)

The master receive mode prevents the MPU from refilling the MCSPI_TXx register (minimizing data movement) when only reception is meaningful.

The master receive mode is programmable per channel (the MCSPI_CHxCONF[13:12] TRM bit field).

The master receive-only mode enables channel scheduling only on the empty state of the MCSPI_RXx register.

Rule 1 and Rule 3, defined in Section 24.4.4.3.2, apply in this mode.

Rule 2, defined in Section 24.4.4.3.2, does not apply.

In the master receive-only mode, software must write dummy data to the MCSPI_TXx register. Only one dummy write is enough to receive any number of words from the slave. Software must ensure that the MCSPI_TXx register is always full (the TXx_EMPTY bits of MCSPI_IRQSTATUS) when receiving. The content of the MCSPI_TXx register is always loaded into the shift register when the shift register is assigned. After writing the dummy data to the MCSPI_TXx register, the TXx_EMPTY and TXx_UNDERFLOW bits in the MCSPI_IRQSTATUS register are never set in receive-only mode.

The MCSPI_CHxSTAT[2] EOT bit gives the status of serialization. The RXx_FULL bits of the MCSPI_IRQSTATUS register are set when received data is loaded from the shift register to the corresponding MCSPI_RXx register. The MCSPI_IRQSTATUS[3] RX0_OVERFLOW bit is never set in this mode.

24.4.4.3.5 Single-Channel Master Mode

When the McSPI is configured as a master device with a single enabled channel (MCSPI_MODULCTRL[2] MS = 0 and MCSPI_MODULCTRL[0] SINGLE = 1), the assertion of the SPIEN[x] signal is optional depending on device connected to the controller. In 3-pin mode (MCSPI_MODULCTRL[1] PIN34 = 1) the controller starts transmitting data when a write to the MCSPI_TXx register or the FIFO is performed. In 4-pin mode (MCSPI_MODULCTRL[1] PIN34 = 0) the assertion and de-assertion of SPIEN[x] is controlled by software using the MCSPI_CHxCONF[20] FORCE bit.

24.4.4.3.5.1 Programming Tips When Switching to Another Channel

When a single channel is enabled and data transfer is ongoing:

- Wait for the SPI word transfer to complete (wait until the MCSPI_CHxSTAT[2] EOT bit is set to 1) before disabling the current channel and enabling a different channel.
- Disable the current channel, and then enable the other channel.

24.4.4.3.5.2 Force SPIEN[x] Mode

Continuous transfers are allowed manually by keeping the SPIEN[x] signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the SPIEN[x] line. This mode is supported by all channels and any master sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the SPIEN[x] active mode is supported when:

- A single channel is used (with the MCSPI_MODULCTRL[0] SINGLE bit set to 1).
- Transfer parameters are loaded in the configuration register of the appropriate channel (MCSPI_CHxCONF).

  The state of the SPIEN[x] signal is programmable:
  - Writing 1 to the MCSPI_CHxCONF[20] FORCE bit drives the SPIEN[x] line high when the MCSPI_CHxCONF[6] EPOL bit is set to 0. SPIEN[x] is driven low when the MCSPI_CHxCONF[6] EPOL bit is set to 1.
  - Writing 0 to the MCSPI_CHxCONF[20] FORCE bit drives the SPIEN[x] line low when the MCSPI_CHxCONF[6] EPOL bit is set to 0. SPIEN[x] is driven high when the MCSPI_CHxCONF[6] EPOL bit is set to 1.
- A single channel is enabled (the MCSPI_CHxCTRL[0] EN bit is set to 1). The first enabled channel activates the SPIEN[x] line.

When the channel is enabled, the SPIEN[x] signal activates with the programmed polarity. As in the multichannel master mode, the transfer start depends on the status of the MCSPI_TXx register (the MCSPI_CHxSTAT[1] TXS bit), the status of the MCSPI_RXx register (the MCSPI_CHxSTAT[1] RXS bit), and the defined mode (the MCSPI_CHxCONF[13:12] TRM bit field) of the channel enabled.

The MCSPI_CHxSTAT[2] EOT bit gives the transfer status of each SPI word. The RXx_FULL bit in the MCSPI_IRQSTATUS register is set when received data is loaded from the shift register to the MCSPI_RXx register.

A change in the configuration parameters is propagated directly on the SPI interface. If the SPIEN[x] signal is activated, ensure that the configuration is changed only between SPI words to avoid corrupting the current transfer.

Note

To avoid data corruption, SPIEN polarity and SPICLK phase and SPICLK polarity must not be modified when the SPIEN[x] signal is activated.

A delay between SPI words that requires the connected SPI slave device to switch from one configuration to another (for instance, from transmit-only to receive-only) must be handled by software.

At the end of the last SPI word, the channel must be deactivated (the MCSPI_CHxCTRL[0] EN bit set to 0) and SPIEN[x] can be forced to its INACTIVE state using the MCSPI_CHxCONF[20] FORCE bit.

Figure 24-85 and Figure 24-86 show successive transfers with SPIEN[x] maintained active low with a different configuration for each SPI word in single-data-pin and dual-data-pin interface modes, respectively.

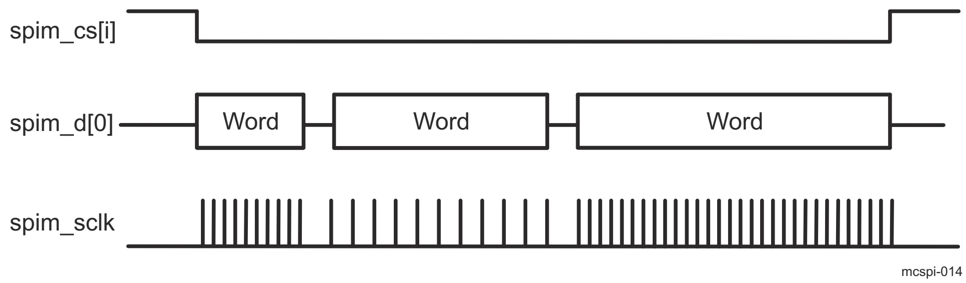


Figure 24-85. Continuous Transfers With SPIEN[x] Maintained Active (Single-Data-Pin Interface Mode)

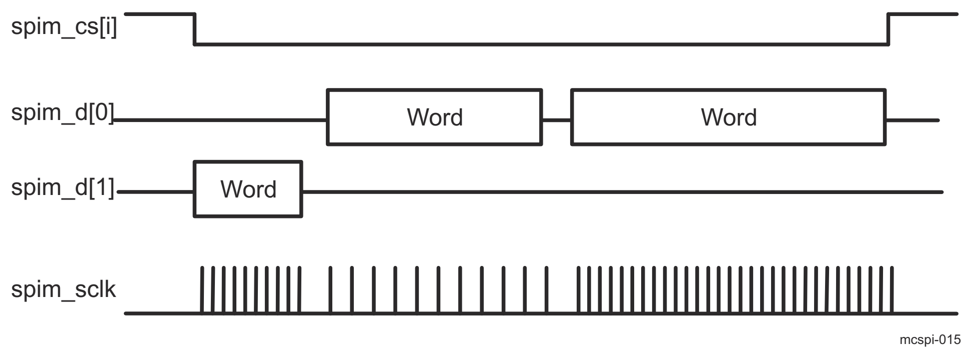


Figure 24-86. Continuous Transfers With SPIEN[x] Maintained Active (Dual-Data-Pin Interface Mode)

Note

The SPIEN[x] signal can be maintained active via software using the MCSPI_CHxCONF[20] FORCE bit only when the MCSPI_MODULCTRL[0] SINGLE bit is set to 0x1.

24.4.4.3.5.3 Turbo Mode

Turbo mode improves the throughput of the SPI interface when a single channel is enabled by allowing transfers until the shift register and the MCSPI_RXx register are full. Turbo mode is time saving when a transfer exceeds two words. This mode is programmable per channel (through the SPI1.MCSPI_CHxCONF[9] TURBO bit).

When several channels are enabled, the TURBO bit has no effect and the channel access to the shift registers remains as previously described.

In turbo mode, Rule 1 and Rule 2 apply, but Rule 3 does not (see Section 24.4.4.3.2, *Master Transmit-and-Receive Mode (Full Duplex)*). An enabled channel can be scheduled if its receive register is full (the MCSPI_CHxSTAT[0] RXS bit) when the shift-register is assigned until the shift register is full.

The MCSPI_RXx register cannot be overwritten in turbo mode. Consequently, the MCSPI_IRQSTATUS[3] RX0_OVERFLOW bit is never set in this mode.

24.4.4.3.6 Start-Bit Mode

In start-bit mode, an extended bit is added before the SPI word to indicate whether the next SPI word must be handled as a command or as data. This feature is available only in master mode. Start-bit mode cannot be used at the same time as turbo mode and/or force SPIEN[x] mode. In this case, only one channel can be used; round-robin arbitration is not possible.

This mode is programmable per channel by setting the MCSPI_CHxCONF[23] SBE bit to 1. The polarity of the extended bit is programmable per channel. When the MCSPI_CHxCONF[24] SBPOL bit is set to 0, the SPI word must be handled as a command. When the MCSPI_CHxCONF[24] SBPOL bit is set to 1, the SPI word must be handled as data. Moreover, start-bit polarity can be changed dynamically during start-bit transfer without disabling the channel for reconfiguration; in this case, users must configure the MCSPI_CHxCONF[24] SBPOL bit before writing the SPI word to be transmitted to the TX register.

24.4.4.3.7 Chip-Select Timing Control

The chip-select (CS) timing control is available only in master mode with automatic CS generation (the MCSPI_MODULCTRL[0] SINGLE bit set to 0) to add a programmable delay between CS assertion and first clock edge, or CS removal and last clock edge. This option is available only in 4-pin mode when MCSPI_MODULCTRL[1] PIN34 set to 0.

This mode is programmable per channel through the MCSPI_CHxCONF[26:25] TCS0 bit field.

Figure 24-87 shows the CS SPIEN timing controls.

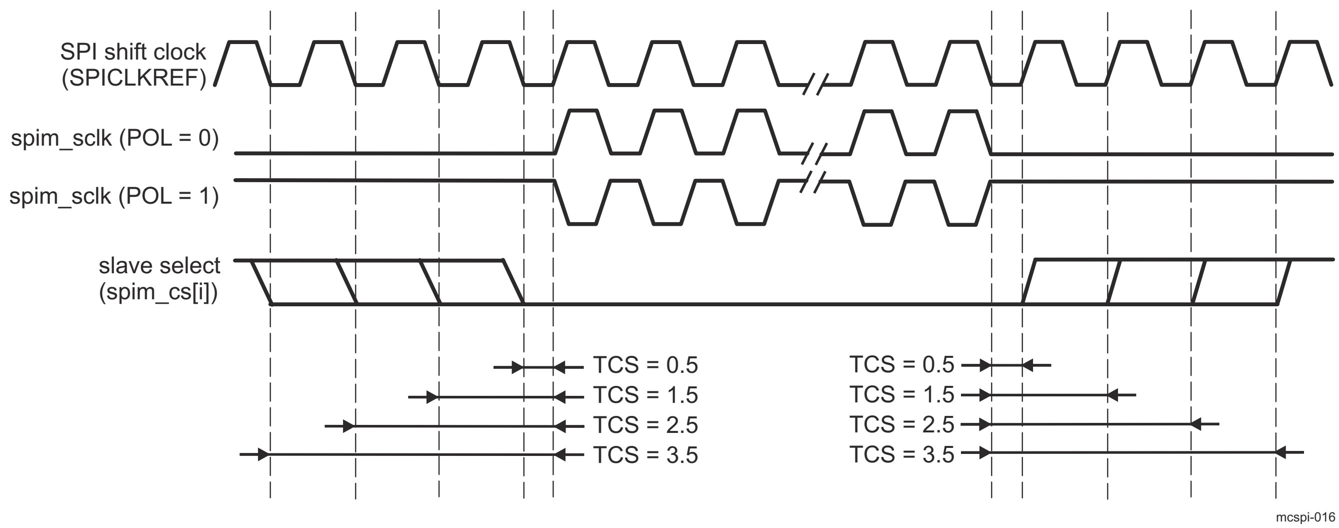


Figure 24-87. CS (SPIEN) Timing Controls

Note

Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between CS assertion and the first clock edge with PHA = 1 or between CS removal and the last clock edge with PHA = 0.

24.4.4.3.8 Programmable SPI Clock

In master mode, the baud rate of the SPI serial clock is programmable.

An internal reference clock, SPIm_FCLK, is used as input of a programmable divider (the MCSPI_CHxCONF[5:2] CLKD bit field) to generate the bit rate of the serial output clock SPICLK. Table 24-203 summarizes the supported divisor values.

Table 24-203. SPI Master Clock Rates

Divider	Clock Rate
1	48 MHz[(1)]
2	24 MHz[(1)]

Table 24-203. SPI Master Clock Rates (continued)

Divider	Clock Rate
4	12 MHz
8	6 MHz
16	3 MHz
32	1.5 MHz
64	750 kHz
128	375 kHz
256	~187 kHz
512	~93.7 kHz
1024	~46.8 kHz
2048	~23.4 kHz
4096	~11.7 kHz
8192 and higher: Division not supported	–

(1) These frequencies are not necessarily supported by all SPI modules. For more information, see the *Timing Requirements and Switching Characteristics* chapter in the device data manual.

24.4.4.3.8.1 Clock Ratio Granularity

By default, the clock division ratio is defined by the MCSPI_CHxCONF[5:2] CLKD bit field with power-of-2 granularity leading to a clock division in the range 1 to 4096; in this case, the duty cycle is always 50 percent. With the MCSPI_CHxCONF[29] CLKG bit, clock division granularity can be changed to one clock cycle; in that case the MCSPI_CHxCTRL[15:8] EXTCLK bit field is concatenated with the MCSPI_CHxCONF[5:2] CLKD bit field to give a 12-bit-wide division ratio in the range 1 to 4096.

When granularity is one clock cycle (the CLKG bit set to 1), for the odd value of the clock ratio, the clock high level lasts one clock cycle more than the low level, depending on the MCSPI_CHxCONF[1] POL and MCSPI_CHxCONF[0] PHA bits (see Table 24-204).

Table 24-204. CLKSPIO High/Low Time Computation

Clock Ratio F_{RATIO}	CLKSPIO High Time	CLKSPIO Low Time
1	T_{HIGH_REF}	T_{LOW_REF}
Even ≥ 2	T_ref × (F_{RATIO}/2)	T_ref × (F_{RATIO}/2)
Odd ≥ (POL = PHA)	T_ref × (F_{RATIO}– 1)/2	T_ref × (F_{RATIO} + 1)/2
Odd ≥ (POL ≠ PHA)	T_ref × (F_{RATIO}+ 1)/2	T_ref × (F_{RATIO} – 1)/2

Note

F_{RATIO} = SPICLK frequency (F_{OUT}) division ratio
T_{HIGH} = SPICLK high time period
T_{LOW} = SPICLK low time period
T_ref = FCLK period
T_{HIGH_REF} = FCLK high time period
T_{LOW_REF} = FCLK low time period

If the CLKG bit is set to 1; F_{RATIO} = EXTCLK concatenated with CLKD + 1.

For odd ratio values, the duty cycle is calculated as follows:

Duty_cycle = (1 – 1/F_{RATIO})/2

Table 24-205 shows examples of clock granularity with a clock source frequency of 48 MHz.

Table 24-205. Clock Granularity Examples

EXTCLK	CLKD	CLKG	F_{RATIO}	PHA	POL	T_{HIGH} (ns)	T_{LOW} (ns)	T_{PERIOD} (ns)	Duty Cycle	F_{OUT} (MHz)
X	0	0	1	X	X	10.4	10.4	20.8	50–50	48
X	1	0	2	X	X	20.8	20.8	41.6	50–50	24
X	2	0	4	X	X	41.6	41.6	83.2	50–50	12
X	3	0	8	X	X	83.2	83.2	166.4	50–50	6
0	0	1	1	X	X	10.4	10.4	20.8	50–50	48
0	1	1	2	X	X	20.8	20.8	41.6	50–50	24
0	2	1	3	1	0	41.6	20.8	62.4	66–33	16
0	2	1	3	1	1	20.8	41.6	62.4	33–66	16
0	3	1	4	X	X	41.6	41.6	83.2	50–50	12
5	0	1	81	1	0	852.8	832	1684.8	50.6–49.4	0.592
5	7	1	88	X	X	915.2	915.2	1830.4	50–50	0.545

24.4.4.4 Slave Mode

To select the McSPI slave mode, set the MCSPI_MODULCTRL[2] MS bit.

A McSPI slave device can be connected to up to four external SPI master devices but handles transactions with one SPI master device at a time.

In slave mode, the McSPI initiates data transfer on the data lines (SPIDAT[0] and SPIDAT[1]) when it is selected by an active control signal (SPIEN[x]) and receives an SPI clock (SPICLK) from the external SPI master device. Only channel 0 can be configured as a slave but through the MCSPI_CH0CONF[22:21] SPIENSLV bit field any of the SPIEN[x] signals can be used to select the McSPI module. In slave mode and when the MCSPI_MODULCTRL[1] PIN34 is set to 0x0 (default behavior), the McSPI uses the edge of SPIEN[x] to detect word length. For this reason, SPIEN[x] must become inactive between each word.

When the MCSPI_MODULCTRL[1] PIN34 is set to 0x0, the McSPI does not support SPIEN[x] active between SPI words. In this case, the McSPI uses the edge to detect word length.

When the MCSPI_MODULCTRL[1] PIN34 is set to 0x1, a multiword transfer can be performed without needing the external SPI master to deactivate SPIEN[x] between each word as in this case the McSPI module works in 3-pin slave mode and SPIEN[x] is not needed.

24.4.4.4.1 Dedicated Resources

Only channel 0 can be enabled in slave mode. In this section, register names such as SPI1.MCSPI_CHxCTRL stand for SPI1.MCSPI_CH0CTRL, where x = 0 (channel 0 control register).

Figure 24-88 shows an example of four slaves wired on a single master device.

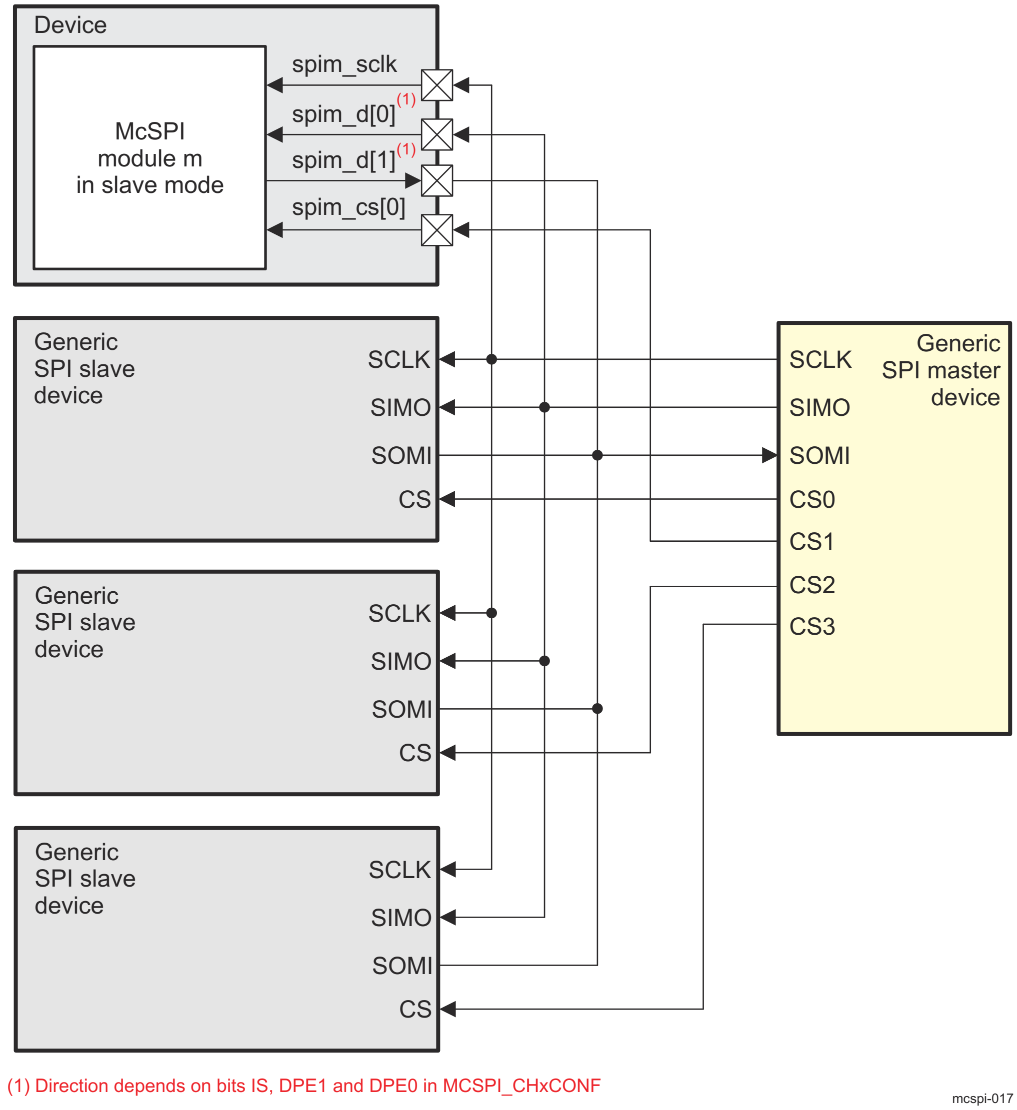


Figure 24-88. Example of McSPI Slave With One Master and Multiple Slave Devices on Channel 0

Channel 0 in slave mode has the following resources:

- Its own channel enable, programmable with the MCSPI_CHxCTRL[0] EN bit (where x = 0). This channel must be enabled before transmission and reception.
- For this mode, the slave-select signal can be detected on any of the SPIEN[x] ports. This is programmable with the MCSPI_CHxCONF[22:21] SPIENSLV bit field (where x = 0).
- Its own transmitter register, MCSPI_TXx (where x = 0), on top of the common transmit shift register. If the MCSPI_TXx register is empty, the MCSPI_CHxSTAT[1] TXS bit (where x = 0) is set. If McSPI is selected by an external master (the active signal on the SPIEN[x] port assigned to channel 0), the MCSPI_TXx register content of channel 0 is always loaded into the shift register, whether its content is updated or not. The MCSPI_TXx register must be loaded before McSPI is selected by a master.
- Its own receiver register, MCSPI_RXx (where x = 0), on top of the common receive shift register. If the MCSPI_RXx register is full, the MCSPI_CHxSTAT[0] RXS bit (where x = 0) is set.

Note

The MCSPI_TXx and MCSPI_RXx registers of the other channels are not used. Reading from or writing to a channel register other than channel 0 has no effect.

- Its own communication configuration with the following parameters through the MCSPI_CHxCONF register (where x = 0):
  - Transmit and receive modes, programmable with the TRM field
  - Interface mode (two data pins or single data pin) and data pins assignment, both programmable with the IS and DPE bits. (The SPIm modules are in slave mode after reset and must be properly configured for the modules to act in master mode.)
  - SPI word length, programmable with the WL bits
  - SPIEN[x] polarity, programmable with the EPOL bit
  - SPICLK polarity, programmable with the POL bit
  - SPICLK phase, programmable with the PHA bit

  The SPICLK frequency of a transfer is controlled by the external SPI master connected to the McSPI slave device. The MCSPI_CHxCONF[5:2] CLKD bit field (where x = 0) is not used in slave mode.

  Note

  The configuration of the channel can be loaded in the MCSPI_CHxCONF register (where x = 0) only when the channel is disabled.
- Two DMA request events, read and write, synchronize read/write accesses of the DMA controller with the activity of McSPI. DMA requests are asserted using the MCSPI_CHxCONF[15] DMAR bit (where x = 0) for reading and the MCSPI_CHxCONF[14] DMAW bit (where x = 0) for writing.
- Four interrupt events (see Section 24.4.4.7.2, *Interrupt Events in Slave Mode*)

24.4.4.4.2 Slave Transmit-and-Receive Mode

The slave receive mode is programmable (set the MCSPI_CHxCONF[13:12] TRM bit field [where x = 0] to 0x0).

In slave transmit-and-receive mode, the MCSPI_TXx register must be loaded before McSPI is selected by an external SPI master device.

After a channel is enabled, transmission and reception proceed with interrupt and DMA request events.

The MCSPI_TXx register content is always loaded in the shift register whether it is updated or not. The event TXx_UNDERFLOW is activated accordingly and does not prevent transmission.

When the SPI word transfer completes (the MCSPI_CHxSTAT0[2] EOT bit [where x = 0] is set to 1), the received data is transferred to the channel receive register.

To use McSPI as a slave transmit-only device, the RXx_FULL and RX0_OVERFLOW interrupts and DMA read requests must be disabled due to the state of the MCSPI_RXx register (see Section 24.4.4.7.2, *Interrupt Events in Slave Mode*).

24.4.4.4.3 Slave Transmit-Only Mode

The slave transmit-only mode is programmable (set the MCSPI_CHxCONF[13:12] TRM bit field [where x = 0] to 0x2) and avoids the requirement for the MPU to read the MCSPI_RXx register (minimizing data movement) only when transmission is meaningful.

To use the McSPI as a slave transmit-only device, the RXx_FULL and RX0_OVERFLOW interrupts and DMA read requests must be disabled due to the state of the MCSPI_RXx register.

When the SPI word transfer completes, the MCSPI_CHxSTAT[2] EOT bit is set (where x = 0).

Figure 24-89 shows a half-duplex system with a master device on the left and a transmit-only slave device on the right. Each time a bit transfers out from the slave, 1 bit transfers in the master. After eight cycles of the serial clock SPICLK, WordB transfers from the slave to the master.

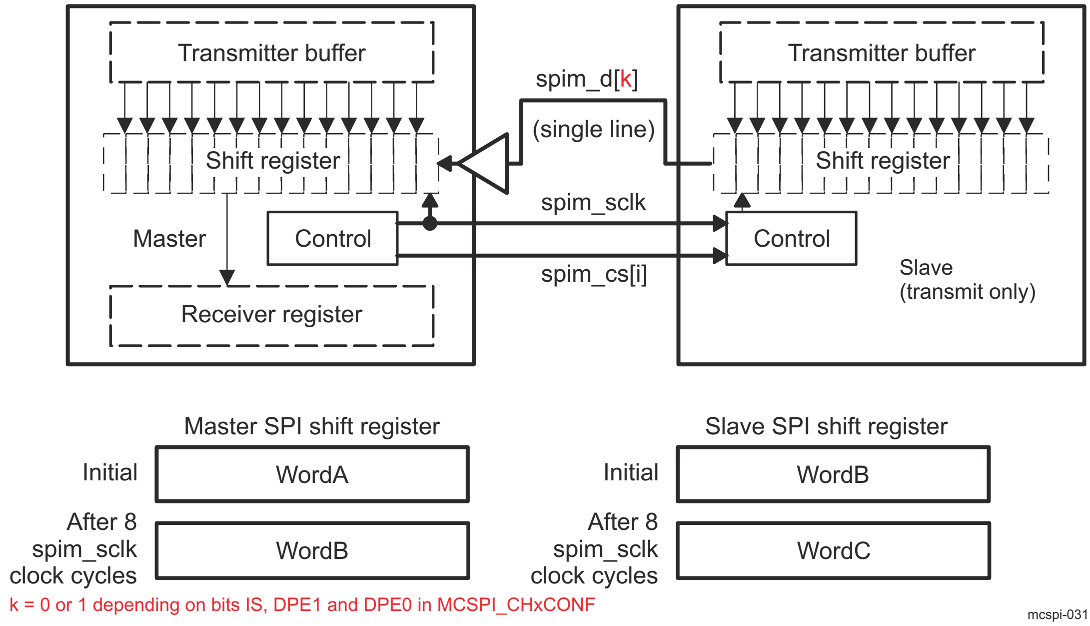


Figure 24-89. SPI Half-Duplex Transmission (Transmit-Only Slave)

24.4.4.4.4 Slave Receive-Only Mode

The slave receive mode is programmable (set the MCSPI_CHxCONF[13:12] TRM bit field [where x = 0] to 0x1).

In receive-only mode, the MCSPI_TXx register must be loaded before the McSPI is selected by an external SPI master device. The MCSPI_TXx register content is always loaded into the shift register whether it is updated or not. The TXx_UNDERFLOW event is activated accordingly and does not prevent transmission.

When the SPI word transfer completes (the MCSPI_CHxSTAT0[2] EOT bit [where x = 0] is set to 1), the received data is transferred to the channel receive register.

To use the McSPI as a slave receive-only device, the TXx_EMPTY and TXx_UNDERFLOW interrupts and the DMA write requests must be disabled due to the state of the MCSPI_TXx register.

For a full-duplex transmission, the serial clock (SPICLK) synchronizes shifting and sampling of the information on the single serial data line. For full duplex, two data lines are required. If SPICLK synchronizes on a single serial data line, the data line should be half-duplex.

Figure 24-90 shows a half-duplex system with a master device on the left and a receive-only slave device on the right. Each time a bit transfers out from the master, 1 bit transfers in from the slave. After eight cycles of the serial clock SPICLK, WordA transfers from the master to the slave.

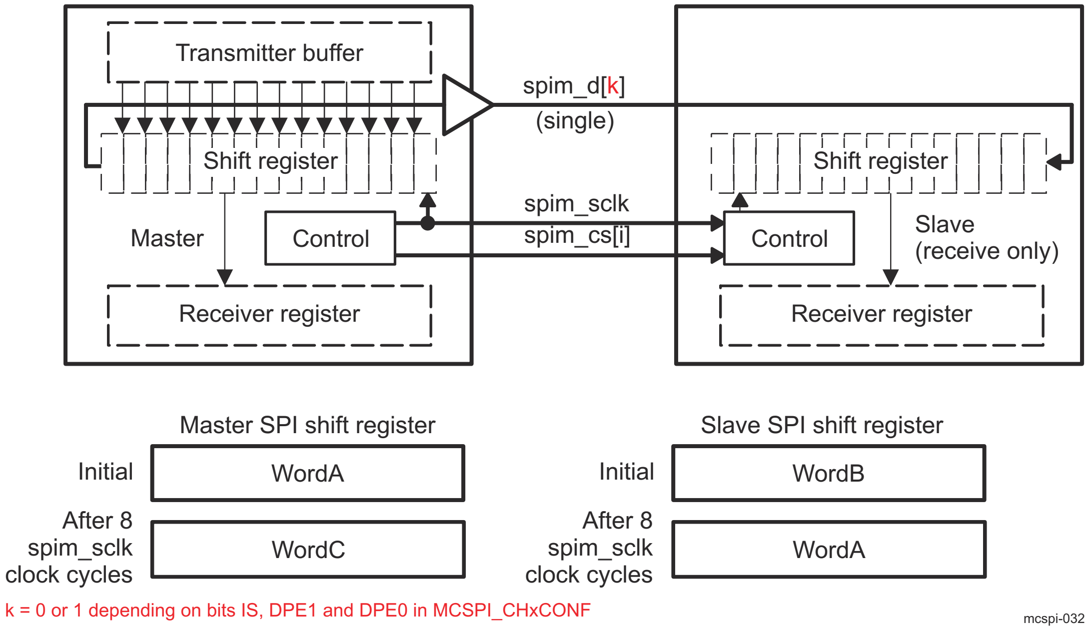


Figure 24-90. SPI Half-Duplex Transmission (Receive-Only Slave)

24.4.4.5 3-Pin or 4-Pin Mode

Depending on targeted application the SPI interface can be configured to use 3 or 4 pins through the MCSPI_MODULCTRL[1] PIN34 bit. If this bit is set to 0, McSPI is in 4-pin mode using the SPICLK, SPIDAT[0], SPIDAT[1] and SPIEN[x] signals. If PIN34 is set to 1 the controller is in 3-pin mode and SPIEN[x] is not used. In this mode all options related to chip select management are useless (EPOL, FORCE and TCS0 bits of MCSPI_CHxCONF). 3-pin and 4-pin operation applies to both master and slave modes.

24.4.4.6 FIFO Buffer Management

The McSPI controller has a built-in 128-byte buffer to unload the DMA or interrupt handler and improve data throughput.

This buffer can be used by only one channel at a time and is selected by setting the MCSPI_CHxCONF[28] FFER or MCSPI_CHxCONF[27] FFEW bit to 1. If several channels are selected and several FIFO enable bit fields are set to 1, the controller forces the buffer not to be used; the driver must set only one FIFO enable bit field.

The buffer can be used in the following modes:

- Master or slave mode
- Transmit-only, receive-only, or transmit-and-receive mode
- Single channel or turbo mode, or normal round-robin mode. In round-robin mode the buffer is used by only one channel.

Every word length (MCSPI_CHxCONF[11:7] WL) is supported.

In transmit-and-receive mode, the buffer can be used in transmit (see Figure 24-91) or receive (see Figure 24-92) directions, or in both directions. If only one direction is chosen in transmit-and-receive mode, the full buffer is used for this direction. In both directions, the buffer is split into two halves, one for each direction (see Figure 24-93).

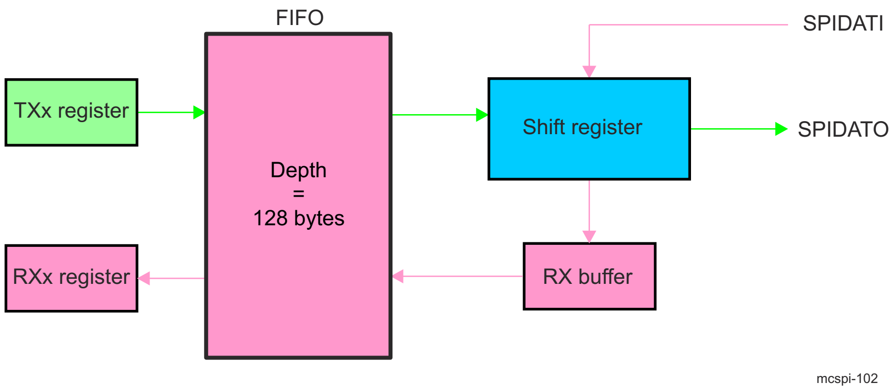


Figure 24-91. Buffer Used in Transmit Direction Only

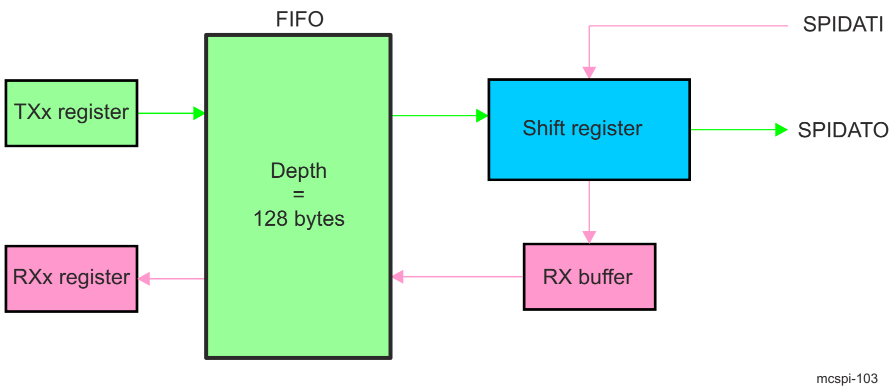


Figure 24-92. Buffer Used in Receive Direction Only

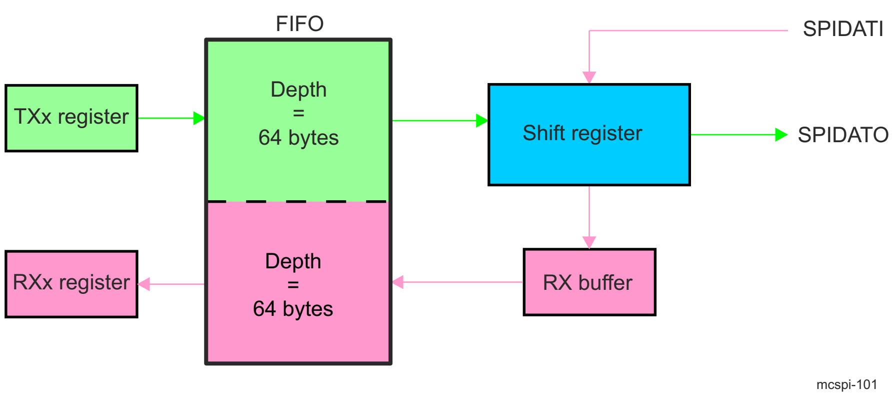


Figure 24-93. Buffer Used for Transmit and Receive Directions

Two levels (MCSPI_XFERLEVEL[5:0] AEL and MCSPI_XFERLEVEL[13:8] AFL) rule the buffer management. The granularity of these levels is 1 byte; it is not aligned with the SPI word length. The driver must set these values as a multiple of the SPI word length defined in MCSPI_CHxCONF[11:7] WL. Table 24-206 lists the number of bytes written in the FIFO, depending on the word length.

Table 24-206. FIFO Writes, Word Length Relationship

	SPI Word Length (WL)		
	3 ≤ WL≤ 7	8 ≤ WL ≤ 15	16 ≤ WL ≤ 31
Number of bytes written in the FIFO	1 byte	2 bytes	4 bytes

The FIFO buffer pointers are reset when the corresponding channel is enabled or the FIFO configuration changes.

24.4.4.6.1 Buffer Almost Full

The MCSPI_XFERLEVEL[15:8] AFL bit field is needed when the buffer is used to receive an SPI word from a slave (the MCSPI_CHxCONF[28] FFER bit must be set to 1). It defines the almost-full buffer status. See Figure 24-94.

When the FIFO pointer reaches this level, an interrupt or a DMA request is sent to the MPU to enable the system to read AFL + 1 bytes from the receive register.

Note

AFL + 1 must correspond to a multiple value of the MCSPI_CHxCONF[11:7] WL bit field.

When DMA is used, the request is deasserted after the first receive register read.

No new request is asserted again as long as the system has not performed the correct number of read accesses.

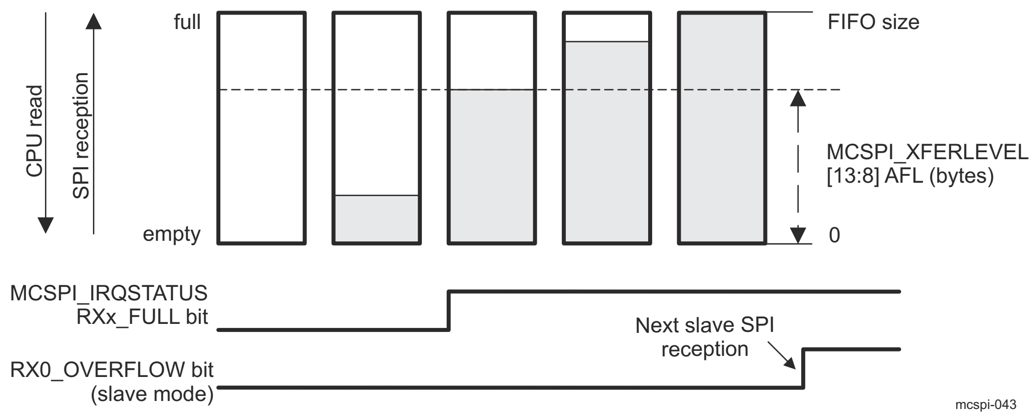


Figure 24-94. Buffer Almost Full Level (AFL)

Note

The MCSPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_RXx request is asserted on the same conditions as the MCSPI_IRQSTATUS RXx_FULL flag.

24.4.4.6.2 Buffer Almost Empty

The MCSPI_XFERLEVEL[7:0] AEL bit field is needed when the buffer is used to transmit an SPI word to a slave (the MCSPI_CHxCONF[27] FFEW bit must be set to 1). It defines the almost-empty buffer status. See Figure 24-95.

When the FIFO pointer does not reach this level, an interrupt or a DMA request is sent to the MPU to enable the system to write AEL + 1 bytes to the transmit register.

Note

AEL + 1 must correspond to a multiple value of the MCSPI_CHxCONF[11:7] WL bit field.

When DMA is used, the request is deasserted after the first transmit register write.

No new request is asserted again as long as the system has not performed the correct number of write accesses.

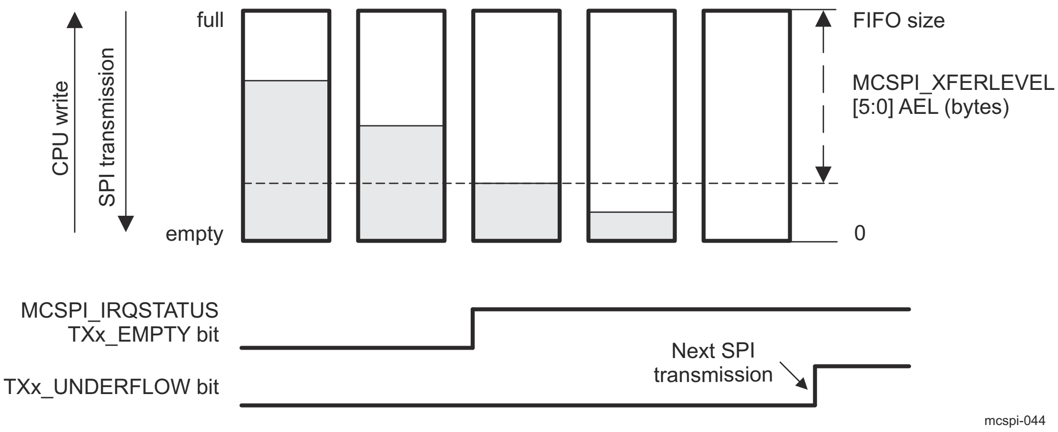


Figure 24-95. Buffer Almost Empty Level (AEL)

Note

The MCSPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_TXx request is asserted on the same conditions as the MCSPI_IRQSTATUS TXx_EMPTY flag.

24.4.4.6.3 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user must previously configure in the MCSPI_XFERLEVEL register the AEL and AFL levels and especially the MCSPI_XFERLEVEL[31:16] WCNT bit field to define the number of SPI words to be transferred using the FIFO before enabling the channel.

This counter lets the controller stop the transfer correctly after a defined number of SPI word transfers. If WNCT is set to 0x0000, the counter is not used and the user must stop the transfer manually by disabling the channel; in this case, the user does not know how many SPI transfers have been done. For received words, software must poll the CHxSTAT[5] RXFFE bit and read the MCSPI_RXx receive register to empty the FIFO buffer.

When the end-of-word count interrupt is generated (the MCSPI_IRQSTATUS[17] EOW bit is set), the user can disable the channel and poll the MCSPI_CHxSTAT[5] RXFFE bit to know the last SPI words in the FIFO buffer and read them.

No new request is asserted as long as the system has not performed the correct number of write accesses.

24.4.4.7 Interrupts

Each channel can issue interrupt events.

Each interrupt event has status bits in the MCSPI_IRQSTATUS register (RXx_FULL, TXx_UNDERFLOW, TXx_EMPTY, etc.) (where x = 0, 3) that indicate whether service is required. Each status bit has an interrupt enable bit (a mask) in the MCSPI_IRQENABLE register (RXx_FULL_ENABLE, TXx_UNDERFLOW_ENABLE, TXx_EMPTY_ENABLE, etc.).

When an interrupt occurs and a mask is later applied on it, the interrupt line is not asserted again, even if the interrupt source is not serviced.

The McSPI supports interrupt-driven and polling operations.

24.4.4.7.1 Interrupt Events in Master Mode

In master mode, the interrupt events related to the state of the MCSPI_TXx register are TXx_EMPTY and TXx_UNDERFLOW. The interrupt event related to the state of the MCSPI_RXx register is RXx_FULL.

24.4.4.7.1.1 TXx_EMPTY

The TXx_EMPTY event is activated when a channel is enabled and its MCSPI_TXx register is empty (transient event). Enabling a channel automatically triggers this event, except in master receive-only mode (see Section 24.4.4.3.4, *Master Receive-Only Mode*). When the FIFO buffer is enabled (the MCSPI_CHxCONF[27] FFEW bit is set to 1), the MCSPI_IRQSTATUS TXx_EMPTY bit is set as soon as there is enough space in the buffer to write a number of bytes defined by the MCSPI_XFERLEVEL[5:0] AEL bit field.

The MCSPI_TXx register must be loaded with data to remove the source of the interrupt; the MCSPI_IRQSTATUS TXx_EMPTY interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new TXx_EMPTY event is asserted as long as the MPU has not performed the number of writes into the MCSPI_TXx register defined by the MCSPI_XFERLEVEL[5:0] AEL bit field. The MPU must perform the correct number of writes.

24.4.4.7.1.2 TXx_UNDERFLOW

The event TXx_UNDERFLOW is activated when the channel is enabled and if the MCSPI_TXx register or the FIFO is empty (not updated with new data) when an external master device starts a data transfer with the McSPI (transmit and receive).

The TXx_UNDERFLOW is a harmless warning in master mode.

To avoid having a TXx_UNDERFLOW event at the beginning of a transmission, the TXx_UNDERFLOW event is not activated when no data has been loaded into the MCSPI_TXx register, because the channel is enabled. To avoid having a TXx_UNDERFLOW event, the MCSPI_TXx register must seldom be loaded.

The MCSPI_IRQSTATUS TXx_UNDERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

24.4.4.7.1.3 RXx_ FULL

The RXx_FULL event is activated when a channel is enabled and the MCSPI_RXx register becomes filled (transient event). When the FIFO buffer is enabled (the MCSPI_CHxCONF[28] FFER bit is set to 1), RXx_ FULL is asserted as soon as the number of bytes held in the FIFO to be read reaches the MCSPI_XFERLEVEL[13:8] AFL threshold.

The MCSPI_RXx register must be read to remove the source of the interrupt; the MCSPI_IRQSTATUS RXx_FULL interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new RXx_FULL event is asserted as long as the MPU has not performed AFL + 1 reads into MCSPI_RXx. The MPU must perform the correct number of reads.

24.4.4.7.1.4 End Of Word Count

The MCSPI_IRQSTATUS[17] EOW event (end of word count) is activated when the channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller performs the number of transfers defined in the MCSPI_XFERLEVEL[31:16] WCNT bit field. If WCNT is set to 0x0000, the counter is not enabled and this interrupt is not generated.

The end of word count interrupt also indicates that the SPI transfer is halted on the channel using the FIFO buffer as soon as MCSPI_XFERLEVEL[31:16] WCNT is not reloaded and the channel is not re-enabled.

The MCSPI_IRQSTATUS[17] EOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

24.4.4.7.2 Interrupt Events in Slave Mode

In slave mode, the interrupt events related to the state of the MCSPI_TXx register are TXx_EMPTY and TXx_UNDERFLOW. The interrupt events related to the state of the MCSPI_RXx are RXx_FULL and RX0_OVERFLOW (channels 1, 2, and 3 do not have a receiver overflow status bit). See the MCSPI_IRQSTATUS register.

24.4.4.7.2.1 TXx_EMPTY

The TXx_EMPTY event is activated when a channel is enabled and its MCSPI_TXx register is empty. Enabling the channel automatically raises this event. If the FIFO buffer is enabled (the MCSPI_CHxCONF[27] FFEW bit is set to 1), the TXx_EMPTY event is asserted as soon as there is enough space in buffer to write a number of bytes defined by the MCSPI_XFERLEVEL[5:0] AEL bit field.

The MCSPI_TXx register must be loaded with data to remove the source of the interrupt; the MCSPI_IRQSTATUS TXx_EMPTY interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new TXx_EMPTY event is asserted as long as the MPU has not performed the number of writes into the MCSPI_TXx register defined by MCSPI_XFERLEVEL[5:0] AEL bit field. The MPU must perform the correct number of writes.

24.4.4.7.2.2 TXx_UNDERFLOW

The TXx_UNDERFLOW event is activated when a channel is enabled and if the MCSPI_TXx register is empty (not updated with new data) when an external master device starts a data transfer with the McSPI (transmit and receive).

When FIFO is enabled, the data emitted while the underflow event is raised is not the last data written in the FIFO but the next data in the FIFO (an old transmitted value or a dummy data in the FIFO has been reset).

TXx_UNDERFLOW indicates an error (data loss) in slave mode.

To avoid having a TXx_UNDERFLOW event at the beginning of a transmission, the TXx_UNDERFLOW event is not activated when no data has been loaded into the MCSPI_TXx register because the channel is enabled.

The MCSPI_IRQSTATUS TXx_UNDERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

24.4.4.7.2.3 RXx_FULL

The RXx_FULL event is activated when a channel is enabled and the MCSPI_RXx register is being filled (transient event). When the FIFO buffer is enabled (the MCSPI_CHxCONF[28] FFER bit is set to 1), RXx_FULL is asserted as soon as the number of bytes held in the buffer to read defined by the MCSPI_XFERLEVEL[13:8] AFL bit field.

The MCSPI_RXx register must be read to remove the source of the interrupt; the MCSPI_IRQSTATUS RXx_FULL interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new RXx_FULL event is asserted as long as the MPU has not performed AFL + 1 reads into MCSPI_RXx. The MPU must perform the correct number of reads.

24.4.4.7.2.4 RX0_OVERFLOW

The RX0_OVERFLOW event is activated in slave mode in transmit-and-receive mode or receive-only mode when a channel is enabled and the MCSPI_RXx register or FIFO is full when a new SPI word is received. The MCSPI_RXx register is always overwritten with the new SPI word. If the FIFO is enabled, data within the FIFO are overwritten; it must be considered as corrupted. The RX0_OVERFLOW event should not appear in slave mode using the FIFO.

The RX0_OVERFLOW event indicates an error (data loss) in slave mode.

The MCSPI_IRQSTATUS[3] RX0_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

24.4.4.7.2.5 End Of Word Count

The MCSPI_IRQSTATUS[17] EOW event (end of word count) is activated when the channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller performs the number of transfers defined in the MCSPI_XFERLEVEL[31:16] WCNT bit field. If WCNT is set to 0x0000, the counter is not enabled and this interrupt is not generated.

The end of word count interrupt also indicates that the SPI transfer is halted on the channel using the FIFO buffer as soon as WCNT is not reloaded and the channel is not re-enabled.

The MCSPI_IRQSTATUS[17] EOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

24.4.4.7.3 Interrupt-Driven Operation

An interrupt enable bit in the MCSPI_IRQENABLE register can be set to enable each event to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the MPU must :

1. Read the MCSPI_IRQSTATUS register to identify which event occurred.
2. Read the MCSPI_RXx register that corresponds to the event to remove the source of an RXx_FULL event or write into the MCSPI_TXx register that corresponds to the event to remove the source of a TXx_EMPTY event. No action is required to remove the source of the WKS (wake-up), TXx_ UNDERFLOW, and RX0_OVERFLOW events.
3. Set the corresponding bit of the MCSPI_IRQSTATUS register to 1 to clear an interrupt status and then release the interrupt line.

The interrupt status bit must always be reset after channel enabling and before events are enabled as interrupt sources.

24.4.4.7.4 Polling

When the interrupt capability of an event is disabled in the MCSPI_IRQENABLE register, the interrupt line is not asserted, but the status bits in the MCSPI_IRQSTATUS register can be polled by software to detect when the corresponding event occurs.

Once the expected event occurs:

- RXx_FULL: To remove the source of the event, the MPU must read the corresponding MCSPI_RXx register .
- TXx_EMPTY: To remove the source of the event, the MPU must write into the corresponding MCSPI_TXx register.
- WKS (wake-up), TXx_UNDERFLOW, and RX0_OVERFLOW: No action is required to remove the source of the event.

To clear an interrupt, set the corresponding status bit of the MCSPI_IRQSTATUS register to 1. This does not affect the interrupt line state.

24.4.4.8 DMA Requests

Each McSPI channel, if enabled, can issue DMA requests. There are two DMA request lines per McSPI channel (one for read and one for write).

The DMA read request line is asserted when the McSPI channel is enabled and new data is available in the receive register of the McSPI channel. A DMA read request can be individually masked with the SPI1.MCSPI_CHxCONF[15] DMAR bit. The DMA read request line is deasserted when reading of the MCSPI_RXx register of the McSPI channel completes.

The DMA write request line is asserted when the McSPI channel is enabled and the MCSPI_TXx register of the McSPI channel is empty. A DMA write request can be individually masked with the SPI1.MCSPI_CHxCONF[14] DMAW bit. The DMA write request line is deasserted when loading of the MCSPI_TXx register of the channel completes.

24.4.4.9 Power Saving Management

Power consumption can be optimized by switching off internal clocks (interface and functional clock) when there is no activity. The McSPI is compliant with the idle and wake-up system handshake protocol.

24.4.4.9.1 Normal Mode

In normal mode, internal SPI module clocks are automatically switched off (autogated) when there is no activity in slave or master mode.

Autogating of the module interface clock and functional clock occurs when the following conditions are met:

- The MCSPI_SYSCONFIG[0] AUTOIDLE bit is set.
- In master mode, there is no data to transmit or receive in all channels.
- In slave mode, the McSPI is not selected by the external master and there are no register accesses.

Autogating of the module interface clock and functional clock stops when the following conditions are met:

- In master mode, an internal access occurs.
- In slave mode, an internal access occurs or the McSPI is selected by the external master.

24.4.4.9.2 Idle Mode

At the PRCM module level, when all conditions are met to shut off the PER_48M_GFCLK or L4PER_L3_GICLK output clock, the PRCM module automatically launches a hardware handshake protocol to ensure that the McSPI is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the McSPI.

Although this handshake is completely hardware-oriented and out of software control, the method in which the McSPI module acknowledges the PRCM IDLE request is configurable through the MCSPI_SYSCONFIG[4:3] SIDLEMODE bit field.

The settings of the SIDLEMODE bit field and the related acknowledgment modes are:

- Force-idle mode (the MCSPI_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0): The McSPI module acknowledges unconditionally the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully in this case because it does not prevent the loss of data when the clock is switched off.
- No-idle mode (the SIDLEMODE bit field is set to 0x1): The McSPI never acknowledges an IDLE request from the PRCM module and is safe from a module point of view because it ensures that the clocks remain active. However, it is not efficient to save power because it does not allow the PRCM output clock to be shut off and thus the power domain to be set to a lower power state.
- Smart-idle mode (the SIDLEMODE bit field is set to 0x2): The McSPI acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, IRQs, or DMA requests are treated. This is the best approach for efficient system power management.

When configured in smart-idle mode, the McSPI also offers an additional granularity on the PER_48M_GFCLK and L4PER_L3_GICLK gating. The MCSPI_SYSCONFIG[9:8] CLOCKACTIVITY bit field determines which clock shuts down (the PER_48M_GFCLK, L4PER_L3_GICLK, neither clock, or both clocks).

The setting of the CLOCKACTIVITY bit field is used internally to the McSPI to determine on which part of the module the conditions to acknowledge the PRCM IDLE request are tested. For example, if PER_48M_GFCLK is not shut down on a PRCM IDLE request, the McSPI considers only L4PER_L3_GICLK and the associated pending activities before acknowledging the request.

Some McSPI features are associated with L4PER_L3_GICLK and others with PER_48M_GFCLK. Using the CLOCKACTIVITY bit field with the smart-idle mode ensures that the features associated with the clock that remains active are always enabled, even if the McSPI acknowledges an IDLE request.

The settings of the CLOCKACTIVITY bit field and the associated features are:

- CLOCKACTIVITY set to 00: ICLK off and FCLK off, ICLK and FCLK are considered for generating the acknowledge. This setting also means that FCLK and ICLK are likely to be shut down on a PRCM IDLE request.
- CLOCKACTIVITY set to 01: ICLK on and FCLK off, ICLK is not shut down on a PRCM IDLE request; only FCLK is concerned.
- CLOCKACTIVITY set to 10: ICLK off and FCLK on, FCLK is not shut down on a PRCM IDLE request; only ICLK is concerned.
- CLOCKACTIVITY set to 11: ICLK on and FCLK on, none of the clocks are shut down. This means the McSPI can potentially acknowledge the IDLE request without checking the internal functions linked to its clocks.

CAUTION

The PRCM module does not have a hardware means of reading the CLOCKACTIVITY settings. Therefore, software must ensure consistent programming between CLOCKACTIVITY and the PER_48M_GFCLK and L4PER_L3_GICLK control bits in the PRCM module. If the McSPI is disabled in the CM_FCLKEN and CM_ICLKEN PRCM registers while CLOCKACTIVITY is set to 11, nothing prevents the PRCM module from asserting its IDLE request, which is acknowledged regardless of the features associated with the McSPI clocks. This can lead to unpredictable behavior.

24.4.4.9.2.1 Wake-Up Event in Smart-Idle Mode

The module wake-up feature is enabled when the MCSPI_SYSCONFIG[2] ENAWAKEUP and MCSPI_WAKEUPENABLE[0] WKEN bits are set. Wake-up capability is relevant only when the module is configured in slave mode.

The module generates an asynchronous wake-up request to the system power manager to switch back the interface clock and the functional clock. A wakeup is requested when channel 0 is enabled and an asynchronous selection occurs on the mcspim.csx port associated with channel 0 (see the definition for the MCSPI_CHxCONF[22:21] SPIENSLV bit field [where x = 0] in the register table description).

After the McSPI wake-up request, the system power manager must reactivate the interface clock:

- Before the beginning of the second SPI word serialization when the McSPI is in slave transmit-only mode or in slave transmit-and-receive mode
- Before the end of the second received SPI word in slave receive-only mode. To avoid data loss, the first received SPI word must be read from the MCSPI_RXx register (where x = 0) before the completion of the second SPI word serialization.

Table 24-207 lists the supported cases in smart-idle mode.

Table 24-207. Smart-Idle Mode and Wake-Up Capabilities

Mode	Interface Clock	SPI Clock Ref	Functionality	Wake-Up Event
Master	Must be maintained	Must be maintained	Full functionality, but the module does not generate a new interrupt or DMA request until the system exits wake-up mode	No wake-up event
Slave	Can be switched off	Can be switched off	An SPI word can be transmitted and/or received, but the module does not generate any new interrupts or DMA requests until the system exits wake-up mode.	The module asynchronously sends a wake-up request if an event on the SPIEN[x] port associated with channel 0 is detected.

In wake-up mode, the interrupt and DMA request lines are no longer asserted.

Any access to the module in wake-up mode generates an error as long as the interface clock is alive.

24.4.4.9.2.2 Transitions From Smart-Idle Mode to Normal Mode

The McSPI detects the end of the wake period through the idle and wake-up hardware handshake protocol.

The interrupt status register (the MCSPI_IRQSTATUS[16] WKS bit) is updated with the event causing the wakeup; the wake-up event at the origin of the transition to the normal mode is converted to its corresponding interrupt when enabled by the MCSPI_IRQENABLE[16] WKE bit or the DMA request.

Interrupts and wake-up events have independent enable and disable controls, accessible through the MCSPI_IRQENABLE and MCSPI_WAKEUPENABLE registers. Software must ensure the overall consistency.

The interrupt status register MCSPI_IRQSTATUS is updated with the event causing the wakeup; the wake-up event at the origin of the transition to normal mode is converted to its corresponding interrupt request or DMA request. The module is fully operational.

24.4.4.9.2.3 Force-Idle Mode

Force-idle mode is enabled and exited as follows:

- Force-idle mode is enabled when the MCSPI_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0.
  - In force-idle mode, the McSPI responds unconditionally to the IDLE request by deasserting unconditionally the interrupt and DMA request lines, if asserted. In addition, the wake-up capability is totally inhibited even if the MCSPI_SYSCONFIG[2] ENAWAKEUP and MCSPI_WAKEUPENABLE[0] WKEN bits are set.
  - The transition from normal mode to idle mode does not affect the interrupt event bits of the MCSPI_IRQSTATUS register.
  - In force-idle mode, because the module must be disabled, the interrupt and DMA request lines are likely deasserted. The interface clock and SPI clock provided to the McSPI can be switched off.
  - An IDLE request during an SPI data transfer can lead to an unexpected and unpredictable result. Software must avoid such a request.
- The module exits force-idle mode through the idle and wake-up hardware handshake protocol.

  The module is fully operational. The interrupt and DMA request lines are optionally asserted one clock cycle later.

24.4.5 McSPI Programming Guide

This section describes the low-level hardware programming sequences for the configuration and use of the McSPI module.

24.4.5.1 Global Initialization

24.4.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the McSPI module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the McSPI. For further information, see Section 24.4.3, *McSPI Integration* and Section 24.4.2, *McSPI Environment*.

Table 24-208 lists the information on the global initialization of the surrounding modules.

Table 24-208. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	MCSPI_FCLK functional clock must be enabled. See *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.
L4 Interconnect	For information about L4_PER1 interconnect configuration, see *L4 Interconnects*, in *Interconnect*.
DMA_CROSSBAR	DMA_CROSSBAR configuration must be done to allow module DREQs to be mapped to certain device DMA line. For more information see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
Device DMAs	Device DMAs configuration must be done to enable the module DMA channel requests.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see *Interrupt Controllers*

24.4.5.1.2 McSPI Global Initialization

24.4.5.1.2.1 Main Sequence – McSPI Global Initialization

The procedure in Table 24-209 can be used to initialize McSPI when performing software reset.

Table 24-209. McSPI Global Initialization

Step	Register/Bit Field/Programming Model	Value
Perform a software reset.	MCSPI_SYSCONFIG[1] SOFTRESET	1
Wait until reset is finished?	MCSPI_SYSSTATUS[0] RESETDONE	=1
Configure static settings (such as SPI master or slave) as required.	MCSPI_MODULCTRL[8:0]	0x-
Write MCSPI_SYSCONFIG	MCSPI_SYSCONFIG	0x-

24.4.5.2 Operational Mode Configuration

24.4.5.2.1 McSPI Operational Modes

The selection of the working mode is done with the MCSPI_CHxCONF register.

Table 24-210. McSPI Receive Mode Initialization

Step	Register/Bit Field/Programming Model	Value
Set receive mode for the channel.	MCSPI_CHxCONF[13:12] TRM	0x1
Configure SPI clock polarity/phase, clock divider, word length, and others for the channel.	MCSPI_CHxCONF	0x-

Table 24-210. McSPI Receive Mode Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Reset the status bits.	MCSPI_IRQSTATUS	0x0

Table 24-211. McSPI Transmit Mode Initialization

Step	Register/Bit Field/Programming Model	Value
Set transmit mode for the channel.	MCSPI_CHxCONF[13:12] TRM	0x2
Configure SPI clock polarity/phase, clock divider, word length, and others for the channel.	MCSPI_CHxCONF	0x-
Reset the status bits.	MCSPI_IRQSTATUS	0x0

Table 24-212. McSPI Transmit-and-Receive Mode Initialization

Step	Register/Bit Field/Programming Model	Value
Set transmit and receive mode for the channel.	MCSPI_CHxCONF[13:12] TRM	0x0
Configure SPI clock polarity/phase, clock divider, word length, and others for the channel.	MCSPI_CHxCONF	0x-
Reset the status bits.	MCSPI_IRQSTATUS	0x0

24.4.5.2.1.1 Common Transfer Sequence

McSPI module allows the transfer of one or several words, according to different modes:

- MASTER Normal, MASTER Turbo, SLAVE
- TRANSMIT–RECEIVE, TRANSMIT-ONLY, RECEIVE-ONLY
- Write and Read requests: Interrupts, DMA
- SPIEN[x] lines assertion/deassertion: automatic, manual

For all these sequences, the host process contains the main process and the interrupt routines.

The interrupt routines are called on the interrupt signals or by an internal call if the module is used in polling mode.

Table 24-213 represents the main sequence which is common to all transfers.

In multi-channel master mode, the sequences of different channels can be run simultaneously.

Table 24-213. Common Transfer Sequence (Main Process)

Step	Register/Bit Field/Programming Model	Value
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
Write MCSPI_IRQENABLE to enable interrupts	MCSPI_IRQENABLE	0x-
Write MCSPI_CHxCONF to configure the channel	MCSPI_CHxCONF	0x-
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait for the first write request (TX empty or DMA write)		
Write the transmitter register with data	MCSPI_TXx	0x-
Wait for the host event for end of transfer		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

24.4.5.2.1.2 End of Transfer Sequences

The end of transfer depends on the transfer mode. Table 24-214 summarizes the type of end of transfer per transfer mode and gives a reference to the appropriate section for details.

Table 24-214. End of Transfer Sequences

		TRANSMIT-AND-RECEIVE		TRANSMIT-ONLY		RECEIVE-ONLY	
		INTERRUPT	DMA	INTERRUPT	DMA	INTERRUPT	DMA
MASTER Normal	End of transfer sequence	See Section 24.4.5.2.1.3		See Section 24.4.5.2.1.4.1	See Section 24.4.5.2.1.4.2	See Section 24.4.5.2.1.5.1	See Section 24.4.5.2.1.5.2
	Minimum number of word	1	1	1	1	1	2
	DMA transfer size		N		N		N-1
MASTER Turbo	End of transfer sequence	See Section 24.4.5.2.1.3		See Section 24.4.5.2.1.4.1	See Section 24.4.5.2.1.4.2	See Section 24.4.5.2.1.6.1	See Section 24.4.5.2.1.6.2
	Minimum number of word	1	1	1	1	2	3
	DMA transfer size		N		N		N-2
SLAVE	End of transfer sequence	See Section 24.4.5.2.1.3		See Section 24.4.5.2.1.4.1	See Section 24.4.5.2.1.4.2	See Section 24.4.5.2.1.7	
	Minimum number of word	1	1	1	1	1	1
	DMA transfer size		N		N	N	N

The transfer to execute has a size of N words.

The different sequences can be merged in one process to manage transfers of several types. The end of transfer sequences are described from the start of the channel.

In these sequences, some soft variables are used:

- write_count = 0
- read_count = 0
- channel_enable = FALSE
- last_transfer = FALSE
- last_request = FALSE

They are initialized before starting the channel.

24.4.5.2.1.3 Transmit-and-Receive (Master and Slave)

If the requests are configured in DMA, write_count and read_count are assigned with 'N' when the DMA handlers have completed their 'N' OCP accesses.

Table 24-215. Transmit-and-Receive (Master and Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait for write_count = N **AND** read_count = N		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 24-216. Transmit-and-Receive (Master and Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: TXx_EMPTY		
Write the transmitter register with data	MCSPI_TXx	0x-
Increment write_count +1		
IF: RXx_FULL		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		

Table 24-216. Transmit-and-Receive (Master and Slave) (Interrupt Routine) (continued)

Step	Register/Bit Field/Programming Model	Value
ENDIF		

24.4.5.2.1.4 Transmit-Only (Master and Slave)

24.4.5.2.1.4.1 Based on Interrupt Requests

Table 24-217. Transmit-Only With Interrupts (Master and Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until last transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 24-218. Transmit-Only With Interrupts (Master and Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: TXx_EMPTY **AND** write_count < N		
Write the transmitter register with data	MCSPI_TXx	0x-
Increment write_count +1		
ELSEIF: write_count ≥ N		
last_transfer = TRUE		
ENDIF		

24.4.5.2.1.4.2 Based on DMA Write Requests

When the DMA handler has completed its ‘N’ OCP accesses, write_count is assigned with ‘N’.

Table 24-219. Transmit-Only With DMA (Master and Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until write_count = N		
Disable DMA write request	MCSPI_CHxCONF[14] DMAW	0
Wait until last_transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 24-220. Transmit-Only With DMA (Master and Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: TXx_EMPTY **AND** write_count = N		
last_transfer = TRUE		
ENDIF		

24.4.5.2.1.5 Master Normal Receive-Only

24.4.5.2.1.5.1 Based on Interrupt Requests

Table 24-221. Receive-Only With Interrupt (Master Normal) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1

Table 24-221. Receive-Only With Interrupt (Master Normal) (Main Process) (continued)

Step	Register/Bit Field/Programming Model	Value
Wait until last_request = TRUE		
Stop the channel	MCSPI_CHxCTRL[0] EN	0
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		

Table 24-222. Receive-Only With Interrupt (Master Normal) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL **AND** read_count = N - 1		
last_request = TRUE		
ELSEIF: read_count ≠ N - 1		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		

24.4.5.2.1.5.2 Based on DMA Read Requests

When the DMA handler has completed its ‘N-1’ OCP accesses, read_count is assigned with ‘N-1’.

Table 24-223. Receive-Only With DMA (Master Normal) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until read_count = N - 1		
Disable DMA read request	MCSPI_CHxCONF[15] DMAR	0
Wait until last_transfer = TRUE		
Stop the channel	MCSPI_CHxCTRL[0] EN	0
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		

Table 24-224. Receive-Only With DMA (Master Normal) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL **AND** read_count = N		
last_transfer = TRUE		
ENDIF		

24.4.5.2.1.6 Master Turbo Receive-Only

24.4.5.2.1.6.1 Based on Interrupt Requests

Table 24-225. Receive-Only With Interrupt (Master Turbo) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until channel_enable = TRUE		
Wait until last_transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 24-225. Receive-Only With Interrupt (Master Turbo) (Main Process) (continued)

Step	Register/Bit Field/Programming Model	Value
Wait until channel_enable = FALSE		

Table 24-226. Receive-Only With Interrupt (Master Turbo) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL		
IF: read_count = N - 2		
last_transfer = TRUE		
Wait until channel_enable = FALSE		
ENDIF		
IF: read_count < N		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		
ENDIF		

24.4.5.2.1.6.2 Based on DMA Read Requests

When the DMA handler has completed its ‘N-2’ OCP accesses read_count is assigned with ‘N-2‘.

Table 24-227. Receive-Only With DMA (Master Turbo) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until channel_enable = TRUE		
Wait until read_count = TRUE		
Disable DMA read request	MCSPI_CHxCONF[15] DMAR	0
Wait until last_transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0
Wait until channel_enable = FALSE		

Table 24-228. Receive-Only With DMA (Master Turbo) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL		
IF: read_count = N - 2		
last_transfer = TRUE		
Wait until channel_enable = FALSE		
ENDIF		
IF: read_count < N		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		
ENDIF		

24.4.5.2.1.7 Slave Receive-Only

If the requests are configured in DMA, read_count is assigned with 'N' when the DMA handler has completed its 'N' OCP accesses.

Table 24-229. Receive-Only (Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until read_count = N		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 24-230. Receive-Only (Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		

24.4.5.2.1.8 Transfer Procedures With FIFO

These flows describe the transfer with FIFO.

The McSPI module allows the transfer of one or several words, according to different modes:
- MASTER Normal, MASTER Turbo, SLAVE
- TRANSMIT–RECEIVE, TRANSMIT-ONLY, RECEIVE-ONLY
- Write and Read requests: IRQ, DMA

For all these flows, the host process contains the main process and the interrupt routine. This routine is called on the IRQ signals or by an internal call if the module is used in polling mode.

For more information, see Section 24.4.4.6, *FIFO Buffer Management*.

Table 24-231. FIFO Mode Common Sequence (Master) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS	1
Write MCSPI_IRQENABLE to enable interrupts	MCSPI_IRQENABLE	1
Write MCSPI_CHxCONF to configure the channel	MCSPI_CHxCONF	0x-
Write MCSPI_XFERLEVEL	MCSPI_XFERLEVEL	0x-
Start the channel	MCSPI_CHxCTRL[0] EN	1
IF: Receive only		
Wait for the write request (TX empty or DMA write)		
Write for the transmitter register with data	MCSPI_TXx	0x-
ENDIF		
Wait for the host event for end of transfer		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

24.4.5.2.1.8.1 Common Transfer Sequence in FIFO Mode

This flow describes the host sequence for a transfer of any type defined in Section 24.4.5.2.1.8, *Transfer Procedures With FIFO*.

In multi-channel, only one channel can use the FIFO.

Before enabling the FIFO for a channel (MCSPI_CHxCONF[28] FFER and MCSPI_CHxCONF[27] FFEW bits), the host must check that the FIFO is not enabled for another channel, even if these channels are not used.

In transmit-and-receive mode, the FIFO can be enabled for write or read request only, without FIFO for the other request.

In Slave mode, the channel 0 only can be activated. The correct SPIEN line is chosen in MCSPI_CHxCONF[22:21] SPIENSLV bits (where x = 0).

The McSPI module can start the transfer only when the first write request has been released by writing the MCSPI_TXx register, even in receive-only mode (only one write request occurs in this case).

24.4.5.2.1.8.2 End of Transfer Sequences in FIFO Mode

Table 24-232 summarizes the type of end of transfer per transfer mode and gives a reference to the appropriate section for details.

Table 24-232. End of Transfer Sequences in FIFO Mode

Word count	TRANSMIT AND RECEIVE	TRANSMIT-ONLY	RECEIVE-ONLY
Yes	See Figure 24-96	See Figure 24-98	See Figure 24-99
No	See Figure 24-97	See Figure 24-98	See Figure 24-100

The end of transfer sequences are described from the start of the channel.

In these sequences, some soft variables are used:
- write_count = N
- read_count = N
- last_request = FALSE

They are initialized before starting the channel.

24.4.5.2.1.8.3 Transmit-and-Receive With Word Count

Figure 24-96 shows the flow of a transfer in transmit-and-receive mode, with word count.

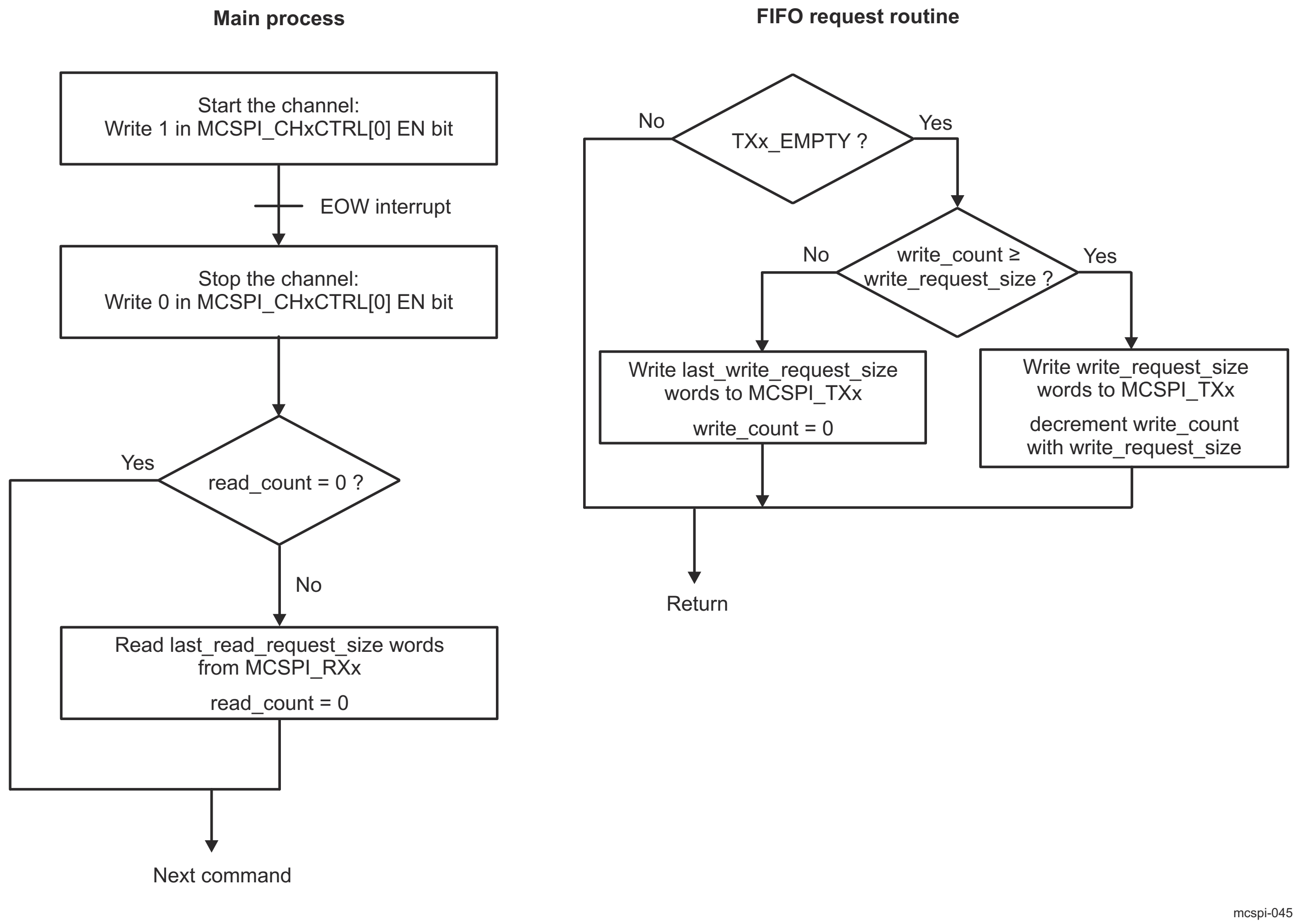


Figure 24-96. FIFO Mode Transmit-and-Receive With Word Count (Master)

24.4.5.2.1.8.4 Transmit-and-Receive Without Word Count

Figure 24-97 shows the flow of a transfer in transmit-and-receive mode, without word count.

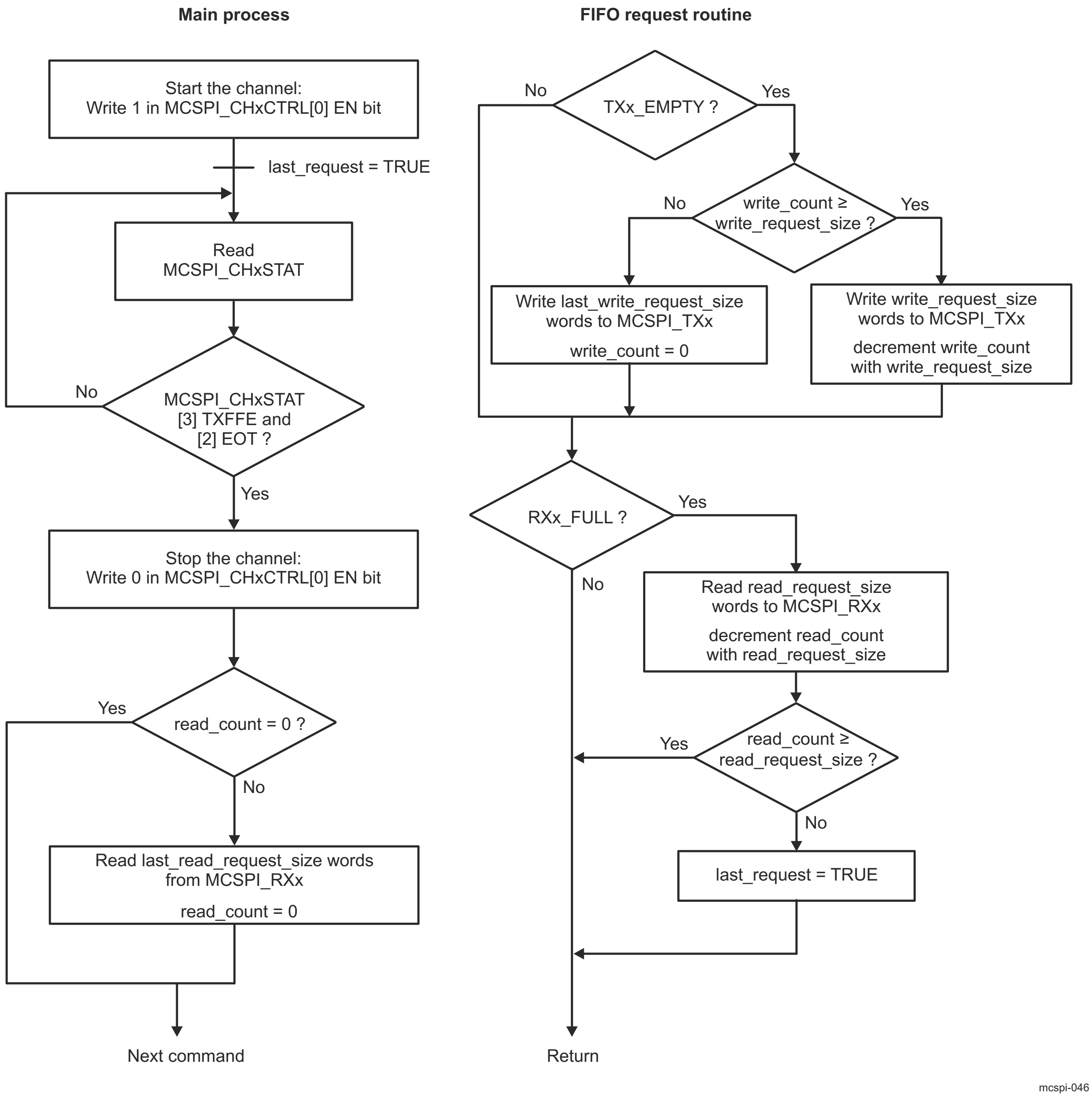


Figure 24-97. FIFO Mode Transmit-and-Receive Without Word Count (Master)

24.4.5.2.1.8.5 Transmit-Only

Figure 24-98 shows the flow of a transfer in transmit-only mode, with or without word count. The difference between word count enabled or not is just on the condition after starting the channel:

- word count enable: wait for EOW interrupt
- word count disable: wait for write_count = 0

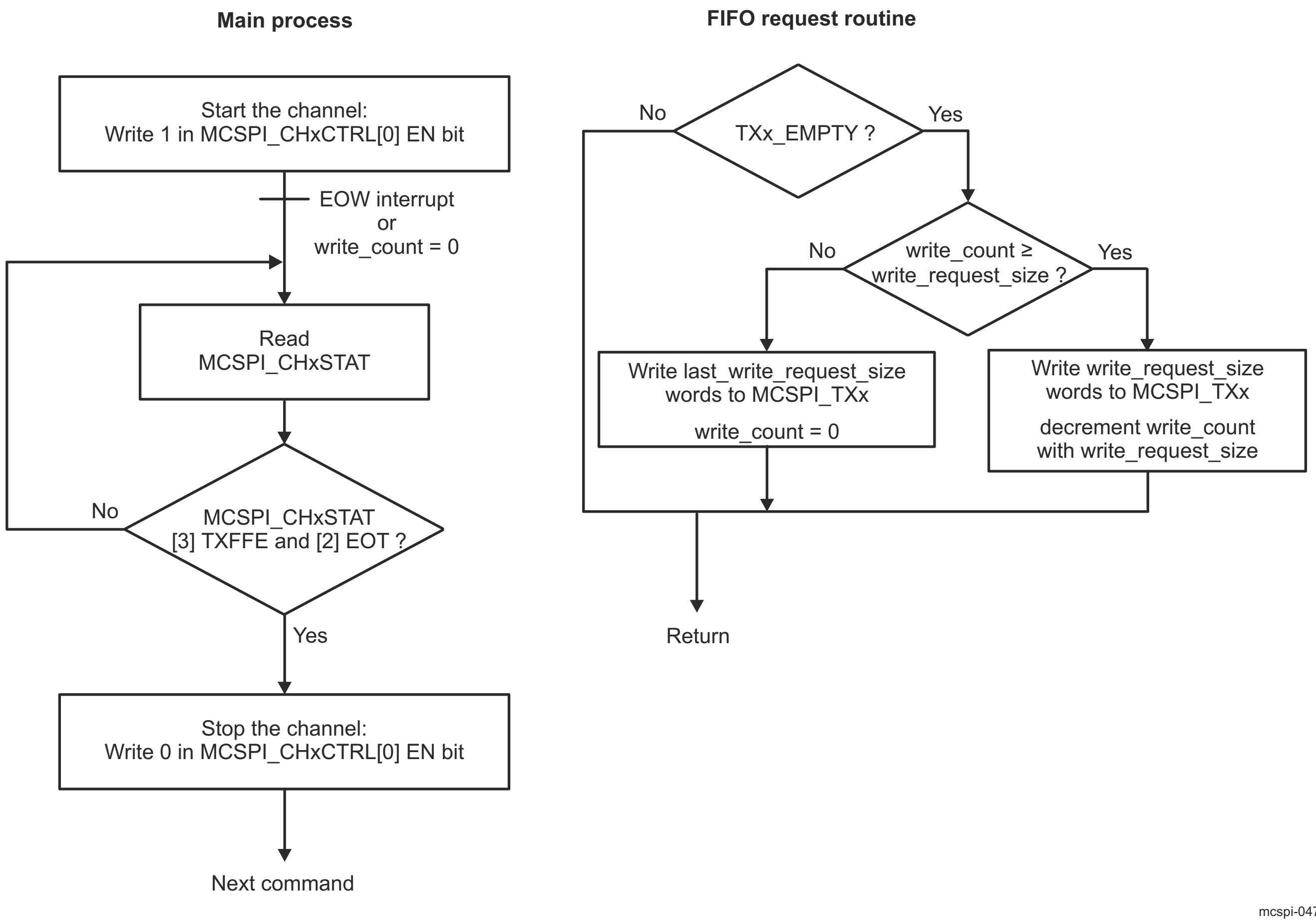


Figure 24-98. FIFO Mode Transmit-Only (Master)

24.4.5.2.1.8.6 Receive-Only With Word Count

Figure 24-99 shows the flow of a transfer in receive-only mode, with word count.

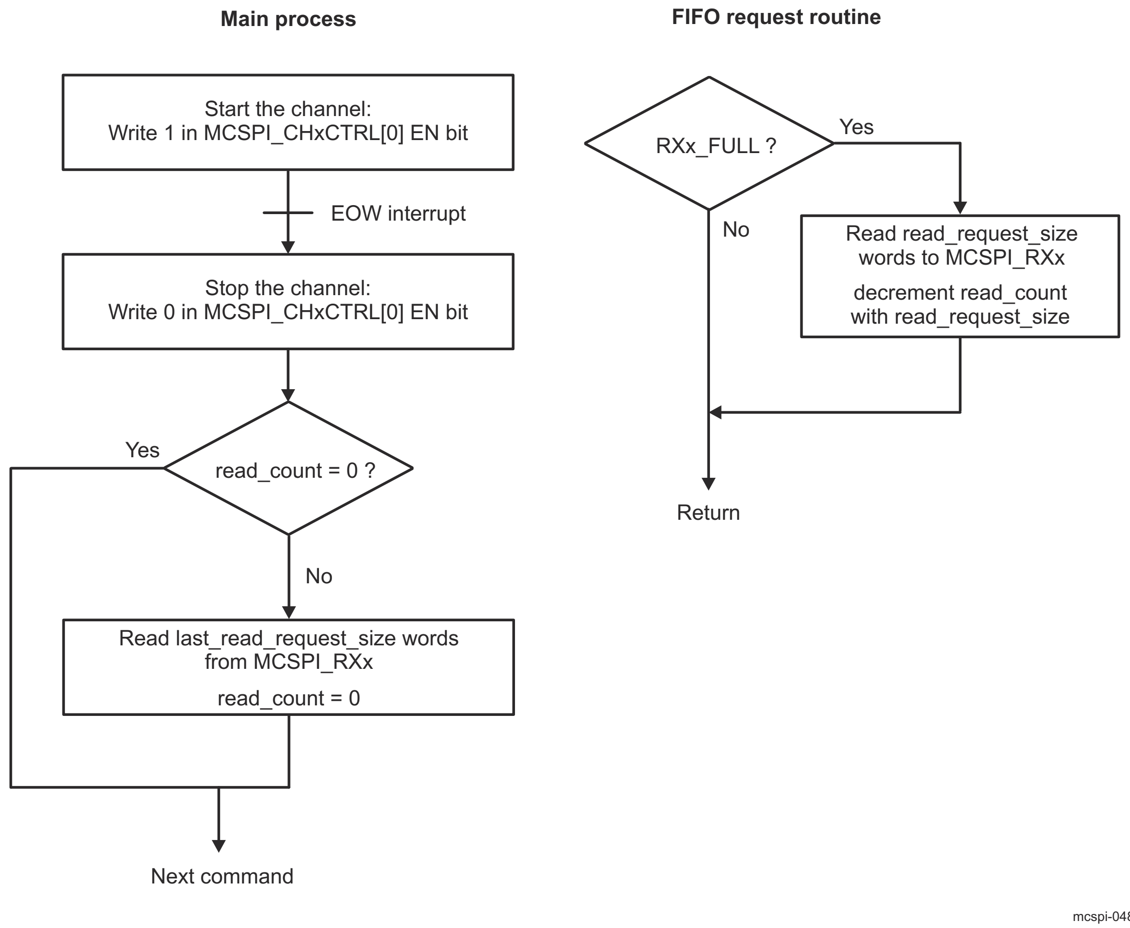


Figure 24-99. FIFO Mode Receive-Only With Word Count (Master)

24.4.5.2.1.8.7 Receive-Only Without Word Count

Figure 24-100 shows the flow of a transfer in receive-only mode, without word count.

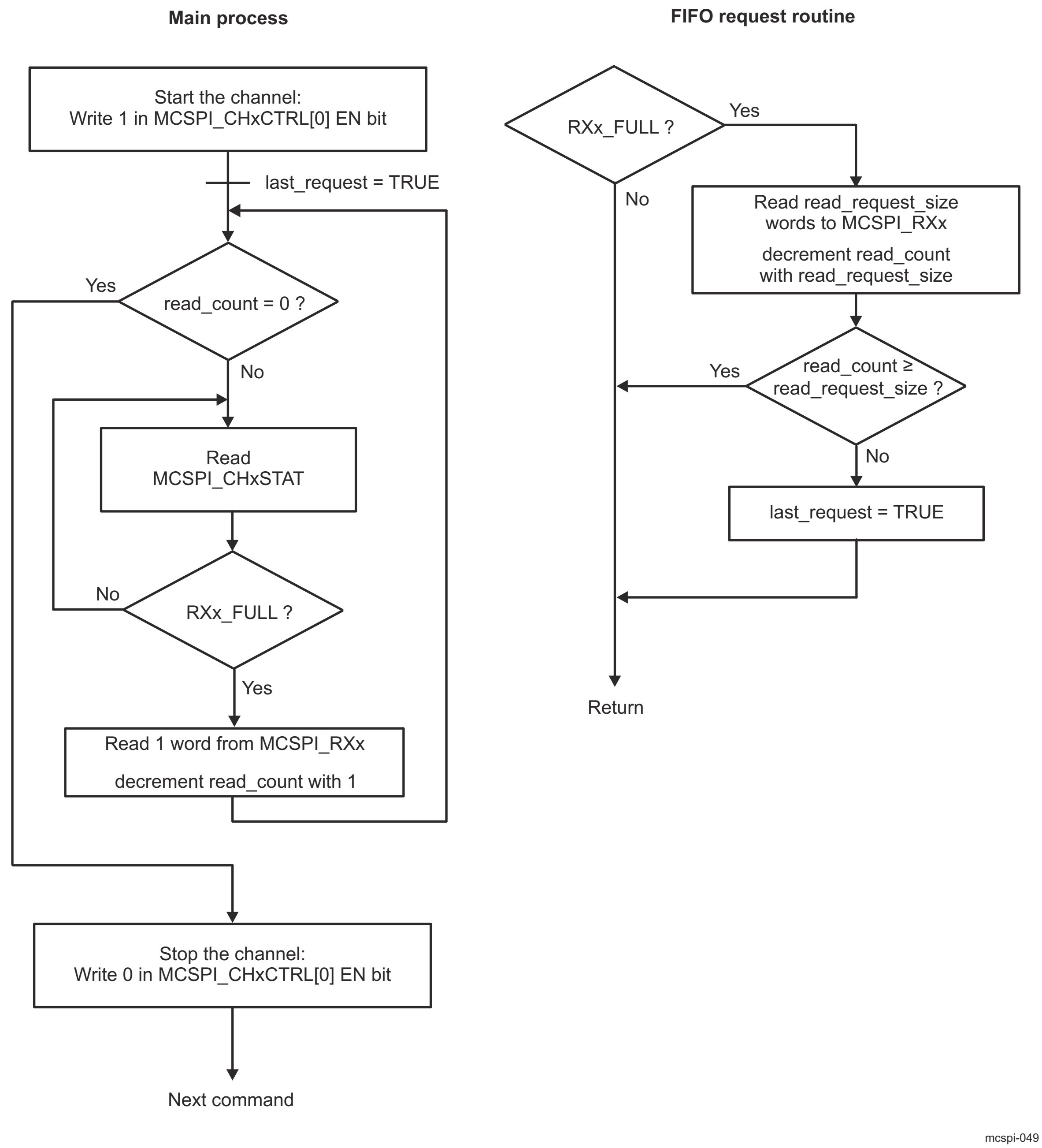


Figure 24-100. FIFO Mode Receive-Only Without Word Count (Master)

24.4.5.3 Common Transfer Procedures Without FIFO – Polling Method

24.4.5.3.1 Receive-Only Procedure – Polling Method

Table 24-233 lists the receive-only procedure using the polling method.

Table 24-233. Receive-Only Procedure – Polling Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode	See Table 24-210.	
Start the channel.	MCSPI_CHxCTRL[0] EN	1
Wait for end-of-transfer.	MCSPI_CHxSTAT[2] EOT	=1
Read the receiver register.	MCSPI_RXx	0x-

Table 24-233. Receive-Only Procedure – Polling Method (continued)

Step	Register/Bit Field/Programming Model	Value
Stop the channel if no more data is expected.	MCSPI_CHxCTRL[0] EN	0

24.4.5.3.2 Receive-Only Procedure – Interrupt Method

Table 24-234 lists the receive-only procedure using the interrupt method.

Table 24-234. Receive-Only Procedure – Interrupt Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 24-210.	
Start the channel.	MCSPI_CHxCTRL[0] EN	1
Enable the interrupt for the receiver register.	MCSPI_IRQENABLE[2] RX_FULL_ENABLE	1
Wait for interrupt.		
Read the status register.	MCSPI_IRQSTATUS[2] RX_FULL	1
Disable the interrupt if no more data is expected.	MCSPI_IRQENABLE[2] RX_FULL_ENABLE	0
Stop the channel if no more data is expected.	MCSPI_CHxCTRL[0] EN	0
Read the receiver register.	MCSPI_RXx	0x-

24.4.5.3.3 Transmit-Only Procedure – Polling Method

Table 24-235 lists the transmit-only procedure using the polling method.

Table 24-235. Transmit-Only Procedure – Polling Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 24-211	
Start the channel.	MCSPI_CHxCTRL[0] EN	1
Write the transmitter register with data.	MCSPI_TXx	0x-
Wait until end of transfer?	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel.	MCSPI_CHxCTRL[0] EN	0

24.4.5.3.4 Transmit-and-Receive Procedure – Polling Method

Table 24-235 lists the transmit-and-receive procedure using the polling method.

Table 24-236. Transmit-and-Receive Procedure – Polling Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 24-212.	
Start the channel.	MCSPI_CHxCTRL[0] EN	1
Write the transmitter register with data.	MCSPI_TXx	0x-
Wait until transmit/receive word?	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel.	MCSPI_CHxCTRL[0] EN	0
Read the receiver register.	MCSPI_TXx	0x-

24.4.6 McSPI Register Manual

24.4.6.1 McSPI Instance Summary

Table 24-237. McSPI Instance Summary

Module Name	Base Address	Size
McSPI1	0x4809 8000	4 KiB
McSPI2	0x4809 A000	4 KiB
McSPI3	0x480B 8000	4 KiB
McSPI4	0x480B A000	4 KiB

24.4.6.2 McSPI Registers

24.4.6.2.1 McSPI Register Summary

Table 24-238 lists the McSPI registers. Each register is 32 bits wide.

Table 24-238. McSPI Register Summary

Register	Type	Offset Address	McSPI1 Physical Address	McSPI2 Physical Address	McSPI3 Physical Address	McSPI4 Physical Address
MCSPI_HL_REV	RW	0x00	0x4809 8000	0x4809 A000	0x480B 8000	0x480B A000
MCSPI_HL_HWINFO	RW	0x04	0x4809 8004	0x4809 A004	0x480B 8004	0x480B A004
MCSPI_HL_SYSCONFIG	RW	0x10	0x4809 8010	0x4809 A010	0x480B 8010	0x480B A010
MCSPI_REVISION	R	0x100	0x4809 8100	0x4809 A100	0x480B 8100	0x480B A100
MCSPI_SYSCONFIG	RW	0x110	0x4809 8110	0x4809 A110	0x480B 8110	0x480B A110
MCSPI_SYSSTATUS	R	0x114	0x4809 8114	0x4809 A114	0x480B 8114	0x480B A114
MCSPI_IRQSTATUS	RW	0x118	0x4809 8118	0x4809 A118	0x480B 8118	0x480B A118
MCSPI_IRQENABLE	RW	0x11C	0x4809 811C	0x4809 A11C	0x480B 811C	0x480B A11C
MCSPI_WAKEUPENABLE	RW	0x120	0x4809 8120	0x4809 A120	0x480B 8120	0x480B A120
MCSPI_SYST	RW	0x124	0x4809 8124	0x4809 A124	0x480B 8124	0x480B A124
MCSPI_MODULCTRL	RW	0x128	0x4809 8128	0x4809 A128	0x480B 8128	0x480B A128
MCSPI_CHxCONF (1)	RW	0x12C + (0x14 * x)	0x4809 812C + (0x14 * x)	0x4809 A12C + (0x14 * x)	0x480B 812C + (0x14 * x)	0x480B A12C + (0x14 * x)
MCSPI_CHxSTAT (1)	R	0x130 + (0x14 * x)	0x4809 8130 + (0x14 * x)	0x4809 A130 + (0x14 * x)	0x480B 8130 + (0x14 * x)	0x480B A130 + (0x14 * x)
MCSPI_CHxCTRL (1)	RW	0x134 + (0x14 * x)	0x4809 8134 + (0x14 * x)	0x4809 A134 + (0x14 * x)	0x480B 8134 + (0x14 * x)	0x480B A134 + (0x14 * x)
MCSPI_TXx (1)	RW	0x138 + (0x14 * x)	0x4809 8138 + (0x14 * x)	0x4809 A138 + (0x14 * x)	0x480B 8138 + (0x14 * x)	0x480B A138 + (0x14 * x)
MCSPI_RXx (1)	R	0x13C + (0x14 * x)	0x4809 813C + (0x14 * x)	0x4809 A13C + (0x14 * x)	0x480B 813C + (0x14 * x)	0x480B A13C + (0x14 * x)
MCSPI_XFERLEVEL	RW	0x17C	0x4809 817C	0x4809 A17C	0x480B 817C	0x480B A17C
MCSPI_DAFTX	RW	0x0000 0180	0x4809 8180	0x4809 A180	0x480B 8180	0x480B A180
MCSPI_DAFRX	RW	0x0000 01A0	0x4809 81A0	0x4809 A1A0	0x480B 81A0	0x480B A1A0

(1) x = 0 to 3 for McSPI1, McSPI2, McSPI3 and McSPI4

24.4.6.2.2 McSPI Register Description

Table 24-239 through Table 24-275 describe the individual McSPI register bits.

Table 24-239. MCSPI_HL_REV

Address Offset	0x00		
Physical Address	0x4809 8000 0x4809 A000 0x480B 8000 0x480B A000	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	McSPI module revision identifier Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	McSPI Module Revision	R	TI internal data

Table 24-240. Register Call Summary for Register MCSPI_HL_REV

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]

Table 24-241. MCSPI_HL_HWINFO

Address Offset	0x04		
Physical Address	0x4809 8004 0x4809 A004 0x480B 8004 0x480B A004	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	Information about the module's hardware configuration.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									RETMODE	FFNBYTE					USEFIFO

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved These bits are initialized to 0, and writes to them are ignored.	R	0x0000000
6	RETMODE	Retention Mode. This bit field indicates whether the retention mode is supported. 0x0: Retention mode disabled 0x1: Retention mode enabled	R	0
5:1	FFNBYTE	FIFO number of bytes parameter Read 0x1: FIFO 16 bytes depth Read 0x2: FIFO 32 bytes depth Read 0x4: FIFO 64 bytes depth Read 0x8: FIFO 128 bytes depth Read 0x10: FIFO 256 bytes depth	R	0x8

Bits	Field Name	Description	Type	Reset
0	USEFIFO	Use of a FIFO enable. This bit indicates if a FIFO is integrated within controller design with its management. Read 0x0: FIFO not implemented in design Read 0x1: FIFO and its management implemented in design	R	1

Table 24-242. Register Call Summary for Register MCSPI_HL_HWINFO

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]

Table 24-243. MCSPI_HL_SYSCONFIG

Address Offset	0x10		
Physical Address	0x4809 8010 0x4809 A010 0x480B 8010 0x480B A010	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	Clock management configuration		
Type	RW		

31:4	3:2	1	0
RESERVED	IDLEMODE	FREEEMU	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's IDLE state follows (acknowledges) the system's IDLE requests unconditionally, that is, regardless of the module's internal requirements. Backup mode, for debug only. 0x1: No-idle mode: local target never enters IDLE state. Backup mode, for debug only. 0x2: Smart-idle mode: local target's IDLE state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements. Module shall not generate (IRQ- or DMA-request-related) wake-up events. 0x3: Smart-idle wake-up-capable mode: local target's IDLE state eventually follows (acknowledges) the system's IDLE requests, depending on the module's internal requirements. Module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. Mode is relevant only if the appropriate IP module "swake-up" output(s) is (are) implemented.	RW	0x2
1	FREEEMU	Sensitivity to emulation (debug) suspend input signal. 0x0: Module is sensitive to emulation suspend. 0x1: Module is not sensitive to emulation suspend.	RW	0

Bits	Field Name	Description	Type	Reset
0	SOFTRESET	Software reset. (Optional) Write 0x0: No action Read 0x0: Reset done, no pending action Read 0x1: Reset (software or other) ongoing Write 0x1: Initiate software reset	RW	0

Table 24-244. Register Call Summary for Register MCSPI_HL_SYSCONFIG

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]

Table 24-245. MCSPI_REVISION

Address Offset	0x100		
Physical Address	0x4809 8100 0x4809 A100 0x480B 8100 0x480B A100	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains the McSPI revision number.		
Type	R		

31:8	7:0
RESERVED	REVISION

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads return 0	R	0x0
7:0	REVISION	McSPI core revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0, 0x21 for 2.1	R	TI Internal data

Table 24-246. Register Call Summary for Register MCSPI_REVISION

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]

Table 24-247. MCSPI_SYSCONFIG

Address Offset	0x110		
Physical Address	0x4809 8110 0x4809 A110 0x480B 8110 0x480B A110	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register allows controlling various parameters of the configuration interface and is not affected by software reset.		
Type	RW		

31:10	9:8	7:5	4:3	2	1	0
RESERVED	CLOCKACTIVITY	RESERVED	SIDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reads return 0	RW	0x000000

Bits	Field Name	Description	Type	Reset
9:8	CLOCKACTIVITY	Clocks activity during wake-up mode period 0x0: Interface and functional clocks may be switched off. 0x1: Interface clock is maintained. Functional clock may be switched off. 0x2: Functional clock is maintained. Interface clock may be switched off. 0x3: Interface and functional clocks are maintained.	RW	0x0
7:5	RESERVED	Reads returns 0	RW	0x0
4:3	SIDLEMODE	Power management 0x0: If an IDLE request is detected, the McSPI acknowledges it unconditionally and goes in inactive mode. Interrupt, DMA requests and wake-up lines are unconditionally deasserted and the module wake-up capability is deactivated even if the [2] ENAWAKEUP bit is set. 0x1: If an IDLE request is detected, the request is ignored and the module does not switch to wake-up mode, and keeps on behaving normally. 0x2: If an IDLE request is detected, the module will switch to wake-up mode based on its internal activity, and the wake-up capability can be used if the bit [2] ENAWAKEUP is set. 0x3: Reserved - do not use.	RW	0x2
2	ENAWAKEUP	Wake-up feature control 0x0: Wake-up capability is disabled. 0x1: Wake-up capability is enabled.	RW	1
1	SOFTRESET	Software reset. During reads it always returns 0. 0x0: (write) Normal mode 0x1: (write) Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware.	RW	0
0	AUTOIDLE	Internal interface clock-gating strategy 0x0: Interface clock is free-running. 0x1: Automatic interface clock gating strategy is applied, based on the interface activity.	RW	1

Table 24-248. Register Call Summary for Register MCSPI_SYSCONFIG

Multichannel Serial Peripheral Interface

- Reset: [0] [1]
- Normal Mode: [2]
- Idle Mode: [3] [4] [5] [6] [7] [8]
- McSPI Global Initialization: [9] [10] [11]
- McSPI Register Summary: [12]
- McSPI Register Description: [13] [14]

Table 24-249. MCSPI_SYSSTATUS

Address Offset	0x114		
Physical Address	0x4809 8114 0x4809 A114 0x480B 8114 0x480B A114	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register provides status information about the module excluding the interrupt status information.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved for future module specific status information. Read returns 0.	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is ongoing Read 0x1: Reset completed	R	0

Table 24-250. Register Call Summary for Register MCSPI_SYSSTATUS

Multichannel Serial Peripheral Interface

- McSPI Global Initialization: [0]
- McSPI Register Summary: [1]

Table 24-251. MCSPI_IRQSTATUS

Address Offset	0x118		
Physical Address	0x4809 8118 0x4809 A118 0x480B 8118 0x480B A118	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	The interrupt status regroups all the status of the module internal events that can generate an interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EOW	WKS	RESERVED	RX3_FULL	TX3_UNDERFLOW	TX3_EMPTY	RESERVED	RX2_FULL	TX2_UNDERFLOW	TX2_EMPTY	RESERVED	RX1_FULL	TX1_UNDERFLOW	TX1_EMPTY	RX0_OVERFLOW	RX0_FULL	TX0_UNDERFLOW	TX0_EMPTY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads return 0	RW	0x0000
17	EOW	End of word count event when a channel is enabled using the FIFO buffer and the channel had sent the number of SPI word defined by MCSPI_XFERLEVEL[31:16] WCNT. Write 0x0: Event status bit unchanged Read 0x0: Event false Read 0x1: Event is pending Write 0x1: Event status bit is reset	RW W1toClr	0
16	WKS	Wake-up event in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CHxCONF[22:21] SPIENSLV Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reads returns 0	RW	0
14	RX3_FULL	Receiver register is full or almost full. Only when Channel 3 is enabled Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
13	TX3_UNDERFLOW	Transmitter register underflow. Only when Channel 3 is enabled. The transmitter register is empty (not updated by host or DMA with new data) before its time slot assignment. Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled. Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
12	TX3_EMPTY	Transmitter register is empty or almost empty. Note: Enabling the channel automatically rises this event. Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
11	RESERVED	Reads returns 0.	RW	0
10	RX2_FULL	Receiver register full or almost full. Channel 2 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
9	TX2_UNDERFLOW	Transmitter register underflow. Channel 2 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
8	TX2_EMPTY	Transmitter register empty or almost empty. Channel 2 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
7	RESERVED	Reads returns 0	RW	0
6	RX1_FULL	Receiver register full or almost full. Channel 1 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
5	TX1_UNDERFLOW	Transmitter register underflow. Channel 1 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
4	TX1_EMPTY	Transmitter register empty or almost empty. Channel 1 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
3	RX0_OVERFLOW	Receiver register overflow (slave mode only). Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
2	RX0_FULL	Receiver register full or almost full. Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
1	TX0_UNDERFLOW	Transmitter register underflow. Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
0	TX0_EMPTY	Transmitter register empty or almost empty. Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0

Table 24-252. Register Call Summary for Register MCSPI_IRQSTATUS

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0]
- Master Transmit-Only Mode (Half Duplex): [1]
- Master Receive-Only Mode (Half Duplex): [2] [3] [4] [5]
- Single-Channel Master Mode: [6] [7]
- Buffer Almost Full: [8] [9]
- Buffer Almost Empty: [10] [11]
- End of Transfer Management: [12]
- Interrupts: [13]
- Interrupt Events in Master Mode: [14] [15] [16] [17] [18] [19]
- Interrupt Events in Slave Mode: [20] [21] [22] [23] [24] [25] [26]
- Interrupt-Driven Operation: [27] [28]
- Polling: [29] [30]
- Idle Mode: [31] [32] [33]
- McSPI Operational Modes: [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72]
- Receive-Only Procedure – Interrupt Method: [73]
- McSPI Register Summary: [74]
- McSPI Register Description: [75] [76] [77]

Table 24-253. MCSPI_IRQENABLE

Address Offset	0x11C		
Physical Address	0x4809 811C 0x4809 A11C 0x480B 811C 0x480B A11C	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register allows enabling/disabling of the module internal sources of interrupt, on an event-by-event basis.		
Type	RW		

31:18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	EOW_ENABLE	WKE	RESERVED	RX3_FULL_ENABLE	TX3_UNDERFLOW_ENABLE	TX3_EMPTY_ENABLE	RESERVED	RX2_FULL_ENABLE	TX2_UNDERFLOW_ENABLE	TX2_EMPTY_ENABLE	RESERVED	RX1_FULL_ENABLE	TX1_UNDERFLOW_ENABLE	TX1_EMPTY_ENABLE	RX0_OVERFLOW_ENABLE	RX0_FULL_ENABLE	TX0_UNDERFLOW_ENABLE	TX0_EMPTY_ENABLE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads return 0.	RW	0x0000
17	EOW_ENABLE	End of Word count Interrupt Enable. 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
16	WKE	Wake-up event interrupt enable in slave mode when an active control signal is detected on the SPIEN line programmed in the MCSPI_CHxCONF[22:21] SPIENSLV bits 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reads returns 0.	RW	0
14	RX3_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
13	TX3_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
12	TX3_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
11	RESERVED	Reads return 0.	RW	0
10	RX2_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
9	TX2_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
8	TX2_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
7	RESERVED	Reads return 0.	RW	0
6	RX1_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
5	TX1_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
4	TX1_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
3	RX0_OVERFLOW_ENABLE	Receiver register Overflow Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
2	RX0_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
1	TX0_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
0	TX0_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0

Table 24-254. Register Call Summary for Register MCSPI_IRQENABLE

Multichannel Serial Peripheral Interface

- Interrupts: [0]
- Interrupt-Driven Operation: [1]
- Polling: [2]
- Idle Mode: [3] [4]
- McSPI Operational Modes: [5] [6] [7] [8]
- Receive-Only Procedure – Interrupt Method: [9] [10]
- McSPI Register Summary: [11]

Table 24-255. MCSPI_WAKEUPENABLE

Address Offset	0x120		
Physical Address	0x4809 8120 0x4809 A120 0x480B 8120 0x480B A120	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	The wake-up enable register allows enabling and disabling of the module internal sources of wakeup on event-by-event basis.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															WKEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	RW	0x0000 0000
0	WKEN	Wake-up functionality in slave mode when an active control signal is detected on the SPIEN line programmed in the MCSPI_CHxCONF[22:21] SPIENSLV bits 0x0: The event is not allowed to wake-up the system, even if the global control bit MCSPI_SYSCONFIG[2] ENAWAKEUP is set. 0x1: The event is allowed to wake-up the system if the global control bit MCSPI_SYSCONFIG[2] ENAWAKEUP is set.	RW	0

Table 24-256. Register Call Summary for Register MCSPI_WAKEUPENABLE

Multichannel Serial Peripheral Interface

- Idle Mode: [0] [1] [2]
- McSPI Register Summary: [3]

Table 24-257. MCSPI_SYST

Address Offset	0x124		
Physical Address	0x4809 8124 0x4809 A124 0x480B 8124 0x480B A124	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is used to check the correctness of the system interconnect either internally to peripheral bus, or externally to device I/O pads, when the module is configured in system test (SYSTEST) mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SSB	SPIENDIR	SPIDATDIR1	SPIDATDIR0	WAKD	SPICLK	SPIDAT_1	SPIDAT_0	SPIEN_3	SPIEN_2	SPIEN_1	SPIEN_0

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reads returns 0.	RW	0x00000
11	SSB	Set status bit 0x0: No action. Writing 0 does not clear already set status bits. This bit must be cleared before trying to clear a status bit of the MCSPI_IRQSTATUS register. 0x1: Force to 1 all status bits of MCSPI_IRQSTATUS register. Writing 1 into this bit sets to 1 all status bits in the MCSPI_IRQSTATUS register.	RW	0
10	SPIENDIR	Set the direction of the SPIEN[3:0] lines and SPICLK line. 0x0: Output (as in master mode) 0x1: Input (as in slave mode)	RW	0
9	SPIDATDIR1	Set the direction of the SPIDAT[1]. 0x0: Output 0x1: Input	RW	0
8	SPIDATDIR0	Set the direction of the SPIDAT[0]. 0x0: Output 0x1: Input	RW	0
7	WAKD	SWAKEUP output (signal data value of internal signal to system). The signal is driven high or low according to the value written into this bit. 0x0: The pin is driven low. 0x1: The pin is driven high.	RW	0
6	SPICLK	SPICLK line (signal data value) If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the CLKSPI line (high or low), and a write into this bit has no effect. If [10] SPIENDIR = 0 (output mode direction), the CLKSPI line is driven high or low according to the value written into this bit.	RW	0
5	SPIDAT_1	SPIDAT[1] line (signal data value) If [9] SPIDATDIR1 = 0 (output mode direction), the SPIDAT[1] line is driven high or low according to the value written into this bit. If [9] SPIDATDIR1 = 1 (input mode direction), this bit returns the value on the SPIDAT[1] line (high or low), and a write into this bit has no effect.	RW	0
4	SPIDAT_0	SPIDAT[0] line (signal data value) If [8] SPIDATDIR0 = 0 (output mode direction), the SPIDAT[0] line is driven high or low according to the value written into this bit. If [8] SPIDATDIR0 = 1 (input mode direction), this bit returns the value on the SPIDAT[0] line (high or low), and a write into this bit has no effect.	RW	0

Bits	Field Name	Description	Type	Reset
3	SPIEN_3	SPIEN[3] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[3] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[3] line (high or low), and a write into this bit has no effect.	RW	0
2	SPIEN_2	SPIEN[2] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[2] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[2] line (high or low), and a write into this bit has no effect.	RW	0
1	SPIEN_1	SPIEN[1] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[1] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[1] line (high or low), and a write into this bit has no effect.	RW	0
0	SPIEN_0	SPIEN[0] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[0] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[0] line (high or low), and a write into this bit has no effect.	RW	0

Table 24-258. Register Call Summary for Register MCSPI_SYST

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]

Table 24-259. MCSPI_MODULCTRL

Address Offset	0x128		
Physical Address	0x4809 8128 0x4809 A128 0x480B 8128 0x480B A128	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is dedicated to the configuration of the serial peripheral interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							FDAA	MOA	INITDLY			SYSTEM_TEST	MS	PIN34	SINGLE

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads returns 0.	RW	0x000000

Bits	Field Name	Description	Type	Reset
8	FDAA	FIFO DMA address 256-bit aligned This bit is used when a FIFO is managed by the module and DMA connected to the controller provides only 256-bit aligned address. If this bit is set the enabled channel which uses the FIFO has its data managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TXx and MCSPI_RXx registers. 0x0: FIFO data managed by MCSPI_TXx and MCSPI_RXx registers. 0x1: FIFO data managed by MCSPI_DAFTX and MCSPI_DAFRX registers.	RW	0
7	MOA	Multiple word interface access: this bit can only be used when a channel is enabled using a FIFO. It allows the system to perform multiple SPI word access for a single 32-bit interface word access. This is possible for WL < 16. 0x0: Multiple word access disabled 0x1: Multiple word access enabled with FIFO	RW	0
6:4	INITDLY	Initial SPI delay for first transfer: this field is an option only available in SINGLE master mode. The controller waits for a delay to transmit the first SPI word after channel enabled and corresponding TX register filled. This delay is based on SPI output frequency clock. No clock output provided to the boundary and chip select is not active in 4-pin mode within this period. 0x0: No delay for first spi transfer. 0x1: The controller wait 4 SPI bus clock 0x2: The controller wait 8 SPI bus clock 0x3: The controller wait 16 SPI bus clock 0x4: The controller wait 32 SPI bus clock	RW	0x0
3	SYSTEM_TEST	Enables the system test mode 0x0: Functional mode 0x1: System test mode (SYSTEST)	RW	0
2	MS	Master/slave 0x0: Master - The module generates the SPICLK and SPIEN[3:0]. 0x1: Slave - The module receives the SPICLK and SPIEN[3:0].	RW	1
1	PIN34	Pin mode selection: This bit is used in master or slave mode to configure the SPI pin mode (3-pin or 4-pin). If asserted the controller only uses SIMO, SOMI, and SPICLK clock pin for SPI transfers. 0x0: SPIEN is used as a chip-select. 0x1: SPIEN is not used. In this mode all related options to chip-select have no meaning.	RW	0
0	SINGLE	Single channel/Multi Channel (master mode only) 0x0: More than one channel will be used in master mode. 0x1: Only one channel will be used in master mode. This bit must be set in Force SPIEN[x] mode.	RW	0

Table 24-260. Register Call Summary for Register MCSPI_MODULCTRL

Multichannel Serial Peripheral Interface

- Single-Channel Master Mode: [0] [1] [2] [3] [4] [5]
- Chip-Select Timing Control: [6] [7]
- Slave Mode: [8] [9] [10] [11]
- 3-Pin or 4-Pin Mode: [12]
- McSPI Global Initialization: [13]
- McSPI Register Summary: [14]
- McSPI Register Description: [15] [16]

Table 24-261. MCSPI_CHxCONF

Address Offset	0x12C + (0x14 * x)	**Index**	x = 0 to 3
Physical Address	0x4809 812C + (0x14 * x) 0x4809 A12C + (0x14 * x) 0x480B 812C + (0x14 * x) 0x480B A12C + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is dedicated to the configuration of the channel x		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		CLKG	FFER	FFEW	TCS0		SBPOL	SBE	SPIENSLV		FORCE	TURBO	IS	DPE1	DPE0	DMAR	DMAW	TRM		WL					EPOL	CLKD				POL	PHA

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Read returns 0.	R	0x0
29	CLKG	Clock divider granularity this bit defines the granularity of channel clock divider: power of 2 or one clock cycle granularity. When this bit is set the register MCSPI_CHxCTRL[15:8] EXTCLK must be configured to reach a maximum of 4096 clock divider ratio. Then the clock divider ratio is a concatenation of [5:2] CLKD and MCSPI_CHxCTRL[15:8] EXTCLK values 0x0: Clock granularity of power of 2 0x1: One clock cycle granularity	RW	0
28	FFER	FIFO enabled for receive: Only one channel can have this bit field set. 0x0: The buffer is not used to receive data. 0x1: The buffer is used to receive data.	RW	0
27	FFEW	FIFO enabled for transmit: Only one channel can have this bit field set. 0x0: The buffer is not used to transmit data. 0x1: The buffer is used to transmit data.	RW	0
26:25	TCS0	Chip-select time control This 2-bit field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. 0x0: 0.5 clock cycle 0x1: 1.5 clock cycles 0x2: 2.5 clock cycles 0x3: 3.5 clock cycles	RW	0x0

Bits	Field Name	Description	Type	Reset
24	SBPOL	Start-bit polarity 0x0: Start-bit polarity is held to 0 during SPI transfer. 0x1: Start-bit polarity is held to 1 during SPI transfer.	RW	0
23	SBE	Start-bit enable for SPI transfer 0x0: Default SPI transfer length as specified by WL bit field 0x1: Start bit D/CX added before SPI transfer. Polarity is defined by bit [24] SBPOL	RW	0
22:21	SPIENSLV	Channel 0 only and slave mode only: SPI slave select signal detection. Reserved bits for other cases. 0x0: Detection enabled only on SPIEN[0] 0x1: Detection enabled only on SPIEN[1] 0x2: Detection enabled only on SPIEN[2] 0x3: Detection enabled only on SPIEN[3]	RW	0x0
20	FORCE	Manual SPIEN assertion to keep SPIEN active between SPI words (single channel master mode only). 0x0: Writing 0 into this bit drives low the SPIEN line when [6] EPOL=0, and drives it high when [6] EPOL=1. 0x1: Writing 1 into this bit drives high the SPIEN line when [6] EPOL=0, and drives it low when [6] EPOL=1.	RW	0
19	TURBO	Turbo mode 0x0: Turbo is deactivated (recommended for single SPI word transfer). 0x1: Turbo is activated to maximize the throughput for multiple SPI words transfer.	RW	0
18	IS	Input Select 0x0: Data line 0 (SPIDAT[0]) selected for reception 0x1: Data line 1 (SPIDAT[1]) selected for reception	RW	1
17	DPE1	Transmission enable for data line 1 0x0: Data line 1 (SPIDAT[1]) selected for transmission 0x1: No transmission on Data Line1 (SPIDAT[1])	RW	1
16	DPE0	Transmission Enable for data line 0 0x0: Data Line0 (SPIDAT[0]) selected for transmission 0x1: No transmission on data line 0 (SPIDAT[0])	RW	0
15	DMAR	DMA read request The DMA read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. The DMA read request line is deasserted on read completion of the receive register of the channel. 0x0: DMA read request disabled 0x1: DMA read request enabled	RW	0
14	DMAW	DMA write request. The DMA write request line is asserted when The channel is enabled and the transmitter register of the channel is empty. The DMA write request line is deasserted on load completion of the transmitter register of the channel. 0x0: DMA write request disabled 0x1: DMA write request enabled	RW	0

Bits	Field Name	Description	Type	Reset
13:12	TRM	Transmit/receive modes 0x0: Transmit-and-receive mode 0x1: Receive-only mode 0x2: Transmit-only mode 0x3: Reserved	RW	0x0
11:7	WL	SPI word length 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: The SPI word is 4 bits long 0x4: The SPI word is 5 bits long 0x5: The SPI word is 6 bits long 0x6: The SPI word is 7 bits long 0x7: The SPI word is 8 bits long 0x8: The SPI word is 9 bits long 0x9: The SPI word is 10 bits long 0xA: The SPI word is 11 bits long 0xB: The SPI word is 12 bits long 0xC: The SPI word is 13 bits long 0xD: The SPI word is 14 bits long 0xE: The SPI word is 15 bits long 0xF: The SPI word is 16 bits long 0x10: The SPI word is 17 bits long 0x11: The SPI word is 18 bits long 0x12: The SPI word is 19 bits long 0x13: The SPI word is 20 bits long 0x14: The SPI word is 21 bits long 0x15: The SPI word is 22 bits long 0x16: The SPI word is 23 bits long 0x17: The SPI word is 24 bits long 0x18: The SPI word is 25 bits long 0x19: The SPI word is 26 bits long 0x1A: The SPI word is 27 bits long 0x1B: The SPI word is 28 bits long 0x1C: The SPI word is 29 bits long 0x1D: The SPI word is 30 bits long 0x1E: The SPI word is 31 bits long 0x1F: The SPI word is 32 bits long	RW	0x00
6	EPOL	SPIEN polarity 0x0: SPIEN is held high during the ACTIVE state. 0x1: SPIEN is held low during the ACTIVE state.	RW	0

Bits	Field Name	Description	Type	Reset
5:2	CLKD	Frequency divider for SPICLK (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (FCLK) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default, the clock divider ratio has a power of 2 granularity when [29] CLKG is cleared. Otherwise, this field is the 4-LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHxCTRL[15:8] EXTCLK register. The value description below defines the clock ratio when [29] CLKG is set to 0. 0x0: 1 0x1: 2 0x2: 4 0x3: 8 0x4: 16 0x5: 32 0x6: 64 0x7: 128 0x8: 256 0x9: 512 0xA: 1024 0xB: 2048 0xC: 4096 0xD: 8192 0xE: 16384 0xF: 32768	RW	0x0
1	POL	SPICLK polarity (see Section 24.4.2.3.1, *Transfer Format*) 0x0: SPICLK is held low during the INACTIVE state 0x1: SPICLK is held high during the INACTIVE state	RW	0
0	PHA	SPICLK phase (see Section 24.4.2.3.1, *Transfer Format*) 0x0: Data are latched on odd-numbered edges of SPICLK. 0x1: Data are latched on even-numbered edges of SPICLK.	RW	0

Table 24-262. Register Call Summary for Register MCSPI_CHxCONF

Multichannel Serial Peripheral Interface

- Basic McSPI Pins for Master Mode: [0] [1] [2] [3]
- Basic McSPI Pins for Slave Mode: [4] [5] [6] [7]
- Multichannel SPI Protocol and Data Format: [8] [9] [10] [11] [12] [13] [14] [15]
- Transfer Format: [16]
- SPI in Slave Mode: [17]
- Master Transmit-and-Receive Mode (Full Duplex): [18]
- Master Transmit-Only Mode (Half Duplex): [19]
- Master Receive-Only Mode (Half Duplex): [20]
- Single-Channel Master Mode: [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32]
- Start-Bit Mode: [33] [34] [35] [36]
- Chip-Select Timing Control: [37]
- Programmable SPI Clock: [38] [39] [40] [41] [42] [43]
- Dedicated Resources: [44] [45] [46] [47] [48] [49]
- Slave Transmit-and-Receive Mode: [50]
- Slave Transmit-Only Mode: [51]
- Slave Receive-Only Mode: [52]
- 3-Pin or 4-Pin Mode: [53]
- FIFO Buffer Management: [54] [55] [56] [57]
- Buffer Almost Full: [58] [59]
- Buffer Almost Empty: [60] [61]
- Interrupt Events in Master Mode: [62] [63]
- Interrupt Events in Slave Mode: [64] [65]
- DMA Requests: [66] [67]
- Idle Mode: [68]
- McSPI Operational Modes: [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85]
- McSPI Register Summary: [86]
- McSPI Register Description: [87] [88] [89] [90] [91] [92] [93]

Table 24-263. MCSPI_CHxSTAT

Address Offset	0x130 + (0x14 * x)	**Index**	x = 0 to 3
Physical Address	0x4809 8130 + (0x14 * x) 0x4809 A130 + (0x14 * x) 0x480B 8130 + (0x14 * x) 0x480B A130 + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register provides status information about transmitter and receiver registers of channel x.		
Type	R		

31–7	6	5	4	3	2	1	0
RESERVED	RXFFF	RXFFE	TXFFF	TXFFE	EOT	TXS	RXS

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Read returns 0.	R	0x0000000
6	RXFFF	Channel x FIFO receive buffer full status Read 0x0: FIFO receive buffer is not full Read 0x1: FIFO receive buffer is full	R	0
5	RXFFE	Channel x FIFO receive buffer empty status Read 0x0: FIFO receive buffer is not empty Read 0x1: FIFO receive buffer is empty	R	0

Bits	Field Name	Description	Type	Reset
4	TXFFF	Channel x FIFO transmit buffer full status Read 0x0: FIFO transmit buffer is not full Read 0x1: FIFO transmit buffer is full	R	0
3	TXFFE	Channel x FIFO transmit buffer empty status Read 0x0: FIFO transmit buffer is not empty Read 0x1: FIFO transmit buffer is empty	R	0
2	EOT	Channel x end of transfer status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (transmit/receive modes, turbo mode). See dedicated chapters for details. Read 0x0: This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer). Read 0x1: This flag is automatically set to one at the end of an SPI transfer.	R	0
1	TXS	Channel x transmitter register status Read 0x0: Register is full. Read 0x1: Register is empty.	R	0
0	RXS	Channel x receiver register status Read 0x0: Register is empty. Read 0x1: Register is full.	R	0

Table 24-264. Register Call Summary for Register MCSPI_CHxSTAT

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0] [1] [2]
- Master Transmit-Only Mode (Half Duplex): [3]
- Master Receive-Only Mode (Half Duplex): [4]
- Single-Channel Master Mode: [5] [6] [7] [8] [9]
- Dedicated Resources: [10] [11]
- Slave Transmit-and-Receive Mode: [12]
- Slave Transmit-Only Mode: [13]
- Slave Receive-Only Mode: [14]
- End of Transfer Management: [15]
- McSPI Operational Modes: [16] [17] [18] [19]
- Receive-Only Procedure – Polling Method: [20]
- Transmit-Only Procedure – Polling Method: [21]
- Transmit-and-Receive Procedure – Polling Method: [22]
- McSPI Register Summary: [23]

Table 24-265. MCSPI_CHxCTRL

Address Offset	0x134 + (0x14 * x)	**Index**	x = 0 to 3
Physical Address	0x4809 8134 + (0x14 * x) 0x4809 A134 + (0x14 * x) 0x480B 8134 + (0x14 * x) 0x480B A134 + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is dedicated to enable channel x.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EXTCLK								RESERVED							EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Read returns 0.	RW	0x0000
15:8	EXTCLK	Clock ratio extension: this field is used to concatenate with MCSPI_CHxCONF[5:2] CLKD register for clock ratio only when granularity is one clock cycle (MCSPI_CHxCONF[29] CLKG set to 1). Then the maximum value reached is 4096 clock divider ratio. 0x0: Clock ratio is CLKD + 1. 0x1: Clock ratio is CLKD + 1 + 16. ... 0xFF: Clock ratio is CLKD + 1 + 4080.	RW	0x00
7:1	RESERVED	Read returns 0.	RW	0x00
0	EN	Channel enable 0x0: Channel x is not active. 0x1: Channel x is active.	RW	0

Table 24-266. Register Call Summary for Register MCSPI_CHxCTRL

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0]
- Single-Channel Master Mode: [1] [2]
- Programmable SPI Clock: [3]
- Dedicated Resources: [4] [5]
- McSPI Operational Modes: [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25]
- Receive-Only Procedure – Polling Method: [26] [27]
- Receive-Only Procedure – Interrupt Method: [28] [29]
- Transmit-Only Procedure – Polling Method: [30] [31]
- Transmit-and-Receive Procedure – Polling Method: [32] [33]
- McSPI Register Summary: [34]
- McSPI Register Description: [35] [36] [37]

Table 24-267. MCSPI_TXx

Address Offset	0x138 + (0x14 * x)	**Index**	x = 0 to 3
Physical Address	0x4809 8138 + (0x14 * x) 0x4809 A138 + (0x14 * x) 0x480B 8138 + (0x14 * x) 0x480B A138 + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains a single SPI word for channel x to transmit on the serial link, whatever SPI word length is.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDATA																															

Bits	Field Name	Description	Type	Reset
31:0	TDATA	Channel x data to transmit	RW	0x0000 0000

Table 24-268. Register Call Summary for Register MCSPI_TXx

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0] [1] [2] [3]
- Master Transmit-Only Mode (Half Duplex): [4]
- Master Receive-Only Mode (Half Duplex): [5] [6] [7] [8] [9]
- Single-Channel Master Mode: [10] [11]
- Dedicated Resources: [12] [13] [14] [15] [16]
- Slave Transmit-and-Receive Mode: [17] [18]
- Slave Receive-Only Mode: [19] [20] [21]
- Interrupt Events in Master Mode: [22] [23] [24] [25] [26] [27] [28]
- Interrupt Events in Slave Mode: [29] [30] [31] [32] [33] [34]
- Interrupt-Driven Operation: [35]
- Polling: [36]
- DMA Requests: [37] [38]
- McSPI Operational Modes: [39] [40] [41] [42] [43]
- Transmit-Only Procedure – Polling Method: [44]
- Transmit-and-Receive Procedure – Polling Method: [45] [46]
- McSPI Register Summary: [47]
- McSPI Register Description: [48] [49] [50]

Table 24-269. MCSPI_RXx

Address Offset	0x13C + (0x14 * x)	**Index**	x = 0 to 3
Physical Address	0x4809 813C + (0x14 * x) 0x4809 A13C + (0x14 * x) 0x480B 813C + (0x14 * x) 0x480B A13C + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains a single SPI word for channel x received through the serial link, whatever SPI word length is.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA																															

Bits	Field Name	Description	Type	Reset
31:0	RDATA	Channel x received data	R	0x0000 0000

Table 24-270. Register Call Summary for Register MCSPI_RXx

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0] [1]
- Master Transmit-Only Mode (Half Duplex): [2] [3] [4]
- Master Receive-Only Mode (Half Duplex): [5] [6]
- Single-Channel Master Mode: [7] [8] [9] [10]
- Dedicated Resources: [11] [12] [13]
- Slave Transmit-and-Receive Mode: [14]
- Slave Transmit-Only Mode: [15] [16]
- End of Transfer Management: [17]
- Interrupt Events in Master Mode: [18] [19] [20] [21]
- Interrupt Events in Slave Mode: [22] [23] [24] [25] [26] [27]
- Interrupt-Driven Operation: [28]
- Polling: [29]
- DMA Requests: [30]
- Idle Mode: [31]
- McSPI Operational Modes: [32] [33] [34] [35] [36] [37] [38]
- Receive-Only Procedure – Polling Method: [39]
- Receive-Only Procedure – Interrupt Method: [40]
- McSPI Register Summary: [41]
- McSPI Register Description: [42] [43] [44]

Table 24-271. MCSPI_XFERLEVEL

Address Offset	0x17C		
Physical Address	0x4809 817C 0x4809 A17C 0x480B 817C 0x480B A17C	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register provides transfer levels needed while using FIFO buffer during transfer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WCNT																AFL								AEL							

Bits	Field Name	Description	Type	Reset
31:16	WCNT	SPI word counter. This field holds the programmable value of number of SPI word to be transferred on channel which is using the FIFO buffer. When transfer had started, a read back in this field returns the current SPI word transfer index. 0x0: Counter not used 0x1: One word ... 0xFFFE: 65534 SPI words 0xFFFF: 65535 SPI words	RW	0x0000

Bits	Field Name	Description	Type	Reset
15:8	AFL	Buffer almost full This field holds the programmable almost-full level value used to determine almost full buffer condition. If the user wants an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least n bytes, then the buffer AFL must be set with n-1. 0x0: 1 byte 0x1: 2 bytes ... 0xFE: 255 bytes 0xFF: 256 bytes	RW	0x00
7:0	AEL	Buffer almost empty. this field holds the programmable almost-empty level value used to determine almost empty buffer condition. If the user wants an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive n bytes, then the buffer AEL must be set with n-1. 0x0: 1 byte 0x1: 2 bytes ... 0xFE: 255 bytes 0xFF: 256 bytes	RW	0x00

Table 24-272. Register Call Summary for Register MCSPI_XFERLEVEL

Multichannel Serial Peripheral Interface

- FIFO Buffer Management: [0] [1]
- Buffer Almost Full: [2]
- Buffer Almost Empty: [3]
- End of Transfer Management: [4] [5]
- Interrupt Events in Master Mode: [6] [7] [8] [9] [10]
- Interrupt Events in Slave Mode: [11] [12] [13] [14]
- McSPI Operational Modes: [15] [16]
- McSPI Register Summary: [17]
- McSPI Register Description: [18]

Table 24-273. MCSPI_DAFTX

Address Offset	0x0000 0180		
Physical Address	0x4809 8180 0x4809 A180 0x480B 8180 0x480B A180	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains the SPI words to be transmitted on the SPI bus when FIFO is used and DMA address is aligned on 256 bit. This register is an image of one of the MCSPI_TXx registers corresponding to the channel which has its FIFO enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAFTDATA																															

Bits	Field Name	Description	Type	Reset
31:0	DAFTDATA	FIFO data to transmit with DMA 256 bit aligned address. This field is only used when MCSPI_MODULCTRL[8] FDAA is set to 0x1 and only one of the enabled channels has the MCSPI_CHxCONF[27] FFEW bit set to 0x1. If these conditions are not met any access to this field returns a null value.	RW	0x00000000

Table 24-274. Register Call Summary for Register MCSPI_DAFTX

Multichannel Serial Peripheral Interface
- McSPI Register Summary: [0]
- McSPI Register Description: [1] [2]

Table 24-275. MCSPI_DAFRX

Address Offset	0x0000 01A0		
Physical Address	0x4809 81A0 0x4809 A1A0 0x480B 81A0 0x480B A1A0	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains the SPI words received from the SPI bus when FIFO is used and DMA address is aligned on 256 bit. This register is an image of one of the MCSPI_RXx registers corresponding to the channel which has its FIFO enabled.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAFRDATA																															

Bits	Field Name	Description	Type	Reset
31:0	DAFRDATA	FIFO data received with DMA 256 bit aligned address. This field is only used when MCSPI_MODULCTRL[8] FDAA is set to 0x1 and only one of the enabled channels has the MCSPI_CHxCONF[28] FFER bit set to 0x1. If these conditions are not met any access to this field returns a null value.	R	0x00000000

Table 24-276. Register Call Summary for Register MCSPI_DAFRX

Multichannel Serial Peripheral Interface
- McSPI Register Summary: [0]
- McSPI Register Description: [1] [2]

24.5 Quad Serial Peripheral Interface

24.5.1 Quad Serial Peripheral Interface Overview

The quad serial peripheral interface (QSPI) module is a kind of SPI module that allows single, dual, or quad read access to external SPI devices. This module has a memory mapped register interface, which provides a direct interface for accessing data from external SPI devices and thus simplifying software requirements. The QSPI works as a master only.

The one QSPI in the device is primarily intended for fast booting from quad-SPI flash memories. Figure 24-101 shows the QSPI module overview.

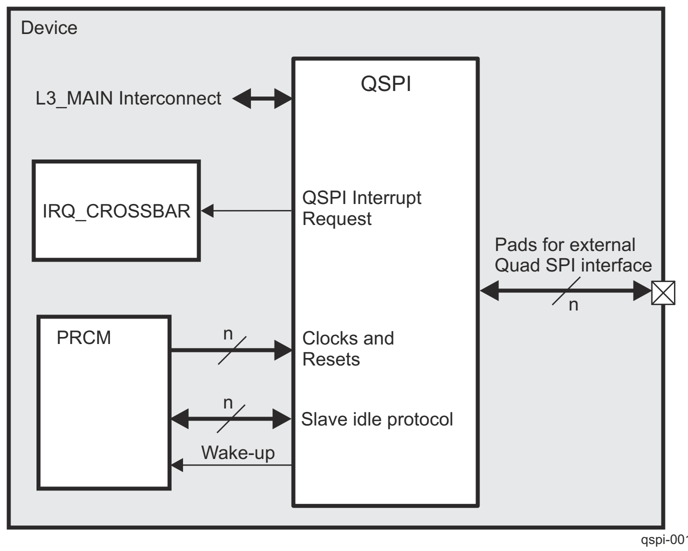


Figure 24-101. QSPI Overview

The QSPI supports the following features:

- General SPI features:
  - Programmable clock divider
  - Six pin interface
  - Programmable length (from 1 to 128 bits) of the words transferred
  - Programmable number (from 1 to 4096) of the words transferred
  - 4 external chip-select signals
  - Support for 3-, 4-, or 6-pin SPI interface
  - Optional interrupt generation on word or frame (number of words) completion
  - Programmable delay between chip select activation and output data from 0 to 3 QSPI clock cycles
  - Programmable signal polarities
  - Programmable active clock edge
  - Software-controllable interface allowing for any type of SPI transfer
  - Control through L3_MAIN configuration port
- Serial flash interface (SFI) features:
  - Serial flash read/write interface
  - Additional registers for defining read and write commands to the external serial flash device
  - 1 to 4 address bytes
  - Fast read support, where fast read requires dummy bytes after address bytes; 0 to 3 dummy bytes can be configured.
  - Dual read support
  - Quad read support
  - Little-endian support (only for memory mapped registers used to configure QSPI controller and not SPI content accesses)

– Linear increment addressing mode only

The QSPI supports only dual and quad reads. Dual or quad writes are not supported. In addition, there is no "pass through" mode supported where the data present on the QSPI input is sent to its output.

Note

The QSPI module does not support cache line wrap mode.

24.5.2 QSPI Environment

Figure 24-102 shows a typical connection of the QSPI module to the external quad-SPI flash memory.

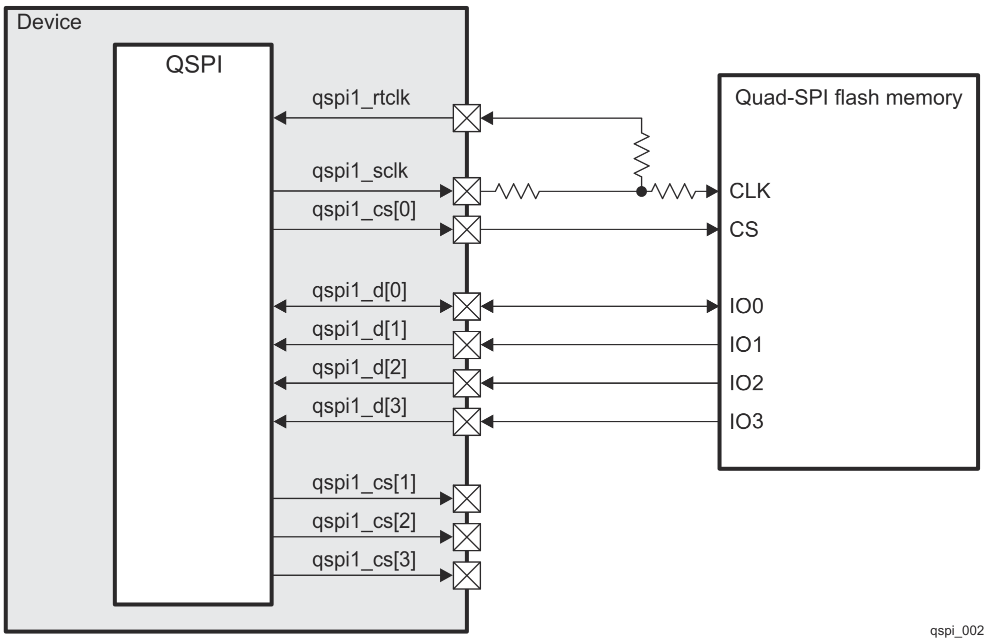


Figure 24-102. QSPI Connected to an External Quad-SPI Flash Memory

Table 24-277 lists and describes the QSPI I/O signals.

Table 24-277. QSPI I/O Signals

QSPI Signal/Pad name	I/O[1]	Description					
		3-pin[2] SPI Read (Single Read)	3-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Single Read)	4-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Dual Read)	6-pin[2] SPI Read (Quad Read)
qspi1_d[0]	IO	Used as SPI data input	Used as SPI data output	Not used	Used as SPI data output	Used as SPI data input 0	Used as SPI data input 0
qspi1_d[1]	I	Not used	Not used	Used as SPI data input	Not used	Used as SPI data input 1	Used as SPI data input 1
qspi1_d[2]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 2
qspi1_d[3]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 3
qspi1_sclk	O	Clock for the external SPI device					
qspi1_cs[0]	O	External SPI device chip-select 0					
qspi1_cs[1]	O	External SPI device chip-select 1					
qspi1_cs[2]	O	External SPI device chip-select 2					
qspi1_cs[3]	O	External SPI device chip-select 3					

Table 24-277. QSPI I/O Signals (continued)

QSPI Signal/Pad name	I/O(1)	Description					
		3-pin(2) SPI Read (Single Read)	3-pin(2) SPI Write (Single Write)	4-pin(2) SPI Read (Single Read)	4-pin(2) SPI Write (Single Write)	4-pin(2) SPI Read (Dual Read)	6-pin(2) SPI Read (Quad Read)
qspi1_rtclk	I	The qspi1_sclk output must be connected to the qspi1_rtclk input, and is used for controlling the timing of the read return data when the QSPI module operates in Mode 0. In case Mode 3 is used, there is no need to connect the qspi1_sclk to the qspi1_rtclk.					

(1) I = Input; O = Output
(2) This is the pin count at the SPI flash memory side. The pin count at the device side is increased by one because of the qspi1_rtclk signal. References to the pin count throughout this chapter consider the pin count at the SPI flash memory side.

Note

In order to ensure proper timing, precise layout and routing requirements must be followed. For layout and routing requirements for all QSPI signals, see section “PCB Guidelines” of the device Data Manual.

24.5.3 QSPI Integration

Figure 24-103 shows the integration of the QSPI module in the device.

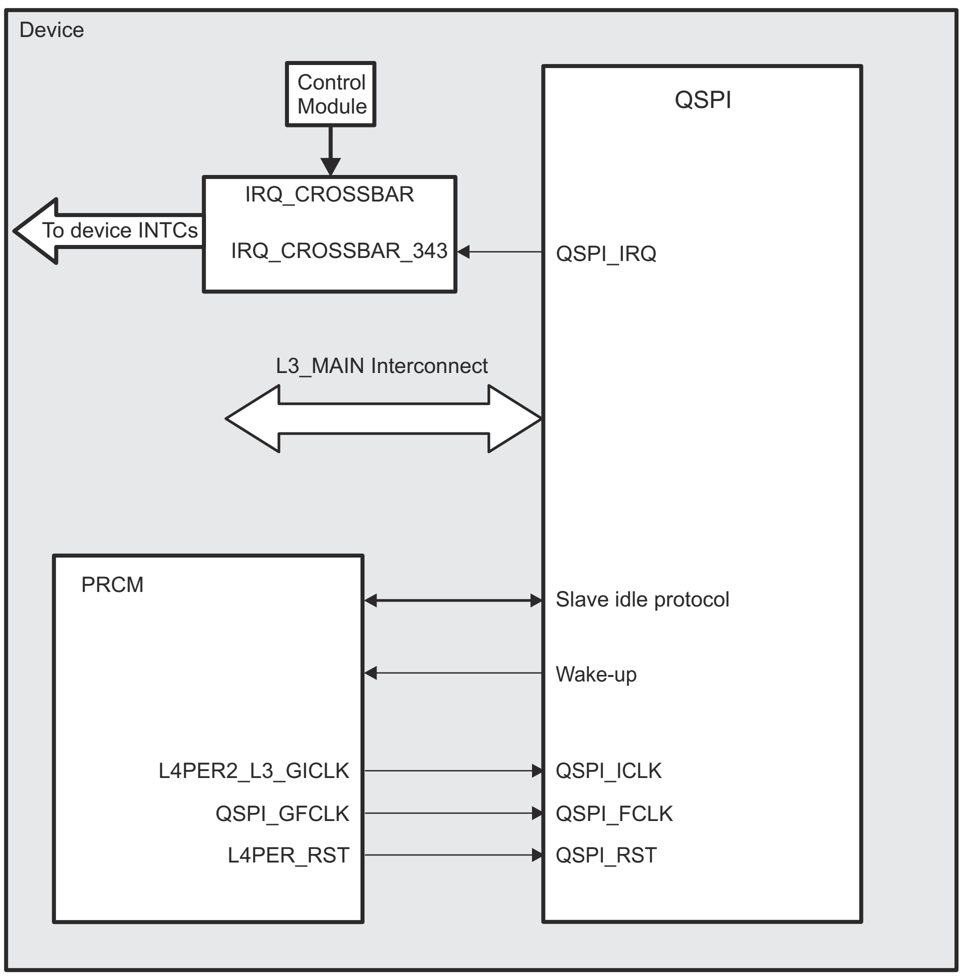


Figure 24-103. QSPI Integration

Table 24-278 through Table 24-280 summarize the integration of the QSPI in the device.

Table 24-278. QSPI Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
QSPI	PD_COREAON	Yes	L3_MAIN

Table 24-279. QSPI Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
QSPI	QSPI_ICLK	L4PER2_L3_GICLK	PRCM	Interface clock for the QSPI
	QSPI_FCLK	QSPI_GFCLK	PRCM	Functional clock for the QSPI
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
QSPI	QSPI_RST	L4PER_RST	PRCM	Asynchronous reset signal for the QSPI

Table 24-280. QSPI Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
QSPI	QSPI_IRQ	IRQ_CROSSBAR_343	–	QSPI interrupt request

Note

The Default Mapping column in Table 24-280 shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device interrupt controller (INTC) through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device INTCs, see *Interrupt Controllers*.

Note

For the description of the interrupt source, see Section 24.5.4.3, *QSPI Interrupt Requests*.

24.5.4 QSPI Functional Description

24.5.4.1 QSPI Block Diagram

Iinitial device boot from external SPI flash memory can be accomplished through the QSPI module. The interface is a simple 4-wire SPI used for control or data transfers. The QSPI also supports a 3-wire SPI protocol where the qspi1_d[0] signal is used as a bidirectional for reads and writes. In addition, a 6-wire mode can be used to support quad read devices. Figure 24-104 shows the QSPI block diagram.

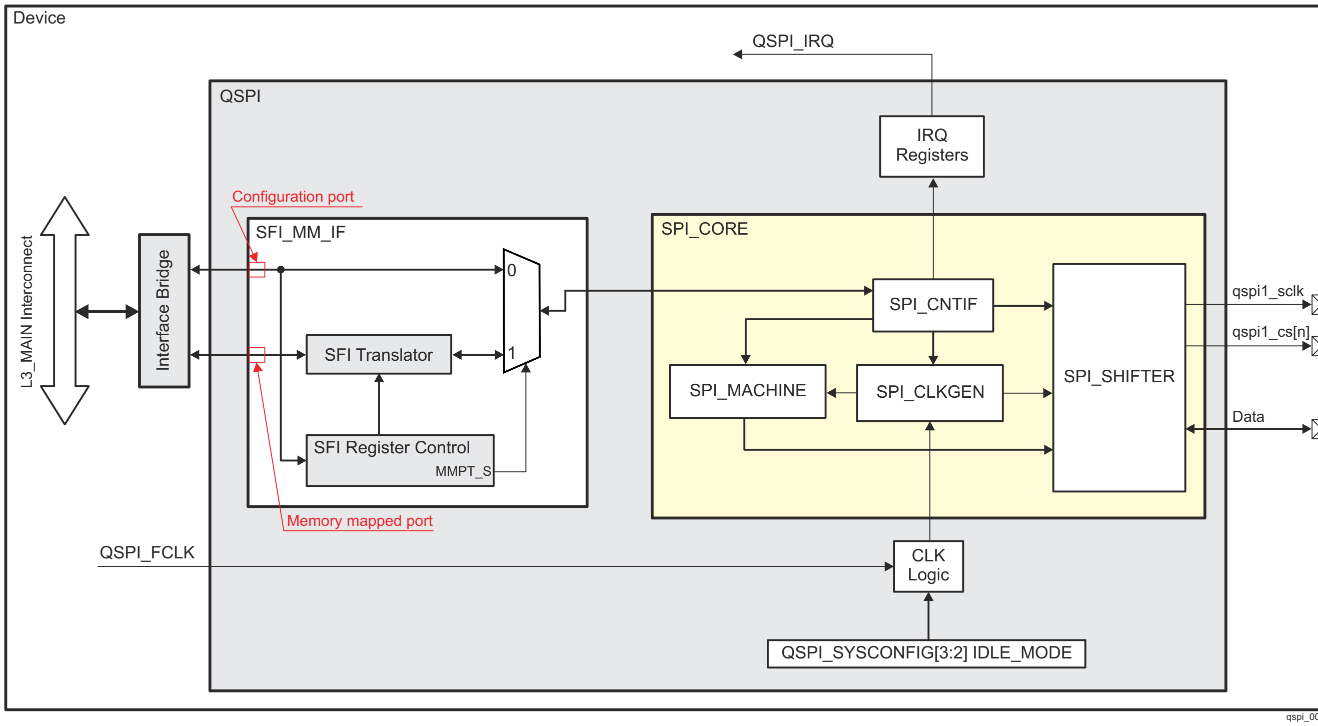


Figure 24-104. QSPI Block Diagram

The QSPI is composed of two blocks. The first one is the SFI memory-mapped interface (SFI_MM_IF) and the second one is the SPI core (SPI_CORE). The SFI_MM_IF block is associated only with SPI flash memories and is used for specifying typical for the SPI flash memories settings (read or write command, number of address and dummy bytes, and so on) unlike the SPI_CORE block, which is associated with the SPI interface itself and is used to configure typical SPI settings (chip-select polarity, serial clock inactive state, SPI clock mode, length of the words transferred, and so on).

The SFI_MM_IF comprises the following two subblocks:

- SFI register control
- SFI translator

The SPI_CORE comprises the following four subblocks:

- SPI control interface (SPI_CNTIF)
- SPI clock generator (SPI_CLKGEN)
- SPI control state machine (SPI_MACHINE)
- SPI data shifter (SPI_SHIFTER)

In addition, an interface bridge connects the two ports (configuration port and memory-mapped port) of the SFI_MM_IF block to the L3_MAIN interconnect. There are no software controls associated with this interface bridge.

The QSPI supports long transfers through a frame-style sequence. In its generic SPI use mode, a word can be defined up to 128 bits and multiple words can be transferred during a single access. For each word, a device initiator must read or write the new data and then tell the QSPI to continue the current operation. Using this sequence, a maximum of 4096 128-bit words can be transferred in a single SPI read or write operation. This allows great flexibility when connecting the QSPI to various types of devices.

As opposed to the generic SPI use mode, the communication with serial flash-type devices requires sending a byte command, followed by sending bytes of data. Commands can be sent through the SPI_CORE block to communicate with a serial flash device; however, it is easier to do this using the SFI_MM_IF block because it is intended to ease the communication with serial flash devices. If the SPI_CORE is used to communicate with a serial flash device, software must load the command into the SPI data transfer register with additional configuration fields, perform the byte transfer, then place the data to be sent (or configure for receive) along with additional configuration fields, and perform that transfer. Reads and writes to serial flash devices are more specific. First, the read or write command byte is sent, followed by 1 to 4 bytes of address (corresponding to the address to read/write), then followed by the data write/receive phase. Data is always sent byte oriented. When the address is loaded, data can be continuously read or written, and the address will automatically increment to each byte address internally to the serial flash device.

Note

The SFI_MM_IF block only allows reading and writing to an externaly connected SPI flash device. The SFI_MM_IF block does not allow reads or writes to internal configuration and status registers of the SPI flash device. These registers must be accessed through the features of the SPI_CORE block.

24.5.4.1.1 SFI Register Control

The SFI register control block consists of the following five configuration registers:

- QSPI_SPI_SETUP0_REG
- QSPI_SPI_SETUP1_REG
- QSPI_SPI_SETUP2_REG
- QSPI_SPI_SETUP3_REG
- QSPI_SPI_SWITCH_REG

The first four registers let the user define the following:

- Byte command for a serial flash read specified by the QSPI_SPI_SETUPi_REG[7:0] RCMD bit field
- Byte command for a serial flash write specified by the QSPI_SPI_SETUPi_REG[23:16] WCMD bit field
- Number of address bytes required for the particular type of serial flash specified by the QSPI_SPI_SETUPi_REG[9:8] NUM_A_BYTES bit field
- Number of “dummy bytes” that may be needed to support the fast read mode function of some serial flash devices. The QSPI_SPI_SETUPi_REG[11:10] NUM_D_BYTES bit field specifies the number of "dummy bits." In addition, the QSPI_SPI_SETUPi_REG[28:24] NUM_D_BITS bit field can also specify the number of "dummy bits."
- Whether the read command is single (normal), dual, or quad read mode command. This is specified by the QSPI_SPI_SETUPi_REG[13:12] READ_TYPE bit field. (i is equal to 0, 1, 2 and 3 and means that the QSPI_SPI_SETUPi_REG registers are associated with each of the four supported chip-selects [that is, four supported output SPI flash devices])

The QSPI_SPI_SWITCH_REG register acts as a static switch which allows the configuration port (shown in Figure 24-104) to connect directly to the SPI_CORE block, or allows the memory-mapped port (also shown in Figure 24-104) to connect to the SPI_CORE block. This is done using the QSPI_SPI_SWITCH_REG[0] MMPT_S bit.

In addition, the QSPI_SPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

24.5.4.1.2 SFI Translator

The SFI translator block represents an FSM which, based on the configuration information loaded into the SFI register control block, converts each input read/write sequence into an SPI_CORE configuration sequence for access to the external serial flash memory.

A read sequence is converted into the following actions:

1. SPI chip-select goes active.
2. Read command byte is issued.
3. 1 to 4 address bytes, which correspond to the first address supplied, are issued.
4. 0 to 3 dummy bytes are issued, if “fast read” is supported.
5. Data bytes are read from the external SPI flash memory.

6. SPI chip-select goes inactive.

For linear addressing mode, action 5 is repeated until the byte count to be transferred reaches zero.

A write sequence is identical to a read sequence, except that a write sequence does not use dummy bytes.

Another important aspect with regard to writes is that a serial flash memory location can only be written to if the bits are erased in advance. Erased means the bits are set to 1. This means that writing only changes 1 contents to 0. It is not possible with this write to change the contents of a bit from 0 to 1. An erase command must be performed to do this operation. Erase commands cannot be executed on single byte locations. Depending on device types, there are page, block, and chip erase commands. To perform an erase command, the particular command must be sent over the SPI bus, and an internal register of the serial flash device must then be polled to determine when the erase completes. The erases must be done through the configuration port by software before performing any writes through the memory-mapped port. This means that writes are passed through to the serial flash device, but if the memory locations being modified are not properly erased before the write, the contents may not result in what was sent.

24.5.4.1.3 SPI Control Interface

The SPI control interface contains configuration registers used to configure the SPI core functionality of the QSPI. This block maintains all configuration settings for the SPI core (that is, settings specific for the SPI interface itself but not for the SPI flash memories).

The registers defined for this block are:
- The QSPI_PID register, which is read only and contains QSPI revision associated information
- The QSPI_SPI_CLOCK_CNTRL_REG register, which is used to control external SPI clock (qspi1_sclk)
- The QSPI_SPI_DC_REG register used to define the SPI clock mode and chip-select polarity for the four external SPI devices
- The QSPI_SPI_CMD_REG register used to control the operation of the SPI command. This register is also used to configure and transfer data.
- Four data registers used for reading the data received and for writing the data to be transferred. These registers are:
  – QSPI_SPI_DATA_REG
  – QSPI_SPI_DATA_REG_1
  – QSPI_SPI_DATA_REG_2
  – QSPI_SPI_DATA_REG_3

  These four registers compose a 128-bit shift register.
- The QSPI_SPI_STATUS_REG register, which contains status information

All of these registers can only be written if the QSPI is not busy. This means that they can be written if the QSPI_SPI_STATUS_REG[0] BUSY bit is 0x0. The QSPI becomes busy when a write to the QSPI_SPI_CMD_REG[18:16] CMD bit field is performed. Writing to this bit field starts an SPI transaction and sets the QSPI_SPI_STATUS_REG[0] BUSY bit to 0x1. The CMD bit field can be written again when the BUSY bit is 0x0. In addition, the start of the SPI transaction is synchronized to the qspi1_sclk clock and clearing of the BUSY bit is synchronized to the QSPI_FCLK clock.

The register group QSPI_SPI_DATA_REG_3, QSPI_SPI_DATA_REG_2, QSPI_SPI_DATA_REG_1 and QSPI_SPI_DATA_REG is treated as a single 128-bit word for shifting data in and out. The QSPI_SPI_DATA_REG_3 register is used for the most significant bits and the QSPI_SPI_DATA_REG is used for the least significant bits.This applies for both reads and writes. For example, after reading a 128-bit word (WLEN = 0x7F) the most significant bit of the data read, that is bit 127, will be located at QSPI_SPI_DATA_REG_3[31] position and the least significant bit, that is bit 0 of the data read, will be located at the QSPI_SPI_DATA_REG[0] position.

The data written to this register group should be right justified so that a data pre-shifting is not required. The QSPI_SPI_CMD_REG[25:19] WLEN bit field determines the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the WLEN bit field should be set to 0x7 and the data byte should be written to the lower byte of the QSPI_SPI_DATA_REG register. By setting the word length to 0x7 the QSPI_SPI_DATA_REG register will look like a pseudo 8-bit shift register. When the user wants to write 40-bit long word the WLEN bit field should be set to 0x27, the 32 least significant bits of data should be written to the QSPI_SPI_DATA_REG and the rest 8 most significant bits of data should be written to the lower byte of the QSPI_SPI_DATA_REG_1 register. By setting WLEN to 0x27 these two registers will look like a pseudo 40-bit shift register. When the word length is greater than 64 bits the QSPI_SPI_DATA_REG_2 register is also used and the previously described logic applies. The QSPI_SPI_DATA_REG_3 register is used together with the other three data registers when the word length is greater than 96 bits.

When dual or quad read mode is used the number of the words transferred must be even. This number is configured through the QSPI_SPI_CMD_REG[11:0] FLEN bit field.

Note

The QSPI module does not support a "pass through" mode where the data present on qspi1_d[1] is sent to qspi1_d[0], when 4-pin non-dual read mode is used. This means that setting the QSPI_SPI_CMD_REG[18:16] CMD bit field to 0x1 causes the QSPI only to read from an external device using the qspi1_d[1] pad as an input and if a write to the same external device is desired, the CMD bit field should be set to 0x2, which causes the qspi1_d[0] pad to be used as an output.

24.5.4.1.4 SPI Clock Generator

The SPI clock generator uses the QSPI_FCLK clock as an input, and generates the qspi1_sclk, which is a divided version of the QSPI_FCLK clock. The divide ratio is a 16-bit value configured through the QSPI_SPI_CLOCK_CNTRL_REG[15:0] DCLK_DIV bit field and thus provides a division factor in a range from 1 to 65536. The QSPI_FCLK clock is divided by the DCLK_DIV value + 1 to provide the qspi1_sclk clock. When DCLK_DIV = 0x0 the QSPI_FCLK clock equals the DCLK clock. The value in the DCLK_DIV bit field applies only when the QSPI_SPI_CLOCK_CNTRL_REG[31] CLKEN bit is set to 0x1. Figure 24-105 shows the SPI_CLKGEN block.

If the CLKEN bit is 0x0 the command specified in the QSPI_SPI_CMD_REG[18:16] CMD bit field is not executed and the QSPI_SPI_STATUS_REG[0] BUSY bit is not set. The command is executed only if the CLKEN bit is 0x1 before write to the CMD bit field.

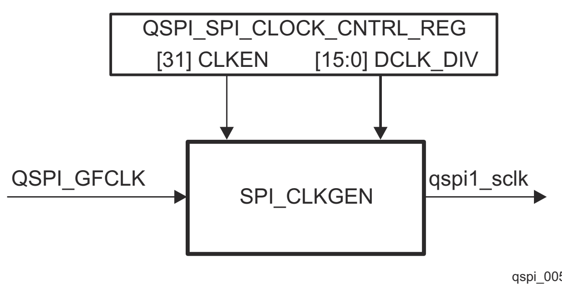


Figure 24-105. SPI_CLKGEN Block

24.5.4.1.5 SPI Control State-Machine

The SPI control state-machine (SPI_MACHINE) manages the operation of the SPI_CORE block. SPI_MACHINE takes control and configuration information from the registers in the SPI_CNTIF block as input and provides control information to the SPI data shifter. This information is used to control the SPI data port. The SPI_MACHINE also generates status information, which is sent back to the SPI_CNTIF block.

Writing a valid value to the QSPI_SPI_CMD_REG[18:16] CMD bit field sets immediately the QSPI_SPI_STATUS_REG[0] BUSY bit to 0x1, activates the corresponding qspi1_cs[n] (n = 0 to 3) and starts the SPI data transaction. The BUSY bit is cleared automatically when QSPI_SPI_CMD_REG[25:19] WLEN number of bits are shifted in or out. If the value of the QSPI_SPI_STATUS_REG[27:16] WDCNT bit field is different than 0x0 and WLEN number of bits are shifted already, the SPI_MACHINE waits until another write to the CMD bit field is performed. If the command written to the CMD bit field is valid, then this increments the value of the WDCNT bit field from 0x0 and starts shifting data in or out again. This is repeated until the WDCNT bit field reaches the frame length (QSPI_SPI_CMD_REG[11:0] FLEN), that is, all words of the frame are shifted or till earlier frame termination occurs. While the SPI_MACHINE is waiting for write to the CMD bit field the corresponding qspi1_cs[n] (n = 0 to 3) remains active and the BUSY flag is set to 0x0. In addition, the bit length for each word can be changed during a frame from 1 to 128 bits using the QSPI_SPI_CMD_REG[25:19] WLEN bit field.

The SPI_MACHINE also provides a mechanism to terminate the frame earlier. This is done by writing an invalid command to the CMD bit field. An invalid command corresponds to the 0x0 and 0x4 (reserved) values of the CMD bit field. Writing one of these values when the the WDCNT bit field is not equal to 0x0 and when the BUSY flag is 0x0 terminates the frame earlier.

The corresponding qspi1_cs[n] (n = 0 to 3) becomes inactive when all words are shifted or when the frame terminates earlier.

24.5.4.1.6 SPI Data Shifter

The SPI data shifter handles the capture and generation of the SPI interface signals. Based on control signals from the SPI_MACHINE and SPI_CNTIF blocks, data is shifted in or out on falling or rising edge of qspi1_sclk clock depending on the SPI clock mode selected. Table 24-281 lists the four defined clock modes of operation for the QSPI.

Table 24-281. SPI Clock Modes Definition

Mode	Settings in the QSPI_SPI_DC_REG Register		Description
	Value of the CKP bits	Value of the CKPH bits	
0	0	0	Data input captured on falling edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock
1	0	1	Data input captured on rising edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock
2	1	0	Data input captured on rising edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock
3	1	1	Data input captured on falling edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock

Note

Mode 1 and Mode 2 are not supported and should not be used.

The CKPi and CKPHi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register control the clock modes. Each of these 4 bits corresponds to an output chip select.

Figure 24-106 shows all four clock modes. In addition, through the DDi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register the data can be delayed from one to three qspi1_sclk clock cycles after the corresponding qspi1_cs[n] (n = 0 to 3) goes active. The active state of each chip-select can also be controlled through the CSPi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register.

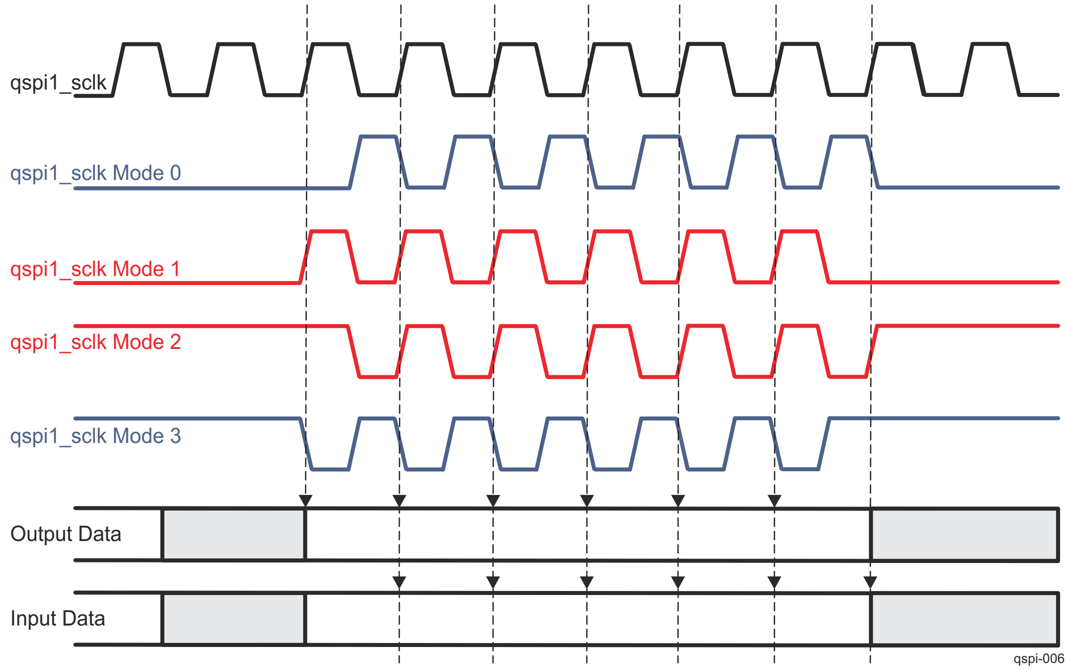


Figure 24-106. SPI Clock Modes

24.5.4.2 QSPI Clock Configuration

The QSPI complies with the PRCM slave-idle protocol. The QSPI_FCLK clock is gated based on the values loaded in the QSPI_SYSCONFIG[3:2] IDLE_MODE bit field. Three modes are supported:

- Force-idle: The QSPI_FCLK clock is gated unconditionally by the QSPI.
- No-idle: The QSPI_FCLK clock is never gated by the QSPI.
- Smart-idle: The QSPI_FCLK clock is gated by the QSPI, depending on its internal requirements.

24.5.4.3 QSPI Interrupt Requests

The QSPI generates one interrupt request which is connected to the IRQ_CROSSBAR module. This interrupt request, QSPI_IRQ, is connected to the IRQ_CROSSBAR_343 input. The QSPI_IRQ interrupt line can be activated by one of the interrupt events listed in Table 24-282.

Figure 24-107 shows a logical representation of the QSPI interrupt generation scheme.

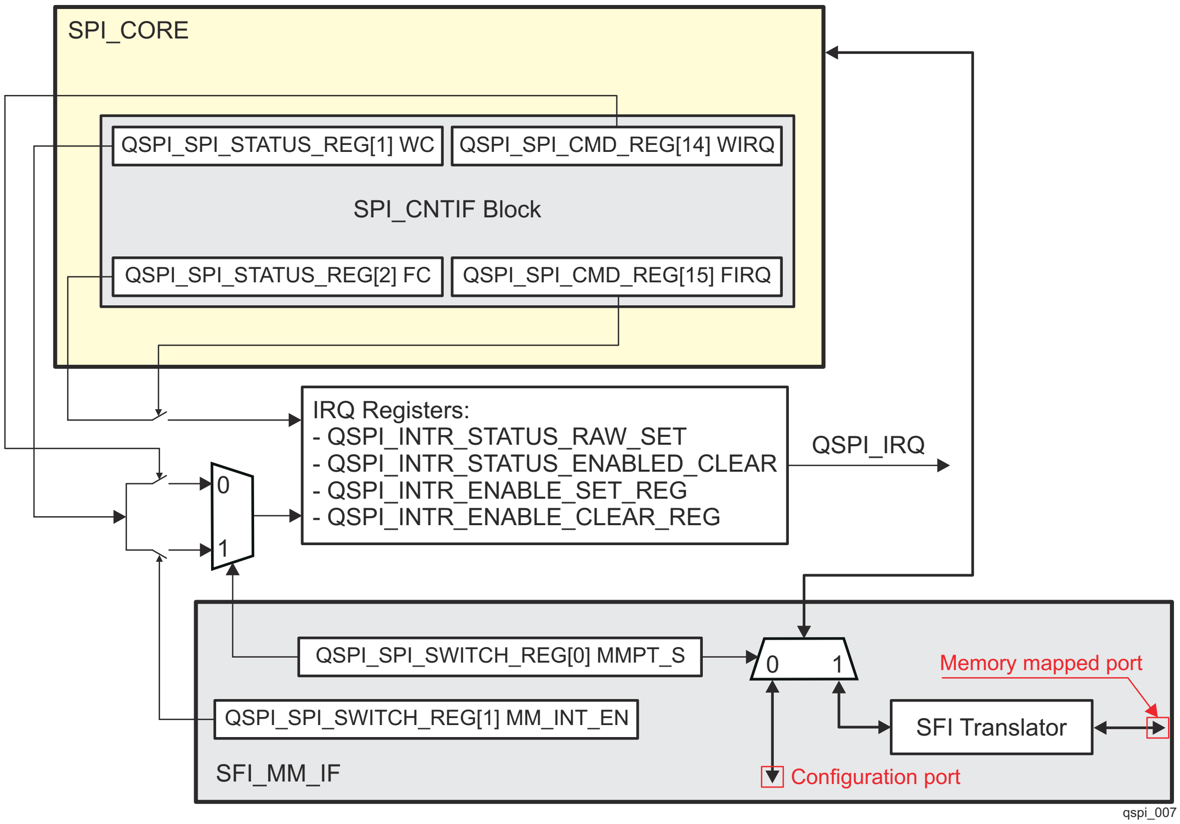


Figure 24-107. Logical Representation of the QSPI Interrupt Generation Scheme

QSPI_SPI_STATUS_REG[1] WC and QSPI_SPI_STATUS_REG[2] FC are status bits indicating whether word or frame transfer is complete. Setting the corresponding interrupt enable bit (WIRQ or FIRQ) in the QSPI_SPI_CMD_REG register allows these events (WC and FC) to generate an interrupt. The WC and FC bits are reset every time the user writes to the QSPI_SPI_CMD_REG register or reads the QSPI_SPI_STATUS_REG register. This is done to keep control parameters from changing the interface protocol signals while a transfer is in progress. Additionally, the QSPI_SPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

When the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_SPI_CMD_REG[15] FIRQ bits are set to 0x1 the following applies:

- The QSPI activates its interrupt line only if the interrupts are enabled by setting to 0x1 the corresponding bits in the QSPI_INTR_ENABLE_SET_REG register. These interrupts can be disabled by setting the corresponding bits in the QSPI_INTR_ENABLE_CLEAR_REG register to 0x1.
- After an interrupt has been serviced, software must clear the corresponding status flag. This is done by setting the corresponding bit in the QSPI_INTR_STATUS_ENABLED_CLEAR register to 0x1, which also clears the corresponding bit in the QSPI_INTR_STATUS_RAW_SET register. The status flags in the QSPI_INTR_STATUS_RAW_SET register are set even if the corresponding interrupt is disabled unlike those in the QSPI_INTR_STATUS_ENABLED_CLEAR register, which are set only if the corresponding interrupt is enabled.
- The QSPI also generates an interrupt if a certain bit in the QSPI_INTR_STATUS_RAW_SET register is set to 0x1 and the corresponding interrupt is enabled through the QSPI_INTR_ENABLE_SET_REG register. This feature is useful during user software debugging. In addition, even if interrupts are not enabled a corresponding raw flag in the QSPI_INTR_STATUS_RAW_SET register is set to 0x1 when an IRQ condition occurs.

- Even if interrupts are not enabled, a certain status bit in the QSPI_INTR_STATUS_RAW_SET register can also be cleared by setting to 0x1 the corresponding bit in the QSPI_INTR_STATUS_ENABLED_CLEAR register.

It must be considered that the previously described scenario applies if the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_SPI_CMD_REG[15] FIRQ bits are set to 0x1.

Note

The QSPI_IRQ interrupt line is activated only if at least one of the following conditions is met:

- The word complete interrupt is enabled:
  - during operations using the memory-mapped port by setting to 0x1 both the QSPI_SPI_SWITCH_REG[1] MM_INT_EN and QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET bits.
  - during operations using the configuration port by setting to 0x1 both the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET bits.
- The frame complete interrupt is enabled setting to 0x1 both the QSPI_SPI_CMD_REG[15] FIRQ and QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET bits.

The QSPI_IRQ interrupt line is also activated when both the conditions are met.

Table 24-282 lists the event flags and the corresponding mask bits of the sources which can cause interrupts.

Table 24-282. QSPI Events

Event Flag	Event Mask	Description
QSPI_INTR_STATUS_RAW_SET[1] WIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[1] WIRQ_ENA QSPI_SPI_STATUS_REG[1] WC	QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[1] WIRQ_ENA_CLR QSPI_SPI_CMD_REG[14] WIRQ	Word complete interrupt event. Asserted each time after a word is transferred or received.
QSPI_INTR_STATUS_RAW_SET[0] FIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[0] FIRQ_ENA QSPI_SPI_STATUS_REG[2] FC	QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[0] FIRQ_ENA_CLR QSPI_SPI_CMD_REG[15] FIRQ	Frame complete interrupt event. Asserted each time after a frame is transferred or received.

24.5.4.4 QSPI Memory Regions

Two memory regions are associated with the QSPI. The first memory region is dedicated to the configuration port. Using this memory region, all internal registers can be programmed and serial transfers made from the four supported external SPI devices. The L3_MAIN start address at which the configuration port is available is 0x4B30 0000. The second memory region is associated mainly with the memory-mapped port and is used for communication directly with one of the four supported external SPI devices. This memory region starts at 0x5C00 0000 and ends at 0x5FFF FFFF L3_MAIN address.

The CTRL_CORE_CONTROL_IO_2[10:8] QSPI_MEMMAPPED_CS bit field provides a functionality for remapping the previously described address space which starts at 0x5C00 0000 L3_MAIN address to one of the four supported chip selects or to the configuration registers. The CTRL_CORE_CONTROL_IO_2 register resides in the CTRL_MODULE_CORE.

It is important to keep in mind that the configuration port provides an access to all the QSPI registers listed in Table 24-284. These are configuration registers and also four data registers. The configuration registers are used to configure typical SPI and serial flash memory settings and the four data registers are used for read and write operations. When communicating with an external SPI device (but not an SPI flash memory) the SPI_CORE module should be used and the data exchanged is available thorough these four data registers, which can be accessed only through the configuration port. When a communication with an external SPI flash memory is desired, the memory-mapped port should be used.

In other words, to read from an external SPI flash memory, first configure the QSPI through the configuration port and then perform a read through the memory-mapped port.

24.5.5 QSPI Register Manual

24.5.5.1 QSPI Instance Summary

Table 24-283. QSPI Instance Summary

Module Name	Module Base Address	Size
QSPI	0x4B30 0000	1MiB

24.5.5.2 QSPI registers

24.5.5.2.1 QSPI Register Summary

Table 24-284. QSPI Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	QSPI Base Address
QSPI_PID	R	32	0x0000 0000	0x4B30 0000
QSPI_SYSCONFIG	RW	32	0x0000 0010	0x4B30 0010
QSPI_INTR_STATUS_RAW_SET	RW	32	0x0000 0020	0x4B30 0020
QSPI_INTR_STATUS_ENABLED_CLEAR	RW	32	0x0000 0024	0x4B30 0024
QSPI_INTR_ENABLE_SET_REG	RW	32	0x0000 0028	0x4B30 0028
QSPI_INTR_ENABLE_CLEAR_REG	RW	32	0x0000 002C	0x4B30 002C
QSPI_INTC_EOI_REG	RW	32	0x0000 0030	0x4B30 0030
QSPI_SPI_CLOCK_CNTRL_REG	RW	32	0x0000 0040	0x4B30 0040
QSPI_SPI_DC_REG	RW	32	0x0000 0044	0x4B30 0044
QSPI_SPI_CMD_REG	RW	32	0x0000 0048	0x4B30 0048
QSPI_SPI_STATUS_REG	R	32	0x0000 004C	0x4B30 004C
QSPI_SPI_DATA_REG	RW	32	0x0000 0050	0x4B30 0050
QSPI_SPI_SETUP0_REG	RW	32	0x0000 0054	0x4B30 0054
QSPI_SPI_SETUP1_REG	RW	32	0x0000 0058	0x4B30 0058
QSPI_SPI_SETUP2_REG	RW	32	0x0000 005C	0x4B30 005C
QSPI_SPI_SETUP3_REG	RW	32	0x0000 0060	0x4B30 0060
QSPI_SPI_SWITCH_REG	RW	32	0x0000 0064	0x4B30 0064
QSPI_SPI_DATA_REG_1	RW	32	0x0000 0068	0x4B30 0068
QSPI_SPI_DATA_REG_2	RW	32	0x0000 006C	0x4B30 006C
QSPI_SPI_DATA_REG_3	RW	32	0x0000 0070	0x4B30 0070

24.5.5.2.2 QSPI Register Description

Table 24-285. QSPI_PID

Address Offset	0x0000 0000		
Physical Address	0x4B30 0000	**Instance**	QSPI
Description	Revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI Internal data

Table 24-286. Register Call Summary for Register QSPI_PID

Quad Serial Peripheral Interface

- SPI Control Interface: [0]
- QSPI Register Summary: [1]

Table 24-287. QSPI_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4B30 0010	**Instance**	QSPI
Description			
Type	RW		

31:4	3:2	1:0
RESERVED	IDLE_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x2
3:2	IDLE_MODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode 0x3: Reserved.	RW	0x2
1:0	RESERVED		R	0x0

Table 24-288. Register Call Summary for Register QSPI_SYSCONFIG

Quad Serial Peripheral Interface

- QSPI Clock Configuration: [0]
- QSPI Register Summary: [1]

Table 24-289. QSPI_INTR_STATUS_RAW_SET

Address Offset	0x0000 0020		
Physical Address	0x4B30 0020	**Instance**	QSPI
Description	This register contains raw interrupt status flags.		
Type	RW		

31:2	1	0
RESERVED	WIRQ_RAW	FIRQ_RAW

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		RW	0x0
1	WIRQ_RAW	Word Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0

Bits	Field Name	Description	Type	Reset
0	FIRQ_RAW	Frame Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0

Table 24-290. Register Call Summary for Register QSPI_INTR_STATUS_RAW_SET

Quad Serial Peripheral Interface

- QSPI Interrupt Requests: [0] [1] [2] [3] [4] [5] [6]
- QSPI Register Summary: [7]

Table 24-291. QSPI_INTR_STATUS_ENABLED_CLEAR

Address Offset	0x0000 0024		
Physical Address	0x4B30 0024	**Instance**	QSPI
Description	This register contains status flags of the enabled interrupts.		
Type	RW		

31–2	1	0
RESERVED	WIRQ_ENA	FIRQ_ENA

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA	Word Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the word interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0
0	FIRQ_ENA	Frame Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the frame interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0

Table 24-292. Register Call Summary for Register QSPI_INTR_STATUS_ENABLED_CLEAR

Quad Serial Peripheral Interface

- QSPI Interrupt Requests: [0] [1] [2] [3] [4]
- QSPI Register Summary: [5]

Table 24-293. QSPI_INTR_ENABLE_SET_REG

Address Offset	0x0000 0028		
Physical Address	0x4B30 0028	**Instance**	QSPI
Description	This register enables the interrupts.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														WIRQ_ENA_SET	FIRQ_ENA_SET

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_SET	Word interrupt enable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt enabled Write: 0x0: Has no effect 0x1: Enables the word interrupt	RW	0x0
0	FIRQ_ENA_SET	Frame interrupt enable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Enables the frame interrupt	RW	0x0

Table 24-294. Register Call Summary for Register QSPI_INTR_ENABLE_SET_REG

Quad Serial Peripheral Interface

- QSPI Interrupt Requests: [0] [1] [2] [3] [4] [5] [6]
- QSPI Register Summary: [7]

Table 24-295. QSPI_INTR_ENABLE_CLEAR_REG

Address Offset	0x0000 002C		
Physical Address	0x4B30 002C	**Instance**	QSPI
Description	This register disables the interrupts.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	WIRQ_ENA_CLR	FIRQ_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_CLR	Word interrupt disable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the word interrupt	RW	0x0
0	FIRQ_ENA_CLR	Frame interrupt disable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the frame interrupt	RW	0x0

Table 24-296. Register Call Summary for Register QSPI_INTR_ENABLE_CLEAR_REG

Quad Serial Peripheral Interface

- QSPI Interrupt Requests: [0] [1] [2]
- QSPI Register Summary: [3]

Table 24-297. QSPI_INTC_EOI_REG

Address Offset	0x0000 0030		
Physical Address	0x4B30 0030	**Instance**	QSPI
Description	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if a new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOI_VECTOR																															

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the interrupt outputs. There is one interrupt output. Write 0x0 after servicing the interrupt to be able to generate another interrupt if pulse interrupts are used. Any other write value is ignored.	RW	0x0

Table 24-298. Register Call Summary for Register QSPI_INTC_EOI_REG

Quad Serial Peripheral Interface

- QSPI Register Summary: [0]

Table 24-299. QSPI_SPI_CLOCK_CNTRL_REG

Address Offset	0x0000 0040		
Physical Address	0x4B30 0040	**Instance**	QSPI
Description	This register controls the external SPI clock generation. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
	RW		

31	30:16	15:0
CLKEN	RESERVED	DCLK_DIV

Bits	Field Name	Description	Type	Reset
31	CLKEN	External SPI clock (qspi1_sclk) enable. 0x0: The qspi1_sclk clock is turned off 0x1: The qspi1_sclk clock is enabled	RW	0x0
30:16	RESERVED		R	0x0
15:0	DCLK_DIV	Divide ratio for the external SPI clock (qspi1_sclk)	RW	0x0

Table 24-300. Register Call Summary for Register QSPI_SPI_CLOCK_CNTRL_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0]
- SPI Clock Generator: [1] [2]
- QSPI Register Summary: [3]

Table 24-301. QSPI_SPI_DC_REG

Address Offset	0x0000 0044		
Physical Address	0x4B30 0044	**Instance**	QSPI
Description	This register controls the different modes for each output chip select. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31:29	28:27	26	25	24	23:21	20:19	18	17	16	15:13	12:11	10	9	8	7:5	4:3	2	1	0
RESERVED	DD3	CKPH3	CSP3	CKP3	RESERVED	DD2	CKPH2	CSP2	CKP2	RESERVED	DD1	CKPH1	CSP1	CKP1	RESERVED	DD0	CKPH0	CSP0	CKP0

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:27	DD3	Data delay for chip select 3 0x0: Data is output on the same cycle as the qspi1_cs[3] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[3] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[3] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[3] goes active	RW	0x0

Bits	Field Name	Description	Type	Reset
26	CKPH3	Clock phase for chip select 3. If CKP3 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP3 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
25	CSP3	Chip select polarity for chip select 3. 0x0: Active low 0x1: Active high	RW	0x0
24	CKP3	Clock polarity for chip select 3. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0
23:21	RESERVED		R	0x0
20:19	DD2	Data delay for chip select 2 0x0: Data is output on the same cycle as the qspi1_cs[2] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[2] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[2] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[2] goes active	RW	0x0
18	CKPH2	Clock phase for chip select 2. If CKP2 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP2 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
17	CSP2	Chip select polarity for chip select 2. 0x0: Active low 0x1: Active high	RW	0x0
16	CKP2	Clock polarity for chip select 2. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0
15:13	RESERVED		R	0x0
12:11	DD1	Data delay for chip select 1 0x0: Data is output on the same cycle as the qspi1_cs[1] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[1] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[1] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[1] goes active	RW	0x0

Bits	Field Name	Description	Type	Reset
10	CKPH1	Clock phase for chip select 1. If CKP1 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP1 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
9	CSP1	Chip select polarity for chip select 1. 0x0: Active low 0x1: Active high	RW	0x0
8	CKP1	Clock polarity for chip select 1. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0
7:5	RESERVED		R	0x0
4:3	DD0	Data delay for chip select 0 0x0: Data is output on the same cycle as the qspi1_cs[0] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[0] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[0] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[0] goes active	RW	0x0
2	CKPH0	Clock phase for chip select 0. If CKP0 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP0 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
1	CSP0	Chip select polarity for chip select 0. 0x0: Active low 0x1: Active high	RW	0x0
0	CKP0	Clock polarity for chip select 0. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0

Table 24-302. Register Call Summary for Register QSPI_SPI_DC_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0]
- SPI Data Shifter: [1] [2] [3] [4]
- QSPI Register Summary: [5]

Table 24-303. QSPI_SPI_CMD_REG

Address Offset	0x0000 0048		
Physical Address	0x4B30 0048	**Instance**	QSPI

Table 24-303. QSPI_SPI_CMD_REG (continued)

Description	This register sets up the SPI command. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		CSNUM		RESERVED		WLEN							CMD			FIRQ	WIRQ	RESERVED		FLEN											

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	CSNUM	Device select. Sets the active chip select for the current transfer. 0x0: Chip Select 0 active 0x1: Chip Select 1 active 0x2: Chip Select 2 active 0x3: Chip Select 3 active	RW	0x0
27:26	RESERVED		R	0x0
25:19	WLEN	Word length. Sets the size of the individual transfers from 1 to 128 bits. When a word length greater than 32 bits is configured, not only the QSPI_SPI_DATA_REG register, but also the QSPI_SPI_DATA_REG_1, QSPI_SPI_DATA_REG_2, QSPI_SPI_DATA_REG_3 are used. One or all of these registers are used depending on the length of words transferred. 0x0: 1 bit 0x1: 2 bits ... 0x7F: 128 bits	RW	0x0
18:16	CMD	Transfer command. 0x0: Reserved 0x1: 4-pin Read Single 0x2: 4-pin Write Single 0x3: 4-pin Read Dual 0x4: Reserved 0x5: 3-pin Read Single 0x6: 3-pin Write Single 0x7: 6-pin Read Quad	RW	0x0
15	FIRQ	Frame complete interrupt enable. 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0
14	WIRQ	Word complete interrupt enable 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0
13:12	RESERVED		R	0x0
11:0	FLEN	Frame Length. 0x0: 1 word 0x1: 2 words ... 0xFFF: 4096 words	RW	0x0

Table 24-304. Register Call Summary for Register QSPI_SPI_CMD_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0] [1] [2] [3] [4]
- SPI Clock Generator: [5]
- SPI Control State-Machine: [6] [7] [8] [9]
- QSPI Interrupt Requests: [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]
- QSPI Register Summary: [20]

Table 24-305. QSPI_SPI_STATUS_REG

Address Offset	0x0000 004C		
Physical Address	0x4B30 004C	**Instance**	QSPI
Description	This register contains indicators to allow the user to monitor the progression of a frame transfer. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	R		

31:28	27:16	15:3	2	1	0
RESERVED	WDCNT	RESERVED	FC	WC	BUSY

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WDCNT	Word count. This field will reflect the 1-4096 words transferred	R	0x0
15:3	RESERVED		R	0x0
2	FC	Frame complete. This bit is set after the transmision of all the requested words completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Transfer is not complete 0x1: Transfer is complete	R	0x0
1	WC	Word complete. This bit is set after each word transfer completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Word transfer is not complete 0x1: Word transfer is complete	R	0x0
0	BUSY	Busy bit. Active transfer in progress. This bit is only set during an active word transfer. Between words it is cleared. 0x0: Idle 0x1: Busy	R	0x0

Table 24-306. Register Call Summary for Register QSPI_SPI_STATUS_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0] [1] [2]
- SPI Clock Generator: [3]
- SPI Control State-Machine: [4] [5]
- QSPI Interrupt Requests: [6] [7] [8] [9] [10]
- QSPI Register Summary: [11]
- QSPI Register Description: [12] [13] [14] [15] [16] [17] [18] [19] [20] [21]

Table 24-307. QSPI_SPI_DATA_REG

Address Offset	0x0000 0050		
Physical Address	0x4B30 0050	**Instance**	QSPI

Table 24-307. QSPI_SPI_DATA_REG (continued)

Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the first 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 24-308. Register Call Summary for Register QSPI_SPI_DATA_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0] [1] [2] [3] [4] [5] [6]
- QSPI Register Summary: [7]
- QSPI Register Description: [8]

Table 24-309. QSPI_SPI_SETUP0_REG

Address Offset	0x0000 0054		
Physical Address	0x4B30 0054	**Instance**	QSPI
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 0 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0

Bits	Field Name	Description	Type	Reset
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 24-310. Register Call Summary for Register QSPI_SPI_SETUP0_REG

Quad Serial Peripheral Interface

- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 24-311. QSPI_SPI_SETUP1_REG

Address Offset	0x0000 0058		
Physical Address	0x4B30 0058	**Instance**	QSPI
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 1 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.		
Type	RW		

31:29	28:24	23:16	15:14	13:12	11:10	9:8	7:0
RESERVED	NUM_D_BITS	WCMD	RESERVED	READ_TYPE	NUM_D_BYTES	NUM_A_BYTES	RCMD

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 24-312. Register Call Summary for Register QSPI_SPI_SETUP1_REG

Quad Serial Peripheral Interface

- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 24-313. QSPI_SPI_SETUP2_REG

Address Offset	0x0000 005C		
Physical Address	0x4B30 005C	**Instance**	QSPI
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 2 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 24-314. Register Call Summary for Register QSPI_SPI_SETUP2_REG

Quad Serial Peripheral Interface

- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 24-315. QSPI_SPI_SETUP3_REG

Address Offset	0x0000 0060		
Physical Address	0x4B30 0060	**Instance**	QSPI

Table 24-315. QSPI_SPI_SETUP3_REG (continued)

Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 3 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
Type	RW

31:29	28:24	23:16	15:14	13:12	11:10	9:8	7:0
RESERVED	NUM_D_BITS	WCMD	RESERVED	READ_TYPE	NUM_D_BYTES	NUM_A_BYTES	RCMD

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 24-316. Register Call Summary for Register QSPI_SPI_SETUP3_REG

Quad Serial Peripheral Interface

- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 24-317. QSPI_SPI_SWITCH_REG

Address Offset	0x0000 0064		
Physical Address	0x4B30 0064	**Instance**	QSPI
Description	This register allows initiators to switch control of the SPI core port between the configuration port and the SFI translator. In addition, an interrupt enable field is defined which is used to enable or disable word complete interrupt generation in memory mapped mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MM_INT_EN	MMPT_S

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	MM_INT_EN	Memory mapped mode interrupt enable. 0x0: Word complete interrupt is disabled during memory mapped operations 0x1: Word complete interrupt is enabled for memory mapped operations	RW	0x0
0	MMPT_S	MPT select. 0x0: Configuration port is selected to control the SPI_CORE. 0x1: SFI translator is selected to control the SPI_CORE.	RW	0x0

Table 24-318. Register Call Summary for Register QSPI_SPI_SWITCH_REG

Quad Serial Peripheral Interface

- SFI Register Control: [0] [1] [2] [3]
- QSPI Interrupt Requests: [4] [5]
- QSPI Register Summary: [6]

Table 24-319. QSPI_SPI_DATA_REG_1

Address Offset	0x0000 0068		
Physical Address	0x4B30 0068	**Instance**	QSPI
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the second 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 24-320. Register Call Summary for Register QSPI_SPI_DATA_REG_1

Quad Serial Peripheral Interface

- SPI Control Interface: [0] [1] [2]
- QSPI Register Summary: [3]
- QSPI Register Description: [4]

Table 24-321. QSPI_SPI_DATA_REG_2

Address Offset	0x0000 006C		
Physical Address	0x4B30 006C	**Instance**	QSPI
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the third 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DATA

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 24-322. Register Call Summary for Register QSPI_SPI_DATA_REG_2

Quad Serial Peripheral Interface

- SPI Control Interface: [0] [1] [2]
- QSPI Register Summary: [3]
- QSPI Register Description: [4]

Table 24-323. QSPI_SPI_DATA_REG_3

Address Offset	0x0000 0070		
Physical Address	0x4B30 0070	**Instance**	QSPI
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the fourth 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 24-324. Register Call Summary for Register QSPI_SPI_DATA_REG_3

Quad Serial Peripheral Interface

- SPI Control Interface: [0] [1] [2] [3] [4]
- QSPI Register Summary: [5]
- QSPI Register Description: [6]

24.6 Multichannel Audio Serial Port

This section describes the multichannel audio serial port (McASP).

24.6.1 McASP Overview

This section introduces the multichannel audio serial port (McASP) module and describes its main functions and connections in the device.

The McASP functions as a general-purpose audio serial port optimized to the requirements of various audio applications. The McASP module can operate in both transmit and receive modes. The McASP is useful for time-division multiplexed (TDM) stream, Inter-IC Sound (I2S) protocols reception and transmission as well as for an intercomponent digital audio interface transmission (DIT). The McASP has the flexibility to gluelessly connect to a Sony/Philips digital interface (S/PDIF) transmit physical layer component.

Although intercomponent digital audio interface reception (DIR) mode (i.e. S/PDIF stream receiving) is not natively supported by the McASP module, a specific TDM mode implementation for the McASP receivers allows an easy connection to external DIR components (for example, S/PDIF to I2S format converters).

The device have integrated eight McASP modules with:

- McASP1 and McASP2 supporting up to 16 channels with independent TX/RX clock/sync domain
- McASP3 through McASP8 modules supporting up to 4 channels with independent TX/RX clock/sync domain

Figure 24-108 shows the McASP modules in the device.

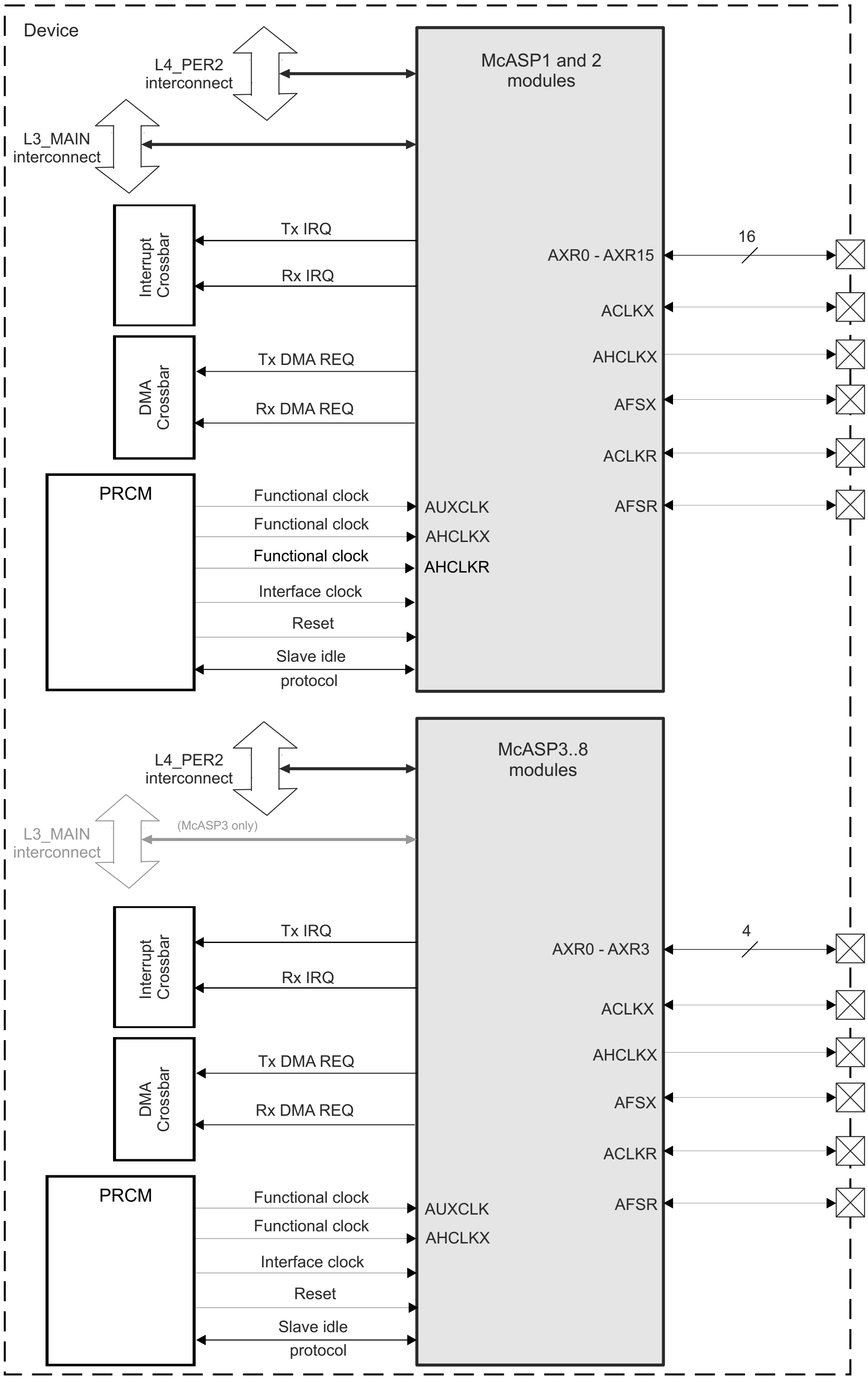


Figure 24-108. McASP Modules Overview

McASP module includes the following main features:

- Two modules (McASP1 and McASP2) supporting up to 16 channels each and independent TX/RX clock/sync domains.
- Six modules (McASP3, McASP4, McASP5, McASP6, McASP7, and McASP8) supporting up to 4 channels each and unified clock/sync domain.
- Independent serializer for each AXRx channel of each McASPx module.
- Idle request/acknowledge protocol
- A single 32-bit buffer per serializer for transmit and receive operations
- 2 x interconnect slave interface ports:
  - A configuration (CFG) port supplied with an internal L4-interconnect interface clock
  - A slave DMA data port synchronized with functional clock
- Two independent clock generator modules for transmit and receive.
  - Clocking flexibility allows the McASP to receive and transmit at different rates. For example, the McASP can receive data at 48 kHz but output up-sampled data at 96 kHz or 192 kHz.
- McASP module functional clock can be generated:
  - internally (master mode)
  - supplied over McASP serial interface (slave mode)
  - has a controllable functional clock divide ratio
- Independent transmit and receive modules, each includes:
  - Programmable clock and frame sync generator.
  - TDM streams from 2 to 32, and 384 time slots.
  - Support for time slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits.
  - Data formatter for bit manipulation.
- Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components.
- Wide variety of I2S and similar bit-stream format.
- Integrated digital audio interface transmitter (DIT):
  - S/PDIF, IEC60958-1, AES-3 formats.
  - Enhanced channel status/user data RAM.
- 384-slot TDM with external digital audio interface receiver (DIR) device.
  - For DIR reception, an external DIR receiver integrated circuit should be used with I2S output format and connected to the McASP receive section.
- Support for 2x DMA requests (one per direction):
  - 1 level-sensitive transmit direct memory access (DMA) request common for all of the McASP serializers
  - 1 level-sensitive receive direct memory access (DMA) request common for all of the McASP serializers
  - All transmit DMA requests are mapped to the device DMA crossbar
- One transmit interrupt request common for all serializers
- One receive interrupt request common for all serializers
- Each of the Rx and Tx interrupts is propagated to different host processors via the device Interrupt Crossbar

Note

Because a serializer receive and transmit channels data is shared on the same McASP data pin, user can choose to have either Tx or Rx function from a serializer, not both at the same time.

24.6.2 McASP Environment

This section describes the McASP application fields from an environment point of view (external connections), along with the McASP connectivity options. This section also lists all of the possible interfaces and describes the protocol and data format used in each case. Figure 24-109 shows the McASP modules in their environment in the device.

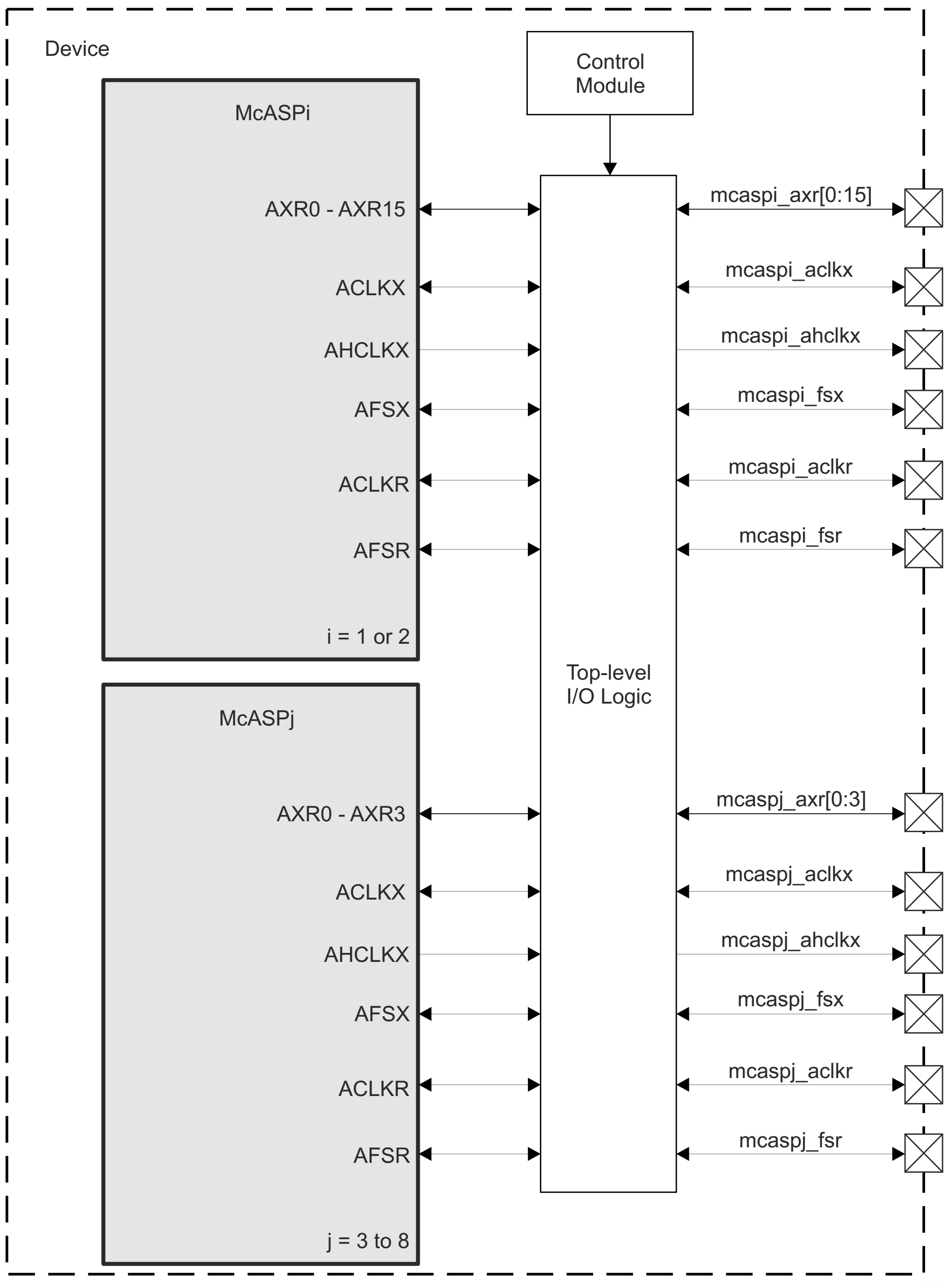


Figure 24-109. McASP Environment

24.6.2.1 McASP Signals

Table 24-325 describes the McASP pins, their corresponding signal names at device level and specifies their links to functions.

Table 24-325. McASP I/O Signals

Module Pin Name	Device Level Signal Name	I/O[(1)]	Description	Module Pin Reset Value
		McASP1 module		
AXR0	mcasp1_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp1_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp1_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp1_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
AXR4	mcasp1_axr[4]	I/O	Audio transmit/receive data - channel 4	HiZ
AXR5	mcasp1_axr[5]	I/O	Audio transmit/receive data - channel 5	HiZ
AXR6	mcasp1_axr[6]	I/O	Audio transmit/receive data - channel 6	HiZ
AXR7	mcasp1_axr[7]	I/O	Audio transmit/receive data - channel 7	HiZ
AXR8	mcasp1_axr[8]	I/O	Audio transmit/receive data - channel 8	HiZ
AXR9	mcasp1_axr[9]	I/O	Audio transmit/receive data - channel 9	HiZ
AXR10	mcasp1_axr[10]	I/O	Audio transmit/receive data - channel 10	HiZ
AXR11	mcasp1_axr[11]	I/O	Audio transmit/receive data - channel 11	HiZ
AXR12	mcasp1_axr[12]	I/O	Audio transmit/receive data - channel 12	HiZ
AXR13	mcasp1_axr[13]	I/O	Audio transmit/receive data - channel 13	HiZ
AXR14	mcasp1_axr[14]	I/O	Audio transmit/receive data - channel 14	HiZ
AXR15	mcasp1_axr[15]	I/O	Audio transmit/receive data - channel 15	HiZ
ACLKX	mcasp1_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp1_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp1_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp1_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp1_fsr	I/O	Receive frame synchronization	HiZ
		McASP2 module		
AXR0	mcasp2_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp2_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp2_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp2_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
AXR4	mcasp2_axr[4]	I/O	Audio transmit/receive data - channel 4	HiZ
AXR5	mcasp2_axr[5]	I/O	Audio transmit/receive data - channel 5	HiZ
AXR6	mcasp2_axr[6]	I/O	Audio transmit/receive data - channel 6	HiZ
AXR7	mcasp2_axr[7]	I/O	Audio transmit/receive data - channel 7	HiZ
AXR8	mcasp2_axr[8]	I/O	Audio transmit/receive data - channel 8	HiZ
AXR9	mcasp2_axr[9]	I/O	Audio transmit/receive data - channel 9	HiZ
AXR10	mcasp2_axr[10]	I/O	Audio transmit/receive data - channel 10	HiZ
AXR11	mcasp2_axr[11]	I/O	Audio transmit/receive data - channel 11	HiZ
AXR12	mcasp2_axr[12]	I/O	Audio transmit/receive data - channel 12	HiZ
AXR13	mcasp2_axr[13]	I/O	Audio transmit/receive data - channel 13	HiZ
AXR14	mcasp2_axr[14]	I/O	Audio transmit/receive data - channel 14	HiZ
AXR15	mcasp2_axr[15]	I/O	Audio transmit/receive data - channel 15	HiZ
ACLKX	mcasp2_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp2_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp2_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp2_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp2_fsr	I/O	Receive frame synchronization	HiZ

Table 24-325. McASP I/O Signals (continued)

Module Pin Name	Device Level Signal Name	I/O[1]	Description	Module Pin Reset Value
		McASP3 module		
AXR0	mcasp3_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp3_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp3_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp3_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
ACLKX	mcasp3_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp3_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp3_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp3_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp3_fsr	I/O	Receive frame synchronization	HiZ
		McASP4 module		
AXR0	mcasp4_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp4_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp4_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp4_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
ACLKX	mcasp4_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp4_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp4_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp4_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp4_fsr	I/O	Receive frame synchronization	HiZ
		McASP5 module		
AXR0	mcasp5_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp5_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp5_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp5_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
ACLKX	mcasp5_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp5_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp5_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp5_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp5_fsr	I/O	Receive frame synchronization	HiZ
		McASP6 module		
AXR0	mcasp6_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp6_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp6_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp6_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
ACLKX	mcasp6_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp6_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp6_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp6_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp6_fsr	I/O	Receive frame synchronization	HiZ
		McASP7 module		
AXR0	mcasp7_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp7_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp7_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ

Table 24-325. McASP I/O Signals (continued)

Module Pin Name	Device Level Signal Name	I/O[1]	Description	Module Pin Reset Value
AXR3	mcasp7_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
ACLKX	mcasp7_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp7_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp7_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp7_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp7_fsr	I/O	Receive frame synchronization	HiZ
		McASP8 module		
AXR0	mcasp8_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp8_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp8_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp8_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
ACLKX	mcasp8_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp8_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp8_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp8_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp8_fsr	I/O	Receive frame synchronization	HiZ

(1) I = Input; O = Output; I/O = Bidirectional

Note

For the mcaspx_aclkx, mcaspx_ahclkx and mcaspx_aclkr signals to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers should be set to 0x1 because of retiming purposes.

Note

The path from module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. For more information, refer to the *Pad Configuration Registers*, and *Control Module Register Manual*, in *Control Module*.

Note

A serializer AXR data pin is shared between the transmit and receive logic of that serializer. The direction of data is determined in the MCASP_PDIR and the function (Tx or Rx) is selected in the corresponding serializer control register MCASP_XRSRCTLn.

24.6.2.2 Protocols and Data Formats

24.6.2.2.1 Protocols Supported

The McASP supports a wide variety of protocols:

- Transmit section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - S/PDIF, IEC60958-1, AES-3 formats.
- Receive section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.

  - TDM stream of 384 time slots specifically designed for easy interface to external digital interface receiver (DIR) device transmitting DIR frames to McASP using the I2S protocol (one time slot for each DIR subframe).

The transmit and receive sections of the module may be individually programmed to support the following options on the basic serial protocol:

- Programmable clock and frame sync polarity (rising or falling edge): ACLKR/X, AHCLKR/X, and AFSR/X.
- Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported.
- Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot length.
- First-bit data delay: 0, 1, 2 bit clocks.
- Left/right alignment of word inside slot.
- Bit order: MSB first or LSB first.
- Bit mask/pad/rotate function.
  - Automatically aligns data internally in either Q31 or integer formats.
  - Automatically masks nonsignificant bits (sets to 0, 1, or extends value of another bit).

Note

In I2S mode, the transmit and receive sections can support simultaneous transfers on up to all serial data pins operating as 192 kHz stereo channels.

In DIT mode for McASP, additional features of the transmitters are:

- Transmit-only mode 384 time slots (subframe) per frame.
- Biphase encoded LVCMOS output
- Channel status RAM (384 bits).
- User data RAM (384 bits).
- Separate valid bit (V) for subframe A, B.
- Stereo Support Only (Mono means send data 2x via software)

In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to all serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for I2S mode, due to the need to generate Biphase Mark Encoded Data).

Note

The McASP does NOT natively support DIR-mode reception (i.e. receiving in the S/PDIF format). To allow this, the McASP can use a DIR-input to I2S-output converter implemented by an external device (i.e. external DIR component). To facilitate reception in this case, the TDM mode of McASP receivers logic is extended to support a non-standard 384-slot TDM stream.

Note

An external transceiver must be connected to the McASP port in the device to translate the electrical signals delivered by the McASP (1.2 V or 1.8 V LVCMOS levels) to the electrical levels of the S/PDIF standard.

24.6.2.2.2 Definition of Terms

The serial bitstream transmitted or received by a McASP serializer is a long sequence of 1s and 0s on an audio transmit/receive pins AXRn. However, the sequence has a hierarchical organization that can be described in terms of frames of data, slots, words, and bits.

A basic synchronous serial interface consists of three important components: clock, frame sync, and data. Figure 24-110 shows two of the three basic components: the clock signal (ACLKX/ACLKR) and the data signals AXRn. Figure 24-110 does not specify whether the clock is for transmit (ACLKX) or receive (ACLKR) because the definitions of terms apply to both receive and transmit interfaces. In operation, each transmitter and receiver uses the signals ACLKX and ACLKR as serial clock, respectively. Optionally, a receiver can use ACLKX as the serial clock when a transmitter and receiver (not from the same serializer) of the McASP are configured to operate synchronously.

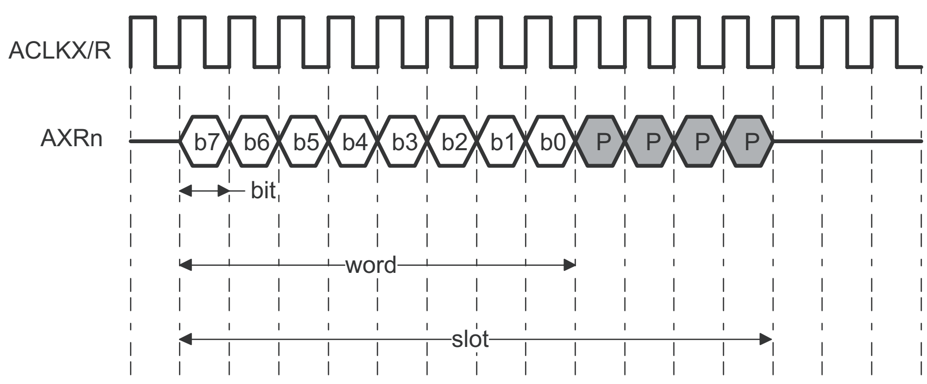
- Bit:

  A bit is the smallest entity in the serial data stream. The beginning and end of each bit is marked by an edge of the serial clock. The duration of a bit is a serial clock period. A '1' is represented by a logic high on AXRn pins for the entire duration of the bit. A 0 is represented by a logic low on an AXRn pin for the entire duration of the bit.
- Word:

  A word is a group of bits that make up the data being transferred between the McASP and the external device. Figure 24-110 shows an 8-bit word.
- Slot:

  A slot consists of the bits that make up the word and can consist of additional bits used to pad the word to a convenient number of bits for the interface between the McASP and the external device. In Figure 24-110, the audio data consists of only 8 bits of useful data (8-bit word), but it is padded with four 0s (12-bit slot) to satisfy the desired protocol in interfacing to an external device. Within a slot, the bits can be shifted out of the McASP on an AXRn pin with either MSB or LSB first.

  When the word size is smaller than the slot size, the word can be aligned to the left of the slot (beginning) or to the right of the slot (end). The additional bits in the slot not belonging to the word can be padded with 0, 1, or with one of the bits (typically, the MSB or LSB) from the data word, i.e. left-aligned words within a slot are terminated with padding bits and right-aligned words within a slot are preceded by padding bits to fit in the slot size. Figure 24-111 shows these options.



(1) b7:b0 - bits. Bits b7 to b0 form a word.

(2) P - pad bits. Bits b7 to b0, together with the 4 pad bits, form a slot.

(3) In this example, the data is transmitted MSB first, left-aligned.

mcasp-003

Figure 24-110. Definition of Bit, Word, and Slot

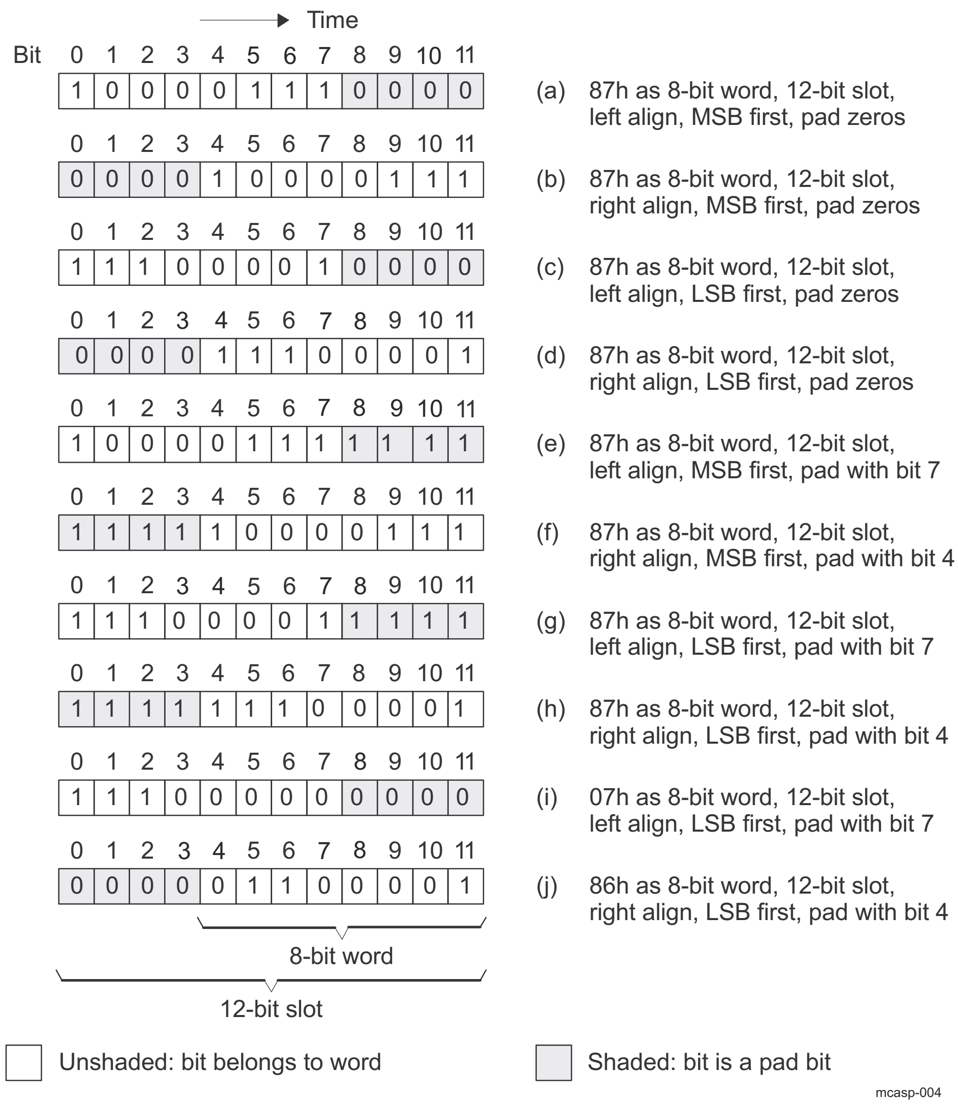


Figure 24-111. Bit Order and Word Alignment Within a Slot Examples

- Frame

  The third basic element of a synchronous serial interface is the frame synchronization signal, also referred to as frame sync in this chapter. A frame contains one or multiple slots, as determined by the desired protocol. Figure 24-112 shows an example frame of data and the frame definitions. In operation, the transmitter uses AFSX, and the receiver - AFSR signal. Figure 24-112 does NOT specify whether the frame sync (FS) is for transmit (AFSX) or receive (AFSR) because the definitions of terms apply to both receive and transmit interfaces. In operation, each transmitter/receiver uses AFSX/AFSR as a frame synchronization signal, respectively. Optionally, the receiver can use AFSX as the frame sync when the transmitter and receiver of the McASP are configured to operate synchronously. This example shows two slots in a frame (I2S format) and a frame-sync (FS) duration of a slot length.

  This section shows only the generic definition of the frame sync. For more information about the frame-sync formats required for the transfer modes and protocols (TDM-mode and DIT-mode supported formats), see Section 24.6.2.2.3, *TDM Format* and Section 24.6.2.2.5, *S/PDIF-Coding Format*.

Note

All of the McASP serializers share the same, device pad accessible, clock and frame signals, as follows:

– AHCLKX, ACLKX, and AFSX for the transmitting section
– ACLKR and AFSR for the receiving section

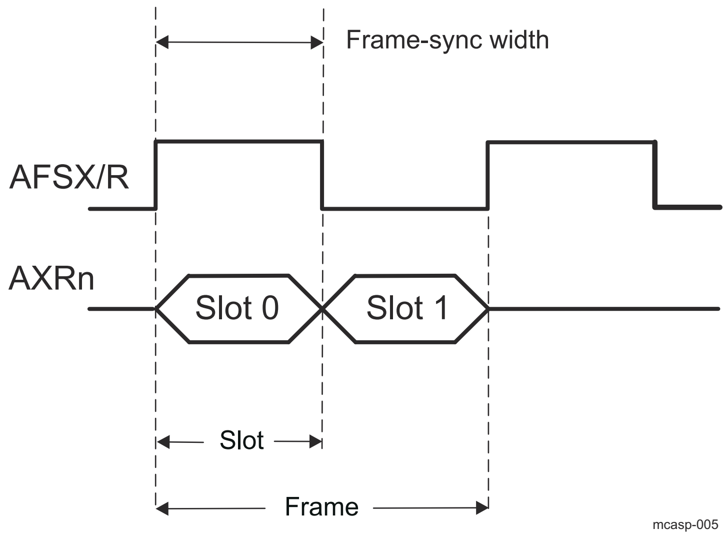


Figure 24-112. Definition of Frame and Frame-Sync Width

The following terms are used throughout this chapter:

- TDM: Time-division multiplexed. See Section 24.6.2.2.3 for details on the TDM protocol
- I2S: Inter-Integrated Sound protocol, commonly used on audio interfaces. The McASP supports the I2S protocol as part of the TDM mode (when configured as a 2-slot frame).
- DIT: Digital audio interface transmit. The McASP supports transmitting in S/PDIF format on each AXRn data pin.
- DIR: Digital audio interface receive. The McASP does NOT natively support receiving in S/PDIF format on AXRn data pins and requires an external DIR-to-TDM or DIR-to-I2S converter chip for a DIR-frame reception.
- Slot or time slot: For DIT/DIR format, a McASP time slot corresponds to a DIT/DIR subframe.

24.6.2.2.3 TDM Format

The TDM format is used to transfer data between the host CPU and one or more analog-to-digital converter (ADC), digital-to-analog converter (DAC), or S/PDIF receiver (DIR) devices. An example for a 6-slot (channel) TDM transmission on one McASP data pin - AXRn is illustrated on Figure 24-113.

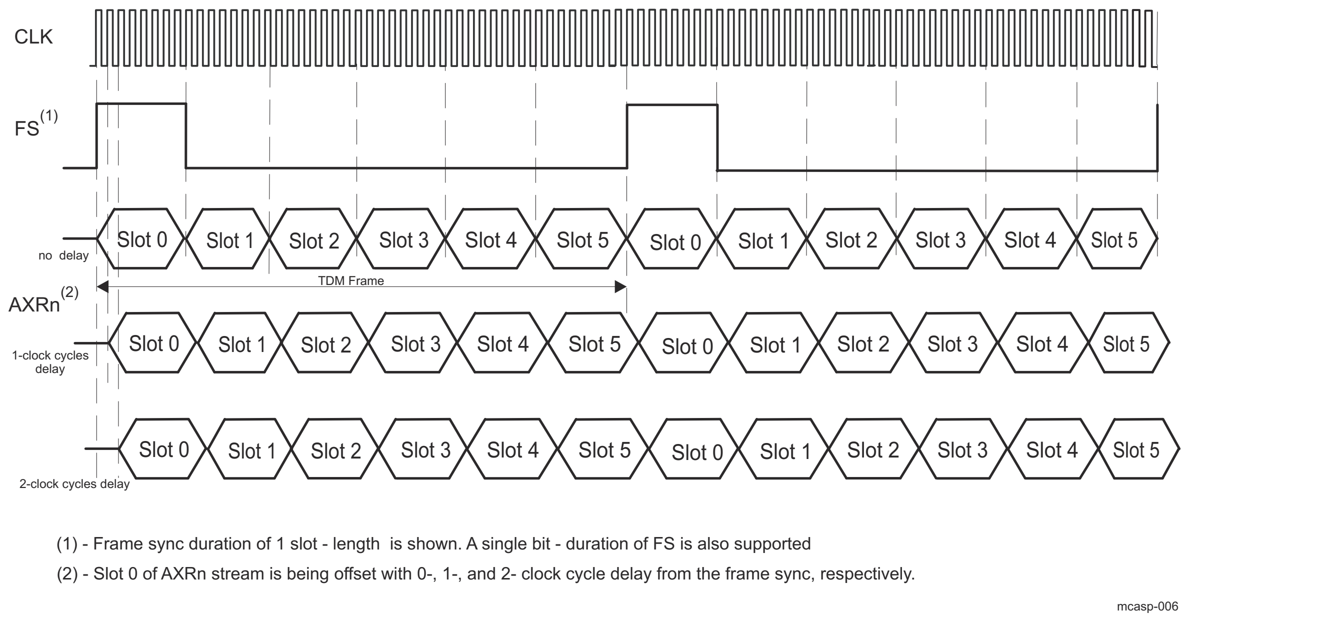


Figure 24-113. TDM Format - 6 channel example

The TDM format uses three signals In a basic synchronous serial interface: data (AXRn), clock (CLK) and frame sync (FS). The data signal present on AXRn pin is fully synchronous to the serial clock (ACLKX or ACLKR). The data bits are grouped into words and slots (see also Section 24.6.2.2.2), the latter being also refered to as the "time-slots" or "channels" in TDM terminology. A frame consists of multiple time-slots. Each TDM frame is marked by the frame sync signal (AFSX or AFSR). The TDM transfer is continuous and periodic, with no delays between slots.

Within a certain frame, the last bit of slot N is followed immediately on the next serial clock with the first bit of the next slot N+1. On the boundary between two adjacent TDM-frames, the last bit of the last slot from the frame M, is followed immediately on the next clock cycle with the first bit of the first slot from the next frame M+1. For McASP, there is an option to offset the first bit of the first slot with a 0-,1- or 2-cycle delay from the frame sync signal.

The frame sync - AFSX/AFSR only marks the beginning of slot 0 and start of a new frame. Since it does not determine the boundaries of a slot, there is a requirement for a connected transmitter and receiver to agree on the number of transferred bits per slot.

In a typical audio system involving McASP module, a single TDM data frame is transferred during each sample period Ts of a data converter. The user has following choices to implement multiple channels:

- Use more data slots (on a price of higher speed serial clock) per frame transmitted/received on just one of the available McASP data pins AXRn.
- Use less number of slots per TDM frame (requires a slower serial clock), making them available on several of the McASP pins AXRn.

24.6.2.2.4 I2S Format

The TDM transfer mode of the McASP supports the I2S format when frame is configured to have 2 slots. I2S format is specifically designed to transfer a stereo channel (left and right) over a single data pin AXRn. "Slots" are also commonly referred to as "channels". The frame width duration in the I2S format equals size of a slot. The frame signal is also referred to as "word select" in the I2S format.

The I2S protocol is illustrated on Figure 24-114.

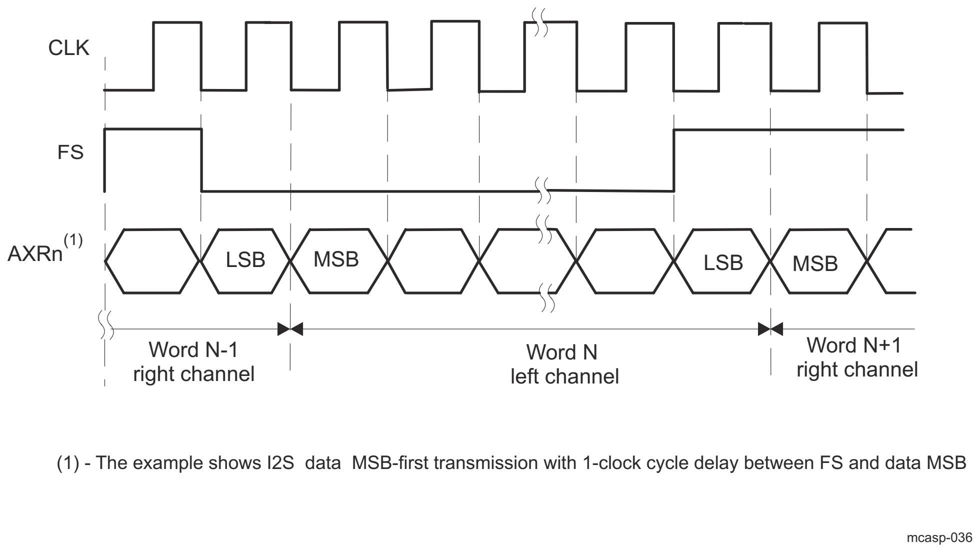


Figure 24-114. I2S Format Overview

24.6.2.2.5 S/PDIF Coding Format

The McASP transmitter supports the S/PDIF format with 3.3V biphase-mark encoded output. The S/PDIF format is supported by the DIT- transfer mode of the McASP. This section briefly discusses the S/PDIF coding format.

Note

The DIR- reception of S/PDIF format frames is NOT natively supported from the device McASP. For this purpose, an external DIR-to-TDM transfer mode adapter can be used between the remote device S/PDIF transmitter output and the McASP TDM-only compatible receiver input.

24.6.2.2.5.1 Biphase-Mark Code

In S/PDIF format, the digital signal is coded using the biphase-mark code (BMC). For each serializer transmitter n, the clock, frame, and data are embedded in only one signal - the data signal AXRn. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. These two logical states form a cell. The duration of the cell, which equals the duration of the data bit, is called a time interval. A logical 1 is represented by two transitions of the signal within a time interval, which corresponds to a cell with logical states 01 or 10. A logical 0 is represented by one transition within a time interval, which corresponds to a cell with logical states 00 or 11. In addition, the logical level at the start of a cell is inverted from the level at the end of the previous cell. Figure 24-115 and Table 24-326 show how data is encoded to the BMC format.

As shown in Figure 24-115, the clock frequency is twice the unencoded data bit rate. In addition, the clock is always programmed to 128 × f_S, where f_S is the sample rate (see Section 24.6.2.2.5.3, *Frame Format*, for details on how this clock rate is derived based on the S/PDIF format). The device receiving in S/PDIF format can recover the clock and frame information from the BMC signal.

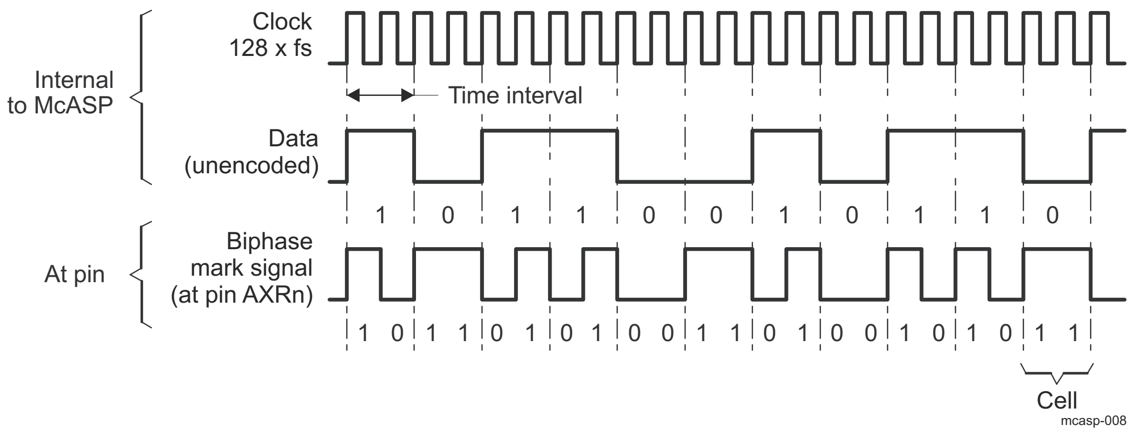


Figure 24-115. Biphase-Mark Code

Table 24-326. Biphase-Mark Encoder

Data (Unencoded)	Previous State at Pin AXRn	BMC-Encoded Cell Output at Pin AXRn
0	0	11
0	1	00
1	0	10
1	1	01

24.6.2.2.5.2 S/PDIF Subframe Format

Every audio sample transmitted in a subframe consists of 32 S/PDIF time intervals (or cells), numbered 0 to 31. Figure 24-116 shows a subframe.

- Time intervals 0–3 carry one of the three permitted preambles to signify the type of audio sample in the current subframe. The preamble is not encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row. See Table 24-327.
- Time intervals 4–27 carry the audio sample word in linear 2s-complement representation. The MSB is carried by time interval 27. When a 24-bit coding range is used, the LSB is in time interval 4. When a 20-bit coding range is used, time intervals 8-27 carry the audio sample word with the LSB in time interval 8. Time intervals 4–7 may be used for other applications and are designated auxiliary sample bits.
- If the source provides fewer bits than the interface allows (20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field can carry any other information.
- Time interval 28 carries the validity bit (V) associated with the main data field in the subframe.
- Time interval 29 carries the user data channel (U) associated with the main data field in the subframe.
- Time interval 30 carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.
- Time interval 31 carries a parity bit (P) such that time intervals 4–31 carry an even number of 1s and an even number of 0s (even parity). As listed in Table 24-327, the preambles (time intervals 0–3) are also defined with even parity.

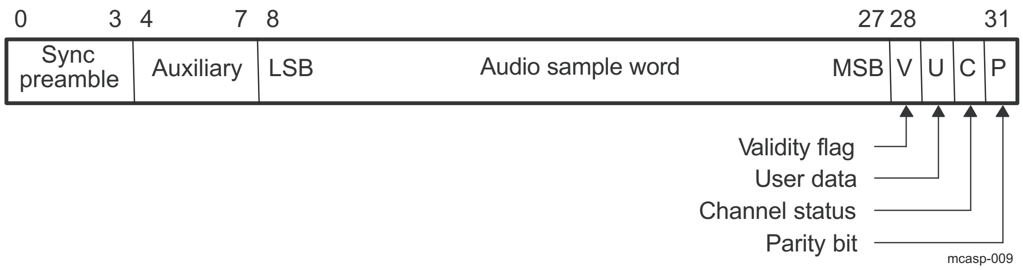


Figure 24-116. S/PDIF Subframe Format

As listed in Table 24-327, the McASP DIT generates only one polarity of preambles, and it assumes the previous logical state is 0. This is because the McASP assures an even-polarity encoding scheme when transmitting in DIT mode. If an underrun condition occurs, the DIT resynchronizes to the correct logic level on the AXRn pin before continuing with the next transmission.

Table 24-327. Preamble Codes

Preamble Code[(1)]	Previous Logical State	Logical States on pin AXRn[(2)]	Description
B (or Z)	0	1110 1000	Start of a block and subframe 1
M (or X)	0	1110 0010	Subframe 1
W (or Y)	0	1110 0100	Subframe 2

(1) Historically, preamble codes are referred to as B, M, and W. For use in professional applications, preambles are referred to as Z, X, and Y, respectively.
(2) The preamble is not BMC-encoded. Each logical state is synchronized to the serial clock. These eight logical states make up time slots (cells) 0 to 3 in the S/PDIF stream.

24.6.2.2.5.3 Frame Format

An S/PDIF frame is composed of two subframes (see Figure 24-117). For linear coded audio applications, the rate of frame transmission normally corresponds exactly to the source sampling frequency f_s. The S/PDIF format clock rate is therefore 128 × f_s (128 = 32 cells per subframe × 2 clocks per cell × 2 subframes per sample). For example, for an S/PDIF stream at a 192-kHz sampling frequency, the serial clock is 128 × 192 kHz = 24.58 MHZ.

In 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive subframes. Both subframes contain valid data (cell 28 validity bits for A- and B- channels, both set to '0'). The first subframe (left or A channel in stereophonic operation and primary channel in monophonic operation) normally starts with preamble M. However, the preamble of the first subframe changes to preamble B once every 192 frames to identify the start of the block structure used to organize the channel status information. The second subframe (right or B channel in stereophonic operation and secondary channel in monophonic operation) always starts with preamble W.

In single-channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried in the first subframe and may be duplicated in the second subframe. If the second subframe is not carrying duplicate data, cell 28 (validity bit) is set to logical 1.

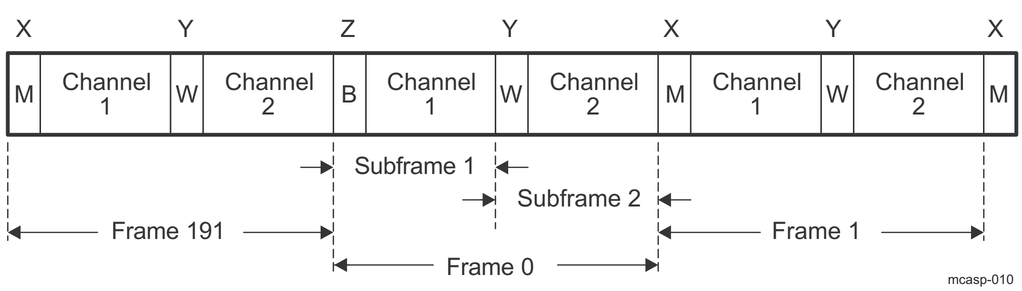


Figure 24-117. S/PDIF Frame Format

24.6.3 McASP Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

McASP module includes the following features:

- Slave idle protocol
- One DMA request for transmit event
- One DMA request for receive event
- One interrupt request (IRQ) for transmit
- One interrupt request (IRQ) for receive

Figure 24-118 shows McASP integration.

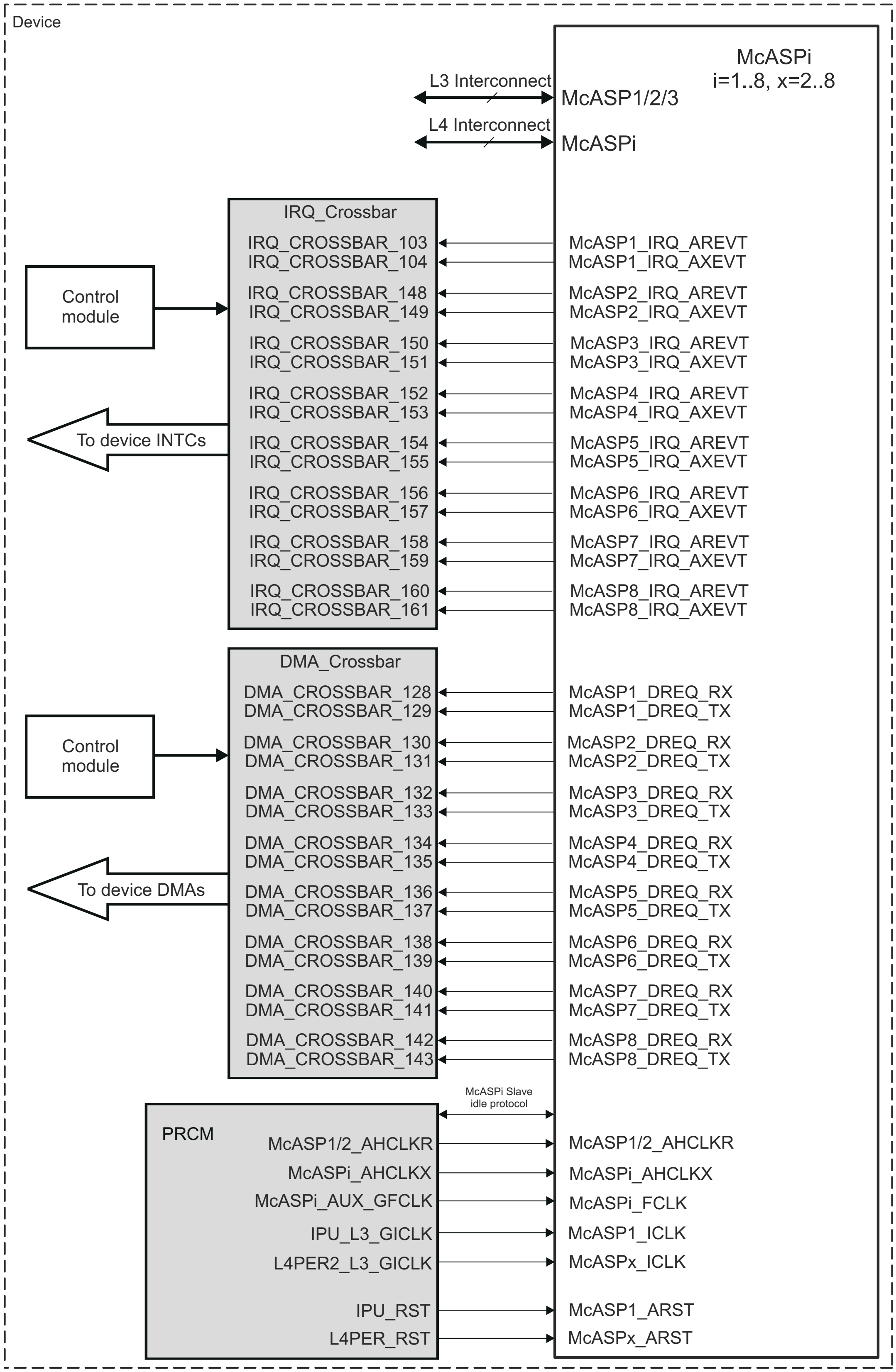


Figure 24-118. McASP Integration

Note

For more information about the slave idle protocol, see *Module Level Clock Management* of *Power, Reset, and Clock Management*.

Table 24-328 through Table 24-330 summarize McASP integration in the device.

Table 24-328. McASP Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
McASP1	PD_COREAON	No	L3_MAIN L4_PER2
McASP2	PD_COREAON	No	L3_MAIN L4_PER2
McASP3	PD_COREAON	No	L3_MAIN L4_PER2
McASP4	PD_COREAON	No	L4_PER2
McASP5	PD_COREAON	No	L4_PER2
McASP6	PD_COREAON	No	L4_PER2
McASP7	PD_COREAON	No	L4_PER2
McASP8	PD_COREAON	No	L4_PER2

Table 24-329. McASP Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
McASP1	MCASP1_AHCLKR	MCASP1_AHCLKR	PRCM	McASP1 AHCLKR receive high-frequency master clock
	MCASP1_AHCLKX	MCASP1_AHCLKX		McASP1 AHCLKX transmit high-frequency master clock
	MCASP1_FCLK	MCASP1_AUX_GFCLK		McASP1 functional clock
	MCASP1_ICLK	IPU_L3_GICLK		McASP1 interface clock
McASP2	MCASP2_AHCLKR	MCASP2_AHCLKR	PRCM	McASP2 AHCLKR receive high-frequency master clock
	MCASP2_AHCLKX	MCASP2_AHCLKX		McASP2 AHCLKX transmit high-frequency master clock
	MCASP2_FCLK	MCASP2_AUX_GFCLK		McASP2 functional clock
	MCASP2_ICLK	L4PER2_L3_GICLK		McASP2 interface clock
McASP3	MCASP3_AHCLKX	MCASP3_AHCLKX	PRCM	McASP3 AHCLKX transmit high-frequency master clock
	MCASP3_FCLK	MCASP3_AUX_GFCLK		McASP3 functional clock
	MCASP3_ICLK	L4PER2_L3_GICLK		McASP3 interface clock
McASP4	MCASP4_AHCLKX	MCASP4_AHCLKX	PRCM	McASP4 AHCLKX transmit high-frequency master clock
	MCASP4_FCLK	MCASP4_AUX_GFCLK		McASP4 functional clock
	MCASP4_ICLK	L4PER2_L3_GICLK		McASP4 interface clock
McASP5	MCASP5_AHCLKX	MCASP5_AHCLKX	PRCM	McASP5 AHCLKX transmit high-frequency master clock
	MCASP5_FCLK	MCASP5_AUX_GFCLK		McASP5 functional clock
	MCASP5_ICLK	L4PER2_L3_GICLK		McASP5 interface clock
McASP6	MCASP6_AHCLKX	MCASP6_AHCLKX	PRCM	McASP6 AHCLKX transmit high-frequency master clock

Table 24-329. McASP Clocks and Resets (continued)

	MCASP6_FCLK	MCASP6_AUX_GFCLK		McASP6 functional clock
	MCASP6_ICLK	L4PER2_L3_GICLK		McASP6 interface clock
McASP7	MCASP7_AHCLKX	MCASP7_AHCLKX	PRCM	McASP7 AHCLKX transmit high-frequency master clock
	MCASP7_FCLK	MCASP7_AUX_GFCLK		McASP7 functional clock
	MCASP7_ICLK	L4PER2_L3_GICLK		McASP7 interface clock
McASP8	MCASP8_AHCLKX	MCASP8_AHCLKX	PRCM	McASP8 AHCLKX transmit high-frequency master clock
	MCASP8_FCLK	MCASP8_AUX_GFCLK		McASP8 functional clock
	MCASP8_ICLK	L4PER2_L3_GICLK		McASP8 interface clock
		Resets		
McASP1	MCASP1_ARST	IPU_RST	PRCM	McASP1 reset
McASP2	MCASP2_ARST	L4PER_RST	PRCM	McASP2 reset
McASP3	MCASP3_ARST	L4PER_RST	PRCM	McASP3 reset
McASP4	MCASP4_ARST	L4PER_RST	PRCM	McASP4 reset
McASP5	MCASP5_ARST	L4PER_RST	PRCM	McASP5 reset
McASP6	MCASP6_ARST	L4PER_RST	PRCM	McASP6 reset
McASP7	MCASP7_ARST	L4PER_RST	PRCM	McASP7 reset
McASP8	MCASP8_ARST	L4PER_RST	PRCM	McASP8 reset

Table 24-330. McASP Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
McASP1	MCASP1_IRQ_AREVT	IRQ_CROSSBAR_103	MPU_IRQ_108	McASP1 module receive interrupt request
	MCASP1_IRQ_AXEVT	IRQ_CROSSBAR_104	MPU_IRQ_109	McASP1 module transmit interrupt request
McASP2	MCASP2_IRQ_AREVT	IRQ_CROSSBAR_148	-	McASP2 module receive interrupt request
	MCASP2_IRQ_AXEVT	IRQ_CROSSBAR_149	-	McASP2 module transmit interrupt request
McASP3	MCASP3_IRQ_AREVT	IRQ_CROSSBAR_150	-	McASP3 module receive interrupt request
	MCASP3_IRQ_AXEVT	IRQ_CROSSBAR_151	-	McASP3 module transmit interrupt request
McASP4	MCASP4_IRQ_AREVT	IRQ_CROSSBAR_152	-	McASP4 module receive interrupt request
	MCASP4_IRQ_AXEVT	IRQ_CROSSBAR_153	-	McASP4 module transmit interrupt request
McASP5	MCASP5_IRQ_AREVT	IRQ_CROSSBAR_154	-	McASP5 module receive interrupt request
	MCASP5_IRQ_AXEVT	IRQ_CROSSBAR_155	-	McASP5 module transmit interrupt request
McASP6	MCASP6_IRQ_AREVT	IRQ_CROSSBAR_156	-	McASP6 module receive interrupt request
	MCASP6_IRQ_AXEVT	IRQ_CROSSBAR_157	-	McASP6 module transmit interrupt request
McASP7	MCASP7_IRQ_AREVT	IRQ_CROSSBAR_158	-	McASP7 module receive interrupt request
	MCASP7_IRQ_AXEVT	IRQ_CROSSBAR_159	-	McASP7 module transmit interrupt request

Table 24-330. McASP Hardware Requests (continued)

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
McASP8	MCASP8_IRQ_AREVT	IRQ_CROSSBAR_160	-	McASP8 module receive interrupt request
	MCASP8_IRQ_AXEVT	IRQ_CROSSBAR_161	-	McASP8 module transmit interrupt request
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
McASP1	MCASP_DREQ_RX	DMA_CROSSBAR_128	DMA_DSP1_DREQ_0 DMA_DSP2_DREQ_0	McASP module receive event request
	MCASP_DREQ_TX	DMA_CROSSBAR_129	DMA_DSP1_DREQ_1 DMA_DSP2_DREQ_1	McASP module transmit event request
McASP2	MCASP2_DREQ_RX	DMA_CROSSBAR_130	DMA_DSP1_DREQ_2 DMA_DSP2_DREQ_2	McASP2 module receive event request
	MCASP2_DREQ_TX	DMA_CROSSBAR_131	DMA_DSP1_DREQ_3 DMA_DSP2_DREQ_3	McASP2 module transmit event request
McASP3	MCASP3_DREQ_RX	DMA_CROSSBAR_132	DMA_DSP1_DREQ_4 DMA_DSP2_DREQ_4	McASP3 module receive event request
	MCASP3_DREQ_TX	DMA_CROSSBAR_133	DMA_DSP1_DREQ_5 DMA_DSP2_DREQ_5	McASP3 module transmit event request
McASP4	MCASP4_DREQ_RX	DMA_CROSSBAR_134	DMA_DSP1_DREQ_6 DMA_DSP2_DREQ_6	McASP4 module receive event request
	MCASP4_DREQ_TX	DMA_CROSSBAR_135	DMA_DSP1_DREQ_7 DMA_DSP2_DREQ_7	McASP4 module transmit event request
McASP5	MCASP5_DREQ_RX	DMA_CROSSBAR_136	DMA_DSP1_DREQ_8 DMA_DSP2_DREQ_8	McASP5 module receive event request
	MCASP5_DREQ_TX	DMA_CROSSBAR_137	DMA_DSP1_DREQ_9 DMA_DSP2_DREQ_9	McASP5 module transmit event request
McASP6	MCASP6_DREQ_RX	DMA_CROSSBAR_138	DMA_DSP1_DREQ_10 DMA_DSP2_DREQ_10	McASP6 module receive event request
	MCASP6_DREQ_TX	DMA_CROSSBAR_139	DMA_DSP1_DREQ_11 DMA_DSP2_DREQ_11	McASP6 module transmit event request
McASP7	MCASP7_DREQ_RX	DMA_CROSSBAR_140	DMA_DSP1_DREQ_12 DMA_DSP2_DREQ_12	McASP7 module receive event request
	MCASP7_DREQ_TX	DMA_CROSSBAR_141	DMA_DSP1_DREQ_13 DMA_DSP2_DREQ_13	McASP7 module transmit event request
McASP8	MCASP8_DREQ_RX	DMA_CROSSBAR_142	DMA_DSP1_DREQ_14 DMA_DSP2_DREQ_14	McASP8 module receive event request
	MCASP8_DREQ_TX	DMA_CROSSBAR_143	DMA_DSP1_DREQ_15 DMA_DSP2_DREQ_15	McASP8 module transmit event request

Note

The **Default Mapping** column in Table 24-330 *McASP Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.

For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

For more information about the device DSP1_EDMA and DSP2_EDMA modules, see *DSP Integrated EDMA Subsystem*.

Note

- For the description of the interrupt source, see Section 24.6.4.12, *McASP Events and Interrupt Requests*.
- For the description of the DMA source, see Section 24.6.4.13, *DMA Requests*.

24.6.4 McASP Functional Description

In the text throughout this section a single instance of McASP is described assuming that all modules are functionally identical. For module availability and integration differences, see Section 24.6.2, *McASP Environment*, and Section 24.6.3, *McASP Integration*

24.6.4.1 McASP Block Diagram

Figure 24-119 shows the major blocks of the McASP module. McASP1 and McASP2 have 16 serializers each, and McASPn (n = 3 to 8) have 4 serializers each. The serializers share a clock and frame-sync generator, format unit, and error-checking logic independently for the receive and transmit part.

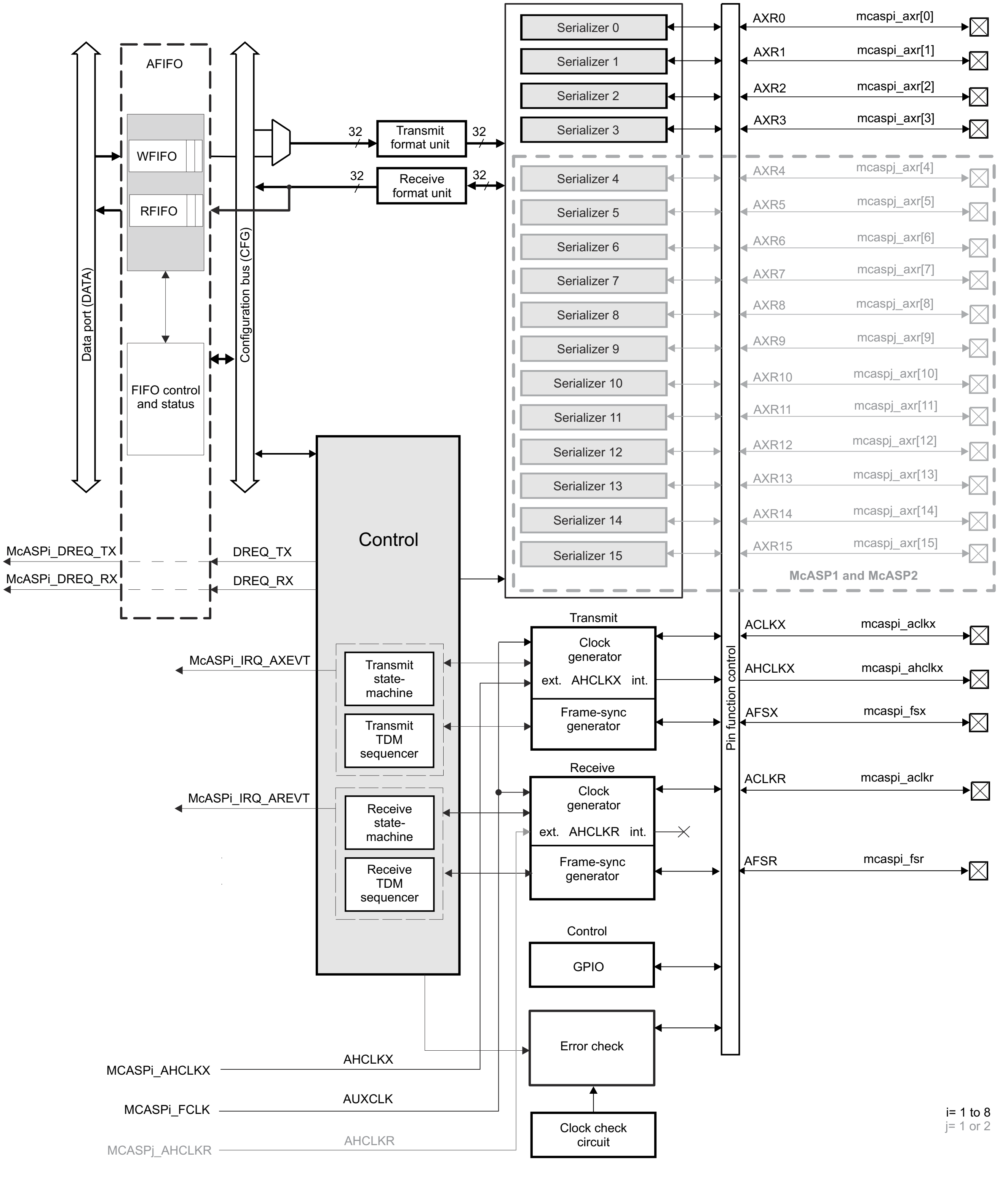


Figure 24-119. McASP Module Block Diagram

Note

The internal and external clocks mentioned in this section are with respect to clock and frame-sync generator modules.

24.6.4.2 McASP Clock and Frame-Sync Configurations

There are three scenarios to provide clock source signals for the Tx part and four scenarios for the Rx part of the McASP serializers. The first three scenarios are identical between the Tx and Rx part of the McASP. They feature an asynchronous operation between receiver and transmitter channels using independent Tx/Rx bit rate clock sources (either internal or external).

In the first scenario, the transmit - XCLK and receive - RCLK serial clocks (clock at the bit rate) are generated internally by passing through a couple of clock dividers off the internal functional clock source (AUXCLK). In this case, the bit rate clock is generated internally and is driven out on the pin ACLKX for the Tx part and pin ACLKR for the Rx part, respectively. An internally generated high-frequency clock can be optionally driven out onto the AHCLKX pin for the Tx part to serve as a reference clock for other components in the system.

In the second scenario, an external for the device clock, is passed on the ACLKX (for the TX part) and ACLKR (for the RX part) pins which are configured as inputs. In this case the Rx- /Tx- high-speed clock logic is bypassed for the XCLK/RCLK generation.

In the third (mixed) scenario, an externally driven (master) high-frequency clock is applied on the AHCLKX (for the TX part) pin, which is configured as input. In this case the AHCLKX clock frequency can be divided down via programming the ACLKX associated divider to produce the necessary bit rate clock. The high-speed clock divider can NOT be used.

In the fourth clock generation scenario the bit rate clock for McASP receivers - RCLK is derived from the bit rate clock of the McASP transmitters - XCLK for a synchronous operation between transmitters and receivers. Hence, the whole receiver clock generator logic is bypassed.

A typical role of the McASP frame sync signal is to carry the left/right clock (LRCLK) signal when transmitting and receiving stereo data.

For an asynchronous operation, the AFSX (Tx part) and AFSR (Rx part) frame synchronization signals can be sourced internally or delivered externally independently for the Tx and Rx channels. During synchronous operation the receive frame sync - AFSR signal is derived from the transmit frame sync - AFSX signal. A synchronous and asynchronous mode applies to bit rate clock and frame sync signals at the same time.

24.6.4.2.1 McASP Transmit Clock

The transmit high-speed and transmit clock configuration is controlled by the following registers:

- MCASP_ACLKXCTL
- MCASP_AHCLKXCTL

In case, the transmit bit clock, ACLKX, is generated internally, the MCASP_ACLKXCTL[5] CLKXM bit must be set to 1. Thus, the clock is divided down by a programmable bit clock divider (the MCASP_ACLKXCTL[4:0] CLKXDIV bit field) from the source signal.

If the transmit high-frequency master clock, AHCLKX, is also sourced internally (that is first scenario described in Section 24.6.4.2, the MCASP_AHCLKXCTL[15] HCLKXM bit must be set to 1. Thus, the clock is divided down by a programmable high-clock divider (the MCASP_AHCLKXCTL[11:0] HCLKXDIV bit field) from the McASP internal clock source AUXCLK.

Internally, the McASP always shifts transmit data at the rising edge of the internal transmit clock - XCLK, (see Figure 24-120). The CLKXP mux determines if ACLKX needs to be inverted to become XCLK. If MCASP_ACLKXCTL[7] CLKXP = 0, the CLKXP mux directly passes ACLKX signal to XCLK. As a result, the McASP shifts transmit data at the rising edge of ACLKX. If MCASP_ACLKXCTL[7] CLKXP = 1, the CLKX mux passes the inverted version of ACLKX to XCLK. As a result, the McASP shifts transmit data at the falling edge of ACLKX.

It can be seen in Figure 24-120 that XCLK is propagated to the Rx clock logic, to allow an internally synchronous operation between McASP transmitters and receivers. This is used for example in the McASP loopback mode.

Note

The polarity of ACLKX can be controlled in MCASP_ACLKXCTL[7] CLKXP, regardless of ACLKX signal being internally or externally sourced.

In addition, there is an option to invert polarity of the AHCLKX master high speed clock via writing the MCASP_AHCLKXCTL[14] HCLKXP bit.

Note

In a similar way, the polarity of AHCLKX clock can be controlled in MCASP_AHCLKXCTL[14] HCLKXP, regardless of the AHCLKX signal being internally or externally sourced.

Figure 24-120 is the block diagram of the transmit clock generator.

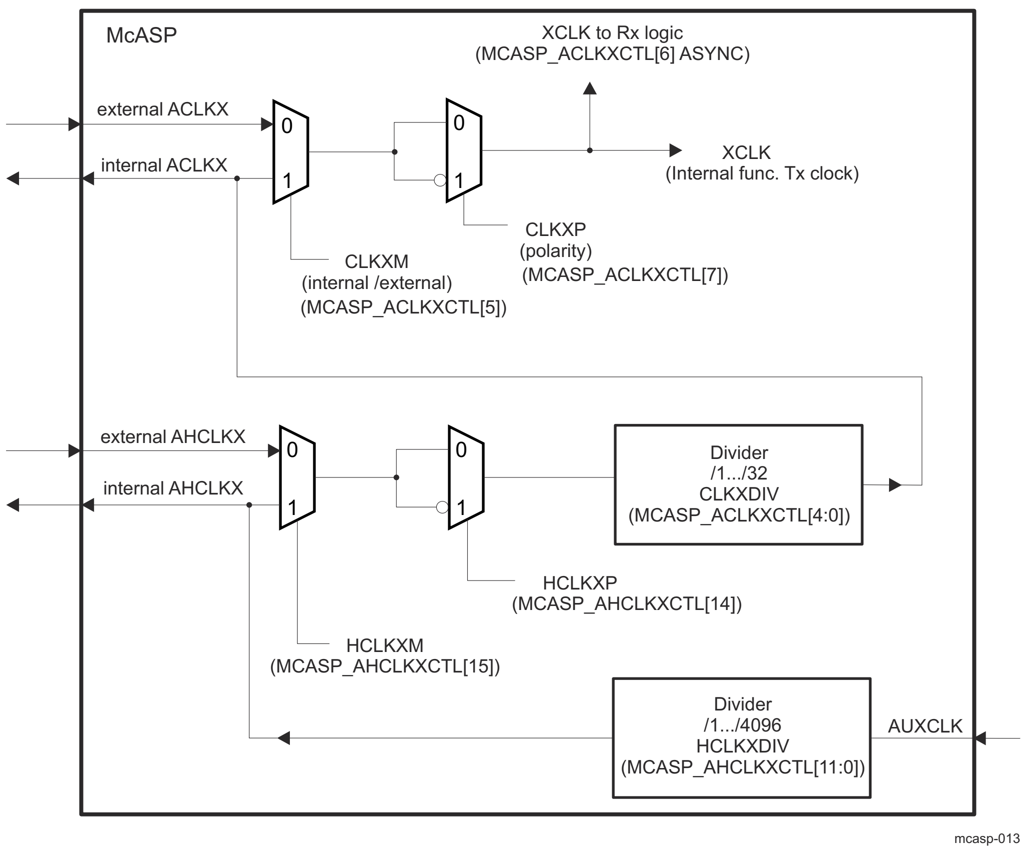


Figure 24-120. Transmit Clock Generator Block Diagram

Note

In this device:

- ACLKX is mapped on the device ball mcaspi_aclkx, where i = 1 to 8
- internal AHCLKX is mapped on the device ball mcaspi_ahclkx
- external AHCLKX is mapped on MCASPi_AHCLKX clock from the PRCM

For more on McASP integration, see Section 24.6.2, *McASP Environment*, and Section 24.6.3, *McASP Integration*.

24.6.4.2.2 McASP Receive Clock

The McASP receive clock generator is built on a very similar to the transmit clock generator (but independent) circuit.

The receive clock configuration is controlled by the following registers:

- MCASP_ACLKRCTL
- MCASP_AHCLKRCTL

In case, the receive bit clock, ACLKR, is generated internally (but asynchronously to XCLK), the MCASP_ACLKRCTL[5] CLKRM bit must be set to 1. Thus, the clock is divided down by a programmable bit clock divider (the MCASP_ACLKRCTL[4:0] CLKRDIV bit field) from the source signal.

If the receive high-frequency master clock, AHCLKR, is also sourced internally (that is, first scenario described in Section 24.6.4.2) and the MCASP_AHCLKRCTL[15] HCLKRM bit must be set to 1. Thus, the clock is divided down by a programmable high-clock divider (the MCASP_AHCLKRCTL[11:0] HCLKRDIV bit field) from the McASP internal clock source AUXCLK.

Note

The polarity of ACLKR can be controlled in MCASP_ACLKRCTL[7] CLKRP, regardless of ACLKR signal being internally or externally sourced.

In a similar way, the polarity of AHCLKR clock can be controlled in MCASP_AHCLKRCTL[14] HCLKRP, regardless of the AHCLKR signal being internally or externally sourced.

There is an option for the McASP receiver to be configured to operate synchronously to the ACLKX and AFSX signals. The XCLK output of the Tx Clock generator (see Figure 24-120 and Figure 24-121) becomes source of the receive clock (RCLK output), when the MCASP_ACLKXCTL[6] ASYNC bit in the transmit clock control register is set to '0b0'. For more information, refer to Section 24.6.4.2.4.

Figure 24-121 is the block diagram of the receive clock generator.

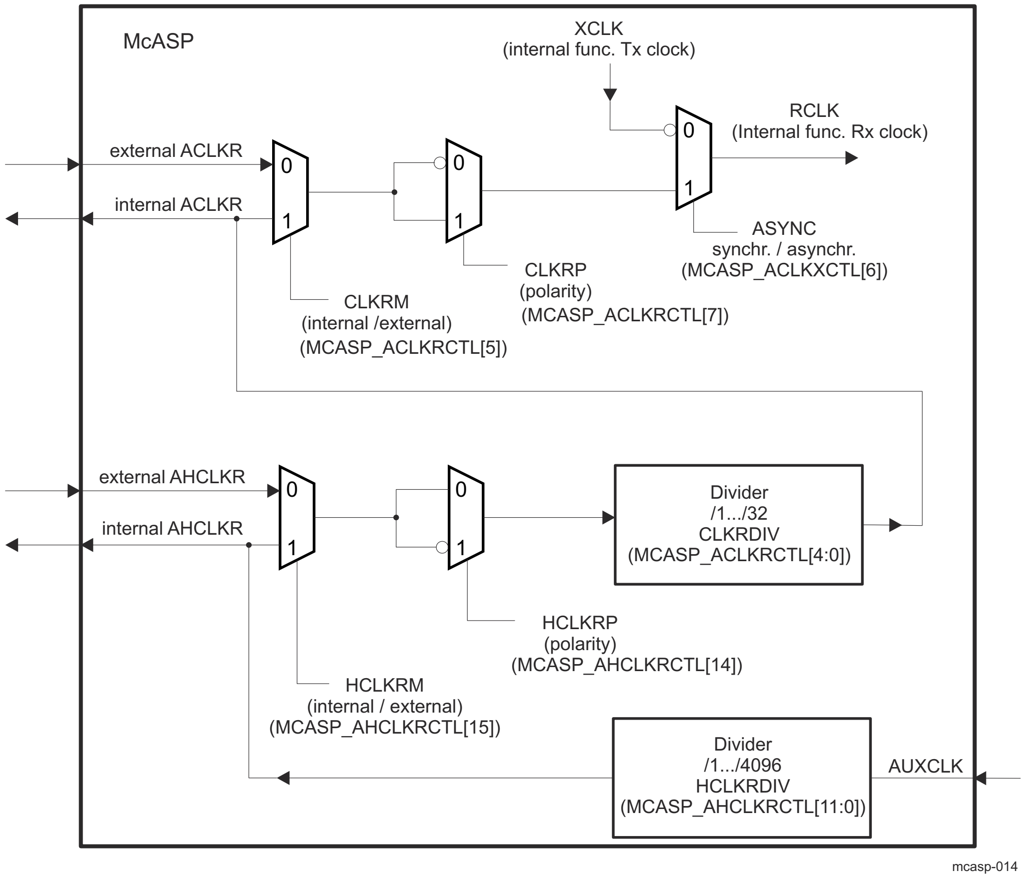


Figure 24-121. Receive Clock Generator Block Diagram

Note

In this device:

- ACLKR is mapped on the device ball mcaspi_aclkr, where i = 1 to 8
- internal AHCLKR is tied-off
- external AHCLKR is mapped on PRCM MCASPi_AHCLKR from PRCM for McASP1 and McASP2
- external AHCLKR is tied-off for McASP3 through McASP8

For more on McASP integration, see Section 24.6.2, *McASP Environment*, and Section 24.6.3, *McASP Integration*.

24.6.4.2.3 Frame-Sync Generator

There are two different modes for frame sync: burst and TDM. The McASP frame sync generator logic is illustrated in Figure 24-122. I/O buffers are not part of the McASP module, and are not shown in the figure.

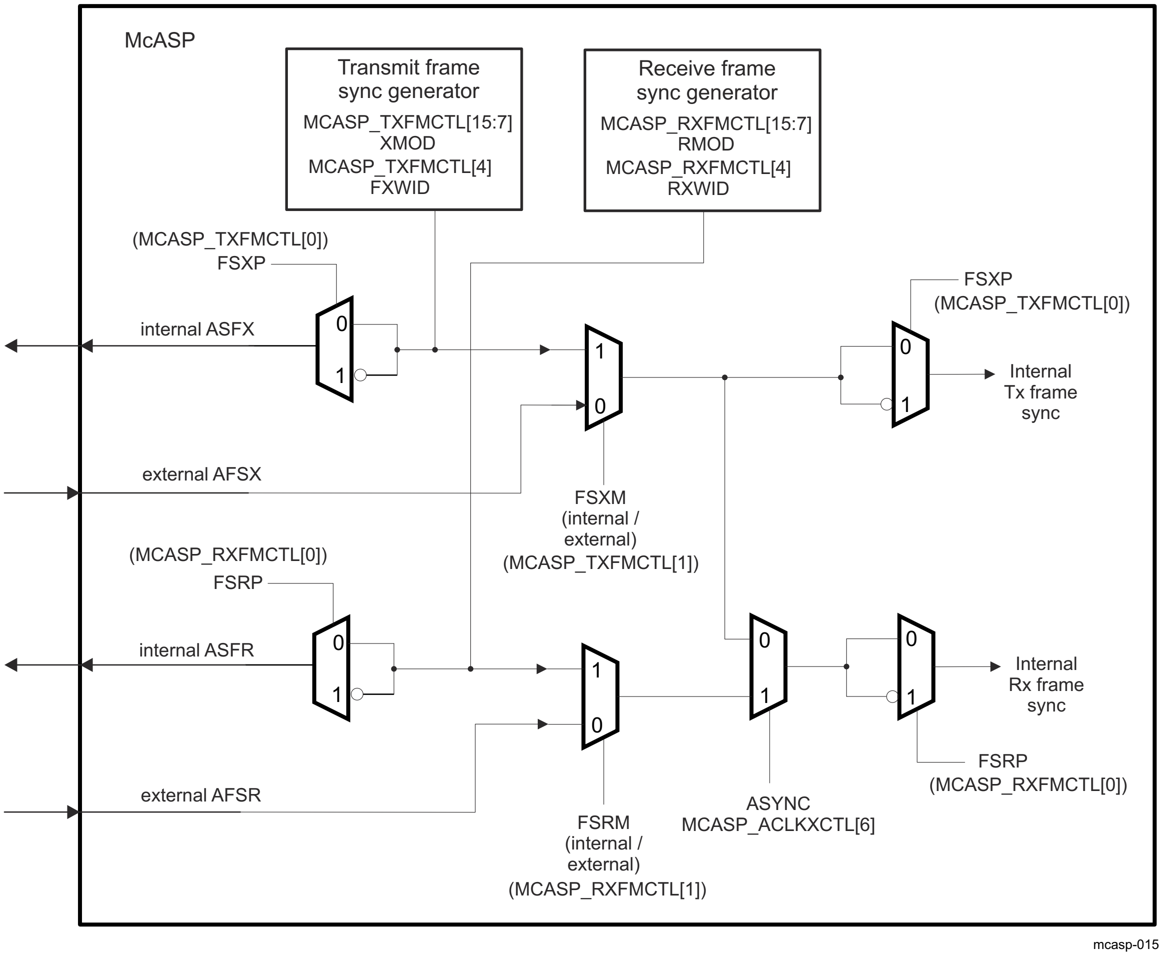


Figure 24-122. Frame Sync Generator Block Diagram

Note

For more on McASP integration, see Section 24.6.2, *McASP Environment*, and Section 24.6.3, *McASP Integration*.

For the transmit logic, following frame-sync generator configurations can be selected:

- Internally/externally generated frame-sync via configuring bit MCASP_TXFMCTL[1] FSXM
- Frame-sync polarity: Rising edge or falling edge via configuring bit MCASP_TXFMCTL[0] FSXP
- Frame-sync width: "single bit" or "single word" via configuring bit MCASP_TXFMCTL[4] FXWID
- Frame sync mode - the appropriate frame sync generation pattern for the selected transfer mode is defined in the bitfield MCASP_TXFMCTL[15:7] XMOD, as follows:
  - For DIT mode (384 slots) - MCASP_TXFMCTL[15:7] XMOD = 0x180
  - For I2S mode (2 TDM slots) - MCASP_TXFMCTL[15:7] XMOD = 0x2
  - For TDM mode (from 3 to 32 TDM slots) - MCASP_TXFMCTL[15:7] XMOD set in range 0x3 - 0x20
- Bit delay: 0, 1, or 2 cycles before the first data bit. This delay is defined in MCASP_TXFMT[17:16] XDATDLY

For the receive logic, following frame-sync generator configurations can be selected:

- Internally/externally generated frame-sync via configuring bit MCASP_RXFMCTL[1] FSRM
- Frame-sync polarity: Rising edge or falling edge via configuring bit MCASP_RXFMCTL[0] FSRP
- Frame-sync width: "single bit" or "single word" via configuring bit MCASP_RXFMCTL[4] FRWID

- Frame sync mode - the appropriate frame sync generation pattern for the selected transfer mode is defined in the bitfield MCASP_RXFMCTL[15:7] RMOD, as follows:
  – For I2S mode (2 TDM slots) - MCASP_RXFMCTL[15:7] RMOD = 0x2
  – For TDM mode (from 3 to 32 TDM slots) - MCASP_RXFMCTL[15:7] RMOD set in range 0x3 - 0x20
  – For the special 384-slot TDM mode - MCASP_RXFMCTL[15:7] RMOD=0x180
- Bit delay: 0, 1, or 2 cycles before the first data bit. This delay is defined in MCASP_RXFMT[17:16] RDATDLY
- Selecting the source (AFSX or AFSR) of receiver internal frame synchronization. This is done in the same bit - MCASP_ACLKXCTL[6] ASYNC, used to define the receiver internal clock source. For more details, refer to Section 24.6.4.2.4.

Regardless of the AFSX/AFSR being internally generated or externally sourced, the polarity of AFSX/AFSR is determined by FSXP/FSRP, respectively, to be either rising or falling edge. If FSXP/FSRP = 0, the frame sync polarity is rising edge. If FSXP/FSRP = 1, the frame sync polarity is falling edge.

Note

Certain restrictions apply to the receive and transmit logic settings, when MCASP_ACLKXCTL[6] ASYNC is set to 0b0. They are described in Section 24.6.4.2.4.

24.6.4.2.4 Synchronous and Asynchronous Transmit and Receive Operations

Synchronous Transmit and Receive Operations -

When MCASP_ACLKXCTL[6] ASYNC is written to 0b0, the transmit and receive sections operate synchronously to the transmit section clock and transmit frame sync signals.

Though Rx section may have a different data format, it has to be configured to have the same slot size than the transmit section one. As shown on the Figure 24-121, with the ASYNC bit set to 0b0, the RCLK becomes an inverted version of the transmit clock generator XCLK output.

When MCASP_ACLKXCTL[6] ASYNC = 0b0, both Rx and Tx sections use the same clock and frame sync signals. For this reason, they must be aligned on the following settings:

- MCASP_TXDITCTL[0] DITEN = 0 (that is, transmission in TDM mode is enabled)
- The total number of bits per frame must be the same (that is, RSSZ * RMOD product value must equal XSSZ * XMOD product value)
- The settings in MCASP_ACLKRCTL are NOT considered
- FSXM must match FSRM
- FXWID must match FRWID

For all other settings, the transmit and receive sections may be programmed independently.

Asynchronous Transmit and Receive Operations -

When MCASP_ACLKXCTL[6] ASYNC = 0b1, Tx and Rx operate independently from each other with separate clock and frame sync signals.

Note

Synchronous transmit and receive operations are allowed only in the McASP TDM (I2S) mode (i.e. when MCASP_TXDITCTL[0] DITEN=0b0).

24.6.4.3 Serializers

The McASP serializers shift serial data in (Rx) and out (Tx) of the McASP. A given serializer n consists of a shift register (XRSRn) with a single-entry data buffer XRBUFn used either for transmitting (write accessible in register MCASP_TXBUFn) or for receiving (read accessible in register MCASP_RXBUFn) data. In addition, each serializer has a dedicated control register (MCASP_XRSRCTLn) and a serial bidirectional data pin - AXRn. The register MCASP_XRSRCTLn allows n-th serializer to be configured as a transmitter, receiver, or as inactive.

There are transmit and receive data formatting units to support data alignment options of the McASP which are shared between all Tx and Rx serializers, respectively.

A given serializer XRSRn shifter configured as a receiver in MCASP_XRSRCTLn, shifts in data through McASP corresponding device level bidirectional data pad AXRn. A given serializer XRSRn shifter configured as a transmitter in MCASP_XRSRCTLn, shifts out data on McASP corresponding device level bidirectional data pad AXRn.

The serializer is clocked from the transmit section clock (ACLKX signal) if configured to transmit or clocked from the receive section clock (ACLKR signal) if configured to receive. A serializer configured to transmit and receive operates in lockstep, which means that for McASP there are at most a couple of zones, one for transmit and one for receive.

Figure 24-123 is the serializer block diagram.

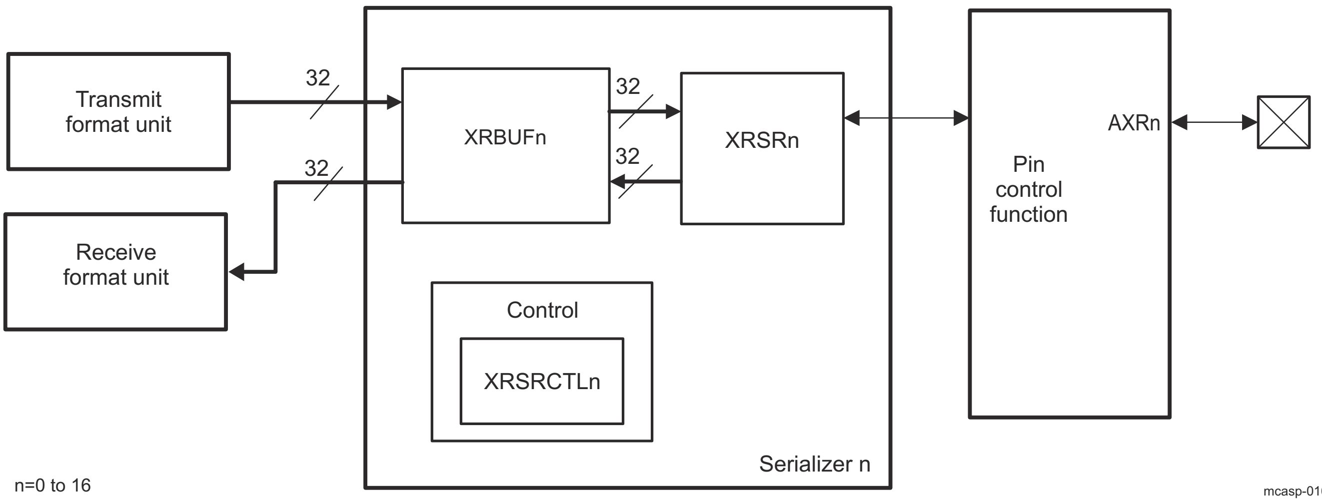


Figure 24-123. Individual Serializer and Connections Within McASP

Transmission on the n-th serializer is performed as follows:

The McASP is serviced by writing data into the register MCASP_TXBUFn, which is an alias of the serializer data buffer - XRBUFn for transmit function. The data automatically passes through the transmit format unit before reaching the XRBUFn register in the serializer. The data is then copied from the XRBUFn to XRSRn and shifted out from AXRn synchronously to the serial clock.

Reception from the n-th serializer is performed as follows:

The data is shifted into the McASP XRSRn serializer register through the AXRn pin, bit by bit. Once the entire slot becomes available within the XRSRn shift register, the data is copied into the serializer data buffer - XRBUFn,and can be accessed in the MCASP_RXBUFn register, which is an alias of the serializer data buffer - XRBUFn for receive function. When software reads the data from this register, the McASP passes the data through the receive format unit, hence it returns the formatted data.

Serializer controls:

A serializer n is configured as inactive via setting bitfield MCASP_XRSRCTLn[1:0] SRMOD to 0x0.

For a transmitting serializer, the MCASP_XRSRCTLn[3:2] DISMOD bitfield, defines the AXRn pin output state, during inactive slots (HIGH, LOW or Hi-Z).

Transmit function for the n-th serializer is selected via setting bitfield MCASP_XRSRCTLn[1:0] SRMOD to 0x1.

Receive function for the n-th serializer is selected via setting bitfield MCASP_XRSRCTLn[1:0] SRMOD to 0x2.

In the DIT-transmission mode (that is S/PDIF format data transmission): in addition to the data, the serializer shifts out other DIT-specific information accordingly (preamble, user data, etc.). For more information, see Section 24.6.2.2.5

24.6.4.4 Format Units

The McASP has one transmit data formatting unit and one receive data formatting unit, shared between the device McASP serializers. These units automatically remap the data bits within the transmitted or received words between a natural format for the device processors (for example, a Q31 representation) and the required format for the external serial device (for example I2S format). During the remapping process, the format unit can also mask off certain bits.

Since all transmitters share the same data formatting unit, the McASP only supports one transmit format at a time. For example, the McASP does NOT transmit in "I2S format" on serializer 0, while transmitting "Left Justified" on serializer 1. Likewise, the receiver section of the McASP only supports one data format at a time, and this format applies to all receiving serializers.

Note

The McASP can transmit in one format while receiving in a completely different format.

The bit mask and pad stage of each of Tx and Rx format units includes a full 32-bit mask register, allowing selected individual bits to either pass through the stage unchanged, or be masked off. The bit mask and pad then pad the value of the masked off bits by inserting either a 0, a 1, or one of the original 32 bits as the pad value. The last option allows for sign-extension when the sign bit is selected to pad the remaining bits. The rotate right stage performs bitwise rotation by a multiple of 4 bits (between 0 and 28 bits), programmable by the MCASP_RXFMT/MCASP_TXFMT register. Note that this is a rotation process, not a shifting process, so bit 0 gets shifted back into bit 31 during the rotation. The bit order - reversal stage either passes all 32 bits directly through, or swaps them. This allows for either MSB or LSB first data formats. If bit order reversal is not enabled, then the McASP will naturally transmit and receive in an LSB first order. Finally, note that the RDATDLY/XDATDLY bits in the MCASP_RXFMT/MCASP_TXFMT also determine the data format. For example, the difference between I2S format and left-justified is determined by the delay between the frame sync edge and the first data bit of a given time slot. For I2S format, RDATDLY/XDATDLY should be set to a 1-bit delay, whereas for left-justified format, it should be set to a 0-bit delay. The combination of all the options in MCASP_RXFMT/ MCASP_TXFMT means that the McASP supports a wide variety of data formats, both on the serial data lines, and in the device CPU data representation.

24.6.4.4.1 Transmit Format Unit

The McASP transmit formatting unit consists of three stages :

- Bit mask (masks off bits)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB-first or LSB-first)

Figure 24-124 shows the transmit formatting unit.

The McASP transmitter supports serial formats of:

- Slot (or Time slot) size = 8, 12, 16, 20, 24, 28, 32 bits
- Word size ≤ Slot size
- Alignment: when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB
  - LSB: least-significant bit of word is shifted out last, last bit is MSB

Hardware support for these serial formats comes from the programmable options in the bitstream format register - MCASP_TXFMT:

- XRVRS: bit reverse (1) or no bit reverse (0)
- XROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits
- XSSZ: transmit slot size of 8, 12, 16, 20, 24, 28, or 32 bits

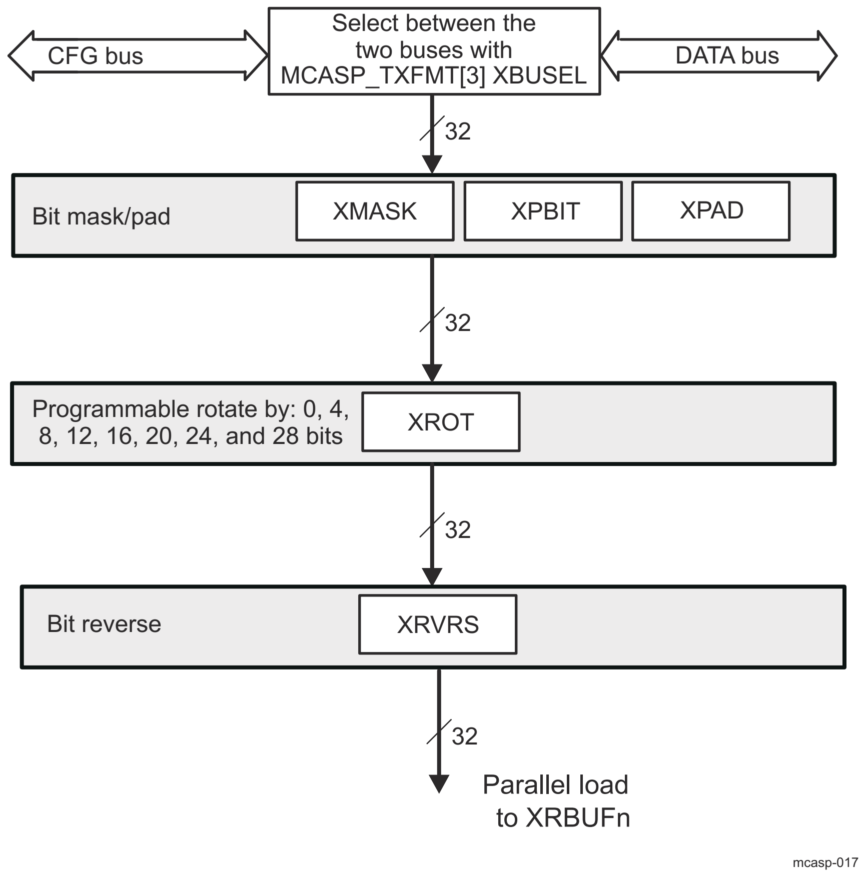


Figure 24-124. Transmit Format Unit

As shown in Figure 24-124, the data to the transmit format unit can come from the configuration port (CFG) or the data port (DATA). The selection is made through the MCASP_TXFMT[3] XBUSEL bit. According to port type selected, data transfer has different behaviour. For more details, refer to the Section 24.6.4.10.1.3, *Transfers Through the DATA Port*, and Section 24.6.4.10.1.4, *Transfers Through the Configuration (CFG) Bus*.

In the transmit format unit (TFU), the input data bits are first masked-off with the MCASP_TXMASK[31:0] XMASK contents. The masked data is then right-rotated to MCASP_TXFMT[2:0] XROT positions, to produce the output word for a TDM- or DIT- transmission.

The bit mask stage includes a full 32-bit mask register, allowing selected individual bits to pass through the stage unchanged or be masked off.

24.6.4.4.1.1 TDM Mode Transmission Data Alignment Settings

The TDM-mode transmission settings are relevant for I2S-protocol and protocols using more than 2 TDM-slots.

XSSZ should always be programmed to match the slot size of the serial stream.

Note

Note that, TDM word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the XROT field.

The Table 24-331 show the XRVRS and XROT fields for each serial format and for both integer and Q31 fractional internal representations.

The Table 24-331 assumes that all slot size (SLOT in Table 24-331) and word size (WORD in Table 24-331) options are multiples of 4, since the transmit rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The transmit bit mask/pad unit operates on data as an initial step of the transmit format unit, and the data is aligned in the same representation as it is written to the transmitter by the MPU or DSP (typically Q31 or integer).

Table 24-331. McASP TFU TDM Mode Settings

Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	MCASP_TXFMT bits	
			XROT(1)	XRVRS
MSB first(2)	Left aligned	Q31 fraction	0	1
MSB first	Right aligned	Q31 fraction	SLOT - WORD	1
LSB first	Left aligned	Q31 fraction	32 - WORD	0
LSB first	Right aligned	Q31 fraction	32 - SLOT	0
MSB first(2)	Left aligned	Integer	WORD	1
MSB first	Right aligned	Integer	SLOT	1
LSB first	Left aligned	Integer	0	0
LSB first	Right aligned	Integer	(32 - (SLOT - WORD)) % 32	0

(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator
(2) To transmit in I2S format, select MSB first, left aligned, and also select XDATDLY = 01 (1 bit delay)

24.6.4.4.1.2 DIT Mode Transmission Data Alignment Settings

In case of a DIT-mode (S/PDIF protocol) transmission, while left-aligned Q31 data should be right-rotated to a multiple by 4 positions, no right-rotation is required for a right-aligned Q31 data. Because this is a rotation process, not a shifting process, bit 0 gets shifted back into bit 31 during the process.

The MCASP_TXFMT[17:16] XDATDLY bit field must be set to a 0-bit delay (0x0 value).

For left-aligned Q31 data, the following transmit format unit settings process the data into right-aligned data, ready for transmission:

- MCASP_TXFMT[2:0] XROT =
  - 0x2 (rotate right by 8 bits) - for a 24-bit output audio data
  - 0x3 (rotate right by 12 bits) - for a 20-bit output audio data
  - 0x4 (rotate right by 16 bits) - for a 16-bit output audio data
- MCASP_TXFMT[15] XRVRS = 0x0 – Bit reversal is not enabled; the McASP naturally transmits and receives in a LSB-first order.
- MCASP_TXMASK[32] XMASK = 0xFFFFFF00 – 0xFFFF0000
- MCASP_TXFMT[14:13] XPAD = 0x0 (Pad extra bits with 0s.)

For right-aligned data, the following transmit format unit settings process the data into right-aligned audio data ready for transmission:

- MCASP_TXFMT[2:0] XROT = 0x0 (rotate right by 0 bits regardless of the audio word length)
- MCASP_TXFMT[15] XRVRS = 0x0 – Bit reversal is not enabled; the McASP naturally transmits and receives in a LSB-first order.
- MCASP_TXMASK[32] XMASK = 0x00FFFFFF – 0x0000FFFF
- MCASP_TXFMT[14:13] XPAD = 0x0 (Pad extra bits with 0s.)

The example settings provided in Table 24-332 should be applied to McASP in cases of DIT-transmitting a Q31 data as a 24-bit, 20-bit and 16-bit left- or right- aligned audio word, respectively. Note that the listed settings let the McASP TFU preserve the most significant bits and cut only the LSBs of the original Q31 CPU data:

Table 24-332. McASP TFU DIT-Mode Example Settings

Output Audio Word Alignment	Audio Word Length	Right-rotation (multiple of 4-bit positions)	XMASK	XROT
LEFT	16	16	0xFFFF0000	0x4
LEFT	20	12	0xFFFFF000	0x3
LEFT	24	8	0xFFFFFF00	0x2
RIGHT	16	0	0x0000FFFF	0x0
RIGHT	20	0	0x000FFFFF	0x0
RIGHT	24	0	0x00FFFFFF	0x0

Assuming that a Q31 data word 0xFA5AFxxx (where x-marked nibbles of the data are applied as padding bits of the word) is generated by MPU on the McASP CFG (peripheral) port. To transmit a left-aligned 20-bit version of same word, preserving the MSBs, according to the Table 24-332, the user must set XMASK=0xFFFFF000, and to select a right-rotation to 12 positions (XROT=0x3).

- After applying 0-s (XPAD=0) as masking-off bits at the first TFU stage, word is transformed to the word 0xFA5AF000.
- After a rotation by 12 positions to the right is performed in TFU, the 20-bit output word obtained is: 0x000FA5AF.Thus the word gets ready for transmission being mapped with its LS-bit as bit 8 and its MS-bit as bit 27 within a S/PDIF bitstream. This word is shifted in a LSB-to-MSB order (XRVRS = 0x0) out of the XRSR register during a DIT-transmission.

Assuming that a right-aligned Q31 data word - 0x yyyyE4B4 is generated by software on the McASP CFG (peripheral) port (with the presumption that y-marked nibbles of the input data are applied as padding bits). To transmit a right-aligned 16-bit version of same word, preserving the MSBs, according to the table McASP TFU Example Settings, user is supposed to set XMASK=0x0000FFFF, and to select right-rotation to 0 positions (XROT=0x0).

- After masking-off with 0s at first TFU stage, word is transformed to 0x0000E4B4.
- Since no rotation is applied, the 16-bit output word obtained is actually the one obtained in the masking stage – 0x0000E4B4.

The above examples use internal representation in integer and Q31 notation, but other fractional notations are also possible.

24.6.4.4.2 Receive Format Unit

The McASP receive formatting unit consists of three stages:

- Bit mask (masks off bits)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB first or LSB first)

Figure 24-125 shows the receive format unit (RFU).

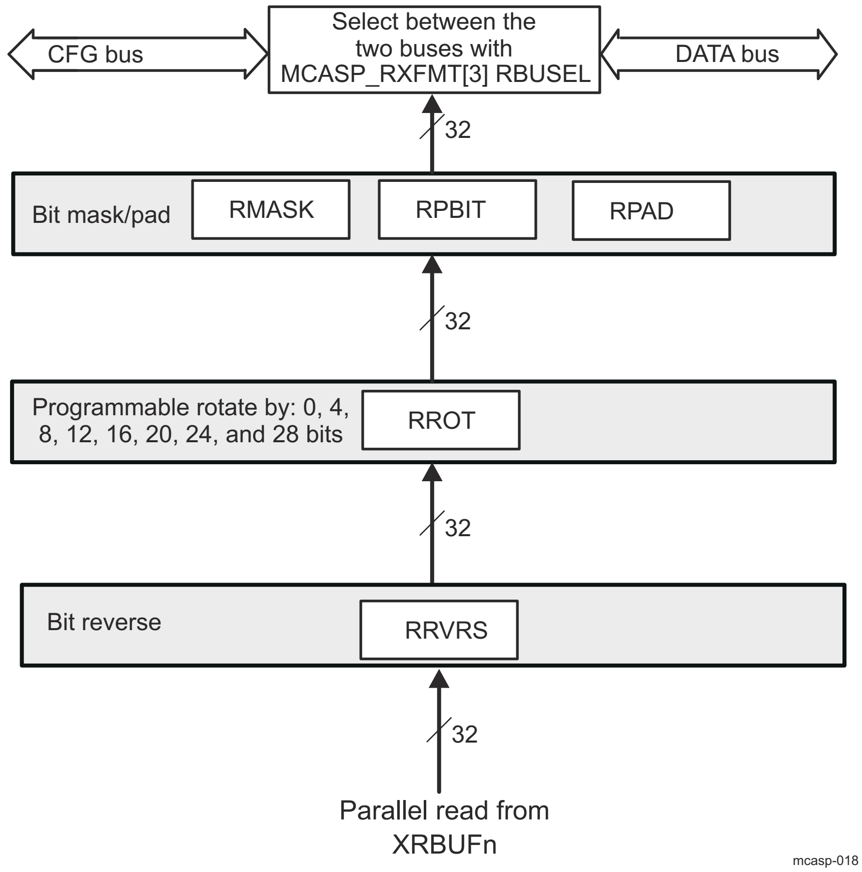


Figure 24-125. Receive Format Unit

The McASP receiver supports serial formats of:

- Slot or time slot size = 8, 12, 16, 20, 24, 28, 32 bits
- Word size ≤ Slot size
- Alignment when more bits are available per slot than bits per word within the slot, then:
  - Left aligned = word shifted first, remaining bits are pad
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB
  - LSB: least-significant bit of word is shifted out last, last bit is MSB

Hardware support for these serial formats comes from the programmable options in the receive bitstream format register - MCASP_RXFMT:

- RRVRS: bit reverse (1) or no bit reverse (0)
- RROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits
- RSSZ: receive slot size of 8, 12, 16, 20, 24, 28, or 32 bits

As shown on Figure 24-125, the data processed in the RFU can be output to host CPU through the configuration port (CFG) or the data port (DATA). The selection is made through the MCASP_RXFMT[3] RBUSEL bit. According to port type selected, data transfer has different behaviour. For more details, refer to the Section 24.6.4.10.1.3, *Transfers Through the DATA Port*, and Section 24.6.4.10.1.4, *Transfers Through the Configuration (CFG) Bus*.

24.6.4.4.2.1 TDM Mode Reception Data Alignment Settings

RSSZ should always be programmed to match the slot size of the serial stream.

Note

Note that the word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the RROT field.

Table 24-333 shows the RRVRS and RROT fields for each serial format and for both integer and Q31 fractional internal representations.

Table 24-333. McASP RFU Settings

Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	MCASP_RXFMT bits	
			RROT[1]	RRVRS
MSB first[2]	Left aligned	Q31 fraction	SLOT	1
MSB first	Right aligned	Q31 fraction	WORD	1
LSB first	Left aligned	Q31 fraction	(32 - (SLOT - WORD)) % 32	0
LSB first	Right aligned	Q31 fraction	0	0
MSB first[2]	Left aligned	Integer	SLOT - WORD	1
MSB first	Right aligned	Integer	0	1
LSB first	Left aligned	Integer	32 - SLOT	0
LSB first	Right aligned	Integer	32 - WORD	0

(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator
(2) To receive in I2S format, select MSB first, left aligned, and also select RDATDLY = 01 (1 bit delay)

The Table 24-333 assumes that all slot size and word size options are multiples of 4; since the receive rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be received in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1. The receive bit mask/pad unit operates on data as the final step of the receive format unit (see Figure 24-125), and the data is aligned in the same representation as it is read from the receiver (typically Q31 or integer).

24.6.4.5 State-Machines

The receive and transmit sections have independent state machines.

Each state-machine controls the interactions between the various units in the McASP Rx and Tx sections, respectively. In addition, each state-machine keeps track of error conditions and serial port status. No serial transfers can occur until the RX/TX state-machine is released from reset.

The transmit state-machine is controlled by the transmit bitstream format register (MCASP_TXFMT) and it reports the McASP status and error conditions in the transmitter status register (MCASP_TXSTAT).

Similarly, the receive state-machine is controlled by the receive bitstream format register (MCASP_RXFMT) and it reports the McASP status and error conditions in the receiver status register (MCASP_RXSTAT).

24.6.4.6 TDM Sequencers

There are separate TDM sequencers for the transmit section and the receive section. Each TDM sequencer keeps track of the slot count. In addition, the TDM sequencer checks the bits of MCASP_RXTDM/ MCASP_TXTDM and determines if the McASP should receive/transmit in that time slot.

There are two possibilities for a slot: The McASP either performs Rx/Tx operations during the time slot (transmit/ receive bit is active), or the McASP skips Rx/Tx operations during the time slot (transmit/receive bit is inactive). In the latter case, no transfers between the XRBUF and XRSR registers in the serializer would occur during that time slot.

In addition, during time of inactive slots, the serializers programmed as transmitters place their data output pins - AXRn in a predetermined state - logic low, high, or high impedance (tri-stated) as programmed in each serializer control register MCASP_XRSRCTLn[3:2] DISMOD. Refer also to Section 24.6.4.9.2.1, *TDM Time Slots Generation and Processing*, for details on how DMA event or interrupt generations are handled during inactive time slots in TDM mode.

In case of a DIT-transmission (S/PDIF transfers): the time division multiplexing (TDM) sequencer is used to count the 384 subframes (slots) in the DIT block. If currently transmitting slot 1, slot 2 (next value of the TDM slot counter) should be used during the encode phase to select the appropriate C, V, and U bit, because the data encoded and written to a MCASP_TXBUFn register during the current time slot (slot 1) is actually shifted out on the next time slot.

The transmit TDM sequencer is controlled by the MCASP_TXTDM register and reports the current transmit slot to the MCASP_TXTDMSLOT[9:0] XSLOTCNT bit field.

24.6.4.7 McASP Software Reset

The McASP can be put into reset through the global transmit and receive control register (MCASP_GBLCTL). A valid serial clock must be supplied to the McASP to assert the software reset bits in the MCASP_GBLCTL register.

24.6.4.8 McASP Power Management

Table 24-334 describes power-management features available to the McASP.

Table 24-334. Local Power-Management Features

Feature	Registers	Description
Slave idle modes	PWRIDLESYSCONFIG[1:0] IDLE_MODE	Force-idle, no-idle, and smart-idle modes are available.

> **CAUTION**
>
> No wakeup schema is supported for the McASP. To ensure a correct behavior after enabling McASP at device PRCM level, the user software is strongly recommended to choose *No Idle* mode, setting PWRIDLESYSCONFIG[1:0] IDLE_MODE to 0x1. Before disabling McASP at device PRCM level, user software is strongly recommended to choose a *Smart-Idle* mode, setting PWRIDLESYSCONFIG[1:0] IDLE_MODE to 0x2.

24.6.4.9 Transfer Modes

24.6.4.9.1 Burst Transfer Mode

The McASP supports a burst transfer mode, which is useful for nonaudio data such as passing control information between two processors. Burst transfer mode uses a synchronous serial format similar to the TDM mode. The frame sync generation is not periodic or time-driven as in TDM mode, but data driven, and the frame sync is generated for each data word transferred.

When operating in burst frame sync mode (see Figure 24-126), as specified for transmit (MCASP_TXFMCTL[15:7] = 0) and receive (MCASP_RXFMCTL[15:7] RMOD = 0), one slot is shifted for each active edge of the frame sync signal that is recognized. Additional clocks after the slot and before the next frame sync edge are ignored.

In burst frame sync mode, the frame sync delay may be specified as 0, 1, or 2 serial clock cycles. This is the delay between the frame sync active edge and the start of the slot. The frame sync signal lasts for a single bit clock duration (MCASP_RXFMCTL[4] FRWID = 0, MCASP_TXFMCTL[4] FXWID = 0).

For transmit, when generating the transmit frame sync internally, the frame sync begins when the previous transmission has completed and when all the XBUFn (for every serializer set to operate as a transmitter) has been updated with new data.

For receive, when generating the receive frame sync internally, frame sync begins when the previous transmission has completed and when all the RBUFn (for every serializer set to operate as a receiver) has been read.

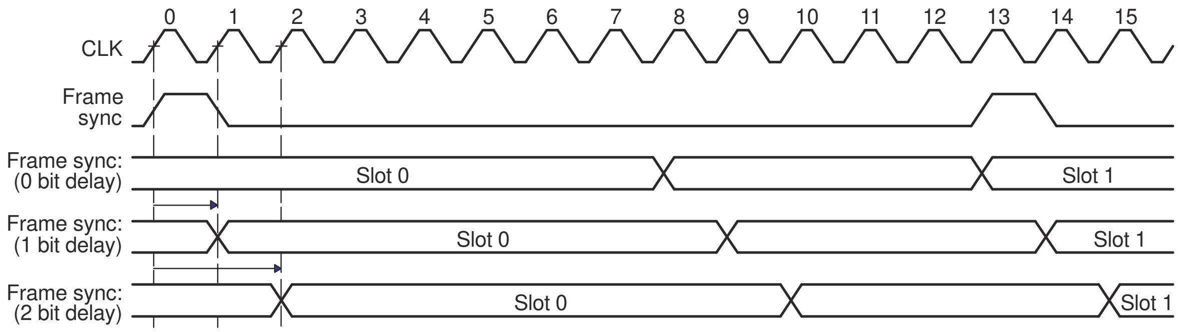


Figure 24-126. Burst Frame Sync Mode

The control registers must be configured as follows for the burst transfer mode. The burst mode specific bit fields are in bold face:

- MCASP_PFUNC: The clock, frame, data pins must be configured for McASP function.
- MCASP_PDIR: The clock, frame, data pins must be configured to the direction desired.
- MCASP_PDOUT, MCASP_PDIN, MCASP_PDSET, MCASP_PDCLR: Not applicable. Leave at default.
- MCASP_GBLCTL: Follow the initialization sequence in Section 24.6.5.1.2, *McASP Global Initialization*, to configure this register.
- MCASP_AMUTE: Not applicable. Leave at default.
- MCASP_LBCTL: If loopback mode is desired, configure this register according to Section 24.6.4.14 Loopback Modes, otherwise leave this register at default.
- MCASP_TXDITCTL: DITEN must be left at default 0 to select non-DIT mode. Leave the register at default.
- MCASP_RXMASK/MCASP_TXMASK: Mask desired bits according to Section 24.6.4.4, *Format Units*.
- MCASP_RXFMT/MCASP_TXFMT: Program all fields according to data format desired. See Section 24.6.4.4, *Format Units*.
- MCASP_RXFMT/MCASP_TXFMT: Clear RMOD/XMOD bits to 0 to indicate burst mode. Clear FRWID/ FXWID bits to 0 for single bit frame sync duration. Configure other fields as desired.
- MCASP_ACLKRCTL/MCASP_ACLKXCTL: Program all fields according to bit clock desired. See Section 24.6.4.2, *McASP Clock and Frame-Sync Configurations*.
- MCASP_AHCLKRCTL/MCASP_AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 24.6.4.2, *McASP Clock and Frame-Sync Configurations*.
- MCASP_RXTDM/MCASP_TXTDM: Program RTDMS0/XTDMS0 to 1 to indicate one active slot only. Leave other fields at default.
- MCASP_EVTCTLR/MCASP_EVTCTLX: Program all fields according to interrupts desired.
- MCASP_RXCLKCHK/MCASP_TXCLKCHK: Not applicable. Leave at default.
- MCASP_XRSRCTLn: Program SRMOD to inactive/transmitter/receiver as desired. DISMOD is not applicable and should be left at default.
- MCASP_DITCSRAi, MCASP_DITCSRBi, MCASP_DITUDRAi, MCASP_DITUDRBi: Not applicable. Leave at default.

24.6.4.9.2 Time-Division Multiplexed (TDM) Transfer Mode

The McASP time-division multiplexed (TDM) transfer mode supports the TDM format discussed in Section 24.6.2.2.3.

Transmitting data in the TDM transfer mode requires a minimum set of pins:

- ACLKX - transmit bit clock
- AFSX - transmit frame sync (or commonly called left/right clock)
- One or more serial data pins, AXRn, whose serializers are configured to transmit

For more details on McASP transmitting serializers clock and frame sync options, refer to the Section 24.6.4.2.1, *Transmit Clock*, and Section 24.6.4.2.3, *Frame-Sync Generator* .

Similarly, to receive data in the TDM transfer mode requires a minimum set of pins:
- ACLKR - receive bit clock
- AFSR - receive frame sync (or commonly called left/right clock)
- One or more serial data pins, AXRn, whose serializers are configured to receive

For more details on McASP receiving serializers clock and frame sync options, refer to Section 24.6.4.2.2, *Receive Clock*, and Section 24.6.4.2.3, *Frame-Sync Generator* .

The control registers must be configured as follows for the TDM mode. The TDM mode specific bit fields are highlighted in bold:
- MCASP_PFUNC: The clock, frame, data pins must be configured for McASP function.
- MCASP_PDIR: The clock, frame, data pins must be configured to the direction desired.
- MCASP_PDOUT, MCASP_PDIN, MCASP_PDSET, MCASP_PDCLR: Not applicable. Leave at default.
- MCASP_GBLCTL: Follow the initialization sequence is described in the Section 24.6.5.2, *Operational Modes Configuration*.
- MCASP_AMUTE: Leave this register at default state.
- MCASP_LBCTL: If loopback mode is desired, configure this register according to Section 24.6.4.14, otherwise leave this register at default.
- MCASP_TXDITCTL: DITEN must be left at default 0 to select TDM mode (transmitters only).
- MCASP_RXMASK/MCASP_TXMASK: Mask desired bits according to Section 24.6.4.4, *Format Units*.
- MCASP_RXFMT/MCASP_TXFMT: Program all fields according to data format desired. See the Section 24.6.4.4, *Format Units*.
- MCASP_RXFMCTL/MCASP_TXFMCTL: Set RMOD/XMOD bits to (0x2 - 0x20) for Rx/Tx (2- 32 slots) TDM mode. In addition, set RMOD to 0x180 if 384-slot TDM stream has to be received by McASP. Configure other fields as desired.
- MCASP_ACLKRCTL/MCASP_ACLKXCTL: Program all fields according to bit clock desired. For more inforamtion, refer to Section 24.6.4.2.
- MCASP_AHCLKRCTL/MCASP_AHCLKXCTL: Program all fields according to high-frequency clock desired. For more details, refer to Section 24.6.4.2.
- MCASP_RXTDM/MCASP_TXTDM: Program all fields according to the time slot characteristics desired.
- MCASP_EVTCTLX: Program all fields according to transmit interrupts desired.
- MCASP_RXCLKCHK/MCASP_TXCLKCHK: Program all fields according to clock checking desired.
- MCASP_XRSRCTLn: Program all fields according to serializer operation desired.

Note

The MCASP_DITCSRAi, MCASP_DITCSRBi, MCASP_DITUDRAi, MCASP_DITUDRBi (i=0 to 5) settings are NOT applicable in TDM transfer modes. They have to be kept at their default values.

24.6.4.9.2.1 TDM Time Slots Generation and Processing

TDM mode on the McASP can extend to support multiprocessor applications, with up to 32 time slots per frame. For each of the time slots, the McASP may be configured to participate or to be inactive by configuring MCASP_TXTDM and/or MCASP_RXTDM registers.

The TDM sequencer (separate ones for transmit and receive) functions in this mode. The TDM sequencer counts the slots beginning with the frame sync. For each slot, the TDM sequencer checks the respective bit in either MCASP_TXTDM or MCASP_RXTDM to determine if the McASP transmits/receives in that time slot.

Note

If a MCASP_TXTDM/MCASP_RXTDM bit defines an active slot (number of slot matches the bit position), the McASP functions normally during that time slot; otherwise, the McASP is inactive during that time slot; no update to the buffer occurs, and no event is generated. McASP (transmit only) data pins are automatically set to a high-impedance state, 0, or 1 during that slot, as determined by bitfield MCASP_XRSRCTLn[3:2] DISMOD.

Figure 24-127 shows when the transmit DMA event - AXEVT is generated. See Section 24.6.4.10.1, *Data Ready Status and Event/Interrupt Generation* for details on data ready and the initialization period indication. The transmit DMA event for an active time slot (slot N) is generated during the previous time slot (slot N - 1), regardless of the previous time slot (slot N - 1) being active or inactive.

During an active transmit time slot (slot N), if the next time slot (slot N + 1) is configured to be active, the copy from XRBUFn to XRSRn generates the DMA event for time slot N + 1. If the next time slot (slot N + 1) is configured to be inactive, then the DMA event will be delayed to time slot M - 1. In this case, slot M is the next active time slot. The DMA event for time slot M is generated during the first bit time of slot M - 1.

The receive DMA event is generated after data is received in the buffer (looks back in time). If a time slot is disabled, then no data is copied to the buffer for that time slot and no DMA event is generated.

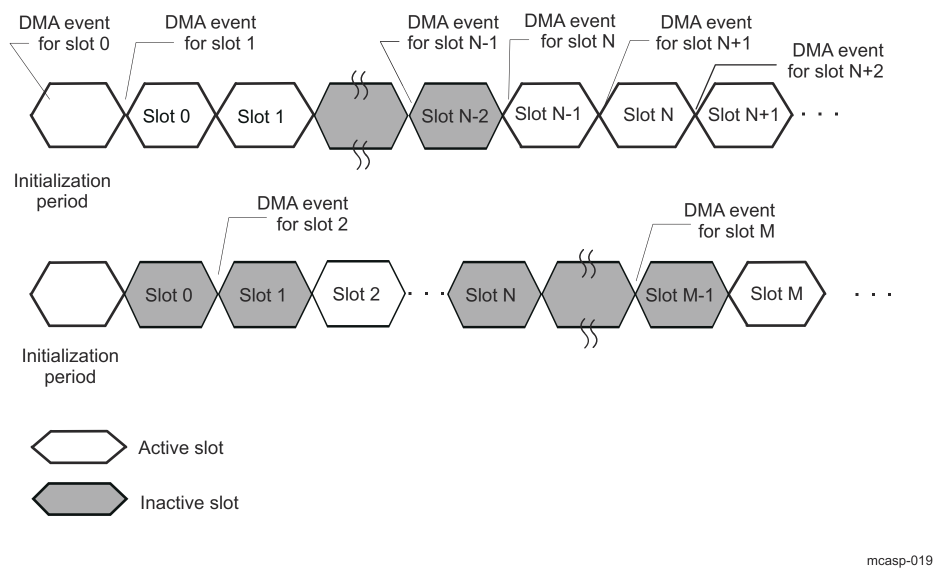


Figure 24-127. Transmit DMA Event (AXEVT) Generation in TDM Time Slots

24.6.4.9.2.2 Special 384-Slot TDM Mode for Connection to External DIR

The McASP receiver also supports a 384 time slot TDM mode (DIR mode), to support S/PDIF receiver ICs whose natural block (block corresponds to McASP frame) size is 384 samples. The receive TDM time slot register (MCASP_RXTDM) should be programmed to all 1s during reception of a DIR block. Other TDM functionalities (for example, inactive slots) are not supported (only the slot counter counts the 384 subframes in a block). To receive data in DIR mode, the following pins are typically needed:

- ACLKR - receive bit clock
- AFSR - receive frame sync (or commonly called left/right clock)
- In this mode, AFSR should be connected to a DIR which outputs a start of block signal, instead of LRCLK
- One or more serial data pins, AXRn, whose serializers have been configured to receive
- For this special DIR mode, the control registers can be configured just as for TDM mode, except set RMOD in MCASP_RXFMCTL to 384 (0x180) to receive 384 time slots

24.6.4.9.3 DIT Transfer Mode

The DIT transfer mode of the McASP also supports transmission of audio data in S/PDIF, AES-3, and IEC-60958 formats. These formats are designed to carry audio data between different systems through an optical or coaxial cable. The DIT mode applies only to a serializer configured as transmitter, not as receiver. For a description of the S/PDIF format, see Section 24.6.2.2.5, *S/PDIF Coding Format*.

24.6.4.9.3.1 Transmit DIT Encoding

When the McASP operates in DIT mode, the data transmitted is output as a biphase-mark encoded bitstream, with preamble, channel status, user data, validity, and parity automatically stuffed into the bitstream by the McASP. The McASP includes separate validity bits for even/odd subframes and two 384-bit RAM modules to hold channel status and user data bits.

Note

The transmit TDM time slot register (MCASP_TXTDM) should be programmed to all 1s during DIT mode. TDM functionality is not supported in DIT mode, except that the TDM slot counter counts the DIT subframes.

To transmit data in DIT mode, the following pins are typically required:

- AHCLKX – transmit high-frequency master clock (The internal clock source can be used instead.)
- One serial data pin (AXRn) of a serializer n configured to transmit.

For DIT Mode Transmission Data Alignment Settings see Section 24.6.4.4.1.2.

If the McASP is configured to transmit in the DIT mode on more than one serial data pin, the bit streams on all pins will be synchronized. In addition, although they will carry unique audio data, they will carry the same channel status, user data, and validity information.

The actual 24-bit audio data must always be in bit positions 23–0 after passing through the first three stages of the transmit format unit.

24.6.4.9.3.2 Transmit DIT Clock and Frame-Sync Generation

The DIT transmitter works only in the following configuration:

- In the transmit frame control register (MCASP_TXFMCTL):
  - Internally generated transmit frame sync, FSXM = 1
  - Rising-edge frame sync, FSXP = 0
  - Bit-width frame sync, FXWID = 0
  - 384-slot TDM, XMOD = 1 1000 0000b
- In the transmit clock control register (MCASP_ACLKXCTL), ASYNC = 1
- In the transmit bitstream format register (MCASP_TXFMT), XSSZ = 1111 (32-bit slot size)

All combinations of AHCLKX and ACLKX are supported.

The following summarizes the register configurations required for DIT mode. DIT mode-specific bit fields are in bold face:

- MCASP_PFUNC: The data pin - AXRn must be configured for McASP function. If AHCLKX is used, it must also be configured for McASP function. Other pins can be configured to function as GPIOs, if desired.
- MCASP_PDIR: The data pin must be configured as output. If internal clock source AUXCLK is used as the reference clock, it may be output as the AHCLKX device level signal by configuring AHCLKX pin as an output.
- MCASP_GBLCTL: Global initialization
- MCASP_AMUTE: Leave this register at default state.
- MCASP_TXDITCTL: The DITEN bit must be set to 0b1 to enable DIT mode. Configure other bits as desired.
- MCASP_TXMASK: Mask the desired bits, depending upon left-aligned or right-aligned internal data.

- MCASP_TXFMT: XDATDLY = 0. XRVRS = 0. XPAD = 0. XSSZ = Fh (32-bit slot). XBUSEL = configured as desired. The XROT bit is configured, as described in the Section 24.6.4.4.1.2 .
- MCASP_TXFMCTL: Configure the bits according to former discussions.
- MCASP_ACLKXCTL: ASYNC = 1. Program the CLKXDIV bits to obtain the bit clock rate desired. CLKXM = 1.
- MCASP_AHCLKXCTL: Program the HCLKXDIV bits to obtain the high-frequency bit clock rate desired.
- MCASP_TXTDM: Set to FFFF FFFFh for all active slots for DIT transfers.
- MCASP_EVTCTLX: Program all fields according to the interrupts desired.
- MCASP_TXCLKCHK: Program all fields according to the clock checking desired.
- MCASP_XRSRCTLn: Set SRMOD = 1 (transmitter) for the DIT pins.
- MCASP_DITCSRAi and MCASP_DITCSRBi: Program the channel status bits as desired.
- MCASP_DITUDRAi and MCASP_DITUDRBi: Program the user data bits as desired.

Note

In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to 2 serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for TDM (I2S) mode, due to the need to generate Biphase Mark Encoded Data - see Section 24.6.2.2.5.1).

24.6.4.9.3.3 DIT Channel Status and User Data Register Files

The channel status registers (MCASP_DITCSRAi and MCASP_DITCSRBi) and user data registers (MCASP_DITUDRAi and MCASP_DITUDRBi) are not double-buffered. Typically, programmers use one of the synchronizing interrupts, such as the last slot, to create an event at a safe time so the register may be updated. In addition, the software reads the transmit TDM slot counter to determine which word of the register is being used.

It is a software requirement to avoid writing to the word of user data and channel status that are being used to encode the current time slot; otherwise, it is undetermined whether old or new data is used to encode the bitstream.

The DIT subframe format is defined in Section 24.6.2.2.5.2, *S/PDIF Subframe Format*. The channel status information (C) and user data (U) are defined in the following DIT control registers:

- MCASP_DITCSRA0 to MCASP_DITCSRA5: The 192 bits in these six registers contain the channel status information for the left channel within each frame.
- MCASP_DITCSRB0 to MCASP_DITCSRB5: The 192 bits in these six registers contain the channel status information for the right channel within each frame.
- MCASP_DITUDRA0 to MCASP_DITUDRA5: The 192 bits in these six registers contain the user data information for the left channel within each frame.
- MCASP_DITUDRB0 to MCASP_DITUDRB5: The 192 bits in these six registers contain the user data information for the right channel within each frame.
- The S/PDIF block format is shown in Figure 24-117. There are 192 frames within a block (frame 0 to frame 191). There are two subframes within each frame (subframes 1 and 2 for the left and right channels, respectively).

The channel status and user data information sent on each subframe is summarized in Table 24-335.

Table 24-335. Channel Status and User Data for Each DIT Block

Frame	Subframe	Preamble	Channel Status Defined in:	User Data Defined in:
Defined by DITCSRA0, DITCSRB0, DITUDRA0, DITUDRB0				
0	1 (L)	B	DITCSRA0[0]	DITUDRA0[0]
0	2 (R)	W	DITCSRB0[0]	DITUDRB0[0]
1	1 (L)	M	DITCSRA0[1]	DITUDRA0[1]
1	2 (R)	W	DITCSRB0[1]	DITUDRB0[1]
2	1 (L)	M	DITCSRA0[2]	DITUDRA0[2]

Table 24-335. Channel Status and User Data for Each DIT Block (continued)

Frame	Subframe	Preamble	Channel Status Defined in:	User Data Defined in:
2	2 (R)	W	DITCSRB0[2]	DITUDRB0[2]
…	…	…	…	…
31	1 (L)	M	DITCSRA0[31]	DITUDRA0[31]
31	2 (R)	W	DITCSRB0[31]	DITUDRB0[31]
Defined by DITCSRA1, DITCSRB1, DITUDRA1, DITUDRB1				
32	1 (L)	M	DITCSRA1[0]	DITUDRA1[0]
32	2 (R)	W	DITCSRB1[0]	DITUDRB1[0]
…	…	…	…	…
63	1 (L)	M	DITCSRA1[31]	DITUDRA1[31]
63	2 (R)	W	DITCSRB1[31]	DITUDRB1[31]
Defined by DITCSRA2, DITCSRB2, DITUDRA2, DITUDRB2				
64	1 (L)	M	DITCSRA2[0]	DITUDRA2[0]
64	2 (R)	W	DITCSRB2[0]	DITUDRB2[0]
…	…	…	…	…
95	1 (L)	M	DITCSRA2[31]	DITUDRA2[31]
95	2 (R)	W	DITCSRB2[31]	DITUDRB2[31]
Defined by DITCSRA3, DITCSRB3, DITUDRA3, DITUDRB3				
96	1 (L)	M	DITCSRA3[0]	DITUDRA3[0]
96	2 (R)	W	DITCSRB3[0]	DITUDRB3[0]
…	…	…	…	…
127	1 (L)	M	DITCSRA3[31]	DITUDRA3[31]
127	2 (R)	W	DITCSRB3[31]	DITUDRB3[31]
Defined by DITCSRA4, DITCSRB4, DITUDRA4, DITUDRB4				
128	1 (L)	M	DITCSRA4[0]	DITUDRA4[0]
128	2 (R)	W	DITCSRB4[0]	DITUDRB4[0]
…	…	…	…	…
159	1 (L)	M	DITCSRA4[31]	DITUDRA4[31]
159	2 (R)	W	DITCSRB4[31]	DITUDRB4[31]
Defined by DITCSRA5, DITCSRB5, DITUDRA5, DITUDRB5				
160	1 (L)	M	DITCSRA5[0]	DITUDRA5[0]
160	2 (R)	W	DITCSRB5[0]	DITUDRB5[0]
…	…	…	…	…
191	1 (L)	M	DITCSRA5[31]	DITUDRA5[31]
191	2 (R)	W	DITCSRB5[31]	DITUDRB5[31]

24.6.4.10 Data Transmission and Reception

The McASP is serviced by writing data to the MCASP_TXBUFn registers for transmit operations, and by reading data from the MCASP_RXBUFn registers for receive operations. The McASP sets status flags and notifies the software whenever data is ready to be serviced. The Section 24.6.4.10.1, *Data Ready Status and Event/Interrupt Generation*, discusses data-ready status in details.

The McASP transmit/receive XRBUFn buffer can be accessed through one of the two peripheral ports of the device:

- DATA port: This port is dedicated to DMA initiated data transfers on the device for McASP transmit (Tx) purposes.
- Configuration bus (CFG): The configuration bus- CFG port is used for peripheral configuration control and receive/transmit data transfers initiated by the host CPU in the device.

Section 24.6.4.10.1.3, *Transfers Through the Data Port (DATA)*, and Section 24.6.4.10.1.4, *Transfers Through the Configuration Bus (CFG),* discuss how to perform transfers through the data port (DATA) and the configuration port (CFG), respectively.

A device CPU and DMA usages are discussed in Section 24.6.4.10.1.5, *Using the device CPUs for McASP Servicing*, and Section 24.6.4.10.1.6, *Using the DMA for McASP Servicing*, respectively.

McASP DATA port allows DMAs to access the McASP transmit buffer more efficiently on the L3_MAIN-interconnect or L4_PER2 interconect, using burst transfers. The physical addresses to access these registers are listed in Section 24.6.6.2.5.

24.6.4.10.1 Data Ready Status and Event/Interrupt Generation

24.6.4.10.1.1 Transmit Data Ready

The transmit data ready flag - XDATA in the MCASP_TXSTAT register reflects the data ready status of XRBUFn buffers for all of the active slot transmitting serializers. The XDATA flag is set whenever data is transferred from a transmitting serializer buffer - XRBUFn to its corresponding XRSRn shift register. Thus, the XDATA bit indicates the global event that some of the serializers data buffer - XRBUFn is emptied and ready to accept new data from the host (CPU or DMA). The transmit data ready event is individually indicated per serializer in its corresponding control register MCASP_XRSRCTLn[4] XRDY status bit. When this bit is set to 0b1, it notifies to host that this serializer Tx buffer must be serviced (written). When MCASP_TXBUFn is written to by the host, the MCASP_XRSRCTLn[4] XRDY is deasserted to 0b0. As XDATA global flag is an OR-event of all active serializers XRDY flags, it indicates to software the moment, when write service operation has to be initiated by the McASP host (XDATA=0b1). The XRDY flags have to be sequentially scanned by user software to determine which serializer MCASP_TXBUFn register has to be currently written. Once all requested MCASP_TXBUFn are written, the serializers control XRDY flags are cleared to 0b0. As a consequence, XDATA flag is deasserted to 0b0, to indicate to SW that write operation is completed for all serializers.

The global XDATA flag can be cleared when the MCASP_TXSTAT[5] XDATA bit is written to 0b1, or once MCASP_TXBUFn registers of all the serializers, that have previously raised their XRDY flags, are written with corresponding active slot data by the host.

Whenever XDATA is set, the AXEVT event is automatically generated on MCASPi_DREQ_TX line (if enabled in the MCASP_XEVTCTL register) to notify the DMA of the MCASP_TXBUFn empty status. An interrupt - MCASPi_IRQ_AXEVT can be also generated if the XDATA interrupt is enabled in the MCASP_EVTCTLX register (for details, see Section 24.6.4.12.1, *Transmit Data Ready Interrupt*).

For DMA requests, the McASP does not require that MCASP_TXSTAT be read between DMA events. This means that, even if MCASP_TXSTAT already has the XDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

Because the serializer acts in lockstep, only one DMA event is generated to indicate that the transmit serializer is ready to be written to with new data.

Figure 24-128 shows the timing details of when AXEVT is generated at the McASP boundary. In this example, as soon as the last bit (A0) of word A is transmitted, the McASP sets the XDATA flag and generates an AXEVT event. However, it takes up to five McASP interface clocks (AXEVT latency) before AXEVT is active at the McASP boundary. Upon AXEVT, the CPU can begin servicing the McASP by writing word C into the MCASP_TXBUFn (service time). The CPU must write word C into the MCASP_TXBUFn within the setup time required by the McASP (setup time).

The maximum service time (see Figure 24-128) can be calculated as:

Service Time = Time Slot – AXEVT Latency –Setup Time

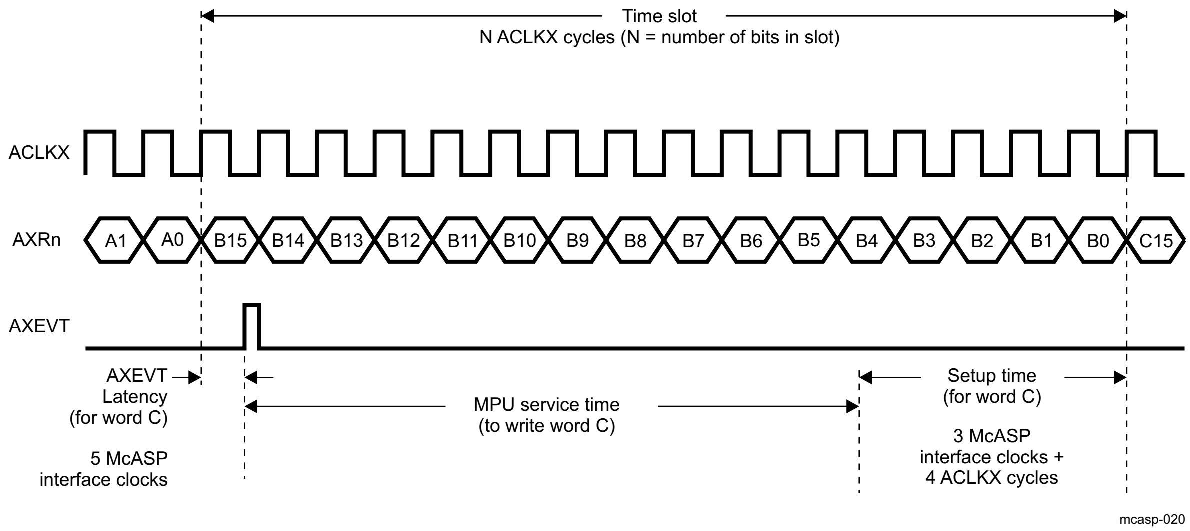


Figure 24-128. MPU Service Time Upon Transmit DMA Event (AXEVT)

24.6.4.10.1.2 Receive Data Ready

Similarly, the receive data ready flag - RDATA in the MCASP_RXSTAT register reflects the data ready status of XRBUFn buffers for all of the active slot receiving serializers. The RDATA flag is set whenever data is transferred from a receiving serializer shift register XRSRn to its corresponding XRBUFn data buffer. Thus, the RDATA bit indicates the global event that some of the receivers data buffer - RXBUFn already contains received data (i.e. a buffer is full) and is ready to transfer it to the host (MPU/DSP). The receive data ready event is individually indicated per serializer in its corresponding control register MCASP_XRSRCTLn [5] RRDY status bit. When this bit is set to 0b1, it notifies to host that this serializer Rx buffer must be serviced (read). When MCASP_RXBUFn is read from the host, the MCASP_XRSRCTLn [5] RRDY is deasserted to 0b0. As RDATA global flag is an OR-event of all active serializers RRDY flags, it indicates to software the moment, when read service operation has to be initiated by the McASP host (RDATA=0b1). The RRDY flags have to be sequentially scanned by user software to determine which serializer MCASP_RXBUFn register has to be currently read. Once all requested MCASP_RXBUFn are read, the serializers control RRDY flags are cleared to 0b0. As a consequence, RDATA flag is deasserted to 0b0, to indicate to SW that read operation is completed for all serializers.

The global RDATA flag can be cleared when the MCASP_RXSTAT[5] RDATA bit is written to 0b1, or once MCASP_RXBUFn registers of all the serializers, that have previously raised their RRDY flags, are read by the host.

Whenever RDATA is set, the AREVT event is automatically generated on MCASPi_DREQ_RX line (if enabled in the MCASP_REVTCTL register) to notify the DMA of the MCASP_RXBUFn full status. An interrupt - MCASPi_IRQ_AREVT can be also generated if the RDATA interrupt is enabled in the MCASP_EVTCTLR register (for details, see Section 24.6.4.12.1, *Receive Data Ready Interrupt*).

Figure 24-129 shows the timing details of when AREVT event is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is received, the McASP sets the RDATA flag and generates an AREVT event. However, it takes up to five McASP interface clocks (AREVT Latency) before AREVT is active at the McASP boundary. Upon AREVT, the CPU can begin servicing the McASP by reading Word A from the MCASP_RXBUFn (service time). The CPU must read Word A from the MCASP_RXBUFn register no later than the setup time required by the McASP (Setup Time).

The maximum service time (see Figure 24-129) can be calculated as:
Service Time = Time Slot - AREVT Latency - Setup Time

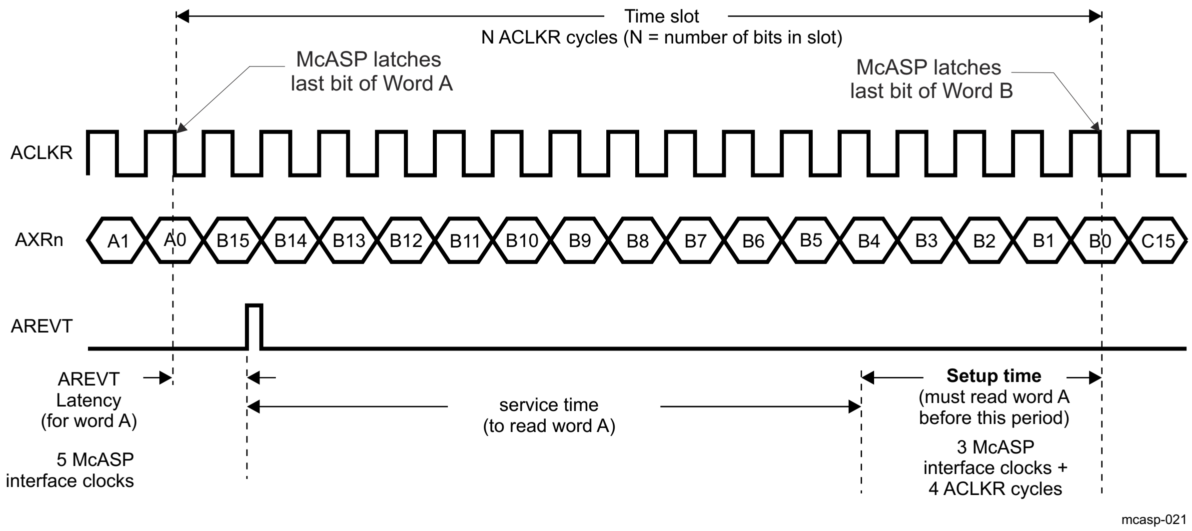


Figure 24-129. CPU Service Time Upon Receive Event (AREVT)

24.6.4.10.1.3 Transfers Through the Data Port (DATA)

> **CAUTION**
>
> To perform internal transfers through the DATA port, clear the XBUSEL/RBUSEL bit to 0b0 in the MCASP_TXFMT/MCASP_RXFMT register, respectively. Failure to do so may result in software malfunction.

> **Note**
>
> McASP1, McASP2, and McASP3, whose data ports are accessible directly via L3_MAIN, do not support FIFO/constant addressing modes. Incrementing transfers must be used instead.

In a typical McASP transfer scenario, the DMA Controller write accesses the XRBUFn transmit buffer through the McASP data port (DATA) on L3_MAIN Interconect for McASP1/2/3 and on L4_PER2 Interconect for McASP4/5/6/7/8 . CPU hosts can access both XRBUFn transmit and receive data buffers on their corresponding DATA port address via DATA port corresponding address. To perform transfers through the DATA port, simply have the DMA Controller write the McASP Tx buffer through Interconect DATA port location. Refer to Section 24.6.6.2.5. Although the transfer is passed through an integrated AFIFO transmit/receive buffer, the host (DMA or CPU) must follow the described below procedure to access the data buffers of each serializer, regardless the AFIFO is enabled or disabled. The AFIFO operation is described in Section 24.6.4.11.

For accesses through the DATA port, the DMA/CPU services all the serializers through accessing only a single address. In addition, as can be seen in Section 24.6.6.2.5, the same physical DATA port address is used regardless of a read or write access is performed. The McASP automatically cycles through the active slot transmitting/receiving serializers, internally generating the appropriate offsets.

> **Note**
>
> DATA port allows the DMA/CPU to automatically access only the data buffers. There is no way for DMA/CPU to access the McASP configuration registers addressing their corresponding McASP DATA port.

For transmit operations through the DATA port, the host must always write to the same transmit buffer DATA port address (which is same than the receive buffer DATA port adress) to service all of the active slot transmitting serializers. Regardless of McASP serializer 0 being configured inactive or active, the user software must always configure the destination address to match the DATA port location of TXBUF buffer (See Section 24.6.6.2.5).

In addition, the DMA/CPU must write the buffers of all transmitting serializers in incremental (although not necessarily consecutive) order. For example, if only serializers 1 and 3 are set up as active transmitters, the DMA/CPU should write to the same transmit buffer DATA port address twice - first data for serializer 1 and second data for serializer 3 upon each transmit data ready event. This exact servicing order must be followed so that data appears in the appropriate serializers.

Note

For write transfers through McASP DATA port it is preferable to use DMA on corresponding Interconect. This is because DMAs initiated traffic gets better advantage of the burst transfers supported by DATA port.

For receive operations through the DATA port, the DMA/CPU must always read from the same receive buffer DATA port address (which is same than the transmit buffer DATA port adress) to service all of the active slot receiving serializers. Regardless of McASP serializer 0 being configured inactive or active, the user software must always configure the DMA/CPU source address to match the DATA port location of RXBUF buffer (See Section 24.6.6.2.5).

In addition, reads from the receive buffer for all active slot receiving serializers through the Rx DATA port return data in incremental (although not necessarily consecutive) order. For example, if serializers 0, 1 and 3 are set up as active receivers, the MPU should read from the same receive buffer DATA port address three times to obtain data for serializers 0,1 and 3 in this exact order, upon each receive data ready event.

Note

To service a serializer for a transmit or receive operation through the McASP DATA port, the initiator always writes (preferably DMA) and reads from the same address (refer to Section 24.6.6.2.5), respectively.

See Section 24.6.6.2.5, *McASP_DATA Register Summary*, for more details about XRBUFn buffer physical address corresponding to the McASP DATA port on:

- Main Interconect (L3_MAIN or L4_PER2)

Note

When transmitting through the DATA port, the DMA/CPU must write data (at the same address) to each serializer configured as *active* (active slot selected in MCASP_TXTDM) and *transmit* (Tx enabled in MCASP_XRSRCTLn) within each time slot. Failure to do so results in a buffer underrun condition (see Section 24.6.4.15.1, *Buffer Underrun Error - Transmitter*). Similarly, when DMA/CPU receives, data must be read from each serializer configured as *active* (active slot selected in MCASP_RXTDM) and *receive* (Rx enabled in MCASP_XRSRCTLn) within each time slot. Failure to do results in a buffer overrun condition (see Section 24.6.4.15.2, *Buffer Overrun Error - Receiver*).

24.6.4.10.1.4 Transfers Through the Configuration Bus (CFG)

CAUTION

To perform internal transfers through the configuration bus, set the XBUSEL/RBUSEL bit to 1 in the MCASP_TXFMT/MCASP_RXFMT registers, respectively. Failure to do so may result in software malfunction.

Note

McASP1, McASP2, and McASP3, whose data ports are accessible directly via L3_MAIN do not support FIFO/constant addressing modes. Incrementing transfers must be used instead.

In this method, the DMA/CPU accesses the XRBUFn transmit or receive buffer through corresponding configuration bus (CFG) address.

The exact XRBUFn transmit/receive buffer physical address for any particular serializer is determined by adding the transmit/receive buffer alias register offset for that particular serializer to the base address of McASP CFG port actual for L4_PER2 accesses. The XRBUFn buffer of the n-th serializer configured as a transmitter is aliased - MCASP_TXBUFn in the CFG port address space. For example, the XRBUF2 transmit buffer is mapped as the MCASP_TXBUF2 register. Similarly, the XRBUFn buffer of the n-th serializer configured as a receiver is aliased - MCASP_RXBUFn in the CFG port address space. For example, the XRBUF3 receive buffer is mapped as the MCASP_RXBUF3 register.

Accessing the XRBUF through the DATA port (see Section 24.6.4.10.1.3) is different than CFG port accesses because the DATA port access demands the same physical address, regardless of transfer direction or current channel index , while accessing through the peripheral configuration port - CFG, the DMA/CPU must provide the exact MCASP_TXBUFn or MCASP_RXBUFn address upon accessing n-th serializer TX or RX buffer, respectively. For more details about MCASP_TXBUFn and MCASP_RXBUFn addresses corresponding to McASP CFG port, see Section 24.6.6.2.1, *MCASP_CFG Register Summary*.

24.6.4.10.1.5 Using a Device CPU for McASP Servicing

The device CPUs can be used to service the McASP transmit channels through interrupts (upon MCASPi_IRQ_AXEVT and MCASPi_IRQ_AREVT interrupts). Because these interrupt events are connected to device IRQ_CROSSBAR module, they could be software mapped to input IRQ lines of any device CPU. Another way to service the transmit and receive channels, a polling of the XDATA bit in the MCASP_TXSTAT register and RDATA bit in the MCASP_RXSTAT register can be performed by device CPUs, respectively. As discussed in Section 24.6.4.10.1.3, *Transfers Through the Data Port (DATA)*, and Section 24.6.4.10.1.4, *Transfers Through the Configuration Bus (CFG)* , the device CPUs can access McASP XRBUF serializer buffer through their corresponding DATA and CFG port locations.

To use the device CPUs to service the McASP through interrupts, the XDATA/RDATA bit must be enabled in the respective MCASP_EVTCTLX/MCASP_EVTCTLR registers, to generate interrupts MCASPi_IRQ_AXEVT/ MCASPi_IRQ_AREVT to the device CPUs upon data ready

24.6.4.10.1.6 Using the DMA for McASP Servicing

The typical scenario is to use the DMA to service the McASP transmit and receive logic through the DATA port, although the DMA can also service the McASP through the configuration bus (CFG). The transfer passes through integrated AFIFO transmit/receive buffer. If AFIFO is enabled, DMA requests are collected and fed to a device DMA controller (see Figure 24-119). The data transfer is managed by the AFIFO according to generated transmit and receive events in the McASP and data is fed to transmit buffers and fetched from receive buffers as described in Section 24.6.4.11. The generation of transmit and receive request is described below. After generation of transmit/receive DMA events from McASP module, these events are collected in AFIFO and on specific AFIFO conditions described in Section 24.6.4.11 the requests (transmit or receive) are forwarded to a DMA controller via MCASPi_DREQ_TX and MCASPi_DREQ_RX outputs. If the AFIFO is disabled (default state) it is transperrant for the McASP module and all request are directly sent to the DMA controller.

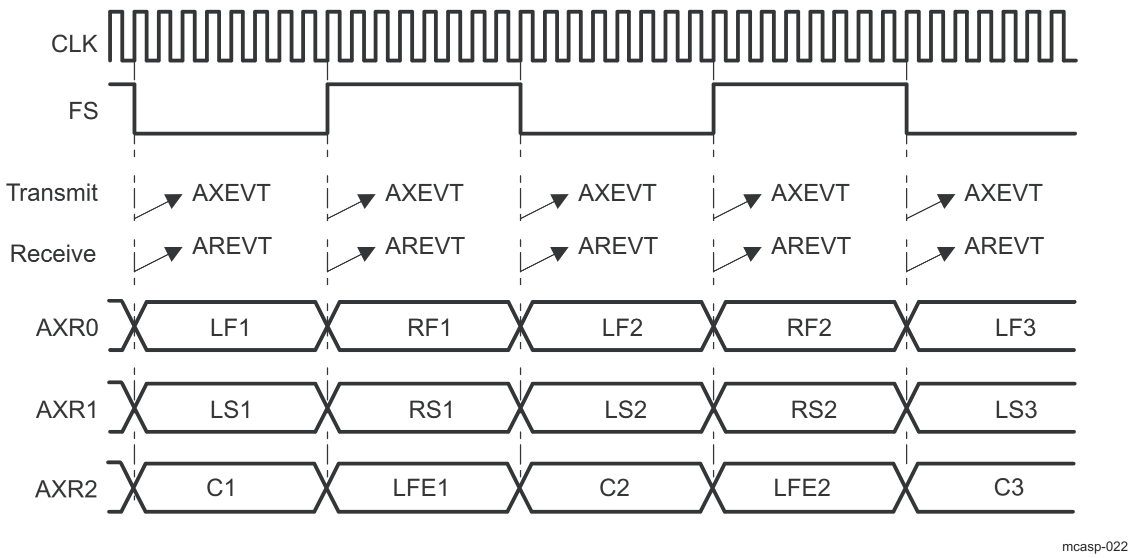


Figure 24-130. DMA Transmit and Receive Event in an Audio Example – One Event

In transmit mode, the DMA event - AXEVT (MCASPi_DREQ_TX output), which is triggered upon each XDATA transition from 0 to 1, is used to service the McASP TXBUFn transmit buffers. In receive mode, the DMA event AREVT (MCASPi_DREQ_RX output) which is triggered upon each RDATA transition from 0 to 1, is used to service the McASP RXBUFn receive buffers.

Figure 24-130 is an example of an audio system with six audio channels (LF, RF, LS, RS, C and LFE) transmitted or received through the McASP signals - AXR0, AXR1 and AXR2. It shows the points at which events AXEVT/AREVT are triggered.

In Figure 24-130, a Tx DMA event AXEVT is triggered on each time slot. In the example, AXEVT is triggered for each of the transmit audio channel time slot (time slot for channels LF, LS, and C; and time slot for channels RF, RS, LFE). Transmit DMA events are generated automatically upon transmit data ready, provided that DMA TX requests generation is enabled in the MCASP_XEVTCTL register. Similarly, Rx DMA event AREVT is triggered for each of the receive audio channel time slot. Receive DMA events are generated automatically upon receive data ready, provided that DMA RX requests generation is enabled in the MCASP_REVTCTL register.

24.6.4.11 McASP Audio FIFO (AFIFO)

The AFIFO contains two FIFOs: one Read FIFO (RFIFO), and one Write FIFO (WFIFO). The RFIFO and the WFIFO are the same size: 64 32-bit Words. To ensure backward compatibility with existing software, both the Read and Write FIFOs are disabled by default. See Figure 24-131 for a high-level block diagram of the AFIFO. The AFIFO may be enabled/disabled and configured via the WFIFOCTL and RFIFOCTL registers. Note that if the Read or Write FIFO is to be enabled, it must be enabled prior to initializing the receive/transmit section of the McASP.

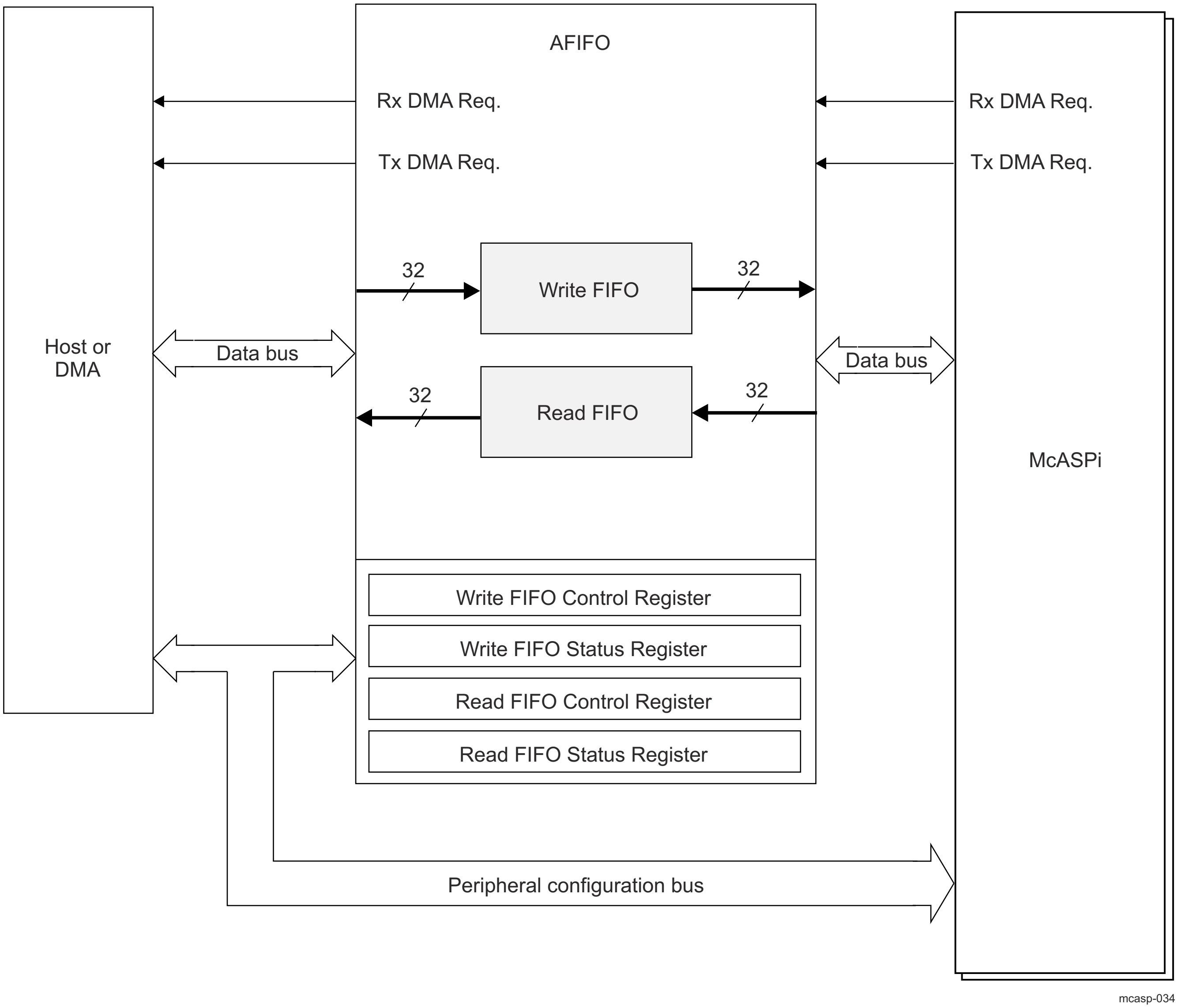


Figure 24-131. McASP Audio FIFO (AFIFO) Block Diagram

24.6.4.11.1 AFIFO Data Transmission

When the Write FIFO is disabled, transmit DMA requests pass through directly from the McASP to the host/DMA controller. Whether the WFIFO is enabled or disabled, the McASP generates transmit DMA requests as needed; the AFIFO is "invisible" to the McASP. When the Write FIFO is enabled, transmit DMA requests from the McASP are sent to the AFIFO, which in turn generates transmit DMA requests to the host/DMA controller. If the Write FIFO is enabled, upon a transmit DMA request from the McASP, the WFIFO writes WNUMDMA 32-bit words to the McASP if and when there are at least WNUMDMA words in the Write FIFO. If there are not, the WFIFO waits until this condition has been satisfied. At that point, it writes WNUMDMA words to the McASP. (See description for WFIFOCTL[7:0] WNUMDMA.) If the host CPU writes to the Write FIFO, independent of a transmit DMA request, the WFIFO will accept host writes until full. After this point, excess data will be discarded. Note that when the WFIFO is first enabled, it will immediately issue a transmit DMA request to the host. This is because it begins in an empty state, and is therefore ready to accept data.

24.6.4.11.1.1 Transmit DMA Event Pacer

The AFIFO may be configured to delay making a transmit DMA request to the host until the Write FIFO has enough space for a specified number of words. In this situation, the number of transmit DMA requests to the host or DMA controller is reduced. If the Write FIFO has space to accept WNUMEVT 32-bit words, it generates a transmit DMA request to the host and then waits for a response. Once WNUMEVT words have been written to the FIFO, it checks again to see if there is space for WNUMEVT 32-bit words. If there is space, it generates another transmit DMA request to the host, and so on. In this fashion, the Write FIFO will attempt to stay filled. Note that if transmit DMA event pacing is desired, WFIFOCTL[15:8] WNUMEVT should be set to a non-zero integer multiple of the value in WFIFOCTL[7:0] WNUMDMA. If transmit DMA event pacing is not desired, then the value in WFIFOCTL[15:8] WNUMEVT should be set equal to the value in WFIFOCTL[7:0] WNUMDMA.

24.6.4.11.2 AFIFO Data Reception

When the Read FIFO is disabled, receive DMA requests pass through directly from McASP to the host/DMA controller. Whether the RFIFO is enabled or disabled, the McASP generates receive DMA requests as needed; the AFIFO is “invisible” to the McASP. When the Read FIFO is enabled, receive DMA requests from the McASP are sent to the AFIFO, which in turn generates receive DMA requests to the host/DMA controller. If the Read FIFO is enabled and the McASP makes a receive DMA request, the RFIFO reads RNUMDMA 32-bit words from the McASP, if and when the RFIFO has space for RNUMDMA words. If it does not, the RFIFO waits until this condition has been satisfied; at that point, it reads RNUMDMA words from the McASP. (See description for RFIFOCTL[7:0] RNUMDMA.) If the host CPU reads the Read FIFO, independent of a receive DMA request, and the RFIFO at that time contains less than RNUMEVT words, those words will be read correctly, emptying the FIFO.

24.6.4.11.2.1 Receive DMA Event Pacer

The AFIFO may be configured to delay making a receive DMA request to the host until the Read FIFO contains a specified number of words. In this situation, the number of receive DMA requests to the host or DMA controller is reduced. If the Read FIFO contains at least RNUMEVT 32-bit words, it generates a receive DMA request to the host and then waits for a response. Once RNUMEVT 32-bit words have been read from the RFIFO, the RFIFO checks again to see if it contains at least another RNUMEVT words. If it does, it generates another receive DMA request to the host, and so on. In this fashion, the Read FIFO will attempt to stay empty. Note that if receive DMA event pacing is desired, RFIFOCTL[15:8] RNUMEVT should be set to a non-zero integer multiple of the value in RFIFOCTL[7:0] RNUMDMA. If receive DMA event pacing is not desired, then the value in RFIFOCTL[15:8] RNUMEVT should be set equal to the value in RFIFOCTL[7:0] RNUMDMA.

24.6.4.11.3 Arbitration Between Transmit and Receive DMA Requests

If both the WFIFO and the RFIFO are enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete. If only the WFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete. If only the RFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the receive DMA request. Once a transfer is in progress, it is allowed to complete.

24.6.4.12 McASP Events and Interrupt Requests

Table 24-336 lists all the transmit event flags. Table 24-337 lists all the Receive event flags. Source of each of these TX/RX events can be a TX/RX channel from any McASPi serializer configured as transmitter or receiver respectivly.

Table 24-336. TX Events

Event Mask[2]	Event Flag	Map to [1]	Description
MCASP_EVTCTLX[0] XUNDRN	MCASP_TXSTAT[0] XUNDRN	MCASPi_IRQ_AXEVT	Transmit buffer underrun
MCASP_EVTCTLX[1] XSYNCERR	MCASP_TXSTAT[1] XSYNCERR	MCASPi_IRQ_AXEVT	Unexpected transmit frame sync
MCASP_EVTCTLX[2] XCKFAIL	MCASP_TXSTAT[2] XCKFAIL	MCASPi_IRQ_AXEVT	Transmit clock failure
MCASP_EVTCTLX[3] XDMAERR	MCASP_TXSTAT[7] XDMAERR	MCASPi_IRQ_AXEVT	DATA port transmit error
MCASP_EVTCTLX[4] XLAST	MCASP_TXSTAT[4] XLAST	MCASPi_IRQ_AXEVT	Transmit last slot interrupt
MCASP_EVTCTLX[5] XDATA	MCASP_TXSTAT[5] XDATA	MCASPi_IRQ_AXEVT	Transmit data-ready interrupt
MCASP_EVTCTLX[7] XSTAFRM	MCASP_TXSTAT[6] XSTAFRM	MCASPi_IRQ_AXEVT	Transmit start of frame interrupt

Table 24-336. TX Events (continued)

Event Mask[(2)]	Event Flag	Map to [(1)]	Description
n.a.	MCASP_TXSTAT[8] XERR	n.a.	OR-event of all Tx-error events: (XDMAERR | XCKFAIL | XUNDRN | XSYNCERR). It is cleared ONLY when all error flags are cleared
n.a.	MCASP_TXSTAT[3] XTDMSLOT	n.a.	Qualifies the current TDM slot as an odd or an even slot.

(1) Every McASPi module generates separate IRQ event.
(2) Global events for all transmitting serializers in a single McASPi module.

Table 24-337. RX Events

Event Mask[(2)]	Event Flag	Map to[(1)]	Description
MCASP_EVTCTLR[0] ROVRN	MCASP_RXSTAT[0] ROVRN	MCASPi_IRQ_AREVT	Receive buffer overrun
MCASP_EVTCTLR[1] RSYNCERR	MCASP_RXSTAT[1] RSYNCERR	MCASPi_IRQ_AREVT	Unexpected receive frame sync
MCASP_EVTCTLR[2] RCKFAIL	MCASP_RXSTAT[2] RCKFAIL	MCASPi_IRQ_AREVT	Receive clock failure
MCASP_EVTCTLR[3] RDMAERR	MCASP_RXSTAT[7] RDMAERR	MCASPi_IRQ_AREVT	DATA port receive error
MCASP_EVTCTLR[4] RLAST	MCASP_RXSTAT[4] RLAST	MCASPi_IRQ_AREVT	Receive last slot
MCASP_EVTCTLR[5] RDATA	MCASP_RXSTAT[5] RDATA	MCASPi_IRQ_AREVT	Receive data-ready
MCASP_EVTCTLR[7] RSTAFRM	MCASP_RXSTAT[6] RSTAFRM	MCASPi_IRQ_AREVT	Receive start of frame
n.a.	MCASP_RXSTAT[8] RERR	n.a.	OR-event of all Rx-error events: (RDMAERR | RCKFAIL | ROVRN | RSYNCERR). RERR event is cleared once all error flags are cleared.
n.a.	MCASP_RXSTAT[3] RTDMSLOT	n.a.	Qualifies the current TDM slot as an odd or an even slot.

(1) Every McASP module generates separate IRQ event.
(2) Global events for all receiving serializers in a single McASPi module. These events and masks are available in same format for every McASPi module

Software has to read the MCASP_TXSTAT/MCASP_RXSTAT register to determine which event occurs at a global level for McASP Tx/Rx logic. In addition user software has to scan the XRDY/RRDY read-only flags in the MCASP_XRSRCTLn registers to determine which active serializer is the actual source of the event.

A Tx interrupt line (MCASPi_IRQ_AXEVT) is asserted (active high) when one of the MCASP_TXSTAT notified events occurs, provided that it is enabled in its corresponding MCASP_EVTCTLX bit. Similarly, a Rx interrupt line (MCASPi_IRQ_AREVT) is asserted (active high) when one of MCASP_RXSTAT notified events occurs, provided that it is enabled in its corresponding MCASP_EVTCTLR bit. See also Section 24.6.4.12.4, *Multiple Interrupts* and the Section 24.6.4.10.1, *Data Ready Status and Event/Interrupt Generation*.

24.6.4.12.1 Transmit Data Ready Event and Interrupt

The transmit data-ready interrupt (XDATA) is generated if XDATA is 1 in the MCASP_TXSTAT register and XDATA is enabled in MCASP_EVTCTLX. The Section 24.6.4.10.1, *Data Ready Status and Event/Interrupt Generation*, provides details on when XDATA is set in the MCASP_TXSTAT register.

A transmit-start-of-frame interrupt (XSTAFRM) is triggered by the recognition of a transmit frame sync.

A transmit-last-slot interrupt (XLAST) is a qualified version of the data-ready interrupt (XDATA). It has the same behavior than the data-ready interrupt, but is further qualified by having the data requested belonging to the last slot (the slot that just ended is the next-to-last TDM slot, the current slot is the last slot).

24.6.4.12.2 Receive Data Ready Event and Interrupt

The receive data-ready interrupt (RDATA) is generated if RDATA is 1 in the MCASP_RXSTAT register and RDATA is enabled in MCASP_EVTCTLR. The Section 24.6.4.10.1, *Data Ready Status and Event/Interrupt Generation*, provides details on when RDATA flag is set in the MCASP_RXSTAT register.

A receiver start of frame (RSTAFRM) interrupt is triggered by the recognition of a receiver frame sync.

A receiver last slot (RLAST) interrupt is a qualified version of the data ready interrupt (RDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data in the buffer come from the last TDM time slot (the slot that just ended was last TDM slot).

24.6.4.12.3 Error Interrupt

Upon detection, the following error conditions generate interrupt flags:

In the transmit status register (MCASP_TXSTAT):

- Transmit underrun (XUNDRN)
- Unexpected transmit frame sync (XSYNCERR)
- Transmit clock failure (XCKFAIL)
- Transmit DATA port error (XDMAERR)

Each interrupt source also has a corresponding enable bit in the transmit interrupt control register (MCASP_EVTCTLX). If the enable bit is set, an interrupt is requested when the interrupt flag is set in MCASP_TXSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

In the receive status register (MCASP_RXSTAT) :

- Receiver overrun (ROVRN)
- Unexpected receive frame sync (RSYNCERR)
- Receive clock failure (RCKFAIL)
- Receive DATA port error (RDMAERR)

Each interrupt source also has a corresponding enable bit in the receive interrupt control register (MCASP_EVTCTLR). If the enable bit is set, an interrupt is requested when the interrupt flag is set in MCASP_RXSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

24.6.4.12.4 Multiple Interrupts

This only applies to interrupts and not to DMA requests. The following terms are defined:

- **Active Interrupt Request:** a flag in MCASP_TXSTAT is set and the interrupt is enabled in MCASP_EVTCTLX.
- **Outstanding Interrupt Request:** An interrupt request has been issued on one of the McASP transmit interrupt port, but that request has not yet been serviced.
- **Serviced:** The CPUs write to MCASP_TXSTAT to clear one or more of the active interrupt request flags.

The first interrupt request to become active for the serializer with the interrupt flag set in MCASP_TXSTAT/ MCASP_RXSTAT and the interrupt enabled in MCASP_EVTCTLX/MCASP_EVTCTLR generates a request on the McASP transmit or receive interrupt port.

If more than one interrupt request becomes active in the same cycle, a single interrupt request is generated on the McASP transmit or receive interrupt port. Subsequent interrupt requests that become active while the first interrupt request is outstanding do not immediately generate a new request pulse on the McASP transmit or receive interrupt port.

The interrupt is serviced with the CPU writing to MCASP_TXSTAT/MCASP_RXSTAT. If any interrupt requests are active after the write, a new request is generated on the McASP transmit or receive interrupt port.

One outstanding interrupt request is allowed on each port, so a transmit and a receive interrupt request may both be outstanding at the same time.

24.6.4.13 DMA Requests

The McASP can generate one DMA request to the DMA_CROSSBAR to transmit (MCASPi_DREQ_TX) or receive (MCASPi_DREQ_RX) data. A DMA request to transmit data is generated if the XDATDMA bit in the

MCASP_XEVTCTL register is cleared. A DMA request to receive data is generated if the RDATDMA bit in the MCASP_REVTCTL register is cleared.

24.6.4.14 Loopback Modes

The McASP features a digital loopback mode (DLB) that allows loopback test transfers in TDM mode between McASP transmitters and receivers within the same device. In loopback mode, the output of a transmit serializer is connected internally to the input of a receive serializer. Therefore, a receiver data can be checked against a transmitter data to ensure that the McASP settings are correct. Digital loopback mode applies to TDM mode only (2 to 32 slots in a frame). It does not apply to DIT mode (XMOD = 0x180) or burst mode (XMOD = 0).

Figure 24-132shows the basic logical connection of the serializers in loopback mode.

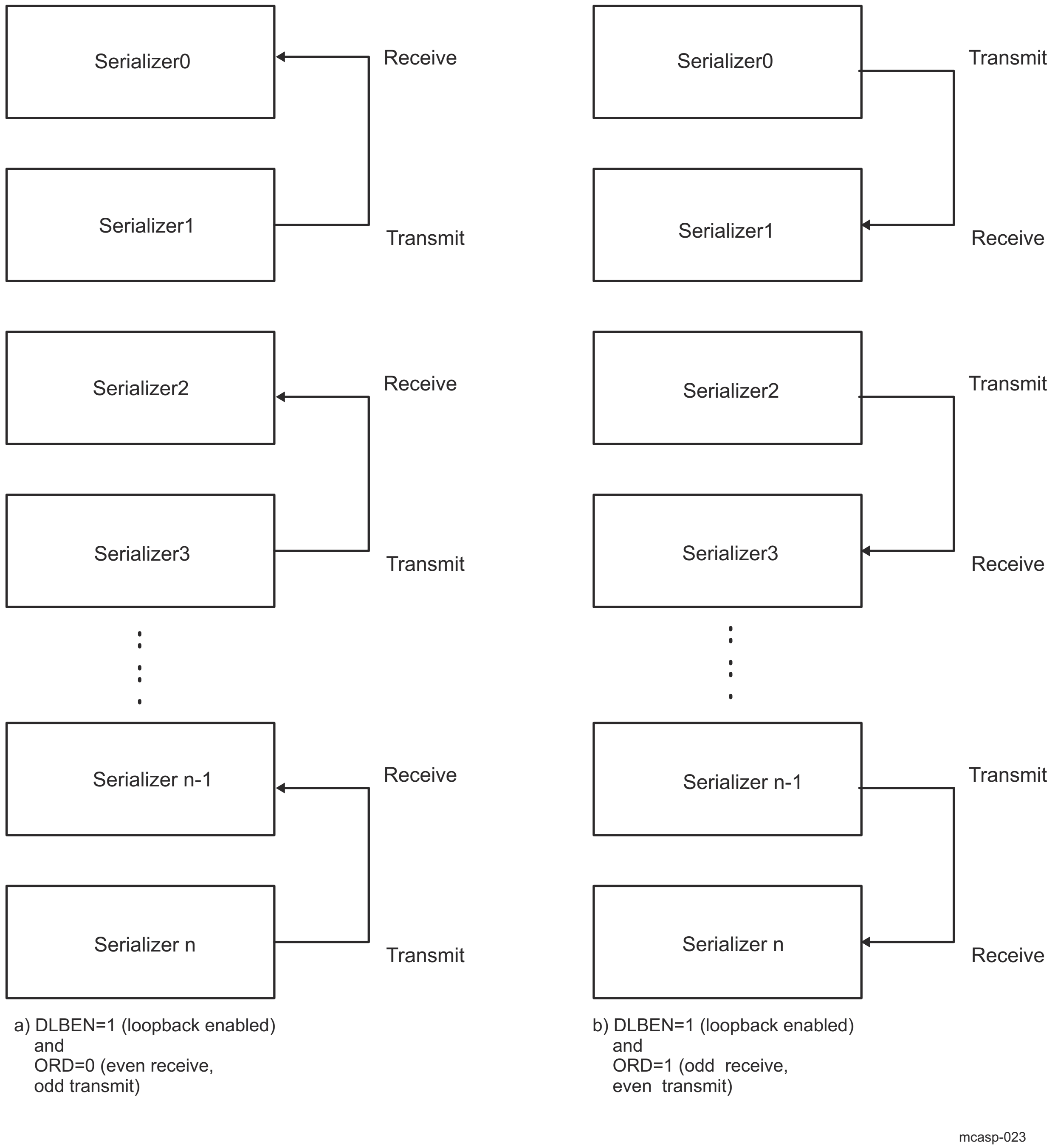


Figure 24-132. McASP Serializers Operation in Loopback Mode

Two types of loopback connections are possible, selected by the ORD bit in the digital loopback control register - MCASP_LBCTL as follows:

- ORD = 0: Outputs of odd serializers are connected to inputs of even serializers. If this mode is selected, the odd serializers must be configured as transmitters and even serializers as receivers.
- ORD = 1: Outputs of even serializers are connected to inputs of odd serializers. If this mode is selected, the even serializers must be configured as transmitters and odd serializers as receivers.

User can choose in software (bit IOLBEN of the MCASP_LBCTL) between a McASP module internal loopback and a device I/O level loopback.

When a **McASP internal loopback** is selected (MCASP_LBCTL[4] IOLBEN=0b0), it is NOT necessary to configure MCASP_PFUNC and MCASP_PDIR registers for McASP pin settings. Nevertheless, data can be optionally made externally visible at the I/O pin of the transmit serializer, if the pin is configured as a McASP output pin by setting the corresponding MCASP_PFUNC bit to 0 (i.e. to function as McASP, not GPIO) and MCASP_PDIR bit to 1 (output).

When a **device I/O level loopback** is selected (MCASP_LBCTL[4] IOLBEN=0b1), the MCASP_PFUNC and MCASP_PDIR registers must be configured with the appropriate settings for all AXRn pins, according to ORD bit configuration.

In case of device I/O loopback, the connectivity is externally applied between device pads (i.e. reaching device I/O buffers).

Hence, the corresponding padconfiguration registers must be appropriately configured in the device Control Module - CTRL_MODULE_CORE_PAD. For more details, see *Pad Configuration Rregisters*, in *Control Module*.

When In loopback mode, the transmit clock and frame sync are used by both the transmit and receive sections of the McASP. The transmit and receive sections operate synchronously. This is achieved by setting the MODE bitfield of the MCASP_LBCTL register to 0x1 and the ASYNC bit of the MCASP_ACLKXCTL register to 0b0.

24.6.4.14.1 Loopback Mode Configurations

This is a summary of the settings required for digital loopback mode for TDM format :

- The MCASP_LBCTL[0] DLBEN bit must be set to 0b1 to enable a loopback mode. It must be kept at 0b0 during normal McASP operation.
- The MCASP_LBCTL[4] IOLBEN bit must be set to select between internal (McASP local) loopback mode or device I/O level loopback mode.
- The MCASP_LBCTL[3:2] MODE bitfield must be set to 0x1 for both the transmit and receive sections to use the transmit clock and frame sync generator.
- The MCASP_LBCTL[1] ORD must be programmed appropriately to select odd or even serializers to be transmitters or receivers.
- The corresponding serializers must be configured accordingly.
- The bit - MCASP_ACLKXCTL[6] ASYNC must be cleared to 0b0 to ensure synchronous transmit and receive operations.
- The bitfields - MCASP_RXFMCTL[15:7] RMOD and MCASP_TXFMCTL[15:7] XMOD must be set within range (0x2- 0x20) to indicate TDM mode.

Note

Loopback mode does not apply to DIT or burst mode, because McASP receivers do NOT natively support DIR - reception.

24.6.4.15 Error Reporting

The McASP includes error-checking capability for the serial protocol and data underrun. In addition, the McASP includes a timer that continually measures the high-frequency master clock every 32 AHCLKX clock cycles. The value of the timer can be read to get a measurement of the clock frequency and has a minimum and maximum range setting that can set an error flag if the master clock goes out of a specified range.

When one or more errors (software selectable) are detected, an interrupt can be generated if desired, based on one or more error sources.

24.6.4.15.1 Buffer Underrun Error -Transmitter

A buffer underrun occurs when a serializer is instructed by the transmit state-machine to transfer data from XRBUFn buffer to XRSRn shift register, but the corresponding (MCASP_TXBUFn) register has not yet been written with new data since the last time the transfer occurred. When this occurs, the transmit state-machine sets the XUNDRN flag.

An underrun is checked only once per time slot. The MCASP_TXSTAT[0] XUNDRN flag is set when an underrun condition occurs. Once set, the XUNDRN flag remains set until the host explicitly writes 1 to the XUNDRN bit to clear it.

In DIT mode, a pair of BMC zeros is shifted out when an underrun occurs (four bit times at 128 bfs). By shifting out a pair of zeros, a clock can be recovered on the receiver. To recover, reset the McASP and restart with the proper initialization.

In TDM mode, during an underrun case, a long stream of zeros are shifted out causing the DACs to mute. To recover, reset the McASP and start again with the proper initialization.

24.6.4.15.2 Buffer Overrun Error-Receiver

A buffer overrun occurs when a serializer is instructed to transfer data from XRSRn shift register to XRBUFn receiver buffer, but the corresponding MCASP_RXBUFn register has not yet been read since the last time the transfer occured. When this occurs, the receiver state machine sets the overrun flag - ROVRN. However, the individual serializer writes over the data in the XRBUFn buffer register (destroying the previous sample) and continues shifting.

An overrun is checked only once per time slot. The MCASP_RXSTAT[0] ROVRN flag is set when an overrun condition occurs. It is possible that an overrun occurs on one time slot but then the host catches up and does not cause an overrun on the following time slots. However, once the ROVRN flag is set, it remains set until the host explicitly writes a 1 to the ROVRN bit to clear the ROVRN bit.

24.6.4.15.3 DATA Port Error - Transmitter

A transmit DATA port error, as indicated by the XDMAERR flag in the MCASP_TXSTAT register, occurs when the DMA or device CPU writes more words to the DATA port of the McASP than it should.

The MCASP_TXSTAT[7] XDMAERR=0b1 indicates that the DMA or device CPU wrote too many words to the McASP DATA port for a given transmit DMA event. Writing too few words results in a transmit underrun error setting XUNDRN in MCASP_TXSTAT.

While XDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or device CPU. The McASP transmitter and the DMA must be reinitialized to resynchronize them.

24.6.4.15.4 DATA Port Error - Receiver

A receive DATA port error, as indicated by the RDMAERR flag in the MCASP_RXSTAT register, occurs when the DMA or device CPU reads more words from the DATA port of the McASP than it should.

The MCASP_RXSTAT[7] RDMAERR indicates that the DMA or device CPU read too many words from the McASP DATA port for a given receive AREVT event. Reading too few words results in a receiver overrun error setting ROVRN in MCASP_RXSTAT.

While RDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or device CPU. The McASP receiver and the DMA must be reinitialized to resynchronize them.

24.6.4.15.5 Unexpected Frame Sync Error

An unexpected frame sync occurs in when:

- in burst mode and TDM mode, the next active edge of the frame sync occurs early such that the current slot will not be completed by the time the next slot is scheduled to begin.
- in TDM mode, an unexpected frame sync occurs also if the frame sync does NOT occur exactly during the correct bit clock (not a cycle earlier or later) and before slot 0.

When an unexpected frame sync occurs, there are two possible actions depending upon when the unexpected frame sync occurs:

1. **Early:** An early unexpected frame sync occurs when the McASP is in the process of completing the current frame and a new frame sync is detected (not including overlap that occurs due to a 1 or 2 bit frame sync delay). When an early unexpected frame sync occurs:
   - Error event flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Current frame is not resynchronized. The number of bits in the current frame is completed. The next frame sync, which occurs after the current frame is completed, will be resynchronized.
2. **Late:** A late unexpected frame sync occurs when there is a gap or delay between the last bit of the previous frame and the first bit of the next frame. When a late unexpected frame sync occurs (as soon as the gap is detected):
   - Error event flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Resynchronization occurs upon the arrival of the next frame sync.

Late frame sync is detected the same way in burst mode and TDM mode. However, in burst mode, late frame sync is not meaningful and its interrupt enable should not be set.

24.6.4.15.6 Clock Failure Detection

24.6.4.15.6.1 Clock Failure Check Startup

It is initially expected of the clock failure circuits to generate an error until at least one measurement is taken. Therefore, the clock failure interrupts, clock switch, and mute functions should not be enabled immediately, but only after a specific startup procedure.

To start the transmit clock failure check procedure:

1. Configure the transmit clock failure detect logic (XMIN, XMAX, XPS) in the transmit clock check control register (MCASP_TXCLKCHK).
2. Clear the transmit clock failure flag (XCKFAIL) in the transmit status register (MCASP_TXSTAT).
3. Wait until the first measurement is taken (> 32 AHCLKX clock periods).
4. Verify that no clock failure is detected.
5. Repeat Step 2 through Step 4 until the clock is running and is no longer issuing clock failure errors.
6. After the transmit clock is measured and falls within the acceptable range, the following can be enabled:
   a. The transmit clock failure interrupt enable bit (XCKFAIL) in the transmitter interrupt control register (MCASP_EVTCTLX)

To start the receive clock failure check procedure:

1. Configure receive clock failure detect logic (RMIN, RMAX, RPS) in the receive clock check control register (MCASP_RXCLKCHK).
2. Clear receive clock failure flag (RCKFAIL) in the receive status register (MCASP_RXSTAT).
3. Wait until first measurement is taken (> 32 AHCLKR clock periods).
4. Verify no clock failure is detected.
5. Repeat steps 2–4 until clock is running and is no longer issuing clock failure errors.
6. After the receive clock is measured and falls within the acceptable range, the following may be enabled:
   a. the receive clock failure (RCKFAIL) interrupt enable bit in the receive interrupt control register (MCASP_EVTCTLR)

24.6.4.15.6.2 Transmit Clock Failure Check and Recovery

The transmit clock failure check circuit (see Figure 24-133) works off the internal McASP interface clock and the external high-frequency serial clock (AHCLKX). It continually counts the number of interface clocks for every 32 high-rate serial clock (AHCLKX) periods, and stores the count in XCNT of the transmit clock check control register (MCASP_TXCLKCHK) every 32 high-rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (XMIN), and automatically flags an interrupt (XCKFAIL in MCASP_TXSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is less than XMIN. The logic continually compares the current count (from the running interface clock counter) to the maximum allowable boundary (XMAX). This is so that if the external clock completely stops, the counter value is not copied to XCNT. An out-of-range maximum condition occurs when the count is greater than XMAX. The XMIN and XMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

For the transmit clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset.

If a clock failure is detected, the transmit clock failure flag (XCKFAIL) in MCASP_TXSTAT is set. This causes an interrupt if the transmit clock failure interrupt enable bit (XCKFAIL) in MCASP_EVTCTLX is set.

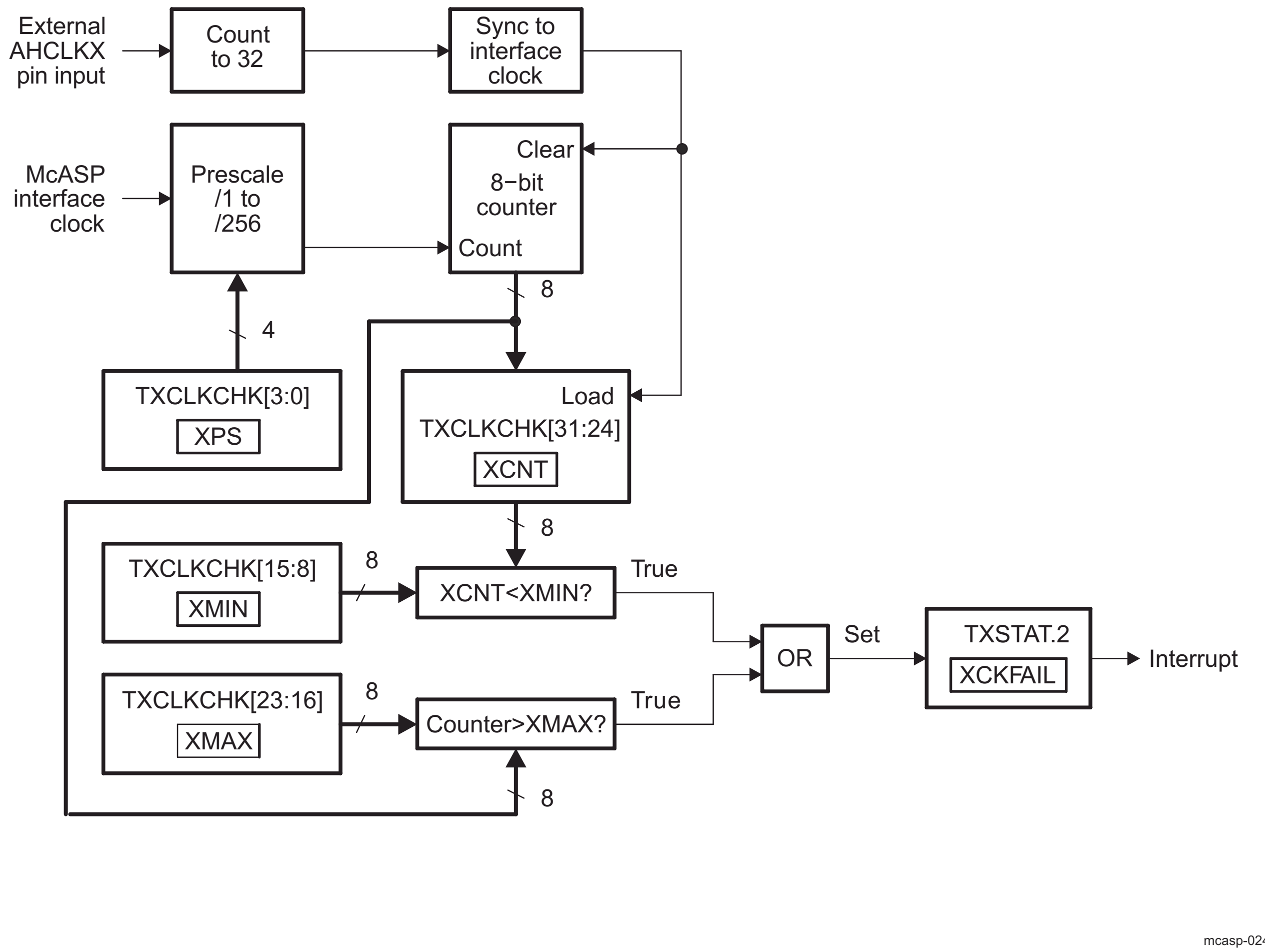


Figure 24-133. Transmit Clock Failure Detection Circuit Block Diagram

24.6.4.15.6.3 Receive Clock Failure Check and Recovery

The receive clock failure check circuit (see Figure 24-134) works off both the internal McASP interface clock and the high-frequency serial clock (AHCLKR) coming from the device clock generator. It continually counts the number of interface clocks for every 32 high rate serial clock (AHCLKR) periods, and stores the count in RCNT of the receive clock check control register (MCASP_RXCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (RMIN) and automatically flags an event (RCKFAIL in MCASP_RXSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than RMIN. The logic continually compares the current count (from the running interface clock counter) against the maximum allowable boundary (RMAX). This is in case the external clock completely stops, so that the counter value is not copied to RCNT. An out-of-range maximum condition occurs when the count is greater than RMAX. Note that the RMIN and RMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

In order for the receive clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset.

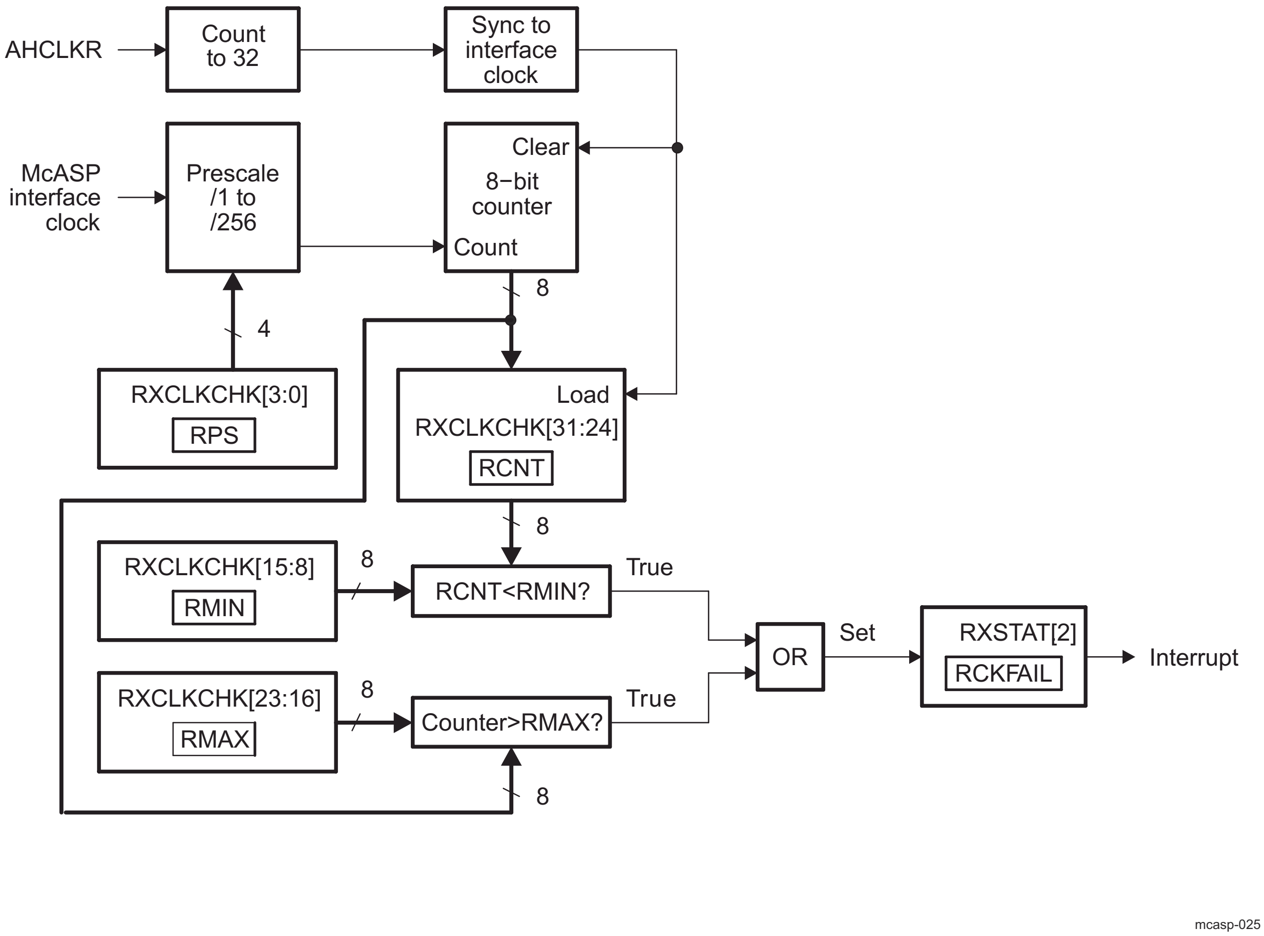


Figure 24-134. Receive Clock Failure Detection Circuit Block Diagram

24.6.5 McASP Low-Level Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the McASP module.

24.6.5.1 Global Initialization

24.6.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the McASP module is used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the McASP (for more information, see *McASP Integration*, and *McASP Environment*).

Table 24-338, describes the global initialization of surrounding modules.

Table 24-338. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module functional and interface clocks must be enabled. (See *Clock Management Functional Description*, in *Power, Reset, and Clock Management*.)
Control module	Module-specific pad muxing and other pad configurations must be set in the control module. (See *Pad Configuration Registers*, in *Control Module*).
(Optional) IRQ_CROSSBAR	Interrupt crossbar configuration must be done to enable the interrupts from the McASP. For more details on IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
(Optional) DMA_CROSSBAR	DMA configuration must be done to enable the McASP DMA data channel requests. For more information on DMA_CROSSBAR module configuration, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
(Optional) L4_PER2 and L3_MAIN Interconnects	For more information about the interconnect configuration, see *L3_MAIN Interconnect Overview* in *L3 Interconnect*.

Note

The IRQ_CROSSBAR and the DMA_CROSSBAR configurations are required when the interrupt and DMA-based communication modes are used. Further initialization of the selected IRQ and DMA controllers of the host CPU must be done for full functionality of the McASP DMA and IRQ lines.

24.6.5.1.2 McASP Global Initialization

24.6.5.1.2.1 Main Sequence – McASP Global Initialization for DIT-Transmission

The procedure in Table 24-339 initializes the McASP serializers transmitters to operate in DIT-mode (S/PDIF-transmission protocol) after a power-on reset (POR).

CAUTION

Before performing McASP global initialization, If external clock ACLKR is used, it must be running already for proper synchronization of the MCASP_GBLCTL register.

Table 24-339. McASP Transmitters Global Initialization for DIT-Mode Operation

Step	Register/Bit Field/Programming Model	Value
1. Apply software reset to different McASP components.	MCASP_GBLCTL[12:8]	0x00
2. Poll the bits to ensure the active reset value (0x00) is successfully latched into the register.	MCASP_GBLCTL[12:8]	=0x00
3. Configure the local power management.	PWRIDLESYSCONFIG[1:0] IDLE_MODE	0x1
4. Configure the transmit format unit.	See Section 24.6.5.1.2.1.1.	
5. Configure the transmit frame sync generator.	See Section 24.6.5.1.2.1.2.	
6. Configure the transmit clock generator.	See Section 24.6.5.1.2.1.3.	

Table 24-339. McASP Transmitters Global Initialization for DIT-Mode Operation (continued)

Step	Register/Bit Field/Programming Model	Value
7. Configure the TDM sequencer—set all slots active.	MCASP_TXTDM[31:0] XTDMS	0xFFFF FFFF
8. Configure the desired n-th serializer (n=0 to 3) for transmit mode operation. [(3)]	MCASP_XRSRCTLn [1:0] SRMOD	0x1
9. Configure the McASP pins functionality.	See Section 24.6.5.1.2.1.4.	
10. Enable the McASP DIT - transmission mode.	MCASP_TXDITCTL[0] DITEN	0x1[(2)]
11. Configure DIT-specific subframe fields.	See Table 24-344.	
12. Release from reset state the divider that outputs the AHCLKX clock.[(1)]	MCASP_GBLCTL[9] XHCLKRST	0x1
13. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[9] XHCLKRST	=0x1
14. Release from reset state the divider that outputs the ACLKX clock. [(1)]	MCASP_GBLCTL[8] XCLKRST	0x1
15. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[8] XCLKRST	=0x1

(1) During reset state the local McASP internal clock dividers maintain a 1:1 ratio at their outputs. The values stored in the MCASP_AHCLKXCTL and MCASP_ACLKXCT registers are ignored; hence, the transmission clock does not stop during the reset state of the dividers.
(2) This globally configures all active transmitters to operate in DIT-mode.
(3) For an unused serializer n, write MCASP_XRSRCTLn [1:0] SRMOD=0x0 to disable it.

24.6.5.1.2.1.1 Subsequence – Transmit Format Unit Configuration for DIT-Transmission

The procedure in Table 24-340 configures the transmit frame format unit of the McASP module for a DIT-transmission.

Note

- The first transmit data bit always has a 0-bit delay.
- The bitstream is always transmitted in least-significant-bit (LSB)-first order.
- Pad value for extra bits in a certain slot is always 0.

Table 24-340. Transmit Format Unit Configuration for DIT-Transmission

Step	Register/Bit Field/Programming Model	Value
Configure the slot size to 32 bits.	MCASP_TXFMT[7:4] XSSZ	0xF
IF: the data to transmit is left- aligned	Software test condition	
Set data mask in the range 0xFFFF FF00 – 0xFFFF 0000.	MCASP_TXMASK[31:0] XMASK	0x-[(1)]
Rotate data right by a multiple-of-4- bit positions.	MCASP_TXFMT[2:0] XROT	0x- [(1)]
ELSE		
Set data mask in the range 0x00FF FFFF– 0x0000 FFFF.	MCASP_TXMASK[31:0] XMASK	0x-[(1)]
Rotate data right by 0-bit positions.	MCASP_TXFMT[2:0] XROT	0x0
ENDIF		
Select to write data to active serializers transmit buffers using peripheral (CFG) or DATA port	MCASP_TXFMT[3] XBUSEL	0x-

(1) Refer to Section 24.6.4.4.1, *Transmit Fromat Unit* and Section 24.6.4.4.1.2, *DIT-Mode Transmission Data Alignment Settings.*

24.6.5.1.2.1.2 Subsequence – Transmit Frame Synchronization Generator Configuration for DIT-Transmission

The procedure in Table 24-341 configures the transmit frame synchronization generator of the McASP module.

Note

The frame synchronization signal is always rising-edge active and always has a single-bit width.

Table 24-341. Transmit Frame-Synchronization Generator Configuration for DIT-Transmission

Step	Register/Bit Field/Programming Model	Value
Select 384-slot size block.	MCASP_TXFMCTL[15:7] XMOD	0x180
Select internally-generated transmit frame sync.	MCASP_TXFMCTL[1] FSXM	0x1

24.6.5.1.2.1.3 Subsequence – Transmit Clock Generator Configuration for DIT-Transmission

Note

By default, the ACLKX and AHCLKX clocks are generated only from the McASP internal clock source.

The procedure in Table 24-342 configures the transmit clock generator of the McASP module.

Table 24-342. Transmit Clock Generator Configuration in DIT-Mode

Step	Register/Bit Field/Programming Model	Value
Set the divisor for the internally generated high frequency clock– AHCLKX.	MCASP_AHCLKXCTL[11:0] HCLKXDIV	0x-
Set he divisor for the internally generated transmission clock– ACLKX.	MCASP_ACLKXCTL[4:0] CLKXDIV	0x-
Configure the transmit clock failure detect logic.	See Section 24.6.4.15.6.1, *Clock Failure Check Startup*.	

24.6.5.1.2.1.4 Subsequence - McASP Pins Functional Configuration

The procedure in Table 24-343 configures the McASP pins for McASP functionality.

Table 24-343. McASP Pins Functional Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module different pins to have McASP functionality.	MCASP_PFUNC[31:0]	0x0
Configure the McASP pins as outputs: AFSX AHCLKX ACLKX Desired i-th McASP data pin AXRi is configured as an output for DIT-transmission.	MCASP_PDIR[28] AFSX; MCASP_PDIR[27] AHCLKX; MCASP_PDIR[26] ACLKX; MCASP_PDIR [i] AXRi	0x1 0x1 0x1 0x1

24.6.5.1.2.1.5 Subsequence – DIT-specific Subframe Fields Configuration

The procedure in Table 24-344 configures the DIT-specific subframe fields as part of the S/PDIF format data.

Table 24-344. DIT-Specific Subframe Fields Configuration

Step	Register/Bit Field/Programming Model	Value
Configure the valid bit value for odd time slots.	MCASP_TXDITCTL[3] VB	0x-
Configure the valid bit value for even time slots.	MCASP_TXDITCTL[2] VA	0x-
Configure the user data bit for each subframe A and B in a 384-slot S/PDIF block.	MCASP_DITUDRAi[31:0] DITUDRAi, where i = 0 to 5 MCASP_DITUDRBi[31:0] DITUDRBi, where i = 0 to 5	0x- 0x-
Configure the channel status bit for each subframe A and B in a 384-slot S/PDIF block.	MCASP_DITCSRAi[31:0], where i = 0 to 5 MCASP_DITCSRBi[31:0], where i = 0 to 5	0x- 0x-

24.6.5.1.2.2 Main Sequence – McASP Global Initialization for TDM-Reception

The procedure in Table 24-345 initializes a McASP serializer n receiver(s) to operate in TDM-mode (the only mode supported by McASP receivers) after a power-on reset (POR). This is used for I2S (2-slot TDM) and other TDM-based audio protocols reception.

CAUTION

Before performing McASP global initialization, If external clock ACLKR is used, it must be running already for proper synchronization of the MCASP_GBLCTL register.

Note

The McASP receivers support only TDM-frames (including 384-TDM frames) reception. DIT-frames reception (i.e. S/PDIF stream) can be implemented indirectly via an external DIR-chip converter with DIT-input and TDM (I2S)-compatible output connected to device McASP receiver input (TDM-only compatible).

Table 24-345. McASP Receivers Global Initialization for TDM-Mode Operation

Step	Register/Bit Field/Programming Model	Value
1. Apply software reset to different McASP receive components.	MCASP_GBLCTL[4:0]	0x00
2. Poll the bits to ensure the active reset value (0x00) is successfully latched into the register.	MCASP_GBLCTL[4:0]	=0x00
3. Configure the local power management.	PWRIDLESYSCONFIG[1:0] IDLE_MODE	0x1
4. Configure the receive format unit.	See Section 24.6.5.1.2.2.1.	
5. Configure the receive frame sync generator.	See Section 24.6.5.1.2.2.2.	
6. Configure the receive clock generator.	See Section 24.6.5.1.2.2.3.	
7. Program all bits -RTDMSk (where k=0 to 31) according to the time slot characteristics desired (positions of active versus inactive slots within a frame).	MCASP_RXTDM [k] RTDMSk , where k=0 to 31	0x-
8. Configure the desired n-th serializer for receive mode operation. [4]	MCASP_XRSRCTLn [1:0] SRMOD	0x2
9. Configure the McASP pins functionality.	See Section 24.6.5.1.2.2.4.	
10. Optional: Configure a McASP Rx channel for a loopback operation (TDM mode only) in MCASP_LBCTL [31:0].	See Section 24.6.4.14.1, *Loopback Mode Configurations*.	0x-[5]
11. Release from reset state the divider that outputs the AHCLKR clock.[1] See also [2].	MCASP_GBLCTL[1] RHCLKRST	0x1
12. Poll the bit to ensure that it is successfully latched in the register. See also [2].	MCASP_GBLCTL[1] RHCLKRST	=0x1
13. Release from reset state the divider that outputs the ACLKR clock. [1] See also [3].	MCASP_GBLCTL[0] RCLKRST	0x1
14. Poll the bit to ensure that it is successfully latched in the register. See also [3].	MCASP_GBLCTL[0] RCLKRST	=0x1

(1) During reset state the local McASP internal clock dividers maintain a 1:1 ratio at their outputs. The values stored in the MCASP_AHCLKRCTL and MCASP_ACLKRCTL registers are ignored; hence, the reception clock does not stop during the reset state of the dividers.
(2) This step is necessary even if external high-frequency serial clocks are used.
(3) This step can be skipped if external serial clocks are used and they are running.
(4) For an unused serializer n, write MCASP_XRSRCTLn [1:0] SRMOD=0x0 to disable it.

(5) In this case the receiver clock and frame sync are derived from the McASP transmitter logic, so MCASP_ACLKXCTL[6] ASYNC must be set to 0b0. Neither McASP internal receiver clock and frame sync generators, nor external clock and frame sync source are used.

24.6.5.1.2.2.1 Subsequence – Receive Format Unit Configuration in TDM Mode

The procedure in Table 24-346 configures the receive frame format unit of the McASP module for TDM slots reception.

Table 24-346. Receive Format Unit Configuration for TDM-Reception

Step	Register/Bit Field/Programming Model	Value
Configure the desired TDM-slot size	MCASP_RXFMT[7:4] RSSZ	0x-[(1)]
Set data mask out value (0x0000 0000 - 0xFFFF FFFF).	MCASP_RXMASK[31:0] RMASK	0x-[(2)]
Select a padding value for masked-out bits.	MCASP_RXFMT[14:13] RPAD	0x-
Specify position (0x0-0x1F) of the bit in corresponding register MCASP_RXBUFn which value to be used as a pad value in case MCASP_RXFMT[14:13] RPAD=0x2.	MCASP_RXFMT[12:8] RPBIT	0x-
Rotate data right by a multiple of 4- bit positions.	MCASP_RXFMT[2:0] RROT	0x- [(3)]
Received stream bit order (LSB- or MSB-first). Must be set to 0x1 for an I2S stream reception (MSB-first).	MCASP_RXFMT[15] RRVRS	0x- [(3)]
Specify a delay between frame sync and first bit of data in number of bits. Must be set to 0x1 for an I2S stream reception.	MCASP_RXFMT[17:16] RDATDLY	0x-
Select to read data from active serializers receive buffers using peripheral (CFG) or DATA port	MCASP_RXFMT[3] RBUSEL	0x-

(1) Refer to Section 24.6.4.4.2, *Receive Format Unit*, regarding options for received TDM-slot sizes.
(2) For more details on Rx masking value, refer to Section 24.6.4.4.2.1, *TDM - Mode Reception Data Alignment Settings*
(3) For more details on rotation and received TDM stream bit order, refer to Section 24.6.4.4.2.1, *TDM - Mode Reception Data Alignment Settings* and Table 24-333, *McASP RFU Settings*.

24.6.5.1.2.2.2 Subsequence – Receive Frame Synchronization Generator Configuration in TDM Mode

The procedure in Table 24-347 configures the transmit frame synchronization generator of the McASP module.

Note

The same bit - MCASP_ACLKXCTL[6] ASYNC which is used to determine if McASP receivers and transmitters work synchronously on the same clock, is also used to define if receiver frame sync is derived from the transmit frame sync generator, or generated independently in the receiver (either inernally or externally sourced). Hence, the settings in below table Table 24-347 have no effect, if MCASP_ACLKXCTL[6] ASYNC = 0.

Table 24-347. Receive Frame-Synchronization Generator Configuration for TDM-Reception

Step	Register/Bit Field/Programming Model	Value
Select number of TDM slots per frame. Must be set to 0x2, in case of an I2S-reception. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_RXFMCTL[15:7] RMOD	0x-[(1)]
Choose the receive frame sync width -single bit/single word. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_RXFMCTL[4] FRWID	0x-
Select start of received frame sync polarity - rising / falling edge. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_RXFMCTL[0] FSRP	0x-
IFreceive frame sync - FS is internally generated	Software test condition	

Table 24-347. Receive Frame-Synchronization Generator Configuration for TDM-Reception (continued)

Step	Register/Bit Field/Programming Model	Value
Select internally- generated receive frame sync. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_RXFMCTL[1] FSRM	0b1
If McASP receiver is required to output internally generated frame, AFSR pin must be set as an output in step 9 of the sequence documented in the Table 24-345. This must not be done in current step because the frame control register - MCASP_TXFMCTL must be appropriately configured prior to AFSR pin outputting a frame to an external device.	MCASP_PDIR[31] AFSR	0b1
ELSE		
Select externally- generated receive frame sync. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_RXFMCTL[1] FSRM	0b0
Setup the AFSR pin as input (device level: mcaspi_fsr)	MCASP_PDIR[31] AFSR	0b0
ENDIF		
To generate McASP receive frame sync in receiver logic, select an asynchronous frame sync.	MCASP_ACLKXCTL[6] ASYNC	0b1

(1) Must be set to 0x180 in case of 384-TDM slot frame reception from a DIR component I2S-output. For more details on TDM-frame settings, refer to Section 24.6.4.9.2 .

24.6.5.1.2.2.3 Subsequence – Receive Clock Generator Configuration

The procedure in Table 24-348 configures the receive clock generator of the McASP module.

Note

The settings in below table Table 24-348 have no effect, if MCASP_ACLKXCTL[6] ASYNC = 0 (i.e. receive clock is sourced from the inverted version of the transmit clock). For example, such is the case when McASP loopback mode is used.

Table 24-348. Receive Clock Generator Configuration

Step	Register/Bit Field/Programming Model	Value
To use the McASP receive clock generator, select an asynchronous receiver clock schema (ASYNC=1). Otherwise an inverted version of transmit clock XCLK is used (receiver synchronized with transmitter).	MCASP_ACLKXCTL[6] ASYNC	0b1
IF receive clock - RCLK is internally generated	Software test condition	
The high-speed receive clock - AHCLKR is internally generated based on AUXCLK		
Select an internally-generated high-frequency clock.	MCASP_AHCLKRCTL[15] HCLKRM	0b1
Select the internal high-speed clock source polarity: non-inverted or inverted.	MCASP_AHCLKRCTL[14] HCLKRP	0x-
Set the divisor for the internally generated high-frequency clock – AHCLKR in range (1 - 4096).	MCASP_AHCLKRCTL[11:0] HCLKRDIV	0x-
Select an internally-generated receive clock.	MCASP_ACLKRCTL[5] CLKRM	0b1
Receiver samples on rising/falling edge. Select Rx sampling on the rising edge if transmitter shifts out on falling edge, and vice versa.	MCASP_ACLKRCTL[7] CLKRP	0x-
Set the divisor for the internally generated receive clock– ACLKR in range (1 - 32).	MCASP_ACLKRCTL[4:0] CLKRDIV	0x-

Table 24-348. Receive Clock Generator Configuration (continued)

Step	Register/Bit Field/Programming Model	Value
Optional: If McASP receiver is required to output internally generated clock, ACLKR pin must be set as an output in step 9 of the sequence documented in the Table 24-345. This must not be done in current step because the clock control register - MCASP_ACLKRCTL must be appropriately configured prior to ACLKR pin outputting a receive clock to an external device.	MCASP_PDIR[29] ACLKR	0b1
ELSE		
Select an externally-generated receive clock. Note that in this case the AHCLKR signal path and the CLKRDIV divider are NOT used.	MCASP_ACLKRCTL[5] CLKRM	0b0
Receiver samples on rising/falling edge. Select Rx sampling on the rising edge if transmitter shifts out on falling edge, and vice versa.	MCASP_ACLKRCTL[7] CLKRP	0x-
Setup an input directon for the ACLKR pin	MCASP_PDIR[29] ACLKR	0b0
ENDIF		
Configure the transmit clock failure detect logic.	See Section 24.6.4.15.6.1, *Clock Failure Check Startup*.	

24.6.5.1.2.2.4 Subsequence—McASP Receiver Pins Functional Configuration

The procedure in Table 24-349 configures the McASP pins for McASP functionality.

Table 24-349. McASP Receiver Pins Functional Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module different pins to have McASP functionality.	MCASP_PFUNC[31:0]	0x0
Configure the McASP pins direction: AFSR ACLKR Desired n-th McASP data pin AXRn is configured as an input for receiving.	MCASP_PDIR[31] AFSR; MCASP_PDIR[29] ACLKR; MCASP_PDIR[n] AXRn;	0x-(1) 0x- (2) 0x0

(1) See Table 24-347.
(2) For more details on McASP clock configurations, refer to Table 24-348.

24.6.5.1.2.3 Main Sequence – McASP Global Initialization for TDM -Transmission

The procedure in Table 24-350 initializes a McASP serializer n transmitter(s) to operate in TDM-mode after a power-on reset (POR). This is used for I2S (2-slot TDM) and other TDM-based audio protocols transmission.

> **CAUTION**
>
> Before performing McASP global initialization, If external clock ACLKR is used, it must be running already for proper synchronization of the MCASP_GBLCTL register.

Table 24-350. McASP Transmitters Global Initialization for TDM-Mode Operation

Step	Register/Bit Field/Programming Model	Value
1. Apply software reset to different McASP transmit components.	MCASP_GBLCTL[12:8]	0x00
2. Poll the bits to ensure the active reset value (0x00) is successfully latched into the register.	MCASP_GBLCTL[12:8]	=0x00
3. Configure the local power management.	PWRIDLESYSCONFIG[1:0] IDLE_MODE	0x1
4. Configure the transmit format unit.	See Section 24.6.5.1.2.3.1.	
5. Configure the transmit frame sync generator.	See Section 24.6.5.1.2.3.2.	

Table 24-350. McASP Transmitters Global Initialization for TDM-Mode Operation (continued)

Step	Register/Bit Field/Programming Model	Value
6. Configure the transmit clock generator.	See Section 24.6.5.1.2.3.3.	
7. Program all bits - XTDMSk, where k=0 to 31, according to the time slot characteristics desired (positions of active versus inactive slots within a frame).	MCASP_TXTDM [k] XTDMSk, where k=0 to 31[4]	0x-
8. Configure the desired n-th serializer for transmit mode operation. [3]	MCASP_XRSRCTLn[1:0] SRMOD;	0x1
9. Setup all active transmitters to operate in TDM mode.	MCASP_TXDITCTL[0] DITEN	0x0[2]
10. Configure the McASP pins functionality.	See Section 24.6.5.1.2.3.4.	
11. Optional: Configure a McASP Tx channel for loopback operation (TDM mode only) in MCASP_LBCTL [31:0].	See Section 24.6.4.14.1, *Loopback Mode Configurations*.	0x-
12. Release from reset state the divider that outputs the AHCLKR clock. See [1]	MCASP_GBLCTL[9] XHCLKRST	0x1
13. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[9] XHCLKRST	=0x1
14. Release from reset state the divider that outputs the ACLKR clock. See [1]	MCASP_GBLCTL[8] XCLKRST	0x1
15. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[8] XCLKRST	=0x1

(1) During reset state the local McASP internal clock dividers maintain a 1:1 ratio at their outputs. The values stored in the MCASP_AHCLKX and MCASP_ACLKX registers are ignored; hence, the transmission clock does not stop during the reset state of the dividers.
(2) All active transmit channels operate either in TDM mode or in DIT mode depending on DITEN value. There is no option to choose Tx Mode between DIT and TDM separately per serializer transmitter.
(3) For an unused serializer n, write MCASP_XRSRCTLn [1:0] SRMOD=0x0 to disable it.
(4) Appropriately program in bitfield MCASP_XRSRCTLn [3:2] DISMOD, the desired level (high-impedance state, 0, or 1) at AXRn output, during time of inactive slots. Note, that this setting does NOT apply when all slots are programmed to be active within a frame (in particular DIT-mode).

24.6.5.1.2.3.1 Subsequence – Transmit Format Unit Configuration in TDM Mode

The procedure in Table 24-351 configures the transmit frame format unit of the McASP module for TDM slots transmission.

Table 24-351. Transmit Format Unit Configuration for TDM-Transmission

Step	Register/Bit Field/Programming Model	Value
Configure the desired TDM-slot size	MCASP_TXFMT[7:4] XSSZ	0x-[1]
Set data mask out value (0x0000 0000 - 0xFFFF FFFF).	MCASP_TXMASK[31:0] XMASK	0x-[2]
Select a padding value for masked-out bits.	MCASP_TXFMT[14:13] XPAD	0x-
Specify position (0x0-0x1F) of the bit in corresponding register MCASP_TXBUFn which value to be used as a pad value in case MCASP_TXFMT[14:13] XPAD=0x2.	MCASP_TXFMT[12:8] XPBIT	0x-
Rotate data right by a multiple of 4- bit positions.	MCASP_TXFMT[2:0] XROT	0x- [3]
transmitted stream bit order (LSB- or MSB-first). Must be set to 0x1 for an I2S stream transmission (MSB-first).	MCASP_TXFMT[15] XRVRS	0x- [3]
Specify a delay between frame sync and first bit of data in number of bits. Must be set to 0x1 for an I2S stream transmission.	MCASP_TXFMT[17:16] XDATDLY	0x-
Select to write data to active serializers transmit buffers using peripheral (CFG) or DATA port	MCASP_TXFMT[3] XBUSEL	0x-

(1) Refer to Section 24.6.4.4.1, *Transmit Format Unit* , regarding options for transmitted TDM-slot sizes.

(2) For more details on Tx masking value, refer to Section 24.6.4.4.1.1, *TDM - Mode Transmission Data Alignment Settings*
(3) For more details on rotation and transmitd TDM stream bit order, refer to Section 24.6.4.4.1.1, *TDM - Mode Transmission Data Alignment Settings* and Table 24-331, *McASP TFU TDM Mode Settings*.

24.6.5.1.2.3.2 Subsequence – Transmit Frame Synchronization Generator Configuration in TDM Mode

The procedure in Table 24-352 configures the transmit frame synchronization generator of the McASP module.

Table 24-352. Transmit Frame-Synchronization Generator Configuration for TDM-Transmission

Step	Register/Bit Field/Programming Model	Value
Select number of TDM slots per frame (2 - 32). Must be set to 0x2, in case of an I2S-transmission. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_TXFMCTL[15:7] XMOD	0x-
Choose the transmit frame sync width -single bit/single word. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_TXFMCTL[4] FXWID	0x-
Select start of transmit frame sync polarity - rising / falling edge. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_TXFMCTL[0] FSXP	0x-
IF transmit frame sync - FS is internally generated	Software test condition	
Select internally- generated transmit frame sync. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_TXFMCTL[1] FSXM	0b1
If McASP transmitter is required to output internally generated frame, AFSX pin must be set as an output in step 10 of the sequence documented in the Table 24-350. This must NOT be done in current step because the frame control register - MCASP_TXFMCTL must be appropriately configured prior to AFSX pin outputting a frame sync to an external device.	MCASP_PDIR[28] AFSX	0b1
ELSE		
Select externally- generated transmit frame sync. For more details on frame-sync generator, refer to Section 24.6.4.2.3.	MCASP_TXFMCTL[1] FSXM	0b0
Setup the AFSX pin as input	MCASP_PDIR[28] AFSX	0b0

24.6.5.1.2.3.3 Subsequence – Transmit Clock Generator Configuration for TDM Cases

The procedure in Table 24-353 configures the transmit clock generator of the McASP module.

Table 24-353. Transmit Clock Generator Configuration for TDM Cases

Step	Register/Bit Field/Programming Model	Value
IF transmit clock - XCLK is internally generated	Software test condition	
IF high-speed transmit clock - AHCLKX is internally generated based on AUXCLK	Software test condition	
Select an internally-generated high-frequency clock.	MCASP_AHCLKXCTL[15] HCLKXM	0b1
Select the high-frequency clock source polarity: non-inverted or inverted.	MCASP_AHCLKXCTL[14] HCLKXP	0x-
Set the divisor for the internally generated high-frequency clock – AHCLKX in range (1 - 4096).	MCASP_AHCLKXCTL[11:0] HCLKXDIV	0x-

Table 24-353. Transmit Clock Generator Configuration for TDM Cases (continued)

Step	Register/Bit Field/Programming Model	Value
Optional: If McASP transmitter is required to output internally generated high-frequency clock, AHCLKX pin must be set as an output in step 10 of the sequence documented in the Table 24-350. This must NOT be done in current step because the clock control register - MCASP_AHCLKXCTL must be appropriately configured prior to AHCLKX pin outputting a high-speed clock to an external device.	MCASP_PDIR[27] AHCLKX	0b1
ELSE		
Select an externally-generated high frequency clock (HCLKXDIV divider can not be used).	MCASP_AHCLKXCTL[15] HCLKXM	0b0
Select the high-speed transmit clock source polarity: non-inverted or inverted.	MCASP_AHCLKXCTL[14] HCLKXP	0x-
Setup an input directon for the AHCLKX pin	MCASP_PDIR[27] AHCLKX	0b0
ENDIF		
Select an internally-generated transmit clock.	MCASP_ACLKXCTL[5] CLKXM	0b1
Transmitter samples on rising/falling edge. Select Tx shifting out data on the rising edge if receiver samples on falling edge, and vice versa.	MCASP_ACLKXCTL[7] CLKXP	0x-
Set the divisor for the internally generated transmit clock– ACLKX in range (1 - 32).	MCASP_ACLKXCTL[4:0] CLKXDIV	0x-
Optional: If McASP transmitter is required to output internally generated clock, ACLKX pin) must be set as an output in step 10 of the sequence documented in the Table 24-350. This must NOT be done in current step because the clock control register - MCASP_ACLKXCTL must be appropriately configured prior to ACLKX pin outputting a transmit clock to an external device.	MCASP_PDIR[26] ACLKX	0b1
ELSE		
Select an externally-generated transmit clock. Note that in this case the AHCLKX signal path and the CLKXDIV divider are NOT used.	MCASP_ACLKXCTL[5] CLKXM	0b0
Transmitter samples on rising/falling edge. Select Tx shifting out data on the rising edge if receiver samples on falling edge, and vice versa.	MCASP_ACLKXCTL[7] CLKXP	0x-
Setup an input directon for the ACLKX pin	MCASP_PDIR[26] ACLKX	0b0
ENDIF		
Configure the transmit clock failure detect logic.	See Section 24.6.4.15.6.1, *Clock Failure Check Startup*.	

24.6.5.1.2.3.4 Subsequence—McASP Transmit Pins Functional Configuration

The procedure in Table 24-354 configures the McASP pins for McASP functionality.

Table 24-354. McASP Transmit Pins Functional Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module different pins to have McASP functionality.	MCASP_PFUNC[31:0]	0x0
Configure the McASP pins direction: AFSX AHCLKX ACLKX Desired n-th McASP data pin AXRn is configured as an ouput for transmission.	MCASP_PDIR[28] AFSR; MCASP_PDIR[27] AHCLKR; MCASP_PDIR[26] ACLKR; MCASP_PDIR[n] AXRn	0x-(1) 0x-(2) 0x- (2) 0x1

(1) See Table 24-352.

24.6.5.2 Operational Modes Configuration

24.6.5.2.1 McASP Transmission Modes

24.6.5.2.1.1 Main Sequence – McASP DIT- /TDM- Polling Transmission Method

Figure 24-135 shows the McASP DIT-/TDM- polling method.

Note

- The McASP polling transmission model considers the MPU/DSP as the source of audio data for the McASP transmission buffer.
- The transmit DMA request is disabled and the XDMAERR event is not analyzed.

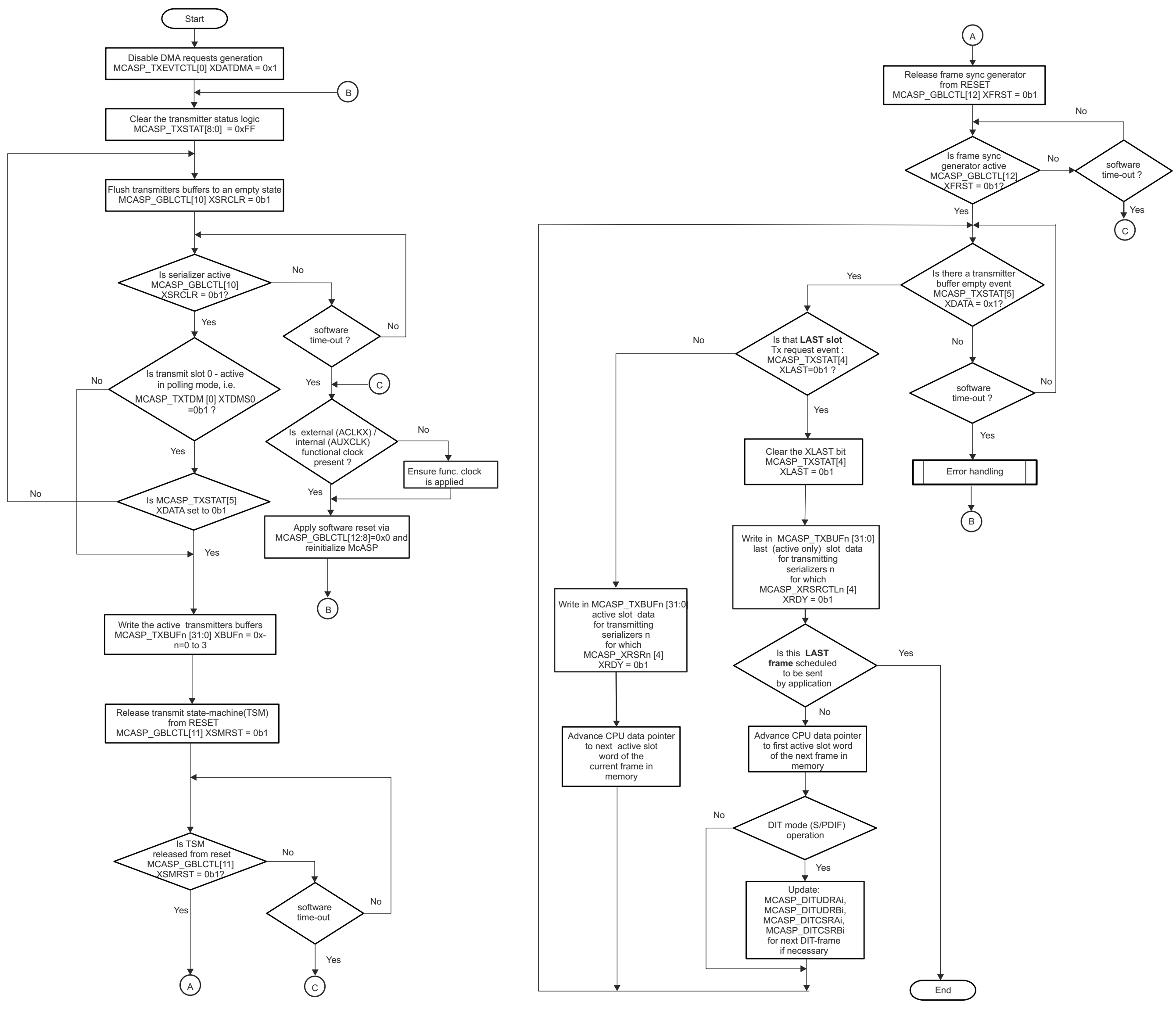


Figure 24-135. McASP DIT- /TDM- Transmission Polling Method

Table 24-355 summarizes the register call for the transmission DIT-/TDM- polling mode.

Table 24-355. Register Call Summary for Main Sequence – McASP DIT-/TDM- Transmission Polling Method

Register Name
MCASP_XEVTCTL
MCASP_TXSTAT
MCASP_GBLCTL
MCASP_TXTDM
MCASP_TXBUFn
MCASP_XRSRCTLn
MCASP_DITUDRAi (i=0 to 5)
MCASP_DITUDRBi (i=0 to 5)
MCASP_DITCSRAi (i=0 to 5)
MCASP_DITCSRBi (i=0 to 5)

Table 24-356 summarizes the subprocess call for the DIT-/TDM- transmission polling mode.

Table 24-356. Subprocess Call Summary for Main Sequence – McASP DIT-/TDM- Transmission Polling Method

Subprocess Name	Cross-Reference
Error handling	Figure 24-141

24.6.5.2.1.2 Main Sequence – McASP DIT- /TDM - Interrupt Transmission Method

Figure 24-136 shows the initial setup for interrupt-based transmission.

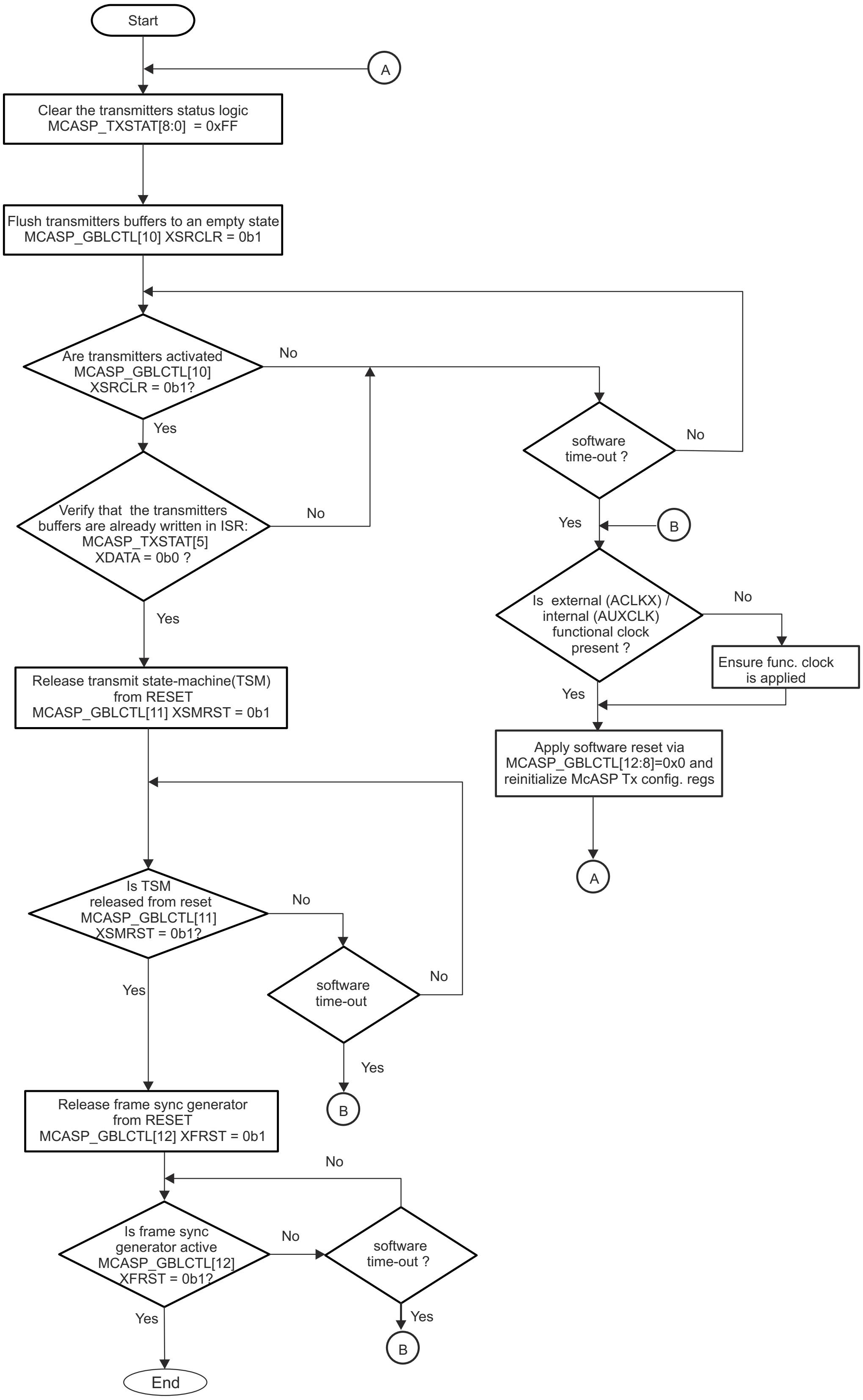


Figure 24-136. Subsequence – DIT-/TDM- Transmission Startup Procedure

Table 24-357 shows the configuration of the McASP using an interrupt method for DIT-/TDM- transmission.

Table 24-357. McASP DIT-/TDM- Interrupt Transmission Model

Step	Register/Bit Field/Programming Model	Value
Disable Tx DMA requests generation.	MCASP_XEVTCTL[0] XDATDMA	0x1
Enable the data ready event transmit interrupt.	MCASP_EVTCTLX[5] XDATA	0x1
Optional: Enable the transmit error event interrupts.	MCASP_EVTCTLX[2] XCKFAIL MCASP_EVTCTLX[1] XSYNCERR MCASP_EVTCTLX[0] XUNDRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt (useful for DIT user data/ channel status next S/PDIF frame info update.)	MCASP_EVTCTLX [7] XSTAFRM MCASP_EVTCTLX[4] XLAST	0x1 0x1
IFwrite transfer is through the McASP DATA port (MCASP_TXFMT[3] XBUSEL is set to 0b0).	Software test condition (setting is done in step4 of the *McASP Transmitters Global Initialization* - see Table 24-339)	
Enable the DATA port error based interrupt.	MCASP_EVTCTLX[3] XDMAERR	0x1
ELSE		
Disable the DATA port error based interrupt.	MCASP_EVTCTLX[3] XDMAERR	0x0
ENDIF		
DIT/TDM - Transmission Startup Procedure	See Figure 24-136.	

Table 24-358 summarizes the register call to initialize the McASP to transmit using interrupt events.

Table 24-358. Register Call Summary for Subsequence – McASP DIT-/TDM- Transmission Startup Procedure

Register Name	Register Name
MCASP_GBLCTL	MCASP_TXSTAT

24.6.5.2.1.3 Main Sequence –McASP DIT- /TDM - Mode DMA Transmission Method

Table 24-359 shows the configuration of the McASP using the DMA method for transmission. Possible interrupt error event servicing is also considered. Table 24-358 shows the initial setup for DMA - based transmission.

Note

Because of the DATA port burst access capability with the DMA method, it is strongly recommended that DMA transfers are initiated through the McASP DATA port.

Table 24-359. McASP DMA Transmission Model with Interrupt Events Servicing

Step	Register/Bit Field/Programming Model	Value
Recommended: Select DATA port to access the transmit buffers.	MCASP_TXFMT[3] XBUSEL	0x0
Enable the Tx DMA requests generation.	MCASP_XEVTCTL[0] XDATDMA	0x0
Enable the Tx DMA error event, because of McASP DATA port usage.	MCASP_EVTCTLX[3] XDMAERR	0x1
Optional: Enable the transmit error event interrupts.	MCASP_EVTCTLX[2] XCKFAIL MCASP_EVTCTLX[1] XSYNCERR MCASP_EVTCTLX[0] XUNDRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt.	MCASP_EVTCTLX [7] XSTAFRM MCASP_EVTCTLX[4] XLAST	0x1 0x1
Disable the data ready event transmit interrupt, as DMA is used to service this request.	MCASP_EVTCTLX[5] XDATA	0x0

Table 24-359. McASP DMA Transmission Model with Interrupt Events Servicing (continued)

Step	Register/Bit Field/Programming Model	Value
DMA startup transmission procedure. This procedure is identical than the one shown in Figure 24-136. The only difference is that DMA automatically services all the AXEVT events raised by the McASP, and no CPU data processing intervention is required. The CPU is involved only in error handling shown in Figure 24-139.	See Figure 24-136.	

24.6.5.2.2 McASP Reception Modes

24.6.5.2.2.1 Main Sequence – McASP Polling Reception Method

Figure 24-137 shows the McASP polling reception method.

Note

The McASP polling reception model considers thedevice CPUs as the accessor of audio data from the McASP receive buffers.

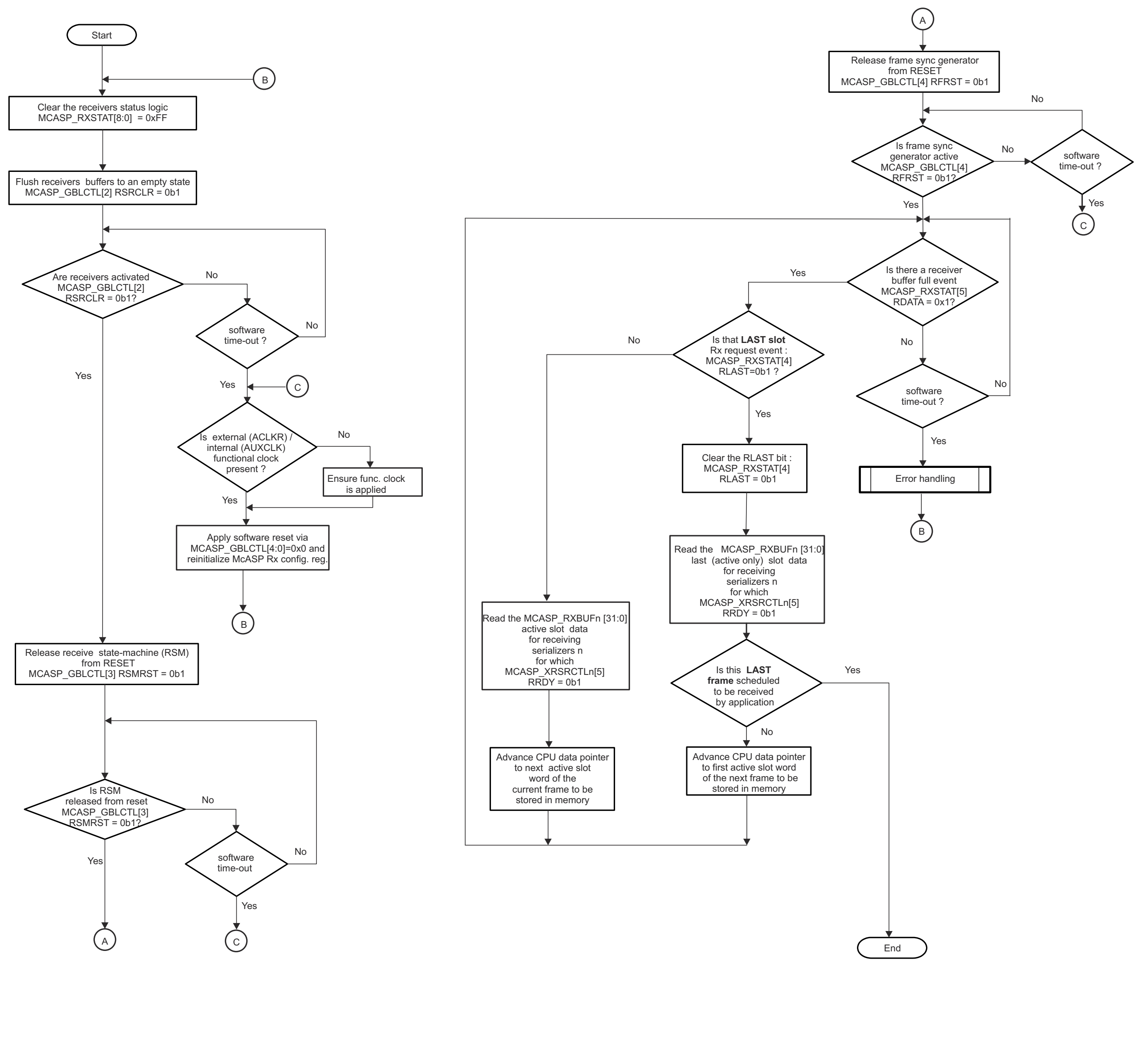


Figure 24-137. McASP Polling Reception Method

Table 24-360 summarizes the register call for the reception polling mode.

Table 24-360. Register Call Summary for Main Sequence – McASP Reception Polling Method

Register Name
MCASP_RXSTAT
MCASP_GBLCTL
MCASP_RXBUFn
MCASP_XRSRCTLn

Table 24-361 summarizes the subprocess call for the polling mode.

Table 24-361. Subprocess Call Summary for Main Sequence – McASP Reception Polling Method

Subprocess Name	Cross-Reference
Error handling	Figure 24-142

24.6.5.2.2.2 Main Sequence – McASP TDM - Interrupt Reception Method

Figure 24-138 shows the initial setup for interrupt-based reception.

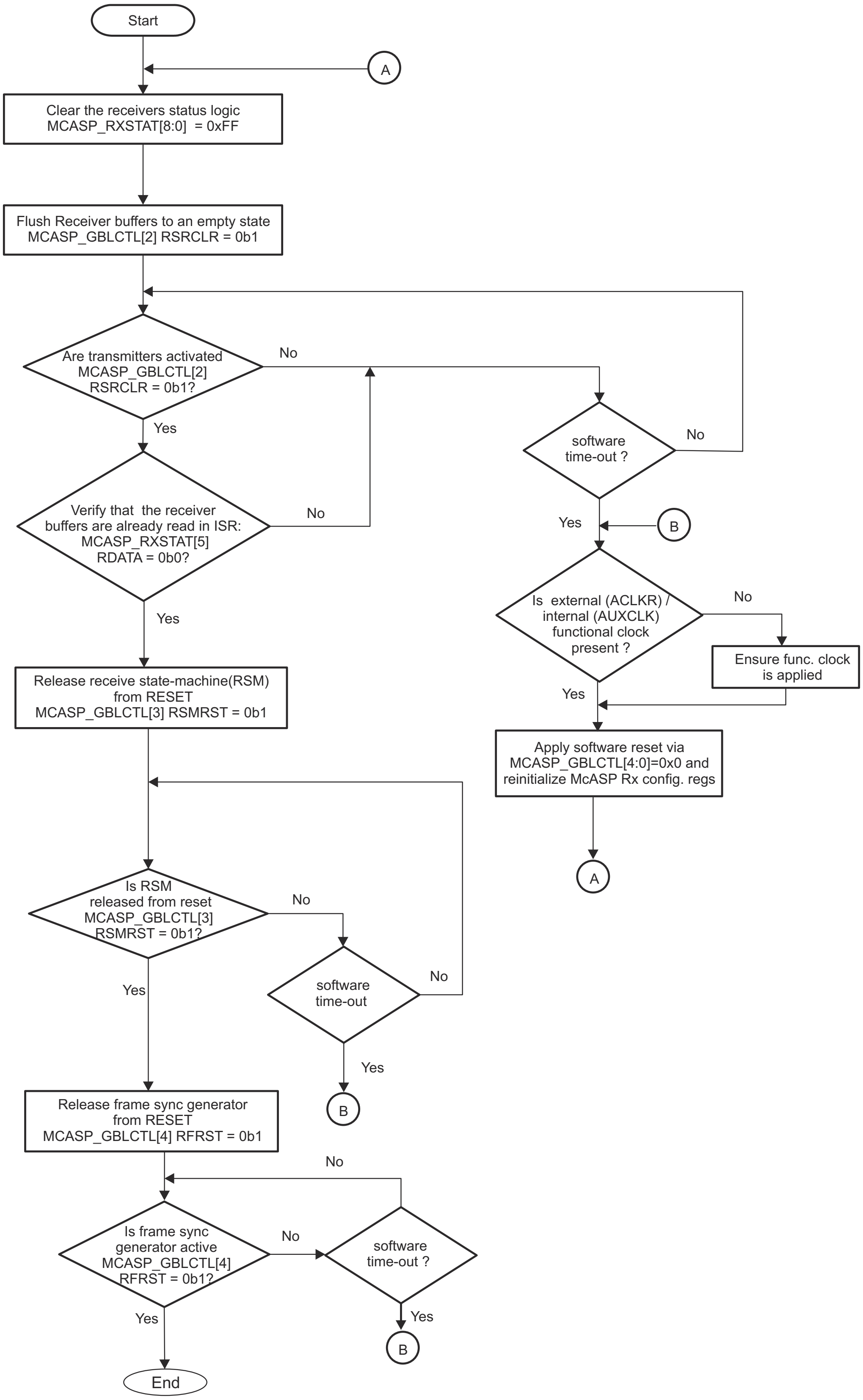


Figure 24-138. Subsequence – TDM - Reception Startup Procedure

Table 24-362 shows the configuration of the McASP using an interrupt method for TDM- reception.

Table 24-362. McASP TDM- Interrupt Reception Model

Step	Register/Bit Field/Programming Model	Value
Disable Rx DMA requests generation.	MCASP_REVTCTL[0] RDATDMA	0x1
Enable the data ready event receive interrupt.	MCASP_EVTCTLR[5] RDATA	0x1
Optional: Enable the receive error event interrupts.	MCASP_EVTCTLR[2] RCKFAIL MCASP_EVTCTLR[1] RSYNCERR MCASP_EVTCTLR[0] ROVRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt	MCASP_EVTCTLR [7] RSTAFRM MCASP_EVTCTLR[4] RLAST	0x1 0x1
IF read transfer is through the McASP DATA port (MCASP_RXFMT[3] RBUSEL is set to 0b0).	Software test condition (setting is done in step4 of the *McASP Receivers Global Initialization for TDM-Mode Operation* - see Table 24-345)	
Enable the DATA port error based interrupt.	MCASP_EVTCTLR[3] RDMAERR	0x1
ELSE		
Disable the DATA port error based interrupt.	MCASP_EVTCTLR[3] RDMAERR	0x0
ENDIF		
TDM - Transmission Startup Procedure	See Figure 24-138.	

Table 24-363 summarizes the register call to initialize the McASP to transmit using interrupt events.

Table 24-363. Register Call Summary for Subsequence – McASP TDM- Reception Startup Procedure

Register Name	Register Name
MCASP_GBLCTL	MCASP_RXSTAT

24.6.5.2.2.3 Main Sequence – McASP TDM - Mode DMA Reception Method

Table 24-364 shows the configuration of the McASP using the DMA method for reception. Possible interrupt error event servicing is also considered. Table 24-358 shows the initial setup for DMA - based transmission.

Note

Because of the DATA port burst access capability with the DMA method, it is strongly recommended that DMA transfers are initiated through the McASP DATA port.

Table 24-364. McASP DMA Reception Model with Interrupt Events Servicing

Step	Register/Bit Field/Programming Model	Value
Recommended: Select DATA port to access the transmit buffers.	MCASP_RXFMT[3] RBUSEL	0x0
Enable the Rx DMA requests generation.	MCASP_REVTCTL[0] RDATDMA	0x0
Enable the Rx DMA error event, because of McASP DATA port usage.	MCASP_EVTCTLR[3] RDMAERR	0x1
Optional: Enable the receive error event interrupts.	MCASP_EVTCTLR2] RCKFAIL MCASP_EVTCTLR[1] RSYNCERR MCASP_EVTCTLR[0] ROVRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt.	MCASP_EVTCTLR [7] RSTAFRM MCASP_EVTCTLR[4] RLAST	0x1 0x1
Disable the data ready event receive interrupt, as DMA is used to service this request.	MCASP_EVTCTLR[5] RDATA	0x0

Table 24-364. McASP DMA Reception Model with Interrupt Events Servicing (continued)

Step	Register/Bit Field/Programming Model	Value
DMA startup reception procedure. This procedure is identical than the one shown in Figure 24-138. The only difference is that DMA automatically services all the AREVT events raised by the McASP, and no CPU data processing intervention is required. The CPU is involved only in error handling shown in Figure 24-140.	See Figure 24-138.	

24.6.5.2.3 McASP Event Servicing

24.6.5.2.3.1 McASP DIT-/TDM- Transmit Interrupt Events Servicing

Figure 24-139 shows the flow of DIT-/TDM- mode transmit interrupt events servicing for the McASP module.

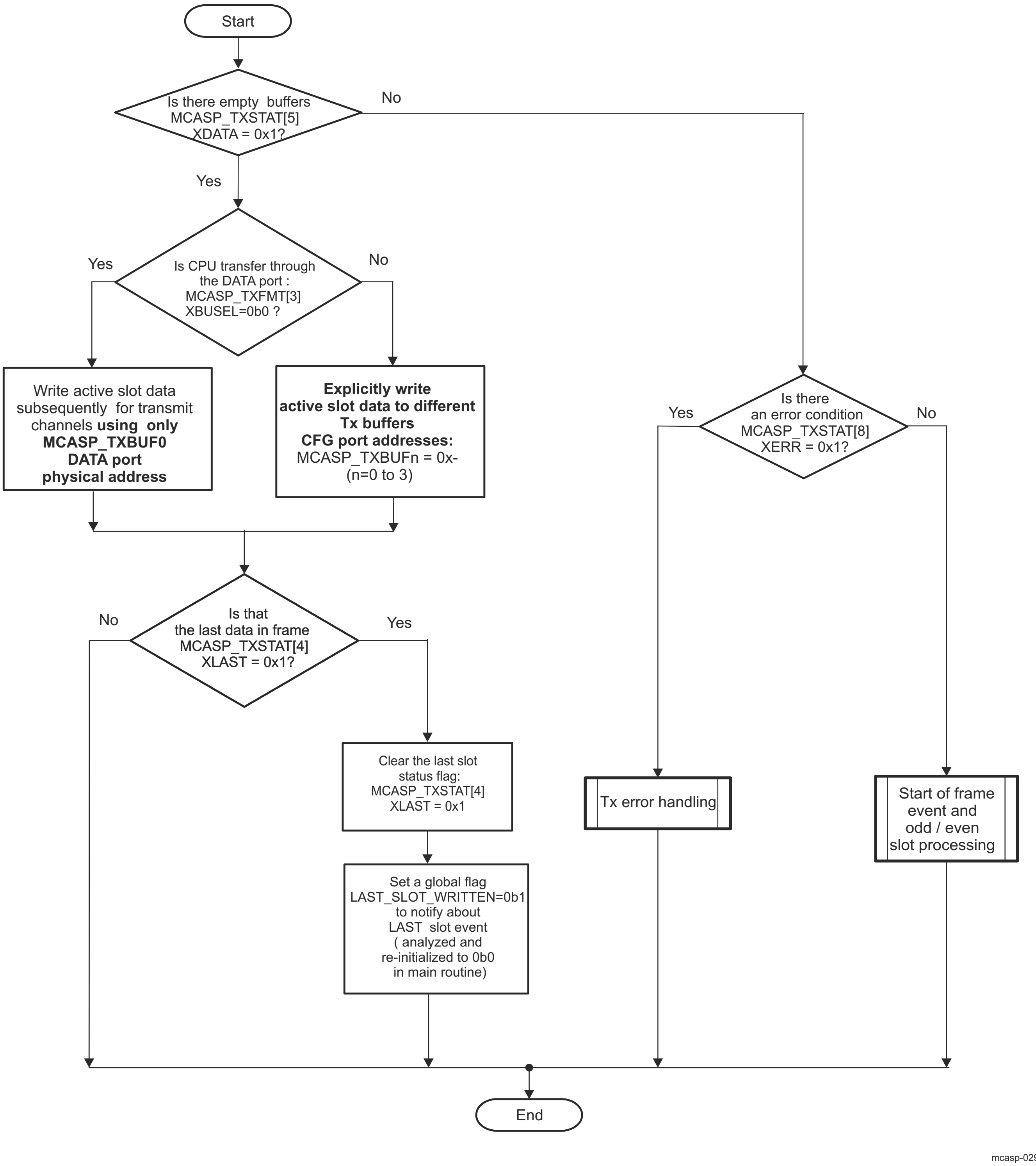


Figure 24-139. McASP Transmit Interrupt Events Servicing

24.6.5.2.3.2 McASP TDM- Receive Interrupt Events Servicing

Figure 24-140 shows the flow of DIT-/TDM- mode transmit interrupt events servicing for the McASP module.

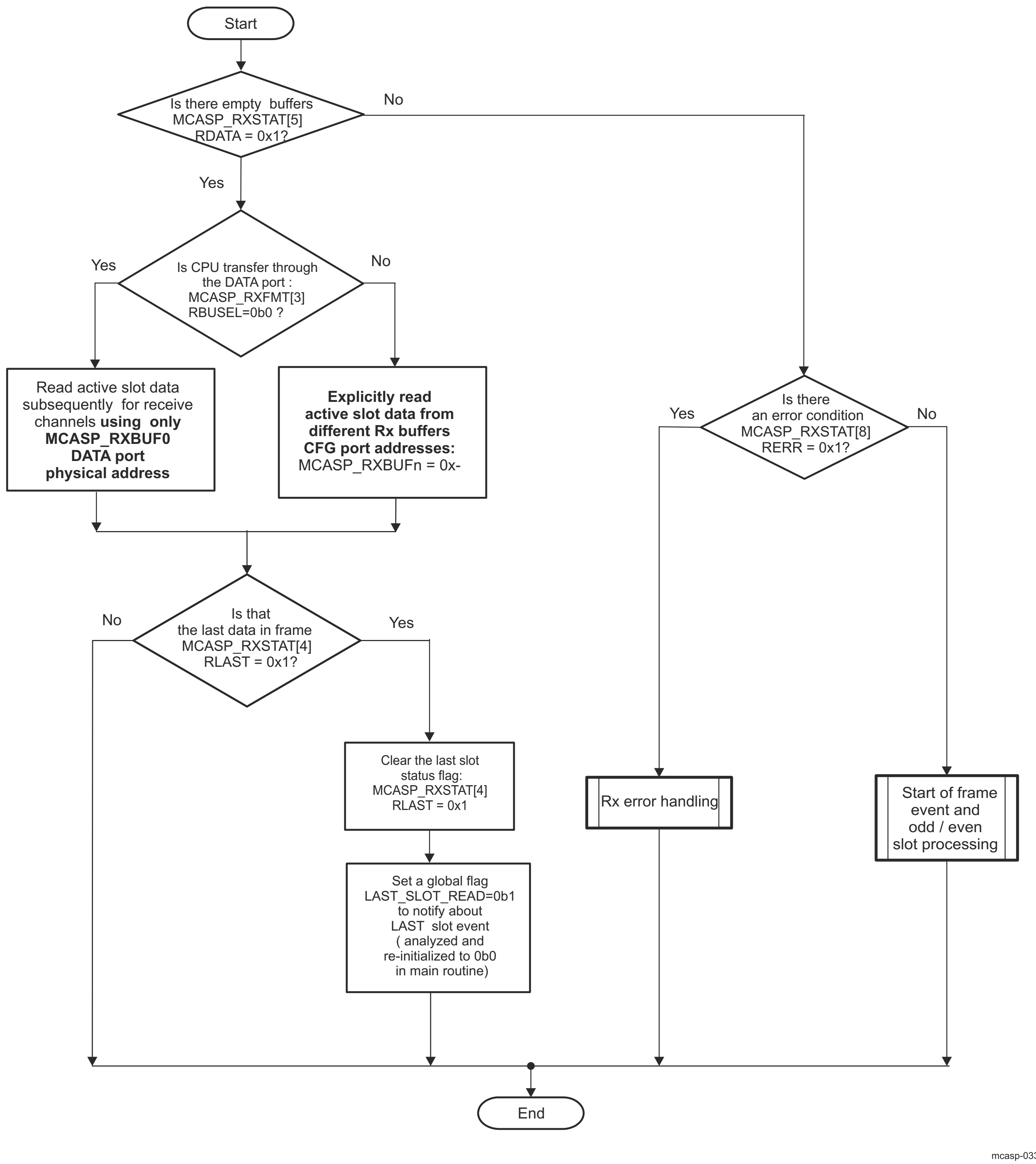


Figure 24-140. McASP Receive Interrupt Events Servicing

24.6.5.2.3.3

Table 24-365 lists the register call summary for the receive interrupt events servicing.

Table 24-365. Register Call Summary for McASP Receive Interrupt Events Servicing

Register Name	Register Name	Register Name
MCASP_RXSTAT	MCASP_RXBUFn	MCASP_RXFMT

Table 24-366 lists the subprocess call summary for receive interrupt events servicing.

Table 24-366. Subprocess Call Summary for Receive Interrupt Events Servicing

Subprocess Name	Cross-Reference
McASP receive error handling	Figure 24-142
Start of frame handling	Section 24.6.4.12.2

24.6.5.2.3.4 Subsequence – McASP DIT-/TDM -Modes Transmit Error Handling

Figure 24-141 shows the transmit error handling schema for the McASP, which can be implemented as part of the Tx interrupt service routine or as part of the Tx polling sequence.

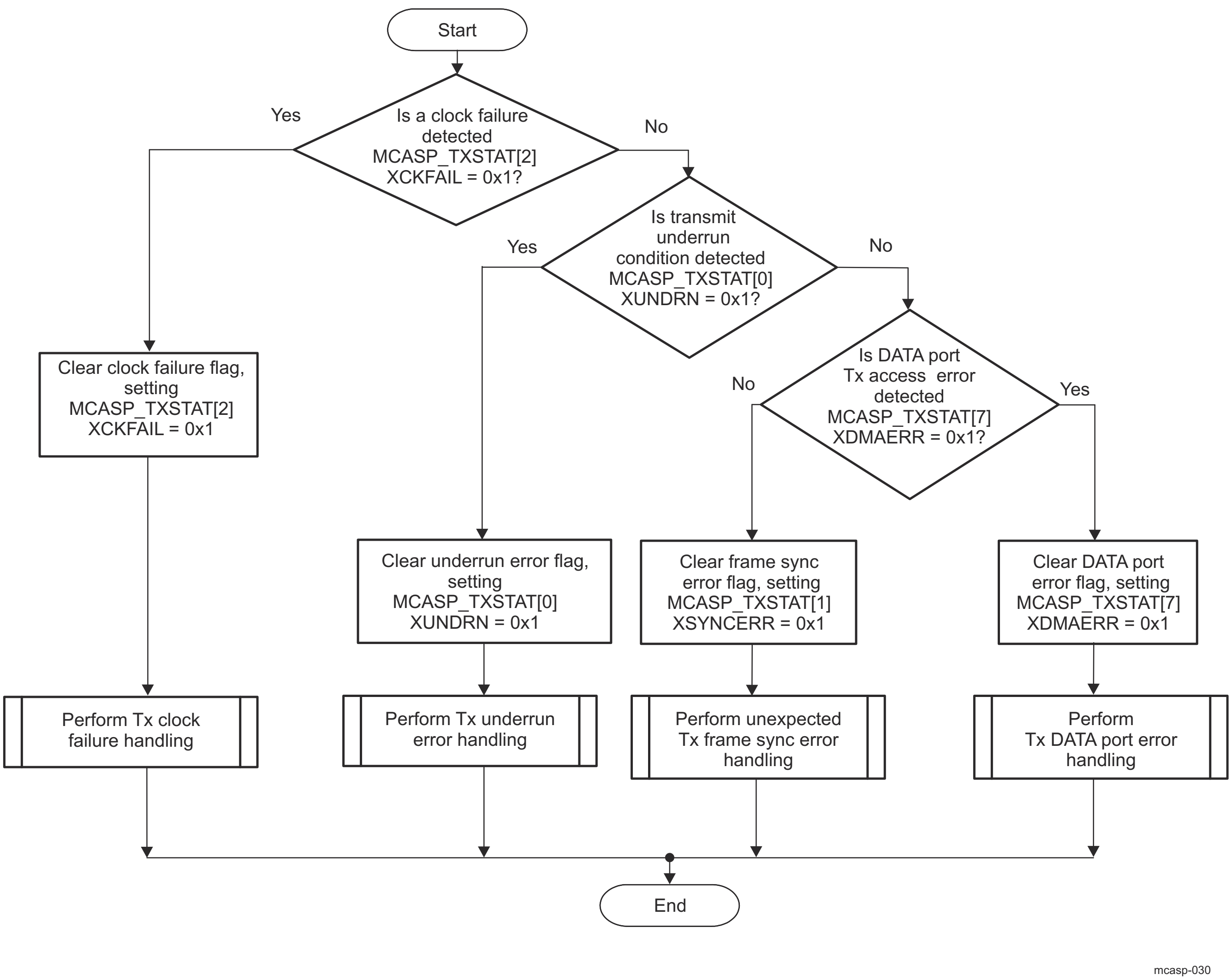


Figure 24-141. McASP Transmit Error Handling

Table 24-367 lists the register call summary for the McASP transmit error handling.

Table 24-367. Register Call Summary for McASP Transmit Error Handling

Register Name
MCASP_TXSTAT

Note

- For more information about transmit clock failure handling, see Section 24.6.4.15.6.2, *Transmit Clock Failure Check and Recovery*.
- For more information about transmit buffer underrun handling, see Section 24.6.4.15.1, *Buffer Underrun Error -Transmitter*.
- For more information about DATA port Tx error handling, see Section 24.6.4.15.3, *DATA Port Error - Transmitter*.
- For more information about unexpected Tx frame sync error handling, see Section 24.6.4.15.5, *Unexpected Frame Sync Error*.

24.6.5.2.3.5 Subsequence – McASP Receive Error Handling

Figure 24-142 shows the receive error handling schema for the McASP, which can ONLY be implemented as part of the Rx polling sequence.

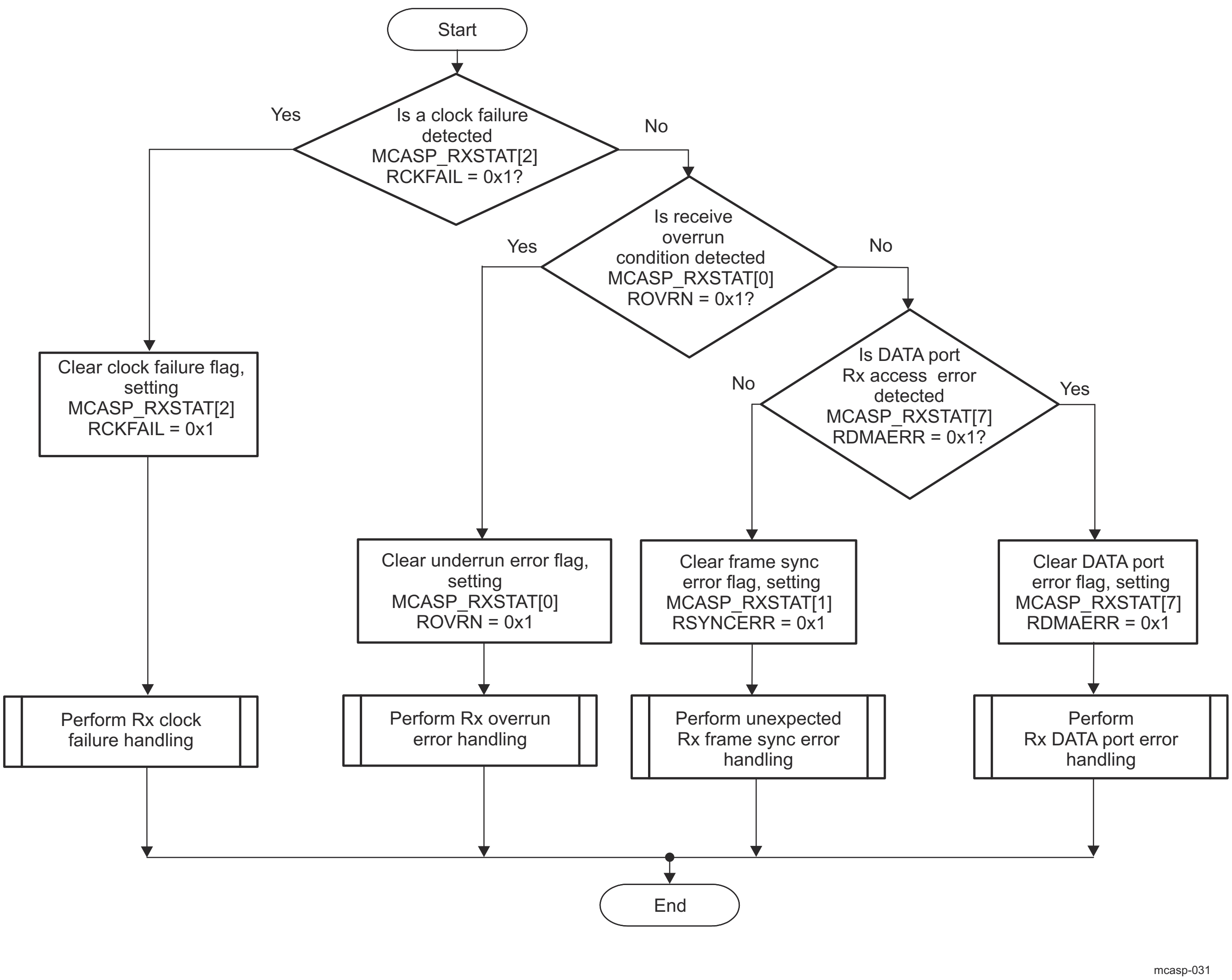


Figure 24-142. McASP Receive Error Handling

Table 24-368 lists the register call summary for the McASP receive error handling.

Table 24-368. Register Call Summary for McASP Receive Error Handling

Register Name
MCASP_RXSTAT

Note

- For more information about receive clock failure handling, see Section 24.6.4.15.6.3, *Receive Clock Failure Check and Recovery*.
- For more information about receive buffer overrun handling, see Section 24.6.4.15.2, *Buffer Overrun Error - Receiver*.
- For more information about DATA port Rx error handling, see Section 24.6.4.15.4, *DATA Port Error - Receiver*.
- For more information about unexpected Rx frame sync error handling, see Section 24.6.4.15.5, *Unexpected Frame Sync Error*.

24.6.6 McASP Register Manual

24.6.6.1 McASP Instance Summary

Table 24-369 summarizes the McASP instances.

Table 24-369. McASP Instance Summary

Module Name	Base Address L3_MAIN Interconnect	Base Address L4_PER2 Interconnect	Size
MCASP1_CFG	-	0x4846 0000	4 KiB
MCASP2_CFG	-	0x4846 4000	4 KiB
MCASP3_CFG	-	0x4846 8000	4 KiB
MCASP4_CFG	-	0x4846 C000	4 KiB
MCASP5_CFG	-	0x4847 0000	4 KiB
MCASP6_CFG	-	0x4847 4000	4 KiB
MCASP7_CFG	-	0x4847 8000	4 KiB
MCASP8_CFG	-	0x4847 C000	4 KiB
MCASP1_AFIFO	-	0x4846 1000	4 KiB
MCASP2_AFIFO	-	0x4846 5000	4 KiB
MCASP3_AFIFO	-	0x4846 9000	4 KiB
MCASP4_AFIFO	-	0x4846 D000	4 KiB
MCASP5_AFIFO	-	0x4847 1000	4 KiB
MCASP6_AFIFO	-	0x4847 5000	4 KiB
MCASP7_AFIFO	-	0x4847 9000	4 KiB
MCASP8_AFIFO	-	0x4847 D000	4 KiB
MCASP1_DAT	0x4580 0000	-	4MB
MCASP2_DAT	0x45C0 0000	-	4MB
MCASP3_DAT	0x4600 0000	-	4MB
MCASP4_DAT	-	0x4843 6000	4 KiB
MCASP5_DAT	-	0x4843 A000	4 KiB
MCASP6_DAT	-	0x4844 C000	4 KiB
MCASP7_DAT	-	0x4845 0000	4 KiB
MCASP8_DAT	-	0x4845 4000	4 KiB

24.6.6.2 McASP Registers

24.6.6.2.1 MCASP_CFG Register Summary

Table 24-370 to Table 24-373 summarize the MCASP_CFG register mapping.

Table 24-370. MCASP_CFG Register Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_CFG L4_PER2 Physical Address
MCASP_PID	R	32	0x0000 0000	0x4846 0000
PWRIDLESYSCONFIG	RW	32	0x0000 0004	0x4846 0004
MCASP_PFUNC	RW	32	0x0000 0010	0x4846 0010
MCASP_PDIR	RW	32	0x0000 0014	0x4846 0014
MCASP_PDOUT	RW	32	0x0000 0018	0x4846 0018
MCASP_PDIN	R	32	0x0000 001C	0x4846 001C
MCASP_PDSET	W	32	0x0000 001C	0x4846 001C
MCASP_PDCLR	RW	32	0x0000 0020	0x4846 0020
RESERVED	RW	32	0x0000 0030	0x4846 0030
RESERVED	RW	32	0x0000 0034	0x4846 0034
RESERVED	RW	32	0x0000 0038	0x4846 0038

Table 24-370. MCASP_CFG Register Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_CFG L4_PER2 Physical Address
MCASP_GBLCTL	RW	32	0x0000 0044	0x4846 0044
MCASP_AMUTE	RW	32	0x0000 0048	0x4846 0048
MCASP_LBCTL	RW	32	0x0000 004C	0x4846 004C
MCASP_TXDITCTL	RW	32	0x0000 0050	0x4846 0050
MCASP_GBLCTLR	RW	32	0x0000 0060	0x4846 0060
MCASP_RXMASK	RW	32	0x0000 0064	0x4846 0064
MCASP_RXFMT	RW	32	0x0000 0068	0x4846 0068
MCASP_RXFMCTL	RW	32	0x0000 006C	0x4846 006C
MCASP_ACLKRCTL	RW	32	0x0000 0070	0x4846 0070
MCASP_AHCLKRCTL	RW	32	0x0000 0074	0x4846 0074
MCASP_RXTDM	RW	32	0x0000 0078	0x4846 0078
MCASP_EVTCTLR	RW	32	0x0000 007C	0x4846 007C
MCASP_RXSTAT	RW	32	0x0000 0080	0x4846 0080
MCASP_RXTDMSLOT	R	32	0x0000 0084	0x4846 0084
MCASP_RXCLKCHK	RW	32	0x0000 0088	0x4846 0088
MCASP_REVTCTL	RW	32	0x0000 008C	0x4846 008C
MCASP_GBLCTLX	RW	32	0x0000 00A0	0x4846 00A0
MCASP_TXMASK	RW	32	0x0000 00A4	0x4846 00A4
MCASP_TXFMT	RW	32	0x0000 00A8	0x4846 00A8
MCASP_TXFMCTL	RW	32	0x0000 00AC	0x4846 00AC
MCASP_ACLKXCTL	RW	32	0x0000 00B0	0x4846 00B0
MCASP_AHCLKXCTL	RW	32	0x0000 00B4	0x4846 00B4
MCASP_TXTDM	RW	32	0x0000 00B8	0x4846 00B8
MCASP_EVTCTLX	RW	32	0x0000 00BC	0x4846 00BC
MCASP_TXSTAT	RW	32	0x0000 00C0	0x4846 00C0
MCASP_TXTDMSLOT	R	32	0x0000 00C4	0x4846 00C4
MCASP_TXCLKCHK	RW	32	0x0000 00C8	0x4846 00C8
MCASP_XEVTCTL	RW	32	0x0000 00CC	0x4846 00CC
MCASP_CLKADJEN	RW	32	0x0000 00D0	0x4846 00D0
MCASP_DITCSRAi (1)	RW	32	0x0000 0100 + (0x4*i)	0x4846 0100 + (0x4*i)
MCASP_DITCSRBi (1)	RW	32	0x0000 0118 + (0x4*i)	0x4846 0118 + (0x4*i)
MCASP_DITUDRAi (1)	RW	32	0x0000 0130 + (0x4*i)	0x4846 0130 + (0x4*i)
MCASP_DITUDRBi (1)	RW	32	0x0000 0148 + (0x4*i)	0x4846 0148 + (0x4*i)
MCASP_XRSRCTLn (2)	RW	32	0x0000 0180 + (0x4*n)	0x4846 0180 + (0x4*n)
MCASP_TXBUFn (2)	RW	32	0x0000 0200 + (0x4*n)	0x4846 0200 + (0x4*n)
MCASP_RXBUFn (2)	RW	32	0x0000 0280 + (0x4*n)	0x4846 0280 + (0x4*n)

(1) i = 0 to 5
(2) n = 0 to 15

Table 24-371. MCASP_CFG Register Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_CFG L4_PER2 Physical Address
MCASP_PID	R	32	0x0000 0000	0x4846 4000
PWRIDLESYSCONFIG	RW	32	0x0000 0004	0x4846 4004
MCASP_PFUNC	RW	32	0x0000 0010	0x4846 4010
MCASP_PDIR	RW	32	0x0000 0014	0x4846 4014

Table 24-371. MCASP_CFG Register Summary 2 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_CFG L4_PER2 Physical Address
MCASP_PDOUT	RW	32	0x0000 0018	0x4846 4018
MCASP_PDIN	R	32	0x0000 001C	0x4846 401C
MCASP_PDSET	W	32	0x0000 001C	0x4846 401C
MCASP_PDCLR	RW	32	0x0000 0020	0x4846 4020
RESERVED	RW	32	0x0000 0030	0x4846 4030
RESERVED	RW	32	0x0000 0034	0x4846 4034
RESERVED	RW	32	0x0000 0038	0x4846 4038
MCASP_GBLCTL	RW	32	0x0000 0044	0x4846 4044
MCASP_AMUTE	RW	32	0x0000 0048	0x4846 4048
MCASP_LBCTL	RW	32	0x0000 004C	0x4846 404C
MCASP_TXDITCTL	RW	32	0x0000 0050	0x4846 4050
MCASP_GBLCTLR	RW	32	0x0000 0060	0x4846 4060
MCASP_RXMASK	RW	32	0x0000 0064	0x4846 4064
MCASP_RXFMT	RW	32	0x0000 0068	0x4846 4068
MCASP_RXFMCTL	RW	32	0x0000 006C	0x4846 406C
MCASP_ACLKRCTL	RW	32	0x0000 0070	0x4846 4070
MCASP_AHCLKRCTL	RW	32	0x0000 0074	0x4846 4074
MCASP_RXTDM	RW	32	0x0000 0078	0x4846 4078
MCASP_EVTCTLR	RW	32	0x0000 007C	0x4846 407C
MCASP_RXSTAT	RW	32	0x0000 0080	0x4846 4080
MCASP_RXTDMSLOT	R	32	0x0000 0084	0x4846 4084
MCASP_RXCLKCHK	RW	32	0x0000 0088	0x4846 4088
MCASP_REVTCTL	RW	32	0x0000 008C	0x4846 408C
MCASP_GBLCTLX	RW	32	0x0000 00A0	0x4846 40A0
MCASP_TXMASK	RW	32	0x0000 00A4	0x4846 40A4
MCASP_TXFMT	RW	32	0x0000 00A8	0x4846 40A8
MCASP_TXFMCTL	RW	32	0x0000 00AC	0x4846 40AC
MCASP_ACLKXCTL	RW	32	0x0000 00B0	0x4846 40B0
MCASP_AHCLKXCTL	RW	32	0x0000 00B4	0x4846 40B4
MCASP_TXTDM	RW	32	0x0000 00B8	0x4846 40B8
MCASP_EVTCTLX	RW	32	0x0000 00BC	0x4846 40BC
MCASP_TXSTAT	RW	32	0x0000 00C0	0x4846 40C0
MCASP_TXTDMSLOT	R	32	0x0000 00C4	0x4846 40C4
MCASP_TXCLKCHK	RW	32	0x0000 00C8	0x4846 40C8
MCASP_XEVTCTL	RW	32	0x0000 00CC	0x4846 40CC
MCASP_CLKADJEN	RW	32	0x0000 00D0	0x4846 40D0
MCASP_DITCSRAi (1)	RW	32	0x0000 0100 + (0x4*i)	0x4846 4100 + (0x4*i)
MCASP_DITCSRBi (1)	RW	32	0x0000 0118 + (0x4*i)	0x4846 4118 + (0x4*i)
MCASP_DITUDRAi (1)	RW	32	0x0000 0130 + (0x4*i)	0x4846 4130 + (0x4*i)
MCASP_DITUDRBi (1)	RW	32	0x0000 0148 + (0x4*i)	0x4846 4148 + (0x4*i)
MCASP_XRSRCTLn (2)	RW	32	0x0000 0180 + (0x4*n)	0x4846 4180 + (0x4*n)
MCASP_TXBUFn (2)	RW	32	0x0000 0200 + (0x4*n)	0x4846 4200 + (0x4*n)
MCASP_RXBUFn (2)	RW	32	0x0000 0280 + (0x4*n)	0x4846 4280 + (0x4*n)

(1) i = 0 to 5
(2) n = 0 to 15

Table 24-372. MCASP_CFG Register Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	MCASP3_CFG L4_PER2 Physical Address	MCASP4_CFG L4_PER2 Physical Address	MCASP5_CFG L4_PER2 Physical Address
MCASP_PID	R	32	0x0000 0000	0x4846 8000	0x4846 C000	0x4847 0000
PWRIDLESYSCONFIG	RW	32	0x0000 0004	0x4846 8004	0x4846 C004	0x4847 0004
MCASP_PFUNC	RW	32	0x0000 0010	0x4846 8010	0x4846 C010	0x4847 0010
MCASP_PDIR	RW	32	0x0000 0014	0x4846 8014	0x4846 C014	0x4847 0014
MCASP_PDOUT	RW	32	0x0000 0018	0x4846 8018	0x4846 C018	0x4847 0018
MCASP_PDIN	R	32	0x0000 001C	0x4846 801C	0x4846 C01C	0x4847 001C
MCASP_PDSET	W	32	0x0000 001C	0x4846 801C	0x4846 C01C	0x4847 001C
MCASP_PDCLR	RW	32	0x0000 0020	0x4846 8020	0x4846 C020	0x4847 0020
RESERVED	RW	32	0x0000 0030	0x4846 8030	0x4846 C030	0x4847 0030
RESERVED	RW	32	0x0000 0034	0x4846 8034	0x4846 C034	0x4847 0034
RESERVED	RW	32	0x0000 0038	0x4846 8038	0x4846 C038	0x4847 0038
MCASP_GBLCTL	RW	32	0x0000 0044	0x4846 8044	0x4846 C044	0x4847 0044
MCASP_AMUTE	RW	32	0x0000 0048	0x4846 8048	0x4846 C048	0x4847 0048
MCASP_LBCTL	RW	32	0x0000 004C	0x4846 804C	0x4846 C04C	0x4847 004C
MCASP_TXDITCTL	RW	32	0x0000 0050	0x4846 8050	0x4846 C050	0x4847 0050
MCASP_GBLCTLR	RW	32	0x0000 0060	0x4846 8060	0x4846 C060	0x4847 0060
MCASP_RXMASK	RW	32	0x0000 0064	0x4846 8064	0x4846 C064	0x4847 0064
MCASP_RXFMT	RW	32	0x0000 0068	0x4846 8068	0x4846 C068	0x4847 0068
MCASP_RXFMCTL	RW	32	0x0000 006C	0x4846 806C	0x4846 C06C	0x4847 006C
MCASP_ACLKRCTL	RW	32	0x0000 0070	0x4846 8070	0x4846 C070	0x4847 0070
MCASP_AHCLKRCTL	RW	32	0x0000 0074	0x4846 8074	0x4846 C074	0x4847 0074
MCASP_RXTDM	RW	32	0x0000 0078	0x4846 8078	0x4846 C078	0x4847 0078
MCASP_EVTCTLR	RW	32	0x0000 007C	0x4846 807C	0x4846 C07C	0x4847 007C
MCASP_RXSTAT	RW	32	0x0000 0080	0x4846 8080	0x4846 C080	0x4847 0080
MCASP_RXTDMSLOT	R	32	0x0000 0084	0x4846 8084	0x4846 C084	0x4847 0084
MCASP_RXCLKCHK	RW	32	0x0000 0088	0x4846 8088	0x4846 C088	0x4847 0088
MCASP_REVTCTL	RW	32	0x0000 008C	0x4846 808C	0x4846 C08C	0x4847 008C
MCASP_GBLCTLX	RW	32	0x0000 00A0	0x4846 80A0	0x4846 C0A0	0x4847 00A0
MCASP_TXMASK	RW	32	0x0000 00A4	0x4846 80A4	0x4846 C0A4	0x4847 00A4
MCASP_TXFMT	RW	32	0x0000 00A8	0x4846 80A8	0x4846 C0A8	0x4847 00A8
MCASP_TXFMCTL	RW	32	0x0000 00AC	0x4846 80AC	0x4846 C0AC	0x4847 00AC
MCASP_ACLKXCTL	RW	32	0x0000 00B0	0x4846 80B0	0x4846 C0B0	0x4847 00B0
MCASP_AHCLKXCTL	RW	32	0x0000 00B4	0x4846 80B4	0x4846 C0B4	0x4847 00B4
MCASP_TXTDM	RW	32	0x0000 00B8	0x4846 80B8	0x4846 C0B8	0x4847 00B8
MCASP_EVTCTLX	RW	32	0x0000 00BC	0x4846 80BC	0x4846 C0BC	0x4847 00BC
MCASP_TXSTAT	RW	32	0x0000 00C0	0x4846 80C0	0x4846 C0C0	0x4847 00C0
MCASP_TXTDMSLOT	R	32	0x0000 00C4	0x4846 80C4	0x4846 C0C4	0x4847 00C4
MCASP_TXCLKCHK	RW	32	0x0000 00C8	0x4846 80C8	0x4846 C0C8	0x4847 00C8
MCASP_XEVTCTL	RW	32	0x0000 00CC	0x4846 80CC	0x4846 C0CC	0x4847 00CC
MCASP_CLKADJEN	RW	32	0x0000 00D0	0x4846 80D0	0x4846 C0D0	0x4847 00D0
MCASP_DITCSRAi	RW	32	0x0000 0100 + (0x4*i)	0x4846 8100 + (0x4*i)	0x4846 C100 + (0x4*i)	0x4847 0100 + (0x4*i)
MCASP_DITCSRBi	RW	32	0x0000 0118 + (0x4*i)	0x4846 8118 + (0x4*i)	0x4846 C118 + (0x4*i)	0x4847 0118 + (0x4*i)
MCASP_DITUDRAi	RW	32	0x0000 0130 + (0x4*i)	0x4846 8130 + (0x4*i)	0x4846 C130 + (0x4*i)	0x4847 0130 + (0x4*i)

Table 24-372. MCASP_CFG Register Summary 3 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP3_CFG L4_PER2 Physical Address	MCASP4_CFG L4_PER2 Physical Address	MCASP5_CFG L4_PER2 Physical Address
MCASP_DITUDRBi	RW	32	0x0000 0148 + (0x4*i)	0x4846 8148 + (0x4*i)	0x4846 C148 + (0x4*i)	0x4847 0148 + (0x4*i)
MCASP_XRSRCTLn	RW	32	0x0000 0180 + (0x4*n)	0x4846 8180 + (0x4*n)	0x4846 C180 + (0x4*n)	0x4847 0180 + (0x4*n)
MCASP_TXBUFn	RW	32	0x0000 0200 + (0x4*n)	0x4846 8200 + (0x4*n)	0x4846 C200 + (0x4*n)	0x4847 0200 + (0x4*n)
MCASP_RXBUFn	RW	32	0x0000 0280 + (0x4*n)	0x4846 8280 + (0x4*n)	0x4846 C280 + (0x4*n)	0x4847 0280 + (0x4*n)

Table 24-373. MCASP_CFG Register Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	MCASP6_CFG L4_PER2 Physical Address	MCASP7_CFG L4_PER2 Physical Address	MCASP8_CFG L4_PER2 Physical Address
MCASP_PID	R	32	0x0000 0000	0x4847 4000	0x4847 8000	0x4847 C000
PWRIDLESYSCONFIG	RW	32	0x0000 0004	0x4847 4004	0x4847 8004	0x4847 C004
MCASP_PFUNC	RW	32	0x0000 0010	0x4847 4010	0x4847 8010	0x4847 C010
MCASP_PDIR	RW	32	0x0000 0014	0x4847 4014	0x4847 8014	0x4847 C014
MCASP_PDOUT	RW	32	0x0000 0018	0x4847 4018	0x4847 8018	0x4847 C018
MCASP_PDIN	R	32	0x0000 001C	0x4847 401C	0x4847 801C	0x4847 C01C
MCASP_PDSET	W	32	0x0000 001C	0x4847 401C	0x4847 801C	0x4847 C01C
MCASP_PDCLR	RW	32	0x0000 0020	0x4847 4020	0x4847 8020	0x4847 C020
RESERVED	RW	32	0x0000 0030	0x4847 4030	0x4847 8030	0x4847 C030
RESERVED	RW	32	0x0000 0034	0x4847 4034	0x4847 8034	0x4847 C034
RESERVED	RW	32	0x0000 0038	0x4847 4038	0x4847 8038	0x4847 C038
MCASP_GBLCTL	RW	32	0x0000 0044	0x4847 4044	0x4847 8044	0x4847 C044
MCASP_AMUTE	RW	32	0x0000 0048	0x4847 4048	0x4847 8048	0x4847 C048
MCASP_LBCTL	RW	32	0x0000 004C	0x4847 404C	0x4847 804C	0x4847 C04C
MCASP_TXDITCTL	RW	32	0x0000 0050	0x4847 4050	0x4847 8050	0x4847 C050
MCASP_GBLCTLR	RW	32	0x0000 0060	0x4847 4060	0x4847 8060	0x4847 C060
MCASP_RXMASK	RW	32	0x0000 0064	0x4847 4064	0x4847 8064	0x4847 C064
MCASP_RXFMT	RW	32	0x0000 0068	0x4847 4068	0x4847 8068	0x4847 C068
MCASP_RXFMCTL	RW	32	0x0000 006C	0x4847 406C	0x4847 806C	0x4847 C06C
MCASP_ACLKRCTL	RW	32	0x0000 0070	0x4847 4070	0x4847 8070	0x4847 C070
MCASP_AHCLKRCTL	RW	32	0x0000 0074	0x4847 4074	0x4847 8074	0x4847 C074
MCASP_RXTDM	RW	32	0x0000 0078	0x4847 4078	0x4847 8078	0x4847 C078
MCASP_EVTCTLR	RW	32	0x0000 007C	0x4847 407C	0x4847 807C	0x4847 C07C
MCASP_RXSTAT	RW	32	0x0000 0080	0x4847 4080	0x4847 8080	0x4847 C080
MCASP_RXTDMSLOT	R	32	0x0000 0084	0x4847 4084	0x4847 8084	0x4847 C084
MCASP_RXCLKCHK	RW	32	0x0000 0088	0x4847 4088	0x4847 8088	0x4847 C088
MCASP_REVTCTL	RW	32	0x0000 008C	0x4847 408C	0x4847 808C	0x4847 C08C
MCASP_GBLCTLX	RW	32	0x0000 00A0	0x4847 40A0	0x4847 80A0	0x4847 C0A0
MCASP_TXMASK	RW	32	0x0000 00A4	0x4847 40A4	0x4847 80A4	0x4847 C0A4
MCASP_TXFMT	RW	32	0x0000 00A8	0x4847 40A8	0x4847 80A8	0x4847 C0A8
MCASP_TXFMCTL	RW	32	0x0000 00AC	0x4847 40AC	0x4847 80AC	0x4847 C0AC
MCASP_ACLKXCTL	RW	32	0x0000 00B0	0x4847 40B0	0x4847 80B0	0x4847 C0B0
MCASP_AHCLKXCTL	RW	32	0x0000 00B4	0x4847 40B4	0x4847 80B4	0x4847 C0B4
MCASP_TXTDM	RW	32	0x0000 00B8	0x4847 40B8	0x4847 80B8	0x4847 C0B8

Table 24-373. MCASP_CFG Register Summary 4 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP6_CFG L4_PER2 Physical Address	MCASP7_CFG L4_PER2 Physical Address	MCASP8_CFG L4_PER2 Physical Address
MCASP_EVTCTLX	RW	32	0x0000 00BC	0x4847 40BC	0x4847 80BC	0x4847 C0BC
MCASP_TXSTAT	RW	32	0x0000 00C0	0x4847 40C0	0x4847 80C0	0x4847 C0C0
MCASP_TXTDMSLOT	R	32	0x0000 00C4	0x4847 40C4	0x4847 80C4	0x4847 C0C4
MCASP_TXCLKCHK	RW	32	0x0000 00C8	0x4847 40C8	0x4847 80C8	0x4847 C0C8
MCASP_XEVTCTL	RW	32	0x0000 00CC	0x4847 40CC	0x4847 80CC	0x4847 C0CC
MCASP_CLKADJEN	RW	32	0x0000 00D0	0x4847 40D0	0x4847 80D0	0x4847 C0D0
MCASP_DITCSRAi (1)	RW	32	0x0000 0100 + (0x4*i)	0x4847 4100 + (0x4*i)	0x4847 8100 + (0x4*i)	0x4847 C100 + (0x4*i)
MCASP_DITCSRBi (1)	RW	32	0x0000 0118 + (0x4*i)	0x4847 4118 + (0x4*i)	0x4847 8118 + (0x4*i)	0x4847 C118 + (0x4*i)
MCASP_DITUDRAi (1)	RW	32	0x0000 0130 + (0x4*i)	0x4847 4130 + (0x4*i)	0x4847 8130 + (0x4*i)	0x4847 C130 + (0x4*i)
MCASP_DITUDRBi (1)	RW	32	0x0000 0148 + (0x4*i)	0x4847 4148 + (0x4*i)	0x4847 8148 + (0x4*i)	0x4847 C148 + (0x4*i)
MCASP_XRSRCTLn (2)	RW	32	0x0000 0180 + (0x4*n)	0x4847 4180 + (0x4*n)	0x4847 8180 + (0x4*n)	0x4847 C180 + (0x4*n)
MCASP_TXBUFn (2)	RW	32	0x0000 0200 + (0x4*n)	0x4847 4200 + (0x4*n)	0x4847 8200 + (0x4*n)	0x4847 C200 + (0x4*n)
MCASP_RXBUFn (2)	RW	32	0x0000 0280 + (0x4*n)	0x4847 4280 + (0x4*n)	0x4847 8280 + (0x4*n)	0x4847 C280 + (0x4*n)

(1) i = 0 to 5
(2) n = 0 to 15

Note

The address locations listed in Table 24-370 to Table 24-373 , *MCASP_CFG Register Mapping Summary*, are relevant for accessing:

- All McASP configuration registers
- MCASP_TXBUFn registers
- MCASP_RXBUFn registers

through the McASP peripheral configuration (CFG) port.

The MCASP_TXFMT[3] XBUSEL bit must be set to 0b1, to allow CFG port write accesses to the McASP XRBUFn buffer. The MCASP_RXFMT[3] RBUSEL bit must be set to 0b1, to allow CFG port read accesses to the McASP XRBUFn buffer.

24.6.6.2.2 MCASP_CFG Register Description

The tables below describe the individual MCASP_CFG register bits.

Note

For all of the below described registers the indexes n and N are applying to serializers (not slots).

Register descriptions cover the superset McASP (16 serializers and all signals pinned out). For the particular McASP instantiation, please refer to *McASP Environment*, and *McASP Integration*.

Table 24-374. MCASP_PID

Address Offset	0x0000 0000

Table 24-374. MCASP_PID (continued)

Physical Address	0x4846 0000	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Peripheral identification register		
Type	R		

31:30	29:28	27:16	15:11	10:8	7:6	5:0
SCHEME	RESV	FUNCTION	RTL	REVMAJOR	CUSTOM	REVMINOR

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	Scheme. Distinguishes between old scheme and current.	R	0x1
29:28	RESV	Reserved.	R	0x0
27:16	FUNCTION	McASP. Indicates a software-compatible module family.	R	0x430
15:11	RTL	RTL version.	R	0x1
10:8	REVMAJOR	Major revision number.	R	0x0
7:6	CUSTOM	Non-custom. Indicates a special version for a given device.	R	0x0
5:0	REVMINOR	Minor revision number.	R	0x0

Table 24-375. PWRIDLESYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x4846 0004	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Power idle module configuration register.		
Type	RW		

31:6	5:2	1:0
RESERVED	OTHER	IDLE_MODE

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0000000
5:2	OTHER	Reserved for future expansion	RW	0x0
1:0	IDLE_MODE	0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode - default state 0x3: Reserved	RW	0x2

Table 24-376. MCASP_PFUNC

Address Offset	0x0000 0010

Table 24-376. MCASP_PFUNC (continued)

Physical Address	0x4846 0010	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Specifies the function of the pins as either a McASP pin or a GPIO pin. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Determines if AFSR pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
30	RESERVED	Reserved	RW	0
29	ACLKR	Determines if ACLKR pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
28	AFSX	Determines if AFSX pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
27	AHCLKX	Determines if AHCLKX pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
26	ACLKX	Determines if ACLKX pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Determines if AXR15 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
14	AXR14	Determines if AXR14 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
13	AXR13	Determines if AXR13 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
12	AXR12	Determines if AXR12 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0

Bits	Field Name	Description	Type	Reset
11	AXR11	Determines if AXR11 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
10	AXR10	Determines if AXR10 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
9	AXR9	Determines if AXR9 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
8	AXR8	Determines if AXR8 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
7	AXR7	Determines if AXR7 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
6	AXR6	Determines if AXR6 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
5	AXR5	Determines if AXR5 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
4	AXR4	Determines if AXR4 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
3	AXR3	Determines if AXR3 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
2	AXR2	Determines if AXR2 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
1	AXR1	Determines if AXR1 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
0	AXR0	Determines if AXR0 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0

Table 24-377. MCASP_PDIR

Address Offset	0x0000 0014		
Physical Address	0x4846 0014	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Pin direction register - specifies the direction of the McASP pins as either an input or an output pin.		

Table 24-377. MCASP_PDIR (continued)

Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Determines if AFSR pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
30	RESERVED	Reserved	RW	0
29	ACLKR	Determines if ACLKR pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
28	AFSX	Determines if AFSX pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
27	AHCLKX	Determines if AHCLKX pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
26	ACLKX	Determines if ACLKX pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Determines if AXR15 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
14	AXR14	Determines if AXR14 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
13	AXR13	Determines if AXR13 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
12	AXR12	Determines if AXR12 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
11	AXR11	Determines if AXR11 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
10	AXR10	Determines if AXR10 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0

Bits	Field Name	Description	Type	Reset
9	AXR9	Determines if AXR9 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
8	AXR8	Determines if AXR8 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
7	AXR7	Determines if AXR7 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
6	AXR6	Determines if AXR6 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
5	AXR5	Determines if AXR5 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
4	AXR4	Determines if AXR4 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
3	AXR3	Determines if AXR3 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
2	AXR2	Determines if AXR2 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
1	AXR1	Determines if AXR1 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
0	AXR0	Determines if AXR0 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0

Table 24-378. MCASP_PDOUT

Address Offset	0x0000 0018		
Physical Address	0x4846 0018	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Pin data output register - holds a value for data out at all times, and may be read back at all times. The value held by MCASP_PDOUT is not affected by writing to MCASP_PDIR and MCASP_PFUNC. However, the data value in MCASP_PDOUT is driven out onto the McASP pin only if the corresponding bit in MCASP_PFUNC is set to 1 (GPIO function) and the corresponding bit in MCASP_PDIR is set to 1 (output). When reading data, it returns the corresponding bit value in MCASP_PDOUT[n]; it does not return the input from the I/O pin. When writing data, writes to the corresponding MCASP_PDOUT[n] bit.		

Table 24-378. MCASP_PDOUT (continued)

PDOUT has these aliases or alternate addresses:

- MCASP_PDSET - when written to at this address, writing a 1 to a bit in MCASP_PDSET sets the corresponding bit in MCASP_PDOUT to 1; writing a 0 has no effect and keeps the bits in MCASP_PDOUT unchanged.
- MCASP_PDCLR - when written to at this address, writing a 1 to a bit in MCASP_PDCLR clears the corresponding bit in MCASP_PDOUT to 0; writing a 0 has no effect and keeps the bits in MCASP_PDOUT unchanged

There is only one set of data-out bits, MCASP_PDOUT[31:0]. The other registers, MCASP_PDSET and MCASP_PDCLR, are just different addresses for the same control bits, with different behaviors during writes.

Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Determines drive on AFSR output pin when the corresponding MCASP_PFUNC[31] and MCASP_PDIR[31] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
30	AHCLKR	Determines drive on AHCLKR output pin when the corresponding MCASP_PFUNC[30] and MCASP_PDIR[30] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
29	ACLKR	Determines drive on ACLKR output pin when the corresponding MCASP_PFUNC[29] and MCASP_PDIR[29] bits are set to 1 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
28	AFSX	Determines drive on AFSX output pin when the corresponding MCASP_PFUNC[28] and MCASP_PDIR[28] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
27	AHCLKX	Determines drive on AHCLKX output pin when the corresponding MCASP_PFUNC[27] and MCASP_PDIR[27] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
26	ACLKX	Determines drive on ACLKX output pin when the corresponding MCASP_PFUNC[26] and MCASP_PDIR[26] bits are set to 1 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Determines drive on AXR15 output pin when the corresponding MCASP_PFUNC[15] and MCASP_PDIR[15] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
14	AXR14	Determines drive on AXR14 output pin when the corresponding MCASP_PFUNC[14] and MCASP_PDIR[14] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0

Bits	Field Name	Description	Type	Reset
13	AXR13	Determines drive on AXR13 output pin when the corresponding MCASP_PFUNC[13] and MCASP_PDIR[13] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
12	AXR12	Determines drive on AXR12 output pin when the corresponding MCASP_PFUNC[12] and MCASP_PDIR[12] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
11	AXR11	Determines drive on AXR11 output pin when the corresponding MCASP_PFUNC[11] and MCASP_PDIR[11] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
10	AXR10	Determines drive on AXR10 output pin when the corresponding MCASP_PFUNC[10] and MCASP_PDIR[10] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
9	AXR9	Determines drive on AXR9 output pin when the corresponding MCASP_PFUNC[9] and MCASP_PDIR[9] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
8	AXR8	Determines drive on AXR8 output pin when the corresponding MCASP_PFUNC[8] and MCASP_PDIR[8] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
7	AXR	Determines drive on AXR7 output pin when the corresponding MCASP_PFUNC[7] and MCASP_PDIR[7] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
6	AXR6	Determines drive on AXR6 output pin when the corresponding MCASP_PFUNC[6] and MCASP_PDIR[6] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
5	AXR5	Determines drive on AXR5 output pin when the corresponding MCASP_PFUNC[5] and MCASP_PDIR[5] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
4	AXR4	Determines drive on AXR4 output pin when the corresponding MCASP_PFUNC[4] and MCASP_PDIR[4] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
3	AXR3	Determines drive on AXR3 output pin when the corresponding MCASP_PFUNC[3] and MCASP_PDIR[3] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
2	AXR2	Determines drive on AXR2 output pin when the corresponding MCASP_PFUNC[2] and MCASP_PDIR[2] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0

Bits	Field Name	Description	Type	Reset
1	AXR1	Determines drive on AXR1 output pin when the corresponding MCASP_PFUNC[1] and MCASP_PDIR[1] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
0	AXR0	Determines drive on AXR0 output pin when the corresponding MCASP_PFUNC[0] and MCASP_PDIR[0] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0

Table 24-379. MCASP_PDIN

Address Offset	0x0000 001C		
Physical Address	0x4846 001C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Pin data input register - holds the state of all the McASP pins. MCASP_PDIN allows reading the actual value of the pin, regardless of the state of MCASP_PFUNC and MCASP_PDIR. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Logic level on AFSR pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
30	RESERVED	Reserved	R	0
29	ACLKR	Logic level on ACLKR pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
28	AFSX	Logic level on AFSX pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
27	AHCLKX	Logic level on AHCLKX pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
26	ACLKX	Logic level on ACLKX pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
25:16	RESERVED	Reserved	R	0x000

Bits	Field Name	Description	Type	Reset
15	AXR15	Logic level on AXR15 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
14	AXR14	Logic level on AXR14 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
13	AXR13	Logic level on AXR13 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
12	AXR12	Logic level on AXR12 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
11	AXR11	Logic level on AXR11 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
10	AXR10	Logic level on AXR10 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
9	AXR9	Logic level on AXR9 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
8	AXR8	Logic level on AXR8 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
7	AXR7	Logic level on AXR7 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
6	AXR6	Logic level on AXR6 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
5	AXR5	Logic level on AXR5 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
4	AXR4	Logic level on AXR4 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
3	AXR3	Logic level on AXR3 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
2	AXR2	Logic level on AXR2 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
1	AXR1	Logic level on AXR1 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0

Bits	Field Name	Description	Type	Reset
0	AXR0	Logic level on AXR0 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0

Table 24-380. MCASP_PDSET

Address Offset	0x0000 001C		
Physical Address	0x4846 001C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The pin data set register is an alias of the pin data output register (MCASP_PDOUT) for writes only. Writing a 1 to the MCASP_PDSET bit sets the corresponding bit in MCASP_PDOUT and, if MCASP_PFUNC = 1 (GPIO function) and MCASP_PDIR = 1 (output), drives a logic high on the pin. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Allows the corresponding AFSR bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[31] bit is set to 1.	W	0
30	RESERVED	Reserved	W	0
29	ACLKR	Allows the corresponding ACLKR bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[29] bit is set to 1.	W	0
28	AFSX	Allows the corresponding AFSX bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[28] bit is set to 1.	W	0
27	AHCLKX	Allows the corresponding AHCLKX bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[27] bit is set to 1.	W	0
26	ACLKX	Allows the corresponding ACLKX bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[26] bit is set to 1.	W	0
25:16	RESERVED	Reserved	W	0x000

Bits	Field Name	Description	Type	Reset
15	AXR15	Allows the AXR15 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[15] bit is set to 1.	W	0
14	AXR14	Allows the AXR14 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[14] bit is set to 1.	W	0
13	AXR13	Allows the AXR13 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[13] bit is set to 1.	W	0
12	AXR12	Allows the AXR12 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[12] bit is set to 1.	W	0
11	AXR11	Allows the AXR11 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[11] bit is set to 1.	W	0
10	AXR10	Allows the AXR10 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[10] bit is set to 1.	W	0
9	AXR9	Allows the AXR9 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[9] bit is set to 1.	W	0
8	AXR8	Allows the AXR8 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[8] bit is set to 1.	W	0
7	AXR7	Allows the AXR7 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[7] bit is set to 1.	W	0
6	AXR6	Allows the AXR6 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[6] bit is set to 1.	W	0
5	AXR5	Allows the AXR5 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[5] bit is set to 1.	W	0
4	AXR4	Allows the AXR4 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[4] bit is set to 1.	W	0

Bits	Field Name	Description	Type	Reset
3	AXR3	Allows the AXR3 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[3] bit is set to 1.	W	0
2	AXR2	Allows the AXR2 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[2] bit is set to 1.	W	0
1	AXR1	Allows the AXR1 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[1] bit is set to 1.	W	0
0	AXR0	Allows the AXR0 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[0] bit is set to 1.	W	0

Table 24-381. MCASP_PDCLR

Address Offset	0x0000 0020		
Physical Address	0x4846 0020	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The pin data clear register is an alias of the pin data output register (MCASP_PDOUT) for writes only. Writing a 1 to the MCASP_PDCLR bit clears the corresponding bit in MCASP_PDOUT and, if MCASP_PFUNC = 1 (GPIO function) and MCASP_PDIR = 1 (output), drives a logic low on the pin. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Allows the corresponding AFSR bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[31] bit is cleared to 0.	RW	0
30	RESERVED	Reserved	RW	0
29	ACLKR	Allows the corresponding ACLKR bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[29] bit is cleared to 0.	RW	0

Bits	Field Name	Description	Type	Reset
28	AFSX	Allows the corresponding AFSX bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[28] bit is cleared to 0.	RW	0
27	AHCLKX	Allows the corresponding AHCLKX bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[27] bit is cleared to 0.	RW	0
26	ACLKX	Allows the corresponding ACLKX bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[26] bit is cleared to 0.	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Allows the AXR15 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[15] bit is cleared to 0.	RW	0
14	AXR14	Allows the AXR14 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[14] bit is cleared to 0.	RW	0
13	AXR13	Allows the AXR13 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[13] bit is cleared to 0.	RW	0
12	AXR12	Allows the AXR12 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[12] bit is cleared to 0.	RW	0
11	AXR11	Allows the AXR11 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[11] bit is cleared to 0.	RW	0
10	AXR10	Allows the AXR10 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[10] bit is cleared to 0.	RW	0
9	AXR9	Allows the AXR9 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[9] bit is cleared to 0.	RW	0
8	AXR8	Allows the AXR8 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[8] bit is cleared to 0.	RW	0
7	AXR7	Allows the AXR7 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[7] bit is cleared to 0.	RW	0

Bits	Field Name	Description	Type	Reset
6	AXR6	Allows the AXR6 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[6] bit is cleared to 0.	RW	0
5	AXR5	Allows the AXR5 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[5] bit is cleared to 0.	RW	0
4	AXR4	Allows the AXR4 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[4] bit is cleared to 0.	RW	0
3	AXR3	Allows the AXR3 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[3] bit is cleared to 0.	RW	0
2	AXR2	Allows the AXR2 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[2] bit is cleared to 0.	RW	0
1	AXR1	Allows the AXR1 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[1] bit is cleared to 0.	RW	0
0	AXR0	Allows the AXR0 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[0] bit is cleared to 0.	RW	0

Table 24-382. MCASP_GBLCTL

Address Offset	0x0000 0044		
Physical Address	0x4846 0044	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Global transmit control register - provides initialization of the transmit and receive sections. The bit fields in MCASP_GBLCTL are synchronized and latched by the transmitter and receiver corresponding clocks - ACLKX (bits [12:8]) and ACLKR (bits [4:0]), respectively. Before programming MCASP_GBLCTL, ensure that the serial clocks are running. If the corresponding external serial clocks - ACLKX and ACLKR, are not yet running, select the internal serial clock source in AHCLKXCTL, AHCLKRCTL, ACLKXCTL and ACLKRCTL before programming the MCASP_GBLCTL. Also, after programming any bits in MCASP_GBLCTL, do not proceed until reading back from MCASP_GBLCTL and verifying that the bits in MCASP_GBLCTL are latched.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST	RESERVED	RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	RW	0x00000
12	XFRST	Transmit frame-sync generator reset enable bit 0x0: The transmit frame-sync generator is reset. 0x1: The transmit frame-sync generator is active. When released from reset, the transmit frame-sync generator begins counting serial clocks and generating frame sync as programmed.	RW	0
11	XSMRST	Transmit state-machine reset enable bit 0x0: The transmit state-machine is held in reset. AXR[n] pin state: If MCASP_PFUNC[n] = 0 and MCASP_PDIR[n] = 1, the corresponding serializer [n] drives the AXR[n] pin to the state specified for inactive time slot. 0x1: The transmit state-machine is released from reset. When released from reset, the transmit state-machine immediately transfers data from XBUF[n] to XRSR[n]. The transmit state-machine sets the underrun flag (XUNDRN) in MCASP_XSTAT, if XBUF[n] have not been preloaded with data before reset is released. The transmit state-machine also immediately begins detecting frame sync and is ready to transmit. Transmission of TDM time slot begins at slot 0 after reset is released.	RW	0
10	XSRCLR	Transmit serializer clear enable bit. By clearing and then setting this bit, the transmit buffer is flushed to an empty state (XDATA = 1). If XSMRST = 1, XSRCLR = 1, XDATA = 1, and XBUF is not loaded with new data before the start of the next active time slot, an underrun occurs. 0x0: The transmit serializer is cleared. 0x1: The transmit serializer is active. When the transmit serializer is first taken out of reset (XSRCLR changes from 0 to 1), the transmit data ready bit (XDATA) in MCASP_XSTAT is set to indicate XBUF is ready to be written.	RW	0
9	XHCLKRST	Transmit high-frequency clock divider reset enable bit 0x0: The transmitter high-frequency clock divider is held in reset and passes through its input as divide-by-1. 0x1: The transmitter high-frequency clock divider is running.	RW	0
8	XCLKRST	Transmit clock divider reset enable bit 0x0: The transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. 0x1: The transmit clock divider is running.	RW	0
7:5	RESERVED	Reserved	RW	0x0
4	RFRST	Receive frame sync generator reset enable bit. 0x0: Receive frame sync generator is reset. 0x1: Receive frame sync generator is active. When released from reset, the receive frame sync generator begins counting serial clocks and generating frame sync as programmed.	RW	0
3	RSMRST	Receive state machine reset enable bit. 0x0: Receive state machine is held in reset. 0x1: Receive state machine is released from reset. When released from reset, the receive state machine immediately begins detecting frame sync and is ready to receive. Receive TDM time slot begins at slot 0 after reset is released.	RW	0

Bits	Field Name	Description	Type	Reset
2	RSRCLR	Receive serializer clear enable bit. By clearing then setting this bit, the receive buffer is flushed. 0x0: Receive serializers are cleared. 0x1: Receive serializers are active.	RW	0
1	RHCLKRST	Receive high-frequency clock divider reset enable bit. 0x0: Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1. 0x1: Receive high-frequency clock divider is running.	RW	0
0	RCLKRST	Receive clock divider reset enable bit. 0x0: Receive clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. 0x1: Receive clock divider is running.	RW	0

Table 24-383. MCASP_AMUTE

Address Offset	0x0000 0048		
Physical Address	0x4846 0048	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Mute control register - Controls the McASP mute output pin - AMUTE (Not implemented at device level)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0000 0000

Table 24-384. MCASP_LBCTL

Address Offset	0x0000 004C		
Physical Address	0x4846 004C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The digital loopback control register (MCASP_LBCTL) controls the internal (McASP module)- level and chip-level loopback settings of the McASP in TDM mode. Note that loopback is NOT supported if McASP is configured in DIT mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IOLBEN	MODE		ORD	DLBEN

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	RW	0x000 0000

Bits	Field Name	Description	Type	Reset
4	IOLBEN	If DLBEN=0b1, the IOLBEN bit selects between **internal-level (McASP module-level)** and **chip I/O-level** loopback modes. IOLBEN bit value is irrelevant, If DLBEN=0b0. 0x0: McASP internal loopback mode enabled. This selects a direct loopback between corresponding McASP AXRn and AXRn+1 pins, bypassing device pad I/O buffers. 0x1: Chip I/O-level loopback mode enabled. The McASP data is looped back through the device pad I/O buffers.	RW	0
3:2	MODE	Loopback generator mode bits. 0x0: RESERVED 0x1: MODE must be set to 0x1 when McASP operates in loopback mode (DLBEN =0b1). This is necessary to allow transmit clock and frame sync generators to be used by both transmit and receive sections. 0x2, 0x3: Reserved	RW	0x0
1	ORD	Loopback order bit when loopback mode is enabled (DLBEN = 1). 0x0: Odd serializers N + 1 transmit to even serializers N that receive. The corresponding serializers must be programmed properly. 0x1: Even serializers N transmit to odd serializers N+1 that receive. The corresponding serializers must be programmed properly.	RW	0
0	DLBEN	Loop back mode enable bit. 0x0: Loop back mode is disabled (normal McASP operation). 0x1: Loop back is enabled (TDM mode only). Loopback type is selected in IOLBEN bit.	RW	0

Table 24-385. MCASP_TXDITCTL

Address Offset	0x0000 0050		
Physical Address	0x4846 0050	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit DIT mode control register, controls DIT operations of the McASP		
Type	RW		

31–4	3	2	1	0
RESERVED	VB	VA	RESERVED	DITEN

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW	0x0000000
3	VB	Valid bit for odd time slots (DIT right subframe). 0x0: V bit is 0 during odd DIT subframes. 0x1: V bit is 1 during odd DIT subframes.	RW	0
2	VA	Valid bit for even time slots (DIT left subframe). 0x0: V bit is 0 during even DIT subframes. 0x1: V bit is 1 during even DIT subframes.	RW	0

Bits	Field Name	Description	Type	Reset
1	RESERVED	Reserved	RW	0
0	DITEN	DIT mode enable bit 0x0: DIT mode is disabled. 0x1: DIT mode is enabled. Transmitter operates in DIT encoded mode.	RW	0

Table 24-386. MCASP_GBLCTLR

Address Offset	0x0000 0060
Physical Address	0x4846 0060
Instance	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Alias of GBLCTL. When writing to this register, only the TRANSMIT bits of GBLCTL are affected (This means GBLCTL bits 8,9,10,11,12). Reads return GBLCTL
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST	RESERVED			RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		RW	0x0
12	XFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	R	0x0
11	XSMRST	XMT state machine reset 0x0: RESET 0x1: ACTIVE	R	0x0
10	XSRCLR	XMT serializer clear 0x0: CLEAR 0x1: ACTIVE	R	0x0
9	XHCLKRST	XMT High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	R	0x0
8	XCLKRST	XMT clock divder reset 0x0: RESET 0x1: ACTIVE	R	0x0
7:5	RESERVED		RW	0x0
4	RFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	RW	0x0
3	RSMRST	RCV state machine reset 0x0: RESET 0x1: ACTIVE	RW	0x0

Bits	Field Name	Description	Type	Reset
2	RSRCLR	RCV serializer clear 0x0: CLEAR 0x1: ACTIVE	RW	0x0
1	RHCLKRST	RCV High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	RW	0x0
0	RCLKRST	RCV clock divder reset 0x0: RESET 0x1: ACTIVE	RW	0x0

Table 24-387. MCASP_RXMASK

Address Offset	0x0000 0064		
Physical Address	0x4846 0064	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receive format unit bit mask register (MCASP_RXMASK) determines which bits of the received data are masked off and padded with a known value before being read by the CPU.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMASK[31:0]																															

Bits	Field Name	Description	Type	Reset
31: 0	RMASK[31:0]	Receive data mask enable bit. 0x0: Corresponding bit of receive data (after passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (RPAD and RPBIT bits in RFMT). 0x1: Corresponding bit of receive data (after passing through reverse and rotate units) is returned to CPU or DMA.	RW	0

Table 24-388. MCASP_RXFMT

Address Offset	0x0000 0068		
Physical Address	0x4846 0068	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receive bit stream format register (MCASP_RXFMT) configures the receive data format.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														RDATDLY		RRVRS	RPAD		RPBIT					RSSZ				RBUSEL	RROT		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		RW	0x0000
17:16	RDATDLY	Receive Frame sync delay of AXR[n] 0x0: 0-bit delay. The first receive data bit, AXR[n], occurs in same ACLKR cycle as the receive frame sync (AFSR). 0x1: 1-bit delay. The first receive data bit, AXR[n], occurs one ACLKR cycle after the receive frame sync (AFSR). 0x2: 2-bit delay. The first receive data bit, AXR[n], occurs two ACLKR cycles after the receive frame sync (AFSR). 0x3: Reserved	RW	0x0
15	RRVRS	Receive serial bitstream order 0x0: Bitstream is LSB first. No bit reversal is performed in receive format bit reverse unit. 0x1: Bitstream is MSB first. Bit reversal is performed in receive format bit reverse unit.	RW	0
14:13	RPAD	Pad value for extra bits in slot not belonging to the word. This field only applies to bits when RMASK[n] = 0. 0x0: Pad extra bits with 0. 0x1: Pad extra bits with 1. 0x2: Pad extra bits with one of the bits from the word as specified by RPBIT bits. 0x3: Reserved	RW	0x0
12:8	RPBIT	RPBIT value determines which bit (as read by the CPU from RBUF[n]) is used to pad the extra bits. This field only applies when RPAD = 2h. 0x0: Pad with value of bit RBUFn[0]. 0x01 - 0x1F: Pad with value of the bit positioned within the range RBUFn[31:1].	RW	0x00
7:4	RSSZ	Receive slot size. 0x0 - 0x2: Reserved 0x3: Slot size is 8 bits 0x4: Reserved 0x5: Slot size is 12 bits 0x6: Reserved 0x7: Slot size is 16 bits 0x8: Reserved 0x9: Slot size is 20 bits 0xA: Reserved 0xB: Slot size is 24 bits 0xC: Reserved 0xD: Slot size is 28 bits 0xE: Reserved 0xF: Slot size is 32 bits	RW	0x0
3	RBUSEL	Selects whether reads from serializer buffer RBUF[n] originate from the peripheral configuration CFG port or the DATA port. 0x0: Reads from XRBUF[n] originate on DATA port. Reads from XRBUF[n] on the peripheral configuration port are ignored. 0x1: Reads from XRBUF[n] originate on peripheral configuration port. Reads from XRBUF[n] on the DATA port are ignored.	RW	0

Bits	Field Name	Description	Type	Reset
2:0	RROT	Right-rotation value for receive rotate right format unit. 0x0: Rotate right by 0 (no rotation). 0x1: Rotate right by 4 bit positions. 0x2: Rotate right by 8 bit positions. 0x3: Rotate right by 12 bit positions. 0x4: Rotate right by 16 bit positions. 0x5: Rotate right by 20 bit positions. 0x6: Rotate right by 24 bit positions. 0x7: Rotate right by 28 bit positions.	RW	0x0

Table 24-389. MCASP_RXFMCTL

Address Offset	0x0000 006C		
Physical Address	0x4846 006C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receive frame sync control register (MCASP_RXFMCTL) configures the receive frame sync (AFSR).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RMOD									RESERVED		FRWID	RESERVED		FSRM	FSRP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0x0000
15:7	RMOD	Receive frame sync mode select bits. 0x0: Burst mode 0x1: Reserved 0x2: 2-slot TDM mode (I2S receive mode) 0x3 - 0x20: 3-slot TDM to 32-slot TDM mode 0x21 - 0x17F: Reserved 0x180: 384-slot TDM (external DIR IC inputting 384-slot DIR frames to McASP) 0x181 - 0x1FF: Reserved	RW	0x000
6:5	RESERVED		RW	0x0
4	FRWID	Receive frame sync width select bit indicates the width of the receive frame sync (AFSR) during its active period. 0x0: Single bit 0x1: Single word. Single word is not supported if RMOD is set to burst mode.	RW	0
3:2	RESERVED		RW	0x0
1	FSRM	Receive frame sync generation select bit. 0x0: Externally-generated receive frame sync 0x1: Internally-generated receive frame sync	RW	0

Bits	Field Name	Description	Type	Reset
0	FSRP	Receive frame sync polarity select bit. 0x0: A rising edge on receive frame sync (AFSR) indicates the beginning of a frame. 0x1: A falling edge on receive frame sync (AFSR) indicates the beginning of a frame.	RW	0

Table 24-390. MCASP_ACLKRCTL

Address Offset	0x0000 0070		
Physical Address	0x4846 0070	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receive clock control register (MCASP_ACLKRCTL) configures the receive bit clock (ACLKR) and the receive clock generator.		
Type	RW		

31 30 29 28 27 26 25 24 23 22 21	20	19	18	17 16	15 14 13 12 11 10 9 8	7	6	5	4 3 2 1 0
RESERVED	BUSY	DIVBUSY	ADJBUSY	CLKRADJ	RESERVED	CLKRP	RESERVED	CLKRM	CLKRDIV

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		RW	0x000000
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
17:16	CLKRADJ	CLKRDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0.If CLKRDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15:8	RESERVED		RW	0x0

Bits	Field Name	Description	Type	Reset
7	CLKRP	Receive bitstream clock polarity select bit. Note that this bitfield does not have any effect, if MCASP_ACLKXCTL[6] ASYNC = 0 0x0: Falling edge. Receiver samples data on the falling edge of the serial clock, so the external transmitter driving this receiver must shift data out on the rising edge of the serial clock. 0x1: Rising edge. Receiver samples data on the rising edge of the serial clock, so the external transmitter driving this receiver must shift data out on the falling edge of the serial clock.	RW	0
6	RESERVED		RW	0
5	CLKRM	Receive bit clock source bit. Note that this bitfield does not have any effect, if MCASP_ACLKXCTL[6] ASYNC = 0 0x0: External receive clock source from ACLKR pin. 0x1: Internal receive clock source from output of programmable bit clock divider.	RW	1
4:0	CLKRDIV	Receive bit clock divide ratio bits determine the divide-down ratio from AHCLKR to ACLKR. Note that this bitfield does not have any effect, if MCASP_ACLKXCTL[6] ASYNC = 0. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 - 0x1F: Divide-by-3 to divide-by-32	RW	0x00

Table 24-391. MCASP_AHCLKRCTL

Address Offset	0x0000 0074		
Physical Address	0x4846 0074	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receive high-frequency clock control register (MCASP_AHCLKRCTL) configures the receive high-frequency master clock (AHCLKR) and the receive clock generator.		
Type	RW		

31–21	20	19	18	17–16	15	14	13–12	11–0
RESERVED	BUSY	DIVBUSY	ADJBUSY	HCLKRADJ	HCLKRM	HCLKRP	RESERVED	HCLKRDIV

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		RW	0x0000
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0

Bits	Field Name	Description	Type	Reset
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
17:16	HCLKRADJ	HCLKRDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0. If HCLKRDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15	HCLKRM	High Freq. RCV clock Source 0x0: EXTERNAL 0x1: INTERNAL	RW	0x1
14	HCLKRP	Receive bitstream high-frequency clock polarity select bit. 0x0: Not inverted. AHCLKR is not inverted before programmable bit clock divider. 0x1: Inverted. AHCLKR is inverted before programmable bit clock divider.	RW	0
13:12	RESERVED		RW	0x0
11:0	HCLKRDIV	Receive high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKR. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 - 0xFFF: Divide-by-3 to divide-by-4096	RW	0x000

Table 24-392. MCASP_RXTDM

Address Offset	0x0000 0078		
Physical Address	0x4846 0078	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receive TDM time slot register (MCASP_RXTDM) specifies which TDM time slot the receiver is active.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMS[31:0]																															

Bits	Field Name	Description	Type	Reset
31:0	RTDMS[31:0]	Receiver mode during TDM time slot n. 0x0: Receive TDM time slot n is inactive. The receive serializer does not shift in data during this slot. 0x1: Receive TDM time slot n is active. The receive serializer shifts in data during this slot.	RW	0

Table 24-393. MCASP_EVTCTLR

Address Offset	0x0000 007C		
Physical Address	0x4846 007C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Receiver Interrupt control register - controls generation of the McASP receive interrupt (RINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates RINT.		
Type	RW		

31–8	7	6	5	4	3	2	1	0
RESERVED	RSTAFRM	RESERVED	RDATA	RLAST	RDMAERR	RCKFAIL	RSYNCERR	ROVRN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x000000
7	RSTAFRM	Receive start of frame interrupt enable bit 0x0: Interrupt is disabled. A receive-start-of-frame interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive-start-of-frame interrupt generates a McASP receive interrupt (RINT).	RW	0
6	RESERVED	Reserved	RW	0
5	RDATA	Receive data-ready interrupt enable bit 0x0: Interrupt is disabled. A receive data-ready interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive data-ready interrupt generates a McASP receive interrupt (RINT).	RW	0
4	RLAST	Receive last slot interrupt enable bit 0x0: Interrupt is disabled. A receive-last-slot interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive-last-slot interrupt generates a McASP receive interrupt (RINT).	RW	0
3	RDMAERR	Receive DMA error interrupt enable bit 0x0: Interrupt is disabled. A receive DMA error interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive DMA error interrupt generates a McASP receive interrupt (RINT).	RW	0

Bits	Field Name	Description	Type	Reset
2	RCKFAIL	Receive clock failure interrupt enable bit 0x0: Interrupt is disabled. A receive clock failure interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive clock failure interrupt generates a McASP receive interrupt (RINT).	RW	0
1	RSYNCERR	Unexpected receive frame-sync interrupt enable bit 0x0: Interrupt is disabled. An unexpected receive frame-sync interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. An unexpected receive frame-sync interrupt generates a McASP receive interrupt (RINT).	RW	0
0	ROVRN	Receiver overrun interrupt enable bit 0x0: Interrupt is disabled. A receiver overrun interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receiver overrun interrupt generates a McASP receive interrupt (RINT).	RW	0

Table 24-394. MCASP_RXSTAT

Address Offset	0x0000 0080		
Physical Address	0x4846 0080	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receiver status register (MCASP_RXSTAT) provides the receiver status and receive TDM time slot number.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RERR	RDMAERR	RSTAFRM	RDATA	RLAST	RTDMSLOT	RCKFAIL	RSYNCERR	ROVRN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		RW	0x00 0000
8	RERR	RERR bit always returns a logic-OR of: ROVRN | RSYNCERR | RCKFAIL | RDMAERR Allows a single bit to be checked to determine if a receiver error has occurred. 0x0: No errors have occurred. 0x1: An error has occurred.	RW	0
7	RDMAERR	Receive DMA error flag. RDMAERR is set when the CPU or DMA reads more serializers through the DMA port in a given time slot than were programmed as receivers. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Receive DMA error did not occur. 0x1: Receive DMA error did occur.	RW	0

Bits	Field Name	Description	Type	Reset
6	RSTAFRM	Receive start of frame flag. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: No new receive frame sync (AFSR) is detected. 0x1: A new receive frame sync (AFSR) is detected.	RW	0
5	RDATA	Receive data ready flag. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: No new data in RBUF. 0x1: Data is transferred from XRSR to RBUF and ready to be serviced by the CPUs or DMA. When RDATA is set, it always causes a DMA event (AREVT).	RW	0
4	RLAST	Receive last slot flag. RLAST is set along with RDATA, if the current slot is the last slot in a frame. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Current slot is not the last slot in a frame. 0x1: Current slot is the last slot in a frame. RDATA is also set.	RW	0
3	RTDMSLOT	Returns the LSB of RSLOT. Allows a single read of MCASP_RXSTAT to determine whether the current TDM time slot is even or odd. 0x0: Current TDM time slot is odd. 0x1: Current TDM time slot is even.	RW	0
2	RCKFAIL	Receive clock failure flag. RCKFAIL is set when the receive clock failure detection circuit reports an error. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Receive clock failure did not occur. 0x1: Receive clock failure did occur.	RW	0
1	RSYNCERR	Unexpected receive frame sync flag. RSYNCERR is set when a new receive frame sync (AFSR) occurs before it is expected. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Unexpected receive frame sync did not occur. 0x1: Unexpected receive frame sync did occur.	RW	0
0	ROVRN	Receiver overrun flag. ROVRN is set when the receive serializer is instructed to transfer data from XRSR to RBUF, but the former data in RBUF has not yet been read by the CPU or DMA. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Receiver overrun did not occur. 0x1: Receiver overrun did occur.	RW	0

Table 24-395. MCASP_RXTDMSLOT

Address Offset	0x0000 0084		
Physical Address	0x4846 0084	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The current receive TDM time slot register (MCASP_RXTDMSLOT) indicates the current time slot for the receive data frame.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RSLOTCNT								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x00 0000
8:0	RSLOTCNT	0x0 - 0x17F: Current receive time slot count. Legal values: 0 to 383 (17Fh). TDM function is not supported for > 32 time slots. However, TDM time slot counter may count to 383 when used to receive a DIR block (transferred over TDM format).	R	0x000

Table 24-396. MCASP_RXCLKCHK

Address Offset	0x0000 0088		
Physical Address	0x4846 0088	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	The receive clock check control register (RCLKCHK) configures the receive clock failure detection circuit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCNT								RMAX								RMIN								RESERVED				RPS			

Bits	Field Name	Description	Type	Reset
31:24	RCNT	0x0 - 0xFF: Receive clock count value (from previous measurement). The clock circuit continually counts the number of interface clocks for every 32 receive high-frequency master clock (AHCLKR) signals, and stores the count in RCNT until the next measurement is taken.	R	0x00
23:16	RMAX	0x00-0xFF: Receive clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If the current counter value is greater than RMAX after counting 32 AHCLKR signals, RCKFAIL in MCASP_RXSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
15:8	RMIN	0x00 - 0xFF: Receive clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If RCNT is less than RMIN after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
7:4	RESERVED		RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	RPS	Receive clock check prescaler value. 0x0: McASP interface clock divided by 1 0x1: McASP interface clock divided by 2 0x2: McASP interface clock divided by 4 0x3: McASP interface clock divided by 8 0x4: McASP interface clock divided by 16 0x5: McASP interface clock divided by 32 0x6: McASP interface clock divided by 64 0x7: McASP interface clock divided by 128 0x8: McASP interface clock divided by 256 0x9 - 0xF: Reserved	RW	0x0

Table 24-397. MCASP_REVTCTL

Address Offset	0x0000 008C		
Physical Address	0x4846 008C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Receiver DMA event control register		
Type	RW		

31–1	0
RESERVED	RDATDMA

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	RW	0x0000 0000
0	RDATDMA	Receive data DMA request enable bit. 0x0: The receive data DMA request is enabled. 0x1: The receive data DMA request is disabled.	RW	0

Table 24-398. MCASP_GBLCTLX

Address Offset	0x0000 00A0		
Physical Address	0x4846 00A0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Alias of GBLCTL. When writing to this register, only the TRANSMIT bits of GBLCTL are affected (This means GBLCTL bits 8,9,10,11,12.). Reads return GBLCTL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST	RESERVED	RFRST	RSMRST	RSRCLKR	RHCLKRST	RCLKRST

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		RW	0x0
12	XFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	RW	0x0
11	XSMRST	XMT state machine reset 0x0: RESET 0x1: ACTIVE	RW	0x0
10	XSRCLR	XMT serializer clear 0x0: CLEAR 0x1: ACTIVE	RW	0x0
9	XHCLKRST	XMT High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	RW	0x0
8	XCLKRST	XMT clock divder reset 0x0: RESET 0x1: ACTIVE	RW	0x0
7:5	RESERVED		RW	0x0
4	RFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	R	0x0
3	RSMRST	RCV state machine reset 0x0: RESET 0x1: ACTIVE	R	0x0
2	RSRCLKR	RCV serializer clear 0x0: CLEAR 0x1: ACTIVE	R	0x0
1	RHCLKRST	RCV High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	R	0x0
0	RCLKRST	RCV clock divder reset 0x0: RESET 0x1: ACTIVE	R	0x0

Table 24-399. MCASP_TXMASK

Address Offset	0x0000 00A4		
Physical Address	0x4846 00A4	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit format unit bit mask register - Determines which bits of the transmitted data are masked off before being shifted out the McASP		

Table 24-399. MCASP_TXMASK (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMASK[31:0]																															

Bits	Field Name	Description	Type	Reset
31:0	XMASK[31:0]	Transmit data mask enable bit 0x0: The corresponding bit of transmit data is masked out and then transmitted out the McASP in place of the original bit. 0x1: The corresponding bit of transmit data is transmitted out the McASP.	RW	0

Table 24-400. MCASP_TXFMT

Address Offset	0x0000 00A8		
Physical Address	0x4846 00A8	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit bitstream format register - configures the transmit data format		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														XDATDLY		XRVRS	XPAD		XPBIT					XSSZ				XBUSEL	XROT		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	RW	0x0000
17:16	XDATDLY	Transmit sync bit delay 0x0: 0 bit delay - The first transmit data bit, on the AXR[n], occurs in the same ACLKX cycle as the transmit frame sync (AFSX). 0x1: 1-bit delay. The first transmit data bit, AXR[n], occurs one ACLKX cycle after the transmit frame sync (AFSX). 0x2: 2-bit delay. The first transmit data bit, AXR[n], occurs two ACLKX cycles after the transmit frame sync (AFSX). 0x3: Reserved	RW	0x0
15	XRVRS	Transmit serial bitstream order 0x0: Bitstream is LSB first. No bit reversal is performed in transmit format unit. 0x1: Bitstream is MSB first. Bit reversal is performed in transmit format bit reverse unit.	RW	0x0
14:13	XPAD	Pad value for extra bits in slot not belonging to word defined by XMASK. This field only applies to bits when XMASK[n] = 0. 0x0: Pad extra bits with 0. 0x1: Pad extra bits with 1. 0x2: Pad extra bits with one of the bits from the word as specified by XPBIT bits. 0x3: Reserved	RW	0x00

Bits	Field Name	Description	Type	Reset
12:8	XPBIT	XPBIT value determines which bit (as written by the CPU or DMA to XBUF[n]) is used to pad the extra bits before shifting. This field only applies when XPAD = 0x2. 0x0: Pad with bit 0 value. 0x1 - 0x1F: Pad with bit 1 to bit 31 value.	RW	0x0
7:4	XSSZ	Transmit slot size 0x0 - 0x2: Reserved 0x3: Slot size is 8 bits 0x4: Reserved 0x5: Slot size is 12 bits 0x6: Reserved 0x7: Slot size is 16 bits 0x8: Reserved 0x9: Slot size is 20 bits 0xA: Reserved 0xB: Slot size is 24 bits 0xC: Reserved 0xD: Slot size is 28 bits 0xE: Reserved 0xF: Slot size is 32 bits.	RW	0x0
3	XBUSEL	Selects whether writes to the serializer buffer XBUF[n] originate from the peripheral configuration CFG port or the DATA port. 0x0: Writes to XBUF[n] originate from the DATA port. Writes to XBUF[n] from the peripheral configuration port are ignored with no effect on the McASP. 0x1: Writes to XBUF[n] originate from the peripheral configuration port - CFG port. Writes to XBUF[n] from the DATA port are ignored with no effect on the McASP.	RW	0
2:0	XROT	Right-rotation value for transmit rotate right format unit 0x0: Rotate right by 0 (no rotation). 0x1: Rotate right by 4 bit positions. 0x2: Rotate right by 8 bit positions. 0x3: Rotate right by 12 bit positions. 0x4: Rotate right by 16 bit positions. 0x5: Rotate right by 20 bit positions. 0x6: Rotate right by 24 bit positions. 0x7: Rotate right by 28 bit positions.	RW	0x0

Table 24-401. MCASP_TXFMCTL

Address Offset	0x0000 00AC		
Physical Address	0x4846 00AC	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit frame-sync control register - configures the transmit frame sync (AFSX).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	XMOD	RESERVED	FXWID	RESERVED	FSXM	FSXP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW	0x0000
15:7	XMOD	Transmit frame-sync mode select bits 0x0: Burst mode 0x1: Reserved 0x2: 2-slot TDM mode (I2S transmit mode) 0x3 - 0x20: 3-slot TDM to 32-slot TDM mode 0x21 - 0x17F: Reserved 0x180: 384-slot DIT mode All other: Reserved	RW	0x000
6:5	RESERVED	Reserved	RW	0x0
4	FXWID	The transmit frame-sync width select bit indicates the width of the transmit frame sync (AFSX) during its active period. 0x0: Single bit 0x1: Single word . Single word is not supported if XMOD is set to burst mode.	RW	0
3:2	RESERVED	Reserved	RW	0x0
1	FSXM	Transmit frame-sync generation select bit 0x0: Externally-generated transmit frame 0x1: Internally-generated transmit frame sync	RW	0
0	FSXP	Transmit frame-sync polarity select bit 0x0: Rising Edge - A rising edge on transmit frame sync (AFSX) indicates the beginning of a frame. 0x1: Falling Edge - A falling edge on transmit frame sync (AFSX) indicates the beginning of a frame.	RW	0

Table 24-402. MCASP_ACLKXCTL

Address Offset	0x0000 00B0		
Physical Address	0x4846 00B0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit clock control register - Configures the transmit bit clock (ACLKX) and the transmit clock generator.		
Type	RW		

31:21	20	19	18	17:16	15:8	7	6	5	4:0
RESERVED	BUSY	DIVBUSY	ADJBUSY	CLKXADJ	RESERVED	CLKXP	ASYNC	CLKXM	CLKXDIV

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	RW	0x00
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0

Bits	Field Name	Description	Type	Reset
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
17:16	CLKXADJ	CLKXDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0. If CLKXDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15:8	RESERVED		RW	0x0
7	CLKXP	Transmit bitstream clock polarity select bit. 0x0: Rising edge. External receiver samples data on the falling edge of the serial clock, so the transmitter must shift data out on the rising edge of the serial clock. 0x1: Falling edge. External receiver samples data on the rising edge of the serial clock, so the transmitter must shift data out on the falling edge of the serial clock.	RW	0
6	ASYNC	Transmit operation asynchronous enable bit 0x0: Synchronous. Transmit clock and frame sync provides the source for both the transmit and receive sections. Note that in this mode, the receive bit clock is an inverted version of the transmit bit clock. 0x1: Asynchronous. Separate clock and frame sync used by transmit and receive sections.	RW	1
5	CLKXM	Transmit bit clock source bit 0x0: External transmit clock source from ACLKX pin. 0x1: Internal (output of divider)	RW	1
4:0	CLKXDIV	Transmit bit clock divide ratio bits, determine the divide-down ratio from AHCLKX to ACLKX. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 to 0x1F: Divide-by-3 to divide-by-32	RW	0x00

Table 24-403. MCASP_AHCLKXCTL

Address Offset	0x0000 00B4		
Physical Address	0x4846 00B4	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	High-frequency transmit clock control register - Configures the transmit high-frequency master clock (AHCLKX) and the transmit clock generator.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	BUSY	DIVBUSY	ADJBUSY	HCLKXADJ	HCLKXM	HCLKXP	RESERVED	HCLKXDIV

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	RW	0x000
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
17:16	HCLKXADJ	HCLKXDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0. If HCLKXDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15	HCLKXM	Transmit high-frequency clock source bit 0x0: External transmit high-frequency clock source from AHCLKX pin. 0x1: Internal transmit high-frequency clock source from output of programmable high clock divider	RW	1
14	HCLKXP	Transmit bitstream high-frequency clock polarity select bit. 0x0: Not inverted. AHCLKX is not inverted before programmable bit clock divider. 0x1: Inverted. AHCLKX is inverted before programmable bit clock divider.	RW	0
13:12	RESERVED	Reserved	RW	0x0
11:0	HCLKXDIV	Transmit high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKX. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 to 0xFFF: Divide-by-3 to divide-by-4096	RW	0x000

Table 24-404. MCASP_TXTDM

Address Offset	0x0000 00B8		
Physical Address	0x4846 00B8	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit TDM slot 0-31 register - TDM time slot counter range is to 384 slots (to support SPDIF blocks of 384 subframes).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

XTDMS[31:0]

Bits	Field Name	Description	Type	Reset
31:0	XTDMS[31:0]	Transmitter mode during TDM time slot n (n = 0..31) 0x0: Transmit TDM time slot n is inactive. The transmit serializer does not shift out data during this slot. 0x1: The transmit TDM time slot n is active. The transmit serializer shifts out data during this slot according to the serializer control registers - MCASP_XRSRCTLn.	RW	0

Table 24-405. MCASP_EVTCTLX

Address Offset	0x0000 00BC		
Physical Address	0x4846 00BC	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmitter Interrupt control register - controls generation of the McASP transmit interrupt (XINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates XINT.		
Type	RW		

31–8	7	6	5	4	3	2	1	0
RESERVED	XSTAFRM	RESERVED	XDATA	XLAST	XDMAERR	XCKFAIL	XSYNCERR	XUNDRN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x000000
7	XSTAFRM	Transmit start of frame interrupt enable bit 0x0: Interrupt is disabled. A transmit-start-of-frame interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit-start-of-frame interrupt generates a McASP transmit interrupt (XINT).	RW	0
6	RESERVED	Reserved	RW	0
5	XDATA	Transmit data-ready interrupt enable bit 0x0: Interrupt is disabled. A transmit data-ready interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit data-ready interrupt generates a McASP transmit interrupt (XINT).	RW	0
4	XLAST	Transmit last slot interrupt enable bit 0x0: Interrupt is disabled. A transmit-last-slot interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit-last-slot interrupt generates a McASP transmit interrupt (XINT).	RW	0
3	XDMAERR	Transmit DMA error interrupt enable bit 0x0: Interrupt is disabled. A transmit DMA error interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit DMA error interrupt generates a McASP transmit interrupt (XINT).	RW	0

Bits	Field Name	Description	Type	Reset
2	XCKFAIL	Transmit clock failure interrupt enable bit 0x0: Interrupt is disabled. A transmit clock failure interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit clock failure interrupt generates a McASP transmit interrupt (XINT).	RW	0
1	XSYNCERR	Unexpected transmit frame-sync interrupt enable bit 0x0: Interrupt is disabled. An unexpected transmit frame-sync interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. An unexpected transmit frame-sync interrupt generates a McASP transmit interrupt (XINT).	RW	0
0	XUNDRN	Transmitter underrun interrupt enable bit 0x0: Interrupt is disabled. A transmitter underrun interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmitter underrun interrupt generates a McASP transmit interrupt (XINT).	RW	0

Table 24-406. MCASP_TXSTAT

Address Offset	0x0000 00C0		
Physical Address	0x4846 00C0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmitter status register - If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes the generation of a new interrupt request.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							XERR	XDMAERR	XSTAFRM	XDATA	XLAST	XTDMSLOT	XCKFAIL	XSYNCERR	XUNDRN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	RW	0x000000
8	XERR	XERR bit always returns a logic-OR of: XUNDRN | XSYNCERR | XCKFAIL | XDMAERR. Allows a single bit to be checked to determine if a transmitter error interrupt has occurred. 0x0: No errors have occurred. 0x1: An error has occurred.	RW	0
7	XDMAERR	Transmit DMA error flag. XDMAERR is set when the CPU or DMA writes more words to the DATA port of the McASP in a given time slot than it should. Causes a transmit interrupt (XINT) if this bit and XDMAERR in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect. 0x0: Transmit DMA error did not occur. 0x1: Transmit DMA error occurred.	RW	0

Bits	Field Name	Description	Type	Reset
6	XSTAFRM	Transmit start of frame flag. Causes a transmit interrupt (XINT) if this bit and XSTAFRM in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect. 0x0: No new transmit frame sync (AFSX) is detected. 0x1: A new transmit frame sync (AFSX) is detected.	RW	0
5	XDATA	Transmit data ready flag. Causes a transmit interrupt (XINT) if this bit and XDATA in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect 0x0: XBUF[n] is written and is full 0x1: Data is copied from XBUF[n] to XRSR[n]. XBUF[n] is empty and ready to be written. XDATA is also set when the transmit serializers are taken out of reset. When XDATA is set, it always causes a DMA event (AXEVT).	RW	0
4	XLAST	Transmit last slot flag. XLAST, along with XDATA, are set if the current slot is the last slot in a frame. Causes a transmit interrupt (XINT) if this bit and XLAST in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Current slot is not the last slot in a frame. 0x1: Current slot is the last slot in a frame. XDATA is also set.	RW	0
3	XTDMSLOT	Returns the LSB of XSLOT. Allows a single read of XSTAT to determine whether the current TDM time slot is even or odd. read 0x0: Current TDM time slot is odd. read 0x1: Current TDM time slot is even.	R	0
2	XCKFAIL	Transmit clock failure flag. XCKFAIL is set when the transmit clock failure detection circuit reports an error. Causes a transmit interrupt (XINT) if this bit and XCKFAIL in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Transmit clock failure did not occur. 0x1: Transmit clock failure occurred	RW	0
1	XSYNCERR	Unexpected transmit frame-sync flag. XSYNCERR is set when a new transmit frame sync (AFSX) occurs before it is expected. Causes a transmit interrupt (XINT) if this bit and XSYNCERR in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Unexpected transmit frame sync did not occur 0x1: Unexpected transmit frame sync occurred.	RW	0
0	XUNDRN	Transmitter underrun flag. XUNDRN is set when the transmit serializer is instructed to transfer data from XBUF[n] to XRSR[n], but XBUF[n] has not yet been serviced with new data since the last transfer. Causes a transmit interrupt (XINT) if this bit and XUNDRN in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Transmitter underrun did not occur 0x1: Transmitter underrun occurred.	RW	0

Table 24-407. MCASP_TXTDMSLOT

Address Offset	0x0000 00C4		
Physical Address	0x4846 00C4	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Current transmit TDM time slot register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							XSLOTCNT								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8:0	XSLOTCNT	Current transmit time slot count. the value of this register is 0b0101111111 (0x17f) during reset and 0 after reset.	R	0x000

Table 24-408. MCASP_TXCLKCHK

Address Offset	0x0000 00C8		
Physical Address	0x4846 00C8	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit clock check control register - configures the transmit clock failure detection circuit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCNT								XMAX								XMIN								RESERVED				XPS			

Bits	Field Name	Description	Type	Reset
31:24	XCNT	Transmit clock count value (from previous measurement). The clock circuit continually counts the number of interface clocks for every 32 transmit high-frequency master clock (AHCLKX) signals, and stores the count in XCNT until the next measurement is taken	R	0x00
23:16	XMAX	0x0 to 0xFF: Transmit clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If the current counter value is greater than XMAX after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
15:8	XMIN	0x0 to 0xFF: Transmit clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If XCNT is less than XMIN after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
7:4	RESERVED	Reserved	RW	0x0
3:0	XPS	Transmit clock check prescaler value 0x0: McASP interface clock divided by 1 0x1: McASP interface clock divided by 2 0x2: McASP interface clock divided by 4 0x3: McASP interface clock divided by 8 0x4: McASP interface clock divided by 16 0x5: McASP interface clock divided by 32 0x6: McASP interface clock divided by 64 0x7: McASP interface clock divided by 128 0x8: McASP interface clock divided by 256 0x9 to 0xF: Reserved	RW	0x0

Table 24-409. MCASP_XEVTCTL

Address Offset	0x0000 00CC

Table 24-409. MCASP_XEVTCTL (continued)

Physical Address	0x4846 00CC	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmitter DMA event control register		
Type	RW		

31–1	0
RESERVED	XDATDMA

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	RW	0x0000 0000
0	XDATDMA	Transmit data DMA request enable bit. 0x0: The transmit data DMA request is enabled. 0x1: The transmit data DMA request is disabled.	RW	0

Table 24-410. MCASP_CLKADJEN

Address Offset	0x0000 00D0		
Physical Address	0x4846 00D0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	One-Shot Clock Adjustment Enable		
Type	RW		

31–1	0
RESERVED	ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		RW	0x0
0	ENABLE	One-shot clock adjust enable. Not supported. Bit field must always be written as 0x0. 0x0: DISABLE 0x1: ENABLE	RW	0x0

Table 24-411. MCASP_DITCSRAi

Address Offset	0x0000 0100 + (0x4*i)	**Index**	i = 0 to 5

Table 24-411. MCASP_DITCSRAi (continued)

Physical Address	0x4846 0100 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	DIT left channel status register - All six 32-bit registers (i = 0 to 5) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register file before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITCSRAi																															

Bits	Field Name	Description	Type	Reset
31:0	DITCSRAi	Left (even TDM slot) channel status	RW	0x0000 0000

Table 24-412. MCASP_DITCSRBi

Address Offset	0x0000 0118+ (0x4*i)	**Index**	i = 0 to 5
Physical Address	0x4846 0118 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	DIT right channel status register - All six 32-bit registers (i = 0 to 5) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register file before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITCSRBi																															

Bits	Field Name	Description	Type	Reset
31:0	DITCSRBi	Right (odd TDM slot) channel status	RW	0x0000 0000

Table 24-413. MCASP_DITUDRAi

Address Offset	0x0000 0130 + (0x4*i)	**Index**	i = 0 to 5
Physical Address	0x4846 0130 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	DIT left channel user data register - provides the user data of each left channel (even TDM time slot). All six 32-bit registers (i = 0 to 5) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DITUDRAi																															

Bits	Field Name	Description	Type	Reset
31:0	DITUDRAi	Left (even TDM slot) user data	RW	0x0000 0000

Table 24-414. MCASP_DITUDRBi

Address Offset	0x0000 0148+ (0x4*i)	**Index**	i = 0 to 5
Physical Address	0x4846 0148 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	DIT right user data register - provides the user data of each right channel (odd TDM time slot). All six 32-bit registers (i = 0 to 5) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITUDRBi																															

Bits	Field Name	Description	Type	Reset
31:0	DITUDRBi	Right (odd TDM slot) user data	RW	0x0000 0000

Table 24-415. MCASP_XRSRCTLn

Address Offset	0x0000 0180 + (0x4*n)	**Index**	n = 0 to 15
Physical Address	0x4846 0180 + (0x04*n)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Serializer n control register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										RRDY	XRDY	DISMOD		SRMOD	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0x0000000
5	RRDY	Receive buffer ready bit. RRDY indicates the current receive buffer state. Always reads 0 when programmed as a transmitter or as inactive. If SRMOD bit is set to receive (2h), RRDY switches from 0 to 1 whenever data is transferred from XRSRn to RBUFn. Read 0x0: Receive buffer (MCASP_RXBUFn) is empty. Read 0x1: Receive buffer (MCASP_RXBUFn) contains data and needs to be read before the start of the next time slot or a receiver overrun occurs.	R	0

Bits	Field Name	Description	Type	Reset
4	XRDY	Transmit buffer ready bit. XRDY indicates the current transmit buffer state. Always reads 0 when programmed as a receiver or as inactive. If SRMOD bit is set to transmit (1h), XRDY switches from 0 to 1 when XSRCLR in GBLCTL is switched from 0 to 1 to indicate an empty transmitter. XRDY remains set until XSRCLR is forced to 0, data is written to the corresponding transmit buffer, or SRMOD bit is changed to receive (2h) or inactive (0). Read 0x0: The transmit buffer (MCASP_TXBUFn) contains data. Read 0x1: The transmit buffer (MCASP_TXBUFn) is empty and needs to be written before the start of the next time slot or a transmit underrun occurs.	R	0
3:2	DISMOD	Serializer pin drive mode bit. Drive on pin when in inactive TDM slot of transmit mode or when serializer is inactive. This field only applies if the pin is configured as a McASP pin (PFUNC = 0). 0x0: Drive on pin is 3-state. 0x1: Reserved 0x2: Drive on pin is logic low. 0x3: Drive on pin is logic high.	RW	0x0
1:0	SRMOD	Serializer mode bit 0x0:The serializer is inactive 0x1:The serializer is operating in transmit mode. 0x2: The serializer is operating in receive mode. 0x3: Reserved	RW	0x0

Table 24-416. MCASP_TXBUFn

Address Offset	0x0000 0200 + (0x4*n)	**Index**	n = 0 to 15
Physical Address	0x4846 0200 + (0x04*n)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Transmit buffer n - The transmit buffer for the serializer n holds data from the transmit format unit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XBUFn																															

Bits	Field Name	Description	Type	Reset
31:0	XBUFn	Transmit buffer n	RW	0x0000 0000

Table 24-417. MCASP_RXBUFn

Address Offset	0x0000 0280 + (0x4*n)	**Index**	n = 0 to 15
Physical Address	0x4846 0280 + (0x04*n)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4 MCASP4_CFG_PER2_L4 MCASP5_CFG_PER2_L4 MCASP6_CFG_PER2_L4 MCASP7_CFG_PER2_L4 MCASP8_CFG_PER2_L4
Description	Receive buffer n - The receive buffer for the serializer n holds data before the data goes to the receive format unit.		

Table 24-417. MCASP_RXBUFn (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBUFn																															

Bits	Field Name	Description	Type	Reset
31:0	RBUFn	Receive Buffer n	RW	0x0000 0000

24.6.6.2.3 MCASP_AFIFO Register Summary

Table 24-418. MCASP_AFIFO Register Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_AFIFO L4_PER2 Physical Address
WFIFOCTL	RW	32	0x0000 0000	0x4846 1000
WFIFOSTS	R	32	0x0000 0004	0x4846 1004
RFIFOCTL	RW	32	0x0000 0008	0x4846 1008
RFIFOSTS	R	32	0x0000 000C	0x4846 100C

Table 24-419. MCASP_AFIFO Register Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_AFIFO L4_PER2 Physical Address
WFIFOCTL	RW	32	0x0000 0000	0x4846 5000
WFIFOSTS	R	32	0x0000 0004	0x4846 5004
RFIFOCTL	RW	32	0x0000 0008	0x4846 5008
RFIFOSTS	R	32	0x0000 000C	0x4846 500C

Table 24-420. MCASP_AFIFO Register Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	MCASP3_AFIFO L4_PER2 Physical Address	MCASP4_AFIFO L4_PER2 Physical Address	MCASP5_AFIFO L4_PER2 Physical Address
WFIFOCTL	RW	32	0x0000 0000	0x4846 9000	0x4846 D000	0x4847 1000
WFIFOSTS	R	32	0x0000 0004	0x4846 9004	0x4846 D004	0x4847 1004
RFIFOCTL	RW	32	0x0000 0008	0x4846 9008	0x4846 D008	0x4847 1008
RFIFOSTS	R	32	0x0000 000C	0x4846 900C	0x4846 D00C	0x4847 100C

Table 24-421. MCASP_AFIFO Register Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	MCASP6_AFIFO L4_PER2 Physical Address	MCASP7_AFIFO L4_PER2 Physical Address	MCASP8_AFIFO L4_PER2 Physical Address
WFIFOCTL	RW	32	0x0000 0000	0x4847 5000	0x4847 9000	0x4847 D000
WFIFOSTS	R	32	0x0000 0004	0x4847 5004	0x4847 9004	0x4847 D004
RFIFOCTL	RW	32	0x0000 0008	0x4847 5008	0x4847 9008	0x4847 D008
RFIFOSTS	R	32	0x0000 000C	0x4847 500C	0x4847 900C	0x4847 D00C

24.6.6.2.4 MCASP_AFIFO Register Description

Table 24-422. WFIFOCTL

Address Offset	0x0000 0000

Table 24-422. WFIFOCTL (continued)

Physical Address	0x4846 1000 0x4846 5000 0x4846 9000 0x4846 D000 0x4847 1000 0x4847 5000 0x4847 9000 0x4847 D000	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4 MCASP4_AFIFO_PER2_L4 MCASP5_AFIFO_PER2_L4 MCASP6_AFIFO_PER2_L4 MCASP7_AFIFO_PER2_L4 MCASP8_AFIFO_PER2_L4
Description	The Write FIFO control register. The WNUMEVT and WNUMDMA values must be set prior to enabling the Write FIFO. If the Write FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.		
Type	RW		

31:17	16	15:8	7:0
RESERVED	WENA	WNUMEVT	WNUMDMA

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000 0000
16	WENA	Write FIFO enable bit. 0x0: Write FIFO is disabled (default). Data access by the host must pass through the FIFO block to the McASP transparently. DMA requests must also pass through the FIFO block transparently. WLVL is reset to 0 and pointers are initialized, i.e., the write FIFO is “flushed.” 0x1: Write FIFO is enabled. If write FIFO is to be enabled, it must be enabled prior to enabling McASP.	RW	0
15:8	WNUMEVT	Write word count (32-bit) to generate TX event to host. When Write FIFO has word space for more or equal to this value then transmit event will be generated to host/ DMA. This value must be set prior to enabling the write FIFO. 0x0: 0 words. 0x1: 1 word. 0x2: 2 words. 0x3 - 0x40: 3 to 64 words currently in write FIFO. 0x41 - 0xFF: Reserved.	RW	0x10
7:0	WNUMDMA	Write word count (32-bit words). On the transmit DMA event from McASP the WNUMDMA word will be transferred from DMA engine to McASP. This value must equal the number of McASP serializers used as transmitters. This value must be set prior to enabling the write FIFO. 0x0: 0 words. 0x1: 1 word. 0x2: 2 words. 0x3 - 0x10: 3 to 16 words. 0x11 - 0xFF: Reserved.	RW	0x04

Table 24-423. WFIFOSTS

Address Offset	0x0000 0004

Table 24-423. WFIFOSTS (continued)

Physical Address	0x4846 1004 0x4846 5004	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4 MCASP4_AFIFO_PER2_L4 MCASP5_AFIFO_PER2_L4 MCASP6_AFIFO_PER2_L4 MCASP7_AFIFO_PER2_L4 MCASP8_AFIFO_PER2_L4
Description	The Write FIFO status register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								WLVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000 0000
7:0	WLVL	Write level (read-only). Number of 32-bit words currently in write FIFO. 0x0: 0 words currently in write FIFO. 0x1: 1 word currently in write FIFO. 0x2: 2 words currently in write FIFO. 0x3 - 0x40: 3 to 64 words currently in write FIFO. 0x41 - 0xFF: Reserved.	R	0

Table 24-424. RFIFOCTL

Address Offset	0x0000 0008		
Physical Address	0x4846 1008	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4 MCASP4_AFIFO_PER2_L4 MCASP5_AFIFO_PER2_L4 MCASP6_AFIFO_PER2_L4 MCASP7_AFIFO_PER2_L4 MCASP8_AFIFO_PER2_L4
Description	The Read FIFO control register. The RNUMEVT and RNUMDMA values must be set prior to enabling the Read FIFO. If the Read FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															RENA	RNUMEVT								RNUMDMA							

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000 0000
16	RENA	Read FIFO enable bit. 0x0: Read FIFO is disabled (default). Data access by the host must pass through the FIFO block to the McASP transparently. DMA requests must also pass through the FIFO block transparently. RLVL is reset to 0 and pointers are initialized, i.e., the read FIFO is “flushed.” 0x1: Read FIFO is enabled. If read FIFO is to be enabled, it must be enabled prior to enabling McASP.	RW	0

Bits	Field Name	Description	Type	Reset
15:8	RNUMEVT	Read word count (32-bit) to generate RX event to host. When Read FIFO has number of word available which is more or equal to this value then receive event will be generated to host/DMA. This value must be set prior to enabling the write FIFO. 0x0: 0 words currently in read FIFO. 0x1: 1 word currently in read FIFO. 0x2: 2 words currently in read FIFO. 0x3 - 0x40: 3 to 64 words currently in read FIFO. 0x41 - 0xFF: Reserved	RW	0x10
7:0	RNUMDMA	Read word count (32-bit words). On receive DMA event from McASP, the DMA engine will read specified number of words from McASP. This value must equal the number of McASP serializers used as transmitters. This value must be set prior to enabling the read FIFO. 0x0: 0 words 0x1: 1 word 0x2: 2 words 0x3 - 0x10: 3-16 words 0x11 - 0xFF: Reserved		

Table 24-425. RFIFOSTS

Address Offset	0x0000 000C		
Physical Address	0x4846 100C	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4 MCASP4_AFIFO_PER2_L4 MCASP5_AFIFO_PER2_L4 MCASP6_AFIFO_PER2_L4 MCASP7_AFIFO_PER2_L4 MCASP8_AFIFO_PER2_L4
Description	The Read FIFO status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RLVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000 0000
7:0	RLVL	Read level (read-only). Number of 32-bit words currently in read FIFO. 0x0: 0 words currently in read FIFO. 0x1: 1 word currently in read FIFO. 0x2: 2 words currently in read FIFO. 0x3 - 0x40: 3 to 64 words currently in read FIFO. 0x41 - 0xFF: Reserved.	R	0

24.6.6.2.5 MCASP_DAT Register Summary

Table 24-426 to Table 24-429 summarize the MCASP_DAT register mapping.

Table 24-426. MCASP_DAT Register Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_DAT L3_MAIN Physical Address
MCASP_RXBUF	R	32	0x0000 0000[1]	0x4580 0000

Table 24-426. MCASP_DAT Register Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_DAT L3_MAIN Physical Address
MCASP_TXBUF	W	32	0x0000 0000(1)	0x4580 0000

(1) 0x000 is just an example DATA port offset value. Actually, whatever the offset value is added to base address, it is ignored (don't care) when MPU/DSP perfomrs accesses to XRBUFn RX/TX buffers through the McASP DATA port.

Table 24-427. MCASP_DAT Register Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_DAT L3_MAIN Physical Address
MCASP_RXBUF	R	32	0x0000 0000(1)	0x45C0 0000
MCASP_TXBUF	W	32	0x0000 0000(1)	0x45C0 0000

(1) 0x000 is just an example DATA port offset value. Actually, whatever the offset value is added to base address, it is ignored (don't care) when MPU/DSP perfomrs accesses to XRBUFn RX/TX buffers through the McASP DATA port.

Table 24-428. MCASP_DAT Register Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	MCASP3_DAT L3_MAIN Physical Address	MCASP4_DAT L4_PER2 Physical Address	MCASP5_DAT L4_PER2 Physical Address
MCASP_RXBUF	R	32	0x0000 0000	0x4600 0000	0x4843 6000	0x4843 A000
MCASP_TXBUF	W	32	0x0000 0000	0x4600 0000	0x4843 6000	0x4843 A000

Table 24-429. MCASP_DAT Register Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	MCASP6_DAT L4_PER2 Physical Address	MCASP7_DAT L4_PER2 Physical Address	MCASP8_DAT L4_PER2 Physical Address
MCASP_RXBUF	R	32	0x0000 0000(1)	0x4844 C000	0x4845 0000	0x4845 4000
MCASP_TXBUF	W	32	0x0000 0000(1)	0x4844 C000	0x4845 0000	0x4845 4000

(1) 0x000 is just an example DATA port offset value. Actually, whatever the offset value is added to base address, it is ignored (don't care) when MPU/DSP perfomrs accesses to XRBUFn RX/TX buffers through the McASP DATA port.

Note

For MCASP_RXBUF and MCASP_TXBUF buffer accesses through the McASP DATA port, the destination physical address is always the same regardless of current channel index or transfer direction. The MCASP_TXFMT[3] XBUSEL bit must be set to 0b0, to allow write transfers through the DATA port. The MCASP_RXFMT[3] RBUSEL bit must be set to 0b0, to allow read transfers through the DATA port.

Note

The McASP DATA port is exclusively assigned for DMAs/device CPUs accesses to the McASP channels transmit and receive buffer registers. All other McASP module registers must be accessed through the McASP CFG (peripheral) port.

Note

McASP1, McASP2, and McASP3, whose data port are accessible directly via L3_MAIN, do not support FIFO/constant addressing modes. Incrementing transfers must be used instead.

24.6.6.2.6 MCASP_DAT Register Description

Table 24-430. MCASP_RXBUF

Address Offset	0x0000 0000

Table 24-430. MCASP_RXBUF (continued)

Physical Address	0x4580 0000 0x45C0 0000 0x4600 0000	**Instance**	MCASP1_DAT_MAIN_L3 MCASP2_DAT_MAIN_L3 MCASP3_DAT_MAIN_L3 MCASP4_DAT_PER2_L4 MCASP5_DAT_PER2_L4 MCASP6_DAT_PER2_L4 MCASP7_DAT_PER2_L4 MCASP8_DAT_PER2_L4
Description	Through the DATA port, the Host can service all serializers through a single address and the McASP automatically cycles through the appropriate serializers. For receive operations through the DATA port, the Host should read from the same RBUF DATA port address to service all of the active receive serializers upon each receive data ready event. To enable accesses from the Host to the McASP XRBUF registers through the DATA port, one must clear the RBUSEL bits to 0 in the respective MCASP_RXFMT registers in the MCASP_CFG Memory Map.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXBUF																															

Bits	Field Name	Description	Type	Reset
31:0	RXBUF	Rx buffer data.	R	0x0000 0000

Table 24-431. MCASP_TXBUF

Address Offset	0x0000 0000		
Physical Address	0x4580 0000 0x45C0 0000	**Instance**	MCASP1_DAT_MAIN_L3 MCASP2_DAT_MAIN_L3 MCASP3_DAT_MAIN_L3 MCASP4_DAT_PER2_L4 MCASP5_DAT_PER2_L4 MCASP6_DAT_PER2_L4 MCASP7_DAT_PER2_L4 MCASP8_DAT_PER2_L4
Description	Through the DATA port, the Host can service all serializers through a single address and the McASP automatically cycles through the appropriate serializers. For transmit operations through the DATA port, the Host should write to the same DATA port address to service all of the active transmit serializers upon each transmit data ready event. To enable accesses from the Host to the McASP XRBUF registers through the DATA port, one must clear the XBUSEL bits to 0 in the respective MCASP_TXFMT registers in the MCASP_CFG Memory Map.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXBUF																															

Bits	Field Name	Description	Type	Reset
31:0	TXBUF	Tx buffer data.	W	0x0000 0000

24.7 SuperSpeed USB DRD

This section describes the SuperSpeed (SS) USB 3.0 Dual-Role-Device (DRD) subsystem of the device.

Note

This chapter describes a module (subsystem) in the superset device. The available number of instances and supported set of features is device part number dependent. Refer to device *Data Manual*, for more information.

24.7.1 SuperSpeed USB DRD Subsystem Overview

SuperSpeed USB DRD Subsystem has four instances in the device providing the following functions:

- USB1: SuperSpeed (SS) USB 3.0 Dual-Role-Device (DRD) subsystem with integrated SS (USB3.0) PHY and HS/FS (USB2.0) PHY
- USB2: High-Speed (HS) USB 2.0 Dual-Role-Device (DRD) subsystem with integrated HS/FS PHY
- USB3: HS USB 2.0 Dual-Role-Device (DRD) subsystem with ULPI (SDR) interface to external HS/FS PHYs
- USB4: HS USB 2.0 Dual-Role-Device (DRD) subsystem with ULPI (SDR) interface to external HS/FS PHYs.

Figure 24-143 shows the USB1 subsystem overview.

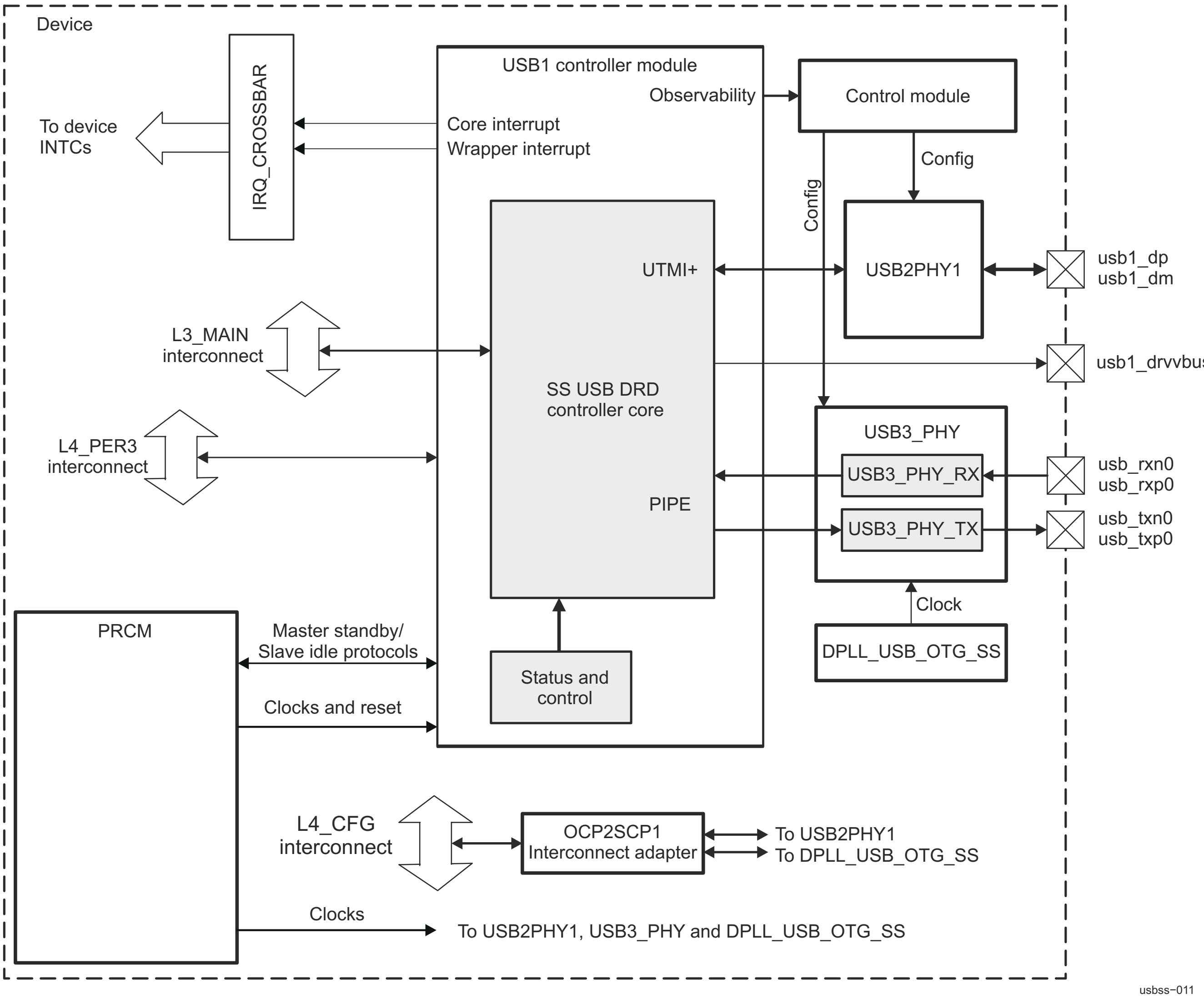


Figure 24-143. USB1 Highlight

Figure 24-144 shows the USB2 subsystem overview.

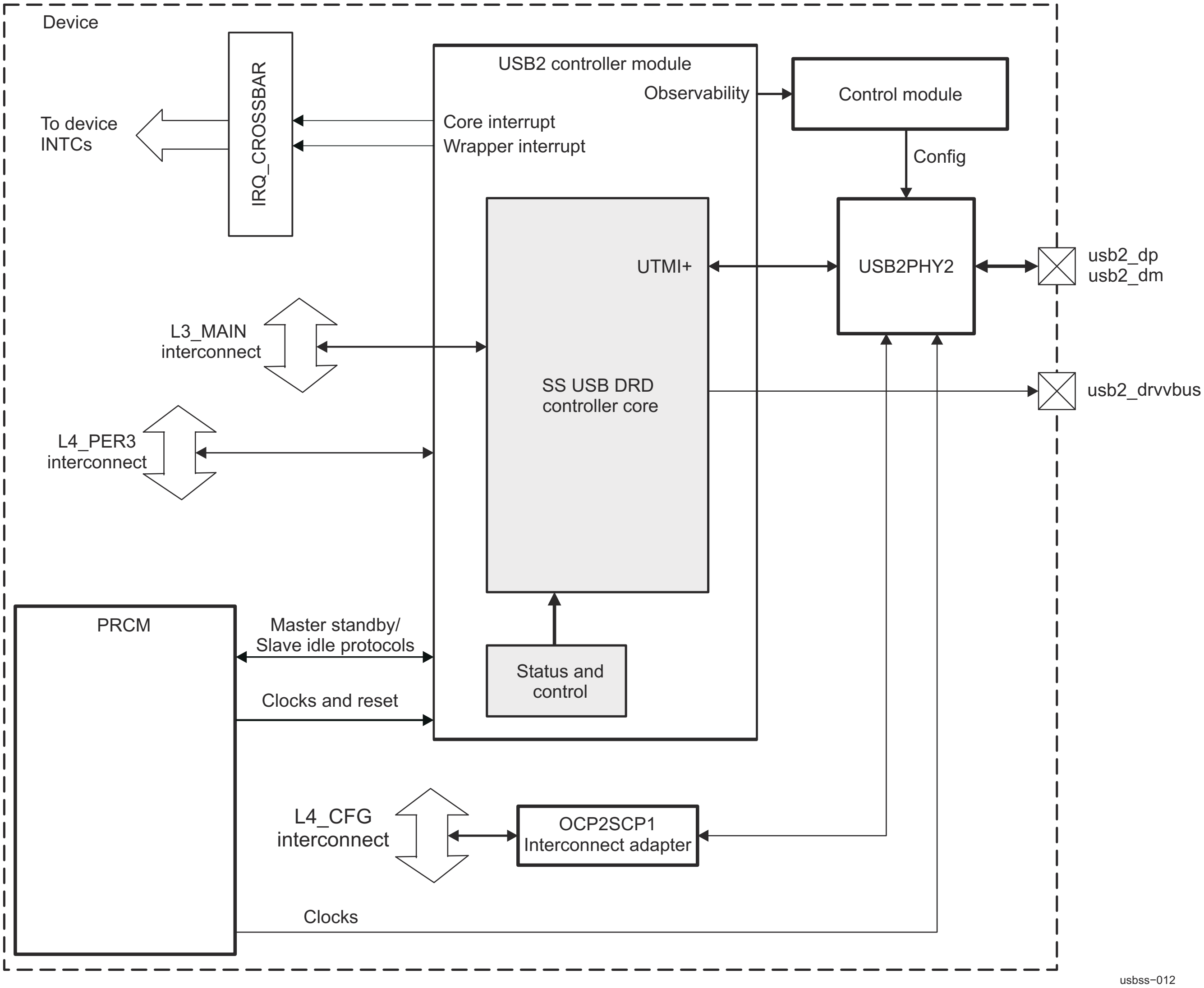


Figure 24-144. USB2 Highlight

Figure 24-145 shows the USB3 and USB4 subsystem overview.

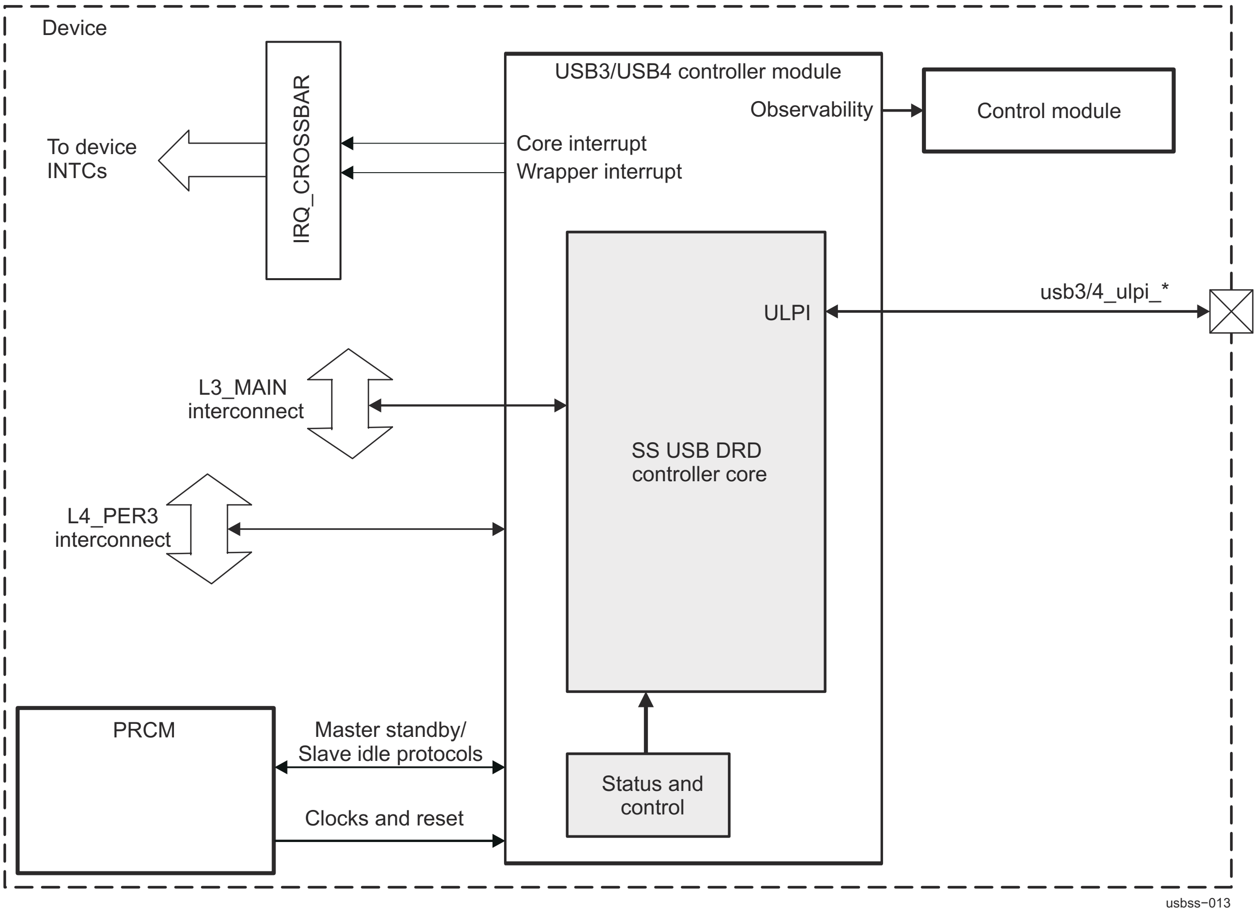


Figure 24-145. USB3 and USB4 Highlight

24.7.1.1 Main Features

SuperSpeed USB DRD Subsystem has the following features:

- Dual-role-device (DRD) capability:
  - Supports USB Peripheral (or Device) mode at speeds SS (5Gbps)(USB1 only), HS (480 Mbps), and FS (12 Mbps)
  - Supports USB Host mode at speeds SS (5Gbps)(USB1 only), HS (480 Mbps), FS (12 Mbps), and LS (1.5 Mbps)
  - USB static peripheral operation
  - USB static host operation
  - Flexible stream allocation
  - Stream priority
  - External Buffer Control
- Each instance contains single xHCI controller with the following features:
  - Internal DMA controller
  - Descriptor caching and data prefetching
  - Interrupt moderation and blocking
  - Power management USB3.0 states for U0, U1, U2, and U3
  - Dynamic FIFO memory allocation for all endpoints
  - Supports all modes of transfers (control, bulk, interrupt, and isochronous)
  - Supports high bandwidth ISO mode
- Connects to an external charge pump for VBUS 5 V generation

- USB-HS PHY (USB2PHY1 and USB2PHY2 for USB1 and USB2, respectively): contain the USB functions, drivers, receivers, and pads for correct D+/D– signalling
- USB3_PHY. The USB3_PHY is embedded in the USB1 subsystem and contains:
  - USB3_PHY_RX deserializer to receive data at SuperSpeed mode
  - USB3_PHY_TX serializer to transmit data at SuperSpeed mode
  - Power sequencer that contains a power control state machine, generating the sequences to power up/down the USB3_PHY_RX/USB3_PHY_TX
  - Dedicated DPLL (DPLL_USB_OTG_SS)

24.7.2 SuperSpeed USB DRD Subsystem Environment

24.7.2.1 SuperSpeed USB DRD Subsystem I/O Interfaces

Figure 24-146 describes the I/O signals of the SuperSpeed USB subsystem interfaces.

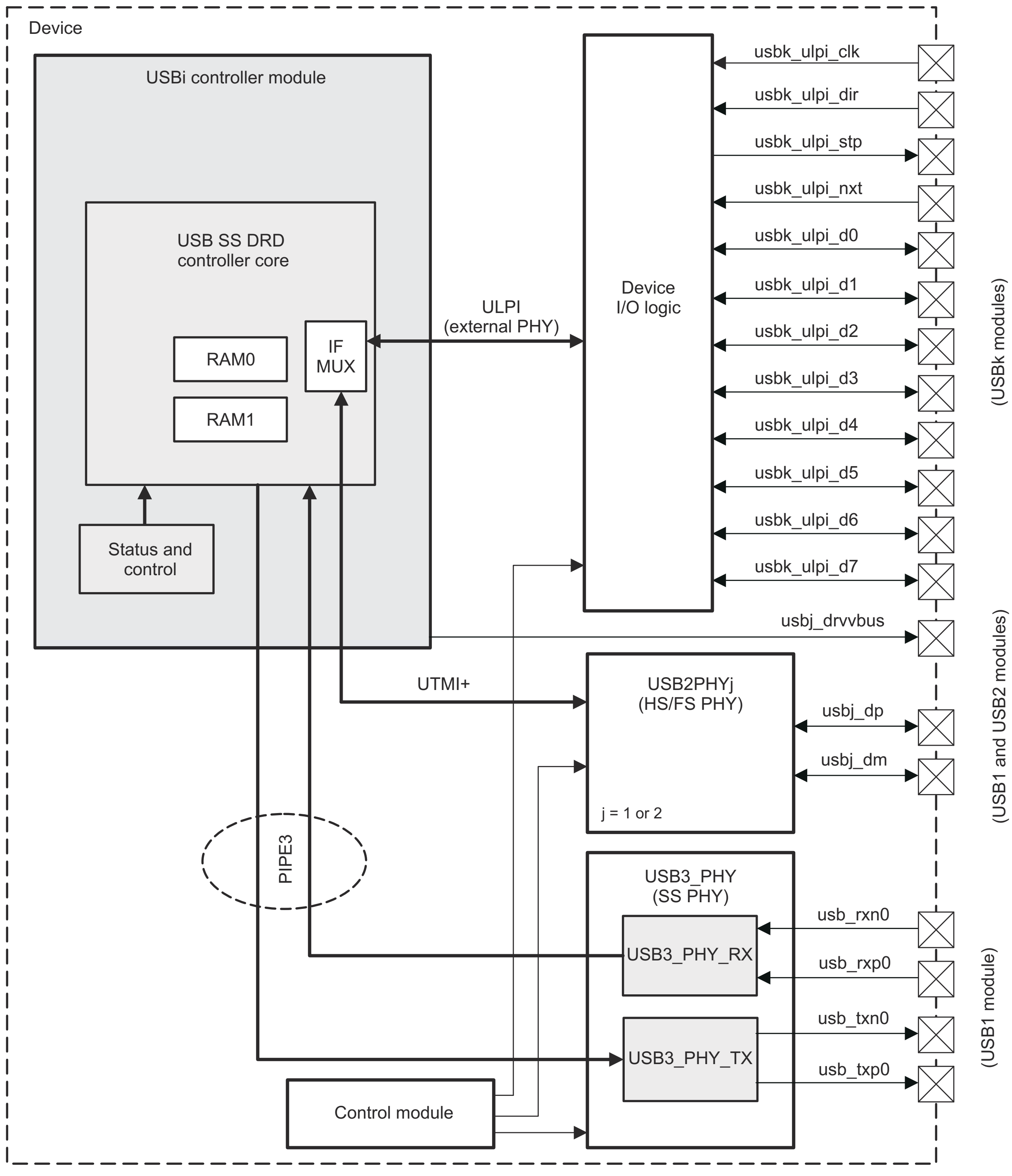


Figure 24-146. SuperSpeed USB Subsystem Environment

i = 1 to 4

k = 3 or 4

Table 24-432 through Table 24-435 describe the I/O signals of the SuperSpeed USB subsystem interfaces shown in Figure 24-146

Table 24-432. USB1 Input/Output Description

Module Pin	Device-Level Signal Name	I/O[(1)]	Description	Reset Value
USB2PHY1				
DRVVBUS	usb1_drvvbus	O	Drive-VBUS enable to external charge pump/power switch	0
DP	usb1_dp	I/O	USB2.0 half-duplex differential pair	HiZ
DM	usb1_dm	I/O		HiZ
USB3_PHY				
TX	usb_txn0	O	USB3.0 transmitter differential pair	HiZ
TY	usb_txp0	O		HiZ
RX	usb_rxn0	I	USB3.0 receiver differential pair	HiZ
RY	usb_rxp0	I		HiZ

(1) I = Input; O = Output; I/O = Bidirectional

Table 24-433. USB2 Input/Output Description

Module Pin	Device-Level Signal Name	I/O	Description	Reset Value
USB2PHY2				
DRVVBUS	usb2_drvvbus	O	Drive-VBUS enable to external charge pump/power switch	0
DP	usb2_dp	I/O	USB2.0 half-duplex differential pair	HiZ
DM	usb2_dm	I/O		HiZ

Table 24-434. USB3 Input/Output Description

Module Pin	Device-Level Signal Name	I/O	Description	Reset Value
ULPI interface				
ULPI_CLK	usb3_ulpi_clk	I	Clock input from external transceiver	HiZ
ULPI_DIR	usb3_ulpi_dir	I	Data direction control from external transceiver	HiZ
ULPI_STP	usb3_ulpi_stp	O	Output to external transceiver to stop data stream	1
ULPI_NXT	usb3_ulpi_nxt	I	Next signal control from external transceiver	HiZ
ULPI_DATA0	usb3_ulpi_d0	I/O	Data bit 0 to/from external transceiver	HiZ
ULPI_DATA1	usb3_ulpi_d1	I/O	Data bit 1 to/from external transceiver	HiZ
ULPI_DATA2	usb3_ulpi_d2	I/O	Data bit 2 to/from external transceiver	HiZ
ULPI_DATA3	usb3_ulpi_d3	I/O	Data bit 3 to/from external transceiver	HiZ
ULPI_DATA4	usb3_ulpi_d4	I/O	Data bit 4 to/from external transceiver	HiZ
ULPI_DATA5	usb3_ulpi_d5	I/O	Data bit 5 to/from external transceiver	HiZ
ULPI_DATA6	usb3_ulpi_d6	I/O	Data bit 6 to/from external transceiver	HiZ
ULPI_DATA7	usb3_ulpi_d7	I/O	Data bit 7 to/from external transceiver	HiZ

Table 24-435. USB4 Input/Output Description

Module Pin	Device-Level Signal Name	I/O	Description	Reset Value
ULPI interface				
ULPI_CLK	usb4_ulpi_clk	I	Clock input from external transceiver	HiZ

Table 24-435. USB4 Input/Output Description (continued)

Module Pin	Device-Level Signal Name	I/O	Description	Reset Value
ULPI_DIR	usb4_ulpi_dir	I	Data direction control from external transceiver	HiZ
ULPI_STP	usb4_ulpi_stp	O	Output to external transceiver to stop data stream	1
ULPI_NXT	usb4_ulpi_nxt	I	Next signal control from external transceiver	HiZ
ULPI_DATA0	usb4_ulpi_d0	I/O	Data bit 0 to/from external transceiver	HiZ
ULPI_DATA1	usb4_ulpi_d1	I/O	Data bit 1 to/from external transceiver	HiZ
ULPI_DATA2	usb4_ulpi_d2	I/O	Data bit 2 to/from external transceiver	HiZ
ULPI_DATA3	usb4_ulpi_d3	I/O	Data bit 3 to/from external transceiver	HiZ
ULPI_DATA4	usb4_ulpi_d4	I/O	Data bit 4 to/from external transceiver	HiZ
ULPI_DATA5	usb4_ulpi_d5	I/O	Data bit 5 to/from external transceiver	HiZ
ULPI_DATA6	usb4_ulpi_d6	I/O	Data bit 6 to/from external transceiver	HiZ
ULPI_DATA7	usb4_ulpi_d7	I/O	Data bit 7 to/from external transceiver	HiZ

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see *Pad Configuration Registers* in *Control Module*.

24.7.2.2 SuperSpeed USB Subsystem Application

24.7.2.2.1 USB3.0 DRD Application

Figure 24-147 shows a typical application using the SuperSpeed USB subsystem. The application represents a full-featured integration and allows all of the capabilities of the controller used.

VBUS sensing can be exported to the external Power-Management IC (PMIC) in case it supports this function. This would eliminate the need for additional components such as level shifters and comparators. In this case, software driver must be modified to be able to react on PMIC interrupts for VBUS-rise and VBUS-fall through the I2C1 interface.

Some PMICs support the VBUS_DETECT output from the VBUS comparator to be muxed directly on their GPIO pin. In this case PMIC VBUS_DETECT can be connected to a SoC GPIO pin, thus, this functionally is identical to Figure 24-147.

See the respective PMIC datasheet for details.

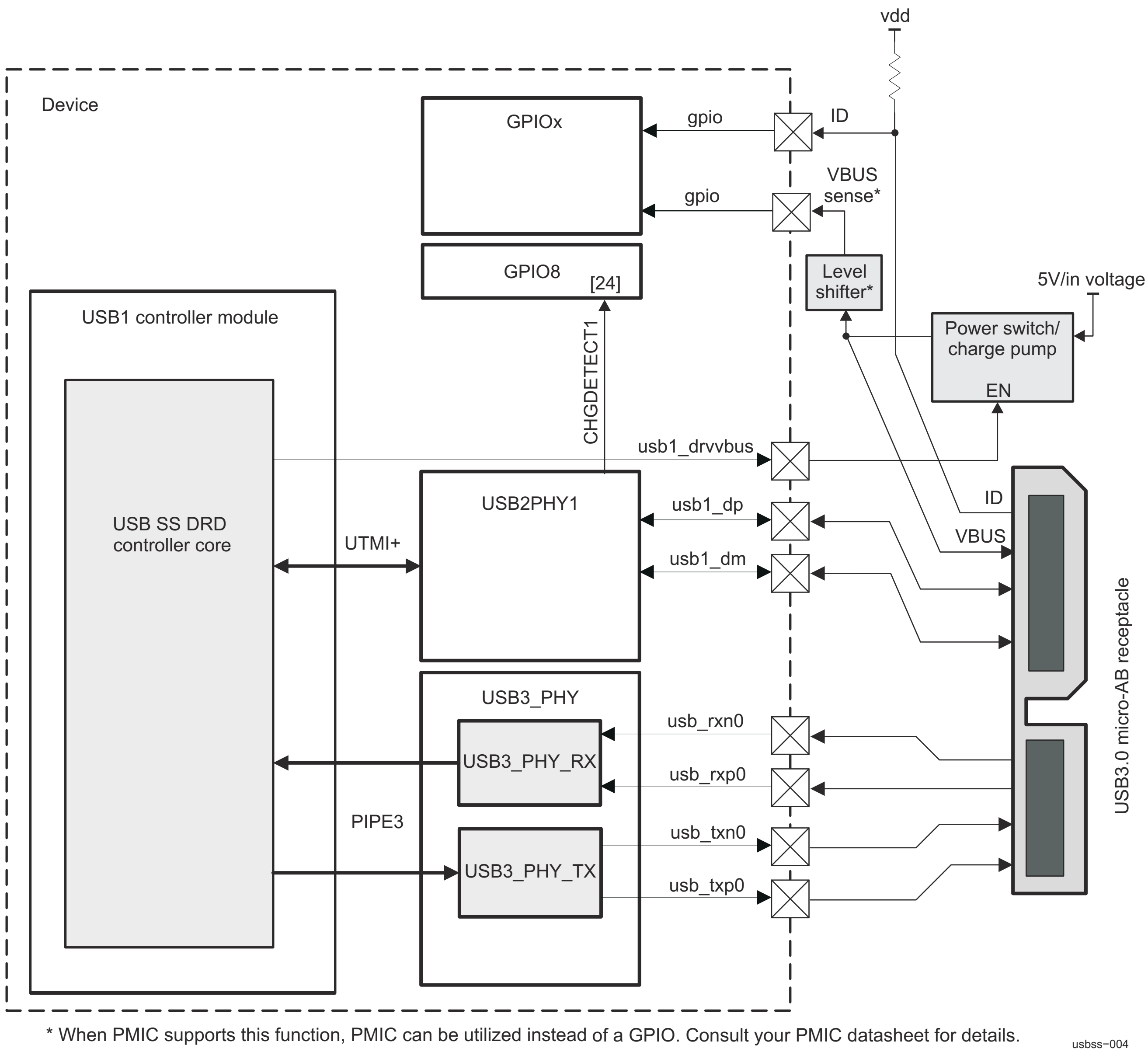


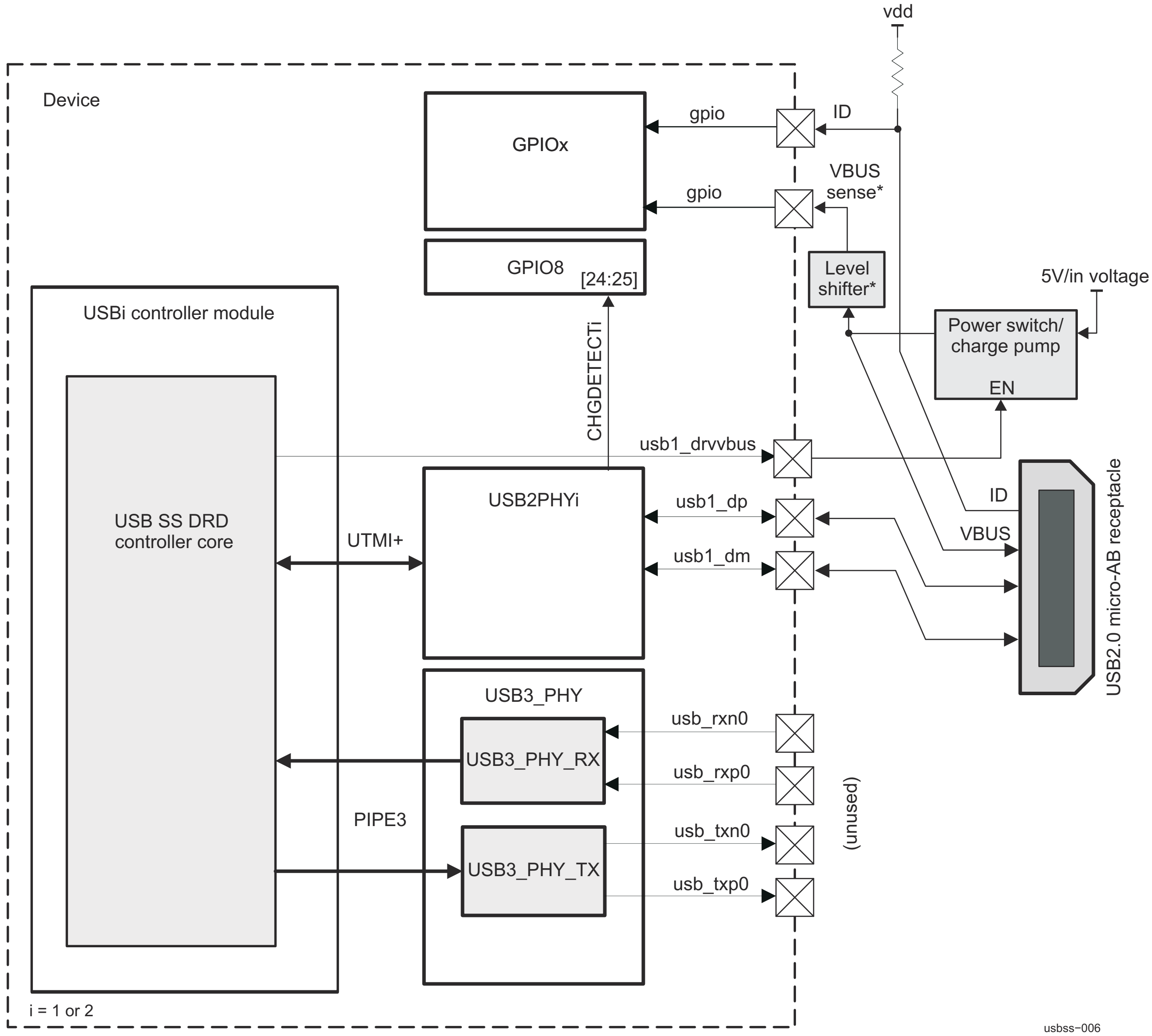
Figure 24-147. SuperSpeed USB Controller Application: USB3.0 DRD

The USB3.0 micro-AB receptacle characterizes a USB3.0 DRD, and allows the following attachments:

- Micro-A-type plugs, to attach to a USB peripheral/hub upstream port (or to another DRD, as host)
- Micro-B-type plugs, to attach to a USB host/hub downstream port (or to another DRD, as peripheral)
- To USB3.0 devices: The USB3.0 micro plug covers all contacts of the receptacle.
- To USB2.0 devices: The USB2.0 micro plug occupies only the wider, USB2.0 half of the receptacle. SuperSpeed signal pairs remain floating.

24.7.2.2.2 USB2.0 DRD Internal PHY

Figure 24-148 shows a USB2.0-only integration in which the USB3.0 capability is not available. Similarly to Figure 24-147, VBUS sensing can be exported to the companion PMIC.



* When PMIC supports this function, PMIC can be utilized instead of a GPIO. Consult your PMIC datasheet for details.

Figure 24-148. SuperSpeed USB Controller Application: USB2.0 DRD (Internal PHY)

The USB2.0 micro-AB receptacle characterizes a USB2.0 DRD and accepts the following:

- Micro-A-type plugs, which attach to a USB peripheral/hub upstream port (or another DRD, as host)
- Micro-B-type plugs, which attach to a USB host/hub downstream port (or another DRD, as peripheral)
- USB2.0 devices only: The USB2.0 micro plug covers all contacts of the receptacle.
- USB3.0 devices: The USB3.0 micro plug is incompatible and the SuperSpeed signal pairs of the cable remain floating.

24.7.2.2.3 USB2.0 DRD External PHY

Figure 24-149 shows an external ULPI PHY IC is used instead of the on-chip UTMI PHY (USB2PHY). Seen from outside the system, both integrations are functionally equivalent. Internally, the functions of the ID and VBUS are taken over by the ULPI PHY. However, power on VBUS can be supplied by an VBUS switch with more power than the ULPI PHY is capable of.

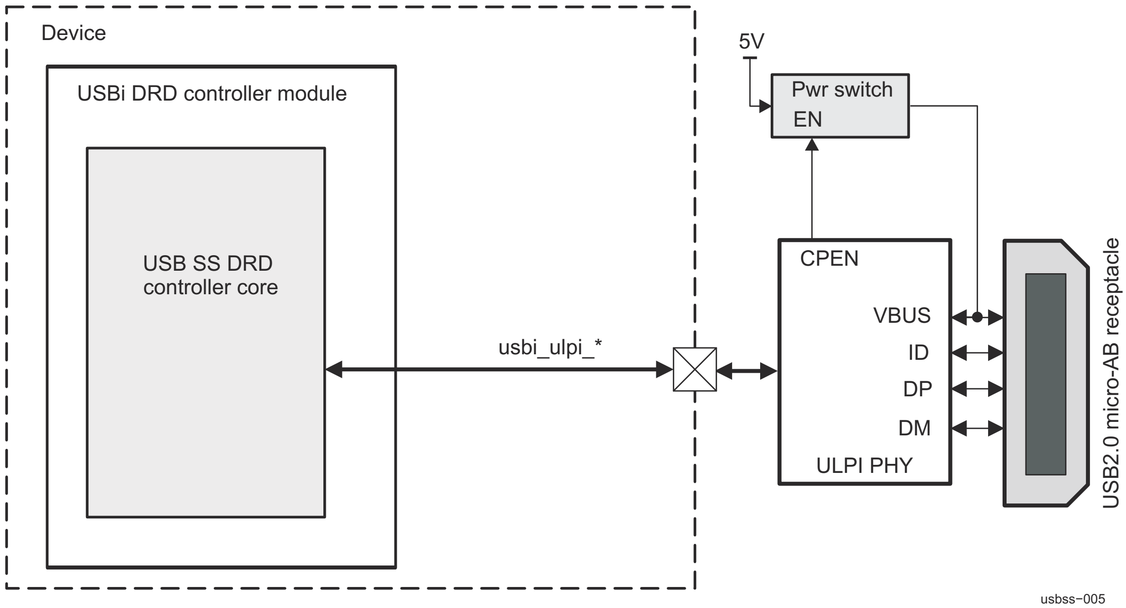


Figure 24-149. SuperSpeed USB Controller Application: USB2.0 DRD (ULPI PHY)

24.7.2.2.4

i = 3 or 4

24.7.2.2.5 Host Mode

In always-host mode:

- ID pin can be omitted. Driver must be set explicitly to operate in host mode
- USB receptacle used must be type-A USB3.0 or USB2.0 respectively.

24.7.2.2.6 Device Mode

In always-device (peripheral) mode

- ID pin can be omitted. Driver must be set explicitly to operate in device mode
- VBUS power switch is not needed. USB devices never drive VBUS
- USB receptacle used must be type-B standard/micro USB3.0 or USB2.0 respectively.

24.7.3 SuperSpeed USB Subsystem Integration

The L3 (master) interconnect generates data traffic within the device. The L4 (slave) interconnect is a configuration port for register setting.

Figure 24-150 shows the SuperSpeed USB subsystem integration in the device.

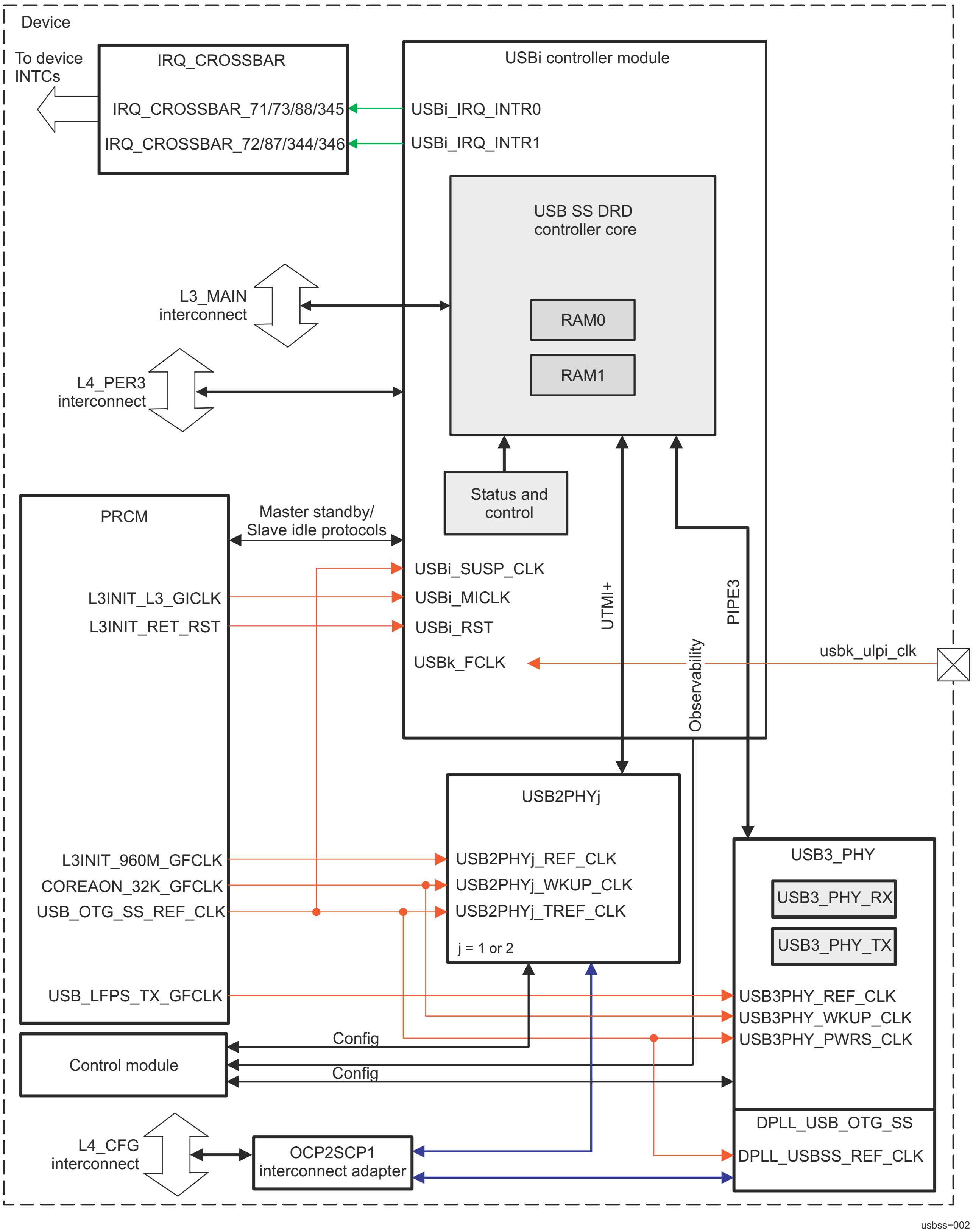


Figure 24-150. SuperSpeed USB Subsystem Integration

i = 1 to 4

k = 3 or 4

Table 24-436 through Table 24-438 summarize the integration of the module in the device.

Table 24-436. SuperSpeed USB Subsystem Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
USB1	PD_L3INIT	L3_MAIN L4_PER3
USB2	PD_L3INIT	L3_MAIN L4_PER3
USB3	PD_L3INIT	L3_MAIN L4_PER3
USB4	PD_L3INIT	L3_MAIN L4_PER3
USB2PHY1	PD_L3INIT	L4_CFG
USB2PHY2	PD_L3INIT	L4_CFG
USB3_PHY	PD_L3INIT	L4_CFG

Table 24-437 lists the clocks provided to the SuperSpeed USB subsystem.

Table 24-437. SuperSpeed USB Subsystem Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
USB1	USB1_FCLK	USB2PHY1_UTMI_CLK	USB2PHY1	60-MHz UTMI clock from PHY
	USB1_MICLK	L3INIT_L3_GICLK	PRCM	L3 interconnect clock, for the L3 master port interface *CD_L3INIT Clock Domain* in *Power, Reset and Clock Management*.
	USB1_SUSP_CLK	USB_OTG_SS_REF_CLK	PRCM	Suspend clock
USB2	USB2_FCLK	USB2PHY2_UTMI_CLK	USB2PHY2	60-MHz UTMI clock from PHY
	USB2_MICLK	L3INIT_L3_GICLK	PRCM	L3 interconnect clock, for the L3 master port interface.
	USB2_SUSP_CLK	USB_OTG_SS_REF_CLK	PRCM	Suspend clock
	USB2_125M_CLK	L3INIT_60M_FCLK	PRCM	125-MHz clock for the non-USB3.0 instances.
USB3	USB3_FCLK	OTG_60M_FCLK	usb3_ulpi_clk pad	60-MHz ULPI clock from PHY
	USB3_MICLK	L3INIT_L3_GICLK	PRCM	L3 interconnect clock, for the L3 master port interface .
	USB3_SUSP_CLK	USB_OTG_SS_REF_CLK	PRCM	Suspend clock
	USB3_125M_CLK	L3INIT_60M_FCLK	PRCM	125-MHz clock for the non-USB3.0 instances.
USB4	USB4_FCLK	OTG_60M_FCLK	usb4_ulpi_clk pad	60-MHz ULPI clock from PHY
	USB4_MICLK	L3INIT_L3_GICLK	PRCM	L3 interconnect clock, for the L3 master port interface .
	USB4_SUSP_CLK	USB_OTG_SS_REF_CLK	PRCM	Suspend clock
	USB4_125M_CLK	L3INIT_60M_FCLK	PRCM	125-MHz clock for the non-USB3.0 instances.
USB2PHY1	USB2PHY1_REF_CLK	L3INIT_960M_GFCLK	PRCM	Functional REF 960-MHz clock (from the DPLL_USB, PRCM controlled)
	USB2PHY1_WKUP_CLK	FUNC_32K_CLK	PRCM	Wakeup 32-kHz functional clock

Table 24-437. SuperSpeed USB Subsystem Clocks and Resets (continued)

	USB2PHY1_TREF_CLK	USB_OTG_SS_REF_CLK	PRCM	Functional TREF clock derived from SYS_CLK1
USB2PHY2	USB2PHY2_REF_CLK	L3INIT_960M_GFCLK	PRCM	Functional REF 960-MHz clock (from the DPLL_USB, PRCM controlled)
	USB2PHY2_WKUP_CLK	FUNC_32K_CLK	PRCM	Wakeup 32-kHz functional clock
	USB2PHY2_TREF_CLK	USB_OTG_SS_REF_CLK	PRCM	Functional TREF clock derived from SYS_CLK1
USB3_PHY	USB3PHY_REF_CLK	USB_LFPS_TX_GFCLK	PRCM	Fixed-frequency USB3 transmitter REF functional clock
	DPLL_USBSS_REF_CLK	USB_OTG_SS_REF_CLK	PRCM	Functional DPLL REF clock derived from SYS_CLK1
	USB3PHY_WKUP_CLK	COREAON_32K_GFCLK	PRCM	Wakeup and debounce 32-kHz functional clock
		Resets		
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
USB1	USB1_RST	L3INIT_RET_RST	PRCM	USB1 controller module hardware retention reset
USB2	USB2_RST	L3INIT_RET_RST	PRCM	USB2 controller module hardware retention reset
USB3	USB3_RST	L3INIT_RET_RST	PRCM	USB3 controller module hardware retention reset
USB4	USB4_RST	L3INIT_RET_RST	PRCM	USB4 controller module hardware retention reset

Table 24-438 lists the interrupt lines that are driven out from the SuperSpeed USB controller modules.

Table 24-438. SuperSpeed USB Subsystem Hardware Requests

		Interrupt Requests		
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
USB1	USB1_IRQ_INTR0	IRQ_CROSSBAR_71	MPU_IRQ_76	USB1 main (core) interrupt request
	USB1_IRQ_INTR1	IRQ_CROSSBAR_72	MPU_IRQ_77 IPU1_IRQ_73 IPU2_IRQ_73	USB1 wrapper interrupt request
USB2	USB2_IRQ_INTR0	IRQ_CROSSBAR_73	MPU_IRQ_78 IPU1_IRQ_74 IPU2_IRQ_74	USB2 main (core) interrupt request
	USB2_IRQ_INTR1	IRQ_CROSSBAR_87	MPU_IRQ_92 IPU1_IRQ_76 IPU2_IRQ_76	USB2 wrapper interrupt request
USB3	USB3_IRQ_INTR0	IRQ_CROSSBAR_88	MPU_IRQ_93 IPU1_IRQ_77 IPU2_IRQ_77	USB3 main (core) interrupt request
	USB3_IRQ_INTR1	IRQ_CROSSBAR_344	-	USB3 wrapper interrupt request
USB4	USB4_IRQ_INTR0	IRQ_CROSSBAR_345	-	USB4 main (core) interrupt request
	USB4_IRQ_INTR1	IRQ_CROSSBAR_346	-	USB4 wrapper interrupt request

Note

The **Default Mapping** column in Table 24-438 shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module. For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

24.8 SATA Controller

24.8.1 SATA Controller Overview

The SATA host controller handles data interactions between a local host system memory and a SATA mass storage device with minimal local host (LH) intervention.

In contrast to the parallel 16-bit - ATA (PATA) interface, the SATA interface takes advantage of serial data transmission/reception over a differential pair of conductors. SATA uses the command set from the ATA/ATAPI-6 standard augmented with native command queuing (NCQ) commands optimized for the serialized interface.

The device has one embedded SATA host bus adapter (HBA) controller with a single port. The device SATA host subsystem is composed of several functional components (see Figure 24-151):

- The core component of the SATA host controller subsystem (signified as SATA controller on the Figure 24-151) implements the transport and link layers of the SATA interface protocol, i.e. all SATA media access control (MAC) functionalities.
- SATA_PHY encompasses the physical layer (PHY) components - serializer, de-serializer, etc. which adapt the generated by the SATA controller MAC logic parallel 10-bit output data stream for serial electrical transmission and reception.
- DPLL_SATA is a programmable (through registers of the integrated PLL controller - PLLCTRL_SATA) DPLL clock source that provides a high-speed clock to the SATA_PHY serializer/de-serializer components.
- An L4_CFG interface adapter - OCP2SCP3, which enables accessing the PLLCTRL_SATA registers via L4_CFG interconnect accesses.

For details regarding the device SATA host subsystem components - SATA_PHY , PLLCTRL_SATA, DPLL_SATA and interface adapter), see Section 26.1.1, *SATA PHY Subsystem Overview*, in Chapter 26, *Shared PHY Component Subsystems*.

Figure 24-151 shows an overview of the device-embedded SATA host controller subsystem.

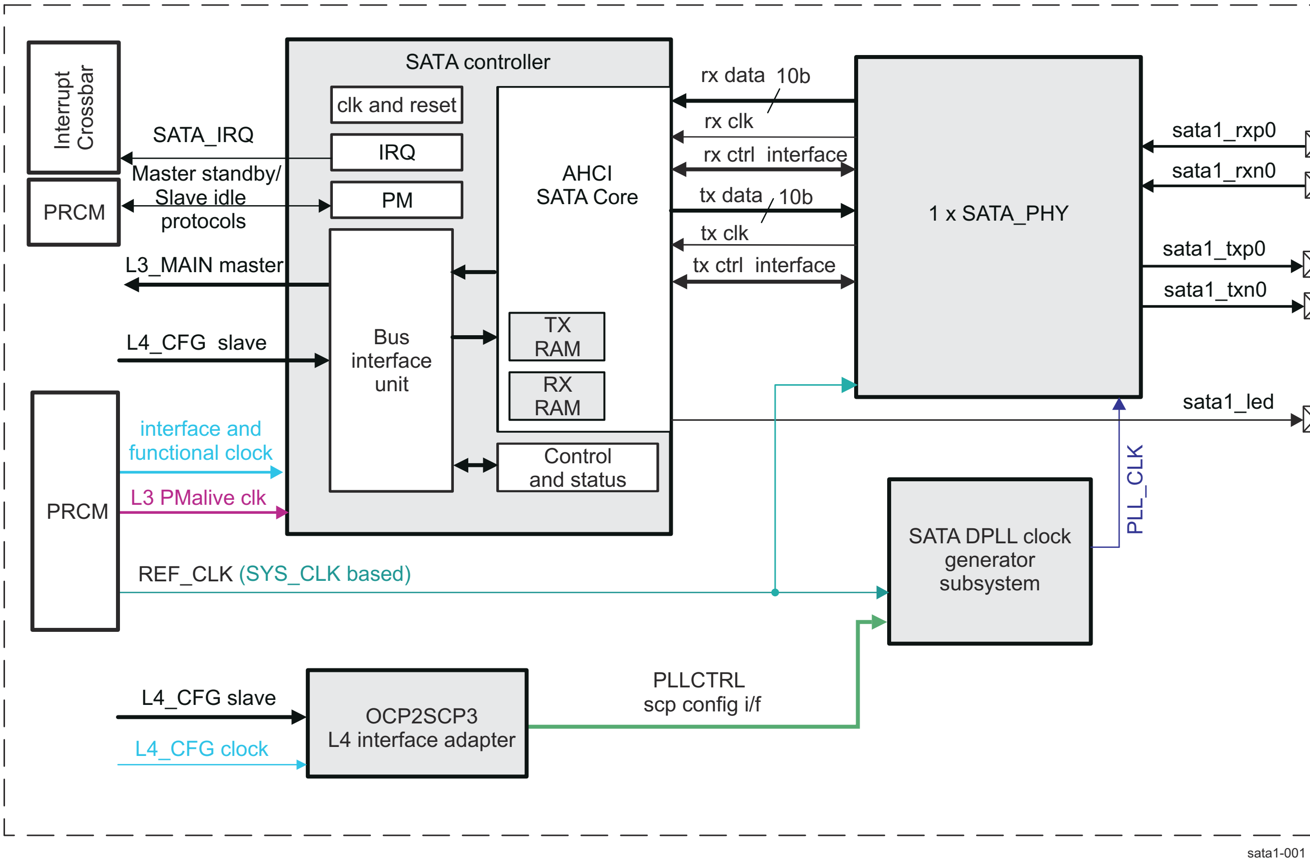


Figure 24-151. SATA Host Controller Subsystem Overview

24.8.1.1 SATA Controller

The SATA controller is the main functional component of the device-embedded SATA (Host Bus Adapter) HBA and is based on advanced host controller interface (AHCI) module. The SATA controller handles data transactions at the link and transport layers of the SATA interface using the advanced host controller interface (AHCI) mechanism. The SATA controller core engine is a generation 2-compliant host (supporting 3 Gbps transfer speed) with integrated DMA and FIFO RAM buffers.

The SATA controller AHCI-based interactions involve extensive DMA processing of both data and command transfers, reducing much of the user overhead associated with standard ATA task file register servicing. The SATA HBA port programmable DMA acts as a master on the device L3_MAIN interconnect, which facilitates direct command/data transfers between host system memory and attached SATA storage devices.

The SATA controller has a slave configuration port accessible on the device L4_CFG interconnect. This port provides the user with an appropriate register interface for HBA setup, control and status, interrupt settings, DMA configuration, etc.

24.8.1.1.1 AHCI Mode Overview

The SATA core supports the AHCI hardware mechanism, which provides the user software with a suitable HBA register interface to manage SATA interface operations. Generally, the AHCI introduces a system memory structure that includes some generic control and status area and a list of command entries (which can have a depth from 1 up to 32 entries) assigned per HBA port. Each of the command list entries contains information necessary to program a SATA device and pointers to data transfer descriptors.

The AHCI mode of operation disregards the master/slave communication model. A certain SATA AHCI host controller port establishes a point-to-point connectivity with only a single SATA peripheral device at a time, which is always a master device.

24.8.1.1.2 Native Command Queuing

The SATA AHCI engine supports NCQ and is capable of command queued protocol interactions with NCQ-compliant SATA mass storage devices. The NCQ commands generated by a SATA HBA port are loaded into command queues maintained at the peripheral device. The commands are stored in a queue and subsequently fetched and processed by peripheral device controller in sequences, which imply more native and highly-optimized for the device order of execution. The synchronization between an NCQ-aware SATA HBA and an NCQ-aware peripheral storage device involves implementation of the so called FPDMA queued command protocol, which ensures that HBA posts NCQ commands to the target SATA device in its demanded NCQ order.

24.8.1.1.3 SATA Transport Layer Functionalities

The SATA controller handles all of the transport layer functions of the SATA protocol. During reception it receives a frame information structure (FIS) from its link layer through the RX FIFO, decodes the type, and routes it to the proper location through the port DMA. During transmission it transfers a FIS constructed by the port DMA to the link layer through the TX FIFO. It also passes link layer errors and checks for transport layer errors to pass up to the system. The transport layer also contains the TX and RX FIFOs. These FIFOs are used as asynchronous data buffers between the serial domain and the bus clock domain. The size of these FIFOs affects the subsystem ability to buffer data before flow control must be asserted. It also affects the maximum transaction size that can be programmed into the port DMA.

24.8.1.1.4 SATA Link Layer Functionalities

The link layer maintains the link and supports all SATA link layer functionality, including:

- Out-of-band (OOB) transmit signaling
- Frame negotiation and arbitration
- Envelope framing/deframing
- Cyclic redundancy check (CRC) calculation (receive and transmit)
- 8b/10b encoding/decoding
- Flow control
- Frame acknowledgment and status
- Data width conversion
- Data scrambling/descrambling
- Primitive transmission
- Primitive detection and dropping
- Power management

24.8.1.2 SATA Controller Features

This section describes the features supplied by the SATA controller module. The SATA controller complies with the following standards:

- *Serial ATA Standard* specification (revision 2.6)
- *Serial ATA Advanced Host Controller Interface* specification (revision 1.1). The device SATA host controller also complies with the *Serial ATA Advanced Host Controller Interface* specification (revision 1.3), excluding support for the port multiplier FIS-based switching feature.

The main features of the SATA host controller are:

- Serial ATA 1.5-Gbps and 3-Gbps speeds (SATA-1 and SATA-2)
- Integrated RxFIFO and TxFIFO RAM data buffers
- Support of all SATA power management features
- AHCI support of 64-bit addressing mode (see device limitation note below)
- HBA port associated Internal DMA engine
- Hardware-assisted NCQ for up to 32 entries
- Command completion coalescing (CCC) interrupts
- Support of port multiplier with command-based switching
- Activity LED generation

Additionally, link layer supports:

- 8b/10b encoding and decoding functionality
- RX elasticity buffer
- TX OOB sequence generation
- RX OOB sequence generation

The differences between the SATA host controller and a standard SATA host controller defined by the *Serial ATA Gold Standard* specification (v2.6) are:

- Staggered spin-up is not supported (only one HBA AHCI port embedded).
- Only one port is supported by the device-embedded SATA HBA from up to 32 possible, according to the standard.
- The SATA controller is AHCI-mode compliant only and does not support standard ATA legacy modes of operation.
- Cold presence detection signals are not available. The SATA controller targets support for permanently attached SATA drives only.

The following features are NOT supported by the SATA host subsystem:

- Far-end analog loopback
- Cold presence detect (CPD) signal is not available at the system level. As a consequence, the SATA device hot-plug operation is not supported.
- Mechanical presence switch signal is not available at system level.
- Message signaled interrupts
- PHY layer functionalities of SATA controller are not integrated. These are assigned to the SATA_PHY transceiver integrated at the device level (outside SATA controller module itself).

The SATA controller master (DMA) interface features:

- 32-bit data
- 36 bits of the AHCI master address bus (byte-aligned addressing) implemented in the device. See device limitation note below.
- 1-bit tag-ID port, but the tag value is 1 on all accesses (that is, the feature is not used)
- Sequential burst support (16 x 32 bit – Dwords)

Note

Even though the SATA AHCI Controller supports 64-bit -addressing mode, only 36-lower bits of the 64-bit address bus are integrated (meaningful) in the device. This defines a 64-GiB AHCI master address space.

The SATA controller slave interface features:

- 32-bit data, 13-bit (byte-aligned) address
- Single access (no burst support)
- All accesses are expected to be 4 bytes wide, 32-bit aligned.

Note

Accesses smaller than 4 bytes wide (that is, byte enable patterns different from 4’b1111) are functional and do not generate an error. Misaligned addresses do not generate an error.

As can be seen in Figure 24-151, the SATA Controller has all events merged to a single interrupt output - SATA_IRQ, mapped to the device Interrupt Crossbar.

24.8.2 SATA Controller Environment

Apart from the SATA communication-dedicated electrical connections to the SATA_PHY component, there is an additional SATA-defined-led drive generation signal that can be output at the device pad: *spi1_cs[1]* . For more information about this signal pad configuration register mapping, see *Pad Configuration Registers* in *Control Module*. For more details regarding the behavior of activity LED generation, see Section 24.8.4.10, *Activity LED Generation Functionality*.

Table 24-439. SATA Controller I/O Signals

Module Pin Name	Device Level Signal Name	I/O[(1)]	Description	Reset Value
TXP[(2)]	sata1_txp0	O	TXP output of the SATA PHY differential transmission line	0
TXN[(2)]	sata1_txn0	O	TXN output of the SATA PHY differential transmission line	0
RXP[(2)]	sata1_rxp0	I	RXP input of the SATA PHY differential reception line	Hi-Z
RXN[(2)]	sata1_rxn0	I	RXN input of the SATA PHY differential reception line	Hi-Z
ACT0_LED	sata1_led	O	HBA port activity LED indication	Hi-Z

(1) I = Input; O = Output
(2) These signals are part of the interface between SATA PHY serializer / de-serializer and the externally attached SATA storage device.

Note

The path from module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. For more information, refer to the sections *Pad Configuration Rregisters*, and *Control Module Register Manual*, in *Control Module*, for more information.

The Figure 24-152 shows the SATA host controller subsystem environment summarizing the interface signals exported directly from SATA controller at the device boundary, as well as those exported after SATA_PHY processing at the device boundary. For more information on the interface between the SATA controller and SATA_PHY , see Chapter 26, *Shared PHY Component Subsytsems*.

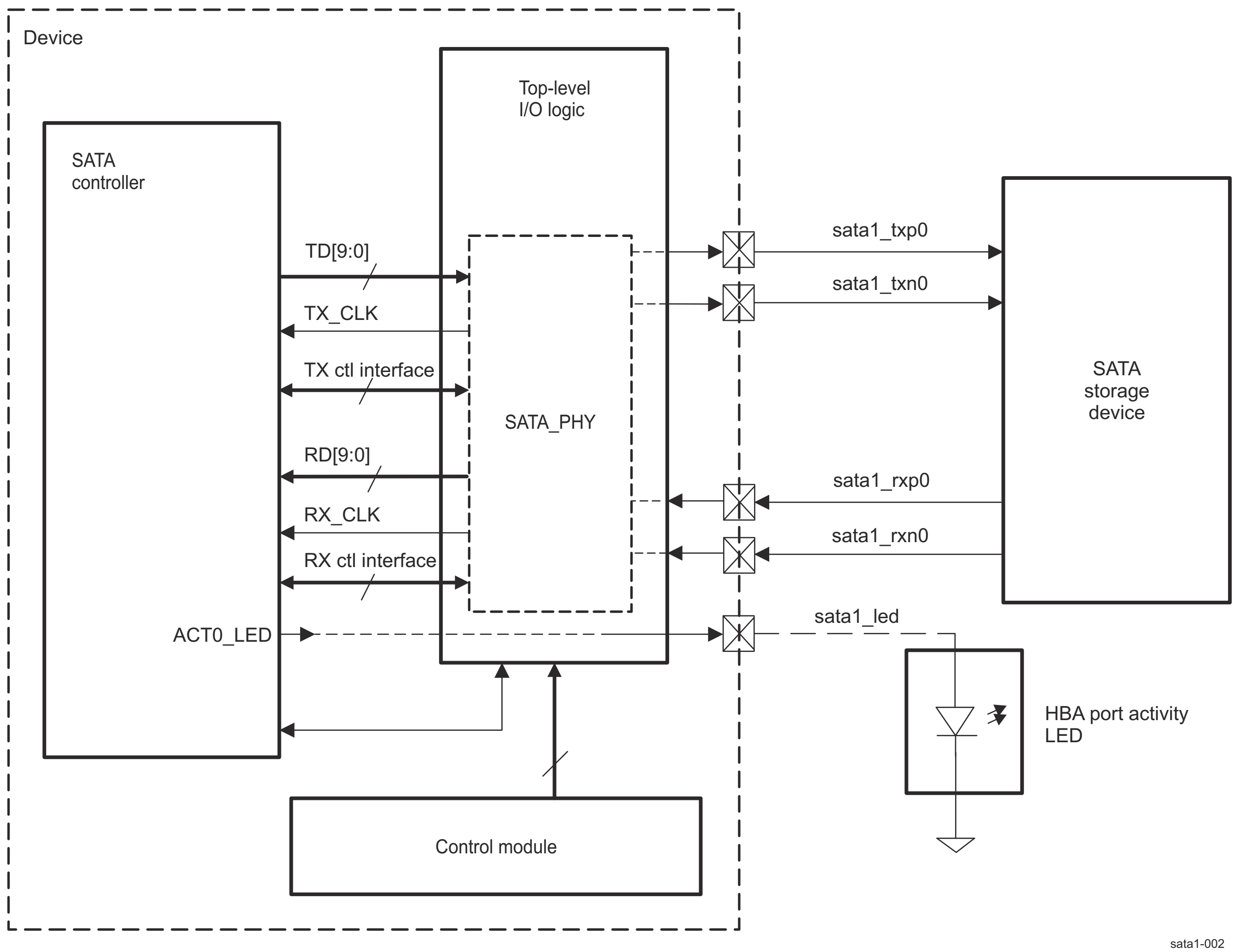


Figure 24-152. SATA Subsystem Environment

24.8.3 SATA Controller Integration

This section describes the SATA AHCI host controller integration in the device, including information about clocks, resets, and hardware requests. Figure 24-153 shows the SATA controller integration.

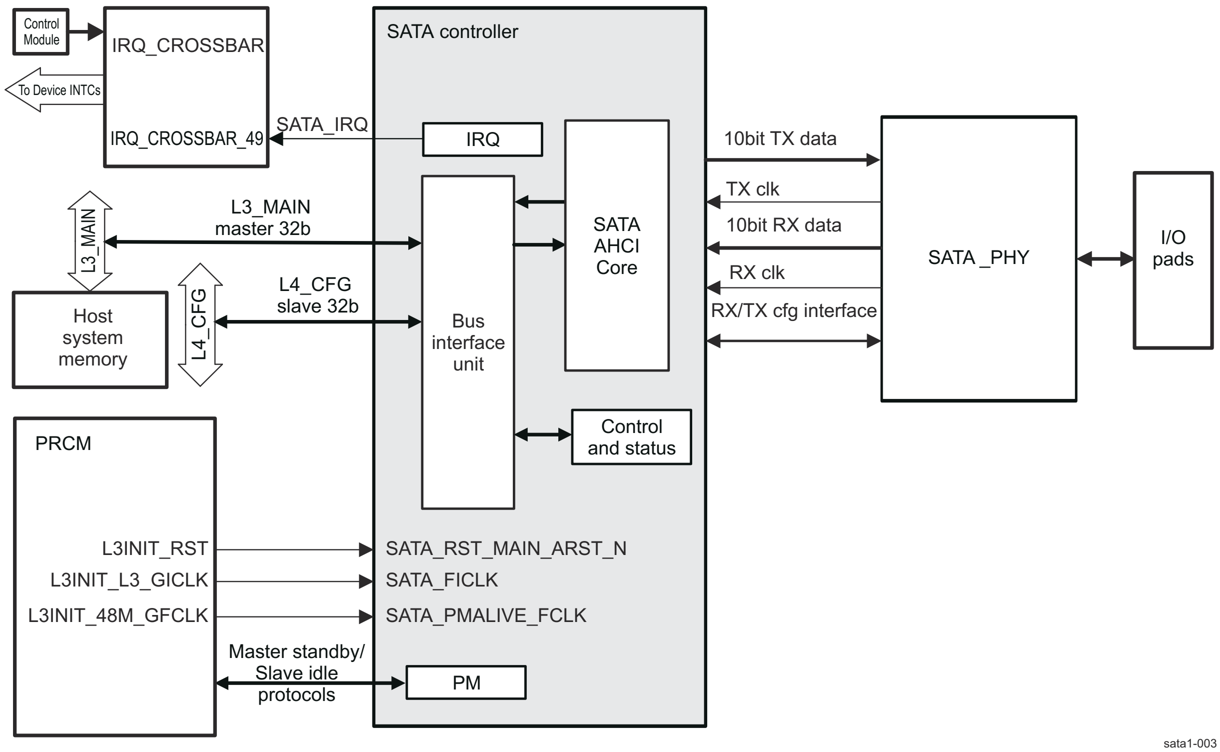


Figure 24-153. SATA Controller Integration

SATA controller integration includes these features:

- A single functional and interface clock (SATA_FICLK, shared between the master and the slave interfaces)
- A functional always-on clock (SATA_PMALIVE_FCLK to support entrance into and exit from the SATA defined low-power modes)
- A single hardware nonretention reset input (SATA_RST_MAIN_ARST_N)
- Master standby and slave idle protocols with the power, reset, and clock management (PRCM) module
- A single interrupt request generation line (SATA_IRQ, mapped to IRQ_CROSSBAR_49 line of IRQ CROSSBAR)
- No DMA or wakeup requests generation to surrounding modules
- L3_MAIN master interface
- L4_CFG-configuration slave interface

The SATA encoded symbol transmission/decoded symbol reception relies on two serial clocks , which are supplied indirectly from the DPLL_SATA through the SATA_PHY component.

Note

For more information about the slave idle protocol and the wakeup request, see *Module-Level Clock Management* in *Power, Reset, and Clock Management*.

Table 24-440 through Table 24-442 summarize the integration of the module in the device.

Table 24-440. SATA Controller Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect

Table 24-440. SATA Controller Integration Attributes (continued)

SATA	PD_L3INIT	L4_CFG L3_MAIN

Table 24-441. SATA Controller Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
SATA	SATA_FICLK	L3INIT_L3_GICLK	PRCM	SATA interface and functional clock
	SATA_PMALIVE_FCLK	L3INIT_48M_GFCLK	PRCM	Always-on SATA-specific, power-management support clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
SATA	SATA_RST_MAIN_ARST_N	L3INIT_RST	PRCM	A nonretention hardware reset

Table 24-442. SATA Controller Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
SATA	SATA_IRQ	IRQ_CROSSBAR_49	MPU_IRQ_54 DSP1_IRQ_80 DSP2_IRQ_80	SATA Controller IRQ line

Note

The "**Default Mapping**" column in Table 24-442 *SATA Controller Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

- For a description of the interrupt source, see *Interrupt Requests*.

Note

- The SATA HBA does not generate DMA and wakeup hardware requests to the surrounding modules integrated in the device

24.8.4 SATA Controller Functional Description

24.8.4.1 SATA Controller Block Diagram

Figure 24-154 shows a block diagram of the SATA host controller.

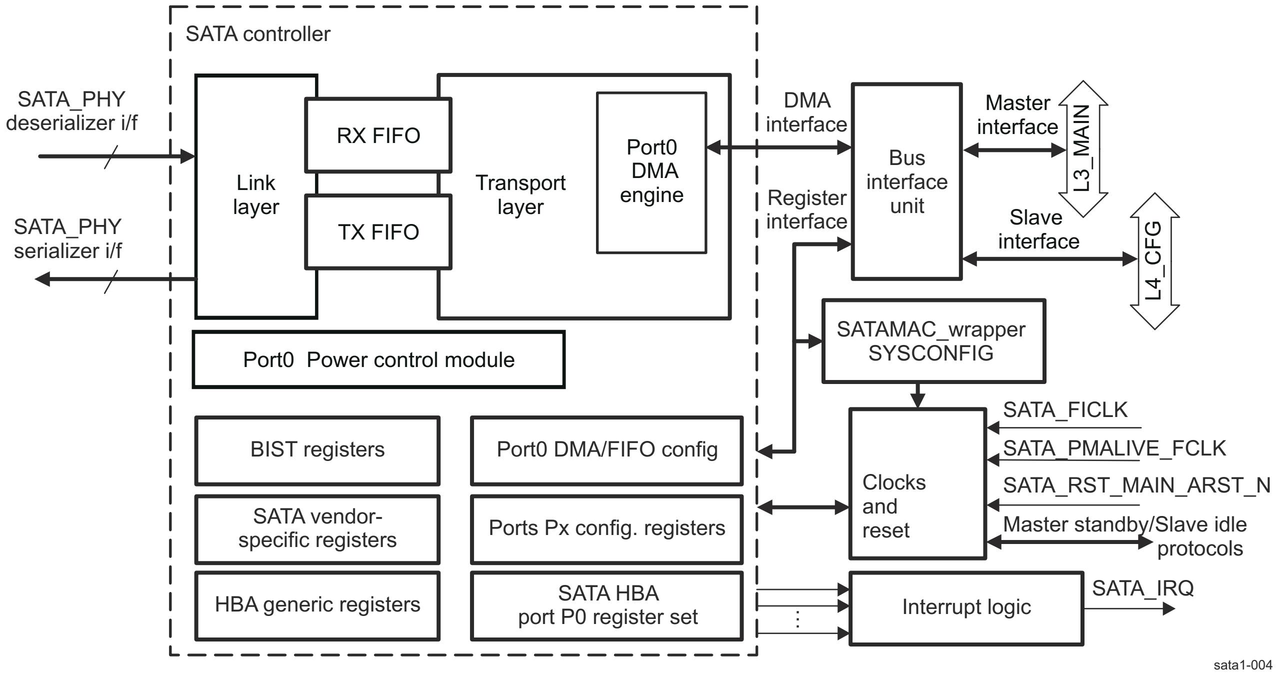


Figure 24-154. SATA Controller Functional Block Diagram

The SATA HBA port-associated DMA is a master on the device L3_MAIN interconnect. For a write to the attached storage device, this port reads FISs from the host system memory and pushes them into the transmit FIFO. When frame Dwords are received at RxFIFO, this interface is used to write the data out to the dedicated host system memory area. For details about DMA features and operation, see Section 24.8.4.8, *DMA Port Configuration*.

The SATA controller has a configuration slave interface used to read and write all system registers. The slave interface includes the following functions:

- Configuring the SATA AHCI core (and indirectly, to a certain extent, the SATA_PHY physical component)
- Configuring the DMA for transmit and receive operations
- Initiating write transfers to the attached storage device
- Responding to interrupts and reads

Both the slave and master interface of the SATA host controller share the same functional and interface clock, that is, operate on the same frequency (see also Section 24.8.3, *SATA Controller Integration*).

A bus interface unit adapts the SATA controller master and slave buses to the corresponding L3_MAIN and L4_CFG device interconnects.

Note

The **SATA controller AHCI Core registers** are part of the instance signified as **DWC_ahsata** inside the Section 24.8.6.1, *SATA Controller Instance Summary*.

24.8.4.2 SATA Controller Link Layer Protocol and Data Format

One of the main tasks of the SATA HBA Link layer is to perform a parallel 8-bit FIS data to a 10-bit parallel code conversion (8b/10b encoding) . The encoded stream, which is a sequence of 10-bit parallel-encoded characters, is subsequently passed to the SATA_PHY serializer component to adapt to a serial 10-bit character transmission to the SATA peripheral device. For more details, see Section 26.1.1, *SATA PHY Subsystem Overview*, and Section 26.1.4, *SATA PHY Subsystem Functional Description,* in Chapter 26, *Shared PHY Component Subsystems*.

One of the main tasks of the SATA HBA Link receiver is to perform a 8b/10b decoding, in which process the parallel 10-bit characters coming out from the SATA_PHY de-serializer are decoded to parallel data or control bytes. The received symbols with invalid codes are rejected by the link layer and corresponding illegal reception errors are generated. For more details regarding the SATA_PHY de-serializer, see Section 26.1.1, *SATA PHY Subsystem Overview*, and Section 26.1.4, *SATA PHY Subsystem Functional Description,* in Chapter 26, *Shared PHY Component Subsystems*.

There are also other data process stages at the link layer, such as CRC calculation, scrambling/descrambling, primitive insertion/detection, etc. Figure 24-155 shows the primary stages of link Dword data processing.

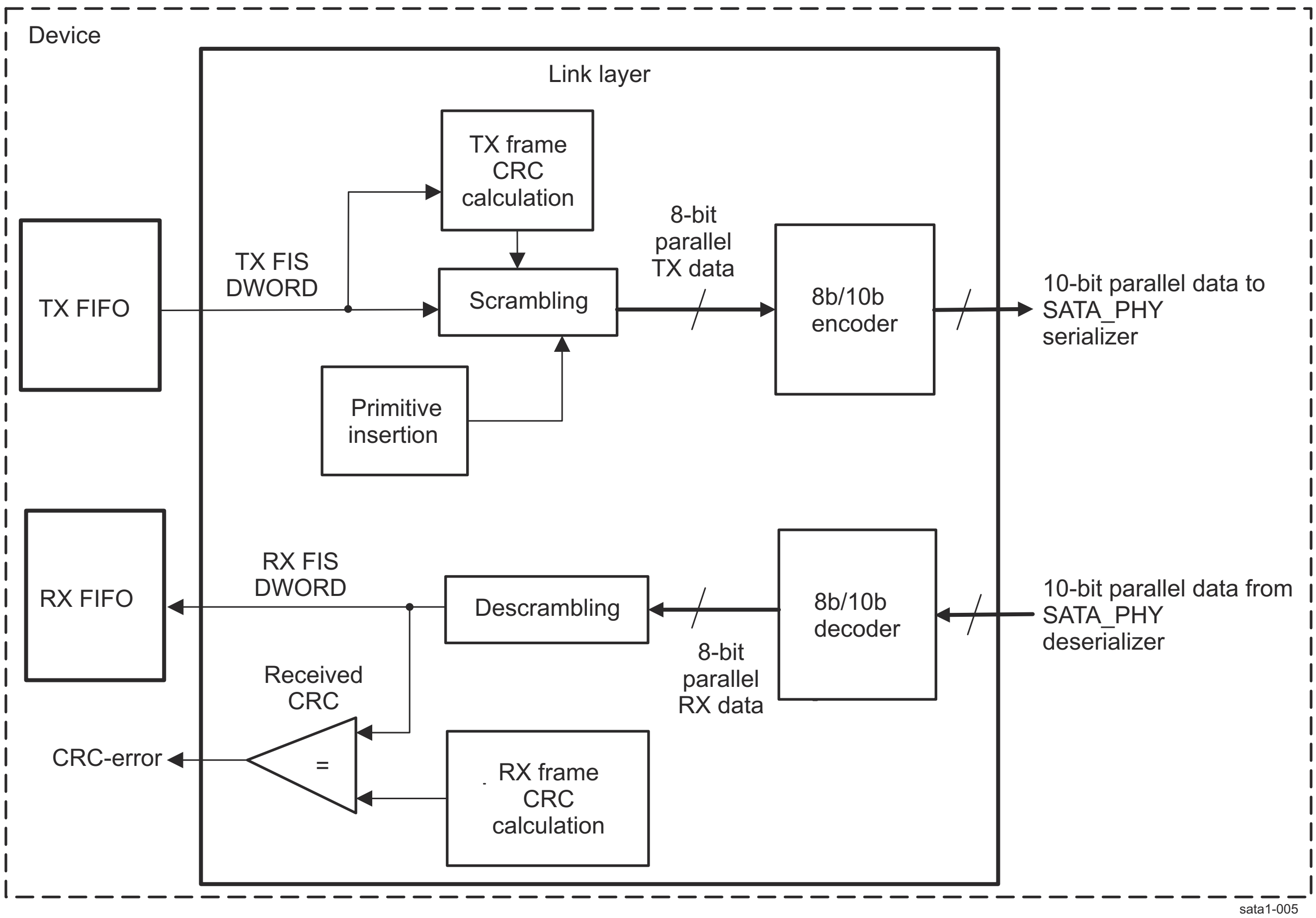


Figure 24-155. Simplified Schema of Link Dword Processing

24.8.4.2.1 SATA 8b/10b Parallel Encoding/Decoding

The input to the link coder logic is a parallel (unencoded) data byte and an associated control variable: Z with two possible values (K, D). The Z value signifies the byte to be encoded either as a control character (K value) or a data character (D value). The data byte (ABCDEFGH, where A matches the least significant bit [LSB]) is divided into two asymmetric portions: a 5-bit one ABCDE (range: 0 to 31) and a 3-bit one FGH (range: 0 to 7).

The encoding process is performed in two stages. At the first stage, the 5-bit portion is converted to a 6-bit subblock (abcdei); at the second stage, the 3-bit portion is converted to a 4-bit subblock (fghj). An additional parameter called running disparity (rd) is taken into account during code calculation at both the character transmission and reception link sides. It introduces code correlation between adjacent bytes so that the current subblock code can depend on the code of the previous subblock and its associated rd output value. As a consequence, depending on the input rd parameter and the 8b/10b encoded value, the encoding/decoding lookup tables might have two different entries for the same byte.

The 10-bit parallel-encoded character is passed to the 10-bit data input of the SATA_PHY serializer component, which further transmits it serially in a LSB to most significant bit (MSB) order (abcdeifghj). The receiver link receives and decodes the sequence in the same LSB to MSB order (abcdeifghj). While SATA encoder takes the full range of a data byte (0–255) encoding values, it uses only two values to encode control characters. For more information on the 8b/10b encoding/decoding process, see the SATA standard specification.

24.8.4.2.2 SATA Stream Dword Components

The basic unit of SATA information is Dword – a double sized word (32 bits). An encoded Dword has a 40-bit length (4 x 10 bits), but, for simplicity, unencoded Dword lengths are considered here. Each Dword is serially transmitted or received by the SATA_PHY components in LSB character to MSB character order.

- Primitives

  The shortest meaningful component of the SATA stream is the primitive, which consists of a single Dword. Different types of primitive characters are inserted into the SATA stream to maintain synchronization between the host and the SATA peripheral device. Primitives convey real-time state information regarding the SATA communication channel itself. For example, the SATA link layer generates primitives to perform flow control, synchronize power management state transitions, frame enveloping, etc. The transport layer is not aware of the lower level primitive Dwords. They are subject to processing only to the link layer.
- Primitive handshakes

  In some cases, the transmitter of a primitive can require that the receiver send an acknowledge Dword to confirm primitive reception, which is a primitive handshake.

  The first byte of a primitive Dword is always a control character (or Z = K) with two possible encodings. The remaining three data bytes (Z = D) contain the function of the primitive. According to the SATA standard, the transmitter and the receiver are not required to match the number of primitives sent and received.

  Table 24-443 briefly summarizes the primitives used by SATA.

Table 24-443. SATA Primitives Summary

Primitive Name	Short Description
ALIGNp	Sent in pairs to PHY component to readjust its internal operations
CONTp	After CONTp insertion, previous primitives are repeated continuously until a different primitive is inserted
DMATp	terminates DMA data transmission
EOFp	End of frame
HOLDp	Inserted to hold data when transmitter doesn't have data ready to transmit
HOLDAp	Acknowledges a HOLDp primitive
PMACKp	Acknowledges a power management request
PMNAKp	A power management request denial
PMREQ_Pp	Power management request to PARTIAL state
PMREQ_Sp	Power management request to SLUMBER state
R_ERRp	Reception error occurred
R_IPp	Current node is receiving data
R_OKp	Reception without error
R_RDYp	Receiver is ready for reception

Table 24-443. SATA Primitives Summary (continued)

Primitive Name	Short Description
SOFp	Start of frame
SYNCp	Synchronization primitive
WTRMp	Transmitter inserts this primitive while waiting to get the reception status from receiver
XRDYp	Transmitter has data ready for transmission

- SATA data frames

  SATA data frames incorporate data FISs constructed at the transport layer along with some primitives (see Figure 24-156). The maximum number of Dwords between SOFp to EOFp does not exceed 2064 Dwords, including the FIS type and CRC fields described here.
- Frame FIS components

  The FIS (Frame Information Structure) contents of a frame incorporate the Dwords that correspond to SATA target data structures. FIS Dwords are output from transport layer TxFIFO to the transmitter link 8b/10b encoder during transmission. In the case of reception, the FIS content is obtained at transport layer RxFIFO after being successfully decoded by the receiver link logic. The commands and data embedded into Frame FISs are generated/interpreted at a higher transport level of the SATA controller and are user software-accessible in the host system memory.
- CRC

  The CRC is the last Dword in a frame, immediately following the last FIS Dword of the frame and preceding the EOFp primitive. The 32-bit CRC calculation is performed only on the FIS Dwords of the frame contents, so that frame-inserted primitives such as HOLDp, CONTp, etc., are not taken into account. The CRC is aligned on a Dword boundary. CRC code computation expects an input of an even number of words and should be Dword-multiple. If an FIS block contains an odd number of words, the last FIS word is padded with 0s to a full Dword before the CRC is calculated.
  – During transmission: The CRC Dword is calculated on original FIS contents, before scrambling and serial 8b/10b encoding are performed.
  – During reception: The CRC Dword is calculated on the 8b/10b decoded and subsequently descrambled 8-bit FIS Dword contents. It is then compared to the CRC received from the source to perform error check analysis.
- Frame embedded primitives

  The beginning of a frame is marked with a SOFp primitive, and the end is marked with an EOFp primitive. Some primitives, such as HOLDp and HOLDAp acknowledge, can be inserted into frames between data FISs.
- SATA data stream

  As shown in Figure 24-156, the SATA data stream is composed of frames carrying the target data/command contents combined with various primitives generated to maintain consistency of transfers over the SATA interface.

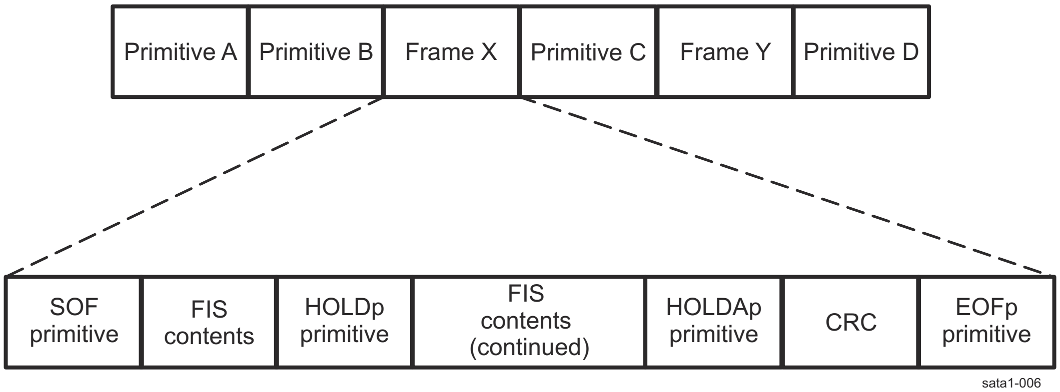


Figure 24-156. SATA Data Stream Components

24.8.4.2.3 Scrambling/Descrambling Processing

The contents of a frame are scrambled before being passed to the link 8b/10b encoder. Scrambling is performed on Dword quantities by XORing the data to be transmitted with the output of a linear feedback shift register (LFSR). All data words between the SOFp and EOFp are scrambled, including the CRC. Figure 24-155 shows the basic flow.

The SATA controller (as defined by the SATA standard) performs scrambling/descrambling processing for EMI reduction purposes. Two scramblers are instantiated within the SATA controller: The first, the data payload scrambler, scrambles FIS data payload contents (including the calculated CRC Dword); the second, the repeated primitive suppression scrambler, scrambles only the repeated primitive stream contents.

The scrambling runs as follows: A certain data payload Dword located between SOFp and EOFp is XORed with the data payload scrambler output. The resulting scrambled Dword is submitted to the 8b/10b encoder for transmission.

On reception, the Dword is decoded using a 10b/8b decoder, the scrambler output is XORed with the resulting Dword, and the resulting Dword is presented to the link layer. It is then used for CRC computation. The CRC Dword is scrambled in the same way as data payload characters.

For details on the scrambling/descrambling feature, see the *SATA Standard Specification* (rev.2.6).

24.8.4.3 Resets

24.8.4.3.1 Hardware Reset

The module is reinitialized in hardware upon activation of PRCM.L3INIT_RST reset signal (for more information about hardware reset, see Section 24.8.3, *SATA Controller Integration*).

For more information on the PRCM.L3INIT_RST hardware reset signal, refer to *Reset Domains*, in *Power, Reset, and Clock Management*.

This reset normally occurs on device power up or during a system bus failure. All components of the SATA AHCI core are initialized, including the HBA port and generic registers. The bus interface unit (BIU) and SATA AHCI core wrapper components are also impacted.

24.8.4.3.2 Software Initiated Resets

The SATA controller supports three levels of software reset, each of which has different impact over the SATA HBA hardware components and AHCI interface registers:

- Software reset (least intrusive)
- Port reset
- HBA reset (most intrusive)

Note

It is recommended that user software always starts with the least intrusive reset approach. Software chooses the next most intrusive reset only if the less intrusive reset does not succeed in clearing the condition.

24.8.4.3.2.1 Software Reset

Software reset impacts only the attached to the HBA port SATA peripheral device, without affecting the established communication between physical layers or HBA engine.

To issue a software reset, the user must prepare two H2D register FISs into the emptied command list of the port. The first FIS must have bits SRST = 0b1 and C = 0b0. The first FIS corresponding command header bits C and R are set as follows: C = 0b1 and R = 0b1.The second FIS has bits SRST = 0b0 and C = 0b0. The second FIS corresponding command header bits C and R are set as follows: C = 0b0 and R = 0b0.

However, steps must be taken before issuing the software reset to the peripheral device attached to the port. For details on software reset, see the AHCI specification.

24.8.4.3.2.2 Port Reset

The port reset (also known as COMRESET) must be applied when the HBA port does not function properly after a software reset is issued to the port. The effect is that the SATA HBA port is reinitialized and communication between the phy layers of both the SATA host and the SATA target device is reestablished.

To trigger a COMRESET sequence on the SATA interface, user software must write 0x1 to the SATA_PxSCTL[3:0]DET bit field. The host receives a COMINIT signal sequence, indicating that the peripheral device has successfully reestablished communication. For more information on port reset, see the AHCI specification (revision 1.1).

Note

The port reset (COMRESET) method is the preferred AHCI mechanism for error recovery and should be used instead of the software reset.

24.8.4.3.2.3 HBA Reset

Software can globally reset the SATA controller, if a port reset does not clear the conditions successfully. On HBA reset, all SATA AHCI core interface registers are reset, **excluding those from the hardware initialization domain** - the SATA_PxCLB, SATA_PxCLBU, SATA_PxFB and SATA_PxFBU registers. All state machines that relate to data transfers and queuing return to IDLE state, and the port is reinitialized by sending COMRESET. The global reset is triggered when user software sets the SATA_GHC[0]HR = 0b1 bit. The HBA indicates that global reset is finished, deasserting the SATA_GHC[0]HR bit to 0b0 in hardware. Hence, user software must poll the HR bit to detect this condition. For more information, see the mentioned AHCI standard specification.

24.8.4.4 Power Management

The device SATA host controller subsystem power management can be identified at the following levels:

- SATA controller low power-management features (defined also in the SATA standard specification, revision 2.6)
- Master standby and idle slave protocols established between SATA controller and the device PRCM through a SATA AHCI core wrapper interface (signified as SATAMAC_wrapper instance inside the Section 24.8.6.1, *SATA Controller Instance Summary*).

This section describes the relationships that exist between both levels of SATA Controller power management.

24.8.4.4.1 SATA Specific Power Management

The SATA controller core supports both PARTIAL and SLUMBER low-power states, as described in the SATA standard specification (revision 2.6). These modes allow power savings by powering down part of SATA_PHY and gating off the clocks to the link layer. The port power control module is used to enter and exit these modes, which can have the normal functional clocks gated off.

24.8.4.4.1.1 PARTIAL Power Mode

In the PARTIAL power state, the SATA_PHY component transmits electrical IDLE to the device and the receiver is left on to receive OOB signals. This mode can also be used to support near-end analog loopback, if a device is attached.

24.8.4.4.1.2 Slumber Power Mode

In the SLUMBER power state, the transmitter is completely turned off. This allows for greater power savings but near-end analog loopback cannot be performed and there is a longer latency to exit the mode.

24.8.4.4.1.3 Software Control over Low Power States

The user has manual control over low power states through the SATA_PxCMD [31:28]ICC register bit field. This works if the link is in IDLE state; otherwise, writes to this field are ignored.

24.8.4.4.1.4 Aggressive Power Management

Low power modes can be initiated by software when there are no pending transactions, but the SATA AHCI core also supports aggressive power management. In this mode, the subsystem automatically initiates the PARTIAL or SLUMBER state when there are no inflight commands. This is useful for power savings, but the price is increased latency if a low-power mode is entered and a transfer request is made. Software enables aggressive link-layer power management by setting the SATA_PxCMD[26]ALPE bit to 0x1. The link aggressive power management state (PARTIAL or SLUMBER) is selected using the SATA_PxCMD[27]ASP bit.

24.8.4.4.2 Master Standby and Slave Idle Management Protocols

The master standby and slave idle protocols are implemented between the PRCM and SATA controller, with mode settings located in the SATAMAC_wrapper register (SATA_SYSCONFIG).

- The standard idle request/acknowledge handshake is associated with the L4_CFG slave port. The default idle mode is smart-idle, as indicated by the power-on reset (POR) value: SATA_SYSCONFIG[3:2]IDLEMODE = 0x2.

  Note

  No other mode is expected to be required in normal operation: no-idle and force-idle are used for debugging only. Because the module is not capable of asynchronous wakeup, the smart-idle/wakeup (IDLEMODE = 0x3) is strictly equivalent to smart-idle.

- The standby/wait handshake is associated with the L3_MAIN master port. Software is responsible to keep the DMA operations and the standby status in sync.

The default standby mode is smart-standby, as indicated by the POR value: SATA_SYSCONFIG[5:4]STANDBYMODE = 0x2. However, the SATA controller hardware does not provide any indication about DMA activity that could be used to drive the standby dynamically (that is, the smart mode): Smart-standby is therefore strictly equivalent to the force-standby mode (that is, the module remains in or goes to permanent standby).

The standby control procedures are therefore software-driven.

To exit standby:

1. Change the standby mode to no-standby by setting SATA_SYSCONFIG[5:4] = 0x1.
2. Optional: Confirm the SATA controller standby status through the PRCM. For more information, see *Clock Domain Module Attributes*, in *Power, Reset, and Clock Management*.
3. Only then, enable the controller DMA (AHCI list processor). For more information, see the AHCI specification.

To enter standby:

1. Disable the controller DMA (AHCI list processor). For more information, see the AHCI specification.
2. Ensure all pending transactions have completed. For more information, see the AHCI specification.
3. Change the standby mode to smart-standby (or force-standby) by setting SATA_SYSCONFIG[5:4] = 0x2.

CAUTION

The software is responsible to keep the DMA operations and the standby mode status in sync.

Note

Because the standby exit is software driven, it is by definition synchronous: the asynchronous wakeup is never activated, which means that the smart-standby and smart-standby/wakeup are equivalent (as well as force-standby).

Note

For more information on the SATA Controller master standby and slave idle management protocols with the device PRCM , see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

24.8.4.4.3 Clock Gating Synchronization

The bus clock can also be gated off in either the PARTIAL or SLUMBER state, if the software guarantees there is nothing in flight, the clock can be stopped and started without glitches, and the clock can be restarted quickly when activity is initiated from the device side.

The following steps outline the procedure to stop the clock:

1. Put the link in a low power state (either PARTIAL or SLUMBER)
   - SATA MAC link layer can enter low power either manually see Section 24.8.4.4.1.3, *Software Control over Low Power States*) or automatically (see Section 24.8.4.4.1.4, *Aggressive Power Management*)
   - If there is no device attached (making it impossible to enter a low power state), the SATA_SYSCONFIG[16] OVERRIDE0 override bit can be set to 0b1 to stop the interface/functional clock (SATA_FICLK) and save power.
2. Initiate the clock stopping procedure.

CAUTION

If a SATA drive attached to the SATA HBA port has not entered a low power state (PARTIAL or SLUMBER), the user must not set the SATA_SYSCONFIG[16]OVERRIDE0 bit; otherwise, the link layer can be ruined, resulting in indeterminate SATA controller behavior.

24.8.4.4.4

Table 24-444. Local Power-Management Features

Feature	Registers	Description
Slave idle modes	SATA_SYSCONFIG [3:2]IDLEMODE bit field	The available modes are: Force-idle, no-idle, and smart-idle wakeup disabled modes are available.
Master standby modes	SATA_SYSCONFIG [5:4]STANDBYMODE bit field	The available modes are: Force-standby, no-standby, smart-standby wakeup disabled
Clock-gate overriding mode	SATA_SYSCONFIG [16]OVERRIDE0 bit field	Clock stopping override function. Should not be applied when attached to port SATA peripheral device has not entered a Low-power mode.

For more information on the L3INIT_L3_GICLK functional clock gating in the device PRCM , refer to *Clock Domain-Level Clock Management*, in the chapter, *Power, Reset and Clock Management*.

24.8.4.5 Interrupt Requests

This section describes all of the conditions that can generate an interrupt. Each condition is subject to the masking capabilities described in Section 24.8.4.5.4, *Interrupt Condition Control*.

24.8.4.5.1 Interrupt Generation

Both level and pulse interrupts are available. Exercise care when using the pulse interrupt. Because of the synchronization and interrupt aggregation from the internal core, clearing one interrupt as another interrupt occurs can prevent a new pulse from being generated. A new pulse is generated only if the first interrupt is completely cleared (that is, the level interrupt goes away) before the new interrupt event occurs. Thus, an interrupt can occur after the register write is issued to clear the first interrupt but before the interrupt is fully cleared. In this case, a new pulse is not issued and the level interrupt remains asserted. After issuing the write to clear an interrupt, software must read the interrupt status registers to ensure that a new interrupt has not occurred.

24.8.4.5.2 Levels of Interrupt Control

The AHCI directs that interrupts be indicated in a two-tier structure and thus associated with different HBA ports. Considering the device SATA HBA single-port implementation for the first tier, only the SATA_IS[0] IPS bit should be considered. It indicates if the only port available to the user (HBA port 0) has pending interrupts.

For the second tier, the SATA_PxIS register (where x = 0) indicates which specific interrupt condition(s) occurs to trigger an interrupt on port 0. In most cases, software writes 0x1 (Write One to Clear) to clear these bits, and then writes 0x1 to the SATA_IS [0]IPS bit to clear the interrupt. However, some interrupts in the SATA_PxIS register have alternate methods of being cleared (for details, see Section 24.8.6.2.2, *DWC_ahsata Register Description*). Section 24.8.4.5.3, *Interrupt Events Description*, describes all interrupt generating conditions.

Figure 24-157 shows the SATA controller two-tier interrupt propagation structure. Besides SATA_IS[0], which gates all interrupts at the port 0 level, the tier 1 interrupt is globally gated by the SATA_GHC[1] IE bit.

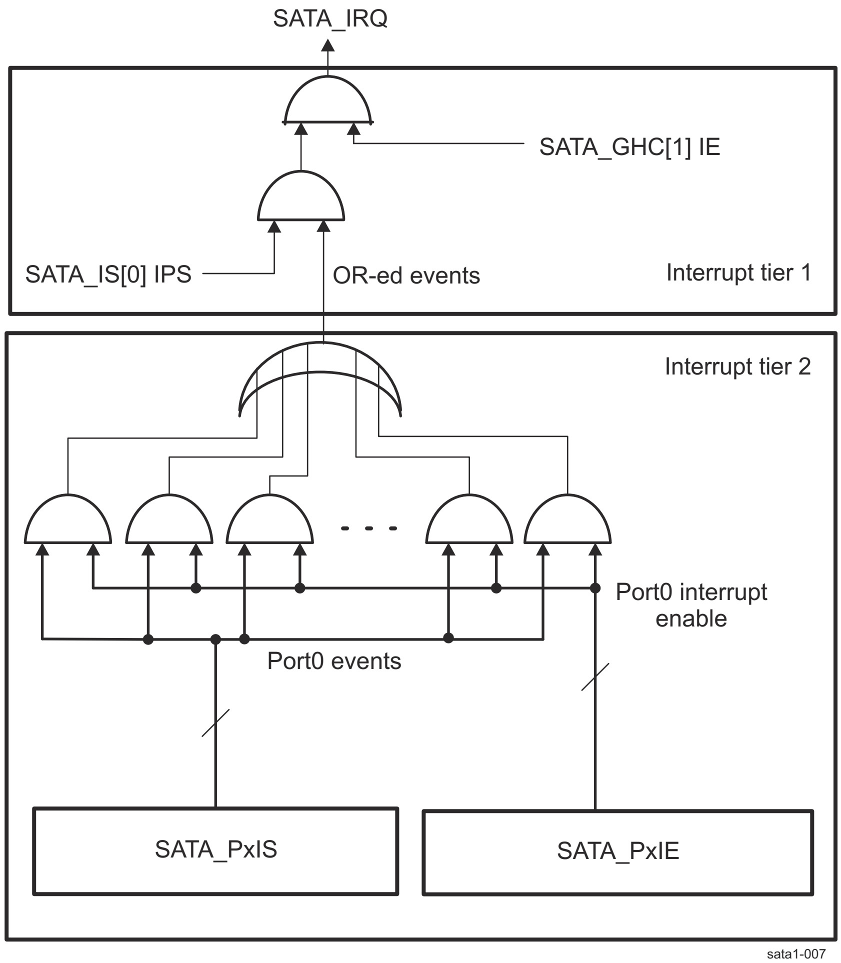


Figure 24-157. SATA Controller Interrupt Propagation Schema

Note

Regardless of the device SATA controller single HBA port implementation, user software must read/write the tier 1 SATA_IS[0] IPS register bit to process interrupts.

24.8.4.5.3 Interrupt Events Description

This section describes the different events that are reflected in the SATA_PxIS register and that cause SATA_IRQ interrupt generation, if enabled, in the SATA_PxIE register. For more information regarding SATA_IRQ handling at the IRQ CROSSBAR subsystem, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

24.8.4.5.3.1 Task File Error Status

This event is registered whenever the SATA_PxTFD[7:0] STS bit field is updated by the device and the error bit SATA_PxTFD[0]STS_ERR is set. SATA_PxTFD[7:0] STS is updated when a new register FIS, PIO setup FIS, or set device bits FIS is received from the device.

24.8.4.5.3.2 Host Bus Fatal Error

This event is registered whenever an error is received by the SATA controller internal DMA master port from the configuration slave port.

24.8.4.5.3.3 Interface Fatal Error Status

This event is registered if an error is generated on the interface. The following conditions can cause this:

- SYNC escape during an H2D register or data FIS transmission
- One of the following errors during a data FIS transfer:
  - Protocol (SATA_PSERR[10] ERR_P)
  - CRC (SATA_PSERR[21] DIAG_C)
  - Handshake (SATA_PSERR[22] DIAG_H)
  - SATA_PHY not ready (SATA_PSERR[9] ERR_C)
- Unknown FIS received with a good CRC, but length greater than 64 bytes
- Physical region descriptor (PRD) table byte count of 0

Note

This is a fatal error in which the DMA is in an error state until software clears the SATA_PxCMD[0] ST bit or resets the interface with a port or global reset. For more information about software-initiated resets, see Section 24.8.4.3.2, *Software Initiated Resets*.

24.8.4.5.3.4 Interface Non-Fatal Error Status

This event is registered on the following error conditions:

- One or more of the following errors are registered during a non-data FIS transfer:
  - Protocol (SATA_PSERR[10] ERR_P)
  - CRC (SATA_PSERR[21] DIAG_C)
  - Handshake (SATA_PSERR[22] DIAG_H)
  - SATA_PHY not ready (SATA_PSERR[9] ERR_C)
- Command list underflow during a read operation when software builds a command table that has more total bytes than the transaction given to the device

---

Note

The conditions previously described are non-fatal and the SATA host controller continues to operate normally.

24.8.4.5.3.5 Overflow Status

This event is registered if command list overflow is detected during read or write when software builds a command table with fewer total bytes than the transaction given to the device.

Note

This is a fatal error in which the DMA is in an error state until software clears the SATA_PxCMD[0] ST bit or resets the interface with a port or global reset. For more information about software-initiated resets, see Section 24.8.4.3.2, *Software Initiated Resets*.

24.8.4.5.3.6 Incorrect Port Multiplier Status

This event is registered when the SATA controller receives a FIS from a device with a PM field not matching what is expected. The SATA_PxIS[23] IPMS event corresponding bit can be set during device enumeration, but software should ignore it until enumeration is finished.

24.8.4.5.3.7 PHYReady Change Status

This event is registered when the status of PHYReady changes. Its corresponding status bit (SATA_PxIS[22] PRCS) reflects the SATA_PxSERR[16]DIAG_N bit.

24.8.4.5.3.8 Port Connect Change Status

This event is registered when there was a change in the current connect status as reflected by the SATA_PxSERR[26]DIAG_X bit.

24.8.4.5.3.9 Descriptor Processed

This event is registered when a transaction completes. Its corresponding flag (SATA_PxIS[5] DPS) is set whenever a PRD with the I bit set transfers all of its data.

Note

The PRD interrupt is an opportunistic interrupt and should not be used to definitely indicate the end of a transfer. Two PRD interrupts can happen close in time so that the second interrupt is missed while the first PRD interrupt is being cleared.

When a PRD entry is exhausted, the HBA can be directed to generate an interrupt through the I bit in the PRD entry (although a PRD is not considered exhausted until all data FISs that transfer data pointed to by that PRD entry is complete). For example, if the data FIS is 8 KiB, and this is covered by three PRD entries, the data is not considered valid at the end of the first or second PRD because the CRC has not been checked, even though the data has been copied to memory or to the device. Therefore, if the I bit is set in the PRD entry, the HBA must hold onto it internally and not set PxIS.DPS until the data FIS is complete and the CRC is correct. Once correct, PxIS.DPS can be set and, if PxIE.DPE and GHC.IE are set, the HBA generates an interrupt.

Conversely, if the PRD entry is 16 KiB and two 8 KiB Data FISS are used to transfer all of the data pointed to by the PRD entry, the PRD interrupt associated with that PRD entry is not signaled until after the second data FIS transfer completes successfully.

24.8.4.5.3.10 Unknown FIS Interrupt

This status event indicates that an unknown FIS is received and copied into system memory.

24.8.4.5.3.11 Set Device Bits Interrupt

This status event indicates that a Set Device Bits FIS is received with the I bit set and copied to system memory.

24.8.4.5.3.12 DMA Setup FIS Interrupt

This status event indicates that a DMA Setup FIS is received with the I bit set and copied into system memory.

24.8.4.5.3.13 PIO Setup FIS Interrupt

This status event indicates that a PIO Setup FIS with the I bit set is copied into system memory and the data related to that FIS is transferred.

24.8.4.5.3.14 Device to Host Register FIS Interrupt

This status event indicates that a D2H Register FIS is received with the I bit set and copied to system memory.

24.8.4.5.4 Interrupt Condition Control

There are several ways for software to control the conditions causing a non - command completion coalescing **(non-CCC)** SATA_IRQ interrupt assertion. The SATA_GHC[1] IE register bit serves as a global mask. If it is 0x0, all interrupts are masked. If it is 0x1, interrupts not masked by the SATA_IS (all interrupts for port 0) and SATA_PxIE registers can trigger an interrupt. The SATA_PxIE register matches the SATA_PxIS register bit for bit. If the corresponding bit in the SATA_PxIE register is set to 0x1, the interrupt is unmasked. If the corresponding is 0x0, the interrupt is masked off.

Table 24-445 summarizes the SATA controller interrupt status bits and their corresponding interrupt enable bits at the second tier of the SATA controller interrupt generation schema.

Table 24-445. Interrupt Events Summary

Event Status Flag	Interrupt Event Mask	Mapped to Interrupt Output	Brief Interrupt Event Description
SATA_PxIS [30]TFES	SATA_PxIE [30]TFEE	SATA_IRQ	Task File Error
SATA_PxIS [29]HBFS	SATA_PxIE [29]HBFE	SATA_IRQ	Host Bus Fatal Error
SATA_PxIS [27]IFS	SATA_PxIE [27]IFE	SATA_IRQ	Interface Fatal Error
SATA_PxIS [26]INFS	SATA_PxIE [26]INFE	SATA_IRQ	Interface Non-Fatal Error
SATA_PxIS [24]OFS	SATA_PxIE [24]OFE	SATA_IRQ	Overflow
SATA_PxIS [23]IPMS	SATA_PxIE [23]IPME	SATA_IRQ	Incorrect Port Multiplier
SATA_PxIS [22]PRCS	SATA_PxIE [22]PRCE	SATA_IRQ	Phy Ready Change
SATA_PxIS [6]PCS	SATA_PxIE [6]PCE	SATA_IRQ	Port Connect Change
SATA_PxIS [5]DPS	SATA_PxIE [5]DPE	SATA_IRQ	Descriptor Processed
SATA_PxIS [4]UFS	SATA_PxIE [4]UFE	SATA_IRQ	Unknown FIS Received
SATA_PxIS [3]SDBS	SATA_PxIE [3]SDBE	SATA_IRQ	Set Device Bits FIS Received
SATA_PxIS [2]DSS	SATA_PxIE [2]DSE	SATA_IRQ	DMA Setup FIS Received
SATA_PxIS [1]PSS	SATA_PxIE [1]PSE	SATA_IRQ	PIO Setup FIS Received
SATA_PxIS[0]DHRS	SATA_PxIE [0]DHRE	SATA_IRQ	Device to Host Register FIS Received

Note

To ensure normal generation of the standard (single-event driven, non-CCC) interrupts described in Table 24-445, the user must ensure that the CCC feature is disabled (that is, the SATA_CCC_CTL[0] EN bit is set to 0x0).

24.8.4.5.5 Command Completion Coalescing Interrupts

The SATA controller supports CCC, which can be thought of as interrupt pacing. Instead of being generated for every command completed, an interrupt can be triggered based on a certain number of commands completed and/or a timer time-out. This significantly reduces the load on the CPU by not allowing software to receive an interrupt every time a command is processed. Software can thus queue many commands or wait for many received FISes and process them as a batch. All CCC functions are controlled by the SATA_CCC_CTL, SATA_CCC_PORTS, and SATA_TIMER1MS registers. Setting SATA_CCC_PORTS[0]PRT = 0x1 makes the only available SATA controller HBA port 0 - CCC-aware. The SATA_CCC_CTL register can be used to program CCC logic to trigger an interrupt whenever a certain number of commands are complete and/or a timeout value (as specified by the TV field of SATA_CCC_CTL and SATA_TIMER1MS registers) occurs. For specific programming instructions, see the SATA_CCC_CTL description.

24.8.4.5.5.1 CCC Interrupt Based on Expired Timeout Value

To trigger a CCC interrupt on a port after an elapsed timer value, the following programming sequence must be implemented by user software:

1. Software disables the CCC feature, clearing SATA_CCC_CTL[0] EN to 0b0.
2. Software sets SATA_CCC_PORTS[0] PRT = 0x1 to make HBA port 0 CCC-aware.
3. Software sets SATA_CCC_CTL[15:8] CC = 0x0, which provides that the expired timeout trigger condition is selected.
4. Software defines the 1-ms time-based unit by loading an appropriate number of interface/functional clock cycles into the 20-bit SATA_TIMER1MS[19:0] TIMV bit field. For example, its reset value of 0x186A0 (100 000 x SATA_FICLK cycles) provides that a 1-ms timeout base will be generated if SATA_FICLK = 100 MHz.
5. Software chooses the timeout value as the number of 1-ms intervals through the 16-bit SATA_CCC_CTL[31:16]TV bit field.
6. Software enables the CCC-interrupt feature, asserting SATA_CCC_CTL[0] EN bit to 0b1.

Assertion of bit SATA_CCC_CTL[0] EN to 0x1 is required to start the 1MS TIMER after the above described configuration steps are completed.

Note

On CCC event configuration, the CCC feature should be disabled (that is, SATA_CCC_CTL[0] EN should be kept at 0b0).

Note

Due to the single SATA controller HBA port integration, the source of the CCC interrupt, if enabled, is always HBA port 0. On a CCC interrupt, the SATA_CCC_CTL[7:3] INT read-only bit changes to 0x0 (its reset value is 0x1) . As a consequence, the STA_IS[0] IPS bit is set to 0x1.

24.8.4.5.5.2 CCC Interrupt Based on Completion Count

If the desired method to receive an interrupt is based on the completion of a certain number of commands, the following settings must be performed:

1. Software disables the CCC feature, clearing SATA_CCC_CTL[0] EN to 0b0.
2. Software sets SATA_CCC_PORTS[0] PRT = 0x1 to make HBA port 0 CCC-aware.
3. Software sets SATA_CCC_CTL[15:8] CC! = 0x0 to the desired number (1–255) of commands to be completed. The CC! = 0x0 provides that the count completion condition is selected.
4. Software enables the CCC-interrupt feature, asserting the SATA_CCC_CTL[0] EN bit to 0b1.

Note

On CCC event configuration , the CCC feature should be disabled (that is, SATA_CCC_CTL[0] EN should be kept at 0b0).

Note

Due to the single SATA controller HBA port integration, the source of the CCC interrupt, if enabled, is always HBA port 0. On a CCC interrupt, the SATA_CCC_CTL [7:3]INT read-only bit changes to 0x0 (its reset value is 0x1) . As a consequence, the STA_IS[0] IPS bit is set to 0x1.

24.8.4.6 System Memory FIS Descriptors

The SATA AHCI mode supports a command list of from 1 up to 32 entries (command slots) assigned to HBA port 0, which should be allocated and built into host system memory by user software. After receiving command/data messages from the attached port 0 SATA mass-storage drive, the HBA extracts the FIS Dwords, sorts them out, and stores them into different areas of RX FIS-allocated host system memory.

24.8.4.6.1 Command List Structure Basics

Each command list entry has an associated command header (CH) that takes 32 bytes in host system memory, from which only the first 4 Dwords (16 bytes) are used. The CH contains different fields that detail the direction of transfer (H2D or D2H), type of command (ATA/ATAPI), reset behavior, PM ID, byte count and byte length of the PRD table, etc.

A command list slot CH specifies the base address of the underlying command table and information about the associated command FIS length. The command table in system memory has three main areas:

- A 64-byte command FIS area in which the user loads the command-associated FIS contents
- An ATAPI area in which a 12- or 16-byte ATAPI command is loaded, if specified in the CH (through bit A)
- A PRD table area that allows up to 65535 data descriptor entries

An entry of a PRD table (that is, a scatter/gather list item) specifies descriptor parameters of a data block transfer handled by the HBA port DMA.

Figure 24-158 shows the interrelations between the FIS-related structures that user software must allocate and build into the host system memory.

For details on command and data structures in host system memory, see the AHCI and SATA standard specifications.

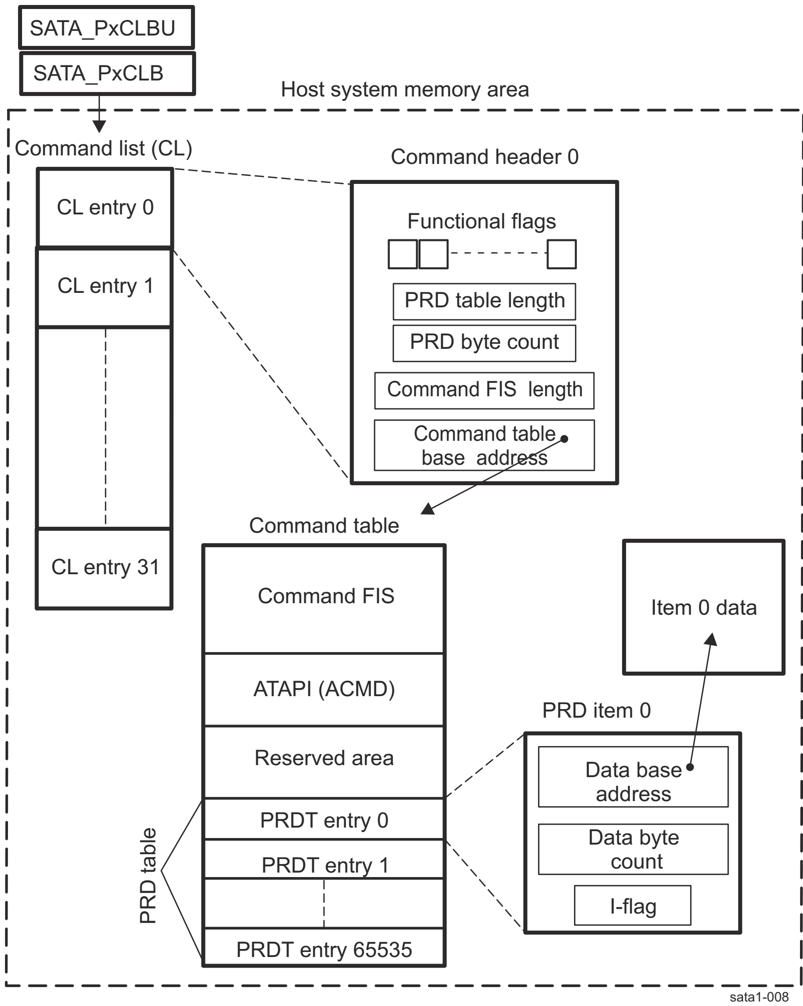


Figure 24-158. Command List Descriptor Structures

After allocating the necessary portions for the CL, command slots underlying data descriptors, and received FISs into host system memory, the base addresses of these memory locations must be provided by user software to the HBA through the hardware init domain registers: SATA_PxCLB (command list base address - lower bits), SATA_PxCLBU (command list base address-upper bits), SATA_PxFB (received FISs system memory area base address-lower bits) and SATA_PxFBU (received FISs system memory area base address-upper bits) . Thus, the command and data DMA engines can access the transfer context programming structures.

Note

SATA_PxCLBU / SATA_PxFBU registers store the upper half of the native 64-bit AHCI addresses. Because only 36-lower bits of the AHCI 64-bit address bus are integrated in the device, actually only the 4-lower bits within SATA_PxCLBU / SATA_PxFBU are meaningful. **The remaining 28-bits must be always written to '0'-s in the descriptors (SATA_PxCLBU[31:4]=0x0 / SATA_PxFBU[31:4]=0x0) .**
Default value for the SATA_PxCLBU / SATA_PxFBU registers is 0x0, so that a 32-bit SW driver which never accesses them works seamlessly, by accessing only the 32-bit (lower half) addresses which reside in the SATA_PxCLB / SATA_PxFB registers.

24.8.4.6.2 Supported Types of Commands

The following types commands are supported:

- Standard serial ATA nonqueued commands set.

  As opposed to NCQ, these commands are serviced, acknowledged one by one after a successful completion. The nonqueued SATA CL entries can take advantage of the PRD prefetching option indicated by the P flag in the corresponding CH (for more information, see the SATA specification). The ATA legacy commands are not supported by the SATA AHCI core.
- ATAPI commands

  ATAPI commands are supported by the SATA HBA engine. A request for an ATAPI command service is indicated by setting CH bit A to 0b1. The ATAPI command that the user builds into the ATAPI area of the command table can be 12 or 16 bytes in length and is indicated by the PIO Setup FIS sent by the SATA peripheral storage device requesting the ATAPI command. In addition, the SATA controller supports ATAPI active-LED generation (See Section 24.8.4.10, *Activity LED Generation Functionality*).
- NCQ-commands

  NCQ-commands are supported by the SATA controller HBA engine, the execution of which can be greatly facilitated by the CCC interrupt feature (for more information, see Section 24.8.4.5.5, *Command Completion Coalescing Interrupts*). As described in Section 24.8.1.1.2, *Native Command Queuing*, the NCQ commands have FPDMA queued command protocol mechanism, which implies that they are treated in a different way then standard SATA commands. The AHCI interface register set include specific registers, such as SATA_PxSACT, etc., to make the HBA aware of the outstanding NCQ slots in a command list. As opposed to standard nonqueued commands, NCQ commands are acknowledged before their actual completion, which accelerates the CL servicing for NCQ slots. The queue tag of the command requested by the SATA peripheral device is indicated to the HBA port by the TAG field in a DMA Setup FIS that an attached SATA peripheral device sends to the SATA controller host . The HBA then uses the tag value to index the NCQ commands updated by user software in the command list. The NCQ command descriptors are not allowed to use the prefetching option in CH because their order of execution is determined by the SATA target device.

24.8.4.6.3 Received FIS Structures

After stream decoding and primitive removal by the link layer, the validated FIS Dwords are extracted from the RxFIFO buffer by the transport layer DMA, and the original SATA storage drive FIS structures are recovered to a dedicated host system memory space. The DMA stores the received FISs into different memory areas based on FIS type. The unknown FIS reception is legal to HBA, as long as the length of the unknown FIS does not overrun the 64-byte size limit of the FIS descriptors. For details on received FISs memory area organization, see the AHCI and SATA standard specifications.

24.8.4.6.4 FIS Descriptors Summary

The FISs that encapsulate SATA target commands and data contents can be divided into:

- FISs transmitted from the host to an attached SATA peripheral device (H2D FISs)
- FISs sent by an attached SATA peripheral device to the host (D2H FISs)

The H2D FISs are constructed by user software into system memory. he H2D FIS types are:

- H2D Register FIS
- H2D DMA Setup FIS
- H2D Data FIS

The D2H FISs are constructed by the attached SATA peripheral device transport layer. The D2H FIS types are as follows:

- D2H Register FIS
- D2H DMA Setup FIS
- PIO Setup FIS
- DMA Activate FIS
- Set Device Bits FIS
- D2H Data FIS

For details on different type of FIS contents, see the SATA standard specification (revision 2.6).

24.8.4.7 Transport Layer FIS-Based Interactions

24.8.4.7.1 Software Processing of the Port Command List

After user software instantiates the CL structures in host system memory, it must maintain them appropriately. The command list is advanced when the BSY, DRQ, and ERR bits of a serial ATA device task file are cleared, which is communicated through a D2H register FIS to HBA port. The HBA port 0 hardware clears the corresponding to the i-th completed command: SATA_PxCI [i] CI (SATA_PxCI [TAG] CI for NCQ) bit to notify user software that the next command FIS can be advanced into the CL. The software then builds the command slot components: CH fields, command table, and associated PRD data descriptors in system memory. Software asserts the SATA_PxCI[i+1] CI (or SATA_PxCI[TAG] CI bit for an NCQ indicated command) to activate the prepared CL command, which is possible only when SATA_PxCMD[0] ST = 0x1. The nonqueued commands in the CL are processed in ascending order of slots (linear processing). The SATA_PxCMD [12:8]CCS read-only field indicates the CL index of the command currently being issued by the HBA port.

The processing of NCQ command slots invokes additional register update/check steps. Before setting the SATA_PxCI[TAG] CI bit to 0b1 and posting an active NCQ command to the SATA HBA port, software must indicate an NCQ outstanding command at the position defined by the SATA peripheral device queue TAG by setting SATA_PxSACT[TAG] DS = 0x1. After the queued command is posted, the peripheral SATA device sends a Set Device Bits FIS to the HBA port to clear the SATA_PxSACT bits corresponding to the successfully completed NCQ commands.

Command-list processing is initially triggered by software setting SATA_PxCMD[0] ST = 0x1, which requires certain conditions to be satisfied.

Note

There are some restrictions to setting SATA_PxCMD[0] ST to 0x1. For more information, see the AHCI specification (revision 1.1).

The SATA controller is capable of command list override (CLO) operation. CLO is activated by setting SATA_PxCMD[3] CLO = 0b1, which is expected only when SATA_PxCMD[0] ST = 0x0 (that is, the command list is not processed by the HBA) to avoid HBA-indeterminate behavior. As a consequence of a CLO operation, the SATA_PxTFD register STS_DRQ and STS_BSY flags are cleared. For example, CLO is necessary when the DRQ and BSY flags cannot be cleared by HBA before issuing a software reset (because of some hang condition). For more information on CLO behavior, see the AHCI standard specification.

24.8.4.7.2 Handling the Received FIS Descriptors

The HBA DMA facilitates the reception of the D2H FISs into host system memory. The reception is enabled by the user setting SATA_PxCMD[4] FRE to 0b1. The software is allowed to change the SATA_PxFB / SATA_PxFBU contents and thus move the FIS reception area to different system memory locations, if SATA_PxCMD[14] FR is cleared, which occurs after software writes SATA_PxCMD[4] FRE = 0b0. In this case, further FIS reception is blocked when the internal RxFIFO becomes full. Before setting SATA_PxCMD[4] FRE to 0x1, the user software must ensure that a valid address is programmed into the SATA_PxFB / SATA_PxFBU registers.

Note

The HBA port stores the unknown FISs (up to 64 bytes) in system memory but does not have a specified behavior to indicate an error condition on reception. The unknown FISs must be handled specifically by user software.

24.8.4.8 DMA Port Configuration

The SATA controller handles all data operations on its port with an internal DMA integrated in SATA AHCI core.

The SATA controller DMA transfers all information between system memory and the attached SATA peripheral device, as well as configuration and status FISs. The transmit and receive DMA data paths are independently programmable.

The burst size in both directions is fixed to 16 Dwords. The DMA issues transactions of this size or smaller (in Dword increments), depending on programmed transaction size.

Note

If the programmed transaction size is lower than 16-Dwords (burst size), then the DMA maxes out at transaction size.

The user can also separately program the transaction size for receive and transmit channels by setting the SATA_PxDMACR[7:4] RXTS and SATA_PxDMACR[3:0] TXTS bit fields. The transaction size is the minimum amount of data on which the DMA works. For example, if a FIS comes from the device to the host, the DMA does not begin transferring data into system memory until there is at least SATA_PxDMACR[7:4] RXTS number of Dwords in the receive FIFO. During transmit, the DMA reads data from system memory in SATA_PxDMACR[3:0] TXTS Dword increments to put into the transmit FIFO. Transactions can be broken up into multiple bursts of 16 Dword - size, crossing of a 1-KiB boundary (boundary between 1-KiB address blocks in memory), or end-of-frame. RXFIFO and TXFIFO transaction sizes are limited to maximum half the size of the integrated RXFIFO and TXFIFO buffers. For more information on maximum transaction sizes and RXFIFO/ TXFIFO depths, see the SATA_PxDMACR and SATA_PPARAMR descriptions.

24.8.4.9 Port Multiplier Operation

The SATA controller HBA port supports connection to a bunch of maximum 15 devices through a port multiplier (PM). The communication mechanism is based on PM command-based switching.

24.8.4.9.1 Command-Based Switching Mode

In this mode of operation, user software must initiate transfers to different PM ports so that multiple commands are not outstanding to different devices connected to the PM extension ports.

24.8.4.9.1.1 Port Multiplier NCQ and Non-NCQ Commands Generation

The PM port command servicing order depends on the type of commands in the CL:

- Non-queued command generation to PMs

  When processing commands that are non-queued, the HBA executes each command entry in its entirety before moving to the next entry in the command list. The next entry can include a command to a different PM device port. Because the HBA operates in the order of its command list, system software must not fill the list in sequential order with commands to a single device; otherwise, another device might get starved. For more information, see the AHCI standard specification.
- NCQ command generation to PMs

  Because the SATA controller supports command-based switching only, system software must only add NCQ commands to the command list that target a single port behind the PM, wait for the commands to finish

(SATA_PxSACT bits all cleared), and then add commands for a different port. For more information, see the AHCI standard specification.

24.8.4.9.2 Port Multiplier Enumeration

To enumerate PM devices, a software reset should be issued by SATA controller to the PM port #0xF (this is a point of connection for the PM or a single endpoint device). The software must prepare two H2D Register FISs into the host system memory for a software reset with the PM-port select (PMP) field set to 0xF. To ensure reliable PM device enumeration, software must also perform a CLO operation. For more information, see the AHCI standard specification.

24.8.4.10 Activity LED Generation Functionality

The HBA port drives the device-level sata1_led signal that is active on the following conditions (see also Table 24-439):

- If (SATA_PxCI [31:0]! = 0x0 or SATA_PxSACT[31:0]! = 0x0) and SATA_PxCMD[24] ATAPI = 0b0 (for active nonATAPI commands indication)
- If (SATA_PxCI [31:0]! = 0x0 or SATA_PxSACT[31:0]! = 0x0) and SATA_PxCMD[24] ATAPI = 0b1 and SATA_PxCMD[25] DLAE = 0b1 (for active ATAPI commands indication)

When SATA_PxCI and SATA_PxSACT are cleared to 0x0, the LED on the sata1_led signal-associated pad is driven off.

24.8.4.11 Supported Types of SATA Transfers

SATA controller AHCI mode defines two types of transfers between the host system memory and a SATA peripheral device: DMA and PIO data transfers. Whether the transaction is of a DMA type or PIO type, the HBA fetches and stores data to memory, offloading the CPU. No register is implemented for direct user access to the data port.

PIO transfers specified by PIO Setup FISs are strongly discouraged because of their limited support of error indication. PIO commands use is limited to only few cases in which commands support only PIO mechanism, such as execution of the IDENTIFY DEVICE command and ATAPI command transfers in which the PACKET command is invoked. The SATA controller AHCI mechanism allows multiple DRQ blocks of data per PIO command.

24.8.4.11.1 Supported Higher Level Protocols

The following DMA, PIO, FPDMA-queued, and ATAPI command/data protocols are supported by the SATA controller AHCI engine:

- Standard SATA command-related
  - Nondata command protocol
  - PIO write protocol
  - PIO read protocol
  - DMA write protocol
  - DMA read protocol
- NCQ command-related
  - Write FPDMA-queued protocol
  - Read FPDMA-queued protocol
- ATAPI PACKET command-related
  - PACKET Nondata command protocol
  - PACKET PIO read/write protocol
  - PACKET DMA read/write protocol

24.8.4.12 SATA Controller AHCI Hardware Register Interface

Table 24-446 summarizes the SATA host controller subsystem registers and divides them into functional categories.

Table 24-446. Summary of SATA Host Subsystem Functional Registers

Register	Functional Category	Brief Register Description
SATA_SYSCONFIG	Device-level registers (wrapper of SATA AHCI core)	TI wrapper idle/standby power management configuration register
SATA_CDRLOCK [3]		TI wrapper register which defines default delay of received data valid information.
SATA_IDR	SATA vendor-specific (VS) register	TI high lander identification register
SATA_CAP	SATA controller generic registers	SATA HBA capabilities set 1
SATA_CAP2		SATA HBA capabilities set 2
SATA_VS		SATA HBA register showing supported AHCI specification revision
SATA_GPARAM1R		SATA HBA global parameters set 1
SATA_GPARAM2R		SATA HBA global parameters set 2
SATA_GHC		SATA HBA global AHCI settings
SATA_PPARAMR	SATA HBA ports-related registers	SATA HBA port x[1] parameter settings
SATA_PI		Enables SATA HBA port x[1] functionalities
SATA_IS		Enables all SATA HBA port x[1] associated interrupts (first tier of interrupt schema)
SATA_CCC_PORTS		Enables SATA CCC feature for the port x[1]
SATA_CCC_CTL		CCC feature parameters control and status
SATA_TIMER1MS		Time base for 1-ms unit of CCC timeout
SATA_PxCLB	SATA HBA Port Px[1] SATA Specification- defined Functional Registers	CL base address (lower-half of AHCI 64-bit address)
SATA_PxCLBU		CL base address (upper-half of AHCI 64-bit address) [2]
SATA_PxFB		Received FISs base address (lower-half of AHCI 64-bit address)
SATA_PxFBU		Received FISs base address (upper-half of AHCI 64-bit address)[2]
SATA_PxIE		Port x[1] non-CCC interrupt enable
SATA_PxIS		Port x[1] non-CCC interrupt event status
SATA_PxCMD		Port x[1] command register
SATA_PxTFD		Port x[1] task file data register
SATA_PxSIG		Port x[1] associated SATA signature register
SATA_PxSSTS		Port x[1] SStatus SATA register (See the SATA standard specification.)
SATA_PxSCTL		Port x [1]SControl SATA register (See the SATA standard specification.)
SATA_PxSERR		Port x [1]SError SATA register (See the SATA standard specification.)
SATA_PxSACT		Port x[1] SActive SATA register (See SATA standard specification.)
SATA_PxCI		Port x [1]command activation/completion indication register
SATA_PxSNTF		Port x[1] SNotification register (See the SATA standard specification.)
SATA_OOBR		OOB detection counters setup
SATA_PxDMACR	Non-standard Port Px[1] Control register	Port x[1] associated DMA configuration register

Table 24-446. Summary of SATA Host Subsystem Functional Registers (continued)

Register	Functional Category	Brief Register Description
SATA_TESTR	SATA HBA BIST related registers	Test/normal mode switching
SATA_BISTAFR		BIST FIS-activate register
SATA_BISTCR		BIST control register
SATA_BISTFCTR		BIST FIS count register
SATA_BISTSR		BIST status register
SATA_BISTDECR		BIST Dword error count register

(1) x = 0, as only SATA HBA port 0 is implemented
(2) Only the 4-lower bits meaningful in the device, the others must be kept at '0'
(3) Under normal conditions, the user is supposed to keep this register at its default (power-on-reset) value.

24.8.5 SATA Controller Low Level Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the SATA host controller.

24.8.5.1 Global Initialization

24.8.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements to initialize the surrounding modules when the SATA host controller is used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the SATA host controller (for more information, see Section 24.6.3 and Section 24.6.2). Table 24-447 describes the global initialization of surrounding modules.

Table 24-447. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module functional and interface clocks must be enabled (see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*).
Control Module	Module-specific pad muxing and other PHY power configurations must be set in the control module (see *Pad Configuration Registers* and *Control Module Register Manual*, in *Control Module*).
IRQ_CROSSBAR	IRQ_CROSSBAR must be configured to enable the interrupt from the SATA controller (see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*).
L4_CFG and L3_MAIN interconnects	For more information about the interconnect configuration, see *L3_MAIN Interconnect Agents* and *L4 Interconnect*, in *Interconnect.*

Note

The MPU INTC configuration is required when using the interrupt communication mode.

24.8.5.1.2 SATA Controller Global Initialization

24.8.5.1.2.1 Main Sequence SATA Controller Global Initialization

Table 24-448 describes the procedure to initialize the SATA host controller after a POR.

Table 24-448. SATA Controller Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure the OCP2SCP3 adapter for PLLCTRL_SATA.	See Section 26.1.5, *SATA PHY Subsystem Low - Level Programming Model* in Chapter 26, *Shared PHY Component Subsystems*.	-
Configure the DPLL_SATA.	See Section 26.1.5, *SATA PHY Subsystem Low - Level Programming Model* in Chapter 26, *Shared PHY Component Subsystems*.	-
Configure the SATA_PHY.	See Section 26.1.5, *SATA PHY Subsystem Low - Level Programming Model* in Chapter 26, *Shared PHY Component Subsystems*.	-

Table 24-448. SATA Controller Global Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Perform all firmware capability writes. [(1)]	See Section 24.8.5.1.2.2, *SubSequence – Firmware Capability Writes*.	-
Set up all appropriate structures in memory per the AHCI standard specification.	See Section 24.8.4.6, *System Memory FIS Descriptors*. See also the AHCI standard specification.	-
Write a valid CL memory base address to the SATA_PxCLB register.	SATA_PxCLB [31:10] CLB	CL Base address (lower-half)
	SATA_PxCLBU [3:0] CLBU	CL Base address (upper-half) Only 4-lower bits meaningful in the device, other must be always written to '0'-s
Write a valid Rx FIS memory base address to the SATA_PxFB register.	SATA_PxFB [31:8] FB	Rx FIS Base address (lower-half)
	SATA_PxFBU [3:0] FBU	Rx FIS Base address (upper-half) Only 4-lower bits meaningful in the device, other must be always written to '0'-s
Issue a software type of reset depending on the error condition.	See Section 24.8.4.3.2, *Software Initiated Resets*.	-
Set the appropriate bits in the SATA_PxCMD register.	See Section 24.8.4.7.1, *Software Processing of the Port Command List*. See also the AHCI standard specification.	-
Program the SATA_PxSCTL register to configure SATA interface capabilities.	See the AHCI standard specification.	-
Program the SATA_PxDMACR register.	See Section 24.8.4.8, *DMA Port Configuration*.	-
Enable the interrupts at port 0.	SATA_IS[0] IPS	0x1
Enable the involved port 0 specific interrupts.	See Section 24.8.4.5.2, *Levels of Interrupt Control*, and Section 24.8.4.5.4, *Interrupt Condition Controll*.	-
Enable the FIS reception in the SATA_PxCMD register.	SATA_PxCMD[4] FRE	0x1
Spin up the device (if required).	See the AHCI standard specification and the SATA standard specification.	-

(1) This step has an effect only on hardware initialization.

Note

The SATA AHCI Controller always manipulates 64-bit memory pointers, and the master interface has 64-bit addresses, but the actual AHCI master interface integration in the device uses ONLY the 36-lower bits. In that case, even though the AHCI support indicates “64-bit” in the SATA_CAP[31] S64A, ONLY the 4-lower bits (SATA_PxCLBU [3:0] / SATA_PxFBU [3:0]) are meaningful in the device.**The upper 28 bits of the address, i.e.** SATA_PxCLBU [31:4] / SATA_PxFBU [31:4] **, must be always SW-programmed to '0'.**

Note

Default value of the SATA_PxCLBU / SATA_PxFBU registers is 0x0, so that a 32-bit SW driver which never accesses them works seamlessly, by accessing only the 32-bit (lower half) addresses which reside in the SATA_PxCLB / SATA_PxFB registers.

24.8.5.1.2.2 SubSequence – Firmware Capability Writes

Table 24-449 lists the firmware capability write sequence by which SATA HBA initialization is performed.

Note

In the following sequence, SATA controller accesses hardware initialization domain registers that can be written only one time after POR.

Table 24-449. Firmware Capability Writes

Step	Register/Bit Field/Programming Model	Value
Disable the staggered spin-up support feature.(1)	SATA_CAP[27] SSS	0x0
Enable the SATA controller port 0.	SATA_PI[0] PI	0x1
Disable the mechanical presence detect feature.(1)	SATA_CAP[28] SMPS	0x0
Disable the external signal-only connector.(1)	SATA_PxCMD[22:21] ESP	0x0
Disable the cold presence detect feature.(1)	SATA_PxCMD[20] CPD	0x0
Disable support for the mechanical presence switch.	SATA_PxCMD[19] MPSP	0x0
Disable the hot-plug-capable port feature.	SATA_PxCMD[18] HPCP	0x0
If PM is attached:		
Set the SATA_PxCMD [17] PMA bit to notify HBA that the PM is attached to HBA port 0.	SATA_PxCMD[17] PMA	0x1
ELSE		
Clear the SATA_PxCMD[17] PMA bit as a notification that a single-target SATA device is attached to port 0.	SATA_PxCMD[17] PMA	0x0

(1) It makes no sense to enable staggered spin-up because only one HBA port is used.

24.8.5.1.3 Issue Command - Main Sequence

The procedure in Table 24-450 prepares a command list entry and issues its corresponding command.

Table 24-450. Prepare and Issue a Command

Step	Register/Bit Field/Programming Model	Value
Build command header and command table into host system memory.	For details, see Section 24.8.4.6.1, *Command List Structure Basics*.	-
Create the command corresponding PRD table.		
Queue the command to the command list.	See Section 24.8.4.7.1, *Software Processing of the Port Command List*.	-
Set the SATA_PxCMD[0] ST bit to direct the DMA to start processing the CL.	SATA_PxCMD[0] ST	0x1(1)
IF a nonNCQ command is issued:		
Set the corresponding SATA_PxCI.CI bit to activate a command already prepared in the i-th slot of the CL .	SATA_PxCI[i] CI	0x-
ELSE		
Indicate the NCQ slot positions through the SATA_PxSACT register. For details on TAG tracking, see Section 24.8.4.7.1.	SATA_PxSACT[TAG] DS	0x-
Set the corresponding SATA_PxCI.CI bit to activate the NCQ command already prepared in the i-th slot of the CL.	SATA_PxCI[TAG] CI	0x-

(1) See the AHCI standard specification for important restrictions when PXCMD[0] ST can be software-asserted to 1.

24.8.5.1.4 Receive FIS—Main Sequence

The procedure in Table 24-451 is used to handle a FIS reception.

Table 24-451. FIS Reception

Step	Register/Bit Field/Programming Model	Value
Read SATA_PxIS to determine the type of FIS.	For more information, see Section 24.8.4.7.2, *Handling the Received FIS Descriptors*.	-

24.8.6 SATA Controller Register Manual

24.8.6.1 SATA Controller Instance Summary

Table 24-452. SATA Instance Summary

Module Name	Module Base Address	Size (Bytes)
DWC_ahsata	0x4A14 0000	4352
SATAMAC_wrapper	0x4A14 1100	256

24.8.6.2 DWC_ahsata Registers

24.8.6.2.1 DWC_ahsata Register Summary

Table 24-453. DWC_ahsata Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address DWC_ahsata
SATA_CAP	RW	32	0x0000 0000	0x4A14 0000
SATA_GHC	RW	32	0x0000 0004	0x4A14 0004
SATA_IS	RW	32	0x0000 0008	0x4A14 0008
SATA_PI	RW	32	0x0000 000C	0x4A14 000C
SATA_VS	R	32	0x0000 0010	0x4A14 0010
SATA_CCC_CTL	RW	32	0x0000 0014	0x4A14 0014
SATA_CCC_PORTS	RW	32	0x0000 0018	0x4A14 0018
SATA_CAP2	R	32	0x0000 0024	0x4A14 0024
SATA_BISTAFR	R	32	0x0000 00A0	0x4A14 00A0
SATA_BISTCR	RW	32	0x0000 00A4	0x4A14 00A4
SATA_BISTFCTR	R	32	0x0000 00A8	0x4A14 00A8
SATA_BISTSR	R	32	0x0000 00AC	0x4A14 00AC
SATA_BISTDECR	R	32	0x0000 00B0	0x4A14 00B0
SATA_OOBR	RW	32	0x0000 00BC	0x4A14 00BC
SATA_TIMER1MS	RW	32	0x0000 00E0	0x4A14 00E0
SATA_GPARAM1R	R	32	0x0000 00E8	0x4A14 00E8
SATA_GPARAM2R	R	32	0x0000 00EC	0x4A14 00EC
SATA_PPARAMR	R	32	0x0000 00F0	0x4A14 00F0
SATA_TESTR	RW	32	0x0000 00F4	0x4A14 00F4
SATA_VERSIONR	R	32	0x0000 00F8	0x4A14 00F8
SATA_IDR	R	32	0x0000 00FC	0x4A14 00FC
SATA_PxCLB	RW	32	0x0000 0100	0x4A14 0100
SATA_PxCLBU	RW	32	0x0000 0104	0x4A14 0104
SATA_PxFB	RW	32	0x0000 0108	0x4A14 0108
SATA_PxFBU	RW	32	0x0000 010C	0x4A14 010C
SATA_PxIS	RW	32	0x0000 0110	0x4A14 0110
SATA_PxIE	RW	32	0x0000 0114	0x4A14 0114
SATA_PxCMD	RW	32	0x0000 0118	0x4A14 0118

Table 24-453. DWC_ahsata Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address DWC_ahsata
SATA_PxTFD	R	32	0x0000 0120	0x4A14 0120
SATA_PxSIG	R	32	0x0000 0124	0x4A14 0124
SATA_PxSSTS	R	32	0x0000 0128	0x4A14 0128
SATA_PxSCTL	RW	32	0x0000 012C	0x4A14 012C
SATA_PxSERR	RW	32	0x0000 0130	0x4A14 0130
SATA_PxSACT	RW	32	0x0000 0134	0x4A14 0134
SATA_PxCI	RW	32	0x0000 0138	0x4A14 0138
SATA_PxSNTF	RW	32	0x0000 013C	0x4A14 013C
SATA_PxDMACR	RW	32	0x0000 0170	0x4A14 0170

24.8.6.2.2 DWC_ahsata Register Description

Table 24-454. SATA_CAP

Address Offset	0x0000 0000		
Physical Address	0x4A14 0000	**Instance**	DWC_ahsata
Description	Capabilities register: Basic capabilities of the SATA AHCI core. Some fields can be written once after reset, read-only.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S64A	SNCQ	SSNTF	SMPS	SSS	SALP	SAL	SCLO	ISS				SNZO	SAM	SPM	FBSS	PMD	SSC	PSC	NCS					CCCS	EMS	SXS	NP				

Bits	Field Name	Description	Type	Reset
31	S64A	Supports 64-bit addressing Read 0x1: 64-bit addressing supported Read 0x0: 32-bit addressing supported	R	1
30	SNCQ	Supports NCQ (Native Command Queuing) Controller supports SATA NCQ by handling DMA setup FIS natively. Read 0x1: Supported Read 0x0: Not supported	R	1
29	SSNTF	Supports SNotification register Controller supports SATA_PxSNTF (SNotification) register and its associated functionality. Read 0x1: Supported Read 0x0: Not supported	R	1
28	SMPS	Supports mechanical presence switch Support of a mechanical presence switch for hot plug operation, depending on integration Writable once after power up, read-only afterward 0x0: Not supported 0x1: Supported	RW WSpecial	0
27	SSS	Supports staggered spin-up Controller can support this feature through SATA_PxCMD.SUD Writable once after power up, read-only afterward 0x0: Not supported 0x1: Supported	RW WSpecial	0

Bits	Field Name	Description	Type	Reset
26	SALP	Supports aggressive link power management Read 0x1: Supported Read 0x0: Not supported	R	1
25	SAL	Supports Activity LED Read 0x1: Supported Read 0x0: Not supported	R	1
24	SCLO	Supports command list override Supports the SATA_PxCMD.CLO bit functionality for enumeration of PM devices Read 0x1: Supported Read 0x0: Not supported	R	1
23:20	ISS	Interface speed support Maximum speed the HBA can support Read 0x3: Gen 3 = 6 Gbps Read 0x2: Gen 2 = 3 Gbps Read 0x1: Gen 1 = 1.5 Gbps	R	0x2
19	SNZO	Supports Non-zero DMA offsets Read 0x1: Supported Read 0x0: Not supported	R	0
18	SAM	Supports AHCI mode only SATA controller supports AHCI mode only and does not support legacy, task file-based register interface. Read 0x1: Supported Read 0x0: Not supported	R	1
17	SPM	Supports PM (Port Multiplier) SATA controller supports command-based switching PM on any port. Read 0x1: Supported Read 0x0: Not supported	R	1
16	FBSS	FIS-based switching supported Support of PM FIS-based switching. Read 0x1: Supported Read 0x0: Not supported	R	0
15	PMD	PIO Multiple DRQ Support of multiple DRQ block data transfers for the PIO command protocol Read 0x1: Supported Read 0x0: Not supported	R	1
14	SSC	SLUMBER state capable Support of transitions to the interface SLUMBER power management state Read 0x1: Supported Read 0x0: Not supported	R	1
13	PSC	PARTIAL state capable Support of transitions to the interface PARTIAL power management state Read 0x1: Supported Read 0x0: Not supported	R	1
12:8	NCS	Number of command slots: slots supported by the SATA controller, minus 1 Read 0x1F: 32 command slots	R	0x1F

Bits	Field Name	Description	Type	Reset
7	CCCS	Command completion coalescing supported Read 0x1: Supported Read 0x0: Not supported	R	1
6	EMS	Enclosure management supported Read 0x1: Supported Read 0x0: Not supported	R	0
5	SXS	Supports external SATA Read 0x1: Supported Read 0x0: Not supported	R	0
4:0	NP	Number of ports: ports supported by the SATA controller, minus 1 Read 0x1: 2 ports Read 0x0: 1 port	R	0x00

Table 24-455. SATA_GHC

Address Offset	0x0000 0004		
Physical Address	0x4A14 0004	**Instance**	DWC_ahsata
Description	Global HBA control		
Type	RW		

31	30–2	1	0
AE	RESERVED	IE	HR

Bits	Field Name	Description	Type	Reset
31	AE	AHCI enable Always set because SATA controller supports AHCI mode only, as indicated by the SATA_CAP.SAM = 1	R	1
30:2	RESERVED		R	0x0000 0000
1	IE	Interrupt enable Global enable of SATA controller interrupts. Reset on global reset (SATA_GHC.HR = 1). 0x0: All interrupt sources from all ports are disabled (masked). 0x1: Interrupts are enabled and any SATA controller interrupt event causes interrupt output assertion.	RW	0
0	HR	HBA reset Global reset control Write 0x0: No action Write 0x1: Start global reset: All state machines that relate to data transfers and queuing return to an IDLE state, and all ports are reinitialized by sending COMRESET if staggered spin-up is not supported. If staggered spin-up is supported, it is the responsibility of the software to spin up each port after this reset completes. Read 0x1: Reset is ongoing. Read 0x0: Reset is inactive (done).	RW	0

Table 24-456. SATA_IS

Address Offset	0x0000 0008		
Physical Address	0x4A14 0008	**Instance**	DWC_ahsata
Description	Interrupt status Indicates which port has a pending interrupt		

Table 24-456. SATA_IS (continued)

Type	RW

31-1	0
RESERVED	IPS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	IPS	Interrupt pending status. Bit-significant field. Bits are set by ports that have interrupt events pending in the SATA_PxIS bits and enabled in SATA_PxIE. Set bits are cleared by software writing 1 to them.	RW W1toClr	0

Table 24-457. SATA_PI

Address Offset	0x0000 000C		
Physical Address	0x4A14 000C	**Instance**	DWC_ahsata
Description	Ports implemented Indicates which ports are exposed by the SATA controller and available for use		
Type	RW		

31-1	0
RESERVED	PI

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	PI	Ports implemented. Bit-significant field. Writable once after power up, read-only afterward. If a bit is set (1), the corresponding port is available; else (0) it is not. Only bits 0 to SATA_CAP.NP can be set to 1. At least one bit must be set to 1.	RW WSpecial	0

Table 24-458. SATA_VS

Address Offset	0x0000 0010		
Physical Address	0x4A14 0010	**Instance**	DWC_ahsata
Description	AHCI version supported: 1.3 WARNING: Controller complies fully with AHCI version 1.10 and also complies with AHCI version 1.3 except for FIS-based switching, which is not currently supported.		
Type	R		

31-16	15-0
MJR	MNR

Bits	Field Name	Description	Type	Reset
31:16	MJR	Major Version Number: 1	R	0x0001
15:0	MNR	Minor Version Number: 3.00	R	0x0300

Table 24-459. SATA_CCC_CTL

Address Offset	0x0000 0014		
Physical Address	0x4A14 0014	**Instance**	DWC_ahsata
Description	CCC (Command Completion Coalescing) control Used to configure the CCC feature for the SATA controller Reset on global reset		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TV	CC	INT	RESERVED	EN

Bits	Field Name	Description	Type	Reset
31:16	TV	Time-out value. Specifies the CCC time-out value in 1-ms intervals Loaded prior to enabling CCC; becomes read-only when SATA_CCC_CTL.EN = 1 0x0: Reserved value; do not use. 0x1 - 0xFFFF: timeout slectable between within the range (1 - 65535) ms.	RW	0x0001
15:8	CC	Command completions Number of command completions necessary to cause a CCC interrupt Loaded prior to enabling CCC, becomes read-only when SATA_CCC_CTL.EN = 1 0x0: CCC interrupts generated based on the timer, not on completed commands count 0x1 - 0xFF: specifies the number of commands upon which completion a CCC interrupt is generated. The number of commands to complete before interrupt is triggered are selectable within the range (1 - 255) commands.	RW	0x01
7:3	INT	Interrupt Number of the interrupt used by the CCC feature, using the number of ports configured for the core When a CCC interrupt occurs, the SATA_IS.IPS[INT] bit is set to 1.	R	0x01
2:1	RESERVED		R	0x0
0	EN	Enable CCC enable 0x0: CCC feature is disabled and no CCC interrupts are generated. SATA_CCC_CTL.TV and .CC are writable. 0x1: CCC feature is enabled and CCC interrupts can be generated based on the time-out or command completion conditions. All other SATA_CCC_CTL fields are read-only.	RW	0

Table 24-460. SATA_CCC_PORTS

Address Offset	0x0000 0018		
Physical Address	0x4A14 0018	**Instance**	DWC_ahsata
Description	CCC ports Specifies the ports that are coalesced as part of the CCC feature when SATA_CCC_CTL.EN = 1 Reset on global reset		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															PRT

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	PRT	Ports Bit-significant field Set a bit to 1 to make the corresponding port part of the CCC feature. Bits set to 1 in this register have the same bit set to 1 in register PI.	RW	0

Table 24-461. SATA_CAP2

Address Offset	0x0000 0024		
Physical Address	0x4A14 0024	**Instance**	DWC_ahsata
Description	Extended capabilities		
Type	R		

31–3	2	1	0
RESERVED	APST	NVMP	BOH

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	APST	Automatic PARTIAL to SLUMBER transitions Read 0x1: Supported Read 0x0: Not supported	R	1
1	NVMP	NVMHCI present Read 0x1: Supported Read 0x0: Not supported	R	0
0	BOH	BIOS/OS Handoff Read 0x1: Supported Read 0x0: Not supported	R	0

Table 24-462. SATA_BISTAFR

Address Offset	0x0000 00A0		
Physical Address	0x4A14 00A0	**Instance**	DWC_ahsata
Description	Built-In, Self-Test (BIST) Activate FIS Register Reset on global reset or port reset		
Type	R		

31–16	15–8	7–0
RESERVED	NCP	PD

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	NCP	Noncompliant pattern Least significant byte of the received BIST Activate FIS second DWORD (bits [7:0]). This value defines the required pattern for far-end transmit-only mode (SATA_BISTAFR.PD =0x80 or 0xA0). If none of the listed values is decoded, the simultaneous switching pattern is transmitted by default. 0x4A: High frequency test pattern (HFTP) 0x7F: Simultaneous switching outputs pattern (SSOP) 0xF1: Low transition density pattern (LTDP) 0x8B: Lone Bit pattern (LBP) 0xB5: High transition density pattern (HTDP) 0x7E: Low frequency test pattern (LFTP) 0x78: Mid frequency test pattern (MFTP) 0xAB: Low frequency spectral component pattern (LFSCP)	R	0x00

Bits	Field Name	Description	Type	Reset
7:0	PD	Pattern definition Pattern definition field of the received BIST Activate FIS - bits [23:16] of the first DWORD. Puts the SATA controller in one of the listed BIST modes Read 0x10: Far-end retimed Read 0xC0: Far-end transmit only Read 0xE0: Far-end transmit only with scrambler bypassed Read 0x8: Far-end analog (if PHY supports this mode)	R	0x00

Table 24-463. SATA_BISTCR

Address Offset	0x0000 00A4		
Physical Address	0x4A14 00A4	**Instance**	DWC_ahsata
Description	BIST control register Reset on global reset or port reset		
Type	RW		

31:21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3:0
RESERVED	FERLB	RESERVED	TXO	CNTCLR	NEALB	LLB	RESERVED	ERRLOSSEN	SDFE	RESERVED	LLC_RPD	LLC_DESCRAM	LLC_SCRAM	RESERVED	ERREN	FLIP	PV	PATTERN

Bits	Field Name	Description	Type (1)	Reset
31:21	RESERVED		R	0x000
20	FERLB	Far-end retimed loopback Write 0x0: No action Write 0x1: Puts the DWC_ahsata link into far-end retimed mode without the BIST activate FIS, regardless of whether the device is connected or disconnected (link in NOCOMM state) Read 0x0: Read returns 0	WO	0
19	RESERVED		R	0
18	TXO	Transmit only 0x0: No action 0x1: Initiate transmission of one of the noncompliant patterns defined by the SATA_BISTCR.PATTERN value when the device is disconnected.	W	0
17	CNTCLR	Counter clear Clears BIST error count registers Write 0x0: No action Write 0x1: Clear SATA_BISTFCTR, SATA_BISTSR, and SATA_BISTDECR registers Read 0x0: Read returns 0	WO	0
16	NEALB	Near-end analog loopback This mode should be initiated in the PARTIAL or SLUMBER power state or with the device disconnected from the port PHY (link NOCOMM state). BIST Activate FIS is not sent to the device in this mode. Write 0x0: No action Write 0x1: Places the port PHY in near-end analog loopback mode. SATA_BISTCR.PATTERN bit field contains the appropriate pattern.	WO	0

Bits	Field Name	Description	Type[(1)]	Reset
15	LLB	Lab Loopback Mode Masks out phy_sig_det from the OOB detector in BIST Loopback Mode. To exit BIST Loopback mode, clear the register bit then issue COMRESET / receive COMINIT.	RW	0
14	RESERVED		R	0
13	ERRLOSSEN	Always keep this bit at default value.	RW	0
12	SDFE	Signal detect feature enable Not affected by global reset or port reset 0x0: Link layer feature to handle unstable/absent phy_sig_det signal is disabled. 0x1: Link layer feature to handle unstable/absent phy_sig_det signal is enabled.	RW	0
11	RESERVED	Only write 0 into this reserved field to avoid undefined results.	RW	0
10	LLC_RPD	Link layer control, repeat primitive drop In normal mode, the function can be changed only during port reset (SATA_PxSCTL.DET = 0x1). 0x0: Repeat primitive drop function disabled in normal mode, enabled in BIST mode 0x1: Repeat primitive drop function enabled in normal mode, disabled in BIST mode	RW	1
9	LLC_DESCRAM	Link layer control, descrambler In normal mode, the function can be changed only during port reset (SATA_PxSCTL.DET = 0x1). 0x0: Descrambler disabled in normal mode, enabled in BIST mode 0x1: Descrambler enabled in normal mode, disabled in BIST mode	RW	1
8	LLC_SCRAM	Link layer control, scrambler In normal mode, the function can be changed only during port reset (SATA_PxSCTL.DET = 0x1). Hardware–cleared (enabled) when the port enters a responder far-end transmit BIST mode with scrambling enabled (SATA_BISTAFR.PD = 0x80). 0x0: Scrambler disabled in normal mode, enabled in BIST mode. 0x1: Scrambler enabled in normal mode, disabled in BIST mode.	RW	1
7	RESERVED		R	0
6	ERREN	Error enable Allow or filter (disable) PHY internal errors outside the FIS boundary to set corresponding SATA_PxSERR bits 0x0: Filter errors outside the FIS; allow errors inside the FIS. 0x1: Allow errors outside or inside the FIS.	RW	0
5	FLIP	Flip disparity Change disparity of the current test pattern to the opposite each time its state is changed by software.	RW	0
4	PV	Pattern version Selects either short or long version of the SSOP, HTDP, LTDP, LFSCP, COMP pattern 0x0: Short pattern version 0x1: Long pattern version	RW	0

Bits	Field Name	Description	Type(1)	Reset
3:0	PATTERN	Pattern Defines one of the listed SATA-compliant patterns for far-end retimed/ far-end analog/ near-end analog initiator modes, or noncompliant patterns for transmit-only responder mode when initiated by software writing to the SATA_BISTCR.TXO bit 0x0: Simultaneous switching outputs pattern (SSOP) 0x1: High transition density pattern (HTDP) 0x2: Low transition density pattern (LTDP) 0x3: Low frequency spectral component pattern (LFSCP) 0x4: Composite pattern (COMP) 0x5: Lone bit pattern (LBP) 0x6: Mid-frequency test pattern (MFTP) 0x7: High frequency test pattern (HFTP) 0x8: Low frequency test pattern (LFTP)	RW	0x0

(1) WO = A write-only accessible bit field

Table 24-464. SATA_BISTFCTR

Address Offset	0x0000 00A8		
Physical Address	0x4A14 00A8	**Instance**	DWC_ahsata
Description	BIST frame-information-structure CounT register Received BIST FIS count in the loopback initiator far-end retimed, far-end analog, and near-end analog modes. Updated each time a new BIST FIS is received. Reset by global reset, port reset (COMRESET), or by writing 1 to SATA_BISTCR.CNTCLR Does not roll over and freezes when the FFFF_FFFFh value is reached. It takes approximately 65 hours of continuous BIST operation to reach this value.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	BIST FIS Count	R	0x0000 0000

Table 24-465. SATA_BISTSR

Address Offset	0x0000 00AC		
Physical Address	0x4A14 00AC	**Instance**	DWC_ahsata
Description	BIST status register Errors detected in the received BIST FIS in the loopback initiator far-end retimed, far-end analog, and near-end analog modes Updated each time a new BIST FIS is received Reset on global reset, port reset (COMRESET), or by writing 1 to SATA_BISTCR.CNTCLR		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BRSTERR								FRAMERR															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
23:16	BRSTERR	Burst error count. Accumulated each time a burst error condition is detected: DWORD error is detected in the received frame and 1.5 seconds (27,000 frames) passed since the previous burst error was detected. Value does not roll over and freezes at FFh. Read 0xFF: Max error count reached or exceeded Read 0x0: No error detected	R	0x00
15:0	FRAMERR	Frame error count. New value is added to the old value each time a new BIST frame with a CRC error is received. Does not roll over and freezes at FFFFh Read 0xFFFF: Maximum error count reached or exceeded. Read 0x0: No error detected	R	0x0000

Table 24-466. SATA_BISTDECR

Address Offset	0x0000 00B0		
Physical Address	0x4A14 00B0	**Instance**	DWC_ahsata
Description	BIST double-word error count register Number of DWORD errors detected in the received BIST frame in the loopback initiator far-end retimed, far-end analog, and near-end analog modes Updated each time a new BIST frame is received, when the parameter BIST_MODE = DWORD. Reset on global reset, port reset (COMRESET), or by writing 1 to SATA_BISTCR.CNTCLR.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWERR																															

Bits	Field Name	Description	Type	Reset
31:0	DWERR	DWORD error count. New value is added to the old value each time a new BIST frame is received. The DWERR value does not roll over, and freezes when it exceeds 0xFFFF_F000. Read 0x0: No error detected Read 0xFFFFF000: Max error count reached or exceeded	R	0x0000 0000

Table 24-467. SATA_OOBR

Address Offset	0x0000 00BC		
Physical Address	0x4A14 00BC	**Instance**	DWC_ahsata
Description	OOB (Out Of Band Register) register Controls the link layer OOB detection counters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WE	CWMIN							CWMAX								CIMIN								CIMAX							

Bits	Field Name	Description	Type	Reset
31	WE	WRITE_ENABLE 0x0: SATA_OOBR bits [30:0] are read-only. 0x1: SATA_OOBR bits [30:0] can be written.	RW	0
30:24	CWMIN	COMWAKE_MIN, in OOB rx clock cycles Read-only when SATA_OOBR.WE = 0	RW WSpecial	0x0B
23:16	CWMAX	COMWAKE_MAX, in OOB rx clock cycles Read-only when SATA_OOBR.WE = 0	RW WSpecial	0x15

Bits	Field Name	Description	Type	Reset
15:8	CIMIN	COMINIT_MIN, in OOB rx clock cycles Read-only when SATA_OOBR.WE = 0	RW WSpecial	0x24
7:0	CIMAX	COMINIT_MAX, in OOB rx clock cycles Read-only when SATA_OOBR.WE=0	RW WSpecial	0x40

Table 24-468. SATA_TIMER1MS

Address Offset	0x0000 00E0		
Physical Address	0x4A14 00E0	**Instance**	DWC_ahsata
Description	Timer 1 ms Configuration to generate the 1-ms tick for the CCC logic Must be initialized before using the CCC feature Reset on power up, not affected by global reset		
Type	RW		

31:20	19:0
RESERVED	TIMV

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	TIMV	OCP bus clock frequency in kHz (for example, reset value is 100,000 = 100 MHz)	RW	0x1 86A0

Table 24-469. SATA_GPARAM1R

Address Offset	0x0000 00E8		
Physical Address	0x4A14 00E8	**Instance**	DWC_ahsata
Description	Global parameters register 1 Hardware configuration of the DWC AHCI SATA core		
Type	R		

31	30	29:28	27	26:21	20:15	14	13	12:11	10	9:8	7	6	5:3	2:0
ALIGN_M	RX_BUFFER	PHY_DATA	PHY_RST	PHY_CTRL	PHY_STAT	LATCH_M	BIST_M	PHY_TYPE	RETURN_ERR	AHB_ENDIAN	S_HADDR	M_HADDR	S_HDATA	M_HDATA

Bits	Field Name	Description	Type	Reset
31	ALIGN_M	RX data alignment Read 0x1: Yes Read 0x0: No	R	1
30	RX_BUFFER	RX data buffer implemented Read 0x1: Yes Read 0x0: No	R	1
29:28	PHY_DATA	PHY data width (in 8- or 10-bit characters) Read 0x2: 4 characters Read 0x1: 2 characters Read 0x0: 1 character	R	0x0
27	PHY_RST	PHY reset mode Read 0x1: High Read 0x0: Low	R	1
26:21	PHY_CTRL	PHY control width (in bits)	R	0x00
20:15	PHY_STAT	PHY status width (in bits)	R	0x00

Bits	Field Name	Description	Type	Reset
14	LATCH_M	Test mode lock-up latches Read 0x1: Yes Read 0x0: No	R	0
13	BIST_M	BIST loopback checking depth Read 0x1: DWORD Read 0x0: FIS	R	0
12:11	PHY_TYPE	PHY interface type Read 0x1: Preset Read 0x0: Configurable 0x2, 0x3: Reserved	R	0x0
10	RETURN_ERR	Error response on illegal access Read 0x1: Yes Read 0x0: No	R	0
9:8	AHB_ENDIAN	Endianness of master and slave Read 0x2: Pin-configurable dynamic endianness Read 0x1: Big-endian Read 0x0: Little-endian	R	0x0
7	S_HADDR	Slave address bus width Read 0x1: 64-bit address Read 0x0: 32-bit address	R	0
6	M_HADDR	Master address bus width Read 0x1: 64-bit address Read 0x0: 32-bit address	R	1
5:3	S_HDATA	Slave Data Bus Width Read 0x3: 256-bit Read 0x2: 128-bit Read 0x1: 64-bit Read 0x0: 32-bit	R	0x0
2:0	M_HDATA	Master Data Bus Width Read 0x3: 256-bit Read 0x2: 128-bit Read 0x1: 64-bit Read 0x0: 32-bit	R	0x0

Table 24-470. SATA_GPARAM2R

Address Offset	0x0000 00EC		
Physical Address	0x4A14 00EC	**Instance**	DWC_ahsata
Description	Global parameters register 2 Hardware configuration of the DWC AHCI SATA core, continued		
Type	R		

31:15	14	13	12	11	10	9	8:0
RESERVED	DEV_CP	DEV_MP	ENCODE_M	RXOOB_CLK_M	RXOOB_M	TXOOB_M	RXOOB_CLK

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0 0000
14	DEV_CP	Cold presence detection implemented in core Read 0x1: Yes Read 0x0: No	R	1
13	DEV_MP	Mechanical presence switch implemented in core Read 0x1: Yes Read 0x0: No	R	1
12	ENCODE_M	8b/10b Encoding/decoding implemented in core Read 0x1: Yes Read 0x0: No	R	1
11	RXOOB_CLK_M	RX OOB clocking mode: Read 0x1: RX OOB detection uses separate clock Read 0x0: Rx OOB detection uses RX clock	R	0
10	RX_OOB_M	RX OOB mode: sequence generation implemented Read 0x1: Yes Read 0x0: No	R	1
9	TX_OOB_M	TX OOB mode: sequence generation implemented Read 0x1: Yes Read 0x0: No	R	1
8:0	RXOOB_CLK	RX OOB clock frequency, in MHz	R	0x096

Table 24-471. SATA_PPARAMR

Address Offset	0x0000 00F0		
Physical Address	0x4A14 00F0	**Instance**	DWC_ahsata
Description	Port parameter register Hardware configuration of the DWC AHCI SATA core port selected by SATA_TESTR.PSEL		
Type	R		

31:12	11	10	9	8	7:4	3:0
RESERVED	TX_MEM_M	TX_MEM_S	RX_MEM_M	RX_MEM_S	TXFIFO_DEPTH	RXFIFO_DEPTH

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0 0000
11	TX_MEM_M	TX FIFO memory mode: Read 0x1: Synchronous Read 0x0: Asynchronous	R	0
10	TX_MEM_S	TX FIFO memory selection: Read 0x1: Internal memory Read 0x0: External memory	R	0
9	RX_MEM_M	RX FIFO memory mode: Read 0x1: Synchronous Read 0x0: Asynchronous	R	0
8	RX_MEM_S	RX FIFO memory selection: Read 0x1: Internal memory Read 0x0: External memory	R	0

Bits	Field Name	Description	Type	Reset
7:4	TXFIFO_DEPTH	Tx FIFO Depth, in dwords (log2) Read 0x3: 8 dwords Read 0x4: 16 dwords Read 0x5: 32 dwords Read 0x6: 64 dwords	R	0x6
3:0	RXFIFO_DEPTH	Rx FIFO Depth, in dwords (log2) Read 0x4: 16 dwords Read 0x5: 32 dwords Read 0x6: 64 dwords Read 0x7: 128 dwords	R	0x7

Table 24-472. SATA_TESTR

Address Offset	0x0000 00F4		
Physical Address	0x4A14 00F4	**Instance**	DWC_ahsata
Description	Test register Puts the SATA controller slave interface in a test mode and selects a port for BIST operation		
Type	RW		

31:19	18:16	15:1	0
RESERVED	PSEL	RESERVED	TEST_IF

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18:16	PSEL	Port select: Selects the port for BIST operation 0x0: Port 0 is selected	RW	0x0
15:1	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
0	TEST_IF	Test interface 0x0: Normal mode: read-back value of some registers might not match the value written, depending on ongoing operations. 0x1: Test mode: Normal operation is disabled; read-back value of the registers match the value written. The following registers/fields can be accessed in this mode: - SATA_GHC.IE - SATA_BISTAFR.NCP and .PD bits become writable. - SATA_BISTCR.LLC .ERREN .FLIP .PV, and .PATTERN - SATA_BISTFCTR, SATA_BISTSR, SATA_BISTDECR become writeable. - SATA_PxCLB / SATA_PxCLBU, SATA_PxFB / SATA_PxFBU - SATA_PxIS.UFS and write-1-to-clear bits become writeable. - SATA_PxIE - SATA_PxCMD.ASP .ALPE .DLAE .ATAPI and .PMA - SATA_PxTFD, SATA_PxSIG become writeable. - SATA_PxSCTL - SATA_PxSERR (write-1-to-clear) bits become writeable. Notes: 1) Interrupt is asserted if any IS register bit is set after setting the corresponding SATA_PxIS and SATA_PxIE bits, and SATA_GHC.IE = 1. 2) SATA_CAP.SMPS/SSS, SATA_PI, SATA_PxCMD.ESP/CPD/MPSP/HPCP cannot be used in test mode. They are written once after POR and become read-only. 3) Global reset must be issued (SATA_GHC.HR=1) after the TEST_IF bit is cleared following the test mode operation.	RW	0

Table 24-473. SATA_VERSIONR

Address Offset	0x0000 00F8		
Physical Address	0x4A14 00F8	**Instance**	DWC_ahsata
Description	Version register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION																															

Bits	Field Name	Description	Type	Reset
31:0	VERSION	Version of DWC SATA controller, ASCII. See [(1)].	R	0x-[(1)]

(1) TI internal data

Table 24-474. SATA_IDR

Address Offset	0x0000 00FC		
Physical Address	0x4A14 00FC	**Instance**	DWC_ahsata
Description	ID register, containing the 32-bit Highlander (HL) revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	See [1].	R	0x-[1]

(1) TI internal data

Table 24-475. SATA_PxCLB

Address Offset	0x0000 0100		
Physical Address	0x4A14 0100	**Instance**	DWC_ahsata
Description	Port command List base address 32-bit base physical address for the command list for this port. Used when fetching commands to execute. The structure pointed to by this address range is 1 KiB in length.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLB																						ZERO									

Bits	Field Name	Description	Type	Reset
31:10	CLB	Command list base address (bits 31:10)	RW	0x00 0000
9:0	ZERO	Always 0 as address is 1 KiB-aligned	R	0x000

Table 24-476. SATA_PxCLBU

Address Offset	0x0000 0104		
Physical Address	0x4A14 0104	**Instance**	DWC_ahsata
Description	Port Command List Base Upper address Upper half of the 64-bit base physical address for the command list for this Port. Used when fetching commands to execute. Remains all 0 when in 32-bit mode. Reserved & read-only when CAP.S64A=0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLBU																															

Bits	Field Name	Description	Type	Reset
31:0	CLBU	Command List Base Upper Address (bits 63:32)[1]	RW	0x0000 0000

(1) Only bits [3:0] are meaningful , the others must be always written to '0'.

Table 24-477. SATA_PxFB

Address Offset	0x0000 0108		
Physical Address	0x4A14 0108	**Instance**	DWC_ahsata
Description	Port Frame-information-structure Base address 32-bit base physical address for received FISes for this port. The structure pointed to by this address range is 256 bytes in length.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB																								ZERO							

Bits	Field Name	Description	Type	Reset
31:8	FB	FIS base address (bits 31:8)	RW	0x00 0000
7:0	ZERO	Always 0 as address is 256-bytes aligned	R	0x00

Table 24-478. SATA_PxFBU

Address Offset	0x0000 010C		
Physical Address	0x4A14 010C	**Instance**	DWC_ahsata

Table 24-478. SATA_PxFBU (continued)

Description	FIS Base Upper Address Upper half of the 64-bit base physical address for received FISes for this port. Remains all 0 with a 32-bit SW driver. Reserved & read-only when CAP.S64A=0
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBU																															

Bits	Field Name	Description	Type	Reset
31:0	FBU	FIS Base Upper Address (bits 63:32)(1)	RW	0x0000 0000

(1) Only bits [3:0] are meaningful , the others must be always written to '0'.

Table 24-479. SATA_PxIS

Address Offset	0x0000 0110		
Physical Address	0x4A14 0110	**Instance**	DWC_ahsata
Description	Port interrupt status Bits are set by internal conditions and cleared (when possible) by writing 1 to them. Reset on global reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPDS	TFES	HBFS	HBDS	IFS	INFS	RESERVED	OFS	IPMS	PRCS	RESERVED														DMPS	PCS	DPS	UFS	SDBS	DSS	PSS	DHRS

Bits	Field Name	Description	Type	Reset
31	CPDS	Cold port detect status Set when the pX_cp_det input changes its state due to the insertion or removal of a device Valid only if the port supports cold presence detection as indicated by the SATA_PxCMD.CPD bit set to 1. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
30	TFES	Task file error status Set whenever the SATA_PxTFD.STS register is updated by the device and the error bit (bit 0) is set. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
29	HBFS	Host bus fatal error status Set when master (DMA) detects an ERROR response from the slave Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
28	HBDS	Host bus data error status This bit is always cleared to 0. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
27	IFS	Interface fatal error status This bit is set when any of the following conditions is detected: 1) SYNC escape is received from the device during H2D register or data FIS transmission. 2) One or more of the following errors are detected during data FIS transfer: - 10B to 8B Decode Error (SATA_PxSERR.DIAG_B) - Protocol (SATA_PxSERR.ERR_P) - CRC (SATA_PxSERR.DIAG_C) - Handshake (SATA_PxSERR.DIAG_H) - PHY not ready (SATA_PxSERR.ERR_C) 3) Unknown FIS is received with good CRC, but the length exceeds 64 bytes. 4) PRD table byte count is 0. 5) DMA setup FIS is received with a TAG corresponding to inactive (SATA_PxSACT bit is cleared) command slot. Port DMA transitions to a fatal state until the software clears SATA_PxCMD.ST bit or resets the interface by way of port reset or global reset. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
26	INFS	Interface nonfatal error status Set when any of the following conditions is detected: 1) One or more of the following errors are detected during nondata FIS transfer: - 10b to 8b decode error (SATA_PxSERR.DIAG_B) - Protocol (SATA_PxSERR.ERR_P) - CRC (SATA_PxSERR.DIAG_C) - Handshake (SATA_PxSERR.DIAG_H) - PHY not ready (SATA_PxSERR.ERR_C) 2) Command list underflow during read operation (that is, DMA read) when the software builds a command table that has more total bytes than the transaction given to the device. In both cases port operation continues normally. When an error is detected during nondata FIS transmission, this FIS is retransmitted continuously until it succeeds, or until the software times out and resets the interface. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
25	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
24	OFS	Overflow status Set when command list overflow is detected during read or write operation when the software builds command table that has fever total bytes than the transaction given to the device. Port DMA transitions to a fatal state until the software clears SATA_PxCMD.ST bit or resets the interface by way of port reset or global reset. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
23	IPMS	Incorrect PM status FIS received from a device in which the PM field did not match what was expected May be set during enumeration of devices on a PM due to the normal PM enumeration process Must be used only after enumeration is complete on the PM Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
22	PRCS	PhyRdy change status Reflects the state of SATA_PxSERR.DIAG_N To clear this bit, clear the SATA_PxSERR.DIAG_N bit to 0. Read 0x1: Internal pX_phy_ready signal changed state Read 0x0: Internal pX_phy_ready signal has not changed state since its last reset.	R	0
21:8	RESERVED		R	0x0000
7	DMPS	Device mechanical presence status Set when the pX_mp_switch input changes its state as a result of a mechanical switch attached to this port opening or closing Valid only when SATA_CAP.SMPS and SATA_PxCMD.MPSP are set Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
6	PCS	Port connect change status This bit reflects the state of the SATA_PxSERR.DIAG_X bit. Cleared only when SATA_PxSERR.DIAG_X is cleared Read 0x1: Change in current connect status Read 0x0: No change in current connect status	R	0

Bits	Field Name	Description	Type	Reset
5	DPS	Descriptor processed A PRD with the I bit set has transferred all of its data. Note. This is an opportunistic interrupt and must not be used to definitively indicate the end of a transfer. Two PRD interrupts could occur close enough together that the second interrupt is missed when the first PRD interrupt is cleared. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
4	UFS	Unknown FIS interrupt An unknown FIS was received and has been copied into system memory. Cleared to 0 by the software clearing the SATA_PxSERR.DIAG_F bit to 0. Note: The UFS bit does not directly reflect the SATA_PxSERR.DIAG_F bit. SATA_PxSERR.DIAG_F bit is set immediately when an unknown FIS is detected, whereas the UFS bit is set when that FIS is posted to memory. The software should wait to act on an unknown FIS until the UFS bit is set to 1 or the two bits may become out of sync. Read 0x1: IRQ event active Read 0x0: Event inactive	R	0
3	SDBS	Set device bits interrupt A Set Device Bits FIS is received with the I bit set and copied into system memory. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
2	DSS	DMA setup FIS interrupt A DMA Setup FIS is received with the I bit set and copied into system memory. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0
1	PSS	PIO setup FIS interrupt A PIO Setup FIS is received with the I bit set, copied into system memory, and the data related to the FIS is transferred. Note: This bit is set even when the data transfer resulted in an error. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	DHRS	Device to host register FIS interrupt A D2H register FIS is received with the I bit set and copied into system memory. Write 0x0: No action Write 0x1: Clear event Read 0x1: IRQ event active Read 0x0: Event inactive	RW W1toClr	0

Table 24-480. SATA_PxIE

Address Offset	0x0000 0114		
Physical Address	0x4A14 0114	**Instance**	DWC_ahsata
Description	Port interrupt enable Enables and disables the reporting of the corresponding interrupt to system software When a bit is set (1), SATA_GHC.IE = 1, and the corresponding interrupt condition in SATA_PxIS is active, then the SATA controller interrupt output is asserted. When a bit is cleared (0), interrupt sources are still reflected in the status registers. Reset on global reset		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21-8	7	6	5	4	3	2	1	0
CPDE	TFEE	HBFE	HBDE	IFE	INFE	RESERVED	OFE	IPME	PRCE	RESERVED	DMPE	PCE	DPE	UFE	SDBE	DSE	PSE	DHRE

Bits	Field Name	Description	Type	Reset
31	CPDE	Cold port detect enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
30	TFEE	Task file error enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
29	HBFE	Host bus fatal error enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
28	HBDE	Host bus data error enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
27	IFE	Interface fatal error enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
26	INFE	Interface non fatal error enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
25	RESERVED		R	0
24	OFE	Overflow enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
23	IPME	Incorrect PM enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0

Bits	Field Name	Description	Type	Reset
22	PRCE	PhyRdy change enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
21:8	RESERVED		R	0x0000
7	DMPE	Device mechanical presence enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
6	PCE	Port connect change enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
5	DPE	Descriptor processed interrupt enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
4	UFE	Unknown FIS interrupt enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
3	SDBE	Set device bits interrupt enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
2	DSE	DMA setup FIS interrupt enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
1	PSE	PIO setup FIS interrupt enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
0	DHRE	Device to host register FIS interrupt enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0

Table 24-481. SATA_PxCMD

Address Offset	0x0000 0118		
Physical Address	0x4A14 0118	**Instance**	DWC_ahsata
Description	Port command		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">ICC</td><td>ASP</td><td>ALPE</td><td>DLAE</td><td>ATAPI</td><td>APSTE</td><td>FBSCP</td><td>ESP</td><td>CPD</td><td>MPSP</td><td>HPCP</td><td>PMA</td><td>CPS</td><td>CR</td><td>FR</td><td>MPSS</td><td colspan="5">CCS</td><td colspan="3">RESERVED</td><td>FRE</td><td>CLO</td><td>POD</td><td>SUD</td><td>ST</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	ICC	Interface communication control Control of power management states of the interface If the link layer is in the L_IDLE state, writes cause the port to request a transition to a given interface state. If the link layer is not in the L_IDLE state, writes have no effect. When a nonreserved, non-0 (No-Op) value is written, the core performs the action and clears the field back to 0 (Idle) Write 0x0: No-Op Read 0x0: Port is ready to accept a new interface control command, although the transition to the previously selected state might not have occurred yet. 0x1: Active 0x2: PARTIAL. SATA device can reject the request and the interface then remains in its current state. 0x6: SLUMBER. SATA device can reject the request and the interface then remains in its current state.	RW	0x0
27	ASP	Aggressive SLUMBER/PARTIAL 0x0: If SATA_PxCMD.ALPE = 1, the port aggressively enters the PARTIAL state when it clears the SATA_PxCI register and the SATA_PxSACT register is cleared when it clears the SATA_PxSACT register and SATA_PxCI is cleared. 0x1: If SATA_PxCMD.ALPE = 1, the port aggressively enters the SLUMBER state when it clears the SATA_PxCI and the SATA_PxSACT register is cleared or when it clears the SATA_PxSACT register and SATA_PxCI is cleared.	RW	0
26	ALPE	Aggressive link power management enable 0x0: Aggressive power management state transition is disabled. 0x1: Port aggressively enters a lower link power state (PARTIAL or SLUMBER) based on the setting of SATA_PxCMD.ASP.	RW	0
25	DLAE	Drive LED on ATAPI enable 0x0: LED is never enabled. 0x1: Port asserts the pX_act_led output when commands are active and SATA_PxCMD.ATAPI = 1.	RW	0
24	ATAPI	Device is ATAPI Used by the port to determine whether or not to assert pX_act_led output when commands are active. 0x0: Connected device is not an ATAPI. 0x1: Connected device is an ATAPI.	RW	0
23	APSTE	Auto PARTIAL to SLUMBER transition enable 0x0: No automatic transition from PARTIAL to SLUMBER 0x1: Link layer transitions from its PARTIAL power management state to SLUMBER state automatically, whether host software-, port (aggressive)-, or device-initiated.	RW	0
22	FBSCP	FIS-based Switching Capable Port May only be set to ? 1? if CAP.SPM = CAP.FBSS = 1 (not the case). Writable once after power up, read-only afterwards. 0x0: port does not support FIS-based switching. 0x1: Port supports Port Multiplier FIS-based switching.	RW	0x0

Bits	Field Name	Description	Type	Reset
21	ESP	External SATA port Writable once after power up, read-only afterward 0x0: Port signal-only connector is not externally accessible. 0x1: Port signal-only connector is externally accessible. SATA_CAP.SXS is also set to 1. Mutually exclusive with SATA_PxCMD.HPCP	RW	0x0
20	CPD	Cold presence detect Writable once after power up, read-only afterward 0x0: Platform does not support cold presence detection on this port. 0x1: Platform supports cold presence detection on this port. SATA_PxCMD.HPCP should be set to 1.	RW	0
19	MPSP	Mechanical presence switch attached to port Writable once after power up, read-only afterward 0x0: Platform does not support a mechanical presence switch on this port. 0x1: Platform supports a mechanical presence switch attached to this port. SATA_PxCMD.HPCP should be set to 1.	RW	0
18	HPCP	Hot plug capable port Writable once after power up, read-only afterward 0x0: Port signal and power connectors are not externally accessible. 0x1: Port signal and power connectors are externally accessible through a joint signal-power connector for blindmate device hot plug.	RW	0
17	PMA	PM attached Software is responsible for detecting the presence of a PM. There is no autodetection. 0x0: No port Multiplier is attached to this Port 0x1: Port Multiplier is attached to this Port	RW	0
16	CPS	Cold presence state Reports whether a device is currently detected on this port as indicated by the pX_cp_det input state (assuming SATA_PxCMD.CPD = 1). Read 0x1: Device detected Read 0x0: No device detected	R	0
15	CR	Command list running For details, see the AHCI state-machine in Section 5.3.2 of the AHCI specification. Read 0x1: Command list DMA engine for this port is running. Read 0x0: Command list is stopped for this port.	R	0
14	FR	FIS receive running For details, see Section 10.3.2 of the AHCI specification. Read 0x1: FIS receive DMA engine for the port is running. Read 0x0: FIS receive DMA engine for the port is stopped.	R	0

Bits	Field Name	Description	Type	Reset
13	MPSS	Mechanical presence switch state Reports the state of a mechanical presence switch attached to this port as indicated by the pX_mp_switch input state (assuming SATA_CAP.SMPS = 1 and SATA_PxCMD.MPSP = 1) Cleared to 0 when SATA_CAP.SMPS = 0 Read 0x1: Switch is open. Read 0x0: Switch is closed.	R	0
12:8	CCS	Current command slot This field is valid when SATA_PxCMD.ST is set to 1 and is set to the command slot value of the command currently issued by the port. When SATA_PxCMD.ST transitions from 1 to 0, this field is reset to 0x00. After SATA_PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0. After the first command is issued, the highest priority slot to issue from next is SATA_PxCMD.CCS + 1. For example, after the port issues its first command, if CCS = 0x00 and SATA_PxCI is set to 0x3, the next command issued is from command slot 1.	R	0x00
7:5	RESERVED		R	0x0
4	FRE	FIS receive enable Must not be set until SATA_PxFB / SATA_PxFBU is programmed with a valid pointer to the FIS receive area Base can be moved after clearing FRE and waiting for FR to clear to 0. 0x0: Received FISes are not accepted by the port, except for the first D2H register FIS after the initialization sequence, and no FISes are posted to the FIS receive area. 0x1: Port can post received FISes into the FIS receive area pointed to by SATA_PxFB and SATA_PxFBU.	RW	0
3	CLO	Command list override Write 0x0: No effect Write 0x1: Request to clear SATA_PxTFD.STS_BSY and SATA_PxTFD.STS_DRQ to 0. Use only immediately prior to setting SATA_PxCMD.ST bit to 1 from a previous value of 0. Any other case results in indeterminate behavior. Read 0x1: Override is active, SATA_PxTFD.STS_BSY and SATA_PxTFD.STS_DRQ are being cleared. Read 0x0: Override is inactive.	RW	0
2	POD	Power-on device Writable if SATA_PxCMD.CPD = 1 (cold presence detection enabled), otherwise read-only -1. 0x0: Disabled 0x1: Port asserts the pX_cp_pod output pin so that it can be used to provide power to a cold-presence detectable port.	RW	0

Bits	Field Name	Description	Type	Reset
1	SUD	Spin-up device Writable if SATA_CAP.SSS = 1 (staggered spin-up supported), else read-only 1. Read-only-0 on power-up until SATA_CAP.SSS bit is written with the required value. 0x0: Clearing the bit from 1 to 0 causes no action on the interface. 0x1: On edge-detect from 0 to 1, the port starts a COMRESET initialization sequence to the device.	RW	0
0	ST	Start 0x0: Port does not process the command list. On transition from 1 to 0, the SATA_PxCI register is cleared by the port on transition to an IDLE state. 0x1: Port processes the command list. On transition from 0 to 1, the port starts processing the command list at entry 0. SATA_PxSERR must be cleared prior to setting ST to 1. For important restrictions on when ST can be set to 1, See Section 10.3.1 of the AHCI specification.	RW	0

Table 24-482. SATA_PxTFD

Address Offset	0x0000 0120		
Physical Address	0x4A14 0120	**Instance**	DWC_ahsata
Description	Port Task File Data: copies specific fields of the task file when FISes are received		
Type	R		

31:16	15:8	7	6:4	3	2:1	0
RESERVED	ERR	STS_BSY	STS_CS2	STS_DRQ	STS_CS	STS_ERR

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	ERR	Err: Latest copy of the task file error register	R	0x00
7	STS_BSY	Status, busy Latest copy of the 8-bit task file status register, bit 7 STS_BSY = Interface is busy	R	0
6:4	STS_CS2	Status, command-specific Latest copy of the 8-bit task file status register, bits 6:4	R	0x7
3	STS_DRQ	Status, data request Latest copy of the 8-bit task file status register, bit 3 STS_DRQ = Data transfer is requested	R	1
2:1	STS_CS	Status, command-specific Latest copy of the 8-bit task file status register, bits 2:1	R	0x3
0	STS_ERR	Status, error Latest copy of the 8-bit task file status register, bit 0 STS_ERR = Error during the transfer	R	1

Table 24-483. SATA_PxSIG

Address Offset	0x0000 0124		
Physical Address	0x4A14 0124	**Instance**	DWC_ahsata
Description	Port signature: Signature received from a device on the first D2H register FIS. Updated once after a reset sequence.		

Table 24-483. SATA_PxSIG (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIG_LBAH								SIG_LBAM								SIG_LBAL								SIG_SCR							

Bits	Field Name	Description	Type	Reset
31:24	SIG_LBAH	Signature, LBA high (cylinder high) register	R	0xFF
23:16	SIG_LBAM	Signature, LBA mid (cylinder low) register	R	0xFF
15:8	SIG_LBAL	Signature, LBA low (sector number) register	R	0xFF
7:0	SIG_SCR	Signature, sector count register	R	0xFF

Table 24-484. SATA_PxSSTS

Address Offset	0x0000 0128		
Physical Address	0x4A14 0128	**Instance**	DWC_ahsata
Description	Port SATA status Current state of the interface and host, updated continuously and asynchronously. When the port transmits a COMRESET to the device, this register is updated to its reset values (that is, global reset, port reset, or COMINIT from the device).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				IPM				SPD				DET			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0 0000
11:8	IPM	Interface power management: Current interface state Read 0x0: Device not present or communication not established Read 0x1: Interface in ACTIVE state Read 0x2: Interface in PARTIAL power management state Read 0x6: Interface in SLUMBER power management state	R	0x0
7:4	SPD	Current interface speed: Negotiated interface communication speed Read 0x3: Generation 3 communication rate negotiated (6 Gbps) Read 0x2: Generation 2 communication rate negotiated (3 Gbps) Read 0x1: Generation 1 communication rate negotiated (1.5 Gbps) Read 0x0: Device not present or communication not established	R	0x0
3:0	DET	Device detection: Interface device detection and PHY state Read 0x0: No device detected and PHY communication not established Read 0x1: Device presence detected but PHY communication not established Read 0x3: Device presence detected and PHY communication established Read 0x4: PHY in offline mode as a result of the interface being disabled or running in a BIST loopback mode	R	0x0

Table 24-485. SATA_PxSCTL

Address Offset	0x0000 012C		
Physical Address	0x4A14 012C	**Instance**	DWC_ahsata
Description	Port SATA control Control of SATA interface capabilities. Writes to this register result in action taken by the port PHY interface. Reads from the register return the last value written to it. Reset on global reset. Wait for at least seven periods of the slower clock (OCP or parallel serdes clock) between writes, due to the internal clock domain crossing between the transport (OCP) and link (serdes I/F) layers.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PMP				SPM				IPM				SPD				DET			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:16	PMP	PM port: This field is not used by the AHCI.	R	0x0
15:12	SPM	Select power management: This field is not used by the AHCI.	R	0x0
11:8	IPM	Interface power management transitions allowed: Indicates which power states the HBA is allowed to transition to. If an interface power management state is disabled, the HBA is not allowed to initiate that state and the HBA must PMNAK_P any request from the device to enter that state. The two MSBs are always 2'b00 (not writable), as for all unreserved field values. 0x0: No interface power management state restrictions 0x1: Transitions to the PARTIAL state disabled 0x2: Transitions to the SLUMBER state disabled 0x3: Transitions to both PARTIAL and SLUMBER states disabled	RW	0x0
7:4	SPD	Speed allowed: Highest allowable speed of the interface The two MSBs are always 2'b00 (not writable), as for all unreserved field values. 0x0: No speed negotiation restrictions 0x1: Limit speed negotiation to generation 1 communication rate. 0x2: Limit speed negotiation to a rate not greater than generation 2 communication rate.	RW	0x0
3:0	DET	Device detection initialization: Controls the HBA device detection and interface initialization. Can be modified only when SATA_PxCMD.ST = 0. Must have a value of 0x0 when SATA_PxCMD.ST = 1. MSB is always 1'b0 (not writable), as for all unreserved field values. 0x0: No device detection or initialization action requested 0x1: Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications reinitialized. While this field is 1h, COMRESET is transmitted on the interface. Software must leave the DET field set to 1h for a minimum of 1 ms to ensure that a COMRESET is sent on the interface. 0x4: Disable the serial ATA interface and put PHY in offline mode.	RW	0x0

Table 24-486. SATA_PxSERR

Address Offset	0x0000 0130		
Physical Address	0x4A14 0130	**Instance**	DWC_ahsata
Description	Port SATA error Detected interface errors accumulated since the last time it cleared. When set, indicates that the corresponding error condition became true one or more times since the last time cleared. Write 1 to a bit to clear it. Cleared by global reset or port reset (COMRESET).		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>DIAG_X</td><td>DIAG_F</td><td>DIAG_T</td><td>DIAG_S</td><td>DIAG_H</td><td>DIAG_C</td><td>DIAG_D</td><td>DIAG_B</td><td>DIAG_W</td><td>DIAG_I</td><td>DIAG_N</td><td colspan="4">RESERVED</td><td>ERR_E</td><td>ERR_P</td><td>ERR_C</td><td>ERR_T</td><td colspan="6">RESERVED</td><td>ERR_M</td><td>ERR_I</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	DIAG_X	Exchanged: PHY COMINIT signal detected. Reflected in SATA_PxIS.PCS.	RW W1toClr	0
25	DIAG_F	Unknown FIS type: One or more FISes were received by the transport layer with good CRC, but had a type field that was not recognized/known and the length was = 64 bytes. Note: If the unknown FIS length exceeds 64 bytes, DIAG_F is not set and DIAG_T is set instead.	RW W1toClr	0
24	DIAG_T	Transport state transition error: Transport Layer protocol violation detected.	RW W1toClr	0
23	DIAG_S	Link sequence error: One or more Link state machine error conditions encountered, including device doing SYNC escape during FIS transmission.	RW W1toClr	0
22	DIAG_H	Handshake error: One or more R-ERRp received in response to frame transmission. May be the result of a CRC error detected by the device, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.	RW W1toClr	0
21	DIAG_C	CRC error: One ore more CRC errors detected by the link layer during FIS reception.	RW W1toClr	0
20	DIAG_D	Disparity error: Not used by AHCI, always 0.	R	0
19	DIAG_B	10bit-to-8bit decode error: Errors detected by the 10b8b decoder. Note: Set only when an error is detected on the received FIS data word. Not set when an error is detected on the primitive, regardless of whether it is inside or outside the FIS.	RW W1toClr	0
18	DIAG_W	Comm wake: Comm wake signal detected by the PHY.	RW W1toClr	0
17	DIAG_I	PHY internal error: Internal error detected by the PHY. Note: If the PHY does not support any errors, this bit is never set.	RW W1toClr	0
16	DIAG_N	PhyRdy change: Indicates that the PHY Ready signal changed state. Reflected in SATA_PxIS.PRCS.	RW W1toClr	0
15:12	RESERVED		R	0x0
11	ERR_E	Internal error: One or more errors detected on the master (DMA) or the slave (MMR access) interfaces.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
10	ERR_P	Protocol error: Any of the following conditions: - Transport state transition error (DIAG_T) - Link sequence error (DIAG_S) - RxFIFO overflow - Link bad end error (WTRM instead of EOF received)	RW W1toClr	0
9	ERR_C	Nonrecovered persistent communication error: PHY Ready signal is negated due to loss of communication with the device or problems with the interface, but not after transition from ACTIVE to PARTIAL or SLUMBER power management state.	RW W1toClr	0
8	ERR_T	Nonrecovered transient data integrity error: Any of the following conditions are set during data FIS transfer: - ERR_P (Protocol) - DIAG_C (CRC) - DIAG_H (Handshake) - ERR_C (PHY Ready negation)	RW W1toClr	0
7:2	RESERVED		R	0x00
1	ERR_M	Recovered communication error: PHY Ready condition is detected after interface initialization, but not after transition from PARTIAL or SLUMBER power management state to ACTIVE state.	RW W1toClr	0
0	ERR_I	Recovered data integrity error: Any of the following conditions are set during non-data FIS transfer: - ERR_P (Protocol) - DIAG_C (CRC) - DIAG_H (Handshake) - ERR_C (PHY Ready negation)	RW W1toClr	0

Table 24-487. SATA_PxSACT

Address Offset	0x0000 0134		
Physical Address	0x4A14 0134	**Instance**	DWC_ahsata
Description	Port SATA active (SActive): Indicates which command slots contain commands.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS																															

Bits	Field Name	Description	Type	Reset
31:0	DS	Device status: Field is bit-significant. Each bit corresponds to the TAG and command slot of a native queued command, where bit 0 corresponds to TAG 0 and command slot 0. Set by Software prior to issuing a native queued command for a particular command slot. Prior to writing SATA_PxCI[TAG] to 1, software sets DS[TAG] to 1 to indicate that a command with that TAG is outstanding. The device clears bits by sending a set device bits FIS to the port. The port clears bits in this field that are set to 1 in the SActive field of the set device bits FIS. The port only clears bits that correspond to native queued commands completed successfully. Write only when SATA_PxCMD.ST bit is set to 1. Cleared when SATA_PxCMD.ST is written from 1 to 0. Not cleared by a port reset (COMRESET) or a software reset.	RW	0x0000 0000

Table 24-488. SATA_PxCI

Address Offset	0x0000 0138		
Physical Address	0x4A14 0138	**Instance**	DWC_ahsata
Description	Port command issue: Indicates that a command is constructed and may be carried out.		

Table 24-488. SATA_PxCI (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CI																															

Bits	Field Name	Description	Type	Reset
31:0	CI	Commands issue: Field is bit-significant. Each bit corresponds to a command slot, where bit 0 corresponds to command slot 0. This field is set by software to indicate to the port that a command is built in system memory for a command slot and may be sent to the device. When the port receives a FIS that clears the BSY, DRQ, and ERR bits for the command, it clears the corresponding bit in this register for that command slot. Bits in this field can only be set to 1 by software when SATA_PxCMD.ST is set to 1. Also cleared when SATA_PxCMD.ST is written from 1 to 0 by software.	RW	0x0000 0000

Table 24-489. SATA_PxSNTF

Address Offset	0x0000 013C		
Physical Address	0x4A14 013C	**Instance**	DWC_ahsata
Description	Port SATA notification: Used to determine if asynchronous notification events have occurred for directly connected devices and devices connected to a PM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PMN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	PMN	PM notify: Indicates whether a particular device with the corresponding PM port number issued a set device bits FIS to the SATA controller Port with the notification bit set: - PM Port 0h sets bit 0. - PM Port 0h sets bit 1. - etc. Write 1 to a bit to clear it. Reset on global reset but not on port reset (COMRESET) or software reset.	RW W1toClr	0x0000

Table 24-490. SATA_PxDMACR

Address Offset	0x0000 0170		
Physical Address	0x4A14 0170	**Instance**	DWC_ahsata
Description	Port DMA control register. Not AHCI-standard. Writable only when SATA_PxCMD.ST = 0. Attempts to write a field value less than the minimum or more than the maximum cause the field to be set to the minimum or the maximum. Reset on global reset and port reset (COMRESET)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RXTS				TXTS			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x00 0000

Bits	Field Name	Description	Type	Reset
7:4	RXTS	Receive transaction size: DMA transaction size for receive operations (system bus write, device read). 0x0: 1 dword 0x1: 2 dwords 0x2: 4 dwords 0x3: 8 dwords 0x4: 16 dwords 0x5: 32 dwords 0x6: 64 dwords; maximum value for the 128-dword RX FIFO of this implementation.	RW	0x6
3:0	TXTS	Transmit transaction size: DMA transaction size for transmit operations (system bus read, device write). 0x0: 1 dword 0x1: 2 dwords 0x2: 4 dwords 0x3: 8 dwords 0x4: 16 dwords 0x5: 32 dwords; maximum value for this implementation.	RW	0x5

24.8.6.3 SATAMAC_wrapper Registers

24.8.6.3.1 SATAMAC_wrapper Register Summary

Table 24-491. SATAMAC_wrapper Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address SATAMAC_wrapper
SATA_SYSCONFIG	RW	32	0x0000 0000	0x4A14 1100
SATA_CDRLOCK	RW	32	0x0000 0004	0x4A14 1104

24.8.6.3.2 SATAMAC_wrapper Register Description

Table 24-492. SATA_SYSCONFIG

Address Offset	0x0000 0000		
Physical Address	0x4A14 1100	**Instance**	SATAMAC_wrapper
Description	This register controls the idle and standby modes of Highlander 08 modules.		
Type	RW		

31–17	16	15–6	5–4	3–2	1–0
RESERVED	OVERRIDE0	RESERVED	STANDBYMODE	IDLEMODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
16	OVERRIDE0	Override for clock stopping. Normally the functional clock can be stopped only if the link is put into PARTIAL or SLUMBER power state. However, if there is no device attached (such as in a removable media situation) or the device is not started, the user can stop the functional clocks but not be able to enter a low-power state. In this case, software can set the OVERRIDE bit to 1, removing the requirement for a low-power state WARNING: If there is a device attached, the OVERRIDE bit is used, and the functional clock is stopped when the link is not in a low-power state it ruins the link and causes undetermined behavior. A port reset or full SATASS reset might be required to recover. 0x0: Normal mode 0x1: Override mode	RW	0
15:6	RESERVED		R	0x000
5:4	STANDBYMODE	Configuration of the local initiator-state management mode. By definition, the initiator can generate a read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode: Local initiator is unconditionally placed in STANDBY state. Backup mode, for debug only 0x1: No-standby mode: local initiator is unconditionally placed out of STANDBY state. Backup mode, for debug only. 0x2: Smart-standby mode: Local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. The IP module does not generate (initiated-related) wakeup events. 0x3: Smart-Standby wakeup-capable mode: Local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. The IP module can generate (master related) wakeup events when in STANDBY state. Mode is relevant only if the appropriate IP module mwakeup output is implemented.	RW	0x2

Bits	Field Name	Description	Type	Reset
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, the target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: The local target IDLE state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the internal requirements of th e IP module. Backup mode, for debug only. 0x1: No-idle mode: The local target never enters IDLE state. Backup mode, for debug only. 0x2: Smart-idle mode: The local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the internal requirements of the IP module. IP module does not generate (IRQ- or DMA-request-related) wakeup events. 0x3: Smart-idle wakeup-capable mode: The local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the internal requirements of the IP module. IP module can generate (IRQ- or DMA-request-related) wakeup events when in IDLE state. Mode is only relevant if the appropriate IP module swakeup output(s) is (are) implemented.	RW	0x2
1:0	RESERVED		R	0x0

Table 24-493. SATA_CDRLOCK

Address Offset	0x0000 0004		
Physical Address	0x4A14 1104	**Instance**	SATAMAC_wrapper
Description	Programmable delay for CDR lock indication		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CDR_LOCK_DELAY											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0 0000
11:0	CDR_LOCK_DELAY(1)	CDR lock delay, in parallel (10-bit) serdes interface clock cycles. Parallel clock is 300 MHz (3.3 ns period) for SATA-3GT/s, 150 MHz (6.7 ns) for SATA-1.5GT/s. 0x0: No CDR lock delay 0x7D0: Default CDR lock delay: 13.33 us (1.5GT/s mode) or 6.67 (3 GT/s mode)	RW	0x7D0

(1) Under normal conditions, this bitfield must be kept at its default (power-on-reset) value.

24.9 PCIe Controller

This section describes the features and functions of the device Peripheral Component Interconnect Express (PCIe) Controller which provides a high-speed glueless serial interconnect to peripherals utilizing high bandwidth applications.

24.9.1 PCIe Controller Subsystem Overview

The Peripheral Component Interconnect Express (PCIe) module is a multi-lane I/O interconnect that provides low pin-count, high reliability, and high-speed data transfer at rates of up to 5.0 Gbps per lane, per direction, for serial links on backplanes and printed wiring boards. It is a 3-rd Generation I/O Interconnect technology succeeding PCI and ISA bus that is designed to be used as a general-purpose serial I/O interconnect. It is also used as a bridge to other interconnects like SATA, USB2/3.0, GbE MAC, and so forth.

The device instantiates two PCIe subsystems (PCIe_SS1 and PCIe_SS2) . The PCIe controller is capable to operate either in Root Complex (RC) or in End Point (EP) PCIe mode. The device PCIe_SS1 controller supports up to two 16-bit data lanes on its PIPE port. The device PCIe_SS2 controller supports only one 16-bit data lane on its PIPE port.

When the PCIe_SS1 controller PIPE port is configured to operate in a single-lane mode, it operates on a single pair of PCIe PHY serializer and deserializer - PCIe1_PHY_TX/PCIe1_PHY_RX. When PCIe_SS1 PIPE is configured to operate in dual-lane mode, it operates on two pairs of PCIe PHY serializer and deserializer - PCIe1_PHY_TX/PCIe1_PHY_RX and PCIe2_PHY_TX/PCIe2_PHY_RX, respectively. The single-lane PCIe_SS2 controller PIPE port (if enabled) can operate only on the PCIe2_PHY_TX/PCIe2_PHY_RX pair. Hereby, if PCIe_SS2 controller is used, the PCIe_SS1 can operate only in a single-lane mode on the PCIe1_PHY_TX/PCIe1_PHY_RX. In addition, PCIe PHY subsystem encompasses a PCIe PCS (physical coding sublayer), a PCIe power management logic, APLL, a DPLL reference clock generator and an APLL clock low-jitter buffer. For more details on PCIe PHY subsystem, please refer to Section 26.4, *PCIe Shared Phy Subsystem*.

- The PCIe Controller implements the transport and link layers of the PCIe interface protocol.
- PCIe PCS (a physical coding sublayer component) converts a 8-bit portion of parallel data over a PCIe lane to a 10-bit parallel data to adapt the process of serialization and deserialization in the TX/RX PHYs to various requirements. At the same time it transforms the transmission rate to maintain the PCIe Gen2 bandwidth (5 Gbps) on both sides (PCIe controller and PHY).
- A multiplexer logic which adds flexibility to connect a PCIe controller hardware mapped PCS logic output to a single (for the single-lane PCIe_SS2 controller) or to a couple (for the 2-lane PCIe_SS1 controller) of PHY ports at a time
- Physical layer (PHY) serializer/deserializer components with associated power control logic, building the so called PMA (physical media attachment) part of the PCIe_PHY transceiver, as follows:
  – PCIe physical port 0 associated serializer (TX) - PCIe1_PHY_TX and deserializer (RX) - PCIe1_PHY_RX
  – PCIe physical port 1 associated serializer (TX) - PCIe2_PHY_TX and deserializer (RX) - PCIe2_PHY_RX
- DPLL_PCIe_REF is a DPLL clock source, controlled from the device PRCM, that provides a 100-MHz clock to the PCIe PHY serializer/deserializer components reference clock inputs.
- Both the PCIe_SS1 and PCIe_SS2 share the same APLL (APLLPCIe) which by default multiplies the DPLL_PCIe_REF (typically 100 MHz or 20 MHz) clock to 2.5 GHz.
- The APLLPCIe low-jitter buffer (ACSPCIE) and additional logic takes care to provide the PCIe APLL reference input clock.

Figure 24-159 shows an overview of a device PCIe subsystem with its integrated components.

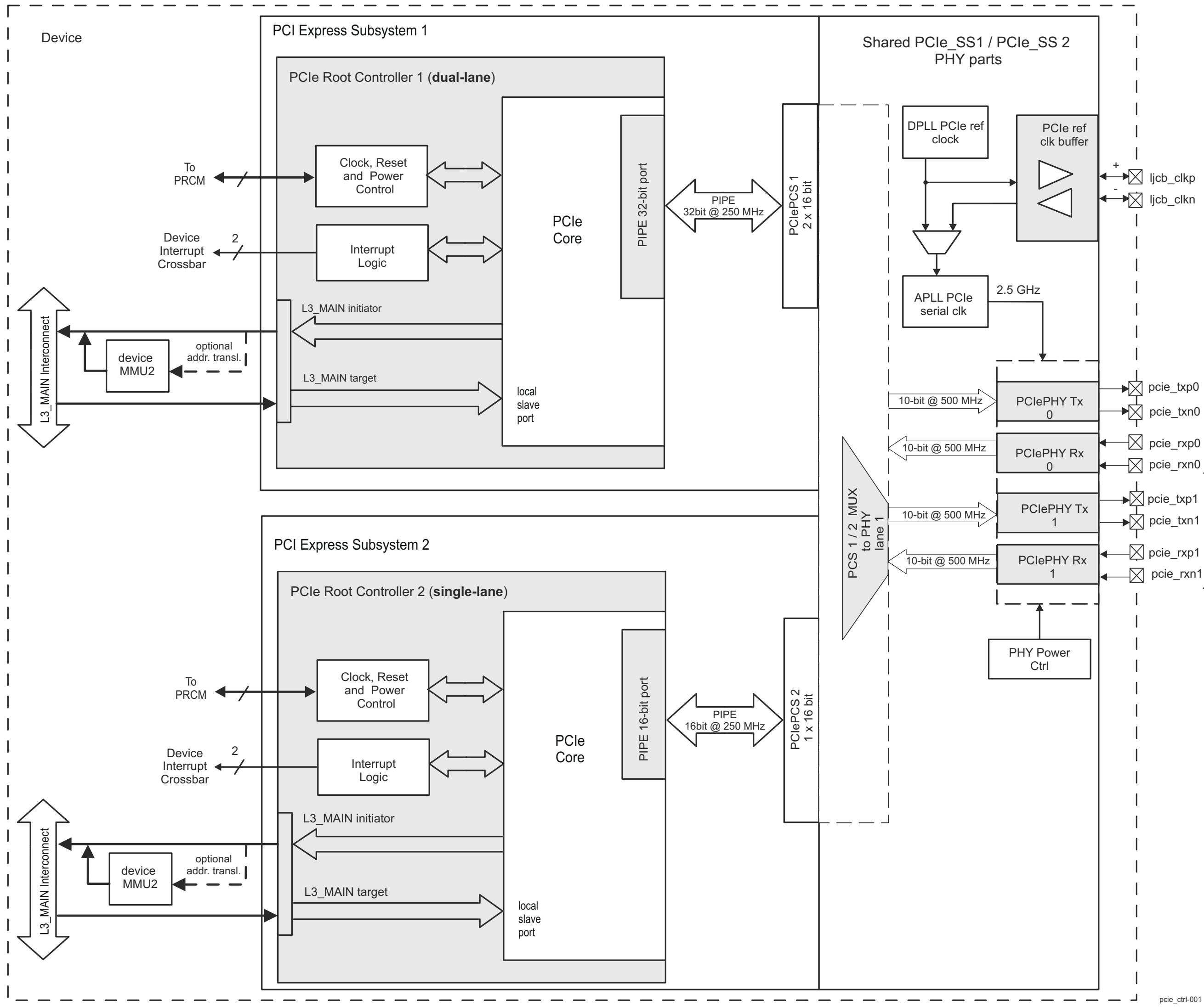


Figure 24-159. PCIe Controller Subsystem Overview

24.9.1.1 PCIe Controllers Key Features

This section describes the features supplied by the PCIe controller modules (including PCIe at device level). The PCIe_SS1 and PCIe_SS2 controllers comply with the following standards:

- PCI Local Bus Specification revision 3.0
- PCI Express Base 3.0 Specification, revision 1.0.

At system level the device supports PCI express interface in the following configurations:

- Each PCIe subsystem controller has support for PCIe Gen2 mode (5.0 Gbps per lane) and Gen1 mode (2.5 Gbps per lane).
- One PCIe (PCIe_SS1) operates as Gen2 2-lanes supporting in either root-complex (RC) or end-point EP.
- Two PCIe (PCIe_SS1 and PCIe_SS2) operates Gen2 1-lane supporting either RC or EP with the possibility of one operating in Gen1 and one in Gen2.
- PCIe_SS1 can be configured to operate in either 2-Lane (dual lane) or 1-Lane (single lane) mode, as follows:
  - Single Lane - lane 0 mapped to the PCIe port 0 of the device
  - Flexible dual lane configuration - lanes 0 and 1 can be swapped on the two PCIe ports
- PCIe_SS2 can only operate in 1-Lane mode, as follows:

  – Single Lane - lane 0 mapped to the device PCIe port 1

When PCie_SS1 is configured to operate in dual-lane mode, PCIe_SS2 is not available.

The main features of a device PCIe controller are:

- 16-bit operation at 250 MHz on PIPE interface (per 16-bit lane)
- One master port on the L3_MAIN supporting 32-bit address and 64-bit data bus.
- PCIe_SS1/PCIe_SS2 master port dedicated MMU (device MMU2) on L3_MAIN path, to which PCIe traffic can be optionally mapped.
- One slave port on the L3_MAIN supporting 29-bit address and 64-bit data bus.
- Maximum outbound payload size of 64 Bytes (the L3 Interconnect PCIe1/2 target ports split bursts of size >64 Bytes to the into multiple 64 Byte bursts)
- Maximum inbound payload size of 256 Bytes (internally converted to 128 Byte - bursts)
- No remote read request size limit: implicit support for 4 KiB-size and greater
- Support of EP legacy mode
- Support of inbound I/O accesses in EP legacy mode
- PIPE interface features fixed-width (16-bit data per lane) and dynamic frequency to switch between PCIe Gen1 and Gen2.
- Ultra-low transmit and receive latency
- Automatic Lane reversal as specified in the PCI Express Base 3.0 Specification, revision 1.0 (transmit and receive)
- Polarity inversion on receive
- Single Virtual Channel (VC0) and Single Traffic Class (TC0)
- Single Function in End point mode
- Automatic credit management
- ECRC generation and checking
- All PCI Device Power Management D-states with the exception of $D3_{cold}$/L2 state
- PCI Express Active State Power Management (ASPM) state L0s and L1 (with exceptions)
- PCI Express Link Power Management states except for L2 state
- PCI Express Advanced Error Reporting (AER)
- PCI Express messages for both transmit and receive
- Filtering for Posted, Non-Posted, and Completion traffic
- Configurable BAR filtering, I/O filtering, configuration filtering and completion lookup/timeout
- Access to configuration space registers and external application memory mapped registers through ECAM mechanism.
- Legacy PCI Interrupts reception (RC) and generation (EP)
- 2× hardware interrupts per PCIe_SS1and PCIe_SS2 controller mapped via the device Interrupt Crossbar (IRQ_CROSSBAR) to multiple device host (MPU, DSP, and so forth) interrupt controllers in the device
- MSIs generation and reception
- PCIe_PHY Loopback in RC mode

24.9.2 PCIe Controller Environment

The below Table 24-494 shows the device integrated PCI Express subsystem interface signals to external PCIe devices.

Table 24-494. PCIe_SS I/O Signals

Device Level Signal Name	I/O(1)	Description	Reset Value
pcie_txp0	O	TX output of the PCIe port 0 PHY differential transmission line (positive by default) Section 24.9.3	0
pcie_txn0	O	TX output of the PCIe port 0 PHY differential transmission line (negative by default) Section 24.9.3	0
pcie_rxp0	I	RX input of the PCIe port 0 PHY differential reception line (positive by default) Figure 24-160	HiZ
pcie_rxn0	I	RX input of the PCIe port 0 PHY differential reception line (negative by default) Figure 24-160	HiZ
pcie_txp1	O	TX output of the PCIe port 0 PHY differential transmission line (positive by default) Section 24.9.3	0
pcie_txn1	O	TX output of the PCIe port 0 PHY differential transmission line (negative by default) Section 24.9.3	0
pcie_rxp1	I	RX input of the PCIe port 1 PHY differential reception line (positive by default) Figure 24-160	HiZ
pcie_rxn1	I	RX input of the PCIe port 1 PHY differential reception line (negative by default) Figure 24-160	HiZ
ljcb_clkp	I/O	Differential clock positive input or output	HiZ
ljcb_clkn	I/O	Differential clock negative input or output	HiZ

(1) I = Input; O = Output; I/O = Bidirectional

Table 24-495. PCIe_SS Port Configuration

PCIE_B1C0_MODE_SEL(1)	PCIE_B0_B1_TSYNCEN(1)	Port 0	Port 1
0 (C0) (default)	0	PCS1 (MAC B) lane 0	PCS2 (MAC C) lane 0
1 (B1)	0	PCS1 (MAC B) lane 0	-
1 (B1)	1	PCS1 (MAC B) lane 0	PCS1 (MAC B) lane 1

(1) See CTRL_CORE_PCIE_CONTROL register in *CTRL_MODULE_CORE Registers*

For more information on the interface between PCIe_SS controller and PCIe_PHY, see Section 26.4.4.1, *PCIe Shared PHY Subsystem Block Diagram*, and Section 26.4.4, *PCIe Shared PHY Subsystem Functional Descriptions* in Section 26.4, *PCIe Shared PHY Subsystem*.

24.9.3 PCIe Controllers Integration

This section describes the PCIe controllers integration in the device, including information about clocks, resets, and hardware requests. Figure 24-160 shows the PCIe controllers integration.

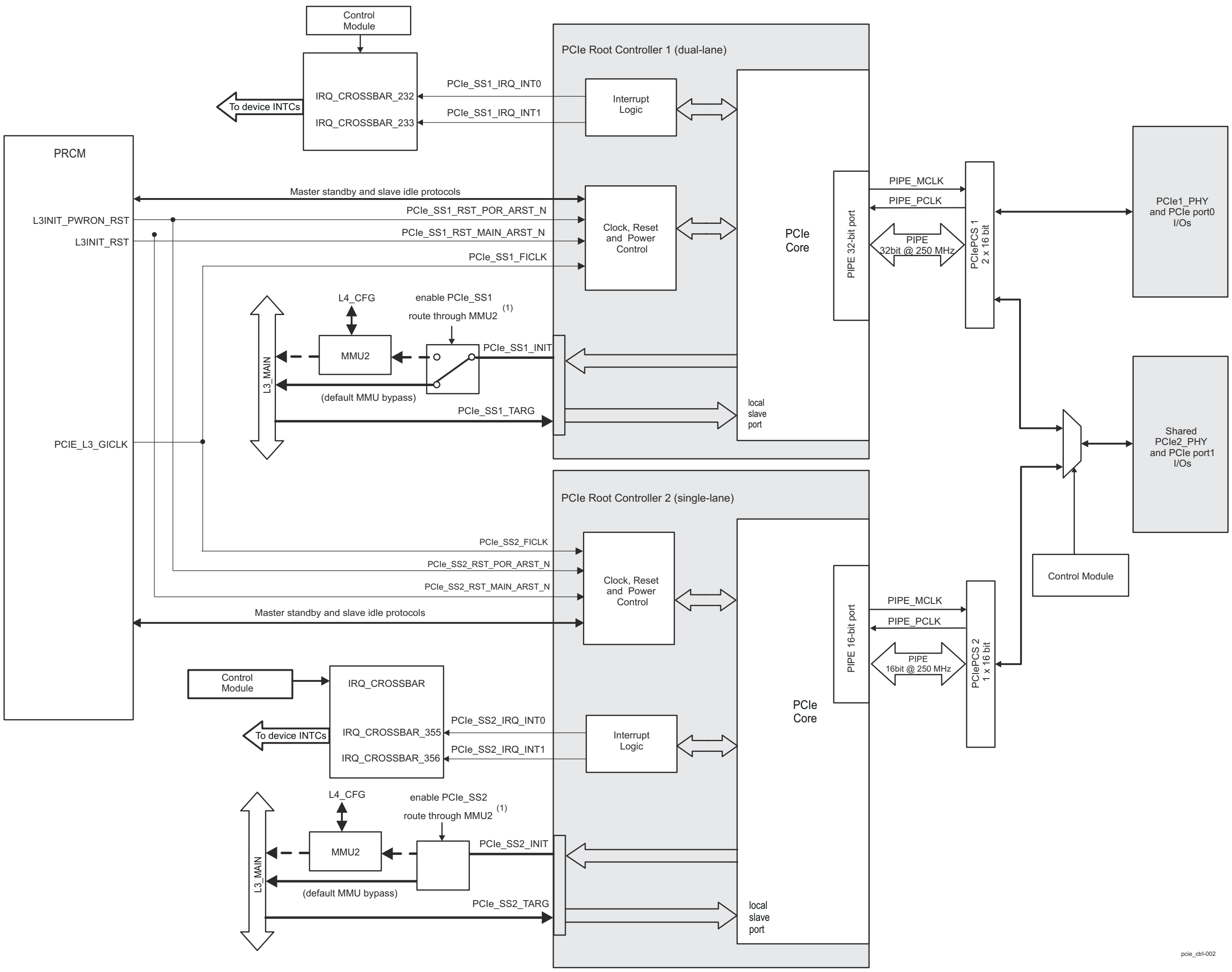


Figure 24-160. PCIe Controllers Integration

Note

By default, the traffic from the PCIe_SS1 and PCIe_SS2 controller master (initiator) port - PCIe_SS1_INIT/PCIe_SS2_INIT to the L3_MAIN interconnect, bypasses the device MMU2. Mapping the PCIe traffic from the PCIe master port to MMU2 on the L3_MAIN path is enabled via a bit in the device Core Control Module. For more details, refer to the Section 24.9.4.3.1.1.

PCIe controller integration includes these features:

- A single interface/functional clock (PCIe_SS_FICLK, shared between the master and the slave interfaces)
- A couple of functional clocks:
  - A PCIE controller clock input which receives the PIPE_PCLK generated by the assocaited PCIe PHY PCS logic
  - A PCIE controller clock output PIPE_MCLK which mirrors PIPE_PCLK back to the PCIe PHY PCS component

- Two hardware non-retention resets - PCIe_SS_RST_POR_ARST_N and PCIe_SS_RST_MAIN_ARST_N by PRCM
- Master standby and slave idle protocols with the power, reset, and clock management (PRCM) module
- Two interrupt request lines (per each PCIe controller):
  - PCIe_SS1_IRQ_INT0/PCIe_SS2_IRQ_INT0 - a main interrupt line
  - PCIe_SS1_IRQ_INT1/PCIe_SS2_IRQ_INT1 - a MSI interrupt line
- No DMA requests generation to surrounding modules
- One 64-bit data/32-bit address master port on the L3_MAIN interconnect (with an option to pass the PCIe initiator traffic to L3_MAIN through device MMU2). For details, refer to the Section 24.9.4.3.
- One 64-bit data/29-bit address slave port on the L3_MAIN interconnect (No MMU included on the slave address path). For details, refer to the Section 24.9.4.3.

For more details on the MMU2 integration in the device, refer to the Section 20.2, *MMU Integration* of the Chapter 20, *Memory Management Units*.

Note

For more information about the slave idle protocol and the wakeup request, see *Module-Level Clock Management* in *Power, Reset, and Clock Management*.

Table 24-496 through Table 24-498 summarize the integration of the module in the device.

Table 24-496. PCIe Controllers Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
PCIe_SS1	PD_L3INIT	L3_MAIN
PCIe_SS2	PD_L3INIT	L3_MAIN

Table 24-497. PCIe Controllers Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
PCIe_SS1	PCIe_SS1_FICLK	PCIE_L3_GICLK	PRCM	PCIe_SS1 root controller interface and functional clock (shared between slave and master ports)
PCIe_SS2	PCIe_SS2_FICLK	PCIE_L3_GICLK	PRCM	PCIe_SS2 root controller interface and functional clock (shared between slave and master ports)
Resets				
PCIe_SS1	PCIe_SS1_RST_POR_ARST_N	L3INIT_PWRON_RST	PRCM	A nonretention hardware power-on reset to the PCIe_SS1 controller
	PCIe_SS1_RST_MAIN_ARST_N	L3INIT_RST	PRCM	A nonretention hardware main reset to the PCIe_SS1 controller
PCIe_SS2	PCIe_SS2_RST_POR_ARST_N	L3INIT_PWRON_RST	PRCM	A non-retention hardware power-on reset to the PCIe_SS2 controller
	PCIe_SS2_RST_MAIN_ARST_N	L3INIT_RST	PRCM	A non-retention hardware main reset to the PCIe_SS2 controller

Table 24-498. PCIe Controllers Hardware Requests

Interrupt Requests				
Module Instance	**IRQ Source Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
PCIe_SS1	PCIe_SS1_IRQ_INT0	IRQ_CROSSBAR_232	N/A	PCIe_SS1 controller main interrupt request.

Table 24-498. PCIe Controllers Hardware Requests (continued)

Interrupt Requests				
Module Instance	**IRQ Source Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
	PCIe_SS1_IRQ_INT1	IRQ_CROSSBAR_233	N/A	PCIe_SS1 controller MSI interrupt request.
PCIe_SS2	PCIe_SS2_IRQ_INT0	IRQ_CROSSBAR_355	N/A	PCIe_SS2 controller main interrupt request
	PCIe_SS2_IRQ_INT1	IRQ_CROSSBAR_356	N/A	PCIe_SS2 controller MSI interrupt request.

Note

The **Default Mapping** column in Table 24-498, *PCIe Controller Hardware Requests* shows the default mapping of the IRQ sources listed in column **IRQ Source Name** to a certain interrupt line of one of the device interrupt controllers. These IRQ sources can also be mapped to other interrupt lines of each device interrupt controller through the IRQ_CROSSBAR module. For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*. For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

No DMA requests are generated by the PCIe controller to the surrounding modules.

For more details on interrupt request management at the PCIe subsystem local level, refer to the Section 24.9.4.6, *PCIe Controller Interrupt Requests*.

24.9.4 PCIe SS Controller Functional Description

24.9.4.1 PCIe Controller Functional Block Diagram

This section describes how the overall PCI express functionality is implemented by the current controller, both as EP and as RC, and how each type of traffic is transmitted and received.

The PCI express interface allows memory-mapped read and write transactions to be performed across different partners of the same PCIe fabric. Locally the device local PCIe system is composed of the PCIe_SS controller itself, device local hosts (such as MPU, DSP, and so forth), device DMAs (EDMA, and so forth) and device system memory (EMIFs SDRAM), connected to the PCIe controller via L3_MAIN interconnect.

Figure 24-161 summarizes the functional components of the PCIe_SS controller, as well as the connectivity between PCIe_SS and other components (MMU2, IRQ_CROSBBAR, PRCM, and so forth) within the device.

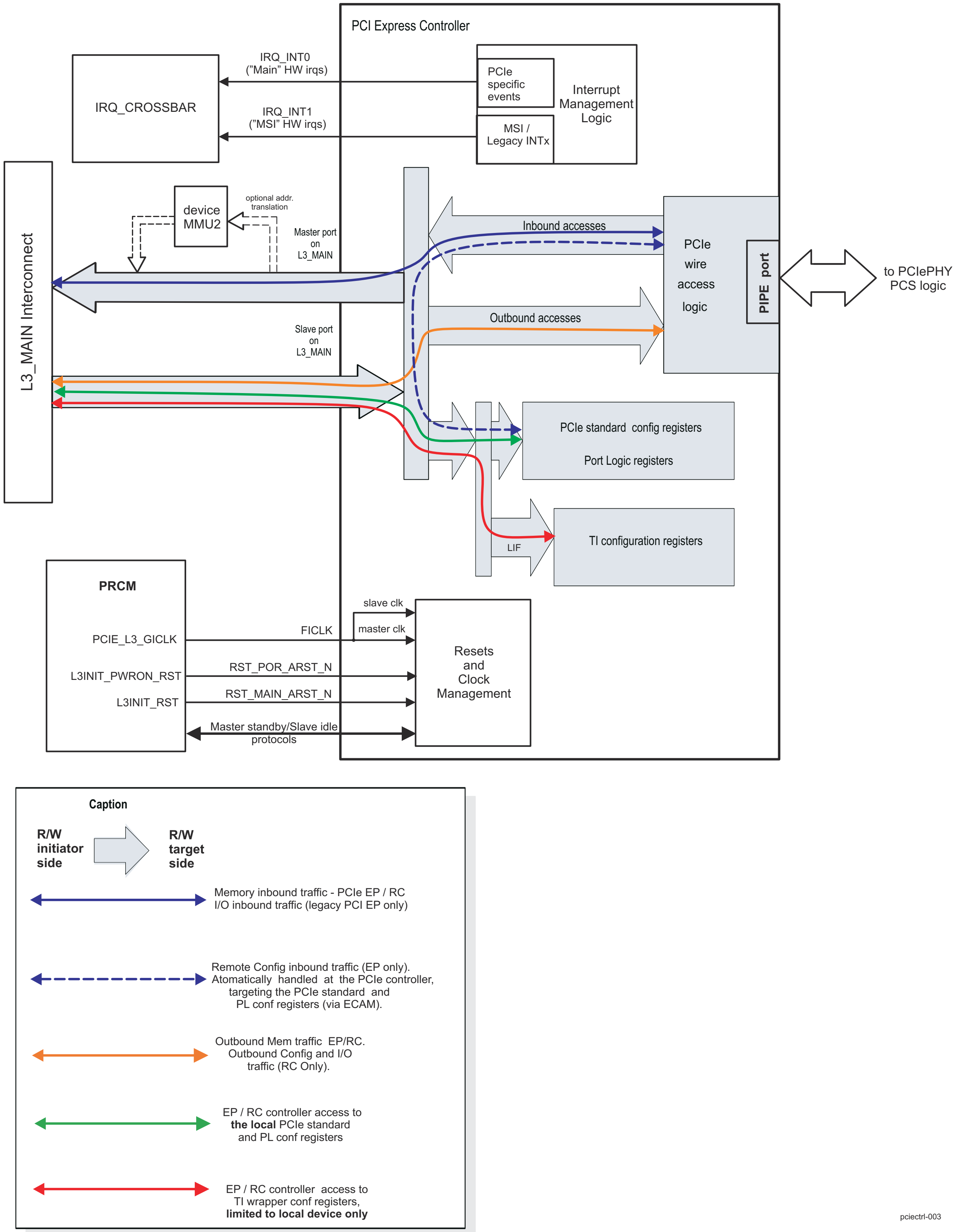


Figure 24-161. PCIe Controller Functional Block Diagram

24.9.4.2 PCIe Traffics

Most of the accesses (Memory-space) are typically flowing straight through the controller, from PCIe bus to L3_MAIN or the other way. However, some transactions are accesses to the PCIe device's configuration registers:

- The source of those accesses is either the local running application, through the PCIe slave port, or a distant PCIe device on the bus, as an inbound transaction from the PCIe wire. Access may or may not be possible depending on the mode of operation (see also transfer descriptions on the Figure 24-161).
- The destination of those accesses is either the PCIe controller core, for the PCIe-standard configuration registers and the non-standard port logic (PL) registers, or the TI wrapper configuration registers accessed over the local interface (LIF).

Figure 24-161 also gives a simplified view of the accesses allowed by device PCIe controller.

On the right hand side of the diagram is the PCIe_PHY (PCS+PMA), itself connected to the PCIe wires. The same structure is expected in the link partner on the other side of the link. Each partner can initiate and complete transactions on the link, using a packet-based protocol. Each lane is a full-duplex serial channel, that is, packets can be exchanged in both directions simultaneously. When more than one lane is used, each packet is transmitted in parallel over all active lanes.

The bidirectional accesses, initiated from components on the PCIe bus (fabric) to/from the device local PCIe subsystem are identified as *Inbound traffic*, while the bidirectional accesses, initiated by device local PCIe system to/from the PCIe bus are identified as *Outbound traffic*.

Note that there are two ways to transmit data from the local PCIe system to a distant PCIe partner:

- **Outbound write accesses**, locally initiated by, for example, a local DMA accessing controller through the PCIe slave port, connected on L3_MAIN, to push data towards a distant PCIe partner.
- **Inbound read accesses**, remotely initiated by, for example, a distant DMA though the PCIe wire, and issued on the PCIe master port, connected on L3_MAIN, to fetch data from the local memory.

Likewise, there are two ways for the local PCIe system to receive data from a distant PCIe partner:

- **Outbound read accesses**, locally initiated by, for example, a local DMA accessing the PCIe slave port, connected on L3_MAIN, to fetch data from a distant PCIe partner.
- **Inbound write accesses**, remotely initiated by, for example, a distant DMA though the PCIe wire, and issued on the PCIe master port towards the local system memory.

Note

Inbound PCIe I/O transfers are supported by the PCIe controller only in EP mode (typically for legacy PCI - EPs).

24.9.4.3 PCIe Controller Ports on L3_MAIN Interconnect

The PCIe controller to L3_MAIN connectivity is illustrated, on the left-hand side of the Figure 24-161. Read/write accesses from external PCIe components to the device local system memory on L3_MAIN are performed through the PCIe_SS master (initiator) port - PCIe_SS_INIT. The addresses issued on this port are by default directly routed to the L3_MAIN, but they can be optionally routed to and further translated by the device memory management unit - MMU2 before they reach the L3_MAIN target. For more details on MMU port, refer to Section 24.9.4.3.1.1.

On the other hand, the device local hosts (MPU, DSP, and so forth) initiate accesses to the PCIe controller local registers and PCIe bus remote devices via a PCIe_SS slave (target) port - PCIe_SS_TARG on the L3_MAIN. No MMU is included on the slave path. The slave interface has two uses: manage the local PCIe controller, and carry outbound PCIe traffic, that is initiated locally by PCIe hosts (MPU, DSP, DMAs, and so forth) within the device.

24.9.4.3.1 PCIe Controller Master Port

The PCIe controller master port (using AXI protocol with an adapter to the device L3_MAIN interconnect) has the following key fetures:

- 64-bit data, 32-bit address bus width
- Up to 16 simultaneous outstanding transactions, on up to 16 different IDs (4-bit ID port) total for read and write ports together, as follows:
  – Posted incoming PCIe transactions are all mapped to AXI ID 0, which ensures that they are executed in order. This includes all high-bandwidth writes, which are memory-type - posted.
  – Non-posted incoming PCIe transactions are mapped to AXI ID 1 and above. This includes the high-bandwidth memory-type reads.
- uses the same clock source as the PCIe slave port, but potentially on a different divider ratio
- PCIe master port, does not use a disconnect interface, because unlike the PCIe slave port, it is expected to be always active when the PCIe controller has to initiate a transaction over the device L3_MAIN.
- The PCIe controller maximum inbound payload size is 256 Bytes. The PCIe master port burst maximum length is 16 words (16 x 64-bit data words = 128 Bytes per burst). Hence a 256 Byte inbound data payload is converted by the PCIe master port to 2 max-sized bursts (128 Byte each) towards the device L3_MAIN interconnect PCIe slave port which is 128 Byte-burst compatible.
- Only incremental bursts (INCR) are supported by the PCIe controller master port.
- Non-aligned bursts can be generated - bursts that are not aligned with their own size: a burst-aligned portion of 2^N-bytes aligned burst starts on a byte address multiple of 2^N, that is a byte address with the N LSBs at '0').

The PCIe memory space supports a 64-bit address mode but in the device PCIe controller, the AXI master port address size is restricted to 32 bits.

Note

A PCIe controller remains fully functional, and able to receive transactions from , for example, anywhere within the 64-bit PCIe memory space, as long as they are correctly remapped by the inbound address translation unit (ATU) to a 32-bit L3_MAIN space address (device L3_MAIN 4 GiB - memory space). The limitation is that the range addressable by the PCIe master is reduced, and that regions larger than 2^{32} bytes (4GiB) total, cannot be mapped to the AXI master port on L3_MAIN.

24.9.4.3.1.1 PCIe Controller Master Port to MMU Routing

A standalone MMU2 module is included at the system level, primarily for use by the PCIe_SS1 and PCIe_SS2 controller. This provides several benefits including protection of the MPU host memory regions from corruption by PCIe_SS1 and PCIe_SS2 accidental accesses.

PCIe controller initiated (by remote side) traffic can be optionally routed through the MMU2 as controlled by a Control Core Module register bit. The PCIe_SS1 and PCIe_SS2 routing allows these sub-systems to be used to perform transfers.

Note

By default, the CTRL_MODULE_CORE instance located CTRL_CORE_CONTROL_IO_1[20] MMU2_DISABLE bit is set to 0b0 and the MMU2 is functionally enabled. Despite this, the PCIe_SS1 and PCIe_SS2 controller traffic still bypasses this MMU.
In order, for PCIe_SS1 and PCIe_SS2 master traffic
to be routed through the MMU2, the CTRL_MODULE_CORE
bits: CTRL_CORE_SMA_software_7[13] PCIE_SS1_MMU_ROUTE_ENABLE and
CTRL_CORE_SMA_software_7[12] PCIE_SS2_MMU_ROUTE_ENABLE, must be set to 0b1 by user software.

Accesses are directed to the MMU2 by the L3_MAIN switch fabric through the use of a 33-rd address bit. For the PCIe_SS1 and PCIe_SS2 master port, the value of the 33rd bit is actually defined by the above mentioned control module CTRL_CORE_SMA_software_7 register bits. For descriptions of the CTRL_CORE_SMA_software_7 register bitfields related to MMU2 route controls, refer to the *Control Module Register Manual*, in the chapter, *Control Module*.

For more information on device MMU2 functionality and register settings, refer to the *MMU Functional Description* and *MMU Register Manual*, in the chapter, *Memory Management Units*, respectively.

24.9.4.3.2 PCIe Controller Slave Port

The PCIe controller slave port gives access to two internal targets: the local target, to manage the local controller, and the outbound window, which forwards accesses onto the PCIe wire.

The PCIe slave port (using AXI protocol with an adapter to the device L3_MAIN interconnect) has the following key features:

- **64-bit data, 29-bit address** bus width
- Up to 16 simultaneous outstanding transactions, up to 16 different IDs (4-bit ID port).
- Uses the same clock source as the PCIe controller master port, but potentially on a different divider ratio.
- Supports L3_MAIN disconnect protocol
- Exclusive and locked accesses are not supported for read and write port

PCIe Controller slave port purposes are:

- **Configuration and management of the PCIe controller:**
  - This low-bandwidth activity accesses the device PCIe controller local configuration/status registers, the Port Logic registers and the PCIe controller wrapper TI-specific registers.
  - The L3_MAIN base address corresponding to the PCIe_SS1 controller local configuration/status registers (DIF access targets) is 0x5100_0000. For the PCIe_SS2 controller local configuration/status registers (DIF access targets), the L3_MAIN base address is 0x5180_0000.
  - Bursts are not supported when accessing these registers.
- **Outbound PCIe traffic**:
  - This activity can consume the entire bandwidth available on the PCIe wires (up to 1 Gbyte/s in each direction for 2-lane, Gen2 operation), in concurrence with the AXI master ports' operation (that carries inbound PCIe traffic).
  - PCIe controller maximum outbound payload size is 64 Bytes (the L3 Interconnect PCIe1/2 target ports split bursts of size > 64 Bytes to the into multiple 64 Byte bursts). Only bursts of incremental type (INCR) are supported on the PCIe controller slave port.
  - Non-aligned bursts can be generated.
  - All outbound traffic activities are mapped within a device L3_MAIN space region which is 256 MiB in size.
  - For the PCIe_SS1 controller the PCIe outbound window is mapped within range 0x2000_0000 - 0x2FFF_FFFF of the device L3_MAIN memory space. For the PCIe_SS2 controller the PCIe outbound window is mapped within range 0x3000_0000 - 0x3FFF_FFFF (256 MiB) of the device L3_MAIN memory space.

Note

The PCIe controller remains fully functional, and able to send transactions to, for example, anywhere within the 64-bit PCIe memory space, with the appropriate remapping of the 28-bit address by the outbound address translation unit (iATU). The limitation is that the total size of addressed PCIe regions (in config, memory, IO spaces) must be less than 2^{28} bytes.

24.9.4.3.3

Note

Because only incremental burst mode is supported, the PCIe addresses can not be in a cacheable memory space. Subsequently, cache coherence protocol is not involved, and the PCIe controller is not expected to receive coherent PCIe TLPs (NS=0). For safety purpose, the PCIe controller coherent feature must be permanently disabled (and this applies for the device by default) by keeping PCIECTRL_TI_CONF_SYSCONFIG[16] MCOHERENT_EN at value 0b0. In that case, inbound coherent PCIe TLPs (that means with NS=0) proceed normally, but coherence will not be guaranteed by the device.

24.9.4.4 PCIe Controller Reset Management

The PCIe controller has two hardware reset inputs - a hardware **main** and **fundamental** resets which are asynchronous and active low.

In addition, a number of internal reset conditions can cause some sections of the controller to reset, depending on the power transitions and configuration. All reset conditions and categories are described below.

24.9.4.4.1 PCIe Reset Types and Stickiness

Three types of reset conditions are defined by the PCI standard. Each reset condition listed below falls into one of these categories. The same reset condition can take different types depending on the usecase:
- Cold reset condition: upon device power-up.
- Warm reset condition: does not follow a device power-up.
- Hot reset condition: transmitted over the PCIe link.

A reset including the physical layer (PCIe_PHY) is **called a fundamental reset**. According to the PCI Express Base 3.0 Specification, revision 1.0, it clears the programming registers non-sticky bits, and may clear or may not clear the sticky bits as well, depending on a Vaux power supply. If Vaux integrated in a PCIe subsystem, it is enabled via PM specific sticky bits - PM_CSR[8] PME_EN and DEV_CAS[10] AUXPM_EN register bits.

CAUTION

No Vaux power supply source is defined in hardware for the device integrated PCIe_SS. Subsequently, all PCIe bits defined by the PCIe standard or device specific implementation as "sticky", are reset along with the "non-sticky" bits upon a PCIe fundamental reset assertion. Because it is mandatory to communicate the permanent Vaux absence to the device PCIe EP/RC controller, the EP/RC DEV_CAS[10] AUXPM_EN (or PM_CSR[8] PM_EN) and PCIECTRL_TI_CONF_PM_CTRL [11] AUX_PWR_DET bits have to be always software written ‘0’ during initialization.

A function-level reset (FLR) impacts only one of the EP device’s functions.

Note

The device PCIe controller supports only a single function, so it does not support FLR.

24.9.4.4.2 PCIe Reset Conditions

24.9.4.4.2.1 PCIe Main Reset

Main hardware reset: Assertion of the reset signal on the main hardware reset input of the PCIe subsystem unconditionally brings the entire PCIe controller back to its default state. The main HW reset impacts both non-sticky and sticky bits of the PCIe controller.

Note

The PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE (RC/EP) bitfield and the remaining TI wrapper configuration registers (PCIe_SS_TI_CONF) are sensitive to main hardware reset only.

24.9.4.4.2.1.1 PCIe Subsystem Cold Main Reset Source

The cold main reset has as a source - the device PRCM L3INIT_PWRON_RST power-on reset. For more information on the PCIe controller hardware reset inputs see Section 24.9.3, *PCIe Controller Integration*).

The L3INIT_PWRON_RST assertion by PRCM resets all PCIe controller registers including the TI wrapper configuration registers (note that the PCIe device type in the PCIECTRL_TI_CONF_DEVICE_TYPE register is reset by default to "Root Complex").

For more information on the PRCM.L3INIT_PWRON_RST hardware reset, refer to *Reset Domains*, in *Power, Reset, and Clock Management*.

24.9.4.4.2.1.2 PCIe Subsystem Warm Main Reset Sources

The warm main reset of the PCIe controller is sourced by the device PRCM L3INIT_RST output. For more information on the PCIe controller hardware reset inputs see Section 24.9.3, *PCIe Controller Integration*). For more information on the PRCM.L3INIT_RST hardware reset, refer to *Reset Domains*, in *Power, Reset, and Clock Management*.

The L3INIT_RST main reset is tiggered from different warm reset sources within the PD_L3INIT power domain (global warm software reset, MPU watchdog reset, and so forth).

Software triggered main reset to PCIe controller : To the above group of PRCM warm main reset sources, there is also a software-triggered reset which can be asserted locally and independently to each of the PCIe_SS1 and PCIe_SS2 in the PD_L3INIT domain. The main software reset is triggered via assertion of bit:

- RM_PCIESS_RSTCTRL[0] RST_LOCAL_PCIE1 for the device PCIe_SS1 controller
- RM_PCIESS_RSTCTRL[1] RST_LOCAL_PCIE2 for the device PCIe_SS2 controller

The software (warm) reset completion triggered through device PRCM is flagged in the *Write-one to Clear* PRCM register:

- RM_PCIESS_RSTST[0] RST_LOCAL_PCIE1 for the device PCIe_SS1 controller
- RM_PCIESS_RSTST[1] RST_LOCAL_PCIE2 for the device PCIe_SS2 controller

The user software running on PCIe controller is supposed to clear the reset status bit upon reset completion event via writing 0b1 to RM_PCIESS_RSTST[0] RST_LOCAL_PCIE1 and RM_PCIESS_RSTST[1] RST_LOCAL_PCIE2 bit in PRCM.

For more details on above PCIe controller software reset bits, refer to the *PRCM Register Manual*, in the *Power, Reset and Clock Management*

Note

The warm main reset (icluding PCIe subsystem software reset) from PRCM impacts all PCIe core register bits (sticky and non-sticky) and the PCIe TI wrapper configuration registers (PCIe_SS_TI_CONF). Note that the PCIe device type in the PCIECTRL_TI_CONF_DEVICE_TYPE register is reset by default to "Root Complex".

For more details on sticky/non-sticky bits behaviour refer to the PCI Express Base 3.0 Specification, revision 1.0.

24.9.4.4.2.2 PCIe Standard Specific Resets to the PCIe Core Logic

Fundamental hardware reset - This reset impacts all non-sticky PCIe core bits. It is automatically triggered:

- upon a software initiated device type change in the PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE (to RC/EP)

- upon exitting certain low-power modes
- upon PCIe controller's main power supply (Vmain) being powered-up or Vmain detected below a defined threshold by the PCIe RC. This is a cold fundamental reset.

Note

All (sticky and non-sticky bits) of the PCIe controller core are reset upon fundamental hardware reset assertion because of Vaux not implemented. Hereby all PCIe core registers have to be re-initialized from scratch after a PCIe fundamental reset.

Hardware fundamental reset operation: The hardware fundamental reset is an input to the PCIe core, to be controlled by the local PCIe software driver. Its impact is conditional: a fundamental reset clears the entire controller (including the PCIe_PHY part), with the following exceptions:

- DEVICE_TYPE register (unconditional) and all other PCIe_SS_TI_CONF registers
- PCIe link PM FSM, if the module is out of "IDLE"

The fundamental reset is applied to PCIe core in the following two cases :

- After reprogramming the device type (to EP or RC) in the PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE bitfield, to latch in the new type and clear the programming registers. Remain out of idle and clear PCIECTRL_TI_CONF_PM_CTRL[11] AUX_PWR_DET to 0 beforehand, in order to reset also the sticky bits.
- L3 exit at initial power-up unless the restart is already initiated by the other side of the link

---

Note

The TYPE value ("EP", "legacy EP" or "RC") has to be read back by software until desired type-value is seen. This insures that the fundamental reset triggered by "TYPE" change has completed successfully. The user software must not change the PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE register value during PCIe core operation.

Note

The fundamental hardware reset does not impact the PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE (RC/EP) bitfield and the remaining TI wrapper configuration registers (PCIe_SS_TI_CONF). These can be reset only by a PCIe main reset input assertion.

PCIe link-down reset condition - When the PCIe link has gone down and goes up again, namely transitions from D3cold/L3 back to D0, the PCIe core auto-applies this **internal fundamental reset** and reports it on IRQ event (PCIECTRL_TI_CONF_IRQSTATUS_MAIN[11] LINK_REQ_RST) after link-up.

PCIe hot-reset condition - As defined by the PCIe standard: reset condition propagated in-band (over the PCIe wire) from an upstream port to a downstream port, using the TS1 and TS2 OS (see the PCI Express Base 3.0 Specification, revision 1.0). Requires the link to be already powered and up. The hot reset causes the link to go down and up again, causing a link-down reset condition, and as such is a fundamental reset.

The hot reset sequence can be tracked in the LTSSM by the transition to "hot reset" state, then on to "detect" and the rest of the link-up sequence.

- When RC, a hot reset is generated by writing the bit “Secondary Bus Reset” of the PCIe standard “Bridge Control Register” - BRIDGE_INT [22] SEC_BUS_RST bit .
- When EP, a host reset is automatically processed by the controller hardware.

PCIe soft reset condition - As defined by the PCIe standard: When the D-state of a device is changed from $D3_{hot}$ to D0 by software, namely by the RC doing a remote CFG write to the EP (EP) PM_CSR [1:0] PWR_STATE register, the EP device reacts depending on its DIF CS (local) setting of the no-soft-reset (NSR) bit (EP) PM_CSR [3] NSR:

- If bit NSR=0, the soft reset is applied, and the device transitions to the $D0_{unitialized}$ state (see Figure 24-162). Non-sticky bits are reset.
- If bit NSR=1, the soft reset is inhibited, the device transitions directly to the $D0_{active}$ state (see Figure 24-162), no reset is applied. Full context is retained, including non-sticky bits.

Note

The soft reset is a hot reset. The soft reset is not a fundamental reset; that means that it does not reset the PHY, only the programming registers.

24.9.4.5 PCIe Controller Power Management

This section is a brief summary of the supported power modes and power transitions, as well as the consequences (clock gating, power gating, resets, and so forth) on the PCIe controller.

24.9.4.5.1 PCIe Protocol Power Management

The power state of a PCIe controller port is described by four different FSMs, each running at a different level of the protocol, each with its own independent state. The FSMs are covered in Section 24.9.4.5.1.1 through Section 24.9.4.5.1.2.

Table 24-499summarizes the typical power stable state combinations, namely the combinations of states the controller can settle into, after a number of potential transitions and transient states, and assuming that the lowest power state possible is used.

The PCIe standard defined optional gating of the reference clock on PCIe_PHY PLL input (and therefore of the PIPE clock) when in L1-state is not supported. As shown in the Table 24-499, the PIPE clock is always running while link is in L1.

Table 24-499. Power States Summary

Device D-state	Link L-state	PHY LTSSM state	PIPE power down	PIPE clock(1)	Vmain	LP_CLK(4) (auxiliary clock)	Slave Idle	Master Standby
D0_uninitialized(2)	LDn	Detect	P1	OFF	don't care	ON	1	1
D0_uninitialized	L0	n/a	P0	ON	ON	ON	0	1
D0_active	L0	L0	P0	ON	ON	ON	0	0/1(3)
D0_active	L0s	L0s	P0s	ON	ON	ON	0	0/1(3)
D0_active	L1	L1	P1	ON	ON	ON	0	0/1(3)
D1	L1	L1	P1	ON	ON	ON	0	1
D2	L1	L1	P1	ON	ON	ON	0	1
D3_hot	L1	L1	P1	ON	ON	ON	0	1
D3_cold	L3	n/a	P2	OFF	OFF	OFF	1	1

(1) The PIPE clock activity, as visible from the PCIe core boundary, implies the activity of the high-speed PLL it is derived from, which implies the activity of the reference clock on the PLL input.
(2) Default controller state after reset. Not a real PCIe state.
(3) Master standby value depends on the PCI express application (MSE, ISE, and so forth)
(4) See Section 24.9.4.5.2

24.9.4.5.1.1 PCIe Device/function power state (D-state)

The function power state machine is the highest-level PM FSM in a PCIe device. The power state control bitfield is part of standard PCIe register bitfield- (EP)PM_CSR/(RC)PM_CSR [1:0] PWR_STATE.

Note

A Vaux power source is not tied to the PCIe core, that is why the state D3_cold with Vaux switched-on is unsupported by the PCIe PM FSM.

Note

The six available D-states are, by order of decreasing power: D0_uninitialized (default), D0_active, D1, D2, D3_hot, D3_cold. Note that the function power state Cfg field - is only 2 bit, encoding only 4 combinations: D0, D1, D2, D3. The difference between uninitialized and active (D0), and between hot and cold (D3) is implicit.

The eventual (“stable” or “quiescent”) power state of a PCIe link (link state) is determined by the D-state of the downstream device. For the current controller, it means that the D-state determines the link state in EP type mode, but not in RC type mode.

Note also that the device may go through a number of transient link states before entering/after exiting its quiescent link state, so that a device, for example, in D3_hot may be consuming maximum instantaneous power .

D0_uninitialized is the function’s default state, entered after powerup and/or reset. It is used by the RC software to enumerate the function (Cfg transactions). Mem and IO transaction are disabled. It can only be entered though a reset, partial or total. Together with D0_active, it is the highest-power D-state.

D0_active is the function’s active state, where all Memory and I/O transactions take place. By definition, a function transitions from D0_uninitialized to D0_active after the RC software enables Memory and I/O transactions, either as completer ("Cfg" write 0b1 to memory space enable: MSE/IO space enable: ISE) or as requester ("Cfg" write 1 to bus master enable: BME). D0_active can be exited to/entered from D1, D2, or D3_hot by the RC software writing to the function’s (EP)PM_CSR[1:0] PWR_STATE power state. Together with D0_uninitialized, it is the highest-power D-state.

D1 and D2 are two optional states, with reduced power compared to D0. Both operate the same way, and have the same action on lower-level PM FSM, with D1 expected to have a faster wakeup and D2 a lower power. Both are supported by the current controller. D1 and D2 can be exited to/entered from D0_active, D1, D2, or exited to D3_hot , by the RC software writing to the function’s (EP)PM_CSR[1:0] PWR_STATE power state bitfield. The current controller supports D1 and D2 but a fully compliant implementation also depends on the system providing the right combination of HW and software.

D3_hot is a lower-power state (compared to D0, D1, D2). It can be entered from D0_active, D1, or D2 by the RC software writing to the function’s (EP)PM_CSR[1:0] PWR_STATE. Transition to D3_hot is typically continued to D3_cold and main power (VMAIN) shutdown. It can also be exited to D0 by the RC writing (0b00) to the (EP)PM_CSR[1:0] PWR_STATE. The state transition to either D0_active or D0_uninitialized depends on the NSR value written in the bit (EP)PM_CSR[3] NSR (No_Soft_Reset bit) of 0b1 (that means - function is not reset) or 0b0 (that means - function is reset, link is restarted and must be reconfigured) respectively.

D3_cold is the lowest-power state, defined by the VMAIN being turned off. It it typically entered from D3_hot as shown in the figure below, but any turn-off request will actually place the function in that state. It can only be exited to D0_uninitialized, by restoring VMAIN and after resetting the function, totally or partially (see link states below).

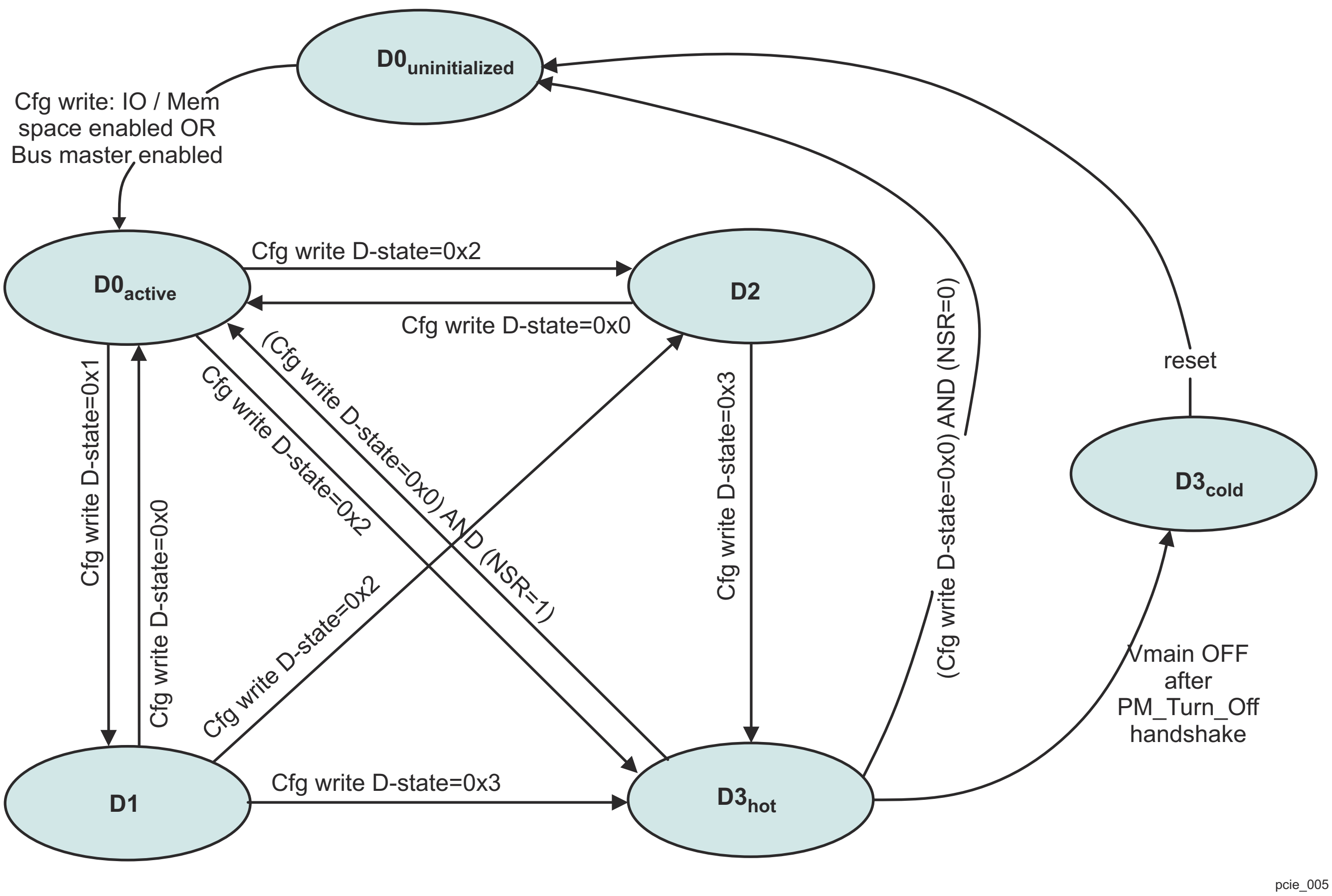


Figure 24-162. PCIe D-state (function power state) FSM diagram

The Figure 24-162 shows the state transitions in the power state. NSR is the “No_Soft_Reset” bit in the function’s (EP)PM_CSR configuration register. NSR advertizes the ability of the device to go into D3_hot while retaining its register settings (when 1), or conversely, the inability to do so and the reset of those setting upon D3_hot exit (when 0).

NSR is read-only to the RC software over Cfg accesses, but it can be set by the EP's local software over DIF (CS access) prior to enumeration. The controller shall apply the NSR to itself accordingly.

Most function power state transitions are done explicitly by the RC-side software writing into the EP function’s (EP)PM_CSR[1:0] PWR_STATE, over the PCIe wire. This is one of the two PM control mechanisms of PCIe, the other one being ASPM . The remaining function power state transitions are caused by resets.

The D-state of a device sets the link power state (L-state) for the PCIe link above. In other words, the L-state of a PCIe link (connecting an upstream device, closer to the RC, and a downstream device, closer to an EP) is set by the downstream device.

24.9.4.5.1.2 PCIe Controller PIPE Powerstate (Powerdown Control)

The PIPE interfaces the MAC, or controller, above PIPE in the protocol hierarchy and the PHY, or transceiver, below PIPE. The current controller implementation has a PIPE interface at its boundary.

> **CAUTION**
>
> The physical layer as defined by the PCIe standard extends into the PCIe MAC layer, to include, for example, the LTSSM, so that “physical layer” and PHY do not coincide, with the latter included in the former.

The PIPE interface has its own power state, driven by the PCIe controller towards PCIe_SS PHY on a 2-bit powerdown signal, depending on the PM state machines above. It has four states: P0, P0s, P1, P2:

- **P0 (pipe_powerdown = 0b00):** active state, for high-speed (HS) data transmission and reception. Used for L0 link state.
- **P0s (pipe_powerdown = 0b01):** active HS receiver, suspended transmitter. Used for L0s link state.
- **P1 (pipe_powerdown = 0b10):** Suspended HS receiver and transmitter, but PIPE clock still running.
- **P2 (pipe_powerdown = 0b11):** Suspended HS receiver and transmitter, PIPE clock stopped - asynchronous mode.

24.9.4.5.2 PCIE Controller Clocks Management

24.9.4.5.2.1 PCIe Clock Domains

The controller has two main clock domains: the L3_MAIN bridge (ports) and the PCIe core.

The L3_MAIN bridge runs on a single clock. That same clock is used to run both the PCIe controller master and slave ports. It can be divided, with independently ratios for master and slave. However, both sides are typically expected to run at the same speed, as both are capable of the same performance.

There is an auxiliary clock - LP_CLK which is a PCIe_SS internal low-power clock, used when pipe clock is stopped. It is assumed to be always available when the module is not idle.

The PCIe core runs on the PIPE clock whenever that clock is provided by the relevant PCIe_PHY. When the PCIe link is in a low-power state and the PIPE clock is not available, the core splits in two sub-domains:

- Most of the core stops
- A small “auxiliary” part of the core (which includes the configuration registers) switches to a low-power clock LP_CLK
- The two subdomains can still be considered synchronous despite their different clocking scheme, since they are either running on the same clock, or one of the subdomain clocks is stopped.

24.9.4.5.2.2 PCIe Controller Idle/Standby Clock Management Interfaces

A PCIe controller has two clock management (CM) interfaces with device PRCM:

- A master port clock management interface based on master standby protocol with the device PRCM
- A slave port clock management interface based on slave idle protocol with the device PRCM

The PCIe wrapper logic located register PCIECTRL_TI_CONF_SYSCONFIG[3:2] IDLEMODE and PCIECTRL_TI_CONF_SYSCONFIG[5:4] STANDBYMODE bitfields are used to software configure the idle / standby CM behaviour locally for the PCIe_SS controller. The Table 24-500, shows the available CM options.

Table 24-500. Local Power-Management Features

Feature	Registers	Description
Slave idle modes	PCIECTRL_TI_CONF_SYSCONFIG [3:2] IDLEMODE bit field	The available modes are: Force-Idle, No-Idle, wakeup-disabled Smart-Idle and wakeup-capable Smart-Idle.
Master standby modes	PCIECTRL_TI_CONF_SYSCONFIG [5:4] STANDBYMODE bit field	The available modes are: Force-Standby, No-Standby, wakeup-disabled Smart-Standby and wakeup-capable Smart-Standby.

For more information on the PCIeSS functional clock gating in the device PRCM (clock PCIE_L3_GICLK), refer to *Clock Domain-Level Clock Management*, in the chapter, *Power, Reset and Clock Management*.

The relations between the PCIeSS master (standby)/slave (idle) behaviour to various PCIe protocol link power management features are covered in the Section 24.9.4.5.2.2.1 and Section 24.9.4.5.2.2.2, respectively.

24.9.4.5.2.2.1 PCIe Controller Master Standby Behavior

As a master (initiator) on the device L3_MAIN interconnect, the PCIe controller supports the standby protocol over the standard three signals: mstandby/mwait/mwakeup. The behavior of that interface can be configured in the PCIeSS controller wrapper (PCIe_SS_TI_CONF) register - PCIECTRL_TI_CONF_SYSCONFIG[5:4] STANDBYMODE bitfield.

The standby protocol notifies the local system if inbound PCIe transactions are likely to be routed to the controller's master port.

Note

Note that CFG-type transactions are typically not part of the PCIe controller master port transfers: in RC mode they are only outbound, and in EP mode they are processed inside the controller.

Automatic mode (default): most operations are expected to take place in "smart-standby" mode. In that mode, the master standby interface is automatically managed by the PCIe status, featuring (Mstandby = 0) if (D-state=$D0_{active}$ and MSE=1 that is bit (RC) STATUS_COMMAND_REGISTER/(EP) STATUS_COMMAND_REGISTER[1] MEM_SPACE_EN) :

- D-state = D0_active indicates that the link is up, that data can be exchanged. It's also a condition for memory-type transactions to take place.
- The Memory Space Enable bit (RC) STATUS_COMMAND_REGISTER[1] MEM_SPACE_EN, indicating that the RC has enumerated the PCIe fabric and configured the memory space for operation, before EP writing the MSE bit in the local CFG register - (EP) STATUS_COMMAND_REGISTER via DIF CS access.

In automatic (smart-standby) mode, the standby is therefore controlled remotely by the RC software, which sets both the D-state and the MSE.

Note that the master does not go to standby during ASPM (so called active state power management) transitions to L0s or L1 on the link, as it remains in "D0_active" during that time. Although the physical layer is partially or fully stopped during L0s and L1, the short restart time of a few microseconds does not justify a transition to standby by the master. In other words, the master goes or doesn't go to standby after a link layer transition to L1, depending on the cause, if a transition in "D-state" or "ASPM" occured, respectively.

More generally, the link state is not directly correlated to the automatic standby state of the master:

- Master in standby (mstandby=1): link can be in L0 (link startup), L1 (D1, D2, D3hot), L3 (D3cold)
- Master out of standby (mstandby=0): link can be in L0, L0S/L1 (ASPM)

The "Mstandby" status depends on the possibility of transactions, not on actual transactions. The master may remain out of standby indefinitely without any transaction, and an incoming transaction shall not cause the master to exit standby. The latter is an error case which occurance is prevented in hardware.

The mwait=0 handshake, eventually returned after the mstandby=0 transition, should prevent the master from issuing transactions until the system is ready. However, no back-pressure can actually be applied to the PCIe to prevent inbound accesses after, for example, MSE has been set. It is assumed that there will be sufficient time for the mwait to transition, that is typically for the system to reopen the path to its local memory, before the first inbound transaction is routed to the master.

When the PCIe local controller itself is in RC mode, note that the D-state may remain D0active even after the has gone down. It is therefore advised to use the clearing of the local MSE bit to set the master in standby. Alternatively, the RC can be placed in D3hot mode, or the standby can be forced (to mstandby = 1) by setting the standbymode to "force-standby". Whatever the method, the RC is aware of the state of the entire PCIe fabric, and is expected to put its master in standby in full knowledge of the situation.

For more information on the PCIe Controller Standby Management Protocol with the device PRCM , see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

Manual control, EP type: When incoming transactions are not memory-type (e.g.: IO type), the master is capable of issuing transactions while the automatic out-of-standby conditions above are not met.

Note

In the EP, non-memory type transaction cases, it is the user software responsibility to manually force the master out of standby via setting the bit PCIECTRL_TI_CONF_SYSCONFIG[5:4] STANDBYMODE = 0x1, that means - "No-Standby", before the first transaction, and to return it to automatic mode via setting PCIECTRL_TI_CONF_SYSCONFIG[5:4] STANDBYMODE = 0x2, that means - "Smart-Standby" after the last transaction.

For instance, an EP shall have to monitor the ISE (IO space enable) bit and the link status, and set the PCIECTRL_TI_CONF_SYSCONFIG [5:4] STANDBYMODE bitfield accordingly if EP is configured to have an IO BAR.

24.9.4.5.2.2.2 PCIe Controller Slave Idle/Disconnect Behavior

As a target of the device L3_MAIN bus (with target being the PCIe controller slave port on L3_MAIN respectively), the controller supports the "Idle" protocol. The behaviour of that interface can be configured in the PCIeSS controller wrapper register - PCIECTRL_TI_CONF_SYSCONFIG[3:2] IDLEMODE bitfield.

Additionally, the slave port clock management supports a disconnect protocol which is controlled automatically.

The idle transition of the PCIe controller is strictly synchronized to PIPE clock. The PRCM places the PCIe controller to an "Idle state" before PCIePCS generated PIPE clock is stopped due to some reason: g.h. PCIe PHY low power transition with gating local PLL clocks, the APLL_PCIe clock or gating PCIeREF_DPLL clock. The IDLE state is automatically maintained by PRCM during periods in which PIPE clock is stopped.

PCIe controller transitions to IDLE initiated by the system PRCM are strictly synchronized with the PCIe operational state. The PRCM allows the PCIe controller to be brought out of "IDLE" state, only if the generation of the PIPE clock is enabled by configuring the high-frequency PLL (source of the PIPE clock) and the 100 MHz PCIe reference clock scheme (source of the high-frequency clock). For more details on the PCIe PLL and PHY related clocks, refer to the Section 26.4, *PCIe Shared PHY Susbsystem* in the chapter, *Shared PHY Component Subsystems* .

CAUTION

All accesses to the device PCIe controller on its slave port, namely to the local target configuration PL/PCIe standard/wrapper registers or to the PCIe bus, require the PIPE clock to be running. The presence of the PCIe interface clock - PCIE_L3_GICLK is not sufficient to perform such accesses.

24.9.4.5.2.2.2.1 PCIe Controller Idle Sequence During D3cold/L3 State

When the link of a RC enters L3 mode, the PIPE interface must go to P2 mode for power saving. The PCIe controller maintains the PIPE interface in P1 mode (PIPE clock running) and the internal registers accessible until the system requests entry into "Idle", and only then transition to P2: it is the device PRCM responsibility to Idle the module (PCIECTRL_TI_CONF_SYSCONFIG [3:2] IDLEMODE set to 0x2 - *Smart Idle mode* should be selected), in order to reach a low-power state. Transitioning to "P2", stops the PIPE clock and makes the register inaccessible, which in turn corresponds to an "IDLED" and disconnected from L3_MAIN PCIe slave port.

Note

The "L3" state can only be exited when the RC re-enabling the Vmain power supply, and reconfiguring all PCIe registers from scratch: In both EP and RC cases, the system should detect Vmain power assertion, apply a fundamental reset, then bring the module out of "IDLE" and restart the link normally. All registers are reset.

24.9.4.6 PCIe Controller Interrupt Requests

In the rest of this section, and for clarity:

- "**Interrupts**" refer by default to the virtual interrupt lines emulated by the PCIe protocol using either the PCI legacy or PCIe Message-Signaled Interrupts (MSI) methods, and used for non-PCIe purposes by the system.
- "**Hardware interrupts**" refer to the actual interrupt signals coming out of the current PCIe controller, and used to manage various events of the PCIe protocol, including PCI interrupts.

Each PCIe controller has two **hardware interrupt lines** described in Table 24-501 and Table 24-503, which are level-sensitive. PCIe pulse interrupts are unsupported in the device.

The main HW interrupt line - PCIe_SS1_IRQ_INT0/PCIe_SS2_IRQ_INT0 receives all interrupt events related to the control of the PCIe operation, including power management and errors.

The MSI HW interrupt line - PCIe_SS1_IRQ_INT1/PCIe_SS2_IRQ_INT1 receives all interrupt events propagated by a remote EP over the PCIe interface, for specific (non-PCIe) applicative purposes, in conjunction with the PCIe transactions. This includes the legacy interrupts (INTx) as well as the Message Signaled Interrupts (MSI) proper, as defined by the PCIe standard.

Note

MSI-X is unsupported by PCIe controller.

The difference between the two hardware interrupt types - "main" and "MSI" is that the main interrupt events are used by the PCIe interface, whereas the MSI interrupt uses the PCIe interface.

Note

Incoming MSI events only exist in RC mode when MSI events are received over the PCIe wire. In EP mode, other mechanisms allow controller to generate (transmit) of the same type of interrupt events onto the PCIe bus.

The main interrupt events are further divided into PCI management and error events. The MSI interrupts are divided into legacy and MSI proper. Each of these four categories of events is described in Section 24.9.4.6.1 and Section 24.9.4.6.2.2.

24.9.4.6.1 PCIe Controller Main Hardware Management

24.9.4.6.1.1 PCIe Management Interrupt Events

The events described here trigger the main interrupt line. Each status bit is set to one by a specific PCIe event, and must be cleared by the local SW.

CFG_MSE_EVT: (EP mode) PCIe controller’s Memory Space Enable (MSE) config bit has been set by the RC software - written to 1 by a configuration write. This notifies the local EP that incoming memory-type PCIe transactions may now target the local function, over the AXI master port.

CFG_BME_EVT: (EP mode) PCIe controller’s Bus Master Enable (BME) config bit has been set by the RC software - written to 1 by a configuration write. This allows the local EP to initiate (outgoing) PCIe transactions, over the AXI slave’s outbound window.

LINK_UP_EVT: Link layer has successfully booted and reached L0active state. Data can be transmitted and received over the PCIe wire.

LINK_REQ_RST: Link-down reset was triggered (by link-down condition). Non-sticky bits have been reset.

PM_PME: (RC mode) PCIe controller has received a power management event PM message on its PCIe downstream port, requesting wakeup.

PME_TO_ACK: (RC mode) PCIe controller has received a Turn-Off Acknowledge PM message on its PCIe downstream port, acknowledging a previously transmitted turnoff message (PME_Turn_Off).

PME_TURN_OFF: (EP mode) PCIe controller has received a Turn-Off PM message on its PCIe upstream port, requesting an acknowledge (PME_TO_Ack) and the transition of the link to L2/L3

24.9.4.6.1.2 PCIe Error Interrupt Events

The events described here trigger the main interrupt line.

ERR_COR: (RC mode) Correctable Error message received on PCIe downstream port.

ERR_NONFATAL: (RC mode) Uncorrectable Error message received on PCIe downstream port.

ERR_FATAL: (RC mode) Fatal Error message received on PCIe downstream port.

ERR_SYS: (RC mode) Either of the three error messages above (correctable, non-fatal, fatal) was reported and enabled in the root control register.

ERR_ECRC: End-to-end CRC error detected.

ERR_AXI: AXI slave error caused by response table overflow. This error is not supposed to happen.

24.9.4.6.1.3 Summary of PCIe Controller Main Hardware Interrupt Events

Table 24-501 describes the events, related to control of the PCIe interface operation itself, including power management, reset and error handling, that trigger the "main" hardware interrupt line.

Table 24-501. PCIe Main Hardware Interrupt Events Summary

Event Status and Clear	Interrupt Enable	Interrupt Disable	Mapping	Description
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[14] CFG_MSE_EVT	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[14] CFG_MSE_EVT_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[14] CFG_MSE_EVT_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	CFG "Memory Space Enable" change event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[13] CFG_BME_EVT	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[13] CFG_BME_EVT_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[13] CFG_BME_EVT_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	CFG "Bus Master Enable" change event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[12] LINK_UP_EVT	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[12] LINK_UP_EVT_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[12] LINK_UP_EVT_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Link-up state change" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[11] LINK_REQ_RST	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[11] LINK_REQ_RST_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[11] LINK_REQ_RST_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Link Request Reset" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[10] PM_PME	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[10] PM_PME_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[10] PM_PME_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Power Management PME message received" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[9] PM_TO_ACK	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[9] PM_TO_ACK_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[9] PM_TO_ACK_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Power Management acknowledge" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[8] PM_TURNOFF	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[8] PM_TURNOFF_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[8] PM_TURNOFF_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Power Management Turn-off" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[5] ERR_AER	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[5] ERR_AER_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[5] ERR_AER_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"ECRC Error" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[4] ERR_AXI	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[4] ERR_AXI_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[4] ERR_AXI_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"AXI tag lookup fatal error" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[3] ERR_COR	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[3] ERR_CORR_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[3] ERR_CORR_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Correctable Error" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[2] ERR_NONFATAL	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[2] ERR_NONFATAL_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[2] ERR_NONFATAL_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Non-Fatal Error" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[1] ERR_FATAL	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[1] ERR_FATAL_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[1] ERR_FATAL_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"Fatal Error" event
PCIECTRL_TI_CONF_IRQSTATUS_MAIN[0] ERR_SYS	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN[0] ERR_SYS_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN[0] ERR_SYS_EN	PCIe_SS1_IRQ_INT0 PCIe_SS2_IRQ_INT0	"System Error" event

Note

While the PCIECTRL_TI_CONF_IRQSTATUS_MAIN register is updated only if corresponding interrupt is enabled in the PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN register, the PCIECTRL_TI_CONF_IRQSTATUS_RAW_MAIN register always provides the Legacy/MSI event status regardless of an interrupt being enabled or disabled in the PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN register.

24.9.4.6.2 PCIe Controller Legacy and MSI Virtual Interrupts Management

The PCIe protocol uses TLPs to create "virtual" interrupt lines across the PCIe fabric. Interrupts do not constitute a separate transaction type like Mem, Cfg, I/O, and so forth, but reuse the existing types.

24.9.4.6.2.1 Legacy PCI Interrupts (INTx)

Legacy PCI interrupts are supported over PCIe by dedicated message codes (posted-write in Msg space) that flag the assertion or deassertion of one of four "virtual" interrupt lines (or pins): INTA, INTB, INTC, INTD. They can only be sent by EPs, and routed to/received by the RC.

PCIe controller supports single-function mode and can send only INTA when configured as EP.

Note

Use of legacy PCI interrupt is exclusive with use of MSI (defined in Section 24.9.4.6.2.1.2). MSI should be the preferred method whenever possible.

24.9.4.6.2.1.1 Legacy PCI Interrupt Events Overview

Legacy INTx (with x= A/B/C/D) are interrupt events inherited from PCI, where each was implemented on a physical bus wire. The RC can allocate each line to a given EP function, which is then allowed to assert/deassert that line (and only that line). PCI legacy interrupt events are mapped on the MSI interrupt line of PCIe controller. For more details, refer to Section 24.9.4.6.2.1.2 and Section 24.9.4.6.2.1.3.

The controller HW shall both set and clear the IRQ status bit according to the assert and deassert messages received over the PCIe wire, respectively. In other words, the legacy interrupt status bit do not need to be cleared by the local software after processing.

Typical legacy interrupt service routine (ISR) on the RC-configured PCIe controller is:

- Remote EP function transmits an "assert INTx" message. Value of x (within A/B/C/D) was previously assigned by the RC software using configuration accesses.
- "assert INTx" message is routed up the PCIe topology to the RC (local controller)
- PCIECTRL_TI_CONF_IRQSTATUS_MSI status bit INTx goes to 1
- MSI interrupt line is asserted
- software reads PCIECTRL_TI_CONF_IRQSTATUS_MSI status register. The asserted bit (A,B,C,D) allows to determines the EP function that generated the event.
- software accesses this EP function over PCIe to process the interrupt: ad-hoc service routine
- As a result of the processing, remote EP function transmits the "deassert INTx" message
- software clears the PCIECTRL_TI_CONF_IRQSTATUS_MSI status bit INTx to 0.
- MSI interrupt line is deasserted, assuming there is no other asserted event.

24.9.4.6.2.1.2 Legacy PCI Interrupt Transmission (EP mode only)

Writing 0b1 to TI wrapper register PCIECTRL_TI_CONF_INTX_ASSERT[0] ASSERT_F0 bit sends an ASSERT message on the "interrupt pin" specified in the config header space.

Writing 0b1 to TI wrapper register PCIECTRL_TI_CONF_INTX_DEASSERT[0] DEASSERT_F0 bit will send a DEASSERT message on the "interrupt pin" specified in the config header space.

A status of the virtual "interrupt line" is available by reading out either bitfield.

Note

This status is just a local record of the last message sent, it is not the result of any feedback from the PCIe fabric, since messages are posted, no response is generated. For that reason, the PCIe protocol does not guarantee the coherence between assertion and deassertion INTx messages, that means that it's up to the application to ensure that the INTx assertion has been seen by the RC before sending the deassertion, and vice versa.

CAUTION

The operations above should only be used when in EP mode, and will produce undefined results if applied to an RC-configured controller.

24.9.4.6.2.1.3 Legacy PCI Interrupt Reception (RC mode only)

The reception of either of the 8 INTX message codes (ASSERT/DEASSERT of INTA/B/C/D) shall assert/deassert the corresponding HW interrupt event in status register:

- PCIECTRL_TI_CONF_IRQSTATUS_MSI[0] INTA
- PCIECTRL_TI_CONF_IRQSTATUS_MSI[1] INTB
- PCIECTRL_TI_CONF_IRQSTATUS_MSI[2] INTC
- PCIECTRL_TI_CONF_IRQSTATUS_MSI[3] INTD

This implies that the status of an asserted event does not need to be cleared locally (by software writing 1 to the PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI[n] INTx_EN bit, where n=0 to 3 and x=A, B, C, D, respectively) as it is expected to be cleared remotely over PCIe, by the EP function.

Note

The RC does not keep track of the INTx requester, even though the information is available in the incoming TLP's RequesterID: each legacy interrupt is expected to be used by only one EP function, known apriori by the RC.

When messages are posted, there is no response generated by the RC.

INTx messages can only be received (and the corresponding HW interrupts asserted) when in RC mode.

Note

The HW interrupts have to be maintained disabled when in EP mode.

24.9.4.6.2.2 PCIe Controller Message Signaled Interrupts (MSI)

Message Signal Interrupts are the preferred option to map interrupt over PCIe. It allows up to 32 interrupts vectors/events to be triggered by each EP function. MSI interrupts are implemented memory-space (posted) writes, 1-DWORD long. They should only be sent by EPs to the RC.

Note

While legacy PCI interrupts are mapped onto PCIe messages (Msg), the Message Signaled Interrupts are mapped onto PCIe memory writes (MWr), not onto messages.

Use of MSI is exclusive with use of legacy PCI interrupts (defined above).

An MSI is a memory (posted) write done by an EP to a mailbox in the RC. The data size is always 32 bit, the address and the data are programmed by the RC during PCIe enumeration, into the EP function's MSI capability descriptor (in the EP function's PCI config space).

Like legacy PCI interrupts, MSI does pass any status information to the RC, other than the interrupt vector (event) itself: the data payload of the write access is used exclusively to identify the interrupt's requester and its vector. The exchange of status information is outside the scope of MSI.

Note

Unlike the legacy PCI interrupts, MSI does not emulate an interrupt "level" that is either high or low, gets asserted or deasserted. Each MSI interrupt vector corresponds to a univocal event, that does not get "cleared" later on.

Note

The device integrated PCIe controller does not support extended MSI-X message signaled interrupts.

24.9.4.6.2.2.1 PCIe Specific MSI Interrupt Event Overview

MSI are interrupts designed specifically for PCIe, unlike the legacy PCI events seen in the Section 24.9.4.6.2.1. They are mapped on the MSI interrupt line of PCIe controller, using a single status bit in the PCIECTRL_TI_CONF_IRQSTATUS_MSI register. The "MSI controller" decodes and records each received MSI write as a specific "vector" coming from a specific EP function. Each application assigns a dedicated meaning to a given vector. Each function can use up to 32 vectors, as configured by the RC. MSI vectors can only be "asserted" by the EP function, unlike PCI legacy events that can be asserted and deasserted. In other words, the local software has to "clear" the asserted vector before it can be reused.

The PCIECTRL_TI_CONF_IRQSTATUS_MSI status bit MSI must remain set as long as a vector is still set. Once the last vector has been cleared, the MSI bit must be cleared by software.

See also Section 24.9.4.6.2.2.3.

Typical MSI interrupt service routine on the RC-configured PCIe controller:

1. Software reads PCIECTRL_TI_CONF_IRQSTATUS_MSI status register, and identifies an (unspecified) MSI event, as opposed to a PCI legacy event.
2. Software clears PCIECTRL_TI_CONF_IRQSTATUS_MSI[4] MSI.
3. Software reads PCIECTRL_PL_MSI_CTRL_INT_STATUS_N and identifies EP functions that raised MSI.
4. Software clears PCIECTRL_PL_MSI_CTRL_INT_STATUS_N status bit.
5. Software calls the EP function specific interrupt handlers for each of the vectors.
6. Repeat steps 3 to 5 until PCIECTRL_PL_MSI_CTRL_INT_STATUS_N reads 0.
7. Exit MSI IRQ handler, only if PCIECTRL_PL_MSI_CTRL_INT_STATUS_N is 0.

24.9.4.6.2.2.2 PCIe Controller MSI Transmission Methods (EP mode)

There are two methods for MSI transmission: hardware and software. MSI transmission applies only to EP configured local PCIe system.

24.9.4.6.2.2.2.1 PCIe Controller MSI transmission, hardware method

The device PCIe controller (EP) transmits an MSI by writing 0b1 to the TI config wrapper register PCIECTRL_TI_CONF_MSI_XMT[0] MSI_REQ_GRANT bit. The other fields indicate function number, TC, vector, for the MSI. The address, data (potentially modulated by the vector) are automatically extracted from the EP function's MSI capability descriptor, previously programmed by the remote RC during enumeration.

For status, PCIECTRL_TI_CONF_MSI_XMT[0] MSI_REQ_GRANT can be polled. The bit remains HIGH (0b1) following a request (write to 0b1) and goes back LOW (0b0) once the MSI transmit request has been granted.

Note

Bit PCIECTRL_TI_CONF_MSI_XMT[0] MSI_REQ_GRANT reflects only the PCIe local status, that is a confirmation that the MSI has been transmitted. Since the MSI is a posted write, there is no way for the EP to ensure that it has been correctly received by the RC.

24.9.4.6.2.2.2.2 PCIe Controller MSI transmission, software method

Alternatively, an MSI can also be sent as a SW-composed memory write (MWr). MSI address and data shall be taken from the local EP function's MSI capability descriptor, where they were programmed by the RC at enumeration, along with the "multiple message enabling". If more than one vector is enabled, MSI data can be modulated accordingly. Furthermore, the transaction shall match the mandatory characteristics of an MSI MWr access, as follows :

- Header attribute bits ATTR = 2'b00 (no snoop = 0, relaxed ordering = 0)
- 1 DWORD long
- First byte enable: 0b0000
- Last byte enable: 0b0011 (16 bits of data)

Such a SW-generated MSI can be scheduled at the end of a series of memory transactions (MRd/MWr), with the PCI transaction ordering rules guaranteeing that the MSI will not overtake the other transactions.

24.9.4.6.2.2.3 PCIe Controller MSI Reception (RC mode)

Unlike legacy PCI interrupts, MSI must be configured by the RC in each function before use. Depending on its reception capacities, the RC can enable between 1 and 32 vectors (individual events) per MSI-enabled EP function.

The current RC implementation supports MSI reception for up to 8 different EP functions, with up to the protocol's maximum (32) vectors a piece, all mapped to a single MSI address. MSI reception is handled by the "MSI controller" though non-standard "port logic" (PL) registers (mapped in the PCI config space): For more details see the PCIe controller port logic associated PCIECTRL_PL_MSI_CTRL_x registers in the Section 24.9.7.5.1.

The reception of MSI implicitly requires the MSI mailbox address (in PCI memory space) to be routed to the RC: refer to Section 24.9.4.7, *PCIe Controller Address Spaces and Address Translation* for that. The MSI address (32-bit or 64-bit) is programmed in dedicated PCIe PL registers.

The MSI message data (MWr DWORD) is encoded as follows for the current RC. This encoding shall be used by the RC to configure the remote EP discovered at enumeration.

Table 24-502. MSI Message Data Encoding (RC)

Bit range	Size	Value	Comment
31:16	16	0x0000	Reserved as per PCI standard
15:8	8	0x00	Unused, always zero
7:5	3	0 to 7	Identifier of EP function in MSI controller (8 possible values)
4:0	5	0 to 31	Interrupt vector for the EP (32 possible values)

An inbound access recognized as MSI is terminated inside the controller, and does not appear on the PCIe controller's master port like other inbound memory writes in the same range.

24.9.4.6.3 PCIe Controller MSI Hardware Interrupt Events

Table 24-503 describes the events, related to non-PCIe interrupts (PCI legacy and PCIe MSI interrupts), which trigger the "MSI" hardware interrupt line.

Table 24-503. MSI Hardware Interrupt Events Summary

Event Status and Clear[(1)]	Interrupt Enable Bit	Interrupt Disable Bit	Mapping	Description
PCIECTRL_TI_CONF_IRQSTATUS_MSI[0] INTA	PCIECTRL_TI_CONF_IRQENABLE_SET_MSI[0] INTA_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI[0] INTA_EN	PCIe_SS1_IRQ_INT1 PCIe_SS2_IRQ_INT1	Legacy interrupt caused by EP sending inband ASSERT/ DEASSERT event on virtually emulated by PCIe pin INTA. Typically set and cleared by the remote EP, without local software intervention.
PCIECTRL_TI_CONF_IRQSTATUS_MSI[1] INTB	PCIECTRL_TI_CONF_IRQENABLE_SET_MSI[1] INTB_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI[1] INTB_EN	PCIe_SS1_IRQ_INT1 PCIe_SS2_IRQ_INT1	
PCIECTRL_TI_CONF_IRQSTATUS_MSI[2] INTC	PCIECTRL_TI_CONF_IRQENABLE_SET_MSI[2] INTC_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI[2] INTC_EN	PCIe_SS1_IRQ_INT1 PCIe_SS2_IRQ_INT1	
PCIECTRL_TI_CONF_IRQSTATUS_MSI[3] INTD	PCIECTRL_TI_CONF_IRQENABLE_SET_MSI[3] INTD_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI[3] INTD_EN	PCIe_SS1_IRQ_INT1 PCIe_SS2_IRQ_INT1	
PCIECTRL_TI_CONF_IRQSTATUS_MSI[4] MSI	PCIECTRL_TI_CONF_IRQENABLE_SET_MSI[4] MSI_EN	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI[4] MSI_EN	PCIe_SS1_IRQ_INT1 PCIe_SS2_IRQ_INT1	MSI interrupt status. It is cleared by clearing all vectors in the MSI controller (PL) registers.

(1) The MSI hardware interrupts have to be maintained disabled when in EP mode.

Note

While the PCIECTRL_TI_CONF_IRQSTATUS_MSI register is updated only if corresponding interrupt is enabled in the PCIECTRL_TI_CONF_IRQENABLE_SET_MSI register, the PCIECTRL_TI_CONF_IRQSTATUS_RAW_MSI register always provides the Legacy/MSI event status regardless of an interrupt being enabled or disabled in the PCIECTRL_TI_CONF_IRQENABLE_SET_MSI register.

24.9.4.7 PCIe Controller Address Spaces and Address Translation

The PCIe controller acts as a general-purpose bridge between the device local PCIe system and the PCIe fabric. Both are address-based mappings, and the bridging requires address translation between the different spaces.

The Table 24-504 lists the memory-mapped spaces interconnected by each of the PCIe_SS1 and PCIe_SS2 controllers. For each address space (row), the local PCIe subsystem (including the PCIe core) provides “entry points” (4th column) to inject transactions into the specified space, and ”exit points” (5th column) to receive transactions from the specified space.

Table 24-504. PCIe_SS Address Space Map

Space name	Address width	Size (bytes)	Entry point	Exit point	Comments
Device L3_MAIN memory space visible through PCIe controller's master port	32-bit	4 GiB	iATU IBR output	device system RAM, miscellaneous modules	Accessed over PCIe controller initiator (master) port; Includes the outbound window below.
PCIe_SS controller outbound window in device	28-bit	256 MiB	Device CPU hosts (MPU, DSPs, and so forth) of PCIe subsystem on device L3_MAIN interconnect	iATU OBR input	Mapped within PCIe controller target (slave) port, the outbound window is itself remapped within the device L3_MAIN memory space (256 MiB sized region).
PCIe memory space	32-bit/64-bit[(1)]	4 GiB/16 EiB	ATU OBR output (mem mode)	ATU IBR input (mem mode)	Contains all memory BARs
PCIe I/O space	32-bit	4 GiB	ATU OBR output (I/O mode)	ATU IBR input (I/O mode)	Contains all I/O BARs
PCIe config space (ECAM)	28-bit	256 MiB	ATU OBR output (configuration mode)	Configuration registers inside PCIe controller	Contains the PCIe-standard configuration spaces for EP functions (a 4 KiB- descriptor per function).

(1) Regarding the PCIe address space shared between all devices identified on the PCIe fabric, the local PCIe controller OBR iATU output generates a 32-bit address (it addresses a total of 4 GiB external PCIe space), while the IBR iATU input receives accesses from within a total of 16 ExaBytes of external PCIe space (a 64-bit address).

By definition, an iATU (Address Translation Unit) has an input (the space the access originates from) and an output (the space the access is headed for), with the required address translation taking place in the middle. An iATU inbound region (IBR) routes from PCIe (input) to device L3_MAIN space (output), an iATU outbound region (OBR) routes from the L3_MAIN space (input) to the PCIe (output).

Table 24-505summarizes the action of the different ATU regions. It contains the same information as the table above, organized differently. Note that inbound regions are not required to managed incoming configuration accesses, which are completed automatically by the PCIe controller hardware.

PCIe controller core outbound and inbound ATUs define:

- 16 outbound iATU regions
- 4 inbound iATU regions

Table 24-505. Address Translation Unit Region Access Summary

ATU region direction	ATU region type	Source space	Destination space
Outbound (OBR)	Memory	PCIe controller outbound window	PCIe memory space
Inbound (IBR)	Memory	PCIe memory space	Device PCIe subsystem space
Outbound (OBR)	I/O	PCIe controller outbound window	PCIe I/O space
Inbound (IBR)	I/O	PCIe I/O space	Device PCIe subsystem space
Outbound (OBR)	Configuration	PCIe controller outbound window	PCIe Cfg space

In a read/write access to an iATU outbound region (input), which is mapped in the PCIe controller target's 256 MiB-wide outbound window, do not confuse:

- Physical address accessed in the device L3_MAIN address space - 4 GiB
- Physical address within the PCIe_SS1/PCIe_SS2 controller target (slave) port:
  – PCIe outbound window - Cfg ; Memory and I/O space accessible within (0x2000_0000 - 0x2FFF_FFFF)/ (0x3000_0000 - 0x3FFF_FFFF)
  – PCIe controller local configuration/status registers accessible within (0x5100_0000 - 0x517F_FFFF)/ (0x5180_0000 - 0x51FF_FFFF)
- Address offset within the outbound window (0 - 256 MiB)

CAUTION

In the calculation of an ATU outbound region translation, the input is the address offset within the outbound window, ranging from 0x0000_0000 (0 bytes) to 0x0FFF_FFFC (256 MiB – 4 Bytes), they are the third and last address type in the list above.

PCI express is an interconnect that propagates read/write accesses from/to systems beyond the local device PCIe subsystem. The figure below shows such a link, where the transaction initiator (PCIe requester device, on the left) and the transaction target (PCIe completer device, on the right) are two instances of the PCIe controller subsystem, connected over the PCI express fabric (in the middle). The physical routing of the transaction over the PCIe topology (through switches) is implicit and transparent for addresses. A simplified chain of address translation is represented for each PCIe transaction type (Configuration, Memory or IO).

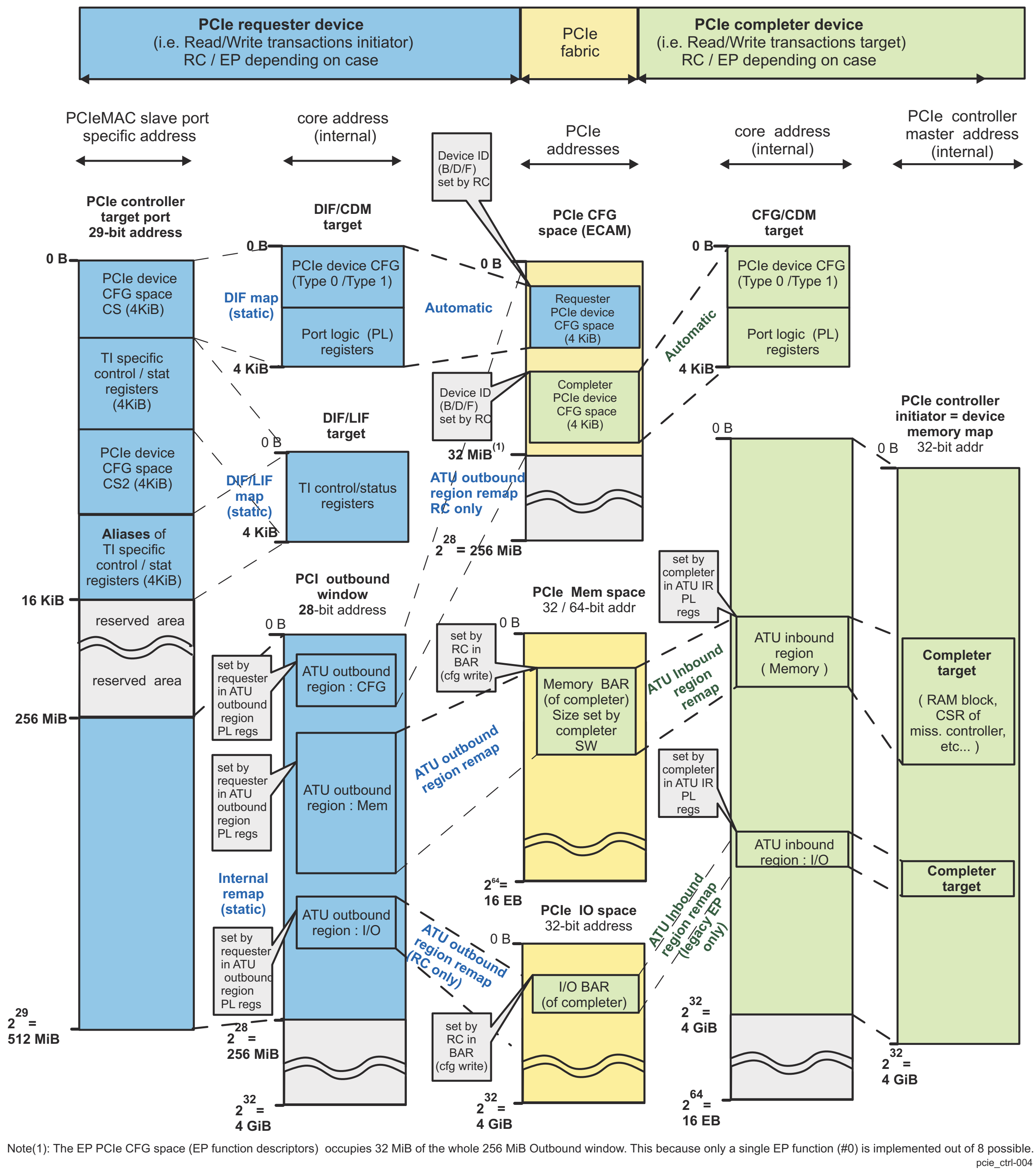


Figure 24-163. PCIe Core to PCIe Core Address Mapping and Translation

Note

There is a difference between requester/completer and RC/EP relationships, an EP device can be either the requester or the completer in a memory transaction.

24.9.4.8 PCIe Traffic Requesting and Responding

The PCIe wire can carry several types of packet-based traffic. The current chapter details how the current controller can be requester and completer of each type:

- **Requester:** initiates and transmits an outbound request, receives a response if required by (non-posted) request type.
- **Completer:** receives and processes an inbound request, generates and transmits a response if required by (non-posted) request type.

24.9.4.8.1 PCIe Memory-type (Mem) Traffic Management

The bulk of PCIe traffic is expected to take place in the memory (MEM) space, making use of large TLP/large L3_MAIN interface bursts.

Before PCI activity starts, one or several BAR of each device (EP function or RC) shall be initialized in Memory mode, as described in the Section 24.9.4.9.3, *Base Address Registers (BAR) Initialization*.

The following fields shall be set, using (CS) access (they are RO over PCI) :

- BARn[0]= "space decoder" shall be set to 0 = MEM space. This is the default setting for all BAR.
- BARn[2:1] = "type" shall be configured to either 32 or 64-bit. In the latter case, BARn+1 above shall contain the 32-bit upper part of the BAR base address.
- BARn[3] = "prefetchable" (P)

The BAR shall be enabled, using (CS2) access to BARn[0]:

- Write 1: BARn is enabled, can be configured by the RC writing to the MSBits of the BARn base address field.
- Write 0: BARn is disabled: register is read-only and return 0 over PCIe.

As part of the standard enumeration process, the RC then maps all the Memory BARs it finds (while discovering the connected EP functions) into the memory address space (32 or 64-bit address), using Cfg accesses over the PCIe wire. Optionally, the RC can also configure its own two BAR, over the DIF interface.

Once that process is complete, memory requests can take place point-to-point, that means that the requests can be initiated by any device (EP or RC) towards any other device function.

24.9.4.8.1.1 PCIe Memory Requesting

An ATU outbound region can then be programmed to remap the desired parts of the memory space (which correspond to BARs on other devices) into the local controller target's outbound window, accessed through the L3_MAIN slave port.

Alternatively, the iATU can be bypassed, and the outgoing access forwarded to the PCIe wire without address translation. In that *pass-through* mode, the PCIe address is equal to the 28 LSBits of the PCIe controller target address, that is, the address offset within the 256 MiB-wide outbound window. This gives access to the lowest 256 MiB of the PCIe memory space, from 0x0 to 0x0FFF_FFFC.

The forwarded PCIe TLPs shall have the following (hardcoded) header field values:

- TYPE = 0 = MRd/ MWr (Memory read or write)
- EP (poisoned bit) = 0 (not poisoned)
- ATTR (Attributes) = 2b01:
  - ATTR[1] = RO (relaxed ordering) = 0
  - ATTR[0] = NS (no snoop) = 1
- TC (traffic class) = 0

Initiators in the device local PCIe subsystem (typically CPUs or DMA) can then access memory space by initiating read/write accesses on the controller's slave port over L3_MAIN, either to the range of an enabled ATU outbound region, for the translated mode (with all TLP fields configurable), or outside any enabled ATU outbound region, for the pass-through mode (with the TLP fields hardcoded as explained above).

24.9.4.8.1.2 PCIe Memory Responding

An ATU inbound region can be programmed in BAR matching mode to remap (part or all of) each memory BAR to the controller's inbound window (this means to translate their address to the memory space of the device chip), and issue them on the PCIe controller's master port towards L3_MAIN. BAR matching mode is expected to be the most common method for an EP. In that mode, the ATU can be set up before enumeration - before the BAR's actual range (offset) is configured by the remote RC.

Alternatively, the ATU inbound region can be programmed in address matching mode. In that case, any access routed to the device (EP or RC) and matching the range specified in the ATU is remapped to the controller's inbound window (this means to translate their address to the memory space of the device), and issued on the master port towards L3_MAIN.

Address matching is the preferred method for an RC, which would route any access outside its base/limit range, as defined in its PCIe standard (Type-1) configuration space, and independently of any BAR so that BAR matching is not possible. In that case, only space not already taken up by a BAR can be used. Note the setup of an address matching iATU inbound region can only be done after enumeration, that is, after the RC has discovered and mapped all enabled Memory BAR of all EP functions to their address.

Finally, in case an incoming access does not match any enabled ATU inbound region (BAR or address matching), the transaction is just passed through untranslated, that is, with the address unchanged and the TLP attributes stripped off. This is not expected to be commonly used.

Incoming memory requests (to the local EP's Memory BAR) shall then cause the master port on L3_MAIN to access a given address range of the local system (potentially the entire 32-bit address range).

24.9.4.8.2 PCIe Configuration Type (Cfg) Traffic Management

Configuration (Cfg) traffic is not symmetrical: it can only be requested by the RC, and completed (in the case of PCIe controller) by the EP. It is used by the RC for the enumeration of the fabric: EP, functions, switches.

Note

The RC cannot use configuration traffic over the PCIe wire to access its own 4 KiB configuration space. To do that, it uses DIF accesses (CS) with bit PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN = 0b0.

24.9.4.8.2.1 RC Self-configuration (RC Only)

Like any other device on the PCIe fabric, the RC has its own 4-KiB configuration space, where RC local configuration registers are stored.

Note

Since the RC configuration registers are programmed by the RC itself, no remote access over the PCIe wire (Cfg TLP) is required. Local DIF accesses (CS) are used instead.

However, some register fields normally defined as read-only by the PCI standard are actually programmable over DIF (see Section 24.9.4.9.2), which may cause a standard software driver to fail. In that case, the CS-writability of those fields has to be disabled by reprogramming the located in the PCIe controller port logic bit PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN to 0b0. Note that this is normally done after CS-writable fields have been configured by the non-standard initialization "firmware", thus making the RC's configuration space PCI compliant.

The Section 24.9.4.8.2.2 and Section 24.9.4.8.2.3 only covers configuration of an EP function by an RC, using actual Configuration TLPs (transport layer packets).

24.9.4.8.2.2 Configuration Requests over PCIe (RC Only)

After coming out of reset, the PCI fabric is enumerated by the RC through configuration (Cfg) transactions to the switches and EP present in the topology. The enumeration process is covered by the PCI and PCIe standards - for more details refer to the PCI Express Base 3.0 Specification, revision 1.0.

Configuration transactions are created by an outbound ATU region set up to translate the PCIe controller slave port accesses into Cfg TLP on the PCIe wire.

- The L3_MAIN interconnect port (read or write) determines the TLP type (CfgRd/CfgWr)
- The data is passed unchanged from the L3_MAIN interconnect to the PCI world (write) and reciprocally (read)
- The entire PCI configuration space can be mapped as a standard, contiguous block, as specified in the **Enhanced Configuration Access Mechanism (ECAM)**. The ATU maps this 256 MiB block at a programmable base address within the controller slave port's 256 MiB "outbound window", using the ATU "Cfg shift" feature.
- The other parameters are programmed beforehand in the ATU's configuration registers (IATU_*)

The (28-bit) address offset within the 256 MiB ECAM block is the concatenation of :

- **Cfg register address** (Offset[11:0] = 12-bit = 4 KiB configuration space per EP function)
- **Function number** (Offset[14:12] =3 bit = 8 functions per PCIe EP)
- **Device number** (Offset[19:15] = 5 bit = 32 devices per bus, always 0 for PCIe controller itself)
- **Bus number** (Offset[27:20] = 8-bit = 256 busses in the PCIe fabric)
- Total for ECAM: 8 + 5 + 3 + 12 = 28-bit offset, or 256 x 32 x 8 x 4 KiB = 256 MiB

Note

The device PCIe controller itself supports only one function in EP mode. On the PCIe fabric, however when in RC mode, the PCIe controller may need to configure up to 8 functions per EP (potentially for all 65536 EPs). Hence the ECAM config. space visible to RC is 256 MiB in size.

A Cfg read access to the Vendor ID register is used by the RC to detect the presence of a function. If that specific read (address 0x000 in the 4-KiB PCI config space) fails with an UR PCIe response (Unsupported Request: no function present), the PCIe controller shall return a correct response to L3_MAIN with data 0xFFFF. This indicates the absence of the function to the application, without triggering an error on L3_MAIN.

24.9.4.8.2.3 Configuration Responding over PCIe (EP Only)

As an EP, the device only expects to receive (and never transmit) configuration transactions, and expects those to be CFG0 and not CFG1.

Incoming Cfg accesses with the appropriate Target ID are automatically routed to the targeted EP function's 4-KiB PCI configuration space, which contains both the PCI standard registers (header, descriptors, extende descriptors, and so forth) and the PL registers. The request read/write complete automatically, with no transaction appearing on the PCIe controller master port. Since Cfg accesses are non-posted, each incoming request shall generate a response.

Note

An ATU inbound region is also capable of processing incoming Cfg transactions, and translating them into read/write accesses on the master port towards L3_MAIN. However, this method is not expected to be used, and automatic completion –described above– is expected to be used instead.

24.9.4.8.3 PCIe I/O-type (IO) traffic management

I/O traffic is not symmetrical: it can only be requested by the RC, and responded to by the EP. It is a mostly deprecated feature of PCI, and could be replaced by Memory traffic. However, it is still required by the standard for a PCIe RC. It is supported by the current controller in both RC (requester) and **legacy EP mode (completer)**.

24.9.4.8.3.1 PCIe I/O requesting (RC only)

As part of the standard enumeration process, the RC maps all the I/O BARs it finds (while discovering the connected EP functions) into the 4 GiB I/O address space (32-bit address).

An ATU outbound region shall then be programmed to map a zone of the I/O space into the controller target's outbound window, accessed through the slave port.

Initiators in the RC host (typically a local CPU) can then access I/O space by initiating (1 DWORD) read/write accesses to the ATU region's range on the controller's slave port on L3_MAIN.

Note

There is NO specific hardware mechanism to notify the RC that it is allowed to issue I/O requests, since that configuration decision comes from the RC in the first place.

24.9.4.8.3.2 PCIe IO BAR initialization before enumeration (EP only)

Before PCI enumeration, each BAR can be initialized/enabled individually for IO space, using the DIF access (CS or CS2). The size of an enabled IO BAR (that is, the number of writable bits in the base address) is fixed to 256 bytes, that is, a mask of 0xFF that is, the writable BAR base address range for the RC is BARn[31:8], while BARn[7:0] is read-only.

In (CS) access, the BARn[0] "space decoder" field becomes RW and shall be written to 1 = IO space

In (CS2) access, BARn[0] becomes a write-only BAR enable :

- Write '1': BARn is enabled, can be configured by the RC writing to the MSBits of the BARn base address field.
- Write '0': BARn is disabled: register is read-only and return 0x0 over PCIe.

24.9.4.8.3.3 PCIe I/O responding (PCI legacy EP only)

I/O support is permitted only for the "legacy EP" type, that is, EP's side it is required that PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE has been preliminary set to 0x1 by EP's local software. Before PCI activity starts, one or several BAR of the EP function shall be enabled in I/O mode, to get enumerated by the RC as per the PCI standard.

An iATU (internal address translation unit) inbound region can be programmed in **BAR matching mode** to map the BAR to the controller's inbound window (that is, translate their address to the memory space of the SoC), and issue them on the PCIe master port towards device L3_MAIN.

BAR matching mode is expected to be the most common method for an EP. In that mode, the iATU can be set up before enumeration, that is, before the BAR's actual range (offset) is configured by the remote RC.

Alternatively, the ATU inbound region could be programmed in address matching mode. However, a valid I/O access coming into an EP is necessarily within a BAR, so BAR matching should be used. Additionally, BAR matching can be configured before enumeration, that is, before the RC has mapped the I/O BAR to their actual address.

Incoming I/O requests to the local EP's IO BAR shall then cause the PCIe master port to access a given address range of the local system (potentially the entire 32-bit address range), in 32-bit read/write accesses.

24.9.4.8.4 PCIe Message-type (Msg) traffic management

Most messages are generated and received automatically by logic in the controller HW, and do not require any specific intervention from the user.

The application can transmitted the following power management messages by writing to the TI configuration wrapper instance PCIECTRL_TI_CONF_PM_CTRL register:

- PCIECTRL_TI_CONF_PM_CTRL [0]PME_TURN_OFF: (power management event, turn off) request from RC to turn off, broadcasted downstream.

- PCIECTRL_TI_CONF_PM_CTRL[1] PM_PME: (power management event): request from EP to wake up, routed upstream towards RC.

No dedicated logic exists in the controller to support vendor-defined messages.

The ATU is capable of generating (outgoing) messages, and receiving (incoming) ones, by translating them from read/write accesses on the PCIe controller slave port, or to read/write accesses on the PCIe controller master port, respectively. However, this functionality is not required for standard PCI operation.

24.9.4.9 PCIe Programming Register Interface

PCI is an memory-mapped, dynamically reconfigurable interconnect protocol, so that the same physical register may be accessed in several different ways. The current chapter targets to describe only the statically-mapped programming registers of the local PCIe controller, accessed over the local AXI slave.

They are composed of:

- **PCIe-standard configuration registers:** by definition accessible over the PCIe wire through the PCIe configuration (and extended configuration - ECAM) space. Those registers are different depending on the selected mode (Type-0: EP or Type1: RC).
- **Port Logic (PL) registers:** non-standard user-accessible registers for additional PCIe functionality. They are also accessible over PCIe wire, in the configuration (ECAM) space.
- **TI-specific control and status registers:** more non-standard registers, accessible only locally. These are associated with the PCIe controller wrapper level functional control and status.

24.9.4.9.1 PCIe Register Access

The Table 24-506 summarizes how the resources of the local controller can be accessed. Note that the rightmost column represent accesses to the PCIe controller hosts system memory.

Table 24-506. Summary of Access Types related to PCIe Controller Local Resources

Access	PCIe standard configuration registers	Port Logic (PL) registers	TI-specific registers	System memory available to PCIe
Local DIF access (over PCIe controller target (slave) port, from device host CPUs and DMAs)	Yes[(1)] (CS or CS2)	Yes	Yes	No
Remote PCIe config (Cfg) transaction (over PCIe wire)	EP: Yes[(1)] RC: No[(2)]	EP: Yes RC: No[(2)]	No	No
Remote PCIe Memory transaction (over PCIe wire)	No	No	No	Yes (iATU)
Remote PCIe IO transaction (over PCIe wire)	No	No	No	EP: Yes (iATU) RC: No[(3)]

(1) 1. Read and writes to PCIe standard configuration registers (in PCIe config space, address 0x0 to 0x700) will give different results depending on the access method (DIF CS/CS2, PCIe): see [(1)].
(2) 2. PCIe Cfg transactions can only be issued by the RC (to an EP): A RC cannot receive one.
(3) 3. PCIe IO transactions can only be issued by the RC (to an EP): A RC cannot receive one.

24.9.4.9.2 Double Mapping of the PCIe Local Control Registers

Each local programming register is mapped twice inside the controller's local target, with an offset of 0x1000 (4 KiB) between the two mappings.

The "low" mapping is the Data Interface (DIF) chip-select (CS) access. In DIF CS mode, some fields that are read-only (RO) over PCI become writable if PL bit PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN=0b1 (default). This access is used by the local host (e.g. BIOS) for the initialization of the PCI device, prior to PCI protocol startup (LTSSM enable). Fields and registers that are CS-sensitive are labeled (CS) in the Section 24.9.7.

The "high" mapping is the Data Interface (DIF) chip-select-2 (CS2) access. In DIF CS2 mode, some fields that are read-only (RO) over PCI or in DIF CS mode, become writable if PL bitfield

PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN=0b1 (default). This access is also used by the local host for further initialization of the PCI device. Fields and registers that are CS2-sensitive are labeled (CS2) in the tables below.

Note

The following important notes on accesses must be considered by user for device PCIe controller:

- If PCIe_SS_PL_CONF instance located bitfield PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN is cleared to 0b0, the "CS and CS2-sensitive" fields become read-only (RO) again, that is, they revert to the behaviour defined in the PCI/PCIe standards.
- The standard PCI configuration registers (header, descriptors, extended descriptors) are the only ones that can change their behaviour in CS and CS2 access.
- The port logic (PL) registers and the PCIe controller wrapper TI configuration registers are not affected by the CS: for those, the two mappings have the same properties, that is, they are perfect aliases
- For an RC, which by definition cannot access its own (locally stored) configuration registers over the PCIe wire, the DIF access is the only alternative. The PCIe controller must present a PCI-standard-compliant type-1 configuration space to the RC software driver, that is, with none of the read-only bits actually writable, as follows:
  – the DIF CS access should be used
  – PL bitfield PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN should be cleared to 0b0 (after a pre-initialization, if required)

24.9.4.9.3 Base Address Registers (BAR) Initialization

The 6 base address registers ((EP) BAR0 through (EP) BAR5) supported in EP mode (Type0), get discovered and configured by the RC during PCI enumeration, as per the PCIe standard. Via the PCIe ECAM (enhanced configuration access) mechanism the **RC** maps a remote EP function's BAR registers in 64-bit PCIe memory space and software run on RC access them as the PCIECTRL_EP_PCIEWIRE_BAR0 through PCIECTRL_EP_PCIEWIRE_BAR5 registers. Note that the configuration space of the RC (Type1 registers) also contains two BARs ((RC) BAR0 and (RC) BAR1), but they are not expected to be used.

Before enumeration, BARs can be initialized by the local software, using DIF access (CS or CS2), to modify the properties seen by the RC at enumeration. The following can be configured in each BAR: enabling, IO or memory type, size (mask width). Even BAR numbers (0,2,4) can also be paired with the BAR number above (1,3,5) to create a single, 64-bit address (memory).

Note

If PCIe_SS_PL_CONF instance located register PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN bit is cleared, most BAR fields become read-only again, behaving like in over-the-wire PCI Cfg writes by the RC. Only the BAR base address (MSbits) is still writable. The size/mask, type, and enabling become read-only.

Table 24-507. PCIe Default BAR Configuration

	Enabled(1)	Space(2) BARn[0]	Type(2) BARn[2:1]	Prefetch(2) BARn[3]	Mem Size(1) (M-bit mask)	I/O Size (M-bit mask)
(EP) BAR0	Yes	Memory (0)	32-bit (2'b00)	Yes (1)	1 MiB (20-bit)	256 Bytes (8-bit)
(EP) BAR1	Yes	Memory (0)	32-bit (2'b00)	Yes (1)	64 KiB (16-bit)	256 Bytes (8-bit)
(EP) BAR2	Yes	Memory (0)	32-bit (2'b00)	Yes (1)	1 MiB (20-bit)	256 Bytes (8-bit)
(EP) BAR3	Yes	Memory (0)	32-bit (2'b00)	Yes (1)	64 KiB (16-bit)	256 Bytes (8-bit)
(EP) BAR4	Yes	Memory (0)	32-bit (2'b00)	Yes (1)	4 KiB (12-bit)	256 Bytes (8-bit)
(EP) BAR5	Yes	Memory (0)	32-bit (2'b00)	Yes (1)	64 KiB (16-bit)	256 Bytes (8-bit)

(1) reprogrammable with DIF (CS2) access

(2) reprogrammable with DIF (CS) access

The Table 24-508 and Table 24-509 describe how a memory and IO BAR can be accessed, respectively. A single address of the PCIe config space is accessed for a given BAR, with each line of the table showing a different mode: read vs. write; PCIe Cfg access vs. DIF (CS) access vs. DIF (CS2) access.

"M" is the width of the BAR mask in bits, with 2^M the BAR size in bytes. Bits M and above are writable by the RC over the PCIe wire to set the BAR base address, as follows:

- The value of M for a Memory BAR varies from 4 (2^4=16 Byte = minimum size) to 32 (2^{32} = 4 GiB for a 32-bit BAR, potentially more for a 64-bit BAR) depending on the BAR and on the configuration.
- The current implementation forces M=8 for an I/O BAR, that is, an IO BAR has a fixed size of 2^8=256 bytes, the maximum allowed by the PCIe protocol.

Table 24-508. BARn Register Access Methods (Mem BAR)

Method	BAR[31:M][1]	BAR[M-1:4]	BAR[3]	BAR[2:1]	BAR[0]
PCI read	Base address readout	Read returns zero	Read out "Prefetchable"	Read out "Type"	Read out 0 "space decoder"[2]
PCI write	Base address write	No effect (masked = R/O)	No effect	No effect	No effect
CS read	Base address readout	Read returns zero (masked)	Read out "Prefetchable"	Read out "Type"	Read out 0 "space decoder"[2]
CS write	Base address write	No effect (masked = R/O)	Write "Prefetchable"	Write "Type"	Write "space decoder"[2]
CS2 read	Base address readout	Read returns zero (masked)	Read out "Prefetchable"	Read out "Type"	Read out 0 "space decoder"[2]
CS2 write	Write 0 to unmask (Sets M)	Write 1 to mask (Sets M)	Write 1	Write 1	Write "BAR enable"

(1) For M=32, the BAR mask is 32-bit wide, the BAR size is 4 GiB and the column disappears.
(2) Space decoder is Memory = 0 in this table.

Table 24-509. BARn Register Access Methods (IO BAR)

Method	BAR[31:8]	BAR[7:2]	BAR[1]	BAR[0]
PCI read	Base address readout	Read returns zero	Read out 0 (reserved bit)	Read out 1 "space decoder"[1]
PCI write	Base address write	No effect (masked = R/O)	No effect	No effect
CS read	Base address readout	Read returns zero (masked)	Read out 0 (reserved bit)	Read out 1 "space decoder"[1]
CS write	Base address write	No effect (masked = R/O)	Write 0 No effect	Write "space decoder"[1]
CS2 read	Base address readout	Read returns zero (masked)	Read out 0 (reserved)	Read out 1 "space decoder"[1]
CS2 write	Write 0. No effect in fixed-sized I/O BAR	Write 1. No effect in fixed-sized I/O BAR	Write 0 No effect	Write "BAR enable"

(1) Space decoder is I/O = 1 in this table.

24.9.5 PCIe Controller Low Level Programming Model

24.9.5.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the PCIe_SS1 and PCIe_SS2 controllers are used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the PCIe_SS (for more information, see PCIe Controllers Integration and PCIe Controllers Environment). Table 24-510 describes the global initialization of the surrounding modules.

Table 24-510. Global Initialization of the PCIe Surrounding Modules

Surrounding Modules	Comments
PRCM	Module functional and interface clocks must be enabled. (See *Clock Management Functional Description*, in *Power, Reset, and Clock Management*.)
Control Module	Module configuration must be done in control module. See *CTRL_MODULE_CORE Registers* and Table 24-495, *PCIe Port Configuration*
(Optional) IRQ_CROSSBAR	Interrupt crossbar configuration must be done to map the interrupts from the PCIe Controllers to a certain device processor INTC via the IRQ_CROSSBAR module. For more details on IRQ_CROSSBAR programming, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
(Optional) A local INTC	The local interrupt controller must be configured to enable the interrupt once IRQ output is mapped on INTC via the IRQ_CROSSBAR. See *Interrupt Controllers*.
L4 and L3_MAIN interconnects	For more information about the interconnect configuration, see *L3_MAIN Interconnect Agents*, and *L4 Interconnect Overview*, in *Interconnects*.
(Optional) Device MMU2	For more information on the configuration for routing the PCIe_SS1,2 master port traffic through the device MMU2, see Section 24.9.4.3.1.1, as well as Chapter 20, *Memory Management Units*.

Note

The interrupt configuration is required when using interrupt communication mode.

24.9.5.2 Main Sequence of PCIe Controllers Initalization

Note

This section aims to describe only the basic steps that must be followed by PCIe application software in order for a PCIe RC or EP type device to initialize, establish PCIe connectivity and interact with devices on the PCIe fabric. For this reason, it focuses on settings of the PCIe controllers wrapper registers (and not on PCI/PCIe standard and non-standard (PL) registers), through which PCIe application interacts with PCIe devices.

Table 24-511. Main Sequence PCIe Controller Global Initialization

RC and/or EP Initialization Step[1]	Register/Bit Field/Programming Model	Value
1. Initialize the DPLL_PCIe_REF (clock source, dividers, power dependencies) in PRCM	Refer to Section 26.4.5	
2. Assert a local software main reset to the PCIe_SS1,2 controllers	PRCM.RM_PCIESS_RSTCTRL[0] RST_LOCAL_PCIE1 PRCM.RM_PCIESS_RSTCTRL[1] RST_LOCAL_PCIE2	0b1
3. Poll main reset completion status for the PCIe_SS1,2 controllers	PRCM.RM_PCIESS_RSTST[0] RST_LOCAL_PCIE1 PRCM.RM_PCIESS_RSTST[1] RST_LOCAL_PCIE2	=0b1
4. Perform tuning of the PCIe PHY	Refer to Section 26.4.5	
IF: required operation mode for PCIe controller is “RC”	Software Test Condition	
5. Select RC type	PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE	0x4 (default)
6. Poll to insure a fundamental reset has completed	PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE	=0x4

Table 24-511. Main Sequence PCIe Controller Global Initialization (continued)

RC and/or EP Initialization Step[1]	Register/Bit Field/Programming Model	Value
7. Make sure LTSSM_EN is kept by RC's software at value 0b0	PCIECTRL_TI_CONF_DEVICE_CMD[0] LTSSM_EN	0b0
8. Initialize RC PCIe controller's core registers.	See Table 24-513	
9. Clear the IRQ main status register in case raw status is not clear (to avoid spurious interrupts)	PCIECTRL_TI_CONF_IRQSTATUS_MAIN[31:0]	0xFFFFFFFF
10. First time enable PCIe main interrupts. Depends on which PCIe protocol events and error events will be handled by application.	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN [14:0]	0x-
11. Clear the IRQ msi status register in case raw status is not clear (to avoid spurious interrupts)	PCIECTRL_TI_CONF_IRQSTATUS_MSI[31:0]	0xFFFFFFFF
12. [optional step] Enable INTx (x=A,B,C,D) interrupts reception (generated from legacy EPs to RC) but disable MSI interrupts reception at the same time. OR Enable the MSI interrupts reception (MSI interrupts are generated to RC from PCIe EPs only) but disable the INTx interrupts reception at the same time	PCIECTRL_TI_CONF_IRQENABLE_SET_MSI [3:0] INTx_EN	0x-
	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI [4] MSI_EN	0b1
	PCIECTRL_TI_CONF_IRQENABLE_SET_MSI [4] MSI_EN	0b1
	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI [3:0] INTx_EN	0xF
ELSE IF: required operation mode for PCIe controller is "PCIe EP"		
5. Select "PCIe EP" type	PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE	0x0
6. Poll to insure fundamental reset has completed	PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE	=0x0
7. Make sure LTSSM_EN is kept by EP's software at value 0b0	PCIECTRL_TI_CONF_DEVICE_CMD[0] LTSSM_EN	0b0
8. Initialize EP PCIe controller's core registers	See Table 24-512	
9. Clear the IRQ main status register in case raw status is not clear (to avoid spurious interrupts)	PCIECTRL_TI_CONF_IRQSTATUS_MAIN[31:0]	0xFFFFFFFF
10. First time enable –PCIe main interrupts. Depends on which PCIe protocol events and error events will be handled by application.	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN [14:0]	0x-
11. Clear the IRQ msi status register in case raw status is not clear (to avoid spurious interrupts)	PCIECTRL_TI_CONF_IRQSTATUS_MSI[31:0]	0xFFFFFFFF
12. Disable both INTx (x=A,B,C,D) and MSI interrupts reception.	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI [4:0]	0x1F
ELSE:		
5. Select "PCI Legacy EP " type	PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE	0x1
6. Poll to insure fundamental reset has completed	PCIECTRL_TI_CONF_DEVICE_TYPE[3:0] TYPE	=0x1
7. Make sure LTSSM_EN is kept by EP's software at value 0b0	PCIECTRL_TI_CONF_DEVICE_CMD[0] LTSSM_EN	0b0
8. Initialize EP controller local registers via DIF CS/CS2 accesses	See Table 24-512	
9. Clear the IRQ main status register in case raw status is not clear (to avoid spurious interrupts)	PCIECTRL_TI_CONF_IRQSTATUS_MAIN[31:0]	0xFFFFFFFF
10. First time enable PCIe main interrupts. Depends on which PCIe protocol events and error events will be handled by application.	PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN [14:0]	0x-
11. Clear the IRQ main status register in case raw status is not clear (to avoid spurious interrupts)	PCIECTRL_TI_CONF_IRQSTATUS_MSI[31:0]	0xFFFFFFFF
12. Disable both INTx (x=A,B,C,D) and MSI interrupts reception	PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI [4:0]	0x1F
ENDIF		

Table 24-511. Main Sequence PCIe Controller Global Initialization (continued)

RC and/or EP Initialization Step[(1)]	Register/Bit Field/Programming Model	Value
13. PCIe (RC/EP) controller enables the link. It is supposed that all active EPs attached on the PCIe fabric synchronously enable their LTSSM engine with the RC.	PCIECTRL_TI_CONF_DEVICE_CMD[0] LTSSM_EN	0b1
14. Local CPU polls until the PCIe Controller link is up (that is, brought into L0- power state)	PCIECTRL_TI_CONF_PHY_CS[16] LINK_UP	=0b1
15. If PCIECTRL_TI_CONF_IRQSTATUS_MAIN[3] LINK_UP_EVT has been enabled in step (10), this means it should be cleared	PCIECTRL_TI_CONF_IRQSTATUS_MAIN[12] LINK_UP_EVT	0b1
16. RC host starts link training sequence, negotiation with EPs, enumeration of the EPs, and initializes the descriptors per each of the discovered EP functions. Example: RC Initializes the MSI message vectors table per each of the EPs. RC maps the EPs global PCIe Memory per each EP function appropriately initializing the BARn registers MSB part.		
17. [RC only] **IF:** an EP is a requester, the RC remotely enables its BME bit via an ECAM access to the EP's command /status register.	PCIe_SS1_EP_CFG_PCIe. PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_R EGISTER[2] BUSMASTER_EN	0b1
18. [RC only] **IF:** an EP is a completer, the RC remotely enables its MSE (for Mem transfers) or ISE bits (for legacy I/O transfers) via an ECAM access to the EP's command/status register.	PCIe_SS1_EP_CFG_PCIe. PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_R EGISTER[1] MEM_SPACE_EN OR PCIe_SS1_EP_CFG_PCIe. PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_R EGISTER[0] IO_SPACE_EN	0b1
19. [EP only] Before EP's application generates a request (that is, requester EP), the EP application must insure that its local BME_EVT status has already been asserted to 0b1 by the RC host controller.	PCIECTRL_TI_CONF_IRQSTATUS_MAIN[13] BME_EVT_MSG OR PCIECTRL_TI_CONF_IRQSTATUS_RAW_MAIN[13] BME_EVT_MSG	=0b1
The EP must locally clear the BME interrupt, writing BME_EVT_MSG flag to 0b1 (Wr1toClr).	Wr1ToClr PCIECTRL_TI_CONF_IRQSTATUS_MAIN[13] BME_EVT_MSG	0b1
20. [RC only] Before RC application initiates a request to an completer EP function with a *Memory type BAR*, the RC must check that EP's MEM_SPACE_EN flag has already been asserted by the RC host. This is done via an ECAM remote access by RC to EP's command/status register. [EP Only] The EP's CPU is notified about remote MEM_SPACE_EN flag assertion in a dedicated bit that can be enabled to generate an interrupt. The interrupt flag is cleared via writing to 0b1 (Wr1ToClr).	PCIe_SS1,2_EP_CFG_PCIe. PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_R EGISTER[1] MEM_SPACE_EN	=0b1
	The EP application may receive an interrupt (if enabled – see step 10) indicated in the PCIECTRL_TI_CONF_IRQSTATUS_MAIN[14] MSE_EVT_MSG. Or poll this in PCIECTRL_TI_CONF_IRQSTATUS_RAW_MAIN[14] MSE_EVT_MSG.	0b1
20. [RC only] Before RC application initiates a request to a completer EP function with an *I/O type BAR*, the RC must check that EP's IO_SPACE_EN flag has already been asserted by the RC host. This is done via an ECAM remote read access by RC to EP's command/status register.	PCIe_SS1,2_EP_CFG_PCIe. PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_R EGISTER[0] IO_SPACE_EN	=0b1
[EP Only] Unlike Mem type EPs, there is no interrupt event to notify the EP about its ISE flag being asserted by the RC. So, to be aware of an incoming transaction, I/O legacy EP must poll local IO_SPACE_EN flag until its asserted by RC to 0b1.	PCIECTRL_EP_DBICS_STATUS_COMMAND_REGIS TER[0] IO_SPACE_EN	=0b1

Table 24-511. Main Sequence PCIe Controller Global Initialization (continued)

RC and/or EP Initialization Step(1)	Register/Bit Field/Programming Model	Value
[EP Only] 21. IF IO_SPACE_EN bit has been set to '1' for a legacy EP, user must force the legacy EP master port out-of-standby (before first transaction). There is no automatical standby generation for I/O transfers.	PCIECTRL_TI_CONF_SYSCONFIG[5:4] STANDBYMODE	0x1

(1) Unless a step is not marked as "[EP Only]" or "[RC Only]", the initialization step should be considered as applicable in both EP and RC modes of PCIe controller.

Table 24-512. Local EP (Type 0) PCIe Standard and Port Logic Registers Initialization Subsequence

8.1 Initialize PCI standard configuration header registers via corresponding DIF CS/CS2 accesses. Example: BAR MASK0 – BAR MASK5 (that must be entirely initialized by the EP, and LSB part of the BAR0 – BAR5 registers)	For more details on these settings, refer to the *PCI Express Base 3.0 Specification, revision 1.0*	
8.2 Initialize the RC PCIe capability registers via corresponding DIF CS/CS2 accesses.	For more details on these settings, refer to the *PCI Express Base 3.0 Specification, revision 1.0*	
8.3 Initialize the RC PCIe Port Logic registers. Example: iATU settings of internal local address ranges, link settings – link powers, timings, and so forth		

Table 24-513. RC (Type 1) PCIe Standard and Port Logic Registers Initialization Subsequence

RC must enable write operation to be able to configure its own PCIe standard and PL registers.	PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN	0b1
8. 1 Initialize RC PCI standard configuration header registers via corresponding DIF CS/CS2 accesses. (Example: RC BAR 0,1 and BAR MASK0,1 registers.)	For more details on these settings, refer to the *PCI Express Base 3.0 Specification, revision 1.0*	
8.2 Initialize the RC PCIe capability registers via corresponding DIF CS/CS2 accesses.	For more details on these settings, refer to the *PCI Express Base 3.0 Specification, revision 1.0*	
8.3 Initialize the RC PCIe Port Logic registers. Example: iATU settings of internal local address ranges, link settings – link power, timings, and so forth Prepare the MSI vectors, Setup of the MSI, others.		
Disable write operation in RC PCIe standard and PL registers.	PCIECTRL_PL_DBI_RO_WR_EN[0] CX_DBI_RO_WR_EN	0b0

Note

When the PCIe_SS1 controller is configured to operate in two lane mode, by default lane 0 is mapped to the PCIe1_PHY (device PCIe PHY port 0), and lane 1 is mapped to the PCIe2_PHY (device PCIe PHY port 1). The PCIe_SS1_TI_CONF wrapper located software control - PCIECTRL_TI_CONF_PHY_CS[0] REVERSE_LANES allows user software to swap mapping of the PCIe_SS1 controller lane 0 and lane 1 on the two device PCIe PHY ports (swapping the RX and TX paths simultaneously with polarity of an individual lane unchanged). Note that the identical control is not functional in the PCIe_SS2_TI_CONF, because the PCIe_SS2 can operate in a single lane mode only, and (if selected) its lane 0 is always mapped to the PCIe PHY port 1 (that is, PCIe2_PHY).

24.9.6 PCIe Standard Registers vs PCIe Subsystem Hardware Registers Mapping

The below tables provide the PCIe/PCI standard logical registers versus PCIe controller hardware registers mapping in the device.

Table 24-514. PCIe Type-0 (EP) Configuration Standard Capability Registers vs PCIe Controller Hardware Registers Mapping

PCIe Standard Register Name	PCIe_SS1_EP_CFG_PCIe and PCIe_SS2_EP_CFG_PCIe Corresponding Register(1)	PCIe_SS1_EP_CFG_DBICS and PCIe_SS2_EP_CFG_DBICS Corresponding Register(2)	PCIe_SS1_EP_CFG_DBICS2 and PCIe_SS2_EP_CFG_DBICS2 Corresponding Register(3)
PCIE_CAP	PCIECTRL_EP_PCIEWIRE_PCIE_CAP	PCIECTRL_EP_DBICS_PCIE_CAP	PCIECTRL_EP_DBICS2_PCIE_CAP
DEV_CAP	PCIECTRL_EP_PCIEWIRE_DEV_CAP	PCIECTRL_EP_DBICS_DEV_CAP	PCIECTRL_EP_DBICS2_DEV_CAP
DEV_CAS	PCIECTRL_EP_PCIEWIRE_DEV_CAS	PCIECTRL_EP_DBICS_DEV_CAS	PCIECTRL_EP_DBICS2_DEV_CAS
LNK_CAP	PCIECTRL_EP_PCIEWIRE_LNK_CAP	PCIECTRL_EP_DBICS_LNK_CAP	PCIECTRL_EP_DBICS2_LNK_CAP
LNK_CAS	PCIECTRL_EP_PCIEWIRE_LNK_CAS	PCIECTRL_EP_DBICS_LNK_CAS	PCIECTRL_EP_DBICS2_LNK_CAS
DEV_CAP_2	PCIECTRL_EP_PCIEWIRE_DEV_CAP_2	PCIECTRL_EP_DBICS_DEV_CAP_2	PCIECTRL_EP_DBICS2_DEV_CAP_2
DEV_CAS_2	PCIECTRL_EP_PCIEWIRE_DEV_CAS_2	PCIECTRL_EP_DBICS_DEV_CAS_2	PCIECTRL_EP_DBICS2_DEV_CAS_2
LNK_CAP_2	PCIECTRL_EP_PCIEWIRE_LNK_CAP_2	PCIECTRL_EP_DBICS_LNK_CAP_2	PCIECTRL_EP_DBICS2_LNK_CAP_2
LNK_CAS_2	PCIECTRL_EP_PCIEWIRE_LNK_CAS_2	PCIECTRL_EP_DBICS_LNK_CAS_2	PCIECTRL_EP_DBICS2_LNK_CAS_2

(1) PCIe (EP) register mapped in the ECAM Cfg Space
(2) PCIe (EP) register accessible in the controller local DIF CS space
(3) PCIe (EP) register accessible in the controller local DIF CS2 space

Table 24-515. PCIe Type-0 (EP) Standard Configuration Header Registers vs PCIe Controller Hardware Registers Mapping

PCIe Standard Register Name	PCIe_SS1_EP_CFG_PCIe and PCIe_SS2_EP_CFG_PCIe Corresponding Register(1)	PCIe_SS1_EP_CFG_DBICS and PCIe_SS2_EP_CFG_DBICS Corresponding Register(2)	PCIe_SS1_EP_CFG_DBICS2 and PCIe_SS2_EP_CFG_DBICS2 Corresponding Register(3)
DEVICE_VENDORID	PCIECTRL_EP_PCIEWIRE_DEVICE_VENDORID	PCIECTRL_EP_DBICS_DEVICE_VENDORID	PCIECTRL_EP_DBICS2_DEVICE_VENDORID
STATUS_COMMAND_REGISTER	PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_REGISTER	PCIECTRL_EP_DBICS_STATUS_COMMAND_REGISTER	PCIECTRL_EP_DBICS2_STATUS_COMMAND_REGISTER
CLASSCODE_REVISIONID	PCIECTRL_EP_PCIEWIRE_CLASSCODE_REVISIONID	PCIECTRL_EP_DBICS_CLASSCODE_REVISIONID	PCIECTRL_EP_DBICS2_CLASSCODE_REVISIONID
BIST_HEAD_LAT_CACH	PCIECTRL_EP_PCIEWIRE_BIST_HEAD_LAT_CACH	PCIECTRL_EP_DBICS_BIST_HEAD_LAT_CACH	PCIECTRL_EP_DBICS2_BIST_HEAD_LAT_CACH
BAR0/BAR0_MASK	PCIECTRL_EP_PCIEWIRE_BAR0	PCIECTRL_EP_DBICS_BAR0	PCIECTRL_EP_DBICS2_BAR0_MASK
BAR1/BAR1_MASK	PCIECTRL_EP_PCIEWIRE_BAR1	PCIECTRL_EP_DBICS_BAR1	PCIECTRL_EP_DBICS2_BAR1_MASK
BAR2/BAR2_MASK	PCIECTRL_EP_PCIEWIRE_BAR2	PCIECTRL_EP_DBICS_BAR2	PCIECTRL_EP_DBICS2_BAR2_MASK
BAR3/BAR3_MASK	PCIECTRL_EP_PCIEWIRE_BAR3	PCIECTRL_EP_DBICS_BAR3	PCIECTRL_EP_DBICS2_BAR3_MASK
BAR4/BAR4_MASK	PCIECTRL_EP_PCIEWIRE_BAR4	PCIECTRL_EP_DBICS_BAR4	PCIECTRL_EP_DBICS2_BAR4_MASK
BAR5/BAR5_MASK	PCIECTRL_EP_PCIEWIRE_BAR5	PCIECTRL_EP_DBICS_BAR5	PCIECTRL_EP_DBICS2_BAR5_MASK
CARDBUS_CIS_POINTER	PCIECTRL_EP_PCIEWIRE_CARDBUS_CIS_POINTER	PCIECTRL_EP_DBICS_CARDBUS_CIS_POINTER	PCIECTRL_EP_DBICS2_CARDBUS_CIS_POINTER
SUBID_SUBVENDORID	PCIECTRL_EP_PCIEWIRE_SUBID_SUBVENDORID	PCIECTRL_EP_DBICS_SUBID_SUBVENDORID	PCIECTRL_EP_DBICS2_SUBID_SUBVENDORID
EXPANSION_ROM_BAR	PCIECTRL_EP_PCIEWIRE_EXPANSION_ROM_BAR	PCIECTRL_EP_DBICS_EXPANSION_ROM_BAR	PCIECTRL_EP_DBICS2_EXPANSION_ROM_BAR
CAPPTR	PCIECTRL_EP_PCIEWIRE_CAPPTR	PCIECTRL_EP_DBICS_CAPPTR	PCIECTRL_EP_DBICS2_CAPPTR
INTERRUPT	PCIECTRL_EP_PCIEWIRE_INTERRUPT	PCIECTRL_EP_DBICS_INTERRUPT	PCIECTRL_EP_DBICS2_INTERRUPT
PM_CAP	PCIECTRL_EP_PCIEWIRE_PM_CAP	PCIECTRL_EP_DBICS_PM_CAP	PCIECTRL_EP_DBICS2_PM_CAP
PM_CSR	PCIECTRL_EP_PCIEWIRE_PM_CSR	PCIECTRL_EP_DBICS_PM_CSR	PCIECTRL_EP_DBICS2_PM_CSR

(1) PCIe (EP) register mapped in the ECAM Cfg Space
(2) PCIe (EP) register accessible in the controller local DIF CS space
(3) PCIe (EP) register accessible in the controller local DIF CS2 space

Table 24-516. PCIe Type-1 (RC) Configuration Standard Capability Registers vs PCIe Controller Hardware Registers Mapping

PCIe EP Standard Register Name	PCIe_SS1_RC_CFG_DBICS and PCIe_SS2_RC_CFG_DBICS Corresponding Register(1)	PCIe_SS1_RC_CFG_DBICS2 and PCIe_SS2_RC_CFG_DBICS2 Corresponding Register(2)
PCIE_CAP	PCIECTRL_RC_DBICS_PCIE_CAP	PCIECTRL_RC_DBICS2_PCIE_CAP
DEV_CAP	PCIECTRL_RC_DBICS_DEV_CAP	PCIECTRL_RC_DBICS2_DEV_CAP
DEV_CAS	PCIECTRL_RC_DBICS_DEV_CAS	PCIECTRL_RC_DBICS2_DEV_CAS
LNK_CAP	PCIECTRL_RC_DBICS_LNK_CAP	PCIECTRL_RC_DBICS2_LNK_CAP
LNK_CAS	PCIECTRL_RC_DBICS_LNK_CAS	PCIECTRL_RC_DBICS2_LNK_CAS
SLOT_CAP	PCIECTRL_RC_DBICS_SLOT_CAP	PCIECTRL_RC_DBICS2_SLOT_CAP
SLOT_CAS	PCIECTRL_RC_DBICS_SLOT_CAS	PCIECTRL_RC_DBICS2_SLOT_CAS
ROOT_CAC	PCIECTRL_RC_DBICS_ROOT_CAC	PCIECTRL_RC_DBICS2_ROOT_CAC
ROOT_STS	PCIECTRL_RC_DBICS_ROOT_STS	PCIECTRL_RC_DBICS2_ROOT_STS
DEV_CAP_2	PCIECTRL_RC_DBICS_DEV_CAP_2	PCIECTRL_RC_DBICS2_DEV_CAP_2
DEV_CAS_2	PCIECTRL_RC_DBICS_DEV_CAS_2	PCIECTRL_RC_DBICS2_DEV_CAS_2
LNK_CAP_2	PCIECTRL_RC_DBICS_LNK_CAP_2	PCIECTRL_RC_DBICS2_LNK_CAP_2
LNK_CAS_2	PCIECTRL_RC_DBICS_LNK_CAS_2	PCIECTRL_RC_DBICS2_LNK_CAS_2

(1) PCIe (RC) register accessible in the controller local DIF CS space
(2) PCIe (RC) register accessible in the controller local DIF CS2 space

Table 24-517. PCIe Type-1 (RC) Standard Configuration Header Registers vs PCIe Controller Hardware Registers Mapping

PCIe Standard Register Name	PCIe_SS1_RC_CFG_DBICS and PCIe_SS2_RC_CFG_DBICS Corresponding Register(1)	PCIe_SS1_RC_CFG_DBICS2 and PCIe_SS2_RC_CFG_DBICS2 Corresponding Register(2)
DEVICE_VENDORID	PCIECTRL_RC_DBICS_DEVICE_VENDORID	PCIECTRL_RC_DBICS2_DEVICE_VENDORID
STATUS_COMMAND_REGISTER	PCIECTRL_RC_DBICS_STATUS_COMMAND_REGISTER	PCIECTRL_RC_DBICS2_STATUS_COMMAND_REGISTER
CLASSCODE_REVISIONID	PCIECTRL_RC_DBICS_CLASSCODE_REVISIONID	PCIECTRL_RC_DBICS2_CLASSCODE_REVISIONID
BIST_HEAD_LAT_CACH	PCIECTRL_RC_DBICS_BIST_HEAD_LAT_CACH	PCIECTRL_RC_DBICS2_BIST_HEAD_LAT_CACH
BAR0/BAR0_MASK	PCIECTRL_RC_DBICS_BAR0	PCIECTRL_RC_DBICS2_BAR0_MASK
BAR1/BAR1_MASK	PCIECTRL_RC_DBICS_BAR1	PCIECTRL_RC_DBICS2_BAR1_MASK
BUS_NUM_REG	PCIECTRL_RC_DBICS_BUS_NUM_REG	PCIECTRL_RC_DBICS2_BUS_NUM_REG
IOBASE_LIMIT_SEC_STATUS	PCIECTRL_RC_DBICS_IOBASE_LIMIT_SEC_STATUS	PCIECTRL_RC_DBICS2_IOBASE_LIMIT_SEC_STATUS
MEM_BASE_LIMIT	PCIECTRL_RC_DBICS_MEM_BASE_LIMIT	PCIECTRL_RC_DBICS2_MEM_BASE_LIMIT
PREF_MEM_BASE_LIMIT	PCIECTRL_RC_DBICS_PREF_MEM_BASE_LIMIT	PCIECTRL_RC_DBICS2_PREF_MEM_BASE_LIMIT
UPPER_32BIT_PREF_BASEADDR	PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_BASEADDR	PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_BASEADDR
UPPER_32BIT_PREF_LIMITADDR	PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_LIMITADDR	PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_LIMITADDR
IO_BASE_LIMIT	PCIECTRL_RC_DBICS_IO_BASE_LIMIT	PCIECTRL_RC_DBICS2_IO_BASE_LIMIT
CAPPTR	PCIECTRL_RC_DBICS_CAPPTR	PCIECTRL_RC_DBICS2_CAPPTR
EXPANSION_ROM_BAR	PCIECTRL_RC_DBICS_EXPANSION_ROM_BAR	PCIECTRL_RC_DBICS2_EXPANSION_ROM_BAR
BRIDGE_INT	PCIECTRL_RC_DBICS_BRIDGE_INT	PCIECTRL_RC_DBICS2_BRIDGE_INT

(1) PCIe (RC) register accessible in the controller local DIF CS space
(2) PCIe (RC) register accessible in the controller local DIF CS2 space

24.9.7 PCIe Controller Register Manual

24.9.7.1 PCIe Controller Instance Summary

Table 24-518. PCIe_SS1 Instance Summary

Module Name	Module Base Address	Size
PCIe_SS1_EP_CFG_PCIe	ECAM_Param_Base_Addr(1) + 0x2000 0000	4 KiB
PCIe_SS1_EP_CFG_DBICS	0x5100 0000	128 Bytes
PCIe_SS1_RC_CFG_DBICS	0x5100 0000	128 Bytes
PCIe_SS1_PL_CONF	0x5100 0700	500 Bytes
PCIe_SS1_EP_CFG_DBICS2	0x5100 1000	128 Bytes
PCIe_SS1_RC_CFG_DBICS2	0x5100 1000	128 Bytes
PCIe_SS1_TI_CONF	0x5100 2000	332 Bytes

(1) ECAM_Param_Base_Addr = 0x0000_0000 to 0x0FFF_F000

Table 24-519. PCIe_SS2 Instance Summary

Module Name	Module Base Address	Size
PCIe_SS2_EP_CFG_PCIe	ECAM_Param_Base_Addr(1) + 0x3000 0000	4 KiB
PCIe_SS2_EP_CFG_DBICS	0x5180 0000	128 Bytes
PCIe_SS2_RC_CFG_DBICS	0x5180 0000	128 Bytes
PCIe_SS2_PL_CONF	0x5180 0700	500 Bytes
PCIe_SS2_EP_CFG_DBICS2	0x5180 1000	128 Bytes
PCIe_SS2_RC_CFG_DBICS2	0x5180 1000	128 Bytes
PCIe_SS2_TI_CONF	0x5180 2000	332 Bytes

(1) ECAM_Param_Base_Addr = 0x0000_0000 to 0x0FFF_F000

Note

The *ECAM_Param_Base_Addr* variable is the 16-bit **base address** (this address is 4 KiB-aligned with value in the range: 0x0000_0000 - 0x0FFF_F000) of the relevant EP function descriptor (relevant EP's function PCIe standard and PL configuration registers) **in a contiguous 256-MiB ECAM space** (0x0000_0000 - 0x0FFF_FFFF). For more information on the PCIe ECAM configuration space mapping mechanism, refer to the section *PCI Express Enhanced Configuration Access Mechanism (ECAM)*, in the *PCI Express Base 3.0 Specification, Revision 1.0*.

For information on the EP's function PL register offsets and descriptions, refer to the Section 24.9.7.5.

24.9.7.2 PCIe_SS_EP_CFG_PCIe Registers

Note

This section describes the PCIe EP mode (PCIe type-0) standard configuration registers as they are accessed over PCIe wire logic and not via DIF CS/CS2 accesses. These register names are prefixed with "PCIECTRL_EP_PCIEWIRE".

24.9.7.2.1 PCIe_SS_EP_CFG_PCIe Register Summary

Table 24-520. PCIe_SS1_EP_CFG_PCIe Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_EP_CFG_PCIe Physical Address
PCIECTRL_EP_PCIEWIRE_DEVICE_VENDORID	R	32	0x0	ECAM_Param_Base_Addr(1) + 0x2000 0000

Table 24-520. PCIe_SS1_EP_CFG_PCIe Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_EP_CFG_PCIe Physical Address
PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_REGISTER	RW	32	0x4	ECAM_Param_Base_Addr(1) + 0x2000 0004
PCIECTRL_EP_PCIEWIRE_CLASSCODE_REVISIONID	R	32	0x8	ECAM_Param_Base_Addr (1) + 0x2000 0008
PCIECTRL_EP_PCIEWIRE_BIST_HEAD_LAT_CACH	RW	32	0xC	ECAM_Param_Base_Addr(1) + 0x2000 000C
PCIECTRL_EP_PCIEWIRE_BAR0	RW	32	0x10	ECAM_Param_Base_Addr(1) + 0x2000 0010
PCIECTRL_EP_PCIEWIRE_BAR1	RW	32	0x14	ECAM_Param_Base_Addr(1) + 0x2000 0014
PCIECTRL_EP_PCIEWIRE_BAR2	RW	32	0x18	ECAM_Param_Base_Addr(1) + 0x2000 0018
PCIECTRL_EP_PCIEWIRE_BAR3	RW	32	0x1C	ECAM_Param_Base_Addr(1) + 0x2000 001C
PCIECTRL_EP_PCIEWIRE_BAR4	RW	32	0x20	ECAM_Param_Base_Addr(1) + 0x2000 0020
PCIECTRL_EP_PCIEWIRE_BAR5	RW	32	0x24	ECAM_Param_Base_Addr(1) + 0x2000 0024
PCIECTRL_EP_PCIEWIRE_CARDBUS_CIS_POINTER	R	32	0x28	ECAM_Param_Base_Addr(1) + 0x2000 0028
PCIECTRL_EP_PCIEWIRE_SUBID_SUBVENDORID	R	32	0x2C	ECAM_Param_Base_Addr(1) + 0x2000 002C
PCIECTRL_EP_PCIEWIRE_EXPANSION_ROM_BAR	RW	32	0x30	ECAM_Param_Base_Addr(1) + 0x2000 0030
PCIECTRL_EP_PCIEWIRE_CAPPTR	R	32	0x34	ECAM_Param_Base_Addr(1) + 0x2000 0034
PCIECTRL_EP_PCIEWIRE_INTERRUPT	RW	32	0x3C	ECAM_Param_Base_Addr(1) + 0x2000 003C
PCIECTRL_EP_PCIEWIRE_PM_CAP	R	32	0x40	ECAM_Param_Base_Addr(1) + 0x2000 0040
PCIECTRL_EP_PCIEWIRE_PM_CSR	RW	32	0x44	ECAM_Param_Base_Addr(1) + 0x2000 0044
PCIECTRL_EP_PCIEWIRE_PCIE_CAP	R	32	0x70	ECAM_Param_Base_Addr(1) + 0x2000 0070
PCIECTRL_EP_PCIEWIRE_DEV_CAP	R	32	0x74	ECAM_Param_Base_Addr(1) + 0x2000 0074
PCIECTRL_EP_PCIEWIRE_DEV_CAS	RW	32	0x78	ECAM_Param_Base_Addr(1) + 0x2000 0078
PCIECTRL_EP_PCIEWIRE_LNK_CAP	R	32	0x7C	ECAM_Param_Base_Addr(1) + 0x2000 007C
PCIECTRL_EP_PCIEWIRE_LNK_CAS	RW	32	0x80	ECAM_Param_Base_Addr(1) + 0x2000 0080
PCIECTRL_EP_PCIEWIRE_DEV_CAP_2	R	32	0x94	ECAM_Param_Base_Addr(1) + 0x2000 0094
PCIECTRL_EP_PCIEWIRE_DEV_CAS_2	RW	32	0x98	ECAM_Param_Base_Addr(1) + 0x2000 0098
PCIECTRL_EP_PCIEWIRE_LNK_CAP_2	R	32	0x9C	ECAM_Param_Base_Addr(1) + 0x2000 009C
PCIECTRL_EP_PCIEWIRE_LNK_CAS_2	RW	32	0xA0	ECAM_Param_Base_Addr(1) + 0x2000 00A0

(1) ECAM_Param_Base_Addr = 0x0000_0000 to 0x0FFF_F000

Table 24-521. PCIe_SS2_EP_CFG_PCIe Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_EP_CFG_PCIe Physical Address
PCIECTRL_EP_PCIEWIRE_DEVICE_VENDORID	R	32	0x0	ECAM_Param_Base_Addr(1) + 0x3000 0000
PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_REGISTER	RW	32	0x4	ECAM_Param_Base_Addr(1) + 0x3000 0004
PCIECTRL_EP_PCIEWIRE_CLASSCODE_REVISIONID	R	32	0x8	ECAM_Param_Base_Addr(1) + 0x3000 0008
PCIECTRL_EP_PCIEWIRE_BIST_HEAD_LAT_CACH	RW	32	0xC	ECAM_Param_Base_Addr(1) + 0x3000 000C
PCIECTRL_EP_PCIEWIRE_BAR0	RW	32	0x10	ECAM_Param_Base_Addr(1) + 0x3000 0010
PCIECTRL_EP_PCIEWIRE_BAR1	RW	32	0x14	ECAM_Param_Base_Addr(1) + 0x3000 0014
PCIECTRL_EP_PCIEWIRE_BAR2	RW	32	0x18	ECAM_Param_Base_Addr(1) + 0x3000 0018
PCIECTRL_EP_PCIEWIRE_BAR3	RW	32	0x1C	ECAM_Param_Base_Addr(1) + 0x3000 001C
PCIECTRL_EP_PCIEWIRE_BAR4	RW	32	0x20	ECAM_Param_Base_Addr(1) + 0x3000 0020
PCIECTRL_EP_PCIEWIRE_BAR5	RW	32	0x24	ECAM_Param_Base_Addr(1) + 0x3000 0024
PCIECTRL_EP_PCIEWIRE_CARDBUS_CIS_POINTER	R	32	0x28	ECAM_Param_Base_Addr(1) + 0x3000 0028
PCIECTRL_EP_PCIEWIRE_SUBID_SUBVENDORID	R	32	0x2C	ECAM_Param_Base_Addr(1) + 0x3000 002C
PCIECTRL_EP_PCIEWIRE_EXPANSION_ROM_BAR	RW	32	0x30	ECAM_Param_Base_Addr(1) + 0x3000 0030
PCIECTRL_EP_PCIEWIRE_CAPPTR	R	32	0x34	ECAM_Param_Base_Addr(1) + 0x3000 0034
PCIECTRL_EP_PCIEWIRE_INTERRUPT	RW	32	0x3C	ECAM_Param_Base_Addr(1) + 0x3000 003C
PCIECTRL_EP_PCIEWIRE_PM_CAP	R	32	0x40	ECAM_Param_Base_Addr(1) + 0x3000 0040
PCIECTRL_EP_PCIEWIRE_PM_CSR	RW	32	0x44	ECAM_Param_Base_Addr(1) + 0x3000 0044
PCIECTRL_EP_PCIEWIRE_PCIE_CAP	R	32	0x70	ECAM_Param_Base_Addr(1) + 0x3000 0070
PCIECTRL_EP_PCIEWIRE_DEV_CAP	R	32	0x74	ECAM_Param_Base_Addr(1) + 0x3000 0074
PCIECTRL_EP_PCIEWIRE_DEV_CAS	RW	32	0x78	ECAM_Param_Base_Addr(1) + 0x3000 0078
PCIECTRL_EP_PCIEWIRE_LNK_CAP	R	32	0x7C	ECAM_Param_Base_Addr(1) + 0x3000 007C
PCIECTRL_EP_PCIEWIRE_LNK_CAS	RW	32	0x80	ECAM_Param_Base_Addr(1) + 0x3000 0080
PCIECTRL_EP_PCIEWIRE_DEV_CAP_2	R	32	0x94	ECAM_Param_Base_Addr(1) + 0x3000 0094
PCIECTRL_EP_PCIEWIRE_DEV_CAS_2	RW	32	0x98	ECAM_Param_Base_Addr(1) + 0x3000 0098
PCIECTRL_EP_PCIEWIRE_LNK_CAP_2	R	32	0x9C	ECAM_Param_Base_Addr(1) + 0x3000 009C
PCIECTRL_EP_PCIEWIRE_LNK_CAS_2	RW	32	0xA0	ECAM_Param_Base_Addr(1) + 0x3000 00A0

(1) ECAM_Param_Base_Addr = 0x0000_0000 to 0x0FFF_F000

Note

The *ECAM_Param_Base_Addr* variable is the 16-bit base address (this address is 4 KiB-aligned with value in the range: 0x0000_0000 - 0x0FFF_F000) of the relevant EP function descriptor (relevant EP's function PCIe standard and PL configuration registers) in a contiguous 256-MiB ECAM space (0x0000_0000 - 0x0FFF_FFFF). For more information on the PCIe ECAM configuration space mapping mechanism, refer to the section *PCI Express Enhanced Configuration Access Mechanism (ECAM)*, in the *PCI Express Base 3.0 Specification, Revision 1.0*.

For information on the EP's function PL register offsets and descriptions, refer to *PCIe_SS_PL_CONF Registers*.

24.9.7.2.2 PCIe_SS_EP_CFG_PCIe Register Description

Table 24-522. PCIECTRL_EP_PCIEWIRE_DEVICE_VENDORID

Address offset	0x0		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0000	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Device and Vendor ID		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVICEID																VENDORID															

Bits	Field Name	Description	Type	Reset
31:16	DEVICEID	Device ID (CS)	R	0x8888
15:0	VENDORID	Vendor ID (CS)	R	0x104C

Table 24-523. PCIECTRL_EP_PCIEWIRE_STATUS_COMMAND_REGISTER

Address offset	0x4		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0004	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Status and Command registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DETECT_PARERR	SIGNAL_SYSERR	RCVD_MASTERABORT	RCVD_TRGTABORT	SIGNAL_TRGTABORT	DEVSEL_TIME		MASTER_DATA_PARERR	FAST_B2B	RESERVED	C66MHZ_CAP	CAP_LIST	INTX_STATUS	RESERVED								INTX_ASSER_DIS	FAST_BEN	SERR_EN	IDSEL_CTRL	PARITYERR_RESP	VGA_SNOOP	MEMWR_INVA	SPEC_CYCLE_EN	BUSMASTER_EN	MEM_SPACE_EN	IO_SPACE_EN

Bits	Field Name	Description	Type	Reset
31	DETECT_PARERR	Detected Parity Error	RW	0x0
30	SIGNAL_SYSERR	Signaled System Error	RW	0x0
29	RCVD_MASTERABORT	Received Master Abort	RW	0x0
28	RCVD_TRGTABORT	Received Target Abort	RW	0x0
27	SIGNAL_TRGTABORT	Signaled Target Abort	RW	0x0
26:25	DEVSEL_TIME	DevSel Timing, Harsdwired to 0 for PCIExpress	R	0x0

Bits	Field Name	Description	Type	Reset
24	MASTERDATA_PARERR	Master Data Parity Error	RW	0x0
23	FAST_B2B	Back to Back Capable, Harsdwired to 0 for PCIExpress	R	0x0
22	RESERVED	Reserved	R	0x0
21	C66MHZ_CAP	66MHz Capable, Harsdwired to 0 for PCIExpress	R	0x0
20	CAP_LIST	Capabilities List Hardwired to 1	R	0x1
19	INTX_STATUS	INTx Status	R	0x0
18:11	RESERVED		R	0x0
10	INTX_ASSER_DIS	INTx Assertion Disable	RW	0x0
9	FAST_BBEN	Bit hardwired to 0 for PCIExpress	R	0x0
8	SERR_EN	SERR Enable	RW	0x0
7	IDSEL_CTRL	Bit hardwired to 0 for PCIExpress	R	0x0
6	PARITYERRRESP	Parity Error Response	RW	0x0
5	VGA_SNOOP	Not Applicable forPCI Express Bit hardwired to 0 for PCIExpress	R	0x0
4	MEMWR_INVA	Not Applicable for PCI Express Bit hardwired to 0 for PCIExpress	R	0x0
3	SPEC_CYCLE_EN	Not Applicable for PCI Express Bit hardwired to 0 for PCIExpress	R	0x0
2	BUSMASTER_EN	Bus Master Enable	RW	0x0
1	MEM_SPACE_EN	Memory Space Enable	RW	0x0
0	IO_SPACE_EN	IO Space Enable	RW	0x0

Table 24-524. PCIECTRL_EP_PCIEWIRE_CLASSCODE_REVISIONID

Address offset	0x8		
Physical Address	ECAM_Param_Base_ Addr + 0x2000 0008	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Class code and Revision ID		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_CLS_CD								SUBCLS_CD								PROG_IF_CODE								REVID							

Bits	Field Name	Description	Type	Reset
31:24	BASE_CLS_CD	Base Class Code (CS)	R	0x0
23:16	SUBCLS_CD	Sub Class Code (CS)	R	0x0
15:8	PROG_IF_CODE	Programming Interface Code (CS)	R	0x0
7:0	REVID	Revision ID (CS)	R	0x1

Table 24-525. PCIECTRL_EP_PCIEWIRE_BIST_HEAD_LAT_CACH

Address offset	0xC		
Physical Address	ECAM_Param_Base_ Addr + 0x2000 000C	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	BIST, Header Type, Latency Timer, Cache Line Size		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BIST	MFD	HEAD_TYP	MSTR_LAT_TIM	CACH_LN_SZE

Bits	Field Name	Description	Type	Reset
31:24	BIST	BIST	R	0x0
23	MFD	MultiFunction Device	R	0x0
22:16	HEAD_TYP	Header Type 0x0 = EP header 0x1 = RC header	R	0x0
15:8	MSTR_LAT_TIM	Master Latency Timer, Not Applicable for PCIe hence hardwired to 0	R	0x0
7:0	CACH_LN_SZE	Cache Line Size, No impact on write, write is allowed only for legacy purpose	RW	0x0

24.9.7.2.3

Table 24-526. PCIECTRL_EP_PCIEWIRE_BAR0

Address offset	0x10		
Physical Address	ECAM_Param_Base_ Addr + 0x2000 0010	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Base Address Register 0 Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31:20	19:4	3	2:1	0
BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Unmasked MSBs, as set by BAR mask **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	R	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	R	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0 = BAR type is Memory 0x1 = BAR type is I/O	R	0x0

Table 24-527. PCIECTRL_EP_PCIEWIRE_BAR1

Address offset	0x14		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0014	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Base Address Register 1 If BAR0.AS = 64-bit: upper half of BAR0 base address If BAR0.AS = 32-bit: independent 32-bit BAR Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">BASE_ADDR_RW</td><td colspan="16">BASE_ADDR_RO</td><td>PREFETCHABLE</td><td colspan="2">AS</td><td>SPACE_INDICATOR</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	R	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	R	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0 = BAR type is Memory 0x1 = BAR type is I/O	R	0x0

Table 24-528. PCIECTRL_EP_PCIEWIRE_BAR2

Address offset	0x18		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0018	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Base Address Register 2 Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	R	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	R	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0 = BAR type is Memory 0x1 = BAR type is I/O	R	0x0

Table 24-529. PCIECTRL_EP_PCIEWIRE_BAR3

Address offset	0x1C		
Physical Address	ECAM_Param_Base_ Addr + 0x2000 001C	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Base Address Register 3 If BAR2.AS = 64-bit: upper half of BAR2 base address If BAR2.AS = 32-bit: independent 32-bit BAR Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_ADDR_RW												BASE_ADDR_RO																PREFETCHABLE	AS		SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0

Bits	Field Name	Description	Type	Reset
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	R	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	R	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0 = BAR type is Memory 0x1 = BAR type is I/O	R	0x0

Table 24-530. PCIECTRL_EP_PCIEWIRE_BAR4

Address offset	0x20		
Physical Address	ECAM_Param_Base_ Addr + 0x2000 0020	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Base Address Register 4 Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31:12	11:4	3	2:1	0
BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:12	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
11:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	R	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	R	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0 = BAR type is Memory 0x1 = BAR type is I/O	R	0x0

Table 24-531. PCIECTRL_EP_PCIEWIRE_BAR5

Address offset	0x24

Table 24-531. PCIECTRL_EP_PCIEWIRE_BAR5 (continued)

Physical Address	ECAM_Param_Base_ Addr + 0x2000 0024	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Base Address Register 5 If BAR4.AS = 64-bit: upper half of BAR4 base address If BAR4.AS = 32-bit: independent 32-bit BAR Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31:20	19:4	3	2:1	0
BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	R	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	R	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0 = BAR type is Memory 0x1 = BAR type is I/O	R	0x0

Table 24-532. PCIECTRL_EP_PCIEWIRE_CARDBUS_CIS_POINTER

Address offset	0x28		
Physical Address	ECAM_Param_Base_ Addr + 0x2000 0028	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description			
Type	R		

31:0
CARDBUS_CIS_PTR_N

Bits	Field Name	Description	Type	Reset
31:0	CARDBUS_CIS_PTR_N	Cardbus CIS pointer (CS)	R	0x0

Table 24-533. PCIECTRL_EP_PCIEWIRE_SUBID_SUBVENDORID

Address offset	0x2C

Table 24-533. PCIECTRL_EP_PCIEWIRE_SUBID_SUBVENDORID (continued)

Physical Address	ECAM_Param_Base_Addr + 0x2000 002C	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description			
Type	R		

31–16	15–0
SUBSYS_DEV_ID_N	SUBSYS_VENDOR_ID_N

Bits	Field Name	Description	Type	Reset
31:16	SUBSYS_DEV_ID_N	Subsystem ID (CS)	R	0x1
15:0	SUBSYS_VENDOR_ID_N	Subsystem Vendor ID (CS)	R	0x0

Table 24-534. PCIECTRL_EP_PCIEWIRE_EXPANSION_ROM_BAR

Address offset	0x30		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0030	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Expansion ROM Base Address Register		
Type	RW		

31–16	15–11	10–1	0
EXROM_ADDRESS	EXROM_ADDRESS_RO	RESERVED	EXROM_EN

Bits	Field Name	Description	Type	Reset
31:16	EXROM_ADDRESS	Expansion ROM address, unmasked (that is, programmable).	RW	0x0
15:11	EXROM_ADDRESS_RO	Expansion ROM address, masked.	R	0x0
10:1	RESERVED		R	0x0
0	EXROM_EN	Expansion ROM Enable	RW	0x0

Table 24-535. PCIECTRL_EP_PCIEWIRE_CAPPTR

Address offset	0x34		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0034	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	CapPtr		
Type	R		

31–8	7–0
RESERVED	CAPTR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	CAPTR	First Capability Pointer (CS)	R	0x40

Table 24-536. PCIECTRL_EP_PCIEWIRE_INTERRUPT

Address offset	0x3C		
Physical Address	ECAM_Param_Base_Addr + 0x2000 003C	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Int Pin and line		

Table 24-536. PCIECTRL_EP_PCIEWIRE_INTERRUPT (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INT_PIN								INT_LIN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	INT_PIN	Interrupt Pin (CS)	R	0x1
7:0	INT_LIN	Interrupt Line	RW	0xFF

Table 24-537. PCIECTRL_EP_PCIEWIRE_PM_CAP

Address Offset	0x40		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0040	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Power Management Capability structure header		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PME_SP					D2_SP	D1_SP	AUX_CUR			DSI	RESERVED	PME_CLK	PMC_VER			PM_NX_PTR								CAP_ID							

Bits	Field Name	Description	Type	Reset
31:27	PME_SP	PME Support (CS); Power states from which PME messages can be sent (active hi, one bit per state) Bit 0: from D0 Bit 1: from D1 Bit 2: from D2 Bit 3: from D3hot Bit 4: from D3cold (if Vaux present)	R	0x0B
26	D2_SP	D2 Support (CS)	R	0
25	D1_SP	D1 Support (CS)	R	1
24:22	AUX_CUR	AUX Current (CS)	R	0x0
21	DSI	Device Specific Initialization (CS)	R	0
20	RESERVED		R	0
19	PME_CLK	PME Clock, hardwired to 0 (CS)	R	0
18:16	PMC_VER	Power Management specification version (CS)	R	0x3
15:8	PM_NX_PTR	Next Capability Pointer (CS)	R	0x50
7:0	CAP_ID	Capability ID Read 0x1: PM	R	0x01

Table 24-538. PCIECTRL_EP_PCIEWIRE_PM_CSR

Address Offset	0x44		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0044	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Power Management Control and Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DATA1	BP_CCE	B2B3_SP	RESERVED	PME_STATUS	DATA_SCALE	DATA_SEL	PME_EN	RESERVED	NSR	RESERVED	PWR_STATE

Bits	Field Name	Description	Type	Reset
31:24	DATA1	Data register for additional information (not supported)	R	0x00
23	BP_CCE	Bus Power/Clock Control Enable, hardwired to 0	R	0
22	B2B3_SP	B2/B3 Support, hardwired to 0	R	0
21:16	RESERVED		R	0x00
15	PME_STATUS	PME Status (Sticky bit)	RW W1toClr	0
14:13	DATA_SCALE	Data Scale (not supported)	R	0x0
12:9	DATA_SEL	Data Select (not supported)	R	0x0
8	PME_EN	PME Enable (Sticky bit) 0x0: Device not enabled to generate PME 0x1: Device enabled to generate PME; implies that Vaux is ON, ie sticky bits will be preserved over reset	RW	0
7:4	RESERVED		R	0x0
3	NSR	No Soft Reset (CS)	R	0
2	RESERVED		R	0
1:0	PWR_STATE	Device Power State 0x0: D0 state 0x1: D1 state 0x2: D2 state 0x3: D3 state	RW	0x0

Table 24-539. PCIECTRL_EP_PCIEWIRE_PCIE_CAP

Address offset	0x70		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0070	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	PCIE cap structure		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		IM_NUM					SLOT	DEV_TYPE				PCIE_VER				PCIE_NX_PTR								CAP_ID							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:25	IM_NUM	Interrupt Message Number (CS)	R	0x0
24	SLOT	Slot Implemented Must be 0 for an endpoint	R	0x0
23:20	DEV_TYPE	Device/Port Type Value depends on assigned type 0x0 = PCIe endpoint 0x1 = Legacy PCIe endpoint	R	0x0
19:16	PCIE_VER	PCI Express Capability Version	R	0x2
15:8	PCIE_NX_PTR	Next Capability Pointer (CS)	R	0x0

Bits	Field Name	Description	Type	Reset
7:0	CAP_ID	Capability ID	R	0x10

Table 24-540. PCIECTRL_EP_PCIEWIRE_DEV_CAP

Address offset	0x74		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0074	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	PCIE Device Capabilities		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			FLR_EN	CAPT_SLOW_PWRLIMIT_SCALE		CAPT_SLOW_PWRLIMIT_VALUE								RESERVED		ROLEBASED_ERRRPT	UNDEFINED			DEFAULT_EP_L1_ACCPT_LATENCY			DEFAULT_EP_L0S_ACCPT_LATENCY			EXTTAGFIELD_SUPPORT	PHANTOMFUNC		MAX_PAYLOAD_SIZE		

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	FLR_EN	Function Level Reset Capability (CS)	R	0x0
27:26	CAPT_SLOW_PWRLIMIT_SCALE	Captured Slow Power Scale Value (CS)	R	0x0
25:18	CAPT_SLOW_PWRLIMIT_VALUE	Captured Slow Power Limit Value (CS)	R	0x0
17:16	RESERVED		R	0x0
15	ROLEBASED_ERRRPT	Role Based Error Reporting (CS)	R	0x1
14:12	UNDEFINED	Undefined from PCIe 1.1 onwards (CS)	R	0x0
11:9	DEFAULT_EP_L1_ACCPT_LATENCY	Endpoint L1 Acceptable Latency (CS)	R	0x3
8:6	DEFAULT_EP_L0S_ACCPT_LATENCY	Endpoint L0s Acceptable Latency (CS)	R	0x4
5	EXTTAGFIELD_SUPPORT	Value derived from DEFAULT_EXT_TAG_FIELD_SUPPORTED	R	0x0
4:3	PHANTOMFUNC	Phantom Function Support, not SUPPORTED (CS)	R	0x0
2:0	MAX_PAYLOAD_SIZE	Maximum Payload Size (CS) Read 0x1 = 256 Byte	R	0x1

Table 24-541. PCIECTRL_EP_PCIEWIRE_DEV_CAS

Address offset	0x78		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0078	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	PCIE Device Control and Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED										TRANS_PEND	AUXP_DET	UR_DET	FT_DET	NFT_DET	COR_DET	INIT_FLR	MRRS			NOSNP_EN	AUXPM_EN	PHFUN_EN	EXTAG_EN	MPS			EN_RO	UR_RE	FT_RE	NFT_RE	COR_RE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	TRANS_PEND	Transaction Pending	R	0x0
20	AUXP_DET	Aux Power Detected	R	0x0
19	UR_DET	Unsupported Request Detected	RW	0x0
18	FT_DET	Fatal Error Detected	RW	0x0
17	NFT_DET	Non-Fatal Error Detected	RW	0x0
16	COR_DET	Correctable Error Detected	RW	0x0
15	INIT_FLR	Reserved	R	0x0
14:12	MRRS	Max_Read_Request_Size	RW	0x2
11	NOSNP_EN	Enable No Snoop	RW	0x0
10	AUXPM_EN	AUX Power PM Enable	RW	0x0
9	PHFUN_EN	Phantom Function Enable	RW	0x0
8	EXTAG_EN	Extended Tag Field Enable	RW	0x0
7:5	MPS	Max_Payload_Size	RW	0x0
4	EN_RO	Enable Relaxed Ordering	RW	0x1
3	UR_RE	Unsupported Request Reporting Enable	RW	0x0
2	FT_RE	Fatal Error Reporting Enable	RW	0x0
1	NFT_RE	Non-Fatal Error Reporting Enable	RW	0x0
0	COR_RE	Correctable Error Reporting Enable	RW	0x0

Table 24-542. PCIECTRL_EP_PCIEWIRE_LNK_CAP

Address offset	0x7C		
Physical Address	ECAM_Param_Base_Addr + 0x2000 007C	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	PCIE Link Capabilities		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORT_NUM								RESERVED	ASPM_OPT_COMP	LNK_BW_not_CAP	DLL_ACTRPT_CAP	UNSUP	CLK_PWR_MGMT	L1_EXIT_LAT			L0S_EXIT_LAT			AS_LINK_PM_SUPPORT		MAX_LINK_WIDTH						MAX_LINK_SPEEDS			

Bits	Field Name	Description	Type	Reset
31:24	PORT_NUM	Port Number (CS)	R	0x0
23	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
22	ASPM_OPT_COMP	ASPM Optionality Compliance (CS)	R	0x1
21	LNK_BW_not_CAP	Link Bandwidth Notification Capability (CS)	R	0x0
20	DLL_ACTRPT_CAP	Data Link Layer Active Reporting Capable	R	0x0
19	UNSUP	Unsupported, Surprise Down Error Reporting Capable, Hardwired to 0	R	0x0
18	CLK_PWR_MGMT	Clock Power Management (CS)	R	0x0
17:15	L1_EXIT_LAT	L1 Exit Latency (CS2)	R	0x6
14:12	L0S_EXIT_LAT	L0s Exit Latency (CS2)	R	0x3
11:10	AS_LINK_PM_SUPPORT	Active State Link PM (ASPM) Support (CS)	R	0x3
9:4	MAX_LINK_WIDTH	Max Link Width (lanes) (CS)	R	0x2
3:0	MAX_LINK_SPEEDS	Supported Max Link Speed (CS) 0x1 = 2.5 GT/s (Gen1) 0x2 = 5 GT/s (Gen2) 0x4 = 8 GT/s (Gen3)	R	0x2

Table 24-543. PCIECTRL_EP_PCIEWIRE_LNK_CAS

Address offset	0x80		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0080	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	PCIE Link Control and Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAB_STATUS	LBW_STATUS	DLL_ACT	SLOT_CLK_CONFIG	LINK_TRAIN	UNDEF	NEG_LW						LINK_SPEED				RESERVED				LABIE	LBMIE	HAWD	EN_CPM	EXT_SYN	COM_CLK_CFG	RETRAIN_LINK	LINK_DIS	RCB	RESERVED	ASPM_CTRL	

Bits	Field Name	Description	Type	Reset
31	LAB_STATUS	Link Autonomous Bandwidth Status	R	0x0
30	LBW_STATUS	Link Bandwidth Management Status	R	0x0
29	DLL_ACT	Data Link Layer Active	R	0x0
28	SLOT_CLK_CONFIG	Slot Clock Configuration (CS)	R	0x1
27	LINK_TRAIN	LINK training	R	0x0
26	UNDEF	Undefined	R	0x0
25:20	NEG_LW	Negotiated Link Width UNDEFINED UNTIL LINK IS UP.	R	0x1
19:16	LINK_SPEED	Link Speed UNDEFINED UNTIL LINK IS UP.	R	0x1
15:12	RESERVED		R	0x0
11	LABIE	Link Autonomous Bandwidth Interrupt Enable.	RW	0x0
10	LBMIE	Link Bandwidth Management Interrupt Enable	RW	0x0
9	HAWD	Hardware Autonomous Width Disable	R	0x0

Bits	Field Name	Description	Type	Reset
8	EN_CPM	Enable Clock Power Management	RW	0x0
7	EXT_SYN	Extended Synch	RW	0x0
6	COM_CLK_CFG	Common Clock Configuration	RW	0x0
5	RETRAIN_LINK	Retrain Link	R	0x0
4	LINK_DIS	Link Disable	R	0x0
3	RCB	Read Completion Boundary (CS) 0x0 = 64 Byte 0x1 = 128 Byte	R	0x1
2	RESERVED		R	0x0
1:0	ASPM_CTRL	Active State Link PM Control 0x0: DISABLED 0x1: L0S_ENABLED 0x2: L1_ENABLED 0x3: L0S_AND_L1_ENABLED	RW	0x0

Table 24-544. PCIECTRL_EP_PCIEWIRE_DEV_CAP_2

Address offset	0x94		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0094	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Device Capabilities 2 Register		
Type	R		

31:14	13:12	11	10	9	8	7	6	5	4	3:0
RESERVED	TPHC_SP	RESERVED	NOROPR	CASC128_SP	AOC64_SP	AOC32_SP	AOR_SP	ARI_FWD_SP	CPL_TIMEOUT_DIS_SUPPORTED	CPL_TIMEOUT_RNG_SUPPORTED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	TPHC_SP	TPH Completer Supported	R	0x0
11	RESERVED		R	0x0
10	NOROPR	No RO-enabled PR-PR Passing	R	0x0
9	CASC128_SP	128-bit CAS Completer Supported	R	0x0
8	AOC64_SP	64-bit AtomicOp Completer Supported	R	0x0
7	AOC32_SP	32-bit AtomicOp Completer Supported	R	0x0
6	AOR_SP	AtomicOp Routing Supported	R	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	R	0x0
4	CPL_TIMEOUT_DIS_SUPPORTED	Completion Timeout Disable Supported	R	0x1

Bits	Field Name	Description	Type	Reset
3:0	CPL_TIMEOUT_RNG_SUPPORTED	Completion Timeout Ranges Supported	R	0x1

Table 24-545. PCIECTRL_EP_PCIEWIRE_DEV_CAS_2

Address offset	0x98		
Physical Address	ECAM_Param_Base_Addr + 0x2000 0098	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Device Control 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	OBFF_EN		RESERVED		LTR_EN	IDO_CPL_EN	IDO_REQ_EN	AOP_EG_BLK	AOP_REQ_EN	ARI_FWD_SP	CPL_TIMEOUT_DIS	CPL_TIMEOUT_VALUE			

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:13	OBFF_EN	OBFF Enable	RW	0x0
12:11	RESERVED		R	0x0
10	LTR_EN	LTR Mechanism Enable	RW	0x0
9	IDO_CPL_EN	IDO Completion Enable	RW	0x0
8	IDO_REQ_EN	IDO Request Enable	RW	0x0
7	AOP_EG_BLK	AtomicOp Egress Blocking	RW	0x0
6	AOP_REQ_EN	AtomicOp Requester Enable	RW	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	RW	0x0
4	CPL_TIMEOUT_DIS	Completion Timeout Disable	RW	0x0
3:0	CPL_TIMEOUT_VALUE	Completion Timeout Values	RW	0x0

Table 24-546. PCIECTRL_EP_PCIEWIRE_LNK_CAP_2

Address offset	0x9C		
Physical Address	ECAM_Param_Base_Addr + 0x2000 009C	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	PCIE Link Capabilities 2 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CROSSLINK_SP	SP_LS_VEC							RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CROSSLINK_SP	Crosslink Supported	R	0x0
7:1	SP_LS_VEC	Supported Link Speeds Vector	R	0x3
0	RESERVED		R	0x0

Table 24-547. PCIECTRL_EP_PCIEWIRE_LNK_CAS_2

Address offset	0xA0		
Physical Address	ECAM_Param_Base_Addr + 0x2000 00A0	**Instance**	PCIe_SS1_EP_CFG_PCIe PCIe_SS2_EP_CFG_PCIe
Description	Link Control and Status 2 Register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td>LINK_EQ_REQ</td><td>EQ_PH3</td><td>EQ_PH2</td><td>EQ_PH1</td><td>EQ_COMPLETE</td><td>DEEMPH_LEVEL</td><td colspan="4">COMPL_PRST_DEEPH</td><td>COMPL_SOS</td><td>ENT_MOD_COMPL</td><td colspan="3">TX_MARGIN</td><td>SEL_DEEMP</td><td>HW_AUTO_SP_DIS</td><td>ENTR_COMPL</td><td colspan="4">TRGT_LINK_SPEED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	LINK_EQ_REQ	Link Equilization Request	RW Wr1toClr	0x0
20	EQ_PH3	Equalization Ph3 Success, Gen3 Only	R	0x0
19	EQ_PH2	Equalization Ph2 Success, Gen3 Only	R	0x0
18	EQ_PH1	Equalization Ph1 Success, Gen3 Only	R	0x0
17	EQ_COMPLETE	Equalization Complete, Gen3 Only	R	0x0
16	DEEMPH_LEVEL	Current De-emphasis Level	R	0x1
15:12	COMPL_PRST_DEEPH	Compliance Pre-set/ De-emphasis	RW	0x0
11	COMPL_SOS	Compliance SOS	RW	0x0
10	ENT_MOD_COMPL	Enter Modified Compliance	RW	0x0
9:7	TX_MARGIN	Transmit Margin	RW	0x0
6	SEL_DEEMP	Selectable De-emphasize	R	0x0
5	HW_AUTO_SP_DIS	Hardware Autonomous Speed Disable	R	0x0
4	ENTR_COMPL	Enter Compliance	RW	0x0
3:0	TRGT_LINK_SPEED	Target Link Speed	RW	0x1

24.9.7.3 PCIe_SS_EP_CFG_DBICS Registers

Note

This section describes the PCIe EP mode (PCIe type-0) standard configuration registers as they are locally accessed within the DIF CS space. These register names are prefixed with "PCIECTRL_EP_DBICS".

24.9.7.3.1 PCIe_SS_EP_CFG_DBICS Register Summary

Table 24-548. PCIe_SS1_EP_CFG_DBICS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_EP_CFG_DBICS Physical Address
PCIECTRL_EP_DBICS_DEVICE_VENDORID	RW	32	0x0	0x5100 0000
PCIECTRL_EP_DBICS_STATUS_COMMAND_REGISTER	RW	32	0x4	0x5100 0004
PCIECTRL_EP_DBICS_CLASSCODE_REVISIONID	RW	32	0x8	0x5100 0008
PCIECTRL_EP_DBICS_BIST_HEAD_LAT_CACH	RW	32	0xC	0x5100 000C
PCIECTRL_EP_DBICS_BAR0	RW	32	0x10	0x5100 0010
PCIECTRL_EP_DBICS_BAR1	RW	32	0x14	0x5100 0014
PCIECTRL_EP_DBICS_BAR2	RW	32	0x18	0x5100 0018
PCIECTRL_EP_DBICS_BAR3	RW	32	0x1C	0x5100 001C
PCIECTRL_EP_DBICS_BAR4	RW	32	0x20	0x5100 0020
PCIECTRL_EP_DBICS_BAR5	RW	32	0x24	0x5100 0024
PCIECTRL_EP_DBICS_CARDBUS_CIS_POINTER	RW	32	0x28	0x5100 0028
PCIECTRL_EP_DBICS_SUBID_SUBVENDORID	RW	32	0x2C	0x5100 002C
PCIECTRL_EP_DBICS_EXPANSION_ROM_BAR	RW	32	0x30	0x5100 0030
PCIECTRL_EP_DBICS_CAPPTR	RW	32	0x34	0x5100 0034
PCIECTRL_EP_DBICS_INTERRUPT	RW	32	0x3C	0x5100 003C
PCIECTRL_EP_DBICS_PM_CAP	RW	32	0x40	0x5100 0040
PCIECTRL_EP_DBICS_PM_CSR	RW	32	0x44	0x5100 0044
PCIECTRL_EP_DBICS_MSI_CAP	RW	32	0x50	0x5100 0050
PCIECTRL_EP_DBICS_MSI_ADD_L32	RW	32	0x54	0x5100 0054
PCIECTRL_EP_DBICS_MSI_ADD_U32	RW	32	0x58	0x5100 0058
PCIECTRL_EP_DBICS_MSI_DATA	RW	32	0x5C	0x5100 005C
PCIECTRL_EP_DBICS_PCIE_CAP	RW	32	0x70	0x5100 0070
PCIECTRL_EP_DBICS_DEV_CAP	RW	32	0x74	0x5100 0074
PCIECTRL_EP_DBICS_DEV_CAS	RW	32	0x78	0x5100 0078
PCIECTRL_EP_DBICS_LNK_CAP	RW	32	0x7C	0x5100 007C
PCIECTRL_EP_DBICS_LNK_CAS	RW	32	0x80	0x5100 0080
PCIECTRL_EP_DBICS_DEV_CAP_2	R	32	0x94	0x5100 0094
PCIECTRL_EP_DBICS_DEV_CAS_2	RW	32	0x98	0x5100 0098
PCIECTRL_EP_DBICS_LNK_CAP_2	R	32	0x9C	0x5100 009C
PCIECTRL_EP_DBICS_LNK_CAS_2	RW	32	0xA0	0x5100 00A0

Table 24-549. PCIe_SS2_EP_CFG_DBICS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_EP_CFG_DBICS Physical Address
PCIECTRL_EP_DBICS_DEVICE_VENDORID	RW	32	0x0	0x5180 0000
PCIECTRL_EP_DBICS_STATUS_COMMAND_REGISTER	RW	32	0x4	0x5180 0004
PCIECTRL_EP_DBICS_CLASSCODE_REVISIONID	RW	32	0x8	0x5180 0008

Table 24-549. PCIe_SS2_EP_CFG_DBICS Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_EP_CFG_DBICS Physical Address
PCIECTRL_EP_DBICS_BIST_HEAD_LAT_CACH	RW	32	0xC	0x5180 000C
PCIECTRL_EP_DBICS_BAR0	RW	32	0x10	0x5180 0010
PCIECTRL_EP_DBICS_BAR1	RW	32	0x14	0x5180 0014
PCIECTRL_EP_DBICS_BAR2	RW	32	0x18	0x5180 0018
PCIECTRL_EP_DBICS_BAR3	RW	32	0x1C	0x5180 001C
PCIECTRL_EP_DBICS_BAR4	RW	32	0x20	0x5180 0020
PCIECTRL_EP_DBICS_BAR5	RW	32	0x24	0x5180 0024
PCIECTRL_EP_DBICS_CARDBUS_CIS_POINTER	RW	32	0x28	0x5180 0028
PCIECTRL_EP_DBICS_SUBID_SUBVENDORID	RW	32	0x2C	0x5180 002C
PCIECTRL_EP_DBICS_EXPANSION_ROM_BAR	RW	32	0x30	0x5180 0030
PCIECTRL_EP_DBICS_CAPPTR	RW	32	0x34	0x5180 0034
PCIECTRL_EP_DBICS_INTERRUPT	RW	32	0x3C	0x5180 003C
PCIECTRL_EP_DBICS_PM_CAP	RW	32	0x40	0x5180 0040
PCIECTRL_EP_DBICS_PM_CSR	RW	32	0x44	0x5180 0044
PCIECTRL_EP_DBICS_MSI_CAP	RW	32	0x50	0x5180 0050
PCIECTRL_EP_DBICS_MSI_ADD_L32	RW	32	0x54	0x5180 0054
PCIECTRL_EP_DBICS_MSI_ADD_U32	RW	32	0x58	0x5180 0058
PCIECTRL_EP_DBICS_MSI_DATA	RW	32	0x5C	0x5180 005C
PCIECTRL_EP_DBICS_PCIE_CAP	RW	32	0x70	0x5180 0070
PCIECTRL_EP_DBICS_DEV_CAP	RW	32	0x74	0x5180 0074
PCIECTRL_EP_DBICS_DEV_CAS	RW	32	0x78	0x5180 0078
PCIECTRL_EP_DBICS_LNK_CAP	RW	32	0x7C	0x5180 007C
PCIECTRL_EP_DBICS_LNK_CAS	RW	32	0x80	0x5180 0080
PCIECTRL_EP_DBICS_DEV_CAP_2	R	32	0x94	0x5180 0094
PCIECTRL_EP_DBICS_DEV_CAS_2	RW	32	0x98	0x5180 0098
PCIECTRL_EP_DBICS_LNK_CAP_2	R	32	0x9C	0x5180 009C
PCIECTRL_EP_DBICS_LNK_CAS_2	RW	32	0xA0	0x5180 00A0

24.9.7.3.2 PCIe_SS_EP_CFG_DBICS Register Description

Table 24-550. PCIECTRL_EP_DBICS_DEVICE_VENDORID

Address offset	0x0		
Physical Address	0x5100 0000 0x5180 0000	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Device and Vendor ID		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">DEVICEID</td><td colspan="16">VENDORID</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	DEVICEID	Device ID (CS)	RW	0x8888
15:0	VENDORID	Vendor ID (CS)	RW	0x104C

Table 24-551. PCIECTRL_EP_DBICS_STATUS_COMMAND_REGISTER

Address offset	0x4		
Physical Address	0x5100 0004 0x5180 0004	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Status and Command registers		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>DETECT_PARERR</td><td>SIGNAL_SYSERR</td><td>RCVD_MASTERABORT</td><td>RCVD_TRGTABORT</td><td>SIGNAL_TRGTABORT</td><td colspan="2">DEVSEL_TIME</td><td>MASTERDATA_PARERR</td><td>FAST_B2B</td><td>RESERVED</td><td>C66MHZ_CAP</td><td>CAP_LIST</td><td>INTX_STATUS</td><td colspan="8">RESERVED</td><td>INTX_ASSER_DIS</td><td>FAST_BBEN</td><td>SERR_EN</td><td>IDSEL_CTRL</td><td>PARITYERR_RESP</td><td>VGA_SNOOP</td><td>MEMWR_INVA</td><td>SPEC_CYCLE_EN</td><td>BUSMASTER_EN</td><td>MEM_SPACE_EN</td><td>IO_SPACE_EN</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	DETECT_PARERR	Detected Parity Error	RW	0x0
30	SIGNAL_SYSERR	Signaled System Error	RW	0x0
29	RCVD_MASTERABORT	Received Master Abort	RW	0x0
28	RCVD_TRGTABORT	Received Target Abort	RW	0x0
27	SIGNAL_TRGTABORT	Signaled Target Abort	RW	0x0
26:25	DEVSEL_TIME	DevSel Timing, Harsdwired to 0 for PCIExpress	R	0x0
24	MASTERDATA_PARERR	Master Data Parity Error	RW	0x0
23	FAST_B2B	Back to Back Capable, Harsdwired to 0 for PCIExpress	R	0x0
22	RESERVED	Reserved	R	0x0
21	C66MHZ_CAP	66MHz Capable, Harsdwired to 0 for PCIExpress	R	0x0
20	CAP_LIST	Capabilities List Hardwired to 1	R	0x1
19	INTX_STATUS	INTx Status	R	0x0
18:11	RESERVED		R	0x0
10	INTX_ASSER_DIS	INTx Assertion Disable	RW	0x0

Bits	Field Name	Description	Type	Reset
9	FAST_BBEN	Bit hardwired to 0 for PCIExpress	R	0x0
8	SERR_EN	SERR Enable	RW	0x0
7	IDSEL_CTRL	Bit hardwired to 0 for PCIExpress	R	0x0
6	PARITYERRRESP	Parity Error Response	RW	0x0
5	VGA_SNOOP	Not Applicable forPCI Express Bit hardwired to 0 for PCIExpress	R	0x0
4	MEMWR_INVA	Not Applicable for PCI Express Bit hardwired to 0 for PCIExpress	R	0x0
3	SPEC_CYCLE_EN	Not Applicable for PCI Express Bit hardwired to 0 for PCIExpress	R	0x0
2	BUSMASTER_EN	Bus Master Enable	RW	0x0
1	MEM_SPACE_EN	Memory Space Enable	RW	0x0
0	IO_SPACE_EN	IO Space Enable	RW	0x0

Table 24-552. PCIECTRL_EP_DBICS_CLASSCODE_REVISIONID

Address offset	0x8		
Physical Address	0x5100 0008 0x5180 0008	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Class code and Revision ID		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_CLS_CD								SUBCLS_CD								PROG_IF_CODE								REVID							

Bits	Field Name	Description	Type	Reset
31:24	BASE_CLS_CD	Base Class Code (CS)	RW	0x0
23:16	SUBCLS_CD	Sub Class Code (CS)	RW	0x0
15:8	PROG_IF_CODE	Programming Interface Code (CS)	RW	0x0
7:0	REVID	Revision ID (CS)	RW	0x1

Table 24-553. PCIECTRL_EP_DBICS_BIST_HEAD_LAT_CACH

Address offset	0xC		
Physical Address	0x5100 000C 0x5180 000C	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	BIST, Header Type, Latency Timer, Cache Line Size		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIST								MFD	HEAD_TYP							MSTR_LAT_TIM								CACH_LN_SZE							

Bits	Field Name	Description	Type	Reset
31:24	BIST	BIST	R	0x0
23	MFD	MultiFunction Device	R	0x0
22:16	HEAD_TYP	Header Type 0x0 = EP header 0x1 = RC header	R	0x0
15:8	MSTR_LAT_TIM	Master Latency Timer, Not Applicable for PCIe hence hardwired to 0	R	0x0

Bits	Field Name	Description	Type	Reset
7:0	CACH_LN_SZE	Cache Line Size, No impact on write, write is allowed only for legacy purpose	RW	0x0

Table 24-554. PCIECTRL_EP_DBICS_BAR0

Address offset	0x10		
Physical Address	0x5100 0010 0x5180 0010	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Base Address Register 0 Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31:20	19:4	3	2:1	0
BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	RW	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0(R) = BAR type is Memory 0x1(R) = BAR type is I/O	RW	0x0

Table 24-555. PCIECTRL_EP_DBICS_BAR1

Address offset	0x14		
Physical Address	0x5100 0014 0x5180 0014	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Base Address Register 1 If BAR0.AS = 64-bit: upper half of BAR0 base address If BAR0.AS = 32-bit: independent 32-bit BAR Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	RW	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0(R) = BAR type is Memory 0x1(R) = BAR type is I/O	RW	0x0

Table 24-556. PCIECTRL_EP_DBICS_BAR2

Address offset	0x18		
Physical Address	0x5100 0018 0x5180 0018	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Base Address Register 2 Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_ADDR_RW												BASE_ADDR_RO																PREFETCHABLE	AS		SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0

Bits	Field Name	Description	Type	Reset
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	RW	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0(R) = BAR type is Memory 0x1(R) = BAR type is I/O	RW	0x0

Table 24-557. PCIECTRL_EP_DBICS_BAR3

Address offset	0x1C		
Physical Address	0x5100 001C 0x5180 001C	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Base Address Register 3 If BAR2.AS = 64-bit: upper half of BAR2 base address If BAR2.AS = 32-bit: independent 32-bit BAR Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">BASE_ADDR_RW</td><td colspan="16">BASE_ADDR_RO</td><td>PREFETCHABLE</td><td colspan="2">AS</td><td>SPACE_INDICATOR</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	RW	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0(R) = BAR type is Memory 0x1(R) = BAR type is I/O	RW	0x0

Table 24-558. PCIECTRL_EP_DBICS_BAR4

Address offset	0x20		
Physical Address	0x5100 0020 0x5180 0020	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS

Table 24-558. PCIECTRL_EP_DBICS_BAR4 (continued)

Description	Base Address Register 4 Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_ADDR_RW																				BASE_ADDR_RO								PREFETCHABLE	AS		SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:12	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
11:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	RW	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0(R) = BAR type is Memory 0x1(R) = BAR type is I/O	RW	0x0

Table 24-559. PCIECTRL_EP_DBICS_BAR5

Address offset	0x24		
Physical Address	0x5100 0024 0x5180 0024	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Base Address Register 5 If BAR4.AS = 64-bit: upper half of BAR4 base address If BAR4.AS = 32-bit: independent 32-bit BAR Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_ADDR_RW												BASE_ADDR_RO																PREFETCHABLE	AS		SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 Bit Read 0x2 = 64 Bit	RW	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0(R) = BAR type is Memory 0x1(R) = BAR type is I/O	RW	0x0

Table 24-560. PCIECTRL_EP_DBICS_CARDBUS_CIS_POINTER

Address offset	0x28		
Physical Address	0x5100 0028 0x5180 0028	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDBUS_CIS_PTR_N																															

Bits	Field Name	Description	Type	Reset
31:0	CARDBUS_CIS_PTR_N	Cardbus CIS pointer (CS)	RW	0x0

Table 24-561. PCIECTRL_EP_DBICS_SUBID_SUBVENDORID

Address offset	0x2C		
Physical Address	0x5100 002C 0x5180 002C	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUBSYS_DEV_ID_N																SUBSYS_VENDOR_ID_N															

Bits	Field Name	Description	Type	Reset
31:16	SUBSYS_DEV_ID_N	Subsystem ID (CS)	RW	0x1
15:0	SUBSYS_VENDOR_ID_N	Subsystem Vendor ID (CS)	RW	0x0

Table 24-562. PCIECTRL_EP_DBICS_EXPANSION_ROM_BAR

Address offset	0x30		
Physical Address	0x5100 0030 0x5180 0030	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Expansion ROM Base Address Register		

Table 24-562. PCIECTRL_EP_DBICS_EXPANSION_ROM_BAR (continued)

Type	RW

31:16	15:11	10:1	0
EXROM_ADDRESS	EXROM_ADDRESS_RO	RESERVED	EXROM_EN

Bits	Field Name	Description	Type	Reset
31:16	EXROM_ADDRESS	Expansion ROM address, unmasked (ie programmable)	RW	0x0
15:11	EXROM_ADDRESS_RO	Expansion ROM address, masked.	R	0x0
10:1	RESERVED		R	0x0
0	EXROM_EN	Expansion ROM Enable	RW	0x0

Table 24-563. PCIECTRL_EP_DBICS_CAPPTR

Address offset	0x34		
Physical Address	0x5100 0034 0x5180 0034	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	CapPtr		
Type	RW		

31:8	7:0
RESERVED	CAPTR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	CAPTR	First Capability Pointer (CS)	RW	0x40

Table 24-564. PCIECTRL_EP_DBICS_INTERRUPT

Address offset	0x3C		
Physical Address	0x5100 003C 0x5180 003C	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Int Pin and line		
Type	RW		

31:16	15:8	7:0
RESERVED	INT_PIN	INT_LIN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:8	INT_PIN	Interrupt Pin (CS)	RW	0x1
7:0	INT_LIN	Interrupt Line	RW	0xFF

Table 24-565. PCIECTRL_EP_DBICS_PM_CAP

Address Offset	0x40		
Physical Address	0x5100 0040 0x5180 0040	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Power Management Capability structure header		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PME_SP					D2_SP	D1_SP	AUX_CUR			DSI	RESERVED	PME_CLK	PMC_VER			PM_NX_PTR								CAP_ID							

Bits	Field Name	Description	Type	Reset
31:27	PME_SP	PME Support (CS); Power states from which PME messages can be sent (active hi, one bit per state) Bit 0: from D0 Bit 1: from D1 Bit 2: from D2 Bit 3: from D3hot Bit 4: from D3cold (if Vaux present)	RW	0x0B
26	D2_SP	D2 Support (CS)	RW	0
25	D1_SP	D1 Support (CS)	RW	1
24:22	AUX_CUR	AUX Current (CS)	RW	0x0
21	DSI	Device Specific Initialization (CS)	RW	0
20	RESERVED	Reserved	R	0
19	PME_CLK	PME Clock, hardwired to 0 (CS)	RW	0
18:16	PMC_VER	Power Management specification version (CS)	RW	0x3
15:8	PM_NX_PTR	Next Capability Pointer (CS)	RW	0x50
7:0	CAP_ID	Capability ID Read 0x1: PM	R	0x01

Table 24-566. PCIECTRL_EP_DBICS_PM_CSR

Address Offset	0x44		
Physical Address	0x5100 0044 0x5180 0044	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Power Management Control and Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1								BP_CCE	B2B3_SP	RESERVED						PME_STATUS	DATA_SCALE		DATA_SEL				PME_EN	RESERVED				NSR	RESERVED	PM_STATE	

Bits	Field Name	Description	Type	Reset
31:24	DATA1	Data register for additional information (not supported)	R	0x00
23	BP_CCE	Bus Power/Clock Control Enable, hardwired to 0	R	0
22	B2B3_SP	B2/B3 Support, hardwired to 0	R	0
21:16	RESERVED	Reserved	R	0x00
15	PME_STATUS	PME Status (Sticky bit)	RW W1toClr	0
14:13	DATA_SCALE	Data Scale (not supported)	R	0x0
12:9	DATA_SEL	Data Select (not supported)	R	0x0
8	PME_EN	PME Enable (Sticky bit) 0x0: Device not enabled to generate PME 0x1: Device enabled to generate PME; implies that Vaux is ON, ie sticky bits will be preserved over reset	RW	0

Bits	Field Name	Description	Type	Reset
7:4	RESERVED	Reserved	R	0x0
3	NSR	No Soft Reset (CS)	RW	0
2	RESERVED	Reserved	R	0
1:0	PM_STATE	Power Management Control and Status Register 0x0: D0 state 0x1: D1 state 0x2: D2 state 0x3: D3 state	RW	0x0

Table 24-567. PCIECTRL_EP_DBICS_MSI_CAP

Address Offset	0x0000 0050		
Physical Address	0x5100 0050 0x5180 0050	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Message Signaled Interrupt Capability structure header		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							PVM_EN	MSI_64_EN	MME			MMC			MSI_EN	MSI_NX_PTR								CAP_ID							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x00
24	PVM_EN	MSI Per Vector Masking (PVM) supported	R	0
23	MSI_64_EN	64-bit Address Capable (CS)	R	1
22:20	MME	Multiple Message Enable	RW	0x0
19:17	MMC	Multiple Message Capable (CS)	R	0x0
16	MSI_EN	MSI Enable	RW	0
15:8	MSI_NX_PTR	Next Capability Pointer (CS)	R	0x70
7:0	CAP_ID	MSI Capability ID Read 0x05 MSI	R	0x05

Table 24-568. PCIECTRL_EP_DBICS_MSI_ADD_L32

Address Offset	0x0000 0054		
Physical Address	0x5100 0054 0x5180 0054	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIe memory space address of MSI write TLP request, lower 32 bits.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																														RESERVED	

Bits	Field Name	Description	Type	Reset
31:2	ADDR	Lower 32-bit address (DWORD aligned)	RW	0x0000 0000
1:0	RESERVED	Reserved	R	0x0

Table 24-569. PCIECTRL_EP_DBICS_MSI_ADD_U32

Address Offset	0x0000 0058

Table 24-569. PCIECTRL_EP_DBICS_MSI_ADD_U32 (continued)

Physical Address	0x5100 0058 0x5180 0058	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIe memory space address of MSI write TLP request, upper 32 bits (used if MSI_64_EN = 1).		
Type	RW		

31–0
ADDR

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Upper 32-bit address	RW	0x0000 0000

Table 24-570. PCIECTRL_EP_DBICS_MSI_DATA

Address Offset	0x0000 005C		
Physical Address	0x5100 005C 0x5180 005C	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Data of MSI write TLP request (modified for multiple vectors)		
Type	RW		

31–16	15–0
RESERVED	DATA

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	DATA	MSI data	RW	0x0000

Table 24-571. PCIECTRL_EP_DBICS_PCIE_CAP

Address offset	0x70		
Physical Address	0x5100 0070 0x5180 0070	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIE cap structure		
Type	RW		

31–30	29–25	24	23–20	19–16	15–8	7–0
RESERVED	IM_NUM	SLOT	DEV_TYPE	PCIE_VER	PCIE_NX_PTR	CAP_ID

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29:25	IM_NUM	Interrupt Message Number (CS)	RW	0x0
24	SLOT	Slot Implemented Must be 0 for an endpoint	RW	0x0
23:20	DEV_TYPE	Device/Port Type Value depends on assigned type 0x0 = PCIe endpoint 0x1 = Legacy PCIe endpoint	R	0x0
19:16	PCIE_VER	PCI Express Capability Version	R	0x2
15:8	PCIE_NX_PTR	Next Capability Pointer (CS)	RW	0x0
7:0	CAP_ID	Capability ID	R	0x10

Table 24-572. PCIECTRL_EP_DBICS_DEV_CAP

Address offset	0x74

Table 24-572. PCIECTRL_EP_DBICS_DEV_CAP (continued)

Physical Address	0x5100 0074 0x5180 0074	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIE Device Capabilities		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			FLR_EN	CAPT_SLOW_PWR LIMIT_SCALE		CAPT_SLOW_PWRLIMIT_VALUE								RESERVED		ROLEBASED_ERRRPT	UNDEFINED			DEFAULT_EP_L1_ACCPT_LATENCY			DEFAULT_EP_L0S_ACCPT_LATENCY			EXTTAGFIELD_SUPPORT	PHANTOMFUNC		MAX_PAYLOAD_SIZE		

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28	FLR_EN	Function Level Reset Capability (CS)	RW	0x0
27:26	CAPT_SLOW_PWRLIMIT_SCALE	Captured Slow Power Scale Value (CS)	RW	0x0
25:18	CAPT_SLOW_PWRLIMIT_VALUE	Captured Slow Power Limit Value (CS)	RW	0x0
17:16	RESERVED	Reserved	R	0x0
15	ROLEBASED_ERRRPT	Role Based Error Reporting (CS)	RW	0x1
14:12	UNDEFINED	Undefined from PCIe 1.1 onwards (CS)	R	0x0
11:9	DEFAULT_EP_L1_ACCPT_LATENCY	Endpoint L1 Acceptable Latency (CS)	R	0x3
8:6	DEFAULT_EP_L0S_ACCPT_LATENCY	Endpoint L0s Acceptable Latency (CS)	R	0x4
5	EXTTAGFIELD_SUPPORT	Value derived from DEFAULT_EXT_TAG_FIELD_SUPPORTED	RW	0x0
4:3	PHANTOMFUNC	Phantom Function Support, not SUPPORTED (CS)	RW	0x0
2:0	MAX_PAYLOAD_SIZE	Maximum Payload Size (CS) Read 0x1 = 256 Byte	RW	0x1

Table 24-573. PCIECTRL_EP_DBICS_DEV_CAS

Address offset	0x78		
Physical Address	0x5100 0078 0x5180 0078	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIE Device Control and Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										TRANS_PEND	AUXP_DET	UR_DET	FT_DET	NFT_DET	COR_DET	INIT_FLR	MRRS			NOSNP_EN	AUXPM_EN	PHFUN_EN	EXTAG_EN	MPS			EN_RO	UR_RE	FT_RE	NFT_RE	COR_RE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x0
21	TRANS_PEND	Transaction Pending	R	0x0
20	AUXP_DET	Aux Power Detected	R	0x0
19	UR_DET	Unsupported Request Detected	RW	0x0
18	FT_DET	Fatal Error Detected	RW	0x0
17	NFT_DET	Non-Fatal Error Detected	RW	0x0
16	COR_DET	Correctable Error Detected	RW	0x0
15	INIT_FLR	Reserved	R	0x0
14:12	MRRS	Max_Read_Request_Size	RW	0x2
11	NOSNP_EN	Enable No Snoop	RW	0x0
10	AUXPM_EN	AUX Power PM Enable	RW	0x0
9	PHFUN_EN	Phantom Function Enable	RW	0x0
8	EXTAG_EN	Extended Tag Field Enable	RW	0x0
7:5	MPS	Max_Payload_Size	RW	0x0
4	EN_RO	Enable Relaxed Ordering	RW	0x1
3	UR_RE	Unsupported Request Reporting Enable	RW	0x0
2	FT_RE	Fatal Error Reporting Enable	RW	0x0
1	NFT_RE	Non-Fatal Error Reporting Enable	RW	0x0
0	COR_RE	Correctable Error Reporting Enable	RW	0x0

Table 24-574. PCIECTRL_EP_DBICS_LNK_CAP

Address offset	0x7C		
Physical Address	0x5100 007C 0x5180 007C	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIE Link Capabilities		
Type	RW		

31-24	23	22	21	20	19	18	17-15	14-12	11-10	9-4	3-0
PORT_NUM	RESERVED	ASPM_OPT_COMP	LNK_BW_not_CAP	DLL_ACTRPT_CAP	UNSUP	CLK_PWR_MGMT	L1_EXIT_LAT	L0S_EXIT_LAT	AS_LINK_PM_SUPPORT	MAX_LINK_WIDTH	MAX_LINK_SPEEDS

Bits	Field Name	Description	Type	Reset
31:24	PORT_NUM	Port Number (CS)	RW	0x0
23	RESERVED		R	0x0
22	ASPM_OPT_COMP	ASPM Optionality Compliance (CS)	RW	0x1
21	LNK_BW_not_CAP	Link Bandwidth Notification Capability (CS)	RW	0x0
20	DLL_ACTRPT_CAP	Data Link Layer Active Reporting Capable	R	0x0
19	UNSUP	Unsupported, Surprise Down Error Reporting Capable, Hardwired to 0	R	0x0

Bits	Field Name	Description	Type	Reset
18	CLK_PWR_MGMT	Clock Power Management (CS)	RW	0x0
17:15	L1_EXIT_LAT	L1 Exit Latency (CS2)	R	0x6
14:12	L0S_EXIT_LAT	L0s Exit Latency (CS2)	R	0x3
11:10	AS_LINK_PM_SUPPORT	Active State Link PM (ASPM) Support (CS)	RW	0x3
9:4	MAX_LINK_WIDTH	Max Link Width (lanes) (CS)	RW	0x2
3:0	MAX_LINK_SPEEDS	Supported Max Link Speed (CS) 0x1(R) = 2.5 GT/s (Gen1) 0x2(R) = 5 GT/s (Gen2) 0x4(R) = 8 GT/s (Gen3)	RW	0x2

Table 24-575. PCIECTRL_EP_DBICS_LNK_CAS

Address offset	0x80		
Physical Address	0x5100 0080 0x5180 0080	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIE Link Control and Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAB_STATUS	LBW_STATUS	DLL_ACT	SLOT_CLK_CONFIG	LINK_TRAIN	UNDEF	NEG_LW						LINK_SPEED				RESERVED				LABIE	LBMIE	HAWD	EN_CPM	EXT_SYN	COM_CLK_CFG	RETRAIN_LINK	LINK_DIS	RCB	RESERVED	ASPM_CTRL	

Bits	Field Name	Description	Type	Reset
31	LAB_STATUS	Link Autonomous Bandwidth Status	R	0x0
30	LBW_STATUS	Link Bandwidth Management Status	R	0x0
29	DLL_ACT	Data Link Layer Active	R	0x0
28	SLOT_CLK_CONFIG	Slot Clock Configuration (CS)	RW	0x1
27	LINK_TRAIN	LINK training	R	0x0
26	UNDEF	Undefined	R	0x0
25:20	NEG_LW	Negotiated Link Width UNDEFINED UNTIL LINK IS UP.	R	0x1
19:16	LINK_SPEED	Link Speed UNDEFINED UNTIL LINK IS UP.	R	0x1
15:12	RESERVED		R	0x0
11	LABIE	Link Autonomous Bandwidth Interrupt Enable.	RW	0x0
10	LBMIE	Link Bandwidth Management Interrupt Enable	RW	0x0
9	HAWD	Hardware Autonomous Width Disable	R	0x0
8	EN_CPM	Enable Clock Power Management	RW	0x0
7	EXT_SYN	Extended Synch	RW	0x0
6	COM_CLK_CFG	Common Clock Configuration	RW	0x0
5	RETRAIN_LINK	Retrain Link	R	0x0
4	LINK_DIS	Link Disable	R	0x0

Bits	Field Name	Description	Type	Reset
3	RCB	Read Completion Boundary (CS) 0x0 = 64 Byte 0x1 = 128 Byte	RW	0x1
2	RESERVED		R	0x0
1:0	ASPM_CTRL	Active State Link PM Control 0x0: DISABLED 0x1: L0S_ENABLED 0x2: L1_ENABLED 0x3: L0S_AND_L1_ENABLED	RW	0x0

Table 24-576. PCIECTRL_EP_DBICS_DEV_CAP_2

Address offset	0x94		
Physical Address	0x5100 0094 0x5180 0094	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Device Capabilities 2 Register		
Type	R		

31:14	13:12	11	10	9	8	7	6	5	4	3:0
RESERVED	TPHC_SP	RESERVED	NOROPR	CASC128_SP	AOC64_SP	AOC32_SP	AOR_SP	ARI_FWD_SP	CPL_TIMEOUT_DIS_SUPPORTED	CPL_TIMEOUT_RNG_SUPPORTED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	TPHC_SP	TPH Completer Supported	R	0x0
11	RESERVED		R	0x0
10	NOROPR	No RO-enabled PR-PR Passing	R	0x0
9	CASC128_SP	128-bit CAS Completer Supported	R	0x0
8	AOC64_SP	64-bit AtomicOp Completer Supported	R	0x0
7	AOC32_SP	32-bit AtomicOp Completer Supported	R	0x0
6	AOR_SP	AtomicOp Routing Supported	R	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	R	0x0
4	CPL_TIMEOUT_DIS_SUPPORTED	Completion Timeout Disable Supported	R	0x1
3:0	CPL_TIMEOUT_RNG_SUPPORTED	Completion Timeout Ranges Supported	R	0x1

Table 24-577. PCIECTRL_EP_DBICS_DEV_CAS_2

Address offset	0x98		
Physical Address	0x5100 0098 0x5180 0098	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS

Table 24-577. PCIECTRL_EP_DBICS_DEV_CAS_2 (continued)

Description	Device Control 2 Register
Type	RW

31:15	14:13	12:11	10	9	8	7	6	5	4	3:0
RESERVED	OBFF_EN	RESERVED	LTR_EN	IDO_CPL_EN	IDO_REQ_EN	AOP_EG_BLK	AOP_REQ_EN	ARI_FWD_SP	CPL_TIMEOUT_DIS	CPL_TIMEOUT_VALUE

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:13	OBFF_EN	OBFF Enable	RW	0x0
12:11	RESERVED		R	0x0
10	LTR_EN	LTR Mechanism Enable	RW	0x0
9	IDO_CPL_EN	IDO Completion Enable	RW	0x0
8	IDO_REQ_EN	IDO Request Enable	RW	0x0
7	AOP_EG_BLK	AtomicOp Egress Blocking	RW	0x0
6	AOP_REQ_EN	AtomicOp Requester Enable	RW	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	RW	0x0
4	CPL_TIMEOUT_DIS	Completion Timeout Disable	RW	0x0
3:0	CPL_TIMEOUT_VALUE	Completion Timeout Values	RW	0x0

Table 24-578. PCIECTRL_EP_DBICS_LNK_CAP_2

Address offset	0x9C		
Physical Address	0x5100 009C 0x5180 009C	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	PCIE Link Capabilities 2 Register		
Type	R		

31:9	8	7:1	0
RESERVED	CROSSLINK_SP	SP_LS_VEC	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CROSSLINK_SP	Crosslink Supported	R	0x0
7:1	SP_LS_VEC	Supported Link Speeds Vector	R	0x3
0	RESERVED		R	0x0

Table 24-579. PCIECTRL_EP_DBICS_LNK_CAS_2

Address offset	0xA0

Table 24-579. PCIECTRL_EP_DBICS_LNK_CAS_2 (continued)

Physical Address	0x5100 00A0 0x5180 00A0	**Instance**	PCIe_SS1_EP_CFG_DBICS PCIe_SS2_EP_CFG_DBICS
Description	Link Control and Status 2 Register		
Type	RW		

31:22	21	20	19	18	17	16	15:12	11	10	9:7	6	5	4	3:0
RESERVED	LINK_EQ_REQ	EQ_PH3	EQ_PH2	EQ_PH1	EQ_COMPLETE	DEEMPH_LEVEL	COMPL_PRST_DEEPH	COMPL_SOS	ENT_MOD_COMPL	TX_MARGIN	SEL_DEEMP	HW_AUTO_SP_DIS	ENTR_COMPL	TRGT_LINK_SPEED

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	LINK_EQ_REQ	Link Equilization Request	RW Wr1toClr	0x0
20	EQ_PH3	Equalization Ph3 Success, Gen3 Only	R	0x0
19	EQ_PH2	Equalization Ph2 Success, Gen3 Only	R	0x0
18	EQ_PH1	Equalization Ph1 Success, Gen3 Only	R	0x0
17	EQ_COMPLETE	Equalization Complete, Gen3 Only	R	0x0
16	DEEMPH_LEVEL	Current De-emphasis Level	R	0x1
15:12	COMPL_PRST_DEEPH	Compliance Pre-set/ De-emphasis	RW	0x0
11	COMPL_SOS	Compliance SOS	RW	0x0
10	ENT_MOD_COMPL	Enter Modified Compliance	RW	0x0
9:7	TX_MARGIN	Transmit Margin	RW	0x0
6	SEL_DEEMP	Selectable De-emphasize	R	0x0
5	HW_AUTO_SP_DIS	Hardware Autonomous Speed Disable	R	0x0
4	ENTR_COMPL	Enter Compliance	RW	0x0
3:0	TRGT_LINK_SPEED	Target Link Speed	RW	0x1

24.9.7.4 PCIe_SS_RC_CFG_DBICS Registers

Note

This section describes the PCIe RC mode (PCIe type-1) standard configuration registers as they are locally accessed within the DIF CS space. These register names are prefixed with "PCIECTRL_RC_DBICS".

24.9.7.4.1 PCIe_SS_RC_CFG_DBICS Register Summary

Table 24-580. PCIe_SS1_RC_CFG_DBICS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_RC_CFG_DBICS Physical Address
PCIECTRL_RC_DBICS_DEVICE_VENDORID	RW	32	0x0000 0000	0x5100 0000
PCIECTRL_RC_DBICS_STATUS_COMMAND_REGISTER	RW	32	0x0000 0004	0x5100 0004
PCIECTRL_RC_DBICS_CLASSCODE_REVISIONID	RW	32	0x0000 0008	0x5100 0008
PCIECTRL_RC_DBICS_BIST_HEAD_LAT_CACH	RW	32	0x0000 000C	0x5100 000C
PCIECTRL_RC_DBICS_BAR0	RW	32	0x0000 0010	0x5100 0010
PCIECTRL_RC_DBICS_BAR1	RW	32	0x0000 0014	0x5100 0014
PCIECTRL_RC_DBICS_BUS_NUM_REG	RW	32	0x0000 0018	0x5100 0018
PCIECTRL_RC_DBICS_IOBASE_LIMIT_SEC_STATUS	RW	32	0x0000 001C	0x5100 001C
PCIECTRL_RC_DBICS_MEM_BASE_LIMIT	RW	32	0x0000 0020	0x5100 0020
PCIECTRL_RC_DBICS_PREF_MEM_BASE_LIMIT	RW	32	0x0000 0024	0x5100 0024
PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_BASEADDR	RW	32	0x0000 0028	0x5100 0028
PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_LIMITADDR	RW	32	0x0000 002C	0x5100 002C
PCIECTRL_RC_DBICS_IO_BASE_LIMIT	RW	32	0x0000 0030	0x5100 0030
PCIECTRL_RC_DBICS_CAPPTR	RW	32	0x0000 0034	0x5100 0034
PCIECTRL_RC_DBICS_EXPANSION_ROM_BAR	RW	32	0x0000 0038	0x5100 0038
PCIECTRL_RC_DBICS_BRIDGE_INT	RW	32	0x0000 003C	0x5100 003C
PCIECTRL_RC_DBICS_PCIE_CAP	RW	32	0x0000 0070	0x5100 0070
PCIECTRL_RC_DBICS_DEV_CAP	RW	32	0x0000 0074	0x5100 0074
PCIECTRL_RC_DBICS_DEV_CAS	RW	32	0x0000 0078	0x5100 0078
PCIECTRL_RC_DBICS_LNK_CAP	RW	32	0x0000 007C	0x5100 007C
PCIECTRL_RC_DBICS_LNK_CAS	RW	32	0x0000 0080	0x5100 0080
PCIECTRL_RC_DBICS_SLOT_CAP	RW	32	0x0000 0084	0x5100 0084
PCIECTRL_RC_DBICS_SLOT_CAS	RW	32	0x0000 0088	0x5100 0088
PCIECTRL_RC_DBICS_ROOT_CAC	RW	32	0x0000 008C	0x5100 008C
PCIECTRL_RC_DBICS_ROOT_STS	RW	32	0x0000 0090	0x5100 0090
PCIECTRL_RC_DBICS_DEV_CAP_2	R	32	0x0000 0094	0x5100 0094
PCIECTRL_RC_DBICS_DEV_CAS_2	RW	32	0x0000 0098	0x5100 0098
PCIECTRL_RC_DBICS_LNK_CAP_2	R	32	0x0000 009C	0x5100 009C
PCIECTRL_RC_DBICS_LNK_CAS_2	RW	32	0x0000 00A0	0x5100 00A0

Table 24-581. PCIe_SS2_RC_CFG_DBICS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_RC_CFG_DBICS Physical Address
PCIECTRL_RC_DBICS_DEVICE_VENDORID	RW	32	0x0000 0000	0x5180 0000
PCIECTRL_RC_DBICS_STATUS_COMMAND_REGISTER	RW	32	0x0000 0004	0x5180 0004
PCIECTRL_RC_DBICS_CLASSCODE_REVISIONID	RW	32	0x0000 0008	0x5180 0008
PCIECTRL_RC_DBICS_BIST_HEAD_LAT_CACH	RW	32	0x0000 000C	0x5180 000C

Table 24-581. PCIe_SS2_RC_CFG_DBICS Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_RC_CFG_DBICS Physical Address
PCIECTRL_RC_DBICS_BAR0	RW	32	0x0000 0010	0x5180 0010
PCIECTRL_RC_DBICS_BAR1	RW	32	0x0000 0014	0x5180 0014
PCIECTRL_RC_DBICS_BUS_NUM_REG	RW	32	0x0000 0018	0x5180 0018
PCIECTRL_RC_DBICS_IOBASE_LIMIT_SEC_STATUS	RW	32	0x0000 001C	0x5180 001C
PCIECTRL_RC_DBICS_MEM_BASE_LIMIT	RW	32	0x0000 0020	0x5180 0020
PCIECTRL_RC_DBICS_PREF_MEM_BASE_LIMIT	RW	32	0x0000 0024	0x5180 0024
PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_BASEADDR	RW	32	0x0000 0028	0x5180 0028
PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_LIMITADDR	RW	32	0x0000 002C	0x5180 002C
PCIECTRL_RC_DBICS_IO_BASE_LIMIT	RW	32	0x0000 0030	0x5180 0030
PCIECTRL_RC_DBICS_CAPPTR	RW	32	0x0000 0034	0x5180 0034
PCIECTRL_RC_DBICS_EXPANSION_ROM_BAR	RW	32	0x0000 0038	0x5180 0038
PCIECTRL_RC_DBICS_BRIDGE_INT	RW	32	0x0000 003C	0x5180 003C
PCIECTRL_RC_DBICS_PCIE_CAP	RW	32	0x0000 0070	0x5180 0070
PCIECTRL_RC_DBICS_DEV_CAP	RW	32	0x0000 0074	0x5180 0074
PCIECTRL_RC_DBICS_DEV_CAS	RW	32	0x0000 0078	0x5180 0078
PCIECTRL_RC_DBICS_LNK_CAP	RW	32	0x0000 007C	0x5180 007C
PCIECTRL_RC_DBICS_LNK_CAS	RW	32	0x0000 0080	0x5180 0080
PCIECTRL_RC_DBICS_SLOT_CAP	RW	32	0x0000 0084	0x5180 0084
PCIECTRL_RC_DBICS_SLOT_CAS	RW	32	0x0000 0088	0x5180 0088
PCIECTRL_RC_DBICS_ROOT_CAC	RW	32	0x0000 008C	0x5180 008C
PCIECTRL_RC_DBICS_ROOT_STS	RW	32	0x0000 0090	0x5180 0090
PCIECTRL_RC_DBICS_DEV_CAP_2	R	32	0x0000 0094	0x5180 0094
PCIECTRL_RC_DBICS_DEV_CAS_2	RW	32	0x0000 0098	0x5180 0098
PCIECTRL_RC_DBICS_LNK_CAP_2	R	32	0x0000 009C	0x5180 009C
PCIECTRL_RC_DBICS_LNK_CAS_2	RW	32	0x0000 00A0	0x5180 00A0

24.9.7.4.2 PCIe_SS_RC_CFG_DBICS Register Description

Table 24-582. PCIECTRL_RC_DBICS_DEVICE_VENDORID

Address Offset	0x0000 0000		
Physical Address	0x5100 0000 0x5180 0000	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Device and Vendor ID		
Type	RW		

31–16	15–0
DEVICEID	VENDORID

Bits	Field Name	Description	Type	Reset
31:16	DEVICEID	Device ID (CS)	RW	0x8888
15:0	VENDORID	Vendor ID (CS)	RW	0x104c

Table 24-583. PCIECTRL_RC_DBICS_STATUS_COMMAND_REGISTER

Address Offset	0x0000 0004		
Physical Address	0x5100 0004 0x5180 0004	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Status and Command registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DETECT_PARERR	SIGNAL_SYSERR	RCVD_MASTERABORT	RCVD_TRGTABORT	SIGNAL_TRGTABORT	DEVSEL_TIME		MASTERDATA_PARERR	FAST_B2B	RESERVED	C66MHZ_CAP	CAP_LIST	INTX_STATUS	RESERVED								INTX_ASSER_DIS	FAST_BBEN	SERR_EN	IDSEL_CTRL	PARITYERRRESP	VGA_SNOOP	MEMWR_INVA	SPEC_CYCLE_EN	BUSMASTER_EN	MEM_SPACE_EN	IO_SPACE_EN

Bits	Field Name	Description	Type	Reset
31	DETECT_PARERR	Detected Parity Error	RW	0x0
30	SIGNAL_SYSERR	Signaled System Error	RW	0x0
29	RCVD_MASTERABORT	Received Master Abort	RW	0x0
28	RCVD_TRGTABORT	Received Target Abort	RW	0x0
27	SIGNAL_TRGTABORT	Signaled Target Abort	RW	0x0
26:25	DEVSEL_TIME	DevSel Timing, Harsdwired to 0 for PCIExpress	R	0x0
24	MASTERDATA_PARERR	Master Data Parity Error	RW	0x0
23	FAST_B2B	Back to Back Capable, Harsdwired to 0 for PCIExpress	R	0x0
22	RESERVED	Reserved	R	0x0
21	C66MHZ_CAP	66MHz Capable, Harsdwired to 0 for PCIExpress	R	0x0
20	CAP_LIST	Capabilities List Hardwired to 1	R	0x1
19	INTX_STATUS	INTx Status	R	0x0
18:11	RESERVED		R	0x0
10	INTX_ASSER_DIS	INTx Assertion Disable	RW	0x0
9	FAST_BBEN	Bit hardwired to 0 for PCIExpress	R	0x0
8	SERR_EN	SERR Enable	RW	0x0
7	IDSEL_CTRL	Bit hardwired to 0 for PCIExpress	R	0x0
6	PARITYERRRESP	Parity Error Response	RW	0x0
5	VGA_SNOOP	Not Applicable forPCI Express; Bit hardwired to 0 for PCIExpress	R	0x0
4	MEMWR_INVA	Not Applicable for PCI Express; Bit hardwired to 0 for PCIExpress	R	0x0
3	SPEC_CYCLE_EN	Not Applicable for PCI Express; Bit hardwired to 0 for PCIExpress	R	0x0
2	BUSMASTER_EN	Bus Master Enable (BME)	RW	0x0
1	MEM_SPACE_EN	Memory Space Enable (MSE)	RW	0x0
0	IO_SPACE_EN	IO Space Enable (ISE)	RW	0x0

Table 24-584. PCIECTRL_RC_DBICS_CLASSCODE_REVISIONID

Address Offset	0x0000 0008		
Physical Address	0x5100 0008 0x5180 0008	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Class code and Revision ID		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_CLS_CD								SUBCLS_CD								PROG_IF_CODE								REVID							

Bits	Field Name	Description	Type	Reset
31:24	BASE_CLS_CD	Base Class Code (CS)	RW	0x0

Bits	Field Name	Description	Type	Reset
23:16	SUBCLS_CD	Sub Class Code (CS)	RW	0x0
15:8	PROG_IF_CODE	Programming Interface Code (CS)	RW	0x0
7:0	REVID	Revision ID (CS)	RW	0x1

Table 24-585. PCIECTRL_RC_DBICS_BIST_HEAD_LAT_CACH

Address Offset	0x0000 000C		
Physical Address	0x5100 000C 0x5180 000C	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	BIST, Header Type, Latency Timer, Cache Line Size		
Type	RW		

31:24	23	22:16	15:8	7:0
BIST	MFD	HEAD_TYP	MSTR_LAT_TIM	CACH_LN_SZE

Bits	Field Name	Description	Type	Reset
31:24	BIST	BIST	R	0x0
23	MFD	MultiFunction Device	R	0x0
22:16	HEAD_TYP	Header Type 0x0: EP header (Type0) 0x1: RC header (Type1)	R	0x1
15:8	MSTR_LAT_TIM	Master Latency Timer, Not Applicable for PCIe hence hardwired to 0	R	0x0
7:0	CACH_LN_SZE	Cache Line Size, No impact on write, write is allowed only for legacy purpose	RW	0x0

Table 24-586. PCIECTRL_RC_DBICS_BAR0

Address Offset	0x0000 0010		
Physical Address	0x5100 0010 0x5180 0010	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Base Address Register 0 Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31:20	19:4	3	2:1	0
BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0

Bits	Field Name	Description	Type	Reset
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, upper base address bits are in BAR above). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LS Bit of I/O address Read 0x0 = 32 bit Read 0x2 = 64 bit	RW	0x0
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0: BAR type is Memory (MEM) 0x1: BAR type is I/O (IO)	RW	0x0

Table 24-587. PCIECTRL_RC_DBICS_BAR1

Address Offset	0x0000 0014		
Physical Address	0x5100 0014 0x5180 0014	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Base Address Register 1 If BAR0.AS = 64-bit: upper half of BAR0 base address If BAR0.AS = 32-bit: independent 32-bit BAR Bit #0 is also a WO BAR enable (CS2) BAR Mask is writable (CS2)		
Type	RW		

31:20	19:4	3	2:1	0
BASE_ADDR_RW	BASE_ADDR_RO	PREFETCHABLE	AS	SPACE_INDICATOR

Bits	Field Name	Description	Type	Reset
31:20	BASE_ADDR_RW	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Unmasked MSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	RW	0x0
19:4	BASE_ADDR_RO	Base address bits (for a 64-bit BAR, lower base address bits are in BAR below). Masked LSBs, as set by BAR mask. **NOTE: The RO and RW division between bits 19 and 20 is based on the assumption that BAR mask is 20 bits (1MB).**	R	0x0
3	PREFETCHABLE	MEM BAR: Prefetchable (CS) I/O BAR: bit 1 is part of I/O address	RW	0x1
2:1	AS	MEM BAR: Address Size (CS) I/O BAR: bit 0 is always 0, bit 1 is LSBit of I/O address Read 0x0 = 32 bit Read 0x2 = 64 bit	RW	0x0

Bits	Field Name	Description	Type	Reset
0	SPACE_INDICATOR	BAR I/O vs memory space indicator (CS) 0x0: BAR type is Memory (MEM) 0x1: BAR type is I/O (IO)	RW	0x0

Table 24-588. PCIECTRL_RC_DBICS_BUS_NUM_REG

Address Offset	0x0000 0018		
Physical Address	0x5100 0018 0x5180 0018	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Bus Number Registers		
Type	RW		

31:24	23:16	15:8	7:0
SEC_LAT_TIMER	SUBORD_BUS_NUM	SEC_BUS_NUM	PRIM_BUS_NUM

Bits	Field Name	Description	Type	Reset
31:24	SEC_LAT_TIMER	Secondary Latency Timer, Not Applicable for PCI Express hence hardwired to 0	R	0x0
23:16	SUBORD_BUS_NUM	Subordinate Bus Number	RW	0x0
15:8	SEC_BUS_NUM	Secondary Bus Number	RW	0x0
7:0	PRIM_BUS_NUM	Primary Bus Number	RW	0x0

Table 24-589. PCIECTRL_RC_DBICS_IOBASE_LIMIT_SEC_STATUS

Address Offset	0x0000 001C		
Physical Address	0x5100 001C 0x5180 001C	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	IO Base,Limit and Secondary Status Register		
Type	RW		

31	30	29	28	27	26:25	24	23	22	21	20:16	15:12	11:9	8	7:4	3:1	0
DET_PAR_ERR	RCVD_SYS_ERR	RCVD_MSTR_ABORT	RCVD_TRGT_ABORT	SGNLD_TRGT_ABORT	DEVSEL_TIMING	MSTR_DATA_PRTY_ERR	FAST_B2B_CAP	RESERVED	C66MHZ_CAPA	RESERVED	IO_SPACE_LIMIT	RESERVED	IO_DECODE_32	IO_SPACE_BASE	RESERVED	IO_DECODE_32_0

Bits	Field Name	Description	Type	Reset
31	DET_PAR_ERR	Detected Parity Error	RW	0x0
30	RCVD_SYS_ERR	Received System Error	RW	0x0
29	RCVD_MSTR_ABORT	Received Master Abort	RW	0x0
28	RCVD_TRGT_ABORT	Received Target Error	RW	0x0
27	SGNLD_TRGT_ABORT	Signaled Target Error	RW	0x0
26:25	DEVSEL_TIMING	DEVSEL Timing, Not Applicable for PCI Express hence hardwired to 0	R	0x0
24	MSTR_DATA_PRTY_ERR	Mastered Data Parity Error	RW	0x0
23	FAST_B2B_CAP	Fast Back to Back Capable, Not Applicable for PCI Express hence hardwired to 0	R	0x0
22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21	C66MHZ_CAPA	66MHz Capable, Not Applicable for PCI Express hence hardwired to 0	R	0x0
20:16	RESERVED		R	0x0
15:12	IO_SPACE_LIMIT	IO_Space_Limit	RW	0x0
11:9	RESERVED		R	0x0
8	IODECODE_32	32 or 16 Bit IO Space	R	0x0
7:4	IO_SPACE_BASE	IO_Space_Limit	RW	0x0
3:1	RESERVED		R	0x0
0	IODECODE_32_0	32 or 16 Bit IO Space (CS)	R	0x0

Table 24-590. PCIECTRL_RC_DBICS_MEM_BASE_LIMIT

Address Offset	0x0000 0020		
Physical Address	0x5100 0020 0x5180 0020	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Memory Base and Limit Register		
Type	RW		

31:20	19:16	15:4	3:0
MEM_LIMIT_ADDR	RESERVED	MEM_BASE_ADDR	RESERVED

Bits	Field Name	Description	Type	Reset
31:20	MEM_LIMIT_ADDR	Memory Limit Address	RW	0x0
19:16	RESERVED		R	0x0
15:4	MEM_BASE_ADDR	Memory Base Address	RW	0x0
3:0	RESERVED		R	0x0

Table 24-591. PCIECTRL_RC_DBICS_PREF_MEM_BASE_LIMIT

Address Offset	0x0000 0024		
Physical Address	0x5100 0024 0x5180 0024	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Prefetchable Memory Base and Limit Register		
Type	RW		

31:20	19:17	16	15:4	3:1	0
PREF_MEM_ADDR	RESERVED	MEMDECODE_64	UPPPREF_MEM_ADDR	RESERVED	MEMDECODE_64_0

Bits	Field Name	Description	Type	Reset
31:20	PREF_MEM_ADDR	Upper 12 bits of 32-bit Prefetchable Memory End Address	RW	0x0
19:17	RESERVED		R	0x0
16	MEMDECODE_64	64-Bit Memory Addressing	R	0x0
15:4	UPPPREF_MEM_ADDR	Upper 12 bits of 32-bit Prefetchable Memory start Address	RW	0x0
3:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	MEMDECODE_64_0	64-Bit Memory Addressing	R	0x0

Table 24-592. PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_BASEADDR

Address Offset	0x0000 0028		
Physical Address	0x5100 0028 0x5180 0028	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Upper 32 Bit Prefetachable Base Address Register		
Type	RW		

31:0
ADDRUPP

Bits	Field Name	Description	Type	Reset
31:0	ADDRUPP	Upper 32 Bits of Base Address of Prefetachable Memory Space, Used only if 64 Bit Prefetachable Addresing is enabled	RW	0x0

Table 24-593. PCIECTRL_RC_DBICS_UPPER_32BIT_PREF_LIMITADDR

Address Offset	0x0000 002C		
Physical Address	0x5100 002C 0x5180 002C	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Upper 32 Bit Prefetachable Limit Address Register		
Type	RW		

31:0
ADDRUPP_LIMIT

Bits	Field Name	Description	Type	Reset
31:0	ADDRUPP_LIMIT	Upper 32 Bits of Limit Address of Prefetachable Memory Space, Used only if 64 Bit Prefetachable Addresing is enabled	RW	0x0

Table 24-594. PCIECTRL_RC_DBICS_IO_BASE_LIMIT

Address Offset	0x0000 0030		
Physical Address	0x5100 0030 0x5180 0030	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	IO Base and Limit Register		
Type	RW		

31:16	15:0
UPP16_IOLIMIT	UPP16_IOBASE

Bits	Field Name	Description	Type	Reset
31:16	UPP16_IOLIMIT	Upper 16 IO Limit Address	RW	0x0
15:0	UPP16_IOBASE	Upper 16 IO Base Address	RW	0x0

Table 24-595. PCIECTRL_RC_DBICS_CAPPTR

Address Offset	0x0000 0034		
Physical Address	0x5100 0034 0x5180 0034	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	CapPtr		
Type	RW		

31:8	7:0
RESERVED	CAPTR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	CAPTR	First Capability Pointer (CS)	RW	0x40

Table 24-596. PCIECTRL_RC_DBICS_EXPANSION_ROM_BAR

Address Offset	0x0000 0038		
Physical Address	0x5100 0038 0x5180 0038	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Expansion ROM Base Address Register		
Type	RW		

31:16	15:11	10:1	0
EXROM_ADDRESS	EXROM_ADDRESS_RO	RESERVED	EXP_ROM_EN

Bits	Field Name	Description	Type	Reset
31:16	EXROM_ADDRESS	Expansion ROM address, unmasked (ie programmable)	RW	0x0
15:11	EXROM_ADDRESS_RO	Expansion ROM address, masked.	R	0x0
10:1	RESERVED		R	0x0
0	EXP_ROM_EN	Expansion ROM Enable	RW	0x0

Table 24-597. PCIECTRL_RC_DBICS_BRIDGE_INT

Address Offset	0x0000 003C		
Physical Address	0x5100 003C 0x5180 003C	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Bridge Control and Int Pin and line		
Type	RW		

31:28	27	26	25	24	23	22	21	20	19	18	17	16	15:8	7:0
RESERVED	DT_SERR_EN	DT_STS	SEC_DT	PRI_DT	FAST_B2B_EN	SEC_BUS_RST	MST_ABT_MOD	VGA_16B_DEC	VGA_EN	ISA_EN	SERR_EN	PERR_RESP_EN	INT_PIN	INT_LIN

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	DT_SERR_EN	Discard Timer SERR Enable Status	R	0x0
26	DT_STS	Discard Timer Status	R	0x0
25	SEC_DT	Secondary Discard Timer	R	0x0
24	PRI_DT	Primary Discard Timer	R	0x0
23	FAST_B2B_EN	Fast Back-to-Back Transactions Enable	R	0x0
22	SEC_BUS_RST	Secondary Bus Reset (initiate hot reset)	RW	0x0
21	MST_ABT_MOD	Master Abort Mode	R	0x0
20	VGA_16B_DEC	VGA 16-Bit Decode	RW	0x0

Bits	Field Name	Description	Type	Reset
19	VGA_EN	VGA Enable	RW	0x0
18	ISA_EN	ISA Enable	RW	0x0
17	SERR_EN	SERR Enable	RW	0x0
16	PERR_RESP_EN	Parity Error Response Enable	RW	0x0
15:8	INT_PIN	Interrupt Pin (CS)	R	0x1
7:0	INT_LIN	Interrupt Line	RW	0xff

Table 24-598. PCIECTRL_RC_DBICS_PCIE_CAP

Address Offset	0x0000 0070		
Physical Address	0x5100 0070 0x5180 0070	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	PCI Express Capability structure header		
Type	RW		

31:30	29:25	24	23:20	19:16	15:8	7:0
RESERVED	IM_NUM	SLOT	DEV_TYPE	PCIE_VER	PCIE_NX_PTR	CAP_ID

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:25	IM_NUM	Interrupt Message Number (CS)	RW	0x0
24	SLOT	Slot Implemented (CS)	RW	0x0
23:20	DEV_TYPE	Device/Port Type Read 0x4: RC root port (RC)	R	0x4
19:16	PCIE_VER	PCI Express Capability Version	R	0x2
15:8	PCIE_NX_PTR	Next Capability Pointer (CS)	RW	0x0
7:0	CAP_ID	Capability ID Read 0x10: PCIE	R	0x10

Table 24-599. PCIECTRL_RC_DBICS_DEV_CAP

Address Offset	0x0000 0074		
Physical Address	0x5100 0074 0x5180 0074	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	PCIE Device Capabilities		
Type	RW		

31:28	27:26	25:18	17:16	15	14:12	11:9	8:6	5	4:3	2:0
RESERVED	CAPT_SLOW_PWRLIMIT_SCALE	CAPT_SLOW_PWRLIMIT_VALUE	RESERVED	ROLEBASED_ERR_RPT	UNDEFINED	DEFAULT_EP_L1_ACCPT_LATENCY	DEFAULT_EP_L0S_ACCPT_LATENCY	EXTTAGFIELD_SUPPORT	PHANTOMFUNC	MAX_PAYLOAD_SIZE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CAPT_SLOW_PWRLIMIT_SCALE	Captured Slow Power Scale Value, for Upstream Port Only (CS)	RW	0x0

Bits	Field Name	Description	Type	Reset
25:18	CAPT_SLOW_PWRLIMIT_VALUE	Captured Slow Power Limit Value, for Upstream Port Only (CS)	RW	0x0
17:16	RESERVED		R	0x0
15	ROLEBASED_ERRRPT	Role Based Error Reporting (CS)	RW	0x1
14:12	UNDEFINED	Undefined from PCIe 1.1 onwards	R	0x0
11:9	DEFAULT_EP_L1_ACCPT_LATENCY	Endpoint L1 Acceptable Latency; Must be 0 for RC.	R	0x0
8:6	DEFAULT_EP_L0S_ACCPT_LATENCY	Endpoint L0s Acceptable Latency; Must be 0 for RC.	R	0x0
5	EXTTAGFIELD_SUPPORT	Extended Tag Field Support (CS)	RW	0x0
4:3	PHANTOMFUNC	Phantom Function Support, not SUPPORTED (CS)	RW	0x0
2:0	MAX_PAYLOAD_SIZE	Maximum Payload Size (CS) Read 0x1: 256 Byte	RW	0x1

Table 24-600. PCIECTRL_RC_DBICS_DEV_CAS

Address Offset	0x0000 0078		
Physical Address	0x5100 0078 0x5180 0078	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	PCIE Device Control and Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										TRANS_PEND	AUXP_DET	UR_DET	FT_DET	NFT_DET	COR_DET	INIT_FLR	MRRS			NOSNP_EN	AUXPM_EN	PHFUN_EN	EXTAG_EN	MPS			EN_RO	UR_RE	FT_RE	NFT_RE	COR_RE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	TRANS_PEND	Transaction Pending	R	0x0
20	AUXP_DET	Aux Power Detected	R	0x0
19	UR_DET	Unsupported Request Detected	RW	0x0
18	FT_DET	Fatal Error Detected	RW	0x0
17	NFT_DET	Non-Fatal Error Detected	RW	0x0
16	COR_DET	Correctable Error Detected	RW	0x0
15	INIT_FLR	Reserved	R	0x0
14:12	MRRS	Max_Read_Request_Size	RW	0x2
11	NOSNP_EN	Enable No Snoop	RW	0x1
10	AUXPM_EN	AUX Power PM Enable (Sticky bit) 0x0: Vaux not used by device 0x1: Device can draw Vaux power; Sticky bits will be preserved over reset	RW	0x0
9	PHFUN_EN	Phantom Function Enable	RW	0x0
8	EXTAG_EN	Extended Tag Field Enable	RW	0x0
7:5	MPS	Max_Payload_Size	RW	0x0
4	EN_RO	Enable Relaxed Ordering	RW	0x1
3	UR_RE	Unsupported Request Reporting Enable	RW	0x0
2	FT_RE	Fatal Error Reporting Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
1	NFT_RE	Non-Fatal Error Reporting Enable	RW	0x0
0	COR_RE	Correctable Error Reporting Enable	RW	0x0

Table 24-601. PCIECTRL_RC_DBICS_LNK_CAP

Address Offset	0x0000 007C		
Physical Address	0x5100 007C 0x5180 007C	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	PCIE Link Capabilities		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PORT_NUM</td><td>RESERVED</td><td>ASPM_OPT_COMP</td><td>LNK_BW_not_CAP</td><td>DLL_ACTRPT_CAP</td><td>UNSUP</td><td>CLK_PWR_MGMT</td><td colspan="3">L1_EXIT_LAT</td><td colspan="3">L0S_EXIT_LAT</td><td colspan="2">AS_LINK_PM_SUPPORT</td><td colspan="6">MAX_LINK_WIDTH</td><td colspan="4">MAX_LINK_SPEEDS</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:24	PORT_NUM	Port Number (CS)	RW	0x0
23	RESERVED		R	0x0
22	ASPM_OPT_COMP	ASPM Optionality Compliance (CS)	RW	0x1
21	LNK_BW_not_CAP	Link Bandwidth Notification Capability (CS)	RW	0x1
20	DLL_ACTRPT_CAP	Data Link Layer Active Reporting Capable	R	0x1
19	UNSUP	Unsupported, Surprise Down Error Reporting Capable, Hardwired to 0	R	0x0
18	CLK_PWR_MGMT	Clock Power Management; Hardwired to 0 for DS port (RC); (CS)	RW	0x0
17:15	L1_EXIT_LAT	L1 Exit Latency (CS) Compare CS2	RW	0x6
14:12	L0S_EXIT_LAT	L0s Exit Latency (CS) Compare CS2	RW	0x3
11:10	AS_LINK_PM_SUPPORT	Active State Link PM (ASPM) Support (CS)	RW	0x3
9:4	MAX_LINK_WIDTH	Max Link Width (lanes) (CS)	RW	0x2
3:0	MAX_LINK_SPEEDS	Supported Max Link Speed (CS) Read 0x1: 2.5 GT/s (Gen1) Read 0x2: 5 GT/s (Gen2) Read 0x3: 8 GT/s (Gen3)	RW	0x2

Table 24-602. PCIECTRL_RC_DBICS_LNK_CAS

Address Offset	0x0000 0080		
Physical Address	0x5100 0080 0x5180 0080	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	PCIE Link Control and Status		
Type	RW Wr1toClr		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

LAB_STATUS	LBW_STATUS	DLL_ACT	SLOT_CLK_CONFIG	LINK_TRAIN	UNDEF	NEG_LW	LINK_SPEED	RESERVED	LABIE	LBMIE	HAWD	EN_CPM	EXT_SYN	COM_CLK_CFG	RETRAIN_LINK	LINK_DIS	RCB	RESERVED	ASPM_CTRL

Bits	Field Name	Description	Type	Reset
31	LAB_STATUS	Link Autonomous Bandwidth Status	RW Wr1toClr	0x0
30	LBW_STATUS	Link Bandwidth Management Status	RW Wr1toClr	0x0
29	DLL_ACT	Data Link Layer Active	R	0x0
28	SLOT_CLK_CONFIG	Slot Clock Configuration (CS)	RW	0x1
27	LINK_TRAIN	LINK training	R	0x0
26	UNDEF	Undefined	R	0x0
25:20	NEG_LW	Negotiated Link Width; UNDEFINED UNTIL LINK IS UP.	R	0x1
19:16	LINK_SPEED	Link Speed; UNDEFINED UNTIL LINK IS UP.	R	0x1
15:12	RESERVED		R	0x0
11	LABIE	Link Autonomous Bandwidth Interrupt Enable	RW	0x0
10	LBMIE	Link Bandwidth Management Interrupt Enable	RW	0x0
9	HAWD	Hardware Autonomous Width Disable	R	0x0
8	EN_CPM	Enable Clock Power Management	RW	0x0
7	EXT_SYN	Extended Synch	RW	0x0
6	COM_CLK_CFG	Common Clock Configuration 0x0: Asynchronous reference clocks (ASYNC) 0x1: Distributed common reference clock (COMMON)	RW	0x0
5	RETRAIN_LINK	Retrain Link	RW	0x0
4	LINK_DIS	Link Disable	RW	0x0
3	RCB	Read Completion Boundary (CS) 0x0: 64 Byte 0x1: 128 Byte	RW	0x1
2	RESERVED		R	0x0
1:0	ASPM_CTRL	Active State Link PM Control 0x0: DISABLED 0x1: L0S_ENABLED 0x2: L1_ENABLED 0x3: L0S_AND_L1_ENABLED	RW	0x0

Table 24-603. PCIECTRL_RC_DBICS_SLOT_CAP

Address Offset	0x0000 0084		
Physical Address	0x5100 0084 0x5180 0084	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Slot Capabilities Register		
Type	RW		

31-19	18	17	16-15	14-7	6	5	4	3	2	1	0
PSN	NCCS	EIP	SPLS	SPLV	HPC	HPS	PIP	AIP	MRLSP	PCP	ABP

Bits	Field Name	Description	Type	Reset
31:19	PSN	Physical Slot Number (CS)	RW	0x0
18	NCCS	No Command Complete Support (CS)	RW	0x0
17	EIP	Electromechanical Interlock Present (CS)	RW	0x0
16:15	SPLS	Slot Power Limit Scale (CS)	RW	0x0
14:7	SPLV	Slot Power Limit Value (CS)	RW	0x0
6	HPC	Hot-Plug Capable (CS)	RW	0x0
5	HPS	Hot-Plug Surprise (CS)	RW	0x0
4	PIP	Power Indicator Present (CS)	RW	0x0
3	AIP	Attention Indicator Present (CS)	RW	0x0
2	MRLSP	MRL Sensor Present (CS)	RW	0x0
1	PCP	Power Controller Present (CS)	RW	0x0
0	ABP	Attention Button Present (CS)	RW	0x0

Table 24-604. PCIECTRL_RC_DBICS_SLOT_CAS

Address Offset	0x0000 0088		
Physical Address	0x5100 0088 0x5180 0088	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Slot Control and Status Register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>DSC</td><td>EIS</td><td>PDS</td><td>MRLSS</td><td>CC</td><td>PDC</td><td>MRCSC</td><td>PFD</td><td>ABP</td><td colspan="3">RESERVED</td><td>DSC_EN</td><td>EIC</td><td>PCC</td><td colspan="2">PIC</td><td colspan="2">AIC</td><td>HPI_EN</td><td>CCI_EN</td><td>PDC_EN</td><td>MRLSC_EN</td><td>PFD_EN</td><td>ABP_EN</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	DSC	Data Link Layer State Changed	RW	0x0
23	EIS	Electromechanical Interlock Status	R	0x0
22	PDS	Presence Detect State NO PRESENCE DETECTION IMPLEMENTED: TIED TO 1	R	0x1
21	MRLSS	MRL Sensor State	R	0x0
20	CC	Command Completed	RW	0x0
19	PDC	Presence Detect Changed	RW	0x0
18	MRCSC	MRL Sensor Changed	RW	0x0
17	PFD	Power Fault Detected	RW	0x0
16	ABP	Attention Button Pressed	RW	0x0
15:13	RESERVED		R	0x0
12	DSC_EN	Data Link Layer State Changed Enable	RW	0x0
11	EIC	Electromechanical Interlock Control	RW	0x0
10	PCC	Power Controller Control	RW	0x0
9:8	PIC	Power Indicator Control	RW	0x3
7:6	AIC	Attention Indicator Control	RW	0x3
5	HPI_EN	Hot-Plug Interrupt Enable	RW	0x0
4	CCI_EN	Command Completed Interrupt Enable	RW	0x0
3	PDC_EN	Presence Detect Changed Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
2	MRLSC_EN	MRL Sensor Changed Enable	RW	0x0
1	PFD_EN	Power Fault Detected Enable	RW	0x0
0	ABP_EN	Attention Button Pressed Enable	RW	0x0

Table 24-605. PCIECTRL_RC_DBICS_ROOT_CAC

Address Offset	0x0000 008C		
Physical Address	0x5100 008C 0x5180 008C	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Root Control and Capability Register		
Type	RW		

31:17	16	15:5	4	3	2	1	0
RESERVED	CRSSV	RESERVED	CRSSV_EN	PMEI_EN	SEFE_EN	SENE_EN	SECE_EN

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CRSSV	CRS Software Visibility	R	0x0
15:5	RESERVED		R	0x0
4	CRSSV_EN	CRS Software Visibility Enable	R	0x0
3	PMEI_EN	PME Interrupt Enable	RW	0x0
2	SEFE_EN	System Error on Fatal Error Enable	RW	0x0
1	SENE_EN	System Error on Non-fatal Error Enable	RW	0x0
0	SECE_EN	System Error on Correctable Error Enable	RW	0x0

Table 24-606. PCIECTRL_RC_DBICS_ROOT_STS

Address Offset	0x0000 0090		
Physical Address	0x5100 0090 0x5180 0090	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Root Status Register		
Type	RW		

31:18	17	16	15:0
RESERVED	PME_PND	PME_STS	PME_RID

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	PME_PND	PME Pending	R	0x0
16	PME_STS	PME Status (Sticky bit)	RW	0x0
15:0	PME_RID	PME Requester ID	R	0x0

Table 24-607. PCIECTRL_RC_DBICS_DEV_CAP_2

Address Offset	0x0000 0094		
Physical Address	0x5100 0094 0x5180 0094	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS

Table 24-607. PCIECTRL_RC_DBICS_DEV_CAP_2 (continued)

Description	Device Capabilities 2 Register
Type	R

31–14	13–12	11	10	9	8	7	6	5	4	3–0
RESERVED	TPHC_SP	RESERVED	NOROPR	CASC128_SP	AOC64_SP	AOC32_SP	AOR_SP	ARI_FWD_SP	CPL_TIMEOUT_DIS_SUPPORTED	CPL_TIMEOUT_RNG_SUPPORTED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	TPHC_SP	TPH Completer Supported	R	0x0
11	RESERVED		R	0x0
10	NOROPR	No RO-enabled PR-PR Passing	R	0x1
9	CASC128_SP	128-bit CAS Completer Supported	R	0x0
8	AOC64_SP	64-bit AtomicOp Completer Supported	R	0x0
7	AOC32_SP	32-bit AtomicOp Completer Supported	R	0x0
6	AOR_SP	AtomicOp Routing Supported	R	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	R	0x0
4	CPL_TIMEOUT_DIS_SUPPORTED	Completion Timeout Disable Supported	R	0x1
3:0	CPL_TIMEOUT_RNG_SUPPORTED	Completion Timeout Ranges Supported	R	0xf

Table 24-608. PCIECTRL_RC_DBICS_DEV_CAS_2

Address Offset	0x0000 0098		
Physical Address	0x5100 0098 0x5180 0098	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Device Control 2 Register		
Type	RW		

31–15	14–13	12–11	10	9	8	7	6	5	4	3–0
RESERVED	OBFF_EN	RESERVED	LTR_EN	IDO_CPL_EN	IDO_REQ_EN	AOP_EG_BLK	AOP_REQ_EN	ARI_FWD_SP	CPL_TIMEOUT_DIS	CPL_TIMEOUT_VALUE

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:13	OBFF_EN	OBFF Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
12:11	RESERVED		R	0x0
10	LTR_EN	LTR Mechanism Enable	RW	0x0
9	IDO_CPL_EN	IDO Completion Enable	RW	0x0
8	IDO_REQ_EN	IDO Request Enable	RW	0x0
7	AOP_EG_BLK	AtomicOp Egress Blocking	RW	0x0
6	AOP_REQ_EN	AtomicOp Requester Enable	RW	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	RW	0x0
4	CPL_TIMEOUT_DIS	Completion Timeout Disable	RW	0x0
3:0	CPL_TIMEOUT_VALUE	Completion Timeout Values	RW	0x0

Table 24-609. PCIECTRL_RC_DBICS_LNK_CAP_2

Address Offset	0x0000 009C		
Physical Address	0x5100 009C 0x5180 009C	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	PCIE Link Capabilities 2 Register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>CROSSLINK_SP</td><td colspan="7">SP_LS_VEC</td><td>RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CROSSLINK_SP	Crosslink Supported	R	0x0
7:1	SP_LS_VEC	Supported Link Speeds Vector	R	0x3
0	RESERVED		R	0x0

Table 24-610. PCIECTRL_RC_DBICS_LNK_CAS_2

Address Offset	0x0000 00A0		
Physical Address	0x5100 00A0 0x5180 00A0	**Instance**	PCIe_SS1_RC_CFG_DBICS PCIe_SS2_RC_CFG_DBICS
Description	Link Control and Status 2 Register (Sticky)		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td>LINK_EQ_REQ</td><td>EQ_PH3</td><td>EQ_PH2</td><td>EQ_PH1</td><td>EQ_COMPLETE</td><td>DEEMPH_LEVEL</td><td colspan="4">COMPL_PRST_DEEPH</td><td>COMPL_SOS</td><td>ENT_MOD_COMPL</td><td colspan="3">TX_MARGIN</td><td>SEL_DEEMP</td><td>HW_AUTO_SP_DIS</td><td>ENTR_COMPL</td><td colspan="4">TRGT_LINK_SPEED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21	LINK_EQ_REQ	Link Equilization Request	RW Wr1toClr	0x0
20	EQ_PH3	Equalization Ph3 Success, Gen3 Only	R	0x0
19	EQ_PH2	Equalization Ph2 Success, Gen3 Only	R	0x0
18	EQ_PH1	Equalization Ph1 Success, Gen3 Only	R	0x0
17	EQ_COMPLETE	Equalization Complete, Gen3 Only	R	0x0
16	DEEMPH_LEVEL	Current De-emphasis Level	R	0x1
15:12	COMPL_PRST_DEEPH	Compliance Pre-set/ De-emphasis	RW	0x0
11	COMPL_SOS	Compliance SOS	RW	0x0
10	ENT_MOD_COMPL	Enter Modified Compliance	RW	0x0
9:7	TX_MARGIN	Transmit Margin	RW	0x0
6	SEL_DEEMP	Selectable De-emphasis (CS)	RW	0x0
5	HW_AUTO_SP_DIS	Hardware Autonomous Speed Disable	RW	0x0
4	ENTR_COMPL	Enter Compliance	RW	0x0
3:0	TRGT_LINK_SPEED	Target Link Speed Read 0x1: 2.5 GT/s (Gen1) Read 0x2: 5 GT/s (Gen2) Read 0x3: 8 GT/s (Gen3)	RW	0x2

24.9.7.5 PCIe_SS_PL_CONF Registers

Note

The Section 24.9.7.5.1 lists the implementation-specific (that is, not PCIe-standard) “port logic” - PL registers, mapped in the 4-KiB PCIe configuration space along with the PCIe-standard registers already described. Unlike the standard registers, port logic (PL) registers are not affected by the device type (Type-0/EP vs. Type-1/RC).

Note

Besides at the (DIF CS space) physical addresses listed in below tables, the PCIe PL configuration registers are also locally accessible (aliased) at the (DIF CS2 space) base address 0x5100_1700 (PCIe_SS1) and 0x5180_1700 (PCIe_SS2), without any difference in PL registers behaviour between CS and CS2 spaces. There is also no difference in PL registers behaviour between PCIe wire remote accesses and local (DIF CS,CS2) accesses.

24.9.7.5.1 PCIe_SS_PL_CONF Register Summary

Table 24-611. PCIe_SS1_PL_CONF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_PL_CONF Physical Address
PCIECTRL_PL_LAT_REL_TIM	RW	32	0x0000 0000	0x5100 0700
PCIECTRL_PL_VENDOR_SPECIFIC_DLLP	RW	32	0x0000 0004	0x5100 0704
PCIECTRL_PL_PT_LNK_R	RW	32	0x0000 0008	0x5100 0708
PCIECTRL_PL_ACK_FREQ_ASPM	RW	32	0x0000 000C	0x5100 070C
PCIECTRL_PL_PT_LNK_CTRL_R	RW	32	0x0000 0010	0x5100 0710
PCIECTRL_PL_LN_SKW_R	RW	32	0x0000 0014	0x5100 0714
PCIECTRL_PL_SYMB_N_R	RW	32	0x0000 0018	0x5100 0718
PCIECTRL_PL_SYMB_T_R	RW	32	0x0000 001C	0x5100 071C
PCIECTRL_PL_FL_MSK_R2	RW	32	0x0000 0020	0x5100 0720
PCIECTRL_PL_OBNP_SUBREQ_CTRL	RW	32	0x0000 0024	0x5100 0724
RESERVED	R	32	0x0000 0028	0x5100 0728
RESERVED	R	32	0x0000 002C	0x5100 072C
PCIECTRL_PL_TR_P_STS_R	R	32	0x0000 0030	0x5100 0730
PCIECTRL_PL_TR_NP_STS_R	R	32	0x0000 0034	0x5100 0734
PCIECTRL_PL_TR_C_STS_R	R	32	0x0000 0038	0x5100 0738
PCIECTRL_PL_Q_STS_R	RW	32	0x0000 003C	0x5100 073C
PCIECTRL_PL_VC_TR_A_R1	R	32	0x0000 0040	0x5100 0740
PCIECTRL_PL_VC_TR_A_R2	R	32	0x0000 0044	0x5100 0744
PCIECTRL_PL_VC0_PR_Q_C	RW	32	0x0000 0048	0x5100 0748
PCIECTRL_PL_VC0_NPR_Q_C	RW	32	0x0000 004C	0x5100 074C
PCIECTRL_PL_VC0_CR_Q_C	RW	32	0x0000 0050	0x5100 0750
PCIECTRL_PL_WIDTH_SPEED_CTL	RW	32	0x0000 010C	0x5100 080C
PCIECTRL_PL_PHY_STS_R	R	32	0x0000 0110	0x5100 0810
PCIECTRL_PL_PHY_CTRL_R	RW	32	0x0000 0114	0x5100 0814
PCIECTRL_PL_MSI_CTRL_ADDRESS	RW	32	0x0000 0120	0x5100 0820
PCIECTRL_PL_MSI_CTRL_UPPER_ADDRESS	RW	32	0x0000 0124	0x5100 0824
PCIECTRL_PL_MSI_CTRL_INT_ENABLE_N (1)	RW	32	0x0000 0128 + (0xc*N)	0x5100 0828 + (0xc*N)
PCIECTRL_PL_MSI_CTRL_INT_MASK_N (1)	RW	32	0x0000 012C + (0xc*N)	0x5100 082C + (0xc*N)

Table 24-611. PCIe_SS1_PL_CONF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_PL_CONF Physical Address
PCIECTRL_PL_MSI_CTRL_INT_STATUS_N [(1)]	RW	32	0x0000 0130 + (0xc*N)	0x5100 0830 + (0xc*N)
PCIECTRL_PL_MSI_CTRL_GPIO	RW	32	0x0000 0188	0x5100 0888
PCIECTRL_PL_PIPE_LOOPBACK	RW	32	0x0000 01B8	0x5100 08B8
PCIECTRL_PL_DBI_RO_WR_EN	RW	32	0x0000 01BC	0x5100 08BC
PCIECTRL_PL_AXIS_SLV_ERR_RESP	RW	32	0x0000 01D0	0x5100 08D0
PCIECTRL_PL_AXIS_SLV_TIMEOUT	RW	32	0x0000 01D4	0x5100 08D4
PCIECTRL_PL_IATU_INDEX	RW	32	0x0000 0200	0x5100 0900
PCIECTRL_PL_IATU_REG_CTRL_1	RW	32	0x0000 0204	0x5100 0904
PCIECTRL_PL_IATU_REG_CTRL_2	RW	32	0x0000 0208	0x5100 0908
PCIECTRL_PL_IATU_REG_LOWER_BASE	RW	32	0x0000 020C	0x5100 090C
PCIECTRL_PL_IATU_REG_UPPER_BASE	RW	32	0x0000 0210	0x5100 0910
PCIECTRL_PL_IATU_REG_LIMIT	RW	32	0x0000 0214	0x5100 0914
PCIECTRL_PL_IATU_REG_LOWER_TARGET	RW	32	0x0000 0218	0x5100 0918
PCIECTRL_PL_IATU_REG_UPPER_TARGET	RW	32	0x0000 021C	0x5100 091C
PCIECTRL_PL_IATU_REG_CTRL_3	R	32	0x0000 0220	0x5100 0920

(1) N=0 to 7

Table 24-612. PCIe_SS2_PL_CONF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_PL_CONF Physical Address
PCIECTRL_PL_LAT_REL_TIM	RW	32	0x0000 0000	0x5180 0700
PCIECTRL_PL_VENDOR_SPECIFIC_DLLP	RW	32	0x0000 0004	0x5180 0704
PCIECTRL_PL_PT_LNK_R	RW	32	0x0000 0008	0x5180 0708
PCIECTRL_PL_ACK_FREQ_ASPM	RW	32	0x0000 000C	0x5180 070C
PCIECTRL_PL_PT_LNK_CTRL_R	RW	32	0x0000 0010	0x5180 0710
PCIECTRL_PL_LN_SKW_R	RW	32	0x0000 0014	0x5180 0714
PCIECTRL_PL_SYMB_N_R	RW	32	0x0000 0018	0x5180 0718
PCIECTRL_PL_SYMB_T_R	RW	32	0x0000 001C	0x5180 071C
PCIECTRL_PL_FL_MSK_R2	RW	32	0x0000 0020	0x5180 0720
PCIECTRL_PL_OBNP_SUBREQ_CTRL	RW	32	0x0000 0024	0x5180 0724
RESERVED	R	32	0x0000 0028	0x5180 0728
RESERVED	R	32	0x0000 002C	0x5180 072C
PCIECTRL_PL_TR_P_STS_R	R	32	0x0000 0030	0x5180 0730
PCIECTRL_PL_TR_NP_STS_R	R	32	0x0000 0034	0x5180 0734
PCIECTRL_PL_TR_C_STS_R	R	32	0x0000 0038	0x5180 0738
PCIECTRL_PL_Q_STS_R	RW	32	0x0000 003C	0x5180 073C
PCIECTRL_PL_VC_TR_A_R1	R	32	0x0000 0040	0x5180 0740
PCIECTRL_PL_VC_TR_A_R2	R	32	0x0000 0044	0x5180 0744
PCIECTRL_PL_VC0_PR_Q_C	RW	32	0x0000 0048	0x5180 0748
PCIECTRL_PL_VC0_NPR_Q_C	RW	32	0x0000 004C	0x5180 074C
PCIECTRL_PL_VC0_CR_Q_C	RW	32	0x0000 0050	0x5180 0750
PCIECTRL_PL_WIDTH_SPEED_CTL	RW	32	0x0000 010C	0x5180 080C
PCIECTRL_PL_PHY_STS_R	R	32	0x0000 0110	0x5180 0810
PCIECTRL_PL_PHY_CTRL_R	RW	32	0x0000 0114	0x5180 0814
PCIECTRL_PL_MSI_CTRL_ADDRESS	RW	32	0x0000 0120	0x5180 0820

Table 24-612. PCIe_SS2_PL_CONF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_PL_CONF Physical Address
PCIECTRL_PL_MSI_CTRL_UPPER_ADDRESS	RW	32	0x0000 0124	0x5180 0824
PCIECTRL_PL_MSI_CTRL_INT_ENABLE_N (1)	RW	32	0x0000 0128 + (0xc*N)	0x5180 0828 + (0xc*N)
PCIECTRL_PL_MSI_CTRL_INT_MASK_N (1)	RW	32	0x0000 012C + (0xc*N)	0x5180 082C + (0xc*N)
PCIECTRL_PL_MSI_CTRL_INT_STATUS_N (1)	RW	32	0x0000 0130 + (0xc*N)	0x5180 0830 + (0xc*N)
PCIECTRL_PL_MSI_CTRL_GPIO	RW	32	0x0000 0188	0x5180 0888
PCIECTRL_PL_PIPE_LOOPBACK	RW	32	0x0000 01B8	0x5180 08B8
PCIECTRL_PL_DBI_RO_WR_EN	RW	32	0x0000 01BC	0x5180 08BC
PCIECTRL_PL_AXIS_SLV_ERR_RESP	RW	32	0x0000 01D0	0x5180 08D0
PCIECTRL_PL_AXIS_SLV_TIMEOUT	RW	32	0x0000 01D4	0x5180 08D4
PCIECTRL_PL_IATU_INDEX	RW	32	0x0000 0200	0x5180 0900
PCIECTRL_PL_IATU_REG_CTRL_1	RW	32	0x0000 0204	0x5180 0904
PCIECTRL_PL_IATU_REG_CTRL_2	RW	32	0x0000 0208	0x5180 0908
PCIECTRL_PL_IATU_REG_LOWER_BASE	RW	32	0x0000 020C	0x5180 090C
PCIECTRL_PL_IATU_REG_UPPER_BASE	RW	32	0x0000 0210	0x5180 0910
PCIECTRL_PL_IATU_REG_LIMIT	RW	32	0x0000 0214	0x5180 0914
PCIECTRL_PL_IATU_REG_LOWER_TARGET	RW	32	0x0000 0218	0x5180 0918
PCIECTRL_PL_IATU_REG_UPPER_TARGET	RW	32	0x0000 021C	0x5180 091C
PCIECTRL_PL_IATU_REG_CTRL_3	R	32	0x0000 0220	0x5180 0920

(1) N=0 to 7

24.9.7.5.2 PCIe_SS_PL_CONF Register Description

Table 24-613. PCIECTRL_PL_LAT_REL_TIM

Address Offset	0x0000 0000		
Physical Address	0x5100 0700 0x5180 0700	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Ack Latency and Replay Timer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REPLAY_TIME_LIMIT																ACK_LATENCY_TIME_LIMIT															

Bits	Field Name	Description	Type	Reset
31:16	REPLAY_TIME_LIMIT	The replay timer expires when it reaches this limit; The core initiates a replay upon reception of a Nak or when the replay timer expires; The default value depends on number of bytes (NB) per cycle, which is defined by the maximum core base frequency of the device PCIe core, corresponding to 250 MHz for PCIe-Gen2 (5 Gbps) operation. The default is then updated based on the Negotiated Link Width and Max_Payload_Size; Note: If operating at 5 Gb/s, then the rounded-up value of an additional (153/CX_NB) cycles is added, where CX_NB correspond to the number of PCIEPCS 8-bit input symbols per single 16-bit lane, that is, CX_NB=2. This means at 5Gbps, 77 extra cycles should be considered for the replay time limit. This is for additional internal processing for received TLPs and transmitted DLLPs.	RW	0xc0

Bits	Field Name	Description	Type	Reset
15:0	ACK_LATENCY_TIME_LIMIT	The Ack/Nak latency timer expires when it reaches this limit; The default value depends on number of bytes (NB) per cycle, which is defined by the maximum core base frequency of the device PCIe core, corresponding to 250 MHz for PCIe-Gen2 (5 Gbps) operation. The default is then updated based on the Negotiated Link Width and Max_Payload_Size. Note: If operating at 5 Gb/s, then the rounded-up value of an additional (51 /CX_NB) cycles is added, where CX_NB correspond to the number of PCIEPCS 8-bit input symbols per single 16-bit lane, that is, CX_NB=2. This means at 5Gbps, 26 extra cycles should be considered for the acknowledge latency time limit. This is for additional internal processing for received TLPs and transmitted DLLPs.	RW	0x40

Table 24-614. PCIECTRL_PL_VENDOR_SPECIFIC_DLLP

Address Offset	0x0000 0004		
Physical Address	0x5100 0704 0x5180 0704	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Vendor Specific DLLP Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VEN_DLLP_REG																															

Bits	Field Name	Description	Type	Reset
31:0	VEN_DLLP_REG	To send custom DLLP, write 8-bit DLLP Type and 24-bits of Payload data, then set PT_LNK_CTRL_R[0]	RW	0xFFFFFFFF

Table 24-615. PCIECTRL_PL_PT_LNK_R

Address Offset	0x0000 0008		
Physical Address	0x5100 0708 0x5180 0708	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Port Force Link Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOW_POWER_ENTR_CNT								RESERVED		FORCED_LINK_COMMAND						FORCE_LINK	RESERVED			FORCED_LTSSM_STATE				LINK_NUM							

Bits	Field Name	Description	Type	Reset
31:24	LOW_POWER_ENTR_CNT	The Power Management state will wait for this many clock cycles for the associated completion of a CfgWr to D-state register to go low-power; This register is intended for applications that do not let the core handle a completion for configuration request to the PMCSCR register; Note: Only used in the DM core (in EP mode), EP core, and the Upstream Port of a Switch	RW	0x7
23:22	RESERVED		R	0x0
21:16	FORCED_LINK_COMMAND	Link command transmitted by setting Force_Link (bit 15);	RW	0x0
15	FORCE_LINK	Forces the LTSSM state and the Link command specified in this register; Self-clearing 0x1: FORCE	RW	0x0
14:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11:8	FORCED_LTSSM_STATE	LTSSM state forced by setting Force_Link (bit 15)	RW	0x0
7:0	LINK_NUM	Link Number; Not used for Endpoint	RW	0x4

Table 24-616. PCIECTRL_PL_ACK_FREQ_ASPM

Address Offset	0x0000 000C		
Physical Address	0x5100 070C 0x5180 070C	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Ack Frequency and L0-L1 ASPM Control Register (Sticky)		
Type	RW		

31	30	29:27	26:24	23:16	15:8	7:0
RESERVED	L1_ENTR_WO_L0S	L1_ENTR_LAT	L0S_ENTR_LAT	COMMOM_CLK_N_FTS	N_FTS	ACK_FREQ

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0x0
30	L1_ENTR_WO_L0S	Enter ASPM L1 without receive in L0s; Allow core to enter ASPM L1 even when link partner did not go to L0s (receive is not in L0s); When not set, core goes to ASPM L1 only after idle period during which both receive and transmit are in L0s	RW	0x0
29:27	L1_ENTR_LAT	L1 Entrance Latency 0x0: 1 uS 0x1: 2 uS 0x2: 4 uS 0x3: 8 uS 0x4: 16 uS 0x5: 32 uS 0x6: 64 uS 0x7: 64 uS (alternate encoding)	RW	0x3
26:24	L0S_ENTR_LAT	L0s Entrance Latency; Values correspond to: 0b000: 1 us 0b001: 2 us 0b010: 3 us 0b011: 4 us 0b100: 5 us 0b101: 6 us 0b110: 7 us 0b111: 7 us (alternate encoding)	RW	0x3
23:16	COMMOM_CLK_N_FTS	Alternative N_FTS value, for common clock mode	RW	0xf
15:8	N_FTS	Number of Fast Training Sequence (FTS) ordered sets to be transmitted when exiting L0s to L0; The maximum that can be requested is 255; Value 0 is not supported, and may cause LTSSM to go into Recovery upon L0s exit	RW	0xf
7:0	ACK_FREQ	Ack Frequency; Number of pending ACKs accumulated before sending an ACK DLLP	RW	0x0

Table 24-617. PCIECTRL_PL_PT_LNK_CTRL_R

Address Offset	0x0000 0010		
Physical Address	0x5100 0710 0x5180 0710	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Port Link Control Register (Sticky)		

Table 24-617. PCIECTRL_PL_PT_LNK_CTRL_R (continued)

Type	RW

31-24	23	22	21-16	15-8	7	6	5	4	3	2	1	0
RESERVED	CROSSLINK_ACT	CROSSLINK_EN	LINK_MODE	RESERVED	FAST_LINK	RESERVED	DL_EN	RESERVED	RESET_ASSERT	LB_EN	SCRAMBLE_DIS	VEN_DLLP_REQ

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23	CROSSLINK_ACT	Crosslink Active	R	0x0
22	CROSSLINK_EN	Crosslink Enable	RW	0x0
21:16	LINK_MODE	Link Mode Enable; Write 1 to bit N to enable (2**N)-lane mode 0x01: _1x 0x03: _2x 0x07: _4x	RW	0x3
15:8	RESERVED		R	0x1
7	FAST_LINK	Fast Link Mode	RW	0x0
6	RESERVED		R	0x0
5	DL_EN	DLL Link Enable	RW	0x1
4	RESERVED		R	0x0
3	RESET_ASSERT	Reset Assert	RW	0x0
2	LB_EN	Loopback Enable	RW	0x0
1	SCRAMBLE_DIS	Scramble Disable	RW	0x0
0	VEN_DLLP_REQ	Vendor Specific DLLP transmit Request	RW	0x0

Table 24-618. PCIECTRL_PL_LN_SKW_R

Address Offset	0x0000 0014		
Physical Address	0x5100 0714 0x5180 0714	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Lane Skew Register (Sticky)		
Type	RW		

31	30-26	25	24	23-0
DIS_L2L_SKEW	RESERVED	ACKNAK_DIS	FC_DIS	LANE_SKEW

Bits	Field Name	Description	Type	Reset
31	DIS_L2L_SKEW	Disable Lane-to-Lane Deskew	RW	0x0
30:26	RESERVED	Reserved	R	0x0
25	ACKNAK_DIS	Ack/Nak Disable	RW	0x0
24	FC_DIS	Flow Control Disable	RW	0x0
23:0	LANE_SKEW	Insert Lane Skew for Transmit	RW	0x0

Table 24-619. PCIECTRL_PL_SYMB_N_R

Address Offset	0x0000 0018		
Physical Address	0x5100 0718 0x5180 0718	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Timer Control and Symbol Number Register (Sticky)		
Type	RW		

31:24	23:19	18:14	13:8	7:0
RESERVED	ACK_LATENCY_INC	REPLAY_ADJ	RESERVED	MAX_FUNC

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:19	ACK_LATENCY_INC	Timer Modifier for Ack/Nak Latency Timer	RW	0x0
18:14	REPLAY_ADJ	Timer Modifier for Replay Timer	RW	0x1
13:8	RESERVED		R	0x0
7:0	MAX_FUNC	Configuration Requests targeted at function numbers above this value will be returned with UR (unsupported request).	RW	0x0

Table 24-620. PCIECTRL_PL_SYMB_T_R

Address Offset	0x0000 001C		
Physical Address	0x5100 071C 0x5180 071C	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Symbol Timer Register and Filter Mask Register 1 (Sticky)		
Type	RW		

31:16	15	14:11	10:0
FLT_MSK_1	DIS_FC_TIM	RESERVED	SKP_INT

Bits	Field Name	Description	Type	Reset
31:16	FLT_MSK_1	Mask RADM Filtering and Error Handling Rules: Mask 1	RW	0x0
15	DIS_FC_TIM	Disable FC Watchdog Timer	RW	0x0
14:11	RESERVED	Reserved	R	0x0
10:0	SKP_INT	SKP Interval Value minus one, PIPE clock cycles. (1 PIPE cycle = 2 symbols in 16-bit-per-lane PIPE)	RW	0x280

Table 24-621. PCIECTRL_PL_FL_MSK_R2

Address Offset	0x0000 0020		
Physical Address	0x5100 0720 0x5180 0720	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Filter Mask Register 2 (Sticky)		
Type	RW		

31:0
FLT_MSK_2

Bits	Field Name	Description	Type	Reset
31:0	FLT_MSK_2	Mask RADM Filtering and Error Handling Rules: Mask 2	RW	0x0

Table 24-622. PCIECTRL_PL_OBNP_SUBREQ_CTRL

Address Offset	0x0000 0024

Table 24-622. PCIECTRL_PL_OBNP_SUBREQ_CTRL (continued)

Physical Address	0x5100 0724 0x5180 0724	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	AXI Multiple Outbound Decomposed NP SubRequests Control Register (Sticky)		
Type	RW		

31–1	0
RESERVED	EN_OBNP_SUBREQ

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EN_OBNP_SUBREQ	Enable AXI Multiple Outbound Decomposed NP Sub-Requests.	RW	0x1

Table 24-623. PCIECTRL_PL_TR_P_STS_R

Address Offset	0x0000 0030		
Physical Address	0x5100 0730 0x5180 0730	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Transmit Posted FC Credit Status Register (Sticky)		
Type	R		

31–20	19–12	11–0
RESERVED	PH_CRDT	PD_CRDT

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x0
19:12	PH_CRDT	Transmit Posted Header FC Credits	R	0x0
11:0	PD_CRDT	Transmit Posted Data FC Credits	R	0x0

Table 24-624. PCIECTRL_PL_TR_NP_STS_R

Address Offset	0x0000 0034		
Physical Address	0x5100 0734 0x5180 0734	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Transmit Non-Posted FC Credit Status Register (Sticky)		
Type	R		

31–20	19–12	11–0
RESERVED	NPH_CRDT	NPD_CRDT

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x0
19:12	NPH_CRDT	Transmit Non-Posted Header FC Credits	R	0x0
11:0	NPD_CRDT	Transmit Non-Posted Data FC Credits	R	0x0

Table 24-625. PCIECTRL_PL_TR_C_STS_R

Address Offset	0x0000 0038		
Physical Address	0x5100 0738 0x5180 0738	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Transmit Completion FC Credit Status Register (Sticky)		

Table 24-625. PCIECTRL_PL_TR_C_STS_R (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CPLH_CRDT								CPLD_CRDT											

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x0
19:12	CPLH_CRDT	Transmit Completion Header FC Credits	R	0x0
11:0	CPLD_CRDT	Transmit Completion Data FC Credits	R	0x0

Table 24-626. PCIECTRL_PL_Q_STS_R

Address Offset	0x0000 003C		
Physical Address	0x5100 073C 0x5180 073C	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Queue Status Register (Sticky)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FC_LATENCY_OVR_EN	RESERVED		FC_LATENCY_OVR													RESERVED													RCVQ_not_EMPTY	RTYB_not_EMPTY	CRDT_not_RTRN

Bits	Field Name	Description	Type	Reset
31	FC_LATENCY_OVR_EN	FC Latency Timer Override Enable	RW	0x0
30:29	RESERVED		R	0x0
28:16	FC_LATENCY_OVR	FC Latency Timer Override Value	RW	0x0
15:3	RESERVED		R	0x0
2	RCVQ_not_EMPTY	Received Queue Not Empty	R	0x0
1	RTYB_not_EMPTY	Transmit Retry Buffer Not Empty	R	0x0
0	CRDT_not_RTRN	Received TLP FC Credits Not Returned	R	0x0

Table 24-627. PCIECTRL_PL_VC_TR_A_R1

Address Offset	0x0000 0040		
Physical Address	0x5100 0740 0x5180 0740	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	VC Transmit Arbitration Register 1 (Sticky)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRR_VC3								WRR_VC2								WRR_VC1								WRR_VC0							

Bits	Field Name	Description	Type	Reset
31:24	WRR_VC3	WRR Weight for VC3	R	0x0
23:16	WRR_VC2	WRR Weight for VC2	R	0x0
15:8	WRR_VC1	WRR Weight for VC1	R	0x0
7:0	WRR_VC0	WRR Weight for VC0	R	0xf

Table 24-628. PCIECTRL_PL_VC_TR_A_R2

Address Offset	0x0000 0044

Table 24-628. PCIECTRL_PL_VC_TR_A_R2 (continued)

Physical Address	0x5100 0744 0x5180 0744	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	VC Transmit Arbitration Register 2 (Sticky)		
Type	R		

31–24	23–16	15–8	7–0
WRR_VC7	WRR_VC6	WRR_VC5	WRR_VC4

Bits	Field Name	Description	Type	Reset
31:24	WRR_VC7	WRR Weight for VC7	R	0x0
23:16	WRR_VC6	WRR Weight for VC6	R	0x0
15:8	WRR_VC5	WRR Weight for VC5	R	0x0
7:0	WRR_VC4	WRR Weight for VC4	R	0x0

Table 24-629. PCIECTRL_PL_VC0_PR_Q_C

Address Offset	0x0000 0048		
Physical Address	0x5100 0748 0x5180 0748	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	VC0 Posted Receive Queue Control (Sticky)		
Type	RW		

31	30	29–24	23–21	20	19–12	11–0
STRICT_VC_PRIORITY	ORDERING_RULES	RESERVED	P_QMODE	RESERVED	P_HCRD	P_DCRD

Bits	Field Name	Description	Type	Reset
31	STRICT_VC_PRIORITY	VC Ordering for Receive Queues 0x0: ROUND_ROBIN 0x1: STRICT Ordering by VC	RW	0x0
30	ORDERING_RULES	VC0 TLP Type Ordering Rules 0x0: STRICT Posted, then Completion, then Non-Posted 0x1: STANDARD As per PCIe standard	RW	0x1
29:24	RESERVED		R	0x0
23:21	P_QMODE	VC0 Poster TLP Queue Mode Read 0x1: STORE_AND_FORWARD Read 0x2: CUT_THROUGH Read 0x4: BYPASS Others: Reserved	RW	0x1
20	RESERVED		R	0x0
19:12	P_HCRD	VC0 Posted Header Credits	R	0x15
11:0	P_DCRD	VC0 Posted Data Credits	R	0x2d

Table 24-630. PCIECTRL_PL_VC0_NPR_Q_C

Address Offset	0x0000 004C		
Physical Address	0x5100 074C 0x5180 074C	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF

Table 24-630. PCIECTRL_PL_VC0_NPR_Q_C (continued)

Description	VC0 Non-Posted Receive Queue Control (Sticky)
Type	RW

31–24	23–21	20	19–12	11–0
RESERVED	NP_QMODE	RESERVED	NP_HCRD	NP_DCRD

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:21	NP_QMODE	VC0 Non-Poster TLP Queue Mode Read 0x1: STORE_AND_FORWARD Read 0x2: CUT_THROUGH Read 0x4: BYPASS Others: Reserved	RW	0x1
20	RESERVED		R	0x0
19:12	NP_HCRD	VC0 Non-Posted Header Credits	R	0x15
11:0	NP_DCRD	VC0 Non-Posted Data Credits	R	0x5

Table 24-631. PCIECTRL_PL_VC0_CR_Q_C

Address Offset	0x0000 0050		
Physical Address	0x5100 0750 0x5180 0750	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	VC0 Completion Receive Queue Control (Sticky)		
Type	RW		

31–24	23–21	20	19–12	11–0
RESERVED	CPL_QMODE	RESERVED	CPL_HCRD	CPL_DCRD

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:21	CPL_QMODE	VC0 Completion TLP Queue Mode Read 0x1: STORE_AND_FORWARD Read 0x2: CUT_THROUGH Read 0x4: BYPASS	RW	0x4
20	RESERVED		R	0x0
19:12	CPL_HCRD	VC0 Completion Header Credits	R	0x0
11:0	CPL_DCRD	VC0 Completion Data Credits	R	0x0

Table 24-632. PCIECTRL_PL_WIDTH_SPEED_CTL

Address Offset	0x0000 010C		
Physical Address	0x5100 080C 0x5180 080C	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Link Width and Speed Change Control Register (Sticky)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CFG_UP_SEL_DEEMPH	CFG_TX_COMPLIANCE_RCV	CFG_PHY_TXSWING	CFG_DIRECTED_SPEED_CHANGE	CFG_LANE_EN	CFG_Gen2_N_FTS

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20	CFG_UP_SEL_DEEMPH	Used to set the de-emphasis level for Upstream Ports	RW	0x0
19	CFG_TX_COMPLIANCE_RCV	Config Tx Compliance Receive Bit	RW	0x0
18	CFG_PHY_TXSWING	Config PHY Tx Swing	RW	0x0
17	CFG_DIRECTED_SPEED_CHANGE	Directed Speed Change	RW	0x1
16:8	CFG_LANE_EN	Predetermined Number of Lanes	RW	0x2
7:0	CFG_Gen2_N_FTS	Number of Fast Training Sequences	RW	0xf

Table 24-633. PCIECTRL_PL_PHY_STS_R

Address Offset	0x0000 0110		
Physical Address	0x5100 0810 0x5180 0810	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	PHY Status Register (Sticky)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_STS																															

Bits	Field Name	Description	Type	Reset
31:0	PHY_STS	PHY Status	R	0x0

Table 24-634. PCIECTRL_PL_PHY_CTRL_R

Address Offset	0x0000 0114		
Physical Address	0x5100 0814 0x5180 0814	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	PHY Control Register (Sticky)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_CTRL																															

Bits	Field Name	Description	Type	Reset
31:0	PHY_CTRL	PHY Control	RW	0x0

Table 24-635. PCIECTRL_PL_MSI_CTRL_ADDRESS

Address Offset	0x0000 0120		
Physical Address	0x5100 0820 0x5180 0820	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	MSI Controller Address Register (RC-mode MSI receiver)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSI_CTRL_ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	MSI_CTRL_ADDRESS		RW	0x0

Table 24-636. PCIECTRL_PL_MSI_CTRL_UPPER_ADDRESS

Address Offset	0x0000 0124		
Physical Address	0x5100 0824 0x5180 0824	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	MSI Controller Upper Address Register (RC-mode MSI receiver)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSI_CTRL_UPPER_ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	MSI_CTRL_UPPER_ADDRESS		RW	0x0

Table 24-637. PCIECTRL_PL_MSI_CTRL_INT_ENABLE_N

Address Offset	0x0000 0128		
Physical Address	0x5100 0828 + (0xc*N) 0x5180 0828 + (0xc*N)	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	MSI Controller Interrupt #N(1) Enable Register (RC-mode MSI receiver) with N = MSI data [7:5] and ENABLE[i] = enable MSI vector #i, with i = MSI data [4:0]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSI_CTRL_INT_ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	MSI_CTRL_INT_ENABLE	Status of an enabled bit (vectors) is set upon incoming MSI.	RW	0x0

(1) N=0 to 7

Table 24-638. PCIECTRL_PL_MSI_CTRL_INT_MASK_N

Address Offset	0x0000 012C		
Physical Address	0x5100 082C + (0xc*N) 0x5180 082C + (0xc*N)	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	MSI Controller Interrupt #N(1) Mask Register (RC-mode MSI receiver) with N = MSI data [7:5] and MASK[i] = mask of MSI vector #i, with i = MSI data [4:0]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSI_CTRL_INT_MASK																															

Bits	Field Name	Description	Type	Reset
31:0	MSI_CTRL_INT_MASK	Status of a masked bit (vector) triggers no IRQ to MPU when set.	RW	0x0

(1) N=0 to 7

Table 24-639. PCIECTRL_PL_MSI_CTRL_INT_STATUS_N

Address Offset	0x0000 0130		
Physical Address	0x5100 0830 + (0xc*N) 0x5180 0830 + (0xc*N)	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF

Table 24-639. PCIECTRL_PL_MSI_CTRL_INT_STATUS_N (continued)

Description	MSI Controller Interrupt #N(1) Status Register (RC-mode MSI receiver) with N = MSI data [7:5] and STATUS[i] = status of MSI vector #i, with i = MSI data [4:0]
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSI_CTRL_INT_STATUS																															

Bits	Field Name	Description	Type	Reset
31:0	MSI_CTRL_INT_STATUS	Status of an enabled bit (vectors) is set upon incoming MSI.	RW	0x0

(1) N=0 to 7

Table 24-640. PCIECTRL_PL_MSI_CTRL_GPIO

Address Offset	0x0000 0188		
Physical Address	0x5100 0888 0x5180 0888	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	MSI Controller General Purpose IO Register (RC-mode MSI receiver)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSI_CTRL_GPIO																															

Bits	Field Name	Description	Type	Reset
31:0	MSI_CTRL_GPIO		RW	0x0

Table 24-641. PCIECTRL_PL_PIPE_LOOPBACK

Address Offset	0x0000 01B8		
Physical Address	0x5100 08B8 0x5180 08B8	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	PIPE loopback control register (Sticky)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOOPBACK_EN	RESERVED																														

Bits	Field Name	Description	Type	Reset
31	LOOPBACK_EN	PIPE Loopback Enable	RW	0x0
30:0	RESERVED		R	0x0

Table 24-642. PCIECTRL_PL_DBI_RO_WR_EN

Address Offset	0x0000 01BC		
Physical Address	0x5100 08BC 0x5180 08BC	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	DIF Read-Only register Write Enable (Sticky)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CX_DBI_RO_WR_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CX_DBI_RO_WR_EN	Control the writability over DIF of certain configuration fields that are RO over the PCIe wire 0x0: WRDIS, RO fields are also RO over DIF; Use for RC mode (Type-1) config to mimic PCIe wire access when using DIF 0x1: WREN, Some RO fields are writable over DIF	RW	0x1

Table 24-643. PCIECTRL_PL_AXIS_SLV_ERR_RESP

Address Offset	0x0000 01D0		
Physical Address	0x5100 08D0 0x5180 08D0	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	AXI Slave Error Response Register (Sticky)		
Type	RW		

31:4	3	2	1	0
RESERVED	RESET_TIMEOUT_ERR_MAP	NO_VID_ERR_MAP	DBI_ERR_MAP	SLAVE_ERR_MAP

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RESET_TIMEOUT_ERR_MAP	Graceful Reset and Link Timeout Slave Error Response Mapping	RW	0x0
2	NO_VID_ERR_MAP	Vendor ID Non-existent Slave Error Response Mapping	RW	0x0
1	DBI_ERR_MAP	DIF Slave Error Response Mapping	RW	0x0
0	SLAVE_ERR_MAP	Global Slave Error Response Mapping	RW	0x0

Table 24-644. PCIECTRL_PL_AXIS_SLV_TIMEOUT

Address Offset	0x0000 01D4		
Physical Address	0x5100 08D4 0x5180 08D4	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	Link Down AXI Slave Timeout Register (Sticky)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	FLUSH_EN	TIMEOUT_VALUE

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	FLUSH_EN	Enable flush	RW	0x0
7:0	TIMEOUT_VALUE	Timeout Value (ms)	RW	0x32

Table 24-645. PCIECTRL_PL_IATU_INDEX

Address Offset	0x0000 0200		
Physical Address	0x5100 0900 0x5180 0900	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Viewport Register: makes the registers of the corresponding iATU region accessible.		
Type	RW		

31	30:4	3:0
REGION_DIRECTION	RESERVED	REGION_INDEX

Bits	Field Name	Description	Type	Reset
31	REGION_DIRECTION	0x0: OUTBOUND 0x1: INBOUND	RW	0x0
30:4	RESERVED		R	0x0
3:0	REGION_INDEX	Outbound region, from 0 to 15. Inbound region, from 0 to 3.	RW	0x0

Table 24-646. PCIECTRL_PL_IATU_REG_CTRL_1

Address Offset	0x0000 0204		
Physical Address	0x5100 0904 0x5180 0904	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Control 1 Register		
Type	RW		

31:25	24:20	19:18	17:16	15:11	10:9	8	7:5	4:0
RESERVED	FUNCTION_NUMBER	RESERVED	AT	RESERVED	ATTR	TD	TC	TYPE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:20	FUNCTION_NUMBER	Outbound: F.N; applied to outgoing TLP (RID) with matching address Inbound: F.N.-match criteria for incoming TLP (if Function_Number_match_enable=1)	RW	0x0
19:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	AT	Outbound: AT applied to outgoing TLP with matching addess Inbound: AT-match criteria for matching TLP (if AT_match_enable=1)	RW	0x0
15:11	RESERVED		R	0x0
10:9	ATTR	Outbound: ATTR applied to outgoing TLP with matching addess Inbound: ATTR-match criteria (if ATTR_match_enable=1)	RW	0x0
8	TD	Outbound: TD applied to outgoing TLP with matching addess Inbound: TD-match criteria (if TD_match_enable=1)	RW	0x0
7:5	TC	Outbound: TC applied to outgoing TLP with matching addess Inbound: TC-match criteria (if TC_match_enable=1)	RW	0x0
4:0	TYPE	Outbound: TYPE applied to outgoing TLP with matching addess Inbound: TYPE-match criteria	RW	0x0

Table 24-647. PCIECTRL_PL_IATU_REG_CTRL_2

Address Offset	0x0000 0208		
Physical Address	0x5100 0908 0x5180 0908	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Control 2 Register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>REGION_ENABLE</td><td>MATCH_MODE</td><td>INVERT_MODE</td><td>CFG_SHIFT_MODE</td><td>FUZZY_TYPE_MATCH_MODE</td><td>RESERVED</td><td colspan="2">RESPONSE_CODE</td><td colspan="2">RESERVED</td><td>MESSAGE_CODE_MATCH_ENABLE</td><td>VIRTUAL_FUNCTION_NUMBER_MATCH_ENABLE</td><td>FUNCTION_NUMBER_MATCH_ENABLE</td><td>AT_MATCH_ENABLE</td><td>RESERVED</td><td>ATTR_MATCH_ENABLE</td><td>TD_MATCH_ENABLE</td><td>TC_MATCH_ENABLE</td><td colspan="3">RESERVED</td><td colspan="3">BAR_NUMBER</td><td colspan="8">MESSAGECODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	REGION_ENABLE	Enable AT for this region	RW	0x0

Bits	Field Name	Description	Type	Reset
30	MATCH_MODE	Sets inbound TLP match mode, depending on TYPE 0x0: MEM,I/O: Address Match: as per region base & limit registers; CFG0: Routing ID Match: Completer ID (BDF) + reg address matches base & limit-defined region; MSG[D]: Address Match: as per region base & limit registers 0x1: MEM,I/O: BAR match: as defined in BAR_number field; CFG0: Accept mode: Completer ID (BDF) is ignored; MSG[D]: VendorID match: VendorID = upper_base[15:0] + VendorDefined = lower_base/limit	RW	0x0
29	INVERT_MODE	Redefine match criteria as outside the defined range (instead of inside)	RW	0x0
28	CFG_SHIFT_MODE	Enable the shifting of CFG CID (BDF), incoming and outgoing TLP; CFG get mapped to a contiguous 2**28 = 256 MByte address space Untranslated CID = CFG_DW#3[31:16] Shifted CID = CFG_DW#3[27:12]	RW	0x0
27	FUZZY_TYPE_MATCH_MODE	Outbound: DMA Bypass Mode Inbound: Relax matching on inbound TLP TYPE: CfgRd0 == CfgRd1 CfgWr0 == CfgWr1 MRd == MRdLk routing field of Msg/MsgD ignored	RW	0x0
26	RESERVED		R	0x0
25:24	RESPONSE_CODE	Override HW-generated completion status when responding inbound TLP 0x0: No override, use HW-generated CS 0x1: Unsupported Request: CS= 3'b001 0x2: Completer Abort: CS= 3'b100	RW	0x0
23:22	RESERVED		R	0x0
21	MESSAGE_CODE_MATCH_ENABLE	Enable MessageCode match criteria on inbound TLP	RW	0x0
20	VIRTUAL_FUNCTION_NUMBER_MATCH_ENABLE	VIRTUAL FUNCTIONS not IMPLEMENTED: not USED	RW	0x0
19	FUNCTION_NUMBER_MATCH_ENABLE	Outbound: Function Number Translation Bypass Inbound: Enable Function Number match criteria	RW	0x0
18	AT_MATCH_ENABLE	Enable AT match criteria on inbound TLP ATS not SUPPORTED: DO not USE	RW	0x0
17	RESERVED		R	0x0
16	ATTR_MATCH_ENABLE	Enable ATTR match criteria on inbound TLP	RW	0x0
15	TD_MATCH_ENABLE	Enable TD match criteria on inbound TLP	RW	0x0
14	TC_MATCH_ENABLE	Enable TC match criteria on inbound TLP	RW	0x0
13:11	RESERVED		R	0x0
10:8	BAR_NUMBER	BAR number for maybching with incoming MEM, I/O TLP (if Match_Mode = 1) 0x0: BAR0 0x1: BAR1 0x2: BAR2 0x3: BAR3 0x4: BAR4 0x5: BAR5 0x6: ROM	RW	0x0

Bits	Field Name	Description	Type	Reset
7:0	MESSAGECODE	Outbound: MessageCode applied to outgoing message TLP with matching addess Inbound: MessageCode-match criteria for infoming message TLP (if Message_Code_match_enable=1)	RW	0x0

Table 24-648. PCIECTRL_PL_IATU_REG_LOWER_BASE

Address Offset	0x0000 020C		
Physical Address	0x5100 090C 0x5180 090C	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Lower Base Address Register (2**12 = 4kbyte - aligned)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IATU_REG_LOWER_BASE																				ZERO											

Bits	Field Name	Description	Type	Reset
31:12	IATU_REG_LOWER_BASE		RW	0x0
11:0	ZERO		R	0x0

Table 24-649. PCIECTRL_PL_IATU_REG_UPPER_BASE

Address Offset	0x0000 0210		
Physical Address	0x5100 0910 0x5180 0910	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Upper Base Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IATU_REG_UPPER_BASE																															

Bits	Field Name	Description	Type	Reset
31:0	IATU_REG_UPPER_BASE		RW	0x0

Table 24-650. PCIECTRL_PL_IATU_REG_LIMIT

Address Offset	0x0000 0214		
Physical Address	0x5100 0914 0x5180 0914	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Limit Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IATU_REG_LIMIT																				ONES											

Bits	Field Name	Description	Type	Reset
31:12	IATU_REG_LIMIT		RW	0x0
11:0	ONES		R	0xfff

Table 24-651. PCIECTRL_PL_IATU_REG_LOWER_TARGET

Address Offset	0x0000 0218		
Physical Address	0x5100 0918 0x5180 0918	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Lower Target Address Register (2**12 = 4kbyte - aligned)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IATU_REG_LOWER_TARGET																				ZERO											

Bits	Field Name	Description	Type	Reset
31:12	IATU_REG_LOWER_TARGET		RW	0x0
11:0	ZERO		R	0x0

Table 24-652. PCIECTRL_PL_IATU_REG_UPPER_TARGET

Address Offset	0x0000 021C		
Physical Address	0x5100 091C 0x5180 091C	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Upper Target Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IATU_REG_UPPER_TARGET																															

Bits	Field Name	Description	Type	Reset
31:0	IATU_REG_UPPER_TARGET		RW	0x0

Table 24-653. PCIECTRL_PL_IATU_REG_CTRL_3

Address Offset	0x0000 0220		
Physical Address	0x5100 0920 0x5180 0920	**Instance**	PCIe_SS1_PL_CONF PCIe_SS2_PL_CONF
Description	iATU Region Control 3 Register; VIRTUAL FUNCTIONS not IMPLEMENTED: not USED		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IATU_REG_CTRL_3																															

Bits	Field Name	Description	Type	Reset
31:0	IATU_REG_CTRL_3		R	0x0

24.9.7.6 PCIe_SS_EP_CFG_DBICS2 Registers

Note

This section describes the PCIe EP mode (PCIe type-0) standard configuration registers as they are locally accessed within the DIF CS2 space. These register names are prefixed with "PCIECTRL_EP_DBICS2".

24.9.7.6.1 PCIe_SS_EP_CFG_DBICS2 Register Summary

Table 24-654. PCIe_SS1_EP_CFG_DBI_CS2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_EP_CFG _DBICS2 Physical Address
PCIECTRL_EP_DBICS2_DEVICE_VENDORID	RW	32	0x0	0x5100 1000
PCIECTRL_EP_DBICS2_STATUS_COMMAND_REGISTER	RW	32	0x4	0x5100 1004
PCIECTRL_EP_DBICS2_CLASSCODE_REVISIONID	RW	32	0x8	0x5100 1008
PCIECTRL_EP_DBICS2_BIST_HEAD_LAT_CACH	RW	32	0xC	0x5100 100C
PCIECTRL_EP_DBICS2_BAR0_MASK	RW	32	0x10	0x5100 1010
PCIECTRL_EP_DBICS2_BAR1_MASK	RW	32	0x14	0x5100 1014
PCIECTRL_EP_DBICS2_BAR2_MASK	RW	32	0x18	0x5100 1018
PCIECTRL_EP_DBICS2_BAR3_MASK	RW	32	0x1C	0x5100 101C
PCIECTRL_EP_DBICS2_BAR4_MASK	RW	32	0x20	0x5100 1020
PCIECTRL_EP_DBICS2_BAR5_MASK	RW	32	0x24	0x5100 1024
PCIECTRL_EP_DBICS2_CARDBUS_CIS_POINTER	RW	32	0x28	0x5100 1028
PCIECTRL_EP_DBICS2_SUBID_SUBVENDORID	RW	32	0x2C	0x5100 102C
PCIECTRL_EP_DBICS2_EXPANSION_ROM_BAR	RW	32	0x30	0x5100 1030
PCIECTRL_EP_DBICS2_CAPPTR	RW	32	0x34	0x5100 1034
PCIECTRL_EP_DBICS2_INTERRUPT	RW	32	0x3C	0x5100 103C
PCIECTRL_EP_DBICS2_PM_CAP	RW	32	0x40	0x5100 1040
PCIECTRL_EP_DBICS2_PM_CSR	RW	32	0x44	0x5100 1044
PCIECTRL_EP_DBICS2_PCIE_CAP	RW	32	0x70	0x5100 1070
PCIECTRL_EP_DBICS2_DEV_CAP	RW	32	0x74	0x5100 1074
PCIECTRL_EP_DBICS2_DEV_CAS	RW	32	0x78	0x5100 1078
PCIECTRL_EP_DBICS2_LNK_CAP	RW	32	0x7C	0x5100 107C
PCIECTRL_EP_DBICS2_LNK_CAS	RW	32	0x80	0x5100 1080
PCIECTRL_EP_DBICS2_DEV_CAP_2	R	32	0x94	0x5100 1094
PCIECTRL_EP_DBICS2_DEV_CAS_2	RW	32	0x98	0x5100 1098
PCIECTRL_EP_DBICS2_LNK_CAP_2	R	32	0x9C	0x5100 109C
PCIECTRL_EP_DBICS2_LNK_CAS_2	RW	32	0xA0	0x5100 10A0

Table 24-655. PCIe_SS2_EP_CFG_DBI_CS2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_EP_CFG _DBICS2 Physical Address
PCIECTRL_EP_DBICS2_DEVICE_VENDORID	RW	32	0x0	0x5180 1000
PCIECTRL_EP_DBICS2_STATUS_COMMAND_REGISTER	RW	32	0x4	0x5180 1004
PCIECTRL_EP_DBICS2_CLASSCODE_REVISIONID	RW	32	0x8	0x5180 1008
PCIECTRL_EP_DBICS2_BIST_HEAD_LAT_CACH	RW	32	0xC	0x5180 100C
PCIECTRL_EP_DBICS2_BAR0_MASK	RW	32	0x10	0x5180 1010
PCIECTRL_EP_DBICS2_BAR1_MASK	RW	32	0x14	0x5180 1014
PCIECTRL_EP_DBICS2_BAR2_MASK	RW	32	0x18	0x5180 1018

Table 24-655. PCIe_SS2_EP_CFG_DBI_CS2 Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_EP_CFG_DBICS2 Physical Address
PCIECTRL_EP_DBICS2_BAR3_MASK	RW	32	0x1C	0x5180 101C
PCIECTRL_EP_DBICS2_BAR4_MASK	RW	32	0x20	0x5180 1020
PCIECTRL_EP_DBICS2_BAR5_MASK	RW	32	0x24	0x5180 1024
PCIECTRL_EP_DBICS2_CARDBUS_CIS_POINTER	RW	32	0x28	0x5180 1028
PCIECTRL_EP_DBICS2_SUBID_SUBVENDORID	RW	32	0x2C	0x5180 102C
PCIECTRL_EP_DBICS2_EXPANSION_ROM_BAR	RW	32	0x30	0x5180 1030
PCIECTRL_EP_DBICS2_CAPPTR	RW	32	0x34	0x5180 1034
PCIECTRL_EP_DBICS2_INTERRUPT	RW	32	0x3C	0x5180 103C
PCIECTRL_EP_DBICS2_PM_CAP	RW	32	0x40	0x5180 1040
PCIECTRL_EP_DBICS2_PM_CSR	RW	32	0x44	0x5180 1044
PCIECTRL_EP_DBICS2_PCIE_CAP	RW	32	0x70	0x5180 1070
PCIECTRL_EP_DBICS2_DEV_CAP	RW	32	0x74	0x5180 1074
PCIECTRL_EP_DBICS2_DEV_CAS	RW	32	0x78	0x5180 1078
PCIECTRL_EP_DBICS2_LNK_CAP	RW	32	0x7C	0x5180 107C
PCIECTRL_EP_DBICS2_LNK_CAS	RW	32	0x80	0x5180 1080
PCIECTRL_EP_DBICS2_DEV_CAP_2	R	32	0x94	0x5180 1094
PCIECTRL_EP_DBICS2_DEV_CAS_2	RW	32	0x98	0x5180 1098
PCIECTRL_EP_DBICS2_LNK_CAP_2	R	32	0x9C	0x5180 109C
PCIECTRL_EP_DBICS2_LNK_CAS_2	RW	32	0xA0	0x5180 10A0

24.9.7.6.2 PCIe_SS_EP_CFG_DBICS2 Register Description

Table 24-656. PCIECTRL_EP_DBICS2_DEVICE_VENDORID

Address offset	0x0		
Physical Address	0x5100 1000 0x5180 1000	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Device and Vendor ID		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">DEVICEID</td><td colspan="16">VENDORID</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	DEVICEID	Device ID (CS)	RW	0x8888
15:0	VENDORID	Vendor ID (CS)	RW	0x104C

Table 24-657. PCIECTRL_EP_DBICS2_STATUS_COMMAND_REGISTER

Address offset	0x4		
Physical Address	0x5100 1004 0x5180 1004	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Status and Command registers		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>DETECT_PARERR</td><td>SIGNAL_SYSERR</td><td>RCVD_MASTERABORT</td><td>RCVD_TRGTABORT</td><td>SIGNAL_TRGTABORT</td><td colspan="2">DEVSEL_TIME</td><td>MASTERDATA_PARERR</td><td>FAST_B2B</td><td>RESERVED</td><td>C66MHZ_CAP</td><td>CAP_LIST</td><td>INTX_STATUS</td><td colspan="8">RESERVED</td><td>INTX_ASSER_DIS</td><td>FAST_B2BEN</td><td>SERR_EN</td><td>IDSEL_CTRL</td><td>PARITYERRRESP</td><td>VGA_SNOOP</td><td>MEMWR_INVA</td><td>SPEC_CYCLE_EN</td><td>BUSMASTER_EN</td><td>MEM_SPACE_EN</td><td>IO_SPACE_EN</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	DETECT_PARERR	Detected Parity Error	RW	0x0
30	SIGNAL_SYSERR	Signaled System Error	RW	0x0
29	RCVD_MASTERABORT	Received Master Abort	RW	0x0
28	RCVD_TRGTABORT	Received Target Abort	RW	0x0
27	SIGNAL_TRGTABORT	Signaled Target Abort	RW	0x0
26:25	DEVSEL_TIME	DevSel Timing, Harsdwired to 0 for PCIExpress	R	0x0
24	MASTERDATA_PARERR	Master Data Parity Error	RW	0x0
23	FAST_B2B	Back to Back Capable, Harsdwired to 0 for PCIExpress	R	0x0
22	RESERVED	Reserved	R	0x0
21	C66MHZ_CAP	66MHz Capable, Harsdwired to 0 for PCIExpress	R	0x0
20	CAP_LIST	Capabilities List Hardwired to 1	R	0x1
19	INTX_STATUS	INTx Status	R	0x0
18:11	RESERVED		R	0x0
10	INTX_ASSER_DIS	INTx Assertion Disable	RW	0x0

Bits	Field Name	Description	Type	Reset
9	FAST_BBEN	Bit hardwired to 0 for PCIExpress	R	0x0
8	SERR_EN	SERR Enable	RW	0x0
7	IDSEL_CTRL	Bit hardwired to 0 for PCIExpress	R	0x0
6	PARITYERRRESP	Parity Error Response	RW	0x0
5	VGA_SNOOP	Not Applicable forPCI Express Bit hardwired to 0 for PCIExpress	R	0x0
4	MEMWR_INVA	Not Applicable for PCI Express Bit hardwired to 0 for PCIExpress	R	0x0
3	SPEC_CYCLE_EN	Not Applicable for PCI Express Bit hardwired to 0 for PCIExpress	R	0x0
2	BUSMASTER_EN	Bus Master Enable	RW	0x0
1	MEM_SPACE_EN	Memory Space Enable	RW	0x0
0	IO_SPACE_EN	IO Space Enable	RW	0x0

Table 24-658. PCIECTRL_EP_DBICS2_CLASSCODE_REVISIONID

Address offset	0x8		
Physical Address	0x5100 1008 0x5180 1008	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Class code and Revision ID		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_CLS_CD								SUBCLS_CD								PROG_IF_CODE								REVID							

Bits	Field Name	Description	Type	Reset
31:24	BASE_CLS_CD	Base Class Code (CS)	RW	0x0
23:16	SUBCLS_CD	Sub Class Code (CS)	RW	0x0
15:8	PROG_IF_CODE	Programming Interface Code (CS)	RW	0x0
7:0	REVID	Revision ID (CS)	RW	0x1

Table 24-659. PCIECTRL_EP_DBICS2_BIST_HEAD_LAT_CACH

Address offset	0xC		
Physical Address	0x5100 100C 0x5180 100C	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	BIST, Header Type, Latency Timer, Cache Line Size		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIST								MFD	HEAD_TYP							MSTR_LAT_TIM								CACH_LN_SZE							

Bits	Field Name	Description	Type	Reset
31:24	BIST	BIST	R	0x0
23	MFD	MultiFunction Device	R	0x0
22:16	HEAD_TYP	Header Type 0x0 = EP header 0x1 = RC header	R	0x0
15:8	MSTR_LAT_TIM	Master Latency Timer, Not Applicable for PCIe hence hardwired to 0	R	0x0

Bits	Field Name	Description	Type	Reset
7:0	CACH_LN_SZE	Cache Line Size, No impact on write, write is allowed only for legacy purpose	RW	0x0

Table 24-660. PCIECTRL_EP_DBICS2_BAR0_MASK

Address offset	0x10		
Physical Address	0x5100 1010 0x5180 1010	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Base Address Register 0 Mask (CS2 mode only) Write ones to BAR[M-1:1] for a 2**M byte BAR		
Type	RW		

31:1	0
BAR_MASK	BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-661. PCIECTRL_EP_DBICS2_BAR1_MASK

Address offset	0x14		
Physical Address	0x5100 1014 0x5180 1014	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Base Address Register 1 (CS2 mode only) Write ones to BAR[M-1:1] for a 2**M byte BAR If BAR0 is in 64-bit mode, contains the upper bits of BAR0 mask.		
Type	RW		

31:1	0
BAR_MASK	BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-662. PCIECTRL_EP_DBICS2_BAR2_MASK

Address offset	0x18		
Physical Address	0x5100 1018 0x5180 1018	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Base Address Register 2 Mask (CS2 mode only) Write ones to BAR[M-1:1] for a 2**M byte BAR		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BAR_MASK	BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-663. PCIECTRL_EP_DBICS2_BAR3_MASK

Address offset	0x1C		
Physical Address	0x5100 101C 0x5180 101C	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Base Address Register 3 (CS2 mode only) Write ones to BAR[M-1:1] for a 2**M byte BAR If BAR2 is in 64-bit mode, contains the upper bits of BAR2 mask.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAR_MASK																															BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-664. PCIECTRL_EP_DBICS2_BAR4_MASK

Address offset	0x20		
Physical Address	0x5100 1020 0x5180 1020	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Base Address Register 4 Mask (CS2 mode only) Write ones to BAR[M-1:1] for a 2**M byte BAR		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAR_MASK																															BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-665. PCIECTRL_EP_DBICS2_BAR5_MASK

Address offset	0x24		
Physical Address	0x5100 1024 0x5180 1024	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2

Table 24-665. PCIECTRL_EP_DBICS2_BAR5_MASK (continued)

Description	Base Address Register 5 (CS2 mode only) Write ones to BAR[M-1:1] for a 2**M byte BAR If BAR4 is in 64-bit mode, contains the upper bits of BAR4 mask.
Type	RW

31–1	0
BAR_MASK	BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-666. PCIECTRL_EP_DBICS2_CARDBUS_CIS_POINTER

Address offset	0x28		
Physical Address	0x5100 1028 0x5180 1028	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description			
Type	RW		

31–0
CARDBUS_CIS_PTR_N

Bits	Field Name	Description	Type	Reset
31:0	CARDBUS_CIS_PTR_N	Cardbus CIS pointer (CS)	RW	0x0

Table 24-667. PCIECTRL_EP_DBICS2_SUBID_SUBVENDORID

Address offset	0x2C		
Physical Address	0x5100 102C 0x5180 102C	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description			
Type	RW		

31–16	15–0
SUBSYS_DEV_ID_N	SUBSYS_VENDOR_ID_N

Bits	Field Name	Description	Type	Reset
31:16	SUBSYS_DEV_ID_N	Subsystem ID (CS)	RW	0x1
15:0	SUBSYS_VENDOR_ID_N	Subsystem Vendor ID (CS)	RW	0x0

Table 24-668. PCIECTRL_EP_DBICS2_EXPANSION_ROM_BAR

Address offset	0x30		
Physical Address	0x5100 1030 0x5180 1030	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Expansion ROM Base Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EXROM_ADDRESS	EXROM_ADDRESS_RO	RESERVED	EXROM_EN

Bits	Field Name	Description	Type	Reset
31:16	EXROM_ADDRESS	Expansion ROM address, unmasked (ie programmable)	RW	0x0
15:11	EXROM_ADDRESS_RO	Expansion ROM address, masked.	R	0x0
10:1	RESERVED		R	0x0
0	EXROM_EN	Expansion ROM Enable	RW	0x0

Table 24-669. PCIECTRL_EP_DBICS2_CAPPTR

Address offset	0x34		
Physical Address	0x5100 1034 0x5180 1034	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	CapPtr		
Type	RW		

31:8	7:0
RESERVED	CAPTR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	CAPTR	First Capability Pointer (CS)	RW	0x40

Table 24-670. PCIECTRL_EP_DBICS2_INTERRUPT

Address offset	0x3C		
Physical Address	0x5100 103C 0x5180 103C	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Int Pin and line		
Type	RW		

31:16	15:8	7:0
RESERVED	INT_PIN	INT_LIN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	INT_PIN	Interrupt Pin (CS)	RW	0x1
7:0	INT_LIN	Interrupt Line	RW	0xFF

Table 24-671. PCIECTRL_EP_DBICS2_PM_CAP

Address Offset	0x40		
Physical Address	0x5100 1040 0x5180 1040	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Power Management Capability structure header		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

<table>
<tr><td colspan="5">PME_SP</td><td>D2_SP</td><td>D1_SP</td><td colspan="3">AUX_CUR</td><td>DSI</td><td>RESERVED</td><td>PME_CLK</td><td colspan="3">PMC_VER</td><td colspan="8">PM_NX_PTR</td><td colspan="8">CAP_ID</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:27	PME_SP	PME Support (CS); Power states from which PME messages can be sent (active hi, one bit per state) Bit 0: from D0 Bit 1: from D1 Bit 2: from D2 Bit 3: from D3hot Bit 4: from D3cold (if Vaux present)	RW	0x0B
26	D2_SP	D2 Support (CS)	RW	0
25	D1_SP	D1 Support (CS)	RW	1
24:22	AUX_CUR	AUX Current (CS)	RW	0x0
21	DSI	Device Specific Initialization (CS)	RW	0
20	RESERVED		R	0
19	PME_CLK	PME Clock, hardwired to 0 (CS)	RW	0
18:16	PMC_VER	Power Management specification version (CS)	RW	0x3
15:8	PM_NX_PTR	Next Capability Pointer (CS)	RW	0x50
7:0	CAP_ID	Capability ID Read 0x1: PM	R	0x01

Table 24-672. PCIECTRL_EP_DBICS2_PM_CSR

Address Offset	0x44		
Physical Address	0x5100 1044 0x5180 1044	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Power Management Control and Status Register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DATA1</td><td>BP_CCE</td><td>B2B3_SP</td><td colspan="6">RESERVED</td><td>PME_STATUS</td><td colspan="2">DATA_SCALE</td><td colspan="4">DATA_SEL</td><td>PME_EN</td><td colspan="4">RESERVED</td><td>NSR</td><td>RESERVED</td><td colspan="2">PWR_STATE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:24	DATA1	Data register for additional information (not supported)	R	0x00
23	BP_CCE	Bus Power/Clock Control Enable, hardwired to 0	R	0
22	B2B3_SP	B2/B3 Support, hardwired to 0	R	0
21:16	RESERVED		R	0x00
15	PME_STATUS	PME Status (Sticky bit)	RW W1toClr	0
14:13	DATA_SCALE	Data Scale (not supported)	R	0x0
12:9	DATA_SEL	Data Select (not supported)	R	0x0
8	PME_EN	PME Enable (Sticky bit) 0x0: Device not enabled to generate PME 0x1: Device enabled to generate PME; implies that Vaux is ON, ie sticky bits will be preserved over reset	RW	0
7:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3	NSR	No Soft Reset (CS)	RW	0
2	RESERVED		R	0
1:0	PWR_STATE	Device Power State 0x0: D0 state 0x1: D1 state 0x2: D2 state 0x3: D3 state	RW	0x0

Table 24-673. PCIECTRL_EP_DBICS2_PCIE_CAP

Address offset	0x70		
Physical Address	0x5100 1070 0x5180 1070	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	PCIE cap structure		
Type	RW		

31:30	29:25	24	23:20	19:16	15:8	7:0
RESERVED	IM_NUM	SLOT	DEV_TYPE	PCIE_VER	PCIE_NX_PTR	CAP_ID

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:25	IM_NUM	Interrupt Message Number (CS)	RW	0x0
24	SLOT	Slot Implemented Must be 0 for an endpoint	RW	0x0
23:20	DEV_TYPE	Device/Port Type Value depends on assigned type 0x0 = PCIe endpoint 0x1 = Legacy PCIe endpoint	R	0x0
19:16	PCIE_VER	PCI Express Capability Version	R	0x2
15:8	PCIE_NX_PTR	Next Capability Pointer (CS)	RW	0x0
7:0	CAP_ID	Capability ID	R	0x10

Table 24-674. PCIECTRL_EP_DBICS2_DEV_CAP

Address offset	0x74		
Physical Address	0x5100 1074 0x5180 1074	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	PCIE Device Capabilities		
Type	RW		

31:29	28	27:26	25:18	17:16	15	14:12	11:9	8:6	5	4:3	2:0
RESERVED	FLR_EN	CAPT_SLOW_PWRLIMIT_SCALE	CAPT_SLOW_PWRLIMIT_VALUE	RESERVED	ROLEBASED_ERRRPT	UNDEFINED	DEFAULT_EP_L1_ACCPT_LATENCY	DEFAULT_EP_L0S_ACCPT_LATENCY	EXTTAGFIELD_SUPPORT	PHANTOMFUNC	MAX_PAYLOAD_SIZE

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	FLR_EN	Function Level Reset Capability (CS)	RW	0x0
27:26	CAPT_SLOW_PWRLIMIT_SCALE	Captured Slow Power Scale Value (CS)	RW	0x0
25:18	CAPT_SLOW_PWRLIMIT_VALUE	Captured Slow Power Limit Value (CS)	RW	0x0
17:16	RESERVED		R	0x0
15	ROLEBASED_ERRRPT	Role Based Error Reporting (CS)	RW	0x1
14:12	UNDEFINED	Undefined from PCIe 1.1 onwards (CS)	R	0x0
11:9	DEFAULT_EP_L1_ACCPT_LATENCY	Endpoint L1 Acceptable Latency (CS)	R	0x3
8:6	DEFAULT_EP_L0S_ACCPT_LATENCY	Endpoint L0s Acceptable Latency (CS)	R	0x4
5	EXTTAGFIELD_SUPPORT	Value derived from DEFAULT_EXT_TAG_FIELD_SUPPORTED	RW	0x0
4:3	PHANTOMFUNC	Phantom Function Support, not SUPPORTED (CS)	RW	0x0
2:0	MAX_PAYLOAD_SIZE	Maximum Payload Size (CS) Read 0x1 = 256 Byte	RW	0x1

Table 24-675. PCIECTRL_EP_DBICS2_DEV_CAS

Address offset	0x78		
Physical Address	0x5100 1078 0x5180 1078	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	PCIE Device Control and Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										TRANS_PEND	AUXP_DET	UR_DET	FT_DET	NFT_DET	COR_DET	INIT_FLR	MRRS			NOSNP_EN	AUXPM_EN	PHFUN_EN	EXTAG_EN	MPS			EN_RO	UR_RE	FT_RE	NFT_RE	COR_RE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	TRANS_PEND	Transaction Pending	R	0x0
20	AUXP_DET	Aux Power Detected	R	0x0
19	UR_DET	Unsupported Request Detected	RW	0x0
18	FT_DET	Fatal Error Detected	RW	0x0
17	NFT_DET	Non-Fatal Error Detected	RW	0x0
16	COR_DET	Correctable Error Detected	RW	0x0
15	INIT_FLR	Reserved	R	0x0
14:12	MRRS	Max_Read_Request_Size	RW	0x2
11	NOSNP_EN	Enable No Snoop	RW	0x0
10	AUXPM_EN	AUX Power PM Enable	RW	0x0
9	PHFUN_EN	Phantom Function Enable	RW	0x0
8	EXTAG_EN	Extended Tag Field Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
7:5	MPS	Max_Payload_Size	RW	0x0
4	EN_RO	Enable Relaxed Ordering	RW	0x1
3	UR_RE	Unsupported Request Reporting Enable	RW	0x0
2	FT_RE	Fatal Error Reporting Enable	RW	0x0
1	NFT_RE	Non-Fatal Error Reporting Enable	RW	0x0
0	COR_RE	Correctable Error Reporting Enable	RW	0x0

Table 24-676. PCIECTRL_EP_DBICS2_LNK_CAP

Address offset	0x7C		
Physical Address	0x5100 107C 0x5180 107C	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	PCIE Link Capabilities		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORT_NUM								RESERVED	ASPM_OPT_COMP	LNK_BW_not_CAP	DLL_ACTRPT_CAP	UNSUP	CLK_PWR_MGMT	L1_EXIT_LAT			L0S_EXIT_LAT			AS_LINK_PM_SUPPORT		MAX_LINK_WIDTH						MAX_LINK_SPEEDS			

Bits	Field Name	Description	Type	Reset
31:24	PORT_NUM	Port Number (CS)	RW	0x0
23	RESERVED		R	0x0
22	ASPM_OPT_COMP	ASPM Optionality Compliance (CS)	RW	0x1
21	LNK_BW_not_CAP	Link Bandwidth Notification Capability (CS)	RW	0x0
20	DLL_ACTRPT_CAP	Data Link Layer Active Reporting Capable	R	0x0
19	UNSUP	Unsupported, Surprise Down Error Reporting Capable, Hardwired to 0	R	0x0
18	CLK_PWR_MGMT	Clock Power Management (CS)	RW	0x0
17:15	L1_EXIT_LAT	L1 Exit Latency (CS2)	RW	0x6
14:12	L0S_EXIT_LAT	L0s Exit Latency (CS2)	RW	0x3
11:10	AS_LINK_PM_SUPPORT	Active State Link PM (ASPM) Support (CS)	RW	0x3
9:4	MAX_LINK_WIDTH	Max Link Width (lanes) (CS)	RW	0x2
3:0	MAX_LINK_SPEEDS	Supported Max Link Speed (CS) 0x1(R) = 2.5 GT/s (Gen1) 0x2(R) = 5 GT/s (Gen2) 0x4(R) = 8 GT/s (Gen3)	RW	0x2

Table 24-677. PCIECTRL_EP_DBICS2_LNK_CAS

Address offset	0x80		
Physical Address	0x5100 1080 0x5180 1080	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	PCIE Link Control and Status		

Table 24-677. PCIECTRL_EP_DBICS2_LNK_CAS (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAB_STATUS	LBW_STATUS	DLL_ACT	SLOT_CLK_CONFIG	LINK_TRAIN	UNDEF	NEG_LW						LINK_SPEED				RESERVED				LABIE	LBMIE	HAWD	EN_CPM	EXT_SYN	COM_CLK_CFG	RETRAIN_LINK	LINK_DIS	RCB	RESERVED	ASPM_CTRL	

Bits	Field Name	Description	Type	Reset
31	LAB_STATUS	Link Autonomous Bandwidth Status	R	0x0
30	LBW_STATUS	Link Bandwidth Management Status	R	0x0
29	DLL_ACT	Data Link Layer Active	R	0x0
28	SLOT_CLK_CONFIG	Slot Clock Configuration (CS)	RW	0x1
27	LINK_TRAIN	LINK training	R	0x0
26	UNDEF	Undefined	R	0x0
25:20	NEG_LW	Negotiated Link Width UNDEFINED UNTIL LINK IS UP.	R	0x1
19:16	LINK_SPEED	Link Speed UNDEFINED UNTIL LINK IS UP.	R	0x1
15:12	RESERVED		R	0x0
11	LABIE	Link Autonomous Bandwidth Interrupt Enable.	RW	0x0
10	LBMIE	Link Bandwidth Management Interrupt Enable	RW	0x0
9	HAWD	Hardware Autonomous Width Disable	R	0x0
8	EN_CPM	Enable Clock Power Management	RW	0x0
7	EXT_SYN	Extended Synch	RW	0x0
6	COM_CLK_CFG	Common Clock Configuration	RW	0x0
5	RETRAIN_LINK	Retrain Link	R	0x0
4	LINK_DIS	Link Disable	R	0x0
3	RCB	Read Completion Boundary (CS) Read 0x0 = 64 Byte Read 0x1 = 128 Byte	RW	0x1
2	RESERVED		R	0x0
1:0	ASPM_CTRL	Active State Link PM Control 0x0: DISABLED 0x1: L0S_ENABLED 0x2: L1_ENABLED 0x3: L0S_AND_L1_ENABLED	RW	0x0

Table 24-678. PCIECTRL_EP_DBICS2_DEV_CAP_2

Address offset	0x94		
Physical Address	0x5100 1094 0x5180 1094	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Device Capabilities 2 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	TPHC_SP	RESERVED	NOROPR	CASC128_SP	AOC64_SP	AOC32_SP	AOR_SP	ARI_FWD_SP	CPL_TIMEOUT_DIS_SUPPORTED	CPL_TIMEOUT_RNG_SUPPORTED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	TPHC_SP	TPH Completer Supported	R	0x0
11	RESERVED		R	0x0
10	NOROPR	No RO-enabled PR-PR Passing	R	0x0
9	CASC128_SP	128-bit CAS Completer Supported	R	0x0
8	AOC64_SP	64-bit AtomicOp Completer Supported	R	0x0
7	AOC32_SP	32-bit AtomicOp Completer Supported	R	0x0
6	AOR_SP	AtomicOp Routing Supported	R	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	R	0x0
4	CPL_TIMEOUT_DIS_SUPPORTED	Completion Timeout Disable Supported	R	0x1
3:0	CPL_TIMEOUT_RNG_SUPPORTED	Completion Timeout Ranges Supported	R	0x1

Table 24-679. PCIECTRL_EP_DBICS2_DEV_CAS_2

Address offset	0x98		
Physical Address	0x5100 1098 0x5180 1098	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Device Control 2 Register		
Type	RW		

31:15	14:13	12:11	10	9	8	7	6	5	4	3:0
RESERVED	OBFF_EN	RESERVED	LTR_EN	IDO_CPL_EN	IDO_REQ_EN	AOP_EG_BLK	AOP_REQ_EN	ARI_FWD_SP	CPL_TIMEOUT_DIS	CPL_TIMEOUT_VALUE

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:13	OBFF_EN	OBFF Enable	RW	0x0
12:11	RESERVED		R	0x0
10	LTR_EN	LTR Mechanism Enable	RW	0x0
9	IDO_CPL_EN	IDO Completion Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
8	IDO_REQ_EN	IDO Request Enable	RW	0x0
7	AOP_EG_BLK	AtomicOp Egress Blocking	RW	0x0
6	AOP_REQ_EN	AtomicOp Requester Enable	RW	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	RW	0x0
4	CPL_TIMEOUT_DIS	Completion Timeout Disable	RW	0x0
3:0	CPL_TIMEOUT_VALUE	Completion Timeout Values	RW	0x0

Table 24-680. PCIECTRL_EP_DBICS2_LNK_CAP_2

Address offset	0x9C		
Physical Address	0x5100 109C 0x5180 109C	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	PCIE Link Capabilities 2 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CROSSLINK_SP	SP_LS_VEC							RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CROSSLINK_SP	Crosslink Supported	R	0x0
7:1	SP_LS_VEC	Supported Link Speeds Vector	R	0x3
0	RESERVED		R	0x0

Table 24-681. PCIECTRL_EP_DBICS2_LNK_CAS_2

Address offset	0xA0		
Physical Address	0x5100 10A0 0x5180 10A0	**Instance**	PCIe_SS1_EP_CFG_DBICS2 PCIe_SS2_EP_CFG_DBICS2
Description	Link Control and Status 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										LINK_EQ_REQ	EQ_PH3	EQ_PH2	EQ_PH1	EQ_COMPLETE	DEEMPH_LEVEL	COMPL_PRST_DEEPH				COMPL_SOS	ENT_MOD_COMPL	TX_MARGIN			SEL_DEEMP	HW_AUTO_SP_DIS	ENTR_COMPL	TRGT_LINK_SPEED			

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	LINK_EQ_REQ	Link Equilization Request	RW Wr1toClr	0x0

Bits	Field Name	Description	Type	Reset
20	EQ_PH3	Equalization Ph3 Success, Gen3 Only	R	0x0
19	EQ_PH2	Equalization Ph2 Success, Gen3 Only	R	0x0
18	EQ_PH1	Equalization Ph1 Success, Gen3 Only	R	0x0
17	EQ_COMPLETE	Equalization Complete, Gen3 Only	R	0x0
16	DEEMPH_LEVEL	Current De-emphasis Level	R	0x1
15:12	COMPL_PRST_DEEPH	Compliance Pre-set/ De-emphasis	RW	0x0
11	COMPL_SOS	Compliance SOS	RW	0x0
10	ENT_MOD_COMPL	Enter Modified Compliance	RW	0x0
9:7	TX_MARGIN	Transmit Margin	RW	0x0
6	SEL_DEEMP	Selectable De-emphasize	R	0x0
5	HW_AUTO_SP_DIS	Hardware Autonomous Speed Disable	R	0x0
4	ENTR_COMPL	Enter Compliance	RW	0x0
3:0	TRGT_LINK_SPEED	Target Link Speed	RW	0x1

24.9.7.7 PCIe_SS_RC_CFG_DBICS2 Registers

Note

This section describes the PCIe RC mode (PCIe type-1) standard configuration registers as they are locally accessed within the DIF CS2 space. These register names are prefixed with "PCIECTRL_RC_DBICS2".

24.9.7.7.1 PCIe_SS_RC_CFG_DBICS2 Register Summary

Table 24-682. PCIe_SS1_RC_CFG_DBI_CS2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_RC_CFG _DBICS2 Physical Address
PCIECTRL_RC_DBICS2_DEVICE_VENDORID	RW	32	0x0000 0000	0x5100 1000
PCIECTRL_RC_DBICS2_STATUS_COMMAND_REGISTER	RW	32	0x0000 0004	0x5100 1004
PCIECTRL_RC_DBICS2_CLASSCODE_REVISIONID	RW	32	0x0000 0008	0x5100 1008
PCIECTRL_RC_DBICS2_BIST_HEAD_LAT_CACH	RW	32	0x0000 000C	0x5100 100C
PCIECTRL_RC_DBICS2_BAR0_MASK	RW	32	0x0000 0010	0x5100 1010
PCIECTRL_RC_DBICS2_BAR1_MASK	RW	32	0x0000 0014	0x5100 1014
PCIECTRL_RC_DBICS2_BUS_NUM_REG	RW	32	0x0000 0018	0x5100 1018
PCIECTRL_RC_DBICS2_IOBASE_LIMIT_SEC_STATUS	RW	32	0x0000 001C	0x5100 101C
PCIECTRL_RC_DBICS2_MEM_BASE_LIMIT	RW	32	0x0000 0020	0x5100 1020
PCIECTRL_RC_DBICS2_PREF_MEM_BASE_LIMIT	RW	32	0x0000 0024	0x5100 1024
PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_BASEADDR	RW	32	0x0000 0028	0x5100 1028
PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_LIMITADDR	RW	32	0x0000 002C	0x5100 102C
PCIECTRL_RC_DBICS2_IO_BASE_LIMIT	RW	32	0x0000 0030	0x5100 1030
PCIECTRL_RC_DBICS2_CAPPTR	RW	32	0x0000 0034	0x5100 1034
PCIECTRL_RC_DBICS2_EXPANSION_ROM_BAR	RW	32	0x0000 0038	0x5100 1038
PCIECTRL_RC_DBICS2_BRIDGE_INT	RW	32	0x0000 003C	0x5100 103C
PCIECTRL_RC_DBICS2_PCIE_CAP	RW	32	0x0000 0070	0x5100 1070
PCIECTRL_RC_DBICS2_DEV_CAP	RW	32	0x0000 0074	0x5100 1074
PCIECTRL_RC_DBICS2_DEV_CAS	RW	32	0x0000 0078	0x5100 1078
PCIECTRL_RC_DBICS2_LNK_CAP	RW	32	0x0000 007C	0x5100 107C
PCIECTRL_RC_DBICS2_LNK_CAS	RW	32	0x0000 0080	0x5100 1080
PCIECTRL_RC_DBICS2_SLOT_CAP	RW	32	0x0000 0084	0x5100 1084
PCIECTRL_RC_DBICS2_SLOT_CAS	RW	32	0x0000 0088	0x5100 1088
PCIECTRL_RC_DBICS2_ROOT_CAC	RW	32	0x0000 008C	0x5100 108C
PCIECTRL_RC_DBICS2_ROOT_STS	RW	32	0x0000 0090	0x5100 1090
PCIECTRL_RC_DBICS2_DEV_CAP_2	R	32	0x0000 0094	0x5100 1094
PCIECTRL_RC_DBICS2_DEV_CAS_2	RW	32	0x0000 0098	0x5100 1098
PCIECTRL_RC_DBICS2_LNK_CAP_2	R	32	0x0000 009C	0x5100 109C
PCIECTRL_RC_DBICS2_LNK_CAS_2	RW	32	0x0000 00A0	0x5100 10A0

Table 24-683. PCIe_SS2_RC_CFG_DBI_CS2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_RC_CFG_DBICS2 Physical Address
PCIECTRL_RC_DBICS2_DEVICE_VENDORID	RW	32	0x0000 0000	0x5180 1000
PCIECTRL_RC_DBICS2_STATUS_COMMAND_REGISTER	RW	32	0x0000 0004	0x5180 1004
PCIECTRL_RC_DBICS2_CLASSCODE_REVISIONID	RW	32	0x0000 0008	0x5180 1008
PCIECTRL_RC_DBICS2_BIST_HEAD_LAT_CACH	RW	32	0x0000 000C	0x5180 100C

Table 24-683. PCIe_SS2_RC_CFG_DBI_CS2 Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_RC_CFG_DBICS2 Physical Address
PCIECTRL_RC_DBICS2_BAR0_MASK	RW	32	0x0000 0010	0x5180 1010
PCIECTRL_RC_DBICS2_BAR1_MASK	RW	32	0x0000 0014	0x5180 1014
PCIECTRL_RC_DBICS2_BUS_NUM_REG	RW	32	0x0000 0018	0x5180 1018
PCIECTRL_RC_DBICS2_IOBASE_LIMIT_SEC_STATUS	RW	32	0x0000 001C	0x5180 101C
PCIECTRL_RC_DBICS2_MEM_BASE_LIMIT	RW	32	0x0000 0020	0x5180 1020
PCIECTRL_RC_DBICS2_PREF_MEM_BASE_LIMIT	RW	32	0x0000 0024	0x5180 1024
PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_BASEADDR	RW	32	0x0000 0028	0x5180 1028
PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_LIMITADDR	RW	32	0x0000 002C	0x5180 102C
PCIECTRL_RC_DBICS2_IO_BASE_LIMIT	RW	32	0x0000 0030	0x5180 1030
PCIECTRL_RC_DBICS2_CAPPTR	RW	32	0x0000 0034	0x5180 1034
PCIECTRL_RC_DBICS2_EXPANSION_ROM_BAR	RW	32	0x0000 0038	0x5180 1038
PCIECTRL_RC_DBICS2_BRIDGE_INT	RW	32	0x0000 003C	0x5180 103C
PCIECTRL_RC_DBICS2_PCIE_CAP	RW	32	0x0000 0070	0x5180 1070
PCIECTRL_RC_DBICS2_DEV_CAP	RW	32	0x0000 0074	0x5180 1074
PCIECTRL_RC_DBICS2_DEV_CAS	RW	32	0x0000 0078	0x5180 1078
PCIECTRL_RC_DBICS2_LNK_CAP	RW	32	0x0000 007C	0x5180 107C
PCIECTRL_RC_DBICS2_LNK_CAS	RW	32	0x0000 0080	0x5180 1080
PCIECTRL_RC_DBICS2_SLOT_CAP	RW	32	0x0000 0084	0x5180 1084
PCIECTRL_RC_DBICS2_SLOT_CAS	RW	32	0x0000 0088	0x5180 1088
PCIECTRL_RC_DBICS2_ROOT_CAC	RW	32	0x0000 008C	0x5180 108C
PCIECTRL_RC_DBICS2_ROOT_STS	RW	32	0x0000 0090	0x5180 1090
PCIECTRL_RC_DBICS2_DEV_CAP_2	R	32	0x0000 0094	0x5180 1094
PCIECTRL_RC_DBICS2_DEV_CAS_2	RW	32	0x0000 0098	0x5180 1098
PCIECTRL_RC_DBICS2_LNK_CAP_2	R	32	0x0000 009C	0x5180 109C
PCIECTRL_RC_DBICS2_LNK_CAS_2	RW	32	0x0000 00A0	0x5180 10A0

24.9.7.7.2 PCIe_SS_RC_CFG_DBICS2 Register Description

Table 24-684. PCIECTRL_RC_DBICS2_DEVICE_VENDORID

Address Offset	0x0000 0000		
Physical Address	0x5100 1000 0x5180 1000	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Device and Vendor ID		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVICEID																VENDORID															

Bits	Field Name	Description	Type	Reset
31:16	DEVICEID	Device ID (CS)	RW	0x8888
15:0	VENDORID	Vendor ID (CS)	RW	0x104c

Table 24-685. PCIECTRL_RC_DBICS2_STATUS_COMMAND_REGISTER

Address Offset	0x0000 0004		
Physical Address	0x5100 1004 0x5180 1004	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Status and Command registers		

Table 24-685. PCIECTRL_RC_DBICS2_STATUS_COMMAND_REGISTER (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DETECT_PARERR	SIGNAL_SYSERR	RCVD_MASTERABORT	RCVD_TRGTABORT	SIGNAL_TRGTABORT	DEVSEL_TIME		MASTERDATA_PARERR	FAST_B2B	RESERVED	C66MHZ_CAP	CAP_LIST	INTX_STATUS	RESERVED								INTX_ASSER_DIS	FAST_BBEN	SERR_EN	IDSEL_CTRL	PARITYERRRESP	VGA_SNOOP	MEMWR_INVA	SPEC_CYCLE_EN	BUSMASTER_EN	MEM_SPACE_EN	IO_SPACE_EN

Bits	Field Name	Description	Type	Reset
31	DETECT_PARERR	Detected Parity Error	RW	0x0
30	SIGNAL_SYSERR	Signaled System Error	RW	0x0
29	RCVD_MASTERABORT	Received Master Abort	RW	0x0
28	RCVD_TRGTABORT	Received Target Abort	RW	0x0
27	SIGNAL_TRGTABORT	Signaled Target Abort	RW	0x0
26:25	DEVSEL_TIME	DevSel Timing, Harsdwired to 0 for PCIExpress	R	0x0
24	MASTERDATA_PARERR	Master Data Parity Error	RW	0x0
23	FAST_B2B	Back to Back Capable, Harsdwired to 0 for PCIExpress	R	0x0
22	RESERVED	Reserved	R	0x0
21	C66MHZ_CAP	66MHz Capable, Harsdwired to 0 for PCIExpress	R	0x0
20	CAP_LIST	Capabilities List Hardwired to 1	R	0x1
19	INTX_STATUS	INTx Status	R	0x0
18:11	RESERVED		R	0x0
10	INTX_ASSER_DIS	INTx Assertion Disable	RW	0x0
9	FAST_BBEN	Bit hardwired to 0 for PCIExpress	R	0x0
8	SERR_EN	SERR Enable	RW	0x0
7	IDSEL_CTRL	Bit hardwired to 0 for PCIExpress	R	0x0
6	PARITYERRRESP	Parity Error Response	RW	0x0
5	VGA_SNOOP	Not Applicable forPCI Express; Bit hardwired to 0 for PCIExpress	R	0x0
4	MEMWR_INVA	Not Applicable for PCI Express; Bit hardwired to 0 for PCIExpress	R	0x0
3	SPEC_CYCLE_EN	Not Applicable for PCI Express; Bit hardwired to 0 for PCIExpress	R	0x0
2	BUSMASTER_EN	Bus Master Enable (BME)	RW	0x0
1	MEM_SPACE_EN	Memory Space Enable (MSE)	RW	0x0
0	IO_SPACE_EN	IO Space Enable (ISE)	RW	0x0

Table 24-686. PCIECTRL_RC_DBICS2_CLASSCODE_REVISIONID

Address Offset	0x0000 0008		
Physical Address	0x5100 1008 0x5180 1008	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Class code and Revision ID		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:24	23:16	15:8	7:0
BASE_CLS_CD	SUBCLS_CD	PROG_IF_CODE	REVID

Bits	Field Name	Description	Type	Reset
31:24	BASE_CLS_CD	Base Class Code (CS)	RW	0x0
23:16	SUBCLS_CD	Sub Class Code (CS)	RW	0x0
15:8	PROG_IF_CODE	Programming Interface Code (CS)	RW	0x0
7:0	REVID	Revision ID (CS)	RW	0x1

Table 24-687. PCIECTRL_RC_DBICS2_BIST_HEAD_LAT_CACH

Address Offset	0x0000 000C		
Physical Address	0x5100 100C 0x5180 100C	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	BIST, Header Type, Latency Timer, Cache Line Size		
Type	RW		

31:24	23	22:16	15:8	7:0
BIST	MFD	HEAD_TYP	MSTR_LAT_TIM	CACH_LN_SZE

Bits	Field Name	Description	Type	Reset
31:24	BIST	BIST	R	0x0
23	MFD	MultiFunction Device	R	0x0
22:16	HEAD_TYP	Header Type Read 0x0: EP header (Type0) Read 0x1: RC header (Type1)	R	0x1
15:8	MSTR_LAT_TIM	Master Latency Timer, Not Applicable for PCIe hence hardwired to 0	R	0x0
7:0	CACH_LN_SZE	Cache Line Size, No impact on write, write is allowed only for legacy purpose	RW	0x0

Table 24-688. PCIECTRL_RC_DBICS2_BAR0_MASK

Address Offset	0x0000 0010		
Physical Address	0x5100 1010 0x5180 1010	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Base Address Register 0 Mask (CS2 mode only) Write 1 to BAR[0] to enable the BAR Write ones to BAR[M-1:1] for a 2**M byte BAR Reads like in CS mode		
Type	RW		

31:1	0
BAR_MASK	BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-689. PCIECTRL_RC_DBICS2_BAR1_MASK

Address Offset	0x0000 0014

Table 24-689. PCIECTRL_RC_DBICS2_BAR1_MASK (continued)

Physical Address	0x5100 1014 0x5180 1014	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Base Address Register 1 Mask (CS2 mode only) Write 1 to BAR[0] to enable the BAR Write ones to BAR[M-1:1] for a 2**M byte BAR If BAR0 is in 64-bit mode, contains the upper bits of BAR0 mask Reads like in CS mode		
Type	RW		

31–1	0
BAR_MASK	BAR_ENABLED

Bits	Field Name	Description	Type	Reset
31:1	BAR_MASK	Write 1 to unmask/0 to mask the BAR address bit (CS2 only)	RW	0xFFFF
0	BAR_ENABLED	BAR enabled (CS2 only)	RW	0x1

Table 24-690. PCIECTRL_RC_DBICS2_BUS_NUM_REG

Address Offset	0x0000 0018		
Physical Address	0x5100 1018 0x5180 1018	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Bus Number Registers		
Type	RW		

31–24	23–16	15–8	7–0
SEC_LAT_TIMER	SUBORD_BUS_NUM	SEC_BUS_NUM	PRIM_BUS_NUM

Bits	Field Name	Description	Type	Reset
31:24	SEC_LAT_TIMER	Secondary Latency Timer, Not Applicable for PCI Express hence hardwired to 0	R	0x0
23:16	SUBORD_BUS_NUM	Subordinate Bus Number	RW	0x0
15:8	SEC_BUS_NUM	Secondary Bus Number	RW	0x0
7:0	PRIM_BUS_NUM	Primary Bus Number	RW	0x0

Table 24-691. PCIECTRL_RC_DBICS2_IOBASE_LIMIT_SEC_STATUS

Address Offset	0x0000 001C		
Physical Address	0x5100 101C 0x5180 101C	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	IO Base,Limit and Secondary Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DET_PAR_ERR	RCVD_SYS_ERR	RCVD_MSTR_ABORT	RCVD_TRGT_ABORT	SGNLD_TRGT_ABORT	DEVSEL_TIMING	MSTR_DATA_PRTY_ERR	FAST_B2B_CAP	RESERVED	C66MHZ_CAPA	RESERVED	IO_SPACE_LIMIT	RESERVED	IODECODE_32	IO_SPACE_BASE	RESERVED	IODECODE_32_0

Bits	Field Name	Description	Type	Reset
31	DET_PAR_ERR	Detected Parity Error	RW	0x0
30	RCVD_SYS_ERR	Received System Error	RW	0x0
29	RCVD_MSTR_ABORT	Received Master Abort	RW	0x0
28	RCVD_TRGT_ABORT	Received Target Error	RW	0x0
27	SGNLD_TRGT_ABORT	Signaled Target Error	RW	0x0
26:25	DEVSEL_TIMING	DEVSEL Timing, Not Applicable for PCI Express hence hardwired to 0	R	0x0
24	MSTR_DATA_PRTY_ERR	Mastered Data Parity Error	RW	0x0
23	FAST_B2B_CAP	Fast Back to Back Capable, Not Applicable for PCI Express hence hardwired to 0	R	0x0
22	RESERVED		R	0x0
21	C66MHZ_CAPA	66MHz Capable, Not Applicable for PCI Express hence hardwired to 0	R	0x0
20:16	RESERVED		R	0x0
15:12	IO_SPACE_LIMIT	IO_Space_Limit	RW	0x0
11:9	RESERVED		R	0x0
8	IODECODE_32	32 or 16 Bit IO Space	R	0x0
7:4	IO_SPACE_BASE	IO_Space_Limit	RW	0x0
3:1	RESERVED		R	0x0
0	IODECODE_32_0	32 or 16 Bit IO Space (CS)	R	0x0

Table 24-692. PCIECTRL_RC_DBICS2_MEM_BASE_LIMIT

Address Offset	0x0000 0020		
Physical Address	0x5100 1020 0x5180 1020	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Memory Base and Limit Register		
Type	RW		

31:20	19:16	15:4	3:0
MEM_LIMIT_ADDR	RESERVED	MEM_BASE_ADDR	RESERVED

Bits	Field Name	Description	Type	Reset
31:20	MEM_LIMIT_ADDR	Memory Limit Address	RW	0x0
19:16	RESERVED		R	0x0
15:4	MEM_BASE_ADDR	Memory Base Address	RW	0x0
3:0	RESERVED		R	0x0

Table 24-693. PCIECTRL_RC_DBICS2_PREF_MEM_BASE_LIMIT

Address Offset	0x0000 0024		
Physical Address	0x5100 1024 0x5180 1024	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2

Table 24-693. PCIECTRL_RC_DBICS2_PREF_MEM_BASE_LIMIT (continued)

Description	Prefetchable Memory Base and Limit Register
Type	RW

31:20	19:17	16	15:4	3:1	0
PREF_MEM_ADDR	RESERVED	MEMDECODE_64	UPPPREF_MEM_ADDR	RESERVED	MEMDECODE_64_0

Bits	Field Name	Description	Type	Reset
31:20	PREF_MEM_ADDR	Upper 12 bits of 32-bit Prefetchable Memory End Address	RW	0x0
19:17	RESERVED		R	0x0
16	MEMDECODE_64	64-Bit Memory Addressing	R	0x0
15:4	UPPPREF_MEM_ADDR	Upper 12 bits of 32-bit Prefetchable Memory start Address	RW	0x0
3:1	RESERVED		R	0x0
0	MEMDECODE_64_0	64-Bit Memory Addressing	R	0x0

Table 24-694. PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_BASEADDR

Address Offset	0x0000 0028		
Physical Address	0x5100 1028 0x5180 1028	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Upper 32 Bit Prefetachable Base Address Register		
Type	RW		

31:0
ADDRUPP

Bits	Field Name	Description	Type	Reset
31:0	ADDRUPP	Upper 32 Bits of Base Address of Prefetachable Memory Space, Used only if 64 Bit Prefetachable Addresing is enabled	RW	0x0

Table 24-695. PCIECTRL_RC_DBICS2_UPPER_32BIT_PREF_LIMITADDR

Address Offset	0x0000 002C		
Physical Address	0x5100 102C 0x5180 102C	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Upper 32 Bit Prefetachable Limit Address Register		
Type	RW		

31:0
ADDRUPP_LIMIT

Bits	Field Name	Description	Type	Reset
31:0	ADDRUPP_LIMIT	Upper 32 Bits of Limit Address of Prefetachable Memory Space, Used only if 64 Bit Prefetachable Addresing is enabled	RW	0x0

Table 24-696. PCIECTRL_RC_DBICS2_IO_BASE_LIMIT

Address Offset	0x0000 0030		
Physical Address	0x5100 1030 0x5180 1030	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	IO Base and Limit Register		
Type	RW		

31:16	15:0
UPP16_IOLIMIT	UPP16_IOBASE

Bits	Field Name	Description	Type	Reset
31:16	UPP16_IOLIMIT	Upper 16 IO Limit Address	RW	0x0
15:0	UPP16_IOBASE	Upper 16 IO Base Address	RW	0x0

Table 24-697. PCIECTRL_RC_DBICS2_CAPPTR

Address Offset	0x0000 0034		
Physical Address	0x5100 1034 0x5180 1034	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	CapPtr		
Type	RW		

31:8	7:0
RESERVED	CAPTR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	CAPTR	First Capability Pointer (CS)	RW	0x40

Table 24-698. PCIECTRL_RC_DBICS2_EXPANSION_ROM_BAR

Address Offset	0x0000 0038		
Physical Address	0x5100 1038 0x5180 1038	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Expansion ROM Base Address Register		
Type	RW		

31:16	15:11	10:1	0
EXROM_ADDRESS	EXROM_ADDRESS_RO	RESERVED	EXP_ROM_EN

Bits	Field Name	Description	Type	Reset
31:16	EXROM_ADDRESS	Expansion ROM address, unmasked (ie programmable)	RW	0x0
15:11	EXROM_ADDRESS_RO	Expansion ROM address, masked.	R	0x0
10:1	RESERVED		R	0x0
0	EXP_ROM_EN	Expansion ROM Enable	RW	0x0

Table 24-699. PCIECTRL_RC_DBICS2_BRIDGE_INT

Address Offset	0x0000 003C		
Physical Address	0x5100 103C 0x5180 103C	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Bridge Control and Int Pin and line		

Table 24-699. PCIECTRL_RC_DBICS2_BRIDGE_INT (continued)

Type RW

31 30 29 28	27	26	25	24	23	22	21	20	19	18	17	16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DT_SERR_EN	DT_STS	SEC_DT	PRI_DT	FAST_B2B_EN	SEC_BUS_RST	MST_ABT_MOD	VGA_16B_DEC	VGA_EN	ISA_EN	SERR_EN	PERR_RESP_EN	INT_PIN	INT_LIN

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	DT_SERR_EN	Discard Timer SERR Enable Status	R	0x0
26	DT_STS	Discard Timer Status	R	0x0
25	SEC_DT	Secondary Discard Timer	R	0x0
24	PRI_DT	Primary Discard Timer	R	0x0
23	FAST_B2B_EN	Fast Back-to-Back Transactions Enable	R	0x0
22	SEC_BUS_RST	Secondary Bus Reset (initiate hot reset)	RW	0x0
21	MST_ABT_MOD	Master Abort Mode	R	0x0
20	VGA_16B_DEC	VGA 16-Bit Decode	RW	0x0
19	VGA_EN	VGA Enable	RW	0x0
18	ISA_EN	ISA Enable	RW	0x0
17	SERR_EN	SERR Enable	RW	0x0
16	PERR_RESP_EN	Parity Error Response Enable	RW	0x0
15:8	INT_PIN	Interrupt Pin (CS)	R	0x1
7:0	INT_LIN	Interrupt Line	RW	0xff

Table 24-700. PCIECTRL_RC_DBICS2_PCIE_CAP

Address Offset	0x0000 0070		
Physical Address	0x5100 1070 0x5180 1070	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	PCI Express Capability structure header		
Type	RW		

31 30	29 28 27 26 25	24	23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	IM_NUM	SLOT	DEV_TYPE	PCIE_VER	PCIE_NX_PTR	CAP_ID

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:25	IM_NUM	Interrupt Message Number (CS)	RW	0x0
24	SLOT	Slot Implemented (CS)	RW	0x0
23:20	DEV_TYPE	Device/Port Type Read 0x4: RC root port (RC)	R	0x4
19:16	PCIE_VER	PCI Express Capability Version	R	0x2
15:8	PCIE_NX_PTR	Next Capability Pointer (CS)	RW	0x0
7:0	CAP_ID	Capability ID Read 0x10: PCIE	R	0x10

Table 24-701. PCIECTRL_RC_DBICS2_DEV_CAP

Address Offset	0x0000 0074		
Physical Address	0x5100 1074 0x5180 1074	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	PCIE Device Capabilities		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">CAPT_SLOW_PWRLIMIT_SCALE</td><td colspan="8">CAPT_SLOW_PWRLIMIT_VALUE</td><td colspan="2">RESERVED</td><td>ROLEBASED_ERRRPT</td><td colspan="3">UNDEFINED</td><td colspan="3">DEFAULT_EP_L1_ACCPT_LATENCY</td><td colspan="3">DEFAULT_EP_L0S_ACCPT_LATENCY</td><td>EXTTAGFIELD_SUPPORT</td><td colspan="2">PHANTOMFUNC</td><td colspan="3">MAX_PAYLOAD_SIZE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CAPT_SLOW_PWRLIMIT_SCALE	Captured Slow Power Scale Value, for Upstream Port Only (CS)	RW	0x0
25:18	CAPT_SLOW_PWRLIMIT_VALUE	Captured Slow Power Limit Value, for Upstream Port Only (CS)	RW	0x0
17:16	RESERVED		R	0x0
15	ROLEBASED_ERRRPT	Role Based Error Reporting (CS)	RW	0x1
14:12	UNDEFINED	Undefined from PCIe 1.1 onwards	R	0x0
11:9	DEFAULT_EP_L1_ACCPT_LATENCY	Endpoint L1 Acceptable Latency; Must be 0 for RC.	R	0x0
8:6	DEFAULT_EP_L0S_ACCPT_LATENCY	Endpoint L0s Acceptable Latency; Must be 0 for RC.	R	0x0
5	EXTTAGFIELD_SUPPORT	Extended Tag Field Support (CS)	RW	0x0
4:3	PHANTOMFUNC	Phantom Function Support, not SUPPORTED (CS)	RW	0x0
2:0	MAX_PAYLOAD_SIZE	Maximum Payload Size (CS) Read 0x1: 256 Byte	RW	0x1

Table 24-702. PCIECTRL_RC_DBICS2_DEV_CAS

Address Offset	0x0000 0078		
Physical Address	0x5100 1078 0x5180 1078	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	PCIE Device Control and Status		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td>TRANS_PEND</td><td>AUXP_DET</td><td>UR_DET</td><td>FT_DET</td><td>NFT_DET</td><td>COR_DET</td><td>INIT_FLR</td><td colspan="3">MRRS</td><td>NOSNP_EN</td><td>AUXPM_EN</td><td>PHFUN_EN</td><td>EXTAG_EN</td><td colspan="3">MPS</td><td>EN_RO</td><td>UR_RE</td><td>FT_RE</td><td>NFT_RE</td><td>COR_RE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	TRANS_PEND	Transaction Pending	R	0x0

Bits	Field Name	Description	Type	Reset
20	AUXP_DET	Aux Power Detected	R	0x0
19	UR_DET	Unsupported Request Detected	RW	0x0
18	FT_DET	Fatal Error Detected	RW	0x0
17	NFT_DET	Non-Fatal Error Detected	RW	0x0
16	COR_DET	Correctable Error Detected	RW	0x0
15	INIT_FLR	Reserved	R	0x0
14:12	MRRS	Max_Read_Request_Size	RW	0x2
11	NOSNP_EN	Enable No Snoop	RW	0x1
10	AUXPM_EN	AUX Power PM Enable (Sticky bit) 0x0: Vaux not used by device 0x1: Device can draw Vaux power; Sticky bits will be preserved over reset	RW	0x0
9	PHFUN_EN	Phantom Function Enable	RW	0x0
8	EXTAG_EN	Extended Tag Field Enable	RW	0x0
7:5	MPS	Max_Payload_Size	RW	0x0
4	EN_RO	Enable Relaxed Ordering	RW	0x1
3	UR_RE	Unsupported Request Reporting Enable	RW	0x0
2	FT_RE	Fatal Error Reporting Enable	RW	0x0
1	NFT_RE	Non-Fatal Error Reporting Enable	RW	0x0
0	COR_RE	Correctable Error Reporting Enable	RW	0x0

Table 24-703. PCIECTRL_RC_DBICS2_LNK_CAP

Address Offset	0x0000 007C		
Physical Address	0x5100 107C 0x5180 107C	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	PCIE Link Capabilities		
Type	RW		

31-24	23	22	21	20	19	18	17-15	14-12	11-10	9-4	3-0
PORT_NUM	RESERVED	ASPM_OPT_COMP	LNK_BW_not_CAP	DLL_ACTRPT_CAP	UNSUP	CLK_PWR_MGMT	COMM_L1_EXIT_LAT	COMM_L0S_EXIT_LAT	AS_LINK_PM_SUPPORT	MAX_LINK_WIDTH	MAX_LINK_SPEEDS

Bits	Field Name	Description	Type	Reset
31:24	PORT_NUM	Port Number (CS)	RW	0x0
23	RESERVED		R	0x0
22	ASPM_OPT_COMP	ASPM Optionality Compliance (CS)	RW	0x1
21	LNK_BW_not_CAP	Link Bandwidth Notification Capability (CS)	RW	0x1
20	DLL_ACTRPT_CAP	Data Link Layer Active Reporting Capable	R	0x1
19	UNSUP	Unsupported, Surprise Down Error Reporting Capable, Hardwired to 0	R	0x0
18	CLK_PWR_MGMT	Clock Power Management; Hardwired to 0 for DS port (RC); (CS)	RW	0x0
17:15	COMM_L1_EXIT_LAT	Common-clock-mode L1 Exit Latency (CS2) Compare CS	RW	0x6

Bits	Field Name	Description	Type	Reset
14:12	COMM_L0S_EXIT_LAT	Common-clock-mode L0s Exit Latency (CS2) Compare CS	RW	0x3
11:10	AS_LINK_PM_SUPPORT	Active State Link PM (ASPM) Support (CS)	RW	0x3
9:4	MAX_LINK_WIDTH	Max Link Width (lanes) (CS)	RW	0x2
3:0	MAX_LINK_SPEEDS	Supported Max Link Speed (CS) Read 0x1: 2.5 GT/s (Gen1) Read 0x2: 5 GT/s (Gen2) Read 0x3: 8 GT/s (Gen3)	RW	0x2

Table 24-704. PCIECTRL_RC_DBICS2_LNK_CAS

Address Offset	0x0000 0080		
Physical Address	0x5100 1080 0x5180 1080	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	PCIE Link Control and Status		
Type	RW Wr1toClr		

31	30	29	28	27	26	25:20	19:16	15:12	11	10	9	8	7	6	5	4	3	2	1:0
LAB_STATUS	LBW_STATUS	DLL_ACT	SLOT_CLK_CONFIG	LINK_TRAIN	UNDEF	NEG_LW	LINK_SPEED	RESERVED	LABIE	LBMIE	HAWD	EN_CPM	EXT_SYN	COM_CLK_CFG	RETRAIN_LINK	LINK_DIS	RCB	RESERVED	ASPM_CTRL

Bits	Field Name	Description	Type	Reset
31	LAB_STATUS	Link Autonomous Bandwidth Status	RW Wr1toClr	0x0
30	LBW_STATUS	Link Bandwidth Management Status	RW Wr1toClr	0x0
29	DLL_ACT	Data Link Layer Active	R	0x0
28	SLOT_CLK_CONFIG	Slot Clock Configuration (CS)	RW	0x1
27	LINK_TRAIN	LINK training	R	0x0
26	UNDEF	Undefined	R	0x0
25:20	NEG_LW	Negotiated Link Width; UNDEFINED UNTIL LINK IS UP.	R	0x1
19:16	LINK_SPEED	Link Speed; UNDEFINED UNTIL LINK IS UP.	R	0x1
15:12	RESERVED		R	0x0
11	LABIE	Link Autonomous Bandwidth Interrupt Enable	RW	0x0
10	LBMIE	Link Bandwidth Management Interrupt Enable	RW	0x0
9	HAWD	Hardware Autonomous Width Disable	R	0x0
8	EN_CPM	Enable Clock Power Management	RW	0x0
7	EXT_SYN	Extended Synch	RW	0x0
6	COM_CLK_CFG	Common Clock Configuration 0x0: Asynchronous reference clocks 0x1: Distributed common reference clock	RW	0x0
5	RETRAIN_LINK	Retrain Link	RW	0x0
4	LINK_DIS	Link Disable	RW	0x0
3	RCB	Read Completion Boundary (CS) 0x0: 64 Byte 0x1: 128 Byte	RW	0x1

Bits	Field Name	Description	Type	Reset
2	RESERVED		R	0x0
1:0	ASPM_CTRL	Active State Link PM Control 0x0: DISABLED 0x1: L0S_ENABLED 0x2: L1_ENABLED 0x3: L0S_AND_L1_ENABLED	RW	0x0

Table 24-705. PCIECTRL_RC_DBICS2_SLOT_CAP

Address Offset	0x0000 0084		
Physical Address	0x5100 1084 0x5180 1084	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Slot Capabilities Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSN													NCCS	EIP	SPLS		SPLV								HPC	HPS	PIP	AIP	MRLSP	PCP	ABP

Bits	Field Name	Description	Type	Reset
31:19	PSN	Physical Slot Number (CS)	RW	0x0
18	NCCS	No Command Complete Support (CS)	RW	0x0
17	EIP	Electromechanical Interlock Present (CS)	RW	0x0
16:15	SPLS	Slot Power Limit Scale (CS)	RW	0x0
14:7	SPLV	Slot Power Limit Value (CS)	RW	0x0
6	HPC	Hot-Plug Capable (CS)	RW	0x0
5	HPS	Hot-Plug Surprise (CS)	RW	0x0
4	PIP	Power Indicator Present (CS)	RW	0x0
3	AIP	Attention Indicator Present (CS)	RW	0x0
2	MRLSP	MRL Sensor Present (CS)	RW	0x0
1	PCP	Power Controller Present (CS)	RW	0x0
0	ABP	Attention Button Present (CS)	RW	0x0

Table 24-706. PCIECTRL_RC_DBICS2_SLOT_CAS

Address Offset	0x0000 0088		
Physical Address	0x5100 1088 0x5180 1088	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Slot Control and Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DSC	EIS	PDS	MRLSS	CC	PDC	MRCSC	PFD	ABP	RESERVED			DSC_EN	EIC	PCC	PIC		AIC		HPI_EN	CCI_EN	PDC_EN	MRLSC_EN	PFD_EN	ABP_EN

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	DSC	Data Link Layer State Changed	RW	0x0
23	EIS	Electromechanical Interlock Status	R	0x0

Bits	Field Name	Description	Type	Reset
22	PDS	Presence Detect State NO PRESENCE DETECTION IMPLEMENTED: TIED TO 1	R	0x1
21	MRLSS	MRL Sensor State	R	0x0
20	CC	Command Completed	RW	0x0
19	PDC	Presence Detect Changed	RW	0x0
18	MRCSC	MRL Sensor Changed	RW	0x0
17	PFD	Power Fault Detected	RW	0x0
16	ABP	Attention Button Pressed	RW	0x0
15:13	RESERVED		R	0x0
12	DSC_EN	Data Link Layer State Changed Enable	RW	0x0
11	EIC	Electromechanical Interlock Control	RW	0x0
10	PCC	Power Controller Control	RW	0x0
9:8	PIC	Power Indicator Control	RW	0x3
7:6	AIC	Attention Indicator Control	RW	0x3
5	HPI_EN	Hot-Plug Interrupt Enable	RW	0x0
4	CCI_EN	Command Completed Interrupt Enable	RW	0x0
3	PDC_EN	Presence Detect Changed Enable	RW	0x0
2	MRLSC_EN	MRL Sensor Changed Enable	RW	0x0
1	PFD_EN	Power Fault Detected Enable	RW	0x0
0	ABP_EN	Attention Button Pressed Enable	RW	0x0

Table 24-707. PCIECTRL_RC_DBICS2_ROOT_CAC

Address Offset	0x0000 008C		
Physical Address	0x5100 108C 0x5180 108C	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Root Control and Capability Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															CRSSV	RESERVED											CRSSV_EN	PMEI_EN	SEFE_EN	SENE_EN	SECE_EN

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CRSSV	CRS Software Visibility	R	0x0
15:5	RESERVED		R	0x0
4	CRSSV_EN	CRS Software Visibility Enable	R	0x0
3	PMEI_EN	PME Interrupt Enable	RW	0x0
2	SEFE_EN	System Error on Fatal Error Enable	RW	0x0
1	SENE_EN	System Error on Non-fatal Error Enable	RW	0x0
0	SECE_EN	System Error on Correctable Error Enable	RW	0x0

Table 24-708. PCIECTRL_RC_DBICS2_ROOT_STS

Address Offset	0x0000 0090		
Physical Address	0x5100 1090 0x5180 1090	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2

Table 24-708. PCIECTRL_RC_DBICS2_ROOT_STS (continued)

Description	Root Status Register
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td>PME_PND</td><td>PME_STS</td><td colspan="16">PME_RID</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	PME_PND	PME Pending	R	0x0
16	PME_STS	PME Status (Sticky bit)	RW	0x0
15:0	PME_RID	PME Requester ID	R	0x0

Table 24-709. PCIECTRL_RC_DBICS2_DEV_CAP_2

Address Offset	0x0000 0094		
Physical Address	0x5100 1094 0x5180 1094	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Device Capabilities 2 Register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="18">RESERVED</td><td colspan="2">TPHC_SP</td><td>RESERVED</td><td>NOROPR</td><td>CASC128_SP</td><td>AOC64_SP</td><td>AOC32_SP</td><td>AOR_SP</td><td>ARI_FWD_SP</td><td>CPL_TIMEOUT_DIS_SUPPORTED</td><td colspan="4">CPL_TIMEOUT_RNG_SUPPORTED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	TPHC_SP	TPH Completer Supported	R	0x0
11	RESERVED		R	0x0
10	NOROPR	No RO-enabled PR-PR Passing	R	0x1
9	CASC128_SP	128-bit CAS Completer Supported	R	0x0
8	AOC64_SP	64-bit AtomicOp Completer Supported	R	0x0
7	AOC32_SP	32-bit AtomicOp Completer Supported	R	0x0
6	AOR_SP	AtomicOp Routing Supported	R	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	R	0x0
4	CPL_TIMEOUT_DIS_SUPPORTED	Completion Timeout Disable Supported	R	0x1
3:0	CPL_TIMEOUT_RNG_SUPPORTED	Completion Timeout Ranges Supported	R	0xf

Table 24-710. PCIECTRL_RC_DBICS2_DEV_CAS_2

Address Offset	0x0000 0098		
Physical Address	0x5100 1098 0x5180 1098	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Device Control 2 Register		
Type	RW		

31:15	14:13	12:11	10	9	8	7	6	5	4	3:0
RESERVED	OBFF_EN	RESERVED	LTR_EN	IDO_CPL_EN	IDO_REQ_EN	AOP_EG_BLK	AOP_REQ_EN	ARI_FWD_SP	CPL_TIMEOUT_DIS	CPL_TIMEOUT_VALUE

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:13	OBFF_EN	OBFF Enable	RW	0x0
12:11	RESERVED		R	0x0
10	LTR_EN	LTR Mechanism Enable	RW	0x0
9	IDO_CPL_EN	IDO Completion Enable	RW	0x0
8	IDO_REQ_EN	IDO Request Enable	RW	0x0
7	AOP_EG_BLK	AtomicOp Egress Blocking	RW	0x0
6	AOP_REQ_EN	AtomicOp Requester Enable	RW	0x0
5	ARI_FWD_SP	ARI Forwarding Supported	RW	0x0
4	CPL_TIMEOUT_DIS	Completion Timeout Disable	RW	0x0
3:0	CPL_TIMEOUT_VALUE	Completion Timeout Values	RW	0x0

Table 24-711. PCIECTRL_RC_DBICS2_LNK_CAP_2

Address Offset	0x0000 009C		
Physical Address	0x5100 109C 0x5180 109C	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	PCIE Link Capabilities 2 Register		
Type	R		

31:9	8	7:1	0
RESERVED	CROSSLINK_SP	SP_LS_VEC	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CROSSLINK_SP	Crosslink Supported	R	0x0
7:1	SP_LS_VEC	Supported Link Speeds Vector	R	0x3
0	RESERVED		R	0x0

Table 24-712. PCIECTRL_RC_DBICS2_LNK_CAS_2

Address Offset	0x0000 00A0		
Physical Address	0x5100 10A0 0x5180 10A0	**Instance**	PCIe_SS1_RC_CFG_DBICS2 PCIe_SS2_RC_CFG_DBICS2
Description	Link Control and Status 2 Register (Sticky)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										LINK_EQ_REQ	EQ_PH3	EQ_PH2	EQ_PH1	EQ_COMPLETE	DEEMPH_LEVEL	COMPL_PRST_DEEPH				COMPL_SOS	ENT_MOD_COMPL	TX_MARGIN			SEL_DEEMP	HW_AUTO_SP_DIS	ENTR_COMPL	TRGT_LINK_SPEED			

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	LINK_EQ_REQ	Link Equilization Request	RW Wr1toClr	0x0
20	EQ_PH3	Equalization Ph3 Success, Gen3 Only	R	0x0
19	EQ_PH2	Equalization Ph2 Success, Gen3 Only	R	0x0
18	EQ_PH1	Equalization Ph1 Success, Gen3 Only	R	0x0
17	EQ_COMPLETE	Equalization Complete, Gen3 Only	R	0x0
16	DEEMPH_LEVEL	Current De-emphasis Level	R	0x1
15:12	COMPL_PRST_DEEPH	Compliance Pre-set/ De-emphasis	RW	0x0
11	COMPL_SOS	Compliance SOS	RW	0x0
10	ENT_MOD_COMPL	Enter Modified Compliance	RW	0x0
9:7	TX_MARGIN	Transmit Margin	RW	0x0
6	SEL_DEEMP	Selectable De-emphasis (CS)	RW	0x0
5	HW_AUTO_SP_DIS	Hardware Autonomous Speed Disable	RW	0x0
4	ENTR_COMPL	Enter Compliance	RW	0x0
3:0	TRGT_LINK_SPEED	Target Link Speed: Read 0x1: 2.5 GT/s (Gen1) Read 0x2: 5 GT/s (Gen2) Read 0x3: 8 GT/s (Gen3)	RW	0x2

24.9.7.8 PCIe_SS_TI_CONF Registers

24.9.7.8.1 PCIe_SS_TI_CONF Register Summary

Note

Though the PCIe controller wrapper TI configuration registers are also accessible (aliased) at base address 0x5100_3000 (PCIe_SS1) and 0x5180_3000 (PCIe_SS2), the preferable base addresses to access them are the ones used in above table, that is, 0x5100 2000 and 0x5180 2000. These registers are visible only within the device L3_MAIN space.

Table 24-713. PCIe_SS1_TI_CONF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS1_TI_CONF Physical Address
PCIECTRL_TI_CONF_REVISION	R	32	0x0000 0000	0x5100 2000
PCIECTRL_TI_CONF_SYSCONFIG	RW	32	0x0000 0010	0x5100 2010
PCIECTRL_TI_CONF_IRQ_EOI	RW	32	0x0000 0018	0x5100 2018
PCIECTRL_TI_CONF_IRQSTATUS_RAW_MAIN	RW	32	0x0000 0020	0x5100 2020
PCIECTRL_TI_CONF_IRQSTATUS_MAIN	RW	32	0x0000 0024	0x5100 2024
PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN	RW	32	0x0000 0028	0x5100 2028
PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN	RW	32	0x0000 002C	0x5100 202C
PCIECTRL_TI_CONF_IRQSTATUS_RAW_MSI	RW	32	0x0000 0030	0x5100 2030
PCIECTRL_TI_CONF_IRQSTATUS_MSI	RW	32	0x0000 0034	0x5100 2034
PCIECTRL_TI_CONF_IRQENABLE_SET_MSI	RW	32	0x0000 0038	0x5100 2038
PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI	RW	32	0x0000 003C	0x5100 203C
PCIECTRL_TI_CONF_DEVICE_TYPE	RW	32	0x0000 0100	0x5100 2100
PCIECTRL_TI_CONF_DEVICE_CMD	RW	32	0x0000 0104	0x5100 2104
PCIECTRL_TI_CONF_PM_CTRL	RW	32	0x0000 0108	0x5100 2108
PCIECTRL_TI_CONF_PHY_CS	RW	32	0x0000 010C	0x5100 210C
PCIECTRL_TI_CONF_INTX_ASSERT	RW	32	0x0000 0124	0x5100 2124
PCIECTRL_TI_CONF_INTX_DEASSERT	RW	32	0x0000 0128	0x5100 2128
PCIECTRL_TI_CONF_MSI_XMT	RW	32	0x0000 012C	0x5100 212C
PCIECTRL_TI_CONF_DEBUG_CFG	RW	32	0x0000 0140	0x5100 2140
PCIECTRL_TI_CONF_DEBUG_DATA	R	32	0x0000 0144	0x5100 2144
PCIECTRL_TI_CONF_DIAG_CTRL	RW	32	0x0000 0148	0x5100 2148

Table 24-714. PCIe_SS2_TI_CONF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_TI_CONF Physical Address
PCIECTRL_TI_CONF_REVISION	R	32	0x0000 0000	0x5180 2000
PCIECTRL_TI_CONF_SYSCONFIG	RW	32	0x0000 0010	0x5180 2010
PCIECTRL_TI_CONF_IRQ_EOI	RW	32	0x0000 0018	0x5180 2018
PCIECTRL_TI_CONF_IRQSTATUS_RAW_MAIN	RW	32	0x0000 0020	0x5180 2020
PCIECTRL_TI_CONF_IRQSTATUS_MAIN	RW	32	0x0000 0024	0x5180 2024
PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN	RW	32	0x0000 0028	0x5180 2028
PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN	RW	32	0x0000 002C	0x5180 202C
PCIECTRL_TI_CONF_IRQSTATUS_RAW_MSI	RW	32	0x0000 0030	0x5180 2030
PCIECTRL_TI_CONF_IRQSTATUS_MSI	RW	32	0x0000 0034	0x5180 2034
PCIECTRL_TI_CONF_IRQENABLE_SET_MSI	RW	32	0x0000 0038	0x5180 2038
PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI	RW	32	0x0000 003C	0x5180 203C

Table 24-714. PCIe_SS2_TI_CONF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe_SS2_TI_CONF Physical Address
PCIECTRL_TI_CONF_DEVICE_TYPE	RW	32	0x0000 0100	0x5180 2100
PCIECTRL_TI_CONF_DEVICE_CMD	RW	32	0x0000 0104	0x5180 2104
PCIECTRL_TI_CONF_PM_CTRL	RW	32	0x0000 0108	0x5180 2108
PCIECTRL_TI_CONF_PHY_CS	RW	32	0x0000 010C	0x5180 210C
PCIECTRL_TI_CONF_INTX_ASSERT	RW	32	0x0000 0124	0x5180 2124
PCIECTRL_TI_CONF_INTX_DEASSERT	RW	32	0x0000 0128	0x5180 2128
PCIECTRL_TI_CONF_MSI_XMT	RW	32	0x0000 012C	0x5180 212C
PCIECTRL_TI_CONF_DEBUG_CFG	RW	32	0x0000 0140	0x5180 2140
PCIECTRL_TI_CONF_DEBUG_DATA	R	32	0x0000 0144	0x5180 2144
PCIECTRL_TI_CONF_DIAG_CTRL	RW	32	0x0000 0148	0x5180 2148

24.9.7.8.2 PCIe_SS_TI_CONF Register Description

Table 24-715. PCIECTRL_TI_CONF_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5100 2000 0x5180 2000	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	IP Revision Identifier		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-(1)

(1) TI Internal data.

Table 24-716. PCIECTRL_TI_CONF_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5100 2010 0x5180 2010	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Controls various parameters of the master and slave interfaces.		
Type	RW		

31–17	16	15–6	5–4	3–2	1–0
RESERVED	MCOHERENT_EN	RESERVED	STANDBYMODE	IDLEMODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
16	MCOHERENT_EN	Allows the no-snoop (NS) attribute of inbound PCIe TLPs to be passed to SoC system bus (AXI) master as a 'coherent' inband flag. 0x0: DIS AXI not coherent 0x1: EN AXI coherent = not(PCIE "NS") that is, cache-coherence is preserved	RW	0x0
15:6	RESERVED		R	0x0
5:4	STANDBYMODE	PM mode of local initiator (master); Initiator may generate read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode = Initiator is unconditionally placed in standby state. 0x1: No-standby mode = initiator is unconditionally placed out of standby state. 0x2: Smart-standby mode = initiator's standby state depends on internal conditions, that is, the module's functional requirements. Asynchronous wakeup events cannot be generated. 0x3: Smart-Standby, wakeup-capable mode = initiator's standby state depends on internal conditions, ie the module's functional requirements. Asynchronous wakeup events can be generated.	RW	0x2
3:2	IDLEMODE	PM mode of local target (slave); Target shall be capable of handling read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode = local target's idle state follows (acknowledges) the system's idle requests unconditionally, regardless of the IP module's internal requirements. 0x1: No-idle mode = local target never enters idle state. 0x2: Smart-idle mode = local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. Module shall not generate (IRQ- or DMA-request-related) wakeup events. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.	RW	0x2
1:0	RESERVED		R	0x0

Table 24-717. PCIECTRL_TI_CONF_IRQ_EOI

Address offset	0x18		
Physical Address	0x5100 2018 0x5180 2018	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if an new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).		
Type	RW		

31–4	3–0
RESERVED	LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x000 0000

Bits	Field Name	Description	Type	Reset
3:0	LINE_NUMBER	Write the IRQ line number to apply software EOI to it. Write 0x0: software EOI on main interrupt line Read 0x0: Read always returns zeros Write 0x1: software EOI on message-signalled (MSI) interrupt line	RW	0x0

Table 24-718. PCIECTRL_TI_CONF_IRQSTATUS_RAW_MAIN

Address Offset	0x0000 0020		
Physical Address	0x5100 2020 0x5180 2020	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Raw status of 'main' interrupt requests; Set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug (regular status also gets set).		
Type	RW W1toSet		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CFG_MSE_EVT	CFG_BME_EVT	LINK_UP_EVT	LINK_REQ_RST	PM_PME	PME_TO_ACK	PME_TURN_OFF	RESERVED		ERR_ECRC	ERR_AXI	ERR_COR	ERR_NONFATAL	ERR_FATAL	ERR_SYS

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CFG_MSE_EVT	CFG 'Memory Space Enable' change IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
13	CFG_BME_EVT	CFG "Bus Master Enable" change IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
12	LINK_UP_EVT	Link-up state change IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
11	LINK_REQ_RST	Link Request Reset IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
10	PM_PME	PM Power Management Event message received IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
9	PME_TO_ACK	Power Management Event Turn-Off Ack message received IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0

Bits	Field Name	Description	Type	Reset
8	PME_TURN_OFF	Power Management Event Turn-Off message received IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
7:6	RESERVED		R	0x0
5	ERR_ECRC	ECRC Error IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
4	ERR_AXI	AXI tag lookup fatal Error IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
3	ERR_COR	Correctable Error message received IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
2	ERR_NONFATAL	Non-Fatal Error message received IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
1	ERR_FATAL	Fatal Error message received IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
0	ERR_SYS	System Error IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0

Table 24-719. PCIECTRL_TI_CONF_IRQSTATUS_MAIN

Address Offset	0x0000 0024		
Physical Address	0x5100 2024 0x5180 2024	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Regular status of 'main' interrupt requests; Set only when enabled. Write 1 to clear after interrupt has been serviced (raw status also gets cleared).		
Type	RW W1toClr		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CFG_MSE_EVT	CFG_BME_EVT	LINK_UP_EVT	LINK_REQ_RST	PM_PME	PME_TO_ACK	PME_TURN_OFF	RESERVED		ERR_ECRC	ERR_AXI	ERR_COR	ERR_NONFATAL	ERR_FATAL	ERR_SYS

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
14	CFG_MSE_EVT	CFG 'Memory Space Enable' change IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
13	CFG_BME_EVT	CFG 'Bus Master Enable' change IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
12	LINK_UP_EVT	Link-up state change IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
11	LINK_REQ_RST	Link Request Reset IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
10	PM_PME	PM Power Management Event message received IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
9	PME_TO_ACK	Power Management Event Turn-Off Ack message received IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
8	PME_TURN_OFF	Power Management Event Turn-Off message received IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
7:6	RESERVED		R	0x0
5	ERR_ECRC	ECRC Error IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
4	ERR_AXI	AXI tag lookup fatal Error IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
3	ERR_COR	Correctable Error message received IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
2	ERR_NONFATAL	Non-Fatal Error message received IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
1	ERR_FATAL	Fatal Error message received IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0
0	ERR_SYS	System Error IRQ status Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any Read 1: IRQ event pending	RW W1toClr	0x0

Table 24-720. PCIECTRL_TI_CONF_IRQENABLE_SET_MAIN

Address Offset	0x0000 0028		
Physical Address	0x5100 2028 0x5180 2028	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Enable of 'main' interrupt requests; Write 1 to set (ie to enable interrupt). Readout is the same as corresponding _CLR register.		
Type	RW W1toSet		

31:15	14	13	12	11	10	9	8	7:6	5	4	3	2	1	0
RESERVED	CFG_MSE_EVT_EN	CFG_BME_EVT_EN	LINK_UP_EVT_EN	LINK_REQ_RST_EN	PM_PME_EN	PME_TO_ACK_EN	PME_TURN_OFF_EN	RESERVED	ERR_ECRC_EN	ERR_AXI_EN	ERR_COR_EN	ERR_NONFATAL_EN	ERR_FATAL_EN	ERR_SYS_EN

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CFG_MSE_EVT_EN	CFG 'Memory Space Enable' change IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
13	CFG_BME_EVT_EN	CFG 'Bus Master Enable' change IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
12	LINK_UP_EVT_EN	Link-up state change IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
11	LINK_REQ_RST_EN	Link Request Reset IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
10	PM_PME_EN	PM Power Management Event message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0

Bits	Field Name	Description	Type	Reset
9	PME_TO_ACK_EN	Power Management Event Turn-Off Ack message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
8	PME_TURN_OFF_EN	Power Management Event Turn-Off message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
7:6	RESERVED		R	0x0
5	ERR_ECRC_EN	ECRC Error IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
4	ERR_AXI_EN	AXI tag lookup fatal Error IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
3	ERR_COR_EN	Correctable Error message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
2	ERR_NONFATAL_EN	Non-Fatal Error message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
1	ERR_FATAL_EN	Fatal Error message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
0	ERR_SYS_EN	System Error IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0

Table 24-721. PCIECTRL_TI_CONF_IRQENABLE_CLR_MAIN

Address Offset	0x0000 002C		
Physical Address	0x5100 202C 0x5180 202C	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Enable of 'main' interrupt requests; Write 1 to clear (ie to disable interrupt). Readout is the same as corresponding _SET register.		
Type	RW Wr1toClr		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	CFG_MSE_EVT_EN	CFG_BME_EVT_EN	LINK_UP_EVT_EN	LINK_REQ_RST_EN	PM_PME_EN	PME_TO_ACK_EN	PME_TURN_OFF_EN	RESERVED	ERR_ECRC_EN	ERR_AXI_EN	ERR_COR_EN	ERR_NONFATAL_EN	ERR_FATAL_EN	ERR_SYS_EN

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CFG_MSE_EVT_EN	CFG 'Memory Space Enable' change IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
13	CFG_BME_EVT_EN	CFG 'Bus Master Enable' change IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
12	LINK_UP_EVT_EN	Link-up state change IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
11	LINK_REQ_RST_EN	Link Request Reset IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
10	PM_PME_EN	PM Power Management Event message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
9	PME_TO_ACK_EN	Power Management Event Turn-Off Ack message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
8	PME_TURN_OFF_EN	Power Management Event Turn-Off message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
7:6	RESERVED		R	0x0
5	ERR_ECRC_EN	ECRC Error IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
4	ERR_AXI_EN	AXI tag lookup fatal Error IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0

Bits	Field Name	Description	Type	Reset
3	ERR_COR_EN	Correctable Error message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
2	ERR_NONFATAL_EN	Non-Fatal Error message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
1	ERR_FATAL_EN	Fatal Error message received IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0
0	ERR_SYS_EN	System Error IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW Wr1toClr	0x0

Table 24-722. PCIECTRL_TI_CONF_IRQSTATUS_RAW_MSI

Address Offset	0x0000 0030		
Physical Address	0x5100 2030 0x5180 2030	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Raw status of legacy and MSI interrupt requests; Set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug (regular status also gets set).		
Type	RW W1toSet		

31–5	4	3	2	1	0
RESERVED	MSI	INTD	INTC	INTB	INTA

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	MSI	Message Signaled Interrupt IRQ status Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
3	INTD	INTD IRQ status (Legacy PCIe message interrupt D); RC mode only. Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
2	INTC	INTC IRQ status (Legacy PCIe message interrupt C); RC mode only. Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0
1	INTB	INTB IRQ status (Legacy PCIe message interrupt B); RC mode only. Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0

Bits	Field Name	Description	Type	Reset
0	INTA	INTA IRQ status (Legacy PCIe message interrupt A); RC mode only. Write 0: No action Read 0: No event pending Write 1: Trigger IRQ event by software Read 1: IRQ event pending	RW W1toSet	0x0

Table 24-723. PCIECTRL_TI_CONF_IRQSTATUS_MSI

Address Offset	0x0000 0034		
Physical Address	0x5100 2034 0x5180 2034	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Regular status of legacy and MSI interrupt requests; Set only when enabled. Write 1 to clear after interrupt has been serviced (raw status also gets cleared). HW-generated events are self-clearing.		
Type	RW W1toClr		

31–5	4	3	2	1	0
RESERVED	MSI	INTD	INTC	INTB	INTA

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	MSI	Message Signaled Interrupt IRQ status. Cleared by clearing all vectors in the MSI controller (PL) registers Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any. Read 1: IRQ event pending	RW W1toClr	0x0
3	INTD	INTD IRQ status (Legacy PCIe message interrupt D); RC mode only. Typically set AND cleared by the remote EP, without local software intervention. Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any. Read 1: IRQ event pending	RW W1toClr	0x0
2	INTC	INTC IRQ status (Legacy PCIe message interrupt C); RC mode only. Typically set AND cleared by the remote EP, without local software intervention. Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any. Read 1: IRQ event pending	RW W1toClr	0x0
1	INTB	INTB IRQ status (Legacy PCIe message interrupt B); RC mode only. Typically set AND cleared by the remote EP, without local software intervention. Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any. Read 1: IRQ event pending	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
0	INTA	INTA IRQ status (Legacy PCIe message interrupt A); RC mode only. Typically set AND cleared by the remote EP, without local software intervention. Write 0: No action Read 0: No event pending Write 1: Clear pending event, if any. Read 1: IRQ event pending	RW W1toClr	0x0

Table 24-724. PCIECTRL_TI_CONF_IRQENABLE_SET_MSI

Address Offset	0x0000 0038		
Physical Address	0x5100 2038 0x5180 2038	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Enable of legacy and MSI interrupt requests; Write 1 to set (ie to enable interrupt). Readout is the same as corresponding _CLR register.		
Type	RW W1toSet		

31:5	4	3	2	1	0
RESERVED	MSI_EN	INTD_EN	INTC_EN	INTB_EN	INTA_EN

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	MSI_EN	Message Signaled Interrupt IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
3	INTD_EN	INTD IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
2	INTC_EN	INTC IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
1	INTB_EN	INTB IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0
0	INTA_EN	INTA IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Set IRQ enable (that is, enable event) Read 1: IRQ event is enabled	RW W1toSet	0x0

Table 24-725. PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI

Address Offset	0x0000 003C		
Physical Address	0x5100 203C 0x5180 203C	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Enable of legacy and MSI interrupt requests; Write 1 to clear (ie to disable interrupt). Readout is the same as corresponding _SET register.		

Table 24-725. PCIECTRL_TI_CONF_IRQENABLE_CLR_MSI (continued)

Type	RW W1toClr

31–5	4	3	2	1	0
RESERVED	MSI_EN	INTD_EN	INTC_EN	INTB_EN	INTA_EN

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	MSI_EN	Message Signaled Interrupt IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW W1toClr	0x0
3	INTD_EN	INTD IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW W1toClr	0x0
2	INTC_EN	INTC IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW W1toClr	0x0
1	INTB_EN	INTB IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW W1toClr	0x0
0	INTA_EN	INTA IRQ enable Write 0: No action Read 0: IRQ event is disabled Write 1: Clear IRQ enable (that is, disable event) Read 1: IRQ event is enabled	RW W1toClr	0x0

Table 24-726. PCIECTRL_TI_CONF_DEVICE_TYPE

Address Offset	0x0000 0100		
Physical Address	0x5100 2100 0x5180 2100	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Sets the Dual-Mode device's type		
Type	RW		

31–4	3–0
RESERVED	TYPE

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	TYPE	PCIe device type including the contents of the PCI config space (Type-0 for EP, Type-1 for RC); Apply fundamental reset after change; Do not change during core operation; 0x0: PCIe endpoint (EP) 0x1: Legacy PCIe endpoint (LEG_EP) 0x4: Root Complex (RC) Other values: Reserved	RW	0x4

Table 24-727. PCIECTRL_TI_CONF_DEVICE_CMD

Address Offset	0x0000 0104		
Physical Address	0x5100 2104 0x5180 2104	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Device command (startup control and status); WARNING: cleared by all reset conditions, including fundamental reset		
Type	RW		

31:29	28:21	20:16	15:8	7:2	1	0
RESERVED	BUS_NUM	DEV_NUM	RESERVED	LTSSM_STATE	APP_REQ_RETRY_EN	LTSSM_EN

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:21	BUS_NUM	PCIe bus number	R	0x0
20:16	DEV_NUM	PCIe device number	R	0x0
15:8	RESERVED		R	0x0
7:2	LTSSM_STATE	LTSSM state /substate, implementation-specific, for debug Read 0x00: DETECT_QUIET Read 0x01: DETECT_ACT Read 0x02: POLL_ACTIVE Read 0x03: POLL_COMPLIANCE Read 0x04: POLL_CONFIG Read 0x05: PRE_DETECT_QUIET Read 0x06: DETECT_WAIT Read 0x07: CFG_LINKWD_START Read 0x08: CFG_LINKWD_ACEPT Read 0x09: CFG_LANENUM_WAIT Read 0x0A: CFG_LANENUM_ACEPT Read 0x0B: CFG_COMPLETE Read 0x0C: CFG_IDLE Read 0x0D: RCVRY_LOCK Read 0x0E: RCVRY_SPEED Read 0x0F: RCVRY_RCVRCFG Read 0x10: RCVRY_IDLE Read 0x11: L0 Read 0x12: L0S Read 0x13: L123_SEND_EIDLE Read 0x14: L1_IDLE Read 0x15: L2_IDLE Read 0x16: L2_WAKE Read 0x17: DISABLED_ENTRY Read 0x18: DISABLED_IDLE Read 0x19: DISABLED Read 0x1A: LPBK_ENTRY Read 0x1B: LPBK_ACTIVE Read 0x1C: LPBK_EXIT Read 0x1D: LPBK_EXIT_TIMEOUT Read 0x1E: HOT_RESET_ENTRY Read 0x1F: HOT_RESET Read 0x20: RCVRY_EQ0 Read 0x21: RCVRY_EQ1 Read 0x22: RCVRY_EQ2 Read 0x23: RCVRY_EQ3	R	0x0

Bits	Field Name	Description	Type	Reset
1	APP_REQ_RETRY_EN	Application Request Retry Enable (This bit is CLEARED BY FUNDAMENTAL RESET) 0x0: DISABLED Incoming PCI transactions are processed normally 0x1: ENABLED Incoming PCI transactions are responded with "retry"	RW	0x0
0	LTSSM_EN	LTSSM enable: start the PCI link (This bit is CLEARED BY FUNDAMENTAL RESET) 0x0: DISABLED 0x1: ENABLED	RW	0x0

Table 24-728. PCIECTRL_TI_CONF_PM_CTRL

Address Offset	0x0000 0108		
Physical Address	0x5100 2108 0x5180 2108	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Power Management Control		
Type	RW		

31:12	11	10	9	8	7:2	1	0
RESERVED	AUX_PWR_DET	REQ_EXIT_L1	REQ_ENTR_L1	L23_READY	RESERVED	PM_PME	PME_TURN_OFF

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	AUX_PWR_DET	Auxilliary Power Detection; Status of Vaux detection for the PCIe controller; Determines transition to L2 vs L3 upon Vmain turn-off. 0x0: UNPOWERED Vaux not present: D3cold maps to L3 link state 0x1: POWERED Vaux present: D3cold maps to L2 link state	RW	0x0
10	REQ_EXIT_L1	Request to exit L1 state (to L0) 0x0: INACTIVE No request 0x1: ACTIVE L1 exit request	RW	0x0
9	REQ_ENTR_L1	Request to transition to L1 state 0x0: INACTIVE No request 0x1: ACTIVE L1 entry request	RW	0x0
8	L23_READY	Indicates system readiness for the link to enter L2/L3 ready state (EP mode only); Allows the transmission of PM_Enter_L23 following PM_Turn_OFF/PME_TO_Ack handshake. Self-cleared upon transition to L2/L3. 0x0: not_READY 0x1: READY	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	PM_PME	Transmits PM_PME wakeup message (EP mode only) 0x0: NOACTION 0x1: TRANSMIT	W	0x0
0	PME_TURN_OFF	Transmits PME_Turn_Off message downstream (RC mode only); Eventually sends all links of hierarchy domain to L2L/3_ready 0x0: NOACTION 0x1: TRANSMIT	W	0x0

Table 24-729. PCIECTRL_TI_CONF_PHY_CS

Address Offset	0x0000 010C		
Physical Address	0x5100 210C 0x5180 210C	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Physical Layer Control and Status		
Type	RW		

31:17	16	15:1	0
RESERVED	LINK_UP	RESERVED	REVERSE_LANES

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	LINK_UP	Link status, from LTSSM 0x0: DOWN 0x1: UP	R	0x0
15:1	RESERVED		R	0x0
0	REVERSE_LANES	Manual lane reversal control, allowing lane 0 and lane 1 to be swapped by default; Both Tx and Rx are reversed; Polarity of the individual lane is unchanged 0x0: STRAIGHT 0x1: REVERSED	RW	0x0

Table 24-730. PCIECTRL_TI_CONF_INTX_ASSERT

Address Offset	0x0000 0124		
Physical Address	0x5100 2124 0x5180 2124	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Legacy INTx ASSERT message control, with 'x' in (A,B,C,D) set by the 'Interrupt Pin' field. Write 1 to send message, read to get the status; EP mode only		
Type	RW		

31:1	0
RESERVED	ASSERT_F0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	ASSERT_F0	INTx ASSERT for function 0 Write 0: No action Read 0: INTx is inactive (has been deasserted) Write 1: Transmit **INTx ASSERT** to RC Read 1: INTx is active (has been asserted)	RW	0x0

Table 24-731. PCIECTRL_TI_CONF_INTX_DEASSERT

Address Offset	0x0000 0128		
Physical Address	0x5100 2128 0x5180 2128	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Legacy INTx DEASSERT message control, with 'x' in (A,B,C,D) set by the 'Interrupt Pin' field. Write 1 to send message, read to get the status; EP mode only		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DEASSERT_F0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	DEASSERT_F0	INTx DEASSERT for function 0 Write 0: No action Read 0: INTx is inactive (has been deasserted) Write 1: Transmit **INTx DEASSERT** to RC Read 1: INTx is active (has been asserted)	RW	0x0

Table 24-732. PCIECTRL_TI_CONF_MSI_XMT

Address Offset	0x0000 012C		
Physical Address	0x5100 212C 0x5180 212C	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	MSI transmitter (EP mode); Specifies parameters of MSI, together with MSI capability descriptor already configured by remote RC.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				MSI_VECTOR					MSI_TC			MSI_FUNC_NUM			MSI_REQ_GRANT

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:7	MSI_VECTOR	Vector number for transmitted MSI (as allowed by RC at enumeration)	RW	0x0
6:4	MSI_TC	Traffic class (TC) for transmitted MSI	RW	0x0
3:1	MSI_FUNC_NUM	Function number for transmitted MSI; Always 0 for single-function EP	RW	0x0

Bits	Field Name	Description	Type	Reset
0	MSI_REQ_GRANT	MSI transmit request (and grant status) Write 0: No Action Read 0: MSI transmission request pending Read 1: No MSI request pending (last request granted) Write 1: Request MSI transmission	RW	0x0

Table 24-733. PCIECTRL_TI_CONF_DEBUG_CFG

Address Offset	0x0000 0140		
Physical Address	0x5100 2140 0x5180 2140	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Configuration of debug_data output and register (observability)		
Type	RW		

31:6	5:0
RESERVED	SEL

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	SEL	Debug_data mode	RW	0x0

Table 24-734. PCIECTRL_TI_CONF_DEBUG_DATA

Address Offset	0x0000 0144		
Physical Address	0x5100 2144 0x5180 2144	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Debug data vector, depending on DEBUG_CFG.sel value		
Type	R		

31:0
DEBUG

Bits	Field Name	Description	Type	Reset
31:0	DEBUG		R	0x0

Table 24-735. PCIECTRL_TI_CONF_DIAG_CTRL

Address Offset	0x0000 0148		
Physical Address	0x5100 2148 0x5180 2148	**Instance**	PCIe_SS1_TI_CONF PCIe_SS2_TI_CONF
Description	Diagnostic control		
Type	RW		

31:2	1	0
RESERVED	INV_ECRC	INV_LCRC

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	RESERVED	software must always keep this bit at its default value - 0.	RW	0
1	INV_ECRC	Corrupt LSB of ECRC in the next packet, then self-clears. Read 0: No CRC corruption pending Read 1: CRC corruption pending Write 1: Request CRC corruption	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INV_LCRC	Corrupts LSB of LCRC in the next packet, then self-clears. Read 0: No CRC corruption pending Read 1: CRC corruption pending Write 1: Request CRC corruption	RW	0x0

24.10 DCAN

This chapter describes the controller area network modules (DCAN) in the device.

24.10.1 DCAN Overview

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time applications. CAN has high immunity to electrical interference and the ability to self-diagnose and repair data errors. In a CAN network, many short messages are broadcast to the entire network, which provides for data consistency in every node of the system.

The device supports two DCAN modules, referred to as DCAN1 and DCAN2, connecting to the CAN network through external (for the device) transceivers. The DCAN modules support bit rates up to 1 Mbit/s and are compliant to the CAN 2.0B protocol specification

Figure 24-164 shows the DCAN1 module highlights.

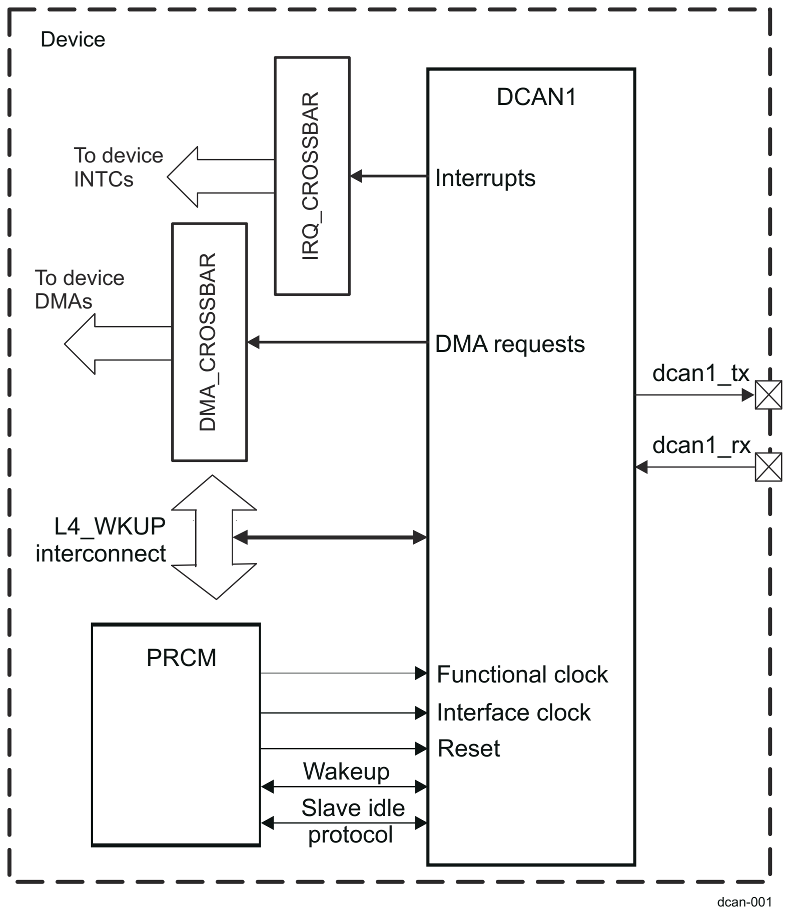


Figure 24-164. DCAN1 Overview

Figure 24-165 shows the DCAN2 module highlights .

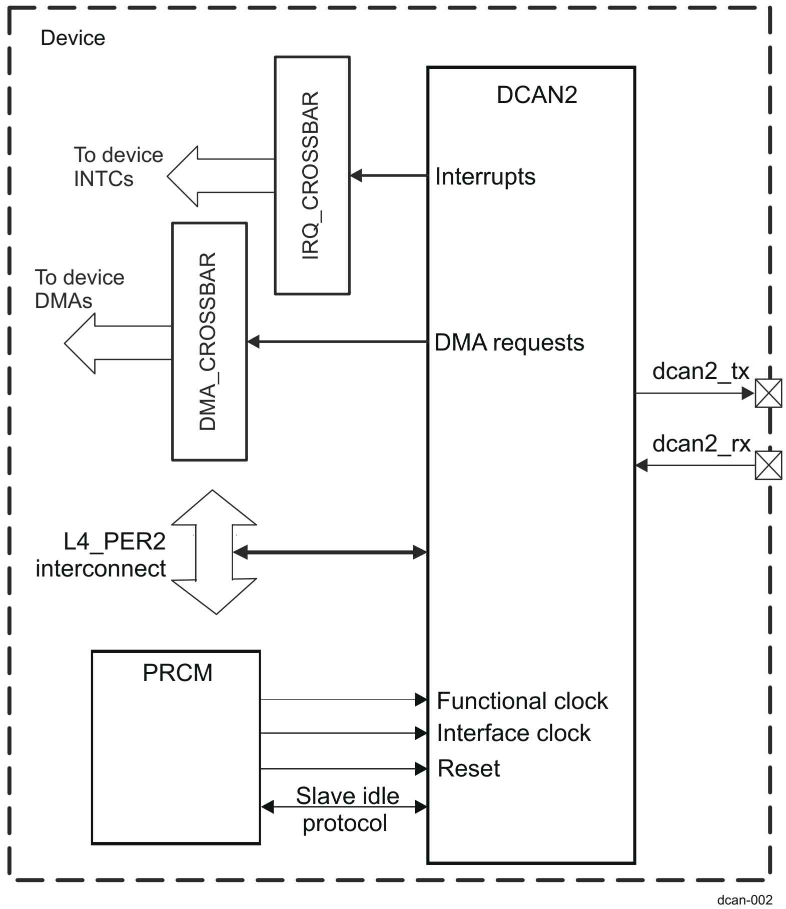


Figure 24-165. DCAN2 Overview

24.10.1.1 Features

The DCAN module implements the following features:

- Support for CAN protocol version 2.0 part A, B
- Bit rates up to 1 Mbit/s
- 64 message objects in a dedicated message RAM
- Individual identifier mask for each message object
- Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation
- Suspend mode for debug support
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- Message RAM parity check mechanism
- Direct access to message RAM during test mode
- Support for two interrupt lines: Level 0 and Level 1, plus separate parity error interrupt line
- Local power down and wakeup support
- Automatic message RAM initialization
- Support for DMA access

24.10.2 DCAN Environment

CAN network physical layer consists of two-wire differential bus, usually twisted pair, and provides high level of interference immunity. External CAN transceiver IC is needed to access a CAN bus by the DCAN.

Figure 24-166 shows an overview of a typical DCAN application.

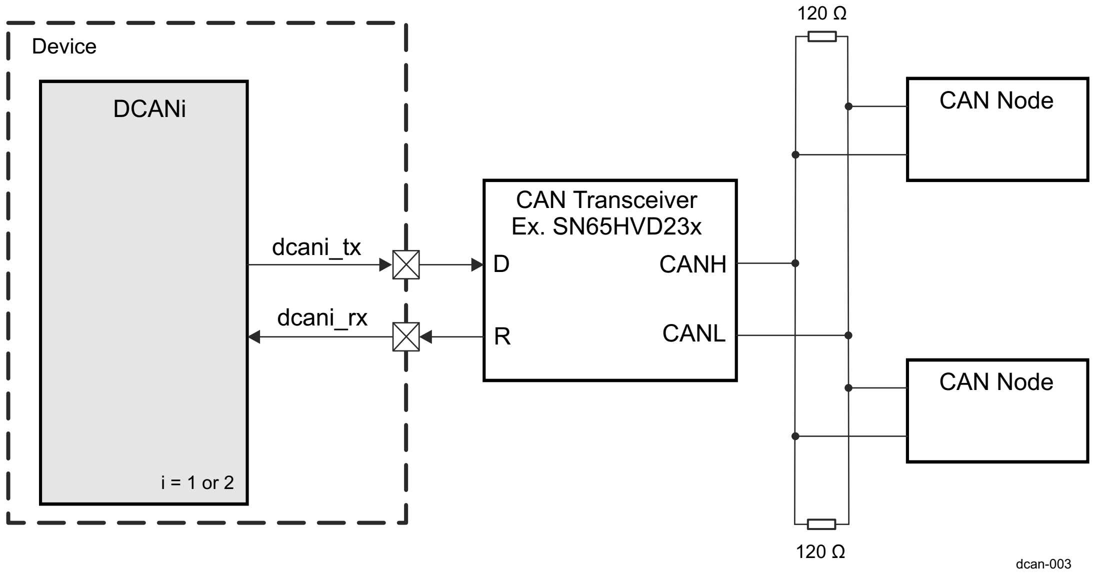


Figure 24-166. DCAN Typical Application

Table 24-736 describes the external signals of the DCAN1 module .

Table 24-736. DCAN2 I/O Description

Module Signal	Device Signal	I/O[(1)]	Description	Value at Reset
CAN_RX	dcan2_rx	I	Serial data input from external CAN tranceiver	HiZ
CAN_TX	dcan2_tx	O	Serial data output to external CAN tranceiver	1

(1) I = Input; O = Output

Table 24-737 describes the external signals of the DCAN2 module .

Table 24-737. DCAN1 I/O Description

Module Signal	Device Signal	I/O[(1)]	Description	Value at Reset
CAN_RX	dcan1_rx	I	Serial data input from external CAN tranceiver	HiZ
CAN_TX	dcan1_tx	O	Serial data output to external CAN tranceiver	1

(1) I = Input; O = Output

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see *Pad Configuration Registers* of *Control Module*.

24.10.2.1 CAN Network Basics

- CAN bus is a 2-wire differential bus using NRZ encoding and has two states:
  – Recessive state (logic 1)

  - Dominant state (logic 0)
- The network is multimaster. When two or more nodes attempt to transmit at the same time, a non-destructive arbitration technique guarantees messages are sent in order of priority and no messages are lost
- The message transmission is multicast. Data messages transmitted are identifier based, not address based
- Content of message is labeled by the identifier that is unique throughout the network (for example, rpm, temperature, position, pressure, etc.)
- All nodes on network receive the message and each performs an acceptance test on the identifier. If message is relevant, it is processed, otherwise it is ignored
- The unique identifier also determines the priority of the message (the lower the numerical value of the identifier, the higher the priority is)
- Data is transmitted and received using message frames, consisting of:
  - Arbitration field
  - Control field
  - Data field (0 ÷ 8 bytes)
  - CRC field
  - ACK field.

24.10.3 DCAN Integration

Figure 24-167 shows the integration of the DCAN1 module in the device.

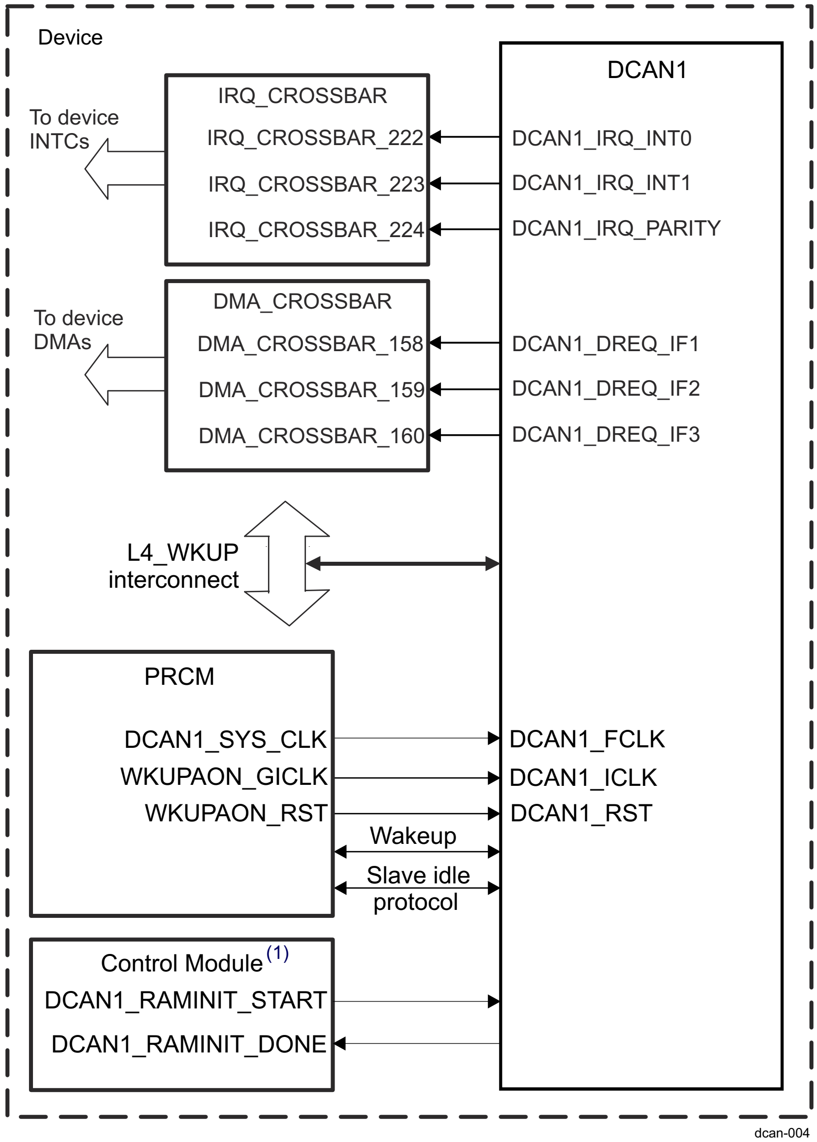


Figure 24-167. DCAN1 Integration

(1) For more information on DCAN RAM Initialization see Section 24.10.4.12.1.3, *DCAN RAM Hardware Initialization*

Figure 24-168 shows the integration of the DCAN2 module in the device .

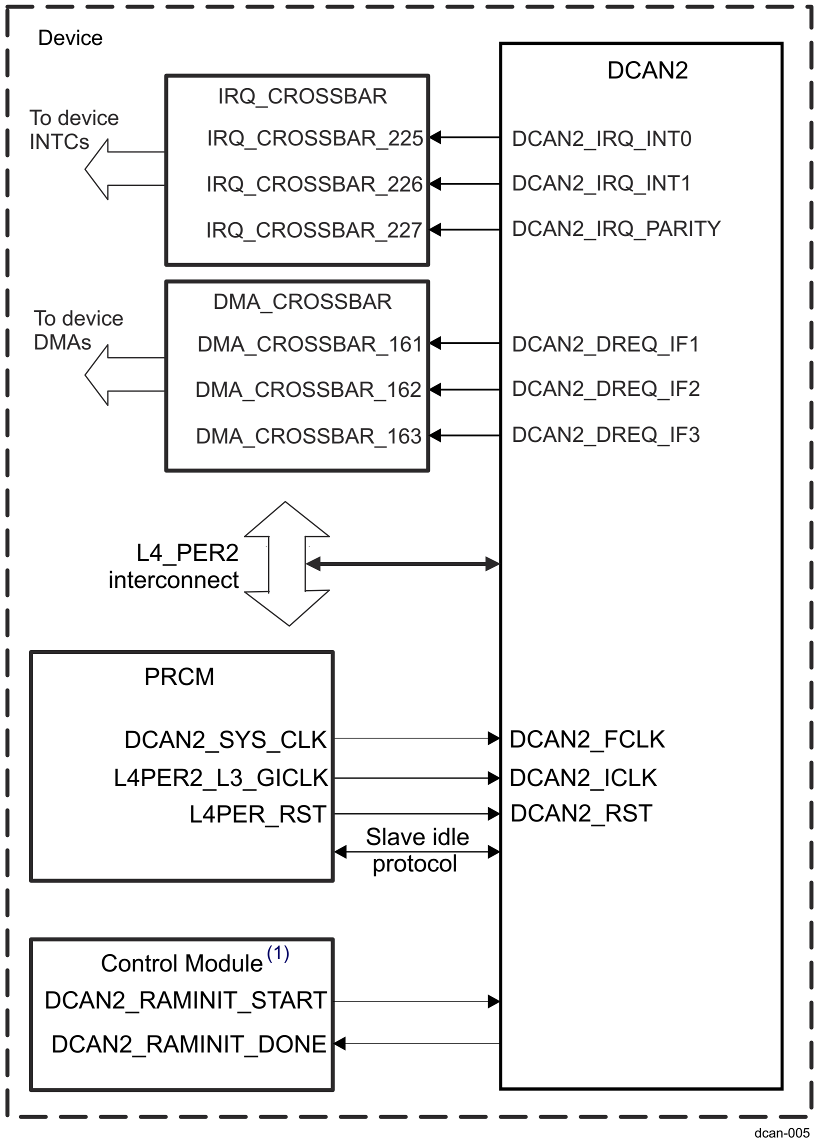


Figure 24-168. DCAN2 Integration

(1) For more information on DCAN RAM Initialization see Section 24.10.4.12.1.3, *DCAN RAM Hardware Initialization*

Table 24-738 through Table 24-740 summarize the integration of the DCAN modules in the device.

Table 24-738. DCAN Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
DCAN1	PD_WKUPAON	Yes	L4_WKUP
DCAN2	PD_COREAON	No	L4_PER2

Table 24-739. DCAN Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DCAN1	DCAN1_ICLK	WKUPAON_GICLK	PRCM	Interface clock for the DCAN1 module

Table 24-739. DCAN Clocks and Resets (continued)

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
	DCAN1_FCLK	DCAN1_SYS_CLK	PRCM	Functional clock for the DCAN1 core (CAN_CLK). Gated SYS_CLK1 or SYS_CLK2 version.
DCAN2	DCAN2_ICLK	L4PER2_L3_GICLK	PRCM	Interface clock for the DCAN2 module
	DCAN2_FCLK	DCAN2_SYS_CLK	PRCM	Functional clock for the DCAN2 core (CAN_CLK). Gated SYS_CLK1 version.
		Resets		
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DCAN1	DCAN1_RST	WKUPAON_RST	PRCM	Asynchronous reset signal to the DCAN1 module
DCAN2	DCAN2_RST	L4PER_RST	PRCM	Asynchronous reset signal to the DCAN2 module

Table 24-740. DCAN Hardware Requests

		Interrupt Requests		
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DCAN1	DCAN1_IRQ_INT0	IRQ_CROSSBAR_222	-	Error, Status, and Message Objects interrupt
	DCAN1_IRQ_INT1	IRQ_CROSSBAR_223	-	Message Objects interrupt
	DCAN1_IRQ_PARITY	IRQ_CROSSBAR_224	-	Parity error interrupt
DCAN2	DCAN2_IRQ_INT0	IRQ_CROSSBAR_225	-	Error, Status, and Message Objects interrupt
	DCAN2_IRQ_INT1	IRQ_CROSSBAR_226	-	Message Objects interrupt
	DCAN2_IRQ_PARITY	IRQ_CROSSBAR_227	-	Parity error interrupt
		DMA Requests		
Module Instance	**Source Signal Name**	**DMA_CROSSBAR Input**	**Default Mapping**	**Description**
DCAN1	DCAN1_DREQ_IF1	DMA_CROSSBAR_158	-	DMA request for IF1 register set
	DCAN1_DREQ_IF2	DMA_CROSSBAR_159	-	DMA request for IF2 register set
	DCAN1_DREQ_IF3	DMA_CROSSBAR_160	-	DMA request for IF3 register set
DCAN2	DCAN2_DREQ_IF1	DMA_CROSSBAR_161	-	DMA request for IF1 register set
	DCAN2_DREQ_IF2	DMA_CROSSBAR_162	-	DMA request for IF2 register set
	DCAN2_DREQ_IF3	DMA_CROSSBAR_163	-	DMA request for IF3 register set

Note

DCAN has no default IRQ mappings through the IRQ_CROSSBAR. For DCAN, the IRQ_CROSSBAR module must be configured prior to unmask interrupts in the interrupt controller(s).
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.

Note

For the description of the interrupt source, see Section 24.10.4.2, *Interrupt Functionality*.

24.10.4 DCAN Functional Description

The DCAN module performs CAN protocol communication according to ISO 11898-1. The bit rate can be programmed to values up to 1 Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

For communication on a CAN network, individual message objects can be configured. The message objects and identifier masks are stored in the message RAM.

All functions concerning the handling of messages are implemented in the message handler. Those functions are acceptance filtering, the transfer of messages between the CAN core and the message RAM, and the handling of transmission requests, as well as the generation of interrupts or DMA requests.

The register set of the DCAN module can be accessed directly via the module interface. These registers are used to control and configure the CAN core and the message handler, and to access the message RAM.

Figure 24-169 shows the DCAN block diagram and its features are described below.

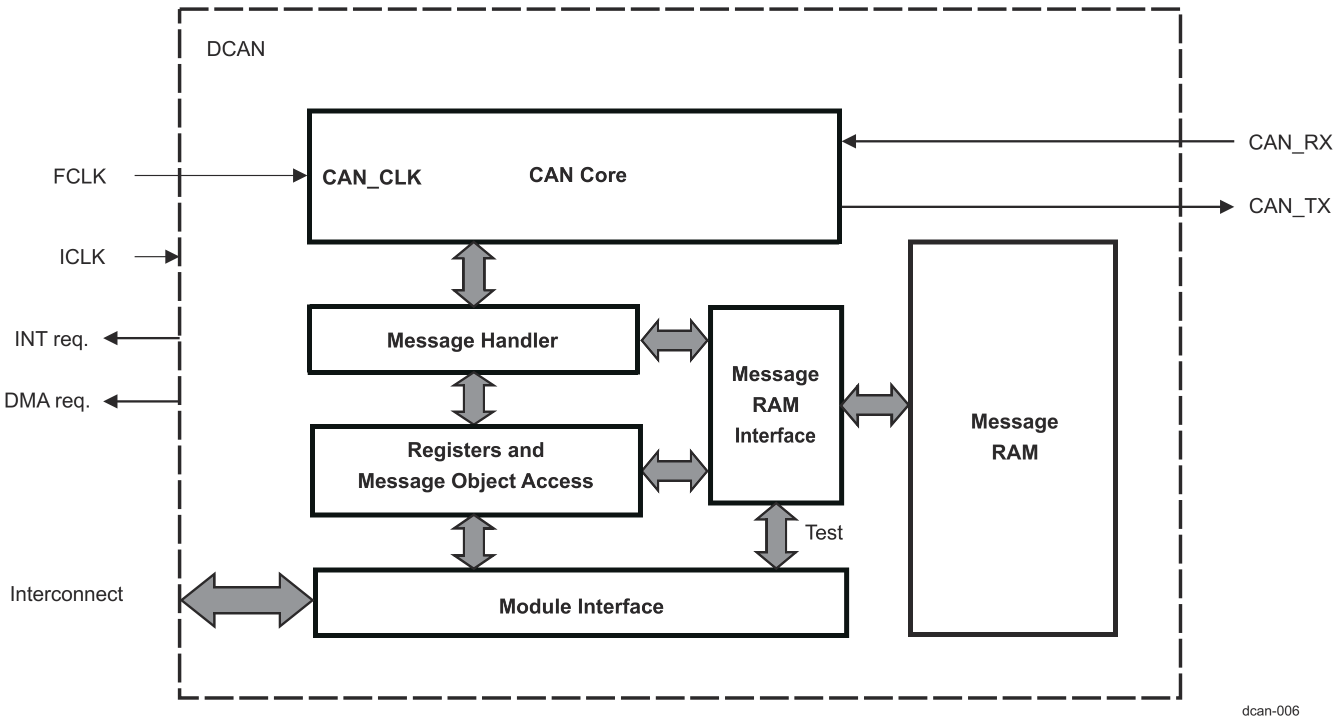


Figure 24-169. DCAN Block Diagram

CAN Core: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1 protocol functions.

Message Handler: The message handler is a state machine that controls the data transfer between the single-ported message RAM and the CAN core's Rx/Tx shift register. It also handles acceptance filtering and the interrupt/DMA request generation as programmed in the control registers.

Message RAM: The DCAN enables a storage of 64 CAN messages.

Message RAM Interface: Three interface register sets control the MPU read and write accesses to the message RAM. There are two interface registers sets for read and write access, IF1 and IF2, and one interface register set for read access only, IF3. Additional information can be found in Section 24.10.4.8.12, *Reading From a FIFO Buffer*.

The interface registers have the same word-length as the message RAM.

Registers and Message Object Access: Data consistency is ensured by indirect accesses to the message objects. During normal operation, all software and DMA accesses to the message RAM are done through interface registers. In a dedicated test mode, the message RAM is memory mapped and can be directly accessed by either MPU or DMA.

Module Interface: The DCAN module registers are accessed by the user software through a 32-bit peripheral bus interface.

Clocking: Two clocks are provided to the DCAN module: the peripheral synchronous clock (interface clock [ICLK]) and the peripheral asynchronous clock (functional clock [FCLK]) .

24.10.4.1 Module Clocking Requirements

Two clocks are provided to the DCAN module:

- the peripheral synchronous clock (ICLK) as the general module clock source
- and the peripheral asynchronous clock (FCLK) provided to the CAN core for generating the CAN bit timing.

Note

ICLK must always be higher or equal to FCLK, in order to achieve a stable functionality of the DCAN. Here, also the frequency shift of the modulated ICLK has to be considered:

$f_{0,ICLK}(OCP) \pm \Delta f_{FM,ICLK}(OCP) \geq f_{FCLK}$

For more information on how to configure the relevant clock source registers, see *PRCM* and the device data manual.

24.10.4.2 Interrupt Functionality

Interrupts can be generated on two interrupt lines: INT0 and INT1. These lines can be enabled by setting the DCAN_CTL[1] IE0 and [17] IE1 bits, respectively. The interrupts are level triggered at the chip level.

The DCAN provides three groups of interrupt sources: message object interrupts, status change interrupts, and error interrupts (see Figure 24-170, *Error and Status Change Interrupts* and Figure 24-171, *Message Objects Interrupts*).

The source of an interrupt can be determined by the interrupt identifiers DCAN_INT[15:0] INT0ID/[23:16] INT1ID. When no interrupt is pending, the register will hold the value zero.

Each interrupt line remains active until the dedicated field in the interrupt register DCAN_INT[15:0] INT0ID/ [23:16] INT1ID again reach zero (this means the cause of the interrupt is reset), or until IE0/IE1 are reset.

The value 0x8000 in the INT0ID field indicates that an interrupt is pending because the CAN core has updated (not necessarily changed) the Error and Status register (DCAN_ES). This interrupt has the highest priority. The software can update (reset) the status bits [9] WAKEUPPND, [4] RXOK, [3] TXOK and [2:0] LEC by reading DCAN_ES, but a write access of the software will never generate or reset an interrupt.

Values between 1 and the number of the last message object indicates that the source of the interrupt is one of the message objects, INT0ID resp. INT1ID will point to the pending message interrupt with the highest priority. The Message Object 1 has the highest priority; the last message object has the lowest priority.

An interrupt service routine that reads the message that is the source of the interrupt may read the message and reset the message object's IntPnd at the same time (DCAN_IF1CMD/DCAN_IF2CMD[19] CLRINTPND bit). When IntPnd is cleared, DCAN_INT will point to the next message object with a pending interrupt.

24.10.4.2.1 Message Object Interrupts

Message object interrupts are generated by events from the message objects. They are controlled by the flags IntPnd, TxIE and RxIE that are described in Section 24.10.4.11.1, *Structure of Message Objects*.

Message object interrupts can be routed to either INT0 or INT1 line, controlled by the interrupt multiplexer registers (DCAN_INTMUX12 to DCAN_INTMUX78).

24.10.4.2.2 Status Change Interrupts

The events WAKEUPPND, RXOK, TXOK and LEC in the error and status register (DCAN_ES) belong to the status change interrupts. The status change interrupt group can be enabled by DCAN_CTL[2] SIE bit.

If SIE is set, a status change interrupt will be generated at each CAN frame, independent of bus errors or valid CAN communication, and also independent of the message RAM configuration.

Status change interrupts can only be routed to interrupt line INT0, which has to be enabled by setting DCAN_CTL[1] IE0 = 1.

Note

Reading DCAN_ES will clear the WAKEUPPND flag. If in global power-down mode, the WAKEUPPND flag is cleared by such a read access before the DCAN module has been waken up by the system, the DCAN may re-assert the WAKEUPPND flag, and a second interrupt may occur.

24.10.4.2.3 Error Interrupts

The events PER, BOFF and EWARN, monitored in the DCAN_ES register belong to the error interrupts. The error interrupt group can be enabled by setting bit DCAN_CTL[3] EIE = 1.

Error interrupts can only be routed to interrupt line INT0, which has to be enabled by setting DCAN_CTL[1] IE0 = 1.

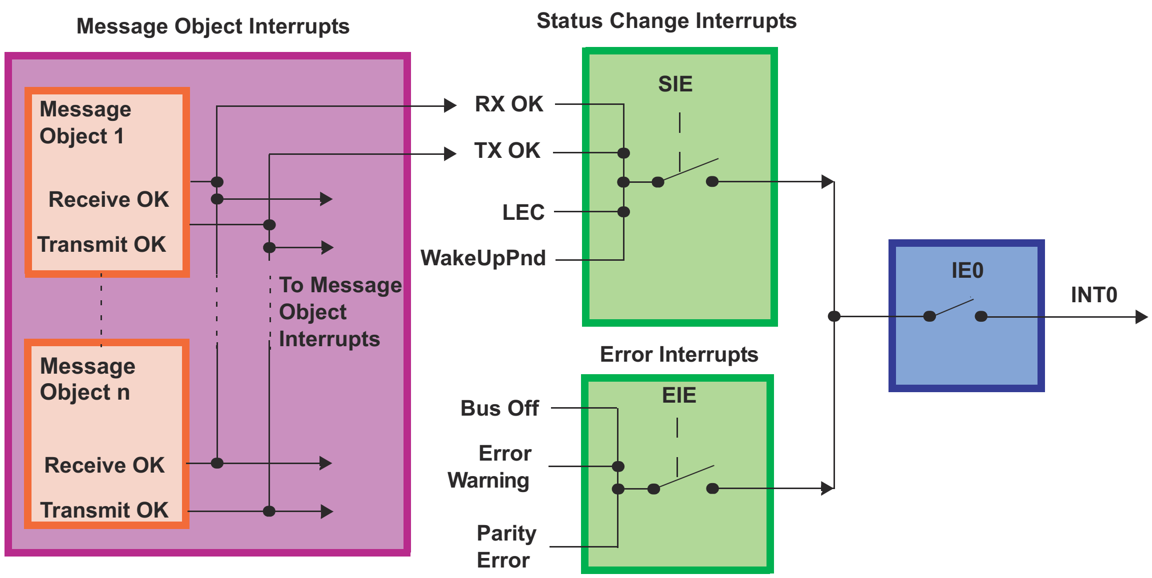


Figure 24-170. Error and Status Change Interrupts

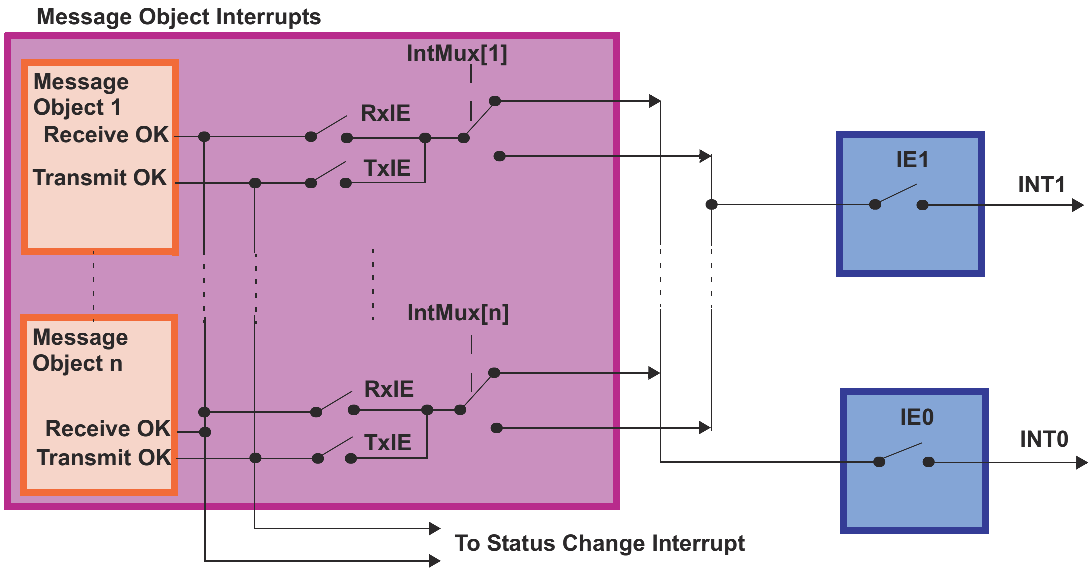


Figure 24-171. Message Objects Interrupts

24.10.4.3 DMA Functionality

The DCAN provides three DMA request lines, each indicating new data in one of the three interface register sets IF1, IF2 and IF3.

The update of IF1 and IF2 registers will be initiated by a write access to the IF1 respective IF2 Command Registers (DCAN_IF1CMD, DCAN_IF2CMD).

The IF3 registers content can be automatically updated on reception of CAN messages in message objects which are programmed for automatic IF3 update, see Section 24.10.4.10.2, *IF3 Register Set*.

When a DCAN internal IFx (x = 1 to 3) update is complete, a DMA request will be activated and stays active until the first access to one of the relevant IFx registers. The DMA functionality has to be enabled by setting bit [18] DE0/[19] DE1/[20] DE3 in DCAN_CTL.

24.10.4.4 Local Power-Down Mode

The DCAN supports a local power-down mode, which can be controlled within the DCAN registers.

24.10.4.4.1 Entering Local Power-Down Mode

The local power-down mode is requested by setting the DCAN_CTL[24] PDR bit (=1).

The DCAN then finishes all transmit requests of the message objects. When all requests are done, DCAN waits until a bus idle state is recognized. Then it will automatically set the DCAN_CTL[0] INIT bit to prevent any further CAN transfers, and it will also set the DCAN_ES[10] PDA bit. With setting the PDA bit, the DCAN module indicates that the local power-down mode has been entered.

During local power-down mode, the internal clocks of the DCAN module are turned off, but there is a wakeup logic (see Section 24.10.4.4.2, *Wakeup From Local Power Down*) that can be active, if enabled. Also, the actual contents of the control registers can be read back.

Note

In local low-power mode, the software must not clear the INIT bit while PDR is set. If there are any messages in the message RAM which are configured as transmit messages and the application resets the INIT bit, these messages may get sent.

24.10.4.4.2 Wakeup From Local Power Down

There are two ways to wake up the DCAN from local power-down mode:

- The application could wake up the DCAN module manually by clearing the DCAN_CTL[24] PDR bit and then clearing the DCAN_CTL[0] INIT.
- Alternatively, a CAN bus activity detection circuit can be activated by setting the wakeup on bus activity bit (DCAN_CTL[25] WUBA). If this circuit is active, on occurrence of a dominant CAN bus level, the DCAN will automatically start the wakeup sequence. It will clear the DCAN_CTL[24] PDR bit and also clear the DCAN_ES[10] PDA bit. The DCAN_ES[10] WAKEUPPND bit will be set. If status interrupts are enabled, also an interrupt will be generated. Finally the DCAN_CTL[0] INIT bit will be cleared.

After the INIT bit has been cleared, the module waits until it detects 11 consecutive recessive bits on the CAN_RX pin and then goes bus-active again.

Note

The CAN transceiver circuit has to stay active in order to detect any CAN bus activity while the DCAN is in local power down mode. The first CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power-down and automatic wake-up mode, is lost.

Figure 24-172 shows a flow diagram about entering and leaving local power-down mode.

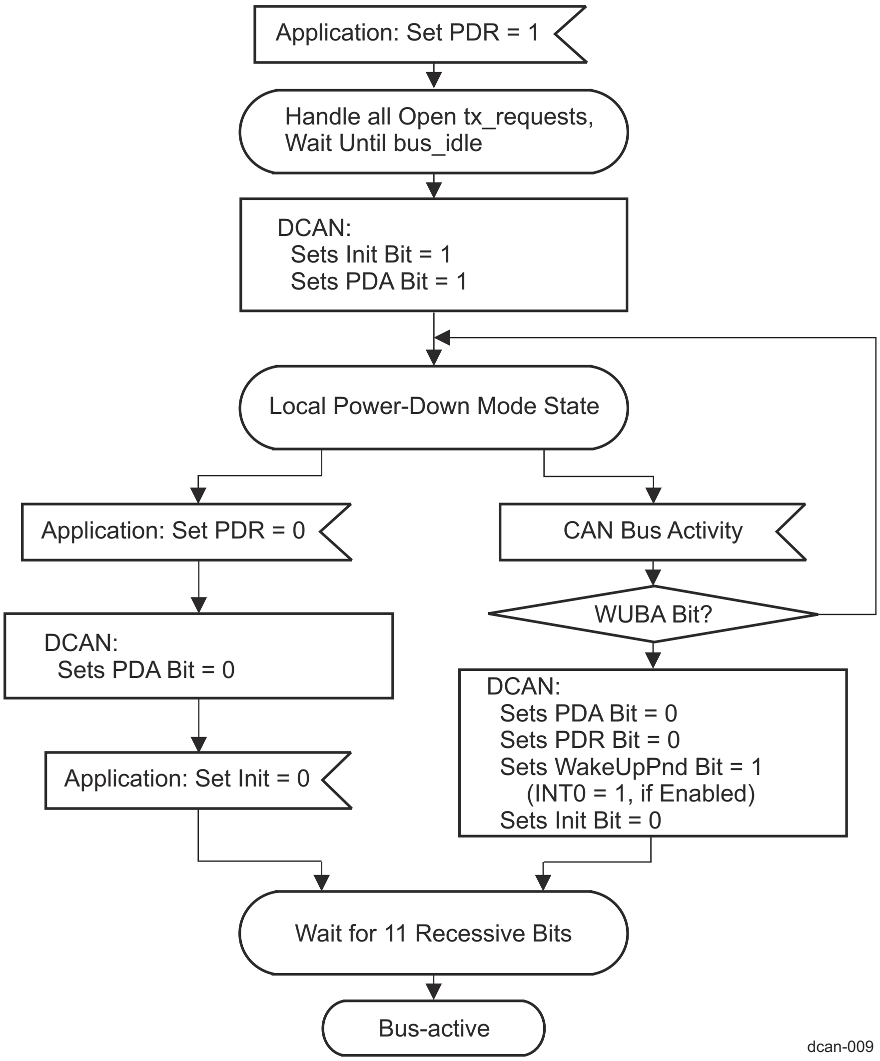


Figure 24-172. Local Power-Down Mode Flow Diagram

24.10.4.5 Parity Check Mechanism

The DCAN provides a parity check mechanism to ensure data integrity of message RAM data. For each word (32 bits) in message RAM, one parity bit will be calculated. The formation of the different words is according to the message RAM representation in RDA mode, see Section 24.10.4.11.4, *Message RAM Representation in Direct Access Mode*.

Parity information is stored in the message RAM on write accesses and will be checked against the stored parity bit from message RAM on read accesses.

The Parity check functionality can be enabled or disabled by DCAN_CTL[13:10] PMD bit field.

In case of disabled parity check, the parity bits in message RAM will be left unchanged on write access to data area and no check will be done on read access.

If parity checking is enabled, parity bits will be automatically generated and checked by the DCAN. The parity bits could be read in debug/suspend mode (see Section 24.10.4.11.3, *Message RAM Representation in Debug/Suspend Mode*) or in RDA mode (see Section 24.10.4.11.4, *Message RAM Representation in Direct Access Mode*). However, direct write access to the parity bits is only possible in this two modes with parity check disabled.

A parity bit will be set, if the modulo-2-sum of the data bits is 1. This definition is equivalent to: The parity bit will be set, if the number of 1 bits in the data is odd.

Note

DCAN is configured to even parity by the design.

24.10.4.5.1 Behavior on Parity Error

On any read access to message RAM (e.g., during start of a CAN frame transmission), the parity of the message object will be checked. If a parity error is detected, the DCAN_ES[8] PER bit will be set. If error interrupts are enabled, an interrupt would also be generated. In order to avoid the transmission of invalid data over the CAN bus, the MsgVal bit of the message object will be reset to 0.

The message object data can be read by the software, independently of parity errors. Thus, the software has to ensure that the read data is valid, for example, by immediately checking the parity error code register (DCAN_PERR) on parity error interrupt.

Note

During RAM initialization, no parity check is done, but if the PMD bit is set, the parity bits will be generated.

24.10.4.5.2 Parity Testing

Testing the parity mechanism can be done by enabling the bit RamDirectAccess (DCAN_TEST[9] RDA) and manually writing the parity bits directly to the dedicated RAM locations. With this, data and parity bits could be checked when reading directly from RAM.

Note

If parity check is disabled, the software has to ensure correct parity bit handling in order to prevent parity errors later on when parity check is enabled.

24.10.4.6 Debug/Suspend Mode

The module supports the usage of an external debug unit by providing functions like pausing DCAN activities and making message RAM content accessible via interconnect interface.

Before entering debug/suspend mode, the DCAN will either wait until a started transmission or reception will be finished and bus idle state is recognized, or immediately interrupt a current transmission or reception. This is depending on bit DCAN_CTL[8] IDS in the CAN control register.

Afterwards, the DCAN enters debug/suspend mode, indicated by DCAN_CTL [16] INITDBG flag.

During debug/suspend mode, all DCAN registers can be accessed. Reading reserved bits will return ‘0’. Writing to reserved bits will have no effect.

Also, the message RAM will be memory mapped. This allows the external debug unit to read the message RAM. For the memory organization, see Section 24.10.4.11.3, *Message RAM Representation in Debug/Suspend Mode*.

Note

During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

Note

Writing to control registers in debug/suspend mode may influence the CAN state machine and further message handling.

For debug support, the auto clear functionality of the following DCAN registers is disabled:

- DCAN_ES register (clear of status flags by read)
- DCAN_IF1CMD/DCAN_IF2CMD command registers (clear of [14] DMAACTIVE flag by read/write)

24.10.4.7 Configuration of Message Objects Description

The whole message RAM should be configured before the end of the initialization, however it is also possible to change the configuration of message objects during CAN communication.

The CAN software driver must offer subroutines that:

- Transfer a complete message structure into a message object. (Configuration)
- Transfer the data bytes of a message into a message object and set TxRqst and NewDat. (Start a new transmission)
- Get the data bytes of a message from a message object and clear NewDat (and IntPnd). (Read received data)
- Get the complete message from a message object and clear NewDat (and IntPnd). (Read a received message, including identifier, from a message object with UMask = ‘1’)

Parameters of the subroutines are the Message Number and a pointer to a complete message structure or to the data bytes of a message structure.

Two examples of assigning the IFx interface register sets to these subroutines are shown here:

In the first method, the tasks of the application program that may access the module are divided into two groups. Each group is restricted to the use of one of the interface register sets. The tasks of one group may interrupt tasks of the other group, but not of the same group.

In the second method, which may be a special case of the first method, there are only two tasks in the application program that access the module. A Read_Message task that uses IF2 or IF3 to get received messages from the message RAM and a Write_Message task that uses IF1 to write messages to be transmitted (or to be configured) into the message RAM. Both tasks may interrupt each other.

24.10.4.7.1 Configuration of a Transmit Object for Data Frames

Table 24-741 shows how a transmit object can be initialized.

Table 24-741. Initialization of a Transmit Object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of the outgoing message. If an 11-bit identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored.

The data length and data itself (DLC[3:0] and Data0-7) are given by the application. TxRqst and RmtEn should not be set before the data is valid.

If the TXIE bit is set, the IntPnd bit will be set after a successful transmission of the message object.

If the RmtEn bit is set, a matching received remote frame will cause the TxRqst bit to be set; the remote frame will autonomously be answered by a data frame.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask=’1’) to allow groups of remote frames with similar identifiers to set the TxRqst bit. The Dir bit should not be masked. For details, see Section 24.10.4.8.8, *Reception of Remote Frames*.

Identifier masking must be disabled (UMask = ‘0’) if no remote frames are allowed to set the TxRqst bit (RmtEn = ‘0’).

24.10.4.7.2 Configuration of a Transmit Object for Remote Frames

It is not necessary to configure transmit objects for the transmission of remote frames. Setting TxRqst for a receive object causes the transmission of a remote frame with the same identifier as the data rrame for which this receive object is configured.

24.10.4.7.3 Configuration of a Single Receive Object for Data Frames

Table 24-742 shows how a receive object for data frames can be initialized.

Table 24-742. Initialization of a single Receive Object for Data Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a data frame with an 11-bit Identifier is received, ID[17:0] is set to ‘0’.

The data length code (DLC[3:0]) is given by the application. When the message handler stores a data frame in the message object, it will store the received data length code and eight data bytes. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by non specified values.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask = ’1’) to allow groups of data frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. If some bits of the mask bits are set to “don’t care,” the corresponding bits of the arbitration register (DCAN_IF1ARB, DCAN_IF2ARB, DCAN_IF3ARB) will be overwritten by the bits of the stored data frame.

If the RxIE bit is set, the IntPnd bit will be set when a received data frame is accepted and stored in the message object.

If the TxRqst bit is set, the transmission of a remote frame with the same identifier as actually stored in the arbitration bits will be triggered. The content of the arbitration bits may change if the mask bits are used (UMask = ’1’) for acceptance filtering.

24.10.4.7.4 Configuration of a Single Receive Object for Remote Frames

Table 24-743 shows how a receive object for remote frames can be initialized.

Table 24-743. Initialization of a Single Receive Object for Remote Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	appl.	0	0	0	0

A receive object for remote frames may be used to monitor remote frames on the CAN bus. The remote frame stored in the receive object will not trigger the transmission of a data frame. Receive objects for remote frames may be expanded to a FIFO buffer (see Section 24.10.4.7.5, *Configuration of a FIFO Buffer*).

UMask must be set to ‘1.’ The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be set to “must-match” or to “don’t care,” to allow groups of remote frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. For details, see Section 24.10.4.8.8, *Reception of Remote Frames*.

The arbitration bits (ID[28:0] and Xtd bit) may be given by the application. They define the identifier and type of accepted received remote frames. If some bits of the mask bits are set to “don’t care,” the corresponding bits of the arbitration bits will be overwritten by the bits of the stored remote frame. If an 11-bit Identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a remote frame with an 11-bit Identifier is received, ID[17:0] will be set to ‘0.’

The data length code (DLC[3:0]) may be given by the application. When the message handler stores a remote frame in the message object, it will store the received data length code. The data bytes of the message object will remain unchanged.

If the RxIE bit is set, the IntPnd bit will be set when a received remote frame is accepted and stored in the message object.

24.10.4.7.5 Configuration of a FIFO Buffer

With the exception of the EoB bit, the configuration of receive objects belonging to a FIFO buffer is the same as the configuration of a single receive object.

To concatenate multiple message objects to a FIFO buffer, the identifiers and masks (if used) of these message objects have to be programmed to matching values. Due to the implicit priority of the message objects, the message object with the lowest number will be the first message object of the FIFO buffer. The EoB bit of all message objects of a FIFO buffer except the last one have to be programmed to zero. The EoB bits of the last message object of a FIFO Buffer is set to one, configuring it as the end of the block.

24.10.4.8 Message Handling

When initialization is finished, the DCAN module synchronizes itself to the traffic on the CAN bus. It does acceptance filtering on received messages and stores those frames that are accepted into the designated message objects. The application has to update the data of the messages to be transmitted and to enable and request their transmission. The transmission is requested automatically when a matching remote frame is received.

The application may read messages which are received and accepted. Messages that are not read before the next messages is accepted for the same message object will be overwritten.

Messages may be read interrupt-driven or after polling of NewDat.

24.10.4.8.1 Message Handler Overview

The message handler state machine controls the data transfer between the Rx/Tx shift register of the CAN core and the message RAM. It performs the following tasks:

- Data transfer from message RAM to CAN core (messages to be transmitted)
- Data transfer from CAN core to the message RAM (received messages)
- Data transfer from CAN core to the acceptance filtering unit
- Scanning of message RAM for a matching message object (acceptance filtering)
- Scanning the same message object after being changed by IF1/IF2 registers when priority is the same or higher as the message the object found by last scanning
- Handling of TxRqst flags
- Handling of interrupt flags

The message handler registers contains status flags of all message objects grouped into the following topics:

- Transmission Request Flags
- New Data Flags
- Interrupt Pending Flags
- Message Valid Registers

Instead of collecting above listed status information of each message object via IFx registers separately, these message handler registers provides a fast and easy way to get an overview, for example, about all pending transmission requests.

All message handler registers are read-only.

24.10.4.8.2 Receive/Transmit Priority

The receive/transmit priority for the message objects is attached to the message number, not to the CAN identifier. Message object 1 has the highest priority, while the last implemented message object has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding message object so messages with the highest priority, for example, can be placed in the message objects with the lowest numbers.

The acceptance filtering for received data frames or remote frames is also done in ascending order of message objects, so a frame that has been accepted by a message object cannot be accepted by another message object with a higher message number. The last message object may be configured to accept any data frame or remote frame that was not accepted by any other message object, for nodes that need to log the complete message traffic on the CAN bus.

24.10.4.8.3 Transmission of Messages in Event Driven CAN Communication

If the shift register of the CAN core is ready for loading and if there is no data transfer between the IFx registers and message RAM, the MSGVAL bits in the Message Valid register (DCAN_MSGVAL12 to DCAN_MSGVAL78) and the TXRQST bits in the transmission request register (DCAN_TXRQ12 to DCAN_TXRQ78) are evaluated. The valid message object with the highest priority pending transmission request is loaded into the shift register by the message handler and the transmission is started. The message object’s NewDat bit is reset.

After a successful transmission and if no new data was written to the message object (NewDat = ‘0’) since the start of the transmission, the TxRqst bit will be reset. If TxIE is set, IntPnd will be set after a successful transmission. If the DCAN has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. If meanwhile the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

If automatic retransmission mode is disabled by setting the DCAN_CTL[5] DAR bit, the behavior of bits TXRQST and NEWDAT in the DCAN_IF1CMD/DCAN_IF2CMD register of the interface register set is as follows:

- When a transmission starts, the TxRqst bit of the respective interface register set is reset, while bit NewDat remains set.
- When the transmission has been successfully completed, the NewDat bit is reset.

When a transmission failed (lost arbitration or error), bit NewDat remains set. To restart the transmission, the application has to set TxRqst again.

Received remote frames do not require a receive object. They will automatically trigger the transmission of a data frame, if in the matching transmit object the RmtEn bit is set.

24.10.4.8.4 Updating a Transmit Object

The software may update the data bytes of a transmit object any time via the IF1/IF2 interface registers, neither MSGVAL, nor TXRQST have to be reset before the update.

Even if only a part of the data bytes are to be updated, all four bytes in the corresponding IF1/IF2 Data A register (DCAN_IF1DATA/DCAN_IF2DATA) or IF1/IF2 Data B register (DCAN_IF1DATB/DCAN_IF2DATB) have to be valid before the content of that register is transferred to the message object. Either the software has to write all four bytes into the IF1/IF2 data register or the message object is transferred to the IF1/IF2 data register before the software writes the new data bytes.

When only the data bytes are updated, first 0x87 can be written to bits [23:16] of the IF1/IF2 Command register (DCAN_IF1CMD/DCAN_IF2CMD) and then the number of the message object is written to bits [7:0] MESSAGE_NUMBER of the command register, concurrently updating the data bytes and setting TXRQST with NEWDAT.

To prevent the reset of TxRqst at the end of a transmission that may already be in progress while the data is updated, NewDat has to be set together with TxRqst in event driven CAN communication. For details, see Section 24.10.4.8.3, *Transmission of Messages in Event Driven CAN Communication*.

When NewDat is set together with TxRqst, NewDat will be reset as soon as the new transmission has started.

24.10.4.8.5 Changing a Transmit Object

If the number of implemented message objects is not sufficient to be used as permanent message objects only, the transmit objects may be managed dynamically. The software can write the whole message (arbitration, control, and data) into the interface register. The bits [23:16] of the command register (DCAN_IF1CMD/ DCAN_IF2CMD) can be set to 0xB7 for the transfer of the whole message object content into the message object. Neither Dir, nor TxRqst have to be reset before this operation.

If a previously requested transmission of this message object is not completed but already in progress, it will be continued; however, it will not be repeated if it is disturbed.

To only update the data bytes of a message to be transmitted, bits [23:16] of the command register (DCAN_IF1CMD/ DCAN_IF2CMD) should be set to 0x87.

Note

After the update of the transmit object, the interface register set will contain a copy of the actual contents of the object, including the part that had not been updated.

24.10.4.8.6 Acceptance Filtering of Received Messages

When the arbitration and control bits (Identifier + IDE + RTR + DLC) of an incoming message are completely shifted into the shift register of the CAN core, the message handler starts the scan of the message RAM for a matching valid message object:

- The acceptance filtering unit is loaded with the arbitration bits from the CAN core shift register.
- Then the arbitration and mask bits (including MsgVal, UMask, NewDat, and EoB) of message object 1 are loaded into the acceptance filtering unit and are compared with the arbitration bits from the shift register. This is repeated for all following message objects until a matching message object is found, or until the end of the message RAM is reached.
- If a match occurs, the scanning is stopped and the message handler proceeds depending on the type of the frame (data frame or remote frame) received.

24.10.4.8.7 Reception of Data Frames

The message handler stores the message from the CAN core shift register into the respective message object in the message RAM. Not only the data bytes, but all arbitration bits and the data length code are stored into the corresponding message object. This ensures that the data bytes stay associated to the identifier even if arbitration mask registers are used.

The NewDat bit is set to indicate that new data (not yet seen by the software) has been received. The software should reset the NewDat bit when it reads the message object. If at the time of the reception the NewDat bit was already set, MsgLst is set to indicate that the previous data (supposedly not seen by the software) is lost. If the RxIE bit is set, the IntPnd bit is set, causing the DCAN_INT to point to this message object.

The TxRqst bit of this message object is reset to prevent the transmission of a remote frame, while the requested data frame has just been received.

24.10.4.8.8 Reception of Remote Frames

When a remote frame is received, three different configurations of the matching message object have to be considered:

- Dir = ‘1’ (direction = transmit), RmtEn = ‘1’, UMask = ‘1’ or ‘0’

  The TxRqst bit of this message object is set at the reception of a matching remote frame. The rest of the message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘0’

  The remote frame is ignored, this message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘1’

The remote frame is treated similar to a received data frame. At the reception of a matching Remote Frame, the TxRqst bit of this message object is reset. The arbitration and control bits (Identifier + IDE + RTR + DLC) from the shift register are stored in the message object in the message RAM and the NewDat bit of this message object is set. The data bytes of the message object remain unchanged.

24.10.4.8.9 Reading Received Messages

The software may read a received message any time via the IFx interface register. The data consistency is guaranteed by the message handler state machine. Typically the software will write first 0x7F to bits [23:16] and then the number of the message object to bits [7:0] MESSAGE_NUMBER of the command register (DCAN_IF1CMD/ DCAN_IF2CMD). That combination will transfer the entire received message from the message RAM into the interface register set. Additionally, the bits NewDat and IntPnd are cleared in the message RAM (not in the interface register set). The values of these bits in the message control register (DCAN_IF1MCTL/DCAN_IF2MCTL/DCAN_IF3MCTL) always reflect the status before resetting the bits. If the message object uses masks for acceptance filtering, the arbitration bits show which of the different matching messages has been received.

The actual value of NewDat shows whether a new message has been received since last time when this message object was read. The actual value of MsgLst shows whether more than one message has been received since the last time when this message object was read. MsgLst will not be automatically reset.

24.10.4.8.10 Requesting New Data for a Receive Object

By means of a remote frame, the software may request another CAN node to provide new data for a receive object. Setting the TxRqst bit of a receive object will cause the transmission of a remote frame with the identifier of the receive object. This remote frame triggers the other CAN node to start the transmission of the matching data frame. If the matching data frame is received before the remote frame could be transmitted, the TxRqst bit is automatically reset. Setting the TxRqst bit without changing the contents of a message object requires the value 0x84 in bits [23:16] of the command register (DCAN_IF1CMD/ DCAN_IF2CMD).

24.10.4.8.11 Storing Received Messages in FIFO Buffers

Several message objects may be grouped to form one or more FIFO buffers. Each FIFO buffer configured to store received messages with a particular (group of) identifier(s). arbitration and mask registers of the FIFO buffer’s message objects are identical. The end of buffer (EoB) bits of all but the last of the FIFO buffer’s message objects are ‘0’; in the last one the EoB bit is ‘1.’

Received messages with identifiers matching to a FIFO buffer are stored into a message object of this FIFO buffer, starting with the message object with the lowest message number. When a message is stored into a message object of a FIFO buffer, the NewDat bit of this message object is set. By setting NewDat while EoB is ‘0’, the message object is locked for further write accesses by the message handler until the software has cleared the NewDat bit.

Messages are stored into a FIFO buffer until the last message object of this FIFO buffer is reached. If none of the preceding message objects is released by writing NewDat to ‘0,’ all further messages for this FIFO buffer will be written into the last message object of the FIFO buffer (EoB = ‘1’) and therefore overwrite previous messages in this message object.

24.10.4.8.12 Reading From a FIFO Buffer

Several messages may be accumulated in a set of message objects which are concatenated to form a FIFO buffer before the application program is required (in order to avoid the loss of data) to empty the buffer.

A FIFO buffer of length N will store N–1, plus the last received message since last time it was cleared.

A FIFO buffer is cleared by reading and resetting the NewDat bits of all its message objects, starting at the FIFO Object with the lowest message number. This should be done in a subroutine following the example shown in Figure 24-173.

Note

All message objects of a FIFO buffer needs to be read and cleared before the next batch of messages can be stored. Otherwise, true FIFO functionality can not be guaranteed, since the message objects of a partly read buffer will be re-filled according to the normal (descending) priority.

Reading from a FIFO buffer message object and resetting its NewDat bit is handled the same way as reading from a single message object.

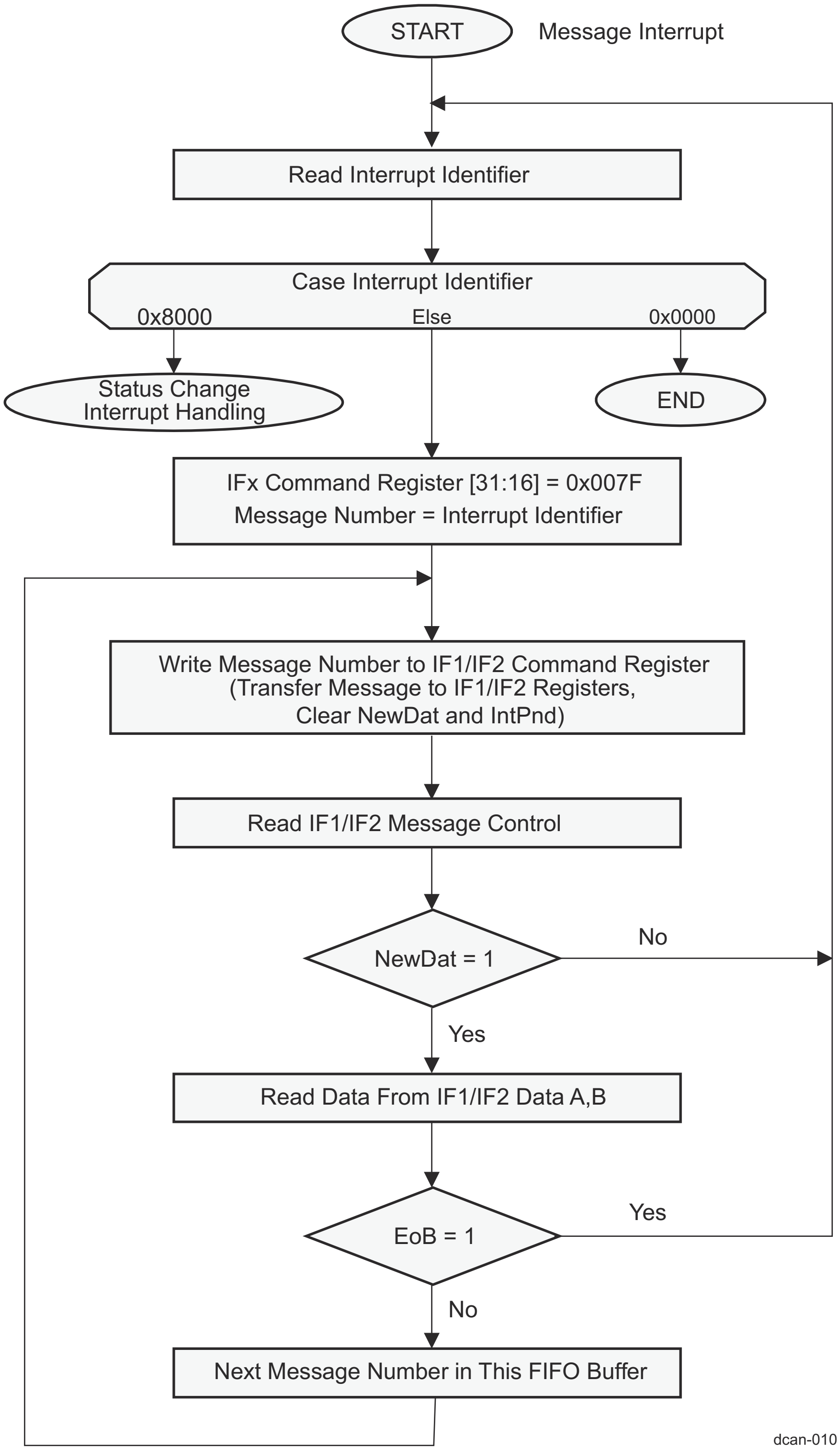


Figure 24-173. Software Handling of a FIFO Buffer (Interrupt Driven)

24.10.4.9 CAN Bit Timing

The DCAN supports bit rates between < 1 kBit/s and 1000 kBit/s.

Each member of the CAN network has its own clock generator, typically derived from a crystal oscillator. The bit timing parameters can be configured individually for each CAN node, creating a common bit rate even though the CAN nodes' oscillator periods (f_{osc}) may be different.

The frequencies of these oscillators are not absolutely stable. Small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range (df), the CAN nodes are able to compensate for the different bit rates by resynchronizing to the bit stream.

In many cases, the CAN bit synchronization will amend a faulty configuration of the CAN bit timing to such a degree that only occasionally an error frame is generated. In the case of arbitration however, when two or more CAN nodes simultaneously try to transmit a frame, a misplaced sample point may cause one of the transmitters to become error passive.

The analysis of such sporadic errors requires a detailed knowledge of the CAN bit synchronization inside a CAN node and of the CAN nodes' interaction on the CAN bus.

Even if minor errors in the configuration of the CAN bit timing do not result in immediate failure, the performance of a CAN network can be reduced significantly.

24.10.4.9.1 Bit Time and Bit Rate

According to the CAN specification, the bit time is divided into four segments (see Figure 24-174):

- Synchronization segment (Sync_Seg)
- Propagation time segment (Prop_Seg)
- Phase buffer segment 1 (Phase_Seg1)
- Phase buffer segment 2 (Phase_Seg2)

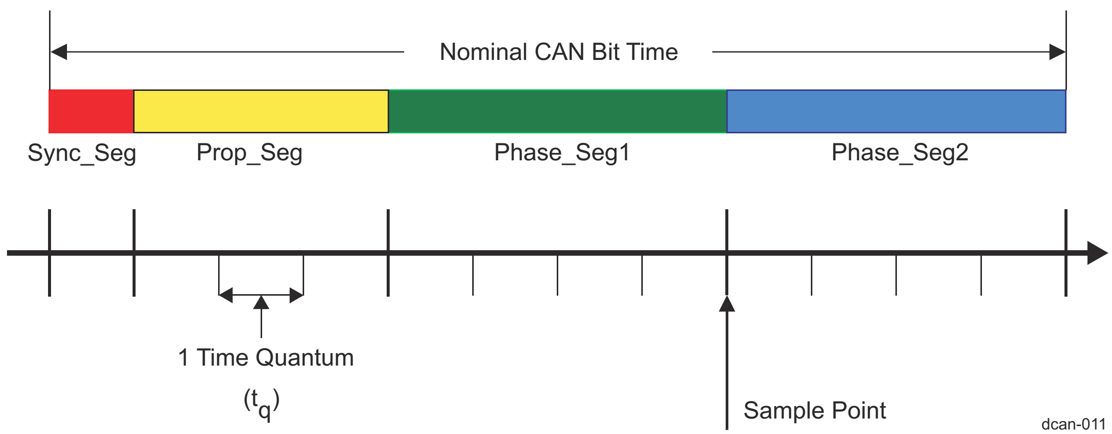


Figure 24-174. Bit Timing

Each segment consists of a specific number of time quanta. The length of one time quantum (t_q), which is the basic time unit of the bit time, is given by the FCLK and the baud rate prescalers (BRPE and BRP). With these two baud rate prescalers combined, divider values from 1 to 1024 can be programmed:

t_q = Baud Rate Prescaler / FCLK

Apart from the fixed length of the synchronization segment, these numbers are programmable. Table 24-744 describes the minimum programmable ranges required by the CAN protocol.

A given bit rate may be met by different bit time configurations.

Table 24-744. Parameters of the CAN Bit Time

Parameter	Range	Remark
Sync_Seg	1 t_q (fixed)	Synchronization of bus input to FCLK
Prop_Seg	[1 … 8] t_q	Compensates for the physical delay times

Table 24-744. Parameters of the CAN Bit Time (continued)

Parameter	Range	Remark
Phase_Seg1	[1 … 8] t_q	May be lengthened temporarily by synchronization
Phase_Seg2	[1 … 8] t_q	May be shortened temporarily by synchronization
Synchronization Jump Width (SJW)	[1 … 4] t_q	May not be longer than either Phase Buffer Segment

Note

For proper functionality of the CAN network, the physical delay times and the oscillator’s tolerance range have to be considered.

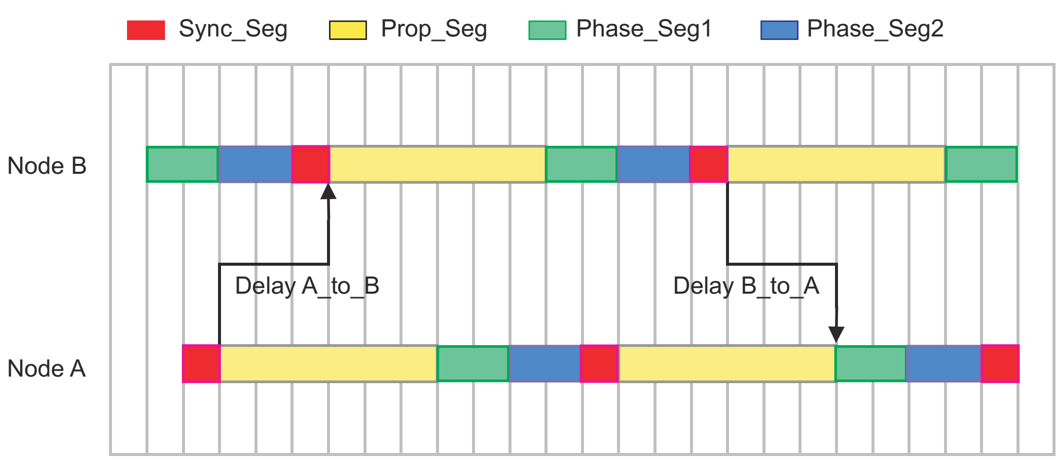
24.10.4.9.1.1 Synchronization Segment

The synchronization segment (Sync_Seg) is the part of the bit time where edges of the CAN bus level are expected to occur. If an edge occurs outside of Sync_Seg, its distance to the Sync_Seg is called the phase error of this edge.

24.10.4.9.1.2 Propagation Time Segment

This part of the bit time is used to compensate physical delay times within the CAN network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

Any CAN node synchronized to the bit stream on the CAN bus can be out of phase with the transmitter of the bit stream, caused by the signal propagation time between the two nodes. The CAN protocol’s nondestructive bitwise arbitration and the dominant acknowledge bit provided by receivers of CAN messages require that a CAN node transmitting a bit stream must also be able to receive dominant bits transmitted by other CAN nodes that are synchronized to that bit stream. The example in Figure 24-175 shows the phase shift and propagation times between two CAN nodes.



Delay A_to_B ≥ node output delay(A) + bus line delay(AÆB) + node input delay(B)

Prop_Seg ≥ Delay A_to_B + Delay B_to_A

Prop_Seg ≥ 2 [max(node output delay + bus line delay + node input delay)]

dcan-012

Figure 24-175. The Propagation Time Segment

In this example, both nodes A and B are transmitters performing an arbitration for the CAN bus. The node A has sent its start of frame bit less than one bit time earlier than node B, therefore node B has synchronized itself to the received edge from recessive to dominant. Since node B has received this edge delay(A_to_B) after it has been transmitted, node B’s bit timing segments are shifted with regard to node A. Node B sends an identifier with higher priority and so it will win the arbitration at a specific identifier bit when it transmits a dominant bit while node A transmits a recessive bit. The dominant bit transmitted by node B will arrive at node A after the delay (B_to_A).

Due to oscillator tolerances, the actual position of node A's sample point can be anywhere inside the nominal range of node A's Phase Buffer Segments, so the bit transmitted by node B must arrive at node A before the start of Phase_Seg1. This condition defines the length of Prop_Seg.

If the edge from recessive to dominant transmitted by node B would arrive at node A after the start of Phase_Seg1, it could happen that node A samples a recessive bit instead of a dominant bit, resulting in a bit error and the destruction of the current frame by an error flag.

This error only occurs when two nodes arbitrate for the CAN bus which have oscillators of opposite ends of the tolerance range and are separated by a long bus line; this is an example of a minor error in the bit timing configuration (Prop_Seg too short) that causes sporadic bus errors.

Some CAN implementations provide an optional 3-Sample Mode. The DCAN does not. In this mode, the CAN bus input signal passes a digital low-pass filter, using three samples and a majority logic to determine the valid bit value. This results in an additional input delay of 1 t_q, requiring a longer Prop_Seg.

24.10.4.9.1.3 Phase Buffer Segments and Synchronization

The phase buffer segments (Phase_Seg1 and Phase_Seg2) and the synchronization jump width (SJW) are used to compensate for the oscillator tolerance.

The phase buffer segments surround the sample point and may be lengthened or shortened by synchronization.

The synchronization jump width (SJW) defines how far the resynchronizing mechanism may move the sample point inside the limits defined by the phase buffer segments to compensate for edge phase errors.

Synchronizations occur on edges from recessive to dominant. Their purpose is to control the distance between edges and sample points.

Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous sample point. A synchronization may be done only if a recessive bit was sampled at the previous sample point and if the actual time quantum's bus level is dominant.

An edge is synchronous if it occurs inside of Sync_Seg; otherwise, its distance to the Sync_Seg is the edge phase error, measured in time quanta. If the edge occurs before Sync_Seg, the phase error is negative, else it is positive.

Two types of synchronization exist: hard synchronization and resynchronizing. A hard synchronization is done once at the start of a frame; inside a frame, only resynchronization is possible.

- Hard Synchronization

  After a hard synchronization, the bit time is restarted with the end of Sync_Seg, regardless of the edge phase error. Thus hard synchronization forces the edge which has caused the hard synchronization, to lie within the synchronization segment of the restarted bit time.
- Bit Resynchronizations

  Resynchronization leads to a shortening or lengthening of the Bit time such that the position of the sample point is shifted with regard to the edge.

  When the phase error of the edge which causes resynchronization is positive, Phase_Seg1 is lengthened. If the magnitude of the phase error is less than SJW, Phase_Seg1 is lengthened by the magnitude of the phase error, else it is lengthened by SJW.

  When the phase error of the edge which causes Resynchronization is negative, Phase_Seg2 is shortened. If the magnitude of the phase error is less than SJW, Phase_Seg2 is shortened by the magnitude of the phase error, else it is shortened by SJW.

If the magnitude of the phase error of the edge is less than or equal to the programmed value of SJW, the results of hard synchronization and resynchronization are the same. If the magnitude of the phase error is larger than

SJW, the resynchronization cannot compensate the phase error completely, and an error of (phase error - SJW) remains.

Only one synchronization may be done between two sample points. The synchronizations maintain a minimum distance between edges and sample points, giving the bus level time to stabilize and filtering out spikes that are shorter than (Prop_Seg + Phase_Seg1).

Apart from noise spikes, most synchronizations are caused by arbitration. All nodes synchronize “hard” on the edge transmitted by the “leading” transceiver that started transmitting first, but due to propagation delay times, they cannot become ideally synchronized. The leading transmitter does not necessarily win the arbitration; therefore, the receivers have to synchronize themselves to different transmitters that subsequently take the lead and that are differently synchronized to the previously leading transmitter. The same happens at the acknowledge field, where the transmitter and some of the receivers will have to synchronize to that receiver that takes the lead in the transmission of the dominant acknowledge bit.

Synchronizations after the end of the arbitration will be caused by oscillator tolerance, when the differences in the oscillator’s clock periods of transmitter and receivers sum up during the time between synchronizations (at most ten bits). These summarized differences may not be longer than the SJW, limiting the oscillator’s tolerance range.

Figure 24-176 shows how the phase buffer segments are used to compensate for phase errors. There are three drawings of each two consecutive bit timings. The upper drawing shows the synchronization on a “late” edge, the lower drawing shows the synchronization on an “early” edge, and the middle drawing is the reference without synchronization.

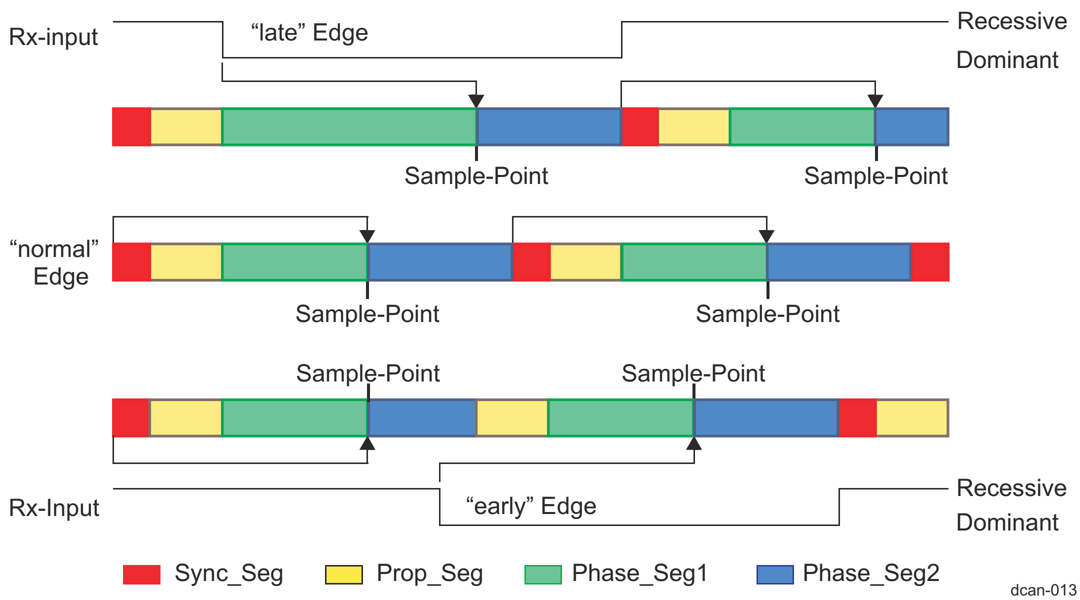


Figure 24-176. Synchronization on Late and Early Edges

In the first example, an edge from recessive to dominant occurs at the end of Prop_Seg. The edge is "late" since it occurs after the Sync_Seg. Reacting to the late edge, Phase_Seg1 is lengthened so that the distance from the edge to the sample point is the same as it would have been from the Sync_Seg to the sample point if no edge had occurred. The phase error of this late edge is less than SJW, so it is fully compensated and the edge from dominant to recessive at the end of the bit, which is one nominal bit time long, occurs in the Sync_Seg.

In the second example, an edge from recessive to dominant occurs during Phase_Seg2. The edge is "early" since it occurs before a Sync_Seg. Reacting to the early edge, Phase_Seg2 is shortened and Sync_Seg is omitted, so that the distance from the edge to the sample point is the same as it would have been from a Sync_Seg to the sample point if no edge had occurred. As in the previous example, the magnitude of this early edge’s phase error is less than SJW, so it is fully compensated.

The phase buffer segments are lengthened or shortened temporarily only; at the next bit time, the segments return to their nominal programmed values.

In these examples, the bit timing is seen from the point of view of the CAN implementation's state machine, where the bit time starts and ends at the sample points. The state machine omits Sync_Seg when synchronizing on an early edge because it cannot subsequently redefine that time quantum of Phase_Seg2 where the edge occurs to be the Sync_Seg.

Figure 24-177 shows how short dominant noise spikes are filtered by synchronizations. In both examples, the spike starts at the end of Prop_Seg and has the length of (Prop_Seg + Phase_Seg1).

In the first example, the synchronization jump width is greater than or equal to the phase error of the spike's edge from recessive to dominant. Therefore the sample point is shifted after the end of the spike; a recessive bus level is sampled.

In the second example, SJW is shorter than the phase error, so the sample point cannot be shifted far enough; the dominant spike is sampled as actual bus level.

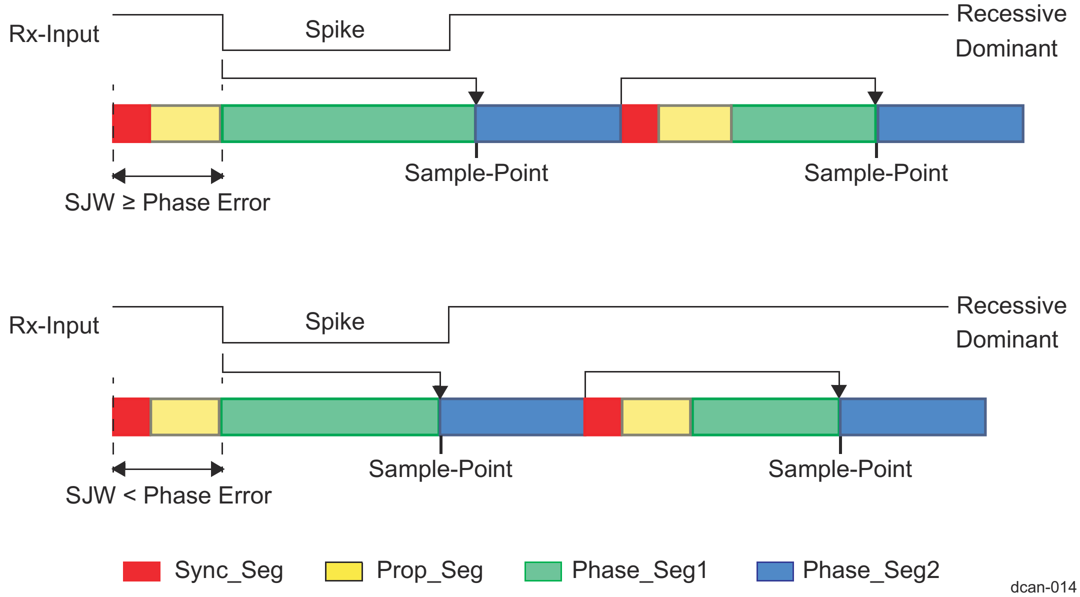


Figure 24-177. Filtering of Short Dominant Spikes

24.10.4.9.1.4 Oscillator Tolerance Range

With the introduction of CAN protocol version 1.2, the option to synchronize on edges from dominant to recessive became obsolete. Only edges from recessive to dominant are considered for synchronization. The protocol update to version 2.0 (A and B) had no influence on the oscillator tolerance.

The tolerance range df for an oscillator's frequency f_{osc} around the nominal frequency f_{nom} with:

$$(1 - df) \bullet f_{nom} \le f_{osc} \le (1 + df) \bullet f_{nom}$$

dcan-e015

depends on the proportions of Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df is the defined by two conditions (both shall be met):

$$\text{I:} \quad df \le \frac{\min(TSeg1, TSeg2)}{2x(13x(bit\_time - TSeg2))}$$

$$\text{II:} \quad df \le \frac{SJW}{20x bit\_time}$$

dcan-e016

It has to be considered that SJW may not be larger than the smaller of the phase buffer segments and that the propagation time segment limits that part of the bit time that may be used for the phase buffer segments.

The combination Prop_Seg = 1 and Phase_Seg1 = Phase_Seg2 = SJW = 4 allows the largest possible oscillator tolerance of 1.58%. This combination with a propagation time segment of only 10% of the bit time is not suitable for short bit times; it can be used for bit rates of up to 125 kBit/s (bit time = 8 µs) with a bus length of 40 m.

24.10.4.9.2 DCAN Bit Timing Registers

In the DCAN, the bit timing configuration is programmed in DCAN_BTR[14:0], additionally a baud rate prescaler extension of four bits (DCAN_BTR[19:16] BRPE) is provided.

- The sum of Prop_Seg and Phase_Seg1 is set in [11:8] TSEG1
- Phase_Seg2 in [14:12] TSEG2
- SynchronizationJumpWidth in [7:6] SJW
- and baud rate prescaler [5:0] BRP (plus [19:16] BRPE)

Figure 24-178 shows the CAN protocol controller.

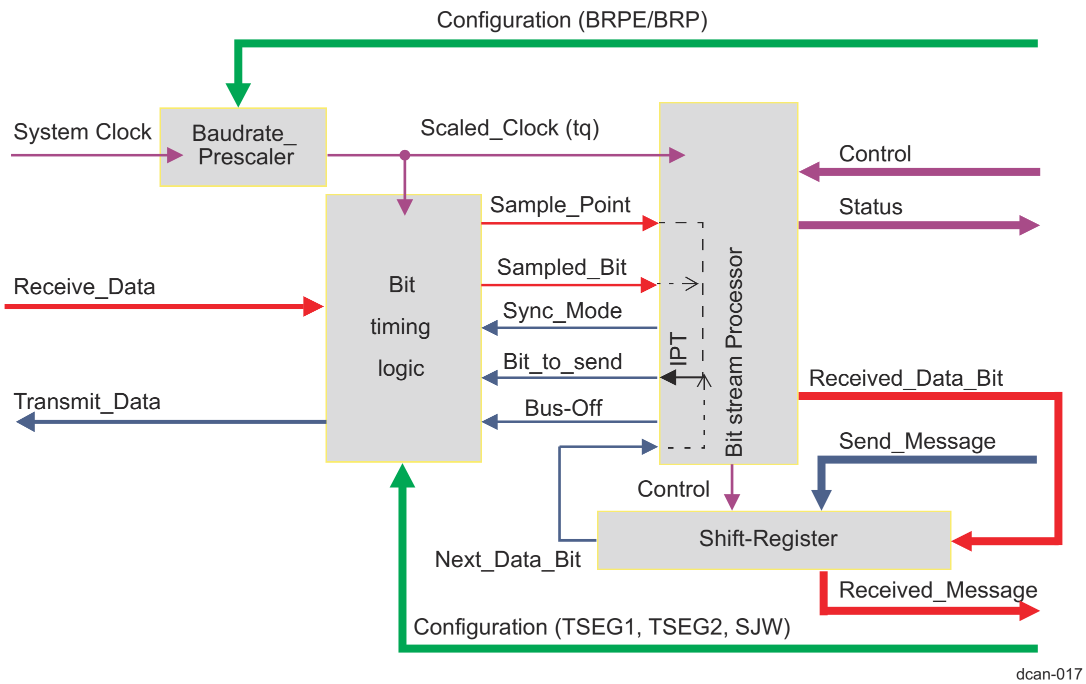


Figure 24-178. Structure of the CAN Core's CAN Protocol Controller

In DCAN_BTR register, the components TSEG1, TSEG2, SJW and BRP have to be programmed to a numerical value that is one less than its functional value; so instead of values in the range of [1…n], values in the range of [0…n–1] are programmed. That way, e.g., SJW (functional range of [1…4]) is represented by only two bits.

Therefore, the length of the bit time is (programmed values) [TSEG1 + TSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The data in the bit timing register (DCAN_BTR) is the configuration input of the CAN protocol controller. The baud rate prescaler (configured by BRPE/BRP) defines the length of the time quantum (the basic time unit of the bit time); the bit timing logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

The processing of the bit time, the calculation of the position of the sample point, and occasional synchronizations are controlled by the bit timing state machine, which is evaluated once each time quantum. The rest of the CAN protocol controller, the bit stream processor (BSP) state machine, is evaluated once each bit time, at the sample point.

The shift register serializes the messages to be sent and parallelizes received messages. Its loading and shifting is controlled by the BSP. The BSP translates messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the CRC code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the sample point and processes the sampled bus input bit. The time after the sample point that is needed to calculate the next bit to be sent (e.g., data bit, CRC bit, stuff bit, error flag, or idle) is called the information processing time (IPT), which is 0 tq for the DCAN.

Generally, the IPT is CAN controller-specific, but may not be longer than 2 tq. The IPC length is the lower limit of the programmed length of Phase_Seg2. In case of a synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

24.10.4.9.2.1 Calculation of the Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1 / Bit rate) must be an integer multiple of the CAN clock period.

The bit time may consist of 8 to 25 time quanta. The length of the time quantum t_q is defined by the baud rate prescaler with t_q = (Baud Rate Prescaler) / FCLK. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

First part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded up to the nearest integer multiple of t_q).

The Sync_Seg is 1 t_q long (fixed), leaving (bit time – Prop_Seg – 1) t_q for the two Phase Buffer Segments. If the number of remaining t_q is even, the Phase Buffer Segments have the same length, Phase_Seg2 = Phase_Seg1, else Phase_Seg2 = Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than any CAN controller's Information Processing Time in the network, which is device dependent and can be in the range of [0…2] t_q.

The length of the synchronization jump width is set to its maximum value, which is the minimum of four (4) and Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formulas given in Table 24-745.

If more than one configurations are possible to reach a certain Bit rate, it is recommended to choose the configuration which allows the highest oscillator tolerance range.

CAN nodes with different clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The CAN system's oscillator tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the oscillator frequencies' stability has to be increased in order to find a protocol compliant configuration of the CAN bit timing.

The resulting configuration is written into the bit timing register (DCAN_BTR):

[14:12] TSEG2 = Phase_Seg2 - 1

[11:8] TSEG1 = Phase_Seg1+ Prop_Seg - 1

[7:6] SJW = SynchronizationJumpWidth - 1

[5:0] BRP = Prescaler - 1

24.10.4.9.2.2 Example for Bit Timing Calculation

In this example, the frequency of FCLK is 20 MHz, BRP is 2, the bit rate is 500 KBit/s.

Table 24-745. Example For Bit Timing

Parameter	Formula	Value	t_q
bit time (500 KBit/s)	1/bit rate, consists of $t_{Sync\text{-}Seg} + t_{TSeg1} + t_{TSeg2}$	2000 ns	20
delay of bus driver		280 ns	-
delay of receiver circuit		29 ns	-
delay of bus line (16 m)	16 × 5.5 ns/m	88 ns	-
t_q	BRP/FCLK	100 ns	1
t_{Sync_Seg}	1 × t_q (fixed)	100 ns	1
t_{Prop_Seg}	INT (2×delays + 1) = 8 × t_q	800 ns	8
t_{Seg1}	$t_{Prop\_Seg} + t_{Phase\_Seg1}$	1400 ns	14
t_{Seg2}	bit time - ($_{Sync\_Seg}$ + t_{Seg1})	500 ns	5
t_{SJWmax}	MIN (4 × t_q, t_{Phase_Seg1})	400 ns	4

In this example, the bit timing register DCAN_BTR is programmed to:

- BRP = 2 - 1 = 1
- BRPE = 0
- TSEG1 = 14 - 1 = 13 (0xC)
- TSEG2 = 5 - 1 = 4
- SJW = 4 - 1 = 3

24.10.4.10 Message Interface Register Sets

The interface register sets control the software read and write accesses to the message RAM. There are two interface registers sets for read/write access, IF1 and IF2 and one interface register set for read access only, IF3.

Due to the structure of the message RAM, it is not possible to change single bits or bytes of a message object. Instead, always a complete message object in the message RAM is accessed. Therefore the data transfer from the IF1/IF2 registers to the message RAM requires the message handler to perform a read-modify-write cycle: First those parts of the message object that are not to be changed are read from the message RAM into the interface register set, and after the update the whole content of the interface register set is written into the message object.

After the partial write of a message object, those parts of the interface register set that are not selected in DCAN_IF1CMD/DCAN_IF2CMD, will be set to the actual contents of the selected message object.

After the partial read of a message object, those parts of the interface register set that are not selected in DCAN_IF1CMD/DCAN_IF2CMD, will be left unchanged.

By buffering the data to be transferred, the interface register sets avoid conflicts between concurrent software accesses to the message RAM and CAN message reception and transmission. A complete message object (see Section 24.10.4.11.1, *Structure of Message Objects*) or parts of the message object may be transferred between the message RAM and the IF1/IF2 register set (see Section 24.10.5, *DCAN Register Manual*) in one single transfer. This transfer, performed in parallel on all selected parts of the message object, guarantees the data consistency of the CAN message.

24.10.4.10.1 Message Interface Register Sets 1 and 2

The IF1 and IF2 register sets control the data transfer to and from the message object. DCAN_IF1CMD/ DCAN_IF2CMD address the desired message object in the message RAM and specifies whether a complete message object or only parts should be transferred. The data transfer is initiated by writing the message number to the bits [7:0] MESSAGE_NUMBER.

When the software initiates a data transfer between the IF1/IF2 registers and message RAM, the message handler sets the [15] BUSY bit in respective DCAN_IF1CMD/DCAN_IF2CMD to 1. After the transfer has completed, the BUSY bit is set back to 0 (see Figure 24-179).

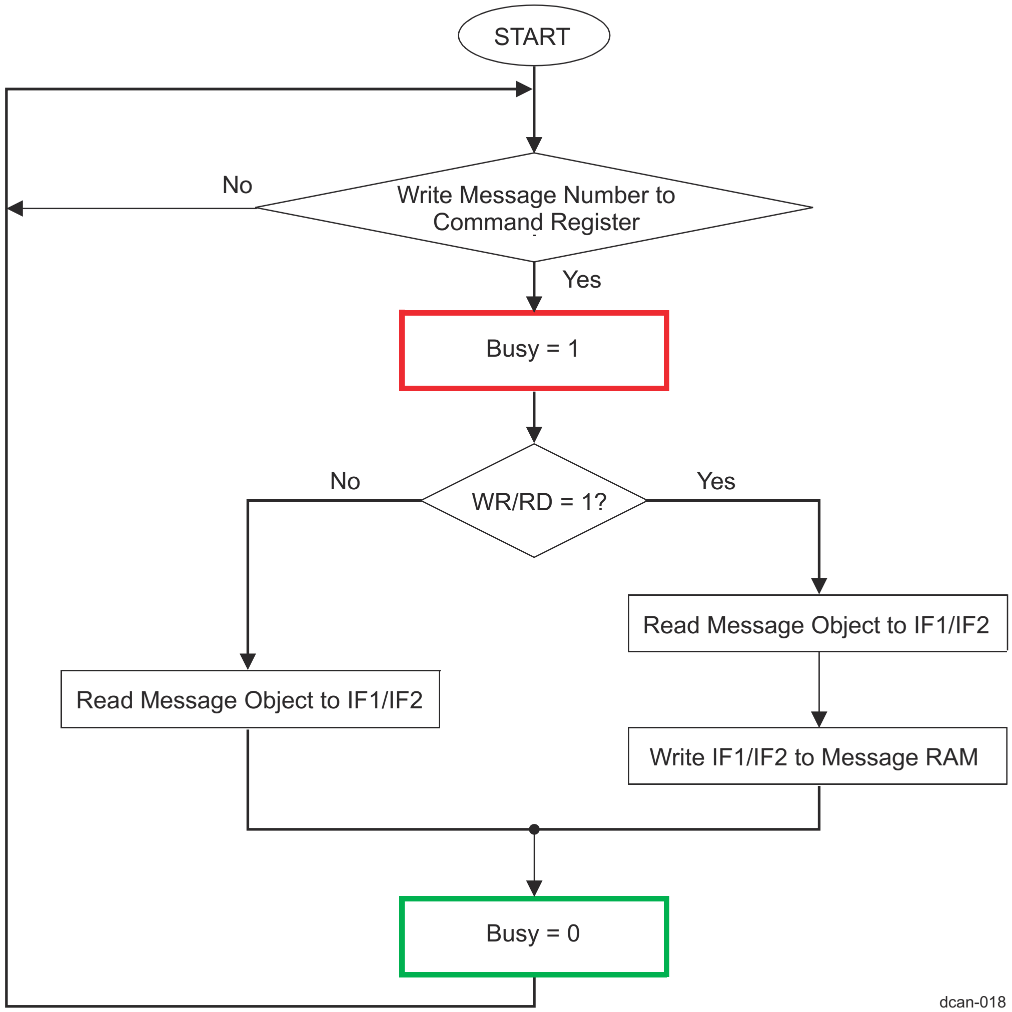


Figure 24-179. Data Transfer Between IF1/IF2 Registers and Message RAM

24.10.4.10.2 IF3 Register Set

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by software. The intention of this feature of IF3 is to provide an interface for the DMA to read packets efficiently. The automatic update functionality can be programmed for each message object (DCAN_IF3UPD12 to DCAN_IF3UPD78).

All valid message objects in message RAM which are configured for automatic update, will be checked for active NewDat flags. If such a message object is found, it will be transferred to the IF3 register (if no previous DMA transfers are ongoing), controlled by IF3 Observation register (DCAN_IF3OBS). If more than one NewDat flag is active, the message object with the lowest number has the highest priority for automatic IF3 update.

The NewDat bit in the message object will be reset by a transfer to IF3.

If DCAN internal IF3 update is complete, a DMA request is generated. The DMA request stays active until first read access to one of the IF3 registers. The DMA functionality has to be enabled by setting bit DCAN_CTL[20] DE3 = 1.

Note

The IF3 register set cannot be used for transferring data into message objects.

24.10.4.11 Message RAM

The DCAN message RAM contains message objects and parity bits for the message objects. There are up to 64 message objects in the message RAM.

During normal operation, accesses to the message RAM are performed via the interface register sets, and the software cannot directly access the message RAM.

The interface register sets IF1 and IF2 provide indirect read/write access from the software to the message RAM. The IF1 and IF2 register sets can buffer control and user data to be transferred to and from the message objects.

The third interface register set IF3 can be configured to automatically receive control and user data from the message RAM when a message object has been updated after reception of a CAN message. The software does not need to initiate the transfer from message RAM to IF3 register set.

The message handler avoids potential conflicts between concurrent accesses to message RAM and CAN frame reception/transmission.

There are two modes where the message RAM can be directly accessed by the software:

- Debug/Suspend mode (see Section 24.10.4.11.3, *Message RAM Representation in Debug/Suspend Mode*)
- RAM Direct Access (RDA) mode (see Section 24.10.4.11.4, *Message RAM Representation in Direct Access Mode*)

CAUTION

Writes to the DCAN RAM must not be done while it is in low-power mode. If such writes occur, the behavior is undefined and RAM content is corrupted. It has to be ensured in software that the low-power mode is removed from the DCAN RAM before writing to it.

24.10.4.11.1 Structure of Message Objects

Table 24-746 shows the structure of a message object.

The highlighted fields are those parts of the message object which are represented in dedicated registers. For example, the transmit request flags of all message objects are represented in centralized transmit request registers.

Table 24-746. Structure of a Message Object

UMask	Msk[28:0]	MXtd	MDir	EoB	unused	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID[28:0]	Xtd	Dir	DLC[3:0]	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7

Table 24-747. Message Object Field Descriptions

Field Name	Value	Description
MsgVal		Message valid
	0	The message object is ignored by the message handler.
	1	The message object is to be used by the message handler.
		Note: This bit may be kept at level ‘1’ even when the identifier bits ID[28:0], the control bits Xtd, Dir, or the data length code are changed. It should be reset if the Messages Object is no longer required. The software must reset the MsgVal bit of all unused Messages Objects during the initialization before it resets the INIT bit in the DCAN_CTL register.

Table 24-747. Message Object Field Descriptions (continued)

Field Name	Value	Description
UMask		Use acceptance mask
	0	Mask bits (Msk[28:0], MXtd and MDir) are ignored and not used for acceptance filtering.
	1	Mask bits are used for acceptance filtering.
		Note: If the UMask bit is set to one, the message object’s mask bits have to be programmed during initialization of the message object before MsgVal is set to one.
ID[28:0]		Message identifier
	ID[28:0]	29-bit (“extended”) identifier bits
	ID[28:18]	11-bit (“standard”) identifier bits
Msk[28:0]		Identifier mask
	0	The corresponding bit in the message identifier is not used for acceptance filtering (don’t care).
	1	The corresponding bit in the message identifier is used for acceptance filtering.
Xtd		Mask extended identifier
	0	The extended identifier bit (IDE) has no effect on the acceptance filtering.
	1	The extended identifier bit (IDE) is used for acceptance filtering.
		Note: When 11-bit (“standard”) Identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.
Dir		Message direction
	0	Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, the message is stored in this message object.
	1	Direction = transmit: On TxRqst, a data frame is transmitted. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = one).
MDir		Mask message direction
	0	The message direction bit (Dir) has no effect on the acceptance filtering.
	1	The message direction bit (Dir) is used for acceptance filtering.
EOB		End of block
	0	The message object is part of a FIFO Buffer block and is not the last message object of this FIFO Buffer block.
	1	The message object is a single message object or the last message object in a FIFO Buffer Block.
		Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.
NewDat		New data
	0	No new data has been written into the data bytes of this message object by the message handler since the last time when this flag was cleared by the software.
	1	The message handler or the software has written new data into the data bytes of this message object.
MsgLst		Message lost (only valid for message objects with direction = receive)
	0	No message was lost since the last time when this bit was reset by the software.
	1	The message handler stored a new message into this message object when NewDat was still set, so the previous message has been overwritten.
RxIE		Receive interrupt enable
	0	IntPnd will not be triggered after the successful reception of a frame.
	1	IntPnd will be triggered after the successful reception of a frame.
TxIE		Transmit interrupt enable
	0	IntPnd will not be triggered after the successful transmission of a frame.
	1	IntPnd will be triggered after the successful transmission of a frame.
IntPnd		Interrupt pending
	0	This message object is not the source of an interrupt.
	1	This message object is the source of an interrupt. The interrupt Identifier in the interrupt register (DCAN_INT) will point to this message object if there is no other interrupt source with higher priority.

Table 24-747. Message Object Field Descriptions (continued)

Field Name	Value	Description
RmtEn		Remote enable
	0	At the reception of a remote frame, TxRqst is not changed.
	1	At the reception of a remote frame, TxRqst is set.
TxRqst		Transmit request
	0	This message object is not waiting for a transmission.
	1	The transmission of this message object is requested and is not yet done.
DLC[3:0]		Data length code
	0 - 8	Data frame has 0 - 8 data bytes.
	9 - 15	Data frame has 8 data bytes.
		Note: The data length code of a message object must be defined to the same value as in the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
Data 0		1st data byte of a CAN data frame
Data 1		2nd data byte of a CAN data frame
Data 2		3rd data byte of a CAN data frame
Data 3		4th data byte of a CAN data frame
Data 4		5th data byte of a CAN data frame
Data 5		6th data byte of a CAN data frame
Data 6		7th data byte of a CAN data frame
Data 7		8th data byte of a CAN data frame
		Note: Byte Data 0 is the first data byte shifted into the shift register of the CAN core during a reception, byte Data 7 is the last. When the message handler stores a data frame, it will write all the eight data bytes into a message object. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by undefined values.

24.10.4.11.2 Addressing Message Objects in RAM

The starting location of a particular message object in RAM is:
Message RAM base address + (message object number) × 0x20

That is, message object 1 starts at offset 0x0020; message object 2 starts at offset 0x0040, etc.

Note

Because 0 is not a valid message object number, at offset 0x0000 is not located message object 0, but the last implemented: 64.

The base address for DCAN1 RAM is 0x4AE3 D000 and DCAN2 RAM is 0x4848 1000.

Message object number 1 has the highest priority.

Table 24-748. Message RAM Addressing in Debug/Suspend and RDA Modes

Message Object Number	Offset	Word Number	Debug/Suspend Mode, see Section 24.10.4.11.3	RDA Mode, see Section 24.10.4.11.4
64 (last implemented)	0x0000	1	Parity	Data Bytes 4-7
	0x0004	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0008	3	Xtd,Dir,ID	ID[27:0],DLC
	0x000C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0010	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x0014	6	Data Bytes 7-4	-

Table 24-748. Message RAM Addressing in Debug/Suspend and RDA Modes (continued)

Message Object Number	Offset	Word Number	Debug/Suspend Mode, see Section 24.10.4.11.3	RDA Mode, see Section 24.10.4.11.4
1	0x0020	1	Parity	Data Bytes 4-7
	0x0024	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0028	3	Xtd,Dir,ID	ID[27:0],DLC
	0x002C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0030	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x0034	6	Data Bytes 7-4	-
2	0x0040	1	Parity	Data Bytes 4-7
	0x0044	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0048	3	Xtd,Dir,ID	ID[27:0],DLC
	0x004C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0050	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x0054	6	Data Bytes 7-4	-
…	…	…	…	…
63	0x07E0	1	Parity	Data Bytes 4-7
	0x07E4	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x07E8	3	Xtd,Dir,ID	ID[27:0],DLC
	0x07EC	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x07F0	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x07F4	6	Data Bytes 7-4	-

24.10.4.11.3 Message RAM Representation in Debug/Suspend Mode

In debug/suspend mode, the message RAM will be memory mapped. This allows the external debug unit to access the message RAM.

Note

During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

Table 24-749. Message RAM Representation in Debug/Suspend Mode

Offset Within MO 0x00

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="27">RESERVED</td><td colspan="5">Parity[4:0]</td></tr>
</table>

Offset Within MO 0x04

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>MXtd</td><td>MDir</td><td>RESERVED</td><td colspan="29">Msk[28:0]</td></tr>
</table>

Offset Within MO 0x08

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td>Xtd</td><td>Dir</td><td colspan="29">ID[28:0]</td></tr>
</table>

Offset Within MO	0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	MsgLst	RESERVED	UMask	TxIE	RxTE	RmtEn	RESERVED	EOB	RESERVED			DLC[3:0]			

Offset Within MO	0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							

Offset Within MO	0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_7								DATA_6								DATA_5								DATA_4							

24.10.4.11.4 Message RAM Representation in Direct Access Mode

When the DCAN_TEST[9] RDA bit is set while the DCAN module is in test mode (DCAN_CTL[7] TEST = 1), the software has direct access to the message RAM. Due to the 32-bit bus structure, the RAM is split into word lines to support this feature. The software has access to one word line at a time only.

In RAM direct access mode, the RAM is represented by a continuous memory space within the address frame of the DCAN module, starting at the message RAM base address.

Note: During direct access mode, the message RAM cannot be accessed via the IFx register sets.

Any read or write to the RAM addresses for RAM Direct Access during normal operation mode (TEST bit or RDA bit not set) will be ignored.

Table 24-750. Message RAM Representation in RAM Direct Access Mode

Offset Within MO	0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_4								DATA_5								DATA_6								DATA_7							

Offset Within MO	0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_0								DATA_1								DATA_2								DATA_3							

Offset Within MO	0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[27:0]																												DLC[3:0]			

Offset Within MO	0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Msk[28:0]																													Xtd	Dir	ID[28]

Offset Within MO	0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	Parity[4:0]	Unused	MsgLst	UMask	TxIE	RxTE	RmtEn	EOB	MXtd	MDir

Note

Writes to unused bits have no effect.

24.10.4.12 CAN Operation

After device reset, the DCAN_CTL[0] INIT is set and all CAN protocol functions are disabled. The CAN module must be initialized before operating it. Figure 24-180 illustrates the basic initialization flow for the CAN module.

24.10.4.12.1 CAN Module Initialization

A general CAN module initialization would mean the following two critical steps:

- Configuration of the CAN bit timing
- Configuration of message objects

To initialize the CAN controller, the software has to set up the CAN bit timing and those message objects that have to be used for CAN communication. Message objects that are not needed, can be deactivated.

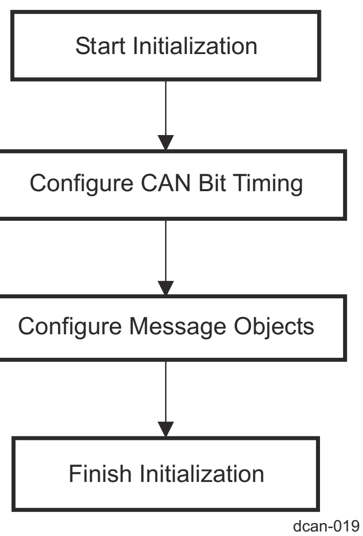


Figure 24-180. CAN Module General Initialization Flow

24.10.4.12.1.1 Configuration of CAN Bit Timing

The CAN module must be in initialization mode to configure the CAN bit timing.

For CAN bit timing software configuration flow, see Figure 24-181, *CAN Bit-Timing Configuration*.

Step 1: Enter *initialization mode* (DCAN_CTL[0] INIT = 1).

While the INIT bit is set, the message transfer from and to the CAN bus is stopped, and the status of the CAN_TX output is recessive (high).

The CAN error counters are not updated. Setting the INIT bit does not change any other configuration register.

Also, note that the CAN module is in initialization mode on device reset and during Bus-Off.

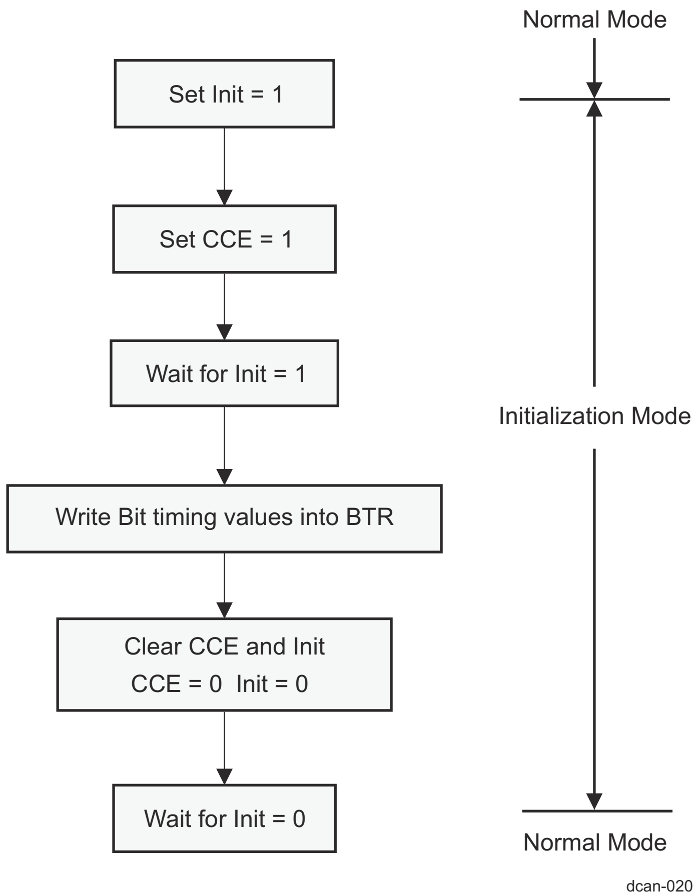


Figure 24-181. CAN Bit-Timing Configuration

Step 2: Set the Configure Change Enable bit (DCAN_CTL[6] CCE = 1).

The access to the Bit Timing register (DCAN_BTR) for the configuration of the bit timing is enabled when both INIT = 1 and CCE = 1.

Step 3: Wait for the INIT bit to get set (=1). This would make sure that the module has entered Initialization mode.

Step 4: Write the bit timing values into DCAN_BTR. See Section 24.10.4.9.2, *DCAN Bit Timing Registers* for the values calculation for a given bit timing.

Step 5: Clear the CCE and INIT bits (=0).

Step 6: Wait for the INIT bit to clear (=0). This would ensure that the module has come out of Initialization mode.

Following these steps, the module comes to operation by synchronizing itself to the CAN bus, provided the DCAN_BTR is configured as per the CAN bus baud rate, although the message objects have to be configured before carrying out any communication.

Note

The module will not come out of the Initialization mode if any incorrect DCAN_BTR values are written in step 4.

Note

The required message objects should be configured as transmit or receive objects before the start of data transfer as explained in Section 24.10.4.12.1, *CAN Module Initialization*.

24.10.4.12.1.2 Configuration of Message Objects

The message objects can be configured only through the interface registers; the software does not have direct access to the message object (message RAM) . Familiarize yourself with the interface register set (IFx) usage (see Section 24.10.4.10, *Message Interface Register Sets*) and the message object structure (see Section 24.10.4.11, *Message RAM*) before configuring the message objects.

For more information regarding the procedure to configure the message objects, see Section 24.10.4.7, *Configuration of Message Objects Description*. All the message objects should be configured to particular identifiers or set to not valid before the message transfer is started. It is possible to change the configuration of message objects during normal operation (that is between data transfers).

Note

The message objects initialization is independent of the bit-timing configuration.

24.10.4.12.1.3 DCAN RAM Hardware Initialization

The memory hardware initialization for the DCAN module is enabled in the device control module. Setting RAMINIT_START to 1, causes RAM initialization with zeros and sets parity bits accordingly. Software must wait for the RAMINIT_DONE bit to be set to ensure successful RAM initialization.

For more details on CTRL_CORE_CONTROL_IO_2 register, see *Control Module Register Manual*.

24.10.4.12.2 CAN Message Transfer (Normal Operation)

Once the DCAN is initialized and DCAN_CTL[0] INIT bit is reset (=0), the CAN core synchronizes itself to the CAN bus and is ready for message transfer as per the configured message objects.

The software may enable the interrupt lines (DCAN_CTL[1] IE0 and [17] IE1 = 1) at the same time when it clears [0] INIT and [6] CCE = 0. The status interrupts [3] EIE and [2] SIE may be enabled (=1) simultaneously.

The CAN communication can be carried out in any of the following two modes: interrupt and polling.

The DCAN_INT register points to those message objects with IntPnd = 1. It is updated even if the interrupt lines to the host processor are disabled (DCAN_CTL[1] IE0 and [17] IE1 = 0).

The software may poll all MessageObject’s NewDat and TxRqst bits in parallel from the DCAN_NWDAT_X and the DCAN_TXRQ_X registers respectively. Polling can be made easier if all transmit objects are grouped at the low numbers and all receive objects are grouped at the high numbers.

Received messages are stored into their appropriate message objects if they pass acceptance filtering.

The whole message (including all arbitration bits, DLC and up to eight data bytes) is stored into the message object. As a consequence (e.g., when the identifier mask is used), the arbitration bits which are masked to “don’t care” may change in the message object when a received message is stored.

The software may read or write each message at any time via the interface registers, as the message handler guarantees data consistency in case of concurrent accesses.

If a permanent message object (arbitration and control bits set up during configuration and leaving unchanged for multiple CAN transfers) exists for the message, it is possible to only update the data bytes.

If several transmit messages should be assigned to one message object, the whole message object has to be configured before the transmission of this message is requested.

The transmission of multiple message objects may be requested at the same time. They are subsequently transmitted, according to their internal priority.

Messages may be updated or set to not valid at any time, even if a requested transmission is still pending. However, the data bytes will be discarded if a message is updated before a pending transmission has started.

Depending on the configuration of the message object, a transmission may be automatically requested by the reception of a remote frame with a matching identifier.

24.10.4.12.2.1 Automatic Retransmission

According to the CAN Specification (ISO11898), the DCAN provides a mechanism to automatically retransmit frames which have lost arbitration or have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed.

By default, this automatic retransmission is enabled. It can be disabled by setting bit disable automatic retransmission (DCAN_CTL[5] DAR = 1). Further details to this mode are provided in Section 24.10.4.8.3, *Transmission of Messages in Event Driven CAN Communication*.

24.10.4.12.2.2 Auto-Bus-On

By default, after the DCAN has entered Bus-Off state, the software can start a Bus-Off-Recovery sequence by resetting the DCAN_CTL[0] INIT bit to 0. If this is not done, the module will stay in Bus-Off state.

The DCAN provides an automatic Auto-Bus-On feature which is enabled by bit DCAN_CTL[9] ABO. If set, the DCAN will automatically start the Bus-Off-Recovery sequence. The sequence can be delayed by a user-defined number of interface clock cycles which can be defined in the Auto-Bus-On Time register (DCAN_ABOTR).

Note

If the DCAN goes to Bus-Off state due to a massive occurrence of CAN bus errors, it stops all bus activities and automatically sets the INIT bit. Once the INIT bit has been reset by the software or due to the Auto-Bus-On feature, the device will wait for 129 occurrences of bus Idle (equal to 129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus-Off recovery sequence, the error counters will be reset.

24.10.4.12.3 Test Modes

The DCAN module provides several test modes which are mainly intended for production tests or self test.

For all test modes, the DCAN_CTL[7] TEST bit needs to be set to 1. This enables write access to the test register (DCAN_TEST).

Note

It must be ensured by software that all message transfers are completed before entering test mode.

24.10.4.12.3.1 Silent Mode

The silent mode may be used to analyze the traffic on the CAN bus without affecting it by sending dominant bits (e.g., acknowledge bit, overload flag, active error flag). The DCAN is still able to receive valid data frames and valid remote frames, but it will not send any dominant bits. However, these are internally routed to the CAN core.

Figure 24-182 shows the connection of signals CAN_TX and CAN_RX to the CAN core in silent mode. Silent mode can be activated by setting the DCAN_TEST[3] SILENT bit to 1. In ISO 11898-1, the silent mode is called the bus monitoring mode.

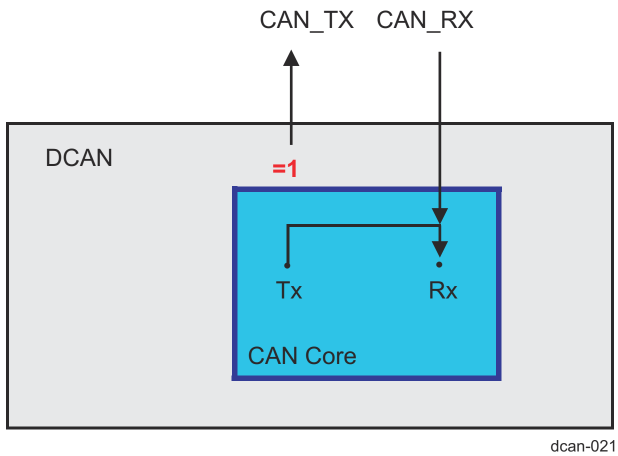


Figure 24-182. CAN Core in Silent Mode

24.10.4.12.3.2 Loopback Mode

The loopback mode is mainly intended for hardware self-test functions. In this mode, the CAN core uses internal feedback from Tx output to Rx input. Transmitted messages are treated as received messages, and can be stored into message objects if they pass acceptance filtering. The actual value of the CAN_RX input pin is disregarded by the CAN core. Transmitted messages can still be monitored at the CAN_TX pin.

In order to be independent from external stimulation, the CAN core ignores acknowledge sampled in the acknowledge slot of a data/remote frame.

Figure 24-183 shows the connection of signals CAN_TX and CAN_RX to the CAN core in loopback mode.

Loopback mode can be activated by setting bit DCAN_TEST[4] LBACK to 1.

Note

In loopback mode, the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core are disregarded. For including these into the testing, see Section 24.10.4.12.3.3, *External Loopback Mode*.

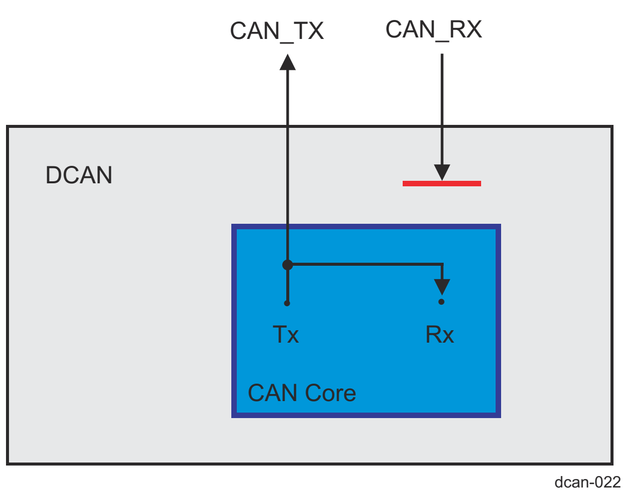


Figure 24-183. CAN Core in Loopback Mode

24.10.4.12.3.3 External Loopback Mode

The external loopback mode is similar to the loopback mode; however, it includes the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core. When external loopback mode is selected, the input of the CAN core is connected to the input buffer of the Tx pin.

With this configuration, the Tx pin IO circuit can be tested.

External loopback mode can be activated by setting bit DCAN_TEST[8] EXL to 1.

Figure 24-184 shows the connection of signals CAN_TX and CAN_RX to the CAN core in external loopback mode.

Note

When loopback mode is active (LBACK bit set), the EXL bit will be ignored.

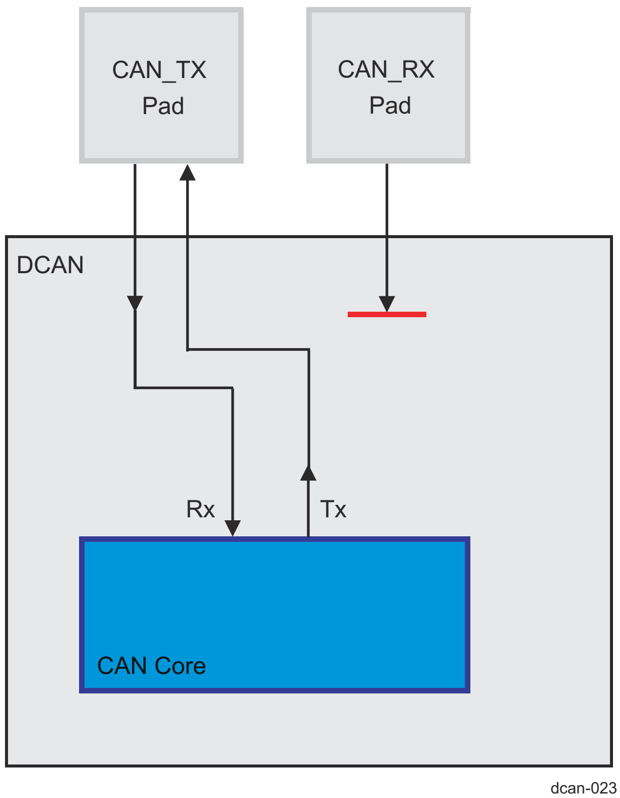


Figure 24-184. CAN Core in External Loopback Mode

24.10.4.12.3.4 Loopback Mode Combined With Silent Mode

It is also possible to combine loopback mode and silent mode by setting bits DCAN_TEST[4] LBACK and [3] SILENT at the same time. This mode can be used for a "Hot Selftest", that is, the DCAN hardware can be tested without affecting the CAN network. In this mode, the CAN_RX pin is disconnected from the CAN core and no dominant bits will be sent on the CAN_TX pin.

Figure 24-185 shows the connection of the signals CAN_TX and CAN_RX to the CAN core in case of the combination of loopback mode with silent mode.

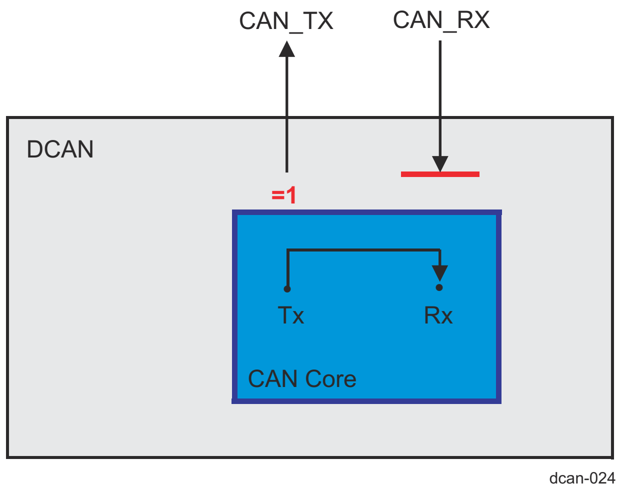


Figure 24-185. CAN Core in Loop Back Combined With Silent Mode

24.10.4.12.3.5 Software Control of CAN_TX Pin

Four output functions are available for the CAN transmit pin CAN_TX. In addition to its default function (serial data output), the CAN_TX pin can drive constant dominant or recessive values, or it can drive the CAN sample point signal to monitor the CAN core's bit timing.

Combined with the readable value of the CAN_RX pin, this function can be used to check the physical layer of the CAN bus.

The output mode of pin CAN_TX is selected by programming the DCAN_TEST[6:5] TX:
- 0x0: Normal operation, CAN_TX is controlled by the DCAN core
- 0x1: Sample point can be monitored at CAN_TX pin
- 0x2: CAN_TX pin drives a dominant value
- 0x3: CAN_TX pin drives a recessive value.

Note

The software control for the CAN_TX pin interferes with CAN protocol functions. For CAN message transfer or any of the test modes (loopback mode, external loopback mode or silent mode), the CAN_TX pin should operate in its default functionality.

24.10.4.13 GPIO Support

The CAN_RX and CAN_TX pins of the DCAN module can be used as general purpose IO pins, if CAN functionality is not needed. This function is controlled by the CAN TX IO control register (DCAN_TIOC) and the CAN RX IO control register (DCAN_RIOC).

24.10.5 DCAN Register Manual

24.10.5.1 DCAN Instance Summary

Table 24-751. DCAN Instance Summary

Module Name	Base Address	Size
DCAN1	0x4AE3 C000	8 KiB
DCAN2	0x4848 0000	8 KiB

24.10.5.2 DCAN Registers

Note

After device reset, the registers of the DCAN hold the values shown in the register descriptions.

Additionally, the Bus-Off state is reset and the CAN_TX pin is set to recessive (HIGH). The INIT bit in the DCAN_CTL is set to enable the software initialization. The DCAN will not influence the CAN bus until the software resets INIT to 0.

24.10.5.2.1 DCAN Register Summary

Table 24-752. DCAN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DCAN1 Physical Address	DCAN2 Physical Address
DCAN_CTL	RW	32	0x0000 0000	0x4AE3 C000	0x4848 0000
DCAN_ES	R	32	0x0000 0004	0x4AE3 C004	0x4848 0004
DCAN_ERRC	R	32	0x0000 0008	0x4AE3 C008	0x4848 0008
DCAN_BTR	RW	32	0x0000 000C	0x4AE3 C00C	0x4848 000C
DCAN_INT	R	32	0x0000 0010	0x4AE3 C010	0x4848 0010
DCAN_TEST	RW	32	0x0000 0014	0x4AE3 C014	0x4848 0014
DCAN_PERR	R	32	0x0000 001C	0x4AE3 C01C	0x4848 001C
DCAN_REL	R	32	0x0000 0020	0x4AE3 C020	0x4848 0020
RESERVED	RW	32	0x0000 0024	0x4AE3 C024	0x4848 0024
RESERVED	RW	32	0x0000 0028	0x4AE3 C028	0x4848 0028
RESERVED	RW	32	0x0000 002C	0x4AE3 C02C	0x4848 002C
RESERVED	R	32	0x0000 0030	0x4AE3 C030	0x4848 0030
DCAN_ABOTR	RW	32	0x0000 0080	0x4AE3 C080	0x4848 0080
DCAN_TXRQ_X	R	32	0x0000 0084	0x4AE3 C084	0x4848 0084
DCAN_TXRQ12	R	32	0x0000 0088	0x4AE3 C088	0x4848 0088
DCAN_TXRQ34	R	32	0x0000 008C	0x4AE3 C08C	0x4848 008C
DCAN_TXRQ56	R	32	0x0000 0090	0x4AE3 C090	0x4848 0090
DCAN_TXRQ78	R	32	0x0000 0094	0x4AE3 C094	0x4848 0094
DCAN_NWDAT_X	R	32	0x0000 0098	0x4AE3 C098	0x4848 0098
DCAN_NWDAT12	R	32	0x0000 009C	0x4AE3 C09C	0x4848 009C
DCAN_NWDAT34	R	32	0x0000 00A0	0x4AE3 C0A0	0x4848 00A0
DCAN_NWDAT56	R	32	0x0000 00A4	0x4AE3 C0A4	0x4848 00A4
DCAN_NWDAT78	R	32	0x0000 00A8	0x4AE3 C0A8	0x4848 00A8
DCAN_INTPND_X	R	32	0x0000 00AC	0x4AE3 C0AC	0x4848 00AC
DCAN_INTPND12	R	32	0x0000 00B0	0x4AE3 C0B0	0x4848 00B0
DCAN_INTPND34	R	32	0x0000 00B4	0x4AE3 C0B4	0x4848 00B4
DCAN_INTPND56	R	32	0x0000 00B8	0x4AE3 C0B8	0x4848 00B8
DCAN_INTPND78	R	32	0x0000 00BC	0x4AE3 C0BC	0x4848 00BC

Table 24-752. DCAN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DCAN1 Physical Address	DCAN2 Physical Address
DCAN_MSGVAL_X	R	32	0x0000 00C0	0x4AE3 C0C0	0x4848 00C0
DCAN_MSGVAL12	R	32	0x0000 00C4	0x4AE3 C0C4	0x4848 00C4
DCAN_MSGVAL34	R	32	0x0000 00C8	0x4AE3 C0C8	0x4848 00C8
DCAN_MSGVAL56	R	32	0x0000 00CC	0x4AE3 C0CC	0x4848 00CC
DCAN_MSGVAL78	R	32	0x0000 00D0	0x4AE3 C0D0	0x4848 00D0
DCAN_INTMUX12	RW	32	0x0000 00D8	0x4AE3 C0D8	0x4848 00D8
DCAN_INTMUX34	RW	32	0x0000 00DC	0x4AE3 C0DC	0x4848 00DC
DCAN_INTMUX56	RW	32	0x0000 00E0	0x4AE3 C0E0	0x4848 00E0
DCAN_INTMUX78	RW	32	0x0000 00E4	0x4AE3 C0E4	0x4848 00E4
DCAN_IF1CMD	RW	32	0x0000 0100	0x4AE3 C100	0x4848 0100
DCAN_IF1MSK	RW	32	0x0000 0104	0x4AE3 C104	0x4848 0104
DCAN_IF1ARB	RW	32	0x0000 0108	0x4AE3 C108	0x4848 0108
DCAN_IF1MCTL	RW	32	0x0000 010C	0x4AE3 C10C	0x4848 010C
DCAN_IF1DATA	RW	32	0x0000 0110	0x4AE3 C110	0x4848 0110
DCAN_IF1DATB	RW	32	0x0000 0114	0x4AE3 C114	0x4848 0114
DCAN_IF2CMD	RW	32	0x0000 0120	0x4AE3 C120	0x4848 0120
DCAN_IF2MSK	RW	32	0x0000 0124	0x4AE3 C124	0x4848 0124
DCAN_IF2ARB	RW	32	0x0000 0128	0x4AE3 C128	0x4848 0128
DCAN_IF2MCTL	RW	32	0x0000 012C	0x4AE3 C12C	0x4848 012C
DCAN_IF2DATA	RW	32	0x0000 0130	0x4AE3 C130	0x4848 0130
DCAN_IF2DATB	RW	32	0x0000 0134	0x4AE3 C134	0x4848 0134
DCAN_IF3OBS	RW	32	0x0000 0140	0x4AE3 C140	0x4848 0140
DCAN_IF3MSK	RW	32	0x0000 0144	0x4AE3 C144	0x4848 0144
DCAN_IF3ARB	R	32	0x0000 0148	0x4AE3 C148	0x4848 0148
DCAN_IF3MCTL	R	32	0x0000 014C	0x4AE3 C14C	0x4848 014C
DCAN_IF3DATA	R	32	0x0000 0150	0x4AE3 C150	0x4848 0150
DCAN_IF3DATB	R	32	0x0000 0154	0x4AE3 C154	0x4848 0154
DCAN_IF3UPD12	RW	32	0x0000 0160	0x4AE3 C160	0x4848 0160
DCAN_IF3UPD34	RW	32	0x0000 0164	0x4AE3 C164	0x4848 0164
DCAN_IF3UPD56	RW	32	0x0000 0168	0x4AE3 C168	0x4848 0168
DCAN_IF3UPD78	RW	32	0x0000 016C	0x4AE3 C16C	0x4848 016C
DCAN_TIOC	RW	32	0x0000 01E0	0x4AE3 C1E0	0x4848 01E0
DCAN_RIOC	RW	32	0x0000 01E4	0x4AE3 C1E4	0x4848 01E4

24.10.5.2.2 DCAN Register Description

Table 24-753. DCAN_CTL

Address Offset	0x0000 0000		
Physical Address	0x4AE3 C000 0x4848 0000	**Instance**	DCAN1 DCAN2
Description	DCAN control register **NOTE**: The Bus-Off recovery sequence (refer to CAN specification) cannot be shortened by setting or resetting INIT bit. If the module goes Bus-Off, it will automatically set the INIT bit and stop all bus activities. When the INIT bit is cleared by the application again, the module will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the bus-off recovery sequence, the error counters will be reset. After the INIT bit is reset, each time when a sequence of 11 recessive bits is monitored, a Bit0 error code is written to DCAN_ES, enabling the software to check whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the proceeding of the bus-off recovery sequence.		

Table 24-753. DCAN_CTL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						WUBA	PDR	RESERVED			DE3	DE2	DE1	IE1	INITDBG	SWR	RESERVED	PMD				ABO	IDS	TEST	CCE	DAR	RESERVED	EIE	SIE	IE0	INIT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
25	WUBA	Automatic wake up on bus activity when in local power-down mode. Note: The CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down and automatic wake-up mode, will be lost. 0: No detection of a dominant CAN bus level while in local power-down mode. 1: Detection of a dominant CAN bus level while in local power-down mode is enabled. On occurrence of a dominant CAN bus level, the wake up sequence is started (Additional information can be found in *Local Power-Down Mode*).	RW	0
24	PDR	Request for local low power-down mode 0: No application request for local low power-down mode. If the application has cleared this bit while DCAN in local power-down mode, also the INIT bit has to be cleared. 1: Local power-down mode has been requested by application. The DCAN will acknowledge the local power-down mode by setting bit PDA in the DCAN_ES register. The local clocks will be turned off by DCAN internal logic (Additional information can be found in *Local Power-Down Mode*).	RW	0
23:21	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
20	DE3	Enable DMA request line for IF3. Note: A pending DMA request for IF3 remains active until first access to one of the IF3 registers. 0: Disabled 1: Enabled	RW	0
19	DE2	Enable DMA request line for IF2. Note: A pending DMA request for IF2 remains active until first access to one of the IF2 registers. 0: Disabled 1: Enabled	RW	0
18	DE1	Enable DMA request line for IF1. Note: A pending DMA request for IF1 remains active until first access to one of the IF1 registers. 0: Disabled 1: Enabled	RW	0
17	IE1	Interrupt line 1 enable 0: Disabled - Module interrupt INT1 is always low. 1: Enabled - interrupts will assert line INT1 to one; line remains active until pending interrupts are processed.	RW	0

Bits	Field Name	Description	Type	Reset
16	INITDBG	Internal init state while debug access 0: Not in debug mode, or debug mode requested but not entered. 1: Debug mode requested and internally entered; the DCAN is ready for debug accesses.	RW	0
15	SWR	Software reset enable. Note: To execute software reset, the following procedure is necessary: 1. Set INIT bit to shut down CAN communication. 2. Set SWR (this) bit additionally to INIT bit. 0: Normal Operation 1: Module is forced to reset state. This bit will automatically get cleared after execution of software reset after one OCP clock cycle.	RW	0
14	RESERVED	This bit is always read as 0. Writes have no effect.	R	0
13:10	PMD	Parityon/off 0x5: function disabled Others: function enabled	RW	0x5
9	ABO	Auto-Bus-On enable 0: The Auto-Bus-On feature is disabled 1: The Auto-Bus-On feature is enabled	RW	0
8	IDS	Interruption debug support enable 0: When Debug/Suspend mode is requested, DCAN will wait for a started transmission or reception to be completed before entering Debug/Suspend mode 1: When Debug/Suspend mode is requested, DCAN will interrupt any transmission or reception, and enter Debug/ Suspend mode immediately.	RW	0
7	TEST	Test mode enable 0: Normal Operation 1: Test Mode	RW	0
6	CCE	Configuration change enable 0: The software has no write access to the configuration registers. 1: The software has write access to the configuration registers (when INIT bit is set).	RW	0
5	DAR	Disable automatic retransmission 0: Automatic retransmission of not successful messages enabled. 1: Automatic retransmission disabled.	RW	0
4	RESERVED	This bit is always read as 0. Writes have no effect.	R	0
3	EIE	Error interrupt enable 0: Disabled - PER, BOFF and EWARN bits can not generate an interrupt. 1: Enabled - PER, BOFF and EWARN bits can generate an interrupt at INT0 line and affect the interrupt register.	RW	0
2	SIE	Status change interrupt enable 0: Disabled - WAKEUPPND, RXOK, TXOK and LEC bits can not generate an interrupt. 1: Enabled - WAKEUPPND, RXOK, TXOK and LEC can generate an interrupt at INT0 line and affect the interrupt register.	RW	0

Bits	Field Name	Description	Type	Reset
1	IE0	Interrupt line 0 enable 0: Disabled - Module interrupt INT0 is always low. 1: Enabled - interrupts will assert line INT0 to one; line remains active until pending interrupts are processed.	RW	0
0	INIT	Initialization 0: Normal operation 1: Initialization mode is entered	RW	1

Table 24-754. Register Call Summary for Register DCAN_CTL

DCAN

- Interrupt Functionality: [0]
- Status Change Interrupts: [1] [2]
- Error Interrupts: [3] [4]
- DMA Functionality: [5]
- Entering Local Power-Down Mode: [6] [7]
- Wakeup From Local Power Down: [8] [9] [10] [11] [12]
- Parity Check Mechanism: [13]
- Debug/Suspend Mode: [16] [17]
- Transmission of Messages in Event Driven CAN Communication: [18]
- IF3 Register Set: [19]
- Structure of Message Objects: [20]
- Message RAM Representation in Direct Access Mode: [21]
- CAN Operation: [22]
- CAN Module Initialization: [23] [24]
- CAN Message Transfer (Normal Operation): [25] [26] [27] [28] [29] [30]
- Test Modes: [31]
- DCAN Registers: [32]
- DCAN Register Summary: [33]
- DCAN Register Description: [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52]

Table 24-755. DCAN_ES

Address Offset	0x0000 0004		
Physical Address	0x4AE3 C004 0x4848 0004	**Instance**	DCAN1 DCAN2
Description	Error and Status Register Interrupts are generated by bits PER, BOFF and EWARN (if EIE bit in DCAN_CTL is 1) and by bits WAKEUPPND, RXOK, TXOK, and LEC (if SIE bit in DCAN_CTL is 1). A change of bit EPASS will not generate an interrupt. Reading the DCAN_ES clears the WAKEUPPND, PER, RXOK and TXOK bits and set the LEC to value '7.' Additionally, the status interrupt value (0x8000) in the DCAN_INT will be replaced by the next lower priority interrupt value. For debug support, the auto clear functionality of DCAN_ES (clear of status flags by read) is disabled when in debug/suspend mode.		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">RESERVED</td><td>PDA</td><td>WAKEUPPND</td><td>PER</td><td>BOFF</td><td>EWARN</td><td>EPASS</td><td>RXOK</td><td>TXOK</td><td colspan="3">LEC</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00 0000

Bits	Field Name	Description	Type	Reset
10	PDA	Local power-down mode acknowledge 0: DCAN is not in local power-down mode. 1: Application request for setting DCAN to local power-down mode was successful. DCAN is in local power-down mode.	R	0
9	WAKEUPPND	Wake up pending. This bit can be used by the software to identify the DCAN as the source to wake up the system. This bit will be reset if DCAN_ES is read. 0: No Wake Up is requested by DCAN. 1: DCAN has initiated a wake up of the system due to dominant CAN bus while module power down.	R	0
8	PER	Parity error detected. This bit will be reset if DCAN_ES register is read. Read 0: No parity error has been detected since last read access. Read 1: The parity check mechanism has detected a parity error in the Message RAM. Write 0: No effect Write 1: End of interrupt (EOI) for parity error on DCAN_PARITY interrupt line	RW	0
7	BOFF	Bus-Off state 0: The CAN module is not bus-off state. 1: The CAN module is in bus-off state.	R	0
6	EWARN	Warning state 0: Both error counters are below the error warning limit of 96. 1: At least one of the error counters has reached the error warning limit of 96.	R	0
5	EPASS	Error passive state 0: On CAN Bus error, the DCAN could send active error frames. 1: The CAN core is in the error passive state as defined in the CAN Specification.	R	0
4	RXOK	Received a message successfully. This bit will be reset if DCAN_ES register is read. 0: No message has been successfully received since the last time when this bit was read by the software. This bit is never reset by DCAN internal events. 1: A message has been successfully received since the last time when this bit was reset by a read access of the software (independent of the result of acceptance filtering).	R	0
3	TXOK	Transmitted a message successfully. This bit will be reset if DCAN_ES register is read. 0: No message has been successfully transmitted since the last time when this bit was read by the software. This bit is never reset by DCAN internal events. 1: A message has been successfully transmitted (error free and acknowledged by at least one other node) since the last time when this bit was reset by a read access of the software.	R	0

Bits	Field Name	Description	Type	Reset
2:0	LEC	Last error code. The LEC field indicates the type of the last error on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. 0x0: No error 0x1: Stuff error: More than five equal bits in a row have been detected in a part of a received message where this is not allowed. 0x2: Form error: A fixed format part of a received frame has the wrong format. 0x3: Ack error: The message this CAN core transmitted was not acknowledged by another node. 0x4: Bit1 error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 0x5: Bit0 error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (logical value '0'), but the monitored bus level was recessive. During Bus-Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the software to monitor the proceeding of the Bus-Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 0x6: CRC error: In a received message, the CRC check sum was incorrect. (CRC received for an incoming message does not match the calculated CRC for the received data). 0x7: No CAN bus event was detected since the last time the software read DCAN_ES. Any read access to DCAN_ES re-initializes the LEC to value '7.'	R	0x7

Table 24-756. Register Call Summary for Register DCAN_ES

DCAN

- Interrupt Functionality: [0] [1]
- Status Change Interrupts: [2] [3]
- Error Interrupts: [4]
- Entering Local Power-Down Mode: [5]
- Wakeup From Local Power Down: [6] [7]
- Behavior on Parity Error: [8]
- Debug/Suspend Mode: [10]
- DCAN Register Summary: [11]
- DCAN Register Description: [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24]

Table 24-757. DCAN_ERRC

Address Offset	0x0000 0008		
Physical Address	0x4AE3 C008 0x4848 0008	**Instance**	DCAN1 DCAN2
Description	Error Counter Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RP	REC							TEC							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	RP	Receive error passive 0: The receive error counter is below the error passive level. 1: The receive error counter has reached the error passive level as defined in the CAN specification.	R	0
14:8	REC	Receive error counter. Actual state of the receive error counter	R	0x00
7:0	TEC	Transmit error counter. Actual state of the transmit error counter	R	0x00

Table 24-758. Register Call Summary for Register DCAN_ERRC

DCAN

- DCAN Register Summary: [0]

Table 24-759. DCAN_BTR

Address Offset	0x0000 000C		
Physical Address	0x4AE3 C00C 0x4848 000C	**Instance**	DCAN1 DCAN2
Description	Bit timing register This register is only writable if CCE and INIT bits in the DCAN_CTL are set. The CAN bit time may be programmed in the range of 8 to 25 time quanta The CAN time quantum may be programmed in the range of 1 to 1024 CAN_CLK periods.		
Type	RW		

31:20	19:16	15	14:12	11:8	7:6	5:0
RESERVED	BRPE	RESERVED	TSEG2	TSEG1	SJW	BRP

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x000
19:16	BRPE	Baud rate prescaler extension. Valid programmed values are 0 to 15. By programming BRPE the baud rate prescaler can be extended to values up to 1024.	RW	0x0
15	RESERVED	These bits are always read as 0. Writes have no effect.	R	0
14:12	TSEG2	Time segment after the sample point Valid programmed values are 0 to 7. The actual TSeg2 value which is interpreted for the bit timing will be the programmed TSeg2 value + 1.	RW	0x2
11:8	TSEG1	Time segment before the sample point Valid programmed values are 1 to15. The actual TSeg1 value interpreted for the bit timing will be the programmed TSeg1 value + 1.	RW	0x3
7:6	SJW	Synchronization Jump Width Valid programmed values are 0 to 3. The actual SJW value interpreted for the synchronization will be the programmed SJW value + 1.	RW	0x0

Bits	Field Name	Description	Type	Reset
5:0	BRP	Baud rate prescaler Value by which the CAN_CLK frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid programmed values are 0 to 63. The actual BRP value interpreted for the bit timing will be the programmed BRP value + 1.	RW	0x1

Table 24-760. Register Call Summary for Register DCAN_BTR

DCAN

- DCAN Bit Timing Registers: [0] [1] [2] [3] [4] [5]
- CAN Module Initialization: [6] [7] [8] [9]
- DCAN Register Summary: [10]

Table 24-761. DCAN_INT

Address Offset	0x0000 0010		
Physical Address	0x4AE3 C010 0x4848 0010	**Instance**	DCAN1 DCAN2
Description	Interrupt register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INT1ID								INT0ID															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
23:16	INT1ID	Interrupt 1 Identifier (indicates the message object with the highest pending interrupt) 0x00: No interrupt is pending 0x01-0x80: Number of message object which caused the interrupt. 0x81-0xFF: Unused If several interrupts are pending, DCAN_INT will point to the pending interrupt with the highest priority. The INT1 interrupt line remains active until INT1ID reaches value 0 (the cause of the interrupt is reset) or until IE1 is cleared. A message interrupt is cleared by clearing the message object's IntPnd bit. Among the message interrupts, the message object's interrupt priority decreases with increasing message number.	R	0x00

Bits	Field Name	Description	Type	Reset
15:0	INT0ID	Interrupt Identifier (the number here indicates the source of the interrupt) 0x0000: No interrupt is pending 0x0001-0x0080: Number of message object which caused the interrupt. 0x0081-0x7FFF: Unused 0x8000: DCAN_ES value is not 0x07. 0x8001-0xFFFF: Unused If several interrupts are pending,DCAN_INTwill point to the pending interrupt with the highest priority. The INT0 interrupt line remains active until INT0ID reaches value 0 (the cause of the interrupt is reset) or until IE0 is cleared. The Status interrupt has the highest priority. Among the message interrupts, the message object's interrupt priority decreases with increasing message number.	R	0x0000

Table 24-762. Register Call Summary for Register DCAN_INT

DCAN

- Interrupt Functionality: [0] [1] [2]
- Reception of Data Frames: [3]
- Structure of Message Objects: [4]
- CAN Message Transfer (Normal Operation): [5]
- DCAN Register Summary: [6]
- DCAN Register Description: [7] [8] [9] [10] [11] [12] [13] [14] [15]

Table 24-763. DCAN_TEST

Address Offset	0x0000 0014		
Physical Address	0x4AE3 C014 0x4848 0014	**Instance**	DCAN1 DCAN2
Description	Test Register For all test modes, the TEST bit in DCAN_CTL control register needs to be set to 1. If TEST bit is set, the RDA, EXL, TX1, TX0, LBACK and SILENT bits are writable. Bit RX monitors the state of pin CAN_RX and therefore is only readable. All test register functions are disabled when TEST bit is cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RDA	EXL	RX	TX		LBACK	SILENT	RESERVED		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00 0000
9	RDA	RAM direct access enable 0: Normal operation 1: Direct access to the RAM is enabled while in test mode	RW	0
8	EXL	External loopback mode. When the internal loop-back mode is active (bit LBACK is set), bit EXL will be ignored. 0: Disabled 1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
7	RX	Receive pin. Monitors the actual value of the CAN_RX pin 0: The CAN bus is dominant 1: The CAN bus is recessive	R	-
6:5	TX	Control of CAN_TX pin. Setting Tx[1:0] other than '00' will disturb message transfer. 0x0: Normal operation, CAN_TX is controlled by the CAN core. 0x1: Sample point can be monitored at CAN_TX pin. 0x2: CAN_TX pin drives a dominant value. 0x3: CAN_TX pin drives a recessive value.	RW	0x0
4	LBACK	Loopback mode. When the internal loop-back mode is active (bit LBACK is set), bit EXL will be ignored. 0: Disabled 1: Enabled	RW	0
3	SILENT	Silent mode 0: Disabled 1: Enabled	RW	0
2:0	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0

Table 24-764. Register Call Summary for Register DCAN_TEST

DCAN

- Parity Testing: [0]
- Message RAM Representation in Direct Access Mode: [1]
- Test Modes: [2] [3] [4] [5] [6] [7]
- DCAN Register Summary: [8]

Table 24-765. DCAN_PERR

Address Offset	0x0000 001C		
Physical Address	0x4AE3 C01C 0x4848 001C	**Instance**	DCAN1 DCAN2
Description	Parity Error Code Register If a parity error is detected, the PER flag will be set in the DCAN_ES. This bit is not reset by the parity check mechanism; it must be reset by reading DCAN_ES. In addition to the PER flag, the parity error code register will indicate the memory area where the parity error has been detected (message number and word number). If more than one word with a parity error was detected, the highest word number with a parity error will be displayed. After a parity error has been detected, the register will hold the last error code until power is removed.		
Type	R		

31:11	10:8	7:0
RESERVED	WORD_NUMBER	MESSAGE_NUMBER

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00 0000
10:8	WORD_NUMBER	Word number where parity error has been detected RDA word number (1 to 5) of the message object (according to the message RAM representation in RDA mode).	R	0x-
7:0	MESSAGE_NUMBER	Message object number where parity error has been detected (0x01-0x80)	R	0x-

Table 24-766. Register Call Summary for Register DCAN_PERR

DCAN
- Behavior on Parity Error: [0]
- DCAN Register Summary: [3]

Table 24-767. DCAN_REL

Address Offset	0x0000 0020		
Physical Address	0x4AE3 C020 0x4848 0020	**Instance**	DCAN1 DCAN2
Description	Core revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	DCAN core revision number	R	0x-

Table 24-768. Register Call Summary for Register DCAN_REL

DCAN
- DCAN Register Summary: [0]

Table 24-769. DCAN_ABOTR

Address Offset	0x0000 0080		
Physical Address	0x4AE3 C080 0x4848 0080	**Instance**	DCAN1 DCAN2
Description	Auto-Bus-On Time Register On write access to the DCAN_CTL while Auto-Bus-On timer is running, the Auto-Bus-On procedure will be aborted. During Debug/Suspend mode, running Auto-Bus-On timer will be paused.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABO_TIME																															

Bits	Field Name	Description	Type	Reset
31:0	ABO_TIME	Number of OCP clock cycles before a Bus-Off recovery sequence is started by clearing the INIT bit. This function has to be enabled by setting bit ABO in DCAN_CTL. The Auto-Bus-On timer is realized by a 32-bit counter which starts to count down to zero when the module goes Bus-Off. The counter will be reloaded with the preload value of the DCAN_ABOTR after this phase.	RW	0x0000 0000

Table 24-770. Register Call Summary for Register DCAN_ABOTR

DCAN
- CAN Message Transfer (Normal Operation): [0]
- DCAN Register Summary: [1]
- DCAN Register Description: [2]

Table 24-771. DCAN_TXRQ_X

Address Offset	0x0000 0084		
Physical Address	0x4AE3 C084 0x4848 0084	**Instance**	DCAN1 DCAN2

Table 24-771. DCAN_TXRQ_X (continued)

Description	Transmission Request X Register The software can detect if one or more bits in the different transmission request registers are set. Each register bit represents a group of eight message objects. If at least one of the TxRqst bits of these message objects are set, the corresponding bit in the transmission request X register will be set.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXRQSTREG8		TXRQSTREG7		TXRQSTREG6		TXRQSTREG5		TXRQSTREG4		TXRQSTREG3		TXRQSTREG2		TXRQSTREG1	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	RESERVED	R	0x0000
15:14	TXRQSTREG8	Transmission request bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	TXRQSTREG7	Transmission request bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	TXRQSTREG6	Transmission request bits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	TXRQSTREG5	Transmission request bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	TXRQSTREG4	Transmission request bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
5:4	TXRQSTREG3	Transmission request bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	TXRQSTREG2	Transmission request bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	TXRQSTREG1	Transmission request bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 24-772. Register Call Summary for Register DCAN_TXRQ_X

DCAN

- CAN Message Transfer (Normal Operation): [0]
- DCAN Register Summary: [1]

Table 24-773. DCAN_TXRQ12

Address Offset	0x0000 0088		
Physical Address	0x4AE3 C088 0x4848 0088	**Instance**	DCAN1 DCAN2
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TXRQS

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 1-32 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 24-774. Register Call Summary for Register DCAN_TXRQ12

DCAN

- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 24-775. DCAN_TXRQ34

Address Offset	0x0000 008C		
Physical Address	0x4AE3 C08C 0x4848 008C	**Instance**	DCAN1 DCAN2
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 33-64 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 24-776. Register Call Summary for Register DCAN_TXRQ34

DCAN

- DCAN Register Summary: [0]

Table 24-777. DCAN_TXRQ56

Address Offset	0x0000 0090		
Physical Address	0x4AE3 C090 0x4848 0090	**Instance**	DCAN1 DCAN2
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 65-96 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 24-778. Register Call Summary for Register DCAN_TXRQ56

DCAN

- DCAN Register Summary: [0]

Table 24-779. DCAN_TXRQ78

Address Offset	0x0000 0094		
Physical Address	0x4AE3 C094 0x4848 0094	**Instance**	DCAN1 DCAN2
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 97-128 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 24-780. Register Call Summary for Register DCAN_TXRQ78

DCAN

- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 24-781. DCAN_NWDAT_X

Address Offset	0x0000 0098		
Physical Address	0x4AE3 C098 0x4848 0098	**Instance**	DCAN1 DCAN2
Description	New Data X Register With the new data X register, the software can detect if one or more bits in the different new data registers are set. Each register bit represents a group of eight message objects. If at least on of the NewDat bits of these message objects are set, the corresponding bit in the new data X register will be set		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NEWDATREG8		NEWDATREG7		NEWDATREG6		NEWDATREG5		NEWDATREG4		NEWDATREG3		NEWDATREG2		NEWDATREG1	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000

Bits	Field Name	Description	Type	Reset
15:14	NEWDATREG8	New data bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	NEWDATREG7	New data bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	NEWDATREG6	New data bits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	NEWDATREG5	New data bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	NEWDATREG4	New data bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
5:4	NEWDATREG3	New data bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	NEWDATREG2	New data bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	NEWDATREG1	New data bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 24-782. Register Call Summary for Register DCAN_NWDAT_X

DCAN

- CAN Message Transfer (Normal Operation): [0]
- DCAN Register Summary: [1]

Table 24-783. DCAN_NWDAT12

Address Offset	0x0000 009C		
Physical Address	0x4AE3 C09C 0x4848 009C	**Instance**	DCAN1 DCAN2
Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 1-32 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 24-784. Register Call Summary for Register DCAN_NWDAT12

DCAN

- DCAN Register Summary: [0]

Table 24-785. DCAN_NWDAT34

Address Offset	0x0000 00A0		
Physical Address	0x4AE3 C0A0 0x4848 00A0	**Instance**	DCAN1 DCAN2
Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.		
Type	R		

31–0
NEWDAT

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 33-64 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 24-786. Register Call Summary for Register DCAN_NWDAT34

DCAN

- DCAN Register Summary: [0]

Table 24-787. DCAN_NWDAT56

Address Offset	0x0000 00A4		
Physical Address	0x4AE3 C0A4 0x4848 00A4	**Instance**	DCAN1 DCAN2
Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.		
Type	R		

31–0
NEWDAT

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 65-96 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 24-788. Register Call Summary for Register DCAN_NWDAT56

DCAN

- DCAN Register Summary: [0]

Table 24-789. DCAN_NWDAT78

Address Offset	0x0000 00A8		
Physical Address	0x4AE3 C0A8 0x4848 00A8	**Instance**	DCAN1 DCAN2

Table 24-789. DCAN_NWDAT78 (continued)

Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 97-128 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 24-790. Register Call Summary for Register DCAN_NWDAT78

DCAN

- DCAN Register Summary: [0]

Table 24-791. DCAN_INTPND_X

Address Offset	0x0000 00AC		
Physical Address	0x4AE3 C0AC 0x4848 00AC	**Instance**	DCAN1 DCAN2
Description	Interrupt Pending X Register With the interrupt pending X register, the software can detect if one or more bits in the different interrupt pending registers are set. Each bit of this register represents a group of eight message objects. If at least one of the IntPnd bits of these message objects are set, the corresponding bit in the interrupt pending X register will be set.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INTPNDREG8		INTPNDREG7		INTPNDREG6		INTPNDREG5		INTPNDREG4		INTPNDREG3		INTPNDREG2		INTPNDREG1	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:14	INTPNDREG8	Interrupt Pending bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	INTPNDREG7	Interrupt Pending bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	INTPNDREG6	Interrupt Pendingbits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	INTPNDREG5	Interrupt Pending bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	INTPNDREG4	Interrupt Pending bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Bits	Field Name	Description	Type	Reset
5:4	INTPNDREG3	Interrupt Pending bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	INTPNDREG2	Interrupt Pending bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	INTPNDREG1	Interrupt Pending bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 24-792. Register Call Summary for Register DCAN_INTPND_X

DCAN

- DCAN Register Summary: [0]

Table 24-793. DCAN_INTPND12

Address Offset	0x0000 00B0		
Physical Address	0x4AE3 C0B0 0x4848 00B0	**Instance**	DCAN1 DCAN2
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 1-32 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 24-794. Register Call Summary for Register DCAN_INTPND12

DCAN

- DCAN Register Summary: [0]

Table 24-795. DCAN_INTPND34

Address Offset	0x0000 00B4		
Physical Address	0x4AE3 C0B4 0x4848 00B4	**Instance**	DCAN1 DCAN2
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 33-64 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 24-796. Register Call Summary for Register DCAN_INTPND34

DCAN

- DCAN Register Summary: [0]

Table 24-797. DCAN_INTPND56

Address Offset	0x0000 00B8		
Physical Address	0x4AE3 C0B8 0x4848 00B8	**Instance**	DCAN1 DCAN2
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 65-96 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 24-798. Register Call Summary for Register DCAN_INTPND56

DCAN

- DCAN Register Summary: [0]

Table 24-799. DCAN_INTPND78

Address Offset	0x0000 00BC		
Physical Address	0x4AE3 C0BC 0x4848 00BC	**Instance**	DCAN1 DCAN2
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 97-128 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 24-800. Register Call Summary for Register DCAN_INTPND78

DCAN

- DCAN Register Summary: [0]

Table 24-801. DCAN_MSGVAL_X

Address Offset	0x0000 00C0		
Physical Address	0x4AE3 C0C0 0x4848 00C0	**Instance**	DCAN1 DCAN2
Description	Message Valid X Register With the message valid X register, the software can detect if one or more bits in the different message valid registers are set. Each bit of this register represents a group of eight message objects. If at least one of the MsgVal bits of these message objects are set, the corresponding bit in the message valid X register will be set.		
Type	R		

31:16	15:14	13:12	11:10	9:8	7:6	5:4	3:2	1:0
RESERVED	MSGVALREG8	MSGVALREG7	MSGVALREG6	MSGVALREG5	MSGVALREG4	MSGVALREG3	MSGVALREG2	MSGVALREG1

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:14	MSGVALREG8	Message valid bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	MSGVALREG7	Message valid bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	MSGVALREG6	Message valid bits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	MSGVALREG5	Message valid bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	MSGVALREG4	Message valid bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
5:4	MSGVALREG3	Message valid bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	MSGVALREG2	Message valid bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	MSGVALREG1	Message valid bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 24-802. Register Call Summary for Register DCAN_MSGVAL_X

DCAN

- DCAN Register Summary: [0]

Table 24-803. DCAN_MSGVAL12

Address Offset	0x0000 00C4		
Physical Address	0x4AE3 C0C4 0x4848 00C4	**Instance**	DCAN1 DCAN2

Table 24-803. DCAN_MSGVAL12 (continued)

Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 1-32 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 24-804. Register Call Summary for Register DCAN_MSGVAL12

DCAN
- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 24-805. DCAN_MSGVAL34

Address Offset	0x0000 00C8		
Physical Address	0x4AE3 C0C8 0x4848 00C8	**Instance**	DCAN1 DCAN2
Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 33-64 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 24-806. Register Call Summary for Register DCAN_MSGVAL34

DCAN
- DCAN Register Summary: [0]

Table 24-807. DCAN_MSGVAL56

Address Offset	0x0000 00CC		
Physical Address	0x4AE3 C0CC 0x4848 00CC	**Instance**	DCAN1 DCAN2

Table 24-807. DCAN_MSGVAL56 (continued)

Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 65-96 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 24-808. Register Call Summary for Register DCAN_MSGVAL56

DCAN

- DCAN Register Summary: [0]

Table 24-809. DCAN_MSGVAL78

Address Offset	0x0000 00D0		
Physical Address	0x4AE3 C0D0 0x4848 00D0	**Instance**	DCAN1 DCAN2
Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 97-128 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 24-810. Register Call Summary for Register DCAN_MSGVAL78

DCAN

- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 24-811. DCAN_INTMUX12

Address Offset	0x0000 00D8		
Physical Address	0x4AE3 C0D8 0x4848 00D8	**Instance**	DCAN1 DCAN2

Table 24-811. DCAN_INTMUX12 (continued)

Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in DCAN_CTL. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp. INT1ID flags in the DCAN_INT register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bit 0 -> last implemented message object) (bits 1:31 -> 1-31 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 24-812. Register Call Summary for Register DCAN_INTMUX12

DCAN

- Message Object Interrupts: [0]
- DCAN Register Summary: [1]

Table 24-813. DCAN_INTMUX34

Address Offset	0x0000 00DC		
Physical Address	0x4AE3 C0DC 0x4848 00DC	**Instance**	DCAN1 DCAN2
Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp INT1ID flags in the DCAN_INT register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bits 0:31 -> 32-63 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 24-814. Register Call Summary for Register DCAN_INTMUX34

DCAN

- DCAN Register Summary: [0]

Table 24-815. DCAN_INTMUX56

Address Offset	0x0000 00E0		
Physical Address	0x4AE3 C0E0 0x4848 00E0	**Instance**	DCAN1 DCAN2

Table 24-815. DCAN_INTMUX56 (continued)

Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp INT1ID flags in the DCAN_INT register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bits 0:31 -> 64-95 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 24-816. Register Call Summary for Register DCAN_INTMUX56

DCAN

- DCAN Register Summary: [0]

Table 24-817. DCAN_INTMUX78

Address Offset	0x0000 00E4		
Physical Address	0x4AE3 C0E4 0x4848 00E4	**Instance**	DCAN1 DCAN2
Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp INT1ID flags in the DCAN_INT register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bits 0:31 -> 96-127 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 24-818. Register Call Summary for Register DCAN_INTMUX78

DCAN

- Message Object Interrupts: [0]
- DCAN Register Summary: [1]

Table 24-819. DCAN_IF1CMD

Address Offset	0x0000 0100		
Physical Address	0x4AE3 C100 0x4848 0100	**Instance**	DCAN1 DCAN2

Table 24-819. DCAN_IF1CMD (continued)

Description	IF1 Command Register The IF1 Command Register (DCAN_IF1CMD) configure and initiate the transfer between the IF1 register set and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the software writes the message number to bits [7:0] MESSAGE_NUMBER. With this write operation, the BUSY bit is automatically set to 1 to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the BUSY bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the software writes to both DCAN_IF1CMD/DCAN_IF2CMD consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While BUSY bit is one, IF1/IF2 register sets are write protected. For debug support, the auto clear functionality of the IF1/IF2 command registers (clear of DMAACTIVE flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] MESSAGE_NUMBER, the message handler may access an implemented (valid) message object instead.
Type	RW

31–24	23	22	21	20	19	18	17	16	15	14	13–8	7–0
RESERVED	WR_RD	MASK	ARB	CONTROL	CLRINTPND	TXRQST_NEWDAT	DATA_A	DATA_B	BUSY	DMAACTIVE	RESERVED	MESSAGE_NUMBER

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
23	WR_RD	Write/Read 0: Direction = Read: Transfer direction is from the message object addressed by MESSAGE_NUMBER to the IF1 register set. 1: Direction = Write: Transfer direction is from the IF1 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
22	MASK	Access mask bits 0: Mask bits will not be changed 1: Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by MESSAGE_NUMBER to the IF1 register set. 1: Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF1 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
21	ARB	Access arbitration bits 0: Arbitration bits will not be changed 1: Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF1 register set. 1: Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF1 register set to the message object addressed by MESSAGE_NUMBER.	RW	0

Bits	Field Name	Description	Type	Reset
20	CONTROL	Access control bits 0: Control bits will not be changed 1: Direction = Read: The message control bits will be transferred from the message object addressed by MESSAGE_NUMBER to the IF1 register set. 1: Direction = Write: The message control bits will be transferred from the IF1 registerset to the message object addressed by MESSAGE_NUMBER. If the TXRQST_NEWDAT bit in this register(Bit [18]) is set, the TXRQST/ NEWDAT bits in the DCAN_IF1MCTL will be ignored.	RW	0
19	CLRINTPND	Clear interrupt pending bit 0: IntPnd bit will not be changed 1: Direction = Read: Clears IntPnd bit in the message object. 1: Direction = Write: This bit is ignored. Copying of IntPnd flag from IF1 Registers to message RAM can only be controlled by the CONTROL flag (Bit [20]).	RW	0
18	TXRQST_NEWDAT	Access transmission request bit 0: Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the CONTROL bit. 1: Direction = Read: Clears NewDat bit in the message object. 1: Direction = Write: Sets TxRqst/NewDat in message object. Note: If a CAN transmission is requested by setting TXRQST_NEWDAT in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in DCAN_IF1MCTL. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the DCAN_IF1MCTL always reflect the status before resetting them.	RW	0
17	DATA_A	Access Data Bytes 0-3 0: Data Bytes 0-3 will not be changed. 1: Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the MESSAGE_NUMBER to the corresponding IF1 registerset. 1: Direction = Write: The data bytes 0-3 will be transferred from the IF1 registerset to the message object addressed by the MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0
16	DATA_B	Access Data Bytes 4-7 0: Data Bytes 4-7 will not be changed. 1: Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF1 registerset. 1: Direction = Write: The data bytes 4-7 will be transferred from the IF1 registerset to the message object addressed by MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0

Bits	Field Name	Description	Type	Reset
15	BUSY	Busy flag 0: No transfer between IF1 register set and message RAM is in progress. 1: Transfer between IF1 register set and message RAM is in progress. This bit is set to one after the message number has been written to bits [7:0] MESSAGE_NUMBER. IF1 register set will be write protected. The bit is cleared after read/write action has been finished.	RW	0
14	DMAACTIVE	Activation of DMA feature for subsequent internal IF1 update 0: DMA request line is independent of IF1 activities. 1: DMA is requested after completed transfer between IF1 register set and message RAM. The DMA request remains active until the first read or write to one of the IF1 registers; an exception is a write to MESSAGE_NUMBER when DMAACTIVE is one. Note: Due to the auto reset feature of the DMAACTIVE bit, this bit has to be set for each subsequent DMA cycle separately.	RW	0
13:8	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
7:0	MESSAGE_NUMBER	Number of message object in message RAM which is used for data transfer 0x00: Invalid message number 0x01-0x80: Valid message numbers 0x81-0xFF: Invalid message numbers	RW	0x1

Table 24-820. Register Call Summary for Register DCAN_IF1CMD

DCAN

- Interrupt Functionality: [0]
- DMA Functionality: [1]
- Debug/Suspend Mode: [2]
- Transmission of Messages in Event Driven CAN Communication: [3]
- Updating a Transmit Object: [4]
- Changing a Transmit Object: [5] [6]
- Reading Received Messages: [7]
- Requesting New Data for a Receive Object: [8]
- Message Interface Register Sets: [9] [10]
- Message Interface Register Sets 1 and 2: [11] [12]
- DCAN Register Summary: [13]
- DCAN Register Description: [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26]

Table 24-821. DCAN_IF1MSK

Address Offset	0x0000 0104		
Physical Address	0x4AE3 C104 0x4848 0104	**Instance**	DCAN1 DCAN2
Description	IF1 Mask Register The bits of the IF1/IF2 mask registers mirror the mask bits of a message object. The function of the relevant message objects bits is described in Section 24.10.4.11.1 *Structure of Message Objects*. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

M XTD	M DIR	RESERVED	MSK

Bits	Field Name	Description	Type	Reset
31	MXTD	Mask Extended Identifier 0: The extended identifier bit (IDE) has no effect on the acceptance filtering. 1: The extended identifier bit (IDE) is used for acceptance filtering. When 11-bit ("standard") identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.	RW	1
30	MDIR	Mask Message Direction 0: The message direction bit (Dir) has no effect on the acceptance filtering. 1: The message direction bit (Dir) is used for acceptance filtering.	RW	1
29	RESERVED	This bit is always read as 1. Writes have no effect.	R	1
28:0	MSK	Identifier Mask 0: The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1: The corresponding bit in the identifier of the message object is used for acceptance filtering.	RW	0x1FFF FFFF

Table 24-822. Register Call Summary for Register DCAN_IF1MSK

DCAN

- DCAN Register Summary: [0]

Table 24-823. DCAN_IF1ARB

Address Offset	0x0000 0108		
Physical Address	0x4AE3 C108 0x4848 0108	**Instance**	DCAN1 DCAN2
Description	IF1 arbitration register The Arbitration bits ID[28:0], XTD, and DIR are used to define the identifier and type of outgoing messages and (together with the mask bits MSK[28:0], MXTD, and MDIR) for acceptance filtering of incoming messages. A received message is stored into the valid message object with matching identifier and Direction = receive (data frame) or Direction = transmit (remote frame). Extended frames can be stored only in message objects with XTD = 1, standard frames in message objects with XTD = 0. If a received message (data frame or remote frame) matches more than one valid message objects, it is stored into the one with the lowest message number. The bits of the IF1/IF2 arbitration registers mirror the arbitration bits of a message object. The function of the relevant message objects bits is described in Section 24.10.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28:0
MSGVAL	XTD	DIR	ID

Bits	Field Name	Description	Type	Reset
31	MSGVAL	Message valid 0: The message object is ignored by the message handler. 1: The message object is to be used by the message handler. The software should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit INIT in the DCAN_CTL. This bit must also be reset if the messages object is no longer required.	RW	0
30	XTD	Extended identifier 0: The 11-bit (“standard”) Identifier is used for this message object. 1: The 29-bit (“extended”) Identifier is used for this message object.	RW	0
29	DIR	Message direction 0: Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1: Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).	RW	0
28:0	ID	Message identifier ID[28:0]: 29-bit identifier (extended frame) ID[28:18]: 11-bit identifier (standard frame)	RW	0x0000 0000

Table 24-824. Register Call Summary for Register DCAN_IF1ARB

DCAN

- Configuration of a Single Receive Object for Data Frames: [0]
- DCAN Register Summary: [1]

Table 24-825. DCAN_IF1MCTL

Address Offset	0x0000 010C		
Physical Address	0x4AE3 C10C 0x4848 010C	**Instance**	DCAN1 DCAN2
Description	IF1 Message Control Register The bits of the IF1/IF2 message control registers mirror the message control bits of a message object. The function of the relevant message objects bits is described in Section 24.10.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31:16	15	14	13	12	11	10	9	8	7	6:4	3:0
RESERVED	NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB	RESERVED	DLC

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000

Bits	Field Name	Description	Type	Reset
15	NEWDAT	New data 0: No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the software. 1: The message handler or the software has written new data into the data portion of this message object.	RW	0
14	MSGLST	Message lost (only valid for message objects with direction = receive) 0: No message lost since the last time when this bit was reset by the software. 1: The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.	RW	0
13	INTPND	Interrupt pending 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt. The Interrupt Identifier in DCAN_INT will point to this message object if there is no other interrupt source with higher priority.	RW	0
12	UMASK	Use acceptance mask 0: Mask ignored 1: Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering If the UMASK bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.	RW	0
11	TXIE	Transmit interrupt enable 0: IntPnd will not be triggered after the successful transmission of a frame. 1: IntPnd will be triggered after the successful transmission of a frame.	RW	0
10	RXIE	Receive interrupt enable 0: IntPnd will not be triggered after the successful reception of a frame. 1: IntPnd will be triggered after the successful reception of a frame.	RW	0
9	RMTEN	Remote enable 0: At the reception of a remote frame, TxRqst is not changed. 1: At the reception of a remote frame, TxRqst is set.	RW	0
8	TXRQST	Transmit request 0: This message object is not waiting for a transmission. 1: The transmission of this message object is requested and is not yet done.	RW	0
7	EOB	End of Block 0: The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. 1: The message object is a single message object or the last message object in a FIFO Buffer Block. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to 1.	RW	0
6:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0

Bits	Field Name	Description	Type	Reset
3:0	DLC	Data length code 0-8: Data frame has 0-8 data bytes. 9-15 Data frame has 8 data bytes. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.	RW	0x0

Table 24-826. Register Call Summary for Register DCAN_IF1MCTL

DCAN

- Reading Received Messages: [0]
- DCAN Register Summary: [1]
- DCAN Register Description: [2] [3] [4] [5] [6] [7]

Table 24-827. DCAN_IF1DATA

Address Offset	0x0000 0110		
Physical Address	0x4AE3 C110 0x4848 0110	**Instance**	DCAN1 DCAN2
Description	IF1 Data A Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							

Bits	Field Name	Description	Type	Reset
31:24	DATA_3	Data byte 3	RW	0x0
23:16	DATA_2	Data byte 2	RW	0x0
15:8	DATA_1	Data byte 1	RW	0x0
7:0	DATA_0	Data byte 0	RW	0x0

Table 24-828. Register Call Summary for Register DCAN_IF1DATA

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 24-829. DCAN_IF1DATB

Address Offset	0x0000 0114		
Physical Address	0x4AE3 C114 0x4848 0114	**Instance**	DCAN1 DCAN2
Description	IF1 Data B Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DATA_7	DATA_6	DATA_5	DATA_4

Bits	Field Name	Description	Type	Reset
31:24	DATA_7	Data byte 7	RW	0x0
23:16	DATA_6	Data byte 6	RW	0x0
15:8	DATA_5	Data byte 5	RW	0x0
7:0	DATA_4	Data byte 4	RW	0x0

Table 24-830. Register Call Summary for Register DCAN_IF1DATB

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 24-831. DCAN_IF2CMD

Address Offset	0x0000 0120		
Physical Address	0x4AE3 C120 0x4848 0120	**Instance**	DCAN1 DCAN2
Description	IF2 Command Register The IF2 Command Register (DCAN_IF2CMD) configure and initiate the transfer between the IF2 register set and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the software writes the message number to bits [7:0] MESSAGE_NUMBER. With this write operation, the BUSY bit is automatically set to 1 to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the BUSY bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the software writes to both DCAN_IF1CMD/DCAN_IF2CMD consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While BUSY bit is one, IF1/IF2 register sets are write protected. For debug support, the auto clear functionality of the IF1/IF2 command registers (clear of DMAACTIVE flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] MESSAGE_NUMBER, the message handler may access an implemented (valid) message object instead.		
Type	RW		

31:24	23	22	21	20	19	18	17	16	15	14	13:8	7:0
RESERVED	WR_RD	MASK	ARB	CONTROL	CLRINTPND	TXRQST_NEWDAT	DATA_A	DATA_B	BUSY	DMAACTIVE	RESERVED	MESSAGE_NUMBER

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
23	WR_RD	Write/Read 0: Direction = Read: Transfer direction is from the message object addressed by MESSAGE_NUMBER to the IF2 register set. 1: Direction = Write: Transfer direction is from the IF2 register set to the message object addressed by MESSAGE_NUMBER.	RW	0

Bits	Field Name	Description	Type	Reset
22	MASK	Access mask bits 0: Mask bits will not be changed 1: Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by MESSAGE_NUMBER to the IF2 register set. 1: Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF2 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
21	ARB	Access arbitration bits 0: Arbitration bits will not be changed 1: Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF2 register set. 1: Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF2 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
20	CONTROL	Access control bits 0: Control bits will not be changed 1: Direction = Read: The message control bits will be transferred from the message object addressed by MESSAGE_NUMBER to the IF2 register set. 1: Direction = Write: The message control bits will be transferred from the IF2 registerset to the message object addressed by MESSAGE_NUMBER. If the TXRQST_NEWDAT bit in this register(Bit [18]) is set, the TXRQST/ NEWDAT bits in the DCAN_IF1MCTL/ DCAN_IF2MCTL will be ignored.	RW	0
19	CLRINTPND	Clear interrupt pending bit 0: IntPnd bit will not be changed 1: Direction = Read: Clears IntPnd bit in the message object. 1: Direction = Write: This bit is ignored. Copying of IntPnd flag from IF2 Registers to message RAM can only be controlled by the CONTROL flag (Bit [20]).	RW	0
18	TXRQST_NEWDAT	Access transmission request bit 0: Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the CONTROL bit. 1: Direction = Read: Clears NewDat bit in the message object. 1: Direction = Write: Sets TxRqst/NewDat in message object. Note: If a CAN transmission is requested by setting TXRQST_NEWDAT in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in DCAN_IF1MCTL/DCAN_IF2MCTL. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the DCAN_IF1MCTL/DCAN_IF2MCTL always reflect the status before resetting them.	RW	0

Bits	Field Name	Description	Type	Reset
17	DATA_A	Access Data Bytes 0-3 0: Data Bytes 0-3 will not be changed. 1: Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the MESSAGE_NUMBER to the corresponding IF2 registerset. 1: Direction = Write: The data bytes 0-3 will be transferred from the IF2 registerset to the message object addressed by the MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0
16	DATA_B	Access Data Bytes 4-7 0: Data Bytes 4-7 will not be changed. 1: Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF2 registerset. 1: Direction = Write: The data bytes 4-7 will be transferred from the IF2 registerset to the message object addressed by MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0
15	BUSY	Busy flag 0: No transfer between IF2 register set and message RAM is in progress. 1: Transfer between IF2 register set and message RAM is in progress. This bit is set to one after the message number has been written to bits [7:0] MESSAGE_NUMBER. IF2 register set will be write protected. The bit is cleared after read/write action has been finished.	RW	0
14	DMAACTIVE	Activation of DMA feature for subsequent internal IF2 update 0: DMA request line is independent of IF2 activities. 1: DMA is requested after completed transfer between IF2 register set and message RAM. The DMA request remains active until the first read or write to one of the IF2 registers; an exception is a write to MESSAGE_NUMBER when DMAACTIVE is one. Note: Due to the auto reset feature of the DMAACTIVE bit, this bit has to be set for each subsequent DMA cycle separately.	RW	0
13:8	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
7:0	MESSAGE_NUMBER	Number of message object in message RAM which is used for data transfer 0x00: Invalid message number 0x01-0x80: Valid message numbers 0x81-0xFF: Invalid message numbers	RW	0x1

Table 24-832. Register Call Summary for Register DCAN_IF2CMD

DCAN

- Interrupt Functionality: [0]
- DMA Functionality: [1]
- Debug/Suspend Mode: [2]
- Transmission of Messages in Event Driven CAN Communication: [3]
- Updating a Transmit Object: [4]
- Changing a Transmit Object: [5] [6]
- Reading Received Messages: [7]
- Requesting New Data for a Receive Object: [8]
- Message Interface Register Sets: [9] [10]
- Message Interface Register Sets 1 and 2: [11] [12]
- DCAN Register Summary: [13]
- DCAN Register Description: [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26]

Table 24-833. DCAN_IF2MSK

Address Offset	0x0000 0124		
Physical Address	0x4AE3 C124 0x4848 0124	**Instance**	DCAN1 DCAN2
Description	IF2 Mask Register The bits of the IF1/IF2 mask registers mirror the mask bits of a message object. The function of the relevant message objects bits is described in Section 24.10.4.11.1 *Structure of Message Objects*. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28–0
MXTD	MDIR	RESERVED	MSK

Bits	Field Name	Description	Type	Reset
31	MXTD	Mask Extended Identifier 0: The extended identifier bit (IDE) has no effect on the acceptance filtering. 1: The extended identifier bit (IDE) is used for acceptance filtering. When 11-bit ("standard") identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.	RW	1
30	MDIR	Mask Message Direction 0: The message direction bit (Dir) has no effect on the acceptance filtering. 1: The message direction bit (Dir) is used for acceptance filtering.	RW	1
29	RESERVED	This bit is always read as 1. Writes have no effect.	R	1
28:0	MSK	Identifier Mask 0: The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1: The corresponding bit in the identifier of the message object is used for acceptance filtering.	RW	0x1FFF FFFF

Table 24-834. Register Call Summary for Register DCAN_IF2MSK

DCAN

- DCAN Register Summary: [0]

Table 24-835. DCAN_IF2ARB

Address Offset	0x0000 0128		
Physical Address	0x4AE3 C128 0x4848 0128	**Instance**	DCAN1 DCAN2
Description	IF2 arbitration register The Arbitration bits ID[28:0], XTD, and DIR are used to define the identifier and type of outgoing messages and (together with the mask bits MSK[28:0], MXTD, and MDIR) for acceptance filtering of incoming messages. A received message is stored into the valid message object with matching identifier and Direction = receive (data frame) or Direction = transmit (remote frame). Extended frames can be stored only in message objects with Xtd = 1, standard frames in message objects with Xtd = 0. If a received message (data frame or remote frame) matches more than one valid message objects, it is stored into the one with the lowest message number. The bits of the IF1/IF2 arbitration registers mirror the arbitration bits of a message object. The function of the relevant message objects bits is described in Section 24.10.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL	XTD	DIR	ID																												

Bits	Field Name	Description	Type	Reset
31	MSGVAL	Message valid 0: The message object is ignored by the message handler. 1: The message object is to be used by the message handler. The software should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit INIT in the DCAN_CTL. This bit must also be reset if the messages object is no longer required.	RW	0
30	XTD	Extended identifier 0: The 11-bit (“standard”) Identifier is used for this message object. 1: The 29-bit (“extended”) Identifier is used for this message object.	RW	0
29	DIR	Message direction 0: Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1: Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).	RW	0
28:0	ID	Message identifier ID[28:0]: 29-bit identifier (extended frame) ID[28:18]: 11-bit identifier (standard frame)	RW	0x000 0000

Table 24-836. Register Call Summary for Register DCAN_IF2ARB

DCAN

- Configuration of a Single Receive Object for Data Frames: [0]
- DCAN Register Summary: [1]

Table 24-837. DCAN_IF2MCTL

Address Offset	0x0000 012C		
Physical Address	0x4AE3 C12C 0x4848 012C	**Instance**	DCAN1 DCAN2
Description	IF2 Message Control Register The bits of the IF1/IF2 message control registers mirror the message control bits of a message object. The function of the relevant message objects bits is described in Section 24.10.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB	RESERVED			DLC			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	NEWDAT	New data 0: No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the software. 1: The message handler or the software has written new data into the data portion of this message object.	RW	0
14	MSGLST	Message lost (only valid for message objects with direction = receive) 0: No message lost since the last time when this bit was reset by the software. 1: The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.	RW	0
13	INTPND	Interrupt pending 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt. The Interrupt Identifier in DCAN_INT will point to this message object if there is no other interrupt source with higher priority.	RW	0
12	UMASK	Use acceptance mask 0: Mask ignored 1: Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.	RW	0
11	TXIE	Transmit interrupt enable 0: IntPnd will not be triggered after the successful transmission of a frame. 1: IntPnd will be triggered after the successful transmission of a frame.	RW	0

Bits	Field Name	Description	Type	Reset
10	RXIE	Receive interrupt enable 0: IntPnd will not be triggered after the successful reception of a frame. 1: IntPnd will be triggered after the successful reception of a frame.	RW	0
9	RMTEN	Remote enable 0: At the reception of a remote frame, TxRqst is not changed. 1: At the reception of a remote frame, TxRqst is set.	RW	0
8	TXRQST	Transmit request 0: This message object is not waiting for a transmission. 1: The transmission of this message object is requested and is not yet done.	RW	0
7	EOB	End of Block 0: The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. 1: The message object is a single message object or the last message object in a FIFO Buffer Block. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.	RW	0
6:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
3:0	DLC	Data length code 0-8: Data frame has 0-8 data bytes. 9-15 Data frame has 8 data bytes. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.	RW	0x0

Table 24-838. Register Call Summary for Register DCAN_IF2MCTL

DCAN

- Reading Received Messages: [0]
- DCAN Register Summary: [1]
- DCAN Register Description: [2] [3] [4]

Table 24-839. DCAN_IF2DATA

Address Offset	0x0000 0130		
Physical Address	0x4AE3 C130 0x4848 0130	**Instance**	DCAN1 DCAN2
Description	IF2 Data A Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							

Bits	Field Name	Description	Type	Reset
31:24	DATA_3	Data byte 3	RW	0x0
23:16	DATA_2	Data byte 2	RW	0x0
15:8	DATA_1	Data byte 1	RW	0x0
7:0	DATA_0	Data byte 0	RW	0x0

Table 24-840. Register Call Summary for Register DCAN_IF2DATA

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 24-841. DCAN_IF2DATB

Address Offset	0x0000 0134		
Physical Address	0x4AE3 C134 0x4848 0134	**Instance**	DCAN1 DCAN2
Description	IF2 Data B Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_7								DATA_6								DATA_5								DATA_4							

Bits	Field Name	Description	Type	Reset
31:24	DATA_7	Data byte 7	RW	0x0
23:16	DATA_6	Data byte 6	RW	0x0
15:8	DATA_5	Data byte 5	RW	0x0
7:0	DATA_4	Data byte 4	RW	0x0

Table 24-842. Register Call Summary for Register DCAN_IF2DATB

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 24-843. DCAN_IF3OBS

Address Offset	0x0000 0140		
Physical Address	0x4AE3 C140 0x4848 0140	**Instance**	DCAN1 DCAN2

Table 24-843. DCAN_IF3OBS (continued)

Description	IF3 Observation Register The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by software (Additional information can be found in Section 24.10.4.11.1 *Structure of Message Objects*). The observation flags (Bits [4:0]) are used to determine, which data sections of the IF3 interface register set have to be read in order to complete a DMA read cycle. After all marked data sections are read, the DCAN is enabled to update the IF3 interface register set with new data. Any access order of single bytes or half-words is supported. When using byte or half-word accesses, a data section is marked as completed, if all bytes are read. NOTE: If IF3 Update Enable is used and no Observation flag is set, the corresponding message objects will be copied to IF3 without activating the DMA request line and without waiting for DMA read accesses. A write access to this register aborts a pending DMA cycle by resetting the DMA line and enables updating of IF3 interface register set with new data. To avoid data inconsistency, the DMA controller should be disabled before reconfiguring IF3 observation register. The status of the current read-cycle can be observed via status flags (Bits [12:8]).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IF3_UPD	RESERVED		IF3_SDB	IF3_SDA	IF3_SC	IF3_SA	IF3_SM	RESERVED			DATAB	DATAA	CTRL	ARB	MASK

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	IF3_UPD	IF3 Update Data 0: No new data has been loaded since last IF3 read. 1: New data has been loaded since last IF3 read.	R	0
14:13	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
12	IF3_SDB	IF3 Status of Data B read access 0: All Data B bytes are already read out, or are not marked to be read. 1: Data B section has still data to be read out.	R	0
11	IF3_SDA	IF3 Status of Data A read access 0: All Data A bytes are already read out, or are not marked to be read. 1: Data A section has still data to be read out.	R	0
10	IF3_SC	IF3 Status of control bits read access 0: All control section bytes are already read out, or are not marked to be read. 1: Control section has still data to be read out.	R	0
9	IF3_SA	IF3 Status of Arbitration data read access 0: All Arbitration data bytes are already read out, or are not marked to be read. 1: Arbitration section has still data to be read out.	R	0
8	IF3_SM	IF3 Status of Mask data read access 0: All mask data bytes are already read out, or are not marked to be read. 1: Mask section has still data to be read out.	R	0
7:5	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
4	DATAB	Data B read observation 0: Data B section has not to be read. 1: Data B section has to be read to enable next IF3 update.	RW	0

Bits	Field Name	Description	Type	Reset
3	DATAA	Data A read observation 0: Data A section has not to be read. 1: Data A section has to be read to enable next IF3 update.	RW	0
2	CTRL	Ctrl read observation 0: Ctrl section has not to be read. 1: Ctrl section has to be read to enable next IF3 update.	RW	0
1	ARB	Arbitration data read observation 0: Arbitration data has not to be read. 1: Arbitration data has to be read to enable next IF3 update.	RW	0
0	MASK	Mask data read observation 0: Mask data has not to be read. 1: Mask data has to be read to enable next IF3 update.	RW	0

Table 24-844. Register Call Summary for Register DCAN_IF3OBS

DCAN

- IF3 Register Set: [0]
- DCAN Register Summary: [1]

Table 24-845. DCAN_IF3MSK

Address Offset	0x0000 0144		
Physical Address	0x4AE3 C144 0x4848 0144	**Instance**	DCAN1 DCAN2
Description	IF3 Mask Register		
Type	RW		

31	30	29	28-0
MXTD	MDIR	RESERVED	MSK

Bits	Field Name	Description	Type	Reset
31	MXTD	Mask Extended Identifier 0: The extended identifier bit (IDE) has no effect on the acceptance filtering. 1: The extended identifier bit (IDE) is used for acceptance filtering. When 11-bit (“standard”) identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.	R	1
30	MDIR	Mask Message Direction 0: The message direction bit (Dir) has no effect on the acceptance filtering. 1: The message direction bit (Dir) is used for acceptance filtering.	R	1
29	RESERVED	These bits are always read as 1. Writes have no effect.	R	1

Bits	Field Name	Description	Type	Reset
28:0	MSK	Identifier Mask 0: The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1: The corresponding bit in the identifier of the message object is used for acceptance filtering.	RW	0x1FFF FFFF

Table 24-846. Register Call Summary for Register DCAN_IF3MSK

DCAN

- DCAN Register Summary: [0]

Table 24-847. DCAN_IF3ARB

Address Offset	0x0000 0148		
Physical Address	0x4AE3 C148 0x4848 0148	**Instance**	DCAN1 DCAN2
Description	IF3 Arbitration Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL	XTD	DIR	ID																												

Bits	Field Name	Description	Type	Reset
31	MSGVAL	Message Valid 0: The message object is ignored by the message handler. 1: The message object is to be used by the message handler. The software should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit INIT in the DCAN_CTL. This bit must also be reset before the identifier ID[28:0], the control bits Xtd, Dir or DLC[3:0] are modified, or if the messages object is no longer required.	R	0
30	XTD	Extended Identifier 0: The 11-bit (“standard”) Identifier is used for this message object. 1: The 29-bit (“extended”) Identifier is used for this message object.	R	0
29	DIR	Message Direction 0: Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1: Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).	R	0
28:0	ID	Message Identifier ID[28:0]: 29-bit Identifier (“extended frame”) ID[28:18]: 11-bit Identifier (“standard frame”)	R	0x0000 0000

Table 24-848. Register Call Summary for Register DCAN_IF3ARB

DCAN

- Configuration of a Single Receive Object for Data Frames: [0]
- DCAN Register Summary: [1]

Table 24-849. DCAN_IF3MCTL

Address Offset	0x0000 014C		
Physical Address	0x4AE3 C14C 0x4848 014C	**Instance**	DCAN1 DCAN2
Description	IF3 Message Control Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB	RESERVED			DLC			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	NEWDAT	New Data 0: No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the software. 1: The message handler or the software has written new data into the data portion of this message object.	R	0
14	MSGLST	Message Lost (only valid for message objects with direction = receive) 0: No message lost since the last time when this bit was reset by the software. 1: The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.	R	0
13	INTPND	Interrupt Pending 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt. The Interrupt Identifier in DCAN_INT will point to this message object if there is no other interrupt source with higher priority.	R	0
12	UMASK	Use Acceptance Mask 0: Mask ignored 1: Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering If the UMASK bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.	R	0
11	TXIE	Transmit Interrupt enable 0: IntPnd will not be triggered after the successful transmission of a frame. 1: IntPnd will be triggered after the successful transmission of a frame.	R	0

Bits	Field Name	Description	Type	Reset
10	RXIE	Receive Interrupt enable 0: IntPnd will not be triggered after the successful reception of a frame. 1: IntPnd will be triggered after the successful reception of a frame.	R	0
9	RMTEN	Remote enable 0: At the reception of a remote frame, TxRqst is not changed. 1: At the reception of a remote frame, TxRqst is set.	R	0
8	TXRQST	Transmit Request 0: This message object is not waiting for a transmission. 1: The transmission of this message object is requested and is not yet done.	R	0
7	EOB	End of Block 0: The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. 1: The message object is a single message object or the last message object in a FIFO Buffer Block. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.	R	0
6:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
3:0	DLC	Data Length Code 0-8: Data frame has 0-8 data bits. 9-15: Data frame has 8 data bytes. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.	R	0x0

Table 24-850. Register Call Summary for Register DCAN_IF3MCTL

DCAN

- Reading Received Messages: [0]
- DCAN Register Summary: [1]

Table 24-851. DCAN_IF3DATA

Address Offset	0x0000 0150		
Physical Address	0x4AE3 C150 0x4848 0150	**Instance**	DCAN1 DCAN2
Description	IF3 Data A The data bytes of CAN messages are stored in the IF3 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							

Bits	Field Name	Description	Type	Reset
31: 24	DATA_3	Data byte 3	R	0x0
23:16	DATA_2	Data byte 2	R	0x0

Bits	Field Name	Description	Type	Reset
15:8	DATA_1	Data byte 1	R	0x0
7:0	DATA_0	Data byte 0	R	0x0

Table 24-852. Register Call Summary for Register DCAN_IF3DATA

DCAN

- DCAN Register Summary: [0]

Table 24-853. DCAN_IF3DATB

Address Offset	0x0000 0154		
Physical Address	0x4AE3 C154 0x4848 0154	**Instance**	DCAN1 DCAN2
Description	IF3 Data B The data bytes of CAN messages are stored in the IF3 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.		
Type	R		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
DATA_7	DATA_6	DATA_5	DATA_4

Bits	Field Name	Description	Type	Reset
31:24	DATA_7	Data byte 7	R	0x0
23:16	DATA_6	Data byte 6	R	0x0
15:8	DATA_5	Data byte 5	R	0x0
7:0	DATA_4	Data byte 4	R	0x0

Table 24-854. Register Call Summary for Register DCAN_IF3DATB

DCAN

- DCAN Register Summary: [0]

Table 24-855. DCAN_IF3UPD12

Address Offset	0x0000 0160		
Physical Address	0x4AE3 C160 0x4848 0160	**Instance**	DCAN1 DCAN2
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
IF3UPDEN			

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 1-32 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 24-856. Register Call Summary for Register DCAN_IF3UPD12

DCAN

- IF3 Register Set: [0]
- DCAN Register Summary: [1]

Table 24-857. DCAN_IF3UPD34

Address Offset	0x0000 0164		
Physical Address	0x4AE3 C164 0x4848 0164	**Instance**	DCAN1 DCAN2
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 33-64 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 24-858. Register Call Summary for Register DCAN_IF3UPD34

DCAN

- DCAN Register Summary: [0]

Table 24-859. DCAN_IF3UPD56

Address Offset	0x0000 0168		
Physical Address	0x4AE3 C168 0x4848 0168	**Instance**	DCAN1 DCAN2
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 65-96 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 24-860. Register Call Summary for Register DCAN_IF3UPD56

DCAN

- DCAN Register Summary: [0]

Table 24-861. DCAN_IF3UPD78

Address Offset	0x0000 016C		
Physical Address	0x4AE3 C16C 0x4848 016C	**Instance**	DCAN1 DCAN2
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 97-128 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 24-862. Register Call Summary for Register DCAN_IF3UPD78

DCAN

- IF3 Register Set: [0]
- DCAN Register Summary: [1]

Table 24-863. DCAN_TIOC

Address Offset	0x0000 01E0		
Physical Address	0x4AE3 C1E0 0x4848 01E0	**Instance**	DCAN1 DCAN2
Description	TX I/O Control Register The CAN_TX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are only writable if INIT bit of the DCAN_CTL is set to 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PU	PD	OD	RESERVED	FUNC	DIR	OUT	IN

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
18	PU	CAN_TX pull up/pull down select. This bit is only active when CAN_TX is configured to be an input. 0: CAN_TX pull down is selected, when pull logic is active (PD = 0). 1: CAN_TX pull up is selected, when pull logic is active(PD = 0).	RW	0
17	PD	CAN_TX pull disable. This bit is only active when CAN_TX is configured to be an input. 0: CAN_TX pull is active 1: CAN_TX pull is disabled	RW	0
16	OD	CAN_TX open drain enable. This bit is only active when CAN_TX is configured to be in GIO mode (FUNC=0). 0: The CAN_TX pin is configured in push/pull mode. 1: The CAN_TX pin is configured in open drain mode. Forced to '0' if INIT bit of DCAN_CTL is reset.	RW	0
15:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x000
3	FUNC	CAN_TX function. This bit changes the function of the CAN_TX pin 0: CAN_TX pin is in GIO mode. 1: CAN_TX pin is in functional mode (as an output to transmit CAN data). Forced to Tx output of the CAN core, if INIT bit of DCAN_CTL is reset.	RW	0
2	DIR	CAN_TX data direction. This bit controls the direction of the CAN_TX pin when it is configured to be in GIO mode only (FUNC=0) 0: The CAN_TX pin is an input. 1: The CAN_TX pin is an output Forced to '1' if INIT bit of DCAN_CTL is reset.	RW	0
1	OUT	CAN_TX data out write. This bit is only active when CAN_TX pin is configured to be in GIO mode (FUNC = 0) and configured to be an output pin (DIR = 1). The value of this bit indicates the value to be output to the CAN_TX pin. 0: The CAN_TX pin is driven to logic low 1: The CAN_TX pin is driven to logic high Forced to 1 if INIT bit of DCAN_CTL is reset.	RW	0
0	IN	CAN_TX data in 0: The CAN_TX pin is at logic low 1: The CAN_TX pin is at logic high Note: When CAN_TX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module).	RW	-

Table 24-864. Register Call Summary for Register DCAN_TIOC

DCAN

- GPIO Support: [0]
- DCAN Register Summary: [1]

Table 24-865. DCAN_RIOC

Address Offset	0x0000 01E4		
Physical Address	0x4AE3 C1E4 0x4848 01E4	**Instance**	DCAN1 DCAN2
Description	RX I/O Control Register The CAN_RX pin of the DCAN_module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are only writable if INIT bit of the DCAN_CTL is set to 1.		
Type	RW		

31 ... 19	18	17	16	15 ... 4	3	2	1	0
RESERVED	PU	PD	OD	RESERVED	FUNC	DIR	OUT	IN

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
18	PU	CAN_RX pull up/pull down select. This bit is only active when CAN_RX is configured to be an input. 0: CAN_RX pull down is selected, when pull logic is active (PD = 0). 1: CAN_RX pull up is selected, when pull logic is active(PD = 0).	RW	0
17	PD	CAN_RX pull disable. This bit is only active when CAN_TX is configured to be an input. 0: CAN_RX pull is active 1: CAN_RX pull is disabled	RW	0
16	OD	CAN_RX open drain enable. This bit is only active when CAN_RX is configured to be in GIO mode (FUNC=0). 0: The CAN_RX pin is configured in push/pull mode. 1: The CAN_RX pin is configured in open drain mode. Forced to '0' if INIT bit of DCAN_CTL is reset.	RW	0
15:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x000
3	FUNC	CAN_RX function. This bit changes the function of the CAN_RX pin 0: CAN_RX pin is in GIO mode. 1: CAN_RX pin is in functional mode (as an input to receive CAN data). Forced to '1' if INIT bit of DCAN_CTL is reset.	RW	0
2	DIR	CAN_RX data direction. This bit controls the direction of the CAN_RX pin when it is configured to be in GIO mode only (FUNC=0) 0: The CAN_RX pin is an input. 1: The CAN_RX pin is an output Forced to '0' if INIT bit DCAN_CTL is reset.	RW	0

Bits	Field Name	Description	Type	Reset
1	OUT	CAN_RX data out write. This bit is only active when CAN_RX pin is configured to be in GIO mode (FUNC = 0) and configured to be an output pin (DIR = 1). The value of this bit indicates the value to be output to the CAN_RX pin. 0: The CAN_RX pin is driven to logic low 1: The CAN_RX pin is driven to logic high	RW	0
0	IN	CAN_RX data in 0: The CAN_RX pin is at logic low 1: The CAN_RX pin is at logic high	RW	-

Table 24-866. Register Call Summary for Register DCAN_RIOC

DCAN

- GPIO Support: [0]
- DCAN Register Summary: [1]

24.11 Gigabit Ethernet Switch (GMAC_SW)

This chapter describes the three-port gigabit switch ethernet subsystem in the device.

24.11.1 GMAC_SW Overview

The three-port gigabit ethernet switch subsystem (GMAC_SW) provides ethernet packet communication and can be configured as an ethernet switch. It provides the gigabit media independent interface (G/MII) in MII mode, reduced gigabit media independent interface (RGMII), reduced media independent interface (RMII), and the management data input output (MDIO) for physical layer device (PHY) management.

Figure 24-186 shows the GMAC_SW subsystem overview.

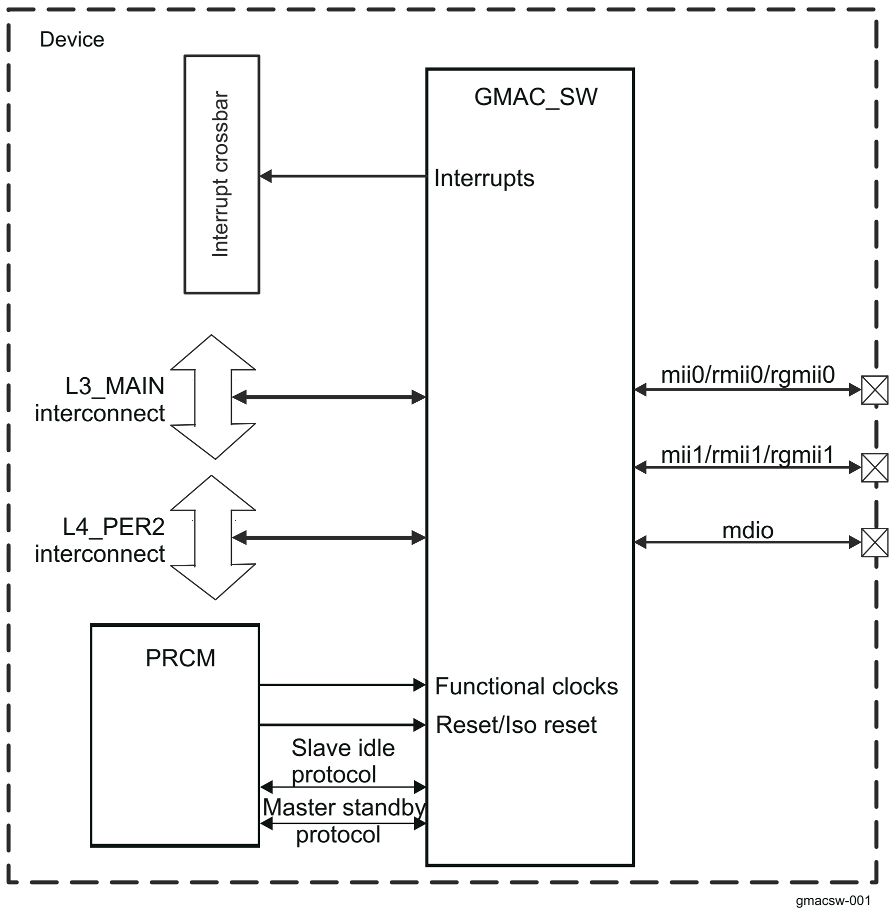


Figure 24-186. GMAC_SW Overview

24.11.1.1 Features

The GMAC_SW subsystem provides the following features:

- Two Ethernet ports (port 1 and port 2) with selectable RGMII, RMII, and G/MII (in MII mode only) interfaces plus internal Communications Port Programming Interface (CPPI 3.1) on port 0
- Synchronous 10/100/1000 Mbit operation
- Wire rate switching (802.1d)
- Non-blocking switch fabric
- Flexible logical FIFO-based packet buffer structure
- Four priority level Quality Of Service (QOS) support (802.1p)
- CPPI 3.1 compliant DMA controllers
- Support for Audio/Video Bridging (P802.1Qav/D6.0)

- Support for IEEE 1588 Clock Synchronization (2008 Annex D and Annex F)
  - Timing FIFO and time stamping logic embedded in the subsystem
- Device Level Ring (DLR) Support
- Energy Efficient Ethernet (EEE) support (802.3az)
- Flow Control Support (802.3x)
- Address Lookup Engine (ALE)
  - 1024 total address entries plus VLANs
  - Wire rate lookup
  - Host controlled time-based aging
  - Multiple spanning tree support (spanning tree per VLAN)
  - L2 address lock and L2 filtering support
  - MAC authentication (802.1x)
  - Receive-based or destination-based multicast and broadcast rate limits
  - MAC address blocking
  - Source port locking
  - OUI (Vendor ID) host accept/deny feature
  - Remapping of priority level of VLAN or ports
- VLAN support
  - 802.1Q compliant
    - Auto add port VLAN for untagged frames on ingress
    - Auto VLAN removal on egress and auto pad to minimum frame size
- Ethernet Statistics:
  - EtherStats and 802.3Stats Remote network Monitoring (RMON) statistics gathering (shared)
  - Programmable statistics interrupt mask when a statistic is above one half its 32-bit value
- Flow Control Support (802.3x)
- Digital loopback and FIFO loopback modes supported
- Maximum frame size 2016 bytes (2020 with VLAN)
- 8k (2048 × 32) internal CPPI buffer descriptor memory
- Management Data Input/Output (MDIO) module for PHY Management
- Programmable interrupt control with selected interrupt pacing
- Emulation support
- Programmable Transmit Inter Packet Gap (IPG)
- Reset isolation (switch function remains active even in case of all device resets except for POR pin reset and ICEPICK cold reset)
- Full duplex mode supported in 10/100/1000 Mbps. Half-duplex mode supported only in 10/100 Mbps.
- IEEE 802.3 gigabit Ethernet conformant

Note

G/MII interface is functional in MII mode only.

24.11.1.2

Terminology:

AVB	Audio Video Bridging
AVBTP	Audio Video Bridging Trasport Protocol
BMCA	Best Master Clock Algorithm
CFI	Canonical Format Indicator
CPPI	Communications Port Programming Interface
DLR	Device Level Ring
DSCP	Differentiated Services Code Point
EEE	Energy Efficient Ethernet
EMAC	Ethernet Media Access Control
EOP	End of Packet
EOQ	End of Queue
IPG	Inter-Packet Gap
LPI	Low Power Indicator
MDIO	Management Data Input/Output
MOF	Middle of Frame
PTP	Precision Time Protocol
RMON	Remote Monitoring
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
SCR	Switched Central Resource
SRP	Stream Reservation Protocol
TOS	Type of Service
VLAN	Virtual Local Area Network

24.11.2 GMAC_SW Environment

24.11.2.1 G/MII Interface

G/MII Interface can operate only in MII Mode.

- In MII Mode (100/10 Mbps): GMAC_SW operates in full-duplex and half-duplex.

The pin connections of the G/MII Interface is shown in Figure 24-187. The detailed description of the signals are listed in the following tables.

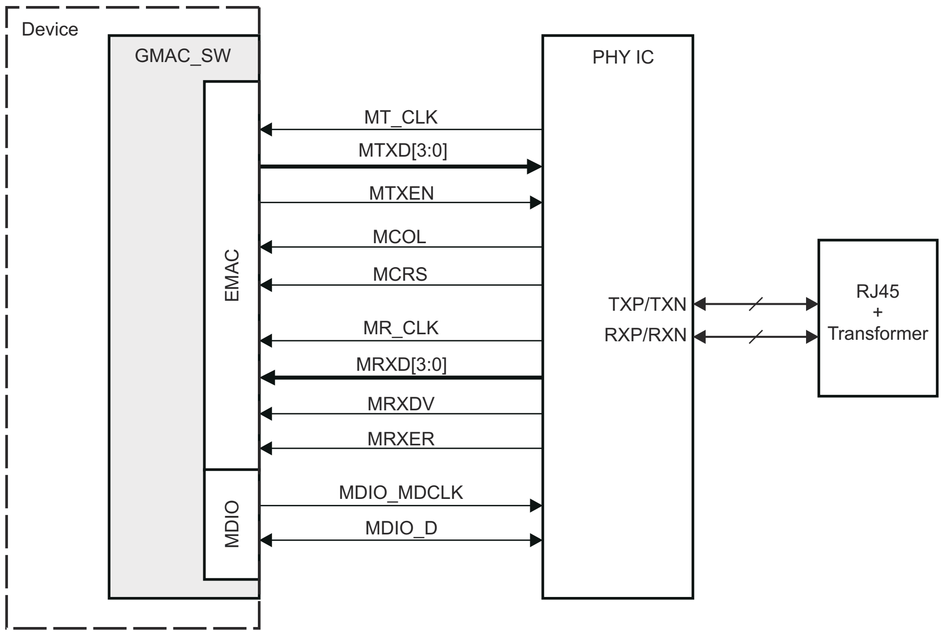


Figure 24-187. MII Interface Typical Application

Table 24-867. MII I/O Description

Signal	Device Pin(s)	I/O[1]	Description
MT_CLK	mii0_txclk mii1_txclk	I	The transmit clock is a continuous clock that provides the timing reference for transmit operations. The MTXD and MTXEN signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation.
MTXD[3:0]	mii0_txd[3:0] mii1_txd[3:0]	O	The transmit data pins are a collection of 4 data signals MTXD[3:0] comprising 4 bits of data. MTXD0 is the least-significant bit (LSB). The signals are synchronized by MT_CLK and valid only when MTXEN is asserted.
MTXEN	mii0_txen mii1_txen	O	The transmit enable signal indicates that the MTXD pins are generating 4bit data for use by the PHY. It is driven synchronously by MT_CLK.
MTXER	mii0_txer mii1_txer	O	Transmit data error. Used only for EEE to request PHY for low power transition.
MCOL	mii0_col mii1_col	I	In half-duplex operation, the MCOL pin is asserted by the PHY when it detects a collision on the network. It remains asserted while the collision condition persists. This signal is not necessarily synchronous to MT_CLK nor MR_CLK. In full-duplex operation, the MCOL pin is used for hardware transmit flow control. Asserting the MCOL pin will stop packet transmissions; packets in the process of being transmitted when MCOL is asserted will complete transmission. The MCOL pin should be held low if hardware transmit flow control is not used.
MCRS	mii0_crs mii1_crs	I	In half-duplex operation, the MCRS pin is asserted by the PHY when the network is not idle in either transmit or receive. The pin is de-asserted when both transmit and receive are idle. This signal is not necessarily synchronous to MT_CLK nor MR_CLK. In full-duplex operation, the MCRS pin should be held low.

Table 24-867. MII I/O Description (continued)

Signal	Device Pin(s)	I/O[(1)]	Description
MR_CLK	mii0_rxclk mii1_rxclk	I	The receive clock is a continuous clock that provides the timing reference for receive operations. The MRXD, MRXDV, and MRXER signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation.
MRXD	mii0_rxd[0:3] mii1_rxd[0:3]	I	The receive data pins are a collection of 4 data signals comprising 4 bits of data. MRXD0 is the least-significant bit (LSB).The signals are synchronized by MR_CLK and valid only when MRXDV is asserted.
MRXDV	mii0_rxdv mii1_rxdv	I	The receive data valid signal indicates that the MRXD pins are generating nibble data for use by the GMAC_SW. It is driven synchronously to MR_CLK.
MRXER	mii0_rxer mii1_rxer	I	Receive Data Error input
MDIO_MDCLK	mdio_mclk	O	Management data clock. The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO_D pin.
MDIO_D	mdio_d	I/O	MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

(1) I = Input; O = Output

24.11.2.2 RMII Interface

Figure 24-188 shows a device with integrated GMAC_SW and MDIO interfaced via a RMII connection in a typical system. The individual CPSW and MDIO signals for the RMII interface are summarized in Table 24-868.

For more information, refer to either the IEEE 802.3 standard or ISO/IEC 8802-3:2000(E).

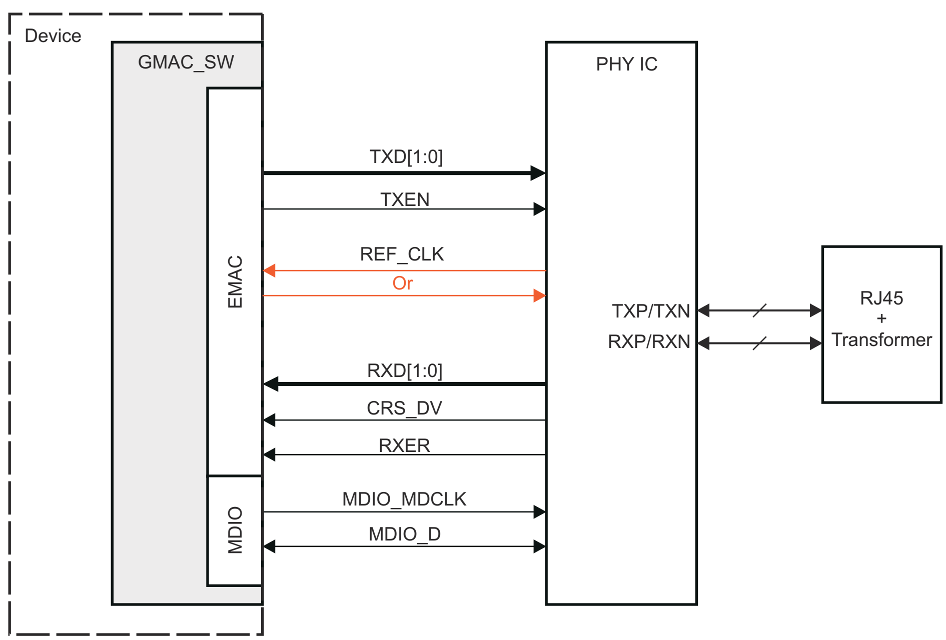


Figure 24-188. RMII Interface Typical Application

Table 24-868. RMII I/O Description

Signal	Device Pin(s)	I/O[(1)]	Description
TXD[1:0]	rmii0_txd[1:0] rmii1_txd[1:0]	O	Transmit data . The transmit data pins are a collection of 2 bits of data. TXD0 is the least-significant bit (LSB). The signals are synchronized by RMII_MHZ_50_CLK and valid only when TXEN is asserted.

Table 24-868. RMII I/O Description (continued)

Signal	Device Pin(s)	I/O[(1)]	Description
TXEN	rmii0_txen rmii1_txen	O	Transmit enable. The transmit enable signal indicates that the RMII_TXD pins are generating data for use by the PHY. RMII_TXEN is synchronous to RMII_MHZ_50_CLK
RMII0_MHZ_50_CLK RMII1_MHZ_50_CLK	RMII_MHZ_50_CLK	I/O	RMII reference clock. The reference clock is used to synchronize all RMII signals. RMII_MHZ_50_CLK must be continuous and fixed at 50 MHz.
RXD[1:0]	rmii0_rxd[1:0] rmii1_rxd[1:0]	I	Receive data. The receive data pins are a collection of 2 bits of data. RXD0 is the least-significant bit (LSB). The signals are synchronized by RMII_MHZ_50_CLK and valid only when CRS_DV is asserted and RXER is de-asserted.
CRS_DV	rmii0_crs rmii1_crs	I	Carrier sense/receive data valid. Multiplexed signal between carrier sense and receive data valid.
RXER	rmii0_rxer rmii1_rxer	I	Receive error. The receive error signal is asserted to indicate that an error was detected in the received frame.
MDIO_MDCLK	mdio_mclk	O	Management data clock (MDIO_MCLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO_D pin.
MDIO_D	mdio_d	I/O	MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

(1) I = Input; O = Output

24.11.2.3 RGMII Interface

Figure 24-189 shows a device with integrated CPSW and MDIO interfaced via a RGMII connection in a typical system. The individual CPSW and MDIO signals for the RGMII interface are summarized in Table 24-869.

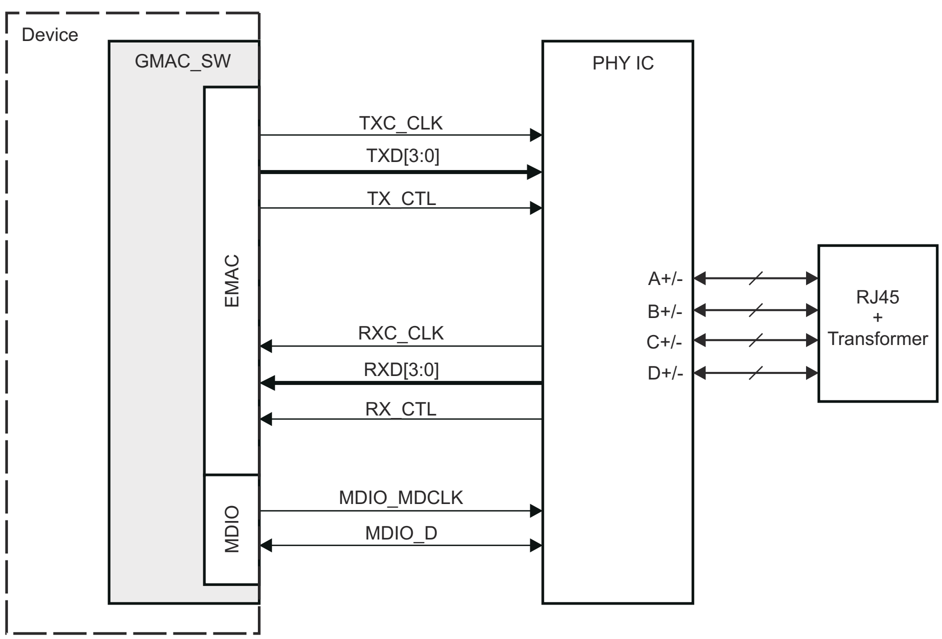


Figure 24-189. RGMII Interface Typical Application

Table 24-869. RGMII I/O Description

Signal	Device Pin(s)	I/O[(1)]	Description
TXD[3:0]	rgmii0_txd[3:0] rgmii1_txd[3:0]	O	The transmit data pins are a collection of 4 bits of data. TXD0 is the least-significant bit (LSB). The signals are valid only when TX_CTL is asserted.

Table 24-869. RGMII I/O Description (continued)

Signal	Device Pin(s)	I/O[1]	Description
TX_CTL	rgmii0_rxctl rgmii1_rxctl	O	Transmit Control/enable. The transmit enable signal indicates that the TXD pins are generating data for use by the PHY.
TXC_CLK	rgmii0_txc rgmii1_txc	O	The transmit reference clock. The clock is 2.5 MHz at 10 Mbps operation, 25 MHz at 100 Mbps operation, and 125 MHz at 1000 Mbps of operation.
RXD[3:0]	rgmii0_rxd[3:0] rgmii1_rxd[3:0]	I	The receive data pins are a collection of 4 bits of data. RXD0 is the least-significant bit (LSB). The signals are valid only when RX_CTL is asserted
RX_CTL	rgmii0_rxctl rgmii1_rxctl	I	The receive data valid/control signal indicates that the RXD pins are nibble data for use by the GMAC_SW.
RXC_CLK	rgmii0_rxc rgmii1_rxc	I	The receive clock is a continuous clock that provides the timing reference for receive operations. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation, 25 MHz at 100 Mbps operation, 125 MHz at 1000 Mbps of operation.
MDIO_MDCLK	mdio_mclk	O	Management data clock (MDIO_MCLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO_D	mdio_d	I/O	MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

(1) I = Input; O = Output

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see *Pad Configuration Registers* in *Control Module*.

24.11.3 GMAC_SW Integration

Figure 24-190 shows the integration of the GMAC_SW module in the device.

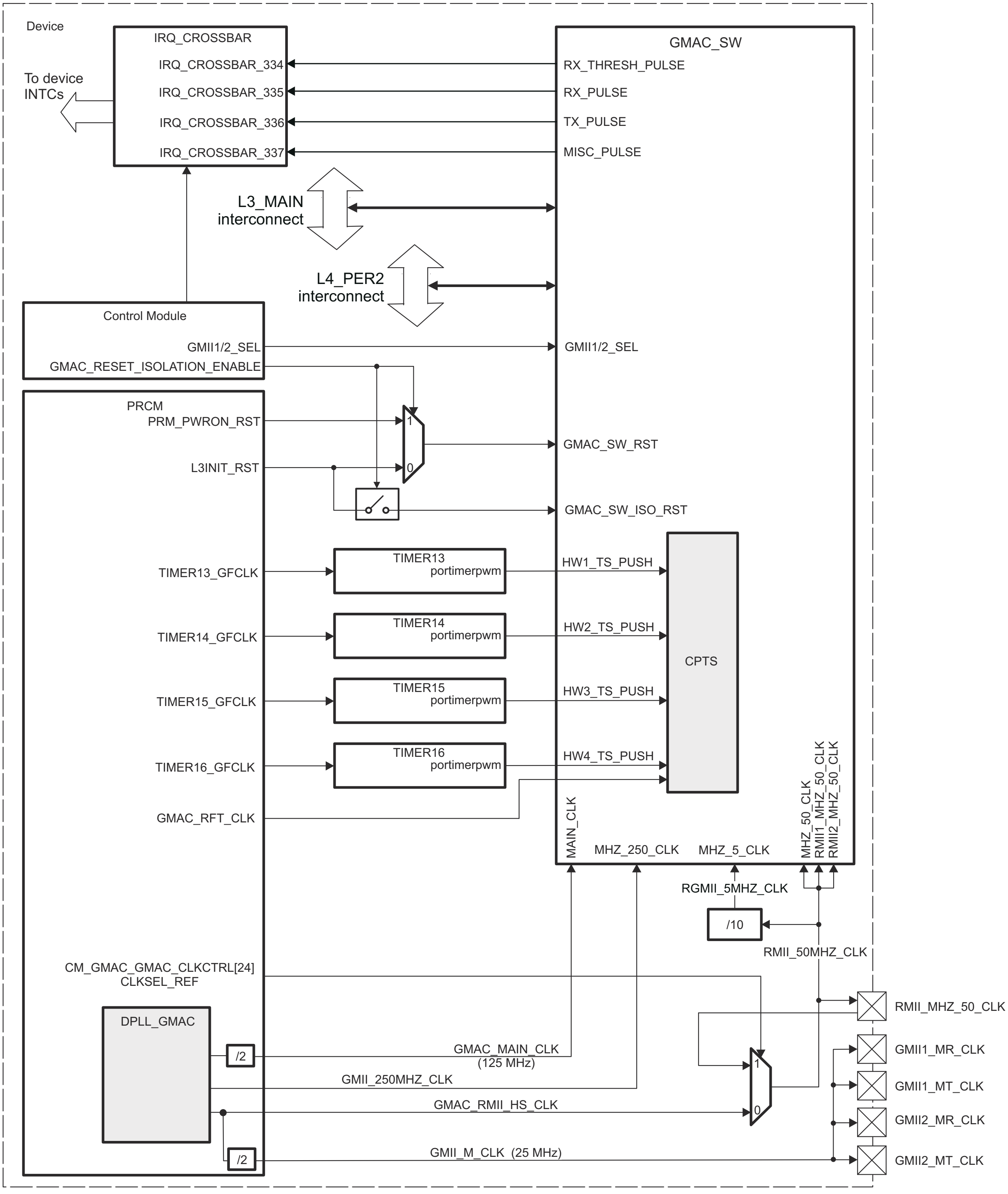


Figure 24-190. GMAC_SW Integration

Note

The device provides the option to receive or output the RMII reference clock (RMII_50MHZ_CLK) from/to external pin as shown in Figure 24-190. See the RMII_CLK_SETTING register bit in the *Control Module Register Manual*, and CLKSEL_REF in , *PRCM Register Manual* to configure the clocking option of RMII.

See *PRCM* for details about power, clock and reset options for GMAC_SW.

Table 24-870 through Table 24-872 summarize the integration of the GMAC_SW module in the device.

Table 24-870. GMAC_SW Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
GMAC_SW	PD_COREAON	Yes	L3_MAIN L4_PER2

Table 24-871. GMAC_SW Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GMAC_SW	MAIN_CLK	GMAC_MAIN_CLK	PRCM	Interface clock for the GMAC_SW module (125 MHz)
	MHZ_5_CLK	RGMII_5MHZ_CLK	PRCM	5-MHz RGMII clock
	MHZ_50_CLK	RMII_50MHZ_CLK	PRCM or RMII_MHZ_50_CLK pin	50-MHz RGMII clock
	MHZ_250_CLK	GMII_250MHZ_CLK	PRCM	250-MHz RGMII clock
	RMII1_MHZ_50_CLK	RMII_50MHZ_CLK	PRCM or RMII_MHZ_50_CLK pin	50-MHz RMII1 clock
	RMII2_MHZ_50_CLK	RMII_50MHZ_CLK	PRCM or RMII_MHZ_50_CLK pin	50-MHz RMII2 clock
	CPTS_RFT_CLK	GMAC_RFT_CLK	PRCM	IEEE 1588 clock
	GMII1_MR_CLK	GMII_M_CLK	PRCM or the external pin	MII1 receive clock
	GMII1_MT_CLK	GMII_M_CLK	PRCM or the external pin	MII1 transmit clock
	GMII2_MR_CLK	GMII_M_CLK	PRCM or the external pin	MII2 receive clock
	GMII2_MT_CLK	GMII_M_CLK	PRCM or the external pin	MII2 transmit clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GMAC_SW	GMAC_SW_RST	L3INIT_RST/ PRM_PWRON_RST	PRCM	GMAC_SW main reset
	GMAC_SW_ISO_RST	tied off/L3INIT_RST	PRCM	GMAC_SW isolate reset (must be enabled in Control module)

Table 24-872. GMAC_SW Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
GMAC_SW	RX_THRESH_PULSE	IRQ_CROSSBAR_334	-	Receive threshold interrupt
	RX_PULSE	IRQ_CROSSBAR_335	-	Receive packet completion interrupt
	TX_PULSE	IRQ_CROSSBAR_336	-	Transmit packet completion interrupt

Table 24-872. GMAC_SW Hardware Requests (continued)

MISC_PULSE	IRQ_CROSSBAR_337	-	Miscellaneous interrupt (SPF2_PEND, SPF1_PEND, EVNT_PEND,STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT)

Note

GMAC_SW has no default IRQ mappings through the IRQ_CROSSBAR. For GMAC_SW, the IRQ_CROSSBAR module must be configured prior to unmask interrupts in the interrupt controller(s).
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For the description of the interrupt source, see Section 24.11.4.5, *Interrupt Functionality*.

24.11.4 GMAC_SW Functional Description

The 3-port switch (GMAC_SW) Ethernet subsystem modules are compliant to the IEEE Std 802.3 Specification.GMAC_SW is shown in Figure 24-191.

24.11.4.1 Functional Block Diagram

The GMAC_SW subsystem consists of:

- CPSW_3G which contains two G/MII interfaces
- Two RGMII interface modules
- Two RMII interface modules
- One MDIO interface module
- One Interrupt Controller module
- One CPPI interface
- Local CPPI memory of size 8 KiB
- Two Static Packet Filters

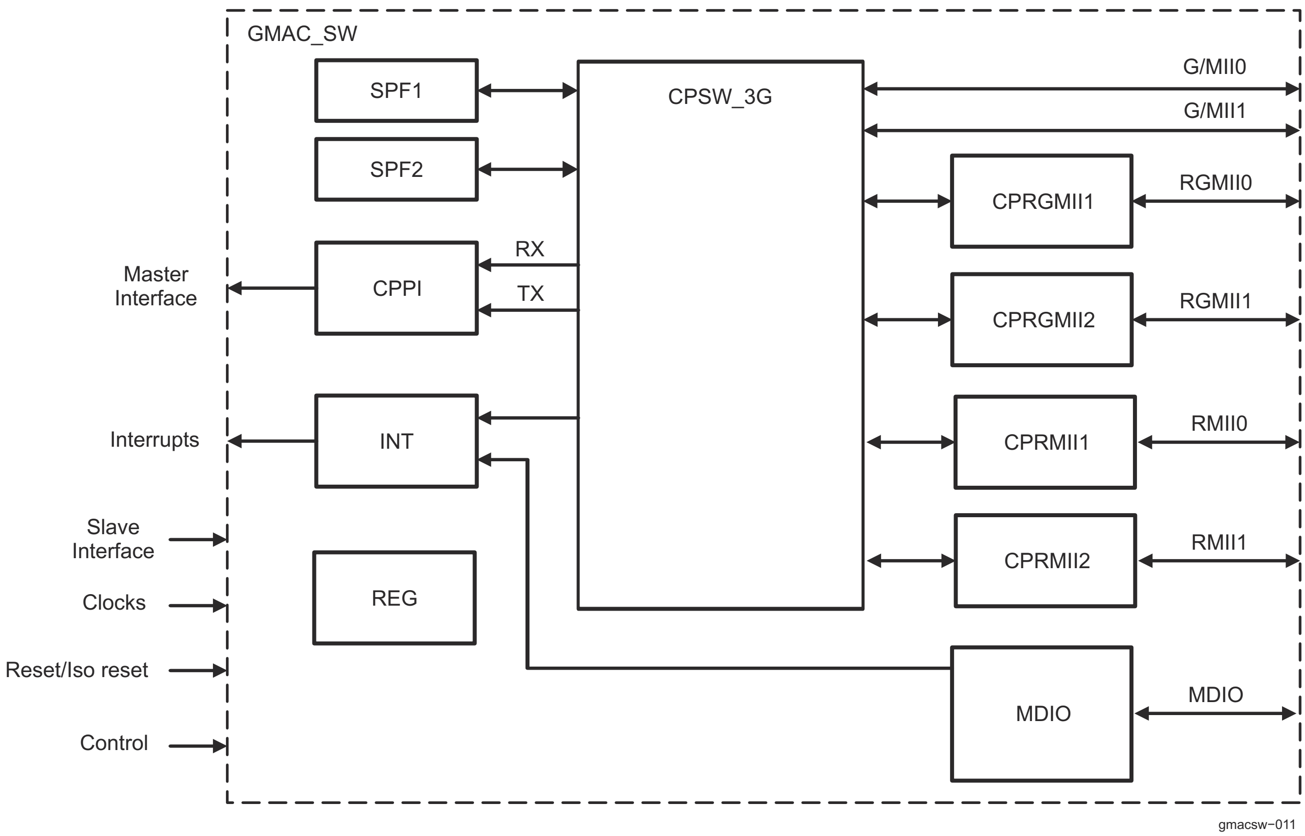


Figure 24-191. GMAC_SW Top Level Block Diagram

24.11.4.2 GMAC_SW Ports

Ethernet Subsystem has three Ports. Port 0 is the Host port (internal to the Subsystem). Ports 1 and 2 are the external ports connected to G/MII, RGMII, or RMII interfaces as per the interface selected.

Naming conventions followed in this chapter:

- Port0 is referred to the Host Port
- Port1 is referred to the interfaces GMII0/RGMII0/RMII0
- Port2 is referred to the interfaces GMII1/RGMII1/RMII1

24.11.4.2.1 Interface Mode Selection

The 3-port switch (GMAC_SW) Ethernet Subsystem has two 10/100/1000 Ethernet ports with selectable MII, RMII, and RGMII interfaces.

The interface mode is selected by configuring the MII mode selection register bitfields (GMII1_SEL and GMII2_SEL) in the control module. See *Control Module Register Manual* for details.

See device data manual for configuring the pin mux mode as per the interface selected.

24.11.4.3 Clocking

24.11.4.3.1 Subsystem Clocking

GMAC_SW clocking summary is shown in Section 24.11.3, *GMAC_SW Integration*.

24.11.4.3.2 Interface Clocking

Data is transmitted and received with respect to the reference clocks of the interface pins.

24.11.4.3.2.1 G/MII Interface Clocking

GMII1_MR_CLK, GMII1_MT_CLK, GMII2_MR_CLK, GMII2_MT_CLK frequencies are fixed by the 802.3 specification.

- 2.5 MHz at 10 Mbps
- 25 MHz at 100 Mbps

24.11.4.3.2.2 RGMII Interface Clocking

RGMII_RXC, RGMII_TXC frequencies are:

- 2.5 MHz at 10 Mbps
- 25 MHz at 100 Mbps
- 125 MHz at 1000 Mbps

24.11.4.3.2.3 RMII Interface Clocking

RMII interface clock RMII_50MHZ_CLK frequency is:

- 50 MHz at 10 Mbps
- 50 MHz at 100 Mbps

See the RMII_CLK_SETTING bit in the device Control Module and CLKSEL_REF in PRCM to configure the clocking option of RMII (external pin or PRCM).

24.11.4.3.2.4 MDIO Clocking

The MDIO clock is based on a divide-down of the interface (MAIN_CLK) clock, running at 125 MHz.The application software or driver must control the divide-down value.

See the MDIO_CONTROL register for configuring the Clock Divider (CLKDIV) value.

24.11.4.4 Software IDLE

The submodule software idle register bits enable CPSW_3G operation to be completely or partially suspended by software control. There are three CPSW_3G submodules that contain software idle register bits (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to enter the idle state. The idle state is entered at packet boundaries, and no further packet operations will occur on an idled submodule until the idle command is removed. The CPSW_3G module enters the idle state when all three submodules are commanded to enter and have entered the idle state. Idle status is determined by reading or polling the three submodule idle bits. The CPSW_3G is in the idle state when all three submodules are in the idle state. The CPSW_SOFT_IDLE[0] SOFT_IDLE bit may be set if desired after the submodules are in the idle state. The CPSW SOFT_IDLE bit causes packets to not be transferred from one FIFO to another FIFO internal to the switch.

24.11.4.5 Interrupt Functionality

GMAC_SW Ethernet Subsystem has four Interrupt outputs:

- RX_PULSE - Receive Interrupt
- TX_PULSE - Transmit Interrupt
- RX_THRESH_PULSE - Receive Threshold Interrupt
- MISC_PULSE - Miscellaneous Interrupt.

24.11.4.5.1 Receive Packet Completion Pulse Interrupt (RX_PULSE)

The RX_PULSE interrupt is a pulse interrupt selected from the GMAC_SW RX_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding (RX_PEND[7:0]).

The following steps will enable the receive packet completion interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the CPDMA_RX_INTMASK_SET register.
2. The receive completion interrupt(s) to be routed to RX_PULSE is selected by setting one or more bits in the receive interrupt enable register (WR_C0_RX_EN) . The masked interrupt status can be read in the address location of RX_STAT bit in the WR_C0_RX_STAT register.

When the GMAC_SW completes a packet reception, the subsystem issues an interrupt to the host processor by writing the packet's last buffer descriptor address to the appropriate channel queue's receive completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon interrupt reception, the software processes one or more packets from the buffer chain and then acknowledges one or more interrupt(s) by writing the address of the last buffer descriptor processed to the queue's associated receive completion pointer (RX *n*_CP) in the receive DMA state RAM.

Upon reception of an interrupt, software should perform the following:

1. Read the WR_C0_RX_STAT[7:0] RX_STAT bit address location to determine which channel(s) caused the interrupt.
2. Process received packets for the interrupting channel(s).
3. Write the GMAC_SW completion pointer(s) (RX *n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the subsystem (address of last buffer descriptor used by the subsystem). If the two values are not equal (which means that the GMAC_SW has received more packets than the software has processed), the receive packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the system has received), the pending interrupt is de-asserted. The value that the GMAC_SW is expecting is found by reading the receive channel *n* completion pointer register (RX *n*_CP).
4. Write the value 1h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

1. The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the CPDMA_RX_INTMASK_CLEAR register.
2. The receive completion pulse interrupt could be disabled by clearing to 0 all the bits in the WR_C0_RX_EN register.

The software could still poll for the CPDMA_RX_INTSTAT_RAW and CPDMA_RX_INTSTAT_MASKED registers if the corresponding interrupts are enabled.

24.11.4.5.2 Transmit Packet Completion Pulse Interrupt (TX_PULSE)

The TX_PULSE interrupt is a pulse interrupt selected from the GMAC_SW TX_PEND[7:0] interrupts. The transmit DMA controller has eight channels with each channel having a corresponding (TX_PEND[7:0]).

To enable the transmit packet completion interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the CPDMA_TX_INTMASK_SET register.
2. The transmit completion interrupt(s) to be routed to TX_PULSE is selected by setting one or more bits in the transmit interrupt enable register WR_C0_TX_EN .The masked interrupt status can be read in the address location of TX_STAT bit in the WR_C0_TX_STAT register.

When the GMAC_SW completes the transmission of a packet, the GMAC_SW subsystem issues an interrupt to the host processor by writing the packet's last buffer descriptor address to the appropriate channel queue's transmit completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon reception of an interrupt, software should perform the following:

1. Read the TX_STAT bit address location to determine which channel(s) caused the interrupt
2. Process received packets for the interrupting channel(s).
3. Write the GMAC_SW completion pointer(s) (TX *n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the GMAC_SW (address of last buffer descriptor used by the GMAC_SW). If the two values are not equal (which means that the GMAC_SW has transmitted more packets than the software has processed), the transmit packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the subsystem has transferred), the pending interrupt is cleared. The value that the GMAC_SW is expecting is found by reading the transmit channel *n*completion pointer register (TX *n*_CP).
4. Write the 2h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

1. The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the CPDMA_TX_INTMASK_CLEAR register.
2. The receive completion pulse interrupt could be disabled by clearing to 0 all the bits in the WR_C0_TX_EN register.

The software could still poll for the CPDMA_TX_INTSTAT_RAW and CPDMA_TX_INTSTAT_MASKED registers, if the corresponding interrupts are enabled.

24.11.4.5.3 Receive Threshold Pulse Interrupt (RX_THRESH_PULSE)

The RX_THRESH_PULSE interrupt is an immediate (non-paced) pulse interrupt selected from the CPSW_3G RX_THRESH_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding threshold pulse interrupt (RX_THRESH_PEND[7:0]).

To enable the receive threshold pulse Interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the CPDMA_RX_INTMASK_SET register.
2. The receive threshold interrupt(s) to be routed to RX_THRESH_PULSE is selected by setting one or more bits in the receive threshold interrupt enable register WR_C0_RX_THRESH_EN. The masked interrupt status can be read in the address location of RX_THRESH_STAT bit in the WR_C0_RX_THRESH_STAT register.

The RX_THRESH_PULSE is asserted when enabled when the channel's associated free buffer count RX *n*_FREEBUFFER is less than or equal to the corresponding RX *n*_PENDTHRESH register.

Upon reception of an interrupt, software should perform the following:

1. Read the RX_THRESH_STAT bit address location to determine which channel(s) caused the interrupt.
2. Process the received packets in order to add more buffers to any channel that is below the threshold value.
3. Write the CPSW_3G completion pointer(s).
4. Write the value 0h to the CPDMA_EOI_VECTOR register.

The threshold pulse interrupt is an immediate interrupt intended to indicate that software should immediately process packets to preclude an overrun condition from occurring for the particular channels.

To disable the interrupt:

1. The eight channel receive threshold interrupts may be individually disabled by writing to 1 the appropriate bit in the CPDMA_RX_INTMASK_CLEAR register.
2. The receive threshold pulse interrupt could be disabled by clearing to 0 the corresponding bits in the WR_C0_RX_THRESH_EN register.

The software could still poll for the CPDMA_RX_INTSTAT_RAW and CPDMA_RX_INTSTAT_MASKED registers, if the corresponding interrupts are enabled.

24.11.4.5.4 Miscellaneous Pulse Interrupt (MISC_PULSE)

The MISC_PULSE interrupt is an immediate pulse interrupt selected from the miscellaneous interrupts (SPF2_PEND, SPF1_PEND, EVNT_PEND,STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT).

The miscellaneous interrupt(s) is selected by setting one or more bits in the miscellaneous interrupt enable register (WR_C0_MISC_EN).

Upon reception of an interrupt, software should perform the following:

- Read the MISC_STAT bit address location to determine the source of the interrupt.
- Process the interrupt.
- Write the value 3h to the CPDMA_EOI_VECTOR register.

Note

The WR_C0_MISC_STAT register's MDIO_LINKINT and MDIO_USERINT bitfields represent the Port0/Phy0 status. MDIO_LINKINT[1] and MDIO_USERINT[1] are not provided in WR_C0_MISC_STAT register. As such, MDIO Link and User interrupts can only be generated for Port0. For Port1, software must poll the status, visible in the MDIO_LINKINTMASKED or MDIO_USERINTMASKED registers.

24.11.4.5.4.1 EVNT_PEND(CPTS_PEND) Interrupt

See Section 24.11.4.10, *Common Platform Time Sync (CPTS)* for more details on this interrupt.

24.11.4.5.4.2 Statistics Interrupt

The statistics level interrupt (STAT_PEND) will be asserted, if enabled when any statistics value is greater than or equal to 8000 0000h. The statistics interrupt is cleared by writing to decrement all statistics values greater than 8000 0000h (such that their new values are less than 8000 0000h). The raw and masked statistics interrupt status may be read by reading the CPDMA_TX_INTSTAT_RAW and CPDMA_TX_INTSTAT_MASKED registers, respectively.

The Statistics interrupt is enabled by setting to 1 the STAT_INT_MASK bit in the CPDMA_DMA_INTMASK_SET register

24.11.4.5.4.3 Host Error interrupt

The host error interrupt (HOST_PEND) will be asserted, if enabled when a host error is detected during transmit or receive CPDMA transactions. The host error interrupt is intended for software debug, and is cleared by a warm reset or a system reset. The raw and masked host interrupt status can be read by reading the CPDMA_DMA_INTSTAT_RAW and CPDMA_DMA_INTSTAT_MASKED registers, respectively.

The transmit host error conditions are:

- SOP error
- OWNERSHIP bit not set in SOP buffer
- next buffer descriptor pointer without EOP cleared to 0
- buffer pointer cleared to 0

- buffer length cleared to 0
- packet length error

The receive host error conditions are:

- OWNERSHIP bit not set in input buffer
- Zero buffer pointer
- Zero buffer Length on non-SOP descriptor
- SOP buffer length not greater than offset

The HOST_PEND is enabled by setting to 1 the HOST_ERR_INTMASK in the CPDMA_DMA_INTMASK_SET register. The host error interrupt is disabled by setting to 1 the appropriate bit in the CPDMA_DMA_INTMASK_CLEAR register.

24.11.4.5.4.4 MDIO Interrupts

MDIO_LINKINT is set if there is a change in the link state of the PHY corresponding to the address in the PHYADR_MON field of the MDIO_USERPHYSEL*n* register and the corresponding LINKINT_ENABLE bit is set. The MDIO_LINKINT event is also captured in the MDIO_LINKINTMASKED register. When the GO bit in the MDIO_USERACCESS*n* register transitions from 1 to 0, indicating the completion of a user access, and the corresponding USERINTMASKSET bit in the MDIO_USERINTMASKSET register is set, the MDIO_USERINT signal is asserted. The MDIO_USERINT event is also captured in the MDIO_USERINTMASKED register.

24.11.4.5.5 Interrupt Pacing

RX_PULSE and TX_PULSE interrupts can be paced. The RX_THRESH_PULSE and MISC_PULSE interrupts are not paced. The Interrupt pacing feature limits the number of interrupts that occur during a given period of time. For heavily loaded systems in which interrupts can occur at a very high rate (for example, 148,800 packets per second for Ethernet), the performance benefit is significant due to minimizing the overhead associated with servicing each interrupt. Interrupt pacing increases the processor cache hit ratio by minimizing the number of times that large interrupt service routines are moved to and from the processor instruction cache.

Each RX_PULSE and TX_PULSE interrupt contains an interrupt pacing sub-block (six total). Each sub-block is disabled by default allowing the selected interrupt inputs to pass through unaffected. The interrupt pacing module counts the number of interrupts that occur over a 1 ms interval of time. At the end of each 1 ms interval, the current number of interrupts is compared with a target number of interrupts (specified by the associated maximum number of interrupts register). Based on the results of the comparison, the length of time during which interrupts are blocked is dynamically adjusted. The 1 ms interval is derived from a 4 µs pulse that is created from a prescale counter whose value is set in the INT_PRESCALE field in the WR_INT_CONTROL register. The INT_PRESCALE value should be written with the number of ICLK periods in 4 µs. The pacing timer determines the interval during which interrupts are blocked and decrements every 4 µs. It is reloaded each time a zero count is reached. The value loaded into the pacing timer is calculated by hardware every 1 ms according to the following algorithm:

```
if (intr_count > 2*intr_max)
    pace_timer = 255;
  else if (intr_count > 1.5*intr_max)
    pace_timer = last_pace_timer*2 + 1;
  else if (intr_count > 1.0*intr_max)
    pace_timer = last_pace_timer + 1;
  else if (intr_count > 0.5*intr_max)
    pace_timer = last_pace_timer - 1;
  else if (intr_count != 0)
    pace_timer = last_pace_timer/2;
else
pace_timer = 0;
```

If the rate of interrupt inputs is much less than the target interrupt rate specified in the associated maximum interrupts register, then the interrupt is not blocked. If the interrupt rate is greater than the target rate, the interrupt will be "paced" at the rate specified in the interrupt maximum register. The WR_C0_RX_IMAX/ WR_C0_TX_IMAX register should be written with a value between 2 and 63 inclusive, indicating the target number of interrupts per millisecond.

24.11.4.6 Reset Isolation

Reset isolation for the GMAC_SW allows the switch function to remain active in during all device resets except for POR pin reset and ICEPICK COLD reset. Packet traffic to/from the GMAC_SW host will be flushed/dropped, but the ethernet switch will remain operational for all traffic between external devices on the switch even though the device is undergoing a device reset. Pin mux configuration for ethernet related I/O and reference clocks needed by the GMAC_SW to be active is controlled by a protected control module bit.

24.11.4.6.1 Reset Isolation Functional Description

The device has two modes of operation concerning the reset of the GMAC_SW Ethernet switch. The mode is controlled by the GMAC_RESET_ISOLATION_ENABLE bit in the Control Module. This bit defaults to 0. Any modification of this bit first requires writing an unlock pattern to lock register in device control module. After modification, the bit should again be locked by writing appropriate value to the lock register. Writes to the GMAC_RESET_ISOLATION_ENABLE bit and to the lock register must all be supervisor mode writes.

GMAC_RESET_ISOLATION_ENABLE = 0 (disabled)

1. This is the default state of the bit after control module reset.
2. Upon any device level resets, the entire GMAC_SW, DPLL_GMAC, L3/L4 interconnect, control module (including all pin mux control and the GMAC_RESET_ISOLATION_ENABLE bit itself) are immediately reset.

GMAC_RESET_ISOLATION_ENABLE = 1 (enabled)

1. This mode is selected when the GMAC_RESET_ISOLATION_ENABLE bit is set to 1 by software.
2. Upon any device reset source other than porz pin or ICEPICK cold (that is, this includes software global cold, any watchdog reset, warm resetn pin, ICEPICK warm, software global warm or security violation), the following is true:
   a. The CPSW_3GSS_R is put into "isolate" mode and non-switch related portions of the subsystem are reset.
   b. The 50 MHz and 125 MHz reference clocks to the GMAC_SW Ethernet Subsystem remain active throughout the entire reset condition.
   c. The control for pin multiplexing for all of the signals maintain their current configuration throughout the entire reset condition.
   d. The reset-isolated logic inside GMAC_SW Ethernet Subsystem maintains the switch functionality
3. Upon any cold reset sources, the entire GMAC_SW Ethernet Subsystem, DPLL_GMAC, control module (including all pin mux control and the GMAC_RESET_ISOLATION_ENABLE bit itself) are reset.

For more details on the register configuration, see the Control Module CTRL_CORE_CONTROL_IO_2 register in *Control Module Register Manual*.

24.11.4.7 Software Reset

The CPSW_3G software reset register (CPSW_SOFT_RESET), CPSW_3GSS software reset register (WR_SOFT_RESET) and the three submodule software reset registers enable the CPSW_3GSS to be reset by software.

There are three CPSW_3G submodules that contain software reset registers (CPGMAC_SL1, CPGMAC_SL2 (SL_SOFT_RESET), and CPDMA (CPDMA_SOFT_RESET)). Each of the three submodules may be individually commanded to be reset by software.

For the CPDMA, the reset state is entered at packet boundaries, at which time the CPDMA reset occurs. The CPGMAC_SL soft reset is immediate. Submodule reset status is determined by reading or polling the submodule reset bit. If the submodule reset bit is read as a one, then the reset process has not yet completed. The submodule soft reset process could take up to 2ms each. The reset has completed if the submodule reset bit is read as a zero.

After all three submodules (in any order) have been reset and a read of each submodule reset bit indicates that the reset process is complete, the CPSW_3G software reset register bit may be written to complete the CPSW_3G module software reset operation. The CPSW_3G software reset bit controls the reset of the FIFO's, the statistics submodule, and the address lookup engine (ALE). The CPSW_3G software reset is immediate and will be indicated by reading a zero from the soft reset bit.

The CPSW_3GSS software reset bit controls the reset of the INT, REGS, and CPPI. The CPSW_3GSS software reset is immediate and will be indicated by reading a zero from the soft reset bit.

24.11.4.8 CPSW_3G

The CPSW_3G G/MII interfaces are compliant to the IEEE Std 802.3 Specification.

The CPSW_3G contains two CPGMAC_SL interfaces (ports 1 and 2), one CPPI interface Host Port (port 0), Common Platform Time Sync (CPTS), ALE Engine and CPDMA. A top-level block diagram of the CPSW_3G is shown in Figure 24-192.

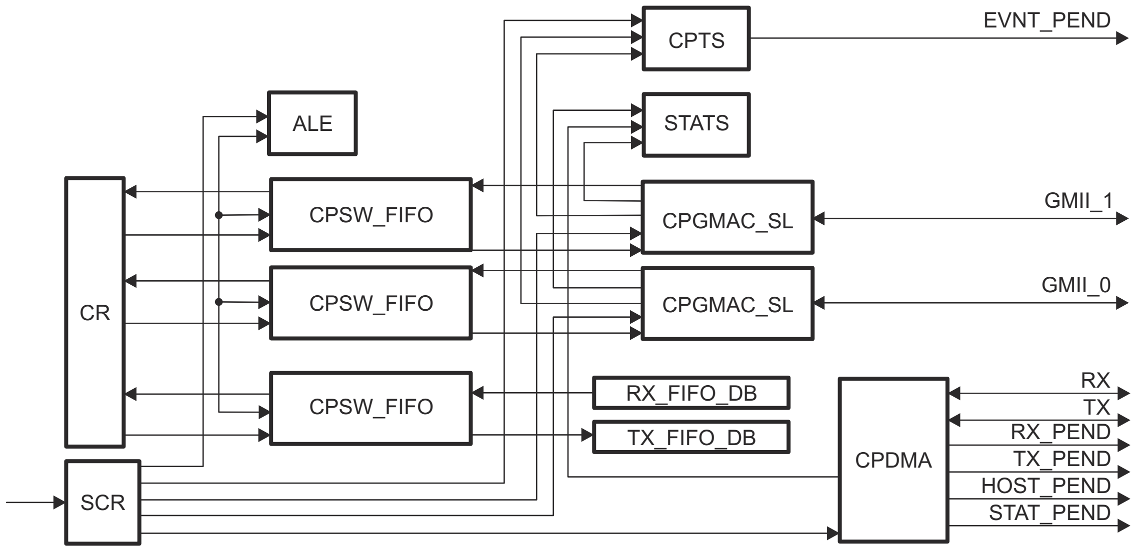


Figure 24-192. CPSW_3G Block Diagram

24.11.4.8.1 CPDMA RX and TX Interfaces

The CPDMA submodule is a CPPI compliant packet DMA transfer controller. The CPPI interface is port 0.

24.11.4.8.1.1 Functional Operation

After reset, initialization, and configuration the host may initiate transmit operations. Transmit operations are initiated by host writes to the appropriate transmit channel head descriptor pointer contained in the STATERAM block. The transmit DMA controller then fetches the first packet in the packet chain from memory in accordance with CPPI protocol. The DMA controller writes the packet into the external transmit FIFO in 64-byte bursts (maximum).

Receive operations are initiated by host writes to the appropriate receive channel head descriptor pointer after host initialization and configuration. The receive DMA controller writes the receive packet data to external memory in accordance with CPPI protocol. For a detailed description of buffer descriptors, see Section 24.11.4.11, *CPPI Buffer Descriptors*.

24.11.4.8.1.2 Receive DMA Interface

The receive DMA is an eight channel CPPI compliant interface. Each channel has a single queue for frame reception.

24.11.4.8.1.2.1 Receive DMA Host Configuration

To configure the RX DMA for operation the software must perform the following:

1. Initialize the receive addresses.
2. Initialize the RX_HDP registers to 0.

3. Enable the desired receive interrupts in the CPDMA_RX_INTMASK_SET register.
4. Write the CPDMA_RX_BUFFER_OFFSET register value.
5. Setup the receive channel(s) buffer descriptors in host memory as required by CPPI
6. Enable the RX DMA controller by setting the RX_EN bit in the CPDMA_RX_CONTROL register.

24.11.4.8.1.2.2 Receive Channel Teardown

The host commands a receive channel teardown by writing the channel number to the CPDMA_RX_TEARDOWN register. When a teardown command is issued to an enabled receive channel the following will occur:

- Any current frame in reception will complete normally.
- The teardown_complete bit will be set in the next buffer descriptor in the chain (if there is one).
- The channel head descriptor pointer will be cleared to 0.
- A receive interrupt for the channel will be issued to the host.
- The software should acknowledge a teardown interrupt with a FFFF FFFCh Acknowledge value.

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by a set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a FFFF FFFCh acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be FFFF FFFCh if the interrupt was due to a teardown command.

24.11.4.8.1.3 Transmit DMA Interface

The tranmsit DMA is an eight channel CPPI compliant interface. Priority between the eight queues may be either fixed or round robin as selected by the TX_PTYPE bit in the CPDMA_DMACONTROL register. If the priority type is fixed, then channel 7 has the highest priority and channel 0 has the lowest priority. Round robin priority proceeds from channel 0 to channel 7.

24.11.4.8.1.3.1 Transmit DMA Host Configuration

To configure the TX DMA for operation the software must do the following:

1. Initialize the TX_HDP registers to 0.
2. Enable the desired transmit interrupts in the CPDMA_TX_INTMASK_SET register.
3. Setup the transmit channel(s) buffer descriptors in host memory as defined in CPPI.
4. Configure and enable the transmit operation as desired in the CPDMA_TX_CONTROL register.
5. Write the appropriate TX_HDP registers with the appropriate values to start transmit operations.

24.11.4.8.1.3.2 Transmit Channel Teardown

The host commands a transmit channel teardown by writing the channel number to the CPDMA_TX_TEARDOWN register. When a teardown command is issued to an enabled transmit channel the following will occur:

- Any frame currently in transmission will complete normally
- The teardown complete bit will be set in the next sop buffer descriptor (if there is one).
- The channel head descriptor pointer will be cleared to 0.
- An interrupt will be issued to inform the host of the channel teardown.
- The software should acknowledge a teardown interrupt with a FFFF FFFCh acknowledge value

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a FFFF FFFCh acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be FFFF FFFCh if the interrupt was due to a teardown command.

24.11.4.8.1.4 Transmit Rate Limiting

Transmit operations can be configured to rate limit the transmit data for each transmit priority. Rate limiting is enabled for a channel when the TX_RLIM bit associated with that channel is set in the CPDMA_DMACONTROL register. Rate limited channels must be the highest priority channels. For example, if two rate limited channels are required then TX_RLIM should be set to 11000000b with the MSB corresponding to channel 7. When any channels are configured to be rate-limiting, the priority type must be fixed for transmit. Round-robin priority type is not allowed when rate-limiting. Each of the eight transmit priorities has an associated register to control the rate at which the priority is allowed to send data (TX_PRI(0..7)_RATE) when the channel is rate-limiting. Each priority has a send count (PRI(0..7)_SEND_CNT) and an idle count (PRI(0..7)_IDLE_CNT). The transfer rate includes the inter-packet gap (12 bytes) and the preamble (8 bytes). The rate in Mbits/second that each priority is allowed to send is controlled by the equation:

Priority Transfer rate in Mbit/s = ((PRI_IDLE_CNT/(PRI_IDLE_CNT + PRI_SEND_CNT)) × frequency × 32

Where *frequency* is the CPDMA interface clock (MAIN_CLK) frequency.

24.11.4.8.1.5 Command IDLE

The CMD_IDLE bit in the CPDMA_DMACONTROL register allows CPDMA operation to be suspended. When the idle state is commanded, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell FIFO will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Commanded idle is similar in operation to emulation control and clock stop.

24.11.4.8.2 Address Lookup Engine (ALE)

The address lookup engine (ALE) processes all received packets to determine which port(s) if any that the packet should the forwarded to. The ALE uses the incoming packet received port number, destination address, source address, length/type, and VLAN information to determine how the packet should be forwarded. The ALE outputs the port mask to the switch fabric that indicates the port(s) the packet should be forwarded to. The ALE is enabled when the ENABLE_ALE bit in the ALE_CONTROL register is set. All packets are dropped when the ENABLE_ALE bit is cleared to 0.

In normal operation, the CPGMAC_SL modules are configured to issue an abort, instead of an end of packet, at the end of a packet that contains an error (runt, frag, oversize, jabber, crc, alignment, code etc.) or at the end of a mac control packet. However, when the SL_MACCONTROL configuration bit(s) RX_CEF_EN, RX_CSF_EN, or RX_CMF_EN are set, error frames, short frames or mac control frames have a normal end of packet instead of an abort at the end of the packet. When the ALE receives a packet that contains errors (due to a set header error bit), or a mac control frame and does not receive an abort, the packet will be forwarded only to the host port (port 0). No ALE learning occurs on packets with errors or mac control frames. Learning is based on source address and lookup is based on destination address.

The ALE may be configured to operate in bypass mode by setting the ALE_BYPASS bit in the ALE_CONTROL register. When in bypass mode, all CPGMAC_SL received packets are forwarded only to the host port (port 0). Packets from the two ports can be on separate RX DMA channels by configuring the P0_CPDMA_RX_CH_MAP register. In bypass mode, the ALE processes host port transmit packets the same as in normal mode. In general, packets would be directed by the host in bypass mode.

The ALE may be configured to operate in OUI deny mode by setting the ENABLE_OUI_DENY bit in the ALE_CONTROL register. When in OUI deny mode, a packet with a non-matching OUI source address will be dropped unless the destination address matches a multicast table entry with the super bit set. Broadcast packets will be dropped unless the broadcast address is entered into the table with the super bit set. Unicast packets will be dropped unless the unicast address is in the table with block and secure both set (supervisory unicast packet).

Multicast supervisory packets are designated by the super bit in the table entry. Unicast supervisory packets are indicated when block and secure are both set. Supervisory packets are not dropped due to rate limiting, OUI, or VLAN processing.

24.11.4.8.2.1 Address Table Entry

The ALE table contains 1024 entries. Each table entry represents a free entry, an address, a VLAN, an address/VLAN pair, or an OUI address. Software should ensure that there are not double address entries in the table. The double entry used would be indeterminate. Reserved table bits must be written with zeroes.

Source Address learning occurs for packets with a unicast, multicast or broadcast destination address and a unicast or multicast (including broadcast) source address. Multicast source addresses have the group bit (bit 40) cleared before ALE processing begins, changing the multicast source address to a unicast source address. A multicast address of all ones is the broadcast address which may be added to the table. A learned unicast source address is added to the table with the following control bits:

Table 24-873. Learned Address Control Bits

Bit(s)	Value
unicast_type	11
Block	0
Secure	0

If a received packet has a source address that is equal to the destination address then the following occurs:

- The address is learned if the address is not found in the table.
- The address is updated if the address is found.
- The packet is dropped.

Table Entry Type

00 - Free Entry

01 - Address Entry : unicast or multicast determined by destination **address bit 40.**

10 - VLAN entry

11 - VLAN Address Entry : unicast or multicast determined by **address bit 40.**

24.11.4.8.2.1.1 Free Table Entry

Table 24-874. Free (Unused) Address Table Entry Bit Values

71:62	61:60	59:0
Reserved	ENTRY_TYPE(00)	Reserved

24.11.4.8.2.1.2 Multicast Address Table Entry

Table 24-875. Multicast Address Table Entry Bit Values

71:69	68:66	65	64	63:62	61:60	59:48	47:0
Reserved	PORT_MASK	SUPER	Reserved	MCAST_FWD_STATE	ENTRY_TYPE(01)	Reserved	MULTICAST_ADDRESS

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This 3-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT_MASK has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

24.11.4.8.2.1.3 VLAN/Multicast Address Table Entry

Table 24-876. VLAN/Multicast Address Table Entry Bit Values

71:69	68:66	65	64	63:62	61:60	59:48	47:0
Reseved	PORT_MASK	SUPER	Reserved	MCAST_FWD_STATE	ENTRY_TYPE(11)	VLAN_ID	MULTICAST_ADDRESS

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This 3-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT_MASk has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

24.11.4.8.2.1.4 Unicast Address Table Entry

Table 24-877. Unicast Address Table Entry Bit Values

71:68	67:66	65	64	63:62	61:60	59:48	47:0
Reseved	PORT_NUMBER	BLOCK	SECURE	UNICAST_TYPE (00) or (X1)	ENTRY_TYPE(01)	Reserved	UNICAST_ADDRESS

Port Number (PORT_NUMBER)

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).

Block (BLOCK)

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 - Address is not blocked.

1 - Drop a packet with a matching source or destination address (secure

must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/ Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port_number.

0 - Received port number is a don't care.

1 - Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

24.11.4.8.2.1.5 OUI Unicast Address Table Entry

Table 24-878. OUI Unicast Address Table Entry Bit Values

71:64	63:62	61:60	59:48	47:24	23:0
Reserved	UNICAST_TYPE(10)	ENTRY_TYPE(01)	Reserved	UNICAST_OUI	Reserved

Unicast Type (UNICAST_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (UNICAST_OUI)

For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup.

24.11.4.8.2.1.6 VLAN/Unicast Address Table Entry

Table 24-879. Unicast Address Table Entry Bit Values

71:68	67:66	65	64	63:62	61:60	59:48	47:0
Reseved	PORT_NUMBER	BLOCK	SECURE	UNICAST_TYPE (00) or (X1)	ENTRY_TYPE(11)	VLAN_ID	UNICAST_ADDRESS

Port Number (PORT_NUMBER)

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).]

Block (BLOCK)

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 - Address is not blocked.

1 - Drop a packet with a matching source or destination address (secure must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/ Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry PORT_NUMBER.

0 - Received port number is a don't care.

1 - Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

24.11.4.8.2.1.7 VLAN Table Entry

Table 24-880. VLAN Table Entry

71:62	61:60	59:48	47:27	26:24	23:19	18:16	15:11	10:8	7:3	2:0
Reserved	ENTRY_TYPE(10)	VLAN_ID	Reserved	FORCE_UNTAGGED_EGRESS	Reserved	REG_MCAST_FLOOD_MASK	Reserved	UNREG_MCAST_FLOOD_MASK	Reserved	VLAN_MEMBER_LIST

Table Entry Type (ENTRY_TYPE)

10: VLAN entry

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Force Untagged Packet Egress (FORCE_UNTAGGED_EGRESS)

This field causes the packet VLAN tag to be removed on egress (except on port 0).

Registered Multicast Flood Mask (REG_MCAST_FLOOD_MASK)

Mask [port 2-1-0] used for multicast when the multicast address is found.

Unregistered Multicast Flood Mask (UNREG_MCAST_FLOOD_MASK)

Mask [port 2-1-0] used for multicast when the multicast address is not found.

VLAN Member List (VLAN_MEMBER_LIST)

This 3-bit field indicates which port(s) are members of the associated VLAN.

24.11.4.8.2.2 Packet Forwarding Processes

There are four processes that an incoming received packet may go through to determine packet forwarding. The processes are Ingress Filtering, VLAN_Aware Lookup, VLAN_Unaware Lookup, and Egress.

Packet processing begins in the Ingress Filtering process. Each port has an associated packet forwarding state that can be one of four values (Disabled, Blocked, Learning, or Forwarding). The default state for all ports is Disabled. The host sets the packet forwarding state for each port.

In the packet ingress process (receive packet process), there is a forward state test for unicast destination addresses and a forward state test for multicast addresses. The multicast forward state test indicates the port states required for the receiving port in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state for the packet to be forwarded for transmission. The mcast_fwd_state indicates the required port state for the receiving port as indicated in the preceding table. The unicast forward state test indicates the port state required for the receiving port in order to forward the unicast packet. The transmit port must be in the Forwarding state in order to forward the packet. The block and secure bits determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state. The transmit port must be in the Forwarding state regardless. The forward state test used in the ingress process is determined by the destination address packet type (multicast/ unicast).

In general, packets received with errors are dropped by the address lookup engine without learning, updating, or touching the address. The error condition and the abort are indicated by the CPGMAC_SL to the ALE. Packets with errors may be passed to the host (not aborted) by a CPGMAC_SL port, if the port has the RX_CMF_EN, RX_CEF_EN, or RX_CSF_EN bit(s) set. Error packets that are passed to the host by the CPGMAC_SL are considered to be bypass packets by the ALE and are sent only to the host. Error packets do not learn, update, or touch addresses regardless of whether they are aborted or sent to the host. Packets with errors received by the host are forwarded as normal.

The following control bits are in the SL_MACCONTROL register:

- [22] RX_CEF_EN - enables frames that are fragments, long, jabber, CRC, code, and alignment errors to be forwarded
- [23] RX_CSF_EN - enables short frames to be forwarded
- [24] RX_CMF_EN - enables MAC control frames to be forwarded.

24.11.4.8.2.3 Learning Process

The learning process is applied to each receive packet that is not aborted. The learning process is a concurrent process with the packet forwarding process.

24.11.4.8.2.4 VLAN Aware Mode

The CPSW_3G is in VLAN aware mode when the VLAN_AWARE bit is set in the CPSW_CONTROL register. In VLAN aware mode, ports 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the RX_VLAN_ENCAP bit in the CPSW_CONTROL register. Port 0 receive packet data is never modified. VLAN is not removed regardless of the force untagged egress bit for Port 0. VLAN encapsulated receive packets have a 32-bit VLAN header encapsulation word added to the packet data. VLAN encapsulated packets are specified by a set rx_vlan_encap bit in the packet buffer descriptor.

Port 0 transmit packets are never VLAN encapsulated (encapsulation is not allowed).

In VLAN aware mode, transmitted packet data is changed depending on the packet type (pkt_type), packet priority (pkt_pri), and VLAN information.

24.11.4.8.2.5 VLAN Unaware Mode

The CPSW_3G is in VLAN unaware mode when the VLAN_AWARE bit is cleared to 0 in the CPSW_CONTROL register. Port 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the RX_VLAN_ENCAP bit in the CPSW_CONTROL register. Port 0 transmit packets are never VLAN encapsulated.

24.11.4.8.3 Packet Priority Handling

Packets are received on three ports, two are CPGMAC_SL Ethernet ports and the third port is the CPPI host port. Received packets have a received packet priority (0 to 7, with 7 being the highest priority).

The received packet priority is the port priority for untagged packets, and the actual packet priority for priority tagged and VLAN tagged packets. The received packet priority is mapped through the receive ports associated packet priority to header packet priority mapping register to obtain the header packet priority (the CPDMA RX and TX nomenclature is reversed from the CPGMAC_SL nomenclature).

The header packet priority is mapped through the header priority to switch priority mapping register to obtain the hardware switch priority (0 to 3, with 3 being the highest priority). The header packet priority is then used as the actual transmit packet priority if the VLAN information is to be sent on egress.

24.11.4.8.4 FIFO Memory Control

Each of the three CPSW_3G ports has an identical associated FIFO. Each FIFO contains a single logical receive queue and four logical transmit queues (priority 0 through 3). Each FIFO memory contains 20,480 bytes (20k) total organized as 2560 by 64-bit words contained in a single memory instance. The FIFO memory is used for the associated port transmit and receive queues. The TX_MAX_BLKS field in the FIFOs associated Px_MAX_BLKS register determines the maximum number of 1k FIFO memory blocks to be allocated to the four logical transmit queues (transmit total). The RX_MAX_BLKS field in the FIFO's associated Px_MAX_BLKS register determines the maximum number of 1k memory blocks to be allocated to the logical receive queue. The TX_MAX_BLKS value plus the RX_MAX_BLKS value must sum to 20 (the total number of blocks in the FIFO). If the sum were less than 20, then some memory blocks would be unused. The default is 17 (decimal) transmit blocks and three receive blocks. The FIFOs follow the naming convention of the Ethernet ports. Host Port is Port0 and External Ports are Port1 and Port2.

24.11.4.8.5 FIFO Transmit Queue Control

There are four transmit queues in each transmit FIFO. Software has some flexibility in determining how packets are loaded into the queues and on how packet priorities are selected for transmission (how packets are removed and transmitted from queues). All ports on the switch have identical FIFO's. For the purposes of the below the transmit FIFO is switch egress even though the port 0 transmit FIFO is connected to the CPDMA receive (also switch egress). The CPDMA nomenclature is reversed from the CPGMAC_SL nomenclature due to legacy reasons.

24.11.4.8.5.1 Normal Priority Mode

When operating in normal mode, lower priority frames are dropped before higher priority frames. The intention is to give preference to higher priority frames. Priority 3 is the highest priority and is allowed to fill the FIFO. Priority 2 will drop packets if the packet is going to take space in the last 2k available. Priority 1 will drop packets if the packet is going to take space in the last 4k available. Priority 0 will drop packets if the packet is going to take space in the last 6k available. If fewer than 4 priorities are to be implemented then the priorities should be mapped such that the highest priorities are used. For example, if two priorities are going to be used then all packets should be mapped to priorities 3 and 2 and priorities 1 and 0 should be unused. Priority escalation may be used in normal priority mode if desired. Normal priority mode is configured as described below:

- Select normal priority mode by setting TX_IN_SEL = 00 for all ports (default value in P0/1/2_TX_IN_CTL)
- Configure priority mapping to use only the highest priorities if less than 4 priorities are used. Refer to Section 24.11.4.8.3, *Packet Priority Handling*.

24.11.4.8.5.2 Dual MAC Mode

When operating in dual MAC mode the intention is to transfer packets between ports 0 and 1 and ports 0 and 2, but not between ports 1 and 2. Each CPGMAC_SL appears as a single MAC with no bridging between MAC's. Each CPGMAC_SL has at least one unique (not the same) mac address.

Dual MAC mode is configured as described below:

- Set the ALE_VLAN_AWARE bit in the ALE_CONTROL register. This bit configures the ALE to process in VLAN aware mode. The CPSW_3G VLAN aware bit (VLAN_AWARE in CPSW_CONTROL) determines how packets VLAN's are processed on CPGMAC_SL egress and does not affect how the ALE processes packets or the packet destination. The CPSW_3G VLAN aware bit may be set or not as required (must be set, if VLAN's are to exit the switch).
- Configure the Port 1 to Port 0 VLAN
  – Add a VLAN Table Entry with ports 0 and 1 as members (clear the flood masks).
  – Add a VLAN/Unicast Address Table Entry with the Port1/0 VLAN and a port number of 0. Packets received on port 1 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.
- Configure the Port 2 to Port 0 VLAN
  – Add a VLAN Table Entry with ports 0 and 2 as members (clear the flood masks).
  – Add a VLAN/Unicast Address Table Entry with the Port2/0 VLAN and a port number of 0. Packets received on port 2 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.
- Packets from the host (port 0) to ports 1 and 2 should be directed. If directed packets are not desired then VLAN with addresses can be added for both destination ports.
- Select the dual mac mode on the port 0 FIFO by setting TX_IN_SEL = 01 in P0_TX_IN_CTL. The intention of this mode is to allow packets from both ethernet ports to be written into the FIFO without one port starving the other port.
- The priority levels may be configured such that packets received on port 1 egress on one CPDMA RX channel while packets received on port 2 egress on a different CPDMA RX channel.

24.11.4.8.5.3 Rate Limit Mode

Rate-limit mode is intended to allow some CPDMA transmit (switch ingress) channels and some CPGMAC_SL FIFO priorities (switch egress) to be rate-limited. Non rate-limited traffic (bulk traffic) is allowed on non rate-limited channels and FIFO priorities. The bulk traffic does not impact the rate-limited traffic. Rate-limited traffic must be configured to be sent to rate-limited queues (via packet priority handling). The allocated rates for rate-limited traffic must not be oversubscribed. For example, if port 1 is sending 15% rate limited traffic to port 2 priority 3, and port 0 is also sending 10% rate-limited traffic to port 2 priority 3, then the port 2 priority 3 egress rate must be configured to be 25% plus a percent or two for margin. The switch must be configured to allow some percentage of non rate-limited traffic. Non rate-limited traffic must be configured to be sent to non rate-limited queues. No packets from the host should be dropped, but non rate-limited traffic received on an ethernet port can be dropped. Rate-limited mode is configured as shown below:

1. Set TX_IN_SEL = 10 in P1/2_TX_IN_CTL to enable ports 1 and 2 transmit FIFO inputs to be configured for rate-limiting queues. Enabling a queue to be rate-limiting with this field affects only the packet being loaded into the FIFO, it does not configure the transmit for queue shaping.
2. Configure the number of rate-limited queues for port 1 and 2 transmit FIFO's by setting the TX_RATE_EN field in P1/2_TX_IN_CTL. Rate limited queues must be the highest number. For example, if there are two rate limited queues then 1100 would be written to this field for priorities 3 and 2. This field enables the FIFO to allow rate-limited traffic into rate-limited queues while discriminating against non rate-limited queues.
3. Set P1_PRIN_SHAPE_EN and P2_PRIN_SHAPE_EN in the CPSW_PTYPE register. These bits determine which queues actually shape the output data stream. In general, the same priorities that are set in TX_RATE_EN are set in these bits as well, but the FIFO input and output enable bits are separate to allow rate-limiting from the host to non shaped channels if desired. When queue shaping is not enabled for a queue then packets are selected for egress based on priority. When queue shaping is enabled then packets are selected for egress based on queue percentages. If shaping is required on a single queue then it must

be priority 3 (priorities 2, 1 and 0 are strict priority). If shaping is required on two queues then it must be on priorities 2 and 3 (priorities 1 and 0 are strict priority). If shaping is required on three queues then it must be priorities 3, 2, and 1 (priority 0 would then get the leftovers). Priority shaping follows the requirements in the IEEE P802.1Qav/D6.0 specification. Priority shaping is not compatible with priority escalation (escalation must be disabled).

4. P0_TX_IN_CTL[17:16] TX_IN_SEL should be set to 00, so Port 0 egress (CPDMA RX) is not rate-limited.
5. The CPDMA is configured for rate-limited transmit (switch ingress) channels by setting the highest bits of the TX_RLIM field in the CPDMA_DMACONTROL register. If there are two rate-limited channels, then TX_RLIM = 11000000 (the rate limited channels must be the highest priorities). Also, the TX_PTYPE bit in the CPDMA_DMACONTROL register must be set (fixed priority mode). Rate-limited channels must go to rate-limited FIFO queues, and the FIFO queue rate must not be oversubscribed.

24.11.4.8.6 Audio Video Bridging

Audio Video Bridging is an ongoing project of IEEE 802.1 concerned with enabling low-latency streaming of time-sensitive audiovisual data over networks. Devices are designated as talkers (transmitters), bridges, or listeners (receivers). It is suggested that the maximum latency could be 2 ms over 7 hops for Class A devices and 20 ms over 7 hops for Class B devices. A hop is essentially a single local area network stage in the journey of a packet. Every time a bridge is encountered between one network section and another a hop is involved. One of the performance goals is that AVB streams will not use more than 75 percent of a link's bandwidth, leaving the remaining capacity for non-AVB streams.

The goal of developing AVB is simply--extend Ethernet's data-networking capabilities to the realm of reliable real-time audio/video networking.

An "Audio Video Bridging" network is one that implements a set of protocols being developed by the IEEE 802.1 Audio/Video Bridging Task Group. There are four primary differences between the proposed Audio Video Bridging architecture and existing 802 architectures (from now on the term "AVB" will be used instead of "Audio Video Bridging"):

1. Precise synchronization - IEEE 802.1AS: "*Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.* "a.k.a Precision Time Protocol (PTP).
2. Traffic shaping for media streams - IEEE 802.1Qav: "*Virtual Bridged Local Area Networks:* Forwarding and Queuing for Time-Sensitive Streams."
3. Admission controls - IEEE 802.1Qat: "*Virtual Bridged Local Area Networks - Amendment 9:* Stream Reservation Protocol (SRP)."
4. Identification of non-participating devices - IEEE 802.1BA: "Audio/Video Bridging (AVB) Systems"

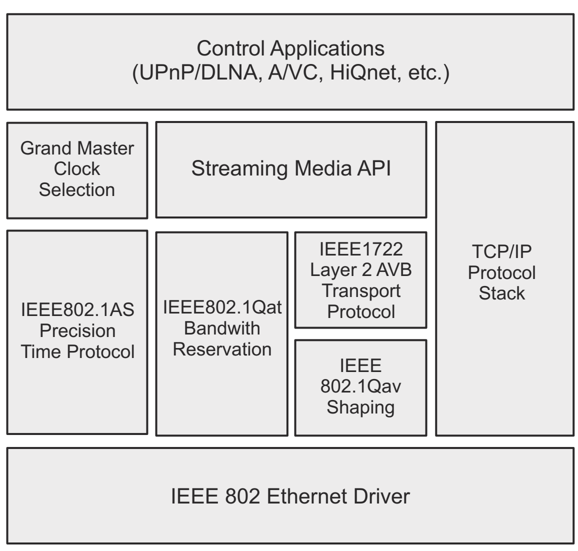


Figure 24-193. The Network Static with AVB

The following sections describe the media transport protocols that work within the AVB framework.

24.11.4.8.6.1 IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks (Precision Time Protocol (PTP))

The protocol defined by 802.1AS automatically selects a device to be the master clock, and then distributes this clock throughout the bridged LAN / IP subnet to all other network devices using link-specific transmit/receive time-stamping. However, we only use a two-step solution only on transmit. That is, we do not modify a packet with the timestamp on the way out. The timestamp packet is sent out and then a separate message with the timestamp is sent by the host afterward. Receive can be one or two step.

Note

The 802.1AS-distributed clock is not used as a media clock. Rather, the shared 802.1AS clock reference is used to regenerate the media clock at the listener/renderer. Such a reference removes the need to force the latency of the network to be constant, or compute long running averages in order to estimate the actual media rate of the transmitter in the presence of substantial network jitter. IEEE 802.1AS is based on the ratified IEEE 1588 standard.

Based on IEEE 1588:2002,Â PTP devices exchange standard Ethernet messages that synchronize network nodes to a common time reference by defining clock master selection and negotiation algorithms, link delay measurement and compensation, and clock rate matching and adjustment mechanisms.

Designed as a simplified profile of IEEE 1588, a primary difference between 1588 and IEEE 802.1AS is that PTP is a layer 2--in other words, a non-IP routable protocol. Like IEEE 1588, PTP defines an automatic method for negotiating the network clock master, the Best Master Clock Algorithm (BMCA). PTP nodes can be assigned one of eight priority levels, presumably based on clock quality. BMCA defines the underlying negotiation and signaling mechanism whose purpose is to identify the AVB LAN Grandmaster. Once a Grandmaster has been selected, synchronization automatically begins.

At the core of 802.1AS synchronization is time-stamping. In short, during PTP message ingress/egress from the 802.1AS-capable MAC, the PTP Ether type triggers the sampling of the value of a local real-time counter (RTC). Slave nodes compare the value of their RTC against the PTP Grandmaster and, by use of link delay measurement and compensation techniques, match their RTC value to the time of the AVB LAN PTP domain. After network time throughout the AVB LAN has converged, periodic SYNC and FOLLOW_UP messages provide the information that enables the PTP rate matching adjustment algorithms. The result is all PTP nodes are then synchronized to the same "Wall Clock" time. PTP assures 1-µs accuracy over seven network hops.

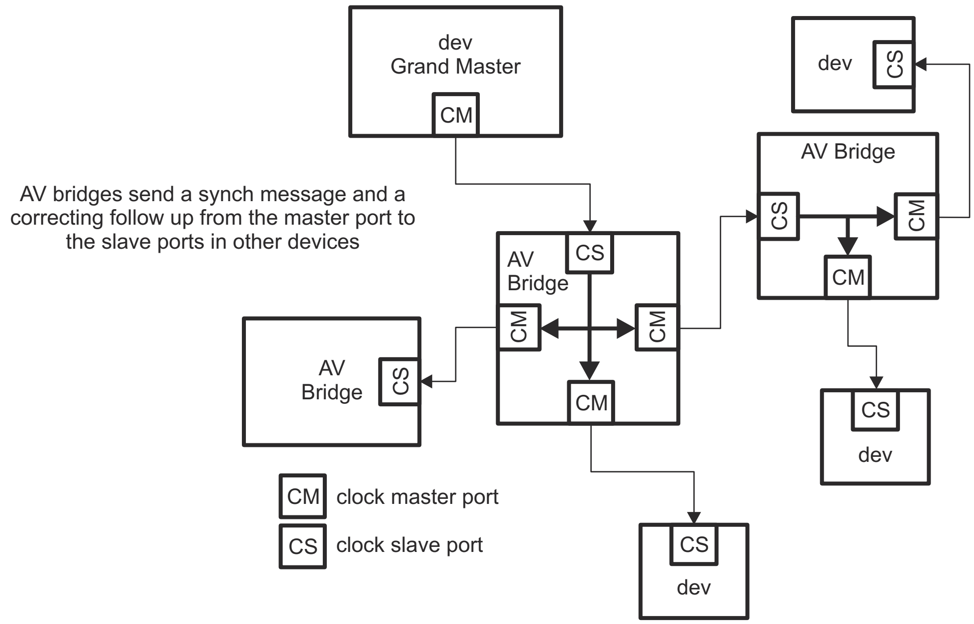


Figure 24-194. AVB Network & PTP Clock Entities

The media transport protocols that work within the AVB framework are:

24.11.4.8.6.1.1 IEEE 1722: "Layer 2 Transport Protocol for Time-Sensitive Streams"

AVBTP or 1722 sits above the IEEE 802.1 AVB plumbing and below the application layer. It acts as the conduit between an Ethernet MAC and a streaming application. AVBTP abstracts the underlying network transmission channel to enable the virtual connection of distributed audio and video CODECs over reliable Ethernet networks. A complete AVBTP Ethernet packet is shown in Figure 24-195 and illustrates how IEC 61883-6 AM824 uncompressed audio samples are encapsulated in an Ethernet frame.

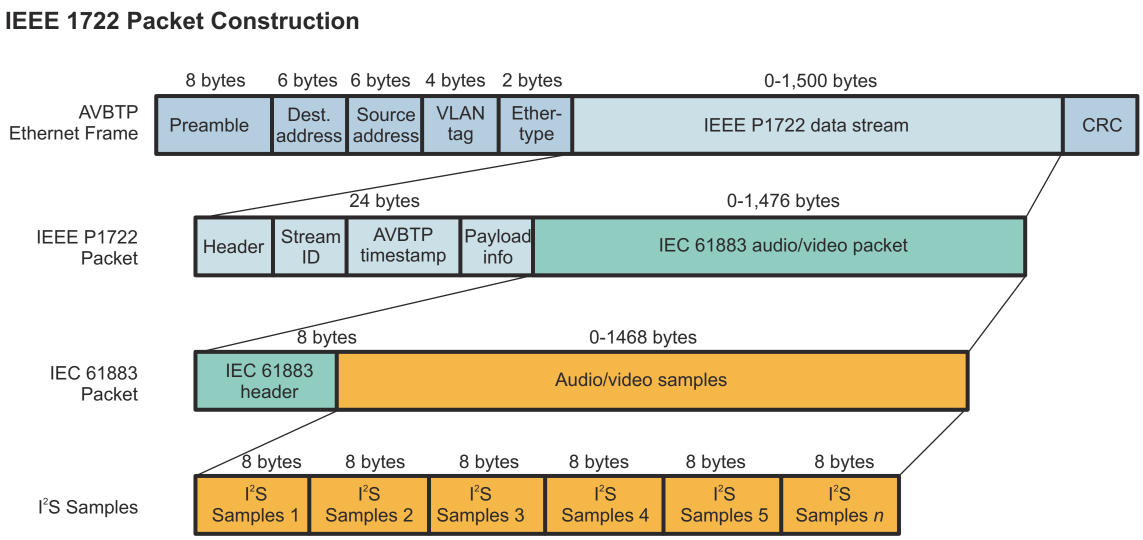


Figure 24-195. IEEE 1722 Packets

1722 or AVBTP Presentation Time and Synchronization:

Synchronization in an AVB network starts with the Precision Time Protocol but ends with synchronized media clocks. PTP is responsible for synchronizing all nodes in an AVB network to identical wall clock time; not for synchronizing media clocks. In other words, PTP does not actually transport synchronized media clocks but instead provides a low-level building block crucial for managing a distributed media synchronization system.

A crucial benefit of this approach is coexistence of multiple, independent media clock domains on an AVB network. Unrelated audio and video streams can simultaneously exist in the same LAN.

AVBTP assumes that AVB node media clocks are clocked by free-running oscillators. It is also assumed that the node's internal concept of wall clock time has been synchronized to the PTP Grandmaster. AVBTP media clock sources embed "AVBTP Presentation Timestamps" in AVBTP streaming packets. Figure 24-196 illustrates the relationship between PTP network time and AVBTP Presentation Timestamps.

24.11.4.8.6.1.2 IEEE 1733: Extends RTCP for RTP Streaming over AVB-supported Networks

This standard specifies the protocol, data encapsulations, connection management and presentation time procedures used to ensure interoperability between audio and video based end stations that use standard networking services provided by all IEEE 802 networks meeting QoS requirements for time-sensitive applications by leveraging the Real-time Transport Protocol (RTP) family of protocols and IEEE 802.1 Audio/ Video Bridging (AVB) protocols.

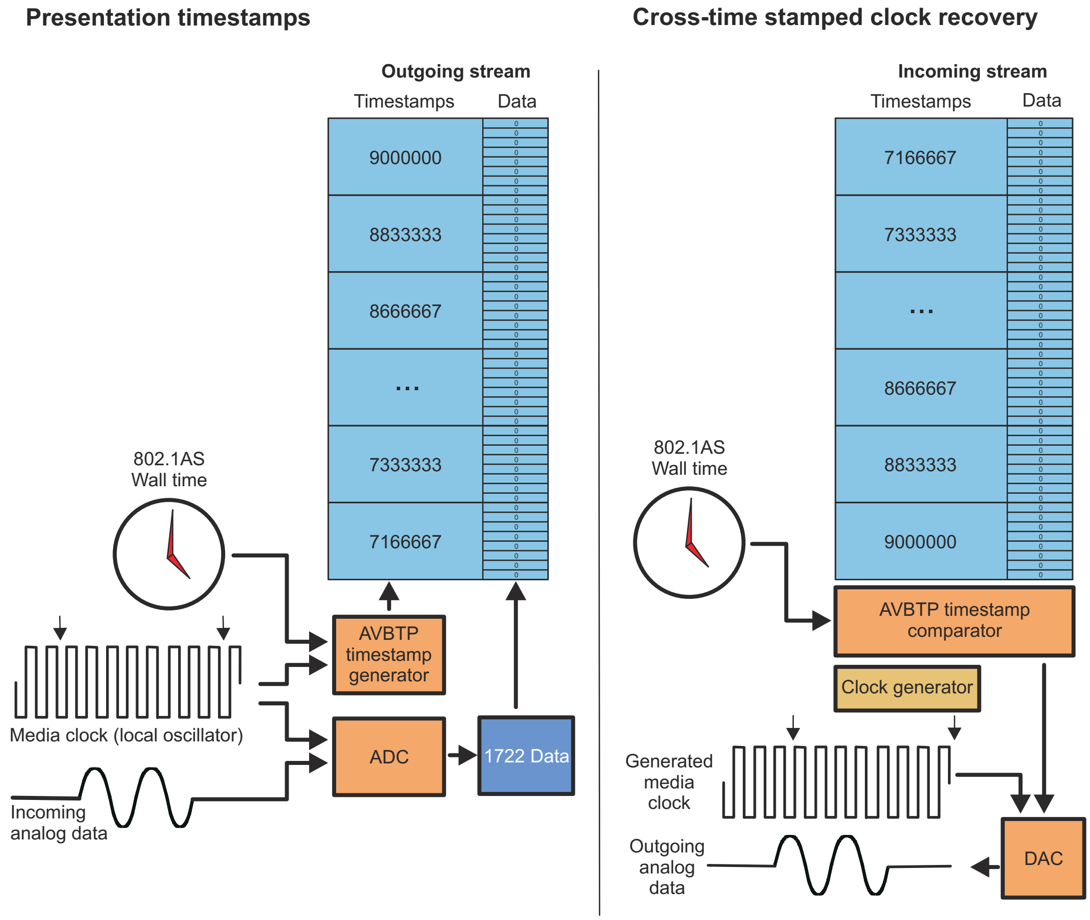


Figure 24-196. Cross Time Stamping and Presentation Timestamps

24.11.4.8.6.2 IEEE 802.1Qav: "Virtual Bridged Local Area Networks: Forwarding and Queuing for Time-Sensitive Streams"

This standard allows bridges to provide guarantees for time-sensitive (that is, bounded latency and delivery variation), loss-sensitive real-time audio video (AV) data transmission (AV traffic). It specifies per priority ingress metering, priority regeneration, and timing-aware queue draining algorithms. This standard uses the timing derived from IEEE 802.1AS. Virtual Local Area Network (VLAN) tag encoded priority values are allocated, in aggregate, to segregate frames among controlled and non-controlled queues, allowing simultaneous support of both AV traffic and other bridged traffic over and between wired and wireless Local Area Networks (LANs).

Such a guarantee in bandwidth is provided by two functional entities:

- A registration protocol, which registers the service and its maximum network utilization with a device or switch (IEEE 802.1Qat: "Virtual Bridged Local Area Networks - Amendment 9: Stream Reservation Protocol (SRP)")
- A hardware bandwidth management service.
  - Receive policing and
  - Transmit rate control.

End Station Behavior

In order for an end station to successfully participate in the transmission and reception of time-sensitive streams, it is necessary for their behavior to be compatible with the operation of the forwarding and queuing mechanisms employed in bridges.

The requirements for end stations that participate as "talkers" i.e., sources of time-sensitive streams are different from the requirements that apply to "listeners", the destination station(s) for the streams.

Talker Behavior

In order for Talker-originated data streams to make use of the credit-based shaper behavior in Bridges, it is a requirement for a Talker to use the priorities that the Bridges in the network recognize as being associated with SR classes exclusively for transmitting stream data.

It is also necessary for the Talker and the Bridges in the path to the Listener(s), to have a common view of the bandwidth required in order to transmit the Talker's streams, and for that bandwidth to be reserved along the path to the Listener(s). This latter requirement can be met by means of stream reservation mechanisms, such as defined in SRP, or by other management means.

End stations that are Talkers shall exhibit transmission behavior for frames that are part of "time-sensitive streams" that is consistent with the operation of the credit-based shaper algorithm, both in terms of the way they transmit frames that are part of an individual data stream, and in terms of the way they transmit stream data frames from a Port.

In effect, the queuing model for a Talker Port (and a Listener port), and for given priorities, can be considered to look like Figure 24-197.

Listener Behavior

The primary requirement for a listener station is that it is capable of buffering the amount of data that could be transmitted for a stream during a time period equivalent to the accumulated maximum jitter that could be experienced by stream data frames in transmission between Talker and Listener.

From the point of view of the specification of the forwarding and queuing requirements for time-sensitive streams, it is assumed that the listener will assess the buffering required for a stream as part of the stream bandwidth reservation mechanisms employed by the implementation.

The credit-based shaper's operation details are beyond the scope of this document.

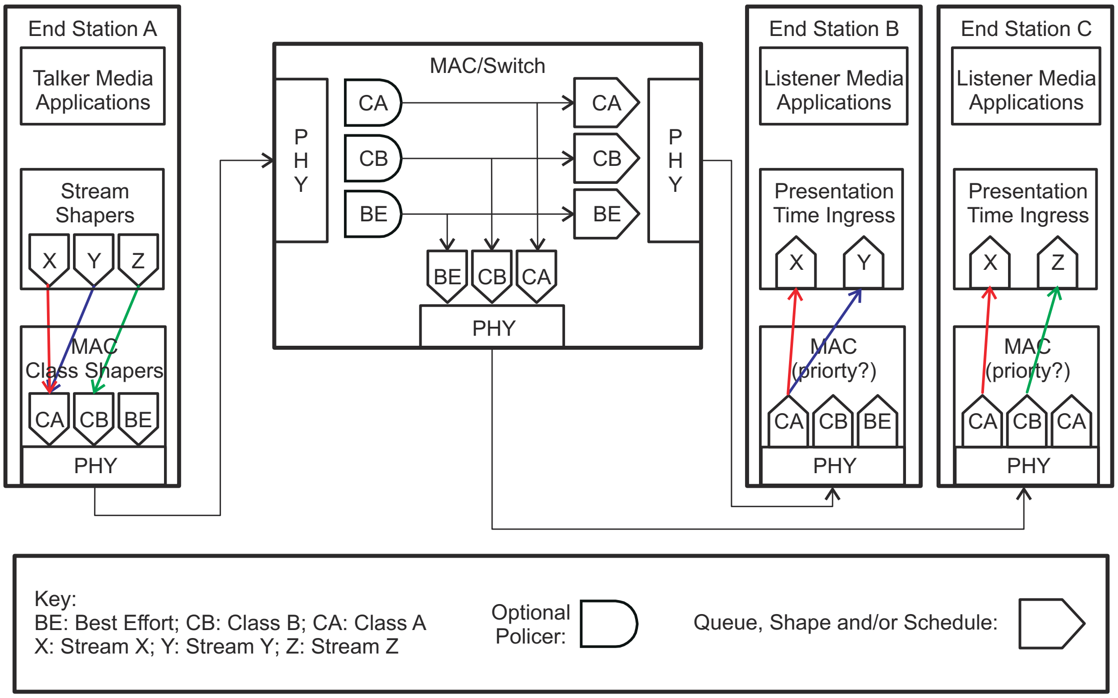


Figure 24-197. AV Stream Queuing/Policing

24.11.4.8.6.2.1 Configuring the Device for 802.1Qav Operation:

There is no dedicated register-set to be configured for the time-sensitive stream handling. The list of functional features of CPSW_3G that will have to be configured are:

- DESCRIPTORS and CHANNEL CONFIGURATIONS:
  – CPPI TX and RX descriptors
  – VLAN and Priority tags

Table 24-881. Example of TX Configuration

TX DMA CHANNEL	Packet Priority	Switch Queue Priority
7	7	3
6	5	2
5	3	1
4	1	0

Table 24-882. Example of RX Configuration

RX DMA CHANNEL	Packet Priority	Switch Queue Priority
0	7	0
0	5	0
0	3	0
0	1	0

- ALE Configuration:
  – ALE in VLAN-ware mode, Non-ALE in bypass mode.

Table 24-883. Example of Rate-limit Configurations

Register	Value	Description
CPSW_PTYPE	0x0006 0000	For Port1 -- 2 highest priority channels.

Table 24-883. Example of Rate-limit Configurations (continued)

Register	Value	Description
P1_TX_IN_CTL[23:20] TX_RATE_EN	0b1100	2 highest priority channels are rate limited
P1_TX_IN_CTL[17:16] TX_IN_SEL	0b10	Rate limit mode
P1_SEND_PERCENT	0x14 3E00	20% PRI7, 62% PRI5
CPDMA_DMACONTROL	0xC001	Chan7, Chan6 are Rate Limited. Round-robin selection of DMA channel.
CPDMA_TX_PRI7_RATE	0x1 0013	200 Mbps
CPDMA_TX_PRI6_RATE	0x1 0005	~600 Mbps

24.11.4.8.7 Ethernet MAC Sliver (CPGMAC_SL)

The CPGMAC_SL peripheral shall be compliant to the IEEE Std 802.3 Specification. Half-duplex mode is supported in 10/100 Mbps mode, but not in 1000 Mbps (gigabit) mode.

Features:

- Synchronous 10/100/1000 Mbit operation
- G/MII Interface
- Hardware Error handling including CRC
- Full-Duplex Gigabit operation (half-duplex gigabit is not supported)
- EtherStats and 802.3Stats RMON statistics gathering support for external statistics collection module
- Transmit CRC generation selectable on a per channel basis
- Emulation Support
- VLAN Aware Mode Support
- Hardware flow control
- Programmable Inter Packet Gap (IPG).

24.11.4.8.7.1 G/MII Media Independent Interface

The following sections cover operation of the Media Independent Interface in 10/100/1000 Mbps modes. An IEEE 802.3 compliant Ethernet MAC controls the interface.

24.11.4.8.7.1.1 Data Reception

24.11.4.8.7.1.1.1 Receive Control

Data received from the PHY is interpreted and output. Interpretation involves detection and removal of the preamble and start of frame delimiter, extraction of the address and frame length, data handling, error checking and reporting, cyclic redundancy checking (CRC), and statistics control signal generation.

24.11.4.8.7.1.1.2 Receive Inter-Frame Interval

The 802.3 required inter-packet gap (IPG) is 24 GMII clocks (96 bit times) for 10/100 Mbit modes, and 12 GMII clocks (96 bit times) for 1000 Mbit mode. However, the MAC can tolerate a reduced IPG (2 GMII clocks in 10/100 mode and 5 GMII clocks in 1000 mode) with a correct preamble and start frame delimiter.

This interval between frames must comprise (in the following order):

- An Inter-Packet Gap (IPG).
- A seven octet preamble (all octets 0x55).
- A one octet start frame delimiter (0x5D).

24.11.4.8.7.1.2 Data Transmission

The Gigabit Ethernet Mac Sliver (GMII) passes data to the PHY when enabled. Data is synchronized to the transmit clock rate. The smallest frame that can be sent is two bytes of data with four bytes of CRC (6 byte frame).

24.11.4.8.7.1.2.1 Transmit Control

A jam sequence is output if a collision is detected on a transmit packet. If the collision was late (after the first 64 bytes have been transmitted) the collision is ignored. If the collision is not late, the controller will back off before retrying the frame transmission. When operating in full duplex mode the carrier sense (CRS) and collision sensing modes are disabled.

24.11.4.8.7.1.2.2 CRC Insertion

The MAC generates and appends a 32-bit Ethernet CRC onto the transmitted data, if the transmit packet header PASS_CRC bit is 0. For the CPMAC_SL generated CRC case, a CRC at the end of the input packet data is not allowed.

If the header word PASS_CRC bit is set, then the last four bytes of the TX data are transmitted as the frame CRC. The four CRC data bytes should be the last four bytes of the frame and should be included in the packet byte count value. The MAC performs no error checking on the outgoing CRC when the PASS_CRC bit is set.

24.11.4.8.7.1.2.3 MTXER

The MTXER signal is only used for EEE. If an underflow condition occurs on a transmitted frame, the frame CRC will be inverted to indicate the error to the network. Underflow is a hardware error.

24.11.4.8.7.1.2.4 Adaptive Performance Optimization (APO)

The Ethernet MAC port incorporates Adaptive Performance Optimization (APO) logic that may be enabled by setting the TX_PACE bit in the SL_MACCONTROL register. Transmission pacing to enhance performance is enabled when set. Adaptive performance pacing introduces delays into the normal transmission of frames, delaying transmission attempts between stations, reducing the probability of collisions occurring during heavy traffic (as indicated by frame deferrals and collisions) thereby increasing the chance of successful transmission.

When a frame is deferred, suffers a single collision, multiple collisions or excessive collisions, the pacing counter is loaded with an initial value of 31. When a frame is transmitted successfully (without experiencing a deferral, single collision, multiple collision or excessive collision) the pacing counter is decremented by one, down to zero.

With pacing enabled, a new frame is permitted to immediately (after one IPG) attempt transmission only if the pacing counter is zero. If the pacing counter is non zero, the frame is delayed by the pacing delay, a delay of approximately four inter-packet gap delays. APO only affects the IPG preceding the first attempt at transmitting a frame. It does not affect the back-off algorithm for re-transmitted frames.

24.11.4.8.7.1.2.5 Inter-Packet-Gap Enforcement

The measurement reference for the IPG of 96 bit times is changed depending on frame traffic conditions. If a frame is successfully transmitted without collision, and MCRS is de-asserted within approximately 48 bit times of MTXEN being de-asserted, then 96 bit times is measured from MTXEN. If the frame suffered a collision, or if MCRS is not de-asserted until more than approximately 48 bit times after MTXEN is de-asserted, then 96 bit times (approximately, but not less) is measured from MCRS.

The transmit IPG can be shortened by eight bit times when enabled and triggered. The TX_SHORT_GAP_EN bit in the SL_MACCONTROL register enables the TX_SHORT_GAP input to determine whether the transmit IPG is shorted by eight bit times.

24.11.4.8.7.1.2.6 Back Off

The Gigabit Ethernet Mac Sliver (GMII) implements the 802.3 binary exponential back-off algorithm.

24.11.4.8.7.1.2.7 Programmable Transmit Inter-Packet Gap

The transmit inter-packet gap (IPG) is programmable through the SL_TX_GAP register. The default value is decimal 12. The transmit IPG may be increased to the maximum value of 1FFh. Increasing the IPG is not compatible with transmit pacing. The short gap feature will override the increased gap value, so the short gap feature may not be compatible with an increased IPG.

24.11.4.8.7.1.2.8 Speed, Duplex and Pause Frame Support Negotiation

The CPMAC_SL can operate in half duplex or full duplex in 10/100 Mbit modes, and can operate in full duplex only in 1000 Mbit mode. Pause frame support is included in 10/100/1000 Mbit modes as configured by the host.

24.11.4.8.7.2 RMII Interface

The CPRMII peripheral is compliant to the RMII specification document.

24.11.4.8.7.2.1 Features

- Source Synchronous 10/100 Mbit operation
- Full and Half Duplex support

24.11.4.8.7.2.2 RMII Receive (RX)

The CPRMII receive (RX) interface converts the input data from the external RMII PHY (or switch) into the required MII (CPGMAC) signals. The carrier sense and collision signals are determined from the RMII input data stream and transmit inputs as defined in the RMII specification.

An asserted RMRXER on any di-bit in the received packet will cause an MRXER assertion to the CPGMAC during the packet. In 10Mbps mode, the error is not required to be duplicated on 10 successive clocks. Any di-bit which has an asserted RMII_RXER during any of the 10 replications of the data will cause the error to be propagated.

Any received packet that ends with an improper nibble boundary aligned RMCRSDV toggle will issue an MRXER during the packet to the CPGMAC. Also, a change in speed or duplex mode during packet operations will cause packet corruption.

The CPRMII can accept receive packets with shortened preambles, but 0x55 followed by a 0x5D is the shortest preamble that will be recognized (1 preamble byte with the start of frame byte). At least one byte of preamble with the start of frame indicator is required to begin a packet. An asserted RMCRSDV without at least a single correct preamble byte followed by the start of frame indicator will be ignored.

24.11.4.8.7.2.3 RMII Transmit (TX)

The CPRMII transmit (TX) interface converts the GMAC_SW MII input data into the RMII transmit format. The data is then output to the external RMII PHY.

The GMAC_SW does not source the transmit error (MII TXERR) signal. Any transmit frame from the CPGMAC with an error (underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be de-asserted at all times and is not an input into the CPRMII module. Zeroes are output on RMTXD[1:0] for each clock that RMTXEN is de-asserted.

24.11.4.8.7.3 RGMII Interface

The CPRGMII peripheral is compliant to the RGMII specification document.

24.11.4.8.7.3.1 RGMII Features

- Supports 1000/100/10 Mbps speed
- In SR1.1, Internal TXC delay on transmit is always enabled
- In SR2.x, Internal TXC delay on transmit can be enabled or disabled using bits [26] RGMII2_ID_MODE_N and [25] RGMII1_ID_MODE_N of the CTRL_CORE_SMA_SW_1 register.
- MII mode is not supported

24.11.4.8.7.3.2 RGMII Receive (RX)

The CPRGMII receive (RX) interface converts the source synchronous DDR input data from the external RGMII PHY into the required G/MII (CPGMAC) signals.

24.11.4.8.7.3.3 In-Band Mode of Operation

The CPRGMII is operating in the in-band mode of operation when the EXT_EN bit of the SL_MACCONTROL register is set to 1. The link status, duplexity, and speed are determined from the RGMII input data stream

RXD[3:0] when RX_CTL is deasserted, as defined in the RGMII specification. The PHY might need to be configured beforehand to output in-band data. The in-band data is indicated as shown in Table 24-884.

Table 24-884. In-Band Data

RXD3	RXD[2:1]		RXD0
Duplex status:	Link Speed:	RXC_CLK Speed:	Link Status:
0: half-duplex	00: 10-Mbps mode	2.5 MHz	0: Link is down
1: full-duplex	01: 100-Mbps mode	25 MHz	1: Link is up
	10: 1000-Mbps mode	125 MHz	
	11: reserved	reserved	

24.11.4.8.7.3.4 Forced Mode of Operation

The CPRGMII is operating in the forced mode of operation when the EXT_EN bit of the SL_MACCONTROL register is set to 0. In the forced mode of operation, the in-band data is ignored if present. The link status is forced high, and the duplexity and speed are determined from the SL_MACCONTROL[0] FULLDUPLEX and [7] GIG bits. If GIG = 1, then operation is gigabit mode. If the GIG is 0, the operation is 100 Mbps mode.

24.11.4.8.7.3.5 RGMII Transmit (TX)

The CPRGMII transmit (TX) interface converts the CPGMAC G/MII input data into the DDR RGMII format. The DDR data is then output to the external PHY.

The CPGMAC does not source the transmit error (TXERR) signal. Any transmit frame from the CPGMAC with an error (underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be de-asserted at all times and is not an input into the CPRGMII module.

The TXD[7:0] data bus uses only the lower nibble. The CPRGMII will output the lower nibble twice in 10/100 mode to avoid unnecessary signal switching.

Packets will be precluded from transmission through the CPRGMII module for 4096 transmit clocks after the rising edge of RGMII_LINK. Packet transmission will begin on the first TX_CTL rising edge after the 4096 transmit clock count has expired.

24.11.4.8.7.4 Frame Classification

Received frames are proper (good) frames if they are between 64 and RX_MAXLEN in length (inclusive) and contain no errors (code/align/CRC).

Received frames are long frames if their frame count exceeds the value in the SL_RX_MAXLEN register. The SL_RX_MAXLEN register reset (default) value is 1518 (decimal). Long received frames are either oversized or jabber frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment errors are jabber frames.

Received frames are short frames if their frame count is less than 64 bytes. Short frames that contain no errors are undersized frames. Short frames with CRC, code, or alignment errors are fragment frames.

A received long packet will always contain RX_MAXLEN number of bytes transferred to memory (if RX_CEF_EN = 1). An example with RX_MAXLEN = 1518 is:

- If the frame length is 1518, then the packet is not a long packet and there will be 1518 bytes transferred to memory.
- If the frame length is 1519, there will be 1518 bytes transferred to memory. The last three bytes will be the first three CRC bytes.
- If the frame length is 1520, there will be 1518 bytes transferred to memory. The last two bytes will be the first two CRC bytes.
- If the frame length is 1521, there will be 1518 bytes transferred to memory. The last byte will be the first CRC byte.

If the frame length is 1522, there will be 1518 bytes transferred to memory. The last byte will be the last data byte.

24.11.4.8.8 Embedded Memories

Table 24-885. Embedded Memories

Memory Type Description	Number of Instances	
Single-port 2560 × 64 RAM	3	(Packet FIFO's)
Single-port 64-word × 1152-bit RAM	1	(ALE)
Single-port 2048-word × 32-bit RAM	1	(CPPI)

24.11.4.8.9 Flow Control

There are two types of switch flow control: CPPI port flow control and Ethernet port flow control. The CPPI and Ethernet port naming conventions for data flow into and out of the switch are reversed. For the CPPI port (port 0), transmit operations move packets from external memory into the switch and then out to either or both Ethernet transmit ports (ports 1 and 2). CPPI receive operations move packets that were received on either or both Ethernet receive ports to external memory.

24.11.4.8.9.1 CPPI Port Flow Control

The CPPI port has flow control available for transmit (switch ingress). CPPI receive operations (switch egress) do not require flow control. CPPI Transmit flow control is initiated when enabled and triggered. CPPI transmit flow control is enabled by setting the P0_FLOW_EN bit in the CPSW_FLOW_CONTROL register. CPPI transmit flow control is enabled by default on reset because host packets should not be dropped in any mode of operation.

24.11.4.8.9.2 Ethernet Port Flow Control

The Ethernet ports have flow control available for transmit and receive. Transmit flow control stops the Ethernet port from transmitting packets to the wire (switch egress) in response to a received pause frame. Transmit flow control does not depend on FIFO usage.

The ethernet ports have flow control available for receive operations (packet ingress). Ethernet port receive flow control is initiated when enabled and triggered. Packets received on an ethernet port can be sent to the other ethernet port or the CPPI port (or both). Each destination port can trigger the receive ethernet port flow control. An ethernet destination port triggers another ethernet receive flow control when the destination port is full.

When a packet is received on an ethernet port interface with enabled flow control the below occurs:

- The packet will be sent to all ports that currently have room to take the entire packet.
- The packet will be retried until successful to all ports that indicate they don't have room for the packet.

The flow control trigger to the CPGMAC_SL will be asserted until the packet has been sent, and there is room in the logical receive FIFO for packet runout from another flow control trigger (RX_PKT_CNT = 0). Ethernet port receive flow control is disabled by default on reset. Ethernet port receive flow control requires that the RX_FLOW_EN bit in the associated CPGMAC_SL be set to 1. When receive flow control is enabled on a port, the port's associated FIFO block allocation must be adjusted. The port RX allocation must increase from the default three blocks to accommodate the flow control runout. A corresponding decrease in the TX block allocation is required. If a sending port ignores a pause frame then packets may overrun on receive (and be dropped) but will not be dropped on transmit. If flow control is disabled for G/MII ports, then any packets that are dropped are dropped on transmit and not on receive.

24.11.4.8.9.2.1 Receive Flow Control

When enabled and triggered, receive flow control is initiated to limit the CPGMAC_SL from further frame reception. Half-duplex mode receive flow control is collision based while full duplex mode issues 802.3X pause frames. In either case, receive flow control prevents frame reception by issuing the flow control appropriate for the current mode of operation. Receive flow control is enabled by the RX_FLOW_EN bit in the SL_MACCONTROL register. Receive flow control is triggered (when enabled) when the RX_FLOW_TRIGGER input is asserted. The CPGMAC_SL is configured for collision or IEEE 802.3X flow control via the FULLDUPLEX bit in the SL_MACCONTROL register.

24.11.4.8.9.2.1.1 Collision Based Receive Buffer Flow Control

Collision-based receive buffer flow control provides a means of preventing frame reception when the port is operating in half-duplex mode (FULLDUPLEX is cleared in SL_MACCONTROL). When receive flow control is enabled and triggered, the port will generate collisions for received frames. The jam sequence transmitted will be the twelve byte sequence C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3 (hex). The jam sequence will begin no later than approximately as the source address starts to be received. Note that these forced collisions will not be limited to a maximum of 16 consecutive collisions, and are independent of the normal back-off algorithm. Receive flow control does not depend on the value of the incoming frame destination address. A collision will be generated for any incoming packet, regardless of the destination address.

24.11.4.8.9.2.1.2 IEEE 802.3X Based Receive Flow Control

IEEE 802.3x based receive flow control provides a means of preventing frame reception when the port is operating in full-duplex mode (FULLDUPLEX is set in SL_MACCONTROL). When receive flow control is enabled and triggered, the port will transmit a pause frame to request that the sending station stop transmitting for the period indicated within the transmitted pause frame.

The CPGMAC_SL will transmit a pause frame to the reserved multicast address at the first available opportunity (immediately if currently idle, or following the completion of the frame currently being transmitted). The pause frame will contain the maximum possible value for the pause time (FFFFh). The MAC will count the receive pause frame time (decrements FF00h down to 0) and retransmit an outgoing pause frame if the count reaches zero. When the flow control request is removed, the MAC will transmit a pause frame with a zero pause time to cancel the pause request.

Note that transmitted pause frames are only a request to the other end station to stop transmitting. Frames that are received during the pause interval will be received normally (provided the RX FIFO is not full).

Pause frames will be transmitted if enabled and triggered regardless of whether or not the port is observing the pause time period from an incoming pause frame.

The CPGMAC_SL will transmit pause frames as:

- The 48-bit reserved multicast destination address 01.80.C2.00.00.01.
- The 48-bit source address - from SL_SA[47:0] input.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause time value FF.FF. A pause-quantum is 512 bit-times. Pause frames sent to cancel a pause request will have a pause time value of 00.00.
- Zero padding to 64-byte data length (The CPGMAC_SL will transmit only 64 byte pause frames).
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

If RX_FLOW_EN is cleared to 0 while the pause time is nonzero, then the pause time will be cleared to 0 and a zero count pause frame will be sent.

24.11.4.8.9.2.2 Transmit Flow Control

Incoming pause frames are acted upon, when enabled, to prevent the CPGMAC_SL from transmitting any further frames. Incoming pause frames are only acted upon when the FULLDUPLEX and TX_FLOW_EN bits in the SL_MACCONTROL register are set. Pause frames are not acted upon in half-duplex mode. Pause frame action will be taken if enabled, but normally the frame will be filtered and not transferred to memory. MAC control frames will be transferred to memory if the RX_CMF_EN (copy MAC frames) bit in the SL_MACCONTROL register is set. The TX_FLOW_EN and FULLDUPLEX bits effect whether or not MAC control frames are acted upon, but they have no effect upon whether or not MAC control frames are transferred to memory or filtered.

Pause frames are a subset of MAC Control Frames with an opcode field = 0001h. Incoming pause frames will only be acted upon by the port if:

- TX_FLOW_EN is set in SL_MACCONTROL register, and
- the frame's length is 64 to SL_RX_MAXLEN bytes inclusive, and
- the frame contains no CRC error or align/code errors.

The pause time value from valid frames will be extracted from the two bytes following the opcode. The pause time will be loaded into the port's transmit pause timer and the transmit pause time period will begin.

If a valid pause frame is received during the transmit pause time period of a previous transmit pause frame then:

- if the destination address is not equal to the reserved multicast address or any enabled or disabled unicast address, then the transmit pause timer will immediately expire, or
- if the new pause time value is zero then the transmit pause timer will immediately expire, else
- the port transmit pause timer will immediately be set to the new pause frame pause time value. (Any remaining pause time from the previous pause frame will be discarded).

If TX_FLOW_EN in SL_MACCONTROL register is cleared, then the pause-timer will immediately expire.

The port will not start the transmission of a new data frame any sooner than 512-bit times after a pause frame with a non-zero pause time has finished being received (MRXDV going inactive). No transmission will begin until the pause timer has expired (the port may transmit pause frames in order to initiate outgoing flow control). Any frame already in transmission when a pause frame is received will be completed and unaffected.

Incoming pause frames consist of the below:

- A 48-bit destination address equal to:
  - The reserved multicast destination address 01.80.C2.00.00.01
  - , or the CPGMAC_SL SL_SA [47:0] input.
- The 48-bit source address of the transmitting device.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause_time. A pause-quantum is 512 bit-times.
- Padding to 64-byte data length.
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

The padding is required to make up the frame to a minimum of 64 bytes. The standard allows pause frames longer than 64 bytes to be discarded or interpreted as valid pause frames. The CPGMAC_SL will recognize any pause frame between 64 bytes and RX_MAXLEN bytes in length.

24.11.4.8.10 Short Gap

The port 1 (and port 2) transmit inter-packet gap (IPG) may be shortened by eight bit times when enabled and triggered. The TX_SHORT_GAP_EN bit in the SL_MACCONTROL register enables the gap to be shortened when triggered. The condition is triggered when the port 1 (port 2) transmit FIFO has a user defined number of FIFO blocks used. The port 1 transmit FIFO blocks used determines if the port 1 gap is shortened, and the port 2 transmit FIFO blocks used determines if the port 2 gap is shortened. The CPSW_GAP_THRESH register value determines the port 1 short gap threshold, and the CPSW_GAP_THRESH register value determines the port 2 short gap threshold.

24.11.4.8.11 Switch Latency

The CPSW_3G is a store and forward switch. The switch latency is defined as the amount of time between the end of packet reception of the received packet to the start of the output packet transmit.

Table 24-886. Switch Latency

Mode	Latency
Gig (1000)	880 ns
100	1.3 µs
10	6.5 µs

24.11.4.8.12 Emulation Control

The emulation control input (EMUSUSP) and submodule emulation control registers allow CPSW_3G operation to be completely or partially suspended. There are three CPSW_3G submodules that contain emulation control registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). The submodule emulation control registers must be accessed to facilitate CPSW_3G emulation control. The CPSW_3G module enters the emulation suspend state if all three submodules are configured for emulation suspend and the emulation suspend input is asserted. A partial emulation suspend state is entered if one or two submodules is configured for emulation suspend and the emulation suspend input is asserted. Emulation suspend occurs at packet boundaries. The emulation control feature is implemented for compatibility with other peripherals.

CPGMAC_SL Emulation Control

The emulation control input (TBEMUSUP) and register bits (SOFT and FREE bits in the SL_EMCONTROL register) allow CPGMAC_SL operation to be suspended. When the emulation suspend state is entered, the CPGMAC_SL will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For receive, frames that are detected by the CPGMAC_SL after the suspend state is entered are ignored. Emulation control is implemented for compatibility with other peripherals.

CPDMA Emulation Control

The emulation control input (TBEMUSUP) and register bits (SOFT and FREE bits in the CPDMA_EMCONTROL register) allow CPDMA operation to be suspended. When the emulation suspend state is entered, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell FIFO will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Emulation control is implemented for compatibility with other peripherals

Table 24-887 shows the operations of the emulation control input and register bits.

Table 24-887. Emulation Control Input

EMUSUSP	SOFT	FREE	Description
0	X	X	Normal Operation
1	0	0	Normal Operation
1	1	0	Emulation Suspend
1	X	1	Normal Operation

24.11.4.8.13 FIFO Loopback

FIFO loopback mode is entered when the FIFO_LOOPBACK bit in the CPSW_CONTROL register is set. FIFO loopback mode causes packets received on a port to be turned around and transmitted back on the same port. Port 0 receive is fixed on channel 0 in FIFO loopback mode. The RXSOFOVERRUN statistic is incremented for each packet sent in FIFO loopback mode. Packets sent in with errors are returned with errors (they are not dropped). FIFO loopback is intended as a simple mechanism for test purposes. FIFO loopback should be performed in fullduplex mode only.

24.11.4.8.14 Device Level Ring (DLR) Support

Device Level Ring (DLR) support is enabled by setting the DLR_EN bit in the CPSW_CONTROL register. When enabled, incoming received DLR packets are detected and sent to queue 3 (highest priority) of the egress port(s). If the host port is the egress port for a DLR packet then the packet is sent on the CPDMA Rx channel selected by the P0_DLR_CPDMA_CH field in the P0_CONTROL register. The supervisor node MAC address feature is supported with the dlr_unicast bit in the unicast address table entry. When set, the dlr_unicast bit causes a packet with the matching destination address to be flooded to the vlan_member_list minus the receive port and minus the host port (the port_number field in the unicast address table entry is a don't care). Matching dlr_unicast packets are flooded regardless of whether the packet is a DLR packet or not. The EN_P0_UNI_FLOOD bit in the ALE_CONTROL register has no effect on DLR unicast packets. Packets are determined to be DLR packets as shown below:

1. DLR is enabled (DLR_EN is set in the CPSW_CONTROL).
2. One of the sequences below are true.
   a. The first packet ltype matches CPSW_VLAN_LTYPE[15:0] VLAN_LTYPE1 and Px_CONTROL[20] Px_VLAN_LTYPE1_EN is set and the second packet ltype matches CPSW_DLR_LTYPE[15:0] DLR_LTYPE.
   b. The first packet ltype matches CPSW_VLAN_LTYPE[31:16] VLAN_LTYPE2 and Px_CONTROL[21] Px_VLAN_LTYPE2_EN is set and the second packet ltype matches CPSW_DLR_LTYPE[15:0] DLR_LTYPE.
   c. The first packet ltype matches CPSW_VLAN_LTYPE[15:0] VLAN_LTYPE1 and Px_CONTROL[20] Px_VLAN_LTYPE1_EN is set and the second packet ltype matches CPSW_VLAN_LTYPE[31:16] VLAN_LTYPE2 and Px_CONTROL[21] Px_VLAN_LTYPE2_EN is set and the third packet ltype matches CPSW_DLR_LTYPE[15:0] DLR_LTYPE.

24.11.4.8.15 Energy Efficient Ethernet Support (802.3az)

Note

CM_GMAC_CLKSTCTRL[1:0]CLKTRCTRL bit field must not be programmed for SW_SLEEP or HW_AUTO.

Energy Efficient Ethernet (EEE) allows the PRCM to turn off the module clock during inactive periods as determined by network and host traffic. The module can then be awakened by host queued transmit packet(s) or by a port's external Ethernet PHY. EEE operations are configured as shown below:

1. The 10-bit EEE clock pre-scale value is written to the CPSW_EEE_PRESCALE register. The pre-scaler is used to clock all EEE-related counters
2. The port Idle to LPI count values (Px_IDLE2LPI) are written with the desired values
3. The port LPI to Wake count values (Px_LPI2WAKE) are written with the desired values
4. The EEE_EN bit is set in the switch CPSW_CONTROL register
5. Set the WR_CONTROL[8] SS_EEE_EN bit to enable energy efficient operations at subsystem level.

EEE operation can begin after configuration. The host allows (through PRCM) the CPSW_3G to enter a low power state by asserting the CLKSTOP_REQ signal. There are no requirements on host queues or traffic in order for the host to assert or de-assert CLKSTOP_REQ to the CPSW_3G.

Each ethernet port has a transmit and a receive LPI (low power indicate) state. The receive LPI state is entered when the port's corresponding PHY indicates the LPI state via the CPSW_3G GMII interface. The PHY indicates LPI by asserting MRXER with a MRXD[7:0] value of 0x01 while MRXDV is deasserted (inter-packet gap). The Ethernet transmit port indicates LPI after the Px_IDLE2LPI value has been counted (the transmit port has gone idle for the configured amount of time). If another packet is received for transmit during the count then the count is restarted. When the transmit port has been idle for the Idle to LPI time, the transmit port enters the LPI state and indicates LPI to the associated PHY. The LPI is indicated to the external PHY by an asserted MTXER with a MTXD[7:0] while MTXEN is deasserted (inter-packet gap). The CPDMA LPI state includes transmit and receive. The CPDMA LPI state is entered when the CPDMA transmit and receive have both been idle for the Idle to LPI time (P0_IDLE2LPI). The Idle to LPI time value for all ports must be large relative to the switch latency to ensure that the count is not able to complete between successive packets.

Note

The procedure above is described for the GMII interfaces at the CPSW_3G boundary. External PHY signaling has the following conditions:

- GMII interface is used only in MII mode. Therefore, only MRXD[3:0] and MTXD[3:0] are pinned out
- RGMII is a DDR interface. TXEN and TXER are the sampled values of TX_CTL at the rising and the falling TXC_CLK edges, respectively. RXDV and RXER are the sampled values of RX_CTL at the rising and the falling RXC_CLK edges, respectively
- In RMII mode, EEE is not supported.

When all transmit and receive ports are in the LPI state (CPSW LPI state), the CLKSTOP_ACK signal is asserted, and the PRCM is allowed to stop the module clock. When CLKSTOP_ACK is asserted, the clock may be turned on and off as desired by the host. The host is allowed to restart the clock, perform slave read/write operations to the CPSW_3G memory address space, and then turn off the clock again while CLKSTOP_ACK is asserted.

The software can remove and disable from re-entering the CPSW LPI state by restarting the module clock and then de-asserting CLKSTOP_REQ. The host may queue transmit packets at any time including without regard to the CPSW_3G LPI state (the clock must be restarted in order to write the CPSW_3G slave address space as described above). Host writes to transmit head descriptor pointers will cause the CLKSTOP_WAKEUP signal to be asserted if the CPSW_3G is in the low power state (if CLKSTOP_ACK is asserted).

The external Ethernet PHY's can also wakeup the PRCM by removing the Ethernet receive LPI indication. If the module is in the CPSW Idle state with CLKSTOP_ACK asserted and the receive LPI indication is removed, the CLKSTOP_WAKEUP signal will be asynchronously asserted. On wakeup, the PRCM restarts the clock and de-assert the CLKSTOP_REQ signal. The CLKSTOP_WAKEUP signal will be synchronously deasserted with CLKSTOP_ACK. Upon the deassertion of CLKSTOP_REQ, the Ethernet ports will count the Px_LPI2WAKE time for each port at which time the port is available for transmit. Upon the de-assertion of CLKSTOP_REQ, the CPDMA transmit will count the P0_LPI2WAKE count at which time the CPDMA will begin to send any packets that the host has queued (switch ingress). The wait time on CPDMA transmit is included to preclude the host from filling up the Ethernet port transmit FIFO's while the Ethernet ports are in the LPI to wake time. There is no LPI to wake time on CPDMA receive (switch egress).

24.11.4.8.16 CPSW_3G Network Statistics

The CPSW_3G has a set of statistics that record events associated with frame traffic on selected switch ports. The statistics values are cleared to zero 38 clocks after the rising edge of GMAC_RST. When one or more port enable (Pn_STAT_EN) bits in the CPSW_STAT_PORT_EN register are set, all statistics registers are write to decrement. The value written will be subtracted from the register value with the result being stored in the register. If a value greater than the statistics value is written, then zero will be written to the register (writing 0xFFFF FFFF clears a statistics location). When all port enable bits are cleared to zero, all statistics registers are read/write (normal write direct, so writing 0x0000 0000 clears a statistics location). All write accesses must be 32-bit accesses. In the below statistics descriptions, "the port" refers to any enabled port (with a corresponding set Pn_STAT_EN bit).

The statistics interrupt (STAT_PEND) will be issued if enabled when any statistics value is greater than or equal to 0x8000 0000. The statistics interrupt is removed by writing to decrement any statistics value greater than 0x8000 0000. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from 0xFFFF FFFF to 0x0000 0000.

See *STATS Registers* for description of every network statistic.

Rx Statistics Summary and *Tx Statistics Summary* summarize network statistics.

24.11.4.8.16.1

Table 24-888. Rx Statistics Summary

Rx Statistic	Frame/Oct	Rx/Rx+Tx	Frame Type					Frame Size (bytes)								Event				
			MAC control		Data[5]															
			Pause frame	Non-pause[4]	Multicast	Broadcast	Unicast	<64	64	65-127	128-255	256-511	512-1023	1024-rx_max len	>rx_max len	Flow Coll.[8]	CRC Error	Align/Code	Overrun	Addr. Disc.
Good Rx Frames	F	R	(y|[1]	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-[2]	n	n	-	n
Broadcast Rx Frames	F	R	(%|[6]	%|	n	y)	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Multicast Rx Frames	F	R	(%|	%|	y)	n	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Pause Rx Frames	F	R	y	n	n	n	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	-
Rx CRC Errors	F	R	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	y	n	-	n
Rx Align/Code Errors	F	R	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	-	y	-	n
Oversized Rx Frames	F	R	(y|	y|	y|	y|	y)	n	n	n	n	n	n	n	y	-	n	n	-	n
Rx Jabbers	F	R	(y|	y|	y|	y|	y)	n	n	n	n	n	n	n	y	-	(y|	y)	-	n
Undersized Rx Frames	F	R	n	n	(y|	y|	y)	y	n	n	n	n	n	n	n	-	n	n	-	n
Rx Fragments	F	R	n	n	(y|	y|	y)	y^[7]	n	n	n	n	n	n	n	-	(y|	y)	-	-
Rx Overruns[9]	F	R	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y|	y)	-	-	-	y	n
64octet Frames	F	R+T[3]	(y|	y|	y|	y|	y)	n	y	n	n	n	n	n	n	-	-	-	-	n
65-127octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	y	n	n	n	n	n	-	-	-	-	n
128-255octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	y	n	n	n	n	-	-	-	-	n
256-511octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	y	n	n	n	-	-	-	-	n
512-1023octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	n	y	n	n	-	-	-	-	n
1024-UPoctet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	n	n	y	n	-	-	-	-	n
Rx Octets	O	R	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Net Octets	O	R+T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y|	y|	y)	-	-	-	-

(1) "AND" is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning "OR" is indicated. Parentheses are significant.
(2) "-" indicates conditions which are ignored in the formations of the statistic.
(3) Statistics marked "R+T" are formed by summing the Rx and Tx statistics, each of which is formed independently.
(4) The non-pause column refers to all MAC control frames (for example, frames with length/type=88.08) with opcodes other than 0x0001. The pauseframe column refers to MAC frames with the opcode=0x0001.
(5) The multicast, broadcast and unicast columns in the table refer to non-MAC Control/non-pause frames (i.e. data frames).
(6) "%" If either a MAC control frame or pause frame has a multicast or broadcast destination address then the appropriate statistics will be updated.
(7) "y^" Frame fragments are not counted if less than 8 bytes.
(8) Flow coll. are half-duplex collisions forced by the MAC to achieve flow-control. A collision will be forced during the first 8 bytes so should not show in frame fragments. Some of the '-'s in this column might in reality be 'n's.
(9) The rx_overruns stat is for RX_MOF_OVERRUNS and RX_SOF_OVERRUNS added together.

Table 24-889. Tx Statistics Summary

Tx Statistic[9]	Frame/Oct	Tx/Rx+Tx	Frame Type					Frame Size (bytes)							Event									
			MAC control[4]		Data			64	65-127	128-255	256-511	512-1023	1024-1535	>1535	CRC Error	Collision Type					No Carrier	Queued	Deferred	Underrun
			Pause-MAC	Any-CPU	Multicast	Broadcast	Unicast									Flow[8]	1	2-15	16	Late				
Good Tx Frames	F	T	(y|[1]	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-[2]	-	-	-	n	n	n	-	-	n
Broadcast Tx Frames	F	T	n	(%|[5]	n	y)	n	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Multicast Tx Frames	F	T	(y|	%|	y)	n	n	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Pause Tx Frames	F	T	y	n	n	n	n	y	n	n	n	n	n	n	-	-	-	-	-	-	-	-	-	-
Collisions	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	(+[6]	+	+	+	+)	n	-	-	-
Single Collision Tx Frames	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	y	n	n	n	n	-	-	-
Multiple Collision Tx Frames	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	y	n	n	n	-	-	-
Excessive Collisions	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	n	y	n	n	-	-	-
Late Collisions	F	T	n	(y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	-	-	-	-	-	y	-	-	-	-
Deferred Tx Frames	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	n	n	n	n	-	y	n
Carrier Sense Errors	F	T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	-	-	y	-	-	-
64octet Frames	F	R+T[3]	(y|	y|	y|	y|	y)	y	n	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
65-127octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	y	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
128-255octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	y	n	n	n	n	-	-	-	-	n	n	n	-	-	-
256-511octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	y	n	n	n	-	-	-	-	n	n	n	-	-	-
512-1023octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	y	n	n	-	-	-	-	n	n	n	-	-	-
1024-UPoctet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	n	y	y	-	-	-	-	n	n	n	-	-	-
Tx Octets	O	T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Net Octets	O	R+T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	$[7]	$	$	$	$	-	-	-

(1) "AND" is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning “OR” is indicated. Parentheses are significant.
(2) “-“ indicates conditions which are ignored in the formations of the statistic.
(3) Statistics marked “R+T” are formed by summing the Rx and Tx statistics, each of which is formed independently.
(4) Pause (MAC) frames are issued in the MAC as perfect (no CRC error) 64 byte frames in full duplex only, so they cannot collide.
(5) “%” If a CPU sourced MAC control frame has a multicast or broadcast destination address then the appropriate statistics will be updated.

(6) “+” indicates collisions which are “summed” (i.e. every collision is counted in the Collisions statistic). Jam sequences used for halfduplex flow control are also counted.
(7) “$” Every byte written on the wire during each retry attempt is also counted in addition to frames which experience no collisions or carrier loss.
(8) The flow collision type is for half-duplex collisions forced by the MAC to achieve flow control. Some of the ‘-‘s in this column might in reality be ‘n’s. To prevent double-counting, Net Octets are unaffected by the jam sequence – the ‘received’ bytes, however, are counted. (See Table 24-888.)
(9) When the transmit Tx FIFO is drained due to the MAC being disabled or link being lost, then the frames being purged will not appear in the Tx statistics.

24.11.4.9 Static Packet Filter (SPF)

24.11.4.9.1 SPF Overview

The Static Packet Filter (SPF) provides protection against some of the most common forms of Denial of Service (DoS) attacks that exploit the Layer 3 and Layer 4 functions of the network stack running on a system. The DoS attacks that are mitigated by the SPF module are:

- ARP Flood
- Fraggle
- Illegal Fragment Offset
- Illegal TCP Options
- Jolt2
- Jolt
- Land
- Micro Fragment
- Null Scan
- Ping Flood
- Short ICMP Packet
- Smurf
- SSPing
- SYN Fragment
- TCP SYN Flood
- Xmas Scan

24.11.4.9.2 SPF Functional Description

24.11.4.9.2.1 SPF Block Diagram

Figure 24-198 shows the block diagram of the Static Packet Filter module. There are three main components of the static packet filter – packet header extractor to decode packet formats, rule engine to check for malformed protocol header fields and a host interface to provide control and configuration access to a host processor and to keep a log of filtered packets in system memory.

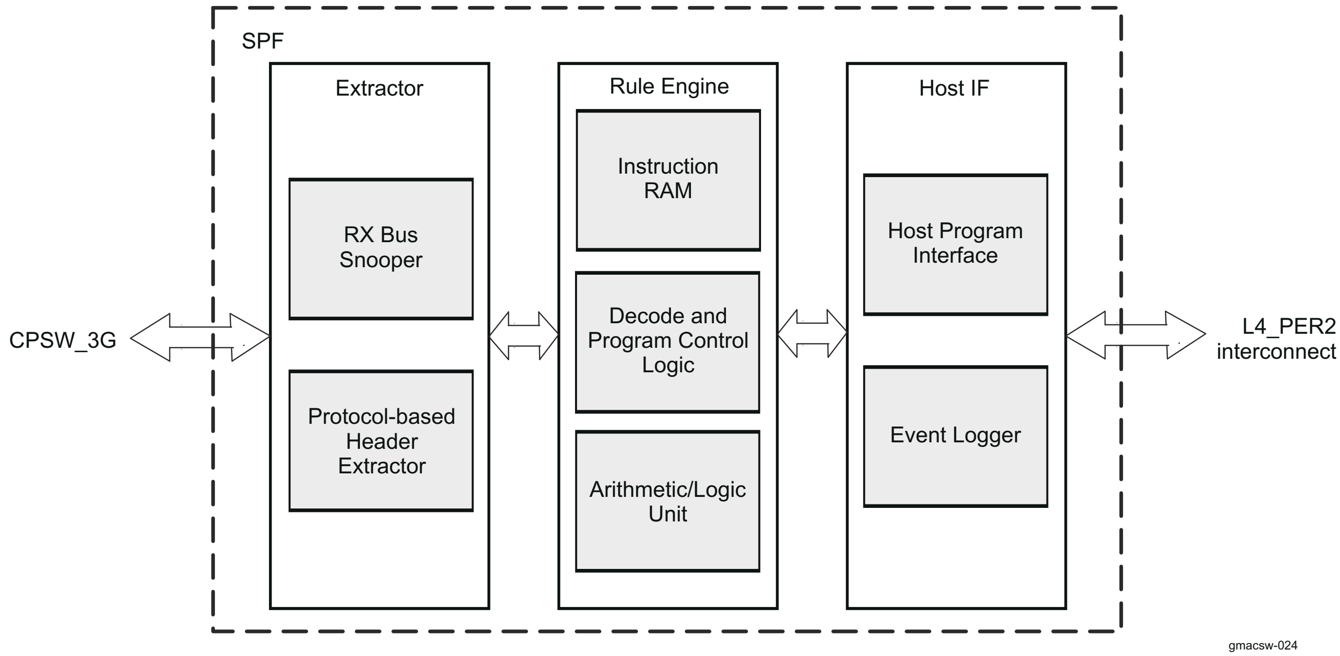


Figure 24-198. SPF Block Diagram

The packet parser interfaces to the three-port Ethernet switch (CPSW_3G). The CPSW_3G signals are snooped for receive traffic and the packet contents are provided to the packet header extractor. The parser can decode various network packet formats and determine the location of the header corresponding to each of the specified protocols and store the values in internal registers. The packet parser can decode VLAN, PPPoE, IP, TCP, UDP and ICMP frame formats and determine the location of each protocol header in a frame. The location of each of these headers is used by the rule engine for performing checks against preprogrammed conditions.

The rule engine monitors information in the header fields and upon detection of an abnormal combination of values in these packet header fields, issues an instruction to drop the packet to the external RFIFO interface. The rule engine can also monitor the receive rate of a particular class of packets and can limit the number of packets that actually pass through the system. The operation of rule engine is programmable and that gives flexibility to perform a range of different checks on the contents of packet to determine whether or not it should be accepted.

The event logger captures the activity in the packet filter. In addition, based upon the settings programmed by the host software, the event logger writes detailed information about any frames that have been dropped thus far. This information is written to a part of the system memory as configured by the host software.

24.11.4.9.2.2 Interrupts

The Static Packet Filter has one interrupt that is used to inform the host about excessive number of logged packet drops. The addresses of the instructions (or drop codes) in the instruction memory that cause packet to be dropped are associated with a threshold. When the threshold is met, the packet drop is logged. Each time a packet drop is logged, it is counted. When the number of logged records reaches the value specified by the SPF_INTCNT register, the host is interrupted. The threshold settings can be controlled by host software to limit the frequency of interrupts. Writing a zero to SPF_INTCNT disables the interrupt in this scenario. Whenever interrupt is enabled and is triggered, it can be cleared by writing one to either Raw register or Masked register.

Interrupts are controlled by the following registers:

1. Raw register SPF_INT_RAW: Holds the Raw state of interrupt, that is, without mask. The hardware interrupt is latched in this register and is only cleared when a one is written to either Raw register or Masked register. Writing a zero has no effect on this register.

2. Masked register SPF_INT_MASKED: This is the actual interrupt given to the system. It is the result of bitwise AND operation of Raw and Mask registers. Once the interrupt is sensed by the system, this register should be cleared by writing one to this register or to Raw register during ISR execution. Writing zero has no effect on this register
3. Mask set register SPF_MASK_SET: Writing one to this register enables the interrupt. By default, the interrupt is disabled and needs to be enabled by writing a one to this register. Writing a zero has no effect on this register. Writing to this register also set clear register.
4. Mask clear register SPF_MASK_CLR: Writing one to this register disables interrupt. Writing a zero has no effect on this register.

24.11.4.9.2.3 Protocol Header Extractor

The extractor module has a protocol aware state machine. It decodes the header field in the current protocol header to determine the encapsulating protocol type and extracts various parameters used for software logging. The extractor module also provides the offset to octets in the packet, where Layer 3 and Layer 4 protocol headers start, to the rule engine. These protocol headers correspond to the location of IP and TCP/UDP/ICMP headers from the beginning of the packet.

Ethernet packets with VLAN, PPPoE, IP, IP Options, and ICMP/TCP/UDP protocol are supported by the extractor. When unknown protocols are detected, the extractor skips any additional packet processing. Some examples of how packet parsing is done by the Extractor are given below.

- Ethernet – VLAN - PPPoE – IP – IP Options –TCP/UDP/ICMP – Payload
- Ethernet – VLAN – PPPoE – IP – IP options – Unknown – Payload
- Ethernet – VLAN – PPPoE – IP – TCP/UDP/ICMP – Payload
- Ethernet – VLAN – PPPoE – IP – Unknown – Payload
- Ethernet – VLAN – PPPoE – Fragmented IP – Unknown – Payload
- Ethernet – VLAN – IP – IP Options –TCP/UDP/ICMP – Payload
- Ethernet – VLAN – IP – IP options – Unknown – Payload
- Ethernet – VLAN – IP – TCP/UDP/ICMP – Payload
- Ethernet – VLAN – IP – Unknown – Payload
- Ethernet – VLAN – Fragmented IP – Unknown – Payload
- Ethernet – IP – IP Options –TCP/UDP/ICMP – Payload
- Ethernet – IP – IP options – Unknown – Payload
- Ethernet – IP – TCP/UDP/ICMP – Payload
- Ethernet – IP – Unknown – Payload
- Ethernet – Fragmented IP – Unknown – Payload
- Ethernet – Unknown

The flow of Extractor state machine flow is as follows.

1. Wait for start of packet and proceed to step 2 if start of packet detected.
2. Read Ethernet header Length/Type field (at offset 13). Go to step 3 if next header is VLAN, to step 4 if it is PPP header, to step 5 if it is IP header and to step 8 otherwise.
3. Allow VLAN header to pass and read the Length/Type field. Go to step 4 if next header is PPP, to step 5 if it is IP header and to step 8 otherwise.
4. Read the protocol type in PPP header. Go to step 5 if protocol is IP and to step 8 for unknown protocol
5. Extract source IP address and destination IP address from the IP header. Determine if the packet has IP options or if the packet is fragmented. Go to step 6 if the packet has IP options and to step 8 if the packet is fragmented. If there are no options and the next protocol header is TCP/UDP, go to step 7. Otherwise, skip processing and go to step 8 because the packet has unrecognized protocol header
6. Skip IP options and determines the next protocol header. Go to step 7 if next header is TCP/UDP/ICMP. In case the packet contains an IP fragment or if the protocol in the next header is unknown, skip processing and go to step 8.
7. The packet has TCP/UDP/ICMP header. Extract TCP/UDP source and destination port numbers for logging or ICMP type and code fields and go to step 8.
8. Wait for end of frame and go to step 1 at end of frame.

In case the packet is aborted (indicated by address 0x10 on VBUSP with request and write ready asserted), the extractor state machine flushes its current state, goes back to the idle state and waits for next packet.

The Extractor provides information to rule engine about location of Layer 3 and Layer 4 protocols. When this information is provided to rule engine, the Base Register 1 and Base register 2 are loaded with Layer 3 start offset and Layer 4 start offset respectively. Until the extractor completes decoding of these protocol headers and sends the offset values to rule engine, the rule engine does not execute any instruction that operates on fields in these protocol headers.

Extractor module extracts IP protocol, Source IP address, Destination IP address, TCP/UDP source port, TCP/UDP destination port and ICMP type/code from incoming packet and provides this information to the Host logger module for software reporting. In case IP header or TCP/UDP header does not contain the extractable fields then the corresponding field is logged as zero.

In case packet has unknown Layer 4 protocol or IP packet is fragmented such that the packet does not contain TCP/UDP/ICMP header then TCP/UDP source/destination ports or ICMP type/code fields are logged as zeros.

24.11.4.9.2.4 Programmable Rule Engine

The rule engine is a micro-coded machine that is programmed by the host software. The rule engine is programmed to evaluate various expressions involving the header fields of different protocols that the incoming packet belongs to. These expressions are coded into the instructions that are stored in an internal RAM and are used to check whether a packet is mal-formed or is potentially a DoS attack packet. The rule engine executes the instructions for each incoming packet and makes a decision about accepting or rejecting the packet.

Since each packet can have multiple distinct protocol headers, the location of the headers can be different from packet to packet. The rule engine gets information about the location of the protocol headers from the packet header extractor. The packet header extractor can decode several different protocol types. The octet number at which a particular protocol header is stored is loaded into SPF Base Registers. An application may require SPF to filter packets based on protocols that are not decoded by the extractor. In such cases, SPF allows bypassing the header extractor and then the rule engine can be programmed to analyze packets and figure out the location of each header.

The Rule Engine instructions are programmed before the SPF module is enabled. The instructions cannot be overwritten while the rule engine is processing packets. To modify the contents of code RAM during operation, SPF must be disabled temporarily and then new instructions can be loaded.

Once SPF is enabled, the rule engine starts to fetch instructions one at a time. For each instruction, the operands are obtained from either the packet octets that are being received, from internal registers or from the immediate values inside the instruction itself. If the octet that is needed for execution has not yet been received, then the execution stalls until the required octet is received. In case the operand specified in the instruction refers to a packet octet that has already gone by and is not available in the packet buffer, then the execution stalls until the end of packet. An instruction is executed only when all the required operands are available. Based on the instruction execution results, the packet may immediately be dropped or the results of the evaluated expressions may be stored for future use. In addition, the current instruction can also cause the rule engine to skip a specified number of instructions (immediately following the current instruction) and resume from another location in the instruction RAM.

The rule engine operates on multiple operands and performs multiple tasks in each clock cycle. In each cycle, it can perform one arithmetic and/or logical operation on two pairs of operands. The operands are masked with a 32-bit mask that is generated from the information provided in the instruction. The mask allows for operations that involve variable sized operands. Each operation generates a 32-bit number and a flag bit. The 32-bit number is typically either the sum or the output of bit-wise logical operation. The flag is a single bit result of a comparison operation. The result of each operation can be saved in the internal registers if a specified condition is satisfied. Similarly, depending on the result of the operation, the rule engine can jump to another location in the instruction memory. The program can instruct the rule engine to either perform two save operations, two conditional jumps or one jump and one save. The conditions must be mutually exclusive to prevent unspecified operation. In addition to providing an alternate path for execution, the instruction can cause the rule engine to accept or reject the current packet and exit from the program until the next packet is received.

When the execution of an instruction does not result in a decision to accept or reject the packet, the rule engine progresses to the next instruction. Instructions are executed until a final decision is made. If a packet is aborted in the middle at the network interface, the rule engine aborts execution and clears all base registers, program counter, octet counter and packet buffer. It is then ready to process next packet.

From a hardware perspective, the rule engine contains a buffer to store packet octets, several internal registers for temporary storage, instruction decoding logic and control circuitry. A description of each of the hardware resources in the rule engine follows.

24.11.4.9.2.4.1 Internal Registers

Several different types of registers are provided in the rule engine. The description of each of theese is provided below.

- Base Registers (B0-B4): There are four hardware base registers (B1-B4) that store the location of various protocol headers. The B0 register is not a hardware register and it always means a reference to the first octet in a packet. Each base register is eight bits wide and is used to reference octets in a packet. The base registers can point to any octet in the packet upto 255 octets. These registers are readable and writable by the rule engine. In addition, base registers B1 and B2 are loaded with data provided by the extractor when the rule engine is not running in extractor bypass mode (SPF_CONTROL[2] SPF_EXT_BYPASS). The rule engine can write to base registers irrespective of whether extractor bypass is enabled. When an operand needs to use contents of B1-B4 registers, the instruction only executes if the specified base register was loaded at least once after the beginning of current packet. If the specified base register was not loaded during the current packet, then the operand referenced by this base register cannot be extracted from the packet and will cause the rule engine to stall.
- Constant Registers (C0-C7): The rule engine can refer to any of eight 32-bit constant values. These are programmed by the host (SPF_CONSTj, j = 0 to 7) and the rule engine only has read access to these registers. Each of these registers can be changed at any time by the host software. However, changing the value of these registers is not recommended as unpredictable behavior may occur if contents are changed while the rule engine is executing instructions that use the C0-C7 values.
- General Purpose 32-bit registers (R0-R7): There are eight 32-bit registers that can be read/written by the rule engine. These are general purpose registers and can be used as temporary storage. In addition, when the rule engine is required to provide logging information, R4-R7 are used to store information that will be written to memory.
- Rate limit registers (L0-L3): Four 8-bit rate limit registers are used by rule engine to count specific types of packets that are to be rate limited. These registers are counters that are loaded with programmed threshold values (SPF_RATELIMi, i = 0 to 3) at the end of time interval determined by the clock pre-scaler (SPF_PRESCALE)
- General Purpose 1-bit registers (T0-T3): Four 1-bit registers are provided to store comparison results by the rule engine. These registers can be used to store 1-bit output from ALUs. In addition, the logical OR and logical AND of the flags can also be stored in T0-T3 registers. The 32-bit ALU results may also be stored in T0-T3 registers but only the LSB will be stored.

The rule engine instructions refer to these internal registers during execution. More details about instruction encoding are in subsequent sections.

24.11.4.9.2.4.2 Packet Buffer

When the rule engine is processing a packet, it stored a snapshot of latest 32 octets in an internal buffer. This buffer allows evaluation of expressions on packet octets even after the packet octets have gone by on the external receive VBUSP network interface.

At the start of packet, the first eight octets received are loaded into the first eight locations of the packet buffer. When the next eight octets arrive, these are loaded into the first eight locations and the existing octets are moved to the following eight locations. This process is continued until the packet buffer is full. When packet buffer is full and additional octets arrive, the oldest eight octets are shifted out and are no longer available to the rule engine. Thus, the packet buffer provides a snapshot of most recent octets received.

When the rule engine encounters instructions that reference packet octets which have not yet been received then the rule engine stalls until the required octets are available. Similarly, instructions which require octets that have already been shifted out of the packet buffer will cause rule engine to stall. And since these octets will never be available, the rule engine will not execute any further instructions for the current packet. It will return to the first instruction when the packet ends.

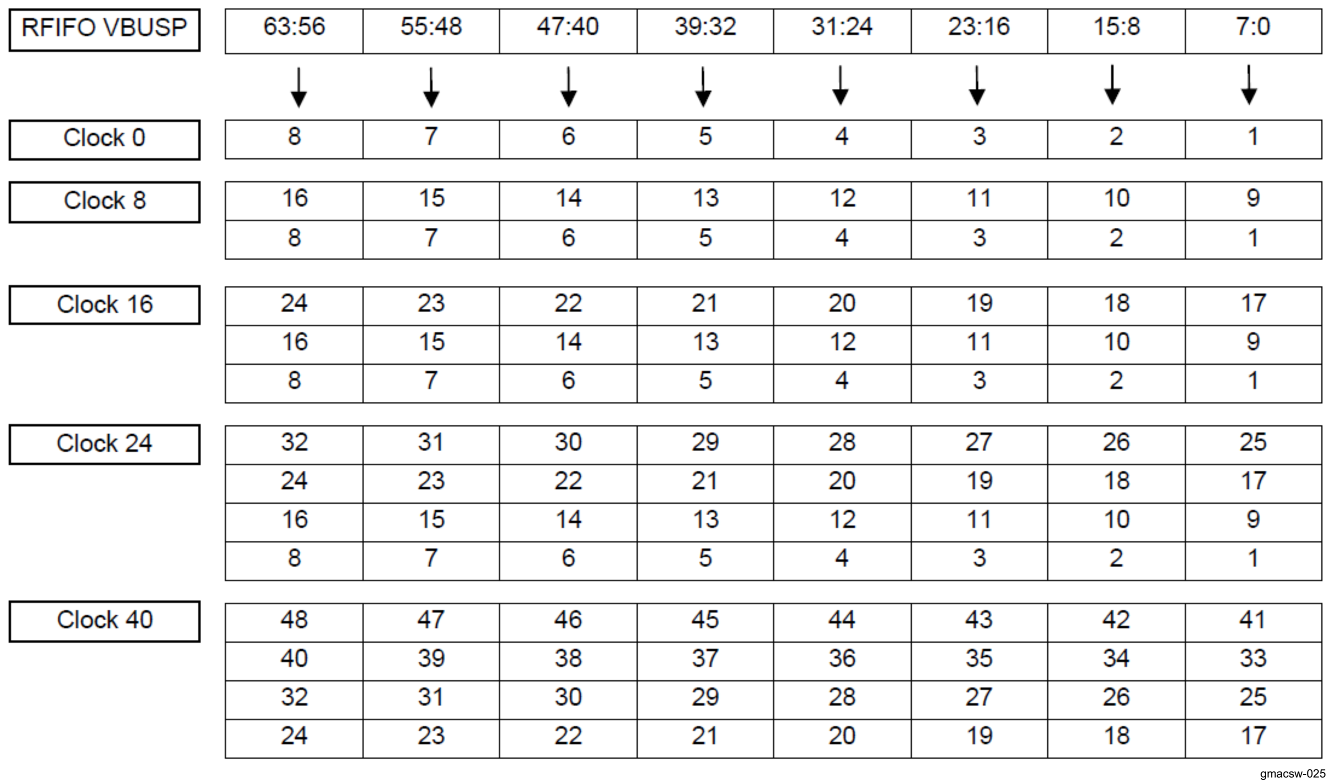


Figure 24-199. Packet Octets as Stored in the Packet Buffer

The contents of a packet, when used as one or more operands in an instruction, are always fetched from the packet buffer. The Base registers (B0-B4) in conjunction with an offset and bits are used to specify which octets are to be used as operands.

24.11.4.9.2.5 Intrusion Event Logger

The activity of the static packet filter is logged by the intrusion event logger. Based on the configuration, the event logger writes specific fields of the incoming packet that violate a particular rule to the system memory for software diagnostics.

The logging module allows controlling the frequency at which events are logged to the memory. A set of nine counters are provided and, of these, eight can be mapped to any address of the rule engine's instruction RAM using map registers. This mapping logically associates the log information with the instruction that caused the packet to be dropped. The 9th counter is not associated with any particular instruction and it can be used to count the packets that were dropped at instructions not mapped to any of the other eight counters.

For each of the nine counters, there is a register to store the minimum number of attacks that must be detected before any information is logged to memory. Only when this threshold is met and logging is enabled, the host interface module writes log data in system memory. The memory area used to log is specified by the host in the SPF_LOG_BEGIN address and SPF_LOG_END address registers.

The event logger continues to write data to memory until it runs out of space and the log overwrite is disabled. With each update of the log, the SPF_LOG_HWPTR is updated. This information can be used by the host to determine how much data has been logged. The software in turn keeps the SPF_LOG_SWPTR updated to inform the SPF about the next address from where information will be read by the host software. It is required that software program correct value of SPF_LOG_END address to enable hardware to determine roll-over location. The SPF considers all space before the SPF_LOG_SWPTR as available for logging. In case SPF_CONTROL[9] SPF_LOGOW_EN control bit is set, SPF ignores SPF_LOG_SWPTR and logs data irrespective of software read log status. Note that the memory space allocated for logging is a multiple of four 32-bit words and end address SPF_LOG_END should be loaded with byte address of the next byte following the last log entry. For example if 16 bytes space is allocated for SPF logging from address 0x1400_0000, then SPF_LOG_BEGIN should be 0x1400_0000 and SPF_LOG_END should be 0x1400_0010.

The host software must set SPF_CONTROL[8] SPF_LOG_EN bit to activate logging. The setting of SPF_LOG_EN bit has no effect on the threshold based counters and they are always active. In addition to the log counters, SPF has one master drop count register (SPF_DROPCNT) which tracks the total number of packets dropped thus far. This counter does not roll over and must be cleared for re-run by the host processor once it reaches the maximum value.

The format in which packet information is written in the memory by the event logger is shown in Table 24-890 and Table 24-891.

Table 24-890. Format for TCP/UDP Packets

Drop Code	-	-	Protocol
Source IP Address			
Destination IP Address			
Source TCP/UDP Port		Destination TCP/UDP Port	

Table 24-891. Format for ICMP Packets

Drop Code	-	-	Protocol
Source IP Address			
Destination IP Address			
Type	Code	Checksum	

Each entry logged to memory has a drop code associated with it. The drop code is the address in instruction memory that actually triggered the drop. Up to eight drop codes can be monitored in this manner. In addition to drop code, the protocol, IP addresses and source/destination ports associated with the dropped packet are recorded.

Log data can be supplied by either Extractor module or by Rule engine. The SPF_CONTROL[3] SPF_RULE_LOG bit must be set to use log data from Rule Engine and cleared to use data supplied by the Extractor.

When logging is done through the Rule Engine, contents of internal registers R4-R7 are written to memory. The format of packet information stored can be programmed in the rule engine except for the drop code field which is static and cannot be changed. The format in which rule engine information is written in the memory by the event logger is shown in Table 24-892.

Table 24-892. Rule Engine Format

Drop Code	Register 4[23:0]
Register 5[31:0]	
Register 6[31:0]	
Register 7[31:0]	

The rule engine programming must ensure that these registers contain all required information that is to be recorded before the packet drop instruction is executed. As soon as the drop instruction is executed, the logging module starts to send data from the registers R4, R5, R6 and R7 to the system memory and is not possible to modify the contents of these registers.

24.11.4.9.2.6 Rate Limiter

The static packet filter provides a way to limit the rate of specific types of incoming packets. The SPF module provides four rate limit registers (L0-L3) that are used in conjunction with a clock prescaler (SPF_PRESCALE) and rule engine instructions. The clock prescaler is a clock divider and every count down to zero and the subsequent roll-over indicates end of a time interval. At every such event, each of the four rate limit registers is loaded with a preset value (SPF_RATELIMi, i= 0 to 3) that is programmable by the host processor. When packets are received, the rule engine can identify packets of particular types and execute a limit operation. The limit operation specifies a condition (described in Table 24-910) and a limit register. If the condition evaluates to true, the packet is dropped if the rate limit register is zero. If it is non-zero, then the rate limit register is decremented by one.

Thus, by using the prescaler and the limit registers, it is possible to control the rate of specific type of packets. By controlling the value of prescale counter (common to all rate limit registers) and the rate limit thresholds, the granularity of the rate can be modified. A lower value of prescale counter will cause frequent reloads of the rate limit registers and will allow rate control over small time intervals. Keeping the prescale counter at extremely small values may cause the rate limiter to be ineffective because the limit registers will be reloaded too frequently to count down to zero. Conversely, a higher value of prescale counter will cause less frequent reloads of the rate limit registers and the rate control will be over longer time intervals. The reload value of each 8-bit rate limit register and the prescale register are programmed by the host processor before SPF is enabled. A reload value of 0xFF can be used to disable any of the rate limiters at run-time without changing the firmware. Both the prescale and limit registers can be modified during run-time.

24.11.4.9.2.7 Rule Engine Instruction Set Architecture

24.11.4.9.2.7.1 Instruction Format

The design supports 64 deep instruction memory. Each of the instructions is 78 bit wide and is stored in a RAM internal to the SPF module. Each instruction can have up to four operands, two arithmetic/logical operations and two conditional jump/save actions based on the outcome of the operations.

The operand field is 52 bits wide and it can have up to four operands. The arithmetic and logical operation codes are four bits each. The operation codes are nine bits each.

Table 24-893. Instruction Format

Bits	Field Name	Description
77:26	OPERAND	Up to four operands are specified in this field. The source of the operands can be the packet that is being received at the network interface, one of the internal register values or an 8/16/32 bit operand specified within the instruction.
25:22	FUNCTION0	This field specifies a single or dual operand Arithmetic/Logic Function. This function is applied to one or both of first two operands specified in the instruction.
21:18	FUNCTION1	This field also specifies a single or dual operand Arithmetic/Logic Function. This function is applied to one or both of the third and fourth operands specified in the instruction.

Table 24-893. Instruction Format (continued)

Bits	Field Name	Description
17:9	OPERATION0	A Save/Jump/Limit/Nop operation is specified in this function. Save operation code includes the source and destination information. The save source is the output of functions and the destination is one of the internal registers where data should be saved. Jump operation code has information about a condition and a destination. The jump occurs when the condition, which is based on the result of function0 and function1, is true. The Jump destination controls the flow of instruction execution. Jump to location 1 results in the packet being dropped. Rule engine goes back to initial instruction and waits for next packet. The event logger writes information to memory. Jump to location zero results in the packet being accepted. The rule engine goes back to initial instruction and waits for next packet. Limit operation code has information about a condition and a rate limit register. The Limit operation either causes the packet to be dropped or results in the specified rate limit register to be decremented by one.
8:0	OPERATION1	This field specifies a second save/jump operation. The format for this field is same as for Operation0.

24.11.4.9.2.7.2 Operand Field

The operand field specifies up to four operands. Each operand is obtained from the incoming packet data, from an internal register or as an immediate value specified in the instruction itself. In addition, there is a bit mask associated with each operand. The mask values are encoded in the instruction itself and the mask for each operand is applied before it is used by the Arithmetic and Logic Unit.

The operands are input to the two 32-bit Arithmetic and Logic units. For all calculations, a mask is used with each operand. The mask is a 32-bit number that is generated from a 5-bit code provided in the instruction or from an immediate value in the instruction. The 5-bit mask code specifies how many bits (from the LSB side) will be input to the ALU. An immediate mask must be used when the mask is not a continuous sequence of 1's. Whenever an immediate mask value is specified, the mask value specified by the 'Bits' fields is ignored.

Each operand is at least 13-bits long and the encoding for each operand types is shown in Table 24-894 to Table 24-901.

Table 24-894. Packet Data Operand

Description	This type of operand is derived from the packet itself. The encoding specifies a number of bits (up to 32) to be extracted from a location in the packet indicated by the selected base register and offset. The format of 13-bit operand code is shown below. Internally, a 32-bit number is obtained from the packet. Then, a mask is created from the bits[4:0] field and bitwise ANDed with the 32- bits extracted from the packet.

12	11	10	9	8	7	6	5	4	3	2	1	0
Base[2:0]			Offset[4:0]					Bits[4:0]				

Bits	Field Name	Description
12:10	Base[2:0]	The Base field selects one of the Base registers B0, B1, B2, B3 or B4. The operand comprises of (Bits+1) bits extracted from the packet. The offset from where the operand octets are extracted is determined by the sum of the value of specified base register (Base 0 to Base 4) and the specified offset. Base 0 is start of packet.
9:5	Offset[4:0]	The value of the selected base register plus the specified offset is the octet location in the packet from where the specified number of bits will be picked.
4:0	Bits[4:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, only one bit is used as operand. When Bits[4:0] is 31, then 32 bits are used as operand.

Table 24-895. Constant Operand

Description	One of the eight constants programmed by the host software can be used as operand. The bits field specifies the mask to be applied to the selected 32-bit constant.

12	11	10	9	8	7	6	5	4	3	2	1	0

1	0	1	0	0	C[2:0]	Bits[4:0]

Bits	Field Name	Description
7:5	C[2:0]	This field selects one of the constants (C0-C7).
4:0	Bits[4:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, the least significant bit is used as operand. When Bits[4:0] is 2, then two LSBs are used as operand and so on.

Table 24-896. 32-bit Register Operand

Description	One of eight 32-bit registers is used as operand and bits[4:0] field specifies the bitmask to be applied to that register.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	1	R[2:0]			Bits[4:0]				

Bits	Field Name	Description
7:5	R[2:0]	This field selects one of the 32-bits registers (R0-R7).
4:0	Bits[4:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, the least significant bit is used as operand. When Bits[4:0] is 2, then two LSBs are used as operand and so on.

Table 24-897. 1-bit Register Operand

Description	One of the four 1-bit registers is used as operand.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	0	0	T[1:0]		0	0	0	0	0

Bits	Field Name	Description
6:5	T[1:0]	This field selects one of the four 1-bits registers (T0-T3).

Table 24-898. Base Register Operand

Description	One of the base registers is used. The 2-bit codes are 00, 01, 10 and 11 for B1 to B4 respectively.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	1	B[1:0]		0	0	Bits[2:0]		

Bits	Field Name	Description
6:5	B[1:0]	This field selects one of the four base registers (B1-B4). Note that B0 is a virtual register which is always zero and specifies the start of packet.
2:0	Bits[2:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, the least significant bit is used as operand. When Bits[4:0] is 2, then two LSBs are used as operand and so on.

Table 24-899. End of Packet Operand

Description	The End-of-Packet (EOP) operand can be used to detect the end of current packet. Use of this operand will stall the CPU until the end of packet is reached. The end of packet operand will always be equal to one when the CPU detects end of packet. It will be zero until packet data is being received and will also stall the CPU. The recommended use of this operand is in the last instruction where a check may be made for size of current packet. Note that this operand only tracks the end of a complete successfully received packet by the RFIFO. The EOP operand does not detect abort of a packet.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	0	0	1	0	Bits[4:0]				

Bits	Field Name	Description
4:0	Bits[4:0]	The number of bits specifies the mask. However, these bits are inconsequential for EOP operand. Irrespective of the mask specified by bits[4:0], the EOP operand is always equal to 0x1 in the execution cycle of the instruction it is used.

Table 24-900. Octet Count Operand

Description	The Octet count operand is used to determine the number of bytes that have been received by rule engine.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	1	1	Bits[4:0]				

Bits	Field Name	Description
4:0	Bits[4:0]	The number of bits specifies the mask that is applied to octet counter value. The recommend value for this field is 0x1F.

Table 24-901. Immediate Data Operands

Description	The immediate data operands can be bigger than 13 bits as shown in the codes below. Each of such operands is directly sent to the ALU without applying any addition bit mask. The immediate data operands can be 8, 16 or 32 bits wide.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	Data[7:0]							

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	1	Data[15:0]															

36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	Data[31:0]																															

Table 24-902. Immediate Operand Masks

Description	The operand fields can also be used to specify bit masks. These are used when only specific bits from an operand are to be extracted. There is provision for 8-bit and 16-bit masks that can be encoded into an instruction's operand field.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	Mask[7:0]							

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	1	Mask[15:0]															

When the operand field is all-ones it does not specify any operand and is considered zero.

The tables below list the combinations in which operands may be programmed in an instruction. In this list, Pkt/Reg operand is an operand consisting of octets from the packet or the contents of one of the internal registers; mask specifies the value that is used as a mask instead of the mask generated from the Bits field in the instruction; immediate operand (immediate[7:0], immediate[15:0] and immediate[31:0]) is an 8/16/32 bit number that is used as an operand. The format of Pkt/Reg operand is as shown previously. The decoding of operands for each ALU is dependent upon the type of operation specified in the ALU and the format of the operand field. When there are four operands in the operand field, the first and second operands are used in ALU0 and the rest by ALU1. When ALU0 needs only one operand, then the remaining operands are fed to ALU1. The order of decoding operands is designed to first provide operands to ALU0 from the more significant side of operand field and then to ALU1. If the number of operands specified in instruction does not match with the number of operands required by the ALUs, unspecified behavior can occur. For an instruction to execute, both ALUs must have valid data. If any of the ALUs uses a packet data operand which is not yet available then the rule engine stalls until data is available. During the stall phase, the instruction is decoded every cycle and all required operands must be available simultaneously for the instruction to be executed In addition, if an instruction refers to packet octets that are no longer available, the rule engine stalls until the end of packet.

Table 24-903. Operand Field With Four Operands

Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]

Table 24-904. Operand Field With Three Operands

Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt / Reg / Imm [7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]
11101 Immediate[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm[7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	11101 Immediate[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]

Table 24-905. Operand Field With Two Operands

Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Reserved	Reserved
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt / Reg / Imm[7:0] Operand[12:0]	Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]
11101 Immediate[15:0] 11111		Pkt/Reg Operand[12:0]	11000 Mask[7:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	11101 Immediate[15:0] 11111		Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	11101 Immediate[15:0] 11111	
11101 Immediate[15:0] 11111		11101 Immediate[15:0] 11111	
11101 Immediate[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]	Reserved
Pkt/Reg Operand[12:0]	11001 Mask[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11001 Mask[15:0] 11111	
Pkt/Reg Operand[12:0]	1101 Mask[15:0] 01 Immediate[15:0]		

Pkt / Reg / Imm[7:0] Operand[12:0]	11110 Immediate[31:0] 11

11110 Immediate[31:0] 11	Pkt / Reg / Imm[7:0] Operand[12:0]

Table 24-906. Operand Field With One Operand

Pkt / Reg / Imm[7:0] Operand[12:0]	Reserved	Reserved	Reserved

Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Reserved	Reserved

Pkt/Reg Operand[12:0]	11001 Mask[15:0] 11111	Reserved

11101 Immediate[15:0] 11111	Reserved	Reserved

11110 Immediate[31:0] 11	Reserved

The formats in which operands are specified in the operand field have certain restrictions:

- a mask cannot follow an immediate value
- a mask cannot follow a previous mask
- a reserved field should have all following fields reserved as well
- the first field cannot be a mask

24.11.4.9.2.7.3 Arithmetic/Logical Function Field

There can be two codes for arithmetic and logical functions in each instruction. Each of the operations can involve one or two operands.

The location of each function field in an instruction is listed in Table 24-907.

Table 24-907. Arithmetic/Logical Instruction Field

Bits	Description
23:20	Single or Dual operand Arithmetic/Logic Function (function 0)
19:16	Single or Dual operand Arithmetic/Logic Function (function 1)

The bit codes for each arithmetic/logic function that can be performed on the operands of an instruction are listed in Table 24-908.

Table 24-908. Arithmetic/Logical Operation Codes

Code	Description	Operands Required	32-bit result	1-bit flag result
0x0	Less Than	2	0	Set if true
0x1	Less Than Equal To	2	0	Set if true
0x2	Greater Than	2	0	Set if true
0x3	Greater Than Equal To	2	0	Set if true
0x4	Equal to Zero	1	0	Set if true
0x5	Not Equal to Zero	1	0	Set if true
0x6	Equal to One	1	0	Set if true
0x7	Not Equal to One	1	0	Set if true
0x8	Equal	2	0	Set if true
0x9	Not Equal	2	0	Set if true
0xA	Add	2	Sum	Set if overflow
0xB	Subtract	2	Difference	Set if output is negative
0xC	Bitwise AND	2	Bitwise AND	0

Table 24-908. Arithmetic/Logical Operation Codes (continued)

Code	Description	Operands Required	32-bit result	1-bit flag result
0xD	Bitwise OR	2	Bitwise OR	0
0xE	Result is same as Operand	1	Same as operand	0
0xF	No Function (NOF)	0	0	0

The ALU0 and ALU1 are assigned function0 and function1 respectively. In case only one function is to be performed in an instruction, function0 field should be used to do that. An instruction with no function for ALU0 and a valid function for ALU1 will result in unspecified behavior.

The output of each function is in the form of a 32-bit number and a single bit value. The single bit value is either the carry bit from an add/subtract operation or the Boolean result from a comparison function. The single bit output from each function is referred as the flag0 or flag1 result. The 32-bit number is the outcome of arithmetic or a bitwise logical operation and it is referred to as the word0 or word1 result.

24.11.4.9.2.7.4 Operation Field

The operation field specifies a limit, save or a jump operation. All of these operations involve two arguments that are specified within the instruction.

In case of limit operation, one of the arguments is a condition and the other argument is a limit register. In case of save operation, one of the arguments is source data and the other argument is a destination where the data will be saved. In jump operation, the first argument is a condition that must be true for the jump to occur and the second argument is a destination to which the program control will be moved to. The destination argument in a jump operation also specifies specific destinations that lead to the current packet being immediately rejected or accepted.

The location of operation field in an instruction is shown in Table 24-909.

Table 24-909. Operation Fields

Bits	Description
17:9	Limit/Save/Jump Operation (operation 0) For limit operation, [17:16] is the operation field, [15:13] is the condition field and [12:9] is a limit register. For save operation, [17:16] is the operation field, [15:13] is the source field, and [12:9] is the destination field. For jump operation, [17:16] is the operation field, [15:13] is the condition field, and [12:9] is the destination field
8:0	Limit/Save/Jump Operation (operation 1) For limit operation, [8:7] is the operation field, [6:4] is the condition field and [3:0] is a limit register. For save operation, [8:7] is the operation field, [6:4] is the source field, and [3:0] is the destination field. For jump operation, [8:7] is the operation field, [6:4] is the condition field, and [3:0] is the destination field.

Table 24-910. Limit Operation

Description	In a limit operation, the condition argument specifies a condition that must be true for the specified limit register to be decremented if non-zero or the packet to be dropped if the limit register is zero.

<table>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td colspan="3">Condition Code</td><td colspan="4">Limit Register</td></tr>
</table>

Bits	Field Name	Description
6:4	Condition Code	It is a 3-bit code that specifies the condition that must be true for the operation to be executed. 0x0: Single bit flag (flag0 from function 0) 0x1: Single bit flag (flag1 from function 1) 0x2: Logical OR of flags (flag0 | flag1) 0x3: Logical NOR of flags (!(flag0 | flag1)) 0x4: Logical AND of flags (flag0 & flag1) 0x5: Logical NAND of flags (!(flag0 & flag1)) 0x6: XOR of flags (flag0 ^ flag1) 0x7: Unconditional Decrement/Limit

Bits	Field Name	Description
3:0	Limit Register	It is a 4-bit code that specifies the limit register that is to be decremented or checked for zero value. 0x0: Rate limit register L0 0x1: Rate limit register L1 0x2: Rate limit register L2 0x3: Rate limit register L3 0x4-0xF: Reserved

Table 24-911. Save Operation

Description	The arguments for a save operation include a code for source data and a code for the destination. The source is either a single bit data or a 32-bit word. The single bit data is either a flag from the functions or a logical OR/AND of the flags. The 32-bit word is the result of the ALU arithmetic/logical functions.

8	7	6	5	4	3	2	1	0
0	1	Source Data Code			Destination Register Code			

Bits	Field Name	Description
6:4	Source Data Code	It is a 3-bit code that specifies the source data which is to be save in another register 0x0: Single bit flag (flag0 from function 0) 0x1: Single bit flag (flag1 from function 1) 0x2: Logical OR of flags (flag0 | flag1) 0x3: Logical AND of flags (flag0 & flag1) 0x4: Output of function (word0) 0x5: Output of function (word1) 0x6-0x7: Reserved
3:0	Destination Register Code	It is a 4-bit code that specified the register in which source data is to be stored 0x0-0x3: Single Bit register (T0-T3) 0x4-0x7: Base Register (B1-B3) 0x8-0xF: 32-bit Register (R0-R7)

For some instructions, the rule engine extracts the operand from the current packet. To determine the packet bits to be used as operands, it refers to one or more base registers. Whenever an instruction causes a change in value of any of the base registers, it cause the rule engine to wait for one clock cycle before executing the next instruction. The rule engine is a pipelined processor and change in the value of a base register causes the prefetched operand values to become stale and these need to be fetched again. The clock cycle inserted allows rule-engine to fetch the data again before the instruction using this data is executed. Only the instructions that use packet data will be delayed by a clock cycle if the previous instruction caused a change in the base register that is referenced in the current instruction.

Table 24-912. Jump Operation

Description	For jump operation, one of the operands is a condition that must evaluate to true for the jump operation to occur. The other operand is the destination to which the program counter will move to if the jump operation is executed. The condition in a jump operation is either one of the flags from the output of the arithmetic/logic unit or a logical AND/OR of the flags.

8	7	6	5	4	3	2	1	0
1	0	Condition Code			Destination Code			

Bits	Field Name	Description
6:4	Condition Code	Condition Code 0x0: Single bit flag (flag0 from function 0) 0x1: Single bit flag (flag1 from function 1) 0x2: Logical OR of flags (flag0 | flag1) 0x3: Logical NOR of flags (!(flag0 | flag1)) 0x4: Logical AND of flags (flag0 & flag1) 0x5: Logical NAND of flags (!(flag0 & flag1)) 0x6: XOR of flags (flag0 ^ flag1) 0x7: Unconditional Jump

Bits	Field Name	Description
3:0	Destination Code	Destination Code 0x0: Accept packet, return to instruction zero and wait for the next packet. 0x1: Reject packet and return to instruction zero. 0x2-0xF: Go to instruction at offset +2 ...+15

The jump operation results in a change of program execution flow. The jump destination determines the next instruction that will be executed by the rule engine. The destination is only specified as a positive offset to the current value of the program counter. The program counter can result in the rule engine skipping a given number of instructions instead of executing the next instruction. In addition, the destination field can instruct the rule engine to immediately reject or accept the current packet without any further checks. The rule engine then goes back to the first instruction and waits for a new packet to arrive before it executes any further instructions.

The rule engine is a pipelined processor. When it executes a jump operation, there is a delay of one clock cycle before the next instruction is executed.

The operations specified in the two operation fields are independent of each other. However, it is possible that conflicting operations are specified in the operation fields of an instruction. When conflicting operations are specified and conditions of both operations evaluate to true; then the execution flow is determined as described below:

1. If one of the ALUs has a limit/jump operation that will cause the packet to be dropped, then the packet will be dropped irrespective of what the other ALU indicates.
2. If case 1 above is false and one of the ALUs has a jump operation that will cause the packet to be accepted, then the packet will be accepted and the rule engine will return to idle state.
3. If case 1 and 2 above are false and one of the ALUs has a jump to offset with the respective condition true, the jump will be executed. If there is conflicting jump to offset information in the two ALUs, ALU0 is given priority.

Table 24-913. No Operation

Description	If there is no Save, Jump or Limit operation specified in the operation field, it is interpreted as a no operation. There is no change in execution flow, no change in any register values or rate limit registers when a no operation is encountered.

8	7	6	5	4	3	2	1	0
1	1	Reserved			Reserved			

It is expected that the rule engine will be programmed such that a decision to accept or discard the current packet is made before all instructions have executed. If the rule engine reaches the last instruction and it does not result in a decision to accept/reject the packet, then the execution flow does not stop and all instructions from the first instruction are executed again until the end of packet is reached.

24.11.4.9.3 Programming Guide

24.11.4.9.3.1 Initialization Routine

The packet filter must be initialized in order for it to operate as specified. A typical order in which the module can be initialized is shown below:

1. Initialize the firmware by programming the internal memory and verify it (SPF_INSTR_W2, SPF_INSTR_W1, SPF_INSTR_W0, SPF_INSTR_CTL)
2. Allocate memory for logs written by SPF and initialized the log space parameters (SPF_LOG_BEGIN, SPF_LOG_END, SPF_SW_PTR)
3. Program the log map registers to associate drop codes with the log thresholds. (SPF_LOG_MAP0, SPF_LOG_MAP1)
4. Program log thresholds (SPF_LOG_THRESHk, where k = 0 to 8)
5. Program constant registers if required by the firmware (SPF_CONSTj, where j = 0 to 7)
6. Program clock prescale counters and rate limit registers if required by the firmware (SPF_PRESCALE, SPF_RATELIMi , where i = 0 to 3)

7. Program the interrupt frequency control register and interrupt mask register to setup interrupts (SPF_INTCNT, SPF_MASK_SET)
8. Program the control registers to enable SPF filter and logger (SPF_CONTROL).

24.11.4.9.3.2 Interrupt Service Routine

The interrupt service routine in SPF should be designed to process the data logs generated by SPF. It should trigger a higher layer application software that can analyze the data logs and determine if any remedial measures are required based on the information in the logs. At the minimum, the interrupt service routine must reprogram the software pointer for the logging machine to be able to continue logging.

The typical tasks performed following an interrupt are listed in the pseudo code below.

```
SPF_ISR {
read log hardware pointer
read log software pointer
determine how many log entries are to be read
read each log entry starting with the software pointer
update software pointer to reflect the next unread entry
clear interrupt
}
```

24.11.4.9.3.3 Rule Engine Example Program

Here is an example of how the rule engine can be programmed to detect packets that resemble Denial of Service traffic. The pseudo code is shown below.

```
IDLE:
jump to ICMP if start_of_packet
ICMP:
if (protocol==ICMP)
limit_ICMP
if (fragmented packet)
drop and jump to IDLE
accept and jump to IDLE
else
jump to IP
IP:
if (source_ip==dest_ip)
drop and jump to IDLE
if (fragmented and (fragment_offset+ip_size)>2^16)
drop and jump to IDLE
accept the packet and jump to IDLE
```

24.11.4.10 Common Platform Time Sync (CPTS)

The Common Platform Time Sync (CPTS) module is used to facilitate host control of time sync operations. It enables compliance with the IEEE 1588-2008 standard for a precision clock synchronization protocol.

24.11.4.10.1 CPTS Architecture

Figure 24-200 shows the architecture of the CPTS module inside the GMAC_SW Ethernet Subsystem. Time stamp values for every packet transmitted or received on either port of the GMAC_SW are recorded. At the same time, each packet is decoded to determine if it is a valid time sync event. If so, an event is loaded into the Event FIFO for processing containing the recorded time stamp value when the packet was transmitted or received.

In addition, both hardware (HWx_TS_PUSH) and software (TS_PUSH) can be used to read the current time stamp value though the Event FIFO

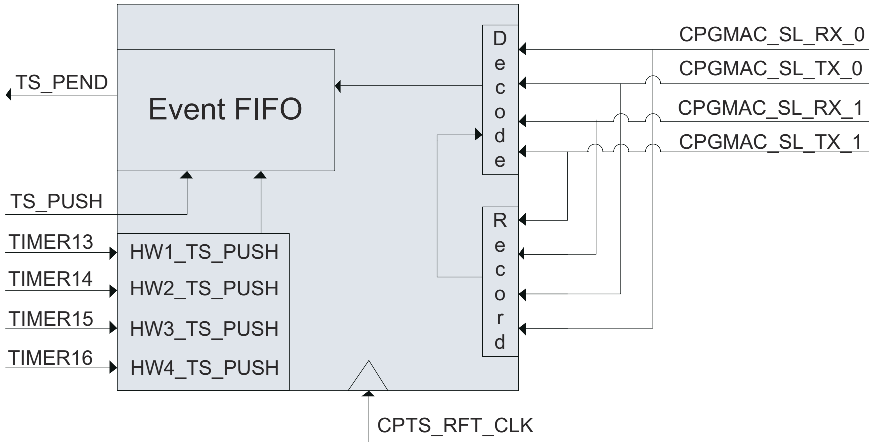


Figure 24-200. CPTS Block Diagram

The reference clock used for the time stamp (CPTS_RFT_CLK) can be derived from several sources. See *PRCM* for more details.

24.11.4.10.2 CPTS Initialization

The CPTS module should be configured as:

1. Reset the CPTS module.
2. Clear the CPTS_EN bit in the CPTS_CONTROL register.
3. Write the CLKSEL_RFT value in the CM_GMAC_GMAC_CLKCTRL register in the PRCM with the desired reference clock selection.
4. Set the CPTS_EN bit in the CPTS_CONTROL register.
5. If using interrupts and not polling, enable the interrupt by setting the TS_PEND_EN bit in the CPTS_INT_ENABLE register.

24.11.4.10.3 Time Stamp Value

The time stamp value is a 32-bit value that increments on each CPTS_RFT_CLK rising edge when CPTS_EN is set to 1. When CPTS_EN is cleared to 0, the time stamp value is reset to 0.

If more than 32-bits of time stamp are required by the application, the host software must maintain the necessary number of upper bits. The upper time stamp value should be incremented by the host when the rollover event is detected.

For test purposes, the time stamp can be written via the time stamp load function (CPTS_TS_LOAD_VAL and CPTS_TS_LOAD_EN registers).

24.11.4.10.4 Event FIFO

All time sync events are push onto the Event FIFO. There are 16 locations in the event FIFO with no overrun indication supported. Software must service the event FIFO in a timely manner to prevent FIFO overrun.

24.11.4.10.5 Time Sync Events

Time Sync events are 64-bit values that are pushed onto the event FIFO and read in two 32-bit reads. Two 32-bit registers, CPTS_EVENT_HIGH and CPTS_EVENT_LOW hold the data of a time sync event. There are five types of sync events:

- Time stamp push event
- Time stamp counter rollover event
- Time stamp counter half-rollover event
- Ethernet receive event
- Ethernet transmit event

24.11.4.10.5.1 Time Stamp Push Event

Software can obtain the current time stamp value (at the time of the write) by initiating a time stamp push event. The push event is initiated by setting the TS_PUSH bit of the CPTS_TS_PUSH register. The time stamp value is returned in the event, along with a time stamp push event code. Software should not push a second time stamp event on to the FIFO until the first time stamp value has been read from the event FIFO.

24.11.4.10.5.2 Time Stamp Counter Rollover Event

The CPTS module contains a 32-bit time stamp value. The counter upper bits are maintained by host software. The rollover event indicates to software that the time stamp counter has rolled over from FFFF FFFFh to 0000 0000h, and the software maintained upper count value should be incremented.

24.11.4.10.5.3 Time Stamp Counter Half-rollover Event

The CPTS includes a time stamp counter half-rollover event. The half-rollover event indicates to software that the time stamp value has incremented from 7FFF FFFFh to 8000 0000h. The half-rollover event is included to enable software to correct a misaligned event condition. The half-rollover event is included to enable software to determine the correct time for each event that contains a valid time stamp value, such as an Ethernet event. If an Ethernet event occurs around a counter rollover (full rollover), the rollover event could possibly be loaded into the event FIFO before the Ethernet event, even though the Ethernet event time was actually taken before the rollover. Figure 24-201 shows a misalignment condition.

Host software must detect and correct for misaligned event conditions. For every event after a rollover and before a half-rollover, software must examine the time stamp most significant bit. If bit 31 of the time stamp value is low (0000 0000h through 7FFF FFFFh), then the event time stamp was taken after the rollover and no correction is required. If the value is high (8000 0000h through FFFF FFFFh), the time stamp value was taken before the rollover and a misalignment is detected. The misaligned case indicates to software that it must subtract one from the upper count value stored in software to calculate the correct time for the misaligned event. The misaligned event occurs only on the rollover boundary and not on the half-rollover boundary. Software only needs to check for misalignment from a rollover event to a half-rollover event.

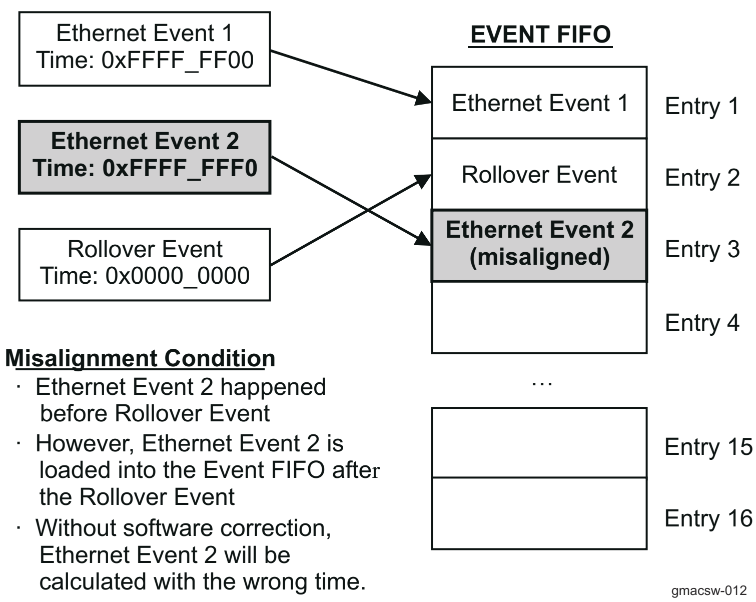


Figure 24-201. Event FIFO Misalignment Condition

24.11.4.10.5.4 Hardware Time Stamp Push Event

There are four hardware time stamp inputs (HW1/4_TS_PUSH) that can cause hardware time stamp push events to be loaded into the Event FIFO. Each hardware time stamp input is internally connected to the PORTIMERPWM output of each timer as shown in Figure 24-202.

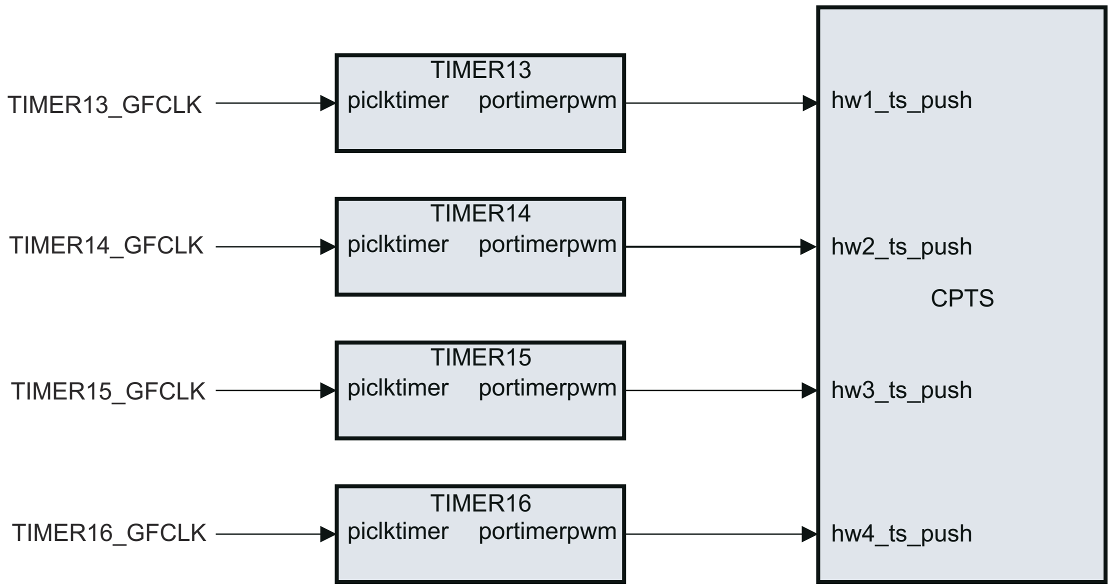


Figure 24-202. HW1/4_TSP_PUSH Connection

The event is loaded into the event FIFO on the rising edge of the timer, and the PORT_NUMBER field in the CPTS_EVENT_HIGH register indicates the hardware time stamp input that caused the event.

Each hardware time stamp input must be asserted for at least 10 periods of the selected RCLK clock. Each input can be enabled or disabled by setting the respective bits in the CPTS_CONTROL register.

Hardware time stamps are intended to be an extremely low frequency signals, such that the event FIFO does not overrun. Software must keep up with the event FIFO and ensure that there is no overrun, or events will be lost.

24.11.4.10.5.5 Ethernet Port Events

Packets transmitted or received on each Ethernet port can generate Ethernet Transmit Events or Ethernet Receive Events, respectively. The CPTS hardware will decode each packet to determine if it is a valid CPTS time sync event.

According to the IEEE 802.3 Ethernet standard, each Ethernet frame contains a 2-octet EtherType field to indicate which protocol is encapsulated in the PayLoad field, as shown in Figure 24-203. For standard time sync packets, this will contain the EtherType for the Precision Time Protocol (IEEE 1588), which is defined as 0x88F7. The CPTS hardware will compare this field to the TS_LTYPE1 field or the TS_LTYPE2 field (depending on which enable bit was set) in the CPSW_TS_LTYPE register, which should also be programmed to 88F7h.

When a virtual LAN is used, an additional 4-octet 802.1Q tag is inserted in the Ethernet frame before the EtherType field, as shown in Figure 24-203. To indicate to the CPTS hardware that a virtual LAN is in use, the Pn_TS_VLAN_LTYPE1_EN (or Pn_TS_VLAN_LTYPE2_EN) enable bit must be set in the Pn_CONTROL register. The EtherType for the 802.1Q tag is defined as 0x8100, and the CPTS hardware will compare this value to the VLAN_LTYPE1 (or VLAN_LTYPE2 depending on which enable bit was set) field in the CPSW_VLAN_LTYPE register, which should also be programmed to 0x8100.

When two stacked VLANs are used, two additional 4-octet 801.Q tags are inserted in the Ethernet frame before the EtherType field, as shown in Figure 24-203. In this case, both VLAN_LTYPE1 and VLAN_LTYPE2 must be enabled. The outer tag must match the value of the VLAN_LTYPE1 field, and the inner tag must match the value of the VLAN_LTYPE2 field.

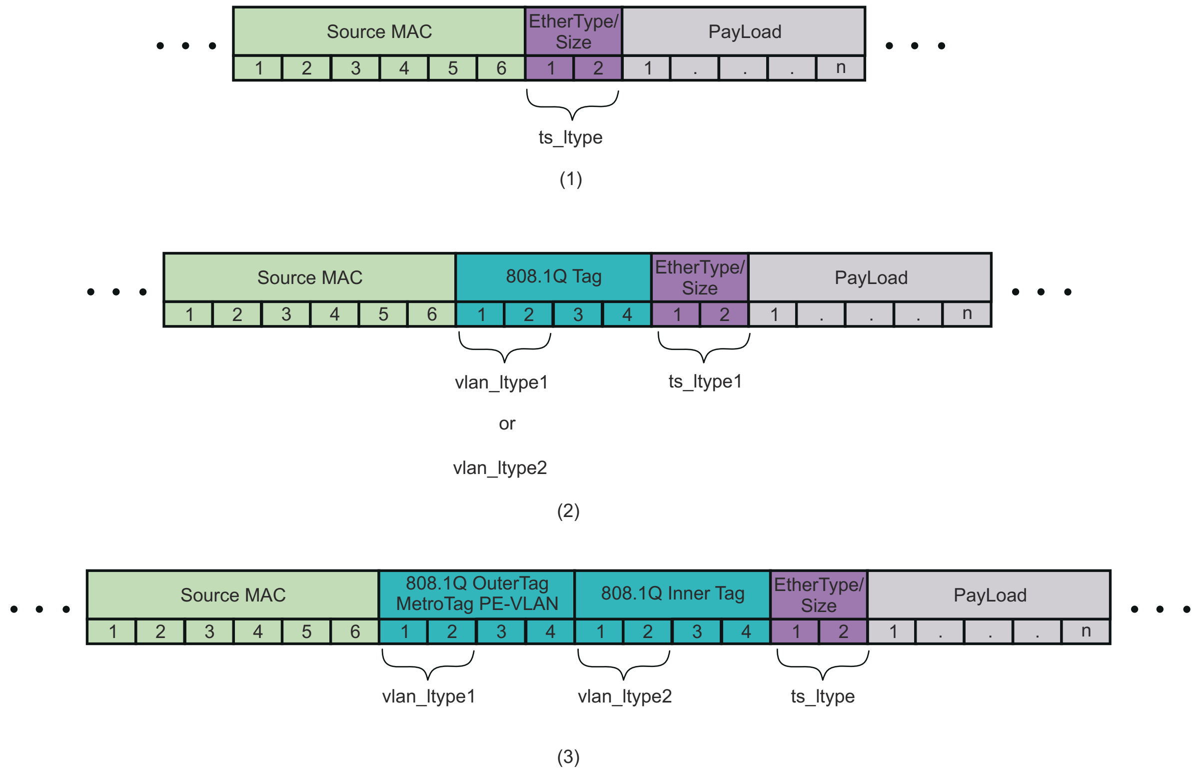


Figure 24-203. Partial Ethernet-II Frames Showing Register Mapping of EtherTypes for a Simple Frame (1), a Single 1Q Tag Added (2), and Two 1Q Tags Added (3)

To enable transmit/receive event packets on a given Ethernet port, perform the following steps, where n is the port number:

1. Set the [1] Pn_TS_TX_EN or the [0] Pn_TS_RX_EN bit in the Pn_CONTROL register to enable transmit/ receive event packets
2. Configure the register fields as follows:
   a. Set the Pn_CONTROL[2] Pn_TS_LTYPE1_EN or [3] Pn_TS_LTYPE2_EN bit to 1
   b. Set the CPSW_TS_LTYPE[15:0] TS_LTYPE1 or [31:16] TS_LTYPE2 (depending on which enable bit was set in 2(a)) field to 88F7h, which corresponds to the Precision Time Protocol (IEEE 1588) EtherType.
   c. Set the Pn_TS_SEQ_MTYPE[21:16] Pn_TS_SEQ_ID_OFFSET field to 1Eh, which is the sequence ID offset in the common message header given in the IEEE 1588 specification.
3. To enable support for VLAN tagging (IEEE 802.1Q):
   a. Set the [20] Pn_TS_VLAN_LTYPE1_EN or [21] Pn_TS_VLAN_LTYPE2_EN bit in the Pn_CONTROL register.
   b. Set the CPSW_VLAN_LTYPE[15:0] VLAN_LTYPE1 or [31:16] VLAN_LTYPE2 field (matching what was used in step 3a) to 8100h, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType.
4. To enable support for up to two stacked VLANs (IEEE 802.1ad):
   a. Set the Pn_TS_VLAN_LTYPE1_EN and Pn_TS_VLAN_LTYPE2_EN bits in the Pn_CONTROL register.
   b. Set the VLAN_LTYPE1 field in the CPSW_VLAN_LTYPE register to match the EtherType of the outer tag.
   c. Set the VLAN_LTYPE2 field in the CPSW_VLAN_LTYPE register to 8100h, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType of the inner tag.
5. Set the Pn_TS_SEQ_MTYPE[15:0] Pn_TS_MSG_TYPE_EN field to choose which types of time stamp messages will push events onto the Event FIFO. Table 9-20 lists the message types defined in the IEEE 1588-2008 specification. Table 24-914 lists the message types defined in the IEEE 1588-2008 specification.

Table 24-914. Values of Message Type Field

Message Type	Value (hex)
Sync	0
Delay_Req	1
Pdelay_Req	2
Pdelay_Resp	3
Reserved	4-7
Follow_Up	8
Delay_Resp	9
Pdelay_Resp_Follow_Up	A
Announce	B
Signaling	C
Management	D
Reserved	E-F

Once a transmitted or received packet is determined to be a valid time sync packet, the Ethernet Transmit Event or Ethernet Receive Event is loaded onto the Event FIFO. The CPTS_EVENT_HIGH register contains the Message Type and Sequence ID values from the original time sync packet. The CPTS_EVENT_LOW register contains the time stamp value when the packet arrived at the corresponding port.

24.11.4.10.6 CPTS Interrupt Handling

When an event is push onto the Event FIFO, an interrupt can be generated to indicate to software that a time sync event occurred. The following steps should be taken to process time sync events using interrupts:

1. Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_INT_ENABLE register.
2. Upon interrupt, read the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers values.
3. Set the EVENT_POP field (bit 0) of the CPTS_EVENT_POP register to pop the previously read value off of the event FIFO.
4. Process the interrupt as required by the application software.

Software has the option of processing more than a single event from the event FIFO in the interrupt service routine in the following way:

1. Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_INT_ENABLE register.
2. Upon interrupt, read the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers values.
3. Set the EVENT_POP bit of the CPTS_EVENT_POP register to pop the previously read value off of the event FIFO.
4. Wait for an amount of time greater than four CPTS_RFT_CLK periods plus four MAIN_CLK periods.
5. Read the TS_PEND_RAW bit in the CPTS_INTSTAT_RAW register to determine if another valid event is in the event FIFO. If it is asserted, go to step 2; otherwise, go to step 6.
6. Process the interrupt(s) as required by the application software.

Software also has the option of disabling the interrupt and polling the TS_PEND_RAW bit of the CPTS_INTSTAT_RAW register to determine if a valid event is on the event FIFO.

24.11.4.11 CPPI Buffer Descriptors

The buffer descriptor is a central part of the GMAC_SW Ethernet Subsystem and is how the application software describes Ethernet packets to be sent and empty buffers to be filled with incoming packet data.

Host Software sends and receives network frames via the CPPI compliant host interface. The host interface includes module registers and host memory data structures. The host memory data structures are buffer descriptors and data buffers. Buffer descriptors are data structures that contain information about a single data buffer. Buffer descriptors may be linked together to describe frames or queues of frames for transmission of data and free buffer queues available for received data.

Note

The 8K bytes of Ethernet Subsystem CPPI RAM begin at address 0x4848 6000 and end at 0x4848 7FFF from the GMAC_SW perspective. The buffer descriptors programmed to access the CPPI RAM memory should use this address range.

24.11.4.11.1 TX Buffer Descriptors

A TX buffer descriptor () is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Descriptor Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Buffer Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Buffer Offset																Buffer Length															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOP	EOP	OWNER	EOQ	TDOWNCMPLT	PASSCRC	RESERVED					TO_PORT_EN	RESERVED		TO_PORT		RESERVED					Packet Length										

24.11.4.11.1.1 CPPI TX Data Word 0

Next Descriptor Pointer

The next descriptor pointer points to the 32-bit word aligned memory address of the next buffer descriptor in the transmit queue. This pointer is used to create a linked list of buffer descriptors. If the value of this pointer is zero, then the current buffer is the last buffer in the queue. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC. The value of pNext should never be altered once the descriptor is in an active transmit queue, unless its current value is NULL. If the pNext pointer is initially NULL, and more packets need to be queued for transmit, the software application may alter this pointer to point to a newly appended descriptor. The EMAC will use the new pointer value and proceed to the next descriptor unless the pNext value has already been read. In this latter case, the transmitter will halt on the transmit channel in question, and the software application may restart it at that time. The software can detect this case by checking for an end of queue (EOQ) condition flag on the updated packet descriptor when it is returned by the EMAC

24.11.4.11.1.2 CPPI TX Data Word 1

Buffer Pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the buffer_pointer. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC.

24.11.4.11.1.3 CPPI TX Data Word 2

Buffer Offset

Indicates how many unused bytes are at the start of the buffer. A value of 0000h indicates that no unused bytes are at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 000Fh (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The host sets the Buffer_Offset value (which may be zero to the buffer length minus 1). Valid only on SOP.

Buffer Length

Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host sets the Buffer_Length. The Buffer_Length must be greater than zero.

24.11.4.11.1.4 CPPI TX Data Word 3

Packet Length

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the Buffer_Length fields should equal the Packet_Length. Valid only on SOP. The packet length must be greater than zero. The packet data will be truncated to the packet length if the packet length is shorter than the sum of the packet buffer descriptor buffer lengths. A host error occurs if the packet length is greater than the sum of the packet buffer descriptor buffer lengths.

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet. In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise,,the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not start of packet buffer.

1 - Start of packet buffer.

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not end of packet buffer.

1 - End of packet buffer.

Ownership (OWNER) Flag

When set this flag indicates that all the descriptors for the given packet (from SOP to EOP) are currently owned by the EMAC. This flag is set by the software application on the SOP packet descriptor before adding the descriptor to the transmit descriptor queue. For a single fragment packet, the SOP, EOP, and OWNER flags are all set. The OWNER flag is cleared by the EMAC once it is finished with all the descriptors for the given packet. Note that this flag is valid on SOP descriptors only.

0 - The packet is owned by the host

1 - The packet is owned by the port

End of Queue (EOQ) Flag

When set, this flag indicates that the descriptor in question was the last descriptor in the transmit queue for a given transmit channel, and that the transmitter has halted. This flag is initially cleared by the software application prior to adding the descriptor to the transmit queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet (the EOP flag is set), and there are no more descriptors in the transmit list (next descriptor pointer is NULL).

The software application can use this bit to detect when the EMAC transmitter for the corresponding channel has halted. This is useful when the application appends additional packet descriptors to a transmit queue list that is already owned by the EMAC. Note that this flag is valid on EOP descriptors only.

0 - The TX queue has more packets to transfer.

1 - The Descriptor buffer is the last buffer in the last packet in the queue.

Teardown Complete (TDOWNCMPLT) Flag

This flag is used when a transmit queue is being torn down, or aborted, instead of allowing it to be transmitted. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the SOP descriptor of each packet as it is aborted from transmission. Note that this flag is valid on SOP descriptors only. Also note that only the first packet in an unsent list has the TDOWNCMPLT flag set. Subsequent descriptors are not processed by the EMAC.

0 - The port has not completed the teardown process.

1 - The port has completed the commanded teardown process.

Pass CRC (PASSCRC) Flag

This flag is set by the software application in the SOP packet descriptor before it adds the descriptor to the transmit queue. Setting this bit indicates to the EMAC that the 4 byte Ethernet CRC is already present in the packet data, and that the EMAC should not generate its own version of the CRC. When the CRC flag is cleared, the EMAC generates and appends the 4-byte CRC. The buffer length and packet length fields do not include the CRC bytes. When the CRC flag is set, the 4-byte CRC is supplied by the software application and is already appended to the end of the packet data. The buffer length and packet length fields include the CRC bytes, as they are part of the valid packet data. Note that this flag is valid on SOP descriptors only.

0 - The CRC is not included with the packet data and packet length.

1 - The CRC is included with the packet data and packet length.

TO_PORT

Port number to send the directed packet to. This field is set by the host. This field is valid on SOP. Directed packets go to the directed port, but an ALE lookup is performed to determine untagged egress in VLAN_AWARE mode.

1 - Send the packet to port 1 if TO_PORT_EN is asserted.

2 - Send the packet to port 2 if TO_PORT_EN is asserted.

TO_PORT_ENABLE

Indicates when set that the packet is a directed packet to be sent to the TO_PORT field port number. This field is set by the host. The packet is sent to one port only (index not mask). This bit is valid on SOP.

0 - not a directed packet

1 - directed packet

24.11.4.11.2 RX Buffer Descriptors

A RX buffer descriptor () is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Descriptor Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Buffer Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					Buffer Offset											RESERVED					Buffer Length										

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SOP	EOP	OWNER	EOQ	TDOWNCMPLT	PASSCRC	LONG	SHORT	MAC_CTL	OVERRUN	PKT_ERR	RX_VLAN_ENCAP	FROM_PORT	RESERVED	Packet Length

24.11.4.11.2.1 CPPI RX Data Word 0

Next Descriptor Pointer

The 32-bit word aligned memory address of the next buffer descriptor in the RX queue. This is the mechanism used to reference the next buffer descriptor from the current buffer descriptor. If the value of this pointer is zero then the current buffer is the last buffer in the queue. The host sets the Next_Descriptor_Pointer.

24.11.4.11.2.2 CPPI RX Data Word 1

Buffer Pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the Buffer_Pointer.

24.11.4.11.2.3 CPPI RX Data Word 2

Buffer Offset

Indicates how many unused bytes are at the start of the buffer. A value of 0000h indicates that there are no unused bytes at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 000Fh (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The port writes the Buffer_Offset with the value from the CPDMA_RX_BUFFER_OFFSET register value. The host initializes the Buffer_Offset to zero for free buffers. The Buffer_Length must be greater than the CPDMARX_BUFFER_OFFSET register value. The buffer offset is valid only on SOP.

Buffer Length

Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host initializes the Buffer_Length, but the port may overwrite the host initiated value with the actual buffer length value on SOP and/or EOP buffer descriptors. SOP buffer length values will be overwritten if the packet size is less than the size of the buffer or if the offset is nonzero. EOP buffer length values will be overwritten if the entire buffer is not filled up with data. The Buffer_Length must be greater than zero.

24.11.4.11.2.4 CPPI RX Data Word 3

Packet Length

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the Buffer_Length fields should equal the Packet_Length. Valid only on SOP.

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet. In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on SOP descriptors.

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on EOP descriptors.

Ownership (OWNER) Flag

When set, this flag indicates that the descriptor is currently owned by the EMAC. This flag is set by the software application before adding the descriptor to the receive descriptor queue. This flag is cleared by the EMAC once it is finished with a given set of descriptors, associated with a received packet. The flag is updated by the EMAC on SOP descriptor only. So when the application identifies that the OWNER flag is cleared on an SOP descriptor, it may assume that all descriptors up to and including the first with the EOP flag set have been released by the EMAC. (Note that in the case of single buffer packets, the same descriptor will have both the SOP and EOP flags set.)

End of Queue (EOQ) Flag

When set, this flag indicates that the descriptor in question was the last descriptor in the receive queue for a given receive channel, and that the corresponding receiver channel has halted. This flag is initially cleared by the software application prior to adding the descriptor to the receive queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet received (also sets the EOP flag), and there are no more descriptors in the receive list (next descriptor pointer is NULL).The software application can use this bit to detect when the EMAC receiver for the corresponding channel has halted. This is useful when the application appends additional free buffer descriptors to an active receive queue. Note that this flag is valid on EOP descriptors only.

Teardown Complete (TDOWNCMPLT) Flag

This flag is used when a receive queue is being torn down, or aborted, instead of being filled with received data. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the descriptor of the first free buffer when the tear down occurs. No additional queue processing is performed.

Pass CRC (PASSCRC) Flag

This flag is set by the EMAC in the SOP buffer descriptor if the received packet includes the 4-byte CRC. This flag should be cleared by the software application before submitting the descriptor to the receive queue.

Long (Jabber) Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is a jabber frame and was not discarded because the RX_CEF_EN bit was set in the SL_MACCONTROL register. Jabber frames are frames that exceed the RXMAXLEN in length, and have CRC, code, or alignment errors.

Short (Fragment) Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is only a packet fragment and was not discarded because the RX_CSF_EN bit was set in the SL_MACCONTROL register.

Control Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is an EMAC control frame and was not discarded because the RX_CMF_EN bit was set in the SL_MACCONTROL register.

Overrun Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet was aborted due to a receive overrun.

Packet Error (PKT_ERR) Flag

Packet Contained Error on Ingress:

00 - no error

01 - CRC error on ingress

10 - Code error on ingress

11 - Align error on ingress

VLAN Encapsulated Packet (RX_VLAN_ENCAP)

Indicates when set that the packet data contains a 32-bit VLAN header word that is included in the packet byte count. This field is set by the port to be the value of the CPSW_CONTROL register RX_VLAN_ENCAP bit.

FROM_PORT

Indicates the port number that the packet was received on (ingress to the switch).

24.11.4.12 MDIO

The MII Management interface module implements the 802.3 serial management interface to interrogate and control two Ethernet PHYs simultaneously using a shared two-wire bus. Two user access registers (MDIO_USERACCESS0/MDIO_USERACCESS1) control and monitor up to two PHYs simultaneously.

24.11.4.12.1 MDIO Frame Formats

Table 24-915 shows the read format and Table 24-916 shows the write format of the 32-bit MII Management interface frames.

Table 24-915. MDIO Read Frame Format

Pre-amble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
FFFF FFFFh	01	10	AAAAA	RRRRR	Z0	DDDD.DDDD.DDDD.DDDD

Table 24-916. MDIO Write Frame Format

Pre-amble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
FFFF FFFFh	01	01	AAAAA	RRRRR	10	DDDD.DDDD.DDDD.DDDD

The default or idle state of the two wire serial interface is a logic one. All tri-state drivers should be disabled and the PHY's pull-up resistor will pull the MDIO line to a logic 1. Prior to initiating any other transaction, the station management entity shall send a preamble sequence of 32 contiguous logic 1 bits on the MDIO line with 32 corresponding cycles on MDCLK to provide the PHY with a pattern that it can use to establish synchronization. A PHY shall observe a sequence of 32 contiguous logic one bits on MDIO with 32 corresponding MDCLK cycles before it responds to any other transaction.

Preamble

The start of a frame is indicated by a preamble, which consists of a sequence of 32 contiguous bits all of which are a 1. This sequence provides the PHY a pattern to use to establish synchronization.

Start Delimiter

The preamble is followed by the start delimiter which is indicated by a 01 pattern. The pattern assures transitions from the default logic 1 state to logic 0, and back to logic 1.

Operation Code

The operation code for a read is 10, while the operation code for a write is a 01.

PHY Address

The PHY address is 5 bits allowing 32 unique values. The first bit transmitted is the MSB of the PHY address.

Register Address

The Register address is 5 bits allowing 32 registers to be addressed within each PHY. Refer to the 10/100 PHY address map for addresses of individual registers.

Turnaround

An idle bit time during which no device actively drives the MDIO signal shall be inserted between the register address field and the data field of a read frame in order to avoid contention. During a read frame, the PHY shall drive a zero bit onto MDIO for the first bit time following the idle bit and preceding the Data field. During a write frame, this field shall consist of a one bit followed by a zero bit.

Data

The Data field is 16 bits. The first bit transmitted and received is the MSB of the data word.

24.11.4.12.2 MDIO Functional Description

The MII Management I/F will remain idle until enabled by setting the ENABLE bit in the MDIO_CONTROL register. The MII Management I/F will then continuously poll the link status from within the Generic Status Register of all possible 32 PHY addresses in turn recording the results in the MDIO_LINK register. The LINKSEL bit in the MDIO_USERPHYSEL0/1 register determines the status input that is used. A change in the link status of the two PHYs being monitored will set the appropriate bit in the MDIO_LINKINTRAW register and the MDIO_LINKINTMASKED register, if enabled by the LINKINT_ENABLE bit in the MDIO_USERPHYSEL0/1 register.

The MDIO_ALIVE register is updated by the MII Management I/F module if the PHY acknowledged the read of the generic status register. In addition, any PHY register read transactions initiated by the host also cause the MDIO_ALIVE register to be updated.

At any time, the host can define a transaction for the MII Management interface module to undertake using the DATA, PHYADR, REGADR, and WRITE fields in a MDIO_USERACCESS0/1 register. When the host sets the GO bit in this register, the MII Management interface module will begin the transaction without any further intervention from the host. Upon completion, the MII Management interface will clear the GO bit and set the USERINTRAW bit in the MDIO_USERINTRAW register corresponding to the MDIO_USERACCESS0/1 register being used. The corresponding bit in the MDIO_USERINTMASKED register may also be set depending on the mask setting in the MDIO_USERINTMASKSET and MDIO_USERINTMASKCLR registers. A round-robin arbitration scheme is used to schedule transactions that may be queued by the host in different MDIO_USERACCESS0/1 registers. The host should check the status of the GO bit in the MDIO_USERACCESS0/1 register before initiating a new transaction to ensure that the previous transaction has completed. The host can use the ACK bit in the MDIO_USERACCESS0/1 register to determine the status of a read transaction.

It is necessary for software to use the MII Management interface module to setup the auto-negotiation parameters of each PHY attached to a MAC port, retrieve the negotiation results, and setup the SL_MACCONTROL register in the corresponding MAC.

24.11.5 GMAC_SW Programming Guide

24.11.5.1 Transmit Operation

After reset, the host must write zeroes to all TX DMA State head descriptor pointers. The TX port may then be enabled. To initiate packet transmission the host constructs transmit queues in memory (one or more packets for transmission) and then writes the appropriate TX DMA state head descriptor pointers. For each buffer added to a transmit queue, the host must initialize the TX buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
2. Write the Buffer Pointer with the byte aligned address of the buffer data
3. Write the Buffer Length with the number of bytes in the buffer
4. Write the Buffer Offset with the number of bytes in the offset to the data (nonzero with SOP only)
5. Set the SOP, EOP, and Ownership bits as appropriate
6. Clear the End Of Queue bit

The port begins TX packet transmission on a given channel when the host writes the channel's TX queue head descriptor pointer with the address of the first buffer descriptor in the queue (nonzero value). Each channel may have one or more queues, so each channel may have one or more head descriptor pointers. The first buffer descriptor for each TX packet must have the Start of Packet (SOP) bit and the Ownership bit set to one by the host. The last buffer descriptor for each TX packet must have the End of Packet (EOP) bit set to one by the host. The port will transmit packets until all queued packets have been transmitted and the queue(s) are empty. When each packet transmission is complete, the port will clear the Ownership bit in the packet's SOP buffer descriptor and issue an interrupt to the host by writing the packet's last buffer descriptor address to the queue's TX DMA State Completion Pointer. The interrupt is generated by the write, regardless of the value written. When the last packet in a queue has been transmitted, the port sets the End Of Queue bit in the EOP buffer descriptor, clears the Ownership bit in the SOP Descriptor, zeroes the appropriate DMA state head descriptor pointer, and then issues a TX interrupt to the host by writing to the queue's associated TX completion pointer (address of the last buffer descriptor processed by the port). The port issues a maskable level interrupt (which may then be routed through external interrupt control logic to the host).

On interrupt from the port, the host processes the buffer queue, detecting transmitted packets by the status of the Ownership bit in the SOP buffer descriptor. If the Ownership bit is cleared to zero, then the packet has been transmitted and the host may reclaim the buffers associated with the packet. The host continues queue processing until the end of the queue or until a SOP buffer descriptor is read that contains a set Ownership bit indicating that the packet transmission is not complete. The host determines that all packets in the queue have been transmitted when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, the End of Queue bit is set in the last packet EOP buffer descriptor, and the Next Descriptor Pointer of the last packet EOP buffer descriptor is zero. The host acknowledges an interrupt by writing the address of the last buffer descriptor to the queue's associated TX Completion Pointer in the TX DMA State. If the host written buffer address value is different from the buffer address written by the port, then the level interrupt remains asserted. If the host written buffer address value is equal to the port written value, then the level interrupt is de-asserted. The port write to the completion pointer actually stores the value in the state register (RAM). The host written value is actually not written to the register location. The host written value is compared to the register contents (which was written by the port) and if the two values are equal, the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet.

A mis-queued packet condition may occur when the host adds a packet to a queue for transmission as the port finishes transmitting the previous last packet in the queue. The mis-queued packet is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor. A mis-queued packet means that the port read the last EOP buffer descriptor before the host added the new last packet to the queue, so the port determined queue empty just before the last packet was added. The host corrects the mis-queued packet condition by initiating a new packet transfer for the mis-queued packet by writing the mis-queued packet's SOP buffer descriptor address to the appropriate DMA State TX Queue head Descriptor Pointer.

The host may add packets to the tail end of an active TX queue at any time by writing the Next Descriptor Pointer to the current last descriptor in the queue. If a TX queue is empty (inactive), the host may initiate packet transmission at any time by writing the appropriate TX DMA State head descriptor pointer. The host software should always check for and reinitiate transmission for mis-queued packets during queue processing on interrupt from the port. In order to preclude software underrun, the host should avoid adding buffers to an active queue for any TX packet that is not complete and ready for transmission.

The port determines that a packet is the last packet in the queue by detecting the End of Packet bit set with a zero Next Descriptor Pointer in the packet buffer descriptor. If the End of Packet bit is set and the Next Descriptor Pointer is nonzero, then the queue still contains one or more packets to be transmitted. If the EOP bit is set with a zero Next Descriptor Pointer, then the port will set the EOQ bit in the packet's EOP buffer descriptor and then zero the appropriate head descriptor pointer previous to interrupting the port (by writing the completion pointer) when the packet transmission is complete.

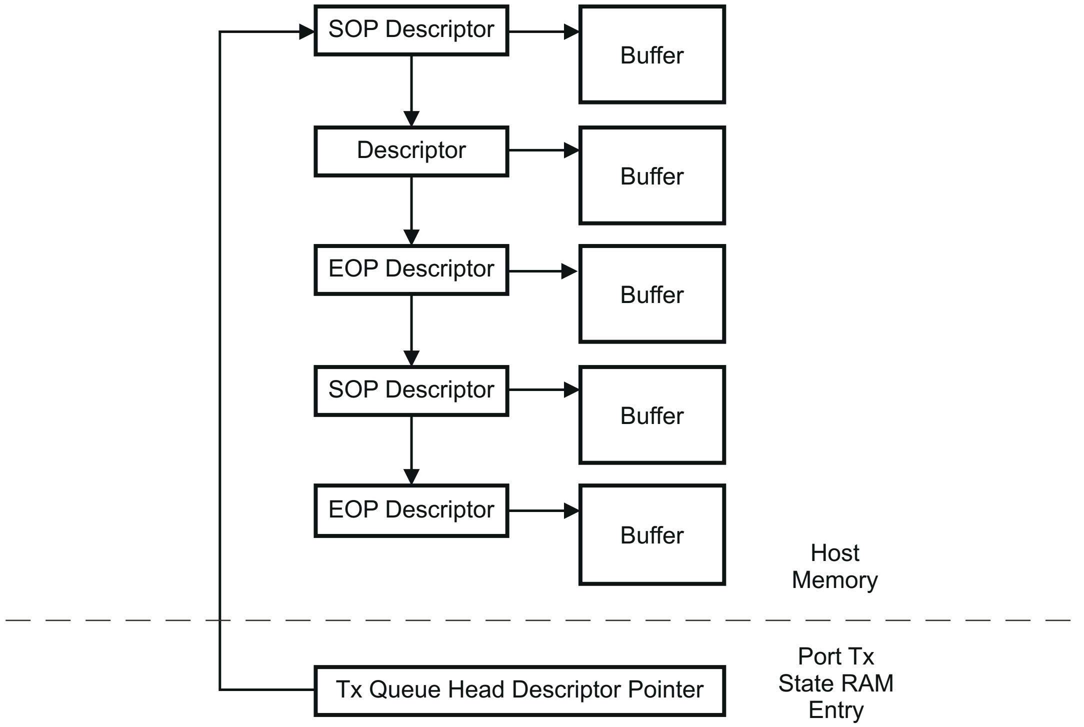


Figure 24-204. TX Queue Head Descriptor

24.11.5.2 Receive Operation

After reset, the host must write zeroes to all RX DMA State head descriptor pointers. The RX port may then be enabled. To initiate packet reception, the host constructs receive queues in memory and then writes the appropriate RX DMA state head descriptor pointer. For each RX buffer descriptor added to the queue, the host must initialize the RX buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
2. Write the Buffer Pointer with the byte aligned address of the buffer data
3. Clear the Offset field
4. Write the Buffer Length with the number of bytes in the buffer
5. Clear the SOP, EOP, and EOQ bits
6. Set the Ownership bit

The host enables packet reception on a given channel by writing the address of the first buffer descriptor in the queue (nonzero value) to the channel's head descriptor pointer in the channel's RX DMA state. When packet reception begins on a given channel, the port fills each RX buffer with data in order starting with the first buffer and proceeding through the RX queue. If the Buffer Offset in the RX DMA State is nonzero, then the port will begin writing data after the offset number of bytes in the SOP buffer. The port performs the following operations at the end of each packet reception:

1. Overwrite the buffer length in the packet's EOP buffer descriptor with the number of bytes actually received in the packet's last buffer. The host initialized value is the buffer size. The overwritten value will be less than or equal to the host initialized value.
2. Set the EOP bit in the packet's EOP buffer descriptor.
3. Set the EOQ bit in the packet's EOP buffer descriptor if the current packet is the last packet in the queue.
4. Overwrite the packet's SOP buffer descriptor Buffer Offset with the RX DMA state value (the host initialized the buffer descriptor Buffer Offset value to zero). All non SOP buffer descriptors must have a zero Buffer Offset initialized by the host.

5. Overwrite the packet's SOP buffer descriptor buffer length with the number of valid data bytes in the buffer. If the buffer is filled up, the buffer length will be the buffer size minus buffer offset.
6. Set the SOP bit in the packet's SOP buffer descriptor.
7. Write the SOP buffer descriptor Packet Length field.
8. Clear the Ownership bit in the packet's SOP buffer descriptor.
9. Issue an RX host interrupt by writing the address of the packet's last buffer descriptor to the queue's RX DMA State Completion Pointer. The interrupt is generated by the write to the RX DMA State Completion Pointer address location, regardless of the value written.

On interrupt the host processes the RX buffer queue detecting received packets by the status of the Ownership bit in each packet's SOP buffer descriptor. If the Ownership bit is cleared then the packet has been completely received and is available to be processed by the host. The host may continue RX queue processing until the end of the queue or until a buffer descriptor is read that contains a set Ownership bit indicating that the next packet's reception is not complete. The host determines that the RX queue is empty when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and the Next Descriptor Pointer in the EOP buffer descriptor is zero.

A mis-queued buffer may occur when the host adds buffers to a queue as the port finishes the reception of the previous last packet in the queue. The mis-queued buffer is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor. A mis-queued buffer means that the port read the last EOP buffer descriptor before the host added buffer descriptor(s) to the queue, so the port determined queue empty just before the host added more buffer descriptor(s). In the transmit case, the packet transmission is delayed by the time required for the host to determine the condition and reinitiate the transaction, but the packet is not actually lost. In the receive case, receive overrun condition may occur in the mis-queued buffer case. If a new packet reception is begun during the time that the port has determined the end of queue condition, then the received packet will overrun (start of packet overrun). If the mis-queued buffer occurs during the middle of a packet reception then middle of packet overrun may occur. If the mis-queued buffer occurs after the last packet has completed, and is corrected before the next packet reception begins, then overrun will not occur. The host acts on the mis-queued buffer condition by writing the added buffer descriptor address to the appropriate RX DMA State Head Descriptor Pointer.

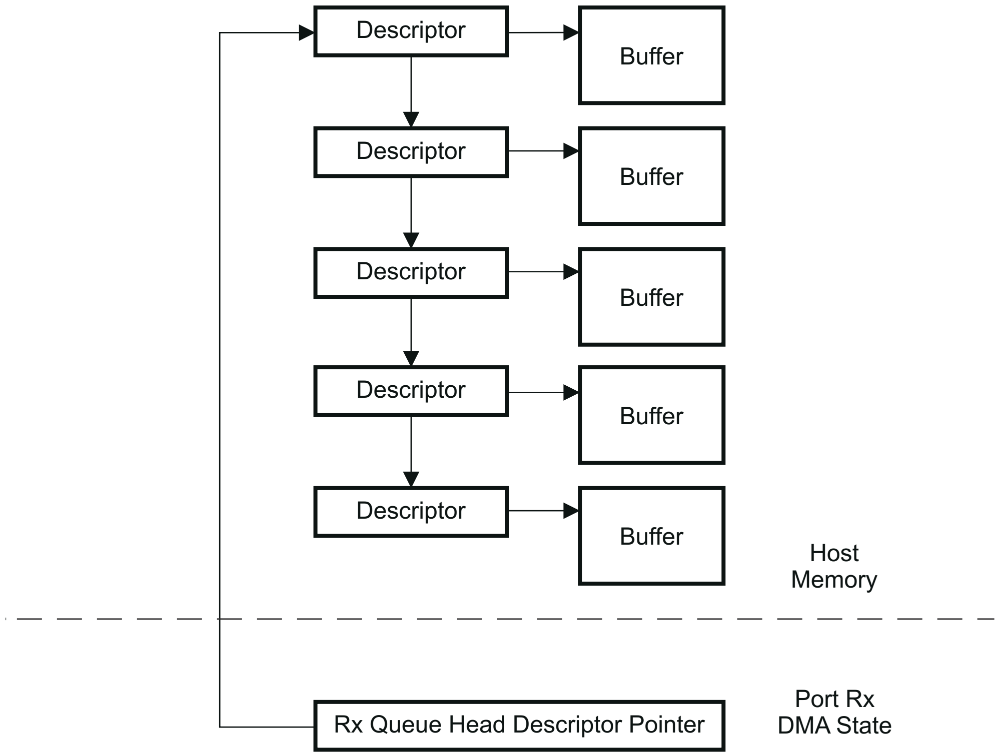


Figure 24-205. RX Queue Head Descriptor

24.11.5.3 MDIO Software Interface

24.11.5.3.1 Initializing the MDIO Module

The following steps are performed by the application software or device driver to initialize the MDIO device:

1. Configure the PREAMBLE and CLKDIV bits in the MDIO Control register (MDIO_CONTROL).
2. Enable the MDIO module by setting the ENABLE bit in MDIO_CONTROL.
3. The MDIO PHY alive status register (MDIO_ALIVE) can be read in polling fashion until a PHY connected to the system responded, and the MDIO PHY link status register (MDIO_LINK) can determine whether this PHY already has a link.
4. Setup the appropriate PHY addresses in the MDIO user PHY select register (MDIO_USERPHYSEL0/1), and set the LINKINTENB bit to enable a link change event interrupt if desirable.
5. If an interrupt on general MDIO register access is desired, set the corresponding bit in the MDIO user command complete interrupt mask set register (MDIO_USERINTMASKSET) to use the MDIO user access register (MDIO_USERACCESS0/1). Since only one PHY is used in this device, the application software can use one MDIO_USERACCESS0/1 to trigger a completion interrupt; the other MDIO_USERACCESS0/1 is not setup.

24.11.5.3.2 Writing Data To a PHY Register

The MDIO module includes a user access register (MDIO_USERACCESS0/1) to directly access a specified PHY device.To write a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (MDIO_USERACCESS0/1) is cleared.
2. Write to the GO, WRITE, REGADR, PHYADR, and DATA bits in MDIO_USERACCESS0/1 corresponding to the PHY and PHY register you want to write.
3. The write operation to the PHY is scheduled and completed by the MDIO module. Completion of the write operation can be determined by polling the GO bit in MDIO_USERACCESS0/1 for a 0.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIO_USERINTRAW) corresponding to MDIO_USERACCESS0/1 used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register

(MDIO_USERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIO_USERINTMASKED) and an interrupt is triggered on the host processor.

24.11.5.3.3 Reading Data From a PHY Register

The MDIO module includes a user access register (MDIO_USERACCESS0/1) to directly access a specified PHY device. To read a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (MDIO_USERACCESS*n*) is cleared.
2. Write to the GO, REGADR, and PHYADR bits in MDIO_USERACCESS0/1 corresponding to the PHY and PHY register you want to read.
3. The read data value is available in the DATA bits in MDIO_USERACCESS0/1 after the module completes the read operation on the serial bus. Completion of the read operation can be determined by polling the GO and ACK bits in MDIO_USERACCESS0/1. After the GO bit has cleared, the ACK bit is set on a successful read.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIO_USERINTRAW) corresponding to MDIO_USERACCESS0/1 used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (MDIO_USERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIO_USERINTMASKED) and an interrupt is triggered on the host processor.

24.11.5.4 Initialization and Configuration of CPSW

To configure the GMAC_SW Ethernet Subsystem for operation, the host must perform the following:

1. Select the Interface (G/MII, RGMII, RMII) Mode
2. Configure pads (PIN muxing), as per the interface selected.
3. Enable the GMAC_SW Ethernet Subsystem Clocks
4. Configure the PRCM registers CM_GMAC_CLKSTCTRL and CM_GMAC_GMAC_CLKCTRL to enable power and clocks to GMAC_SW Ethernet Subsystem.
5. Apply Soft Reset to GMAC_SW Subsystem, CPSW_3G, CPGMAC_SL1/2, and CPDMA
6. Initialize the HDPs (Header Description Pointer) and CPs (Completion Pointer) to NULL
7. Configure the Interrupts
8. Configure the CPSW_CONTROL register
9. Configure the CPSW_STAT_PORT_EN register
10. Configure the ALE
11. Configure the MDIO
12. Configure the CPDMA receive DMA controller
13. Configure the CPDMA transmit DMA controller
14. Configure the CPPI TX and RX Descriptors
15. Configure CPGMAC_SL1 and CPGMAC_SL2, as per the desired mode of operations.
16. Start up RX and TX DMA (Write to HDP of RX and TX)
17. Wait for the completion of Transfer (HDP cleared to 0)

24.11.6 GMAC_SW Register Manual

24.11.6.1 GMAC_SW Instance Summary

Table 24-917. GMAC_SW Instance Summary

Module Name	Module Base Address	Size
SS	0x4848 4000	80 Bytes
PORT	0x4848 4100	1792 Bytes
CPDMA	0x4848 4800	256 Bytes
STATS	0x4848 4900	128 Bytes
STATERAM	0x4848 4A00	288 Bytes
CPTS	0x4848 4C00	240 Bytes
ALE	0x4848 4D00	88 Bytes
SL1	0x4848 4D80	64 Bytes
SL2	0x4848 4DC0	64 Bytes
MDIO	0x4848 5000	192 Bytes
WR	0x4848 5200	352 Bytes
SPF1	0x4848 5C00	512 Bytes
SPF2	0x4848 5E00	512 Bytes

Note

CPPI RAM address space starts at 0x4848 6000 and is 8 KiB deep. For details about CPPI RAM, see Section 24.11.4.11, *CPPI Buffer Descriptors*..

24.11.6.2 SS Registers

24.11.6.2.1 SS Register Summary

Table 24-918. SS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SS Physical Address
CPSW_ID_VER	R	32	0x0000 0000	0x4848 4000
CPSW_CONTROL	RW	32	0x0000 0004	0x4848 4004
CPSW_SOFT_RESET	RW	32	0x0000 0008	0x4848 4008
CPSW_STAT_PORT_EN	RW	32	0x0000 000C	0x4848 400C
CPSW_PTYPE	RW	32	0x0000 0010	0x4848 4010
CPSW_SOFT_IDLE	RW	32	0x0000 0014	0x4848 4014
CPSW_THRU_RATE	RW	32	0x0000 0018	0x4848 4018
CPSW_GAP_THRESH	RW	32	0x0000 001C	0x4848 401C
CPSW_TX_START_WDS	RW	32	0x0000 0020	0x4848 4020
CPSW_FLOW_CONTROL	RW	32	0x0000 0024	0x4848 4024
CPSW_VLAN_LTYPE	RW	32	0x0000 0028	0x4848 4028
CPSW_TS_LTYPE	RW	32	0x0000 002C	0x4848 402C
CPSW_DLR_LTYPE	RW	32	0x0000 0030	0x4848 4030
CPSW_EEE_PRESCALE	RW	32	0x0000 0034	0x4848 4034

24.11.6.2.2 SS Register Description

Table 24-919. CPSW_ID_VER

Address Offset	0x0000 0000		
Physical Address	0x4848 4000	**Instance**	SS
Description	CPSW_3G ID version register		

Table 24-919. CPSW_ID_VER (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPSW_3G Revision Value	R	0x-

Table 24-920. CPSW_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4004	**Instance**	SS
Description	Switch control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											EEE_EN	DLR_EN	RX_VLAN_ENCAP	VLAN_AWARE	FIFO_LOOPBACK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	EEE_EN	EEE (Energy Efficient Ethernet) enable 0 – EEE is disabled. 1 – EEE is enabled	RW	0x0
3	DLR_EN	DLR enable 0 - DLR is disabled. DLR packets will not be moved to queue priority 3 and will not be separated out onto dlr_cpdma_ch. 1 - DLR is disabled. DLR packets be moved to destination port transmit queue priority 3 and will be separated out onto dlr_cpdma_ch when packet is to egress on port 0.	RW	0x0
2	RX_VLAN_ENCAP	Port 0 VLAN Encapsulation (egress): 0 - Port 0 receive packets (from CPSW_3G) are not VLAN encapsulated. 1 - Port 0 receive packets (from CPSW_3G) are VLAN encapsulated.	RW	0x0
1	VLAN_AWARE	VLAN Aware Mode: 0 - CPSW_3G is in the VLAN unaware mode. 1 - CPSW_3G is in the VLAN aware mode.	RW	0x0
0	FIFO_LOOPBACK	FIFO Loopback Mode 0 - Loopback is disabled 1 - FIFO Loopback mode enabled. Each packet received is turned around and sent out on the same port's transmit path. Port 2 receive is fixed on channel zero. The RXSOFOVERRUN statistic will increment for every packet sent in FIFO loopback mode.	RW	0x0

Table 24-921. CPSW_SOFT_RESET

Address Offset	0x0000 0008		
Physical Address	0x4848 4008	**Instance**	SS
Description	Soft reset register		

Table 24-921. CPSW_SOFT_RESET (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SOFT_RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the 3G logic (INT, REGS, CPPI, and SPF modules) to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.	RW	0x0

Table 24-922. CPSW_STAT_PORT_EN

Address Offset	0x0000 000C		
Physical Address	0x4848 400C	**Instance**	SS
Description	Statistics port enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													P2_STAT_EN	P1_STAT_EN	P0_STAT_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	P2_STAT_EN	Port 2 (GMII2 and Port 2 FIFO) Statistics Enable 0 - Port 2 statistics are not enabled. 1 - Port 2 statistics are enabled.	RW	0x0
1	P1_STAT_EN	Port 1 (GMII1 and Port 1 FIFO) Statistics Enable 0 - Port 1 statistics are not enabled. 1 - Port 1 statistics are enabled.	RW	0x0
0	P0_STAT_EN	Port 0 Statistics Enable 0 - Port 0 statistics are not enabled 1 - Port 0 statistics are enabled. FIFO overruns (SOFOVERRUNS) are the only port 0 statistics that are enabled to be kept.	RW	0x0

Table 24-923. CPSW_PTYPE

Address Offset	0x0000 0010		
Physical Address	0x4848 4010	**Instance**	SS
Description	Transmit priority type register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	P2_PRI3_SHAPE_EN	P2_PRI2_SHAPE_EN	P2_PRI1_SHAPE_EN	P1_PRI3_SHAPE_EN	P1_PRI2_SHAPE_EN	P1_PRI1_SHAPE_EN	RESERVED	P2_PTYPE_ESC	P1_PTYPE_ESC	P0_PTYPE_ESC	RESERVED	ESC_PRI_LD_VAL

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	P2_PRI3_SHAPE_EN	Port 2 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.	RW	0x0
20	P2_PRI2_SHAPE_EN	Port 2 Queue Priority 2 Transmit Shape Enable - If there are two shaping queues then they must be priorities 3 and 2.	RW	0x0
19	P2_PRI1_SHAPE_EN	Port 2 Queue Priority 1 Transmit Shape Enable - If there are three shaping queues all three bits should be set.	RW	0x0
18	P1_PRI3_SHAPE_EN	Port 1 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.	RW	0x0
17	P1_PRI2_SHAPE_EN	Port 1 Queue Priority 2 Transmit Shape Enable- If there are two shaping queues then they must be priorities 3 and 2.	RW	0x0
16	P1_PRI1_SHAPE_EN	Port 1 Queue Priority 1 Transmit Shape Enable- If there are three shaping queues all three bits should be set.	RW	0x0
15:11	RESERVED		R	0x0
10	P2_PTYPE_ESC	Port 2 Priority Type Escalate - 0 - Port 2 priority type fixed 1 - Port 2 priority type escalate Escalate should not be used with queue shaping.	RW	0x0
9	P1_PTYPE_ESC	Port 1 Priority Type Escalate - 0 - Port 1 priority type fixed 1 - Port 1 priority type escalate Escalate should not be used with queue shaping.	RW	0x0
8	P0_PTYPE_ESC	Port 0 Priority Type Escalate - 0 - Port 0 priority type fixed 1 - Port 0 priority type escalate Escalate should not be used with queue shaping.	RW	0x0
7:5	RESERVED		R	0x0
4:0	ESC_PRI_LD_VAL	Escalate Priority Load Value When a port is in escalate priority, this is the number of higher priority packets sent before the next lower priority is allowed to send a packet. Escalate priority allows lower priority packets to be sent at a fixed rate relative to the next higher priority.	RW	0x0

Table 24-924. CPSW_SOFT_IDLE

Address Offset	0x0000 0014		
Physical Address	0x4848 4014	**Instance**	SS
Description	Software idle		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SOFT_IDLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_IDLE	Software Idle - Setting this bit causes the switch fabric to stop forwarding packets at the next start of packet.	RW	0x0

Table 24-925. CPSW_THRU_RATE

Address Offset	0x0000 0018		
Physical Address	0x4848 4018	**Instance**	SS
Description	Throughput rate		
Type	RW		

31-16	15-12	11-4	3-0
RESERVED	SL_RX_THRU_RATE	RESERVED	CPDMA_THRU_RATE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:12	SL_RX_THRU_RATE	CPGMAC_SL Switch FIFO receive through rate. This register value is the maximum throughput of the ethernet ports to the crossbar SCR. The default is one 8-byte word for every 3 MAIN_CLK periods maximum.	RW	0x3
11:4	RESERVED		R	0x0
3:0	CPDMA_THRU_RATE	CPDMA Switch FIFO receive through rate. This register value is the maximum throughput of the CPDMA host port to the crossbar SCR. The default is one 8-byte word for every 3 MAIN_CLK periods maximum.	RW	0x3

Table 24-926. CPSW_GAP_THRESH

Address Offset	0x0000 001C		
Physical Address	0x4848 401C	**Instance**	SS
Description	CPGMAC_SL short gap threshold		
Type	RW		

31-5	4-0
RESERVED	GAP_THRESH

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	GAP_THRESH	CPGMAC_SL Short Gap Threshold - This is the CPGMAC_SL associated FIFO transmit block usage value for triggering TX_SHORT_GAP.	RW	0xB

Table 24-927. CPSW_TX_START_WDS

Address Offset	0x0000 0020		
Physical Address	0x4848 4020	**Instance**	SS
Description	Transmit start words		
Type	RW		

31-11	10-0
RESERVED	TX_START_WDS

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	TX_START_WDS	FIFO Packet Transmit (egress) Start Words. This value is the number of required packet words in the transmit FIFO before the packet egress will begin. This value is non-zero to preclude underrun. Decimal 32 is the recommended value. It should not be increased unnecessarily to prevent adding to the switch latency.	RW	0x20

Table 24-928. CPSW_FLOW_CONTROL

Address Offset	0x0000 0024		
Physical Address	0x4848 4024	**Instance**	SS
Description	Flow control		
Type	RW		

31:3	2	1	0
RESERVED	P2_FLOW_EN	P1_FLOW_EN	P0_FLOW_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	P2_FLOW_EN	Port 2 Receive flow control enable	RW	0x0
1	P1_FLOW_EN	Port 1 Receive flow control enable	RW	0x0
0	P0_FLOW_EN	Port 0 Receive flow control enable	RW	0x1

Table 24-929. CPSW_VLAN_LTYPE

Address Offset	0x0000 0028		
Physical Address	0x4848 4028	**Instance**	SS
Description	LTYPE1 and LTYPE 2 Register		
Type	RW		

31:16	15:0
VLAN_LTYPE2	VLAN_LTYPE1

Bits	Field Name	Description	Type	Reset
31:16	VLAN_LTYPE2	Time Sync VLAN LTYPE2 This VLAN LTYPE value is used for tx and rx. This is the inner VLAN if both are present.	RW	0x8100
15:0	VLAN_LTYPE1	Time Sync VLAN LTYPE1 This VLAN LTYPE value is used for tx and rx. This is the outer VLAN if both are present.	RW	0x8100

Table 24-930. CPSW_TS_LTYPE

Address Offset	0x0000 002C		
Physical Address	0x4848 402C	**Instance**	SS
Description	VLAN_LTYPE1 and VLAN_LTYPE2 Register		
Type	RW		

31:16	15:0
TS_LTYPE2	TS_LTYPE1

Bits	Field Name	Description	Type	Reset
31:16	TS_LTYPE2	Time Sync LTYPE2 This is an Ethertype value to match for tx and rx time sync packets.	RW	0x0
15:0	TS_LTYPE1	Time Sync LTYPE1 This is an ethertype value to match for tx and rx time sync packets.	RW	0x0

Table 24-931. CPSW_DLR_LTYPE

Address Offset	0x0000 0030		
Physical Address	0x4848 4030	**Instance**	SS
Description	DLR LTYPE register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DLR_LTYPE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	DLR_LTYPE	DLR LTYPE. This is the ethertype value to match for DLR packets.	RW	0x80E1

Table 24-932. CPSW_EEE_PRESCALE

Address Offset	0x0000 0034		
Physical Address	0x4848 4034	**Instance**	SS
Description	EEE Pre-scale Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				EEE_PRESCALE											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	EEE_PRESCALE	Energy Efficient Ethernet Pre-scale count load value – This value is loaded into the EEE pre-scale counter each time the pre-scale count decrements to zero. The EEE counters are enabled to decrement each time the pre-scale counter reaches zero (and the EEE counters are enabled to count time). If this value is zero then the EEE counters decrement on every clock. If this value is 0x001 then the counters decrement on every other clock (and so on).	RW	0x0

24.11.6.3 PORT Registers

24.11.6.3.1 PORT Register Summary

Table 24-933. PORT Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PORT Physical Address
P0_CONTROL	RW	32	0x0000 0000	0x4848 4100
P0_MAX_BLKS	RW	32	0x0000 0008	0x4848 4108
P0_BLK_CNT	RW	32	0x0000 000C	0x4848 410C
P0_TX_IN_CTL	RW	32	0x0000 0010	0x4848 4110
P0_PORT_VLAN	RW	32	0x0000 0014	0x4848 4114
P0_TX_PRI_MAP	RW	32	0x0000 0018	0x4848 4118
P0_CPDMA_TX_PRI_MAP	RW	32	0x0000 001C	0x4848 411C

Table 24-933. PORT Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PORT Physical Address
P0_CPDMA_RX_CH_MAP	RW	32	0x0000 0020	0x4848 4120
P0_RX_DSCP_PRI_MAP0	RW	32	0x0000 0030	0x4848 4130
P0_RX_DSCP_PRI_MAP1	RW	32	0x0000 0034	0x4848 4134
P0_RX_DSCP_PRI_MAP2	RW	32	0x0000 0038	0x4848 4138
P0_RX_DSCP_PRI_MAP3	RW	32	0x0000 003C	0x4848 413C
P0_RX_DSCP_PRI_MAP4	RW	32	0x0000 0040	0x4848 4140
P0_RX_DSCP_PRI_MAP5	RW	32	0x0000 0044	0x4848 4144
P0_RX_DSCP_PRI_MAP6	RW	32	0x0000 0048	0x4848 4148
P0_RX_DSCP_PRI_MAP7	RW	32	0x0000 004C	0x4848 414C
P0_IDLE2LPI	RW	32	0x0000 0050	0x4848 4150
P0_LPI2WAKE	RW	32	0x0000 0054	0x4848 4154
P1_CONTROL	RW	32	0x0000 0100	0x4848 4200
P1_MAX_BLKS	RW	32	0x0000 0108	0x4848 4208
P1_BLK_CNT	RW	32	0x0000 010C	0x4848 420C
P1_TX_IN_CTL	RW	32	0x0000 0110	0x4848 4210
P1_PORT_VLAN	RW	32	0x0000 0114	0x4848 4214
P1_TX_PRI_MAP	RW	32	0x0000 0118	0x4848 4218
P1_TS_SEQ_MTYPE	RW	32	0x0000 011C	0x4848 421C
P1_SA_LO	RW	32	0x0000 0120	0x4848 4220
P1_SA_HI	RW	32	0x0000 0124	0x4848 4224
P1_SEND_PERCENT	RW	32	0x0000 0128	0x4848 4228
P1_RX_DSCP_PRI_MAP0	RW	32	0x0000 0130	0x4848 4230
P1_RX_DSCP_PRI_MAP1	RW	32	0x0000 0134	0x4848 4234
P1_RX_DSCP_PRI_MAP2	RW	32	0x0000 0138	0x4848 4238
P1_RX_DSCP_PRI_MAP3	RW	32	0x0000 013C	0x4848 423C
P1_RX_DSCP_PRI_MAP4	RW	32	0x0000 0140	0x4848 4240
P1_RX_DSCP_PRI_MAP5	RW	32	0x0000 0144	0x4848 4244
P1_RX_DSCP_PRI_MAP6	RW	32	0x0000 0148	0x4848 4248
P1_RX_DSCP_PRI_MAP7	RW	32	0x0000 014C	0x4848 424C
P1_IDLE2LPI	RW	32	0x0000 0150	0x4848 4250
P1_LPI2WAKE	RW	32	0x0000 0154	0x4848 4254
P2_CONTROL	RW	32	0x0000 0200	0x4848 4300
P2_MAX_BLKS	RW	32	0x0000 0208	0x4848 4308
P2_BLK_CNT	RW	32	0x0000 020C	0x4848 430C
P2_TX_IN_CTL	RW	32	0x0000 0210	0x4848 4310
P2_PORT_VLAN	RW	32	0x0000 0214	0x4848 4314
P2_TX_PRI_MAP	RW	32	0x0000 0218	0x4848 4318
P2_TS_SEQ_MTYPE	RW	32	0x0000 021C	0x4848 431C
P2_SA_LO	RW	32	0x0000 0220	0x4848 4320
P2_SA_HI	RW	32	0x0000 0224	0x4848 4324
P2_SEND_PERCENT	RW	32	0x0000 0228	0x4848 4328
P2_RX_DSCP_PRI_MAP0	RW	32	0x0000 0230	0x4848 4330
P2_RX_DSCP_PRI_MAP1	RW	32	0x0000 0234	0x4848 4334
P2_RX_DSCP_PRI_MAP2	RW	32	0x0000 0238	0x4848 4338
P2_RX_DSCP_PRI_MAP3	RW	32	0x0000 023C	0x4848 433C

Table 24-933. PORT Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PORT Physical Address
P2_RX_DSCP_PRI_MAP4	RW	32	0x0000 0240	0x4848 4340
P2_RX_DSCP_PRI_MAP5	RW	32	0x0000 0244	0x4848 4344
P2_RX_DSCP_PRI_MAP6	RW	32	0x0000 0248	0x4848 4348
P2_RX_DSCP_PRI_MAP7	RW	32	0x0000 024C	0x4848 434C
P2_IDLE2LPI	RW	32	0x0000 0250	0x4848 4350
P2_LPI2WAKE	RW	32	0x0000 0254	0x4848 4354

24.11.6.3.2 PORT Register Description

Table 24-934. P0_CONTROL

Address Offset	0x0000 0000		
Physical Address	0x4848 4100	**Instance**	PORT
Description	CPSW PORT 0 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	P0_DLR_CPDMA_CH			RESERVED			P0_PASS_PRI_TAGGED	RESERVED		P0_VLAN_LTYPE2_EN	P0_VLAN_LTYPE1_EN	RESERVED			P0_DSCP_PRI_EN	RESERVED															

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	P0_DLR_CPDMA_CH	Port 0 DLR CPDMA Channel This field indicates the CPDMA channel that DLR packets will be received on.	RW	0x0
27:25	RESERVED		R	0x0
24	P0_PASS_PRI_TAGGED	Port 0 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P0_PORT_VLAN [11:0] 1 - Priority tagged packets are processed unchanged.	RW	0x0
23:22	RESERVED		R	0x0
21	P0_VLAN_LTYPE2_EN	Port 0 VLAN LTYPE 2 enable 0 - disabled 1 - enabled	RW	0x0
20	P0_VLAN_LTYPE1_EN	Port 0 VLAN LTYPE 1 enable 0 - disabled 1 - enabled	RW	0x0
19:17	RESERVED		R	0x0
16	P0_DSCP_PRI_EN	Port 0 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.	RW	0x0
15:0	RESERVED		RW	0x0

Table 24-935. Register Call Summary for Register P0_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Device Level Ring (DLR) Support: [0]
- PORT Register Summary: [1]

Table 24-936. P0_MAX_BLKS

Address Offset	0x0000 0008		
Physical Address	0x4848 4108	**Instance**	PORT
Description	CPSW PORT 0 maximum FIFO blocks register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							P0_TX_MAX_BLKS					P0_RX_MAX_BLKS			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P0_TX_MAX_BLKS	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x10 is the recommended value of P0_TX_MAX_BLKS. Port 0 should remain in flow control mode. 0xE is the minimum value P0_TX_MAX_BLKS.	RW	0x10
3:0	P0_RX_MAX_BLKS	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. 0x4 is the recommended value. 0x3 is the minimum value P0_RX_MAX_BLKS and 0x6 is the maximum value.	RW	0x4

Table 24-937. Register Call Summary for Register P0_MAX_BLKS

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-938. P0_BLK_CNT

Address Offset	0x0000 000C		
Physical Address	0x4848 410C	**Instance**	PORT
Description	CPSW PORT 0 FIFO block usage count (read only)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							P0_TX_BLK_CNT					P0_RX_BLK_CNT			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P0_TX_BLK_CNT	Port 0 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.	R	0x4
3:0	P0_RX_BLK_CNT	Port 0 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.	R	0x1

Table 24-939. Register Call Summary for Register P0_BLK_CNT

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-940. P0_TX_IN_CTL

Address Offset	0x0000 0010		
Physical Address	0x4848 4110	**Instance**	PORT
Description	CPSW PORT 0 transmit FIFO control		
Type	RW		

31:24	23:20	19:18	17:16	15:12	11:10	9:0
RESERVED	TX_RATE_EN	RESERVED	TX_IN_SEL	TX_BLKS_REM	RESERVED	TX_PRI_WDS

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:20	TX_RATE_EN	Transmit FIFO Input Rate Enable	RW	0x0
19:18	RESERVED		R	0x0
17:16	TX_IN_SEL	Transmit FIFO Input Queue Type Select 00 - Normal priority mode 01 - Dual MAC mode 10 - Rate Limit mode 11 - reserved Note that Dual MAC mode is not compatible with escalation or shaping because dual MAC mode forces round robin priority on FIFO egress. Rate-limiting and shaping are still available for Port 1 and Port 2 when Port 0 is set in dual MAC mode.	RW	0x0
15:12	TX_BLKS_REM	Transmit FIFO Input Blocks to subtract in dual MAC mode	RW	0x4
11:10	RESERVED		R	0x0
9:0	TX_PRI_WDS	Transmit FIFO Words in queue	RW	0xC0

Table 24-941. Register Call Summary for Register P0_TX_IN_CTL

Gigabit Ethernet Switch (GMAC_SW)

- FIFO Transmit Queue Control: [0] [1]
- PORT Register Summary: [2]

Table 24-942. P0_PORT_VLAN

Address Offset	0x0000 0014		
Physical Address	0x4848 4114	**Instance**	PORT
Description	CPSW PORT 0 VLAN register		
Type	RW		

31:16	15:13	12	11:0
RESERVED	PORT_PRI	PORT_CFI	PORT_VID

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:13	PORT_PRI	Port VLAN Priority (7 is highest priority)	RW	0x0

Bits	Field Name	Description	Type	Reset
12	PORT_CFI	Port CFI bit	RW	0x0
11:0	PORT_VID	Port VLAN ID	RW	0x0

Table 24-943. Register Call Summary for Register P0_PORT_VLAN

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 24-944. P0_TX_PRI_MAP

Address Offset	0x0000 0018		
Physical Address	0x4848 4118	**Instance**	PORT
Description	CPSW PORT 0 TX header priority to switch priority mapping register		
Type	RW		

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PRI7	Priority 7 - A packet header priority of 0x7 is given this switch queue priority.	RW	0x3
27:26	RESERVED		R	0x0
25:24	PRI6	Priority 6 - A packet header priority of 0x6 is given this switch queue priority.	RW	0x3
23:22	RESERVED		R	0x0
21:20	PRI5	Priority 5 - A packet header priority of 0x5 is given this switch queue priority.	RW	0x2
19:18	RESERVED		R	0x0
17:16	PRI4	Priority 4 - A packet header priority of 0x4 is given this switch queue priority.	RW	0x2
15:14	RESERVED		R	0x0
13:12	PRI3	Priority 3 - A packet header priority of 0x3 is given this switch queue priority.	RW	0x1
11:10	RESERVED		R	0x0
9:8	PRI2	Priority 2 - A packet header priority of 0x2 is given this switch queue priority.	RW	0x0
7:6	RESERVED		R	0x0
5:4	PRI1	Priority 1 - A packet header priority of 0x1 is given this switch queue priority.	RW	0x0
3:2	RESERVED		R	0x0
1:0	PRI0	Priority 0 - A packet header priority of 0x0 is given this switch queue priority.	RW	0x1

Table 24-945. Register Call Summary for Register P0_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-946. P0_CPDMA_TX_PRI_MAP

Address Offset	0x0000 001C		
Physical Address	0x4848 411C	**Instance**	PORT

Table 24-946. P0_CPDMA_TX_PRI_MAP (continued)

Description	CPSW CPDMA TX (PORT 0 RX) packet priority to header priority
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI7			RESERVED	PRI6			RESERVED	PRI5			RESERVED	PRI4			RESERVED	PRI3			RESERVED	PRI2			RESERVED	PRI1			RESERVED	PRI0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet pri of 0x7 is mapped (changed) to this header packet priority.	RW	0x7
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet pri of 0x6 is mapped (changed) to this header packet priority.	RW	0x6
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet pri of 0x5 is mapped (changed) to this header packet priority.	RW	0x5
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet pri of 0x4 is mapped (changed) to this header packet priority.	RW	0x4
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet pri of 0x3 is mapped (changed) to this header packet priority.	RW	0x3
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet pri of 0x2 is mapped (changed) to this header packet priority.	RW	0x2
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet pri of 0x1 is mapped (changed) to this header packet priority.	RW	0x1
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet pri of 0x0 is mapped (changed) to this header packet priority.	RW	0x0

Table 24-947. Register Call Summary for Register P0_CPDMA_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-948. P0_CPDMA_RX_CH_MAP

Address Offset	0x0000 0020		
Physical Address	0x4848 4120	**Instance**	PORT
Description	CPSW CPDMA RX (PORT 0 TX) switch priority to DMA channel		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	P2_PRI3			RESERVED	P2_PRI2			RESERVED	P2_PRI1			RESERVED	P2_PRI0			RESERVED	P1_PRI3			RESERVED	P1_PRI2			RESERVED	P1_PRI1			RESERVED	P1_PRI0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
30:28	P2_PRI3	Port 2 Priority 3 packets go to this CPDMA Rx Channel	RW	0x0
27	RESERVED		R	0x0
26:24	P2_PRI2	Port 2 Priority 2 packets go to this CPDMA Rx Channel	RW	0x0
23	RESERVED		R	0x0
22:20	P2_PRI1	Port 2 Priority 1 packets go to this CPDMA Rx Channel	RW	0x0
19	RESERVED		R	0x0
18:16	P2_PRI0	Port 2 Priority 0 packets go to this CPDMA Rx Channel	RW	0x0
15	RESERVED		R	0x0
14:12	P1_PRI3	Port 1 Priority 3 packets go to this CPDMA Rx Channel	RW	0x0
11	RESERVED		R	0x0
10:8	P1_PRI2	Port 1 Priority 2 packets go to this CPDMA Rx Channel	RW	0x0
7	RESERVED		R	0x0
6:4	P1_PRI1	Port 1 Priority 1 packets go to this CPDMA Rx Channel	RW	0x0
3	RESERVED		R	0x0
2:0	P1_PRI0	Port 1 Priority 0 packets go to this CPDMA Rx Channel	RW	0x0

Table 24-949. Register Call Summary for Register P0_CPDMA_RX_CH_MAP

Gigabit Ethernet Switch (GMAC_SW)

- Address Lookup Engine (ALE): [0]
- PORT Register Summary: [1]

Table 24-950. P0_RX_DSCP_PRI_MAP0

Address Offset	0x0000 0030		
Physical Address	0x4848 4130	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 0		
Type	RW		

31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:8	PRI2	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.	RW	0x0

Table 24-951. Register Call Summary for Register P0_RX_DSCP_PRI_MAP0

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-952. P0_RX_DSCP_PRI_MAP1

Address Offset	0x0000 0034		
Physical Address	0x4848 4134	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI15			RESERVED	PRI14			RESERVED	PRI13			RESERVED	PRI12			RESERVED	PRI11			RESERVED	PRI10			RESERVED	PRI9			RESERVED	PRI8		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI15	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI14	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI13	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI12	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI11	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI10	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI9	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI8	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.	RW	0x0

Table 24-953. Register Call Summary for Register P0_RX_DSCP_PRI_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-954. P0_RX_DSCP_PRI_MAP2

Address Offset	0x0000 0038		
Physical Address	0x4848 4138	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI23			RESERVED	PRI22			RESERVED	PRI21			RESERVED	PRI20			RESERVED	PRI19			RESERVED	PRI18			RESERVED	PRI17			RESERVED	PRI16		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI23	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI22	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI21	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI20	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI19	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI18	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI17	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI16	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.	RW	0x0

Table 24-955. Register Call Summary for Register P0_RX_DSCP_PRI_MAP2

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-956. P0_RX_DSCP_PRI_MAP3

Address Offset	0x0000 003C		
Physical Address	0x4848 413C	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PRI31	RESERVED	PRI30	RESERVED	PRI29	RESERVED	PRI28	RESERVED	PRI27	RESERVED	PRI26	RESERVED	PRI25	RESERVED	PRI24

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI31	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI30	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI29	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI28	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI27	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI26	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI25	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI24	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.	RW	0x0

Table 24-957. Register Call Summary for Register P0_RX_DSCP_PRI_MAP3

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-958. P0_RX_DSCP_PRI_MAP4

Address Offset	0x0000 0040		
Physical Address	0x4848 4140	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI39			RESERVED	PRI38			RESERVED	PRI37			RESERVED	PRI36			RESERVED	PRI35			RESERVED	PRI34			RESERVED	PRI33			RESERVED	PRI32		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI39	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI38	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.	RW	0x0

Bits	Field Name	Description	Type	Reset
23	RESERVED		R	0x0
22:20	PRI37	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI36	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI35	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI34	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI33	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI32	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.	RW	0x0

Table 24-959. Register Call Summary for Register P0_RX_DSCP_PRI_MAP4

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-960. P0_RX_DSCP_PRI_MAP5

Address Offset	0x0000 0044		
Physical Address	0x4848 4144	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI47			RESERVED	PRI46			RESERVED	PRI45			RESERVED	PRI44			RESERVED	PRI43			RESERVED	PRI42			RESERVED	PRI41			RESERVED	PRI40		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI47	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI46	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI45	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI44	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI43	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:8	PRI42	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI41	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI40	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.	RW	0x0

Table 24-961. Register Call Summary for Register P0_RX_DSCP_PRI_MAP5

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-962. P0_RX_DSCP_PRI_MAP6

Address Offset	0x0000 0048		
Physical Address	0x4848 4148	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI55			RESERVED	PRI54			RESERVED	PRI53			RESERVED	PRI52			RESERVED	PRI51			RESERVED	PRI50			RESERVED	PRI49			RESERVED	PRI48		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI55	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI54	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI53	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI52	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI51	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI50	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI49	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI48	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.	RW	0x0

Table 24-963. Register Call Summary for Register P0_RX_DSCP_PRI_MAP6

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-964. P0_RX_DSCP_PRI_MAP7

Address Offset	0x0000 004C		
Physical Address	0x4848 414C	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI63			RESERVED	PRI62			RESERVED	PRI61			RESERVED	PRI60			RESERVED	PRI59			RESERVED	PRI58			RESERVED	PRI57			RESERVED	PRI56		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI63	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI62	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI61	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI60	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI59	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI58	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI57	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI56	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.	RW	0x0

Table 24-965. Register Call Summary for Register P0_RX_DSCP_PRI_MAP7

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-966. P0_IDLE2LPI

Address Offset	0x0000 0050		
Physical Address	0x4848 4150	**Instance**	PORT
Description	Port 0 EEE Idle to LPI Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	P0_IDLE2LPI

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P0_IDLE2LPI	Port 0 EEE Idle to LPI counter load value - After CLKSTOP_REQ is asserted, this value is loaded into the port 0 idle to LPI counter on each clock that the port 0 transmit is not idle. Port 0 enters the transmit LPI state when this counter decrements to zero. This counter decrements each time the EEE prescale counter decrements to zero.	RW	0x0

Table 24-967. Register Call Summary for Register P0_IDLE2LPI

Gigabit Ethernet Switch (GMAC_SW)
- Energy Efficient Ethernet Support (802.3az): [0]
- PORT Register Summary: [1]
- PORT Register Description: [2]

Table 24-968. P0_LPI2WAKE

Address Offset	0x0000 0054		
Physical Address	0x4848 4154	**Instance**	PORT
Description	Port 0 EEE LPI to Wake Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												P0_LPI2WAKE																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P0_LPI2WAKE	Port 0 EEE LPI to wake counter load value – When the port is in the transmit LPI state and the CLKSTOP_REQ signal is deasserted, this value is loaded into the port 0 LPI to wake counter. Transmit packet operations may begin (resume) when the LPI to wake count decrements to zero (on the pre-scale count). This is the time that the CPDMA transmit must wait before transmit (switch ingress) packet operations begin (resume after wakeup).	RW	0x0

Table 24-969. Register Call Summary for Register P0_LPI2WAKE

Gigabit Ethernet Switch (GMAC_SW)
- Energy Efficient Ethernet Support (802.3az): [0]
- PORT Register Summary: [1]
- PORT Register Description: [2]

Table 24-970. P1_CONTROL

Address Offset	0x0000 0100		
Physical Address	0x4848 4200	**Instance**	PORT
Description	CPSW PORT 1 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	P1_TX_CLKSTOP_EN	P1_PASS_PRI_TAGGED	RESERVED	P1_VLAN_LTYPE2_EN	P1_VLAN_LTYPE1_EN	RESERVED	P1_DSCP_PRI_EN	P1_TS_107	P1_TS_320	P1_TS_319	P1_TS_132	P1_TS_131	P1_TS_130	P1_TS_129	P1_TS_TTL_NONZERO	P1_TS_UNI_EN	P1_TS_ANNEX_F_EN	P1_TS_ANNEX_E_EN	P1_TS_ANNEX_D_EN	P1_TS_LTYPE2_EN	P1_TS_LTYPE1_EN	P1_TS_TX_EN	P1_TS_RX_EN

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	P1_TX_CLKSTOP_EN	Port 1 Transmit clockstop enable 0 – RGMII transmit clockstop not enabled 1 – RGMII transmit clockstop enabled. The transmit clock will be stopped after the LPI state is entered (and indicated to the CPRGMII) and the P1_Idle2LPI time is counted (counter value reused). The P1_Idle2LPI counter value must be greater than 9 transmit clocks (slowest clock)	RW	0x0
24	P1_PASS_PRI_TAGGED	Port 1 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P1_PORT_VLAN [11:0] 1 - Priority tagged packets are processed unchanged.	RW	0x0
23:22	RESERVED		R	0x0
21	P1_VLAN_LTYPE2_EN	Port 1 VLAN LTYPE 2 enable 0 - disabled 1 - VLAN LTYPE2 enabled on transmit and receive	RW	0x0
20	P1_VLAN_LTYPE1_EN	Port 1 VLAN LTYPE 1 enable 0 - disabled 1 - VLAN LTYPE1 enabled on transmit and receive	RW	0x0
19:17	RESERVED		R	0x0
16	P1_DSCP_PRI_EN	Port 1 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.	RW	0x0
15	P1_TS_107	Port 1 Time Sync Destination IP Address 107 enable 0 – disabled 1 – destination IP address (dec) 224.0.0.107 is enabled.	RW	0x0
14	P1_TS_320	Port 1 Time Sync Destination Port Number 320 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.	RW	0x0
13	P1_TS_319	Port 1 Time Sync Destination Port Number 319 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.	RW	0x0
12	P1_TS_132	Port 1 Time Sync Destination IP Address 132 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.	RW	0x0
11	P1_TS_131	Port 1 Time Sync Destination IP Address 131 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.	RW	0x0

Bits	Field Name	Description	Type	Reset
10	P1_TS_130	Port 1 Time Sync Destination IP Address 130 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.	RW	0x0
9	P1_TS_129	Port 1 Time Sync Destination IP Address 129 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.	RW	0x0
8	P1_TS_TTL_NONZERO	Port 1 Time Sync Time To Live Non-zero enable. 0 = TTL must be zero. 1 = TTL may be any value.	RW	0x0
7	P1_TS_UNI_EN	Port 1 Time Sync Unicast Enable 0 – Unicast disabled 1 – Unicast enabled	RW	0x0
6	P1_TS_ANNEX_F_EN	Port 1 Time Sync Annex F enable 0 – Annex F disabled 1 – Annex F enabled	RW	0x0
5	P1_TS_ANNEX_E_EN	Port 1 Time Sync Annex E enable 0 – Annex E disabled 1 – Annex E enabled	RW	0x0
4	P1_TS_ANNEX_D_EN	Port 1 Time Sync Annex D enable 0 - Annex D disabled 1 - Annex D enabled	RW	0x0
3	P1_TS_LTYPE2_EN	Port 1 Time Sync LTYPE 2 enable 0 - disabled 1 - enabled	RW	0x0
2	P1_TS_LTYPE1_EN	Port 1 Time Sync LTYPE 1 enable 0 - disabled 1 - enabled	RW	0x0
1	P1_TS_TX_EN	Port 1 Time Sync Transmit Enable 0 - disabled 1 - enabled	RW	0x0
0	P1_TS_RX_EN	Port 1 Time Sync Receive Enable 0 - Port 1 Receive Time Sync disabled 1 - Port 1 Receive Time Sync enabled	RW	0x0

Table 24-971. Register Call Summary for Register P1_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-972. P1_MAX_BLKS

Address Offset	0x0000 0108		
Physical Address	0x4848 4208	**Instance**	PORT
Description	CPSW PORT 1 maximum FIFO blocks register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							P1_TX_MAX_BLKS					P1_RX_MAX_BLKS			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:4	P1_TX_MAX_BLKS	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of P1_TX_MAX_BLKS unless the port is in fullduplex flow control mode. In flow control mode, the P1_RX_MAX_BLKS will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in P1_RX_MAX_BLKS. 0xE is the minimum value for P1_TX_MAX_BLKS.	RW	0x11
3:0	P1_RX_MAX_BLKS	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The P1_TX_MAX_BLKS value must be decreased by the amount of increase in P1_RX_MAX_BLKS. 0x6 is the maximum value for P1_RX_MAX_BLKS.	RW	0x3

Table 24-973. Register Call Summary for Register P1_MAX_BLKS

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-974. P1_BLK_CNT

Address Offset	0x0000 010C		
Physical Address	0x4848 420C	**Instance**	PORT
Description	CPSW PORT 1 FIFO block usage count (read only)		
Type	R		

31–9	8–4	3–0
RESERVED	P1_TX_BLK_CNT	P1_RX_BLK_CNT

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P1_TX_BLK_CNT	Port 1 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.	R	0x4
3:0	P1_RX_BLK_CNT	Port 1 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.	R	0x1

Table 24-975. Register Call Summary for Register P1_BLK_CNT

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-976. P1_TX_IN_CTL

Address Offset	0x0000 0110		
Physical Address	0x4848 4210	**Instance**	PORT
Description	CPSW PORT 1 transmit FIFO control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	HOST_BLKS_REM	TX_RATE_EN	RESERVED	TX_IN_SEL	TX_BLKS_REM	RESERVED	TX_PRI_WDS

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	HOST_BLKS_REM	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.	RW	0x8
23:20	TX_RATE_EN	Transmit FIFO Input Rate Enable	RW	0x0
19:18	RESERVED		R	0x0
17:16	TX_IN_SEL	Transmit FIFO Input Queue Type Select 0x0 - Normal priority mode 0x1 - reserved 0x2 - Rate Limit mode 0x3 - reserved	RW	0x0
15:12	TX_BLKS_REM	Transmit FIFO Input blocks to subtract on non rate-limited traffic in rate limit mode.	RW	0x4
11:10	RESERVED		R	0x0
9:0	TX_PRI_WDS	Transmit FIFO Words in queue	RW	0xc0

Table 24-977. Register Call Summary for Register P1_TX_IN_CTL

Gigabit Ethernet Switch (GMAC_SW)

- Audio Video Bridging: [0] [1]
- PORT Register Summary: [2]

Table 24-978. P1_PORT_VLAN

Address Offset	0x0000 0114		
Physical Address	0x4848 4214	**Instance**	PORT
Description	CPSW PORT 1 VLAN register		
Type	RW		

31–16	15–13	12	11–0
RESERVED	PORT_PRI	PORT_CFI	PORT_VID

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:13	PORT_PRI	Port VLAN Priority (7 is highest priority)	RW	0x0
12	PORT_CFI	Port CFI bit	RW	0x0
11:0	PORT_VID	Port VLAN ID	RW	0x0

Table 24-979. Register Call Summary for Register P1_PORT_VLAN

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 24-980. P1_TX_PRI_MAP

Address Offset	0x0000 0118		
Physical Address	0x4848 4218	**Instance**	PORT
Description	CPSW PORT 1 TX header priority to switch priority mapping register		

Table 24-980. P1_TX_PRI_MAP (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRI7		RESERVED		PRI6		RESERVED		PRI5		RESERVED		PRI4		RESERVED		PRI3		RESERVED		PRI2		RESERVED		PRI1		RESERVED		PRI0	

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PRI7	Priority 7 - A packet header priority of 0x7 is given this switch queue priority	RW	0x3
27:26	RESERVED		R	0x0
25:24	PRI6	Priority 6 - A packet header priority of 0x6 is given this switch queue priority	RW	0x3
23:22	RESERVED		R	0x0
21:20	PRI5	Priority 5 - A packet header priority of 0x5 is given this switch queue priority	RW	0x2
19:18	RESERVED		R	0x0
17:16	PRI4	Priority 4 - A packet header priority of 0x4 is given this switch queue priority	RW	0x2
15:14	RESERVED		R	0x0
13:12	PRI3	Priority 3 - A packet header priority of 0x3 is given this switch queue priority	RW	0x1
11:10	RESERVED		R	0x0
9:8	PRI2	Priority 2 - A packet header priority of 0x2 is given this switch queue priority	RW	0x0
7:6	RESERVED		R	0x0
5:4	PRI1	Priority 1 - A packet header priority of 0x1 is given this switch queue priority	RW	0x0
3:2	RESERVED		R	0x0
1:0	PRI0	Priority 0 - A packet header priority of 0x0 is given this switch queue priority	RW	0x1

Table 24-981. Register Call Summary for Register P1_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-982. P1_TS_SEQ_MTYPE

Address Offset	0x0000 011C		
Physical Address	0x4848 421C	**Instance**	PORT
Description	CPSW PORT 1 time sync sequence ID offset and message type.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										P1_TS_SEQ_ID_OFFSET						P1_TS_MSG_TYPE_EN															

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:16	P1_TS_SEQ_ID_OFFSET	Port 1 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.	RW	0x1E

Bits	Field Name	Description	Type	Reset
15:0	P1_TS_MSG_TYPE_EN	Port 1 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.).	RW	0x0

Table 24-983. Register Call Summary for Register P1_TS_SEQ_MTYPE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-984. P1_SA_LO

Address Offset	0x0000 0120		
Physical Address	0x4848 4220	**Instance**	PORT
Description	CPSW CPGMAC_SL1 source address low register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MACSRCADDR_7_0								MACSRCADDR_15_8							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	MACSRCADDR_7_0	Source Address Lower 8 bits (byte 0)	RW	0x0
7:0	MACSRCADDR_15_8	Source Address bits 15:8 (byte 1)	RW	0x0

Table 24-985. Register Call Summary for Register P1_SA_LO

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-986. P1_SA_HI

Address Offset	0x0000 0124		
Physical Address	0x4848 4224	**Instance**	PORT
Description	CPSW CPGMAC_SL1 source address high register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACSRCADDR_23_16								MACSRCADDR_31_24								MACSRCADDR_39_32								MACSRCADDR_47_40							

Bits	Field Name	Description	Type	Reset
31:24	MACSRCADDR_23_16	Source Address bits 23:16 (byte 2)	RW	0x0
23:16	MACSRCADDR_31_24	Source Address bits 31:24 (byte 3)	RW	0x0
15:8	MACSRCADDR_39_32	Source Address bits 39:32 (byte 4)	RW	0x0
7:0	MACSRCADDR_47_40	Source Address bits 47:40 (byte 5)	RW	0x0

Table 24-987. Register Call Summary for Register P1_SA_HI

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-988. P1_SEND_PERCENT

Address Offset	0x0000 0128		
Physical Address	0x4848 4228	**Instance**	PORT
Description	CPSW PORT 1 transmit queue send percentages		

Table 24-988. P1_SEND_PERCENT (continued)

Type	RW

31–23	22–16	15	14–8	7	6–0
RESERVED	PRI3_SEND_PERCENT	RESERVED	PRI2_SEND_PERCENT	RESERVED	PRI1_SEND_PERCENT

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:16	PRI3_SEND_PERCENT	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when CPSW_PTYPE[18] P1_PRI3_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
15	RESERVED		R	0x0
14:8	PRI2_SEND_PERCENT	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when CPSW_PTYPE[17] P1_PRI2_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
7	RESERVED		R	0x0
6:0	PRI1_SEND_PERCENT	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the CPSW_PTYPE[16] P1_PRI1_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0

Table 24-989. Register Call Summary for Register P1_SEND_PERCENT

Gigabit Ethernet Switch (GMAC_SW)

- Audio Video Bridging: [0]
- PORT Register Summary: [1]

Table 24-990. P1_RX_DSCP_PRI_MAP0

Address Offset	0x0000 0130		
Physical Address	0x4848 4230	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 0		
Type	RW		

31	30–28	27	26–24	23	22–20	19	18–16	15	14–12	11	10–8	7	6–4	3	2–0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
30:28	PRI7	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.	RW	0x0

Table 24-991. Register Call Summary for Register P1_RX_DSCP_PRI_MAP0

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-992. P1_RX_DSCP_PRI_MAP1

Address Offset	0x0000 0134		
Physical Address	0x4848 4234	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI15			RESERVED	PRI14			RESERVED	PRI13			RESERVED	PRI12			RESERVED	PRI11			RESERVED	PRI10			RESERVED	PRI9			RESERVED	PRI8		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI15	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI14	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI13	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	PRI12	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI11	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI10	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI9	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI8	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.	RW	0x0

Table 24-993. Register Call Summary for Register P1_RX_DSCP_PRI_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-994. P1_RX_DSCP_PRI_MAP2

Address Offset	0x0000 0138		
Physical Address	0x4848 4238	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI23			RESERVED	PRI22			RESERVED	PRI21			RESERVED	PRI20			RESERVED	PRI19			RESERVED	PRI18			RESERVED	PRI17			RESERVED	PRI16		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI23	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI22	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI21	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI20	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI19	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI18	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:4	PRI17	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI16	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.	RW	0x0

Table 24-995. Register Call Summary for Register P1_RX_DSCP_PRI_MAP2

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-996. P1_RX_DSCP_PRI_MAP3

Address Offset	0x0000 013C		
Physical Address	0x4848 423C	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI31			RESERVED	PRI30			RESERVED	PRI29			RESERVED	PRI28			RESERVED	PRI27			RESERVED	PRI26			RESERVED	PRI25			RESERVED	PRI24		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI31	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI30	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI29	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI28	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI27	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI26	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI25	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI24	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.	RW	0x0

Table 24-997. Register Call Summary for Register P1_RX_DSCP_PRI_MAP3

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-998. P1_RX_DSCP_PRI_MAP4

Address Offset	0x0000 0140		
Physical Address	0x4848 4240	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI39			RESERVED	PRI38			RESERVED	PRI37			RESERVED	PRI36			RESERVED	PRI35			RESERVED	PRI34			RESERVED	PRI33			RESERVED	PRI32		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI39	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI38	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI37	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI36	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI35	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI34	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI33	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI32	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.	RW	0x0

Table 24-999. Register Call Summary for Register P1_RX_DSCP_PRI_MAP4

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1000. P1_RX_DSCP_PRI_MAP5

Address Offset	0x0000 0144		
Physical Address	0x4848 4244	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI47			RESERVED	PRI46			RESERVED	PRI45			RESERVED	PRI44			RESERVED	PRI43			RESERVED	PRI42			RESERVED	PRI41			RESERVED	PRI40		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI47	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI46	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI45	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI44	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI43	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI42	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI41	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI40	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.	RW	0x0

Table 24-1001. Register Call Summary for Register P1_RX_DSCP_PRI_MAP5

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1002. P1_RX_DSCP_PRI_MAP6

Address Offset	0x0000 0148		
Physical Address	0x4848 4248	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI55			RESERVED	PRI54			RESERVED	PRI53			RESERVED	PRI52			RESERVED	PRI51			RESERVED	PRI50			RESERVED	PRI49			RESERVED	PRI48		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI55	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI54	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI53	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	PRI52	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI51	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI50	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI49	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI48	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.	RW	0x0

Table 24-1003. Register Call Summary for Register P1_RX_DSCP_PRI_MAP6

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1004. P1_RX_DSCP_PRI_MAP7

Address Offset	0x0000 014C		
Physical Address	0x4848 424C	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI63			RESERVED	PRI62			RESERVED	PRI61			RESERVED	PRI60			RESERVED	PRI59			RESERVED	PRI58			RESERVED	PRI57			RESERVED	PRI56		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI63	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI62	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI61	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI60	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI59	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI58	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:4	PRI57	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI56	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.	RW	0x0

Table 24-1005. Register Call Summary for Register P1_RX_DSCP_PRI_MAP7

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1006. P1_IDLE2LPI

Address Offset	0x0000 0150		
Physical Address	0x4848 4250	**Instance**	PORT
Description	Port 1 EEE Idle to LPI Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												P1_IDLE2LPI																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P1_IDLE2LPI	Port 1 EEE Idle to LPI counter load value - After CLKSTOP_REQ is asserted, this value is loaded into the port 1 idle to LPI counter on each clock that the port 1 transmit is not idle. Port 0 enters the transmit LPI state when this counter decrements to zero. This counter decrements each time the EEE prescale counter decrements to zero.	RW	0x0

Table 24-1007. Register Call Summary for Register P1_IDLE2LPI

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 24-1008. P1_LPI2WAKE

Address Offset	0x0000 0154		
Physical Address	0x4848 4254	**Instance**	PORT
Description	Port 1 EEE LPI to Wake Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												P1_LPI2WAKE																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P1_LPI2WAKE	Port 1 EEE LPI to wake counter load value – When the port is in the transmit LPI state and the CLKSTOP_REQ signal is deasserted, this value is loaded into the port 1 LPI to wake counter. Transmit packet operations may begin (resume) when the LPI to wake count decrements to zero (on the pre-scale count). This is the time that the CPDMA transmit must wait before transmit (switch ingress) packet operations begin (resume after wakeup).	RW	0x0

Table 24-1009. Register Call Summary for Register P1_LPI2WAKE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 24-1010. P2_CONTROL

Address Offset	0x0000 0200		
Physical Address	0x4848 4300	**Instance**	PORT
Description	CPSW_3GF PORT 2 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						P2_TX_CLKSTOP_EN	P2_PASS_PRI_TAGGED	RESERVED		P2_VLAN_LTYPE2_EN	P2_VLAN_LTYPE1_EN	RESERVED			P2_DSCP_PRI_EN	P2_TS_107	P2_TS_320	P2_TS_319	P2_TS_132	P2_TS_131	P2_TS_130	P2_TS_129	P2_TS_TTL_NONZERO	P2_TS_UNI_EN	P2_TS_ANNEX_F_EN	P2_TS_ANNEX_E_EN	P2_TS_ANNEX_D_EN	P2_TS_LTYPE2_EN	P2_TS_LTYPE1_EN	P2_TS_TX_EN	P2_TS_RX_EN

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	P2_TX_CLKSTOP_EN	Port 2 Transmit clockstop enable 0 – RGMII transmit clockstop not enabled 1 – RGMII transmit clockstop enabled. The transmit clock will be stopped after the LPI state is entered (and indicated to the CPRGMII) and the P2_Idle2LPI time is counted (counter value reused). The P2_Idle2LPI counter value must be greater than 9 transmit clocks (slowest clock)	RW	0x0
24	P2_PASS_PRI_TAGGED	Port 2 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P2_PORT_VLAN [11:0] 1 - Priority tagged packets are processed unchanged.	RW	0x0
23:22	RESERVED		R	0x0
21	P2_VLAN_LTYPE2_EN	Port 2 VLAN LTYPE 2 enable 0 - disabled 1 - VLAN LTYPE2 enabled on transmit and receive	RW	0x0
20	P2_VLAN_LTYPE1_EN	Port 2 VLAN LTYPE 1 enable 0 - disabled 1 - VLAN LTYPE1 enabled on transmit and receive	RW	0x0
19:17	RESERVED		R	0x0
16	P2_DSCP_PRI_EN	Port 0 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.	RW	0x0
15	P2_TS_107	Port 2 Time Sync Destination IP Address 107 enable 0 – disabled 1 – destination IP address (dec) 224.0.0.107 is enabled.	RW	0x0
14	P2_TS_320	Port 2 Time Sync Destination Port Number 320 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	P2_TS_319	Port 2 Time Sync Destination Port Number 319 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.	RW	0x0
12	P2_TS_132	Port 2 Time Sync Destination IP Address 132 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.	RW	0x0
11	P2_TS_131	Port 2 Time Sync Destination IP Address 131 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.	RW	0x0
10	P2_TS_130	Port 2 Time Sync Destination IP Address 130 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.	RW	0x0
9	P2_TS_129	Port 2 Time Sync Destination IP Address 129 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.	RW	0x0
8	P2_TS_TTL_NONZERO	Port 2 Time Sync Time To Live Non-zero enable. 0 = TTL must be zero. 1 = TTL may be any value.	RW	0x0
7	P2_TS_UNI_EN	Port 2 Time Sync Unicast Enable 0 – Unicast disabled 1 – Unicast enabled	RW	0x0
6	P2_TS_ANNEX_F_EN	Port 2 Time Sync Annex F enable 0 – Annex F disabled 1 – Annex F enabled	RW	0X0
5	P2_TS_ANNEX_E_EN	Port 2 Time Sync Annex E enable 0 – Annex E disabled 1 – Annex E enabled	RW	0X0
4	P2_TS_ANNEX_D_EN	Port 2 Time Sync Annex D enable 0 - Annex D disabled 1 - Annex D enabled	RW	0x0
3	P2_TS_LTYPE2_EN	Port 2 Time Sync LTYPE 2 enable 0 - disabled 1 - enabled	RW	0x0
2	P2_TS_LTYPE1_EN	Port 2 Time Sync LTYPE 1 enable 0 - disabled 1 - enabled	RW	0x0
1	P2_TS_TX_EN	Port 2 Time Sync Transmit Enable 0 - disabled 1 - enabled	RW	0x0
0	P2_TS_RX_EN	Port 2 Time Sync Receive Enable 0 - Port 1 Receive Time Sync disabled 1 - Port 1 Receive Time Sync enabled	RW	0x0

Table 24-1011. Register Call Summary for Register P2_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1012. P2_MAX_BLKS

Address Offset	0x0000 0208		
Physical Address	0x4848 4308	**Instance**	PORT
Description	CPSW PORT 2 maximum FIFO blocks register		

Table 24-1012. P2_MAX_BLKS (continued)

Type	RW

31–9	8–4	3–0
RESERVED	P2_TX_MAX_BLKS	P2_RX_MAX_BLKS

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		RW	0x0
8:4	P2_TX_MAX_BLKS	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of P2_TX_MAX_BLKS unless the port is in fullduplex flow control mode. In flow control mode, the P2_RX_MAX_BLKS will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in P2_RX_MAX_BLKS. 0xE is the minimum value P2_TX_MAX_BLKS.	RW	0x11
3:0	P2_RX_MAX_BLKS	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The P2_TX_MAX_BLKS value must be decreased by the amount of increase in P2_RX_MAX_BLKS. 0x3 is the minimum value P2_RX_MAX_BLKS and 0x6 is the maximum value.	RW	0x3

Table 24-1013. Register Call Summary for Register P2_MAX_BLKS

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1014. P2_BLK_CNT

Address Offset	0x0000 020C		
Physical Address	0x4848 430C	**Instance**	PORT
Description	CPSW PORT 2 FIFO block usage count (read only)		
Type	R		

31–9	8–4	3–0
RESERVED	P2_TX_BLK_CNT	P2_RX_BLK_CNT

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P2_TX_BLK_CNT	Port 2 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.	R	0x4
3:0	P2_RX_BLK_CNT	Port 2 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.	R	0x1

Table 24-1015. Register Call Summary for Register P2_BLK_CNT

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1016. P2_TX_IN_CTL

Address Offset	0x0000 0210		
Physical Address	0x4848 4310	**Instance**	PORT
Description	CPSW PORT 2 transmit FIFO control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HOST_BLKS_REM				TX_RATE_EN				RESERVED		TX_IN_SEL		TX_BLKS_REM				RESERVED		TX_PRI_WDS									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		RW	0x0
27:24	HOST_BLKS_REM	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.	RW	0x8
23:20	TX_RATE_EN	Transmit FIFO Input Rate Enable	RW	0x0
19:18	RESERVED		RW	0x0
17:16	TX_IN_SEL	Transmit FIFO Input Queue Type Select 0x0 - Normal priority mode 0x1 - reserved 0x2 - Rate Limit mode 0x3 - reserved	RW	0x0
15:12	TX_BLKS_REM	Transmit FIFO Input blocks to subtract on non rate-limited traffic in rate limit mode.	RW	0x4
11:10	RESERVED		RW	0x0
9:0	TX_PRI_WDS	Transmit FIFO Words in queue	RW	0xc0

Table 24-1017. Register Call Summary for Register P2_TX_IN_CTL

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1018. P2_PORT_VLAN

Address Offset	0x0000 0214		
Physical Address	0x4848 4314	**Instance**	PORT
Description	CPSW PORT 2 VLAN register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PORT_PRI			PORT_CFI	PORT_VID											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:13	PORT_PRI	Port VLAN Priority (7 is highest priority)	RW	0x0
12	PORT_CFI	Port CFI bit	RW	0x0
11:0	PORT_VID	Port VLAN ID	RW	0x0

Table 24-1019. Register Call Summary for Register P2_PORT_VLAN

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 24-1020. P2_TX_PRI_MAP

Address Offset	0x0000 0218		
Physical Address	0x4848 4318	**Instance**	PORT
Description	CPSW PORT 2 TX header priority to switch priority mapping register		
Type	RW		

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PRI7	Priority 7 - A packet header priority of 0x7 is given this switch queue priority.	RW	0x3
27:26	RESERVED		R	0x0
25:24	PRI6	Priority 6 - A packet header priority of 0x6 is given this switch queue priority.	RW	0x3
23:22	RESERVED		R	0x0
21:20	PRI5	Priority 5 - A packet header priority of 0x5 is given this switch queue priority.	RW	0x2
19:18	RESERVED		R	0x0
17:16	PRI4	Priority 4 - A packet header priority of 0x4 is given this switch queue priority.	RW	0x2
15:14	RESERVED		R	0x0
13:12	PRI3	Priority 3 - A packet header priority of 0x3 is given this switch queue priority.	RW	0x1
11:10	RESERVED		R	0x0
9:8	PRI2	Priority 2 - A packet header priority of 0x2 is given this switch queue priority.	RW	0x0
7:6	RESERVED		R	0x0
5:4	PRI1	Priority 1 - A packet header priority of 0x1 is given this switch queue priority.	RW	0x0
3:2	RESERVED		R	0x0
1:0	PRI0	Priority 0 - A packet header priority of 0x0 is given this switch queue priority.	RW	0x1

Table 24-1021. Register Call Summary for Register P2_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1022. P2_TS_SEQ_MTYPE

Address Offset	0x0000 021C		
Physical Address	0x4848 431C	**Instance**	PORT
Description	CPSW_3GF PORT 2 time sync sequence ID offset and message type.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	P2_TS_SEQ_ID_OFFSET	P2_TS_MSG_TYPE_EN

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:16	P2_TS_SEQ_ID_OFFSET	Port 2 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.	RW	0x1E
15:0	P2_TS_MSG_TYPE_EN	Port 2 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.).	RW	0x0

Table 24-1023. Register Call Summary for Register P2_TS_SEQ_MTYPE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1024. P2_SA_LO

Address Offset	0x0000 0220		
Physical Address	0x4848 4320	**Instance**	PORT
Description	CPSW CPGMAC_SL2 source address low register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MACSRCADDR_7_0								MACSRCADDR_15_8							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	MACSRCADDR_7_0	Source Address Lower 8 bits (byte 0)	RW	0x0
7:0	MACSRCADDR_15_8	Source Address bits 15:8 (byte 1)	RW	0x0

Table 24-1025. Register Call Summary for Register P2_SA_LO

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1026. P2_SA_HI

Address Offset	0x0000 0224		
Physical Address	0x4848 4324	**Instance**	PORT
Description	CPSW CPGMAC_SL2 source address high register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACSRCADDR_23_16								MACSRCADDR_31_23								MACSRCADDR_39_32								MACSRCADDR_47_40							

Bits	Field Name	Description	Type	Reset
31:24	MACSRCADDR_23_16	Source Address bits 23:16 (byte 2)	RW	0x0
23:16	MACSRCADDR_31_23	Source Address bits 31:23 (byte 3)	RW	0x0
15:8	MACSRCADDR_39_32	Source Address bits 39:32 (byte 4)	RW	0x0
7:0	MACSRCADDR_47_40	Source Address bits 47:40 (byte 5)	RW	0x0

Table 24-1027. Register Call Summary for Register P2_SA_HI

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1028. P2_SEND_PERCENT

Address Offset	0x0000 0228		
Physical Address	0x4848 4328	**Instance**	PORT
Description	CPSW PORT 2 transmit queue send percentages		
Type	RW		

31:23	22:16	15	14:8	7	6:0
RESERVED	PRI3_SEND_PERCENT	RESERVED	PRI2_SEND_PERCENT	RESERVED	PRI1_SEND_PERCENT

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:16	PRI3_SEND_PERCENT	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when the CPSW_PTYPE[21] P2_PRI3_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
15	RESERVED		R	0x0
14:8	PRI2_SEND_PERCENT	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when the CPSW_PTYPE[20] P2_PRI2_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
7	RESERVED		R	0x0
6:0	PRI1_SEND_PERCENT	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the CPSW_PTYPE[19] P2_PRI1_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0

Table 24-1029. Register Call Summary for Register P2_SEND_PERCENT

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1030. P2_RX_DSCP_PRI_MAP0

Address Offset	0x0000 0230		
Physical Address	0x4848 4330	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.	RW	0x0

Table 24-1031. Register Call Summary for Register P2_RX_DSCP_PRI_MAP0

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1032. P2_RX_DSCP_PRI_MAP1

Address Offset	0x0000 0234		
Physical Address	0x4848 4334	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI15			RESERVED	PRI14			RESERVED	PRI13			RESERVED	PRI12			RESERVED	PRI11			RESERVED	PRI10			RESERVED	PRI9			RESERVED	PRI8		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI15	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI14	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.	RW	0x0

Bits	Field Name	Description	Type	Reset
23	RESERVED		R	0x0
22:20	PRI13	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI12	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI11	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI10	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI9	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI8	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.	RW	0x0

Table 24-1033. Register Call Summary for Register P2_RX_DSCP_PRI_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1034. P2_RX_DSCP_PRI_MAP2

Address Offset	0x0000 0238		
Physical Address	0x4848 4338	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI23			RESERVED	PRI22			RESERVED	PRI21			RESERVED	PRI20			RESERVED	PRI19			RESERVED	PRI18			RESERVED	PRI17			RESERVED	PRI16		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI23	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI22	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI21	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI20	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI19	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:8	PRI18	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI17	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI16	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.	RW	0x0

Table 24-1035. Register Call Summary for Register P2_RX_DSCP_PRI_MAP2

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1036. P2_RX_DSCP_PRI_MAP3

Address Offset	0x0000 023C		
Physical Address	0x4848 433C	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI31			RESERVED	PRI30			RESERVED	PRI29			RESERVED	PRI28			RESERVED	PRI27			RESERVED	PRI26			RESERVED	PRI25			RESERVED	PRI24		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI31	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI30	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI29	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI28	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI27	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI26	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI25	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI24	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.	RW	0x0

Table 24-1037. Register Call Summary for Register P2_RX_DSCP_PRI_MAP3

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1038. P2_RX_DSCP_PRI_MAP4

Address Offset	0x0000 0240		
Physical Address	0x4848 4340	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI39			RESERVED	PRI38			RESERVED	PRI37			RESERVED	PRI36			RESERVED	PRI35			RESERVED	PRI34			RESERVED	PRI33			RESERVED	PRI32		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI39	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI38	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI37	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI36	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI35	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI34	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI33	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI32	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.	RW	0x0

Table 24-1039. Register Call Summary for Register P2_RX_DSCP_PRI_MAP4

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1040. P2_RX_DSCP_PRI_MAP5

Address Offset	0x0000 0244		
Physical Address	0x4848 4344	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PRI47	RESERVED	PRI46	RESERVED	PRI45	RESERVED	PRI44	RESERVED	PRI43	RESERVED	PRI42	RESERVED	PRI41	RESERVED	PRI40

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI47	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI46	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI45	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI44	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI43	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI42	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI41	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI40	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.	RW	0x0

Table 24-1041. Register Call Summary for Register P2_RX_DSCP_PRI_MAP5

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1042. P2_RX_DSCP_PRI_MAP6

Address Offset	0x0000 0248		
Physical Address	0x4848 4348	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI55			RESERVED	PRI54			RESERVED	PRI53			RESERVED	PRI52			RESERVED	PRI51			RESERVED	PRI50			RESERVED	PRI49			RESERVED	PRI48		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI55	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI54	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.	RW	0x0

Bits	Field Name	Description	Type	Reset
23	RESERVED		R	0x0
22:20	PRI53	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI52	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI51	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI50	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI49	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI48	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.	RW	0x0

Table 24-1043. Register Call Summary for Register P2_RX_DSCP_PRI_MAP6

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1044. P2_RX_DSCP_PRI_MAP7

Address Offset	0x0000 024C		
Physical Address	0x4848 434C	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI63			RESERVED	PRI62			RESERVED	PRI61			RESERVED	PRI60			RESERVED	PRI59			RESERVED	PRI58			RESERVED	PRI57			RESERVED	PRI56		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI63	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI62	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI61	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI60	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI59	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:8	PRI58	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI57	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI56	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.	RW	0x0

Table 24-1045. Register Call Summary for Register P2_RX_DSCP_PRI_MAP7

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 24-1046. P2_IDLE2LPI

Address Offset	0x0000 0250		
Physical Address	0x4848 4350	**Instance**	PORT
Description	Port 2 EEE Idle to LPI Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												P2_IDLE2LPI																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P2_IDLE2LPI	Port 2 EEE Idle to LPI counter load value - After CLKSTOP_REQ is asserted, this value is loaded into the port 2 idle to LPI counter on each clock that the port 2 transmit is not idle. Port 2 enters the transmit LPI state when this counter decrements to zero. This counter decrements each time the EEE prescale counter decrements to zero.	RW	0x0

Table 24-1047. Register Call Summary for Register P2_IDLE2LPI

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 24-1048. P2_LPI2WAKE

Address Offset	0x0000 0254		
Physical Address	0x4848 4354	**Instance**	PORT
Description	Port 2 EEE LPI to Wake Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												P2_LPI2WAKE																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19:0	P2_LPI2WAKE	Port 2 EEE LPI to wake counter load value – When the port is in the transmit LPI state and the CLKSTOP_REQ signal is deasserted, this value is loaded into the port 2 LPI to wake counter. Transmit packet operations may begin (resume) when the LPI to wake count decrements to zero (on the pre-scale count). This is the time that the CPDMA transmit must wait before transmit (switch ingress) packet operations begin (resume after wakeup).	RW	0x0

Table 24-1049. Register Call Summary for Register P2_LPI2WAKE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

24.11.6.4 CPDMA registers

24.11.6.4.1 CPDMA Register Summary

Table 24-1050. CPDMA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CPDMA Physical Address
CPDMA_TX_IDVER	R	32	0x0000 0000	0x4848 4800
CPDMA_TX_CONTROL	RW	32	0x0000 0004	0x4848 4804
CPDMA_TX_TEARDOWN	RW	32	0x0000 0008	0x4848 4808
CPDMA_RX_IDVER	R	32	0x0000 0010	0x4848 4810
CPDMA_RX_CONTROL	RW	32	0x0000 0014	0x4848 4814
CPDMA_RX_TEARDOWN	RW	32	0x0000 0018	0x4848 4818
CPDMA_SOFT_RESET	RW	32	0x0000 001C	0x4848 481C
CPDMA_DMACONTROL	RW	32	0x0000 0020	0x4848 4820
CPDMA_DMASTATUS	R	32	0x0000 0024	0x4848 4824
CPDMA_RX_BUFFER_OFFSET	RW	32	0x0000 0028	0x4848 4828
CPDMA_EMCONTROL	RW	32	0x0000 002C	0x4848 482C
CPDMA_TX_PRI0_RATE	RW	32	0x0000 0030	0x4848 4830
CPDMA_TX_PRI1_RATE	RW	32	0x0000 0034	0x4848 4834
CPDMA_TX_PRI2_RATE	RW	32	0x0000 0038	0x4848 4838
CPDMA_TX_PRI3_RATE	RW	32	0x0000 003C	0x4848 483C
CPDMA_TX_PRI4_RATE	RW	32	0x0000 0040	0x4848 4840
CPDMA_TX_PRI5_RATE	RW	32	0x0000 0044	0x4848 4844
CPDMA_TX_PRI6_RATE	RW	32	0x0000 0048	0x4848 4848
CPDMA_TX_PRI7_RATE	RW	32	0x0000 004C	0x4848 484C
CPDMA_TX_INTSTAT_RAW	R	32	0x0000 0080	0x4848 4880
CPDMA_TX_INTSTAT_MASKED	R	32	0x0000 0084	0x4848 4884
CPDMA_TX_INTMASK_SET	W	32	0x0000 0088	0x4848 4888
CPDMA_TX_INTMASK_CLEAR	W	32	0x0000 008C	0x4848 488C
CPDMA_IN_VECTOR	R	32	0x0000 0090	0x4848 4890
CPDMA_EOI_VECTOR	RW	32	0x0000 0094	0x4848 4894
CPDMA_RX_INTSTAT_RAW	R	32	0x0000 00A0	0x4848 48A0
CPDMA_RX_INTSTAT_MASKED	R	32	0x0000 00A4	0x4848 48A4
CPDMA_RX_INTMASK_SET	RW	32	0x0000 00A8	0x4848 48A8
CPDMA_RX_INTMASK_CLEAR	RW	32	0x0000 00AC	0x4848 48AC
CPDMA_DMA_INTSTAT_RAW	R	32	0x0000 00B0	0x4848 48B0
CPDMA_DMA_INTSTAT_MASKED	R	32	0x0000 00B4	0x4848 48B4

Table 24-1050. CPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CPDMA Physical Address
CPDMA_DMA_INTMASK_SET	W	32	0x0000 00B8	0x4848 48B8
CPDMA_DMA_INTMASK_CLEAR	RW	32	0x0000 00BC	0x4848 48BC
CPDMA_RX0_PENDTHRESH	RW	32	0x0000 00C0	0x4848 48C0
CPDMA_RX1_PENDTHRESH	RW	32	0x0000 00C4	0x4848 48C4
CPDMA_RX2_PENDTHRESH	RW	32	0x0000 00C8	0x4848 48C8
CPDMA_RX3_PENDTHRESH	RW	32	0x0000 00CC	0x4848 48CC
CPDMA_RX4_PENDTHRESH	RW	32	0x0000 00D0	0x4848 48D0
CPDMA_RX5_PENDTHRESH	RW	32	0x0000 00D4	0x4848 48D4
CPDMA_RX6_PENDTHRESH	RW	32	0x0000 00D8	0x4848 48D8
CPDMA_RX7_PENDTHRESH	RW	32	0x0000 00DC	0x4848 48DC
CPDMA_RX0_FREEBUFFER	W	32	0x0000 00E0	0x4848 48E0
CPDMA_RX1_FREEBUFFER	W	32	0x0000 00E4	0x4848 48E4
CPDMA_RX2_FREEBUFFER	W	32	0x0000 00E8	0x4848 48E8
CPDMA_RX3_FREEBUFFER	W	32	0x0000 00EC	0x4848 48EC
CPDMA_RX4_FREEBUFFER	W	32	0x0000 00F0	0x4848 48F0
CPDMA_RX5_FREEBUFFER	W	32	0x0000 00F4	0x4848 48F4
CPDMA_RX6_FREEBUFFER	W	32	0x0000 00F8	0x4848 48F8
CPDMA_RX7_FREEBUFFER	W	32	0x0000 00FC	0x4848 48FC

24.11.6.4.2 CPDMA Register Description

Table 24-1051. CPDMA_TX_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4800	**Instance**	CPDMA
Description	CPDMA_REGS TX revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPDMA TX Revision Value	R	0x-

Table 24-1052. Register Call Summary for Register CPDMA_TX_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1053. CPDMA_TX_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4804	**Instance**	CPDMA
Description	CPDMA_REGS TX control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TX_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TX_EN	TX Enable 0 - Disabled 1 - Enabled	RW	0x0

Table 24-1054. Register Call Summary for Register CPDMA_TX_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 24-1055. CPDMA_TX_TEARDOWN

Address Offset	0x0000 0008		
Physical Address	0x4848 4808	**Instance**	CPDMA
Description	CPDMA_REGS TX teardown register		
Type	RW		

31	30–3	2–0
TX_TDN_RDY	RESERVED	TX_TDN_CH

Bits	Field Name	Description	Type	Reset
31	TX_TDN_RDY	Tx Teardown Ready - read as zero, but is always assumed to be one (unused).	R	0x0
30:3	RESERVED		R	0x0
2:0	TX_TDN_CH	Tx Teardown Channel - Transmit channel teardown is commanded by writing the encoded value of the transmit channel to be torn down. The teardown register is read as zero.	RW	0x0

Table 24-1056. Register Call Summary for Register CPDMA_TX_TEARDOWN

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 24-1057. CPDMA_RX_IDVER

Address Offset	0x0000 0010		
Physical Address	0x4848 4810	**Instance**	CPDMA
Description	CPDMA_REGS RX revision register		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	RX Revision Value	R	0x-

Table 24-1058. Register Call Summary for Register CPDMA_RX_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1059. CPDMA_RX_CONTROL

Address Offset	0x0000 0014		
Physical Address	0x4848 4814	**Instance**	CPDMA
Description	CPDMA_REGS RX control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RX_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	RX_EN	RX DMA Enable 0 - Disabled 1 - Enabled	RW	0x0

Table 24-1060. Register Call Summary for Register CPDMA_RX_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 24-1061. CPDMA_RX_TEARDOWN

Address Offset	0x0000 0018		
Physical Address	0x4848 4818	**Instance**	CPDMA
Description	CPDMA_REGS RX teardown register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_TDN_RDY	RESERVED																												RX_TDN_CH		

Bits	Field Name	Description	Type	Reset
31	RX_TDN_RDY	Teardown Ready - read as zero, but is always assumed to be one (unused).	R	0x0
30:3	RESERVED		R	0x0
2:0	RX_TDN_CH	Rx Teardown Channel -Receive channel teardown is commanded by writing the encoded value of the receive channel to be torn down. The teardown register is read as zero.	RW	0x0

Table 24-1062. Register Call Summary for Register CPDMA_RX_TEARDOWN

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 24-1063. CPDMA_SOFT_RESET

Address Offset	0x0000 001C		
Physical Address	0x4848 481C	**Instance**	CPDMA
Description	CPDMA_REGS soft reset register		
Type	RW		

31–1	0
RESERVED	SOFT_RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the CPDMA logic to be reset. Software reset occurs when the RX and TX DMA Controllers are in an idle state to avoid locking up the VBUSP bus. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.	RW	0x0

Table 24-1064. Register Call Summary for Register CPDMA_SOFT_RESET

Gigabit Ethernet Switch (GMAC_SW)

- Software Reset: [0]
- CPDMA Register Summary: [1]

Table 24-1065. CPDMA_DMACONTROL

Address Offset	0x0000 0020		
Physical Address	0x4848 4820	**Instance**	CPDMA
Description	CPDMA_REGS CPDMA control register		
Type	RW		

31–16	15–8	7–5	4	3	2	1	0
RESERVED	TX_RLIM	RESERVED	RX_CEF	CMD_IDLE	RX_OFFLEN_BLOCK	RX_OWNERSHIP	TX_PTYPE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
15:8	TX_RLIM	Transmit Rate Limit Channel Bus 00000000 - no rate-limited channels 10000000 - channel 7 is rate-limited 11000000 - channels 7 downto 6 are rate-limited 11100000 - channels 7 downto 5 are rate-limited 11110000 - channels 7 downto 4 are rate-limited 11111000 - channels 7 downto 3 are rate-limited 11111100 - channels 7 downto 2 are rate-limited 11111110 - channels 7 downto 1 are rate-limited 11111111 - channels 7 downto 0 are rate-limited all others invalid - this bus must be set MSB towards LSB. TX_PTYPE must be set if any TX_RLIM bit is set for fixed priority.	RW	0x0
7:5	RESERVED		R	0x0
4	RX_CEF	RX Copy Error Frames Enable - Enables DMA overrun frames to be transferred to memory (up to the point of overrun). The overrun error bit will be set in the frame EOP buffer descriptor. Overrun frame data will be filtered when RX_CEF is not set. Frames coming from the receive FIFO with other error bits set are not effected by this bit. 0 - Frames containing overrun errors are filtered. 1 - Frames containing overrun errors are transferred to memory.	RW	0x0
3	CMD_IDLE	Command Idle 0 - Idle not commanded 1 - Idle Commanded (read IDLE in CPDMA_DMASTATUS)	RW	0x0
2	RX_OFFLEN_BLOCK	Receive Offset/Length word write block. 0 - Do not block the DMA writes to the receive buffer descriptor offset/buffer length word. The offset/buffer length word is written as specified in CPPI 3.0. 1 - Block all CPDMA DMA controller writes to the receive buffer descriptor offset/buffer length words during CPPI packet processing. when this bit is set, the CPDMA will never write the third word to any receive buffer descriptor.	RW	0x0
1	RX_OWNERSHIP	Receive Ownership Write Bit Value. 0 - The CPDMA writes the receive ownership bit to zero at the end of packet processing as specified in CPPI 3.0. 1 - The CPDMA writes the receive ownership bit to one at the end of packet processing. Users who do not use the ownership mechanism can use this mode to preclude the necessity of software having to set this bit each time the buffer descriptor is used.	RW	0x0
0	TX_PTYPE	Transmit Queue Priority Type 0 - The queue uses a round robin scheme to select the next channel for transmission. 1 - The queue uses a fixed (channel 7 highest priority) priority scheme to select the next channel for transmission	RW	0x0

Table 24-1066. Register Call Summary for Register CPDMA_DMACONTROL

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0] [1] [2]
- FIFO Transmit Queue Control: [3] [4]
- Audio Video Bridging: [5]
- CPDMA Register Summary: [6]

Table 24-1067. CPDMA_DMASTATUS

Address Offset	0x0000 0024

Table 24-1067. CPDMA_DMASTATUS (continued)

Physical Address	0x4848 4824	**Instance**	CPDMA
Description	CPDMA_REGS CPDMA status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLE	RESERVED							TX_HOST_ERR_CODE				RESERVED	TX_ERR_CH			RX_HOST_ERR_CODE				RESERVED	RX_ERR_CH			RESERVED							

Bits	Field Name	Description	Type	Reset
31	IDLE	Idle Status Bit - Indicates when set that the CPDMA is not transferring a packet on transmit or receive.	R	0x0
30:24	RESERVED		R	0x0
23:20	TX_HOST_ERR_CODE	TX Host Error Code - This field is set to indicate CPDMA detected TX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. A zero packet length is an error, but it is not detected. 0x0 - No error 0x1 - SOP error. 0x2 - Ownership bit not set in SOP buffer. 0x3 - Zero Next Buffer Descriptor Pointer Without EOP 0x4 - Zero Buffer Pointer. 0x5 - Zero Buffer Length 0x6 - Packet Length Error (sum of buffers is less than packet length) 0x7 - 1xF - reserved	R	0x0
19	RESERVED		R	0x0
18:16	TX_ERR_CH	TX Host Error Channel - This field indicates which TX channel (if applicable) the host error occurred on. This field is cleared to zero on a host read.	R	0x0
15:12	RX_HOST_ERR_CODE	RX Host Error Code - This field is set to indicate CPDMA detected RX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. 0x0 - No error 0x1 - reserved 0x2 - Ownership bit not set in input buffer. 0x3 - reserved 0x4 - Zero Buffer Pointer. 0x5 - Zero buffer length on non-SOP descriptor 0x6 - SOP buffer length not greater than offset 0x7 - 1xF - reserved	R	0x0
11	RESERVED		R	0x0
10:8	RX_ERR_CH	RX Host Error Channel - This field indicates which RX channel the host error occurred on. This field is cleared to zero on a host read.	R	0x0
7:0	RESERVED		R	0x0

Table 24-1068. Register Call Summary for Register CPDMA_DMASTATUS

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1069. CPDMA_RX_BUFFER_OFFSET

Address Offset	0x0000 0028		
Physical Address	0x4848 4828	**Instance**	CPDMA
Description	CPDMA_REGS receive buffer offset		
Type	RW		

31–16	15–0
RESERVED	RX_BUFFER_OFFSET

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_BUFFER_OFFSET	Receive Buffer Offset Value - The RX_BUFFER_OFFSET will be written by the port into each frame SOP buffer descriptor buffer_offset field. The frame data will begin after the rx_buffer_offset value of bytes. A value of 0x0000 indicates that there are no unused bytes at the beginning of the data and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. This value is used for all channels.	RW	0x0

Table 24-1070. Register Call Summary for Register CPDMA_RX_BUFFER_OFFSET

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- RX Buffer Descriptors: [1]
- CPDMA Register Summary: [2]

Table 24-1071. CPDMA_EMCONTROL

Address Offset	0x0000 002C		
Physical Address	0x4848 482C	**Instance**	CPDMA
Description	CPDMA_REGS emulation control		
Type	RW		

31–2	1	0
RESERVED	SOFT	FREE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	SOFT	Emulation Soft Bit	RW	0x0
0	FREE	Emulation Free Bit	RW	0x0

Table 24-1072. Register Call Summary for Register CPDMA_EMCONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Emulation Control: [0]
- CPDMA Register Summary: [1]

Table 24-1073. CPDMA_TX_PRI0_RATE

Address Offset	0x0000 0030		
Physical Address	0x4848 4830	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 0 rate		

Table 24-1073. CPDMA_TX_PRI0_RATE (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1074. Register Call Summary for Register CPDMA_TX_PRI0_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1075. CPDMA_TX_PRI1_RATE

Address Offset	0x0000 0034		
Physical Address	0x4848 4834	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 1 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1076. Register Call Summary for Register CPDMA_TX_PRI1_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1077. CPDMA_TX_PRI2_RATE

Address Offset	0x0000 0038		
Physical Address	0x4848 4838	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 2 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1078. Register Call Summary for Register CPDMA_TX_PRI2_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1079. CPDMA_TX_PRI3_RATE

Address Offset	0x0000 003C		
Physical Address	0x4848 483C	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 3 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1080. Register Call Summary for Register CPDMA_TX_PRI3_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1081. CPDMA_TX_PRI4_RATE

Address Offset	0x0000 0040		
Physical Address	0x4848 4840	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 4 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1082. Register Call Summary for Register CPDMA_TX_PRI4_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1083. CPDMA_TX_PRI5_RATE

Address Offset	0x0000 0044		
Physical Address	0x4848 4844	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 5 rate		

Table 24-1083. CPDMA_TX_PRI5_RATE (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1084. Register Call Summary for Register CPDMA_TX_PRI5_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1085. CPDMA_TX_PRI6_RATE

Address Offset	0x0000 0048		
Physical Address	0x4848 4848	**Instance**	CPDMA
Description	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 6 RATE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1086. Register Call Summary for Register CPDMA_TX_PRI6_RATE

Gigabit Ethernet Switch (GMAC_SW)

- Audio Video Bridging: [0]
- CPDMA Register Summary: [1]

Table 24-1087. CPDMA_TX_PRI7_RATE

Address Offset	0x0000 004C		
Physical Address	0x4848 484C	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 7 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRIN_IDLE_CNT														RESERVED		PRIN_SEND_CNT													

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 24-1088. Register Call Summary for Register CPDMA_TX_PRI7_RATE

Gigabit Ethernet Switch (GMAC_SW)

- Audio Video Bridging: [0]
- CPDMA Register Summary: [1]

Table 24-1089. CPDMA_TX_INTSTAT_RAW

Address Offset	0x0000 0080		
Physical Address	0x4848 4880	**Instance**	CPDMA
Description	CPDMA_INT TX interrupt status register (raw value)		
Type	R		

31–8	7	6	5	4	3	2	1	0
RESERVED	TX7_PEND	TX6_PEND	TX5_PEND	TX4_PEND	TX3_PEND	TX2_PEND	TX1_PEND	TX0_PEND

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	TX7_PEND	TX7_PEND raw int read (before mask).	R	0x0
6	TX6_PEND	TX6_PEND raw int read (before mask).	R	0x0
5	TX5_PEND	TX5_PEND raw int read (before mask).	R	0x0
4	TX4_PEND	TX4_PEND raw int read (before mask).	R	0x0
3	TX3_PEND	TX3_PEND raw int read (before mask).	R	0x0
2	TX2_PEND	TX2_PEND raw int read (before mask).	R	0x0
1	TX1_PEND	TX1_PEND raw int read (before mask).	R	0x0
0	TX0_PEND	TX0_PEND raw int read (before mask).	R	0x0

Table 24-1090. Register Call Summary for Register CPDMA_TX_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- Miscellaneous Pulse Interrupt (MISC_PULSE): [1] [2]
- CPDMA Register Summary: [3]

Table 24-1091. CPDMA_TX_INTSTAT_MASKED

Address Offset	0x0000 0084		
Physical Address	0x4848 4884	**Instance**	CPDMA
Description	CPDMA_INT TX interrupt status register (masked value)		
Type	R		

31–8	7	6	5	4	3	2	1	0
RESERVED	TX7_PEND	TX6_PEND	TX5_PEND	TX4_PEND	TX3_PEND	TX2_PEND	TX1_PEND	TX0_PEND

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	TX7_PEND	TX7_PEND masked interrupt read.	R	0x0
6	TX6_PEND	TX6_PEND masked interrupt read.	R	0x0
5	TX5_PEND	TX5_PEND masked interrupt read.	R	0x0
4	TX4_PEND	TX4_PEND masked interrupt read.	R	0x0
3	TX3_PEND	TX3_PEND masked interrupt read.	R	0x0
2	TX2_PEND	TX2_PEND masked interrupt read.	R	0x0
1	TX1_PEND	TX1_PEND masked interrupt read.	R	0x0
0	TX0_PEND	TX0_PEND masked interrupt read.	R	0x0

Table 24-1092. Register Call Summary for Register CPDMA_TX_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- Miscellaneous Pulse Interrupt (MISC_PULSE): [1] [2]
- CPDMA Register Summary: [3]

Table 24-1093. CPDMA_TX_INTMASK_SET

Address Offset	0x0000 0088		
Physical Address	0x4848 4888	**Instance**	CPDMA
Description	CPDMA_INT TX interrupt mask set register		
Type	RW		

31–8	7	6	5	4	3	2	1	0
RESERVED	TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	TX7_MASK	TX Channel 7 Mask - Write one to enable interrupt.	RW	0x0
6	TX6_MASK	TX Channel 6 Mask - Write one to enable interrupt.	RW	0x0
5	TX5_MASK	TX Channel 5 Mask - Write one to enable interrupt.	RW	0x0
4	TX4_MASK	TX Channel 4 Mask - Write one to enable interrupt.	RW	0x0
3	TX3_MASK	TX Channel 3 Mask - Write one to enable interrupt.	RW	0x0
2	TX2_MASK	TX Channel 2 Mask - Write one to enable interrupt.	RW	0x0
1	TX1_MASK	TX Channel 1 Mask - Write one to enable interrupt.	RW	0x0
0	TX0_MASK	TX Channel 0 Mask - Write one to enable interrupt.	RW	0x0

Table 24-1094. Register Call Summary for Register CPDMA_TX_INTMASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- CPDMA RX and TX Interfaces: [1]
- CPDMA Register Summary: [2]

Table 24-1095. CPDMA_TX_INTMASK_CLEAR

Address Offset	0x0000 008C		
Physical Address	0x4848 488C	**Instance**	CPDMA
Description	CPDMA_INT TX Interrupt mask clear register		

Table 24-1095. CPDMA_TX_INTMASK_CLEAR (continued)

Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	TX7_MASK	TX Channel 7 Mask - Write one to disable interrupt.	RW	0x0
6	TX6_MASK	TX Channel 6 Mask - Write one to disable interrupt.	RW	0x0
5	TX5_MASK	TX Channel 5 Mask - Write one to disable interrupt.	RW	0x0
4	TX4_MASK	TX Channel 4 Mask - Write one to disable interrupt.	RW	0x0
3	TX3_MASK	TX Channel 3 Mask - Write one to disable interrupt.	RW	0x0
2	TX2_MASK	TX Channel 2 Mask - Write one to disable interrupt.	RW	0x0
1	TX1_MASK	TX Channel 1 Mask - Write one to disable interrupt.	RW	0x0
0	TX0_MASK	TX Channel 0 Mask - Write one to disable interrupt.	RW	0x0

Table 24-1096. Register Call Summary for Register CPDMA_TX_INTMASK_CLEAR

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- CPDMA Register Summary: [1]

Table 24-1097. CPDMA_IN_VECTOR

Address Offset	0x0000 0090		
Physical Address	0x4848 4890	**Instance**	CPDMA
Description	CPDMA_INT input vector (read only)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_IN_VECTOR																															

Bits	Field Name	Description	Type	Reset
31:0	DMA_IN_VECTOR	DMA Input Vector - The value of DMA_IN_VECTOR is reset to zero, but will change to the IN_VECTOR bus value one clock after reset is deasserted. Thereafter, this value will change to a new IN_VECTOR value one clock after the IN_VECTOR value changes.	R	0x0

Table 24-1098. Register Call Summary for Register CPDMA_IN_VECTOR

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1099. CPDMA_EOI_VECTOR

Address Offset	0x0000 0094		
Physical Address	0x4848 4894	**Instance**	CPDMA
Description	CPDMA_INT end of interrupt vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	DMA_EOI_VECTOR

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	DMA_EOI_VECTOR	DMA End of Interrupt Vector - The EOI_VECTOR(4:0) pins reflect the value written to this location one MAIN_CLK cycle after a write to this location. The EOI_WR signal is asserted for a single clock cycle after a latency of two MAIN_CLK cycles when a write is performed to this location.	RW	0x0

Table 24-1100. Register Call Summary for Register CPDMA_EOI_VECTOR

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [1]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [2]
- Miscellaneous Pulse Interrupt (MISC_PULSE): [3]
- CPDMA Register Summary: [4]

Table 24-1101. CPDMA_RX_INTSTAT_RAW

Address Offset	0x0000 00A0		
Physical Address	0x4848 48A0	**Instance**	CPDMA
Description	CPDMA_INT RX Interrupt status register (raw value)		
Type	R		

31:16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND	RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND	RX7_THRESH_PEND raw int read (before mask).	R	0x0
14	RX6_THRESH_PEND	RX6_THRESH_PEND raw int read (before mask).	R	0x0
13	RX5_THRESH_PEND	RX5_THRESH_PEND raw int read (before mask).	R	0x0
12	RX4_THRESH_PEND	RX4_THRESH_PEND raw int read (before mask).	R	0x0
11	RX3_THRESH_PEND	RX3_THRESH_PEND raw int read (before mask).	R	0x0
10	RX2_THRESH_PEND	RX2_THRESH_PEND raw int read (before mask).	R	0x0
9	RX1_THRESH_PEND	RX1_THRESH_PEND raw int read (before mask).	R	0x0
8	RX0_THRESH_PEND	RX0_THRESH_PEND raw int read (before mask).	R	0x0
7	RX7_PEND	RX7_PEND raw int read (before mask).	R	0x0
6	RX6_PEND	RX6_PEND raw int read (before mask).	R	0x0
5	RX5_PEND	RX5_PEND raw int read (before mask).	R	0x0
4	RX4_PEND	RX4_PEND raw int read (before mask).	R	0x0
3	RX3_PEND	RX3_PEND raw int read (before mask).	R	0x0
2	RX2_PEND	RX2_PEND raw int read (before mask).	R	0x0
1	RX1_PEND	RX1_PEND raw int read (before mask).	R	0x0

Bits	Field Name	Description	Type	Reset
0	RX0_PEND	RX0_PEND raw int read (before mask).	R	0x0

Table 24-1102. Register Call Summary for Register CPDMA_RX_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA Register Summary: [2]

Table 24-1103. CPDMA_RX_INTSTAT_MASKED

Address Offset	0x0000 00A4		
Physical Address	0x4848 48A4	**Instance**	CPDMA
Description	CPDMA_INT RX interrupt status register (masked value)		
Type	R		

31:16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND	RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND	RX7_THRESH_PEND masked int read.	R	0x0
14	RX6_THRESH_PEND	RX6_THRESH_PEND masked int read.	R	0x0
13	RX5_THRESH_PEND	RX5_THRESH_PEND masked int read.	R	0x0
12	RX4_THRESH_PEND	RX4_THRESH_PEND masked int read.	R	0x0
11	RX3_THRESH_PEND	RX3_THRESH_PEND masked int read.	R	0x0
10	RX2_THRESH_PEND	RX2_THRESH_PEND masked int read.	R	0x0
9	RX1_THRESH_PEND	RX1_THRESH_PEND masked int read.	R	0x0
8	RX0_THRESH_PEND	RX0_THRESH_PEND masked int read.	R	0x0
7	RX7_PEND	RX7_PEND masked int read.	R	0x0
6	RX6_PEND	RX6_PEND masked int read.	R	0x0
5	RX5_PEND	RX5_PEND masked int read.	R	0x0
4	RX4_PEND	RX4_PEND masked int read.	R	0x0
3	RX3_PEND	RX3_PEND masked int read.	R	0x0
2	RX2_PEND	RX2_PEND masked int read.	R	0x0
1	RX1_PEND	RX1_PEND masked int read.	R	0x0
0	RX0_PEND	RX0_PEND masked int read.	R	0x0

Table 24-1104. Register Call Summary for Register CPDMA_RX_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA Register Summary: [2]

Table 24-1105. CPDMA_RX_INTMASK_SET

Address Offset	0x0000 00A8

Table 24-1105. CPDMA_RX_INTMASK_SET (continued)

Physical Address	0x4848 48A8	**Instance**	CPDMA
Description	CPDMA_INT RX interrupt mask set register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK	RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND_MASK	RX Channel 7 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
14	RX6_THRESH_PEND_MASK	RX Channel 6 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
13	RX5_THRESH_PEND_MASK	RX Channel 5 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
12	RX4_THRESH_PEND_MASK	RX Channel 4 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
11	RX3_THRESH_PEND_MASK	RX Channel 3 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
10	RX2_THRESH_PEND_MASK	RX Channel 2 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
9	RX1_THRESH_PEND_MASK	RX Channel 1 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
8	RX0_THRESH_PEND_MASK	RX Channel 0 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
7	RX7_PEND_MASK	RX Channel 7 Pending Int. Mask - Write one to enable Int.	RW	0x0
6	RX6_PEND_MASK	RX Channel 6 Pending Int. Mask - Write one to enable Int.	RW	0x0
5	RX5_PEND_MASK	RX Channel 5 Pending Int. Mask - Write one to enable Int.	RW	0x0
4	RX4_PEND_MASK	RX Channel 4 Pending Int. Mask - Write one to enable Int.	RW	0x0
3	RX3_PEND_MASK	RX Channel 3 Pending Int. Mask - Write one to enable Int.	RW	0x0
2	RX2_PEND_MASK	RX Channel 2 Pending Int. Mask - Write one to enable Int.	RW	0x0
1	RX1_PEND_MASK	RX Channel 1 Pending Int. Mask - Write one to enable Int.	RW	0x0
0	RX0_PEND_MASK	RX Channel 0 Pending Int. Mask - Write one to enable Int.	RW	0x0

Table 24-1106. Register Call Summary for Register CPDMA_RX_INTMASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA RX and TX Interfaces: [2]
- CPDMA Register Summary: [3]

Table 24-1107. CPDMA_RX_INTMASK_CLEAR

Address Offset	0x0000 00AC		
Physical Address	0x4848 48AC	**Instance**	CPDMA
Description	CPDMA_INT RX interrupt mask clear register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK	RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND_MASK	RX Channel 7 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
14	RX6_THRESH_PEND_MASK	RX Channel 6 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
13	RX5_THRESH_PEND_MASK	RX Channel 5 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
12	RX4_THRESH_PEND_MASK	RX Channel 4 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
11	RX3_THRESH_PEND_MASK	RX Channel 3 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
10	RX2_THRESH_PEND_MASK	RX Channel 2 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
9	RX1_THRESH_PEND_MASK	RX Channel 1 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
8	RX0_THRESH_PEND_MASK	RX Channel 0 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
7	RX7_PEND_MASK	RX Channel 7 Pending Int. Mask - Write one to disable Int.	RW	0x0
6	RX6_PEND_MASK	RX Channel 6 Pending Int. Mask - Write one to disable Int.	RW	0x0
5	RX5_PEND_MASK	RX Channel 5 Pending Int. Mask - Write one to disable Int.	RW	0x0
4	RX4_PEND_MASK	RX Channel 4 Pending Int. Mask - Write one to disable Int.	RW	0x0
3	RX3_PEND_MASK	RX Channel 3 Pending Int. Mask - Write one to disable Int.	RW	0x0
2	RX2_PEND_MASK	RX Channel 2 Pending Int. Mask - Write one to disable Int.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	RX1_PEND_MASK	RX Channel 1 Pending Int. Mask - Write one to disable Int.	RW	0x0
0	RX0_PEND_MASK	RX Channel 0 Pending Int. Mask - Write one to disable Int.	RW	0x0

Table 24-1108. Register Call Summary for Register CPDMA_RX_INTMASK_CLEAR

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA Register Summary: [2]

Table 24-1109. CPDMA_DMA_INTSTAT_RAW

Address Offset	0x0000 00B0		
Physical Address	0x4848 48B0	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt status register (raw value)		
Type	R		

31–2	1	0
RESERVED	HOST_PEND	STAT_PEND

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_PEND	Host Pending Interrupt - raw int read (before mask).	R	0x0
0	STAT_PEND	Statistics Pending Interrupt - raw int read (before mask).	R	0x0

Table 24-1110. Register Call Summary for Register CPDMA_DMA_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1111. CPDMA_DMA_INTSTAT_MASKED

Address Offset	0x0000 00B4		
Physical Address	0x4848 48B4	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt status register (masked value)		
Type	R		

31–2	1	0
RESERVED	HOST_PEND	STAT_PEND

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_PEND	Host Pending Interrupt - masked interrupt read.	R	0x0
0	STAT_PEND	Statistics Pending Interrupt - masked interrupt read.	R	0x0

Table 24-1112. Register Call Summary for Register CPDMA_DMA_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1113. CPDMA_DMA_INTMASK_SET

Address Offset	0x0000 00B8		
Physical Address	0x4848 48B8	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt mask set register		
Type	W		

31:2	1	0
RESERVED	HOST_ERR_INT_MASK	STAT_INT_MASK

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_ERR_INT_MASK	Host Error Interrupt Mask - Write one to enable interrupt.	W	0x0
0	STAT_INT_MASK	Statistics Interrupt Mask - Write one to enable interrupt.	R	0x0

Table 24-1114. Register Call Summary for Register CPDMA_DMA_INTMASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0] [1]
- CPDMA Register Summary: [2]

Table 24-1115. CPDMA_DMA_INTMASK_CLEAR

Address Offset	0x0000 00BC		
Physical Address	0x4848 48BC	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt mask clear register		
Type	RW		

31:2	1	0
RESERVED	HOST_ERR_INT_MASK	STAT_INT_MASK

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_ERR_INT_MASK	Host Error Interrupt Mask - Write one to disable interrupt.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	STAT_INT_MASK	Statistics Interrupt Mask - Write one to disable interrupt.	RW	0x0

Table 24-1116. Register Call Summary for Register CPDMA_DMA_INTMASK_CLEAR

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0]
- CPDMA Register Summary: [1]

Table 24-1117. CPDMA_RX0_PENDTHRESH

Address Offset	0x0000 00C0		
Physical Address	0x4848 48C0	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 0		
Type	RW		

31–8	7–0
RESERVED	RX_PENDTHRESH

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1118. Register Call Summary for Register CPDMA_RX0_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1119. CPDMA_RX1_PENDTHRESH

Address Offset	0x0000 00C4		
Physical Address	0x4848 48C4	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 1		
Type	RW		

31–8	7–0
RESERVED	RX_PENDTHRESH

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1120. Register Call Summary for Register CPDMA_RX1_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1121. CPDMA_RX2_PENDTHRESH

Address Offset	0x0000 00C8		
Physical Address	0x4848 48C8	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 2		

Table 24-1121. CPDMA_RX2_PENDTHRESH (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1122. Register Call Summary for Register CPDMA_RX2_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1123. CPDMA_RX3_PENDTHRESH

Address Offset	0x0000 00CC		
Physical Address	0x4848 48CC	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1124. Register Call Summary for Register CPDMA_RX3_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1125. CPDMA_RX4_PENDTHRESH

Address Offset	0x0000 00D0		
Physical Address	0x4848 48D0	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1126. Register Call Summary for Register CPDMA_RX4_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1127. CPDMA_RX5_PENDTHRESH

Address Offset	0x0000 00D4		
Physical Address	0x4848 48D4	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1128. Register Call Summary for Register CPDMA_RX5_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1129. CPDMA_RX6_PENDTHRESH

Address Offset	0x0000 00D8		
Physical Address	0x4848 48D8	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1130. Register Call Summary for Register CPDMA_RX6_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1131. CPDMA_RX7_PENDTHRESH

Address Offset	0x0000 00DC		
Physical Address	0x4848 48DC	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RX_PENDTHRESH

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 24-1132. Register Call Summary for Register CPDMA_RX7_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 24-1133. CPDMA_RX0_FREEBUFFER

Address Offset	0x0000 00E0		
Physical Address	0x4848 48E0	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 0		
Type	W		

31:16	15:0
RESERVED	RX_FREEBUFFER

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX0_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1134. Register Call Summary for Register CPDMA_RX0_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1135. CPDMA_RX1_FREEBUFFER

Address Offset	0x0000 00E4		
Physical Address	0x4848 48E4	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 1		
Type	W		

31:16	15:0
RESERVED	RX_FREEBUFFER

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX1_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1136. Register Call Summary for Register CPDMA_RX1_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1137. CPDMA_RX2_FREEBUFFER

Address Offset	0x0000 00E8		
Physical Address	0x4848 48E8	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 2		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX2_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1138. Register Call Summary for Register CPDMA_RX2_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1139. CPDMA_RX3_FREEBUFFER

Address Offset	0x0000 00EC		
Physical Address	0x4848 48EC	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 3		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	RX_FREEBUFFER

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX3_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1140. Register Call Summary for Register CPDMA_RX3_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1141. CPDMA_RX4_FREEBUFFER

Address Offset	0x0000 00F0		
Physical Address	0x4848 48F0	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 4		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX4_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1142. Register Call Summary for Register CPDMA_RX4_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1143. CPDMA_RX5_FREEBUFFER

Address Offset	0x0000 00F4		
Physical Address	0x4848 48F4	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 5		

Table 24-1143. CPDMA_RX5_FREEBUFFER (continued)

Type	W

31–16	15–0
RESERVED	RX_FREEBUFFER

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX5_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1144. Register Call Summary for Register CPDMA_RX5_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1145. CPDMA_RX6_FREEBUFFER

Address Offset	0x0000 00F8		
Physical Address	0x4848 48F8	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 6		
Type	W		

31–16	15–0
RESERVED	RX_FREEBUFFER

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX6_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1146. Register Call Summary for Register CPDMA_RX6_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 24-1147. CPDMA_RX7_FREEBUFFER

Address Offset	0x0000 00FC		
Physical Address	0x4848 48FC	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 7		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX7_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 24-1148. Register Call Summary for Register CPDMA_RX7_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

24.11.6.5 STATS Registers

24.11.6.5.1 STATS Register Summary

Table 24-1149. STATS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	STATS Physical Address
GOOD_RX_FRAMES	RW	32	0x0000 0000	0x4848 4900
BROADCAST_RX_FRAMES	RW	32	0x0000 0004	0x4848 4904
MULTICAST_RX_FRAMES	RW	32	0x0000 0008	0x4848 4908
PAUSE_RX_FRAMES	RW	32	0x0000 000C	0x4848 490C
RX_CRC_ERRORS	RW	32	0x0000 0010	0x4848 4910
RX_ALIGN_CODE_ERRORS	RW	32	0x0000 0014	0x4848 4914
OVERSIZE_RX_FRAMES	RW	32	0x0000 0018	0x4848 4918
RX_JABBERS	RW	32	0x0000 001C	0x4848 491C
UNDERSIZE_RX_FRAMES	RW	32	0x0000 0020	0x4848 4920
RX_FRAGMENTS	RW	32	0x0000 0024	0x4848 4924
RX_OCTETS	RW	32	0x0000 0030	0x4848 4930
GOOD_TX_FRAMES	RW	32	0x0000 0034	0x4848 4934
BROADCAST_TX_FRAMES	RW	32	0x0000 0038	0x4848 4938
MULTICAST_TX_FRAMES	RW	32	0x0000 003C	0x4848 493C
PAUSE_TX_FRAMES	RW	32	0x0000 0040	0x4848 4940
DEFERRED_TX_FRAMES	RW	32	0x0000 0044	0x4848 4944
COLLISIONS	RW	32	0x0000 0048	0x4848 4948
SINGLE_COLLISION_TX_FRAMES	RW	32	0x0000 004C	0x4848 494C

Table 24-1149. STATS Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	STATS Physical Address
MULTIPLE_COLLISION_TX_FRAMES	RW	32	0x0000 0050	0x4848 4950
EXCESSIVE_COLLISIONS	RW	32	0x0000 0054	0x4848 4954
LATE_COLLISIONS	RW	32	0x0000 0058	0x4848 4958
TX_UNDERRUN	RW	32	0x0000 005C	0x4848 495C
CARRIER_SENSE_ERRORS	RW	32	0x0000 0060	0x4848 4960
TX_OCTETS	RW	32	0x0000 0064	0x4848 4964
RX_TX_64_OCTET_FRAMES	RW	32	0x0000 0068	0x4848 4968
RX_TX_65_127_OCTET_FRAMES	RW	32	0x0000 006C	0x4848 496C
RX_TX_128_255_OCTET_FRAMES	RW	32	0x0000 0070	0x4848 4970
RX_TX_256_511_OCTET_FRAMES	RW	32	0x0000 0074	0x4848 4974
RX_TX_512_1023_OCTET_FRAMES	RW	32	0x0000 0078	0x4848 4978
RX_TX_1024_UP_OCTET_FRAMES	RW	32	0x0000 007C	0x4848 497C
NET_OCTETS	RW	32	0x0000 0080	0x4848 4980
RX_START_OF_FRAME_OVERRUNS	RW	32	0x0000 0084	0x4848 4984
RX_MIDDLE_OF_FRAME_OVERRUNS	RW	32	0x0000 0088	0x4848 4988
RX_DMA_OVERRUNS	RW	32	0x0000 008C	0x4848 498C

24.11.6.5.2 STATS Register Description

Table 24-1150. GOOD_RX_FRAMES

Address Offset	0x0000 0000		
Physical Address	0x4848 4900	**Instance**	STATS
Description	The total number of good frames received on the port. A good frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Had a length of 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error. See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1151. Register Call Summary for Register GOOD_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1152. BROADCAST_RX_FRAMES

Address Offset	0x0000 0004		
Physical Address	0x4848 4904	**Instance**	STATS
Description	The total number of good broadcast frames received on the port. A good broadcast frame is defined to be: - Any data or MAC control frame which was destined for only address 0xFFFFFFFFFFFF - Had a length of SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error. See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		

Table 24-1152. BROADCAST_RX_FRAMES (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1153. Register Call Summary for Register BROADCAST_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)
- STATS Register Summary: [0]

Table 24-1154. MULTICAST_RX_FRAMES

Address Offset 0x0000 0008

Physical Address 0x4848 4908 **Instance** STATS

Description The total number of good multicast frames received on the port. A good multicast frame is defined to be:
- Any data or MAC control frame which was destined for any multicast address other than 0xFFFFFFFFFFFF
- Had a length of SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive
- Had no CRC error, alignment error or code error.
See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1155. Register Call Summary for Register MULTICAST_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)
- STATS Register Summary: [0]

Table 24-1156. PAUSE_RX_FRAMES

Address Offset 0x0000 000C

Physical Address 0x4848 490C **Instance** STATS

Description The total number of IEEE 802.3X pause frames received by the port (whether acted upon or not). Such a frame:
- Contained any unicast, broadcast, or multicast address
- Contained the length/type field value 88.08 (hex) and the opcode 0x0001
- Was of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive
- Had no CRC error, alignment error or code error
- Pause-frames had been enabled on the port (SL_MACCONTROL[4] TX_FLOW_EN = 1). The port could have been in either half or full-duplex mode.
See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1157. Register Call Summary for Register PAUSE_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1158. RX_CRC_ERRORS

Address Offset	0x0000 0010		
Physical Address	0x4848 4910	**Instance**	STATS
Description	The total number of frames received on the port that experienced a CRC error. Such a frame: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no code/align error, - Had a CRC error Overruns have no effect upon this statistic. A CRC error is defined to be: - A frame containing an even number of nibbles - Failing the Frame Check Sequence test		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1159. Register Call Summary for Register RX_CRC_ERRORS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]
- STATS Register Description: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]

Table 24-1160. RX_ALIGN_CODE_ERRORS

Address Offset	0x0000 0014		
Physical Address	0x4848 4914	**Instance**	STATS
Description	The total number of frames received on the port that experienced an alignment error or code error. Such a frame: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had either an alignment error or a code error Overruns have no effect upon this statistic. An alignment error is defined to be: - A frame containing an odd number of nibbles - Failing the Frame Check Sequence test if the final nibble is ignored A code error is defined to be a frame which has been discarded because the port's MRXER pin driven with a one for at least one bit-time's duration at any point during the frame's reception. Note: RFC 1757 etherStatsCRCAlignErrors Ref. 1.5 can be calculated by summing RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1161. Register Call Summary for Register RX_ALIGN_CODE_ERRORS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]
- STATS Register Description: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]

Table 24-1162. OVERSIZE_RX_FRAMES

Address Offset	0x0000 0018		
Physical Address	0x4848 4918	**Instance**	STATS
Description	The total number of oversized frames received on the port. An oversized frame is defined to be: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was greater than SL_RX_MAXLEN[13:0] RX_MAXLEN in bytes - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1163. Register Call Summary for Register OVERSIZE_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1164. RX_JABBERS

Address Offset	0x0000 001C		
Physical Address	0x4848 491C	**Instance**	STATS
Description	The total number of jabber frames received on the port. A jabber frame: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was greater than SL_RX_MAXLEN[13:0] RX_MAXLEN in bytes - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1165. Register Call Summary for Register RX_JABBERS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1166. UNDERSIZE_RX_FRAMES

Address Offset	0x0000 0020		
Physical Address	0x4848 4920	**Instance**	STATS
Description	The total number of undersized frames received on the port. An undersized frame is defined to be: - Was any data frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was less than 64 octets long - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1167. Register Call Summary for Register UNDERSIZE_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1168. RX_FRAGMENTS

Address Offset	0x0000 0024		
Physical Address	0x4848 4924	**Instance**	STATS
Description	The total number of frame fragments received on the port. A frame fragment is defined to be: - Any data frame (address matching does not matter) - Less than 64 bytes long - Having a CRC error, an alignment error, or a code error - Not the result of a collision caused by half duplex, collision based flow control See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1169. Register Call Summary for Register RX_FRAGMENTS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1170. RX_OCTETS

Address Offset	0x0000 0030		
Physical Address	0x4848 4930	**Instance**	STATS
Description	The total number of bytes in all good frames received on the port. A good frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		

Table 24-1170. RX_OCTETS (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1171. Register Call Summary for Register RX_OCTETS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1172. GOOD_TX_FRAMES

Address Offset	0x0000 0034		
Physical Address	0x4848 4934	**Instance**	STATS
Description	The total number of good frames received on the port. A good frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length - Had no late or excessive collisions, no carrier loss and no underrun		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1173. Register Call Summary for Register GOOD_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1174. BROADCAST_TX_FRAMES

Address Offset	0x0000 0038		
Physical Address	0x4848 4938	**Instance**	STATS
Description	The total number of good broadcast frames received on the port. A good broadcast frame is defined to be: - Any data or MAC control frame which was destined for only address 0xFFFFFFFFFFFF - Any length - Had no late or excessive collisions, no carrier loss and no underrun		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1175. Register Call Summary for Register BROADCAST_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1176. MULTICAST_TX_FRAMES

Address Offset	0x0000 003C		
Physical Address	0x4848 493C	**Instance**	STATS
Description	The total number of good multicast frames received on the port. A good multicast frame is defined to be: - Any data or MAC control frame which was destined for any multicast address other than 0xFFFFFFFFFFFF - Any length - Had no late or excessive collisions, no carrier loss and no underrun		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1177. Register Call Summary for Register MULTICAST_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)
- STATS Register Summary: [0]
- STATS Register Description: [1]

Table 24-1178. PAUSE_TX_FRAMES

Address Offset	0x0000 0040		
Physical Address	0x4848 4940	**Instance**	STATS
Description	This statistic indicates the number of IEEE 802.3X pause frames transmitted by the port. Pause frames cannot underrun or contain a CRC error because they are created in the transmitting MAC, so these error conditions have no effect upon the statistic. Pause frames sent by software will not be included in this count. Since pause frames are only transmitted in full duplex carrier loss and collisions have no effect upon this statistic. Transmitted pause frames are always 64 byte multicast frames so will appear in the MULTICAST_TX_FRAMES and RX_TX_64_OCTET_FRAMES statistics.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1179. Register Call Summary for Register PAUSE_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)
- STATS Register Summary: [0]

Table 24-1180. DEFERRED_TX_FRAMES

Address Offset	0x0000 0044		
Physical Address	0x4848 4944	**Instance**	STATS
Description	The total number of frames transmitted on the port that first experienced deferment. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced no collisions before being successfully transmitted - Found the medium busy when transmission was first attempted, so had to wait. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

VALUE

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1181. Register Call Summary for Register DEFERRED_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1182. COLLISIONS

Address Offset	0x0000 0048		
Physical Address	0x4848 4948	**Instance**	STATS
Description	This statistic records the total number of times that the port experienced a collision. Collisions occur under two circumstances. 1. When a transmit data or MAC control frame: - Was destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced a collision. A jam sequence is sent for every non-late collision, so this statistic will increment on each occasion if a frame experiences multiple collisions (and increments on late collisions) CRC errors have no effect upon this statistic. 2. When the port is in half-duplex mode, flow control is active, and a frame reception begins.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1183. Register Call Summary for Register COLLISIONS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1184. SINGLE_COLLISION_TX_FRAMES

Address Offset	0x0000 004C		
Physical Address	0x4848 494C	**Instance**	STATS
Description	The total number of frames transmitted on the port that experienced exactly one collision. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced one collision before successful transmission. The collision was not late. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1185. Register Call Summary for Register SINGLE_COLLISION_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1186. MULTIPLE_COLLISION_TX_FRAMES

Address Offset	0x0000 0050		
Physical Address	0x4848 4950	**Instance**	STATS
Description	The total number of frames transmitted on the port that experienced multiple collisions. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced 2 to 15 collisions before being successfully transmitted. None of the collisions were late. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1187. Register Call Summary for Register MULTIPLE_COLLISION_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1188. EXCESSIVE_COLLISIONS

Address Offset	0x0000 0054		
Physical Address	0x4848 4954	**Instance**	STATS
Description	The total number of frames for which transmission was abandoned due to excessive collisions. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced 16 collisions before abandoning all attempts at transmitting the frame. None of the collisions were late. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1189. Register Call Summary for Register EXCESSIVE_COLLISIONS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1190. LATE_COLLISIONS

Address Offset	0x0000 0058		
Physical Address	0x4848 4958	**Instance**	STATS
Description	The total number of frames on the port for which transmission was abandoned because they experienced a late collision. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Experienced a collision later than 512 bit-times into the transmission. There may have been up to 15 previous (non-late) collisions which had previously required the transmission to be re-attempted. The Late Collisions statistic dominates over the single, multiple and excessive Collisions statistics - if a late collision occurs the frame will not be counted in any of these other three statistics. CRC errors have no effect upon this statistic.		

Table 24-1190. LATE_COLLISIONS (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1191. Register Call Summary for Register LATE_COLLISIONS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1192. TX_UNDERRUN

Address Offset	0x0000 005C		
Physical Address	0x4848 495C	**Instance**	STATS
Description	There should be no transmitted frames that experience underrun.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1193. Register Call Summary for Register TX_UNDERRUN

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1194. CARRIER_SENSE_ERRORS

Address Offset	0x0000 0060		
Physical Address	0x4848 4960	**Instance**	STATS
Description	The total number of frames received on the port that had a CPDMA middle of frame (MOF) overrun. MOF overrun frame is defined to be: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - The carrier sense condition was lost or never asserted when transmitting the frame (the frame is not retransmitted). This is a transmit only statistic. Carrier Sense is a don't care for received frames. Transmit frames with carrier sense errors are sent until completion and are not aborted. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1195. Register Call Summary for Register CARRIER_SENSE_ERRORS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1196. TX_OCTETS

Address Offset	0x0000 0064

Table 24-1196. TX_OCTETS (continued)

Physical Address	0x4848 4964	**Instance**	STATS
Description	The total number of bytes in all good frames transmitted on the port. A good frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Was any size - Had no late or excessive collisions, no carrier loss and no underrun.		
Type	RW		

31–0
VALUE

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1197. Register Call Summary for Register TX_OCTETS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1198. RX_TX_64_OCTET_FRAMES

Address Offset	0x0000 0068		
Physical Address	0x4848 4968	**Instance**	STATS
Description	The total number of 64-byte frames received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was exactly 64 bytes long. (If the frame was being transmitted and experienced carrier loss that resulted in a frame of this size being transmitted, then the frame will be recorded in this statistic). CRC errors, code/align errors and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31–0
VALUE

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1199. Register Call Summary for Register RX_TX_64_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]
- STATS Register Description: [1]

Table 24-1200. RX_TX_65_127_OCTET_FRAMES

Address Offset	0x0000 006C		
Physical Address	0x4848 496C	**Instance**	STATS
Description	The total number of frames of size 65 to 127 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 65 to 127 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31–0
VALUE

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1201. Register Call Summary for Register RX_TX_65_127_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1202. RX_TX_128_255_OCTET_FRAMES

Address Offset	0x0000 0070		
Physical Address	0x4848 4970	**Instance**	STATS
Description	The total number of frames of size 128 to 255 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 128 to 255 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1203. Register Call Summary for Register RX_TX_128_255_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1204. RX_TX_256_511_OCTET_FRAMES

Address Offset	0x0000 0074		
Physical Address	0x4848 4974	**Instance**	STATS
Description	The total number of frames of size 256 to 511 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 256 to 511 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1205. Register Call Summary for Register RX_TX_256_511_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1206. RX_TX_512_1023_OCTET_FRAMES

Address Offset	0x0000 0078		
Physical Address	0x4848 4978	**Instance**	STATS

Table 24-1206. RX_TX_512_1023_OCTET_FRAMES (continued)

Description	The total number of frames of size 512 to 1023 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 512 to 1023 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1207. Register Call Summary for Register RX_TX_512_1023_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1208. RX_TX_1024_UP_OCTET_FRAMES

Address Offset	0x0000 007C		
Physical Address	0x4848 497C	**Instance**	STATS
Description	The total number of frames of size 1024 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes for receive or 1024 up for transmit on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 1024 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes long on receive, or any size on transmit CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1209. Register Call Summary for Register RX_TX_1024_UP_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1210. NET_OCTETS

Address Offset	0x0000 0080		
Physical Address	0x4848 4980	**Instance**	STATS

Table 24-1210. NET_OCTETS (continued)

Description	The total number of bytes of frame data received and transmitted on the port. Each frame counted: - was any data or MAC control frame destined for any unicast, broadcast or multicast address (address match does not matter) - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) Also counted in this statistic is: - Every byte transmitted before a carrier- loss was experienced - Every byte transmitted before each collision was experienced, (i.e. multiple retries are counted each time) - Every byte received if the port is in half-duplex mode until a jam sequence was transmitted to initiate flow control. (The jam sequence was not counted to prevent double-counting) Error conditions such as alignment errors, CRC errors, code errors, overruns and underruns do not affect the recording of bytes by this statistic. The objective of this statistic is to give a reasonable indication of ethernet utilization
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1211. Register Call Summary for Register NET_OCTETS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1212. RX_START_OF_FRAME_OVERRUNS

Address Offset	0x0000 0084		
Physical Address	0x4848 4984	**Instance**	STATS
Description	The total number of frames received on the port that had a CPDMA start of frame (SOF) overrun or were dropped by due to FIFO resource limitations, or were dropped by the SPF. SOF overrun frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) - The CPDMA had a start of frame overrun or the packet was dropped due to FIFO resource limitations		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1213. Register Call Summary for Register RX_START_OF_FRAME_OVERRUNS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 24-1214. RX_MIDDLE_OF_FRAME_OVERRUNS

Address Offset	0x0000 0088		
Physical Address	0x4848 4988	**Instance**	STATS
Description	The total number of frames received on the port that had a CPDMA middle of frame (MOF) overrun. MOF overrun frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) - The CPDMA had a middle of frame overrun		

Table 24-1214. RX_MIDDLE_OF_FRAME_OVERRUNS (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1215. Register Call Summary for Register RX_MIDDLE_OF_FRAME_OVERRUNS

Gigabit Ethernet Switch (GMAC_SW)
- STATS Register Summary: [0]

Table 24-1216. RX_DMA_OVERRUNS

Address Offset	0x0000 008C		
Physical Address	0x4848 498C	**Instance**	STATS
Description	The total number of frames received on the port that had either a DMA start of frame (SOF) overrun or a DMA MOF overrun. An Rx DMA overrun frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) - The CPGMAC_SL was unable to receive it because it did not have the DMA buffer resources to receive it (zero head descriptor pointer at the start or during the middle of the frame reception) CRC errors, alignment errors and code errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 24-1217. Register Call Summary for Register RX_DMA_OVERRUNS

Gigabit Ethernet Switch (GMAC_SW)
- STATS Register Summary: [0]

24.11.6.6 STATERAM Registers

24.11.6.6.1 STATERAM Register Summary

Table 24-1218. STATERAM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	STATERAM Physical Address
TX0_HDP	RW	32	0x0000 0000	0x4848 4A00
TX1_HDP	RW	32	0x0000 0004	0x4848 4A04
TX2_HDP	RW	32	0x0000 0008	0x4848 4A08
TX3_HDP	RW	32	0x0000 000C	0x4848 4A0C
TX4_HDP	RW	32	0x0000 0010	0x4848 4A10
TX5_HDP	RW	32	0x0000 0014	0x4848 4A14
TX6_HDP	RW	32	0x0000 0018	0x4848 4A18
TX7_HDP	RW	32	0x0000 001C	0x4848 4A1C
RX0_HDP	RW	32	0x0000 0020	0x4848 4A20
RX1_HDP	RW	32	0x0000 0024	0x4848 4A24
RX2_HDP	RW	32	0x0000 0028	0x4848 4A28

Table 24-1218. STATERAM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	STATERAM Physical Address
RX3_HDP	RW	32	0x0000 002C	0x4848 4A2C
RX4_HDP	RW	32	0x0000 0030	0x4848 4A30
RX5_HDP	RW	32	0x0000 0034	0x4848 4A34
RX6_HDP	RW	32	0x0000 0038	0x4848 4A38
RX7_HDP	RW	32	0x0000 003C	0x4848 4A3C
TX0_CP	RW	32	0x0000 0040	0x4848 4A40
TX1_CP	RW	32	0x0000 0044	0x4848 4A44
TX2_CP	RW	32	0x0000 0048	0x4848 4A48
TX3_CP	RW	32	0x0000 004C	0x4848 4A4C
TX4_CP	RW	32	0x0000 0050	0x4848 4A50
TX5_CP	RW	32	0x0000 0054	0x4848 4A54
TX6_CP	RW	32	0x0000 0058	0x4848 4A58
TX7_CP	RW	32	0x0000 005C	0x4848 4A5C
RX0_CP	RW	32	0x0000 0060	0x4848 4A60
RX1_CP	RW	32	0x0000 0064	0x4848 4A64
RX2_CP	RW	32	0x0000 0068	0x4848 4A68
RX3_CP	RW	32	0x0000 006C	0x4848 4A6C
RX4_CP	RW	32	0x0000 0070	0x4848 4A70
RX5_CP	RW	32	0x0000 0074	0x4848 4A74
RX6_CP	RW	32	0x0000 0078	0x4848 4A78
RX7_CP	RW	32	0x0000 007C	0x4848 4A7C

24.11.6.6.2 STATERAM Register Description

Table 24-1219. TX0_HDP

Address Offset	0x0000 0000		
Physical Address	0x4848 4A00	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 0 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1220. Register Call Summary for Register TX0_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1221. TX1_HDP

Address Offset	0x0000 0004		
Physical Address	0x4848 4A04	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 1 head descriptor pointer		

Table 24-1221. TX1_HDP (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1222. Register Call Summary for Register TX1_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1223. TX2_HDP

Address Offset	0x0000 0008		
Physical Address	0x4848 4A08	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 2 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1224. Register Call Summary for Register TX2_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1225. TX3_HDP

Address Offset	0x0000 000C		
Physical Address	0x4848 4A0C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 3 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1226. Register Call Summary for Register TX3_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1227. TX4_HDP

Address Offset	0x0000 0010		
Physical Address	0x4848 4A10	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 4 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1228. Register Call Summary for Register TX4_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1229. TX5_HDP

Address Offset	0x0000 0014		
Physical Address	0x4848 4A14	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 5 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1230. Register Call Summary for Register TX5_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1231. TX6_HDP

Address Offset	0x0000 0018		
Physical Address	0x4848 4A18	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 6 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TX_HDP

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1232. Register Call Summary for Register TX6_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1233. TX7_HDP

Address Offset	0x0000 001C		
Physical Address	0x4848 4A1C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 7 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1234. Register Call Summary for Register TX7_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1235. RX0_HDP

Address Offset	0x0000 0020		
Physical Address	0x4848 4A20	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 0 channel 0 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1236. Register Call Summary for Register RX0_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1237. RX1_HDP

Address Offset	0x0000 0024		
Physical Address	0x4848 4A24	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 1 channel 1 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1238. Register Call Summary for Register RX1_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1239. RX2_HDP

Address Offset	0x0000 0028		
Physical Address	0x4848 4A28	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 2 channel 2 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1240. Register Call Summary for Register RX2_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1241. RX3_HDP

Address Offset	0x0000 002C		
Physical Address	0x4848 4A2C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 3 channel 3 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RX_HDP

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1242. Register Call Summary for Register RX3_HDP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 24-1243. RX4_HDP

Address Offset	0x0000 0030		
Physical Address	0x4848 4A30	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 4 channel 4 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1244. Register Call Summary for Register RX4_HDP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 24-1245. RX5_HDP

Address Offset	0x0000 0034		
Physical Address	0x4848 4A34	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 5 channel 5 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1246. Register Call Summary for Register RX5_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1247. RX6_HDP

Address Offset	0x0000 0038		
Physical Address	0x4848 4A38	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 6 channel 6 head desc pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1248. Register Call Summary for Register RX6_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1249. RX7_HDP

Address Offset	0x0000 003C		
Physical Address	0x4848 4A3C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 7 channel 7 head desc pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 24-1250. Register Call Summary for Register RX7_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1251. TX0_CP

Address Offset	0x0000 0040		
Physical Address	0x4848 4A40	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 0 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TX_CP

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1252. Register Call Summary for Register TX0_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1253. TX1_CP

Address Offset	0x0000 0044		
Physical Address	0x4848 4A44	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 1 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1254. Register Call Summary for Register TX1_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1255. TX2_CP

Address Offset	0x0000 0048		
Physical Address	0x4848 4A48	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 2 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1256. Register Call Summary for Register TX2_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1257. TX3_CP

Address Offset	0x0000 004C

Table 24-1257. TX3_CP (continued)

Physical Address	0x4848 4A4C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 3 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1258. Register Call Summary for Register TX3_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 24-1259. TX4_CP

Address Offset	0x0000 0050		
Physical Address	0x4848 4A50	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 4 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1260. Register Call Summary for Register TX4_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 24-1261. TX5_CP

Address Offset	0x0000 0054		
Physical Address	0x4848 4A54	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 5 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1262. Register Call Summary for Register TX5_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1263. TX6_CP

Address Offset	0x0000 0058		
Physical Address	0x4848 4A58	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 6 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1264. Register Call Summary for Register TX6_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1265. TX7_CP

Address Offset	0x0000 005C		
Physical Address	0x4848 4A5C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 7 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 24-1266. Register Call Summary for Register TX7_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1267. RX0_CP

Address Offset	0x0000 0060		
Physical Address	0x4848 4A60	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 0 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1268. Register Call Summary for Register RX0_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1269. RX1_CP

Address Offset	0x0000 0064		
Physical Address	0x4848 4A64	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 1 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1270. Register Call Summary for Register RX1_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1271. RX2_CP

Address Offset	0x0000 0068		
Physical Address	0x4848 4A68	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 2 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1272. Register Call Summary for Register RX2_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1273. RX3_CP

Address Offset	0x0000 006C		
Physical Address	0x4848 4A6C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 3 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1274. Register Call Summary for Register RX3_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1275. RX4_CP

Address Offset	0x0000 0070		
Physical Address	0x4848 4A70	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 4 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1276. Register Call Summary for Register RX4_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1277. RX5_CP

Address Offset	0x0000 0074		
Physical Address	0x4848 4A74	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 5 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1278. Register Call Summary for Register RX5_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1279. RX6_CP

Address Offset	0x0000 0078		
Physical Address	0x4848 4A78	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 6 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1280. Register Call Summary for Register RX6_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 24-1281. RX7_CP

Address Offset	0x0000 007C		
Physical Address	0x4848 4A7C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 7 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 24-1282. Register Call Summary for Register RX7_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

24.11.6.7 CPTS registers

24.11.6.7.1 CPTS Register Summary

Table 24-1283. CPTS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CPTS Physical Address
CPTS_IDVER	R	32	0x0000 0000	0x4848 4C00
CPTS_CONTROL	RW	32	0x0000 0004	0x4848 4C04
CPTS_TS_PUSH	W	32	0x0000 000C	0x4848 4C0C
CPTS_TS_LOAD_VAL	RW	32	0x0000 0010	0x4848 4C10
CPTS_TS_LOAD_EN	W	32	0x0000 0014	0x4848 4C14
CPTS_INTSTAT_RAW	RW	32	0x0000 0020	0x4848 4C20

Table 24-1283. CPTS Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CPTS Physical Address
CPTS_INTSTAT_MASKED	R	32	0x0000 0024	0x4848 4C24
CPTS_INT_ENABLE	RW	32	0x0000 0028	0x4848 4C28
CPTS_EVENT_POP	W	32	0x0000 0030	0x4848 4C30
CPTS_EVENT_LOW	R	32	0x0000 0034	0x4848 4C34
CPTS_EVENT_HIGH	R	32	0x0000 0038	0x4848 4C38

24.11.6.7.2 CPTS Register Description

Table 24-1284. CPTS_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4C00	**Instance**	CPTS
Description	CPTS revision		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPTS revision value	R	0x-

Table 24-1285. Register Call Summary for Register CPTS_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Register Summary: [0]

Table 24-1286. CPTS_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4C04	**Instance**	CPTS
Description	Time sync control register		
Type	RW		

31–12	11	10	9	8	7–2	1	0
RESERVED	HW4_TS_PUSH_EN	HW3_TS_PUSH_EN	HW2_TS_PUSH_EN	HW1_TS_PUSH_EN	RESERVED	INT_TEST	CPTS_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	HW4_TS_PUSH_EN	Hardware push 4 enable	RW	0x0
10	HW3_TS_PUSH_EN	Hardware push 3 enable	RW	0x0
9	HW2_TS_PUSH_EN	Hardware push 2 enable	RW	0x0
8	HW1_TS_PUSH_EN	Hardware push 1 enable	RW	0x0
7:2	RESERVED		R	0x0
1	INT_TEST	Interrupt Test - When set, this bit allows the raw interrupt to be written to facilitate interrupt test.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	CPTS_EN	Time Sync Enable - When disabled (cleared to zero), the RCLK domain is held in reset. 0 - Time Sync Disabled 1 - Time Sync Enabled	RW	0x0

Table 24-1287. Register Call Summary for Register CPTS_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Initialization: [0] [1]
- Time Sync Events: [2]
- CPTS Register Summary: [3]
- CPTS Register Description: [4]

Table 24-1288. CPTS_TS_PUSH

Address Offset	0x0000 000C		
Physical Address	0x4848 4C0C	**Instance**	CPTS
Description	Time stamp event push register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TS_PUSH

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PUSH	Time stamp event push - When a logic high is written to this bit a time stamp event is pushed onto the event FIFO. The time stamp value is the time of the write of this register, not the time of the event read. The time stamp value can then be read on interrupt via the event registers. Software should not push a second time stamp event onto the event FIFO until the first time stamp value has been read from the event FIFO (there should be only one time stamp event in the event FIFO at any given time). This bit is write only and always reads zero.	W	0x0

Table 24-1289. Register Call Summary for Register CPTS_TS_PUSH

Gigabit Ethernet Switch (GMAC_SW)

- Time Sync Events: [0]
- CPTS Register Summary: [1]

Table 24-1290. CPTS_TS_LOAD_VAL

Address Offset	0x0000 0010		
Physical Address	0x4848 4C10	**Instance**	CPTS
Description	Time stamp load value register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS_LOAD_VAL																															

Bits	Field Name	Description	Type	Reset
31:0	TS_LOAD_VAL	Time Stamp Load Value - Writing the CPTS_TS_LOAD_EN[0] TS_LOAD_EN bit causes the value contained in this register to be written into the time stamp. The time stamp value is read by initiating a time stamp push event, not by reading this register. When reading this register, the value read is not the time stamp, but is the value that was last written to this register.	RW	0x0

Table 24-1291. Register Call Summary for Register CPTS_TS_LOAD_VAL

Gigabit Ethernet Switch (GMAC_SW)

- Time Stamp Value: [0]
- CPTS Register Summary: [1]
- CPTS Register Description: [2]

Table 24-1292. CPTS_TS_LOAD_EN

Address Offset	0x0000 0014		
Physical Address	0x4848 4C14	**Instance**	CPTS
Description	Time stamp load enable register		
Type	W		

31–1	0
RESERVED	TS_LOAD_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_LOAD_EN	Time Stamp Load - Writing a one to this bit enables the time stamp value to be written via the CPTS_TS_LOAD_VAL register. This feature is included for test purposes. This bit is write only.	W	0x0

Table 24-1293. Register Call Summary for Register CPTS_TS_LOAD_EN

Gigabit Ethernet Switch (GMAC_SW)

- Time Stamp Value: [0]
- CPTS Register Summary: [1]
- CPTS Register Description: [2]

Table 24-1294. CPTS_INTSTAT_RAW

Address Offset	0x0000 0020		
Physical Address	0x4848 4C20	**Instance**	CPTS
Description	Time sync interrupt status raw register		
Type	RW		

31–1	0
RESERVED	TS_PEND_RAW

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PEND_RAW	TS_PEND_RAW int read (before enable). Writable when CPTS_CONTROL[1] INT_TEST = 1. A one in this bit indicates that there is one or more events in the event FIFO.	RW	0x0

Table 24-1295. Register Call Summary for Register CPTS_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Interrupt Handling: [0] [1]
- CPTS Register Summary: [2]

Table 24-1296. CPTS_INTSTAT_MASKED

Address Offset	0x0000 0024		
Physical Address	0x4848 4C24	**Instance**	CPTS
Description	Time sync interrupt status masked register		
Type	R		

31–1	0
RESERVED	TS_PEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PEND	TS_PEND masked interrupt read (after enable).	R	0x0

Table 24-1297. Register Call Summary for Register CPTS_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Register Summary: [0]

Table 24-1298. CPTS_INT_ENABLE

Address Offset	0x0000 0028		
Physical Address	0x4848 4C28	**Instance**	CPTS
Description	Time sync interrupt enable register		
Type	RW		

31–1	0
RESERVED	TS_PEND_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PEND_EN	TS_PEND masked interrupt enable.	RW	0x0

Table 24-1299. Register Call Summary for Register CPTS_INT_ENABLE

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Initialization: [0]
- CPTS Interrupt Handling: [1] [2]
- CPTS Register Summary: [3]

Table 24-1300. CPTS_EVENT_POP

Address Offset	0x0000 0030		
Physical Address	0x4848 4C30	**Instance**	CPTS
Description	Event interrupt pop register		
Type	W		

31–1	0
RESERVED	EVENT_POP

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EVENT_POP	Event Pop - When a logic high is written to this bit an event is popped off the event FIFO. The event FIFO pop occurs as part of the interrupt process after the event has been read in the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers. Popping an event discards the event and causes the next event, if any, to be moved to the top of the FIFO ready to be read by software on interrupt.	W	0x0

Table 24-1301. Register Call Summary for Register CPTS_EVENT_POP

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Interrupt Handling: [0] [1]
- CPTS Register Summary: [2]

Table 24-1302. CPTS_EVENT_LOW

Address Offset	0x0000 0034		
Physical Address	0x4848 4C34	**Instance**	CPTS
Description	Lower 32-bits of the event value		
Type	R		

31–0
TIME_STAMP

Bits	Field Name	Description	Type	Reset
31:0	TIME_STAMP	Time Stamp - The timestamp is valid for transmit, receive, and time stamp push event types. The timestamp value is not valid for counter roll event types.	R	0x0

Table 24-1303. Register Call Summary for Register CPTS_EVENT_LOW

Gigabit Ethernet Switch (GMAC_SW)

- Time Sync Events: [0] [1]
- CPTS Interrupt Handling: [2] [3]
- CPTS Register Summary: [4]
- CPTS Register Description: [5]

Table 24-1304. CPTS_EVENT_HIGH

Address Offset	0x0000 0038		
Physical Address	0x4848 4C38	**Instance**	CPTS
Description	Upper 32-bits of the event value		
Type	R		

31 30 29	28 27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	PORT_NUMBER	EVENT_TYPE	MESSAGE_TYPE	SEQUENCE_ID

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	PORT_NUMBER	Port Number - indicates the port number of an ethernet event or the hardware push pin number (1 to 4).	R	0x0
23:20	EVENT_TYPE	Time Sync Event Type 0x0 - Time Stamp Push Event 0x1 - Time Stamp Rollover Event 0x2 - Time Stamp Half Rollover Event 0x3 - Hardware Time Stamp Push Event 0x4 - Ethernet Receive Event 0x5 - Ethernet Transmit Event	R	0x0
19:16	MESSAGE_TYPE	Message type - The message type value that was contained in an ethernet transmit or receive time sync packet. This field is valid only for ethernet transmit or receive events.	R	0x0
15:0	SEQUENCE_ID	Sequence ID - The 16-bit sequence id is the value that was contained in an ethernet transmit or receivetime sync packet. This field is valid only for ethernet transmit or receive events.	R	0x0

Table 24-1305. Register Call Summary for Register CPTS_EVENT_HIGH

Gigabit Ethernet Switch (GMAC_SW)

- Time Sync Events: [0] [1] [2]
- CPTS Interrupt Handling: [3] [4]
- CPTS Register Summary: [5]
- CPTS Register Description: [6]

24.11.6.8 ALE registers

24.11.6.8.1 ALE Register Summary

Table 24-1306. ALE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ALE Physical Address
ALE_IDVER	R	32	0x0000 0000	0x4848 4D00
ALE_CONTROL	RW	32	0x0000 0008	0x4848 4D08
ALE_PRESCALE	RW	32	0x0000 0010	0x4848 4D10
ALE_UNKNOWN_VLAN	RW	32	0x0000 0018	0x4848 4D18
ALE_TBLCTL	RW	32	0x0000 0020	0x4848 4D20

Table 24-1306. ALE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ALE Physical Address
ALE_TBLW2	RW	32	0x0000 0034	0x4848 4D34
ALE_TBLW1	RW	32	0x0000 0038	0x4848 4D38
ALE_TBLW0	RW	32	0x0000 003C	0x4848 4D3C
ALE_PORTCTL0	RW	32	0x0000 0040	0x4848 4D40
ALE_PORTCTL1	RW	32	0x0000 0044	0x4848 4D44
ALE_PORTCTL2	RW	32	0x0000 0048	0x4848 4D48
ALE_PORTCTL3	RW	32	0x0000 004C	0x4848 4D4C
ALE_PORTCTL4	RW	32	0x0000 0050	0x4848 4D50
ALE_PORTCTL5	RW	32	0x0000 0054	0x4848 4D54

24.11.6.8.2 ALE Register Description

Table 24-1307. ALE_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4D00	**Instance**	ALE
Description	ADDRESS LOOKUP ENGINE revision register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REVISION</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	REVISION	ALE Revision Value	R	0x-

Table 24-1308. ALE_CONTROL

Address Offset	0x0000 0008		
Physical Address	0x4848 4D08	**Instance**	ALE
Description	Address lookup engine control register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>ENABLE_ALE</td><td>CLEAR_TABLE</td><td>AGE_OUT_NOW</td><td colspan="20">RESERVED</td><td>EN_P0_UNI_FLOOD</td><td>LEARN_NO_VID</td><td>EN_VID0_MODE</td><td>ENABLE_OUI_DENY</td><td>BYPASS</td><td>RATE_LIMIT_TX</td><td>VLAN_AWARE</td><td>ENABLE_AUTH_MODE</td><td>ENABLE_RATE_LIMIT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	ENABLE_ALE	Enable ALE - 0 - Drop all packets 1 - Enable ALE packet processing	RW	0x0

Bits	Field Name	Description	Type	Reset
30	CLEAR_TABLE	Clear ALE address table - Setting this bit causes the ALE hardware to write all table bit values to zero. Software must perform a clear table operation as part of the ALE setup/configuration process. Setting this bit causes all ALE accesses to be held up for 64 clocks while the clear is performed. Access to all ALE registers will be blocked (wait states) until the 64 clocks have completed. This bit cannot be read as one because the read is blocked until the clear table is completed at which time this bit is cleared to zero.	RW	0x0
29	AGE_OUT_NOW	Age Out Address Table Now - Setting this bit causes the ALE hardware to remove (free up) any ageable table entry that does not have a set touch bit. This bit is cleared when the age out process has completed. This bit may be read. The age out process takes 4096 clocks best case (no ale packet processing during ageout) and 66550 clocks absolute worst case.	RW	0x0
28:9	RESERVED		R	0x0
8	EN_P0_UNI_FLOOD	Enable Port 0 (Host Port) unicast flood 0 - do not flood unknown unicast packets to host port (p0) 1 - flood unknown unicast packets to host port (p0)	RW	0x0
7	LEARN_NO_VID	Learn No VID - 0 - VID is learned with the source address 1 - VID is not learned with the source address (source address is not tied to VID).	RW	0x0
6	EN_VID0_MODE	Enable VLAN ID = 0 Mode 0 - Process the packet with VID = PORT_VLAN[11:0]. 1 - Process the packet with VID = 0.	RW	0x0
5	ENABLE_OUI_DENY	Enable OUI Deny Mode - When set this bit indicates that a packet with a non OUI table entry matching source address will be dropped to the host unless the destination address matches a multicast table entry with the super bit set.	RW	0x0
4	BYPASS	ALE Bypass - When set, all packets received on ports 0 and 1 are sent to the host (only to the host).	RW	0x0
3	RATE_LIMIT_TX	Rate Limit Transmit mode - 0 - Broadcast and multicast rate limit counters are received port based 1 - Broadcast and multicast rate limit counters are transmit port based.	RW	0x0
2	VLAN_AWARE	ALE VLAN Aware - Determines what is done if VLAN not found. 0 - Flood if VLAN not found 1 - Drop packet if VLAN not found	RW	0x0
1	ENABLE_AUTH_MODE	Enable MAC Authorization Mode - Mac authorization mode requires that all table entries be made by the host software. There are no learned address in authorization mode and the packet will be dropped if the source address is not found (and the destination address is not a multicast address with the super table entry bit set). 0 - The ALE is not in MAC authorization mode 1 - The ALE is in MAC authorization mode	RW	0x0
0	ENABLE_RATE_LIMIT	Enable Broadcast and Multicast Rate Limit 0 - Broadcast/Multicast rates not limited 1 - Broadcast/Multicast packet reception limited to the port control register rate limit fields.	RW	0x0

Table 24-1309. ALE_PRESCALE

Address Offset	0x0000 0010		
Physical Address	0x4848 4D10	**Instance**	ALE

Table 24-1309. ALE_PRESCALE (continued)

Description	Address lookup engine prescale register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PRESCALE																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	PRESCALE	ALE Prescale Register - The input clock is divided by this value for use in the multicast/broadcast rate limiters. The minimum operating value is 0x10. The prescaler is off when the value is zero.	RW	0x0

Table 24-1310. ALE_UNKNOWN_VLAN

Address Offset	0x0000 0018		
Physical Address	0x4848 4D18	**Instance**	ALE
Description	Address lookup engine unknown vlan register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		UNKNOWN_FORCE_UNTAGGED_EGRESS						RESERVED		UNKNOWN_REG_MCAST_FLOOD_MASK						RESERVED		UNKNOWN_MCAST_FLOOD_MASK						RESERVED		UNKNOWN_VLAN_MEMBER_LIST					

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:24	UNKNOWN_FORCE_UNTAGGED_EGRESS	Unknown VLAN Force Untagged Egress.	RW	0x0
23:22	RESERVED		R	0x0
21:16	UNKNOWN_REG_MCAST_FLOOD_MASK	Unknown VLAN Registered Multicast Flood Mask	RW	0x0
15:14	RESERVED		R	0x0
13:8	UNKNOWN_MCAST_FLOOD_MASK	Unknown VLAN Multicast Flood Mask	RW	0x0
7:6	RESERVED		R	0x0
5:0	UNKNOWN_VLAN_MEMBER_LIST	Unknown VLAN Member List	RW	0x0

Table 24-1311. ALE_TBLCTL

Address Offset	0x0000 0020		
Physical Address	0x4848 4D20	**Instance**	ALE
Description	Address lookup engine table control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRITE_RDZ	RESERVED																					ENTRY_POINTER									

Bits	Field Name	Description	Type	Reset
31	WRITE_RDZ	Write Bit - This bit is always read as zero. Writing a 1 to this bit causes the three table word register values to be written to the entry_pointer location in the address table. Writing a 0 to this bit causes the three table word register values to be loaded from the entry_pointer location in the address table so that they may be subsequently read. A read of any ALE address location will be stalled until the read or write has completed.	RW	0x0
30:10	RESERVED		R	0x0
9:0	ENTRY_POINTER	Table Entry Pointer - The entry_pointer contains the table entry value that will be read/written with accesses to the table word registers.	RW	0x0

Table 24-1312. ALE_TBLW2

Address Offset	0x0000 0034		
Physical Address	0x4848 4D34	**Instance**	ALE
Description	Address lookup engine table word 2 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ENTRY71_64							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ENTRY71_64	Table entry bits 71:64	RW	0x0

Table 24-1313. ALE_TBLW1

Address Offset	0x0000 0038		
Physical Address	0x4848 4D38	**Instance**	ALE
Description	Address lookup engine table word 1 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTRY63_32																															

Bits	Field Name	Description	Type	Reset
31:0	ENTRY63_32	Table entry bits 63:32	RW	0x0

Table 24-1314. ALE_TBLW0

Address Offset	0x0000 003C		
Physical Address	0x4848 4D3C	**Instance**	ALE
Description	Address lookup engine table word 0 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTRY31_0																															

Bits	Field Name	Description	Type	Reset
31:0	ENTRY31_0	Table entry bits 31:0	RW	0x0

Table 24-1315. ALE_PORTCTL0

Address Offset	0x0000 0040		
Physical Address	0x4848 4D40	**Instance**	ALE

Table 24-1315. ALE_PORTCTL0 (continued)

Description	Address lookup engine port 0 control register
Type	RW

31-24	23-16	15-6	5	4	3	2	1-0
BCAST_LIMIT	MCAST_LIMIT	RESERVED	NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 24-1316. ALE_PORTCTL1

Address Offset	0x0000 0044		
Physical Address	0x4848 4D44	**Instance**	ALE
Description	Address lookup engine port 1 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BCAST_LIMIT	MCAST_LIMIT	RESERVED	NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 24-1317. ALE_PORTCTL2

Address Offset	0x0000 0048		
Physical Address	0x4848 4D48	**Instance**	ALE
Description	Address lookup engine port 2 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BCAST_LIMIT	MCAST_LIMIT	RESERVED	NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 24-1318. ALE_PORTCTL3

Address Offset	0x0000 004C		
Physical Address	0x4848 4D4C	**Instance**	ALE
Description	Address lookup engine port 3 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BCAST_LIMIT	MCAST_LIMIT	RESERVED	NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 24-1319. ALE_PORTCTL4

Address Offset	0x0000 0050		
Physical Address	0x4848 4D50	**Instance**	ALE
Description	Address lookup engine port 4 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BCAST_LIMIT	MCAST_LIMIT	RESERVED	NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 24-1320. ALE_PORTCTL5

Address Offset	0x0000 0054		
Physical Address	0x4848 4D54	**Instance**	ALE
Description	Address lookup engine port 5 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BCAST_LIMIT	MCAST_LIMIT	RESERVED	NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

24.11.6.9 SL registers

24.11.6.9.1 SL Register Summary

Table 24-1321. SL Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SL1 Physical Address	SL2 Physical Address
SL_IDVER	R	32	0x0000 0000	0x4848 4D80	0x4848 4DC0
SL_MACCONTROL	RW	32	0x0000 0004	0x4848 4D84	0x4848 4DC4
SL_MACSTATUS	R	32	0x0000 0008	0x4848 4D88	0x4848 4DC8
SL_SOFT_RESET	RW	32	0x0000 000C	0x4848 4D8C	0x4848 4DCC
SL_RX_MAXLEN	RW	32	0x0000 0010	0x4848 4D90	0x4848 4DD0
SL_BOFFTEST	RW	32	0x0000 0014	0x4848 4D94	0x4848 4DD4
SL_RX_PAUSE	R	32	0x0000 0018	0x4848 4D98	0x4848 4DD8

Table 24-1321. SL Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	SL1 Physical Address	SL2 Physical Address
SL_TX_PAUSE	R	32	0x0000 001C	0x4848 4D9C	0x4848 4DDC
SL_EMCONTROL	RW	32	0x0000 0020	0x4848 4DA0	0x4848 4DE0
SL_RX_PRI_MAP	RW	32	0x0000 0024	0x4848 4DA4	0x4848 4DE4
SL_TX_GAP	RW	32	0x0000 0028	0x4848 4DA8	0x4848 4DE8

24.11.6.9.2 SL Register Description

Table 24-1322. SL_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4D80 0x4848 4DC0	**Instance**	SL1 SL2
Description	CPGMAC_SL revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPGMAC_SL revision Value	R	0x-

Table 24-1323. Register Call Summary for Register SL_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]

Table 24-1324. SL_MACCONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4D84 0x4848 4DC4	**Instance**	SL1 SL2
Description	CPGMAC_SL MAC control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							RX_CMF_EN	RX_CSF_EN	RX_CEF_EN	TX_SHORT_GAP_LIM_EN	RESERVED		EXT_EN	GIG_FORCE	IFCTL_B	IFCTL_A	RESERVED			CMD_IDLE	TX_SHORT_GAP_EN	RESERVED		GIG	TX_PACE	GMII_EN	TX_FLOW_EN	RX_FLOW_EN	MTEST	LOOPBACK	FULLDUPLEX

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
24	RX_CMF_EN	RX Copy MAC Control Frames Enable - Enables MAC control frames to be transferred to memory. MAC control frames are normally acted upon (if enabled), but not copied to memory. MAC control frames that are pause frames will be acted upon if enabled in the SL_MACCONTROL register, regardless of the value of RX_CMF_EN. Frames transferred to memory due to RX_CMF_EN will have the control bit set in their EOP buffer descriptor. 0 - MAC control frames are filtered (but acted upon if enabled). 1 - MAC control frames are transferred to memory.	RW	0x0
23	RX_CSF_EN	RX Copy Short Frames Enable - Enables frames or fragments shorter than 64 bytes to be copied to memory. Frames transferred to memory due to RX_CSF_EN will have the fragment or undersized bit set in their receive footer. Fragments are short frames that contain CRC/align/code errors and undersized are short frames without errors. 0 - Short frames are filtered 1 - Short frames are transferred to memory.	RW	0x0
22	RX_CEF_EN	RX Copy Error Frames Enable - Enables frames containing errors to be transferred to memory. The appropriate error bit will be set in the frame receive footer. Frames containing errors will be filtered when RX_CEF _EN is not set. 0 - Frames containing errors are filtered. 1 - Frames containing errors are transferred to memory.	RW	0x0
21	TX_SHORT_GAP_LIM_EN	Transmit Short Gap Limit Enable When set this bit limits the number of short gap packets transmitted to 100ppm. Each time a short gap packet is sent, a counter is loaded with 10,000 and decremented on each wireside clock. Another short gap packet will not be sent out until the counter decrements to zero. This mode is included to preclude the host from filling up the FIFO and sending every packet out with short gap which would violate the maximum number of packets per second allowed.	RW	0x0
20:19	RESERVED		R	0x0
18	EXT_EN	Control Enable - Enables the fullduplex and gigabit mode to be selected from the FULLDUPLEX_IN and GIG_IN input signals and not from the FULLDUPLEX and GIG bits in this register. The FULLDUPLEX_MODE bit reflects the actual fullduplex mode selected 0 - Use this setting for RMII/GMII mode . 1 - Use this setting for RGMII mode	RW	0x0
17	GIG_FORCE	Gigabit Mode Force - This bit is used to force the CPGMAC_SL into gigabit mode if the input GMII_MTCLK has been stopped by the PHY.	RW	0x0
16	IFCTL_B	Interface Control B (NOT FUNCTIONAL) 0 - 10Mbps operation 1 - 100Mbps operation	RW	0x0
15	IFCTL_A	Interface Control A 0 - 10Mbps operation 1 - 100Mbps operation	RW	0x0
14:12	RESERVED		R	0x0
11	CMD_IDLE	Command Idle 0 - Idle not commanded 1 - Idle Commanded (read IDLE in SL_MACSTATUS)	RW	0x0

Bits	Field Name	Description	Type	Reset
10	TX_SHORT_GAP_EN	Transmit Short Gap Enable 0 - Transmit with a short IPG is disabled 1 - Transmit with a short IPG (when TX_SHORT_GAP input is asserted) is enabled.	RW	0x0
9:8	RESERVED		R	0x0
7	GIG	Gigabit Mode - 0 - 10/100 mode 1 - Gigabit mode (full duplex only) The GIG_OUT output is the value of this bit.	RW	0x0
6	TX_PACE	Transmit Pacing Enable 0 - Transmit Pacing Disabled 1 - Transmit Pacing Enabled	RW	0x0
5	GMII_EN	GMII Enable - 0 - GMII RX and TX held in reset. 1 - GMII RX and TX released from reset.	RW	0x0
4	TX_FLOW_EN	Transmit Flow Control Enable - Determines if incoming pause frames are acted upon in full-duplex mode. Incoming pause frames are not acted upon in half-duplex mode regardless of this bit setting. The RX_MBP_Enable bits determine whether or not received pause frames are transferred to memory. 0 - Transmit Flow Control Disabled. Full-duplex mode - Incoming pause frames are not acted upon. 1 - Transmit Flow Control Enabled . Full-duplex mode - Incoming pause frames are acted upon.	RW	0x0
3	RX_FLOW_EN	Receive Flow Control Enable - 0 - Receive Flow Control Disabled Half-duplex mode - No flow control generated collisions are sent. Full-duplex mode - No outgoing pause frames are sent. 1 - Receive Flow Control Enabled Half-duplex mode - Collisions are initiated when receive flow control is triggered. Full-duplex mode - Outgoing pause frames are sent when receive flow control is triggered.	RW	0x0
2	MTEST	Manufacturing Test mode - This bit must be set to allow writes to the SL_BOFFTEST and SL_RX_PAUSE/ SL_TX_PAUSE registers.	RW	0x0
1	LOOPBACK	Loop Back Mode - Loopback mode forces internal fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The LOOPBACK bit should be changed only when GMII_EN is deasserted. 0 - Not looped back 1 - Loop Back Mode enabled	RW	0x0
0	FULLDUPLEX	Full Duplex mode - Gigabit mode forces fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The FULLDUPLEX_OUT output is the value of this register bit 0 - half duplex mode 1 - full duplex mode	RW	0x0

Table 24-1325. Register Call Summary for Register SL_MACCONTROL

Gigabit Ethernet Switch (GMAC_SW)

- G/MII Interface:
- Address Lookup Engine (ALE): [1] [2]
- Ethernet MAC Sliver (CPGMAC_SL): [3] [4] [5] [6] [7]
- Flow Control: [8] [9] [10] [11] [12] [13] [14] [15]
- Short Gap: [16]
- RX Buffer Descriptors: [17] [18] [19]
- MDIO Functional Description: [20]
- STATS Register Description: [21]
- SL Register Summary: [22]
- SL Register Description: [23]

Table 24-1326. SL_MACSTATUS

Address Offset	0x0000 0008		
Physical Address	0x4848 4D88 0x4848 4DC8	**Instance**	SL1 SL2
Description	CPGMAC_SL MAC status register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>IDLE</td><td colspan="26">RESERVED</td><td>EXT_GIG</td><td>EXT_FULLDUPLEX</td><td>RESERVED</td><td>RX_FLOW_ACT</td><td>TX_FLOW_ACT</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31	IDLE	CPGMAC_SL IDLE - The CPGMAC_SL is in the idle state (valid after an idle command) 0 - The CPGMAC_SL is not in the idle state. 1 - The CPGMAC_SL is in the idle state.	R	0x1
30:5	RESERVED		R	0x0
4	EXT_GIG	External GIG - This is the value of the EXT_GIG input bit.	R	0x0
3	EXT_FULLDUPLEX	External Fullduplex - This is the value of the EXT_FULLDUPLEX input bit.	R	0x0
2	RESERVED		R	0x0
1	RX_FLOW_ACT	Receive Flow Control Active - When asserted, indicates that receive flow control is enabled and triggered.	R	0x0
0	TX_FLOW_ACT	Transmit Flow Control Active - When asserted, this bit indicates that the pause time period is being observed for a received pause frame. No new transmissions will begin while this bit is asserted except for the transmission of pause frames. Any transmission in progress when this bit is asserted will complete.	R	0x0

Table 24-1327. Register Call Summary for Register SL_MACSTATUS

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 24-1328. SL_SOFT_RESET

Address Offset	0x0000 000C

Table 24-1328. SL_SOFT_RESET (continued)

Physical Address	0x4848 4D8C 0x4848 4DCC	**Instance**	SL1 SL2
Description	CPGMAC_SL soft reset register		
Type	RW		

31:1	0
RESERVED	SOFT_RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the CPGMAC_SL logic to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.	RW	0x0

Table 24-1329. Register Call Summary for Register SL_SOFT_RESET

Gigabit Ethernet Switch (GMAC_SW)

- Software Reset: [0]
- SL Register Summary: [1]

Table 24-1330. SL_RX_MAXLEN

Address Offset	0x0000 0010		
Physical Address	0x4848 4D90 0x4848 4DD0	**Instance**	SL1 SL2
Description	CPGMAC_SL RX Maximum length register		
Type	RW		

31:14	13:0
RESERVED	RX_MAXLEN

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	RX_MAXLEN	RX Maximum Frame Length - This field determines the maximum length of a received frame. The reset value is 1518 (dec). Frames with byte counts greater than rx_maxlen are long frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment error are jabber frames. The maximum value is 16,383.	RW	0x5EE

Table 24-1331. Register Call Summary for Register SL_RX_MAXLEN

Gigabit Ethernet Switch (GMAC_SW)

- Ethernet MAC Sliver (CPGMAC_SL): [0] [1]
- Flow Control: [2]
- STATS Register Description: [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17]
- SL Register Summary: [18]

Table 24-1332. SL_BOFFTEST

Address Offset	0x0000 0014

Table 24-1332. SL_BOFFTEST (continued)

Physical Address	0x4848 4D94 0x4848 4DD4	**Instance**	SL1 SL2
Description	CPGMAC_SL backoff test register		
Type	RW		

31	30:26	25:16	15:12	11:10	9:0
RESERVED	PACEVAL	RNDNUM	COLL_COUNT	RESERVED	TX_BACKOFF

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:26	PACEVAL	Pacing Register Current Value. A non-zero value in this field indicates that transmit pacing is active. A transmit frame collision or deferral causes PACEVAL to loaded with decimal 31, good frame transmissions (with no collisions or deferrals) cause PACEVAL to be decremented down to zero. When PACEVAL is nonzero, the transmitter delays 4 IPGs between new frame transmissions after each successfully transmitted frame that had no deferrals or collisions. Transmit pacing helps reduce 'capture' effects improving overall network bandwidth.	RW	0x0
25:16	RNDNUM	Backoff Random Number Generator - This field allows the Backoff Random Number Generator to be read (or written in test mode only). This field can be written only when mtest has previously been set. Reading this field returns the generator's current value. The value is reset to zero and begins counting on the clock after the deassertion of reset.	RW	0x0
15:12	COLL_COUNT	Collision Count - The number of collisions the current frame has experienced.	R	0x0
11:10	RESERVED		R	0x0
9:0	TX_BACKOFF	Backoff Count - This field allows the current value of the backoff counter to be observed for test purposes. This field is loaded automatically according to the backoff algorithm, and is decremented by one for each slot time after the collision.	R	0x0

Table 24-1333. Register Call Summary for Register SL_BOFFTEST

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 24-1334. SL_RX_PAUSE

Address Offset	0x0000 0018		
Physical Address	0x4848 4D98 0x4848 4DD8	**Instance**	SL1 SL2
Description	CPGMAC_SL receive pause timer register		
Type	R		

31:16	15:0
RESERVED	RX_PAUSETIMER

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_PAUSETIMER	RX Pause Timer Value - This field allows the contents of the receive pause timer to be observed (and written in test mode). The receive pause timer is loaded with 0xFF00 when the CPGMAC_SL sends an outgoing pause frame (with pause time of 0xFFFF). The receive pause timer is decremented at slot time intervals. If the receive pause timer decrements to zero, then another outgoing pause frame will be sent and the load/ decrement process will be repeated.	R	0x0

Table 24-1335. Register Call Summary for Register SL_RX_PAUSE

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 24-1336. SL_TX_PAUSE

Address Offset	0x0000 001C		
Physical Address	0x4848 4D9C 0x4848 4DDC	**Instance**	SL1 SL2
Description	CPGMAC_SL transmit pause timer register		
Type	R		

31–16	15–0
RESERVED	TX_PAUSETIMER

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	TX_PAUSETIMER	TX Pause Timer Value - This field allows the contents of the transmit pause timer to be observed (and written in test mode). The transmit pause timer is loaded by a received (incoming) pause frame, and then decremented, at slottime intervals, down to zero at which time CPGMAC_SL transmit frames are again enabled.	R	0x0

Table 24-1337. Register Call Summary for Register SL_TX_PAUSE

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 24-1338. SL_EMCONTROL

Address Offset	0x0000 0020		
Physical Address	0x4848 4DA0 0x4848 4DE0	**Instance**	SL1 SL2
Description	CPGMAC_SL emulation control register		
Type	RW		

31–2	1	0
RESERVED	SOFT	FREE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	SOFT	Emulation Soft Bit. Emulation soft bit. This bit is used in conjunction with FREE bit to determine the emulation suspend mode. This bit has no effect if FREE = 1.	RW	0x0
0	FREE	Emulation Free Bit. Emulation free bit. This bit is used in conjunction with SOFT bit to determine the emulation suspend mode.	RW	0x0

Table 24-1339. Register Call Summary for Register SL_EMCONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Emulation Control: [0]
- SL Register Summary: [1]

Table 24-1340. SL_RX_PRI_MAP

Address Offset	0x0000 0024		
Physical Address	0x4848 4DA4 0x4848 4DE4	**Instance**	SL1 SL2
Description	CPGMAC_SL RX packet priority to header priority mapping register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI7			RESERVED	PRI6			RESERVED	PRI5			RESERVED	PRI4			RESERVED	PRI3			RESERVED	PRI2			RESERVED	PRI1			RESERVED	PRI0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet priority of 0x7 is mapped (changed) to this value.	RW	0x7
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet priority of 0x6 is mapped (changed) to this value.	RW	0x6
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet priority of 0x5 is mapped (changed) to this value.	RW	0x5
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet priority of 0x4 is mapped (changed) to this value.	RW	0x4
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet priority of 0x3 is mapped (changed) to this value.	RW	0x3
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet priority of 0x2 is mapped (changed) to this value.	RW	0x2
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet priority of 0x1 is mapped (changed) to this value.	RW	0x1
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet priority of 0x0 is mapped (changed) to this value.	RW	0x0

Table 24-1341. Register Call Summary for Register SL_RX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]

Table 24-1342. SL_TX_GAP

Address Offset	0x0000 0028		
Physical Address	0x4848 4DA8 0x4848 4DE8	**Instance**	SL1 SL2
Description	Transmit inter-packet gap register		
Type	RW		

31:9	8:0
RESERVED	TX_GAP

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	TX_GAP	Transmit Inter-Packet Gap	RW	0xC

Table 24-1343. Register Call Summary for Register SL_TX_GAP

Gigabit Ethernet Switch (GMAC_SW)

- Ethernet MAC Sliver (CPGMAC_SL): [0]
- SL Register Summary: [1]

24.11.6.10 MDIO registers

24.11.6.10.1 MDIO Register Summary

Table 24-1344. MDIO Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MDIO Physical Address
MDIO_VER	RW	32	0x0000 0000	0x4848 5000
MDIO_CONTROL	RW	32	0x0000 0004	0x4848 5004
MDIO_ALIVE	RW	32	0x0000 0008	0x4848 5008
MDIO_LINK	R	32	0x0000 000C	0x4848 500C
MDIO_LINKINTRAW	RW	32	0x0000 0010	0x4848 5010
MDIO_LINKINTMASKED	RW	32	0x0000 0014	0x4848 5014
MDIO_USERINTRAW	RW	32	0x0000 0020	0x4848 5020
MDIO_USERINTMASKED	RW	32	0x0000 0024	0x4848 5024
MDIO_USERINTMASKSET	RW	32	0x0000 0028	0x4848 5028
MDIO_USERINTMASKCLR	RW	32	0x0000 002C	0x4848 502C
MDIO_USERACCESS0	RW	32	0x0000 0080	0x4848 5080
MDIO_USERPHYSEL0	RW	32	0x0000 0084	0x4848 5084
MDIO_USERACCESS1	RW	32	0x0000 0088	0x4848 5088
MDIO_USERPHYSEL1	RW	32	0x0000 008C	0x4848 508C

24.11.6.10.2 MDIO Register Description

Table 24-1345. MDIO_VER

Address Offset	0x0000 0000		
Physical Address	0x4848 5000	**Instance**	MDIO
Description	MDIO Revision		
Type	RW		

31:0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	MDIO revision value	RW	0x-

Table 24-1346. Register Call Summary for Register MDIO_VER

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Register Summary: [0]

Table 24-1347. MDIO_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 5004	**Instance**	MDIO
Description	MDIO Control register		
Type	RW		

31	30	29	28:24	23:21	20	19	18	17	16	15:0
IDLE	ENABLE	RESERVED	HIGHEST_USER_CHANNEL	RESERVED	PREAMBLE	FAULT	FAULTENB	INTTESTENB	RESERVED	CLKDIV

Bits	Field Name	Description	Type	Reset
31	IDLE	MDIO state machine IDLE. Set to 1 when the state machine is in the idle state. 0: State machine is not in idle state. 1: State machine is in idle state.	R	0x0
30	ENABLE	Enable control. If the MDIO state machine is active at the time it is disabled, it will complete the current operation before halting and setting the IDLE bit. If using byte access, the ENABLE bit has to be the last bit written in this register. 0: Disables the MDIO state machine. 1: Enable the MDIO state machine.	RW	0x0
29	RESERVED		R	0x0
28:24	HIGHEST_USER_CHANNEL	Highest user channel. This field specifies the highest user access channel that is available in the module and is currently set to 1. This implies that the MDIO_USERACCESS1 register is the highest available user access channel.	R	0x0
23:21	RESERVED		R	0x0
20	PREAMBLE	Preamble disable. 0: Standard MDIO preamble is used. 1: Disables this device from sending MDIO frame preambles.	RW	0x0
19	FAULT	Fault indicator. This bit is set to 1 if the MDIO pins fail to read back what the device is driving onto them. This indicates a physical layer fault and the module state machine is reset. Writing a 1 to it clears this bit. 0: No failure. 1: Physical layer fault; the MDIO state machine is reset.	RW	0x0
18	FAULTENB	Fault detect enable. This bit has to be set to 1 to enable the physical layer fault detection. 0: Disables the physical layer fault detection. 1: Enables the physical layer fault detection.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INTTESTENB	Interrupt test enable. This bit can be set to 1 to enable the host to set the USERINT and LINKINT bits for test purposes. 0: Interrupt bits are not set. 1: Enables the host to set the USERINT and LINKINT bits for test purposes.	RW	0x0
16	RESERVED		R	0x0
15:0	CLKDIV	Clock divider. This field specifies the division ratio between ICLK and the frequency of MDCLK. MDCLK is disabled when CLKDIV is set to 0. MDCLK frequency = ICLK frequency/(CLKDIV+1).	RW	0x0

Table 24-1348. Register Call Summary for Register MDIO_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Interface Clocking: [0]
- MDIO Functional Description: [1]
- Initializing the MDIO Module: [2] [3]
- MDIO Register Summary: [4]
- MDIO Register Description: [5] [6] [7]

Table 24-1349. MDIO_ALIVE

Address Offset	0x0000 0008		
Physical Address	0x4848 5008	**Instance**	MDIO
Description	PHY Alive Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIVE																															

Bits	Field Name	Description	Type	Reset
31:0	ALIVE	MDIO alive. Each of the 32 bits of this register is set if the most recent access to the PHY with address corresponding to the register bit number was acknowledged by the PHY, the bit is reset if the PHY fails to acknowledge the access. Both the user and polling accesses to a PHY will cause the corresponding alive bit to be updated. The alive bits are only meant to be used to give an indication of the presence or not of a PHY with the corresponding address. Writing a 1 to any bit will clear it, writing a 0 has no effect.	RW	0x0

Table 24-1350. Register Call Summary for Register MDIO_ALIVE

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0] [1]
- Initializing the MDIO Module: [2]
- MDIO Register Summary: [3]

Table 24-1351. MDIO_LINK

Address Offset	0x0000 000C		
Physical Address	0x4848 500C	**Instance**	MDIO
Description	PHY Link Status		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LINK

Bits	Field Name	Description	Type	Reset
31:0	LINK	MDIO link state. This register is updated after a read of the Generic Status Register of a PHY. The bit is set if the PHY with the corresponding address has link and the PHY acknowledges the read transaction. The bit is reset if the PHY indicates it does not have link or fails to acknowledge the read transaction. Writes to the register have no effect. In addition, the status of the two PHYs specified in the MDIO_USERPHYSEL registers can be determined using the MLINK input pins (NOT PINNED OUT). This is determined by the LINKSEL bit in the MDIO_USERPHYSEL register.	R	0x0

Table 24-1352. Register Call Summary for Register MDIO_LINK

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0] [1] [2] [3]
- MDIO Functional Description: [4]
- Initializing the MDIO Module: [5]
- MDIO Register Summary: [6]

Table 24-1353. MDIO_LINKINTRAW

Address Offset	0x0000 0010		
Physical Address	0x4848 5010	**Instance**	MDIO
Description			
Type	RW		

31:2	1:0
RESERVED	LINKINTRAW

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	LINKINTRAW	MDIO link change event, raw value.	RW	0x0

Table 24-1354. Register Call Summary for Register MDIO_LINKINTRAW

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0]
- MDIO Register Summary: [1]

Table 24-1355. MDIO_LINKINTMASKED

Address Offset	0x0000 0014		
Physical Address	0x4848 5014	**Instance**	MDIO
Description	MDIO Link Status Change Interrupt Register		
Type	RW		

31:2	1:0
RESERVED	LINKINTMASKED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	LINKINTMASKED	MDIO link change interrupt, masked value. When asserted 1, a bit indicates that there was an MDIO link change event (i.e. change in the MDIO Link register) corresponding to the PHY address in the MDIO_USERPHYSEL register and the corresponding LINKINTENB bit was set. LINKINTMASKED[0] and LINKINTMASKED[1] correspond to MDIO_USERPHYSEL0 and MDIO_USERPHYSEL1, respectively. Writing a 1 will clear the interrupt and writing 0 has no effect. If the INTTESTENB bit in the MDIO_CONTROL register is set, the host may set the LINKINT bits to a 1. This mode may be used for test purposes.	RW	0x0

Table 24-1356. Register Call Summary for Register MDIO_LINKINTMASKED

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0] [1]
- MDIO Functional Description: [2]
- MDIO Register Summary: [3]

Table 24-1357. MDIO_USERINTRAW

Address Offset	0x0000 0020		
Physical Address	0x4848 5020	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt		
Type	RW		

31–2	1–0
RESERVED	USERINTRAW

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTRAW	Raw value of MDIO user command complete event for the MDIO_USERACCESS1 and MDIO_USERACCESS0 register, respectively. When asserted 1, a bit indicates that the previously scheduled PHY read or write command using that particular MDIO_USERACCESS register has completed. Writing a 1 will clear the event and writing 0 has no effect. If the INTTESTENB bit in the MDIO_CONTROL register is set, the host may set the USERINTRAW bits to a 1. This mode may be used for test purposes.	RW	0x0

Table 24-1358. Register Call Summary for Register MDIO_USERINTRAW

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0]
- Writing Data To a PHY Register: [1]
- Reading Data From a PHY Register: [2]
- MDIO Register Summary: [3]

Table 24-1359. MDIO_USERINTMASKED

Address Offset	0x0000 0024		
Physical Address	0x4848 5024	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt		

Table 24-1359. MDIO_USERINTMASKED (continued)

Type	RW

31–2	1–0
RESERVED	USERINTMASKED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTMASKED	Masked value of MDIO user command complete interrupt for the MDIO_USERACCESS1 and MDIO_USERACCESS0 register, respectively. When asserted 1, a bit indicates that the previously scheduled PHY read or write command using that particular MDIO_USERACCESS register has completed and the corresponding USERINTMASKSET bit is set to 1. Writing a 1 will clear the interrupt and writing 0 has no effect. If the INTTESTENB bit in the MDIO_CONTROL register is set, the host may set the USERINTMASKED bits to a 1. This mode may be used for test purposes.	RW	0x0

Table 24-1360. Register Call Summary for Register MDIO_USERINTMASKED

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0] [1]
- MDIO Functional Description: [2]
- Writing Data To a PHY Register: [3]
- Reading Data From a PHY Register: [4]
- MDIO Register Summary: [5]

Table 24-1361. MDIO_USERINTMASKSET

Address Offset	0x0000 0028		
Physical Address	0x4848 5028	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt Mask Set		
Type	RW		

31–2	1–0
RESERVED	USERINTMASKSET

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTMASKSET	MDIO user interrupt mask set for USERINTMASKED[1:0], respectively. Writing a bit to 1 will enable MDIO user command complete interrupts for that particular MDIO_USERACCESS register. MDIO user interrupt for a particular MDIO_USERACCESS register is disabled if the corresponding bit is 0. Writing a 0 to this register has no effect.	RW	0x0

Table 24-1362. Register Call Summary for Register MDIO_USERINTMASKSET

Gigabit Ethernet Switch (GMAC_SW)
- Miscellaneous Pulse Interrupt (MISC_PULSE): [0]
- MDIO Functional Description: [1]
- Initializing the MDIO Module: [2]
- Writing Data To a PHY Register: [3]
- Reading Data From a PHY Register: [4]
- MDIO Register Summary: [5]

Table 24-1363. MDIO_USERINTMASKCLR

Address Offset	0x0000 002C		
Physical Address	0x4848 502C	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt Mask Clear		
Type	RW		

31:2	1:0
RESERVED	USERINTMASKCLEAR

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTMASKCLEAR	MDIO user command complete interrupt mask clear for USERINTMASKED[1:0], respectively. Writing a bit to 1 will disable further user command complete interrupts for that particular MDIO_USERACCESS register. Writing a 0 to this register has no effect.	RW	0x0

Table 24-1364. Register Call Summary for Register MDIO_USERINTMASKCLR

Gigabit Ethernet Switch (GMAC_SW)
- MDIO Functional Description: [0]
- MDIO Register Summary: [1]

Table 24-1365. MDIO_USERACCESS0

Address Offset	0x0000 0080		
Physical Address	0x4848 5080	**Instance**	MDIO
Description	MDIO_User_Access		
Type	RW		

31	30	29	28:26	25:21	20:16	15:0
GO	WRITE	ACK	RESERVED	REGADR	PHYADR	DATA

Bits	Field Name	Description	Type	Reset
31	GO	Go. Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIO_USERACCESS0 register are blocked when the GO bit is 1. If byte access is being used, the GO bit should be written last.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	WRITE	Write enable. Setting this bit to a 1 causes the MDIO transaction to be a register write, otherwise it is a register read.	RW	0x0
29	ACK	Acknowledge. This bit is set if the PHY acknowledged the read transaction.	RW	0x0
28:26	RESERVED		R	0x0
25:21	REGADR	Register address. Specifies the PHY register to be accessed for this transaction.	RW	0x0
20:16	PHYADR	PHY address. Specifies the PHY to be accesses for this transaction.	RW	0x0
15:0	DATA	User data. The data value read from or to be written to the specified PHY register.	RW	0x0

Table 24-1366. Register Call Summary for Register MDIO_USERACCESS0

Gigabit Ethernet Switch (GMAC_SW)

- MDIO: [0]
- MDIO Functional Description: [1] [2] [3] [4] [5]
- Initializing the MDIO Module: [6] [7] [8]
- Writing Data To a PHY Register: [9] [10] [11] [12] [13]
- Reading Data From a PHY Register: [14] [15] [16] [17] [18]
- MDIO Register Summary: [19]
- MDIO Register Description: [20] [21] [22]

Table 24-1367. MDIO_USERPHYSEL0

Address Offset	0x0000 0084		
Physical Address	0x4848 5084	**Instance**	MDIO
Description	MDIO User PHY Select		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td>LINKSEL</td><td>LINKINTENB</td><td>RESERVED</td><td colspan="5">PHYADDRMON</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	LINKSEL	Link status determination select. Set to 1 to determine link status using the MLINK pin (NOT PINNED OUT). Default value is 0 which implies that the link status is determined by the MDIO state machine.	RW	0x0
6	LINKINTENB	Link change interrupt enable. Set to 1 to enable link change status interrupts for PHY address specified in PHYADDRMON. Link change interrupts are disabled if this bit is set to 0. 0: Link change interrupts are disabled. 1: Link change status interrupts for PHY address specified in PHYADDRMON bits are enabled.	RW	0x0
5	RESERVED		R	0x0
4:0	PHYADDRMON	PHY address whose link status is to be monitored.	RW	0x0

Table 24-1368. Register Call Summary for Register MDIO_USERPHYSEL0

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0] [1]
- Initializing the MDIO Module: [2]
- MDIO Register Summary: [3]
- MDIO Register Description: [4]

Table 24-1369. MDIO_USERACCESS1

Address Offset	0x0000 0088		
Physical Address	0x4848 5088	**Instance**	MDIO
Description	MDIO User Access		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GO	WRITE	ACK	RESERVED			REGADR					PHYADR					DATA															

Bits	Field Name	Description	Type	Reset
31	GO	Go. Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIO_USERACCESS1 register are blocked when the GO bit is 1. If byte access is being used, the GO bit should be written last.	RW	0x0
30	WRITE	Write enable. Setting this bit to a 1 causes the MDIO transaction to be a register write, otherwise it is a register read.	RW	0x0
29	ACK	Acknowledge. This bit is set if the PHY acknowledged the read transaction.	RW	0x0
28:26	RESERVED		R	0x0
25:21	REGADR	Register address. Specifies the PHY register to be accessed for this transaction.	RW	0x0
20:16	PHYADR	PHY address. Specifies the PHY to be accesses for this transaction.	RW	0x0
15:0	DATA	User data. The data value read from or to be written to the specified PHY register.	RW	0x0

Table 24-1370. Register Call Summary for Register MDIO_USERACCESS1

Gigabit Ethernet Switch (GMAC_SW)

- MDIO: [0]
- MDIO Register Summary: [1]
- MDIO Register Description: [2] [3] [4] [5]

Table 24-1371. MDIO_USERPHYSEL1

Address Offset	0x0000 008C		
Physical Address	0x4848 508C	**Instance**	MDIO
Description	MDIO User PHY Select		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LINKSEL	LINKINTENB	RESERVED	PHYADDRMON

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	LINKSEL	Link status determination select. Set to 1 to determine link status using the MLINK pin (NOT PINNED OUT). Default value is 0 which implies that the link status is determined by the MDIO state machine.	RW	0x0
6	LINKINTENB	Link change interrupt enable. Set to 1 to enable link change status interrupts for PHY address specified in PHYADDRMON. Link change interrupts are disabled if this bit is set to 0. 0: Link change interrupts are disabled. 1: Link change status interrupts for PHY address specified in PHYADDRMON bits are enabled.	RW	0x0
5	RESERVED		R	0x0
4:0	PHYADDRMON	PHY address whose link status is to be monitored.	RW	0x0

Table 24-1372. Register Call Summary for Register MDIO_USERPHYSEL1

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Register Summary: [0]
- MDIO Register Description: [1]

24.11.6.11 WR registers

24.11.6.11.1 WR Register Summary

Table 24-1373. WR Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WR Physical Address
WR_IDVER	R	32	0x0000 0000	0x4848 5200
WR_SOFT_RESET	RW	32	0x0000 0004	0x4848 5204
WR_CONTROL	RW	32	0x0000 0008	0x4848 5208
WR_INT_CONTROL	RW	32	0x0000 000C	0x4848 520C
WR_C0_RX_THRESH_EN	RW	32	0x0000 0010	0x4848 5210
WR_C0_RX_EN	RW	32	0x0000 0014	0x4848 5214
WR_C0_TX_EN	RW	32	0x0000 0018	0x4848 5218
WR_C0_MISC_EN	RW	32	0x0000 001C	0x4848 521C
WR_C0_RX_THRESH_STAT	R	32	0x0000 0040	0x4848 5240
WR_C0_RX_STAT	R	32	0x0000 0044	0x4848 5244
WR_C0_TX_STAT	R	32	0x0000 0048	0x4848 5248
WR_C0_MISC_STAT	R	32	0x0000 004C	0x4848 524C
WR_C0_RX_IMAX	RW	32	0x0000 0070	0x4848 5270
WR_C0_TX_IMAX	RW	32	0x0000 0074	0x4848 5274
WR_RGMII_CTL	R	32	0x0000 0088	0x4848 5288
WR_STATUS	R	32	0x0000 008C	0x4848 528C

24.11.6.11.2 WR Register Description

Table 24-1374. WR_IDVER

Address Offset	0x0000 0000

Table 24-1374. WR_IDVER (continued)

Physical Address	0x4848 5200	**Instance**	WR
Description	Subsystem wrapper revision register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REVISION</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Wrapper revision value	R	0x-

Table 24-1375. Register Call Summary for Register WR_IDVER

Gigabit Ethernet Switch (GMAC_SW)
- WR Register Summary: [0]

Table 24-1376. WR_SOFT_RESET

Address Offset	0x0000 0004		
Physical Address	0x4848 5204	**Instance**	WR
Description	Subsystem soft reset register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="31">RESERVED</td><td>SOFT_RESET</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the CPGMACSS_R logic to be reset (INT, REGS, CPPI). Software reset occurs on the clock following the register bit write.	RW	0x0

Table 24-1377. Register Call Summary for Register WR_SOFT_RESET

Gigabit Ethernet Switch (GMAC_SW)
- Software Reset: [0]
- WR Register Summary: [1]

Table 24-1378. WR_CONTROL

Address Offset	0x0000 0008		
Physical Address	0x4848 5208	**Instance**	WR
Description	Subsystem control register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td>SS_EEE_EN</td><td colspan="4"></td><td colspan="2">MMR_STDBYMODE</td><td colspan="2">MMR_IDLEMODE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8	SS_EEE_EN	Subsystem Energy Efficient Ethernet enable 0: EEE disabled 1: EEE enabled	RW	0x0
7:4	RESERVED	Reserved	R	0x0
3:2	MMR_STDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode : Local initiator is unconditionally placed in standbystate. 0x1: No-standby mode : Local initiator is unconditionally placed out of standby state. 0x3: Reserved : Reserved. 0x2: Reserved : Reserved.	RW	0x0
1:0	MMR_IDLEMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode : Local initiator is unconditionally placed in idle state. 0x1: No-idle mode : Local initiator is unconditionally placed out of idle state. 0x3: Reserved : Reserved. 0x2: Reserved : Reserved.	RW	0x0

Table 24-1379. Register Call Summary for Register WR_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Energy Efficient Ethernet Support (802.3az): [0]
- WR Register Summary: [1]

Table 24-1380. WR_INT_CONTROL

Address Offset	0x0000 000C		
Physical Address	0x4848 520C	**Instance**	WR
Description	Subsystem interrupt control		
Type	RW		

31	30:22	21:16	15:12	11:0
INT_TEST	RESERVED	INT_PACE_EN	RESERVED	INT_PRESCALE

Bits	Field Name	Description	Type	Reset
31	INT_TEST	Interrupt Test - Test bit to the interrupt pacing blocks	RW	0x0
30:22	RESERVED		R	0x0
21:16	INT_PACE_EN	Interrupt Pacing Enable INT_PACE_EN[0] – Enables RX_PULSE Pacing (0 is pacing bypass) INT_PACE_EN[1] – Enables TX_PULSE Pacing (0 is pacing bypass)	RW	0x0
15:12	RESERVED		R	0x0
11:0	INT_PRESCALE	Interrupt Counter Prescaler - The number of MAIN_CLK periods in 4 µs.	RW	0x0

Table 24-1381. Register Call Summary for Register WR_INT_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Interrupt Pacing: [0]
- WR Register Summary: [1]

Table 24-1382. WR_C0_RX_THRESH_EN

Address Offset	0x0000 0010		
Physical Address	0x4848 5210	**Instance**	WR
Description	Subsystem core 0 receive threshold int enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_RX_THRESH_EN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_THRESH_EN	Core 0 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on RX_THRESH_PULSE.	RW	0x0

Table 24-1383. Register Call Summary for Register WR_C0_RX_THRESH_EN

Gigabit Ethernet Switch (GMAC_SW)

- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [0] [1]
- WR Register Summary: [2]

Table 24-1384. WR_C0_RX_EN

Address Offset	0x0000 0014		
Physical Address	0x4848 5214	**Instance**	WR
Description	Subsystem core 0 receive interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_RX_EN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_EN	Core 0 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on RX_PULSE.	RW	0x0

Table 24-1385. Register Call Summary for Register WR_C0_RX_EN

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0] [1]
- WR Register Summary: [2]

Table 24-1386. WR_C0_TX_EN

Address Offset	0x0000 0018		
Physical Address	0x4848 5218	**Instance**	WR
Description	Subsystem core 0 transmit interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	C0_TX_EN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_TX_EN	Core 0 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on TX_PULSE.	RW	0x0

Table 24-1387. Register Call Summary for Register WR_C0_TX_EN

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0] [1]
- WR Register Summary: [2]

Table 24-1388. WR_C0_MISC_EN

Address Offset	0x0000 001C		
Physical Address	0x4848 521C	**Instance**	WR
Description	Subsystem core 0 misc interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											C0_MISC_EN				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	C0_MISC_EN	Core 0 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (SPF2_PEND, SPF1_PEND, EVNT_PEND, STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT) that is enabled to generate an interrupt on MISC_PULSE.	RW	0x0

Table 24-1389. Register Call Summary for Register WR_C0_MISC_EN

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0]
- WR Register Summary: [1]

Table 24-1390. WR_C0_RX_THRESH_STAT

Address Offset	0x0000 0040		
Physical Address	0x4848 5240	**Instance**	WR
Description	Subsystem core 0 rx threshold masked int status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_RX_THRESH_STAT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_THRESH_STAT	Core 0 Receive Threshold Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on RX_THRESH_PULSE.	R	0x0

Table 24-1391. Register Call Summary for Register WR_C0_RX_THRESH_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [0]
- WR Register Summary: [1]

Table 24-1392. WR_C0_RX_STAT

Address Offset	0x0000 0044		
Physical Address	0x4848 5244	**Instance**	WR
Description	Subsystem core 0 rx interrupt masked int status register		
Type	R		

31–8	7–0
RESERVED	C0_RX_STAT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_STAT	Core 0 Receive Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on RX_PULSE.	R	0x0

Table 24-1393. Register Call Summary for Register WR_C0_RX_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0] [1]
- WR Register Summary: [2]

Table 24-1394. WR_C0_TX_STAT

Address Offset	0x0000 0048		
Physical Address	0x4848 5248	**Instance**	WR
Description	Subsystem core 0 tx interrupt masked int status register		
Type	R		

31–8	7–0
RESERVED	C0_TX_STAT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_TX_STAT	Core 0 Transmit Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on TX_PULSE .	R	0x0

Table 24-1395. Register Call Summary for Register WR_C0_TX_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- WR Register Summary: [1]

Table 24-1396. WR_C0_MISC_STAT

Address Offset	0x0000 004C		
Physical Address	0x4848 524C	**Instance**	WR
Description	Subsystem core 0 misc interrupt masked int status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											C0_MISC_STAT				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	C0_MISC_STAT	Core 0 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (SPF2_PEND, SPF1_PEND, EVNT_PEND, STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT) that is enabled and generating an interrupt on MISC_PULSE .	R	0x0

Table 24-1397. Register Call Summary for Register WR_C0_MISC_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0] [1]
- WR Register Summary: [2]

Table 24-1398. WR_C0_RX_IMAX

Address Offset	0x0000 0070		
Physical Address	0x4848 5270	**Instance**	WR
Description	Subsystem core 0 receive interrupts per millisecond		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										C0_RX_IMAX					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	C0_RX_IMAX	Core 0 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on RX_PULSE if pacing is enabled for this interrupt.	RW	0x0

Table 24-1399. Register Call Summary for Register WR_C0_RX_IMAX

Gigabit Ethernet Switch (GMAC_SW)

- Interrupt Pacing: [0]
- WR Register Summary: [1]

Table 24-1400. WR_C0_TX_IMAX

Address Offset	0x0000 0074		
Physical Address	0x4848 5274	**Instance**	WR
Description	Subsystem core 0 transmit interrupts per millisecond		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										C0_TX_IMAX					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	C0_TX_IMAX	Core 0 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on TX_PULSE if pacing is enabled for this interrupt.	RW	0x0

Table 24-1401. Register Call Summary for Register WR_C0_TX_IMAX

Gigabit Ethernet Switch (GMAC_SW)

- Interrupt Pacing: [0]
- WR Register Summary: [1]

Table 24-1402. WR_RGMII_CTL

Address Offset	0x0000 0088		
Physical Address	0x4848 5288	**Instance**	WR
Description	RGMII control signal register		
Type	R		

31:8	7	6:5	4	3	2:1	0
RESERVED	RGMII2_FULLDUPLEX	RGMII2_SPEED	RGMII2_LINK	RGMII1_FULLDUPLEX	RGMII1_SPEED	RGMII1_LINK

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	RGMII2_FULLDUPLEX	RGMII 2 Fullduplex - This is the CPRGMII fullduplex output signal. 0 - Half-duplex mode 1 - Full-duplex mode	R	0x0
6:5	RGMII2_SPEED	RGMII2 Speed - This is the CPRGMII speed output signal 0x0 - 10Mbps mode 0x1 - 100Mbps mode 0x2 - 1000Mbps (gig) mode 0x3 - reserved	R	0x0
4	RGMII2_LINK	RGMII2 Link Indicator - This is the CPRGMII link output signal 0 - RGMII2 link is down 1 - RGMII2 link is up	R	0x0
3	RGMII1_FULLDUPLEX	RGMII1 Fullduplex - This is the CPRGMII fullduplex output signal. 0 - Half-duplex mode 1 - Full-duplex mode	R	0x0
2:1	RGMII1_SPEED	RGMII1 Speed - This is the CPRGMII speed output signal 0x0 - 10Mbps mode 0x1 - 100Mbps mode 0x2 - 1000Mbps (gig) mode 0x3 - reserved	R	0x0
0	RGMII1_LINK	RGMII1 Link Indicator - This is the CPRGMII link output signal 0 - RGMII1 link is down 1 - RGMII1 link is up	R	0x0

Table 24-1403. Register Call Summary for Register WR_RGMII_CTL

Gigabit Ethernet Switch (GMAC_SW)

- WR Register Summary: [0]

Table 24-1404. WR_STATUS

Address Offset	0x0000 008C		
Physical Address	0x4848 528C	**Instance**	WR
Description	Subsystem Status register		
Type	R		

31–3	2	1	0
RESERVED	SPF2_CLKSTOP_ACK	SPF1_CLKSTOP_ACK	EEE_CLKSTOP_ACK

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	SPF2_CLKSTOP_ACK	SPF2 Clockstop Acknowledge – When asserted the subsystem gated clock is not turned on due to SPF2.	R	0x0
1	SPF1_CLKSTOP_ACK	SPF1 Clockstop Acknowledge – When asserted the subsystem gated clock is not turned on due to SPF1.	R	0x0
0	EEE_CLKSTOP_ACK	CPSW_3G Clockstop Acknowledge – When asserted the subsystem gated clock is not turned on due to the CPSW_3G.	R	0x0

Table 24-1405. Register Call Summary for Register WR_STATUS

Gigabit Ethernet Switch (GMAC_SW)

- WR Register Summary: [0]

24.11.6.12 SPF Registers

24.11.6.12.1 SPF Register Summary

Table 24-1406. SPF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SPF1 Physical Address	SPF2 Physical Address
SPF_IDVER	R	32	0x0000 0000	0x4848 5C00	0x4848 5E00
SPF_STATUS	RW	32	0x0000 0004	0x4848 5C04	0x4848 5E04
SPF_CONTROL	RW	32	0x0000 0008	0x4848 5C08	0x4848 5E08
SPF_DROPCOUNT	R	32	0x0000 000C	0x4848 5C0C	0x4848 5E0C
SPF_SWRESET	RW	32	0x0000 0010	0x4848 5C10	0x4848 5E10
SPF_PRESCALE	RW	32	0x0000 0014	0x4848 5C14	0x4848 5E14
SPF_RATELIMi (1)	RW	32	0x0000 0018 + (i * 4)	0x4848 5C18 + (i * 4)	0x4848 5E18 + (i * 4)
SPF_CONSTj (2)	RW	32	0x0000 001C + (j * 4)	0x4848 5C1C + (j * 4)	0x4848 5E1C + (j * 4)
SPF_INSTRW2	RW	32	0x0000 0050	0x4848 5C50	0x4848 5E50
SPF_INSTRW1	RW	32	0x0000 0054	0x4848 5C54	0x4848 5E54
SPF_INSTRW0	RW	32	0x0000 0058	0x4848 5C58	0x4848 5E58
SPF_INSTR_CTL	RW	32	0x0000 005C	0x4848 5C5C	0x4848 5E5C
SPF_LOG_BEGIN	RW	32	0x0000 0060	0x4848 5C60	0x4848 5E60
SPF_LOG_END	RW	32	0x0000 0064	0x4848 5C64	0x4848 5E64
SPF_LOG_HWPTR	R	32	0x0000 0068	0x4848 5C68	0x4848 5E68
SPF_LOG_SWPTR	RW	32	0x0000 006C	0x4848 5C6C	0x4848 5E6C

Table 24-1406. SPF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	SPF1 Physical Address	SPF2 Physical Address
SPF_LOG_MAP0	RW	32	0x0000 0070	0x4848 5C70	0x4848 5E70
SPF_LOG_MAP1	RW	32	0x0000 0074	0x4848 5C74	0x4848 5E74
SPF_LOG_THRESHk (3)	RW	32	0x0000 0078 + (k * 4)	0x4848 5C78 + (k * 4)	0x4848 5E78 + (k * 4)
SPF_INTCNT	RW	32	0x0000 009C	0x4848 5C9C	0x4848 5E9C
SPF_INT_RAW	RW	32	0x0000 00A0	0x4848 5CA0	0x4848 5EA0
SPF_INT_MASKED	RW	32	0x0000 00A4	0x4848 5CA4	0x4848 5EA4
SPF_MASK_SET	RW	32	0x0000 00A8	0x4848 5CA8	0x4848 5EA8
SPF_MASK_CLR	RW	32	0x0000 00AC	0x4848 5CAC	0x4848 5EAC

(1) i = 0 to 3
(2) j = 0 to 7
(3) k = 0 to 8

24.11.6.12.2 SPF Register Description

Table 24-1407. SPF_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 5C00 0x4848 5E00	**Instance**	SPF1 SPF2
Description	SPF revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	SPF revision value	R	0x-

Table 24-1408. Register Call Summary for Register SPF_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 24-1409. SPF_STATUS

Address Offset	0x0000 0004		
Physical Address	0x4848 5C04 0x4848 5E04	**Instance**	SPF1 SPF2
Description	Status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SPF_BUSY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_BUSY	SPF is Busy/Idle, Busy Packet processing or logging in progress.	RW	0x0

Table 24-1410. Register Call Summary for Register SPF_STATUS

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 24-1411. SPF_CONTROL

Address Offset	0x0000 0008		
Physical Address	0x4848 5C08 0x4848 5E08	**Instance**	SPF1 SPF2
Description	SPF control register		
Type	RW		

31:10	9	8	7:4	3	2	1	0
RESERVED	SPF_LOGOW_EN	SPF_LOG_EN	RESERVED	SPF_RULE_LOG	SPF_EXT_BYPASS	SPF_DROP	SPF_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SPF_LOGOW_EN	SPF Log Overwrite Enable. Setting this bit will cause SPF to overwrite previously logged data whether or not software has updated the software_working_pointer. Overwriting only occurs if there is new data but no space to write it in the space indicated by log_start_address and log_end_address.	RW	0x0
8	SPF_LOG_EN	SPF Log Enable. Setting this bit will allow SPF to log information about dropped packets to memory.	RW	0x0
7:4	RESERVED		R	0x0
3	SPF_RULE_LOG	SPF Rule Engine Log Enable. Setting this bit will allow SPF to log data from rule engine. The default is log data from extractor.	RW	0x0
2	SPF_EXT_BYPASS	SPF Extractor Bypass Enable. The extractor will not provide any offset information to rule engine if this bit is set. The rule engine must load each of the base registers it intends to use to determine if the packet should be discarded.	RW	0x0
1	SPF_DROP	SPF Drop Enable. This bit must be set to activate packet drops.	RW	0x0
0	SPF_ENABLE	SPF Enable. This bit must be set to enable any operation in SPF. The SPF instruction memory can only be accessed by host processor when the spf_enable is deasserted. Once spf_enable is set, writing a zero to this bit will only take effect when spf_busy signal is low. This ensures that spf stops only on packet boundaries.	RW	0x0

Table 24-1412. Register Call Summary for Register SPF_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0] [1] [2] [3]
- Programming Guide: [4]
- SPF Register Summary: [5]

Table 24-1413. SPF_DROPCOUNT

Address Offset	0x0000 000C

Table 24-1413. SPF_DROPCOUNT (continued)

Physical Address	0x4848 5C0C 0x4848 5E0C	**Instance**	SPF1 SPF2
Description	Drop Count Register		
Type	R		

31–24	23–0
RESERVED	SPF_DROPCNT

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	SPF_DROPCNT	SPF Drop counter indicates the number of packets dropped so far. This counter does not roll over and must be cleared by writing 0x00FFFFFF.	R	0x0

Table 24-1414. Register Call Summary for Register SPF_DROPCOUNT

Gigabit Ethernet Switch (GMAC_SW)
- SPF Register Summary: [0]

Table 24-1415. SPF_SWRESET

Address Offset	0x0000 0010		
Physical Address	0x4848 5C10 0x4848 5E10	**Instance**	SPF1 SPF2
Description	Software Reset Register		
Type	RW		

31–1	0
RESERVED	SPF_SWRST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_SWRST	SPF Software reset bit can be set to initiate a software reset. It stays high until the reset has not completed, this reset clears all registers to default value.	RW	0x0

Table 24-1416. Register Call Summary for Register SPF_SWRESET

Gigabit Ethernet Switch (GMAC_SW)
- SPF Register Summary: [0]

Table 24-1417. SPF_PRESCALE

Address Offset	0x0000 0014		
Physical Address	0x4848 5C14 0x4848 5E14	**Instance**	SPF1 SPF2
Description	Rate Limit Prescale Register		
Type	RW		

31–20	19–0
RESERVED	SPF_PRESCALE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	SPF_PRESCALE	The MAIN clock is divided by this value for use in Rate Limiters. It is used to create rolling time intervals for use in rate limiting feature.	RW	0x0

Table 24-1418. Register Call Summary for Register SPF_PRESCALE

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0] [1]
- Programming Guide: [2]
- SPF Register Summary: [3]
- SPF Register Description: [4] [5]

Table 24-1419. SPF_RATELIMi

Address Offset	0x0000 0018 + (i * 4)	**Index**	i = 0 to 3
Physical Address	0x4848 5C18 + (i * 4) 0x4848 5E18 + (i * 4)	**Instance**	SPF1 SPF2
Description	Rate Limit Register		
Type	RW		

31–8	7–0
RESERVED	SPF_RATELIM

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	SPF_RATELIM	SPF Rate Limit Register. The number of packets corresponding to a filter that will be allowed per unit time interval. The filters are programmed in the rule engine and time interval is determined by the SPF_PRESCALE register.	RW	0x0

Table 24-1420. Register Call Summary for Register SPF_RATELIMi

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0] [1]
- Programming Guide: [2]
- SPF Register Summary: [3]

Table 24-1421. SPF_CONSTj

Address Offset	0x0000 0028 + (j * 4)	**Index**	j = 0 to 7
Physical Address	0x4848 5C1C + (j * 4) 0x4848 5E1C + (j * 4)	**Instance**	SPF1 SPF2
Description	Constant Register		
Type	RW		

31–0
SPF_CONST

Bits	Field Name	Description	Type	Reset
31:0	SPF_CONST	SPF Constant Register. The contents of this register are used as input to any instruction that references it.	RW	0x0

Table 24-1422. Register Call Summary for Register SPF_CONSTj

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- Programming Guide: [1]
- SPF Register Summary: [2]

Table 24-1423. SPF_INSTRW2

Address Offset	0x0000 0050		
Physical Address	0x4848 5C50 0x4848 5E50	**Instance**	SPF1 SPF2
Description	Instruction Word 2 Register		
Type	RW		

31:14	13:0
RESERVED	SPF_INSTR_W2

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	SPF_INSTR_W2	SPF Rule Engine Instruction Word [75:64] is read from or written to this field.	RW	0x0

Table 24-1424. Register Call Summary for Register SPF_INSTRW2

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 24-1425. SPF_INSTRW1

Address Offset	0x0000 0054		
Physical Address	0x4848 5C54 0x4848 5E54	**Instance**	SPF1 SPF2
Description	Instruction Word 1 Register		
Type	RW		

31:0
SPF_INSTR_W1

Bits	Field Name	Description	Type	Reset
31:0	SPF_INSTR_W1	SPF Rule Engine Instruction Word [63:32] is read from or written to this field.	RW	0x0

Table 24-1426. Register Call Summary for Register SPF_INSTRW1

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 24-1427. SPF_INSTRW0

Address Offset	0x0000 0058		
Physical Address	0x4848 5C58 0x4848 5E58	**Instance**	SPF1 SPF2
Description	Instruction Word 0 Register		
Type	RW		

31:0
SPF_INSTR_W0

Bits	Field Name	Description	Type	Reset
31:0	SPF_INSTR_W0	SPF Rule Engine Instruction Word [31:0] is read from or written to this field.	RW	0x0

Table 24-1428. Register Call Summary for Register SPF_INSTRW0

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 24-1429. SPF_INSTR_CTL

Address Offset	0x0000 005C		
Physical Address	0x4848 5C5C 0x4848 5E5C	**Instance**	SPF1 SPF2
Description	Instruction Control Register		
Type	RW		

31	30	29–6	5–0
SPF_INSTR_WEN	SPF_INSTR_REN	RESERVED	SPF_INSTR_PTR

Bits	Field Name	Description	Type	Reset
31	SPF_INSTR_WEN	SPF Write enable bit specifies whether a write operation is to be performed. To read or write instructions, spf processing must be stopped. When the rule engine is processing instructions, the instruction memory cannot be accessed. This bit is set to perform a write and the data in the SPF_INSTR_W2, SPF_INSTR_W1 and SPF_INSTR_W0 registers is written to the instruction RAM at address specified in the SPF_INSTR_PTR field. This bit is always read as zero.	W	0x0
30	SPF_INSTR_REN	SPF Read enable bit specifies whether a read operation is to be performed. This bit is set to perform a read and read data is available in the SPF_INSTR_W2, SPF_INSTR_W1 and SPF_INSTR_W0 registers once read operation has completed. This bit is always read as zero.	W	0x0
29:6	RESERVED		R	0x0
5:0	SPF_INSTR_PTR	The address in the instruction memory that is to be accessed.	RW	0x0

Table 24-1430. Register Call Summary for Register SPF_INSTR_CTL

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 24-1431. SPF_LOG_BEGIN

Address Offset	0x0000 0060		
Physical Address	0x4848 5C60 0x4848 5E60	**Instance**	SPF1 SPF2
Description	Log Begin Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPF_LOG_BEGIN

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_BEGIN	SPF starts to write log data to memory starting from address given in this field.	RW	0x0

Table 24-1432. Register Call Summary for Register SPF_LOG_BEGIN

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0] [1]
- Programming Guide: [2]
- SPF Register Summary: [3]
- SPF Register Description: [4] [5] [6]

Table 24-1433. SPF_LOG_END

Address Offset	0x0000 0064		
Physical Address	0x4848 5C64 0x4848 5E64	**Instance**	SPF1 SPF2
Description	Log End Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOG_END																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_END	This register along with SPF_LOG_BEGIN register defines the memory range for writing log data, the range(SPF_LOG_END SPF_LOG_BEGIN) should be multiple of 4 words(32 bits), as this is a look ahead register therefore the value progammed should be next word address. (i.e. last word address + 4).	RW	0x00001000

Table 24-1434. Register Call Summary for Register SPF_LOG_END

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0] [1] [2] [3]
- Programming Guide: [4]
- SPF Register Summary: [5]
- SPF Register Description: [6] [7] [8]

Table 24-1435. SPF_LOG_HWPTR

Address Offset	0x0000 0068		
Physical Address	0x4848 5C68 0x4848 5E68	**Instance**	SPF1 SPF2
Description	Log Hardware Pointer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOG_HWPTR																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_HWPTR	This register indicated the address of next location in memory that the SPF will log information to.	RW	0x0

Table 24-1436. Register Call Summary for Register SPF_LOG_HWPTR

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- SPF Register Summary: [1]
- SPF Register Description: [2]

Table 24-1437. SPF_LOG_SWPTR

Address Offset	0x0000 006C		
Physical Address	0x4848 5C6C 0x4848 5E6C	**Instance**	SPF1 SPF2
Description	Log Software Pointer Register		
Type	RW		

31–0
SPF_LOG_SWPTR

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_SWPTR	This register specifies the address where software shall do next read, software must inform SPF about memory roll over by writing SPF_LOG_END into this register.	RW	0x0

Table 24-1438. Register Call Summary for Register SPF_LOG_SWPTR

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0] [1] [2]
- SPF Register Summary: [3]
- SPF Register Description: [4]

Table 24-1439. SPF_LOG_MAP0

Address Offset	0x0000 0070		
Physical Address	0x4848 5C70 0x4848 5E70	**Instance**	SPF1 SPF2
Description	Filter Code Map Register 0		
Type	RW		

31–24	23–16	15–8	7–0
SPF_LOGMAP3	SPF_LOGMAP2	SPF_LOGMAP1	SPF_LOGMAP0

Bits	Field Name	Description	Type	Reset
31:24	SPF_LOGMAP3	Mapping of drop code 3 to log threshold 3	RW	0x0
23:16	SPF_LOGMAP2	Mapping of drop code 2 to log threshold 2	RW	0x0
15:8	SPF_LOGMAP1	Mapping of drop code 1 to log threshold 1	RW	0x0
7:0	SPF_LOGMAP0	Mapping of drop code 0 to log threshold 0	RW	0x0

Table 24-1440. Register Call Summary for Register SPF_LOG_MAP0

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 24-1441. SPF_LOG_MAP1

Address Offset	0x0000 0074		
Physical Address	0x4848 5C74 0x4848 5E74	**Instance**	SPF1 SPF2

Table 24-1441. SPF_LOG_MAP1 (continued)

Description	Filter Code Map Register 1
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOGMAP7								SPF_LOGMAP6								SPF_LOGMAP5								SPF_LOGMAP4							

Bits	Field Name	Description	Type	Reset
31:24	SPF_LOGMAP7	Mapping of drop code 7 to log threshold 7	RW	0x0
23:16	SPF_LOGMAP6	Mapping of drop code 6 to log threshold 6	RW	0x0
15:8	SPF_LOGMAP5	Mapping of drop code 5 to log threshold 5	RW	0x0
7:0	SPF_LOGMAP4	Mapping of drop code 4 to log threshold 4	RW	0x0

Table 24-1442. Register Call Summary for Register SPF_LOG_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 24-1443. SPF_LOG_THRESHk

Address Offset	0x0000 0078 + (k * 4)	**Index**	k = 0 to 8
Physical Address	0x4848 5C78 + (k * 4) 0x4848 5E78 + (k * 4)	**Instance**	SPF1 SPF2
Description	Log Threshold and Count Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_COUNT																SPF_THRESH															

Bits	Field Name	Description	Type	Reset
31:16	SPF_COUNT	Number of packets dropped for drop code k (8 is default)	R	0x0
15:0	SPF_THRESH	Number of packets to be dropped before logging starts	RW	0xA

Table 24-1444. Register Call Summary for Register SPF_LOG_THRESHk

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 24-1445. SPF_INTCNT

Address Offset	0x0000 009C		
Physical Address	0x4848 5C9C 0x4848 5E9C	**Instance**	SPF1 SPF2
Description	Interrupt Frequency Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											SPF_INTCNT				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	SPF_INTCNT	Number of time thresholds must be met before a drop interrupt is triggered.	RW	0x0

Table 24-1446. Register Call Summary for Register SPF_INTCNT

Gigabit Ethernet Switch (GMAC_SW)
- SPF Functional Description: [0] [1]
- Programming Guide: [2]
- SPF Register Summary: [3]
- SPF Register Description: [4]

Table 24-1447. SPF_INT_RAW

Address Offset	0x0000 00A0		
Physical Address	0x4848 5CA0 0x4848 5EA0	**Instance**	SPF1 SPF2
Description	Raw Interrupt Status register		
Type	RW		

31–1	0
RESERVED	SPF_INT_RAW

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_INT_RAW	Status of Raw interrupt signal	RW	0x0

Table 24-1448. Register Call Summary for Register SPF_INT_RAW

Gigabit Ethernet Switch (GMAC_SW)
- SPF Functional Description: [0]
- SPF Register Summary: [1]
- SPF Register Description: [2]

Table 24-1449. SPF_INT_MASKED

Address Offset	0x0000 00A4		
Physical Address	0x4848 5CA4 0x4848 5EA4	**Instance**	SPF1 SPF2
Description	Interrupt Status register		
Type	RW		

31–1	0
RESERVED	SPF_INT_MASKED

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_INT_MASKED	Status of interrupt signal with mask	RW	0x0

Table 24-1450. Register Call Summary for Register SPF_INT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- SPF Register Summary: [1]
- SPF Register Description: [2]

Table 24-1451. SPF_MASK_SET

Address Offset	0x0000 00A8		
Physical Address	0x4848 5CA8 0x4848 5EA8	**Instance**	SPF1 SPF2
Description	Interrupt Mask Set Register		
Type	RW		

31–1	0
RESERVED	SPF_MASKSET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_MASKSET	Write a 1 to this bit to enable the interrupt.	RW	0x0

Table 24-1452. Register Call Summary for Register SPF_MASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- Programming Guide: [1]
- SPF Register Summary: [2]

Table 24-1453. SPF_MASK_CLR

Address Offset	0x0000 00AC		
Physical Address	0x4848 5CAC 0x4848 5EAC	**Instance**	SPF1 SPF2
Description	Interrupt Mask Clear Register		
Type	RW		

31–1	0
RESERVED	SPF_MASKCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_MASKCLR	Write a 1 to this bit to disable the interrupt.	RW	0x0

Table 24-1454. Register Call Summary for Register SPF_MASK_CLR

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- SPF Register Summary: [1]

24.12 Media Local Bus (MLB)

24.12.1 MLB Overview

The Media Local Bus sub system (MLBSS) is based on a module designed by SMSC. This module provides a MediaLB/MediaLB+ controller and an interface to other MediaLB/MediaLB+ devices. The MediaLB/MediaLB+ interface allows also connection to a MOST (Media Oriented Systems Transport) network controller. MOST network is used to transport media and control data between various multimedia nodes in the car.

The MLBSS supports the following features:

- 3-pin MediaLB 3.3V LVCMOS I/Os compliant to MediaLB Physical Layer Specification v4.0
- 6-pin MediaLB+ low-voltage differential signaling (LVDS) I/Os (3 differential pairs) compliant to MediaLB Physical Layer Specification v4.0
- MediaLB core functionality compliant to MediaLB Physical and Link layer specification v4.0
- Supports 256/512/1024Fs in 3-pin mode, and 2048Fs in 6-pin mode
- Supports all types of transfer (synchronous stream data, asynchronous packet data, control message data, and isochronous data) over 64 logical channels
- Supports single 32-bit L4 slave interface for configuration
- Supports single 32-bit L3_MAIN master interface with burst capability for DMA transfers into system memory. The maximum burst size is 32 Bytes
- Has 16 KiB buffer for all types of transfers in the subsystem
- Support of pressure bits for controlling the MLBSS priority on the L3_MAIN interconnect

The MLBSS does not support:

- 400 MHz clock mode which enable half and full entitlement of MediaLB+ and MOST150
- 5-pin mode

Figure 24-206 shows the MLBSS overview.

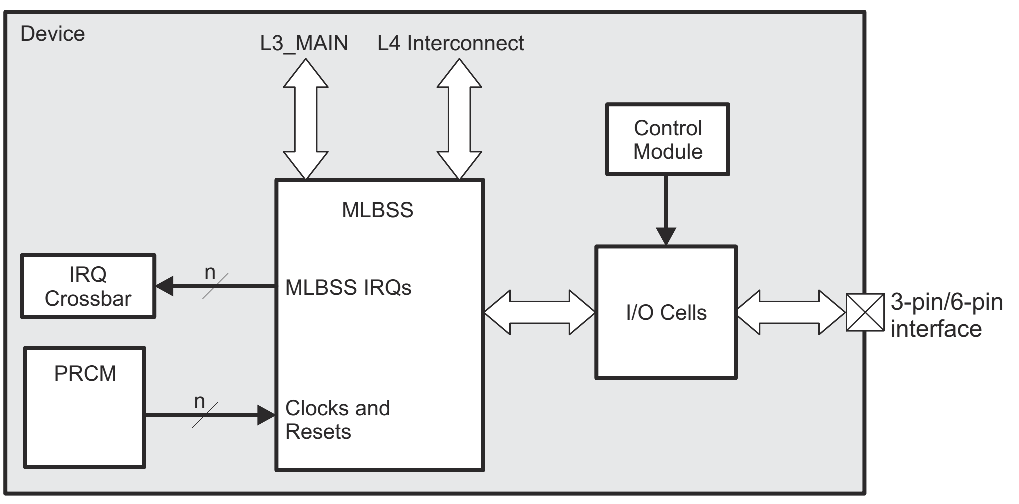


Figure 24-206. MLB Overview

24.12.2 MLB Environment

Figure 24-207 shows the MLBSS associated signals available on device pads.

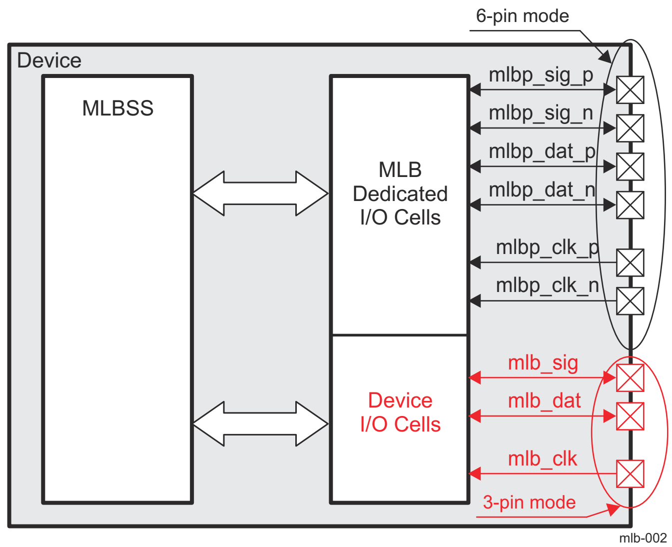


Figure 24-207. MLB Sub System Environment

Table 24-1455 describes the MLBSS I/O signals used in both 3-pin and 6-pin modes.

Table 24-1455. MLB Sub System I/O Description

Signal/Pad Name	I/O[(1)]	Description
		6-pin mode
mlbp_sig_p	IO	Differential pair signal line
mlbp_sig_n	IO	
mlbp_dat_p	IO	Differential pair data line
mlbp_dat_n	IO	
mlbp_clk_p	I	Differential pair clock line
mlbp_clk_n	I	
		3-pin mode
mlb_sig	IO	Single ended signal line
mlb_dat	IO	Single ended data line
mlb_clk	I	Single ended clock line

(1) I = Input; O = Output; I/O = Bidirectional

24.12.2.1 MLB IO Cell Controls

There are I/O cells specially intended for the MLB 6-pin interface. These are low-voltage differential signaling (LVDS) I/O cells with controls, which reside in several registers of the CTRL_MODULE_CORE. In addition, there is also a bandgap cell which provides a reference voltage to the LVDS IOs associated with the MLB interface. Without this reference voltage these differential receivers and transmitters cannot function properly. The LVDS IOs must be used only when 6-pin mode is selected. If this is done the six MLB signals are available on the following pads:

- mlbp_sig_p
- mlbp_sig_n
- mlbp_dat_p
- mlbp_dat_n
- mlbp_clk_p
- mlbp_clk_n

Figure 24-208 shows the LVDS IOs used when 6-pin MLB mode is selected, their controls from CTRL_MODULE_CORE, the MLB bandgap cell and details regarding the multiplexing scheme when 3-pin mode is used.

3-pin mode is used when the MLB_MLBC0[5] MLBPEN bit is set to 0x0. When this bit is set to 0x1 a 6-pin mode is used. In case of 3-pin mode an additional step must be performed to be the MLB signals (mlb_sig, mlb_dat and mlb_clk) available on the device pads. These signals are multiplexed along with other device signals on the following pads:

- mcasp5_aclkx
- mcasp5_axr[0]
- mcasp5_axr[1]

To provide the MLB 3-pin mode signals on these pads the MUXMODE bit field of the following registers should be set to 0x5:

- CTRL_CORE_PAD_MCASP5_ACLKX[3:0] MCASP5_ACLKX_MUXMODE
- CTRL_CORE_PAD_MCASP5_AXR0[3:0] MCASP5_AXR0_MUXMODE
- CTRL_CORE_PAD_MCASP5_AXR1[3:0] MCASP5_AXR1_MUXMODE

For more information about signal multiplexing see *Pad Configuration Registers* in *Control Module* and also table "Multiplexing Characteristics" of the device Data Manual.

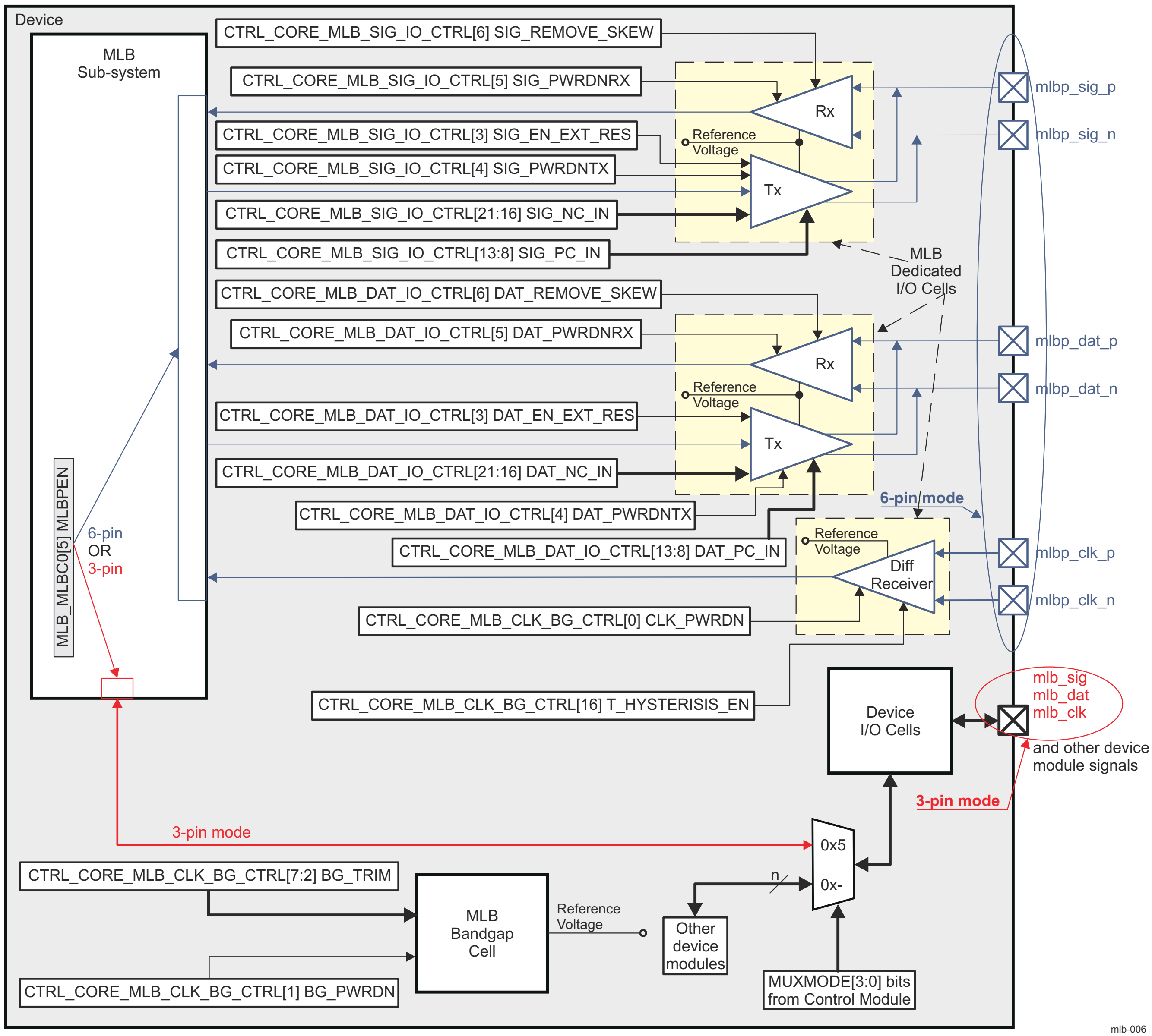


Figure 24-208. MLB I/O Cells And Their Controls

The MLB bandgap cell is intended to provide a stable reference voltage to the LVDS IOs across voltage and temperature variations. This cell is powered when the CTRL_CORE_MLB_CLK_BG_CTRL[1] BG_PWRDN bit is set to 0x0. When powerd the MLB bandgap cell provides its reference voltage to the LVDS IOs which is a prerequisite for their proper working. The CTRL_CORE_MLB_CLK_BG_CTRL[7:2] BG_TRIM bit field is used for trimming the reference voltage generated by the MLB bandgap cell. When reading the BG_TRIM bit field and if its value is 0x0 it is recommended to write 0x20 to this bit field. In all other cases (BG_TRIM != 0x0) software write to the BG_TRIM bit field is not recommended.

The MLB differential clock receiver is enabled when the CTRL_CORE_MLB_CLK_BG_CTRL[0] CLK_PWRDN bit is set to 0x0. The CTRL_CORE_MLB_CLK_BG_CTRL[16] T_HYSTERISIS_EN bit is used to enable/disable the hysteresis for the MLB differential clock receiver. A value of 0x1 enables the hysteresis and 0x0 disables it.

The power of the MLB differential receivers and transmitters can be switched ON and OFF using the PWRDNRX and PWRDNTX bits, respectively. A value of 0x0 powers up and a value of 0x1 powers down the receivers and transmitters. The bits are the following:

- CTRL_CORE_MLB_SIG_IO_CTRL[5] SIG_PWRDNRX for mlbp_sig_p/mlbp_sig_n receiver.
- CTRL_CORE_MLB_SIG_IO_CTRL[4] SIG_PWRDNTX for mlbp_sig_p/mlbp_sig_n transmitter.
- CTRL_CORE_MLB_DAT_IO_CTRL[5] DAT_PWRDNRX for mlbp_dat_p/mlbp_dat_n receiver.
- CTRL_CORE_MLB_DAT_IO_CTRL[4] DAT_PWRDNTX for mlbp_dat_p/mlbp_dat_n transmitter.

There are also CTRL_MODULE_CORE bits to remove the skew effect of the mlbp_sig_p/mlbp_sig_n and mlbp_dat_p/mlbp_dat_n signals caused by difference in the input impedance between "_p" and "_n" lines of the receivers. The bits for doing this are the following:

- CTRL_CORE_MLB_SIG_IO_CTRL[6] SIG_REMOVE_SKEW for mlbp_sig_p/mlbp_sig_n receiver.
- CTRL_CORE_MLB_DAT_IO_CTRL[6] DAT_REMOVE_SKEW for mlbp_dat_p/mlbp_dat_n receiver.

A value of 0x1 enables the skew compensation and 0x0 disables it.

The MLB differential transmitters have also controls for trimming their output impedance and for enabling/disabling their internal pull resistors. Setting to 0x1 the CTRL_CORE_MLB_SIG_IO_CTRL[3] SIG_EN_EXT_RES bit enables the internal pull resistors of the mlbp_sig_p/mlbp_sig_n transmitter. Setting to 0x1 the CTRL_CORE_MLB_DAT_IO_CTRL[3] DAT_EN_EXT_RES bit enables the internal pull resistors of the mlbp_dat_p/mlbp_dat_n transmitter. If the internal pull resistors are disabled external on-board pull resistors must be provided.

In addition, when 3-pin mode is used, there is no need to configure any of the previously described CTRL_MODULE_CORE registers associated with the MLB LVDS IOs. In this case a MUXMODE configuration is required.

24.12.3 MLB Integration

Figure 24-209 shows the integration of the MLB sub system in the device.

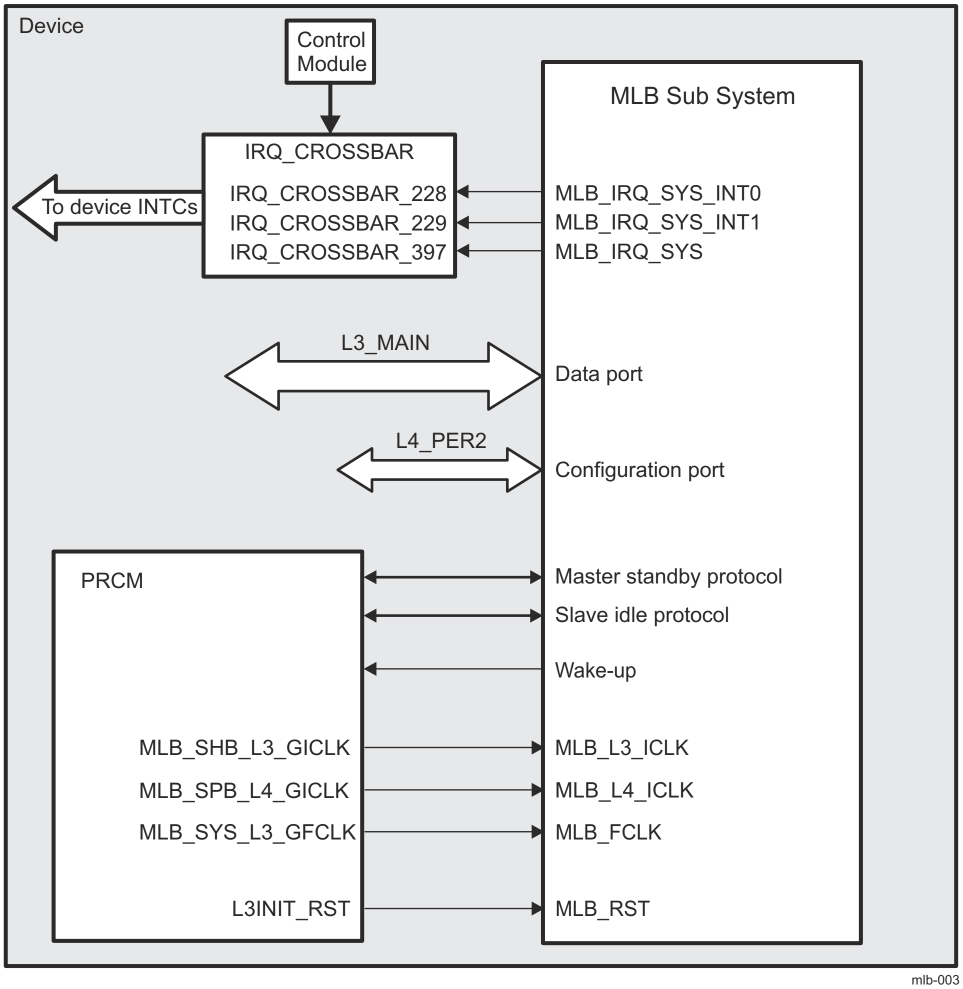


Figure 24-209. MLB Sub System Integration

Table 24-1456 through Table 24-1458 summarize the integration of the MLB sub system in the device.

Table 24-1456. MLB Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
MLB	PD_COREAON	Yes	L3_MAIN L4_PER2

Table 24-1457. MLB Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
MLB	MLB_L3_ICLK	MLB_SHB_L3_GICLK	PRCM	Interface clock for the data port of the MLB sub system
	MLB_L4_ICLK	MLB_SPB_L4_GICLK	PRCM	Interface clock for the configuration port of the MLB sub system
	MLB_FCLK	MLB_SYS_L3_GFCLK	PRCM	Functional clock for the MLB sub system
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**

Table 24-1457. MLB Clocks and Resets (continued)

MLB	MLB_RST	L3INIT_RST	PRCM	Asynchronous reset for the MLB sub system

Table 24-1458. MLB Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
MLB	MLB_IRQ_SYS_INT0	IRQ_CROSSBAR_228	-	MLBSS DMA interrupt request for logical channels 0 to 31 (1). This IRQ source signal is not mapped by default to any device INTC.
	MLB_IRQ_SYS_INT1	IRQ_CROSSBAR_229	-	MLBSS DMA interrupt request for logical channels 32 to 63(1). This IRQ source signal is not mapped by default to any device INTC.
	MLB_IRQ_SYS	IRQ_CROSSBAR_397	-	MLBSS interrupt for error notifications. This IRQ source signal is not mapped by default to any device INTC.

(1) If the MLB_DCTL[1] SMX bit is set to 0x1 the MLB_IRQ_SYS_INT0 line is used for all 64 logical channels.

Note

The "**Default Mapping**" column in Table 24-1458 *MLB Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.

For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

24.12.4 MLB Functional Description

24.12.4.1 Block Diagram

Figure 24-210 shows the MLBSS block diagram. It includes the following functional blocks:

- MediaLB core - Implements the physical and link layer requirements of either a 3-pin or the 6-pin interface. Serial-to-parallel/parallel-to-serial are also implemented along with MediaLB frame synchronization.
- Routing fabric block - Manages the flow of data between the MediaLB core and the DMA block, implementing bus arbiter and muxing logic to the channel table RAM and the data buffer RAM.
- Channel table RAM - Used for storing channel descriptors for managing accesses to dynamic buffers in the data buffer RAM.
- Data buffer RAM - Provides dynamic circular buffering between the transmit and receive devices.
- Memory interface - Implements a bridge between the configuration L4_PER2 slave interface and the channel table RAM or data buffer RAM interfaces.
- DMA - Implements a bus bridge between the DMA master and the routing fabric block.
- Registers - Used for configuration.
- Interrupt logic.

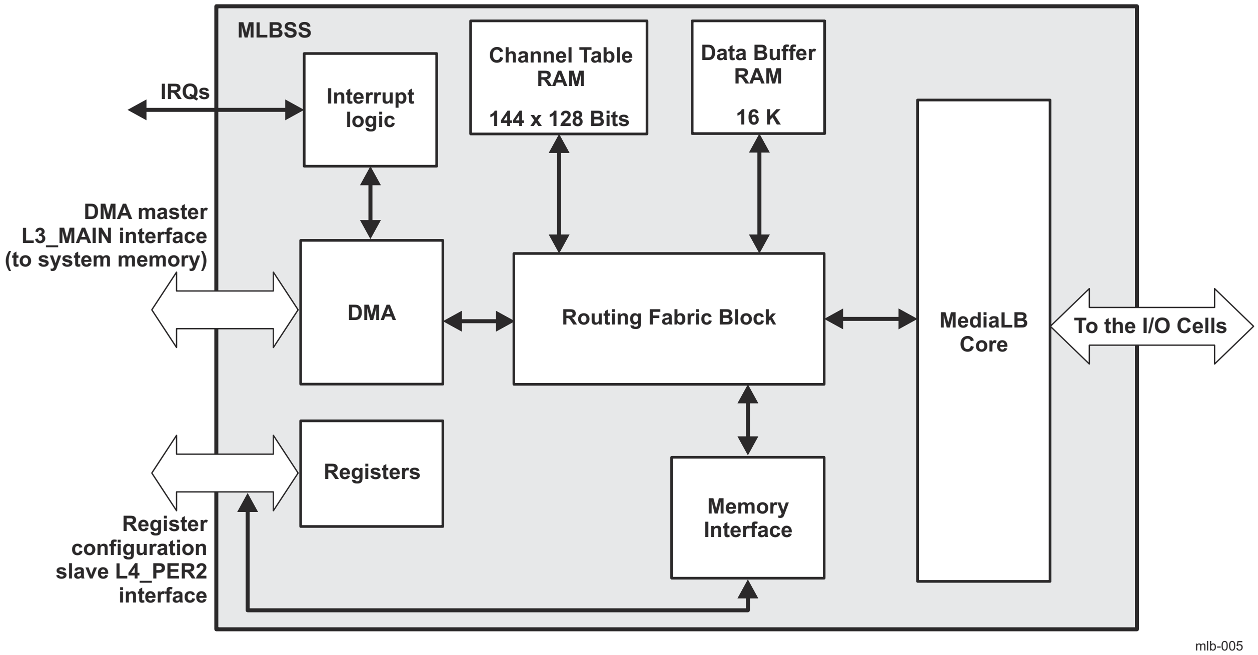


Figure 24-210. MLBSS Structural Overview

24.12.4.1.1 MediaLB Core Block

Although the MediaLB block supports both a MediaLB 3-pin and 6-pin interface, only one of these interfaces is active at any given time. The selection is done through the MLB_MLBC0[5] MLBPEN bit. Table 24-1455 describes the signals associated with 3-pin and 6-pin modes. Both MediaLB interfaces provide real-time access to all network data types including streaming, packet, control, and isochronous data.

The MediaLB 3-pin interface supports the MediaLB protocol for single-ended 3-pin mode with a maximum data rate of 1024Fs.

The MediaLB 6-pin interface supports the MediaLB protocol for high-speed differential 6-pin mode, with a maximum data rate of 2048Fs.

There is a set of physical channels implemented for exchanging data over the MediaLB 3-pin or 6-pin interface. These physical channels (called quadlets) are 4-byte wide and can also be grouped together to form logical channels. These logical channels are referenced using channel addresses, and define a uni-directional path between a specific MediaLB device transmitting data and another specific MediaLB device receiving data. In other words, the logical channel is a group of multiple quadlets, which selects an unique pair of MediaLB devices with a unique channel address. The logical channels, configured by the system software can be of any combination of channel types (synchronous, asynchronous, control, or isochronous) and direction (transmit/receive).

The MediaLB channel addresses are mapped to the logical channels as shown in Table 24-1459.

Table 24-1459. MediaLB Channel Address to Logical Channel Mapping

MediaLB Channel Address	Logical Channel Number
0x02	1
0x04	2
0x06	3
…	…
0x007C	62
0x007E	63

Table 24-1459. MediaLB Channel Address to Logical Channel Mapping (continued)

MediaLB Channel Address	Logical Channel Number
0x01FE	0 (1)

(1) The logical channel 0 is the system channel and is reserved.

Thus, 64 logical channels can be supported and one of them is the system channel which is reserved.

Note

The MediaLB channel address should always be an even number.

24.12.4.1.2 Routing Fabric Block

The routing fabric block manages the flow of data between the MediaLB core and the DMA block. Bus multiplexers and a bus arbiter are implemented in the routing fabric block for accessing the channel table RAM and data buffer RAM. Each DMA controller in the routing fabric uses channel descriptors to manage access to dynamic buffers in the data buffer RAM. These channel descriptors are stored in the channel table RAM.

24.12.4.1.3 Data Buffer RAM

The data buffer RAM is 8-bit × 16k entries deep and provides dynamic circular buffering between the transmit and receive devices. The size and location of each data buffer is defined by software in the channel descriptor table, which is located in the channel table RAM.

The DMA controllers in the routing fabric are responsible for ensuring that the circular buffers do not overflow or underflow. Each channel type (that is, synchronous, isochronous, asynchronous and control) has full and empty detection.

24.12.4.1.4 Channel Table RAM

The channel table RAM is 128-bit × 144-entry (max). It allows system software to dynamically configure channel routing and allocate data buffers in the data buffer RAM.

The channel table RAM is logically divided into three sub-tables:

- Channel descriptor table
- DMA descriptor table
- Channel allocation table

The details of the channel descriptor table and channel allocation table are described in Table 24-1460.

The details of the DMA descriptor table are described in Table 24-1470.

Table 24-1460. Channel Table RAM Address Mapping

Label	Address	Bits 127:96	Bits 95:64	Bits 63:32	Bits 31:0
Channel Descriptor Table (CDT):					
Channel descriptor table	0x00	CDT0[127:0], CL = 0			
	0x01	CDT1[127:0], CL = 1			
	0x02	CDT2[127:0], CL = 2			
	…	…			
	0x3D	CDT61[127:0], CL = 61			
	0x3E	CDT62[127:0], CL = 62			
	0x3F	CDT63[127:0], CL = 63			
DMA Descriptor Table (DDT):					

Table 24-1460. Channel Table RAM Address Mapping (continued)

Label	Address	Bits 127:96		Bits 95:64		Bits 63:32		Bits 31:0	
DMA descriptor table	0x40	DDT0[127:0], CAT64							
	0x41	DDT1[127:0], CAT65							
	0x42	DDT2[127:0], CAT66							
	…	…							
	0x7D	DDT61[127:0], CAT125							
	0x7E	DDT62[127:0], CAT126							
	0x7F	DDT63[127:0], CAT127							
Channel Allocation Table (CAT):									
Channel allocation table for MediaLB	0x80	CAT7	CAT6	CAT5	CAT4	CAT3	CAT2	CAT1	CAT0
	...	...	...	...	...	...	...	...	...
	0x87	CAT63	CAT62	CAT61	CAT60	CAT59	CAT58	CAT57	CAT56
Channel allocation table for DMA	0x88	CAT71	CAT70	CAT69	CAT68	CAT67	CAT66	CAT65	CAT64
	...	...	...	...	...	...	...	...	...
	0x8F	CAT127	CAT126	CAT125	CAT124	CAT123	CAT122	CAT121	CAT120

Note

A fixed relationship between DMA descriptor table entries and DMA channel allocation table entries exists. When using DMA channel 0 (CAT64) software should program DDT0. When using DMA channel 1 (CAT65) software should program DDT1, in case of CAT66 DDT2 should be programmed and so on. In case of CAT127 software should program DDT63.

The following base addresses are valid:

- Channel descriptor table has base address 0x00
- DMA descriptor table has base address 0x40
- Channel allocation table for MediaLB has base address 0x80
- Channel allocation table for DMA has base address 0x88

24.12.4.1.4.1 Channel Allocation Table

The channel allocation table comprises 16 entries of the channel table RAM. Each entry is 16-bit. The first entry starts at address 0x80 and the sixteenth entry at address 0x8F. Each 16-bit channel allocation table entry represents a logical connection to or from a transmit/receive device (that is, MediaLB or DMA channel). All entries are indexed according to a fixed physical address assigned to every Rx/Tx channel. This is shown in Table 24-1461. The value stored in a channel allocation table entry includes a 6-bit connection label (CL[5:0]), which provides a pointer to the channel descriptor table. To complete a logical channel and form a routing connection, system software must assign the same connection label to both the Rx and Tx channels.

Table 24-1461. Channel Allocation Table Entry Map

Peripheral	Number of Tx Channels Allowed	Number of Rx Channels Allowed	Channel Allocation Table Start Index	Channel Allocation Table End Index	Entries
MediaLB	0 to 64	64 - (Number of Tx channels)	0	63	64
DMA	0 to 64	64 - (Number of Tx channels)	64	127	64

The format of a full channel allocation table entry is shown in Table 24-1462. All reserved bits should be written to zero.

Table 24-1462. Format Of Channel Allocation Table Entry

Channel Type	15	14	13	12	11	10 8	7 6	5 0
Isochronous	Reserved	FCE	Reserved	RNW	CE	CT[2:0] = 3	Reserved	CL[5:0]
Asynchronous	Reserved		MT	RNW	CE	CT[2:0] = 2	Reserved	CL[5:0]
Control	Reserved		MT	RNW	CE	CT[2:0] = 1	Reserved	CL[5:0]
Synchronous	Reserved	MFE	MT	RNW	CE	CT[2:0] = 0	Reserved	CL[5:0]

The field descriptions of a channel allocation table entry are shown in Table 24-1463.

Table 24-1463. Field Descriptions Of Channel Allocation Table Entry

Field	Description
CL[5:0]	Connection Label (offset into channel descriptor table)
CT[2:0]	Channel type
	111 = Reserved
	110 = Reserved
	101 = Reserved
	100 = Reserved
	011 = Isochronous
	010 = Asynchronous
	001 = Control
	000 = Synchronous
CE	Channel enable
	0 = Disabled
	1= Enabled
RNW	Read, not Write [(1)]
	1 = Read
	0 = Write
MT	Mute Bit [(2)]
	1= Enabled
	0 = Disabled
FCE	Flow control enable [(3)]
	1 = Enabled
	0 = Disabled
MFE	Multi-frame per sub-buffer enable [(4)]
	1 = Enabled
	0 = Disabled
Reserved	Reserved. Software writes zeros to all reserved bits when the entry is initialized. The reserved bits are read-only after initialization.

(1) For a Tx channel (from the host to the MediaLB interface):

- DMA channel allocation table entry: RNW = 0 (write)
- MediaLB channel allocation table entry: RNW = 1 (read)

For a Rx channel (data from MediaLB to host):

- DMA channel allocation table entry: RNW = 1 (read)
- MediaLB channel allocation table entry: RNW = 0 (write)

(2) When set for synchronous channels, the MT bit forces Rx channels to write zeros into the channel data buffer, and Tx channels to output zeros on the physical interface. When set for asynchronous and control channels, the MT bit causes DMA to halt at a packet boundary. Not valid for isochronous channels.

(3) The FCE bit is used by MediaLB isochronous Rx channels only.

(4) The MFE bit is used by MediaLB synchronous channels only.

24.12.4.1.4.2 Channel Descriptor Table

The channel descriptor table comprises 64 channel table RAM entries. Each entry is 128-bit. The first entry starts at address 0x00 and the 64th entry at address 0x3F. Each 128-bit entry of the channel descriptor table is referenced by a connection label and contains information about data buffer in the data buffer RAM (that is, buffer size, or address pointers). The format of each channel descriptor table entry depends on the channel type (synchronous, isochronous, asynchronous, or control).

Software must write all reserved channel descriptor bits to ’0’ when the entry is initialized.

Synchronous Channel Descriptor Table

The MLBSS provides two modes of operation (standard and multi-frame per sub-buffer) to provide flexibility for implementing synchronous channels.

Channels set up for standard mode require less buffer space. For channels configured for standard mode, the host controller must transfer one full frame of streaming data in/out of the data buffer of each streaming channel.

Channels set up for multiple-frames per sub-buffer mode require more buffer space. For channels configured for multiple-frames per sub-buffer mode, the host controller must transfer N full frames of streaming data in/out of the data buffer of each streaming channel.

To set up a channel in multi-frame per sub-buffer mode:

- Program the MLB_MLBC0[17:15] FCNT bit field to select the number of frames per sub-buffer
- Program the channel allocation table to enable multi-frame sub-buffering (MFE = 1) for each particular channel
- Set the buffer depth (BD[11:0]) in the channel descriptor table
- Repeat for additional synchronous channels

The format of a synchronous channel descriptor table entry is shown in Table 24-1464. All reserved bits should be written to zero.

Table 24-1464. Format Of Synchronous Channel Descriptor Table Entry

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	WSBC[1:0]		Reserved													
16	RSBC[1:0]		Reserved													
32	Reserved															
48	Reserved															
64	WSTS[3:0]				WPTR[11:0]											
80	RSTS[3:0]				RPTR[11:0]											
96	Reserved				BD[11:0]											
112	Reserved		BA[13:0]													

The field descriptions of a synchronous channel descriptor table entry are shown in Table 24-1465.

Table 24-1465. Field Descriptions Of Synchronous Channel Descriptor Table Entry

Field	Description	Details	Accessibility [(1)]
BA	Buffer base address	BA can start at any byte in the 16k data buffer RAM	r, w
BD	Buffer depth	1. BD = size of buffer in bytes - 1 2. Buffer end address = BA + BD 3. BD = 4 × m × bpf - 1, where: m = frames per sub-buffer (for MFE = 0, m = 1) bpf = bytes per frame	r,w
RPTR	Read pointer	1. Software initializes to zero, hardware updates 2. Counts the read address offset within a buffer 3. DMA read address = BA + RPTR	r,w,u

Table 24-1465. Field Descriptions Of Synchronous Channel Descriptor Table Entry (continued)

Field	Description	Details	Accessibility [1]
WPTR	Write pointer	1. Software initializes to zero, hardware updates 2. Counts the write address offset within a buffer 3. DMA write address = BA + WPTR	r,w,u
RSBC	Read sub-buffer counter	1. Software initializes to zero, hardware updates 2. Counts the read sub-buffer offset 3. DMA uses for pointer management	r,w,u
WSBC	Write sub-buffer counter	1. Software initializes to zero, hardware updates 2. Counts the write sub-buffer offset 3. DMA uses for pointer management	r,w,u
RSTS	Read status	1. Software initializes to zero, hardware updates 2. RSTS states: (Only for DMA associated with MLB) xxx0 = normal operation (no mute) xxx1 = normal operation (mute) xx0x = idle	r,w,u
WSTS	Write status	1. Software initializes to zero, hardware updates 2. WSTS states: (Only for DMA associated with MLB) xxx0 = normal operation (no mute) xxx1 = normal operation (mute) xx0x = idle 1xxx = command protocol error	r,w,u
Reserved	Reserved	Software writes a zero to all Reserved bits when the entry is initialized. The Reserved bits are Read-only after initialization.	r,w,u

(1)
- r - software readable
- w - software writable
- u - updated periodically by hardware

Isochronous Channel Descriptor Table

The format of an isochronous channel descriptor table entry is shown in Table 24-1466. All reserved bits should be written to zero.

Table 24-1466. Format Of Isochronous Channel Descriptor Table Entry

<table>
<tr><th>Bit Offset</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="16">Reserved</td></tr>
<tr><td>16</td><td colspan="16">Reserved</td></tr>
<tr><td>32</td><td colspan="7">Reserved</td><td colspan="9">BS[8:0]</td></tr>
<tr><td>48</td><td colspan="16">Reserved</td></tr>
<tr><td>64</td><td colspan="3">WSTS[2:0]</td><td colspan="13">WPTR[12:0]</td></tr>
<tr><td>80</td><td colspan="3">RSTS[2:0]</td><td colspan="13">RPTR[12:0]</td></tr>
<tr><td>96</td><td colspan="3">Reserved</td><td colspan="13">BD[12:0]</td></tr>
<tr><td>112</td><td>BF</td><td>Reserved</td><td colspan="14">BA[13:0]</td></tr>
</table>

The field descriptions of an isochronous channel descriptor table entry are shown in Table 24-1467.

Table 24-1467. Field Descriptions Of Isochronous Channel Descriptor Table Entry

Field	Description	Details	Accessibility [1]
BA	Buffer Base Address	BA can start at any byte in the 16k data buffer RAM	r, w

Table 24-1467. Field Descriptions Of Isochronous Channel Descriptor Table Entry (continued)

Field	Description	Details	Accessibility [(1)]
BD	Buffer Depth	1. BD = size of buffer in bytes - 1 2. Buffer end address = BA + BD 3. Isochronous buffers must be large enough to hold at least 3 blocks (packets) of data 4. Buffer depth must be a integer multiple of blocks	r,w
BF	Buffer Full	1. Software initializes to zero, hardware updates 2. DMA write hardware sets BF when the buffer is full 3. DMA read hardware clears BF when the buffer is empty 4. BF is valid only when the buffer is full or empty, otherwise ignore	r,w,u
BS	Block Size	1. BS defines when to begin the DMA to the data buffer 2. BS = buffer block size in bytes - 1 3. For Rx channels, the DMA writes start when the number of empty bytes (SPACE) in the data buffer >= the block size 4. For Tx channels, the DMA reads start when the number of valid bytes (VALID) in the data buffer >= the block size	r,w,u
RPTR	Read Pointer	1. Software initializes to zero, hardware updates 2. Counts the read address offset within a buffer 3. DMA read address = BA + RPTR	r,w,u
WPTR	Write Pointer	1. Software initializes to zero, hardware updates 2. Counts the write address offset within a buffer 3. DMA write address = BA + WPTR	r,w,u
RSTS	Read status	1. Software initializes to zero, hardware updates 2. RSTS Status states (Only for DMA associated with MLB) xx1 = active xx0 = idle	r,w,u
WSTS	Write status	1. Software initializes to zero, hardware updates 2. WSTS states: (Only for DMA associated with MLB) xx1 = active xx= idle x1x = command protocol error 1xx = buffer overflow (FCE = 0 only)	r,w,u
Reserved	Reserved	Software writes a zero to all reserved bits when the entry is initialized. The reserved bits are read-only after initialization.	r,w,u

(1)
- r - software readable
- w - software writable
- u - updated periodically by hardware

Asynchronous and Control Channel Description Table

The format of an asynchronous and control channel descriptor table entry is shown in Table 24-1468. All reserved bits should be written to zero.

Table 24-1468. Format Of Asynchronous and Control Channel Descriptor Table Entry

<table>
<tr><th>Bit Offset</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="5">WPC[4:0]</td><td colspan="11">Reserved</td></tr>
<tr><td>16</td><td colspan="5">RPC[4:0]</td><td colspan="11">Reserved</td></tr>
<tr><td>32</td><td>Reserved</td><td colspan="3">WPC[7:5]</td><td colspan="12">Reserved</td></tr>
<tr><td>48</td><td>Reserved</td><td colspan="3">RPC[7:5]</td><td colspan="12">Reserved</td></tr>
<tr><td>64</td><td colspan="4">WSTS[3:0]</td><td colspan="12">WPTR[11:0]</td></tr>
<tr><td>80</td><td colspan="4">RSTS[3:0]</td><td colspan="12">RPTR[11:0]</td></tr>
<tr><td>96</td><td>RSTS[4]</td><td>WSTS[4]</td><td colspan="2">Reserved</td><td colspan="12">BD[11:0]</td></tr>
<tr><td>112</td><td colspan="2">Reserved</td><td colspan="14">BA[13:0]</td></tr>
</table>

The field descriptions of an asynchronous and control channel descriptor table entry are shown in Table 24-1469.

Table 24-1469. Field Descriptions Of Asynchronous And Control Channel Descriptor Table Entry

Field	Description	Details	Accessibility [2]
BA	Buffer Base Address	BA can start at any byte in the 16k data buffer RAM	r, w
BD	Buffer Depth	1. BD = size of buffer in bytes - 1 2. Buffer end address = BA + BD 3. BD >= max packet length - 1	r,w
RPC	Read Packet Count	1. Software initializes to zero, hardware updates 2. Used in conjunction with WPC, RPTR and WPTR to determine if the buffer is empty or full	r,w,u
WPC	Write Packet Count	1. Software initializes to zero, hardware updates 2. Used in conjunction with RPC, RPTR and WPTR to determine if the buffer is empty or full	r,w,u
RPTR	Read Pointer	1. Software initializes to zero, hardware updates 2. Counts the read address offset within a buffer 3. DMA read address = BA + RPTR	r,w,u
WPTR	Write Pointer	1. Software initializes to zero, hardware updates 2. Counts the write address offset within a buffer 3. DMA write address = BA + WPTR	r,w,u
RSTS	Read status	1. Software initializes to zero, hardware updates 2. Status states:[1] x0x00 = idle xx1xx = ReceiverProtocolError response received from Rx device 1xxxx = ReceiverBreak response received from Rx device	r,w,u
WSTS	Write status	1. Software initializes to zero, hardware updates 2. Status states:[1] x0x00 = idle xx1xx = command protocol error detected 1xxxx = AsyncBreak/ControlBreak command received from Tx device	r,w,u

Table 24-1469. Field Descriptions Of Asynchronous And Control Channel Descriptor Table Entry (continued)

Field	Description	Details	Accessibility (2)
Reserved	Reserved	Software writes a zero to all reserved bits when the entry is initialized. The Reserved bits are read-only after initialization.	r,w,u

(1) Only valid for DMA pointers associated with the MediaLB core. Not valid for DMA block related pointers.
(2)
- r - software readable
- w - software writable
- u - updated periodically by hardware

24.12.4.1.5 DMA Block

The DMA block manages data exchange between the local channel data buffers which reside in the MLBSS and a memory buffer residing in the system memory. To support system memory buffering, a ping-pong memory structure is implemented on a per channel basis using 128-bit descriptors for DMA descriptor table entries. 64 DMA descriptor table entries are directly mapped to the 64 DMA physical channels.

Each logical channel is assigned a separate 128-bit descriptor, defining data buffers in the system memory which are used by the DMA interface for this logical channel. The descriptors are stored at fixed addresses in the channel table RAM.

The description of the DMA descriptor table fields is shown in Table 24-1470.

Table 24-1470. Field Descriptions Of DMA Descriptor Table

Field	Numbers of bits	Description	Accessibilty (1)
CE	1	Channel enable: 0 = Disabled 1 = Enabled	r,w,u
LE	1	Endianess Enable: 0 = Big endian 1 = Little Endian	r, w
PG	1	Page pointer. Software initializes to zero, hardware writes thereafter. 0 = Ping buffer 1 = Pong buffer	r,w,u
RDY1	1	Buffer ready bit for ping buffer page: 0 = Not ready 1 = Ready	r, w
RDY2	1	Buffer ready bit for pong buffer page: 0 = Not ready 1 = Ready	r, w
DNE1	1	Buffer done bit for ping buffer page: 0 = Not done 1 = Done	r,u,c0
DNE2	1	Buffer done bit for pong buffer page: 0 = Not done 1 = Done	r,u,c0
ERR1	1	DMA error response detected for ping buffer page: 0 = No error 1 = Error	r,u,c0

Table 24-1470. Field Descriptions Of DMA Descriptor Table (continued)

Field	Numbers of bits	Description	Accessibilty [1]
ERR2	1	DMA error response detected for pong buffer page: 0 = No error 1 = Error	r,u,c0
PS1	1	Packet start bit for ping buffer page: 0 = No packet start 1 = Packet start Reserved for synchronous and isochronous channels.	r,w,u (both Tx and Rx)
PS2	1	Packet start bit for pong buffer page: 0 = No packet start 1 = Packet start Reserved for synchronous and isochronous channels.	r,w,u (both Tx and Rx)
MEP1	1	Most ethernet packet (MEP) indicator for ping buffer page: 0 = No MEP 1 = MEP MEP1 only valid for the first page of a segmented buffer. Reserved for control, synchronous and isochronous channels	r,w,u (both Tx and Rx)
MEP2	1	Most ethernet packet (MEP) indicator for pong buffer page: 0 = No MEP 1 = MEP MEP2 only valid for the first page of a segmented buffer. Reserved for control, synchronous and isochronous channels	r,w,u (both Tx and Rx)
BD1[2]	11 to 13	Buffer depth for ping buffer page: 11 or 12-bits for asynchronous and control channels. 13-bits for synchronous and isochronous channels.	r,w
BD2[2]	11 to 13	Buffer depth for pong buffer page: 11 or 12-bits for asynchronous and control channels. 13-bits for synchronous and isochronous channels.	r,w
BA1	32	Buffer base address for ping buffer page	r,w
BA2	32	Buffer base address for pong buffer page	r,w
Reserved	varies	Software writes a zero to all Reserved bits when the entry is initialized. The Reserved bits are Read-only after initialization.	r,w, u

(1)
- r - software readable
- w - software writable
- u - updated periodically by hardware
- c0 - software writes '0' to clear

(2) The buffer depth (BD1 and BD2) for synchronous channels must consider if multi-frame per sub-buffer mode (the MFE bit in the Channel Allocation Table) is enabled.

Data exchange across the DMA interface can be configured as little endian (LE = 0x1) or big endian (LE = 0x0). Figure 24-211 provides an overview of the endian options, chosen by the LE bit.

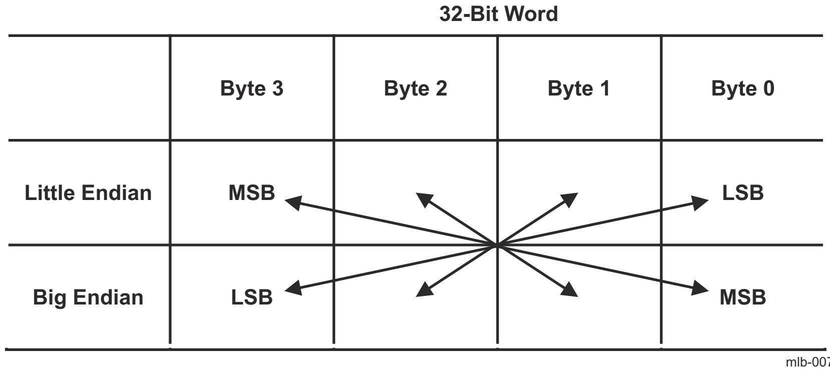


Figure 24-211. DMA Descriptor Table Endian Options

Figure 24-212 shows an example of ping-pong system memory structure. This system memory structure is similar for all channel types and shows the relationship between the BA1, BA2, BD1, BD2 and PG descriptor fields.

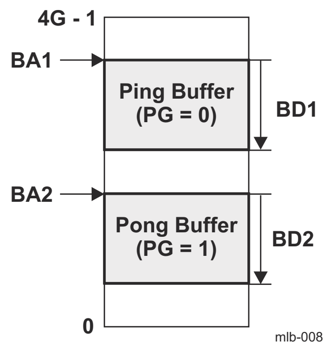


Figure 24-212. Ping-Pong System Memory Structure

Each DMA descriptor table entry (also referred to as a channel descriptor) holds a 32-bit BAn (n = 1 and 2) field which defines the start of each ping or pong buffer within system memory. The BDi (i = 1 and 2) field is used to indicate the size for the respective ping or pong page. The maximum size is 2k-entries for asynchronous and control channels, and 8k-entries for isochronous and synchronous channels.

24.12.4.1.5.1 Synchronous Channel Descriptor

The synchronous buffering scheme allows each ping or pong buffer to contain a single frame or a multiple number of frames. For this reason, the synchronous buffer depth (BDi) must be defined in terms of an integer number (n), frames per sub-buffer (m) and bytes per frame (bpf) of data (that is, BDi = n × m × bpf - 1).

Table 24-1471 shows the format for a synchronous DMA descriptor table entry. For more details about the field descriptions see Table 24-1470. Each synchronous channel buffer can be up to 8k-bytes deep.

Note

In synchronous transmit mode software should take care that the pong DDT is configured before ping DMA completion and ping DDT is configured before pong DMA completion. For more information about ping and pong DMA configuration see Table 24-1483 and Table 24-1484, respectively.

Table 24-1471. Format Of Synchronous DMA Descriptor Table Entry

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	BD1[12:0]												

Table 24-1471. Format Of Synchronous DMA Descriptor Table Entry (continued)

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
48	RDY2	DNE2	ERR2	BD2[12:0]												
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															

24.12.4.1.5.2 Isochronous Channel Descriptors

The isochronous buffering scheme allows each ping or pong buffer to contain a single block or a multiple number of blocks. For this reason, the isochronous buffer depth (BDi) must be defined in terms of an integer number (n) and block size (BS) (that is, BDi = n × (BS + 1) - 1).

Table 24-1472 shows the format for an isochronous DMA descriptor table entry. For more details about the field descriptions see Table 24-1470. Each isochronous channel buffer can be up to 8k-bytes deep.

Table 24-1472. Format Of Isochronous DMA Descriptor Table Entry

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	BD1[12:0]												
48	RDY2	DNE2	ERR2	BD2[12:0]												
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															

24.12.4.1.5.3 Asynchronous and Control Channel Descriptors

Every asynchronous and control designates the first two bytes of each packet as the packet length (PML). Each packet must be no more than 2048 bytes (PML ≤ 2048).

Software must set the buffer ready bit (RDYn) for each buffer as it programs the DMA. As hardware processes each buffer, it sets the done bit (DNEn) and generates an interrupt. When hardware finishes the buffer processing, it can begin processing another buffer if RDYn is set. The application is responsible for setting up and configuring the channel buffer descriptor prior to every DMA access on the channel.

Two-packet modes are supported by hardware for programming the DMA. These are single-packet mode and multiple-packet mode.

24.12.4.1.5.3.1 Single-Packet Mode

The single-packet mode asynchronous and control buffering scheme supports a maximum of one packet per buffer (that is, ping or pong). Both non-segmented and segmented data packets are allowed while using single-packet mode. Non-segmented packets are exchanged when only one buffer (that is ping or pong) is needed for packet transfer. Segmented packets are exchanged when a single packet is too long for one buffer and the packet must span multiple buffers. Figure 24-213 shows the memory space usage for both non-segmented and segmented asynchronous or control packets along with the packet start bit (PSn).

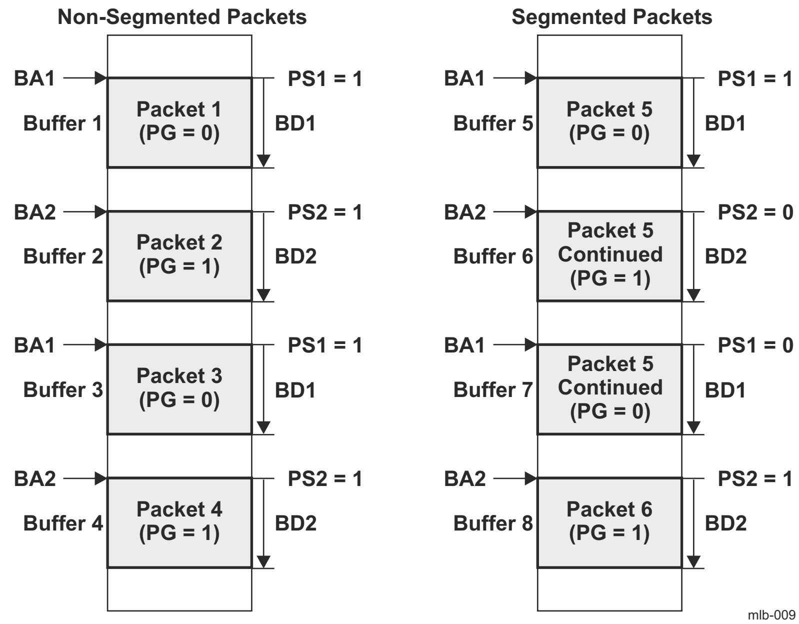


Figure 24-213. Single-Packet Mode Memory Space

While using single-packet mode, buffer done (DNEn) is set in hardware when a packet is done or the buffer is full.

Table 24-1473 shows the format for single-packet mode asynchronous and control DMA descriptor table entries. For more details about the field descriptions see Table 24-1470.

Table 24-1473. Format Of Single-Packet Asynchronous and Control DMA Descriptor Table Entry

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	PS1	MEP1	BD1[10:0]										
48	RDY2	DNE2	ERR2	PS2	MEP2	BD2[10:0]										
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															

24.12.4.1.5.3.2 Multiple-Packet Mode

The multiple-packet mode asynchronous and control buffering scheme supports more than one packet per system memory buffer, as shown in Figure 24-214. The multiple-packet mode reduces the interrupt rate for packet channels at the cost of increasing buffering and latency.

For Tx packet channels in multiple-packet mode, software sets the packet start bit (PSn) for every buffer. Setting PSn informs hardware that the first two bytes of the buffer contains the port message length (PML) of the first packet. After the first packet, hardware keeps track of where packets start and end within the current buffer. Software should not write to PSn while the buffer is active (RDYn = 1 and DNEn = 0). For Tx packet channels, the buffer is done (DNEn = 1) when the last byte of the last packet in the buffer is read from system memory. Software should set the buffer depth to contain the exact number of complete packets for that buffer. Segmented buffers are not supported for Tx packet channels in multiple-packet mode.

For Rx packet channels in multiple-packet mode, PSn has no meaning and should be ignored. Software is responsible for keeping track of where each packet starts and ends within the multiple-packet buffer via the packet PML. The buffer done bit (DNEn) is set in hardware for Rx channels when a buffer is full (see Buffer 1 in Figure 24-214) or if a packet ends exactly 1-byte before the end of the buffer (see Buffer 2 in Figure 24-214). Multiple-packet mode also supports segmented Rx packets spanning two or more buffers (see Buffers 3 – 6 in Figure 24-214).

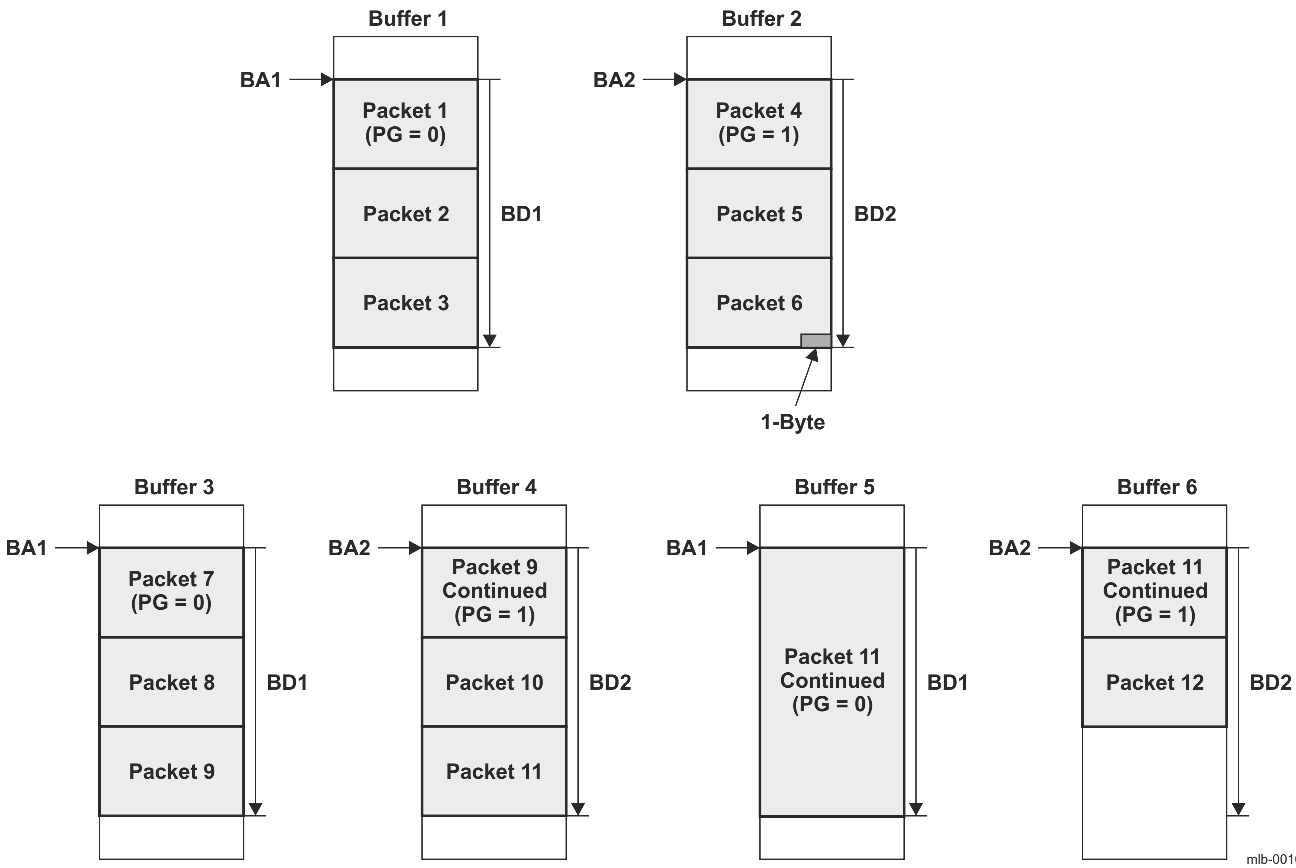


Figure 24-214. Multi-Packet Mode System Memory

Table 24-1474. Format Of Multiple-Packet Asynchronous And Control DMA Descriptor Table Entry

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	PS1 (1)	BD1[11:0]											
48	RDY2	DNE2	ERR2	PS2 (1)	BD2[11:0]											
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															

(1) PS1 and PS2 have no meaning and should be ignored in Rx mode.

24.12.4.2 Software and Data Flow for MLBSS

Figure 24-215 shows an overview of the MLBSS software and data flow.

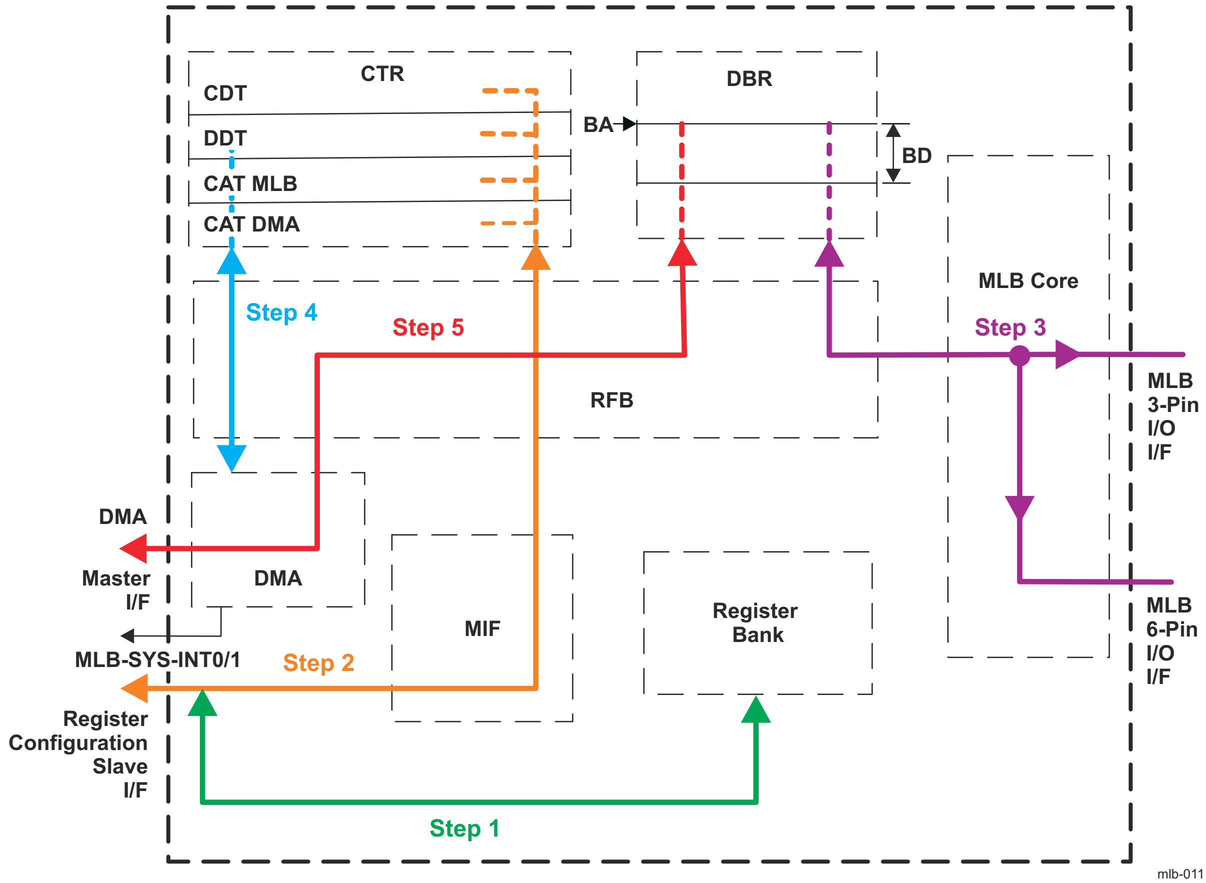


Figure 24-215. MLBSS Software and Data Flow Overview

The MLBSS data flow is defined in the following way:

1. Hardware is configured (see Table 24-1477)
2. Channel table RAM is cleared and configured (see Table 24-1478)
3. Data transaction begins through MediaLB 3-pin or 6-pin interface after MLBSS is synchronized.
4. DMA reads from channel table RAM.
5. Data transaction is done through the DMA master interface.

Steps 1 and 2 are done by software. For more information see Section 24.12.5.1.2.1, *Channel Initialization*

Steps 3-5 are handled by MLBSS internal hardware.

The top-level software tasks that the application must perform can be placed in two categories:

- Channel initialization
- Channel servicing

For more information about channel initialization, see Section 24.12.5.1.2.1, *Channel Initialization*.

For more information about channel servicing, see Section 24.12.5.2.1, *Channel Servicing*.

24.12.4.2.1 Data Flow For Receive Channels

The data flow for receive channels is as follows:

1. After MLBSS is configured (see Section 24.12.5.1.2.1) and MediaLB has synchronized to the incoming MediaLB framesync (MLB_MLBC0[7] MLBLK = 0x1), serial data is received either through the mlbp_dat_p and mlbp_dat_n pads or through the mlb_dat pad, depending on the pin mode (6-pin or 3-pin) selected.
2. The MLBSS logic writes incoming data to the appropriate buffer in the data buffer RAM. The data buffer RAM address to which data is written is the BA field of the channel descriptor table entry. The channel

descriptor table entry is specified by the CL field of the MediaLB channel allocation table entry for the channel. The MediaLB channel allocation table entry equals (1/2)*ChannelAddr.

3. The DMA access for a particular DMA channel (DMA channel allocation table entry) is started when there is sufficient data for the DMA channel in the data buffer RAM to write into system memory and the system memory buffer (ping or pong depending on the PG bit in DMA descriptor table entry) is marked as ready. Note that presence of a frame for synchronous/isochronous traffic (multiple frames in multi frame mode) or a packet for control/asynchronous traffic (multiple packets in multi packet mode) is deemed sufficient data to mark a DMA channel as ready.
4. The DMA hardware reads from the data buffer RAM memory address specified by the BA field of the channel descriptor table entry and writes to the system memory address specified in the DMA descriptor table. The CL field in the DMA channel allocation table entry is used to determine the appropriate channel descriptor table entry. Note that CL field of DMA channel allocation table and MLB channel allocation table entries must be the same, thus providing the common data buffer RAM address for MLB core to write into DMA and the DMA to read from the MLB core. Further, each DMA channel allocation table entry corresponds to a DMA descriptor table entry as per the following mapping:
   - Address of DMA descriptor table entry = 0x40 + Index of DMA channel allocation table entry.

24.12.4.2.2 Data Flow for Transmit Channels

The data flow for transmit channels is as follows:

1. The DMA access for a particular DMA channel (DMA channel allocation table entry) is started when data buffer RAM has sufficient buffer space and the system memory buffer (ping or pong depending on the PG bit in DMA descriptor table entry) is marked as ready for a particular DMA channel allocation table entry. Each DMA channel allocation table entry corresponds to a DMA descriptor table entry as per the following mapping:
   - Address of DMA descriptor table entry = 0x40 + Index of DMA channel allocation table entry.
2. The DMA reads from the system memory address specified in the DMA descriptor table and writes to the data buffer RAM memory address specified by the BA field of the channel descriptor table entry. The CL field in the DMA channel allocation table entry is used to determine the appropriate channel descriptor table entry. Note that CL field of DMA channel allocation table and MLB channel allocation table entries must be the same, thus providing the common data buffer RAM address for DMA to write into MLB core and the MLB core to read from the DMA.
3. Once the MLBSS is synchronized to the incoming MediaLB framesync (MLB_MLBC0[7] MLBLK = 0x1) and a MediaLB channel address is received, the MediaLB core refers to the specific MediaLB channel allocation table entry and reads data from the data buffer RAM buffer. The data buffer RAM address from which data is read is the BA field of the channel descriptor table entry. The channel descriptor table entry is specified by the CL field of the MLB channel allocation table entry for the channel. The MediaLB channel allocation table entry equals (1/2)*ChannelAddr.
4. This data is serialized and transmitted through the MediaLB 3-pin or 6-pin interface.

24.12.4.3 MLB Priority On The L3_MAIN Interconnect

There is a register used for controlling the priority of the MLB sub-system on the L3_MAIN. This is done through the CTRL_CORE_IP_PRESSURE register which reside in the CTRL_MODULE_CORE. The MLB has a default priority (pressure setting) on the L3_MAIN. To override the default setting the CTRL_CORE_IP_PRESSURE[2] MLB_L3_PRESSURE_ENABLE bit must be set to 0x1. When this is done the values configured through the CTRL_CORE_IP_PRESSURE[1:0] MLB_L3_PRESSURE bit field apply. Setting this bit field to 0x3 means that the MLB traffic has highest priority over the other initiator traffics. A value of 0x0 is for lowest priority.

24.12.5 MLB Programming Guide

This section describes the programming sequences for the configuration and use of the MLBSS.

24.12.5.1 Global Initialization

24.12.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module is used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MLB sub-system. For more information, see Section 24.12.3, *MLB Integration*, and Section 24.12.2, *MLB Environment*.Table 24-1475 shows the global initialization of MLBSS surrounding modules.

Table 24-1475. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The interface and functional clocks of the MLBSS must be enabled. See *Power, Reset, and Clock Management*.
Control module	The pad muxing must be configured in the control module. The MLBSS I/Os must be configured also in the control module. See *Pad Configuration Registers* in *Control Module* . See also table "Multiplexing characteristics" in the device Data Manual.
IRQ_CROSSBAR	Mapping of the MLBSS IRQs to a desired line of certain of the device interrupt controllers must be configured. See *IRQ_CROSSBAR Module Functional Description* in *Control Module* .
Interrupt Controllers	An INTC configuration must be done to enable the interrupts from the MLBSS. See *Interrupt Controllers*.

24.12.5.1.2 MLBSS Global Initialization

24.12.5.1.2.1 Channel Initialization

A channel initialization must be performed in order to ensure proper operation. Table 24-1476 shows the sequence.

Table 24-1476. Channel Initialization Steps

Step	Crossreference
Configuring the hardware	Table 24-1477
Programming the routing fabric block	Table 24-1478
Programming the DMA block	Table 24-1482
Synchronizing and unmuting the synchronous channel	Table 24-1485

Configuring The Hardware

To configure the MediaLB interface, the steps shown in Table 24-1477 should be followed.

Table 24-1477. Configuring The Hardware

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Clear channel table RAM, to ensure that no stray interrupts are generated	-	0x0
Clear the interrupt status bits for logical channels 0 to 63	MLB_DCSR0 and MLB_DCSR1	0xFFFF FFFF
Mask all DMA channels (from 0 to 63)	MLB_DCMR0 and MLB_DCMR1	0x0000 0000
Select 3-pin or 6-pin MediaLB operation	MLB_MLBC0[5] MLBPEN	0x0 for 3-pin 0x1 for 6-pin
Select MediaLB clock speed	MLB_MLBC0[4:2] MLBCLK	0x-
Enable the MediaLB	MLB_MLBC0[0] MLBEN	0x1
Activate all channels	MLB_DICER0 and MLB_DICER1	0xFFFF FFFF
Enable the DMA	MLB_DIENR[15] EN	0x1
Set the DMA channel mask bits according to all active DMA channels	MLB_DCMR0 and MLB_DCMR1	0x-
Select DMA Mode 1 to be used	MLB_DCTL[2] DMA_MODE	0x1
Select software to clear the interrupts	MLB_DCTL[0] SCE	0x1

Table 24-1477. Configuring The Hardware (continued)

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Enable MLB interrupts, if required	MLB_MIEN	0x-

Programming The Routing Fabric Block

The channel allocation table and channel descriptor table reside in the channel table RAM and are programmed indirectly through the memory interface block. The steps to be followed are shown in Table 24-1478.

Table 24-1478. Programming The Routing Fabric Block

Step	Register/ Bit Field/ Programming Model/ Comments	Value
1. Initialize all bits of the channel allocation table	-	0x0
2. Select a connection label	CL	N = 0x0 to 0x3F
3. Program the channel descriptor table for channel N	Use the memory interface registers to do a write to the channel descriptor table. See Table 24-1479.	Mask to be used : 0xFFFF FFFF
4. Program the channel allocation table for inbound data buffer RAM access	Write to the data buffer RAM. See Table 24-1480.	-
5. Program the channel allocation table for outbound data buffer RAM access	Read from the data buffer RAM. See Table 24-1481.	-
6. Repeat steps 2-5 to initialize all logical channels	-	-

Table 24-1479. Subsequence - Program The Channel Descriptor Table

Step	Register/ Bit Field/ Programming Model/ Comments	Value
1. Set the 14-bit base address	BA	0x-
2. Set the 12-bit or 13-bit buffer depth	BD (set this according to the BA so that it doesn't overflow the buffer)	BD = buffer depth in bytes - 1
IF: Synchronous channels		
Set the 12-bit or 13-bit buffer depth	BD	(BD + 1) = 4 × frames per sub-buffer (m) × bytes-per-frame (bpf)
ENDIF		
IF: Isochronous channels		
Set the 12-bit or 13-bit buffer depth	BD	(BD + 1) mod (BS + 1) = 0
ENDIF		
IF: Asynchronous channels		
Set the 12-bit or 13-bit buffer depth	BD	(BD + 1) >= max packet length (1024 for a MOST Data Packet (MDP))
ENDIF		
IF: Control channels		
Set the 12-bit or 13-bit buffer depth	BD	(BD + 1) >= max packet length (64)
ENDIF		
3. For isochronous channels, configure the block size	BS	BS = block size in bytes - 1
4. Write to all other bits of the channel descriptor table		0x0

Table 24-1480. Subsequence - Write To The Data Buffer RAM

Step	Register/ Bit Field/ Programming Model/ Comments	Value
For Tx channels the DMA block does the inbound data buffer RAM access	A Tx channel is from MLBSS to external host	-

Table 24-1480. Subsequence - Write To The Data Buffer RAM (continued)

Step	Register/ Bit Field/ Programming Model/ Comments	Value
For Rx channels MediaLB does the inbound data buffer RAM access	An Rx channel is from external host to MLBSS	-
Set the channel direction	RNW	0x0
Set the channel type	CT[2:0]	0x0: synchronous 0x1: control 0x2: asynchronous 0x3: isochronous
Set the connection label	CL[5:0]	CL[5:0] = N
IF: CT[2:0] = 0x0 (synchronous)		
Set the mute bit.	MT (the mute bit is set to avoid any flow of the garbage data that will be there in the data buffer prior to any transfer)	0x1
ENDIF		
Set the channel enable bit	CE	0x1
Clear all other bits of the channel allocation table	-	0x0

Table 24-1481. Subsequence - Read From The Data Buffer RAM

Step	Register/ Bit Field/ Programming Model/ Comments	Value
For Tx channels MediaLB does the outbound data buffer RAM access	A Tx channel is from MLBSS to external host	-
For Rx channels the DMA block does the outbound data buffer RAM access	An Rx channel is from external host to MLBSS	-
Set the channel direction	RNW	0x1
Set the channel type	CT[2:0]	0x0: synchronous 0x1: control 0x2: asynchronous 0x3: isochronous
Set the connection label	CL[5:0]	CL[5:0] = N
IF: CT[2:0] = 000 (synchronous),		
Set the mute bit.	MT	0x1
ENDIF		
Set the channel enable bit	CE	0x1
Clear all other bits of the channel allocation table	-	0x0

Programming The DMA Block

The DMA descriptor table resides in the channel table RAM and is programmed through L4_PER2 slave interface using the registers of the memory interface block. The steps to be followed are shown in Table 24-1482.

Table 24-1482. Programming The DMA Block

Step [(1)]	Register/ Bit Field/ Programming Model/ Comments	Value
1. Initialize all bits of the DMA descriptor table	-	0x0
2. Select a connection label	CL (Note that the CL number chosen in this step should be equal to the CL number chosen in step 2 of Table 24-1478)	N = 0x0 to 0x3F

Table 24-1482. Programming The DMA Block (continued)

Step [(1)]	Register/ Bit Field/ Programming Model/ Comments	Value
3. Program the DMA block ping page for channel N [(2)]	See Table 24-1483	-
4. Program the DMA block pong page for channel N	See Table 24-1484	-
5. Select big or little endian	LE	0x-
6. Select the active page	PG (Program this bit only once in the beginning of the configuration. It shouldn't be touched by the software after that. The hardware updates this bit as the transactions happen)	0x-
7. Set the channel enable bit for all active logical channels	CE	0x1
8. Repeat steps 2-7 for all active logical channels	-	-

(1) **Tip:** In order to program the MLB and DMA channel allocation table entries with respect to corresponding MLB and DMA channel address, the following steps are recommended:
- Given the MLB channel address is X (1-63), the MLB channel allocation table entry to be programmed should be X/2.
- Given the DMA channel address is X (0-63), the DMA channel allocation table entry to be programmed is X.

(2) For programming the 128-bit DMA descriptor table entry,
- IF: The DMA descriptor table is initialized with zeros and the PG bit is intended to be set to 0x0, the following mask should only be used: MLB_MDWE0 = 0x000C 0000
- ELSE: The following steps should be performed:
  – The mask to be used: = 0x000E 0000.
  – Program the DMA descriptor table and the PG bit.
  – Change the MLB_MDWE0 mask value to 0x000C 0000 after the first configuration has been done to prevent any further writes by host to the PG bit during ongoing MLB transactions.
  – Write the 0xFFFF FFFF mask to the MLB_MDWE1, MLB_MDWE2 and MLB_MDWE3 registers

Table 24-1483. Subsequence - Program The DMA Block Ping Page

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Set the 32-bit base address	BA1	0x-
Set the 11-bit buffer depth	BD1	BD1 = buffer depth in bytes - 1
IF: Synchronous channels		
Set the 11-bit buffer depth	BD1	(BD1+ 1) = n × frames per sub-buffer (m) × bytes-per-frame (bpf)
ENDIF		
IF: Isochronous channels		
Set the 11-bit buffer depth	BD1	(BD1 + 1) mod (BS + 1) = 0
ENDIF		
IF: Asynchronous and control Rx channels		
Set the 11-bit buffer depth	BD1	5 <= (BD1 + 1) <= 4096 [(1)]
ENDIF		
IF: Asynchronous and control Tx channels		
Set the 11-bit buffer depth	BD1	5 <= (BD1 + 1) <= 4096 [(1)]
Set the packet start bit if the page contains the start of the packet	PS1	0x1
ENDIF		
Clear the page done bit	DNE1	0x0
Clear the error bit	ERR1	0x0

Table 24-1483. Subsequence - Program The DMA Block Ping Page (continued)

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Set the page ready bit	RDY1	0x1

(1) 4096 is the max packet length.

Table 24-1484. Subsequence - Program The DMA Block Pong Page

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Set the 32-bit base address	BA2	0x-
Set the 11-bit buffer depth	BD2	BD2 = buffer depth in bytes - 1
IF: Synchronous channels		
Set the 11-bit buffer depth	BD2	(BD2+ 1) = n × frames per sub-buffer (m) × bytes-per-frame (bpf)
ENDIF		
IF: Isochronous channels		
Set the 11-bit buffer depth	BD2	(BD2 + 1) mod (BS + 1) = 0
ENDIF		
IF: Asynchronous and control Rx channels		
Set the 11-bit buffer depth	BD2	5 <= (BD2 + 1) <= 4096 (1)
ENDIF		
IF: Asynchronous and control Tx channels		
Set the 11-bit buffer depth	BD2	5 <= (BD2 + 1) <= 4096 (1)
Set the packet start bit if the page contains the start of the packet	PS2	0x1
ENDIF		
Clear the page done bit	DNE2	0x0
Clear the error bit	ERR2	0x0
Set the page ready	RDY2	0x1

(1) 4096 is the max packet length.

Synchronizing And Unmuting The Synchronous Channel

The steps to be followed are shown in Table 24-1485

Table 24-1485. Synchronizing And Unmuting The Synchronous Channel

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Check that MediaLB clock is running	MLB_MLBC1[7] CLKMERR	0x0
IF: The MediaLB clock is not toggling at the pads	MLB_MLBC1[7] CLKMERR	0x1
Clear the register bit	MLB_MLBC1[7] CLKMERR	
Wait one MLB_L4_ICLK or MLB I/O clock cycle		
Check that MediaLB clock is running	MLB_MLBC1[7] CLKMERR	0x0
ENDIF		
Poll for MediaLB lock	MLB_MLBC0[7] MLBLK	0x1
Wait four frames	-	-
Unmute the synchronous channel/ channels	MT	0x0

24.12.5.2 MLBSS Operational Modes Configuration

24.12.5.2.1 Channel Servicing

After initialization, each channel requires periodic servicing. Table 24-1486 shows the steps which can be performed concurrently and in any order.

Table 24-1486. Channel Servicing Steps

Step	Crossreference
Servicing the DMA Interrupts	Table 24-1487
Servicing MLB Interrupts	Table 24-1489
Polling for MediaLB System Commands	Table 24-1490

Servicing the DMA Interrupts

The steps to be followed are shown in Table 24-1487

Table 24-1487. Servicing the DMA Interrupts

Step	Register/ Bit Field/ Programming Model/ Comments	Value
1. Enable (unmask) the interrupts from all active DMA channels	MLB_DCMR0 and MLB_DCMR1	0x1 for each bit associated with an active DMA channel
2. Select the status clear method	MLB_DCTL[0] SCE	0x-
3. Select one (MLB_IRQ_SYS_INT0 only) or two (MLB_IRQ_SYS_INT0 and MLB_IRQ_SYS_INT1) interrupt signals	MLB_DCTL[1] SMX	0x-
4. Wait for an interrupt from MLB_IRQ_SYS_INT0 and/or MLB_IRQ_SYS_INT1 line	-	-
5. Determine which channel or channels are causing an interrupt	Read the MLB_DCSR0 and MLB_DCSR1 registers	0x-
6. IF: MLB_DCTL[0] SCE = 0x1 (software clears the interrupts)		
Write the results of step 5 back to the MLB_DCSR0 and MLB_DCSR1 registers to clear the interrupt	MLB_DCSR0 and MLB_DCSR1	MLB_DCSR0 and MLB_DCSR1
ENDIF		
7. Select a logical channel (N = 0-63) with an interrupt to service	-	-
8. Read the DMA descriptor table entry for channel N	See Table 24-1488	-
9. Reprogram the expired or broken DMA page/pages via steps 3 and 4 of Table 24-1482	-	-
10. Repeat steps 6-9 for all channels with pending interrupts	-	-
11. Repeat steps 4-10 while there are active channels	-	-

Note

The MLB_DCSR0, MLB_DCSR1, MLB_DCMR0 and MLB_DCMR1 register bits correspond to the DMA channel address for a particular logical channel. This should not be confused with the MLB channel address.

Channels that receive a DMA error response are disabled (CE = 0) by hardware.

Table 24-1488. Subsequence - Reading The DMA Descriptor Table Entry

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Determine the active page (ping or pong)	Use the PG bit	0x-
Determine which page/pages are done	Use the DNE1 and DNE2 bits	0x1
Determine which channels encountered a DMA error	Use the ERR1 and ERR2 bits	0x1
Determine which asynchronous and control Rx channel pages contain a packet start	Use the PS1 and PS2 bits	0x1

Servicing MLB Interrupts

To service the MLB interrupts, the steps desctibed in Table 24-1489 should be followed.

Table 24-1489. Servicing MLB Interrupts

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Clear the appropriate MediaLB channel status bits	MLB_MS0 and MLB_MS1 registers	0x0
Enable protocol error interrupts for all active MediaLB channels	MLB_MIEN[28] CTX_PE MLB_MIEN[25] CRX_PE MLB_MIEN[21] ATX_PE MLB_MIEN[18] ARX_PE MLB_MIEN[16] SYNC_PE MLB_MIEN[0] ISOC_PE	0x1 0x1 0x1 0x1 0x1 0x1
Wait for an interrupt on the MLB_IRQ_SYS line		
Determine which channel/channels are causing the interrupt	Read the MLB_MS0 and MLB_MS1 registers	0x1
Determine the interrupt type	Read the RSTS and WSTS bits of the appropriate channel descriptor tables	0x-
IF: Synchronous channels		
Clear WSTS errors to resume channel operation	WSTS[3]	0x0
ENDIF		
IF: Isochronous channels		
Clear WSTS errors to resume channel operation	WSTS[2:1]	0x0
ENDIF		
IF: Asynchronous and control channels		
Clear RSTS and WSTS errors to resume channel operation	RSTS[4], WSTS[4], RSTS[2] and WSTS[2]	0x0
ENDIF		

Polling For MediaLB System Commands

The MediaLB core supports the MediaLB system commands (MlbScan, MlbReset, MOST_Unlock). The MLB_MSS register is used to detect a system command received from the MediaLB controller. The MLB automatically sends the appropriate system response to the MediaLB Controller. Software should follow the procedure shown in Table 24-1490.

Table 24-1490. Polling For MediaLB System Commands

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Poll periodically for a system command received from the MediaLB controller	MLB_MSS	0x-
Clear the appropriate bits in MLB_MSS register after the application finishes the servicing	MLB_MSS	0x0
IF: Software system command is detected	MLB_MSS[4] SWSYSCMD	0x1
Read the MLB_MSD register to receive the system data sent from MediaLB Controller	MLB_MSD	0x-
ENDIF		

24.12.5.2.2 Channel Table RAM Access

The memory interface block allows a direct software access to the channel table RAM. Any write to the MLB_MADR register triggers a single read or write cycle. Reading from the MLB_MADR register does not initiate a read or write access.

Table 24-1491 shows a direct write to the channel table RAM.

Table 24-1491. Direct Channel Table RAM Writes

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Load the 128-bit data entry	MLB_MDAT0, MLB_MDAT1, MLB_MDAT2 and MLB_MDAT3 registers	0x-
Enable writing data	MLB_MDWE0, MLB_MDWE1, MLB_MDWE2 and MLB_MDWE3 registers	0x1
Write the 8-bit channel table RAM target address	MLB_MADR[7:0] ADDR	0x-
Initiate a write cycle	MLB_MADR[31] WNR	0x1
Determine when the transfer is complete	Poll the MLB_MCTL[0] XCMP	0x1

Table 24-1492 shows a direct read from the channel table RAM.

Table 24-1492. Direct Channel Table RAM Reads

Step	Register/ Bit Field/ Programming Model/ Comments	Value
Write the 8-bit channel table RAM target address	MLB_MADR[7:0] ADDR	0x-
Initiate a read cycle	MLB_MADR[31] WNR	0x0
Determine when the transfer is complete	Poll the MLB_MCTL[0] XCMP	0x1
Read the 128-bit data entry	MLB_MDAT0, MLB_MDAT1, MLB_MDAT2 and MLB_MDAT3 registers	0x-

24.12.6 MLB Register Manual

24.12.6.1 MLB Instance Summary

Table 24-1493. MLB Instance Summary

Module Name	Module Base Address	Size
MLB	0x4842 C000	4KiB

24.12.6.2 MLB registers

24.12.6.2.1 MLB Register Summary

Table 24-1494. MLB Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MLB Base Address
MLB_MLBSSREV	R	32	0x0000 0000	0x4842 C000
MLB_MLBSSPWR	RW	32	0x0000 0004	0x4842 C004
MLB_MLBSSPRF	RW	32	0x0000 0100	0x4842 C100
MLB_MLBC0	RW	32	0x0000 0400	0x4842 C400
RESERVED	R	32	0x0000 0408	0x4842 C408
MLB_MS0	RW	32	0x0000 040C	0x4842 C40C
MLB_MS1	RW	32	0x0000 0414	0x4842 C414
MLB_MSS	RW	32	0x0000 0420	0x4842 C420
MLB_MSD	R	32	0x0000 0424	0x4842 C424
MLB_MIEN	RW	32	0x0000 042C	0x4842 C42C
MLB_MLBC1	RW	32	0x0000 043C	0x4842 C43C
MLB_MDAT0	RW	32	0x0000 04C0	0x4842 C4C0
MLB_MDAT1	RW	32	0x0000 04C4	0x4842 C4C4
MLB_MDAT2	RW	32	0x0000 04C8	0x4842 C4C8
MLB_MDAT3	RW	32	0x0000 04CC	0x4842 C4CC
MLB_MDWE0	RW	32	0x0000 04D0	0x4842 C4D0
MLB_MDWE1	RW	32	0x0000 04D4	0x4842 C4D4
MLB_MDWE2	RW	32	0x0000 04D8	0x4842 C4D8
MLB_MDWE3	RW	32	0x0000 04DC	0x4842 C4DC
MLB_MCTL	RW	32	0x0000 04E0	0x4842 C4E0
MLB_MADR	RW	32	0x0000 04E4	0x4842 C4E4
MLB_DIENR	RW	32	0x0000 0480	0x4842 C480
MLB_DICER0	RW	32	0x0000 0488	0x4842 C488
MLB_DICER1	RW	32	0x0000 048C	0x4842 C48C
MLB_DCTL	RW	32	0x0000 07C0	0x4842 C7C0
MLB_DCSR0	RW	32	0x0000 07D0	0x4842 C7D0
MLB_DCSR1	RW	32	0x0000 07D4	0x4842 C7D4
MLB_DCMR0	RW	32	0x0000 07D8	0x4842 C7D8
MLB_DCMR1	RW	32	0x0000 07DC	0x4842 C7DC

24.12.6.2.2 MLB Register Description

Table 24-1495. MLB_MLBSSREV

Address Offset	0x0000 0000		
Physical Address	0x4842 C000	**Instance**	MLB
Description	Revision Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	TI internal data	R	0x-

Table 24-1496. MLB_MLBSSPWR

Address Offset	0x0000 0004

Table 24-1496. MLB_MLBSSPWR (continued)

Physical Address	0x4842 C004	**Instance**	MLB
Description	MLBSS Power management Register		
Type	RW		

31–1	0
RESERVED	MSTANDBY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MSTANDBY	Value to be driven in the MStandby bus of the power management interface Writing a 1 to this bit asserts the MStandby output of MLBSS, thereby initiating the clock disabling sequence for the MLBSS. Write 0 to this register to enable the clock for MLBSS. 0x0: Mstandby output is deasserted 0x1: Mstandby output is asserted	RW	0x0

Table 24-1497. MLB_MLBSSPRF

Address Offset	0x0000 0100		
Physical Address	0x4842 C100	**Instance**	MLB
Description	This register is used to define the values of MFLAG pressure to on-chip network, MREQINFO priority to EMIF, and the non-posted write behavior of the L3_MAIN DMA master interface.		
Type	RW		

31–17	16	15	14–12	11	10–8	7–6	5–4	3–2	1–0
RESERVED	WRNP	RESERVED	ASYNC_PRI	RESERVED	SYNC_PRI	RESERVED	ASYNC_FLAG	RESERVED	SYNC_FLAG

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
16	WRNP	The WRNP bit controls whether the writes issued by the DMA OCP interface are posted (no write reponse required to complete transaction) or non posted (write response required to complete transaction). 0x0: Only posted writes are issued. This means that the performance of DMA OCP writes would be better, but the writes will complete before the data gets written to the destination. Posted writes do not require a write response from the destination to complete a write transaction. 0x1: Only non-posted writes are issued. This means that the performance of DMA OCP writes would be lesser, but would guarantee that when a write has completed, the data gets written into the final destination. Non posted writes require a write response from the destination to complete a write transaction. It is recommended to use only posted writes for better performance. This is a static configuration out of reset based on performance requirements and changing it while a transfer in progress could cause indeterminate behavior.	RW	0x0
15	RESERVED		R	0x0
14:12	ASYNC_PRI	ASYNC_PRI controls the priority carried in MREQINFO attribute of OCP DMA interface, when a asynchronous transaction is requested at the DMA interface. It is recommended that the ASYNC_PRI be set at lower priority compared to other masters in the system, as these transactions are not time-critical. 0x0: highest priority 0x7: lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	SYNC_PRI	SYNC_PRI controls the priority carried in MREQINFO attribute of OCP interface, when a synchronous transaction is requested at the DMA interface. It is recommended that the SYNC_PRI be set at a reasonably higher priority compared to other masters in the system, as these transactions could be time-critical. 0x0: highest priority 0x7: lowest priority	RW	0x0
7:6	RESERVED		R	0x0
5:4	ASYNC_FLAG	ASYNC_FLAG controls the value carried in MFLAG attribute of OCP DMA interface. This attribute is used in determining the priority of these transactions through the L3 system infrastructure. It is recommended that he ASYNC_FLAG be set at lower priority compared to other masters in the system, as these transactions are not time critical 0x0: Lowest priority through on chip network 0x1: Medium priority through on chip network 0x2: Reserved 0x3: Highest priority through on-chip network	RW	0x1
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	SYNC_FLAG	SYNC_FLAG controls the value carried in MFLAG attribute of OCP DMA interface. This attribute is used in determining the priority of these transactions through the L3 system infrastructure. It is recommended that the SYNC_FLAG be set at reasonably higher priority compared to other masters in the system, as these transactions are time critical. 0x0: Lowest priority through on chip network 0x1: Medium priority through on chip network 0x2: Reserved 0x3: Highest priority through on-chip network	RW	0x0

Table 24-1498. MLB_MLBC0

Address Offset	0x0000 0400		
Physical Address	0x4842 C400	**Instance**	MLB
Description	MediaLB Control 0 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														FCNT			CTLRETRY	RESERVED	ASYRETRY	RESERVED				MLBLK	RESERVED	MLBPEN	MLBCLK			RESERVED	MLBEN

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:15	FCNT	The number of frames per sub-buffer for synchronous channels 0x0: 1 frame per sub-buffer (operation is the same as standard mode) 0x1: 2 frames per sub-buffer 0x2: 4 frames per sub-buffer 0x3: 8 frames per sub-buffer 0x4: 16 frames per sub-buffer 0x5: 32 frames per sub-buffer 0x6: 64 frames per sub-buffer 0x7: Reserved	RW	0x0
14	CTLRETRY	Control Tx packet retry. When cleared, a control packet that is flagged with a Break or ProtocolError by the receiver is skipped. When set, a control packet that is flagged with a Break or ProtocolError by the receiver is re-transmitted. 0x0: Control packet is skipped 0x1: Control packet is re-transmitted	RW	0x0
13	RESERVED		R	0x0
12	ASYRETRY	Asynchronous Tx packet retry. When cleared, an asynchronous packet that is flagged with a Break or ProtocolError by the receiver is skipped. When set, an asynchronous packet that is flagged with a Break or ProtocolError by the receiver is re-transmitted. 0x0: Asynchronous packet is skipped 0x1: Asynchronous packet is re-transmitted	RW	0x0

Bits	Field Name	Description	Type	Reset
11:8	RESERVED		R	0x0
7	MLBLK	MediaLB lock status. When set, indicates that the MediaLB block is synchronized to the incoming MediaLB frame. If MLBLK is clear (unlocked), MLBLK is set after FRAMESYNC is detected at the same position for three consecutive frames. If MLBLK is set (locked), MLBLK is cleared after not receiving FRAMESYNC at the expected time for two consecutive frames. While MLBLK is set, FRAMESYNC patterns occurring at locations other than the expected one are ignored. (read-only) 0x0: MediaLB is unlocked 0x1: MediaLB is locked	R	0x0
6	RESERVED		R	0x0
5	MLBPEN	MediaLB 6-pin enable. 0x0: MediaLB 3-pin interface enabled 0x1: MediaLB 6-pin interface enabled	RW	0x0
4:2	MLBCLK	MediaLB clock speed select. 0x0: 256×Fs (for MLBPEN = 0) 0x1: 512×Fs (for MLBPEN = 0) 0x2: 1024×Fs (for MLBPEN = 0) 0x3: 2048xFs (For MLBPEN = 1) 0x4-0x7: Reserved	RW	0x0
1	RESERVED		R	0x0
0	MLBEN	MediaLB enable. When set, MediaLB clock, signal, and data are received and transmitted on the appropriate MediaLB pins. 0x0: MediaLB disabled 0x1: MediaLB enabled	RW	0x0

Table 24-1499. MLB_MS0

Address Offset	0x0000 040C		
Physical Address	0x4842 C40C	**Instance**	MLB
Description	MediaLB Channel Status 0 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCS																															

Bits	Field Name	Description	Type	Reset
31:0	MCS	MediaLB channel status. Indicates the channel status for MediaLB channels 31 to 0. Channel status bits are set by hardware and cleared by software. Status is only set if the appropriate bits in the MIEN register are set. 0x0: A channel status bit is cleared (applies for all bits in this register) 0x1: A channel status bit is set (applies for all bits in this register)	RW W0toClr	0x0

Table 24-1500. MLB_MS1

Address Offset	0x0000 0414		
Physical Address	0x4842 C414	**Instance**	MLB

Table 24-1500. MLB_MS1 (continued)

Description	MediaLB Channel Status 1 Register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCS																															

Bits	Field Name	Description	Type	Reset
31:0	MCS	MediaLB channel status. Indicates the channel status for MediaLB channels 63 to 32. Channel status bits are set by hardware and cleared by software. Status is only set if the appropriate bits in the MIEN register are set. 0x0: A channel status bit is cleared (applies for all bits in this register) 0x1: A channel status bit is set (applies for all bits in this register)	RW W0toClr	0x0

Table 24-1501. MLB_MSS

Address Offset	0x0000 0420		
Physical Address	0x4842 C420	**Instance**	MLB
Description	MediaLB System Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										SERVREQ	SWSYSCMD	CSSYSCMD	ULKSYSCMD	LKSYSCMD	RSTSYSCMD

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	SERVREQ	Service request enabled. When cleared, the MediaLB block responds with a "device present" system response. When set, the MediaLB block responds with a "device present, request service" system response if a matching channel scan system command is detected. 0x0: Service request disabled (not detected) 0x1: Service request enabled (detected)	RW	0x0
4	SWSYSCMD	Software system command detected (in the system quadlet). Set by hardware, cleared by software. Data is stored in the MLB_MSD register for this command. 0x0: Software system command not detected 0x1: Software system command detected	RW W0toClr	0x0
3	CSSYSCMD	Channel scan system command detected (in the system quadlet). Set by hardware, cleared by software. If the node address specified in Data quadlet matches the value in MLBC1.NDA, the device responds either "device present" or "device present, request service" system response in the next system quadlet. 0x0: Channel scan system command not detected 0x1: Channel scan system command detected	RW W0toClr	0x0

Bits	Field Name	Description	Type	Reset
2	ULKSYSCMD	Network unlock system command detected (in the system quadlet). Set by hardware, cleared by software 0x0: Unlock system command not detected 0x1: Unlock system command detected	RW W0toClr	0x0
1	LKSYSCMD	Network lock system command detected (in the system quadlet). Set by hardware, cleared by software. 0x0: Lock system not detected 0x1: Lock system detected	RW W0toClr	0x0
0	RSTSYSCMD	Reset system command detected (in the system quadlet). Set by hardware, cleared by software 0x0: Reset system command not detected 0x1: Reset system command detected	RW W0toClr	0x0

Table 24-1502. MLB_MSD

Address Offset	0x0000 0424		
Physical Address	0x4842 C424	**Instance**	MLB
Description	MediaLB System Data Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SD3								SD2								SD1								SD0							

Bits	Field Name	Description	Type	Reset
31:24	SD3	System data (byte 3). Updated with MediaLB Data[31:24] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD3 is not updated. (read-only). As soon as it is serviced, the MSS.SWSYSCMD bit should be cleared.	R	0x0
23:16	SD2	System data (byte 2). Updated with MediaLB Data[23:16] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD2 is not updated. (read-only). As soon as it is serviced, the MSS.SWSYSCMD bit should be cleared.	R	0x0
15:8	SD1	System data (byte 1). Updated with MediaLB Data[15:8] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD1 is not updated. (read-only). As soon as it is serviced, the MSS.SWSYSCMD bit should be cleared.	R	0x0
7:0	SD0	System data (byte 0). Updated with MediaLB Data[7:0] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD0 is not updated. (read-only). As soon as it is serviced, the MSS.SWSYSCMD bit should be cleared.	R	0x0

Table 24-1503. MLB_MIEN

Address Offset	0x0000 042C		
Physical Address	0x4842 C42C	**Instance**	MLB
Description	MediaLB Interrupt Enable Register		
Type	RW		

31 30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 14 13 12 11 10 9 8 7 6 5 4 3 2	1	0
RESERVED	CTX_BREAK	CTX_PE	CTX_DONE	CRX_BREAK	CRX_PE	CRX_DONE	RESERVED	ATX_BREAK	ATX_PE	ATX_DONE	ARX_BREAK	ARX_PE	ARX_DONE	SYNC_PE	RESERVED	ISOC_BUFO	ISOC_PE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	CTX_BREAK	Control Tx break enable. When set, a ReceiverBreak response received from the receiver on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Control Tx break disabled 0x1: Control Tx break enabled	RW	0x0
28	CTX_PE	Control Tx protocol error enable. When set, a ProtocolError generated by the receiver on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Control Tx protocol error disabled 0x1: Control Tx protocol error enabled	RW	0x0
27	CTX_DONE	Control Tx packet done enable. When set, a packet transmitted with no errors on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Control Tx packet done disabled 0x1: Control Tx packet done enabled	RW	0x0
26	CRX_BREAK	Control Rx break enable. When set, a ControlBreak command received from the transmitter on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Control Rx break disabled 0x1: Control Rx break enabled	RW	0x0
25	CRX_PE	Control Rx protocol error enable. When set, a ProtocolError detected on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Control Rx protocol error disabled 0x1: Control Rx protocol error enabled	RW	0x0
24	CRX_DONE	Control Rx packet done enable. When set, a packet received with no errors on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Control Rx packet done disabled 0x1: Control Rx packet done enabled	RW	0x0
23	RESERVED		R	0x0
22	ATX_BREAK	Asynchronous Tx break enable. When set, a ReceiverBreak response received from the receiver on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set 0x0: Asynchronous Tx break disabled 0x1: Asynchronous Tx break enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
21	ATX_PE	Asynchronous Tx protocol error enable. When set, a ProtocolError generated by the receiver on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Asynchronous Tx protocol error disabled 0x1: Asynchronous Tx protocol error enabled	RW	0x0
20	ATX_DONE	Asynchronous Tx packet done enable. When set, a packet transmitted with no errors on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Asynchronous Tx packet done disabled 0x1: Asynchronous Tx packet done enabled	RW	0x0
19	ARX_BREAK	Asynchronous Rx break enable. When set, a AsyncBreak command received from the transmitter on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Asynchronous Rx break disabled 0x1: Asynchronous Rx break enabled	RW	0x0
18	ARX_PE	Asynchronous Rx protocol error enable. When set, a ProtocolError detected on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Asynchronous Rx protocol error disabled 0x1: Asynchronous Rx protocol error enabled	RW	0x0
17	ARX_DONE	Asynchronous Rx packet done enable. When set, a packet received with no errors on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Asynchronous Rx packet done disabled 0x1: Asynchronous Rx packet done enable	RW	0x0
16	SYNC_PE	Synchronous protocol error enable. When set, a ProtocolError detected on a synchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Synchronous protocol error disabled 0x1: Synchronous protocol error enabled	RW	0x0
15:2	RESERVED		R	0x0
1	ISOC_BUFO	Isochronous Rx buffer overflow enable. When set, a buffer overflow on an isochronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. This occurs only when isochronous flow control is disabled. 0x0: Buffer overflow disabled 0x1: Buffer overflow enabled	RW	0x0
0	ISOC_PE	Isochronous Rx protocol error enable. When set, a ProtocolError detected on an isochronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. 0x0: Isochronous Rx ProtocolError disabled 0x1: Isochronous Rx ProtocolError enabled	RW	0x0

Table 24-1504. MLB_MLBC1

Address Offset	0x0000 043C		
Physical Address	0x4842 C43C	**Instance**	MLB
Description	MediaLB Control 1 Register		
Type	RW		

31:16	15:8	7	6	5:0
RESERVED	NDA	CLKMERR	LOCKERR	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	NDA	Node device address. Used for system commands directed to individual MediaLB nodes. All values from 0x00 to 0xFF can be used.	RW	0x0
7	CLKMERR	MediaLB clock missing status. Set when MediaLB clock is not toggling at the pin, cleared by software 0x0: MediaLB Clock is toggling at the pins. 0x1: MediaLB clock is not toggling at the pins.	RW W0toClr	0x0
6	LOCKERR	MediaLB lock error status. Set when MediaLB is unlocked, cleared by software 0x0: No MediaLB lock error. Implies MediaLB is properly locked. 0x1: MediaLB lock error. Implies MediaLB is unlocked. Write 0 to clear.	RW W0toClr	0x0
5:0	RESERVED		R	0x0

Table 24-1505. MLB_MDAT0

Address Offset	0x0000 04C0		
Physical Address	0x4842 C4C0	**Instance**	MLB
Description	Memory Interface Data 0 Register		
Type	RW		

31:0
DATA

Bits	Field Name	Description	Type	Reset
31:0	DATA	CTR data - bits[31:0] of 128-bit entry or DBR data - bits[7:0] of 8-bit entry	RW	0x0

Table 24-1506. MLB_MDAT1

Address Offset	0x0000 04C4		
Physical Address	0x4842 C4C4	**Instance**	MLB
Description	Memory Interface Data 1 Register		
Type	RW		

31:0
DATA

Bits	Field Name	Description	Type	Reset
31:0	DATA	CTR data - bits[63:32] of 128-bit entry	RW	0x0

Table 24-1507. MLB_MDAT2

Address Offset	0x0000 04C8		
Physical Address	0x4842 C4C8	**Instance**	MLB
Description	Memory Interface Data 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	CTR data - bits[95:64] of 128-bit entry	RW	0x0

Table 24-1508. MLB_MDAT3

Address Offset	0x0000 04CC		
Physical Address	0x4842 C4CC	**Instance**	MLB
Description	Memory Interface Data 3 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	CTR data - bits[127:96] of 128-bit entry	RW	0x0

Table 24-1509. MLB_MDWE0

Address Offset	0x0000 04D0		
Physical Address	0x4842 C4D0	**Instance**	MLB
Description	Memory Interface Data Write Enable 0 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK																															

Bits	Field Name	Description	Type	Reset
31:0	MASK	Bitwise write enable for CTR data - bits[31:0] 0x0: Disabled (applies for all bits in this register) 0x1: Enabled (applies for all bits in this register)	RW	0x0

Table 24-1510. MLB_MDWE1

Address Offset	0x0000 04D4		
Physical Address	0x4842 C4D4	**Instance**	MLB
Description	Memory Interface Data Write Enable 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK																															

Bits	Field Name	Description	Type	Reset
31:0	MASK	Bitwise write enable for CTR data - bits[63:32] 0x0: Disabled (applies for all bits in this register) 0x1: Enabled (applies for all bits in this register)	RW	0x0

Table 24-1511. MLB_MDWE2

Address Offset	0x0000 04D8		
Physical Address	0x4842 C4D8	**Instance**	MLB
Description	Memory Interface Data Write Enable 2 Register		
Type	RW		

31–0
MASK

Bits	Field Name	Description	Type	Reset
31:0	MASK	Bitwise write enable for CTR data - bits[95:64] 0x0: Disabled (applies for all bits in this register) 0x1: Enabled (applies for all bits in this register)	RW	0x0

Table 24-1512. MLB_MDWE3

Address Offset	0x0000 04DC		
Physical Address	0x4842 C4DC	**Instance**	MLB
Description	Memory Interface Data Write Enable 3 Register		
Type	RW		

31–0
MASK

Bits	Field Name	Description	Type	Reset
31:0	MASK	Bitwise write enable for CTR data - bits[127:96] 0x0: Disabled (applies for all bits in this register) 0x1: Enabled (applies for all bits in this register)	RW	0x0

Table 24-1513. MLB_MCTL

Address Offset	0x0000 04E0		
Physical Address	0x4842 C4E0	**Instance**	MLB
Description	Memory Interface Control Register		
Type	RW		

31–1	0
RESERVED	XCMP

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	XCMP	Transfer complete (write 0 to clear) 0x0: Memory interface transfer not completed 0x1: Memory interface transfer is completed	RW W0toClr	0x0

Table 24-1514. MLB_MADR

Address Offset	0x0000 04E4

Table 24-1514. MLB_MADR (continued)

Physical Address	0x4842 C4E4	**Instance**	MLB
Description	Memory Interface Address Register		
Type	RW		

31	30:8	7:0
WNR	RESERVED	ADDR

Bits	Field Name	Description	Type	Reset
31	WNR	Write-Not-Read selection 0x0: Read 0x1: Write	RW	0x0
30:8	RESERVED		R	0x0
7:0	ADDR	CTR address of 128-bit entry. All values from 0x00 to 0xFF can be used.	RW	0x0

Table 24-1515. MLB_DIENR

Address Offset	0x0000 0480		
Physical Address	0x4842 C480	**Instance**	MLB
Description	Internal DMA Enable Register		
Type	RW		

31:16	15	14:0
RESERVED	EN	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	EN	DMA enable. Always program to 1. 0x0: Disabled 0x1: Enabled	RW	0x0
14:0	RESERVED		R	0x0

Table 24-1516. MLB_DICER0

Address Offset	0x0000 0488		
Physical Address	0x4842 C488	**Instance**	MLB
Description	Internal DMA Channel Enable Register 0		
Type	RW		

31:0
CHE

Bits	Field Name	Description	Type	Reset
31:0	CHE	Bitwise channel enable. Bits [31:0]. Always program to 1. 0x0: Disabled (applies for all bits in this register) 0x1: Enabled (applies for all bits in this register)	RW	0x0

Table 24-1517. MLB_DICER1

Address Offset	0x0000 048C		
Physical Address	0x4842 C48C	**Instance**	MLB
Description	Internal DMA Channel Enable Register 1		

Table 24-1517. MLB_DICER1 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHE																															

Bits	Field Name	Description	Type	Reset
31:0	CHE	Bitwise channel enable. Bits [63:32]. Always program to 1. 0x0: Disabled (applies for all bits in this register) 0x1: Enabled (applies for all bits in this register)	RW	0x0

Table 24-1518. MLB_DCTL

Address Offset	0x0000 07C0		
Physical Address	0x4842 C7C0	**Instance**	MLB
Description	DMA Control Register. This register is written by the host to configure the DMA block for channel interrupts. It contains three configuration fields. One used to select the DMA mode, one used to mux channel interrupts onto a single interrupt signal, and the last selects the method of clearing channel interrupts (either software or hardware).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											PKT_MODE	RESERVED	DMA_MODE	SMX	SCE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	PKT_MODE	Packet mode for async/control packets. 0x0: Single packet mode 0x1: Multi Packet mode	RW	0x0
3	RESERVED		R	0x0
2	DMA_MODE	DMA mode. 0x0: DMA Mode 0 (Not supported) 0x1: DMA Mode 1 (Use always this value)	RW	0x0
1	SMX	DMA interrupt mux enable. 0x0: The MLB_DCSR0 register generates an interrupt on the MLB_IRQ_SYS_INT0 line and MLB_DCSR1 register generates an interrupt on the MLB_IRQ_SYS_INT1 0x1: The MLB_DCSR0 and MLB_DCSR1 registers generate an interrupt on MLB_IRQ_SYS_INT0 only	RW	0x0
0	SCE	Software clear enable. 0x0: Hardware clears interrupt after the MLB_DCSR0 and MLB_DCSR1 registers are read 0x1: Software clears interrupt	RW	0x0

Table 24-1519. MLB_DCSR0

Address Offset	0x0000 07D0		
Physical Address	0x4842 C7D0	**Instance**	MLB
Description	DMA Control Status 0 Register		

Table 24-1519. MLB_DCSR0 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHS																															

Bits	Field Name	Description	Type	Reset
31:0	CHS	Interrupt status for logical channels 31 to 0. 0x0: No interrupt (applies for all bits in this register) 0x1: Interrupt (applies for all bits in this register)	RW W1toClr	0x0

Table 24-1520. MLB_DCSR1

Address Offset	0x0000 07D4		
Physical Address	0x4842 C7D4	**Instance**	MLB
Description	DMA Control Status 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHS																															

Bits	Field Name	Description	Type	Reset
31:0	CHS	Interrupt status for logical channels 63 to 32. 0x0: No interrupt (applies for all bits in this register) 0x1: Interrupt (applies for all bits in this register)	RW W1toClr	0x0

Table 24-1521. MLB_DCMR0

Address Offset	0x0000 07D8		
Physical Address	0x4842 C7D8	**Instance**	MLB
Description	DMA Channel Mask 0 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHM																															

Bits	Field Name	Description	Type	Reset
31:0	CHM	Bitwise channel mask. Bits [31:0]. 0x0: Masked (applies for all bits in this register) 0x1: Unmasked (applies for all bits in this register)	RW	0x0

Table 24-1522. MLB_DCMR1

Address Offset	0x0000 07DC		
Physical Address	0x4842 C7DC	**Instance**	MLB
Description	DMA Channel Mask 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHM																															

Bits	Field Name	Description	Type	Reset
31:0	CHM	Bitwise channel mask. Bits [63:32]. 0x0: Masked (applies for all bits in this register) 0x1: Unmasked (applies for all bits in this register)	RW	0x0

Chapter 25

eMMC/SD/SDIO

TEXAS INSTRUMENTS

This chapter describes the features and functions of the eMMC/SD/SDIO interface of the device.

25.1 eMMC/SD/SDIO Overview 6381
25.2 eMMC/SD/SDIO Environment 6385
25.3 eMMC/SD/SDIO Integration 6392
25.4 eMMC/SD/SDIO Functional Description 6398
25.5 eMMC/SD/SDIO Programming Guide 6430
25.6 eMMC/SD/SDIO Register Manual 6455

25.1 eMMC/SD/SDIO Overview

The eMMC/SD/SDIO host controller provides an interface between a local host (LH) such as a microprocessor unit (MPU) or digital signal processor (DSP) and either eMMC, SD® memory cards, or SDIO cards and handles eMMC/SD/SDIO transactions with minimal LH intervention.

Optionally, the controller is connected to the L3_MAIN interconnect to have a direct access to system memory. It also supports two direct memory access (DMA) slave channels or a DMA master access (in this case, slave DMA channels are deactivated) depending on its integration.

The eMMC/SD/SDIO host controller deals with eMMC/SD/SDIO protocol at transmission level, data packing, adding cyclic redundancy checks (CRCs), start/end bit, and checking for syntactical correctness.

The application interface can send every eMMC/SD/SDIO command and poll for the status of the adapter or wait for an interrupt request, which is sent back in case of exceptions or to warn of end of operation.

The application interface can read card responses or flag registers. It can also mask individual interrupt sources. All these operations can be performed by reading and writing control registers. The eMMC/SD/SDIO host controller also supports two DMA channels.

There are four eMMC/SD/SDIO host controllers inside the device. Figure 25-1 gives an overview of the eMMC/SD/SDIOi (i = 1 to 4) controllers.

Each controller has the following data width:

- eMMC/SD/SDIO1 - 4-bit wide data bus
- eMMC/SD/SDIO2 - 8-bit wide data bus
- eMMC/SD/SDIO3 - 8-bit wide data bus
- eMMC/SD/SDIO4 - 4-bit wide data bus

The eMMC/SD/SDIOi controller is also referred to as MMCi.

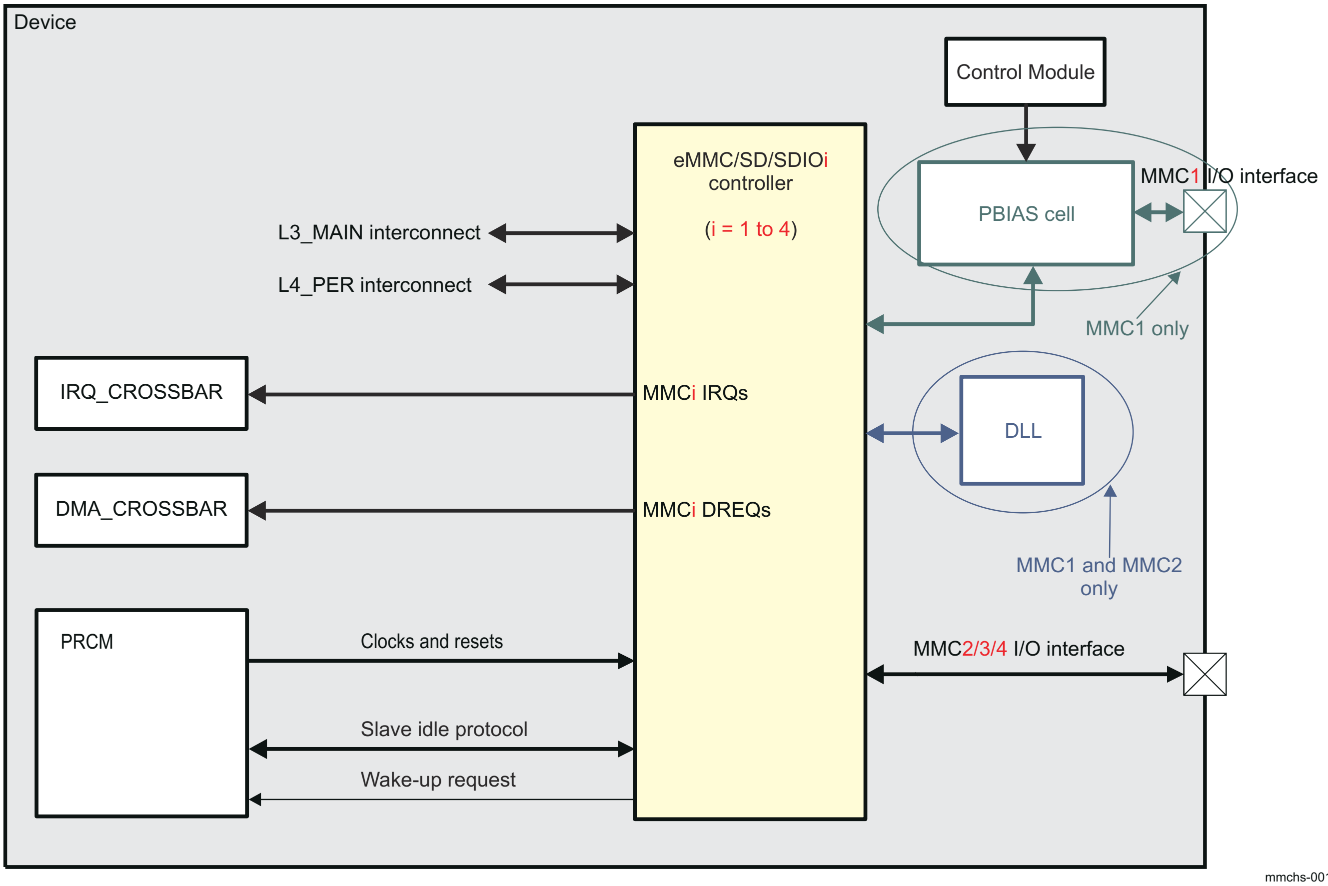


Figure 25-1. eMMC/SD/SDIOi Overview (i = 1 to 4)

25.1.1 eMMC/SD/SDIO Features

This section describes the features supplied by the eMMC/SD/SDIO controllers.

Compliance with standards:

- Full compliance with MMC/eMMC command/response sets as defined in the JC64 MMC/eMMC standard specification, v4.5.
- Full compliance with SD command/response sets as defined in the SD Physical Layer specification v3.01
- Full compliance with SDIO command/response sets and interrupt/read-wait suspend-resume operations as defined in the SD part E1 specification v3.00
- Full compliance with SD Host Controller Standard Specification sets as defined in the SD card specification Part A2 v3.00

Main features of the eMMC/SD/SDIO host controllers:

- Flexible architecture allowing support for new command structure
- 32-bit wide access bus to maximize bus throughput
- Designed for low power
- Programmable clock generation
- Dedicated DLL to support SDR104 mode (MMC1 only)
- Dedicated DLL to support HS200 mode (MMC2 only)
- Card insertion/removal detection and write protect detection
- L4 slave interface supports:
  - 32-bit data bus width
  - 8/16/32 bit access supported
  - 9-bit address bus width

  - Streaming burst supported only with burst length up to 7
  - WNP supported
- L3 initiator interface Supports:
  - 32-bit data bus width
  - 8/16/32 bit access supported
  - 32-bit address bus width
  - Burst supported
- Built-in 1024-byte buffer for read or write
- Two DMA channels, one interrupt line
- Support JC 64 v4.4.1 boot mode operations
- Support SDA 3.00 Part A2 programming model
- Support SDA 3.00 Part A2 DMA feature (ADMA2)
- Supported data transfer rates:
  - MMCi supports the following SD v3.0 data transfer rates:
    - DS mode (3.3V IOs): up to 12 MBps (24 MHz clock)
    - HS mode (3.3V IOs): up to 24 MBps (48 MHz clock)
    - SDR12 (1.8V IOs): up to 12 MBps (24 MHz clock)
    - SDR25 (1.8V IOs): up to 24 MBps (48 MHz clock)
    - SDR50 (1.8V IOs): up to 48 MBps (96 MHz clock) - MMC1 and MMC3 only
    - DDR50 (1.8V IOs): up to 48 MBps (48 MHz clock) - MMC1 only
    - SDR104 (1.8V IOs) cards can be supported up to 192 MHz clock (96 MBps max) - MMC1 only
  - MMCi supports the Default SD mode 1-bit data transfer up to 24Mbps (3MBps)
  - Only MMC2 supports also the following JC64 v4.5 data transfer rates:
    - Up to 192 MBps in eMMC mode, 8-bit SDR mode (192 MHz clock frequency)
    - Up to 96 MBps in eMMC mode, 8-bit DDR mode (48 MHz clock frequency)
- All eMMC/SD/SDIO controllers are connected to 1,8V/3.3V compatible I/Os to support 1,8V/3.3V signaling

Note

eMMC functionality is supported fully by MMC2 only. The other MMC modules are capable of eMMC functionality, but are not timing-optimized for eMMC. For more information about timing limitations, see the data manual of the device.

The differences between the eMMC/SD/SDIO host controllers and a standard SD host controller defined by the *SD Card Specification, Part A2, SD Host Controller Standard Specification*, v3.00 are:

- The clock divider in the eMMC/SD/SDIO host controller supports a wider range of frequency than specified in the *SD Memory Card Specifications*, v3.0. The eMMC/SD/SDIO host controller supports odd and even clock ratio.
- The eMMC/SD/SDIO host controller supports configurable busy time-out.
- ADMA2 64-bit mode is not supported.
- There is no external LED control.

Note

Only even ratios are supported in DDR mode.

Table 25-1 lists the features supported in the 4.5 standard.

Table 25-1. Standard 4.5 Supported Features

Feature	Support	Limitation	Comment
Bus width	1-bit mode		x 1, 4, 8 bits
	4-bit mode		
	8-bit mode		

Table 25-1. Standard 4.5 Supported Features (continued)

Feature	Support	Limitation	Comment
Support density	No hardware limitation for density support	Limitation can come from file system (32 GiB).	
Simple boot (CMD, alternate boot)	Yes		Device ROM code supports the following boot modes: 1. Alternate Boot 2. Raw (UDA) Boot 3. File System For more information about boot modes, see *Memory Booting*. in *Initialization.*
Sleep mode	Yes		
Reliable write	Yes		
Secure write protection	Yes		
Hardware reset	No		Use the reset command if hardware reset is needed.
Secure memory block (RPMB)	Yes		
Partition feature	Yes		
Secure erase	Yes		
DDR interface (bandwidth)	Up to 96 MBps in DDR mode – 8-bit		
High-priority interrupt (read while write)	Yes		
Background operation	Yes		
Enhanced reliable write	Yes		
HS200 mode	Yes		

Table 25-2 shows the supported by each MMCi host controller transfer rates and functionalities (SD, eMMC and SDIO).

Table 25-2. MMCi Supported Transfer Rates and Functionalities

	SD[1]	eMMC	SDIO
MMC1	Yes (Up to SDR104 mode)	Yes (Up to High Speed DDR mode; timings optimized for SD)	Yes (timings optimized for SD)
MMC2	Yes (Up to SDR25 mode; timings optimized for eMMC)	Yes (Up to HS200 mode)	Yes (timings optimized for eMMC)
MMC3	Yes (Up to SDR50 mode)	Yes (Up to High Speed SDR mode; timings optimized for SD/ SDIO)	Yes (Up to SDR50 mode)
MMC4	Yes (Up to SDR25 mode)	Yes (Up to High Speed SDR mode; timings optimized for SD/ SDIO)	Yes (Up to SDR25 mode)

(1) 3.3V IO required for initial SD Card communication.

25.2 eMMC/SD/SDIO Environment

One eMMC/SD/SDIO host controller can support one eMMC memory card, one SD memory card, or one SDIO card.

Other combinations (for example, two SD cards, one eMMC, and one SD card) are not supported through a single controller. The following is supported:

- The MMC1 instance supports 1- and 4-bit data transfers. (mainly used for connection with SD cards)
- The MMC2 instance supports 1-, 4-, and 8-bit data transfers. (mainly used for connection with eMMC cards)
- The MMC3 instance supports 1-, 4-, and 8-bit data transfers. (mainly used for connection with SDIO cards)
- The MMC4 instance supports 1- and 4-bit data transfers. (mainly used for connection with SDIO cards)

25.2.1 eMMC/SD/SDIO Functional Modes

25.2.1.1 eMMC/SD/SDIO Connected to an eMMC, SD, or SDIO Card

Figure 25-2 shows the eMMC/SD/SDIOi host controller (where i = 1 to 4) connected to an eMMC, SD, or SDIO card and its related external connections.

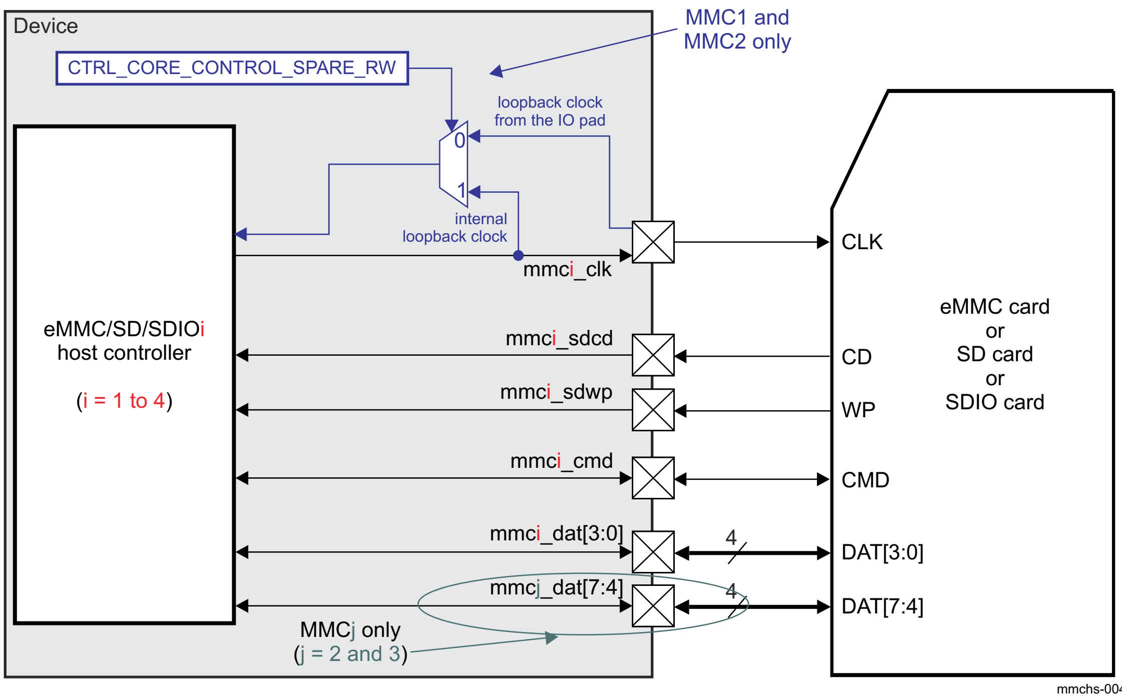


Figure 25-2. eMMC/SD/SDIOi Controller Connected to an eMMC, SD, or SDIO Card (where i = 1 to 4)

Table 25-3 describes the eMMC/SD/SDIOi host controller I/O's (where i = 1 to 4).

Table 25-3. Description of eMMC/SD/SDIOi host controller I/O's (where i = 1 to 4)

Instance	Signal name	I/O[1]	Description	Reset Value[2]
MMC1	mmc1_clk	I/O	External clock for eMMC/SD/SDIO card	0
	mmc1_cmd	I/O	Command line	Hi-Z[3]
	mmc1_dat[3:0]	I/O	Data signals	Hi-Z[3]
	mmc1_sdcd	I	Card insertion/removal detection signal	Hi-Z
	mmc1_sdwp	I	Write protect detection signal	Hi-Z

Table 25-3. Description of eMMC/SD/SDIOi host controller I/O's (where i = 1 to 4) (continued)

Instance	Signal name	I/O[(1)]	Description	Reset Value[(2)]
MMC2	mmc2_clk	I/O	External clock for eMMC/SD/SDIO card	0
	mmc2_cmd	I/O	Command line	Hi-Z[(3)]
	mmc2_dat[7:0]	I/O	Data signals	Hi-Z[(3)]
	mmc2_cd	I	Card insertion/removal detection signal	Hi-Z
	mmc2_wp	I	Write protect detection signal	Hi-Z
MMC3	mmc3_clk	I/O	External clock for eMMC/SD/SDIO card	0
	mmc3_cmd	I/O	Command line	Hi-Z[(3)]
	mmc3_dat[7:0]	I/O	Data signals	Hi-Z[(3)]
	mmc3_cd	I	Card insertion/removal detection signal	Hi-Z
	mmc3_wp	I	Write protect detection signal	Hi-Z
MMC4	mmc4_clk	I/O	External clock for eMMC/SD/SDIO card	0
	mmc4_cmd	I/O	Command line	Hi-Z[(3)]
	mmc4_dat[3:0]	I/O	Data signals	Hi-Z[(3)]
	mmc4_cd	I	Card insertion/removal detection signal	Hi-Z
	mmc4_wp	I	Write protect detection signal	Hi-Z

(1) I = Input; O = Output; I/O = Bidirectional
(2) Hi-Z = High Impedance
(3) Initialized as input upon reset

Note

For mmc2_clk, mmc3_clk and mmc4_clk signals to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers must be set to 0x1 by software. This is because the eMMC/SD/SDIO controller uses the input from the pad as loopback clock, which the controller can use for read capture depending on the mode it is in.

Note

On SR2.0 devices the internal PU/PD resistors on pads mmc2_dat[7:0] can be permanently disabled. For more information, see *Permanent PU/PD disabling (SR 2.0 only)* in *Control Module*.

25.2.2 Protocol and Data Format

The bus protocol between the eMMC/SD/SDIOi host controller and the card is message-based. Each message is represented by one of the following parts:

- Command: A command starts an operation. The command is transferred serially from the eMMC/SD/SDIO host controller to the card on the CMD line.
- Response: A response is an answer to a command. The response is sent from the card to the eMMC/SD/SDIO host controller. It is transferred serially on the CMD line.
- Data: Data are transferred from the eMMC/SD/SDIO host controller to the card or from a card to the eMMC/SD/SDIO host controller using the data lines.
- Busy: The DAT[0] signal is maintained low by the card as far as it is programming the data received.

- CRC status: The CRC result is sent by the card through the DAT[0] line when executing a write transfer. In the case of transmission error, occurring on any of the active data lines, the card sends a negative CRC status on DAT[0]. In the case of successful transmission, over all active data lines, the card sends a positive CRC status on DAT[0] and starts the data programming procedure.

25.2.2.1 Protocol

There are two types of data transfer:

- Sequential operation
- Block-oriented operation

There are specific commands for each type of operation (sequential or block-oriented).

For information about commands and programming sequences supported by the MMC, SD, and SDIO cards, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, and *SDIO Card Specification (Part E1)*.

Figure 25-3 and Figure 25-4 show how sequential operations are defined. Sequential operation is only for 1-bit transfer and initiates a continuous data stream. The transfer terminates when a stop command follows on the mmci_cmd line.

Note

Stream commands are supported only by MMC cards.

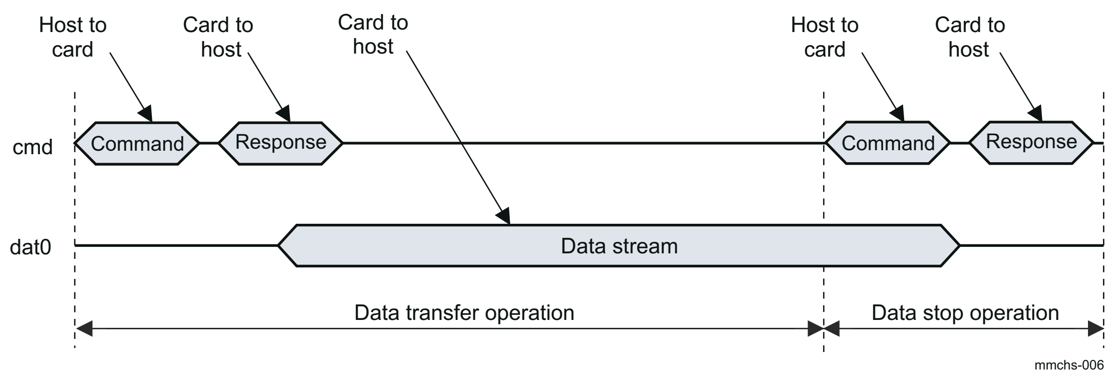


Figure 25-3. Sequential Read Operation (MMC Cards Only)

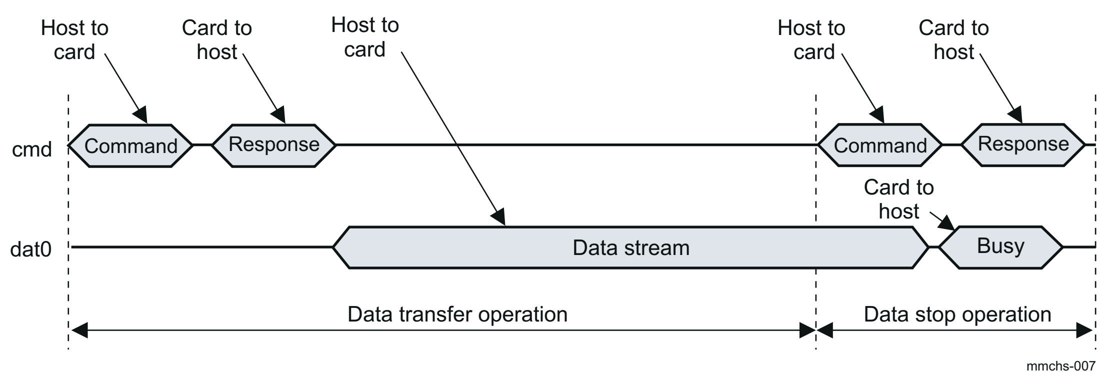


Figure 25-4. Sequential Write Operation (MMC Cards Only)

Figure 25-5 and Figure 25-6 show how multiple block-oriented operations are defined. A multiple block-oriented operation sends a data block plus CRC bits. The transfer terminates when a stop command follows on the mmci_cmd line. These operations are available for all kinds of cards.

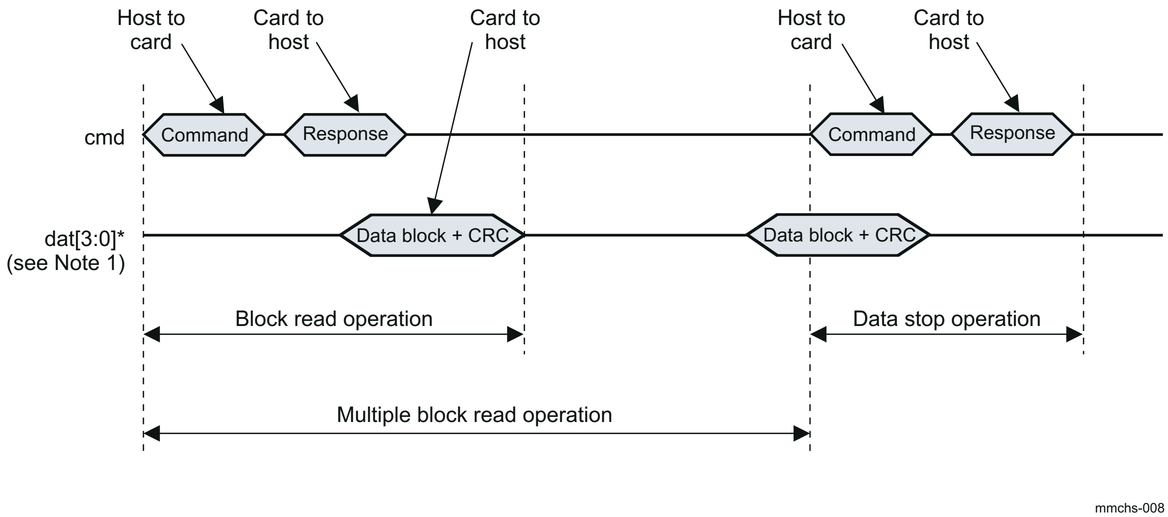


Figure 25-5. Multiple Block Read Operation

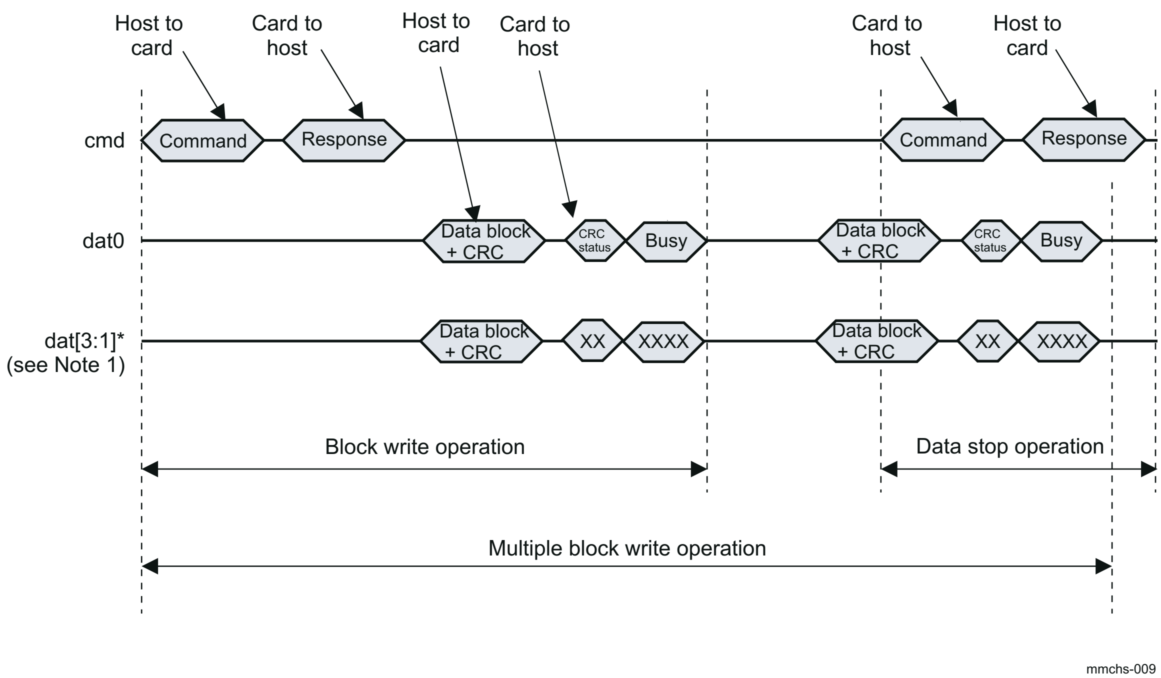


Figure 25-6. Multiple Block Write Operation With Card Busy Signal

Note

- The card busy signal is not always generated by the card; refer to Figure 25-5 and Figure 25-6, that show a particular case.
- Software must perform a software reset (set the MMCi.MMCHS_SYSCTL[26] SRD bit to 0x1) after a data time-out to ensure that CLK is stopped.
- For multiblock transfer, and especially for MMC cards, a transfer can be aborted without using a stop command. If a CMD23 is used before data transfer to define the number of blocks that will be transferred, then the transfer stops automatically after the last block (if the MMC card supports this feature).

25.2.2.2 Data Format

Coding Scheme for Command Token

Command tokens always start with 0 and end with 1. The second bit is a transmitter bit: 1 for a host command. The content is the command index (coded by 6 bits) and an argument (for example, an address), coded by 32 bits. The content is protected by 7-bit CRC checksum (see Figure 25-7).

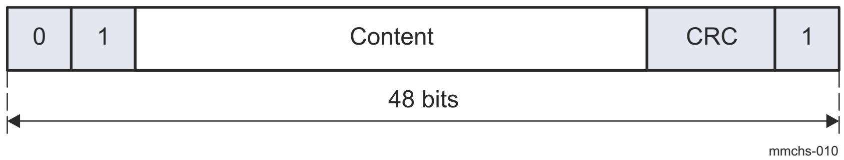


Figure 25-7. Command Token Format

Coding Scheme for Response Token

Response tokens always start with 0 and end with 1. The second bit is a transmitter bit: 0 for a card response. The content is different for each type of response (R1, R2, R3, R4, and R5, R6, R7 [for SD]) and the content is protected by 7-bit CRC checksum (see Figure 25-8 and Figure 25-9). Depending on the type of commands sent to the card, the MMCHS_CMD register must be configured differently to avoid false CRC or index errors to be flagged on command response (see Table 25-4). For more information about response types, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, and *SDIO Card Specification*.

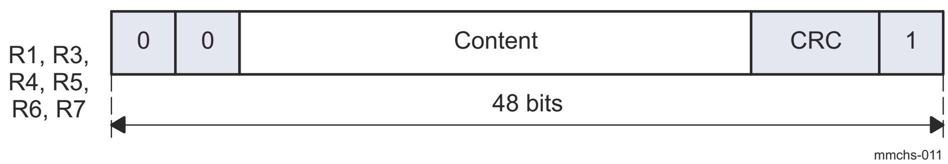


Figure 25-8. Response Token Format (R1, R3, R4, R5, R6, R7)

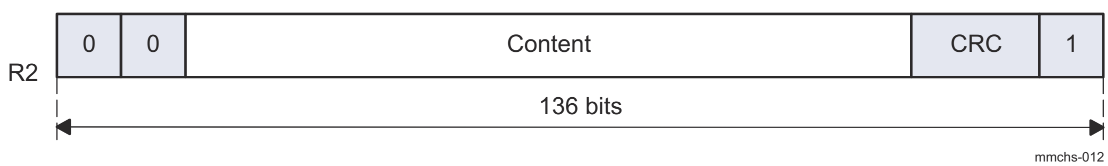


Figure 25-9. Response Token Format (R2)

Table 25-4. Relationship Between Configuration and Name of Response Type

Response Type MMCi.MMCHS_CMD[17:16] RSP_TYPE	Index Check Enable MMCi.MMCHS_CMD[20] CICE	CRC Check Enable MMCi.MMCHS_CMD[19] CCCE	Name of Response Type
00	0	0	No response
01	0	1	R2
10	0	0	R3 (R4 for SD cards)
10	1	1	R1, R6, R5, (R7 for SD cards)
11	1	1	R1b, R5b

Coding Scheme for Data Token

Data tokens always start with 0 and end with 1 (see Figure 25-10 through Figure 25-12).

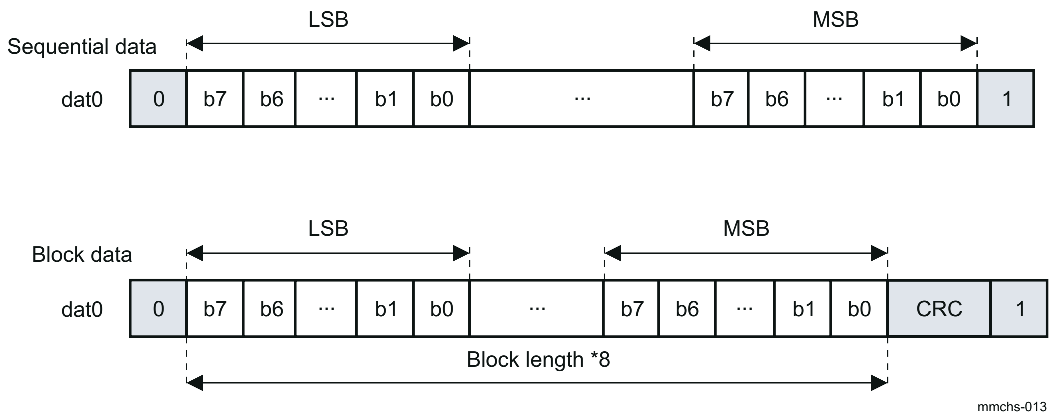


Figure 25-10. Data Token Format for 1-Bit Transfers

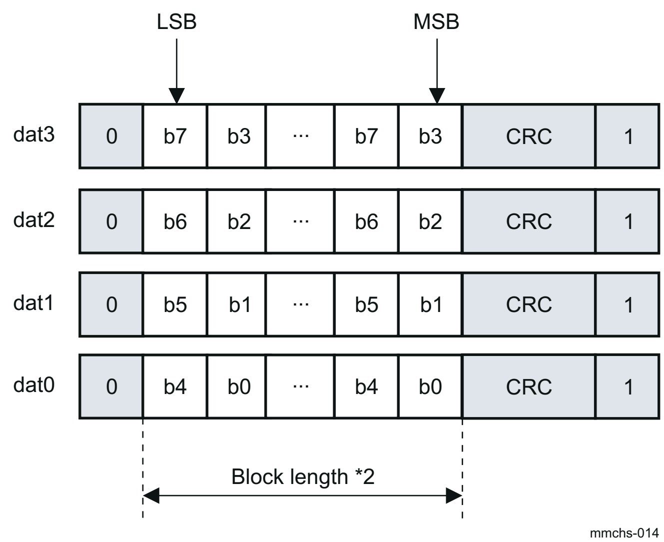


Figure 25-11. Data Token Format for 4-Bit Transfers

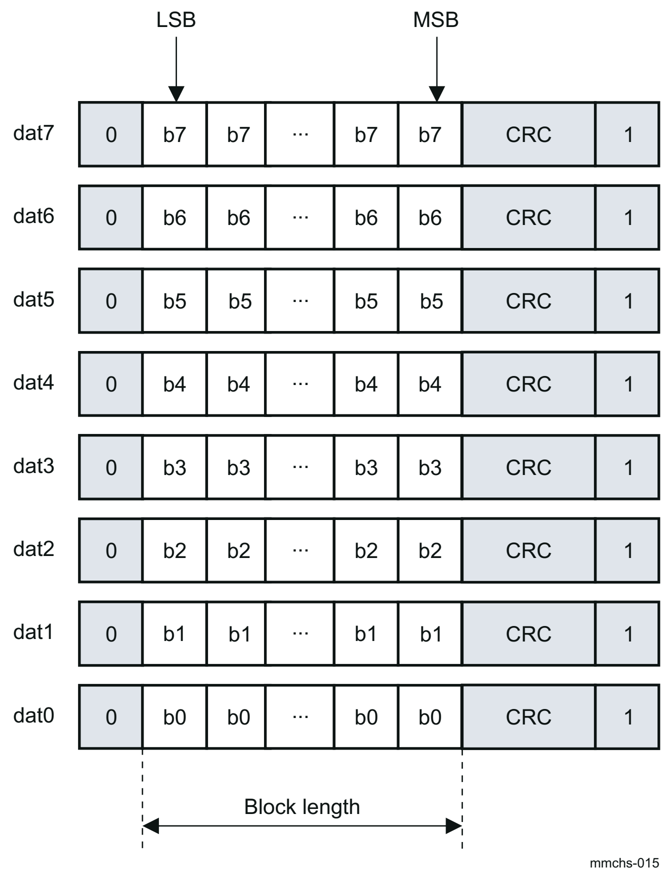


Figure 25-12. Data Token Format for 8-Bit Transfers

25.3 eMMC/SD/SDIO Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 25-13 shows the integration of the MMC1 and MMC2 controllers which are connected to both L3_MAIN and L4_PER1 interconnects and are able to act as master or slave. In master mode the L3_MAIN interconnect is used. In slave mode the L4_PER1 interconnect is used.

Figure 25-14 shows the integration of the MMC3 and MMC4 controllers which are connected to the L4_PER1 interconnect and act as a slave only.

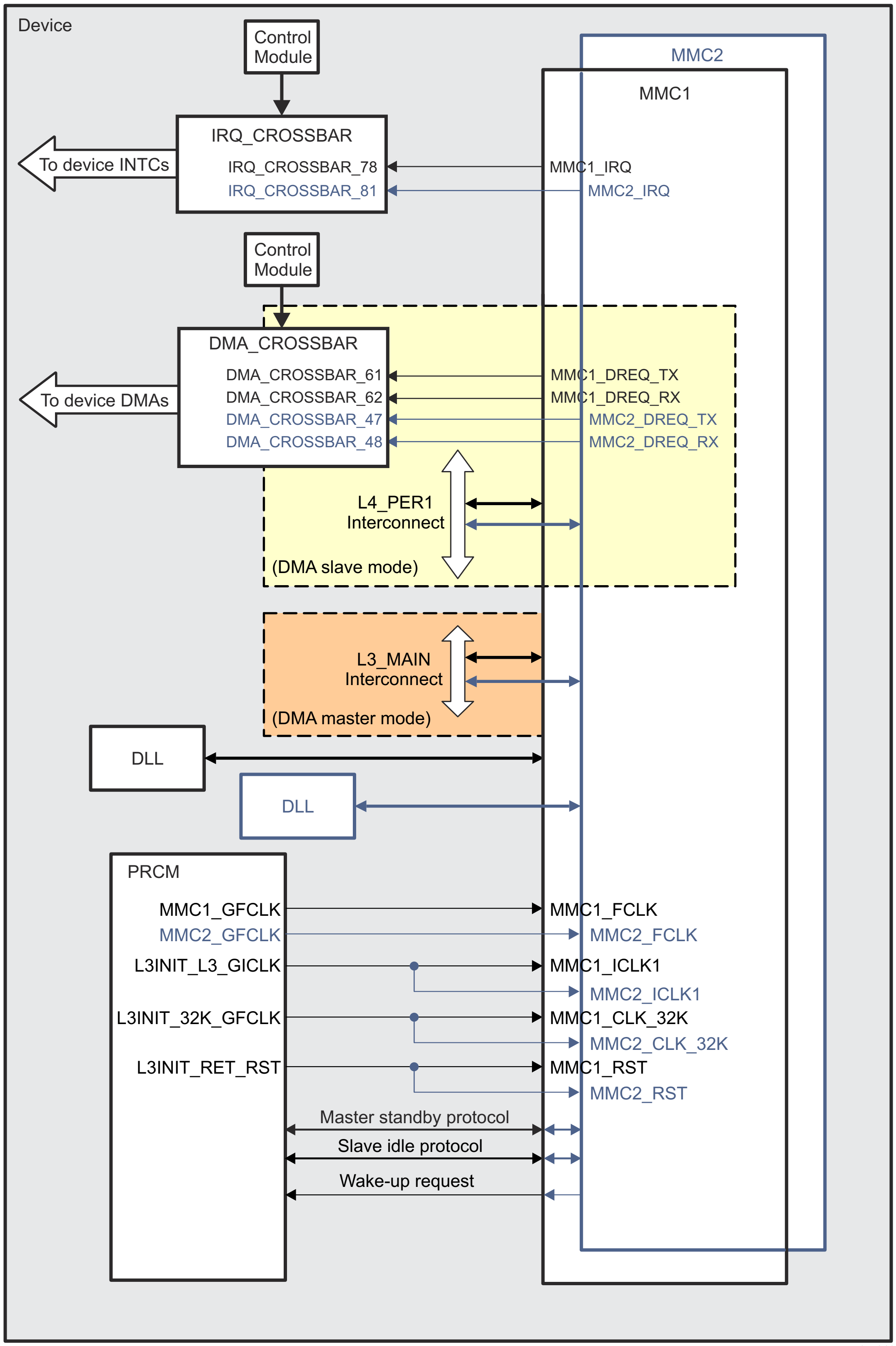


Figure 25-13. Integration of MMC1 and MMC2 Controllers – Master and Slave Capable

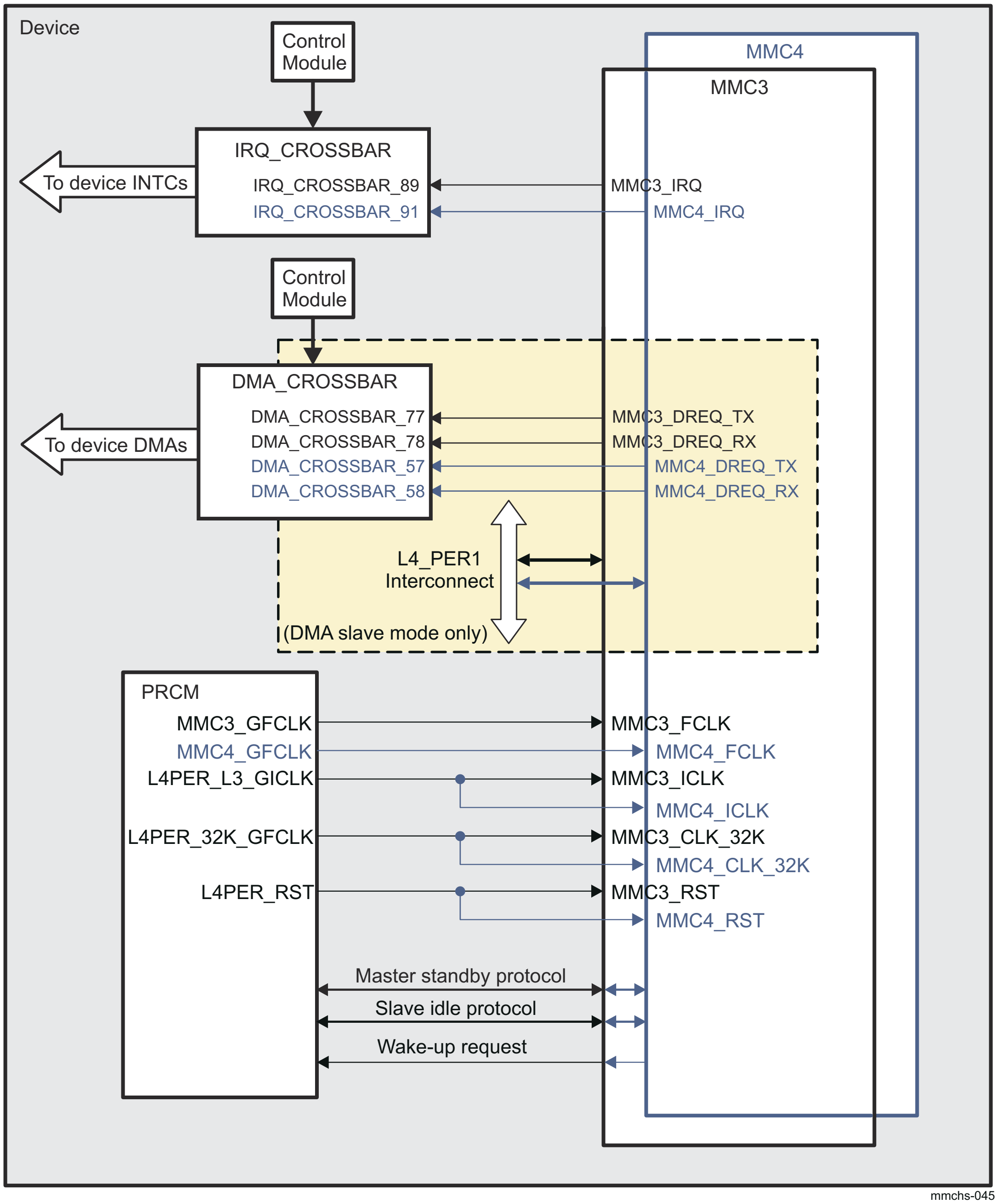


Figure 25-14. Integration of MMC3 and MMC4 Controllers – Slave Capable Only

Note

For more information about the slave idle protocol and the wake-up request, see *Device Power-Management Architecture Building Blocks*, in *Power, Reset, and Clock Management*.

Table 25-5 through Table 25-7 summarize the integration of the module in the device.

Table 25-5. MMC Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
MMC1	PD_COREAON	L3_MAIN L4_PER1

Table 25-5. MMC Integration Attributes (continued)

MMC2	PD_COREAON	L3_MAIN L4_PER1
MMC3	PD_COREAON	L4_PER1
MMC4	PD_COREAON	L4_PER1

Table 25-6. MMC Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MMC1	MMC1_FCLK	MMC1_GFCLK	PRCM	MMC1 function clock
	MMC1_ICLK1	L3INIT_L3_GICLK	PRCM	MMC1 interface clock
	MMC1_CLK_32K	L3INIT_32K_GFCLK	PRCM	MMC1 debounce clock
MMC2	MMC2_FCLK	MMC2_GFCLK	PRCM	MMC2 function clock
	MMC2_ICLK1	L3INIT_L3_GICLK	PRCM	MMC2 interface clock
	MMC2_CLK_32K	L3INIT_32K_GFCLK	PRCM	MMC2 debounce clock
MMC3	MMC3_ICLK	L4PER_L3_GICLK	PRCM	MMC3 interface clock
	MMC3_FCLK	MMC3_GFCLK	PRCM	MMC3 function clock
	MMC3_CLK_32K	L4PER_32K_GFCLK	PRCM	MMC3 debounce clock
MMC4	MMC4_ICLK	L4PER_L3_GICLK	PRCM	MMC4 interface clock
	MMC4_FCLK	MMC4_GFCLK	PRCM	MMC4 function clock
	MMC4_CLK_32K	L4PER_32K_GFCLK	PRCM	MMC4 debounce clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MMC1	MMC1_RST	L3INIT_RET_RST	PRCM	L3 reset to MMC1
MMC2	MMC2_RST	L3INIT_RET_RST	PRCM	L3 reset to MMC2
MMC3	MMC3_RST	L4PER_RST	PRCM	Reset to MMC3
MMC4	MMC4_RST	L4PER_RST	PRCM	Reset to MMC4

Table 25-7. MMC Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
MMC1	MMC1_IRQ	IRQ_CROSSBAR_78	MPU_IRQ_83	MMC1 interrupt request
			IPU1_IRQ_66	
			IPU2_IRQ_66	
MMC2	MMC2_IRQ	IRQ_CROSSBAR_81	MPU_IRQ_86	MMC2 interrupt request
			IPU1_IRQ_67	
			IPU2_IRQ_67	
MMC3	MMC3_IRQ	IRQ_CROSSBAR_89	MPU_IRQ_94	MMC3 interrupt request
			IPU1_IRQ_68	
			IPU2_IRQ_68	
MMC4	MMC4_IRQ	IRQ_CROSSBAR_91	MPU_IRQ_96	MMC4 interrupt request
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
MMC1	MMC1_DREQ_TX	DMA_CROSSBAR_61	DMA_SYSTEM_DREQ_60	MMC1 DMA TX
			DMA_EDMA_DREQ_60	
	MMC1_DREQ_RX	DMA_CROSSBAR_62	DMA_SYSTEM_DREQ_61	MMC1 DMA RX
			DMA_EDMA_DREQ_61	
MMC2	MMC2_DREQ_TX	DMA_CROSSBAR_47	DMA_SYSTEM_DREQ_46	MMC2 DMA TX
			DMA_EDMA_DREQ_46	
	MMC2_DREQ_RX	DMA_CROSSBAR_48	DMA_SYSTEM_DREQ_47	MMC2 DMA RX
			DMA_EDMA_DREQ_47	
MMC3	MMC3_DREQ_TX	DMA_CROSSBAR_77	DMA_SYSTEM_DREQ_76	MMC3 DMA TX
	MMC3_DREQ_RX	DMA_CROSSBAR_78	DMA_SYSTEM_DREQ_77	MMC3 DMA RX
MMC4	MMC4_DREQ_TX	DMA_CROSSBAR_57	DMA_SYSTEM_DREQ_56	MMC4 DMA TX
			DMA_EDMA_DREQ_56	
	MMC4_DREQ_RX	DMA_CROSSBAR_58	DMA_SYSTEM_DREQ_57	MMC4 DMA RX
			DMA_EDMA_DREQ_57	

Note

The “**Default Mapping**” column in Table 25-7 *MMC Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.
For more information about the device DMA_SYSTEM module, see *System DMA*.
For more information about the device EDMA module, see *Enhanced DMA*.

Note

- For a description of the interrupt source, see *Interrupt Requests*.
- For a description of the DMA source, see *DMA Modes*.

25.4 eMMC/SD/SDIO Functional Description

25.4.1 Block Diagram

Figure 25-15 is a block diagram of the eMMC/SD/SDIOi host controller.

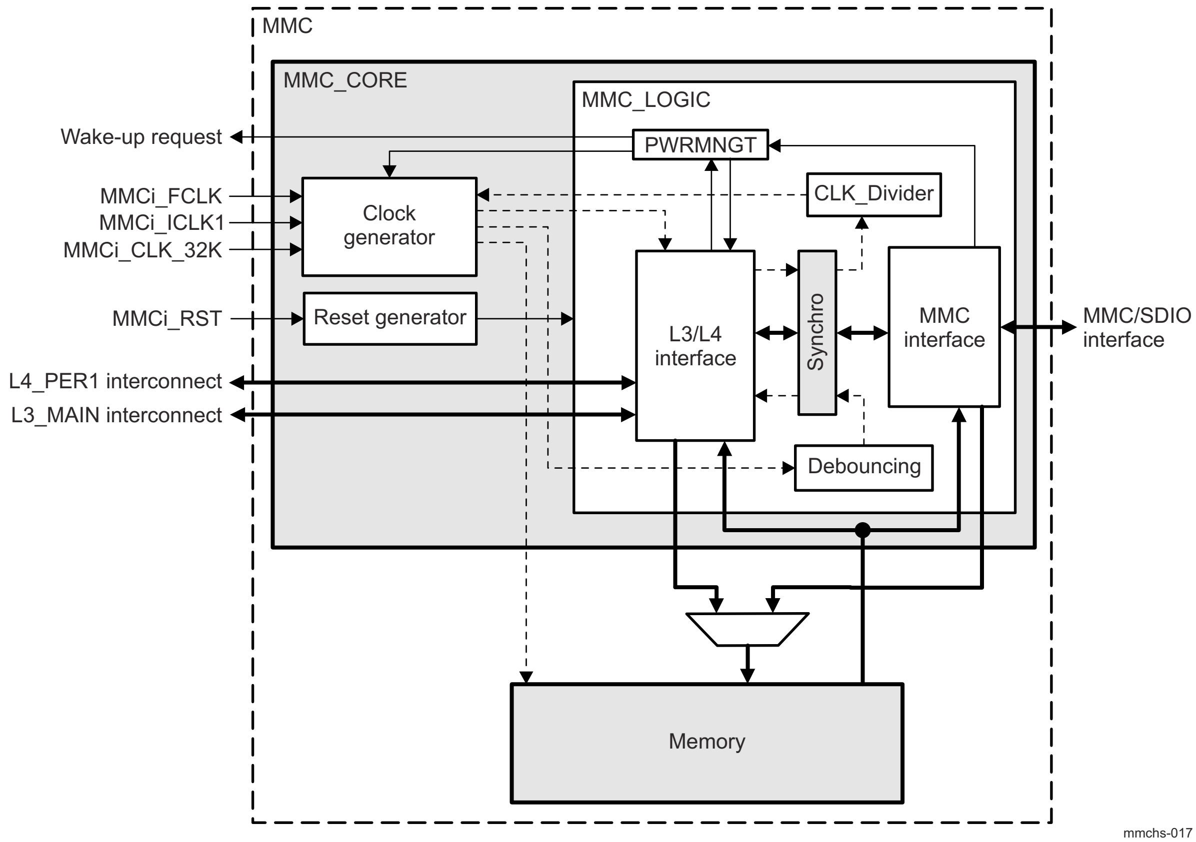


Figure 25-15. eMMC/SD/SDIO Diagram

25.4.2 Resets

25.4.2.1 Hardware Reset

The module is reinitialized by the hardware (see Table 25-6 for more information about reset signals).

The MMCi.MMCHS_SYSSTATUS[0] RESETDONE bit can be monitored by software to check whether the module is ready to use after a hardware reset.

Note

The functional clock (MMCi_FCLK) and interface clock (MMCi_ICLK) must be provided to the module to allow the RESETDONE status bit to be set.

The debounce clock (MMCi_32K) must be active to reset the module correctly.

This hardware reset signal has a global reset action on the module. All configuration registers and all state-machines are reset in all clock domains.

25.4.2.2 Software Reset

The module is reinitialized by software through the MMCi.MMCHS_SYSCONFIG[1] SOFTRESET bit. This bit has the same effect on the module logic as the hardware signal (MMCi_RST), with the following exceptions:

- Debounce logic

- MMCi.MMCHS_PSTATE, MMCi.MMCHS_CAPA, and MMCi.MMCHS_CUR_CAPA registers (see the corresponding register description)

The SOFTRESET bit is active high. The bit is automatically reinitialized to 0 by hardware. The MMCi.MMCHS_SYSCTL[24] SRA bit has the same action on the design as the SOFTRESET bit.

The MMCi.MMCHS_SYSSTATUS[0] RESETDONE bit can be monitored by software to check whether the module is ready to use after a software reset.

Moreover, two partial software reset bits are provided:

- MMCi.MMCHS_SYSCTL[26] SRD
- MMCi.MMCHS_SYSCTL[25] SRC

These 2 reset bits are useful to reinitialize data or command processes, respectively, in case of line conflict. When these bits are set to 1, a reset process is automatically released when the reset completes:

- The MMCi.MMCHS_SYSCTL[26] SRD bit resets all finite state-machines (FSMs) and status management that handle data transfers on the interface and functional sides.
- The MMCi.MMCHS_SYSCTL[25] SRC bit resets all FSMs and status management that handle command transfers on the interface and functional sides.

25.4.3 Power Management

The eMMC/SD/SDIO host controller can enter into different modes and save power:

- Normal mode
- Idle mode

The two modes are mutually exclusive (the module can be in normal mode or in idle mode). The eMMC/SD/SDIO host controller is compliant with the handshake protocol of the power, reset, and clock management (PRCM) module.

Normal Mode

The autogating of interface and functional clocks occurs when the following conditions are met:

- The MMCi.MMCHS_SYSCONFIG[0] AUTOIDLE bit is set to 1.
- There is no transaction on the MMC interface.

The autogating of interface and functional clocks stops when the following conditions are met:

- A register access occurs through the L4 interconnect.
- A wake-up event occurs (card insertion, card removal, an interrupt from a SDIO card).
- A transaction on the MMC/SD/SDIO interface starts.

When a card removal event is detected the eMMC/SD/SDIO host controller automatically clears MMCHS_HCTL[8] SDBP and MMCHS_SYSCTL[2] CEN bits. Then it enters into low power state with auto gated interface clock even if MMCHS_SYSCONFIG[0] AUTOIDLE bit is set to 0. The functional clock is internally switched off and only interconnect read and write accesses are allowed.

Idle Mode

The MMCi_ICLK and MMCi_FCLK clocks provided to the eMMC/SD/SDIO host controller are switched off upon a PRCM module request. They are switched back upon module request.

The eMMC/SD/SDIO host controller complies with the handshaking protocol of the PRCM module:

- IDLE request from the system power manager
- Idle acknowledgment from the eMMC/SD/SDIO host controller
- Wake-up request from the eMMC/SD/SDIO host controller

The idle acknowledgment varies according to the MMCi.MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field:

- 0x0: Force-idle mode. The eMMC/SD/SDIO host controller acknowledges the system power manager request unconditionally.

- 0x1: No-idle mode. The eMMC/SD/SDIO host controller ignores the system power manager request and behaves normally as if the request was not asserted.
- 0x2: Smart-idle mode. The eMMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state.
- 0x3: Smart-idle wake-up-capable mode. The eMMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state. However, the module may generate wake-up events when it is in IDLE state (related to IRQ or DMA requests)

During the smart-idle mode period, the eMMC/SD/SDIO host controller acknowledges that the MMCi_ICLK and MMCi_FCLK clocks may be switched off, regardless of the value set in the MMCi.MMCHS_SYSCONFIG[9:8] CLOCKACTIVITY bit field.

The debounce clock is used to debounce the signals related to the card insertion and the card removal that are also sources of wake-up in idle mode. The debounce clock must never be switched off by the system power manager in order to detect card removal in functional mode, and detect wake-up in idle mode.

Transition From Normal Mode to Smart-Idle Mode

Smart-idle mode is enabled when the MMCi.MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x2 or 0x3.

The eMMC/SD/SDIOi host controller goes into idle mode when the PRCM issues an IDLE request, according to its internal activity.

The eMMC/SD/SDIO host controller acknowledges the IDLE request from the PRCM after ensuring the following:

- The current multi- or single-block transfer is complete.
- Any interrupt or DMA request is asserted.
- There is no card interrupt on the DATA[1] signal.

As long as the eMMC/SD/SDIOi controller does not acknowledge the IDLE request, if an event occurs, the eMMC/SD/SDIOi host controller can still generate an interrupt or a DMA request. In this case, the module ignores the IDLE request from the PRCM module.

As soon as the eMMC/SD/SDIOi controller acknowledges the IDLE request from the PRCM module:

- If smart-idle mode: The module does not assert any new interrupt or DMA request.
- If smart-idle wake-up-capable mode: The module may generate wake-up events related to an interrupt or DMA request.

Wake-Up Event in Smart-Idle Mode

The wake-up feature is enabled when both the MMCi.MMCHS_SYSCONFIG[2] ENAWAKEUP bit and one of the three bits MMCi.MMCHS_HCTL[26] REM, MMCi.MMCHS_HCTL[25] INS, MMCi.MMCHS_HCTL[24] IWE are set to 0x1.

The corresponding interrupt status enable bits must also be set before going in idle mode. Setting one of the following bits:

- MMCi.MMCHS_IE[8] CIRQ_ENABLE
- MMCi.MMCHS_IE[7] CREM_ENABLE
- MMCi.MMCHS_IE[6] CINS_ENABLE

to 0x1 enables the wakeup event detection. The source of wakeup can be identified after idle mode exiting by reading one of the corresponding MMCHS_STAT bits.

The wakeup is generated only in smart-idle mode, when the module is in idle mode.

Table 25-8 lists the supported cases in smart-idle mode.

Table 25-8. Smart-Idle Mode and Wake-Up Capabilities

Mode	MMCi_ICLK Clock	MMCi_FCLK Clock	Wake-Up Event
Card interrupt	May be switched off[(1)]	May be switched off[(1)]	The module sends an asynchronous wake-up request when a card interrupt on the DATA[1] signal is detected.
Card insertion	May be switched off[(1)]	May be switched off[(1)]	The module sends an asynchronous wake-up request when card insertion is detected.
Card removal	May be switched off[(1)]	May be switched off[(1)]	The module sends an asynchronous wake-up request when card removal is detected.

(1) The eMMC/SD/SDIOi host controller assumes that both clocks may be switched off, regardless of the value set in the MMCi.MMCHS_SYSCONFIG[9:8] CLOCKACTIVITY bit field.

Transition From Smart-Idle Mode to Normal Mode

The eMMC/SD/SDIO host controller detects the end of the idle period when the PRCM module deasserts the IDLE request.

For the wake-up event, there is a corresponding interrupt status in the MMCi.MMCHS_STAT register. The eMMC/SD/SDIOi host controller operates the conversion between the wake-up and interrupt (or DMA request) upon exit from smart-idle mode, if the associated enable bit is set in the MMCi.MMCHS_ISE register.

Interrupts and wake-up events have independent enable and disable controls, accessible through the MMCi.MMCHS_HCTL and MMCi.MMCHS_ISE registers. The overall consistency must be ensured by software.

One of the bits MMCHS_STAT[8] CIRQ, MMCHS_STAT[7] CREM, MMCHS_STAT[6] CINS in the interrupt status register is updated with the event that caused the wake-up when one of the bits MMCHS_IE[8] CIRQ_ENABLE, MMCHS_IE[7] CREM_ENABLE, MMCHS_IE[6] CINS_ENABLE is enabled.

Then, the wake-up event at the origin of the transition from smart-idle mode to normal mode is converted into its corresponding interrupt or DMA request. (The MMCi.MMCHS_STAT register is updated and the status of the interrupt signal changes.)

When the IDLE request from the PRCM module is deasserted, the module switches back to normal mode. The module is fully operational.

Force-Idle Mode

Force-idle mode is enabled when the MMCi.MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0.

Force-idle mode is an idle mode in which the eMMC/SD/SDIOi host controller responds unconditionally to the IDLE request from the PRCM module. Moreover, in this mode, the eMMC/SD/SDIOi host controller unconditionally deasserts interrupts and DMA request lines if they are asserted.

The transition from normal mode to force-idle mode does not affect the bits of the MMCi.MMCHS_STAT register.

In force-idle mode, the interrupt and DMA request lines are deasserted. MMCi_ICLK and MMCi_FCLK can be switched off.

> **CAUTION**
>
> In force-idle mode, an IDLE request from the PRCM module during a command or a data transfer can lead to an unexpected and unpredictable result.
> When the module is idle, any access to the module generates an error as long as the MMCi_ICLK clock is alive.

The module exits force-idle mode when the PRCM module deasserts the IDLE request. Then the module switches back to normal mode. The module is fully operational. Interrupt and DMA request lines are optionally asserted one clock cycle later.

Standby Mode

The eMMC/SD/SDIO host controller also provides standby information to the system power manager if the generic parameter MMCHS_HL_HWINFO[0] MADMA_EN is set to 1.

The eMMC/SD/SDIO host controller complies with the handshaking protocol of the PRCM module:

- Standby request from the eMMC/SD/SDIO host controller
- Wait acknowledgement from the PRCM module

The standby request varies according to the MMCi.MMCHS_SYSCONFIG[13:12] STANDBYMODE bit field:

- 0x0: Force-standby. The eMMC/SD/SDIO host controller sends the standby request to the system power manager unconditionally.
- 0x1: No-standby. The eMMC/SD/SDIO host controller does not generate any standby request to the system power manager and behaves normally
- 0x2: Smart-standby. The eMMC/SD/SDIO host controller sends the standby request to the system power manager according to the master L3 interface state.

Power Pad Control

The eMMC/SD/SDIO host controller has the ability to reduce the pad power leakage when no transfer is sent through the pad. According to the MMCHS_CON[15] PADEN there are two different pad power management modes: automatic and manual.

If MMCHS_CON[15] PADEN is set to 1, pads CLK, CMD, DAT[0] and DATA[i] (where i = 2 through 7) are always powerd on.

If MMCHS_CON[15] PADEN is set to 0, the power for pads CLK, CMD, DAT[0] and DATA[i] (where i = 2 through 7) is "ON" only when there is a transfer on going. This is automatically managed by an internal state machine of the eMMC/SD/SDIO host controller.

The DATA[1] pad active state is controlled through MMCHS_CON[11] CTPL in order to detect SDIO asynchronous interrupt when there is no transaction.

The delay between pad power "ON" and the command transmission is controlled through the MMCHS_PWCNT register which act as a programmable counter. It is also used to delay the pad power "OFF" after the end of transmission. By default this counter is reset which means that no additional delay is added. But there is approximately 6-7 clock cycles margin between pad power "ON" and real start of the command due to an internal state machine of the eMMC/SD/SDIO host controller.

Note

The MMCHS_PWCNT register is considered as static. No dynamic configuration during the transfer is supported. This results in an unpredictable behavior.

Local Power Management

Table 25-9 describes the power-management features available for the eMMC/SD/SDIOi modules.

Note

For information about source clock gating and a description of the sleep/wake-up transitions, see *Clock Management Functional Description*, in *Power, Reset, and Clock Management*.

Table 25-9. Local Power-Management Features

Feature	Registers	Description
Clock auto gating	MMCHS_SYSCONFIG[0] AUTOIDLE	This bit allows a local power optimization inside the module by gating the MMCi_ICLK clock upon the interface activity, or gating the MMCi_FCLK clock upon the internal activity.

Table 25-9. Local Power-Management Features (continued)

Feature	Registers	Description
Slave-idle modes	MMCHS_SYSCONFIG[3:4] SIDLEMODE	Force-idle, No-idle, Smart-idle and Smart-idle wake-up-capable modes are available.
Clock activity	MMCHS_SYSCONFIG[8:9] CLOCKACTIVITY	For configuration details, see Table 25-10.
Master standby modes	MMCHS_SYSCONFIG[12:13] STANDBYMODE	Force-standby, No-standby and Smart-standby modes are available.
Global wake-up enable	MMCHS_SYSCONFIG[2] ENAWAKEUP	This bit enables the wake-up feature at the module level.
Wake-up sources enable	MMCHS_HCTL register	This register holds one active-high enable bit per event source that is able to generate a wake-up signal.

Table 25-10. Clock Activity Settings

CLOCKACTIVITY Values	Clock State When Module is in IDLE State		Features Available When Module is in IDLE State	Wake-Up Events
	MMCi_ICLK	MMCi_ FCLK		
00	OFF	OFF	None	Card interrupt, Card insertion, Card removal
10	OFF	ON	None	
01	ON	OFF	None	
11	ON	ON	All	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the CLOCKACTIVITY and MMCi clock PRCM control bits. For a description of the Clock activity feature, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

25.4.4 Interrupt Requests

Several internal module events can generate an interrupt. Each interrupt has a status bit, an interrupt enable bit, and a signal status enable:

- The status of each type of interrupt is automatically updated in the MMCi.MMCHS_STAT register; it indicates which service is required.
- The interrupt status enable bits of the MMCi.MMCHS_IE register enable or disable the automatic update of the MMCi.MMCHS_STAT register on an event-by-event basis.
- The interrupt signal enable bits of the MMCi.MMCHS_ISE register enable or disable the transmission of an interrupt request on the interrupt line MMCi_IRQ on an event-by-event basis.

If an interrupt status is disabled in the MMCi.MMCHS_IE register, then the corresponding interrupt request is not transmitted, and the value of the corresponding interrupt signal enable in the MMCi.MMCHS_ISE register is ignored.

When an interrupt event occurs, the corresponding status bit is automatically set to 0x1 (the eMMC/SD/SDIOi host controller updates the status bit) in the MMCi.MMCHS_STAT register. If a mask is later applied on the interrupt in the MMCi.MMCHS_ISE register, the interrupt request is deactivated.

When the interrupt source has not been serviced, if the interrupt status is cleared in the MMCi.MMCHS_STAT register and the corresponding mask is removed from the MMCi.MMCHS_ISE register, the interrupt status is not asserted again in the MMCi.MMCHS_STAT register and the eMMC/SD/SDIOi host controller does not transmit an interrupt request.

Note

If the buffer write ready (BWR) interrupt or the buffer read ready (BRR) only interrupt are not serviced and are cleared in the MMCi.MMCHS_STAT register, and the corresponding mask is removed, then the eMMC/SD/SDIOi host controller waits for the service of the interrupt without updating the status MMCi.MMCHS_STAT register or transmitting an interrupt request.

Table 25-11 lists the event flags, and their mask, that can cause module interrupts.

Table 25-11. Events

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[29] BADA	MMCHS_IE[29] BADA_ENABLE	MMCi_IRQ	Bad access to data space. This bit is set automatically to indicate a bad access to buffer when not allowed: This bit is set during a read access to the data register (MMCHS_DATA) while buffer reads are not allowed (MMCHS_PSTATE[11] BRE = 0). This bit is set during a write access to the data register (MMCHS_DATA) while buffer writes are not allowed (MMCHS_PSTATE[10] BWE = 0).
MMCHS_STAT[28] CERR	MMCHS_IE[28] CERR_ENABLE	MMCi_IRQ	Card error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5, or R5b. Only bits referenced as type E (error) in the status field in the response can set a card status error. An error bit in the response is flagged only if the corresponding bit in the card status response error MMCHS_CSRE is set. There is no card error detection for the auto CMD12 command.
MMCHS_STAT[26] TE	MMCHS_IE[26] TE_ENABLE	MMCi_IRQ	Tuning Error. This bit is set when an unrecoverable error is detected in a tuning circuit except during tuning procedure. The Tuning Error is with higher priority than the other error interrupts generated during data transfer.
MMCHS_STAT[25] ADMAE	MMCHS_IE[25] ADMAE_ENABLE	MMCi_IRQ	ADMA error. This bit is set when the host controller detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA error status register. In addition, the host controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state.
MMCHS_STAT[24] ACE	MMCHS_IE[24] ACE_ENABLE	MMCi_IRQ	Auto CMD12 error and Auto CMD23 error. This bit is set automatically when one of the bits MMCHS_AC12[4:0] changes from 0 to 1.
MMCHS_STAT[22] DEB	MMCHS_IE[22] DEB_ENABLE	MMCi_IRQ	Data end bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on the DAT line or at the end position of the CRC status in write mode.
MMCHS_STAT[21] DCRC	MMCHS_IE[21] DCRC_ENABLE	MMCi_IRQ	Data CRC error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status difference of a position 010 token during a block write command.
MMCHS_STAT[20] DTO	MMCHS_IE[20] DTO_ENABLE	MMCi_IRQ	Data time-out error. This bit is set automatically according to the following conditions: • Busy time-out for R1b, R5b response type • Busy time-out after write CRC status • Write CRC status time-out • Read data time-out

Table 25-11. Events (continued)

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[19] CIE	MMCHS_IE[19] CIE_ENABLE	MMCi_IRQ	Command index error. This bit is set automatically when the response index differs from the corresponding command index previously emitted. The check is enabled through the MMCHS_CMD[20] CICE bit.
MMCHS_STAT[18] CEB	MMCHS_IE[18] CEB_ENABLE	MMCi_IRQ	Command end bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response.
MMCHS_STAT[17] CCRC	MMCHS_IE[17] CCRC_ENABLE	MMCi_IRQ	Command CRC error. This bit is set automatically when a CRC7 error occurs in the command response. CRC check is enabled through the MMCHS_CMD[19] CCCE bit.
MMCHS_STAT[16] CTO	MMCHS_IE[16] CTO_ENABLE	MMCi_IRQ	Command time-out error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within five clock cycles, the time-out is still detected at 64 clock cycles.
MMCHS_STAT[15] ERRI	MMCHS_IE[15] ERRI_ENABLE	MMCi_IRQ	Error interrupt. If any of the bits in the error interrupt status register (MMCHS_STAT[31:16]) are set, this bit is set to 1.
MMCHS_STAT[10] BSR	MMCHS_IE[10] BSR_ENABLE	MMCi_IRQ	Boot status received interrupt. This bit is set automatically when the MMCHS_CON[18] BOOT_CF0 bit is set to 0x1 or 0x2 and a boot status is received on the dat0 line. This interrupt is useful only for the MMC card.
MMCHS_STAT[8] CIRQ	MMCHS_IE[8] CIRQ_ENABLE	MMCi_IRQ	Card interrupt. This bit is used only for SD, SDIO, and CE-ATA cards. In 1-bit mode, the interrupt source is asynchronous (can be a source of asynchronous wakeup). In 4-bit mode, the interrupt source is sampled during the interrupt cycle. In CE-ATA mode, the interrupt source is detected when the card drives the CMD line to 0 during one cycle after data transmission end.
MMCHS_STAT[7] CREM	MMCHS_IE[7] CREM_ENABLE	MMCi_IRQ	Card removal. This bit is set automatically when MMCHS_PSTATE[16] CINS changes from 1 to 0.
MMCHS_STAT[6] CINS	MMCHS_IE[6] CINS_ENABLE	MMCi_IRQ	Card insertion. This bit is set automatically when MMCHS_PSTATE[16] CINS changes from 0 to 1.
MMCHS_STAT[5] BRR	MMCHS_IE[5] BRR_ENABLE	MMCi_IRQ	Buffer read ready. This bit is set automatically during a read operation to the card (see class 2 block-oriented read commands) when one block specified by the MMCHS_BLK[10:0] BLEN bit field is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the LH must empty the buffer by reading it.
MMCHS_STAT[4] BWR	MMCHS_IE[4] BWR_ENABLE	MMCi_IRQ	Buffer write ready. This bit is set automatically during a write operation to the card (see class 4 block-oriented write command) when the host can write a complete block as specified by the MMCHS_BLK[10:0] BLEN bit field. It indicates that the memory card has emptied one block from the buffer and that the LH can write one block of data into the buffer.
MMCHS_STAT[3] DMA	MMCHS_IE[3] DMA_ENABLE	MMCi_IRQ	DMA interrupt. This status is set when an interrupt is required in the ADMA instruction and after the data transfer completes.
MMCHS_STAT[2] BGE	MMCHS_IE[2] BGE_ENABLE	MMCi_IRQ	Block gap event. When a stop at the block gap is requested (MMCHS_HCTL[16] SBGR), this bit is automatically set when transaction is stopped at the block gap during a read or write operation.

Table 25-11. Events (continued)

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[1] TC	MMCHS_IE[1] TC_ENABLE	MMCi_IRQ	Transfer completed. This bit is always set when a read/write transfer is complete or between two blocks when the transfer is stopped because of a stop at block gap request (MMCHS_HCTL[16] SBGR). • In read mode: This bit is automatically set when a read transfer completes (MMCHS_PSTATE[9] RTA). • In write mode: This bit is automatically set when the DAT line use completes (MMCHS_PSTATE[2] DLA).
MMCHS_STAT[0] CC	MMCHS_IE[0] CC_ENABLE	MMCi_IRQ	Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (MMCHS_PSTATE[0] CMDI). If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command time-out error (MMCHS_STAT[16] CTO) has higher priority than command complete (MMCHS_STAT[0] CC). If a response is expected but none is received, then a command time-out error is detected and signaled, instead of the command complete interrupt.

Note

To send an interrupt request to the MMCi_IRQ line, the mask/unmask bit must be set in the MMCi.MMCHS_IE and MMCi.MMCHS_ISE registers.

The eMMC/SD/SDIOi host controller supports interrupt-driven operation and polling.

25.4.4.1 Interrupt-Driven Operation

An interrupt-enable bit must be set in the MMCi.MMCHS_IE register to enable the module internal source of interrupt.

When an interrupt event occurs, the single interrupt line is asserted and the LH must:

1. Read the MMCi.MMCHS_STAT register to identify which event occurred.
2. Write 1 into the corresponding bit of the MMCi.MMCHS_STAT register to clear the interrupt status and release the interrupt line (if a read is done after this write, this returns 0).

Note

In the MMCi.MMCHS_STAT register, the card interrupt (CIRQ) and error interrupt (ERRI) bits cannot be cleared.

The MMCi.MMCHS_STAT[8] CIRQ status bit must be masked by disabling the MMCi.MMCHS_IE[8] CIRQ_ENABLE bit (set to 0x0), and then the interrupt routine must clear the SDIO interrupt source in the SDIO card common control register (CCCR).

The MMCi.MMCHS_STAT[15] ERRI bit is automatically cleared when all status bits in the MMCi.MMCHS_STATregister (bits 31 through 16) are cleared.

25.4.4.2 Polling

When the interrupt capability of an event is disabled in the MMCi.MMCHS_ISE register, the interrupt line is not asserted:

- Software can poll the status bit in the MMCi.MMCHS_STAT register to detect when the corresponding event occurs.
- Writing 1 into the corresponding bit of the MMCi.MMCHS_STAT register clears the interrupt status and does not affect the interrupt line state.

Note

See the previous note concerning clearing of the CIRQ and ERRI bits.

25.4.4.3 Asynchronous Interrupt

Asynchronous interrupt is defined in *SDIO Card Specification version 3.00, part E*. This interrupt is effective in 4-bit mode and is generated without SD clock. Asynchronous interrupt period is defined in the synchronous interrupt period after the last data block and until a next command is received.

Asynchronous interrupt period in a multiple block write operation.

If MMCHS_CAPA[29] AIS is set to 0, writing to MMCHS_AC12[30] AI_ENABLE is ignored. This bit is set to 0. A synchronous interrupt period starts two clocks after the last data block. If MMCHS_AC12[30] AI_ENABLE is set to 1, the asynchronous interrupt period starts four clocks after the start of the synchronous interrupt period. Four clocks after the start bit of the next command, the asynchronous interrupt period ends and goes back to the synchronous interrupt period.

25.4.5 DMA Modes

Two DMA management modes can be used to load data from memory to the internal buffer of the controller (or vice versa). These modes are exclusive and depend on the module integration.

- DMA master mode:

  DMA master mode is selected by setting the MMCHS_CON[20] DMA_MNS bit to 1. In this case, the controller has direct access to data using a specific algorithm called ADMA2 (prevents the system from being interrupted). Data are exchanged using the L3_MAIN master interface, which supports burst accesses to maximize throughput.

  Note

  This mode is supported only by modules connected to the L3_MAIN interconnect. For more information and/or to check the value of the MMCHS_HL_HWINFO[0] MADMA_EN bit, see Section 25.1, *eMMC/SD/SDIO Overview*.

  This mode is available for modules MMC1 and MMC2.

- DMA slave mode:

  DMA slave mode is selected by setting the MMCHS_CON[20] DMA_MNS bit to 0. In this case, the controller is slave on the DMA transaction managed by two separated requests (MMCi_DMA_TX and MMCi_DMA_RX).

  Note

  This mode is the only mode supported by modules that are not connected to the L3_MAIN interconnect (regardless of the value of the MMCHS_CON[20] DMA_MNS bit). For more information and/or to check the value of the MMCHS_HL_HWINFO[0] MADMA_EN bit, see Section 25.1, *eMMC/SD/SDIO Overview*.

  This mode is available for all MMC modules.

25.4.5.1 Master DMA Operations

The eMMC/SD/SDIOi host controller has direct access to the internal data. This feature is called advanced DMA (ADMA). It follows a specific algorithm (ADMA2) defined by an instruction in memory that starts at an address previously loaded in the MMCHS_ADMASAL register before any data command issued to the MMC card. Only 32-bit address spacing is supported by the controller for data storage.

Note

This mode is supported only by modules connected to the L3_MAIN interconnect. For more information and/or to check the value of the MMCHS_HL_HWINFO[0] MADMA_EN bit, see Section 25.1, *eMMC/SD/SDIO Overview*.

These instructions must be loaded by software in a 32-bit-addressed descriptor table in system memory, as shown in Figure 25-16. In this case the MMCHS_ADMASAL register is used as the program address pointer

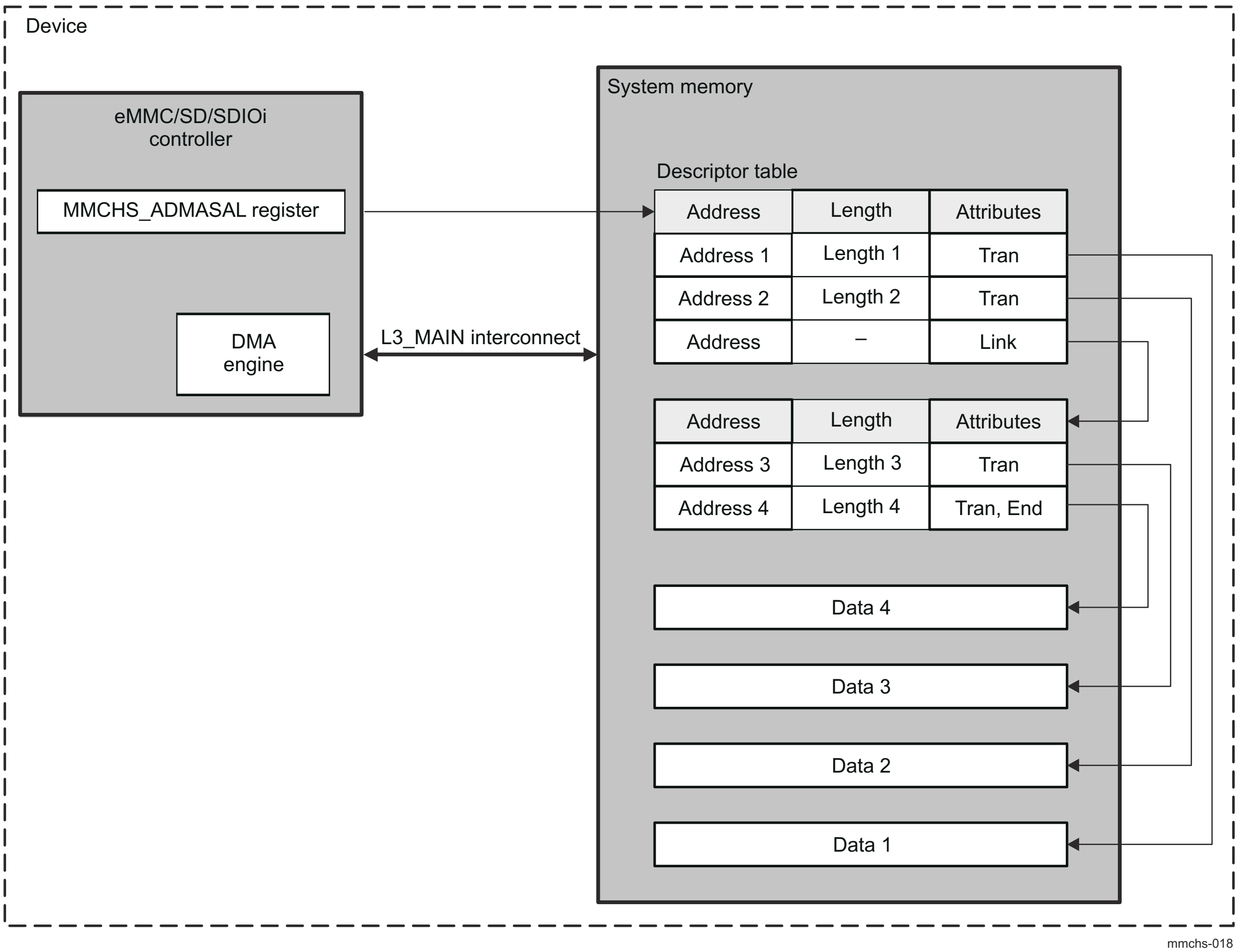


Figure 25-16. ADMA Block Diagram Overview

25.4.5.1.1 Descriptor Table Description

Each descriptor line contains an address, a length, and attributes fields. The attributes define which operation will be processed. Every descriptor line is a 64-bit-wide register that is fetched in the controller using the L3_MAIN master interface, and requires two 32-bit accesses to memory.

Table 25-12 shows the structure of a descriptor line.

Table 25-12. Descriptor Line Overview

Address Field		Length		Reserved		Attributes					
63	32	31	16	15	6	5	4	3	2	1	0
32-bit address		16-bit length		0x0		Act2	Act1	0	Int	Ent	Valid

The attribute of the descriptor line is divided into two parts:

- Attributes[5:4]: The action to be processed by the ADMA engine
- Attributes[3:0]: Additional parameters characterizing the behavior of the ADMA engine

Table 25-13 describes the available actions of a descriptor line.

Table 25-13. Available Actions of a Descriptor Line

Act2	Act1	Symbol	Comment	Operation
0	0	Nop	No operation	Do not execute the current line and go to the next line.
0	1	Rsv	Reserved	Reserved action. Behaves the same as the Nop command.
1	0	Tran	Transfer data	Transfer data of one descriptor line.
1	1	Link	Link descriptor	Link to another descriptor.

Table 25-14 describes the additional parameters of a descriptor line.

Table 25-14. Additional Parameters of a Descriptor Line

Bit	Description
Valid	Valid = 1 indicates that this descriptor line is effective. If Valid = 0, an ADMA error interrupt is generated and the ADMA is stopped. This prevents runaways.
End	End = 1 indicates the end of a descriptor. The transfer-complete interrupt is generated when the operation of the descriptor line is complete.
Int	Int = 1 generates a DMA interrupt when the operation of the descriptor line is complete.

25.4.5.1.2 Requirements for Descriptors

The following sections discuss restrictions and tips on how to correctly configure the descriptors to be used by the ADMA engine.

25.4.5.1.2.1 Data Length

There are three requirements to program descriptors:

- The minimum unit of address is 4 bytes.
- The maximum data length of each descriptor line is less than 64 KiB.
- Total length = $Length_1$ + $Length_2$+ $Length_3$ + ... + $Length_n$ must be a multiple of the block size.

If the total length of a descriptor is not a multiple of the block size, data transfer with the ADMA engine may not have been terminated. In this case, the controller returns a data time-out event and the transfer is aborted.

The block count register (the MMCHS_BLK[31:16] NBLK bit field) is defined as 16 bits and limits data transfers to a maximum of 65,535 blocks. If the ADMA data transfer size is less than or equal to the 65,535-block transfer, the block count register can be used. In this case, the total length of the descriptor table must be equivalent to "block size" by "block count." If the ADMA data transfer is greater than 65,535 blocks, the block count register must be disabled by setting the block count enable bit (MMCHS_CMD[1] BCE) to 0. In this case, the length of the data transfer is not designated by the block count but by the descriptor table.

Note

The timing for detecting the last block on the SD bus may differ, which affects control of the read transfer active (MMCHS_PSTATE[9] RTA), write transfer active (MMCHS_PSTATE[8] WTA), and DAT line active (MMCHS_PSTATE[2] DLA) bits. In case of a read operation, more blocks than required may be read from the card. The host driver must ignore an out-of-range error if the read operation is for the last block of the memory area.

25.4.5.1.2.2 Supported Features

The ADMA engine does not support the suspend/resume function. However, the stop and continue functions are available.

When the stop-at-block-gap request (the MMCHS_HCTL[16] SBGR bit) is set during the ADMA operation, the block gap event interrupt is generated when the ADMA is stopped at block gap (the MMCHS_STAT[2] BGE bit). The host controller must stop the ADMA read operation by using read wait or by stopping the SD clock. While stopping ADMA, SD commands cannot be issued.

25.4.5.1.2.3 Error Generation

When an error occurs during an ADMA transfer, the ADMA operation is stopped and the ADMA error interrupt is generated. The ADMA error state field (the MMCHS_ADMAES[1:0] AES bit field) holds the state of the ADMA when it is stopped. Software can identify the erroneous descriptor line by using the following method:

- If the ADMA stopped at ST_FDS state, the ADMA system address register (MMCHS_ADMASAL) points to the erroneous descriptor line.
- If the ADMA stopped at ST_TFR or ST_STOP state, the ADMA system address register points to the descriptor line following the erroneous one.

25.4.5.1.3 Advanced DMA Description

The ADMA is a DMA controller embedded in each eMMC controller. It can be seen as a small sequencer that fetches a descriptor line and executes the corresponding action. The base address of the descriptor table is stored in the MMCHS_ADMASAL register.

Note

Software must write the base address of the descriptor table in the MMCHS_ADMASAL register before the first use of the ADMA engine.

The ADMA program is executed according to descriptor attributes (see Section 25.4.5.1.1, *Descriptor Table Description*) and an FSM, as shown in Figure 25-17.

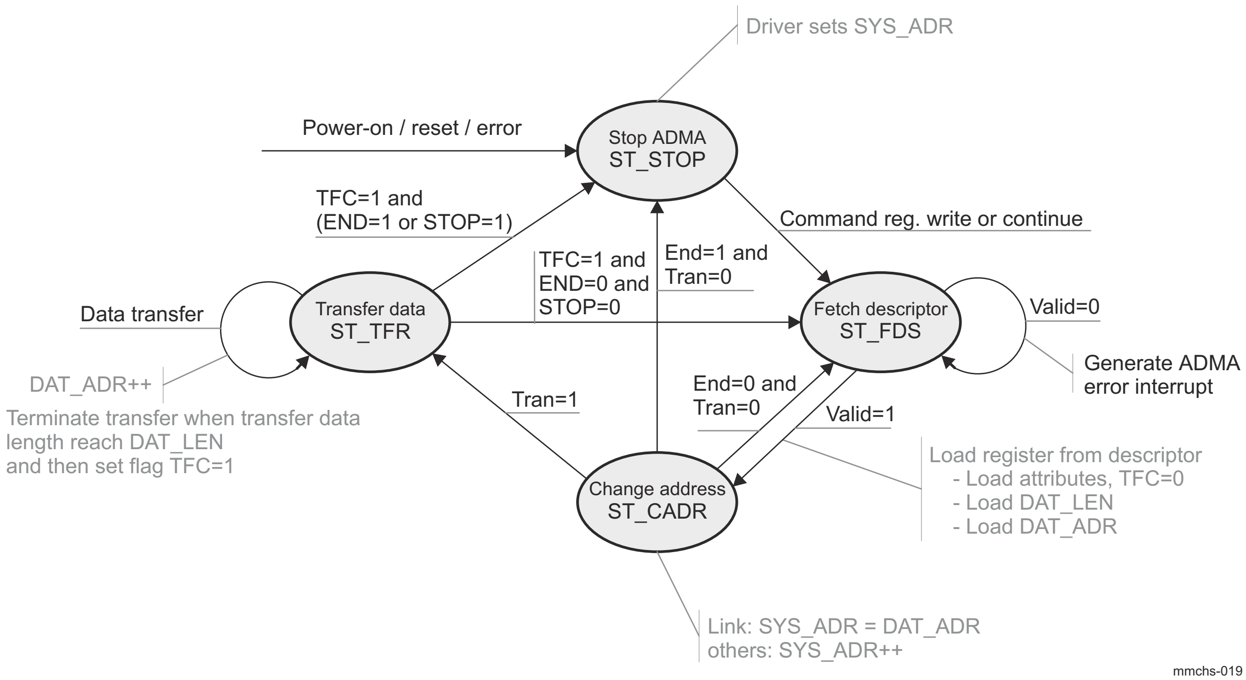


Figure 25-17. ADMA Finite State-Machine

Table 25-15 describes each state of the ADMA FSM.

Table 25-15. ADMA2 States Description

State Name	Operation
ST_FDS (fetch descriptor)	ADMA2 fetches a descriptor line and sets parameters in internal registers. It then goes to ST_CADR state.
ST_CADR (change address)	Link operation loads another descriptor address to the ADMA system address register (MMCHS_ADMASAL). In other operations, the ADMA system address register is incremented to point to the next descriptor line. If End = 0, go to ST_FDS state. **NOTE**: ADMA2 does not stop at this state if some errors occur.
ST_TFR (transfer data)	Data transfer of one descriptor line is executed between system memory and the SD card: • If data transfer continues (End = 0), go to ST_FDS state. • If data transfer completes, go to ST_STOP state.
ST_STOP (stop DMA)	ADMA2 stays in this state in the following cases: • After power-on reset (POR) or software reset • All descriptor data transfers are complete. If a new ADMA operation is stated by writing the command register, go to ST_FDS state.

Table 25-16 gives the description of each symbol used in the ADMA FSM (see Figure 25-17).

Table 25-16. ADMA FSM Symbol Definition

Symbol	Definition
SYS_ADR	ADMA system address register
SYS_ADR++	Point to next descriptor line
DAT_ADR	Data address register (internal)
DAT_LEN	Data length register (internal)
TFC	Transfer complete flag (internal)
STOP	Stop-at-block-gap request

25.4.5.2 Slave DMA Operations

The eMMC/SD/SDIOi host controller can be interfaced with a DMA controller. At the system level, the advantage is to discharge the LH of the data transfers. The module does not support wide DMA access (more than 1024 bytes) for SD cards, as specified in the *SD Card Specification* and *SD Host Controller Standard Specification*.

Note

This mode is implied by modules that are not connected to the L3_MAIN interconnect (regardless of the value of the MMCHS_CON[20] DMA_MNS bit). For more information and/or to check the value of the MMCHS_HL_HWINFO[0] MADMA_EN bit, see Section 25.4.5.1, *Master DMA Operations*.

The DMA request is issued if the following conditions are met:

- The MMCi.MMCHS_CMD[0] DE bit is set to 1 to trigger the initial DMA request (the write must be done when running the data transfer command).
- A command was emitted on the CMD line.
- There is enough space in the buffer of the eMMC/SD/SDIOi host controller to write an entire block (BLEN writes).

25.4.5.2.1 DMA Receive Mode

In a DMA block read operation (single or multiple), the request signal MMCi_DMA_RX is asserted to its active level when a complete block is written in the buffer. The block size transfer is specified in the MMCi.MMCHS_BLK[10:0] BLEN bit field.

MMCi_DMA_RX is deasserted to its inactive level when the a certain device DMA module reads one word from the buffer.

Only one request is sent per block; the DMA controller can make a 1-shot read access or several DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to the BLEN bit field block size.

New DMA requests are internally masked if the DMA has not read exactly BLEN bytes and a new complete block is not ready. Because DMA accesses are 32-bit accesses, the number of DMA reads is Integer(BLEN / 4) + 1.

The receive buffer never overflows. In multiple block transfers for block sizes larger than 512 bytes, when the buffer becomes full, the mmci_clk clock signal (provided to the card) is momentarily stopped until a certain device DMA or the MPU performs a read access, which reads a complete block in the buffer.

To summarize:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 25-18 shows DMA receive mode.

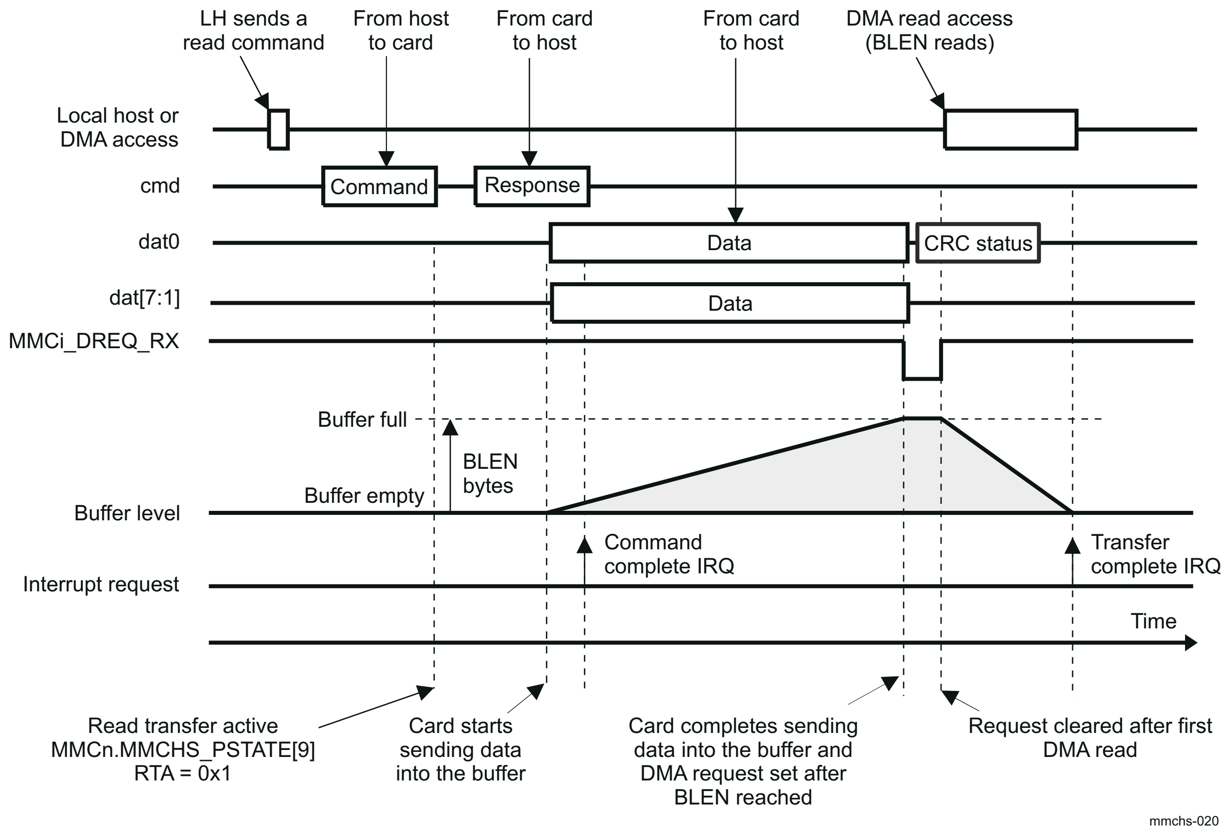


Figure 25-18. DMA Receive Mode

25.4.5.2.2 DMA Transmit Mode

In a DMA block write operation (single or multiple), the request signal MMCi_DMA_TX is asserted to its active level when a complete block is to be written to the buffer. The block size transfer is specified in the MMCi.MMCHS_BLK[10:0] BLEN bit field.

MMCi_DMA_TX is deasserted to its inactive level when a certain device DMA writes one word to the buffer.

Only one request is sent per block; the DMA controller can make a 1-shot write access or multiple write DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to the BLEN bit field block size.

New DMA requests are internally masked if the DMA has not written exactly BLEN bytes (because DMA accesses are 32-bit accesses, the number of DMA reads is Integer(BLEN / 4) + 1) and if there is not enough memory space to write a complete block in the buffer.

To summarize:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 25-19 shows DMA transmit mode.

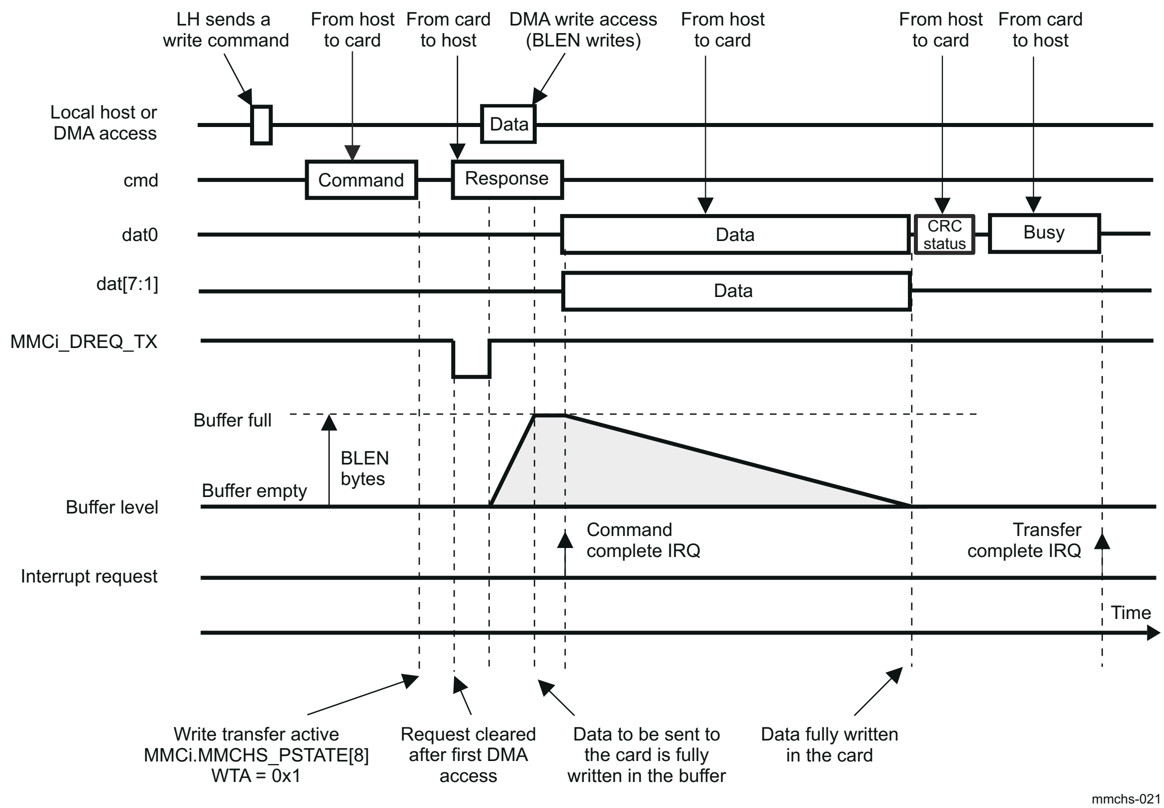


Figure 25-19. DMA Transmit Mode

25.4.6 Mode Selection

The eMMC/SD/SDIO host controller can be used in two modes: MMC and SD/SDIO. It has been designed to be the most transparent with the type of card.

The type of the card connected is differentiated by the software initialization procedure. Software identifies the type of card connected during software initialization. For each card type, there are corresponding commands. Some commands are not supported by all cards. For more information, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, and *SDIO Card Specification, Part E1*.

The purpose of the module is to transfer commands and data to whatever card is connected, respecting the protocol of the connected card.

Writes and reads to the card must respect the appropriate protocol of that card.

25.4.7 Buffer Management

25.4.7.1 Data Buffer

The eMMC/SD/SDIOi host controller uses a data buffer. This buffer transfers data from one data bus (interconnect) to another data bus (SD/SDIO or MMC card bus) and vice versa.

The buffer is the heart of the interface and ensures the transfer between the two interfaces (interconnect and the card).

To enhance performance, the data buffer is completed by a prefetch register and a post-write buffer that are not accessible by the host controller.

The read access time of the prefetch register is faster than that of the data buffer. The prefetch register allows data to be read from the data buffer at an increased speed by preloading data into the prefetch register.

The entry point of the prefetch buffer and the post-write buffer is the 32-bit MMCi.MMCHS_DATA register. A write access to the MMCi.MMCHS_DATA register followed by a read access from the MMCi.MMCHS_DATA register corresponds to a write access to the post-write buffer followed by a read access to the prefetch buffer. As a consequence, it is normal that the data of the write access to the MMCi.MMCHS_DATA register and the data of the read access to the MMCi.MMCHS_DATA register are different.

The number of 32-bit accesses to the MMCi.MMCHS_DATA register that are needed to read (or write) a data block with a size of the MMCi.MMCHS_BLK[11:0] BLEN bit field is equal to the rounded up result of BLEN divided by 4.

The maximum block size supported by the host controller is hard-coded in the MMCi.MMCHS_CAPA[17:16] MBL bit field and cannot be changed.

A read access to the MMCi.MMCHS_DATA register is allowed only when the buffer read-enable status is set to 1 (the MMCi.MMCHS_PSTATE[11] BRE bit); otherwise, a bad access (the MMCi.MMCHS_STAT[29] BADA bit) is signaled.

A write access to the MMCi.MMCHS_DATA register is allowed only when the buffer write-enable status is set to 1 (the MMCi.MMCHS_PSTATE[10] BWE bit); otherwise, a bad access (the MMCi.MMCHS_STAT[29] BADA bit) is signaled and the data are not written.

The data buffer has two modes of operation to store and read of the first and second portions of the data buffer:

- When the size of the data block to transfer is less than or equal to MEM_SIZE/2 (in double-buffering), two data transfers can occur at the same time from one data bus to the other data bus, and vice versa. The eMMC/SD/SDIOi host controller uses the two portions of the data buffer in a ping-pong manner so that storing and reading the first and second portions of the data buffer are automatically interchanged from time to time. In this way, data can be read from one portion (for instance, through a DMA read access on the interconnect bus) while data (for instance, from the card) are being stored into the other portion, and vice versa. When BLEN is less than or equal to 0x200 (that is, less than or equal to 512 bytes), each of the two portions of the buffer that can be used have a size of BLEN (that is, 32 bits × BLEN divided by 4). No more than this total size of 2 × 32 bits × BLEN divided by 4 can be used.

CAUTION

The MMCi.MMCHS_CMD[4] DDIR bit must be configured before a transfer to indicate the direction of the transfer.

Figure 25-20 and Figure 25-21 show the buffer management for a write and a read, respectively.

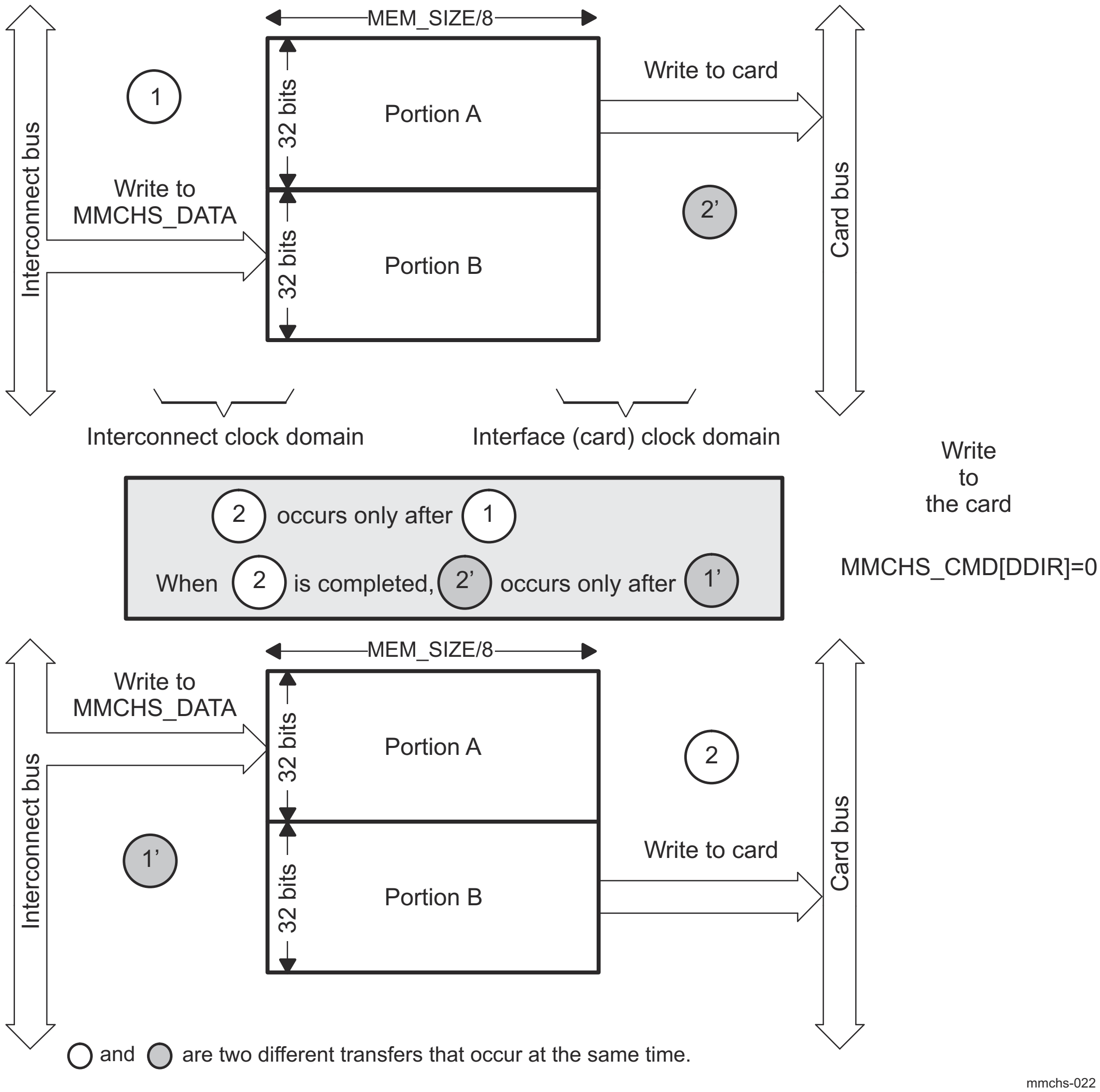


Figure 25-20. Buffer Management for a Write

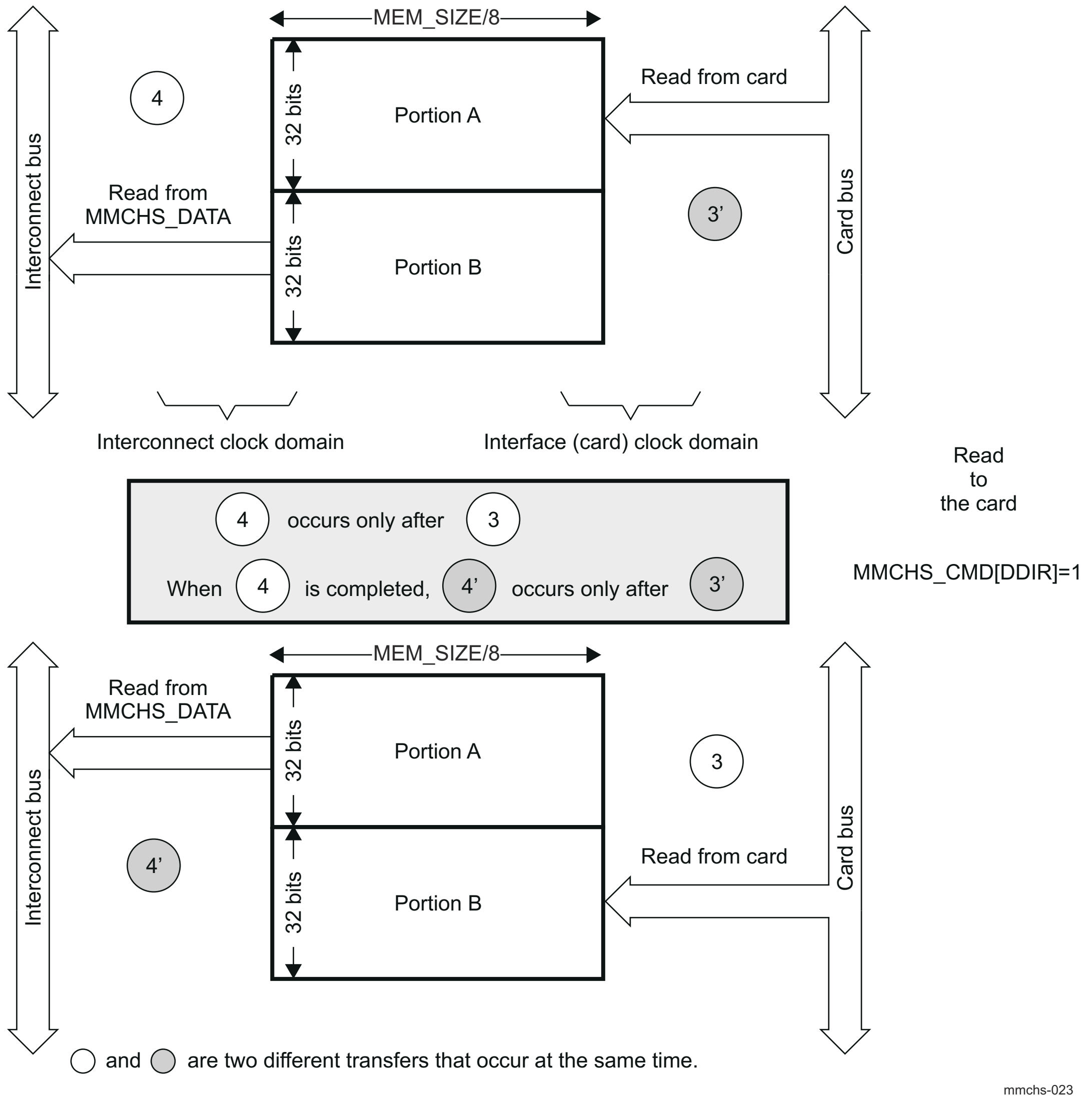


Figure 25-21. Buffer Management for a Read

- When the size of the data block to transfer is larger than MEM_SIZE / 2, only one data transfer at a time can occur from one data bus to the other data bus. The eMMC/SD/SDIOi host controller uses the entire data buffer as a single portion.

  In this mode, a bad access (the MMCi.MMCHS_STAT[29] BADA bit) is signaled when two data transfers occur at the same time from one data bus to the other data bus, and vice versa.

25.4.7.1.1 Memory Size, Block Length, and Buffer-Management Relationship

The maximum block length and buffer management that can be targeted by the system depend on the memory depth setting (see Table 25-17).

Note

Double-buffering is always the buffer management for large memory depth.

Table 25-17. Memory Size, BLEN, and Buffer Relationship

Memory Size (MMCHS_HL_HWINFO[5:2] MEM_SIZE in bytes)	512	1024
Maximum block length supported	512	1024

Table 25-17. Memory Size, BLEN, and Buffer Relationship (continued)

Memory Size (MMCHS_HL_HWINFO[5:2] MEM_SIZE in bytes)	512	1024
Double-buffering for maximum block length	N/A	BLEN <= 512
Single-buffering for block length	BLEN <= 512	512 < BLEN <= 1024

Note

For single-buffering management, throughput on the MMC bus interface deteriorates in multiblock transfers, because the controller must wait for the filling or emptying of the buffer between each block transfer on the MMC bus. The clock is maintained on write MMC transfers (the MMCHS_CMD[4] DDIR bit is 0) and halted on read MMC transfers (the MMCHS_CMD[4] DDIR bit is 1).

25.4.7.1.2 Data Buffer Status

The data buffer status is defined in the following interrupt status register and status register:

- Interrupt status registers:
  - MMCi.MMCHS_STAT[29] BADA: Bad access to data space
  - MMCi.MMCHS_STAT[5] BRR: Buffer read ready
  - MMCi.MMCHS_STAT[4] BWR: Buffer write ready
- Status registers:
  - MMCi.MMCHS_PSTATE[11] BRE: Buffer read enable
  - MMCi.MMCHS_PSTATE[10] BWE: Buffer write enable

25.4.8 Transfer Process

The process of a transfer depends on the type of command. It can be with or without a response, and with or without data.

25.4.8.1 Different Types of Commands

Different types of commands are specific to the MMC, SD, and SDIO cards. For more information, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, *SDIO Card Specification, Part E1*; or *the SD Card Specification, Part A2, SD Host Controller Standard Specification*.

25.4.8.2 Different Types of Responses

Different types of responses are specific to the eMMC, SD, and SDIO cards. For more information, see the *Multimedia Card System Specification*; *SD Memory Card Specifications*; *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2, SD Host Controller Standard Specification*.

Table 25-18 describes how the eMMC, SD, and SDIO responses are stored in the MMCHS_RSPxx registers.

Table 25-18. MMC, SD, SDIO Responses in the MMCHS_RSPxx Registers

Type of Response	Response Field	Response Register
R1, R1b (normal response), R3, R4, R5, R5b, R6, R7	RESP[39:8][1]	MMCHS_RSP10[31:0]
R1b (Auto CMD12 response), R1(Auto CMD23 response)	RESP[39:8][1]	MMCHS_RSP76[31:0]
R2	RESP[127:0][1]	MMCHS_RSP76[31:0] MMCHS_RSP54[31:0] MMCHS_RSP32[31:0] MMCHS_RSP10[31:0]

(1) RESP refers to the command response format described in the specifications mentioned.

When the host controller modifies part of the MMCHS_RSPxx registers, it preserves the unmodified bits.

The host controller stores the Auto CMD12 response in the MMCHS_RSP76[31:0] register because the host controller may execute multiple block data transfers on the DATA line concurrently with a command. This allows the host controller to avoid overwriting the response of Auto CMD12 with the command response stored in the MMCHS_RSP10 register, and vice versa.

While executing Auto CMD23 the response of CMD23 is saved in the MMCHS_RSP76[31:0] register and the response of multiple-block read and write command is saved in the MMCHS_RSP10 register. The response error of CMD23 is indicated in the MMCHS_AC12 register, bits [7:0].

25.4.9 Transfer or Command Status and Errors Reporting

Flags in the eMMC/SD/SDIOi host controller show the status of communication with the card:

- A time-out (of a command, data, or response)
- A CRC error

Error conditions generate interrupts. For more information, see Table 25-19 and the register description.

Table 25-19. CC and TC Values Upon Error Detected

Error Hold in MMCi.MMCHS_STAT		CC	TC	Comments
29	BADA			No dependency with CC or TC BADA is related to the MMCHS_DATA register accesses. Its assertion does not depend on the ongoing transfer.
28	CERR	1		CC is set upon CERR.
22	DEB		1	TC is set upon DEB.
21	DCRC		1	TC is set upon DCRC.
20	DTO			DTO and TC are mutually exclusive. DCRC and DEB cannot occur with DTO.
19	CIE	1		CC is set upon CIE.
18	CEB	1		CC is set upon CEB.
17	CCRC	1		CC can be set upon CCRC. See CTO comment.
16	CTO			CTO and CC are mutually exclusive. CIE, CEB, and CERR cannot occur with CTO. CTO can occur at the same time as CCRC: It indicates a command abort due to contention on the CMD line. In this case no CC appears.

A MMCHS_STAT[20] DTO event can be asserted in the following conditions:

- Busy time-out for R1b, R5b response type
- Busy time-out after write CRC status
- Write CRC status time-out
- Read data time-out
- Boot acknowledge time-out

25.4.9.1 Busy Time-Out for R1b, R5b Response Type

Figure 25-22 shows the DTO event condition asserted when there is a busy time-out for Rb1, R5b response.

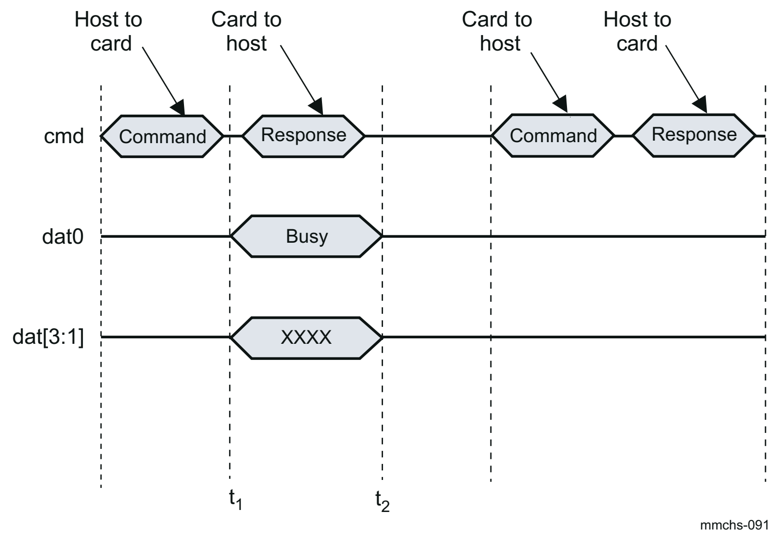


Figure 25-22. Busy Time-Out for R1b, R5b Response Type

t_1 – Data time-out counter is loaded and starts after R1b, R5b response type.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

25.4.9.2 Busy Time-Out After Write CRC Status

Figure 25-23 shows the DTO event condition asserted when there is a busy time-out after write CRC status.

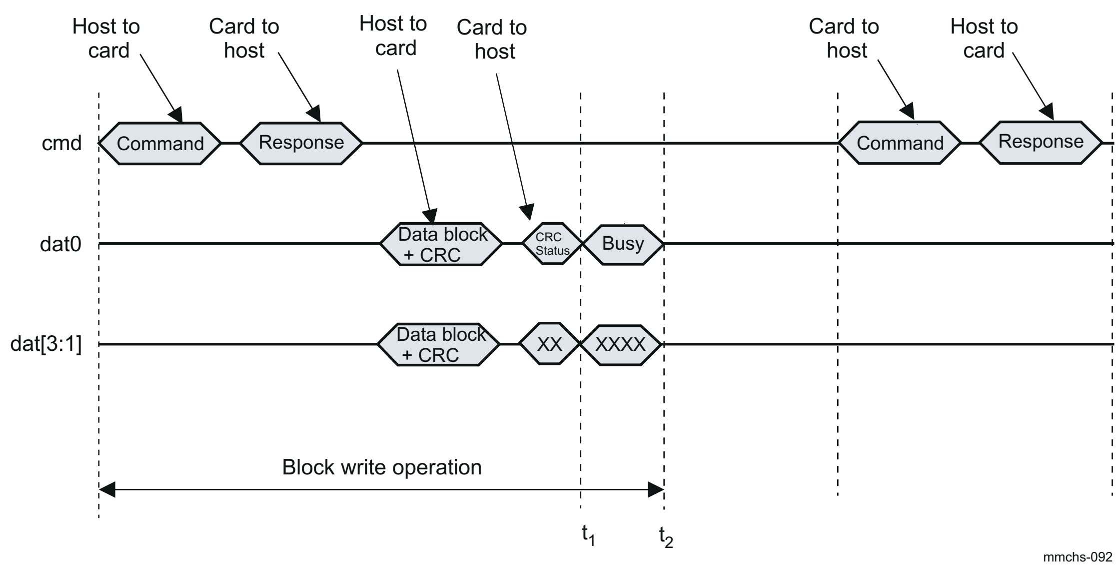


Figure 25-23. Busy Time-Out After Write CRC Status

t_1 – Data time-out counter is loaded and starts after CRC status.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

25.4.9.3 Write CRC Status Time-Out

Figure 25-24 shows the DTO event condition asserted when there is a write CRC status time-out.

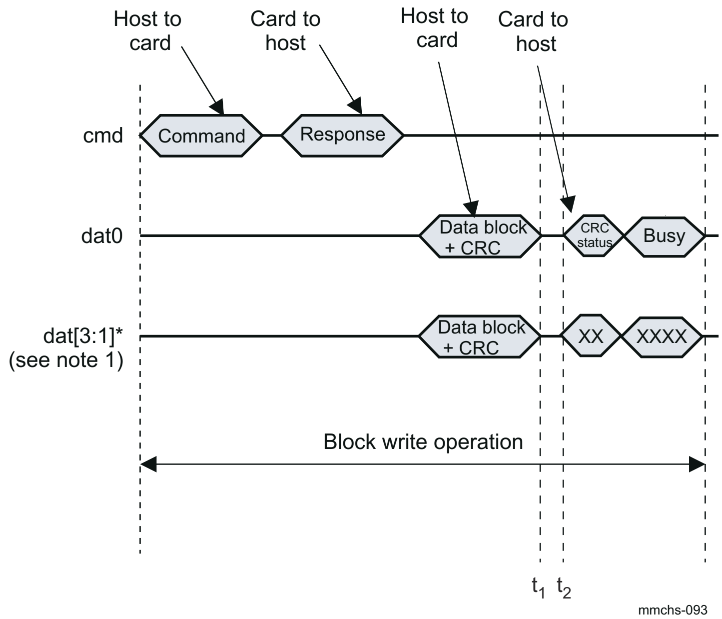


Figure 25-24. Write CRC Status Time-Out

t_1 – Data time-out counter is loaded and starts after data block + CRC.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

25.4.9.4 Read Data Time-Out

Figure 25-25 shows the DTO event condition asserted when there is a read data time-out.

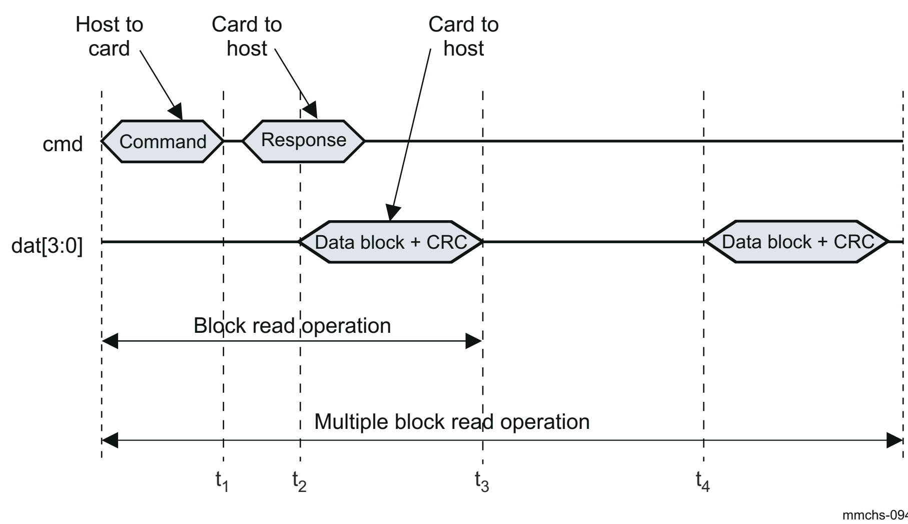


Figure 25-25. Read Data Time-Out

t_1 – Data time-out counter is loaded and starts after command transmission.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

t_3 – Data time-out counter is loaded and starts after data block + CRC transmission.

t_4 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

25.4.9.5 Boot Acknowledge Time-Out

Figure 25-26 shows the DTO event condition asserted when there is a boot acknowledge time-out and CMD0 is used.

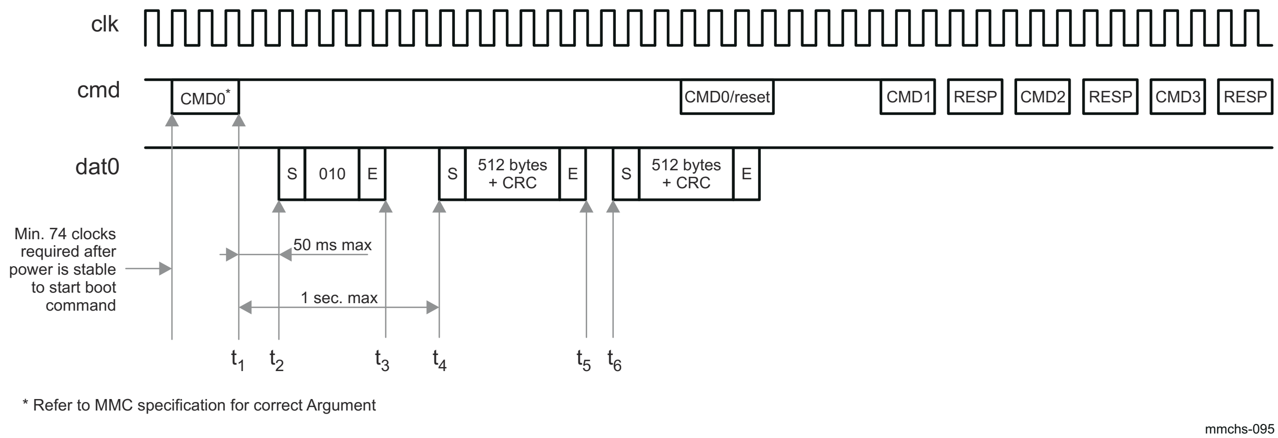


Figure 25-26. Boot Acknowledge Time-Out When Using CMD0

t_1 – Data time-out counter is loaded and starts after CMD0.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

t_3 – Data time-out counter is loaded and starts.

t_4 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

t_5 – Data time-out counter is loaded and starts after data + CRC transmission.

t_6 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

Figure 25-27 shows the DTO event condition asserted when there is a boot acknowledge time-out when the CMD line is tied to 0.

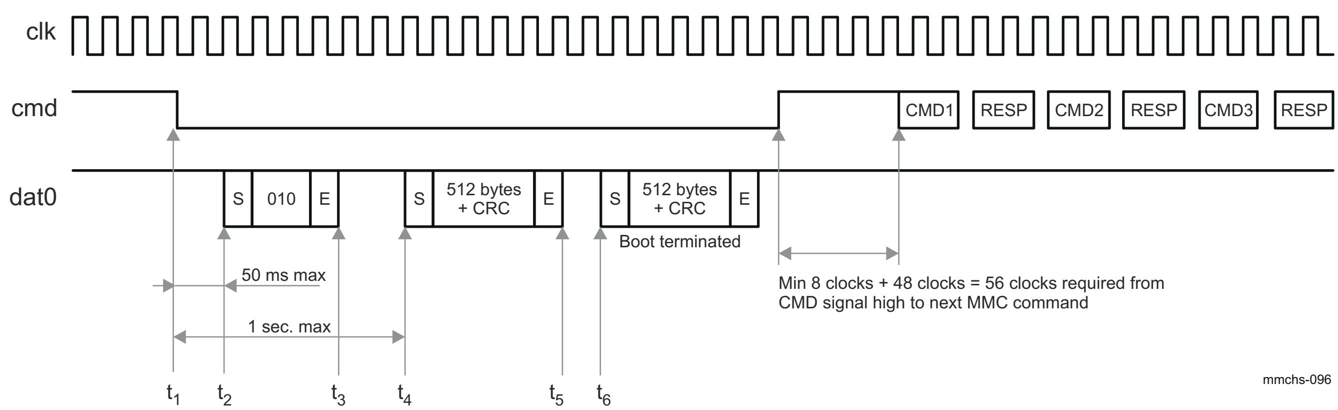


Figure 25-27. Boot Acknowledge Time-Out When CMD Line Tied to 0

t_1 – Data time-out counter is loaded and starts after the CMD line is tied to 0.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

t_3 – Data time-out counter is loaded and starts.

t_4 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

t_5 – Data time-out counter is loaded and starts after data + CRC transmission.

t_6 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

25.4.10 Auto Command 12 Timings

With the UHS definition of SD cards with higher frequency for MMC clock up to 208, the SD standard imposes a specific timing for the arrival of the auto command 12 (Auto CMD12) end bit.

25.4.10.1 Auto CMD12 Timings During Write Transfer

A margin named Ncrc in the range of two to eight cycles has been defined for SDR50 and SDR104 card components for write data transfers, because the Auto CMD12 end bit must arrive after the CRC status end bit.

Figure 25-28 shows the Auto CMD12 timings during write transfer.

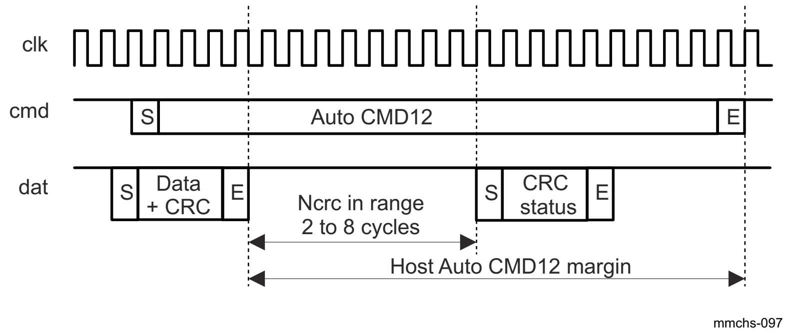


Figure 25-28. Auto CMD12 Timings During Write Transfer

The host controller has a margin of 18 clock cycles to enure that the Auto CMD12 end bit arrives after the CRC status. This margin does not depend on the MMC/SD bus configuration, DDR, or standard transfer, 1-, 4-, or 8-bit bus width.

25.4.10.2 Auto CMD12 Timings During Read Transfer

With UHS cards, the gap timing between two successive cards is extended from two cycles to four cycles. It provides more flexibility for the host Auto CMD12 arrival to receive the last complete and reliable block. The MMCHS controller follows only the left border case defined by the SD UHS specification.

Figure 25-29 shows Auto CMD12 timings during read transfer.

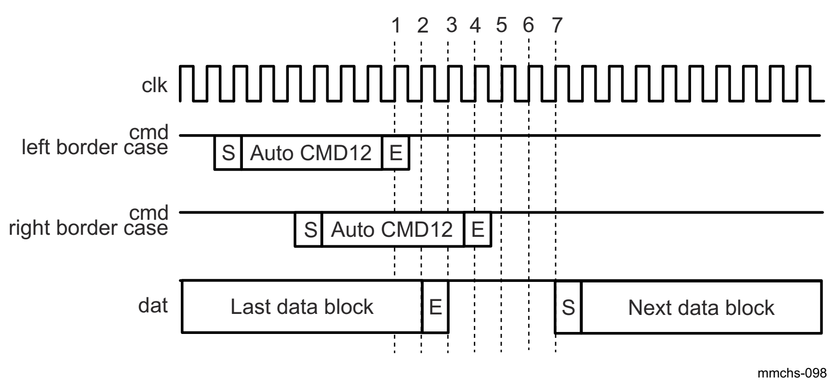


Figure 25-29. Auto CMD12 Timings During Read Transfer

The Auto CMD12 arrival sent by the host controller is not sensitive to the MMC/SD bus configuration, whether it is a DDR or standard transfer and whether it is a 1-, 4-, or 8-bit bus width transfer.

25.4.11 Transfer Stop

Whenever a transfer is initiated, the transmission can be stopped before it finishes. Several cases are possible, depending on the transfer type:

- Multiple-block-oriented transfers (transfer length is known)
- Continuous stream transfers (transfer has an infinite length)

Note

Because the eMMC/SD/SDIOi controller manages transfers based on a block granularity, the buffer accepts a block only if there is enough space to store it completely. Consequently, if a block is pending in the buffer, no command is sent to the card because the card clock will be shut off by the controller.

The eMMC/SD/SDIOi controller includes three features that make a transfer stop more convenient and easier to manage:

- Auto CMD12/Auto CMD23 (for eMMC and SD only):

  Auto CMD12/Auto CMD23 feature is enabled by setting the MMCi.MMCHS_CMD[3:2] ACEN bit field to 0x1 or to 0x2 respectively (this setting is relevant for an MMC/SD transfer with a known number of blocks to transfer). When the Auto CMD12/Auto CMD23 feature is enabled, the eMMC/SD/SDIOi controller automatically issues a CMD12/CMD23 command when the expected number of blocks is exchanged.
- Stop at block gap:

  This feature is enabled by setting the MMCi.MMCHS_HCTL[16] SBGR bit to 0x1. When enabled, this capability holds the transfer on until the end of a block boundary. If a stop transmission is needed, software can use this pause to send a CMD12 to the card.
- ADMA mode:

  For ADMA-capable modules (MMC1 and MMC2) (for more information, see Section 25.4.5, *DMA Modes*), the last instruction can stop the transfer (the END bit is enabled in the descriptor line).

Note

For eMMC and SD cards, the stop-at-block-gap feature is not supported in read mode.

For SDIO cards, this setting can be supported in read mode if the card has read-wait capability.

Note

In SDR104 mode Auto CMD23 is used to stop multiple block read/write operation instead of Auto CMD12. In the other bus speed modes, if the card supports CMD23, Auto CMD23 is used instead of Auto CMD12.

Table 25-20 shows the common way to stop a transfer, indicating the command to send and the features to enable.

Table 25-20. eMMC/SD/SDIOi Controller Transfer Stop Command Summary

	Write Transfer		Read Transfer	
	SD/MMC	SDIO	SD/MMC	SDIO
Single block	Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.

Table 25-20. eMMC/SD/SDIOi Controller Transfer Stop Command Summary (continued)

		Write Transfer		Read Transfer	
Multiblocks (finite or infinite)	Before the programmed block boundary	Send CMD12/ CMD23. Wait TC.	Send CMD52. Wait TC.	Send CMD12/ CMD23. Wait TC.	Send CMD52. Wait TC.
	Stop at the end of the transfer (finite transfer only)	Auto CMD12/Auto CMD23 active. Transfer ends automatically. Wait TC.	Set MMCi.MMCHS_HCTL [16] SBGR bit to 0x1. Send CMD52. Wait TC.	Auto CMD12/Auto CMD23 active. Transfer ends automatically. Wait TC.	**If READ_WAIT supported** Stop at block gap. Wait TC. **If READ_WAIT not supported** Send CMD52. Wait TC.

Note

The eMMC/SD/SDIOi controller sends the stop command to the card on a block boundary, regardless of when the command was written to the controller registers.

25.4.12 Output Signals Generation

The eMMC/SD/SDIO output signals can be driven on the falling edge or rising edge, depending on the MMCHS_HCTL[2] HSPE bit.

25.4.12.1 Generation on Falling Edge of MMC Clock

The controller defaults to this mode to maximize hold timings. In this case, the MMCHS_HCTL[2] HSPE bit is set to 0.

Figure 25-30 shows the output signals of the module when generating from the falling edge of the MMC clock.

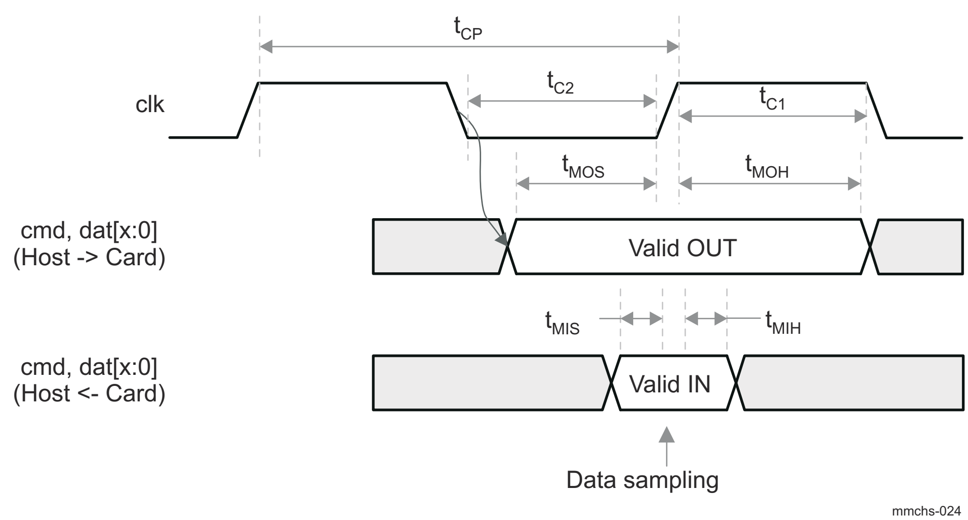


Figure 25-30. Output Driven on Falling Edge

25.4.12.2 Generation on Rising Edge of MMC Clock

This mode is intended to increase setup timings. This feature is activated by setting the MMCHS_HCTL[2] HSPE bit to 1.

Note

Do not use this feature in DDR mode (when the MMCHS_CON[19] DDR bit is set to 1).

Figure 25-31 shows the output signals of the module when generating from the rising edge of the MMC clock.

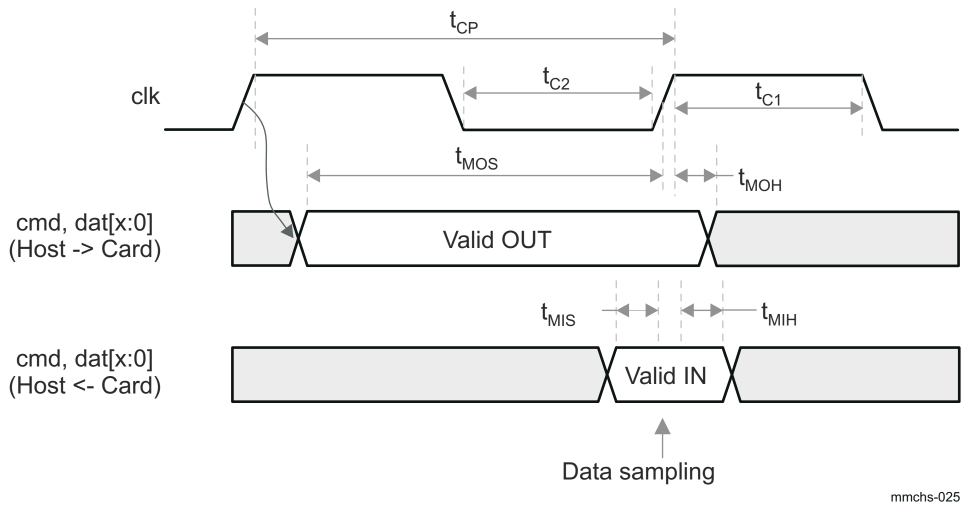


Figure 25-31. Output Driven on Rising Edge

25.4.13 Sampling Clock Tuning

In UHS-I mode the SD bus operates in high clock frequency mode and the data window from the card on CMD and DAT lines get smaller. The position of the data window varies depending on the card and the host system. To adjust the sampling clock when SDR104/HS200 operation mode is used the eMMC/SD/SDIOi host controller supports a tuning circuit. This tuning circuit is a dedicated DLL which delays the clock signal used for data sampling. The DLL is not part of the eMMC/SD/SDIOi host controller. It is instantiated at top level between the IOs and the host controller. There are two DLLs. One for MMC1 when SDR104 mode is used and one for MMC2 when HS200 mode is used.

In the default, lower frequency operation, a fixed sampling clock is used to receive signals on CMD and DAT lines. Before using the SDR104/HS200 or SDR50 (if MMCHS_CAPA2[13] TSDR50 = 0x1) modes software must execute the tuning procedure at the initialization sequence regardless of MMCHS_CAPA2[15:14] RTM value.

The software starts the tuning sequence by setting MMCHS_DLL[20] SWT to 1. Then it issues CMD19 for a SD card or CMD21 for an eMMC device repeatedly while cycling through 32 DLL ratios and recording pass/fail results for each. Thereafter the MMCHS_AC12[23] SCLK_SEL bit is checked and if it is set to 0 this indicates that the tuning procedure has failed. When MMCHS_AC12[23] SCLK_SEL is set to 1, this indicates that the tuning procedure has completed successfully. For more information about the DLL tuning procedure, see Section 25.5.1.2.4 , *SDR104/HS200 DLL Tuning Procedure*.

Note

The eMMC/SD/SDIO controller supports a Conflict Error (CFT Error) on the CMD line. This error detection logic must be disabled for SDR104/HS200 mode by setting the MMCHS_DLL[20] SWT bit to 0x1 during the DLL tuning procedure. This value (0x1) must be kept while SDR104/HS200 mode is used.

The eMMC/SD/SDIO controller generates a CFT error when it detects that the value sent on the CMD line is not the same as the value returned, indicating an external source is overriding the value.

25.4.14 Card Boot Mode Management

Boot operation mode lets the eMMC/SD/SDIOi host controller read boot data from the connected slave (MMC device) by keeping the CMD line low after power on (or sending CMD0 with a specific argument) before issuing CMD1. The data can be read from the boot area or user area, depending on the register setting.

Power-on boot defines a way for the boot code to be accessed by the eMMC/SD/SDIOi host controller without an upper-level software driver, thus speeding the time it takes for a controller to access the boot code.

The two possible ways to issue a boot command (issuing a CMD0 or driving the CMD line to 0 during the whole boot phase) are described in the following sections.

25.4.14.1 Boot Mode Using CMD0

Figure 25-32 shows the timing diagram of a boot sequence using CMD0.

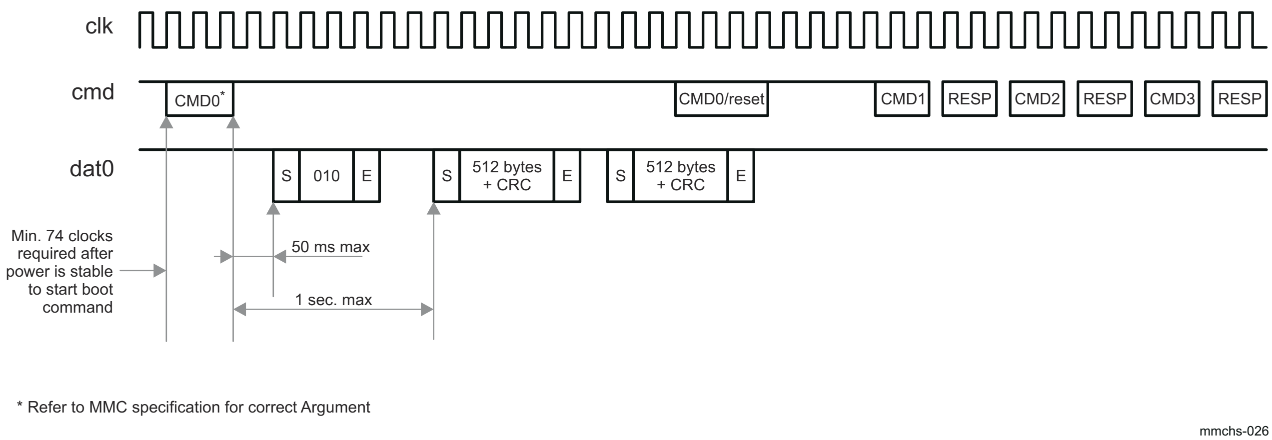


Figure 25-32. Boot Mode Using the CMD0 Timing Diagram

Note

Refer to MMC specification for correct Argument.

For more information about how to configure the eMMC/SD/SDIO host controller, see Section 25.5.1.2.3.1, *Boot Using the CMD0*.

25.4.14.2 Boot Mode With CMD Line Tied to 0

Figure 25-33 shows the timing diagram of a boot sequence with CMD line tied to 0.

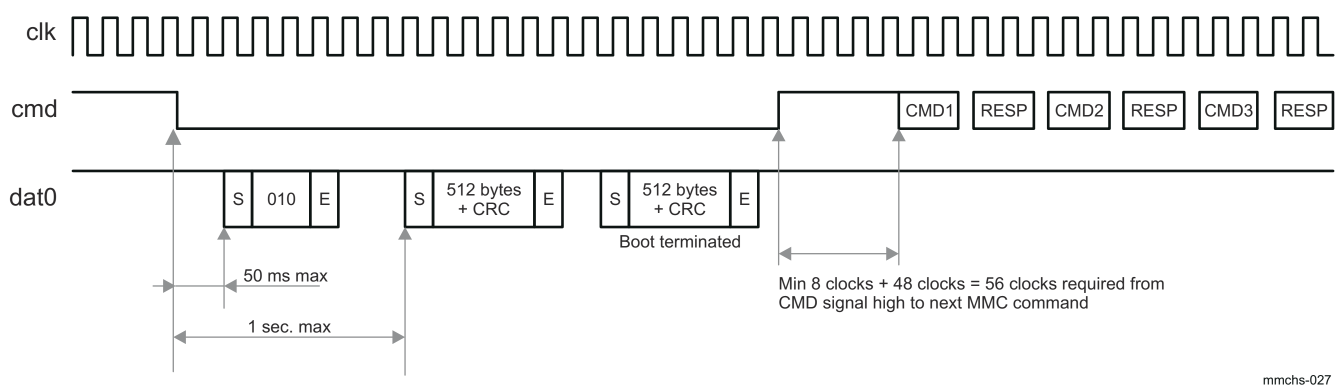


Figure 25-33. Boot Mode With CMD Line Tied to 0 Timing Diagram

For more information about how to configure the eMMC/SD/SDIO host controller, see Section 25.5.1.2.3.2, *Boot With CMD Line Tied to 0*.

25.4.15 MMC CE-ATA Command Completion Disable Management

The eMMC/SD/SDIOi host controller supports CE-ATA features, in particular the detection of the command completion token. When a command that requires a CCS (the MMCHS_CON[12] CEATA bit is set to 1 and the

MMCHS_CMD[3:2] ACEN bit field is set to 0x1) is launched, the host system is no longer allowed to emit a new command in parallel to the data transfer unless it is a command completion disable token.

The settings to emit a command completion disable token are:

- Set the MMCHS_CON[12] CEATA bit to 1.
- Set the MMCHS_CON[2] HR bit to 1.
- Clear the MMCHS_ARG register.
- Write into the MMCHS_CMD register with the value 0x00000000.

When a command completion disable token was emitted (that is, the MMCHS_STAT[0] CC bit is received), the host system is again allowed to emit another type of command (for example, a CMD12 to abort transfer).

A critical case can be encountered when CCSD is emitted during the last data block transfer, and the sequence on the command line is sent close to the CCS token sent by the card.

Three possible cases are:

- CCS is received immediately before CCSD is emitted:

  An interrupt CIRQ is generated when CCS is detected, CCSD is transmitted to the card, and then an interrupt CC is generated when CCSD ends. In this case, the card considers the CCSD sequence.
- CCS is not generated or is generated during the CCSD transfer:

  The CCS bit cannot be detected (conflict is not possible because they drive the same level on the command line, and no CIRQ interrupt is generated; a CC interrupt is generated when CCSD ends).
- CCS is generated without CCSD token required:

  Only the interrupt CIRQ is generated when CCS is detected.

25.4.16 Test Registers

Test registers are available to comply with the *SD Host Controller Specification*. This feature is useful to generate interrupts manually for driver debugging.

The force event register (MMCHS_FE) is used to control the error status and error interrupt status for Auto CMD12 and Auto CMD23.

The system test register (MMCHS_SYSTEST) is used to control the signals that connect to I/O pins when the module is configured in the system test mode (the MMCHS_CON[4] MODE bit = 1) for boundary connectivity verification.

The MMCHS_HCTL[7] CDSS and MMCHS_HCTL[6] CDTL bits enable manual control of MMCHS_PSTATE[16] CINS and interrupt generating indicated in MMCHS_STAT[7] CREM and MMCHS_STAT[6] CINS.

25.4.17 eMMC/SD/SDIO Hardware Status Features

Table 25-21 describes the eMMC/SD/SDIO hardware status features.

Table 25-21. eMMC/SD/SDIO Hardware Status Features

Feature	Type	Register/Bit Field	Description
Interrupt flags		See Section 25.4.4, *Interrupt Requests.*	
CMD line signal level	Status	MMCHS_PSTATE[24] CLEV	Indicates the level of the command line
DAT lines signal level	Status	MMCHS_PSTATE[23:20] DLEV	Indicates the level of the data lines
Write protect switch pin level	Status	MMCHS_PSTATE[19] WP	Indicates whether the SD card is write protected or not.
Card detect pin level	Status	MMCHS_PSTATE[18] CDPL	Indicates the level of the mmci_sdcd signal/pad

Table 25-21. eMMC/SD/SDIO Hardware Status Features (continued)

Feature	Type	Register/Bit Field	Description
Card State Stable	Status	MMCHS_PSTATE[17] CSS	Used for testing. Indicates mmci_sdcd stable state
Card inserted	Status	MMCHS_PSTATE[16] CINS	Indicates whether the SD card is inserted
Buffer read enable	Status	MMCHS_PSTATE[11] BRE	Readable data exists in the buffer.
Buffer write enable	Status	MMCHS_PSTATE[10] BWE	Indicates whether there is enough space in the buffer to write BLEN bytes of data
Read transfer active	Status	MMCHS_PSTATE[9] RTA	Used to detect completion of a read transfer.
Write transfer active	Status	MMCHS_PSTATE[8] WTA	Indicates a write transfer active
Re - Tuning Request	Status	MMCHS_PSTATE[3] RTR	Indicates whether the sampling clock needs re - tuning or not.
Data line active	Status	MMCHS_PSTATE[2] DLA	Indicates whether the data lines are active
Command Inhibit (DAT lines)	Status	MMCHS_PSTATE[1] DATI	Indicates whether issuing of command using data lines is allowed. For example, commands with busy mechanism (that is, R1b response), data transfer commands.
Command inhibit (CMD line)	Status	MMCHS_PSTATE[0] CMDI	Indicates whether issuing of command using command line is allowed

Table 25-22 describes the eMMC/SD/SDIO preset value features.

Table 25-22. eMMC/SD/SDIO Preset Value Registers

Feature	Type	Register	Description
Preset value register	Status	MMCHS_PVINITSD	Preset Values for Initialization and Default Speed modes
Preset value register	Status	MMCHS_PVHSSDR12	Preset Values for High Speed and SDR12 speed modes
Preset value register	Status	MMCHS_PVSDR25SDR50	Preset Values for SDR25 and SDR50 speed modes
Preset value register	Status	MMCHS_PVSDR104DDR50	Preset Values for SDR104 and DDR50 speed modes

25.5 eMMC/SD/SDIO Programming Guide

25.5.1 Low-Level Programming Models

25.5.1.1 Global Initialization

25.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module must be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the eMMC/SD/SDIO modules. For more information, see Section 25.3, *eMMC/SD/SDIO Integration*, and Section 25.2, *eMMC/SD/SDIO Environment*.Table 25-23 shows the global initialization of surrounding modules.

Table 25-23. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing and configuration must be set in the control module. See *Control Module* .
DMA_CROSSBAR	DMA_CROSSBAR configuration must be done to allow module DREQs to be mapped to certain device DMA line. For more information see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
Device DMAs	Device DMAs configuration must be done to enable the module DMA channel requests.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see *Interrupt Controllers*

25.5.1.1.2 eMMC/SD/SDIO Host Controller Initialization Flow

Table 25-24 shows the general boot process.

Table 25-24. eMMC/SD/SDIO Controller Meta Initialization Steps

Step	Access Type	Register/Bit Field/Programming Model	Value
Initialize clocks.		See Section 25.5.1.1.2.1, *Enable Interface and Functional Clock for MMC Controller.*	
Software reset of the controller.		See Section 25.5.1.1.2.2, *MMCHS Soft Reset Flow*.	
Set module hardware capabilities.		See Section 25.5.1.1.2.3,*Set MMCHS Default Capabilities*.	
Set module idle and wake-up modes.		See Section 25.5.1.1.2.4, *Wake-Up Configuration*.	

25.5.1.1.2.1 Enable Interface and Functional Clock for MMC Controller

Before any MMCHS register access, the MMCHS interface clock and functional clock in the PRCM module registers must be enabled. See *Clock Domain Module Attributes*, in *Power, Reset, and Clock Management*.

25.5.1.1.2.2 MMCHS Soft Reset Flow

Figure 25-34 shows the soft reset process of the MMCHS controller.

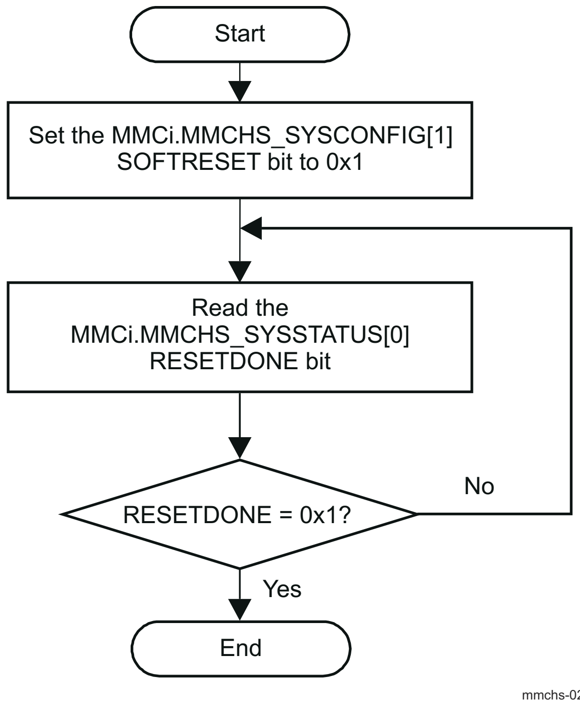


Figure 25-34. eMMC/SD/SDIO Controller Software Reset Flow

Table 25-25. Register Call Summary for Main Sequence – Software Reset Flow

Register Name	Register Name
MMCHS_SYSCONFIG	MMCHS_SYSSTATUS

25.5.1.1.2.3 Set MMCHS Default Capabilities

Software must read capabilities (in boot ROM, for example) and is allowed to set (write) the MMCi.MMCHS_CAPA[26:24] and MMCi.MMCHS_CUR_CAPA[23:0] bit fields before the eMMC/SD/SDIO host driver is started.

25.5.1.1.2.4 Wake-Up Configuration

Table 25-26 describes the MMCHS controller wake-up configuration.

Table 25-26. eMMC/SD/SDIO Controller Wake-Up Configuration

Step	Access Type	Register/Bit Field/Programming Model	Value
Configure wake-up bit (if necessary).	W	MMCi.MMCHS_SYSCONFIG[2] ENAWAKEUP	0x1
Enable wake-up events on SD card interrupt (if necessary).	W	MMCi.MMCHS_HCTL[24] IWE	0x1
SDIO card only: Enable card interrupt (if necessary).	W	MMCi.MMCHS_IE[8] CIRQ_ENABLE	0x1

25.5.1.1.2.5 MMC Host and Bus Configuration

Figure 25-35 shows the MMC bus configuration process.

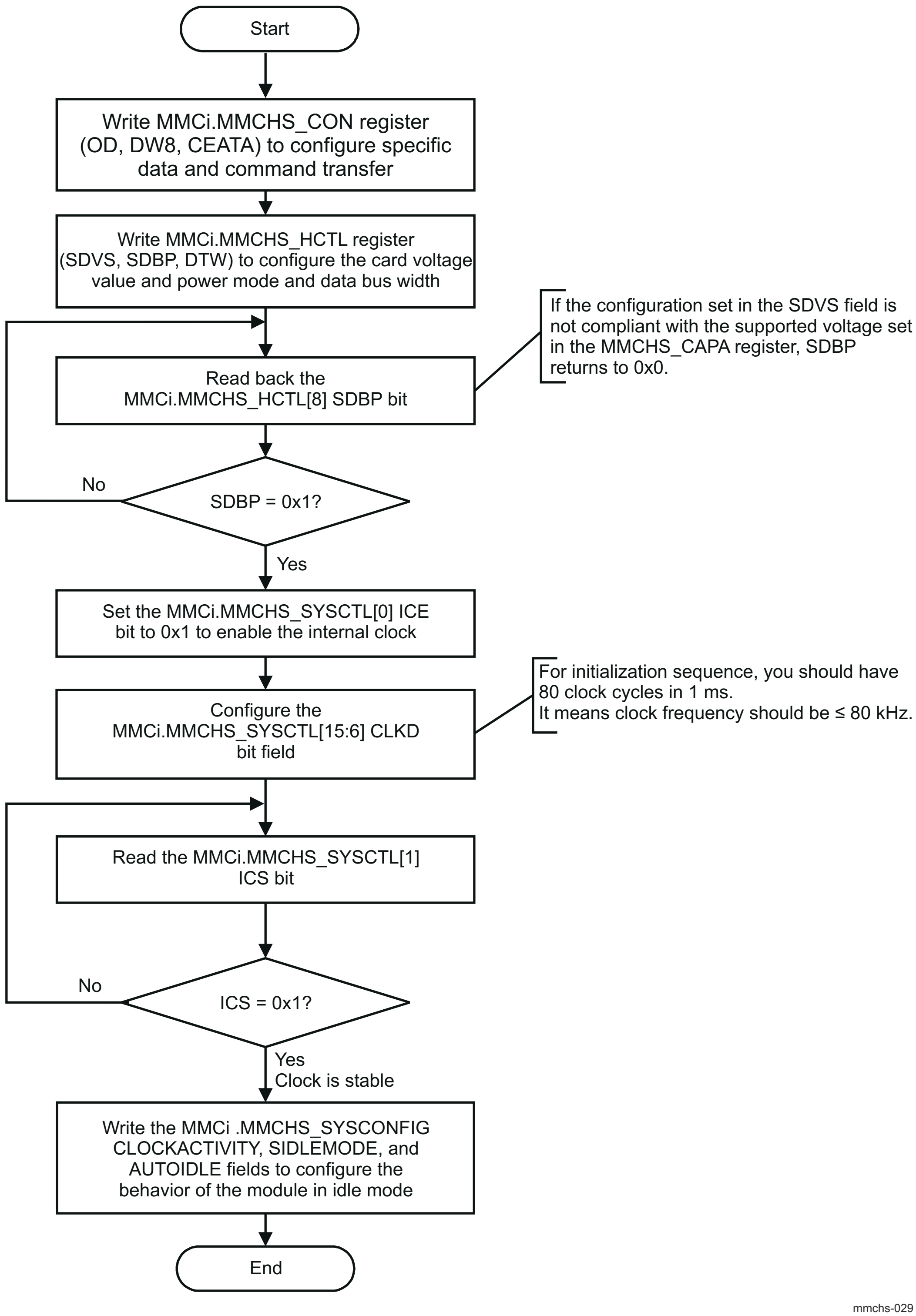


Figure 25-35. eMMC/SD/SDIO Controller Bus Configuration

Table 25-27. Register Call Summary for Main Sequence – Bus Configuration

Register Name	Register Name
MMCHS_CON	MMCHS_HCTL
MMCHS_SYSCONFIG	MMCHS_SYSCTL

25.5.1.2 Operational Modes Configuration

25.5.1.2.1 Basic Operations for eMMC/SD/SDIO Host Controller

The eMMC/SD/SDIO host controller performs data transfers: data to card (referred to as write transfers) and data from card (referred to as read transfers).

The host controller requires transfers to run on a block-by-block basis rather than on a DMA burst size basis. A single DMA request (or block request interrupt) is signaled for each block. Pipelining is supported as long as the block size is less than one half of the memory buffer size.

25.5.1.2.1.1 Card Detection, Identification, and Selection

Figure 25-36 and Figure 25-37 show the card detection, identification and selection process.

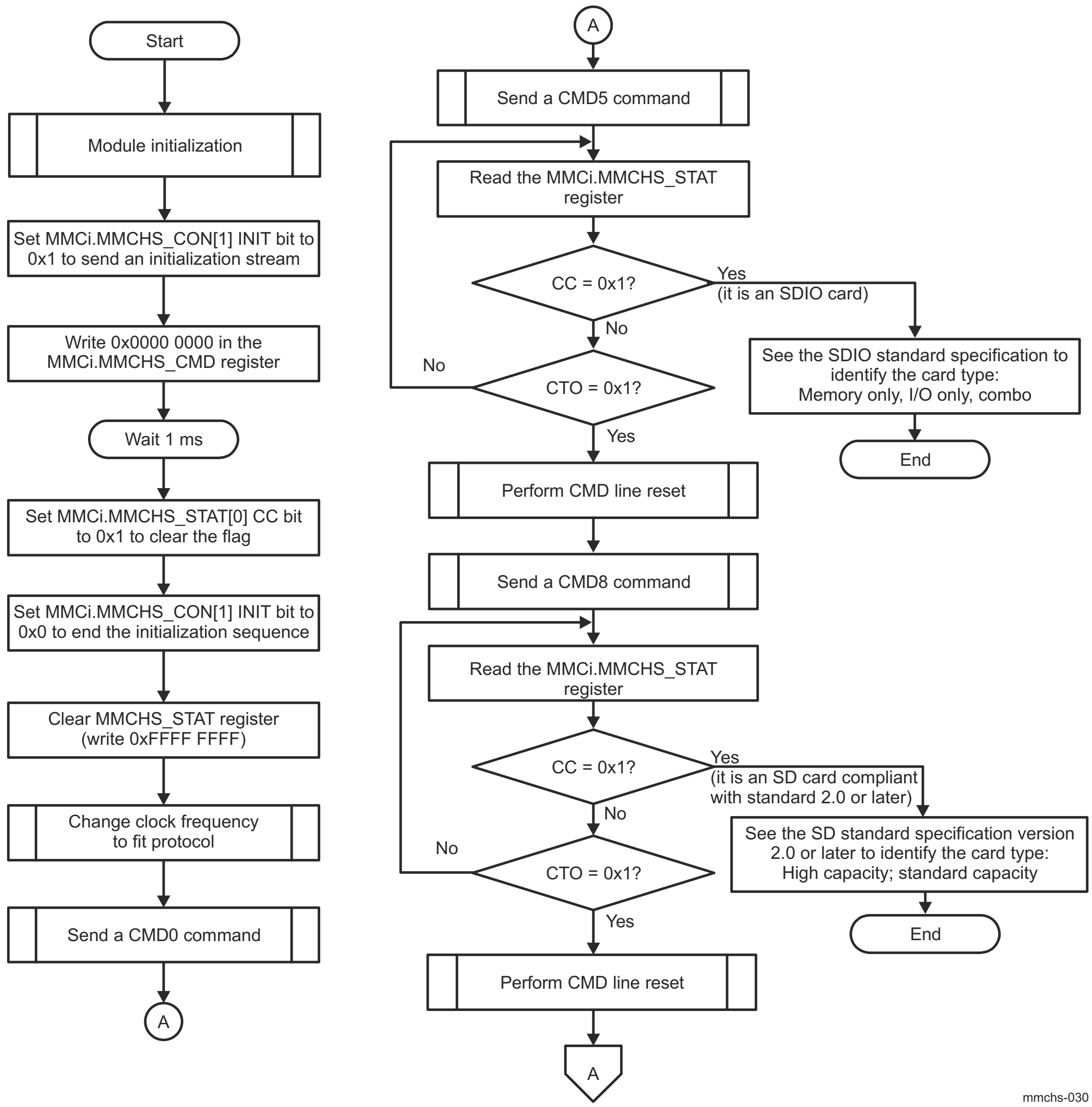


Figure 25-36. eMMC/SD/SDIO Controller Card Identification and Selection – Part 1

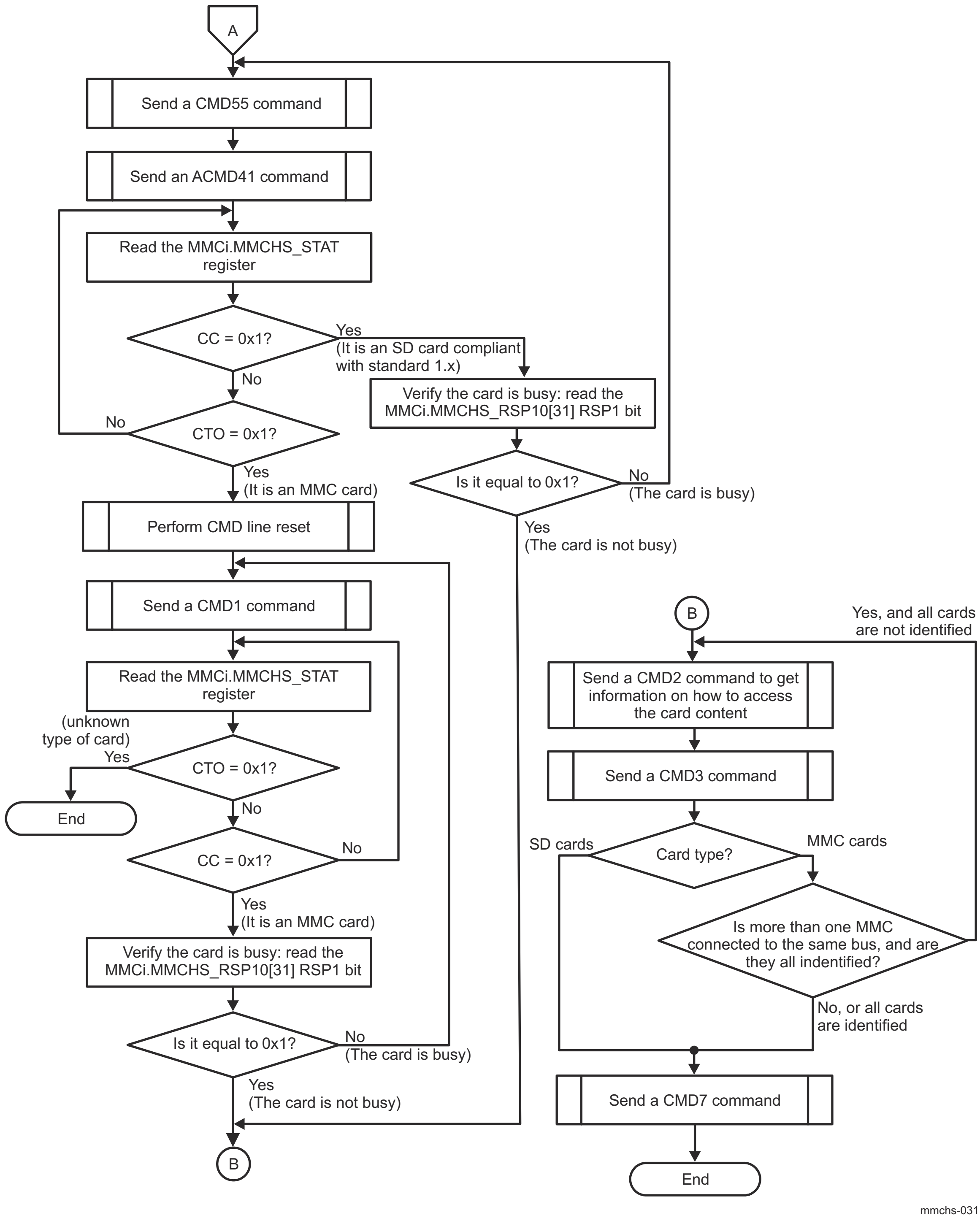


Figure 25-37. eMMC/SD/SDIO Controller Card Identification and Selection – Part 2

Table 25-28. Register Call Summary for Main Sequence – Card Identification and Selection

Register Name	Register Name	Register Name
MMCHS_CON	MMCHS_CMD	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_RSP10	

Table 25-29 lists the subprocess call summary.

Table 25-29. Subprocess Call Summary for Main Sequence – Card Identification and Selection

Subprocess Name	Cross-Reference
Initialize module.	See Section 25.5.1.1.2 , *eMMC/SD/SDIO Host Controller Initialization Flow*.
Change clock frequency to fit protocol.	See Section 25.5.1.2.1.7.2, *MMCHS Clock Frequency Change*.
Send a command.	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.
Perform CMD line reset.	See Section 25.5.1.2.1.1.1, *CMD Line Reset Procedure*.

25.5.1.2.1.1.1 CMD Line Reset Procedure

Table 25-30lists the CML line reset.

Table 25-30. CMD Line Reset

Step	Access Type	Register/Bit Field/Programming Model	Value
Initiate CMD line reset.	W	MMCi.MMCHS_SYSCTL[25] SRC	0x1
Poll the SRC bit until it is set to 0x1.	R	MMCi.MMCHS_SYSCTL[25] SRC	= 0x1
Wait until the SRC bit returns to 0x0 (reset procedure is completed).	R	MMCi.MMCHS_SYSCTL[25] SRC	= 0x0

25.5.1.2.1.2 Read/Write Transfer Flow in DMA Mode With Interrupt

Figure 25-38 shows the read and write protocol in DMA slave mode with interrupt signaling.

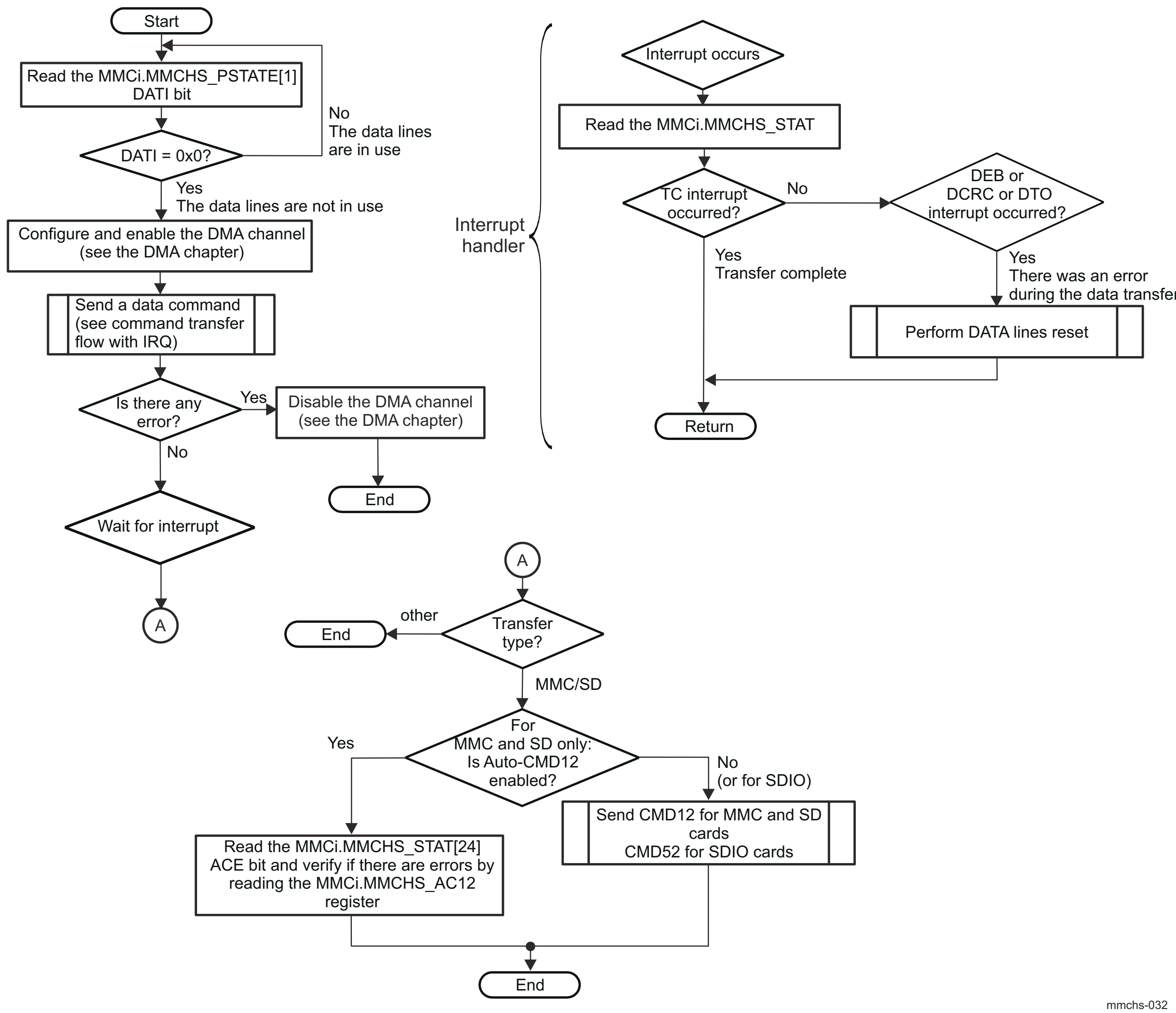


Figure 25-38. eMMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Slave Mode With interrupt

Table 25-31. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With interrupt

Register Name	Register Name
MMCHS_PSTATE	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_CMD

Table 25-32. Subprocess Call Summary for Main Sequence – eMMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Mode With Interrupt

Subprocess Name	Cross-Reference
Send a data command.	See Figure 25-45.
Perform DATA lines reset.	See Section 25.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

25.5.1.2.1.2.1 DATA Lines Reset Procedure

Table 25-33 describes the DATA lines reset.

Table 25-33. DATA Lines Reset

Step	Access Type	Register/Bit Field/Programming Model	Value
Initiate DATA lines reset.	W	MMCi.MMCHS_SYSCTL[26] SRD	0x1
Poll the SRD bit until it is set to 0x1.	R	MMCi.MMCHS_SYSCTL[26] SRD	= 0x1
Wait until the SRD bit returns to 0x0 (reset procedure is complete).	R	MMCi.MMCHS_SYSCTL[26] SRD	= 0x0

25.5.1.2.1.3 Read/Write Transfer Flow in DMA Mode With Polling

Figure 25-39 shows the read and write protocol in DMA mode.

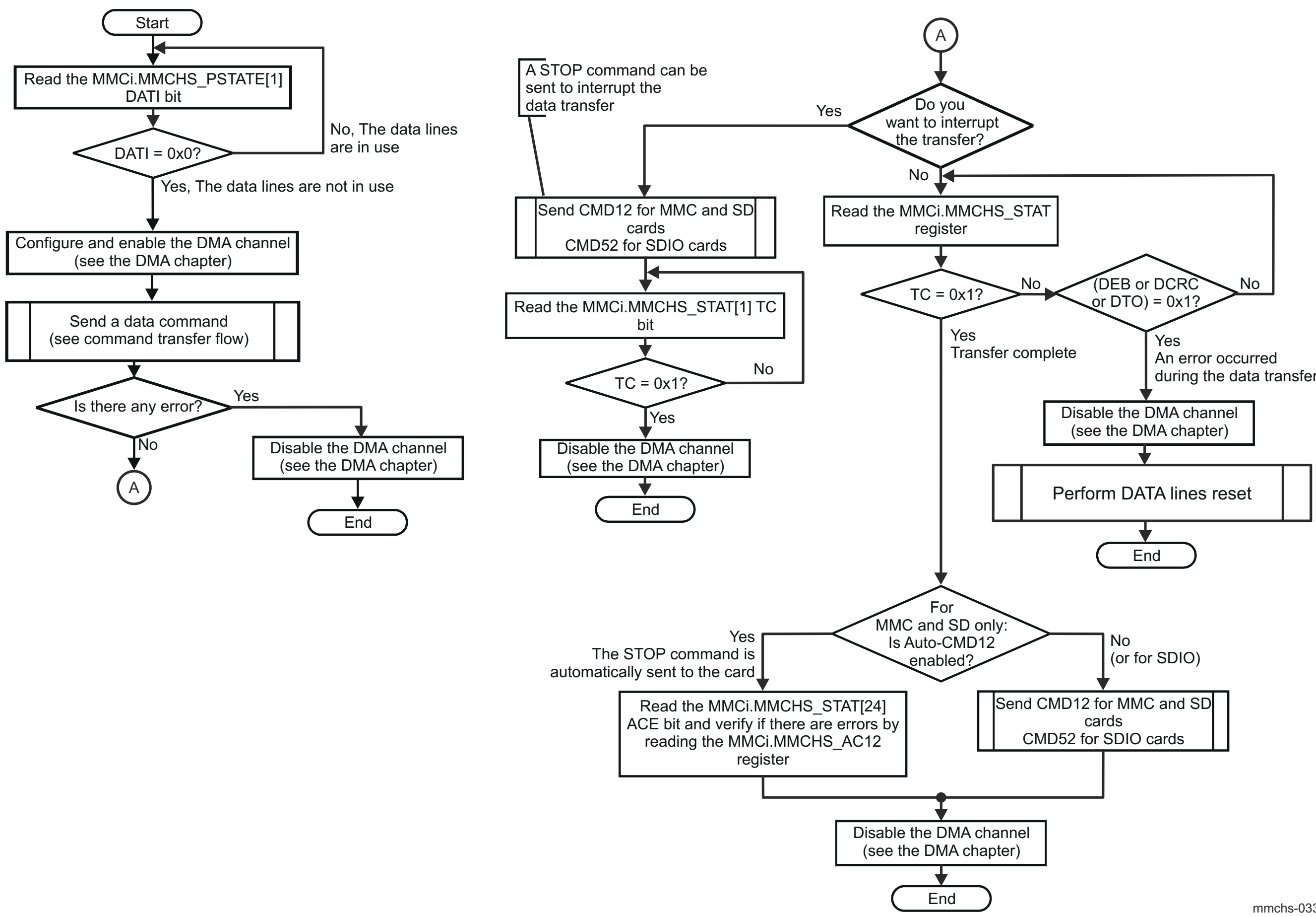


Figure 25-39. eMMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Mode With Polling

Table 25-34. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_STAT	MMCHS_SYSCTL
MMCHS_CMD	MMCHS_AC12	

Table 25-35. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling

Subprocess Name	Cross-Reference
Send command.	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.
Perform DATA lines reset.	See Section 25.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

25.5.1.2.1.4 Read/Write Transfer Flow Without DMA With Polling

Figure 25-40 shows a read/write transfer without using the DMA and with polling.

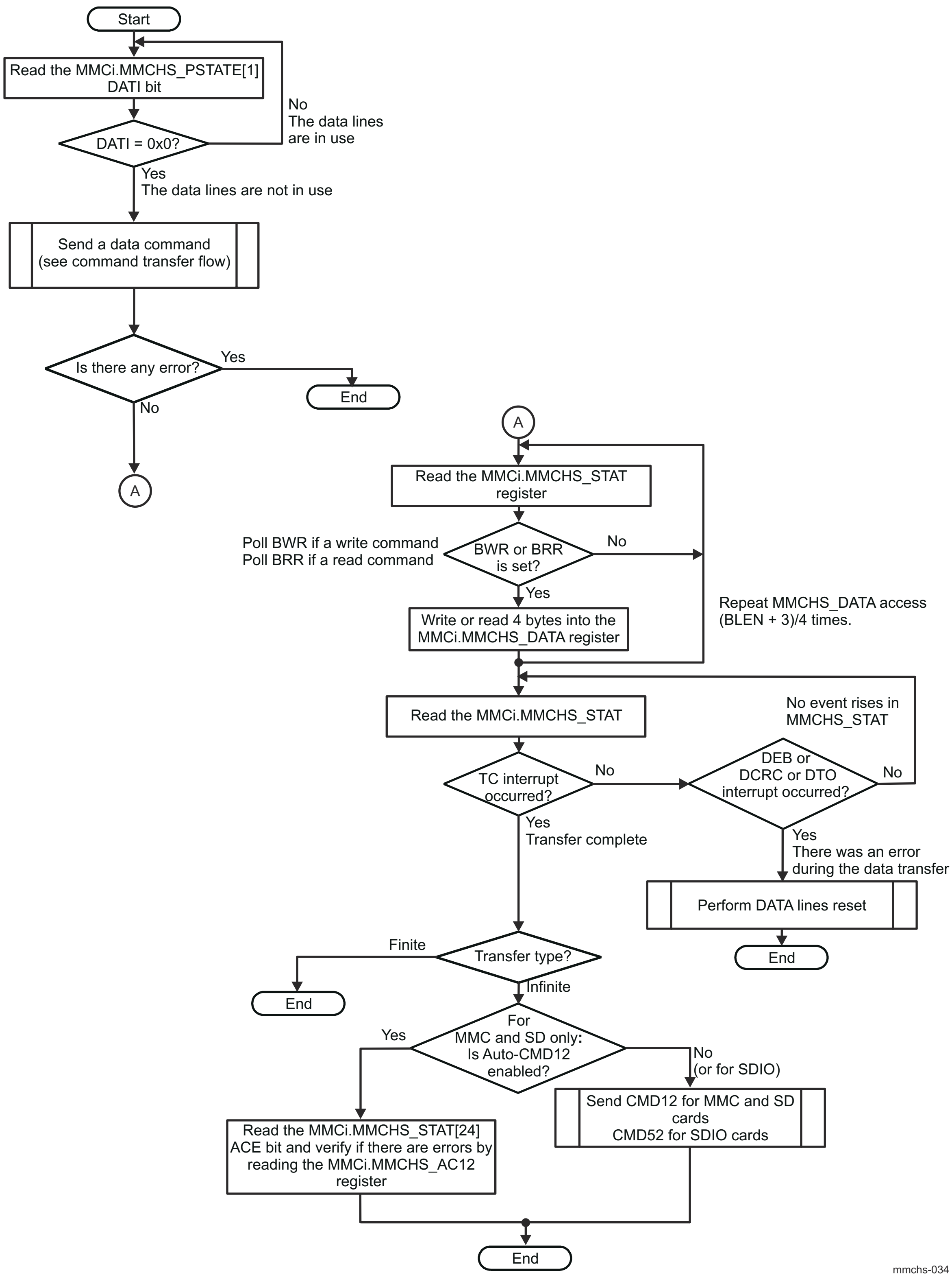


Figure 25-40. eMMC/SD/SDIO Controller Read/Write Transfer Flow Without DMA and With Polling

Table 25-36. Register Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_DATA	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_CMD	MMCHS_AC12

Table 25-37. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling

Subprocess Name	Cross-Reference
Send data command.	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.
Perform DATA lines reset.	See Section 25.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

25.5.1.2.1.5 Read/Write Transfer Flow in CE-ATA Mode

Figure 25-41 shows the read and write CE-ATA protocol when in polling mode.

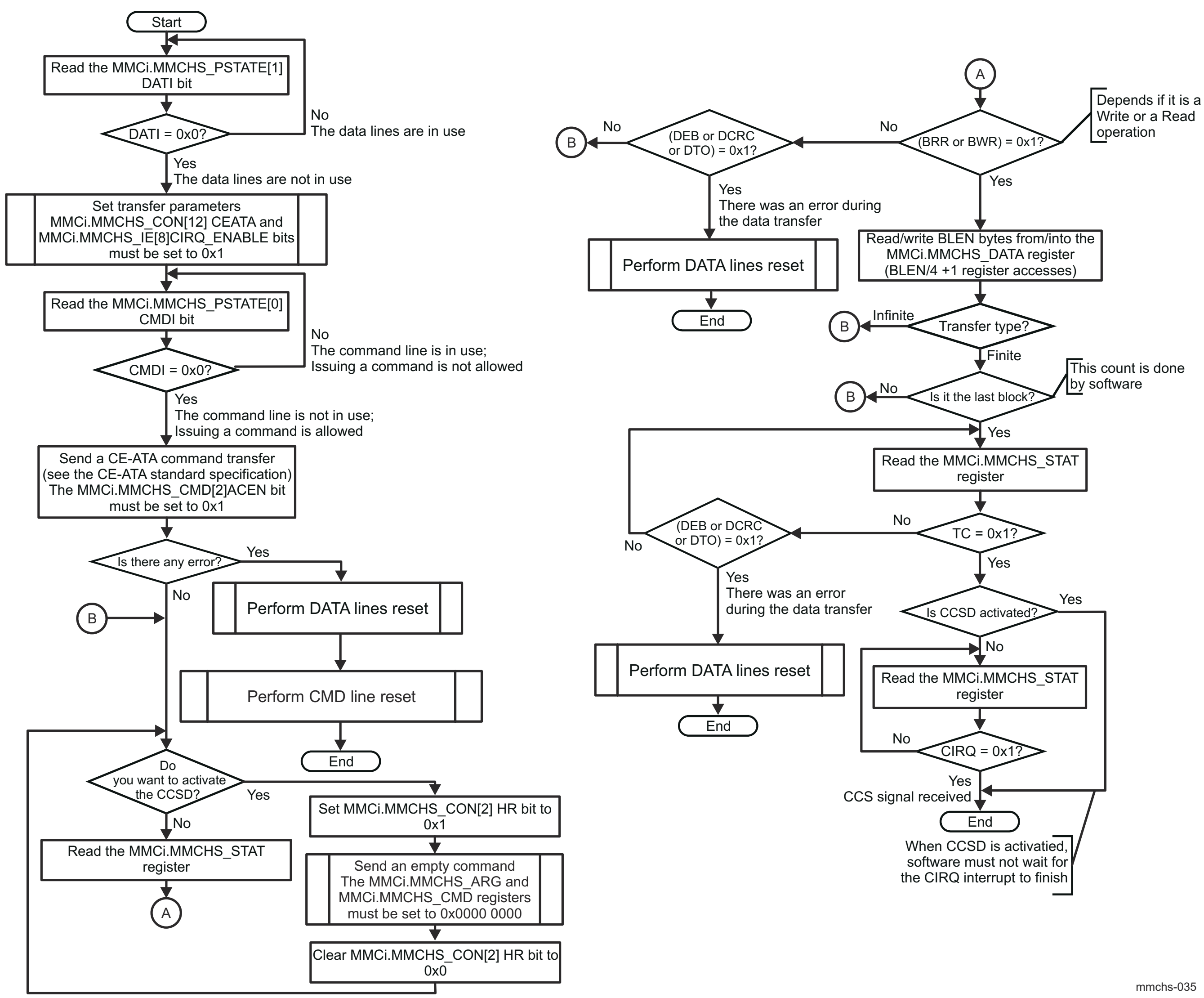


Figure 25-41. eMMC/SD/SDIO Controller Read/Write in CE-ATA Mode

Table 25-38. Register Call Summary for Main Sequence – Read/Write in CE-ATA Mode

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_IE
MMCHS_CMD	MMCHS_STAT	
MMCHS_ARG	MMCHS_SYSCTL	

Table 25-39. Subprocess Call Summary for Main Sequence – Read/Write in CE-ATA Mode

Subprocess Name	Cross-Reference
Perform CMD line reset.	See Section 25.5.1.2.1.1.1, *CMD Line Reset Procedure*.
Perform DATA lines reset.	See Section 25.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

CAUTION

CE-ATA protocol is supported only by MMC cards.

In CE-ATA mode, issuing a command during the transfer (except a CCSD command) is not allowed.

In CE-ATA mode, infinite transfers are not allowed; only finite transfers are permitted.

25.5.1.2.1.6 Suspend-Resume Flow

The suspend-and-resume feature is supported only by SDIO cards.

25.5.1.2.1.6.1 Suspend Flow

Figure 25-42 shows the suspend flow for SDIO cards.

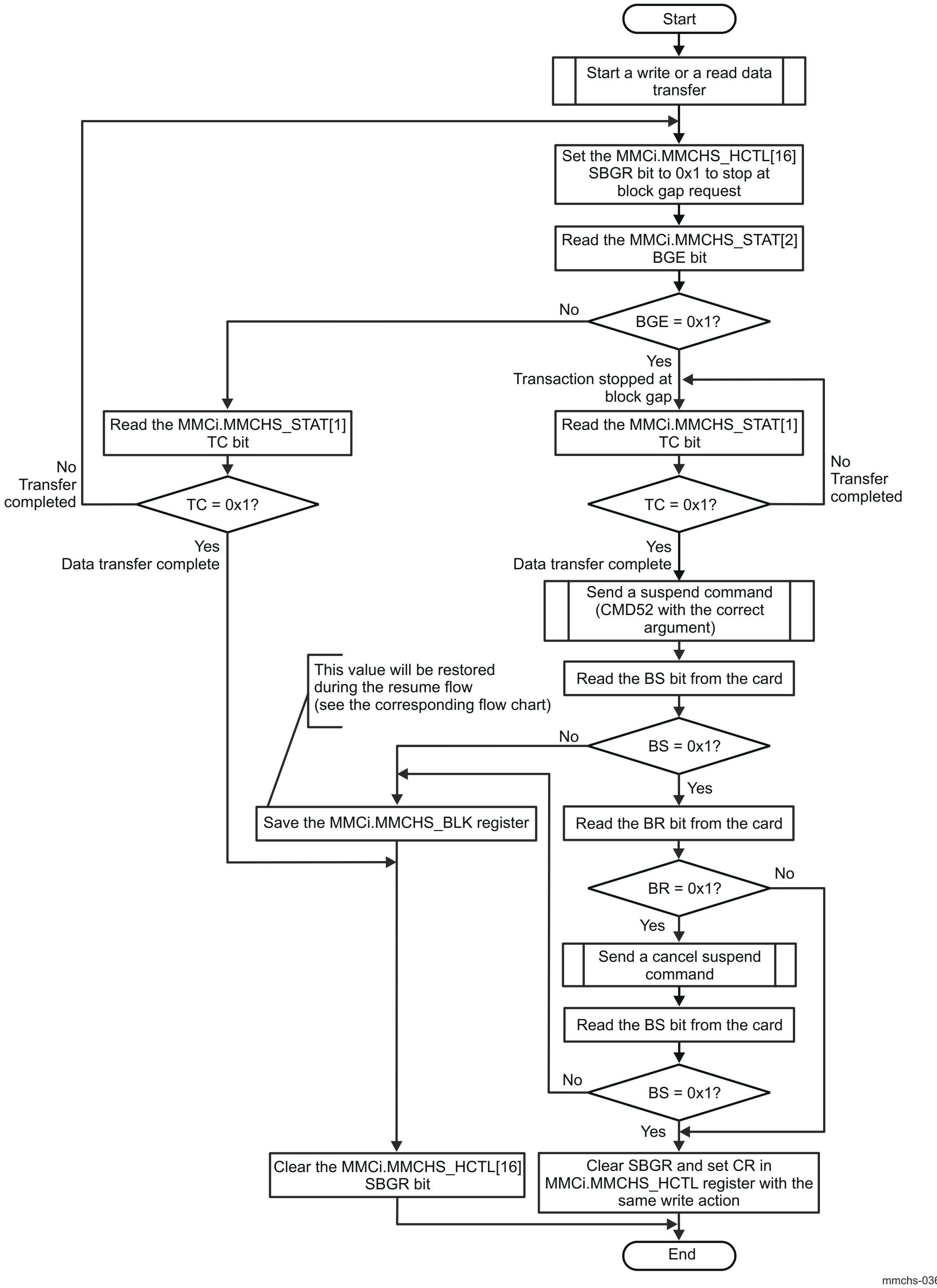


Figure 25-42. eMMC/SD/SDIO Controller Suspend Flow

Table 25-40. Register Call Summary for Main Sequence – Suspend Flow

Register Name	Register Name	Register Name
MMCHS_HCTL	MMCHS_STAT	MMCHS_BLK

Table 25-41. Subprocess Call Summary for Main Sequence – Suspend Flow

Subprocess Name	Cross-Reference
Start a write or a read data transfer.	See Section 25.5.1.2.1, *Basic Operations for eMMC/SD/SDIO Host Controller*.
Send a suspend command (CMD52 with the correct argument).	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.
Send a cancel suspend command.	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.

25.5.1.2.1.6.2 Resume Flow

Figure 25-43 shows the resume flow for SDIO cards.

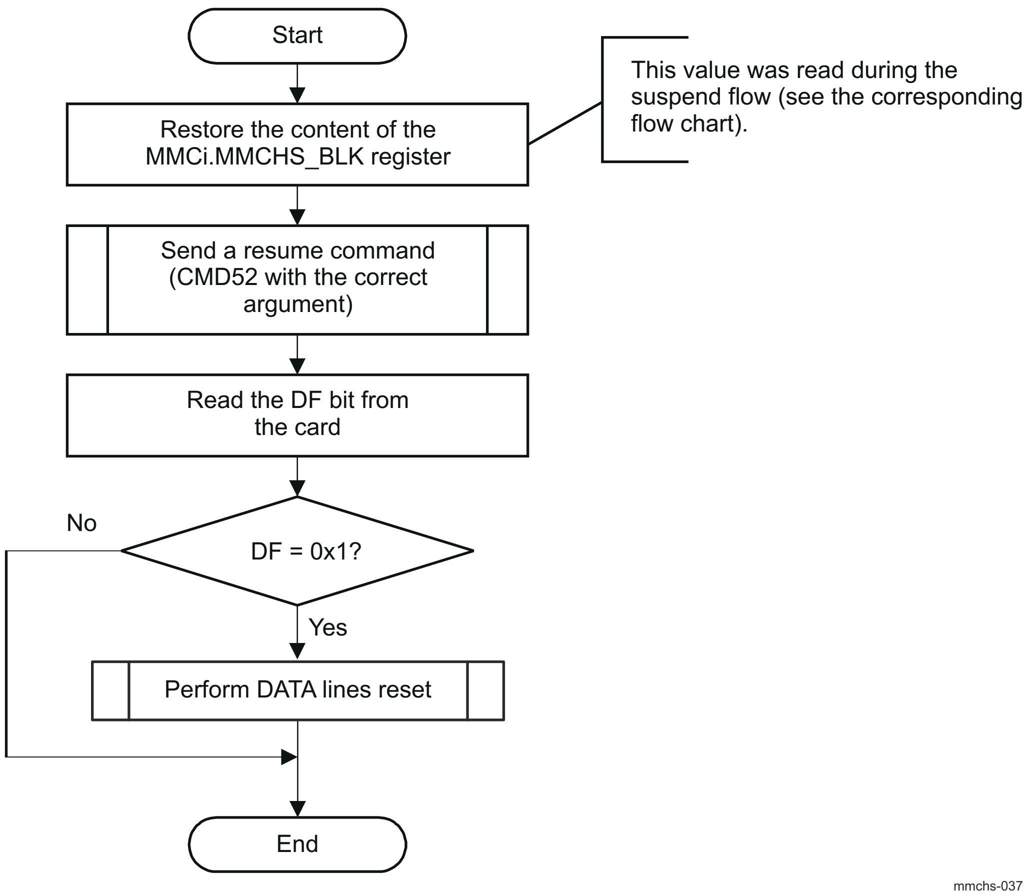


Figure 25-43. eMMC/SD/SDIO Controller Resume Flow

Table 25-42. Register Call Summary for Main Sequence - Resume Flow

Register Name	Register Name
MMCHS_BLK	MMCHS_SYSCTL

Table 25-43. Subprocess Call Summary for Main Sequence - Resume Flow

Subprocess Name	Cross-Reference
Send a resume command (CMD52 with the correct argument).	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.
Perform DATA lines reset.	See Section 25.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

25.5.1.2.1.7 Basic Operations – Steps Detailed

25.5.1.2.1.7.1 Command Transfer Flow

Figure 25-44 shows how to send a command to the card using polling instead of interrupts for event signaling.

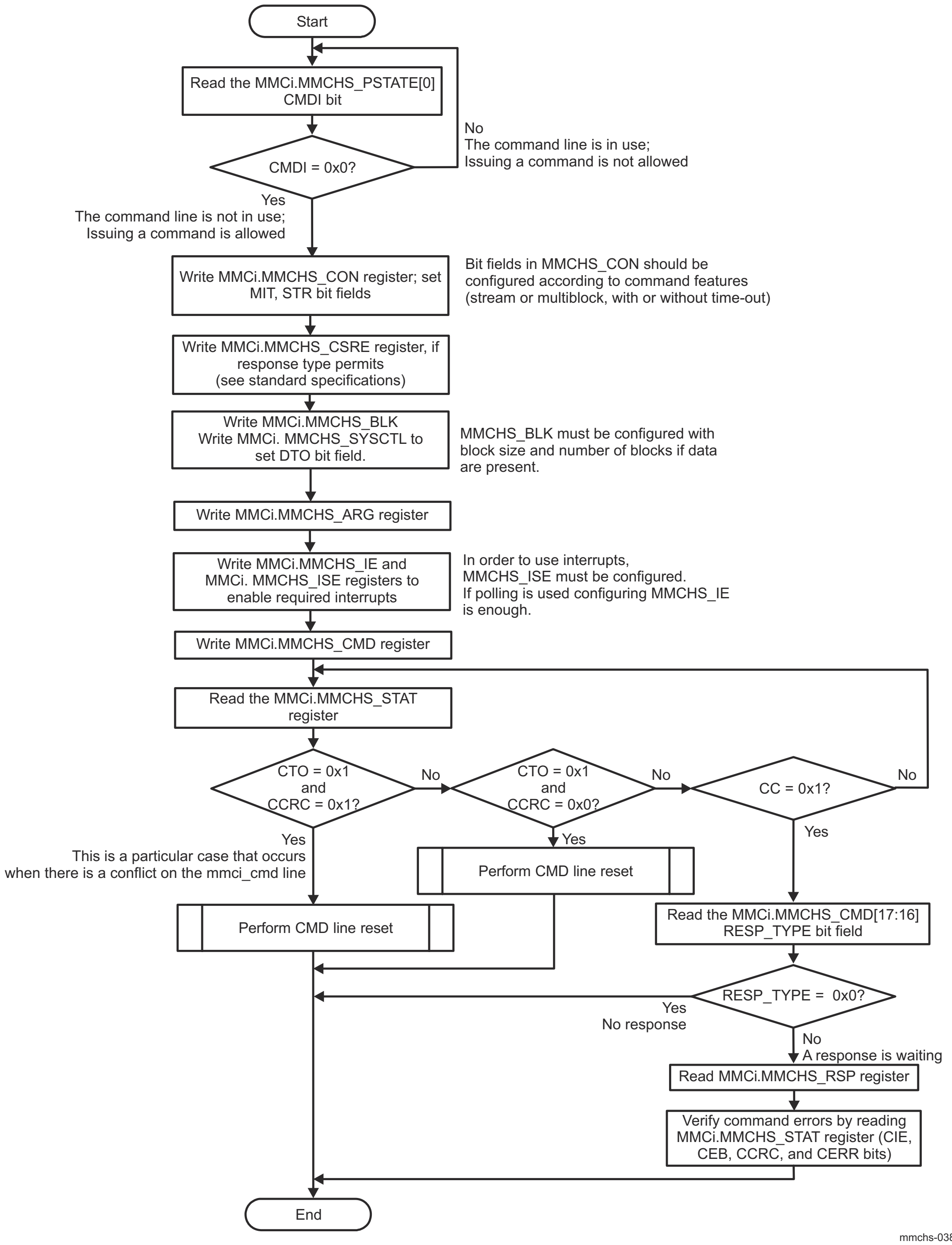


Figure 25-44. eMMC/SD/SDIO Controller Command Transfer Flow With Polling

Table 25-44. Register Call Summary for Main Sequence – Command Transfer Flow With Polling

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_CSRE
MMCHS_STAT	MMCHS_BLK	MMCHS_SYSCTL
MMCHS_ARG	MMCHS_IE	MMCHS_CMD
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76		

Table 25-45. Subprocess Call Summary for Main Sequence – Command Transfer Flow With Polling

Subprocess Name	Cross-Reference
Perform CMD line reset.	See Section 25.5.1.2.1.1.1, *CMD Line Reset Procedure*.

Figure 25-45 shows how to send a command to the card using interrupts for event signaling.

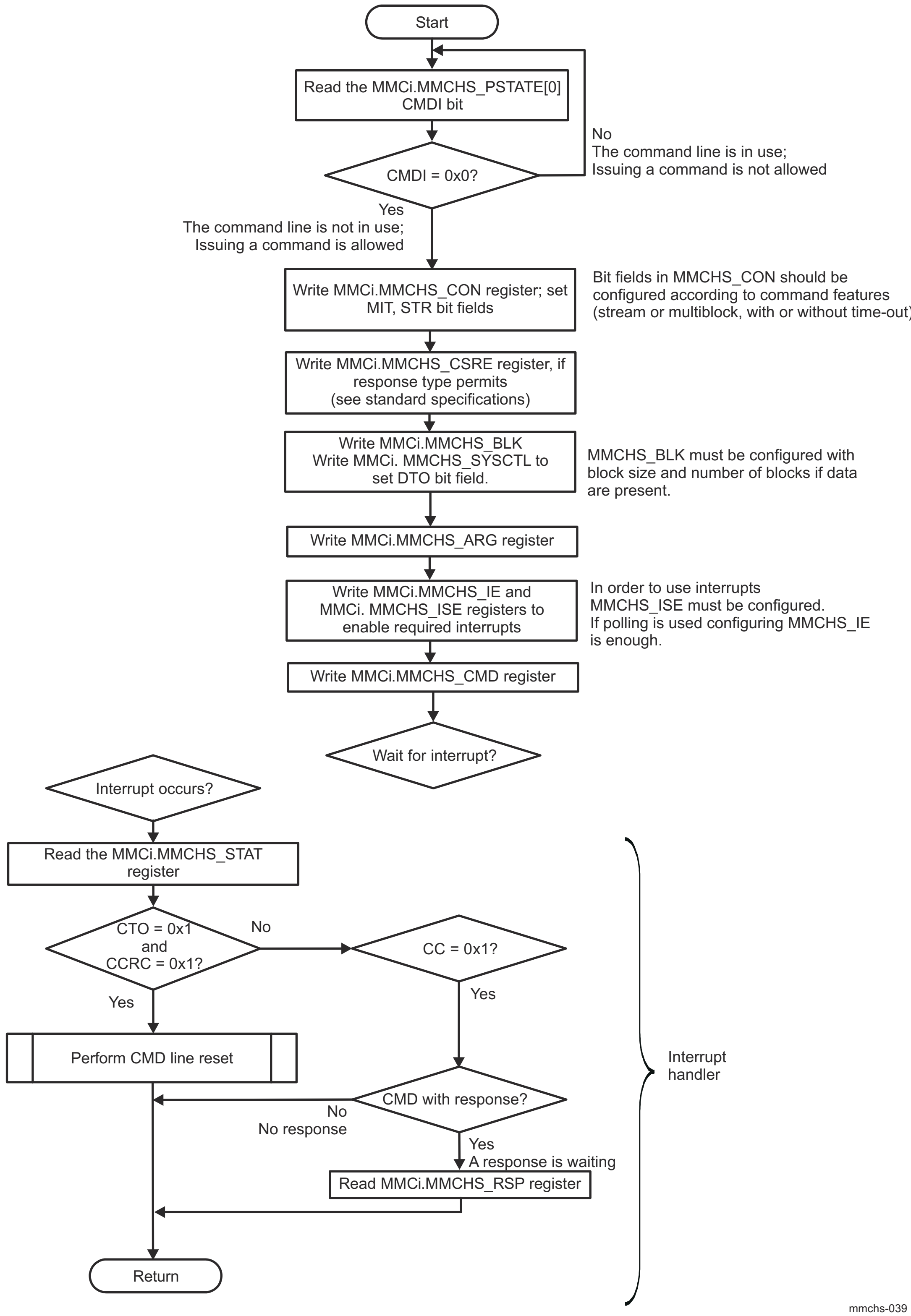


Figure 25-45. eMMC/SD/SDIO Controller Command Transfer Flow With interrupts

Table 25-46. Register Call Summary for Main Sequence – Command Transfer Flow With Interrupts

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_CSRE

Table 25-46. Register Call Summary for Main Sequence – Command Transfer Flow With Interrupts (continued)

Register Name	Register Name	Register Name
MMCHS_STAT	MMCHS_BLK	MMCHS_SYSCTL
MMCHS_ARG	MMCHS_IE	MMCi.MMCHS_ISE
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76	MMCHS_CMD	

Table 25-47. Subprocess Call Summary for Main Sequence – Command Transfer Flow With Interrupts

Subprocess Name	Cross-Reference
Perform CMD line reset.	See Section 25.5.1.2.1.1.1, *CMD Line Reset Procedure*.

25.5.1.2.1.7.2 MMCHS Clock Frequency Change

Figure 25-46 shows the different steps that allow changing the eMMC/SD/SDIO output clock frequency.

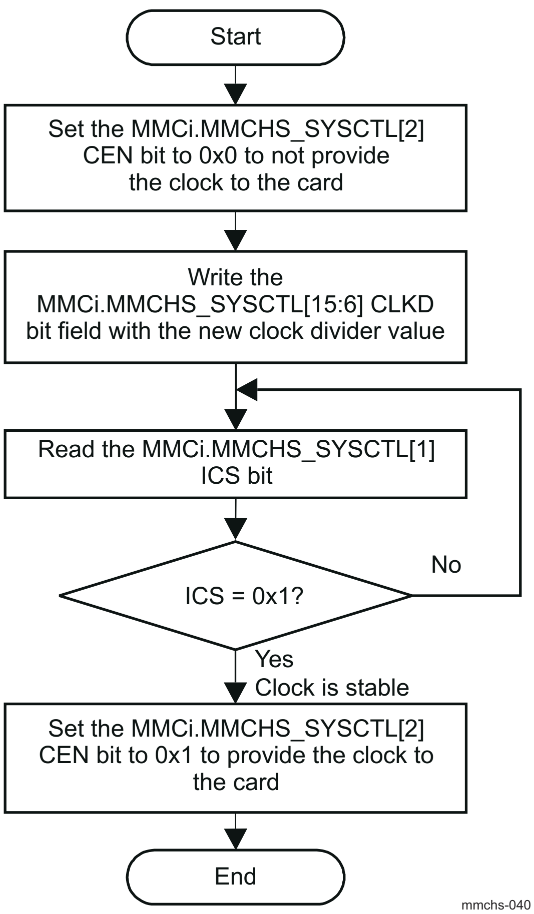


Figure 25-46. eMMC/SD/SDIO Controller Clock Frequency Change Flow

Table 25-48. Register Call Summary for Main Sequence – Clock Frequency Change Flow

Register Name
MMCHS_SYSCTL

25.5.1.2.1.7.3 Bus Width Selection

Figure 25-47 shows the different steps that allow changing the eMMC/SD/SDIO bus width.

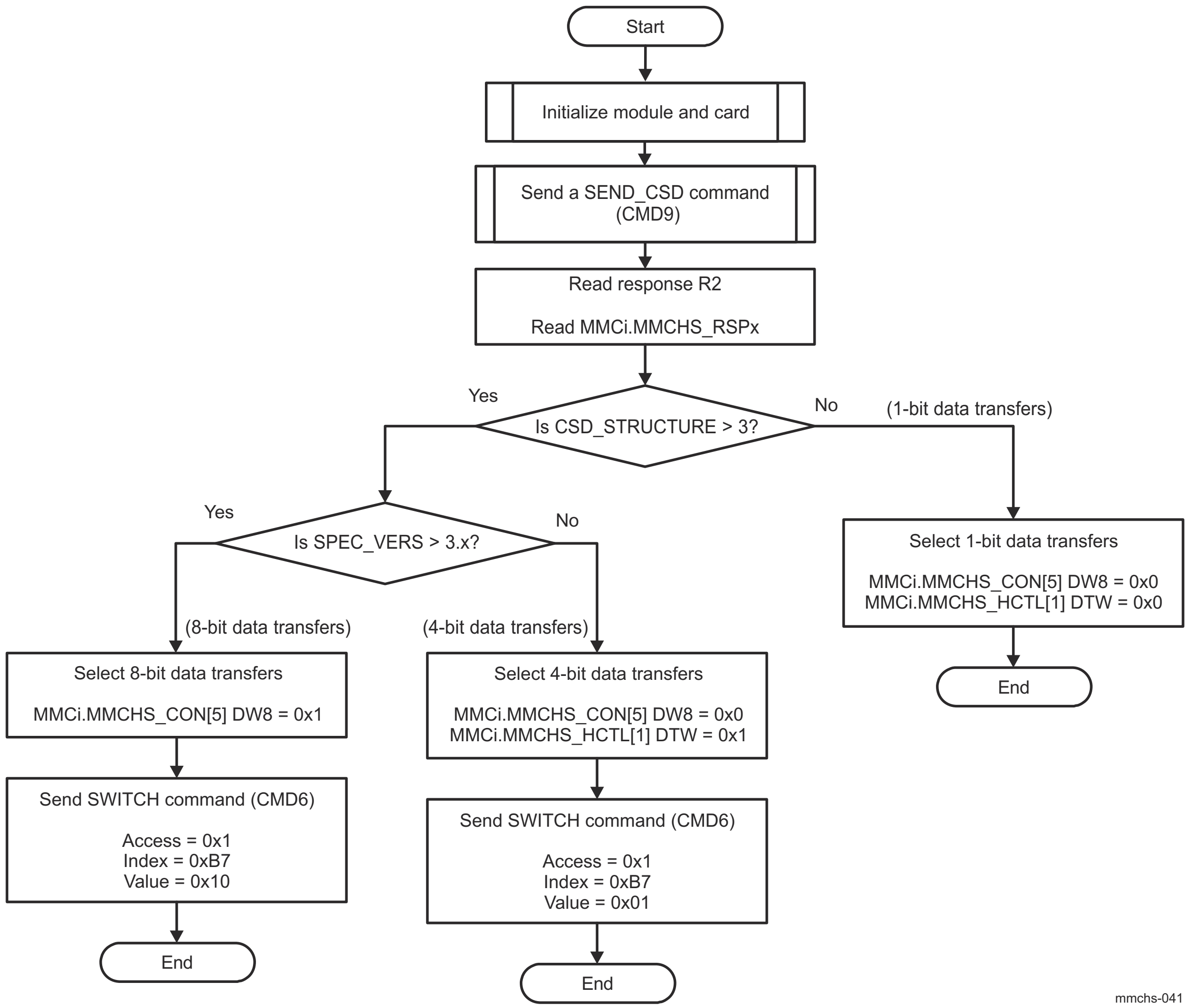


Figure 25-47. eMMC/SD/SDIO Controller Bus Width Configuration Flow

Table 25-49. Register Call Summary for Main Sequence – Bus Width Configuration Flow

Register Name	Register Name	Register Name
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76	MMCHS_CON	MMCHS_HCTL

Table 25-50. Subprocess Call Summary for Main Sequence – Bus Width Configuration Flow

Subprocess Name	Cross-Reference
Initialize module and card.	See Section 25.5.1.1.2, *eMMC/SD/SDIO Host Controller Initialization Flow*. See Section 25.5.1.2.1.1, *Card Detection, Identification, and Selection*.
Send a SEND_CSD command (CMD9).	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.
Send SWITCH command (CMD6).	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.

25.5.1.2.2 Bus Voltage Selection

The eMMC/SD/SDIO1 controller can operate with two types of card voltages: 1.8 V and 3.3 V. For this reason, dual voltage pads are implemented on this interface. For technological concerns, those pads must have an internal bias voltage reference to operate. The PBIAS module supplies this bias voltage, depending on the settings of the CTRL_CORE_CONTROL_PBIAS register.

For more information about the PBIAS cell, see *PBIAS Cell And MMC1 I/O Cells Control Registers*, in *Control Module*.

Figure 25-48 shows how to configure the MMC1 controller to fit with power switching sequence.

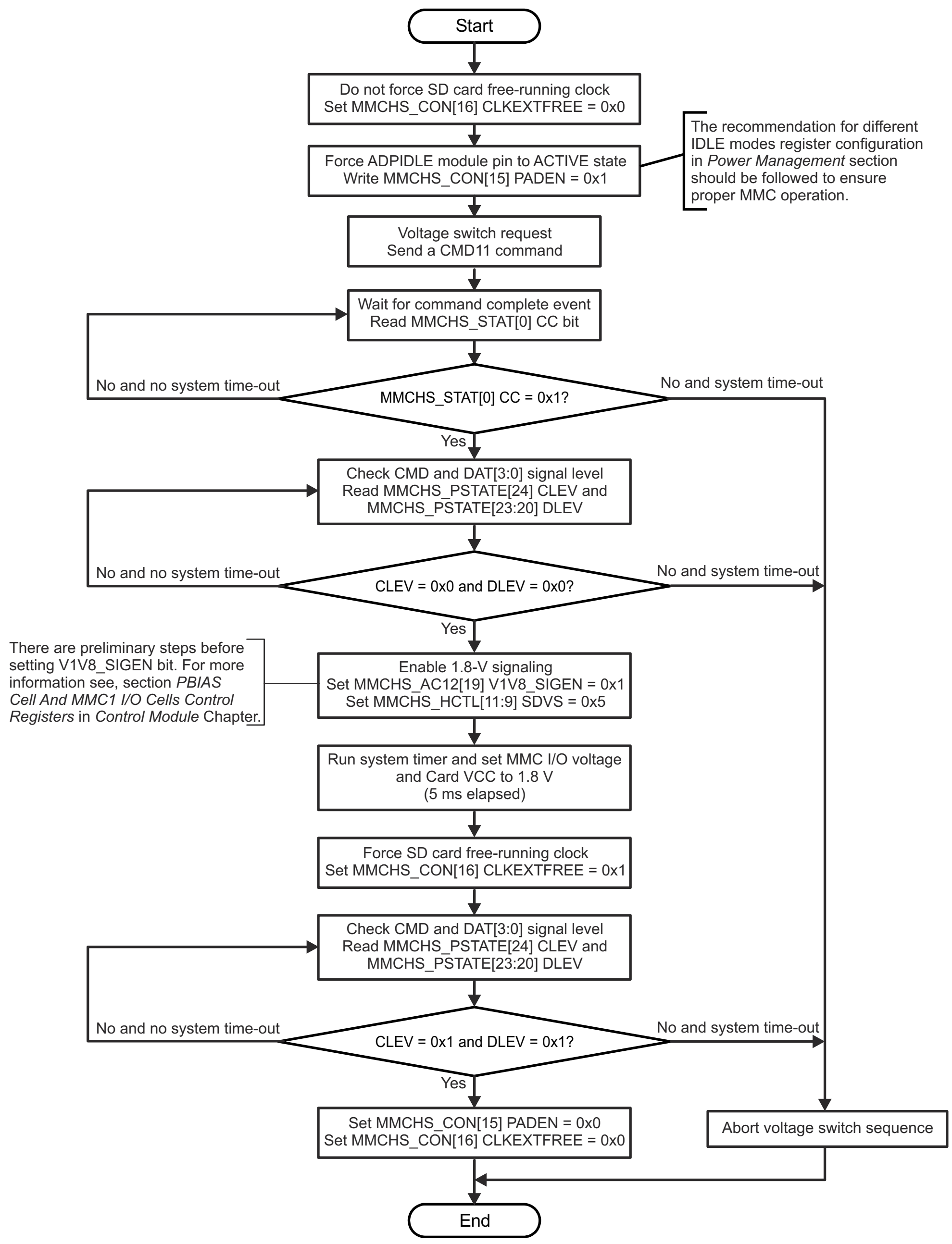


Figure 25-48. eMMC/SD/SDIO Power Switching Procedure

Table 25-51. Register Call Summary for Main Sequence – Power Switching Procedure

Register Name	Register Name
MMCHS_STAT	MMCHS_PSTATE
MMCHS_CMD	MMCHS_CON

Table 25-52. Subprocess Call Summary for Main Sequence – Power Switching Procedure

Subprocess Name	Cross-Reference
Send a READ_DAT_UNTIL_STOP command (CMD11).	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.

25.5.1.2.3 Boot Mode Configuration

The following sections describe the two possible ways to issue a boot command: issue a CMD0 or drive the CMD line to 0 during the whole boot phase.

25.5.1.2.3.1 Boot Using CMD0

Figure 25-49 shows the necessary steps to configure the controller boot mode using CMD0.

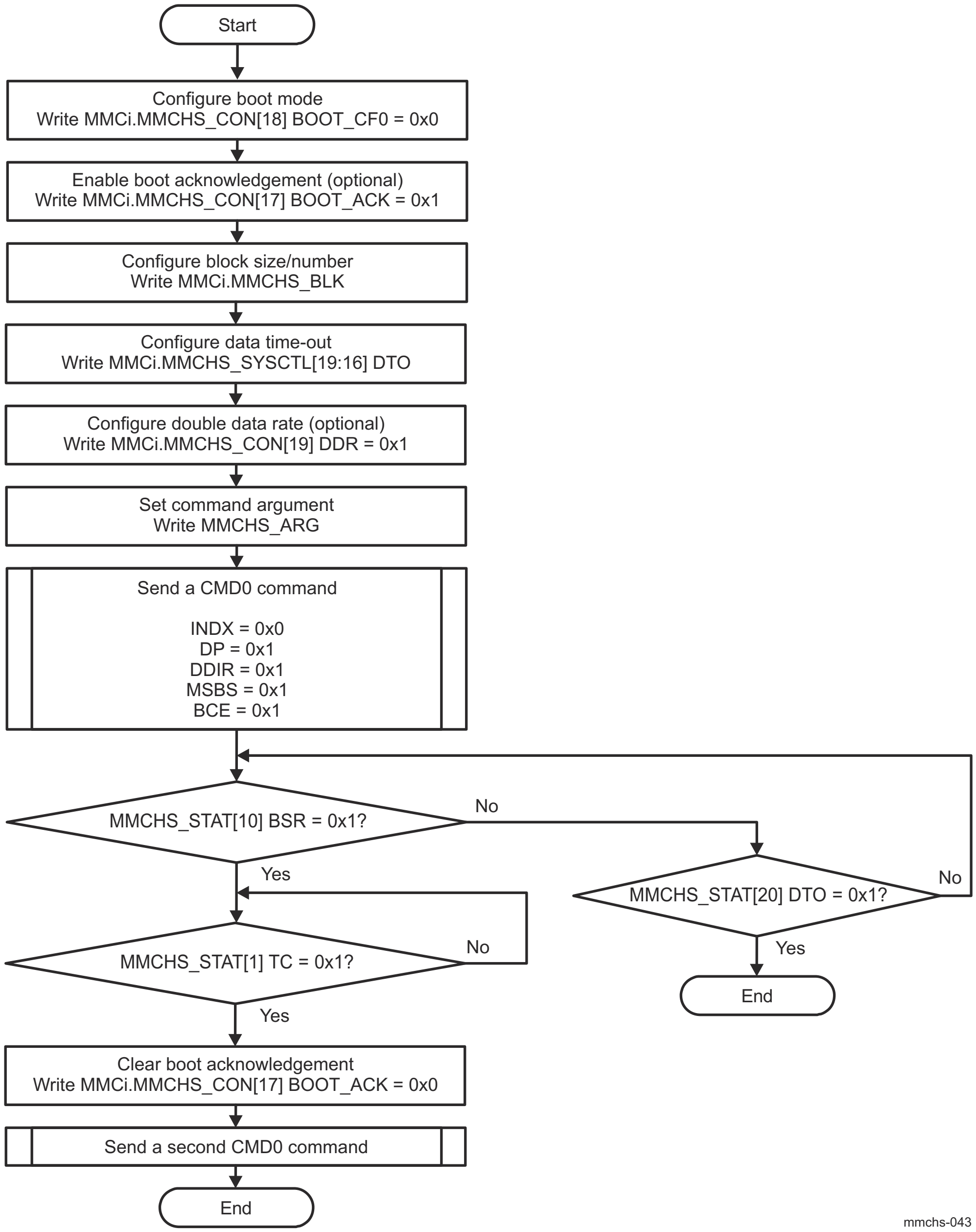


Figure 25-49. eMMC/SD/SDIO Controller Boot Using CMD0

To abort a boot sequence, the system must issue a CMD0 with the MMCHS_CMD[23:22] CMD_TYPE bit field set to 0x3 (the MMCHS_CON[17] BOOT_ACK bit previously cleared to 0x0) during the transfer to abort the transfer and enable the card to exit from boot state.

Table 25-53. Register Call Summary for Main Sequence – Boot Using CMD0

Register Name	Register Name	Register Name
MMCHS_CON	MMCHS_BLK	MMCHS_SYSCTL
MMCHS_ARG	MMCHS_STAT	

Table 25-54. Subprocess Call Summary for Main Sequence – Boot Using CMD0

Subprocess Name	Cross-Reference
Send a CMD0 command.	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.

25.5.1.2.3.2 Boot With CMD Line Tied to 0

Figure 25-50 shows the necessary steps to configure the controller in this mode; the driver must follow this sequence.

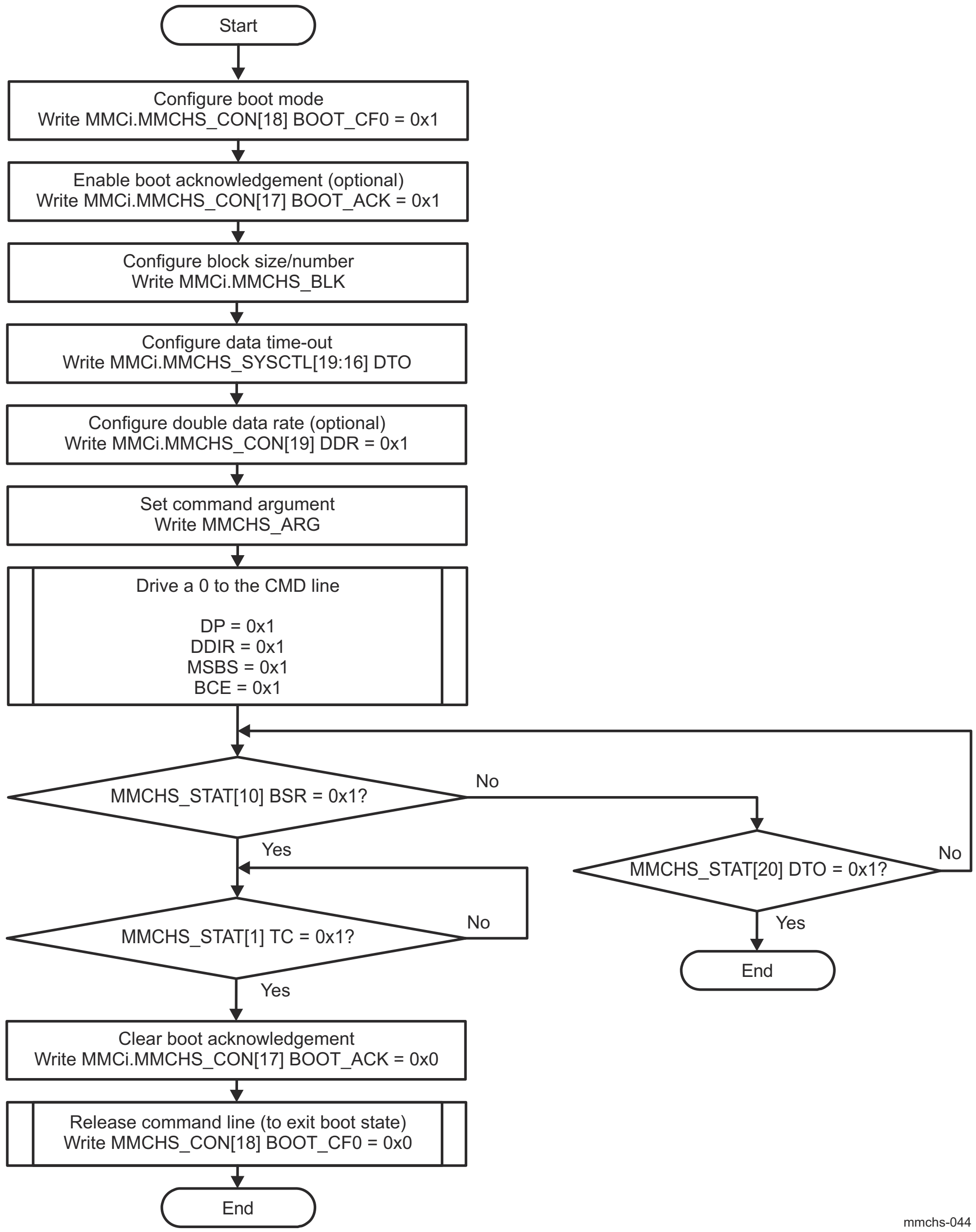


Figure 25-50. eMMC/SD/SDIO Controller Boot With CMD Line Tied to 0

To abort the boot sequence, the system must clear the MMCHS_CON[18] BOOT_CF0 bit to 0x0 during the transfer to abort the transfer and enable the card to exit from boot state.

Table 25-55. Register Call Summary for Main Sequence – Boot Using CMD0

Register Name	Register Name	Register Name
MMCHS_CON	MMCHS_BLK	MMCHS_SYSCTL
MMCHS_ARG	MMCHS_STAT	MMCHS_CMD

Table 25-56. Subprocess Call Summary for Main Sequence – Boot Using CMD0

Subprocess Name	Cross-Reference
Send a CMD0 command.	See Section 25.5.1.2.1.7.1, *Command Transfer Flow*.

25.5.1.2.4 SDR104/HS200 DLL Tuning Procedure

Figure 25-51 shows the DLL tuning procedure when SDR104 and HS200 modes are used.

Note

The BRR interrupt has to be enabled during tuning for functionality to succeed. The other error flags in MMCHS_IE can also be enabled for debug purposes. The MMCHS_SYSCTL[25] SRC bit can be set for command related interrupts, but caution has to be taken such that the MMCHS_SYSCTL[26] SRD bit is not set even when DAT related errors occurred, until after tuning completion.

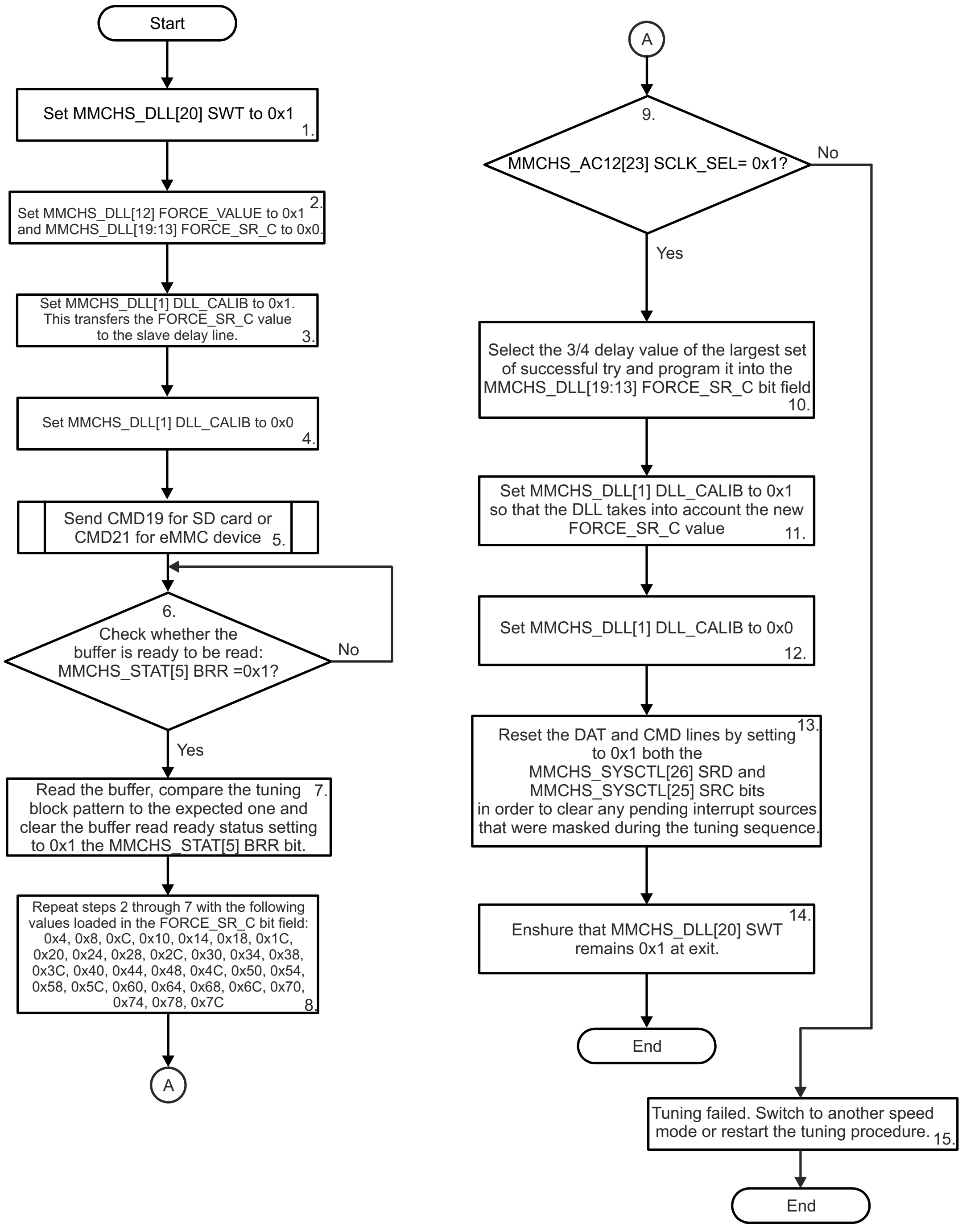


Figure 25-51. SDR104/HS200 DLL Tuning Procedure

25.6 eMMC/SD/SDIO Register Manual

25.6.1 eMMC/SD/SDIO Instance Summary

Table 25-57 lists the eMMC/SD/SDIO instances.

Table 25-57. eMMC/SD/SDIO Instance Summary

Module Name	Module Base Address	Size
MMC1	0x4809 C000	4 KiB
MMC2	0x480B 4000	4 KiB
MMC3	0x480A D000	4 KiB
MMC4	0x480D 1000	4 KiB

25.6.2 eMMC/SD/SDIO Registers

25.6.2.1 eMMC/SD/SDIO Register Summary

Table 25-58 lists the eMMC/SD/SDIO registers.

Table 25-58. MMC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MMC1 Physical Address	MMC2 Physical Address
MMCHS_HL_REV	R	32	0x0000 0000	0x4809 C000	0x480B 4000
MMCHS_HL_HWINFO	R	32	0x0000 0004	0x4809 C004	0x480B 4004
MMCHS_HL_SYSCONFIG	RW	32	0x0000 0010	0x4809 C010	0x480B 4010
MMCHS_SYSCONFIG	RW	32	0x0000 0110	0x4809 C110	0x480B 4110
MMCHS_SYSSTATUS	R	32	0x0000 0114	0x4809 C114	0x480B 4114
MMCHS_CSRE	RW	32	0x0000 0124	0x4809 C124	0x480B 4124
MMCHS_SYSTEST	RW	32	0x0000 0128	0x4809 C128	0x480B 4128
MMCHS_CON	RW	32	0x0000 012C	0x4809 C12C	0x480B 412C
MMCHS_PWCNT	RW	32	0x0000 0130	0x4809 C130	0x480B 4130
MMCHS_DLL	RW	32	0x0000 0134	0x4809 C134	0x480B 4134
MMCHS_SDMASA	RW	32	0x0000 0200	0x4809 C200	0x480B 4200
MMCHS_BLK	RW	32	0x0000 0204	0x4809 C204	0x480B 4204
MMCHS_ARG	RW	32	0x0000 0208	0x4809 C208	0x480B 4208
MMCHS_CMD	RW	32	0x0000 020C	0x4809 C20C	0x480B 420C
MMCHS_RSP10	R	32	0x0000 0210	0x4809 C210	0x480B 4210
MMCHS_RSP32	R	32	0x0000 0214	0x4809 C214	0x480B 4214
MMCHS_RSP54	R	32	0x0000 0218	0x4809 C218	0x480B 4218
MMCHS_RSP76	R	32	0x0000 021C	0x4809 C21C	0x480B 421C
MMCHS_DATA	RW	32	0x0000 0220	0x4809 C220	0x480B 4220
MMCHS_PSTATE	R	32	0x0000 0224	0x4809 C224	0x480B 4224
MMCHS_HCTL	RW	32	0x0000 0228	0x4809 C228	0x480B 4228
MMCHS_SYSCTL	RW	32	0x0000 022C	0x4809 C22C	0x480B 422C
MMCHS_STAT	RW	32	0x0000 0230	0x4809 C230	0x480B 4230
MMCHS_IE	RW	32	0x0000 0234	0x4809 C234	0x480B 4234
MMCHS_ISE	RW	32	0x0000 0238	0x4809 C238	0x480B 4238
MMCHS_AC12	RW	32	0x0000 023C	0x4809 C23C	0x480B 423C
MMCHS_CAPA	RW	32	0x0000 0240	0x4809 C240	0x480B 4240
MMCHS_CAPA2	R	32	0x0000 0244	0x4809 C244	0x480B 4244
MMCHS_CUR_CAPA	RW	32	0x0000 0248	0x4809 C248	0x480B 4248
MMCHS_FE	W	32	0x0000 0250	0x4809 C250	0x480B 4250
MMCHS_ADMAES	RW	32	0x0000 0254	0x4809 C254	0x480B 4254
MMCHS_ADMASAL	RW	32	0x0000 0258	0x4809 C258	0x480B 4258
MMCHS_PVINITSD	R	32	0x0000 0260	0x4809 C260	0x480B 4260

Table 25-58. MMC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MMC1 Physical Address	MMC2 Physical Address
MMCHS_PVHSSDR12	R	32	0x0000 0264	0x4809 C264	0x480B 4264
MMCHS_PVSDR25SDR50	R	32	0x0000 0268	0x4809 C268	0x480B 4268
MMCHS_PVSDR104DDR50	R	32	0x0000 026C	0x4809 C26C	0x480B 426C
MMCHS_REV	R	32	0x0000 02FC	0x4809 C2FC	0x480B 42FC

Table 25-59. MMC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MMC3 Physical Address	MMC4 Physical Address
MMCHS_HL_REV	R	32	0x0000 0000	0x480A D000	0x480D 1000
MMCHS_HL_HWINFO	R	32	0x0000 0004	0x480A D004	0x480D 1004
MMCHS_HL_SYSCONFIG	RW	32	0x0000 0010	0x480A D010	0x480D 1010
MMCHS_SYSCONFIG	RW	32	0x0000 0110	0x480A D110	0x480D 1110
MMCHS_SYSSTATUS	R	32	0x0000 0114	0x480A D114	0x480D 1114
MMCHS_CSRE	RW	32	0x0000 0124	0x480A D124	0x480D 1124
MMCHS_SYSTEST	RW	32	0x0000 0128	0x480A D128	0x480D 1128
MMCHS_CON	RW	32	0x0000 012C	0x480A D12C	0x480D 112C
MMCHS_PWCNT	RW	32	0x0000 0130	0x480A D130	0x480D 1130
RESERVED	R	32	0x0000 0134	0x480A D134	0x480D 1134
MMCHS_SDMASA	RW	32	0x0000 0200	0x480A D200	0x480D 1200
MMCHS_BLK	RW	32	0x0000 0204	0x480A D204	0x480D 1204
MMCHS_ARG	RW	32	0x0000 0208	0x480A D208	0x480D 1208
MMCHS_CMD	RW	32	0x0000 020C	0x480A D20C	0x480D 120C
MMCHS_RSP10	R	32	0x0000 0210	0x480A D210	0x480D 1210
MMCHS_RSP32	R	32	0x0000 0214	0x480A D214	0x480D 1214
MMCHS_RSP54	R	32	0x0000 0218	0x480A D218	0x480D 1218
MMCHS_RSP76	R	32	0x0000 021C	0x480A D21C	0x480D 121C
MMCHS_DATA	RW	32	0x0000 0220	0x480A D220	0x480D 1220
MMCHS_PSTATE	R	32	0x0000 0224	0x480A D224	0x480D 1224
MMCHS_HCTL	RW	32	0x0000 0228	0x480A D228	0x480D 1228
MMCHS_SYSCTL	RW	32	0x0000 022C	0x480A D22C	0x480D 122C
MMCHS_STAT	RW	32	0x0000 0230	0x480A D230	0x480D 1230
MMCHS_IE	RW	32	0x0000 0234	0x480A D234	0x480D 1234
MMCHS_ISE	RW	32	0x0000 0238	0x480A D238	0x480D 1238
MMCHS_AC12	RW	32	0x0000 023C	0x480A D23C	0x480D 123C
MMCHS_CAPA	RW	32	0x0000 0240	0x480A D240	0x480D 1240
MMCHS_CAPA2	R	32	0x0000 0244	0x480A D244	0x480D 1244
MMCHS_CUR_CAPA	RW	32	0x0000 0248	0x480A D248	0x480D 1248
MMCHS_FE	W	32	0x0000 0250	0x480A D250	0x480D 1250
MMCHS_ADMAES	RW	32	0x0000 0254	0x480A D254	0x480D 1254
MMCHS_ADMASAL	RW	32	0x0000 0258	0x480A D258	0x480D 1258
MMCHS_PVINITSD	R	32	0x0000 0260	0x480A D260	0x480D 1260
MMCHS_PVHSSDR12	R	32	0x0000 0264	0x480A D264	0x480D 1264
MMCHS_PVSDR25SDR50	R	32	0x0000 0268	0x480A D268	0x480D 1268
MMCHS_PVSDR104DDR50	R	32	0x0000 026C	0x480A D26C	0x480D 126C
MMCHS_REV	R	32	0x0000 02FC	0x480A D2FC	0x480D 12FC

25.6.2.2 eMMC/SD/SDIO Register Description

Table 25-60. MMCHS_HL_REV

Address Offset	0x0000 0000

Table 25-60. MMCHS_HL_REV (continued)

Physical Address	0x4809 C000 0x480B 4000 0x480A D000 0x480D 1000	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31–0
REVISION

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

Table 25-61. MMCHS_HL_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x4809 C004 0x480B 4004 0x480A D004 0x480D 1004	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Information about the IP module's hardware configuration.		
Type	R		

31–7	6	5–2	1	0
RESERVED	RETMODE	MEM_SIZE	MERGE_MEM	MADMA_EN

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x-
6	RETMODE	Retention Mode generic parameter This bit field indicates whether the retention mode is supported using the pin PIRFFRET. Read 0x1: Retention mode enabled Read 0x0: Retention mode disabled	R	0x-
5:2	MEM_SIZE	Memory size for FIFO buffer: Read 0x2: Memory of 1024 bytes, max block length is 1024 bytes Read 0x1: Memory of 512 bytes, max block length is 512 bytes Read 0x8: Memory of 4096 bytes, max block length is 2048 bytes Read 0x4: Memory of 2048 bytes, max block length is 2048 bytes	R	0x-
1	MERGE_MEM	Memory merged for FIFO buffer: This register defines the configuration of FIFO buffer architecture. If the bit is set STA and DFT shall support clock multiplexing and balancing. Read 0x1: A single memory is used with multiplexed addresses, data and clocks. Read 0x0: 2 memories instantiated, one per data transfer direction.	R	0x-

Bits	Field Name	Description	Type	Reset
0	MADMA_EN	Master DMA enabled generic parameter: This register defines the configuration of the controller to know if it supports the master DMA management called ADMA. Read 0x1: Controller supports ADMA Read 0x0: No Master DMA (ADMA) management supported	R	0x-

Table 25-62. MMCHS_HL_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4809 C010 0x480B 4010 0x480A D010 0x480D 1010	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Clock Management Configuration Register		
Type	RW		

31–6	5–4	3–2	1	0
RESERVED	STANDBYMODE	IDLEMODE	FREEEMU	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode: local initiator is unconditionally placed in standby state.Backup mode, for debug only. 0x1: No-standby mode: local initiator is unconditionally placed out of standby state.Backup mode, for debug only. 0x3: Smart-Standby wakeup-capable mode: local initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module may generate (master-related) wakeup events when in standby state.Mode is only relevant if the appropriate IP module "mwakeup" output is implemented. 0x2: Smart-standby mode: local initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator.IP module shall not generate (initiator-related) wakeup events.	RW	0x2

Bits	Field Name	Description	Type	Reset
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's IDLE requests unconditionally, i.e. regardless of the IP module's internal requirements.Backup mode, for debug only. 0x1: No-idle mode: local target never enters idle state.Backup mode, for debug only. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module shall not generate (IRQ- or DMA-request-related) wakeup events.	RW	0x2
1	FREEEMU	Sensitivity to emulation (debug) suspend input signal. Functionality NOT implemented in MMCHS. 0x0: IP module is sensitive to emulation suspend 0x1: IP module is not sensitive to emulation suspend	RW	0
0	SOFTRESET	Software reset. (Optional) Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing Read 0x0: Reset done, no pending action	RW	0

Table 25-63. MMCHS_SYSCONFIG

Address Offset	0x0000 0110		
Physical Address	0x4809 C110 0x480B 4110 0x480A D110 0x480D 1110	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	System Configuration Register This register allows controlling various parameters of the Interconnect interface.		
Type	RW		

31:14	13:12	11:10	9:8	7:5	4:3	2	1	0
RESERVED	STANDBYMODE	RESERVED	CLOCKACTIVITY	RESERVED	SIDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0 0000

Bits	Field Name	Description	Type	Reset
13:12	STANDBYMODE	Master interface power Management, standby/wait control. The bit field is only useful when generic parameter MMCHS_HL_HWINFO[0] MADMA_EN (Master ADMA enable) is set as active, otherwise it is a read only register read a '0'. 0x0: Force-standby. Mstandby is forced unconditionnaly. 0x1: No-standby. Mstandby is never asserted. 0x2: Smart-standby mode: local initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator.IP module shall not generate (initiator-related) wakeup events.	RW	0x2
11:10	RESERVED		R	0x0
9:8	CLOCKACTIVITY	Clocks activity during wake up mode period. Bit8: Interface clock Bit9: Functional clock 0x0: Interface and Functional clock may be switched off. 0x1: Interface clock is maintained. Functional clock may be switched-off. 0x3: Interface and Functional clocks are maintained. 0x2: Functional clock is maintained. Interface clock may be switched-off.	RW	0x0
7:5	RESERVED	This bit is initialized to zero, and writes to it are ignored. Reads return 0.	R	0x0
4:3	SIDLEMODE	Power management 0x0: If an IDLE request is detected, the MMCHS acknowledges it unconditionally and goes in Inactive mode. Interrupt and DMA requests are unconditionally de-asserted. 0x1: If an IDLE request is detected, the request is ignored and the module keeps on behaving normally. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module shall not generate (IRQ- or DMA-request-related) wakeup events.	RW	0x2
2	ENAWAKEUP	Wakeup feature control 0x0: Wakeup capability is disabled 0x1: Wakeup capability is enabled	RW	1
1	SOFTRESET	Software reset. The bit is automatically reset by the hardware. During reset, it always returns 0. Write 0x0: No effect. Write 0x1: Trigger a module reset. Read 0x1: The module is reset. Read 0x0: Normal mode	RW	0

Bits	Field Name	Description	Type	Reset
0	AUTOIDLE	Internal Clock gating strategy 0x0: Clocks are free-running 0x1: Automatic clock gating strategy is applied, based on the Interconnect and MMC interface activity	RW	1

Table 25-64. MMCHS_SYSSTATUS

Address Offset	0x0000 0114		
Physical Address	0x4809 C114 0x480B 4114 0x480A D114 0x480D 1114	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	System Status Register This register provides status information about the module excluding the interrupt status information		
Type	R		

31:1	0
RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RESETDONE	Internal Reset Monitoring Note: the debounce clock , the system clock (Interface) and the functional clock shall be provided to the MMC/SD/SDIO host controller to allow the internal reset monitoring. Read 0x1: Reset completed. Read 0x0: Internal module reset is on-going	R	0

Table 25-65. MMCHS_CSRE

Address Offset	0x0000 0124		
Physical Address	0x4809 C124 0x480B 4124 0x480A D124 0x480D 1124	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Card Status Response Error This register enables the host controller to detect card status errors of response type R1, R1b for all cards and of R5, R5b and R6 response for cards types SD or SDIO. When a bit MMCHS_CSRE[i] is set to 1, if the corresponding bit at the same position in the response MMCHS_RSP0[i] is set to 1, the host controller indicates a card error (MMCHS_STAT[CERR]) interrupt status to avoid the host driver reading the response register (MMCHS_RSP0). Note: No automatic card error detection for autoCMD12 is implemented; the host system has to check autoCMD12 response register (MMCHS_RESP76) for possible card errors.		
Type	RW		

31:0
CSRE

Bits	Field Name	Description	Type	Reset
31:0	CSRE	Card status response error	RW	0x0000 0000

Table 25-66. MMCHS_SYSTEST

Address Offset	0x0000 0128

Table 25-66. MMCHS_SYSTEST (continued)

Physical Address	0x4809 C128 0x480B 4128 0x480A D128 0x480D 1128	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	System Test Register This register is used to control the signals that connect to I/O pins when the module is configured in system test (SYSTEST) mode for boundary connectivity verification. Note: In SYSTEST mode, a write into MMCHS_CMD register will not start a transfer. The buffer behaves as a stack accessible only by the local host (push and pop operations). In this mode, the Transfer Block Size (MMCHS_BLK[BLEN]) and the Blocks count for current transfer (MMCHS_BLK[NBLK]) are needed to generate a Buffer write ready interrupt (MMCHS_STAT[BWR]) or a Buffer read ready interrupt (MMCHS_STAT[BRR]) and DMA requests if enabled.		
Type	RW		

31:17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	OBI	SDCD	SDWP	WAKD	SSB	D7D	D6D	D5D	D4D	D3D	D2D	D1D	D0D	DDIR	CDAT	CDIR	MCKD

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16	OBI	Out-Of-Band Interrupt (OBI) data value Read 0x1: The Out-of-Band Interrupt pin is driven high. Read 0x0: The Out-of-Band Interrupt pin is driven low.	R	0
15	SDCD	Card detect input signal (mmci_sdcd) data value Read 0x1: The card detect pin is driven high. Read 0x0: The card detect pin is driven low.	R	0
14	SDWP	Write protect input signal (mmci_sdwp) data value Read 0x1: The write protect pin mmci_sdwp is driven high. Read 0x0: The write protect pin mmci_sdwp is driven low.	R	0
13	WAKD	Wake request output signal data value Write 0x0: The pin SWAKEUP is driven low. Write 0x1: The pin SWAKEUP is driven high. Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0
12	SSB	Set status bit This bit must be cleared prior attempting to clear a status bit of the interrupt status register (MMCHS_STAT). Write 0x0: Clear this SSB bitfield. Writing 0 does not clear already set status bits; Write 0x1: Force to 1 all status bits of the interrupt status register (MMCHS_STAT) only if the corresponding bitfield in the Interrupt signal enable register (MMCHS_ISE) is set. Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0

Bits	Field Name	Description	Type	Reset
11	D7D	DAT7 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT7 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT7 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT7 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT7 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
10	D6D	DAT6 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT6 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT6 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT6 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT6 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
9	D5D	DAT5 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT5 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT5 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT5 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT5 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
8	D4D	DAT4 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT4 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT4 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT4 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT4 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0

Bits	Field Name	Description	Type	Reset
7	D3D	DAT3 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT3 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT3 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT3 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT3 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
6	D2D	DAT2 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT2 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT2 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT2 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT2 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
5	D1D	DAT1 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT1 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT1 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT1 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT1 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
4	D0D	DAT0 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT0 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT0 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT0 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT0 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0

Bits	Field Name	Description	Type	Reset
3	DDIR	Control of the DAT[7:0] pins direction. Write 0x0: The DAT lines are outputs (host to card) Write 0x1: The DAT lines are inputs (card to host) Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0
2	CDAT	CMD input/output signal data value Write 0x0: If SYSTEST[CDIR] = 0 (output mode direction), the CMD line is driven low. If SYSTEST[CDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[CDIR] = 0 (output mode direction), the CMD line is driven high. If SYSTEST[CDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[CDIR] = 1 (input mode direction), returns the value on the CMD line (high) If SYSTEST[CDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[CDIR] = 1 (input mode direction), returns the value on the CMD line (low). If SYSTEST[CDIR] = 0 (output mode direction), returns 0	RW	0
1	CDIR	Control of the CMD pin direction. Write 0x0: The CMD line is an output (host to card) Write 0x1: The CMD line is an input (card to host) Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0
0	MCKD	MMC clock output signal data value Write 0x0: The output clock is driven low. Write 0x1: The output clock is driven high. Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0

Table 25-67. MMCHS_CON

Address Offset	0x0000 012C		
Physical Address	0x4809 C12C 0x480B 412C 0x480A D12C 0x480D 112C	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Configuration Register This register is used: - to select the functional mode or the SYSTEST mode for any card. - to send an initialization sequence to any card. - to enable the detection on DAT[1] of a card interrupt for SDIO cards only. and also to configure : - specific data and command transfers for MMC cards only. - the parameters related to the card detect and write protect input signals.		
Type	RW		

31:22	21	20	19	18	17	16	15	14	13	12	11	10:9	8	7	6	5	4	3	2	1	0
RESERVED	SDMA_LNE	DMA_MNS	DDR	BOOT_CF0	BOOT_ACK	CLKEXTFREE	PADEN	OBIE	OBIP	CEATA	CTPL	DVAL	WPP	CDP	MIT	DW8	MODE	STR	HR	INIT	OD

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000

Bits	Field Name	Description	Type	Reset
21	SDMA_LNE	Slave DMA Level/Edge Request: The waveform of the DMA request can be configured either edge sensitive with early de-assertion on first access to MMCHS_DATA register or late de-assertion, request remains active until last allowed data written into MMCHS_DATA. 0x0: Slave DMA edge sensitive, Early DMA de-assertion 0x1: Slave DMA level sensitive, Late DMA de-assertion	RW	0
20	DMA_MNS	DMA Master or Slave selection: When this bit is set and the controller is configured to use the DMA, Interconnect master interface is used to get datas from system using ADMA2 procedure (direct access to the memory).This option is only available if generic parameter MADMA_EN is asserted to '1'. 0x0: The controller is slave on data transfers with system. 0x1: The controller is master on data exchange with system, controller must be configured as using DMA.	RW	0
19	DDR	Dual Data Rate mode: When this register is set, the controller uses both clock edge to emit or receive data. Odd bytes are transmitted on falling edges and even bytes are transmitted on rise edges. It only applies on Data bytes and CRC, Start, end bits and CRC status are kept full cycle. This bit field is only meaningful and active for even clock divider ratio of MMCHS_SYSCTL[CLKD], it is insensitive to MMCHS_HCTL[HSPE] setting. 0x0: Standard mode : data are transmitted on a single edge depending on MMCHS_HCTRL[HSPE]. 0x1: Data Bytes and CRC are transmitted on both edge.	RW	0
18	BOOT_CF0	Boot status supported: This register is set when the CMD line need to be forced to '0' for a boot sequence. CMD line is driven to '0' after writing in MMCHS_CMD. The line is released when this bit field is de-asserted and abort data transfer in case of a pending transaction. Write 0x0: CMD line is released when it was previously forced to '0' by a boot sequence. Write 0x1: CMD line forced to '0' is enabled and will be active after writing into MMCHS_CMD Read 0x1: CMD line forced to '0' is enabled Read 0x0: CMD line not forced	RW	0
17	BOOT_ACK	Boot acknowledge received: When this bit is set the controller should receive a boot status on DAT0 line after next command issued. If no status is received a data timeout will be generated. 0x0: No acknowledge to be received 0x1: A boot status will be received on DAT0 line after issuing a command.	RW	0
16	CLKEXTFREE	External clock free running: This register is used to maintain card clock out of transfer transaction to enable slave module for example to generate a synchronous interrupt on DAT[1]. The Clock will be maintain only if MMCHS_SYSCTL[CEN] is set. 0x0: External card clock is cut off outside active transaction period. 0x1: External card clock is maintain even out of active transaction period only if MMCHS_SYSCTL[CEN] is set.	RW	0

Bits	Field Name	Description	Type	Reset
15	PADEN	Control Power for MMC Lines: This register is only useful when MMC PADs contain power saving mechanism to minimize its leakage power. It works as a GPIO that directly control the ACTIVE pin of PADs. Excepted for DAT[1], the signal is also combine outside the module with the dedicated power control MMCHS_CON[CTPL] bit. 0x0: ADPIDLE module pin is not forced, it is automatically generated by the MMC fsms. 0x1: ADPIDLE module pin is forced to active state.	RW	0
14	OBIE	Out-of-Band Interrupt Enable MMC cards only: This bit enables the detection of Out-of-Band Interrupt on MMCOBI input pin. The usage of the Out-of-Band signal (OBI) is optional and depends on the system integration. 0x0: Out-of-Band interrupt detection disabled 0x1: Out-of-Band interrupt detection enabled	RW	0
13	OBIP	Out-of-Band Interrupt Polarity MMC cards only: This bit selects the active level of the out-of-band interrupt coming from MMC cards. The usage of the Out-of-Band signal (OBI) is optional and depends on the system integration. 0x0: Active high level 0x1: Active low level	RW	0
12	CEATA	CE-ATA control mode MMC cards compliant with CE-ATA:By default, this bit is set to 0. It is used to indicate that next commands are considered as specific CE-ATA commands that potentially use 'command completion' features. 0x0: Standard MMC/SD/SDIO mode. 0x1: CE-ATA mode next commands are considered as CE-ATA commands.	RW	0
11	CTPL	Control Power for DAT[1] line MMC and SD cards: By default, this bit is set to 0 and the host controller automatically disables all the input buffers outside of a transaction to minimize the leakage current. SDIO cards: When this bit is set to 1, the host controller automatically disables all the input buffers except the buffer of DAT[1] outside of a transaction in order to detect asynchronous card interrupt on DAT[1] line and minimize the leakage current of the buffers. 0x0: Disable all the input buffers outside of a transaction. 0x1: Disable all the input buffers except the buffer of DAT[1] outside of a transaction.	RW	0

Bits	Field Name	Description	Type	Reset
10:9	DVAL	Debounce filter value All cards This register is used to define a debounce period to filter the card detect input signal (mmci_sdcd). The usage of the card detect input signal (mmci_sdcd) is optional and depends on the system integration and the type of the connector housing that accommodates the card. 0x0: 33 us debounce period 0x1: 231 us debounce period 0x3: 8,4 ms debounce period 0x2: 1 ms debounce period	RW	0x3
8	WPP	Write protect polarity For SD and SDIO cards only This bit selects the active level of the write protect input signal (mmci_sdwp). The usage of the write protect input signal (mmci_sdwp) is optional and depends on the system integration and the type of the connector housing that accommodates the card. 0x0: Active high level 0x1: Active low level	RW	0
7	CDP	Card detect polarity All cards This bit selects the active level of the card detect input signal (mmci_sdcd). The usage of the card detect input signal (mmci_sdcd) is optional and depends on the system integration and the type of the connector housing that accommodates the card. 0x0: Active low level 0x1: Active high level	RW	0
6	MIT	MMC interrupt command Only for MMC cards. This bit must be set to 1, when the next write access to the command register (MMCHS_CMD) is for writing a MMC interrupt command (CMD40) requiring the command timeout detection to be disabled for the command response. 0x0: Command timeout enabled 0x1: Command timeout disabled	RW	0
5	DW8	8-bit mode MMC select For SD/SDIO cards, this bit must be set to 0. For MMC card, this bit must be set following a valid SWITCH command (CMD6) with the correct value and extend CSD index written in the argument. Prior to this command, the MMC card configuration register (CSD and EXT_CSD) must be verified for compliancy with MMC standard specification 4.x (see section 3.6). 0x0: 1-bit or 4-bit Data width (DAT[0] used, MMC, SD cards) 0x1: 8-bit Data width (DAT[7:0] used, MMC cards)	RW	0

Bits	Field Name	Description	Type	Reset
4	MODE	Mode select All cards This bit select between Functional mode and SYSTEST mode. 0x0: Functional mode. Transfers to the MMC/SD/SDIO cards follow the card protocol. MMC clock is enabled. MMC/SD transfers are operated under the control of the CMD register. 0x1: SYSTEST mode The signal pins are configured as general-purpose input/output and the 1024-byte buffer is configured as a stack memory accessible only by the local host or system DMA. The pins retain their default type (input, output or in-out). SYSTEST mode is operated under the control of the SYSTEST register.	RW	0
3	STR	Stream command Only for MMC cards. This bit must be set to 1 only for the stream data transfers (read or write) of the adtc commands. Stream read is a class 1 command (CMD11: READ_DAT_UNTIL_STOP). Stream write is a class 3 command (CMD20: WRITE_DAT_UNTIL_STOP). 0x0: Block oriented data transfer 0x1: Stream oriented data transfer	RW	0
2	HR	Broadcast host response Only for MMC cards. This register is used to force the host to generate a 48-bit response for bc command type. It can be used to terminate the interrupt mode by generating a CMD40 response by the core (see section 4.3, "Interrupt Mode", in the MMC specification). In order to have the host response to be generated in open drain mode, the register MMCHS_CON[OD] must be set to 1. When MMCHS_CON[CEATA] is set to 1 and MMCHS_ARG set to 0x00000000 when writing 0x00000000 into MMCHS_CMD register, the host controller performs a 'command completion signal disable' token i.e. CMD line held to '0' during 47 cycles followed by a 1. 0x0: The host does not generate a 48-bit response instead of a command. 0x1: The host generates a 48-bit response instead of a command or a command completion signal disable token.	RW	0
1	INIT	Send initialization stream All cards. When this bit is set to 1, and the card is idle, an initialization sequence is sent to the card. An initialization sequence consists of setting the CMD line to 1 during 80 clock cycles. The initialisation sequence is mandatory - but it is not required to do it through this bit - this bit makes it easier. Clock divider (MMCHS_SYSCTL[CLKD]) should be set to ensure that 80 clock periods are greater than 1ms. (see section 9.3, "Power-Up", in the MMC card specification, or section 6.4 in the SD card specification). Note: in this mode, there is no command sent to the card and no response is expected 0x0: The host does not send an initialization sequence. 0x1: The host sends an initialization sequence.	RW	0

Bits	Field Name	Description	Type	Reset
0	OD	Card open drain mode. Only for MMC cards. This bit must be set to 1 for MMC card commands 1, 2, 3 and 40, and if the MMC card bus is operating in open-drain mode during the response phase to the command sent. Typically, during card identification mode when the card is either in idle, ready or ident state. It is also necessary to set this bit to 1, for a broadcast host response (see Broadcast host response register MMCHS_CON[HR]) 0x0: No Open Drain 0x1: Open Drain or Broadcast host response	RW	0

Table 25-68. MMCHS_PWCNT

Address Offset	0x0000 0130		
Physical Address	0x4809 C130 0x480B 4130 0x480A D130 0x480D 1130	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Power Counter Register This register is used to program a mmc counter to delay command transfers after activating the PAD power, this value depends on PAD characteristics and voltage.		
Type	RW		

31–16	15–0
RESERVED	PWRCNT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	PWRCNT	Power counter register. This register is used to introduce a delay between the PAD ACTIVE pin assertion and the command issued. 0xFFFF: TCF x 65535 delay (card clock period) 0x0: No additional delay added 0x1: TCF delay (card clock period) 0xFFFE: TCF x 65534 delay (card clock period) 0x2: TCF x 2 delay (card clock period)	RW	0x0000

Table 25-69. MMCHS_DLL

Address Offset	0x0000 0134		
Physical Address	0x4809 C134 0x480B 4134	**Instance**	MMC1 MMC2
Description	DLL control and status register This register is used for tuning procedure required for SDR104/HS200 speed mode. It gives visibility and control on the DLL		
Type	RW		

31	30	29–22	21	20	19–13	12	11–6	5–4	3	2	1	0
DLL_SOFT_RESET	LOCK_TIMER	MAX_LOCK_DIFF	FORCE_SR_F	SWT	FORCE_SR_C	FORCE_VALUE	SLAVE_RATIO	RESERVED	DLL_UNLOCK_CLEAR	DLL_UNLOCK_STICKY	DLL_CALIB	DLL_LOCK

Bits	Field Name	Description	Type	Reset
31	DLL_SOFT_RESET	Soft reset for DLL, active HIGH. Write 0x0: No action. Write 0x1: Issue soft reset Read 0x1: Reset is in progress Read 0x0: Reset completed.	RW	1
30	LOCK_TIMER	Timer for the dll_lock signal to be asserted after reset. 0x0: 1024 cycles (equivalent to DLL fast mode lock) 0x1: 66560 cycles	RW	0
29:22	MAX_LOCK_DIFF	Maximum number of taps that the master DLL clock period measurement can deviate without resulting in the master DLL losing lock.	RW	0x00
21	FORCE_SR_F	Forced fine delay value.	RW	0x0
20	SWT	Software Tuning enable. The bit shall be set to manage the tuning sequence fully in software. NOTE: For proper operation when SDR104/HS200 mode is used this bit must be set to 0x1 which disables the Conflict Error (CFT Error) on the CMD line. 0x0: No software tuning sequence. 0x1: Execute software tuning sequence.	RW	0x0
19:13	FORCE_SR_C	Forced coarse delay value	RW	0x00
12	FORCE_VALUE	Put forced values to slave DLL, ignoring master DLL output and ratio value. 0x0: Do not put force value 0x1: Put force value.	RW	0
11:6	SLAVE_RATIO	Fraction of a clock cycle for the shift to be implemented, in units of 256ths of a clock cycle. 0x6: 135 degrees delay 0x3F: 4 clocks delay 0x8: 180 degrees delay 0x2: 45 degrees delay 0xA: 225 degrees delay 0x10: Full clock delay 0x0: 0 degree delay 0xC: 270 degrees delay 0x4: 90 degrees delay 0xE: 315 degrees delay	RW	0x00
5:4	RESERVED		R	0x0
3	DLL_UNLOCK_CLEAR	Clears the phy_reg_status_mdll_unlock_sticky flags of the DLL. 0x0: No effect. 0x1: Clears the flag.	RW	0
2	DLL_UNLOCK_STICKY	Asserted when any single period measurement exceeds MAX_LOCK_DIFF.	R	0
1	DLL_CALIB	Enables Slave DLL to update new delay values. 0x0: Disabled 0x1: Enabled	RW	0
0	DLL_LOCK	Master DLL lock status. Read 0x1: DLL is locked Read 0x0: DLL is not locked	R	0

Table 25-70. MMCHS_SDMASA

Address Offset	0x0000 0200		
Physical Address	0x4809 C200 0x480B 4200 0x480A D200 0x480D 1200	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	SDMA System Address / Argument 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMA_ARG2																															

Bits	Field Name	Description	Type	Reset
31:0	SDMA_ARG2	SDMA System Address / Argument 2 This register contains the physical system memory address used for DMA transfers or the second argument for the Auto CMD23. (1) SDMA System Address This register contains the system memory address for a SDMA transfer. When the Host Controller stops a SDMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the Block Size register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the SDMA transfer. When restarting SDMA by the Resume command or by setting Continue Request in the Block Gap Control register, the Host Controller shall start at the next contiguous address stored here in the SDMA System Address register. ADMA does not use this register. (2) Argument 2 This register is used with the Auto CMD23 to set a 32-bit block count value to the argument of the CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without AMDA, the available block count value is limited by the Block Count register. 65535 blocks is the maximum value in this case.	RW	0x0000 0000

Table 25-71. MMCHS_BLK

Address Offset	0x0000 0204		
Physical Address	0x4809 C204 0x480B 4204 0x480A D204 0x480D 1204	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Transfer Length Configuration Register MMCHS_BLK[BLEN] is the block size register. MMCHS_BLK[NBLK] is the block count register. This register shall be used for any card.		

Table 25-71. MMCHS_BLK (continued)

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NBLK																RESERVED				BLEN											

Bits	Field Name	Description	Type	Reset
31:16	NBLK	Blocks count for current transfer This register is enabled when Block count Enable (MMCHS_CMD[BCE]) is set to 1 and is valid only for multiple block transfers. Setting the block count to 0 results no data blocks being transferred. Note: The host controller decrements the block count after each block transfer and stops when the count reaches zero. This register can be accessed only if no transaction is executing (i.e, after a transaction has stopped). Read operations during transfers may return an invalid value and write operation will be ignored. In suspend context, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, The local host shall restore the previously saved block count. 0xFFFF: 65535 blocks 0x0: Stop count 0x1: 1 block 0x2: 2 blocks	RW	0x0000
15:12	RESERVED		R	0x0
11:0	BLEN	Transfer Block Size. This register specifies the block size for block data transfers. Read operations during transfers may return an invalid value, and write operations are ignored. When a CMD12 command is issued to stop the transfer, a read of the BLEN field after transfer completion (MMCHS_STAT[TC] set to 1) will not return the true byte number of data length while the stop occurs but the value written in this register before transfer is launched. 0x1: 1 byte block length 0x7FF: 2047 bytes block length 0x0: No data transfer 0x1FF: 511 bytes block length 0x800: 2048 bytes block length 0x2: 2 bytes block length 0x3: 3 bytes block length 0x200: 512 bytes block length	RW	0x000

Table 25-72. MMCHS_ARG

Address Offset	0x0000 0208		
Physical Address	0x4809 C208 0x480B 4208 0x480A D208 0x480D 1208	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Command Argument Register This register contains command argument specified as bit 39-8 of Command-Format These registers must be initialized prior to sending the command itself to the card (write action into the register MMCHS_CMD register). Only exception is for a command index specifying stuff bits in arguments, making a write unnecessary.		

Table 25-72. MMCHS_ARG (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG																															

Bits	Field Name	Description	Type	Reset
31:0	ARG	Command argument bits [31:0]	RW	0x0000 0000

Table 25-73. MMCHS_CMD

Address Offset	0x0000 020C		
Physical Address	0x4809 C20C 0x480B 420C 0x480A D20C 0x480D 120C	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Command and Transfer Mode Register MMCHS_CMD[31:16] = the command register MMCHS_CMD[15:0] = the transfer mode. This register configures the data and command transfers. A write into the most significant byte send the command. A write into MMCHS_CMD[15:0] registers during data transfer has no effect. This register shall be used for any card. Note: In SYSTEST mode, a write into MMCHS_CMD register will not start a transfer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		INDX						CMD_TYPE		DP	CICE	CCCE	RESERVED	RSP_TYPE		RESERVED										MSBS	DDIR	ACEN		BCE	DE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:24	INDX	Command index Binary encoded value from 0 to 63 specifying the command number send to card 0xD: CMD13 or ACMD13 0x33: CMD51 or ACMD51 0x3B: CMD59 or ACMD59 0x15: CMD21 or ACMD21 0x1E: CMD30 or ACMD30 0x8: CMD8 or ACMD8 0x5: CMD5 or ACMD5 0x2E: CMD46 or ACMD46 0x1B: CMD27 or ACMD27 0x2C: CMD44 or ACMD44 0x36: CMD54 or ACMD54 0x2: CMD2 or ACMD2 0x3E: CMD62 or ACMD62 0x4: CMD4 or ACMD4 0x39: CMD57 or ACMD57 0x32: CMD50 or ACMD50 0x6: CMD6 or ACMD6 0x1: CMD1 or ACMD1 0x1D: CMD29 or ACMD29 0x3F: CMD63 or ACMD63	RW	0x00

Bits	Field Name	Description	Type	Reset
		0x28: CMD40 or ACMD40 0x3A: CMD58 or ACMD58 0x24: CMD36 or ACMD36 0x0: CMD0 or ACMD0 0x2D: CMD45 or ACMD45 0x38: CMD56 or ACMD56 0x3C: CMD60 or ACMD60 0xB: CMD11 or ACMD11 0x3D: CMD61 or ACMD61 0x20: CMD32 or ACMD32 0x3: CMD3 or ACMD3 0x17: CMD23 or ACMD23 0x30: CMD48 or ACMD48 0x31: CMD49 or ACMD49 0x11: CMD17 or ACMD17 0x23: CMD35 or ACMD35 0x35: CMD53 or ACMD53 0x2F: CMD47 or ACMD47 0xA: CMD10 or ACMD10 0x9: CMD9 or ACMD9 0x10: CMD16 or ACMD16 0x26: CMD38 or ACMD38 0x21: CMD33 or ACMD33 0x25: CMD37 or ACMD37 0x12: CMD18 or ACMD18 0x13: CMD19 or ACMD19 0x2B: CMD43 or ACMD43 0x37: CMD55 or ACMD55 0x18: CMD24 or ACMD24 0x14: CMD20 or ACMD20 0xE: CMD14 or ACMD14 0x16: CMD22 or ACMD22 0x2A: CMD42 or ACMD42 0x1C: CMD28 or ACMD28 0x7: CMD7 or ACMD7 0x19: CMD25 or ACMD25 0x1F: CMD31 or ACMD31 0x34: CMD52 or ACMD52 0x1A: CMD26 or ACMD26 0x29: CMD41 or ACMD41 0xF: CMD15 or ACMD15 0xC: CMD12 or ACMD12 0x27: CMD39 or ACMD39 0x22: CMD34 or ACMD34		
23:22	CMD_TYPE	Command type This register specifies three types of special command: Suspend, Resume and Abort.	RW	0x0

Bits	Field Name	Description	Type	Reset
		These bits shall be set to 00b for all other commands. 0x0: Others Commands 0x1: CMD52 for writing "Bus Suspend" in CCCR 0x3: Abort command CMD12, CMD52 for writing " I/O Abort" in CCCR 0x2: CMD52 for writing "Function Select" in CCCR		
21	DP	Data present select This register indicates that data is present and DAT line shall be used. It must be set to 0 in the following conditions: - command using only CMD line - command with no data transfer but using busy signal on DAT[0] - Resume command 0x0: Command with no data transfer 0x1: Command with data transfer	RW	0
20	CICE	Command Index check enable This bit must be set to 1 to enable index check on command response to compare the index field in the response against the index of the command. If the index is not the same in the response as in the command, it is reported as a command index error (MMCHS_STAT[CIE] set to1). **Note:** The register CICE cannot be configured for an Auto CMD12, then index check is automatically checked when this command is issued. 0x0: Index check disable 0x1: Index check enable	RW	0
19	CCCE	Command CRC check enable This bit must be set to 1 to enable CRC7 check on command response to protect the response against transmission errors on the bus. If an error is detected, it is reported as a command CRC error (MMCHS_STAT[CCRC] set to 1). **Note:** The register CCCE cannot be configured for an Auto CMD12, and then CRC check is automatically checked when this command is issued. 0x0: CRC7 check disable 0x1: CRC7 check enable	RW	0
18	RESERVED		R	0
17:16	RSP_TYPE	Response type This bits defines the response type of the command 0x0: No response 0x1: Response Length 136 bits 0x3: Response Length 48 bits with busy after response 0x2: Response Length 48 bits	RW	0x0
15:6	RESERVED		R	0x000
5	MSBS	Multi/Single block select This bit must be set to 1 for data transfer in case of multi block command. For any others command this bit shall be set to 0.	RW	0

Bits	Field Name	Description	Type	Reset
		0x0: Single block. If this bit is 0, it is not necessary to set the register MMCHS_BLK[NBLK]. 0x1: Multi block. When Block Count is disabled (MMCHS_CMD[BCE] is set to 0) in Multiple block transfers (MMCHS_CMD[MSBS] is set to 1), the module can perform infinite transfer.		
4	DDIR	Data transfer Direction Select This bit defines either data transfer will be a read or a write. 0x0: Data Write (host to card) 0x1: Data Read (card to host)	RW	0
3:2	ACEN	Auto CMD Enable - SD card only. This field determines use of auto command functions. There are two methods to stop Multiple-block read and write operation 1. Auto CMD12 Enable When this field is set to 01b, the Host Controller issues CMD12 automatically when last block transfer is completed. Auto CMD12 error is indicated to the Auto CMD Error Status register (MMCHS_AC12). The Host Driver shall not set this bit if the command does not require CMD12. In particular, secure commands defined in the Part 3 File Security specification do not require CMD12. 2. Auto CMD23 Enable When this bit field is set to 10b, the Host Controller issues a CMD23 automatically before issuing a command specified in the Command Register. The Host Controller Version 3.00 and later shall support this function. The following conditions are required to use the Auto CMD23. – Auto CMD23 Supported (Host Controller Version is 3.00 or later) – A memory card that supports CMD23 (SCR[33]=1) – If DMA is used, it shall be ADMA. –Only when CMD18 or CMD25 is issued (**Note:** the Host Controller does not check command index.) Auto CMD23 can be used with or without ADMA. By writing the Command register, the Host Controller issues a CMD23 first and then issues a command specified by the Command Index in Command register. If response errors of CMD23 are detected, the second command is not issued. A CMD23 error is indicated in the Auto CMD Error Status register (MMCHS_AC12). 32-bit block count value for CMD23 is set to SDMA System Address / Argument 2 register (MMCHS_SDMASA). 0x0: Auto Command Disabled 0x1: Auto CMD12 enable or CCS detection enabled. 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x2: Auto CMD23 Enable		
1	BCE	Block Count Enable Multiple block transfers only. This bit is used to enable the block count register (MMCHS_BLK[NBLK]). When Block Count is disabled (MMCHS_CMD[BCE] is set to 0) in Multiple block transfers (MMCHS_CMD[MSBS] is set to 1), the module can perform infinite transfer. 0x0: Block count disabled for infinite transfer. 0x1: Block count enabled for multiple block transfer with known number of blocks	RW	0
0	DE	DMA Enable This bit is used to enable DMA mode for host data access. 0x0: DMA mode disable 0x1: DMA mode enable	RW	0

Table 25-74. MMCHS_RSP10

Address Offset	0x0000 0210		
Physical Address	0x4809 C210 0x480B 4210 0x480A D210 0x480D 1210	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Command Response[31:0] Register (bits [31:0] of the internal RSP register) This 32-bit register holds bits positions [31:0] of command response type R1/R1b/R2/R3/R4/R5/R5b/R6/R7		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP1																RSP0															

Bits	Field Name	Description	Type	Reset
31:16	RSP1	Command Response [31:16]	R	0x0000
15:0	RSP0	Command Response [15:0]	R	0x0000

Table 25-75. MMCHS_RSP32

Address Offset	0x0000 0214		
Physical Address	0x4809 C214 0x480B 4214 0x480A D214 0x480D 1214	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Command Response[63:32] Register (bits [63:32] of the internal RSP register) This 32-bit register holds bits positions [63:32] of command response type R2		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP3																RSP2															

Bits	Field Name	Description	Type	Reset
31:16	RSP3	Command Response [63:48]	R	0x0000
15:0	RSP2	Command Response [47:32]	R	0x0000

Table 25-76. MMCHS_RSP54

Address Offset	0x0000 0218

Table 25-76. MMCHS_RSP54 (continued)

Physical Address	0x4809 C218 0x480B 4218 0x480A D218 0x480D 1218	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Command Response[95:64] Register (bits [95:64] of the internal RSP register) This 32-bit register holds bits positions [95:64] of command response type R2		
Type	R		

31:16	15:0
RSP5	RSP4

Bits	Field Name	Description	Type	Reset
31:16	RSP5	Command Response [95:80]	R	0x0000
15:0	RSP4	Command Response [79:64]	R	0x0000

Table 25-77. MMCHS_RSP76

Address Offset	0x0000 021C		
Physical Address	0x4809 C21C 0x480B 421C 0x480A D21C 0x480D 121C	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Command Response[127:96] Register (bits [127:96] of the internal RSP register) This 32-bit register holds bits positions [127:96] of command response type R1(Auto CMD23)/R1b(Auto CMD12)/R2		
Type	R		

31:16	15:0
RSP7	RSP6

Bits	Field Name	Description	Type	Reset
31:16	RSP7	Command Response [127:112]	R	0x0000
15:0	RSP6	Command Response [111:96]	R	0x0000

Table 25-78. MMCHS_DATA

Address Offset	0x0000 0220		
Physical Address	0x4809 C220 0x480B 4220 0x480A D220 0x480D 1220	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Data Register This register is the 32-bit entry point of the buffer for read or write data transfers. The buffer size is 32bits x256(1024 bytes). Bytes within a word are stored and read in little endian format. This buffer can be used as two 512 byte buffers to transfer data efficiently without reducing the throughput. Sequential and contiguous access is necessary to increment the pointer correctly. Random or skipped access is not allowed. In little endian, if the local host accesses this register byte-wise or 16bit-wise, the least significant byte (bits [7:0]) must always be written/read first. The update of the buffer address is done on the most significant byte write for full 32-bit DATA register or on the most significant byte of the last word of block transfer. Example 1: Byte or 16-bit access Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1100 (2-bytes) OK Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=0100 (1-byte) OK Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1000 (1-byte) Bad		
Type	RW		

31:0
DATA

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data Register [31:0] In functional mode (MMCHS_CON[MODE] set to the default value 0) , A read access to this register is allowed only when the buffer read enable status is set to 1 (MMCHS_PSTATE[BRE]), otherwise a bad access (MMCHS_STAT[BADA]) is signaled. A write access to this register is allowed only when the buffer write enable status is set to 1(MMCHS_STATE[BWE]), otherwise a bad access (MMCHS_STAT[BADA]) is signaled and the data is not written.	RW	0x0000 0000

Table 25-79. MMCHS_PSTATE

Address Offset	0x0000 0224		
Physical Address	0x4809 C224 0x480B 4224 0x480A D224 0x480D 1224	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Present State Register The Host can get status of the Host Controller from this 32-bit read only register.		
Type	R		

31:25	24	23:20	19	18	17	16	15:12	11	10	9	8	7:4	3	2	1	0
RESERVED	CLEV	DLEV	WP	CDPL	CSS	CINS	RESERVED	BRE	BWE	RTA	WTA	RESERVED	RTR	DLA	DATI	CMDI

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLEV	CMD line signal level This status is used to check the CMD line level to recover from errors, and for debugging. The value of this register after reset depends on the CMD line level at that time. Read 0x1: The CMD line level is 1. Read 0x0: The CMD line level is 0.	R	-
23:20	DLEV	DAT[3:0] line signal level DAT[3] => bit 23 DAT[2] => bit 22 DAT[1] => bit 21 DAT[0] => bit 20 This status is used to check DAT line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0]. The value of these registers after reset depends on the DAT lines level at that time.	R	0x-
19	WP	Write protect switch pin level For SDIO cards only. This bit reflects the write protect input pin (mmci_sdwp) level. The value of this register after reset depends on the protect input pin (mmci_sdwp) level at that time. Read 0x1: If MMCHS_CON[WPP] is set to 0 (default), the card is not write protected, otherwise the card is protected. Read 0x0: If MMCHS_CON[WPP] is set to 0 (default), the card is write protected, otherwise the card is not protected.	R	-

Bits	Field Name	Description	Type	Reset
18	CDPL	Card detect pin level This bit reflects the inverse value of the card detect input pin (mmci_sdcd), debouncing is not performed on this bit and bit is valid only when Card State Stable (MMCHS_PSTAE[CSS]) is set to 1. Use of this bit is limited to testing since it must be debounced y software. The value of this register after reset depends on the card detect input pin (mmci_sdcd) level at that time. Read 0x1: The value of the card detect input pin (mmci_sdcd) is 0 Read 0x0: The value of the card detect input pin (mmci_sdcd) is 1	R	-
17	CSS	Card State Stable This bit is used for testing. It is set to 1 only when Card Detect Pin Level is stable (MMCHS_PSTATE[CDPL]). Debouncing is performed on the card detect input pin (mmci_sdcd) to detect card stability. This bit is not affected by a software reset. Read 0x1: No card or card inserted Read 0x0: Reset or Debouncing	R	0
16	CINS	Card inserted This bit is the debounced value of the card detect input pin (mmci_sdcd). An inactive to active transition of the card detect input pin (mmci_sdcd) will generate a card insertion interrupt (MMCHS_STAT[CINS]). A active to inactive transition of the card detect input pin (mmci_sdcd) will generate a card removal interrupt (MMCHS_STAT[REM]). This bit is not affected by a software reset. Read 0x1: If MMCHS_CON[CDP] is set to 1, the card has been inserted from the card slot. If MMCHS_CON[CDP] is set to 0, no card is detected. The card may have been removed from the card slot. Read 0x0: If MMCHS_CON[CDP] is set to 1, no card is detected. The card may have been removed from the card slot. If MMCHS_CON[CDP] is set to 0, the card has been inserted.	R	0
15:12	RESERVED		R	0x0
11	BRE	Buffer read enable This bit is used for non-DMA read transfers. It indicates that a complete block specified by MMCHS_BLK[BLEN] has been written in the buffer and is ready to be read. It is set to 0 when the entire block is read from the buffer. It is set to 1 when a block data is ready in the buffer and generates the Buffer read ready status of interrupt (MMCHS_STAT[BRR]). Read 0x1: Read BLEN bytes enable. Readable data exists in the buffer. Read 0x0: Read BLEN bytes disable	R	0
10	BWE	Buffer Write enable This status is used for non-DMA write transfers. It indicates if space is available for write data. Read 0x1: There is enough space in the buffer to write BLEN bytes of data. Read 0x0: There is no room left in the buffer to write BLEN bytes of data.	R	0

Bits	Field Name	Description	Type	Reset
9	RTA	Read transfer active This status is used for detecting completion of a read transfer. It is set to 1 after the end bit of read command or by activating a continue request (MMCHS_HCTL[CR]) following a stop at block gap request. This bit is set to 0 when all data have been read by the local host after last block or after a stop at block gap request. Read 0x1: read data transfer on going. Read 0x0: No valid data on the DAT lines.	R	0
8	WTA	Write transfer active This status indicates a write transfer active. It is set to 1 after the end bit of write command or by activating a continue request (MMCHS_HCTL[CR]) following a stop at block gap request. This bit is set to 0 when CRC status has been received after last block or after a stop at block gap request. Read 0x1: Write data transfer on going. Read 0x0: No valid data on the DAT lines.	R	0
7:4	RESERVED		R	0x0
3	RTR	Re-Tuning Request Host Controller may request Host Driver to execute re-tuning sequence by setting this bit when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This bit is cleared when a command is issued with setting MMCHS_AC12[22] ET. This bit isn't set to 1 if MMCHS_AC12[23] SCLK_SEL is set to 0 (using fixed sampling clock). Refer to MMCHS_CAPA2[15:14] RTM for more detail. Read 0x1: Sampling clock needs re-tuning Read 0x0: Fixed or well tuned sampling clock	R	0
2	DLA	DAT line active This status bit indicates whether one of the DAT line is in use. In the case of read transactions (card to host): This bit is set to 1 after the end bit of read command or by activating continue request MMCHS_HCTL[CR]. This bit is set to 0 when the host controller received the end bit of the last data block or at the beginning of the read wait mode. In the case of write transactions (host to card): This bit is set to 1 after the end bit of write command or by activating continue request MMCHS_HCTL[CR]. This bit is set to 0 on the end of busy event for the last block; host controller must wait 8 clock cycles with line not busy to really consider not "busy state" or after the busy block as a result of a stop at gap request. Read 0x1: DAT Line active Read 0x0: DAT Line inactive	R	0

Bits	Field Name	Description	Type	Reset
1	DATI	Command inhibit(DAT) This status bit is generated if either DAT line is active (MMCHS_PSTATE[DLA]) or Read transfer is active (MMCHS_PSTATE[RTA]) or when a command with busy is issued. This bit prevents the local host to issue a command. A change of this bit from 1 to 0 generates a transfer complete interrupt (MMCHS_STAT[TC]). Read 0x1: Issuing of command using DAT lines is not allowed Read 0x0: Issuing of command using the DAT lines is allowed	R	0
0	CMDI	Command inhibit(CMD) This status bit indicates that the CMD line is in use. This bit is set to 0 when the most significant byte is written into the command register. This bit is not set when Auto CMD12 is transmitted. This bit is set to 0 in either the following cases: - After the end bit of the command response, excepted if there is a command conflict error (MMCHS_STAT[CCRC] or MMCHS_STAT[CEB] set to 1) or a Auto CMD12 is not executed (MMCHS_AC12[ACNE]). - After the end bit of the command without response (MMCHS_CMD[RSP_TYPE] set to "00") In case of a command data error is detected (MMCHS_STAT[CTO] set to 1), this register is not automatically cleared. Read 0x1: Issuing of command using CMD line is not allowed Read 0x0: Issuing of command using CMD line is allowed	R	0

Table 25-80. MMCHS_HCTL

Address Offset	0x0000 0228		
Physical Address	0x4809 C228 0x480B 4228 0x480A D228 0x480D 1228	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Host Control Register This register defines the host controls to set power, wakeup and transfer parameters. MMCHS_HCTL[31:24] = Wakeup control MMCHS_HCTL[23:16] = Block gap control MMCHS_HCTL[15:8] = Power control MMCHS_HCTL[7:0] = Host control		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>OBWE</td><td>REM</td><td>INS</td><td>IWE</td><td colspan="4">RESERVED</td><td>IBG</td><td>RWC</td><td>CR</td><td>SBGR</td><td colspan="4">RESERVED</td><td colspan="3">SDVS</td><td>SDBP</td><td>CDSS</td><td>CDTL</td><td>RESERVED</td><td colspan="2">DMAS</td><td>HSPE</td><td>DTW</td><td>LED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27	OBWE	Wakeup event enable for 'Out-of-Band' Interrupt. This bit enables wakeup events for 'Out-of-Band' assertion. Wakeup is generated if the wakeup feature is enabled (MMCHS_SYSCONFIG[ENAWAKEUP]). The write to this register is ignored when MMCHS_CON[OBIE] is not set. 0x0: Disable wakeup on 'Out-of-Band' Interrupt 0x1: Enable wakeup on 'Out-of-Band' Interrupt	RW	0
26	REM	Wakeup event enable on SD card removal This bit enables wakeup events for card removal assertion. Wakeup is generated if the wakeup feature is enabled (MMCHS_SYSCONFIG[ENAWAKEUP]). 0x0: Disable wakeup on card removal 0x1: Enable wakeup on card removal	RW	0
25	INS	Wakeup event enable on SD card insertion This bit enables wakeup events for card insertion assertion. Wakeup is generated if the wakeup feature is enabled (MMCHS_SYSCONFIG[ENAWAKEUP]). 0x0: Disable wakeup on card insertion 0x1: Enable wakeup on card insertion	RW	0
24	IWE	Wakeup event enable on SD card interrupt This bit enables wakeup events for card interrupt assertion. Wakeup is generated if the wakeup feature is enabled (MMCHS_SYSCONFIG[ENAWAKEUP]). 0x0: Disable wakeup on card interrupt 0x1: Enable wakeup on card interrupt	RW	0
23:20	RESERVED		R	0x0
19	IBG	Interrupt block at gap This bit is valid only in 4-bit mode of SDIO card to enable interrupt detection in the interrupt cycle at block gap for a multiple block transfer. For MMC cards and for SD card this bit should be set to 0. 0x0: Disable interrupt detection at the block gap in 4-bit mode 0x1: Enable interrupt detection at the block gap in 4-bit mode	RW	0
18	RWC	Read wait control The read wait function is optional only for SDIO cards. If the card supports read wait, this bit must be enabled, then requesting a stop at block gap (MMCHS_HCTL[SBGR]) generates a read wait period after the current end of block. Be careful, if read wait is not supported it may cause a conflict on DAT line. 0x0: Disable Read Wait Control. Suspend/Resume cannot be supported. 0x1: Enable Read Wait Control	RW	0

Bits	Field Name	Description	Type	Reset
17	CR	Continue request This bit is used to restart a transaction that was stopped by requesting a stop at block gap (MMCHS_HCTL[SBGR]). Set this bit to 1 restarts the transfer. The bit is automatically set to 0 by the host controller when transfer has restarted i.e DAT line is active (MMCHS_PSTATE[DLA]) or transferring data (MMCHS_PSTATE[WTA]). The Stop at block gap request must be disabled (MMCHS_HCTL[SBGR]=0) before setting this bit. 0x0: No affect 0x1: transfer restart	RW	0
16	SBGR	Stop at block gap request This bit is used to stop executing a transaction at the next block gap. The transfer can restart with a continue request (MMCHS_HCTL[CR]) or during a suspend/resume sequence. In case of read transfer, the card must support read wait control. In case of write transfer, the host driver shall set this bit after all block data written. Until the transfer completion (MMCHS_STAT[TC] set to 1), the host driver shall leave this bit set to 1. If this bit is set, the local host shall not write to the data register (MMCHS_DATA). 0x0: Transfer mode 0x1: Stop at block gap	RW	0
15:12	RESERVED		R	0x0
11:9	SDVS	SD bus voltage select All cards. The host driver should set to these bits to select the voltage level for the card according to the voltage supported by the system (MMCHS_CAPA[VS18,VS30,VS33]) before starting a transfer. 0x6: 3.0V (Typical) 0x7: 3.3V (Typical) 0x5: 1.8V (Typical)	RW	0x0
8	SDBP	SD bus power Before setting this bit, the host driver shall select the SD bus voltage (MMCHS_HCTL[SDVS]). If the host controller detects the No card state, this bit is automatically set to 0. If the module is power off, a write in the command register (MMCHS_CMD) will not start the transfer. A write to this bit has no effect if the selected SD bus voltage MMCHS_HCTL[SDVS] is not supported according to capability register (MMCHS_CAPA[VS*]). 0x0: Power off 0x1: Power on	RW	0
7	CDSS	Card Detect Signal Selection This bit selects source for the card detection.When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupts caused by the glitch. The Interrupt Status/Signal Enable should be disabled during over the period of debouncing. 0x0: mmci_sdcd is selected (for normal use) 0x1: MMCHS_HCTL[6] CDTL is selected (for test purpose)	RW	0

Bits	Field Name	Description	Type	Reset
6	CDTL	Card Detect Test Level: This bit is enabled while MMCHS_HCTL[7] CDSS is set to 1 and it indicates whether the card is inserted or not. 0x0: No Card 0x1: Card Inserted	RW	0
5	RESERVED		R	0
4:3	DMAS	DMA Select Mode: One of supported DMA modes can be selected. The host driver shall check support of DMA modes by referring the Capabilities register MMCHS_CAPA . Use of selected DMA is determined by DMA Enable of the Transfer Mode register. This register is only meaningful when MADMA_EN is set to 1. When MADMA_EN is set to 0 the bit field is read only and returned value is 0. 0x0: Reserved 0x1: Reserved 0x3: Reserved 0x2: 32-bit Address ADMA2 is selected	RW	0x0
2	HSPE	Before setting this bit, the Host Driver shall check the MMCHS_CAPA[21] HSS. This bit shall not be set when dual data rate mode is activated in MMCHS_CON[DDR]. 0x0: The Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock 0x1: The Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock **NOTE: Do not set this bit to 0x1 because device was timing closed with HSPE bit set to 0x0 for all supported modes of operation.**	RW	0
1	DTW	Data transfer width For MMC card, this bit must be set following a valid SWITCH command (CMD6) with the correct value and extend CSD index written in the argument. Prior to this command, the MMC card configuration register (CSD and EXT_CSD) must be verified for compliance with MMC standard specification 4.x (see section 3.6). This register has no effect when the MMC 8-bit mode is selected (register MMCHS_CON[DW8] set to1), For SD/SDIO cards, this bit must be set following a valid SET_BUS_WIDTH command (ACMD6) with the value written in bit 1 of the argument. Prior to this command, the SD card configuration register (SCR) must be verified for the supported bus width by the SD card. 0x0: 1-bit Data width (DAT[0] used) 0x1: 4-bit Data width (DAT[3:0] used)	RW	0
0	LED	Reserved bit. LED control feature is not supported This bit is initialized to zero, and writes to it are ignored.	R	0

Table 25-81. MMCHS_SYSCTL

Address Offset	0x0000 022C		
Physical Address	0x4809 C22C 0x480B 422C 0x480A D22C 0x480D 122C	**Instance**	MMC1 MMC2 MMC3 MMC4

Table 25-81. MMCHS_SYSCTL (continued)

Description	SD System Control Register This register defines the system controls to set software resets, clock frequency management and data timeout. MMCHS_SYSCTL[31:24] = Software resets MMCHS_SYSCTL[23:16] = Timeout control MMCHS_SYSCTL[15:0] = Clock control
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					SRD	SRC	SRA	RESERVED				DTO				CLKD										CGS	RESERVED		CEN	ICS	ICE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	SRD	Software reset for DAT line This bit is set to 1 for reset and released to 0 when completed. For more information about SRD bit manipulation, see *DATA Lines Reset Procedure*. DAT finite state machine in both clock domain are also reset. Here below are the registers cleared by MMCHS_SYSCTL[SRD]: - MMCHS_DATA - MMCHS_PSTATE: BRE, BWE, RTA, WTA, DLA and DATI - MMCHS_HCTL: SBGR and CR - MMCHS_STAT: BRR, BWR, BGE and TC Interconnect and MMC buffer data management is reinitialized. 0x0: Reset completed 0x1: Software reset for DAT line	RW	0
25	SRC	Software reset for CMD line For more information about SRC bit manipulation, see *CMD Line Reset Procedure*. This bit is set to 1 for reset and released to 0 when completed. CMD finite state machine in both clock domain are also reset. Here below the registers cleared by MMCHS_SYSCTL[SRC]: - MMCHS_PSTATE: CMDI - MMCHS_STAT: CC Interconnect and MMC command status management is reinitialized. 0x0: Reset completed 0x1: Software reset for CMD line	RW	0
24	SRA	Software reset for all This bit is set to 1 for reset , and released to 0 when completed. This reset affects the entire host controller except for the capabilities registers (MMCHS_CAPA and MMCHS_CUR_CAPA). 0x0: Reset completed 0x1: Software reset for all the design	RW	0
23:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19:16	DTO	Data timeout counter value and busy timeout. This value determines the interval by which DAT lines timeouts are detected. The host driver needs to set this bitfield based on - the maximum read access time (NAC) (Refer to the SD Specification Part1 Physical Layer), - the data read access time values (TAAC and NSAC) in the card specific data register (CSD) of the card, - the timeout clock base frequency (MMCHS_CAPA[TCF]). If the card does not respond within the specified number of cycles, a data timeout error occurs (MMCHS_STA[DTO]). The MMCHS_SYSCTL[DTO] register is also used to check busy duration, to generate busy timeout for commands with busy response or for busy programming during a write command. Timeout on CRC status is generated if no CRC token is present after a block write. 0xF: Reserved 0x0: TCF x 2^13 0x1: TCF x 2^14 0xE: TCF x 2^27	RW	0x0
15:6	CLKD	Clock frequency select These bits define the ratio between MMCi_FCLK and the output clock frequency on the CLK pin of either the memory card (MMC, SD or SDIO). 0x0: MMCi_FCLK bypass 0x1: MMCi_FCLK bypass 0x2: MMCi_FCLK / 2 0x3: MMCi_FCLK / 3 0x3FF: MMCi_FCLK / 1023	RW	0x000
5	CGS	Clock Generator Select - For SD cards Host Controller Version 3.00 supports this bit. This bit is used to select the clock generator mode in MMCHS_SYSCTL[15:6] CLKD. If the Programmable Clock Mode is supported (non-zero value is set to MMCHS_CAPA2[23:16] CM), this bit attribute is RW, and if not supported, this bit attribute is RO and zero is read. This bit depends on the setting of MMCHS_AC12[31] PV_ENABLE. If PV_ENABLE = 0, this bit is set by Host Driver. If PV_ENABLE = 1, this bit is automatically set to a value specified in one of Preset Value registers, see, Table 25-22.	R	0
4:3	RESERVED		R	0x0
2	CEN	Clock enable This bit controls if the clock is provided to the card or not. 0x0: The clock is not provided to the card . Clock frequency can be changed . 0x1: The clock is provided to the card and can be automatically gated when MMCHS_SYSCONFIG[AUTOIDLE] is set to 1 (default value) . The host driver shall wait to set this bit to 1 until the Internal clock is stable (MMCHS_SYSCTL[ICS]).	RW	0

Bits	Field Name	Description	Type	Reset
1	ICS	Internal clock stable (status) This bit indicates either the internal clock is stable or not. Read 0x1: The internal clock is stable after enabling the clock (MMCHS_SYSCTL[ICE]) or after changing the clock ratio (MMCHS_SYSCTL[CLKD]). Read 0x0: The internal clock is not stable.	R	0
0	ICE	Internal clock enable This register controls the internal clock activity. In very low power state, the internal clock is stopped. Note: The activity of the debounce clock (used for wakeup events) and the interface clock (used for reads and writes to the module register map) are not affected by this register. 0x0: The internal clock is stopped (very low power state). 0x1: The internal clock oscillates and can be automatically gated when MMCHS_SYSCONFIG[AUTOIDLE] is set to 1 (default value) .	RW	0

Table 25-82. MMCHS_STAT

Address Offset	0x0000 0230		
Physical Address	0x4809 C230 0x480B 4230 0x480A D230 0x480D 1230	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Interrupt Status Register The interrupt status regroups all the status of the module internal events that can generate an interrupt. MMCHS_STAT[31:16] = Error Interrupt Status MMCHS_STAT[15:0] = Normal Interrupt Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		BADA	CERR	RESERVED	TE	ADMAE	ACE	RESERVED	DEB	DCRC	DTO	CIE	CEB	CCRC	CTO

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRI	RESERVED				BSR	OBI	CIRQ	CREM	CINS	BRR	BWR	DMA	BGE	TC	CC

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA	Bad access to data space This bit is set automatically to indicate a bad access to buffer when not allowed: -This bit is set during a read access to the data register (MMCHS_DATA) while buffer reads are not allowed (MMCHS_PSTATE[BRE] =0) -This bit is set during a write access to the data register (MMCHS_DATA) while buffer writes are not allowed (MMCHS_STATE[BWE] =0) Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Bad Access Read 0x0: No Interrupt.	RW	0

Bits	Field Name	Description	Type	Reset
28	CERR	Card error This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E(error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response error MMCHS_CSRE in set. There is no card error detection for autoCMD12 command. The host driver shall read MMCHS_RSP76 register to detect error bits in the command response. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Card error Read 0x0: No Error	RW	0
27	RESERVED		R	0
26	TE	Tuning Error This bit is set when an unrecoverable error is detected in a tuning circuit except during tuning procedure (Occurrence of an error during tuning procedure is indicated by Sampling Select). By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning. To reset tuning circuit, Sampling Clock shall be set to 0 before executing tuning procedure. The Tuning Error is higher priority than the other error interrupts generated during data transfer. By detecting Tuning Error, the Host Driver should discard data transferred by a current read/write command and retry data transfer after the Host Controller retrieved from tuning circuit error. The bit is set if the lock is lost (but not during the tuning process) or if the lock counter expires without the lock being asserted. If the latter happens, the SW can decide to ignore the interrupt and wait some more for the lock to be set. 0x0: No Error 0x1: Error	RW	0
25	ADMAE	ADMA Error: This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register.In addition, the Host Controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: ADMA error Read 0x0: No Interrupt.	RW	0

Bits	Field Name	Description	Type	Reset
24	ACE	Auto CMD error Auto CMD12 and Auto CMD23 use this error status. This bit is set when detecting that one of the bits D00-D04 in Auto CMD Error Status register (MMCHS_AC12) has changed from 0 to 1. In case of Auto CMD12, this bit is set to 1, not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Auto CMD error Read 0x0: No Error.	RW	0
23	RESERVED		R	0
22	DEB	Data End Bit error This bit is set automatically when detecting a 0 at the end bit position of read data on DAT line or at the end position of the CRC status in write mode. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Data end bit error Read 0x0: No Error	RW	0
21	DCRC	Data CRC Error This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Data CRC error Read 0x0: No Error.	RW	0
20	DTO	Data timeout error This bit is set automatically according to the following conditions: - busy timeout for R1b, R5b response type - busy timeout after write CRC status - write CRC status timeout - read data timeout Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Time out Read 0x0: No error.	RW	0
19	CIE	Command index error This bit is set automatically when response index differs from corresponding command index previously emitted. It depends on the enable in MMCHS_CMD[CICE] register. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command index error Read 0x0: No error.	RW	0

Bits	Field Name	Description	Type	Reset
18	CEB	Command end bit error This bit is set automatically when detecting a 0 at the end bit position of a command response. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command end bit error Read 0x0: No error.	RW	0
17	CCRC	Command CRC Error This bit is set automatically when there is a CRC7 error in the command response depending on the enable in MMCHS_CMD[CCCE] register. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command CRC error Read 0x0: No Error.	RW	0
16	CTO	Command Timeout Error This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Time Out Read 0x0: No error	RW	0
15	ERRI	Error Interrupt If any of the bits in the Error Interrupt Status register (MMCHS_STAT[31:16]) are set, then this bit is set to 1. Therefore the host driver can efficiently test for an error by checking this bit first. Writes to this bit are ignored. Read 0x1: Error interrupt event(s) occurred Read 0x0: No Interrupt.	R	0
14:11	RESERVED		R	0x0
10	BSR	Boot status received interrupt This bit is set automatically when MMCHS_CON[BOOT] is set 0x1 or 0x2 and a boot status is received on DAT[0] line. This interrupt is only useful for MMC card. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Boot status received interrupt. Read 0x0: No Interrupt.	RW	0
9	OBI	Out-Of-Band interrupt This bit is set automatically when MMCHS_CON[OBIE] is set and an Out-of-Band interrupt occurs on OBI pin. The interrupt detection depends on polarity controlled by MMCHS_CON[OBIP]. This interrupt is only useful for MMC card. The Out-of-Band interrupt signal is a system specific feature for future use, this signal is not required for existing specification implementation. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Interrupt Out-Of-Band occurs Read 0x0: No Out-Of-Band interrupt.	RW	0

Bits	Field Name	Description	Type	Reset
8	CIRQ	Card interrupt This bit is only used for SD and SDIO and CE-ATA cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wakeup). In 4-bit mode, interrupt source is sampled during the interrupt cycle. In CE-ATA mode, interrupt source is detected when the card drives CMD line to zero during one cycle after data transmission end.All modes above are fully exclusive. The controller interrupt must be clear by setting MMCHS_IE[CIRQ] to 0, then the host driver must start the interrupt service with card (clearing card interrupt status) to remove card interrupt source. Otherwise the Controller interrupt will be reasserted as soon as MMCHS_IE[CIRQ] is set to 1. Writes to this bit are ignored. Read 0x1: Generate card interrupt Read 0x0: No card interrupt	R	0
7	CREM	Card removal This bit is set automatically when MMCHS_PSTATE[CINS] changes from 1 to 0. A clear of this bit doesn't affect Card inserted present state (MMCHS_PSTATE[CINS]). Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Card removed Read 0x0: Card state stable or Debouncing	RW	0
6	CINS	Card insertion This bit is set automatically when MMCHS_PSTATE[CINS] changes from 0 to 1. A clear of this bit doesn't affect Card inserted present state (MMCHS_PSTATE[CINS]). Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Card inserted Read 0x0: Card state stable or debouncing	RW	0
5	BRR	Buffer read ready This bit is set automatically during a read operation to the card (see class 2 - block oriented read commands) when one block specified by MMCHS_BLK[BLEN] is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the local host needs to empty the buffer by reading it. Note: If the DMA receive-mode is enabled, this bit is never set; instead a DMA receive request to the main DMA controller of the system is generated. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Ready to read buffer Read 0x0: Not Ready to read buffer	RW	0

Bits	Field Name	Description	Type	Reset
4	BWR	Buffer write ready This bit is set automatically during a write operation to the card (see class 4 - block oriented write command) when the host can write a complete block as specified by MMCHS_BLK[BLEN]. It indicates that the memory card has emptied one block from the buffer and that the local host is able to write one block of data into the buffer. Note: If the DMA transmit mode is enabled, this bit is never set; instead, a DMA transmit request to the main DMA controller of the system is generated. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Ready to write buffer Read 0x0: Not Ready to write buffer	RW	0
3	DMA	DMA interrupt : This status is set when an interrupt is required in the ADMA instruction and after the data transfer completion. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: No dma interrupt Read 0x0: Dma interrupt detected	RW	0
2	BGE	Block gap event When a stop at block gap is requested (MMCHS_HCTL[SBGR]), this bit is automatically set when transaction is stopped at the block gap during a read or write operation. This event does not occur when the stop at block gap is requested on the last block. In read mode, a 1-to-0 transition of the DAT Line active status (MMCHS_PSTATE[DLA]) between data blocks generates a Block gap event interrupt. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Transaction stopped at block gap Read 0x0: No block gap event	RW	0
1	TC	Transfer completed This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap request (MMCHS_HCTL[SBGR]). In Read mode: This bit is automatically set on completion of a read transfer (MMCHS_PSTATE[RTA]). In write mode: This bit is set automatically on completion of the DAT line use (MMCHS_PSTATE[DLA]). Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Data transfer complete Read 0x0: No transfer complete	RW	0

Bits	Field Name	Description	Type	Reset
0	CC	Command complete This bit is set when a 1-to-0 transition occurs in the register command inhibit (MMCHS_PSTATE[CMDI]) If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command timeout error (MMCHS_STAT[CTO]) has higher priority than command complete (MMCHS_STAT[CC]). If a response is expected but none is received, then a command timeout error is detected and signaled instead of the command complete interrupt. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command complete Read 0x0: No Command complete	RW	0

Table 25-83. MMCHS_IE

Address Offset	0x0000 0234		
Physical Address	0x4809 C234 0x480B 4234 0x480A D234 0x480D 1234	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Interrupt Status Enable Register This register allows to enable/disable the module to set status bits, on an event-by-event basis. MMCHS_IE[31:16] = Error Interrupt Status Enable MMCHS_IE[15:0] = Normal Interrupt Status Enable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		BADA_ENABLE	CERR_ENABLE	RESERVED	TE_ENABLE	ADMAE_ENABLE	ACE_ENABLE	RESERVED	DEB_ENABLE	DCRC_ENABLE	DTO_ENABLE	CIE_ENABLE	CEB_ENABLE	CCRC_ENABLE	CTO_ENABLE	NULL	RESERVED				BSR_ENABLE	OBI_ENABLE	CIRQ_ENABLE	CREM_ENABLE	CINS_ENABLE	BRR_ENABLE	BWR_ENABLE	DMA_ENABLE	BGE_ENABLE	TC_ENABLE	CC_ENABLE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA_ENABLE	Bad access to data space Status Enable 0x0: Masked 0x1: Enabled	RW	0
28	CERR_ENABLE	Card Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
27	RESERVED		R	0
26	TE_ENABLE	Tuning Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
25	ADMAE_ENABLE	ADMA Error Status Enable 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
24	ACE_ENABLE	Auto CMD Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
23	RESERVED		R	0
22	DEB_ENABLE	Data End Bit Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
21	DCRC_ENABLE	Data CRC Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
20	DTO_ENABLE	Data Timeout Error Status Enable 0x0: The data timeout detection is deactivated. The host controller provides the clock to the card until the card sends the data or the transfer is aborted. 0x1: The data timeout detection is enabled.	RW	0
19	CIE_ENABLE	Command Index Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
18	CEB_ENABLE	Command End Bit Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
17	CCRC_ENABLE	Command CRC Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
16	CTO_ENABLE	Command Timeout Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
15	NULL	Fixed to 0 The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored	R	0
14:11	RESERVED		R	0x0
10	BSR_ENABLE	Boot Status Enable A write to this register when MMCHS_CON[BOOT_ACK] is set to 0x0 is ignored. 0x0: Masked 0x1: Enabled	RW	0
9	OBI_ENABLE	Out-of-Band Status Enable A write to this register when MMCHS_CON[OBIE] is set to '0' is ignored. 0x0: Masked 0x1: Enabled	RW	0
8	CIRQ_ENABLE	Card Status Enable A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine doesn't remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
7	CREM_ENABLE	Card Removal Status Enable 0x0: Masked 0x1: Enabled	RW	0
6	CINS_ENABLE	Card Insertion Status Enable 0x0: Masked 0x1: Enabled	RW	0
5	BRR_ENABLE	Buffer Read Ready Status Enable 0x0: Masked 0x1: Enabled	RW	0
4	BWR_ENABLE	Buffer Write Ready Status Enable 0x0: Masked 0x1: Enabled	RW	0
3	DMA_ENABLE	DMA Status Enable 0x0: Masked 0x1: Enabled	RW	0
2	BGE_ENABLE	Block Gap Event Status Enable 0x0: Masked 0x1: Enabled	RW	0
1	TC_ENABLE	Transfer Complete Status Enable 0x0: Masked 0x1: Enabled	RW	0
0	CC_ENABLE	Command Complete Status Enable 0x0: Masked 0x1: Enabled	RW	0

Table 25-84. MMCHS_ISE

Address Offset	0x0000 0238		
Physical Address	0x4809 C238 0x480B 4238 0x480A D238 0x480D 1238	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Interrupt Signal Enable Register This register allows to enable/disable the module internal sources of status, on an event-by-event basis. MMCHS_ISE[31:16] = Error Interrupt Signal Enable MMCHS_ISE[15:0] = Normal Interrupt Signal Enable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		BADA_SIGEN	CERR_SIGEN	RESERVED	TE_SIGEN	ADMAE_SIGEN	ACE_SIGEN	RESERVED	DEB_SIGEN	DCRC_SIGEN	DTO_SIGEN	CIE_SIGEN	CEB_SIGEN	CCRC_SIGEN	CTO_SIGEN	NULL	RESERVED				BSR_SIGEN	OBI_SIGEN	CIRQ_SIGEN	CREM_SIGEN	CINS_SIGEN	BRR_SIGEN	BWR_SIGEN	DMA_SIGEN	BGE_SIGEN	TC_SIGEN	CC_SIGEN

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA_SIGEN	Bad access to data space Signal Enable 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
28	CERR_SIGEN	Card Error Interrupt Signal Enable 0x0: Masked 0x1: Enabled	RW	0
27	RESERVED		R	0
26	TE_SIGEN	Tuning Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
25	ADMAE_SIGEN	ADMA Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
24	ACE_SIGEN	Auto CMD Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
23	RESERVED		R	0
22	DEB_SIGEN	Data End Bit Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
21	DCRC_SIGEN	Data CRC Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
20	DTO_SIGEN	Data Timeout Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
19	CIE_SIGEN	Command Index Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
18	CEB_SIGEN	Command End Bit Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
17	CCRC_SIGEN	Command CRC Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
16	CTO_SIGEN	Command timeout Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
15	NULL	Fixed to 0 The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored	R	0
14:11	RESERVED		R	0x0
10	BSR_SIGEN	Boot Status Signal Enable A write to this register when MMCHS_CON[BOOT_ACK] is set to 0x0 is ignored. 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
9	OBI_SIGEN	Out-Of-Band Interrupt Signal Enable A write to this register when MMCHS_CON[OBIE] is set to '0' is ignored. 0x0: Masked 0x1: Enabled	RW	0
8	CIRQ_SIGEN	Card Interrupt Signal Enable 0x0: Masked 0x1: Enabled	RW	0
7	CREM_SIGEN	Card Removal Signal Enable 0x0: Masked 0x1: Enabled	RW	0
6	CINS_SIGEN	Card Insertion Signal Enable 0x0: Masked 0x1: Enabled	RW	0
5	BRR_SIGEN	Buffer Read Ready Signal Enable 0x0: Masked 0x1: Enabled	RW	0
4	BWR_SIGEN	Buffer Write Ready Signal Enable 0x0: Masked 0x1: Enabled	RW	0
3	DMA_SIGEN	DMA Interrupt Signal Enable 0x0: Masked 0x1: Enabled	RW	0
2	BGE_SIGEN	Black Gap Event Signal Enable 0x0: Masked 0x1: Enabled	RW	0
1	TC_SIGEN	Transfer Completed Status Enable 0x0: Masked 0x1: Enabled	RW	0
0	CC_SIGEN	Command Complete Status Enable 0x0: Masked 0x1: Enabled	RW	0

Table 25-85. MMCHS_AC12

Address Offset	0x0000 023C		
Physical Address	0x4809 C23C 0x480B 423C 0x480A D23C 0x480D 123C	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Host Control 2 Register and Auto CMD Error Status Register This register is used to indicate CMD12 response error of Auto CMD12 and CMD23 response error of Auto CMD23. The Host driver can determine what kind of Auto CMD12 / CMD23 errors occur by this register. Auto CMD23 errors are indicated only in bits[4:1]. Bits[7:0] are valid only when the MMCHS_CMD[3:2] ACEN bitfield is configured to enable Auto CMD and the Auto CMD Error bit (MMCHS_STAT[24]ACE) is set.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PV_ENABLE	AI_ENABLE	RESERVED	SCLK_SEL	ET	DS_SEL	V18_SIGEN	UHSMS	RESERVED	CNI	RESERVED	ACIE	ACEB	ACCE	ACTO	ACNE

Bits	Field Name	Description	Type	Reset
31	PV_ENABLE	Preset Value Enable Host Controller Version 3.00 supports this bit. As the operating SDCLK frequency and I/O driver strength depend on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When Preset Value Enable is set, automatic SDCLK frequency generation and driver strength selection is performed without considering system specific conditions. This bit enables the functions defined in the Preset Value registers, see, Table 25-22. If this bit is set to 0, MMCHS_SYSCTL[15:6] CLKD, MMCHS_SYSCTL[5] CGS and MMCHS_AC12[21:20] DS_SEL are set by Host Driver. If this bit is set to 1, MMCHS_SYSCTL[15:6] CLKD, MMCHS_SYSCTL[5] CGS and MMCHS_AC12[21:20] DS_SEL are set by Host Controller as specified in the Preset Value registers, see, Table 25-22. 0x0: SDCLK and Driver Strength (DS_SEL) are controlled by Host Driver. 0x1: Automatic Selection by Preset Value are Enabled.	RW	0
30	AI_ENABLE	Asynchronous Interrupt Enable This bit can be set to 1 if a card supports asynchronous interrupts and MMCHS_CAPA[29] AIS is set to 1. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode (and zero is set to Interrupt Pin Select in the Shared Bus Control register). If this bit is set to 1, the Host Driver can stop the SDCLK during asynchronous interrupt period to save power. During this period, the Host Controller continues to deliver the Card Interrupt to the host when it is asserted by the Card. 0x0: Disabled 0x1: Enabled	RW	0
29:24	RESERVED		R	0x00
23	SCLK_SEL	Sampling Clock Select Host Controller uses this bit to select sampling clock to receive CMD and DAT. This bit is set by tuning procedure and valid after the completion of tuning (when MMCHS_AC12[22] ET is cleared). Setting 1 means that tuning is completed successfully and setting 0 means that tuning is failed. Writing 1 to this bit is meaningless and ignored. A tuning circuit is reset by writing to 0. This bit can be cleared with setting MMCHS_AC12[22] ET. Once the tuning circuit is reset, it will take time to complete tuning sequence. Therefore, Host Driver should keep this bit to 1 to perform re-tuning sequence to compete re-tuning sequence in a short time. Change of this bit is not allowed while the Host Controller is receiving response or a read data block. 0x0: Fixed clock is used to sample data 0x1: Tuned clock is used to sample data	RW	0

Bits	Field Name	Description	Type	Reset
22	ET	Execute Tuning This bit is set to 1 to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to MMCHS_AC12[23] SCLK_SEL. Tuning procedure is aborted by writing 0. This is Read-Write with automatic clear register 0x0: Not Tuned or Tuning Completed 0x1: Execute Tuning	RW	0
21:20	DS_SEL	Driver Strength Select Host Controller output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set depending on Driver Type A, C and D support bits (DTA, DTC and DTD respectively) in the MMCHS_CAPA2 register. This bit depends on setting of Preset Value Enable. If Preset Value Enable = 0, this field is set by Host Driver. If Preset Value Enable = 1, this field is automatically set by a value specified in the one of Preset Value registers, see, Table 25-22. 0x0: Driver Type B is selected (Default) 0x1: Driver Type A is selected 0x3: Driver Type D is selected 0x2: Driver Type C is selected	RW	0x0
19	V1V8_SIGEN	1.8V Signaling Enable This bit controls voltage regulator for I/O cell. 3.3V is supplied to the card regardless of signaling voltage. Setting this bit from 0 to 1 starts changing signal voltage from 3.3V to 1.8V. 1.8V regulator output shall be stable within 5ms. Host Controller clears this bit if switching to 1.8V signaling fails. Clearing this bit from 1 to 0 starts changing signal voltage from 1.8V to 3.3V. 3.3V regulator output shall be stable within 5ms. Host Driver can set this bit to 1 when Host Controller supports 1.8V signaling (One of support bits is set to 1: SDR50, SDR104 or DDR50 in MMCHS_CAPA2 register) and the card or device supports UHS-I (S18A=1. Refer to Bus Signal Voltage Switch Sequence in the Physical Layer Specification Version 3.0x). 0x0: 3.3V Signaling 0x1: 1.8V Signaling	RW	0

Bits	Field Name	Description	Type	Reset
18:16	UHSMS	UHS Mode Select This field is used to select one of UHS-I modes or eMMC HS200 mode and is effective when 1.8V Signaling Enable is set to 1. If MMCHS_AC12[31] PV_ENABLE is set to 1, Host Controller sets MMCHS_SYSCTL[15:6] CLKD, MMCHS_SYSCTL[5] CGS and MMCHS_AC12[21:20] DS_SEL according to Preset Value registers, see, Table 25-22. In this case, one of preset value registers is selected by this field. Host Driver needs to reset MMCHS_SYSCTL[2] CEN before changing this field to avoid generating clock glitch. After setting this field, Host Driver sets MMCHS_SYSCTL[2] CEN again. When SDR50, SDR104 or DDR50 is selected for SDIO card, interrupt detection at the block gap shall not be used. Read Wait timing is changed for these modes. Refer to the SDIO Specification Version 3.00 for more detail. 0x0: SDR12 0x1: SDR25 0x2: SDR50 0x3: SDR104/HS200 0x4: DDR50 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x0
15:8	RESERVED		R	0x00
7	CNI	Command Not Issued By Auto CMD12 Error Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. Read 0x1: Command not issued Read 0x0: No error	R	0
6:5	RESERVED		R	0x0
4	ACIE	Auto CMD Index Error - For Auto CMD12 and Auto CMD23 This bit is set if the Command Index error occurs in response to a command. Read 0x1: Error Read 0x0: No error	R	0
3	ACEB	Auto CMD End Bit Error - For Auto CMD12 and Auto CMD23 This bit is set when detecting that the end bit of command response is 0. Read 0x1: End bit Error Generated Read 0x0: No error	R	0
2	ACCE	Auto CMD CRC Error - For Auto CMD12 and Auto CMD23 This bit is set when detecting a CRC error in the command response. Read 0x1: CRC Error Generated Read 0x0: No error	R	0

Bits	Field Name	Description	Type	Reset
1	ACTO	Auto CMD Timeout Error - For Auto CMD12 and Auto CMD23 This bit is set if no response is returned within 64 SDCLK cycles from the end bit of command. If this bit is set to1, the other error status bits (D04-D02) are meaningless. Read 0x1: Auto CMD Time Out Read 0x0: No error	R	0
0	ACNE	Auto CMD12 Not Executed If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. Read 0x1: Auto CMD12 Not Executed Read 0x0: Auto CMD12 Executed	R	0

Table 25-86. MMCHS_CAPA

Address Offset	0x0000 0240		
Physical Address	0x4809 C240 0x480B 4240 0x480A D240 0x480D 1240	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Capabilities Register This register lists the capabilities of the MMC/SD/SDIO host controller.		
Type	RW		

31:30	29	28	27	26	25	24	23	22	21	20	19	18	17:16	15:8	7	6	5:0
RESERVED	AIS	BIT64	RESERVED	VS18	VS30	VS33	SRS	DS	HSS	RESERVED	AD2S	RESERVED	MBL	BCF	TCU	RESERVED	TCF

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	AIS	Asynchronous Interrupt Support Refer to SDIO Specification Version 3.00 about asynchronous interrupt. Read 0x1: Asynchronous Interrupt Supported Read 0x0: Asynchronous Interrupt Not Supported	R	1
28	BIT64	64 Bit System Bus Support Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus. Read 0x1: 64 bit System bus address Read 0x0: 32 bit System bus address	R	0
27	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
26	VS18	Voltage support 1.8V Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal) Write 0x0: 1.8V Not supported Write 0x1: 1.8V Supported Read 0x1: 1.8V Supported Read 0x0: 1.8V Not Supported	RW	0
25	VS30	Voltage support 3.0V Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal) Write 0x0: 3.0V Not supported Write 0x1: 3.0V Supported Read 0x1: 3.0V Supported Read 0x0: 3.0V Not Supported	RW	0
24	VS33	Voltage support 3.3V Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal) Write 0x0: 3.3V Not supported Write 0x1: 3.3V Supported Read 0x1: 3.3V Supported Read 0x0: 3.3V Not Supported	RW	0
23	SRS	Suspend/Resume support (SDIO cards only) This bit indicates whether the host controller supports Suspend/Resume functionality. Read 0x1: The Host controller supports Suspend/ Resume functionality. Read 0x0: The Host controller does not Suspend/ Resume functionality.	R	1
22	DS	DMA support This bit indicates that the Host Controller is able to use DMA to transfer data between system memory and the Host Controller directly. Read 0x1: DMA Supported Read 0x0: DMA Not Supported	R	1
21	HSS	High speed support This bit indicates that the host controller supports high speed operations and can supply an up-to maximum card frequency. Read 0x1: High Speed Supported Read 0x0: High Speed Not Supported **NOTE: High Speed modes are supported, but MMCHS_HCTL[HSPE] bit must always be set to 0x0 because device was timing closed with HSPE bit set to 0x0 for all supported modes of operation.**	R	1
20	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
19	AD2S	ADMA2 Support This bit indicates whether the Host Controller is capable of using ADMA2. It depends on setting of generic parameter MADMA_EN Read 0x1: ADMA2 Supported Read 0x0: ADMA2 not Supported	R	0
18	RESERVED		R	0
17:16	MBL	Maximum block length This value indicates the maximum block size that the host driver can read and write to the buffer in the host controller. This value depends on definition of generic parameter with a max value of 2048 bytes. The host controller supports 512 bytes and 1024 bytes block transfers. Read 0x2: 2048 bytes Read 0x1: 1024 bytes Read 0x0: 512 bytes	R	0x1
15:8	BCF	Base Clock Frequency For SD Clock This value indicates the base (maximum) clock frequency for the SD Clock. 8-bit Base Clock Frequency This mode is supported by the Host Controller Version 3.00. Unit values are 1MHz. The supported clock range is 10MHz to 255MHz. FFh : 255MHz .... : .... 02h : 2MHz 01h : 1MHz 00h : Get information via another method If the real frequency is 16.5MHz, the lager value shall be set 0001 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to MMCHS_SYSCTL[15:6] CLKD) and it shall not exceed upper limit of the SD Clock frequency. If these bits are all 0, the Host System has to get information via another method. Read 0x0: The value indicating the base (maximum) frequency for the output clock provided to the card is system dependent and is not available in this register. Get the information via another method.	R	0x00
7	TCU	Timeout clock unit This bit shows the unit of base clock frequency used to detect Data Timeout Error (MMCHS_STAT[DTO]). Read 0x1: MHz Read 0x0: KHz	R	1
6	RESERVED		R	0
5:0	TCF	Timeout clock frequency The timeout clock frequency is used to detect Data Timeout Error (MMCHS_STAT[DTO]). Read 0x0: The timeout clock frequency depends on the frequency of the clock provided to the card. The value of the timeout clock frequency is not available in this register.	R	0x00

Table 25-87. MMCHS_CAPA2

Address Offset	0x0000 0244		
Physical Address	0x4809 C244 0x480B 4244 0x480A D244 0x480D 1244	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Capabilities 2 Register This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller may implement these values as fixed or loaded from flash memory during power on initialization. Refer to Software Reset For All in the Software Reset register for loading from flash memory and completion timing control.		
Type	R		

31:24	23:16	15:14	13	12	11:8	7	6	5	4	3	2	1	0
RESERVED	CM	RTM	TSDR50	RESERVED	TCRT	RESERVED	DTD	DTC	DTA	RESERVED	DDR50	SDR104	SDR50

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	CM	Clock Multiplier This field indicates clock multiplier value of programmable clock generator. Refer to MMCHS_SYSCTL [15:0]. Setting 00h means that Host Controller does not support programmable clock generator. 00h : Clock Multiplier is Not Supported 01h : Clock Multiplier M = 2 02h : Clock Multiplier M = 3 .... : .... FFh : Clock Multiplier M = 256	R	0x00

Bits	Field Name	Description	Type	Reset
15:14	RTM	Re-Tuning Modes This field selects re-tuning method and limits the maximum data length. Bit47-46 Re-Tuning Mode Re-Tuning Method Data Length There are two re-tuning timings: Re-Tuning Request controlled by the Host Controller and expiration of a Re-Tuning Timer controlled by the Host Driver. By receiving either timing, the Host Driver executes the re-tuning procedure just before a next command issue. The maximum data length per read/write command is restricted so that re-tuning procedures can be inserted during data transfers. (1) Re-Tuning Mode 1 The host controller does not have any internal logic to detect when the re-tuning needs to be performed. In this case, the Host Driver should maintain all re-tuning timings by using a Re-Tuning Timer. To enable inserting the re-tuning procedure during data transfers, the data length per read/write command shall be limited up to 4 MiB. (2) Re-Tuning Mode 2 The host controller has the capability to indicate the re-tuning timing by Re-Tuning Request during data transfers. Then the data length per read/write command shall be limited up to 4 MiB. During non data transfer, re-tuning timing is determined by either Re-Tuning Request or Re-Tuning Timer. If Re-Tuning Request is used, Re-Tuning Timer should be disabled. (3) Re-Tuning Mode 3 The host controller has the capability to take care of the re-tuning during data transfer (Auto Re-Tuning). Re-Tuning Request shall not be generated during data transfers and there is no limitation to data length per read/write command. During non data transfer, re-tuning timing is determined by either Re-Tuning Request or Re-Tuning Timer. If Re-Tuning Request is used, Re-Tuning Timer should be disabled. Re-Tuning Timer Control Example for Re-Tuning Mode 1 The initial value of re-tuning timer is provided by Timer Count for Re-Tuning field in this register. The timer starts counting by loading the initial value. When the timer expires, the Host Driver marks an expiration flag. On receiving a command request, the Host driver checks the expiration flag. If the expiration flag is set, then the Host Driver should perform the re-tuning procedure before issuing a command. If the expiration flag is not set, then the Host Driver issues a command without performing the re-tuning procedure. Every time the re-tuning procedure is performed, the timer loads the new initial value and the expiration flag is cleared. Re-Tuning Timer Control Example for Re-Tuning Mode 2 and Mode 3 The timer control is almost the same as Re-Tuning Mode 1 except the timer loads the new initial value after data transfer (when receiving Transfer Complete). In case of Mode 3, Timer Count for Re-Tuning is set either smaller value: Tuning effective time after re-tuning procedure or after data transfer. If a Host System goes into power down mode, the Host Driver should stop the re-tuning timer and set the expiration flag to 1 when the Host System resumes from power down mode. Read 0x3: Reserved Read 0x2: Auto Re-Tuning (for transfer) - Timer and Re-Tuning Request Read 0x1: Timer and Re-Tuning Request - Max data length 4 MiB Read 0x0: Timer - Max data length 4 MiB	R	0x0

Bits	Field Name	Description	Type	Reset
13	TSDR50	Use Tuning for SDR50 If this bit is set to 1, this Host Controller requires tuning to operate SDR50. (Tuning is always required to operate SDR104.) Read 0x1: SDR50 requires tuning. Read 0x0: SDR50 does not require tuning.	R	0
12	RESERVED		R	0
11:8	TCRT	Timer Count for Re-Tuning This field indicates an initial value of the Re-Tuning Timer for Re-Tuning Mode 1 to 3. Setting to 0 disables Re-Tuning Timer. Read 0x3: 4 seconds Read 0xE: Reserved Read 0xC: Reserved Read 0x4: 8 seconds Read 0xB: 1024 seconds Read 0xF: Get information from other source Read 0x2: 2 seconds Read 0x0: Re-Tuning Timer disabled Read 0xA: 512 seconds Read 0x6: 32 seconds Read 0x1: 1 second Read 0x8: 128 seconds Read 0x7: 64 seconds Read 0x9: 256 seconds Read 0xD: Reserved Read 0x5: 16 seconds	R	0xF
7	RESERVED		R	0
6	DTD	Driver Type D Support This bit indicates support of Driver Type D for 1.8 Signaling. Read 0x1: Driver Type D is Supported Read 0x0: Driver Type D is Not Supported.	R	1
5	DTC	Driver Type C Support This bit indicates support of Driver Type C for 1.8 Signaling. Read 0x1: Driver Type C is Supported. Read 0x0: Driver Type C is Not Supported.	R	1
4	DTA	Driver Type A Support This bit indicates support of Driver Type A for 1.8 Signaling. Read 0x1: Driver Type A is Supported. Read 0x0: Driver Type A is Not Supported.	R	1
3	RESERVED		R	0
2	DDR50	DDR50 Support Read 0x1: DDR50 is Supported. Read 0x0: DDR50 is Not Supported.	R	1(1)
1	SDR104	SDR104 Support SDR104 requires tuning. Read 0x1: SDR104 is Supported. Read 0x0: SDR104 is Not Supported.	R	1(1)

Bits	Field Name	Description	Type	Reset
0	SDR50	SDR50 Support If SDR104 is supported, this bit shall be set to 1. Bit 13 indicates whether SDR50 requires tuning or not. Read 0x1: SDR50 is Supported. Read 0x0: SDR50 is Not Supported.	R	1[(1)]

(1) This bit is only supported by the MMC modules listed in Table 25-88. The value should be ignored for any modules not listed.

Table 25-88. Supported Data Rate Modes

Supported Mode	Supported By
DDR50	MMC1
SDR104	MMC1
SDR50	MMC1, MMC3

Table 25-89. MMCHS_CUR_CAPA

Address Offset	0x0000 0248		
Physical Address	0x4809 C248 0x480B 4248 0x480A D248 0x480D 1248	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Maximum Current Capabilities Register This register indicates the maximum current capability for each voltage. The value is meaningful if the voltage support is set in the capabilities register (MMCHS_CAPA). Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CUR_1V8								CUR_3V0								CUR_3V3							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	CUR_1V8	Maximum current for 1.8V Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00
15:8	CUR_3V0	Maximum current for 3.0V Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00
7:0	CUR_3V3	Maximum current for 3.3V Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00

Table 25-90. MMCHS_FE

Address Offset	0x0000 0250		
Physical Address	0x4809 C250 0x480B 4250 0x480A D250 0x480D 1250	**Instance**	MMC1 MMC2 MMC3 MMC4

Table 25-90. MMCHS_FE (continued)

Description	Force Event Register for Auto CMD Error Status and Error Interrupt status The Force Event Register is not a physically implemented register. Rather, it is an address at which the Auto CMD Error Status Register (MMCHS_AC12) can be written. Writing 1 : set each bit of the Auto CMD Error Status Register Writing 0 : no effect Rather, it is an address at which the Error Interrupt Status register can be written. The effect of a write to this address will be reflected in the Error Interrupt Status Register if the corresponding bit of the Error Interrupt Status Enable Register is set. Writing 1 : set each bit of the Error Interrupt Status Register Writing 0 : no effect Note: By setting this register, the Error Interrupt can be set in the Error Interrupt Status register. In order to generate interrupt signal, both the Error Interrupt Status Enable and Error Interrupt Signal Enable shall be set.
Type	W

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td>FE_BADA</td><td>FE_CERR</td><td colspan="2">RESERVED</td><td>FE_ADMAE</td><td>FE_ACE</td><td>RESERVED</td><td>FE_DEB</td><td>FE_DCRC</td><td>FE_DTO</td><td>FE_CIE</td><td>FE_CEB</td><td>FE_CCRC</td><td>FE_CTO</td><td colspan="8">RESERVED</td><td>FE_CNI</td><td colspan="2">RESERVED</td><td>FE_ACIE</td><td>FE_ACEB</td><td>FE_ACCE</td><td>FE_ACTO</td><td>FE_ACNE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		NA	0x0
29	FE_BADA	Force Event Bad access to data space. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
28	FE_CERR	Force Event Card error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
27:26	RESERVED		NA	0x0
25	FE_ADMAE	Force Event ADMA Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
24	FE_ACE	Force Event for Auto CMD Error - For Auto CMD12 and Auto CMD23 Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
23	RESERVED		NA	0
22	FE_DEB	Force Event Data End Bit error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
21	FE_DCRC	Force Event Data CRC Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
20	FE_DTO	Force Event Data Timeout Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
19	FE_CIE	Force Event Command Index Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0

Bits	Field Name	Description	Type	Reset
18	FE_CEB	Force Event Command End Bit Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
17	FE_CCRC	Force Event Command CRC Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
16	FE_CTO	Command Timeout Error This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles. Write 0x0: Status bit unchanged Write 0x1: Status is cleared	W	0
15:8	RESERVED		NA	0x00
7	FE_CNI	Force Event Command not issue by Auto CMD12 error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
6:5	RESERVED		NA	0x0
4	FE_ACIE	Force Event for Auto CMD Index Error - For Auto CMD12 and Auto CMD23 Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
3	FE_ACEB	Force Event Auto CMD End Bit Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
2	FE_ACCE	Force Event Auto CMD CRC Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
1	FE_ACTO	Force Event Auto CMD Timeout Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
0	FE_ACNE	Force Event Auto CMD12 Not Executed Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0

Table 25-91. MMCHS_ADMAES

Address Offset	0x0000 0254		
Physical Address	0x4809 C254 0x480B 4254 0x480A D254 0x480D 1254	**Instance**	MMC1 MMC2 MMC3 MMC4

Table 25-91. MMCHS_ADMAES (continued)

Description	ADMA Error Status Register When ADMA Error Interrupt is occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows: ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address ST_FDS: Current location set in the ADMA System Address register is the error descriptor address ST_CADR: This sate is never set because do not generate ADMA error in this state. ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address In case of write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller. The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver may find that the Valid bit is not set in the error descriptor.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													LME	AES	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	LME	ADMA Length Mismatch Error: (1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. (2) Total data length can not be divided by the block length. 0x0: No Error 0x1: Error	RW	0
1:0	AES	ADMA Error State This field indicates the state of ADMA when error occurred during ADMA data transfer. This field will never be 0x2 because ADMA never stops in that state. 0x0: ST_STOP (STOP_ADMA). Previous SYS_ADR is the error descriptor address 0x1: ST_FDS (Fetch Descriptor). Content of current SYS_ADR is the error descriptor address 0x2: Not used. Error never set in this state 0x3: ST_TFR (Transfer Data). Previous SYS_ADR is the error descriptor address	RW	0x0

Table 25-92. MMCHS_ADMASAL

Address Offset	0x0000 0258		
Physical Address	0x4809 C258 0x480B 4258 0x480A D258 0x480D 1258	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	ADMA System address Low bits		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADMA_A32B																															

Bits	Field Name	Description	Type	Reset
31:0	ADMA_A32B	ADMA System address 32 bits.This register holds byte address of executing command of the Descriptor table. 32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold valid Descriptor address depending on the ADMA state. The Host Driver shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 00b.	RW	0x0000 0000

Table 25-93. MMCHS_PVINITSD

Address Offset	0x0000 0260		
Physical Address	0x4809 C260 0x480B 4260 0x480A D260 0x480D 1260	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Preset Value for Initialization and Default Speed modes		
Type	R		

31:30	29:27	26	25:16	15:14	13:11	10	9:0
DSDS_SEL	RESERVED	DSCLKGEN_SEL	DSSDCLK_SEL	INITDS_SEL	RESERVED	INITCLKGEN_SEL	INITSDCLK_SEL

Bits	Field Name	Description	Type	Reset
31:30	DSDS_SEL	Driver Strength Select Value - Default Speed mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
29:27	RESERVED		R	0x0
26	DSCLKGEN_SEL	Clock Generator Select Value - Default Speed mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
25:16	DSSDCLK_SEL	SDCLK Frequency Select Value - Default Speed mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x004
15:14	INITDS_SEL	Driver Strength Select Value - Initialization mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected Read 0x2: Driver Type C is Selected Read 0x1: Driver Type A is Selected Read 0x0: Driver Type B is Selected	R	0x0

Bits	Field Name	Description	Type	Reset
13:11	RESERVED		R	0x0
10	INITCLKGEN_SEL	Clock Generator Select Value - Initialization mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
9:0	INITSDCLK_SEL	SDCLK Frequency Select Value - Initialization mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x1E0

Table 25-94. MMCHS_PVHSSDR12

Address Offset	0x0000 0264		
Physical Address	0x4809 C264 0x480B 4264 0x480A D264 0x480D 1264	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Preset Value for High Speed and SDR12 speed modes		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">SDR12DS_SEL</td><td colspan="3">RESERVED</td><td>SDR12CLKGEN_SEL</td><td colspan="10">SDR12SDCLK_SEL</td><td colspan="2">HSDS_SEL</td><td colspan="3">RESERVED</td><td>HSCLKGEN_SEL</td><td colspan="10">HSSDCLK_SEL</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:30	SDR12DS_SEL	Driver Strength Select Value - SDR12 mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
29:27	RESERVED		R	0x0
26	SDR12CLKGEN_SEL	Clock Generator Select Value - SDR12 mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
25:16	SDR12SDCLK_SEL	SDCLK Frequency Select Value - SDR12 mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x004
15:14	HSDS_SEL	Driver Strength Select Value - High Speed mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0

Bits	Field Name	Description	Type	Reset
13:11	RESERVED		R	0x0
10	HSCLKGEN_SEL	Clock Generator Select Value - High Speed mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
9:0	HSSDCLK_SEL	SDCLK Frequency Select Value - High Speed mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x002

Table 25-95. MMCHS_PVSDR25SDR50

Address Offset	0x0000 0268		
Physical Address	0x4809 C268 0x480B 4268 0x480A D268 0x480D 1268	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Preset Value for SDR25 and SDR50 speed modes		
Type	R		

31 30	29 28 27	26	25 24 23 22 21 20 19 18 17 16	15 14	13 12 11	10	9 8 7 6 5 4 3 2 1 0
SDR50DS_SEL	RESERVED	SDR50CLKGEN_SEL	SDR50SDCLK_SEL	SDR25DS_SEL	RESERVED	SDR25CLKGEN_SEL	SDR25SDCLK_SEL

Bits	Field Name	Description	Type	Reset
31:30	SDR50DS_SEL	Driver Strength Select Value - SDR50 mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
29:27	RESERVED		R	0x0
26	SDR50CLKGEN_SEL	Clock Generator Select Value - SDR50 mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
25:16	SDR50SDCLK_SEL	SDCLK Frequency Select Value - SDR50 mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x001
15:14	SDR25DS_SEL	Driver Strength Select Value - SDR25 mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0

Bits	Field Name	Description	Type	Reset
13:11	RESERVED		R	0x0
10	SDR25CLKGEN_SEL	Clock Generator Select Value - SDR25 mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generato. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
9:0	SDR25SDCLK_SEL	SDCLK Frequency Select Value - SDR25 mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x002

Table 25-96. MMCHS_PVSDR104DDR50

Address Offset	0x0000 026C		
Physical Address	0x4809 C26C 0x480B 426C 0x480A D26C 0x480D 126C	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Preset Value for SDR104 and DDR50 speed modes		
Type	R		

31:30	29:27	26	25:16	15:14	13:11	10	9:0
DDR50DS_SEL	RESERVED	DDR50CLKGEN_SEL	DDR50SDCLK_SEL	SDR104DS_SEL	RESERVED	SDR104CLKGEN_SEL	SDR104SDCLK_SEL

Bits	Field Name	Description	Type	Reset
31:30	DDR50DS_SEL	Driver Strength Select Value - DDR50 mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
29:27	RESERVED		R	0x0
26	DDR50CLKGEN_SEL	Clock Generator Select Value - DDR50 mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator Read 0x0: Host Controller Ver2.00 Compatible Clock Generator	R	0
25:16	DDR50SDCLK_SEL	SDCLK Frequency Select Value - DDR50 mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x002

Bits	Field Name	Description	Type	Reset
15:14	SDR104DS_SEL	Driver Strength Select Value - SDR104 mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
13:11	RESERVED		R	0x0
10	SDR104CLKGEN_SEL	Clock Generator Select Value - SDR104 mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
9:0	SDR104SDCLK_SEL	SDCLK Frequency Select Value - SDR104 mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x000

Table 25-97. MMCHS_REV

Address Offset	0x0000 02FC		
Physical Address	0x4809 C2FC 0x480B 42FC 0x480A D2FC 0x480D 12FC	**Instance**	MMC1 MMC2 MMC3 MMC4
Description	Versions Register This register contains the hard coded RTL vendor revision number, the version number of SD specification compliancy and a slot status bit. MMCHS_REV[31:16] = Host controller version MMCHS_REV[15:0] = Slot Interrupt Status		
Type	R		

31–24	23–16	15–1	0
VREV	SREV	RESERVED	SIS

Bits	Field Name	Description	Type	Reset
31:24	VREV	Vendor Version Number: IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0 0x21 for 2.1	R	0x--
23:16	SREV	Specification Version Number This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version. Read 0x3: Reserved Read 0x2: SD Host Specification Version 3.00. Read 0x1: SD Host Specification Version 2.00 - Including the feature of the ADMA and Test Register. Read 0x0: SD Host Specification Version 1.00.	R	0x02
15:1	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
0	SIS	Slot Interrupt Status This status bit indicates the inverted state of interrupt signal for the module. By a power on reset or by setting a software reset for all (MMCHS_HCTL[SRA]), the interrupt signal shall be de-asserted and this status shall read 0.	R	0

Chapter 26

Shared PHY Component Subsystem

This chapter describes the shared PHY component subsystems of the device.

26.1 SATA PHY Subsystem 6520
26.2 USB3_PHY Subsystem 6539
26.3 USB3 PHY and SATA PHY Register Manual 6558
26.4 PCIe PHY Subsystem 6577

26.1 SATA PHY Subsystem

This chapter describes the features and functions of the serial advanced technology attachment (SATA) PHY subsystem of the device.

26.1.1 SATA PHY Subsystem Overview

The physical layer (PHY) is responsible for transmitting and receiving the parallel 8b/10b encoded information as a serial data stream on the wire.

The SATA host controller subsystem instantiates a single serializer (transmitter), SATA_PHY_TX, and a single deserializer (receiver), SATA_PHY_RX. Together the transmitter and receiver are also signified as SATA_PHY throughout this chapter. The role of the TX and RX components is to adapt SATA Link parallel 10-bit input/output (I/O) data stream for a serialized differential transmission and reception over SATA electrical interface, respectively.

The high speed transmission clock is generated by and integrated into the SATA host subsystem dpll (DPLL_SATA).

The DPLL_SATA is configured and controlled through a SATA dedicated PLL controller (DPLLCTRL_SATA) with associated registers, accessible over a L4_CFG interface adapter.

The components (SATA_PHY_RX, SATA_PHY_TX, DPLL_SATA, DPLLCTRL_SATA, and DPLLCTRL_SATA L4-interface adapter) build the SATA PHY subsystem. This subsystem is responsible for PHY components clock generation and physical layer transmission/reception within the device SATA subsystem.

Figure 26-1 gives an overview of the SATA PHY subsystem. As shown in Figure 26-1, at one side the SATA_PHY components directly interface the attached to host controller SATA mass storage device (over TXP/TXN transmission and RXP/RXN reception interface I/Os) and on the other side they interface the SATA controller, described in detail in Section 24.8, *SATA Controller*.

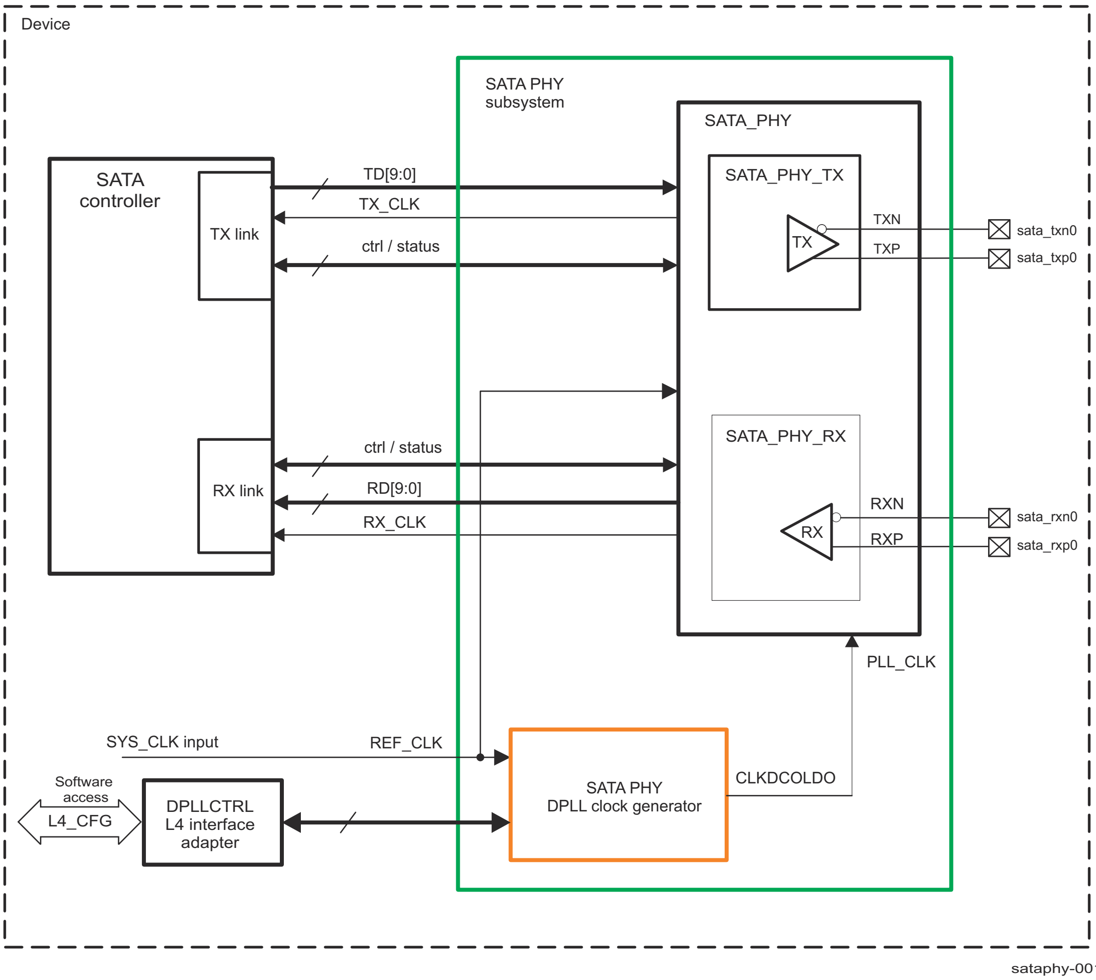


Figure 26-1. SATA PHY Subsystem Overview

26.1.2 SATA PHY Subsystem Environment

26.1.2.1 SATA PHY I/O Signals

Table 26-1 lists the module pins and their corresponding signal names at the device level, and also specifies their links to functions.

Table 26-1. SATA PHY I/O Signals

Module Signal	Device Pin	I/O[(1)]	Description	Pin Reset Value
TXP	sata_txp0	O	TXP output of the SATA PHY differential transmission lane	HiZ
TXN	sata_txn0	O	TXN output of the SATA PHY differential transmission lane	HiZ
RXP	sata_rxp0	I	RXP input of the SATA PHY differential reception lane	HiZ
RXN	sata_rxn0	I	RXN input of the SATA PHY differential reception lane	HiZ

(1) I = Input; O = Output

Figure 26-2 shows module pin signals mapping to SATA PHY I/O signals visible at device pad level.

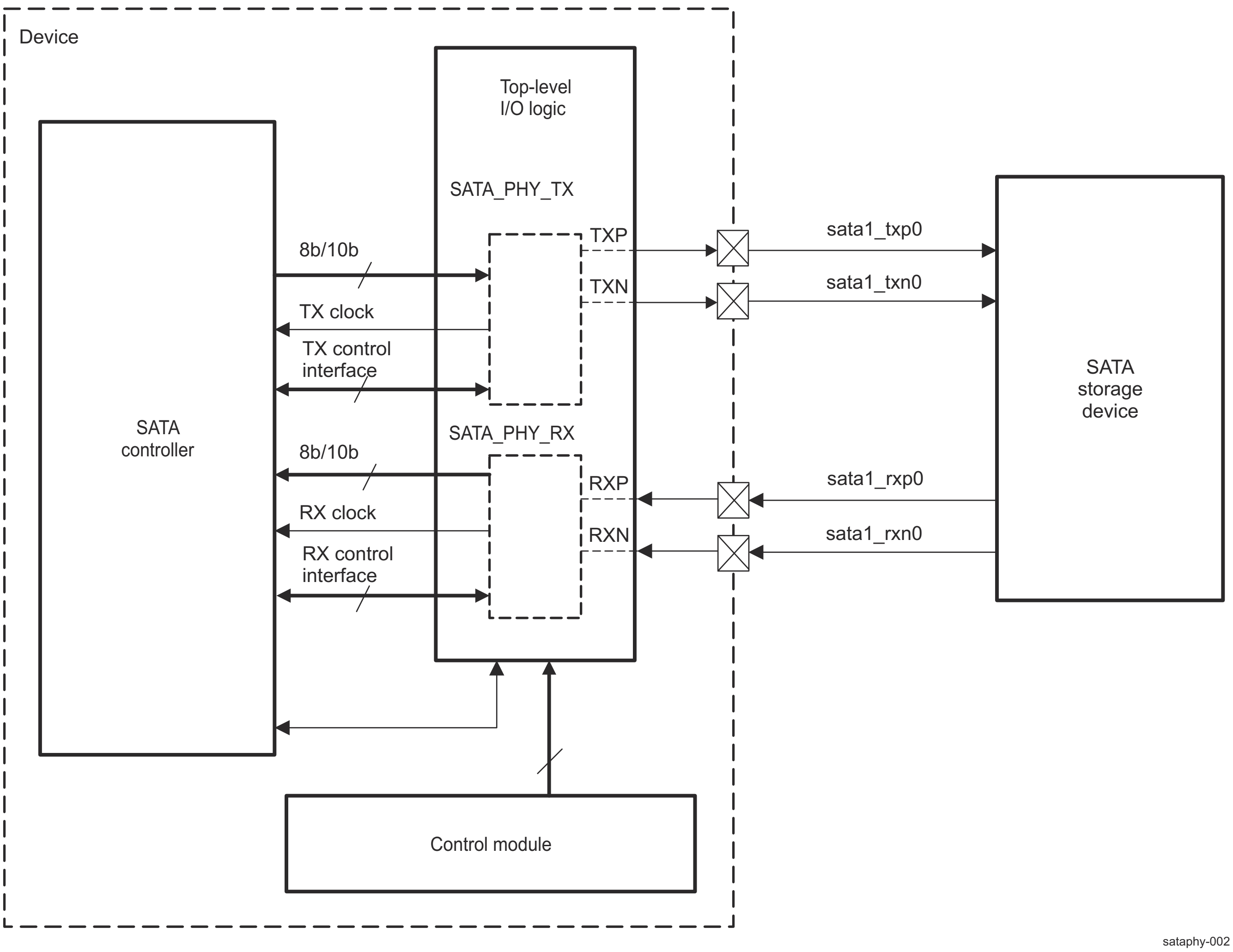


Figure 26-2. SATA PHY I/O Signals

Note

The path from module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the control module registers. See *Control Module*, for more information.

26.1.3 SATA PHY Subsystem Integration

This section describes the SATA PHY subsystem-related components (SATA_PHY, DPLL_SATA, DPLLCTRL_SATA, OCP2SCP3) integration in the device, including information about clocks, resets, and hardware requests.

Figure 26-3 shows the SATA_PHY integration.

The SATA_PHY module integration features are:

- A low-power nonretention reset, L3INIT_RST
- A DPLL reference input clock, PLL_CLK
- Parallel 10-bit TD[9:0] (SATA_PHY_TX) and 10-bit RD[9:0] (SATA_PHY_RX) data interfaces to SATA controller
- TX_CLK output clock for the parallel TX data interface between SATA_PHY.SATA_PHY_TX and SATA controller
- RX_CLK output clock for the parallel RX data interface between SATA_PHY.SATA_PHY_RX and SATA controller
- REF_CLK clock input (SATA_REF_GFCLK)
- A device core control module command port to the SATA_PHY integrated power sequencer

The DPLL_SATA integration features are:

- A low-power nonretention reset, L3INIT_RST
- A power, reset, and clock management (PRCM) gated version of SYS_CLK (SATA_REF_GFCLK) supplying the DPLL_SATA.CLKINP pin
- A single high-frequency clock output (DPLL_SATA.CLKDCOLDO to the SATA_PHY TX/RX components)
- No high-speed (HS) divider integration
- Idle signaling implementation
- LDO and DCO PWRDNZ monitoring signals
- A DPLL_LOCK locked status indication output to SATA_PHY and the SATA controller

The DPLLCTRL_SATA integration features are:

- A low-power nonretention reset, L3INIT_RST
- (OCP2SCP3) Interconnect adapter target interface
- Configuration/control programming interface to the DPLL_SATA
- No direct gate control for DPLL_SATA.CLKINP and CLKDCOLDO

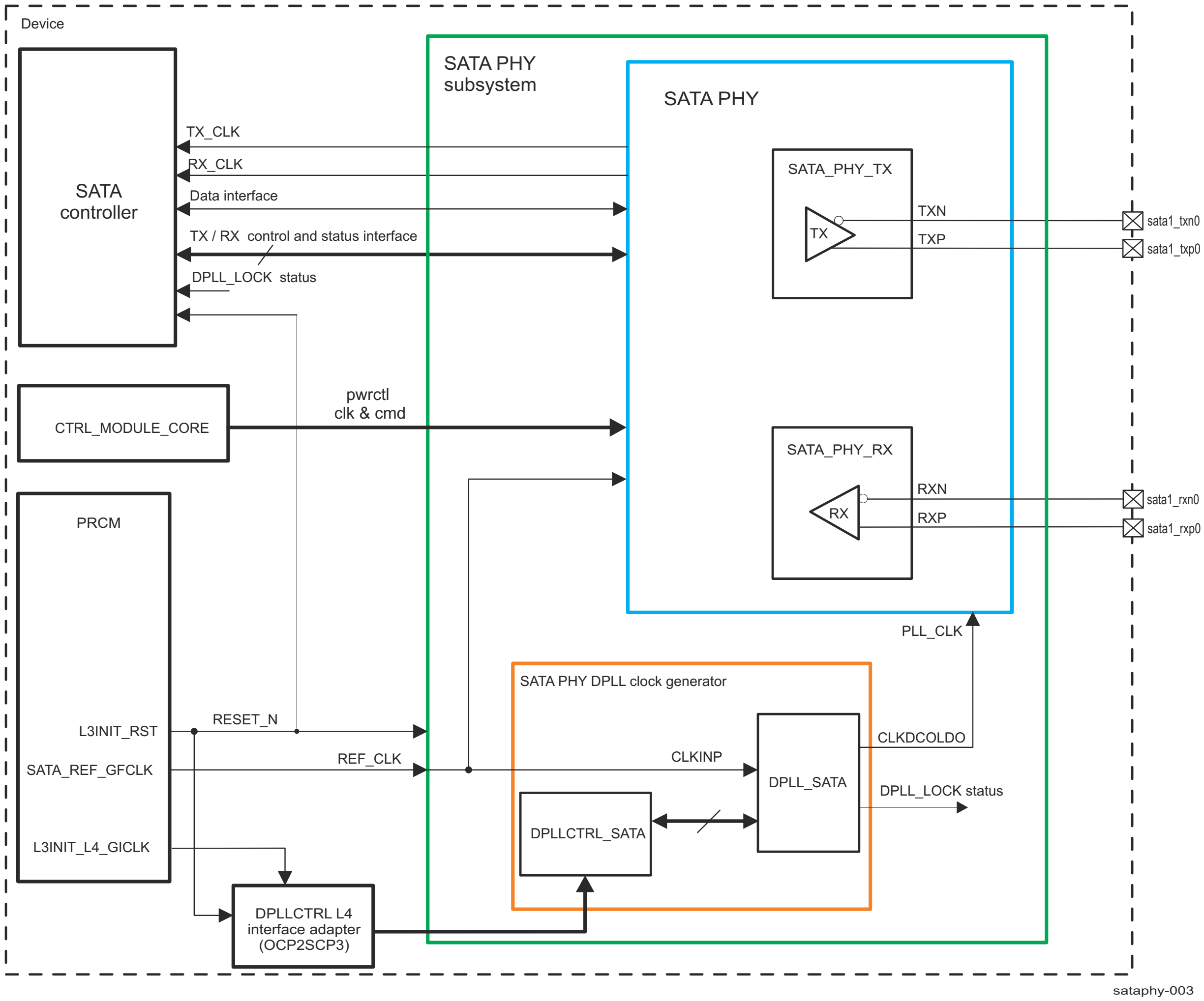


Figure 26-3. SATA PHY Subsystem Integration

Table 26-2 through Table 26-4 summarize the integration of the module in the device.

Table 26-2. SATA PHY Integration Attributes

Module Instance	Attributes	
	Power Domain	**Interconnect**
DPLLCTRL_SATA	PD_L3INIT	OCP2SCP3 adapter SCP interconnects
DPLL_SATA	PD_COREAON	
OCP2SCP3		L4_CFG

Table 26-3. SATA PHY Clocks

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DPLL_SATA/ SATA_PHY	REF_CLK	SATA_REF_GFCLK	PRCM	SATA DPLL and SATA PHY reference functional clock (SYS_CLK)
OCP2SCP3	L4CFG_ADAPTER_CLKIN	L3INIT_L4_GICLK	PRCM	PHY/DPLL L4_CFG adapter interface clock

Table 26-4. SATA PHY Resets

Resets

Table 26-4. SATA PHY Resets (continued)

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SATA_PHY subsystem	RESET_N	L3INIT_RST	PRCM	A nonretention reset to all SATA_PHY subsystem components

Note

The SATA_PHY, DPLL_SATA, and DPLLCTRL_SATA modules do not generate DMA, interrupt, or wakeup hardware requests to the surrounding modules integrated in the device.

26.1.4 SATA PHY Subsystem Functional Description

26.1.4.1 SATA PLL Controller L4 Interface Adapter Functional Description

The L4_CFG interconnect adapter (OCP2SCP3), allows user software to configure the DPLLCTRL_SATA registers over the L4_CFG port. Hence it is expected that this adapter is configured to operate before any programming of the DPLLCTRL_SATA.

Aa value of 0x6 or more should be written to the register bitfield OCP2SCP_TIMING[3:0] SYNC2.

L4_CFG interconnect adapter software reset is performed via writing OCP2SCP_SYSCONFIG[1] SOFTRESET to 0b1. The software reset completion is observed in bit OCP2SCP_SYSSTATUS[0] RESETDONE.

By default a smart-idle power mode is selected for OCP2SCP3. The smart-idle mode supported by OCP2SCP3 is not wake-up capable, which means software must explicitly take care to wake the OCP2SCP3 by setting the OCP2SCP_SYSCONFIG [4:3] IDLEMODE bit field to 0x1 (no idle), once it has previously gone to an idle mode.

By default OCP2SCP_SYSCONFIG[0] AUTOIDLE =0x1, which defines that the PLLCTRL L4 interface adapter automaticaly gates its L4 input clock based on L4_CFG interconnect activity. To enable a free-running clock, one should set bit AUTOIDLE to 0x0.

26.1.4.2 SATA PHY Serializer and Deserializer Functional Descriptions

26.1.4.2.1 SATA PHY Reset

No software reset is applicable to SATA_PHY. L3INIT reset assertion is automatically managed by hardware.

26.1.4.2.2 SATA_PHY Clocking

26.1.4.2.2.1 SATA_PHY Input Clocks

A standard high-speed 1.5 GHz input clock of SATA_PHY is provided by the DPLL_SATA.CLKDCOLDO output. PRCM. SATA_REF_GFCLK (derived from the SYS_CLK) should be enabled, to provide the necessary DPLL_SATA and SATA_PHY clocking within the SATA subsystem.

Depending on the SATA_REF_GFCLK (SYS_CLK derived) frequency, following settings should be made in the CTRL_MODULE_CORE.CONTROL_PHY_POWER_SATA[31:22] SATA_PWRCTL_CLK_FREQ bitfield for proper operation:

- 0x26: If SYSCLK = 38.4 MHz
- 0x1A: If SYSCLK = 26.0 MHz
- 0x13: If SYSCLK = 19.2 MHz
- 0x10: If SYSCLK = 16.8 MHz
- 0x0C: If SYSCLK = 12.0 MHz

26.1.4.2.2.2 SATA_PHY Output Clocks

The 10-bit data parallelly transmitted data between SATA_PHY_TX and SATA controller link logic is sampled with respect to the TX_CLK. Frequency of this clock is obtained as the SATA_PHY.PLL_CLK clock frequency:

- Divided by 10 if a 1.5-Gbps limit is negotiated by software setting DWC_ahsata.SATA_PxSCTL[7:4] SPD = 0x1
- Divided by 5 if a 3-Gbps limit is negotiated by software setting DWC_ahsata.SATA_PxSCTL[7:4] SPD = 0x0 or 0x2

The actually negotiated between host and the attached SATA device speed is indicated in the read-only SATA controller.SATA_PxSSTS[7:4] SPD bit field. For more information, see Section 24.8.6, *SATA Controller Register Manual*, in Section 24.8, *SATA Controller*.

RX_CLK: The SATA_PHY_RX output RX_CLK is a clock that is recovered from the serial data received over the RXP/RXN lanes serial data. The RX_CLK clock supplies the SATA controller link parallel 10-bit data reception logic.

26.1.4.2.3 SATA_PHY Power Management

The SATA_PHY power sequencer receives a power-up/power-down command input from the device general core control module. The power sequencer takes care to execute different stages of the SATA_PHY_TX and SATA_PHY_RX power-up/-down processes. Besides a power-up/-down functionality, the SATA PHY accepts power transition commands/states (Ready state, Partial or Slumber states) from the SATA controller link power management port .

26.1.4.2.3.1 SATA_PHY Power-Up/-Down Sequences

Powering up/down the SATA_PHY_TX and SATA_PHY_RX modules is triggered by software writing corresponding power-up/-down commands in the CTRL_MODULE_CORE. CONTROL_PHY_POWER_SATA register of the device core control module, as follows:

- CONTROL_PHY_POWER_SATA[21:14] SATA_PWRCTL_CLK_CMD = 0x0 (default value), commands both SATA_PHY_TX and SATA_PHY_RX to power-down (OFF) state.
- CONTROL_PHY_POWER_SATA[21:14] SATA_PWRCTL_CLK_CMD = 0x1, powers up the SATA_PHY_RX only.
- CONTROL_PHY_POWER_SATA[21:14] SATA_PWRCTL_CLK_CMD = 0x2, powers up the SATA_PHY_TX only.
- CONTROL_PHY_POWER_SATA[21:14] SATA_PWRCTL_CLK_CMD = 0x3, simultaneously powers up the SATA_PHY_RX and the SATA_PHY_TX.

For more information, see *Control Module*.

26.1.4.2.3.2 SATA_PHY Low-Power Modes

Setting the DWC_ahsata.SATA_PxSCTL[3:0] DET bit field to 0x4 automatically puts the SATA_PHY_RX deserializer in offline mode. This software operation has no impact over the SATA_PHY_TX serializer.

Note

The SATA controller transition to partial or slumber mode has no impact over the SATA_PHY_RX deserializer.

Note

The SATA_PHY_TX.TX_CLK output clock is a free-running clock and cannot be gated, even when the transmitter is disabled by a SATA controller.

26.1.4.2.4 SATA_PHY Hardware Requests

No DMA, interrupt or wake-up requests are generated by the SATA_PHY to the surrounding SATA controller subsystem components.

26.1.4.3 SATA Clock Generator Subsystem Functional Description

The DPLL_SATA, which is located outside the PRCM boundaries and is part of the SATA host controller subsystem, directly injects a high-speed clock into the SATA_PHY serializer/deserializer input, PLL_CLK. The DPLL generator is controlled through a programmable interface from a dedicated PLL controller, DPLLCTRL_SATA.

26.1.4.3.1 SATA DPLL Clock Generator Overview

The SCP interface of the SATA PLL controller (the DPLLCTRL_SATA instance) is used to set the configuration of the digital phase-locked loop (DPLL) modules, primarily the various counter values. Figure 26-4 is an overview of the DPLL clock generator embedded into the SATA host controller subsystem.

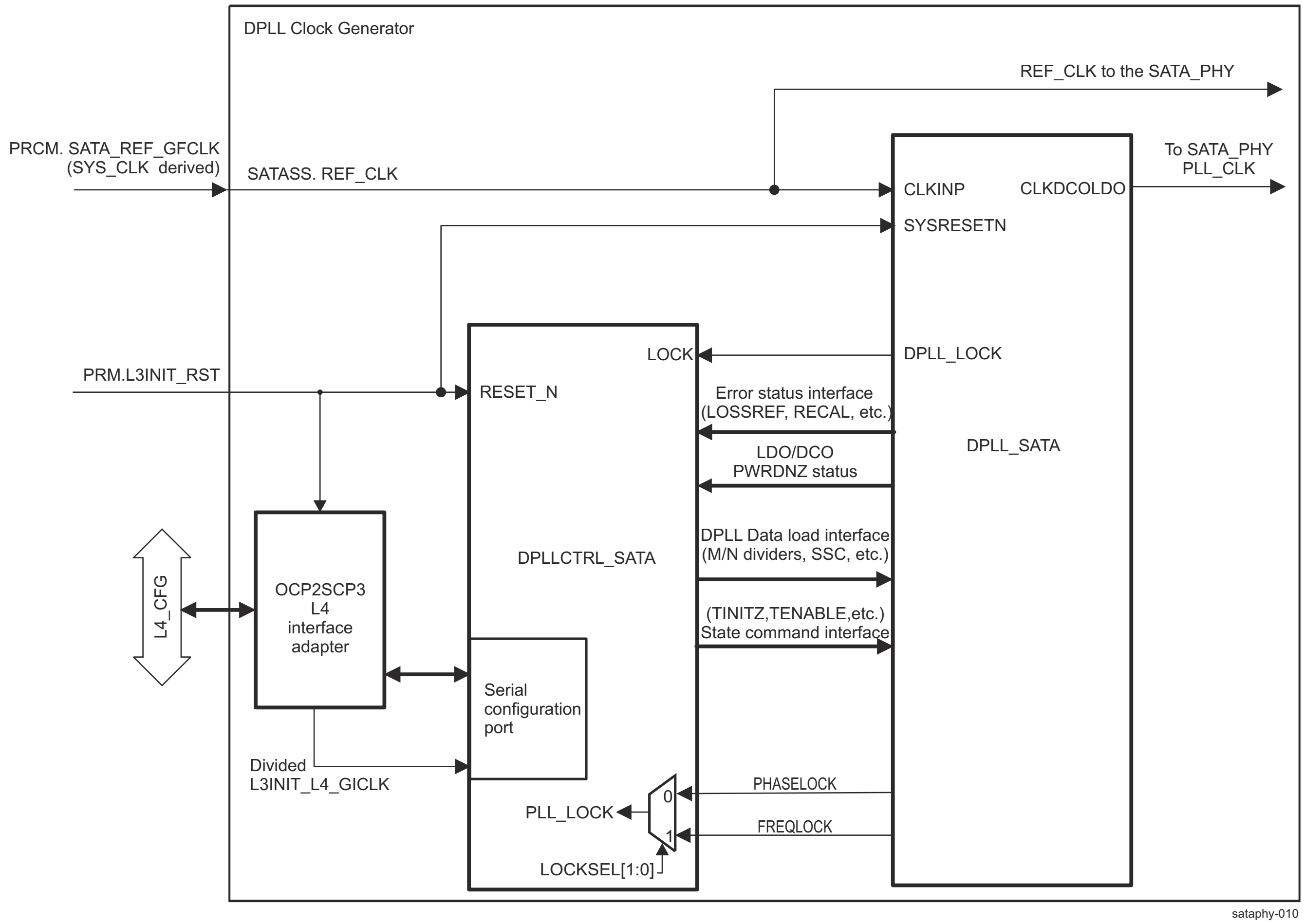


Figure 26-4. SATA DPLL Clock Generator Overview

Note

The DPLLCTRL_SATA module is user accessible on the L4_CFG interconnect. However, to make access possible, the user must configure the OCP2SCP3 instance prior to accessing the DPLLCTRL_SATA registers.

The DPLL_SATA features are:

- A programmable 8-bit input divider: N
- A programmable 12-bit integer, 18-bit fractional loop multiplier: M
- Digital control and loop filter
- Internal oscillator output clock on internal LDO domain (CLKDCOLDO output)
- DPLL output clock spread spectrum clocking (SSC) support
- No retention capabilities
- No idle-bypass fast relock capabilities
- Idle-bypass low-power mode
- M/N bypass mode
- Relock from standby

The DPLLCTRL_SATA components features are:

- DPLL error and status notification
- DPLL initialization and configuration
- DPLL lock criteria selectable between frequency and phase lock

- Idle command implementation
- No software reset implementation

26.1.4.3.2 SATA DPLL Clock Generator Reset

The SATA PLL controller and SATA DPLL clock generator share a common hardware nonretention reset (L3INIT_RST) which comes from the device power and reset manager. When the DPLL_SATA hardware reset completes, the DPLLCTRL_SATA.PLL_STATUS[0] PLLCTRL_RESET_DONE bit is automatically updated to 1. For more information on the hardware reset source, see *Reset Domains*, in *Power, Reset, and Clock Management*.

The DPLLCTRL_SATA itself has no software reset capabilities.

The DPLLCTRL_SATA performs a software reset sequence on the DPLL_SATA in hardware (through the TINITZ signal activation).

26.1.4.3.3 SATA DPLL Low-Power Modes

The power-management port (PMP) is not integrated for DPLLCTRL_SATA. The DPLL_SATA has no retention capabilities. This means the DPLL digital power supply remains switched on during all modes of operation.

The low-power (LP) modes supported by DPLL_SATA are idle-bypass low power and MN-bypass modes, which are both characterized by:

- Internal LDO switched off
- DCO oscillator switched off
- CLKDCOLDO output pulled low

For more details on the PLL settings and conditions necessary to enter idle-bypass and MN-bypass low-power modes, see Section 26.1.4.3.6.4, *SATA DPLL Idle-Bypass Low-Power Mode*, and Section 26.1.4.3.6.5, *SATA DPLL MN-Bypass Mode*.

DPLL_SATA is held in a similar low-power state (DCO and LDO switched off, with CLKDCOLDO = 0) after POR, before the first PLL_GO command has been software-triggered on the PLL controller. See Section 26.1.4.3.6.1, *SATA Clock Generator Power Up*.

26.1.4.3.4 SATA DPLL Clocks Configuration

26.1.4.3.4.1 SATA DPLL Input Clock Control

The DPLL_SATA accepts the functional clock (SATA_REF_GFCLK) on its CLKINP pin (REF_CLK input at SATA SS level) directly from the device PRCM, without involving any DPLLCTRL_SATA interactions. The SATA_REF_GFCLK is derived from SYS_CLK1, and can be optionally software-gated by setting the PRCM.CM_L3INIT_SATA_CLKCTRL[8] OPTFCLKEN_REF_CLK bit. The status of the SATA_REF_GFCLK clock can be monitored in the PRCM.CM_L3INIT_CLKSTCTRL[19] CLKACTIVITY_SATA_REF_GFCLK bit. See *Clock Domain Module Attributes* in *Power, Reset, and Clock Management*.

If CLKINP signal is lost for some time, the LOSSREF output signal, which serves as a feedback to DPLLCTRL_SATA, is asserted high. When CLKINP resumes, the LOSSREF signal goes low (LOSSREF inactive state). The LOSSREF status signal can be monitored by software in the DPLLCTRL_SATA.PLL_STATUS[3] PLL_LOSSREF bit.

26.1.4.3.4.2 SATA DPLL Output Clock Configuration

Only the DPLL_SATA output, CLKDCOLDO, is used to provide the high-speed clock at the PLL_CLK pin of the SATA_PHY. Only the REGM, REGN, and SD divider values are used within the DPLL clock generator subsystem to adjust the CLKDCOLDO output clock frequency. This is done by programming the DPLLCTRL_SATA.PLL_CONFIGURATION1[20:9] PLL_REGM, DPLLCTRL_SATA.PLL_CONFIGURATION1[8:1] PLL_REGN, and DPLLCTRL_SATA.PLL_CONFIGURATION3[17:10] PLL_SD bit fields, respectively. The SATA DPLL CLKOUT and CLKOUTLDO outputs are not used, and internal REGM2 and REGM1 dividers are not software controllable.

Note

At the DPLL/PLLCTRL integration level, the PLL_REGM1[3:0] and PLL_REGM2[6:0] divider control signals are tied-off by hardware to 0x0 and 0x1, respectively.

For more details on output clock settings sequence, see Section 26.1.4.3.7.2, *SATA DPLL Clock Programming Sequence*.

26.1.4.3.4.2.1 SATA DPLL Output Clock Gating

There is no direct software gate control for the DPLL_SATA.CLKDCOLDO output.

The DPLL_SATA.CLKDCOLDO clock output is automatically gated (CLKDCOLDO pulled low) in the following scenarios:

- DPLL power-up sequence. For more information on the power-up sequence, see Section 26.1.4.3.6.1, *SATA Clock Generator Power Up*.
- DPLL entering a relock sequence. For more information on the relocking sequence, see Section 26.1.4.3.6.2, *SATA DPLL Sequences*.
- DPLL entering idle-bypass low-power mode. For more information on idle-bypass mode, see Section 26.1.4.3.6.4, *SATA DPLL Idle-Bypass Mode*.
- DPLL entering MN-bypass mode. For more information on MN-bypass mode, see Section 26.1.4.3.6.5, *SATA DPLL MN Bypass Mode*.

26.1.4.3.5 SATA DPLL Subsystem Architecture

Figure 26-5 is a simplified block diagram of the DPLL_SATA instance integration in the SATA clock generator subsystem.

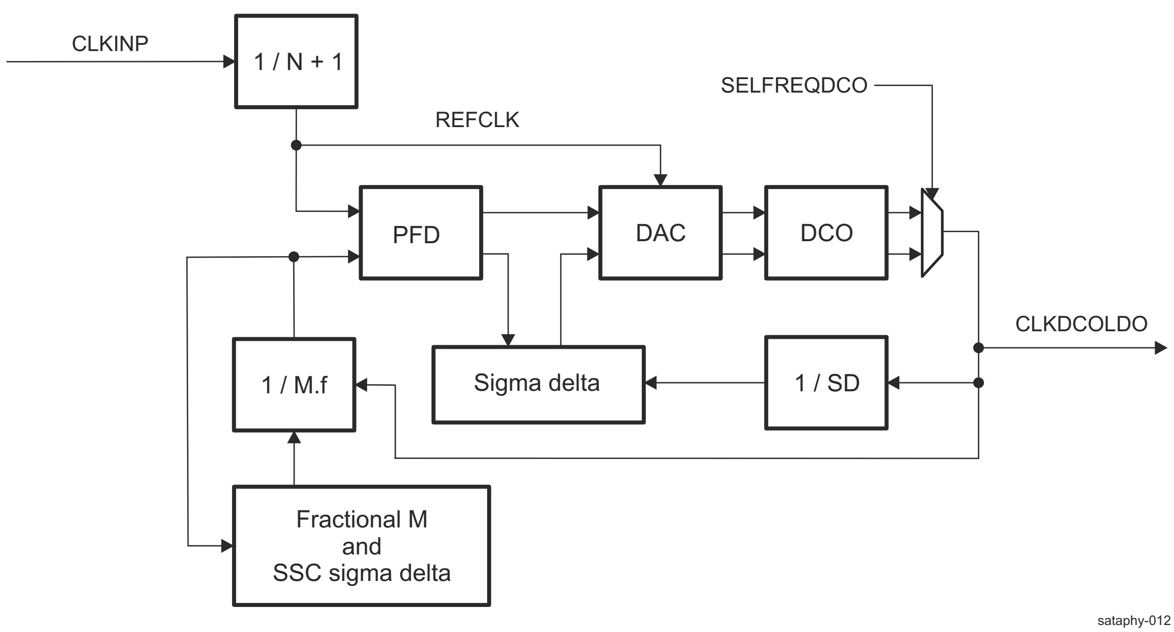


Figure 26-5. DPLL_SATA Functional Block Diagram

The input clock CLKINP goes to a predivider N + 1. The entire loop runs on the REFCLK clock after this predivider. The value of N + 1 is controlled through the DPLLCTRL_SATA.PLL_CONFIGURATION1[8:1] PLL_REGN bit field.

The frequency ranges for the DPLL_SATA input clock (CLKINP) and the DPLL internal reference clock (REFCLK = CLKINP/N + 1) are:

- 0.62 to 60 MHz for CLKINP

- 0.62 to 2.5 MHz for the REFCLK

The output clock CLKDCOLDO is synthesized by digitally controlled oscillator (the DCO block), that automatically detects the frequency range. The CLKDCOLDO frequency can be given with CLKDCOLDO = CLKINP × M/(N + 1). For that purpose the feedback multiplier M must be configured through the DPLLCTRL_SATA.PLL_CONFIGURATION1[20:9] PLL_REGM bit field.

The DPLL_SATA module supports fractional synthesis (that is, the frequency multiplication factor M can be programmed as fractional). This is achieved by having a sigma delta feedback divider (M). A fractional value (Fractional M) of 18 bits is supported enabling control for a better accuracy. Programming the 18-bit Fractional M value is done by setting the DPLLCTRL_SATA.PLL_CONFIGURATION4[17:0] PLL_REGM_F bit field (similar to REGM). To enable integer only division Fractional M should be set to 000…0.

Note

Fractional synthesis is not supported for M > 4093.

26.1.4.3.6 SATA DPLL Clock Generator Modes and State Transitions

The DPLL_SATA can be set in different modes during operation. PLLCTRL triggers DPLL_SATA state transitions to different static modes by means of TINITZ and TENABLE hardware control signals.

26.1.4.3.6.1 SATA Clock Generator Power Up

After power up, the DPLL_SATA.SYSRESETN input is automatically pulled low by PRM, together with DPLLCTRL_SATA.RESET_N input. Because PRM.L3INIT_RST is an asynchronous reset, the DPLL_SATA input clock (DPLL_SATA.CLKINP) is not demanded upon reset. The LOSSREF signal, which monitors the presence of CLKINP clock, remains 1 during SYSRESETN = 0 irrespective of presence/absence on the CLKINP clock. If CLKINP is present upon reset assertion, the LOSSREF signal is deasserted to 0, a certain time after the hardware reset completion. During DPLL power-up mode, CLKDCOLDO clock is maintained inactive (pulled low). After POR, the DPLL_LOCK (internal lock loop) signal is maintained deasserted, too.

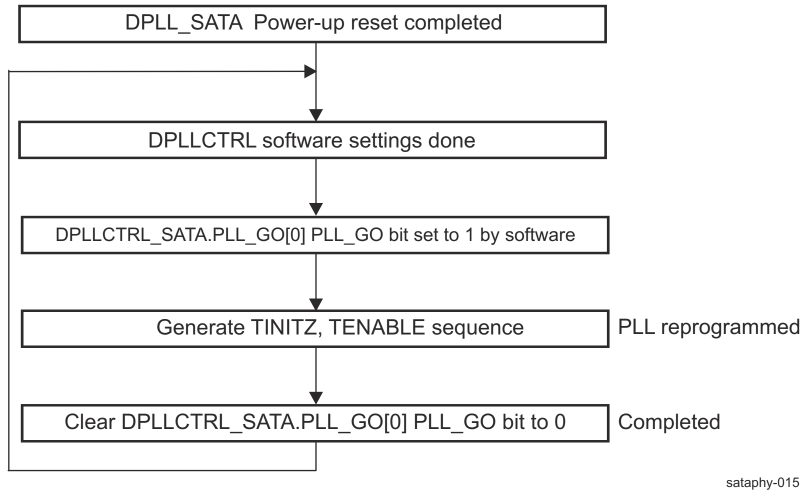
26.1.4.3.6.2 SATA DPLL Sequences

Once all the configuration values have been initially programmed into the DPLLCTRL_SATA registers (see Section 26.1.4.3.7.2, *SATA DPLL Clock Programming Sequence*), the DPLLCTRL_SATA.PLL_GO[0] PLL_GO bit should be set to update the configuration values and start the DPLL calibration and locking sequence.

After DPLLCTRL_SATA.PLL_GO[0] PLL_GO bit is set high in software, the DPLLCTRL_SATA state-machine takes the following action:

1. Sets TINITZ signal to 0. TINITZ acts as a soft reset to DPLL_SATA. This starts the DPLL initialization procedure. During initialization mode, the CLKDCOLDO, FREQLOCK, and PHASELOCK signals are kept at 0 and the BYPASSACK signal is kept at 1.
   VDDA power supply of DPLL should be active before DPLL_SATA initialization is performed, but it is not required to be switched on immediately after device power up.
2. The TENABLE signal is asserted high by the PLLCTRL hardware to load the user-programmed values of REGM, REGN, REGSD, and SELFREQDCO into DPLL registers.
3. After TENABLE is asserted, TINITZ is driven high (disabled) by the PLLCTRL to trigger a DPLL calibration and lock sequence after the loop control values are loaded. The module calibration-lock sequence will begin from the first CLKINP edge after TINITZ is disabled.

Figure 26-6 summarizes the software and hardware sequences flow of DPLL_SATA.



All thick-outlined blocks in Figure 26-6 show operations performed by software. Other blocks show operations performed by hardware.

Figure 26-6. SATA PLL GO Sequence

DPLL_SATA Relock Sequence: When DPLL leaves a lost clock condition (LOSSREF = 1 → 0) or idle-bypass mode it enters relock sequence from the first CLKINP edge (after bypass mode leaving). Relock sequence is the same as calibration-lock sequence already described.

A DPLL relock sequence is also software triggered by setting DPLLCTRL_SATA.PLL_GO[0] PLL_GO bit to 0b1 for DPLL parameters update.

When DPLL_SATA enters a relock sequence, CLKDCOLDO is pulled low. FREQLOCK and PHASELOCK status signals are also low. CLKDCOLDO output clock is activated after FREQLOCK or PHASELOCK signal goes high, depending on the selected locking criteria.

The DPLLCTRL_SATA.PLL_GO[0] PLL_GO bit can be used by software to monitor if PLLCTRL locking process is still pending (PLL_GO = 0b1).

26.1.4.3.6.3 SATA DPLL Locked Mode

When DPLL_SATA finishes calibration and lock sequences it enters the LOCKED state. During the LOCKED state, LOSSREF is deasserted, BYPASSACK is deasserted, and the FREQLOCK or PHASELOCK signals are asserted.

DPLL lock event criteria (FREQLOCK or PHASELOCK) is software selectable through the DPLLCTRL_SATA.PLL_CONFIGURATION2[10:9] PLL_LOCKSEL bit field.

The DPLL indicates it is in the LOCKED state to the DPLLCTRL_SATA, SATA controller core controller, and SATA_PHY through assertion of the DPLL_LOCK signal, which reflects the internal lock loop status. The user software can monitor the DPLL locked event in the DPLLCTRL_SATA.PLL_STATUS[1] PLL_LOCK bit, which is active high.

26.1.4.3.6.4 SATA DPLL Idle-Bypass Mode

Idle-bypass fast relock mode is not supported for DPLL_SATA.

DPLL_SATA supports idle-bypass low-power mode. A transition from a normal operation to idle-bypass mode is performed when software sets the DPLLCTRL_SATA.PLL_CONFIGURATION2[0] PLL_IDLE bit to 0x1. IDLE signal assertion triggers a power-down sequence on DPLL internal LDO analog blocks and DCO oscillator.

In idle-bypass low-power mode, the PHASELOCK and FREQLOCK output signals are asserted low and CLKDCOLDO goes low, respectively. Also, the internal reference clock REFCLK = CLKINP/N + 1 is gated inside the DPLL digital control logic to save power.

In the functional mode, DPLL_SATA.TICOPWDN output indicates to the PLL controller the status of the power-down signal (active high) of the internal DCO oscillator. The internal DCO oscillator is powered down in idle-bypass mode or during period from SYSRESETN 0 → 1 to module initialization. The DCO oscillator exits power down (TICOPWDN goes low) whenever the module internally tries to lock or relock after initialization or exiting idle-bypass mode.

In the functional mode, DPLL_SATA.LDOPWDN output indicates to the PLL controller the status of the power-down signal (active high) of the internal LDO. LDOPWDN goes high as soon as the internal LDO is powered down. LDOPWDN goes low after the LDO output voltage is stable. The internal LDO is powered down in period from SYSRESETN 0 → 1 to module initialization or when entering into idle-bypass mode. LDOPWDN is cleared whenever the module internally tries to lock or relock after initialization or exiting idle-bypass mode after the internal LDO output voltage has stabilized.

The DCO and LDO power ON/OFF states are reflected within the read-only DPLLCTRL_SATA.PLL_STATUS[16] PLL_TICOPWDN and DPLLCTRL_SATA.PLL_STATUS[15] LDOPWDN monitor bits.

To exit idle-bypass mode and restore clock generation, the user should set DPLLCTRL_SATA.PLL_CONFIGURATION2[0] PLL_IDLE to 0x0, which deasserts the IDLE signal and DPLL_SATA automatically enters a relock sequence. The CLKDCOLDO output clock is activated after the FREQLOCK or PHASELOCK signal goes high, depending on selected locking criteria.

26.1.4.3.6.5 SATA DPLL MN-Bypass Mode

The MN-Bypass mode will be activated if REGM = 0 or 1 is loaded into the module on the rising edge of TENABLE. TINITZ also should be pulsed to enter MN-Bypass mode. The module enters a low-power mode by gating all its internal clocks (REFCLK) and powering down the internal LDO (LDOPWDN = 1) and DCO (DCOPWDN = 1). CLKDCOLDO remains gated (low) during this mode.

Note

When the DPLLCTRL_SATA.PLL_CONFIGURATION1[20:9] PLL_REGM bit field is updated to 0x0 or 0x1, the CLKDCOLDO output clock is gated.

26.1.4.3.6.6 SATA DPLL Error Conditions

The PLL lock and recalibration signals can be monitored to detect the loss of lock condition and the DPLL requirement to recalibrate (caused by a large temperature change since the last lock request):

- The DPLLCTRL_SATA.PLL_STATUS[2] PLL_RECAL bit informs whether the DPLL_SATA must be recalibrated.

The PLL reference clock (CLKINP) loss status and PLL-in-high-jitter condition can also be monitored:

- The DPLLCTRL_SATA.PLL_STATUS[1] PLL_LOCK bit gives the SATA PLL LOCKED state.
- The DPLLCTRL_SATA.PLL_STATUS[3] PLL_LOSSREF bit informs whether the DPLLCTRL_SATA has lost the reference clock.
- The DPLLCTRL_SATA.PLL_STATUS[5] PLL_HIGHJITTER bit informs whether the PLL has entered a high-jitter condition.

26.1.4.3.7 SATA PLL Controller Functions

26.1.4.3.7.1 SATA PLL Controller Register Access

The configuration registers are accessed through the OCP2SCP3 L4 adapter register space using the SCP interface of the DPLLCTRL_SATA. This includes all the configuration signals and returning status signals.

CAUTION

All writes must be 32-bit operations, because the SCP interface always transfers 32 bits; 16- or 8-bit operations may lead to unpredictable errors.

Note

Because the SATA_PHY directly provides parallel data interface clocks RX_CLK and TX_CLK to the SATA controller, the DPLLCTRL_SATA and DPLL_SATA must be configured before any data transfer between the SATA controller and an external SATA storage device.

26.1.4.3.7.2 SATA DPLL Clock Programming Sequence

The DPLL_SATA factors must be calculated based on the required input and output frequencies, keeping the PLL internal reference frequency (REFCLK) in the appropriate range (0.62 to 2.5 MHz).

- The REGM factor is programmed in the DPLLCTRL_SATA.PLL_CONFIGURATION1[20:9] PLL_REGM bit field.
- The fractional part of REGM factor is programmed in the DPLLCTRL_SATA.PLL_CONFIGURATION4[17:0] PLL_REGM_F bit field.
- The REGN factor is programmed in the DPLLCTRL_SATA.PLL_CONFIGURATION1[8:1] PLL_REGN bit field.
- The DCO frequency range is set in the DPLLCTRL_SATA.PLL_CONFIGURATION2[3:1] PLL_SELFREQDCO bit field.
  - PLL_SELFREQDCO should be set to 0x2, if 750 MHz < CLKDCOLDO [MHz] < 1500 MHz
  - PLL_SELFREQDCO should be set to 0x4, if 1250 MHz < CLKDCOLDO [MHz] < 2500 MHz
- The SD divider is programmed in the DPLLCTRL_SATA.PLL_CONFIGURATION3[17:10] PLL_SD bit field. The DPLLCTRL_SATA.PLL_CONFIGURATION2[3:1] PLL_SELFREQDCO bit field should be programmed depending on the value of CLKDCOLDO = CLKINP × M/(N + 1). The formulas are shown in Figure 26-7.

Figure 26-7 shows the programming sequence.

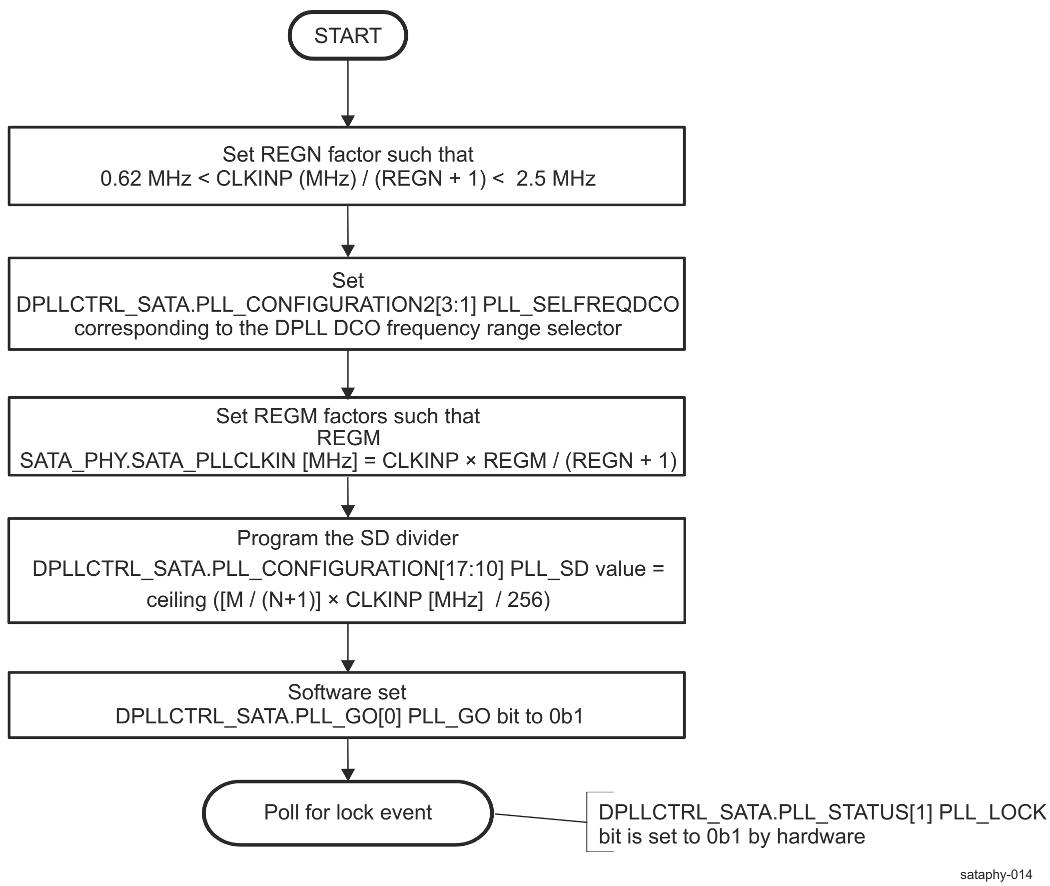


Figure 26-7. SATA PLL Programming Sequence

Note

- The equation for SATA_PHY_TX/SATA_PHY_RX (MHz) applies to the CLKDCOLDO of the DPLL_SATA.
- For normal operation CLKDCOLDO output frequency of the DPLL_SATA should be either 750 MHz (SATA-1 mode) or 1500 MHz (SATA-2 mode).

Table 26-5 summarizes the registers for the DPLLCTRL_SATA programming sequence.

Table 26-5. Register Call Summary for SATA PLL Programming Sequence

Register Name	Register Name	Register Name	Register Name
DPLLCTRL_SATA.PLL_CONFIGURATION2	DPLLCTRL_SATA.PLL_GO	DPLLCTRL_SATA.PLL_STATUS	DPLLCTRL_SATA.PLL_CONFIGURATION3

26.1.4.3.7.3 SATA DPLL Recommended Values

Table 26-6 lists the DPLL_SATA recommended values.

Table 26-6. Recommended Programming Values

Field Name	Value	Description
DPLLCTRL_SATA.PLL_CONFIGURATION1[20:9] PLL_REGM	See (1).	Feedback clock divider
DPLLCTRL_SATA.PLL_CONFIGURATION1[8:1] PLL_REGN	See (1).	Reference clock divider
DPLLCTRL_SATA.PLL_CONFIGURATION2[10:9] PLL_LOCKSEL	0x-	Criteria to lock the PLL
DPLLCTRL_SATA.PLL_CONFIGURATION2[3:1] PLL_SELFREQDCO	See (1).	Program based on the PLL choice and lock frequency.
DPLLCTRL_SATA.PLL_CONFIGURATION2[0] PLL_IDLE	0	PLL active
DPLLCTRL_SATA.PLL_CONFIGURATION3[17:10] PLL_SD	See (1).	Ceiling { [PLL_REGM/(PLL_REGN + 1)] × CLKINP(MHz)/256 }
DPLLCTRL_SATA.PLL_GO[0] PLL_GO	0x1	Write 1 when PLL is to be (re)locked with new parameters. This bit is cleared by hardware when the PLL request completes.

(1) The value of the bit field must be set according to the desired clock frequencies – 750 MHz for SATA-1 and 1.5 GHz for SATA-2 speeds.

26.1.5 SATA PHY Subsystem Low-Level Programming Model

Table 26-7 summarizes the low-level programming sequence to set up the SATA PHY subsystem for SATA I/O operations.

Table 26-7. SATA PHY Subsystem Low-Level Programming Sequence

Step	Description	Comment
1.	Set the startup low-performance OPP in the appropriate PRCM registers.	For more information regarding demanded OPP, see the device *Data Manual*.
2.	Enable the PRCM.SATA_REF_GFCLK.	See *Clock Domain Module Attributes*, in *Power, Reset, and Clock Management*.
3.	Enable the PRCM.L3INIT_L4_GICLK clock to enable the OCP2SCP3 interface adapter operation.	See *Clock Domain Module Attributes*, in *Power, Reset, and Clock Management*.
4.	Software reset the OCP2SCP3 and poll until soft reset completion is indicated in status.	See Section 26.1.4.1.
5.	Set up division ratio between the OCP clock (PRCM.L3INIT_L4_GICLK) and SCP clock to supply the serial configuration register domains of the DPLLCTRL_SATA.	See Section 26.1.4.1.
6.	Set up necessary SYNC1 and SYNC2 timings to ensure no blocking of transactions over the SCP bus.	. See Section 26.1.4.1.After this step user is ready to access DPLLCTRL_SATA .
7.	Configure DPLL_SATA to generate frequency (CLKDCOLDO) = 1.5 GHz.	See Section 26.1.4.3.7.2 and Table 26-8.
8.	Soft assert bit DPLLCTRL_SATA.PLL_GO[0] PLL_GO to 0x1.	Start the DPLL lock with desired parameters.
9.	Poll DPLLCTRL_SATA.PLL_STATUS[1] PLL_LOCK bit until it is seen 1.	DPLL locked event
10.	Perform a SATA_PHY tuning required for SATA i/f operation	Follow steps described in Table 26-9, *SATA PHY Tuning Table*.
11.	Software trigger the SATA_PHY_TX power-up sequence.	For more details, see Section 26.1.4.2.3.1, *SATA_PHY Power-Up/-Down Sequences*.
12.	Software trigger the SATA_PHY_RX power-up sequence.	For more details, see Section 26.1.4.2.3.1, *SATA_PHY Power-Up/-Down Sequences*.

Table 26-8. DPLL CLKDCOLDO Recommended Settings

Parameter	Setting				
F(CLKDCOLDO) MHz	1500				
SYS_CLK (MHz)	12	16.8	19.2	26	38.4
N	4	6	7	12	15
SELFREQDCO[2:0]	100	100	100	100	100
M	625	625	625	750	625
Frac	0	0	0	0	0
SD	6	7	6	6	6

Table 26-9. SATA PHY Tuning Table

Physical address[(2)] [bits to modify]	Preferred value setting[(1)]
0x4A09 600C [31:27]	0b01000 for SATA 1.5 Gbps mode 0b00100 for SATA 3 Gbps mode
0x4A09 600C [17:14]	0b1010 for SATA-Gen1x and SATA-Gen2x[(3)] 0b0101 for SATA-Gen1m and SATA-Gen2m
0x4A09 6028 [23:19]	0b11100 if spread-spectrum is ON 0b00001 if spread-spectrum is OFF
0x4A09 6028 [18:11]	0b01100110 (regardless of spread-spectrum being ON or OFF)
0x4A09 600C [6:5]	0b00
0x4A09 601C [31:30]	0b01

Table 26-9. SATA PHY Tuning Table (continued)

Physical address[2] [bits to modify]	Preferred value setting[1]
0x4A09 6024 [31:30]	0b10
0x4A09 6038 [31:16]	0x0000
0x4A09 6038 [15:7]	0b111110000
0x4A09 6038 [2:1]	0b11
0x4A09 6044 [10:9]	0b00

(1) These are the preferred settings of SATA_PHY, in case that SATA PHY PLL_CLK=1.5 GHz.
(2) Accesses to these locations have to be done in 32-bits only. User must NOT modify SATA PHY bits different than those described in Table 26-9.
(3) For SATA Genx and Genm description, please refer to *Serial ATA II Electrical Specification 1.0.*

26.2 USB3_PHY Subsystem

This chapter describes the features and functions of the USB3_PHY subsystem of the device.

26.2.1 USB3_PHY Subsystem Overview

The USB3.0 physical layer (PHY) is responsible for transmitting the USB1 controller media access layer (MAC) 32-bit parallel data output as a serial data stream over a differential pair (TXP/TXN) and converting the differentially received serial data (RXP/RXN) to 32-bit parallel input data demanded by the MAC receiver logic.

The USB3_PHY component operates with a 2.5 GHz-source clock, to achieve the USB 3.0 super-speed throughput of 5 Gbps. The USB3_PHY serializer/deserializer source clock is generated by a DPLL clock generator (DPLL_USB_OTG_SS) which isintegrated into the USB1 host subsystem.

The DPLL_USB_OTG_SS is configured and controlled through the USB3_PHY dedicated PLL controller (DPLLCTRL_USB_OTG_SS) with associated serial configuration port (SCP) accessible registers.

A common interconnect adapter component, OCP2SCP1, lets the user program the USB3_PHY serializer/deserializer and DPLLCTRL_USB_OTG_SS through the L4_CFG interconnect.

Figure 26-8 gives an overview of the USB3_PHY subsystem. Figure 26-8 shows that the USB3_PHY serializer and deserializer directly interract with the attached to USB OTG SS controller USB3.0 compatible device (over TXP/TXN transmission and RXP/RXN reception interface I/Os). The PIPE port interracts with the USB1 MAC port, described in details in Section 24.7, *SuperSpeed USB DRD*.

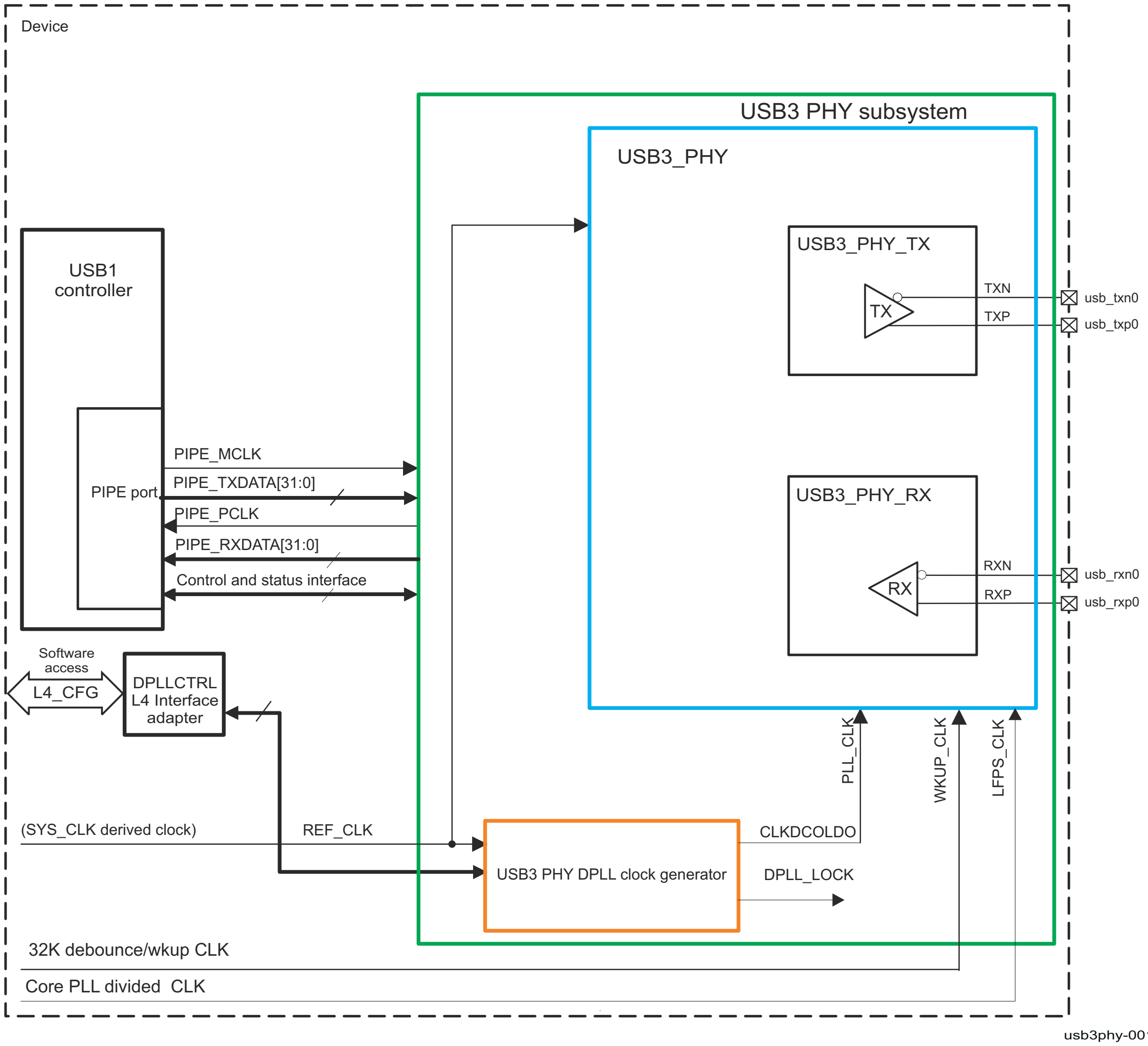


Figure 26-8. USB3_PHY Subsystem Overview

26.2.2 USB3_PHY Subsystem Environment

26.2.2.1 USB3_PHY I/O Signals

Table 26-10 represents module pins and their corresponding signal names at the device level, and also specifies their links to functions.

Table 26-10. USB3_PHY I/O Signals

Module Pin Name	Device-Level Signal Name	I/O(1)	Description	Module Pin Reset Value(2)
TXP	usb_txp0	O	TX output of the USB3_PHY differential transmission line	0
TXN	usb_txn0	O	TY output of the USB3_PHY differential transmission line	0
RXP	usb_rxp0	I	RX input of the USB3_PHY differential reception line	HiZ
RXN	usb_rxn0	I	RY input of the USB3_PHY differential reception line	HiZ

(1) I = Input; O = Output
(2) HiZ = High impedance

Figure 26-9 shows module pin signals mapping to USB3_PHY signals visible at the device pad level.

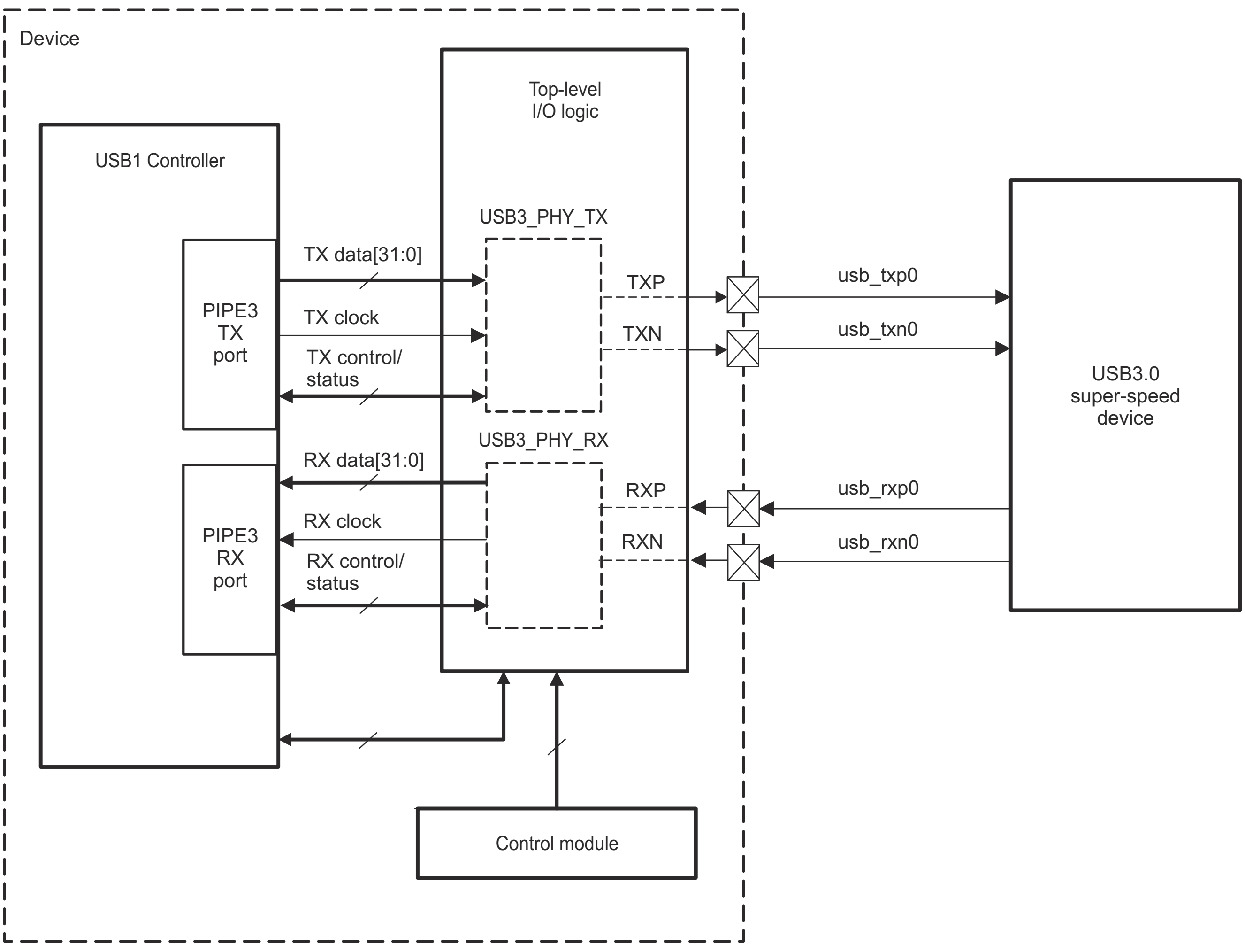


Figure 26-9. USB3_PHY I/O Signals

26.2.3 USB3_PHY Subsystem Integration

This section describes the USB3_PHY subsystem-related components (USB3_PHY module, DPLL_USB_OTG_SS, DPLLCTRL_USB_OTG_SS, and OCP2SPC1) integration in the device, including information about clocks, resets, and hardware requests.

Figure 26-10 shows the USB3_PHY integration.

The USB3_PHY module integration features:

- A low-power nonretention reset, L3INIT_RST
- (OCP2SPC1) Interconnect adapter target interface
- A high-frequency input clock (PLL_CLK) tied to the DPLL_USB_OTG_SS.CLKDCOLDO output
- A local port connected to the USB1 MAC PIPE3 port
- A clock output (PIPE_PCLK) to the USB1 PIPE port
- A clock output (PIPE_MCLK) from the USB1 PIPE port
- A common clock from PRCM - REF_CLK, to supply DPLL and PHY RX/TX logic
- PRCM.CORE_USB_OTG_SS_LFPS_TX_CLK clock tied to the USB3_PHY_TX
- A device CORE CONTROL MODULE command port to the power sequencer

The DPLL_USB_OTG_SS integration features:

- A low-power nonretention reset, L3INIT_RST
- A PRCM gated version of the device SYS_CLK (USB_OTG_SS_REF_CLK) supplying DPLL_USB_OTG_SS.CLKINP pin
- A single high-frequency clock output, DPLL_USB_OTG_SS.CLKDCOLDO, to the PLL_CLK input of the USB3_PHY_TX/RX components.
- No HS divider integration
- IDLE signaling implementation
- LDO and DCO PWRDNZ monitoring signals
- A DPLL_LOCK LOCKED status indication

The DPLLCTRL_USB_OTG_SS integration features:

- A low-power nonretention reset, L3INIT_RST
- (OCP2SPC1) Interconnect adapter target interface also providing the input SCP clock to the DPLLCTRL_USB_OTG_SS
- Configuration/control programming interface to the DPLL_USB_OTG_SS
- No direct gate control for DPLL_USB_OTG_SS.CLKINP and CLKDCOLDO

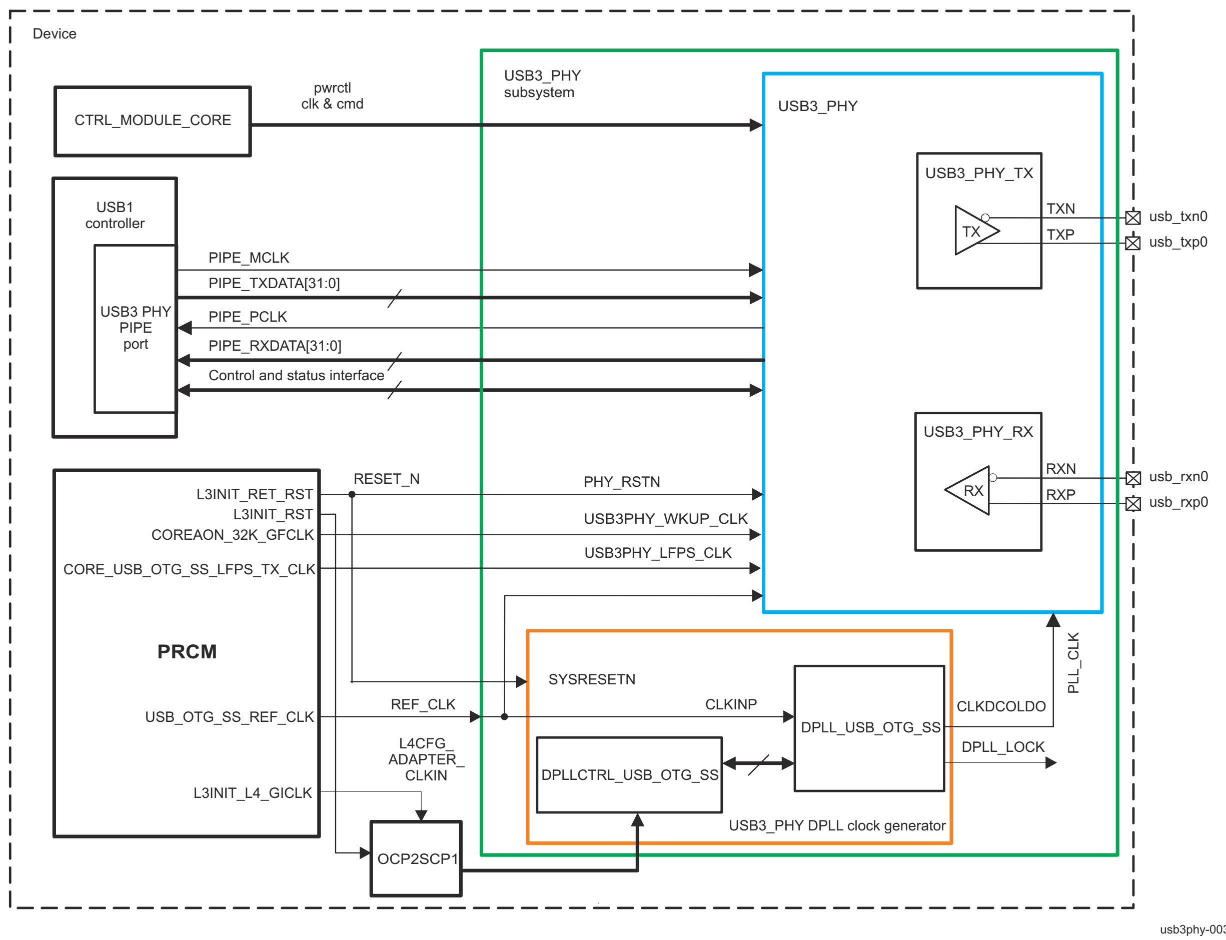


Figure 26-10. USB3_PHY Subsystem Integration

Table 26-11 through Table 26-13 summarize the integration of the module in the device.

Table 26-11. Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
USB3_PHY_TX	PD_L3INIT	OCP2SCP1 SCP interconnects
USB3_PHY_RX		OCP2SCP1 SCP interconnects
USB3_PHY (wrapper)		CTRL_CORE_MODULE power control
DPLLCTRL_USB_OTG_SS		OCP2SCP1 adapter SCP interconnect
DPLL_USB_OTG_SS	PD_COREAON	
OCP2SCP1		L4_CFG

Table 26-12. Clocks

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DPLL_USB_OTG_SS	CLKINP	USB_OTG_SS_REF_CLK[1]	PRCM	DPLL_USB_OTG_SS reference functional clock (SYS_CLK based)
USB3_PHY	USB3PHY_PWRS_CLK	USB_OTG_SS_REF_CLK	PRCM	USB3_PHY wrapper power sequencer functional clock (SYS_CLK)

Table 26-12. Clocks (continued)

USB3_PHY_TX	USB3PHY_LFPS_CLK	CORE_USB_OTG_SS_LFPS_TX_CLK	PRCM	Fixed frequency USB3TX functional clock used for LFPS pattern generation
USB3_PHY_RX	USB3PHY_WKUP_CLK	COREAON_32K_GFCLK	PRCM	I/O wakeup and debounce 32-kHz functional clock at USB3_PHY_RX Receiver side
OCP2SCP1	L4CFG_ADAPTER_CLKIN	L3INIT_L4_GICLK	PRCM	L4_CFG adapter interface clock

(1) This clock is connected to a single clock input pin - REF_CLK at the USB3_PHY subsystem level

Table 26-13. Resets

Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
USB3_PHY subsystem	RESET_N	L3INIT_RST	PRCM	A nonretention reset to all USB3_PHY subsystem components

Note

The USB3_PHY_RX generates a hardware wakeup request to the PRCM module.

26.2.4 USB3_PHY Subsystem Functional Description

26.2.4.1 Super-Speed USB PLL Controller L4 Interface Adapter Functional Description

The L4_CFG interconnect adapter (OCP2SCP1), allows user software to configure the DPLLCTRL_USB_OTG_SS registers over the L4_CFG port. Hence it is expected that this adapter is configured to operate before any programming of the DPLLCTRL_USB_OTG_SS.

Before initiating any software reset to this adapter, a value >= 0x6 should be written to the register bitfield OCP2SCP_TIMING[3:0] SYNC2.

L4_CFG interconnect adapter software reset is performed via writing OCP2SCP_SYSCONFIG[1] SOFTRESET to 0b1. The software reset completion is observed in bit OCP2SCP_SYSSTATUS[0] RESETDONE.

By default a smart-idle power mode is selected for OCP2SCP1. The smart-idle mode supported by OCP2SCP1 is not wake-up capable, which means software must explicitly take care to wake the OCP2SCP1 by setting the OCP2SCP_SYSCONFIG [4:3] IDLEMODE bit field to 0x1 (no idle), once it has previously gone to an idle mode.

By default OCP2SCP_SYSCONFIG[0] AUTOIDLE =0x1, which defines that the DPLLCTRL L4 interface adapter automaticaly gates its L4 input clock based on L4_CFG interconnect activity. To enable a free-running clock, one should set bit AUTOIDLE to 0x0.

26.2.4.2 USB3_PHY Serializer and Deserializer Functional Descriptions

26.2.4.2.1 USB3_PHY Module Resets

26.2.4.2.1.1 Hardware Reset

The USB3_PHY active low reset which is sourced from PRCM module. L3INIT_RST is hardware-asserted upon power up. For more information on the hardware reset source , see *Reset Domains*, in *Power, Reset, and Clock Management*.

26.2.4.2.1.2 Software Reset

The USB3_PHY PIPE i/f logic is software reset from USB1 over PIPE port by software assertion of the USBOTGSS_GUSB3PIPECTL[31] PHYSOFTRST bit to 0x1. The user must set USBOTGSS_GUSB3PIPECTL[31] PHYSOFTRST to 0x1 before triggering the serializer and deserializer power-up sequence in the CTRL_CORE_PHY_POWER_USB register. For more information on power-up sequence, see Section 26.2.4.2.3.1, *USB3_PHY Power-Up/Down Sequences*.

The user should deassert USBOTGSS_GUSB3PIPECTL[31] PHYSOFTRST to 0x0, after the serializer and deserializer power-up sequence completes. For more information, see Section 26.2.4.2.3.1, *USB3_PHY Power-Up/Down Sequences*.

26.2.4.2.2 USB3_PHY Subsystem Clocking

26.2.4.2.2.1 USB3_PHY Subsystem Input Clocks

The USB3_PHY component receives a feedback clock, PIPE_MCLK, which is the reflected version of the PIPE_PCLK (PIPE port synchronizing clock) generated by the USB3_PHY component. The clock is turned on/off according to the PIPE power-down port states. As PIPE port works in source-synchronous mode, all data movement from the MAC layer to the PIPE interface is synchronous to PIPE_PCLK.

The USB3_PHY.PLL_CLK high-speed transmission (2.5 GHz) clock input pin is connected to the DPLL_USB_OTG_SS clock output, CLKDCOLDO. For more information on the DPLL_USB_OTG_SS.CLKDCOLDO output clock settings, see Section 26.2.4.3.4.2, *USB3_PHY DPLL Output Clock Configuration*.

As shown in Figure 26-11, the same PRCM-sourced clock (USB_OTG_SS_REF_CLK), tied at the USB3_PHY subsystem REF_CLK input, supplies the USB3_PHY RX/TX components and the DPLL_USB_OTG_SS.CLKINP inputs.

The PHY associated power sequencer receives PRCM.USB_OTG_SS_REF_CLK as a functional clock . Software must notify the PHY logic about which REF_CLK frequency is selected by writing the CTRL_CORE_PHY_POWER_USB[31:22] USB_PWRCTL_CLK_FREQ bit field, as follows:

- 0x26: If SYSCLK = 38.4 MHz
- 0x1A: If SYSCLK = 26.0 MHz
- 0x13: If SYSCLK = 19.2 MHz
- 0x10: If SYSCLK = 16.8 MHz
- 0x0C: If SYSCLK = 12.0 MHz

The USB3PHY_LFPS_CLK clock input is used to support different USB3_PHY functions, such as the low-frequency periodic signaling (LFPS) generator. This USB3_PHY_TX clock input is tied to the PRCM.CORE_USB_OTG_SS_LFPS_TX_CLK functional clock.

The USB3_PHY_RX deserializer I/O wake-up logic is supported by the PRCM.COREAON_32K_GFCLK clock, applied at the USB3PHY_WKUP_CLK input.

26.2.4.2.2.2 USB3_PHY Subsystem Output Clocks

- PIPE_PCLK: The PIPE interface is source-synchronous. A PIPE_PCLK clock is generated by the USB3_PHY component, used to synchronize USB1 PIPE port inputs, and reflected back as the PIPE_MCLK along with the PIPE outputs. The clock is turned on/off according to the power control port. All data movement from the PIPE interface to the USB1 MAC layer is synchronous to this clock.

The PIPE_PCLK clock which drives the PIPE3 logic is sourced by the on-chip USB3_PHY. The PIPE interface is source-synchronous (that is, the clock is received from the PHY), used to synchronize USB1 PIPE inputs, and reflected back along with the PIPE outputs as the PIPE_MCLK. The PIPE_PCLK clock is turned on/off according to the USB3_PHY power control port. For more details, see Section 26.2.4.2.3, *USB3_PHY Power Management*.

26.2.4.2.3 USB3_PHY Power Management

From one side the control over power up/down states of the USB3_PHY is provided through a power sequencer module tightly integrated with the the USB3_PHY physical layer TX/RX components.

26.2.4.2.3.1 USB3_PHY Power-Up/-Down Sequences

Powering-up/-down the USB3_PHY_TX and USB3_PHY_RX modules is triggered through software writing corresponding power-up/down commands in the CTRL_CORE_PHY_POWER_USB register of the device core control module, as follows:

- CONTROL_PHY_POWER_USB[21:14] USB_PWRCTL_CLK_CMD=0x0 (default value) commands both USB3_PHY_TX and USB3_PHY_RX to power-down (OFF) state.
- CONTROL_PHY_POWER_USB[21:14] USB_PWRCTL_CLK_CMD=0x1 powers up the USB3_PHY_RX only.
- CONTROL_PHY_POWER_USB[21:14] USB_PWRCTL_CLK_CMD=0x2 powers up the USB3_PHY_TX only.
- CONTROL_PHY_POWER_USB[21:14] USB_PWRCTL_CLK_CMD=0x3 simultaneously powers up the USB3_PHY_RX and the USB3_PHY_TX

For more information, see *Control Module*.

26.2.4.2.3.2 USB3_PHY Low-Power Modes

An implemented powerdown control port allows USB3_PHY to support four low-power states:

- 0b00: P0, normal operation (used for all Polling, Configuration, Recovery, Loopback, and Hot_Reset states, and U0 state),
- 0b01: P1, low recovery time latency, power saving state (used for U1 state)
- 0b10: P2, longer recovery time latency, lower power state (used for U2, Rx.Detect and SS.Inactive states)
- 0b11: P2, lowest power state (used for SS.disabled and U3)

The USB3_PHY transitions from P0-active mode to low-power P1, or either of the two P2 states, is synchronized with USB1 transitions from U0 active state to U1, U2, and U3 low-power states through the input PIPE_POWERDOWN[1:0]. The PIPE port low-power behavior is configured from various bit fields inside the USB1.USB_GUSB3PIPECTL register. For more information on PIPE port low-power (LP) configuration, see *USB Register Description* in Section 24.7, *SuperSpeed USB DRD*.

26.2.4.2.3.3 Clock Gating

The USB3_PHY component high-speed clocks, PLL_CLK and PIPE_PCLK, are gated during the P1 low-power state.

26.2.4.2.4 USB3_PHY Hardware Requests

An asynchronous wake-up request is generated to the PRCM module by the USB3_PHY_RX I/Os (RXP, RXN). When the wake-up request is acknowledged, the DPLL_USB_OTG_SS and PIPE_PCLK are restarted.

Neither interrupt nor DMA requests are generated.

26.2.4.3 USB3_PHY Clock Generator Subsystem Functional Description

The DPLL_USB_OTG_SS, which is located outside the PRCM boundaries and is part of the USB1 controller subsystem, directly injects a high-speed clock into the USB3_PHY serializer/deserializer input, PLL_CLK. The DPLL generator is controlled through a programmable interface from a dedicated PLL controller, DPLLCTRL_USB_OTG_SS.

26.2.4.3.1 USB3_PHY DPLL Clock Generator Overview

The SCP interface of the USB3_PHY PLL controller (DPLLCTRL_USB_OTG_SS instance) is used to set the configuration of the DPLL modules, primarily the various counter values. Figure 26-11 is an overview of the DPLL clock generator embedded into the USB1 controller subsystem.

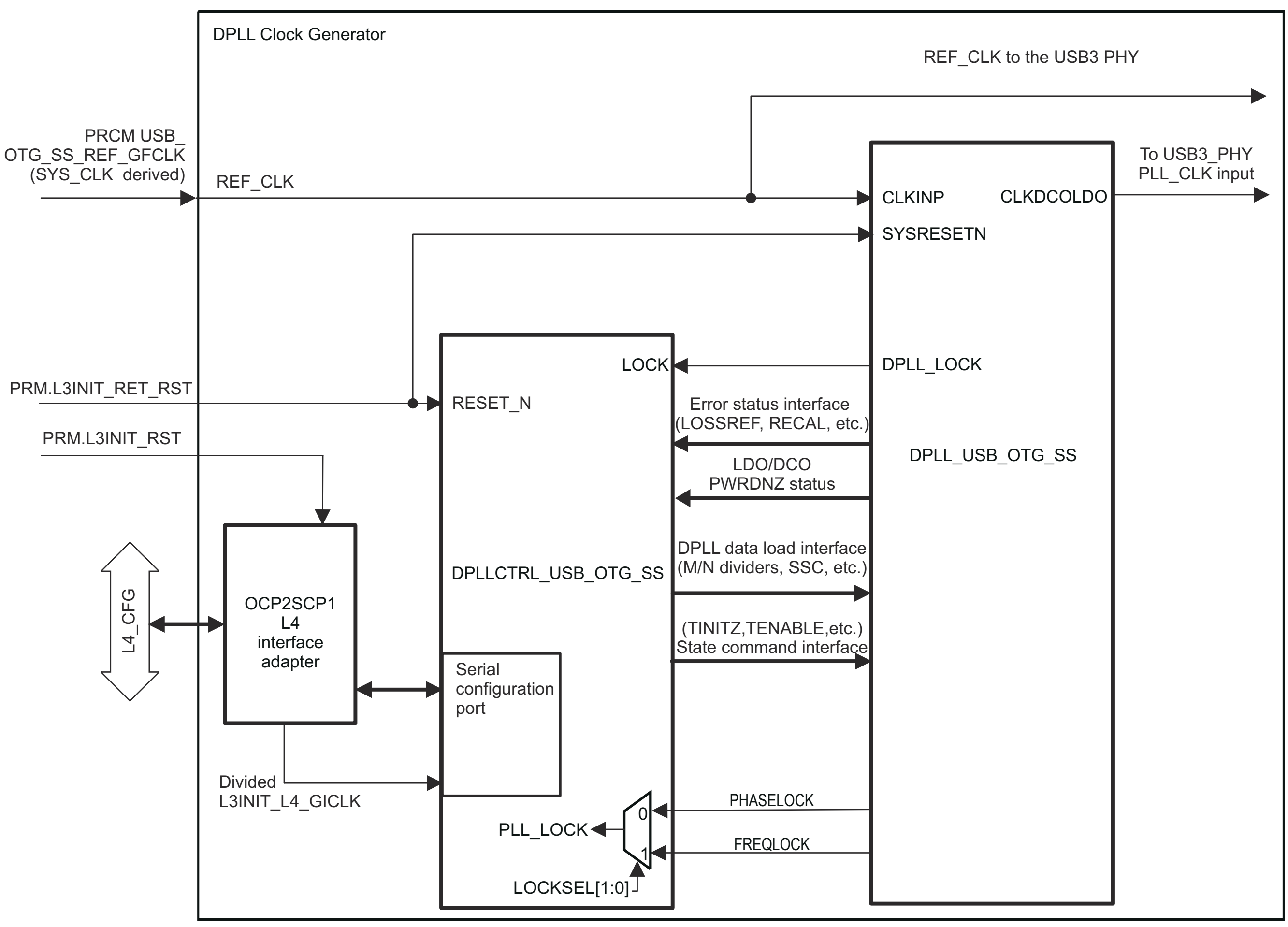


Figure 26-11. USB3_PHY DPLL Clock Generator Overview

The DPLL_USB_OTG_SS features:

- A programmable 8-bit input divider: N
- A programmable 12-bit integer, 18-bit fractional loop multiplier: M
- Digital control and loop filter
- Internal oscillator output clock on internal LDO domain (CLKDCOLDO output)
- DPLL output clock SSC (spread spectrum clocking) support
- No retention capabilities
- No Idle-bypass fast relock capabilities
- Idle-bypass low-power mode
- M/N bypass mode
- Relock from standby

The DPLLCTRL_USB_OTG_SS components features:

- DPLL error and status notification
- DPLL initialization and configuration
- DPLL lock criteria selectable between frequency and phase lock
- Idle command implementation
- No software reset implementation
- Automatic enable/disable control, synchronized with USB1 controller PIPE port commands to set USB3_PHY to P1, P2, and P3 low-power states

26.2.4.3.2 USB3_PHY DPLL Clock Generator Reset

The USB3_PHY PLL controller and USB3_PHY DPLL clock generator share a common hardware nonretention reset, L3INIT_RST, which comes from the device power and reset manager. Upon DPLL_USB_OTG_SS hardware reset completion, the DPLLCTRL_USB_OTG_SS.PLL_STATUS[0] PLLCTRL_RESET_DONE bit is automatically updated to 1. For more information on the hardware reset source, see *Reset Domains* in *Power, Reset, and Clock Management*.

The DPLLCTRL_USB_OTG_SS itself has no software reset capabilities.

DPLLCTRL_USB_OTG_SS performs a software reset sequence on the DPLL_USB_OTG_SS in hardware (through the TINITZ signal activation).

26.2.4.3.3 USB3_PHY DPLL Low-Power Modes

The power-management port (PMP) is not integrated for DPLLCTRL_USB_OTG_SS. The DPLL_USB_OTG_SS has no retention capabilities. This means, the DPLL digital power supply remains switched on during all modes of operation.

The low-power modes supported by DPLL_USB_OTG_SS are Idle-bypass low-power and MN-bypass modes, which are both characterized by:

- Internal LDO switched off
- DCO oscillator switched off
- CLKDCOLDO output pulled low

For more details on the PLL settings and conditions necessary to enter Idle-bypass and MN-bypass low-power modes, see Section 26.2.4.3.6.4, *USB3_PHY DPLL Idle-bypass low-power Mode* , and Section 26.2.4.3.6.5, *USB3_PHY DPLL MN-Bypass Mode*.

DPLL_USB_OTG_SS is held in a similar low-power state (DCO and LDO switched off, with CLKDCOLDO = 0) after Power-up Reset, before first PLL_GO command has been software triggered on the PLL controller. See Section 26.2.4.3.6.1, *USB3_PHY Clock Generator Power Up*.

26.2.4.3.4 USB3_PHY DPLL Clocks Configuration

26.2.4.3.4.1 USB3_PHY DPLL Input Clock Control

The DPLL_USB_OTG_SS accepts the functional clock, USB_OTG_SS_REF_CLK, on its CLKINP pin (REF_CLK input at USB3_PHY subsystem level) directly from the device PRCM, without involving any DPLLCTRL_USB_OTG_SS interactions. The USB_OTG_SS_REF_CLK is derived from SYS_CLK1. See *Clock Domain Module Attributes* in *Power, Reset, and Clock Management*.

If the CLKINP signal is lost for some time, the LOSSREF output signal, which serves as a feedback to DPLLCTRL_USB_OTG_SS, is asserted high. When CLKINP resumes, LOSSREF goes low (LOSSREF inactive state). The LOSSREF status signal can be software-monitored in the DPLLCTRL_USB_OTG_SS.PLL_STATUS[3] PLL_LOSSREF bit.

Note

DPLLCTRL_USB_OTG_SS has no software or hardware mechanisms to control DPLL_USB_OTG_SS input clock (CLKINP).

26.2.4.3.4.2 USB3_PHY DPLL Output Clock Configuration

Only the DPLL_USB_OTG_SS output, CLKDCOLDO, is used to provide the high-speed clock at the PLL_CLK pin of the USB3_PHY. Only the REGM, REGN, and SD divider values are used within the DPLL clock generator subsystem to adjust the CLKDCOLDO output clock frequency. This is done through programming the DPLLCTRL_USB_OTG_SS.DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[20:9], PLL_REGM, DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[8:1], PLL_REGN, and DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION3[17:10] PLL_SD bit fields, respectively. The USB3_PHY DPLL CLKOUT and CLKOUTLDO outputs are not used, and internal REGM2 and REGM1 dividers are not software controllable.

Note

At DPLL/DPLLCTRL integration level the PLL_REGM1[3:0] and PLL_REGM2[6:0] divider control signals are hardware tie-off to 0x0 and 0x1, respectively.

For more details on output clock settings sequence, see Section 26.2.4.3.7.3, *USB3_PHY DPLL Clock Programming Sequence*.

26.2.4.3.4.2.1 USB3_PHY DPLL Output Clock Gating

There is no direct software gate control for the DPLL_USB_OTG_SS.CLKDCOLDO output.

DPLL_USB_OTG_SS.CLKDCOLDO clock output is automatically gated (CLKDCOLDO pulled low) in the following scenarios:

- DPLL power-up sequence. For more information on power-up sequence, see Section 26.2.4.3.6.1, *USB3_PHY Clock Generator Power Up*.
- DPLL entering a relock sequence. For more information on relocking sequence, see Section 26.2.4.3.6.2, *USB3_PHY DPLL Sequences*.
- DPLL entering Idle-bypass low-power mode. For more information on idle-bypass mode, see Section 26.2.4.3.6.4, *USB3_PHY DPLL Idle-Bypass Mode*.
- DPLL entering MN-bypass mode. For more information on MN-bypass mode, see Section 26.2.4.3.6.5, *USB3_PHY DPLL MN-Bypass Mode*.

26.2.4.3.5 USB3_PHY DPLL Subsystem Architecture

Figure 26-12 is a simplified block diagram of the DPLL_USB_OTG_SS instance integration in the USB3_PHY clock generator subsystem.

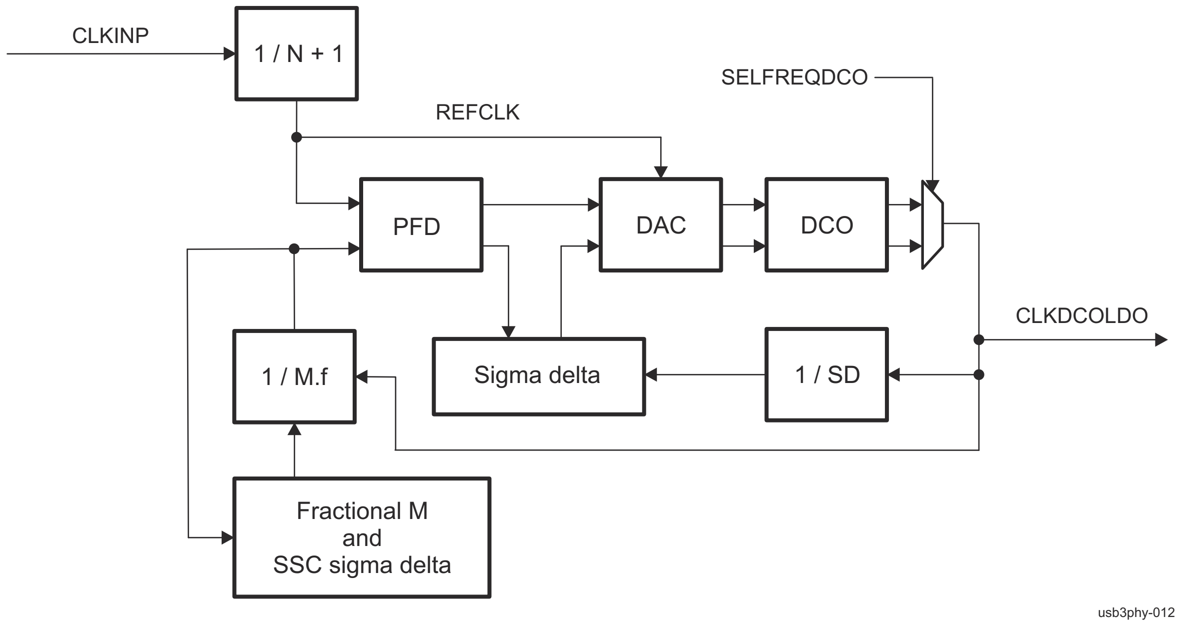


Figure 26-12. DPLL_USB_OTG_SS Functional Block Diagram

The input clock CLKINP goes to a predivider N + 1. The entire loop runs on the REFCLK clock after this predivider. The value of N + 1 is controlled through the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[8:1] PLL_REGN bit field.

The frequency ranges for the DPLL_USB_OTG_SS input clock - CLKINP and the DPLL internal reference clock, REFCLK = CLKINP/N + 1 are:

- 0.62 to 60 MHz for CLKINP
- 0.62 to 2.5 MHz for the REFCLK

The output clock CLKDCOLDO is synthesized by digitally controlled oscillator (the DCO block), that automatically detects the frequency range. The CLKDCOLDO frequency can be given with CLKDCOLDO = CLKINP×M/(N + 1). For that purpose the feedback multiplier M must be configured through the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[20:9] PLL_REGM bit field.

The DPLL_USB_OTG_SS module supports fractional synthesis (that is, the frequency multiplication factor M can be programmed as fractional). This is achieved by having a sigma delta feedback divider (M). A fractional value (Fractional M) of 18 bits is supported, thus enabling control for a better accuracy. Programming the 18-bit Fractional M value is done by setting the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION4[17:0] PLL_REGM_F bit field (similar to REGM). To enable integer only division, Fractional M should be set to 000…0.

Note

Fractional synthesis is not supported for M > 4093.

The module also supports SSC on its output clocks. SSC is used to spread the spectral peaking of the clock to reduce any electromagnetic interference (EMI). When SSC is enabled, the clock spectrum is spread by the amount of frequency spread, and the attenuation is given by the ratio of the frequency spread (df) and the modulation frequency (fm); that is, {10×log10(df/fm)} dB.

The SSC is performed by changing the feedback divider (M) in a triangular pattern, which means the frequency of the output clock varies in a triangular pattern. The frequency of the triangular pattern is modulation frequency (fm). The peak (dM) or the amplitude of the triangular pattern as a percent of M is equal to the percent of the frequency spread (df); that is, $dM/M = df/F_{OUT}$.

Because this is in-band modulation for the DPLL_USB_OTG_SS, the modulation frequency must be within the loop bandwidth of the DPLL. A higher modulation frequency would result in less spreading in the output clock.

The SSC can be enabled and disabled by asserting the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION1[0] EN_SSC bit. The acknowledge signal SSCACK, observed by the DPLLCTRL_USB_OTG_SS.PLL_STATUS[12] SSC_EN_ACK bit, notifies the exact start and end of SSC. When EN_SSC is deasserted, SSC is disabled only after completion of one full cycle of the triangular pattern given by the modulation frequency. This is done to maintain the average frequency.

The modulation frequency (fm) can be programmed as a ratio of REFCLK/4; that is, the value programmed in the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION2[29:20] MODFREQDIVIDER bit field must be = REFCLK/(4×fm).The ModFreqDivider is split into Mantissa and 2^{Exponent} (ModFreqDivider = ModFreqDividerMantissa × $2^{\text{ModFreqDividerExponent}}$).

- The Mantissa is controlled by bits [29:23] of the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION2[29:20] MODFREQDIVIDER bit field.
- The Exponent is controlled by bits [22:20] of the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION2[29:20] MODFREQDIVIDER bit field.

Although the same value of ModFreqDivider could be obtained by different combinations of Mantissa and Exponent values, it is preferred to get the target ModFreqDivider by programming maximum Mantissa and minimum Exponent values.

To define the frequency spread (df), dM must be controlled as previously explained. To define dM, the step size of M for each REFCLK during the triangular pattern must be programmed.This is defined as follows:

- dM =($2^{\text{ModFreqDividerExponent}}$) × ModFreqDividerMantissa × DeltaMStep, if ModFreqDividerExponent <= 3
- dM = 8 × ModFreqDividerMantissa × DeltaMStep, if ModFreqDividerExponent > 3

DeltaMStep value is split into integer part and fractional part, as follows:

- The MSB of 3-bit integer part, DeltaMStepInteger, is controlled by the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION2[30] DELTAM2 bit and the remaining LSBs by bits [19:18] of the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION2[19:0] DELTAM bit field.
- The 18-bit fractional part, DeltaMStepFraction, is controlled by bits [17:0] of the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION2[19:0] DELTAM bit field.

If the DPLLCTRL_USB_OTG_SS.PLL_SSC_CONFIGURATION1[2] DOWNSPREAD bit is set to 1, the frequency spread on the lower side is twice the programmed value. The frequency spread on the higher side is 0 (except for 20 percent overshot).

26.2.4.3.6 USB3_PHY DPLL Clock Generator Modes and State Transitions

The DPLL_USB_OTG_SS can be set in different modes during operation. DPLLCTRL triggers DPLL_USB_OTG_SS state transitions to different static modes by means of the TINITZ and TENABLE hardware control signals.

26.2.4.3.6.1 USB3_PHY Clock Generator Power Up

After power up, the DPLL_USB_OTG_SS.SYSRESETN input is automatically pulled low by the PRM, together with the DPLLCTRL_USB_OTG_SS.RESET_N input. Because PRM.L3INIT_RST is an asynchronous reset, the DPLL_USB_OTG_SS input clock (DPLL_USB_OTG_SS.CLKINP) is not demanded upon reset. The LOSSREF signal, which monitors the presence of CLKINP clock, remains 1 during SYSRESETN = 0 irrespective of presence/absence of the CLKINP clock. If CLKINP is present when reset is asserted, the LOSSREF signal is deasserted to 0, a certain time after the hardware reset completes. During DPLL power-up mode, CLKDCOLDO clock is maintained inactive (pulled low). After power-up reset, the DPLL_LOCK (internal lock loop) signal is maintained deasserted, too.

26.2.4.3.6.2 USB3_PHY DPLL Sequences

Once all the configuration values have been initially programmed into the DPLLCTRL_USB_OTG_SS registers (see Section 26.2.4.3.7.3) , the DPLLCTRL_USB_OTG_SS.PLL_GO[0] PLL_GO bit should be set to update the configuration values and start the DPLL calibration and locking sequence.

After the DPLLCTRL_USB_OTG_SS.PLL_GO[0] PLL_GO bit is set high in software, the DPLLCTRL_USB_OTG_SS state-machine takes the following action:

1. Sets the TINITZ signal to 0, which acts as a soft reset to DPLL_USB_OTG_SS. This starts the DPLL initialization procedure. During initialization mode, the CLKDCOLDO, FREQLOCK, and PHASELOCK signals are kept at 0 and the BYPASSACK signal is kept at 1.
   VDDA power supply of DPLL should be active before DPLL_USB_OTG_SS Initialization is performed, but it is not required to be switched on immediately after device power up.
2. The TENABLE signal is asserted high by DPLLCTRL hardware to load the user-programmed values of REGM, REGN, REGSD, and SELFREQDCO into DPLL registers.
3. After TENABLE is asserted, TINITZ is driven high (disabled) by the DPLLCTRL to trigger a DPLL calibration and lock sequence after the loop control values are loaded. The module calibration-lock sequence will begin from the first CLKINP edge after TINITZ is disabled.

Figure 26-13 summarizes the software and hardware sequences flow of DPLL_USB_OTG_SS.

Note

All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

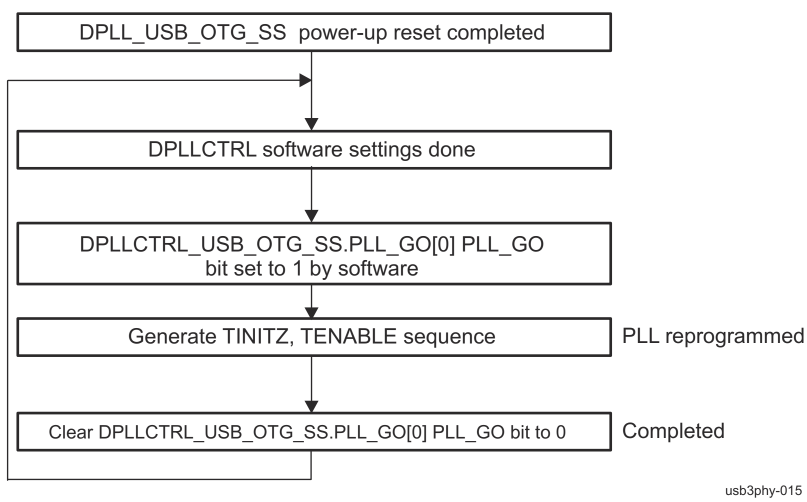


Figure 26-13. USB3_PHY PLL GO Sequence

DPLL_USB_OTG_SS relock sequence:

When the DPLL leaves a lost clock condition (LOSSREF = 1 → 0) or idle-bypass mode it enters relock sequence from the first CLKINP edge (after bypass mode leaving). Relock sequence is the same as calibration-lock sequence already described.

A DPLL relock sequence is also software triggered by setting the DPLLCTRL_USB_OTG_SS.PLL_GO[0] PLL_GO bit to 0b1 for DPLL parameters update.

When DPLL_USB_OTG_SS enters a relock sequence, CLKDCOLDO is pulled low. FREQLOCK and PHASELOCK status signals are also low. CLKDCOLDO output clock is activated after the FREQLOCK or PHASELOCK signal goes high, depending on the selected locking criteria.

The DPLLCTRL_USB_OTG_SS.PLL_GOUSB3PHY_PLL_GO[0] PLL_GO bit can be used by software to monitor if DPLLCTRL locking process is still pending (PLL_GO = 0b1).

26.2.4.3.6.3 USB3_PHY DPLL Locked Mode

When DPLL_USB_OTG_SS finishes calibration and lock sequences it enters a locked state. During the locked state, LOSSREF and BYPASSACK are deasserted, FREQLOCK or PHASELOCK is asserted.

DPLL lock event criteria (FREQLOCK or PHASELOCK) is software-selectable through the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[10:9] PLL_LOCKSEL bit field.

The DPLL signalizes the locked state to DPLLCTRL_USB_OTG_SS, USB_OTG_SS core controller, and USB3_PHY through assertion of the DPLL_LOCK signal, which reflects the internal lock loop status. The user software can monitor the DPLL locked event in the DPLLCTRL_USB_OTG_SS.PLL_STATUS[1] PLL_LOCK bit, which is active high.

26.2.4.3.6.4 USB3_PHY DPLL Idle-Bypass Mode

Idle-bypass fast relock mode is not supported for DPLL_USB_OTG_SS.

DPLL_USB_OTG_SS supports idle-bypass low-power mode. A transition from a normal operation to idle-bypass mode is performed when software sets the DPLLCTRL_USB_OTG_SS.DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[0] PLL_IDLE bit to 0x1. IDLE signal assertion triggers a power-down sequence on DPLL internal LDO analog blocks and the DCO oscillator.

In idle-bypass low-power mode, the PHASELOCK and FREQLOCK output signals are asserted low and CLKDCOLDO goes low. Also, the internal reference clock REFCLK = CLKINP/N + 1 is gated inside the DPLL digital control logic to save power.

In the functional mode, the DPLL_USB_OTG_SS.TICOPWDN output indicates to the PLL controller the status of the power-down signal (active high) of the internal DCO oscillator. The internal DCO oscillator is powered down in idle-bypass mode or during period from SYSRESETN 0->1 to module initialization. The DCO oscillator exits power-down (TICOPWDN goes low) whenever the module internally tries to lock/relock after initialization or exiting idle-bypass mode.

In the functional mode, DPLL_USB_OTG_SS.LDOPWDN output indicates to the PLL controller the status of the power-down signal (active high) of the internal LDO. LDOPWDN goes high as soon as the internal LDO is powered down. LDOPWDN goes low after the LDO output voltage is stable. The internal LDO is powered down in the period from SYSRESETN 0 → 1 to module initialization or when entering into idle-bypass mode. LDOPWDN is cleared whenever the module internally tries to lock/relock after initialization or exiting idle-bypass mode after the internal LDO output voltage has stabilized.

The DCO and LDO power ON and OFF states are reflected within the read-only DPLLCTRL_USB_OTG_SS.PLL_STATUS[16] PLL_TICOPWDN and DPLLCTRL_USB_OTG_SS.PLL_STATUS[15] LDOPWDN monitor bits.

To exit idle-bypass mode and restore clock generation, the user should write DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[0] PLL_IDLE to 0x0, which deasserts the IDLE signal, and DPLL_USB_OTG_SS automatically enters a relock sequence. CLKDCOLDO output clock is activated after the FREQLOCK or the PHASELOCK signal goes high, depending on selected locking criteria.

26.2.4.3.6.5 USB3_PHY DPLL MN-Bypass Mode

The MN-bypass mode will be activated if REGM = 0 or 1 is loaded into the module on the rising edge of TENABLE. TINITZ also should be pulsed to enter MN-bypass mode. The module enters a low-power mode by gating all its internal clocks (REFCLK) and powering down internal LDO (LDOPWDN = 1) and DCO (DCOPWDN = 1). CLKDCOLDO remains gated (low) during this mode.

Note

When the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[20:9] PLL_REGM bit field is updated to 0x0 or 0x1, the CLKDCOLDO is gated.

26.2.4.3.6.6 USB3_PHY DPLL Error Conditions

The PLL lock and recalibration signals can be monitored to detect the loss-of-lock condition and the DPLL requirement to recalibrate (caused by a large temperature change since the last lock request):

- The DPLLCTRL_USB_OTG_SS.PLL_STATUS[2] PLL_RECAL bit informs whether the DPLL_USB_OTG_SS must be recalibrated.

The PLL reference clock (CLKINP) loss status and PLL-in-high-jitter condition can also be monitored:

- The DPLLCTRL_USB_OTG_SS.PLL_STATUS[1] PLL_LOCK bit gives the USB3_PHY PLL lock state.
- The DPLLCTRL_USB_OTG_SS.PLL_STATUS[3] PLL_LOSSREF bit informs whether the DPLLCTRL_USB_OTG_SS has lost the reference clock.
- The DPLLCTRL_USB_OTG_SS.PLL_STATUS[5] PLL_HIGHJITTER bit informs whether the PLL has entered a high-jitter condition.

26.2.4.3.7 USB3_PHY PLL Controller Functions

26.2.4.3.7.1 USB3_PHY PLL Controller Register Access

The configuration registers are accessed through the OCP2SCP1 L4 adapter register space using the SCP interface of the DPLLCTRL_USB_OTG_SS. This includes all the configuration signals and returning status signals.

CAUTION

All writes must be 32-bit operations, because the SCP interface always transfers 32 bits; 16- or 8-bit operations may lead to unpredictable errors.

Note

Because the USB3_PHY directly provides parallel data interface clocks RX_CLK and TX_CLK to the USB1 MAC controller, the DPLLCTRL_USB_OTG_SS and DPLL_USB_OTG_SS must be configured before any data transfer between the USB1 controller MAC layer and an external USB3 device.

26.2.4.3.7.2

26.2.4.3.7.3 USB3_PHY DPLL Clock Programming Sequence

The DPLL_USB_OTG_SS factors must be calculated based on the required input and output frequencies, keeping the PLL internal reference frequency (REFCLK) in the appropriate range (0.62 to 2.5 MHz).

- REGM factor is programmed in the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[20:9] PLL_REGM bit field.
- Fractional part of REGM factor is programmed in the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION4[17:0] PLL_REGM_F bit field.
- REGN factor is programmed in the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[8:1] PLL_REGN bit field.
- DCO frequency range is set in the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[3:1] PLL_SELFREQDCO bit field.
  - PLL_SELFREQDCO should be set to 0x2 if 750 MHz < CLKDCOLDO [MHz] < 1500 MHz.
  - PLL_SELFREQDCO should be set to 0x4 if 1250 MHz < CLKDCOLDO [MHz] < 2500 MHz.
- SD divider is programmed in the DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION3[17:10] PLL_SD bit field. The DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[3:1] PLL_SELFREQDCO register bit field should be programmed depending on the value of CLKDCOLDO = CLKINP × M/(N + 1).

Figure 26-14 shows the formulae and programming sequence.

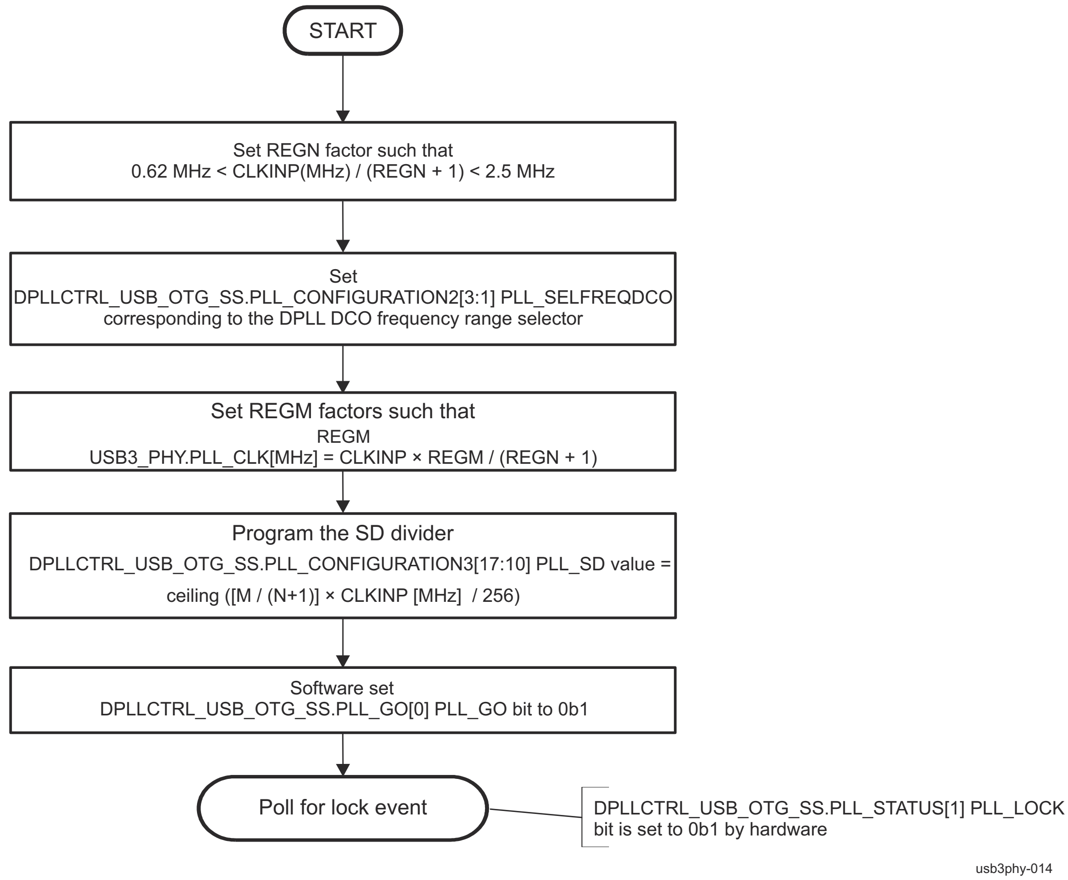


Figure 26-14. USB3_PHY PLL Programming Sequence

Note

- The equation for USB3_PHY_TX/USB3_PHY_RX (MHz) applies to the CLKDCOLDO of the DPLL_USB_OTG_SS.
- CLKDCOLDO output frequency of the DPLL_USB_OTG_SS should be programmed to 2.5 GHz, for the super-speed USB (5 Gbps) mode.

Table 26-14 summarizes the registers for the DPLLCTRL_USB_OTG_SS programming sequence.

Table 26-14. Register Call Summary for USB3_PHY PLL Programming Sequence

Register Name	Register Name	Register Name	Register Name
DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2	DPLLCTRL_USB_OTG_SS.PLL_GO	DPLLCTRL_USB_OTG_SS.PLL_STATUS	DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION3

26.2.4.3.7.4 USB3_PHY DPLL Recommended Values

Table 26-15 lists the DPLL_USB_OTG_SS recommended values.

Table 26-15. Recommended Programming Values

Field Name	Value	Description
DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[20:9] PLL_REGM	See [1].	Feedback clock divider
DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION1[8:1] PLL_REGN	See [1].	Reference clock divider

Table 26-15. Recommended Programming Values (continued)

Field Name	Value	Description
DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[10:9] PLL_LOCKSEL	0x-	Criteria to lock the PLL
DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[3:1] PLL_SELFREQDCO	See (1).	Program based on the PLL choice and lock frequency.
DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION2[0] PLL_IDLE	0	PLL active
DPLLCTRL_USB_OTG_SS.PLL_CONFIGURATION3[17:10] PLL_SD	See (1).	Ceiling { [PLL_REGM/(PLL_REGN+1)] × CLKINP(MHz)/256 }
DPLLCTRL_USB_OTG_SS.PLL_GO[0] PLL_GO	0x1	Write 1 when PLL is to be (re)locked with new parameters. This bit is cleared by hardware when the PLL request completes.

(1) The value of the bit field must be set according to the desired clock frequency.

26.2.5 USB3_PHY Subsystem Low-Level Programming Model

The low-level programming sequence to set up the USB3_PHY subsystem for USB superspeed I/O operations is summarized in the Table 26-16

Table 26-16. USB3_PHY Subsystem Low-Level Programming Sequence

Step	Description	Comment
1.	Set the startup low performance OPP in the appropriate PRCM registers.	For more information regarding demanded OPP, see the device *Data Manual*.
2.	Enable the PRCM.USB_OTG_SS_REF_CLK.	See *Clock Domain Module Attributes*, in *Power, Reset, and Clock Management*.
3.	Enable the PRCM.L3INIT_L4_GICLK to enable the OCP2SCP1 interface adapter operation.	See *Clock Domain Module Attributes*, in *Power, Reset, and Clock Management*.
4.	Software reset the OCP2SCP1 and poll until soft reset completion is indicated in status.	See Section 26.2.4.1.
5.	Set up division ratio between the OCP clock (PRCM.L3INIT_L4_GICLK) and SCP clock to supply the serial configuration register domains of the DPLLCTRL_USB_OTG_SS .	See Section 26.2.4.1.
6.	Set up necessary SYNC1 and SYNC2 timings to ensure no blocking of transactions over the SCP bus.	See Section 26.2.4.1. After this step, the user is ready to access the DPLLCTRL_USB_OTG_SS registers.
7.	Configure DPLL_USB_OTG_SS to generate frequency (CLKDCOLDO) = 2.5 GHz.	See Section 26.2.4.3.7.3.
8.	Software-assert the DPLLCTRL_USB_OTG_SS.PLL_GO[0] PLL_GO bit to 0x1	Start the DPLL lock with desired parameters.
9.	Poll the DPLLCTRL_USB_OTG_SS.PLL_STATUS[1] PLL_LOCK bit until it reads 1.	DPLL locked event
10.	Perform a USB3_PHY tuning required for the Super-Speed OTG USB i/f operation	Follow steps described in Table 26-17, *USB3_PHY Tuning Table*.
11.	Set USBOTGSS_GUSB3PIPECTL[31] PHYSOFTRST to 0x1	Software reset the USB3_PHY over USB OTG SS controller PIPE port. Note that this reset does not impact settings made in step 10.
12.	Software-trigger the USB3_PHY_TX power-up sequence.	For more details, see Section 26.2.4.2.3.1.
13.	Software-trigger the USB3_PHY_RX power-up sequence.	For more details, see Section 26.2.4.2.3.1.
14.	Clear USBOTGSS_GUSB3PIPECTL[31] PHYSOFTRST to 0x0	Deassert the software reset bit after power-up sequence completion is indicated

Table 26-17. USB3_PHY Tuning Table

Physical address[(2)] [bits to modify]	Preferred value setting[(1)]
0x4A08 440C [31:27]	0b10000
0x4A08 440C [17:14]	0b1010

Table 26-17. USB3_PHY Tuning Table (continued)

Physical address(2) [bits to modify]	Preferred value setting(1)
0x4A08 4428 [23:11]	0b1110001100110
0x4A08 4428 [28:26]	0b001
0x4A08 440C [6:5]	0b00
0x4A08 441C [31:30]	0b10
0x4A08 4424 [31:30]	0b11
0x4A08 4438 [10:7]	0b1001
0x4A08 4438 [2]	0b0

(1) These are the preferred settings of USB3_PHY, in case that USB3_PHY PLL_CLK=2.5 GHz.
(2) Accesses to these locations have to be done in 32-bits only. User must not modify USB3_PHY bits different than those described in Table 26-17.

26.3 USB3 PHY and SATA PHY Register Manual

This chapter summarizes and describes the registers for USB3 PHY and SATA PHY subsystems.

26.3.1 USB3 PHY and SATA PHY Instance Summary

Note

For OCP2SCP1 registers, please refer to Section 26.4.6 , *PCIe PHY Subsystem Register Manual*.

Table 26-18. USB3 PHY and SATA PHY Instance Summary

Module Name	Module Base Address	Size
USB3_PHY_RX	0x4A08 4400	128 bytes
USB3_PHY_TX	0x4A08 4800	100 bytes
DPLLCTRL_USB_OTG_SS	0x4A08 4C00	64 bytes
DPLLCTRL_SATA	0x4A09 6800	64 bytes

26.3.2 USB3_PHY_RX Registers

26.3.2.1 USB3_PHY_RX Register Summary

Table 26-19. USB3_PHY_RX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	USB3_PHY_RX Physical Address
RESERVED	R	32	0x0000 0000	0x4A08 4400
RESERVED	R	32	0x0000 0004	0x4A08 4404
RESERVED	R	32	0x0000 0008	0x4A08 4408
USB3PHYRX_ANA_PROGRAMMABILITY_REG1	RW	32	0x0000 000C	0x4A08 440C
RESERVED	R	32	0x0000 0010	0x4A08 4410
RESERVED	R	32	0x0000 0014	0x4A08 4414
RESERVED	R	32	0x0000 0018	0x4A08 4418
USB3PHYRX_TRIM_REG4	RW	32	0x0000 001C	0x4A08 441C
RESERVED	R	32	0x0000 0020	0x4A08 4420
USB3PHYRX_DLL_REG1	RW	32	0x0000 0024	0x4A08 4424
USB3PHYRX_DIGITAL_MODES_REG1	RW	32	0x0000 0028	0x4A08 4428
RESERVED	R	32	0x0000 002C	0x4A08 442C
RESERVED	R	32	0x0000 0030	0x4A08 4430
RESERVED	R	32	0x0000 0034	0x4A08 4434
USB3PHYRX_EQUALIZER_REG1	RW	32	0x0000 0038	0x4A08 4438
RESERVED	R	32	0x0000 003C	0x4A08 443C
RESERVED	R	32	0x0000 0040	0x4A08 4440
RESERVED	R	32	0x0000 0044	0x4A08 4444
RESERVED	R	32	0x0000 0048	0x4A08 4448
RESERVED	R	32	0x0000 004C	0x4A08 444C
RESERVED	R	32	0x0000 0050	0x4A08 4450
RESERVED	R	32	0x0000 0054	0x4A08 4454
RESERVED	R	32	0x0000 0058	0x4A08 4458
RESERVED	R	32	0x0000 005C	0x4A08 445C
RESERVED	R	32	0x0000 0060	0x4A08 4460
RESERVED	R	32	0x0000 0064	0x4A08 4464

26.3.2.2 USB3_PHY_RX Register Description

Table 26-20. USB3PHYRX_ANA_PROGRAMMABILITY_REG1

Address Offset	0x0000 000C		
Physical Address	0x4A08 440C	**Instance**	USB3_PHY_RX
Description	Some programmability for different analog circuits in the PHY.		
Type	RW		

31:8	7	6:5	4:0
MEM_ANAMODE	RESERVED	MEM_PLLDIV	RESERVED

Bits	Field Name	Description	Type	Reset
31:8	MEM_ANAMODE	Programmability for Analog circuits in the IP. The top 5 bits - MEM_ANAMODE[31:27] indicate the serial Interface using this PHY module. To select USB Super-Speed interface mode, user must set bit MEM_ANAMODE [31] to '0b1', and the other bits must be written to '0b0', as follows: **MEM_ANAMODE[31:27] = '0b10000' for the USB Super-Speed.** Bits [17:14] of the MEM_ANAMODE bitfield are used to control loss-of-signal detection (LOSD) threshold.	RW	0x00 0000
7	RESERVED		RW	0
6:5	MEM_PLLDIV	This is a test mode. SoC Users are requested to leave this at default value. The input PLL_CLK (after being muxed with PLLBYPCLK) is divided by the following factors indicated by this register. 00=1 01=2 10=4 11=RESERVED. All references to PLL_CLK in this register descriptions are AFTER considering this division.	RW	0x0
4:0	RESERVED		R	0x00

Table 26-21. USB3PHYRX_TRIM_REG4

Address Offset	0x0000 001C		
Physical Address	0x4A08 441C	**Instance**	USB3_PHY_RX
Description	The IP requires some values to be remembered in EFUSE. This register provides an alternative to EFUSE.		
Type	RW		

31	30:0
MEM_DLL_TRIM_SEL	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	MEM_DLL_TRIM_SEL	Determines which of the 4 EFUSE registers EFUSE_dll_rateN_coarsetrim should be used as the trim code by the DLL. This feature is so that the user may find and store the trim codes corresponding to different (at most 4) DLL frequencies (PLL_CLK (after the bypassing by MEM_EN_PLLBYP and the division by MEM_PLLDIV) frequencies) and at wake-up, instruct the IP to choose one of these available trim values depending on the Application's frequency requirement. 00 selects dll_rate0_coarsetrim 01 selects dll_rate1_coarsetrim 10 selects dll_rate2_coarsetrim 11 selects dll_rate3_coarsetrim.	RW	0x0
29:0	RESERVED		RW	0x0000 0000

Table 26-22. USB3PHYRX_DLL_REG1

Address Offset	0x0000 0024		
Physical Address	0x4A08 4424	**Instance**	USB3_PHY_RX
Description	This register is used to program DLL settings.		
Type	RW		

31	30	29–0
MEM_DLL_PHINT_RATE		RESERVED

Bits	Field Name	Description	Type	Reset
31:30	MEM_DLL_PHINT_RATE	Programs the DLL and the Phase Interpolator analog circuits to work with different clock frequencies. The frequency of PLL_CLK (after the bypassing by MEM_EN_PLLBYP and the division by MEM_PLLDIV) should be indicated by this register. 00=0.625GHz to 0.75GHz 01=RESERVED 10=1.25GHz to 1.5GHz 11=2.5GHz to 2.9GHz.	RW	0x3
29:0	RESERVED		R	0x00A4 1915

Table 26-23. USB3PHYRX_DIGITAL_MODES_REG1

Address Offset	0x0000 0028		
Physical Address	0x4A08 4428	**Instance**	USB3_PHY_RX
Description	This register contains control bits which affect different circuits in digital section of the PHY.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MEM_INV_RXPN_PAIR	MEM_OVRD_INV_RXPN_PAIR	RESERVED	MEM_HS_RATE	MEM_OVRD_HS_RATE	RESERVED	MEM_CDR_FASTLOCK	MEM_CDR_LBW	MEM_CDR_STEPCNT	MEM_CDR_STL	MEM_CDR_THR	MEM_CDR_THR_MODE	MEM_CDR_2NDO_SDM_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31	MEM_INV_RXPN_PAIR	If '1', interchanges RXP and RXN effectively by inverting the received data samples.	RW	0
30	MEM_OVRD_INV_RXPN_PAIR	Pin override control. See register bit MEM_INV_RXPN_PAIR.	RW	0
29	RESERVED		R	0
28:27	MEM_HS_RATE	Determines the ratio of PLL_CLK (after the bypassing by MEM_EN_PLLBYP and the division by MEM_PLLDIV) frequency and the output data rate. Full Rate means PLL_CLK (after the bypassing by MEM_EN_PLLBYP and the division by MEM_PLLDIV) frequency = Data Rate/2 00=Full Rate 01=Half Rate 10=Quarter Rate 11=RESERVED. This takes effect only if register bit MEM_OVRD_HS_RATE is '1', else the same is controlled by input pins hs_rate.	RW	0x0
26	MEM_OVRD_HS_RATE	Pin override control. See register bit MEM_HS_RATE.	RW	0
25:24	RESERVED		R	0x2
23	MEM_CDR_FASTLOCK	'1' to reduce lock time of CDR (clock-data-recovery circuit).	RW	1
22:21	MEM_CDR_LBW	CDR band-width control.	RW	0x3
20:19	MEM_CDR_STEPCNT	CDR 2nd order setting.	RW	0x0
18:16	MEM_CDR_STL	CDR settling time. Determines the number of vote clocks to blank ELV (Early-Late-Voter circuit) after update of phase.	RW	0x3
15:13	MEM_CDR_THR	CDR 1st order threshold. Determines how much early/ late votes should differ by before a phase change in the receiver sampling clock is triggered.	RW	0x1
12	MEM_CDR_THR_MODE	CDR 1st order threshold.	RW	1
11	MEM_CDR_2NDO_SDM_MODE	If '1', the 2nd Order CDR block uses a 1st order Sigma Delta Modulator to accomplish frequency offset If '0', a simple rate transformer is used for the same purpose.	RW	0
10:0	RESERVED		R	0x000

Table 26-24. USB3PHYRX_EQUALIZER_REG1

Address Offset	0x0000 0038		
Physical Address	0x4A08 4438	**Instance**	USB3_PHY_RX

Table 26-24. USB3PHYRX_EQUALIZER_REG1 (continued)

Description	The IP has an Equalizer (with analog and digital parts) which addresses Inter Symbol Interference (ISI). This register is for its controllability. The equalizer can be configured via the EQCTL bits which are part of the SCP register. The options are: No adaptive equalisation. The equalizer provides a flat response at the maximum gain. This setting may be appropriate if jitter at the receiver occurs predominantly as a result of crosstalk rather than frequency dependent loss. Fully adaptive equalisation. Both the low frequency gain and zero position of the equalizer are determined algorithmically by analysing the data patterns and transition positions in the received data. This setting should be used for most applications. Partially adaptive equalisation. The low frequency gain of the equalizer is determined algorithmically by analysing the data patterns and transition positions in the received data. The zero position is fixed in one of eight zero positions. When enabled, the receiver equalization logic analyzes data patterns and transition times to determine whether the low frequency gain of the equalizer should be increased or decreased. For the fully adaptive setting (EQCTL = 0001), if the low frequency gain reaches the minimum value, the zero frequency is then reduced. Likewise, if it reaches the maximum value, the zero frequency is then increased.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM_EQLEV																MEM_EQFTC					MEM_EQCTL				RESERVED				MEM_OVRD_EQLEV	MEM_OVRD_EQFTC	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	MEM_EQLEV	Equalizer level control.	RW	0x0000
15:11	MEM_EQFTC	Equalizer zero freq control.	RW	0x00
10:7	MEM_EQCTL	0000 - Equalizer disabled 0001 - Fully adaptive; FTC normal 0010 - Fully adaptive; FTC inverted 0011 - Hold eq state 01xx - Init eq to fully adaptive start/midpoint 1000 - Partially adaptive; zero=1084 MHz 1001 - Partially adaptive; zero= 805 MHz 1010 - Partially adaptive; zero= 573 MHz 1011 - Partially adaptive; zero= 402 MHZ 1100 - Partially adaptive; zero= 304 MHz 1101 - Partially adaptive; zero= 216 MHz 1110 - Partially adaptive; zero= 156 MHz 1111 - Partially adaptive; zero= 135 MHz	RW	0x0
6:3	RESERVED		R	0
2	MEM_OVRD_EQLEV	Continuosly forces the Equalizer output with the eqlev[15:0].	RW	0
1	MEM_OVRD_EQFTC	Continuosly forces the Equalizer output with the eqftc[4:0].	RW	0
0	RESERVED		R	0

26.3.3 USB3_PHY_TX Registers

26.3.3.1 USB3_PHY_TX Register Summary

Table 26-25. USB3_PHY_TX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	USB3_PHY_TX Physical Address
RESERVED	R	32	0x0000 0000	0x4A08 4800
RESERVED	R	32	0x0000 0004	0x4A08 4804
RESERVED	R	32	0x0000 0008	0x4A08 4808
USB3PHYTX_FUNC_CONFIG_REG	RW	32	0x0000 000C	0x4A08 480C
RESERVED	R	32	0x0000 0010	0x4A08 4810
RESERVED	R	32	0x0000 0014	0x4A08 4814
RESERVED	R	32	0x0000 0018	0x4A08 4818
RESERVED	R	32	0x0000 001C	0x4A08 481C
RESERVED	R	32	0x0000 0020	0x4A08 4820
RESERVED	R	32	0x0000 0024	0x4A08 4824
RESERVED	R	32	0x0000 0028	0x4A08 4828
USB3PHYTX_TEST_CONFIG_REG	RW	32	0x0000 002C	0x4A08 482C
USB3PHYTX_PATTGEN_PRELOAD	RW	32	0x0000 0030	0x4A08 4830
RESERVED	R	32	0x0000 0034	0x4A08 4834

26.3.3.2 USB3_PHY_TX Register Description

Table 26-26. USB3PHYTX_FUNC_CONFIG_REG

Address Offset	0x0000 000C		
Physical Address	0x4A08 480C	**Instance**	USB3_PHY_TX
Description	Functional Configuration registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM_INVPAIR	RESERVED																														

Bits	Field Name	Description	Type	Reset
31	MEM_INVPAIR	Invert polarity of TXP/TXN	RW	0
30:0	RESERVED		R	0x0000 0000

Table 26-27. USB3PHYTX_TEST_CONFIG_REG

Address Offset	0x0000 002C		
Physical Address	0x4A08 482C	**Instance**	USB3_PHY_TX
Description	Test related configuration registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MEM_EN_LPBK	MEM_ENTXPATT	MEM_TESTPATT	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED	Keep at 0	RW	0
30	MEM_EN_LPBK	Loopback enable for test	RW	0
29	MEM_ENTXPATT	Enable Test pattern to input of the serializer instead of TD	RW	0
28:26	MEM_TESTPATT	Select the LFSR mode to generate the required pattern 000: 31-bit LFSR mode 011: 23-bit LFSR mode 010: 7-bit LFSR mode 001: generate 1010 pattern 100: Fixed 31-bit value from PATTGEN_PRELOAD_VAL	RW	0x0
25:0	RESERVED		RW	0x0

Table 26-28. USB3PHYTX_PATTGEN_PRELOAD

Address Offset	0x0000 0030		
Physical Address	0x4A08 4830	**Instance**	USB3_PHY_TX
Description	Pattern generator (31 bit) LFSR Seed or preload value		
Type	RW		

31–1	0
MEM_PATTGEN_PRELOAD_VAL	RESERVED

Bits	Field Name	Description	Type	Reset
31:1	MEM_PATTGEN_PRELOAD_VAL	Preload value to the LFSR pattern generator	RW	0x0000 0000
0	RESERVED		RW	0

26.3.4 SATA_PHY_RX Registers

26.3.4.1 SATA_PHY_RX Register Summary

Table 26-29. SATA_PHY_RX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SATA_PHY_RX Physical Address
RESERVED	R	32	0x0000 0000	0x4A09 6000
RESERVED	R	32	0x0000 0004	0x4A09 6004
RESERVED	R	32	0x0000 0008	0x4A09 6008
SATAPHYRX_ANA_PROGRAMMABILITY_REG1	RW	32	0x0000 000C	0x4A09 600C
RESERVED	R	32	0x0000 0010	0x4A09 6010
RESERVED	R	32	0x0000 0014	0x4A09 6014
RESERVED	R	32	0x0000 0018	0x4A09 6018
SATAPHYRX_TRIM_REG4	RW	32	0x0000 001C	0x4A09 601C
RESERVED	R	32	0x0000 0020	0x4A09 6020
SATAPHYRX_DLL_REG1	RW	32	0x0000 0024	0x4A09 6024
SATAPHYRX_DIGITAL_MODES_REG1	RW	32	0x0000 0028	0x4A09 6028
RESERVED	R	32	0x0000 002C	0x4A09 602C
RESERVED	R	32	0x0000 0030	0x4A09 6030
RESERVED	R	32	0x0000 0034	0x4A09 6034
SATAPHYRX_EQUALIZER_REG1	RW	32	0x0000 0038	0x4A09 6038
RESERVED	R	32	0x0000 003C	0x4A09 603C
RESERVED	R	32	0x0000 0040	0x4A09 6040
SATAPHYRX_IO_AND_A2D_OVERRIDES_REG1	R W	32	0x0000 0044	0x4A09 6044
RESERVED	R	32	0x0000 0048	0x4A09 6048
RESERVED	R	32	0x0000 004C	0x4A09 604C
RESERVED	R	32	0x0000 0050	0x4A09 6050
RESERVED	R	32	0x0000 0054	0x4A09 6054
RESERVED	R	32	0x0000 0058	0x4A09 6058
RESERVED	R	32	0x0000 005C	0x4A09 605C
RESERVED	R	32	0x0000 0060	0x4A09 6060
RESERVED	R	32	0x0000 0064	0x4A09 6064

26.3.4.2 SATA_PHY_RX Register Description

Table 26-30. SATAPHYRX_ANA_PROGRAMMABILITY_REG1

Address Offset	0x0000 000C
Physical Address	**Instance**
Description	Some programmability for different analog circuits in the IP.
Type	RW

31-8	7	6-5	4-0
MEM_ANAMODE	RESERVED	MEM_PLLDIV	RESERVED

Bits	Field Name	Description	Type	Reset
31:8	MEM_ANAMODE	Programmability for Analog circuits in the IP. The top 5 bits -MEM_ANAMODE[31:27] indicate the Serial Interface using this PHY module. The bits MEM_ANAMODE [30:29] correspond to SATA 1.5 Gbps and SATA 3 Gbps modes, respectively. The appropriate bit must be set to '0b1' according to selected SATA speed mode, and the other bits must be written to '0b0', as follows: **MEM_ANAMODE[31:27] = 0b01000 for SATA 1.5 Gbps** **MEM_ANAMODE[31:27] = 0b00100 for SATA 3 Gbps** Bits [17:14] of the MEM_ANAMODE bitfield are used to control loss-of-signal detection (LOSD) threshold.	RW	0x00 0000
7	RESERVED		R	0x0
6:5	MEM_PLLDIV	This is a test mode. SoC Users are requested to leave this at default value. The input pll_clk (after being muxed with pllbypclk) is divided by the following factors indicated by this register. 00=1 01=2 10=4 11=RESERVED. All references to pll_clk in this register descriptions are AFTER considering this division.	RW	0x0
4:0	RESERVED		R	0x00

Table 26-31. SATAPHYRX_TRIM_REG4

Address Offset	0x0000 001C
Physical Address	**Instance**
Description	The IP requires some values to be remembered in EFUSE. This register provides an alternative to EFUSE.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM_DLL_TRIM_SEL		RESERVED																													

Bits	Field Name	Description	Type	Reset
31:30	MEM_DLL_TRIM_SEL	Determines which of the 4 EFUSE registers EFUSE_dll_rateN_coarsetrim should be used as the trim code by the DLL. This feature is so that the user may find and store the trim codes corresponding to different (at most 4) DLL frequencies (pll_clk pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) frequencies) and at wake-up, instruct the IP to choose one of these available trim values depending on the Application's frequency requirement. 00 selects dll_rate0_coarsetrim 01 selects dll_rate1_coarsetrim 10 selects dll_rate2_coarsetrim 11 selects dll_rate3_coarsetrim.	RW	0x0
29:0	RESERVED		RW	0x0000 0000

Table 26-32. SATAPHYRX_DLL_REG1

Address Offset	0x0000 0024
Physical Address	**Instance**
Description	This register is used to program DLL settings.
Type	RW

31–30	29–0
MEM_DLL_PHINT_RATE	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	MEM_DLL_PHINT_RATE	Programs the DLL and the Phase Interpolator analog circuits to work with different clock frequencies. The frequency of pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) should be indicated by this register. 00=0.625GHz to 0.75GHz 01=RESERVED 10=1.25GHz to 1.5GHz 11=2.5GHz to 2.9GHz.	RW	0x3
29:0	RESERVED		R	0x00A4 1915

Table 26-33. SATAPHYRX_DIGITAL_MODES_REG1

Address Offset	0x0000 0028
Physical Address	**Instance**
Description	This register contains control bits which affect different circuits in digital section of the IP.
Type	RW

31	30	29	28–27	26	25–24	23	22–21	20–19	18–16	15–13	12	11	10–0
MEM_INV_RXPN_PAIR	MEM_OVRD_INV_RXPN_PAIR	RESERVED	MEM_HS_RATE	MEM_OVRD_HS_RATE	RESERVED	MEM_CDR_FASTLOCK	MEM_CDR_LBW	MEM_CDR_STEPCNT	MEM_CDR_STL	MEM_CDR_THR	MEM_CDR_THR_MODE	MEM_CDR_2ND_O_SDM_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31	MEM_INV_RXPN_PAIR	If '1', interchanges RXP and RXN effectively by inverting the received data samples.	RW	0
30	MEM_OVRD_INV_RXPN_PAIR	Pin override control. See register bit MEM_inv_rxpn_pair.	RW	0
29	RESERVED		R	0
28:27	MEM_HS_RATE	Determines the ratio of pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) frequency and the output data rate. Full Rate means pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) frequency = Data Rate/2 00=Full Rate 01=Half Rate 10=Quarter Rate 11=RESERVED. This takes effect only if register bit MEM_ovrd_hs_rate is '1', else the same is controlled by input pins hs_rate.	RW	0x0
26	MEM_OVRD_HS_RATE	Pin override control. See register bit MEM_hs_rate.	RW	0
25:24	RESERVED		R	0x2
23	MEM_CDR_FASTLOCK	'1' to reduce lock time of CDR (clock-data-recovery circuit).	RW	1

Bits	Field Name	Description	Type	Reset
22:21	MEM_CDR_LBW	CDR band-width control.	RW	0x3
20:19	MEM_CDR_STEPCNT	CDR 2nd order setting.	RW	0x0
18:16	MEM_CDR_STL	CDR settling time. Determines the number of vote clocks to blank ELV (Early-Late-Voter circuit) after update of phase.	RW	0x3
15:13	MEM_CDR_THR	CDR 1st order threshold. Determines how much early/ late votes should differ by before a phase change in the receiver sampling clock is triggered.	RW	0x1
12	MEM_CDR_THR_MODE	CDR 1st order threshold.	RW	1
11	MEM_CDR_2NDO_SDM_MODE	If '1', the 2nd Order CDR block uses a 1st order Sigma Delta Modulator to accomplish frequency offset If '0', a simple rate transformer is used for the same purpose.	RW	0
10:0	RESERVED		R	0x000

Table 26-34. SATAPHYRX_EQUALIZER_REG1

Address Offset	0x0000 0038
Physical Address	**Instance**
Description	The IP has an Equalizer (with analog and digital parts) which addresses Inter Symbol Interference (ISI). This register is for its controllability.
Type	RW

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">MEM_EQLEV</td><td colspan="5">MEM_EQFTC</td><td colspan="4">MEM_EQCTL</td><td colspan="4">RESERVED</td><td>MEM_OVRD_EQLEV</td><td>MEM_OVRD_EQFTC</td><td>RESERVED</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:16	MEM_EQLEV	Equalizer level control.	RW	0x0000
15:11	MEM_EQFTC	Equalizer zero freq control.	RW	0x00
10:7	MEM_EQCTL	0000 - Equalizer disabled 0001 - Fully adaptive; FTC normal 0010 - Fully adaptive; FTC inverted 0011 - Hold eq state 01xx - Init eq to fully adaptive start/midpoint 1000 - Partially adaptive; zero=1084 MHz 1001 - Partially adaptive; zero= 805 MHz 1010 - Partially adaptive; zero= 573 MHz 1011 - Partially adaptive; zero= 402 MHZ 1100 - Partially adaptive; zero= 304 MHz 1101 - Partially adaptive; zero= 216 MHz 1110 - Partially adaptive; zero= 156 MHz 1111 - Partially adaptive; zero= 135 MHz	RW	0x0
6:3	RESERVED		R	0
2	MEM_OVRD_EQLEV	Continuosly forces the Equalizer output with the eqlev[15:0].	RW	0
1	MEM_OVRD_EQFTC	Continuosly forces the Equalizer output with the eqftc[4:0].	RW	0
0	RESERVED		R	0

Table 26-35. SATAPHYRX_IO_AND_A2D_OVERRIDES_REG1

Address Offset	0x0000 0044
Physical Address	**Instance**
Description	This register has controls for SATA PHY RX tunning
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MEM_CDR_LOS_SOURCE		RESERVED								

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x0
10:9	MEM_CDR_LOS_SOURCE	0x0: the Analog's los_sts (analog's loss of signal detector output) is used by the CDR algorithm to stop the CDR loop. 0x1: the input pin los_in is used by the CDR algorithm for the same purpose. 0x2: the SCP register MEM_los_to_cdr_val is used by the CDR algorithm for the same purpose. This feature may be used if there is an external (above the PHY layer) filtering of the Analog's loss of detection status signal which should be used to stop the CDR loop updates (through los_in) instead of using Analog's unfiltered LOS status directly. 0x3: Reserved	RW	0x1
8:0	RESERVED	Reserved	R	0x000

26.3.5 SATA_PHY_TX Registers

26.3.5.1 SATA_PHY_TX Register Summary

Table 26-36. SATA_PHY_TX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SATA_PHY_TX Physical Address
RESERVED	R	32	0x0000 0000	0x4A09 6400
RESERVED	R	32	0x0000 0004	0x4A09 6404
RESERVED	R	32	0x0000 0008	0x4A09 6408
SATAPHYTX_FUNC_CONFIG_REG	RW	32	0x0000 000C	0x4A09 640C
RESERVED	R	32	0x0000 0010	0x4A09 6410
RESERVED	R	32	0x0000 0014	0x4A09 6414
RESERVED	R	32	0x0000 0018	0x4A09 6418
RESERVED	R	32	0x0000 001C	0x4A09 641C
RESERVED	R	32	0x0000 0020	0x4A09 6420
RESERVED	R	32	0x0000 0024	0x4A09 6424
RESERVED	R	32	0x0000 0028	0x4A09 6428
SATAPHYTX_TEST_CONFIG_REG	RW	32	0x0000 002C	0x4A09 642C
SATAPHYTX_PATTGEN_PRELOAD	RW	32	0x0000 0030	0x4A09 6430
RESERVED	R	32	0x0000 0034	0x4A09 6434

26.3.5.2 SATA_PHY_TX Register Description

Table 26-37. SATAPHYTX_FUNC_CONFIG_REG

Address Offset	0x0000 000C	
Physical Address		**Instance**
Description	Functional Configuration registers	
Type	RW	

31	30–0
MEM_INVPAIR	RESERVED

Bits	Field Name	Description	Type	Reset
31	MEM_INVPAIR	Invert polarity of TXP/TXN	RW	0
30:0	RESERVED		R	0x0000 0000

Table 26-38. SATAPHYTX_TEST_CONFIG_REG

Address Offset	0x0000 002C	
Physical Address		**Instance**
Description	Test related configuration registers	
Type	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MEM_EN_LPBK	MEM_ENTXPATT	MEM_TESTPATT	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		RW	0
30	MEM_EN_LPBK	Loopback enable for test	RW	0
29	MEM_ENTXPATT	Enable Test pattern to input of the serializer instead of TD	RW	0
28:26	MEM_TESTPATT	Select the LFSR mode to generate the required pattern 000=> 31 bit LFSR mode 011 => 23 bit LFSR mode 010 => 7 bit LFSR mode 001=> generate 1010 pattern 100=> Fixed 31 bit value from PATTGEN_PRELOAD_VAL	RW	0x0
25:0	RESERVED		RW	0x00 0000

Table 26-39. SATAPHYTX_PATTGEN_PRELOAD

Address Offset	0x0000 0030
Physical Address	**Instance**
Description	Pattern generator (31 bit) LFSR Seed or preload value
Type	RW

31:1	0
MEM_PATTGEN_PRELOAD_VAL	RESERVED

Bits	Field Name	Description	Type	Reset
31:1	MEM_PATTGEN_PRELOAD_VAL	Preload value to the LFSR pattern generator	RW	0x0000 0000
0	RESERVED		RW	0

26.3.6 DPLLCTRL Registers

26.3.6.1 DPLLCTRL Register Summary

Table 26-40. DPLLCTRL Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DPLLCTRL_USB_OTG_SS Physical Address	DPLLCTRL_SATA Physical Address
RESERVED	R	32	0x0000 0000	0x4A08 4C00	0x4A09 6800
PLL_STATUS	R	32	0x0000 0004	0x4A08 4C04	0x4A09 6804
PLL_GO	RW	32	0x0000 0008	0x4A08 4C08	0x4A09 6808
PLL_CONFIGURATION1	RW	32	0x0000 000C	0x4A08 4C0C	0x4A09 680C
PLL_CONFIGURATION2	RW	32	0x0000 0010	0x4A08 4C10	0x4A09 6810
PLL_CONFIGURATION3	RW	32	0x0000 0014	0x4A08 4C14	0x4A09 6814
PLL_SSC_CONFIGURATION1	RW	32	0x0000 0018	0x4A08 4C18	0x4A09 6818
PLL_SSC_CONFIGURATION2	RW	32	0x0000 001C	0x4A08 4C1C	0x4A09 681C
PLL_CONFIGURATION4	RW	32	0x0000 0020	0x4A08 4C20	0x4A09 6820

26.3.6.2 DPLLCTRL Register Description

Table 26-41. PLL_STATUS

Address Offset	0x0000 0004		
Physical Address	0x4A08 4C04 0x4A09 6804	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description	This register contains the status information		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">RESERVED</td><td>PLL_TICOPWDN</td><td>PLL_LDOPWDN</td><td colspan="2">RESERVED</td><td>SSC_EN_ACK</td><td colspan="6">RESERVED</td><td>PLL_HIGHJITTER</td><td>RESERVED</td><td>PLL_LOSSREF</td><td>PLL_RECAL</td><td>PLL_LOCK</td><td>PLLCTRL_RESET_DONE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16	PLL_TICOPWDN	PLL TICOPWDN status. Read 0x1: Internal oscillator power down Read 0x0: Internal oscillator power up	R	0
15	PLL_LDOPWDN	PLL LDOPWDN status. Read 0x1: PLL's internal LDO is power down Read 0x0: PLL's internal LDO is power up	R	0
14:13	RESERVED			0
12	SSC_EN_ACK	Spread Spectrum Clocking acknowledge Read 0x1: Spread Spectrum Clocking active Read 0x0: Spread Spectrum Clocking inactive	R	0
11:6	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
5	PLL_HIGHJITTER	PLL High Jitter status Read 0x1: PLL in high jitter condition: Phase error > 24% Read 0x0: PLL in normal jitter condition	R	0
4	RESERVED	Read returns zero.	R	0
3	PLL_LOSSREF	PLL Reference Loss status Read 0x1: Reference input inactive Read 0x0: Reference input active	R	0
2	PLL_RECAL	PLL re-calibration status If this bit is active, the PLL needs to be re-calibrated Read 0x1: Recalibration is required Read 0x0: Recalibration is not required	R	0
1	PLL_LOCK	PLL Lock status See the programming guide for the use of this bit Read 0x1: PLL is locked Read 0x0: PLL is not locked	R	0
0	PLLCTRL_RESET_DONE	PLLCTRL reset done status Read 0x1: Reset has completed Read 0x0: Reset is in progress	R	0

Table 26-42. PLL_GO

Address Offset	0x0000 0008		
Physical Address	0x4A08 4C08 0x4A09 6808	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description	This register contains the GO bit		
Type	RW		

31:1	0
RESERVED	PLL_GO

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return zero.	RW	0x0000 0000
0	PLL_GO	Request (re-)locking sequence of the PLL. 0x0: No pending action 0x1: Request PLL (re-)locking/locking pending	RW	0

Table 26-43. PLL_CONFIGURATION1

Address Offset	0x0000 000C		
Physical Address	0x4A08 4C0C 0x4A09 680C	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description	This register contains the latched PLL and HSDIVDER configuration bits		
Type	RW		

31:21	20:9	8:1	0
RESERVED	PLL_REGM	PLL_REGN	RESERVED

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		RW	0x000
20:9	PLL_REGM	M Divider for PLL	RW	0x000
8:1	PLL_REGN	N Divider for PLL (Reference)	RW	0x00
0	RESERVED	Read returns zero.	R	0

Table 26-44. PLL_CONFIGURATION2

Address Offset	0x0000 0010		
Physical Address	0x4A08 4C10 0x4A09 6810	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description	This register contains the unlatched PLL and HSDIVDER configuration bits These bits are "shadowed" when automatic mode is selected		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					PLL_LOCKSEL		RESERVED					PLL_SELFREQDCO			PLL_IDLE

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		RW	0x00004
10:9	PLL_LOCKSEL	Selects the lock criteria for the PLL 0x0: Phase Lock 0x1: Frequency Lock Other values: Reserved	RW	0x0
8:4	RESERVED		RW	0x00
3:1	PLL_SELFREQDCO	DCO frequency range selector for DPLL_USB_OTG_SS / DPLLCTRL_SATA 0x2 Set if DCO frequency is between 750MHz and 1500MHz 0x4 Set if DCO frequency is between 1250MHz and 2500MHz Other values: Reserved	RW	0x4
0	PLL_IDLE	PLL IDLE: 0x0: IDLE is not selected 0x1: IDLE is selected	RW	0

Table 26-45. PLL_CONFIGURATION3

Address Offset	0x0000 0014		
Physical Address	0x4A08 4C14 0x4A09 6814	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description	HSDIVIDER configuration bits for the M5 and M6 dividers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														PLL_SD								RESERVED									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:10	PLL_SD	Sigma delta divider setting for DPLL_USB_OTG_SS based on the PLL lock configuration.	RW	0x00
9:0	RESERVED		RW	0x000

Table 26-46. PLL_SSC_CONFIGURATION1

Address Offset	0x0000 0018		
Physical Address	0x4A08 4C18 0x4A09 6818	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description	Configuration for PLL Spread Spectrum Clocking modulation		
Type	RW		

31:3	2	1	0
RESERVED	DOWNSPREAD	RESERVED	EN_SSC

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	DOWNSPREAD	Forces the clock spreading only in the down spectrum. 0x0: Clock spreading not forced. 0x1: Spectrum spreading only in down direction.	RW	0
1	RESERVED		RW	0
0	EN_SSC	Spread Spectrum Clocking enable 0x0: Spread Spectrum Clocking disabled 0x1: Spread Spectrum Clocking enabled	RW	0

Table 26-47. PLL_SSC_CONFIGURATION2

Address Offset	0x0000 001C		
Physical Address	0x4A08 4C1C 0x4A09 681C	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description			
Type	RW		

31	30	29:20	19:0
RESERVED	DELTAM2	MODFREQDIVIDER	DELTAM

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reads as zero	R	0
30	DELTAM2	MSB of DeltaM control bus.	RW	0
29:20	MODFREQDIVIDER	Modulation Frequency Divider control for SSC.	RW	0x000
19:0	DELTAM	DeltaM control for SSC.	RW	0x0 0000

Table 26-48. PLL_CONFIGURATION4

Address Offset	0x0000 0020		
Physical Address	0x4A08 4C20 0x4A09 6820	**Instance**	DPLLCTRL_USB_OTG_SS DPLLCTRL_SATA
Description	Allows setting the fractional M divider and M2 divider for PLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PLL_REGM_F

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads as 0x1	RW	0x0001
17:0	PLL_REGM_F	Fractional part of M divider.	RW	0x0 0000

26.4 PCIe PHY Subsystem

This chapter describes the features and functions of the device Peripheral Component Interconnect Express (PCIe) physical layer (PHY) subsystem.

26.4.1 PCIe PHY Subsystem Overview

The device has two PCI Express physical ports - PCIe 0 and PCIe 1. Each of the two device PCIe ports has an associated PMA (physical media attachment) component pair of a high-speed operating differential transmitter (serializer) and a differential receiver (de-serializer). While the device PCIe port 0 PHY is always mapped to device PCIe_SS1 controller, the device PCIe port 1 PHY can be software multiplexed to the PCIe_SS1 (dual-lane configuration) or to the PCIe_SS2 (single-lane configuration) . In addition, the PCIe shared PHY subsystem encompasses a PCIe PCS (physical coding sublayer), a PCIe power management logic, APLL, a DPLL reference clock generator and an APLL low-jitter clock buffer.

- PCIe PCS (a physical coding sublayer component) converts an 8-bit portion of parallel data over a PCIe lane to a 10-bit parallel data to adapt the process of serialization and deserialization in the TX/RX PHYs to various requirements. At the same time it transforms the transmission rate to maintain the PCIe Gen2 bandwidth. A PCS component is hardware mapped to each of the PCIe_SS1 (16-bit) and PCIe_SS2 controller (32-bit) PIPE ports.
- A programmable multiplexing logic which maps either PCIe_SS1 or PCIe_SS2 PCS logic to PCIe port 1 PHY.
- PHY serializer (TX) and deserializer (RX) components with associated power control logic, building the so called PMA (physical media attachment) part.
- DPLL_PCIe_REF is a DPLL clock source, controlled from the device PRCM, that provides (typically 20 MHz/100 MHz) clock to the PCIe PHY serializer/de-serializer components reference clock inputs.
- Contains an integrated APLL (APLLPCIe) which multiplies the DPLL_PCIe_REF clock to 2.5 GHz.
- The APLLPCIe low-jitter buffer and additional logic takes care to provide the PCIe APLL reference input clock.
- An L4_CFG interface adapter - OCP2SCP3 enables accessing the PCIe PHY serial configuration protocol compatible (SCP) registers via L4_CFG interconnect accesses.

Figure 26-15 shows an overview of the device PCIe subsystem with its integrated components.

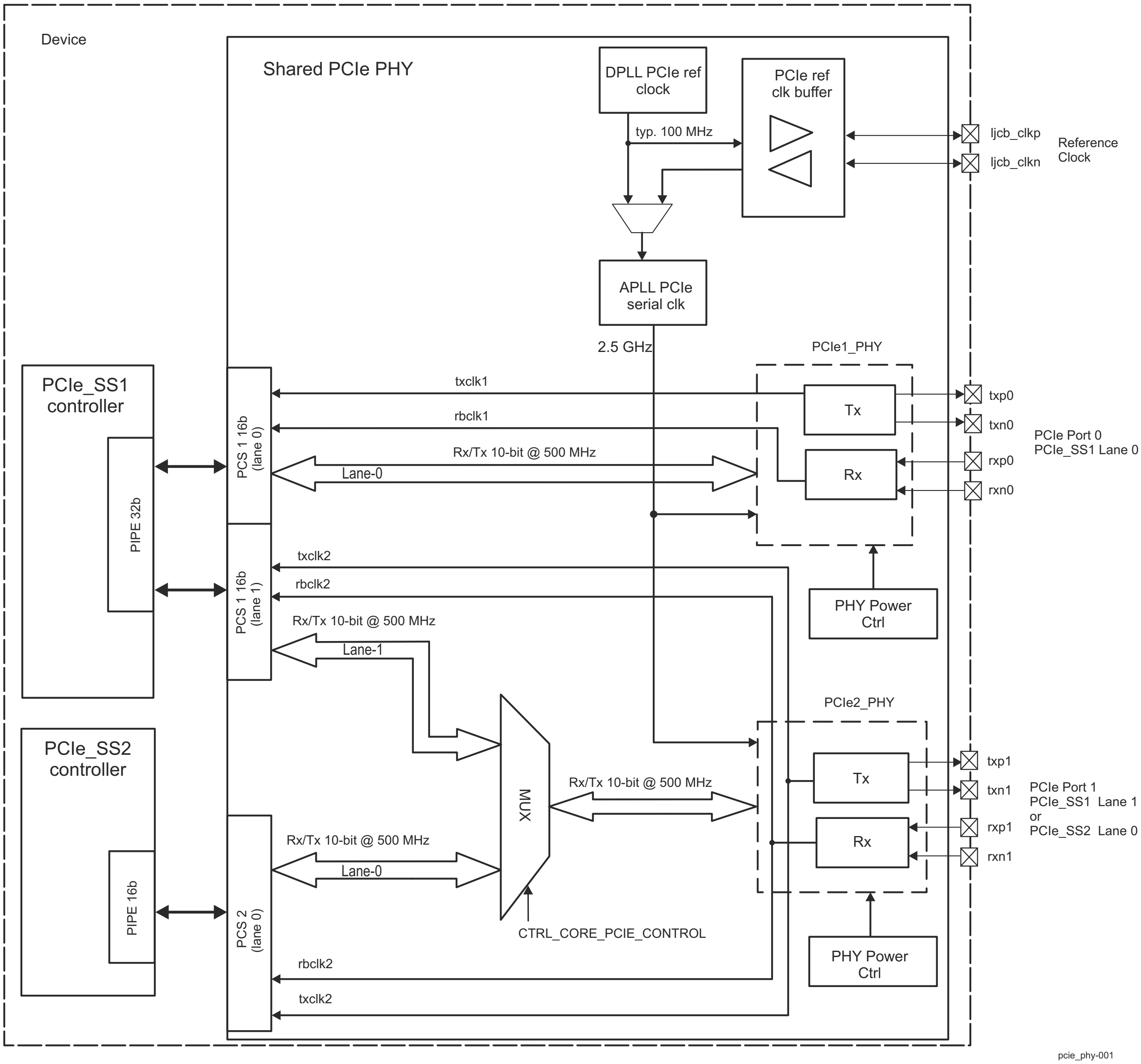


Figure 26-15. PCIe Controller Subsystem Overview

26.4.1.1 PCIe PHY Subsystem Key Features

This section describes the features supplied by the PCIe PHY subsystem. The device integrated PCIe PHY comply with the following standard:

- *PHY Interface for PCI Express Gen 2 and USB3.0 architectures (PIPE for USB3)*

The device supports PCI express interface in the following configurations:

- The PCIe PHYs can be mapped either as 2-lane to one controller (PCIe_SS1) or two separate lanes to two controllers (PCIe_SS1 and PCIe_SS2), as follows:
  - One 2x-lane functional (Gen2) PCIe port, built on the device two PCIe PHY ports (0 and 1), using the dual-lane PCIe_SS1 controller (the PCIe_SS1 controller)

  - Two 1x-lane (Gen2) PCIe ports - PCIe port 0 and PCIe port 1, mapped to the PCIe_SS1 and PCIe_SS2 controllers, each configured to operate in a single-lane mode, respectively. In this case the two device PCIe PHY ports function independently from each other.
- A power control module for each PCIe port which:
  - ensures the Rx/Tx PHYs power-up sequence
  - ensures the Rx/Tx PHYs power-down sequence
- Embedded PCIe DPLL generator, software controlled in device PRCM
- A PLL reference clock - (typically 20MHz or 100MHz), which can be programmed to be:
  - an externally supplied differential clock
  - internally supplied by DPLL_PCIE_REF generated clock, controlled from device PRCM
  - output to external systems when supplied by the DPLL_PCIE_REF
- A bidirectional low-jitter asynchronous buffer which:
  - is used to supply an external reference clock to the PCIe APLL.
  - can output DPLL_PCIE_REF clock to external chips
  - can be bypassed, when DPLL clock is used but NOT output to external chips.
- Polarity inversion on receiver

26.4.2 PCIe PHY Subsystem Environment

26.4.2.1 PCIe PHY I/O Signals

Table 26-49 represents module pins and their corresponding signal names at the device level, and also specifies their links to functions.

Table 26-49. PCIe PHY I/O Signals

Module Pin Name	Device-Level Signal Name	I/O(1)	Description	Pin Reset Value(2)
TXP0	pcie_txp0	O	TXP output of the PCIe Port 0 PHY differential transmission line	0
TXN0	pcie_txn0	O	TXN output of the PCIe Port 0 PHY differential transmission line	0
RXP0	pcie_rxp0	I	RXP input of the PCIe Port 0 PHY differential reception line	HiZ
RXN0	pcie_rxn0	I	RXN input of the PCIe Port 0 PHY differential reception line	HiZ
TXP1	pcie_txp1	O	TXP output of the PCIe Port 1 PHY differential transmission line	0
TXN1	pcie_txn1	O	TXN output of the PCIe Port 1 PHY differential transmission line	0
RXP1	pcie_rxp1	I	RXP input of the PCIe Port 1 PHY differential reception line	HiZ
RXN1	pcie_rxn1	I	RXN input of the PCIe Port 1 PHY differential reception line	HiZ
LJCB_CLKP	ljcb_clkp	I/O	Differential input/output of reference clock buffer (positive)	HiZ
LJCB_CLKN	ljcb_clkn	I/O	Differential input/output of reference clock buffer (negative)	HiZ

(1) I = Input, O = Output
(2) HiZ = High-impedance

To swap polarity of the PHY_TX serializer outputs - TXP and TXN such that, TXP becomes the negative and TXN becomes the positive terminal, user software has to assert bit PCIEPHYTX_FUNC_CONFIG_REG[31] MEM_INVPAIR to 0b1.

To swap polarity of the PHY_RX de-serializer inputs - RXP and RXN, such that, RXP becomes the negative and RXN becomes the positive terminal, user software has to:

- First assert the bit PCIEPHYRX_DIGITAL_MODES_REG1[30] MEM_OVRD_INV_RXPN_PAIR to 0b1, in order to be able to override the internal HW polarity control of the de-serializer.
- Assert bit PCIEPHYRX_DIGITAL_MODES_REG1[31] MEM_INV_RXPN_PAIR to 0b1.

For more information on the necessary SCP register access configuration refer to Section 26.4.6, *PCIe PHY Subsystem Register Manual* and Section 26.4.5, *PCIePHY Subsystem Low-Level Programming Model*.

Figure 26-16 shows module pin signals mapping to PCIe PHY signals visible at the device pad level.

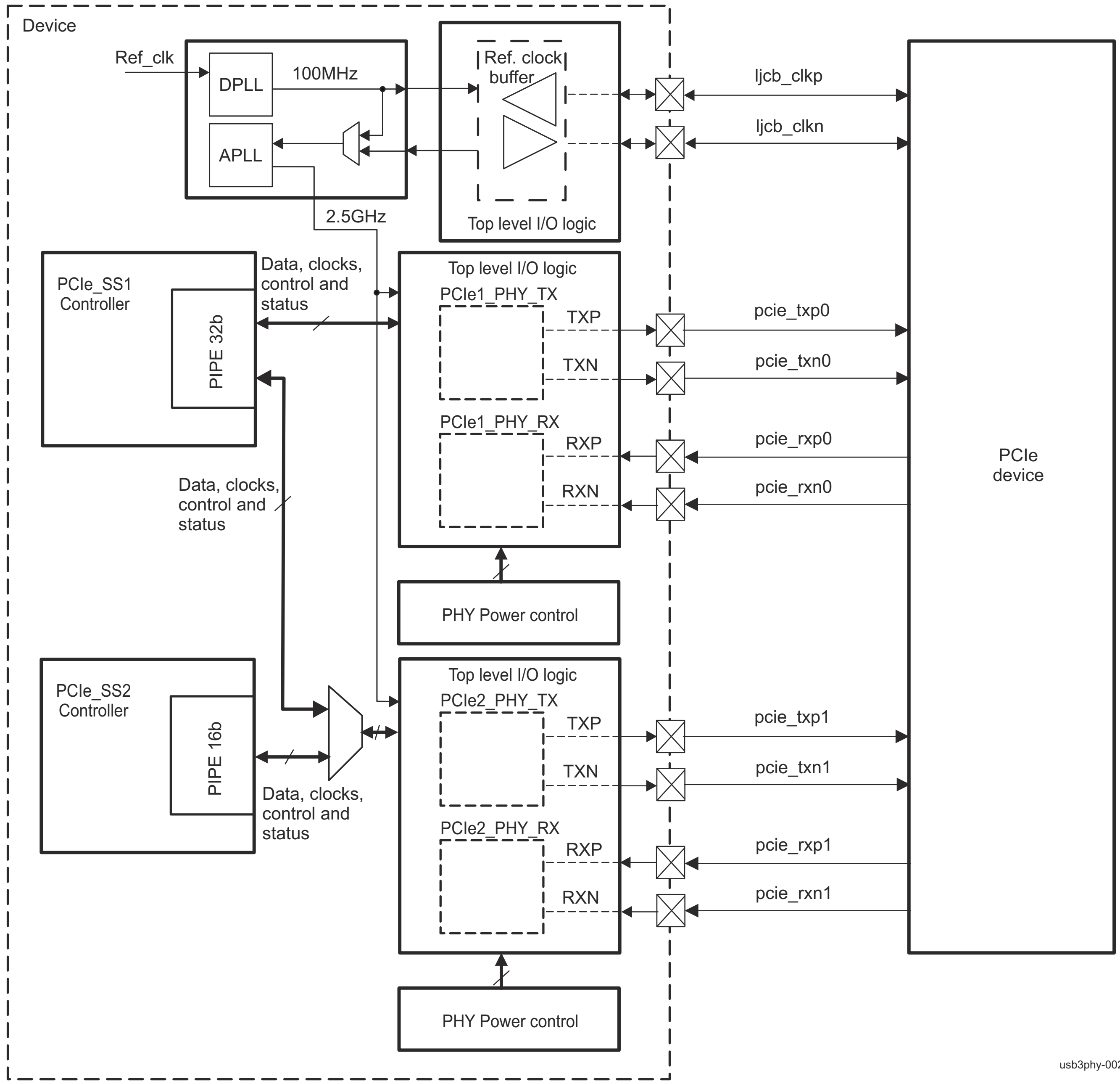


Figure 26-16. PCIe PHY I/O Signals

Note

The path from module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and can be programmed in the control module registers. For more information, see *Pad Configuration Registers* and *Control Module Register Manual* in *Control Module*.

26.4.3 PCIe Shared PHY Subsystem Integration

This section describes the PCIe PHY subsystem-related components (PCIe_PHY, DPLL_PCIE_REF, APLL_PCIE, and OCP2SPC3) integration in the device, including information about clocks, resets, and hardware requests.

Figure 26-17 shows the PCIe PHY integration.

The PCIe PHY module integration features:

- A low-power non retention reset, L3INIT_RST
- A low power non retention Power-on Reset, L3INIT_PWRON_RST
- Interconnect adapter target interface (OCP2SCP3)
- A high-frequency input clock for RX module PCIE_PHY_GCLK, tied to the PCIE_APLL.CLKVCOLDO output
- A high frequency divided input clock for TX module PCIE_PHY_DIV_GCLK, tied to PCIE_APLL.CLKVCOLDO_DIV output
- A local PCIE PHY PIPE compatible port connected to the PCIe_SS1 controller PIPE port
- A clock output (PIPE_PCLK) to the PCIe_SS module PIPE port
- A clock input (PIPE_MCLK) from the PCIe_SS module PIPE port
- A power control logic integrated for PCIe PHY reciever/transmiter pair
- PRCM.PCIE_32K_GFCLK for debounce and wakeup logic inside the PCIe1_PHY_RX
- PRCM.PCIE_REF_GFCLK clock (based on PRCM.CORE_USB_OTG_SS_LFPS_TX_CLK clock) tied to the PCIe1_PHY_TX and PCIe2_PHY_TX input
- PRCM.PCIE_SYS_GFCLK clock tied to power control module clock input
- A device CORE CONTROL MODULE command port to the power control module

The DPLL_PCIE_REF integration features:

- A low-power non retention reset, COREAON_PWRON_RST
- A gated version of the device SYS_CLK1 - PCIE_DPLL_CLK supplying DPLL_PCIE_REF.CLKINP pin.
- A single clock output, DPLL_PCIE_REF.CLKOUTLDO, to the APLL_PCIE input multiplexer
- No HS divider integration
- Direct PRCM register control

The APLL_PCIE integration features:

- A low-power non retention reset, COREAON_PWRON_RST
- Direct PRCM register control
- Selectable input clock from DPLL_PCIE_REF.CLKOUTLDO or external differential input clock conducted from device PCIe ljcb_clkp/n pads via the ACSPCIE buffer
- High frequency output clock on APLL_PCIE.CLKVCOLDO output tied to PCIE PHY RX modules
- High frequency divided output clock on APLL_PCIE.CLKVCOLDO_DIV output tied to PCIE PHY TX modules

The ACSPCIE integration features:

- A difirential clock input from the device pads or differential output to device pads supplied from DPLL_PCIE_REF.CLKOUTLDO.
- Single ended clock output to APLL_PCIE input multiplexer

The OCP2SCP3 (interconnect adapter) integration features:

- L4_CFG slave configuration interface
- One interface clock, L3INIT_L4_GICLK
- A nonretention reset, L3INIT_RST

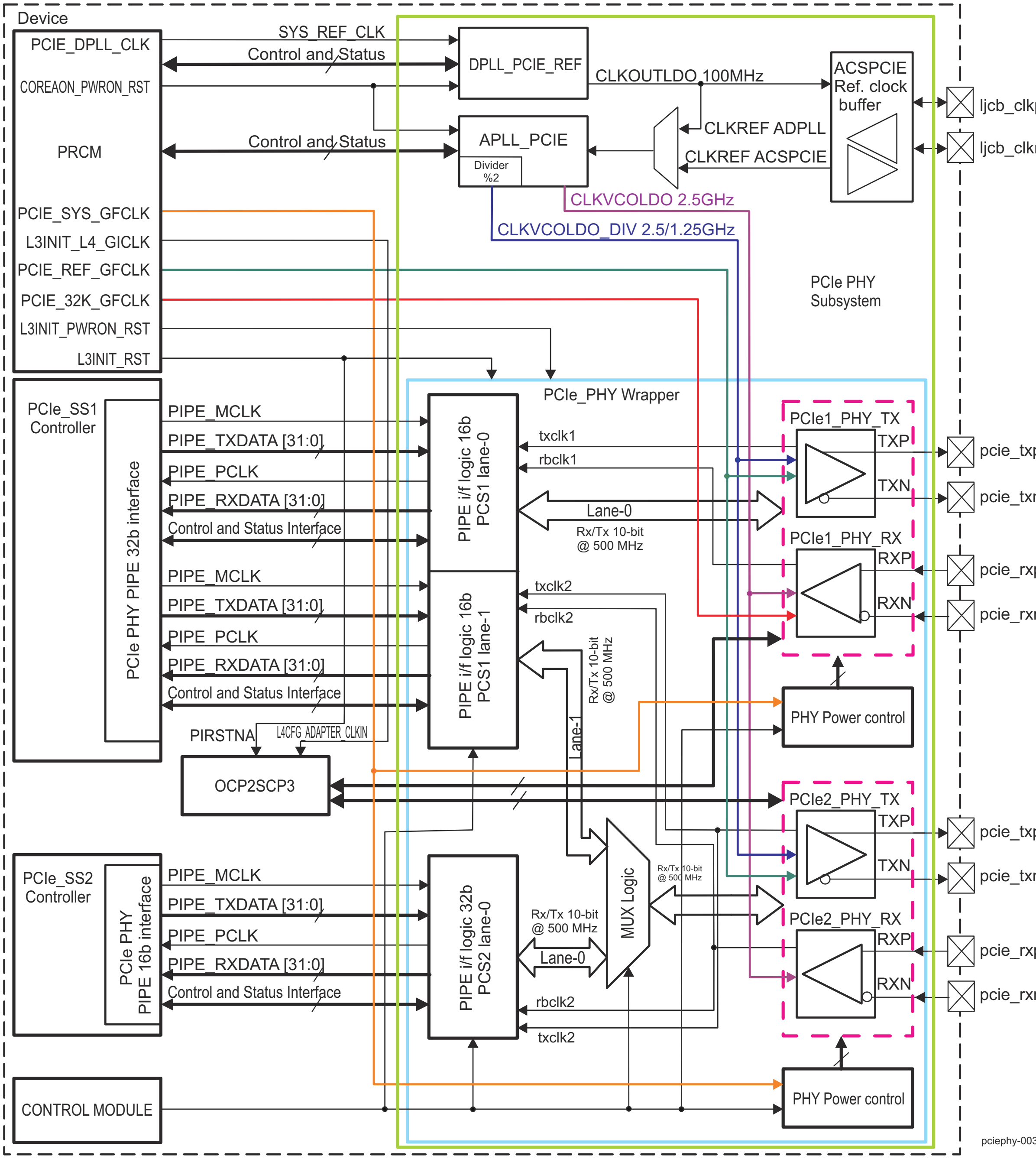


Figure 26-17. PCIe PHY Subsystem Integration

Table 26-50 through Table 26-52 summarize the integration of the module in the device.

Table 26-50. Integration Attributes

Module Instance	Attributes	
	Power Domain	**Interconnect**
PCIe1_PHY_TX	PD_L3INIT	OCP2SCP3 SCP interconnects
PCIe1_PHY_RX	PD_L3INIT	OCP2SCP3 SCP interconnects
PCIe2_PHY_TX	PD_L3INIT	OCP2SCP3 SCP interconnects
PCIe2_PHY_RX	PD_L3INIT	OCP2SCP3 SCP interconnects

Table 26-50. Integration Attributes (continued)

PCIE PHY (wrapper power controller)	PD_L3INIT	A device CTRL_CORE_MODULE power control bus
OCP2SCP3	PD_COREAON	L4_CFG
DPLL_PCIE_REF	PD_COREAON	Direct PRCM module register control
APLL_PCIE	PD_COREAON	Direct PRCM module register control

Table 26-51. Clocks

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DPLL_PCIE_REF	REF_CLK	PCIE_DPLL_CLK	PRCM	DPLL_PCIE_REF reference functional clock (SYS_CLK1 based)
PCIe1_PHY	PCIE1_PWR_CLK	PCIE_SYS_GFCLK	PRCM	PCIE power control logic reference functional clock (SYS_CLK1 based)
PCIe2_PHY	PCIE2_PWR_CLK	PCIE_SYS_GFCLK	PRCM	PCIE power control logic reference functional clock (SYS_CLK1 based)
PCIe1_PHY_TX	PCIE1_REF_CLKIN	PCIE_REF_GFCLK(1)	PRCM	Fixed frequency PCIE functional clock used for LFPS pattern generation (CORE_USB_OTG_SS_LFPS_TX_CLK based)
PCIe2_PHY_TX	PCIE2_REF_CLKIN	PCIE_REF_GFCLK(1)	PRCM	Fixed frequency PCIE functional clock used for LFPS pattern generation (CORE_USB_OTG_SS_LFPS_TX_CLK based)
PCIe1_PHY_RX	PCIE1_PHY_WKUP_CLK	PCIE_32K_GFCLK	PRCM	I/O wakeup and debounce 32-kHz functional clock at PCIe1_PHY_RX Receiver side
OCP2SCP3	L4CFG_ADAPTER_CLKIN	L3INIT_L4_GICLK	PRCM	L4_CFG adapter interface clock

(1) This clock is connected to a single clock input pin - REF_CLKIN at the PCIE PHY subsystem level

Table 26-52. Resets

Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
OCP2SCP3	PIRSTNA	L3INIT_RST	PRCM	A non retention reset to L4_CFG interface adapter
PCIe_PHY	PHY_RSTN_MAIN	L3INIT_RST	PRCM	A non retention reset to the PCIe_PHY
	PHY_RSTN_POR	L3INIT_PWRON_RST	PRCM	A non retention POR reset to the PCIe_PHY
DPLL_PCIE_REF	SYSRESETN	COREAON_PWRON_RST	PRCM	A non retention POR reset to DPLL reference clock generator
APLL_PCIE	APLLRESETN	COREAON_PWRON_RST	PRCM	A non retention POR reset to APLL clock generator

Note

The PCIe1_PHY_RX generates a hardware wakeup request to the PRCM module.

26.4.4 PCIe PHY Subsystem Functional Description

26.4.4.1 PCIe PHY Subsystem Block Diagram

Figure 26-18 shows a block diagram of the PCIe PHY subsystem.

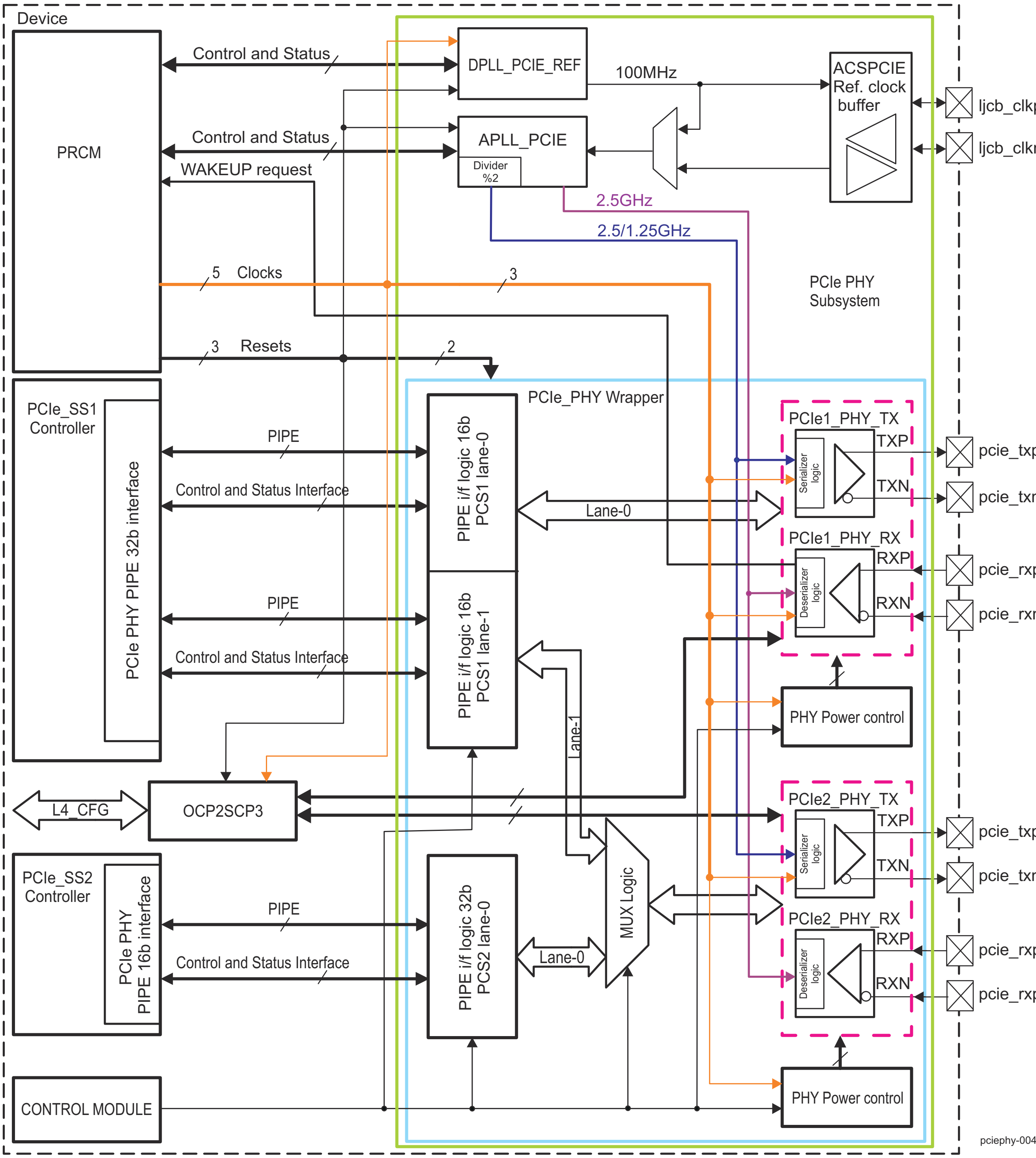


Figure 26-18. PCIe PHY Subsystem Block Diagram

26.4.4.2 OCP2SCP Functional Description

The OCP2SCP3 module (OCP2SCP instantiation 3), allows user software to configure the PCIe_PHY over the L4_CFG port. All PCIe_PHY_TX and PCIe_PHY_RX configuration registers are accessible through OCP2SCP3.

26.4.4.2.1 OCP2SCP Reset

26.4.4.2.1.1 Hardware Reset

The module receives an asynchronous hardware reset (L3INIT_RST) upon power-on reset (POR) at its active low PIRSTNA input. Table 26-52 lists the OCP2SCP3 system reset signal. For more information on the hardware reset source, see *Reset Domains* in *Power Reset and Clock Management*.

26.4.4.2.1.2 Software Reset

Setting the OCP2SCP_SYSCONFIG[1] SOFTRESET bit to 1 triggers a software reset on the OCP2SCP interconnect adapter. The OCP2SCP_SYSSTATUS[0] RESETDONE bit is used by software to monitor the status of reset completion. Hardware keeps this bit at 0 while the reset is being executed. A RESETDONE bit transition from 0 to 1 indicates OCP2SCP reset completion.

26.4.4.2.2 OCP2SCP Power Management

The OCP2SCP features idle acknowledgement protocol with the PRCM module.

26.4.4.2.2.1 Idle Mode

The smart-idle mode supported by OCP2SCP is not wake-up capable, which means software must explicitly take care to wake the OCP2SCP by setting the OCP2SCP_SYSCONFIG[4:3] IDLEMODE bit field to 0x1 (no idle), once it has previously gone to an idle mode.

For more information, see *Module-Level Clock Management* in *Power, Reset, and Clock Management*.

Note

Software must take the OCP2SCP module out of IDLE state by setting the OCP2SCP_SYSCONFIG[4:3] IDLEMODE bit field to 0x1, once smart-idle mode has been selected (OCP2SCP_SYSCONFIG[4:3] IDLEMODE = 0x2).

26.4.4.2.2.2 Clock Gating

The OCP2SCP module has local support for an automatic clock gating based on L4_CFG interconnect activity. This feature is enabled by setting the OCP2SCP_SYSCONFIG[0] AUTOIDLE bit to 0x1, otherwise the module interface clock is free running.

For more information, see *Clock Domain-Level Clock Management* in *Power, Reset, and Clock Management*.

26.4.4.2.3 OCP2SCP Timing Registers

The timing configuration register sets various parameters controlling the timing constraints of the OCP2SCP module.

The division ratio between the L4_CFG interconnect clock (L3INIT_L4_GICLK) and the serial configuration port output clock is set through the OCP2SCP_TIMING[9:7] DIVISIONRATIO bit field, with a valid range of 0x1 to 0x7.

The OCP2SCP_TIMING[6:4] SYNC1 timing information is programmable in the range 0 to 7 clock cycles, and shows the acceptable delay between the enable and command availability on SCP. The value of OCP2SCP_TIMING[3:0] SYNC2 is also programmable in the range of 1 to 15 clock cycles, measured from the moment the command is available on SCP until data is accessible.

Note

When the value 000 is programmed for the SCP clock division ratio, and the transaction to be made is a valid transaction on the SCP interface, the value of the DIVISIONRATIO bit field is set internally to 0x7 (to avoid a block on the L4_CFG interconnect interface).

Note

When the value 0000 is programmed for the SYNC2 bit field, and the transaction to be made is a valid transaction on the SCP interface, the value of SYNC2 is set to the minimum allowed 0x0001 (to avoid a block on the L4_CFG interconnect interface).

CAUTION

To ensure correct operation, DIVISIONRATIO must not be modified and the value of SYNC2 must be set to 0x6 or more. See OCP2SCP_TIMING register for details.

26.4.4.3 PCIe PHY Serializer and Deserializer Functional Descriptions

26.4.4.3.1 PCIe PHY Module Resets

26.4.4.3.1.1 Hardware Reset

The PCIe PHY Serializer and Deserializer modules recives two active low non-retention resets from PRCM modules.

- The Power-on reset L3INIT_PWRON_RST
- The main reset L3INIT_RST

For more information on the hardware reset source, see *Reset Domains*, in *Power, Reset, and Clock Management*.

26.4.4.3.1.2 Software Reset

No software reset is available for PCIe PHY Serializer and Deserializer modules.

26.4.4.3.2 PCIe PHY Subsystem Clocking

There are two clock domains within the PCIe PHY:

- PCIe_PHY PIPE port clock domain (PIPE_PCLK and PIPE_MCLK), where 32/16 bit-data is transferred between PCIe_SS controller and PCIe_PHY
- txclk/rbclk functional clock domain in which PCIe PHY generates clock: txclk for 10-bit parallel data transmission, and recovers the rbclk from serially received data.

26.4.4.3.2.1 PCIe PHY Subsystem Input Clocks

The PCIe PHY component receives a feedback clock, PIPE_MCLK, which is the loopback version of the PIPE_PCLK (PIPE port synchronizing clock) generated by the PCIe PHY component. The clock is turned on/off according to the PIPE power-down port states. As PIPE port works in source-synchronous mode, all data movement from the MAC layer to the PIPE interface is synchronous to PIPE_PCLK.

The high speed transmission clock input of the PCIe_PHY serializer and deserializer are connected to APLL_PCIE output pins. The PHY_RX module input clock pin is tied to APLL_PCIE_REF.CLKVCOLDO (PCIE_PHY_GCLK) clock. The PHY_TX input clock pin is tied to divided APLL_PCIE_REF.CLKVCOLDO_DIV (PCIE_PHY_DIV_GCLK) output clock. For more information on the APLL_PCIE_REF.CLKVCOLDO and APLL_PCIE_REF.CLKVCOLDO_DIV output clocks settings, see Section 26.4.4.4.1.4.2, *PCIe PHY APLL Output Clock Configuration*.

In two lane mode when both PCIe1_PHY and PCIe2_PHY modules are connected to PCIe_SS1 as Lane-0 and Lane-1, there is a requirement for the clocking of the transmitter parts PCIe1_PHY_TX and PCIe2_PHY_TX. Both input clocks of the transmitters of all ports has to be in phase, including the by-2-dividers used when operating in 2.5GT/s. This is implemented by providing the same pre-divided clock to the PCIe1_PHY_TX and PCIe2_PHY_TX, coming from the APLL_PCIE's own by-2-divider, and bypassing the by-2-dividers inside each PCIe1_PHY_TX and PCIe2_PHY_TX. In this way the PCIe1_PHY_TX and PCIe2_PHY_TX receive one high speed clock and are allways in phase. The receivers PCIe1_PHY_RX and PCIe2_PHY_RX do not need synchronization in two lane mode.

The high speed clock APLL_PCIE_REF.CLKVCOLDO_DIV (PCIE_PHY_DIV_GCLK) could be divided by 2 on two places, in the APLL_PCIE own by-2-divider or in the PHY_TX internal by-2-divider. These dividers support force-bypass mode where the divider is bypassed and no division is performed. The dividers of the APLL_PCIE and of the PHY_TX are cascaded: one divider should always be bypassed, and the other active. For one lane operation the APLL_PCIE divider could be force-baypassed and only internal PCIe_PHY divider to be active.

The APLL_PCIE by-2-divider bypass is controlled in PRCM register CM_CLKMODE_APLL_PCIE[8] CLKDIV_BYPASS bit.

The PCIe1_PHY_TX by-2-divider bypass mode is controlled by PCIe1_PHY_TX.PCIEPHYTX_DRIVER_DATA_CONFIG1 register bits as follows:

- Bypassed not forced: FB=0 (default mode):

  PCIEPHYTX_DRIVER_DATA_CONFIG1[1] MEM_OVRD_HS_RATE_ANA_OVERRIDE = 0b0
- Bypassed forced: FB=1:

  PCIEPHYTX_DRIVER_DATA_CONFIG1[1] MEM_OVRD_HS_RATE_ANA_OVERRIDE = 0b1

  PCIEPHYTX_DRIVER_DATA_CONFIG1[3:2] MEM_HS_RATE_ANA_OVERRIDE = 0b00

The receivers PCIe1_PHY_RX and PCIe2_PHY_RX are allways supplied with non divided clock APLL_PCIE_REF.CLKVCOLDO (PCIE_PHY_GCLK) and the internal dividers are always active and controlled by the PCIe controllers.

The APLL_PCIE input clock is MUX selectable between DPLL_PCIE_REF.CLKOUTLDO output and ACSPCIE output derived from device differential clock input pins.

The DPLL_PCIE_REF input clock pin CLKINP is tied to PRCM.PCIE_DPLL_CLK (SYS_CLK1 based) clock.

As shown in Table 26-51, the PCIe PHY Power control module clock input is supplied from SYS_CLK1 based clock PRCM.PCIE_SYS_GFCLK.

Software must notify the PHY logic about which clock frequency is selected by writing CTRL_CORE_PHY_POWER_PCIESS1[31:22] PCIESS1_PWRCTL_CLKFREQ bit field and CTRL_CORE_PHY_POWER_PCIESS2[31:22] PCIESS2_PWRCTL_CLKFREQ bit field.

The REF_CLKIN clock input is used to support different PCIe_PHY functions. REF_CLKIN clock input is tied to the PRCM.PCIE_REF_GFCLK (CORE_USB_OTG_SS_LFPS_TX_CLK based) functional clock.

The PCIE1_PHY_RX deserializer I/O wake-up logic is supported by the PRCM.PCIE_32K_GFCLK clock, applied at the logic clock input.

The input clock for PCIe_PHY SCP port configuration interface is delivered from the OCP2SCP3 interface adaclockpter.

The SCP port clock input of PCIe_PHY is driven by the OCP2SCP3 gateable output . The ratio of the OCP2SCP3 gateable output clock to the OCP2SCP3 input clock (L3INIT_L4_GICLK) is controlled through the OCP2SCP_TIMING[9:7] DIVISIONRATIO bit field. For more information, see Section 26.4.4.2.3, *OCP2SCP3 Timing Registers*.

26.4.4.3.2.2 PCIe PHY Subsystem Output Clocks

- PIPE_PCLK: The PIPE interface is source-synchronous. A PIPE_PCLK clock is generated by the PCIe_PHY components, used to synchronize PCIe_SS PIPE port inputs, and reflected back as the PIPE_MCLK along with the PIPE outputs. The clock is turned on/off according to the power control port. All data movement from the PIPE interface to the PCIe_SS MAC layer is synchronous to this clock.

- txclk1: Some clock division is performed over the high frequency PCIE_PHY_DIV_GCLK signal before its derivative txclk1 clock is output from the PCIe1_PHY_TX.
- txclk2: Some clock division is performed over the high frequency PCIE_PHY_DIV_GCLK signal before its derivative txclk2 clock is output from the PCIe2_PHY_TX.
- rbclk1: The PCIe1_PHY_RX output rbclk1 is the clock recovered by the PCIe1_PHY_RX on base of the serial data, received over the RXP0 and RXN0 lines from the attached to the PCIe PHY external device. The rbclk1 clock supplies the PCS1 link parallel 10-bit data reception logic.
- rbclk2: The PCIe2_PHY_RX output rbclk2 is the clock recovered by the PCIe2_PHY_RX on base of the serial data, received over the RXP1 and RXN1 lines from the attached to the PCIe PHY external device. The rbclk2 clock supplies the PCS1 or PCS2 link parallel 10-bit data reception logic.

The PIPE_PCLK clock which drives the PIPE logic is sourced by the on-chip PCIe_PHY. The PIPE interface is source-synchronous (that is, the clock is received from the PHY), used to synchronize PCIe_SS1 and PSIe_SS2 PIPE inputs, and reflected back along with the PIPE outputs as the PIPE_MCLK. The PIPE_PCLK clock is turned on/off according to the PCIe_PHY power control port. For more details, see Section 26.4.4.3.3, *PCIe PHY Power Management*.

26.4.4.3.3 PCIe PHY Power Management

From one side the control over power up/down states of the USB3_PHY is provided through a Power control modules tightly integrated with the the PCIe1_PHY and PCIe2_PHY physical layer TX/RX components. On the other side, the PCIe_PHY PIPE i/f logic of the PCIe_PHY wrapper conveys power transition commands/ states (L0–L3) from the PCIe_SS1 and PCIe_SS2 MAC PIPE power management-ports to the PCIe1_PHY and PCIe2_PHY components as their corresponding P0 (active) and P0s, P1, P2 (low-power) states.

26.4.4.3.3.1 PCIe PHY Power-Up/-Down Sequences

The PHY power control module receives a power-up/power-down command input from the device general core control module. The power control module takes care to execute different stages of the PHY_TX and PHY_RX power-up/-down processes and ensures that the necessary timing intervals are applied between these stages.

Powering-up/-down the PHY_TX and PHY_RX modules is triggered through software writing corresponding power-up/down commands in the CTRL_CORE_PHY_POWER_PCIESS1 and CTRL_CORE_PHY_POWER_PCIESS2 registers of the device core control module.

For more information, see *Control Module*.

26.4.4.3.3.2 PCIe PHY Low-Power Modes

An implemented powerdown control port allows PCIe_PHY to support four low-power states:

- **P0 (pipe_powerdown = 0b00): active state**, for high-speed (HS) data transmission and reception. Used for L0 link state.
- **P0s (pipe_powerdown = 0b01):** active HS receiver, suspended transmitter. Used for L0s link state.
- **P1 (pipe_powerdown = 0b10):** Suspended HS receiver and transmitter, but PIPE clock still running.
- **P2 (pipe_powerdown = 0b11):** Suspended HS receiver and transmitter, PIPE clock stopped (i.e. asynchronous mode). Th to transmit and receive the "beacon".

The PCIe PHY power transitions are managed from connected PCIe_SS controller power management state machines according to desired Link and physical layer power states via signal PIPE_POWERDOWN[1:0]. For more information see Section 24.9.4.5 *PCIe Controller Power Management* in Section 24.9 *PCIe Controllers* chapter.

26.4.4.3.3.3 Clock Gating

The PCIe_PHY component high-speed clocks, PLL_CLK and PIPE_PCLK, are gated during the P2 low-power state.

26.4.4.3.4 PCIe PHY Hardware Requests

An asynchronous wake-up request is generated to the PRCM module by the PCIe1_PHY_RX I/Os (RXP0, RXN0).

Neither interrupt nor DMA requests are generated.

26.4.4.4 PCIe PHY Clock Generator Subsystem Functional Description

The clock generator subsystem of PCIe PHY module consists of integrated DPLL generator DPLL_PCIE_REF, an APLL high frequency generator for PHY transmision clocks APLL_PCIE and ACSPCIE input/output reference low-jitter clock buffer for connecting with external PCIe device. Figure 26-19 shows the block diagram of Clock Generator Subsistem of PCIe PHY module.

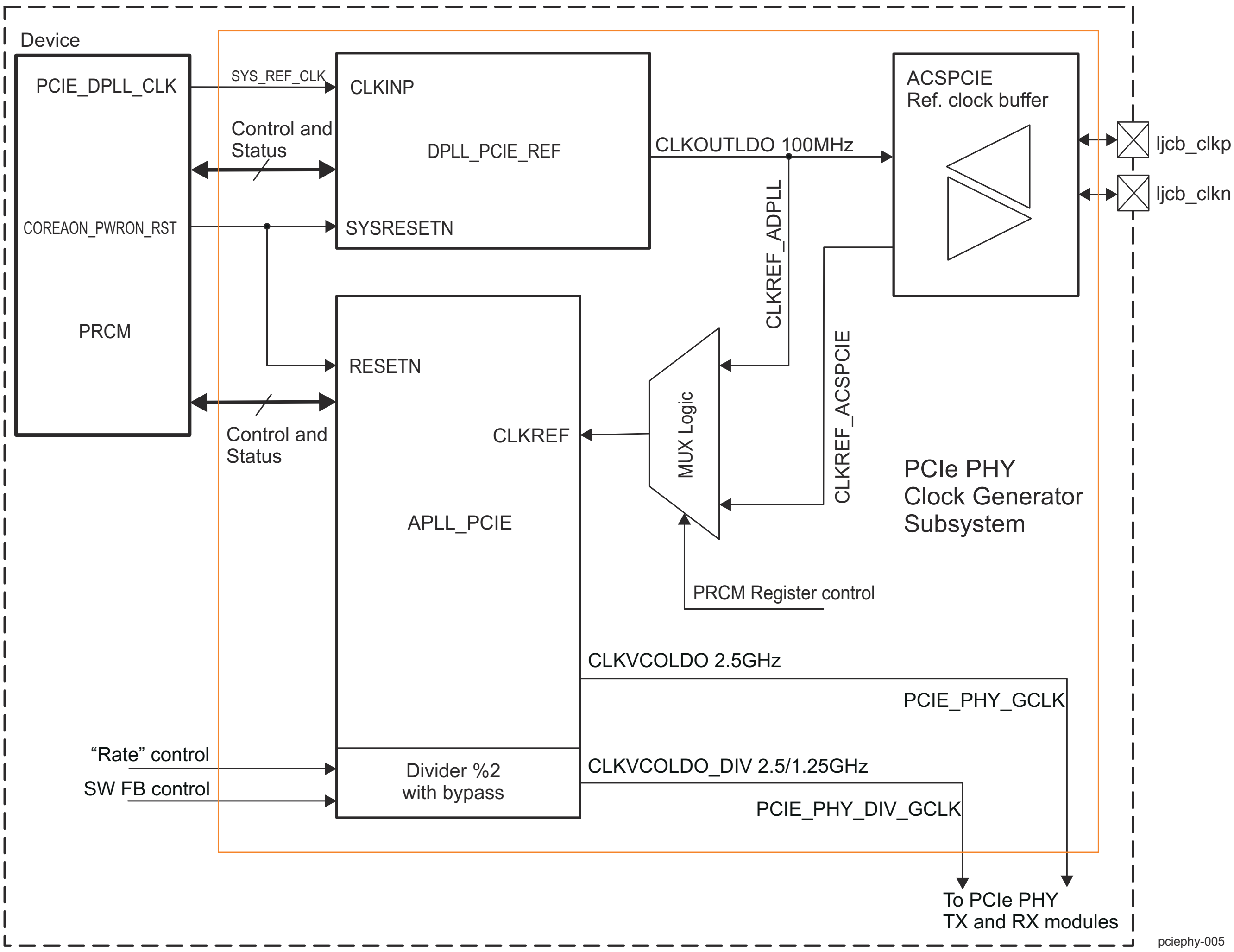


Figure 26-19. PCIe PHY Clock Generator Overview

The DPLL_PCIE_REF clock generator receives its input clock directly from PRCM as PCIE_DPLL_CLK (SYS_CLK1 based). The status and control registers for DPLL_PCIE_REF are located in the PRCM module and allow direct programming of the DPLL_PCIE_REF. The output clock of DPLL_PCIE_REF is on output pin CLKOUTLDO and is fed directly in the APLL_PCIE input clock multiplexer. In input mode, the reference low-jitter clock buffer ACSPCIE receives his clock from diferential input pins directly from outside device and after only buffering feeds this clock to the APLL_PCIE input clock multiplexer.

The APLL_PCIE input clock multiplexer chooses the input clock for APLL_PCIE according to register programming and feeds it to the input of APLL_PCIE. The selection of clock source is made by the register PRCM.CM_CLKMODE_APLL_PCIE[7] REFSEL bit as follows:

PRCM.CM_CLKMODE_APLL_PCIE[7] REFSEL = 0b0 - input clock CLKREF_ADPLL, APLL reference input clock is from DPLL_PCIE_REF

PRCM.CM_CLKMODE_APLL_PCIE[7] REFSEL = 0b1 - input clock CLKREF_ACSPCIE, APLL reference input clock is from ACSPCIE

The APLL_PCIE has two output high frequency transmission clocks. The main undivided clock PCIE_PHY_GCLK is delivered from output CLKVCOLDO and is fed directly to the PHY RX module. The second divided clock PCIE_PHY_DIV_GCLK is delivered after passing the main APLL output clock through a by-2-divider (with baypass capabilities) and is outputed on CLKVCOLDO_DIV output. This clock is fed to the PCIe PHY TX modules and delivers sinchronization of the TX modules in two lane mode.

26.4.4.4.1 PCIe PHY DPLL Clock Generator

This section describes the PCIe PHY DPLL reference clock generator DPLL_PCIE_REF.

26.4.4.4.1.1 PCIe PHY DPLL Clock Generator Overview

The DPLL module integrated in the PCIe PHY is a single instance high speed clock generator, used to deliver the reference clock to the main clock generator APLL_PCIE. The DPLL_PCIE_REF is directly controlled from the PRCM module and all the necessary control and status signals are exported by the subsystem.

The DPLL_PCIE_REF features:

- A programmable 8-bit input divider: N
- A programmable 12-bit integer multiplier: M
- A programmable 7-bit integer post divider M2
- Digital control and loop filter
- Internal oscillator divided output clock (CLKOUTLDO output)
- Idle-bypass low-power mode
- Low Power Stop mode

26.4.4.4.1.2 PCIe PHY DPLL Clock Generator Reset

The PCIe PHY DPLL clock generator receive hardware non-retention reset, COREAON_PWRON_RST, which comes from the device power and reset manager. For more information on the hardware reset source, see *Reset Domains* in *Power, Reset, and Clock Management*.

The DPLL_PCIE_REF itself has no software reset capabilities.

26.4.4.4.1.3 PCIe PHY DPLL Low-Power Modes

The DPLL_PCIE_REF module supports two types of low power mode controlled by the PRCM register PRCM.CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN bits.

The low-power modes supported by DPLL_PCIE_REF are Idle-bypass low-power and Low Power Stop modes, which are both characterized by:

- Internal LDO switched off
- DCO oscillator switched off
- CLKOUTLDO output pulled low

For more details on the DPLL settings and conditions necessary to enter Idle-bypass and Low Power Stop modes, see Section 26.4.4.4.1.6.4, *PCIe PHY DPLL Idle-bypass low-power Mode*, and Section 26.4.4.4.1.6.5, *PCIe PHY DPLL Low Power Stop Mode*.

DPLL_PCIE_PHY is held in a similar low-power state (DCO and LDO switched off, with CLKOUTLDO = 0) after Power-up Reset.

26.4.4.4.1.4 PCIe PHY DPLL Clocks Configuration

26.4.4.4.1.4.1 PCIe PHY DPLL Input Clock Control

The DPLL_PCIE_REF accepts the functional clock, PCIE_DPLL_CLK, on its CLKINP pin directly from the device PRCM, without involving any control interactions. The PCIE_DPLL_CLK is derived from SYS_CLK1. See *Clock Domain Module Attributes* in *Power, Reset, and Clock Management*.

26.4.4.4.1.4.2 PCIe PHY DPLL Output Clock Configuration

Only the DPLL_PCIE_REF output CLKOUTLDO is used to supply the reference clock to the APLL input multiplexer. To adjust the output clock frequency and jitter, the following values must be programmed, dividers SD, N and M2 and the multiplier M. The values for these parameters must be programmed in the corresponding registers in PRCM as follows:

PRCM.CM_CLKSEL_DPLL_PCIE_REF[31:24] DPLL_SD_DIV - Sigma-Delta divider SD, must be set by software to ensure optimum jitter performance.

PRCM.CM_CLKSEL_DPLL_PCIE_REF[19:8] DPLL_MULT - multiplier M, multiplier factor (2 to 4095).

PRCM.CM_CLKSEL_DPLL_PCIE_REF[7:0] DPLL_DIV - dvider N, input clock divider factor (0 to 255) (actual division factor is N+1).

PRCM.CM_DIV_M2_DPLL_PCIE_REF[6:0] DIVHS - divider M2, output clock post-divider factor (1 to 127).

The state of the output clock CLKOUTLDO is indicated by the PRCM.CM_DIV_M2_DPLL_PCIE_REF[10] CLKLDOST bit.

For more details on output clock settings sequence, see Section 26.4.4.4.1.6.6, *PCIe PHY DPLL Clock Programming Sequence*.

26.4.4.4.1.4.2.1 PCIe PHY DPLL Output Clock Gating

There is no direct software gate control for the DPLL_PCIE_REF.CLKOUTLDO output.

DPLL_PCIE_REF.CLKOUTLDO clock output is automatically gated (CLKOUTLDO pulled low) in the following scenarios:

- DPLL power-up sequence. For more information on power-up sequence, see Section 26.4.4.4.1.6.1, *PCIe PHY Clock Generator Power Up*.
- DPLL entering a relock sequence. For more information on relocking sequence, see Section 26.4.4.4.1.6.2, *PCIe PHY DPLL Sequences*.
- DPLL entering Idle-bypass low-power mode. For more information on idle-bypass mode, see Section 26.4.4.4.1.6.4, *PCIe PHY DPLL Idle-Bypass Mode*.
- DPLL entering Low Power Stop mode. For more information on Low Power Stop mode, see Section 26.4.4.4.1.6.5, *PCIe PHY DPLL Low Power Stop Mode*.

26.4.4.4.1.5 PCIe PHY DPLL Subsystem Architecture

Figure 26-20 is a simplified block diagram of the DPLL_PCIE_REF instance integrated in the PCIe PHY clock generator subsystem.

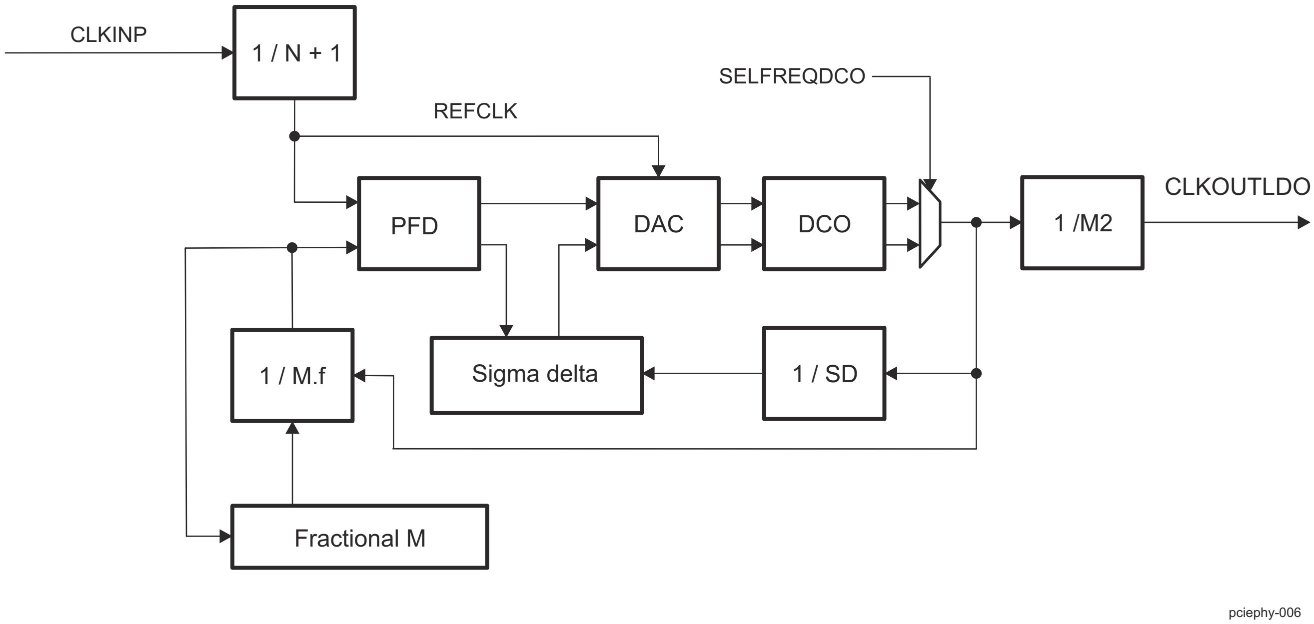


Figure 26-20. DPLL_PCIE_REF Functional Block Diagram

The input clock CLKINP goes to a predivider N + 1. The entire loop runs on the REFCLK clock after this predivider. The value of N + 1 is controlled through the PRCM.CM_CLKSEL_DPLL_PCIE_REF[7:0] DPLL_DIV bit field.

The frequency ranges for the DPLL_PCIE_REF input clock - CLKINP and the DPLL internal reference clock, REFCLK = CLKINP/N + 1 are:

- 0.62 to 60 MHz for CLKINP
- 0.62 to 2.5 MHz for the REFCLK

The output clock CLKOUTLDO is synthesized by digitally controlled oscillator (the DCO block), that automatically detects the frequency range divided by the M2 value. The CLKOUTLDO frequency can be given with CLKOUTLDO = [CLKINP×M/(N + 1)]/M2. For that purpose the feedback multiplier M must be configured through the PRCM.CM_CLKSEL_DPLL_PCIE_REF[19:8] DPLL_MULT bit field and the output clock divider M2 must be configured through the PRCM.CM_DIV_M2_DPLL_PCIE_REF[6:0] DIVHS bit field.

Note

Fractional synthesis is not supported for M.

26.4.4.4.1.6 PCIe PHY DPLL Clock Generator Modes and State Transitions

The DPLL_PCIE_REF can be set in different modes during operation. PRCM triggers DPLL_PCIE_REF state transitions to different static modes by setting the bit field of the PRCM.CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN. Automatic state transitions could be enabled if corresponding bits in PRCM.CM_AUTOIDLE_DPLL_PCIE_REF register are set. The DPLL_PCIE_REF then can automaticly enter the low power condition as follows:

If PRCM.CM_AUTOIDLE_DPLL_PCIE_REF[2:0] AUTO_DPLL_MODE is set to 0x0 - automatic control is disabled.

If PRCM.CM_AUTOIDLE_DPLL_PCIE_REF[2:0] AUTO_DPLL_MODE is set to 0x1- The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.

If PRCM.CM_AUTOIDLE_DPLL_PCIE_REF[2:0] AUTO_DPLL_MODE is set to 0x5 - The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.

For more information see *Power, Reset, and Clock Management*.

26.4.4.4.1.6.1 PCIe PHY Clock Generator Power Up

After power up, the DPLL_PCIE_REF.SYSRESETN input is automatically pulled low by the PRM, together with the APLL_PCIE.RESETN input. Because PRM.COREAON_PWRON_RST is an asynchronous reset, the DPLL_PCIE_REF input clock (DPLL_PCIE_REF.CLKINP) is not demanded upon reset. During DPLL power-up mode, CLKOUTLDO clock is maintained inactive (pulled low). After power-up reset, the DPLL_LOCK (internal lock loop) signal is maintained deasserted, too. The default value of the mode select register bit field PRCM.CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN puts the DPLL_PCIE_REF in Idle Bypass Low Power mode. It is software responability to change the state of the DPLL_PCIE_REF to desired mode after PRCM reset.

26.4.4.4.1.6.2 PCIe PHY DPLL Sequences

Once all the configuration values have been initially programmed into the PRCM registers controlling DPLL_PCIE_REF (see Section 26.4.4.4.1.4.2), the PRCM.CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN bit should be set to 0x7 to start the DPLL calibration and locking sequence. After performing the locking sequence a lock status is indicated for DPLL_PCIE_REF by PRCM.CM_IDLEST_DPLL_PCIE_REF[0] ST_DPLL_CLK bit asserted to 0b1.

26.4.4.4.1.6.3 PCIe PHY DPLL Locked Mode

When DPLL_PCIE_REF finishes calibration and lock sequences it enters a locked state. DPLL_PCIE_REF locked state is indicated by PRCM.CM_IDLEST_DPLL_PCIE_REF[0] ST_DPLL_CLK bit asserted to 0b1. In locked mode all the parameters of DPLL_PCIE_REF are set and the loop is running. The output clock CLKOUTLDO is active.

26.4.4.4.1.6.4 PCIe PHY DPLL Idle-Bypass Mode

Idle-bypass fast relock mode is not supported for DPLL_PCIE_REF.

DPLL_PCIE_REF supports idle-bypass low-power mode. A transition from a normal operation to idle-bypass mode is performed when software sets the PRCM.CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN bit field to 0x5. IDLE signal assertion triggers a power-down sequence on DPLL internal LDO analog blocks and the DCO oscillator. This mode is also the default state after PRCM reset for the DPLL_PCIE_REF.

In idle-bypass low-power mode, the CLKOUTLDO goes low. Also, the internal reference clock REFCLK = CLKINP/N + 1 is gated inside the DPLL digital control logic to save power, the internal LDO and the DCO are turned off.

To exit idle-bypass mode and restore clock generation, the user should write PRCM.CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN bit field to 0x7, which deasserts the IDLE signal, and DPLL_PCIE_REF automatically enters a relock sequence. CLKOUTLDO output clock is activated after DPLL_PCIE_REF enters locked mode.

26.4.4.4.1.6.5 PCIe PHY DPLL Low Power Stop Mode

The Low Power Stop mode will be activated if software writes PRCM.CM_CLKMODE_DPLL_PCIE_REF[2:0] DPLL_EN bit field to 0x1. The module enters a low Power Stop mode by gating all its internal clocks (REFCLK) and powering down internal LDO and DCO. CLKOUTLDO remains gated (low) during this mode.

Note

When the DPLL_PCIE_REF is in Low Power Stop mode, the input clock on CLKINP pin is not needed.

26.4.4.4.1.6.6 PCIe PHY DPLL Clock Programming Sequence

The DPLL_PCIE_REF factors must be calculated based on the required input and output frequencies, keeping the PLL internal reference frequency (REFCLK) in the appropriate range (0.62 to 2.5 MHz).

The values that must be considered during programming are:

- The internal reference frequence REFCLK. Must be kept in the range of 0.62 to 2.5 MHz. The value is calculated as REFCLK = CLKINP/(N+1).
- The Sigma-Delta divider to ensure optimum jitter performance. Must ensure that the sigma-delta operation frequency is as close as possible, but less than 250MHz for optimal performance. The value is calculated as SD = CEILING([(M)/(N+1)] × CLKINP/250) where CLKINP is the input clock of the DPLL in MHz.
- The output clock CLKOUTLDO frequency. This frequency must be programmed to 100 MHz fixed value for correct PCIe operation. The value is calculated as CLKOUTLDO = [CLKINP×M/(N + 1)]/M2
- The DCO frequency range must be set according to needed DCO output clock DCOCLK = CLKINP × [M/(N+1)]. The value of the SELFREQDCO[2:0] DPLL input selects the DCO internal oscilator ICO1 or ICO2 as follows:
  – If 750 < DCOCLK < 1500, SELFREQDCO[2:0] must be set to 0b010 (HS2 mode, ICO2 selected)
  – If 1250 < DCOCLK < 2500, SELFREQDCO[2:0] must be set to 0b100 (HS1 mode, ICO1 selected)
  – All other combinations of SELFREQDCO[2:0] are reserved

The following registers set the requred parameters values for the DPLL_PCIE_REF.

- PRCM.CM_CLKSEL_DPLL_PCIE_REF[7:0] DPLL_DIV - sets dvider N, input clock divider factor (0 to 255) (actual division factor is N+1)
- PRCM.CM_CLKSEL_DPLL_PCIE_REF[21] DPLL_SELFREQDCO bit sets DCO frequency range.
  – DPLL_SELFREQDCO should be set to 0b0 if 750 MHz < CLKDCOLDO [MHz] < 1500 MHz (SELFREQDCO[2:0] set to 0b010)
  – DPLL_SELFREQDCO should be set to 0b1 if 1250 MHz < CLKDCOLDO [MHz] < 2500 MHz (SELFREQDCO[2:0] set to 0b100).
- PRCM.CM_CLKSEL_DPLL_PCIE_REF[19:8] DPLL_MULT - sets multiplier M, multiplier factor (2 to 4095).
- PRCM.CM_DIV_M2_DPLL_PCIE_REF[6:0] DIVHS - sets divider M2, output clock post-divider factor (1 to 127).
- PRCM.CM_CLKSEL_DPLL_PCIE_REF[31:24] DPLL_SD_DIV - sets Sigma-Delta divider SD, must be set by software to ensure optimum jitter performance.

To maintain proper PCIe operation of the PCIe PHY submodule a set of parameters values are recomended. The DCO frequency of the PCIe reference DPLL is not used directly, so it can be changed to optimize jitter, power, or lock time. Only the divided-by-M2 CLKOUTLDO output clock is a required. The tables below give ratios that have been verified to meet PCIe jitter requirements. Other ratios combinations exist, to obtain a lower DCO frequency and lower power, but they should be checked against the PCIe standard.

Table 26-53. DPLL_PCIE_REF Recommended Configuration

CLKINP (MHz)	N	N+1	REFCLK (MHz)	M	DCOCLK (GHz)	DPLL_SELF REQDCO	M2	CLKOUTLDO (MHz)
20	9	10	2	750	1.5	0	15	100

Figure 26-21 shows the programming sequence for DPLL_PCIE_REF.

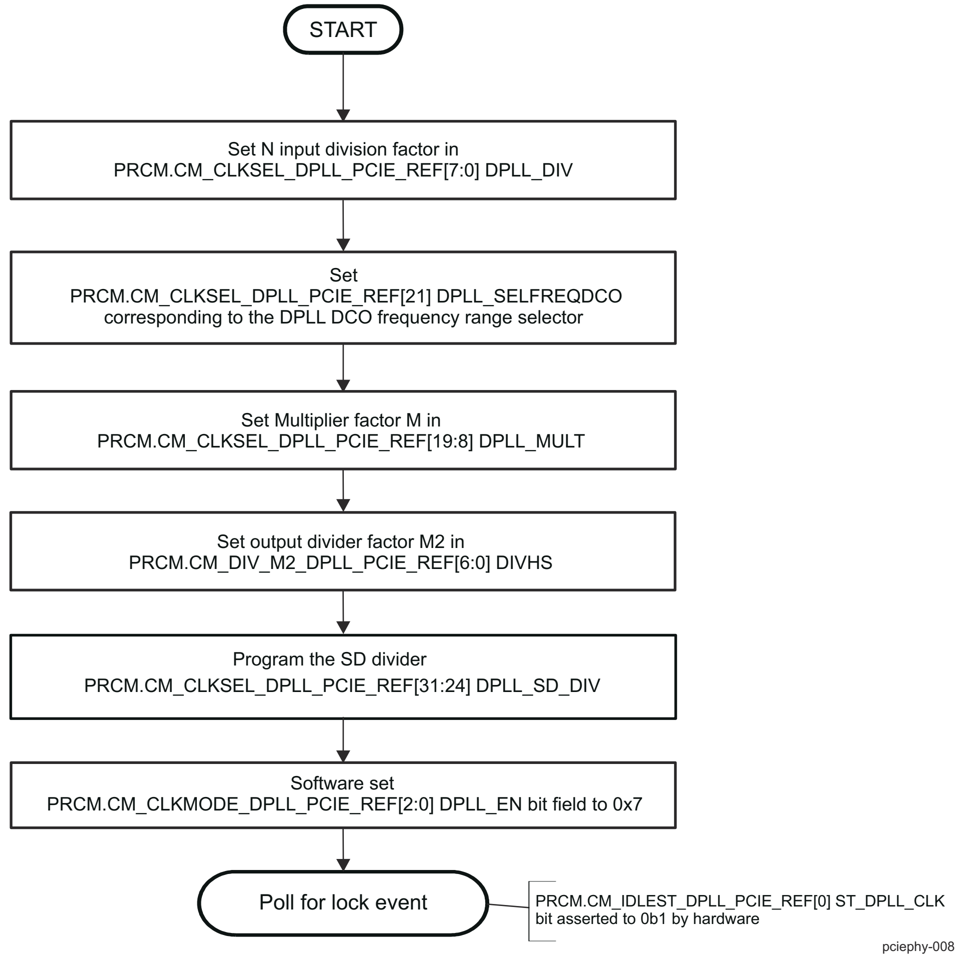


Figure 26-21. DPLL_PCIE_REF Programming Sequence

Note

- The sequence applies to the CLKOUTLDO output of the DPLL_PCIE_REF.
- CLKOUTLDO output frequency of the PLL_PCIE_REF should be programmed to 100 MHz, for the proper PCIe operation.

Table 26-54 summarizes the PRCM registers for the DPLL_PCIE_REF programming sequence.

Table 26-54. Register Call Summary for DPLL_PCIE_REF Programming Sequence

Register Name	Register Name	Register Name
CM_CLKSEL_DPLL_PCIE_REF	CM_DIV_M2_DPLL_PCIE_REF	CM_CLKMODE_DPLL_PCIE_REF

26.4.4.4.1.6.7 PCIe PHY DPLL Recommended Values

Table 26-55 lists the DPLL_PCIE_REF setup registers in PRCM recommended values.

Table 26-55. Recommended Programming Values

Field Name	Value	Description
CM_CLKSEL_DPLL_PCIE_REF[7:0] DPLL_DIV	See Table 26-53	Sets dvider N, input clock divider factor (0 to 255) (actual division factor is N+1)
CM_CLKSEL_DPLL_PCIE_REF[21] DPLL_SELFREQDCO	See Table 26-53	Sets DCO frequency range

Table 26-55. Recommended Programming Values (continued)

Field Name	Value	Description
CM_CLKSEL_DPLL_PCIE_REF[19:8] DPLL_MULT	See Table 26-53	Sets multiplier M, multiplier factor (2 to 4095)
CM_DIV_M2_DPLL_PCIE_REF[6:0] DIVHS	See Table 26-53	Sets divider M2, output clock post-divider factor (1 to 127)
CM_CLKSEL_DPLL_PCIE_REF[31:24] DPLL_SD_DIV	See [1]	Sets Sigma-Delta divider SD, SD = CEILING([(M)/(N+1)] × CLKINP/250)
CM_CLKSEL_DPLL_PCIE_REF[2:0] DPLL_EN	0x7	Enables the DPLL in Lock mode
CM_IDLEST_DPLL_PCIE_REF[0] ST_DPLL_CLK	=1	Poll this bit for DPLL lock status. Set by hardware.

(1) The value of the bit field must be set according to the desired clock frequency and M, N and input frequency values.

26.4.4.4.2 PCIe PHY APLL Clock Generator

This section describes the PCIe PHY APLL high speed clock generator APLL_PCIE.

26.4.4.4.2.1 PCIe PHY APLL Clock Generator Overview

The APLL module integrated in the PCIe PHY is a single instance high speed clock generator, used to deliver the high speed clocks to the PCIe PHY RX and TX modules. The APLL_PCIE is directly controlled from the PRCM module and all the necessary control and status signals are exported by the subsystem.

The APLL_PCIEF features:

- Fixed multiplication ratio to generate 2.5 GHz output clock
- High frequency direct output CLKVCOLDO
- High frequency divided output CLKVCOLDO_DIV
- A bypass feature of by-2-divider of CLKVCOLDO_DIV
- Internal voltage controlled oscilator VCO
- Auto Idle mode

26.4.4.4.2.2 PCIe PHY APLL Clock Generator Reset

The PCIe PHY APLL_PCIE clock generator receive hardware non-retention reset, COREAON_PWRON_RST, which comes from the device power and reset manager. For more information on the hardware reset source, see *Reset Domains* in *Power, Reset, and Clock Management*.

The APLL_PCIE itself has no software reset capabilities.

26.4.4.4.2.3 PCIe PHY APLL Low-Power Mode

The APLL_PCIE module supports one type of low power mode controlled by the PRCM register PRCM.CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT bits.

The low-power mode supported by APLL_PCIE is Auto Idle mode.

26.4.4.4.2.4 PCIe PHY APLL Clocks Configuration

26.4.4.4.2.4.1 PCIe PHY APLL Input Clock Control

The APLL_PCIE accepts the functional clock from its input multiplexer, which selects between the DPLL_PCIE_REF output clock CLKREF_ADPLL and the clock from the reference clock buffer ACSPCIE output (delivered from outside the device) CLKREF_ACSPCIE. The selection is made by PRCM register PRCM.CM_CLKMODE_APLL_PCIE[7] REFSEL bit as folows:

- PRCM.CM_CLKMODE_APLL_PCIE[7] REFSEL = 0b0 - input clock CLKREF_ADPLL, APLL reference input clock is from DPLL_PCIE_REF
- PRCM.CM_CLKMODE_APLL_PCIE[7] REFSEL = 0b1 - input clock CLKREF_ACSPCIE, APLL reference input clock is from ACSPCIE

The APLL module does not have software capabilitites for gating the input clock.

26.4.4.4.2.4.2 PCIe PHY APLL Output Clock Configuration

Two high frequency output clocks are available from APLL_PCIE. The direct output CLKVCOLDO supplies 2.5 GHz high speed transmision clock for the PCIe PHY RX module and the divided by 2 clock output CLKVCOLDO_DIV supplies high speed transmision clock (1.25 or 2.5 GHz) for the PCIe PHY TX module. The internal parameters of APLL (multipliers, dividers) are fixed and do not need programming to generate the output clocks. The output clock CLKVCOLDO_DIV is by default the divided by 2 version of the CLKVCOLDO. The divider is controlled by the RATE signal from the PCIe_SS controller. The by-2-divider has a bypass feature to ignore the division control from PCIe_SS and to output a clock with same frequency like CLKVCOLDO. This bypass feature is controlled by PRCM.CM_CLKMODE_APLL_PCIE[8] CLKDIV_BYPASS bit as follows:

- PRCM.CM_CLKMODE_APLL_PCIE[8] CLKDIV_BYPASS = 0b0 - the division by 2 is controlled from Rate signal from PCIe_SS controller;
- PRCM.CM_CLKMODE_APLL_PCIE[8] CLKDIV_BYPASS = 0b1 - the CLKOUTLDO_DIV is not divided by 2 and has same frequency like CLKVCOLDO. Select this option for the current device.

The frequency of CLKVCOLDO_DIV can be changed on-the-fly, but glitches can occur on the clock. Best approach is to disable the CLKVCOLDO_DIV when changing the frequency. See Section 26.4.4.4.2.4.2.1 *PCIe PHY APLL Output Clock Gating* for CLKVCOLDO_DIV gating options.

The availability of the APLL_PCIE output clocks can be monitored in PRCM.CM_CLKVCOLDO_APLL_PCIE register as follows:

- PRCM.CM_CLKVCOLDO_APLL_PCIE[9] CLKST bit gives the status (gated or enabled) of CLKVCOLDO clock
- PRCM.CM_CLKVCOLDO_APLL_PCIE[10] CLK_DIVST bit gives the status (gated or enabled) of CLKVCOLDO_DIV clock.

26.4.4.4.2.4.2.1 PCIe PHY APLL Output Clock Gating

Because the APLL_PCIE.CLKVCOLDO and APLL_PCIE.CLKVCOLDO_DIV are implemented as optional functional clocks the PRCM has software control on them. The two clocks can be controlled by PRCM registers corresponding to the implemented PCIe_SS controllers.

PRCM.CM_PCIE_PCIESS1_CLKCTRL[9] OPTFCLKEN_PCIEPHY_CLK bit controls the CLKVCOLDO clock for the PCIe PHY modules.

PRCM.CM_PCIE_PCIESS1_CLKCTRL[10] OPTFCLKEN_PCIEPHY_CLK_DIV bit controls the CLKVCOLDO_DIV clock for the PCIe PHY modules.

The same controls are duplicated in the registers below:

PRCM.CM_PCIE_PCIESS2_CLKCTRL[9] OPTFCLKEN_PCIEPHY_CLK bit controls the CLKVCOLDO clock for the PCIe PHY modules.

PRCM.CM_PCIE_PCIESS2_CLKCTRL[10] OPTFCLKEN_PCIEPHY_CLK_DIV bit controls the CLKVCOLDO_DIV clock for the PCIe PHY modules.

APLL_PCIE.CLKVCOLDO and APLL_PCIE.CLKVCOLDO_DIV clock outputs are automatically gated (pulled low) in the following scenarios:

- APLL power-up sequence. For more information on power-up sequence, see Section 26.4.4.4.2.6.1, *PCIe PHY APLL Clock Generator Power Up* .
- APLL entering Auto Idle mode.

26.4.4.4.2.5 PCIe PHY APLL Subsystem Architecture

Figure 26-22 is a simplified block diagram of the APLL_PCIE instance integrated in the PCIe PHY clock generator subsystem.

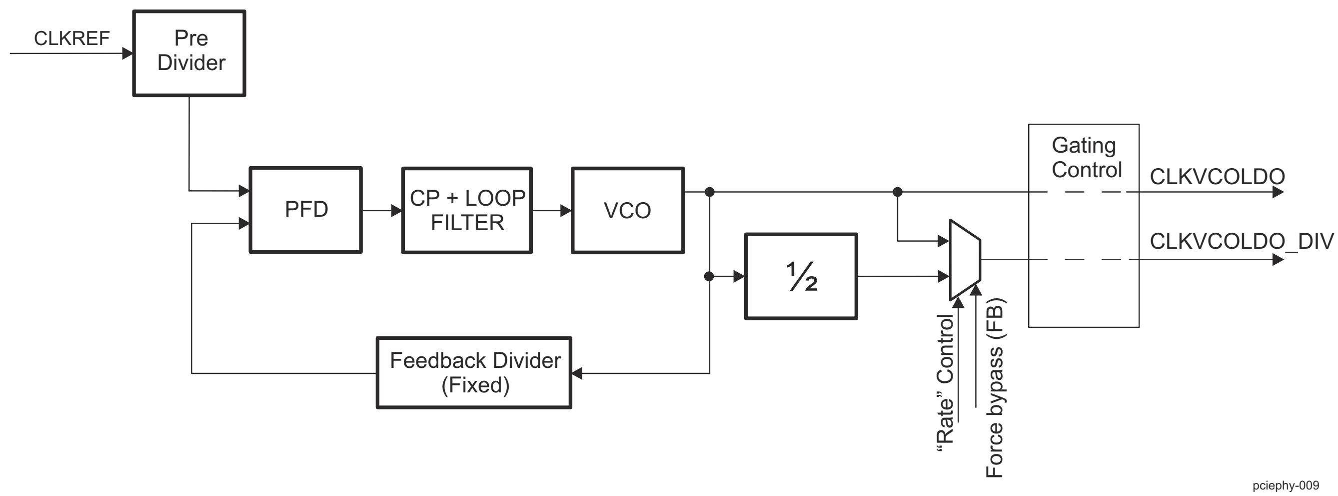


Figure 26-22. APLL_PCIE Functional Block Diagram

The input clock CLKINP goes to a predivider with fixed value. Then goes to the clock generation loop with fixed Feedback Divider. The output of the VCO is split in two. One path directly goes out on CLKVCOLDO output. The other path passes the internal divider by 2 and then goes to bypass multiplexor where is selected the output frequency of the CLKVCOLDO_DIV output. The selection is made either from "Rate" signal from PCIe_SS controller or with higher priority from Force bypass signal controlled from PRCM.CM_CLKMODE_APLL_PCIE[8] CLKDIV_BYPASS bit.

The frequency of the VCO output is fixed to 2.5 GHz and is not programmable in software. No changes could be made on the internal parameters of the APLL_PCIE. Output clocks are controllable through PRCM registers and can be gated by software. See Section 26.4.4.4.2.4.2.1 *PCIe PHY APLL Output Clock Gating* for more information.

26.4.4.4.2.6 PCIe PHY APLL Clock Generator Modes and State Transitions

Note

In order to disable the APLL_PCIE, the user needs to disable PCIe_SSx (where x = 1 or 2) using the CM_PCIE_PCIESSx_CLKCTRL[1:0] MODULEMODE registers. When PCIe_SS is disabled, the PRCM module automatically disables the APLL_PCIE. Please note that setting CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT bitfield to 0x0 does not disable the APLL_PCIE.

The APLL_PCIE can be set in different modes during operation. PRCM triggers APLL_PCIE state transitions to different static modes by setting the bit field of the PRCM.CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT. Only two static modes are available for APLL_PCIE:

- PRCM.CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT = 0x1 – APLL_FORCE_LOCK_MODE is set. This puts the APLL in Force Lock mode
- PRCM.CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT = 0x2 – APLL_FORCE_IDLE_MODE is set. This puts the APLL in Auto Idle mode

For more information see *Power, Reset, and Clock Management*.

26.4.4.4.2.6.1 PCIe PHY APLL Clock Generator Power Up

After power up, the APLL_PCIE.RESETN input is automatically pulled low by the PRM, together with the DPLL_PCIE_REF.SYSRESETN input. Because PRM.COREAON_PWRON_RST is an asynchronous reset, the DPLL_PCIE_REF input clock (DPLL_PCIE_REF.CLKINP) is not demanded upon reset. During APLL power-up mode, CLKVCOLDO and CLKVCOLDO_DIV clocks are maintained inactive (pulled low). After power-up reset, the APLL_LOCK (internal lock loop) signal is maintained deasserted, too. The default value of the mode select register bit field PRCM.CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT = 0x0 puts the DPLL_PCIE_REF in unknown state. It is software responability to change the state of the APLL_PCIE to desired mode after PRCM reset.

26.4.4.4.2.6.2 PCIe PHY APLL Sequences

After setting the APLL_PCIE in Force Lock mode, a inital lock sequence is started and the module enters working state. The output clocks CLKVCOLDO and CLKVCOLDO_DIV are gated until the lock sequence is finished and APLL_LOCK signal is aserted. After that all clocks are operational and their states depend on the software control.

26.4.4.4.2.6.3 PCIe PHY APLL Locked Mode

When APLL_PCIE finishes calibration and lock sequences it enters a locked state. APLL_PCIE locked state is indicated by PRCM.CM_IDLEST_APLL_PCIE[0] ST_APLL_CLK bit asserted to 0b1. In locked mode all the parameters of DPLL_PCIE_REF are set and the loop is running. The output clocks CLKVCOLDO and CLKVCOLDO_DIV are active under software control.

26.4.4.4.3 ACSPCIE reference clock buffer

The ACSPCIE module is a clock buffer circuit, which has both receive (RX) and transmission (TX) mode. This clock buffer is used to provide low jitter input clock to APLL. In receive mode it takes in an external differential clock and converts it to single-ended CMOS level clock signal. In transmission mode it takes in an internal single-ended clock and converts it to differential HCSL[1] level (in presence of external terminations). The I/O pins of ACSPCIE are *ljcb_clkp* and *ljcb_clkn* differential pair.

- Receive mode: RX is a clock slicer that receives HCSL or LVDS differential clock (typically from on-board Crystal Oscillator) between ljcb_clkp and ljcb_clkn and converts it to CMOS single-ended output CLKREF_ACSPCIE. This CMOS clock signal is the input clock for APLL.
- Transmission mode: TX is a current switching driver that receives CMOS single-ended clock CLKREF_ADPLL at it's input pin from DPLL_PCIE_REF and converts it to HCSL differential-ended output ljcb_clkp and ljcb_clkn. The termination for TX is on-board and 50 Ohms single-ended or 100 Ohms differential.

The ACSPCIE module has two modes of operation:

- Functional mode: The module is powered up automatically and acts as either an input or output buffer.
- Power-down mode: The module is powered down. RX output is held low and TX output is tri-stated/ terminated.

26.4.5 PCIePHY Subsystem Low-Level Programming Model

The low-level programming sequence to set up the PCIe PHY subsystem is summarized in the Table 26-56.

Note

Registers prefixed with CM_ are system Clock Manager registers and are described in *Power, Reset, and Clock Management*.

Registers prefixed with CTRL_ are system Control Module registers and are described in *Control Module*.

Table 26-56. PCIePHY Subsystem Low-Level Programming Sequence

Step	Register/Bit Field/Programming Model	Value
Start a software forced wake-up transition on the PCIE clock domain	CM_PCIE_CLKSTCTRL[1:0] CLKTRCTRL	0x2
Start a software forced wake-up transition on the L3INIT clock domain	CM_L3INIT_CLKSTCTRL[1:0] CLKTRCTRL	0x2
Configure the PCIESS1 module to be explicitly enabled	CM_PCIE_PCIESS1_CLKCTRL[1:0] MODULEMODE	0x2
Poll for PCIESS1 module fully functional?	CM_PCIE_PCIESS1_CLKCTRL[17:16] IDLEST	=0x0

Table 26-56. PCIePHY Subsystem Low-Level Programming Sequence (continued)

Step	Register/Bit Field/Programming Model	Value
Optional: Configure the PCIESS2 module to be explicitly enabled	CM_PCIE_PCIESS2_CLKCTRL[1:0] MODULEMODE	0x2
Optional: Poll for PCIESS2 module fully functional?	CM_PCIE_PCIESS2_CLKCTRL[17:16] IDLEST	=0x0
Configure the OCP2SCP3 module to be managed automatically by hardware according to clock domain transition	CM_L3INIT_OCP2SCP3_CLKCTRL[1:0] MODULEMODE	0x1
Perform a software reset on OCP2SCP3	OCP2SCP_SYSCONFIG[1] SOFTRESET	1
Wait until reset is finished?	OCP2SCP_SYSSTATUS[0] RESETDONE	=1
Configure the OCP2SCP3 Division Ratio and SYNC values	OCP2SCP_TIMING	0x8F
Configure DPLL_PCIE_REF registers to select CLKOUTLDO = 100 MHz. Lock the PLL.	See Section 26.4.4.4.1.6.7, *PCIe PHY DPLL Recommended Values*.	
Select the desired direction of the ACSPCIE buffer (connected to the ljcp_clkn/ljcp_clkp pins)	CTRL_CORE_SMA_SW_6[17:16] PCIE_TX_RX_CONTROL	0x1 (output) 0x2 (input)
Select the APLL_PCIE 100MHz reference clock source	CM_CLKMODE_APLL_PCIE[7] REFSEL	0 (DPLL_PCIE) 1 (ACSPCIE)
Configure APLL_PCIE registers to select CLKVCOLDO = 2.5 GHz and CLKVCOLDO_DIV = 2.5 GHz	See Section 26.4.4.4.2.4, *PCIe PHY APLL Clocks Configuration*.	
Request the APLL_PCIE Force Lock mode	CM_CLKMODE_APLL_PCIE[1:0] MODE_SELECT	0x1
Wait for APLL_PCIE Lock	CM_IDLEST_APLL_PCIE[0] ST_APLL_CLK	=1
Configure the PCIE PHYs to ×1 or ×2 mode	CTRL_CORE_PCIE_CONTROL[3:2] PCIE_B1C0_MODE_SEL	0x0 (×1 mode) 0x1 (×2 mode)
	CTRL_CORE_PCIE_CONTROL[0] PCIE_B0_B1_TSYNCEN	0 (×1 mode) 1 (×2 mode)
Enable the CLKVCOLDO clock for the PCIESS1 PHY	CM_PCIE_PCIESS1_CLKCTRL[9] OPTFCLKEN_PCIEPHY_CLK	1
Enable the CLKVCOLDO_DIV clock for the PCIESS1 PHY	CM_PCIE_PCIESS1_CLKCTRL[10] OPTFCLKEN_PCIEPHY_CLK_DIV	1
IF: PCIESS2 is enabled	Software test condition	
Enable the CLKVCOLDO clock for the PCIESS2 PHY	CM_PCIE_PCIESS2_CLKCTRL[9] OPTFCLKEN_PCIEPHY_CLK	1
Enable the CLKVCOLDO_DIV clock for the PCIESS2 PHY	CM_PCIE_PCIESS2_CLKCTRL[10] OPTFCLKEN_PCIEPHY_CLK_DIV	1
ENDIF		
Configure the PCIESS1 Power Control clock frequency to match SYS_CLK1 in MHz	CTRL_CORE_PHY_POWER_PCIESS1[31:22] PCIESS1_PWRCTL_CLKFREQ	0x14 (20 MHz)
Power up PCIESS1_PHY_TX and PCIESS1_PHY_RX	CTRL_CORE_PHY_POWER_PCIESS1[21:14] PCIESS1_PWRCTL_CMD	0x3
IF: Either PCIESS1 Second Lane or PCIESS2 is required	Software test condition	
Configure the PCIESS2 Power Control clock frequency to match SYS_CLK1 in MHz	CTRL_CORE_PHY_POWER_PCIESS2[31:22] PCIESS2_PWRCTL_CLKFREQ	0x14 (20 MHz)
Power up PCIESS2_PHY_TX and PCIESS2_PHY_RX	CTRL_CORE_PHY_POWER_PCIESS2[21:14] PCIESS2_PWRCTL_CMD	0x3
ENDIF		
Configure the proper Delay Count	CTRL_CORE_PCIE_PCS[23:16] PCIESS_PCS_RC_DELAY_COUNT	0x96
Configure PCIe_PHY_RX SCP Settings	See PCIe_PHY_RX preferred SCP settings in Table 26-57.	
PHY_TX settings must remain at default values.	No additional tuning in PHY_TX SCP registers is required.	

Table 26-57. Preferred PCIe_PHY_RX SCP Register Settings

Register	Preferred Value Setting
PCIEPHYRX_ANA_PROGRAMMABILITY_REG1[31:27] MEM_ANATESTMODE	0b00001
PCIEPHYRX_ANA_PROGRAMMABILITY_REG1[17:14] MEM_ANATESTMODE	0b1010
PCIEPHYRX_DIGITAL_MODES_REG1[23] MEM_CDR_FASTLOCK	0b1
PCIEPHYRX_DIGITAL_MODES_REG1[22:21] MEM_CDR_LBW	0b11
PCIEPHYRX_DIGITAL_MODES_REG1[20:19] MEM_CDR_STEPCNT	0b00
PCIEPHYRX_DIGITAL_MODES_REG1[18:16] MEM_CDR_STL	0b011
PCIEPHYRX_DIGITAL_MODES_REG1[15:13] MEM_CDR_THR	0b001
PCIEPHYRX_DIGITAL_MODES_REG1[12] MEM_CDR_THR_MODE	0b1
PCIEPHYRX_DIGITAL_MODES_REG1[11] MEM_CDR_2NDO_SDM_MODE	0b0
PCIEPHYRX_DIGITAL_MODES_REG1[26] MEM_OVRD_HS_RATE	0b0
PCIEPHYRX_ANA_PROGRAMMABILITY_REG1[6:5] MEM_PLLDIV	0b00
PCIEPHYRX_TRIM_REG4[31:30] MEM_DLL_TRIM_SEL	0b10
PCIEPHYRX_DLL_REG1[31:30] MEM_DLL_PHINT_RATE	0b11
PCIEPHYRX_EQUALIZER_REG1[31:16] MEM_EQLEV	0b0000 0000 0000 0000
PCIEPHYRX_EQUALIZER_REG1[15:11] MEM_EQFTC	0b11111
PCIEPHYRX_EQUALIZER_REG1[10:7] MEM_EQCTL	0b0001
PCIEPHYRX_EQUALIZER_REG1[2] MEM_OVRD_EQLEV	0b0
PCIEPHYRX_EQUALIZER_REG1[1] MEM_OVRD_EQFTC	0b0

26.4.6 PCIe PHY Subsystem Register Manual

This chapter summarizes and describes the shared PHY component registers for the PCIe PHY subsystems.

26.4.6.1 PCIe PHY Instance Summary

Table 26-58. PCIe PHY Subsystem Instance Summary

Module Name	Module Base Address	Size
PCIe1_PHY_RX	0x4A09 4000	1 KiB
PCIe1_PHY_TX	0x4A09 4400	1 KiB
PCIe2_PHY_RX	0x4A09 5000	1 KiB
PCIe2_PHY_TX	0x4A09 5400	1 KiB
OCP2SCP1	0x4A08 0000	1 KiB
OCP2SCP3	0x4A09 0000	1 KiB

26.4.6.1.1 PCIe_PHY_RX Registers

26.4.6.1.1.1 PCIe_PHY_RX Register Summary

Table 26-59. PCIe1_PHY_RX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe1_PHY_RX Physical Address
RESERVED	R	32	0x0000 0000	0x4A09 4000
RESERVED	R	32	0x0000 0004	0x4A09 4004
RESERVED	R	32	0x0000 0008	0x4A09 4008
PCIEPHYRX_ANA_PROGRAMMABILITY_REG1	RW	32	0x0000 000C	0x4A09 400C
RESERVED	R	32	0x0000 0010	0x4A09 4010
RESERVED	R	32	0x0000 0014	0x4A09 4014
RESERVED	R	32	0x0000 0018	0x4A09 4018
PCIEPHYRX_TRIM_REG4	RW	32	0x0000 001C	0x4A09 401C
RESERVED	R	32	0x0000 0020	0x4A09 4020
PCIEPHYRX_DLL_REG1	RW	32	0x0000 0024	0x4A09 4024
PCIEPHYRX_DIGITAL_MODES_REG1	RW	32	0x0000 0028	0x4A09 4028
RESERVED	R	32	0x0000 002C	0x4A09 402C
RESERVED	R	32	0x0000 0030	0x4A09 4030
RESERVED	R	32	0x0000 0034	0x4A09 4034
PCIEPHYRX_EQUALIZER_REG1	RW	32	0x0000 0038	0x4A09 4038
RESERVED	R	32	0x0000 003C	0x4A09 403C
RESERVED	R	32	0x0000 0040	0x4A09 4040
RESERVED	R	32	0x0000 0044	0x4A09 4044
RESERVED	R	32	0x0000 0048	0x4A09 4048
RESERVED	R	32	0x0000 004C	0x4A09 404C
RESERVED	R	32	0x0000 0050	0x4A09 4050
RESERVED	R	32	0x0000 0054	0x4A09 4054
RESERVED	R	32	0x0000 0058	0x4A09 4058
RESERVED	R	32	0x0000 005C	0x4A09 405C
RESERVED	R	32	0x0000 0060	0x4A09 4060
RESERVED	R	32	0x0000 0064	0x4A09 4064

Table 26-60. PCIe2_PHY_RX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe2_PHY_RX Physical Address
RESERVED	R	32	0x0000 0000	0x4A09 5000

Table 26-60. PCIe2_PHY_RX Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PCIe2_PHY_RX Physical Address
RESERVED	R	32	0x0000 0004	0x4A09 5004
RESERVED	R	32	0x0000 0008	0x4A09 5008
PCIEPHYRX_ANA_PROGRAMMABILITY_REG1	RW	32	0x0000 000C	0x4A09 500C
RESERVED	R	32	0x0000 0010	0x4A09 5010
RESERVED	R	32	0x0000 0014	0x4A09 5014
RESERVED	R	32	0x0000 0018	0x4A09 5018
PCIEPHYRX_TRIM_REG4	RW	32	0x0000 001C	0x4A09 501C
RESERVED	R	32	0x0000 0020	0x4A09 5020
PCIEPHYRX_DLL_REG1	RW	32	0x0000 0024	0x4A09 5024
PCIEPHYRX_DIGITAL_MODES_REG1	RW	32	0x0000 0028	0x4A09 5028
RESERVED	R	32	0x0000 002C	0x4A09 502C
RESERVED	R	32	0x0000 0030	0x4A09 5030
RESERVED	R	32	0x0000 0034	0x4A09 5034
PCIEPHYRX_EQUALIZER_REG1	RW	32	0x0000 0038	0x4A09 5038
RESERVED	R	32	0x0000 003C	0x4A09 503C
RESERVED	R	32	0x0000 0040	0x4A09 5040
RESERVED	R	32	0x0000 0044	0x4A09 5044
RESERVED	R	32	0x0000 0048	0x4A09 5048
RESERVED	R	32	0x0000 004C	0x4A09 504C
RESERVED	R	32	0x0000 0050	0x4A09 5050
RESERVED	R	32	0x0000 0054	0x4A09 5054
RESERVED	R	32	0x0000 0058	0x4A09 5058
RESERVED	R	32	0x0000 005C	0x4A09 505C
RESERVED	R	32	0x0000 0060	0x4A09 5060
RESERVED	R	32	0x0000 0064	0x4A09 5064

26.4.6.1.1.2 PCIe_PHY_RX Register Description

Table 26-61. PCIEPHYRX_ANA_PROGRAMMABILITY_REG1

Address Offset	0x0000 000C		
Physical Address	0x4A09 400C 0x4A09 500C	**Instance**	PCIe1_PHY_RX PCIe2_PHY_RX
Description	Programmability for different analog circuits in the PHY.		
Type	RW		

31–8	7	6–5	4–0
MEM_ANATESTMODE	RESERVED	MEM_PLLDIV	RESERVED

Bits	Field Name	Description	Type	Reset
31:8	MEM_ANATESTMODE	Programmability for Analog circuits in the PHY. The top 5 bits - MEM_ANATESTMODE[31:27] indicate the serial Interface using this PHY module. Bits [17:14] are used to control loss-of-signal detection (LOSD) threshold.	RW	0x00 0000
7	RESERVED		RW	0

Bits	Field Name	Description	Type	Reset
6:5	MEM_PLLDIV	This is a test mode. SoC Users are requested to leave this at default value. The input PLL_CLK (after being muxed with PLLBYPCLK) is divided by the following factors indicated by this register. 00=1 01=2 10=4 11=RESERVED. All references to PLL_CLK in this register descriptions are AFTER considering this division.	RW	0x0
4:0	RESERVED		R	0x00

Table 26-62. PCIEPHYRX_TRIM_REG4

Address Offset	0x0000 001C		
Physical Address	0x4A09 401C 0x4A09 501C	**Instance**	PCIe1_PHY_RX PCIe2_PHY_RX
Description	The IP requires some values to be remembered in EFUSE. This register provides an alternative to EFUSE.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM_DLL_TRIM_SEL		RESERVED																													

Bits	Field Name	Description	Type	Reset
31:30	MEM_DLL_TRIM_SEL	Determines which of the 4 EFUSE registers EFUSE_dll_rateN_coarsetrim should be used as the trim code by the DLL. This feature is so that the user may find and store the trim codes corresponding to different (at most 4) DLL frequencies (pll_clk pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) frequencies) and at wake-up, instruct the IP to choose one of these available trim values depending on the Application's frequency requirement. 00 selects dll_rate0_coarsetrim 01 selects dll_rate1_coarsetrim 10 selects dll_rate2_coarsetrim 11 selects dll_rate3_coarsetrim.	RW	0x0
29:0	RESERVED		RW	0x0000 0000

Table 26-63. PCIEPHYRX_DLL_REG1

Address Offset	0x0000 0024		
Physical Address	0x4A09 4024 0x4A09 5024	**Instance**	PCIe1_PHY_RX PCIe2_PHY_RX
Description	This register is used to program DLL settings.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM_DLL_PHINT_RATE		RESERVED																													

Bits	Field Name	Description	Type	Reset
31:30	MEM_DLL_PHINT_RATE	Programs the DLL and the Phase Interpolator analog circuits to work with different clock frequencies. The frequency of pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) should be indicated by this register. 00=0.625GHz to 0.75GHz 01=RESERVED 10=1.25GHz to 1.5GHz 11=2.5GHz to 2.9GHz.	RW	0x3
29:0	RESERVED		R	0x00A4 1915

Table 26-64. PCIEPHYRX_DIGITAL_MODES_REG1

Address Offset	0x0000 0028		
Physical Address	0x4A09 4028 0x4A09 5028	**Instance**	PCIe1_PHY_RX PCIe2_PHY_RX
Description	This register contains control bits which affect different circuits in digital section		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM_INV_RXPN_PAIR	MEM_OVRD_INV_RXPN_PAIR	RESERVED	MEM_HS_RATE		MEM_OVRD_HS_RATE	RESERVED		MEM_CDR_FASTLOCK	MEM_CDR_LBW		MEM_CDR_STEPCNT		MEM_CDR_STL			MEM_CDR_THR			MEM_CDR_THR_MODE	MEM_CDR_2NDO_SDM_MODE	RESERVED										

Bits	Field Name	Description	Type	Reset
31	MEM_INV_RXPN_PAIR	If '1', interchanges RXP and RXN effectively by inverting the received data samples.	RW	0
30	MEM_OVRD_INV_RXPN_PAIR	Pin override control. See register bit MEM_inv_rxpn_pair.	RW	0
29	RESERVED		R	0
28:27	MEM_HS_RATE	Determines the ratio of pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) frequency and the output data rate. Full Rate means pll_clk (after the bypassing by MEM_en_pllbyp and the division by MEM_plldiv) frequency = Data Rate/2 00=Full Rate 01=Half Rate 10=Quarter Rate 11=RESERVED. This takes effect only if register bit MEM_ovrd_hs_rate is '1', else the same is controlled by input pins hs_rate.	RW	0x0
26	MEM_OVRD_HS_RATE	Pin override control. See register bit MEM_hs_rate.	RW	0
25:24	RESERVED		R	0x2
23	MEM_CDR_FASTLOCK	'1' to reduce lock time of CDR (clock-data-recovery circuit).	RW	1
22:21	MEM_CDR_LBW	CDR band-width control.	RW	0x3
20:19	MEM_CDR_STEPCNT	CDR 2nd order setting.	RW	0x0

Bits	Field Name	Description	Type	Reset
18:16	MEM_CDR_STL	CDR settling time. Determines the number of vote clocks to blank ELV (Early-Late-Voter circuit) after update of phase.	RW	0x3
15:13	MEM_CDR_THR	CDR 1st order threshold. Determines how much early/ late votes should differ by before a phase change in the receiver sampling clock is triggered.	RW	0x1
12	MEM_CDR_THR_MODE	CDR 1st order threshold.	RW	1
11	MEM_CDR_2NDO_SDM_MODE	If '1', the 2nd Order CDR block uses a 1st order Sigma Delta Modulator to accomplish frequency offset If '0', a simple rate transformer is used for the same purpose.	RW	0
10:0	RESERVED		R	0x000

Table 26-65. PCIEPHYRX_EQUALIZER_REG1

Address Offset	0x0000 0038		
Physical Address	0x4A09 4038 0x4A09 5038	**Instance**	PCIe1_PHY_RX PCIe2_PHY_RX
Description	The module has an Equalizer (with analog and digital parts) which addresses Inter Symbol Interference (ISI). The equalizer can be configured via the EQCTL bits. The options are: No adaptive equalization. The equalizer provides a flat response at the maximum gain. This setting may be appropriate if jitter at the receiver occurs predominantly as a result of crosstalk rather than frequency dependent loss. Fully adaptive equalization. Both the low frequency gain and zero position of the equalizer are determined algorithmically by analysing the data patterns and transition positions in the received data. This setting should be used for most applications. Partially adaptive equalization. The low frequency gain of the equalizer is determined algorithmically by analysing the data patterns and transition positions in the received data. The zero position is fixed in one of eight zero positions. When enabled, the receiver equalization logic analyzes data patterns and transition times to determine whether the low frequency gain of the equalizer should be increased or decreased. For the fully adaptive setting (EQCTL = 0001), if the low frequency gain reaches the minimum value, the zero frequency is then reduced. Likewise, if it reaches the maximum value, the zero frequency is then increased.		
Type	RW		

31:16	15:11	10:8	7:3	2	1	0
MEM_EQLEV	MEM_EQFTC	MEM_EQCTL	RESERVED	MEM_OVRD_EQLEV	MEM_OVRD_EQFTC	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	MEM_EQLEV	Equalizer level control.	RW	0x0000
15:11	MEM_EQFTC	Equalizer zero freq control.	RW	0x00

Bits	Field Name	Description	Type	Reset
10:7	MEM_EQCTL	0000 - Equalizer disabled 0001 - Fully adaptive; FTC normal 0010 - Fully adaptive; FTC inverted 0011 - Hold equalizer state 01xx - Init equalizer to fully adaptive start/midpoint 1000 - Partially adaptive; zero=1084 MHz 1001 - Partially adaptive; zero= 805 MHz 1010 - Partially adaptive; zero= 573 MHz 1011 - Partially adaptive; zero= 402 MHZ 1100 - Partially adaptive; zero= 304 MHz 1101 - Partially adaptive; zero= 216 MHz 1110 - Partially adaptive; zero= 156 MHz 1111 - Partially adaptive; zero= 135 MHz	RW	0x0
6:3	RESERVED		R	0
2	MEM_OVRD_EQLEV	Continuosly forces the Equalizer output with the MEM_EQLEV[15:0].	RW	0
1	MEM_OVRD_EQFTC	Continuosly forces the Equalizer output with the MEM_EQFTC[4:0].	RW	0
0	RESERVED		R	0

26.4.6.1.2 PCIe_PHY_TX Registers

26.4.6.1.2.1 PCIe_PHY_TX Register Summary

Table 26-66. PCIe1_PHY_TX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe1_PHY_TX Physical Address
RESERVED	R	32	0x0000 0000	0x4A09 4400
RESERVED	R	32	0x0000 0004	0x4A09 4404
RESERVED	R	32	0x0000 0008	0x4A09 4408
PCIEPHYTX_FUNC_CONFIG_REG	RW	32	0x0000 000C	0x4A09 440C
PCIEPHYTX_DRIVER_DATA_CONFIG1	R	32	0x0000 0010	0x4A09 4410
RESERVED	R	32	0x0000 0014	0x4A09 4414
RESERVED	R	32	0x0000 0018	0x4A09 4418
RESERVED	R	32	0x0000 001C	0x4A09 441C
RESERVED	R	32	0x0000 0020	0x4A09 4420
RESERVED	R	32	0x0000 0024	0x4A09 4424
RESERVED	R	32	0x0000 0028	0x4A09 4428
PCIEPHYTX_TEST_CONFIG_REG	RW	32	0x0000 002C	0x4A09 442C
PCIEPHYTX_PATTGEN_PRELOAD	RW	32	0x0000 0030	0x4A09 4430
RESERVED	R	32	0x0000 0034	0x4A09 4434

Table 26-67. PCIe2_PHY_TX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PCIe2_PHY_TX Physical Address
RESERVED	R	32	0x0000 0000	0x4A09 5400
RESERVED	R	32	0x0000 0004	0x4A09 5404
RESERVED	R	32	0x0000 0008	0x4A09 5408
PCIEPHYTX_FUNC_CONFIG_REG	RW	32	0x0000 000C	0x4A09 540C
PCIEPHYTX_DRIVER_DATA_CONFIG1	R	32	0x0000 0010	0x4A09 5410
RESERVED	R	32	0x0000 0014	0x4A09 5414
RESERVED	R	32	0x0000 0018	0x4A09 5418
RESERVED	R	32	0x0000 001C	0x4A09 541C
RESERVED	R	32	0x0000 0020	0x4A09 5420
RESERVED	R	32	0x0000 0024	0x4A09 5424
RESERVED	R	32	0x0000 0028	0x4A09 5428
PCIEPHYTX_TEST_CONFIG_REG	RW	32	0x0000 002C	0x4A09 542C
PCIEPHYTX_PATTGEN_PRELOAD	RW	32	0x0000 0030	0x4A09 5430
RESERVED	R	32	0x0000 0034	0x4A09 5434

26.4.6.1.2.2 PCIe_PHY_TX Register Description

Table 26-68. PCIEPHYTX_FUNC_CONFIG_REG

Address Offset	0x0000 000C		
Physical Address	0x4A09 440C 0x4A09 540C	**Instance**	PCIe1_PHY_TX PCIe2_PHY_TX
Description	Functional Configuration registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31	30:0
MEM_INVPAIR	RESERVED

Bits	Field Name	Description	Type	Reset
31	MEM_INVPAIR	Invert polarity of TXP/TXN	RW	0
30:0	RESERVED		R	0x0000 0000

Table 26-69. PCIEPHYTX_DRIVER_DATA_CONFIG1

Address Offset	0x0000 0010		
Physical Address	0x4A09 4410 0x4A09 5410	**Instance**	PCIe1_PHY_TX PCIe2_PHY_TX
Description	Configures the Driver data pattern		
Type	RW		

31:25	24:18	17:11	10:4	3:2	1	0
MEM_EVEN_OUT_CONFIG0	MEM_ODD_OUT_CONFIG0	MEM_EVEN_OUT_CONFIG1	MEM_ODD_OUT_CONFIG1	MEM_HS_RATE_ANA_OVERRIDE	MEM_OVRD_HS_RATE_ANA_OVERRIDE	RESERVED

Bits	Field Name	Description	Type	Reset
31:25	MEM_EVEN_OUT_CONFIG0	Overriding the even TX data driver - to AFE	RW	0x0
24:18	MEM_ODD_OUT_CONFIG0	Overriding the odd TX data driver - to AFE	RW	0x0
17:11	MEM_EVEN_OUT_CONFIG1	Overriding the even TX data driver - to AFE	RW	0x0
10:4	MEM_ODD_OUT_CONFIG1	Overriding the odd TX data driver - to AFE	RW	0x0
3:2	MEM_HS_RATE_ANA_OVERRIDE	Override for the HS rate signal going to the AFE	RW	0x0
1	MEM_OVRD_HS_RATE_ANA_OVERRIDE	Pin override for the hs_rate_ana_override	RW	0x0
0	RESERVED	Reserved	RW	0x0

Table 26-70. PCIEPHYTX_TEST_CONFIG_REG

Address Offset	0x0000 002C		
Physical Address	0x4A09 442C 0x4A09 542C	**Instance**	PCIe1_PHY_TX PCIe2_PHY_TX
Description	Test related configuration registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	MEM_EN_LPBK	MEM_ENTXPATT	MEM_TESTPATT	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED	Keep at 0	R	0
30	MEM_EN_LPBK	Loopback enable for test	RW	0
29	MEM_ENTXPATT	Enable Test pattern to input of the serializer instead of TD	RW	0
28:26	MEM_TESTPATT	Select the LFSR mode to generate the required pattern 000 - 31 bit LFSR mode 011 - 23 bit LFSR mode 010 - 7 bit LFSR mode 001 - generate 1010 pattern 100 - Fixed 31 bit value from pattgen_preload_val	RW	0x0
25:0	RESERVED		RW	0x0

Table 26-71. PCIEPHYTX_PATTGEN_PRELOAD

Address Offset	0x0000 0030		
Physical Address	0x4A09 4430 0x4A09 5430	**Instance**	PCIe1_PHY_TX PCIe2_PHY_TX
Description	Pattern generator (31 bit) LFSR Seed or preload value		
Type	RW		

31–1	0
MEM_PATTGEN_PRELOAD_VAL	RESERVED

Bits	Field Name	Description	Type	Reset
31:1	MEM_PATTGEN_PRELOAD_VAL	Preload value to the LFSR pattern generator	RW	0x0000 0000
0	RESERVED		RW	0

26.4.6.1.3 OCP2SCP Registers

26.4.6.1.3.1 OCP2SCP Register Summary

Table 26-72. OCP2SCP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCP2SCP1 Physical Address	OCP2SCP3 Physical Address
OCP2SCP_REVISION	R	32	0x0000 0000	0x4A08 0000	0x4A09 0000
OCP2SCP_SYSCONFIG	RW	32	0x0000 0010	0x4A08 0010	0x4A09 0010
OCP2SCP_SYSSTATUS	R	32	0x0000 0014	0x4A08 0014	0x4A09 0014
OCP2SCP_TIMING	RW	32	0x0000 0018	0x4A08 0018	0x4A09 0018

26.4.6.1.3.2 OCP2SCP Register Description

Table 26-73. OCP2SCP_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A08 0000 0x4A09 0000	**Instance**	OCP2SCP1 OCP2SCP3
Description	Revision register		
Type	R		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REVISION</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Module revision number	R	0x-

Table 26-74. OCP2SCP_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A08 0010 0x4A09 0010	**Instance**	OCP2SCP1 OCP2SCP3
Description	System configuration register		
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="27">RESERVED</td><td colspan="2">IDLEMODE</td><td>RESERVED</td><td>SOFTRESET</td><td>AUTOIDLE</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x000 0000
4:3	IDLEMODE	Idle mode 0x0: Force Idle mode. An idle request is acknowledged unconditionally. 0x1: No Idle mode. An idle request is never acknowledged. 0x2: Smart Idle mode. The acknowledgement to an idle request is given based on the internal activity. 0x3: Smart Idle Wakeup.	RW	0x2
2	RESERVED	Reserved.	R	0
1	SOFTRESET	Software Reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal Mode 0x1: The module is reset.	RW	0

Bits	Field Name	Description	Type	Reset
0	AUTOIDLE	OCP interface clock gating control. 0x0: Internal OCP interface clock is free-running 0x1: Automatic internal OCP interface clock gating, based on the OCP interface activity	RW	1

Table 26-75. OCP2SCP_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x4A08 0014 0x4A09 0014	**Instance**	OCP2SCP1 OCP2SCP3
Description	System Status register.		
Type	R		

31–1	0
RESERVED	RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Reset done Read 0x1: Reset completed Read 0x0: Internal Reset is on-going	R	1

Table 26-76. OCP2SCP_TIMING

Address Offset	0x0000 0018		
Physical Address	0x4A08 0018 0x4A09 0018	**Instance**	OCP2SCP1 OCP2SCP3
Description	Timing register		
Type	RW		

31–10	9–7	6–4	3–0
RESERVED	DIVISIONRATIO	SYNC1	SYNC2

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved.	R	0x00 0000
9:7	DIVISIONRATIO(1)	Division Ration of the SCP clock in relation to OCP input clock.	RW	0x0
6:4	SYNC1	Number of SCPclock cycles defining SYNC1	RW	0x0
3:0	SYNC2	Number of SCPclock cycles defining SYNC2	RW	0x1

(1) When value “000” is programmed for the SCP clock division ratio, and the transaction to be made is a valid transaction on the SCP interface, the DIVISIONRATIO value is set internally to 0x7 (to avoid a block on the OCP interface).

CAUTION

To ensure correct operation, DIVISIONRATIO must not be modified. CAUTION: To ensure correct operation, the value of SYNC2 must be set to 0x6 or more.

Chapter 27

General-Purpose Interface

Texas Instruments

This chapter describes the general-purpose interface for the device.

27.1 General-Purpose Interface Overview.......... 6615
27.2 General-Purpose Interface Environment.......... 6618
27.3 General-Purpose Interface Integration.......... 6623
27.4 General-Purpose Interface Functional Description.......... 6628
27.5 General-Purpose Interface Programming Guide.......... 6643
27.6 General-Purpose Interface Register Manual.......... 6645

27.1 General-Purpose Interface Overview

The general-purpose interface combines eight general-purpose input/output (GPIO) banks.

Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 256 (8 × 32) pins. Some of the pins may be reserved in this Device. For more information of the supported number of GPIO pins, see the device Data Manual.

These pins can be configured for the following applications:

- Data input (capture)/output (drive)
- Keyboard interface with a debounce cell
- Interrupt generation in active mode upon the detection of external events. Detected events are processed by two parallel independent interrupt-generation submodules to support biprocessor operations.
- Wake-up request generation in idle mode upon the detection of external events

These modules do not include pad control (pullup/down control, open-drain feature). For more information, see *Pad Configuration Registers,* in *Control Module*.

Figure 27-1 is an overview of the general-purpose interface.

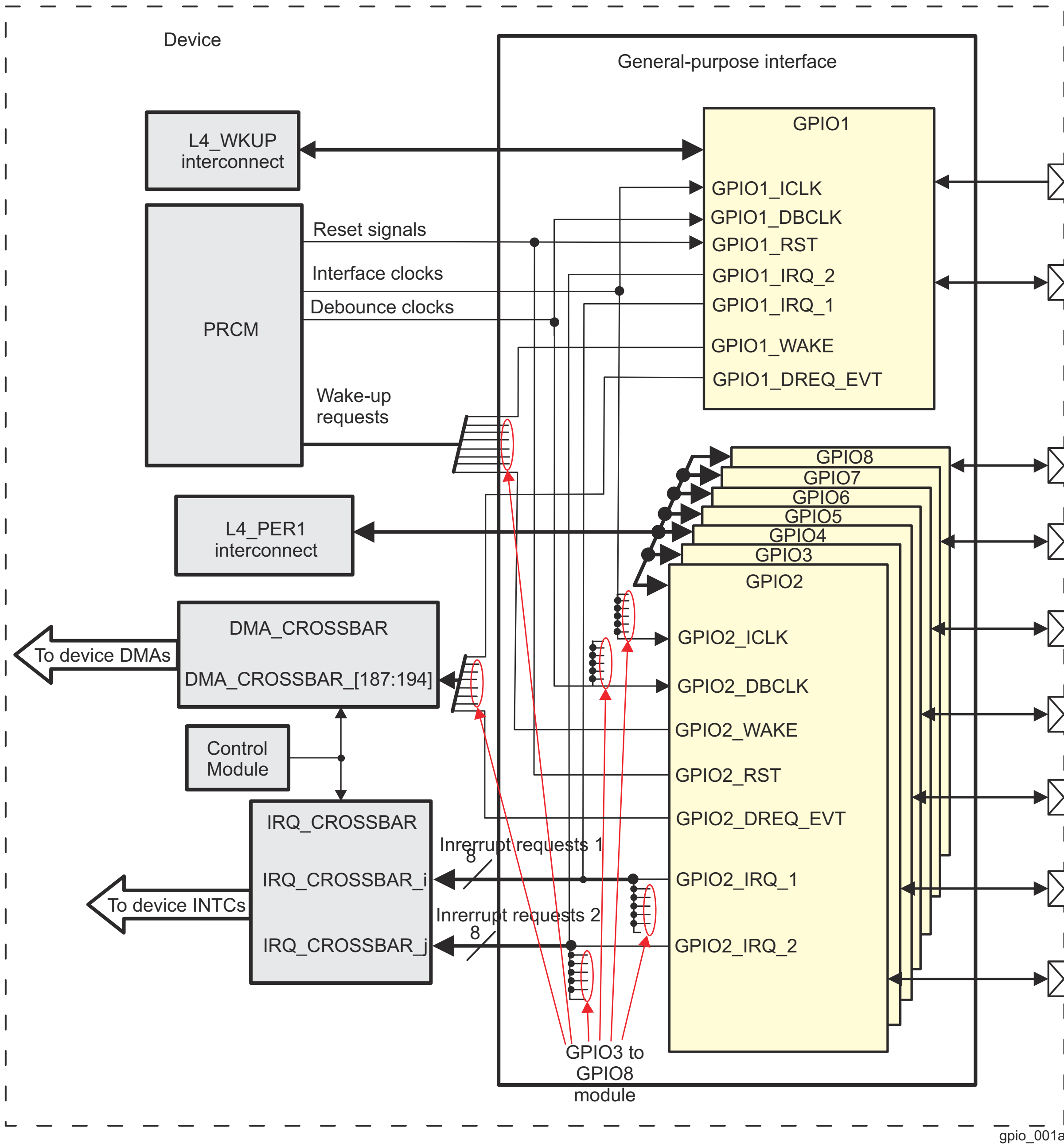


Figure 27-1. General-Purpose Interface Overview

The GPIO modules include the following global features:

- Two identical submodules can process synchronous interrupt requests from each channel to be used independently in a biprocessor environment. Each submodule controls its own synchronous interrupt request line. Each submodule also has its own interrupt-enable and interrupt status registers. The interrupt-enable register (GPIO_IRQSTATUS_SET_x [where x = 0 or 1]) selects the channel considered for the interrupt request generation. The interrupt status register (GPIO_IRQSTATUS_RAW_x) determines which channel has activated the interrupt request. Event detection on GPIO channels is reflected into GPIO_IRQSTATUS_RAW independently from the content of the interrupt-enable registers.
- Wake-up requests in idle mode from input channels are merged together to issue one wake-up signal per GPIO module.
- Data input (capture)/output (drive)

- Power-management support

The general-purpose interface has 16 interrupt lines (two interrupt lines on GPIO1 through GPIO8 modules).

Each GPIO module produces a wake-up request signal to the power, reset, and clock management (PRCM) module.

Each channel in the GPIO modules has the following features:

- The GPIOi.GPIO_OE register controls the output capability for each pin.
- The output line level reflects the value written in the GPIOi.GPIO_DATAOUT register through the level 4 (L4_WKUP and L4_PER1) interconnect.
- The input line can be fed to the GPIO module through an optional and configurable debounce cell. (Because the debouncing time value is global for all ports of one GPIO module, up to five different debouncing time values are possible.)
- The value of the input line is sampled into the GPIOi.GPIO_DATAIN register and can be read through the L4 (L4_WKUP and L4_PER1) interconnect.
- In active mode, the input line can be used through level and edge detectors to trigger synchronous interrupts. The edge (rising, falling, or both) or the level used (logical 0, logical 1, or both) can be configured.
- In idle mode, the input line can be used to activate the asynchronous wake-up request (on edge detection: rising edge, falling edge, or both).

The module provides an alternative to the atomic test and set operations for the data-output register (GPIO_DATAOUT). For this register, the module implements the set-and-clear protocol register update (see Section 27.4.9, *General-Purpose Interface Set-and-Clear Protocol*).

All module registers are 8-, 16-, or 32-bit accessible through the OCP-compatible interface (little-endian encoding)

27.2 General-Purpose Interface Environment

The general-purpose interface combines eightGPIO modules for a flexible, user-programmable, general-purpose input/output (I/O) controller. The general-purpose interface implements functions that are not implemented with the dedicated controllers in the device and require simple input and/or output software-controlled signals. The GPIO allows a variety of custom connections and expands the I/O capabilities of the system to the real world.

Figure 27-2 shows a typical application using the general-purpose interface.

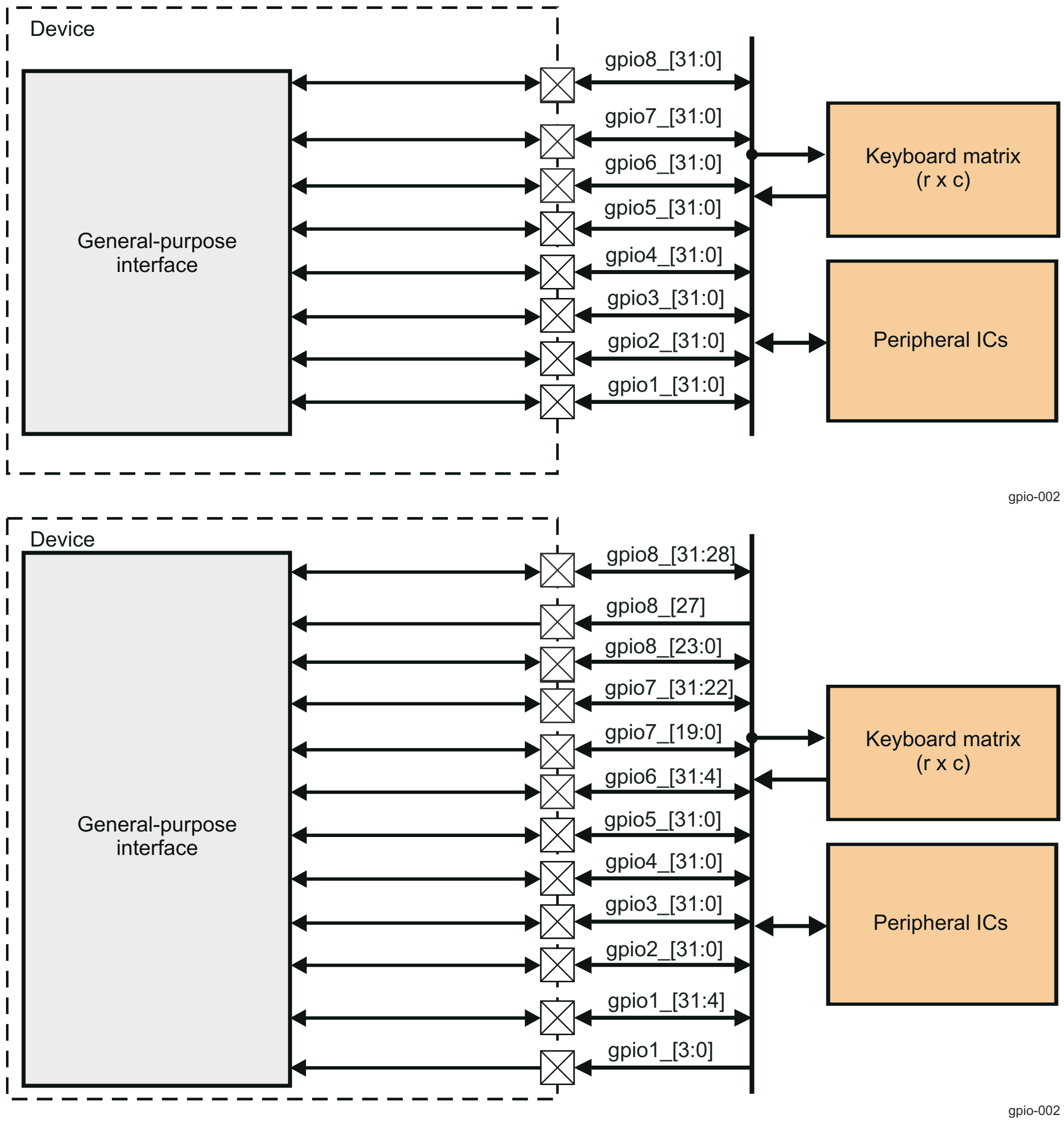


Figure 27-2. General-Purpose Interface Typical Application

The general-purpose interface can physically connect the device to a keyboard matrix and peripheral integrated circuits (ICs).

27.2.1 General-Purpose Interface as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface. Channels can be dedicated based on the keyboard matrix (r × c). Figure 27-3 shows row channels configured as inputs with the input debounce feature enabled. The row channels are driven high with an external pullup. Column channels are configured as outputs and drive a low level.

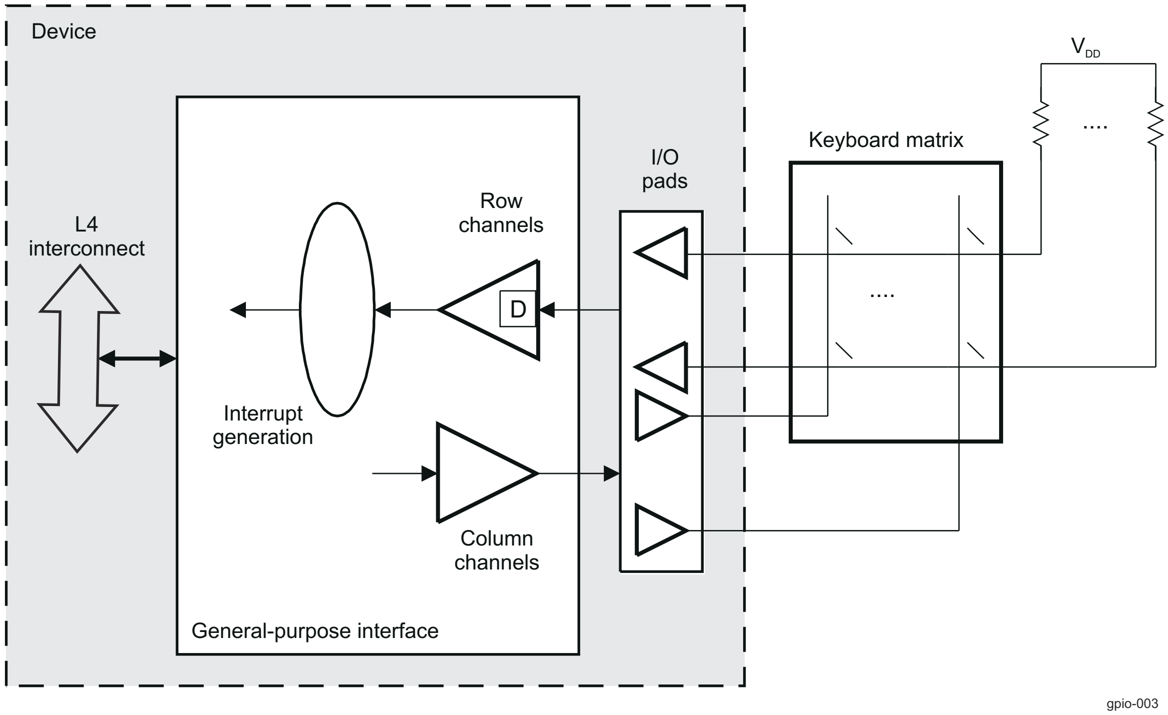


Figure 27-3. General-Purpose Interface Used as a Keyboard Interface

When a keyboard matrix key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row channel. This generates an interrupt based on the proper configuration (see Section 27.4.6, *Interrupt and Wake-up Requests*).

When the keyboard interrupt is received, the processor (microprocessor unit [MPU] and/or digital signal processor [DSP] subsystem) can disable the keyboard interrupt and scan the column channels for the key coordinates.

- The scanning sequence has as many states as column channels: For each step in the sequence, the processor drives one column channel low and the others high.
- The processor reads the values of the row channels and thus detects which keys in the column are pressed.

At the end of the scanning sequence, the processor establishes which keys are pressed. The keyboard interface can then be reconfigured in the interrupt waiting state.

27.2.2 General-Purpose Interface Signals

Table 27-1 describes the module signals.

Table 27-1. I/O Description

Signal	I/O[1]	Description	Reset Value[2]
gpio1_[0:3]	I	GPIO (inputs only)	Hi-Z
gpio1_[4:31]	I/O	GPIO	Hi-Z
gpio2_[0:31]	I/O	GPIO	Hi-Z
gpio3_[0:31]	I/O	GPIO	Hi-Z
gpio4_[0:31]	I/O	GPIO	Hi-Z
gpio5_[0:31]	I/O	GPIO	Hi-Z
gpio6_[4:31]	I/O	GPIO	Hi-Z
gpio7_[0:19]	I/O	GPIO	Hi-Z
gpio7_[22:31]	I/O	GPIO	Hi-Z
gpio8_[0:23]	I/O	GPIO	Hi-Z

Table 27-1. I/O Description (continued)

Signal	I/O[1]	Description	Reset Value[2]
gpio8_27	I	GPIO (input only)	Hi-Z
gpio8_[28:31]	I/O	GPIO	HI-Z

(1) I = Input; O = Output; I/O = Bidirectional
(2) Hi-Z = High Impedance

Figure 27-4 shows the signal connections of GPIO1.

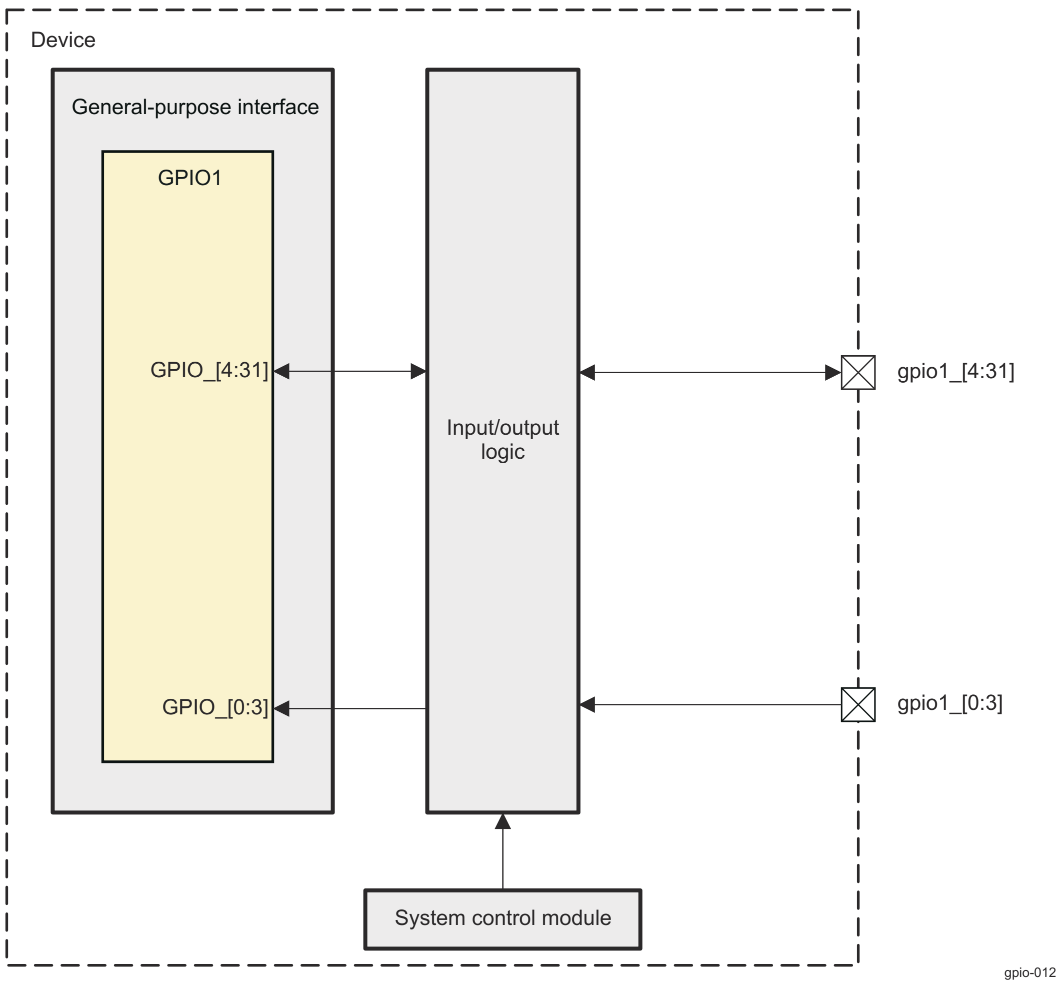


Figure 27-4. GPIO1 Signal Connections

Figure 27-5 shows the signal connections of GPIO2 through GPIO8.

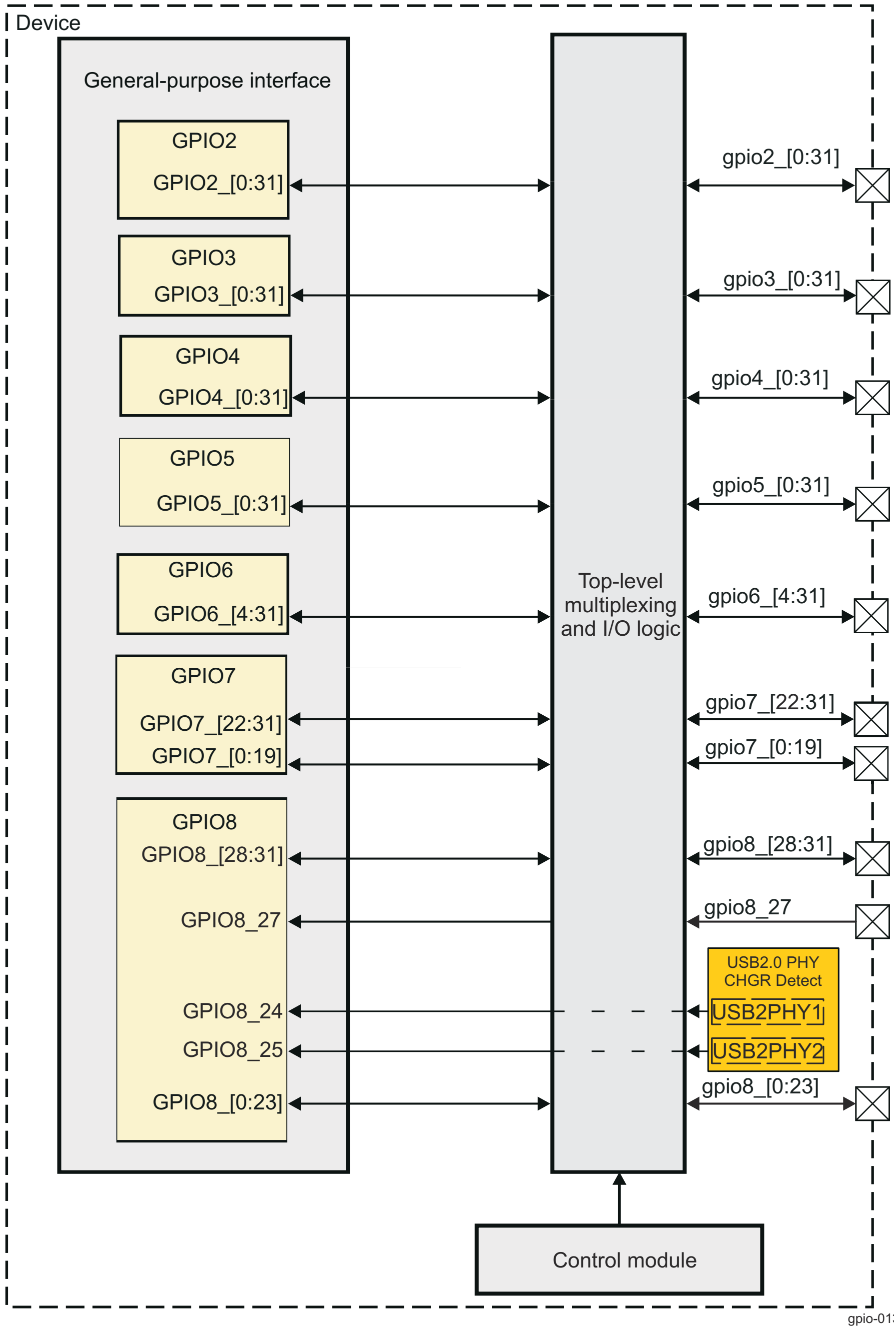


Figure 27-5. GPIO2 Through GPIO8 Signal Connections

Note

For more information about the GPIO1 through GPIO8 signals and channel description, see Section 27.4.7, *General-Purpose Interface Channels Description.*

Note

For more information about GPIO signal multiplexing, see *Pad Configuration Registers, and* in *Control Module*.

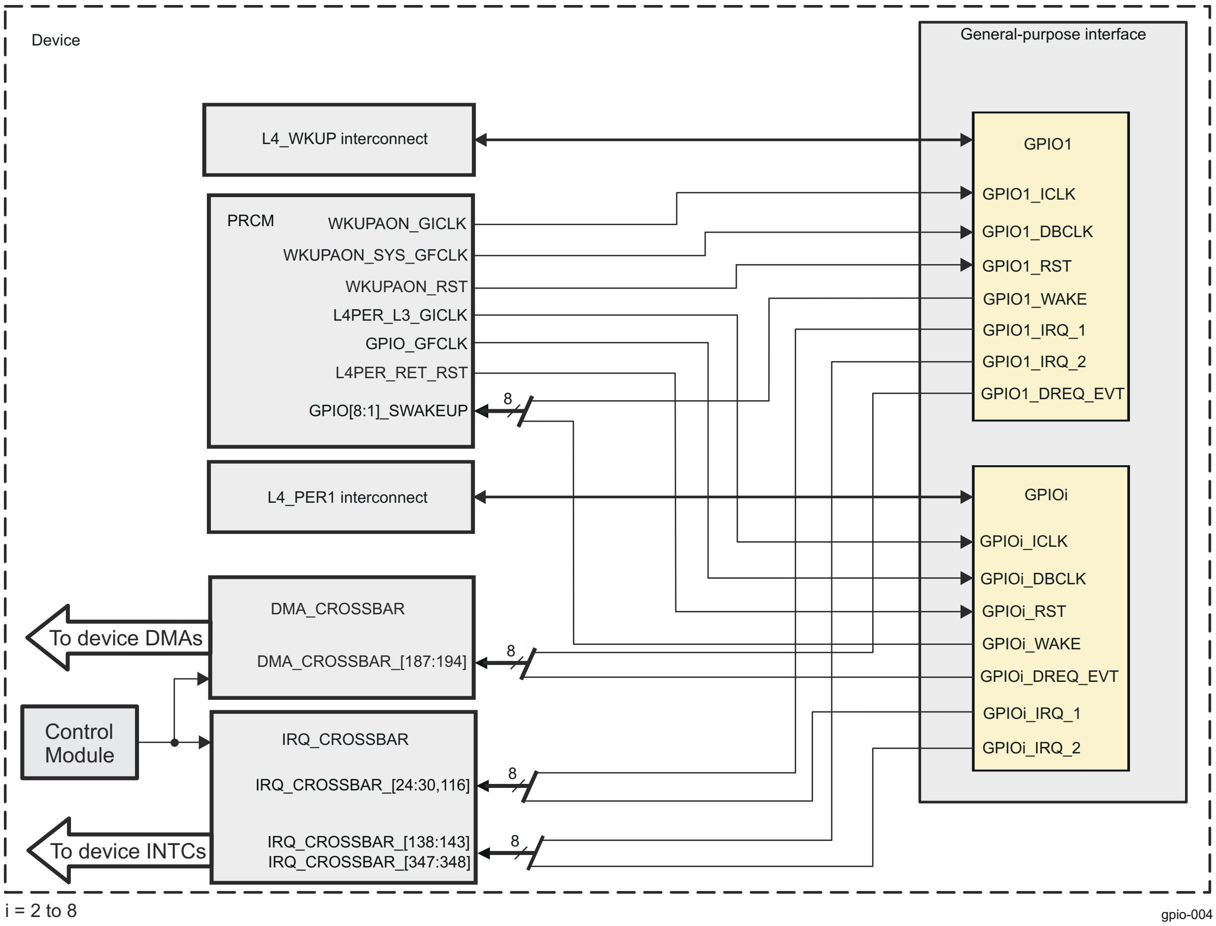
Note

GPIO pins can directly snoop device pads even if those are configured in the other modes. For GPIO output user needs to set pad mux.

27.3 General-Purpose Interface Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 27-6 shows this module integration.



i = 2 to 8
IRQ_CROSSBAR_[24:30] - GPIO1 to GPIO7 interrupt line 1
IRQ_CROSSBAR_116 - GPIO8 interrupt line 1
IRQ_CROSSBAR_[138:143] - GPIO1 to GPIO6 interrupt line 2
IRA_CROSSBAR_[347:348] - GPIO7 and GPIO8 interrupt line 2

Figure 27-6. GPIO Integration

Note

For more information about the slave idle protocol and the wake-up request, see *Clock Management*, in *Power, Reset, and Clock Management*.

Table 27-2 through Table 27-4 summarize the integration of the module in the device.

Table 27-2. GPIO Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
GPIO1	PD_WKUPAON	L4_WKUP
GPIOi (where i = 2 to 8)	PD_COREAON	L4_PER1

Table 27-3. GPIO Clocks and Resets

Clocks

Table 27-3. GPIO Clocks and Resets (continued)

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
GPIO1	GPIO1_ICLK	WKUPAON_GICLK	PRCM	GPIO interface clock
	GPIO1_DBCLK	WKUPAON_SYS_GFCLK	PRCM	GPIO functional clock
GPIO2	GPIO2_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO2_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO3	GPIO3_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO3_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO4	GPIO4_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO4_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO5	GPIO5_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO5_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO6	GPIO6_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO6_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO7	GPIO7_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO7_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO8	GPIO8_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO8_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
		Resets		
GPIO1	GPIO1_RST	WKUPAON_RST	PRCM	GPIO reset signal
GPIO2	GPIO2_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO3	GPIO3_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO4	GPIO4_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO5	GPIO5_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO6	GPIO6_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO7	GPIO7_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO8	GPIO8_RST	L4PER_RET_RST	PRCM	GPIO reset signal

Table 27-4. GPIO Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
GPIO1	GPIO1_IRQ_2	IRQ_CROSSBAR_138	N/A	GPIO1 interrupt request (secont interrupt line). This IRQ source signal is not mapped by default to any device INTC.
	GPIO1_IRQ_1	IRQ_CROSSBAR_24	MPU_IRQ_29	GPIO1 interrupt request to MPU (first interrupt line)
			DSP1_IRQ_55	GPIO1 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_55	GPIO1 interrupt request to DSP2 (first interrupt line)
			IPU1_IRQ_51	GPIO1 interrupt request to IPU1 (first interrupt line)
			IPU2_IRQ_51	GPIO1 interrupt request to IPU2 (first interrupt line)
GPIO2	GPIO2_IRQ_2	IRQ_CROSSBAR_139	N/A	GPIO2 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO2_IRQ_1	IRQ_CROSSBAR_25	MPU_IRQ_30	GPIO2 interrupt request to MPU (first interrupt line)
			DSP1_IRQ_56	GPIO2 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_56	GPIO2 interrupt request to DSP2 (first interrupt line)
			IPU1_IRQ_52	GPIO2 interrupt request to IPU1 (first interrupt line)
			IPU2_IRQ_52	GPIO2 interrupt request to IPU2 (first interrupt line)
GPIO3	GPIO3_IRQ_2	IRQ_CROSSBAR_140	N/A	GPIO3 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO3_IRQ_1	IRQ_CROSSBAR_26	MPU_IRQ_31	GPIO3 interrupt request to MPU (first interrupt line)
			DSP1_IRQ_57	GPIO3 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_57	GPIO3 interrupt request to DSP2 (first interrupt line)
GPIO4	GPIO4_IRQ_2	IRQ_CROSSBAR_141	N/A	GPIO4 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO4_IRQ_1	IRQ_CROSSBAR_27	MPU_IRQ_32	GPIO4 interrupt request to MPU (first interrupt line)
			DSP1_IRQ_58	GPIO4 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_58	GPIO4 interrupt request to DSP2 (first interrupt line)

Table 27-4. GPIO Hardware Requests (continued)

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
GPIO5	GPIO5_IRQ_2	IRQ_CROSSBAR_142	N/A	GPIO5 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO5_IRQ_1	IRQ_CROSSBAR_28	MPU_IRQ_33	GPIO5 interrupt request to MPU (first interrupt line)
			DSP1_IRQ_59	GPIO5 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_59	GPIO5 interrupt request to DSP2 (first interrupt line)
GPIO6	GPIO6_IRQ_2	IRQ_CROSSBAR_143	N/A	GPIO6 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO6_IRQ_1	IRQ_CROSSBAR_29	MPU_IRQ_34	GPIO6 interrupt request to MPU (first interrupt line)
			DSP1_IRQ_60	GPIO6 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_60	GPIO6 interrupt request to DSP2 (first interrupt line)
GPIO7	GPIO7_IRQ_2	IRQ_CROSSBAR_347	N/A	GPIO7 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO7_IRQ_1	IRQ_CROSSBAR_30	MPU_IRQ_35	GPIO7 interrupt request to MPU (first interrupt line)
			DSP1_IRQ_61	GPIO7 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_61	GPIO7 interrupt request to DSP2 (first interrupt line)
GPIO8	GPIO8_IRQ_2	IRQ_CROSSBAR_348	N/A	GPIO8 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO8_IRQ_1	IRQ_CROSSBAR_116	MPU_IRQ_121	GPIO8 interrupt request to MPU (first interrupt line)
		DMA Requests		
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
GPIO1	GPIO1_DREQ_EVT	DMA_CROSSBAR_187	N/A	GPIO1 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO2	GPIO2_DREQ_EVT	DMA_CROSSBAR_188	N/A	GPIO2 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO3	GPIO3_DREQ_EVT	DMA_CROSSBAR_189	N/A	GPIO3 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO4	GPIO4_DREQ_EVT	DMA_CROSSBAR_190	N/A	GPIO4 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.

Table 27-4. GPIO Hardware Requests (continued)

GPIO5	GPIO5_DREQ_EVT	DMA_CROSSBAR_191	N/A	GPIO5 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO6	GPIO6_DREQ_EVT	DMA_CROSSBAR_192	N/A	GPIO6 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO7	GPIO7_DREQ_EVT	DMA_CROSSBAR_193	N/A	GPIO7 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO8	GPIO8_DREQ_EVT	DMA_CROSSBAR_194	N/A	GPIO8 module - event/ interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.

Note

The “**Default Mapping**” column in Table 27-4 *GPIO Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For the description of the interrupt source, see Section 27.4.6, *Interrupt and Wake-Up Requests.*

27.4 General-Purpose Interface Functional Description

27.4.1 General-Purpose Interface Block Diagram

Figure 27-7 shows the general-purpose interface block diagram.

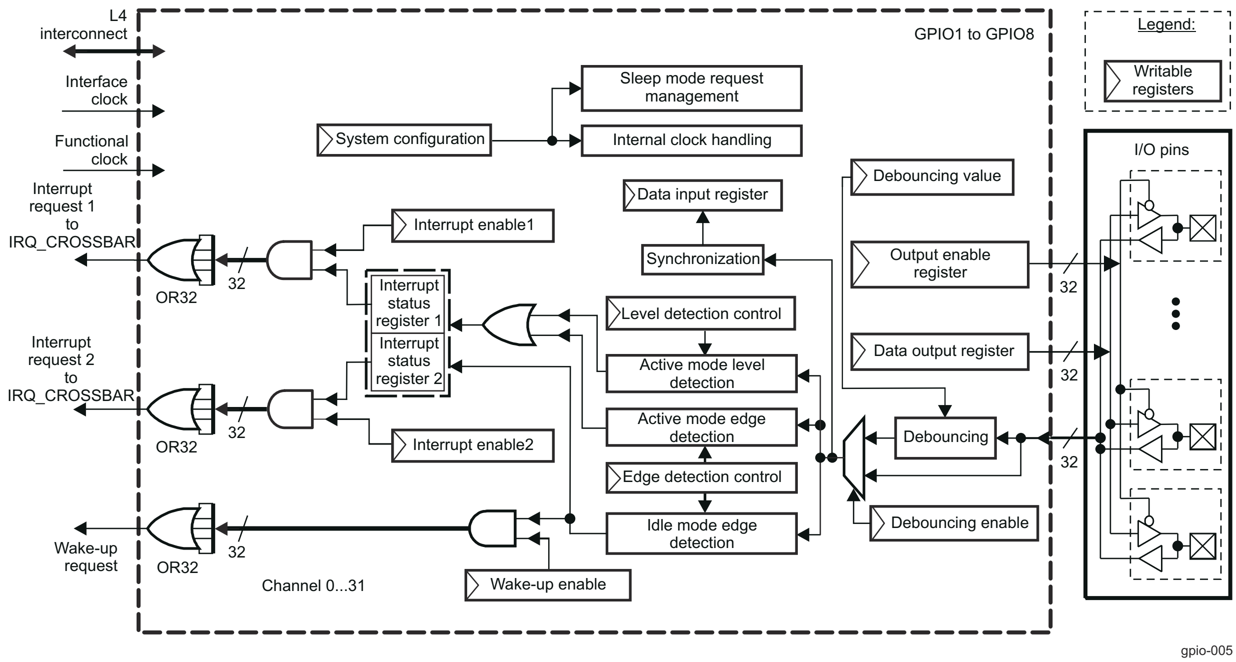


Figure 27-7. General-Purpose Interface Block Diagram

Figure 27-7 shows the details of the GPIO modules in the general-purpose interface block diagram, including their configuration registers and main functional paths:

- The synchronous path (for active mode operation) used to generate a synchronous interrupt request on expected event detection on any input GPIO. Synchronous interrupt request lines 1 and 2 are active based on their respective interrupt-enable1 and 2 registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_1, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1). See Figure 27-8.

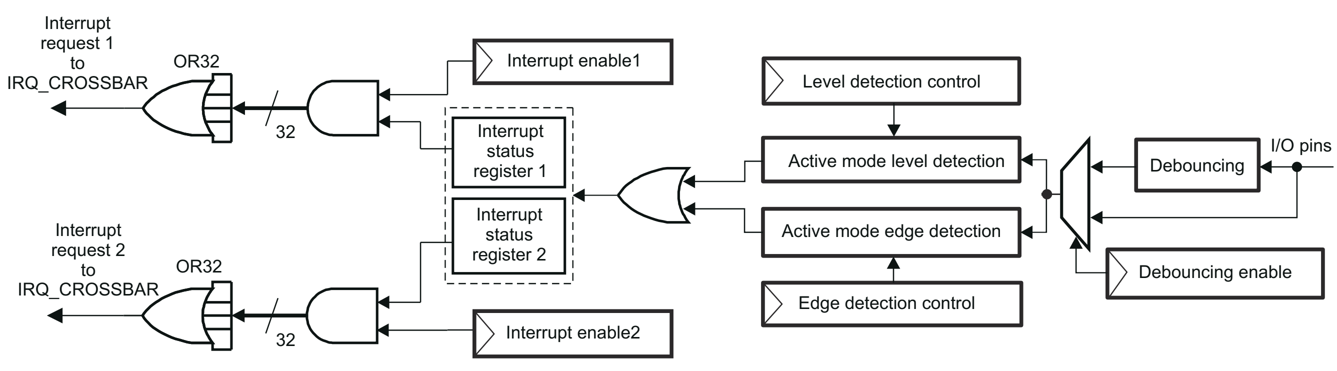


Figure 27-8. Synchronous Path

- The asynchronous path (for idle mode operation) used to generate an asynchronous wake-up request on the expected edge detection on any input GPIO. The asynchronous wake-up request line is active based on the wake-up-enable register. See Figure 27-9.

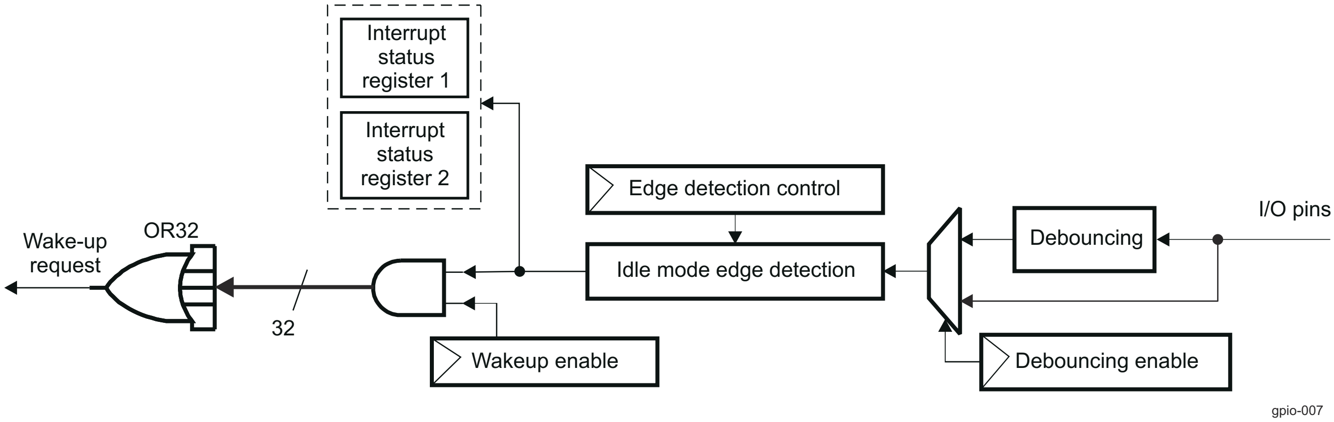


Figure 27-9. Asynchronous Path

- The blocks handling the internal clock (clock gating) and managing the sleep mode request/acknowledge protocol (enabling the synchronous path in active mode and the asynchronous path in idle mode)

27.4.2 General-Purpose Interface Interrupt and Wake-Up Features

27.4.2.1 Synchronous Path: Interrupt Request Generation

The general-purpose interface has 16 interrupt lines (two interrupt lines per GPIO module instance). The 16 interrupt signals are GPIOi_IRQ_1 (used by the MPU, DSP and IPU subsystems) and GPIOi_IRQ_2 (used by the CROSSBAR), where i = 1 to 8.

Synchronous interrupt requests from each channel are processed by two identical interrupt generation submodules used independently by the CROSSBAR subsystem on one side and by the MPU, IPU, and DSP subsystems on the other side. Each submodule controls its own synchronous interrupt request line and has its own interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_1, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1) and interrupt status registers (GPIOi.GPIO_IRQSTATUS_RAW_0 and GPIOi.GPIO_IRQSTATUS_RAW_1). The interrupt-enable register selects the channel(s) considered for the interrupt request generation, and the interrupt status register determines which channel(s) activate the interrupt request. Event detection on GPIO channels is reflected in the interrupt status registers independent of the content of the interrupt-enable registers.

In active mode, when the GPIO configuration registers are set to enable the interrupt generation (see Section 27.4.6, *General-Purpose Interface Interrupt and Wake-Up Requests*), a synchronous path samples the transitions and levels on the input GPIO with the internally gated interface clock (see Section 27.4.5.2.4, *Module Power Saving*). When an event matches the programmed settings (see Section 27.4.6, *General-Purpose Interface Interrupt and Wake-Up Requests*), the corresponding bit in the interrupt status register (GPIO_IRQSTATUS_RAW_x [where x = 0 or 1]) is set to 1, and on the following interface clock cycle, interrupt lines 1 and/or 2 are activated (depending on the interrupt-enable registers GPIO_IRQSTATUS_SET_x [where x = 0 or 1]).

Because of the sampling operation, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the internally gated interface clock period (that is, *N* times the interface clock period; see Section 27.4.5.2.4, *Module Power Saving*). This minimum pulse width must be met before and after any expected level transition detection. Level detection requires the selected level to be stable for at least two times the internally gated interface clock period to trigger a synchronous interrupt.

Because the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This latency must not exceed three internally gated interface clock cycles plus two interface clock cycles when the debounce feature is not used.

When the debounce feature is active, the latency depends on the value of the debouncing time register (GPIOi.GPIO_DEBOUNCINGTIME) (see Section 27.4.3, *General-Purpose Interface Clock Configuration*) and is less than three internally gated interface clock cycles plus two interface clock cycles plus GPIOi.GPIO_DEBOUNCINGTIME register value debounce clock cycles plus three debounce clock cycles.

Synchronous interrupt request line 1 is default mapped on the MPU, IPU, DSP subsystems.

Synchronous interrupt request line 2 is mapped on the CROSSBAR.

Figure 27-10 is an overview of the interrupt request generation.

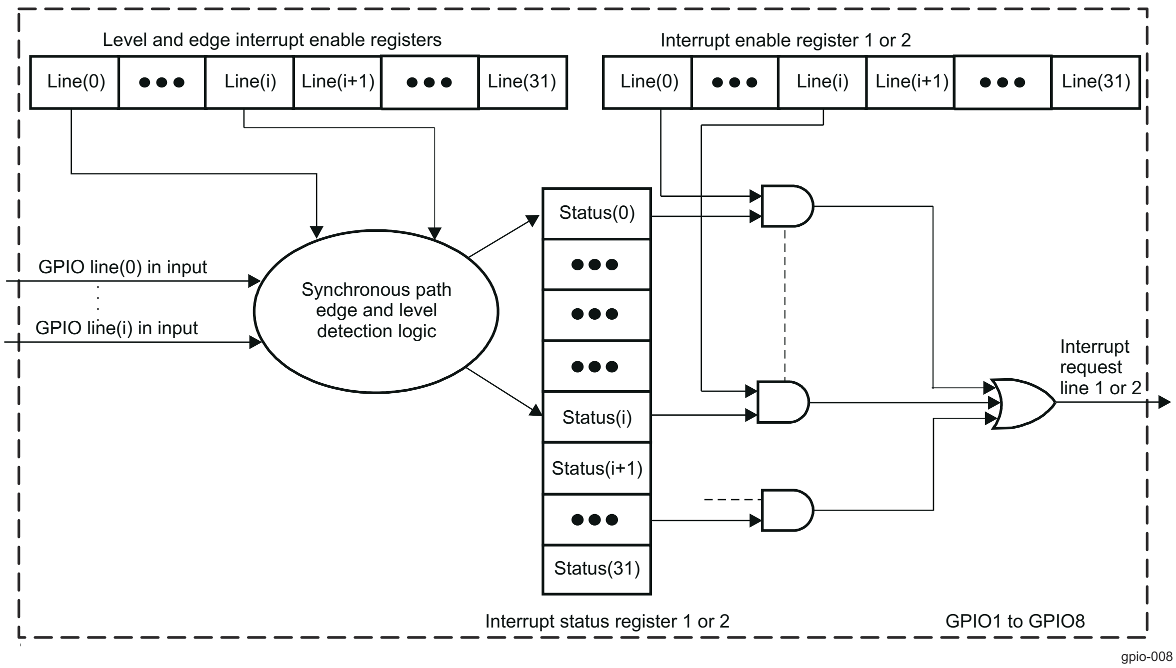


Figure 27-10. Interrupt Request Generation

27.4.2.2 Asynchronous Path: Wake-Up Request Generation

The general-purpose interface has eight wake-up lines (one wake-up line per GPIO module instance) connected to the PRCM module.

Asynchronous wake-up requests from input channels are merged to issue one wake-up signal to the system per GPIO module. The wake-up-enable registers (GPIOi.GPIO_IRQWAKEN_0 and GPIOi.GPIO_IRQWAKEN_1) select the channel(s) considered for the wake-up request generation. The asynchronous wake-up request is reflected into the synchronous interrupt status registers (GPIOi.GPIO_IRQSTATUS_RAW_0 and GPIOi.GPIO_IRQSTATUS_RAW_1).

In idle mode (the interface clock is shut down and the GPIO configuration registers are programmed; see Section 27.4.6, *General-Purpose Interface Interrupt and Wake-Up Requests*), an asynchronous path detects the expected transition(s) on a GPIO input (based on register programming) and activates an asynchronous wake-up request by the sideband signal (GPIOi_SWAKEUP [where i = 1 to 8), if the wake-up-enable register is set.

When the system is awakened, the interface clock is restarted and synchronously set to 1 based on the input GPIO pin triggering the wake-up request and the corresponding bit in the interrupt status registers (GPIOi.GPIO_IRQSTATUS_RAW_0 and GPIOi.GPIO_IRQSTATUS_RAW). On the following internal clock cycle, interrupt lines 1 and/or 2 are active (active high) when the corresponding bits are set in the interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1).

Note

- If the debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request, because there is no sampling operation.
- If the debouncing is used, the minimum pulse width is set by the debouncing specified time.
- The ENAWAKEUP bit of the GPIO_SYSCONFIG register allows the enabling or disabling of the GPIO wake-up feature globally: if this bit is set to 0, GPIO_IRQWAKEN_x has no effect

Figure 27-11 is an overview of the wake-up request generation.

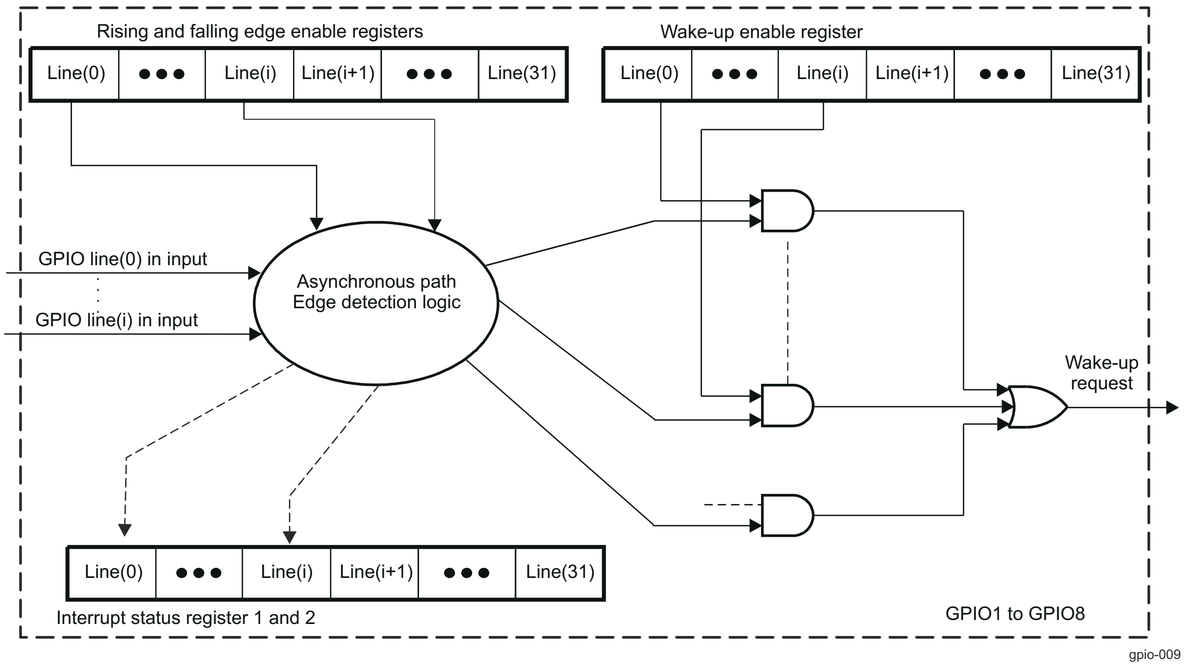


Figure 27-11. Wake-Up Request Generation

27.4.2.3 Wake-Up Event Conditions During Transition To/From IDLE State

In phase A, only the synchronous path is enabled. A synchronous interrupt request (see Section 27.4.2.1, *Synchronous Path: Interrupt Request Generation*) activates the interrupt line(s) and prevents the GPIO from transitioning into IDLE state until the interrupt is cleared.

In phase B, the asynchronous path and synchronous path are enabled during the first five functional clock cycles of SLEEPTRANS state. During this period a synchronous interrupt request (see Section 27.4.2.1, *Synchronous Path: Interrupt Request Generation*) prevents the GPIO from transitioning into IDLE state. A shorter pulse puts the module into IDLE state but triggers a wakeup once in IDLE.

In phase C, only the asynchronous path is enabled. A wake-up request (see Section 27.4.2.2, *Asynchronous Path: Wake-Up Request Generation*) triggers a wake-up request from the GPIO and when the module is awakened an interrupt is generated. If debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request.

In phase D, eight open-core protocol (OCP) clock cycles occur until the module is in FUNCT state, the synchronous path is enabled, and an event that fulfills the pulse width requirements (see Section 27.4.2.1, *Synchronous Path: Interrupt Request Generation*) activates the interrupt line(s).

In phase E, only the synchronous path is enabled. A synchronous interrupt request (see Section 27.4.2.1, *Synchronous Path: Interrupt Request Generation*) activates the interrupt line(s).

Figure 27-12 shows the wake-up event conditions.

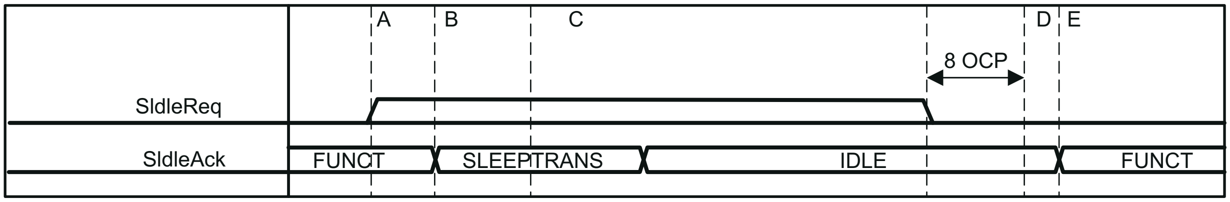


Figure 27-12. Wake-Up Event Conditions

27.4.2.4 Interrupt (or Wake-Up) Line Release

When the host processor (the MPU and/or DSP subsystem in the device) receives an interrupt request issued by the GPIO module, it reads the corresponding interrupt status register (GPIOi.GPIO_IRQSTATUS_0 or GPIOi.GPIO_IRQSTATUS_1) to determine which GPIO input triggered the interrupt (or the wake-up request).

After servicing the interrupt (or acknowledging the wake-up request), the software resets the status bit and releases the interrupt line by setting the corresponding bit of the interrupt status register to 1. If there is still a pending interrupt request to serve (all bits in the interrupt status register that are not masked by the interrupt-enable register are not cleared), the interrupt line is reasserted.

Note

The status bit must be reset to re-enter idle mode.

27.4.3 General-Purpose Interface Clock Configuration

27.4.3.1 Clocking

Each GPIO module uses two clocks:

- Debounce clock: The 32-kHz debounce clock, GPIOi_DBCLK (where i = 1 to 8 with one debounce clock per module), comes from the PRCM module and is used to debounce the submodule logic (without the corresponding configuration registers). This module can sample the input line and filter the input level using a programmed delay.

  The debouncing value register (GPIOi.GPIO_DEBOUNCINGTIME) is used to set the debouncing time for all input lines in the GPIO module. Because the value is global for all the ports of one GPIO module, up to eight different debouncing values are possible. The debounce cell runs with the debounce clock (32 kHz). This register represents the number of clock cycle(s) to be used.

  The following formula describes the required input stable time to be propagated to the debounced output:

  Required input line stable = (the value of the GPIOi.GPIO_DEBOUNCINGTIME[7:0] DEBOUNCETIME bit field + 1) × 31,

  where the value of the GPIOi.GPIO_DEBOUNCINGTIME[7:0] DEBOUNCETIME bit field is from 0 to 255.

  For more information, see *CD_L4_PER1 Clock Domain*, and *CD_WKUPAON Clock Domain*, in *Power, Reset, and Clock Management*.
- Interface clock: The interface clock, GPIOi_ICLK (where i = 1 to 8), comes from the PRCM module and is used throughout the GPIO module (except within the debounce cell logic). GPIOi_ICLK clocks the data exchanges between the L4 interconnect and the internal logic. The clock-gating features allow module power consumption to be adapted to the activity.

  For more information, see *CD_L4_PER1_Clock Domain*, and *CD_WKUPAON Clock Domain*, in *Power, Reset, and Clock Management*.

Table 27-3 describes the clocks in the GPIO modules.

Table 27-5 summarizes the functional clock configuration.

Table 27-5. Functional Clock Configuration

Interface Clock	GPIO_CTRL[2:1]GATINGRATIO	Functional Clock
GPIOi_ICLK (where i = 1 to 8)	00	GPIOi_ICLK /1
GPIOi_ICLK (where i = 1 to 8)	01	GPIOi_ICLK /2
GPIOi_ICLK (where i = 1 to 8)	10	GPIOi_ICLK /4
GPIOi_ICLK (where i = 1 to 8)	11	GPIOi_ICLK /8

27.4.4 General-Purpose Interface Hardware and Software Reset

The GPIO can be reset by using the domain reset (hardware reset) or by setting a dedicated configuration bit (software reset) in each GPIO module.

- Hardware reset: The GPIO2 to GPIO8 modules are attached to the L4PER_RET_RST reset domain. GPIO1 is attached to the WKUPAON_RST reset domain.

  The hardware reset has a global reset action on the GPIO modules of the general-purpose interface. All configuration registers and internal logic are reset when it is active (low level). In each GPIO module, the GPIOi.GPIO_SYSSTATUS[0] RESETDONE bit monitors the internal reset status; it is set when the reset completes. For more information, see *Reset Domains*, in *Power, Reset, and Clock Management*.
- Software reset: Each GPIO module has its own software reset using the GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit. The software reset has the same effect as the hardware reset signal, but this reset can be applied on one or more modules.

Setting the GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit to 1 resets the module. A bit value of 1 remains until the reset completes. When the software reset completes, the GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit is automatically reset to 0 and has the same effect as a hardware reset. The GPIOi.GPIO_SYSSTATUS[0] RESETDONE bit is cleared during a software reset. RESETDONE is set to 1 when the software reset completes.

27.4.5 General-Purpose Interface Power Management

27.4.5.1 Power Domain

GPIO1 is attached to the PD_WKUPAON power domain (see *Power Management*, in *Power, Reset, and Clock Management*). This domain is composed of the logic permanently supplied to manage domain power state transitions and detect wake-up events. The WKUPAON power domain is continuously active. The GPIO2 to GPIO8 modules are attached to the PD_L4PER power domain (see *Power Management*, in *Power, Reset, and Clock Management*). The PD_L4PER power domain is not active continuously.

27.4.5.2 Power Management

27.4.5.2.1 Idle Scheme

To reduce dynamic consumption, an efficient idle scheme is based on the following:

- Efficient local autoclock gating for each module
- The implementation of control sideband signals between the PRCM module and each module

This enhanced idle control allows clocks to be activated and deactivated safely without requiring complex software management.

The idle mode request, idle acknowledge, and wake-up request (GPIOi_WAKEUP) are sideband signals between the PRCM module and the general-purpose interface (see Section 27.4.6.2, *Wake-Up Requests Generation*).

27.4.5.2.2 Operating Modes

Three operating modes are defined for the module:

- Active mode: The module runs synchronously on the interface clock; interrupts can be generated based on the configuration and external signals.
- Idle mode: Power-saving mode with the module in a waiting state. The interface clock can be stopped, an interrupt cannot be generated, and a wake-up signal can be generated based on the configuration and external signals.

  If the debounce clock provided by the PRCM module is active, the debounce cell can sample and filter the input to generate a wake-up event. If the debounce clock is inactive, the debounce cell gates all input signals and thus cannot be used.
- Disabled mode: The module is not used. The internal clock paths are gated, and an interrupt or wake-up request cannot be generated.

Idle mode is configured within the module and activated on request by the host processor through sideband signals (see Section 27.4.5.2.3, *System Power Management and Wakeup*).

The disabled mode is set by software through a dedicated configuration bit, GPIOi.GPIO_CTRL[0] DISABLEMODULE (0: The module is enabled and clocks are not gated; 1: The module is disabled and clocks are gated). It unconditionally gates the internal clock paths that are not used for the system interface. When setting the GPIO_CTRL[0] DISABLEMODULE bit (enabling or disabling the GPIO module), it is important to switch the debouncing clock on or off in the following order:

- The GPIO optional debouncing clock must be enabled before the GPIO module is set (GPIO_CTRL[0]DISABLEMODULE = 0x0).
- The GPIO optional debouncing clock must be disabled after the GPIO module is disabled (GPIO_CTRL[0]DISABLEMODULE = 0x1).

27.4.5.2.3 System Power Management and Wakeup

The PRCM module can require the GPIO modules to be idled for power-saving purposes.

The general-purpose interface has eight identical idle mode request/acknowledge (handshake) mechanisms with the PRCM module (see Section 27.4.6.2, *Wake-Up Requests Generation*): one per GPIO module. The general-purpose interface allows the GPIO modules to enter idle mode based on the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field.

Idle acknowledge depends on the configuration and activity of each GPIO module:

- Smart-idle mode

  When the GPIO module is configured in smart-idle mode, it checks for more activity (capture of the input GPIO pins in the GPIOi.GPIO_DATAIN register is complete with no pending interrupt; all interrupt status bits are cleared), and there is no write access to the GPIO.GPIO_DEBOUNCINGTIME register, which is waiting to be synchronized.

  Idle acknowledge is then asserted and the module enters into idle mode. It waits for active system clock gating by the PRCM module (when all peripherals supplied by the same L4 interface clock domain are also ready for idle).

  In idle mode (that is, when the PRCM module gates the interface clock), no interrupt occurs.
- Smart-idle wake-up mode

  If the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field selects smart-idle or smart-idle wake-up mode, the GPIO module evaluates its internal capability to have the interface clock switched off. Once all internal activity ceases (the DATA INPUT REGISTER completed to capture the input GPIO pins, there is no pending interrupt, all interrupt status bits are cleared, and there is no write access to the GPIO_DEBOUNCINGTIME register pending to be synchronized), the idle acknowledge is asserted and the GPIO enters into Idle mode, ready to issue a wake-up request when the expected transition occurs on an enabled GPIO input pin. This wake-up request is effectively sent only if the GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP bit of the system configuration register enables the GPIO wake-up capability (see GPIO_SYSCONFIG). When the system is awake, the IDLE request goes inactive, the idle acknowledge and wake-up request (if the GPIO triggered the wake-up in the system) signals are immediately deasserted, and the asynchronous wake-up request (if it exists) is reflected into the synchronous interrupt status registers.
- Force-idle mode

  When the GPIO module is configured in force-idle mode (the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field = 0x0) and receives an IDLE request from the PRCM module, the GPIO module waits unconditionally for active system clock gating by the PRCM module. (This occurs only when all peripherals supplied by the same L4 interface clock domain are also ready for idle.)

  When in idle mode (that is, when the PRCM module gates the interface clock), the module (in inactive mode) has no activity, the interface clock paths are gated, an interrupt cannot be generated, and the wake-up feature is totally inhibited.
- No-idle mode

  When the GPIO module is configured in no-idle mode (the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field = 0x1) and receives an IDLE request from the PRCM module, the GPIO module does not go into idle mode and the idle acknowledge is never sent.

Note

For more information about the idle modes, see *Power Management*, in *Power, Reset, and Clock Management*.

27.4.5.2.4 Module Power Saving

The GPIO module has local power management by internal clock-gating features:

- Internal interface clock gating: The clock for the system interface logic can be gated when the module is not accessed, if the GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE bit is set. Otherwise, this logic is free-running on the interface clock.
- Clock gating for the input data sample logic: Clock for the input data sample logic can be gated when the data in the GPIOi.GPIO_DATAIN register is not accessed.
- Clock gating for the event detection logic: Each GPIO module implements four clock groups used for the logic in the synchronous event detection. Each group of eight input GPIO pins has a separate enable signal depending on the edge/level detection register setting (because the input is 32 bits, four groups of eight inputs are defined for each GPIO module). If a group requires no detection, the corresponding clock is gated off.

  All channels are also gated using a one-out-of-N scheme. N is the GPIOi.GPIO_CTRL[2:1] GATINGRATIO bit field and can take the values 1, 2, 4, and 8. The interface clock is enabled for this logic one cycle every N cycles. When N is 1, there is no gating and this logic is free-running on the interface clock. When N is 2, 4, or 8, this logic runs at the equivalent frequency of interface clock frequency divided by N.
- Inactive mode: In inactive mode, all internal clock paths are gated.
- Disabled mode: All internal clock paths not used for the L4 interconnect are gated. The GPIOi.GPIO_CTRL[0] DISABLEMODULE bit controls a clock-gating feature at the module level. Setting this bit to 1 forces clock gating for all internal clock paths. Module internal activity is suspended. The L4 interconnect is not affected by this bit.

The interface clock gating is controlled with the GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE bit, which is used to save power when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits.

Table 27-6 describes the power-management features available for the general-purpose interface module.

Note

For information about source clock gating and sleep/wake-up transitions, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

For descriptions of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see *Module-Level Clock Management*, in *Power, Reset, and Clock Management*.

Table 27-6. Local Power-Management Features

Feature	Register Bits/Bit Fields	Description
Clock autogating	GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE	It sets the clock-gating strategy for the OCP interface block.
Slave-idle modes	GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE	Force-idle, no-idle, and smart-idle wake-up-capable modes are available.
Clock activity	GPIOi.GPIO_CTRL[0] DISABLEMODULE	Enable and disable the module.
Debouncing enable	GPIOi.GPIO_DEBOUNCENABLE[31:0] DEBOUNCEENABLE	Debouncing mode is available.
Global wake-up enable	GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP	This bit enables the wake-up feature at the module level.
Wake-up sources enable	GPIOi.GPIO_IRQWAKEN_0[31:0] INTLINE GPIOi.GPIO_IRQWAKEN_1[31:0] INTLINE	This register enables or disables a specific IRQ request source to generate a wake-up signal.

Table 27-7 describes the clock activity settings.

Table 27-7. Clock Activity Settings

GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE	Selected Mode	Description	Wake-Up Events
00	Force-idle	The GPIO module goes into inactive mode independently of the internal module state, and the IDLE acknowledge is never sent.	No
01	No-idle	The GPIO module does not go into Idle mode and the IDLE acknowledge is never sent.	No

Table 27-7. Clock Activity Settings (continued)

GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE	Selected Mode	Description	Wake-Up Events
10	Smart-idle	The GPIO module evaluates its internal capability to have the interface clock switched off. If there is no more internal activity, the IDLE acknowledge is asserted and the GPIO enters idle mode.	No
11	Smart-idle wake-up	The GPIO module evaluates its internal capability to have the interface clock switched off. If there is no more internal activity, the IDLE acknowledge is asserted and the GPIO enters idle mode.	Yes

27.4.6 General-Purpose Interface Interrupt and Wake-Up Requests

27.4.6.1 Interrupt Requests Generation

All interrupt sources (the 32 GPIO input channels) are merged to issue two synchronous interrupt requests in each GPIO module. Thus, the general-purpose interface has 16 interrupt lines (two interrupt lines per GPIO module instance).

- Synchronous interrupt request line 1 is default mapped on the MPU,DSP and IPU INTC subsystem.
- Synchronous interrupt request line 2 is mapped on the CROSSBAR.

Table 27-8 lists the event flags, and their mask, that can cause module interrupts.

Table 27-8. Events

Event Flag	Event Mask	Synchronous	Sensitivity	Description
GPIOi.GPIO_IRQSTATUS_0[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_0[31:0] INTLINE	Yes	Edge/level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_1[31:0] INTLINE	Yes	Edge/level	Corresponding to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_0[31:0] INTLINE	Yes	Edge/level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_1[31:0] INTLINE	Yes	Edge/level	Corresponding to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_CLR_0[31:0] INTLINE	Yes	Edge/level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_CLR_1[31:0] INTLINE	Yes	Edge/level	Corresponding to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0[31:0] INTLINE	GPIOi.GPIO_IRQWAKEN_0[31:0] INTLINE	No	Edge/level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1[31:0] INTLINE	GPIOi.GPIO_IRQWAKEN_1[31:0] INTLINE	No	Edge/level	Corresponding to the second line of interrupt

Note

For more information about interrupt mapping, see Table 27-4, *GPIO Hardware Request*.

Synchronous interrupt request line 1 and line 2 are active depending on their respective interrupt-enable1 and interrupt-enable 2 registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_1, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1).

- interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_1)

  The interrupt enable1 (or interrupt enable2) register allows masking of the expected transition on input GPIO to prevent the generation of an interrupt request on line 1 (or line 2). The interrupt-enable registers are programmed synchronously with the interface clock.

These registers can be accessed with direct read/write operations or by using the alternate set-and-clear protocol feature for register update. This feature allows setting or clearing explicit bits of these registers with a single write access (see Section 27.4.9, *General-Purpose Interface Set-and-Clear Protocol*).

- Interrupt status registers (GPIOi.GPIO_IRQSTATUS_0 and GPIOi.GPIO_IRQSTATUS_1)

  The interrupt status 1 (or interrupt status 2) register determines which of the input GPIO pins triggered the interrupt line1 (or interrupt line 2) request (or the wake-up line).

  When a bit in this register is set to 1, it indicates that the corresponding GPIO pin is requesting the interrupt (or the wakeup). To reset a bit in this register, set the appropriate bit to 1. However, an interrupt cannot be generated by writing 1 to the interrupt status 1 (or interrupt status 2) register.

  If 0 is written to a bit in this register, the value in the corresponding bit in the interrupt status 1 and remains unchanged. The interrupt status 1 (or interrupt status 2) register is synchronous with the interface clock. In idle mode, the event is detected through an asynchronous path, and interrupt status 2 registers are set when the GPIO module is awaken.

CAUTION

After servicing the interrupt, the status bit in the interrupt status register (GPIOi.GPIO_IRQSTATUS_0 or GPIOi.GPIO_IRQSTATUS_1) must be reset and the interrupt line released (by setting the corresponding bit of the interrupt status register to 1).

Before enabling an interrupt for the GPIO channel in the interrupt-enable register (GPIOi.GPIO_IRQSTATUS_SET_0 or GPIOi.GPIO_IRQSTATUS_SET_1) the corresponding status bit in the interrupt status register (GPIOi.GPIO_IRQSTATUS_0 or GPIOi.GPIO_IRQSTATUS_1) must be reset to prevent the occurrence of unexpected interrupts when enabling an interrupt for the GPIO channel.

27.4.6.2 Wake-Up Requests Generation

The GPIO1 module of the general-purpose interface is attached to the WKUPAON power domain (see *Power Management*, in *Power, Reset, and Clock Management*, and can wake up the system.

Note

The GPIO2 to GPIO8 modules belong to the PD_L4PER power domain and thus their wake-up capabilities are operational only when the PD_L4PER power domain is active.

All wake-up sources (the 32 input GPIO channels) are merged together to issue a single asynchronous wake-up request in each GPIO module following the expected transition(s) (based on register programming). Each GPIO module generates a wake-up signal to the PRCM module.

Note

Only gpio1_[3:0] can be used to generate a direct wake-up event.

The asynchronous wake-up request line is active based on the GPIOi.GPIO_IRQWAKEN_0 and GPIO_IRQWAKEN_1 wake-up-enable registers (where i = 1 to 8).

The wake-up-enable register allows masking of the expected transition on input GPIO to prevent the generation of a wake-up request. The wake-up-enable register is programmed synchronously with the interface clock before any idle mode request coming from the host processor.

This register can be accessed with direct read/write operations.

Note

There must be a correlation between the wake-up enable and interrupt-enable registers. If a GPIO pin has a wake-up configured on it, it must also have the corresponding interrupt enabled (on one of the two interrupt lines). Otherwise, it is possible to have a wake-up event, but after exiting the IDLE state, no interrupt is generated; thus, the corresponding bit from the interrupt status register is not cleared, and the module does not acknowledge a future IDLE request.

Table 27-9 lists the mapping of the wake-up signals.

Table 27-9. Wake-Up Signals

Name	Mapping	Comments
GPIOi_WAKE	GPIOi_SWAKEUP	Where i = 1 to 8. The destination is the PRCM module.

27.4.7 General-Purpose Interface Channels Description

Table 27-10 describes the GPIO channels.

Table 27-10. GPIO Channels Description

Channel Number	Type(1)	Mapping	Wake-Up Feature	Comments
GPIO1				
[3:0]	I	gpio1_[3:0]	Yes	GPIO. Wake-up path. Inputs only.
[31:4]	I/O	gpio1_[31:4]	Yes	GPIO. Wake-up path.
GPIO2				
[31:0]	I/O	gpio2_[31:0]	Yes	GPIO
GPIO3				
[31:0]	I/O	gpio3_[31:0]	Yes	GPIO
GPIO4				
[31:0]	I/O	gpio4_[31:0]	Yes	GPIO
GPIO5				
[31:0]	I/O	gpio5_[31:0]	Yes	GPIO
GPIO6				
[3:0]	—	Reserved	Reserved	Not pinned out.
[31:4]	I/O	gpio6_[31:4]	Yes	GPIO
GPIO7				
[19:0]	I/O	gpio7_[19:0]	Yes	GPIO
[21:20]	—	Reserved	Reserved	Not pinned out.
[31:22]	I/O	gpio7_[31:22]	Yes	GPIO
GPIO8				
[23:0]	I/O	gpio8_[23:0]	Yes	GPIO
[24]	I	CHGDETECT1	Yes	Charge-detect interrupt from USB2PHY1.
[25]	I	CHGDETECT2	Yes	Charge-detect interrupt from USB2PHY2.
[26]	—	Reserved	Reserved	Not pinned out.
[27]	I	gpio8_27	Yes	GPIO. Input only.
[31:28]	I/O	gpio8_in[31:28]	Yes	GPIO

(1) I = Input; O = Output; I/O = bidirectional

Note

Channels 24 and 25 of GPIO8 are internally connected inside the device to the charger detection (CHGDETECT) output, respectively, to USB2PHY1 and USB2PHY2. This allows interrupt generation on the corresponding GPIO channel, if either of the mentioned events occurs.

Note

For more information about pin configuration, see *Pad Configuration Registers*, in *Control Module*.

Note

GPIO pins can directly snoop device pads even if those are configured in the other modes. For GPIO output user needs to set pad mux.

27.4.8 General-Purpose Interface Data Input/Output Capabilities

The output-enable register (GPIOi.GPIO_OE) controls the I/O capability of each pin. At reset, all the GPIO-related pins are configured as inputs, and their output capabilities are disabled. This register is not used within the module. Its only function is to carry the pad configuration.

When configured as an output (the desired bit reset in the GPIOi.GPIO_OE register), the value of the corresponding bit in the GPIOi.GPIO_DATAOUT register is driven on the corresponding GPIO pin. Data is written to the data-output register synchronously with the interface clock. This register can be accessed with read/write operations or by using the alternate set-and-clear protocol register update feature. This feature gives the possibility to set or clear specific bits of this register with a single write access to the set output data register (GPIOi.GPIO_SETDATAOUT) or to the clear output data register (GPIOi.GPIO_CLEARDATAOUT) address (see Section 27.4.9, *General-Purpose Interface Set-and-Clear Protocol*). If the application uses a pin as an output and does not want interrupt/wake-up generation from this pin, the application must properly configure the wake-up enable (GPIOi.GPIO_ WAKEUPENABLE) and the interrupt-enable (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_1) registers.

When configured as an input (the desired bit is set to 1 in the GPIOi.GPIO_OE register), the state of the input can be read from the corresponding bit in the GPIOi.GPIO_DATAIN register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. When the GPIO pin levels change, they are captured into this register after two interface clock cycles (the required cycles to synchronize and write data). If the application uses a pin as an input, the application must properly configure the wake-up enable (GPIOi.GPIO_IRQWAKEN_0 and GPIOi.GPIO_IRQWAKEN_1) and the interrupt-enable (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_0) registers to the interrupt and wake-up feature as needed. For using the alternate set-and-clear protocol, see Section 27.4.9, *General-Purpose Interface Set-and-Clear Protocol.*

27.4.9 General-Purpose Interface Set-and-Clear Protocol

27.4.9.1 Description

The GPIO module implements the set-and-clear protocol register update for the following registers:

- GPIOi.GPIO_DATAOUT
- GPIOi.GPIO_IRQSTATUS_CLR_0
- GPIOi.GPIO_IRQSTATUS_CLR_1
- GPIOi.GPIO_IRQSTATUS_SET_0
- GPIOi.GPIO_IRQSTATUS_SET_1

This protocol is an alternative to the atomic test and set operations and consists of writing operations at dedicated addresses (one address for setting bit[s] and one address for clearing bit[s]). The data to write is 1 at bit position(s) to clear (or to set) and 0 at unaffected bit(s). Registers can be accessed in two ways:

- Standard: Full register read and write operations at the primary register address
- Set and clear: Separate addresses are provided to set (and clear) bits in registers. Writing 1 at these addresses sets (or clears) the corresponding bit into the equivalent register; writing 0 has no effect.

Therefore, for these registers, three addresses are defined for one unique physical register. Reading these addresses has the same effect and returns the register value.

27.4.9.2 Clear Instruction

27.4.9.2.1 Clear Register Addresses

- Clear interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_CLR_0 and GPIOi.GPIO_IRQSTATUS_CLR_1).

  A write operation in the GPIO_IRQSTATUS_CLR_0 (or GPIO_IRQSTATUS_CLR_1) register clears the corresponding bit in the same register when the written bit is 1; a written bit at 0 has no effect.

  A read of the clear interrupt enable0 (or enable1) register returns the value of the GPIO_IRQSTATUS_CLR_0 (or GPIO_IRQSTATUS_CLR_1) register.
- Clear data-output register (GPIOi.GPIO_CLEARDATAOUT).

  A write operation in the clear data-output register clears the corresponding bit in the data-output register when the written bit is 1; a written bit at 0 has no effect.

  A read of the clear data-output register returns the value of the data-output register.

27.4.9.2.2 Clear Instruction Example

Assume the data-output register (or one of the interrupt or wake-up-enable registers) contains the binary value 0b0000 0001 0000 0001 and bit 0 is to be cleared.

With the clear instruction feature, write 0b0000 0000 0000 0001 at the address of the clear data-output register (or at the address of the clear interrupt or wake-up-enable register). After this write operation, a reading of the data-output register (or the interrupt or wake-up-enable register) returns 0b0000 0001 0000 0000; bit 0 is cleared.

Note

Although the general-purpose interface registers are 32 bits wide, only the 16 least-significant bits (LSBs) are represented in this example.

Figure 27-13 is an example of a clear instruction.

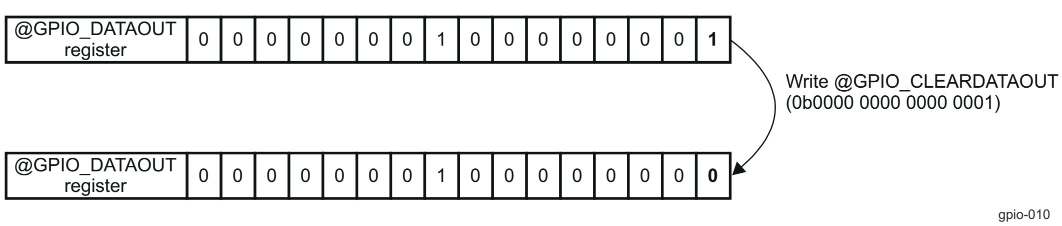


Figure 27-13. GPIO_CLEARDATAOUT Register Example

27.4.9.3 Set Instruction

27.4.9.3.1 Set Register Addresses

- Set interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_1).

  A write operation in the GPIOi.GPIO_IRQSTATUS_SET_0 (or GPIOi.GPIO_IRQSTATUS_SET_1) register sets the corresponding bit in the same register when the written bit is 1; a written bit at 0 has no effect.

  A read of the set interrupt-enable 0 (or enable1) register returns the value of the interrupt GPIOi.GPIO_IRQSTATUS_SET_0 (or GPIOi.GPIO_IRQSTATUS_SET_1) register.
- Set data-output register (GPIOi.GPIO_SETDATAOUT).

A write operation in the set data-output register sets the corresponding bit in the data-output register when the written bit is 1; a written bit at 0 has no effect.

A read of the set data-output register returns the value of the data-output register.

27.4.9.3.2 Set Instruction Example

Assume the interrupt enable1 (or enable2) register (or the data-output register) contains the binary value 0b0000 0001 0000 0000 and bits 15, 3, 2, and 1 are to be set.

With the set instruction feature, the user has only to write 0b1000 0000 0000 1110 at the address of the GPIO_SETDATAOUT register. After this write operation, a reading of the GPIO_SETDATAOUT register returns 0b1000 0001 0000 1110: bits 15, 3, 2, and 1 have been set.

Note

Although the general-purpose interface registers are 32 bits wide, only the 16 LSBs are represented in this example.

Figure 27-14 is an example of a set instruction.

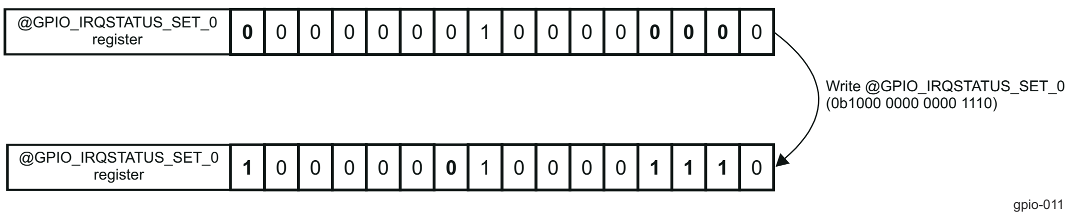


Figure 27-14. Write in GPIO_IRQSTATUS_SET_0 Register Example

The set wake-up-enable register offers the same feature with the wake-up-enable register.

27.5 General-Purpose Interface Programming Guide

27.5.1 General-Purpose Interface Low-Level Programming Models

27.5.1.1 Global Initialization

27.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the general-purpose interface module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the environment and integration of the general-purpose interface. For more information, see Section 27.2, *General-Purpose Interface Environment*, and Section 27.3, *General-Purpose Interface Integration*.

Table 27-11. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information about the module configuration, see *Clock Management*, in *Power, Reset, and Clock Management*.
Control module	Module specific pad muxing must be set in the control module. For more information about the module configuration, see *Pad Configuration Registers*, in *Control Module*.
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see *Interrupt Controllers*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

27.5.1.1.2 General-Purpose Interface Module Global Initialization

This procedure initializes the general-purpose Interface module after a power-on reset (POR) or software reset.

Table 27-12. General-Purpose Interface Global Initialization

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	GPIO_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset completed?	GPIO_SYSSTATUS[0] RESETDONE	= 0x1
Configure idle mode.	GPIO_SYSCONFIG[4:3] IDLEMODE	0x-
Configure interface clock gating.	GPIO_SYSCONFIG[0] AUTOIDLE	0x-
Set clock-gating ratio.	GPIO_CTRL [2:1] GATINGRATIO	0x-
Configure GPIO channels as input or output.	GPIO_OE[31:0] OUTPUTEN	0x-
Set debounce time value.	GPIO_DEBOUNCINGTIME[7:0] DEBOUNCINGTIME	0x-
Enable/disable debouncing for desired input line. (For example, when used with a push-button)	GPIO_DEBOUNCENABLE[31:0] DEBOUNCENABLE	0x-
Interrupt and wake-up requests configuration		
(Optional) Enable/disable wake-up for desired input lines.	GPIO_IRQWAKEN_0[31:0] INTLINE and/or GPIO_IRQWAKEN_1[31:0] INTLINE	0x-
(Optional) Enable wake-up generation.	GPIO_SYSCONFIG[2] ENAWAKEUP	0x1
Configure detection events. **Note** NOTE: Simultaneous enabling of high-level and low-level detection for one given pin creates a constant-interrupt generator.	GPIO_LEVELDETECT0[31:0] LEVELDETECT0 and/or GPIO_LEVELDETECT1[31:0] LEVELDETECT1 and/or GPIO_RISINGDETECT[31:0] RISINGDETECT and/or GPIO_FALLINGDETECT[31:0] FALLINGDETECT	0x-
Clear interrupt status	GPIO_IRQSTATUS_0[31:0] INTLINE and/or GPIO_IRQSTATUS_1[31:0] INTLINE	0xFFFF FFFF

Table 27-12. General-Purpose Interface Global Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Enable interrupts for desired input lines. If wakeup is enabled, it is mandatory to enable the corresponding interrupt.	GPIO_IRQSTATUS_SET_0[31:0] INTLINE and/or GPIO_IRQSTATUS_SET_1[31:0] INTLINE	0x-

Note

Detection of events requires a functional clock running for every group of 8 bits. If detection of events is enabled only within one octet (for example, 0x0012 0000), power saving can be achieved. Else (for example, 0x0102 0000), using two octets requires one more clock to be run by the module.

27.5.1.2 General-Purpose Interface Operational Modes Configuration

27.5.1.2.1 General-Purpose Interface Read Input Register

Table 27-13. General-Purpose Interface Read Input Register

Step	Register/Bit Field/Programming Model	Value
Read interrupt status	GPIO_IRQSTATUS_0[31:0] INTLINE and/or GPIO_IRQSTATUS_1[31:0] INTLINE	0x-
Read input register value.	GPIO_DATAIN[31:0] DATAIN	0x-
Clear interrupt status	GPIO_IRQSTATUS_0[31:0] INTLINE and/or GPIO_IRQSTATUS_1[31:0] INTLINE	0xFFFF FFFF

27.5.1.2.2 General-Purpose Interface Set Bit Function

Table 27-14. General-Purpose Interface Set Bit Function

Step	Register/Bit Field/Programming Model	Value
Write 0x1 to set desired bit(s) in DATAOUT register.	GPIO_SETDATAOUT [31:0] INTLINE	0x-

27.5.1.2.3 General-Purpose Interface Clear Bit Function

Table 27-15. General-Purpose Interface Clear Bit Function

Step	Register/Bit Field/Programming Model	Value
Write 0x1 to clear desired bit(s) in DATAOUT register.	GPIO_CLEARDATAOUT [31:0] INTLINE	0x-

27.6 General-Purpose Interface Register Manual

27.6.1 General-Purpose Interface Instance Summary

Table 27-16 summarizes the general-purpose interface instance.

Table 27-16. Instance Summary

Module Name	Module Base Address	Size
GPIO7	0x4805 1000	408 Bytes
GPIO8	0x4805 3000	408 Bytes
GPIO2	0x4805 5000	408 Bytes
GPIO3	0x4805 7000	408 Bytes
GPIO4	0x4805 9000	408 Bytes
GPIO5	0x4805 B000	408 Bytes
GPIO6	0x4805 D000	408 Bytes
GPIO1	0x4AE1 0000	408 Bytes

27.6.2 General-Purpose Interface Registers

27.6.2.1 General-Purpose Interface Register Summary

Table 27-17 summarizes the general-purpose interface GPIO2, GPIO7 and GPIO8 registers.

Table 27-17. General-Purpose Interface GPIO2, GPIO7 and GPIO8 Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPIO7 L4_PER1 Physical Address	GPIO8 L4_PER1 Physical Address	GPIO2 L4_PER1 Physical Address
GPIO_REVISION	R	32	0x0000 0000	0x4805 1000	0x4805 3000	0x4805 5000
GPIO_SYSCONFIG	RW	32	0x0000 0010	0x4805 1010	0x4805 3010	0x4805 5010
GPIO_EOI	W	32	0x0000 0020	0x4805 1020	0x4805 3020	0x4805 5020
GPIO_IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x4805 1024	0x4805 3024	0x4805 5024
GPIO_IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x4805 1028	0x4805 3028	0x4805 5028
GPIO_IRQSTATUS_0	RW	32	0x0000 002C	0x4805 102C	0x4805 302C	0x4805 502C
GPIO_IRQSTATUS_1	RW	32	0x0000 0030	0x4805 1030	0x4805 3030	0x4805 5030
GPIO_IRQSTATUS_SET_0	RW	32	0x0000 0034	0x4805 1034	0x4805 3034	0x4805 5034
GPIO_IRQSTATUS_SET_1	RW	32	0x0000 0038	0x4805 1038	0x4805 3038	0x4805 5038
GPIO_IRQSTATUS_CLR_0	RW	32	0x0000 003C	0x4805 103C	0x4805 303C	0x4805 503C
GPIO_IRQSTATUS_CLR_1	RW	32	0x0000 0040	0x4805 1040	0x4805 3040	0x4805 5040
GPIO_IRQWAKEN_0	RW	32	0x0000 0044	0x4805 1044	0x4805 3044	0x4805 5044
GPIO_IRQWAKEN_1	RW	32	0x0000 0048	0x4805 1048	0x4805 3048	0x4805 5048
GPIO_SYSSTATUS	R	32	0x0000 0114	0x4805 1114	0x4805 3114	0x4805 5114
RESERVED	RW	32	0x0000 0118	0x4805 1118	0x4805 3118	0x4805 5118
RESERVED	RW	32	0x0000 011C	0x4805 111C	0x4805 311C	0x4805 511C
RESERVED	RW	32	0x0000 0120	0x4805 1120	0x4805 3120	0x4805 5120
RESERVED	RW	32	0x0000 0128	0x4805 1128	0x4805 3128	0x4805 5128

Table 27-17. General-Purpose Interface GPIO2, GPIO7 and GPIO8 Registers Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	GPIO7 L4_PER1 Physical Address	GPIO8 L4_PER1 Physical Address	GPIO2 L4_PER1 Physical Address
RESERVED	RW	32	0x0000 012C	0x4805 112C	0x4805 312C	0x4805 512C
GPIO_CTRL	RW	32	0x0000 0130	0x4805 1130	0x4805 3130	0x4805 5130
GPIO_OE	RW	32	0x0000 0134	0x4805 1134	0x4805 3134	0x4805 5134
GPIO_DATAIN	R	32	0x0000 0138	0x4805 1138	0x4805 3138	0x4805 5138
GPIO_DATAOUT	RW	32	0x0000 013C	0x4805 113C	0x4805 313C	0x4805 513C
GPIO_LEVELDETECT0	RW	32	0x0000 0140	0x4805 1140	0x4805 3140	0x4805 5140
GPIO_LEVELDETECT1	RW	32	0x0000 0144	0x4805 1144	0x4805 3144	0x4805 5144
GPIO_RISINGDETECT	RW	32	0x0000 0148	0x4805 1148	0x4805 3148	0x4805 5148
GPIO_FALLINGDETECT	RW	32	0x0000 014C	0x4805 114C	0x4805 314C	0x4805 514C
GPIO_DEBOUNCENABLE	RW	32	0x0000 0150	0x4805 1150	0x4805 3150	0x4805 5150
GPIO_DEBOUNCINGTIME	RW	32	0x0000 0154	0x4805 1154	0x4805 3154	0x4805 5154
RESERVED	RW	32	0x0000 0160	0x4805 1160	0x4805 3160	0x4805 5160
RESERVED	RW	32	0x0000 0164	0x4805 1164	0x4805 3164	0x4805 5164
RESERVED	RW	32	0x0000 0170	0x4805 1170	0x4805 3170	0x4805 5170
RESERVED	RW	32	0x0000 0174	0x4805 1174	0x4805 3174	0x4805 5174
RESERVED	RW	32	0x0000 0180	0x4805 1180	0x4805 3180	0x4805 5180
RESERVED	RW	32	0x0000 0184	0x4805 1184	0x4805 3184	0x4805 5184
GPIO_CLEARDATAOUT	RW	32	0x0000 0190	0x4805 1190	0x4805 3190	0x4805 5190
GPIO_SETDATAOUT	RW	32	0x0000 0194	0x4805 1194	0x4805 3194	0x4805 5194

Table 27-18 summarizes the general-purpose interface GPIO3 to GPIO5 registers.

Table 27-18. General-Purpose Interface GPIO3 to GPIO5 Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPIO3 L4_PER1 Physical Address	GPIO4 L4_PER1 Physical Address	GPIO5 L4_PER1 Physical Address
GPIO_REVISION	R	32	0x0000 0000	0x4805 7000	0x4805 9000	0x4805 B000
GPIO_SYSCONFIG	RW	32	0x0000 0010	0x4805 7010	0x4805 9010	0x4805 B010
GPIO_EOI	W	32	0x0000 0020	0x4805 7020	0x4805 9020	0x4805 B020
GPIO_IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x4805 7024	0x4805 9024	0x4805 B024
GPIO_IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x4805 7028	0x4805 9028	0x4805 B028
GPIO_IRQSTATUS_0	RW	32	0x0000 002C	0x4805 702C	0x4805 902C	0x4805 B02C
GPIO_IRQSTATUS_1	RW	32	0x0000 0030	0x4805 7030	0x4805 9030	0x4805 B030
GPIO_IRQSTATUS_SET_0	RW	32	0x0000 0034	0x4805 7034	0x4805 9034	0x4805 B034
GPIO_IRQSTATUS_SET_1	RW	32	0x0000 0038	0x4805 7038	0x4805 9038	0x4805 B038
GPIO_IRQSTATUS_CLR_0	RW	32	0x0000 003C	0x4805 703C	0x4805 903C	0x4805 B03C

Table 27-18. General-Purpose Interface GPIO3 to GPIO5 Registers Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	GPIO3 L4_PER1 Physical Address	GPIO4 L4_PER1 Physical Address	GPIO5 L4_PER1 Physical Address
GPIO_IRQSTATUS_CLR_1	RW	32	0x0000 0040	0x4805 7040	0x4805 9040	0x4805 B040
GPIO_IRQWAKEN_0	RW	32	0x0000 0044	0x4805 7044	0x4805 9044	0x4805 B044
GPIO_IRQWAKEN_1	RW	32	0x0000 0048	0x4805 7048	0x4805 9048	0x4805 B048
GPIO_SYSSTATUS	R	32	0x0000 0114	0x4805 7114	0x4805 9114	0x4805 B114
RESERVED	RW	32	0x0000 0118	0x4805 7118	0x4805 9118	0x4805 B118
RESERVED	RW	32	0x0000 011C	0x4805 711C	0x4805 911C	0x4805 B11C
RESERVED	RW	32	0x0000 0120	0x4805 7120	0x4805 9120	0x4805 B120
RESERVED	RW	32	0x0000 0128	0x4805 7128	0x4805 9128	0x4805 B128
RESERVED	RW	32	0x0000 012C	0x4805 712C	0x4805 912C	0x4805 B12C
GPIO_CTRL	RW	32	0x0000 0130	0x4805 7130	0x4805 9130	0x4805 B130
GPIO_OE	RW	32	0x0000 0134	0x4805 7134	0x4805 9134	0x4805 B134
GPIO_DATAIN	R	32	0x0000 0138	0x4805 7138	0x4805 9138	0x4805 B138
GPIO_DATAOUT	RW	32	0x0000 013C	0x4805 713C	0x4805 913C	0x4805 B13C
GPIO_LEVELDETECT0	RW	32	0x0000 0140	0x4805 7140	0x4805 9140	0x4805 B140
GPIO_LEVELDETECT1	RW	32	0x0000 0144	0x4805 7144	0x4805 9144	0x4805 B144
GPIO_RISINGDETECT	RW	32	0x0000 0148	0x4805 7148	0x4805 9148	0x4805 B148
GPIO_FALLINGDETECT	RW	32	0x0000 014C	0x4805 714C	0x4805 914C	0x4805 B14C
GPIO_DEBOUNCENABLE	RW	32	0x0000 0150	0x4805 7150	0x4805 9150	0x4805 B150
GPIO_DEBOUNCINGTIME	RW	32	0x0000 0154	0x4805 7154	0x4805 9154	0x4805 B154
RESERVED	RW	32	0x0000 0160	0x4805 7160	0x4805 9160	0x4805 B160
RESERVED	RW	32	0x0000 0164	0x4805 7164	0x4805 9164	0x4805 B164
RESERVED	RW	32	0x0000 0170	0x4805 7170	0x4805 9170	0x4805 B170
RESERVED	RW	32	0x0000 0174	0x4805 7174	0x4805 9174	0x4805 B174
RESERVED	RW	32	0x0000 0180	0x4805 7180	0x4805 9180	0x4805 B180
RESERVED	RW	32	0x0000 0184	0x4805 7184	0x4805 9184	0x4805 B184
GPIO_CLEARDATAOUT	RW	32	0x0000 0190	0x4805 7190	0x4805 9190	0x4805 B190
GPIO_SETDATAOUT	RW	32	0x0000 0194	0x4805 7194	0x4805 9194	0x4805 B194

Table 27-19 summarizes the general-purpose interface GPIO6 and GPIO1 registers.

Table 27-19. General-Purpose Interface GPIO6 and GPIO1 Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPIO6 L4_PER1 Physical Address	GPIO1 L4_WKUP Physical Address
GPIO_REVISION	R	32	0x0000 0000	0x4805 D000	0x4AE1 0000
GPIO_SYSCONFIG	RW	32	0x0000 0010	0x4805 D010	0x4AE1 0010
GPIO_EOI	W	32	0x0000 0020	0x4805 D020	0x4AE1 0020
GPIO_IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x4805 D024	0x4AE1 0024

Table 27-19. General-Purpose Interface GPIO6 and GPIO1 Registers Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	GPIO6 L4_PER1 Physical Address	GPIO1 L4_WKUP Physical Address
GPIO_IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x4805 D028	0x4AE1 0028
GPIO_IRQSTATUS_0	RW	32	0x0000 002C	0x4805 D02C	0x4AE1 002C
GPIO_IRQSTATUS_1	RW	32	0x0000 0030	0x4805 D030	0x4AE1 0030
GPIO_IRQSTATUS_SET_0	RW	32	0x0000 0034	0x4805 D034	0x4AE1 0034
GPIO_IRQSTATUS_SET_1	RW	32	0x0000 0038	0x4805 D038	0x4AE1 0038
GPIO_IRQSTATUS_CLR_0	RW	32	0x0000 003C	0x4805 D03C	0x4AE1 003C
GPIO_IRQSTATUS_CLR_1	RW	32	0x0000 0040	0x4805 D040	0x4AE1 0040
GPIO_IRQWAKEN_0	RW	32	0x0000 0044	0x4805 D044	0x4AE1 0044
GPIO_IRQWAKEN_1	RW	32	0x0000 0048	0x4805 D048	0x4AE1 0048
GPIO_SYSSTATUS	R	32	0x0000 0114	0x4805 D114	0x4AE1 0114
RESERVED	RW	32	0x0000 0118	0x4805 D118	0x4AE1 0118
RESERVED	RW	32	0x0000 011C	0x4805 D11C	0x4AE1 011C
RESERVED	RW	32	0x0000 0120	0x4805 D120	0x4AE1 0120
RESERVED	RW	32	0x0000 0128	0x4805 D128	0x4AE1 0128
RESERVED	RW	32	0x0000 012C	0x4805 D12C	0x4AE1 012C
GPIO_CTRL	RW	32	0x0000 0130	0x4805 D130	0x4AE1 0130
GPIO_OE	RW	32	0x0000 0134	0x4805 D134	0x4AE1 0134
GPIO_DATAIN	R	32	0x0000 0138	0x4805 D138	0x4AE1 0138
GPIO_DATAOUT	RW	32	0x0000 013C	0x4805 D13C	0x4AE1 013C
GPIO_LEVELDETECT0	RW	32	0x0000 0140	0x4805 D140	0x4AE1 0140
GPIO_LEVELDETECT1	RW	32	0x0000 0144	0x4805 D144	0x4AE1 0144
GPIO_RISINGDETECT	RW	32	0x0000 0148	0x4805 D148	0x4AE1 0148
GPIO_FALLINGDETECT	RW	32	0x0000 014C	0x4805 D14C	0x4AE1 014C
GPIO_DEBOUNCENABLE	RW	32	0x0000 0150	0x4805 D150	0x4AE1 0150
GPIO_DEBOUNCINGTIME	RW	32	0x0000 0154	0x4805 D154	0x4AE1 0154
RESERVED	RW	32	0x0000 0160	0x4805 D160	0x4AE1 0160
RESERVED	RW	32	0x0000 0164	0x4805 D164	0x4AE1 0164
RESERVED	RW	32	0x0000 0170	0x4805 D170	0x4AE1 0170
RESERVED	RW	32	0x0000 0174	0x4805 D174	0x4AE1 0174
RESERVED	RW	32	0x0000 0180	0x4805 D180	0x4AE1 0180
RESERVED	RW	32	0x0000 0184	0x4805 D184	0x4AE1 0184
GPIO_CLEARDATAOUT	RW	32	0x0000 0190	0x4805 D190	0x4AE1 0190
GPIO_SETDATAOUT	RW	32	0x0000 0194	0x4805 D194	0x4AE1 0194

27.6.2.2 General-Purpose Interface Register Description

Table 27-20 through Table 27-45 describe the individual general-purpose interface registers.

Table 27-20. GPIO_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4805 1000 0x4805 3000 0x4805 5000 0x4805 7000 0x4805 9000 0x4805 B000 0x4805 D000 0x4AE1 0000	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	IP revision identifier (X.Y.R)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See[1]

(1) TI Internal Data

Table 27-21. GPIO_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4805 1010 0x4805 3010 0x4805 5010 0x4805 7010 0x4805 9010 0x4805 B010 0x4805 D010 0x4AE1 0010	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	System configuration register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:3	IDLEMODE	0x0: Force-idle: An IDLE request is acknowledged unconditionally. 0x1: No-idle: An IDLE request is never acknowledged. 0x2: Smart-idle: The acknowledgment to an IDLE request is given based on the internal activity (see Section 27.4.5.2.3, *System Power Management and Wakeup*). 0x3: Smart-idle wakeup	RW	0x0
2	ENAWAKEUP	Wake-up control. 0x0: Wake-up generation is disabled. 0x1: Wake-up capability is enabled upon expected transition on input GPIO pin	RW	0

Bits	Field Name	Description	Type	Reset
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	OCP clock gating control. 0x0: Internal interface OCP clock is free-running. 0x1: Automatic internal OCP clock gating, based on the OCP interface activity	RW	0

Table 27-22. GPIO_EOI

Address Offset	0x0000 0020		
Physical Address	0x4805 1020 0x4805 3020 0x4805 5020 0x4805 7020 0x4805 9020 0x4805 B020 0x4805 D020 0x4AE1 0020	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Software end of interrupt.		
Type	W		

31:1	0
RESERVED	LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. 0x0: EOI for interrupt line number 0. Read returns 0. 0x1: EOI for interrupt line number 1. Read returns 0.	W	0x0

Table 27-23. GPIO_IRQSTATUS_RAW_0

Address Offset	0x0000 0024		
Physical Address	0x4805 1024 0x4805 3024 0x4805 5024 0x4805 7024 0x4805 9024 0x4805 B024 0x4805 D024 0x4AE1 0024	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event raw interrupt status vector, showing all active events (enabled and not enabled), (corresponding to first line of interrupt)		
Type	RW		

31:0
INTLINE

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status raw for interrupt line. Writing '1' to a bit will set it to '1.' Writing '0' has no effect	RW	0x0000 0000

Table 27-24. GPIO_IRQSTATUS_RAW_1

Address Offset	0x0000 0028		
Physical Address	0x4805 1028 0x4805 3028 0x4805 5028 0x4805 7028 0x4805 9028 0x4805 B028 0x4805 D028 0x4AE1 0028	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event raw interrupt status vector, showing all active events (enabled and not enabled), (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status raw for interrupt line Writing '1' to a bit will set it to '1.' Writing '0' has no effect	RW	0x0000 0000

Table 27-25. GPIO_IRQSTATUS_0

Address Offset	0x0000 002C		
Physical Address	0x4805 102C 0x4805 302C 0x4805 502C 0x4805 702C 0x4805 902C 0x4805 B02C 0x4805 D02C 0x4AE1 002C	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event interrupt status vector, showing all active and enabled events (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for interrupt line Writing 1 to a bit will clear it to 0. Writing 0 has no effect.	RW	0x0000 0000

Table 27-26. GPIO_IRQSTATUS_1

Address Offset	0x0000 0030		
Physical Address	0x4805 1030 0x4805 3030 0x4805 5030 0x4805 7030 0x4805 9030 0x4805 B030 0x4805 D030 0x4AE1 0030	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event enabled interrupt status vector, showing all active and enabled events (corresponding to second line of interrupt)		

Table 27-26. GPIO_IRQSTATUS_1 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for interrupt line Writing 1 to a bit will clear it to 0. Writing 0 has no effect.	RW	0x0000 0000

Table 27-27. GPIO_IRQSTATUS_SET_0

Address Offset	0x0000 0034		
Physical Address	0x4805 1034 0x4805 3034 0x4805 5034 0x4805 7034 0x4805 9034 0x4805 B034 0x4805 D034 0x4AE1 0034	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event interrupt-enable set vector (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status set for interrupt line Writing 1 to a bit enables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 27-28. GPIO_IRQSTATUS_SET_1

Address Offset	0x0000 0038		
Physical Address	0x4805 1038 0x4805 3038 0x4805 5038 0x4805 7038 0x4805 9038 0x4805 B038 0x4805 D038 0x4AE1 0038	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event enable set interrupt vector (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status set for interrupt line Writing 1 to a bit enables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 27-29. GPIO_IRQSTATUS_CLR_0

Address Offset	0x0000 003C

Table 27-29. GPIO_IRQSTATUS_CLR_0 (continued)

Physical Address	0x4805 103C 0x4805 303C 0x4805 503C 0x4805 703C 0x4805 903C 0x4805 B03C 0x4805 D03C 0x4AE1 003C	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event interrupt-enable clear vector (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status clear for interrupt line Writing 1 to a bit disables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 27-30. GPIO_IRQSTATUS_CLR_1

Address Offset	0x0000 0040		
Physical Address	0x4805 1040 0x4805 3040 0x4805 5040 0x4805 7040 0x4805 9040 0x4805 B040 0x4805 D040 0x4AE1 0040	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event enable clear interrupt vector (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status clear for interrupt line Writing 1 to a bit disables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 27-31. GPIO_IRQWAKEN_0

Address Offset	0x0000 0044		
Physical Address	0x4805 1044 0x4805 3044 0x4805 5044 0x4805 7044 0x4805 9044 0x4805 B044 0x4805 D044 0x4AE1 0044	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event wake-up enable set vector (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Wakeup set for interrupt line Setting a bit to 1 will enable wakeup for the corresponding event. Setting a bit to 0 will disable wakeup for the corresponding event.	RW	0x0000 0000

Table 27-32. GPIO_IRQWAKEN_1

Address Offset	0x0000 0048		
Physical Address	0x4805 1048 0x4805 3048 0x4805 5048 0x4805 7048 0x4805 9048 0x4805 B048 0x4805 D048 0x4AE1 0048	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Per-event wake-up enable set vector (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Wakeup set for interrupt line Setting a bit to 1 will enable wakeup for the corresponding event. Setting a bit to 0 will disable wakeup for the corresponding event.	RW	0x0000 0000

Table 27-33. GPIO_SYSSTATUS

Address Offset	0x0000 0114		
Physical Address	0x4805 1114 0x4805 3114 0x4805 5114 0x4805 7114 0x4805 9114 0x4805 B114 0x4805 D114 0x4AE1 0114	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	System status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Read 0x0: Internal reset is ongoing. Read 0x1: Reset completed	R	0

Table 27-34. GPIO_CTRL

Address Offset	0x0000 0130

Table 27-34. GPIO_CTRL (continued)

Physical Address	0x4805 1130 0x4805 3130 0x4805 5130 0x4805 7130 0x4805 9130 0x4805 B130 0x4805 D130 0x4AE1 0130	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	GPIO control register		
Type	RW		

31:3	2:1	0
RESERVED	GATINGRATIO	DISABLEMODULE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:1	GATINGRATIO	Clock gating ratio for event detection 0x0: N = 1 0x1: N = 2 0x2: N = 4 0x3: N = 8	RW	0x1
0	DISABLEMODULE	0x0: Module is enabled, clocks are not gated. 0x1: Module is disabled, internal clocks are gated	RW	0

Table 27-35. GPIO_OE

Address Offset	0x0000 0134		
Physical Address	0x4805 1134 0x4805 3134 0x4805 5134 0x4805 7134 0x4805 9134 0x4805 B134 0x4805 D134 0x4AE1 0134	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Output enable register. 0 = Output enabled ; 1 = Output disabled		
Type	RW		

31:0
OUTPUTEN

Bits	Field Name	Description	Type	Reset
31:0	OUTPUTEN	Output enable 0x0: Output enabled 0x1: Output disabled	RW	0xFFFF FFFF

Table 27-36. GPIO_DATAIN

Address Offset	0x0000 0138

Table 27-36. GPIO_DATAIN (continued)

Physical Address	0x4805 1138 0x4805 3138 0x4805 5138 0x4805 7138 0x4805 9138 0x4805 B138 0x4805 D138 0x4AE1 0138	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Data input register (with sampled input data)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN																															

Bits	Field Name	Description	Type	Reset
31:0	DATAIN	Sampled input data	R	0x0000 0000

Table 27-37. GPIO_DATAOUT

Address Offset	0x0000 013C		
Physical Address	0x4805 113C 0x4805 313C 0x4805 513C 0x4805 713C 0x4805 913C 0x4805 B13C 0x4805 D13C 0x4AE1 013C	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Data-output register (data to set on output pins)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAOUT																															

Bits	Field Name	Description	Type	Reset
31:0	DATAOUT	Data to set on output pins	RW	0x0000 0000

Table 27-38. GPIO_LEVELDETECT0

Address Offset	0x0000 0140		
Physical Address	0x4805 1140 0x4805 3140 0x4805 5140 0x4805 7140 0x4805 9140 0x4805 B140 0x4805 D140 0x4AE1 0140	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Detect low-level register. 0 = Low-level detection disabled; 1 = Low-level detection enabled		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT0																															

Bits	Field Name	Description	Type	Reset
31:0	LEVELDETECT0	Low-level detection 0x0: Low-level detection disabled 0x1: Low-level detection enabled	RW	0x0000 0000

Table 27-39. GPIO_LEVELDETECT1

Address Offset	0x0000 0144		
Physical Address	0x4805 1144 0x4805 3144 0x4805 5144 0x4805 7144 0x4805 9144 0x4805 B144 0x4805 D144 0x4AE1 0144	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Detect high-level register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT1																															

Bits	Field Name	Description	Type	Reset
31:0	LEVELDETECT1	0x0: High-evel detection disabled 0x1: High-level detection enabled	RW	0x0000 0000

Table 27-40. GPIO_RISINGDETECT

Address Offset	0x0000 0148		
Physical Address	0x4805 1148 0x4805 3148 0x4805 5148 0x4805 7148 0x4805 9148 0x4805 B148 0x4805 D148 0x4AE1 0148	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Detect rising edge register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RISINGDETECT																															

Bits	Field Name	Description	Type	Reset
31:0	RISINGDETECT	0x0: Rising edge detection disabled 0x1: Rising edge detection enabled	RW	0x0000 0000

Table 27-41. GPIO_FALLINGDETECT

Address Offset	0x0000 014C		
Physical Address	0x4805 114C 0x4805 314C 0x4805 514C 0x4805 714C 0x4805 914C 0x4805 B14C 0x4805 D14C 0x4AE1 014C	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Detect falling edge register		

Table 27-41. GPIO_FALLINGDETECT (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALLINGDETECT																															

Bits	Field Name	Description	Type	Reset
31:0	FALLINGDETECT	0x0: Falling edge detection disabled 0x1: Falling edge detection enabled	RW	0x0000 0000

Table 27-42. GPIO_DEBOUNCENABLE

Address Offset	0x0000 0150		
Physical Address	0x4805 1150 0x4805 3150 0x4805 5150 0x4805 7150 0x4805 9150 0x4805 B150 0x4805 D150 0x4AE1 0150	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Debouncing enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEBOUNCEENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	DEBOUNCEENABLE	0x0: No debouncing 0x1: Debouncing activated	RW	0x0000 0000

Table 27-43. GPIO_DEBOUNCINGTIME

Address Offset	0x0000 0154		
Physical Address	0x4805 1154 0x4805 3154 0x4805 5154 0x4805 7154 0x4805 9154 0x4805 B154 0x4805 D154 0x4AE1 0154	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Debouncing value register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DEBOUNCETIME							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	DEBOUNCETIME	8-bit values specifying the debouncing time. It is n-periods of the muxed clock, which can come from either a true 32k oscillator/pad of from the system clock. It depends on which boot mode is selected. For more information see *Initialization*.	RW	0x00

Table 27-44. GPIO_CLEARDATAOUT

Address Offset	0x0000 0190

Table 27-44. GPIO_CLEARDATAOUT (continued)

Physical Address	0x4805 1190 0x4805 3190 0x4805 5190 0x4805 7190 0x4805 9190 0x4805 B190 0x4805 D190 0x4AE1 0190	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Clear data-output register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Clear the corresponding bit in the data-output register	RW	0x0000 0000

Table 27-45. GPIO_SETDATAOUT

Address Offset	0x0000 0194		
Physical Address	0x4805 1194 0x4805 3194 0x4805 5194 0x4805 7194 0x4805 9194 0x4805 B194 0x4805 D194 0x4AE1 0194	**Instance**	GPIO7 GPIO8 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO1
Description	Set data-output register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Set the corresponding bit in the data-output register	RW	0x0000 0000

Chapter 28

Keyboard Controller

This chapter describes the keyboard module (KBD) of the device.

28.1 Keyboard Controller Overview 6661
28.2 Keyboard Controller Environment 6663
28.3 Keyboard Controller Integration 6666
28.4 Keyboard Controller Functional Description 6668
28.5 Keyboard Controller Programming Guide 6677
28.6 Keyboard Controller Register Manual 6681

28.1 Keyboard Controller Overview

The keyboard controller implements a built-in scanning algorithm for hardware-based key-press decoding and reduces overhead in the microprocessor unit (MPU) software.

The keyboard controller includes a debouncing feature to ensure that only one key combination can be registered in the programmed time.

The keyboard controller can handle up to 9 × 9 keys, works on a 32-kHz clock, and can generate wake-up events when the chip is in sleep mode.

Figure 28-1 shows the keyboard controller.

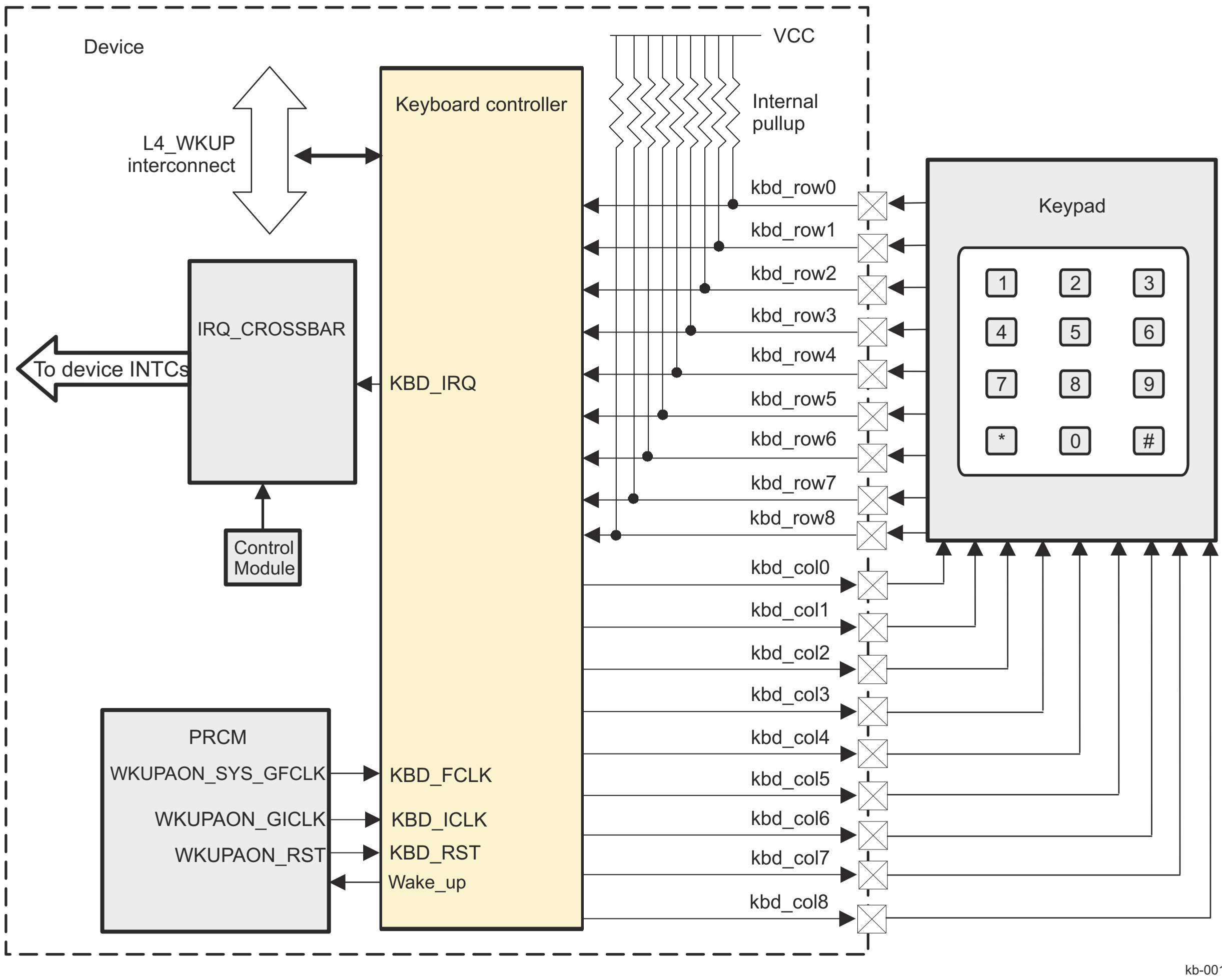


Figure 28-1. Keyboard Controller Overview

The keyboard controller includes the following main features:

- Support of multiconfiguration keyboards up to 9 rows × 9 columns
- Each key coded on 1 bit in two 32-bit registers
- Long-key value or repeat timing reconfigurable on the fly
- Event detection on key press and key release
- Multikey-press detection and decoding
- Long-key detection on prolonged key press
- Integrated timer with four programmable comparison values
- Programmable time-out on permanent key press or after keyboard release
- Programmable interrupt generation on key events

- Software reset capability
- Read/write-posted register access modes
- 8-/16-/32-bit access supported on the level 4 (L4) interface
- 32-bit data bus
- 7-bit address bus
- An over run generation if a master reads a register to late

28.2 Keyboard Controller Environment

The keyboard controller external interface pins map on the device pads when the device IO cells are configured and multiplexed for the keypad function by the control module. For more information, see *Pad Configuration Registers*, and Section 27.2.1, *General-Purpose Interface as a Keyboard interface*.

The external keypad typically connects directly to the keyboard controller of the device (see Figure 28-2).

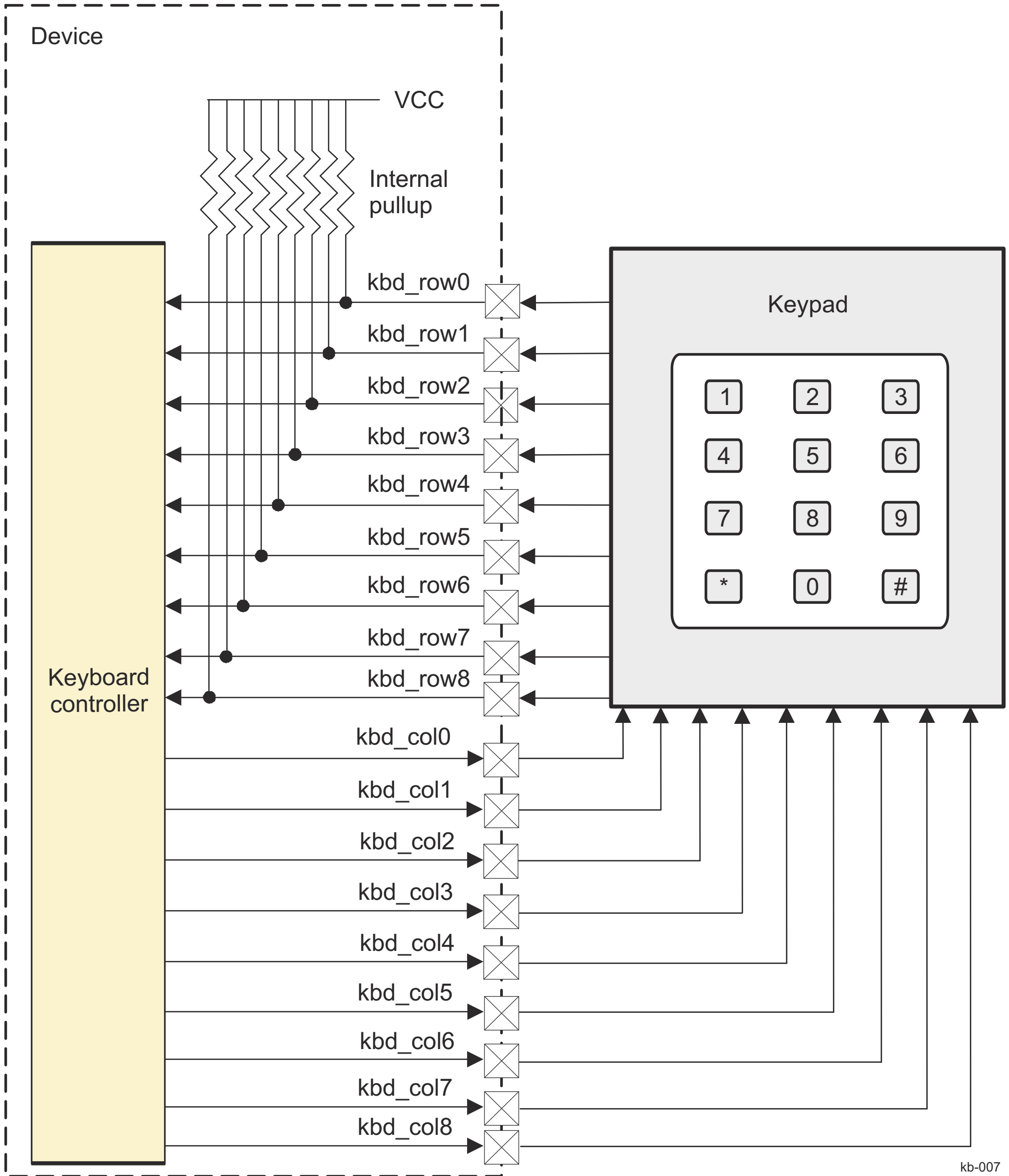


Figure 28-2. Typical Keyboard Environment

Note

To acheive the configuration shown in Figure 28-2 software needs to reconfigure the internal pulls on the kbd_* pads appropriately. The settings of those pads default to pull down resistors selected, software must reconfigure to pull ups on rows and to disable the pulls on columns.

28.2.1 Keyboard Controller Functions/Modes

The keyboard controller executes two functions:

- Keypad scanning and decoding for input of the external key presses. For more information, see Section 28.4.5, *Keyboard Controller Software Mode*, and Section 28.4.6, *Keyboard Controller Hardware Decoding Modes*.
- Device wakeup by interrupt request to the processor when a key is pressed and the device is in idle or sleep mode. For more information, see Section 28.4.6.4, *Keyboard Controller Interrupt Generation*.

28.2.2 Keyboard Controller Signals

Table 28-1 describes the module signals and specifies their links to functions.

Table 28-1. I/O External Keyboard Signals

Signal	I/O[1]	Description	Reset Value	Function	Wake-Up Capability
kbd_row0	I	Keypad row 0 feed	HiZ (pulled up)	Key reading	Yes
kbd_row1	I	Keypad row 1 feed	HiZ (pulled up)	Key reading	Yes
kbd_row2	I	Keypad row 2 feed	HiZ (pulled up)	Key reading	Yes
kbd_row3	I	Keypad row 3 feed	HiZ (pulled up)	Key reading	Yes
kbd_row4	I	Keypad row 4 feed	HiZ (pulled up)	Key reading	Yes
kbd_row5	I	Keypad row 5 feed	HiZ (pulled up)	Key reading	Yes
kbd_row6	I	Keypad row 6 feed	HiZ (pulled up)	Key reading	Yes
kbd_row7	I	Keypad row 7 feed	HiZ (pulled up)	Key reading	Yes
kbd_row8	I	Keypad row 8 feed	HiZ (pulled up)	Key reading	Yes
kbd_col0	O	Keypad column 0 feed, active low	HiZ	Key reading	No
kbd_col1	O	Keypad column 1 feed, active low	HiZ	Key reading	No
kbd_col2	O	Keypad column 2 feed, active low	HiZ	Key reading	No
kbd_col3	O	Keypad column 3 feed, active low	HiZ	Key reading	No
kbd_col4	O	Keypad column 4 feed, active low	HiZ	Key reading	No
kbd_col5	O	Keypad column 5 feed, active low	HiZ	Key reading	No
kbd_col6	O	Keypad column 6 feed, active low	HiZ	Key reading	No
kbd_col7	O	Keypad column 7 feed, active low	HiZ	Key reading	No
kbd_col8	O	Keypad column 8 feed, active low	HiZ	Key reading	No

(1) I = Input; O = Output

28.2.3 Protocols and Data Formats

The keyboard controller detects and decodes multikey combinations using the following rules:

- Any 2-key combination is valid and can be decoded.
- Combinations using more than two keys are valid only if the rows and columns used do not cross over on another key to be detected. This is caused by equipotent propagation on a row/column (multikey limitations).

Figure 28-3 shows an example of multikey limitation.

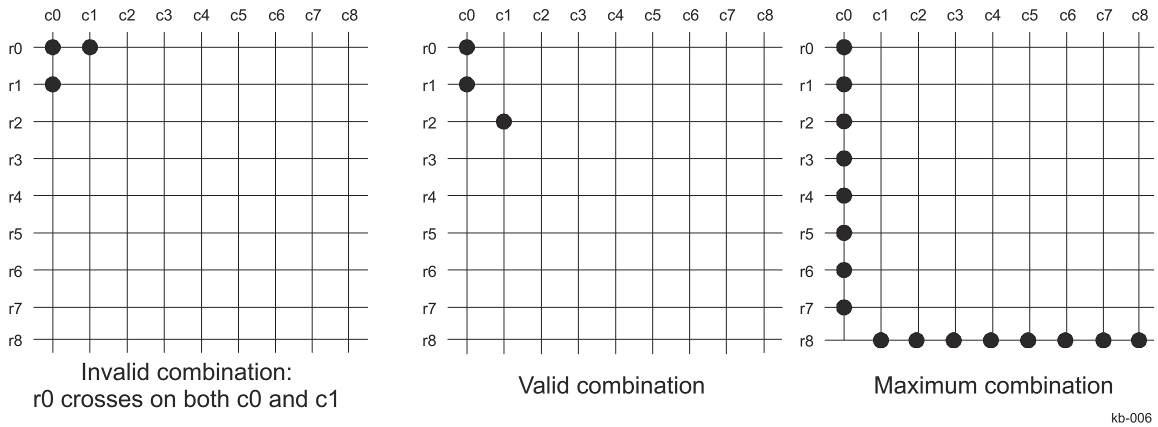


Figure 28-3. Multikey Limitation Example

Note

When using the keyboard controller with a smaller keypad (for example, 5 × 5), unused rows must be tied high to prevent disturbing the scanning process. Normally, all rows must be pulled up internally at the I/O cells of the device.

28.3 Keyboard Controller Integration

Figure 28-4 shows keyboard controller integration.

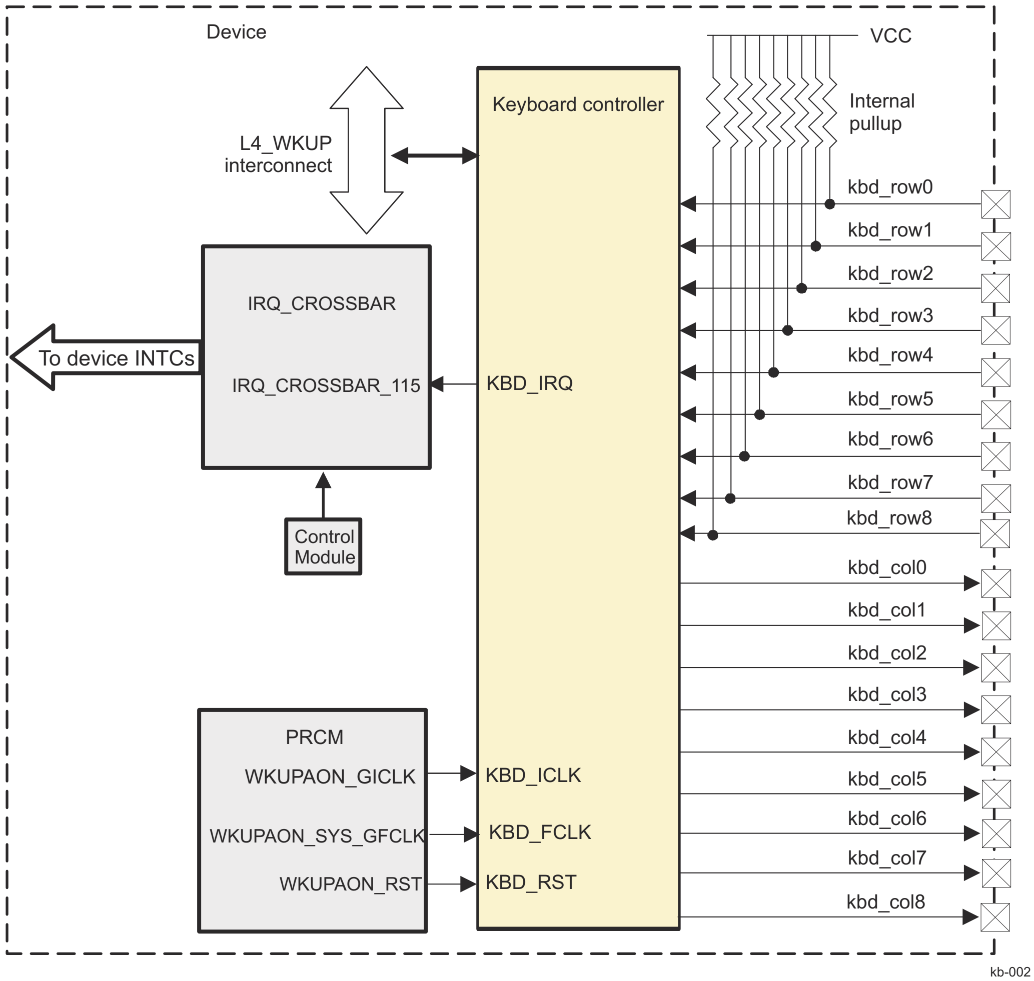


Figure 28-4. Keyboard Controller Integration

The control module must enable the internal pullups of the GPIO cells for all keypad rows (for more information about the configuration, see *Control Module*).

Table 28-2 through Table 28-4 summarize the integration of the module in the device.

Table 28-2. Keyboard Controller Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
KBD	PD_WKUPAON	Yes	L4_WKUP

Table 28-3. Keyboard Controller Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
KBD	KBD_FCLK	WKUPAON_SYS_GFCLK	PRCM	32-kHz functional clock. For information about PRCM clock gating and management, see *Clock Domain-Level Clock Management*, in *Power, Reset, and Clock Management*.

Table 28-3. Keyboard Controller Clocks and Resets (continued)

KBD	KBD_ICLK	WKUPAON_GICLK	PRCM	L4-interconnect interface clock. For information about PRCM clock gating and management, see *Clock Domain-Level Clock Management*, in *Power, Reset, and Clock Management*.
		Resets		
KBD	KBD_RST	WKUPAON_RST	PRCM	Reset signal for the Keyboard controller.

Table 28-4. Keyboard Controller Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
KBD	KBD_IRQ	IRQ_CROSSBAR_115	MPU_IRQ_120	Keyboard Controller interrupt request

Note

The “**Default Mapping**” column in Table 28-4 *Keyboard Controller Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.
For more information about the device interrupt controllers, see *Interrupt Controllers*.

Note

For the description of the interrupt source, see Section 28.4.4, *Interrupt Requests*.

28.4 Keyboard Controller Functional Description

28.4.1 Keyboard Controller Block Diagram

Figure 28-5 shows the functional specification block diagram of the keyboard controller.

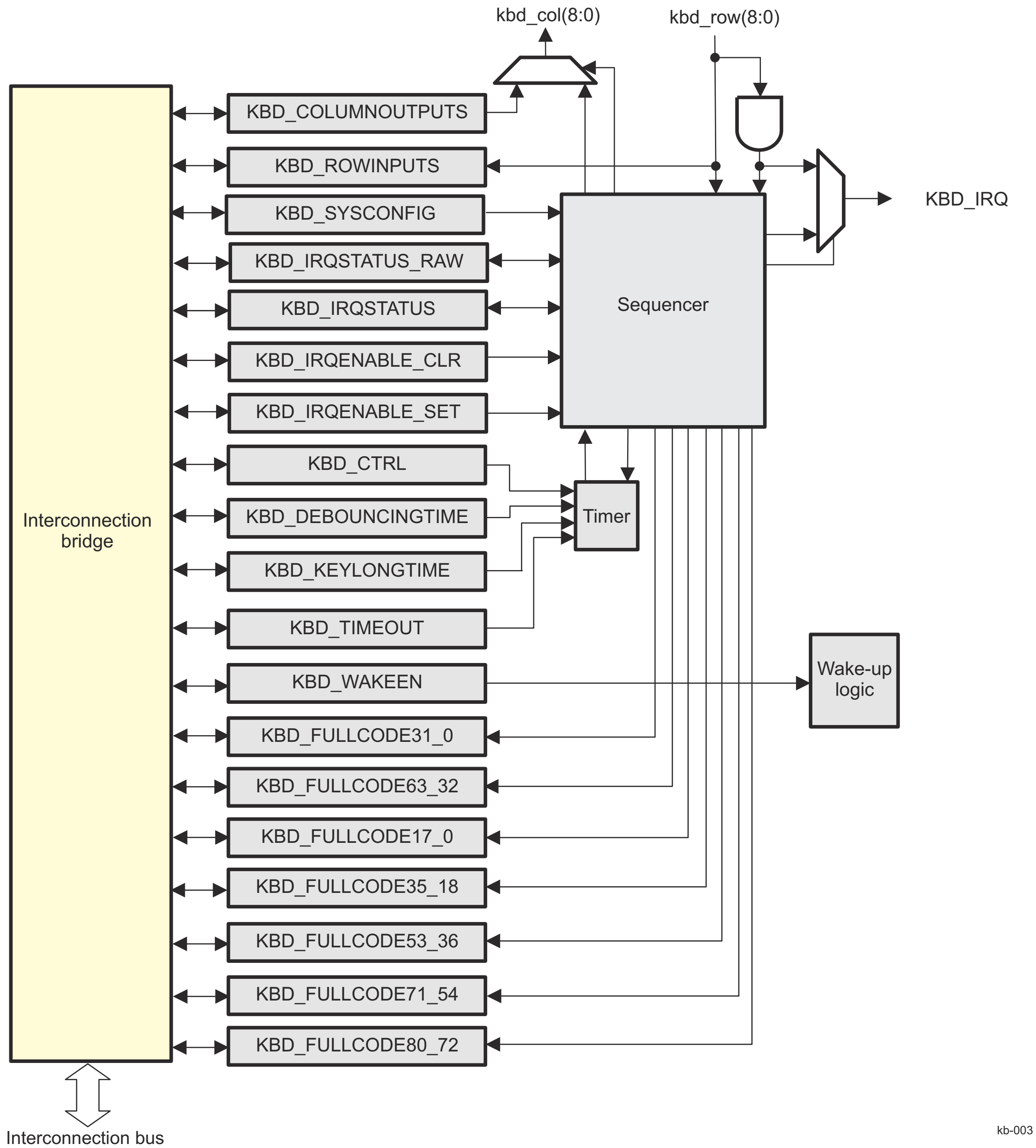


Figure 28-5. Keyboard Controller Block Diagram

The keyboard controller detects events issued on any key of the connected keyboard and generates an interrupt to alert the host processor. The built-in hardware-scan algorithm decodes the pressed keys, including multikey combinations.

To reduce MPU software overhead, the hardware performs detecting and decoding in the keyboard controller state-machine. However, hardware decoding can be deactivated so that software handles the scanning algorithm.

The value of the columns output is determined in the KBD_COLUMNOUTPUTS[8:0] KBC_REG bit field. To activate a keypad row-column connection, the corresponding bit must be 0b0.

The following sections describe subfunctions and subfunction interactions (control and data paths).

28.4.2 Keyboard Controller Software Reset

To perform a software reset, set the KBD_SYSCONFIG[1] SOFTRESET bit to 1. When the software reset completes, the KBD_SYSCONFIG[1] SOFTRESET bit is automatically reset. Software must ensure that the software reset completes before performing mailbox operations.

28.4.3 Keyboard Controller Power Management

Table 28-5 describes the power-management features available for the keyboard controller.

Note

For information about source clock gating and a description of the sleep/wake-up transitions, see *Clock Domain-Level Clock Management*, in *Power, Reset, and Clock Management*.

Table 28-5. Local Power-Management Features

Feature	Registers	Description
Slave idle modes	KBD_SYSCONFIG[4:3] IDLEMODE	Force-idle, no-idle, and smart-idle modes are available.
Master standby modes	N/A	N/A
Wake-up sources enable	KBD_IRQWAKEEN	This register holds one active-high enable bit per event source able to generate a wake-up signal.

28.4.4 Keyboard Controller Interrupt Requests

Table 28-6 lists the event flags, and their mask, that can cause module interrupts.

Table 28-6. Events

Event Flag	Event Mask	Description
KBD_IRQSTATUS[3] MISS_EVENT	–	A miss event occurs.
KBD_IRQSTATUS[2] IT_TIMEOUT	KBD_IRQENABLE_SET/KBD_IRQENABLE_CLR [2] IT_TIMEOUT_EN	A time-out event is detected.
KBD_IRQSTATUS[1] IT_LONG_KEY	KBD_IRQENABLE_SET/KBD_IRQENABLE_CLR [1] IT_LONG_KEY_EN	A long-key event is detected.
KBD_IRQSTATUS[0] IT_EVENT	KBD_IRQENABLE_SET/KBD_IRQENABLE_CLR [0] IT_EVENT_EN	An event is detected.

28.4.5 Keyboard Controller Software Mode

The KBD_CTRL[1] NSOFTWARE_MODE bit selects software mode when it is set to 0.

In software mode, the keyboard controller internal sequencer, which performs automatic scanning and decoding, is disabled. Consequently, software must manually perform the scanning algorithm.

The scanning sequence is managed using the keyboard controller column outputs register (KBD_COLUMNOUTPUTS) and the keyboard controller row inputs register (KBD_ROWINPUTS). In this configuration, the keyboard interrupt is a logical ANDing of all bits of the KBD_ROWINPUTS register.

For more information about the software scan, see Section 28.5.1, *Keyboard Controller Low-Level Programming Model*.

28.4.6 Keyboard Controller Hardware Decoding Modes

28.4.6.1 Functional Modes

When running in hardware (default) decoding mode (the KBD_CTRL[1] NSOFTWARE_ MODE bit is set to 1), the keyboard controller offers several functional modes; these modes are summarized in Table 28-7.

The keyboard interrupt depends on the configuration in the keyboard controller interrupt-enable register (KBD_IRQENABLE_SET). If the event is enabled (bit 0 is set to 1), an interrupt is generated when the sequencer detects an event. Even if this interrupt is disabled, the flag status of the keyboard controller interrupt-status register (KBD_IRQSTATUS) is updated.

Table 28-7. Keyboard Controller Functional Modes

Functional Mode	Associated Interrupt	Associated Timer Value	Description	Control
Keyboard event	Event interrupt	Debouncing value	Occurs when a key is pressed or released Always enabled	KBD_CTRL[8:5] bits must be set to 0 to disable all the other features.
Long key	Long-key interrupt	Long-key value	Used to detect a key that is pressed for a long time Should be associated with the long-key time-out function or repeat mode	KBD_CTRL[5] LONG_KEY
Repeat key	Long-key interrupt	Long-key value	Generates an interrupt every long-key delay No time-out can be associated.	KBD_CTRL[8] REPEAT_MODE
Empty time-out	Time-out interrupt	Empty time-out value	Interrupt generated if no key is pressed during an empty time-out period.	KBD_CTRL[6] TIMEOUT_EMPTY
Long-key time-out	Time-out interrupt	Long-key time-out value	Associated with the long-key function Generated after a long-key interrupt if no event occurs during a long-key time-out period	KBD_CTRL[7] TIMEOUT_LONG_KEY

Each mode can be activated/deactivated by setting the corresponding bits (5, 6, 7, and 8) in the KBD_CTRL register with the appropriate values (for more information, see Section 28.6, *Keyboard Controller Register Manual*).

28.4.6.2 Keyboard Controller Timer

As described in the previous section, each functional mode is associated with a timer value. Depending on the selected mode, the keyboard controller timer is loaded with the corresponding value as set in the related registers:

- KBD_DEBOUNCINGTIME
- KBD_KEYLONGTIME
- KBD_TIMEOUT

Table 28-8 summarizes the values of the keyboard controller timer.

Table 28-8. Keyboard Controller Timer Values

Timer Value	Associated Register Field	Description
Debouncing time	KBD_DEBOUNCINGTIME[5:0] DEBOUNCING_VALUE	To remove the effects of glitches when an event occurs on the keyboard, the controller waits for a debouncing period before taking a snapshot of the current state on the keyboard matrix. The timer is loaded with the debouncing time value after each detected event on the keyboard matrix. An event interrupt is generated after this delay.
Long-key time	KBD_KEYLONGTIME[11:0] LONG_KEY_VALUE	This is the delay before generating a long-key interrupt after an event interrupt. If the long-key mode is selected, the timer is loaded with the long-key time value after an event interrupt is generated. In repeat mode, the timer is reloaded with the same value after a long-key interrupt, and starts to count down again.
Long-key time-out	KBD_TIMEOUT[15:0] TIMEOUT_VALUE	The timer is loaded with the time-out value and then a long-key interrupt is generated and starts to count down. When it reaches 0, a time-out interrupt is generated and the keyboard controller returns to its IDLE state. This long-key time-out does not work in repeat mode.
Empty key time-out	KBD_TIMEOUT[15:0] TIMEOUT_VALUE	The time-out interrupt occurs if no key is pressed during this delay. The keyboard controller then returns to IDLE state.

The timer countdown period depends on three factors:

- The loaded value as set in:
  - KBD_DEBOUNCINGTIME
  - KBD_KEYLONGTIME
  - KBD_TIMEOUT
- The value of the prescale clock timer as set in the KBD_CTRL[4:2] PTV bit field. This programmable clock divider allows the reduction of the clock frequency used by the timer.
- The frequency of the keyboard controller functional clock (32 kHz). This clock is actually either 32K oscillator, or it is the SYSCLK1/610, depending on sysboot[9:8] state at power-on. For more information about sysboot states see *Initialization*.

The period is calculated as follows:

$T_{period} = (T_{value} + 1) \times 2^{PTV + 1} \times T_{clk}$

Where:

T_{value} is the value stored in the KBD_DEBOUNCINGTIME, KBD_KEYLONGTIME, or KBD_TIMEOUT register.

PTV is the value of the KBD_CTRL[4:2] PTV bit field.

T_{clk} is the period of the 32-kHz functional clock; that is, 31.25 μs or other (depending wheather the 32K oscillator or the SYSCLK1/610 clock is selected as source).

The KBD_CTRL[4:2] PTV bit field determines the division factor of the timer clock. Table 28-9 lists the divider rates.

Table 28-9. Timer Prescale Values

KBD_CTRL[4:2] PTV Value	Divisor
0	2
1	4
2	8
3	16
4	32
5	64
6	128
7	256

Note

The timer minimum period is 62.5 μs; its maximum period is 524.288 seconds.

CAUTION

To prevent undefined results, the KBD_CTRL[4:2] PTV bit field must not be changed when the timer is running.

The timer value registers (KBD_DEBOUNCINGTIME, KBD_KEYLONGTIME, and KBD_TIMEOUT) can be updated at any time, whether or not the timer is running. Nevertheless, the timer is updated only on the fly for the long-key time value. The new debouncing and time-out values are loaded only on the next load. Depending on the updated register, two cases can occur:

- The KBD_KEYLONGTIME register is updated; the new value stored in the KBD_KEYLONGTIME register is loaded into the timer when the register is written. If the timer is already counting down when KBD_KEYLONGTIME is updated, it counts down from the new value loaded in KBD_KEYLONGTIME.
- The KBD_DEBOUNCINGTIME or KBD_TIMEOUT register is updated; the new value is considered only when the next timer loads. If the timer is counting down when the registers are updated, the timer continues counting down from the previous value and is loaded with the new one on the next load.

Regardless of the timer state (stopped, counting down any of the values previously described, etc.), when a new event occurs on the keyboard, the timer is stopped and loaded with the debouncing time value. It then starts counting down.

28.4.6.3 State-Machine Status

To facilitate debugging, each state of the state-machine is coded in a register that indicates the current state of the machine. Table 28-10 lists the corresponding codes.

Table 28-10. State-Machine Values

KBD_STATEMACHINE[3:0] Value	Description
0x0	Idle
0x1	Scanning
0x2	Load timer debouncing
0x3	Test timer debouncing
0x4	Generated interrupt event
0x6	Load timer long key
0x7	Test timer long key
0x8	Generated interrupt long key
0x9	Load timer time-out
0xA	Test timer time-out
0xB	Generated interrupt time-out
0xF	Other

28.4.6.4 Keyboard Controller Interrupt Generation

28.4.6.4.1 Interrupt-Generation Scheme

The keyboard controller generates the KBD_IRQ interrupt signal connected to the IRQ_CROSSBAR_115 input. Each functional mode generates dedicated interrupt events (logged in the KBD_IRQSTATUS register) that can be masked using the KBD_IRQENABLE_CLR register or unmasked using the KBD_IRQENABLE_SET register.

The KBD_IRQSTATUS register is updated when the selected functional mode generates an interrupt event. However, the KBD_IRQ signal is asserted on the related event only if the corresponding bit is set to 0x1 in the KBD_IRQENABLE_SET register.

Note

To reset the interrupt status bit, 1 must be written to the appropriate bit of the KBD_IRQSTATUS read/write register.

Figure 28-6 shows the different interrupt events generated in each keyboard controller functional mode and details the relationships between them.

Note

Depending on the selected mode, some interrupt events cannot be generated.

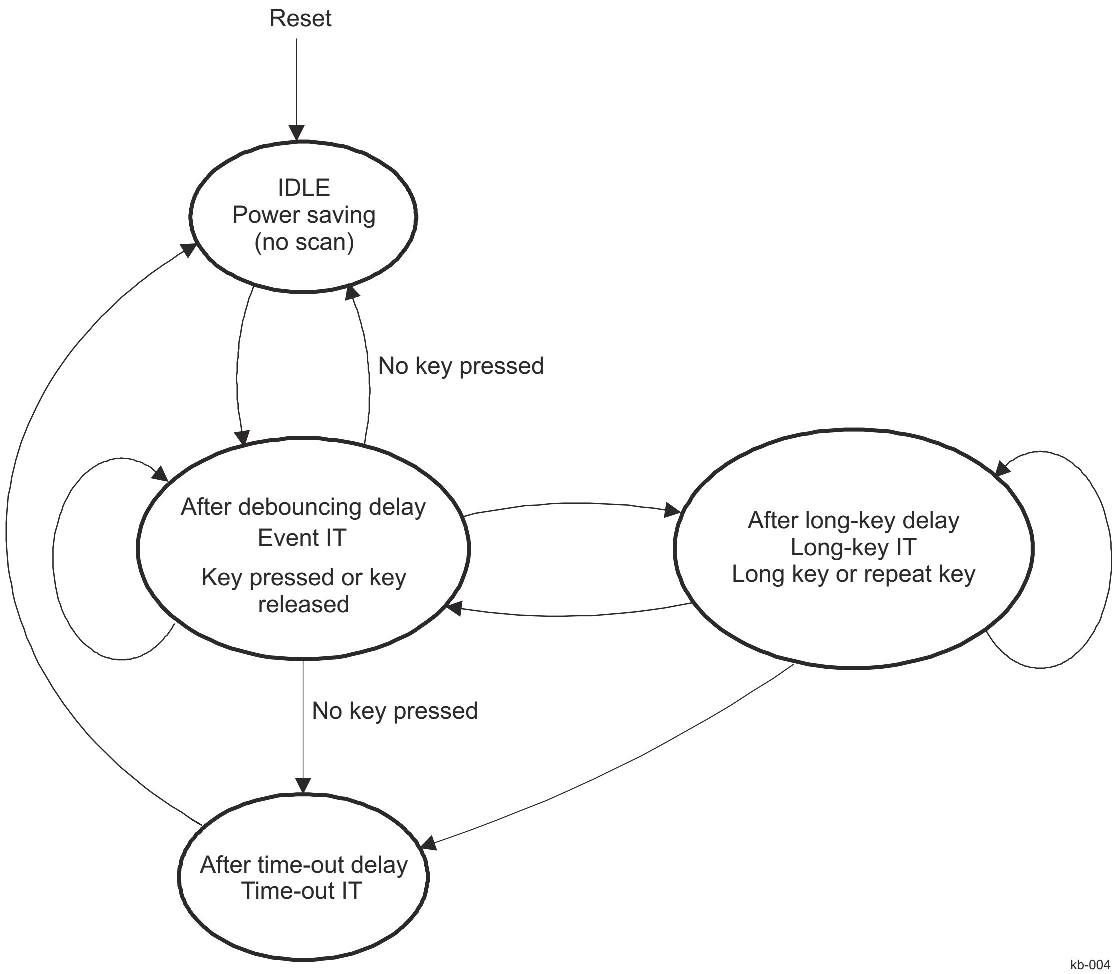


Figure 28-6. Functional Modes and Related Interrupt Events

While running in hardware-decoding mode, the keyboard controller performs automatic scans when not in IDLE state. When a key-press event occurs on the keyboard matrix, the keyboard controller leaves IDLE state and an interrupt event (the KBD_IRQSTATUS[0] IT_EVENT bit) is set after the timer counts down the debouncing delay. An IT_EVENT is generated when a key is pressed or released.

Note

An IT_EVENT is generated regardless of the selected functional mode. If no time-out is set and no more keys are pressed, the keyboard controller returns to IDLE state.

If long-key detection mode is set when the timer counts down the long-key delay, an interrupt long key (the KBD_IRQSTATUS[1] IT_LONG_KEY bit) is generated. If the repeat mode is set, the IT_LONG_KEY interrupt is generated periodically every long-key delay.

A time-out can also be set in event detection or long-key detection mode. In this case, a time-out interrupt (the KBD_IRQSTATUS[2] IT_TIMEOUT bit) is generated after the time-out delay timer expires. After such an interrupt, the keyboard controller always returns to IDLE state.

Note

No time-out can be set in repeat mode. Only a keyboard event can stop the periodic interrupt generation.

28.4.6.4.2 Keyboard Buffer and Missed Events (Overrun Feature)

The keyboard controller has an overrun feature: A dedicated buffer allows the keyboard controller to memorize two successive events. If two successive events occur before a read is performed, the second event is stored and a second interrupt is generated when the first interrupt is cleared, allowing two consecutive key events to be received.

If a third event occurs before the first event is treated, a missed event interrupt (the KBD_IRQSTATUS[3] MISS_EVENT bit) is generated to report the lost event.

Note

The MISS_EVENT interrupt is not routed to the KBD_IRQ signal; software must check the KBD_IRQSTATUS[3] MISS_EVENT bit to detect any missed events.

28.4.7 Keyboard Controller Key Coding Registers

The keyboard controller matrix pressed keys state (indicating which columns/rows are connected) is reflected in the KBD_FULLCODE17_0 to KBD_FULLCODE80_72 registers, as shown in Figure 28-7. These registers are updated only when interrupt event status is inactive to prevent a lost event.

Note

The KBD_FULLCODE31_0 and KBD_FULLCODE63_32 registers, which are limited to supporting keyboards of a maximum 8 × 8 array size, can be used for back-to-back software compatibility.

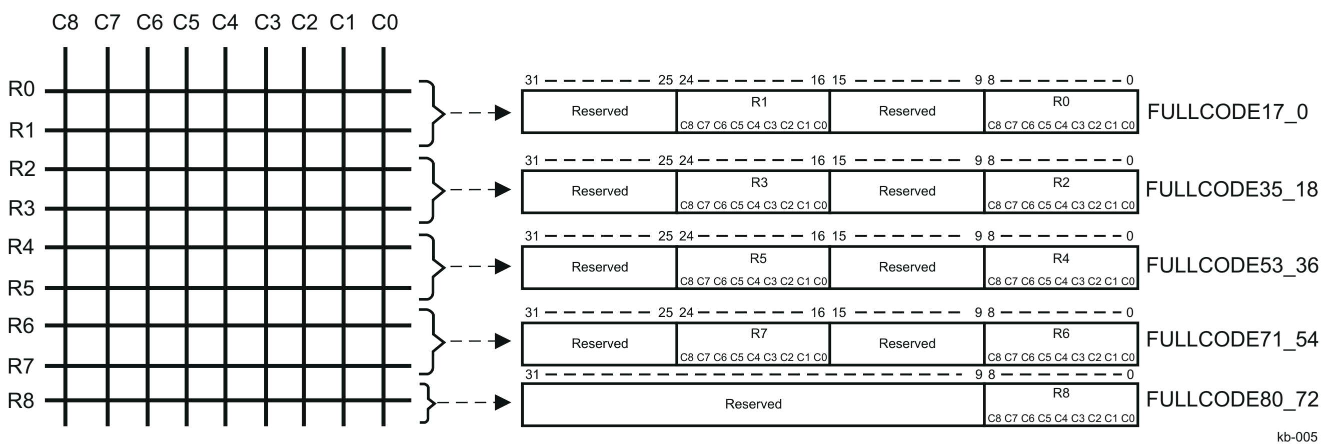


Figure 28-7. Key Coding Registers

Each of the 9 × 9 keyboard size supporting registers (KBD_FULLCODE17_0 to KBD_FULLCODE71_54) stores the state of two keyboard rows. The KBD_FULLCODE80_72 register stores the state of the last row, R8:

- The KBD_FULLCODE17_0 register code rows 0, 1 (row 0 is coded between bits 0 and 8, row 1 is coded between bits 16 and 24)
- The KBD_FULLCODE35_18 register code rows 2, 3 (row 2 is coded between bits 0 and 8, row 3 is coded between bits 16 and 24)
- The KBD_FULLCODE53_36 register code rows 4, 5 (row 4 is coded between bits 0 and 8, row 5 is coded between bits 16 and 24)
- The KBD_FULLCODE71_54 register code rows 6, 7 (row 6 is coded between bits 0 and 8, row 7 is coded between bits 16 and 24)
- The KBD_FULLCODE80_72 register code row 8 (row 8 is coded between bits 0 and 8)

In each of these registers (excluding KBD_FULLCODE80_72):

- Bit 0 corresponds to column(0) – row (i) key, … , bit 8 corresponds to column(8) – row (i) key.

- Bits from 9 to 15 are reserved.
- Bit 16 corresponds to column(0) – row (i + 1) key, … , bit 24 corresponds to column (8) – row (i + 1) key.
- Bits 31 to 25 are reserved, where i = 0, 2, 4, 6.

In the KBD_FULLCODE80_72 register, bit 0 corresponds to column (0) – row (8) key, …, the bit 8 corresponds to column (8) – row (8) key, and the remaining bits (from 9 to 31) are reserved.

Each of the 8 × 8-keyboard size supporting registers (KBD_FULLCODE31_0 and KBD_FULLCODE63_32) stores the state of four keyboard rows:

- The KBD_FULLCODE31_0 register code rows 0, 1, 2, and 3 (row 0 is coded between bits 0 and 7, row 1 is coded between bits 8 and 15, row 2 is coded between bits 16 and 23, and row 3 is coded between bits 24 and 31)
- The KBD_FULLCODE63_32 register code rows 4, 5, 6, and 7 (row 4 is coded between bits 0 and 7, row 5 is coded between bits 8 and 15, row 6 is coded between bits 16 and 23, and row 7 is coded between bits 24 and 31)

In these registers:

- KBD_FULLCODE31_0[0] corresponds to column (0) – row (0) key,…, KBD_FULLCODE31_0[7] corresponds to column (7) – row (0).
- KBD_FULLCODE31_0[8] corresponds to column (0) – row (1) key,..., KBD_FULLCODE31_0[15] corresponds to column (7) – row (1).
- KBD_FULLCODE31_0[16] corresponds to column (0) – row (2) key,..., KBD_FULLCODE31_0[23] corresponds to column (7) – row (2).
- KBD_FULLCODE31_0[24] corresponds to column (0) – row (3) key,..., KBD_FULLCODE31_0[31] corresponds to column (7) – row (3).
- KBD_FULLCODE63_32[0] corresponds to column (0) – row (4) key,..., KBD_FULLCODE63_32[7] corresponds to column (7) – row (4).
- KBD_FULLCODE63_32[8] corresponds to column (0) – row (5) key,..., KBD_FULLCODE63_32[15] corresponds to column (7) – row (5).
- KBD_FULLCODE63_32[16] corresponds to column (0) – row (6) key,..., KBD_FULLCODE63_32[23] corresponds to column (7) – row (6).
- KBD_FULLCODE63_32[24] corresponds to column (0) – row (7) key,..., KBD_FULLCODE63_32[31] corresponds to column (7) – row (7).

Note

The keyboard fullcode registers are not updated in software mode.

Note

When using a smaller keyboard (for example, 5 × 5 or 4 × 4), the bits of the unused columns/rows are not used.

28.4.8 Keyboard Controller Register Access

28.4.8.1 Write Registers Access

The keyboard module uses a posted-write scheme to update any internal registers. This means the write transaction is immediately acknowledged on the L4 interface, although the effective write operation occurs later due to a resynchronization in the functional clock domain. This has the advantage of not stalling the interconnect system or the CPU that requested the write transaction. For each functional register, a pending bit is provided that is set if there is a pending write access to this register. The pending bits are accessible in the keyboard pending write register (KBD_PENDING).

In this mode, it is mandatory that the CPU checks the pending bits before any write access in the functional registers. If a write is attempted to a register with a previous access pending, the previous access is discarded without notice (this can also lead to unexpected results).

A register read following a posted write (on the same register) may not read the previous write value if the write posted process is not complete. Software synchronization must be used to avoid noncoherent read.

This posted period is defined as the interval between the posted write access request and the reset of the pending bit in the KBD_PENDING register, and can be quantified:

T (reset posted bit maximum) = 3 × Ticlk + 3 × Tfuncclk

The time it takes to accomplish the writing is:

T (write accomplish maximum) = 1 × Ticlk + 3 × Tfuncclk

where:

Ticlk is the L4 interface clock period, and Tfuncclk is the functional clock period.

28.4.8.2 Read Registers Access

The keyboard module uses a posted-read scheme for reading any internal register. The read transaction is immediately acknowledged on the L4 interface. The value of the functional register to be read must be previously synchronized. This has the advantage of not stalling the interconnect system or the CPU that requested the read transaction.

The posted-read scheme can be used only if Freq(KBD_FCLK) < Freq(KBD_ICLK)/4.

28.5 Keyboard Controller Programming Guide

28.5.1 Keyboard Controller Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

28.5.1.1 Global Initialization

28.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the keyboard module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the keyboard. For more information, see Section 28.2, *Keyboard Controller Environment,* and Section 28.3, *Keyboard Controller Integration*.

Table 28-11 describes the global initialization of surrounding modules.

Table 28-11. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. See *Power, Reset, and Clock Management*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow the keyboard controller IRQ to be mapped to certain device INTC line. For more information see Section 18.4.6.4, IRQ_CROSSBAR Module Functional Description, in *Control Module*.
Interrupt controllers	Device INTCs must be configured to enable the interrupt request generation. For more information about enabling interrupts, see Chapter 17, Interrupt Controllers.
Control module	The pad configuration registers must be configured to map the keyboard interface signals to the device pads, to determine the signal directions and for all keyboard rows to enable the internal pull-ups of the associated IO cells . For more information about this configuration, see *Pad Configuration Registers.*

28.5.1.1.2 Keyboard Controller Global Initialization

28.5.1.1.2.1 Main Sequence – Keyboard Controller Global Initialization

This procedure initializes the keyboard controller after a POR or software reset (see Table 28-12).

Table 28-12. Keyboard Controller Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure the debouncing time of filtering the glitches on pressing or releasing key.	KBD_DEBOUNCINGTIME[5:0] DEBOUNCING_VALUE	0x–
Configure the duration of a key press to allow shortcut detection (desired value of the long-key interrupt or repeat mode value).	KBD_KEYLONGTIME[11:0] LONG_KEY_VALUE	0x–
Configure the period of the long inactivity on the keyboard (desired value of the time-out interrupt).	KBD_TIMEOUT[15:0] TIMEOUT_VALUE	0x–
Define the logical value of the column outputs (KEYPAD configuration) (logical 0 bit = active column-row).	KBD_COLUMNOUTPUTS[8:0] KBC_REG	0x–
Perform the functional configuration of the keyboard module and the prescale clock timer value.	KBD_CTRL	0x00000–
Clear the interrupt-status register.	KBD_IRQSTATUS	0x0000000F
Enable (0b1)/disable (0b0) certain keyboard events for generating an interrupt request.	KBD_IRQENABLE_SET/KBD_IRQENABLE_CLR [2:0] IT_..._EN	0x–
Unmask (0b1)/mask (0b0) the expected source of wake-up event that generates a wake-up request.	KBD_IRQWAKEEN[2:0] WUP_..._ENA	0x–

28.5.1.2 Operational Modes Configuration

28.5.1.2.1 Keyboard Controller in Hardware Decoding Mode (Default Mode)

28.5.1.2.1.1 Main Sequence – Keyboard Controller Hardware Mode

After reset, all available functional modes are disabled, except detect-event mode, which is always active. Table 28-13 describes the keyboard controller hardware mode.

Table 28-13. Keyboard Controller Hardware Mode

Step	Register/Bit Field/Programming Model	Value
Activate the internal keyboard controller sequencer by setting the bit.	KBD_CTRL[1] NSOFTWARE_MODE	0b1
Select the functional mode by setting its corresponding bit to 1.	KBD_CTRL[8:5]	0x–
Configure the duration of a key press to allow shortcut detection (desired value of the long-key interrupt or repeat mode value).	KBD_KEYLONGTIME[11:0] LONG_KEY_VALUE	0x–
Configure the period of the long inactivity on the keyboard (desired value of the time-out interrupt).	KBD_TIMEOUT[15:0] TIMEOUT_VALUE	0x–
Configure the debouncing time of filtering the glitches on pressing or releasing key.	KBD_DEBOUNCINGTIME[5:0] DEBOUNCING_VALUE	0x–
Clear the interrupt-status register.	KBD_IRQSTATUS	0x0000000F
Enable (by writing 1)/disable (by writing 0) certain keyboard event for generating an interrupt request	KBD_IRQENABLE_SET or KBD_IRQENABLE_CLR[2:0] IT_..._EN	0x–
Unmask (0b1)/mask (0b0) the expected source of wake-up event that generates a wake-up request.	KBD_IRQWAKEEN[2:0] WUP_..._ENA	0x–
Wait for the KBD_IRQ interrupt signal assertion.		
Read the interrupt-status register to determine which event caused the interrupt.	KBD_IRQSTATUS	
Read the KBD_FULLCODE17_0 to KBD_FULLCODE80_72 registers (or 8 × 8 keyboard-size-supporting KBD_FULLCODE31_0 and KBD_FULLCODE63_32 registers) to determine which key matrix combination was pressed.	KBD_FULLCODE17_0 [ROWi bits] (where i = 0 or 1) up to KBD_FULLCODE71_54 [ROWi bits] (where i = 6 or 7); KBD_FULLCODE80_72 [ROWi bits] (where i = 8); (or KBD_FULLCODE31_0[j] FULL_CODE_31_0 and KBD_FULLCODE63_32[k] FULL_CODE_63_32 bits (where j = 0 to 31, k = 32 to 63)	
Clear the corresponding bit(s) in the interrupt-status register by writing logical 1.	KBD_IRQSTATUS	0x1

Note

The long-key detection mode and the repeat mode cannot be used simultaneously, because they share the same interrupt status bit and are mutually exclusive. Software must ensure that only one of these modes at a time is selected.

Note

All interrupts are disabled on reset.

Note

When two events occur successively before the first event is read, the second interrupt is generated when the first interrupt is cleared. When more than two events in a row occur, the KBD_IRQSTATUS[3] MISS_EVENT bit is set. Software must check this bit, which is not reflected on the KBD_IRQ line.

Note

The keyboard controller uses a posted-write scheme to update any internal register. Software must read the pending write status bits to ensure that the next write access is not discarded because of ongoing write synchronization. For more information, see Section 28.4.8.1, *Write Registers Access*.

28.5.1.2.2 Keyboard Controller Software Scanning Mode

28.5.1.2.2.1 Main Sequence – Keyboard Controller Software Mode

Table 28-14 describes the keyboard controller software mode.

Table 28-14. Keyboard Controller Software Mode

Step	Register/Bit Field/Programming Model	Value D
Deactivate the internal keyboard controller sequencer by clearing the bit.	KBD_CTRL[1] NSOFTWARE_MODE	0b0
Enable the interrupt event by setting the bit.	KBD_IRQENABLE_SET[0] IT_EVENT_EN	0b1
Wait for the KBD_IRQ interrupt signal assertion. Begin the software scan when the interrupt signal is asserted:		
1) Disable all columns to drive a logical 0 on the kbd_col[8:0] output by writing 0xFF. (logical 1 bit = inactive column-row)	KBD_COLUMNOUTPUTS[8:0] KBC_REG	0xFF
2) Drive kbd_col(i) output, (where i = 0 to 8), to 0 to capture a pressed-key-event, at the corresponding row input.	KBD_COLUMNOUTPUTS[i] KBC_REG (where i = 0 to 8)	0b0
3) Read the KBR_LATCH bit field of the KBD_ROWINPUTS register to determine which is the pressed key. IF: KBR_LATCH k-bit = 0 , where k = 0 to 8 Then, the corresponding k-row is connected to the column being enabled.	KBD_ROWINPUTS[8:0] KBR_LATCH	
END IF		
Repeat step 2) and step 3) for all existing columns.	KBD_IRQWAKEEN[2:0] WUP_..._ENA	0x–

Note

In software mode, during the manual keyboard scan, an interrupt is generated when a pressed key is detected.

28.5.1.2.3 Using the Timer

For information about programming the keyboard controller timer, see Section 28.4.6.2, *Keyboard Controller Timer*.

28.5.1.2.4 State-Machine Status Register

To see the state of the state-machine, see Section 28.4.6.3, *State-Machine Status*.

28.5.1.3 Keyboard Controller Events Servicing

Table 28-15 lists the keyboard controller event servicing.

Table 28-15. Keyboard Controller Event Servicing

Step	Register/Bit Field/Programming Model	Value
Clear all eventual previous indications of treated interrupts; write the IRQSTATUS register:	KBD_IRQSTATUS	0x0000000F
Enable the desired sources of interrupt by writing in:	KBD_IRQENABLE	0x0000000–

Table 28-15. Keyboard Controller Event Servicing (continued)

Step	Register/Bit Field/Programming Model	Value
Unmask (enable) the desired sources of interrupt by writing logical 1 in the desired bit fields.	KBD_IRQWAKEEN	0x0000000–
When event occurs:		
Read the status register.	KBD_IRQSTATUS	
IF: KBD_IRQSTATUS[3] MISS_EVENT = 1	Start the interrupt handler for missed event.	
ELSIF: KBD_IRQSTATUS[2] IT_TIMEOUT = 1	Start the interrupt handler for timeout event.	
ELSIF: KBD_IRQSTATUS[1] IT_LONG_KEY = 1	Start the interrupt handler for long key pressed event.	
ELSIF: KBD_IRQSTATUS[0] IT_EVENT = 1	Start the interrupt handler for pressed key event.	
ENDIF		
Wait for the execution of the corresponding interrupt handler.	Interrupt controller of the MPU	
Clear the corresponding IRQ status register bits by writing 1 there.	KBD_IRQSTATUS	0x0000000–
Disable the corresponding sources of interrupt if needed by writing 0 at the corresponding bit fields.	KBD_IRQENABLE or KBD_IRQWAKEEN	0x0000000-
Read the status register.	KBD_IRQSTATUS	
IF :	KBD_IRQSTATUS[2] IT_TIMEOUT	=1
Handle the timeout event.		
Clear the corresponding status flag.	KBD_IRQSTATUS[2] IT_TIMEOUT	0x1
ELSE IF:	KBD_IRQSTATUS[1] IT_LONG_KEY	=1
Handle the long pressed key event.		
Clear the corresponding status flag.	KBD_IRQSTATUS[1] IT_LONG_KEY	0x1
ELSE IF:	KBD_IRQSTATUS[0] IT_EVENT	=1
Handle the pressed key event.		
Clear the corresponding status flag.	KBD_IRQSTATUS[0] IT_EVENT	0x1
ENDIF		
Disable the interrupts if needed.	KBD_IRQENABLE_CLR[2:0]	0x1

28.6 Keyboard Controller Register Manual

28.6.1 Keyboard Controller Instance Summary

Table 28-16 summarizes the keyboard controller instances.

Table 28-16. Keyboard Controller Instance Summary

Module Name	Base Address	Size
KBD	0x4AE1 C000	112 Bytes

28.6.2 Keyboard Controller Registers

28.6.2.1 Keyboard Controller Register Summary

Table 28-17 summarizes the keyboard controller register mapping.

Table 28-17. Keyboard Controller Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	KBD Physical Address
KBD_REVISION	R	32	0x0000 0000	0x4AE1 C000
KBD_SYSCONFIG	RW	32	0x0000 0010	0x4AE1 C010
KBD_EOI	RW	32	0x0000 001C	0x4AE1 C01C
KBD_IRQSTATUS_RAW	RW	32	0x0000 0020	0x4AE1 C020
KBD_IRQSTATUS	RW	32	0x0000 0024	0x4AE1 C024
KBD_IRQENABLE_SET	RW	32	0x0000 0028	0x4AE1 C028
KBD_IRQENABLE_CLR	RW	32	0x0000 002C	0x4AE1 C02C
KBD_IRQWAKEEN	RW	32	0x0000 0030	0x4AE1 C030
KBD_PENDING	R	32	0x0000 0034	0x4AE1 C034
KBD_CTRL	RW	32	0x0000 0038	0x4AE1 C038
KBD_DEBOUNCINGTIME	RW	32	0x0000 003C	0x4AE1 C03C
KBD_KEYLONGTIME	RW	32	0x0000 0040	0x4AE1 C040
KBD_TIMEOUT	RW	32	0x0000 0044	0x4AE1 C044
KBD_STATEMACHINE	R	32	0x0000 0048	0x4AE1 C048
KBD_ROWINPUTS	R	32	0x0000 004C	0x4AE1 C04C
KBD_COLUMNOUTPUTS	RW	32	0x0000 0050	0x4AE1 C050
KBD_FULLCODE31_0	R	32	0x0000 0054	0x4AE1 C054
KBD_FULLCODE63_32	R	32	0x0000 0058	0x4AE1 C058
KBD_FULLCODE17_0	R	32	0x0000 005C	0x4AE1 C05C
KBD_FULLCODE35_18	R	32	0x0000 0060	0x4AE1 C060
KBD_FULLCODE53_36	R	32	0x0000 0064	0x4AE1 C064
KBD_FULLCODE71_54	R	32	0x0000 0068	0x4AE1 C068
KBD_FULLCODE80_72	R	32	0x0000 006C	0x4AE1 C06C

28.6.2.2 Keyboard Controller Register Description

Table 28-18. KBD_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4AE1 C000	**Instance**	KBD
Description	This register contains the IP revision code. A write to this register has no effect.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															

Bits	Field Name	Description	Type	Reset
31: 0	Reserved	IP Revision	R	0x1

Table 28-19. KBD_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4AE1 C010	**Instance**	KBD
Description	This register controls the various parameters of the OCP interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										EMUFREE	IDLEMODE		RESERVED	SOFTRESET	RESERVED

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reads return 0.	R	0x000 0000
5	EMUFREE	Emulation mode 0x0: The KBDOCP module is frozen in emulation mode (PINSUSPENDN signal active). 0x1: The KBDOCP module runs free, regardless of PINSUSPENDN value.	RW	0
4:3	IDLEMODE	Power Management, req/ack control 0x0: Force-idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged. 0x3: Reserved. Do not use. 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module.	RW	0x0
2	RESERVED	Reads return 0.	R	0
1	SOFTRESET	Software reset. Write: initiate software reset Read: Reset done (0) / Reset ongoing (1) 0x0: Normal mode 0x1: The module is reset	RW	0
0	RESERVED	Reads return 0.	R	0

Table 28-20. KBD_EOI

Address Offset	0x0000 001C		
Physical Address	0x4AE1 C01C	**Instance**	KBD
Description	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if an new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0.	R	0x0
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write number of interrupt output. 0x0: No event. 0x1: An event occurs.	RW	0x0

Table 28-21. KBD_IRQSTATUS_RAW

Address Offset	0x0000 0020		
Physical Address	0x4AE1 C020	**Instance**	KBD
Description	Per-event raw interrupt status vector Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MISS_EVENT	IT_TIMEOUT	IT_LONG_KEY	IT_EVENT

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0	R	0x000 0000
3	MISS_EVENT	IRQ status for Miss event Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Trigger IRQ event by software	RW	0
2	IT_TIMEOUT	IRQ status for Timeout Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Trigger IRQ event by software	RW	0
1	IT_LONG_KEY	IRQ status for Long key Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Trigger IRQ event by software	RW	0
0	IT_EVENT	IRQ status for Event Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Trigger IRQ event by software	RW	0

Table 28-22. KBD_IRQSTATUS

Address Offset	0x0000 0024		
Physical Address	0x4AE1 C024	**Instance**	KBD

Table 28-22. KBD_IRQSTATUS (continued)

Description	Per-event "enabled" interrupt status vector. Enabled status isn't set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MISS_EVENT	IT_TIMEOUT	IT_LONG_KEY	IT_EVENT

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0	R	0x000 0000
3	MISS_EVENT	IRQ status for Miss event Read always returns zero Write 0 : No action Write 1 : Clear pending event, if any	RW	0
2	IT_TIMEOUT	IRQ status for Timeout Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Clear pending event, if any	RW	0
1	IT_LONG_KEY	IRQ status for Long key Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Clear pending event, if any	RW	0
0	IT_EVENT	IRQ status for Event Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Clear pending event, if any	RW	0

Table 28-23. KBD_IRQENABLE_SET

Address Offset	0x0000 0028		
Physical Address	0x4AE1 C028	**Instance**	KBD
Description	Per-event interrupt enable bit vector Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													IT_TIMEOUT_EN	IT_LONG_KEY_EN	IT_EVENT_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0	R	0x0000 0000

Bits	Field Name	Description	Type	Reset
2	IT_TIMEOUT_EN	IRQ enable for Timeout Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Set IRQ enable	RW	0
1	IT_LONG_KEY_EN	IRQ enable for Long key Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Set IRQ enable	RW	0
0	IT_EVENT_EN	IRQ enable for Event Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Set IRQ enable	RW	0

Table 28-24. KBD_IRQENABLE_CLR

Address Offset	0x0000 002C		
Physical Address	0x4AE1 C02C	**Instance**	KBD
Description	Per-event interrupt enable bit vector Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
Type	RW		

31–3	2	1	0
RESERVED	IT_TIMEOUT_EN	IT_LONG_KEY_EN	IT_EVENT_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0	R	0x0000 0000
2	IT_TIMEOUT_EN	IRQ enable for Timeout Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Clear IRQ enable	RW	0
1	IT_LONG_KEY_EN	IRQ enable for Long key Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Clear IRQ enable	RW	0
0	IT_EVENT_EN	IRQ enable for Event Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Clear IRQ enable	RW	0

Table 28-25. KBD_IRQWAKEEN

Address Offset	0x0000 0030		
Physical Address	0x4AE1 C030	**Instance**	KBD
Description	The Keyboard Wake-up Enable Register allows the user to mask the expected source of wake-up event that will generate a wake-up request. The KBD_IRQWAKEEN is programmed synchronously with the interface clock before any idle mode request comes from the host processor.		

Table 28-25. KBD_IRQWAKEEN (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													WUP_TIMEOUT_ENA	WUP_LONG_KEY_ENA	WUP_EVENT_ENA

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0.	R	0x0000 0000
2	WUP_TIMEOUT_ENA	Timeout wakeup enable. 0x0: Timeout wakeup generation disabled. 0x1: Timeout wakeup generation enabled.	RW	0
1	WUP_LONG_KEY_ENA	Long key wakeup enable. 0x0: Long key wakeup generation disabled. 0x1: Long key wakeup generation enabled.	RW	0
0	WUP_EVENT_ENA	Event wakeup enable. 0x0: Event wakeup generation disabled. 0x1: Event wakeup generation enabled.	RW	0

Table 28-26. KBD_PENDING

Address Offset	0x0000 0034		
Physical Address	0x4AE1 C034	**Instance**	KBD
Description	The software must read the pending write bits to insure that following write access will not be discarded due to on going write synchronization process. These bits are automatically cleared by internal logic when the write to the corresponding register is acknowledged.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												PEND_TIMEOUT	PEND_LONG_KEY	PEND_DEBOUNCING	PEND_CTRL

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0.	R	0x000 0000
3	PEND_TIMEOUT	Write pending bit for KBD_TIMEOUT register Read 0x1: A write is pending to the KBD_TIMEOUT register Read 0x0: No write pending to the KBD_TIMEOUT register	R	0

Bits	Field Name	Description	Type	Reset
2	PEND_LONG_KEY	Write pending bit forKBD_KEYLONGTIME register Read 0x1: A write is pending to the KBD_KEYLONGTIME register Read 0x0: No write pending to the KBD_KEYLONGTIME register	R	0
1	PEND_DEBOUNCING	Write pending bit for KBD_DEBOUNCINGTIME register Read 0x1: A write is pending to the KBD_DEBOUNCINGTIME register Read 0x0: No write pending to the KBD_DEBOUNCINGTIME register	R	0
0	PEND_CTRL	Write pending bit for KBD_CTRL register Read 0x1: A write is pending to the KBD_CTRL register Read 0x0: No write pending to the KBD_CTRL register	R	0

Table 28-27. KBD_CTRL

Address Offset	0x0000 0038		
Physical Address	0x4AE1 C038	**Instance**	KBD
Description	This register sets the functional configuration of the module.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							REPEAT_MODE	TIMEOUT_LONG_KEY	TIMEOUT_EMPTY	LONG_KEY	PTV			NSOFTWARE_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads return 0.	R	0x00 0000
8	REPEAT_MODE	Repeat mode enable. 0x0: Repeat mode detection disabled. 0x1: Repeat mode detection enabled.	RW	0
7	TIMEOUT_LONG_KEY	Timeout long key mode enable. 0x0: Timeout long key mode disabled. 0x1: Timeout long key mode enabled.	RW	0
6	TIMEOUT_EMPTY	Timeout empty mode enable. 0x0: Timeout long key mode disabled. 0x1: Timeout long key mode enabled.	RW	0
5	LONG_KEY	Long key mode enable. 0x0: Long key mode disabled. 0x1: Long key mode enabled.	RW	0
4:2	PTV	Pre-scale clock timer value.	RW	0x7
1	NSOFTWARE_MODE	Select hardware or software mode for key decoding. 0x0: Enable software mode. 0x1: Enable hardware decoding using internal sequencer.	RW	1

Bits	Field Name	Description	Type	Reset
0	RESERVED	Reads return 0.	R	0

Table 28-28. KBD_DEBOUNCINGTIME

Address Offset	0x0000 003C		
Physical Address	0x4AE1 C03C	**Instance**	KBD
Description	This register is used to filter glitches on the press key or release key.		
Type	RW		

31–6	5–0
RESERVED	DEBOUNCING_VALUE

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reads return 0.	R	0x000 0000
5:0	DEBOUNCING_VALUE	This value correspond to the desired value of debouncing time.	RW	0x00

Table 28-29. KBD_KEYLONGTIME

Address Offset	0x0000 0040		
Physical Address	0x4AE1 C040	**Instance**	KBD
Description	This register is used to measure duration of a key press, to allow, shortcut detection.		
Type	RW		

31–12	11–0
RESERVED	LONG_KEY_VALUE

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reads return 0.	R	0x0 0000
11:0	LONG_KEY_VALUE	This value correspond to the desired value of the long key interrupt or repeat mode value.	RW	0x000

Table 28-30. KBD_TIMEOUT

Address Offset	0x0000 0044		
Physical Address	0x4AE1 C044	**Instance**	KBD
Description	This register is used to detect a long inactivity on the keyboard.		
Type	RW		

31–16	15–0
RESERVED	TIMEOUT_VALUE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reads return 0.	R	0x0000
15:0	TIMEOUT_VALUE	This value correspond to the desired value of the time out interrupt.	RW	0x0000

Table 28-31. KBD_STATEMACHINE

Address Offset	0x0000 0048		
Physical Address	0x4AE1 C048	**Instance**	KBD
Description	This register indicates the state of the sequencer.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	STATE_MACHINE

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0	R	0x000 0000
3:0	STATE_MACHINE	The state of internal state machine.	R	0x0

Table 28-32. KBD_ROWINPUTS

Address Offset	0x0000 004C		
Physical Address	0x4AE1 C04C	**Instance**	KBD
Description	This register stores the value of the rows input.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							KBR_LATCH								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads return 0.	R	0x00 0000
8:0	KBR_LATCH	The value of the rows input.	R	0x000

Table 28-33. KBD_COLUMNOUTPUTS

Address Offset	0x0000 0050		
Physical Address	0x4AE1 C050	**Instance**	KBD
Description	This register holds the value of the columns output.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							KBC_REG								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads return 0.	R	0x00 0000
8:0	KBC_REG	The value of the columns output.	RW	0x000

Table 28-34. KBD_FULLCODE31_0

Address Offset	0x0000 0054		
Physical Address	0x4AE1 C054	**Instance**	KBD
Description	The KBD_FULLCODE31_0 register codes the row 0, row 1, row 2 and row 3		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FULL_CODE_31_0																															

Bits	Field Name	Description	Type	Reset
31:0	FULL_CODE_31_0	A bit at one indicate that the corresponding key is pressed.	R	0x0000 0000

Table 28-35. KBD_FULLCODE63_32

Address Offset	0x0000 0058		
Physical Address	0x4AE1 C058	**Instance**	KBD
Description	The KBD_FULLCODE63_32 register codes the row 4, row 5, row 6 and row 7.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FULL_CODE_63_32

Bits	Field Name	Description	Type	Reset
31:0	FULL_CODE_63_32	A bit at one indicate that the corresponding key is pressed.	R	0x0000 0000

Table 28-36. KBD_FULLCODE17_0

Address Offset	0x0000 005C		
Physical Address	0x4AE1 C05C	**Instance**	KBD
Description	The KBD_FULLCODE17_0 register codes the row 0 and row 1. The row 0 is coded between bit 0 and 8, the row 1 is coded between bit 24 and		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							ROW1									RESERVED							ROW0								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:16	ROW1	A bit at one indicate that the corresponding key is pressed.	R	0x000
15:9	RESERVED		R	0x00
8:0	ROW0	A bit at one indicate that the corresponding key is pressed.	R	0x000

Table 28-37. KBD_FULLCODE35_18

Address Offset	0x0000 0060		
Physical Address	0x4AE1 C060	**Instance**	KBD
Description	The KBD_FULLCODE35_18 register codes the row 2 and row 3. The row 2 is coded between bit 0 and 8, the row 3 is coded between bit 24 and 16		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							ROW3									RESERVED							ROW2								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:16	ROW3	A bit at one indicate that the corresponding key is pressed.	R	0x000
15:9	RESERVED		R	0x00
8:0	ROW2	A bit at one indicate that the corresponding key is pressed.	R	0x000

Table 28-38. KBD_FULLCODE53_36

Address Offset	0x0000 0064		
Physical Address	0x4AE1 C064	**Instance**	KBD
Description	The KBD_FULLCODE53_36 register codes the row 4 and row 5. The row 4 is coded between bit 0 and 8, the row 5 is coded between bit 24 and 16.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							ROW5									RESERVED							ROW4								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:16	ROW5	A bit at one indicate that the corresponding key is pressed.	R	0x000
15:9	RESERVED		R	0x00
8:0	ROW4	A bit at one indicate that the corresponding key is pressed.	R	0x000

Table 28-39. KBD_FULLCODE71_54

Address Offset	0x0000 0068		
Physical Address	0x4AE1 C068	**Instance**	KBD
Description	The KBD_FULLCODE71_54 register codes the row 6 and row 7. The row 0 is coded between bit 0 and 8, the row 1 is coded between bit 24 and 16		
Type	R		

31:25	24:16	15:9	8:0
RESERVED	ROW7	RESERVED	ROW6

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:16	ROW7	A bit at one indicate that the corresponding key is pressed.	R	0x000
15:9	RESERVED		R	0x00
8:0	ROW6	A bit at one indicate that the corresponding key is pressed.	R	0x000

Table 28-40. KBD_FULLCODE80_72

Address Offset	0x0000 006C		
Physical Address	0x4AE1 C06C	**Instance**	KBD
Description	The KBD_FULLCODE80_72 register codes the row 8. The row 8 is coded between bit 0 and 8.		
Type	R		

31:9	8:0
RESERVED	ROW8

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x00 0000
8:0	ROW8	A bit at one indicate that the corresponding key is pressed.	R	0x000

Chapter 29

Pulse-Width Modulation Subsystem

This chapter describes the Pulse-Width Modulation (PWM) subsystem in the device.

29.1 PWM Subsystem Resources 6693
29.2 Enhanced PWM (ePWM) Module 6708
29.3 Enhanced Capture (eCAP) Module 6802
29.4 Enhanced Quadrature Encoder Pulse (eQEP) Module 6824

29.1 PWM Subsystem Resources

29.1.1 PWMSS Overview

The device has three embedded Pulse Width Modulation Subsystems (PWMSS1, PWMSS2 and PWMSS3). Each PWMSS includes one instance of :

- Enhanced High Resolution Pulse Width Modulator (eHRPWM)
- Enhanced Capture (eCAP)
- Enhanced Quadrature Encoded Pulse (eQEP)

Figure 29-1 shows an overview of each of the 3 available on the device PWM-subsystems.

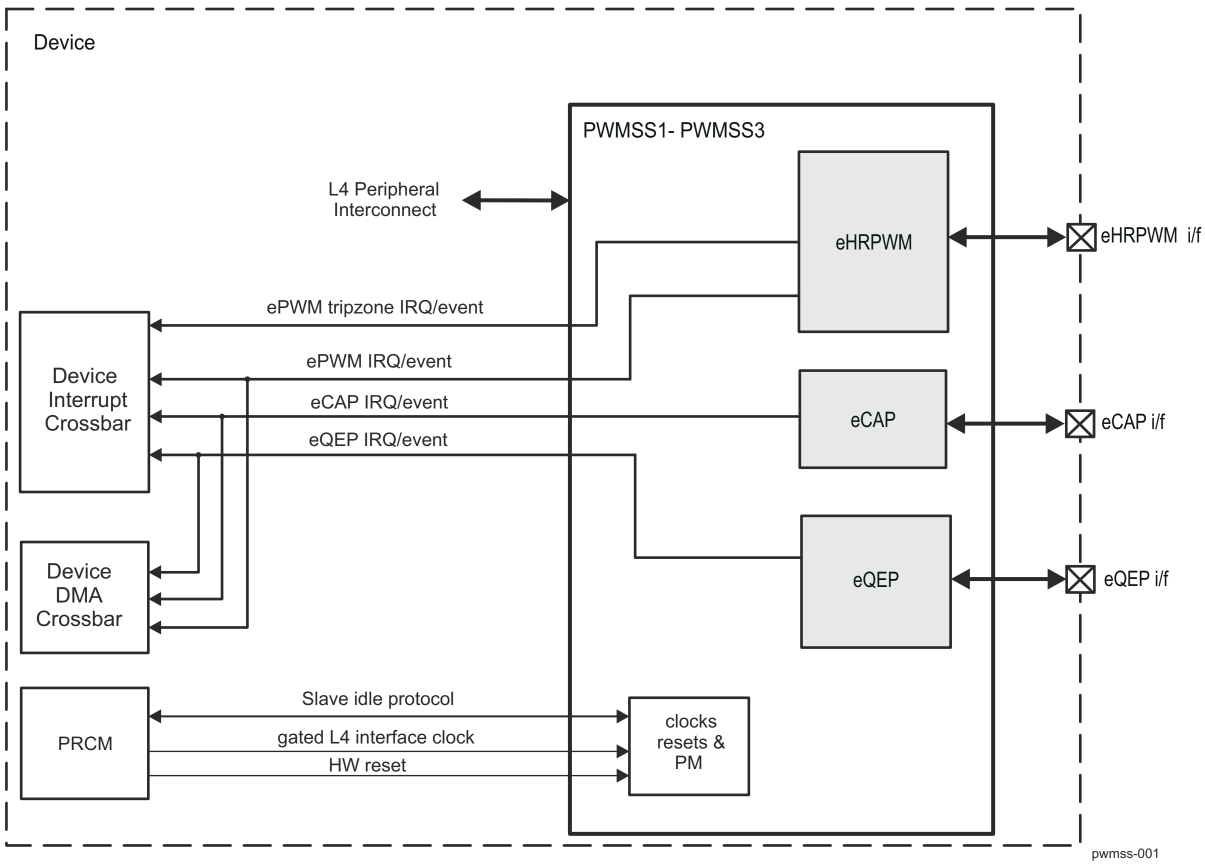


Figure 29-1. PWMSS Block Diagram

29.1.1.1 PWMSS Key Features

The supported features by the device PWMSS-s are:

eHRPWM

- Dedicated 16 bit time-base with Period/Frequency control
- Can support 2 independent PWM outputs with Single edge operation
- Can support 2 independent PWM outputs with Dual edge symmetric operation
- Can support 1 independent PWM output with Dual edge asymmetric operation
- Supports Dead-band generation with independent Rising and Falling edge delay control
- Provides asynchronous over-ride control of PWM signals during fault conditions
- Supports “trip zone” allocation of both latched and un-latched fault conditions
- Allows events to trigger both CPU interrupts and start of ADC conversions
- Support PWM chopping by high frequency carrier signal, used for pulse transformer gate drives.
- High-resolution module with programmable delay line:

- – Programmable on a per PWM period basis
- – Can be inserted either on the rising edge or falling edge of the PWM pulse or both or not at all

eCAP

- Dedicated input Capture pin
- 32 bit Time Base (counter)
- 4 x 32 bit Time-stamp Capture registers (PWMSS_ECAP_CAP1 - PWMSS_ECAP_CAP4)
- 4 stage sequencer (Mod4 counter) which is synchronized to external events (ECAPx pin edges)
- Independent Edge polarity (Rising/Falling edge) selection for all 4 events
- Input Capture signal pre-scaling (from 1 to 16)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 Time-stamp events
- Control for continuous Time-stamp captures using a 4 deep circular buffer (PWMSS_ECAP_CAP1 - PWMSS_ECAP_CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

eQEP

- Input Synchronization
- Three Stage/Six Stage Digital Noise Filter
- Quadrature Decoder Unit
- Position Counter and Control unit for position measurement
- Quadrature Edge Capture unit for low speed measurement
- Unit Time base for speed/frequency measurement
- Watchdog Timer for detecting stalls

At a PWMSS system level :

- each of the three PWMSS generates 2x eHRPWM, 1x eCAP and 1x eQEP event mapped to both the device IRQ crossbar.
- Three of the interrupt events (excluding ePWM tripzone event) are also mapped as DMA requests to the device DMA crossbar.
- The different PWMSS - ePWM/eHRPWM and eCAP modules have their synchronization input /output signals coupled in a daisy chain implemented between PWMSS1, PWMSS2 and PWMSS3 within the device.

29.1.1.2 PWMSS Unsupported Fetaures

The PWMSS limitations in the device are :

- No ePWM inputs are pinned-out (available at the off-chip boundary)
- Only one ePWM tripzone input is pinned-out
- No ePWM digital comparator inputs are pinned-out
- Only input signals of QEP are pinned-out.

Table 29-1. PWMSS Unsupported Features

Feature	Reason
ePWM inputs	Not pinned out
ePWM tripzone 1-5 inputs	Only Tripzone0 is pinned out
ePWM digital comparators	Inputs not connected
eQEP quadrature outputs	Only input signals are connected

29.1.2 PWMSS Environment

29.1.2.1 PWMSS I/O Interface

Table 29-2 shows the device integrated PWMSS subsystems interface signals to external devices.

Table 29-2. PWM Subsystems I/O Signals

PWMSS Modules Level Signal Name	Device Level Signal Name	I/O Type[(1)]	Description	Module Pin Reset Value
		PWMSS1		
EPWM1A	ehrpwm1A	O	PWM1 output A	0
EPWM1B	ehrpwm1B	O	PWM1 output B	0
EPWM1SYNCI	ehrpwm1_synci	I	PWM1 Sync input	HiZ
EPWM1SYNCO	ehrpwm1_synco	O	PWM1 Sync output	0
$\overline{\text{EPWM1\_TRIP\_TZ[0]}}$	ehrpwm1_tripzone_input	I	PWM1 TripZone input	HiZ
ECAP1_CAPIN_APWMOUT	eCAP1_in_PWM1_out	I/O	eCAP1 Capture input/PWM1 output	HiZ
EQEP1_A	eQEP1A_in	I	eQEP1 Quadrature input	HiZ
EQEP1_B	eQEP1B_in	I	eQEP1 Quadrature input	HiZ
EQEP1_INDEX	eQEP1_index	I/O	eQEP1 Index input/output	HiZ
EQEP1_STROBE	eQEP1_strobe	I/O	eQEP1 Strobe input/output	HiZ
		PWMSS2		
EPWM2A	ehrpwm2A	O	PWM2 output A	0
EPWM2B	ehrpwm2B	O	PWM2 output B	0
$\overline{\text{EPWM2\_TRIP\_TZ[0]}}$	ehrpwm2_tripzone_input	I	PWM2 TripZone input	HiZ
ECAP2_CAPIN_APWMOUT	eCAP2_in_PWM2_out	I/O	eCAP2 Capture input/PWM2 output	HiZ
EQEP2_A	eQEP2A_in	I	eQEP2 Quadrature input	HiZ
EQEP2_B	eQEP2B_in	I	eQEP2 Quadrature input	HiZ
EQEP2_INDEX	eQEP2_index	I/O	eQEP2 Index input/output	HiZ
EQEP2_STROBE	eQEP2_strobe	I/O	eQEP2 Strobe input/output	HiZ
		PWMSS3		
EPWM3A	ehrpwm3A	O	PWM3 output A	0
EPWM3B	ehrpwm3B	O	PWM3 output B	0
$\overline{\text{EPWM3\_TRIP\_TZ[0]}}$	ehrpwm3_tripzone_input	I	PWM3 TripZone input	HiZ
ECAP3_CAPIN_APWMOUT	eCAP3_in_PWM3_out	I/O	eCAP3 Capture input/PWM3 output	HiZ
EQEP3_A	eQEP3A_in	I	eQEP3 Quadrature input	HiZ
EQEP3_B	eQEP3B_in	I	eQEP3 Quadrature input	HiZ
EQEP3_INDEX	eQEP3_index	I/O	eQEP3 Index input/output	HiZ
EQEP3_STROBE	eQEP3_strobe	I/O	eQEP3 Strobe input/output	HiZ

(1) I = Input; O = Output

Note

The PWMSSn (where n=1 to 3 synchronization I/O signals which are NOT available at chip pad level are as follows:

- ePWM2SYNCI/SYNCO, ePWM3SYNCI/SYNCO
- eCAP1SYNCI/SYNCO, eCAP2SYNCI/SYNCO, eCAP3SYNCI/SYNCO

These signals are interconnected via a daisy chain implemented within the device. See also Section 29.1.3.1.2.

For more details on synchronization daisy-chain which exist between the PWMSS1, PWMSS2 and PWMSS3, refer to the Section 29.1.3.1.2 .

Figure 29-2 shows the external interface I/Os for the integrated modules in PWMSSn (where n=1 to 3) .

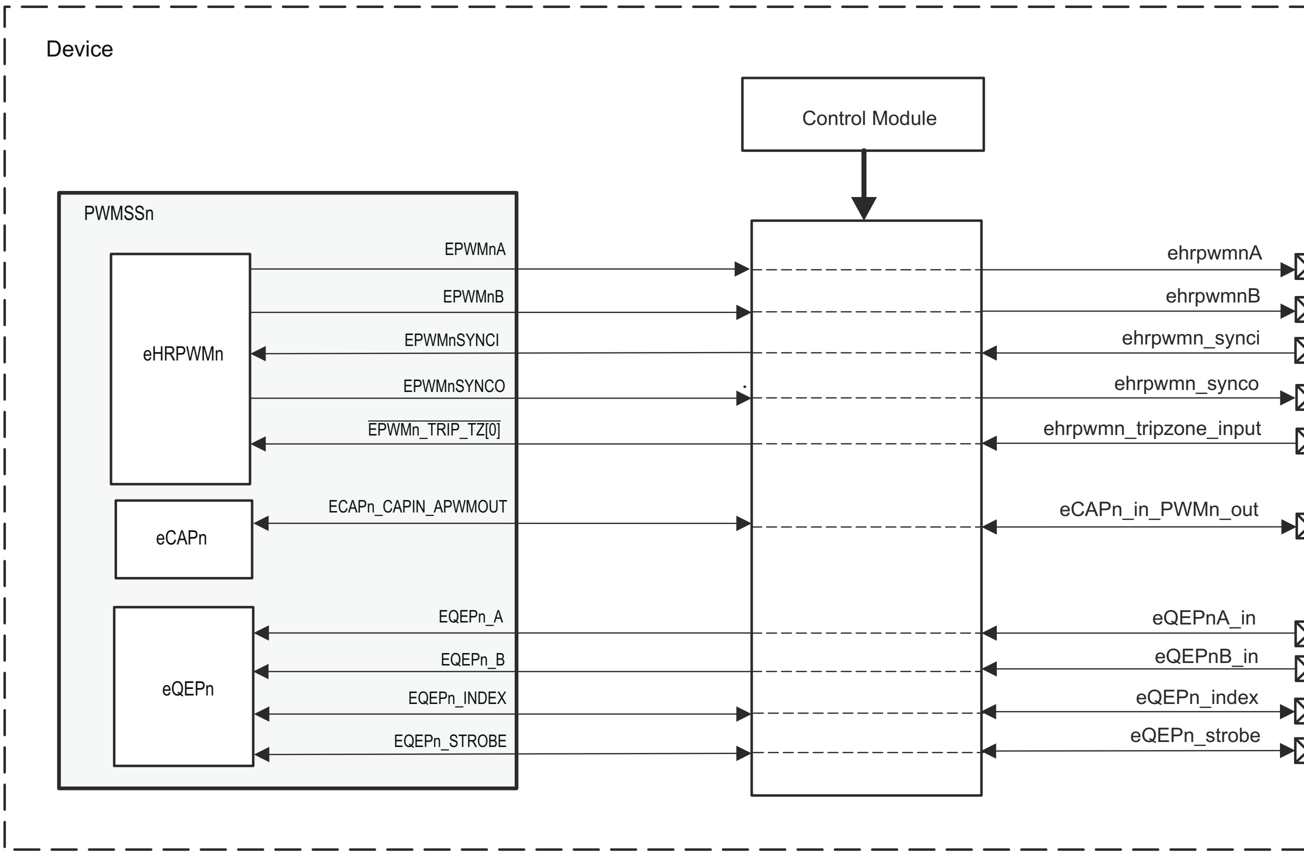


Figure 29-2. PWMSS External Interface I/Os

Note

The path from module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. For more information, refer to the *Pad Configuration Registers*, in the chapter, *Control Module*.

29.1.3 PWMSS Integration

There are three instances of the PWMSS integrated in the device. Each of the the Pulse Width Modulation Subsystems (PWMSS) includes a single instance of one pulse width modulator (ePWM) including an Enhanced High Resolution Modulator (eHRPWM), one Enhanced Capture (eCAP), and one Enhanced Quadrature Encoded Pulse (eQEP) modules.

Note

Let's assume that the letter "x" is used to denote the number of a PWMSS submodule (ePWM/eCAP and eQEP) within the device, and NOT within the PWMSSn itself. Because there is only one ePWM/ eCAP and eQEP per device PWMSS, device point of view, the index - "x" of a module matches the PWMSS index "n". Therefore the ePWM1/eHRPWM1, eCAP1 and eQEP1 correspond to PWMSS1; ePWM2/eHRPWM2/eCAP2 and eQEP2 correspond to PWMSS2; ePWM3/eHRPWM3/eCAP3 and eQEP3 correspond to PWMSS3, i.e. x=n.

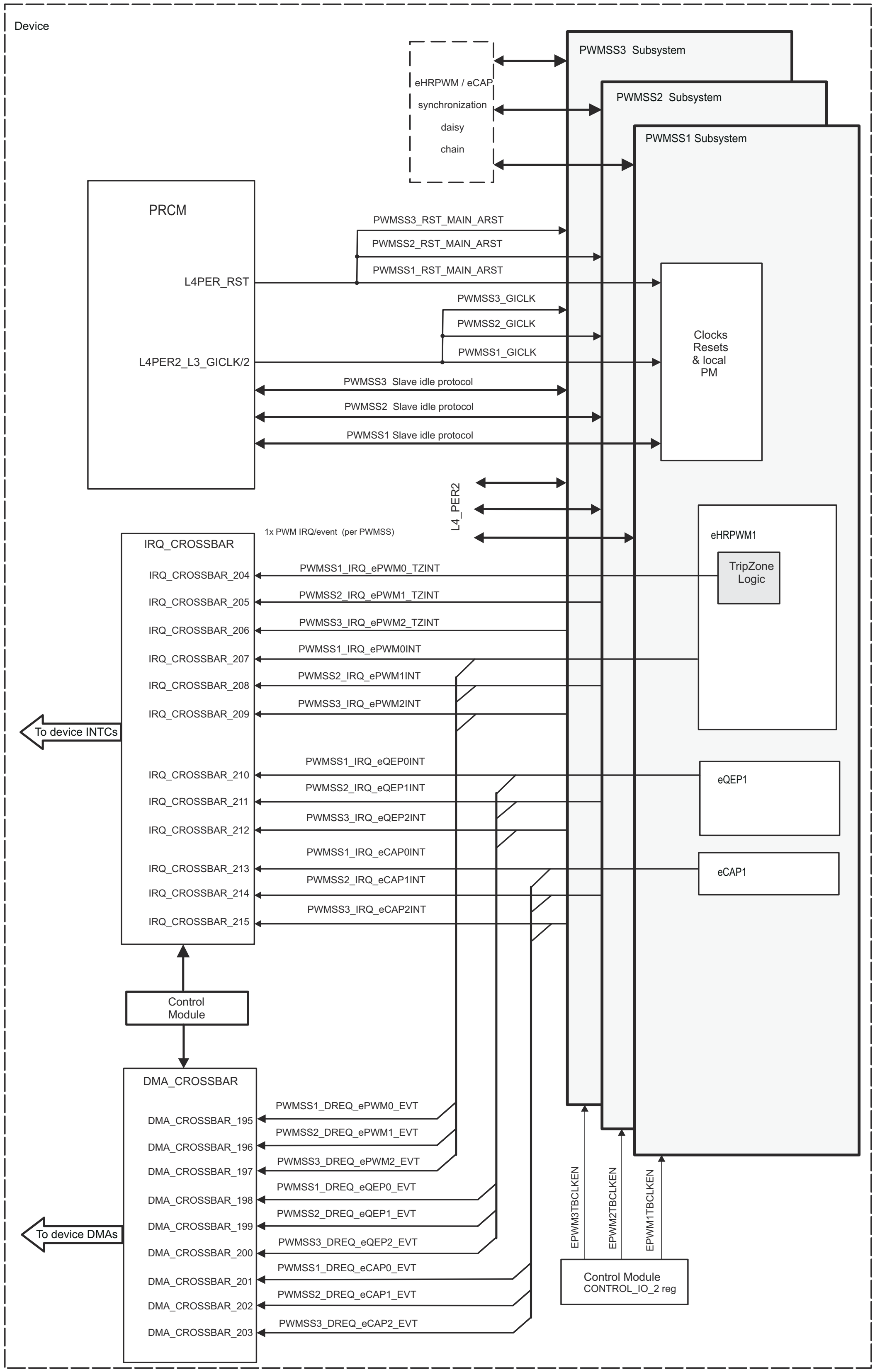


Figure 29-3. PWMSS Integration

At system level the PWMSS1 through PWMSS3 integration features:

- A 32-bit slave configuration port on the L4_PER2 interconnect.
- A single gateable interface and functional clock from PRCM shared between the 3 PWMSS.
- A slave idle protocol with the device PRCM
- A "non-wakeup capable Smart Idle" mode supported

Note

For more information about the slave idle protocol, see *Module Level Clock Management* in *Power, Reset, and Clock Management*.

- A non-retention hardware reset from PRCM to all 3 PWMSS simultaneously
- A PWMSS global level software reset which imapcts all registers of a PWMSS at the same time
- 4 hardware events per PWMSS, i.e. a total of 12 events. From these events each PWMSS :
  – generates 4 interrupts to the device IRQ_CROSSBAR
  – generates 3 DMA requests, mapped to the device DMA_CROSSBAR
- A synchronization input/output daisy chain-like connection exists between the PWMSS1,PWMSS2 and PWMSS3. For more details, refer to Section 29.1.3.1.2.

Table 29-3 through Table 29-5 summarize the integration of the module in the device.

Table 29-3. PWMSS Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
PWMSS1	PD_COREAON	L4_PER2
PWMSS2		
PWMSS3		

Table 29-4. PWMSS Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
PWMSS1	PWMSS1_GICLK	L4PER2_L3_GICLK/2	PRCM	PWMSS1 gated interface and functional clock
PWMSS2	PWMSS2_GICLK	L4PER2_L3_GICLK/2	PRCM	PWMSS2 gated interface and functional clock
PWMSS3	PWMSS3_GICLK	L4PER2_L3_GICLK/2	PRCM	PWMSS3 gated interface and functional clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
PWMSS1	PWMSS1_RST_MAIN_ARST	L4PER_RST	PRCM	A nonretention hardware main reset to the PWMSS1
PWMSS2	PWMSS2_RST_MAIN_ARST	L4PER_RST	PRCM	A nonretention hardware main reset to the PWMSS2
PWMSS3	PWMSS3_RST_MAIN_ARST	L4PER_RST	PRCM	A nonretention hardware main reset to the PWMSS3

Table 29-5. PWMSS Hardware Requests

Interrupt Requests				
Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
PWMSS1	PWMSS1_IRQ_ePWM0_TZINT	IRQ_CROSSBAR_204	-	eHRPWM1 tripzone event/ interrupt.
	PWMSS1_IRQ_ePWM0INT	IRQ_CROSSBAR_207	-	eHRPWM1 event/interrupt.
	PWMSS1_IRQ_eQEP0INT	IRQ_CROSSBAR_210	-	eQEP1 event/interrupt.
	PWMSS1_IRQ_eCAP0INT	IRQ_CROSSBAR_213	-	eCAP1 event/interrupt.

Table 29-5. PWMSS Hardware Requests (continued)

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
PWMSS2	PWMSS2_IRQ_ePWM1_TZINT	IRQ_CROSSBAR_205	-	eHRPWM2 tripzone event/ interrupt.
	PWMSS2_IRQ_ePWM1INT	IRQ_CROSSBAR_208	-	eHRPWM2 event/interrupt.
	PWMSS2_IRQ_eQEP1INT	IRQ_CROSSBAR_211	-	eQEP2 event/interrupt.
	PWMSS2_IRQ_eCAP1INT	IRQ_CROSSBAR_214	-	eCAP2 event/interrupt.
PWMSS3	PWMSS3_IRQ_ePWM2_TZINT	IRQ_CROSSBAR_206	-	eHRPWM3 tripzone event/ interrupt.
	PWMSS3_IRQ_ePWM2INT	IRQ_CROSSBAR_209	-	eHRPWM3 event/interrupt.
	PWMSS3_IRQ_eQEP2INT	IRQ_CROSSBAR_212	-	eQEP3 event/interrupt.
	PWMSS3_IRQ_eCAP2INT	IRQ_CROSSBAR_215	-	eCAP3 event/interrupt.
		DMA Requests		
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
PWMSS1	PWMSS1_DREQ_ePWM0_EVT	DMA_CROSSBAR_195	-	eHRPWM1 event DMA request.
	PWMSS1_DREQ_eQEP0_EVT	DMA_CROSSBAR_198	-	eQEP1 event DMA request.
	PWMSS1_DREQ_eCAP0_EVT	DMA_CROSSBAR_201	-	eCAP1 event DMA request.
PWMSS2	PWMSS2_DREQ_ePWM1_EVT	DMA_CROSSBAR_196	-	eHRPWM2 event DMA request.
	PWMSS2_DREQ_eQEP1_EVT	DMA_CROSSBAR_199	-	eQEP2 event DMA request.
	PWMSS2_DREQ_eCAP1_EVT	DMA_CROSSBAR_202	-	eCAP2 event DMA request.
PWMSS3	PWMSS3_DREQ_ePWM2_EVT	DMA_CROSSBAR_197	-	eHRPWM3 event DMA request.
	PWMSS3_DREQ_eQEP2_EVT	DMA_CROSSBAR_200	-	eQEP3 event DMA request.
	PWMSS3_DREQ_eCAP2_EVT	DMA_CROSSBAR_203	-	eCAP3 event DMA request.

Note

The “Default Mapping” column inTable 29-5,*PWMSS Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.

For more information about the IRQ_CROSSBAR module, see *IRQ_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description*, in *Control Module*.

For more information about the device interrupt controllers, see *Interrupt Controllers*.

For more information about the device DMA_SYSTEM module, see Section 16.1, *System DMA*.

For more information about the device EDMA module, see Section 16.2, *Enhanced DMA*.

29.1.3.1 PWMSS Module Interfaces Implementation

This section describes how the different PWMSSn (where n=1 to 3 for the device) submodules - ePWM/ eHRPWM, eQEP and eCAP can be used in terms of their functional interfaces considering PWMSS **device specific integration**.

29.1.3.1.1 Device Specific PWMSS Features

A High-Resolution PWM (HRPWM) modulator is added to the ePWM module in each of the device PWMSS subsystems, hence the device ePWMs are signified as eHRPWMs.

Note

The HR PWM option applies only to the ePWMxA output channel of the PWMSSn (ehre n=1 to 3 for the device). The PWMSSn ePWMxB channel has conventional capabilities. For more details on HRPWM features of ePWM modules refer to the Section 29.2.45 .

While the eCAP functionalities are fully implemented in the device, the eHRPWM and eQEP module interface have some restrictions in functionality. The eHRPWM and eQEP restrictions which are common between the PWMSS1, PWMSS2 and PWMSS3 subsystems are, as follows:

- For ePWM comparators - only the outputs (ehrpwmxA and ehrpwmxB) are available at chip level
- Only one tripzone input pin - EPWMx_TRIP_TZ[0] of ePWM/eHRPWM is available to user at chip level
- Only QEP inputs (A/ B/INDEX and STROBE) are available to user at chip level

The eHRPWM and eQEP functional interface signals which are NOT available to user for the PWMSS1, PWMSS2 and PWMSS3 are summarized in Table 29-6.

Table 29-6. Device Limitations for the eHRPWM and eQEP Functional Interfaces of PWMSSn

Module Interface	Signal	Description	Comment
PWMSSn eHRPWM (where n= 1 to 3)	EPWM_COMP_EPWMDCMAH	ePWM Comparator A input (HIGH)	Not available at chip boundary (can not be used)
	EPWM_COMP_EPWMDCMAL	ePWM Comparator A input (LOW)	
	EPWM_COMP_EPWMDCMBH	ePWM Comparator B input (HIGH)	
	EPWM_COMP_EPWMDCMBL	ePWM Comparator B input (LOW)	
	EPWM_EPWMA_i	ePWM A input	
	EPWM_EPWMB_i	ePWM B input	
	EPWM_TRIP_TZ[5:1]	ePWM Tripzone inputs [5:1]	
	EPWM_ADC_SOCA	ePWM Start of ADC conversion A output	
	EPWM_ADC_SOCB	ePWM Start of ADC conversion B output	
	EPWM_TRIP_TZ_O[5:0]	ePWM Tripzone outputs	
PWMSSn eQEP (where n= 1 to 3)	EQEP_ERR_PHASE_ERR	eQEP phase error output	Not available at chip boundary (can not be used)
	EQEP_EQEPA_O	eQEP quadrature A output	
	EQEP_EQEPB_O	eQEP quadrature B output	

Note

Only for the device eHRPWM1 module, the SYNCI input and SYNCO output signals are available at chip-level – these are the signals ehrpwm1_synci and ehrpwm1_synco, respectively. The eHRPWM2 and eHRPWM3 SYNCi and SYNCO signals are not available on device pads. For more information, see section, PWMSS Environment.

29.1.3.1.2 Daisy-Chain Connectivity between PWMSS Modules

The PWM (eHRPWM) and capture (eCAP) components of the PWMSSn (where n= 1 through 3 for the device) provide synchronization signals to allow them to be synchronized to other modules or events. In the device, these signals are connected in a daisy-chain fashion as shown in Figure 29-4. Only the PWMSS1 eHRPWM input synchronization signal is terminated at a device pad (ehrpwm1_synci signal), such that device external sync events can be directly applied only to the PWMSS1 eHRPWM (eHRPWM1) submodule. A synchronization output from device PWMSS is available only from the eHRPWM1 on the ehrpwm1_synco output pin . The PWMSS2 and PWMSS3 rely on synchronization signals from its neighbour PWM subsystem within device ,as shown in Figure 29-4.

Note

The PWMSS1.EPWM1SYNCI signal (device ehrpwm1_synci input signal) triggers the event of the eHRPWM1 Phase Register being loaded into the Counter register (TBCNT -> TBPHS). This event is synchronous to the PWMSS1 eHRPWM **time-base clock** (TBCLK).

The PWMSS1 EPWM1SYNCO (device ehrpwm1_synco output signal) is implicitly synchronous to the time-base clock, as this signal has a programmable source of event (in EPWM_TBCTL[5:4] SYNCOSEL) triggered synchronously to the eHRPWM1 TBCLK.

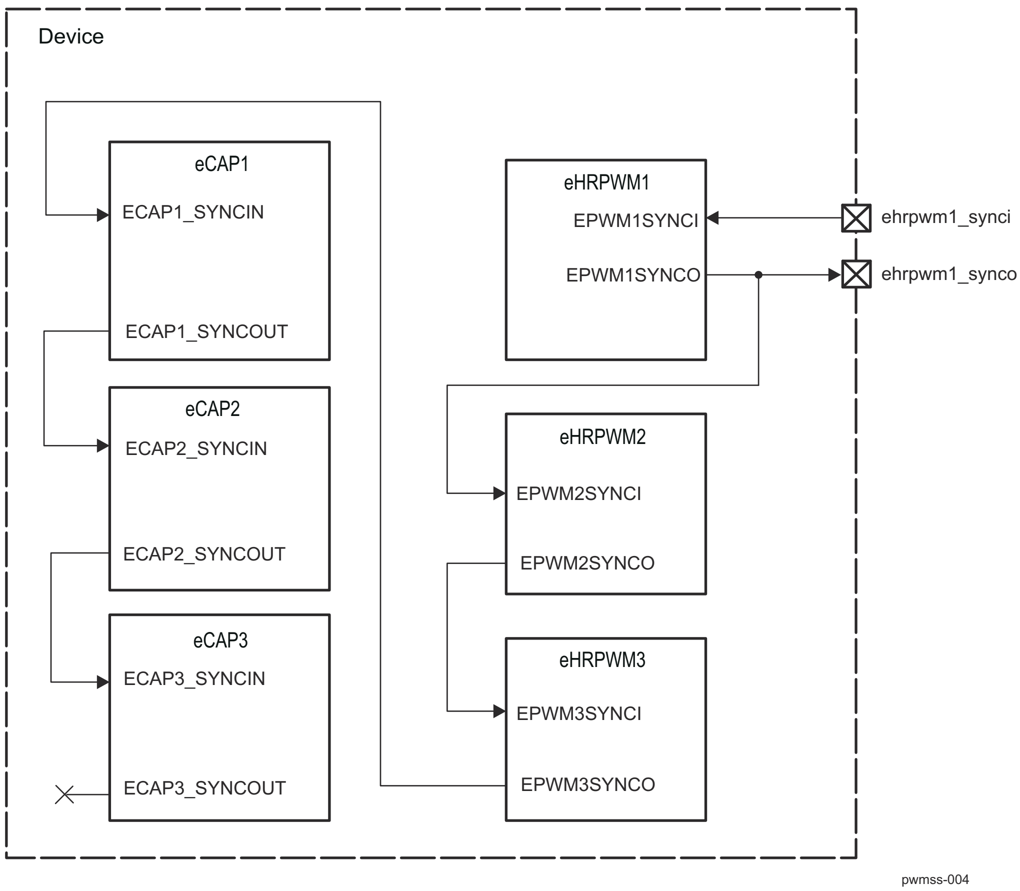


Figure 29-4. Synchronization between PWMSS1, PWMSS2 and PWMSS3

29.1.3.1.3 eHRPWM Modules Time Base Clock Gating

Each PWMSSn ePWM/eHRPWM module has an EPWMTBCLKEN module input used to individually enable/disable its ePWM time-base clock. For each PWMSSn (where n=1 to 3), the ePWM/eHRPWM time-base clock enable input comes from the device core control module register - CTRL_CORE_CONTROL_IO_2, as follows:

- PWMSS1 ePWM -> CTRL_CORE_CONTROL_IO_2[22] PWMSS3_TBCLKEN bit
- PWMSS2 ePWM -> CTRL_CORE_CONTROL_IO_2[21] PWMSS2_TBCLKEN bit
- PWMSS3 ePWM -> CTRL_CORE_CONTROL_IO_2[20] PWMSS1_TBCLKEN bit

This individual TBCLKEN control can be used to align the ePWM time base clock between the three device PWMSS subsystems. PWMSSn_TBCLKEN bit set to 0b0, holds the TBCLK generation counter in its reset state. When PWMSSn_TBCLKEN is set to 0b1, then the TBCLK generation counter is allowed to count.

For more details on the CTRL_CORE_CONTROL_IO_2, refer to the *Control Module Register Manual*, in the chapter *Control Module*.

29.1.4 PWMSS Subsystem Power, Reset and Clock Configuration

29.1.4.1 PWMSS Local Clock Management

The system configuration register - PWMSS_SYSCONFIG is used to configure the local clock management of a PWMSSn slave configuration port on the device L4_PER2 interconnect. Note that this register impacts the behaviour of the root interface and functional clock PWMSSn_GICLK inside PWMSSn (n=1 to 3) shared between the ePWM/eHRPWM, eQEP and eCAP modules within a subsystem. An idle handashake protocol is supported between PWMSSn and the device PRCM.

Note

For more information about the slave idle protocol, see *Module Level Clock Management* in *Power, Reset, and Clock Management*.

Table 29-7. Local IDLE Clock Management Features

Feature	Register bitfield	Description
Slave idle mode	PWMSS_SYSCONFIG[3:2] IDLEMODE	The available modes are: Force-idle, no-idle, and smart-idle (non-wakeup capable) modes.

Note

The device PWM subsystems are part of the CD_L4_PER2 clock domain. For more details on the PWM subsystems top level clock-management modes and control, refer to the *Clock Domain Module Attributes*, in the *Power, Reset, and Clock Management*.

29.1.4.2 PWMSS Modules Local Clock Gating

Note

PWMSS Modules Local Clock Gating feature is not supported in this family of devices.

In addition to PWMSS level IDLE clock management, described above, a clock configuration register - PWMSS_CLKCONFIG is used to individually gate (stop) or enable interface and functional clock to the ePWM/eHRPWM, eCAP and eQEP modules. By default, the interface/functional clocks are enabled to all modules.

The clock status register - PWMSS_CLKSTATUS is used in the PWMSS submodule to indicated acknowledgement of a clock stop or clock enable request.

The PWMSS_CLKCONFIG and PWMSS_CLKSTATUS role is described in Table 29-8.

Table 29-8. Local Module Clock Control and Status Features

Clock Control/Status Feature	PWMSSn Module	Register bit
Request "stop interface and functional clock" to module	ePWM/eHRPWM	PWMSS_CLKCONFIG[9] EPWM_CLKSTOP_REQ
	eQEP	PWMSS_CLKCONFIG[5] EQEP_CLKSTOP_REQ
	eCAP	PWMSS_CLKCONFIG[1] ECAP_CLKSTOP_REQ
"Stop module interface and functional clock" acknowledged status	ePWM/eHRPWM	PWMSS_CLKSTATUS[9] EPWM_CLKSTOP_ACK
	eQEP	PWMSS_CLKSTATUS[5] EQEP_CLKSTOP_ACK
	eCAP	PWMSS_CLKSTATUS[1] ECAP_CLKSTOP_ACK
Request "enable interface and functional clock" to module	ePWM/eHRPWM	PWMSS_CLKCONFIG[8] EPWM_CLK_EN
	eQEP	PWMSS_CLKCONFIG[4] EQEP_CLK_EN
	eCAP	PWMSS_CLKCONFIG[0] ECAP_CLK_EN
"Enable module Interface and functional clock" acknowledged status	ePWM/eHRPWM	PWMSS_CLKSTATUS[8] EPWM_CLK_EN_ACK
	eQEP	PWMSS_CLKSTATUS[4] EQEP_CLK_EN_ACK
	eCAP	PWMSS_CLKSTATUS[0] ECAP_CLK_EN_ACK

Note

In order for PWMSSn to enter "Idle state", all PWMSSn submodules must have acknowledged a stop clock request, i.e. the PWMSS_CLKSTATUS bits EPWM_CLKSTOP_ACK, ECAP_CLKSTOP_ACK and EQEP_CLKSTOP_ACK must be raised 'HIGH'.

29.1.4.3 PWMSS Software Reset

The PWMSS_SYSCONFIG[0] SOFTRESET bit can be used to exert a "PWMSS" software reset impacting ePWM/eHRPWM, eCAP and eQEP modules of PWMSSn at the same time. The software has to poll the same bit until it is deasserted by HW, to insure software reset of all modules is completed.

29.1.5 PWMSS_CFG Register Manual

This section provides description of the PWM subsystem (top) level functional registers.

29.1.5.1 PWMSS_CFG Instance Summary

Table 29-9. PWMSS_CFG Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS1_CFG	0x4843 E000	48 Bytes
PWMSS2_CFG	0x4844 0000	48 Bytes
PWMSS3_CFG	0x4844 2000	48 Bytes

29.1.5.2 PWMSS_CFG Registers

29.1.5.2.1 PWMSS_CFG Register Summary

Table 29-10. PWMSSn_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS1_CFG Physical Address	PWMSS2_CFG Physical Address	PWMSS3_CFG Physical Address
PWMSS_IDVER	RW	32	0x0000 0000	0x4843 E000	0x4844 0000	0x4844 2000
PWMSS_SYSCONFIG	RW	32	0x0000 0004	0x4843 E004	0x4844 0004	0x4844 2004
PWMSS_CLKCONFIG	RW	32	0x0000 0008	0x4843 E008	0x4844 0008	0x4844 2008
PWMSS_CLKSTATUS	RW	32	0x0000 000C	0x4843 E00C	0x4844 000C	0x4844 200C

29.1.5.2.2 PWMSS_CFG Register Description

Table 29-11. PWMSS_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4843 E000 0x4844 0000 0x4844 2000	**Instance**	PWMSS1_CFG PWMSS2_CFG PWMSS3_CFG
Description	IP Revision Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision value	R	0x-(1)

(1) TI Internal data

Table 29-12. PWMSS_SYSCONFIG

Address Offset	0x0000 0004

Table 29-12. PWMSS_SYSCONFIG (continued)

Physical Address	0x4843 E004 0x4844 0004 0x4844 2004	**Instance**	PWMSS1_CFG PWMSS2_CFG PWMSS3_CFG
Description	This register controls the PWMSSn (where n= 1 to 3) local Idle mode clock management and software reset		
Type	RW		

31:4	3:2	1	0
RESERVED	IDLEMODE	RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, the target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: The local target IDLE state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the internal requirements of th e IP module. Backup mode, for debug only. 0x1: No-idle mode: The local target never enters IDLE state. Backup mode, for debug only. 0x2: Smart-idle mode: The local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the internal requirements of the IP module. IP module does not generate (IRQ- or DMA-request-related) wakeup events. 0x3: Reserved	RW	0x2
1	RESERVED		R	0
0	SOFTRESET	Software reset : 0x0 : Software reset is completed 0x1: Software reset assertion	RW	0x0

Table 29-13. PWMSS_CLKCONFIG

Address Offset	0x0000 0008		
Physical Address	0x4843 E008 0x4844 0008 0x4844 2008	**Instance**	PWMSS1_CFG PWMSS2_CFG PWMSS3_CFG
Description	The clock configuration register is used in the PWMSSn (where n = 1 to 3) for clkstop req and clk_en control to the ePWM/ eHRPWM, eCAP and eQEP submodules within the PWMSSn subsystem. **Note:** PWMSS Modules Local Clock Gating feature is not supported. This register should not be modified. Clock gating functionality is controlled by PRCM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	EPWM_CLKSTOP_REQ	EPWM_CLK_EN	RESERVED	EQEP_CLKSTOP_REQ	EQEP_CLK_EN	RESERVED	ECAP_CLKSTOP_REQ	ECAP_CLK_EN

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	EPWM_CLKSTOP_REQ	This bit controls the clock stop input to the ePWM/ eHRPWM module : 0: No effect 1: A request to stop interface clock to the module is asserted	RW	0
8	EPWM_CLK_EN	This bit controls the interface clock enable (clk_en) input to the ePWM/eHRPWM module: 0: No effect 1: Enables the interface clock to the module	RW	1
7:6	RESERVED		R	0x0
5	EQEP_CLKSTOP_REQ	This bit controls the clock stop input to the eQEP module : 0: No effect 1: A request to stop interface clock to the module is asserted	RW	0
4	EQEP_CLK_EN	This bit controls the interface clock enable (clk_en) input to the eQEP module : 0: No effect 1: Enables the interface clock to the module	RW	1
3:2	RESERVED		R	0x0
1	ECAP_CLKSTOP_REQ	This bit controls the clock stop input to the eCAP module : 0: No effect 1: A request to stop interface clock to the module is asserted	RW	0
0	ECAP_CLK_EN	This bit controls the interface clock enable (clk_en) input to the eCAP module : 0: No effect 1: Enables the interface clock to the module	RW	1

Table 29-14. PWMSS_CLKSTATUS

Address Offset	0x0000 000C		
Physical Address	0x4843 E00C 0x4844 000C 0x4844 200C	**Instance**	PWMSS1_CFG PWMSS2_CFG PWMSS3_CFG
Description	The clock status register is used in the PWMSSn (where n = 1 to 3) to indicate clock stop acknowledge (clkstop_ack) and clock enable (clk_en) acknowledge status for the ePWM/ eHRPWM, eCAP and eQEP submodules within the PWMSSn subsystem. **Note:** PWMSS Modules Local Clock Gating feature is not supported. Clock gating functionality is controlled by PRCM.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31:10	9	8	7:6	5	4	3:2	1	0
RESERVED	EPWM_CLKSTOP_ACK	EPWM_CLK_EN_ACK	RESERVED	EQEP_CLKSTOP_ACK	EQEP_CLK_EN_ACK	RESERVED	ECAP_CLKSTOP_ACK	ECAP_CLK_EN_ACK

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	EPWM_CLKSTOP_ACK	This bit is the clkstop_req_ack status output of the ePWM/eHRPWM module : 0: No interface clock stop acknowledged 1: Interface clock stop request is acknowledged for the module	R	0
8	EPWM_CLK_EN_ACK	This bit is the clk_en status output of the ePWM/ eHRPWM module : 0: No clock enable request acknowledged 1: Interface clock enable request is acknowledged for the module	R	0
7:6	RESERVED		R	0x0
5	EQEP_CLKSTOP_ACK	This bit is the clkstop_req_ack status output of the eQEP module : 0: No interface clock stop acknowledged 1: Interface clock stop request is acknowledged for the module	R	0
4	EQEP_CLK_EN_ACK	This bit is the clk_en status output of the eQEP module : 0: No clock enable request acknowledged 1: Interface clock enable request is acknowledged for the module	R	0
3:2	RESERVED		R	0x0
1	ECAP_CLKSTOP_ACK	TThis bit is the clkstop_req_ack status output of the eCAP module : 0: No interface clock stop acknowledged 1: Interface clock stop request is acknowledged for the module	R	0
0	ECAP_CLK_EN_ACK	TThis bit is the clk_en status output of the eCAP module : 0: No clock enable request acknowledged 1: Interface clock enable request is acknowledged for the module	R	0

29.2 Enhanced PWM (ePWM) Module

29.2.1 ePWM Overview

An effective PWM peripheral must be able to generate complex pulse width waveforms with minimal CPU overhead or intervention. It needs to be highly programmable and very flexible while being easy to understand and use. The ePWM unit described here addresses these requirements by allocating all needed timing and control resources on a per PWM channel basis. Cross coupling or sharing of resources has been avoided; instead, the ePWM is built up from smaller single channel modules with separate resources and that can operate together as required to form a system. This modular approach results in an orthogonal architecture and provides a more transparent view of the peripheral structure, helping users to understand its operation quickly.

As already described in Section 29.1.3, the letter x within a signal or module name is used to indicate a generic ePWM instance on a device. For example, output signals EPWMxA and EPWMxB refer to the output signals from the ePWMx instance. Thus, EPWM1A and EPWM1B belong to ePWM1, EPWM2A and EPWM2B belong to ePWM2, etc.

The ePWM module represents one complete PWM channel composed of two PWM outputs: EPWMxA and EPWMxB. A given ePWM module functionality can be extended with the so called **High-Resolution Pulse Width modulator**. Refer to the Section 29.1.3, to determine which ePWM instances include the HRPWM feature. The HRPWM functionalities are described in Section 29.2.45 . Each ePWM module is indicated by a numerical value starting with 1. For example ePWM1 is the first instance, the ePWM2 is the second instance in the device, etc. The ePWMx indicates any instance.

As also described in Section 29.1.3.1.2, the ePWM modules are chained together via a clock synchronization scheme that allows them to operate as a single system when required. Additionally, the PWMSSn integration allows this synchronization scheme to be extended to the capture peripheral modules (eCAP). The number of modules is device-dependent and based on target application needs. Modules can also operate stand-alone.

Each ePWM module supports the following features:

- **Dedicated 16-bit time-base counter with period and frequency control**
- **Two PWM outputs (EPWMxA and EPWMxB) that can be used in the following configurations:**
  – Two independent PWM outputs with single-edge operation
  – Two independent PWM outputs with dual-edge symmetric operation
  – One independent PWM output with dual-edge asymmetric operation
- **Asynchronous override control of PWM signals through software.**
- **Programmable phase-control support for lag or lead operation relative to other ePWM modules.**
- **Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.**
- **Dead-band generation with independent rising and falling edge delay control.**
- **Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.**
- **A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.**
- **Programmable event prescaling minimizes CPU overhead on interrupts.**
- **PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.**

Each ePWM module is connected to the input/output signals shown in Figure 29-5. The signals are described in detail in subsequent sections.

The order in which the ePWM modules are connected may differ from what is shown in Figure 29-5. See Section 29.1.3.1.2 for the actual synchronization scheme implemented in the device. Each ePWM module consists of seven submodules and is connected within a system via the signals shown in Figure 29-6.

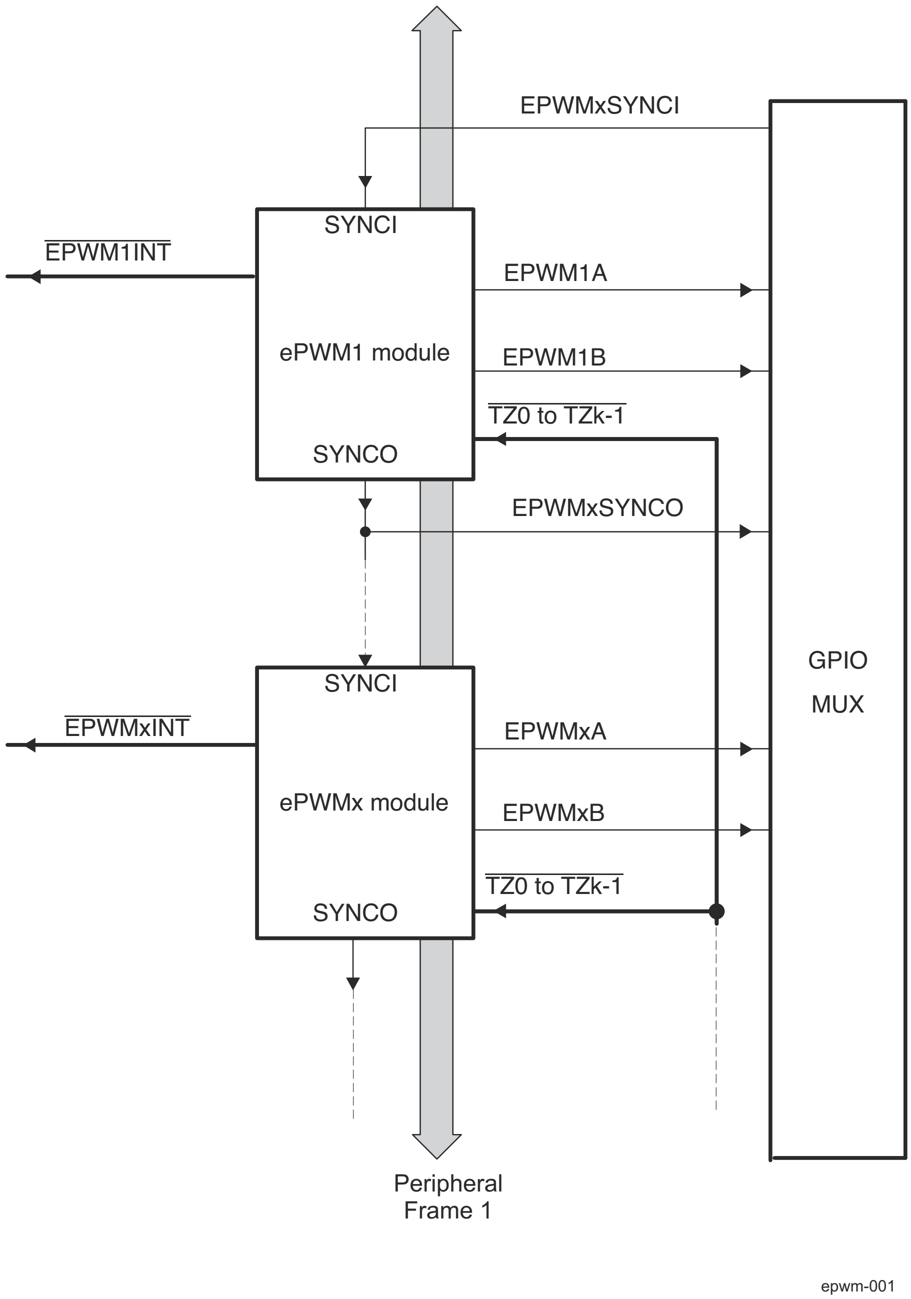


Figure 29-5. Multiple ePWM Modules

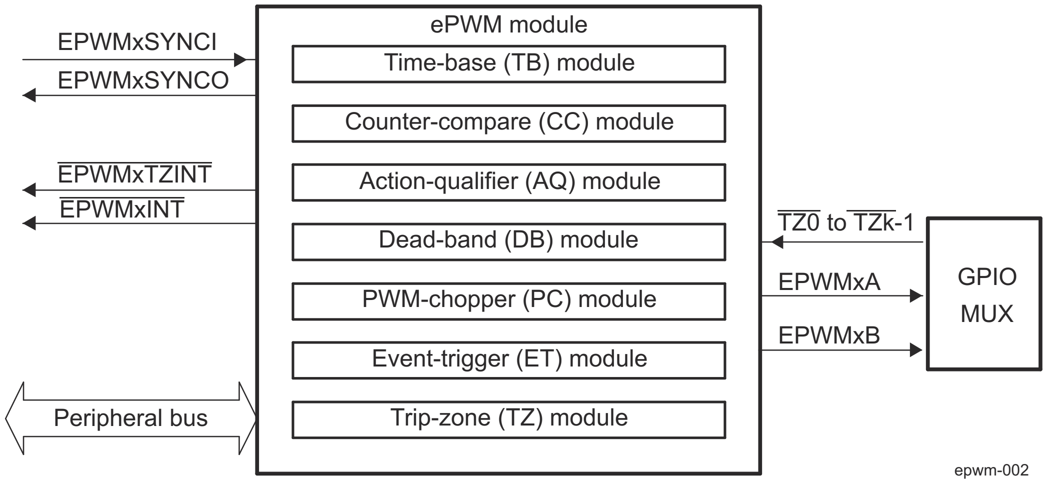


Figure 29-6. Submodules and Signal Connections for an ePWM Module

Figure 29-7 shows more internal details of a single ePWM module. The main signals used by the ePWM module are:

- **PWM output signals (EPWMxA and EPWMxB).** The PWM output signals are made available external to the device through the GPIO peripheral described in the system control and interrupts guide for your device.
- **Trip-zone signals ($\overline{TZ0}$ to $\overline{TZk\text{-}1}$).** These **k** input signals alert the ePWM module of an external fault condition. Each module on a device can be configured to either use or ignore any of the trip-zone signals. The trip-zone signal can be configured as an asynchronous input through the GPIO peripheral.

Note

According to the device ePWMx (where x= 1to 3) trip zone i/f implementation, the number of input signals implemented is k=1, which means **ONLY the ePWMx $\overline{TZ0}$ input is available at chip level**. Refer to the Section 29.1.3, for details on trip zone input implementation.

- **Time-base synchronization input (EPWMxSYNCI) and output (EPWMxSYNCO) signals.** The synchronization signals daisy chain the ePWM modules together. Each module can be configured to either use or ignore its synchronization input. The clock synchronization input and output signal are brought out to pins only for ePWM1 (ePWM module #1). **The synchronization output for ePWM3 (EPWM3SYNCO) is also connected to the eCAP1SYNCI of the first enhanced capture module (eCAP1).**
- **Peripheral Bus.** The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.

Figure 29-7 also shows the key internal submodule interconnect signals. Each submodule is described in Section 29.2.2 .

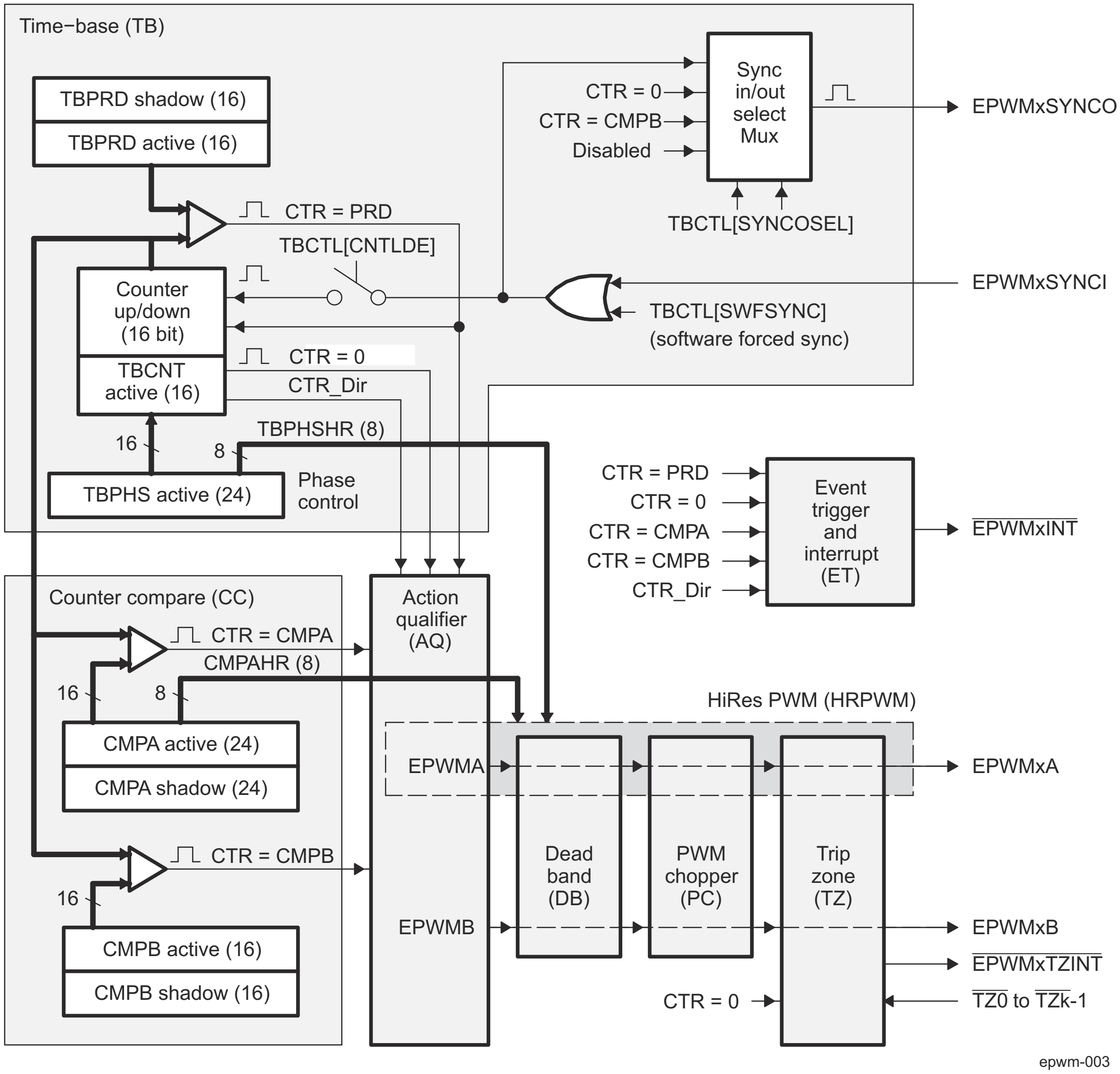


Figure 29-7. ePWM Submodules and Critical Internal Signal Interconnects

29.2.2 ePWM Functional Description

Seven submodules are included in each ePWM peripheral. There are some instances that include a high-resolution submodule that allows more precise control of the PWM outputs. Each of these submodules performs specific tasks that can be configured by software.

29.2.3 ePWM Submodule Features

Table 29-15 lists the eight key submodules together with a list of their main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, then you should see the counter-compare submodule in Section 29.2.14 for relevant details.

Table 29-15. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time-base (TB)	• Scale the time-base clock (TBCLK) relative to the system clock (SYSCLKOUT). • Configure the PWM time-base counter (TBCNT) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-down mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another ePWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Configure how the time-base counter will behave when the device is halted by an emulator. • Specify the source for the synchronization output of the ePWM module: – Synchronization input signal – Time-base counter equal to zero – Time-base counter equal to counter-compare B (CMPB) – No output synchronization signal generated.
Counter-compare (CC)	• Specify the PWM duty cycle for output EPWMxA and/or output EPWMxB • Specify the time at which switching events occur on the EPWMxA or EPWMxB output
Action-qualifier (AQ)	• Specify the type of action taken when a time-base or counter-compare submodule event occurs: – No action taken – Output EPWMxA and/or EPWMxB switched high – Output EPWMxA and/or EPWMxB switched low – Output EPWMxA and/or EPWMxB toggled • Force the PWM output state through software control • Configure and control the PWM dead-band through software
Dead-band (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification.
PWM-chopper (PC)	• Create a chopping (carrier) frequency. • Pulse width of the first pulse in the chopped pulse train. • Duty cycle of the second and subsequent pulses. • Bypass the PWM-chopper module entirely. In this case the PWM waveform is passed through without modification.

Table 29-15. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Trip-zone (TZ)	• Configure the ePWM module to react to one, all, or none of the trip-zone pins. • Specify the tripping action taken when a fault occurs: – Force EPWMxA and/or EPWMxB high – Force EPWMxA and/or EPWMxB low – Force EPWMxA and/or EPWMxB to a high-impedance state – Configure EPWMxA and/or EPWMxB to ignore any trip condition. • Configure how often the ePWM will react to the trip-zone pin: – One-shot – Cycle-by-cycle • Enable the trip-zone to initiate an interrupt. • Bypass the trip-zone module entirely.
Event-trigger (ET)	• Enable the ePWM events that will trigger an interrupt. • Specify the rate at which events cause triggers (every occurrence or every second or third occurrence) • Poll, set, or clear event flags
High-Resolution PWM (HRPWM)	• Enable extended time resolution capabilities • Configure finer time granularity control or edge positioning

Note

The system clock - SYSCLKOUT is the ePWM functional clock derived from the PWMSSn gateable interface and functional clock PWMSSn_GICLK, described in the Section 29.1.3.

Code examples are provided in the remainder of this chapter that show how to implement various ePWM module configurations. These examples use the constant definitions shown in Section 29.2.4 .

29.2.4 Constant Definitions Used in the ePWM Code Examples

```
// TBCTL (Time-Base Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// TBCNT MODE bits
#define         TB_COUNT_UP         0x0
#define         TB_COUNT_DOWN       0x1
#define         TB_COUNT_UPDOWN     0x2
#define         TB_FREEZE           0x3
// PHSEN bit
#define         TB_DISABLE          0x0
#define         TB_ENABLE           0x1
// PRDLD bit
#define         TB_SHADOW           0x0
#define         TB_IMMEDIATE        0x1
// SYNCOSEL bits
#define         TB_SYNC_IN          0x0
#define         TB_CTR_ZERO         0x1
#define         TB_CTR_CMPB         0x2
#define         TB_SYNC_DISABLE     0x3
// HSPCLKDIV and CLKDIV bits
#define         TB_DIV1             0x0
#define         TB_DIV2             0x1
#define         TB_DIV4             0x2
// PHSDIR bit
#define         TB_DOWN             0x0
#define         TB_UP               0x1
```

```
// CMPCTL (Compare Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// LOADAMODE and LOADBMODE bits
#define         CC_CTR_ZERO         0x0
```

```
#define         CC_CTR_PRD        0x1
#define         CC_CTR_ZERO_PRD   0x2
#define         CC_LD_DISABLE     0x3
// SHDWAMODE and SHDWBMODE bits
#define         CC_SHADOW         0x0
#define         CC_IMMEDIATE      0x1
// AQCTLA and AQCTLB (Action-qualifier Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// ZRO, PRD, CAU, CAD, CBU, CBD bits
#define         AQ_NO_ACTION      0x0
#define         AQ_CLEAR          0x1
#define         AQ_SET            0x2
#define         AQ_TOGGLE         0x3
// DBCTL (Dead-Band Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// MODE bits
#define         DB_DISABLE        0x0
#define         DBA_ENABLE        0x1
#define         DBB_ENABLE        0x2
#define DB_FULL_ENABLE 0x3
// POLSEL bits
#define         DB_ACTV_HI        0x0
#define         DB_ACTV_LOC       0x1
#define         DB_ACTV_HIC       0x2
#define         DB_ACTV_LO        0x3
// PCCTL (chopper control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// CHPEN bit
#define         CHP_ENABLE        0x0
#define CHP_DISABLE 0x1
// CHPFREQ bits
#define         CHP_DIV1          0x0
#define         CHP_DIV2          0x1
#define         CHP_DIV3          0x2
#define         CHP_DIV4          0x3
#define         CHP_DIV5          0x4
#define         CHP_DIV6          0x5
#define         CHP_DIV7          0x6
#define         CHP_DIV8          0x7
// CHPDUTY bits
#define         CHP1_8TH          0x0
#define         CHP2_8TH          0x1
#define         CHP3_8TH          0x2
#define         CHP4_8TH          0x3
#define         CHP5_8TH          0x4
#define         CHP6_8TH          0x5
#define         CHP7_8TH          0x6
// TZSEL (Trip-zone Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// CBCn and OSHTn bits
#define         TZ_ENABLE         0x0
#define         TZ_DISABLE        0x1
// TZCTL (Trip-zone Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// TZA and TZB bits
#define         TZ_HIZ            0x0
#define         TZ_FORCE_HI       0x1
#define         TZ_FORCE_LO       0x2
#define         TZ_DISABLE        0x3
// ETSEL (Event-trigger Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTSEL, SOCASEL, SOCBSEL bits
#define         ET_CTR_ZERO       0x1
#define         ET_CTR_PRD        0x2
#define         ET_CTRU_CMPA      0x4
#define         ET_CTRD_CMPA      0x5
#define         ET_CTRU_CMPB      0x6
#define         ET_CTRD_CMPB      0x7
// ETPS (Event-trigger Prescale)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTPRD, SOCAPRD, SOCBPRD bits
#define         ET_DISABLE        0x0
#define         ET_1ST            0x1
#define         ET_2ND            0x2
#define         ET_3RD            0x3
```

29.2.5 Proper ePWM Interrupt Initialization Procedure

When the ePWM peripheral clock is enabled it may be possible that interrupt flags may be set due to spurious events due to the ePWM registers not being properly initialized. The proper procedure for initializing the ePWM peripheral is:

1. Disable global interrupts (CPU INTM flag)
2. Disable ePWM interrupts
3. Initialize peripheral registers
4. Clear any spurious ePWM flags
5. Enable ePWM interrupts
6. Enable global interrupts

29.2.6 ePWM Time-Base (TB) Submodule

Each ePWM module has its own time-base submodule that determines all of the event timing for the ePWM module. Built-in synchronization logic allows the time-base of multiple ePWM modules (ePWMx) to work together as a single system. Figure 29-8 illustrates the time-base module's place within the ePWM.

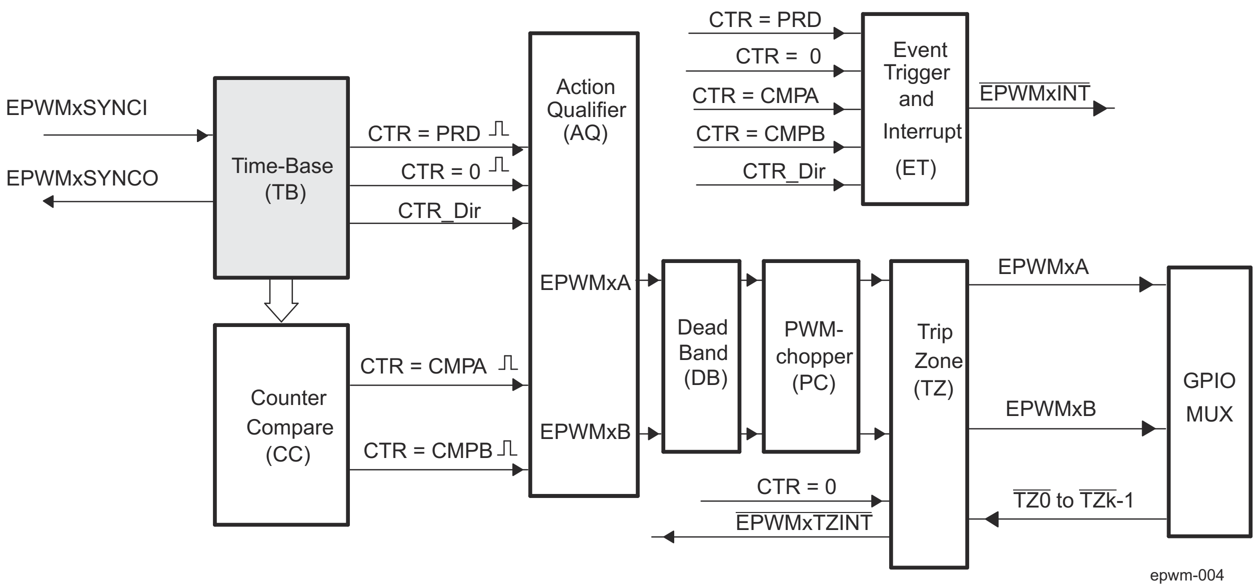


Figure 29-8. ePWM Time-Base Submodule Block Diagram

29.2.7 Purpose of the ePWM Time-Base Submodule

You can configure the time-base submodule for the following:

- Specify the ePWMx time-base counter (TBCNT) frequency or period in the EPWM_TBCNT register to control how often events occur.
- Manage time-base synchronization with other ePWMx modules.
- Maintain a phase relationship with other ePWMx modules.
- Set the time-base counter to count-up, count-down, or count-up-and-down mode.
- Generate the following events:
  - TBCNT = PRD: Time-base counter (EPWM_TBCNT register) equal to the specified period in EPWM_TBPRD register (i.e. TBCNT = TBPRD) .
  - TBCNT = 0: Time-base counter equal to zero (TBCNT = 0000h).
- Configure the rate of the time-base clock; a prescaled version of the CPU system clock (SYSCLKOUT). This allows the time-base counter to increment/decrement at a slower rate.

29.2.8 Controlling and Monitoring the ePWM Time-Base Submodule

Table 29-16 lists the registers used to control and monitor the time-base submodule.

Table 29-16. ePWM Time-Base Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_TBCTL	Time-Base Control Register	0h	No
EPWM_TBSTS	Time-Base Status Register	2h	No
HRPWM_TBPHSHR	HRPWM extension Phase Register (1)	4h	No
EPWM_TBPHS	Time-Base Phase Register	6h	No
EPWM_TBCNT	Time-Base Counter Register	8h	No
EPWM_TBPRD	Time-Base Period Register	Ah	Yes

(1) This register is available only on ePWM instances that include the high-resolution extension (HRPWM). On ePWM modules that do not include the HRPWM, this location is reserved. See Section 29.1.3 to determine which ePWM instances include this feature.

Figure 29-9 shows the critical signals and registers of the time-base submodule. Table 29-17 provides descriptions of the key signals associated with the time-base submodule.

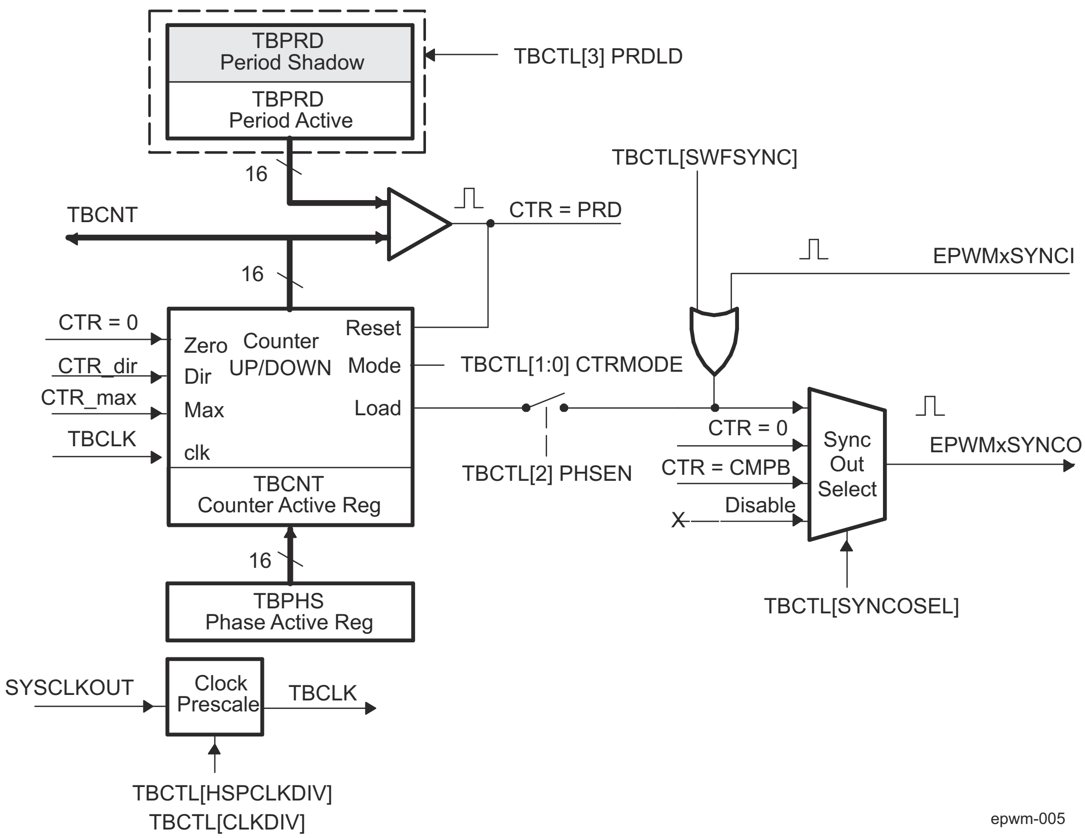


Figure 29-9. ePWM Time-Base Submodule Signals and Registers

Table 29-17. ePWM Key Time-Base Signals

Signal	Description
EPWMxSYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of ePWM module earlier in the synchronization chain. An ePWM peripheral can be configured to use or ignore this signal. For the first ePWM module (EPWM1) this signal comes from a device pin . For subsequent ePWM modules this signal is passed from another ePWM peripheral. For example, EPWM2SYNCI is generated by the ePWM1 peripheral and the EPWM3SYNCI is generated by ePWM2. See Section 29.2.11 for information on the synchronization order of a particular device.

Table 29-17. ePWM Key Time-Base Signals (continued)

Signal	Description
EPWMxSYNCO	Time-base synchronization output. This output pulse is used to synchronize the counter of an ePWM module later in the synchronization chain. The ePWM module generates this signal from one of three event sources: 1. EPWMxSYNCI (Synchronization input pulse) 2. TBCNT = 0: The time-base counter (register EPWM_TBCNT) equal to zero (TBCNT = 0000h). 3. TBCNT = CMPB: The time-base counter (register EPWM_TBCNT) equal to the counter-compare B register - EPWM_CMPB (i.e. bitfield TBCNT = bitfield CMPB) .
TBCNT = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCNT = TBPRD.
TBCNT = 0	Time-base counter equal to zero. This signal is generated whenever the counter value is zero. That is when TBCNT equals 0000h.
TBCNT = CMPB	Time-base counter equal to active counter-compare B register (TBCNT = CMPB). This event is generated by the counter-compare submodule and used by the synchronization out logic.
CTR_dir	Time-base counter direction. Indicates the current direction of the ePWM's time-base counter. This signal is high when the counter is increasing and low when it is decreasing.
CTR_max	Time-base counter equal max value. (TBCNT = FFFFh) Generated event when the EPWM_TBCNT value reaches its maximum value. This signal is only used only as a status bit.
TBCLK	Time-base clock. This is a prescaled version of the system clock - SYSCLKOUT(1) and is used by all submodules within the ePWMn. This clock determines the rate at which time-base counter increments or decrements.

(1) The system clock - SYSCLKOUT is the ePWM functional clock derived from the PWMSSn gateable interface and functional clock PWMSSn_GICLK, described in Section 29.1.3.

29.2.9 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (EPWM_TBPRD) register and the mode of the time-base counter. Figure 29-10 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count, down-count, and up-down-count time-base counter modes when when the period is set to 4 (EPWM_TBPRD register bitfield TBPRD = 0x4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the system clock (SYSCLKOUT).

The time-base counter has three modes of operation selected by the time-base control register (EPWM_TBCTL):

- **Up-Down-Count Mode:** In up-down-count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until it reaches zero. At this point the counter repeats the pattern and begins to increment.
- **Up-Count Mode:** In this mode, the time-base counter starts from zero and increments until it reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.
- **Down-Count Mode:** In down-count mode, the time-base counter starts from the period (TBPRD) value and decrements until it reaches zero. When it reaches zero, the time-base counter is reset to the period value and it begins to decrement once again.

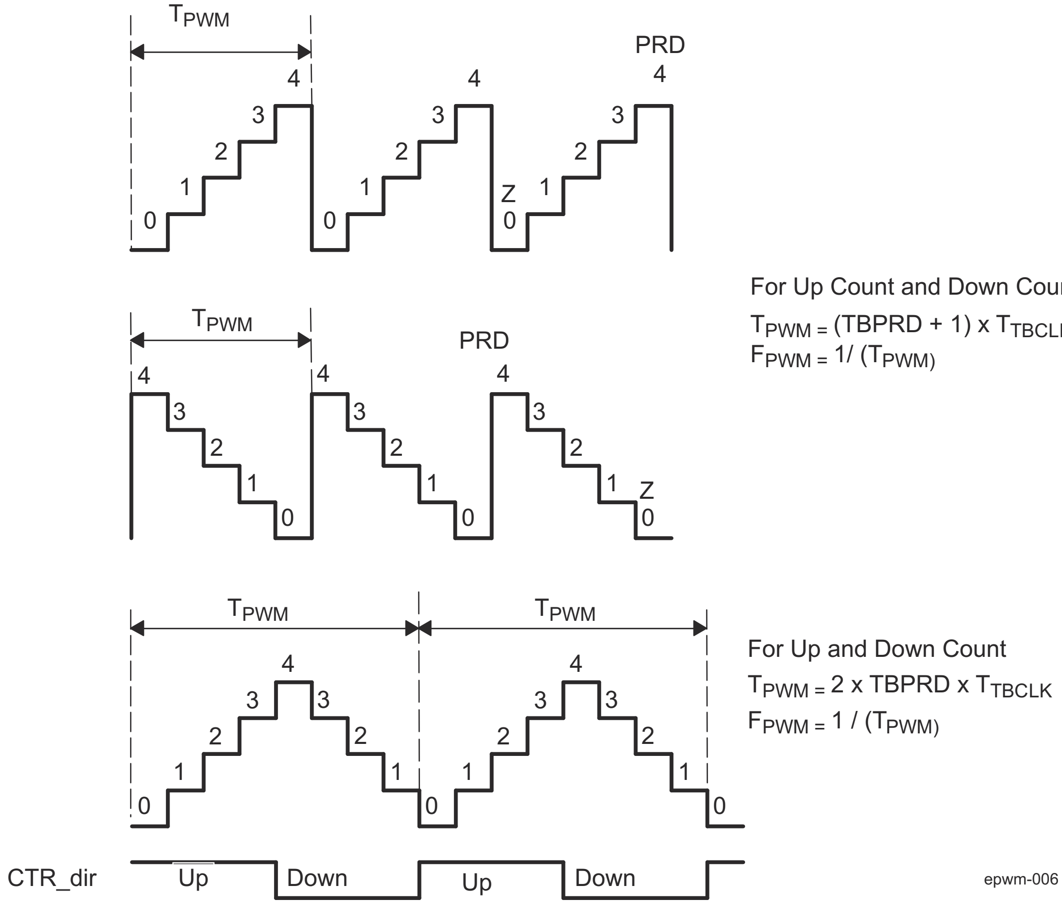


Figure 29-10. ePWM Time-Base Frequency and Period

29.2.10 ePWM Time-Base Period Shadow Register

The time-base period register (EPWM_TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the ePWM module:

- **Active Register:** The active register controls the hardware and is responsible for actions that the hardware causes or invokes.
- **Shadow Register:** The shadow register buffers or provides a temporary holding location for the active register. It has no direct effect on any control hardware. At a strategic point in time the shadow register's content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

The memory address of the shadow period register is the same as the active register. Which register is written to or read from is determined by the EPWM_TBCTL[3] PRDLD bit. This bit enables and disables the EPWM_TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:** The EPWM_TBPRD shadow register is enabled when EPWM_TBCTL[3] PRDLD = 0. Reads from and writes to the EPWM_TBPRD memory address go to the shadow register. The shadow register contents are transferred to the active register (EPWM_TBPRD (Active) ← EPWM_TBPRD (shadow)) when the time-base counter (register (EPWM_TBCNT) equals zero (TBCNT = 0000h). By default the EPWM_TBPRD shadow register is enabled.
- **Time-Base Period Immediate Load Mode:** If immediate load mode is selected (EPWM_TBCTL[3] PRDLD = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

29.2.11 ePWM Time-Base Counter Synchronization

A time-base synchronization scheme connects all of the ePWM modules on a device. Each ePWM module has a synchronization input (EPWMxSYNCI) and a synchronization output (EPWMxSYNCO). The input synchronization for the first instance (ePWM1) comes from an external pin . For the device ePWM environment sync pin details refer to the Section 29.1.2. The possible synchronization connections for the remaining ePWM modules is shown in Figure 29-11.

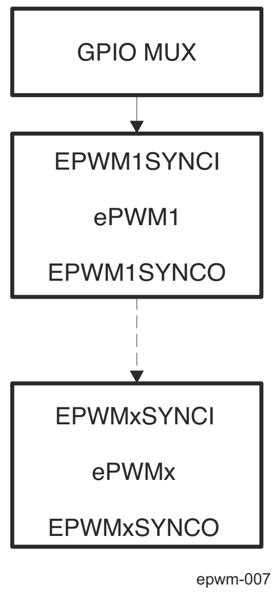


Figure 29-11. ePWM Time-Base Counter Synchronization Scheme 1

Each ePWM module can be configured to use or ignore the synchronization input. If the EPWM_TBCTL[2] PHSEN bit is set, then the time-base counter (TBCNT) of the ePWM module (register EPWM_TBCNT) will be automatically loaded with the phase register (EPWM_TBPHS) contents when one of the following conditions occur:

- **EPWMxSYNCI: Synchronization Input Pulse:** The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (EPWM_TBPHS → EPWM_TBCNT). This operation occurs on the next valid time-base clock (TBCLK) edge.
- **Software Forced Synchronization Pulse:** Writing a 1 to the EPWM_TBCTL[6] SWFSYNC control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on EPWMxSYNCI.

This feature enables the ePWM module to be automatically synchronized to the time base of another ePWM module. Lead or lag phase control can be added to the waveforms generated by different ePWM modules to synchronize them. In up-down-count mode, the EPWM_TBCTL[13] PHSDIR bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The TBPHS bit is ignored in count-up or count-down modes. See Figure 29-12 through Figure 29-15 for examples.

Clearing the EPWM_TBCTL[2] PHSEN bit configures the ePWM to ignore the synchronization input pulse. The synchronization pulse can still be allowed to flow-through to the EPWMxSYNCO and be used to synchronize other ePWM modules. In this way, you can set up a master time-base (for example, ePWM1) and downstream modules (ePWM2 - ePWMx) may elect to run in synchronization with the master.

29.2.12 Phase Locking the Time-Base Clocks of Multiple ePWM Modules

As already described in the Section 29.1.3.1.3, PWMSS1_TBCLKEN through PWMSS3_TBCLKEN bits in the CTRL_CORE_CONTROL_IO_2 register of the device Core Control Module can be used to individually control or globally synchronize the time-base clocks of all enabled ePWM modules on a device. When all PWMSSN_TBCLKEN (where n = 1 to 3) bits are set to 0b0, the time-base clocks of all ePWMx (where x = 1 to 3) modules are stopped (default). When all PWMSSn_TBCLKEN bits are simultaneously set in SW to 0b1, all ePWMx modules (x = 1 to 3) time-base clocks are started with the rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescaler bits in the EPWM_TBCTL register of each ePWM module must be set identically. The proper procedure for enabling the ePWM clocks is as follows:

1. Enable the ePWM module clocks.
2. Set PWMSSn_TBCLKEN = 0. This will stop the time-base clock within any enabled ePWMx module.
3. Configure the prescaler values and desired ePWM modes per each involved ePWMx.
4. Simultaneously set bits PWMSSn_TBCLKEN to 0b1 (where n = 1 to 3) 1.

29.2.13 ePWM Time-Base Counter Modes and Timing Waveforms

The time-base counter operates in one of four modes:

- Up-count mode which is asymmetrical.
- Down-count mode which is asymmetrical.
- Up-down-count which is symmetrical.
- Frozen where the time-base counter is held constant at the current value.

To illustrate the operation of the first three modes, Figure 29-12 to Figure 29-15 show when events are generated and how the time-base responds to an EPWMxSYNCI signal.

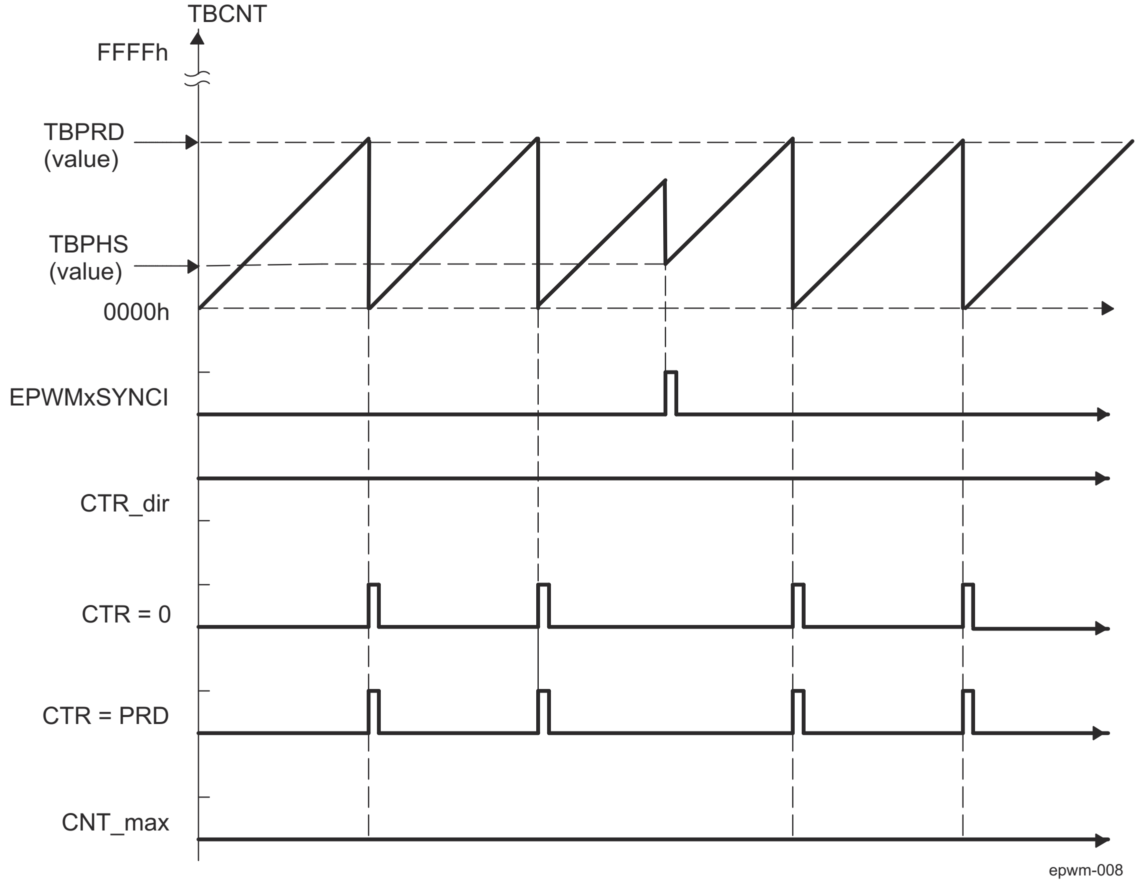


Figure 29-12. ePWM Time-Base Up-Count Mode Waveforms

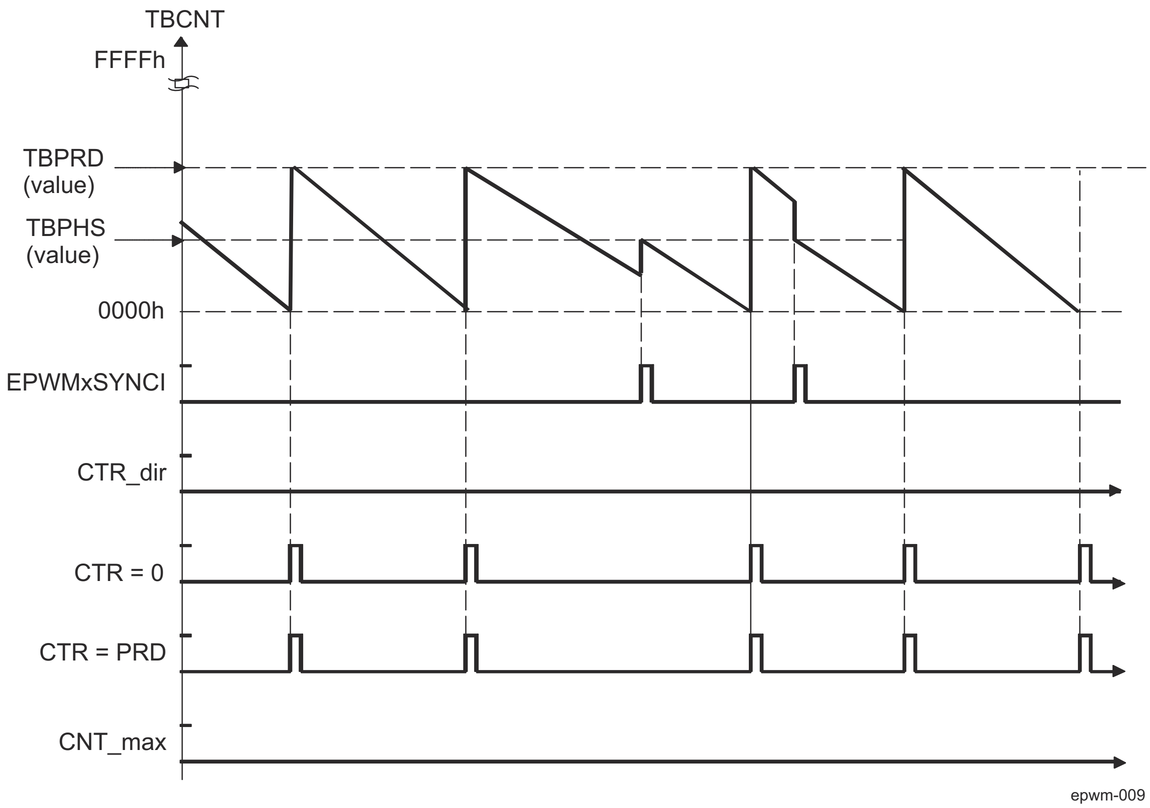


Figure 29-13. ePWM Time-Base Down-Count Mode Waveforms

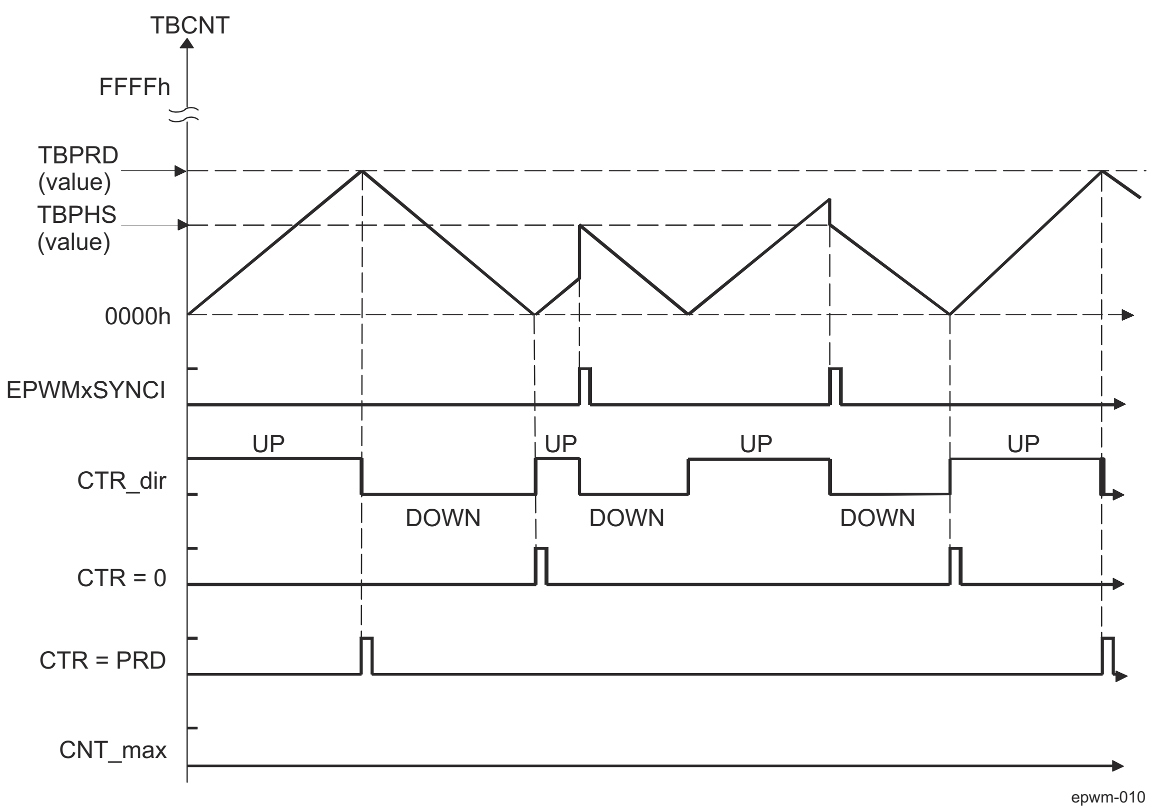


Figure 29-14. ePWM Time-Base Up-Down-Count Waveforms, EPWM_TBCTL[13] PHSDIR = 0 Count Down on

Synchronization Event

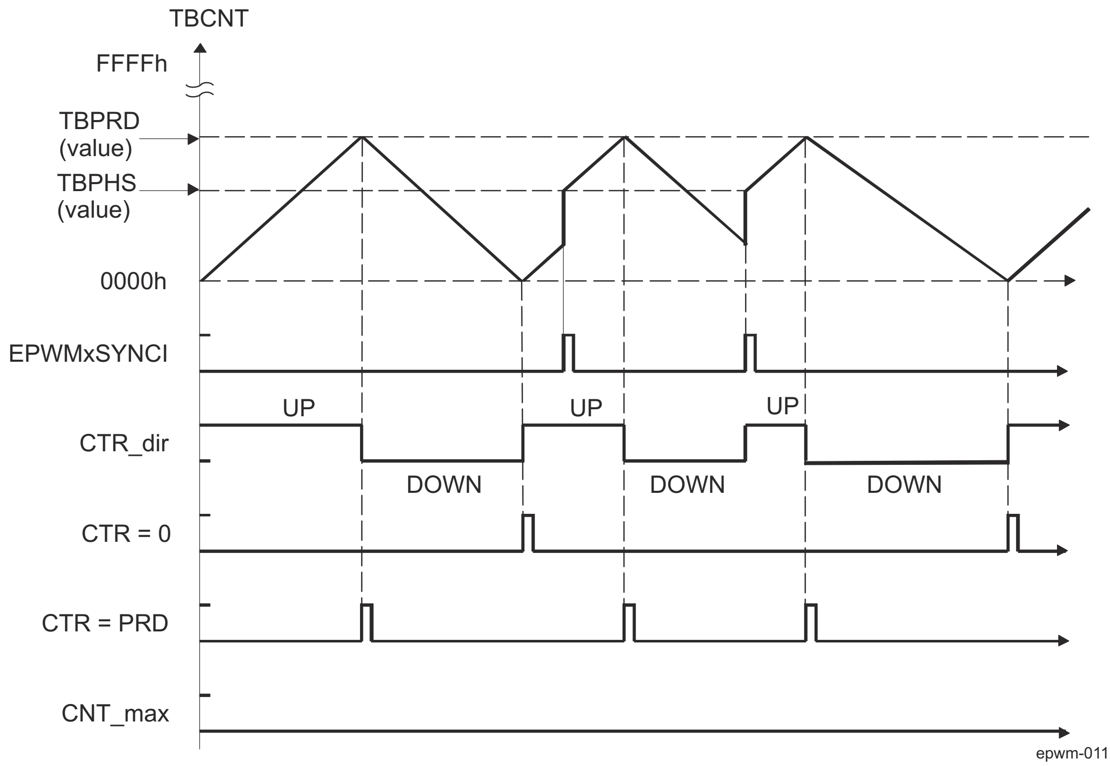


Figure 29-15. ePWM Time-Base Up-Down Count Waveforms, EPWM_TBCTL[13] PHSDIR = 1 Count Up on Synchronization Event

29.2.14 ePWM Counter-Compare (CC) Submodule

Figure 29-16 illustrates the counter-compare submodule within the ePWM. Figure 29-17 shows the basic structure of the counter-compare submodule.

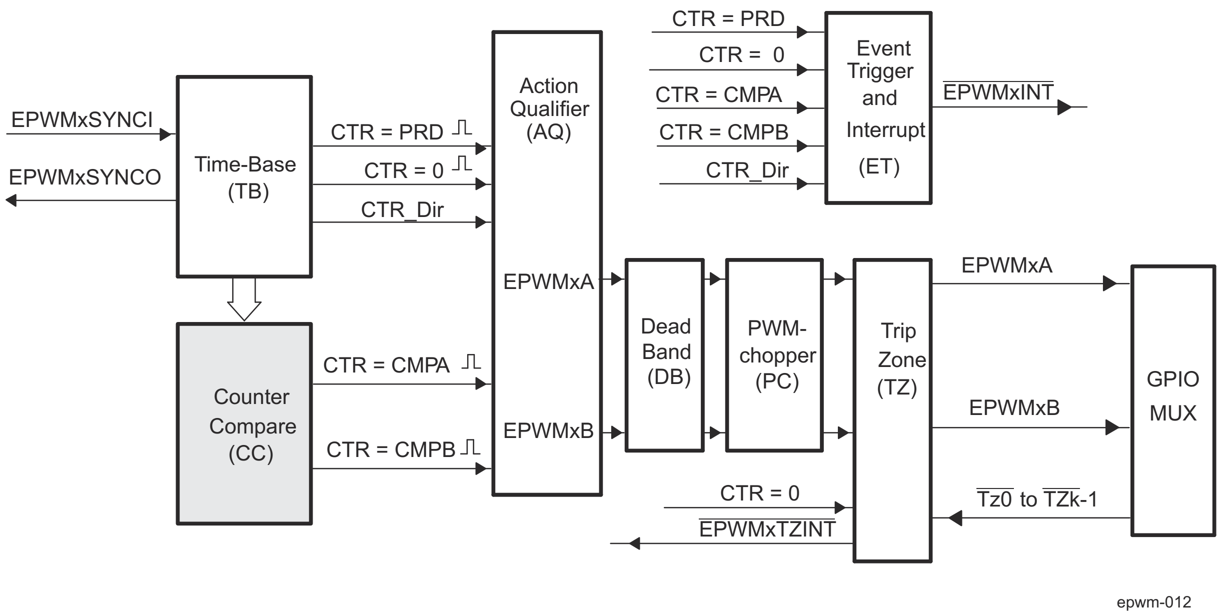


Figure 29-16. ePWM Counter-Compare Submodule

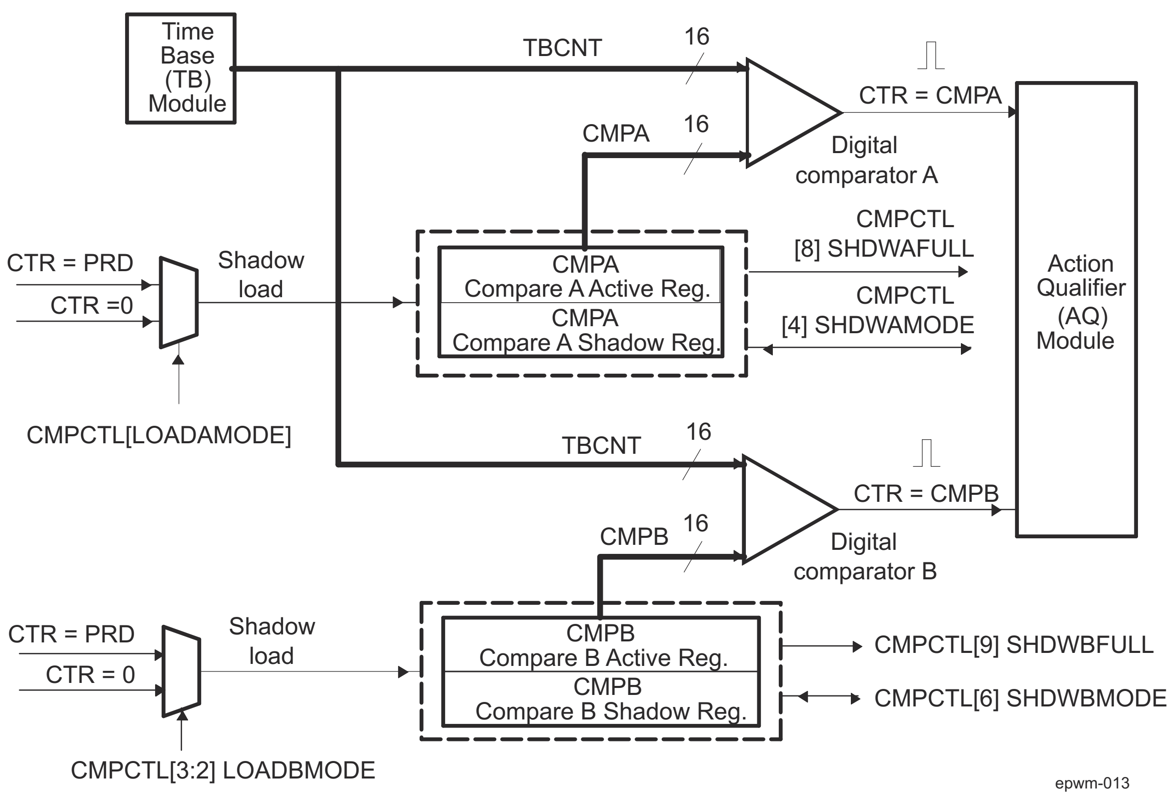


Figure 29-17. ePWM Counter-Compare Submodule Signals and Registers

29.2.15 Purpose of the ePWM Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (EPWM_CMPA) and counter-compare B (EPWM_CMPB) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event.

The counter-compare submodule:

- Generates events based on programmable time stamps using the EPWM_CMPA and EPWM_CMPB registers
  – TBCNT = CMPA: Time-base counter equals counter-compare A register (TBCNT = CMPA).
  – TBCNT = CMPB: Time-base counter equals counter-compare B register (TBCNT = CMPB)
- Controls the PWM duty cycle if the action-qualifier submodule is configured appropriately
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

29.2.16 Controlling and Monitoring the ePWM Counter-Compare Submodule

Table 29-18 lists the registers used to control and monitor the counter-compare submodule. Table 29-19 lists the key signals associated with the counter-compare submodule.

Table 29-18. ePWM Counter-Compare Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_CMPCTL	Counter-Compare Control Register.	Eh	No
HRPWM_CMPAHR	HRPWM Counter-Compare A Extension Register [(1)]	10h	Yes
EPWM_CMPA	Counter-Compare A Register	12h	Yes
EPWM_CMPB	Counter-Compare B Register	14h	Yes

(1) This register is available only on ePWM modules with the high-resolution extension (HRPWM). On ePWM modules that do not include the HRPWM, this location is reserved. See Section 29.1.3 to determine which ePWM instances include this feature.

Table 29-19. ePWM Counter-Compare Submodule Key Signals

Signal	Description of Event	Register Bitfields Compared
TBCNT = CMPA	Time-base counter equal to the active counter-compare A value	TBCNT = CMPA
TBCNT = CMPB	Time-base counter equal to the active counter-compare B value	TBCNT = CMPB
TBCNT = PRD	Time-base counter equal to the active period. Used to load active counter-compare A and B registers from the shadow register	TBCNT = TBPRD
TBCNT = 0	Time-base counter equal to zero. Used to load active counter-compare A and B registers from the shadow register	TBCNT = 0000h

29.2.17 Operational Highlights for the ePWM Counter-Compare Submodule

The counter-compare submodule is responsible for generating two independent compare events based on two compare registers:

1. TBCNT = CMPA: Time-base counter equal to counter-compare A register (EPWM_TBCNT = EPWM_CMPA).
2. TBCNT = CMPB: Time-base counter equal to counter-compare B register (TBCNT = CMPB).

For up-count or down-count mode, each event occurs only once per cycle. For up-down-count mode each event occurs twice per cycle, if the compare value is between 0000h and TBPRD; and occurs once per cycle, if the compare value is equal to 0000h or equal to TBPRD. These events are fed into the action-qualifier submodule where they are qualified by the counter direction and converted into actions if enabled. Refer to Section 29.2.20 for more details.

The counter-compare registers EPWM_CMPA and EPWM_CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occurs at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. The memory address of the active register and the shadow register is identical. Which register is written to or read from is determined by the EPWM_CMPCTL[4] SHDWAMODE and EPWM_CMPCTL[6] SHDWBMODE bits. These bits enable and disable the EPWM_CMPA shadow register and EPWM_CMPB shadow register respectively. The behavior of the two load modes is described below:

- **Shadow Mode:** The shadow mode for the EPWM_CMPA is enabled by clearing the EPWM_CMPCTL[4] SHDWAMODE bit and the shadow register for CMPB is enabled by clearing the EPWM_CMPCTL[6] SHDWBMODE bit. Shadow mode is enabled by default for both EPWM_CMPA and EPWM_CMPB.

  If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events:
  – TBCNT = PRD: Time-base counter equal to the period (TBCNT = TBPRD).
  – TBCNT = 0: Time-base counter equal to zero (TBCNT = 0000h)
  – Both TBCNT = PRD and TBCNT = 0

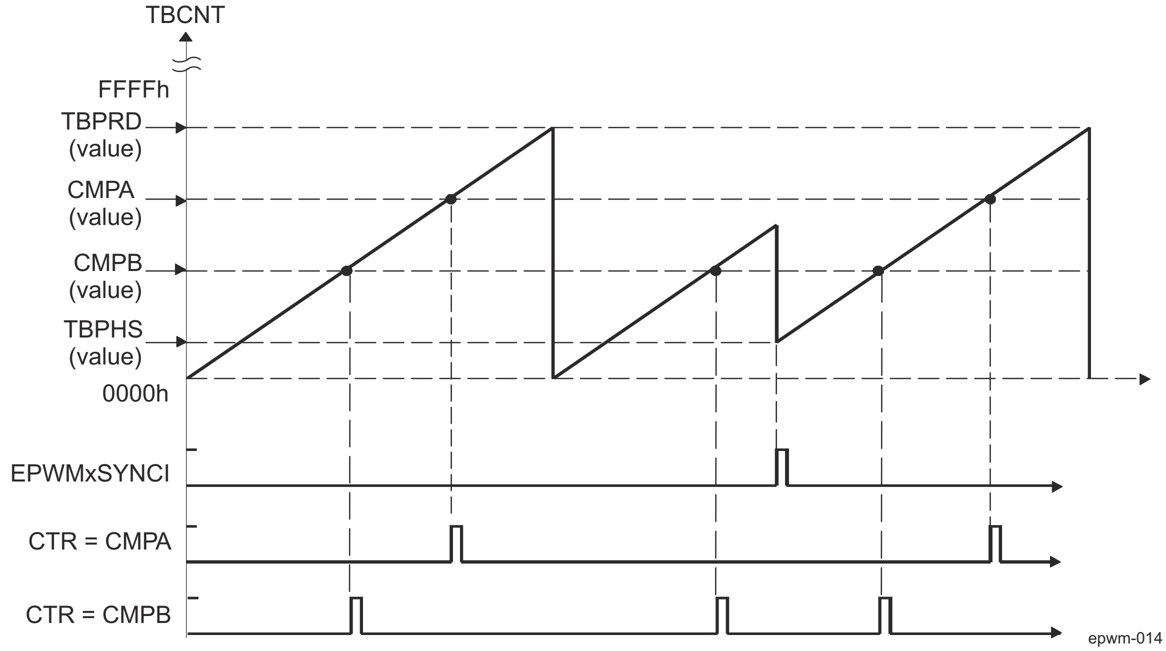
  Which of these three events is specified by the EPWM_CMPCTL[1:0] LOADAMODE and EPWM_CMPCTL[3:2] LOADBMODE register bits. Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.
- **Immediate Load Mode:** If immediate load mode is selected (EPWM_TBCTL[4] SHDWAMODE = 1 or EPWM_TBCTL[6] SHDWBMODE = 1), then a read from or a write to the register will go directly to the active register.

29.2.18 ePWM Count Mode Timing Waveforms

The counter-compare module can generate compare events in all three count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Down-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the first three modes, the timing diagrams in Figure 29-18 to Figure 29-21 show when events are generated and how the EPWMxSYNCI signal interacts.



An EPWMxSYNCI external synchronization event can cause a discontinuity in the TBCNT count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 29-18. ePWM Counter-Compare Event Waveforms in Up-Count Mode

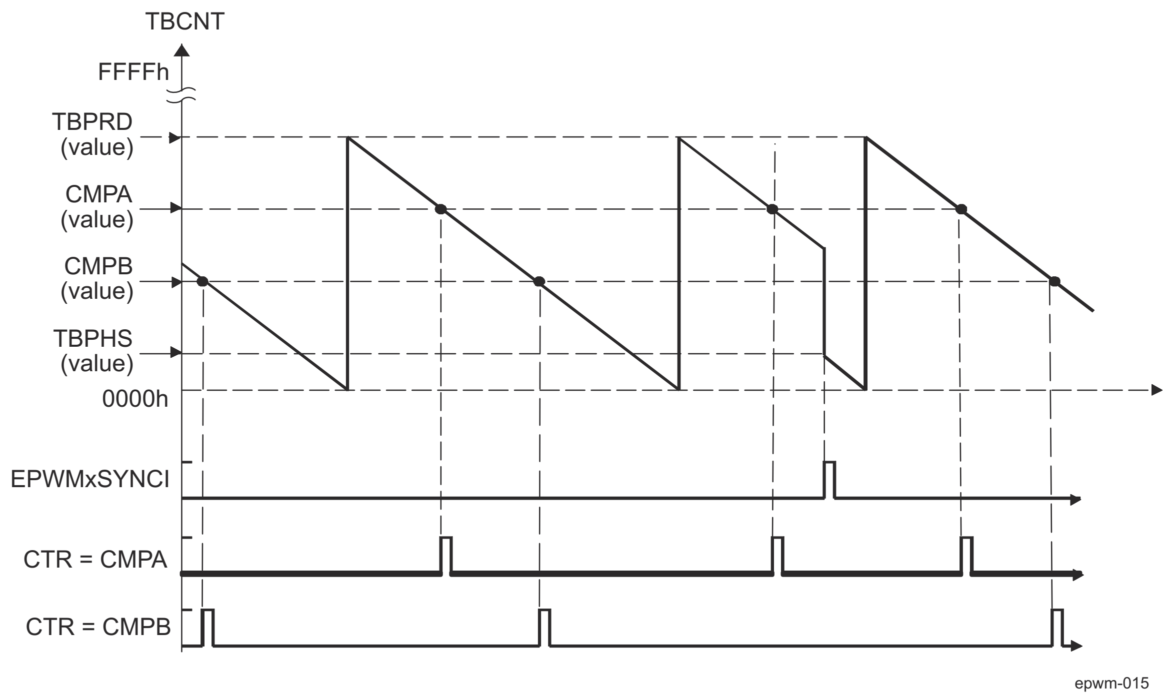


Figure 29-19. ePWM Counter-Compare Events in Down-Count Mode

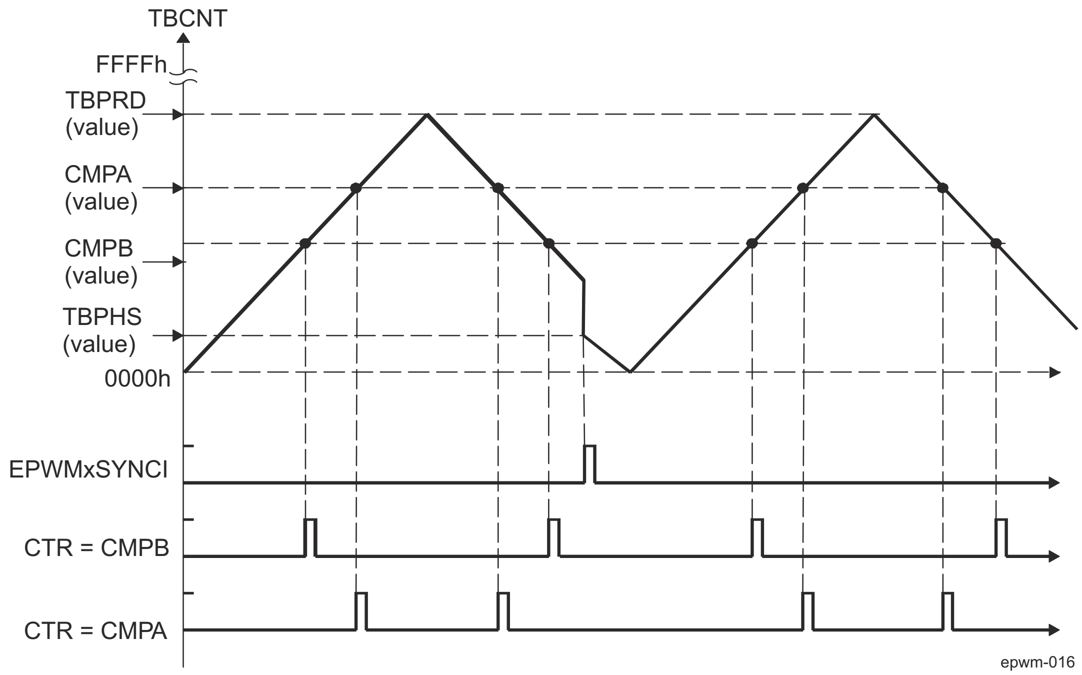


Figure 29-20. ePWM Counter-Compare Events in Up-Down-Count Mode, EPWM_TBCTL[13] PHSDIR = 0 Count Down on Synchronization Event

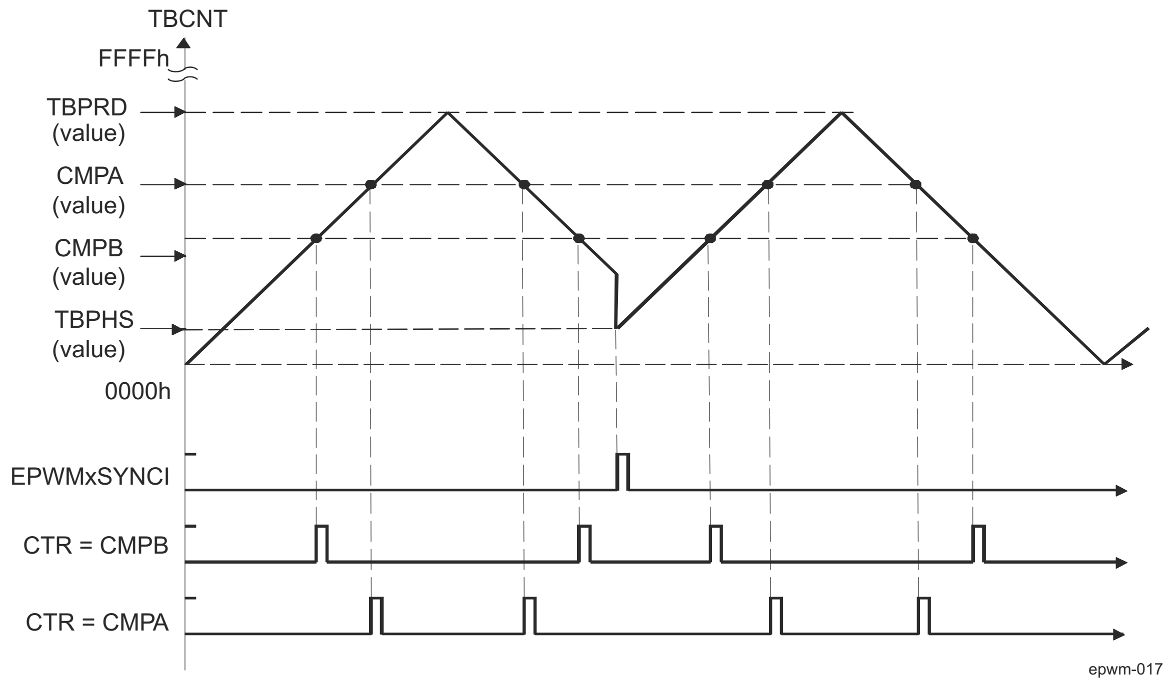


Figure 29-21. ePWM Counter-Compare Events in Up-Down-Count Mode, EPWM_TBCTL[13] PHSDIR = 1 Count Up on Synchronization Event

29.2.19 ePWM Action-Qualifier (AQ) Submodule

Figure 29-22 shows the action-qualifier (AQ) submodule (see shaded block) in the ePWM system. The action-qualifier submodule has the most important role in waveform construction and PWM generation. It decides which events are converted into various action types, thereby producing the required switched waveforms at the EPWMxA and EPWMxB outputs.

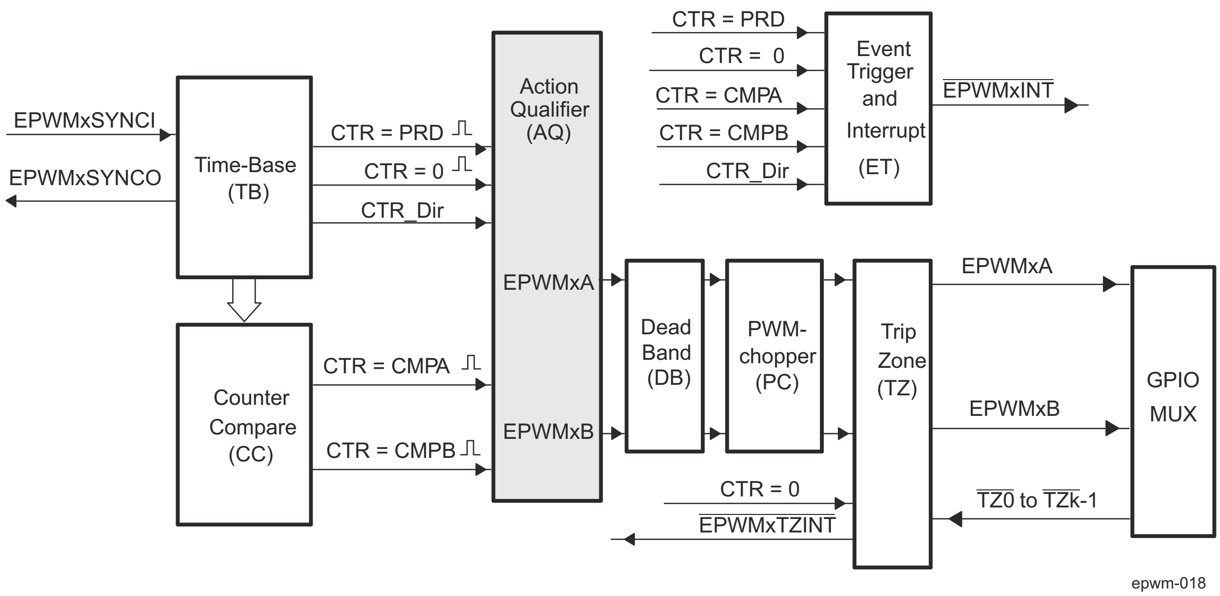


Figure 29-22. ePWM Action-Qualifier Submodule

29.2.20 Purpose of the ePWM Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  – TBCNT = PRD: Time-base counter equal to the period (TBCNT = TBPRD)
  – TBCNT = 0: Time-base counter equal to zero (TBCNT = 0000h)
  – TBCNT = CMPA: Time-base counter equal to the counter-compare A register (TBCNT = CMPA)
  – TBCNT = CMPB: Time-base counter equal to the counter-compare B register (TBCNT = CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when it is decreasing.

29.2.21 Controlling and Monitoring the ePWM Action-Qualifier Submodule

Table 29-20 lists the registers used to control and monitor the action-qualifier submodule.

Table 29-20. Action-Qualifier Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_AQCTLA	Action-Qualifier Control Register For Output A (EPWMxA)	16h	No
EPWM_AQCTLB	Action-Qualifier Control Register For Output B (EPWMxB)	18h	No
EPWM_AQSFRC	Action-Qualifier Software Force Register	1Ah	No
EPWM_AQCSFRC	Action-Qualifier Continuous Software Force	1Ch	Yes

The action-qualifier submodule is based on event-driven logic. It can be thought of as a programmable cross switch with events at the input and actions at the output, all of which are software controlled via the set of registers shown in Figure 29-23. The possible input events are summarized again in Table 29-21.

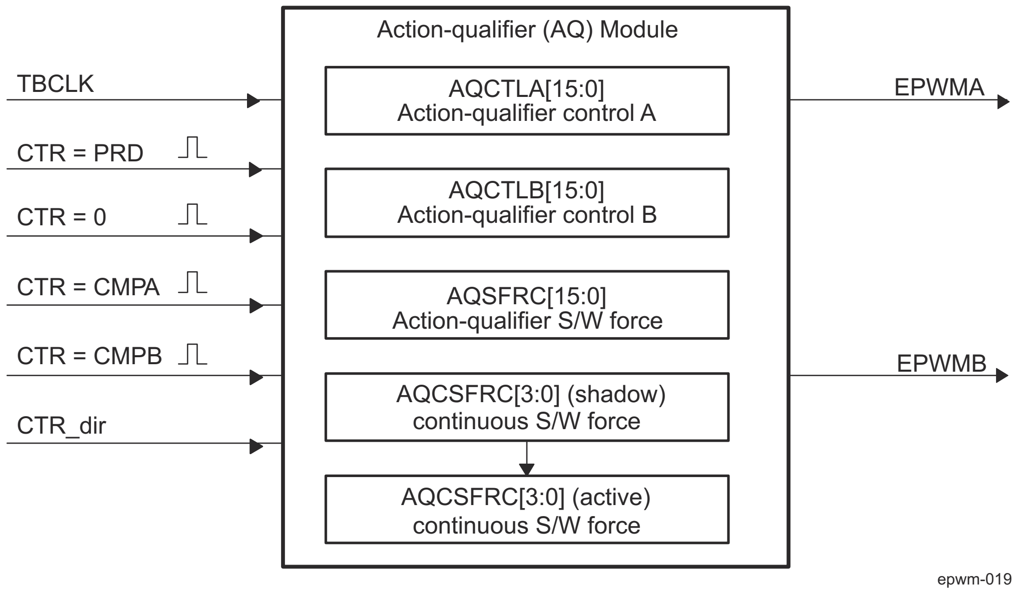


Figure 29-23. ePWM Action-Qualifier Submodule Inputs and Outputs

Table 29-21. ePWM Action-Qualifier Submodule Possible Input Events

Signal	Description	Register Bitfield Compared
TBCNT = PRD	Time-base counter equal to the period value	TBCNT = TBPRD
TBCNT = 0	Time-base counter equal to zero	TBCNT = 0000h
TBCNT = CMPA	Time-base counter equal to the counter-compare A	TBCNT = CMPA
TBCNT = CMPB	Time-base counter equal to the counter-compare B	TBCNT = CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by registers EPWM_AQSFRC and EPWM_AQCSFRC.

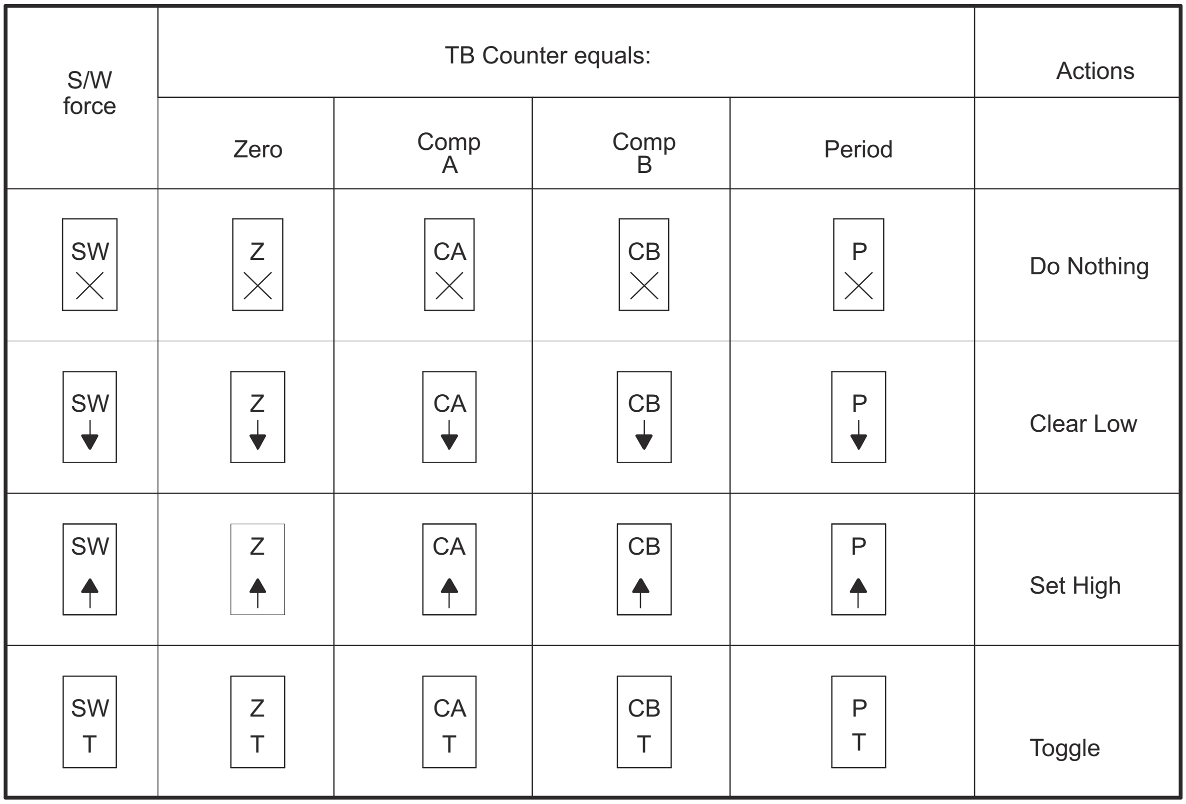
The action-qualifier submodule controls how the two outputs EPWMxA and EPWMxB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs EPWMxA and EPWMxB are:

- **Set High:** Set output EPWMxA or EPWMxB to a high level.
- **Clear Low:** Set output EPWMxA or EPWMxB to a low level.
- **Toggle:** If EPWMxA or EPWMxB is currently pulled high, then pull the output low. If EPWMxA or EPWMxB is currently pulled low, then pull the output high.
- **Do Nothing:** Keep outputs EPWMxA and EPWMxB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the EPWMxA and EPWMxB outputs, this event can still trigger interrupts. See the event-trigger submodule description in Section 29.2.41 for details.

Actions are specified independently for either output (EPWMxA or EPWMxB). Any or all events can be configured to generate actions on a given output. For example, both TBCNT = CMPA and TBCNT = CMPB can operate on output EPWMxA. All qualifier actions are configured via the control registers found at the end of this section.

For clarity, the drawings in this chapter use a set of symbolic actions. These symbols are summarized in Figure 29-24. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and their time positions are programmed via the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"; it is the default at reset.



epwm-020

Figure 29-24. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs

29.2.22 ePWM Action-Qualifier Event Priority

It is possible for the ePWM action qualifier to receive more than one event at the same time. In this case events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down-count mode are shown in Table 29-22. A priority level of 1 is the highest priority and level 7 is the lowest. The priority changes slightly depending on the direction of TBCNT.

Table 29-22. ePWM Action-Qualifier Event Priority for Up-Down-Count Mode

Priority Level	Event if TBCNT is Incrementing TBCNT = 0 up to TBCNT = TBPRD	Event if TBCNT is Decrementing TBCNT = TBPRD down to TBCNT = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4	Counter equals zero	Counter equals period (TBPRD in EPWM_TBPRD active register)
5	Counter equals CMPB on down-count (CBD) [(1)]	Counter equals CMPB on up-count (CBU) [(1)]
6 (Lowest)	Counter equals CMPA on down-count (CAD) [(1)]	Counter equals CMPA on up-count (CBU) [(1)]

(1) To maintain symmetry for up-down-count mode, both up-events (CAU/CBU) and down-events (CAD/CBD) can be generated for TBPRD. Otherwise, up-events can occur only when the counter is incrementing and down-events can occur only when the counter is decrementing.

Table 29-23 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up and thus down-count events will never be taken.

Table 29-23. ePWM Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

Table 29-24 shows the action-qualifier priority for down-count mode. In this case, the counter direction is always defined as down and thus up-count events will never be taken.

Table 29-24. ePWM Action-Qualifier Event Priority for Down-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to Zero
3	Counter equal to CMPB on down-count (CBD)
4	Counter equal to CMPA on down-count (CAD)
5 (Lowest)	Counter equal to period (TBPRD)

It is possible to set the compare value greater than the period. In this case the action will take place as shown in Table 29-25.

Table 29-25. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAU/CBU	Compare on Down-Count Event CAU/CBU
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event will not occur.	Never occurs.
Down-Count Mode	Never occurs.	If CMPA/CMPB < TBPRD, the event will occur on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).
Up-Down-Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCNT = TBPRD).

29.2.23 Waveforms for Common ePWM Configurations

Note

The waveforms in this chapter show the ePWMs behavior for a static compare register value. In a running system, the active compare registers (EPWM_CMPA and EPWM_CMPB) are typically updated from their respective shadow registers once every period. The user specifies when the update will take place; either when the time-base counter reaches zero or when the time-base counter reaches period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down-count mode to generate a symmetric PWM:

- If you load EPWM_CMPA/EPWM_CMPB on zero, then use EPWM_CMPA/EPWM_CMPB values greater than or equal to 1.
- If you load EPWM_CMPA/EPWM_CMPB on period, then use EPWM_CMPA/EPWM_CMPB values less than or equal to TBPRD - 1.

  This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down-count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM use the following configuration: Load EPWM_CMPA/ EPWM_CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

When using up-count mode to generate an asymmetric PWM:

- To achieve 0-100% asymmetric PWM use the following configuration: Load EPWM_CMPA/ EPWM_CMPB on TBPRD. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD+1 to achieve 0-100% PWM duty.

Figure 29-25 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCNT. In this mode 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, CMPA is used to make the comparison. When the counter is incrementing the CMPA match will pull the PWM output high. Likewise, when the counter is decrementing the compare match will pull the PWM signal low. When CMPA = 0, the PWM signal is low for the entire period giving the 0% duty waveform. When EPWM_CMPA = EPWM_TBPRD, the PWM signal is high achieving 100% duty.

When using this configuration in practice, if you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1. If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1. This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

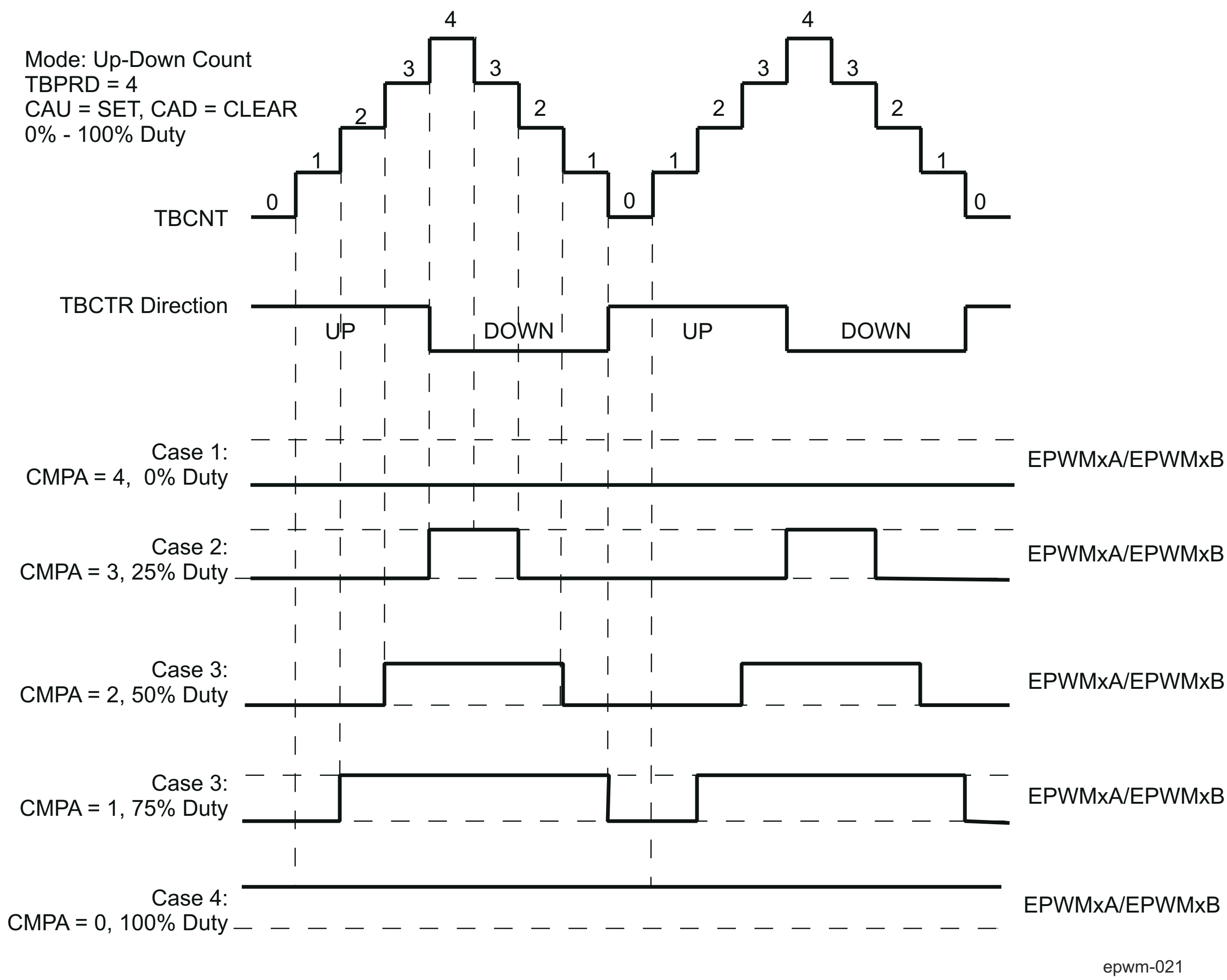
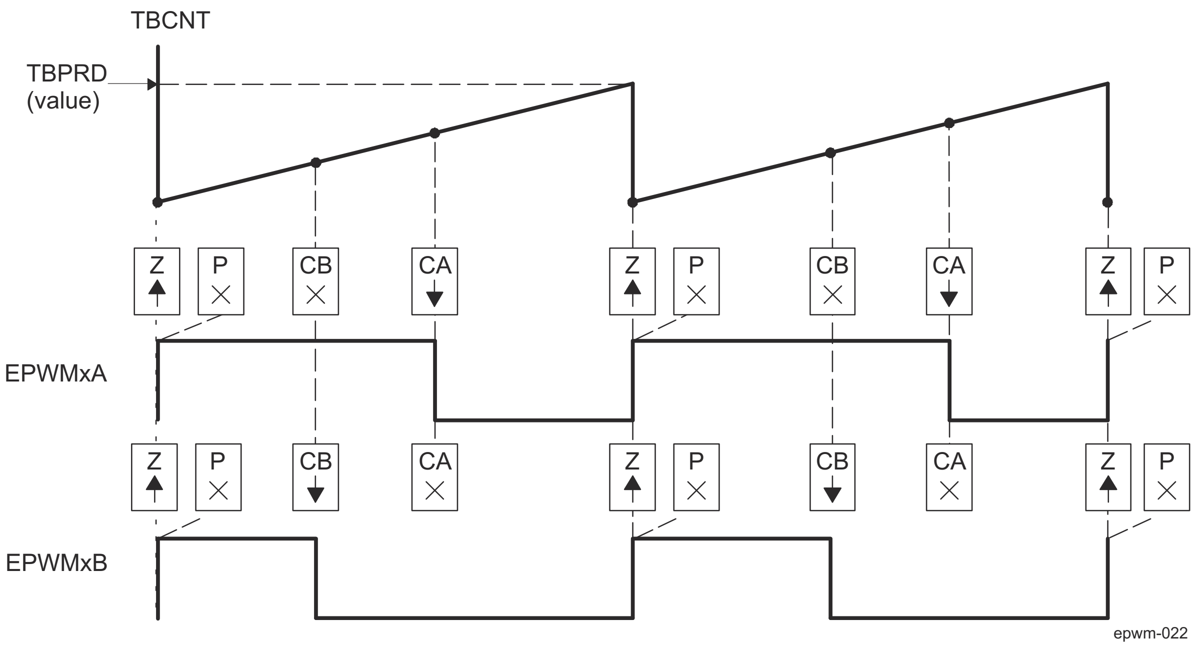


Figure 29-25. ePWM Up-Down-Count Mode Symmetrical Waveform

The PWM waveforms in Figure 29-26 through Figure 29-31 show some common action-qualifier configurations. Some conventions used in the figures are as follows:

- TBPRD, CMPA, and CMPB refer to the value written in their respective registers (EPWM_TBPRD, EPWM_CMPA, and EPWM_CMPB). The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either CMPA or CMPB.
- EPWMxA and EPWMxB refer to the output signals from ePWMx
- Up-Down means Count-up-and-down mode, Up means up-count mode and Dwn means down-count mode
- Sym = Symmetric, Asym = Asymmetric

Table 29-26 and Table 29-27 contains initialization and runtime register configurations for the waveforms in Figure 29-26.



A. PWM period = (TBPRD + 1) × T_{TBCLK}

B. Duty modulation for EPWMxA is set by CMPA, and is active high (that is, high time duty proportional to CMPA).

C. Duty modulation for EPWMxB is set by CMPB and is active high (that is, high time duty proportional to CMPB).

D. The "Do Nothing" actions (X) are shown for completeness, but will not be shown on subsequent diagrams.

E. Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

Figure 29-26. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High

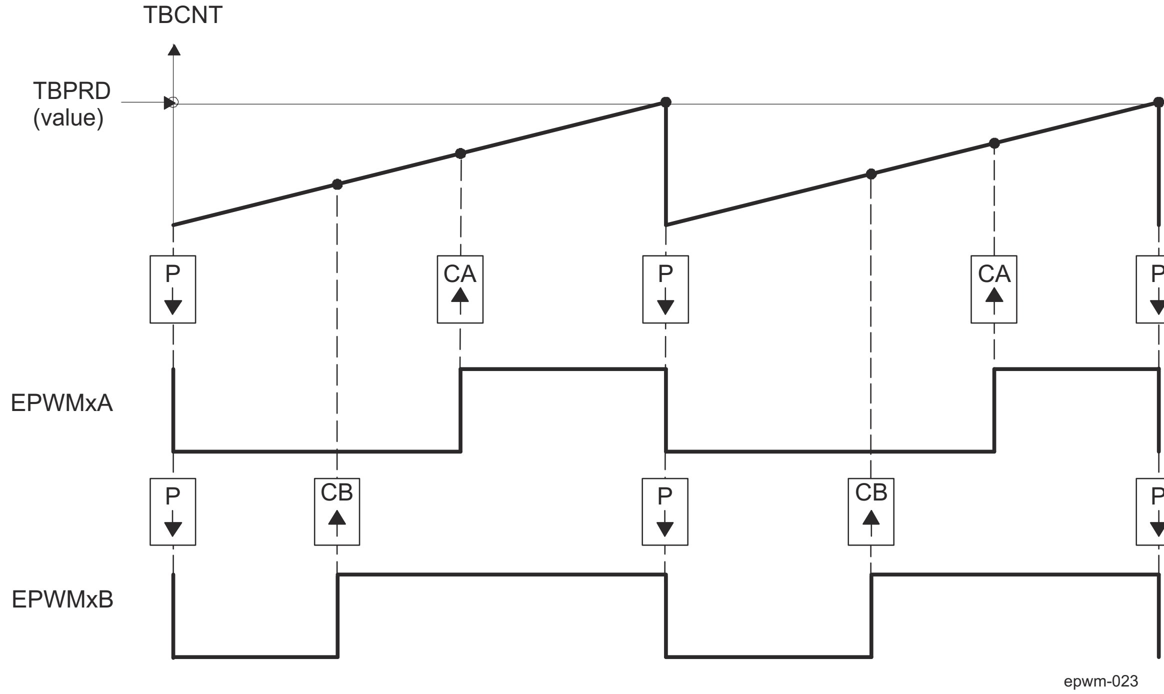
Table 29-26. EPWMx Initialization for Figure 29-26

Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
EPWM_CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	ZRO	AQ_SET	
	CAU	AQ_CLEAR	
EPWM_AQCTLB	ZRO	AQ_SET	
	CBU	AQ_CLEAR	

Table 29-27. EPWMx Run Time Changes for Figure 29-26

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 29-28 and Table 29-29 contains initialization and runtime register configurations for the waveforms in Figure 29-27.



A. PWM period = (TBPRD + 1) × T_{TBCLK}
B. Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).
C. Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).
D. The Do Nothing actions (X) are shown for completeness here, but will not be shown on subsequent diagrams.
E. Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

Figure 29-27. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low

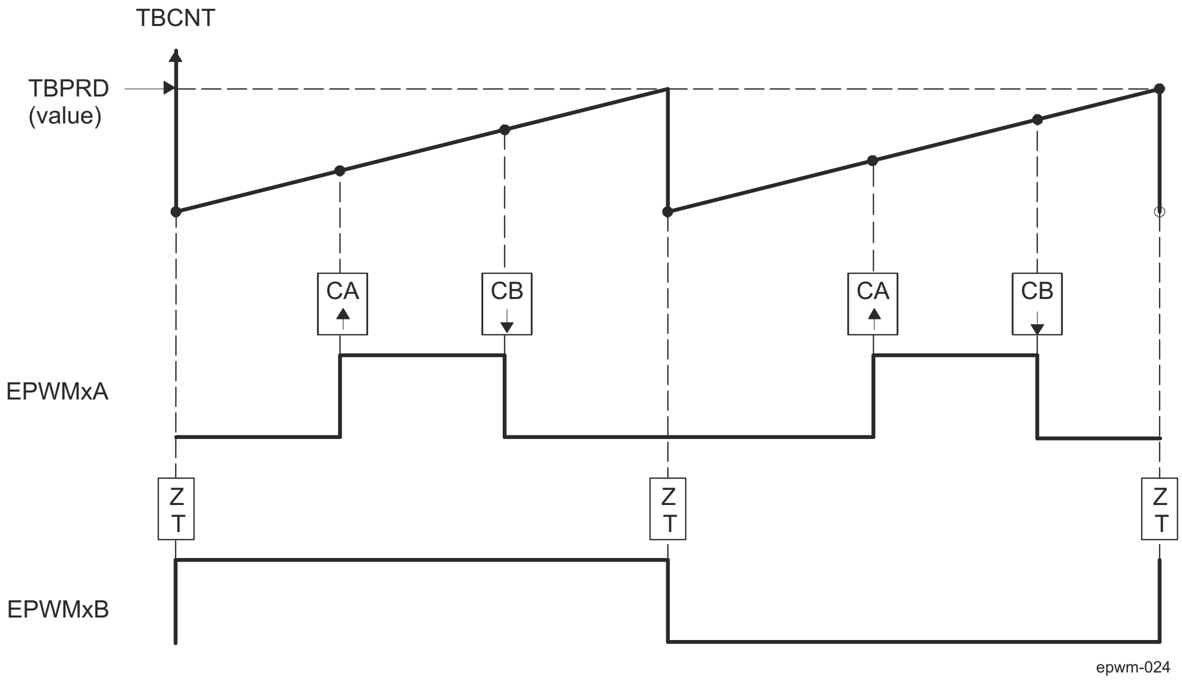
Table 29-28. EPWMx Initialization for Figure 29-27

Register	Bitfiled	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
EPWM_CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	
EPWM_AQCTLB	PRD	AQ_CLEAR	
	CBU	AQ_SET	

Table 29-29. EPWMx Run Time Changes for Figure 29-27

Register	Bit	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 29-30 and Table 29-31 contains initialization and runtime register configurations for the waveforms Figure 29-28. Use the code in Section 29.2.4 to define the headers.



A. PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})

B. Pulse can be placed anywhere within the PWM cycle (0000h - TBPRD)

C. High time duty proportional to (CMPB - CMPA)

D. EPWMxB can be used to generate a 50% duty square wave with frequency = 1/2 × ((TBPRD + 1) × TBCLK)

Figure 29-28. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA

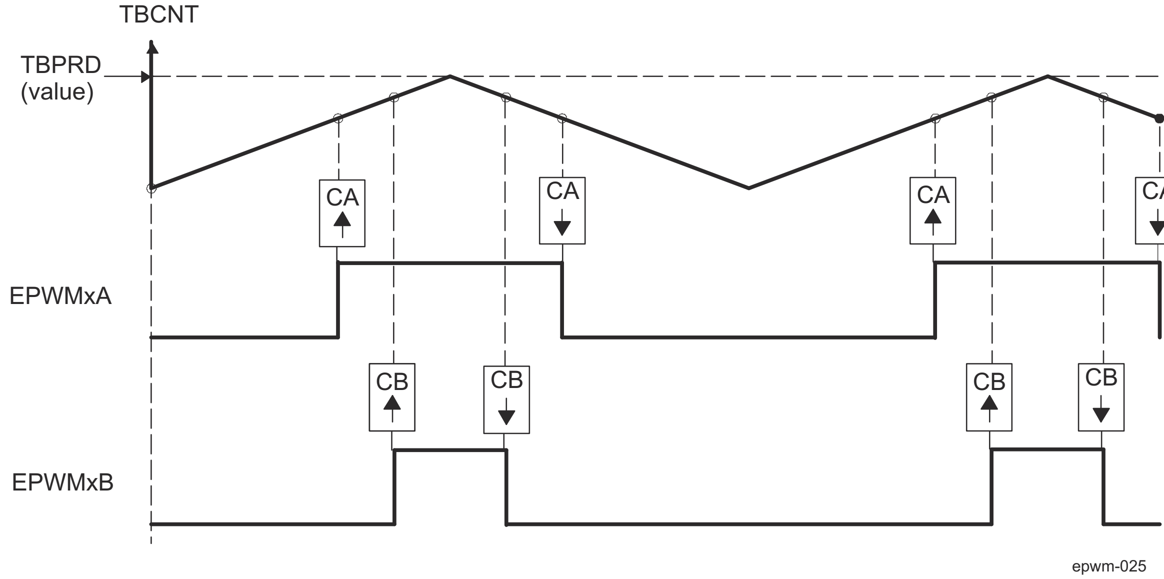
Table 29-30. EPWMx Initialization for Figure 29-28

Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	200 (C8h)	Compare A = 200 TBCLK counts
EPWM_CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CBU	AQ_CLEAR	
EPWM_AQCTLB	ZRO	AQ_TOGGLE	

Table 29-31. EPWMx Run Time Changes for Figure 29-28

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	EdgePosB	

Table 29-32 and Table 29-33 contains initialization and runtime register configurations for the waveforms in Figure 29-29. Use the code in Section 29.2.4 to define the headers.



A. PWM period = 2 x TBPRD × T_{TBCLK}

B. Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

C. Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

D. Outputs EPWMxA and EPWMxB can drive independent power switches

Figure 29-29. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low

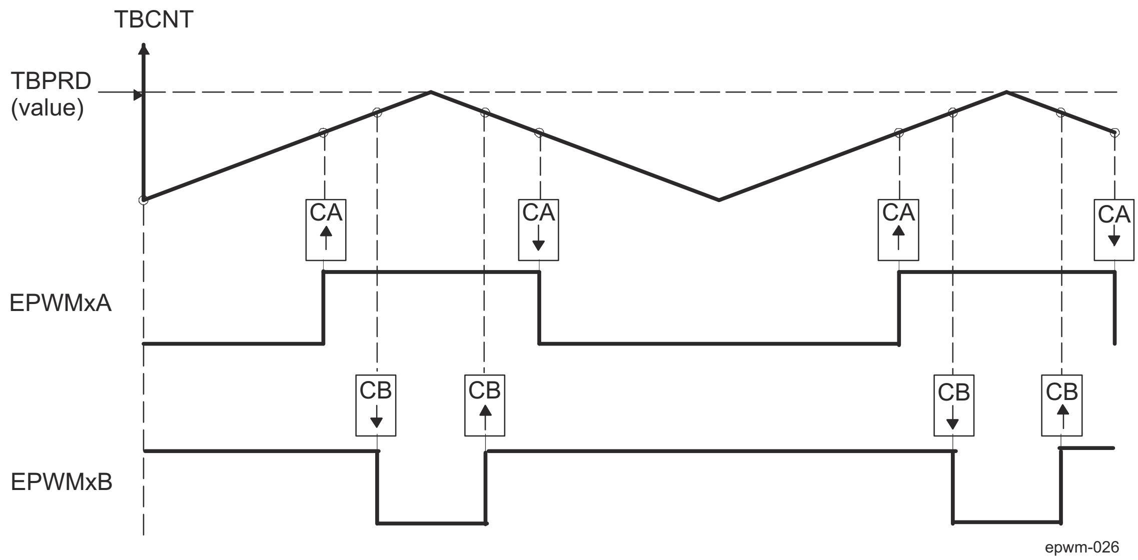
Table 29-32. EPWMx Initialization for Figure 29-29

Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	400 (190h)	Compare A = 400 TBCLK counts
EPWM_CMPB	CMPB	500 (1F4h)	Compare B = 500 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
EPWM_AQCTLB	CBU	AQ_SET	
	CBD	AQ_CLEAR	

Table 29-33. EPWMx Run Time Changes for Figure 29-29

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 29-34 and Table 29-35 contains initialization and runtime register configurations for the waveforms in Figure 29-30. Use the code in Section 29.2.4 to define the headers.



A. PWM period = 2 × TBPRD × T_{TBCLK}
B. Duty modulation for EPWMxA is set by CMPA, and is active low, i.e., low time duty proportional to CMPA
C. Duty modulation for EPWMxB is set by CMPB and is active high, i.e., high time duty proportional to CMPB
D. Outputs EPWMx can drive upper/lower (complementary) power switches
E. Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

Figure 29-30. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary

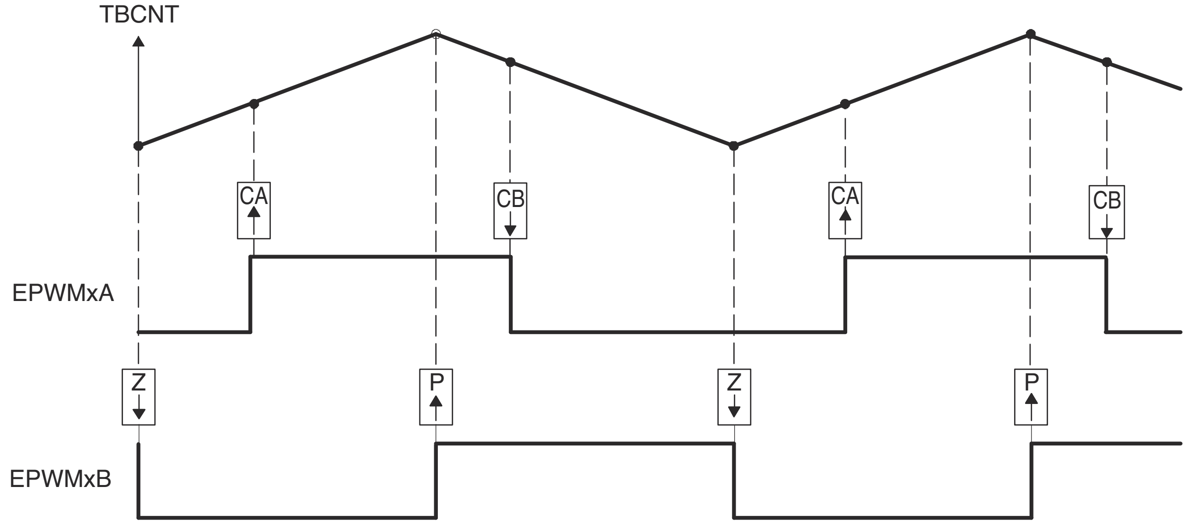
Table 29-34. EPWMx Initialization for Figure 29-30

Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
EPWM_CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
EPWM_AQCTLB	CBU	AQ_CLEAR	
	CBD	AQ_SET	

Table 29-35. EPWMx Run Time Changes for Figure 29-30

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 29-36 and Table 29-37 contains initialization and runtime register configurations for the waveforms in Figure 29-31. Use the code in Section 29.2.4 to define the headers.



A. PWM period = 2 × TBPRD × TBCLK
B. Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.
C. Duty modulation for EPWMxA is set by CMPA and CMPB.
D. Low time duty for EPWMxA is proportional to (CMPA + CMPB).
E. To change this example to active high, CMPA and CMPB actions need to be inverted (i.e., Set ! Clear and Clear Set).
F. Duty modulation for EPWMxB is fixed at 50% (utilizes spare action resources for EPWMxB)

Figure 29-31. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low

Table 29-36. EPWMx Initialization for Figure 29-31

Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	250 (FAh)	Compare A = 250 TBCLK counts
EPWM_CMPB	CMPB	450 (1C2h)	Compare B = 450 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CBD	AQ_CLEAR	
EPWM_AQCTLB	ZRO	AQ_CLEAR	
	PRD	AQ_SET	

Table 29-37. EPWMx Run Time Changes for Figure 29-31

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	EdgePosB	

29.2.24 ePWM Dead-Band Generator (DB) Submodule

Figure 29-32 illustrates the dead-band generator submodule within the ePWM module.

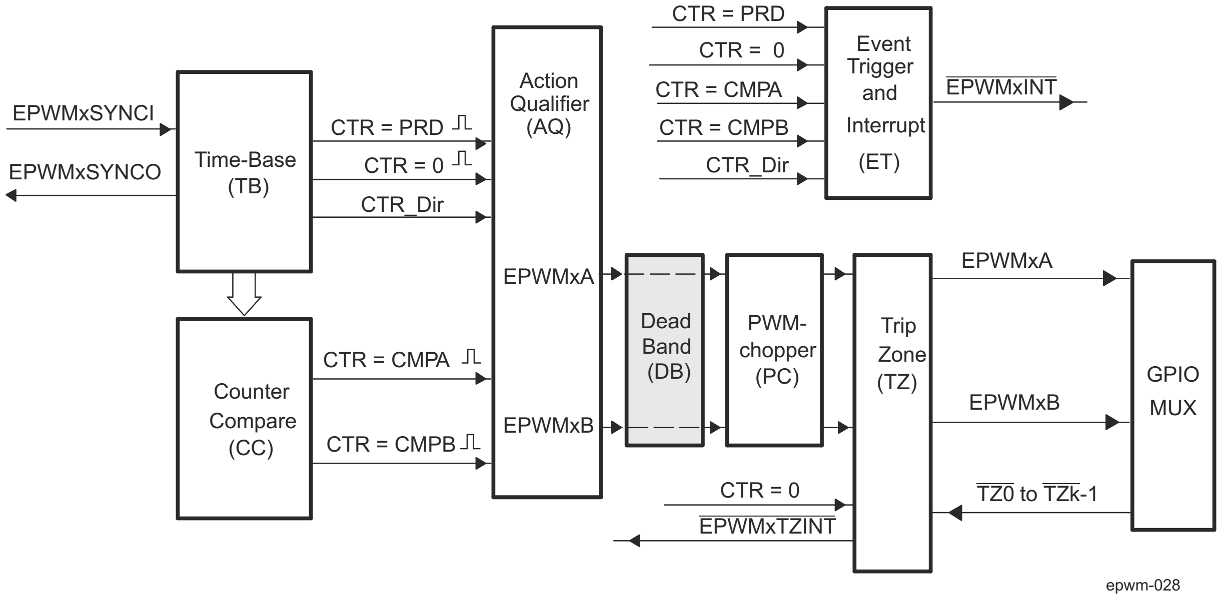


Figure 29-32. Dead-Band Generator Submodule

29.2.25 Purpose of the ePWM Dead-Band Submodule

The "Action-qualifier (AQ) Module" section discussed how it is possible to generate the required dead-band by having full control over edge placement using both the CMPA and CMPB resources of the ePWM module. However, if the more classical edge delay-based dead-band with polarity control is required, then the dead-band generator submodule should be used.

The key functions of the dead-band generator submodule are:

- Generating appropriate signal pairs (EPWMxA and EPWMxB) with dead-band relationship from a single EPWMxA input
- Programming signal pairs for:
  – Active high (AH)
  – Active low (AL)
  – Active high complementary (AHC)
  – Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

29.2.26 Controlling and Monitoring the ePWM Dead-Band Submodule

The dead-band generator submodule operation is controlled and monitored via the following registers:

Table 29-38. Dead-Band Generator Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_DBCTL	Dead-Band Control Register	1Eh	No
EPWM_DBRED	Dead-Band Rising Edge Delay Count Register	20h	No
EPWM_DBFED	Dead-Band Falling Edge Delay Count Register	22h	No

29.2.27 Operational Highlights for the ePWM Dead-Band Generator Submodule

The following sections provide the operational highlights.

The dead-band submodule has two groups of independent selection options as shown in Figure 29-33.

- **Input Source Selection:** The input signals to the dead-band module are the EPWMxA and EPWMxB output signals from the action-qualifier. In this section they will be referred to as EPWMxA In and EPWMxB In. Using the EPWM_DBCTL[5:4] IN_MODE control bits, the signal source for each delay, falling-edge or rising-edge, can be selected:
  - EPWMxA In is the source for both falling-edge and rising-edge delay. This is the default mode.
  - EPWMxA In is the source for falling-edge delay, EPWMxB In is the source for rising-edge delay.
  - EPWMxA In is the source for rising edge delay, EPWMxB In is the source for falling-edge delay.
  - EPWMxB In is the source for both falling-edge and rising-edge delay.
- **Output Mode Control:** The output mode is configured by way of the EPWM_DBCTL[1:0] OUT_MODE bits. These bits determine if the falling-edge delay, rising-edge delay, neither, or both are applied to the input signals.
- **Polarity Control:** The polarity control (EPWM_DBCTL[3:2] POLSEL) allows you to specify whether the rising-edge delayed signal and/or the falling-edge delayed signal is to be inverted before being sent out of the dead-band submodule.

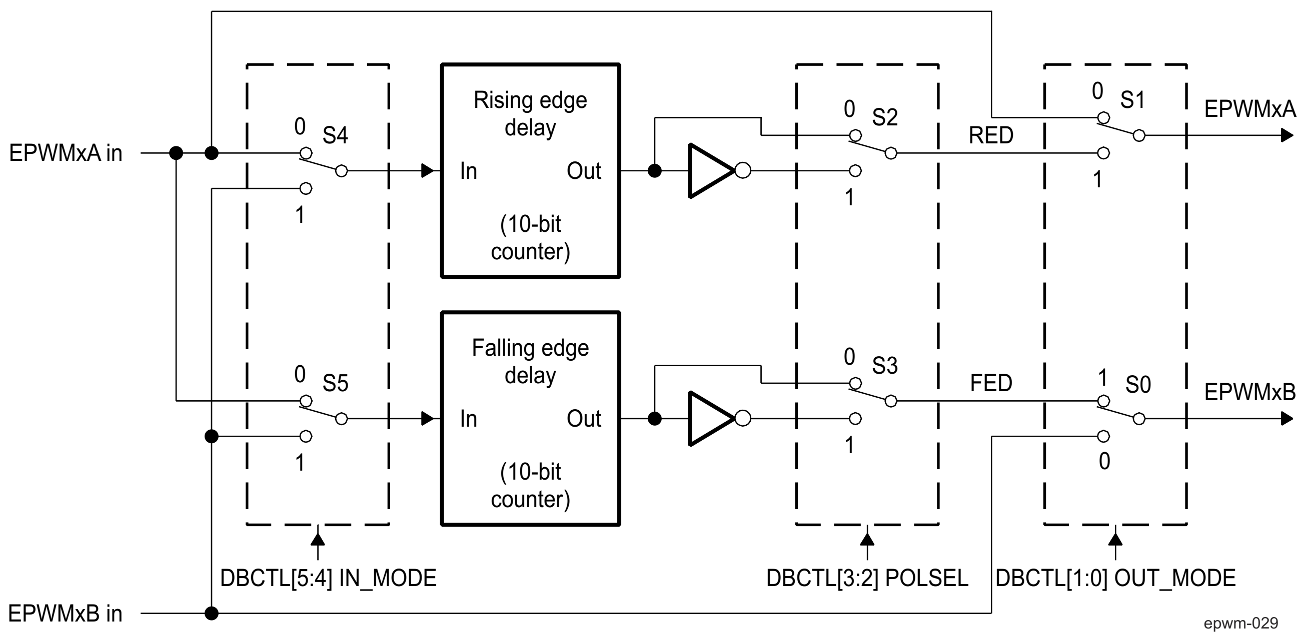


Figure 29-33. Configuration Options for the ePWM Dead-Band Generator Submodule

Although all combinations are supported, not all are typical usage modes. Table 29-39 lists some classical dead-band configurations. These modes assume that the EPWM_DBCTL[5:4] IN_MODE is configured such that EPWMxA In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 29-39 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED)** Allows you to fully disable the dead-band submodule from the PWM signal path.
- **Mode 2-5: Classical Dead-Band Polarity Settings** These represent typical polarity configurations that should address all the active high/low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 29-34. Note that to generate equivalent waveforms to Figure 29-34, configure the action-qualifier submodule to generate the signal as shown for EPWMxA.
- **Mode 6: Bypass rising-edge-delay and Mode 7: Bypass falling-edge-delay** Finally the last two entries in Table 29-39 show combinations where either the falling-edge-delay (FED) or rising-edge-delay (RED) blocks are bypassed.

Table 29-39. Classical Dead-Band Operating Modes

		EPWM_DBCTL[3:2] POLSEL		EPWM_DBCTL[1:0] OUT_MODE	
Mode	**Mode Description** (1)	**S3**	**S2**	**S1**	**S0**
1	EPWMxA and EPWMxB Passed Through (No Delay)	x	x	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	EPWMxA Out = EPWMxA In (No Delay) EPWMxB Out = EPWMxA In with Falling Edge Delay	0 or 1	0 or 1	0	1
7	EPWMxA Out = EPWMxA In with Rising Edge Delay EPWMxB Out = EPWMxB In with No Delay	0 or 1	0 or 1	1	0

(1) These are classical dead-band modes and assume that EPWM_DBCTL[5:4] IN_MODE = 0b00. That is, EPWMxA in is the source for both the falling-edge and rising-edge delays. Enhanced, non-traditional modes can be achieved by changing the IN_MODE configuration.

Figure 29-34 shows waveforms for typical cases where 0% < duty < 100%.

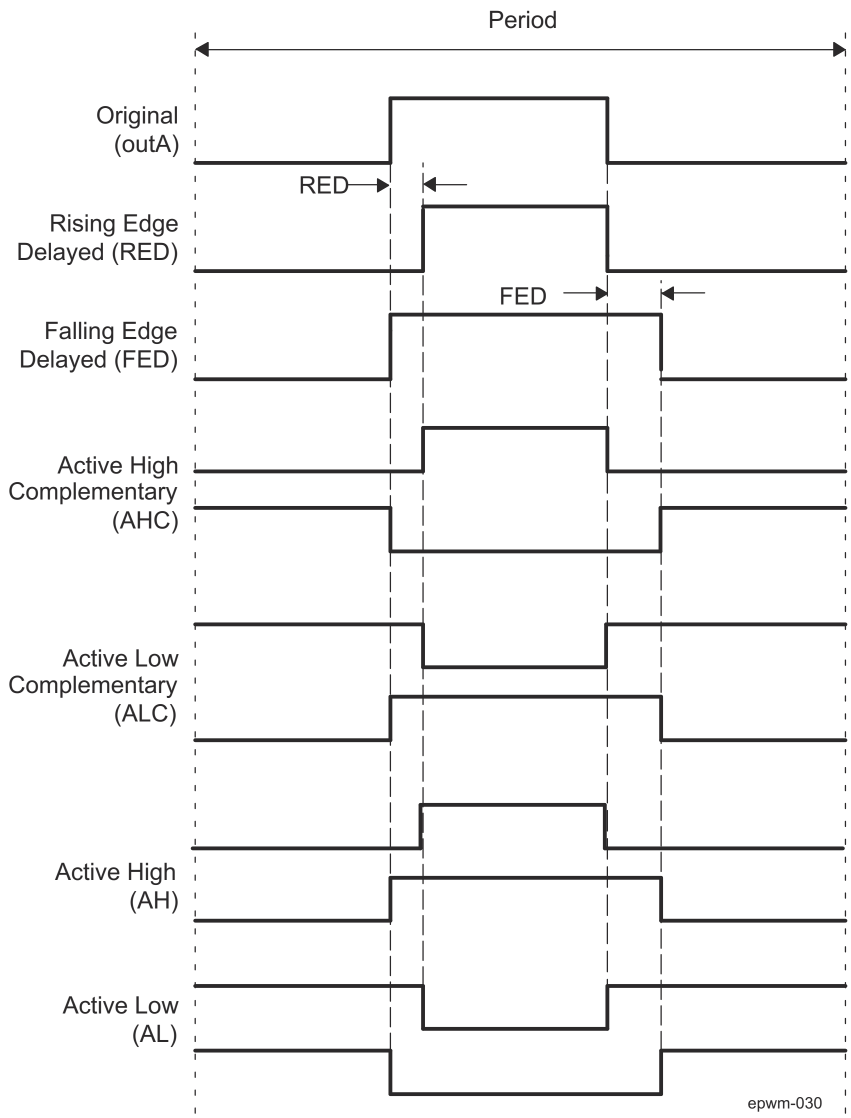


Figure 29-34. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)

The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the EPWM_DBRED and EPWM_DBFED registers. These are 10-bit registers and their value represents the number of time-base clock, TBCLK, periods a signal edge is delayed by. For example, the formula to calculate falling-edge-delay and rising-edge-delay are:

FED = EPWM_DBFED × T_{TBCLK}

RED = EPWM_DBRED × T_{TBCLK}

Where T_{TBCLK} is the period of TBCLK, the prescaled version of SYSCLKOUT.

29.2.28 PWM-Chopper (PC) Submodule

Figure 29-35 illustrates the PWM-chopper (PC) submodule within the ePWM module. The PWM-chopper submodule allows a high-frequency carrier signal to modulate the PWM waveform generated by the action-qualifier and dead-band submodules. This capability is important if you need pulse transformer-based gate drivers to control the power switching elements.

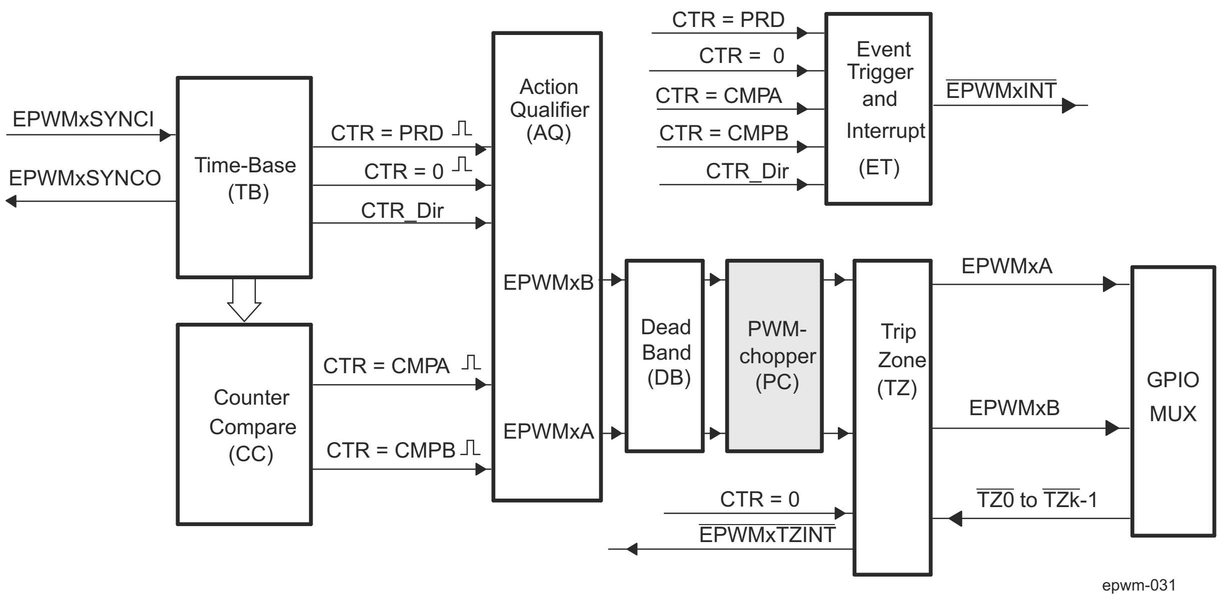


Figure 29-35. PWM-Chopper Submodule

29.2.29 Purpose of the PWM-Chopper Submodule

The key functions of the PWM-chopper submodule are:

- Programmable chopping (carrier) frequency
- Programmable pulse width of first pulse
- Programmable duty cycle of second and subsequent pulses
- Can be fully bypassed if not required

29.2.30 Controlling the PWM-Chopper Submodule

The PWM-chopper submodule operation is controlled via the register in Table 29-40.

Table 29-40. PWM-Chopper Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_PCCTL	PWM-chopper Control Register	3Ch	No

29.2.31 Operational Highlights for the PWM-Chopper Submodule

Figure 29-36 shows the operational details of the PWM-chopper submodule. The carrier clock is derived from SYSCLKOUT. Its frequency and duty cycle are controlled via the CHPFREQ and CHPDUTY bits in the EPWM_PCCTL register. The one-shot block is a feature that provides a high energy first pulse to ensure hard and fast power switch turn on, while the subsequent pulses sustain pulses, ensuring the power switch remains on. The one-shot width is programmed via the OSHTWTH bits. The PWM-chopper submodule can be fully disabled (bypassed) via the CHPEN bit.

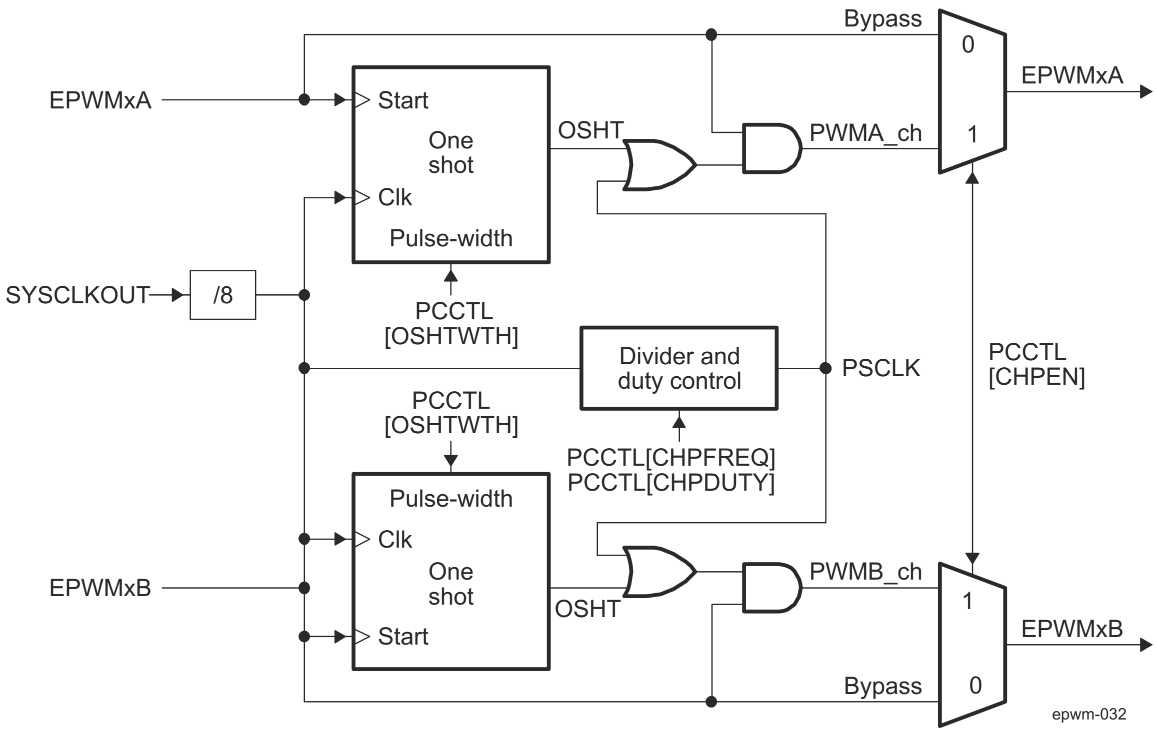


Figure 29-36. PWM-Chopper Submodule Signals and Registers

29.2.32 PWM Chopper Waveforms

Figure 29-37 shows simplified waveforms of the chopping action only; one-shot and duty-cycle control are not shown. Details of the one-shot and duty-cycle control are discussed in the following sections.

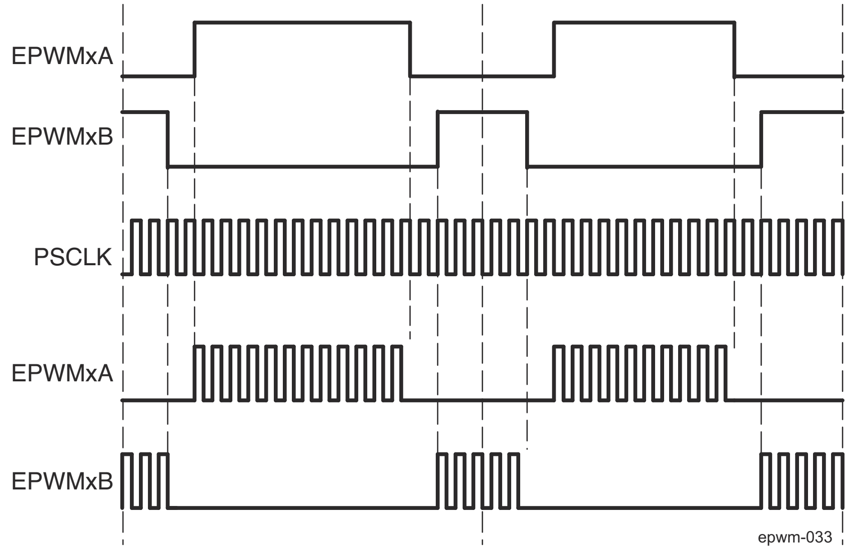


Figure 29-37. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only

29.2.33 PWM-Chopper One-Shot Pulse

The width of the first pulse can be programmed to any of 16 possible pulse width values. The width or period of the first pulse is given by:

$T_{1stpulse} = T_{SYSCLKOUT} \times 8 \times OSHTWTH$

Where $T_{SYSCLKOUT}$ is the period of the system clock (SYSCLKOUT) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 29-38 shows the first and subsequent sustaining pulses.

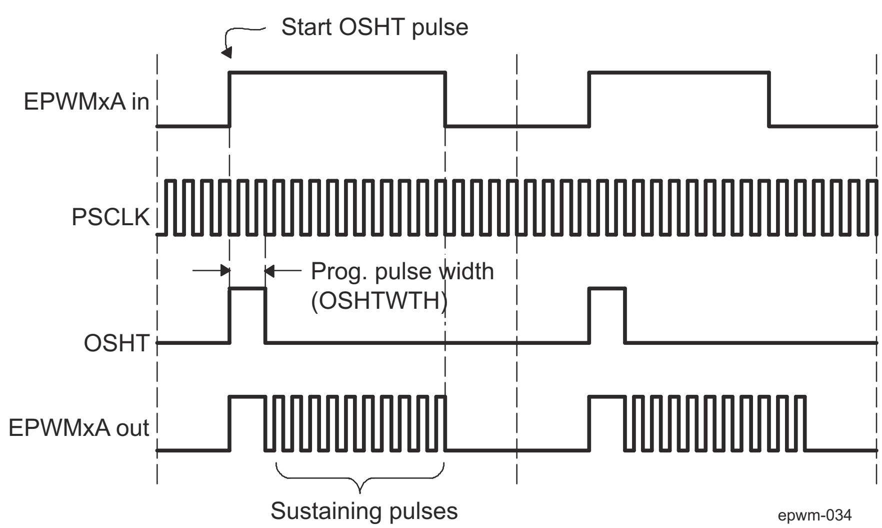


Figure 29-38. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses

29.2.34 PWM-Chopper Duty Cycle Control

Pulse transformer-based gate drive designs need to comprehend the magnetic properties or characteristics of the transformer and associated circuitry. Saturation is one such consideration. To assist the gate drive designer, the duty cycles of the second and subsequent pulses have been made programmable. These sustaining pulses ensure the correct drive strength and polarity is maintained on the power switch gate during the on period, and hence a programmable duty cycle allows a design to be tuned or optimized via software control.

Figure 29-39 shows the duty cycle control that is possible by programming the CHPDUTY bits. One of seven possible duty ratios can be selected ranging from 12.5% to 87.5%.

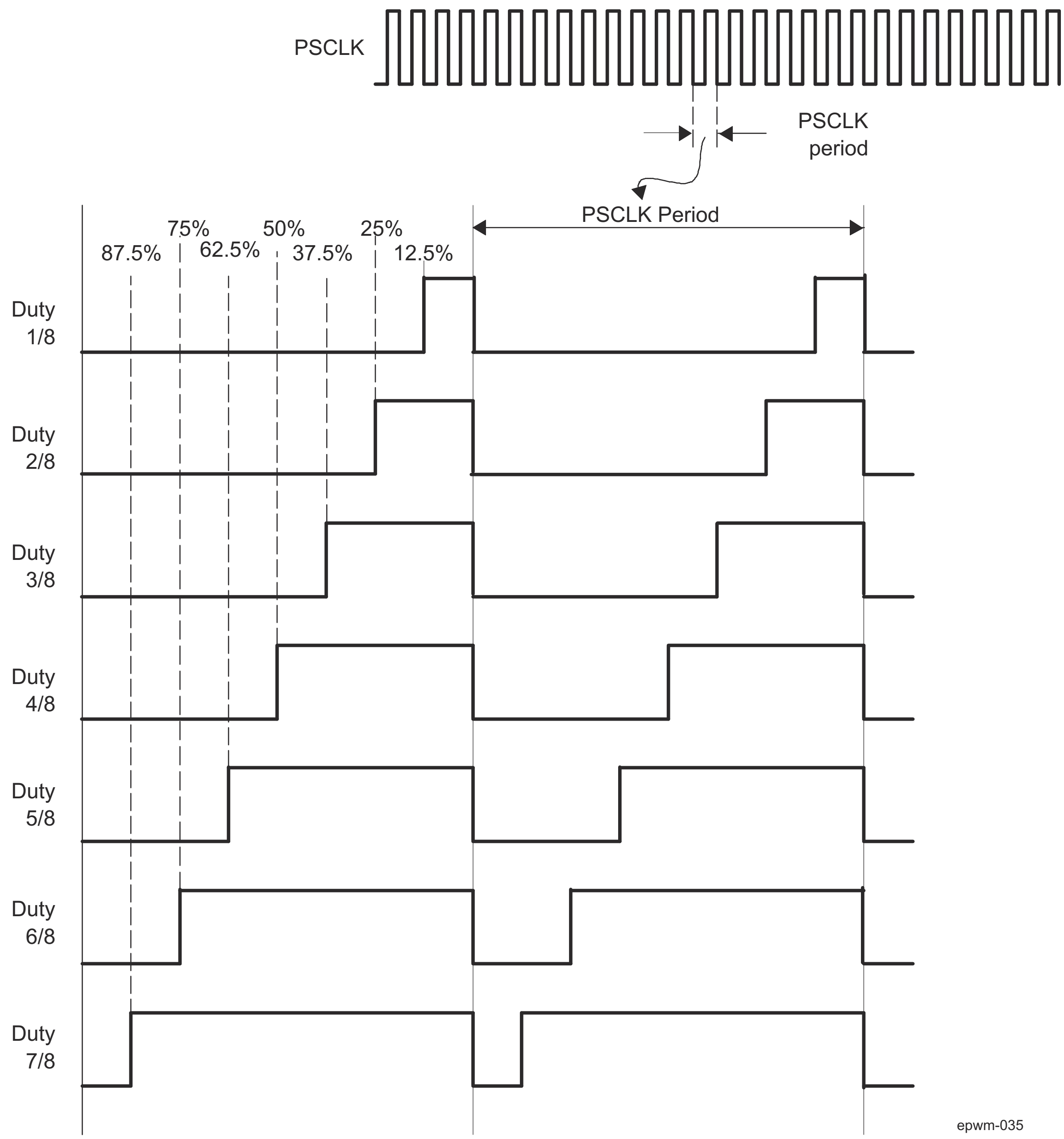


Figure 29-39. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses

29.2.35 ePWM Trip-Zone (TZ) Submodule

Figure 29-40 shows how the trip-zone (TZ) submodule fits within the ePWM module. Each ePWM module is connected to every $\overline{TZ}$ signal that are sourced from the GPIO MUX. These signals indicates external fault or trip conditions, and the ePWM outputs can be programmed to respond accordingly when faults occur. See Section 29.1.3 to determine the number of trip-zone pins available for the device.

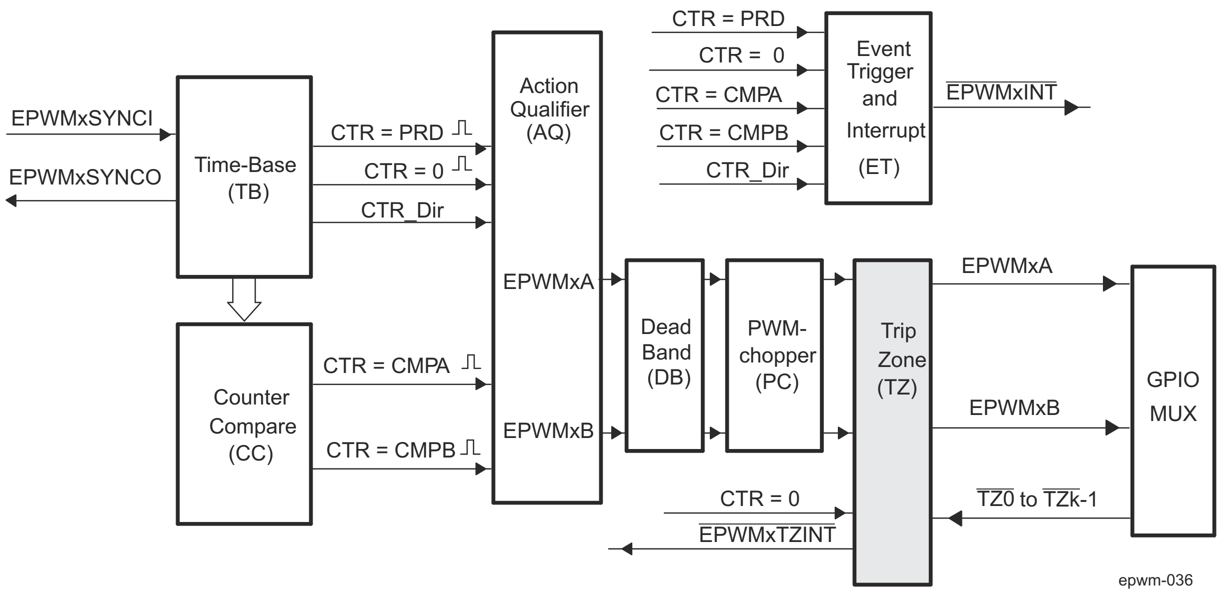


Figure 29-40. ePWM Trip-Zone Submodule

29.2.36 Purpose of the ePWM Trip-Zone Submodule

The key functions of the trip-zone submodule are:

- Trip inputs $\overline{TZ0}$ to $\overline{TZk-1}$ can be flexibly mapped to any ePWM module.
- Upon a fault condition, outputs EPWMxA and EPWMxB can be forced to one of the following:
  - High
  - Low
  - High-impedance
  - No action taken
- Support for one-shot trip (OSHT) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Each trip-zone input pin can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone pin.
- Software-forced tripping is also supported.
- The trip-zone submodule can be fully bypassed if it is not required.

Note

For each ePWMx, from the tripzone inputs $\overline{\text{EPWM\_TRIP\_TZ[5:0]}}$, ONLY the $\overline{\text{EPWM\_TRIP\_TZ[0]}}$ input of ePWM tripzone is chip accessible and usable, i.e. k=1.

29.2.37 Controlling and Monitoring the ePWM Trip-Zone Submodule

The trip-zone submodule operation is controlled and monitored through the following registers:

Table 29-41. ePWM Trip-Zone Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_TZSEL	Trip-Zone Select Register	24h	No
EPWM_TZCTL	Trip-Zone Control Register	28h	No
EPWM_TZEINT	Trip-Zone Enable Interrupt Register	2Ah	No
EPWM_TZFLG	Trip-Zone Flag Register	2Ch	No
EPWM_TZCLR	Trip-Zone Clear Register	2Eh	No
EPWM_TZFRC	Trip-Zone Force Register	30h	No

29.2.38 Operational Highlights for the ePWM Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals at pin $\overline{\text{TZ0}}$ to $\overline{\text{TZk-1}}$ is an active-low input signal. When the pin goes low, it indicates that a trip event has occurred. Each ePWM module can be individually configured to ignore or use each of the trip-zone pins. Which trip-zone pins are used by a particular ePWM module is determined by the EPWM_TZSEL register for that specific ePWM module. The trip-zone signal may or may not be synchronized to the system clock (SYSCLKOUT). A minimum of 1 SYSCLKOUT low pulse on the $\overline{\text{TZ}}$ $\overline{\text{n}}$ inputs is sufficient to trigger a fault condition in the ePWM module. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on the $\overline{\text{TZk}}$ inputs.

Note

Only the TZ[0] input is accessible at chip level (i.e. k=1).

The $\overline{\text{TZk}}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for a ePWM module. The configuration is determined by the EPWM_TZSEL[7:0] CBCk and EPWM_TZSEL[15:8] OSHTk bits (where k corresponds to the trip pin) respectively.

- **Cycle-by-Cycle (CBC):** When a cycle-by-cycle trip event occurs, the action specified in the EPWM_TZCTL register is carried out immediately on the EPWMxA and/or EPWMxB output. Table 29-42 lists the possible actions. In addition, the cycle-by-cycle trip event flag (EPWM_TZFLG[1] CBC) is set and a EPWMxTZINT interrupt is generated if it is enabled in the EPWM_TZEINT register.

  The specified condition on the pins is automatically cleared when the ePWM time-base counter reaches zero (EPWM_TBCNT bitfield TBCNT = 0000h) if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The EPWM_TZFLG[1] CBC flag bit will remain set until it is manually cleared by writing to the EPWM_TZCLR[1] CBC bit. If the cycle-by-cycle trip event is still present when the EPWM_TZFLG[1] CBC bit is cleared, then it will again be immediately set.
- **One-Shot (OSHT):** When a one-shot trip event occurs, the action specified in the EPWM_TZCTL register is carried out immediately on the EPWMxA and/or EPWMxB output. Table 29-42 lists the possible actions. In addition, the one-shot trip event flag (EPWM_TZFLG[2] OST) is set and a EPWMxTZINT interrupt is generated if it is enabled in the EPWM_TZEINT register. The one-shot trip condition must be cleared manually by writing to the EPWM_TZCLR[2] OST bit.

The action taken when a trip event occurs can be configured individually for each of the ePWM output pins by way of the EPWM_TZCTL[1:0] TZA and EPWM_TZCTL[3:2] TZB register bits. One of four possible actions, shown in Table 29-42, can be taken on a trip event.

Table 29-42. Possible Actions On an ePWM Trip Event

EPWM_TZCTL[1:0] TZA and/or EPWM_TZCTL[3:2] TZB	EPWMxA and/or EPWMxB	Comment
0	High-Impedance	Tripped
1h	Force to High State	Tripped
2h	Force to Low State	Tripped
3h	No Change	Do Nothing. No change is made to the output.

29.2.39 ePWM Trip-Zone Configurations

Scenario A:

A one-shot trip event on $\overline{TZ0}$ pulls both EPWM1A, EPWM1B low and also forces EPWM2A and EPWM2B high.

- Configure the ePWM1 registers as follows:
  - EPWM_TZSEL[8] OSHT1 = 1: enables $\overline{TZ}$ as a one-shot event source for ePWM1
  - EPWM_TZCTL[1:0] TZA = 2: EPWM1A will be forced low on a trip event.
  - EPWM_TZCTL[3:2] TZB = 2: EPWM1B will be forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - EPWM_TZSEL[8] OSHT1 = 1: enables $\overline{TZ}$ as a one-shot event source for ePWM2
  - EPWM_TZCTL[1:0] TZA = 1: EPWM2A will be forced high on a trip event.
  - EPWM_TZCTL[3:2] TZB = 1: EPWM2B will be forced high on a trip event.

29.2.40 Generating ePWM Trip Event Interrupts

Figure 29-41 and Figure 29-42 illustrate the trip-zone submodule control and interrupt logic, respectively.

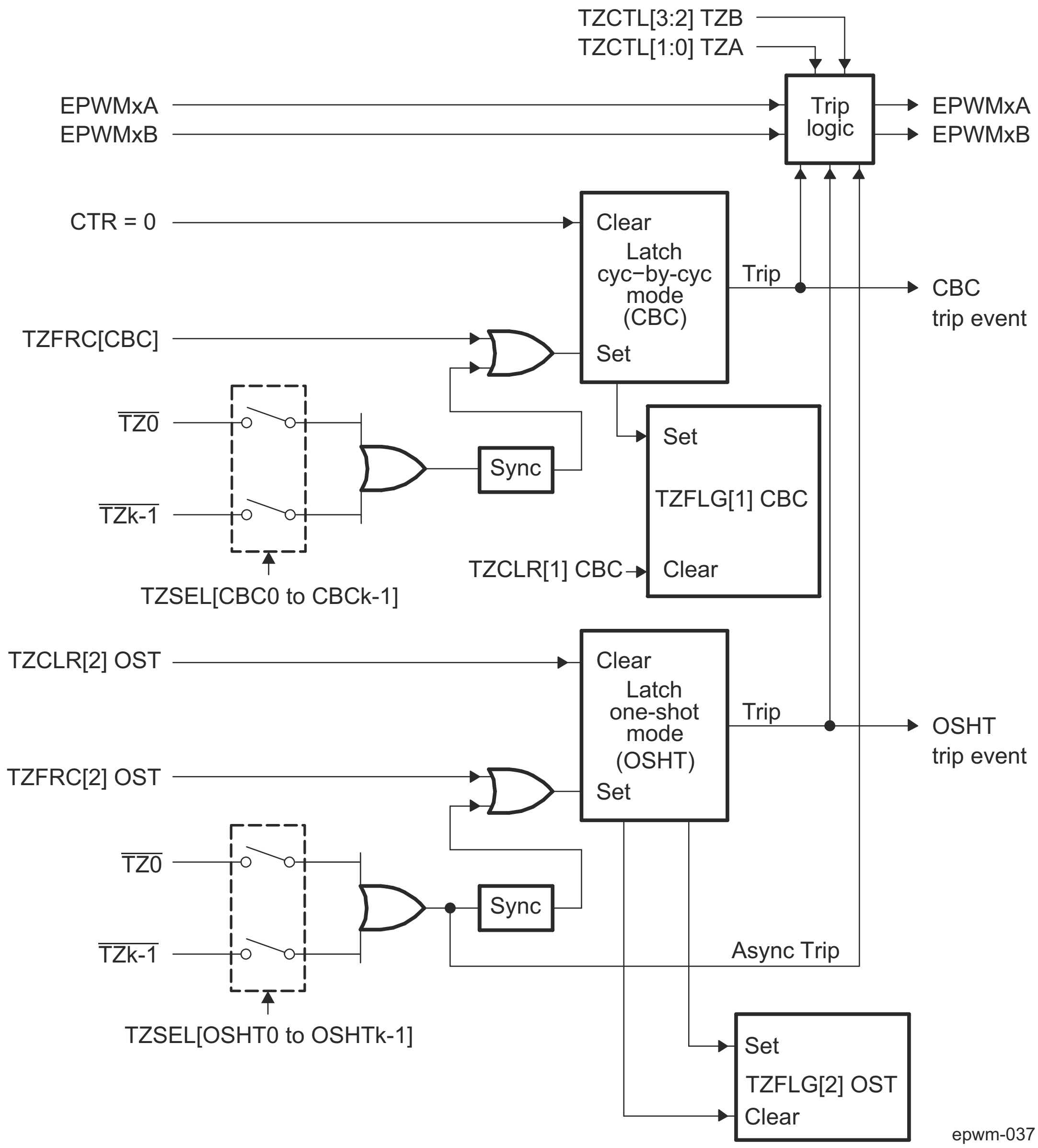


Figure 29-41. ePWM Trip-Zone Submodule Mode Control Logic

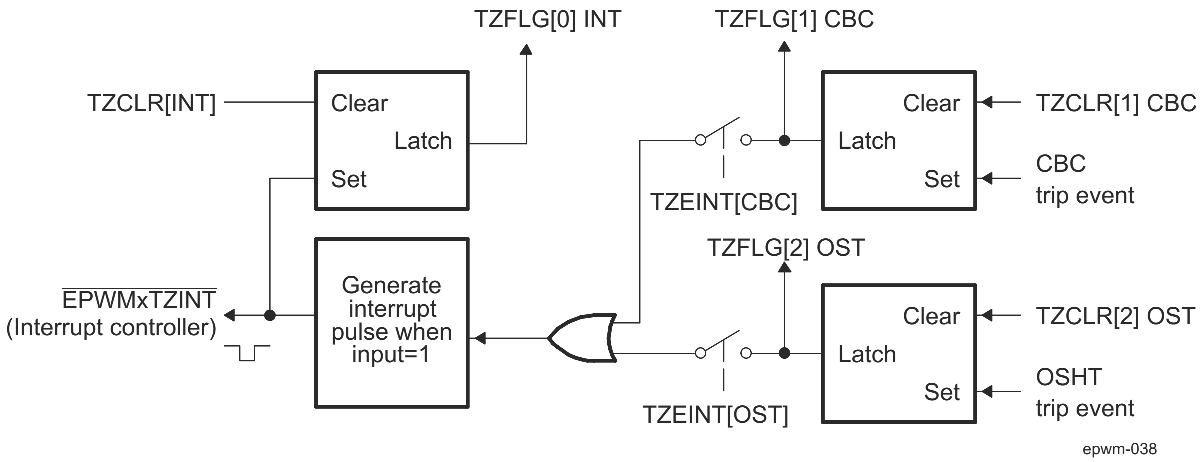


Figure 29-42. ePWM Trip-Zone Submodule Interrupt Logic

29.2.41 ePWM Event-Trigger (ET) Submodule

Figure 29-43 shows the event-trigger (ET) submodule in the ePWM system. The event-trigger submodule manages the events generated by the time-base submodule and the counter-compare submodule to generate an aggregated interrupt request.

Note

The ePWMx ET interrupt request output is further routed via the device IRQ_CROSSBAR, to different device host interrupt controllers, located outside PWMSSn. For more details on interrupt event routing outside the ePWM, refer to the Section 29.1.3.

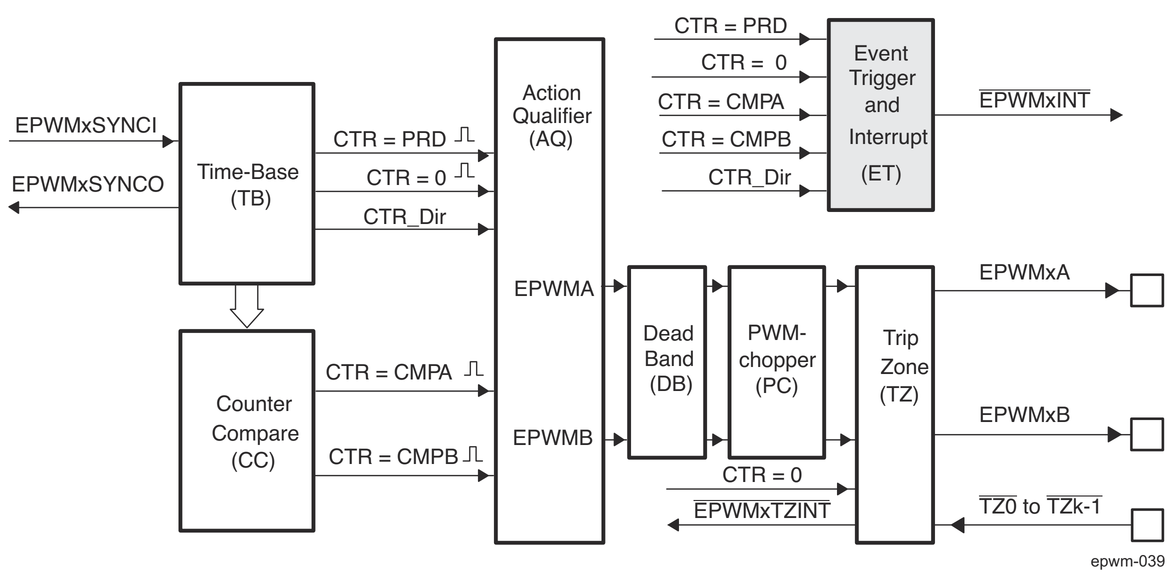


Figure 29-43. ePWM Event-Trigger Submodule

29.2.42 Purpose of the ePWM Event-Trigger Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base and counter-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests at:

  - Every event
  - Every second event
  - Every third event
- Provides full visibility of event generation via event counters and flags

29.2.43 Controlling and Monitoring the ePWM Event-Trigger Submodule

The key registers used to configure the event-trigger submodule are shown in Table 29-43:

Table 29-43. Event-Trigger Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_ETSEL	Event-Trigger Selection Register	32h	No
EPWM_ETPS	Event-Trigger Prescale Register	34h	No
EPWM_ETFLG	Event-Trigger Flag Register	36h	No
EPWM_ETCLR	Event-Trigger Clear Register	38h	No
EPWM_ETFRC	Event-Trigger Force Register	3Ah	No

29.2.44 Operational Overview of the ePWM Event-Trigger Submodule

The following sections describe the event-trigger submodule's operational highlights.

Each ePWM module has one interrupt request line as shown in Figure 29-44. Mapping interrupt lines to device host interrupt controllers is device specific and is covered in the Section 29.1.3.

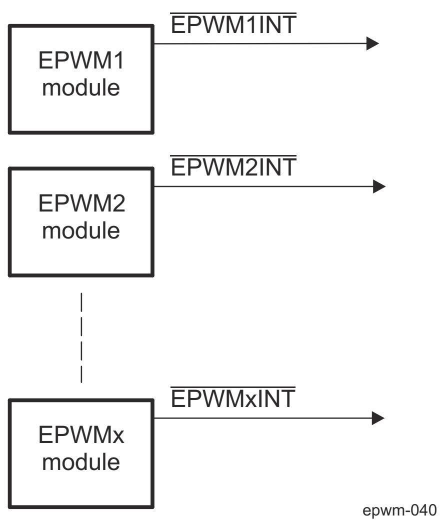


Figure 29-44. ePWM Event-Trigger Submodule Inter-Connectivity to Interrupt Controller

The event-trigger submodule monitors various event conditions (the left side inputs to event-trigger submodule shown in Figure 29-45) and can be configured to prescale these events before issuing an Interrupt request. The event-trigger prescaling logic can issue Interrupt requests at:

- Every event
- Every second event
- Every third event

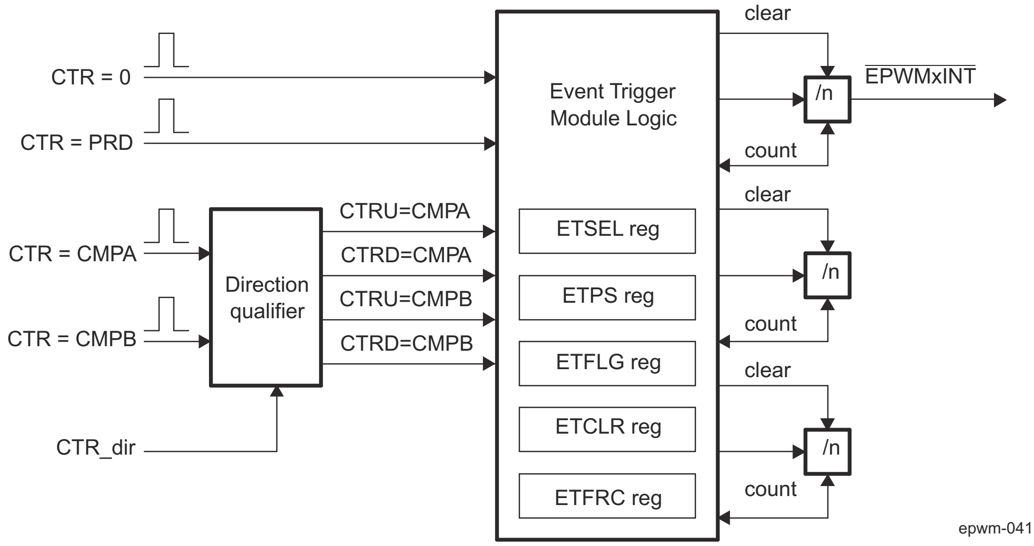


Figure 29-45. ePWM Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs

- ETSEL—This selects which of the possible events will trigger an interrupt.
- ETPS—This programs the event prescaling options previously mentioned.
- ETFLG—These are flag bits indicating status of the selected and prescaled events.
- ETCLR—These bits allow you to clear the flag bits in the EPWM_ETFLG register via software.
- ETFRC—These bits allow software forcing of an event. Useful for debugging or software intervention.

A more detailed look at how the various register bits interact with the Interrupt is shown in Figure 29-46.

Figure 29-46 shows the event-trigger's interrupt generation logic. The interrupt-period (EPWM_ETPS[1:0] INTPRD) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt
- Generate an interrupt on every event
- Generate an interrupt on every second event
- Generate an interrupt on every third event

An interrupt cannot be generated on every fourth or more events.

Which event can cause an interrupt is configured by the interrupt selection (EPWM_ETSEL[2:0] INTSEL) bits. The event can be one of the following:

- Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT = 0000h).
- Time-base counter equal to period (EPWM_TBCNT bitfield TBCNT = TBPRD value in EPWM_TBPRD active register).
- Time-base counter equal to the compare A register (EPWM_CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (EPWM_CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (EPWM_CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (EPWM_CMPB) when the timer is decrementing.

The number of events that have occurred can be read from the interrupt event counter (EPWM_ETPS[3:2] INTCNT) register bits. That is, when the specified event occurs the EPWM_ETPS[3:2] INTCNT bits are incremented until they reach the value specified by EPWM_ETPS[1:0] INTPRD. When EPWM_ETPS[3:2] INTCNT = EPWM_ETPS[1:0] INTPRD the counter stops counting and its output is set. The counter is only cleared when an interrupt is sent to the interrupt controller.

When EPWM_ETPS[3:2] INTCNT reaches EPWM_ETPS[1:0] INTPRD, one of the following behaviors will occur:

- If interrupts are enabled, EPWM_ETSEL[3] INTEN = 1 and the interrupt flag is clear, EPWM_ETFLG[0] INT = 0, then an interrupt pulse is generated and the interrupt flag is set, EPWM_ETFLG[0] INT = 1, and the event counter is cleared EPWM_ETPS[3:2] INTCNT = 0. The counter will begin counting events again.
- If interrupts are disabled, EPWM_ETSEL[3] INTEN = 0, or the interrupt flag is set, EPWM_ETFLG[0] INT = 1, the counter stops counting events when it reaches the period value EPWM_ETPS[3:2] INTCNT = EPWM_ETPS[1:0] INTPRD.
- If interrupts are enabled, but the interrupt flag is already set, then the counter will hold its output high until the ENTFLG[0] INT flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing to the INTPRD bits will automatically clear the counter INTCNT = 0 and the counter output will be reset (so no interrupts are generated). Writing a 1 to the EPWM_ETFRC[0] INT bit will increment the event counter INTCNT. The counter will behave as described above when INTCNT = INTPRD. When INTPRD = 0, the counter is disabled and hence no events will be detected and the EPWM_ETFRC[0] INT bit is also ignored.

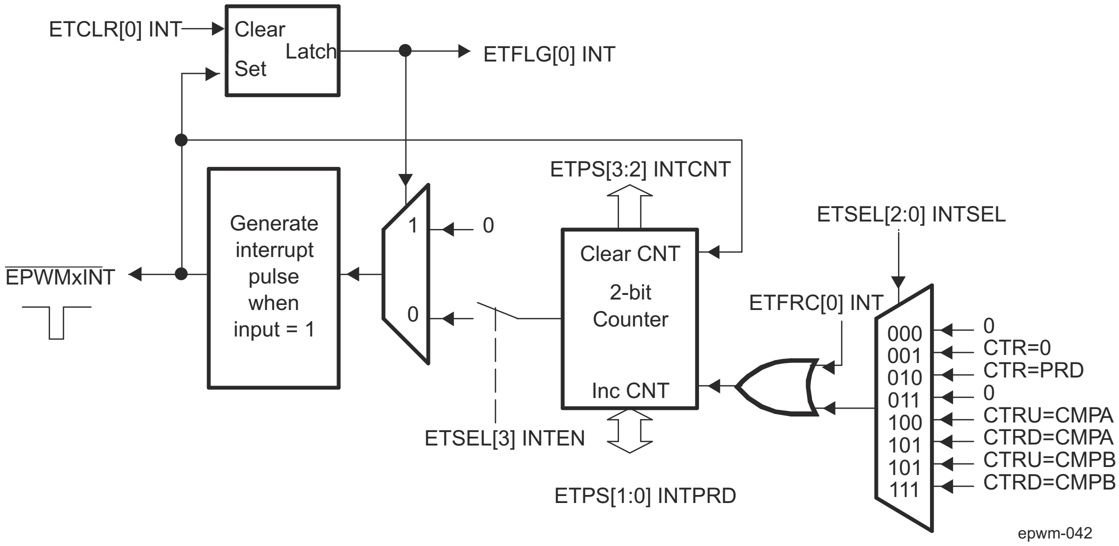


Figure 29-46. ePWM Event-Trigger Interrupt Generator

29.2.45 High-Resolution PWM (HRPWM) Submodule

Figure 29-47 shows the high-resolution PWM (HRPWM) submodule in the ePWM system. Some devices include the high-resolution PWM submodule, see Section 29.1.3 to determine which ePWM instances include this feature.

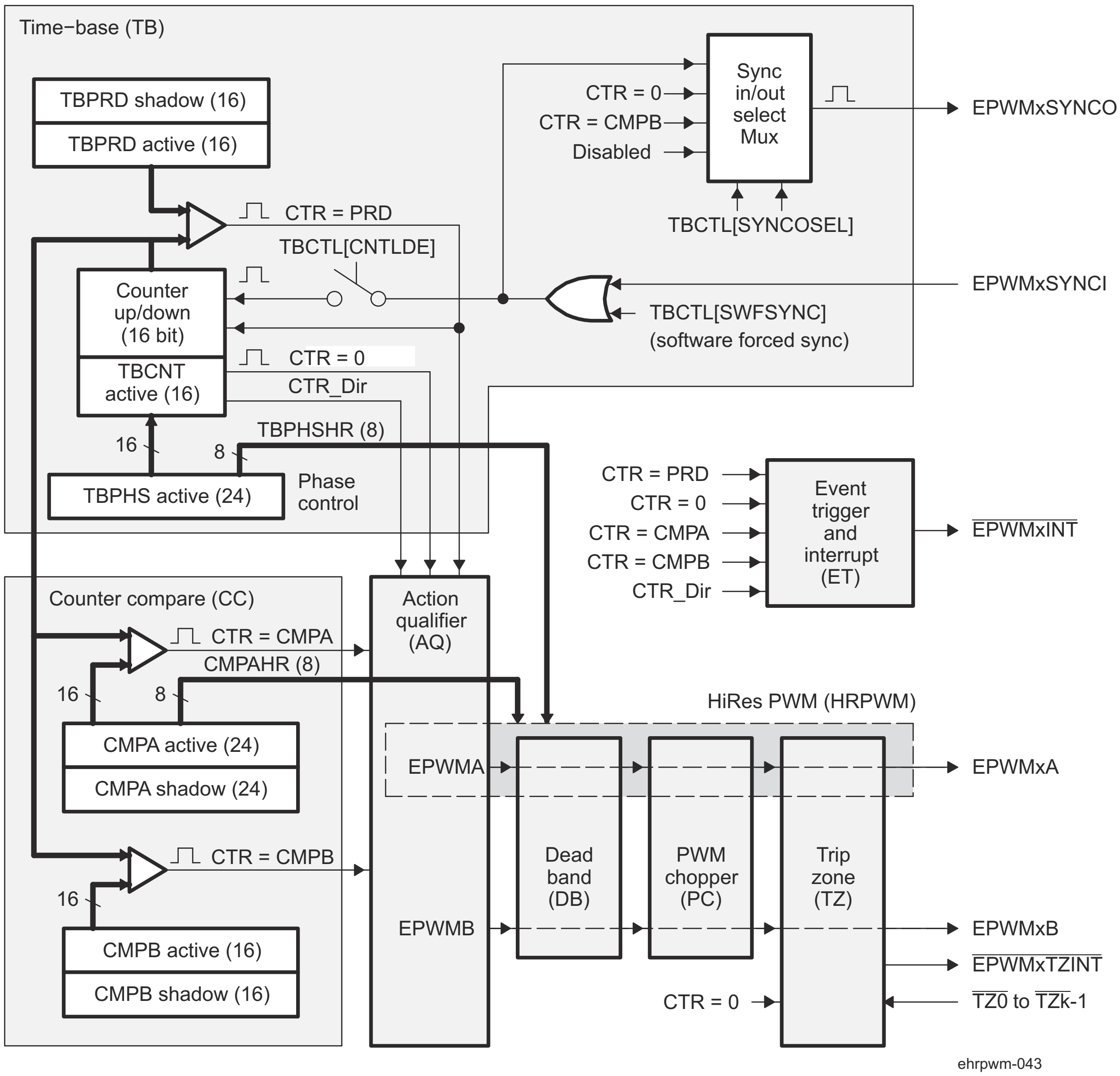


Figure 29-47. HRPWM System Interface

29.2.46 Purpose of the High-Resolution PWM Submodule

The enhanced high-resolution pulse-width modulator (eHRPWM) extends the time resolution capabilities of the conventionally derived digital pulse-width modulator (PWM). HRPWM is typically used when PWM resolution falls below ~9-10 bits. The key features of HRPWM are:

- Extended time resolution capability
- Used in both duty cycle and phase-shift control methods
- Finer time granularity control or edge positioning using extensions to the Compare A and Phase registers
- Implemented using the A signal path of PWM, that is, on the EPWMxA output. **EPWMxB output has conventional PWM capabilities**

The ePWM peripheral is used to perform a function that is mathematically equivalent to a digital-to-analog converter (DAC). As shown in Figure 29-48, the effective resolution for conventionally generated PWM is a function of PWM frequency (or period) and system clock frequency.

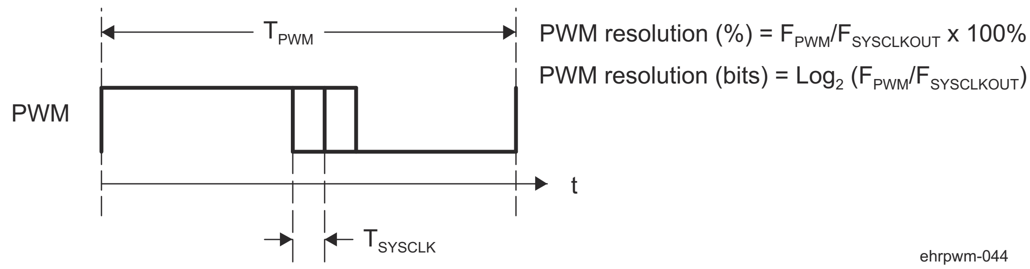


Figure 29-48. Resolution Calculations for Conventionally Generated PWM

If the required PWM operating frequency does not offer sufficient resolution in PWM mode, you may want to consider HRPWM. As an example of improved performance offered by HRPWM, Table 29-44 shows resolution in bits for various PWM frequencies. Table 29-44 values assume a MEP step size of 180 ps. See your device-specific data manual for typical and maximum performance specifications for the MEP.

Table 29-44. Resolution for PWM and HRPWM

	Regular Resolution (PWM)		High Resolution (HRPWM)	
PWM Frequency (kHz)	Bits	%	Bits	%
20	12.3	0.0	18.1	0.000
50	11.0	0.0	16.8	0.001
100	10.0	0.1	15.8	0.002
150	9.4	0.2	15.2	0.003
200	9.0	0.2	14.8	0.004
250	8.6	0.3	14.4	0.005
500	7.6	0.5	13.8	0.007
1000	6.6	1.0	12.4	0.018
1500	6.1	1.5	11.9	0.027
2000	5.6	2.0	11.4	0.036

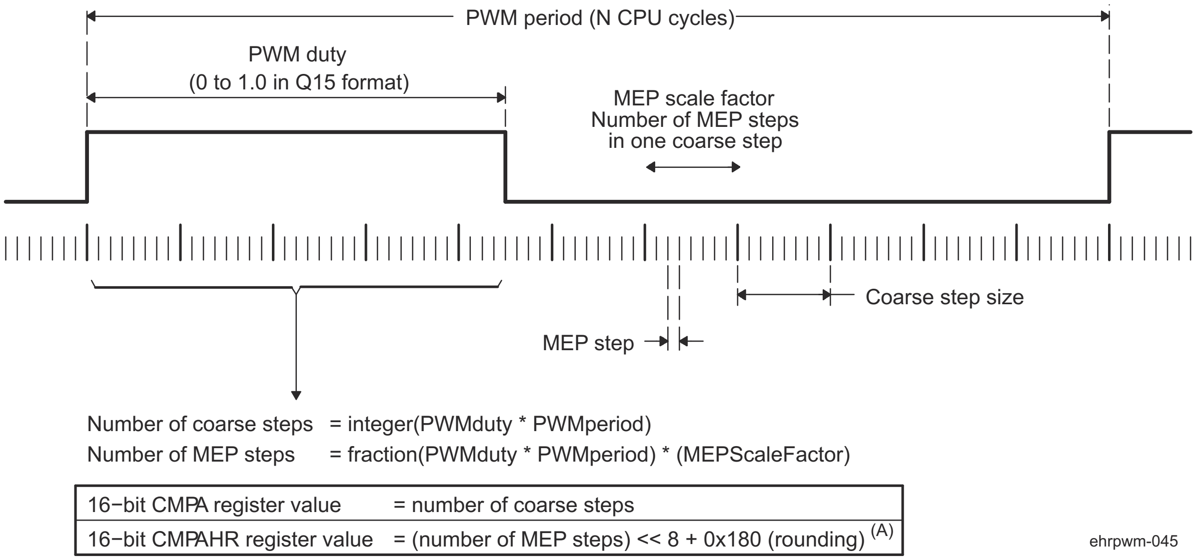
Although each application may differ, typical low-frequency PWM operation (below 250 kHz) may not require HRPWM. HRPWM capability is most useful for high-frequency PWM requirements of power conversion topologies such as:

- Single-phase buck, boost, and flyback
- Multi-phase buck, boost, and flyback
- Phase-shifted full bridge
- Direct modulation of D-Class power amplifiers

29.2.47 Architecture of the High-Resolution PWM Submodule

The HRPWM is based on micro edge positioner (MEP) technology. MEP logic is capable of positioning an edge very finely by sub-dividing one coarse system clock of a conventional PWM generator. The time step accuracy is on the order of 150 ps. The HRPWM also has a self-check software diagnostics mode to check if the MEP logic is running optimally, under all operating conditions.

Figure 29-49 shows the relationship between one coarse system clock and edge position in terms of MEP steps, which are controlled via an 8-bit field in the Compare A extension register (HRPWM_CMPAHR).



A. For MEP range and rounding adjustment.

Figure 29-49. Operating Logic Using MEP

To generate an HRPWM waveform, configure the TBM, CCM, and AQM registers as you would to generate a conventional PWM of a given frequency and polarity. The HRPWM works together with the TBM, CCM, and AQM registers to extend edge resolution, and should be configured accordingly. Although many programming combinations are possible, only a few are needed and practical.

29.2.48 Controlling and Monitoring the High-Resolution PWM Submodule

The MEP of the HRPWM is controlled by two extension registers, each 8-bits wide. These two HRPWM registers are concatenated with the 16-bit EPWM_TBPHS and EPWM_CMPA registers used to control PWM operation.

- HRPWM_TBPHSHR - Time-Base Phase High-Resolution Register
- HRPWM_CMPAHR - Counter-Compare A High-Resolution Register

Table 29-45 lists the registers used to control and monitor the high-resolution PWM submodule.

Table 29-45. HRPWM Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
HRPWM_TBPHSHR	Extension Register for HRPWM Phase	4h	No
HRPWM_CMPAHR	Extension Register for HRPWM Duty	10h	Yes
HRPWM_HRCTL	HRPWM Configuration Register	1040h	No

29.2.49 Configuring the High-Resolution PWM Submodule

Once the ePWM has been configured to provide conventional PWM of a given frequency and polarity, the HRPWM is configured by programming the HRPWM_HRCTL register located at offset address 1040h. This register provides configuration options for the following key operating modes:

- **Edge Mode:** The MEP can be programmed to provide precise position control on the rising edge (RE), falling edge (FE), or both edges (BE) at the same time. FE and RE are used for power topologies requiring duty cycle control, while BE is used for topologies requiring phase shifting, for example, phase shifted full bridge.
- **Control Mode:** The MEP is programmed to be controlled either from the HRPWM_CMPAHR register (duty cycle control) or the HRPWM_TBPHSHR register (phase control). RE or FE control mode should be used with HRPWM_CMPAHR register. BE control mode should be used with HRPWM_TBPHSHR register.
- **Shadow Mode:** This mode provides the same shadowing (double buffering) option as in regular PWM mode. This option is valid only when operating from the HRPWM_CMPAHR register and should be chosen to be the same as the regular load option for the CMPA register. If HRPWM_TBPHSHR is used, then this option has no effect.

29.2.50 Operational Highlights for the High-Resolution PWM Submodule

The MEP logic is capable of placing an edge in one of 255 (8 bits) discrete time steps, each of which has a time resolution on the order of 150 ps. The MEP works with the TBM and CCM registers to be certain that time steps are optimally applied and that edge placement accuracy is maintained over a wide range of PWM frequencies, system clock frequencies and other operating conditions. Table 29-46 shows the typical range of operating frequencies supported by the HRPWM.

Table 29-46. Relationship Between MEP Steps, PWM Frequency and Resolution

System (MHz)	MEP Steps Per SYSCLKOUT(1) (2) (3)	PWM Minimum (Hz)(4)	PWM Maximum (MHz)	Resolution at Maximum (Bits)(5)
50.0	111	763	2.50	11.1
60.0	93	916	3.00	10.9
70.0	79	1068	3.50	10.6
80.0	69	1221	4.00	10.4
90.0	62	1373	4.50	10.3
100.0	56	1526	5.00	10.1

(1) System frequency = SYSCLKOUT, that is, CPU clock. TBCLK = SYSCLKOUT
(2) Table data based on a MEP time resolution of 180 ps (this is an example value)
(3) MEP steps applied = $T_{SYSCLKOUT}$/180 ps in this example.
(4) PWM minimum frequency is based on a maximum period value, TBPRD = 65 535. PWM mode is asymmetrical up-count.
(5) Resolution in bits is given for the maximum PWM frequency stated.

29.2.51 HRPWM Edge Positioning

In a typical power control loop (switch modes, digital motor control (DMC), uninterruptible power supply (UPS)), a digital controller (PID, 2pole/2zero, lag/lead, etc.) issues a duty command, usually expressed in a per unit or percentage terms.

In the following example, assume that for a particular operating point, the demanded duty cycle is 0.405 or 40.5% on-time and the required converter PWM frequency is 1.25 MHz. In conventional PWM generation with a system clock of 100 MHz, the duty cycle choices are in the vicinity of 40.5%. In Figure 29-50, a compare value of 32 counts (duty = 40%) is the closest to 40.5% that you can attain. This is equivalent to an edge position of 320 ns instead of the desired 324 ns. This data is shown in Table 29-47.

By utilizing the MEP, you can achieve an edge position much closer to the desired point of 324 ns. Table 29-47 shows that in addition to the CMPA value, 22 steps of the MEP (HRPWM_CMPAHR register) will position the edge at 323.96 ns, resulting in almost zero error. In this example, it is assumed that the MEP has a step resolution of 180 ns.

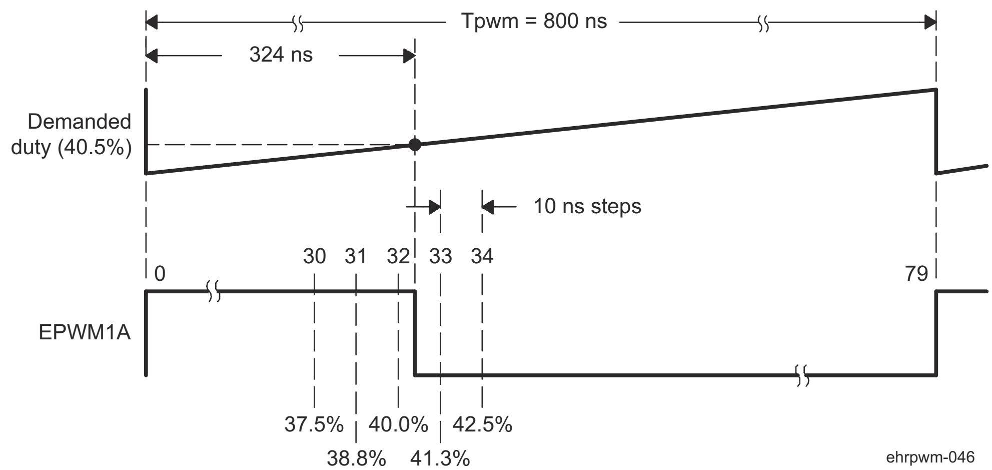


Figure 29-50. Required PWM Waveform for a Requested Duty = 40.5%

Table 29-47. CMPA vs Duty (left), and [CMPA:CMPAHR] vs Duty (right)

CMPA (count)(1) (2) (3)	DUTY (%)	High Time (ns)	CMPA (count)	CMPAHR (count)	Duty (%)	High Time (ns)
28	35.0	280	32	18	40.405	323.24
29	36.3	290	32	19	40.428	323.42
30	37.5	300	32	20	40.450	323.60
31	38.8	310	32	21	40.473	323.78
32	40.0	320	32	22	40.495	323.96
33	41.3	330	32	23	40.518	324.14
34	42.5	340	32	24	40.540	324.32
			32	25	40.563	324.50
Required			32	26	40.585	324.68
32.40	40.5	324	32	27	40.608	324.86

(1) System clock, SYSCLKOUT and TBCLK = 100 MHz, 10 ns
(2) For a PWM Period register value of 80 counts, PWM Period = 80 × 10 ns = 800 ns, PWM frequency = 1/800 ns = 1.25 MHz
(3) Assumed MEP step size for the above example = 180 ps

29.2.52 HRPWM Scaling Considerations

The mechanics of how to position an edge precisely in time has been demonstrated using the resources of the standard (EPWM_CMPA) and MEP (HRPWM_CMPAHR) registers. In a practical application, however, it is necessary to seamlessly provide the CPU a mapping function from a per-unit (fractional) duty cycle to a final integer (non-fractional) representation that is written to the [CMPA:CMPAHR] register combination.

To do this, first examine the scaling or mapping steps involved. It is common in control software to express duty cycle in a per-unit or percentage basis. This has the advantage of performing all needed math calculations without concern for the final absolute duty cycle, expressed in clock counts or high time in ns. Furthermore, it makes the code more transportable across multiple converter types running different PWM frequencies.

To implement the mapping scheme, a two-step scaling procedure is required.

Assumptions for this example:

System clock, SYSCLKOUT	=	10 ns (100 MHz)
PWM frequency	=	1.25 MHz (1/800 ns)
Required PWM duty cycle, **PWMDuty**	=	0.405 (40.5%)
PWM period in terms of coarse steps, **PWMperiod** (800 ns/10 ns)	=	80
Number of MEP steps per coarse step at 180 ps (10 ns/180 ps), **MEP_SF**	=	55
Value to keep CMPAHR within the range of 1-255 and fractional rounding constant (default value)	=	180h

Step 1: Percentage Integer Duty value conversion for EPWM_CMPA register

EPWM_CMPA register value	=	int(**PWMDuty** × **PWMperiod**); int means integer part
	=	int(0.405 × 80)
	=	int(32.4)
EPWM_CMPA register value	=	32 (20h)

Step 2: Fractional value conversion for HRPWM_CMPAHR register

HRPWM_CMPAHR register value	=	(frac(**PWMDuty** × **PWMperiod**) × **MEP_SF**) << 8) + 180h; frac means fractional part
	=	(frac(32.4) × 55 <<8) + 180h; Shift is to move the value as CMPAHR high byte
	=	((0.4 × 55) <<8) + 180h
	=	(22 <<8) + 180h
	=	22 × 256 + 180h ; Shifting left by 8 is the same multiplying by 256.
	=	5632 + 180h
	=	1600h + 180h
HRPWM_CMPAHR value	=	1780h; HRPWM_CMPAHR value = 1700h, lower 8 bits will be ignored by hardware.

29.2.53 HRPWM Duty Cycle Range Limitation

In high resolution mode, the MEP is not active for 100% of the PWM period. It becomes operational 3 SYSCLKOUT cycles after the period starts.

Duty cycle range limitations are illustrated in Figure 29-51. This limitation imposes a lower duty cycle limit on the MEP. For example, precision edge control is not available all the way down to 0% duty cycle. Although for the first 3 or 6 cycles, the HRPWM capabilities are not available, regular PWM duty control is still fully operational down to 0% duty. In most applications this should not be an issue as the controller regulation point is usually not designed to be close to 0% duty cycle.

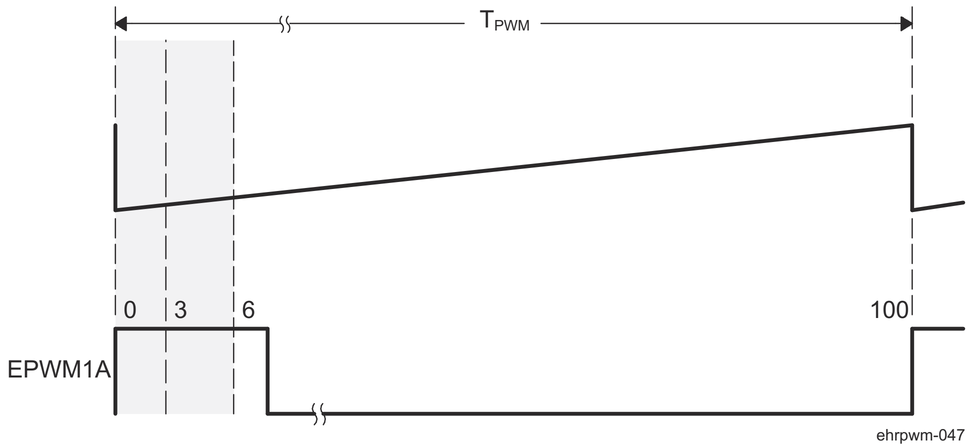


Figure 29-51. Low % Duty Cycle Range Limitation Example When PWM Frequency = 1 MHz

If the application demands HRPWM operation in the low percent duty cycle region, then the HRPWM can be configured to operate in count-down mode with the rising edge position (REP) controlled by the MEP. This is illustrated in Figure 29-52. In this case low percent duty limitation is no longer an issue.

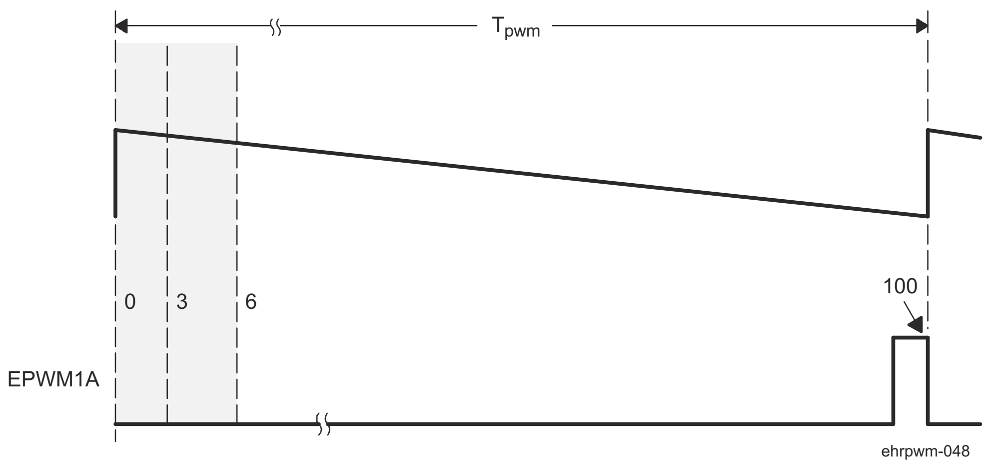


Figure 29-52. High % Duty Cycle Range Limitation Example when PWM Frequency = 1 MHz

29.2.54 eHRPWM Functional Register Groups

The Table 29-48 lists the groups of ePWM and the high-resolution PWM module registers according to their functionalities.

Table 29-48. ePWM/HRPWM Module Control and Status Registers Grouped by Submodule

Register Name	Offset	Size (×16)	Shadow	Register Description
Time-Base Submodule Registers				

Table 29-48. ePWM/HRPWM Module Control and Status Registers Grouped by Submodule (continued)

Register Name	Offset	Size (×16)	Shadow	Register Description
EPWM_TBCTL	0h	1	No	Time-Base Control Register
EPWM_TBSTS	2h	1	No	Time-Base Status Register
EPWM_TBPHS	6h	1	No	Time-Base Phase Register
EPWM_TBCNT	8h	1	No	Time-Base Counter Register
EPWM_TBPRD	Ah	1	Yes	Time-Base Period Register
Counter-Compare Submodule Registers				
EPWM_CMPCTL	Eh	1	No	Counter-Compare Control Register
EPWM_CMPA	12h	1	Yes	Counter-Compare A Register
EPWM_CMPB	14h	1	Yes	Counter-Compare B Register
Action-Qualifier Submodule Registers				
EPWM_AQCTLA	16h	1	No	Action-Qualifier Control Register for Output A (EPWMxA)
EPWM_AQCTLB	18h	1	No	Action-Qualifier Control Register for Output B (EPWMxB)
EPWM_AQSFRC	1Ah	1	No	Action-Qualifier Software Force Register
EPWM_AQCSFRC	1Ch	1	Yes	Action-Qualifier Continuous S/W Force Register Set
Dead-Band Generator Submodule Registers				
EPWM_DBCTL	1Eh	1	No	Dead-Band Generator Control Register
EPWM_DBRED	20h	1	No	Dead-Band Generator Rising Edge Delay Count Register
EPWM_DBFED	22h	1	No	Dead-Band Generator Falling Edge Delay Count Register
Trip-Zone Submodule Registers				
EPWM_TZSEL	24h	1	No	Trip-Zone Select Register
EPWM_TZCTL	28h	1	No	Trip-Zone Control Register
EPWM_TZEINT	2Ah	1	No	Trip-Zone Enable Interrupt Register
EPWM_TZFLG	2Ch	1	No	Trip-Zone Flag Register
EPWM_TZCLR	2Eh	1	No	Trip-Zone Clear Register
EPWM_TZFRC	30h	1	No	Trip-Zone Force Register
Event-Trigger Submodule Registers				
EPWM_ETSEL	32h	1	No	Event-Trigger Selection Register
EPWM_ETPS	34h	1	No	Event-Trigger Pre-Scale Register
EPWM_ETFLG	36h	1	No	Event-Trigger Flag Register
EPWM_ETCLR	38h	1	No	Event-Trigger Clear Register
EPWM_ETFRC	3Ah	1	No	Event-Trigger Force Register
PWM-Chopper Submodule Registers				
EPWM_PCCTL	3Ch	1	No	PWM-Chopper Control Register
High-Resolution PWM (HRPWM) Submodule Registers				
HRPWM_TBPHSHR	4h	1	No	Extension for HRPWM Phase Register
HRPWM_CMPAHR	10h	1	No	Extension for HRPWM Counter-Compare A Register
HRPWM_HRCTL	40h	1	No	HRPWM Control Register

29.2.55 PWMSS_EPWM Register Manual

This section provides description of the device PWMSS ePWM and High Resolution-PWM relevant functional registers.

29.2.56 PWMSS_EPWM Instance Summary

Table 29-49. PWMSS_EPWM Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS1_EPWM	0x4843 E200	136 Bytes
PWMSS2_EPWM	0x4844 0200	136 Bytes
PWMSS3_EPWM	0x4844 2200	136 Bytes

29.2.57 PWMSS_EPWM Registers

29.2.58 PWMSS_EPWM Register Summary

Table 29-50. PWMSSn_EPWM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS1_EPWM Physical Address L4_PER2 Interconnect	PWMSS2_EPWM Physical Address L4_PER2 Interconnect	PWMSS3_EPWM Physical Address L4_PER2 Interconnect
EPWM_TBCTL	RW	16	0x0000 0000	0x4843 E200	0x4844 0200	0x4844 2200
EPWM_TBSTS	RW	16	0x0000 0002	0x4843 E202	0x4844 0202	0x4844 2202
HRPWM_TBPHSHR	RW	16	0x0000 0004	0x4843 E204	0x4844 0204	0x4844 2204
EPWM_TBPHS	RW	16	0x0000 0006	0x4843 E206	0x4844 0206	0x4844 2206
EPWM_TBCNT	RW	16	0x0000 0008	0x4843 E208	0x4844 0208	0x4844 2208
EPWM_TBPRD	RW	16	0x0000 000A	0x4843 E20A	0x4844 020A	0x4844 220A
EPWM_CMPCTL	RW	16	0x0000 000E	0x4843 E20E	0x4844 020E	0x4844 220E
HRPWM_CMPAHR	RW	16	0x0000 0010	0x4843 E210	0x4844 0210	0x4844 2210
EPWM_CMPA	RW	16	0x0000 0012	0x4843 E212	0x4844 0212	0x4844 2212
EPWM_CMPB	RW	16	0x0000 0014	0x4843 E214	0x4844 0214	0x4844 2214
EPWM_AQCTLA	RW	16	0x0000 0016	0x4843 E216	0x4844 0216	0x4844 2216
EPWM_AQCTLB	RW	16	0x0000 0018	0x4843 E218	0x4844 0218	0x4844 2218
EPWM_AQSFRC	RW	16	0x0000 001A	0x4843 E21A	0x4844 021A	0x4844 221A
EPWM_AQCSFRC	RW	16	0x0000 001C	0x4843 E21C	0x4844 021C	0x4844 221C
EPWM_DBCTL	RW	16	0x0000 001E	0x4843 E21E	0x4844 021E	0x4844 221E
EPWM_DBRED	RW	16	0x0000 0020	0x4843 E220	0x4844 0220	0x4844 2220
EPWM_DBFED	RW	16	0x0000 0022	0x4843 E222	0x4844 0222	0x4844 2222
EPWM_TZSEL	RW	16	0x0000 0024	0x4843 E224	0x4844 0224	0x4844 2224
EPWM_TZCTL	RW	16	0x0000 0028	0x4843 E228	0x4844 0228	0x4844 2228
EPWM_TZEINT	RW	16	0x0000 002A	0x4843 E22A	0x4844 022A	0x4844 222A
EPWM_TZFLG	R	16	0x0000 002C	0x4843 E22C	0x4844 022C	0x4844 222C
EPWM_TZCLR	RW	16	0x0000 002E	0x4843 E22E	0x4844 022E	0x4844 222E
EPWM_TZFRC	RW	16	0x0000 0030	0x4843 E230	0x4844 0230	0x4844 2230
EPWM_ETSEL	RW	16	0x0000 0032	0x4843 E232	0x4844 0232	0x4844 2232
EPWM_ETPS	RW	16	0x0000 0034	0x4843 E234	0x4844 0234	0x4844 2234
EPWM_ETFLG	R	16	0x0000 0036	0x4843 E236	0x4844 0236	0x4844 2236
EPWM_ETCLR	RW	16	0x0000 0038	0x4843 E238	0x4844 0238	0x4844 2238
EPWM_ETFRC	RW	16	0x0000 003A	0x4843 E23A	0x4844 023A	0x4844 223A
EPWM_PCCTL	RW	16	0x0000 003C	0x4843 E23C	0x4844 023C	0x4844 223C
HRPWM_HRCTL	RW	16	0x0000 00C0	0x4843 E2C0	0x4844 02C0	0x4844 22C0

29.2.59 PWMSS_EPWM Register Description

Table 29-51. EPWM_TBCTL

Address Offset	0x0000 0000		
Physical Address	0x4843 E200 0x4844 0200 0x4844 2200	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE_SOFT		PHSDIR	CLKDIV			HSPCLKDIV			SWFSYNC	SYNCOSEL		PRDLD	PHSEN	CTRMODE	

Bits	Field Name	Description	Type	Reset
15:14	FREE_SOFT	Emulation Mode Bits. These bits select the behavior of the ePWM time-base counter during emulation events: 0x0 = Stop after the next time-base counter increment or decrement 0x1 = Stop when counter completes a whole cycle. (a) Up-count mode: stop when the time-base counter = period (EPWM_TBCNT bitfield TBCNT = TBPRD in EPWM_TBPRD active register). (b) Down-count mode: stop when the time-base counter = 0000 (EPWM_TBCNT bitfield TBCNT = 0000h). (c) Up-down-count mode: stop when the time-base counter = 0000 (EPWM_TBCNT bitfield TBCNT = 0000h). 0x2 = Free run 0x3 = Free run	RW	0x0
13	PHSDIR	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. The PHSDIR bit indicates the direction the time-base counter (EPWM_TBCNT) will count after a synchronization event occurs and a new phase value is loaded from the phase (EPWM_TBPHS) register. This is irrespective of the direction of the counter before the synchronization event.. In the up-count and down-count modes this bit is ignored. 0x0 = Count down after the synchronization event. 0x1 = Count up after the synchronization event.	RW	0x0
12:10	CLKDIV	Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = SYSCLKOUT/(HSPCLKDIV x CLKDIV) 0x0 = /1 (default on reset) 0x1 = /2 0x2 = /4 0x3 = /8 0x4 = /16 0x5 = /32 0x6 = /64 0x7 = /128	RW	0x0

Bits	Field Name	Description	Type	Reset
9:7	HSPCLKDIV	High-Speed Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = SYSCLKOUT/(HSPCLKDIV x CLKDIV). This divisor emulates the HSPCLK in the TMS320x281x system as used on the Event Manager (EV) peripheral. 0x0 = /1 0x1 = /2 (default on reset) 0x2 = /4 0x3 = /6 0x4 = /8 0x5 = /10 0x6 = /12 0x7 = /14	RW	0x0
6	SWFSYNC	Software Forced Synchronization Pulse. 0x0 = Writing a 0 has no effect and reads always return a 0. 0x1 = Writing a 1 forces a one-time synchronization pulse to be generated. This event is ORed with the EPWMxSYNCI input of the ePWM module. SWFSYNC is valid (operates) only when EPWMxSYNCI is selected by SYNCOSEL = 00.	RW	0x0
5:4	SYNCOSEL	Synchronization Output Select. These bits select the source of the EPWMxSYNCO signal. 0x0 = EPWMxSYNC: 0x1 = TBCNT = 0: Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT= 0000h) 0x2 = TBCNT = CMPB : Time-base counter equal to counter-compare B (EPWM_TBCNT bitfield TBCNT = CMPB bitfield in EPWM_CMPB) 0x3 = Disable EPWMxSYNCO signal	RW	0x0
3	PRDLD	Active Period Register Load From Shadow Register Select 0x0 = The period register (EPWM_TBPRD) is loaded from its shadow register when the time-base counter, TBCNT, is equal to zero. A write or read to the EPWM_TBPRD register accesses the shadow register. 0x1 = Load the EPWM_TBPRD register immediately without using a shadow register. A write or read to the EPWM_TBPRD register directly accesses the active register.	RW	0x0
2	PHSEN	Counter Register Load From Phase Register Enable 0x0 = Do not load the time-base counter (EPWM_TBCNT) from the time-base phase register (EPWM_TBPHS) 0x1 = Load the time-base counter with the phase register when an EPWMxSYNCI input signal occurs or when a software synchronization is forced by the SWFSYNC bit.	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	CTRMODE	Counter Mode. The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 0x0 = Up-count mode 0x1 = Down-count mode 0x2 = Up-down-count mode 0x3 = Stop-freeze counter operation (default on reset)	RW	0x0

Table 29-52. Register Call Summary for Register EPWM_TBCTL

PWM Subsystem Resources

- Daisy-Chain Connectivity between PWMSS Modules: [0]

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Time-Base Submodule: [1]
- Calculating PWM Period and Frequency: [2] [3] [4] [5] [6] [7] [8] [9]
- Phase Locking the Time-Base Clocks of Multiple ePWM Modules: [10]
- Operational Highlights for the ePWM Counter-Compare Submodule: [11] [12]
- Waveforms for Common ePWM Configurations: [13] [14] [15] [16] [17] [18]
- eHRPWM Functional Register Groups: [19]
- PWMSS_EPWM Register Summary: [20]
- PWMSS_EPWM Register Description: [21] [22] [23] [24] [25] [26]

Table 29-53. EPWM_TBSTS

Address offset	0x2		
Physical Address	0x4843 E202 0x4844 0202 0x4844 2202	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													CTRMAX	SYNCI	CTRDIR

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0000
2	CTRMAX	Time-Base Counter Max Latched Status Bit. 0x0 = Reading a 0 indicates the time-base counter never reached its maximum value. Writing a 0 will have no effect. 0x1 = Reading a 1 on this bit indicates that the time-base counter reached the max value 0xFFFF. Writing a 1 to this bit will clear the latched event.	RW1C	0x0
1	SYNCI	Input Synchronization Latched Status Bit. 0x0 = Writing a 0 will have no effect. Reading a 0 indicates no external synchronization event has occurred. 0x1 = Reading a 1 on this bit indicates that an external synchronization event has occurred (EPWMxSYNCI). Writing a 1 to this bit will clear the latched event.	RW1C	0x0

Bits	Field Name	Description	Type	Reset
0	CTRDIR	Time-Base Counter Direction Status Bit. At reset, the counter is frozen, therefore, this bit has no meaning. To make this bit meaningful, you must first set the appropriate mode via EPWM_TBCTL[1:0] CTRMODE. 0x0 = Time-Base Counter is currently counting down. 0x1 = Time-Base Counter is currently counting up.	R	0x0

Table 29-54. Register Call Summary for Register EPWM_TBSTS

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Time-Base Submodule: [0]
- eHRPWM Functional Register Groups: [1]
- PWMSS_EPWM Register Summary: [2]

Table 29-55. HRPWM_TBPHSHR

Address offset	0x4		
Physical Address	0x4843 E204 0x4844 0204 0x4844 2204	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHSH								RESERVED							

Bits	Field Name	Description	Type	Reset
15:8	TBPHSH	Time-base phase high-resolution bits	RW	0x0
7:0	RESERVED		R	0x0

Table 29-56. Register Call Summary for Register HRPWM_TBPHSHR

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Time-Base Submodule: [0]
- Controlling and Monitoring the High-Resolution PWM Submodule: [1] [2]
- Configuring the High-Resolution PWM Submodule: [3] [4] [5]
- eHRPWM Functional Register Groups: [6]
- PWMSS_EPWM Register Summary: [7]

Table 29-57. EPWM_TBPHS

Address offset	0x6		
Physical Address	0x4843 E206 0x4844 0206 0x4844 2206	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS															

Bits	Field Name	Description	Type	Reset
15:0	TBPHS	These bits set time-base counter phase of the selected ePWM relative to the time-base that is supplying the synchronization input signal. (a) If EPWM_TBCTL[2] PHSEN = 0, then the synchronization event is ignored and the time-base counter is not loaded with the phase. (b) If EPWM_TBCTL[2] PHSEN = 1, then the time-base counter (TBCNT) will be loaded with the phase (TBPHS) when a synchronization event occurs. The synchronization event can be initiated by the input synchronization signal (EPWMxSYNCI) or by a software forced synchronization.	RW	0x0

Table 29-58. Register Call Summary for Register EPWM_TBPHS

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Time-Base Submodule: [0]
- Calculating PWM Period and Frequency: [1] [2]
- Waveforms for Common ePWM Configurations: [3] [4] [5] [6] [7] [8]
- Controlling and Monitoring the High-Resolution PWM Submodule: [9]
- eHRPWM Functional Register Groups: [10]
- PWMSS_EPWM Register Summary: [11]
- PWMSS_EPWM Register Description: [12] [13]

Table 29-59. EPWM_TBCNT

Address offset	0x8		
Physical Address	0x4843 E208 0x4844 0208 0x4844 2208	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBCNT															

Bits	Field Name	Description	Type	Reset
15:0	TBCNT	Reading these bits gives the current time-base counter value. Writing to these bits sets the current time-base counter value. The update happens as soon as the write occurs. The write is NOT synchronized to the time-base clock (TBCLK) and the register is not shadowed.	RW	0x0

Table 29-60. Register Call Summary for Register EPWM_TBCNT

Enhanced PWM (ePWM) Module

- Purpose of the ePWM Time-Base Submodule: [0] [1]
- Controlling and Monitoring the ePWM Time-Base Submodule: [2] [3] [4] [5]
- Calculating PWM Period and Frequency: [6] [7] [8]
- Operational Highlights for the ePWM Counter-Compare Submodule: [9]
- Waveforms for Common ePWM Configurations: [10] [11] [12] [13] [14] [15]
- Operational Highlights for the ePWM Trip-Zone Submodule: [16]
- Operational Overview of the ePWM Event-Trigger Submodule: [17] [18]
- eHRPWM Functional Register Groups: [19]
- PWMSS_EPWM Register Summary: [20]
- PWMSS_EPWM Register Description: [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34]

Table 29-61. EPWM_TBPRD

Address offset	0xA

Table 29-61. EPWM_TBPRD (continued)

Physical Address	0x4843 E20A 0x4844 020A 0x4844 220A	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRD															

Bits	Field Name	Description	Type	Reset
15:0	TBPRD	These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the EPWM_TBCTL[3] PRDLD bit. By default this register is shadowed. (a) If EPWM_TBCTL[3] PRDLD = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the active register will be loaded from the shadow register when the time-base counter equals zero. (b) If EPWM_TBCTL[3] PRDLD = 1, then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (c) The active and shadow registers share the same memory map address.	RW	0x0

Table 29-62. Register Call Summary for Register EPWM_TBPRD

Enhanced PWM (ePWM) Module

- Purpose of the ePWM Time-Base Submodule: [0]
- Controlling and Monitoring the ePWM Time-Base Submodule: [1]
- Calculating PWM Period and Frequency: [2] [3] [4] [5] [6] [7] [8] [9] [10]
- ePWM Action-Qualifier Event Priority: [11]
- Waveforms for Common ePWM Configurations: [12] [13] [14] [15] [16] [17] [18] [19]
- Operational Overview of the ePWM Event-Trigger Submodule: [20]
- eHRPWM Functional Register Groups: [21]
- PWMSS_EPWM Register Summary: [22]
- PWMSS_EPWM Register Description: [23] [24] [25] [26] [27] [28] [29]

Table 29-63. EPWM_CMPCTL

Address offset	0xE		
Physical Address	0x4843 E20E 0x4844 020E 0x4844 220E	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						SHDWBFULL	SHDWAFULL	RESERVED	SHDWBMODE	RESERVED	SHDWAMODE	LOADBMODE		LOADAMODE	

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9	SHDWBFULL	Counter-compare B (EPWM_CMPB) Shadow Register Full Status Flag. This bit self clears once a load-strobe occurs. 0x0 = CMPB shadow FIFO not full yet 0x1 = Indicates the CMPB shadow FIFO is full. A CPU write will overwrite current shadow value.	R	0x0
8	SHDWAFULL	Counter-compare A (EPWM_CMPA) Shadow Register Full Status Flag. The flag bit is set when a 32 bit write to CMPA:CMPAHR register or a 16 bit write to EPWM_CMPA register is made. A 16 bit write to HRPWM_CMPAHR register will not affect the flag. This bit self clears once a load-strobe occurs. 0x0 = CMPA shadow FIFO not full yet 0x1 = Indicates the CMPA shadow FIFO is full, a CPU write will overwrite the current shadow value.	R	0x0
7	RESERVED		R	0x0
6	SHDWBMODE	Counter-compare B (EPWM_CMPB) Register Operating Mode. 0x0 = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 0x1 = Immediate mode. Only the active compare B register is used. All writes and reads directly access the active register for immediate compare action.	RW	0x0
5	RESERVED		R	0x0
4	SHDWAMODE	Counter-compare A (EPWM_CMPA) Register Operating Mode. 0x0 = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 0x1 = Immediate mode. Only the active compare register is used. All writes and reads directly access the active register for immediate compare action	RW	0x0
3:2	LOADBMODE	Active Counter-Compare B (CMPB) Load From Shadow Select Mode. This bit has no effect in immediate mode (EPWM_CMPCTL[6] SHDWBMODE = 1). 0x0 = Load on TBCNT = 0: Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT= 0000h) 0x1 = Load on TBCNT = PRD: Time-base counter equal to period (EPWM_TBCNT bitfield TBCNT = TBPRD bitfield of the active EPWM_TBPRD) 0x2 = Load on either TBCNT = 0 or TBCNT = PRD 0x3 = Freeze (no loads possible)	RW	0x0
1:0	LOADAMODE	Active Counter-Compare A (EPWM_CMPA) Load From Shadow Select Mode. This bit has no effect in immediate mode (EPWM_CMPCTL[4] SHDWAMODE = 1). 0x0 = Load on TBCNT = 0: Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT = 0000h) 0x1 = Load on TBCNT = PRD: Time-base counter equal to period (EPWM_TBCNT bitfield TBCNT= TBPRD bitfield of the active EPWM_TBPRD) 0x2 = Load on either TBCNT = 0 or TBCNT = PRD 0x3 = Freeze (no loads possible)	RW	0x0

Table 29-64. Register Call Summary for Register EPWM_CMPCTL

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Counter-Compare Submodule: [0]
- Operational Highlights for the ePWM Counter-Compare Submodule: [1] [2] [3] [4] [5] [6]
- Waveforms for Common ePWM Configurations: [7] [8] [9] [10] [11] [12]
- eHRPWM Functional Register Groups: [13]
- PWMSS_EPWM Register Summary: [14]
- PWMSS_EPWM Register Description: [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27]

Table 29-65. HRPWM_CMPAHR

Address offset	0x10		
Physical Address	0x4843 E210 0x4844 0210 0x4844 2210	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPAHR								RESERVED							

Bits	Field Name	Description	Type	Reset
15:8	CMPAHR	Compare A High-Resolution register bits for MEP step control. A minimum value of 1h is needed to enable HRPWM capabilities. Valid MEP range of operation 1-255h.	RW	0x1
7:0	RESERVED		R	0x0

Table 29-66. Register Call Summary for Register HRPWM_CMPAHR

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Counter-Compare Submodule: [0]
- Architecture of the High-Resolution PWM Submodule: [1]
- Controlling and Monitoring the High-Resolution PWM Submodule: [2] [3]
- Configuring the High-Resolution PWM Submodule: [4] [5] [6]
- Operational Highlights for the High-Resolution PWM Submodule: [7] [8] [9] [10] [11] [12]
- eHRPWM Functional Register Groups: [13]
- PWMSS_EPWM Register Summary: [14]
- PWMSS_EPWM Register Description: [15]

Table 29-67. EPWM_CMPA

Address offset	0x12		
Physical Address	0x4843 E212 0x4844 0212 0x4844 2212	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA															

Bits	Field Name	Description	Type	Reset
15:0	CMPA	The value in the active EPWM_CMPA register is continuously compared to the time-base counter (register EPWM_TBCNT). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare A" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the EPWM_AQCTLA and EPWM_AQCTLB registers. The actions that can be defined in the EPWM_AQCTLA and EPWM_AQCTLB registers include the following. (a) Do nothing the event is ignored. (b) Clear: Pull the EPWMxA and/or EPWMxB signal low. (c) Set: Pull the EPWMxA and/or EPWMxB signal high. (d) Toggle the EPWMxA and/or EPWMxB signal. Shadowing of this register is enabled and disabled by the EPWM_CMPCTL[4] SHDWAMODE bit. By default this register is shadowed. (a) If EPWM_CMPCTL[4] SHDWAMODE = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the EPWM_CMPCTL[1:0] LOADAMODE bit field determines which event will load the active register from the shadow register. (b) Before a write, the EPWM_CMPCTL[8] SHDWAFULL bit can be read to determine if the shadow register is currently full. (c) If EPWM_CMPCTL[4] SHDWAMODE = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (d) In either mode, the active and shadow registers share the same memory map address.	RW	0x0

Table 29-68. Register Call Summary for Register EPWM_CMPA

Enhanced PWM (ePWM) Module

- Purpose of the ePWM Counter-Compare Submodule: [0] [1]
- Controlling and Monitoring the ePWM Counter-Compare Submodule: [2]
- Operational Highlights for the ePWM Counter-Compare Submodule: [3] [4] [5] [6] [7]
- Waveforms for Common ePWM Configurations: [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28]
- Operational Overview of the ePWM Event-Trigger Submodule: [29] [30]
- Controlling and Monitoring the High-Resolution PWM Submodule: [31]
- Operational Highlights for the High-Resolution PWM Submodule: [32] [33] [34] [35]
- eHRPWM Functional Register Groups: [36]
- PWMSS_EPWM Register Summary: [37]
- PWMSS_EPWM Register Description: [38] [39] [40] [41] [42] [43] [44] [45] [46]

Table 29-69. EPWM_CMPB

Address offset	0x14		
Physical Address	0x4843 E214 0x4844 0214 0x4844 2214	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB															

Bits	Field Name	Description	Type	Reset
15:0	CMPB	The value in the active EPWM_CMPB register is continuously compared to the time-base counter (register EPWM_TBCNT). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare B" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the EPWM_AQCTLA and EPWM_AQCTLB registers. The actions that can be defined in the EPWM_AQCTLA and EPWM_AQCTLB registers include the following. (a) Do nothing, the event is ignored. (b) Clear: Pull the EPWMxA and/or EPWMxB signal low. (c) Set: Pull the EPWMxA and/or EPWMxB signal high. (d) Toggle the EPWMxA and/or EPWMxB signal. Shadowing of this register is enabled and disabled by the EPWM_CMPCTL[6] SHDWBMODE bit. By default this register is shadowed. (a) If EPWM_CMPCTL[6] SHDWBMODE = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the EPWM_CMPCTL[3:2] LOADBMODE bit field determines which event will load the active register from the shadow register: (b) Before a write, the EPWM_CMPCTL[9] SHDWBFULL bit can be read to determine if the shadow register is currently full. (c) If EPWM_CMPCTL[6] SHDWBMODE = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (d) In either mode, the active and shadow registers share the same memory map address.	RW	0x0

Table 29-70. Register Call Summary for Register EPWM_CMPB

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Time-Base Submodule: [0]
- Purpose of the ePWM Counter-Compare Submodule: [1] [2]
- Controlling and Monitoring the ePWM Counter-Compare Submodule: [3]
- Operational Highlights for the ePWM Counter-Compare Submodule: [4] [5] [6]
- Waveforms for Common ePWM Configurations: [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26]
- Operational Overview of the ePWM Event-Trigger Submodule: [27] [28]
- eHRPWM Functional Register Groups: [29]
- PWMSS_EPWM Register Summary: [30]
- PWMSS_EPWM Register Description: [31] [32] [33] [34] [35] [36] [37] [38]

Table 29-71. EPWM_AQCTLA

Address offset	0x16		
Physical Address	0x4843 E216 0x4844 0216 0x4844 2216	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11:10	CBD	Action when the time-base counter equals the active EPWM_CMPB register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxA output low. 0x2 = Set: force EPWMxA output high. 0x3 = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
9:8	CBU	Action when the counter equals the active EPWM_CMPB register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxA output low. 0x2 = Set: force EPWMxA output high. 0x3 = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
7:6	CAD	Action when the counter equals the active EPWM_CMPA register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxA output low. 0x2 = Set: force EPWMxA output high. 0x3 = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
5:4	CAU	Action when the counter equals the active EPWM_CMPA register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxA output low. 0x2 = Set: force EPWMxA output high. 0x3 = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
3:2	PRD	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxA output low. 0x2 = Set: force EPWMxA output high. 0x3 = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
1:0	ZRO	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxA output low. 0x2 = Set: force EPWMxA output high. 0x3 = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0

Table 29-72. Register Call Summary for Register EPWM_AQCTLA

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Action-Qualifier Submodule: [0]
- Waveforms for Common ePWM Configurations: [1] [2] [3] [4] [5] [6]
- eHRPWM Functional Register Groups: [7]
- PWMSS_EPWM Register Summary: [8]
- PWMSS_EPWM Register Description: [9] [10] [11] [12]

Table 29-73. EPWM_AQCTLB

Address offset	0x18		
Physical Address	0x4843 E218 0x4844 0218 0x4844 2218	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11:10	CBD	Action when the counter equals the active EPWM_CMPB register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxB output low. 0x2 = Set: force EPWMxB output high. 0x3 = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
9:8	CBU	Action when the counter equals the active EPWM_CMPB register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxB output low. 0x2 = Set: force EPWMxB output high. 0x3 = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
7:6	CAD	Action when the counter equals the active EPWM_CMPA register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxB output low. 0x2 = Set: force EPWMxB output high. 0x3 = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
5:4	CAU	Action when the counter equals the active EPWM_CMPA register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxB output low. 0x2 = Set: force EPWMxB output high. 0x3 = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:2	PRD	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxB output low. 0x2 = Set: force EPWMxB output high. 0x3 = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
1:0	ZRO	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWMxB output low. 0x2 = Set: force EPWMxB output high. 0x3 = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0

Table 29-74. Register Call Summary for Register EPWM_AQCTLB

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Action-Qualifier Submodule: [0]
- Waveforms for Common ePWM Configurations: [1] [2] [3] [4] [5] [6]
- eHRPWM Functional Register Groups: [7]
- PWMSS_EPWM Register Summary: [8]
- PWMSS_EPWM Register Description: [9] [10] [11] [12]

Table 29-75. EPWM_AQSFRC

Address offset	0x1A		
Physical Address	0x4843 E21A 0x4844 021A 0x4844 221A	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RLDCSF		OTSFB	ACTSFB		OTSFA	ACTSFA	

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7:6	RLDCSF	EPWM_AQCSFRC Active Register Reload From Shadow Options. 0x0 = Load on event counter equals zero 0x1 = Load on event counter equals period 0x2 = Load on event counter equals zero or counter equals period 0x3 = Load immediately (the active register is directly accessed by the CPU and is not loaded from the shadow register).	RW	0x0
5	OTSFB	One-Time Software Forced Event on Output B. 0x0 = Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete, that is, a forced event is initiated. This is a one-shot forced event. It can be overridden by another subsequent event on output B. 0x1 = Initiates a single s/w forced event	RW	0x0

Bits	Field Name	Description	Type	Reset
4:3	ACTSFB	Action when One-Time Software Force B Is invoked 0x0 = Does nothing (action disabled) 0x1 = Clear (low) 0x2 = Set (high) 0x3 = Toggle (Low -> High, High -> Low). Note: This action is not qualified by counter direction (CNT_dir)	RW	0x0
2	OTSFA	One-Time Software Forced Event on Output A. 0x0 = Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (that is, a forced event is initiated). 0x1 = Initiates a single software forced event.	RW	0x0
1:0	ACTSFA	Action When One-Time Software Force A Is Invoked. 0x0 = Does nothing (action disabled). 0x1 = Clear (low). 0x2 = Set (high). 0x3 = Toggle (Low -> High, High -> Low). Note: This action is not qualified by counter direction (CNT_dir)	RW	0x0

Table 29-76. Register Call Summary for Register EPWM_AQSFRC

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Action-Qualifier Submodule: [0] [1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Register Summary: [3]
- PWMSS_EPWM Register Description: [4]

Table 29-77. EPWM_AQCSFRC

Address offset	0x1C		
Physical Address	0x4843 E21C 0x4844 021C 0x4844 221C	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CSFB		CSFA	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3:2	CSFB	Continuous Software Force on Output B. In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. To configure shadow mode, use EPWM_AQSFRC[7:6] RLDCSF. 0x0 = Forcing disabled, that is, has no effect 0x1 = Forces a continuous low on output B 0x2 = Forces a continuous high on output B 0x3 = Software forcing is disabled and has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	CSFA	Continuous Software Force on Output A In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. 0x0 = Forcing disabled, that is, has no effect 0x1 = Forces a continuous low on output A 0x2 = Forces a continuous high on output A 0x3 = Software forcing is disabled and has no effect	RW	0x0

Table 29-78. Register Call Summary for Register EPWM_AQCSFRC

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Action-Qualifier Submodule: [0] [1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Register Summary: [3]
- PWMSS_EPWM Register Description: [4]

Table 29-79. EPWM_DBCTL

Address offset	0x1E		
Physical Address	0x4843 E21E 0x4844 021E 0x4844 221E	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										IN_MODE		POLSEL		OUT_MODE	

Bits	Field Name	Description	Type	Reset
15:6	RESERVED		R	0x0
5:4	IN_MODE	Dead Band Input Mode Control. Bit 5 controls the S5 switch and bit 4 controls the S4 switch. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms, the default is EPWMxA In is the source for both falling and rising-edge delays. 0x0 = EPWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 0x1 = EPWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 0x2 = EPWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 0x3 = EPWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:2	POLSEL	Polarity Select Control. Bit 3 controls the S3 switch and bit 2 controls the S2 switch. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that EPWM_DBCTL[1:0] OUT_MODE = 0b11 and EPWM_DBCTL[5:4] IN_MODE = 0b00. Other enhanced modes are also possible, but not regarded as typical usage modes. 0x0 = Active high (AH) mode. Neither EPWMxA nor EPWMxB is inverted (default). 0x1 = Active low complementary (ALC) mode. EPWMxA is inverted. 0x2 = Active high complementary (AHC). EPWMxB is inverted. 0x3 = Active low (AL) mode. Both EPWMxA and EPWMxB are inverted.	RW	0x0
1:0	OUT_MODE	Dead-band Output Mode Control. Bit 1 controls the S1 switch and bit 0 controls the S0 switch. This allows you to selectively enable or bypass the dead-band generation for the falling-edge and rising-edge delay. 0x0 = Dead-band generation is bypassed for both output signals. In this mode, both the EPWMxA and EPWMxB output signals from the action-qualifier are passed directly to the PWM-chopper submodule. In this mode, the POLSEL and IN_MODE bits have no effect. 0x1 = Disable rising-edge delay. The EPWMxA signal from the action-qualifier is passed straight through to the EPWMxA input of the PWM-chopper submodule. The falling-edge delayed signal is seen on output EPWMxB. The input signal for the delay is determined by EPWM_DBCTL[5:4] IN_MODE. 0x2 = Disable falling-edge delay. The EPWMxB signal from the action-qualifier is passed straight through to the EPWMxB input of the PWM-chopper submodule. The rising-edge delayed signal is seen on output EPWMxA. The input signal for the delay is determined by EPWM_DBCTL[5:4] IN_MODE. 0x3 = Dead-band is fully enabled for both rising-edge delay on output EPWMxA and falling-edge delay on output EPWMxB. The input signal for the delay is determined by EPWM_DBCTL[5:4] IN_MODE.	RW	0x0

Table 29-80. Register Call Summary for Register EPWM_DBCTL

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Dead-Band Submodule: [0]
- Operational Highlights for the ePWM Dead-Band Generator Submodule: [1] [2] [3] [4] [5] [6]
- eHRPWM Functional Register Groups: [7]
- PWMSS_EPWM Register Summary: [8]
- PWMSS_EPWM Register Description: [9] [10] [11] [12] [13]

Table 29-81. EPWM_DBRED

Address offset	0x20		
Physical Address	0x4843 E220 0x4844 0220 0x4844 2220	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						DEL									

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	DEL	Rising Edge Delay Count. 10 bit counter.	RW	0x0

Table 29-82. Register Call Summary for Register EPWM_DBRED

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Dead-Band Submodule: [0]
- Operational Highlights for the ePWM Dead-Band Generator Submodule: [1] [2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Register Summary: [4]

Table 29-83. EPWM_DBFED

Address offset	0x22		
Physical Address	0x4843 E222 0x4844 0222 0x4844 2222	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						DEL									

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	DEL	Falling Edge Delay Count. 10 bit counter	RW	0x0

Table 29-84. Register Call Summary for Register EPWM_DBFED

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Dead-Band Submodule: [0]
- Operational Highlights for the ePWM Dead-Band Generator Submodule: [1] [2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Register Summary: [4]

Table 29-85. EPWM_TZSEL

Address offset	0x24		
Physical Address	0x4843 E224 0x4844 0224 0x4844 2224	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSHTN								RESERVED							CBC0

Bits	Field Name	Description	Type	Reset
15:8	OSHTN	Trip-zone n (TZn) select. One-Shot (OSHT) trip-zone enable/disable. When any of the enabled pins go low, a one-shot trip event occurs for this ePWM module. When the event occurs, the action defined in the EPWM_TZCTL register is taken on the EPWMxA and EPWMxB outputs. The one-shot trip condition remains latched until you clear the condition via the EPWM_TZCLR register. 0x0 = Disable TZn as a one-shot trip source for this ePWM module. 0x1 = Enable TZn as a one-shot trip source for this ePWM module.	RW	0x0
7:1	RESERVED		R	0x00
0	CBC0	Trip-zone 0 (TZ0) select. Cycle-by-Cycle (CBC) trip-zone enable/ disable. When any of the enabled pins go low, a cycle-by-cycle trip event occurs for this ePWM module. When the event occurs, the action defined in the EPWM_TZCTL register is taken on the EPWMxA and EPWMxB outputs. A cycle-by-cycle trip condition is automatically cleared when the time-base counter reaches zero. 0x0 = Disable TZ0 as a CBC trip source for this ePWM module. 0x1 = Enable TZ0 as a CBC trip source for this ePWM module.	RW	0x0

Table 29-86. Register Call Summary for Register EPWM_TZSEL

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Trip-Zone Submodule: [0]
- Operational Highlights for the ePWM Trip-Zone Submodule: [1] [2] [3]
- ePWM Trip-Zone Configurations: [4] [5]
- eHRPWM Functional Register Groups: [6]
- PWMSS_EPWM Register Summary: [7]
- PWMSS_EPWM Register Description: [8] [9]

Table 29-87. EPWM_TZCTL

Address offset	0x28		
Physical Address	0x4843 E228 0x4844 0228 0x4844 2228	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												TZB		TZA	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3:2	TZB	When a trip event occurs the following action is taken on output EPWMxB. Which trip-zone pins can cause an event is defined in the EPWM_TZSEL register. 0x0 = High impedance (EPWMxB = High-impedance state) 0x1 = Force EPWMxB to a high state 0x2 = Force EPWMxB to a low state 0x3 = Do nothing, no action is taken on EPWMxB.	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	TZA	When a trip event occurs the following action is taken on output EPWMxA. Which trip-zone pins can cause an event is defined in the EPWM_TZSEL register. 0x0 = High impedance (EPWMxA = High-impedance state) 0x1 = Force EPWMxA to a high state 0x2 = Force EPWMxA to a low state 0x3 = Do nothing, no action is taken on EPWMxA.	RW	0x0

Table 29-88. Register Call Summary for Register EPWM_TZCTL

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Trip-Zone Submodule: [0]
- Operational Highlights for the ePWM Trip-Zone Submodule: [1] [2] [3] [4] [5] [6]
- ePWM Trip-Zone Configurations: [7] [8] [9] [10]
- eHRPWM Functional Register Groups: [11]
- PWMSS_EPWM Register Summary: [12]
- PWMSS_EPWM Register Description: [13] [14]

Table 29-89. EPWM_TZEINT

Address offset	0x2A		
Physical Address	0x4843 E22A 0x4844 022A 0x4844 222A	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	RESERVED

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Trip-zone One-Shot Interrupt Enable 0x0 = Disable one-shot interrupt generation 0x1 = Enable Interrupt generation; a one-shot trip event will cause a EPWMxTZINT interrupt.	RW	0x0
1	CBC	Trip-zone Cycle-by-Cycle Interrupt Enable 0x0 = Disable cycle-by-cycle interrupt generation. 0x1 = Enable interrupt generation; a cycle-by-cycle trip event will cause an EPWMxTZINT interrupt.	RW	0x0
0	RESERVED		R	0x0

Table 29-90. Register Call Summary for Register EPWM_TZEINT

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Trip-Zone Submodule: [0]
- Operational Highlights for the ePWM Trip-Zone Submodule: [1] [2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Register Summary: [4]

Table 29-91. EPWM_TZFLG

Address offset	0x2C

Table 29-91. EPWM_TZFLG (continued)

Physical Address	0x4843 E22C 0x4844 022C 0x4844 222C	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	INT

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Latched Status Flag for A One-Shot Trip Event. 0x0 = No one-shot trip event has occurred. 0x1 = Indicates a trip event has occurred on a pin selected as a one-shot trip source. This bit is cleared by writing the appropriate value to the EPWM_TZCLR register.	R	0x0
1	CBC	Latched Status Flag for Cycle-By-Cycle Trip Event 0x0 = No cycle-by-cycle trip event has occurred. 0x1 = Indicates a trip event has occurred on a pin selected as a cycle-by-cycle trip source. The EPWM_TZFLG[1] CBC bit will remain set until it is manually cleared by the user. If the cycle-by-cycle trip event is still present when the CBC bit is cleared, then CBC will be immediately set again. The specified condition on the pins is automatically cleared when the ePWM time-base counter reaches zero (EPWM_TBCNT bitfield TBCNT = 0000h) if the trip condition is no longer present. The condition on the pins is only cleared when the TBCNT = 0000h no matter where in the cycle the CBC flag is cleared. This bit is cleared by writing the appropriate value to the EPWM_TZCLR register.	R	0x0
0	INT	Latched Trip Interrupt Status Flag 0x0 = Indicates no interrupt has been generated. 0x1 = Indicates an EPWMxTZINT interrupt was generated because of a trip condition. No further EPWMxTZINT interrupts will be generated until this flag is cleared. If the interrupt flag is cleared when either CBC or OST is set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. This bit is cleared by writing the appropriate value to the EPWM_TZCLR register.	R	0x0

Table 29-92. Register Call Summary for Register EPWM_TZFLG

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Trip-Zone Submodule: [0]
- Operational Highlights for the ePWM Trip-Zone Submodule: [1] [2] [3] [4]
- eHRPWM Functional Register Groups: [5]
- PWMSS_EPWM Register Summary: [6]
- PWMSS_EPWM Register Description: [7] [8] [9] [10] [11]

Table 29-93. EPWM_TZCLR

Address offset	0x2E		
Physical Address	0x4843 E22E 0x4844 022E 0x4844 222E	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM

Table 29-93. EPWM_TZCLR (continued)

Description	
Type	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	INT

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Clear Flag for One-Shot Trip (OST) Latch 0x0 = Has no effect. Always reads back a 0. 0x1 = Clears this Trip (set) condition.	RW	0x0
1	CBC	Clear Flag for Cycle-By-Cycle (CBC) Trip Latch 0x0 = Has no effect. Always reads back a 0. 0x1 = Clears this Trip (set) condition.	RW	0x0
0	INT	Global Interrupt Clear Flag 0x0 = Has no effect. Always reads back a 0. 0x1 = Clears the trip-interrupt flag for this ePWM module (EPWM_TZFLG[0] INT). Note: No further EPWMxTZINT interrupts will be generated until the flag is cleared. If the EPWM_TZFLG[0] INT bit is cleared and any of the other flag bits are set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts.	RW	0x0

Table 29-94. Register Call Summary for Register EPWM_TZCLR

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Trip-Zone Submodule: [0]
- Operational Highlights for the ePWM Trip-Zone Submodule: [1] [2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Register Summary: [4]
- PWMSS_EPWM Register Description: [5] [6] [7] [8]

Table 29-95. EPWM_TZFRC

Address offset	0x30		
Physical Address	0x4843 E230 0x4844 0230 0x4844 2230	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	RESERVED

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Force a One-Shot Trip Event via Software 0x0 = Writing of 0 is ignored. Always reads back a 0. 0x1 = Forces a one-shot trip event and sets the EPWM_TZFLG[2] OST bit.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	CBC	Force a Cycle-by-Cycle Trip Event via Software 0x0 = Writing of 0 is ignored. Always reads back a 0. 0x1 = Forces a cycle-by-cycle trip event and sets the EPWM_TZFLG[1] CBC bit.	RW	0x0
0	RESERVED		R	0x0

Table 29-96. Register Call Summary for Register EPWM_TZFRC

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Trip-Zone Submodule: [0]
- eHRPWM Functional Register Groups: [1]
- PWMSS_EPWM Register Summary: [2]

Table 29-97. EPWM_ETSEL

Address offset	0x32		
Physical Address	0x4843 E232 0x4844 0232 0x4844 2232	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												INTEN	INTSEL		

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3	INTEN	Enable ePWM Interrupt (EPWMx_INT) Generation 0x0 = Disable EPWMx_INT generation 0x1 = Enable EPWMx_INT generation	RW	0x0
2:0	INTSEL	ePWM Interrupt (EPWMx_INT) Selection Options 0x0 = Reserved 0x1 = Enable event time-base counter equal to zero. (TBCNT = 0000h) 0x2 = Enable event time-base counter equal to period (TBCNT = TBPRD) 0x3 = Reserved 0x4 = Enable event time-base counter equal to CMPA when the timer is incrementing. 0x5 = Enable event time-base counter equal to CMPA when the timer is decrementing. 0x6 = Enable event: time-base counter equal to CMPB when the timer is incrementing. 0x7 = Enable event: time-base counter equal to CMPB when the timer is decrementing.	RW	0x0

Table 29-98. Register Call Summary for Register EPWM_ETSEL

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Event-Trigger Submodule: [0]
- Operational Overview of the ePWM Event-Trigger Submodule: [1] [2] [3]
- eHRPWM Functional Register Groups: [4]
- PWMSS_EPWM Register Summary: [5]
- PWMSS_EPWM Register Description: [6] [7] [8] [9] [10]

Table 29-99. EPWM_ETPS

Address offset	0x34		
Physical Address	0x4843 E234 0x4844 0234 0x4844 2234	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												INTCNT		INTPRD	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3:2	INTCNT	ePWM Interrupt Event (EPWMx_INT) Counter Register. These bits indicate how many selected EPWM_ETSEL[2:0] INTSEL events have occurred. These bits are automatically cleared when an interrupt pulse is generated. If interrupts are disabled, EPWM_ETSEL[0] INT = 0 or the interrupt flag is set, EPWM_ETFLG[0] INT = 1, the counter will stop counting events when it reaches the period value EPWM_ETPS[3:2] INTCNT = EPWM_ETPS[1:0] INTPRD. 0x0 = No events have occurred. 0x1 = 1 event has occurred. 0x2 = 2 events have occurred. 0x3 = 3 events have occurred.	R	0x0
1:0	INTPRD	ePWM Interrupt (EPWMx_INT) Period Select. These bits determine how many selected EPWM_ETSEL[2:0] INTSEL events need to occur before an interrupt is generated. To be generated, the interrupt must be enabled (EPWM_ETSEL[0] INT = 1). If the interrupt status flag is set from a previous interrupt (EPWM_ETFLG[0] INT = 1) then no interrupt will be generated until the flag is cleared via the EPWM_ETCLR[0] INT bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the EPWM_ETPS[3:2] INTCNT bits will automatically be cleared. Writing a INTPRD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a INTPRD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero INTPRD value is written, the counter is incremented. 0x0 = Disable the interrupt event counter. No interrupt will be generated and EPWM_ETFRC[0] INT is ignored. 0x1 = Generate an interrupt on the first event INTCNT = 01 (first event) 0x2 = Generate interrupt on EPWM_ETPS[3:2] INTCNT = 0b10 (second event) 0x3 = Generate interrupt on EPWM_ETPS[3:2] INTCNT = 0b11 (third event)	RW	0x0

Table 29-100. Register Call Summary for Register EPWM_ETPS

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Event-Trigger Submodule: [0]
- Operational Overview of the ePWM Event-Trigger Submodule: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]
- eHRPWM Functional Register Groups: [12]
- PWMSS_EPWM Register Summary: [13]
- PWMSS_EPWM Register Description: [14] [15] [16] [17] [18]

Table 29-101. EPWM_ETFLG

Address offset	0x36		
Physical Address	0x4843 E236 0x4844 0236 0x4844 2236	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															INT

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0
0	INT	Latched ePWM Interrupt (EPWMx_INT) Status Flag 0x0 = Indicates no event occurred 0x1 = Indicates that an ePWMx interrupt (EWPMx_INT) was generated. No further interrupts will be generated until the flag bit is cleared. Up to one interrupt can be pending while the EPWM_ETFLG[0] INT bit is still set. If an interrupt is pending, it will not be generated until after the EPWM_ETFLG[0] INT bit is cleared.	R	0x0

Table 29-102. Register Call Summary for Register EPWM_ETFLG

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Event-Trigger Submodule: [0]
- Operational Overview of the ePWM Event-Trigger Submodule: [1] [2] [3] [4]
- eHRPWM Functional Register Groups: [5]
- PWMSS_EPWM Register Summary: [6]
- PWMSS_EPWM Register Description: [7] [8] [9] [10] [11]

Table 29-103. EPWM_ETCLR

Address offset	0x38		
Physical Address	0x4843 E238 0x4844 0238 0x4844 2238	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															INT

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	INT	ePWM Interrupt (EPWMx_INT) Flag Clear Bit 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing 1 clears the EPWM_ETFLG[0] INT flag bit and enable further interrupts pulses to be generated.	RW	0x0

Table 29-104. Register Call Summary for Register EPWM_ETCLR

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Event-Trigger Submodule: [0]
- eHRPWM Functional Register Groups: [1]
- PWMSS_EPWM Register Summary: [2]
- PWMSS_EPWM Register Description: [3]

Table 29-105. EPWM_ETFRC

Address offset	0x3A		
Physical Address	0x4843 E23A 0x4844 023A 0x4844 223A	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															INT

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0
0	INT	INT Force Bit. The interrupt will only be generated if the event is enabled in the EPWM_ETSEL register. The INT flag bit will be set regardless. 0x0 = Writing 0 to this bit will be ignored. Always reads back a 0. 0x1 = Writing 1 generates an interrupt on EPWMxINT and set the INT flag bit. This bit is used for test purposes.	RW	0x0

Table 29-106. Register Call Summary for Register EPWM_ETFRC

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the ePWM Event-Trigger Submodule: [0]
- Operational Overview of the ePWM Event-Trigger Submodule: [1] [2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Register Summary: [4]
- PWMSS_EPWM Register Description: [5]

Table 29-107. EPWM_PCCTL

Address offset	0x3C		
Physical Address	0x4843 E23C 0x4844 023C 0x4844 223C	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CHPDUTY			CHPFREQ			OSHTWTH				CHPEN

Bits	Field Name	Description	Type	Reset
15:11	RESERVED		R	0x0
10:8	CHPDUTY	Chopping Clock Duty Cycle 0x0 = Duty = 1/8 (12.5%) 0x1 = Duty = 2/8 (25.0%) 0x2 = Duty = 3/8 (37.5%) 0x3 = Duty = 4/8 (50.0%) 0x4 = Duty = 5/8 (62.5%) 0x5 = Duty = 6/8 (75.0%) 0x6 = Duty = 7/8 (87.5%) 0x7 = Reserved.	RW	0x0
7:5	CHPFREQ	Chopping Clock Frequency 0x0 = Divide by 1 (no prescale). 0x1 = Divide by 2. 0x2 = Divide by 3. 0x3 = Divide by 4. 0x4 = Divide by 5. 0x5 = Divide by 6. 0x6 = Divide by 7. 0x7 = Divide by 8.	RW	0x0
4:1	OSHTWTH	One-Shot Pulse Width 0x0 = 1 - SYSCLKOUT/8 wide 0x1 = 2 - SYSCLKOUT/8 wide 0x2 = 3 - SYSCLKOUT/8 wide 0x3 = 4 - SYSCLKOUT/8 wide 0xF = 16 - SYSCLKOUT/8 wide	RW	0x0
0	CHPEN	PWM-chopping Enable 0x0 = Disable (bypass) PWM chopping function 0x1 = Enable chopping function	RW	0x0

Table 29-108. Register Call Summary for Register EPWM_PCCTL

Enhanced PWM (ePWM) Module

- Controlling the PWM-Chopper Submodule: [0]
- Operational Highlights for the PWM-Chopper Submodule: [1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Register Summary: [3]

Table 29-109. HRPWM_HRCTL

Address offset	0xC0		
Physical Address	0x4843 E2C0 0x4844 02C0 0x4844 22C0	**Instance**	PWMSS1_EPWM PWMSS2_EPWM PWMSS3_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PULSESEL	DELBUSSEL	DELMODE	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3	PULSESEL	Pulse select bits. Selects which pulse to use for timing events in the HRPWM module. Note: The user needs to select the pulse to match the selection in the EPWM module. If TBPHSHR bus is selected, then CNT_zero pulse should be used. If COMPAHR bus is selected, then it should match the bit setting of the EPWM_CMPCTL[LOADMODE] bits in the EPWM module as follows. 0: CNT_zero pulse. 1h: PRD_eq pulse. 2h: CNT_zero or PRD_eq (should not use with HRPWM). 3h: No loads (should not use with HRPWM). 0x0 = Select CNT_zero pulse 0x1 = Select PRD_eq pulse	RW	0x0
2	DELBUSSEL	Delay Bus Select Bit: Selects which bus is used to select the delay for the PWM pulse. 0x0 = Select CMPAHR(8) bus from compare module of EPWM (default on reset). 0x1 = Select TBPHSHR(8) bus from time base module.	RW	0x0
1:0	DELMODE	Delay Mode Bits: Selects which edge of the PWM pulse the delay is inserted. Note: When DELMODE = 0b00, the HRCALM[CALMODE] bits are ignored and the delay line is in by-pass mode. Additionally, DLYIN is connected to CALIN and a continuous low value is fed to the delay line to minimize activity in the module. 0x0 = No delay inserted (default on reset) 0x1 = Delay inserted rising edge 0x2 = Delay inserted falling edge 0x3 = Delay inserted on both edges	RW	0x0

Table 29-110. Register Call Summary for Register HRPWM_HRCTL

Enhanced PWM (ePWM) Module

- Controlling and Monitoring the High-Resolution PWM Submodule: [0]
- Configuring the High-Resolution PWM Submodule: [1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Register Summary: [3]

29.3 Enhanced Capture (eCAP) Module

29.3.1 eCAP Overview

29.3.2 Purpose of the eCAP Peripheral

Uses for eCAP include:

- Sample rate measurements of audio inputs
- Speed measurements of rotating machinery (for example, toothed sprockets sensed via Hall sensors)
- Elapsed time measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

29.3.3 eCAP Features

The eCAP module includes the following features:

- 32-bit time base counter
- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single shot capture of up to four event time-stamps
- Continuous mode capture of time-stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- All above resources dedicated to a single input pin
- When not used in capture mode, the ECAP module can be configured as a single channel PWM output

29.3.4 eCAP Functional Description

The eCAP module represents one complete capture channel that can be instantiated multiple times depending on the target device. In the context of this guide, one eCAP channel has the following independent key resources:

- Dedicated input capture pin
- 32-bit time base counter
- 4 × 32-bit time-stamp capture registers (PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4)
- 4-stage sequencer (Modulo4 counter) that is synchronized to external events, ECAP pin rising/falling edges.
- Independent edge polarity (rising/falling edge) selection for all 4 events
- Input capture signal prescaling (from 2-62)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 time-stamp events
- Control for continuous time-stamp captures using a 4-deep circular buffer (PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

Multiple identical eCAP modules can be contained in a system as shown in Figure 29-53. For actual number of eCAP modules integrated in the device, refer to the Section 29.1.3. As already described in Section 29.1.3, the letter x within a signal or module name is used to indicate a generic eCAP instance on a device. For example, output interrupt request, ECAP1INT belongs to eCAP1, ECAP2INT belongs to eCAP2, etc.

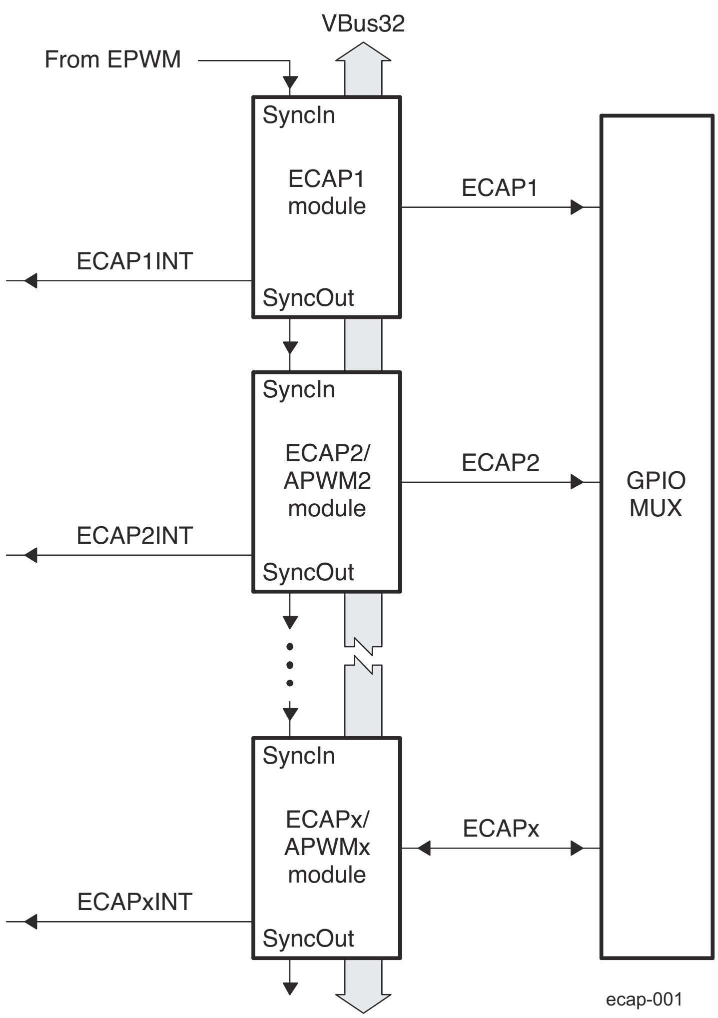
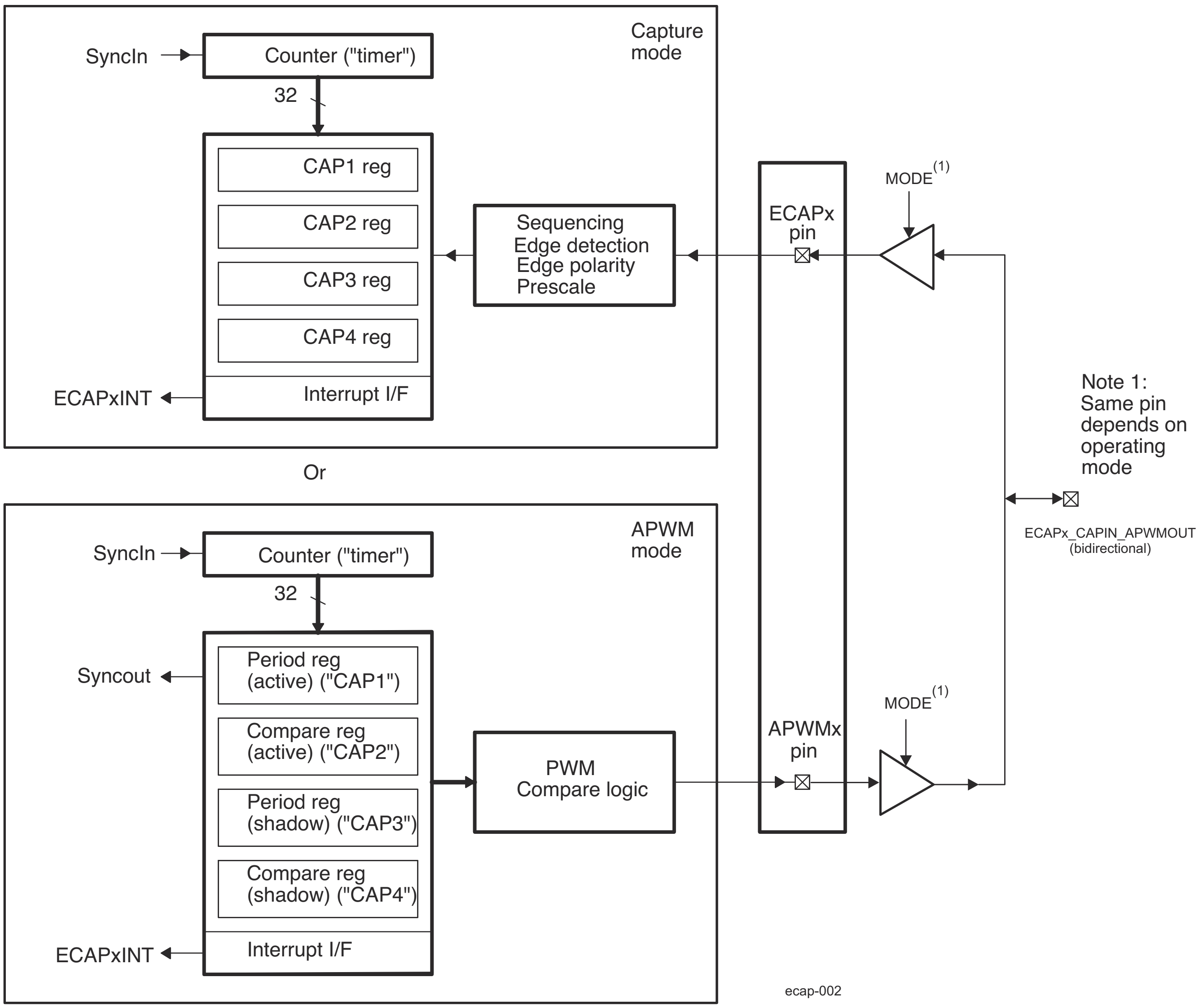


Figure 29-53. Multiple eCAP Modules

29.3.5 Capture and APWM Operating Mode

The eCAP module resources can be used to implement a single-channel PWM generator (with 32 bit capabilities) when it is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers become the active period and compare registers, respectively, while PWMSS_ECAP_CAP3 and PWMSS_ECAP_CAP4 registers become the period and capture shadow registers, respectively. Figure 29-54 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.



A. A single pin is shared between CAP and APWM functions. In capture mode, it is an input; in APWM mode, it is an output.

B. In APWM mode, writing any value to PWMSS_ECAP_CAP1/PWMSS_ECAP_CAP2 active registers also writes the same value to the corresponding shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4. This emulates immediate mode. Writing to the shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4 invokes the shadow mode.

Figure 29-54. Capture and APWM Modes of Operation

29.3.6 eCAP Capture Mode Description

Figure 29-55 shows the various components that implement the capture function.

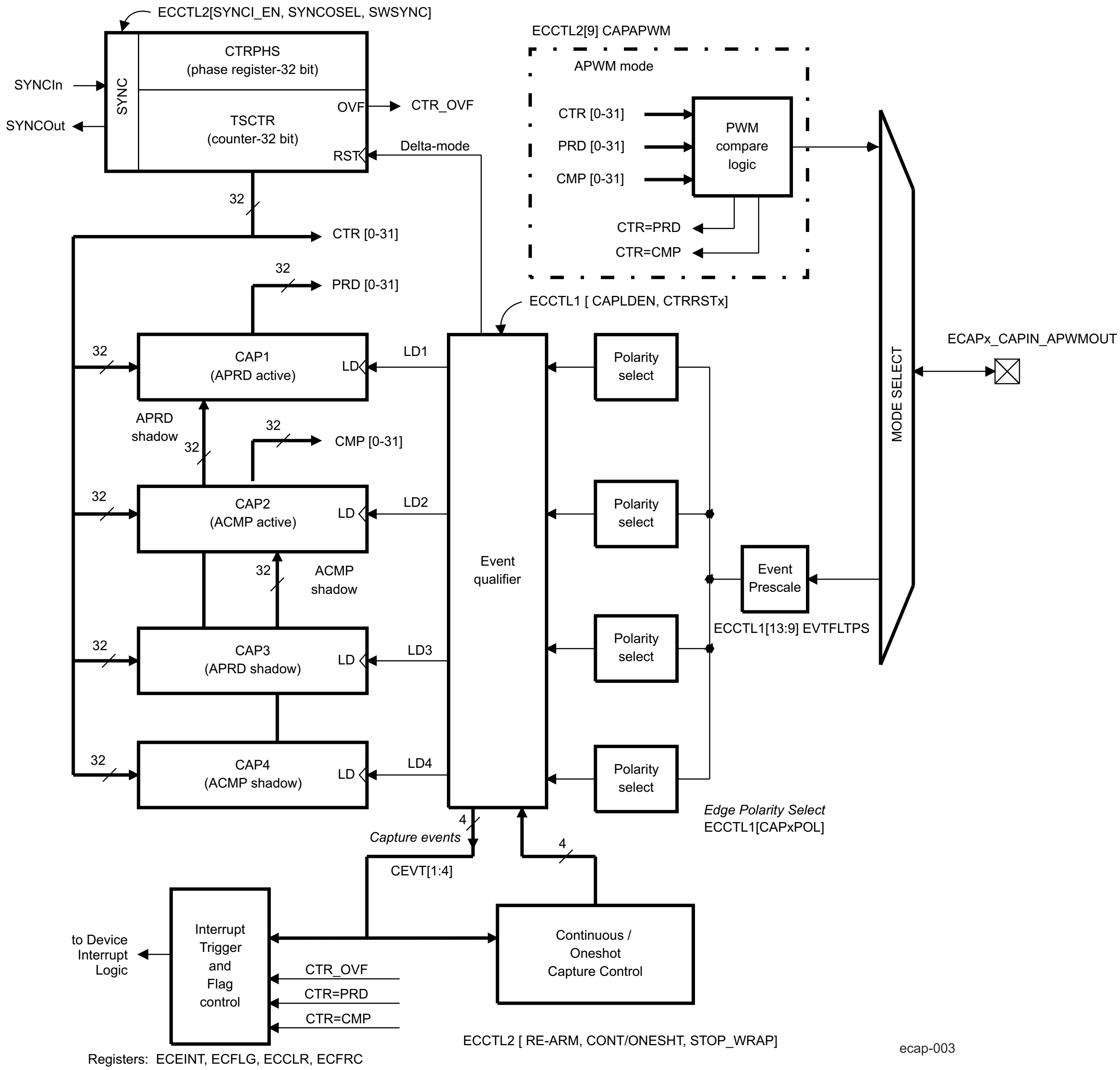
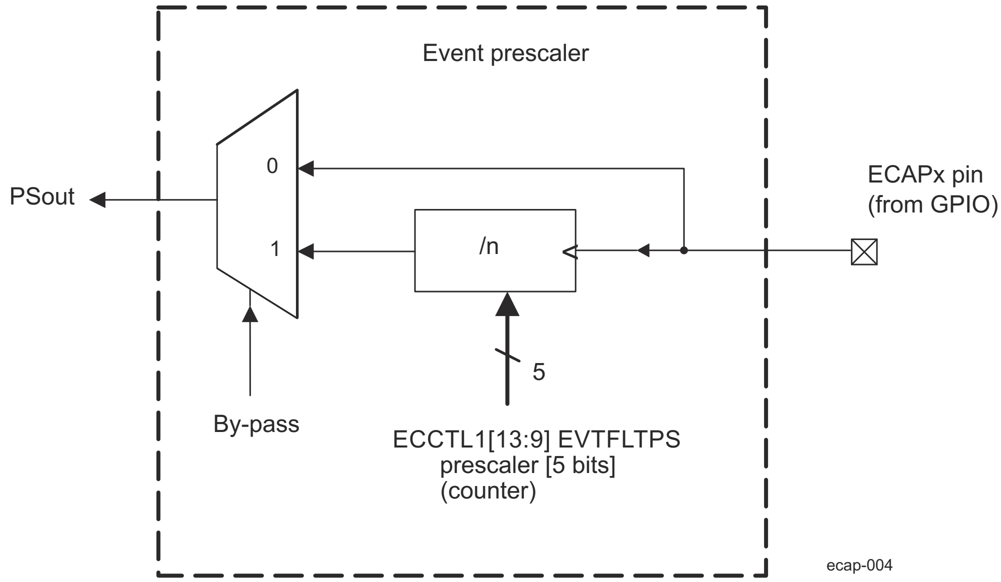


Figure 29-55. Capture Function Diagram

29.3.7 eCAP Event Prescaler

An input capture signal (pulse train) can be prescaled by N = 2-62 (in multiples of 2) or can bypass the prescaler. This is useful when very high frequency signals are used as inputs. Figure 29-56 shows a functional diagram and Figure 29-57 shows the operation of the prescale function.



A. When a prescale value of 1 is chosen (PWMSS_ECAP_ECCTL1[13:9] EVTFLTPS = 0b0000) the input capture signal by-passes the prescale logic completely.

Figure 29-56. Event Prescale Control

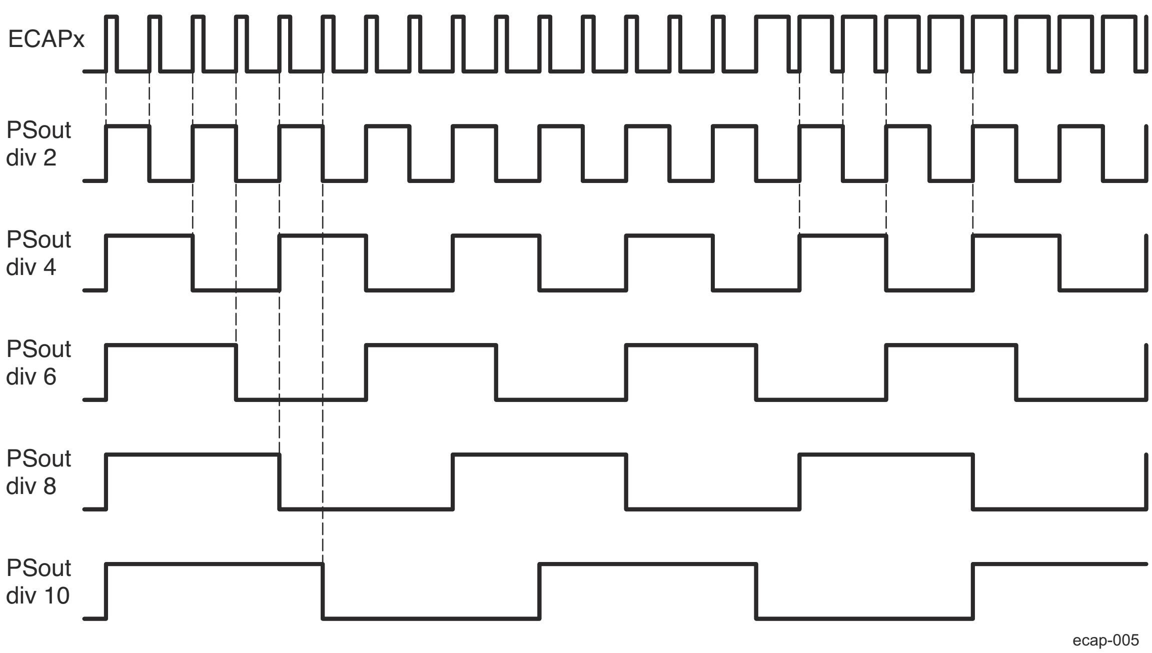


Figure 29-57. Prescale Function Waveforms

29.3.8 eCAP Edge Polarity Select and Qualifier

- Four independent edge polarity (rising edge/falling edge) selection multiplexers are used, one for each capture event.
- Each edge (up to 4) is event qualified by the Modulo4 sequencer.
- The edge event is gated to its respective CAP*n* register by the Mod4 counter. The CAP*n* register is loaded on the falling edge.

29.3.9 eCAP Continuous/One-Shot Control

- The Mod4 (2 bit) counter is incremented via edge qualified events (CEVT1-CEVT4).
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- A 2-bit stop register is used to compare the Mod4 counter output, and when equal stops the Mod4 counter and inhibits further loads of the PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 registers. This occurs during one-shot operation.

The continuous/one-shot block (Figure 29-58) controls the start/stop and reset (zero) functions of the Mod4 counter via a mono-shot type of action that can be triggered by the stop-value comparator and re-armed via software control.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of PWMSS_ECAP_CAP1-4 registers (time-stamps).

Re-arming prepares the eCAP module for another capture sequence. Also re-arming clears (to zero) the Mod4 counter and permits loading of PWMSS_ECAP_CAP1-4 registers again, providing the CAPLDEN bit is set.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to PWMSS_ECAP_CAP1-4 in a circular buffer sequence.

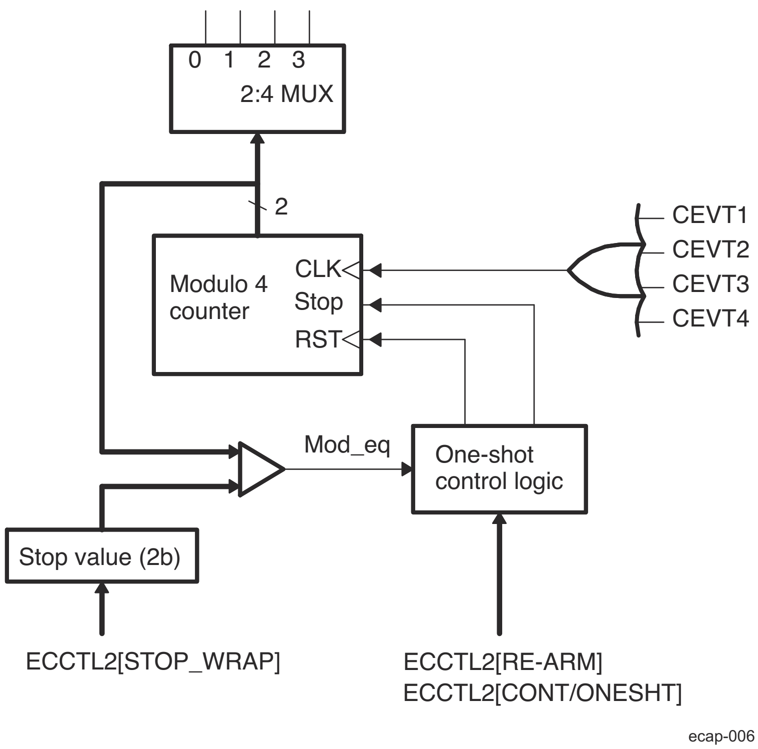


Figure 29-58. eCAP Continuous/One-shot Block Diagram

29.3.10 eCAP 32-Bit Counter and Phase Control

This counter (Figure 29-59) provides the time-base for event captures, and is clocked via the system clock.

A phase register is provided to achieve synchronization with other counters, via a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then it is reset to 0 by any of the LD1-LD4 signals.

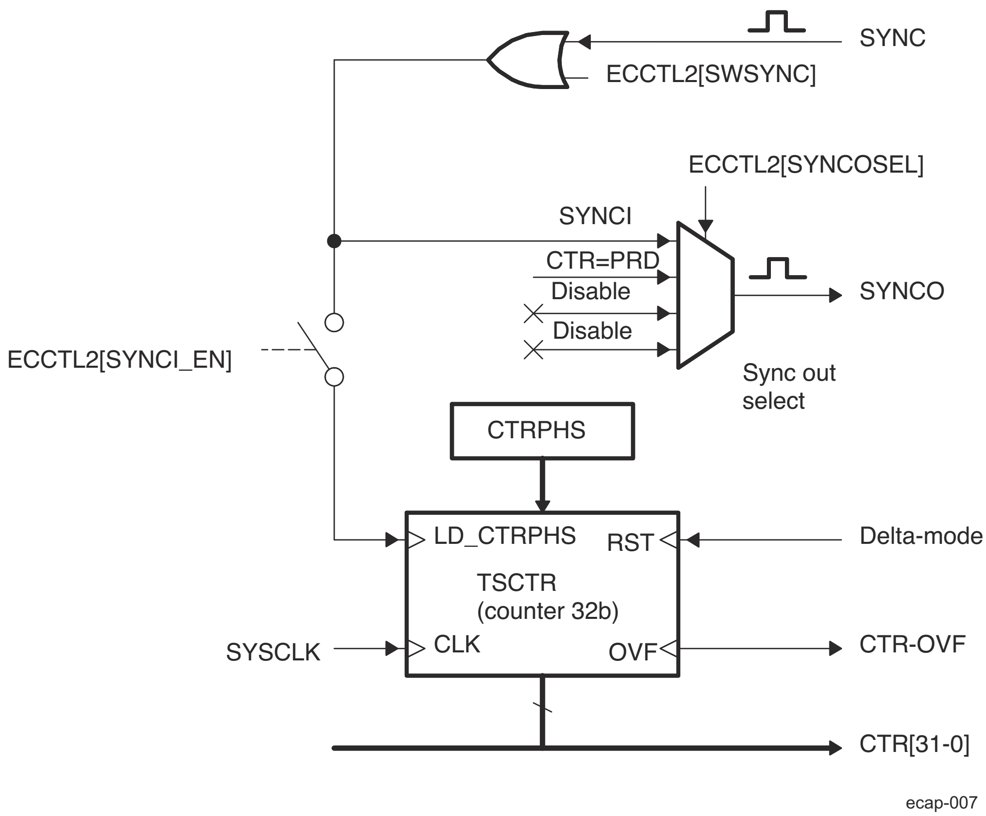


Figure 29-59. eCAP Counter and Synchronization Block Diagram

29.3.11 CAP1-CAP4 Registers

These 32-bit registers are fed by the 32-bit counter timer bus, CTR[0-31] and are loaded (capture a time-stamp) when their respective LD inputs are strobed.

Loading of the capture registers can be inhibited via control bit CAPLDEN. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, StopValue = Mod4.

PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers become the active period and compare registers, respectively, in APWM mode.

PWMSS_ECAP_CAP3 and PWMSS_ECAP_CAP4 registers become the respective shadow registers (APRD and ACMP) for PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 during APWM operation.

29.3.12 eCAP Interrupt Control

An Interrupt can be generated on capture events (CEVT1-CEVT4, CNTOVF) or APWM events (TSCNT = PRD, TSCNT = CMP). See Figure 29-60.

A counter overflow event (FFFF FFFFh->0000 0000h) is also provided as an interrupt source (CNTOVF).

The capture events are edge and sequencer qualified (that is, ordered in time) by the polarity select and Mod4 gating, respectively.

One of these events can be selected as the interrupt source (from the eCAP*n* module) going to the interrupt controller.

Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, TSCNT = PRD, TSCNT = CMP) can be generated. The interrupt enable register (PWMSS_ECAP_ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (PWMSS_ECAP_ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the interrupt controller only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event via the interrupt clear register (PWMSS_ECAP_ECCLR) before any other interrupt pulses are generated. You can force an interrupt event via the interrupt force register (PWMSS_ECAP_ECFRC). This is useful for test purposes.

29.3.13 eCAP Shadow Load and Lockout Control

In capture mode, this logic inhibits (locks out) any shadow loading of PWMSS_ECAP_CAP1 or PWMSS_ECAP_CAP2 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate - APRD or ACMP are transferred to PWMSS_ECAP_CAP1 or PWMSS_ECAP_CAP2 immediately upon writing a new value.
- On period equal, CTR[31:0] = PRD[31:0]

Note

The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (PWMSS_ECAP_ECCTL2[9] CAPAPWM == 0). The TSCNT = PRD, TSCNT = CMP flags are only valid in APWM mode (PWMSS_ECAP_ECCTL2[9] CAPAPWM == 1). CNTOVF flag is valid in both modes.

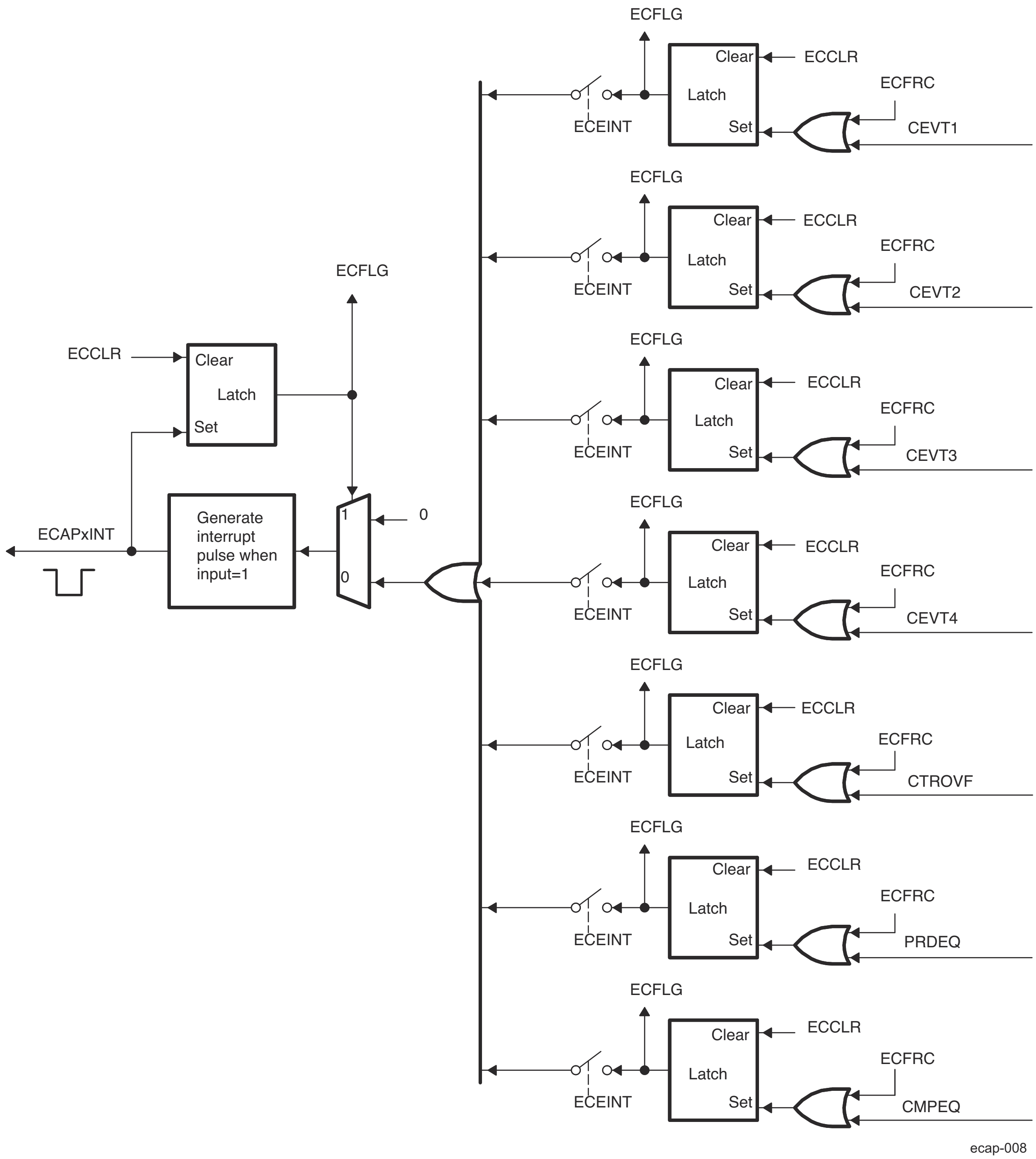


Figure 29-60. Interrupts in eCAP Module

29.3.14 eCAP Module APWM Mode Operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison via 2 digital (32-bit) comparators.
- When PWMSS_ECAP_CAP1/2 registers are not used in capture mode, their contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved via shadow registers APRD and ACMP (PWMSS_ECAP_CAP3/4). The shadow register contents are transferred over to PWMSS_ECAP_CAP1/2 registers either immediately upon a write, or on a TSCNT = PRD trigger.
- In APWM mode, writing to PWMSS_ECAP_CAP1/PWMSS_ECAP_CAP2 active registers will also write the same value to the corresponding shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4. This emulates immediate mode. Writing to the shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4 will invoke the shadow mode.
- During initialization, you must write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates, during run-time, you only need to use the shadow registers.

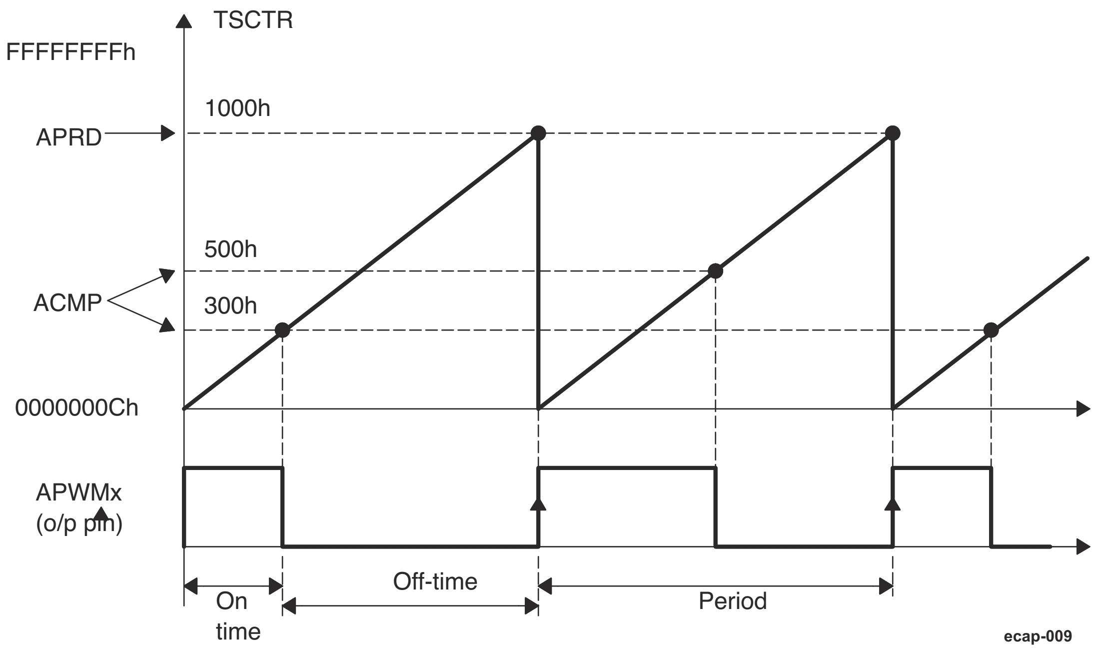


Figure 29-61. PWM Waveform Details Of eCAP APWM Mode Operation

The behavior of APWM active-high mode (APWMPOL == 0) is:

```
CMP = 0x00000000, output low for duration of period (0% duty)
```

```
CMP = 0x00000001, output high 1 cycle
```

```
CMP = 0x00000002, output high 2 cycles
```

```
CMP = PERIOD, output high except for 1 cycle (<100% duty)
```

```
CMP = PERIOD+1, output high for complete period (100% duty)
```

```
CMP > PERIOD+1, output high for complete period
```

The behavior of APWM active-low mode (APWMPOL == 1) is:

```
CMP = 0x00000000, output high for duration of period (0% duty)
```

```
CMP = 0x00000001, output low 1 cycle
```

```
CMP = 0x00000002, output low 2 cycles
```

```
CMP = PERIOD, output low except for 1 cycle (<100% duty)
```

```
CMP = PERIOD+1, output low for complete period (100% duty)
```

```
CMP > PERIOD+1, output low for complete period
```

29.3.15 Summary of eCAP Functional Registers

Table 29-111 shows the eCAP module control and status register set. All 32-bit registers are aligned on even address boundaries and are organized in little-endian mode. The 16 least-significant bits of a 32-bit register are located on lowest address (even address).

Note

In APWM mode, writing to PWMSS_ECAP_CAP1/PWMSS_ECAP_CAP2 active registers also writes the same value to the corresponding shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4. This emulates immediate mode. Writing to the shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4 invokes the shadow mode.

Table 29-111. eCAP Control and Status Functional Registers

Offset	Register Name	Description	Size (×16)
0h	PWMSS_ECAP_TSCNT	Time-Stamp Counter Register	2
4h	PWMSS_ECAP_CNTPHS	Counter Phase Offset Value Register	2
8h	PWMSS_ECAP_CAP1	Capture 1 Register	2
Ch	PWMSS_ECAP_CAP2	Capture 2 Register	2
10h	PWMSS_ECAP_CAP3	Capture 3 Register	2
14h	PWMSS_ECAP_CAP4	Capture 4 Register	2
28h	PWMSS_ECAP_ECCTL1	Capture Control Register 1	1
2Ah	PWMSS_ECAP_ECCTL2	Capture Control Register 2	1
2Ch	PWMSS_ECAP_ECEINT	Capture Interrupt Enable Register	1
2Eh	PWMSS_ECAP_ECFLG	Capture Interrupt Flag Register	1
30h	PWMSS_ECAP_ECCLR	Capture Interrupt Clear Register	1
32h	PWMSS_ECAP_ECFRC	Capture Interrupt Force Register	1
5Ch	PWMSS_ECAP_PID	Revision ID Register	2

29.3.16 PWMSS_ECAP Register Manual

This section provides description of the PWMSS eCAP relevant functional registers.

29.3.17 PWMSS_ECAP Instance Summary

Table 29-112. PWMSS_ECAP Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS1_ECAP	0x4843 E100	400 Bytes
PWMSS2_ECAP	0x4844 0100	400 Bytes
PWMSS3_ECAP	0x4844 2100	400 Bytes

29.3.18 PWMSS_ECAP Registers

29.3.19 PWMSS_ECAP Register Summary

Table 29-113. PWMSSn_ECAP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS1_ECAP Physical Address L4_PER2 Interconnect	PWMSS2_ECAP Physical Address L4_PER2 Interconnect	PWMSS3_ECAP Physical Address L4_PER2 Interconnect
PWMSS_ECAP_TSCNT	RW	32	0x0	0x4843 E100	0x4844 0100	0x4844 2100
PWMSS_ECAP_CNTPHS	RW	32	0x4	0x4843 E104	0x4844 0104	0x4844 2104
PWMSS_ECAP_CAP1	RW	32	0x8	0x4843 E108	0x4844 0108	0x4844 2108
PWMSS_ECAP_CAP2	RW	32	0xC	0x4843 E10C	0x4844 010C	0x4844 210C
PWMSS_ECAP_CAP3	RW	32	0x10	0x4843 E110	0x4844 0110	0x4844 2110
PWMSS_ECAP_CAP4	RW	32	0x14	0x4843 E114	0x4844 0114	0x4844 2114
PWMSS_ECAP_ECCTL1	RW	16	0x28	0x4843 E128	0x4844 0128	0x4844 2128
PWMSS_ECAP_ECCTL2	RW	16	0x2A	0x4843 E12A	0x4844 012A	0x4844 212A
PWMSS_ECAP_ECEINT	RW	16	0x2C	0x4843 E12C	0x4844 012C	0x4844 212C
PWMSS_ECAP_ECFLG	R	16	0x2E	0x4843 E12E	0x4844 012E	0x4844 212E
PWMSS_ECAP_ECCLR	RW	16	0x30	0x4843 E130	0x4844 0130	0x4844 2130
PWMSS_ECAP_ECFRC	RW	16	0x32	0x4843 E132	0x4844 0132	0x4844 2132
PWMSS_ECAP_PID	R	32	0x5C	0x4843 E15C	0x4844 015C	0x4844 215C

29.3.20 PWMSS_ECAP Register Description

Table 29-114. PWMSS_ECAP_TSCNT

Address Offset	0x0000 0000		
Physical Address	0x4843 E100 0x4844 0100 0x4844 2100	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	Time Stamp Counter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCNT																															

Bits	Field Name	Description	Type	Reset
31:0	TSCNT	Active 32 bit-counter register that is used as the capture time-base	RW	0x0

Table 29-115. Register Call Summary for Register PWMSS_ECAP_TSCNT

Enhanced Capture (eCAP) Module

- Summary of eCAP Functional Registers: [0]
- PWMSS_ECAP Register Summary: [1]
- PWMSS_ECAP Register Description: [2]

Table 29-116. PWMSS_ECAP_CNTPHS

Address Offset	0x0000 0004		
Physical Address	0x4843 E104 0x4844 0104 0x4844 2104	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	Counter Phase Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTPHS																															

Bits	Field Name	Description	Type	Reset
31:0	CNTPHS	Counter phase value register that can be programmed for phase lag/lead. This register shadows TSCNT and is loaded into PWMSS_ECAP_TSCNT upon either a SYNCI event or S/W force via a control bit. Used to achieve phase control synchronization with respect to other eCAP and EPWM time-bases.	RW	0x0

Table 29-117. Register Call Summary for Register PWMSS_ECAP_CNTPHS

Enhanced Capture (eCAP) Module

- Summary of eCAP Functional Registers: [0]
- PWMSS_ECAP Register Summary: [1]
- PWMSS_ECAP Register Description: [2]

Table 29-118. PWMSS_ECAP_CAP1

Address Offset	0x0000 0008		
Physical Address	0x4843 E108 0x4844 0108 0x4844 2108	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	Capture-1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1																															

Bits	Field Name	Description	Type	Reset
31:0	CAP1	This register can be loaded (written) by the following. (a) Time-Stamp (that is, counter value) during a capture event. (b) Software may be useful for test purposes. (c) APRD active register when used in APWM mode.	RW	0x0

Table 29-119. Register Call Summary for Register PWMSS_ECAP_CAP1

PWM Subsystem Resources

- PWMSS Key Features: [0] [1]

Enhanced Capture (eCAP) Module

- eCAP Functional Description: [2] [3]
- Capture and APWM Operating Mode: [4] [5]
- eCAP Continuous/One-Shot Control: [6] [7] [8] [9]
- CAP1-CAP4 Registers: [10] [11]
- eCAP Shadow Load and Lockout Control: [12] [13]
- eCAP Module APWM Mode Operation: [14] [15] [16]
- Summary of eCAP Functional Registers: [17] [18]
- PWMSS_ECAP Register Summary: [19]
- PWMSS_ECAP Register Description: [20] [21] [22] [23] [24] [25] [26]

Table 29-120. PWMSS_ECAP_CAP2

Address Offset	0x0000 000C		
Physical Address	0x4843 E10C 0x4844 010C 0x4844 210C	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	Capture-2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP2																															

Bits	Field Name	Description	Type	Reset
31:0	CAP2	This register can be loaded (written) by the following. (a) Time- Stamp (that is, counter value) during a capture event. (b) Software may be useful for test purposes. (c) APRD active register when used in APWM mode.	RW	0x0

Table 29-121. Register Call Summary for Register PWMSS_ECAP_CAP2

Enhanced Capture (eCAP) Module

- Capture and APWM Operating Mode: [0] [1]
- CAP1-CAP4 Registers: [2] [3]
- eCAP Shadow Load and Lockout Control: [4] [5]
- eCAP Module APWM Mode Operation: [6]
- Summary of eCAP Functional Registers: [7] [8]
- PWMSS_ECAP Register Summary: [9]
- PWMSS_ECAP Register Description: [10] [11]

Table 29-122. PWMSS_ECAP_CAP3

Address Offset	0x0000 0010		
Physical Address	0x4843 E110 0x4844 0110 0x4844 2110	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	Capture-3 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP3																															

Bits	Field Name	Description	Type	Reset
31:0	CAP3	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow (APRD) register. User SW updates the PWM period value through this register. In this mode, CAP3 shadows CAP1.	RW	0x0

Table 29-123. Register Call Summary for Register PWMSS_ECAP_CAP3

Enhanced Capture (eCAP) Module

- Capture and APWM Operating Mode: [0] [1] [2]
- CAP1-CAP4 Registers: [3]
- eCAP Module APWM Mode Operation: [4] [5] [6]
- Summary of eCAP Functional Registers: [7] [8] [9]
- PWMSS_ECAP Register Summary: [10]

Table 29-124. PWMSS_ECAP_CAP4

Address Offset	0x0000 0014		
Physical Address	0x4843 E114 0x4844 0114 0x4844 2114	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP

Table 29-124. PWMSS_ECAP_CAP4 (continued)

Description	Capture-4 Register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP4																															

Bits	Field Name	Description	Type	Reset
31:0	CAP4	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow (ACMP) register. User SW updates the PWM compare value through this register. In this mode, CAP4 shadows CAP2.	RW	0x0

Table 29-125. Register Call Summary for Register PWMSS_ECAP_CAP4

PWM Subsystem Resources

- PWMSS Key Features: [0] [1]

Enhanced Capture (eCAP) Module

- eCAP Functional Description: [2] [3]
- Capture and APWM Operating Mode: [4] [5] [6]
- eCAP Continuous/One-Shot Control: [7]
- CAP1-CAP4 Registers: [8]
- eCAP Module APWM Mode Operation: [9] [10]
- Summary of eCAP Functional Registers: [11] [12] [13]
- PWMSS_ECAP Register Summary: [14]
- PWMSS_ECAP Register Description: [15] [16] [17] [18] [19]

Table 29-126. PWMSS_ECAP_ECCTL1

Address Offset	0x0000 0028		
Physical Address	0x4843 E128 0x4844 0128 0x4844 2128	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	ECAP Control Register1		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE_SOFT		EVTFLTPS					CAPLDEN	CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL

Bits	Field Name	Description	Type	Reset
15:14	FREE_SOFT	Emulation Control 0x0 = TSCNT counter stops immediately on emulation suspend. 0x1 = TSCNT counter runs until = 0. 0x2 = TSCNT counter is unaffected by emulation suspend (Run Free). 0x3 = TSCNT counter is unaffected by emulation suspend (Run Free).	RW	0x0
13:9	EVTFLTPS	Event Filter prescale select: 0x0 = Divide by 1 (i.e,. no prescale, by-pass the prescaler) 0x1 = Divide by 2 0x2 = Divide by 4 0x3 = Divide by 6 0x4 = Divide by 8 0x5 = Divide by 10 ... 0x1E = Divide by 60 0x1F = Divide by 62	RW	0x0

Bits	Field Name	Description	Type	Reset
8	CAPLDEN	Enable Loading of PWMSS_ECAP_CAP1 to PWMSS_ECAP_CAP4 registers on a capture event 0x0 = Disable PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 register loads at capture event time. 0x1 = Enable PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 register loads at capture event time.	RW	0x0
7	CTRRST4	Counter Reset on Capture Event 4 0x0 = Do not reset counter on Capture Event 4 (absolute time stamp operation) 0x1 = Reset counter after Capture Event 4 time-stamp has been captured (used in difference mode operation)	RW	0x0
6	CAP4POL	Capture Event 4 Polarity select 0x0 = Capture Event 4 triggered on a rising edge (RE) 0x1 = Capture Event 4 triggered on a falling edge (FE)	RW	0x0
5	CTRRST3	Counter Reset on Capture Event 3 0x0 = Do not reset counter on Capture Event 3 (absolute time stamp) 0x1 = Reset counter after Event 3 time-stamp has been captured (used in difference mode operation)	RW	0x0
4	CAP3POL	Capture Event 3 Polarity select 0x0 = Capture Event 3 triggered on a rising edge (RE) 0x1 = Capture Event 3 triggered on a falling edge (FE)	RW	0x0
3	CTRRST2	Counter Reset on Capture Event 2 0x0 = Do not reset counter on Capture Event 2 (absolute time stamp) 0x1 = Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)	RW	0x0
2	CAP2POL	Capture Event 2 Polarity select 0x0 = Capture Event 2 triggered on a rising edge (RE) 0x1 = Capture Event 2 triggered on a falling edge (FE)	RW	0x0
1	CTRRST1	Counter Reset on Capture Event 1 0x0 = Do not reset counter on Capture Event 1 (absolute time stamp) 0x1 = Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)	RW	0x0
0	CAP1POL	Capture Event 1 Polarity select 0x0 = Capture Event 1 triggered on a rising edge (RE) 0x1 = Capture Event 1 triggered on a falling edge (FE)	RW	0x0

Table 29-127. Register Call Summary for Register PWMSS_ECAP_ECCTL1

Enhanced Capture (eCAP) Module

- eCAP Event Prescaler: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Register Summary: [2]

Table 29-128. PWMSS_ECAP_ECCTL2

Address Offset	0x0000 002A		
Physical Address	0x4843 E12A 0x4844 012A 0x4844 212A	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	ECAP Control Register 2		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	APWM POL	CAPAP WM	SWSY NC	SYNCO_SEL	SYNCI _EN	TSCNT STP	REAR MRES ET	STOPVALUE	CONT ONES HT

Bits	Field Name	Description	Type	Reset
15:11	RESERVED		R	0x0
10	APWMPOL	APWM output polarity select. This is applicable only in APWM operating mode 0x0 = Output is active high (Compare value defines high time) 0x1 = Output is active low (Compare value defines low time)	RW	0x0
9	CAPAPWM	CAP/APWM operating mode select **0x0 = ECAP module operates in capture mode. This mode forces the following configuration.** (a) Inhibits TSCNT resets via TSCNT = PRD event. (b) Inhibits shadow loads on PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers. (c) Permits user to enable PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 register load. (d) ECAP input/APWM output pin operates as a capture input. **0x1 = ECAP module operates in APWM mode. This mode forces the following configuration.** (a) Resets TSCNT on TSCNT = PRD event (period boundary). (b) Permits shadow loading on PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers. (c) Disables loading of time-stamps into PWMSS_ECAP_CAP1 - PWMSS_ECAP_CAP4 registers. (d) ECAP input/APWM ouput pin operates as a APWM output.	RW	0x0
8	SWSYNC	Software-forced Counter (TSCNT) Synchronizing. This provides a convenient software method to synchronize some or all ECAP time bases. In APWM mode, the synchronizing can also be done via the TSCNT = PRD event. Note: Selection TSCNT = PRD is meaningful only in APWM mode. However, you can choose it in CAP mode if you find doing so useful. 0x0 = Writing a zero has no effect. Reading always returns a zero 0x1 = Writing a one forces a TSCNT shadow load of current ECAP module and any ECAP modules down-stream providing the SYNCO_SEL bits are 0b00. After writing a 1, this bit returns to a zero.	RW	0x0
7:6	SYNCO_SEL	Sync-Out Select 0x0 = Select sync-in event to be the sync-out signal (pass through) 0x1 = Select TSCNT = PRD event to be the sync-out signal 0x2 = Disable sync out signal 0x3 = Disable sync out signal	RW	0x0

Bits	Field Name	Description	Type	Reset
5	SYNCI_EN	Counter (TSCNT) Sync-In select mode 0x0 = Disable sync-in option 0x1 = Enable counter (TSCNT) to be loaded from PWMSS_ECAP_CNTPHS register upon either a SYNCI signal or a S/W force event.	RW	0x0
4	TSCNTSTP	Time Stamp (TSCNT) Counter Stop (freeze) Control 0x0 = TSCNT stopped 0x1 = TSCNT free-running	RW	0x0
3	REARMRESET	One-Shot Re-Arming Control, that is, wait for stop trigger. Note: The re-arm function is valid in one shot or continuous mode. 0x0 = Has no effect (reading always returns a 0) 0x1 = Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero. 2) Unfreezes the Mod4 counter. 3) Enables capture register loads.	RW	0x0
2:1	STOPVALUE	Stop value for one-shot mode. This is the number (between 1 and 4) of captures allowed to occur before the CAP (1 through 4) registers are frozen, that is, capture sequence is stopped. Wrap value for continuous mode. This is the number (between 1 and 4) of the capture register in which the circular buffer wraps around and starts again. Notes: STOPVALUE is compared to Mod4 counter and, when equal, the following two actions occur. (1) Mod4 counter is stopped (frozen). (2) Capture register loads are inhibited. In one-shot mode, further interrupt events are blocked until re-armed. 0x0 = Stop after Capture Event 1 in one-shot mode. Wrap after Capture Event 1 in continuous mode. 0x1 = Stop after Capture Event 2 in one-shot mode. Wrap after Capture Event 2 in continuous mode. 0x2 = Stop after Capture Event 3 in one-shot mode. Wrap after Capture Event 3 in continuous mode. 0x3 = Stop after Capture Event 4 in one-shot mode. Wrap after Capture Event 4 in continuous mode.	RW	0x3
0	CONTONESHT	Continuous or one-shot mode control (applicable only in capture mode) 0x0 = Operate in continuous mode 0x1 = Operate in one-shot mode	RW	0x0

Table 29-129. Register Call Summary for Register PWMSS_ECAP_ECCTL2

Enhanced Capture (eCAP) Module

- eCAP Shadow Load and Lockout Control: [0] [1]
- Summary of eCAP Functional Registers: [2]
- PWMSS_ECAP Register Summary: [3]

Table 29-130. PWMSS_ECAP_ECEINT

Address Offset	0x0000 002C		
Physical Address	0x4843 E12C 0x4844 012C 0x4844 212C	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	ECAP Interrupt Enable Register		

Table 29-130. PWMSS_ECAP_ECEINT (continued)

Type	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	CMPEQ	Counter Equal **Compare** Interrupt Enable. 0x0 = Disable Compare Equal as an Interrupt source. 0x1 = Enable Compare Equal as an Interrupt source.	RW	0x0
6	PRDEQ	Counter Equal **Period** Interrupt Enable. 0x0 = Disable Period Equal as an Interrupt source. 0x1 = Enable Period Equal as an Interrupt source.	RW	0x0
5	CNTOVF	Counter Overflow Interrupt Enable. 0x0 = Disable counter Overflow as an Interrupt source. 0x1 = Enable counter Overflow as an Interrupt source.	RW	0x0
4	CEVT4	Capture Event 4 Interrupt Enable. 0x0 = Disable Capture Event 4 as an Interrupt source. 0x1 = Enable Capture Event 4 as an Interrupt source.	RW	0x0
3	CEVT3	Capture Event 3 Interrupt Enable. 0x0 = Disable Capture Event 3 as an Interrupt source. 0x1 = Enable Capture Event 3 as an Interrupt source.	RW	0x0
2	CEVT2	Capture Event 2 Interrupt Enable. 0x0 = Disable Capture Event 2 as an Interrupt source. 0x1 = Enable Capture Event 2 as an Interrupt source.	RW	0x0
1	CEVT1	Capture Event 1 Interrupt Enable . 0x0 = Disable Capture Event 1 as an Interrupt source. 0x1 = Enable Capture Event 1 as an Interrupt source.	RW	0x0
0	RESERVED		R	0x0

Table 29-131. Register Call Summary for Register PWMSS_ECAP_ECEINT

Enhanced Capture (eCAP) Module

- eCAP Interrupt Control: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Register Summary: [2]

Table 29-132. PWMSS_ECAP_ECFLG

Address Offset	0x0000 002E		
Physical Address	0x4843 E12E 0x4844 012E 0x4844 212E	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	ECAP Interrupt Flag Register		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	CMPEQ	Compare Equal Compare Status Flag. This flag is only active in APWM mode. 0x0 = Indicates no event occurred 0x1 = Indicates the counter (TSCNT) reached the compare register value (ACMP)	R	0x0
6	PRDEQ	Counter Equal Period Status Flag. This flag is only active in APWM mode. 0x0 = Indicates no event occurred 0x1 = Indicates the counter (TSCNT) reached the period register value (APRD) and was reset.	R	0x0
5	CNTOVF	Counter Overflow Status Flag. This flag is active in CAP and APWM mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the counter (TSCNT) has made the transition from 0xFFFFFFFF to 0x00000000	R	0x0
4	CEVT4	Capture Event 4 Status Flag This flag is only active in CAP mode. 0x0 = Indicates no event occurred 0x1 = Indicates the fourth event occurred at ECAPn pin	R	0x0
3	CEVT3	Capture Event 3 Status Flag. This flag is active only in CAP mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the third event occurred at ECAPn pin.	R	0x0
2	CEVT2	Capture Event 2 Status Flag. This flag is only active in CAP mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the second event occurred at ECAPn pin.	R	0x0
1	CEVT1	Capture Event 1 Status Flag. This flag is only active in CAP mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the first event occurred at ECAPn pin.	R	0x0
0	INT	Global Interrupt Status Flag 0x0 = Indicates no interrupt generated. 0x1 = Indicates that an interrupt was generated.	R	0x0

Table 29-133. Register Call Summary for Register PWMSS_ECAP_ECFLG

Enhanced Capture (eCAP) Module

- eCAP Interrupt Control: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Register Summary: [2]

Table 29-134. PWMSS_ECAP_ECCLR

Address Offset	0x0000 0030		
Physical Address	0x4843 E130 0x4844 0130 0x4844 2130	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	ECAP Interrupt Clear Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	CMPEQ	Counter Equal Compare Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0 0x1 = Writing a 1 clears the TSCNT=CMP flag condition	RW	0x0
6	PRDEQ	Counter Equal Period Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0 0x1 = Writing a 1 clears the TSCNT=PRD flag condition	RW	0x0
5	CNTOVF	Counter Overflow Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0 0x1 = Writing a 1 clears the CNTOVF flag condition	RW	0x0
4	CEVT4	Capture Event 4 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT3 flag condition.	RW	0x0
3	CEVT3	Capture Event 3 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT3 flag condition.	RW	0x0
2	CEVT2	Capture Event 2 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT2 flag condition.	RW	0x0
1	CEVT1	Capture Event 1 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT1 flag condition.	RW	0x0
0	INT	Global Interrupt Clear Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1.	RW	0x0

Table 29-135. Register Call Summary for Register PWMSS_ECAP_ECCLR

Enhanced Capture (eCAP) Module

- eCAP Interrupt Control: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Register Summary: [2]

Table 29-136. PWMSS_ECAP_ECFRC

Address Offset	0x0000 0034		
Physical Address	0x4843 E132 0x4844 0132 0x4844 2132	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	ECAP Interrupt Forcing Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	CMPEQ	Force Counter Equal Compare Interrupt 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the TSCNT=CMP flag bit.	RW	0x0
6	PRDEQ	Force Counter Equal Period Interrupt 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the TSCNT=PRD flag bit.	RW	0x0
5	CNTOVF	Force Counter Overflow 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 to this bit sets the CNTOVF flag bit.	RW	0x0
4	CEVT4	Force Capture Event 4 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the CEVT4 flag bit	RW	0x0
3	CEVT3	Force Capture Event 3 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the CEVT3 flag bit	RW	0x0
2	CEVT2	Force Capture Event 2 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the CEVT2 flag bit.	RW	0x0
1	CEVT1	Always reads back a 0. Force Capture Event 1 0x0 = No effect. 0x1 = Writing a 1 sets the CEVT1 flag bit.	RW	0x0
0	RESERVED		R	0x0

Table 29-137. Register Call Summary for Register PWMSS_ECAP_ECFRC

Enhanced Capture (eCAP) Module

- eCAP Interrupt Control: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Register Summary: [2]

Table 29-138. PWMSS_ECAP_PID

Address Offset	0x0000 005C		
Physical Address	0x4843 E15C 0x4844 015C 0x4844 215C	**Instance**	PWMSS1_ECAP PWMSS2_ECAP PWMSS3_ECAP
Description	ECAP Revision ID		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-(1)

(1) TI Internal information

Table 29-139. Register Call Summary for Register PWMSS_ECAP_PID

Enhanced Capture (eCAP) Module

- Summary of eCAP Functional Registers: [0]
- PWMSS_ECAP Register Summary: [1]

29.4 Enhanced Quadrature Encoder Pulse (eQEP) Module

29.4.1 eQEP Overview

A single track of slots patterns the periphery of an incremental encoder disk, as shown in Figure 29-62. These slots create an alternating pattern of dark and light lines. The disk count is defined as the number of dark/light line pairs that occur per revolution (lines per revolution). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal: QEPI), which can be used to indicate an absolute position. Encoder manufacturers identify the index pulse using different terms such as index, marker, home position, and zero reference.

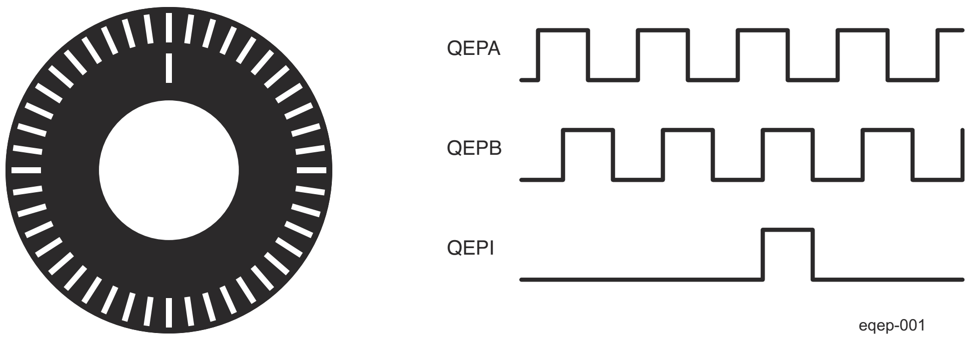


Figure 29-62. Optical Encoder Disk

To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a line pair between them. This shift is realized with a reticle or mask that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90 degrees out of phase from each other. These are commonly called the quadrature QEPA and QEPB signals. The clockwise direction for most encoders is defined as the QEPA channel going positive before the QEPB channel and vise versa as shown in Figure 29-63.

The encoder wheel typically makes one revolution for every revolution of the motor or the wheel may be at a geared rotation ratio with respect to the motor. Therefore, the frequency of the digital signal coming from the QEPA and QEPB outputs varies proportionally with the velocity of the motor. For example, a 2000-line encoder directly coupled to a motor running at 5000 revolutions per minute (rpm) results in a frequency of 166.6 KHz, so by measuring the frequency of either the QEPA or QEPB output, the processor can determine the velocity of the motor.

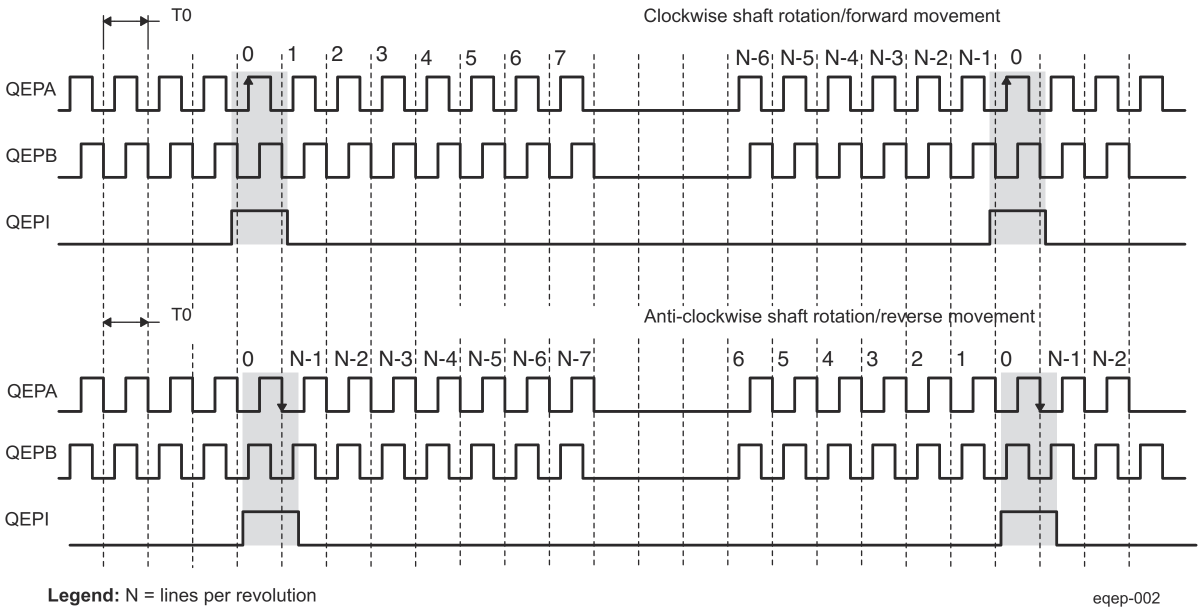


Figure 29-63. QEP Encoder Output Signal for Forward/Reverse Movement

Quadrature encoders from different manufacturers come with two forms of index pulse (gated index pulse or ungated index pulse) as shown in Figure 29-64. A nonstandard form of index pulse is ungated. In the ungated configuration, the index edges are not necessarily coincident with A and B signals. The gated index pulse is aligned to any of the four quadrature edges and width of the index pulse and can be equal to a quarter, half, or full period of the quadrature signal.

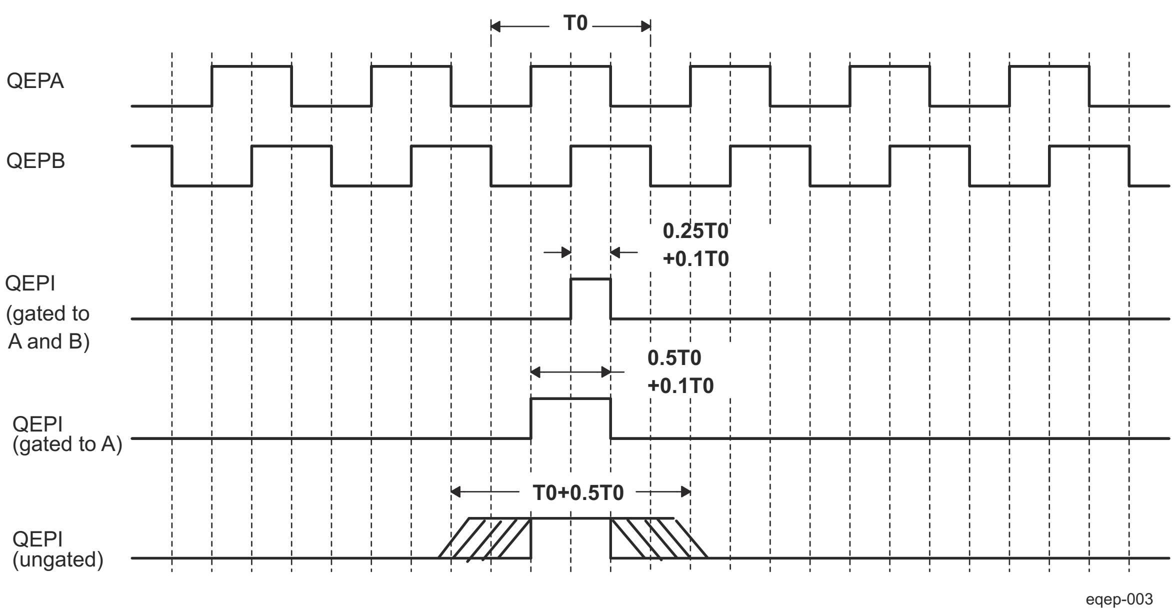


Figure 29-64. Index Pulse Example

Some typical applications of shaft encoders include robotics and even computer input in the form of a mouse. Inside your mouse you can see where the mouse ball spins a pair of axles (a left/right, and an up/down axle). These axles are connected to optical shaft encoders that effectively tell the computer how fast and in what direction the mouse is moving.

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity may be written as:

$$V(k) \approx \frac{X(k) - X(k-1)}{T} = \frac{\Delta X}{T} \tag{8}$$

eqep-004

$$V(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \tag{9}$$

eqep-005

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement.

Equation 8 is the conventional approach to velocity estimation and it requires a time base to provide unit time event for velocity calculation. Unit time is basically the inverse of the velocity calculation rate.

The encoder count (position) is read once during each unit time event. The quantity [x(k) - x(k-1)] is formed by subtracting the previous reading from the current reading. Then the velocity estimate is computed by multiplying by the known constant 1/T (where T is the constant time between unit time events and is known in advance).

Estimation based on Equation 8 has an inherent accuracy limit directly related to the resolution of the position sensor and the unit time period T. For example, consider a 500-line per revolution quadrature encoder with a velocity calculation rate of 400 Hz. When used for position the quadrature encoder gives a four-fold increase in resolution, in this case, 2000 counts per revolution. The minimum rotation that can be detected is therefore 0.0005 revolutions, which gives a velocity resolution of 12 rpm when sampled at 400 Hz. While this resolution may be satisfactory at moderate or high speeds, for example, 1% error at 1200 rpm, it would clearly prove inadequate at low speeds. In fact, at speeds below 12 rpm, the speed estimate would erroneously be zero much of the time.

At low speed, Equation 9 provides a more accurate approach. It requires a position sensor that outputs a fixed interval pulse train, such as the aforementioned quadrature encoder. The width of each pulse is defined by motor speed for a given sensor resolution. Equation 9 can be used to calculate motor speed by measuring the elapsed time between successive quadrature pulse edges. However, this method suffers from the opposite limitation, as does Equation 8. A combination of relatively large motor speeds and high sensor resolution makes the time interval ΔT small, and thus more greatly influenced by the timer resolution. This can introduce considerable error into high-speed estimates.

For systems with a large speed range (that is, speed estimation is needed at both low and high speeds), one approach is to use Equation 9 at low speed and have the software switch over to Equation 8 when the motor speed rises above some specified threshold.

29.4.2 eQEP Module Functional Description

This section provides the eQEP functional description and corrsponding functional details about EQEPx inputs .

Note

Multiple identical eQEP modules can be contained in a system. For actual number of eQEP modules integrated in the device, refer to the Section 29.1.3. The letter x within a signal or module name is used to indicate a generic eQEP instance on a device. For example, output interrupt request, EQEP1A belongs to eQEP1, EQEP2A belongs to eQEP2, etc.

The eQEP peripheral contains the following major functional units (as shown in Figure 29-65):

- Programmable input qualification for each pin (part of the GPIO MUX)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)

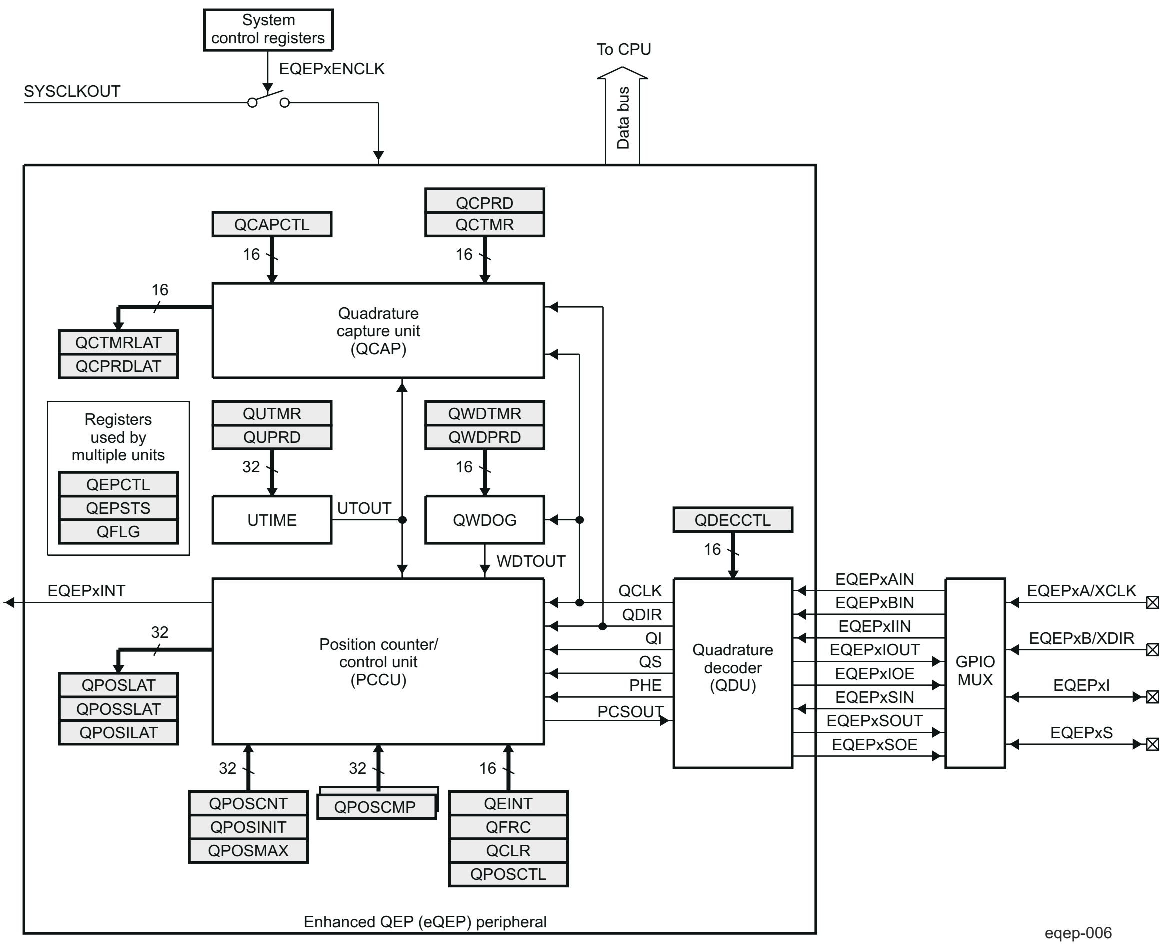


Figure 29-65. Functional Block Diagram of the eQEP Peripheral

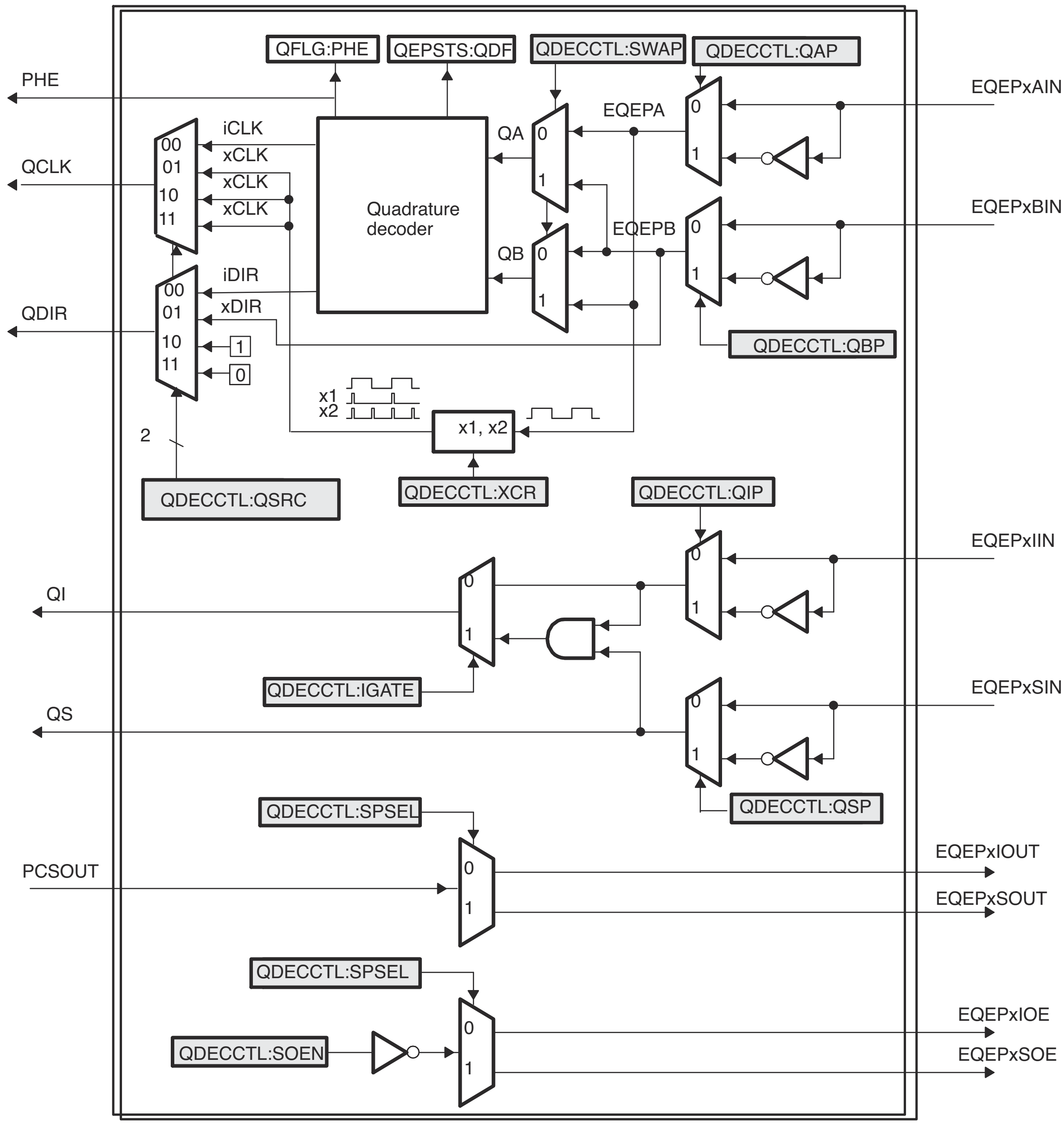
29.4.3 eQEP Inputs

The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input.

- QEPA/XCLK and QEPB/XDIR: These two pins can be used in quadrature-clock mode or direction-count mode.
  - Quadrature-clock Mode: The eQEP encoders provide two square wave signals (A and B) 90 electrical degrees out of phase whose phase relationship is used to determine the direction of rotation of the input shaft and number of eQEP pulses from the index position to derive the relative position information. For forward or clockwise rotation, QEPA signal leads QEPB signal and vice versa. The quadrature decoder uses these two inputs to generate quadrature-clock and direction signals.
  - Direction-count Mode: In direction-count mode, direction and clock signals are provided directly from the external source. Some position encoders have this type of output instead of quadrature output. The QEPA pin provides the clock input and the QEPB pin provides the direction input.
- QEPI: Index or Zero Marker: The eQEP encoder uses an index signal to assign an absolute start position from which position information is incrementally encoded using quadrature pulses. This pin is connected to the index output of the eQEP encoder to optionally reset the position counter for each revolution. This signal can be used to initialize or latch the position counter on the occurrence of a desired event on the index pin.
- QEPS: Strobe Input: This general-purpose strobe signal can initialize or latch the position counter on the occurrence of a desired event on the strobe pin. This signal is typically connected to a sensor or limit switch to notify that the motor has reached a defined position.

29.4.4 eQEP Quadrature Decoder Unit (QDU)

Figure 29-66 shows a functional block diagram of the QDU.



eqep-007

Figure 29-66. Functional Block Diagram of Decoder Unit

29.4.5 eQEP Position Counter Input Modes

Clock and direction input to position counter is selected using the QSRC bit in the eQEP decoder control register (EQEP_QDECCTL), based on interface input requirement as follows:

- Quadrature-count mode
- Direction-count mode
- UP-count mode
- DOWN-count mode

29.4.6 Quadrature Count Mode

The quadrature decoder generates the direction and clock to the position counter in quadrature count mode.

Direction Decoding The direction decoding logic of the eQEP circuit determines which one of the sequences (QEPA, QEPB) is the leading sequence and accordingly updates the direction information in the QDF bit in the eQEP status register (EQEP_QEPSTS). Table 29-140 and Figure 29-67 show the direction decoding logic in truth table and state machine form. Both edges of the QEPA and QEPB signals are sensed to generate count pulses for the position counter. Therefore, the frequency of the clock generated by the eQEP logic is four times that of each input sequence. Figure 29-68 shows the direction decoding and clock generation from the eQEP input signals.

Phase Error Flag In normal operating conditions, quadrature inputs QEPA and QEPB will be 90 degrees out of phase. The phase error flag (PHE) is set in the EQEP_QFLG register when edge transition is detected simultaneously on the QEPA and QEPB signals to optionally generate interrupts. State transitions marked by dashed lines in Figure 29-67 are invalid transitions that generate a phase error.

Count Multiplication The eQEP position counter provides 4x times the resolution of an input clock by generating a quadrature-clock (QCLK) on the rising/falling edges of both eQEP input clocks (QEPA and QEPB) as shown in Figure 29-68.

Reverse Count In normal quadrature count operation, QEPA input is fed to the QA input of the quadrature decoder and the QEPB input is fed to the QB input of the quadrature decoder. Reverse counting is enabled by setting the SWAP bit in the eQEP decoder control register (EQEP_QDECCTL). This will swap the input to the quadrature decoder thereby reversing the counting direction.

Table 29-140. Quadrature Decoder Truth Table

Previous Edge	Present Edge	QDIR	QPOSCNT
QA↑	QB↑	UP	Increment
	QB↓	DOWN	Decrement
	QA↓	TOGGLE	Increment or Decrement
QA↓	QB↓	UP	Increment
	QB↑	DOWN	Decrement
	QA↑	TOGGLE	Increment or Decrement
QB↑	QA↑	DOWN	Increment
	QA↓	UP	Decrement
	QB↓	TOGGLE	Increment or Decrement
QB↓	QA↓	DOWN	Increment
	QA↑	UP	Decrement
	QB↑	TOGGLE	Increment or Decrement

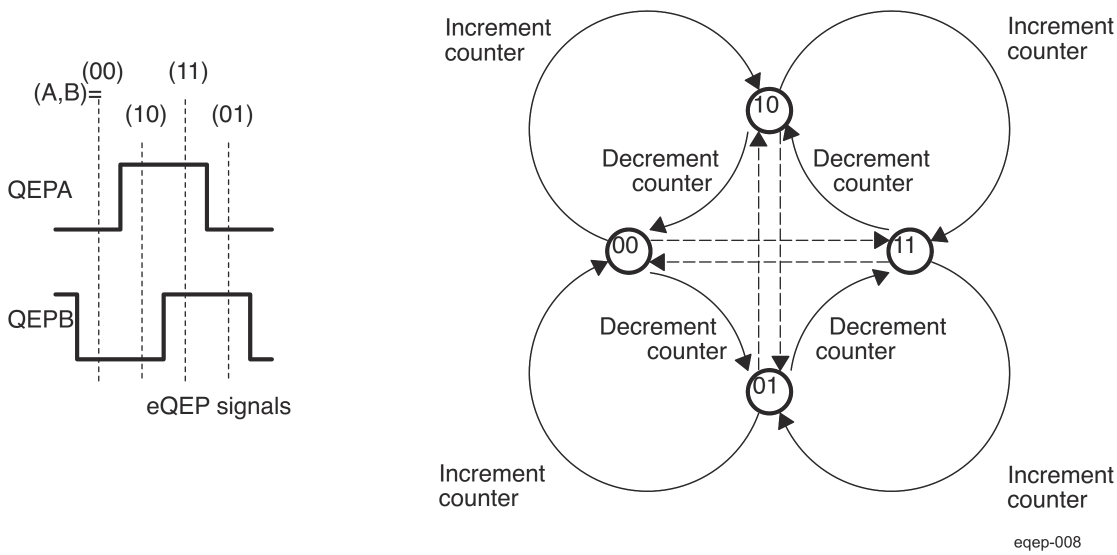


Figure 29-67. Quadrature Decoder State Machine

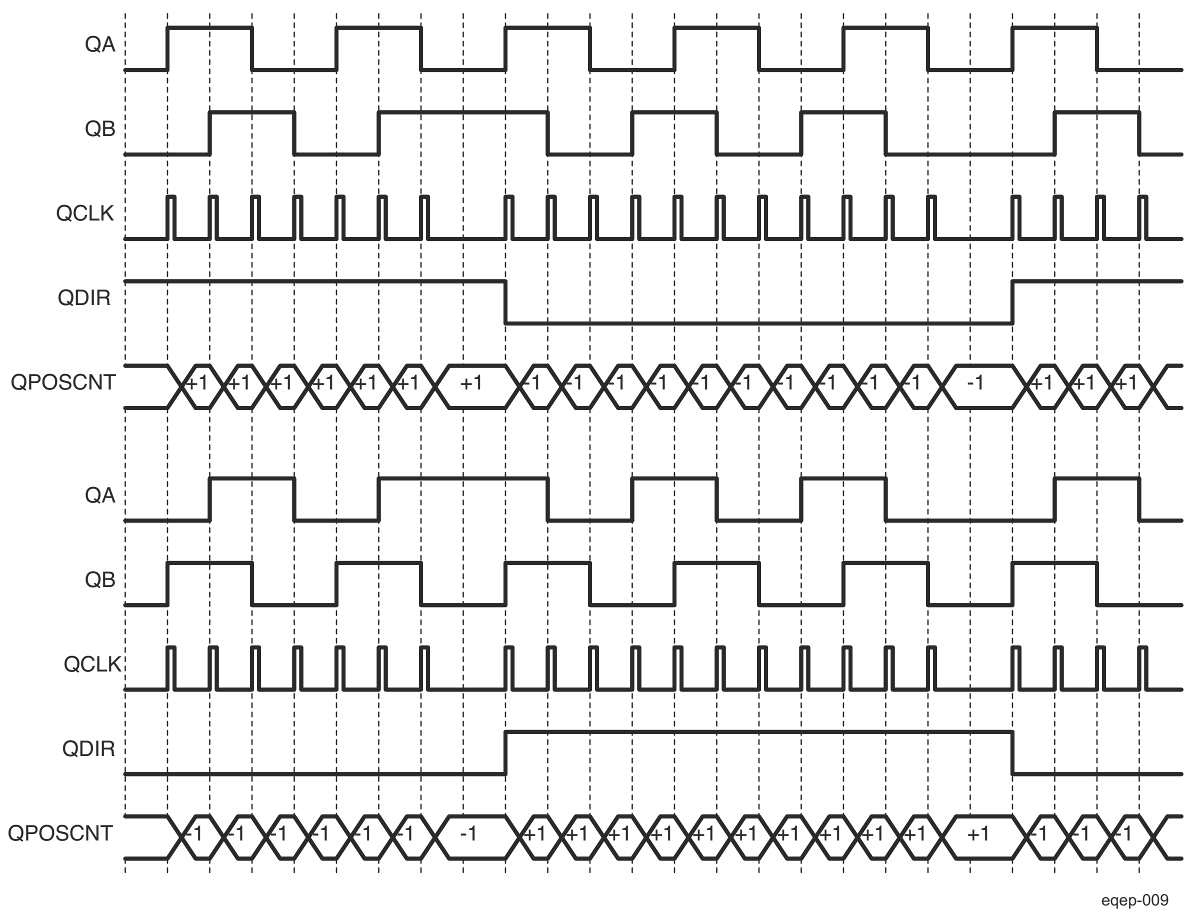


Figure 29-68. Quadrature-clock and Direction Decoding

29.4.7 eQEP Direction-count Mode

Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. QEPA input will provide the clock for position counter and the QEPB input will have the direction information. The position counter is incremented on every rising edge of a QEPA input when the direction input is high and decremented when the direction input is low.

29.4.8 eQEP Up-Count Mode

The counter direction signal is hard-wired for up count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (EQEP_QDECCTL) enables clock generation to the position counter on both edges of the QEPA input, thereby increasing the measurement resolution by 2× factor.

29.4.9 eQEP Down-Count Mode

The counter direction signal is hardwired for a down count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (EQEP_QDECCTL) enables clock generation to the position counter on both edges of a QEPA input, thereby increasing the measurement resolution by 2× factor.

29.4.10 eQEP Input Polarity Selection

Each eQEP input can be inverted using the in the eQEP decoder control register (EQEP_QDECCTL[8:5]) control bits. As an example, setting of the QIP bit in EQEP_QDECCTL inverts the index input.

29.4.11 eQEP Position-Compare Sync Output

The eQEP peripheral includes a position-compare unit that is used to generate the position-compare sync signal on compare match between the position counter register (EQEP_QPOSCNT) and the position-compare register (EQEP_QPOSCMP). This sync signal can be output using an index pin or strobe pin of the EQEP peripheral.

Setting the SOEN bit in the eQEP decoder control register (EQEP_QDECCTL) enables the position-compare sync output and the SPSEL bit in EQEP_QDECCTL selects either an eQEP index pin or an eQEP strobe pin.

29.4.12 eQEP Position Counter and Control Unit (PCCU)

The position counter and control unit provides two configuration registers (EQEP_QEPCTL and EQEP_QPOSCTL) for setting up position counter operational modes, position counter initialization/latch modes and position-compare logic for sync signal generation.

29.4.13 eQEP Position Counter Operating Modes

Position counter data may be captured in different manners. In some systems, the position counter is accumulated continuously for multiple revolutions and the position counter value provides the position information with respect to the known reference. An example of this is the quadrature encoder mounted on the motor controlling the print head in the printer. Here the position counter is reset by moving the print head to the home position and then position counter provides absolute position information with respect to home position.

In other systems, the position counter is reset on every revolution using index pulse and position counter provides rotor angle with respect to index pulse position.

Position counter can be configured to operate in following four modes

- Position Counter Reset on Index Event
- Position Counter Reset on Maximum Position
- Position Counter Reset on the first Index Event
- Position Counter Reset on Unit Time Out Event (Frequency Measurement)

In all the above operating modes, position counter is reset to 0 on overflow and to QPOSMAX bifield value in EQEP_QPOSMAX register on underflow. Overflow occurs when the position counter counts up after QPOSMAX value. Underflow occurs when position counter counts down after "0". Interrupt flag is set to indicate overflow/ underflow in EQEP_QFLG register.

29.4.14 eQEP Position Counter Reset on Index Event (EQEP_QEPCTL[31:12] PCRM] = 0b00)

If the index event occurs during the forward movement, then position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the EQEP_QPOSMAX register on the next eQEP clock.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in EQEP_QEPSTS registers, it also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for index event reset operation.

For example, if the first reset operation occurs on the falling edge of QEPB during the forward direction, then all the subsequent reset must be aligned with the falling edge of QEPB for the forward rotation and on the rising edge of QEPB for the reverse rotation as shown in Figure 29-69.

The position-counter value is latched to the EQEP_QPOSILAT register and direction information is recorded in the EQEP_QEPSTS[QDLF] bit on every index event marker. The position-counter error flag (EQEP_QEPSTS[PCEF]) and error interrupt flag (EQEP_QFLG[PCE]) are set if the latched value is not equal to 0 or QPOSMAX. The position-counter error flag (EQEP_QEPSTS[PCEF]) is updated on every index event marker and an interrupt flag (EQEP_QFLG[PCE]) will be set on error that can be cleared only through software.

The index event latch configuration EQEP_QEPCTL[5:4] IEL bits are ignored in this mode and position counter error flag/interrupt flag are generated only in index event reset mode.

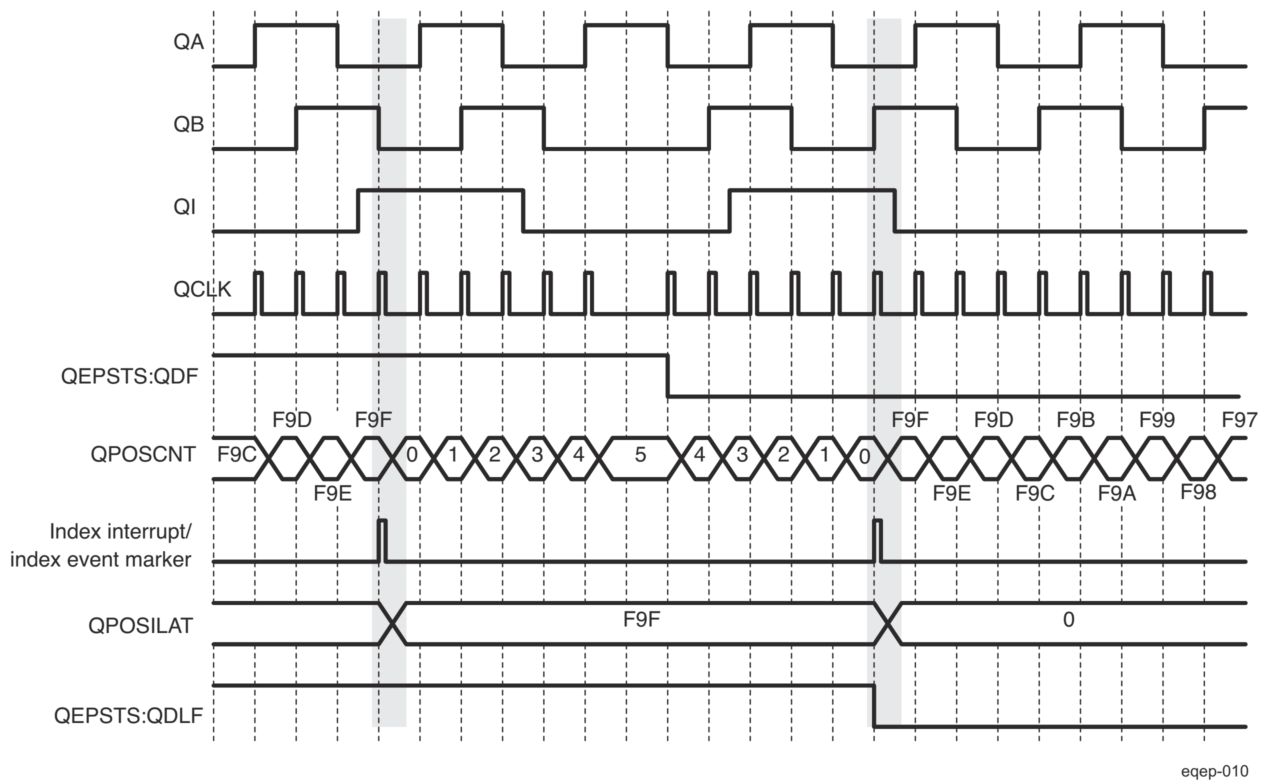


Figure 29-69. Position Counter Reset by Index Pulse for 1000 Line Encoder (QPOSMAX = 3999 or F9Fh)

29.4.15 eQEP Position Counter Reset on Maximum Position (EQEP_QEPCTL[13:12] PCRM=0b01)

If the position counter is equal to QPOSMAX (in EQEP_QPOSMAX register), then the position counter is reset to 0 on the next eQEP clock for forward movement and position counter overflow flag is set. If the position counter is equal to ZERO, then the position counter is reset to QPOSMAX on the next QEP clock for reverse movement and position counter underflow flag is set. Figure 29-70 shows the position counter reset operation in this mode.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in the EQEP_QEPSTS registers; it also remembers the quadrature edge on the first index marker so that the same relative quadrature transition is used for the software index marker (EQEP_QEPCTL[5:4] IEL=0b11).

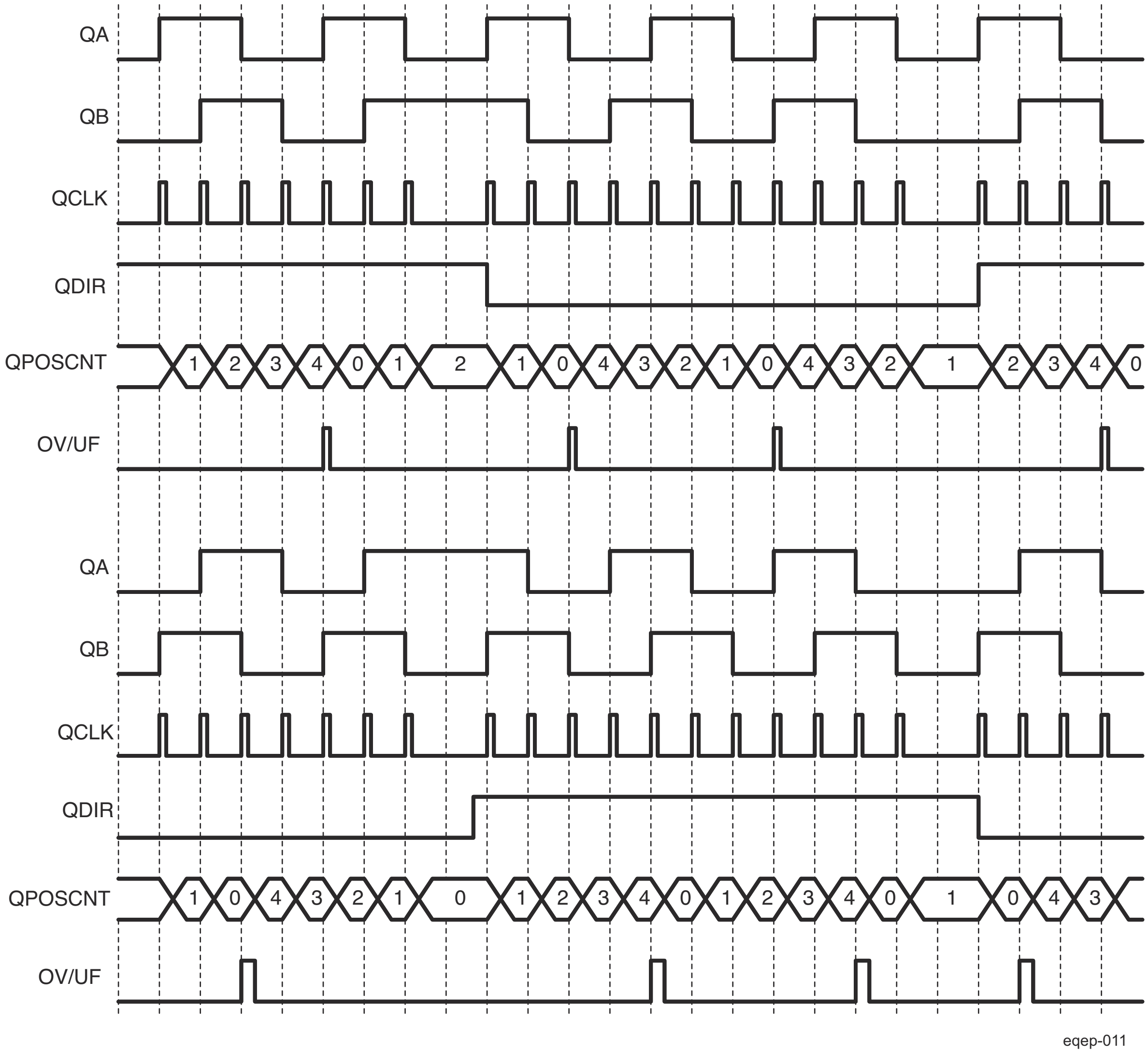


Figure 29-70. Position Counter Underflow/Overflow (QPOSMAX = 4)

29.4.16 Position Counter Reset on the First Index Event (EQEP_QEPCTL[13:12] PCRM = 0b10)

If the index event occurs during forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the EQEP_QPOSMAX register on the next eQEP clock. Note that this is done only on the first occurrence and subsequently the position counter value is not reset on an index event; rather, it is reset based on maximum position as described in Section 29.4.15 .

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in EQEP_QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for software index marker (EQEP_QEPCTL[5:4] IEL= 0b11).

29.4.17 Position Counter Reset on Unit Time out Event (EQEP_QEPCTL[13:12] PCRM = 0b11)

In this mode, the QPOSCNT value is latched to the EQEP_QPOSLAT register and then the QPOSCNT field is reset (to 0 or the QPOSMAX value in the EQEP_QPOSMAX register, depending on the direction mode selected by EQEP_QDECCTL[QSRC] bits on a unit time event). This is useful for frequency measurement.

29.4.18 eQEP Position Counter Latch

The eQEP index and strobe input can be configured to latch the position counter QPOSCNT (EQEP_QPOSCNT) into QPOSILAT (EQEP_QPOSILAT register) and QPOSSLAT (EQEP_QPOSSLAT register) bitfields, respectively, on occurrence of a definite event on these pins.

29.4.19 Index Event Latch

In some applications, it may not be desirable to reset the position counter on every index event and instead it may be required to operate the position counter in full 32-bit mode (EQEP_QEPCTL[13:12] PCRM = 0b01 and EQEP_QEPCTL[13:12] PCRM = 0b10 modes).

In such cases, the eQEP position counter can be configured to latch on the following events and direction information is recorded in the EQEP_QEPSTS[QDLF] bit on every index event marker.

- Latch on Rising edge (EQEP_QEPCTL[5:4] IEL = 0b01)
- Latch on Falling edge (EQEP_QEPCTL[5:4] IEL = 0b10)
- Latch on Index Event Marker (EQEP_QEPCTL[5:4] IEL = 0b11)

This is particularly useful as an error checking mechanism to check if the position counter accumulated the correct number of counts between index events. As an example, the 1000-line encoder must count 4000 times when moving in the same direction between the index events.

The index event latch interrupt flag (EQEP_QFLG[IEL]) is set when the position counter is latched to the EQEP_QPOSILAT register. The index event latch configuration bits (QEPCTZ[IEL]) are ignored when EQEP_QEPCTL[13:12] PCRM = 0b00.

Latch on Rising Edge (EQEP_QEPCTL[5:4] IEL = 0b01)
The position counter value (QPOSCNT) is latched to the EQEP_QPOSILAT register on every rising edge of an index input.

Latch on Falling Edge (EQEP_QEPCTL[5:4] IEL = 0b10)
The position counter value (QPOSCNT) is latched to the EQEP_QPOSILAT register on every falling edge of index input.

Latch on Index Event Marker/Software Index Marker (EQEP_QEPCTL[5:4] IEL = 0b11)
The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in the EQEP_QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same

relative quadrature transition is used for latching the position counter (EQEP_QEPCTL[5:4] IEL = 0b11).

Figure 29-71 shows the position counter latch using an index event marker.

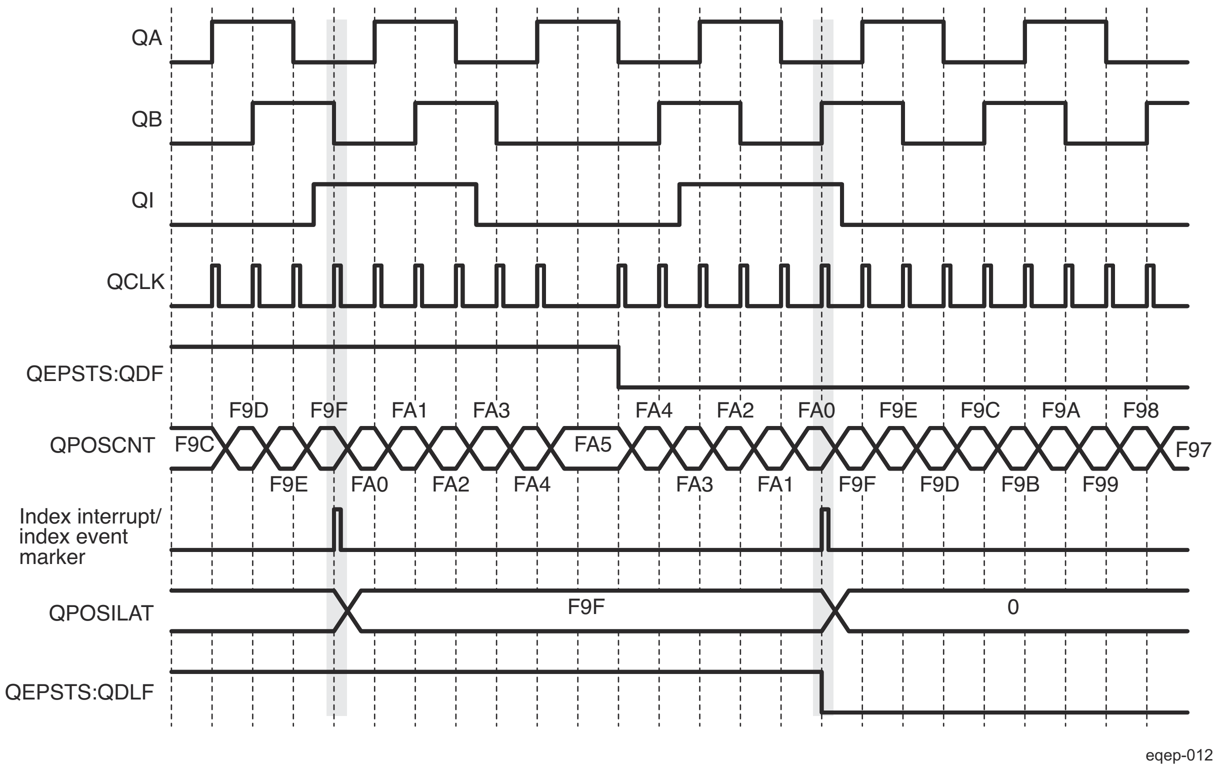


Figure 29-71. Software Index Marker for 1000-line Encoder (EQEP_QEPCTL[5:4] IEL = 0b01)

29.4.20 eQEP Strobe Event Latch

The position-counter value is latched to the EQEP_QPOSSLAT register on the rising edge of the strobe input by clearing the EQEP_QEPCTL[6] SEL bit.

If the EQEP_QEPCTL[6] SEL bit is set, then the position counter value is latched to the EQEP_QPOSSLAT register on the rising edge of the strobe input for forward direction and on the falling edge of the strobe input for reverse direction as shown in Figure 29-72.

The strobe event latch interrupt flag (EQEP_QFLG[SEL]) is set when the position counter is latched to the EQEP_QPOSSLAT register.

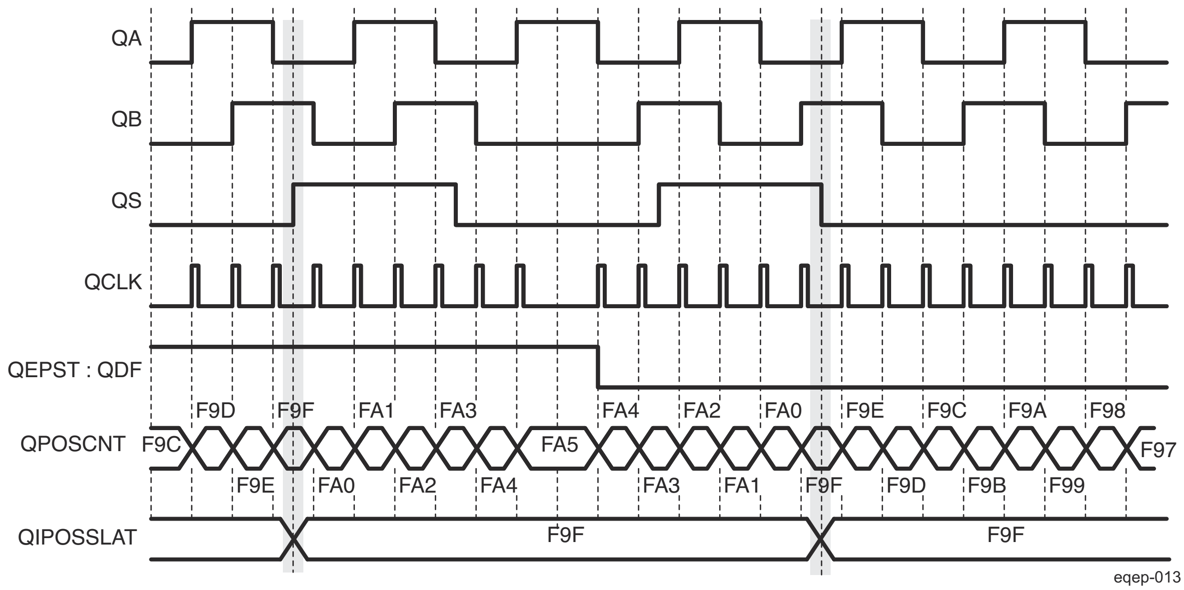


Figure 29-72. eQEP Strobe Event Latch (EQEP_QEPCTL[6] SEL = 0b1)

29.4.21 eQEP Position Counter Initialization

The position counter can be initialized using following events:

- Index event
- Strobe event
- Software initialization

Index Event Initialization (IEI)

The QEPI index input can be used to trigger the initialization of the position counter at the rising or falling edge of the index input.

If the EQEP_QEPCTL[9:8] IEI bits are 0b10, then the position counter (EQEP_QPOSCNT) is initialized with a value in the EQEP_QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

The index event initialization interrupt flag (EQEP_QFLG[IEI]) is set when the position counter is initialized with a value in the EQEP_QPOSINIT register.

Strobe Event Initialization (SEI)

If the EQEP_QEPCTL[11:10] SEI bits are 0b10, then the position counter is initialized with a value in the EQEP_QPOSINIT register on the rising edge of strobe input.

If the EQEP_QEPCTL[11:10] SEI bits are 0b11, then the position counter (EQEP_QPOSCNT) is initialized with a value in the EQEP_QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

The strobe event initialization interrupt flag (EQEP_QFLG[SEI]) is set when the position counter is initialized with a value in the EQEP_QPOSINIT register.

Software Initialization (SWI)

The position counter can be initialized in software by writing a '1' to the EQEP_QEPCTL[7] SWI bit, which will automatically be cleared after initialization.

29.4.22 eQEP Position-Compare Unit

The eQEP peripheral includes a position-compare unit that is used to generate a sync output and/or interrupt on a position-compare match. Figure 29-73 shows a diagram. The position-compare (EQEP_QPOSCMP) register is shadowed and shadow mode can be enabled or disabled using the EQEP_QPOSCTL[PSSHDW] bit. If the shadow mode is not enabled, the CPU writes directly to the active position compare register.

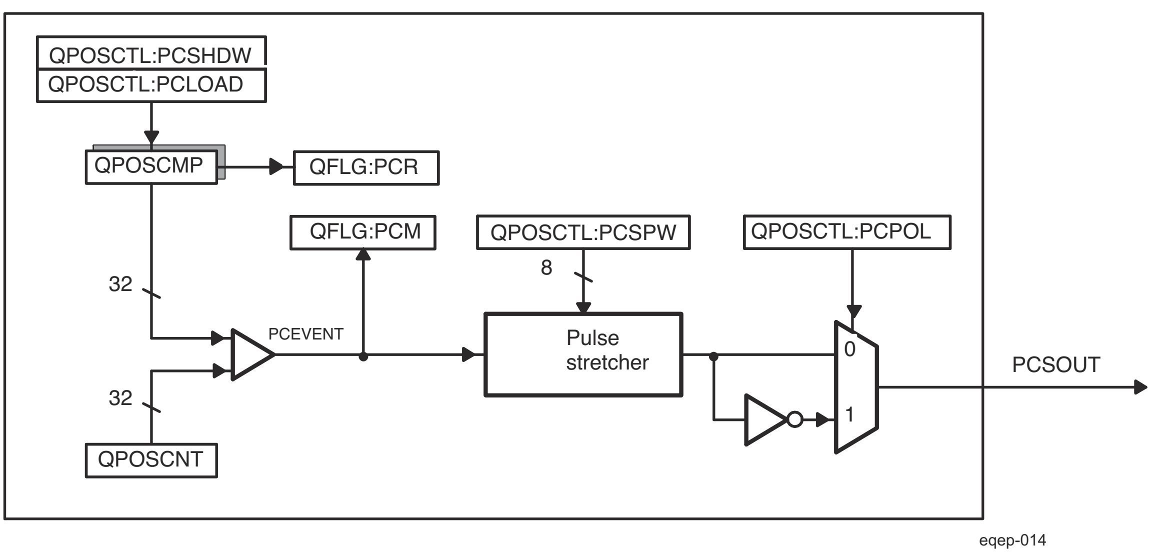


Figure 29-73. eQEP Position-compare Unit

In shadow mode, you can configure the position-compare unit (EQEP_QPOSCTL[PCLOAD]) to load the shadow register value into the active register on the following events and to generate the position-compare ready (EQEP_QFLG[PCR]) interrupt after loading.

- Load on compare match
- Load on position-counter zero event

The position-compare match (EQEP_QFLG[PCM]) is set when the position-counter value (QPOSCNT) matches with the active position-compare register (EQEP_QPOSCMP) and the position-compare sync output of the programmable pulse width is generated on compare match to trigger an external device.

For example, if EQEP_QPOSCMP bitfield QPOSCMP = 0x2, the position-compare unit generates a position-compare event on 1 to 2 transitions of the eQEP position counter for forward counting direction and on 3 to 2 transitions of the eQEP position counter for reverse counting direction (see Figure 29-74).

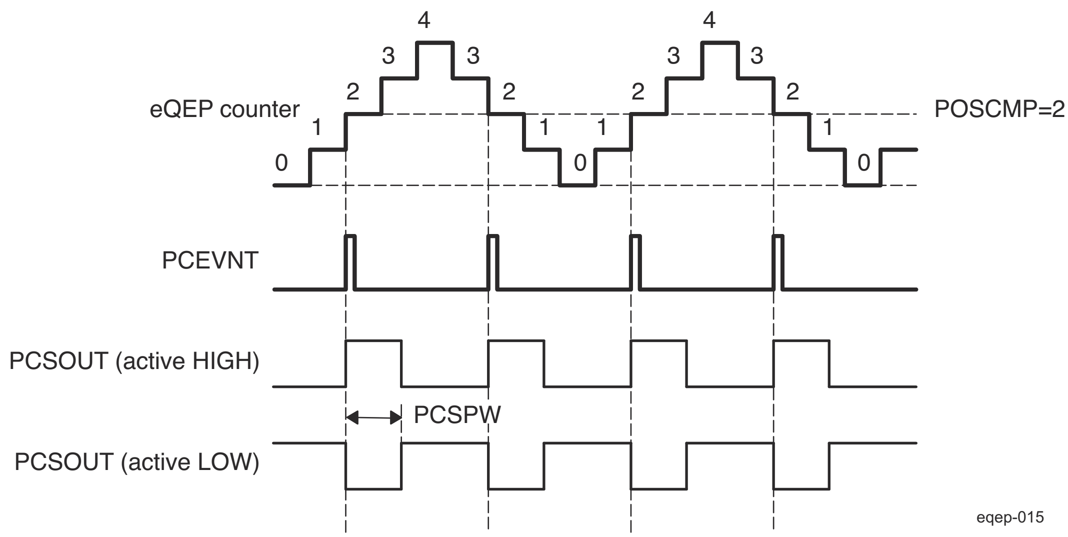


Figure 29-74. eQEP Position-compare Event Generation Points

The pulse stretcher logic in the position-compare unit generates a programmable position-compare sync pulse output on the position-compare match. In the event of a new position-compare match while a previous position-compare pulse is still active, then the pulse stretcher generates a pulse of specified duration from the new position-compare event as shown in Figure 29-75.

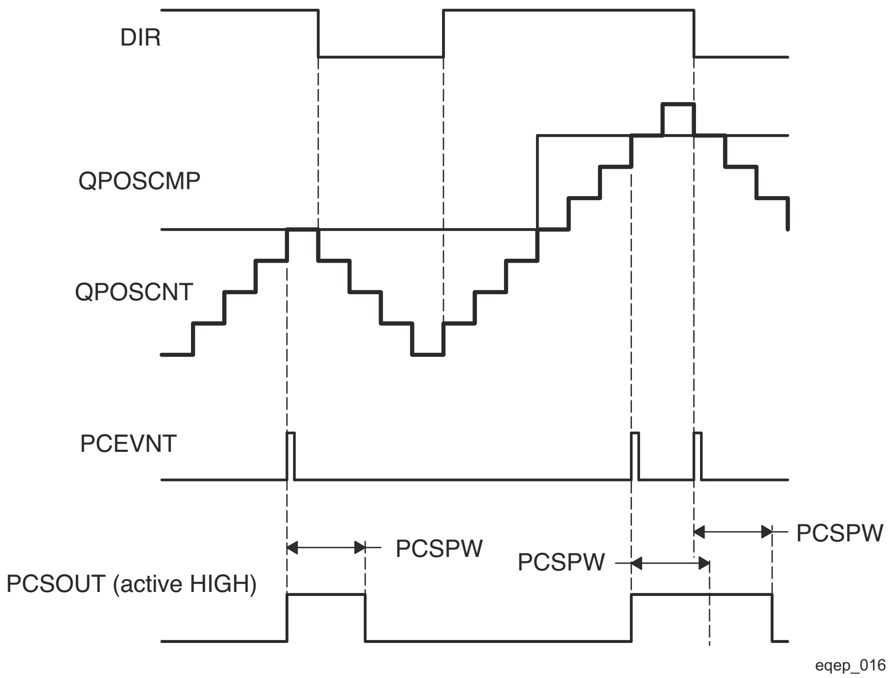


Figure 29-75. eQEP Position-compare Sync Output Pulse Stretcher

29.4.23 eQEP Edge Capture Unit

The eQEP peripheral includes an integrated edge capture unit to measure the elapsed time between the unit position events as shown in Figure 29-76. This feature is typically used for low speed measurement using the following equation:

$$V(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (10)$$

eqep-017

where,

- X - Unit position is defined by integer multiple of quadrature edges (see Figure 29-77)
- ΔT - Elapsed time between unit position events
- v(k) - Velocity at time instant "k"

The eQEP capture timer (QCTMR bitfield in EQEP_QCTMR register) runs from prescaled SYSCLKOUT and the prescaler is programmed by the EQEP_QCAPCTL[CCPS] bits. The capture timer QCTMR value is latched into the capture period register (EQEP_QCPRD) on every unit position event and then the capture timer is reset, a flag is set in EQEP_QEPSTS[UPEVNT] to indicate that new value is latched into the EQEP_QCPRD register. Software can check this status flag before reading the period register for low speed measurement and clear the flag by writing 1.

Note

The system clock - SYSCLKOUT is the eQEP functional clock derived from the PWMSSn gateable interface and functional clock PWMSSn_GICLK, described in Section 29.1.3.

Time measurement (ΔT) between unit position events will be correct if the following conditions are met:

- No more than 65,535 counts have occurred between unit position events.
- No direction change between unit position events.

The capture unit sets the eQEP overflow error flag (EQEP_QEPSTS[COEF]) in the event of capture timer overflow between unit position events. If a direction change occurs between the unit position events, then an error flag is set in the status register (EQEP_QEPSTS[CDEF]).

Capture Timer (EQEP_QCTMR register) and Capture period register (EQEP_QCPRD) can be configured to latch on following events.

- CPU read of EQEP_QPOSCNT register
- Unit time-out event

If the EQEP_QEPCTL[2] QCLM bit is cleared, then the capture timer and capture period values are latched into the EQEP_QCTMRLAT and EQEP_QCPRDLAT registers, respectively, when the CPU reads the position counter in EQEP_QPOSCNT.

If the EQEP_QEPCTL[2] QCLM bit is set, then the position counter, capture timer, and capture period values are latched into the EQEP_QPOSLAT, EQEP_QCTMRLAT and EQEP_QCPRDLAT registers, respectively, on unit time out.

Figure 29-78 shows the capture unit operation along with the position counter.

Note

The EQEP_QCAPCTL register should not be modified dynamically (such as switching CAPCLK prescaling mode from QCLK/4 to QCLK/8). The capture unit must be disabled before changing the prescaler.

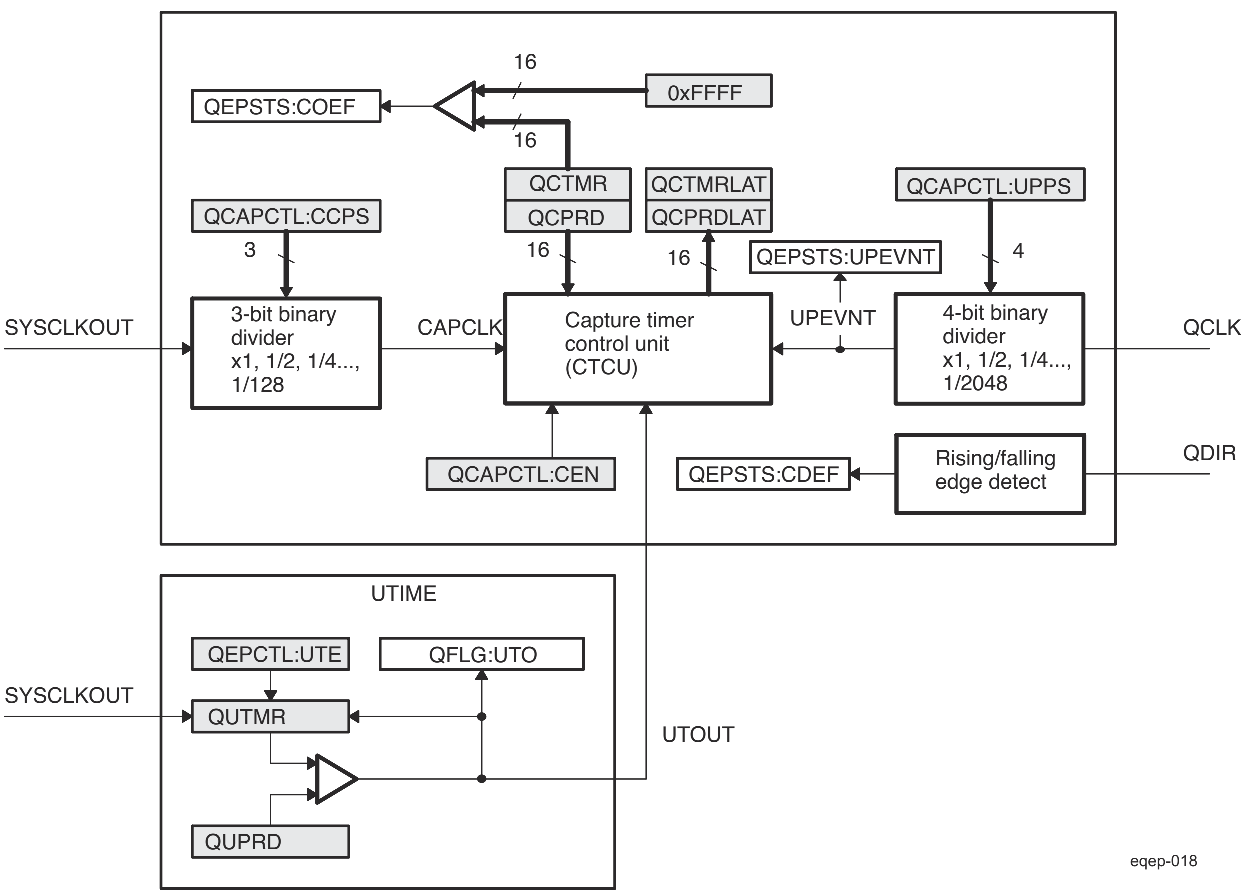
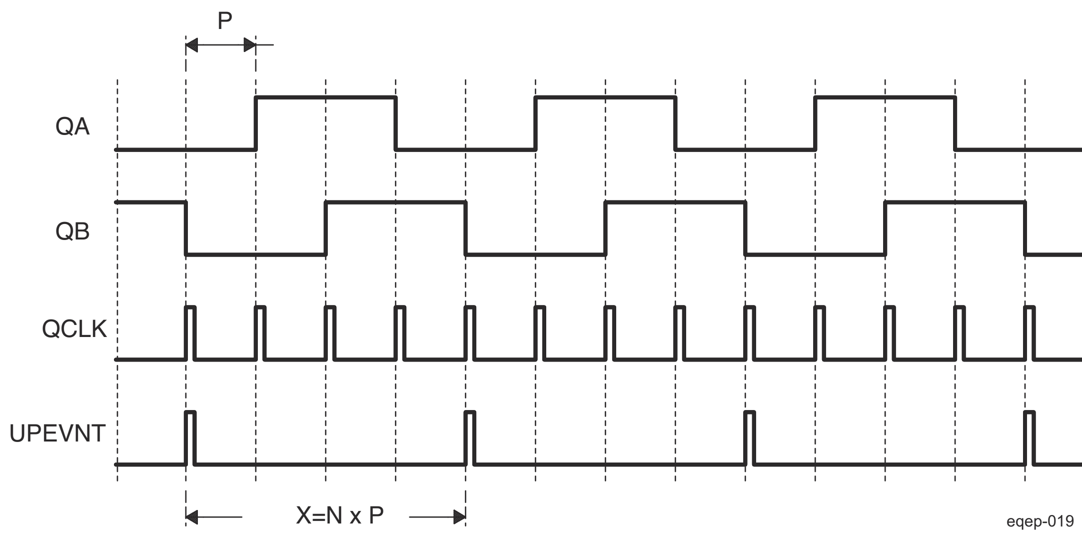


Figure 29-76. eQEP Edge Capture Unit



N - Number of quadrature periods selected using EQEP_QCAPCTL[UPPS] bits

Figure 29-77. Unit Position Event for Low Speed Measurement (EQEP_QCAPCTL[UPPS] = 0010)

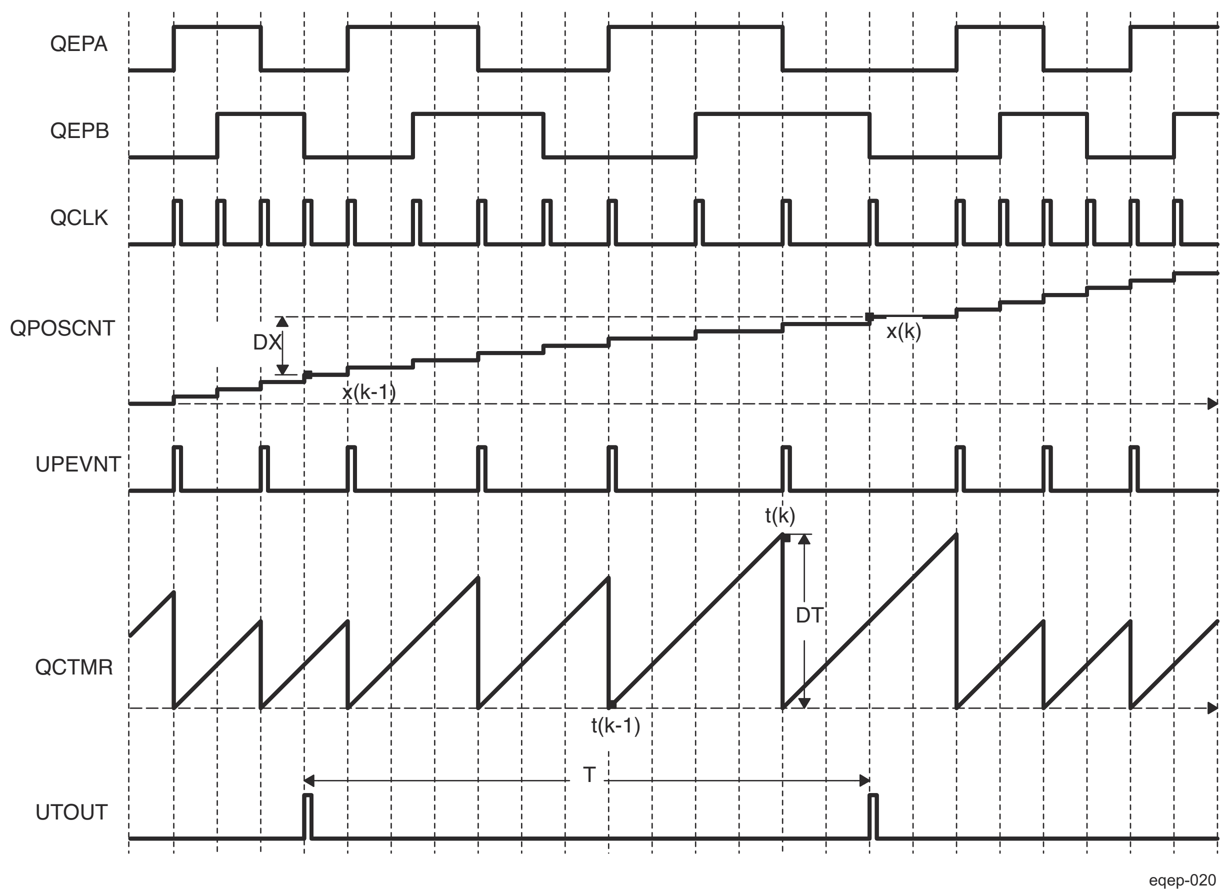


Figure 29-78. eQEP Edge Capture Unit - Timing Details

Velocity Calculation Equations:

$$V(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad o \tag{11}$$

eqep_021

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

Unit time (T) and unit period (X) are configured using the EQEP_QUPRD and EQEP_QCAPCTL[UPPS] registers. Incremental position output and incremental time output is available in the EQEP_QPOSLAT and EQEP_QCPRDLAT registers.

Parameter	Relevant Register to Configure or Read the Information
T	Unit Period Register (EQEP_QUPRD)
ΔX	Incremental Position = QPOSLAT(k) - QPOSLAT(K - 1)
X	Fixed unit position defined by sensor resolution and ZCAPCTL[UPPS] bits
ΔT	Capture Period Latch (QCPRDLAT)

29.4.24 eQEP Watchdog

The eQEP peripheral contains a 16-bit watchdog timer that monitors the quadrature-clock to indicate proper operation of the motion-control system. The eQEP watchdog timer is clocked from SYSCLKOUT/64 and the quadrate clock event (pulse) resets the watchdog timer. If no quadrature-clock event is detected until a period match (QWDPRD = QWDTMR), then the watchdog timer will time out and the watchdog interrupt flag will be set (EQEP_QFLG[WTO]). The time-out value is programmable through the watchdog period register (EQEP_QWDPRD).

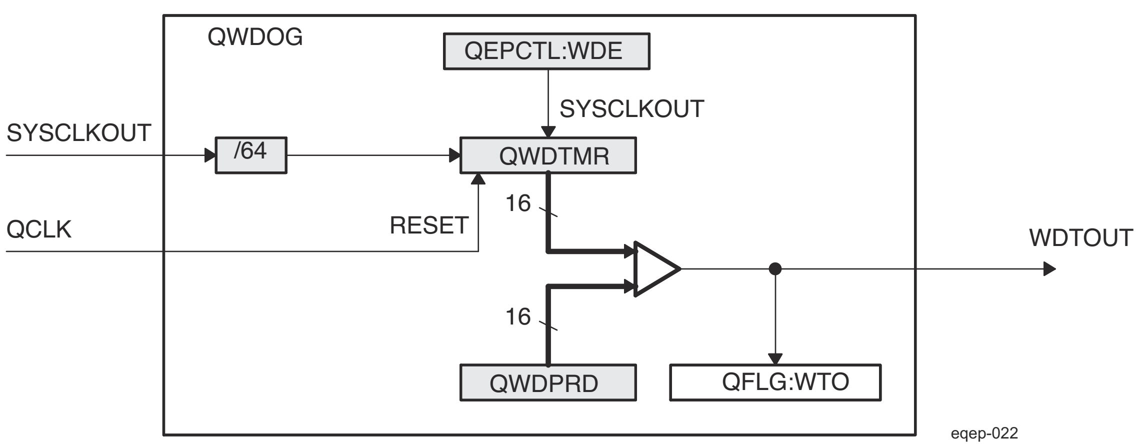


Figure 29-79. eQEP Watchdog Timer

29.4.25 Unit Timer Base

The eQEP peripheral includes a 32-bit timer (QUTMR) that is clocked by SYSCLKOUT to generate periodic interrupts for velocity calculations. The unit time out interrupt is set (EQEP_QFLG[UTO]) when the unit timer (QUTMR) matches the unit period register (EQEP_QUPRD).

The eQEP peripheral can be configured to latch the position counter, capture timer, and capture period values on a unit time out event so that latched values are used for velocity calculation as described in Section 29.4.23 .

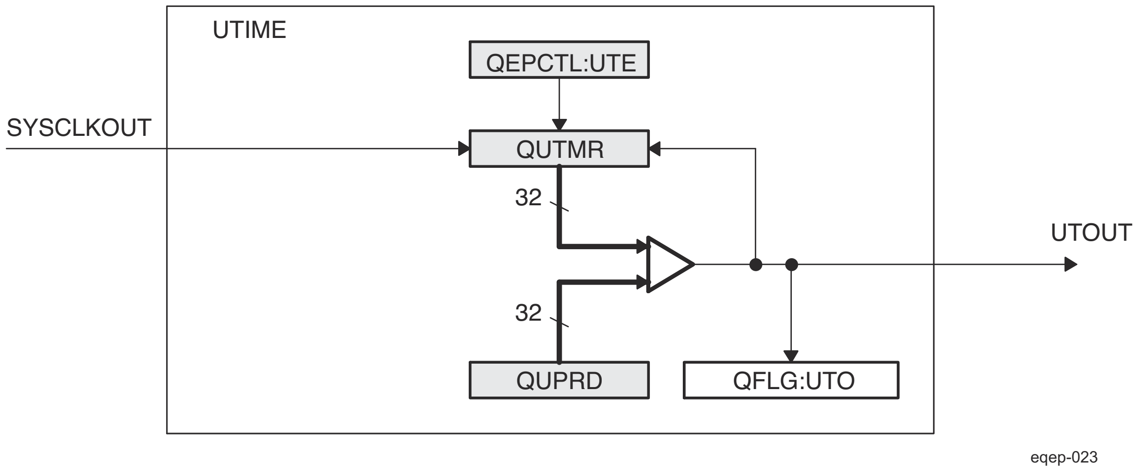


Figure 29-80. eQEP Unit Time Base

29.4.26 eQEP Interrupt Structure

Figure 29-81 shows how the interrupt mechanism works in the EQEP module.

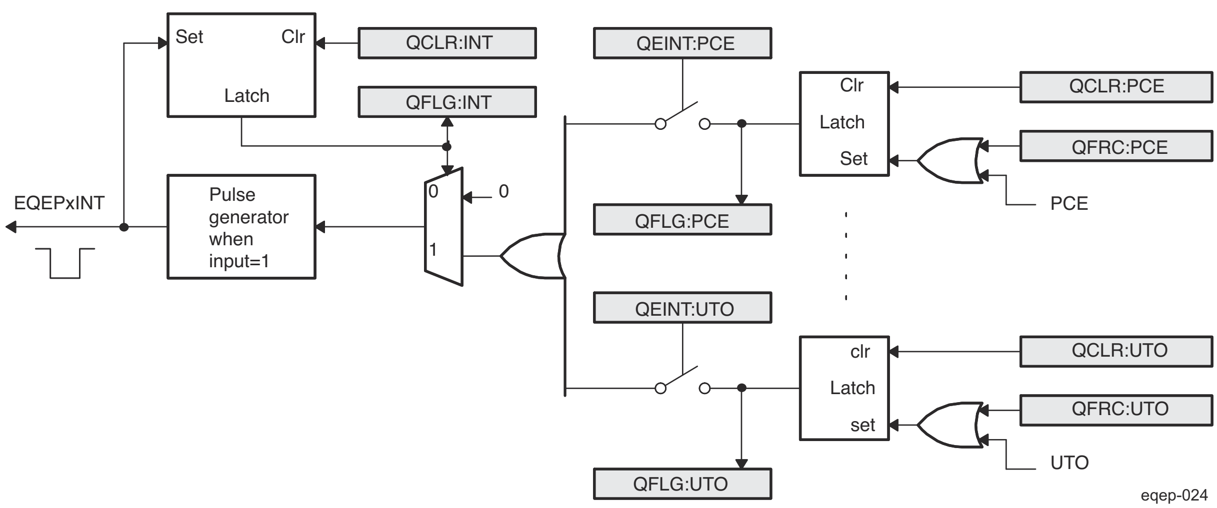


Figure 29-81. EQEP Interrupt Generation

Eleven interrupt events (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL, and UTO) can be generated. The interrupt control register (EQEP_QEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (EQEP_QFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated only to the interrupt controller if any of the interrupt events is enabled, the flag bit is 1 and the INT flag bit is 0. The interrupt service routine will need to clear the global interrupt flag bit and the serviced event, via the interrupt clear register (EQEP_QCLR), before any other interrupt pulses are generated. You can force an interrupt event by way of the interrupt force register (EQEP_QFRC), which is useful for test purposes.

29.4.27 Summary of PWMSS eQEP Functional Registers

Table 29-141 lists the registers with their memory locations, sizes, and reset values.

Table 29-141. eQEP Control and Status Functional Registers

Offset	Acronym	Register Description	Size(×16)/ #shadow
0h	EQEP_QPOSCNT	eQEP Position Counter Register	2/0
4h	EQEP_QPOSINIT	eQEP Position Counter Initialization Register	2/0
8h	EQEP_QPOSMAX	eQEP Maximum Position Count Register	2/0
Ch	EQEP_QPOSCMP	eQEP Position-Compare Register	2/1
10h	EQEP_QPOSILAT	eQEP Index Position Latch Register	2/0
14h	EQEP_QPOSSLAT	eQEP Strobe Position Latch Register	2/0
18h	EQEP_QPOSLAT	eQEP Position Counter Latch Register	2/0
1Ch	EQEP_QUTMR	eQEP Unit Timer Register	2/0
20h	EQEP_QUPRD	eQEP Unit Period Register	2/0
24h	EQEP_QWDTMR	eQEP Watchdog Timer Register	1/0
26h	EQEP_QWDPRD	eQEP Watchdog Period Register	1/0
28h	EQEP_QDECCTL	eQEP Decoder Control Register	1/0
2Ah	EQEP_QEPCTL	eQEP Control Register	1/0
2Ch	EQEP_QCAPCTL	eQEP Capture Control Register	1/0
2Eh	EQEP_QPOSCTL	eQEP Position-Compare Control Register	1/0
30h	EQEP_QEINT	eQEP Interrupt Enable Register	1/0
32h	EQEP_QFLG	eQEP Interrupt Flag Register	1/0
34h	EQEP_QCLR	eQEP Interrupt Clear Register	1/0
36h	EQEP_QFRC	eQEP Interrupt Force Register	1/0
38h	EQEP_QEPSTS	eQEP Status Register	1/0
3Ah	EQEP_QCTMR	eQEP Capture Timer Register	1/0
3Ch	EQEP_QCPRD	eQEP Capture Period Register	1/0
3Eh	EQEP_QCTMRLAT	eQEP Capture Timer Latch Register	1/0
40h	EQEP_QCPRDLAT	eQEP Capture Period Latch Register	1/0
5Ch	EQEP_REVID	eQEP Revision ID Register	2/0

29.4.28 PWMSS_EQEP Register Manual

This section provides description of the PWMSS eQEP relevant functional registers.

29.4.29 PWMSS_EQEP Instance Summary

Table 29-142. PWMSS_EQEP Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS1_EQEP	0x4843 E180	116 Bytes
PWMSS2_EQEP	0x4844 0180	116 Bytes
PWMSS3_EQEP	0x4844 2180	116 Bytes

29.4.30 PWMSS_EQEP Registers

29.4.31 PWMSS_EQEP Register Summary

Table 29-143. PWMSSn_EQEP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS1_EQEP Physical Address L4_PER2 Interconnect	PWMSS2_EQEP Physical Address L4_PER2 Interconnect	PWMSS3_EQEP Physical Address L4_PER2 Interconnect
EQEP_QPOSCNT	RW	32	0x0	0x4843 E180	0x4844 0180	0x4844 2180
EQEP_QPOSINIT	RW	32	0x4	0x4843 E184	0x4844 0184	0x4844 2184
EQEP_QPOSMAX	RW	32	0x8	0x4843 E188	0x4844 0188	0x4844 2188
EQEP_QPOSCMP	RW	32	0xC	0x4843 E18C	0x4844 018C	0x4844 218C
EQEP_QPOSILAT	R	32	0x10	0x4843 E190	0x4844 0190	0x4844 2190
EQEP_QPOSSLAT	R	32	0x14	0x4843 E194	0x4844 0194	0x4844 2194
EQEP_QPOSLAT	R	32	0x18	0x4843 E198	0x4844 0198	0x4844 2198
EQEP_QUTMR	RW	32	0x1C	0x4843 E19C	0x4844 019C	0x4844 219C
EQEP_QUPRD	RW	32	0x20	0x4843 E1A0	0x4844 01A0	0x4844 21A0
EQEP_QWDTMR	RW	16	0x24	0x4843 E1A4	0x4844 01A4	0x4844 21A4
EQEP_QWDPRD	RW	16	0x26	0x4843 E1A6	0x4844 01A6	0x4844 21A6
EQEP_QDECCTL	RW	16	0x28	0x4843 E1A8	0x4844 01A8	0x4844 21A8
EQEP_QEPCTL	RW	16	0x2A	0x4843 E1AA	0x4844 01AA	0x4844 21AA
EQEP_QCAPCTL	RW	16	0x2C	0x4843 E1AC	0x4844 01AC	0x4844 21AC
EQEP_QPOSCTL	RW	16	0x2E	0x4843 E1AE	0x4844 01AE	0x4844 21AE
EQEP_QEINT	RW	16	0x30	0x4843 E1B0	0x4844 01B0	0x4844 21B0
EQEP_QFLG	R	16	0x32	0x4843 E1B2	0x4844 01B2	0x4844 21B2
EQEP_QCLR	RW	16	0x34	0x4843 E1B4	0x4844 01B4	0x4844 21B4
EQEP_QFRC	RW	16	0x36	0x4843 E1B6	0x4844 01B6	0x4844 21B6
EQEP_QEPSTS	RW	16	0x38	0x4843 E1B8	0x4844 01B8	0x4844 21B8
EQEP_QCTMR	RW	16	0x3A	0x4843 E1BA	0x4844 01BA	0x4844 21BA
EQEP_QCPRD	RW	16	0x3C	0x4843 E1BC	0x4844 01BC	0x4844 21BC
EQEP_QCTMRLAT	R	16	0x3E	0x4843 E1BE	0x4844 01BE	0x4844 21BE
EQEP_QCPRDLAT	RW	16	0x40	0x4843 E1C0	0x4844 01C0	0x4844 21C0
EQEP_REVID	R	32	0x5C	0x4843 E1DC	0x4844 01DC	0x4844 21DC

29.4.32 PWMSS_EQEP Register Description

Table 29-144. EQEP_QPOSCNT

Address offset	0x0		
Physical Address	0x4843 E180 0x4844 0180 0x4844 2180	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCNT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSCNT	This 32 bit position counter register counts up/down on every eQEP pulse based on direction input. This counter acts as a position integrator whose count value is proportional to position from a give reference point.	RW	0x0

Table 29-145. Register Call Summary for Register EQEP_QPOSCNT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position-Compare Sync Output: [0]
- eQEP Position Counter Latch: [1]
- eQEP Position Counter Initialization: [2] [3]
- eQEP Edge Capture Unit: [4] [5]
- Summary of PWMSS eQEP Functional Registers: [6]
- PWMSS_EQEP Register Summary: [7]
- PWMSS_EQEP Register Description: [8] [9] [10]

Table 29-146. EQEP_QPOSINIT

Address offset	0x4		
Physical Address	0x4843 E184 0x4844 0184 0x4844 2184	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSINIT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSINIT	This register contains the position value that is used to initialize the position counter based on external strobe or index event. The position counter can be initialized through software.	RW	0x0

Table 29-147. Register Call Summary for Register EQEP_QPOSINIT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Initialization: [0] [1] [2] [3] [4]
- Summary of PWMSS eQEP Functional Registers: [5]
- PWMSS_EQEP Register Summary: [6]

Table 29-148. EQEP_QPOSMAX

Address offset	0x8		
Physical Address	0x4843 E188 0x4844 0188 0x4844 2188	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP

Table 29-148. EQEP_QPOSMAX (continued)

Description

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSMAX																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSMAX	This register contains the maximum position counter value.	RW	0x0

Table 29-149. Register Call Summary for Register EQEP_QPOSMAX

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Operating Modes: [0] [1] [2] [3] [4]
- Summary of PWMSS eQEP Functional Registers: [5]
- PWMSS_EQEP Register Summary: [6]

Table 29-150. EQEP_QPOSCMP

Address offset	0xC		
Physical Address	0x4843 E18C 0x4844 018C 0x4844 218C	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCMP																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSCMP	The position-compare value in this register is compared with the position counter (QPOSCNT field in EQEP_QPOSCNT) to generate sync output and/or interrupt on compare match.	RW	0x0

Table 29-151. Register Call Summary for Register EQEP_QPOSCMP

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position-Compare Sync Output: [0]
- eQEP Position-Compare Unit: [1] [2] [3]
- Summary of PWMSS eQEP Functional Registers: [4]
- PWMSS_EQEP Register Summary: [5]

Table 29-152. EQEP_QPOSILAT

Address offset	0x10		
Physical Address	0x4843 E190 0x4844 0190 0x4844 2190	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSILAT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSILAT	The position-counter value is latched into this register on an index event as defined by the QEPCTL[IEL] bits.	R	0x0

Table 29-153. Register Call Summary for Register EQEP_QPOSILAT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Operating Modes: [0]
- eQEP Position Counter Latch: [1] [2] [3] [4]
- Summary of PWMSS eQEP Functional Registers: [5]
- PWMSS_EQEP Register Summary: [6]
- PWMSS_EQEP Register Description: [7]

Table 29-154. EQEP_QPOSSLAT

Address offset	0x14		
Physical Address	0x4843 E194 0x4844 0194 0x4844 2194	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSSLAT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSSLAT	The position-counter value is latched into this register on strobe event as defined by the QEPCTL[SEL] bits.	R	0x0

Table 29-155. Register Call Summary for Register EQEP_QPOSSLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Latch: [0] [1] [2] [3]
- Summary of PWMSS eQEP Functional Registers: [4]
- PWMSS_EQEP Register Summary: [5]
- PWMSS_EQEP Register Description: [6]

Table 29-156. EQEP_QPOSLAT

Address offset	0x18		
Physical Address	0x4843 E198 0x4844 0198 0x4844 2198	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSLAT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSLAT	The position-counter value is latched into this register on unit time out event.	R	0x0

Table 29-157. Register Call Summary for Register EQEP_QPOSLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Operating Modes: [0]
- eQEP Edge Capture Unit: [1] [2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Register Summary: [4]
- PWMSS_EQEP Register Description: [5]

Table 29-158. EQEP_QUTMR

Address offset	0x1C

Table 29-158. EQEP_QUTMR (continued)

Physical Address	0x4843 E19C 0x4844 019C 0x4844 219C	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUTMR																															

Bits	Field Name	Description	Type	Reset
31:0	QUTMR	This register acts as time base for unit time event generation. When this timer value matches with unit time period value, unit time event is generated.	RW	0x0

Table 29-159. Register Call Summary for Register EQEP_QUTMR

Enhanced Quadrature Encoder Pulse (eQEP) Module

- Summary of PWMSS eQEP Functional Registers: [0]
- PWMSS_EQEP Register Summary: [1]

Table 29-160. EQEP_QUPRD

Address offset	0x20		
Physical Address	0x4843 E1A0 0x4844 01A0 0x4844 21A0	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUPRD																															

Bits	Field Name	Description	Type	Reset
31:0	QUPRD	This register contains the period count for unit timer to generate periodic unit time events to latch the eQEP position information at periodic interval and optionally to generate interrupt.	RW	0x0

Table 29-161. Register Call Summary for Register EQEP_QUPRD

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0] [1]
- Unit Timer Base: [2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Register Summary: [4]

Table 29-162. EQEP_QWDTMR

Address offset	0x24		
Physical Address	0x4843 E1A4 0x4844 01A4 0x4844 21A4	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QWDTMR															

Bits	Field Name	Description	Type	Reset
15:0	QWDTMR	This register acts as time base for watch dog to detect motor stalls. When this timer value matches with watch dog period value, watch dog timeout interrupt is generated. This register is reset upon edge transition in quadrature-clock indicating the motion.	RW	0x0

Table 29-163. Register Call Summary for Register EQEP_QWDTMR

Enhanced Quadrature Encoder Pulse (eQEP) Module

- Summary of PWMSS eQEP Functional Registers: [0]
- PWMSS_EQEP Register Summary: [1]

Table 29-164. EQEP_QWDPRD

Address offset	0x26		
Physical Address	0x4843 E1A6 0x4844 01A6 0x4844 21A6	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QWDPRD															

Bits	Field Name	Description	Type	Reset
15:0	QWDPRD	This register contains the time-out count for the eQEP peripheral watch dog timer. When the watchdog timer value matches the watchdog period value, a watchdog timeout interrupt is generated.	RW	0x0

Table 29-165. Register Call Summary for Register EQEP_QWDPRD

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Watchdog: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Register Summary: [2]

Table 29-166. EQEP_QDECCTL

Address offset	0x28		
Physical Address	0x4843 E1A8 0x4844 01A8 0x4844 21A8	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QSRC		SOEN	SPSEL	XCR	SWAP	IGATE	QAP	QBP	QIP	QSP	RESERVED				

Bits	Field Name	Description	Type	Reset
15:14	QSRC	Position-counter source selection. 0x0 = Quadrature count mode (QCLK = iCLK, QDIR = iDIR) 0x1 = Direction-count mode (QCLK = xCLK, QDIR = xDIR) 0x2 = UP count mode for frequency measurement (QCLK = xCLK, QDIR = 1) 0x3 = DOWN count mode for frequency measurement (QCLK = xCLK, QDIR = 0)	RW	0x0

Bits	Field Name	Description	Type	Reset
13	SOEN	Sync output-enable 0x0 = Disable position-compare sync output 0x1 = Enable position-compare sync output	RW	0x0
12	SPSEL	Sync output pin selection 0x0 = Index pin is used for sync output 0x1 = Strobe pin is used for sync output	RW	0x0
11	XCR	External clock rate 0x0 = 2x resolution: Count the rising/falling edge 0x1 = 1x resolution: Count the rising edge only	RW	0x0
10	SWAP	Swap quadrature clock inputs. This swaps the input to the quadrature decoder, reversing the counting direction. 0x0 = Quadrature-clock inputs are not swapped 0x1 = Quadrature-clock inputs are swapped	RW	0x0
9	IGATE	Index pulse gating option 0x0 = Disable gating of Index pulse 0x1 = Gate the index pin with strobe	RW	0x0
8	QAP	QEPA input polarity 0x0 = No effect 0x1 = Negates QEPA input	RW	0x0
7	QBP	QEPB input polarity 0x0 = No effect 0x1 = Negates QEPB input	RW	0x0
6	QIP	QEPI input polarity 0x0 = No effect 0x1 = Negates QEPI input	RW	0x0
5	QSP	QEPS input polarity 0x0 = No effect 0x1 = Negates QEPS input	RW	0x0
4:0	RESERVED		R	0x0

Table 29-167. Register Call Summary for Register EQEP_QDECCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Input Modes: [0] [1] [2] [3]
- eQEP Input Polarity Selection: [4] [5]
- eQEP Position-Compare Sync Output: [6] [7]
- eQEP Position Counter Operating Modes: [8]
- Summary of PWMSS eQEP Functional Registers: [9]
- PWMSS_EQEP Register Summary: [10]
- PWMSS_EQEP Register Description: [11]

Table 29-168. EQEP_QEPCTL

Address offset	0x2A		
Physical Address	0x4843 E1AA 0x4844 01AA 0x4844 21AA	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE_SOFT		PCRM		SEI		IEI		SWI	SEL	IEL		PHEN	QCLM	UTE	WDE

Bits	Field Name	Description	Type	Reset
15:14	FREE_SOFT	Emulation Control Bits. In the values 0 through 3 listed below, x is different for the four following behaviors. EQEP_QPOSCNT behavior, x refers to the Position counter. QWDTMR behavior, x refers to the Watchdog counter. QUTMR behavior, x refers to the Unit timer. QCTMR behavior, x refers to the Capture timer. 0x0 = x stops immediately. For QPOSCNT behavior, the stop is on emulation suspend. 0x1 = x continues to count until the rollover. 0x2 = x is unaffected by emulation suspend. 0x3 = x is unaffected by emulation suspend.	RW	0x0
13:12	PCRM	Position counter reset mode 0x0 = Position counter reset on an index event 0x1 = Position counter reset on the maximum position 0x2 = Position counter reset on the first index event 0x3 = Position counter reset on a unit time event	RW	0x0
11:10	SEI	Strobe event initialization of position counter 0x0 = Does nothing (action disabled) 0x1 = Does nothing (action disabled) 0x2 = Initializes the position counter on rising edge of the QEPS signal 0x3 = Clockwise Direction: Initializes the position counter on the rising edge of QEPS strobe. Counter Clockwise Direction: Initializes the position counter on the falling edge of QEPS strobe	RW	0x0
9:8	IEI	Index event initialization of position counter 0x0 = Do nothing (action disabled) 0x1 = Do nothing (action disabled) 0x2 = Initializes the position counter on the rising edge of the QEPI signal (QPOSCNT = QPOSINIT) 0x3 = Initializes the position counter on the falling edge of QEPI signal (QPOSCNT = QPOSINIT)	RW	0x0
7	SWI	Software initialization of position counter 0x0 = Do nothing (action disabled) 0x1 = Initialize position counter, this bit is cleared automatically	RW	0x0
6	SEL	Strobe event latch of position counter 0x0 = The position counter is latched on the rising edge of QEPS strobe (QPOSSLAT = POSCCNT). Latching on the falling edge can be done by inverting the strobe input using the QSP bit in the EQEP_QDECCTL register. 0x1 = Clockwise Direction: Position counter is latched on rising edge of QEPS strobe. Counter Clockwise Direction: Position counter is latched on falling edge of QEPS strobe.	RW	0x0
5:4	IEL	Index event latch of position counter (software index marker) 0x0 = Reserved 0x1 = Latches position counter on rising edge of the index signal 0x2 = Latches position counter on falling edge of the index signal 0x3 = Software index marker. Latches the position counter and quadrature direction flag on index event marker. The position counter is latched to the EQEP_QPOSILAT register and the direction flag is latched in the EQEP_QEPSTS[QDLF] bit. This mode is useful for software index marking.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	PHEN	Quadrature position counter enable/software reset 0x0 = Reset the eQEP peripheral internal operating flags/read-only registers. Control/configuration registers are not disturbed by a software reset. 0x1 = eQEP position counter is enabled	RW	0x0
2	QCLM	eQEP capture latch mode 0x0 = Latch on position counter read by CPU. Capture timer and capture period values are latched into EQEP_QCTMRLAT and EQEP_QCPRDLAT registers when CPU reads the EQEP_QPOSCNT register. 0x1 = Latch on unit time out. Position counter, capture timer and capture period values are latched into EQEP_QPOSLAT, EQEP_QCTMRLAT and EQEP_QCPRDLAT registers on unit time out.	RW	0x0
1	UTE	eQEP unit timer enable 0x0 = Disable eQEP unit timer 0x1 = Enable unit timer	RW	0x0
0	WDE	eQEP watchdog enable 0x0 = Disable the eQEP watchdog timer 0x1 = Enable the eQEP watchdog timer	RW	0x0

Table 29-169. Register Call Summary for Register EQEP_QEPCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter and Control Unit (PCCU): [0]
- eQEP Position Counter Operating Modes: [1] [2] [3]
- eQEP Position Counter Latch: [4] [5] [6] [7] [8] [9] [10] [11] [12]
- eQEP Position Counter Initialization: [13] [14] [15] [16]
- eQEP Edge Capture Unit: [17] [18]
- Summary of PWMSS eQEP Functional Registers: [19]
- PWMSS_EQEP Register Summary: [20]

Table 29-170. EQEP_QCAPCTL

Address offset	0x2C		
Physical Address	0x4843 E1AC 0x4844 01AC 0x4844 21AC	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CEN	RESERVED								CCPS			UPPS			

Bits	Field Name	Description	Type	Reset
15	CEN	Enable eQEP capture 0x0 = eQEP capture unit is disabled 0x1 = eQEP capture unit is enabled	RW	0x0
14:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:4	CCPS	eQEP capture timer clock prescaler 0x0 = CAPCLK = SYSCLKOUT/1 0x1 = CAPCLK = SYSCLKOUT/2 0x2 = CAPCLK = SYSCLKOUT/4 0x3 = CAPCLK = SYSCLKOUT/8 0x4 = CAPCLK = SYSCLKOUT/16 0x5 = CAPCLK = SYSCLKOUT/32 0x6 = CAPCLK = SYSCLKOUT/64 0x7 = CAPCLK = SYSCLKOUT/128	RW	0x0
3:0	UPPS	Unit position event prescaler 0x0 = UPEVNT = QCLK/1 0x1 = UPEVNT = QCLK/2 0x2 = UPEVNT = QCLK/4 0x3 = UPEVNT = QCLK/8 0x4 = UPEVNT = QCLK/16 0x5 = UPEVNT = QCLK/32 0x6 = UPEVNT = QCLK/64 0x7 = UPEVNT = QCLK/128 0x8 = UPEVNT = QCLK/256 0x9 = UPEVNT = QCLK/512 0xA = UPEVNT = QCLK/1024 0xB = UPEVNT = QCLK/2048 0xC = Reserved 0xD = Reserved 0xE = Reserved 0xF = Reserved	RW	0x0

Table 29-171. Register Call Summary for Register EQEP_QCAPCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0] [1] [2] [3]
- Summary of PWMSS eQEP Functional Registers: [4]
- PWMSS_EQEP Register Summary: [5]

Table 29-172. EQEP_QPOSCTL

Address offset	0x2E		
Physical Address	0x4843 E1AE 0x4844 01AE 0x4844 21AE	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCSHDW	PCLOAD	PCPOL	PCE	PCSPW											

Bits	Field Name	Description	Type	Reset
15	PCSHDW	Position-compare shadow enable 0x0 = Shadow disabled, load Immediate 0x1 = Shadow enabled	RW	0x0
14	PCLOAD	Position-compare shadow load mode 0x0 = Load on QPOSCNT = 0 0x1 = Load when QPOSCNT = QPOSCMP	RW	0x0

Bits	Field Name	Description	Type	Reset
13	PCPOL	Polarity of sync output 0x0 = Active HIGH pulse output 0x1 = Active LOW pulse output	RW	0x0
12	PCE	Position-compare enable/disable 0x0 = Disable position compare unit 0x1 = Enable position compare unit	RW	0x0
11:0	PCSPW	Select-position-compare sync output pulse width ... 0x0 = 1 x 4 x SYSCLKOUT cycles 0x1 = 2 x 4 x SYSCLKOUT cycles 0x2 = 3 x 4 x SYSCLKOUT cycles to 4096 x 4 x SYSCLKOUT cycles 0xFFF = 3 x 4 x SYSCLKOUT cycles to 4096 x 4 x SYSCLKOUT cycles	RW	0x0

Table 29-173. Register Call Summary for Register EQEP_QPOSCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter and Control Unit (PCCU): [0]
- eQEP Position-Compare Unit: [1] [2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Register Summary: [4]

Table 29-174. EQEP_QEINT

Address offset	0x30		
Physical Address	0x4843 E1B0 0x4844 01B0 0x4844 21B0	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Unit time out interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
10	IEL	Index event latch interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
9	SEL	Strobe event latch interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
8	PCM	Position-compare match interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
7	PCR	Position-compare ready interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	PCO	Position counter overflow interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
5	PCU	Position counter underflow interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
4	WTO	Watchdog time out interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
3	QDC	Quadrature direction change interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
2	PHE	Quadrature phase error interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
1	PCE	Position counter error interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
0	RESERVED		R	0x0

Table 29-175. Register Call Summary for Register EQEP_QEINT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Interrupt Structure: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Register Summary: [2]

Table 29-176. EQEP_QFLG

Address offset	0x32		
Physical Address	0x4843 E1B2 0x4844 01B2 0x4844 21B2	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Unit time out interrupt flag 0x0 = No interrupt generated 0x1 = Set by eQEP unit timer period match	R	0x0
10	IEL	Index event latch interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set after latching the QPOSCNT to QPOSILAT	R	0x0
9	SEL	Strobe event latch interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set after latching the QPOSCNT to EQEP_QPOSSLAT	R	0x0

Bits	Field Name	Description	Type	Reset
8	PCM	eQEP compare match event interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set on position-compare match	R	0x0
7	PCR	Position-compare ready interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set after transferring the shadow register value to the active position compare register.	R	0x0
6	PCO	Position counter overflow interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set on position counter overflow.	R	0x0
5	PCU	Position counter underflow interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set on position counter underflow.	R	0x0
4	WTO	Watchdog timeout interrupt flag 0x0 = No interrupt generated 0x1 = Set by watch dog timeout	R	0x0
3	QDC	Quadrature direction change interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set during change of direction	R	0x0
2	PHE	Quadrature phase error interrupt flag 0x0 = No interrupt generated 0x1 = Set on simultaneous transition of QEPA and QEPB	R	0x0
1	PCE	Position counter error interrupt flag 0x0 = No interrupt generated 0x1 = Position counter error	R	0x0
0	INT	Global interrupt status flag 0x0 = No interrupt generated 0x1 = Interrupt was generated	R	0x0

Table 29-177. Register Call Summary for Register EQEP_QFLG

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Input Modes: [0]
- eQEP Position Counter Operating Modes: [1] [2] [3]
- eQEP Position Counter Latch: [4] [5]
- eQEP Position Counter Initialization: [6] [7]
- eQEP Position-Compare Unit: [8] [9]
- eQEP Watchdog: [10]
- Unit Timer Base: [11]
- eQEP Interrupt Structure: [12]
- Summary of PWMSS eQEP Functional Registers: [13]
- PWMSS_EQEP Register Summary: [14]

Table 29-178. EQEP_QCLR

Address offset	0x34		
Physical Address	0x4843 E1B4 0x4844 01B4 0x4844 21B4	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Clear unit time out interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
10	IEL	Clear index event latch interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
9	SEL	Clear strobe event latch interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
8	PCM	Clear eQEP compare match event interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
7	PCR	Clear position-compare ready interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
6	PCO	Clear position counter overflow interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
5	PCU	Clear position counter underflow interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
4	WTO	Clear watchdog timeout interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
3	QDC	Clear quadrature direction change interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
2	PHE	Clear quadrature phase error interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
1	PCE	Clear position counter error interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
0	INT	Global interrupt clear flag 0x0 = No effect 0x1 = Clears the interrupt flag and enables further interrupts to be generated if an event flags is set to 1.	RW	0x0

Table 29-179. Register Call Summary for Register EQEP_QCLR

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Interrupt Structure: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Register Summary: [2]

Table 29-180. EQEP_QFRC

Address offset	0x36

Table 29-180. EQEP_QFRC (continued)

Physical Address	0x4843 E1B6 0x4844 01B6 0x4844 21B6	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Force unit time out interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
10	IEL	Force index event latch interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
9	SEL	Force strobe event latch interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
8	PCM	Force position-compare match interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
7	PCR	Force position-compare ready interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
6	PCO	Force position counter overflow interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
5	PCU	Force position counter underflow interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
4	WTO	Force watchdog time out interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
3	QDC	Force quadrature direction change interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
2	PHE	Force quadrature phase error interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
1	PCE	Force position counter error interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
0	RESERVED		R	0x0

Table 29-181. Register Call Summary for Register EQEP_QFRC

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Interrupt Structure: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Register Summary: [2]

Table 29-182. EQEP_QEPSTS

Address offset	0x38		
Physical Address	0x4843 E1B8 0x4844 01B8 0x4844 21B8	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								UPEVNT	FDF	QDF	QDLF	COEF	CDEF	FIMF	PCEF

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	UPEVNT	Unit position event flag 0x0 = No unit position event detected 0x1 = Unit position event detected. Write 1 to clear.	R	0x0
6	FDF	Direction on the first index marker. Status of the direction is latched on the first index event marker. 0x0 = Counter-clockwise rotation (or reverse movement) on the first index event 0x1 = Clockwise rotation (or forward movement) on the first index event	R	0x0
5	QDF	Quadrature direction flag 0x0 = Counter-clockwise rotation (or reverse movement) 0x1 = Clockwise rotation (or forward movement)	R	0x0
4	QDLF	eQEP direction latch flag. Status of direction is latched on every index event marker. 0x0 = Counter-clockwise rotation (or reverse movement) on index event marker 0x1 = Clockwise rotation (or forward movement) on index event marker	R	0x0
3	COEF	Capture overflow error flag 0x0 = Sticky bit, cleared by writing 1 0x1 = Overflow occurred in eQEP Capture timer (QEPCTMR)	RW	0x0
2	CDEF	Capture direction error flag 0x0 = Sticky bit, cleared by writing 1 0x1 = Direction change occurred between the capture position event.	RW	0x0
1	FIMF	First index marker flag 0x0 = Sticky bit, cleared by writing 1 0x1 = Set by first occurrence of index pulse	RW	0x0
0	PCEF	Position counter error flag. This bit is not sticky and it is updated for every index event. 0x0 = No error occurred during the last index transition. 0x1 = Position counter error	R	0x0

Table 29-183. Register Call Summary for Register EQEP_QEPSTS

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter Input Modes: [0]
- eQEP Position Counter Operating Modes: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]
- eQEP Position Counter Latch: [13] [14] [15] [16]
- eQEP Edge Capture Unit: [17] [18] [19]
- Summary of PWMSS eQEP Functional Registers: [20]
- PWMSS_EQEP Register Summary: [21]
- PWMSS_EQEP Register Description: [22]

Table 29-184. EQEP_QCTMR

Address offset	0x3A		
Physical Address	0x4843 E1BA 0x4844 01BA 0x4844 21BA	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCTMR															

Bits	Field Name	Description	Type	Reset
15:0	QCTMR	This register provides time base for edge capture unit.	RW	0x0

Table 29-185. Register Call Summary for Register EQEP_QCTMR

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0] [1]
- Summary of PWMSS eQEP Functional Registers: [2]
- PWMSS_EQEP Register Summary: [3]

Table 29-186. EQEP_QCPRD

Address offset	0x3C		
Physical Address	0x4843 E1BC 0x4844 01BC 0x4844 21BC	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCPRD															

Bits	Field Name	Description	Type	Reset
15:0	QCPRD	This register holds the period count value between the last successive eQEP position events	RW	0x0

Table 29-187. Register Call Summary for Register EQEP_QCPRD

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0] [1] [2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Register Summary: [4]

Table 29-188. EQEP_QCTMRLAT

Address offset	0x3E

Table 29-188. EQEP_QCTMRLAT (continued)

Physical Address	0x4843 E1BE 0x4844 01BE 0x4844 21BE	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCTMRLAT															

Bits	Field Name	Description	Type	Reset
15:0	QCTMRLAT	The eQEP capture timer value can be latched into this register on two events, that is, unit timeout event, reading the eQEP position counter.	R	0x0

Table 29-189. Register Call Summary for Register EQEP_QCTMRLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0] [1]
- Summary of PWMSS eQEP Functional Registers: [2]
- PWMSS_EQEP Register Summary: [3]
- PWMSS_EQEP Register Description: [4] [5]

Table 29-190. EQEP_QCPRDLAT

Address offset	0x40		
Physical Address	0x4843 E1C0 0x4844 01C0 0x4844 21C0	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCPRDLAT															

Bits	Field Name	Description	Type	Reset
15:0	QCPRDLAT	eQEP capture period value can be latched into this register on two events, that is, unit timeout event, reading the eQEP position counter.	RW	0x0

Table 29-191. Register Call Summary for Register EQEP_QCPRDLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0] [1] [2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Register Summary: [4]
- PWMSS_EQEP Register Description: [5] [6]

Table 29-192. EQEP_REVID

Address offset	0x5C		
Physical Address	0x4843 E1DC 0x4844 01DC 0x4844 21DC	**Instance**	PWMSS1_EQEP PWMSS2_EQEP PWMSS3_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-[1]

(1) TI Internal data

Table 29-193. Register Call Summary for Register EQEP_REVID

Enhanced Quadrature Encoder Pulse (eQEP) Module

- Summary of PWMSS eQEP Functional Registers: [0]
- PWMSS_EQEP Register Summary: [1]

Chapter 30

Viterbi-Decoder Coprocessor

This chapter describes the Viterbi-Decoder Coprocessor (VCP) module.

30.1 VCP Overview........ 6867
30.2 VCP Integration........ 6869
30.3 VCP Functional Description........ 6871
30.4 VCP Modules Programming Guide........ 6890
30.5 VCP Register Manual........ 6897

30.1 VCP Overview

Channel decoding of voice and low bit-rate data channels found in cellular standards such as 2.5G, 3G and WiMAX require the decoding of convolutional encoded data. The Viterbi-decoder coprocessor 2 (VCP2) performs Viterbi-decoding for IS2000 and 3GPP wireless standards, and performs forward error correction for 2G and 3G wireless systems. The VCP can support 1941 12.2 Kbps class A 3G voice channels running at 333 MHz and offers a very cost effective and synergistic solution when combined with Texas Instruments (TI) DSPs. (for more information, see Section 30.1.1, *VCP Features*).

The device provides two instances of the VCP module respectively VCP1 and VCP2.

Figure 30-1 is an overview of the VCP in the device.

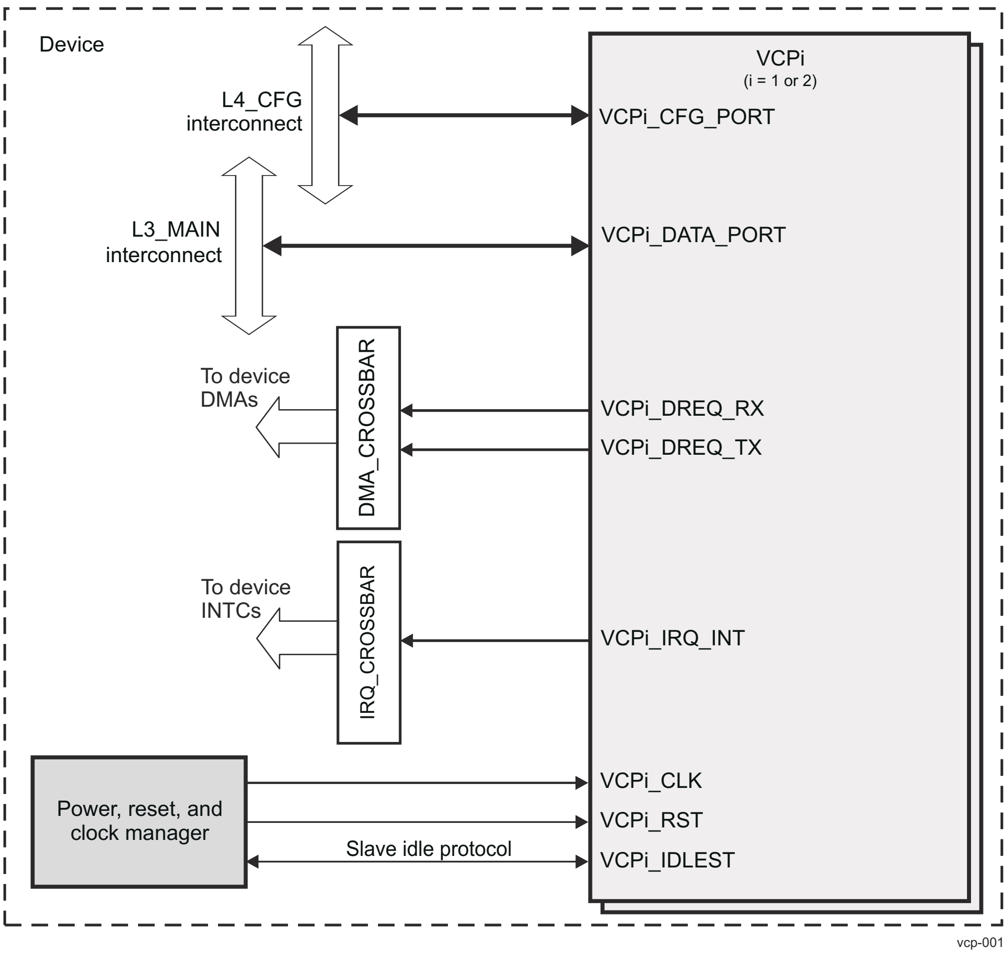


Figure 30-1. VCP Highlight

30.1.1 VCP Features

The main features of the VCP modules:

- L3 interconnect slave interface supports:
  - 64-bit data bus width
- L4 interconnect slave interface supports:
  - 32-bit access and configuration bus supported
  - Transmit and receive direct memory access (DMA) requests supported through DMA_CROSSBAR
  - Interrupt line for multiple interrupt source events to IRQ_CROSSBAR
  - Power management through IDLE power state

The VCP module provides:

- High flexibility
  - Variable constraint length, K = 5, 6, 7, 8, or 9
  - User-supplied code coefficients
  - Code rates (1/2, 1/3, or 1/4)
  - Configurable trace back settings (convergence distance, frame structure)
  - Branch metrics calculation and depuncturing done in software by the DSP
- System and development cost optimization
  - Releases DSP resources for other processing
  - Reduces board space and power consumption by performing decoding on-chip
  - Communicates between the DSP and the VCP1, VCP2 modules through the high performance EDMA engine
  - Uses its own optimized working memories
  - Provides debug capabilities during frame processing
  - Provides libraries for reduced development time

30.2 VCP Integration

This section describes the integration of the VCP1 and VCP2 modules in the device, including information about clocks, resets, and hardware requests.

Figure 30-2 shows the integration of the VCP modules in the device, including interrupt handlers, DMA requests, clock generators, and interconnections.

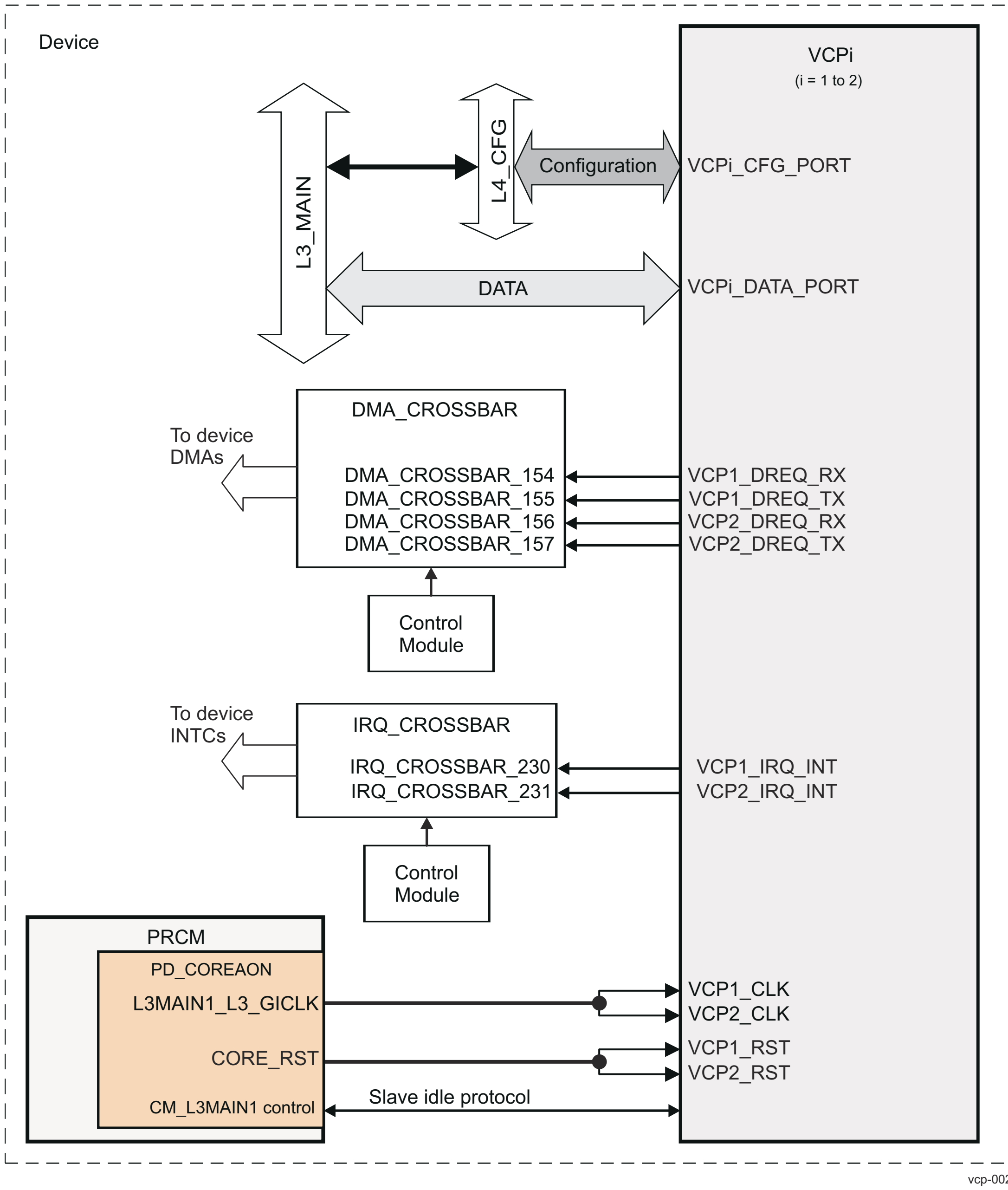


Figure 30-2. VCP1 and VCP2 Integration

Table 30-1 through Table 30-3 summarize the integration of the VCP modules in the device.

Table 30-1. VCP Integration Attributes

Module Instance	Attributes		
	Power Domain	**Wake-Up Capability**	**Interconnect**

Table 30-1. VCP Integration Attributes (continued)

VCP1	PD_COREAON	No	L3_MAIN
VCP2	PD_COREAON	No	L3_MAIN

Table 30-2. VCP Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VCP1	VCP1_CLK	L3MAIN1_L3_GICLK	PRCM	L3_MAIN interface clock
VCP2	VCP2_CLK	L3MAIN1_L3_GICLK	PRCM	L3_MAIN interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VCP1	VCP1_RST	CORE_RST	PRCM	COREAON power domain reset
VCP2	VCP2_RST	CORE_RST	PRCM	COREAON power domain reset

Table 30-3. VCP Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
VCP1	VCP1_IRQ_INT	IRQ_CROSSBAR_230	-	VCP1 interrupt request line
VCP2	VCP2_IRQ_INT	IRQ_CROSSBAR_231	-	VCP2 interrupt request line
DMA Requests				
Module Instance	**Source Signal Name**	**DMA_CROSSBAR Input**	**Default Mapping**	**Description**
VCP1	VCP1_DREQ_RX	DMA_CROSSBAR_154	DMA_DSP1_DREQ_16 DMA_DSP2_DREQ_16	VCP1 RX Event
	VCP1_DREQ_TX	DMA_CROSSBAR_155	DMA_DSP1_DREQ_17 DMA_DSP2_DREQ_17	VCP1 TX Event
VCP2	VCP2_DREQ_RX	DMA_CROSSBAR_156	DMA_DSP1_DREQ_18 DMA_DSP2_DREQ_18	VCP2 RX Event
	VCP2_DREQ_TX	DMA_CROSSBAR_157	DMA_DSP1_DREQ_19 DMA_DSP2_DREQ_19	VCP2 TX Event

Note

The VCP1 and VCP2 interrupt request signals are not mapped by default.

The VCP1 and VCP2 DMA request signals are not mapped by default to DMA_SYSTEM.

For more information about the IRQ_CROSSBAR and DMA_CROSSBAR modules, see sections: Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description* and Section 18.4.6.5 , *DMA_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.

For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

Note

- For more information about the interrupt source, see Section 30.3.11.2, *Interrupt Requests*.
- For more information about the DMA source, see Section 30.3.12, *DMA Requests*.

30.3 VCP Functional Description

This section is a functional description of the VCP1 and VCP2 modules in the device.

30.3.1 VCP Block Diagram

Figure 30-3 shows the functional block diagram of the two instances of the VCP modules. It consists of four main blocks:

- VCP control, this block controls all the blocks within the VCP. It stores the Viterbi decoder parameters, analyzes commands from the DSP, and controls the global sequencing of events. The control unit also generates the two synchronization events to the DMA: a write event (VCPXEVT) and a read event (VCPREVT). The configuration bus loads the VCP with all the parameters and controls that affect the execution of the VCP. These parameters are memory mapped and accessed as 32-bit registers. These registers can be read and written by this bus.
- EDMA I/F unit, connected to L3 Interconnect, typically accessed by the EDMA. The EDMA is used to send the bulk of the input and output data. The data input to the VCP using the EDMA/L3 Interconnect bus are the branch metrics. The data output from the VCP using the EDMA/L3 Interconnect bus are either the hard decisions or soft decisions. The intended use of the EDMA bus is to transfer data that requires high bandwidths. This data includes the branch metrics and the decoded data. The intended use of the configuration bus is load data that requires low bandwidths. This includes the input parameters and output status registers.
- Memory block, contains all of the RAMs used by the Viterbi decoder
- Processing unit, of the VCP is realized by two main modules: the State Metrics Unit and the Trace Back Unit.

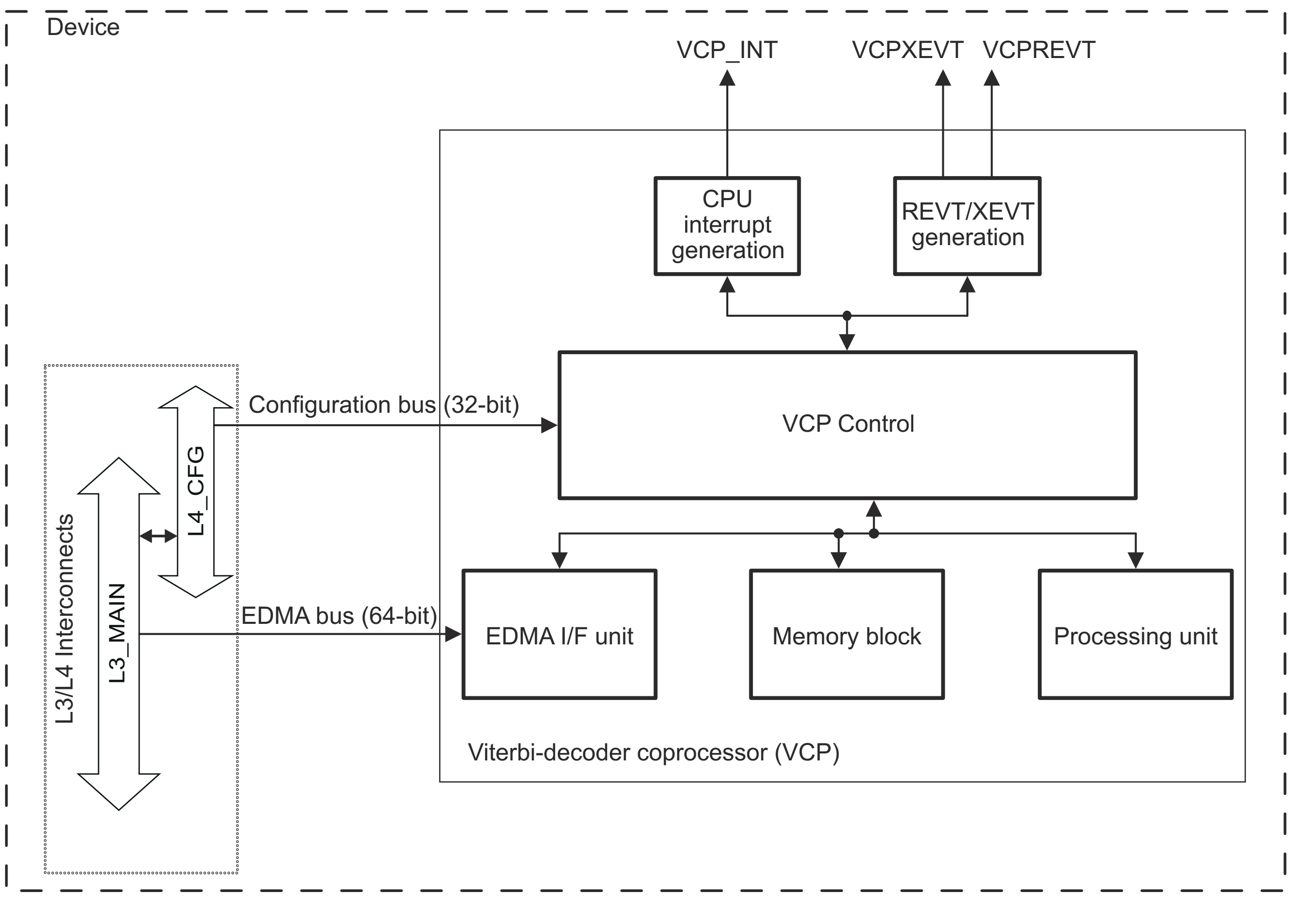


Figure 30-3. VCP Block Diagram

The DSP controls the operation of the VCP on Figure 30-3 using memory-mapped registers. The DSP typically sends and receives data using synchronized EDMA transfers through the 64-bit EDMA bus. The VCP sends two synchronization events to the EDMA:

- receive event (VCP1REVT and VCP2REVT)
- transmit event (VCP1XEVT and VCP2XEVT)

The VCP input data corresponds to the branch metrics and the output data to the hard decisions or soft decisions.

30.3.2 VCP Internal Interfaces

The interface signals can be grouped into the following four categories:

- The clock and reset signals (see Section 30.2, *VCP Integration* for signal mapping).
- The register interface that is used for the connection with the L4 interconnect for configuration and data transfer through L3_MAIN.
- The EDMA and interrupt interface used to request new context and packet data or to indicate available result data.
- Slave idle protocol to PRCM.

30.3.2.1 VCP Power Management

Two operating states are defined for all the VCP modules:

- ACTIVE state: The module is running synchronously on the interface and functional clocks. Interrupts and DMA requests can be generated according to the configuration (register, master or slave mode, etc).
- IDLE state: As part of the system power management, the PRCM module can request the VCP modules to enter IDLE state. Depending on the configured acknowledgment mode, a VCP module effectively enters IDLE state or not. As soon as a VCP module enters IDLE state, its only activities are those unrelated to clock activity and its clocks are likely to be switched off at the PRCM level.

Note

IDLE request and IDLE acknowledge are only internal signals, with no means to observe or to control. The generation and control of the signals is purely hardware (managed automatically by the PRCM module and the VCP depending on the IDLEMODE settings).

Table 30-4 lists power-management features available for the VCP1 and VCP2 modules. For more information on global PRCM features, refer toChapter 3 *Power Reset and Clock Management*.

Table 30-4. Local Power-Management Features

Feature	Registers	Description
Clock Auto Gating	N/A	N/A
Slave Idle Modes	VCP_SYSCONFIG[3:2] IDLEMODE	• Idle request unconditionally acknowledged • Acknowledge always inactive • Idle request acknowledge pending internal conditions
Clock Activity	N/A	N/A
Master Standby Modes	N/A	N/A
Global Wake-up Enable	N/A	N/A
Wake-up Sources Enable	N/A	N/A

30.3.2.1.1 Idle Mode

Setting the VCP_SYSCONFIG[3:2] IDLEMODE bit-field to 0x2 activates the interface clock gating for power saving. In idle mode, the clock is active only when the registers are accessed or a VCP processing is ongoing.

When the IDLEMODE bit-field is set to 0x1- Acknowledge always inactive. When this bit-field is set to 0x0, Idle request unconditionally acknowledged and the clock is free-running.

30.3.2.2 VCP Clocks

The source of VCP1_CLK and VCP2_CLK is a interface clock signal L3MAIN1_L3_GICLK from CD_L3MAIN in PRCM.

The VCP1_CLK and VCP2_CLK run at the L3/L4 interconnect clock speeds. It is used to trigger access to the VCP modules - L3_MAIN to DATA port and L4_CFG to CFG port interfaces through the Cortex-A15 MPU/DSP shared bus.

Note

At the PRCM level, when all the conditions to shut off the L3MAIN1_L3_GICLK clock are met the PRCM module automatically launches a hardware handshake protocol to ensure VCP1 and VCP2 are ready to have this clock switched off. Namely, the PRCM module asserts an IDLE request to the VCP modules.

At the PRCM level the status of VCP modules clock signals observe by registers: CM_L3MAIN1_VCP1_CLKCTRL and CM_L3MAIN1_VCP2_CLKCTRL.

For more information, refer to Chapter 3, *Power, Reset, and Clock Management*.

30.3.2.3 VCP Resets

Table 30-5 lists the resets used in the VCP1 and VCP2 modules.

Table 30-5. VCP1 and VCP2 Reset Description

Type	Name	Source	Description
Hardware	VCP1_RST VCP2_RST	PRCM	Global Reset
Software	VCP_SYSCONFIG[0] RESET_DONE	Internal	Reset from the idle command, reset is complete when its value is 1

This section covers the reset feature of the VCP1 and VCP2 modules.

The VCP_SYSCONFIG[0] RESET_DONE bit shows the status of the reset from the idle command. It indicates that the reset is complete when its value is 1. Software must ensure that the reset completes before doing VCP operations.

The behavior of software reset and hardware reset is the same, except that the software reset bit resets this module without affecting the whole device reset core domain.

30.3.2.4 Interrupt Requests

The VCP1 and VCP2 modules generates events shown in Table 30-6. An interrupt can be enable or disable by using the VCP_IRQENABLE_SET register.

Table 30-6. VCP1 and VCP2 Interrupts Description

Event Name	Description
VCP_IRQSTATUS[0] STATUS	Current active status of the interrupts after the enabling function.
VCP_IRQENABLE_SET[0] ENABLE_SET	User enables the VCP error interrupt
VCP_IRQENABLE_CLR[0] ENABLE_CLR	User disables the VCP error interrupt.

30.3.2.5 EDMA Requests

The VCP1 and VCP2 have EDMA request ports to request a new context, input data.

Each VCP has its own set of EDMA request signals, resulting in a set of four EDMA request signals (see Table 30-3).

The status of the EDMA requests observe by registers:

- VCP_DEBUG[0] DMA_R_REQ - Status of the VCP receive event (VCP1REVT and VCP2REVT)
- VCP_DEBUG[1] DMA_X_REQ - Status of the VCP tranmit event (VCP1XEVT and VCP2XEVT)

30.3.3 Functional Overview

The Viterbi Coprocessor performs the decoding of the following convolutional encoders:

- Linear shift register
- Outputs are combinations of previous inputs, defined by polynomial matrix.
- Constraint length K: number of delay elements + 1
- Code *rate r=1/n*: 1 input and n outputs, the number of outputs is issued from the number of polynomials, ie the number of rows in the matrix.

The Viterbi coprocessor (VCP) supports decoding of convolutional encoders with constraint length K from 5 to 9, and rates of 1/2, 1/3 or 1/4.

30.3.3.1 Theoretical Basics of the Convolutional Code.

A convolutional code is generated by passing the information sequence to be transmitted through a linear finite-state shift register. The VCP module is able to decode only a subset of those codes known as a single-shift register, nonrecursive convolutional code (see Figure 30-4). Important parameters for these types of codes are:

- The constraint length K (length of the delay line, the VCP2 supports K values from 5 to 9)
- The rate R given by R = k/n where k is the number of information bits needed to produce n output bits also known as codewords (the VCP2 supports 1/2, 1/3, and 1/4 codes with rates)
- The generator polynomials Gn describe how the outputs are generated from the inputs

30.3.3.2

Figure 30-4 shows the example block diagram of the Convolutional Encoder in VCP module.

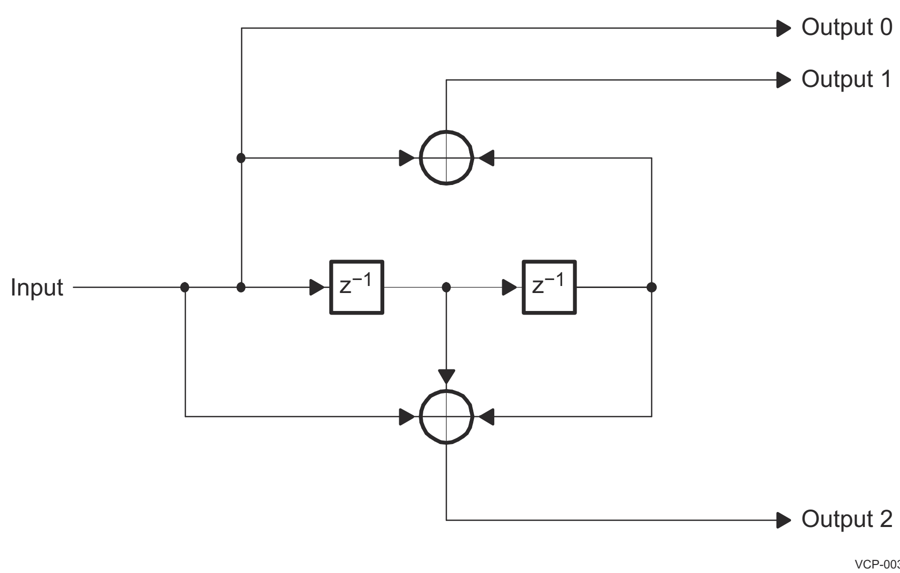


Figure 30-4. Convolutional Encoder Example Block Diagram

From the parameters, it can derive a trellis diagram that provides a useful representation of the code but whose complexity grows exponentially with the constraint length K. Figure 30-5 shows the trellis diagram of the code from Figure 30-4. The fact that there is a limited number of possible transitions from one state to another makes the code powerful and will be used in the decoding process.

As a maximum-likelihood sequence estimation (MLSE) decoder, the Viterbi decoder identifies the code sequence with the highest probability of matching the transmitted sequence based on the received sequence.

The Viterbi algorithm is composed of a metric update and a traceback routine. The metric update performs a forward recursion in the trellis over a finite number of symbol periods where probabilities are accumulated (the VCP accumulates on 13 bits) for each individual state, based on the current input symbol (branch metric information). The accumulated metric is known as path metrics or state metrics. Once a path through the trellis is identified, the traceback routine performs a backward recursion in the trellis and outputs hard decisions or soft decisions.

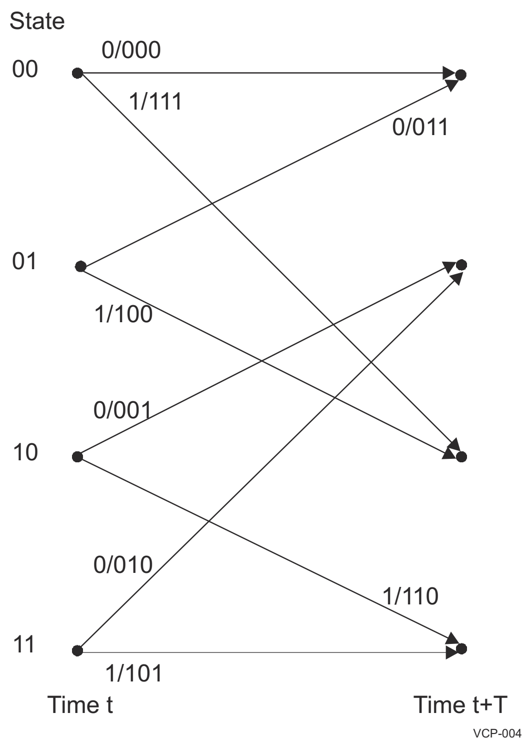


Figure 30-5. Trellis Diagram for Convolutional Encoder Example Block Diagram

30.3.4 VCP Architecture

A block diagram of the VCP processing unit is shown on Figure 30-6. The state metrics unit performs the Viterbi forward recursion using branch metrics as inputs, and updates the states metrics for all states (Add/Compare/Select or ACS operations) at every trellis stage. The state metrics memory is not accessible by the DSP. The traceback unit performs the Viterbi backward recursion and generates harddecisions or soft-decisions. The traceback memories are not directly accessible by the DSP.

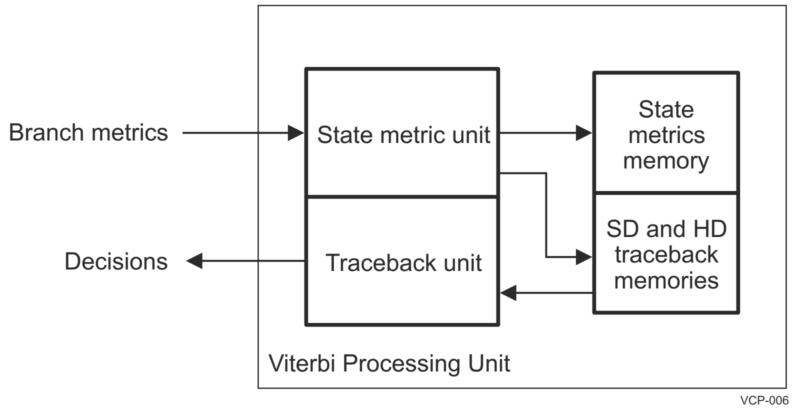


Figure 30-6. Viterbi Processing Unit

30.3.4.1 Sliding Windows Processing

The traceback hard decision memory can store up to 32768 traceback bits and there are $2^{(K-1)}$ bits stored at each trellis stage. Therefore, the traceback hard decision memory can store decisions of $32768/2^{(K-1)}$ symbols. The traceback soft decision memory can store up to 8192 traceback soft values and, therefore, contain up to 8192 soft decisions of $8192/2^{(K-1)}$ symbols. Assume a terminated frame of length F (excluding tail bits) and a constraint length K, F and K determine whether all decisions can be stored in the traceback memories.

If the decisions do not fit in the traceback HD/SD memory, then the convergent or mixed mode is used and the original frame is segmented into sliding windows (SW) otherwise, the traceback mode is set to tailed and no segmentation is required.

30.3.4.1.1 Tailed Traceback Mode

This mode is used when a full frame can reside within the coprocessor traceback memory. The state metrics are computed over $F + K - 1$ symbols, the traceback is initialized with the tails state and executed over $F + K - 1$ symbols. It should be noted that only F decisions are output. They are output in reverse order and in blocks of user-defined size.

Figure 30-7 shows Tailed Traceback Mode.

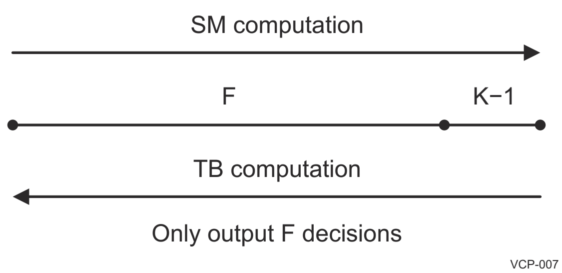


Figure 30-7. Tailed Traceback Mode

30.3.4.1.2 Mixed Traceback Mode

This mode is used when the full frame does not fit into the coprocessor traceback memory and the frame is terminated. The frame is split into sliding windows. The state metrics are computed over $F + K - 1$ symbols, the traceback is initialized with the tails state and executed over $F + K - 1$ symbols. It should be noted that only F decisions are output in blocks of user-defined size (see Section Figure 30-7). The state metrics computation of sliding window $i + 1$ is done in parallel with the traceback computation of sliding window i. Tailed traceback type is used on the last sliding window.

Figure 30-8 shows the sliding windows diagram for Mixed Traceback Mode.

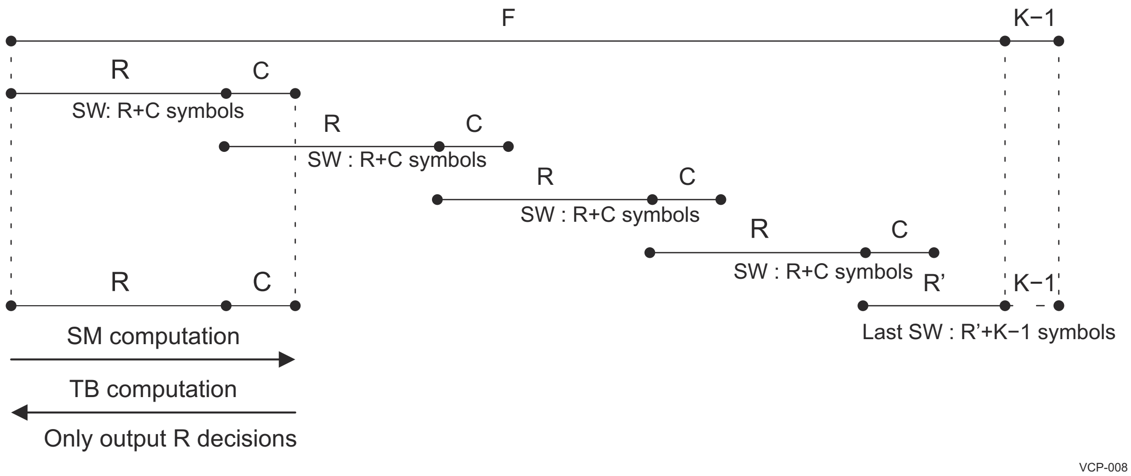


Figure 30-8. Mixed Traceback Mode - Example with Five Sliding Windows

30.3.4.1.3 Convergent Traceback Mode

This mode is used with nonterminated frames or when must decode a portion of the frame. When the frame does not fit into the coprocessor traceback memory, then the frame is split into sliding windows. The state metrics are computed over *F* + *C* symbols, the traceback is initialized with the tails state and executed over *F* + *C* symbols. It should be noted that only *F* decisions are output in blocks of user-defined size (see Section Figure 30-7). The state metrics computation of sliding window *i* + *1* is done in parallel with the traceback computation of sliding window *i*.

Figure 30-9 shows the sliding windows diagram for Convergent Traceback Mode.

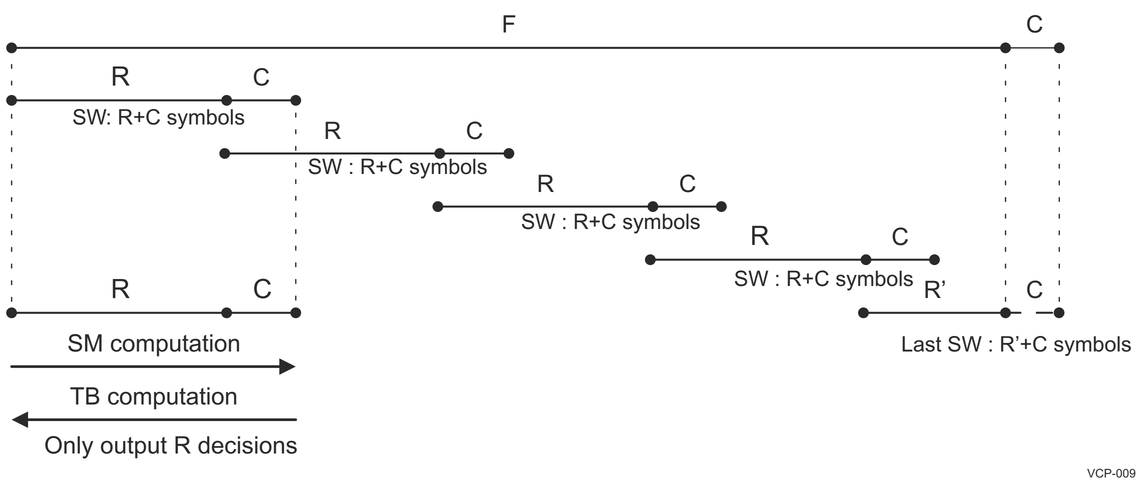


Figure 30-9. Convergent Traceback Mode - Example With Five Sliding Windows

30.3.4.1.4 F, R, and C Limitations

VCP modules have increased the traceback soft decision memory and output FIFO compared to the memory. The Traceback soft decision memory size has been increased from 1024 × 96 bits to 2048 × 64 bits. The output FIFO memory has been increased from 32 × 64 bits to 64 × 64 bits. In addition, the soft decision resolution is increased internally to 13 bits and is clipped to 8 bits when output from VCP module. These memory sizes have an effect on the *Fmax* parameter (i.e., the maximum frame size for tailed processing), and the *(R+C)* max parameter (i.e., the maximum sliding window length for mixed/convergent processing).

The VCP version 2 performance is as follows:

- Soft decisions:

- Allowed values for *C* are *C = N*(K-1)*, where *N* is a convergence multiplier. Table 30-7 and Table 30-8 show the possible values of N for each value of K for hard and soft decisions.
- *R* is constrained to be less than or equal to 248.
- *Fmax* and *(R+C)* max for VCP version 2 have been increased over VCP. This change reduces the number of sliding windows and, therefore, decreases the number of VCP version 2 cycles needed for traceback. Overall, there is no major improvement in the processing delay because a larger portion of cycles is spent in state metric accumulation, which is largely unaffected by the choice of *R* and *C*.

- Hard decisions:
  - (*R+C) max* for mixed/convergent processing for hard decisions has been increased from 605 to 635 for *K = 6*, and from 1020 to 2044 for *K = 5*. As with soft decisions, this change results in a small decrease in VCP version 2 cycle counts during traceback.

The procedure to calculate the reliability length is as follows. (The reliability length cannot be larger than *1920, or C ≤ 1920-(k-1)*).

1. Determine the convergence length
2. Determine the number of sliding windows: *Nsw = ceil(F/((R+ C)max - C)*
3. Determine the reliability length: *R = m x ceil(F/(Nsw × m))*
4. If *R > 248*, then increment number of sliding windows and repeat step 3
5. See Table 30-7 and Table 30-8 for hard decision and soft decision limits

Table 30-7. Traceback Soft Decision Sliding Window Limits

K	Maximum Frame Size	Fmax	(r + c)max	Multiple (m)	Convergent Multiplier (N)	Special Constraint
9	64	56	60	7	3, 6	
8	112	105	105	4	3, 6, 9, 12	
7	192	186	186	6	3, 6, 9, 12, 15, 18	
6	320	315	315	5	3, 6, 9, 12, 15, 18	r ≤ 248
5	1024	1020	320	4	3, 6, 9, 12, 15, 18	r ≤ 248

Table 30-8. Traceback Hard Decision Sliding Window Limits

K	Maximum Frame Size	Fmax	(r + c)max	Multiple (m)	Convergent Multiplier (N)
9	128	120	124	4	3, 6, 9, 12, 15
8	224	217	217	7	3, 6, 9, 12, 15, 18
7	384	378	378	6	3, 6, 9, 12, 15, 18
6	640	635	635	5	3, 6, 9, 12, 15, 18
5	2048	2044	2044	4	3, 6, 9, 12, 15, 18

30.3.4.1.5 Yamamoto Parameters

During the standard forward recursion, an entity called the Yamamoto bit is computed for each state and updated every symbol interval. The Yamamoto bit was proposed by Hirosuke Yamamoto (Hirosuke Yamamoto, "Viterbi Decoding Algorithm for Convolutional Codes with Repeat Request," IEEE Transactions on Information Theory, Vol. IT-26, No. 5, September 1980).

Basically, a bit (the Yamamoto bit) is associated with each state in the decoding process. Initially, all the Yamamoto bits are set (1). During the decoding process, the Yamamoto bit for a particular state comes from a couple of decisions made on the path metrics and the Yamamoto bit of previous states. The metrics of all paths leading to a particular state are compared. If the difference between any two metrics is less than a given threshold VCP_VCPIC1[27:16] YAMT bits, then the Yamamoto bit is cleared otherwise, the Yamamoto bit is inherited from the previous state of the path with the largest metric. The end result of this process (VCP_VCPOUT1[16] YAM) yields a zero (0) if anywhere along the decoding path there was a point where the decision between two paths was ambiguous. The VCP_VCPOUT1[16] YAM bit can therefore be used as a binary frame quality indicator.

The Yamamoto algorithm can be enabled or disabled by toggling the VCP_VCPIC1[28] YAMEN bit.

30.3.4.1.6 Input FIFO (Branch Metrics)

Figure 30-10 shows how the FIFO is used in either a double or quad buffering scheme. The VCP modules generate a VCPXEVT synchronization event each time the *nx1/4* of the buffer is empty, where (*n =1, 2, 3, 4*) or (*n = 2, 4*) depending on the value of VCP_VCPIC5[19:16] SYMX. The VCP_VCPIC5[19:16] SYMX bits define the buffer length as well as the VCPnXEVT event rate. The maximum allocatable size for the buffer in the input FIFO is 16 or 64-bit words. VCP_VCPIC5[19:16] SYMX defines the branch metric buffer length (minus one) to be transmitted to the VCP for each VCPnXEVT. Table 30-9 has the valid values for SYMX, along with the corresponding number of 64-bit transfers. If the number of 64 bit transfers is 16, then the double buffering scheme is used and VCPnXEVTs will be generated when half the FIFO is empty (either top or bottom half). If the number of 64-bit transfers is 8, then a quad buffering scheme is used and VCPnXEVTs will be generated as each quarter of the FIFO is emptied. The VCP version 2 only generates as many VCPXEVTs as needed to produce all the BMs required based on the SYMX value, Frame Length, and Rate. In other words, no potential excess VCPnXEVTs will be generated based on the FIFO being partially empty at the end of processing.

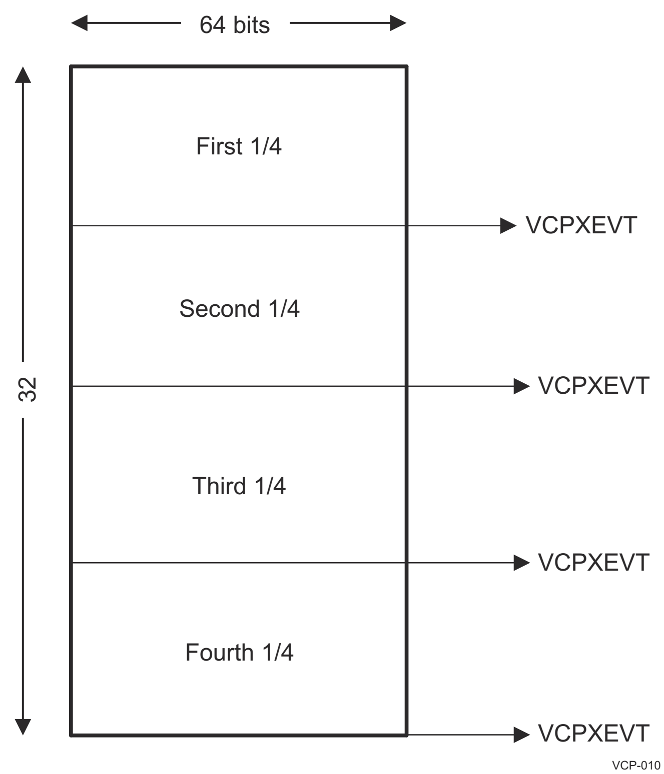


Figure 30-10. Input FIFO (Branch Metrics)

Table 30-9. Traceback Hard Decision Sliding Window Limits

Code Rate	SYMX	Number of 64-bit Transfer
1/4	3	16
1/4	1	8
1/3	7	16
1/3	3	8
1/2	15	16
1/2	7	8

30.3.4.1.7 Output FIFO (Decisions)

Figure 30-11 shows how the output FIFO is used in either a double or quad buffering scheme. The VCP modules generate a VCPREVT synchronization event each time the *nx1/4* of the buffer is empty, where (*n =1, 2, 3, 4*) or (*n = 2, 4*), depending on the value of SYMR. VCP_VCPIC5[24:20] SYMR bits define the buffer length, as well as the VCPnREVT event rate. The maximum size for buffer is 16 64-bit words.

- For hard-decisions:
  - If *F ≤ 2048*, then *SYMR = ceil F/64]-1*
  - If *F > 2048*, then *SYMR = 15 or 31*
- For soft-decisions:
  - If *F ≤ 256*, then *SYMR = ceil [F/8]-1*
  - If *F > 256*, then *SYMR = 15 or 31*

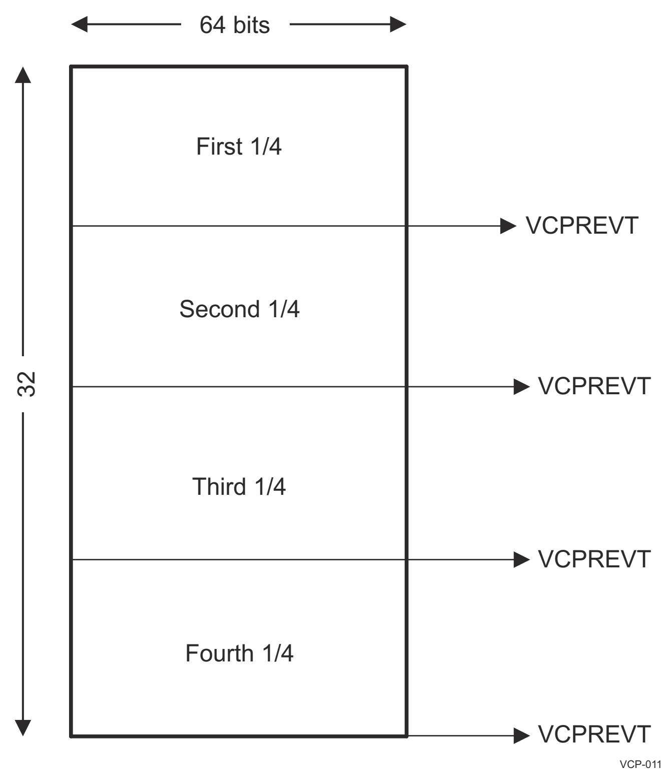


Figure 30-11. Output FIFO (Decisions Data)

If VCP_VCPIC5[23:20] SYMR = 31, double buffering is used and a VCPnREVT is generated when the first or second half of the buffer is filled. If SYMR = 15, then VCPnREVT is generated after each quarter of the buffer is filled. When *F ≤ 2048* for HD or *F ≤ 256* for SD outputs and SYMR is determined as described above, a single VCPnREVT is generated once all decisions have been written to the output FIFO.

30.3.5 VCP Input Data

30.3.5.1 Branch Metrics Calculations

The branch metrics (BM) are calculated by the DSP and stored in the DSP memory subsystem as 8-bit signed values. Per symbol interval T, for a rate *R = k/n* and a constraint length *K*, there are a total of $2^{(K-1+k)}$ branches in the trellis. For rate 1/n codes, only $2^{(n-1)}$ branch metrics need to be computed per symbol period and passed to the VCP module. Moreover, n symbols are required to calculate 1 branch metric.

Assuming BSPK modulated bits (0 → 1, 1→ -1), the branch metrics are calculated as follows:

- Rate 1/2: there are 2 branch metrics per symbol period

– $BM_0(t) = r_0(t) + r_1(t)$
– $BM_1(f) = r_0(t) - r_1(t)$

where r(t) is the received codeword at time t (2 symbols, $r_0(t)$ is the symbol corresponding to the encoder upper branch, see Figure 30-4).

- Rate 1/3: there are 4 branch metrics per symbol period
  – $BM_0(t) = r_0(t) + r_1(t) + r_2(t)$
  – $BM_1(t) = r_0(t) + r_1(t) - r_2(t)$
  – $BM_2(t) = r_0(t) - r_1(t) + r_2(t)$
  – $BM_3(t) = r_0(t) - r_1(t) - r_2(t)$

where r(t) is the received codeword (3 symbols, $r_0(t)$ is the symbol corresponding to the encoder upper branch; see Figure 30-4).

- Rate 1/4: there are 8 branch metrics per symbol period
  – $BM_0(t) = r_0(t) + r_1(t) + r_2(t) + r_3(t)$
  – $BM_1(t) = r_0(t) + r_1(t) + r_2(t) - r_3(t)$
  – $BM_2(t) = r_0(t) + r_1(t) - r_2(t) + r_3(t)$
  – $BM_3(t) = r_0(t) + r_1(t) - r_2(t) - r_3(t)$
  – $BM_4(t) = r_0(t) - r_1(t) + r_2(t) + r_3(t)$
  – $BM_5(t) = r_0(t) - r_1(t) + r_2(t) - r_3(t)$
  – $BM_6(t) = r_0(t) - r_1(t) - r_2(t) + r_3(t)$
  – $BM_7(t) = r_0(t) - r_1(t) - r_2(t) - r_3(t)$

where r(t) is the received codeword (4 symbols, $r_0(t)$ is the symbol corresponding to the encoder upper branch, see Figure 30-4).

The data must be sent to the VCP version 2 as described in Table 30-10, Table 30-11, and Table 30-12 for rates 1/2, 1/3, and 1/4, respectively (the base address must be double-word aligned).

The branch metrics can be saved in the DSP memory subsystem in either its native format or packed in words (user implementation). When working in big-endian mode, the VCP endian mode register (VCP_VCPEND) indicates if the data is 32-bit word packed or Native 8-bit format.

Table 30-10. Branch Metrics for Rate 1/2

Address (hex)	Data			
	MSB			LSB
Base	BM1(t=T)	BM0(t=T)	BM1(t=0)	BM0(t=0)
Base + 4h	BM1(t=3T)	BM0(t=3T)	BM1(t=2T)	BM0(t=2T)
Base + 8h	...			

Table 30-11. Branch Metrics for Rate 1/3

Address (hex)	Data			
	MSB			LSB
Base	BM3(t=0)	BM2(t=0)	BM1(t=0)	BM0(t=0)
Base + 4h	BM3(t=T)	BM2(t=T)	BM1(t=T)	BM0(t=T)
Base + 8h	...			

Table 30-12. Branch Metrics for Rate 1/4

Address (hex)	Data			
	MSB			LSB
Base	BM3(t=0)	BM2(t=0)	BM1(t=0)	BM0(t=0)
Base + 4h	BM7(t=0)	BM6(t=0)	BM5(t=0)	BM4(t=0)
Base + 8h	BM3(t=T)	BM2(t=T)	BM1(t=T)	BM0(t=T)
Base + Ch	BM7(t=T)	BM6(t=T)	BM5(t=T)	BM4(t=T)

Table 30-12. Branch Metrics for Rate 1/4 (continued)

Address (hex)	Data
Base + 10h	...

The state metric accumulation resolution is 13 bits on the VCP modules. Consequently, full 8-bit dynamic range is available for branch metrics on the VCP modules, for all constraint lengths and all code rates.

30.3.6 Soft Input Dynamic Ranges

The VCP implementation implies that the soft inputs need to be quantized so that the branch metrics satisfy the following bound B1 (branch metrics upper bound - absolute value):

$$2(C-1) -1 \geq (2\times(K-1) + 2) \times B1 \quad (12)$$

K is the constraint length and C determines the truncation of state metrics that can be performed without loss of decoding performance.

The VCP is designed with C = 13. The branch metrics can have a maximum dynamic range of 7+1 sign bits do [-128; +127]. This gives another branch metrics, upper bound B2 ≤ 128.

So for a given constraint length, min(B1, B2) will give the final branch metrics, maximum bound B.

To satisfy B in the branch metrics calculation, the soft input values, delivered as 8-bit-signed equalized values, are linearly scaled with the following formula where *1/n* is the rate.

$$Scaled = min(B1, B2)/n \times SoftValue/128 \quad (13)$$

Example

K=9, then B1 3 227.5 and the branch metrics range B2 is [-128; +127]. So the branch metrics need to be in [-128; +127] range.

If the rate is 1/3, then 128/3 ≈ 42, so the soft inputs need to be scaled by a factor 0.333333 and saturated within the range [-42; +42].

Table 30-13 summarizes the calculations for the different constraint length and rate:

Table 30-13. VCP Soft Inputs Quantization

1/Rate	K	Scaling Factor	Range
2	5, 6, 7, 8, 9	0.5	[-64; +63]
3	5, 6, 7, 8, 9	0.333333	[-42; +42]
4	5, 6, 7, 8, 9	0.25	[-31; +31]

30.3.7 VCP Memory Sleep Mode

All VCP memories can be put into a low leakage (sleep) state. Putting the banks into a sleep mode disables the memories. The VCP1/2 can put the banks into dynamic sleep mode if the input control bits VCP_VCPEND[8] SLPZVDD and VCP_VCPEND[9] SLPZVSS in the endianness register are set. Once the bits are set, VCP memories will be put in the sleep mode based on the following truth table Table 30-14:

Table 30-14. VCP Memory Sleep Mode Truth Table

RAM	Conditions
sd	If (sdhd = 0), bank is in sleep mode. Else, bank wake up.
sm	If (exc_cmd = 0), bank is in sleep mode. Else, bank wake up.

30.3.8 Decision Data

The VCPs can be configured to generate either hard decisions (one bit per decision), or soft decisions (8-bit value per decision). Ordering of the VCP decisions depends on the VCP_VCPIC3[28] OUT_ORDER bit-field and the VCP_VCPEND[1] SD bit, if the DSP is set to work in big-endian mode and the results or Soft Decisions (see the VCP version 2 endian mode register VCP_VCPEND). The decisions buffer start address must be double-word aligned and the buffer size must be a multiple of 8 bytes.

The soft decisions in the VCP are initially computed with the path metrics at 13-bit values. The results are then clipped to 8-bit signed integer values before being stored in the traceback soft decision memory.

30.3.9 Endianness

The VCP endian mode register VCP_VCPEND is intended to solve possible little-endian issues.

30.3.9.1 Branch Metrics

When the data are saved in their 8-bit native format (*BM = 1*), they must be organized in the DSP memory as described in Table 30-15. When the data are packed on 32-bit words (BM = 0), they must be organized in the DSP memory as described in Table 30-16.

Table 30-15. Branch Metrics

Little_big_endian	BM	Description (MSB to LSB)
0	0	3, 2, 1, 0, 7, 6, 5, 4 => 7, 6, 5, 4, 3, 2, 1, 0 (bytes)
0	1	0, 1, 2, 3, 4, 5, 6, 7 => 7, 6, 5, 4, 3, 2, 1, 0 (bytes)
1	0	Endianness manager has no effect 7, 6, 5, 4, 3, 2, 1, 0 => 7, 6, 5, 4, 3, 2, 1, 0 (bytes)
1	1	Endianness manager has no effect 7, 6, 5, 4, 3, 2, 1, 0 => 7, 6, 5, 4, 3, 2, 1, 0 (bytes)

Table 30-16. Branch Metrics in DSP Memory (BM = 1)

Address (hex bytes)	Data
Base	BM0
Base + 1	BM1
Base + 2	BM2
Base + 3	BM3
Base + 4	BM4
Base + 5	BM5
Base + 6	BM6
Base + 7	BM7

The data are presented to the EDMA as shown in Figure 30-12. The endianness manager will reorder the BMs as shown in Figure 30-13 for processing.

Figure 30-12. Data Source - EDMA (BM = 1)

63 - 56	55 - 48
BM0	BM1

47 - 40	39 - 32
BM2	BM3

31 - 24	23 - 16

BM4	BM5

15	8	7	0
BM5		BM7	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Figure 30-13. Data Destination - Kernel for Processing Unit (BM = 1)

63	56	55	48
BM7		BM6	

47	40	39	32
BM5		BM4	

31	24	23	16
BM3		BM2	

15	8	7	0
BM1		BM0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30-17. Branch Metrics in DSP Memory (BM = 0)

Address (hex bytes)	Data
Base	BM3
Base + 1	BM2
Base + 2	BM1
Base + 3	BM0
Base + 4	BM7
Base + 5	BM6
Base + 6	BM5
Base + 7	BM4

The data are presented to the EDMA as shown in Figure 30-14. The Endianness manager reorder the BMs as shown in Figure 30-15 for processing

Figure 30-14. Data Source - EDMA (BM = 0)

63 — 56	55 — 48
BM3	BM2

47 — 40	39 — 32
BM1	BM0

31 — 24	23 — 16
BM7	BM6

15 — 8	7 — 0
BM5	BM4

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Figure 30-15. Data Destination - Kernel for Processing Unit (BM = 0)

63 — 56	55 — 48
BM7	BM6

47 — 40	39 — 32
BM5	BM4

31 — 24	23 — 16
BM3	BM2

15 — 8	7 — 0
BM1	BM0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

30.3.9.1.1 Hard Decisions

The VCP modules hard decisions bit-ordering within the 32-bit hard decision word, is programmable via the VCP_VCPIC3[28] OUT_ORDER bit, such that the oldest bit can be either in the MSB or the LSB position.

Figure 30-16. Trellis Stage Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 0)

63	62	...	32	31	...	1	0
Stage N	Stage N - 1	...	Stage N - 31	Stage N - 32	...	Stage N - 62	Stage N - 63

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

OUT_ORDER = 1 orders the hard decision data in the order in which it is calculated in the State Metric computation; from 31 to 0 in the 32-bit word output.

Figure 30-17. Trellis Stage Ordering of Hard Decisions in 32-Bit Word (OUT_ORDER = 1)

63	62	...	32	31	...	1	0
Stage N - 31	Stage N -30	...	Stage N	Stage N - 63	...	Stage N - 33	Stage N - 32

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

30.3.9.1.2 Soft Decisions

The VCP soft decisions are 8-bit results, and output 64 bits at a time. The soft decisions will be organized as shown in Table 30-18, based on the CPU's endianness and whether SD is set for Native or 32-bit packed results.

Table 30-18. Soft Decision Organization

Little_big_endian	BM	Description (MSB to LSB)
0	0	7, 6, 5, 4, 3, 2, 1, 0 => 3, 2, 1, 0, 7, 6, 5, 4 (bytes)
0	1	7, 6, 5, 4, 3, 2, 1, 0 => 0, 1, 2, 3, 4, 5, 6, 7 (bytes)
1	0	Endianness manager has no effect 7, 6, 5, 4, 3, 2, 1, 0 => 7, 6, 5, 4, 3, 2, 1, 0 (bytes)
1	1	Endianness manager has no effect 7, 6, 5, 4, 3, 2, 1, 0 => 7, 6, 5, 4, 3, 2, 1, 0 (bytes)

30.3.10 VCP Output Parameters

The VCP_VCPOUT0[12:0] FMAXS and VCP_VCPOUT0[28:16] FMINS bits indicate the final maximum and minimum state metric values, respectively. The VCP_VCPOUT1[7:0] FMAXI bit indicates the state index for the state with the final maximum state metric.

The VCP_VCPOUT1[16] YAM is described in Section 30.3.4.1.5 *Yamamoto Parameters*.

30.3.11 Event Generation

30.3.11.1 VCPnXEVT Generation

A VCP modules transmit events (VCP1XEVT and VCP2XEVT) are generated when any of the following conditions appear:

- A START command write in VCP_VCPEXE[3:0] COMMAND.
- Top one-fourth and half or bottom one-fourth and half of the input FIFO buffer (see Table 30-9) is empty.
- After all decisions have been read and VCP_VCPIC5[30] OUTF is cleared, or if all decisions have been read and the output registers have been read and VCP_VCPIC5[30] OUTF is set.

Note

The extra XEVT pulls the next set of input configurations via EDMA. If EDMA controller is not set up to link in another set of input configurations, then a dummy transfer should be set up to avoid an SER Event Flag being set for the EDMA Parameter Entry. If the flag is set, it effectively locks the VCP modules, and must be cleared before any future events can be processed for that entry.

30.3.11.2 VCPnREVT Generation

The VCP modules receive event (VCP1REVT and VCP2REVT) are generated when any of the following conditions appear:

- The traceback unit has written top one fourth and half or bottom one-fourth and half of the output FIFO buffer (see Figure 30-11).
- After the traceback is completed (the whole frame has been decoded).
- VCP_VCPIC5[30] OUTF bit is set to 1 and all decisions have been read to read the output registers.

30.3.12 Operational Modes

The following list describes the VCP1 and VCP2 modules operational modes.

- 0: Value at reset or value written by the coprocessor when previous instruction is read and its execution is ongoing. DSP may test the status word in the output control memory to check if the instruction is being executed.
- 1: Start - CPU orders the coprocessor to start a processing block: The first action of the coprocessor is then to generate the first XEVT to trigger DMA transfer of the input control words.
- 2: Pause - CPU orders the coprocessor to pause a processing block at the beginning of traceback.
- 3: Unpause (single_traceback) - CPU orders the coprocessor to restart at the beginning of traceback and halt at next traceback.
- 4: Unpause (finish_traceback) - CPU orders the coprocessor to restart at the beginning of traceback and complete decode.
- 5: Stop - CPU orders the coprocessor to reset: the coprocessor resets all VCP modules registers.

30.3.12.1 Debugging Features

Visibility into the internal operation of the VCP version 2 (i.e., the state metric accumulation, traceback memory) is available to the CPU via a pause command. However, since the pause command is not synchronized with the internal VCP version 2 state machine but is rather sent from the CPU at a random moment in time, this feature is of limited use.

The pause command on the VCP version 2 is augmented to provide visibility into VCP modules operation on a sliding window basis. Instead of using the normal start command which tells the VCPs to perform a complete decode of one frame (including input/output transfers via EDMA controller), halt at beginning of traceback and resume until next traceback commands are used, and the internal VCP memories can be inspected at various points in the decoding process. The procedure for using this command is as follows:

- VCP modules configuration and branch metrics are prepared
- A halt at the beginning of the traceback command is sent
- The VCP modules generates necessary interrupts to the EDMA to transfer input configuration and to start transferring branch metrics. The VCP1 and VCP2 perform state metric accumulation as branch metrics become available. When it reaches the end of the first sliding window (i.e., the reliability portion and the convergence portion), the VCP halts.
- The CPU polls the VCP1 and VCP2 status registers until the VCP's states changes from running to paused. At that point, the state metrics memory can be inspected, as well as the traceback memory. To perform an inspection, halt the CPU via a software breakpoint set at an appropriate point in the code, for instance. Then, the memory can be inspected visually via the debugger GUI, or, alternatively, the CPU can copy the relevant internal VCP memories to another location for later analysis.
  - The CPU sends the resume until next traceback command to the VCP.

– The VCPs perform the traceback, generate a portion of hard or soft decisions, and continue with state metric accumulation until the end of the next sliding window (i.e., another number of R stages, where R is the reliability length).
– The process continues until the decoding is complete. Alternatively, the decoding process can be run to completion after any sliding window by sending the resume to completion command instead of the resume until next traceback command.

Emulation modes are achieved with the programmable SOFT and FREE bits in the VCP emulation control registers VCP_VCPEMU.

When the CPU is halted during emulation, VCPs can be halted based on the setting of the VCP_VCPEMU[1] SOFT and VCP_VCPEMU[0] FREE bits. The VCPs can be halted at the traceback processing or frames processing boundary.

30.3.13 Errors and Status

To handle errors, the IRQ status must be enabled. The IRQ status is enabled by writing a 1 to the VCP_IRQENABLE_SET[0] ENABLE_SET bit. When the coprocessor detects an error, the coprocessor sets the status and error words, then sends an interrupt to the CPU. Any coprocessor processing is paused and the DSP must reset or start the coprocessor. An error occurs if the VCP modules receive an invalid value in the input configuration parameters. If an error is detected, the VCP_VCPERR bit-fields are set accordingly, the VCP_VCPSTAT0[2] ERR bit is set, the VCP_INT interrupts are generated, and no processing are engaged. In order to restart the VCPs the following steps must be taken:

1. Read the VCP_IRQSTATUS register
2. Read the VCP_VCPERR register
3. Write to the VCP_IRQSTATUS register the previously read value to specify that this interrupt had been handled.
4. Send another START command by VCP_VCPEXE[3:0] COMMAND.

The status registers are provided for debugging purposes and are best used when either the processor is halted or the VCP modules are halted. If an error occurs, the VCPs are halted and a VCP_INT interrupts are generated that can be mapped to a CPU interrupt. There may be cases where the status registers must be viewed when the VCP modules are still running. One such case is when the VCPs seem to have taken a long time in processing the current frame. In such cases, a watchdog timer should be used and set according to the frame length and VCPs configuration, in addition to some overhead to allow for EDMA usage.

Table 30-19 through Table 30-21 summarize the cases, when no errors can occur in register VCP_VCPERR.

Table 30-19. E_SYMR 0 (No error for SMAR)

TB Mode	VCP_VCPIC5[24:20] SYMR	# of 64 BitTransfers	ConditionTransfer
Hard	0 to 31	1 to 32	F ≤ 2048
Hard	31	32	F > 2048
Hard	15	16	F ≥ 2048
Soft	0 to 31	1 to 32	F ≤ 256
Soft	31	32	F ≥ 256
Soft	15	16	F > 256

Table 30-20. E_SYMX 0 (No error for SMAX)

Code Rate	VCP_VCPIC5[19:16] SYMX	# of 64-Bit	# of BM per	# trellis per
1/4	3	16	128	16
1/4	1	8	64	8
1/3	7	16	128	32
1/3	3	8	64	16
1/2	15	16	128	64
1/2	7	8	64	32

Table 30-21. MAXIMINERR 0 (Error Check)

Sign Imaxas	Sign Imins	Check Equation(1)
0	0	0 ≤ IMAXS - IMINS < 2048
1	1	0 ≤ IMAXS - IMINS < 2048
0	1	0 ≤ IMAXS - IMINS < 2048
1	0	0 ≤ (2 x 4096 + IMAXS) - IMINS < 2048

(1) See the bit-fields in register VCP_VCPIC4[28:16] IMINS and VCP_VCPIC4[12:0] IMAXS.

30.4 VCP Modules Programming Guide

The VCP1 and VCP2 require setting up the following context per user channel:

- Three to four EDMA parameters (see Table 30-22)
- The input configurations parameters

Several user channels can be programmed prior to starting the VCPs. A suggested implementation is to use the EDMA interrupt generation capabilities (refer to Section 16.2.4.9 *EDMA Interrupts* in Section 16.2 *Enhanced DMA controller*) and program the EDMA to generate an interrupt after the user channel's last VCPnREVT synchronized EDMA transfer has completed.

Table 30-22. Required EDMA Links Per User Channel

Direction	Data	Usage	Required/Optional
Transmit	Input configuration parameters	Send the input configuration parameters	Required
Transmit	Branch metrics	Send branch metrics	Required
Receive	Decisions	Read decisions	Required
Receive	Output parameters	Read output parameters	Optional (OUTF bit)

30.4.1 EDMA Resources

30.4.1.1 VCP1 and VCP2 Dedicated EDMA Resources

Within the available EDMA channel event sources, four are assigned to the VCP1 and VCP2 modules: event 0 and event 1.

- Event 0 is associated to the VCP1 and VCP2 receive events (VCP1REVT and VCP2REVT) and are used as the synchronization events for EDMA transfers from the VCP1 and VCP2 to the DSP (receive).
- Event 1 is associated to the VCP1 and VCP2 transmit event (VCP1XEVT and VCP2XEVT) and are used as the synchronization events for EDMA transfers from the DSP to the VCP1 and VCP2 (transmit).

For the default configuration of the crossbar, the VCP events are event 0 and event 1. If the crossbar is reconfigured, event 0 and event 1 must be programmed to the VCP events.

30.4.1.2 Special VCP EDMA Programming Considerations

The EDMA parameters consist of eight words as shown in Figure 30-18. All EDMA transfers, in the context of the VCPs, must contain an even number of words, and have source and destination addresses double-word aligned.

All EDMA transfers must be double-word aligned and the transfer size for the VCP EDMA transfers must be a multiple of 8. Single-word transfers that are not double-word aligned cause errors in the TCP2/VCP1/VCP2 memory.

Transfers to the VCP FIFOs must be configured to NOT use the EDMA FIFO/Const mode. Instead, the transfers should be set with:

- ACNT=8;
- BCNT=transfer_byte_count/8;
- BIDX=0(via SBIDX or DBIDX);
- ABSync trigger mode (via SYNCDIM = 1);

For more information about EDMA, refer to Section 16.2 *Enchanced DMA controller*.

Figure 30-18. EDMA Parameters Structure

31	0
EDMA Channel Options Parameter (OPT)	
EDMA channel Source Address (SRC)	
Number of arrays of length ACNT (BCNT)	Number of bytes in array (ACNT)
EDMA Channel Destination Address (DST)	
Destination 2nd Dimension Index (DBIDX)	Source 2nd Dimension Index (SBIDX)
BCNTRLD	LINK
Destination 3rd Dimension Index (DCIDX)	Source 3rd Dimension Index (SCIDX)
Reserved	Number of frames in block (CCNT)

30.4.1.2.1 Input Configuration Parameters Transfer

Note

The procedures described in this chapter concern the configuration of EDMA module in the device. For more information about EDMA module refer to Section 16.2 *Enhanced DMA controller*.

The procedure for configuring the EDMA transfer to the input configuration parameters is a 6-word VCPXEVT frame synchronized transfer. The parameters should be set in several EDMA registers (refer to Section 16.2.7 *EDMA Register Manual*), as described in Table 30-23:

Table 30-23. EDMA Transfer Options Parameters

Step	Register/Bit Field/Programming Model	Value
Intermediate transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[23] ITCCHEN	0x0: Disable
Transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[22] TCCHEN	0x0: Disable
Intermediate transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[21] ITCINTEN	0x0: Disable
Backward compatibility mode	EDMA.EDMA_TPCC_OPT_n[19] WIMODE	0x0: Normal operation
Transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[20] TCINTEN	0x0: Disable
Transfer complete code	EDMA.EDMA_TPCC_OPT_n[17:12] TCC	1 to 63
Transfer complete code mode	EDMA.EDMA_TPCC_OPT_n[11] TCCMODE	0x0: Normal operation
FIFO width	EDMA.EDMA_TPCC_OPT_n[10:8] FWID	x
Static Entry	EDMA.EDMA_TPCC_OPT_n[3] STATIC	0x0: Entry is updated as normal
Transfer Synchronization Dimension	EDMA.EDMA_TPCC_OPT_n[2] SYNCDIM	0x0: A-sync transfer, each event triggers the transfer of ACNT elements.
Destination Address Mode	EDMA.EDMA_TPCC_OPT_n[1] DAM	0x0: Destination address within an array increments. Destination is not a FIFO.
Source Address Mode	EDMA.EDMA_TPCC_OPT_n[0] SAM	0x0: Source Address within an array increments. Source is not a FIFO.
Source Address	EDMA.EDMA_TPCC_SRC_n[31:0] SRC	xxxx: User input configuration parameters start address.
Number of bytes in an array	EDMA.EDMA_TPCC_ABCNT_n[15:0] ACNT	24 - Number of bytes in 1st Dimension.
Number of arrays in a frame	EDMA.EDMA_TPCC_ABCNT_n[15:0] BCNT	0x1: One array in the frame. Count for 2nd Dimension.

Table 30-23. EDMA Transfer Options Parameters (continued)

Step	Register/Bit Field/Programming Model	Value
Destination Address	EDMA.EDMA_TPCC_DST_n[31:0] DST	VCP_VCPIC0 - 32-bit destination address parameters.
Source 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[15:0] SBIDX	0x00
Destination 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[31:16] DBIDX	0x00
Source Frame Index	EDMA.EDMA_TPCC_CIDX_n[15:0] SCIDX	0x00
Destination Frame Index	EDMA.EDMA_TPCC_CIDX_n[31:16] DCIDX	0x00
C byte count. Count for 3rd Dimension	EDMA.EDMA_TPCC_CCNT_n[15:0] CCNT	0x1: One frame in the block

Upon completion, this EDMA transfer is linked to the EDMA for branch metrics transfer parameters.

30.4.1.2.2 Branch Metrics Transfer

Table 30-24 shows the sequence for configuring EDMA transfer to the branch metrics FIFO is a VCP1XEVT and VCP2XEVT frames synchronized transfer. The parameters should be set as:

Table 30-24. Configuration EDMA Transfer to the Branch Metrics FIFO

Step	Register/Bit Field/Programming Model	Value
Intermediate transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[23] ITCCHEN	0x0: Disable
Transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[22] TCCHEN	0x0: Disable
Intermediate transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[21] ITCINTEN	0x0: Disable
Transfer complete interrupt is enabled	EDMA.EDMA_TPCC_OPT_n[20] TCINTEN	0x1: Enabled
Backward compatibility mode	EDMA.EDMA_TPCC_OPT_n[19] WIMODE	0x0: Normal operation
Transfer complete code	EDMA.EDMA_TPCC_OPT_n[17:12] TCC	1 to 63
Transfer complete code mode	EDMA.EDMA_TPCC_OPT_n[11] TCCMODE	0x0: Normal completion
FIFO width	EDMA.EDMA_TPCC_OPT_n[10:8] FWID	x
Static Entry	EDMA.EDMA_TPCC_OPT_n[3] STATIC	0x0: Entry is updated as normal.
Transfer Synchronization Dimension	EDMA.EDMA_TPCC_OPT_n[2] SYNCDIM	0x1: AB-Sync, Each event triggers the transfer of BCNT arrays of ACNT elements.
Destination Address Mode	EDMA.EDMA_TPCC_OPT_n[1] DAM	0x0: INCR, Dst addressing within an array increments. Dst is not a FIFO.
Source Address Mode	EDMA.EDMA_TPCC_OPT_n[0] SAM	0x0: Source Address within an array increments. Source is not a FIFO.
Source Address	EDMA.EDMA_TPCC_SRC_n[31:0] SRC	xxxx: Branch Metrics Array start addressBranch Metrics Array start address.
Number of branch metrics bytes in an array	EDMA.EDMA_TPCC_ABCNT_n[15:0] ACNT	0x8, where $ACNT = 4 \times (SYMX+1) \times 2^{(r-1)}$ (14)

Table 30-24. Configuration EDMA Transfer to the Branch Metrics FIFO (continued)

Step	Register/Bit Field/Programming Model	Value
Number of arrays in a frame Where TNBM is Total Number of Branch Metrics in Bytes To calculate total number of branch metrics data in bytes: *Total number of Branch metrics = (F + K - 1) ×(2*$^{(r-1)}$*)* for mixed and tailed traceback mode In case of convergent mode, *Total number of Branch metrics = (F + C) ×(2*$^{(r-1)}$*)*	EDMA.EDMA_TPCC_ABCNT_n[15:0] BCNT	*BCNT = CEIL(TNBM/ ACNT)* (15)
Destination Address	EDMA.EDMA_TPCC_DST_n[31:0] DST	VCP_VCPWBM Branch Metrics FIFO Address - 32-bit destination address parameters.
Source 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[15:0] SBIDX	= ACNT (to create incrementing transfer)
Destination 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[31:16] DBIDX	0x00 (to access the FIFO)
Source Frame Index	EDMA.EDMA_TPCC_CIDX_n[15:0] SCIDX	ACNT × BCNT
Destination Frame Index	EDMA.EDMA_TPCC_CIDX_n[31:16] DCIDX	0x00 (to access the FIFO)
C byte count. Count for 3rd Dimension	EDMA.EDMA_TPCC_CCNT_n[15:0] CCNT	0x1: One frame in the block

Upon completion, this EDMA transfer is linked to one of the following:

- The DMA input configuration parameters transfer parameters of the next user-channel, if there is one ready to be decoded.
- Dummy DMA transfer parameters, if there are no more user channels ready to be decoded.

Note

Do not link to a NULL transfer, as the Secondary Event Register will set the event flag for Event 1. The final VCPnXEVT is generated upon the reading of the decisions and output registers, which is intended to transfer the input configuration of the next user channel. If a NULL transfer link is in place, the final VCPnXEVT will set the event 1 flag of SER and no further VCP execution will occur until it is cleared.

30.4.1.2.3 Decisions Transfer

EDMA transfers from the decision buffer are VCP1REVT and VCP2REVT frame synchronized transfers. The programming of these transfers depend on the decision type and the traceback mode.

Upon completion, this EDMA transfer is linked to one of the following:

1. The decisions EDMA transfer parameters of the next user-channel, if there is one ready to be decoded and the VCP_VCPIC5[30] OUTF bit is 0.
2. Null EDMA transfer parameters (with all zeros), if there are no more user-channels ready to be decoded and the VCP_VCPIC5[30] OUTF bit is 0.
3. The output parameters EDMA transfer parameters, if the VCP_VCPIC5[30] OUTF bit is 1.

30.4.1.2.4 Hard-Decisions Mode

Table 30-25 shows the sequence for configuring EDMA transfer for Hard-Decisions Mode. The parameters should be set as:

Table 30-25. Configuration EDMA Hard-Decisions Mode

Step	Register/Bit Field/Programming Model	Value
Intermediate transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[23] ITCCHEN	0x0: Disable
Transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[22] TCCHEN	0x0: Disable
Intermediate transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[21] ITCINTEN	0x0: Disable

Table 30-25. Configuration EDMA Hard-Decisions Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[20] TCINTEN	0x0: Disable
Backward compatibility mode	EDMA.EDMA_TPCC_OPT_n[19] WIMODE	0x0: Normal operation
Transfer complete code	EDMA.EDMA_TPCC_OPT_n[17:12] TCC	1 to 63
Transfer complete code mode	EDMA.EDMA_TPCC_OPT_n[11] TCCMODE	0x0: Normal completion
FIFO width	EDMA.EDMA_TPCC_OPT_n[10:8] FWID	x
Static Entry	EDMA.EDMA_TPCC_OPT_n[3] STATIC	0x0: Entry is updated as normal.
Transfer Synchronization Dimension	EDMA.EDMA_TPCC_OPT_n[2] SYNCDIM	0x1: AB-Sync, Each event triggers the transfer of BCNT arrays of ACNT elements.
Destination Address Mode	EDMA.EDMA_TPCC_OPT_n[1] DAM	0x0: NCR, Dst addressing within an array increments. Dst is not a FIFO.
Source Address Mode	EDMA.EDMA_TPCC_OPT_n[0] SAM	0x0: INCR, Src addressing within an array increments.Source is not a FIFO.
Source Address	EDMA.EDMA_TPCC_SRC_n[31:0] SRC	xxxx: VCPRDECS Decision FIFO address
Number of Hard decision bytes in an array	EDMA.EDMA_TPCC_ABCNT_n[15:0] ACNT	0x8 $$A(SYMR+1) \times 8 \quad (16)$$
Number of arrays in a frame Where TNHD is the total number of hard decisions in bytes (Framelength/8)	EDMA.EDMA_TPCC_ABCNT_n[15:0] BCNT	$$BCNT = CEIL(TNHD/ACNT) \quad (17)$$
Destination Address	EDMA.EDMA_TPCC_DST_n[31:0] DST	xxxx: Hard Decision Array Address
Source 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[15:0] SBIDX	0x8 (to create incrementing transfer)
Destination 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[31:16] DBIDX	0x0 (to access the FIFO)
Source Frame Index	EDMA.EDMA_TPCC_CIDX_n[15:0] SCIDX	ACNT × BCNT
Destination Frame Index	EDMA.EDMA_TPCC_CIDX_n[31:16] DCIDX	0x00 (to access the FIFO)
C byte count. Count for 3rd Dimension	EDMA.EDMA_TPCC_CCNT_n[15:0] CCNT	0x1: One frame in the block

30.4.1.2.5 Soft-Decisions Mode

Table 30-26 shows the sequence for configuring EDMA transfer for Soft-Decisions Mode. The parameters should be set as:

Table 30-26. Configuration EDMA Soft-Decisions Mode

Step	Register/Bit Field/Programming Model	Value
Intermediate transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[23] ITCCHEN	0x0: Disable
Transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[22] TCCHEN	0x0: Disable
Intermediate transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[21] ITCINTEN	0x0: Disable
Transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[20] TCINTEN	0x0: Disable
Backward compatibility mode	EDMA.EDMA_TPCC_OPT_n[19] WIMODE	0x0: Normal operation
Transfer complete code	EDMA.EDMA_TPCC_OPT_n[17:12] TCC	1 to 63
Transfer complete code mode	EDMA.EDMA_TPCC_OPT_n[11] TCCMODE	0x0: Normal completion
FIFO width	EDMA.EDMA_TPCC_OPT_n[10:8] FWID	x

Table 30-26. Configuration EDMA Soft-Decisions Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Static Entry	EDMA.EDMA_TPCC_OPT_n[3] STATIC	0x0: Entry is updated as normal.
Transfer Synchronization Dimension	EDMA.EDMA_TPCC_OPT_n[2] SYNCDIM	0x1 AB-Sync, Each event triggers the transfer of BCNT arrays of ACNT elements .
Destination Address Mode	EDMA.EDMA_TPCC_OPT_n[1] DAM	0x1: NCR, Dst addressing within an array increments. Dst is not a FIFO .
Source Address Mode	EDMA.EDMA_TPCC_OPT_n[0] SAM	0x0: INCR, Src addressing within an array increments. Source is not a FIFO.
Source Address	EDMA.EDMA_TPCC_SRC_n[31:0] SRC	xxxx: VCPRDECS Decision FIFO address
Number of Hard decision bytes in an array	EDMA.EDMA_TPCC_ABCNT_n[15:0] ACNT	0x8, where $A(SYMR+1) \times 8$ (18)
Number of arrays in a frame Where TNSD is the total number of soft decisions (framelength)	EDMA.EDMA_TPCC_ABCNT_n[15:0] BCNT	$BCNT = CEIL(TNSD/ACNT)$ (19)
Destination Address	EDMA.EDMA_TPCC_DST_n[31:0] DST	xxxx: Soft Decision Array Address
Source 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[15:0] SBIDX	0x8 (to create incrementing transfer)
Destination 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[31:16] DBIDX	0x0 (to access the FIFO)
Source Frame Index	EDMA.EDMA_TPCC_CIDX_n[15:0] SCIDX	ACNT × BCNT
Destination Frame Index	EDMA.EDMA_TPCC_CIDX_n[31:16] DCIDX	0x00 (to access the FIFO)
C byte count. Count for 3rd Dimension	EDMA.EDMA_TPCC_CCNT_n[15:0] CCNT	0x1: One frame in the block

30.4.1.2.6 Output Parameters Transfer

Table 30-27 shows the sequence for configuring EDMA for Output Parameters Transfer. This transfer is optional and depends on the OUTF bit. It is a 2- to 32-bit word VCP1REVT and VCP2REVT frames synchronized transfer.

Table 30-27. Configuration EDMA Output Parameters Transfer

Step	Register/Bit Field/Programming Model	Value
Intermediate transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[23] ITCCHEN	0x0: Disable
Transfer complete chaining is disabled	EDMA.EDMA_TPCC_OPT_n[22] TCCHEN	0x0: Disable
Intermediate transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[21] ITCINTEN	0x0: Disable
Transfer complete interrupt is disabled	EDMA.EDMA_TPCC_OPT_n[20] TCINTEN	0x0: Disable
Backward compatibility mode	EDMA.EDMA_TPCC_OPT_n[19] WIMODE	0x0: Normal operation
Transfer complete code	EDMA.EDMA_TPCC_OPT_n[17:12] TCC	1 to 63
Transfer complete code mode	EDMA.EDMA_TPCC_OPT_n[11] TCCMODE	0x0: Normal completion
FIFO width	EDMA.EDMA_TPCC_OPT_n[10:8] FWID	x
Static Entry	EDMA.EDMA_TPCC_OPT_n[3] STATIC	0x0: Entry is updated as normal.
Transfer Synchronization Dimension	EDMA.EDMA_TPCC_OPT_n[2] SYNCDIM	0x0: A-sync transfer, each event triggers the transfer of ACNT elements.

Table 30-27. Configuration EDMA Output Parameters Transfer (continued)

Step	Register/Bit Field/Programming Model	Value
Destination Address Mode	EDMA.EDMA_TPCC_OPT_n[1] DAM	0x0: Destination address within an array increments. Destination is not a FIFO.
Source Address Mode	EDMA.EDMA_TPCC_OPT_n[0] SAM	0x0: Source Address within an array increments. Source is not a FIFO.
Source Address	EDMA.EDMA_TPCC_SRC_n[31:0] SRC	xxxx: VCP_VCPOUT0 - VCP OUTPUT Register 0 Address
Number of Hard decision bytes in an array	EDMA.EDMA_TPCC_ABCNT_n[15:0] ACNT	0x8
Number of arrays in a frame	EDMA.EDMA_TPCC_ABCNT_n[15:0] BCNT	0x1
Destination Address	EDMA.EDMA_TPCC_DST_n[31:0] DST	xxxx: Output Register Store Array Address
Source 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[15:0] SBIDX	0x0
Destination 2nd Dimension Index	EDMA.EDMA_TPCC_BIDX_n[31:16] DBIDX	0x0
Source Frame Index	EDMA.EDMA_TPCC_CIDX_n[15:0] SCIDX	0x00
Destination Frame Index	EDMA.EDMA_TPCC_CIDX_n[31:16] DCIDX	0x00
C byte count. Count for 3rd Dimension	EDMA.EDMA_TPCC_CCNT_n[15:0] CCNT	0x1: One frame in the block

Upon completion, this EDMA transfer is linked to one of the following:

1. The EDMA decisions transfer parameters of the next user-channel, if there is one ready to be decoded.
2. Null EDMA transfer parameters (with all zeros), if there are no more user-channels ready to be decoded.

30.4.2 Input Configuration Words

The input configuration words should reflect the parameters of the user-channels to be decoded.

The POLYn bits in VCP_VCPIC0 correspond to the generator polynomials in the encoder (see Figure 30-4). The values in each POLYn bit fields must be entered in reverse order. The POLYn least-significant bit is set by the VCPs logic.

- For rate 1/2, POLY0 and POLY1 are required
- For rate 1/3, POLY0, POLY1, and POLY2 are required
- For rate 1/4, all the POLYn bits are required

The VCP_VCPIC1[27:16] YAMT and VCP_VCPIC1[28]YAMEN bits are described in Section 30.5.2.

The VCP_VCPIC2[15:0] F and VCP_VCPIC2[31:15] R bits, the VCP_VCPIC3[15:0] C bit, and the VCP_VCPIC5[29:28] TB bits are described in Section 30.5.2.

The VCP_VCPIC5[7:0] IMAXI bit-field determines which state should be initialized with the maximum state metrics value (IMAXS), all the other states are initialized with the minimum state metrics value (IMINS). The IMAXI can range from 0 to 2^{K}-1-1. The IMAXS and IMINS are 13-bit signed values.

The VCP_VCPIC5[19:16] SYMX and VCP_VCPIC5[24:20] SYMR bits are described in Section 30.5.2.

The VCP_VCPIC5[30] OUTF bit indicates whether the VCP should generate a VCPnREVT for reading the output parameters. The OUTF bit setting will impact the EDMA programming (see Section 30.4.1.2.3).

30.5 VCP Register Manual

30.5.1 VCP1 and VCP2 Instance Summary

Table 30-28 shows the base address and address space for the VCP1 and VCP2 modules instances.

Table 30-28. VCP1 and VCP2 Instance Summary

Module Name	Base Address L3_MAIN	Base Address L4_PER2 Interconnect	Size
VCP1	0x4640 0000	0x4844 6000	4 KB
VCP2	0x4680 0000	0x4844 8000	4 KB

30.5.2 VCP Registers

The VCP1 and VCP2 contain several memory-mapped registers accessible by the CPU, and the EDMA module. A configuration-bus access is faster than an EDMA-bus access for isolated accesses (typically when accessing control registers). EDMA-bus accesses are used for EDMA transfers and provide maximum throughput to/from the VCP modules. The registers are listed in Table 30-29 and Table 30-30 .

The branch metric and traceback decision memories contents are not accessible and the memories can be regarded as FIFOs by the DSP, no indexing on addresses need to be performed.

30.5.2.1 VCP Register Summary

Table 30-29 summarizes the VCP1 and VCP2 data registers accessible by EDMA-bus.

Table 30-29. VCP DATA Registers Summary

Register Name	Type	Register Width (Bits)	L3 Interconnect / EDMA Bus	VCP1 Physical Address L3_MAIN	VCP2 Physical Address L3_MAIN
VCP_VCPIC0	RW	32	0x0000 0000	0x4640 0000	0x4680 0000
VCP_VCPIC1	RW	32	0x0000 0004	0x4640 0004	0x4680 0004
VCP_VCPIC2	RW	32	0x0000 0008	0x4640 0008	0x4680 0008
VCP_VCPIC3	RW	32	0x0000 000C	0x4640 000C	0x4680 000C
VCP_VCPIC4	RW	32	0x0000 0010	0x4640 0010	0x4680 0010
VCP_VCPIC5	RW	32	0x0000 0014	0x4640 0014	0x4680 0014
VCP_VCPOUT0	RW	32	0x0000 0048	0x4640 0048	0x4680 0048
VCP_VCPOUT1	RW	32	0x0000 004C	0x4640 004C	0x4680 004C
VCP_VCPWBM	RW	32	0x0000 0080	0x4640 0080	0x4680 0080
VCP_VCPRDECS	RW	32	0x0000 00C0	0x4640 00C0	0x4680 00C0

Table 30-30 summarizes the VCP1 and VCP2 configuration registers.

Table 30-30. VCP Configuration Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	VCP1 Physical Address L4_PER2 Interconnect	VCP2 Physical Address L4_PER2 Interconnect
REVISION	RW	32	0x0000 0000	0x4844 6000	0x4844 8000
VCP_SYSCONFIG	RW	32	0x0000 0010	0x4844 6010	0x4844 8010
VCP_IRQ_EOI	W	32	0x0000 0020	0x4844 6020	0x4844 8020
RESERVED	R	32	0x0000 0024	0x4844 6024	0x4844 8024
VCP_IRQSTATUS	RW	32	0x0000 0028	0x4844 6028	0x4844 8028
VCP_IRQENABLE_SET	RW	32	0x0000 002C	0x4844 602C	0x4844 802C
VCP_IRQENABLE_CLR	RW	32	0x0000 0030	0x4844 6030	0x4844 8030
RESERVED	R	32	0x0000 0044	0x4844 6044	0x4844 8044
RESERVED	R	32	0x0000 0048	0x4844 6048	0x4844 8048

Table 30-30. VCP Configuration Registers Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VCP1 Physical Address L4_PER2 Interconnect	VCP2 Physical Address L4_PER2 Interconnect
VCP_DEBUG	RW	32	0x0000 0050	0x4844 6050	0x4844 8050
VCP_VCPEXE	RW	32	0x0000 0118	0x4844 6118	0x4844 8118
VCP_VCPEND	RW	32	0x0000 0120	0x4844 6120	0x4844 8120
VCP_VCPSTAT0	RW	32	0x0000 0140	0x4844 6140	0x4844 8140
VCP_VCPSTAT1	RW	32	0x0000 0144	0x4844 6144	0x4844 8144
VCP_VCPERR	RW	32	0x0000 0150	0x4844 6150	0x4844 8150
VCP_VCPEMU	RW	32	0x0000 0160	0x4844 6160	0x4844 8160

Table 30-31 summarizes the VCP1 and VCP2 memories.

Table 30-31. VCP1 and VCP2 Memories Register Summary

EDMA Bus Offsets	VCP1 Physical Address L3_MAIN Interconnect	VCP2 Physical Address L3_MAIN Interconnect	Acronym	Memory Name	Size
0x0000 1000	0x4640 7000	0x4680 9000	BM	Branch Metrics (BM)	256 Bytes
0x0000 2000	0x4640 8000	0x4680 A000	SM	State Metric (SM)	448 Bytes
0x0000 3000	0x4640 9000	0x4680 B000	TBHD	Traceback Hard Decision	4 KBytes
0x0000 6000	0x4640 C000	0x4680 E000	TBSD	Traceback Soft Decision	16 KBytes
0x0000 F000	0x4641 B000	0x4681 D000	IO	Decoded Bits (IO)	512 Bytes

Note

Register and Memory Access

- Data Transfer Alignment: Normal (non-emulation) mode data transfers to/from the VCP1 and VCP2 must be aligned on a double-word (64-bit) boundary. Alignment can be forced in C using the 'DATA_ALIGN' pragma. Non-alignment results in data transfer failure.
  Example:

```
#pragma DATA_ALIGN(configIc, 8) // Should be double-word aligned
VCP_ConfigIc configIc; // VCP Input Configuration Reg
```

- Data Transfer Size: Normal (non-emulation) mode data transfers to/from the VCP1 and VCP2 must be of a length that is an 8-byte (double-word) multiple.
- Emulation mode transfers are performed on 32-bit boundaries and are 4 bytes in length.

30.5.2.2 VCP1 and VCP2 Data Registers Description

Table 30-32 through Table 30-50 describe the individual VCP1 and VCP2 data registers.

Table 30-32. VCP_VCPIC0

Address Offset	0x0000 0000		
Physical Address	0x4640 0000 0x4680 0000	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Input Configuration Register 0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

POLY3	POLY2	POLY1	POLY0

Bits	Field Name	Description	Type	Reset
31:24	POLY3	Polynomial generator G_3.	RW	0xFF
23:16	POLY2	Polynomial generator G_2.	RW	0xFF
15:8	POLY1	Polynomial generator G_1.	RW	0xFF
7:0	POLY0	Polynomial generator G_0.	RW	0xFF

Table 30-33. Register Call Summary for Register VCP_VCPIC0

VCP Modules Programming Guide

- Special VCP EDMA Programming Considerations: [0]
- Input Configuration Words: [1]

VCP Register Manual

- VCP Register Summary: [2]

Table 30-34. VCP_VCPIC1

Address Offset	0x0000 0004		
Physical Address	0x4640 0004 0x4680 0004	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Input Configuration Register 1		
Type	RW		

31:29	28	27:16	15:0
RESERVED	YAMEN	YAMT	RESERVED

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	YAMEN	Yamamoto algorithm enable bit. 0x0: Yamamoto algorithm is disabled. 0x1: Yamamoto algorithm is enabled.	RW	0x0
27:16	YAMT	Yamamoto threshold value bits.	RW	0xFFF
15:0	RESERVED		R	0x0000

Table 30-35. Register Call Summary for Register VCP_VCPIC1

VCP Functional Description

- Sliding Windows Processing: [0] [1]

VCP Modules Programming Guide

- Input Configuration Words: [2] [3]

VCP Register Manual

- VCP Register Summary: [4]

Table 30-36. VCP_VCPIC2

Address Offset	0x0000 0008		
Physical Address	0x4640 0008 0x4680 0008	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Input Configuration Register 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

R	F

Bits	Field Name	Description	Type	Reset
31:16	R	Reliability length bits.	RW	0xFFFF
15:0	F	Frame length bits.	RW	0xFFFF

Table 30-37. Register Call Summary for Register VCP_VCPIC2

VCP Modules Programming Guide

- Input Configuration Words: [0] [1]

VCP Register Manual

- VCP Register Summary: [2]

Table 30-38. VCP_VCPIC3

Address Offset	0x0000 000C		
Physical Address	0x4640 000C 0x4680 000C	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Input Configuration Register 3		
Type	RW		

31 30 29	28	27 26 25	24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	OUT_ORDER	RESERVED	ITBEN	ITBI	C

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	OUT_ORDER	Defines the order of VCP output for decoded data. 0x0: 0 to 31 0x1: 31 to 0	RW	0x0
27:25	RESERVED		R	
24	ITBEN	Traceback state index enable/disable. 0x0: Disabled 0x1: Initialization of traceback starting state is enabled	RW	0x0
23:16	ITBI	Traceback state index. The index of the starting state for the traceback unit.	RW	0xFF
15:0	C	Convergence distance bits. The length of the convergent section of the siding window. This is only used if *F > F + (K-1)* in mixed mode, or if *F > F + C* in convergence mode.	RW	0xFFFF

Table 30-39. Register Call Summary for Register VCP_VCPIC3

VCP Functional Description

- Decision Data: [0]
- Branch Metrics: [1]

VCP Modules Programming Guide

- Input Configuration Words: [2]

VCP Register Manual

- VCP Register Summary: [3]

Table 30-40. VCP_VCPIC4

Address Offset	0x0000 0010		
Physical Address	0x4640 0010 0x4680 0010	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Input Configuration Register 4		
Type	RW		

31:29	28:16	15:13	12:0
RESERVED	IMINS	RESERVED	IMAXS

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	IMINS	Minimum initial state metric value bits. 13 bits.	RW	0x1FFF
15:13	RESERVED		R	0x0
12:0	IMAXS	Maximum initial state metric value bits. 13 bits.	RW	0x1FFF

Table 30-41. Register Call Summary for Register VCP_VCPIC4

VCP Functional Description

- Errors and Status: [0] [1]

VCP Register Manual

- VCP Register Summary: [2]

Table 30-42. VCP_VCPIC5

Address Offset	0x0000 0014		
Physical Address	0x4640 0014 0x4680 0014	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Input Configuration Register 5		
Type	RW		

31	30	29:28	27:25	24:20	19:16	15:8	7:0
SDHD	OUTF	TB	RESERVED	SYMR	SYMX	RESERVED	IMAXI

Bits	Field Name	Description	Type	Reset
31	SDHD	Output decision type select bit. 0x0: Hard decisions 0x1: Soft decisions	RW	0x0
30	OUTF	Output parameters read flag bit. 0x0: VCPnREVT is not generated by VCP for output parameters read 0x1: VCPnREVT generated by VCP for output parameters read	RW	0x0
29:28	TB	Traceback mode select bits. 0x0: Reserved 0x1: Tailed, F ≤ F max. See Section 30.3.4.1.4 0x2: Convergent, (no tail bits) 0x3: Mixed, F ≥ F max and tail bits are used. See Section 30.3.4.1.4	RW	0x1
27:25	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
24:20	SYMR	Determines decision buffer length in output FIFO. When programming register values for the SYMR bits, always subtract 1 from the value calculated. Valid values for the SYMR bits are from 0x0 to 0xF. For hard decision: If F ≤ 2048; then symr = ceil[F/64]-1; If F > 2048; then symr = 15 or 31 For soft decision: If F ≤ 256; then symr = ceil[F/8]-1; If F > 256; then symr = 15 or 31	RW	0xF
19:16	SYMX	Determines branch metrics buffer length in input FIFO.	RW	0xF
15:8	RESERVED		R	0x0
7:0	IMAXI	Maximum initial state metric value bits. IMAXI bits determine which state should be initialized with the maximum state metrics value (IMAXS) bits in VCPIC4; All the other states will be initialized with the value in the IMINS bits.	RW	0xFF

Table 30-43. Register Call Summary for Register VCP_VCPIC5

VCP Functional Description

- Sliding Windows Processing: [0] [1] [2] [3] [4]
- VCPnXEVT Generation: [5] [6]
- VCPnREVT Generation: [7]
- Errors and Status: [8] [9]

VCP Modules Programming Guide

- Special VCP EDMA Programming Considerations: [10] [11] [12]
- Input Configuration Words: [13] [14] [15] [16] [17]

VCP Register Manual

- VCP Register Summary: [18]

Table 30-44. VCP_VCPOUT0

Address Offset	0x0000 0048		
Physical Address	0x4640 0048 0x4680 0048	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Output Register 0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			FMINS													RESERVED			FMAXS												

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	FMINS	Minimum initial state metric value for the final trellis stage. 13 bits.	RW	0xFFF
15:13	RESERVED		R	0x0
12:0	FMAXS	Maximum state metric value for the final trellis stage (at trellis stage R+C). 13 bits.	RW	0xFFF

Table 30-45. Register Call Summary for Register VCP_VCPOUT0

VCP Functional Description

- VCP Output Parameters: [0] [1]

Table 30-45. Register Call Summary for Register VCP_VCPOUT0 (continued)

VCP Modules Programming Guide

- Special VCP EDMA Programming Considerations: [2]

VCP Register Manual

- VCP Register Summary: [3]

Table 30-46. VCP_VCPOUT1

Address Offset	0x0000 004C		
Physical Address	0x4640 004C 0x4680 004C	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	The VCP version 2 Output Register 1		
Type	RW		

31–17	16	15–8	7–0
RESERVED	YAM	RESERVED	FMAXI

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16	YAM	Yamamoto bit result. This bit is a quality indicator bit and is only used if the Yamamoto logic is enabled. 0x0: At least one trellis stage had an absolute difference less than the Yamamoto threshold and the decided frame has poor quality 0x1: No trellis stage had an absolute difference less than the Yamamoto threshold and the frame has good quality	RW	0x0
15:8	RESERVED		R	0x0
7:0	FMAXI	State index for the state with the final maximum state metric. There are 2(k-1) state metrics for each trellis stage. Valid range for FMAXI is 0 to 2(k-1) -1.	RW	0xFFF

Table 30-47. Register Call Summary for Register VCP_VCPOUT1

VCP Functional Description

- Sliding Windows Processing: [0] [1]
- VCP Output Parameters: [2] [3]

VCP Register Manual

- VCP Register Summary: [4]

Table 30-48. VCP_VCPWBM

Address Offset	0x0000 0080		
Physical Address	0x4640 0080 0x4680 0080	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	VCP branch metrics write FIFO register		
Type	RW		

31–0
WBM

Bits	Field Name	Description	Type	Reset
31:0	WBM	VCP branch metrics write FIFO	RW	0x0000 0000

Table 30-49. Register Call Summary for Register VCP_VCPWBM

VCP Modules Programming Guide

- Special VCP EDMA Programming Considerations: [0]

VCP Register Manual

- VCP Register Summary: [1]

Table 30-50. VCP_VCPRDECS

Address Offset	0x0000 00C0		
Physical Address	0x4640 00C0 0x4680 00C0	**Instance**	VCP1_MAIN_L3 VCP2_MAIN_L3
Description	VCP decisions read FIFO register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDECS																															

Bits	Field Name	Description	Type	Reset
31:0	RDECS	VCP decisions read FIFO	R	0x0000 0000

Table 30-51. Register Call Summary for Register VCP_VCPRDECS

VCP Register Manual

- VCP Register Summary: [0]

30.5.2.3 VCP1 and VCP2 Configuration Registers Description

Table 30-52 through Table 30-76 describe the individual VCP1 and VCP2 configuration registers.

Table 30-52. REVISION

Address Offset	0x0000 0000		
Physical Address	0x4844 6000 0x4844 8000	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	VCP version 2 - IP Revision Register is used to track the version of the IP.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOURCE_IP																REV_IP															

Bits	Field Name	Description	Type	Reset
31:16	SOURCE_IP	Source of VCP IP	R	0x5007
15:0	REV_IP	VCP IP Revision number	R	0x1900

Table 30-53. Register Call Summary for Register REVISION

VCP Register Manual

- VCP Register Summary: [0]

Table 30-54. VCP_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4844 6010 0x4844 8010	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	System Configuration Register is used to set the idle modes for the VCP modules		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	IDLEMODE	RESERVED	RESET_DONE

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000
3:2	IDLEMODE	Idle mode bit 0x0: Idle request unconditionally acknowledged (ack = req) 0x1: Acknowledge always inactive (ack=0) 0x2: Idle request acknowledge pending internal conditions 0x3: Reserved	RW	0x0
1	RESERVED		R	0x0
0	RESET_DONE	Reset done is a read only and shows the status of the reset from the idle command.	R	0x0

Table 30-55. Register Call Summary for Register VCP_SYSCONFIG

VCP Functional Description

- VCP Power Management: [0] [1]
- VCP Resets: [2] [3]

VCP Register Manual

- VCP Register Summary: [4]

Table 30-56. VCP_IRQ_EOI

Address Offset	0x0000 0020		
Physical Address	0x4844 6020 0x4844 8020	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	End of interrupt register.		
Type	W		

31:1	0
RESERVED	LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LINE_NUMBER	Software End of Interrupt (EOI) control. Write a value of 0x0 to repulse the interrupt output if any interrupts are pending.	W	0x0

Table 30-57. Register Call Summary for Register VCP_IRQ_EOI

VCP Register Manual

- VCP Register Summary: [0]

Table 30-58. VCP_IRQSTATUS

Address Offset	0x0000 0028

Table 30-58. VCP_IRQSTATUS (continued)

Physical Address	0x4844 6028 0x4844 8028	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	IRQ status register captures the current active status of the interrupts after the enabling function.		
Type	RW		

31–1	0
RESERVED	STATUS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	STATUS	VCP IRQ enable status 0x0: VCP error interrupt is not enabled. 0x1: VCP error interrupt is enabled.	RW	0x0

Table 30-59. Register Call Summary for Register VCP_IRQSTATUS

VCP Functional Description

- Interrupt Requests: [0]
- Errors and Status: [1] [2]

VCP Register Manual

- VCP Register Summary: [3]

Table 30-60. VCP_IRQENABLE_SET

Address Offset	0x0000 002C		
Physical Address	0x4844 602C 0x4844 802C	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	The VCP set enable interrupt register allows the user to enable the VCP error interrupt. The software should enable the interrupt on the VCP by writing writing a 1 to bit 0 of the IRQENABLE_SET register.		
Type	RW		

31–1	0
RESERVED	ENABLE_SET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ENABLE_SET	VCP IRQ enable status 0x0: VCP error interrupt is disabled. 0x1: VCP error interrupt is enabled.	RW	0x0

Table 30-61. Register Call Summary for Register VCP_IRQENABLE_SET

VCP Functional Description

- Interrupt Requests: [0] [1]
- Errors and Status: [2]

VCP Register Manual

- VCP Register Summary: [3]

Table 30-62. VCP_IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	0x4844 6030 0x4844 8030	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	The VCP clear enable interrupt register allows the user to disable the VCP error interrupt.		
Type	RW		

31:1	0
RESERVED	ENABLE_CLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ENABLE_CLR	VCP IRQ enable status 0x0: No effect 0x1: VCP error interrupt is disabled.	RW	0x0

Table 30-63. Register Call Summary for Register VCP_IRQENABLE_CLR

VCP Functional Description

- Interrupt Requests: [0]

VCP Register Manual

- VCP Register Summary: [1]

Table 30-64. VCP_DEBUG

Address Offset	0x0000 0050		
Physical Address	0x4844 6050 0x4844 8050	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	Debug Configuration Register is used to view that status of various events in the VCP1 and VCP2 including emulation suspend mode request and DMA status for read and write requests.		
Type	R		

31:4	3	2	1	0
RESERVED	EMUSUSP	RESERVED	DMA_X_REQ	DMA_R_REQ

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	EMUSUSP	Status of the emulation suspend mode request 0x0: Emulation suspend mode request has not been sent 0x1: Emulation suspend mode requrest has been sent	R	0x0
2	RESERVED		R	0x0
1	DMA_X_REQ	Status of the VCP tranmit event (VCPnXEVT) 0x0: No transmit DMA (write) event is pending 0x1: Transmit DMA (write) event is pending	R	0x0

Bits	Field Name	Description	Type	Reset
0	DMA_R_REQ	Status of the VCP receive event (VCPnREVT) 0x0: No receive DMA (read) event is pending 0x1: Receive DMA (read) event is pending	R	0x0

Table 30-65. Register Call Summary for Register VCP_DEBUG

VCP Functional Description • EDMA Requests: [0] [1]
VCP Register Manual • VCP Register Summary: [2]

Table 30-66. VCP_VCPEXE

Address Offset	0x0000 0118		
Physical Address	0x4844 6118 0x4844 8118	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	VCP version 2 execution register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												COMMAND			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 000
3:0	COMMAND	VCP command select bits: 0x0: Reserved 0x1: Start VCP (normal mode) 0x2: Halt or Pause VCP (debug mode). The VCP is halted (or paused) after processing the state metric for the current sliding window and before the start of the traceback. 0x3: Restart VCP and process one sliding window (debug mode). The VCP is restarted from the pause state and begins the traceback operation. The VCP is again paused after processing the state metrics for next sliding window. 0x4: Restart VCP (debug mode). The VCP is restarted from the paused state and begins the traceback operation. The VCP will run to normal completion. 0x5: Stop. Soft reset all VCP registers to their initial condition. All registers in the VCP are reset in this mode except for the execution register, endian register, emulation register, and other internal registers. 0x6: Reserved	W	0x0

Table 30-67. Register Call Summary for Register VCP_VCPEXE

VCP Functional Description • VCPnXEVT Generation: [0] • Errors and Status: [1]
VCP Register Manual • VCP Register Summary: [2]

Table 30-68. VCP_VCPEND

Address Offset	0x0000 0120

Table 30-68. VCP_VCPEND (continued)

Physical Address	0x4844 6120 0x4844 8120	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	VCP Endian Mode Register		
Type	RW		

31:10	9	8	7:2	1	0
RESERVED	SLPZVSS_EN	SLPZVDD_EN	RESERVED	SD	BM

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000
9	SLPZVSS_EN	Sleep mode for SLPZVSS_EN 0x0: Programed for proper operation.	RW	0x1
8	SLPZVDD_EN	Sleep mode for SLPZVDD_EN 0x0: Programed for proper operation.	RW	0x1
7:2	RESERVED		R	0x00
1	SD	Traceback soft-decision memory format select bit. 0x0: 32-bit-word packed 0x1: Native format (8 bits)	RW	0x0
0	BM	Branch metrics memory format select bit. 0x0: 32-bit-word packed 0x1: Native format (8 bits)	RW	0x0

Table 30-69. Register Call Summary for Register VCP_VCPEND

VCP Functional Description

- Branch Metrics Calculations: [0]
- VCP Memory Sleep Mode: [1] [2]
- Decision Data: [3] [4]
- Endianness: [5]

VCP Register Manual

- VCP Register Summary: [6]

Table 30-70. VCP_VCPSTAT0

Address Offset	0x0000 0140		
Physical Address	0x4844 6140 0x4844 8140	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	VCP Status Register 0		
Type	R		

31:29	28:12	11:7	6	5	4	3	2	1	0
RESERVED	NSYMPROC	RESERVED	EMUHALT	OFFUL	IFEMP	WIC	ERR	RUN	PAUSE

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
28:12	NSYMPROC	Number of symbols processed bits. The NSYMPROC bits indicate how many symbols have been processed in the state metric unit with respect to time. The maximum number of processed stages is equal to F + (k-1) in tailed or mixed mode. The maximum number of processed stages is equal to *F* + *C* in convergent mode.	R	0x0000
11:7	RESERVED		R	0x00
6	EMUHALT	Emulation halt status bit 0x0: No halt due to emulation 0x1: Halt due to emulation	R	0x0
5	OFFUL	Output FIFO buffer full status bit 0x0: Output FIFO buffer is not full 0x1: Output FIFO buffer is full	R	0x0
4	IFEMP	Input FIFO buffer empty status bit 0x0: Input FIFO buffer is not empty 0x1: Input FIFO buffer is empty	R	0x0
3	WIC	Waiting for input configuration bit. The WIC bit indicates that the VCP is waiting for new input control parameters to be written. This bit is always set after decoding of a user channel. 0x0: Not waiting for input configuration words 0x1: Waiting for input configuration words	R	0x0
2	ERR	VCP error status bit. The ERR bit is cleared as soon as the DSP reads the VCP error register (VCPERR). 0x0: No error 0x1: VCP paused due to error	R	0x0
1	RUN	VCP running status bit 0x0: VCP is not running 0x1: VCP is running	R	0x0
0	PAUSE	VCP pause status bit 0x0: VCP is not paused. The UNPAUSE command is acknowledged by clearing the PAUSE bit. 0x1: VCP is paused. The PAUSE command is acknowledged by setting the PAUSE bit. The PAUSE bit can also be set, if the input FIFO buffer is becoming empty or if the output FIFO buffer is full.	R	0x0

Table 30-71. Register Call Summary for Register VCP_VCPSTAT0

VCP Functional Description

- Errors and Status: [0]

VCP Register Manual

- VCP Register Summary: [1]

Table 30-72. VCP_VCPSTAT1

Address Offset	0x0000 0144		
Physical Address	0x4844 6144 0x4844 8144	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	VCP Status Register 1		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NSYMOF	NSYMIF

Bits	Field Name	Description	Type	Reset
31:16	NSYMOF	Number of symbols in the output FIFO buffer.	R	0xFFFF
15:0	NSYMIF	Number of symbols in the input FIFO buffer.	R	0xFFFF

Table 30-73. Register Call Summary for Register VCP_VCPSTAT1

VCP Register Manual

- VCP Register Summary: [0]

Table 30-74. VCP_VCPERR

Address Offset	0x0000 0150		
Physical Address	0x4844 6150 0x4844 8150	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	VCP Error Register		
Type	R		

31–7	6	5	4	3	2	1	0
RESERVED	E_SYMR	E_SYMX	MAXIMINERR	FCTLERR	FTLERR	TBNAERR	ERROR

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x000000
6	E_SYMR	SMAR error 0x0: No error for SMAR. Table 30-19 0x1: Error	R	0x0
5	E_SYMX	SMAX Error 0x0: No error for SMAX. Table 30-20 0x1: Error	R	0x0
4	MAXIMINERR	MAXIMIN ERROR 0x0: No error. Table 30-21 0x1: Error	R	0x0
3	FCTLERR	FCTL error 0x0: No error 0x1: 1 = r + c too large ((R + C) > (R + C)max) for mixed or convergent traceback modes	R	0x0
2	FTLERR	FTL Error 0x0: No error 0x1: F too large (F > Fmax) for tailed traceback mode	R	0x0
1	TBNAERR	TBNA Error 0x0: No error 0x1: Traceback mode is not allowed.	R	0x0
0	ERROR	Error 0x0: No error is detected. 0x1: An error has occurred	R	0x0

Table 30-75. Register Call Summary for Register VCP_VCPERR

VCP Functional Description

- Errors and Status: [0] [1] [2]

VCP Register Manual

- VCP Register Summary: [3]

Table 30-76. VCP_VCPEMU

Address Offset	0x0000 0160		
Physical Address	0x4844 6160 0x4844 8160	**Instance**	VCP1_PER2_L4 VCP2_PER2_L4
Description	VCP Emulation Control Register		
Type	RW		

31–2	1	0
RESERVED	SOFT	FREE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x000000
1	SOFT	Soft bit 0x0: Default mode - VCP completes the current window of state metric processing and halts before starting traceback or at the end of a frame 0x1: VCP completes a frame of data before halting	RW	0x0
0	FREE	Free bit 0x0: SOFT bit takes effect 0x1: Free run mode - peripheral ignores the vcp_emususp signal and functions normally.	RW	0x0

Table 30-77. Register Call Summary for Register VCP_VCPEMU

VCP Functional Description

- Debugging Features: [0] [1] [2]

VCP Register Manual

- VCP Register Summary: [3]

Chapter 31

Audio Tracking Logic

This chapter describes the Audio Tracking Logic (ATL) module.

31.1 ATL Overview.......... 6914
31.2 ATL Environment.......... 6915
31.3 ATL Integration.......... 6916
31.4 ATL Functional Description.......... 6918
31.5 ATL Register Manual.......... 6921

31.1 ATL Overview

The audio tracking logic (ATL) is used by HD Radio™ applications to synchronize the digital audio output to the baseband clock. This same IP can also be used generically to track errors between two reference signals (such as frame syncs) and generate a modulated clock output (using software-controlled cycle stealing) which averages to some desired frequency. This process can be used as a hardware assist for asynchronous sample rate conversion algorithms. The tracking range is limited, so direct conversion between the two standard sample rate groups frequencies from 44.1 to 48 kHz is not possible.

Figure 31-1 shows an overview of the ATL module.

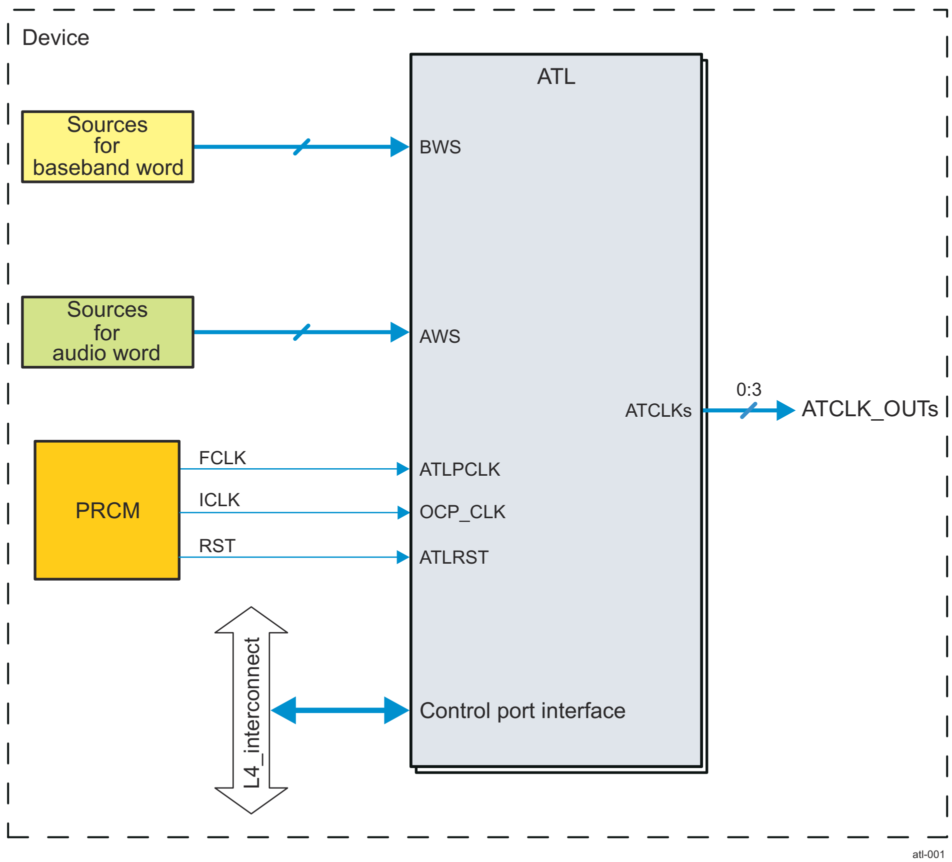


Figure 31-1. ATL Module Overview

The ATL includes the following main features:

- One ATL module, containing four ATL instances, for HD Radio support and asynchronous sample rate conversion assistance
- Each instance tracks the time error between two syncs (local audio word select [AWS] and baseband word select [BWS])
- Each instance selects between 16 mux choices for each of AWS and BWS.
- Each instance generates modulated ATCLK_OUT clock signal with software-initiated pulse stealing.
- Selection between INTERCONNECT clock or functional ATLPCLK to run error couting timers and to derive modulated ATCLK_OUT clock outputs
- Clock and reset management: Receives clock and reset signals from the device PRCM module, and has its own dedicated clock domain (CD_ATL) within the PRCM. The ATL module receives hardware reset from the CORE_RST reset domain.
- Power management: The ATL belongs to the PD_COREAON power domain.

31.2 ATL Environment

This section describes the intended functions for the ATL module from an environmental point of view (that is, external connections to SoC). This section presents the ATL connectivity options and lists the possible interface in the device.

31.2.1 ATL Functions

Four instances of the ATL module are available on the device: ATL0, ATL1, ATL2, and ATL3.

The recommended use for ATL in the device are:

- HD Radio applications to synchronize the digital audio output to the baseband clock.
- Timing reference from:
  - McASP modules
  - Audio clock signal PER_ABE_X1_GFCLK
  - Divided media local bus (MLB) clock
  - System clock modulated by audio/video bridging (AVB) transport protocol

31.2.2 ATL Signals Descriptions

Table 31-1 lists the ATL module pins, their corresponding signal names at device level, and specifies their links to functions.

Table 31-1. I/O Description

Signal Name	Device-Level Signal Name	I/O[(1)]	Description
ATLCLK0	atl_clk0	O	Audio Tracking Logic Clock 0 output
ATLCLK1	atl_clk1	O	Audio Tracking Logic Clock 1 output
ATLCLK2	atl_clk2	O	Audio Tracking Logic Clock 2 output
ATLCLK3	atl_clk3	O	Audio Tracking Logic Clock 3 output

(1) I = Input, O = Output

Note

The path from a module pin to device pad or pads is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see *Pad Configuration Registers* of *Control Module*, and the device data manual.

31.3 ATL Integration

This section describes the integration of the ATL in the device, including information about clocks, resets, and environmental modules.

Figure 31-2 shows the integration of the ATL in the device, input and output signals, clock generators, and interconnections.

The features of the ATL modules are:

- One functional clock
- One interface clock
- Interface port to L4_PER2 interconnect
- BWS mux
- AWS mux

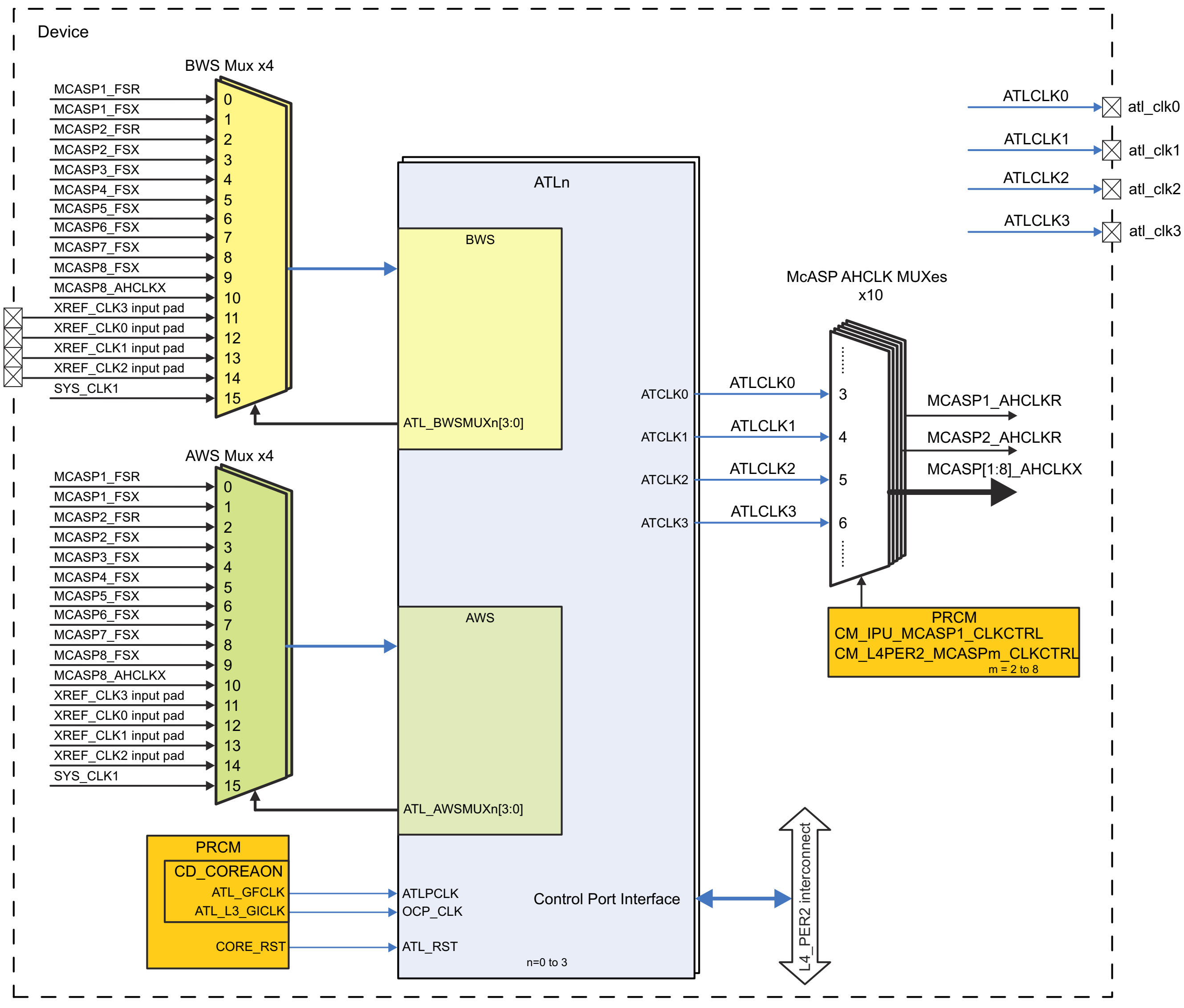


Figure 31-2. ATL Integration

Table 31-2 and Table 31-3 summarize the integration of the module in the device.

Table 31-2. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
ATL	PD_COREAON	No	L4_PER2

Table 31-3. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ATL	ATLPCLK	ATL_GFCLK	PRCM	ATL functional clock
	OCP_CLK	ATL_L3_GICLK	PRCM	ATL interface clock. Can be used as functional clock via ATL_PCLKMUX0
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ATL	ATL_RST	CORE_RST	PRCM	ATL reset from PRCM

Note

For more information about the clock and reset sources, refer to:

- Section 3.5, *Reset Management Functional Description*.
- Section 3.7, *Power Management Functional Description*.

in Chapter 3, *Power, Reset, Clock Management*.

31.3.1 ATL Distribution on Interconnects

Initiator agents (IAs) and target agents (TAs) provide the interface to connect the ATL module to its associated interconnect (see Table 31-4). For a complete description of the distribution of modules on L4 interconnects, see Section 14.3.3.1, *Module Distribution*, in Section 14.3, *Interconnect*.

Table 31-4. L4_PER2 TAs

Module Target Name	Description
TA_ATL_TARG	ATL module

31.3.2 ATL Regions Allocations

The protection mechanism for L4 interconnects is based on a hierarchical segmentation. By default, some regions are attached to specific protection group members (see Table 31-5). For a complete description of the allocation of module regions on L4 interconnects, see Section 14.3.3.3.2 Section 14.3.3.3.2, *Segments and Regions*, in Section 14.3, *L4 Interconnect*.

Note

Module refers to the configuration registers of the ATL.

TA refers to the interconnect configuration registers of the TA associated with the ATL.

Table 31-5. Region Allocations for L4_PER2 Interconnect

Module	Region	Description
ATL	24	TA
	23	Module

31.4 ATL Functional Description

This section is a functional description of the ATL modules.

31.4.1 Block Diagram

The tracking mechanism is capable of handling all input (baseband) and output (audio) sampling rates used by HD Radio. The ATL works on a single-clock domain and is insensitive to a particular choice of the master clock frequency as long as it is divisible by the desired audio frequency.

Figure 31-3 shows the functional block diagram of the four instances of the ATL module.

The ATL consists of two circuits: Measurement Circuit and Adjustment Circuit.

The main interface includes two registers: a read register in the measurement circuit, baseband sample register ATL_BBSR0; and a write register in the adjustment circuit, parts-per-milion register ATL_PPMR0.

The measurement circuit includes an 8-bit rolling counter, a 16-bit rolling counter, and the 16-bit sample count register. The 8-bit counter counts audio clock cycles. The 16-bit counter counts the number of baseband samples received by the modem. When the 8-bit counter rolls over, the content of the 16-bit counter is strobed into the sample count register ATL_BBSR0[15:0] SAMPLE_COUNT. The HD Radio library uses this content to determine the relative frequency offset between the audio clock and baseband sampling clock.

The adjusting circuit produces a timing signal at the top of the audio clock tree, which is used by dividers that make the audio timing signals including DAC oversampling clock, bit clock, and word select. The output is occasionally adjusted by altering the divide count by one “tick” of the master clock. It includes a 20-bit accumulator circuit with a 9-bit PPM ATL_PPMR0[8:0] PPM_SETTING register and an adjustable clock divider ATL_ATLCR0[4:0] ATLINTERNAL_DIVIDER. The content of the ATL_PPMR0 register is subtracted from the accumulator every master clock (ATLPCLK) cycle. On underflow, the audio clock divider adds or subtracts one master clock (ATLPCLK) period from the nominal periods of the audio clocks. The rate of adjustments in PPM (parts-per-220) is exactly the value written to ATL_PPMR0[8:0] PPM_SETTING. The MSB or ATL_PPMR0[15] PPM_SLOWDOWN of the PPM register determines whether the audio clock period is shortened or lengthened.

ATCLK is the adjusted audio clock output from the ATL. ATCLK is internally multiplexed into the McASP high-speed clock (AHCLKX) signal path per the clocking shown in the PRCM chapter (see Chapter 3, *Power, Reset, Clock Management Module*). This lets the McASP drive the master clock for an audio DAC while at the same time driving the McASP dividers for AFSX and ACLKX. ATCLK should be 128 × Fs or higher for the DAC oversampling clock.

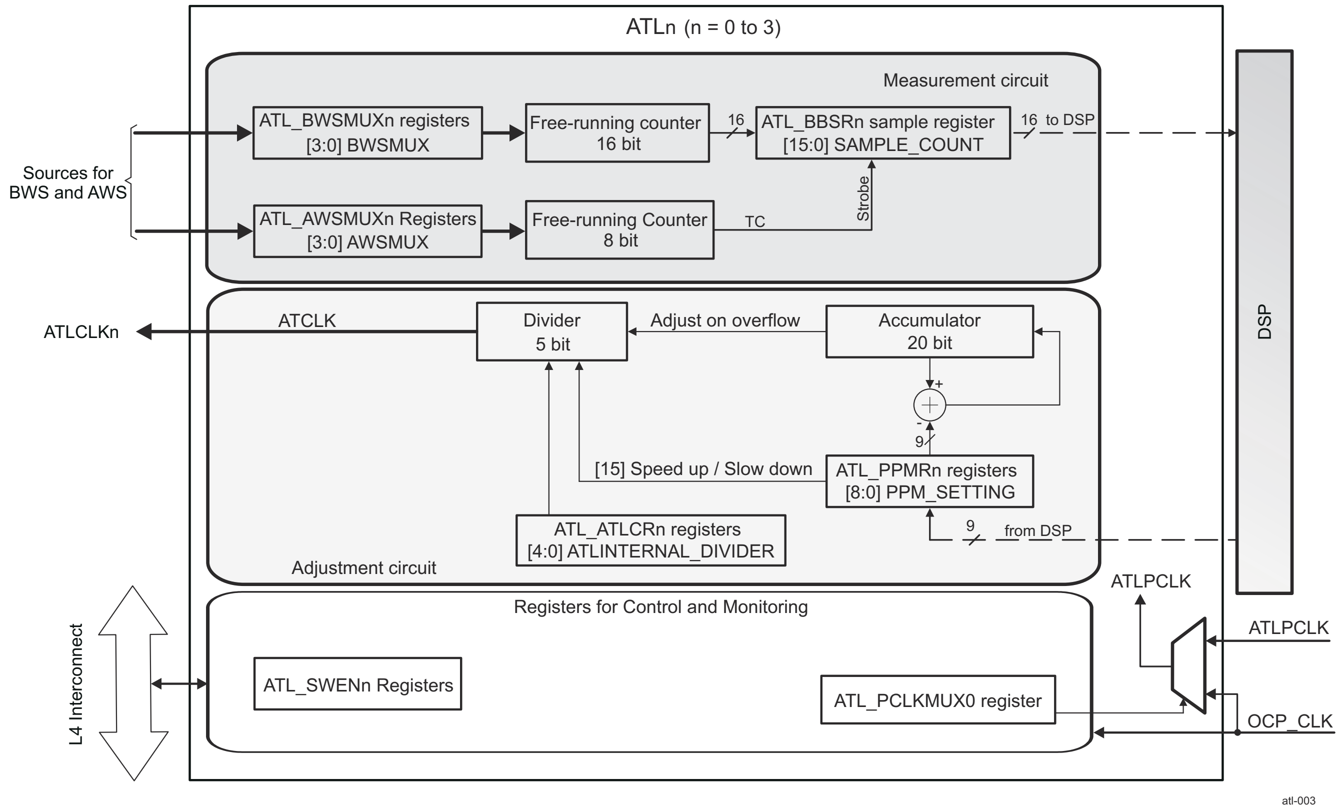


Figure 31-3. ATL Block Diagram

31.4.2 Source Signal Control

The sources for BWS and AWS inputs are defined by internal muxing from multiple reference signals.

BWS and AWS are logic inputs, not clocks, to the ATL module. ATL performs an edge detection on these signals. To minimize jitter in HD Radio clocking implementation, ATLPCLK must be higher than 100 MHz. When ATL is used for HD Radio applications, the measuring circuit is connected to the word select signal from Audio IIS output (AWS) and Baseband IIS input (BWS). The ATL_BWSMUX0 and ATL_AWSMUX0 registers select which inputs are used as the ATL BWS and AWS inputs.

31.4.3 ATL Clock and Reset Configuration

The ATL module is clocked by a functional clock (ATLPCLK) and one interface clock (OCP_CLK).

- The functional clock is used to generate control signals depending on the internal configuration of the module. The ATLPCLK selection register mux ATL_PCLKMUX0[0] PCLKMUX chooses the functional clock for ATLPCLK input to run error counting timers and to derive modulated ATCLK_OUT clock outputs.
- OCP_CLK is used to trigger access to the ATL configuration interface through the L4_PER2 Interconnect.
- A software reset register is provided to hold the four ATL modules in reset while the AWS/BWS and ATPCLK mux selections are set and while the system-level clocking is configured. Software is required to release the module from reset. Software must enable the ATL_SWEN0[0] SWEN bit to enable the ATL. Each ATL in the device must be independently enabled.

Note

The module must be enabled in the PRCM. For more information refer to *PRCM Register Manual*, in *Power, Reset, Clock Management Module*.

31.5 ATL Register Manual

31.5.1 ATL Instance Summary

Table 31-6 shows the base address for the ATL instances in the device.

Table 31-6. ATL Instance Summary

Module Name	Base Address	Size
ATL	0x4843 C000	4KB

31.5.2 ATL Register Summary

Table 31-7 summarizes the ATL0, ATL1, ATL2, and ATL3 registers.

Table 31-7. ATL Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ATL Physical Address
ATL_REVID	R	32	0x0000 0000	0x4843 C000
RESERVED	R	32	0x0000 0004	0x4843 C004
ATL_PPMR0	RW	32	0x0000 0200	0x4843 C200
ATL_BBSR0	R	32	0x0000 0204	0x4843 C204
ATL_ATLCR0	RW	32	0x0000 0208	0x4843 C208
RESERVED	R	32	0x0000 020C	0x4843 C20C
ATL_SWEN0	RW	32	0x0000 0210	0x4843 C210
ATL_BWSMUX0	RW	32	0x0000 0214	0x4843 C214
ATL_AWSMUX0	RW	32	0x0000 0218	0x4843 C218
ATL_PCLKMUX0	RW	32	0x0000 021C	0x4843 C21C
RESERVED	R	32	0x0000 0220	0x4843 C220
ATL_PPMR1	RW	32	0x0000 0280	0x4843 C280
ATL_BBSR1	R	32	0x0000 0284	0x4843 C284
ATL_ATLCR1	RW	32	0x0000 0288	0x4843 C288
RESERVED	R	32	0x0000 028C	0x4843 C28C
ATL_SWEN1	RW	32	0x0000 0290	0x4843 C290
ATL_BWSMUX1	RW	32	0x0000 0294	0x4843 C294
ATL_AWSMUX1	RW	32	0x0000 0298	0x4843 C298
ATL_PCLKMUX1	RW	32	0x0000 029C	0x4843 C29C
RESERVED	R	32	0x0000 02A0	0x4843 C2A0
ATL_PPMR2	RW	32	0x0000 0300	0x4843 C300
ATL_BBSR2	R	32	0x0000 0304	0x4843 C304
ATL_ATLCR2	RW	32	0x0000 0308	0x4843 C308
RESERVED	R	32	0x0000 030C	0x4843 C30C
ATL_SWEN2	RW	32	0x0000 0310	0x4843 C310
ATL_BWSMUX2	RW	32	0x0000 0314	0x4843 C314
ATL_AWSMUX2	RW	32	0x0000 0318	0x4843 C318
ATL_PCLKMUX2	RW	32	0x0000 031C	0x4843 C31C
RESERVED	R	32	0x0000 0320	0x4843 C320
ATL_PPMR3	RW	32	0x0000 0380	0x4843 C380
ATL_BBSR3	R	32	0x0000 0384	0x4843 C384
ATL_ATLCR3	RW	32	0x0000 0388	0x4843 C388
RESERVED	R	32	0x0000 038C	0x4843 C38C
ATL_SWEN3	RW	32	0x0000 0390	0x4843 C390
ATL_BWSMUX3	RW	32	0x0000 0394	0x4843 C394

Table 31-7. ATL Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ATL Physical Address
ATL_AWSMUX3	RW	32	0x0000 0398	0x4843 C398
ATL_PCLKMUX3	RW	32	0x0000 039C	0x4843 C39C
RESERVED	R	32	0x0000 03A0	0x4843 C3A0

31.5.3 ATL Register Description

Table 31-8 through Table 31-36 describe the individual ATL module registers.

Table 31-8. ATL_REVID

Address Offset	0x0000 0000		
Physical Address	0x4843 C000	**Instance**	ATL
Description	ATL IP revision. Value is hard wired and revised for each new IP release		
Type	R		

31–0
REV

Bits	Field Name	Description	Type	Reset
31:0	REV	Identifies the ATL revision.	R	0x0000 0000

Table 31-9. ATL_PPMR0

Address Offset	0x0000 0200		
Physical Address	0x4843 C200	**Instance**	ATL
Description	Parts per million register for the first ATL instance. This register is used by the adjusting circuit. The Parts-Per-Million Registers 0-2 (PPMRn) are used by the audio re-timing code to adjust the DAC clocking rate by changing the McASP clocks. The McASP high-speed clock slows down or speeds up by the PPM written to PPM_SETTING bits [8:0], which changes the DAC over-sampling clock or the frame sync and bit clocks.		
Type	RW		

31–16	15	14–9	8–0
RESERVED	PPM_SLOWDOWN	RESERVED	PPM_SETTING

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PPM_SLOWDOWN	Part-Per-Million Slowdown 0x0: Speed up 0x1: Slow down	RW	0
14:9	RESERVED		R	0x00
8:0	PPM_SETTING	PPM_SETTING PPM adjustment = PPMR[8:0] ÷ 2^{20}		

Table 31-10. ATL_BBSR0

Address Offset	0x0000 0204		
Physical Address	0x4843 C204	**Instance**	ATL

Table 31-10. ATL_BBSR0 (continued)

Description	Baseband sample register for the first ATL instance.
Type	R

31–16	15–0
RESERVED	SAMPLE_COUNT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	SAMPLE_COUNT	The measuring circuit produces a 16- bit Sample Count that is stored in the BBSR register. Its inputs are the Audio IIS Word Select (AWS) and the Baseband IIS Word Select (BWS) from their respective McASP pins.	R	0x0000

Table 31-11. ATL_ATLCR0

Address Offset	0x0000 0208		
Physical Address	0x4843 C208	**Instance**	ATL
Description	ATL configuration register for first ATL instance.		
Type	RW		

31–5	4–0
RESERVED	ATL_INTERNAL_DIVIDER

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000 00
4:0	ATL_INTERNAL_DIVIDER	ATL Internal Divider. Sets ATLPCLK-to-ATCLK ratio. The ATL Internal Divider divides down the ATL master clock (ATLPCLK) to generate ATCLK.	RW	0x00

Table 31-12. ATL_SWEN0

Address Offset	0x0000 0210		
Physical Address	0x4843 C210	**Instance**	ATL
Description	Software enable bit for the first ATL instance. The audio tracking logic (ATL) must be held in reset while the AWS, BWS, and ATPCLK mux selections are set, and while the system level clocking is configured.		
Type	RW		

31–1	0
RESERVED	SWEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	SWEN	Software enable bit. The software must enable this bit to enable the first ATL instance. 0x0: ATL0 disabled 0x1: ATL0 enabled	RW	0

Table 31-13. ATL_BWSMUX0

Address Offset	0x0000 0214		
Physical Address	0x4843 C214	**Instance**	ATL
Description	Select source for BWS input to first ATL instance. The ATL baseband word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the first ATL instance.		

Table 31-13. ATL_BWSMUX0 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												BWSMUX			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 0000
3:0	BWSMUX	Baseband Word Select Mux. Selects the source for the BWS input to the first ATL instance. 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX 0x9: MCASP8_FSX 0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1	RW	0

Table 31-14. ATL_AWSMUX0

Address Offset	0x0000 0218		
Physical Address	0x4843 C218	**Instance**	ATL
Description	Select source for AWS input to the first ATL instance. The ATL audio word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the first ATL instance.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												AWSMUX			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 0000
3:0	AWSMUX	Audio Word Select Mux. Selects the source for the AWS input to the first ATL instance. 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX	RW	0

Bits	Field Name	Description	Type	Reset
		0x9: MCASP8_FSX 0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1		

Table 31-15. ATL_PCLKMUX0

Address Offset	0x0000 021C		
Physical Address	0x4843 C21C	**Instance**	ATL
Description	Select source for ATLPCLK input to all four ATL instances. The ATL clock input can be driven from with the OCP_CLK or the ATLPCLK inputs. The ATLPCLK selection register determines the final clock source for the ATL core audio tracking logic. In most use cases this will be set to the ATLPCLK input to use the system clocks shown in the device specific data manual.		
Type	RW		

31:1	0
RESERVED	PCLKMUX

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	PCLKMUX	ATLPCLK Selection Register Mux. Selects the source for the ATLPCLK input to all four ATL instances. 0x0: OCP_CLK input 0x1: ATLPCLK input	RW	0

Table 31-16. ATL_PPMR1

Address Offset	0x0000 0280		
Physical Address	0x4843 C280	**Instance**	ATL
Description	Parts per million register for the second ATL instance. This register is used by the adjusting circuit. The Parts-Per-Million Registers 0-2 (PPMRn) are used by the audio re-timing code to adjust the DAC clocking rate by changing the McASP clocks. The McASP high-speed clock slows down or speeds up by the PPM written to PPM_SETTING bits [8:0], which changes the DAC over-sampling clock or the frame sync and bit clocks.		
Type	RW		

31:16	15	14:9	8:0
RESERVED	PPM_SLOWDOWN	RESERVED	PPM_SETTING

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
15	PPM_SLOWDOWN	Part-Per-Million Slowdown 0x0: Speed up 0x1: Slow down	RW	0
14:9	RESERVED		R	0x00
8:0	PPM_SETTING	PPM_SETTING PPM adjustment = PPMR[8:0] ÷ 2^{20}		

Table 31-17. ATL_BBSR1

Address Offset	0x0000 0284		
Physical Address	0x4843 C284	**Instance**	ATL
Description	Baseband sample register for the second ATL instance.		
Type	R		

31:16	15:0
RESERVED	SAMPLE_COUNT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	SAMPLE_COUNT	The measuring circuit produces a 16- bit Sample Count that is stored in the BBSR register. Its inputs are the Audio IIS Word Select (AWS) and the Baseband IIS Word Select (BWS) from their respective McASP pins.	R	0x0000

Table 31-18. ATL_ATLCR1

Address Offset	0x0000 0288		
Physical Address	0x4843 C288	**Instance**	ATL
Description	ATL configuration register for the second ATL instance.		
Type	RW		

31:5	4:0
RESERVED	ATL_INTERNAL_DIVIDER

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000 00
4:0	ATL_INTERNAL_DIVIDER	ATL Internal Divider. Sets ATLPCLK-to-ATCLK ratio. The ATL Internal Divider divides down the ATL master clock (ATLPCLK) to generate ATCLK.	RW	0x00

Table 31-19. ATL_SWEN1

Address Offset	0x0000 0290		
Physical Address	0x4843 C290	**Instance**	ATL
Description	Software enable bit for the second ATL instance. The audio tracking logic (ATL) must be held in reset while the AWS, BWS, and ATPCLK mux selections are set, and while the system level clocking is configured.		
Type	RW		

31:1	0
RESERVED	SWEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	SWEN	Software enable bit. The software must enable this bit to enable the second ATL instance. 0x0: ATL1 disabled 0x1: ATL1 enabled	RW	0

Table 31-20. ATL_BWSMUX1

Address Offset	0x0000 0294		
Physical Address	0x4843 C294	**Instance**	ATL
Description	Select source for BWS input to the second ATL instance. The ATL baseband word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the second ATL instance.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												BWSMUX			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 0000
3:0	BWSMUX	Baseband Word Select Mux. Selects the source for the BWS input to the second ATL instance. 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX 0x9: MCASP8_FSX 0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1	RW	0

Table 31-21. ATL_AWSMUX1

Address Offset	0x0000 0298		
Physical Address	0x4843 C298	**Instance**	ATL
Description	Select source for AWS input to the second ATL instance. The ATL audio word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the second ATL instance.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												AWSMUX			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 0000

Bits	Field Name	Description	Type	Reset
3:0	AWSMUX	Audio Word Select Mux. Selects the source for the AWS input to the second ATL instance. 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX 0x9: MCASP8_FSX 0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1	RW	0

Table 31-22. ATL_PCLKMUX1

Address Offset	0x0000 029C		
Physical Address	0x4843 C29C	**Instance**	ATL
Description	Select source for ATLPCLK input to the second ATL instance. The ATL clock input can be driven from with the OCP_CLK or the ATLPCLK inputs. The ATLPCLK selection register determines the final clock source for the ATL core audio tracking logic. In most use cases this will be set to the ATLPCLK input to use the system clocks shown in the device specific data manual.		
Type	RW		

31–1	0
RESERVED	PCLKMUX

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	PCLKMUX	ATLPCLK Selection Register Mux. Selects the source for the ATLPCLK input to the second ATL instance. (Not functional). 0x0: OCP_CLK input 0x1: ATLPCLK input	RW	0

Table 31-23. ATL_PPMR2

Address Offset	0x0000 0300		
Physical Address	0x4843 C300	**Instance**	ATL
Description	Parts per million register for the third ATL instance. This register is used by the adjusting circuit. The Parts-Per-Million Registers 0-2 (PPMRn) are used by the audio re-timing code to adjust the DAC clocking rate by changing the McASP clocks. The McASP high-speed clock slows down or speeds up by the PPM written to PPM_SETTING bits [8:0], which changes the DAC over-sampling clock or the frame sync and bit clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PPM_SLOWDOWN	RESERVED	PPM_SETTING

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PPM_SLOWDOWN	Part-Per-Million Slowdown 0x0: Speed up 0x1: Slow down	RW	0
14:9	RESERVED		R	0x00
8:0	PPM_SETTING	PPM_SETTING PPM adjustment = PPMR[8:0] ÷ 2^{20}		

Table 31-24. ATL_BBSR2

Address Offset	0x0000 0304		
Physical Address	0x4843 C304	**Instance**	ATL
Description	Baseband sample register for third ATL instance.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SAMPLE_COUNT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	SAMPLE_COUNT	The measuring circuit produces a 16- bit Sample Count that is stored in the BBSR register. Its inputs are the Audio IIS Word Select (AWS) and the Baseband IIS Word Select (BWS) from their respective McASP pins.	R	0x0000

Table 31-25. ATL_ATLCR2

Address Offset	0x0000 0308		
Physical Address	0x4843 C308	**Instance**	ATL
Description	ATL configuration register for the third ATL instance.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ATL_INTERNAL_DIVIDER				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000 00
4:0	ATL_INTERNAL_DIVIDER	ATL Internal Divider. Sets ATLPCLK-to-ATCLK ratio. The ATL Internal Divider divides down the ATL master clock (ATLPCLK) to generate ATCLK.	RW	0x00

Table 31-26. ATL_SWEN2

Address Offset	0x0000 0310		
Physical Address	0x4843 C310	**Instance**	ATL

Table 31-26. ATL_SWEN2 (continued)

Description	Software enable bit for the third ATL instance. The audio tracking logic (ATL) must be held in reset while the AWS, BWS, and ATPCLK mux selections are set, and while the system level clocking is configured.
Type	RW

31–1	0
RESERVED	SWEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	SWEN	Software enable bit. The software must enable this bit to enable the third ATL instance. 0x0: ATL2 disabled 0x1: ATL2 enabled	RW	0

Table 31-27. ATL_BWSMUX2

Address Offset	0x0000 0314		
Physical Address	0x4843 C314	**Instance**	ATL
Description	Select source for BWS input to to the third ATL instance. The ATL baseband word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the third ATL instance.		
Type	RW		

31–4	3–0
RESERVED	BWSMUX

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 000
3:0	BWSMUX	Baseband Word Select Mux. Selects the source for the BWS input to the third ATL instance. 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX 0x9: MCASP8_FSX 0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1	RW	0x0

Table 31-28. ATL_AWSMUX2

Address Offset	0x0000 0318		
Physical Address	0x4843 C318	**Instance**	ATL

Table 31-28. ATL_AWSMUX2 (continued)

Description	Select source for AWS input to the third ATL instance. The ATL audio word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the third ATL instance.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												AWSMUX			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 0000
3:0	AWSMUX	Audio Word Select Mux. Selects the source for the AWS input to the third ATL instance. 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX 0x9: MCASP8_FSX 0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1	RW	0

Table 31-29. ATL_PCLKMUX2

Address Offset	0x0000 031C		
Physical Address	0x4843 C31C	**Instance**	ATL
Description	Select source for ATLPCLK input to the third ATL instance. The ATL clock input can be driven from with the OCP_CLK or the ATLPCLK inputs. The ATLPCLK selection register determines the final clock source for the ATL core audio tracking logic. In most use cases this will be set to the ATLPCLK input to use the system clocks shown in the device specific data manual.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															PCLKMUX

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	PCLKMUX	ATLPCLK Selection Register Mux. Selects the source for the ATLPCLK input to the third instance. (Not functional). 0x0: OCP_CLK input 0x1: ATLPCLK input	RW	0

Table 31-30. ATL_PPMR3

Address Offset	0x0000 0380		
Physical Address	0x4843 C380	**Instance**	ATL
Description	Parts per million register fourth ATL instance. This register is used by the adjusting circuit. The Parts-Per-Million Registers 0-2 (PPMRn) are used by the audio re-timing code to adjust the DAC clocking rate by changing the McASP clocks. The McASP high-speed clock slows down or speeds up by the PPM written to PPM_SETTING bits [8:0], which changes the DAC over-sampling clock or the frame sync and bit clocks.		
Type	RW		

31:16	15	14:9	8:0
RESERVED	PPM_SLOWDOWN	RESERVED	PPM_SETTING

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PPM_SLOWDOWN	Part-Per-Million Slowdown 0x0: Speed up 0x1: Slow down	RW	0
14:9	RESERVED		R	0x00
8:0	PPM_SETTING	PPM_SETTING PPM adjustment = PPMR[8:0] ÷ 2^{20}		

Table 31-31. ATL_BBSR3

Address Offset	0x0000 0384		
Physical Address	0x4843 C384	**Instance**	ATL
Description	Baseband sample register fourth ATL instance.		
Type	R		

31:16	15:0
RESERVED	SAMPLE_COUNT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	SAMPLE_COUNT	The measuring circuit produces a 16- bit Sample Count that is stored in the BBSR register. Its inputs are the Audio IIS Word Select (AWS) and the Baseband IIS Word Select (BWS) from their respective McASP pins.	R	0x0000

Table 31-32. ATL_ATLCR3

Address Offset	0x0000 0388		
Physical Address	0x4843 C388	**Instance**	ATL
Description	ATL configuration register fourth ATL instance.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	ATL_INTERNAL_DIVIDER

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000 00
4:0	ATL_INTERNAL_DIVIDER	ATL Internal Divider. Sets ATLPCLK-to-ATCLK ratio. The ATL Internal Divider divides down the ATL master clock (ATLPCLK) to generate ATCLK.	RW	0x00

Table 31-33. ATL_SWEN3

Address Offset	0x0000 0390		
Physical Address	0x4843 C390	**Instance**	ATL
Description	Software enable bit for fourth ATL instance. The audio tracking logic (ATL) must be held in reset while the AWS, BWS, and ATPCLK mux selections are set, and while the system level clocking is configured.		
Type	RW		

31–1	0
RESERVED	SWEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	SWEN	Software enable bit. The software must enable this bit to enable the fourth ATL instance. 0x0: ATL3 disabled 0x1: ATL3 enabled	RW	0

Table 31-34. ATL_BWSMUX3

Address Offset	0x0000 0394		
Physical Address	0x4843 C394	**Instance**	ATL
Description	Select source for BWS input to fourth instance ATL. The ATL baseband word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the fourth ATL instance.		
Type	RW		

31–4	3–0
RESERVED	BWSMUX

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 000
3:0	BWSMUX	Baseband Word Select Mux. Selects the source for the BWS input to the fourth ATL instance. 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX 0x9: MCASP8_FSX	RW	0x0

Bits	Field Name	Description	Type	Reset
		0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1		

Table 31-35. ATL_AWSMUX3

Address Offset	0x0000 0398		
Physical Address	0x4843 C398	**Instance**	ATL
Description	Select source for AWS input to fourth instance ATL. The ATL audio word input can be connected to up to 15 different input sources. This register must be configured to set the McASP (or other input) desired for the fourth ATL instance.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												AWSMUX			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000 0000
3:0	AWSMUX	Audio Word Select Mux. Selects the source for the AWS input to the fourth ATL instance 0x0: MCASP1_FSR 0x1: MCASP1_FSX 0x2: MCASP2_FSR 0x3: MCASP2_FSX 0x4: MCASP3_FSX 0x5: MCASP4_FSX 0x6: MCASP5_FSX 0x7: MCASP6_FSX 0x8: MCASP7_FSX 0x9: MCASP8_FSX 0xA: MCASP8_AHCLKX 0xB: XREF_CLK3 input pad 0xC: XREF_CLK0 input pad 0xD: XREF_CLK1 input pad 0xE: XREF_CLK2 input pad 0xF: SYS_CLK1	RW	0

Table 31-36. ATL_PCLKMUX3

Address Offset	0x0000 039C		
Physical Address	0x4843 C39C	**Instance**	ATL
Description	Select source for ATLPCLK input to fourth instance ATL. The ATL clock input can be driven from with the OCP_CLK or the ATLPCLK inputs. The ATLPCLK selection register determines the final clock source for the ATL core audio tracking logic. In most use cases this will be set to the ATLPCLK input to use the system clocks shown in the device specific data manual.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	PC LK M UX

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	PCLKMUX	ATLPCLK Selection Register Mux. Selects the source for the ATLPCLK input to the fourth ATL instance. (Not functional). 0x0: OCP_CLK input 0x1: ATLPCLK input	RW	0

Chapter 32
Initialization

This chapter introduces the steps of the device initialization.

32.1 Initialization Overview 6937
32.2 Preinitialization 6939
32.3 Device Initialization by ROM Code 6951
32.4 Services for HLOS Support 7020

32.1 Initialization Overview

This chapter provides an overview of the requirements to initialize the device from power on to operating system (OS) and application execution, the overall initialization process (including hardware- and software-related steps), the general ROM code operational requirements, and behavior expectations.

32.1.1 Terminology

- **Bootstrap:** Initial software launched by the ROM code during the memory booting phase
- **Configuration Header (CH):** Optional structure that precedes the initial software and allows the redefinition of the ROM code default settings
- **Downloaded software:** Initial software downloaded into the internal static RAM (SRAM) by the ROM code during the peripheral booting phase
- **eFuse:** A one-time programmable memory location usually set at the factory
- **Flash loader:** Downloaded software launched by the ROM code during the preflashing stage. It also programs an image in external memories.
- **Initial software:** Software executed by any of the ROM code mechanisms (memory booting or peripheral booting). Initial software is a generic term for bootstrap and downloaded software.
- **Memory booting:** ROM code mechanism that consists of executing initial software from external memory
- **Master CPU:** The Arm® Cortex®-A15 MPCore™ CPU for which CPU-ID is 0. It configures the multicore platform and starts the ROM code to ensure device booting from a mass storage memory (memory booting) or a peripheral interface (peripheral booting).
- **Peripheral booting:** ROM code mechanism that consists of polling selected interfaces, downloading, and executing initial software (in this case, downloaded software) in the internal RAM
- **Permanent booting device:** Memory device containing, by default, the image to be executed during the booting sequence. It is the default memory booting device.
- **Preflashing:** A specific case of peripheral booting where the ROM code mechanism is used to program the external flash memory
- **ROM Code:** The on-chip software in device ROM that implements booting
- **ROM Code-controlled Boot Phase:** This phase covers the sequence operations from the time the platform releases the reset to the time first user- or customer-owned software starts execution. This phase is fully controlled by the device ROM code.
- **Slave CPU:** The Arm Cortex-A15 MPCore CPU for which CPU-ID is 1. It is brought to the wait-for-event (WFE) state by the ROM code, waiting to be woken up by the master CPU.

32.1.2 Initialization Process

Figure 32-1 is an overview of the initialization process and its steps:

- **Preinitialization:** Power, clock, and control connections must be present, and the boot configuration pins must be held at the desired logical levels.
- **Power, clock, reset ramp sequence:** Specific sequence that is applied by the power-management chip
- **ROM code:** Responsible for finding, for downloading, and for executing the initial software by using the master CPU
- **Initial software:** Software that prepares and passes control to application software or to the high-level operating system (HLOS)
- **Symmetric multiprocessing (SMP)-capable HLOS** or application (primarily for diagnostics)

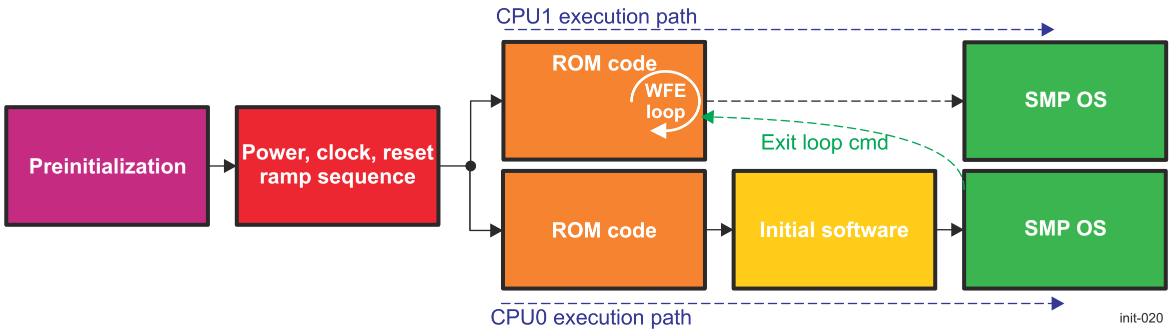


Figure 32-1. Initialization Process

The first two steps in the initialization process are hardware-oriented; however, they require an understanding of the process of configuring these system interface pins (balls on the device), which have software-configurable functionality. This configuration is an essential part of the chip configuration and is application-dependent. This chapter discusses these system-interface pins and the associated configuration registers that are vital to the correct initialization of the device.

32.2 Preinitialization

To accomplish a successful boot-up operation , certain hardware configuration settings must be in place. Clock, reset, and power connections, as well as pins involved in setting the boot memory space for the master CPU, must be connected and driven correctly to successfully ini

tialize the device. The following sections describe the specific requirements for the preinitialization stage.

32.2.1 Power Requirements

The device can be supplied by an external power-management integrated circuit (PMIC). TI provides a global solution with the device connected to the power-management IC companion chip. Refer to *Data Manual* for information about the power-management IC companion chips supported for this device.

Figure 32-2 shows typical power connections between the device and a PMIC companion chip.

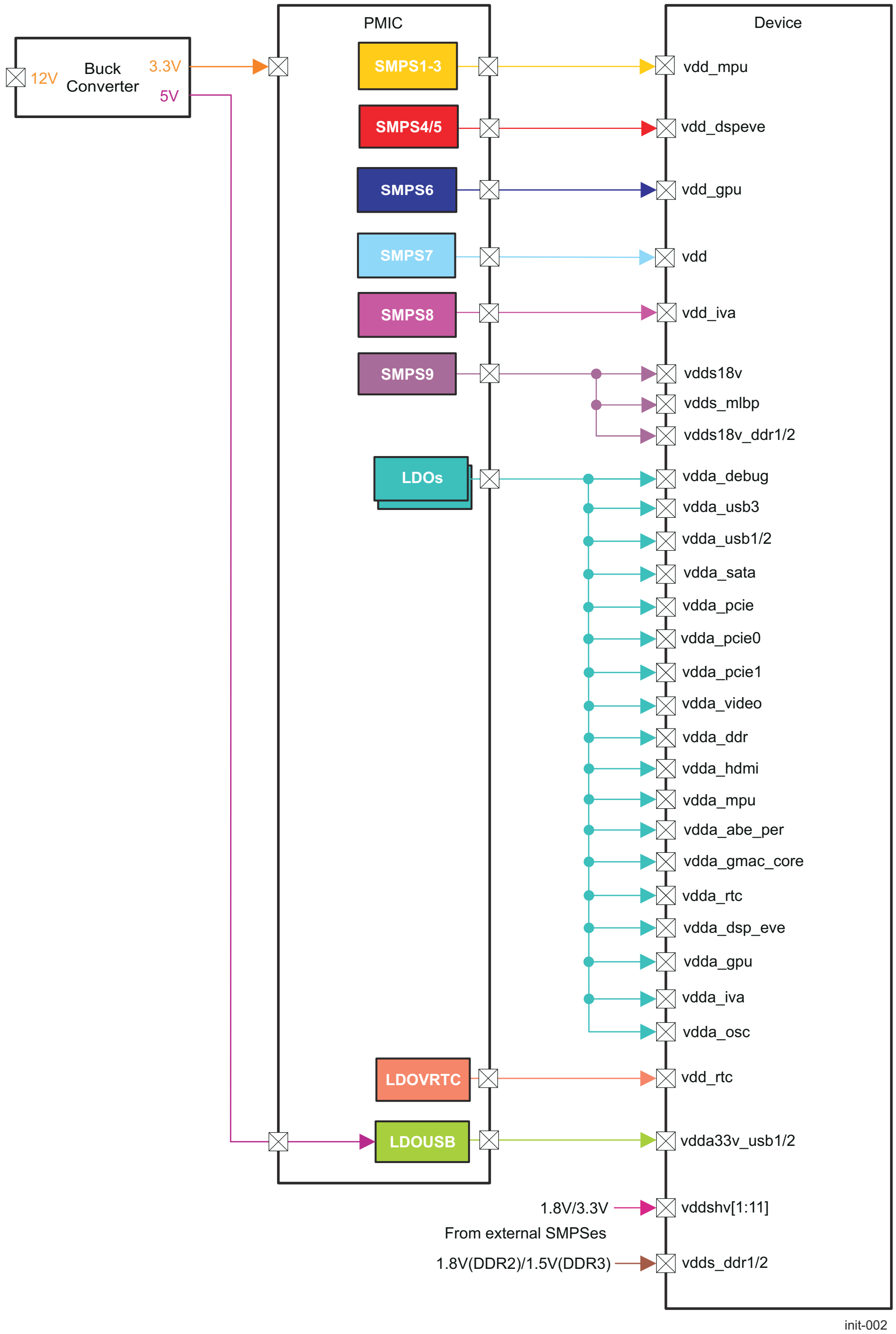


Figure 32-2. Power Supply Connections Example

Note

Figure 32-2 is an example of power connections between the device and the PMIC, representing only one of the multiple PMIC/OTP options.
These connections depend on the actual application, PMIC, and OTP used. Refer to the Device *Data Manual*, respective PMIC Data Sheet, and all related application notes before starting a new design.

Table 32-1 describes the device power balls.

Table 32-1. Device Power Balls

Voltage Ball Name	Subsystems and Peripherals
vdd	Subsystems and modules supplied by CORE voltage domain
vdd_mpu	MPU voltage domain
vdd_iva	IVA voltage domain
vdd_dspeve	DSP and EVE voltage domain
vdd_gpu	GPU voltage domain
vdd_rtc	RTC voltage domain
vdds18v	1.8V I/Os
vdds_mlbp	MLBSS I/Os
vdds_ddr1	EMIF channel 1 (1.8V for DDR2 mode/ 1.5V for DDR3 mode)
vdds_ddr2	EMIF channel 2 (1.8V for DDR2 mode/ 1.5V for DDR3 mode)
vdds18v_ddr1	EMIF channel 1 bias
vdds18v_ddr2	EMIF channel 2 bias
ddr1_vref0	EMIF channel 1 vref
ddr2_vref0	EMIF channel 2 vref
vdda_video	Analog power supply for DPLL_VIDEO0/1
vdda_ddr	Analog power supply for DPLL_DDR and DDR HSDIVIDER
vdda_mpu	Analog power supply for DPLL_MPU
vdda_abe_per	Analog power supply for DPLL_ABE, DPLL_PER, PER HSDIVIDER
vdda_gmac_core	Analog power supply for DPLL_GMAC DPLL_CORE, and CORE HSDIVIDER
vdda_hdmi	Analog power supply for HDMI I/Os and DPLL_HDMI
vdda_pcie	Analog power supply for DPLL_PCIE_REF and APLL_PCIE
vdda_pcie0	Analog power supply for PCIE0 I/Os
vdda_pcie1	Analog power supply for PCIE1 I/Os
vdda_iva	Analog power supply for DPLL_IVA
vdda_gpu	Analog power supply for DPLL_GPU
vdda_dsp_eve	Analog power supply for DPLL_DSP and DPLL_EVE
vdda_usb3	Analog power supply for USB1 DPLL_USB_OTG_SS and USB3.0 I/Os
vdda_usb1	Analog power supply for USB1 DPLL_USB and USB2.0 I/Os (1.8 V)
vdda33v_usb1	Analog power supply for USB1 I/Os (3.3 V)
vdda_usb2	Analog power supply for USB2 I/Os (1.8 V)
vdda33v_usb2	Analog power supply for USB2 I/Os (3.3 V)
vdda_sata	Analog power supply for SATA I/Os and DPLL_SATA
vdda_debug	Analog power supply for DPLL_DEBUG
vddshv1	Dual-voltage VIN2 group I/Os
vddshv2	Dual-voltage VOUT groupI/Os
vddshv3	Dual-voltage GENERAL group I/Os
vddshv4	Dual-voltage MMC4 group I/Os
vddshv5	Dual-voltage RTC group I/Os
vddshv6	Dual-voltage VIN1 group I/Os
vddshv7	Dual-voltage WIFI group I/Os
vddshv8	Dual-voltage MMC1 group I/Os
vddshv9	Dual-voltage RGMII group I/Os
vddshv10	Dual-voltage GPMC group I/Os
vddshv11	Dual-voltage MMC2 group I/Os
vdda_osc	System HF OSC0/1 XTAL oscillators

Table 32-1. Device Power Balls (continued)

Voltage Ball Name	Subsystems and Peripherals
vdda_rtc	RTC bias and LF RTC XTAL oscillator

Note

Table 32-1 is for informative purposes. For a complete description of the power balls of your device, please refer to the *Device Data Manual*.

For more information about power management, see *Clock Management Functional Description*, in *Power, Reset, and Clock Management*.

32.2.2 Interaction With the PMIC Companion

The ROM code does not perform any I^2C transactions with the PMIC. The following system conditions must be met to perform device initialization:

- The USB transceivers (USB2.0), internal to device, are powered on reset, as expected by the USB peripheral booting feature.
- The SD card cage must be appropriately powered before entering the SD card boot feature on any reset.
- Devices must be appropriately powered and be up and ready at platform startup:
  – eMMC
  – QSPI
  – GPMC
  – SATA

Note

The ROM code does not interact with the PMIC companion chip over an I^2C or SPI interface.

32.2.3 Clock, Reset, and Control

32.2.3.1 Overview

Figure 32-3 shows the clock and reset environment where clocks and reset-related signals are gathered at the system level, the system-expansion signals, and the crystal oscillator connection.

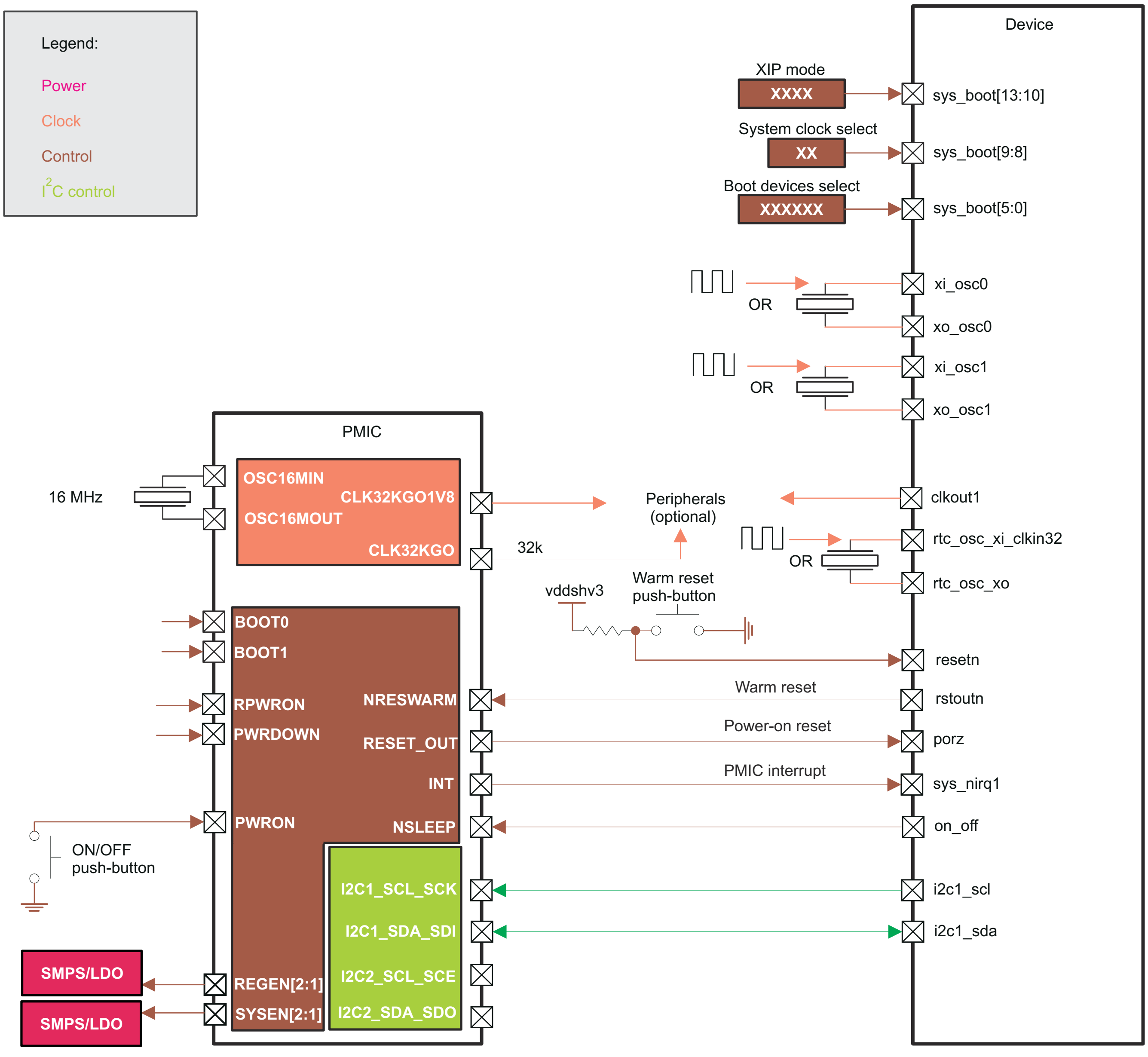


Figure 32-3. Clock, Reset, and Control Environment Overview

Note

Figure 32-3 is a typical example of clock, reset and control connections between the device and a PMIC. Refer to the Device *Data Manual* for the supported PMIC(s) for your device and for more information on these balls.
For PMIC ball description, see the respective PMIC Data Sheet.

The main features of the system interface are:

- Accepts crystals connected between device xi_osc0 and xo_osc0 pads, and xi_osc1 and xo_osc1 pads to generate SYS_CLK1 and SYS_CLK2, resepectively
- Accepts external LVCMOS clock sources connected to xi_osc0, and xi_osc1 pads for SYS_CLK1 and SYS_CLK2, respectively
- A 32-kHz LVCMOS clock input for RTC operation
- Up to four reference clock inputs
- Up to three configurable clock outputs
- sysboot[15:0] input signals to define the boot mode, system clock speed, and GPMC in XIP mode
- Two reset sources
  – Power-on reset (cold reset)
  – Warm reset
- Three external interrupt lines (sys_irq1, sys_irq2, and nmi)
- Four external DMA requests

32.2.3.2 Clocking Scheme

The device operation requires external input clocks, as follows:

- 32k clock: A 16-MHz crystal is connected to the PMIC companion chip that embeds the 32k-oscillator (a square CMOS 32-kHz clock can also be delivered on the OSC16MIN pin of Power-management IC, if the 32k-crystal is connected on another device of the system). The resulting 32k-clock is delivered to the entire system on two outputs:
  – CLK32KGO
  – CLK32KGO1V8
- System clocks: The device supports two system clocks with two clock sources each:
  – SYS_CLK1 (main system clock):
    - Crystal on OSC0 pins. An internal oscillator (OSC0) embedded in the device is used.
    - External LVCMOS clock on xi_osc0
  – SYS_CLK2 (optional system clock):
    - Crystal on OSC1 pins. An internal oscillator (OSC1) embedded in the device is used.
    - External LVCMOS clock on xi_osc1
- The device can deliver digital clocks to peripherals ICs.

The device provides wide choice of clocks that can be delivered on clkout[1:3] pads to companion devices. For more information, see Section 3.3, *PRCM Subsystem Environment*, in Chapter 3, *Power, Reset, and Clock Management*. For more information on pad multiplexing, see Chapter 18, *Control Module*.

Table 32-2 lists the mapping for the device clock input sources. Table 32-3 lists the PMIC clock requirements.

Table 32-2. Mapping for Input Sources

Clock	Clock Source	Ball Mapping	Frequency Range/List	Type
SYS_CLK1	Internal oscillator 0 (OSC0)	xi_osc0 and xo_osc0	19.2, 20, and 27 MHz	Crystal connection pins
	External	xi_osc0	19.2, 20, and 27 MHz	External LVCMOS
SYS_CLK2	Internal oscillator 1 (OSC1)	xi_osc1 and xo_osc1	19.2 ÷ 32 MHz	Crystal connection pins
	External	xi_osc1	12 ÷ 38.4 MHz	External LVCMOS

Table 32-3. PMIC Clock Requirements

Clock Source(1)	Mapping	Frequency Range/List	Type
Internal oscillator	OSC16MIN and OSC16MOUT	16.384 MHz	Crystal connection pins
External	OSC16MIN	32.768 kHz (nom.)	External LVCMOS

(1) The PMIC does not need any external clock to operate properly. PMIC can run on internal RC oscillator. The 16-MHz oscillator is in place to provide an accurate 32-kHz clock to its internal RTC, to the SoC or device in the system. See the corresponding PMIC datasheet for details.

> **CAUTION**
>
> Clock configurations depend on core voltage, and maximum clock frequencies may not apply to production. For more information, see the *Device Data Manual*.

32.2.3.3 Reset Configuration

32.2.3.3.1 ON/OFF Interconnect and Power-On-Reset

The entire system is typically awakened by an ON/OFF push button connected to the PMIC chip. This signal belongs to the VSYS - system power domain and is active low (the PMIC internal pullup ties it to VSYS). The PMIC power-up event is propagated through its NRESPWRON output pin to the device porz pad (that is, the PRM SYS_PWRON_RST signal) when the PMIC power-up sequence is achieved. The device porz input pin is held low all the time during VDD core and I/O power-up.

32.2.3.3.2 Warm Reset

A warm reset can be asserted by the device, by an external button (typically for a development platform), or by any other chip connected to it (normally tied to PMIC companion NRESWARM output pin).

The device warm reset pad (resetn - device signal SYS_WARM_IN_RST) is used to trigger a warm reset on the device, which resets part of the device when it has already booted (for example, to recover from a software crash). When an internal device reset occurs, SYS_NRES_WARM_OUT output and device pad rstoutn go low and reset all the peripherals.

The device releases the SYS_NRES_WARM_OUT output signal after SYS_PWRON_RST is deasserted.

32.2.3.3.3 Peripheral Reset by GPIO

Most peripherals can be reset and powered on or off by GPIO. By default, under POR, most device signals are in safe mode with a default value driven by the I/O cell. The value is driven by an internal pullup or pulldown. Depending on the peripheral reset active level, users must select one GPIO or another (according to the reset value).

Once POR is released, the value on the pad is driven by the default configuration of the device control module. Most of time, this configuration is aligned with the default value selected on the I/O cell.

The next step is application-dependent: Users must configure the device registers to validate GPIO use and the default configuration of the control module.

32.2.3.3.4 Warm Reset Impact on GPIOs

When a warm reset event occurs:

- The GPIO controller is reset. Consequently, the GPIO is automatically turned in input mode.
- The control module is not reset. Information related to signal multiplexing mode and pullup or pulldown configuration is still valid.

Therefore, when a warm reset event occurs, the output buffer is disabled. Consequently, two different behaviors can be defined with regard to what is expected by the platform:

- GPIO sensitive to warm reset:

  To prevent a floating pad, user software is designed to have the internal pad PU and PD resistors enabled immediately, before the software warm reset action, because the warm reset-sensitive GPIO controllers will change I/O direction to input after an device warm reset. This is necessary if the warm reset-sensitive GPIO controller pin has been configured for output before the warm reset occurrence. The pulls-enabled-before-warm-reset condition should be set by default in case the user has configured a GPIO as an input, because in this case the user is expected to have enabled the internal PU and PD pads during GPIO configuration (unless external pull resistors were used).

Note

If the PU and PD resistors are enabled immediately by software after a POR (cold reset) for a GPIO that is planned to be used only as an output, then unnecessary power consumption can occur.

While the dynamically-enable-the-pull-just-before-warm-reset condition is possible during a software warm reset (because the user software is aware of the exact moment a warm reset event occurs), it is not possible when the warm reset is triggered by hardware (for example, a watchdog reset, SYS_NRESWARM signal assertion, and so forth), because the software is not aware of the exact moment of these warm reset assertions.

- GPIO not sensitive to warm reset:

  To avoid getting a floating signal during (and after) a warm reset event and to keep the same value that was driven before the reset, users must align the pull value with the drive value each time a dedicated GPIO register is accessed.

Note

To avoid unnecessary consumption, the user software must ensure that internal pull resistor is disabled when the GPIO buffer is driving.

For the description of the reset sequences and information about the device reset management, see Section 3.5, *Reset Management Functional Description*, in Chapter 3, *Power, Reset and Clock Management*. For more information about the device reset sequences, see the device *Data Manual*.

32.2.3.4 PMIC Control

- I^2C:

  The device interfaces: system interface I/Os, and I2C1 are involved in system interactions between the device and external power, reset and clock management IC companions.

  The device and PMIC companion implement the basic power-management interface, as follows:
  - 32-kHz clock single input
  - Two system resets: power-on (cold) reset and warm reset
  - One interrupt
- INT:

  PMIC companion device can activate its output interrupt request signal (INT) at any time when requires the device to monitor its activity. When receiving such an interruption, the device checks, through the I^2C, to determine the source of the interrupt. INT pin is active low.

32.2.3.5 PMIC Request Signals

The PMIC drives three external enable-output signals, which allows switching on some external resources at different stages of the power-up sequence:

- REGEN1 and REGEN2 are driven high at the beginning of the power-up sequence, before any internal power source is turned on. They belong to the VSYS power domain. REGEN1 can typically be used for buck boost control.
- SYSEN is driven high immediately after the VCORE power output is turned on. SYSEN belongs to the VIO power domain.

The PMIC companion chip can receive two power resource requests: ENABLE1 and NSLEEP. These pins allow an external device to request PMIC internal resources. The PMIC companion power behavior when pins are activated must be programmed after the first boot (resources and ENABLE pins allocated by group, resource behavior upon group activation, and so forth).

The use of NSLEEP is especially required when PMIC chip is in sleep mode, because it cannot handle an I^2C command. Any device that requires the PMIC companion resource to wake up must first activate its associated ENABLE signal. All power-management chip power regulators are off or in sleep mode when it is in sleep mode.

32.2.4 Sysboot Configuration

The device implements 16 sampled-on-reset sysboot pads.

Table 32-4. Sysboot Pads Description

Pad (1)	Description
sysboot[15]	Must be pulled to vdd for proper device operation (SR1.1). Used to permanently disable the internal PU/PD resistors on pads gpmc_a[27:24, 22:19] (SR2.0).
sysboot[14]	Must be pulled to vss for proper device operation.
sysboot[13:10]	Used to configure the GPMC interface when booting from XIP/NAND memory connected to GPMC.
sysboot[9:8]	Selects the SYS_CLK1 clock speed. Must be set correctly according to the speed of the connected crystal.
sysboot[7:6]	Sector offset for the location of the redundant SBL images in QSPI.
sysboot[5:0]	Select interfaces or devices for the booting list

(1) Boards should be implemented with a mechanism to easily modify the pull-up/down state to enable any future modifications.

Table 32-5. MMC2 Configuration (SR2.0)

sysboot[15] (1)	mmc2_dat[7:0] Pull-down resistors
0b0	Software re-configuration of pull resistors is allowed. Internal pull-downs on gpmc_a[n:19] are enabled by default to allow GPMC boot. Pulling low gpmc_a[n:19] is required in order to access the low-order address locations in the flash memory during boot (n = [27:24, 22:19] and depends on the memory volume).
0b1	Internal pull-down resistors permanently disabled to avoid contention with the recommended per eMMC standard pull-ups that should be present on PCB. Software re-configuration of internal pull resistors is disabled.

(1) When internal pull-downs are permanently disabled (sysboot[15] = 1), gpmc_a[n:19] pads (where n = [27:24, 22:19]) require external pull-downs to enable GPMC boot.

All sysboot pads are sampled and latched onto the CTRL_CORE_BOOTSTRAP register (in control module) after POR. After booting, these pads can be used for other functions such as GPIOs, and the associated register bit field is not updated by the new functionality. For more information about pad multiplexing configuration, see Section 18.4.6.1.1, *Pad Configuration Registers*, in Chapter 18, *Control Module*.

Note

If used as GPIOs, the sysboot[15:0] pads must be used only in output mode to ensure that the input values always match a certain hardware predefined boot pattern, interpreted after each POR.

32.2.4.1 GPMC Configuration for XIP/NAND

Table 32-6 describes the GPMC interface configuration used in XIP, fast XIP and NAND modes controlled by sysboot[13:10].

Table 32-6. GPMC for XIP/NAND Configuration

sysboot[13]	Bus Width
0b0	8-bit
0b1	16-bit
sysboot[12:11]	**A/D-muxed/non-muxed Device on CS0**
0b00	Non-muxed device
0b01	A/D-muxed device
sysboot[10]	**Wait-pin Monitoring for Read Accesses**

Table 32-6. GPMC for XIP/NAND Configuration (continued)

0b0	Disabled
0b1	Enabled

32.2.4.2 System Clock Speed Selection

There are three crystal speeds available to be selected by means of sysboot[9:8], as described in Table 32-7. User is responsible to set the correct value depending on the actual clock supplied to the device.

Table 32-7. System Clock (SYS_CLK1) Speed Selection

sysboot[9:8]	SYS_CLK1 Speed
0b00	Reserved
0b01	20 MHz
0b10	27 MHz
0b11	19.2 MHz

32.2.4.3 QSPI Redundant SBL Images Offset

Four options are available to set the offset between the redundant SBL images as described in Table 32-8. If not using the redundant SBL feature, there is no change required to sysboot pins as only the primary image is used and the sector offset is a don't care.

Table 32-8. Offset Between Redundant Images

sysboot[7:6]	Offset
0b00	64 KiB
0b01	128 KiB
0b10	256 KiB
0b11	512 KiB

32.2.4.4 Booting Device Order Selection

The ROM code creates the device list (order) based on information gathered from these locations:

- The first location is the sysboot[5:0] external configuration pins sensed in the device CTRL_CORE_BOOTSTRAP register. The sysboot[5:0] configuration pads have two main purposes: configure ROM code software in terms of interfaces and devices used for booting and configuring hardware after a POR or cold reset.

The SYSBOOT pins are used to index a booting device list from a table with possible booting scenarios. The order of examined booting devices is from the first to the third devices.

The following names are used in the tables:

- Memory types:
  - Execute in place (XIP): NOR (CFI) flash memory or other XIP device
  - NAND: NAND flash memories (non-XIP)
  - SD: Removable SD card device
  - eMMC: eMMC™ memory device
  - QSPI_1: 1-bit SPI flash memories
  - QSPI_4: 4-bit (Quad) SPI flash memories
  - SATA: SATA-compatible devices such as solid state drives (SSDs) or hard-disk drives (HDDs)
- Peripheral interfaces:
  - USB: HS USB 2.0 interface
  - UART: UART interface
- Table 32-9 lists the permanent booting devices in bold typeface.

The above boot modes are described in detail in Section 32.3.5, *Peripheral Booting* and in Section 32.3.7, *Memory Booting*.

32.2.4.5

Note

After a warm reset, the ROM code builds a device list featuring only the permanent booting devices (in **bold**)

Table 32-9 lists the booting device order selected by ROM code depending on sysboot[5:0] pins.

Table 32-9. Booting Devices Order

sysboot[5:4]	sysboot[3:0]	First Device	Second Device	Third Device
Peripheral Preferred Booting				
0b00	0b0000	USB	**eMMC**	
0b00	0b0001	USB	**NAND**	
0b00	0b0010	USB	**SD**	**eMMC**
0b00	0b0011	USB	**SATA**	**SD**
0b00	0b0100	USB	UART	**XIP**
0b00	0b0101	SD	**XIP**	
0b00	0b0110	SD	**QSPI_1**	
0b00	0b0111	SD	**QSPI_4**	
0b00	0b1010	SD	**Fast XIP**	
Automotive Recovery/Upgrade or Development Booting (1)				
0b01	0b0000	**USB**		
0b01	0b0011	**UART**		
0b01	0b0100	**SD**	USB	
0b01	0b0101	**SD**	USB	
0b01	0b0110	**SD**	USB	
0b01	0b0111	**SD**	USB	
0b01	0b1000	**SD**	USB	
0b01	0b1001	**SD**	USB	
0b01	0b1010	**SD**	USB	
0b01	0b1011	**SD**	USB	
Memory Preferred Booting				
0b10	0b0000	**eMMC**	USB	
0b10	0b0001	**NAND**	USB	
0b10	0b0010	**SD**	**eMMC**	USB
0b10	0b0011	**SATA**	**SD**	USB
0b10	0b0100	**XIP**	USB	UART
0b10	0b0101	**XIP**	SD	USB
0b10	0b0110	**QSPI_1**	SD	USB
0b10	0b0111	**QSPI_4**	SD	USB
Automotive Production Booting (1)				
0b11	0b0000	**SD**		
0b11	0b0100	**SATA**		
0b11	0b0101	**XIP**		
0b11	0b0110	**QSPI_1**		

Table 32-9. Booting Devices Order (continued)

sysboot[5:4]	sysboot[3:0]	First Device	Second Device	Third Device
0b11	0b0111	**QSPI_4**		
0b11	0b1000	**eMMC**		
0b11	0b1001	**NAND**		
0b11	0b1010	**Fast XIP**		
0b11	0b1011	**eMMC (boot part.)** (2)		

(1) After 10 failed loops through the device list, ROM code initiates immediate global warm reset.
(2) The ROM code boots from eMMC boot partition. If the ROM code fails to retrieve the booting image, it does not try boot from the user area contrary to other eMMC options.

32.2.4.6 Boot Peripheral Pin Multiplexing

Table 32-10 lists the code pin multiplexing configuration done by ROM code according to selected boot peripheral. These settings are not restored to reset default values at ROM Code exit.

Note

The ROM code examines the interfaces that are selected to be searched until a valid bootable interface or device is found. The activities on the pads of the searched interfaces must be considered if they are connected to any other peripherals for any other purposes (for example, an LED connected to a GPMC pad muxed internally to a GPIO).

Table 32-10. Pin Multiplexing According to Boot Peripheral

Boot Device	Boot Interface	Device Pins	In MuxMode	Interface Signals
eMMC	MMC2	gpmc_a[19:27], gpmc_cs[1]	MuxMode=0x1	mmc2_dat[4:7], mmc2_clk, mmc2_dat[0:3], mmc2_cmd
SD	MMC1	mmc1_clk, mmc1_cmd, mmc1_dat[0:3]	MuxMode=0x0	mmc1_clk, mmc1_cmd, mmc1_dat[0:3]
NAND	GPMC	GPMC on CS0	MuxMode=0x0	GPMC on CS0
XIP	GPMC	GPMC on CS0(1)	MuxMode=0x0	GPMC on CS0, wait signal monitoring according to the SYSBOOT[10] setting
SATA	SATA	sata1_txp0, sata1_txn0, sata1_rxp0, sata1_rxn0	N/A	sata1_txp0, sata1_txn0, sata1_rxp0, sata1_rxn0
QSPI_1/QSPI_4	QSPI1	gpmc_a[13:18], gpmc_cs[2]	MuxMode=0x1	qspi1_rtclk, qspi1_d[3:0], qspi1_sclk, qspi1_cs[0]
USB	USB1	usb1_dp and usb1_dm	N/A	usb1_dp and usb1_dm
UART	UART3	uart2_rtsn uart2_ctsn	MuxMode=0x1 MuxMode=0x2	uart3_txd uart3_rxd

(1) ROM code does not use (mux) address lines above A18. These, however, are configured to pullups or pulldowns by hardware. See Table 32-5.

32.3 Device Initialization by ROM Code

This section describes high-level booting concepts and provides basic knowledge for booting on the device.

32.3.1 Booting Overview

32.3.1.1 Booting Types

Booting is the process of starting a bootstrap from one of the booting devices.

The ROM code has two functions for booting: Peripheral booting and memory booting.

- In peripheral booting, the ROM code polls a selected communication interface such as UART or USB, downloads the executable code over the interface, and executes it in internal RAM. Downloaded software from an external host can be used to program flash memories connected to the device. This special case of peripheral booting is called preflashing; software downloaded for preflashing is called the flash loader. The flash loader burns a new client application image in external flash memory. Initial software is a generic term for bootstrap, downloaded software, and flash loader. A software (warm) reset can be performed after the image is burned.
- In memory booting, the ROM code finds the bootstrap in permanent memories such as flash memory, memory cards, or SATA SSD or HDD memory devices and executes it. This process is normally performed after a cold or warm device reset.

The ROM code detects whether the device should download software from a peripheral interface (USB or UART) by using the sysboot pad configuration. This mechanism encompasses initial flashing in production (external memory is empty) and reflashing in service (external memory is already programmed).

32.3.1.2 ROM Code Architecture

Figure 32-4 shows the ROM code architecture. It is split into three main layers with a top-down approach: high-level, drivers, and hardware abstraction layer (HAL). One layer communicates with a lower-level layer through a unified interface.

- The high-level layer performs the main tasks of the public ROM code: multicore startup, watchdog and clock configurations, interrupt management, and main booting routine.
- The driver layer implements the logical and communication protocols for any booting device in accordance with the interface specification.
- The HAL implements the lowest level code for interacting with the hardware infrastructure modules. End booting devices (typically external flash components) are attached to the device I/O pads.

Figure 32-4 shows the three layers with their modules.

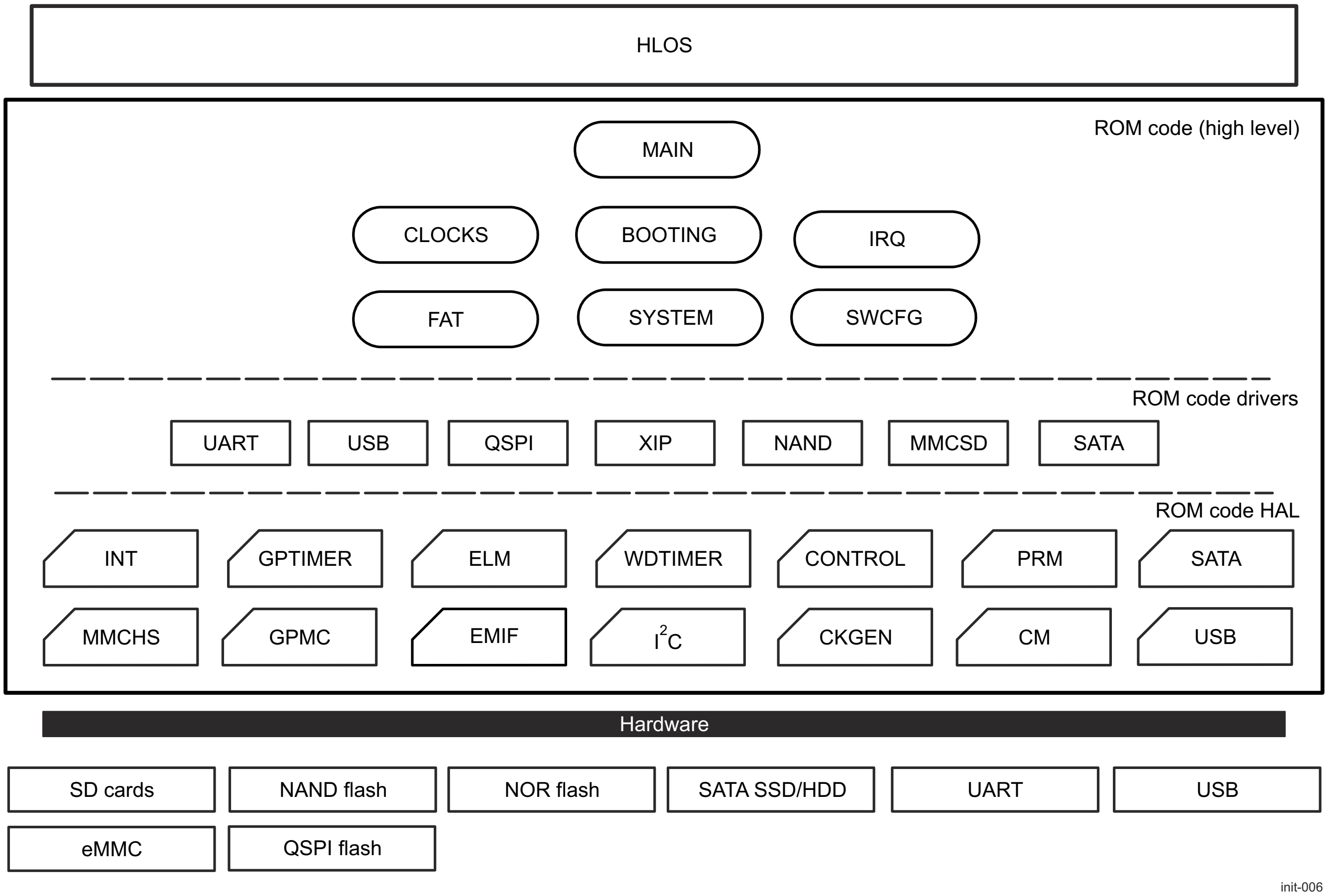


Figure 32-4. ROM Code Architecture

32.3.2 Memory Maps

32.3.2.1 ROM Memory Map

Figure 32-5 shows the 48-KiB ROM memory map.

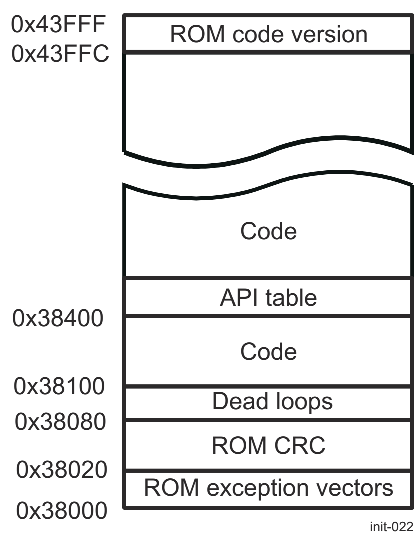


Figure 32-5. ROM Memory Map

- ROM exception vectors

Exceptions are redirected to ROM exception vectors (see Table 32-11). The reset exception is redirected to the public ROM code startup. Other exceptions are redirected to RAM handlers by loading appropriate addresses to the PC register.

Table 32-11. ROM Exception Vectors

Address	Exception	Content
0x38000	Reset	Branch to the ROM code startup
0x38004	Undefined	PC = 0x4037 F004
0x38008	Software interrupt (SWI)	PC = 0x4037 F008
0x3800C	Prefetch abort	PC = 0x4037 F00C
0x38010	Data abort	PC = 0x4037 F010
0x38014	Unused	PC = 0x4037 F014
0x38018	IRQ	PC = 0x4037 F018
0x3801C	FIQ	PC = 0x4037 F01C

- ROM code cyclic redundancy check (CRC)

  The ROM code CRC is calculated as 32-bit CRC code (CRC-32-IEEE 802.3) for the address range 0x38000–0x43FFF. The 4-byte CRC code is stored at location 0x38020.
- Dead loops

  Dead loops are branch instructions coded in Arm mode. They have multiple purposes (see Table 32-12).

Table 32-12. Dead Loops

Address	Purpose
0x38080	Undefined exception default handler
0x38084	SWI exception default handler
0x38088	Prefetch abort exception default handler
0x3808C	Data abort exception default handler
0x38090	Unused exception default handler
0x38094	IRQ exception default handler
0x38098	FIQ exception default handler

- Code

  This space is used to hold code and constant data.
- API table

  The purpose of this table is to allow external code access to system maintenance, utility, and device driver functions which are used for ensuring the ROM code boot functionality. These functions can be reused at run time by calling a fixed address hardcoded in this table.
- ROM code version

  The ROM code version consists of two BCD numbers: major and minor. It can be used to identify the ROM code release version burned in a given IC. The ROM code version is a 32-bit hexadecimal value at address 0x43FFC.

  The ROM code version numbers are:
  – 0x2601 for SR1.0
  – 0x2602 for SR1.1 and SR2.0

32.3.2.2 RAM Memory Map

The partitioning of the on-chip SRAM (L3 OCM RAM) shown in Figure 32-6 is used during the booting process. Tracing areas can also be accessed when calling API functions.

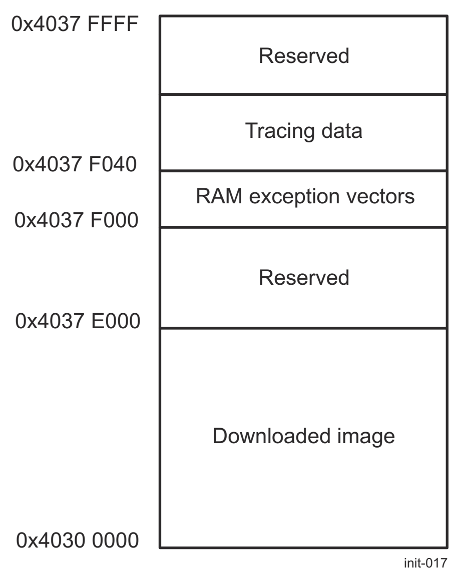


Figure 32-6. RAM Memory Map

- Downloaded image

  This space is used by the public ROM code to store a downloaded booting image. It can be up to 504 KiB.

- RAM exception vectors

  The RAM exception vectors provide an easy way to redirect exceptions to the custom handler. Table 32-13 lists the contents of the RAM space reserved for RAM vectors. The first eight addresses are Arm instructions that load the value in the subsequent eight addresses into the PC. These instructions are executed when an exception occurs because they are called from ROM exception vectors. Undefined, SWI, unused, and FIQ exceptions are redirected to a hardcoded dead loop. Prefetch abort, data abort, and IRQ exception are redirected to predefined ROM handlers. Users can redirect an exception to another handler by writing its address to the appropriate location from 0x4037 F024 to 0x4037 F03C, or by overriding the branch (load into PC) instruction between addresses from 0x4037 F004 to 0x4037 F01C.

Table 32-13. RAM Exception Vectors

Address	Exception	Content
0x4037 F000	Reserved	Reserved
0x4037 F004	Undefined	PC = [0x4037 F024]
0x4037 F008	SWI	PC = [0x4037 F028]
0x4037 F00C	Prefetch abort	PC = [0x4037 F02C]
0x4037 F010	Data abort	PC = [0x4037 F030]
0x4037 F014	Unused	PC = [0x4037 F034]
0x4037 F018	Interrupt request (IRQ)	PC = [0x4037 F038]
0x4037 F01C	Fast interrupt request (FIQ)	PC = [0x4037 F03C]
0x4037 F020	Reserved	0x38090
0x4037 F024	Undefined	0x38080
0x4037 F028	SWI	0x38084
0x4037 F02C	Prefetch abort	Address of default prefetch abort handler(1)
0x4037 F030	Data abort	Address of default data abort handler (1)
0x4037 F034	Unused	0x38090
0x4037 F038	IRQ	Address of default IRQ handler
0x4037 F03C	FIQ	0x38098

(1) The default handlers for prefetch and data abort perform reads from CP15 debug registers to retrieve the reason for the abort:
- In case of prefetch abort: the IFAR register is read from CP15 and stored into R0. The IFSR register is read and stored into the R1 register. Then the ROM code jumps to the prefetch abort dead loop (38088h).
- In case of data abort: the DFAR register is read from CP15 and stored into R0. The DFSR register is read and stored into the R1 register. Then the ROM code jumps to the data abort dead loop (3808Ch).

- Tracing data

  This area contains trace vectors reflecting the execution path of the ROM code. Table 32-14 describes the public ROM code tracing data. For more information about ROM code tracing, see Section 32.3.9, *Tracing*.

Table 32-14. Tracing Data

Address	Size	Description
0x4037 F040	32 bits	Current tracing vector, word 1
0x4037 F044	32 bits	Current tracing vector, word 2
0x4037 F048	32 bits	Current tracing vector, word 3
0x4037 F04C	32 bits	Current tracing vector, word 4

Table 32-14. Tracing Data (continued)

Address	Size	Description
0x4037 F050	32 bits	Cold reset run tracing vector, word 1
0x4037 F054	32 bits	Cold reset run tracing vector, word 2
0x4037 F058	32 bits	Cold reset run tracing vector, word 3
0x4037 F05C	32 bits	Cold reset run tracing vector, word 4
0x4037 F060	32 bits	Current copy of the PRM_RSTST register (reset reasons)

32.3.3 Overall Booting Sequence

Figure 32-7 shows the ROM code flow chart.

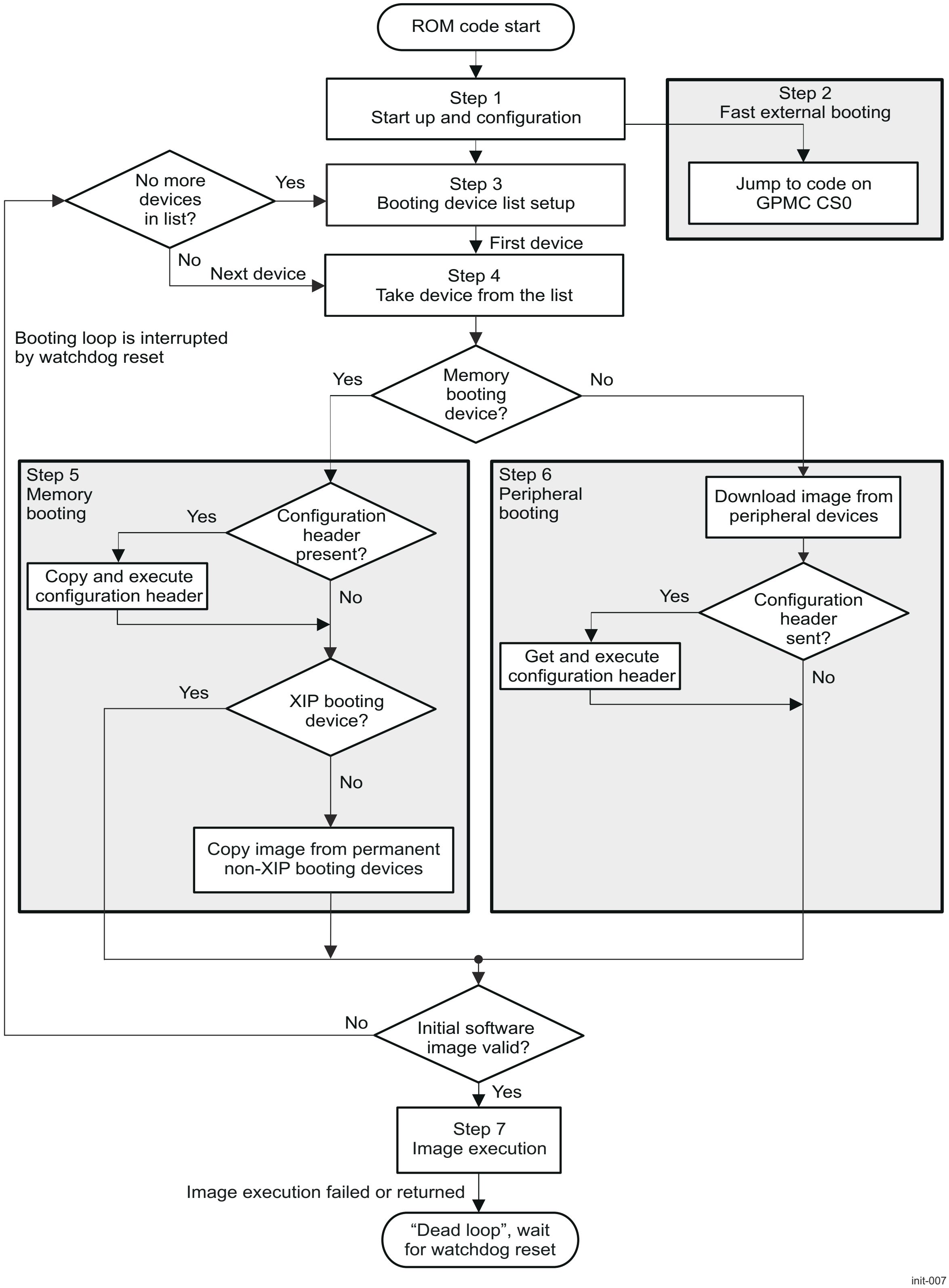


Figure 32-7. Overall Booting Sequence

The main loop of the booting module goes through the booting device list and tries to get an image from the currently selected booting device. The ROM code performs the following steps:

1. Basic configuration and initialization. Reading of SYSBOOT pins .
2. The path named fast external boot is a special low-latency boot mode. It consists of a blind jump to an external addressable memory. See Section 32.3.6, *Fast External Booting.*
3. A booting device list is created (see Section 32.3.4.5, *Booting Device List Setup*). The list consists of all devices to be searched for a booting image. The list is created based on the SYSBOOT pins.
4. The main loop of the booting procedure goes through the booting device list and tries to search for an image from the currently selected booting device. This loop is exited if a valid booting image is found and successfully executed or when the watchdog expires. If an image is found, ROM code executes memory booting or peripheral booting, depending on the type of the current booting device:
   - Memory booting is executed when the booting device is XIP memory, NAND, QSPI, eMMC or SD, SATA SSD/HDD.
   - Peripheral booting is executed when the booting device is UART or USB.
5. Memory booting reads data from memory-type devices. Memory booting is described in detail in Section 32.3.7, *Memory Booting*.
6. Peripheral booting downloads data from communication interfaces. Peripheral booting is described in Section 32.3.5, *Peripheral Booting*.
7. The image automatically starts.

An additional feature of the booting module is the execution of the Configuration Header (CH). The CH configures the system for faster and more flexible booting from the selected permanent or peripheral booting device. The CH, which is optional, is described in *Section 32.3.8.2, Configuration Header*.

32.3.4 Startup and Configuration

32.3.4.1 Startup

Figure 32-8 shows the ROM code start-up sequence.

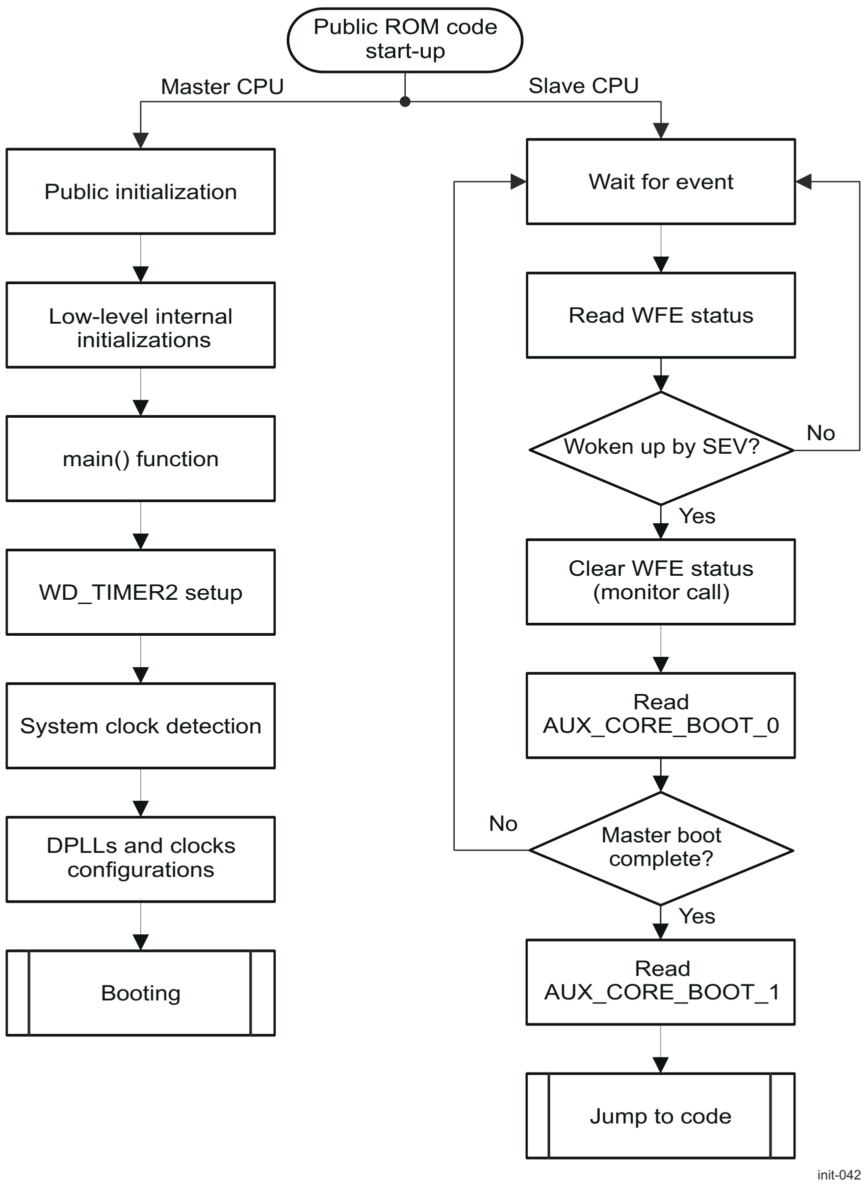


Figure 32-8. ROM Code Multiprocessor Start-Up Sequence

The master CPU L1 instruction cache and branch prediction mechanisms are activated as part of the public boot process. The base address of the public vector is configured to the reset vector of ROM code (0x38000). The memory management unit (MMU) remains switched off during boot (thus, L1 data cache is off). The master CPU performs the basic initialization of the public side. Next, the MMU configures WD_TIMER2 (set to 3 minutes), detects system clock, and configures the system clock. Finally, the MMU jumps to the booting routine.

No specific configuration is performed for the slave CPU, which keeps its default configuration after reset (L1 instruction and data caches off, branch prediction off, MMU off, no remap of the base address of the public vector). The slave CPU is rapidly held in wait-for-event (WFE) state. It stays in this state while the master CPU completes the public boot process and until jumping to the external software (for example, HLOS). At this stage, the external software can wake up the slave CPU by executing an SEV command.

Two internal memory-mapped registers are available to the OS for communicating start-up information. The AUX_CORE_BOOT_0 and AUX_CORE_BOOT_1 registers are in the MPU WakeupGen domain.

- AUX_CORE_BOOT_0 is used as a status register to signal the slave CPU that it must wake up after the send event operation initiated by the master CPU.
- AUX_CORE_BOOT_1 contains the physical address location to which the slave CPU must jump after wakeup.

See the memory mapping of these registers in Section 4.4.10, *MPU_WUGEN Registers* in *Dual Cortex-A15 MPU Subsystem*.

32.3.4.2 Control Module Configuration

Table 32-15 lists the Control Module registers modified at each ROM code startup. It is a Control Module requirement to be met prior to modify any of the pad control registers. These registers are not reverted back to default values (that is, to LOCK state) when ROM code completes.

Table 32-15. Control Module Registers Modified by ROM Code at Each Startup

Register	Value	Meaning
CTRL_CORE_MMR_LOCK_1	0x2FF1AC2B	Unlock Control Module registers starting at address offset 0x0000 0100 and ending at 0x0000 079F
CTRL_CORE_MMR_LOCK_2	0xF757FDC0	Unlock Control Module registers starting at address offset 0x0000 07A0 and ending at 0x0000 0D9F
CTRL_CORE_MMR_LOCK_3	0xE2BC3A6D	Unlock Control Module registers starting at address offset 0x0000 0DA0 and ending at 0x0000 0FFF
CTRL_CORE_MMR_LOCK_4	0x1EBF131D	Unlock Control Module registers starting at address offset 0x0000 1000 and ending at 0x0000 13FF
CTRL_CORE_MMR_LOCK_5	0x6F361E05	Unlock Control Module registers starting at address offset 0x0000 1400 and ending at 0x0000 1FFF

Once the booting device list is completed, the ROM code applies the pin multiplexing settings as described in Section 32.2.4.6, *Pin Multiplexing According to Boot Peripheral*.

32.3.4.3 PRCM Module Mode Configuration

Table 32-16 lists the PRCM module mode, clock control, and power control registers modified at each ROM code startup. These registers are not reverted back to default values when ROM code completes.

Table 32-16. PRCM Module Mode Registers Modified by ROM Code

Register	Field	Value
CM_L3INSTR_L3_INSTR_CLKCTRL	MODULEMODE	DISABLED
CM_L3INSTR_OCP_WP_NOC_CLKCTRL	MODULEMODE	DISABLED
CM_CM_CORE_PROFILING_CLKCTRL	MODULEMODE	DISABLED
CM_PRM_PROFILING_CLKCTRL	MODULEMODE	DISABLED
CM_CM_CORE_AON_PROFILING_CLKCTRL	MODULEMODE	DISABLED
CM_EMU_CLKSTCTRL	CLKTRCTRL	HW_AUTO
CM_DSP1_CLKSTCTRL	CLKTRCTRL	HW_AUTO
CM_DSP1_DSP1_CLKCTRL	MODULEMODE	DISABLED
PM_DSP1_PWRSTCTRL	POWERSTATE	OFF
CM_DSP2_CLKSTCTRL	CLKTRCTRL	HW_AUTO
CM_DSP2_DSP2_CLKCTRL	MODULEMODE	DISABLED
PM_DSP2_PWRSTCTRL	POWERSTATE	OFF
CM_IVA_CLKSTCTRL	CLKTRCTRL	HW_AUTO
PM_IVA_PWRSTCTRL	POWERSTATE	OFF

32.3.4.4 Clocking Configuration

The ROM code detects the system input clock frequency (SYS_CLK1) from the sysboot[9:8] pins value. The supported system frequencies in the device are:

- 19.2 MHz
- 20 MHz
- 27 MHz

See Section 32.2.3.2, *Clocking scheme*, and Section 32.2.4.2, *System clock speed configuration*.

After detecting the input clock, the ROM code configures the clocks and DPLLs required for ROM code execution.

The configured DPLLs are:

- DPLL_PER: locked to provide clocks to peripheral blocks
- DPLL_CORE: locked to provide L3_MAIN interconnect, L4 interconnect, and EMIF clocks
- DPLL_MPU: locked
- DPLL_USB_OTG_SS/DPLL_USB DPLLs: Locked only in case of USB peripheral booting. It is left untouched otherwise.
- DPLL_SATA: locked only in the case of the SATA memory booting. It is left untouched otherwise.

The DPLLs and PRCM clock dividers are configured with the default values of the ROM code (depending on the detected system input clock) after cold or warm reset in order to give the same working conditions to the ROM code sequence.

Table 32-17 summarizes the default ROM code clock settings.

Table 32-17. ROM Code Default Clock Settings

Clock	Frequency (MHz)	Source
DPLL_CORE clock with F_{DPLL} locked frequency	2128	Gated SYS_CLK1
EMIF_PHY_GCLK[2]	44.33	DPLL_DDR (M2)
EMIF_DLL_GCLK	266	DPLL_DDR .HSDIVIDER (H11)
CORE_X2_CLK	266	DPLL_CORE.HSDIVIDER (H12)
CORE_CLK	266	CORE_X2_CLK
CORE_USB_OTG_SS_LFPS_TX_CLK	34.3	DPLL_CORE.HSDIVIDER (H13)
CORE_GPU_CLK	212.8	DPLL_CORE.HSDIVIDER (H14)
CORE_IPU_ISS_BOOST_CLK	212.8	DPLL_CORE.HSDIVIDER (H22)
CORE_ISS_MAIN_CLK	152	DPLL_CORE.HSDIVIDER (H23)
BB2D_GFCLK	177.3	DPLL_CORE.HSDIVIDER(H24)
L3_ICLK	133	CORE_CLK
L4_ICLK	66.5	L3_ICLK
MPU_DPLL_HS_CLK	266	CORE_X2_CLK
IVA_DPLL_HS_CLK	266	CORE_X2_CLK
DPLL_PER – clock with Fdpll locked frequency	768	Gated SYS_CLK1
FUNC_192M_CLK	192	DPLL_PER (M2)
FUNC_256M_CLK	256	DPLL_PER.HSDIVIDER (H11)
DSS_GFCLK	192	DPLL_PER.HSDIVIDER (H12)
PER_QSPI_CLK	192	DPLL_PER.HSDIVIDER(H13)
PER_GPU_CLK	192	DPLL_PER.HSDIVIDER (H14)
DPLL_MPU - clock with Fdpll locked frequency	2200	Gated SYS_CLK1
MPU_DPLL_CLK	588	PRM

Table 32-17. ROM Code Default Clock Settings (continued)

Clock	Frequency (MHz)	Source
MPU_GCLK	588	DPLL_MPU (M2)
DPLL_USB - clock with Fdpll locked frequency (1)	960	Gated SYS_CLK1
L3INIT_480M_GFCLK	480	DPLL_USB (M2)
L3INIT_60M_GFCLK	60	L3INIT_480M_GFCLK
DPLL_USB_OTG_SS - clock with F_{DPLL} locked frequency (1)	2500	Gated SYS_CLK1
DPLL_SATA - clock with Fdpll locked frequency (1)	1500	Gated SYS_CLK1

(1) This clock is locked specifically if USB peripheral or SATA memory booting is selected.
(2) This clock is intentionally set up at low frequency to ensure correct initialization of external DRAM components.

However it is possible to override the default clock settings. There are three ways to change DPLLs and all related clock divider, gating, and multiplexer configurations during the boot:

- ROM code default settings, described in Table 32-17. They are always applied at any reset.
- The CH, described in Section 32.3.8.2, *Configuration Header*. The CH lets users have a known configuration (about GPMC and clock registers) after memory or peripheral booting. .

32.3.4.5 Booting Device List Setup

The ROM code creates a device list based on these sources:

- The sysboot[5:0] signals latched in the control module are used to index the device table from which the list of devices is extracted.

Note

Only permanent booting devices are put to the list when the reset is not power-on and the devices are taken from the sysboot pins.

32.3.5 Peripheral Booting

32.3.5.1 Description

The ROM code can boot from these peripherals:

- USB1: High-, and Full-speed USB from USB1 internal transceivers
- UART3: 115.2 Kbps, 8 bits, even parity, 1 stop-bit, no flow control

The purpose of booting from a peripheral is to download a flash loader code from an external host. This booting method is used primarily for programming flash memories connected to the device (for example, in the case of initial flashing, firmware update or servicing). Figure 32-11 shows the overall peripheral booting procedure. It consists of a synchronization phase (handshake between the host and the device) and a transfer phase. The synchronization phase is similar for UART and USB boots. Both transfer phases use the same procedure.

When booting from the UART, the ROM code first initializes the UART3 interface. Then the ROM code sends an ASIC ID block of data. From there, it expects to receive a boot message from the host within 300 ms, by default. Figure 32-9 shows this procedure.

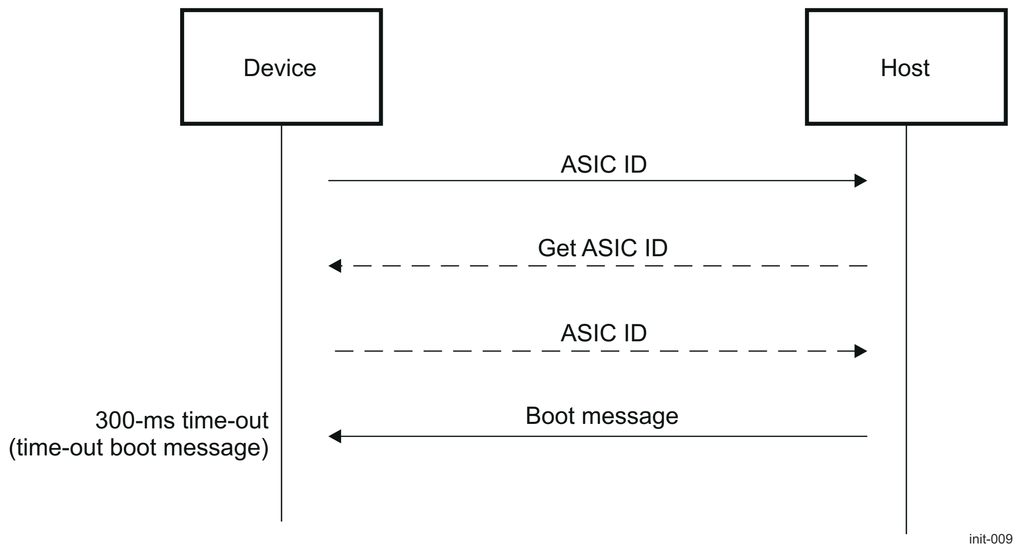


Figure 32-9. Synchronization Phase for UART

Figure 32-10 shows the procedure when booting from the USB.

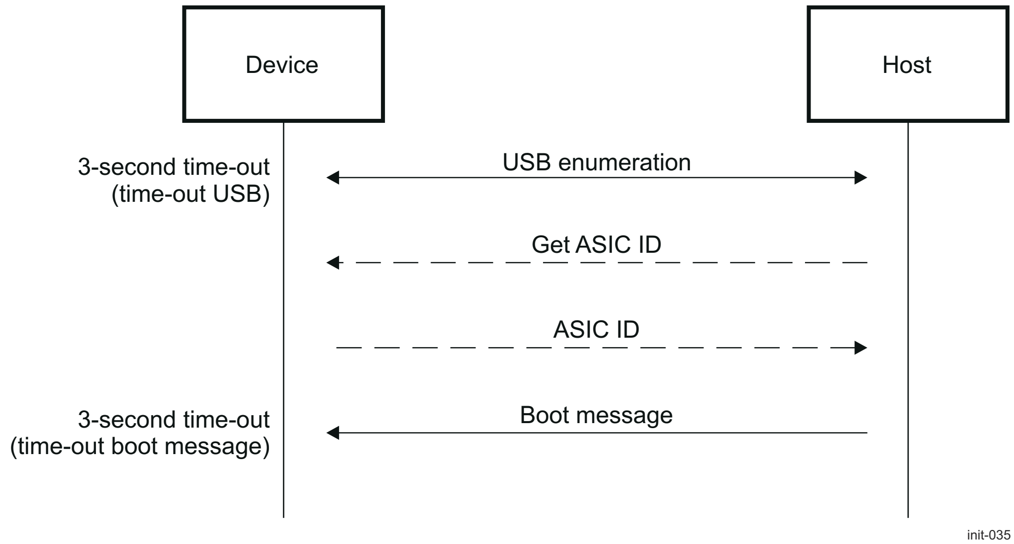


Figure 32-10. Synchronization Phase for USB

During the synchronization phase (see Figure 32-9 and Figure 32-10), the device can provide a small packet of data called the ASIC ID (described in Table 32-18). It is a simple structure that contains different kinds of information, such as ROM version, checksums, and ID.

The host can decide the desired operation by providing a booting message (see Table 32-22). This message can be: Get ASIC ID, peripheral boot, change device, or next device. If the device receives the Get ASIC ID boot message, it sends back the ASIC ID contents.

If the change device or next device message is received, the ROM code stops the current peripheral booting procedure and returns to the main booting, which decides about the next booting device according to the boot message received.

If the peripheral boot message is received without a time-out, the device is entering the transfer phase. From there, the flash loader image size (as a 32-bit word) and the flash loader image itself are expected to be received. The ROM code waits up to 1 minute for completion of image size reception, and up to 1 more minute to download the image. If the download procedure does not complete before this time, the peripheral booting procedure aborts. ROM code continues to examine the devices included in the booting device list. If the download procedure passes, then the image can be executed.

The flash loader image is downloaded directly into internal RAM from address 0x4030 0000 and the maximum size of the downloaded image is 504 KiB.

Note

Sending an image size of zero skips the peripheral booting procedure.

The USB or UART connection is left open at the end of the transfer phase and once exiting the ROM code for the initial software to take over. It means the initial software can reuse the currently established connection. In the case of a USB connection, the endpoints can be reused as such, without closing the connection and performing a full enumeration again.

Table 32-18. ASIC ID Structure

ASIC ID Item	Size (Bytes)	Description
Items	1	Number of subblocks
ID subblock	7	Device identification information
Reserved subblock	4	Reserved
Reserved subblock	23	Reserved

Table 32-18. ASIC ID Structure (continued)

ASIC ID Item	Size (Bytes)	Description
Reserved subblock	35	Reserved
Checksum subblock	11	CRC (4 bytes)

Table 32-19. Items

Offset	Size (Bytes)	Description
0x00	1	0x05: Number of subblocks USB 0x04: Number of subblocks UART[1]

Table 32-20. ID Subblock

Offset	Size (Bytes)	Description
0x01	1	0x01: Subblock ID
0x02	1	0x05: Subblock size
0x03	1	0x01: Fixed value
0x04	2	0x56, 0x41: Device ID number
0x06	1	0x07: CH enabled 0x17: CH disabled
0x07	1	ROM code revision 0x01: SR1.0 0x02: SR1.1 and SR2.0

Table 32-21. Checksum Subblock

Offset[1]	Size (Bytes)	Description
0x46	1	0x15: Subblock ID
0x47	1	0x09: Subblock size
0x48	1	0x01: Fixed value
0x49	4	ROM CRC
0x4D	4	0x0000 0000:

(1) The checksum subblock is not transmitted over the UART.

Table 32-22 summarizes the boot messages sent from a host.

Table 32-22. Booting Messages

Message Name	Value	Description
Peripheral boot	0xF003 0002	Continue peripheral booting.
Get ASIC ID	0xF003 0003	ASIC ID request. The Get ASIC ID request message is optional. If received, the ROM code sends its ASIC ID data to the host in return. The host can issue the Get ASIC ID message multiple times if required. Table 32-18 describes the structure of the ASIC ID.
Change device	0xF003 xx06	Skip current peripheral booting and continue booting from device type indicated by xx: 0x01: XIP 0x02: XIP (with wait monitoring) 0x03: NAND 0x05: SD card 0x06: eMMC (from boot partition BP1 or BP2) 0x07: eMMC 0x09: SATA 0x0A: QSPI_1 0x0B: QSPI_4 0x43: UART 0x45: USB (from internal transceiver) Others: Reserved

Table 32-22. Booting Messages (continued)

Message Name	Value	Description
Next device	0xFFFF FFFF	Skip current device and move to the next device on the device list.
Memory booting	Others	Skip current peripheral booting and move to the first device for memory booting.

Figure 32-11 shows the peripheral booting procedure.

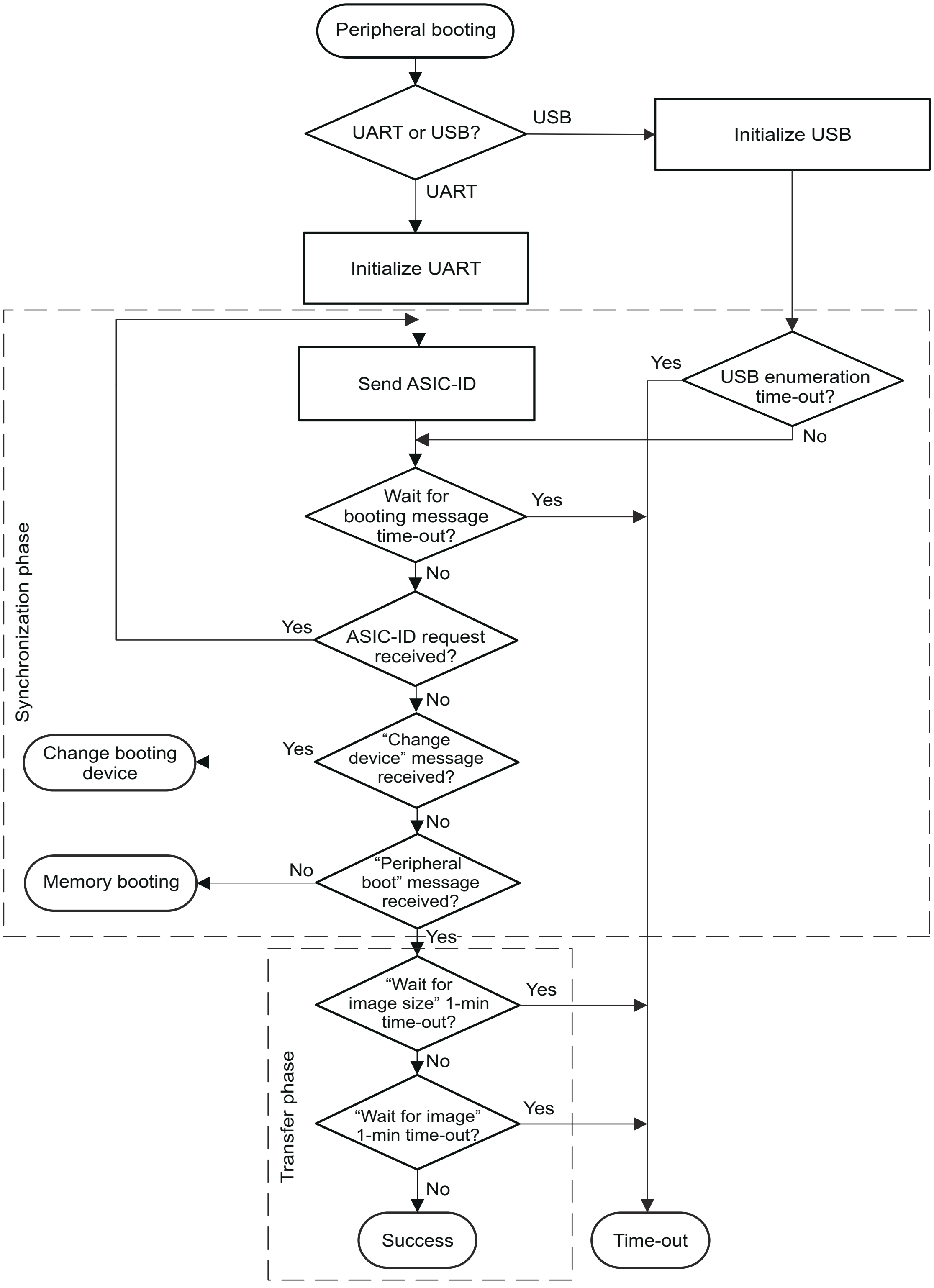


Figure 32-11. Peripheral Booting Procedure

32.3.5.2 Initialization Phase for UART Boot

The ROM code supports booting from a UART interface with the following characteristics:

- UART3 interface
- Communication parameters set to 115.2 Kbps, 8 bits, even parity, 1 stop-bit, no flow control
- Two-pin interface: RX/TX
- The boot message default time-out is 300 ms (time-out boot message)

32.3.5.3 Initialization Phase for USB Boot

The ROM code supports booting from a USB interface with the following characteristics:

- Using the USB1 subsystem in USB2.0 mode
- Device integrated USB transceiver (USB2PHY1)
- Enumeration default time-out is 3 seconds (time-out USB)
- The boot message default time-out is 3 seconds (time-out boot message).

Note

The ROM code does not handle any OTG-specific features.

32.3.5.3.1 Initialization Procedure

Note

The internal USB2PHY1 (HS) transceiver must be powered and configured upon startup. No action is expected from the device ROM code for its configuration.

Figure 32-12 shows the USB initialization procedure.

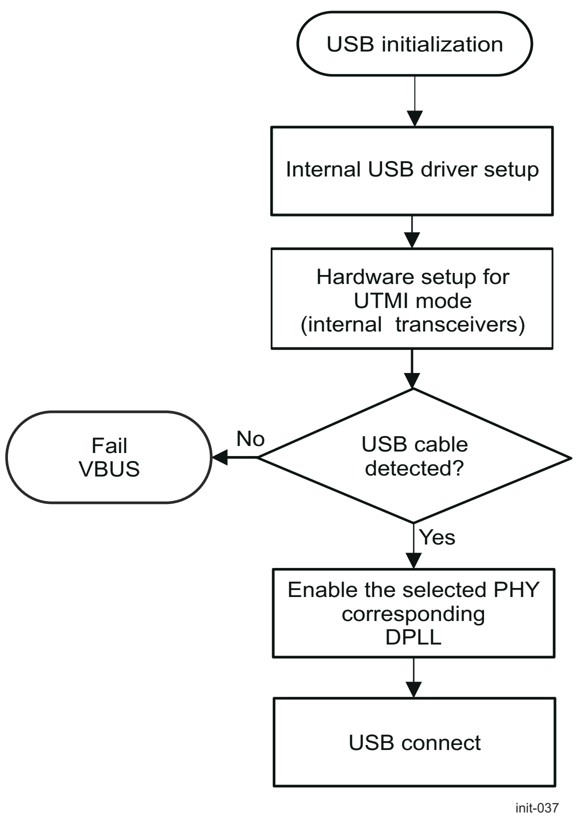


Figure 32-12. USB Initialization Procedure

Note

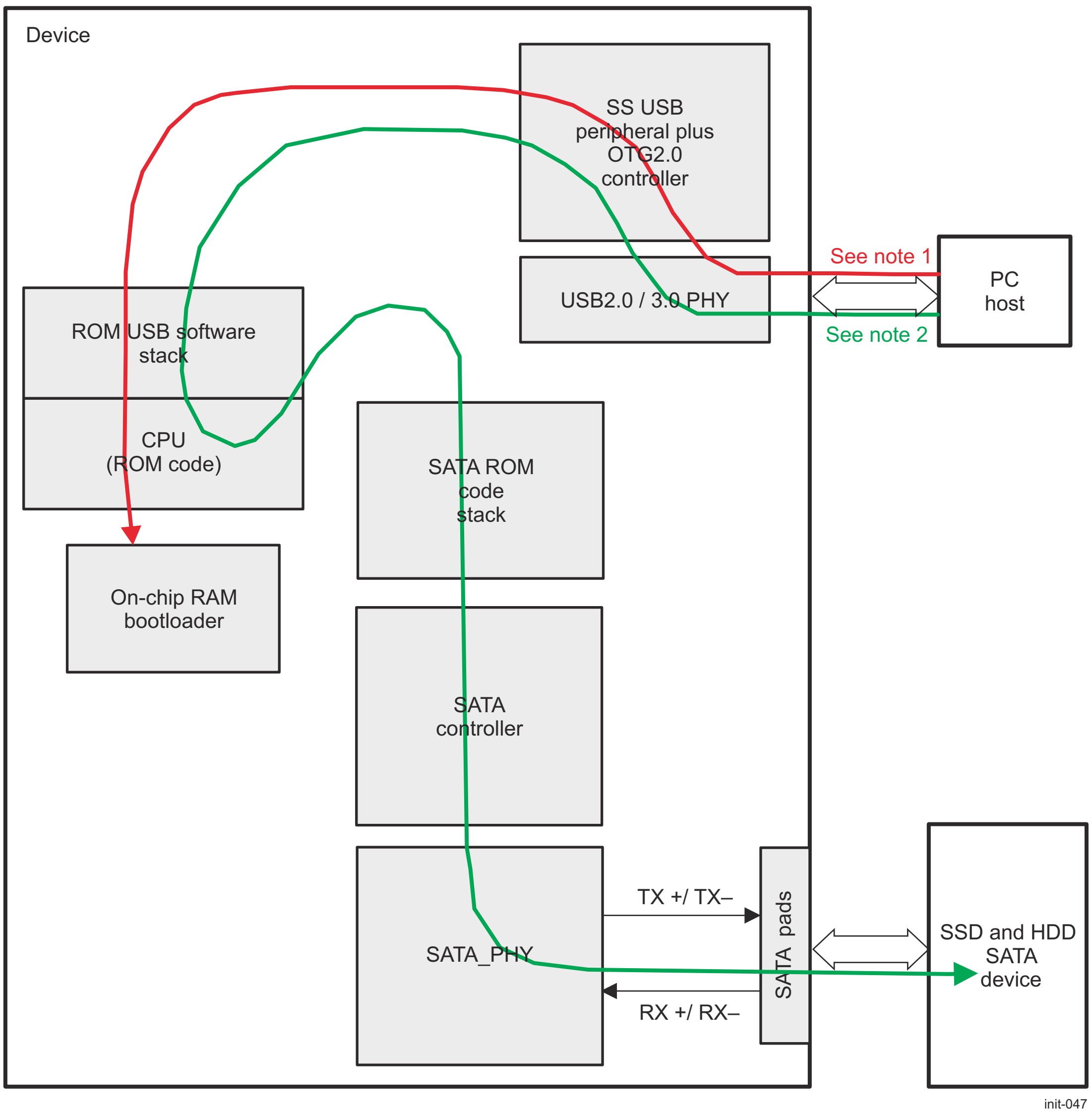
The ROM code does not consider the USB PHYs charger detection circuitry status.

Note

The ROM code assumes that VBUS is always present.

32.3.5.3.2 SATA Peripheral Device Flashing over USB Interface

In the next scheme (see Figure 32-13), it is assumed that the new version of the firmware Flash Loader Host is located in the PC host. In the same way, the application loaded to device internal L3 On-chip RAM (using the peripheral boot function of the device ROM code) is called Flash Loader Second.



A. ROM boots from USB controller and downloads deviceFlash Second in RAM.

B. The PC host is allowed to flash the device firmware located in the SATA device.

Figure 32-13. SATA Flashing Over USB

32.3.5.3.3 USB Driver Descriptors

USB devices use descriptors to report their attributes. A descriptor is a data structure with a defined format. Each descriptor begins with a byte-wide field that contains the total number of bytes in the descriptor followed by a byte-wide field that identifies the descriptor type. Using descriptors allows concise storage of the attributes of individual configurations so that each configuration can reuse descriptors or portions of descriptors from other configurations that have the same characteristics. Where appropriate, descriptors contain references to string descriptors. String descriptors contain displayable, human-readable description information. These descriptor details can be used for tool development or debugging:

- Device descriptor

  A device descriptor contains general information about a USB device, including global information that applies to the device and all device configurations. A USB device has only one device descriptor. A device-qualifier descriptor is required because the ROM code uses the HS feature of the USB core. Table 32-23 lists the device descriptors.

Table 32-23. Device Descriptor

Field	Value	Description
bLength	0x12	Size of this descriptor in bytes
bDescriptorType	0x01	Device descriptor type
bcdUSB	0x0210	USB specification release number in binary coded decimal (BCD) format
bDeviceClass	Vendor-specific (0xFF)	Class code
bDeviceSubClass	Vendor-specific (0xFF)	Subclass code
bDeviceProtocol	Vendor-specific (0xFF)	Protocol code
bMaxPacketSize0	0x40	Maximum packet size for endpoint 0
idVendor	0x0451	Vendor ID (Texas Instruments), TI default value
idProduct	0xD013	Product ID, TI default value
bcdDevice	0x0000	Device release number
iManufacturer	See Section 32.3.5.3.5.	Index of string descriptor describing manufacturer
iProduct	See Section 32.3.5.3.5.	Index of string descriptor describing product
iSerialNumber	See Section 32.3.5.3.5.	Index of string descriptor describing device serial number
bNumConfigurations	0x01	Number of possible configurations

32.3.5.3.4

- Device-qualifier descriptor

  The device-qualifier descriptor contains information about a HS-capable device that changes if the device operates at its other speed. This descriptor is retrieved by the host using the GetDescriptor() request (standard device request). Table 32-24 lists the device-qualifier descriptors.

Table 32-24. Device-Qualifier Descriptor

Field	Value	Description
bLength	0x0A	Size of this descriptor in bytes
bDescriptorType	0x06	Device-qualifier descriptor type
bcdUSB	0x0210	USB specification release number in BCD
bDeviceClass	0xFF	Class code
bDeviceSubClass	0xFF	Subclass code
bDeviceProtocol	0xFF	Protocol code
bMaxPacketSize0	0x40	Maximum packet size for endpoint 0

Table 32-24. Device-Qualifier Descriptor (continued)

Field	Value	Description
bNumConfigurations	0x01	Number of possible configurations
bReserved	0x00	Reserved for future use

- Configuration descriptor

  This descriptor gives information about a specific device configuration. It describes the number of interfaces supported by the configuration (see Table 32-25).

Table 32-25. Configuration Descriptor

Field	Value	Description
bLength	0x09	Size of this descriptor in bytes
bDescriptorType	0x02	Configuration descriptor type
wTotalLength	–	Combined length of all descriptors
bNumInterfaces	0x01	Number of interfaces supported
bConfigurationValue	0x01	Value to use as an argument for the SetConfiguration() request
iConfiguration	Index	Index of string descriptor describing this configuration
bmAttributes	0xC0	Power setting and remote wakeup
bMaxPower	0x32	Maximum power consumption of the USB device

- Other speed configuration descriptor

  This descriptor describes the configuration of a HS-capable device if it operates at its other possible speed (see Table 32-26).

Table 32-26. Other Speed Configuration Descriptor

Field	Value	Description
bLength	0x09	Size of this descriptor in bytes
bDescriptorType	0x07	Other speed configuration descriptor type
wTotalLength	–	Combined length of all descriptors
bNumInterfaces	0x01	Number of interfaces supported
bConfigurationValue	0x01	Value to use as an argument for the SetConfiguration() request
iConfiguration	Index	Index of string descriptor describing this configuration
bmAttributes	0xC0	Power setting and remote wakeup
bMaxPower	0x32	Maximum power consumption of the USB device

- Interface descriptor

  This descriptor describes a specific interface in a configuration (see Table 32-27).

Table 32-27. Interface Descriptor

Field	Value	Description
bLength	0x09	Size of this descriptor in bytes
bDescriptorType	0x04	Interface descriptor type
bInterfaceNumber	0x00	Number of this descriptor
bAlternateSetting	0x00	Value to select the alternate setting

Table 32-27. Interface Descriptor (continued)

Field	Value	Description
bNumEndpoints	0x02	Number of endpoints used for this interface
bInterfaceClass	0xFF	Class code
bInterfaceSubClass	0xFF	Subclass code
bInterfaceProtocol	0xFF	Protocol code
iInterface	Index	Index of string descriptor describing this interface

- Endpoint descriptor

  Each endpoint used for an interface has its own descriptor. This descriptor contains information required by the host to determine the bandwidth requirements of each endpoint. This descriptor is returned as part of the GetDescriptor(Configuration) request (see Table 32-28 and Table 32-29).

Table 32-28. BULK IN Endpoint Descriptor

Field	Value	Description
bLength	0x07	Size of this descriptor in bytes
bDescriptorType	0x05	Endpoint descriptor type
bEndpointAddress	0x81 (1 IN)	Address of the endpoint on the USB device
bmAttributes	0x02 (Bulk)	Type of transfer
wMaxPacketSize	See [(1)].	Number of endpoints used for this interface
bInterval	0x00	Maximum NAK rate

(1) The maximum size is 0x0200 (512 bytes) for HS bulk endpoint and 0x0040 (64 bytes) for FS bulk endpoint.

Table 32-29. BULK OUT Endpoint Descriptor

Field	Value	Description
bLength	0x07	Size of this descriptor in bytes
bDescriptorType	0x05	Endpoint descriptor type
bEndpointAddress	0x01 (1 OUT)	Address of the endpoint on the USB device
bmAttributes	0x02 (Bulk)	Type of transfer
wMaxPacketSize	See [(1)].	Number of endpoints used for this interface
bInterval	0x00	Maximum NAK rate

(1) The maximum size is 0x0200 (512 bytes) for HS bulk endpoint and 0x0040 (64 bytes) for FS bulk endpoint.

- String descriptors

  String descriptors use UNICODE encoding. The strings in a USB device can support multiple languages. When requesting a string descriptor, the requester specifies the desired language using a 16-bit language ID (LANGID) defined by the USB interface. String index 0 for all languages returns a string descriptor that contains an array of 2-byte LANGID codes supported by the device.

  For the description of the different string descriptors, see:
  – The language ID string descriptor (Table 32-30)
  – The manufacturer ID string descriptor (Table 32-31)
  – The product ID string descriptor (Table 32-32)
  – The configuration string descriptor (Table 32-33)
  – The interface string descriptor (Table 32-34)

Table 32-30. Language ID String Descriptor

Field	Value	Description
bLength	0x04	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
wLangId	0x0409 (US English)	Language ID code

Table 32-31. Manufacturer ID String Descriptor

Field	Value	Description
bLength	0x24	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
bString	Texas Instruments	Manufacturer string

Table 32-32. Product ID String Descriptor

Field	Value	Description
bLength	0x12	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
bString	<string>	Product string

Table 32-33. Configuration String Descriptor

Field	Value	Description
bLength	0x08	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
bString	pbc	Configuration string

Table 32-34. Interface String Descriptor

Field	Value	Description
bLength	0x08	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
bString	pbi	Interface string

32.3.5.3.5 USB Customized Vendor and Product IDs

When device ROM Code enumerates through USB, one may want to use his/her own vendor and product IDs rather than TI defaults. Product and Vendor IDs are transmitted as part of the USB Standard Device Descriptor during the USB device enumeration process (resp. idProduct and idVendor 16 bits fields).

Product IDs (PIDs) and vendor IDs (VIDs) are stored as part of device chip eFuses and readable from control module. The built-in USB ROM driver behaves differently whether the VID value is either left unfused, or fused with a customer-specific VID. Thus, two main cases are distinguished:

- VID is fused with a customer-specific ID value
- VID is left unfused

Table 32-35 lists standard USB ROM device descriptors. Table 32-36 lists USB ROM descriptor strings.

Table 32-35. USB ROM Standard Device Descriptor

Device Descriptor Field	Size (Bytes)	VID≠0x0000[1] (fused)	VID=0x0000 (unfused)[2]
bLength	1	0x12	
bDescriptorType	1	0x1	
bcdUSB	2	0x0200 (if enumerated in USB2.0 HS or FS)	

Table 32-35. USB ROM Standard Device Descriptor (continued)

Device Descriptor Field	Size (Bytes)	VID≠0x0000[1] (fused)	VID=0x0000 (unfused)[2]
bDeviceClass	1	0xFF	
bDeviceSubClass	1	0xFF	
bDeviceProtocol	1	0xFF	
bMaxPacketSize0	1	0x40 (64 bytes if enumerated in USB2.0 HS or FS)	
idVendor	2	(VID value from control module)	0x0451
idProduct	2	(PID value from control module)	0xD013
bcdDevice	2	0x0000	
iManufacturer	1	0x20	0x21
iProduct	1	0x24	0x25
iSerialNumber	1	0	
bNumConfigurations	1	1	

(1) VID other than 0x0 and VID other than 0xFFFF and VID other than 0x0451 values
(2) VID=0x0 or VID=0xFFFF or VID=0x0451

Table 32-36. USB ROM Descriptor Strings

String Descriptor	Size (bytes)	VID ≠ 0x0000	VID=0x0000(unfused)[1]
Serial Number		N/A	
Configuration	8	"pbc"	
Interface	8	"pbi"	
Product	8/18	N/A	<string>
Manufacturer	8/36	N/A	"Texas Instruments"

(1) VID=0x0451

The ROM code transmits additional descriptors as part of the enumeration procedure: configuration descriptor, device qualifier, language ID string, configuration string, interface string, and function string descriptors. Those do not depend on the VID or PID value.

32.3.5.3.6 USB Driver Functionality

- Transactions supported:
  - Control transactions: Used for standard device requests
  - Bulk transactions: Used for data transfer in the image downloading stage. The ASIC ID is sent to the BULK IN endpoint and the image is transferred from the host over the BULK OUT endpoint.

  The device USB device first attaches to the host as an FS device. In the reset mechanism, the USB core requests HS operation. If the HS negotiation in the reset phase succeeds, further transactions are at HS; otherwise, they are at FS. After reset, the USB driver checks for the speed of the device, whether it is FS or HS. Depending on the speed configured by the host, the standard USB device requests are answered with the corresponding descriptors.
- Standard device request restrictions:

  Some standard device requests are not supported by the driver because the USB driver is dedicated for use by the ROM code for peripheral booting. Table 32-37 lists the standard device requests supported by the driver.

Table 32-37. Standard Device Requests Supported

Request	Description	Support
CLEAR_FEATURE	Sets or clears a specific feature	Supported only for the ENDPOINT_HALT feature
GET_CONFIGURATION	Returns the current device configuration value	Yes
GET_DESCRIPTOR	Returns the specified descriptor	Yes

Table 32-37. Standard Device Requests Supported (continued)

Request	Description	Support
GET_INTERFACE	Returns the selected alternate setting for the specified interface	Yes
GET_STATUS	Returns the status for the specified recipient	Yes
SET_ADDRESS	Sets the device address	Yes
SET_CONFIGURATION	Sets the device configuration	Yes
SET_DESCRIPTOR	Updates existing descriptors or adds new descriptors	Runtime updating of descriptors is not supported.
SET_FEATURE	Sets or enables a specific feature	Supported only for the ENDPOINT_HALT feature
SET_INTERFACE	Selects an alternate setting in an interface	Runtime setting of alternate features is not supported.
SYNCH_FRAME	Sets and reports an endpoint synchronization frame	No, because isochronous transfers are not used

32.3.6 Fast External Booting

32.3.6.1 Overview

The fast external boot is a special memory booting mode. It consists of a blind jump to a code in an external XIP memory device connected to GPMC CS0 and lets customers create their own booting code. Fast external booting is selected by means of the SYSBOOT[5:0] pins .

The NOR device must fulfill the following requirements for fast XIP mode:

- Non-muxed or address/data multiplexed mode, configured through sysboot[12:11] (=0x1 for A/D mux)
- 8-bit or 16-bit data bus width, configured through sysboot[13] (=1 for 16-bit)
- CS0 chip select
- Device wait signal connected to the WAIT0 GPMC signal (if used)
- The wait monitoring enable/disable is based on the value of SYSBOOT[10]. SYSBOOT[10] value is exported to GPMC by hardware means.

The jump is performed with minimum on-chip ROM code execution.

32.3.6.2 Fast External Booting Procedure

Figure 32-14 shows the procedure for fast external boot. The code does not use any RAM and is designed for fast execution.

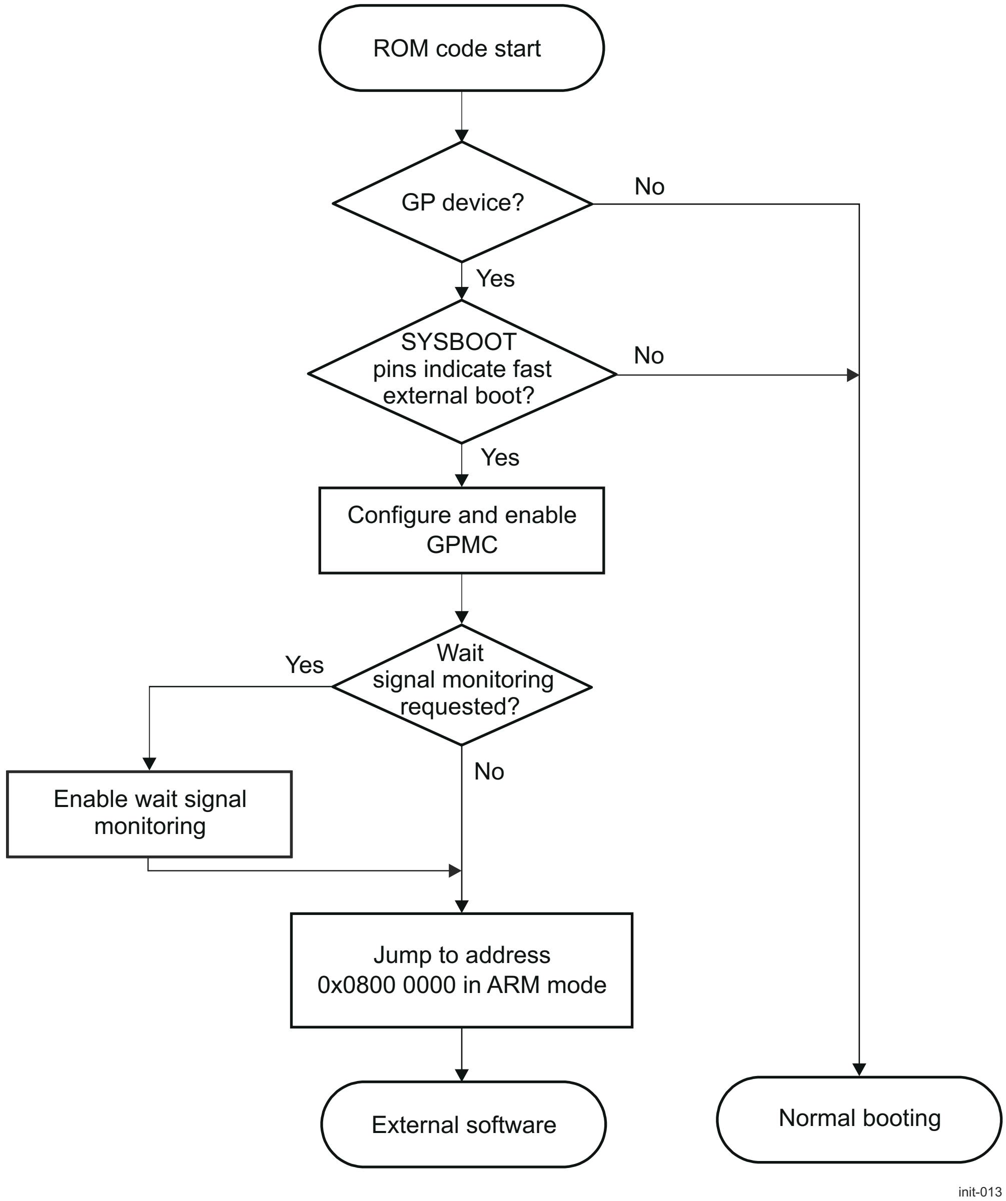


Figure 32-14. Fast External Boot Procedure

32.3.7 Memory Booting

32.3.7.1 Overview

The memory booting process starts an external code in memory devices. ROM code can use only the memory type of booting devices as permanent booting devices (that is, devices examined after both cold [POR] and warm resets). Temporary booting devices are examined only after cold resets. The supported permanent booting devices are:

- NOR flash devices
- NAND flash devices
- SPI/QSPI flash memories
- eMMC memories
- SD cards
- SATA SSD and HDD storage memory devices

Two main groups of permanent booting devices are distinguished by code shadowing. Code shadowing means copying code from a nondirectly addressable device (non-XIP) to RAM, where the code can be executed. Directly addressable devices are XIP devices.

Figure 32-15 shows the general memory booting procedure common to all types of devices. First, CH is copied to internal RAM. It is copied even for XIP devices, because the device can temporarily lose a connection with XIP memory during CH execution (for example, while updating interface timings). The second step is to shadow the image, if the device is not XIP. The last step is image execution and any return from image results in a dead loop.

If CH copying or shadowing fails, memory booting returns to the main booting procedure, which selects the next device for booting.

Note

A booting image is considered to be present , when the first 4-byte word of the sector is not equal to 0000 0000h or FFFF FFFFh.

During the first read sector (512 bytes) call, sectors are copied to a temporary device on-chip SRAM buffer. Once the image is found and the destination address is known, the content of the temporary buffer is moved to the target device on-chip SRAM location so it is required to reread the first image sector. GP header is not copied into target buffer location; therefore, only executable code is in device on-chip RAM, with the first executable instruction at the destination address.

SATA, eMMC, SD cards, SPI/QSPI and NAND devices can hold up to four copies of the booting image. Therefore, the ROM code searches for one valid image out of the four copies, if present, by walking over the first blocks of mass storage space. Other XIP devices (NOR) use only one copy of the booting image.

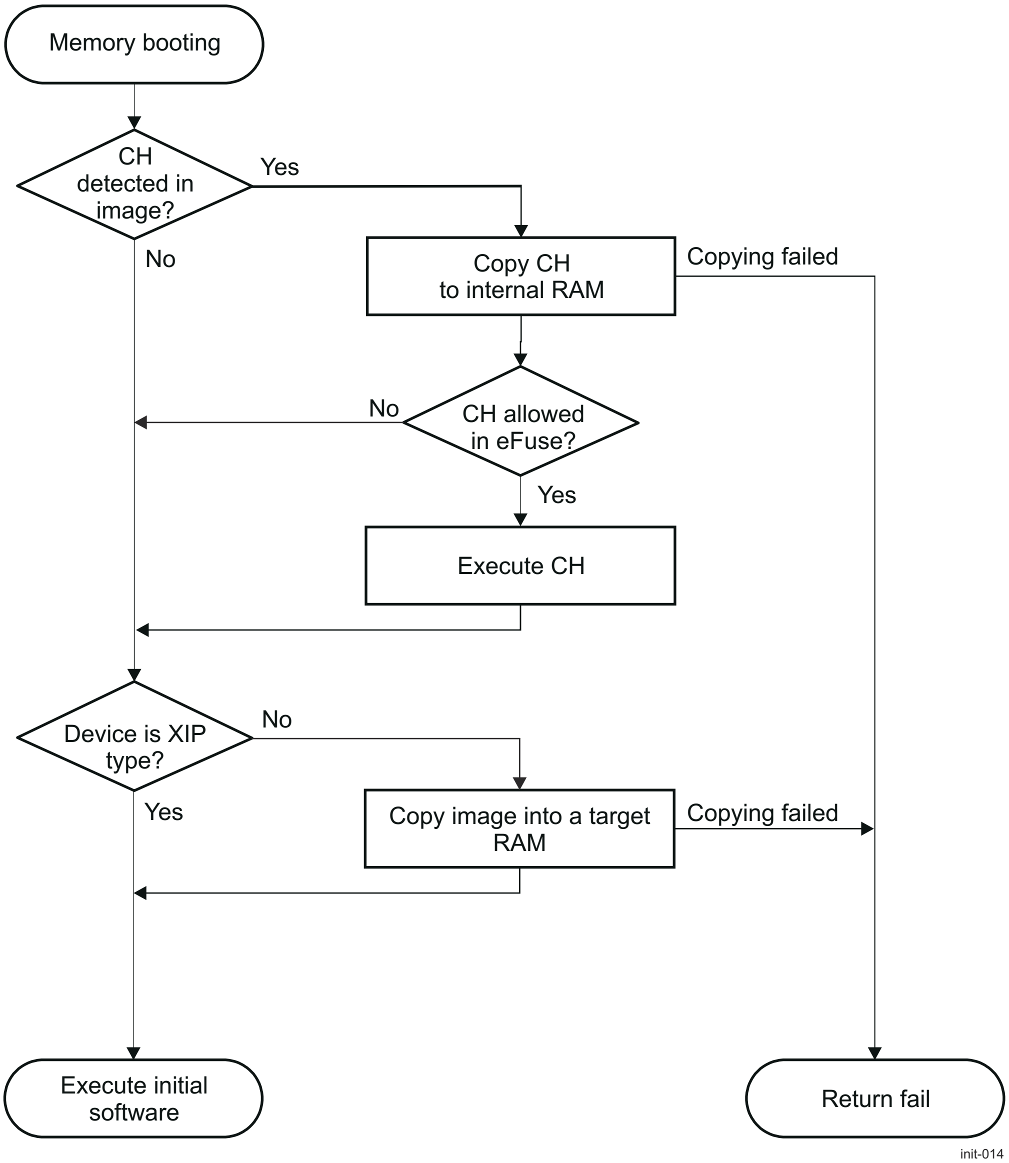


Figure 32-15. Memory Booting Procedure

32.3.7.2 Non-XIP Memory

Figure 32-16 shows the procedure used when memory booting runs with non-XIP memories. The shaded procedures are specific to each memory. The ROM code searches for the image in the first four physical blocks of the memories. Other devices use only one copy of the image and the block loop runs only once.

During image shadowing , the CH is expected to be in a separate sector before the initial software.

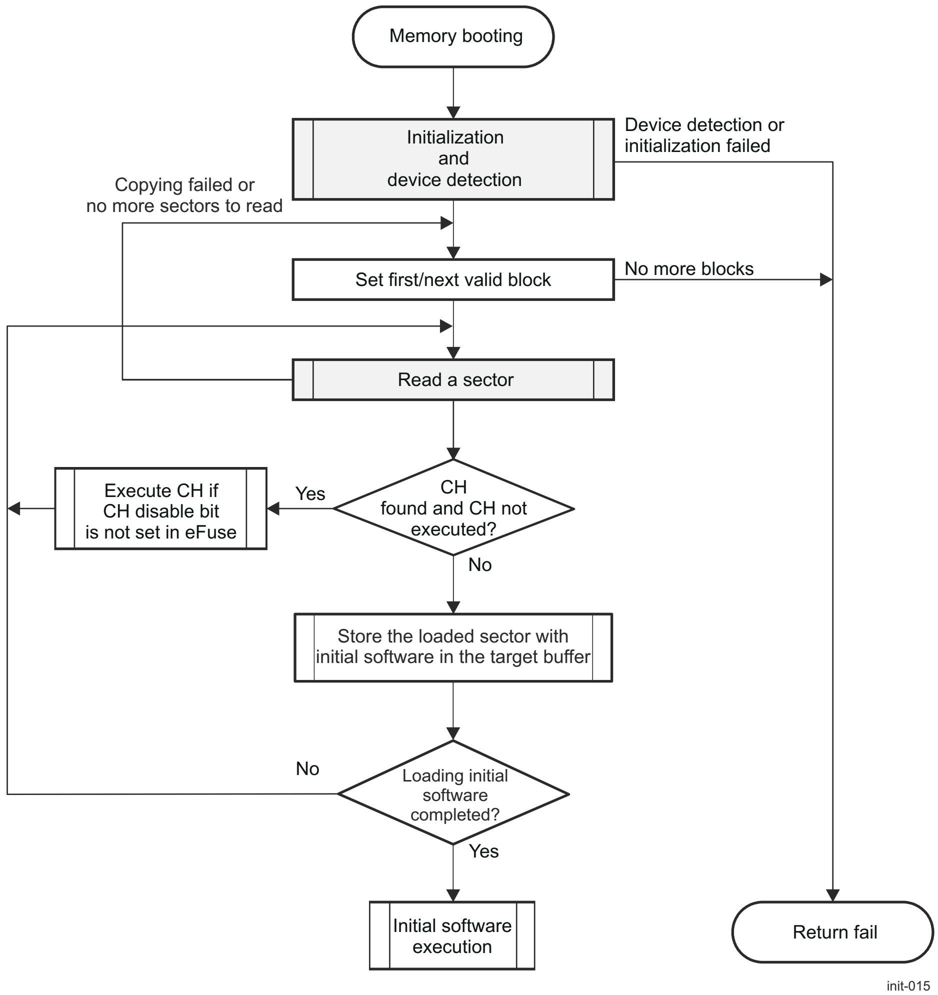


Figure 32-16. Image Shadowing

For more information about the GPMC module, see Section 15.4, *General-Purpose Memory Controller*.

The following sections describe the supported device types.

32.3.7.3 XIP Memory

The ROM code can boot directly from XIP devices, such as NOR flash memories, that have the following characteristics:

- The GPMC is the communication interface.
- Memories up to 1Gibit (128MiB) can be connected.
- 8-bit or 16-bit data bus width, configured through sysboot[13] (=1 for 16-bit)
- Non-muxed or address/data multiplexed mode, configured through sysboot[12:11] (=0x1 for A/D mux)
- The GPMC clock is 133 MHz.
- The device is connected to CS0 mapped to address 0x0800 0000.
- The wait pin gpmc_wait0 signal is monitored according to the sysboot[10] configuration pin (=1 is enabled)

For an XIP memory booting, no user intervention is required; the following debugging steps are described. Only the CH, which is not mandatory, lets users change clock settings and GPMC parameters. Failure in CH copying causes a return to the main booting procedure, which selects the next device for booting.

Booting from an XIP device consists of the following steps:

1. Configure the GPMC for XIP device access.

2. Verify that the CH is present at address 0x0800 0000. If the CH is present, copy the entire sector (512 bytes) to internal RAM and execute the CH.
3. Set the image location:
   - 0x0800 0000 if the CH is not found
   - 0x0800 0200 if the CH is found
4. Verify that a bootable image is at the image location.
5. If the image is found, execute it.
6. If the image is not found, return from XIP booting to the main booting loop.

32.3.7.3.1 GPMC Initialization

Table 32-38 lists the timing settings of the GPMC when set for XIP and other address-data accessible devices. Table 32-38 is included for debug information.

Table 32-38. XIP Timing Parameters

Parameter	Value (Clock Cycles)[(1)]	Register Initialization (where i = 0)
Write cycle period	17	The GPMC_CONFIG5_i[12:8] WRCYCLETIME bit field is set to 0x11.
Read cycle period	17	The GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field is set to 0x11.
CS low time	1	The GPMC_CONFIG2_i[3:0] CSONTIME bit field is set to 0x1.
CS high time	16	The GPMC_CONFIG2_i[12:8] CSRDOFFTIME bit field is set to 0x10.
ADV low time	1	The GPMC_CONFIG3_i[3:0] ADVONTIME bit field is set to 0x1.
ADV high time	2	The GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field is set to 0x2.
OE low time	3	The GPMC_CONFIG4_i[3:0] OEONTIME bit field is set to 0x3.
OE high time	16	The GPMC_CONFIG4_i[12:8] OEOFFTIME bit field is set to 0x10.
WE low time	3	The GPMC_CONFIG4_i[19:16] WEONTIME bit field is set to 0x3.
WE high time	15	The GPMC_CONFIG4_i[28:24] WEOFFTIME bit field is set to 0xF.
Data latch time	15	The GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field is set to 0xF.

(1) The one clock cycle is approximately 7.5 ns, which corresponds to a 133-MHz frequency.

There is no specific identification routine executed before booting from an XIP device.

32.3.7.4 NAND

NAND flash memory is not an XIP device; it requires shadowing before the code can be executed. ROM code support for the NAND flash devices has the following characteristics:

- The GPMC is the communication interface.
- Device from 512 Mibit (64 MiB) to 64 Gibit (8 GiB)
- ×8 and ×16 bus width
- Support for large page size (2048 bytes + 64 spare bytes) or very large page size (4096 bytes + 128/218 spare bytes)
- Chip enable (CE) don't-care devices only
- Single-level cell (SLC) and multilevel cell (MLC) devices
- Device identification based on ONFI or ROM table
  – Note that the full ONFI spec is not supported – specifically, the NAND geometry identification uses the first parameter page only and doesn’t check CRC on that page
- ECC correction: 8 bits per sector for most devices (16 bits per sector for devices with large spare area)
  – Note that ECC detection/correction is not used on ONFI parameter pages
- GPMC timings are adjusted for NAND access.
- The GPMC clock is 133 MHz.
- The device is connected to CS0.
- The gpmc_wait0 wait-pin signal is connected to the NAND BUSY output.
- Four physical blocks are searched for an image. Block size depends on the device.

For NAND memory booting, no user intervention is needed; the information in the following subsections is included for debugging. Only the CH, which is not mandatory, lets the user change clock settings and GPMC parameters. Failure in CH copying causes a return to the main booting procedure, which selects the next device for booting.

32.3.7.4.1 Initialization and NAND Detection

The initialization routine for NAND consists of three parts: GPMC initialization, device detection with parameter determination, and bad block detection.

- GPMC initialization

  The GPMC interface is configured so that it can access NAND. Because NAND memories do not need the address bus, it is released. The data bus width is initially set to 8 bits. If necessary, it is changed to 16 bits after the device parameters are determined. Table 32-39 shows the GPMC configuration used during NAND boot. Table 32-39 is included for debug information.

Table 32-39. NAND Timing Parameters

Parameter	Value (Clock Cycles)	Register Initialization (where i = 0–7)	Reset Value
Write cycle time	20	GPMC_CONFIG5_i[12:8] WRCYCLETIME = 0x14	0x11
Read cycle time	20	GPMC_CONFIG5_i[4:0] RDCYCLETIME = 0x14	0x11
CS low time	0	GPMC_CONFIG2_i[3:0] CSONTIME = 0x0	0x1
CS low to OE low time	5	GPMC_CONFIG4_i[3:0] OEONTIME = 0x5	0x6
CS low to OE high time	16	GPMC_CONFIG4_i[12:8] OEOFFTIME = 0x10	0x10
CS low to WE low time	1	GPMC_CONFIG4_i[19:16] WEONTIME = 0x1	0x5
CS low to WE high time	15	GPMC_CONFIG4_i[28:24] WEOFFTIME = 0xF	0x10
CS low to data latch time	14	GPMC_CONFIG5_i[20:16] RDACCESSTIME = 0xE	0xF

- Device detection and parameters

  The ROM code first performs an initial wait for device auto initialization (with a 250-ms time-out) with polling of the ready information. Then, it must identify the NAND type connected to the GPMC interface. The GPMC is initialized using 8 bits, asynchronous mode. The NAND device is reset (command FFh) and its status is polled until ready for operation (with a 100-ms time-out). The ONFI Read ID (command 90h/address 20h) is sent to the NAND device. If it replies with the ONFI signature (4 bytes) then a Read parameters page (command ECh) is sent. The information provided in Table 32-40 is then extracted: page size, spare area size, number of pages per block, and the addressing mode, and ECC. The remaining data bytes from the parameter page stream are ignored.

Table 32-40. ONFI Parameters Page Description

Offset	Description	Size (Bytes)
6	Features supported	2
80	Number of data bytes per page	4
84	Number of spare bytes per page	2
92	Number of pages per block	4
101	Number of address cycles	1

  If the ONFI Read ID command fails (it will be the case with any device not supporting ONFI), then the device is reset again with polling for device to be ready (with 100-ms time-out). Then, the standard Read ID (command 90h/address 00h) is sent. If the Device ID (second byte of the ID byte stream) is recognized as being a supported device, then the device parameters are extracted from an internal ROM code table. Table 32-41 lists the supported NAND devices.

Note

Not all NAND geometries are supported – the NAND device datasheet must be compared to Table 32-41 to confirm that the Read ID matches the Geometry specified in Table 32-41.

Table 32-41. Supported NAND Devices

Capacity	Device ID	Bus Width	Page Size in Bytes
512 Mibit	F0h	8	2048
512 Mibit	C0h	16	2048
512 Mibit	A0h	8	2048
512 Mibit	B0h	16	2048
512 Mibit	F2h	8	2048
512 Mibit	C2h	16	2048
512 Mibit	A2h	8	2048
512 Mibit	B2h	16	2048
1 Gibit	F1h	8	2048
1 Gibit	C1h	16	2048
1 Gibit	A1h	8	2048
1 Gibit	B1h	16	2048
2 Gibit	DAh	8	2048
2 Gibit	CAh	16	2048
2 Gibit	AAh	8	2048
2 Gibit	BAh	16	2048
2 Gibit	83h	8	2048
2 Gibit	93h	16	2048
4 Gibit	DCh	8	2048
4 Gibit	CCh	16	2048
4 Gibit	ACh	8	2048 (4096)
4 Gibit	BCh	16	2048 (4096)
4 Gibit	84h	8	2048
4 Gibit	94h	16	2048
8 Gibit	D3h	8	2048 (4096)
8 Gibit	C3h	16	2048 (4096)
8 Gibit	A3h	8	2048 (4096)
8 Gibit	B3h	16	2048 (4096)
8 Gibit	85h	8	2048
8 Gibit	95h	16	2048
16 Gibit	D5h	8	2048 (4096)
16 Gibit	C5h	16	2048 (4096)
16 Gibit	A5h	8	2048 (4096)
16 Gibit	B5h	16	2048 (4096)

Table 32-41. Supported NAND Devices (continued)

Capacity	Device ID	Bus Width	Page Size in Bytes
16 Gibit	86h	8	2048
16 Gibit	96h	16	2048
32 Gibit	D7h	8	2048 (4096)
32 Gibit	C7h	16	2048 (4096)
32 Gibit	A7h	8	2048 (4096)
32 Gibit	B7h	16	2048 (4096)
32 Gibit	87h	8	2048
32 Gibit	97h	16	2048
64 Gibit	DEh	8	2048 (4096)
64 Gibit	CEh	16	2048 (4096)
64 Gibit	AEh	8	2048 (4096)
64 Gibit	BEh	16	2048 (4096)

After retrieving parameters from the table, the page size and block size are updated based on the fourth byte of the NAND ID data. Because of inconsistency among manufacturers, only devices recognized to be at least 2 Gibit have these parameters updated. Therefore, the ROM code supports 4-KiB page devices, but only if their size, according to the table, is at least 2 Gibit. Devices that are smaller than 2 Gibit have the block size parameter set to 128 KiB (when the page size is 2 KiB). Table 32-42 lists the fourth ID data byte encoding used in the ROM code.

Table 32-42. Fourth NAND ID Data Byte

Item	Description	I/O Number							
		7	6	5	4	3	2	1	0
Page size	512 bytes							0	0
	2048 bytes							0	1
	4096 bytes							1	0
	8192 bytes							1	1
Cell type (1)	2 levels					0	0		
	4 levels					0	1		
	8 levels					1	0		
	16 levels					1	1		
Block size	64 KiB			0	0				
	128 KiB			0	1				
	256 KiB			1	0				
	512 KiB			1	1				

(1) Read by ROM code only when the manufacturer code (first ID byte) is 98h

Figure 32-17 shows the detection procedure. When the NAND device is successfully detected, the ROM code changes the GPMC to 16-bit bus width, if necessary.

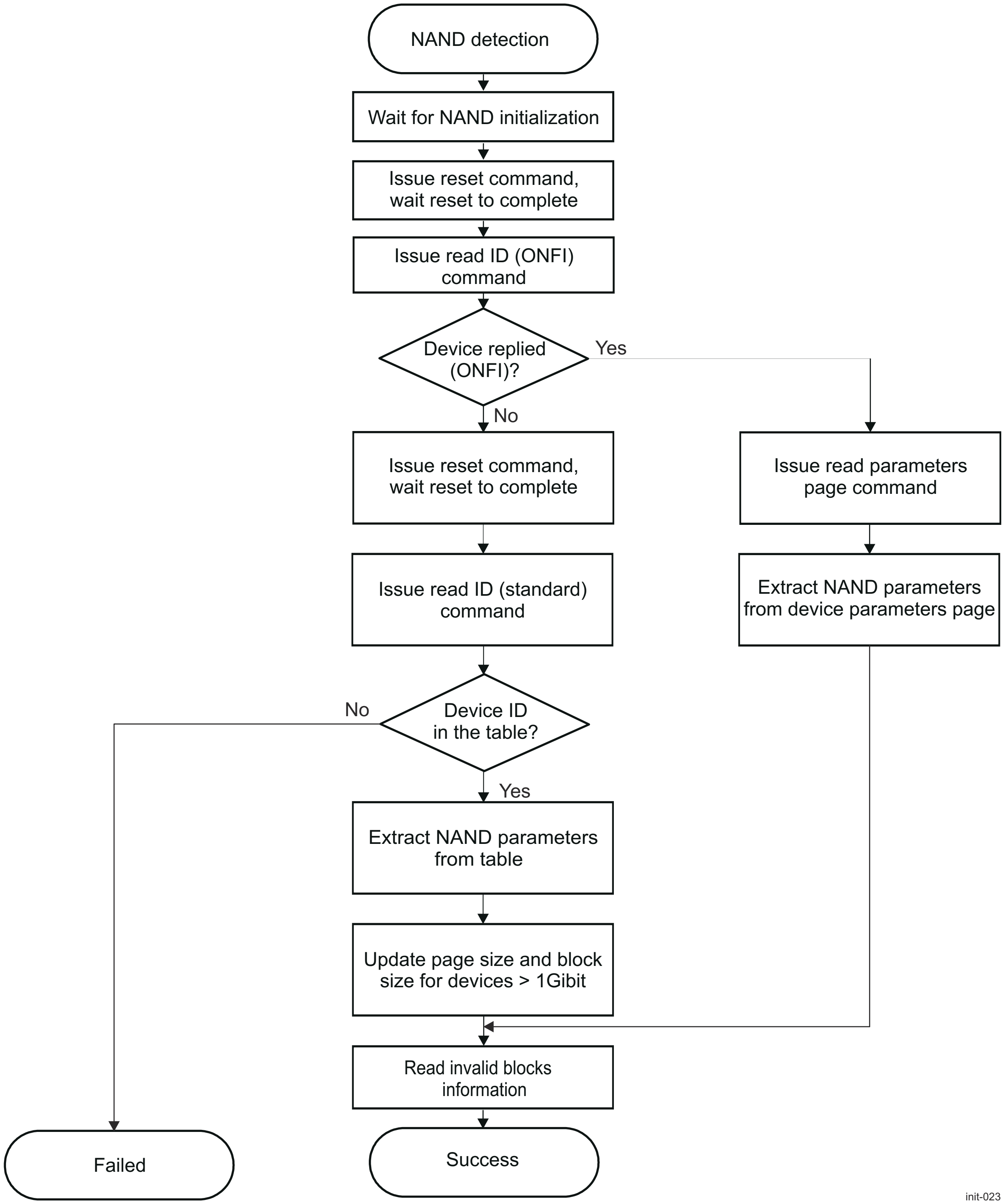


Figure 32-17. NAND Device Detection

- Bad block verification

  Invalid blocks contain invalid bits whose reliability cannot be ensured by the manufacturer. These bits are identified in the factory or during the programming and reported in the initial invalid block information in the spare area on the first and second page of each block. Because the ROM code looks for an image in the first four blocks, it detects the validity status of these blocks. Blocks detected as invalid are not accessed later.

Block validity status is coded in the spare areas of the first two pages of a block (first byte equal to FFh in the first and second pages for an 8-bit device/first word equal to FFFFh in the first and second pages for a 16-bit device).

Figure 32-18 shows the invalid block detection routine. The routine consists in reading spare areas and checking the validity data pattern.

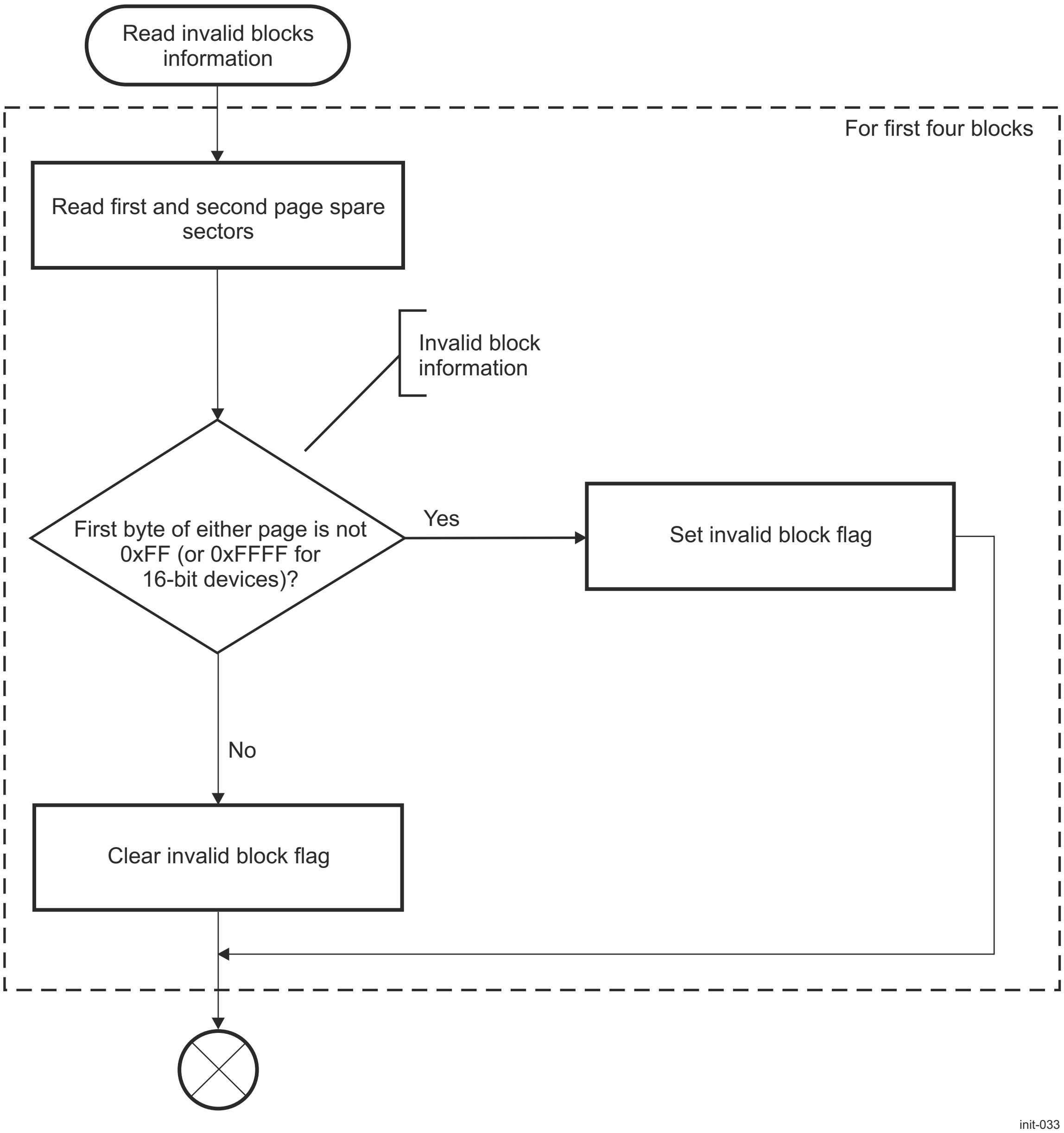


Figure 32-18. Bad NAND – Invalid Block Detection

32.3.7.4.2 NAND Read Sector Procedure

During the booting procedure, the ROM code reads 512-byte sectors from the NAND device. The reading fails in two cases:

- The accessed sector is in a block marked as invalid.
- The accessed sector contains an error that cannot be corrected with ECC.

The ROM code uses normal read (command 00h 30h) for reading NAND page data.

Page data can contain errors caused by memory alteration. The ROM code uses an ECC correction algorithm to detect and possibly correct those errors. The default ECC correction applied is BCH 8b/sector using the GPMC and ELM hardware.

For device ID codes D3h, C3h, D5h, C5h, D7h, C7h, DEh, and CEh when the manufacturer code (first ID byte) is 98h, the cell type information is checked in the fourth byte of ID data. If it is equal to 10b, the ECC correction applied is BCH 16b/sector.

The BCH data is automatically calculated by the GPMC on reading each 512-byte sector. The computed ECC is compared against the ECC stored in the spare area for the corresponding page. Depending on the page size, the amount of ECC data bytes stored in the corresponding spare area is different. Figure 32-19 and Figure 32-20 show the mapping of ECC data inside the spare area for 2-KiB page and 4-KiB page devices, respectively. If both ECC data are equal, the read sector function returns the read 512-byte sector without error. Otherwise, the ROM code tries to correct errors in the corresponding sector (this procedure is assisted by the ELM hardware) and returns the data if successful. If errors are uncorrectable, the function returns with FAIL.

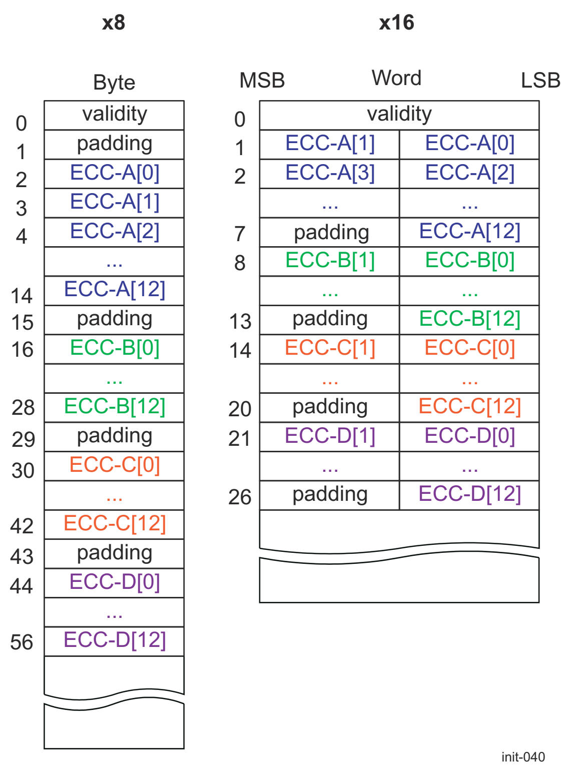


Figure 32-19. ECC Data Mapping for 2-KiB Page and 8b BCH Encoding

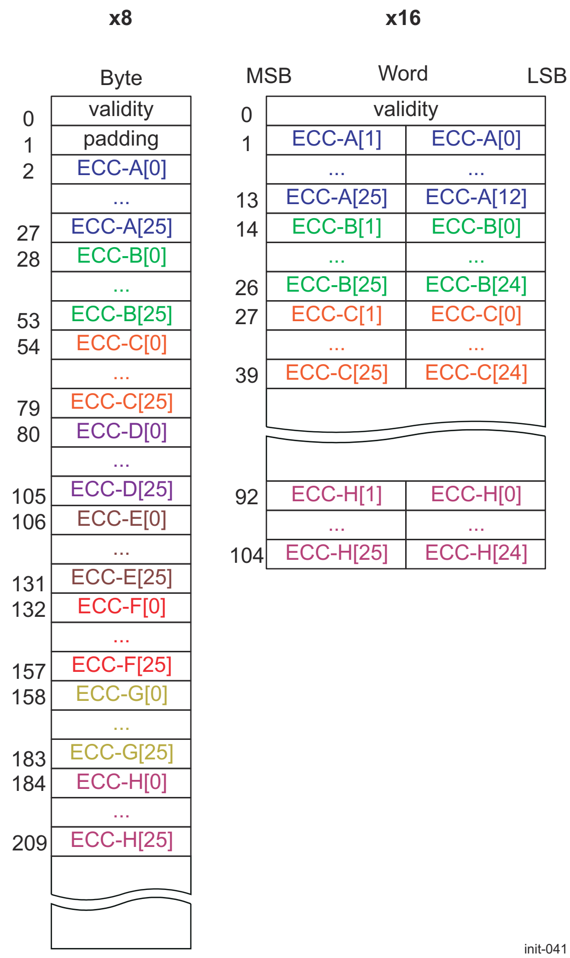


Figure 32-20. ECC Data Mapping for 4-KiB Page and 16b BCH Encoding

32.3.7.5 SPI/QSPI Flash Devices

SPI/QSPI Flash memories provide a storage solution for systems with limited space, pins, and power.

The ROM code support for SPI/QSPI devices has the following characteristics:

- 24-bit addressing, up to 128 Mbit (16 MiB), no banking
- QSPI1 on CS0 is the communication interface
- Uses Mode 3:
  - Clock inactive state = high
  - Data input captured on rising edge of clock
  - Data output generated on falling edge of clock
- QSPI 4-bit data read mode at 48 MHz in configuration port mode
  - Read command is 0x6B (Fast Quad Read), 3 address bytes,1 dummy bytes and read type is quad read
  - ROM will not perform any quad-enable sequence nor bank register update
- SPI 1-bit data read mode at 12 MHz in configuration port mode
  - Read command is 0x03 (Single Read), 3 address bytes, 0 dummy bytes and read type is normal read.
- Up to four redundant images can be stored on SPI/QSPI flash. The offset between them is set in sysboot[7:6] as described in Table 32-8. If not using the redundant SBL feature, the offset set in sysboot[7:6] is a don't care. A booting image is considered to be present , when the first 4 bytes word is not equal to 0000 0000h or FFFF FFFFh.

Note

ROM code does not perform any specific action to detect, reset or power up the QSPI device. QSPI is assumed to be properly powered and reset to be completed before every attempt to boot by ROM code.

32.3.7.6 eMMC Memories and SD Cards

The device allows booting from eMMC embedded memories or SD cards connected to device embedded MMC2 or MMC1 controllers I/Os, respectively. The booting interface is selected by configuration of the SYSBOOT pins.

The device high speed MMC/SD/SDIO host controller handles the physical layer, while the ROM code handles the simplified logical protocol layer (read-only protocol). A limited range of commands is implemented in the ROM code.

The SD card interface supports 1.8-V/3-V digital I/Os. The selection on I/O voltage level is done using the PBIAS circuitry. The PBIAS reference level is sensed by the ROM code and PBIAS cell voltage setup appropriately before any communication with the SD card device. The eMMC interface supports also 1.8-V/3-V digital I/Os but with the difference that there is no PBIAS circuitry to select the I/O voltage level. This voltage depends only on the voltage provided to the eMMC associated power pads.

The device supports three booting modes:

- Raw (Boot): The booting image is read from one of the selectable partitions in raw mode (this mode is also called Alternative Boot Operation mode and applies to eMMC only).
- Raw (UDA): The booting image is read from the user data area (eMMC and SD).
- File system (FAT12/16/32 with or without master boot record): The image data is read from a booting file within a file system in the user data area (eMMC and SD).

Note

File system mode is only supported when booting in the eMMC user area. It is not supported in the eMMC boot area.

32.3.7.6.1 eMMC Memories

The ROM code supports booting from eMMC memories, with the following conditions:

- eMMC memory devices compliant with *Embedded MultiMediaCard (eMMC) eMMC/Card Product Standard, High Capacity, including Reliable Write Boot, and Sleep Modes, Dual Data Rate, Multiple Partitions Supports and Security Enhancement v4.5* from the MMCA Technical Committee. The exception is the hardware reset feature. For example, if the user software requires eMMC device hardware reset, it can be accomplished with a GPIO. To correctly boot from eMMC when using Alternative Boot Operation mode, it is recommended to tie the eMMC device RST_N signal to the platform warm reset.
- eMMC device connected to the device MMC2 controller I/O interface (only one device can be connected to the bus).
- The eMMC memory device is powered externally by a PMIC or other power supply.
- When booting from user area: Initial (default) 1-bit SDR mode, optional 4-bit and 8-bit SDR- and DDR-modes using Configuration Header
- Initial SDR- and DDR- modes supported
- Clock frequency when booting from user area:
  - Identification mode: 400 kHz
  - Data transfer mode: 10 MHz, optionally up to 48 MHz by Configuration Header
  - Support for eMMC boot partitions. eMMC booting from boot partitions (BPs), which is typical for booting in Alternative Boot operation mode, is always done at 8-bit, 48 MHz, DDR.

32.3.7.6.1.1 System Conditions and Limitations

The ROM code does not provide a bus direction control signaling in case of using MMC interface 2 with level shifters.

Note

The ROM Code does not support large 4-KiB sectors as defined in the latest standard.

The ROM code expects that the eMMC device is powered externally and supplies are set and stable when entering the booting procedure.

The ROM code supports the alternative boot operation mode specific to eMMC devices as described in the *Partition Management* and *Boot Operation Mode* sections of the eMMC Standard. It does not support the hardware boot operation mode (CMD line held low for 74 emmc_clk cycles) described in the same document.

32.3.7.6.1.2 eMMC Memory Connection

An eMMC device typically requires two supplies: one for the core memory array (VCC) and one for the device interface I/Os and controller (typical VCCQ = 1.8 V). Both memory device supply pins can possibly be merged into one, thus requiring only one power supply.

Figure 32-21 shows an example of the system connection between the PMIC, memory device, and device.

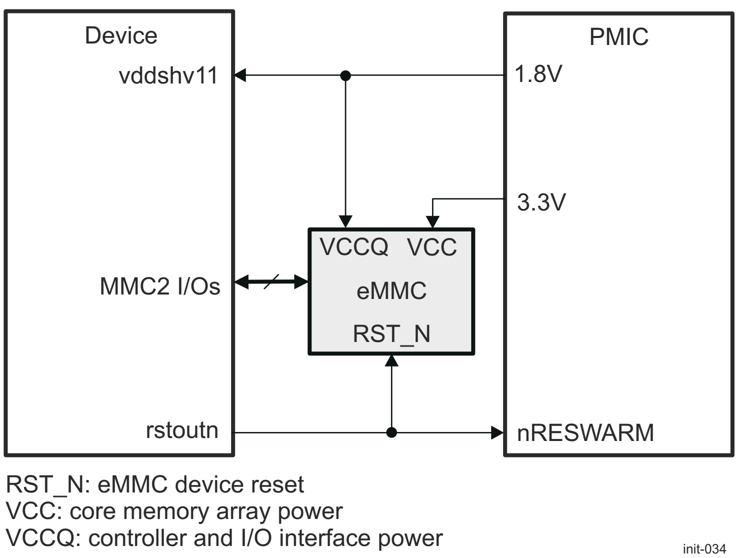


Figure 32-21. eMMC Connection

Note

For correct handling of eMMC boot partitions when using alternative boot operation mode, it is recommended to tie the eMMC RST_N signal to the signal indicating a platform warm reset (nRESWARM).

The ROM code performs the necessary I/O pin muxing configuration to route the MMC2 I/O signals on the eMMC pads depending on the selected SYSBOOT configuration.

32.3.7.6.2 SD Cards

The ROM code supports booting from SD cards under the following conditions:

- SD cards compliant with *SD Specifications Part 1 Physical Layer Specification Version 4.00*and the *SD Specifications Part 2 File System Specification Version 3.00* from the SD Association. These include low- (SDSC) and high-capacity (SDHC) cards.

- SD card connected to MMC1 controller I/Os. Only one card is allowed to be connected to the bus.
- 3-V VCC power supply
- 3-V or 1.8-V I/O voltages
- Initial 1-bit MMC mode, optional 4-bit mode
- Clock frequency:
  - Identification mode: 400 kHz
  - Data transfer mode: 10 MHz (optionally up to 19.2 MHz by Configuration Header)

32.3.7.6.2.1 System Conditions and Limitations

Even though the ROM Code identities SDXC cards, it does not handle its specificities such as exFAT support. It does neither support UHS-I nor UHS-II speed grades.

32.3.7.6.2.2 SD Card Connection

An SD card can be connected to the SD card interface, typically through a card cage.

Figure 32-22 shows the typical connection between the power IC, the card, and the device.

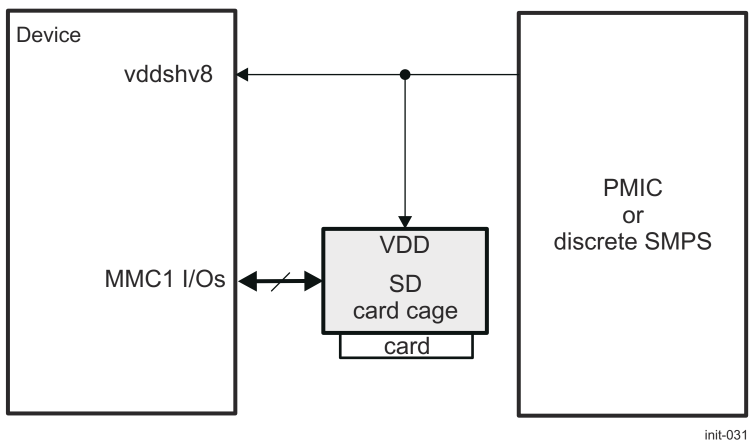


Figure 32-22. MMC/SD Card Connection

Note

The ROM code does not handle the card detection feature on the card cage.

32.3.7.6.2.3 Booting Procedure

If the selected booting device is eMMC, then a first attempt to trigger alternative boot operation mode is tried. If the eMMC device replies within a fixed timeout, then the booting image is directly retrieved from one of the possible partitions specified in the local EXT_CSD.BOOT_PARTITION_ENABLE bit field for the eMMC device (this is referred to as the Raw(Boot) mode). Raw(Boot) mode improves the boot time by avoiding the regular eMMC/SD identification phase (usually done at low speed). When the booting image is retrieved, the ROM code continues the boot process.

If the time-out elapses, then the ROM code proceeds with normal identification. SD cards do not support Alternative Boot Operation mode, and therefore are always going through normal identification.

If normal identification succeeds, then it next determines whether the SD card or eMMC device contains a known file system. If the file system is known, then the booting image is extracted from the file system hierarchy. If a file system is not detected, then the Raw(UDA) mode is assumed.

Figure 32-23 is the high level flow chart of the eMMC and SD booting procedure.

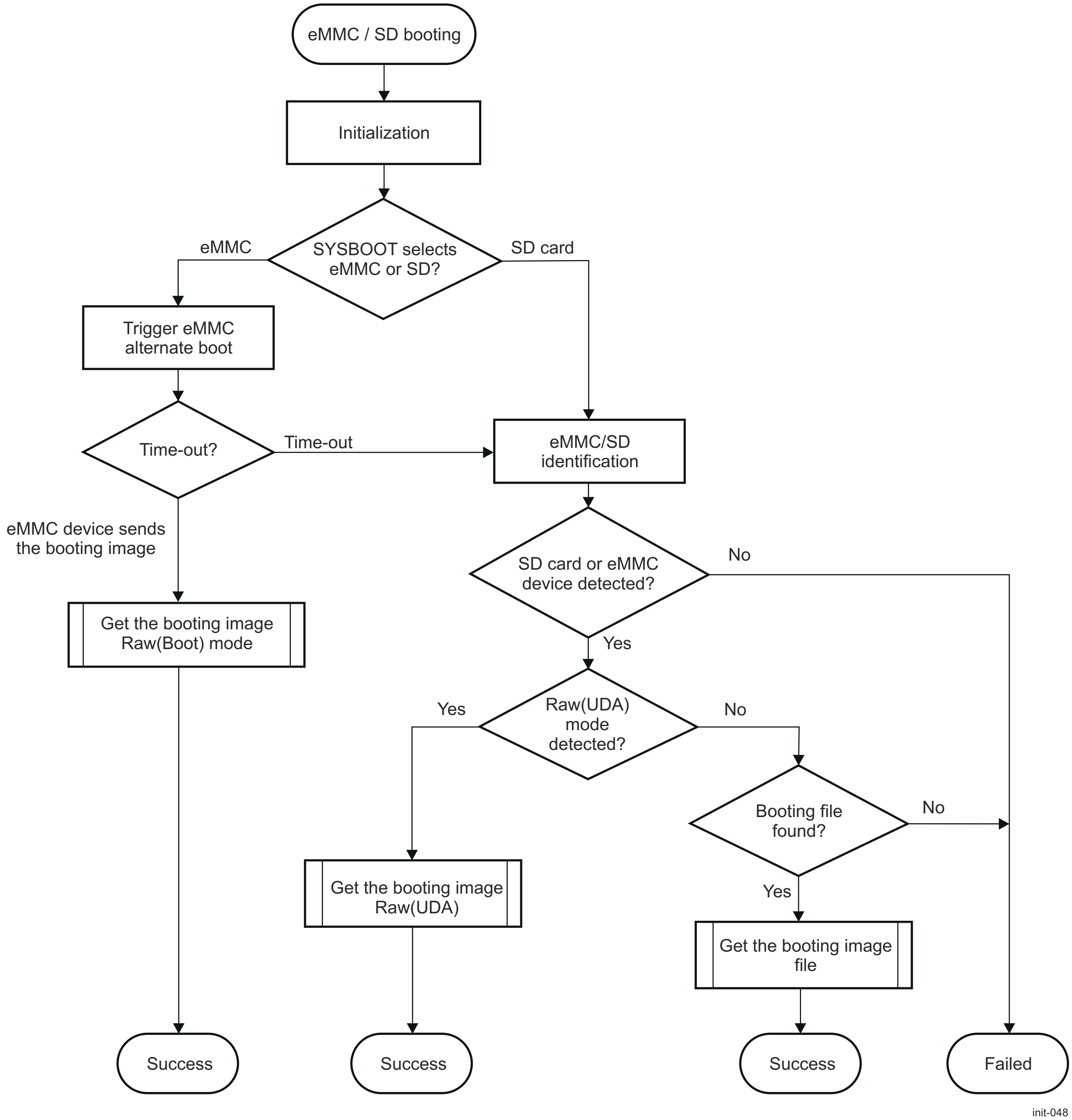


Figure 32-23. eMMC and SD Booting

32.3.7.6.2.4 eMMC Partitions Handling in Alternative Boot Operation Mode

To trigger alternative boot operation mode, the ROM code sends a specific CMD0 + Argument 0xFFFF FFFA. Then, the ROM code waits up to 50 ms for the eMMC device to return a boot-acknowledge signal. If the time-out elapses, then the ROM code skips alternative boot operation mode. Booting from eMMC partitions can be done in 8-bit mode at 48 MHz, DDR. This configuration is static and cannot be changed.

32.3.7.6.2.4.1 eMMC Devices Preflashing

For correct handling of eMMC partitions in alternative boot operation mode, it is necessary to prepare the eMMC device at flashing time with the following settings applied to the device EXT_CSD register:

- The BOOT_ACK and BOOT_PARTITION_ENABLE fields must be updated accordingly (EXT_CSD[179], bit 6 and bits [5:3], respectively) to activate the boot acknowledge signal and select the partition from which to boot. There are several boot options selectable in the BOOT_PARTITION_ENABLE bit field, as follows:
  – Booting from boot partition 1 (BP1)
  – Booting from boot partition 2 (BP2)
  – Booting from User Area

  For more details, see the eMMC Standard documentation.
- RST_N must be enabled for correct handling of the warm reset cases (EXT_CSD[162] bits [1:0] = 0x1).
- The BOOT_BUS_WIDTH fields (EXT_CSD[177]) must be updated properly based on device alternate boot operation: 8-bit, 48 MHz, DDR mode.
- Optionally the BOOT_CONFIG_PROT (EXT_CSD[178]) may be updated for altering access permissions to the selected partitions.

Note

Although an alternate boot from user area is possible, alternate booting from BP1 or BP2 is recommended.

Note

It is possible to permanently or temporarily lock the boot configuration through the use of the BOOT_CONFIG_PROT register. For more details, see the eMMC Standard documentation.

Note

- It is highly recommended to refer to the eMMC Standard for details on EXT_CSD handling.
- The EXT_CSD is updated by using a SWITCH command as detailed in the eMMC standard.
- The mentioned EXT_CSD fields are retained after a power cycle so only one “flashing” phase is required

32.3.7.6.2.4.2 eMMC Device State After ROM Code Execution

If alternative boot operation mode is successful, and the booting image is properly retrieved, then the eMMC device state remains in the BOOT state even after the execution has passed to the initial software. The initial software must bring the device out of the BOOT state.

32.3.7.6.2.4.3 Consideration on device Global Warm Reset

Once the system has booted and upon triggering a warm reset at device level (it can be triggered by the warm-reset push button, a timer, watchdog, etc.), it is important that the eMMC device is properly brought to its PRE-IDLE state so that the next Alternative Boot operation succeeds. This is ensured by the eMMC RST_N pin to be connected to device rstoutn pin (warm reset). Not doing so would make the boot procedure fail after a warm reset.

32.3.7.6.2.4.4 Booting Image Size

In Raw(Boot) mode, the size of the booting image is determined after the first sector read access. The boot image size is contained within the GP header. This ensures that the ROM code is only retrieving the necessary size of data and not the full contents of the partition.

32.3.7.6.2.4.5 Booting Image Layout

In Raw(Boot) mode only one copy of the booting image is maintained, as opposed to up to four copies in Raw(UDA) mode.

32.3.7.6.3 Initialization and Detection

If the eMMC Alternate Boot operation fails in time-out, or the requested boot device is an SD card, the ROM code initializes the memory device or card connected on the eMMC and SD card interface using 1.8 V for eMMC and standard high-voltage range (3.0 V) for SD card I/Os. If neither a card nor memory device is detected, the ROM code moves to the next booting device. The standard identification process and relative card address (RCA) assignment are performed. The ROM code assumes that only one memory or card is connected on the bus. This is done using the CMD line common to the SD and eMMC memory devices. The eMMC and SD standards describe this phase as the initialization phase. They differ in the commands involved, as described in Figure 32-25. The ROM code uses CMD55 to discriminate between eMMC and SD cards; that is, CMD55 is only supported by the SD standard. Depending on response received or not, ACMD41 (the combination of CMD55 and CMD41 for SD) or CMD1 (for MMC) is sent. If no response is received this time, no devices are connected and the ROM code exits eMMC/SD booting with FAIL.

Figure 32-24 shows the eMMC/SD detection procedure.

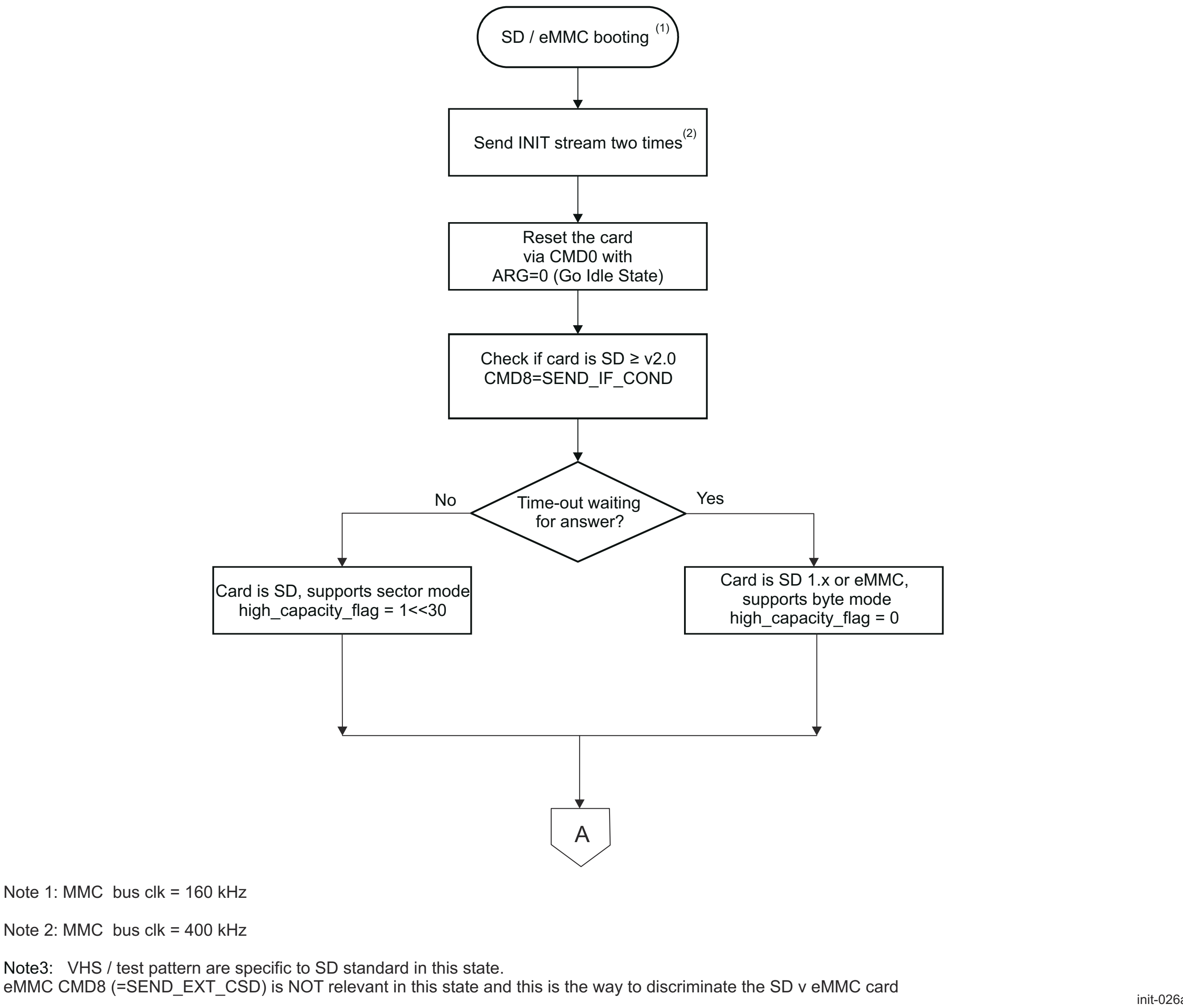


Figure 32-24. SD/eMMC Detection Procedure (part 1)

Figure 32-25 shows the eMMC/SD detection procedure.

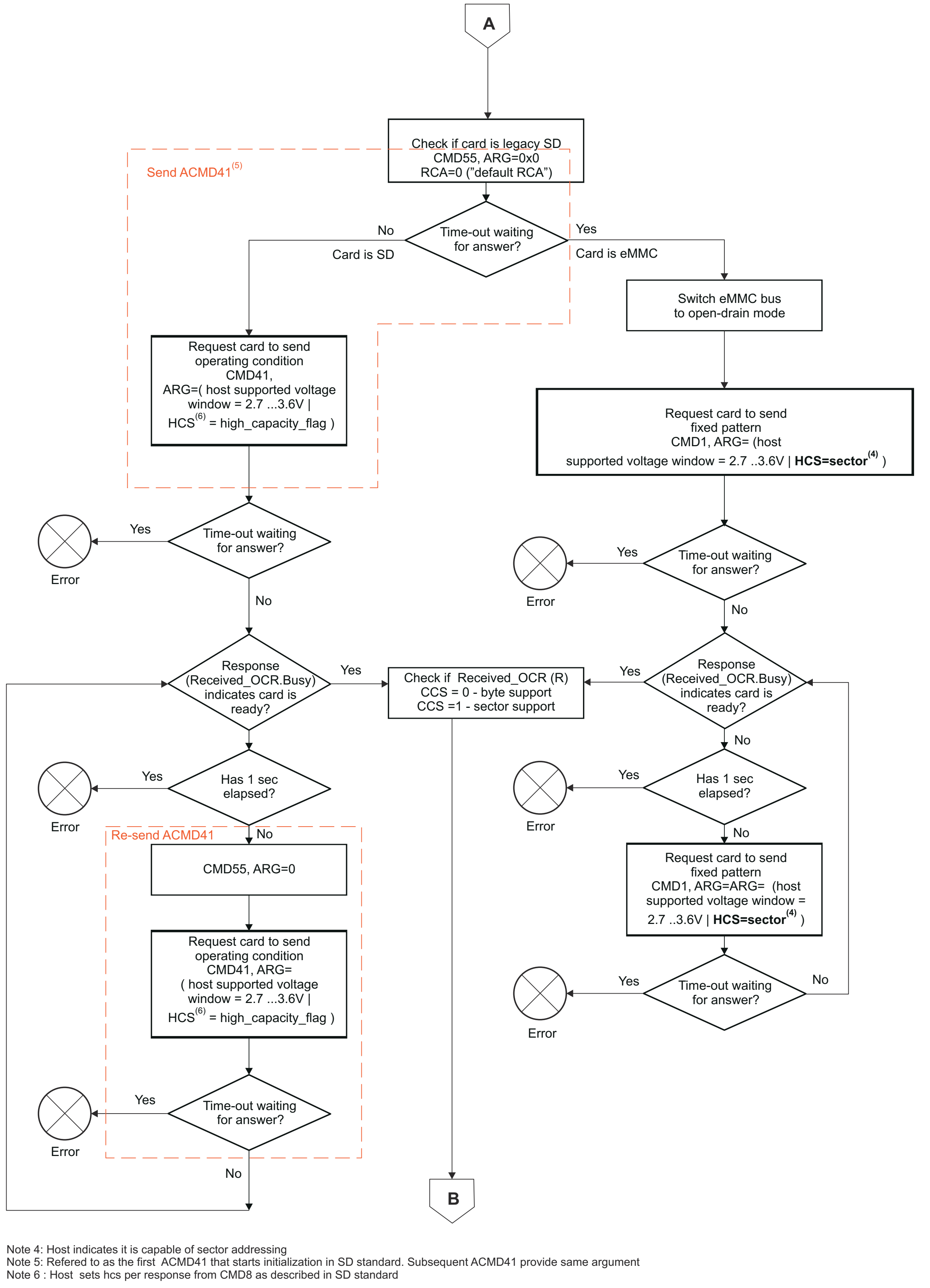


Figure 32-25. SD/eMMC Detection Procedure (part 2)

Figure 32-26 shows the eMMC/SD detection procedure.

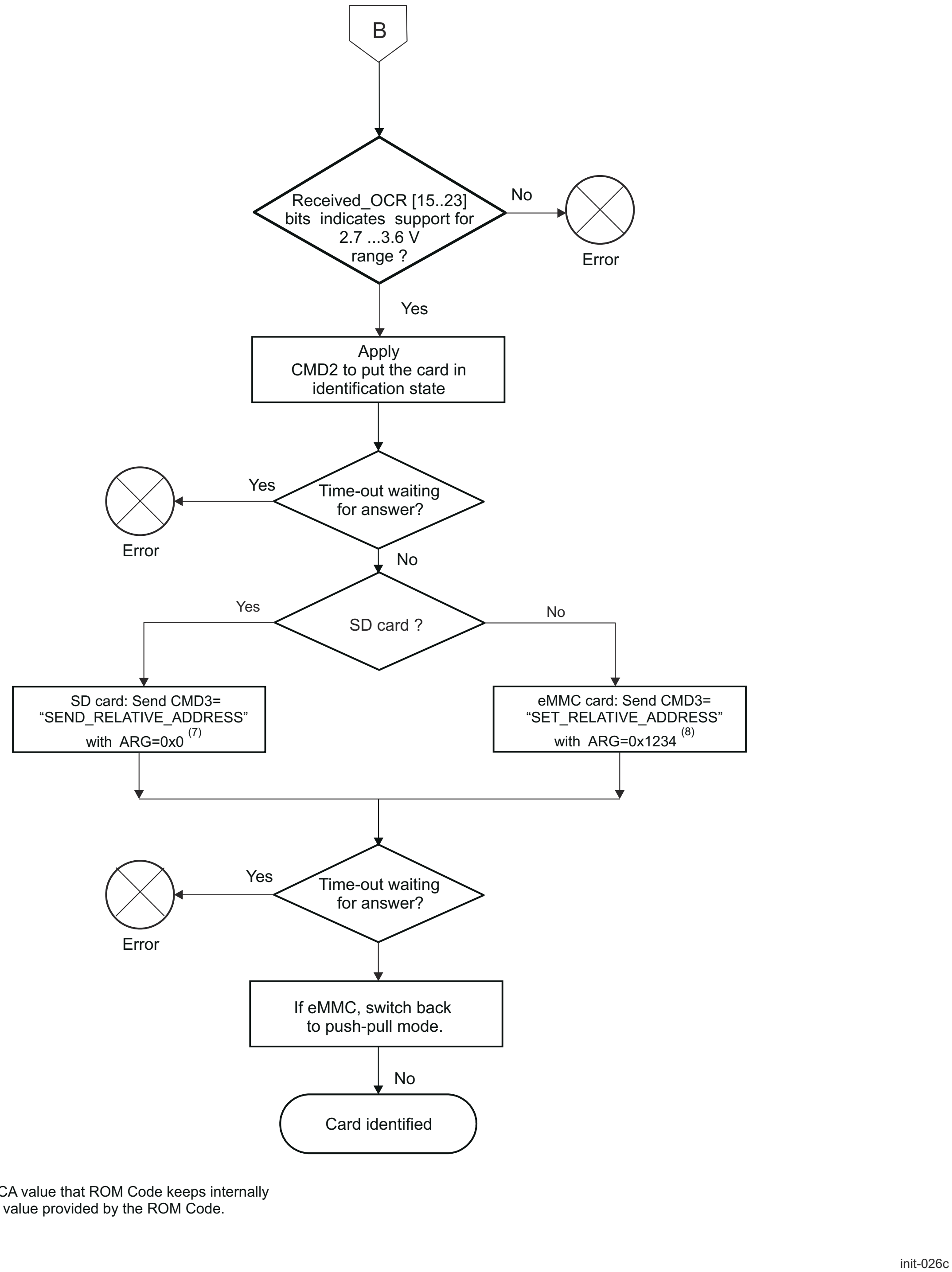


Figure 32-26. SD/eMMC Detection Procedure (part 3)

32.3.7.6.4 Read Sector Procedure

The contents of an eMMC or SD card may be formatted as raw binary (referred to as Raw(UDA) or within a FAT file system. The ROM Code reads out sectors from raw image or the booting file within the file system and boots from it.

- **Raw mode**

  In Raw(UDA) mode, an image can be at one of the four consecutive locations in the main area: offset 0x0 (0 KiB)/0x20000 (128 KiB)/0x40000 (256 KiB)/0x60000 (384 KiB). For this reason, the size of a booting image must not exceed 128 KiB. However, a device with an image greater than 128 KiB can be flash starting at one of the aforementioned locations. Therefore, the ROM code does not check the image size. The only drawback is that the image crosses the subsequent image boundary. Raw mode is detected by reading sectors 0, 256, 512, and 768. The content of these sectors is verified for the presence of a TOC structure. GP header must be located at the beginning of the booting image, as described in Section 32.3.8.2, *Configuration Header* . Image data is read directly from continuous sectors of a card. If raw mode is not detected, file system mode is assumed.
- File system handling

  The read sector procedure uses the standard SD/eMMC read data procedure. The sector address is generated based on the booting memory file map collected during initialization. Thus, the ROM code can freely address sectors in the booting file space.

32.3.7.6.5 File System Handling

The eMMC/SD cards can hold a file system that the ROM code reads. The image used by the booting procedure is taken from a booting file named MLO. This file must be in the root directory on an ***active*** primary partition of type FAT12/16 or FAT32.

An eMMC/SD card can be configured as floppy-like or hard-drive-like:

- When acting like a floppy, the content of the card is a single FAT12/16/32 file system without an MBR holding a partition table.
- When acting like a hard drive, an MBR is present in the first sector of the card. This MBR holds a table of partitions, one of which must be FAT12/16/32, primary, and active.

According to the *MultiMediaCard FAT16 File System Specification* from the MMCA Technical Committee, the card must always hold an MBR, except when using a floppy-like file system. However, depending on the operating system used, the eMMC/SD card is formatted with or without partitions (using an MBR). The ROM code supports both types: floppy-like or hard-drive-like. The ROM code retrieves a map of the booting file from the FAT. The booting file map is a collection of all FAT entries related to the booting file (a FAT entry points to a cluster holding part of the file). The booting procedure uses this map to access any 512-byte sector in the booting file without involving the ROM code FAT module. Figure 32-27 shows the complete process.

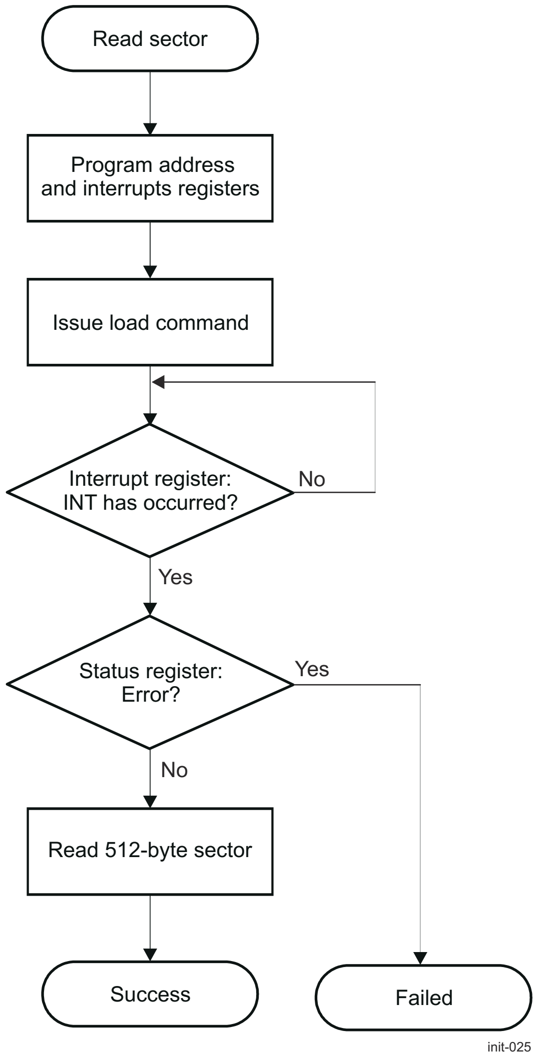


Figure 32-27. SD/MMC Get Booting File

32.3.7.6.5.1 MBR and FAT File System

This paragraph describes functions used by the ROM code to recognize whether an MBR with a FAT is used. It is not intended to fully describe the MBR and the FAT file system detection and reading procedure. The ROM code can detect FAT12/16/32 allocation table types. It cannot boot on devices with NTFS or Linux® FS partitions. Some memory devices that support file systems can be formatted with or without MBR; therefore, the first task of the ROM code is to detect whether the device is holding an MBR in the first sector.

The MBR is the first sector of a memory device. It consists of executable code, four partition entries, and one signature. The aim of such a structure is to divide the hard disk in partitions used primarily to boot different systems (for instance, Microsoft Windows®). Table 32-43 describes this structure, and Table 32-44 describes the partition table entry.

Table 32-43. Master Boot Record Structure

Offset	Length (Bytes)	Entry Description
0000h	446	Optional code
01BEh	16	Partition table entry
01CEh	16	Partition table entry
01DEh	16	Partition table entry
01EEh	16	Partition table entry

Table 32-43. Master Boot Record Structure (continued)

Offset	Length (Bytes)	Entry Description
01FEh	2	Signature (0xAA55)

Table 32-44. Partition Table Entry

Offset	Length (Bytes)	Entry Description	Value
0000h	1	Partition state	00h: Inactive 80h: Active
0001h	1	Partition start head	Hs
0002h	2	Partition start cylinder and sector	Cs[7:0] – Cs[9:8] – Ss[5:0]
0004h	1	Partition type	01h: FAT12 04h, 06h, 0Eh: FAT16 0Bh, 0Ch, 0Fh: FAT32
0005h	1	Partition end head	He
0006h	2	Partition end cylinder and sector	Ce[7:0]–Ce[9:8]–Se[5:0]
0008h	4	First sector position relative to the beginning of media	LBAs = Cs.H.S + Hs.S + Ss – 1
000Ch	4	Number of sectors in partition	LBAe = Ce.H.S + He.S + Se – 1 Nb s = LBAe – LBAs + 1

SD/eMMC booting consists of the following steps:

1. Detection of MBR

   The ROM code first checks whether the MBR signature is present, and then it searches an active FAT12/16/32 partition in all four MBR partition entries, based on the Type field. If the MBR entries are not valid, or if no usable partition is found, the ROM code returns to the booting procedure with FAIL. The extended partitions are not checked; the booting file must reside in a primary partition. Each partition entry is checked to determine the following:

   a. If the partition type is set to 00h, all fields in the entry must be 00h.
   b. The partition is within physical boundaries (that is, the partition is inside and it fits the total physical sectors).

   See Figure 32-28 for more information about MBR detection.

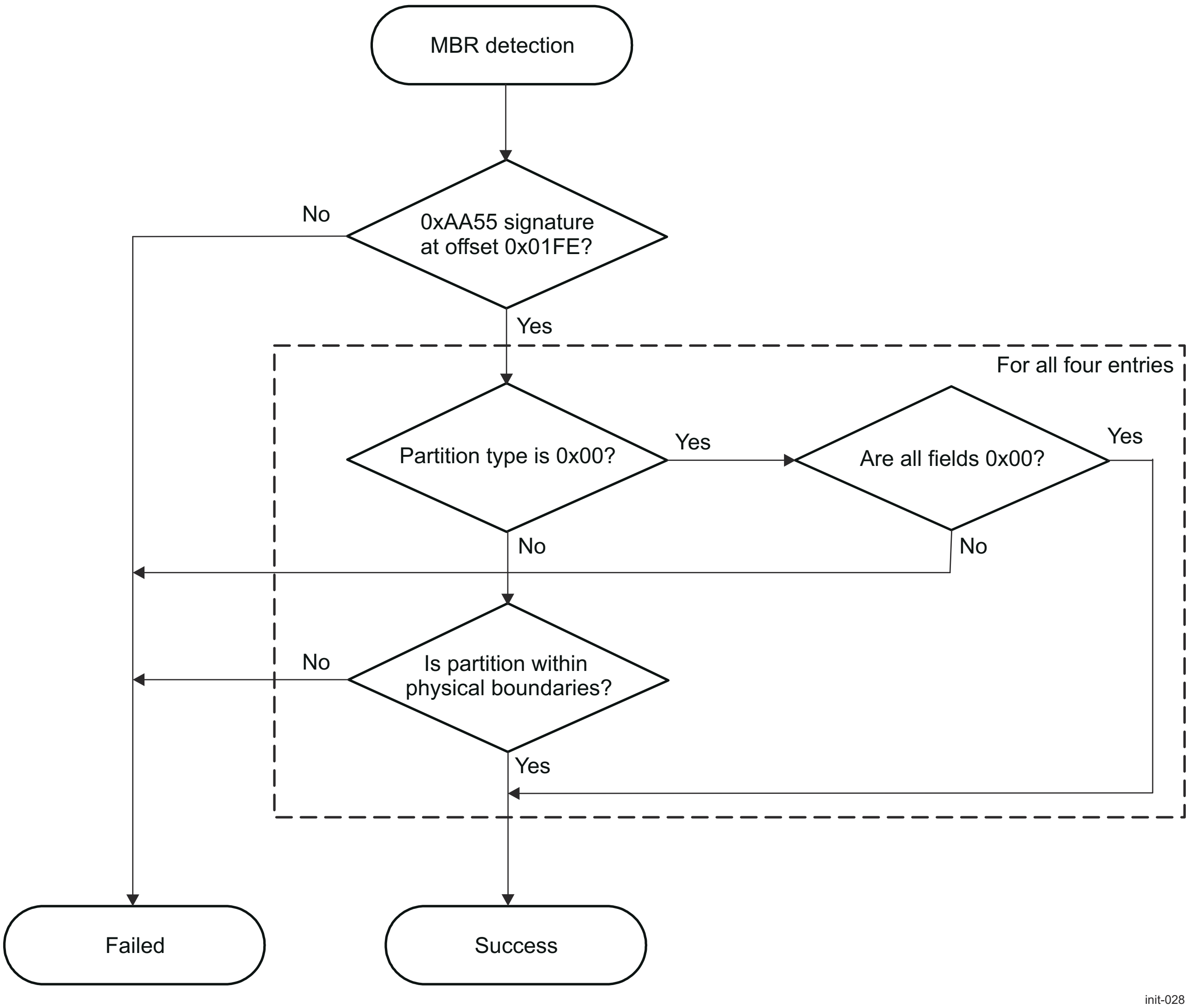


Figure 32-28. MBR Detection Procedure

2. Get the MBR partition.

   Once identified, the ROM code gets the partition using the procedure described in Figure 32-28. The partition type is checked to be FAT12/16 or FAT32. Its state must be 00h (inactive) or 80h (active). The ROM code returns with FAIL if no active primary FAT12/16/32 is found, or the test fails if there is more than one active partition. If an active partition is found, its first sector is read and used later. If no MBR is present (in case of a floppy-like system), the first sector of the device is read and used later. The read sector is checked to be a valid FAT12/16 or FAT32 partition. If this fails, the ROM code returns with FAIL if another partition type is used (for instance, Linux FS) or if the partition is not valid.

   The FAT file system consists of several parts:
   - Boot sector, which holds the BIOS parameter block (BPB). Not all are used by the ROM code.
   - FAT, which describes the use of each cluster of the partition
   - Data area, which holds the files, directories, and root directory (for FAT12/16, the root directory has a specific fixed location)

   To check whether a sector holds a valid FAT12/16/32 partition, many fields of the boot sector (used by all FAT types) that require specific values are checked. Figure 32-29 shows more about getting the partition.

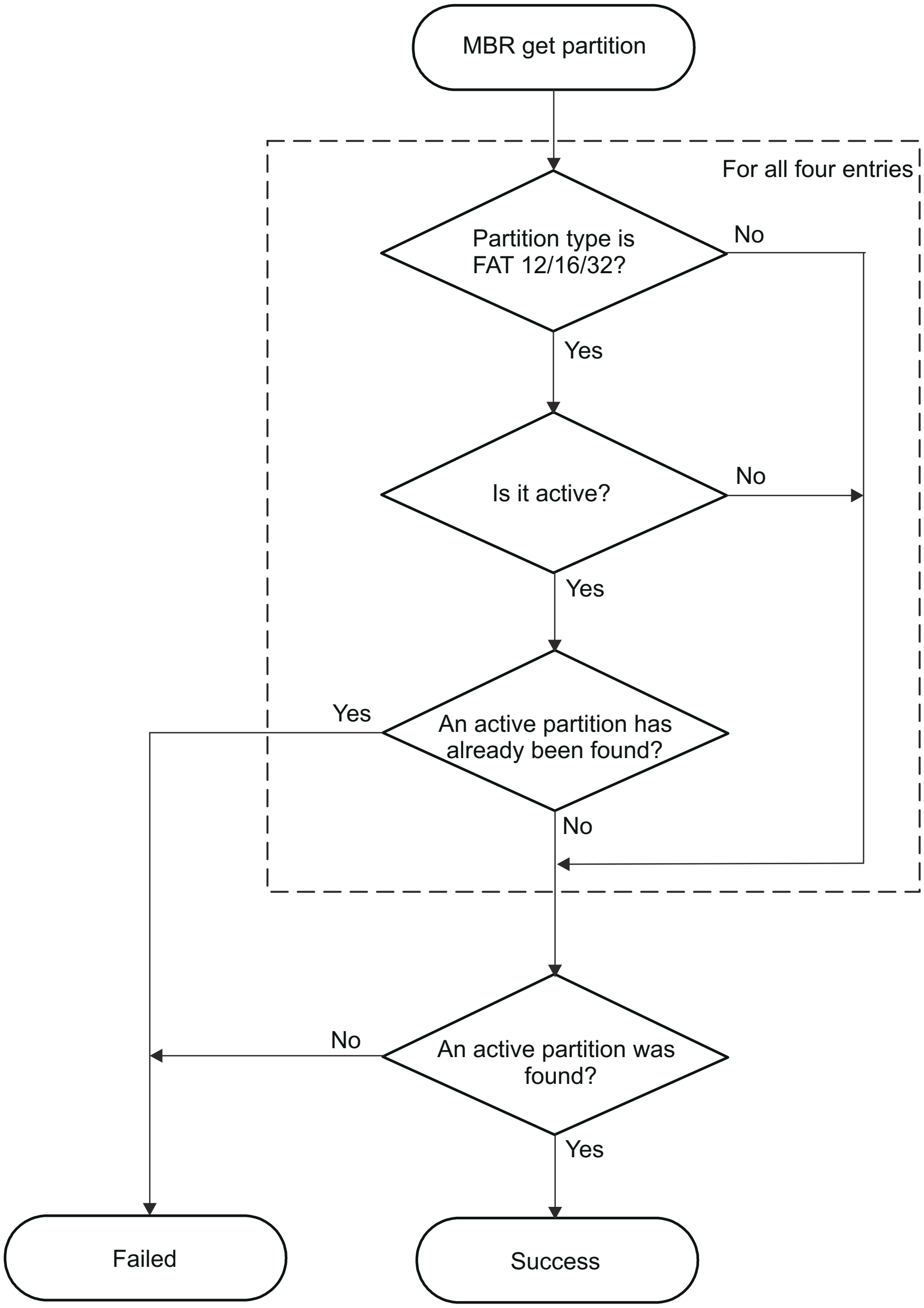


Figure 32-29. MBR, Get Partition

3. Find the booting file.

   When a partition is found, the root directory entries are searched for a booting file named MLO in the root directory of the FAT12/16/32 file system. The file is not searched in any other location. For a FAT12/16 file system, the root directory has a fixed location, which is cluster 0. For a FAT32 file system, its cluster location is given by BPB_RootClus. The formula to find the sector number (relative to device sector 0, not partition sector 0) of a cluster is given by the following equation:

$$Cluster_{sector} = BPB\_HiddSec + BPB\_RsvdSecCnt + BPB\_NumFATs \cdot BPB\_FATSz + Cluster \cdot BPB\_SecPerCLus$$

init-E001

Note

BPB_FATSz is BPB_FATSz16 for FAT12/16, or BPB_FATSz32 for FAT32.

Note

The BPB_HiddSec field can contain 0, even though the FAT file system is somewhere other than on sector 0 (floppy-like). The ROM code uses the partition offset taken from the MBR instead of this field, which can be wrong. If no MBR is found (floppy-like), the value 0 is used.

Each entry in the root directory is 32 bytes and holds information about the file (the filename, date of creation, rights, cluster location, and so forth). See Table 32-45.

Table 32-45. FAT Directory Entry

Offset	Length (Bytes)	Name	Description
0000h	11	DIR_Name	Short Name (8 + 3)
000Bh	1	DIR_Attr	File attributes: 01h – ATTR_READ_ONLY 02h – ATTR_HIDDEN 04h – ATTR_SYSTEM 08h – ATTR_VOLUME_ID 10h – ATTR_DIRECTORY 20h – ATTR_ARCHIVE 0Fh – ATTR_LONG_NAME
000Ch	1	DIR_NTRes	Reserved. Set to 00h.
000Dh	1	DIR_CrtTimeTenth	Millisecond stamp at file creation
000Eh	2	DIR_CrtTime	Time file was created.
0010h	2	DIR_CrtDate	Date file was created.
0012h	2	DIR_LstAccDate	Last access date
0014h	2	DIR_FstClusHi	High word of the first cluster number of this entry
0016h	2	DIR_WrtTime	Time of last write
0018h	2	DIR_WrtDate	Date of last write
001Ah	2	DIR_FstClusLo	Low word of the first cluster number of this entry
001Ch	4	DIR_FileSize	File size in bytes

The ROM code checks each entry in the root directory until either the booting file is found or the entry is empty (first byte is 00h), or when the end of the root directory is reached. Entries with the ATTR_LONG_NAME attribute (LFN) and first byte at E5h (erased file) are ignored. When found, the first cluster offset of the file is read from the DIR_FstClusHi/DIR_FstClusLo fields. There is a slight difference between FAT12/16 and FAT32 when handling the root directory. On FAT12/16, this directory has a fixed location (see Table 32-45) and length fixed by BPB_RootEntCnt, which is the total of 32-byte entries. Therefore, handling this directory is straightforward. On FAT32, the root directory is like a standard file. The FAT must be used to retrieve each sector of the directory. Step 4 describes the way in which the FAT is handled.

4. Buffer FAT entries in the FAT buffer.

When the booting file is found, the ROM code reads the FAT and buffers the singly-linked chain of clusters in the FAT buffer that is used during boot to access the booting file directly, sector by sector. For FAT12/16 and for FAT32, multiple copies of the FAT exist (ROM code supports only two copies), after the boot sector.

$$FATn_{sector} = BPB\_HiddSec + BPB\_RsvdSecCnt + BP\_FatSz \cdot n$$

init-E002

The size of the FAT buffer is given by BPB_FATSz16 or BPB_FATSz32. The ROM code checks each copy of the FAT if the values are identical. If the values are different, the ROM code uses the value from the last FAT copy. With the FAT32 file system, the copy system can be disabled according to a flag in BPB_ExtFlags[7]. If this flag is set, the FAT BPB_ExtFlags[3:0] bit field is used. In this case, no verification is made by the ROM code with other copies of FAT.

The FAT is a simple array of values, each referring to a cluster in the data area. One entry of the array is 12, 16, or 32 bits, depending on the file system in use. The value in an entry defines whether the cluster is being used or not, and if another cluster must be considered. This creates a singly-linked chain of clusters defining the file. Table 32-46 describes the meaning of an entry.

Note

For compatibility, cluster 0 and cluster 1 are not used for files, and these entries must contain:

- FF8h and FFFh (for FAT12)
- FFF8h and FFFFh (for FAT16)
- 0FFFFFF8h and 0FFFFFFFh (for FAT32)

Table 32-46. FAT Entry Description

FAT12	FAT16	FAT32	Description
000h	0000h	00000000h	Free cluster
001h	0001h	00000001h	Reserved cluster
002h–FEFh	0002h–FFEFh	00000002h–0FFFFFEFh	Used cluster; value points to next cluster
FF0h–FF6h	FFF0h–FFF6h	0FFFFFF0h–0FFFFFF6h	Reserved values
FF7h	FFF7h	0FFFFFF7h	Bad cluster
FF8h–FFFh	FFF8h–FFFFh	0FFFFFF8h–0FFFFFFFh	Last cluster in file

Note

FAT32 uses only bits [27:0]; the upper 4 bits are usually 0 and must remain untouched.

When accessing the root directory for FAT32, the ROM code starts from the root directory cluster entry and follows the linked chain to retrieve the clusters.

When the booting file has been found, the ROM code buffers each FAT entry corresponding to the file in a sector way. This means each cluster is translated to one or several sectors, depending on how many sectors are in a cluster (BPB_SecPerClus). This buffer is used later by the booting procedure to access the file.

32.3.7.7 SATA Device Boot Operation

The SATA is a host interface dedicated for mass storage memory devices. SATA aims to connect SATA disk drive compliant SSD (SATA solid state drive) or HDD (hard disk drive) for mass storage use case only.

The mass storage device (SSD or HDD) is intended to be connected through SATA standard connector (in case of HDD) or directly soldered on the board (in case of SSD). In the latter case, the SATA mass-storage device is actually permanently attached to the device.

The device embedded SATA host controller has a single port (P0) implementation, and one internal electrical transceiver (SATA_PHY). For more information on the device-embedded SATA (SATA AHCI core and wrapper), see Section 24.8, *SATA Controller*. For more details on SATA subsystem integrated SATA PHY transceiver components, see Section 26.1, *SATA PHY Subsystem*. A system integration block diagram is provided in following section.

32.3.7.7.1 SATA Booting Overview

The device attached to the device SATA for boot operation must meet the following requirements:

- It must be compliant with the *Serial ATA Standard* specification, rev. 2.6.
- It must support READ SECTOR(s) ATA command (code 0x20) as defined in the ATA/ATAPI - 7 specification.
- It must be connected directly to the device SATA IOs – sata_tx/sata_ty, sata_rx/sata_ry (no port multiplier connected).
- It must have been already powered by the companion chip or external supply before start of the boot procedure.
- It must be taken in consideration for the device that maximum power on ready time allowed by ROM Code is 900ms. First COMRESET procedure targeting the attached device is triggered 300ms after device SATA subsystem initialization by ROM Code.

Figure 32-30 highlights the booting of the platform from the SSD device.

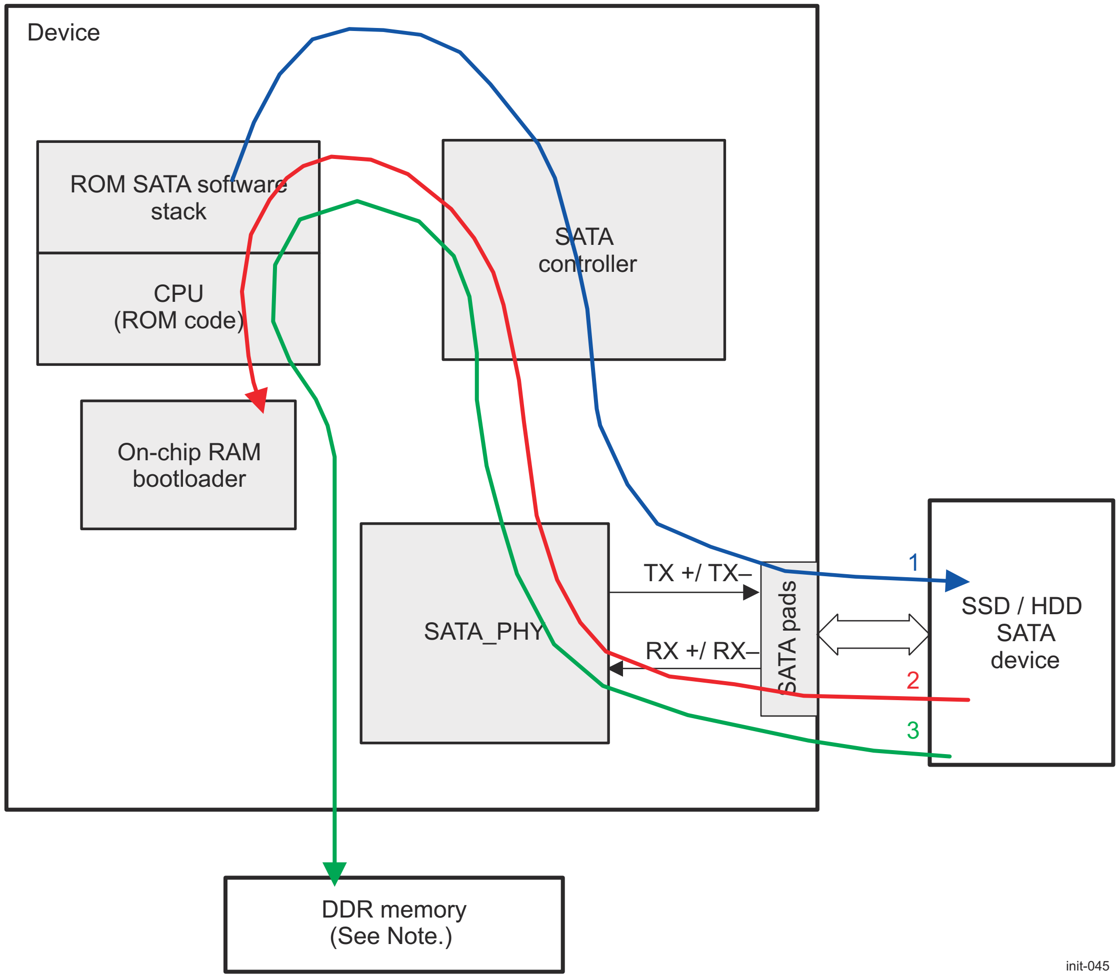


Figure 32-30. Booting from a Permanently-Attached SSD Device

The device booting from an attached SATA mass storage memory device is performed in the following steps:

1. At reset or power on, the device ROM code boots and checks for the SSD/HDD to be ready.
2. The SSD/HDD transfers the boot-loader to the internal RAM of device.
3. The OS in the SSD or HDD is downloaded into DDR memory and the platform is ready.

32.3.7.7.2 SATA Power-Up Initialization Sequence

Figure 32-31 shows all messages exchanges between power-on event up to when the platform is ready. The sequencing aims to depict the complete sequence without any specific connecting issues that may appear.

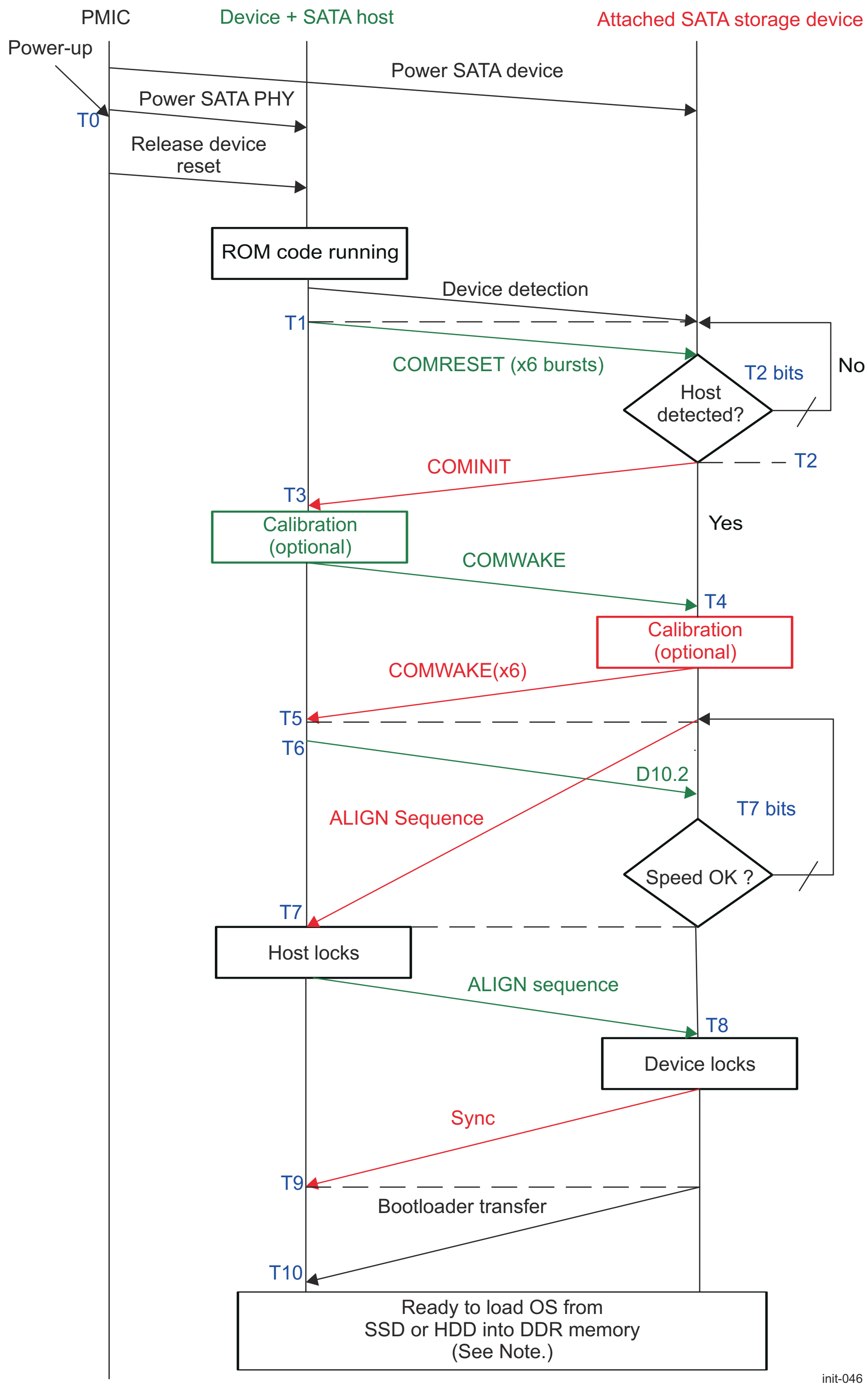


Figure 32-31. SATA Power-on Initialization Sequencing

The following time phases can be identified during booting from SATA interface:

- T0: the SATA_PHY is powered and device reset is maintained until all Power-management IC companion powers are set.
- T1: the ROM code validates a SATA SSD device detection.

- Within the T1-T2 time interval: the SATA host issues a COMRESET sequence for a minimum of 6 bursts (and a multiple of 6) to force a hardware reset to the SATA peripheral device.
- T2: As long as the attached SATA SSD device does not explicit reset request (COMINIT), the host issues COMRESET sequences
- Within the T2-T3 time interval, once the host release the COMRESET sequence (made of 6 bursts minimum), the device responds with a COMINIT to request a communication initialization (It requests a reset from the host).
- Within the T3-T4 time interval, the SATA host controller may calibrate at T3, but issues a COMWAKE sequence to the peripheral in order to inform the other part the wish to use the link.
- Within the T4-T5 time interval, the SSD device responds and may calibrate. The device response is made of 6 burst COMWAKE sequence.
- Within the T5-T6 time interval, SATA host controller shall start transmitting D10.2 characters no later than a defined moment.
- Within the T6-T7 time interval, when the device host detects the COMWAKE from the device, it starts transmitting D10.2 character at its lowest speed.
- At the same time within T5-T7 time interval, continuous stream of device ALIGN sequence (following the 6 bursts COMWAKE sequence) starting at the device highest speed.
- T7: Without any SATA host's answer (D10.2 character), the device ALIGN sequence is repeated for as many slower speeds as are supported. When host receives ALIGN sequence, it locks.
- in case no proposed speeds are supported by the host, the device enters in error state.
- in case no ALIGNp sequence is received by the host within a defined time gap after detecting the release of the device COMWAKE, the host restarts the power on sequence indefinitely until stop by user intervention.
- Within the T7-T8 time interval, as soon as the device SATA host locks, it issues an ALIGN sequence to the attached SATA SSD device that also locks
- Within the T8-T9 time interval, the SATA peripheral device sends a SYNC primitive to inform the communication link is established.
- Within the T9-T10 time interval, once the link has been set successfully, the boot loader is transferred from the SATA peripheral device to device internal on-chip RAM

After passing through all above described phases, the device is ready to load OS source from SSD to the device DDR RAM.

32.3.7.7.3 System Conditions and Limitations for SATA Boot

The following system conditions and limitations are defined by ROM Code implementation:

- The ROM Code sets Gen2 speed (3 Gbps) for a SATA boot operation.
- The ROM Code expects that the SATA I/Os are connected to device SATA subsystem at system reset (i.e. a permanently attached boot device).
- The ROM Code expects that the SATA device is powered externally and supplies are set and stable upon entering the booting procedure. The ROM Code does not perform any software action to the companion device with respect to powering the SATA device.
- The ROM Code allows the SATA device 300ms power on ready time before starting the device detection. In case that the detection fails, the ROM Code retries 3 times with 200ms delay before returning and reporting a SATA boot failure.
- The ROM Code does not handle or enable advanced SATA power modes (slumber and partial).

The ROM Code initializes SATA subsystem after SATA boot is started, as defined in this chapter. The ROM Code does not close SATA connection after it was initialized following procedure defined in Section 32.3.4.5, *Booting Device List Setup*.

32.3.7.7.4 SATA Read Sector Procedure in FAT Mode

The booting medium may hold a FAT file system which the ROM Code is able to read and process. The image used by the booting procedure is taken from a specific booting file named "HLO". This file has to be located in the root directory on an active primary partition of FAT12/16 or FAT32 type.

A SATA drive can be configured either as floppy-like or hard-drive-like.

- When acting as floppy-like, the content of the card is a single file system without any **Master Boot Record (MBR)** holding a partition table.
- When acting as hard-drive-like, a **MBR** is present in the first sector of the card. This **MBR** holds a table of partitions, one of which must be **FAT12/16/32, primary and active**.

Refer to the Section 32.3.7.6.5.1, *MBR and FAT File System*, for a more detailed description of FAT file system support.

32.3.8 Image Format

32.3.8.1 Overview

An image has two major parts:

- An optional CH
- Software to execute

The CH can contain several parameters set by users to speed up booting. It is further described in Section 32.3.8.2, *Configuration Header*.

The second part contains the software that is loaded into the memory and executed.

Figure 32-32 is an overview of the boot image formats. There are two image types:

1. Non-XIP memory booting: This image type is used for memories that require shadowing (for example, NAND). An image for non-XIP memory may not contain a CH and start straight from the GP header. Next, there must be a small header (referred to as a GP header) that contains information about the size and the destination address.
2. When the memory device is of XIP type (for example, NOR), the GP header is not required, and the image can contain code for direct execution. Optionally, the first sector can contain a CH. The same image format is used for peripheral booting (where the code is transferred to internal RAM).

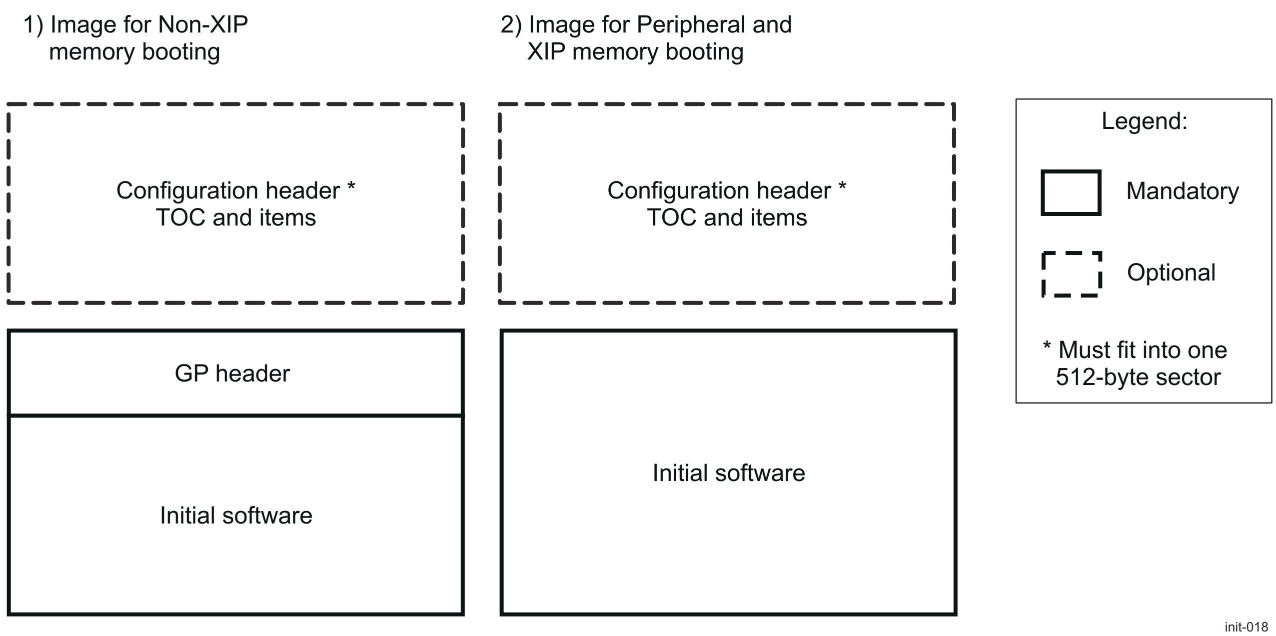


Figure 32-32. Image Formats

32.3.8.2 Configuration Header

The ROM code default settings (such as clock frequencies, EMIF, GPMC, MMCHS, or QSPI interfaces) can be tuned by the user by using the CH.

The CH can contain the following parts:

- Settings: Various clock settings (mandatory)
- FLASH: Flash interface (GPMC) settings
- eMMC/SD: MMC2/MMC1 interface settings
- QSPI: QSPI interface settings

The beginning of the CH is a table of contents (TOC), which points to each item. This is described in Figure 32-33. Each TOC item is a simple structure described in Table 32-47. The complete CH (CH TOC and items) should fit in a 512-byte sector.

The ROM code identifies the presence of a CH by reading the first TOC item if it contains a known string (CHSETTINGS, CHFLASH, etc.). Next, the TOC is identified and searched until a 0xFFFF FFFF offset is found. The CH is read and parameters are executed sequentially.

For the sake of simplicity, each field represents the content of a register to be modified. Only fields required for the configuration are used; fields for status, for instance, are not modified and therefore are not shown in the tables.

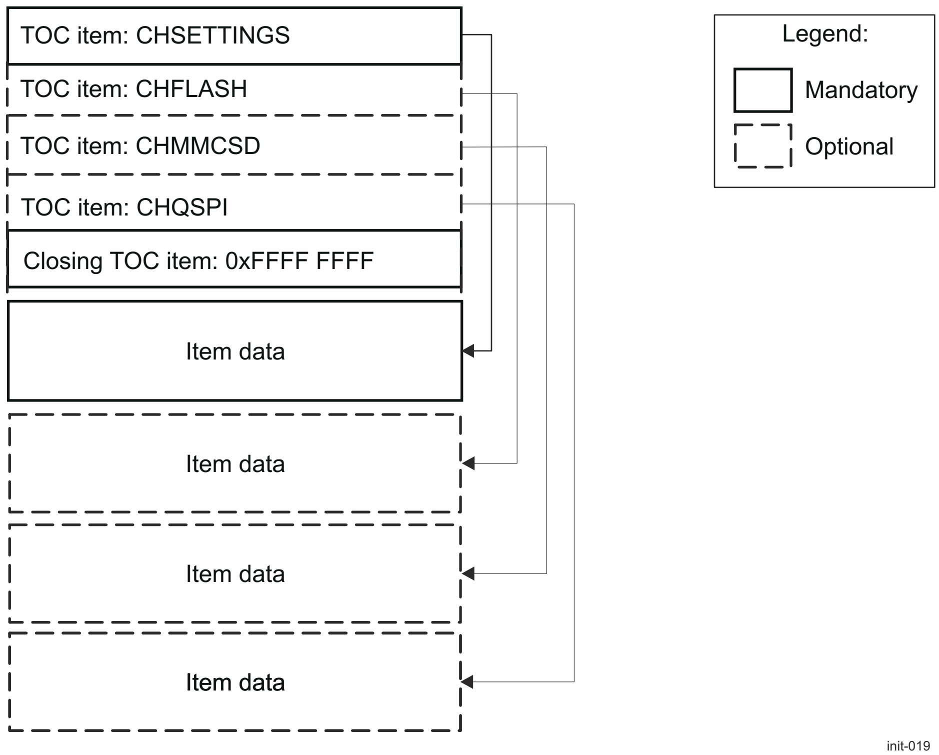


Figure 32-33. CH Format

Table 32-47. CH TOC Item

Offset	Field	Size (Bytes)	Description
0x0000	Start	4	Offset from the start address of the TOC to the actual address of item contents
0x0004	Size	4	Size of item
0x0008	Reserved	4	Unused
0x000C	Reserved	4	Unused
0x0010	Reserved	4	Unused
0x0014	TOC Filename	12	12-character name of a TOC item, including the NULL termination character. That is, this is an array where first byte accomodates the first character.

The ROM Code recognizes sections pointed to by the TOC based on the filename as described in Table 32-48

- The ‘X-LOADER’, ‘2ND', ‘MLO’, ‘ULO’, or ‘HLO’ section contains the Initial Software
- Optionally, TOC may contain CH sections. The 'CHSETTINGS' is a mandatory section of CH and is used to recognize CH presence.

Table 32-48. TOC Filenames

Filename	Item Type	Usage	Description
MLO	Initial Software	eMMC/SD Memory Booting	“Mmc LOader”
HLO	Initial Software	SATA Memory Booting	“Hdd LOader”
ULO	Initial Software	USB or UART Peripheral Booting	“Usb LOader”
2ND	Initial Software	USB or UART Peripheral Booting	“Secondary Loader”
X-LOADER	Initial Software	Other devices	“eXternal LOADER”
CHSETTINGS	Configuration Header	Memory and Peripheral Booting	Configuration Header General Setting Item
CHFLASH	Configuration Header	Memory and Peripheral Booting	Configuration Header GPMC Item
CHMMCSD	Configuration Header	Memory and Peripheral Booting	Configuration Header eMMC/SD Item
CHQSPI	Configuration Header	Memory and Peripheral Booting	Configuration Header QSPI Item

32.3.8.2.1 CHSETTINGS Item

The CHSETTINGS configuration header contains settings specific to the clock system. The ROM code configures the device clocking to some default settings as described in Section 32.3.4.4, *Clocking Configuration*. The CH CHSETTINGS section contains a method to override the ROM code default clock settings.

Table 32-49 describes the fields.

Table 32-49. CHSETTINGS Item

Offset	Field	Description
0000h	Section key	Key used for item verification: C0C0C0C1h
0004h	Valid	Enables/disables the section: 00h: Disable Others: Enable
0005h	Version	Configuration header version 01h Others: Reserved
0006h	Reserved	
0008h	Clocking settings	See Table 32-50.

Table 32-50. Clocking Settings

Field	Size (Bytes)	Description
Flags		
Flags	4	Bit mask of various switches, active when set to 1: Bit [0]: Clock configuration defined in this structure is applied. Bit [1]: Reserved Bit [2]: Apply general clock settings. Bit [3]: Set and lock DPLL_PER. Bit [4]: Set and lock DPLL_MPU. Bit [5]: Set and lock DPLL_CORE. Bit [6]: Set and lock DPLL_USB (USB HS DPLL). Bit [7]: Bypass DPLL_PER before setting clocks. Bit [8]: Bypass DPLL_MPU before setting clocks. Bit [9]: Bypass DPLL_CORE before setting clocks. Bit [10]: Bypass DPLL_USB (USB HS DPLL) before setting clocks. Others: Reserved
General Clock Settings		
CM_CLKSEL_CORE	4	Register value

Table 32-50. Clocking Settings (continued)

Field	Size (Bytes)	Description
CM_BYPCLK_DPLL_MPU	4	Register value
CM_BYPCLK_DPLL_IVA	4	Register value
CM_MPU_MPU_CLKCTRL	4	Register value
CM_CLKSEL_USB_60MHZ	4	Register value
MPU DPLL Settings		
CM_CLKMODE_DPLL_MPU	4	Register value
CM_AUTOIDLE_DPLL_MPU	4	Register value
CM_CLKSEL_DPLL_MPU	4	Register value
CM_DIV_M2_DPLL_MPU	4	Register value
Core DPLL Settings		
CM_CLKMODE_DPLL_CORE	4	Register value
CM_AUTOIDLE_DPLL_CORE	4	Register value
CM_CLKSEL_DPLL_CORE	4	Register value
CM_DIV_M2_DPLL_CORE	4	Register value
CM_DIV_M3_DPLL_CORE	4	Register value
CM_DIV_H11_DPLL_CORE	4	Register value
CM_DIV_H12_DPLL_CORE	4	Register value
CM_DIV_H13_DPLL_CORE	4	Register value
CM_DIV_H14_DPLL_CORE	4	Register value
CM_DIV_H21_DPLL_CORE	4	Register value
CM_DIV_H22_DPLL_CORE	4	Register value
CM_DIV_H23_DPLL_CORE	4	Register value
CM_DIV_H24_DPLL_CORE	4	Register value
PER DPLL Settings		
CM_CLKMODE_DPLL_PER	4	Register value
CM_AUTOIDLE_DPLL_PER	4	Register value
CM_CLKSEL_DPLL_PER	4	Register value
CM_DIV_M2_DPLL_PER	4	Register value
CM_DIV_M3_DPLL_PER	4	Register value
CM_DIV_H11_DPLL_PER	4	Register value
CM_DIV_H12_DPLL_PER	4	Register value
CM_DIV_H13_DPLL_PER	4	Register value
CM_DIV_H14_DPLL_PER	4	Register value
USB DPLL Settings		
CM_CLKMODE_DPLL_USB	4	Register value
CM_AUTOIDLE_DPLL_USB	4	Register value
CM_CLKSEL_DPLL_USB	4	Register value
CM_DIV_M2_DPLL_USB	4	Register value

32.3.8.2.2 CHFLASH Item

The CHFLASH configuration header contains settings specific to the general-purpose memory controller (GPMC). For more information, see Section 15.4, *General-Purpose Memory Controller*. Table 32-51 describes the fields.

Table 32-51. CHFLASH Item

Offset	Field	Description
0000h	Section Key	Key used for section verification: C0C0C0C3h.
0004h	Valid	Enables/disables the section: 00h: Disable Others: Enable
0005h	Reserved	
0008h	GPMC_SYSCONFIG (LSW)	Register values
000Ah	GPMC_IRQENABLE (LSW)	
000Ch	GPMC_TIMEOUT_CONTROL (LSW)	
000Eh	GPMC_CONFIG (LSW)	
0010h	GPMC_CONFIG1_0	
0014h	GPMC_CONFIG2_0	
0018h	GPMC_CONFIG3_0	
001Ch	GPMC_CONFIG4_0	
0020h	GPMC_CONFIG5_0	
0024h	GPMC_CONFIG6_0	
0028h	GPMC_CONFIG7_0	
002Ch	GPMC_PREFETCH_CONFIG1	
0030h	GPMC_PREFETCH_CONFIG2 (LSW)	
0032h	GPMC_PREFETCH_CONTROL (LSW)	
0034h	GPMC_ECC_CONFIG (LSW)	
0036h	GPMC_ECC_CONTROL (LSW)	
0038h	GPMC_ECC_SIZE_CONFIG	
003Ch	Reserved	

32.3.8.2.3 CHMMCSD Item

The CHMMCSD configuration header contains settings specific to the high-speed MMC/SD/SDIO host controller (MMCHS). For more information, see Chapter 25, *eMMC/SD/SDIO*. Table 32-52 describes the fields.

Table 32-52. CHMMCSD Item

Offset	Field	Description
0000h	Section key	Key used for section verification C0C0C0C4h
0004h	Valid	Enables/disables the section: 00h: Disable Other: Enable
0005h	Reserved	
0008h	CLKD	Functional clock divisor MMCHS_SYSCTL[15:6] CLKD = 0x0: FCLK/1 0x1: FCLK/1 0x2: FCLK/2 … 0x3FF: FCLK/1023 0xFFFF FFFF: Do not modify clock divisor

Table 32-52. CHMMCSD Item (continued)

Offset	Field	Description
000Ch	MMCHS interface bus width	Configure the MMCHS interface bus width according to the field value : 1: Configured to 1 bit (SDR) 2: Configured to 4 bits (SDR) 4: Configured to 8 bits (SDR)(1) 8: Configured to 4 bits (DDR)(1) 16: Configured to 8 bits (DDR)(1) 0xFFFF FFFF: Do not update bus width. Others: Reserved

(1) The 8 bit SDR and 4-bit/8-bit DDR (values 4, 8, and 16) are only applicable to eMMC devices.

The ROM code provides a booting parameter structure to the initial software (see Section 32.3.8.4, *Image Execution*). This structure contains a field that indicates whether the configuration header items have been correctly processed. For a CHMMCSD item, if the MMCHS_SYSCTL and bus width fields are set to 0xFFFF FFFF, the booting parameters report that the CHMMCSD section has not been executed, regardless of the value of the Valid field.

32.3.8.2.4 CHQSPI Item

The CHQSPI configuration header contains settings specific to the QSPI interface controller. For more information, see Section 24.5, *Quad Serial Peripherial Interface*. Table 32-53 describes the fields.

Table 32-53. CHQSPI Item

Offset	Register Field Modified	Description
0000h	Section key	Key used for section verification C0C0C0C6h
0004h	Valid	Enables/disables the section: 00h: Disable Other: Enable
0005h	Reserved	
0008h	SPI Clock	SPI clock rate: 0x03: 32 MHz 0x07: 16 MHz 0x13: 48 MHz 0x17: 24 MHz 0x1F: 12 MHz All other: reserved
0009h	RCMD	Read command
000Ah	READ_TYPE	Determines if the read command is a single, dual or quad read mode command
000Bh	NUM_A_BYTES	Number of address bytes to be sent
000Ch	NUM_D_BYTES	Number of dummy bytes to be used for fast read

32.3.8.3 GP Header

When the booting memory device is non-XIP (for example, NAND) the image must contain a small header, located before the executable code, and having the size of the software to load and the destination address of where to store it. Table 32-54 describes the image format. The GP header is not required when booting from an XIP memory device (for example, NOR) or in case of peripheral booting. In this case, the peripheral or memory booting image starts directly with executable code.

Table 32-54. GP Header Image Format

Field	Non-XIP Device Offset	XIP Device Offset	Size (Bytes)	Description
Size	0x0000	–	4	Size of the image (including GP header)
Destination	0x0004	–	4	Address where to store the code or code entry point
Image Code	0x0008	0x0000	x	Executable code

Note

The Destination address field stands for:

- Target address for the image copy from the non-XIP storage to the target XIP location (for example, internal RAM or SDRAM)
- Entry point for image code

Users must take care to locate the code entry point to the target address for image copy.

32.3.8.4 Image Execution

The image is executed when the ROM code performs the branch to the first executable instruction in the initial software. In non-XIP, the execution address is the first word after the GP header. The branch is performed in public Arm supervisor mode. The R0 register points to the booting parameter structure that contains information about booting execution. Table 32-55 shows the booting parameter structure.

Table 32-55. Booting Parameter Structure

Offset	Field	Size (Bytes)	Description
0x00	Booting message	4	Last received booting message
0x04	Memory booting device descriptor	4	Pointer to the memory device descriptor that has been used during the memory booting process
0x08	Current booting device	1	Code of device used for booting: 0x01: XIP 0x02: XIP (with wait monitoring) 0x03: NAND 0x05: SD cards 0x06: eMMC (boot partition) 0x07: eMMC 0x09: SATA 0x0A: QSPI_1 0x0B: QSPI_4 0x43: UART 0x45: USB Others: Reserved
0x09	Reset reason	1	Current reset reason bit mask (bit = 1, event present): direct copy from lower byte of PRM_RSTST (more bits exist in PRM_RSTST): [0]: Power-on (cold) reset [1]: Global software warm reset [2]: Reserved [3]: MPU watchdog reset [4]: Reserved [5]: External warm reset [6]: VDD_MPU voltage manager reset

Table 32-55. Booting Parameter Structure (continued)

Offset	Field	Size (Bytes)	Description
			[7]: VDD_MM voltage manager reset
0x0A	CH flags	1	Configuration header items flag. Each item is described by 1 bit. A set bit indicates that the item was executed: [0]: CHSETTINGS [2]: CHFLASH [3]: CHMMCSD [4]: CHQSPI Other bits: Reserved

32.3.9 Tracing

Tracing in the public ROM code consists in 32-bit vectors for which each bit corresponds to a particular way point in the ROM code execution sequence. Table 32-56 through Table 32-59 list the organization of the tracing data in RAM. Tracing vectors are initialized at the beginning of the start-up phase and are updated all along the boot process.

There are two sets of tracing vectors (Table 32-14, *Tracing Data)*. The first set is the current trace information (after a cold or warm reset). The second set holds a copy of trace vectors collected at the first ROM code run after a cold reset. As a result, after a warm reset it is possible to have visibility on the boot scenario that occurred during a cold reset.

Table 32-56 lists the organization of tracing vector 1.

Table 32-56. Tracing Vector 1

Bit	Group	Meaning
0	Boot	Passed the public reset vector
1	Boot	Entered main function
2	Boot	Running after the cold reset
3	Boot	Main booting routine entered
4	Memory boot	Memory booting started
5	Peripheral boot	Peripheral booting started
6	Boot	Booting loop reached last device
7	Boot	GP header found
8	Peripheral Boot	Booting message Skip peripheral booting received
9	Peripheral Boot	Booting message Change device received
10	Peripheral boot	Booting message Peripheral booting received
11	Peripheral boot	Booting message Get ASIC ID received
12	Peripheral boot	Device initialized
13	Peripheral boot	ASIC ID sent
14	Peripheral boot	Image received
15	Peripheral boot	Peripheral booting failed
16	Peripheral boot	Booting message not received (time-out)
17	Peripheral boot	Image size not received (time-out)
18	Peripheral boot	Image not received (time-out)
19	Reserved	
20	Boot	Configuration header found
21	Boot	CHSETTINGS item processed
22	Boot	Reserved
23	Boot	CHFLASH item processed
24	Boot	CHMMCSD item processed (clock)
25	Boot	CHMMCSD item processed (bus width)
26	Boot	CHMMCSD item processed (eMMC DDR mode)
27	Reserved	
28	Boot	SWCFG general detected
29	Boot	SWCFG clocks detected
30	Boot	SWCFG time-out detected
31	Reserved	

Table 32-57 lists the organization of tracing vector 2.

Table 32-57. Tracing Vector 2

Bit	Group	Meaning
0	Configuration Header	CHSATA item processed
1:3	Reserved	
4	USB	USB connected
5	USB	USB configured
6:7	Reserved	
8	Configuration Header	CHQSPI item processed (clock speed)
9	Configuration Header	CHQSPI item processed (Read command)
10:11	Reserved	Reserved
12	Memory boot	Memory booting trial (first block)
13	Memory boot	Memory booting trial (second block)
14	Memory boot	Memory booting trial (third block)
15	Memory boot	Memory booting trial (fourth block)
16:27	Reserved	
28:29	Reserved	
30	Boot	Jumping to Initial Software
31	Reserved	

Table 32-58 lists the organization of tracing vector 3.

Table 32-58. Tracing Vector 3

Bit	Group	Meaning
0	Reserved	
1	Memory boot	Memory booting device XIP
2	Memory boot	Memory booting device XIPWAIT
3	Memory boot	Memory booting device NAND
4	Reserved	
5	Memory boot	Memory booting device SD
6	Memory boot	Memory booting device: eMMC (from boot partition)
7	Memory boot	Memory booting device MMC2: eMMC (from user area)
8	Reserved	
9	Memory boot	Memory booting device: SATA
10	Memory boot	Memory booting device: QSPI_1
11	Memory boot	Memory booting device: QSPI_4
12:15	Reserved	
16:18	Reserved	
19	Peripheral boot	Peripheral booting device UART3
20	Reserved	
21	Peripheral boot	Peripheral booting device USB
22:23	Reserved	
24:27	Reserved	
28	Boot	Reserved
29	Boot	Reserved
30:31	Reserved	

Table 32-59 lists the organization of tracing vector 4.

Table 32-59. Tracing Vector 4

Bit	Group	Meaning
0:23	Reserved	
24	MMC/SD	SD card detected PBIAS configuration is 1.8V
25:27	Reserved	
28	SATA	SATA is configured
29	SATA	SATA retried
30	SATA	SATA failed
31	Reserved	

32.4 Services for HLOS Support

The ROM code provides different services that can be called for L1 and L2 cache maintenance, Enter in Low Power, etc. These services are implemented in monitor mode and must be called by using the SMC instruction. The caller must ensure the save and restore of the processor registers before and after calling the Monitor Service.

The following code example shows how the monitor ROM code functions can be accessed by an application running in public mode:

```
;-------------------------------------------------------------------------------
; FUNCTION: SetL2CacheLatency
;
; DESCRIPTION: Function calls the Monitor Service to setup the L2 Cache Latency
;
; INPUTS:  r0 Tag RAM Latency to set
;          r1 Data RAM Latency to set
; RETURN:
;
;-------------------------------------------------------------------------------
SetL2CacheLatency           FUNCTION

  PUSH {R1-R12, LR}
  LDR R12, =0x105
  SMC 0x1
  POP {R1-R12, PC}
  ENDFUNC
```

32.4.1 Hypervisor

Table 32-60. Start Hypervisor

Function ID	Description	
R12 = 0x102	This function starts the CPU Hypervisor mode. Start address of the Hypervisor software must be given in R0.	
Parameters		
Type	**Location**	**Description**
Input	R0	Start address of the Hypervisor software
Output		
Return		

32.4.2 Caches Maintenance

Table 32-61. Clean L1 and/or L2 cache

Function ID	Description	
R12 = 0x103	This function cleans and invalidates the CPU L1-cache and if requested in the input parameter R0, clean the full L2-cache.	
Parameters		
Type	**Location**	**Description**
Input	R0	If not equal to 0, clean L2-cache
Output		
Return		

32.4.3 CP15 Registers

Table 32-62. Write L2 Cache Auxiliary Control

Function ID	Description
R12 = 0x104	This function writes the CP15 L2 Cache Auxiliary Control Register with the input given value in R0.
Parameters	

Table 32-62. Write L2 Cache Auxiliary Control (continued)

Function ID	Description	
Type	**Location**	**Description**
Input	R0	Value to set in the register
Output		
Return		

Table 32-63. Write Tag and Data RAM Latency Control Register

Function ID	Description	
R12 = 0x105	This function writes the L2 Cache Tag and Data RAM Latency in the CP15 L2 Control Register with the input given value in R0 and R1.	
Parameters		
Type	**Location**	**Description**
Input	R0 R1	Tag RAM Latency to set Data RAM Latency to set
Output		
Return		

Table 32-64. Write L2 Cache Prefetch Control Register

Function ID	Description	
R12 = 0x106	This function writes the CP15 L2 Cache Prefetch Control Register with the input given value in R0.	
Parameters		
Type	**Location**	**Description**
Input	R0	Value to set in the register
Output		
Return		

Table 32-65. Write Auxiliary Control Register

Function ID	Description	
R12 = 0x107	This function writes the CP15 Auxiliary Control Register with the input given value in R0.	
Parameters		
Type	**Location**	**Description**
Input	R0	Value to set in the register
Output		
Return		

32.4.4 Wakeup Generator

Table 32-66. Write AMBA IF Register

Function ID	Description	
R12 = 0x108	This function writes the WakeupGen AMBA I/F Register with the input given value in R0.	
Parameters		
Type	**Location**	**Description**
Input	R0	Value to set in the register
Output		
Return		

32.4.5 Arm Timer

Table 32-67. Set Timer CNTFRQ Register

Function ID	Description	
R12 = 0x109	This function sets the Arm Timer CNTFRQ (CP15 register) with the input given value in R0.	
Parameters		
Type	**Location**	**Description**
Input	R0	Value to set in the register
Output		
Return		

Chapter 33

On-Chip Debug Support

This chapter describes the on-chip debug support.

Note

The L3_MAIN interconnect is instantiation of the NoC interconnect from Arteris, Inc. Arteris is a registered trademark of Arteris, Inc.

This document contains materials that are ©Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris Reference manuals, or contact Arteris, Inc.

NoC is an abbreviation for Network On Chip.

33.1 Introduction 7024
33.2 Debug Interfaces 7028
33.3 Debugger Connection 7031
33.4 Primary Debug Support 7035
33.5 Real-Time Debug 7040
33.6 Power, Reset, and Clock Management Debug Support 7041
33.7 Performance Monitoring 7044
33.8 MPU Memory Adaptor (MPU_MA) Watchpoint 7048
33.9 Processor Trace 7049
33.10 System Instrumentation 7053
33.11 Concurrent Debug Modes 7076
33.12 DRM Register Manual 7077

33.1 Introduction

Debugging a system that contains an embedded processor involves an environment that connects high-level debugging software running on a host computer to a low-level debug interface supported by the target device. Between these levels, a debug and trace controller (DTC) facilitates communication between the host debugger and the debug support logic on the target chip.

The DTC is a combination of hardware and software that connects the host debugger to the target system. The DTC uses one or more hardware interfaces and/or protocols to convert actions dictated by the debugger user to JTAG® commands and scans that execute the core hardware.

The debug software and hardware components let the user control multiple central processing unit (CPU) cores embedded in the device in a global or local manner. This environment provides:

- Synchronized global starting and stopping of multiple processors
- Starting and stopping of an individual processor
- Each processor can generate triggers that can be used to alter the execution flow of other processors

System topics include but are not limited to:

- System clocking and power-down issues
- Interconnection of multiple devices
- Trigger channels

For easy integration into applications, a set of application-programming interfaces (APIs) for debug-IP programming and a software message library are provided. CToolsLib is a library of embedded target APIs to enable easy programmatic access to the chip tools (CTools), which are system-level debug facilities included in the debug subsystem capabilities of TI devices. More information about the APIs, download files, and other useful links for available libraries can be found on the CToolsLib Wiki site:

http://processors.wiki.ti.com/index.php/CToolsLib

The previous link connects to TI community resources. Linked contents are provided “AS IS” by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

Note

TI also provides a Register Descriptor Tool (RDT). The RDT is a Java™- based stand-alone application. It is an interactive device register database configuration software. It allows the user to:

- Visualize device register settings
- Manipulate device registers easily
- Read in complete device configurations from various file formats into registers easily
- Dump out complete device configurations to various file formats
- Trace the register changes in time and during debugging
- See, understand, and work with device IPs, Instances, Registers, Bitfields, descriptions, and so on
- Extract and read in registers and register dumps for Code Composer Studio, and Lauterbach

The advantage of the tool is that the user can visualize the register state on power-on reset and then customize the configuration of the device for the specific use-case and identify the device register settings associated to that configuration.

Being an interactive visual tool, the RDT gives the user a global view of the device setting architecture and allows determining the exact register settings to obtain the specific configuration.

Figure 33-1 is a screenshot diagram of the RDT.

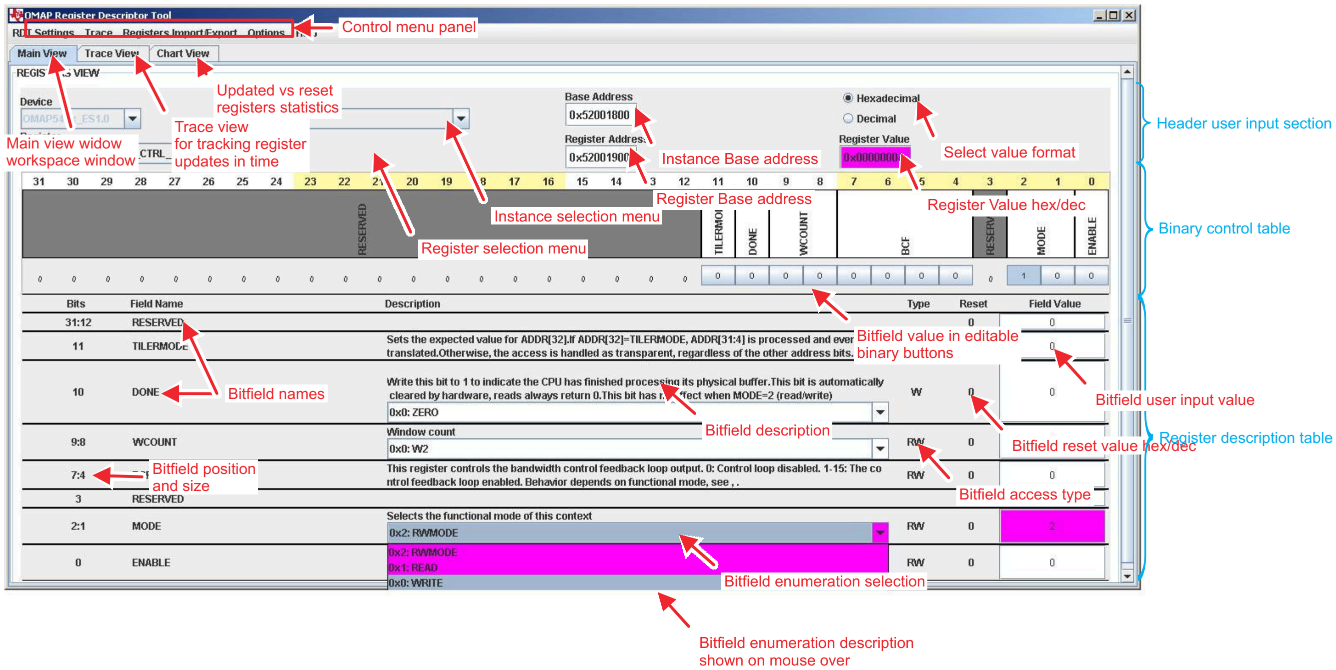


Figure 33-1. Register Descriptor Tool (RDT)

33.1.1 Key Features

The device deploys Texas Instrument's CTools debug technology for on-chip debug and trace support. It provides the following features:

- External debug interfaces:
  - Primary debug interface - IEEE1149.1 (JTAG)
    - Used for debugger connection
    - Controls ICEPick™ (generic test access port [TAP] for dynamic TAP insertion) to allow the debugger to access several debug resources through its secondary (output) JTAG ports (for more information, see Section 33.3.3.1, *ICEPick Secondary TAPs*).
    - For more information about IEEE1149.1, see Section 33.2.1, *IEEE1149.1*.
  - Debug (trace) port
    - Can be used to export processor or system trace off-chip (to an external trace receiver)
    - Can be used for cross-triggering with an external device
    - Configured through debug resources manager (DRM) module instantiated in the debug subsystem
    - For more information about debug (trace) port, see Section 33.2.2, *Debug (Trace) Port* , and Section 33.11, *Concurrent Debug Modes*.
- JTAG based processor debug on:
  - Cortex-A15 (x2) in MPU
  - C66x in DSP1, DSP2
  - Cortex-M4 (x2) in IPU1, IPU2
  - Arm968 (x2) in IVA
  - ARP32 in EVE1, EVE2
- Dynamic TAP insertion
  - Controlled by ICEPick
  - For more information, see Section 33.3.3, *Dynamic TAP Insertion.*
- Power and clock management
  - Debugger can get the status of the power domain associated to each TAP.
  - Debugger may prevent the application software switching off the power domain.
  - Application power management behavior can be preserved during debug across power transitions.

  - For more information, see Section 33.6.1, *Power and Clock Management.*
- Reset management
  - Debugger can configure ICEPick to assert, block, or extend the reset of a given subsystem.
  - For more information, see Section 33.6.2, *Reset Management.*
- Cross-triggering
  - Provides a way to propagate debug (trigger) events from one processor, subsystem, or module to another:
    - Subsystem A can be programmed to generate a debug event, which can then be exported as a global trigger across the device.
    - Subsystem B can be programmed to be sensitive to the trigger line input and to generate an action on trigger detection.
  - Two global trigger lines are implemented
  - Device-level cross-triggering is handled by the XTRIG (TI cross-trigger) module implemented in the debug subsystem
  - Various Arm® CoreSight™ cross-trigger modules implemented to provide support for CoreSight triggers distribution
    - CoreSight Cross-Trigger Interface (CS_CTI) modules
    - CoreSight Cross-Trigger Matrix (CS_CTM) modules
  - For more information about cross-triggering, see Section 33.4.2, *Cross-Triggering.*
- Suspend
  - Provides a way to stop a closely coupled hardware process running on a peripheral module when the host processor enters debug state
  - For more information about suspend, see Section 33.4.3, *Suspend.*
- MPU watchpoint
  - Embedded in MPU subsystem
  - Provides visibility on MPU to EMIF direct paths
  - For more information, see Section 33.8, *MPU Memory Adaptor (MPU_MA) Watchpoint*
- Processor trace
  - Cortex-A15 (MPU) and C66x (DSP) processor trace is supported
  - Program trace only for MPU (no data trace)
  - MPU trace supported by a CoreSight Program Trace Macrocell (CS_PTM) module (per MPU core)
  - Three exclusive trace sinks:
    - CoreSight Trace Port Interface Unit (CS_TPIU) – trace export to an external trace receiver
    - CTools Trace Buffer Router (CT_TBR) in system bridge mode – trace export through USB
    - CT_TBR in buffer mode – trace history store into on-chip trace buffer
  - For more information, see Section 33.9, *Processor Trace.*
- System instrumentation (trace)
  - Supported by a CTools System Trace Module (CT_STM), implementing MIPI System Trace Protocol (STP) (rev 2.0)
  - Real-time software trace
    - MPU software instrumentation through CoreSight STM (CS_STM) (STP2.0)
    - System-on-chip (SoC) software instrumentation through CT_STM (STP2.0)
  - OCP watchpoint (OCP_WP_NOC)
    - OCP target traffic monitoring: OCP_WP_NOC can be configured to generate a trigger upon watchpoint match (that is, when target transaction attributes match the user-defined attributes).
    - SoC events trace
    - DMA transfer profiling
  - Statistics collector (performance probes)
    - Computes traffic statistics within a user-defined window and periodically reports to the user through the CT_STM interface
    - Embedded in the L3_MAIN interconnect
    - 10 instances:
      - 1 instance dedicated to target (SDRAM) load monitoring
      - 9 instances dedicated to master latency monitoring

  - IVA instrumentation (hardware accelerator [HWA] profiling)
    - Supported through a software message and system trace event (SMSET) module embedded in the IVA subsystem
  - EVE instrumentation (hardware accelerator [HWA] profiling)
    - Supported through a software message and system trace event (SMSET) module embedded in the EVE subsystem
  - Power-management events profiling (PM instrumentation [PMI])
    - Monitoring major power-management events. The PM state changes are handled as generic events and encapsulated in STP messages.
  - Clock-management events profiling (CM instrumentation [CMI]) for CM_CORE_AON clocks
    - Monitoring major clock management events. The CM state changes are handled as generic events and encapsulated in STP messages.
  - For more information, see Section 33.10, *System Instrumentation.*
- Performance monitoring
  - Supported by subsystem counter timer module (SCTM) for IPU
  - Supported by performance monitoring unit (PMU) for MPU subsystem
  - For more information, see Section 33.7, *Performance Monitoring*.

33.2 Debug Interfaces

33.2.1 IEEE1149.1

The target debug interface has the following signals:

- Five standard IEEE1149.1 JTAG signals: nTRST, TCK, TMS, TDI, and TDO
- A return clock (RTCK) due to the clocking requirements of the Arm968™ processor
- Two EMU [1:0] TI extensions

Table 33-1 describes the IEEE1149.1 signals.

Table 33-1. IEEE1149.1 Signals

Device Pad Name	Internal Signal Name	Type(1)	Function	Description
trstn	nTRST	I	Test reset	When asserted (active low), resets all test and debug logic in the device along with the IEEE1149.1 interface.
tclk	TCK	I	Test clock	This is the test clock used to drive an IEEE1149.1 TAP state-machine and logic. This is either a free-running clock or a gated clock, depending on the DTC attached to the device and the RTCK monitoring.
rtck	RTCK	O	Returned (synchronized) test clock	Depending on the DTC attached to the device, the JTAG signals are either clocked from RTCK or the RTCK is monitored by the DTC to the gate TCK.
tms	TMS	I/O	Test mode select input	Directs the next state of the IEEE1149.1 TAP state-machine.
tdi	TDI	I	Test data input	Scans data input to the device.
tdo	TDO	O	Test data output	Scans data output by the device.
emu0	EMU0	I/O	Emulation 0	Channel 0 trigger or boot mode or trace port.
emu1	EMU1	I/O	Emulation 1	Channel 1 trigger or boot mode or trace port.

(1) I = Input; O = Output; I/O = bidirectional

Note

For more information about device pads pull type resistors, see the device Data Manual, Section Signal Descriptions in Chapter Terminal Configuration and Functions.

Note

The device JTAG ID code can be accessed through ICEPick. For information about the JTAG ID code value, see Chapter 1, *Introduction*.

33.2.2 Debug (Trace) Port

On-chip debug and trace events can be exported to external equipment through the debug (trace) port of the device. The following exportable debug events and trace sources are supported:

- Debug events
  - Triggers. For more information about triggers, see Section 33.4.2, *Cross-Triggering*.
- Trace sources
  - Processor trace: Cortex-A15 MPU and C66x DSP traces are supported. For more information about the processor trace, see Section 33.9, *Processor Trace*.
  - System trace: Trace coming from various system instrumentation modules, and supported by the CT_STM module. For more information about the system trace, see Section 33.10, *System Instrumentation*.

Note that not all debug and trace features can be supported concurrently because of the limited number of pins allocated to debug. Thus, multiplexing among debug and trace sources is implemented. The configuration and the debug/trace source selection occur through the DRM module embedded in the debug subsystem.

Table 33-2 describes the trace port signals.

Table 33-2. Trace Port Signals

Pin Name	Internal Signal Name	I/O(1)	Pull Type (2)	Description
emu19	EMU19	O	PD	Emulator pin 19
emu18	EMU18	O	PD	Emulator pin 18
emu17	EMU17	O	PD	Emulator pin 17
emu16	EMU16	O	PD	Emulator pin 16
emu15	EMU15	O	PD	Emulator pin 15
emu14	EMU14	O	PD	Emulator pin 14
emu13	EMU13	O	PD	Emulator pin 13
emu12	EMU12	O	PD	Emulator pin 12
emu11	EMU11	O	PD	Emulator pin 11
emu10	EMU10	O	PD	Emulator pin 10
emu9	EMU9	O	PD	Emulator pin 9
emu8	EMU8	O	PD	Emulator pin 8
emu7	EMU7	O	PD	Emulator pin 7
emu6	EMU6	O	PD	Emulator pin 6
emu5	EMU5	O	PD	Emulator pin 5
emu4	EMU4	O	PD	Emulator pin 4
emu3	EMU3	O	PD	Emulator pin 3
emu2	EMU2	O	PD	Emulator pin 2
emu1	EMU1	I/O	PU	Emulator pin 1
emu0	EMU0	I/O	PU	Emulator pin 0

(1) I = Input; O = Output; I/O = bidirectional
(2) PU = internal pullup; PD = internal pulldown

Note

The emu[19:0] pins are shared with other functional (application) pins on the device boundary. To use the emu[19:0] pins, the user must program the device application pin manager (Control Module) appropriately. For more information, see Chapter 18, *Control Module*.

For more information about DRM multiplexing and concurrent debug modes, see Section 33.11, *Concurrent Debug Modes*.

33.2.3 Trace Connector and Board Layout Considerations

Texas Instruments supports a variety of eXtended Development System (XDS) JTAG controllers with various debug capabilities beyond only JTAG support. Because this device supports the export of processor trace and system trace over the EMU pins, if you want your target to be compatible with XDS products capable of acquiring either trace types, see the following document for guidelines: *Emulation and Trace Headers* (literature number SPRU655). You can also find more information at the “XDS Target Connection Guide” TI wiki page:

http://processors.wiki.ti.com/index.php/XDS_Target_Connection_Guide

The previous link connects to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

33.3 Debugger Connection

33.3.1 ICEPick Module

The debugger connects to the device through its JTAG interface. The first level of debug interface seen by the debugger is the ICEPick module embedded in the debug subsystem.

Note

ICEPick version D (ICEPick-D) is used in the device.

SoC designs typically have multiple processors, each having a JTAG TAP embedded in the processor. The ICEPick module manages these TAPs and the power, reset, and clock controls for modules that have TAPs.

The ICEPick module is visible only from the debugger point of view and thus cannot be programmed by application software. The debugger can configure ICEPick through its own TAP controller. The ICEPick TAP has an instruction length of 6 bits and is the primary TAP. It is always visible in the scan chain and is used to control and monitor the other secondary TAPs.

ICEPick provides the following debug capabilities:

- Debug connect logic for enabling or disabling most ICEPick instructions
- Dynamic TAP insertion
  – Serially linking up to 32 TAP controllers
  – Individually selecting one or more of the TAPs for scan without disrupting the instruction register (IR) state of other TAPs
- Power, reset, and clock management (PRCM)
  – Provides the power and clock states of each domain
  – Provides debugger control of the power domain of a processor. Can force the domain power and clocks on, and prohibit the domain from being clock-gated or powered down while a debugger is connected.
  – Applies system reset
  – Provides wait-in-reset (WIR) boot mode
  – Provides global and local WIR release
  – Provides global and local reset blocking

The ICEPick module implements a connect register, which must be configured with a predefined key to enable the full set of JTAG instructions. Once the debug connect key is properly programmed, ICEPick signals and subsystems emulation logics should be turned on.

For more information about ICEPick dynamic TAP insertion, see Section 33.3.3, *Dynamic TAP Insertion*.

For more information about ICEPick PRCM features, see Section 33.6, *Power, Reset, and Clock Management Debug Support*.

33.3.2 ICEPick Boot Modes

The initial configuration of ICEPick is determined by the level of the EMU0 and EMU1 pins at POR release. At POR, EMU0 and EMU1 are automatically configured as inputs. The EMU0 and EMU1 pins are free when POR is released.

Table 33-3 summarizes the ICEPick boot modes.

Table 33-3. ICEPick Boot Modes at POR

EMU1	EMU0	TAPs in the TDI → TDO Path	Other Effects/Comments
0	0	None	Reserved (do not use)
0	1	None	Reserved (do not use)
1	0	ICEPick	TAP only + WIR mode
1	1	ICEPick	TAP only (default mode)

33.3.2.1 Default Boot Mode

In ICEPick-only configuration, none of the secondary TAPs are selected. The ICEPick TAP is the only TAP between device-level TDI and TDO pins. This mode is the recommended boot mode.

33.3.2.2 Wait-In-Reset

The device can boot to invoke WIR mode. If the device is booted in this mode, all processors within the device that support a TAP through ICEPick are held in reset until released. Individual processors can be released from reset (local), or all processors held in the reset state can be released at the same time (global).

33.3.3 Dynamic TAP Insertion

33.3.3.1 ICEPick Secondary TAPs

To include more or fewer secondary TAPs in the scan chain, the debugger must use the ICEPick TAP router to program the TAPs. At its root, ICEPick is a scan-path linker that lets the DTC selectively choose which subsystem TAPs are accessible through the device-level debug interface. Each secondary TAP can be dynamically included in or excluded from the scan path. From the external JTAG interface point of view, secondary TAPs that are not selected appear not to exist.

Table 33-4 lists the secondary debug TAPs connected to the ICEPick scan chain along with the modules that can be accessed. The TAP number shows the position of the TAP in the scan chain.

Table 33-4. ICEPick Secondary Debug TAPs Mapping

Secondary JTAG Port	CoreSight	TAP Number	Modules Accessed Through That JTAG Port	
Debug Bank				
Reserved	No	0	–	
DSP1	No	1	C66x / ICEMaker	
IVA ICONT1	No	2	Arm968 / ICECrusher™-9	
IVA ICONT2	No	3	Arm968 / ICECrusher-9	
IPU1	No	4	Cortex-M4 / ICECrusher-CS	
	No	5	Cortex-M4 / ICECrusher-CS	
Reserved	No	6-7	–	
IPU2	No	8	Cortex-M4 / ICECrusher-CS	
	No	9	Cortex-M4 / ICECrusher-CS	
DSP2	No	10	C66x / ICEMaker	
Reserved	No	11-14	–	
CS_DAP (APB-AP)	Yes	15	MPU Subsystem	Cortex-A15 (x2)
	Yes			CS_PTM (x2)
	Yes			CS_CTI (x2)
	Yes			CS_STM
	Yes			CS_TF_MPU
	No			DAP_PC
	No			ATB_FIFO_SGU
	No		Debug Subsystem	CT_TBR
	Yes			CS_TF_DEBUGSS
	Yes			CS_TPIU
	No			DRM
	No			CT_STM
	Yes			CS_CTI
CS_DAP (AHB-AP)	No		EVE (x2)	SCTM
	No			SMSET
	No			ARP32

Table 33-4. ICEPick Secondary Debug TAPs Mapping (continued)

Secondary JTAG Port	CoreSight	TAP Number	Modules Accessed Through That JTAG Port	
	No		IVA	SMSET
	No			Hardware accelerators
	No		DSP1	ADTF
	No		DSP2	ADTF
	No		L3 NoC statistics collectors	All instances
	No		L3 OCP watchpoint	OCP_WP_NOC
	No		Clock management instrumentation	CMI
	No			
	No		Power management instrumentation	PMI
Test Bank				
DFT-SS	No	0	P1500 for DFT	
CATSCAN	No	1	CATSCAN	
System Control	No	2	P1500	
Reserved	No	3	Reserved	
Reserved	No	4	Reserved	

Note

1. MPU_MA watchpoint programming is restricted to the MPU. Programming from the CS_DAP APB access port is not supported.
2. The MPU timestamp registers are part of the MPU_PRCM programming model and can be accesses from the two MPU cores.

For more information about ICEPick scan sequences (adding one or more TAPs to the scan chain), see:

http://processors.wiki.ti.com/images/f/f6/Router_Scan_Sequence-ICEpick-D.pdf

The preceding links connects to TI community resources. Linked contents are provided “AS IS” by the respective contributors. They do not constitute TI specifications and do not necessarily reflect the views of TI; see the TI Terms of Use.

Besides secondary debug TAPs, ICEPick also supports power, reset, and clock controls for non-JTAG debug cores. The debug cores are accessible through CS_DAP.

Table 33-5 summarizes the ICEPick debug core mapping.

Table 33-5. ICEPick Debug Core Mapping

Debug Core #	Module
0	MPU
1	IVA ILF3
2	IVA IME3
3	IVA CALC3
4	IVA IPE3
5	IVA MC3
6	IVA ECD3
7	Reserved
8	Reserved
9	ARP32 in EVE1
10	ARP32 in EVE2

Table 33-5. ICEPick Debug Core Mapping (continued)

Debug Core #	Module
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved

33.4 Primary Debug Support

33.4.1 Processor Native Debug Support

33.4.1.1 Cortex-A15 Processor

The dual Cortex-A15 processor supports the following native debug features:

- Halt mode and monitor mode debugging
- Six breakpoints and four watchpoints
- Asynchronous aborts
- Performance monitoring
- Cross-triggering: Allows stopping one CPU upon debug event (for example, breakpoint) detection in the other CPU

For more information about Cortex-A15 native debug support features, see the *Cortex-A15 Technical Reference Manual*.

33.4.1.2 Cortex-M4 Processor

The Cortex-M4 processor supports the following native debug features:

- Program halt and stepping
- Hardware breakpoints, breakpoint instruction
- Data watchpoint on access to data add, add range, and data value
- Register value accesses
- Debug monitor exception
- Memories accesses

For more information about Cortex-M4 native debug support features, see the *Cortex-M4 Technical Reference Manual*.

33.4.1.3 DSP C66x

The main components of the DSP subsystem are:

- TMS320C66x DSP core with an execution control and analysis module that uses TI's ICEMaker™ technology. The core includes the following debug capabilities:
  - Up to four hardware breakpoints and a breakpoint counter
  - Software breakpoints
  - Internally and externally generated triggers
  - Reset control
  - Emulation interrupt support
  - Modes for debugging without halting time-critical blocks of code
  - Options to protect user-selected blocks of code against debug activity
- Emulation memory access hardware. This hardware enables several methods for reading and writing memory and registers in the DSP subsystem during debugging.
- Advanced Event Triggering (AET) unit is used to generate debug actions for managing breakpoints, watchpoint, trace, timers/counters, event outputs to external logic of DSP, based on events detected by instruction and data bus comparators or by auxiliary event detectors. DSP's AET can also handle complex events with event state machine and event counters.

33.4.1.4 IVA Arm968

The Arm968E-S processor (in each ICONT) supports the following native debug features through its EmbeddedICE-RT logic:

- Two hardware watchpoints/breakpoints
- Halt mode and monitor mode debugging
- Debug control and status registers
- Debug communications channel

For more information about Arm968 native debug support features, see the *Arm968E-S Technical Reference Manual*.

33.4.1.5 ARP32

The APR32 processor in the EVE subsystem offers the following debug capabilities:

- Manual halt
- Single-step execution
- Breakpoint
- Cross-triggering
- Global run

33.4.1.6

33.4.2 Cross-Triggering

The device supports a cross-triggering feature that provides a way to propagate debug (trigger) events from one processor subsystem/module to another. For example, a given subsystem A can be programmed to generate a debug event, which can then be exported as a global trigger across the device. Another subsystem B can be programmed to be sensitive to the trigger line input and to generate an action on trigger detection.

Examples of debug events are: Processor entering debug state, watchpoint match, CS_PTM trigger, and so forth.

Examples of debug actions are: Debug request generation, restart (Cortex-A15 synchronous run), interrupt request generation, start/stop trace, and so forth.

The device implements two global cross-triggering lines: Trigger0 and Trigger1, also referred to as EMU0 and EMU1, respectively.

Subsystem cross-triggering is consolidated at the device level by the XTRIG module, which is embedded in the debug subsystem.

Note

XTRIG is not programmatically visible from the JTAG interface or any device processor. Thus, cross-triggering is programmed at the subsystem level.

The Trigger0 and Trigger1 lines can also be configured as external triggers and contribute to cross-triggering.

33.4.2.1 SoC-Level Cross-Triggering

Device-level cross-triggering is handled by the XTRIG module. XTRIG manages two emulation triggers: Trigger0 and Trigger1. These trigger lines are shared by all the device subsystems implementing cross-triggering and are used to facilitate co-emulation.

Table 33-6 summarizes the device cross-triggering capabilities.

Table 33-6. Device Cross-Triggering

Subsystem	Module		MPU Cores	CS_PTM / CT_TBR	Cortex-M4	Cortex-M4	DSP-C66x	EVEx ARP32	EVEx SMSET	IVA Arm968	IVA Arm968	HWA	SMSET	PMI	CMI	NOC_SC	OCP_WP_NOC	EMU0/ EMU1
		Trigger Input	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	✓
		Trigger Source																
MPU	MPU Cores	✓		✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	✓
	CS_PTM / CT_TBR	✓	✓		✓	✓	–	–	–	–	–	–	✓	✓	✓	✓	✓	–
IPUx	Cortex-M4	✓	✓	–		✓	✓	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	–
	Cortex-M4	✓	✓	–	✓		✓	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	–
DSPx	C66x	✓	✓	–	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EVEx	ARP32	✓	✓	✓	✓	✓	✓		✓	✓	✓	–	✓	✓	✓	✓	✓	✓
	SMSET	✓	✓	✓	✓	✓	✓	✓		✓	✓	–	✓	✓	✓	✓	✓	✓
IVA	Arm968 ICONT1	✓	✓	–	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
	Arm968 ICONT2	✓	✓	–	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
	HWA	–	–	–	–	–	–	–	–	–	–		–	–	–	–	–	–
	SMSET	–	–	–	–	–	–	–	–	–	–	–		–	–	–	–	–
PM	PMI	–	–	–	–	–	–	–	–	–	–	–	–		–	–	–	–
CM	CMI	–	–	–	–	–	–	–	–	–	–	–	–	–		–	–	–
SoC	NOC_SC	–	–	–	–	–	–	–	–	–	–	–	–	–	–		–	–
	OCP_WP_NOC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	✓		✓
	EMU0, EMU1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	

33.4.2.2 Cross-Triggering With External Device

Based on DRM settings, the Trigger0 and Trigger1 lines may also be used as external triggers. Trigger0 is connected to the EMU0 device pin, and Trigger1 is connected to the EMU1 device pin. The user must make sure to program the DRM module in accordance with Table 33-34.

33.4.3 Suspend

The device supports a suspend feature, which provides a way to stop a closely coupled hardware process running on a peripheral-IP when the host processor enters debug state. The suspend mechanism is important for debug to ensure that peripheral-IPs operate in a lock-step manner with a host controller processor.

An entry is provided for each peripheral-IP that must consider the suspend signals from a number of processors (MPU or DSP). For each peripheral-IP, sensitivity to the suspend signals is defined within two possibilities (and so coded using 1 bit):

- Peripheral-IP is sensitive to the suspend line request.
- Peripheral-IP ignores the suspend line request.

For more information about how to program the sensitivity, see the corresponding peripheral-IP TRM chapter.

33.4.3.1 Debug Aware Peripherals and Host Processors

Table 33-7 lists the mapping of the device processors to the suspend control input lines.

Table 33-7. Debug Subsystem Suspend Input Lines

Suspend Input Line	Host Processor
0	DSP1
1	IVA ICONT1
2	IVA ICONT2
3	IPU1_C0
4	IPU1_C1
5	MPU_C0
6	MPU_C1
7	DSP2
8	IPU2_C0
9	IPU2_C1
10	EVE1_ARP32
11	EVE2_ARP32
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved

Table 33-8 lists the mapping of the device peripheral-IPs to the suspend control output lines.

Table 33-8. Debug Subsystem Suspend Output Lines

Suspend Output Line	Peripheral-IP Module	DRM Suspend Control Register
0	KBD	DRM_SUSPEND_CTRL0
1	DCAN1	DRM_SUSPEND_CTRL1
2	DCAN2	DRM_SUSPEND_CTRL2
3	Reserved	DRM_SUSPEND_CTRL3

Table 33-8. Debug Subsystem Suspend Output Lines (continued)

Suspend Output Line	Peripheral-IP Module	DRM Suspend Control Register
4	Reserved	DRM_SUSPEND_CTRL4
5	TIMER5	DRM_SUSPEND_CTRL5
6	TIMER6	DRM_SUSPEND_CTRL6
7	TIMER7	DRM_SUSPEND_CTRL7
8	TIMER8	DRM_SUSPEND_CTRL8
9	Reserved	DRM_SUSPEND_CTRL9
10	TIMER13	DRM_SUSPEND_CTRL10
11	Reserved	DRM_SUSPEND_CTRL11
12	TIMER14	DRM_SUSPEND_CTRL12
13	PWMSS1	DRM_SUSPEND_CTRL13
14	PWMSS2	DRM_SUSPEND_CTRL14
15	Reserved	DRM_SUSPEND_CTRL15
16	TIMER1	DRM_SUSPEND_CTRL16
17	TIMER2	DRM_SUSPEND_CTRL17
18	TIMER3	DRM_SUSPEND_CTRL18
19	TIMER4	DRM_SUSPEND_CTRL19
20	TIMER9	DRM_SUSPEND_CTRL20
21	TIMER10	DRM_SUSPEND_CTRL21
22	TIMER11	DRM_SUSPEND_CTRL22
23	TIMER12	DRM_SUSPEND_CTRL23
24	TIMER15	DRM_SUSPEND_CTRL24
25	I2C1	DRM_SUSPEND_CTRL25
26	I2C2	DRM_SUSPEND_CTRL26
27	I2C3	DRM_SUSPEND_CTRL27
28	I2C4	DRM_SUSPEND_CTRL28
29	TIMER16	DRM_SUSPEND_CTRL29
30	DMA_SYSTEM	DRM_SUSPEND_CTRL30
31	COUNTER_32K	DRM_SUSPEND_CTRL31
32	Statistics collector	DRM_SUSPEND_CTRL32
33	I2C5	DRM_SUSPEND_CTRL33
34	VCP1	DRM_SUSPEND_CTRL34
35	VCP2	DRM_SUSPEND_CTRL35
36	PWMSS3	DRM_SUSPEND_CTRL36
37	WD_TIMER2	DRM_SUSPEND_CTRL37

Note

Table 33-8 lists only peripherals that support the suspend feature. For modules not listed in this table, the suspend feature is not supported.

33.5 Real-Time Debug

33.5.1 Real-Time Debug Events

33.5.1.1 Emulation Interrupts

A few device interrupt channels are dedicated to debug support. Table 33-9 summarizes the emulation interrupt events.

Table 33-9. Emulation Interrupts

Interrupt Request	Subsystem	Source	Description
COMMRX	MPU	MPCore	Debug Communication Channel
COMMTX			
PMUIRQ			Performance Monitoring
IRQ[0]		MPU	L2 Cache
IRQ[1]		CS_CTI_MPU_C0	Emulation trigger; Trace buffer full; Trace acquisition complete
IRQ[2]		CS_CTI_MPU_C1	
RTOS_INT	DSP	DSP ICEMaker	RTOS interrupt from analysis event
DLOG_INT			Data logging transfer complete
EMUINT2			Monitor mode debug
ICNEMUINTR	IVA	Arm968 ICECrusher-9	Debug Control & Status register; Emulation trigger
ICNCOMMRX			Debug Communication Channel
ICNCOMMTX			
ICNEMUINTR		Arm968 ICECrusher-9	Debug Control & Status register; Emulation trigger
ICNCOMMRX			Debug Communication Channel
ICNCOMMTX			
ICNEMUINTR	IPU	Cortex-M4 ICECrusher-CS	Debug Control & Status register; Emulation trigger
ICNCOMMRX			Debug Communication Channel
ICNCOMMTX			
ICNEMUINTR		Cortex-M4 ICECrusher-CS	Debug Control & Status register; Emulation trigger
ICNCOMMRX			Debug Communication Channel
ICNCOMMTX			
SC_ALERT	L3_MAIN	Statistics collectors	Performance monitoring alert (metric out of range)
DCC_UART_TX	DEBUGSS	CT_UART (BDX client)	Transmit empty
DCC_UART_RX			Receiver data available

33.6 Power, Reset, and Clock Management Debug Support

The global PRCM module implements facilities to support debug across power and clock domain cycles. The debugger can control or get the status of each power and clock domain associated with an ICEPick secondary TAP.

ICEPick provides a set of directives allowing the debugger to:

- Get visibility on the associated power and clock domains state. This includes:
  – Current power setting indicating whether the power domain is on or off
  – Loss of power detected since the software last checked the status
  – Current clock setting indicating whether the clock domain is on or off
  – Sleep desired (PM and CM indicate that the debug settings in ICEPick are changing the application state. If it were not for the ICEPick controls, the power or clock would be turned off.)
  – Subsystem reset state
  – Subsystem has entered a debug state that requires the attention of the host debug software.
- Override power/clock control settings to wake up a power or clock domain or to prevent a power or clock domain from going to sleep once it is in ACTIVE state
- Assert/block/extend reset; release from extended reset (WIR)

33.6.1 Power and Clock Management

33.6.1.1 Power and Clock Control Override From Debugger

The debugger can override the application software power and clock management settings through the ICEPick module. It can configure ICEPick to force a domain active or prevent it from going to sleep once it is active. This can be achieved through the following debugger directives:

- FORCEACTIVE
- INHIBITSLEEP

33.6.1.1.1 Debugger Directives

33.6.1.1.1.1 FORCEACTIVE Debugger Directive

To ensure that the subsystem debug registers can always be accessed, regardless of the application power-management scenarios, a FORCEACTIVE directive can be issued through the debugger along the entire debug session. From an application standpoint, the system state, status, and timing are preserved, but the subsystem power domain is never shut down. Therefore, the emulation setup is preserved across power transitions, regardless of where the debug hardware is implemented.

If the debugger connects and the subsystem were previously powered down, a FORCEACTIVE directive wakes up the system and allows the debugger to take control of the system.

33.6.1.1.1.2 INHIBITSLEEP Debugger Directive

The debugger can use the INHIBITSLEEP directive to keep a subsystem powered and clocked, even if the applicative settings request that this subsystem go to sleep. Contrary to the FORCEACTIVE directive, the INHIBITSLEEP command does not wake up a subsystem that is already powered down by the application.

The typical use of the INHIBITSLEEP directive is to prevent power and clock transitions on the subsystem during a debug session. In this situation, when the applicative scenario initiates a transition, the transition does not take place, but from an applicative point of view the subsystem is not accessible.

33.6.1.1.2 Intrusive Debug Model

The use of debugger directives is intrusive from the standpoint of the power and clock controls, because they affect the power-management behavior of the application.

33.6.1.2 Debug Across Power Transition

33.6.1.2.1 Nonintrusive Debug Model

To preserve the power-management behavior of the device and allow the subsystem power and clock to be switched off by application software, the subsystem TAP must be disconnected from the ICEPick scan chain. The subsystem is then completely ignored by the debugger and the host-to-target communication is no longer affected by the state of the subsystem power and clocks. The debugger can still be informed that the disconnected subsystem entered the debug state by polling the Debug Attention status bit from ICEPick. The debugger can then insert the TAP, take control of the subsystem power and clock, and examine the system state.

This debug model is nonintrusive, because it allows the power-management behavior of the application to be preserved.

33.6.1.2.2 Debug Context Save and Restore

33.6.1.2.2.1 Debug Context Save

The device partitioning is such that not all the debug components are mapped to an always-on domain. Typically, the programmer wants the debug setup to be preserved along a debug session, including subsystem power cycling. When debug registers are memory mapped and not implemented within the emulation power domain, the application software must save the state of the debug registers before going to sleep and restore them upon wakeup.

33.6.1.2.2.2 Debug Context Restore

When the application software performs a context restore, it must be able to write to all the debug registers to restore their contents, regardless of the previous ownership. After subsystem power domain ramp up, the debug resources are in the available state, and ownership is restored. The debug context save and restore sequences are protected. All debug functionality is disabled and debugger accesses are blocked.

33.6.2 Reset Management

The debugger can take control of the system reset for each subsystem through ICEPick. The debugger can configure ICEPick to assert, block, or extend the subsystem reset.

33.6.2.1 Debugger Directives

33.6.2.1.1 Assert Reset

The debugger can program ICEPick to generate a subsystem reset request to the device reset management module. The debugger reset event is then merged with system reset events.

33.6.2.1.2 Block Reset

The debugger can program ICEPick to request the device reset management module to block an unsafe application system or subsystem reset event.

A reset originated by some safe reset sources cannot be blocked by the debugger.

33.6.2.1.3 Wait-In-Reset

WIR mode is latched from boot mode (see Section 33.3.2, *Boot Modes*) and allows the user to hold a secondary TAP module in reset state when a reset is applied (and thus, extend the reset). This mode lets the user control the following system level activities:

- Gain emulation control of any processor in a power domain at POR
- Capture and extend system-generated functional resets while running (under emulation control or not)
- Hold an entire power domain in reset until emulation control of the subsystem can be established
- Stall the entire system while reset is extended to a power domain
- Reset extension is visible external to the power domain.
- Debug execution of code from the first cycle of execution
- Prevent processor execution of random instructions in uninitialized program memory at power up
- Download code before any code execution takes place
- Coordinate debug initialization across multiple cores before code execution begins

WIR mode extends only the processor reset. During reset extension, the debugger can still access modules such as L2 memory and MMU, even if they are embedded in the device subsystem affected by WIR.

When the debugger task is complete, the DTC releases the subsystem reset by programming the corresponding ICEPick TAP control register.

33.7 Performance Monitoring

33.7.1 MPU Subsystem Performance Monitoring

33.7.1.1 Performance Monitoring Unit

The Cortex-A15 processor includes a PMU that enables events, such as cache misses and instructions executed, to be counted over a period of time. The PMU provides six counters to gather statistics about the operation of the processor and memory system. Each counter can count any of the events available in Cortex-A15. Upon counter overflow, PMU can generate an interrupt on its PMUIRQ output. This interrupt signal is mapped to the CTITRIGIN[1] input and routed to an MPU_INTC input (MPU_IRQ_131 for Cortex-A15 CPU0 PMU; MPU_IRQ_132 for Cortex-A15 CPU1 PMU).

The Cortex-A15 PMU outputs events to CS_PTM. For details of PMU events, please refer to the Arm Cortex-A15 TRM, available at http://infocenter.arm.com/help/index.jsp.

33.7.1.2 L2 Cache Controller

The MPU subsystem includes 2 MiB of L2 cache (L2CACHE_MPU) and L2 cache controller (L2CACHE_CTRL_MPU). The L2 cache controller includes logic to support cache event monitoring.

Table 33-10 summarizes the L2 cache events.

Table 33-10. L2 Cache Events

Event	Event Description
0	Eviction of a line from the L2 cache
1	Data read hit in the L2 cache
2	Data read lookup to the L2 cache. Subsequently results in a hit or miss.
3	Data write hit in the L2 cache
4	Data write lookup to the L2 cache. Subsequently results in a hit or miss.
5	Data write lookup to the L2 cache with Write-Through attribute. Subsequently results in a hit or miss.
6	Instruction read hit in the L2 cache
7	Instruction read lookup to the L2 cache. Subsequently results in a hit or miss.
8	Prefetch line-fill sent to L3
9	Allocation into the L2 cache caused by a write (with Write-Allocate attribute) miss

The L2 cache controller implements two 32-bit event counters. The L2 cache controller can be configured to generate interrupts on error conditions or event counter overflow or increment. The L2 cache controller interrupt is routed to MPU_IRQ_0. When an interrupt occurs, software can look at the relevant interrupt status register to determine the source of the interrupt.

33.7.2 IPU Subsystem Performance Monitoring

33.7.2.1 Subsystem Counter Timer Module

The IPU subsystem includes a subsystem counter timer module (SCTM), which is embedded in shared cache and provides additional data to the user timing or profiling capability. SCTM integrates eight profiling counters that collect:

- Forty-four shared cache events
- Four sleep/deep-sleep events from Cortex-M4 cores (one sleep and one deep sleep event per core)

Table 33-11 describes the repartition of the IPU SCTM counters.

Table 33-11. IPU SCTM Counters Repartition

Counters	Features
0–1	Timer and event
2–3	64-bit chained + shadowing
4–5	64-bit chained + shadowing

Table 33-11. IPU SCTM Counters Repartition (continued)

Counters	Features
6–7	Event

33.7.2.2 Cache Events

Table 33-12 summarizes the SCTM events for the IPU subsystem.

Table 33-12. SCTM Events for IPU Subsystem

Input Index	Event Description
1	Cache locks
2	Cache line replacements
3	Cache evictions
4	Cache maintenance operations (slave 0)
5	Cache maintenance operations (slave 1)
6	Cache maintenance operations (slave 2)
7	Cache maintenance operations (slave 3)
8	Cache OCP access (slave 0)
9	Cache OCP access (slave 1)
10	Cache OCP access (slave 2)
11	Cache OCP access (slave 3)
12	Cacheable access (slave 0)
13	Cacheable access (slave 1)
14	Cacheable access (slave 2)
15	Cacheable access (slave 3)
16	Cache bank conflicts (slave 0)
17	Cache bank conflicts (slave 1)
18	Cache bank conflicts (slave 2)
19	Cache bank conflicts (slave 3)
20	Cache allocations
21	Cache write buffer accesses (slave 0)
22	Cache write buffer accesses (slave 1)
23	Cache write buffer accesses (slave 2)
24	Cache write buffer accesses (slave 3)
25	Cache line fills (slave 0)
26	Cache line fills (slave 1)
27	Cache line fills (slave 2)
28	Cache line fills (slave 3)
29	Cache write fills (slave 0)
30	Cache write fills (slave 1)
31	Cache write fills (slave 2)
32	Cache write fills (slave 3)
33	Cache read fills (slave 0)
34	Cache read fills (slave 1)
35	Cache read fills (slave 2)
36	Cache read fills (slave 3)
37	Cache misses (slave 0)
38	Cache misses (slave 1)
39	Cache misses (slave 2)

Table 33-12. SCTM Events for IPU Subsystem (continued)

Input Index	Event Description
40	Cache misses (slave 3)
41	Cache hits (slave 0)
42	Cache hits (slave 1)
43	Cache hits (slave 2)
44	Cache hits (slave 3)
45	IPU_C1 deep sleep
46	IPU_C1 sleep
47	IPU_C0 deep sleep
48	IPU_C0 sleep

Note

Input index [0] is reserved for free-running subsystem clock (used for total cycle profiling).

33.7.3 DSP Subsystem Performance Monitoring

33.7.3.1 Advanced Event Triggering

The Advanced Event Trigger (AET) unit has the capability to generate the debug actions based on events detected by instruction and data bus comparators or by auxiliary event detectors to manage:

- Hardware Program Breakpoints: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- Data Watchpoints: specify data variable address, address ranges, or data values that can generate events such as halting the processor or triggering the trace captures.
- Counters: count the occurrence of an event of cycles for performance monitoring.
- State Sequencing: allows combinations of hardware program breakpoint and data Watchpoints to precisely generate events for complex sequences.

For more information on AET, see the following documents:

- *Using Advanced Event triggering to Find and Fix Intermittent Real-Time Bugs* application report (SPRA753).
- *Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor System* application report (SPRA387)

33.7.4 EVE Subsystem Performance Monitoring

33.7.4.1 EVE Subsystem Counter Timer Module

The EVE subsystem includes an SCTM, which is embedded in the EVE subsystem and provides additional data to the user timing or profiling capability. SCTM integrates eight profiling counters that collect:

- ARP32 program cache events
- EDMA events
- Interrupt events
- VCOP processor events

Table 33-13 describes the configuration of the EVE SCTM counters.

Table 33-13. EVE SCTM Counters Configuration

Counters	Features
0–1	Timer and event
2–3	64-bit chained + shadowing
4–5	64-bit chained + shadowing
6–7	Event

33.7.4.2 EVE Subsystem SCTM Events

Table 33-14 summaries the SCTM events for EVE subsystem.

Table 33-14. SCTM Events for EVE Subsystem

Input Index	Event Description
0	Cache miss
1	Cache hit
2	Cache miss stall
3	Prefetch compulsory count
4	Prefetch look ahead count
5	Prefetch hit count
6	Prefetch flush line count
7	Prefetch flush occur count
8	Prefetch discard stall
9	TPCC AET
10	ARP32 Interrupt 4
11	ARP32 Interrupt 5
12	ARP32 Interrupt 6
13	ARP32 Interrupt 7
14	VCOP busy
15	VCOP idle and done
16	VCOP wait for ARP32
17	VCOP ARP32 awaits
18	VCOP overhead
19	VCOP ld_stall_by_st
20	VCOP op_stall_by_ldst
21	VCOP op_stall_by_dependency
22	VCOP rd_ibufl
23	VCOP rd_ibufh
24	VCOP rd_wbuf
25	VCOP wr_ibufl
26	VCOP wr_ibufh
27	VCOP wr_wbuf
28	VCOP loop_start
29	VCOP done

33.8 MPU Memory Adaptor (MPU_MA) Watchpoint

The MPU subsystem implements a watchpoint allowing the user tracking the MPU_MA transactions from/to external memory.

The watchpoint can be programmed to generate a trigger when a MPU_MA transaction satisfies a set of user-defined attributes:

- Access within or outside an address region
- Access type (instruction, data, read, write)
- Transaction target (on-chip, memory channel, chip select)
- Qualifier
- Transaction originating from a specific core
- Specific transaction followed by memory barrier (DSB, DMB)
- Memory barrier (DSB, DMB) followed by specific transaction

When the MPU_MA transaction attributes match the debug use case setup, the MPU debug logic captures the transaction data and associated qualifiers:

- Read or write data (128-bit) for the specific beat of the burst
- Address (MSB field)
- MReqInfo (19-bit)
- Byte enables
- Burst type
- Burst length
- Target

When the MPU_MA watchpoint is configured to track memory barriers in order to investigate potential ordering issues, the debugger will report:

- Attributes of the transaction immediately following the memory barrier
- Attributes of the matching transaction chained to the memory barrier

33.9 Processor Trace

The device supports:

- MPU Cortex-A15 processor trace
- DSP (C66x) processor trace for each DSP subsystem

Figure 33-2 shows an overview of the MPU and DSP processor traces flow.

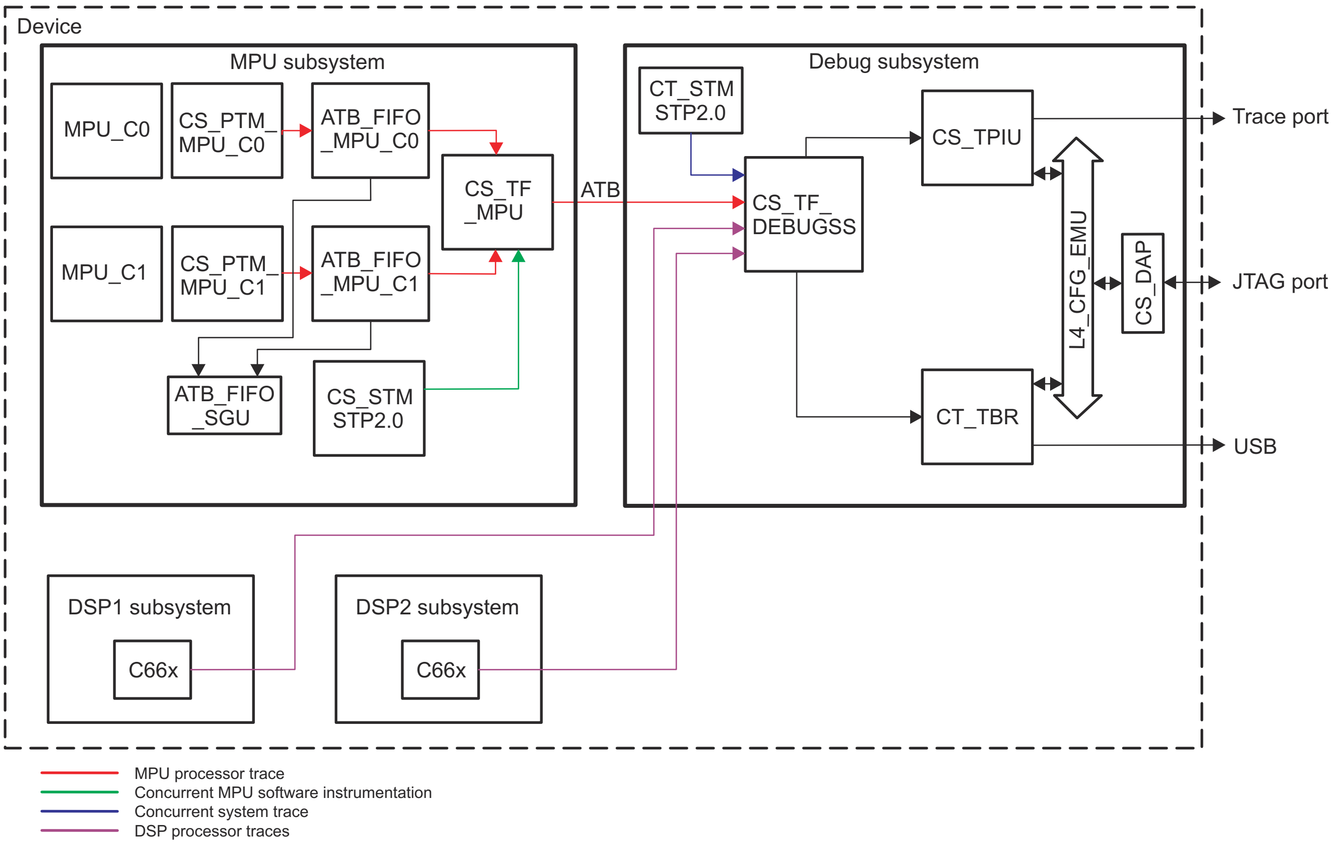


Figure 33-2. MPU and DSP Processor Traces Flow

Note

Mapping of trace funnels inputs:

- CS_TF_MPU:
  - Port 0 – ATB_FIFO_MPU_C0
  - Port 1 – ATB_FIFO_MPU_C1
  - Port 2 – CS_STM
- CS_TF_DEBUGSS_1
  - Port 0 – MPU ATB interface
  - Port 1 – DSP1 ATB interface
  - Port 2 – DSP2 ATB interface
  - Port 7 – CT_STM ATB interface

33.9.1 Cortex-A15 Processor Trace

The main MPU trace characteristics are:

- Program trace only (no data trace)
- Separate CS_PTM for each MPU core

- Three exclusive trace sinks:
  – CS_TPIU – trace exported to an external trace receiver
  – CT_TBR (buffer mode) – trace stored into on-chip buffer
  – CT_TBR (system bridge mode) – trace exported through USB
- Trace can be optionally:
  – Cycle accurate useful for profiling sections of code
  – Locally time-stamped using a 48-bit free-running graycode counter exported to the CS_PTM of each MPU core
  – Interleaved dynamically between the two MPU cores by CS_TF_MPU

Note

Depending on use case, the device may not be able to interleave two cycle accurate traces without overflow (bandwidth restriction).

33.9.2 DSP Processor Trace

Trace targets the debug of unstable code, performance analysis, and quality assurance. This infrastructure component use bus snoopers to collect and export trace data using hardware dedicated to the trace function. The use of dedicated hardware to both collect, buffer, transfer trace data to the host makes trace non-intrusive. DSP processor trace characteristics are:

- Program trace
- Data trace (address and value)
- Time stamp
- Three exclusive trace sinks:
  – CS_TPIU – trace exported to an external trace receiver
  – CT_TBR – trace history stored into on-chip buffer
  – CT_TBR – trace exported through USB

The trace function can collect and export a record of the program flow and timing at the same rate generated by the CPU. Tracing data references must be restricted however as the export mechanism is generally limited to size of on-chip CT_TBR to sustain tracing of all memory references. In the case of data trace, the Advanced Event Triggering facilities provide a means to restrict the trace data exported to data of interest to maintain the non-intrusive aspect of trace. This reduces the export bandwidth needs and facilitates the successful collection of the data of interest. Error indications are embedded in the debug stream in the event the export logic is unable to keep up with the data rate generated by the collection logic. This notification allows the user to scale back the amount of requested data collection.

The user can optionally select the export of all specified trace data. In this case the CPU is stalled to avoid the loss of trace data, with trace becoming intrusive to the application if trace related stalls are generated. The user is notified that trace stalls.

33.9.3 Trace Export

The debug subsystem implements three exclusive trace sinks:

- CS_TPIU
- CT_TBR (buffer mode)
- CT_TBR (system bridge mode)

Two MPU trace streams are supported, one per MPU core. The two MPU cores program traces are exported to the PD_EMU domain (through asynchronous bridges) and then interleaved by the CS_TF_MPU.

A concurrent software instrumentation flow from the local CS_STM component (STP 2.0) can also be interleaved with the MPU program traces.

Two ATB_FIFO modules (ATB_FIFO_MPU_C0 and ATB_FIFO_MPU_C1) are implemented in the trace path between the Cortex-A15 CPUs and the CS_TF_MPU module. The ATB_FIFO allows concurrent trace capture and export. It provides buffering in the trace export path and therefore allows absorbing peaks of trace data. The depth of the ATB_FIFOs is 16 entries.

The CS_TF_MPU sends the MPU trace (or CS_STM software instrumentation) to a trace funnel in the debug subsystem (CS_TF_DEBUGSS) through a single ATB interface. The CS_TF_DEBUGSS redirects the MPU trace to the three exclusive trace sinks.

Note

It is strongly recommended to disable trace sinks not in use when generating trace to an in-use trace sink to eliminate undesirable throughput throttling effects.

Besides the ATB_FIFOs, the MPU subsystem also instantiates an ATB_FIFO statistics gathering unit (ATB_FIFO_SGU) which provides silicon and presilicon characterization information of ATB_FIFO. The FIFO_LEVEL_OUT output of ATB_FIFO, which gives visibility of the number of used entries, is used for the statistics gathering. ATB_FIFO may be full under the following conditions:

- Arbitration collisions in the trace funnel when more than one ATB trace stream is being generated at the same time (tracing of multiple CPUs simultaneously)
- Trace sink does not have the bandwidth to meet peak bandwidth demands from the trace generator on a single ATB trace stream

The following statistics are gathered per ATB_FIFO:

- Peak utilization depth of the FIFO
- Number of consecutive cycles that FIFO level is at or above the programmable FIFO threshold when there is an ATB command pending
- Number of ATB bus stall cycles

33.9.3.1 Trace Exported to External Trace Receiver

Processor trace can be exported to an external trace receiver through the CS_TPIU module. To achieve this, the debugger or application software must ensure to program the DRM module properly (that is, according to Table 33-34).

The CS_TPIU has a configurable export width of maximum 16 data pins (TRACEDATA) plus a dedicated export clock (TRACECLK) and a control signal (TRACECTL).

Note

In case of MPU trace streams interleaved, the trace receiver software must demultiplex the two trace streams before decoding each individual stream.

33.9.3.2 Trace Captured Into On-Chip Trace Buffer

Processor trace flow can be redirected to the on-chip trace buffer (CT_TBR working in buffer mode). The CT_TBR provides on-chip storage of trace data using a 32-KiB RAM memory.

The CT_TBR receives trace flow data through its ATB port. The debugger can then access trace data through CS_DAP (APB port).

The CT_TBR supports two modes when in buffer mode configuration:

- Circular buffer mode for continuous capture. In this mode, the CT_TBR functions as a circular buffer where the oldest data is overwritten with the most recent entries.
- Stop-on-full mode for no-loss (single shot) capture. In this mode, the buffer fills linearly and then halts when the last entry is written.

33.9.3.3 Trace Exported Through USB

Processor trace can be exported off-chip through an application interface (USB). For this purpose, the debug subsystem instantiates a CTools trace buffer rooter (CT_TBR) module, which operates in a system bridge configuration and basically represents a temporary buffer that holds data that is awaiting transfer to the L3_MAIN interconnect for export on functional I/O interfaces (USB) or storage in system memory.

The CT_TBR allows concurrent trace capture and export. A USB3.0 DMA engine (L3_MAIN) master can read trace data from the CT_TBR slave port as soon as a block is available for transfer.

The CT_TBR supports a DMA request generation via a read-only burst-capable interface. The primary purpose of the DMA event generation is to signal a DMA engine when the CT_TBR export FIFO has enough data to satisfy a burst read request on the system interface. The threshold for the DMA trigger is software configurable. A new DMA event is generated when the threshold is met again with new data being written to the FIFO.

The USB3.0 DMA engine supports hardware DMA requests. The CT_TBR can generate a DMA request (CT_TBR_DREQ) to the DMA_SYSTEM module (mapped to DMA_SYSTEM_DREQ_6 request line) for system memory storage.

The USB3.0 master only uses a single AXI-ID and does not support out of order execution. Therefore tracing through USB will prevent application software using USB simultaneously due to stalls from CT_TBR.

Such configuration allows supporting program trace on end product without additional debug pins taking advantage of the USB PHY throughput.

33.10 System Instrumentation

The device supports the following system instrumentation features:

- Real-time software trace
  - MPU software instrumentation via CS_STM (STP2.0) (see Section 33.10.3.1, *MPU Software Instrumentation*)
  - SoC software instrumentation via CT_STM (STP2.0) (see Section 33.10.3.2, *SoC Software Instrumentation*)
- OCP watchpoint
  - OCP target traffic monitoring (see Section 33.10.4.1, *OCP Target Traffic Monitoring*)
  - SoC events trace (see Section 33.10.4.2, *Messages Triggered from System Events*)
  - DMA transfer profiling (see Section 33.10.4.3, *DMA Transfer Profiling*)
- Statistics collector
  - L3 Target Load Monitoring (see Section 33.10.7.1, *L3 Target Load Monitoring*)
  - L3 Master Latency Monitoring (see Section 33.10.7.2, *L3 Master Latency Monitoring*)
- IVA HWA load monitoring (see Section 33.10.5, *IVA Pipeline*)
- PM events trace (see Section 33.10.8, *PM Instrumentation* [PMI])
- CM events trace (see Section 33.10.9, *CM Instrumentation* [CMI])

33.10.1 MIPI STM (CT_STM)

CT_STM is a trace module that aids in software debugging. The main features of this module are:

- Implements MIPI STP protocol (rev 2.0) with the following characteristics:
  - Highly optimized for software-generated traces
  - Automatic timestamping of messages
  - Support for 8-, 16-, and 32-bit data types
- Collects the following information:
  - Software messages
  - Hardware instrumentation trace from hardware agents:
    - OCP_WP_NOC
    - PMI
    - CMI
    - SMSET
    - L3 NoC statistics collectors
- Supports the following trace export paths:
  - ATB export - to CS_TPIU or CT_TBR
- Available in 1-, 2-, or 4-pin mode with single- or dual-edge clock, depending on the trace bandwidth requirements and characteristics of the trace receiver
- Timestamps:
  - Can use local relative timestamp if exported to the CS_TPIU/CT_TBR (through ATB)
- Dedicated 128 × 48-bit FIFO buffer
- Mechanism to generate repeat MASTER and C8 messages periodically even if not needed

A maximum of 255 different bus masters can be connected to the STM trace port through a bus arbiter. STP recognizes two distinct modes of tracing (software and hardware types), which use slightly different message combinations to output different types of data. The bus masters can be configured for either type to optimize the system for the different types of trace data.

33.10.2 System Trace Export

33.10.2.1 CT_STM ATB Export

If a trace receiver cannot be attached to the device, or relevant CT_STM trace port pins are unavailable for a particular reason, the user can configure the CT_STM module to redirect the STP trace stream as an ATB stream to the CS_TPIU or CT_TBR and enable local timestamp. This is accomplished by outputting a local timestamp granularity (LTSG) message, which is a TI addition to the MIPI standard messages.

33.10.2.2 Trace Streams Interleaving

Two levels of interleaving system instrumentation flows and arbitrating between instrumentation masters are implemented:

- CORE L3 instrumentation interconnect interleaves data coming from the following bus masters:
  - OCP_WP_NOC
  - L3 NoC statistics collectors, or software instrumentation (interleaving at the L3 level)
  - IVA SMSET and Arm968 (software instrumentation)
- EMU L3 instrumentation interconnect interleaves data coming from the following bus masters:
  - CORE L3 instrumentation interconnect
  - CMI
  - PMI

33.10.3 Software Instrumentation

33.10.3.1 MPU Software Instrumentation

The CS_STM embedded in the MPU subsystem provides extended stimulus port registers designated to be accessible by software with minimum instrumentation cycles overhead.

Each extended stimulus port occupies 256 consecutive bytes in the memory map and is write-only. Up to 64K instrumentation channels are available at MPU level.

The CS_STM supports "guaranteed" or invariant timing transactions:

- Guaranteed transactions are guaranteed to be traced. This might involve stalling the MPU core to ensure the transaction is accepted by the CS_STM.
- Invariant timing transactions are not guaranteed to be traced. These transactions take an invariant amount of time regardless of the state of the CS_STM.

The CS_STM implements tracing of software writes to its stimulus ports using a dedicated AXI slave interface. In addition to the AXI interface, the CS_STM provides a hardware event input interface (HWEVENTS[31:0]). The HWEVENTS[31:0] input bus is connected to the MPU hardware debug observability signals going out to the MPU hardware debug port (MPUHWDBGOUT[31:0]). This enables any of the signals observed on MPUHWDBGOUT[31:0] to act as a hardware event to the CS_STM. The CS_STM can be programmed to generate a debug packet on a rising edge transition on any signal of the HWVENTS[31:0] input bus. Because some of the signals mapped on the MPUHWDBGOUT[31:0] bus are active low, a 32-bit programmable register – MPU.STM_HWEVENTS_INV, is implemented for polarity inversion. Each bit of MPU.STM_HWEVENTS_INV when set to 0x1 inverts the polarity of the corresponding signal of MPUHWDBGOUT[31:0] going to HWEVENTS[31:0]. This programmable register is part of the MPU_WUGEN address map (see Chapter 4, *Dual Cortex-A15 MPU Subsystem*, for its description).

Note

The MPUHWDBGOUT[31:0] signals can be routed to the hw_dbg[31:0] device pads through various hardware debug observability MUXes controlled by device control module (CTRL_MODULE). For more information, see Chapter 18, *Control Module*.

To save power, the output port can be gated off (to all zeros) by setting the CTRL_MODULE. CONTROL_HWOBS_CONTROL[0] HWOBS_MACRO_ENABLE bit to 0x0. Ungated version of MPUHWDBGOUT[31:0] (not gated with CTRL_MODULE. CONTROL_HWOBS_CONTROL[0] HWOBS_MACRO_ENABLE) is sent to HWEVENTS[31:0] input of STM so each of these signals can generate a debug packet on the trace port.

The message structures in STP-2.0 are optimized to provide an efficient transport.

33.10.3.2 SoC Software Instrumentation

The CT_STM module embedded in the debug subsystem provides a flexible interface for trace instrumentation.

The device provides support for real-time software trace through user-defined message writes to specific memory mapped register (MMR) locations. Software masters can transmit trace data from the operating system (OS) processes or tasks on 256 different channels, with each channel being defined by the software protocol implemented. The different channels can be used to group different types of data logically so that it is easy to filter out the data irrelevant to the ongoing debugging task. The message structures in STP-2.0 are optimized to provide an efficient transport for software data through the CT_STM module.

The software masters are:

- Cortex-A15 MPU subsystem
- DAP (for testing purpose)
- DSP subsystem
- IPU subsystem
- EVE subsystem
- SDMA controller write port (DMA_SYSTEM_WR)
- IVA subsystem
- EDMA TPTC WR1 and WR2 channels

Each software master has a master-ID assigned to it (see Section 33.10.10, *Master-ID Encoding*, for more information).

Software messages can be interleaved with other hardware messages.

Software messages are intrusive and use both processor cycles and memory.

33.10.4 OCP Watchpoint

33.10.4.1 OCP Target Traffic Monitoring

The L3_MAIN interconnect provides several functional probes embedded and attached to the following L3_MAIN targets:

- GPMC
- L4_PER1, L4_PER2, L4_PER3
- L4_CFG
- DMM_P1 (DMM target port 1)
- DMM_P2 (DMM target port 2)
- OCMC_RAM1

The probes output are muxed together and then sent to the L3_MAIN interconnect debug port. A component called OCP_WP_NOC is used to collect data from functional probes and then transmit captured data to the CT_STM module. The OCP_WP_NOC drives a probe-ID signal to the L3_MAIN interconnect for probe selection. The probe selection is exclusive, meaning that interleaving is not possible.

The OCP_WP_NOC provides the following main features:

- Monitoring the OCP traffic originated by all initiators that can access the selected target where the probe is attached
- Filtering OCP monitored bus traffic by:
  - Address range
  - Initiator-ID (see Table 33-16)
  - Transaction type
  - Transaction qualifier
- Generating a trigger upon watchpoint match
- Starting and stopping OCP traffic monitoring upon:
  - WP address match
  - External trigger
- Profiling DMA transfers

- Generating hardware message upon system event
- OCP_WP_NOC messages can be interleaved with software messages
- Programming from:
  - Debugger
  - Application

Note

The OCP_WP_NOC is restricted to monitor request flow only.

Table 33-15 summarizes the OCP targets that can be monitored by the OCP_WP_NOC and their respective probe-ID.

Table 33-15. L3_MAIN Interconnect Functional Probe Mapping

Probe-ID	L3 OCP Target
0000	Reserved
0001	GPMC
0010	L4_PER1
0011	L4_CFG
0100	DMM_P1 (DMM target port 1)[(1)]
0101	DMM_P2 (DMM target port 2)[(1)]
0110	OCMC_RAM1
0111	L4_PER2
1000	L4_PER3

(1) Does not allow tracking MPU transactions routed through Memory Adapter and DMM.

The user can program the OCP_WP to extract the traffic from a specific set of initiators (maximum four master-IDs). Table 33-16 lists the master-ID reported by the L3_MAIN debug port for the device initiators.

Table 33-16. Master-ID Mapping (Debug View)

Master-ID	Initiator
0x1	MPU
0x4	CS_DAP
0x5	IEEE1500_2_OCP
0x8	DSP1 MDMA
0x9	DSP1 CFG
0xA	DSP1 DMA
0xB	DSP2 DMA
0xC	DSP2 CFG
0xD	DSP2 MDMA
0xE	IVA ICONT1
0x10	EVE1 P1
0x11	EVE2 P1
0x12	Reserved
0x13	Reserved
0x14	Reserved
0x15	Reserved
0x16	Reserved
0x17	Reserved
0x18	IPU1
0x19	IPU2

Table 33-16. Master-ID Mapping (Debug View) (continued)

Master-ID	Initiator
0x1A	DMA_SYSTEM RD
0x1A	DMA_SYSTEM WR
0x1B	Reserved
0x1B	Reserved
0x1C	EDMA_TC1 WR
0x1C	EDMA_TC1 RD
0x1D	EDMA_TC2 WR
0x1D	EDMA_TC2 RD
0x20	DSS
0x21	MLB
0x21	MMU1
0x22	PCIe_SS1
0x23	PCIe_SS2
0x23	MMU2
0x24	VIP1 P1
0x24	VIP1 P2
0x25	VIP2 P1
0x25	VIP2 P2
0x26	VIP3 P1
0x26	VIP3 P2
0x27	VPE P1
0x27	VPE P2
0x28	MMC1
0x28	GPU P1
0x29	MMC2
0x29	GPU P2
0x2A	BB2D P1
0x2A	BB2D P2
0x2B	GMAC_SW
0x2B	Reserved
0x2C	USB4
0x2D	USB1
0x2E	USB2
0x2F	USB3
0x30	Reserved
0x31	Reserved
0x33	SATA
0x34	EVE1 P2
0x35	EVE2 P2
0x36	Reserved
0x37	Reserved

Note

For information about master-ID values from a protection/error logging point of view, see Chapter 14, *Interconnect*.

33.10.4.2 Messages Triggered from System Events

The OCP_WP_NOC can be programmed to export a message through the CT_STM upon detection of a system event (interrupt, DMA request, etc.). A bus of 16 system events pre-selected at SoC level from a large number of observable events is routed to the OCP_WP_NOC. This can be useful to determine interrupts latencies:
- Interrupt request traced through OCP_WP_NOC system event detection
- ISR boundary tracked either through CT_STM software trace or OCP_WP_NOC address range detection

33.10.4.3 DMA Transfer Profiling

The OCP_WP_NOC can be configured to profile DMA transfers. This feature provides to the user visibility on:
- DMA logical channel interleaving
- DMA channel ID
- DMA transfer duration (time stamp)
- DMA burst size
- DMA reads
- DMA writes

When operating in this mode, the OCP_WP_NOC exports to CT_STM:
- DMA channel ID
- Burst size
- R/W

The CT_STM module encapsulates the information above in a compact STP message.

The address range filtering remains active but OCP address data are not encapsulated into the trace message to maximize CT_STM throughput.

33.10.5 IVA Pipeline

The device takes advantage of the system trace infrastructure to provide visibility to the user regarding IVA microtask sequencing. This is supported through a SMSET module instantiated in the IVA subsystem. The microtask boundaries are handled as generic events and encapsulated in STP messages with an event-ID and local timestamp and exported through the CT_STM module.

The IVA instrumentation scheme allows the user to understand microtask dependencies, hardware accelerators load balancing, and potential bottlenecks. DMA transfer boundaries are reported as IVA events. Software messages from Arm968 execution can be interleaved with IVA events.

33.10.6 EVE SMSET

The device takes advantage of the system trace infrastructure to provide visibility to the user regarding EVE VCOP micro-task sequencing. This is supported through a SMSET module instantiated in the EVE subsystem. The VCOP start and done events are handled as generic events and encapsulated in STP messages with an event-ID and local timestamp and exported through the CT_STM module.

The EVE instrumentation scheme allows the user to understand micro-task dependencies, and potential bottlenecks. DMA transfer boundaries are reported as EVE events. Software messages from ARP32 execution can be interleaved with EVE VCOP events.

33.10.7 L3 NOC Statistics Collector

The L3_MAIN interconnect supports a built-in performance monitoring feature by implementing a statistics collector (NOC_SC) component, which computes traffic statistics within a user-defined window and periodically reports to the user through the CT_STM interface. Ten NOC_SC instances are instantiated in the device:

- One statistics collector dedicated to SDRAM load monitoring – SC_SDRAM (see Section 33.10.7.1, *L3 Target Load Monitoring*, for more information)
- Nine statistics collectors dedicated to L3 Master Latency Monitoring – SC_LAT0 through SC_LAT8 (see Section 33.10.7.2, *L3 Master Latency Monitoring* , for more information)

Statistics collectors (SC_SDRAM and SC_LAT) can report:

- Average burst length in bytes/packet per sampling window
- Average throughput in bytes/cycle
- Percent link occupancy on the request link (for store transactions) during a sampling window
- Percent link occupancy on the response link (for load transactions) during a sampling window
- Percent arbitration conflict cycles on the request link
- Percent initiator busy cycles on the response link
- Histogram of payload length in bytes (for example, 0–16, 16–32, 32–128) each sampling window.
- Histogram of quality of service (QoS) metric for IVA initiator (for example, low priority, high priority) each sampling window.

The performance metrics are interleaved with software instrumentation data at the L3_MAIN interconnect level.

The performance monitoring probes implement three main functions:

- Events detection
- Transactions filtering
- Aggregation

The probes can be configured to detect the NTTP and OCP link events summarized in Table 33-17.

Table 33-17. Performance Monitoring Events Detection

Link Event	NTTP	OCP	Definition
NONE	✓	✓	No event selected
ANY	✓	✓	Any clock cycles
TRANSFER	✓	✓	Word has been accepted by the receiver.
WAIT	✓	–	Transfer has been initiated but the transmitter currently has no data to send.
BUSY	✓	✓	Receiver applies flow control
PKT	✓	✓	Transfer of a new packet header
DATA	✓	✓	Transfer of a payload word
IDLES	✓	✓	No communication over the link
LATENCY	✓	–	Debug bit detection

The probes can be configured to filter the traffic based on the criteria summarized in Table 33-18.

Table 33-18. Performance Filtering Options

Filters	Comment
Master address	Mask and match
Slave address(2)	
UserInfo	
Read	Opcode is a load
Write	Opcode is a store
Error	Mask and match
OCP address(1)	

(1) SC_SDRAM only
(2) SC_LAT only

Table 33-19 specifies the master address mapping (all statistics collectors).

Table 33-19. Statistics Collector Master Address Mapping

Master-ID	Initiator
0x0	MPU
0x10	CS_DAP
0x14	IEEE1500_2_OCP
0x20	DSP1 MDMA
0x24	DSP1 CFG
0x28	DSP1 DMA
0x2C	DSP2 DMA
0x30	DSP2 CFG
0x34	DSP2 MDMA
0x3A	IVA ICONT1
0x42	EVE1 P1
0x46	EVE2 P1
0x4A	Reserved
0x4E	Reserved
0x50	Reserved
0x54	Reserved
0x58	Reserved
0x5C	Reserved
0x60	IPU1
0x64	IPU2
0x68	DMA_SYSTEM RD
0x6A	DMA_SYSTEM WR
0x6C	Reserved
0x6E	Reserved
0x70	EDMA_TC1 WR
0x72	EDMA_TC1 RD
0x74	EDMA_TC2 WR
0x76	EDMA_TC2 RD
0x80	DSS
0x84	MLB
0x86	MMU1
0x88	PCIe_SS1
0x8C	PCIe_SS2
0x8E	MMU2
0x90	VIP1 P1
0x92	VIP1 P2
0x94	VIP2 P1
0x96	VIP2 P2
0x98	VIP3 P1
0x9A	VIP3 P2
0x9C	VPE P1
0x9E	VPE P2
0xA0	MMC1
0xA2	GPU P1
0xA4	MMC2

Table 33-19. Statistics Collector Master Address Mapping (continued)

Master-ID	Initiator
0xA6	GPU P2
0xA8	BB2D P1
0xAA	BB2D P2
0xAC	GMAC_SW
0xAE	Reserved
0xB0	USB4
0xB4	USB1
0xB8	USB2
0xBC	USB3
0xC0	Reserved
0xC4	Reserved
0xCC	SATA
0xD2	EVE1 P2
0xD6	EVE2 P2
0xDA	Reserved
0xDE	Reserved

Table 33-20 specifies the slave address mapping (SC_LAT only).

Table 33-20. Statistics Collector Slave Address Mapping

Slave	Address (hex)
Reserved	0x0
Reserved	0x1
DMM P1	0x2
DMM P2	0x3
DSP1 SDMA	0x4
DSP2 SDMA	0x5
DSS	0x6
EVE1	0x7
EVE2	0x8
Reserved	0x9
Reserved	0xA
BB2D	0xB
GPMC	0xC
GPU	0xD
HOST_CLK1_1	0xE
HOST_CLK1_2	0xF
IPU1	0x10
IPU2	0x11
IVA CONFIG	0x12
IVA SL2IF	0x13
L4_CFG	0x14
L4_PER1 P1	0x15
L4_PER1 P2	0x16
L4_PER1 P3	0x17
L4_PER2 P1	0x18

Table 33-20. Statistics Collector Slave Address Mapping (continued)

Slave	Address (hex)
L3_INSTR	0x19
L4_PER2 P3	0x1A
L4_PER3 P1	0x1B
L4_PER3 P2	0x1C
L4_PER3 P3	0x1D
L4_WKUP	0x1E
McASP1	0x1F
McASP2	0x20
McASP3	0x21
MMU1	0x22
MMU2	0x23
OCMC_RAM1	0x24
OCMC_RAM2	0x25
OCMC_RAM3	0x26
Reserved	0x27
PCIe_SS1	0x28
PCIe_SS2	0x29
Reserved	0x2A
Reserved	0x2B
Reserved	0x2C
Reserved	0x2D
Reserved	0x2E
Reserved	0x2F
EDMA_TPCC	0x30
EDMA_TC1	0x31
EDMA_TC2	0x32
Reserved	0x35
VCP1	0x36
VCP2	0x37
QSPI	0x39
HOST_CLK2_1	0x40
DEBUGSS CT_TBR	0x41
L4_PER2 P2	0x42

The probes implement a user-defined set of counters that aggregate the events sampled by the detector and filtered according to the user setup.

Note

Statistics collectors counter values are accessible by application software.

Table 33-21 summarizes the performance probe aggregation modes.

Table 33-21. Aggregation Modes

Aggregation Mode	Description
FILTER_HIT	The counter increments by 1 when the filter hits.
MIN_MAX_HIT	The counter increments by 1 when the filter hits and the selected event information is within range.

Table 33-21. Aggregation Modes (continued)

Aggregation Mode	Description
	– Payload length (bytes) – Pressure value – Request/response latency (clock cycles)
EVT_INFO	The selected event information is added to the counter value when the filter hits. Payload length (bytes) Pressure value Request/response latency (clock cycles)
AND_FILTER	The counter increments by 1 when all unit filters hit.
OR_FILTER	The counter increments by 1 when at least one unit filter hits.
SUM_REQ_EVT	The counter sums the events from any request port.
SUM_RSP_EVT	The counter sums the events from any response port.
SUM_ALL_EVT	The counter sums the events from any port.
EXT_EVT	The counter increments by 1 when selected external event input signal is sampled high.

33.10.7.1 L3 Target Load Monitoring

The L3_MAIN interconnect implements five performance monitoring probes on DDR memory channels. The traffic statistics are computed within a user-defined window and periodically reported to the user through the CT_STM interface.

SC_SDRAM supports the following main features:

- Four probe inputs:
  - Probe 0 (128 bit-wide) – EMIF1_SYS
  - Probe 1 (128 bit-wide) – EMIF2_SYS
  - Probe 2 (128 bit-wide) – MPU_MA_P1
  - Probe 3 (128 bit-wide) – MPU_MA_P2
- Eight 32-bit counters shared concurrently:
  - Counter 0 with two filters
  - Counter 1 with one filter
  - Counter 2 with two filters
  - Counter 3 with one filter
  - Counter 4 with one filter
  - Counter 5 with one filter
  - Counter 6 with one filter
  - Counter 7 with one filter
- Simple (with one element) or complex (with several elements) filters available
- Filtering according to:
  - Initiator of traffic
  - Access priorities
  - OCP address
- No latency counter. Only bandwidth measurement on this collector.
- 32-bit collecting window counter
- Dump identifier is 0x0 (tie-off value)
- Dumps frames at L3_MAIN interconnect slave address 0x19 (L3_INSTR)

Table 33-22 shows the SC_SDRAM port mapping.

Table 33-22. SC_SDRAM Port Mapping

Probe	Description	Link	Port
0	EMIF1_SYS	OCP REQ	0
		OCP RSP	1

Table 33-22. SC_SDRAM Port Mapping (continued)

Probe	Description	Link	Port
1	EMIF2_SYS	OCP REQ	2
		OCP RSP	3
2	MPU_MA_P1	OCP REQ	4
		OCP RSP	5
3	MPU_MA_P2	OCP REQ	6
		OCP RSP	7

33.10.7.2 L3 Master Latency Monitoring

The L3 interconnect implements performance monitoring probes on initiator (master) interfaces. The master latency statistics are computed within a user-defined window and periodically reported to the user through the CT_STM interface.

The probes can be configured to filter latencies in four classes and report to the user a latency distribution along execution.

Because the performance metrics and the software events are exported through a unified export channel, it is possible to correlate latency trends with ongoing execution and system context.

Because computing latency requires maintaining the state between request and response ports, the probe cannot compute latency statistics on 100 percent of the initiator traffic. Hence, latency histograms must be extracted on large execution windows to be accurate.

33.10.7.2.1 SC_LAT0 Configuration

SC_LAT0 supports the following main features:

- Six probe inputs:
  - Probe 0: MPU
  - Probe 1: MMU1
  - Probe 2: EDMA_TC0_RD
  - Probe 3: EDMA_TC0_WR
  - Probe 4: EDMA_TC1_RD
  - Probe 5: EDMA_TC1_WR
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x1 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-23 shows the SC_LAT0 port mapping.

Table 33-23. SC_LAT0 Port Mapping

Probe	Description	Link	Port
0	MPU	NTTP REQ	0
		NTTP RSP	1
1	MMU1	NTTP REQ	2
		NTTP RSP	3

Table 33-23. SC_LAT0 Port Mapping (continued)

Probe	Description	Link	Port
2	EDMA_TC0_RD	NTTP REQ	4
		NTTP RSP	5
3	EDMA_TC0_WR	NTTP REQ	6
		NTTP RSP	7
4	EDMA_TC1_RD	NTTP REQ	8
		NTTP RSP	9
5	EDMA_TC1_WR	NTTP REQ	10
		NTTP RSP	11

33.10.7.2.2 SC_LAT1 Configuration

SC_LAT1 supports the following main features:

- Eight probe inputs:
  - Probe 0: VIP1 P1
  - Probe 1: VIP1 P2
  - Probe 2: VIP2 P1
  - Probe 3: VIP2 P2
  - Probe 4: VIP3 P1
  - Probe 5: VIP3 P2
  - Probe 6: VPE P1
  - Probe 7: VPE P2
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x2 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT1 port mapping.

Table 33-24. SC_LAT1 Port Mapping

Probe	Description	Link	Port
0	VIP1 P1	NTTP REQ	0
		NTTP RSP	1
1	VIP1 P2	NTTP REQ	2
		NTTP RSP	3
2	VIP2 P1	NTTP REQ	4
		NTTP RSP	5
3	VIP2 P2	NTTP REQ	6
		NTTP RSP	7
4	VIP3 P1	NTTP REQ	8
		NTTP RSP	9
5	VIP3 P2	NTTP REQ	10
		NTTP RSP	11

Table 33-24. SC_LAT1 Port Mapping (continued)

Probe	Description	Link	Port
6	VPE P1	NTTP REQ	12
		NTTP RSP	13
7	VPE P2	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.3 SC_LAT2 Configuration

SC_LAT2 supports the following main features:

- Eight probe inputs:
  - Probe 0: EVE1 TC0
  - Probe 1: EVE1 TC1
  - Probe 2: EVE2 TC0
  - Probe 3: EVE2 TC1
  - Probe 4: Reserved
  - Probe 5: Reserved
  - Probe 6: Reserved
  - Probe 7: Reserved
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x3 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT2 port mapping.

Table 33-25. SC_LAT2 Port Mapping

Probe	Description	Link	Port
0	EVE1 TC0	NTTP REQ	0
		NTTP RSP	1
1	EVE1 TC1	NTTP REQ	2
		NTTP RSP	3
2	EVE2 TC0	NTTP REQ	4
		NTTP RSP	5
3	EVE2 TC1	NTTP REQ	6
		NTTP RSP	7
4	Reserved	NTTP REQ	8
		NTTP RSP	9
5	Reserved	NTTP REQ	10
		NTTP RSP	11
6	Reserved	NTTP REQ	12
		NTTP RSP	13

Table 33-25. SC_LAT2 Port Mapping (continued)

Probe	Description	Link	Port
7	Reserved	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.4 SC_LAT3 Configuration

SC_LAT3 supports the following main features:

- Eight probe inputs:
  - Probe 0: DSP1 MDMA
  - Probe 1: DSP1 EDMA
  - Probe 2: DSP2 MDMA
  - Probe 3: DSP2 EDMA
  - Probe 4: IVA
  - Probe 5: GPU P1
  - Probe 6: GPU P2
  - Probe 7: BB2D P1
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x4 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT3 port mapping.

Table 33-26. SC_LAT3 Port Mapping

Probe	Description	Link	Port
0	DSP1 MDMA	NTTP REQ	0
		NTTP RSP	1
1	DSP1 EDMA	NTTP REQ	2
		NTTP RSP	3
2	DSP2 MDMA	NTTP REQ	4
		NTTP RSP	5
3	DSP2 EDMA	NTTP REQ	6
		NTTP RSP	7
4	IVA	NTTP REQ	8
		NTTP RSP	9
5	GPU P1	NTTP REQ	10
		NTTP RSP	11
6	GPU P2	NTTP REQ	12
		NTTP RSP	13
7	BB2D P1	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.5 SC_LAT4 Configuration

SC_LAT4 supports the following main features:

- Eight probe inputs:
  - Probe 0: DSS
  - Probe 1: Reserved
  - Probe 2: MMU2
  - Probe 3: IPU1
  - Probe 4: IPU2
  - Probe 5: DMA SYSTEM RD
  - Probe 6: DMA SYSTEM WR
  - Probe 7: Reserved
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x5 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT4 port mapping.

Table 33-27. SC_LAT4 Port Mapping

Probe	Description	Link	Port
0	DSS	NTTP REQ	0
		NTTP RSP	1
1	Reserved	NTTP REQ	2
		NTTP RSP	3
2	MMU2	NTTP REQ	4
		NTTP RSP	5
3	IPU1	NTTP REQ	6
		NTTP RSP	7
4	IP U2	NTTP REQ	8
		NTTP RSP	9
5	DMA SYSTEM RD	NTTP REQ	10
		NTTP RSP	11
6	DMA SYSTEM WR	NTTP REQ	12
		NTTP RSP	13
7	Reserved	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.6 SC_LAT5 Configuration

SC_LAT5 supports the following main features:

- Eight probe inputs:
  - Probe 0: USB1
  - Probe 1: USB2

  - Probe 2: USB3
  - Probe 3: USB4
  - Probe 4: PCIe_SS1
  - Probe 5: PCIe_SS2
  - Probe 6: DSP1 CFG
  - Probe 7: DSP2 CFG
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x6 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT5 port mapping.

Table 33-28. SC_LAT5 Port Mapping

Probe	Description	Link	Port
0	USB1	NTTP REQ	0
		NTTP RSP	1
1	USB2	NTTP REQ	2
		NTTP RSP	3
2	USB3	NTTP REQ	4
		NTTP RSP	5
3	USB4	NTTP REQ	6
		NTTP RSP	7
4	PCIe_SS1	NTTP REQ	8
		NTTP RSP	9
5	PCIe_SS2	NTTP REQ	10
		NTTP RSP	11
6	DSP1 CFG	NTTP REQ	12
		NTTP RSP	13
7	DSP2 CFG	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.7 SC_LAT6 Configuration

SC_LAT6 supports the following main features:

- Seven probe inputs:
  - Probe 0: GMAC_SW
  - Probe 1: Reserved
  - Probe 2: Reserved
  - Probe 3: Reserved
  - Probe 4: Reserved
  - Probe 5: Reserved
  - Probe 6: Reserved
  - Probe 7: MPU

- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x7 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT6 port mapping.

Table 33-29. SC_LAT6 Port Mapping

Probe	Description	Link	Port
0	GMAC_SW	NTTP REQ	0
		NTTP RSP	1
1	Reserved	NTTP REQ	2
		NTTP RSP	3
2	Reserved	NTTP REQ	4
		NTTP RSP	5
3	Reserved	NTTP REQ	6
		NTTP RSP	7
4	Reserved	NTTP REQ	8
		NTTP RSP	9
5	Reserved	NTTP REQ	10
		NTTP RSP	11
6	Reserved	NTTP REQ	12
		NTTP RSP	13
7	MPU	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.8 SC_LAT7 Configuration

SC_LAT7 supports the following main features:

- Eight probe inputs:
  - Probe 0: MMC1
  - Probe 1: MMC2
  - Probe 2: SATA
  - Probe 3: MLB
  - Probe 4: BB2D_P2
  - Probe 5: IEEE1500
  - Probe 6: DEBUGSS
  - Probe 7: VCP1
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:

  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x8 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT7 port mapping.

Table 33-30. SC_LAT7 Port Mapping

Probe	Description	Link	Port
0	MMC1	NTTP REQ	0
		NTTP RSP	1
1	MMC2	NTTP REQ	2
		NTTP RSP	3
2	SATA	NTTP REQ	4
		NTTP RSP	5
3	MLB	NTTP REQ	6
		NTTP RSP	7
4	BB2D_P2	NTTP REQ	8
		NTTP RSP	9
5	IEEE1500	NTTP REQ	10
		NTTP RSP	11
6	DEBUGSS	NTTP REQ	12
		NTTP RSP	13
7	VCP1	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.9 SC_LAT8 Configuration

SC_LAT8 supports the following main features:

- Eight probe inputs:
  - Probe 0: OCMC_RAM1
  - Probe 1: OCMC_RAM2
  - Probe 2: OCMC_RAM3
  - Probe 3: GPMC
  - Probe 4: McASP1
  - Probe 5: McASP2
  - Probe 6: McASP3
  - Probe 7: VCP2
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x9 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 33-24 shows the SC_LAT8 port mapping.

Table 33-31. SC_LAT8 Port Mapping

Probe	Description	Link	Port
0	OCMC_RAM1	NTTP REQ	0
		NTTP RSP	1
1	OCMC_RAM2	NTTP REQ	2
		NTTP RSP	3
2	OCMC_RAM3	NTTP REQ	4
		NTTP RSP	5
3	GPMC	NTTP REQ	6
		NTTP RSP	7
4	McASP1	NTTP REQ	8
		NTTP RSP	9
5	McASP2	NTTP REQ	10
		NTTP RSP	11
6	McASP3	NTTP REQ	12
		NTTP RSP	13
7	VCP2	NTTP REQ	14
		NTTP RSP	15

33.10.7.2.10 Statistics Collector Alarm Mode

Statistic collectors can be used to provide the application software with information about the NoC or SDRAM reaching corner cases (for example, too much traffic for a given OPP) while in application mode where for an end product use case the debug subsystem (which is normally exporting the statistic frames) is off. An interrupt-based scheme using a dedicated signal (L3_MAIN_IRQ_STAT_ALARM) is implemented to avoid CPU polling periodically statistic registers. This interrupt alert is fired when a given metric is out of specified range (below the programmed MIN threshold or above the programmed MAX threshold). The L3_MAIN_IRQ_STAT_ALARM interrupt is connected to the IRQ_CROSSBAR_16 input and the user can map it to any device INTC via Control Module.

Note

NTTP statistic collectors (SC_LAT) support latency measurement. However, comparison cannot be done on the latency counter. In case of latency measurement the comparison has to be done on number of latencies in user defined range, not on the latency value itself.

33.10.7.2.11 Statistics Collector Suspend Mode

The statistics collector module implements a suspend input that is used to avoid statistics collector counters to be updated while the processor has entered the debug state. This avoids triggering false alerts upon execution resume. When the statistics collector is asserted to 1, it freezes the monitoring process. When it goes back to 0, the monitoring resumes. If a frame is being dumped, it will not be stopped by suspend.

Note

Each statistic collector has an ignore suspend register, which can be used to disable the suspend feature.

33.10.8 PM Instrumentation

The device takes advantage of the system trace infrastructure to provide visibility to the user about the major power-management events. This is supported through a PMI module (PM profiler) instantiated in the

PRM module. The PRM state changes are handled as generic events and encapsulated in STP hardware messages and exported through the CT_STM module. The nature of the PM events does not require accurate timestamping and thus, timestamping is handled at CT_STM or trace receiver level.

The PM events are organized by class. Any PM state change from a specific class refreshes the entire instrumentation frame associated with that class. The STP message structure includes a PM event-ID indicating the class of the PM events.

The PM event classes supported are:

- Logic voltage domain OPP change
- Memory voltage domain OPP change
- Logic power domain state change
- Memory power domain state change

The PMI has a unique hardware master-ID assigned to it (see Section 33.10.10, *Master-ID Encoding*).

The PMI supports the possibility to report on activity in different event classes in the same sampling window. The user can size the capture sampling window.

Software events from the PM routines instrumentation can be interleaved with the PM hardware events. The user can take advantage of that feature to understand latencies for a specific power-management scenario or strategy.

The PM module implements an instrumentation port that directly interfaces with the debug subsystem (used to export PM events to CT_STM).

33.10.9 CM Instrumentation

The device instantiates a CMI module (CM profiler). It has a unique hardware master-ID assigned to it (for more information, see Section 33.10.10, *Master-ID Encoding*).

Note

Only CM_CORE_AON instrumentation is supported in this device. CM_CORE instrumentation is not supported.

The CM events profiling is similar to the PMI. Two exclusive instrumentation modes are supported:

- Clock activity:
  – Exposes to the user a snapshot of the state of all the clock domains derived from the same DPLL when CM detects a state change in the clock domain
  – Exposes to the user a snapshot of the DPLL settings when the CM signals a DPLL programming
- Module activity:
  – Exposes to the user periodically the active cycles count of the target modules
  – Exposes to the user periodically the active cycles count of the initiator modules

It provides visibility to the user about the state of the major clock domains along the application code execution. The STP message reports the effective state of the clock domain and therefore can highlight scenarios where a particular dependency is preventing the clock domain from being switched off.

The CM event classes supported are:

- Events capture mode – four classes:
  – Clock domain state
  – DPLL settings update
  – Clock frequency divider ratio
  – Clock source selection update
- Module activity collection mode – two classes:
  – Target module activity

– Initiator module activity

When in events capture mode, the CMI supports the possibility of reporting on activity in different event classes in the same sampling window. The user can size the capture sampling window.

33.10.10 Master-ID Encoding

A master-ID (MReqMstID[7:0]) field is used by the CT_STM to encode a MASTER type of message:

- MReqMstID[7] – Differentiates software versus hardware masters (0 for software masters; 1 for hardware masters)
- MReqMstID[6:2] – Master address exported by the L3_MAIN interconnect debug subsystem target
- MReqMstID[1:0] – Additional qualifier for multicore masters (MPU, IPU)

33.10.10.1 Software Masters

The CT_STM module allows:

- Enabling a maximum of four SoC software masters
- Masking MReqMstID[1:0]
- Differentiating multicore software masters through MReqMstID[1:0]

Table 33-32 summarizes the software initiators that can export trace messages through the CT_STM.

Table 33-32. STM Message Software Masters

Initiator	MReqMstID				Restriction/Comment
	ConnID	[7]	[6:2]	[1:0]	
MPU (SMP)	0x1	0	00001	00	MPU_C0
				01	MPU_C1
CS_DAP	0x4	0	00100	–	STP link testing
DSP1 MDMA	0x8	0	01000	–	
DSP1 CFG	0x9	0	01001	–	
DSP1 DMA	0xA	0	01010	–	Data logging
DSP2 DMA	0xB	0	01011	–	Data logging
DSP2 CFG	0xC	0	01100	–	
DSP2 MDMA	0xD	0	01101	–	
IVA	0xE	0	01110		Software messages routed through the IVA instrumentation port
EVE1 P1	0x10	0	10000	–	
EVE2 P1	0x11	0	10001	–	
IPU1	0x18	0	11000	00	IPU1_C0
				01	IPU1_C1
IPU2	0x19	0	11001	00	IPU2_C0
				01	IPU2_C1
DMA_SYSTEM_WR	0x1A	0	11010	–	Data logging
EDMA_TC1 WR	0x1C	0	11100	–	Data logging
EDMA_TC2 WR	0x1D	0	11101	–	Data logging

33.10.10.2 Hardware Masters

The CT_STM module allows enabling a subset of SoC hardware masters (maximum = 4).

Table 33-33 summarizes the hardware initiators that can export trace messages through the STM.

Table 33-33. STM Message Hardware Masters

Initiator	MReqMstID			
	ConnID	[7]	[6:2]	[1:0]

Table 33-33. STM Message Hardware Masters (continued)

Initiator	MReqMstID			
EVE1 instrumentation (SMSET)	0x34	1	10100	00
EVE2 instrumentation (SMSET)	0x35	1	10101	00
SC_SDRAM (STATCOLL_0)	0x22	1	00010	00
SC_LAT0 (STATCOLL_1)	0x2C	1	01100	00
SC_LAT1 (STATCOLL_2)	0x2D	1	01101	00
SC_LAT2 (STATCOLL_3)	0x2E	1	01110	00
SC_LAT3 (STATCOLL_4)	0x2F	1	01111	00
SC_LAT4 (STATCOLL_5)	0x30	1	10000	00
SC_LAT5 (STATCOLL_6)	0x31	1	10001	00
SC_LAT6 (STATCOLL_7)	0x32	1	10010	00
SC_LAT7 (STATCOLL_8)	0x33	1	10011	00
SC_LAT8 (STATCOLL_9)	0x3C	1	11100	00
OCP watchpoint (OCP_WP_NOC)	0x39	1	11001	00
DMA profiling (OCP_WP_NOC)	0x3A	1	11010	00
System events (OCP_WP_NOC)	0x3B	1	11011	00
IVA instrumentation (SMSET)	0x3C	1	11100	00
PMI events profiling	0x3D	1	11101	00
CMI events profiling	0x3E	1	11110	00

33.11 Concurrent Debug Modes

The debugger or application software can program the DRM to route a specific debug function to each device debug port pin.

Because of the limited number of pins allocated to debug, debug and trace source signals are multiplexed.

Table 33-34 summarizes the trace port configuration.

Table 33-34. Trace Port Configuration

Pin Name	Internal Signal Name	I/O	Trigger	CS_TPIU (CS_PTM, DSP, CS_STM, CT_STM)	
				Mode A (up to 16 data pins)	Mode B[(1)] (up to 18 data pins)
emu19	EMU19	O		TRACEDATA[15]	TRACEDATA[17]
emu18	EMU18	O		TRACEDATA[14]	TRACEDATA[16]
emu17	EMU17	O		TRACEDATA[13]	TRACEDATA[15]
emu16	EMU16	O		TRACEDATA[12]	TRACEDATA[14]
emu15	EMU15	O		TRACEDATA[11]	TRACEDATA[13]
emu14	EMU14	O		TRACEDATA[10]	TRACEDATA[12]
emu13	EMU13	O		TRACEDATA[9]	TRACEDATA[11]
emu12	EMU12	O		TRACEDATA[8]	TRACEDATA[10]
emu11	EMU11	O		TRACEDATA[7]	TRACEDATA[9]
emu10	EMU10	O		TRACEDATA[6]	TRACEDATA[8]
emu9	EMU9	O		TRACEDATA[5]	TRACEDATA[7]
emu8	EMU8	O		TRACEDATA[4]	TRACEDATA[6]
emu7	EMU7	O		TRACEDATA[3]	TRACEDATA[5]
emu6	EMU6	O		TRACEDATA[2]	TRACEDATA[4]
emu5	EMU5	O		TRACEDATA[1]	TRACEDATA[3]
emu4	EMU4	O		TRACEDATA[0]	TRACEDATA[2]
emu3	EMU3	O		TRACECTL	TRACECTL
emu2	EMU2	O		TRACECLK	TRACECLK
emu1	EMU1	I/O	Trigger1		TRACEDATA[1]
emu0	EMU0	I/O	Trigger0		TRACEDATA[0]

(1) This is the recommended mode for primary use.

Note

The configuration of the trace port must comply with Table 33-34; otherwise, it will be ignored by DRM hardware. For example, if Trigger0 is programmed on EMU3 and Trigger1 is programmed on EMU4, this configuration will be ignored.

Table 33-35 summarizes the concurrent debug and trace in the device.

Table 33-35. Concurrent Debug and Trace

Debug Use Case	Concurrent Debug Flows	Debug Pins	Trace Pins		
		Triggers	Data	Control	Clock
1	CS_PTM		16	1	1
	CT_STM				
	Triggers	2			

Table 33-35. Concurrent Debug and Trace (continued)

Debug Use Case	Concurrent Debug Flows	Debug Pins	Trace Pins		
		Triggers	Data	Control	Clock
2	CS_PTM		16	1	1
	CT_STM		1	–	1
	Triggers				
3	CS_PTM		8	1	1
	CT_STM		4	–	1
	Triggers	2			
4	CS_PTM		11	1	1
	CT_STM		4	–	1
	Triggers	2			

33.12 DRM Register Manual

33.12.1 DRM Instance Summary

Table 33-36. DRM Instance Summary

Module Name	Module Base Address	Size
DRM	0x5416 0000	588 Bytes

33.12.2 DRM Registers

33.12.2.1 DRM Register Summary

Note

See Table 33-8 for DRM suspend control registers assignment to corresponding peripherals.

Table 33-37. DRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DRM Base Address
RESERVED	R	32	0x0000 0000 – 0x0000 00CC	0x5416 0000 – 0x5416 00CC
DRM_SUSPEND_CTRL0	RW	32	0x0000 0200	0x5416 0200
DRM_SUSPEND_CTRL1	RW	32	0x0000 0204	0x5416 0204
DRM_SUSPEND_CTRL2	RW	32	0x0000 0208	0x5416 0208
DRM_SUSPEND_CTRL3	RW	32	0x0000 020C	0x5416 020C
DRM_SUSPEND_CTRL4	RW	32	0x0000 0210	0x5416 0210
DRM_SUSPEND_CTRL5	RW	32	0x0000 0214	0x5416 0214
DRM_SUSPEND_CTRL6	RW	32	0x0000 0218	0x5416 0218
DRM_SUSPEND_CTRL7	RW	32	0x0000 021C	0x5416 021C
DRM_SUSPEND_CTRL8	RW	32	0x0000 0220	0x5416 0220
DRM_SUSPEND_CTRL9	RW	32	0x0000 0224	0x5416 0224
DRM_SUSPEND_CTRL10	RW	32	0x0000 0228	0x5416 0228
DRM_SUSPEND_CTRL11	RW	32	0x0000 022C	0x5416 022C
DRM_SUSPEND_CTRL12	RW	32	0x0000 0230	0x5416 0230
DRM_SUSPEND_CTRL13	RW	32	0x0000 0234	0x5416 0234
DRM_SUSPEND_CTRL14	RW	32	0x0000 0238	0x5416 0238
DRM_SUSPEND_CTRL15	RW	32	0x0000 023C	0x5416 023C
DRM_SUSPEND_CTRL16	RW	32	0x0000 0240	0x5416 0240
DRM_SUSPEND_CTRL17	RW	32	0x0000 0244	0x5416 0244

Table 33-37. DRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DRM Base Address
DRM_SUSPEND_CTRL18	RW	32	0x0000 0248	0x5416 0248
DRM_SUSPEND_CTRL19	RW	32	0x0000 024C	0x5416 024C
DRM_SUSPEND_CTRL20	RW	32	0x0000 0250	0x5416 0250
DRM_SUSPEND_CTRL21	RW	32	0x0000 0254	0x5416 0254
DRM_SUSPEND_CTRL22	RW	32	0x0000 0258	0x5416 0258
DRM_SUSPEND_CTRL23	RW	32	0x0000 025C	0x5416 025C
DRM_SUSPEND_CTRL24	RW	32	0x0000 0260	0x5416 0260
DRM_SUSPEND_CTRL25	RW	32	0x0000 0264	0x5416 0264
DRM_SUSPEND_CTRL26	RW	32	0x0000 0268	0x5416 0268
DRM_SUSPEND_CTRL27	RW	32	0x0000 026C	0x5416 026C
DRM_SUSPEND_CTRL28	RW	32	0x0000 0270	0x5416 0270
DRM_SUSPEND_CTRL29	RW	32	0x0000 0274	0x5416 0274
DRM_SUSPEND_CTRL30	RW	32	0x0000 0278	0x5416 0278
DRM_SUSPEND_CTRL31	RW	32	0x0000 027C	0x5416 027C
DRM_SUSPEND_CTRL32	RW	32	0x0000 0280	0x5416 0280
DRM_SUSPEND_CTRL33	RW	32	0x0000 0284	0x5416 0284
DRM_SUSPEND_CTRL34	RW	32	0x0000 0288	0x5416 0288
DRM_SUSPEND_CTRL35	RW	32	0x0000 028C	0x5416 028C
DRM_SUSPEND_CTRL36	RW	32	0x0000 0290	0x5416 0290
DRM_SUSPEND_CTRL37	RW	32	0x0000 0294	0x5416 0294
RESERVED	R	32	0x0000 0400	0x5416 0400
RESERVED	R	32	0x0000 0500	0x5416 0500

33.12.2.2 DRM Register Description

Table 33-38. DRM_SUSPEND_CTRL0

Address Offset	0x0000 0200		
Physical Address	0x5416 0200	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-39. DRM_SUSPEND_CTRL1

Address Offset	0x0000 0204		
Physical Address	0x5416 0204	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-40. DRM_SUSPEND_CTRL2

Address Offset	0x0000 0208		
Physical Address	0x5416 0208	**Instance**	DRM
Description			
Type	RW		

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td colspan="5">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
</table>

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-41. DRM_SUSPEND_CTRL3

Address Offset	0x0000 020C		
Physical Address	0x5416 020C	**Instance**	DRM
Description			

Table 33-41. DRM_SUSPEND_CTRL3 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-42. DRM_SUSPEND_CTRL4

Address Offset	0x0000 0210		
Physical Address	0x5416 0210	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-43. DRM_SUSPEND_CTRL5

Address Offset	0x0000 0214		
Physical Address	0x5416 0214	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-44. DRM_SUSPEND_CTRL6

Address Offset	0x0000 0218		
Physical Address	0x5416 0218	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-45. DRM_SUSPEND_CTRL7

Address Offset	0x0000 021C		
Physical Address	0x5416 021C	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-46. DRM_SUSPEND_CTRL8

Address Offset	0x0000 0220		
Physical Address	0x5416 0220	**Instance**	DRM
Description			

Table 33-46. DRM_SUSPEND_CTRL8 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-47. DRM_SUSPEND_CTRL9

Address Offset	0x0000 0224		
Physical Address	0x5416 0224	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-48. DRM_SUSPEND_CTRL10

Address Offset	0x0000 0228		
Physical Address	0x5416 0228	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-49. DRM_SUSPEND_CTRL11

Address Offset	0x0000 022C		
Physical Address	0x5416 022C	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-50. DRM_SUSPEND_CTRL12

Address Offset	0x0000 0230		
Physical Address	0x5416 0230	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-51. DRM_SUSPEND_CTRL13

Address Offset	0x0000 0234		
Physical Address	0x5416 0234	**Instance**	DRM
Description			

Table 33-51. DRM_SUSPEND_CTRL13 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-52. DRM_SUSPEND_CTRL14

Address Offset	0x0000 0238		
Physical Address	0x5416 0238	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-53. DRM_SUSPEND_CTRL15

Address Offset	0x0000 023C		
Physical Address	0x5416 023C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-54. DRM_SUSPEND_CTRL16

Address Offset	0x0000 0240		
Physical Address	0x5416 0240	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-55. DRM_SUSPEND_CTRL17

Address Offset	0x0000 0244		
Physical Address	0x5416 0244	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-56. DRM_SUSPEND_CTRL18

Address Offset	0x0000 0248		
Physical Address	0x5416 0248	**Instance**	DRM
Description			

Table 33-56. DRM_SUSPEND_CTRL18 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-57. DRM_SUSPEND_CTRL19

Address Offset	0x0000 024C		
Physical Address	0x5416 024C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-58. DRM_SUSPEND_CTRL20

Address Offset	0x0000 0250		
Physical Address	0x5416 0250	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-59. DRM_SUSPEND_CTRL21

Address Offset	0x0000 0254		
Physical Address	0x5416 0254	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-60. DRM_SUSPEND_CTRL22

Address Offset	0x0000 0258		
Physical Address	0x5416 0258	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-61. DRM_SUSPEND_CTRL23

Address Offset	0x0000 025C		
Physical Address	0x5416 025C	**Instance**	DRM
Description			

Table 33-61. DRM_SUSPEND_CTRL23 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-62. DRM_SUSPEND_CTRL24

Address Offset	0x0000 0260		
Physical Address	0x5416 0260	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-63. DRM_SUSPEND_CTRL25

Address Offset	0x0000 0264		
Physical Address	0x5416 0264	**Instance**	DRM
Description			
Type	RW		

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9	8 7 6 5 4	3	2 1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-64. DRM_SUSPEND_CTRL26

Address Offset	0x0000 0268		
Physical Address	0x5416 0268	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-65. DRM_SUSPEND_CTRL27

Address Offset	0x0000 026C		
Physical Address	0x5416 026C	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-66. DRM_SUSPEND_CTRL28

Address Offset	0x0000 0270		
Physical Address	0x5416 0270	**Instance**	DRM
Description			

Table 33-66. DRM_SUSPEND_CTRL28 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-67. DRM_SUSPEND_CTRL29

Address Offset	0x0000 0274		
Physical Address	0x5416 0274	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-68. DRM_SUSPEND_CTRL30

Address Offset	0x0000 0278		
Physical Address	0x5416 0278	**Instance**	DRM
Description			
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-69. DRM_SUSPEND_CTRL31

Address Offset	0x0000 027C		
Physical Address	0x5416 027C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-70. DRM_SUSPEND_CTRL32

Address Offset	0x0000 0280		
Physical Address	0x5416 0280	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-71. DRM_SUSPEND_CTRL33

Address Offset	0x0000 0284		
Physical Address	0x5416 0284	**Instance**	DRM
Description			

Table 33-71. DRM_SUSPEND_CTRL33 (continued)

Type	RW

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-72. DRM_SUSPEND_CTRL34

Address Offset	0x0000 0288		
Physical Address	0x5416 0288	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-73. DRM_SUSPEND_CTRL35

Address Offset	0x0000 028C		
Physical Address	0x5416 028C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							SUSPEND_SEL					SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-74. DRM_SUSPEND_CTRL36

Address Offset	0x0000 0290		
Physical Address	0x5416 0290	**Instance**	DRM
Description			
Type	RW		

31–9	8–4	3	2–1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 33-75. DRM_SUSPEND_CTRL37

Address Offset	0x0000 0294		
Physical Address	0x5416 0294	**Instance**	DRM
Description	susp		
Type	RW		

31:9	8:4	3	2:1	0
RESERVED	SUSPEND_SEL	SUSPEND_DEFAULT_OVERRIDE	RESERVED	SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Chapter 34
Glossary

TI Glossary This glossary lists and explains terms, acronyms, and definitions.

Revision History

Changes from February 1, 2020 to May 30, 2024 (from Revision G (February 2020) to Revision H (May 2024))	Page
• Added Note with SDK link	257

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265